3
Trabalhos Relacionados

Neste capítulo são apresentados sistemas que provêem suporte a adaptação de conteúdo para aplicações móveis sensíveis a contexto, discutindo alguns aspectos das diferentes abordagens encontradas na literatura. As questões mais relevantes para esses sistemas dizem respeito a forma como as informações de contexto dos clientes são obtidas e como essas informações são utilizadas para definir o momento em que é necessário realizar adaptação e qual o tipo de adaptação. Assim, os principais pontos aqui discutidos estão relacionados à maneira como as adaptações são definidas e executadas, bem como quais as informações de contextos são consideradas para a determinação da adaptação de conteúdo adequada. Outros aspectos de interesse dizem respeito à interação dos clientes e servidores, como o tipo de comunicação suportado, e questões sobre mobilidade e desconexão dos clientes móveis, bem como a escalabilidade e flexibilidade dos sistemas.

3.1
Sistemas com adaptação de conteúdo para clientes móveis

A adaptação de conteúdo tem sido bastante explorada na literatura, e em geral, pode ser classificada em dois tipos: estática ou dinâmica [71, 22]. Na adaptação estática [12, 72, 73, 13], diferentes versões (pré-adaptadas) do conteúdo são armazenadas no servidor, e no momento do envio para os clientes móveis, o servidor escolhe a versão mais adequada às características do dispositivo móvel. Esta abordagem apresenta vantagens em relação ao tempo de envio e ao controle total do conteúdo pelo servidor, podendo adequar o conteúdo às especificidades dos dispositivos atendidos. Entretanto, esse tipo de solução não é adequada quando o volume de conteúdo e de condições de contexto aumenta, pois torna-se necessário a autoria de um grande número de versões do conteúdo. Além disso, o número de contextos que podem ser satisfatoriamente atendidos fica limitado ao número de versões do conteúdo existentes, e não suporta variações dinâmicas de contexto.

Em adaptação dinâmica [15, 16, 17, 74], o servidor armazena apenas a versão original do conteúdo, e este é adaptado a cada requisição (*on-the-fly*),

de acordo com o contexto corrente dos clientes, levando em consideração características (estáticas e dinâmicas) dos dispositivos, do enlace sem fio, usuário e do conteúdo. Esta abordagem não apresenta o problema de armazenamento de múltiplas versões (adaptação estática) no servidor, além de permitir um conjunto mais amplo e dinâmico de contextos. Entretanto o acesso ao conteúdo sofre um retardo devido às adaptações.

As tecnologias de adaptação dinâmica de conteúdo podem ser classificadas em três categorias principais: adaptação realizada no cliente, no servidor ou em proxies [75, 22]. As abordagens de adaptação nos pontos finais, no cliente [76] ou no servidor [21], consideram que adaptações mais complexas e otimizadas às necessidades dos clientes podem ser efetuadas nesses pontos. Entretanto, adaptações de conteúdo podem demandar uma quantidade considerável de recursos (CPU, memória), tornando inviável a execução nos clientes. Além disso, efetuar todas estas operações no servidor, para cada cliente móvel, torna-se pouco eficiente e escalável.

Um alternativa comumente adotada na literatura [22, 24, 77, 71, 78], e em alguns produtos comerciais [79], é o uso de proxies que atuam em benefício dos clientes, realizando as adaptações de conteúdo necessárias às variações de contexto do dispositivo, rede, etc. Os proxies podem estar localizados em diferentes pontos da rede fixa. Em [80], mostra-se que proxies próximos aos clientes, i.e. na fronteira da rede fixa com a rede sem fio (*edge side*), apresentam melhor desempenho que aqueles próximos ao servidor (*server-side*). Uma das vantagens da utilização de proxies é possibilitar a criação de clientes leves, e de simplificar o projeto dos servidores ou a integração com sistemas legados. Isto porque a complexidade da adaptação para otimizar a comunicação pelo meio sem fio é transferida para o proxy, que trata as questões relativas ao enlace sem fio, à mobilidade, à conectividade e às variações de contexto; além de compensar as diferenças de comunicação (por exemplo, vazão, latência, confiabilidade) entre as redes com e sem fio.

3.1.1 Proxies para adaptação de conteúdo

Nessa seção, são discutidas as abordagens adotadas por projetos que fornecem suporte ao desenvolvimento de aplicações móveis sensíveis a contexto, e que utilizam arquiteturas genéricas de proxy, que podem ser customizadas ou estendidas para atender às necessidades específicas de adaptação da aplicação. Alguns exemplos incluem: sistemas para aplicações multimídia, como RAPIDware [81] e Mobiware [82]; sistemas para aplicações Web baseadas em HTTP, como WBI [83, 84] e MobiPADS[85, 86]; e sistemas para adaptação

de conteúdo de propósito geral, como MARCH [27] e BARWAN [87] [1].

A decisão de quais adaptações são necessárias e em que momento aplicá-las é um dos principais aspectos abordado pelos diferentes sistemas. Esta decisão, em geral, é formulada através de **regras (ou políticas) de adaptação**. As regras estabelecem associações entre as situações de contextos de interesse (que funcionam como gatilhos) e as adaptações que devem ser aplicadas (disparadas) nestes momentos. Essas regras de decisão podem ser definidas de duas maneiras: através de interfaces programáveis, como no RAPIDware e BARWAN; ou por meio de configuração externa, como MARCH e WBI.

No primeiro caso, as condições nas quais cada adaptação será aplicada, por vezes, são expressas através de funções de utilidade (ou funções de satisfação de usuários), e devem ser programadas usando uma API provida pelo sistema, como em Mobiware. Algumas abordagens, como URICA [88], além das regras programadas, também utilizam técnicas de aprendizado baseadas em dados sobre a história de acesso (requisições) para definir automaticamente as necessidades de adaptação para os usuários.

No segundo caso, é realizada uma configuração externa na qual o desenvolvedor deve definir as regras que contêm as condições de disparo (gatilhos) de adaptações, descritas em termos de estados dos clientes e da rede (ou seja, contextos), as adaptações a serem executadas, e em alguns casos a prioridade da regra. As regras são configuradas, em geral, via arquivos usando na sua maioria as linguagens XML ou RuleML [89], como em MobiPADS e MARCH, respectivamente.

Em uma outra abordagem, apresentada por alguns middlewares, como CARISMA [90] e MobiPADS, após a definição inicial (programável ou configurável), é possível atualizar as regras de adaptação em tempo de execução através de mecanismos de reflexão e meta-interfaces. Essa característica de atualização das políticas de adaptação (regras) durante o funcionamento do sistema, sem a necessidade de pará-lo ou reiniciá-lo, é importante pois permite um funcionamento ininterrupto do mesmo, a despeito de novas situações (contextos) ou novos adaptadores (para novos formatos de conteúdo, etc).

No processo de **escolha das adaptações** que devem ser aplicadas, alguns sistemas adotam uma abordagem baseada nas preferências dos usuários [91, 92], na qual os usuários, através de interfaces ou arquivos de configuração, manipulam as regras que controlam a forma de reação das aplicações às mudanças de contexto. No sistema proposto em [93], a decisão de adaptação é baseada em uma pontuação das adaptações (ou combinação dessas). Essa

[1] As siglas dos projetos discutidos neste capítulo estão descritas em Abreviaturas e Siglas.

pontuação é previamente calculada sobre aspectos de qualidade (definidos manualmente pelos usuários) e no momento de envio da mensagem, a máquina de decisão tenta encontrar a combinação de adaptação com maior pontuação, mas que atenda às restrições de capacidade do dispositivo e da rede. Em CARISMA [90], as regras podem ser definidas pela aplicação cliente, mas se o serviço for requisitado por vários clientes, ele será adaptado de acordo com uma única regra. Assim se houver conflitos nas diferentes regras dos clientes, adota-se um mecanismo de solução de conflitos baseado em um leilão que escolhe a alternativa (regra) que obtiver um maior somatório de ofertas, onde cada oferta de cliente (para que a sua regra seja seguida) é calculada por uma função de utilidade que considera quesitos de disponibilidade, desempenho, segurança, etc.

Após a definição da adaptação adequada à situação de contexto do cliente, o ponto principal tratado pelos sistemas é a **execução da adaptação**. A adaptação pode ser realizada por um único elemento de adaptação, o adaptador (função ou serviço), ou por uma combinação de adaptadores. A maioria dos sistemas permite a composição de adaptadores, enquanto que alguns, como Mobiware e MARCH, também permitem a definição de prioridades e ordenação. Uma questão ainda em discussão é a forma de combinar diferentes adaptadores e encontrar a cadeia de adaptadores que melhor se adequa à tarefa de adaptação (composta) necessária à uma situação de contexto, dado que esta adaptação pode ser executada em vários passos e que pode haver um grande número de possíveis configurações (combinações de adaptadores) para adaptar o conteúdo.

Existem dois tipos principais de **mapeamento entre a ação de adaptação e o(s) adaptador(es)** que a realiza(m). O primeiro, e mais comum, é um mapeamento direto, no qual as regras indicam explicitamente o adaptador a ser utilizado, como na arquitetura CAP [94]. No segundo tipo, uma ação de adaptação pode ser realizada pela concatenação de vários adaptadores. Alguns trabalhos, como o DCAF [75] e o framework proposto em [95, 96], propõem algoritmos para a construção e otimização da cadeia de adaptadores a partir de um grafo de adaptação de conteúdo. A construção do grafo é baseada em perfis dos clientes, características da rede, e meta-dados do conteúdo e dos adaptadores disponíveis. Após a criação do grafo, é determinado, segundo algum critério de seleção, o caminho ótimo de execução da seqüência de adaptadores, que define quais os adaptadores serão utilizados e em que ordem. O critério de seleção pode ser um parâmetro de QoS, como o tempo total de execução da cadeia ou custo computacional, ou funções de satisfação do usuário.

Outro aspecto a ser observado é o tipo de **informação de contexto** utilizado para a definição das adaptações. O contexto usado pela maioria dos sistemas é definido por perfis de dispositivo e preferências do usuário, ou seja, essencialmente informações constantes ou fornecidas pelo usuário. A abordagem predominante para representação destes perfis é a descrição no padrão CC/PP, como em [27, 94, 93], e seus vocabulários UAProf (como em [95, 97]) ou UPS (como em [98]), ou o uso de outros esquemas baseados em RDF, como CSCP (utilizado em [99]). Com freqüência, são usados processos de monitoração para obter as parâmetros de qualidade do enlace sem fio (tais como largura de banda, retardo na transmissão, perdas), que são as principais variáveis de contexto utilizadas, como ocorre nos sistemas BARWAN, Mobiware e Rapidware. Alguns sistemas, como MobiPADS, utilizam um modelo de notificação de eventos para obter informações sobre mudanças nos estados do contexto de interesse. Este modelo permite a utilização de informações de contexto dinâmicas, como taxa de utilização da CPU e memória, ou energia disponível. O MobiPADS também permite uma composição de contextos (por exemplo, "`HighBandwidth = Network_MaxRate > 20000 bits/s AND Network_delay < 100 ms`").Na especificação das regras de adaptação, seria interessante o suporte à utilização/composição de contextos complexos (inferidos, derivados ou agregados de informações de contexto primitivas), e obtenção destes contextos de diferentes fontes ou serviços de contexto.

As características do enlace sem fio, como largura de banda limitada, alta taxa de erros e freqüentes flutuações na qualidade do canal de comunicação, representam mais um desafio para os sistemas para computação móvel. Alguns sistemas fornecem mecanismos para adaptar os protocolos de comunicação, como o protocolo Snoop [100], usado para aumentar o desempenho do TCP em redes sem fio com muitas perdas. Outros sistemas, como o Mobiware e o RAPIDware, utilizam filtros para correção de erros, como os filtros FEC (*Foward Error Correction*) para áudio e vídeo. Outra questão diz respeito à **mobilidade** dos clientes, que introduz requisitos adicionais para a manutenção da operação dos serviços nos clientes: o roteamento é mais complexo; o sistema precisa ser resistente a desconexões freqüentes e lidar com retransmissões de mensagens. O sistema Mobiware provê suporte a *handoff* [2] horizontal (entre domínios de um mesmo tipo de rede sem fio), enquanto que o BARWAN também permite handoff vertical, no qual o dispositivo cliente pode mover-se entre redes com diferentes tecnologias de comunicação sem fio, por exemplo WiFi e GPRS. Entretanto, estes sistemas não tratam da **desconexão**, consi-

[2]Também conhecido como *handover* ou mobilidade de terminal. Permite a um terminal cliente mover-se através de domínios de diferentes redes, mantendo a conexão ativa.

derando que sempre existe conectividade, mesmo que esta seja bastante fraca, para um handoff suave. Uma funcionalidade interessante para proxies para computação móvel é o suporte à desconexão, permitindo que as políticas de tratamento sejam especificadas de acordo com as necessidade da aplicação.

Outra característica desejável para sistemas de adaptação de conteúdo é sua **escalabilidade em relação ao número de clientes** e de adaptações necessárias para atendê-los. Diferentes abordagens são empregadas para aumentar a escalabilidade dos sistemas. O projeto Barwan emprega a abordagem de uma arquitetura proxy baseada em cluster, na qual os adaptadores (chamados *workers*) são processos criados, conforme a necessidade, em nós livres do cluster. Há um controle central (*front-end*) que identifica quais os adaptadores devem ser utilizados, e os instancia realizando um balanceamento de carga entre os nós. Outras abordagens consideram que as adaptações são executadas por serviços de adaptação distribuídos na rede, que podem ser implementados, por exemplo, como Web Services (como em [99]). Em [95, 96] um serviço de diretórios, como Jini [101], é utilizado para anunciar e descobrir os serviços de adaptação disponíveis. Cada intermediário possui um perfil que descreve os serviços de adaptação que ele contém (com informações dos formatos de entrada e saída, custo, etc). A adaptação de conteúdo é realizada pela composição destes serviços de adaptação. Essa composição é definida através de um algoritmo que otimiza uma função de satisfação do usuário, sujeita apenas a restrições de disponibilidade de largura de banda entre os adaptadores.

Estas abordagens lidam com a questão de escalabilidade por meio de um paralelismo externo, através da replicação e distribuição de tarefas de adaptação em diferentes nós capazes de realizar adaptação. Entretanto, apesar dos trabalhos acima serem um passo em direção à escalabilidade, os mesmos ainda não possuem uma solução adequada para o gerenciamento e o balanceamento de carga dos nós. Além disso, o problema de se calcular o grafo de adaptadores e o caminho ótimo (cadeia de adaptadores) é NP-completo [75], tendo portanto um custo de processamento elevado. Um ponto de retardo na adaptação, e que deveria ser levado em consideração na definição da cadeia de adaptadores é o custo de comunicação (transferência de conteúdo) entre os diversos nós da cadeia. Devido a esses fatores, é provável que a solução mais eficiente fosse ter um conjunto de proxies, cada um concentrando a execução das adaptações de cada requisição. Por fim, todo os sistemas apresentados tratam apenas de comunicação do tipo requisição/resposta, e não dão suporte a comunicação 1-para-muitos (como Pub/Sub) usada por aplicações de disseminação de conteúdo.

Nesta tese, é proposto um algoritmo para melhorar a escalabilida-

de interna de proxies de adaptação de conteúdo para comunicação requisição/resposta e Pub/Sub, com um atendimento a um grande número de clientes que recebem o mesmo conteúdo, mas que demandam adaptações específicas para cada cliente. A idéia é reduzir o número de execuções de adaptadores, agrupando clientes que compartilham o mesmo estado de contexto e portanto a mesma seqüência de adaptadores. Ou seja, separar os clientes com contextos similares em grupos, e aplicar as adaptações a estes grupos, ao invés de aplicá-las individualmente a cada cliente. Esta idéia de agrupar clientes baseado em seus contextos também é usada pelo sistema URICA [88]. Entretanto, este sistema utiliza o agrupamento para realizar predições de adaptação para um cliente, baseado na história da comunidade de clientes que compartilham o contexto. Segundo seus experimentos, agrupar os clientes em comunidades baseadas em contexto melhora o desempenho (qualidade de adaptação) do sistema de adaptação.

3.2 Sistemas Push

Os sistemas *Push* (ou de disseminação de conteúdo) [102, 103] oferecem entrega de conteúdo para muitos assinantes. O termo *push* é utilizado para enfatizar que o conteúdo é ativamente enviado aos clientes, em vez de ser requisitado (modelo *pull*). Os clientes (ou assinantes) registram interesse no serviço e a partir de então passam a receber os conteúdos que são publicados.

A maioria dos sistemas de disseminação de conteúdo não oferecem suporte a clientes móveis. A mobilidade dos clientes introduz desafios adicionais ao suporte de serviços de disseminação com qualidade aos clientes móveis, tais como roteamento de mensagens e adaptação de conteúdo. Os autores de [29, 104] propõem a arquitetura MobilePush para sistema *push* em ambientes móveis. O foco é a disseminação confiável de conteúdo, tratando aspectos de handoff horizontal e vertical, de perdas de mensagens e de roteamento e transmissão eficiente e confiável de mensagens para diferentes tecnologias de rede sem fio. A arquitetura é composta por três camadas: de comunicação Pub/Sub, de serviços e de aplicação. A camada de serviços provê serviços auxiliares como gerência de localização, adaptação de conteúdo, gerência de armazenamento e de perfis de usuários. A desconexão de clientes é tratada pelo armazenamento temporário de mensagens, no qual mensagens não entregues a um cliente são enfileiradas (até atingirem o tempo de expiração) para entrega após a reconexão. A gerência de localização mantém um mapeamento do usuário e o dispositivo que este utiliza em certo momento, permitindo assim

a mobilidade pessoal [3]. Os perfis de usuários são utilizados para filtrar o conteúdo de acordo com as preferências dos usuários. Além disso, de acordo com o dispositivo e condições da rede, é possível realizar algum tipo de adaptação de conteúdo. Apesar da adaptação de conteúdo fazer parte dos serviços propostos da arquitetura, ela aparece apenas como um item desejável, mas pouco se explica como esta seria configurada e realizada. Aparentemente, a adaptação deve ser realizada individualmente, e ser ativada apenas por contextos especificados no perfil do usuário.

Outro sistema *push* para clientes móveis foi proposto em [106]. Neste sistema, há três componentes: o servidor de conteúdo, clientes assinantes de conteúdo e agentes. Toda comunicação entre clientes móveis e servidor é intermediada por um agente (um *gateway*), que atende a uma determinada área de serviço, composta por alguns pontos de acessos (AP) aos quais os clientes estão conectados. O servidor provê um serviço *push* aos assinantes, e para isso mantém uma base de conteúdo, os registros dos clientes e suas preferências (*User Profiles*), bem como, o mapeamento entre clientes e os agentes que os atendem. Os clientes, especificam suas preferências (tempo de expiração, horários de entrega e tipos de dados preferidos) no momento do registro de interesse de conteúdo (assinatura) junto ao servidor. Os agentes são responsáveis por tratar as questões de conectividade e localização dos clientes. Os módulos funcionais de um agente compreendem: i) Gerente de localização, que pode enviar informações dependentes de localização (como notícias e promoções locais), e também informa os usuários co-localizados, ou seja, que estão na mesma área de serviço do agente; ii) Gerente de conexão, que detecta a conexão (ou desconexão) dos clientes através de mecanismo de *beacons* periódicos; iii) Gerente de cache inteligente, que adapta a freqüência de atualização do cache local de acordo com as necessidades de coerência temporal dos usuários e a dinâmica (taxa de mudança) dos dados no servidor; iv) Gerente de personalização, que de acordo com as preferências do usuário, filtra e organiza o conteúdo das notificações, e as envia no momento em que o cliente se conectar ao sistema.

A personalização do conteúdo é feita individualmente para cada cliente no agente, o que pode levar a uma sobrecarga deste elemento com o aumento do número de clientes, ou seja, a solução não é escalável. Além disso, essa personalização apenas atua de acordo com preferências (pré-definidas) do usuário, principalmente relacionada a filtragem de conteúdo a ser entregue, como tempo expiração, ou algum tipo de modificação simples do conteúdo,

[3] A mobilidade pessoal trata da mobilidade de usuários-finais que utilizam terminais diferentes em várias redes, e que permanecem unicamente endereçáveis no escopo de um serviço [105].

como entrega apenas do título, ou do conteúdo sem imagens. Não há uma personalização sensível a variações de contexto, com adaptação dinâmica de conteúdo, por exemplo. Outro aspecto é o tratamento de desconexão, realizada pelo agente, que, ao detectar a desconexão de um cliente, ativa um armazenamento temporário para entrega das informações em uma posterior reconexão do cliente (no mesmo agente).

3.3 Resumo

Neste capítulo, foram levantadas algumas questões que são tratadas na literatura por sistemas que fornecem suporte a aplicações móveis com adaptação de conteúdo sensível ao contexto, e a aplicações de disseminação de conteúdo, tais como, adaptação de conteúdo estática ou dinâmica, definição das regras de adaptação programável e configurável, composição de adaptadores, informações de contexto utilizadas, suporte à mobilidade e desconexão, comunicação requisição/resposta ou publish/subscribe, e tratamento escalável para muitos clientes.

A Tabela 3.1 apresenta um resumo comparativo das principais características dos projetos discutidos neste capítulo. Pode-se notar que nenhum dos sistemas analisados na literatura atendem a todos os requisitos conjuntamente.

	RAPIDware	Mobiware	MARCH	BARWAN	WBI	MobiPads	CAP	DCAF	MobilePush
Propósito	Multimídia	Multimídia, QoS	Geral	Geral	Aplicações Web	Aplicações Web	Geral	Aplicações Web	Disseminação de conteúdo
Decisão de Adaptação	Regras Programáveis	Programável	Regras configuráveis	Regras Programáveis	Regras configuráveis	Programável e Regras	Regras Programáveis	Regras Programáveis	?
Atualização de regras *runtime*	não	não	não	não	não	sim	não	não	não
Execução de adaptadores	Centralizada	Distribuída	Centralizada	Cluster	Centralizada	Centralizada	Centralizada	Distribuída	Centralizada
Suporte a mobilidade	não	Handoff horizontal	não	Handoff horizontal e vertical	não	Handoff Horizontal	não	não	Handoff e mobilidade pessoal
Suporte a desconexão	não	não	não	não	não	não	não	não	sim
Funcionalidades	Adaptação de conteúdo	Adaptação de conteúdo Handoff	Adaptação de conteúdo	Caching, Handoff, Adaptação de conteúdo	Caching, Adaptação de conteúdo	Adaptação de conteúdo	Adaptação de conteúdo	Adaptação de conteúdo	Disseminação de conteúdo, Handoff
Comunicação	Req/Resp	Req/Resp	Req/Resp	Req/Resp	Req/Resp	Req/Resp	Req/Resp	Req/Resp	Pub/Sub
Percepção de Contexto	enlace sem fio	enlace sem fio	dispositivo, enlace sem fio	enlace sem fio, perfil usuário e dispositivo	-	eventos: enlace, dispositivo	perfil dispositivo	perfil dispositivo, enlace sem fio	perfil usuário enlace sem fio

Tabela 3.1: Resumo comparativo dos projetos