

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Fabricia Lemos de Faria

SERVIDORES MÓVEIS EM REDES *AD HOC*

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Vitório Bruno Mazzola, Dr.

Florianópolis, fevereiro de 2005.

SERVIDORES MÓVEIS EM REDES *AD HOC*

Fabricia Lemos de Faria

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Raul Sidnei Wazlawick, Dr.
Coordenador do Curso

Banca Examinadora

Vitório Bruno Mazzola, Dr.
Orientador

Carlos Barros Montez, Dr.

Mário Antonio Ribeiro Dantas, Dr.

Antônio Augusto Medeiros Fröhlich, Dr.

Florianópolis, 16 de fevereiro de 2005

“... Basta ser sincero e desejar profundo (...) vai tente outra vez...”

(Raul Seixas)

DEDICATÓRIA

Aos meus amados pais,

Alcides e Santa

AGRADECIMENTOS

Aos meus pais, Alcides e Santa que sempre apoiaram minhas decisões inundando meus caminhos, minha vida com amor infinito, por tê-los sempre em mim, estando eles fisicamente longe ou perto.

A minha querida irmã Jamaina, meu cunhado Patrício, minha afilhada Bruna e finalmente ao meu irmão Darcy. A eles agradeço eternamente, por todo o amor que dedicam a mim, amor que completa a minha vida e me faz muito feliz todos os dias.

Ao Marcio Araújo e a Camila Toledo por serem parte da minha história, ontem, hoje e sempre.

Ao meu querido orientador, Vitório Bruno Mazzola por ter me concedido a oportunidade de demonstrar meu trabalho, dedicação e persistência, por me permitir realizar este grande desejo. Agradeço por sua importante contribuição no trabalho, por suas críticas, suas perguntas, pela análise “meticulosa” do texto, seu incomparável bom humor e principalmente por poder contar com sua experiência e sabedoria.

Ao professor Mário Antonio Ribeiro Dantas pelo incentivo, pelas várias dúvidas sanadas, pelo apoio em um dos momentos mais difíceis da pesquisa e pelas inúmeras conversas, sempre na busca por publicações.

Agradeço profundamente a minha grande amiga, colega de laboratório, Underléa Cabreira Corrêa por nossas conversas, pelo seu apoio, pela sua contribuição nesse trabalho através de questionamentos, sugestões, e principalmente por desfrutar da sua companhia e tudo de bom que eu ganhei com isso.

Aos colegas Carla e Gian Berkenbrock pela contribuição na implementação, e ao meu também colega de laboratório, Guilherme Bertoni Machado, todos eles sempre prontos a ajudar.

Aos meus velhos e novos amigos...

Enfim, são tantas as pessoas a quem agradecer...Obrigada!

SUMÁRIO

<u>LISTA DE ACRÔNIMOS.....</u>	<u>viii</u>
<u>LISTA DE FIGURAS.....</u>	<u>xi</u>
<u>LISTA DE QUADROS.....</u>	<u>xii</u>
<u>RESUMO</u>	<u>xiii</u>
<u>ABSTRACT.....</u>	<u>xv</u>
<u>1 INTRODUÇÃO</u>	<u>1</u>
1.1 OBJETIVO	2
1.2 METODOLOGIA	2
1.3 JUSTIFICATIVA	3
1.4 LIMITAÇÕES	3
1.5 ESTRUTURA DA DISSERTAÇÃO	4
<u>2 WEB SERVICES</u>	<u>5</u>
2.1 INTRODUÇÃO	5
2.2 PADRÕES RELACIONADOS	9
2.2.1 XML (<i>EXTENSIBLE MARKUP LANGUAGE</i>)	9
2.2.2 WSDL (<i>WEB SERVICES DESCRIPTION LANGUAGE</i>).....	13
2.2.3 UDDI (<i>UNIVERSAL DESCRIPTION, DISCOVERY AND INTEGRATION</i>).....	16
2.2.4 SOAP (<i>SIMPLE OBJECT ACCESS PROTOCOL</i>).....	18
2.3 SOBRE AS PLATAFORMAS PARA WEB SERVICES	19
2.4 CONCLUSÕES	22
<u>3 REDES LOCAIS SEM FIO (WLAN).....</u>	<u>23</u>
3.1 INTRODUÇÃO	23
3.2 DESAFIOS	23
3.3 PADRÃO IEEE 802.11	25
3.4 TOPOLOGIAS.....	27
3.4.1 INFRA-ESTRUTURADA	27
3.4.2 <i>AD HOC</i>	28
3.5 MODELOS DE COMUNICAÇÃO MÓVEL	34
3.5.1 MODELO CLIENTE / SERVIDOR.....	34
3.5.2 MODELO PEER-TO-PEER.....	36
3.5.3 MODELO AGENTES MÓVEIS.....	36
3.6 CONCLUSÕES	36
<u>4 Serviços Móveis: Principais Abordagens</u>	<u>38</u>

4.1	INTRODUÇÃO	38
4.2	PRINCIPAIS DESAFIOS	38
4.2.1	DESCOBERTA DE SERVIÇOS	40
4.2.2	CONTEXTO DO USUÁRIO	41
4.2.3	ACESSAR SERVIÇOS EM AMBIENTES MÓVEIS	41
4.2.4	COMPOSIÇÃO DE SERVIÇOS	44
4.2.5	SEGURANÇA.....	45
4.3	CONCLUSÃO.....	46
5	<u>UM MODELO PARA O PROVIMENTO DE WEB SERVICES EM REDES AD HOC.....</u>	47
5.1	INTRODUÇÃO	47
5.2	ARQUITETURA	48
5.2.1	<i>WIRELESS PORTAL NETWORK</i>	48
5.2.2	<i>WIRELESS EXTENDED INTERNET</i>	49
5.2.3	<i>WIRELESS AD HOC NETWORK</i>	50
5.3	CONECTIVIDADE	51
5.4	APLICABILIDADE.....	57
5.4.1	ESTUDO DE CASO	57
5.5	AVALIAÇÃO DA PROPOSTA	57
5.5.1	ELEMENTOS UTILIZADOS NA PARTE EXPERIMENTAL	58
5.5.2	DISPOSITIVOS MÓVEIS.....	58
5.5.3	A REDE FIXA	59
5.5.4	PROTOCOLOS E OUTROS ELEMENTOS	59
5.5.5	COMUNICAÇÃO.....	60
5.5.6	ESCOLHA DA PLATAFORMA E OPÇÕES DE IMPLEMENTAÇÃO.....	60
5.5.7	RESULTADOS DA AVALIAÇÃO PRELIMINAR.....	62
5.6	CONCLUSÃO.....	64
6	<u>CONCLUSÕES E TRABALHOS FUTUROS</u>	65
	<u>REFERÊNCIAS BIBLIOGRÁFICAS.....</u>	67

LISTA DE ACRÔNIMOS

A

API *Application Programming Interface*

B

BPEL4WS *Business Process Execution Language for Web Services*

BSS *Basic Services Set*

C

CSMA/CA *Carrier Sense Multiple Access/Collision Avoidance*

D

DMF *Dispositivo Móvel Fornecedor*

DOM *Document Object Model*

E

ESS *Extended Service Set*

F

FTP *File Transfer Protocol*

G

GPRS *General Packet Radio Service*

GSM *Global System for Mobile*

H

HTTP *Hypertext Transfer Protocol*

I

IAPP *Inter-Access Point Protocol*

IBSS	<i>Independent Basic Services Set</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISO	<i>International Organization for Standardization</i>
ISO/IEC	<i>International Organization for Standardization / International Electrotechnical Commission</i>

M

MAC	<i>Medium Access Control</i>
MANET	<i>Mobile Ad hoc Networking</i>
M-Services	Serviços Móveis

P

PDA	<i>Personal Digital Assistants</i>
-----	------------------------------------

Q

QoS	Qualidade de Serviço
-----	----------------------

R

RBAC	Controle de acesso baseado em papéis
RELAX NG	<i>Regular Language description for XML</i>
RPC	Chamada de Procedimento Remota

S

SAX	<i>Simple API for XML</i>
Schematron	<i>Schema for Object-oriented XML</i>
SGML	<i>Standard Generalized Markup Language</i>
SNMP	<i>Simple Network Management Protocol</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SOAP	<i>Simple Object Access Protocol</i>
SOA	Arquitetura Orientada a Serviço
SOX	<i>Schema for Object-oriented XML</i>

T

TARC	<i>Trusted Authentication Registration Center</i>
TCC	<i>Trusted credential center</i>
TREX	<i>Tree Regular Expressions for XML</i>
U	
URL	<i>Uniform Resource Locator</i>
UDDI	<i>Universal Description Discovery & Integration</i>
W	
W3C	<i>World Wide Web Consortium</i>
WSCl	<i>Web Service Choreography Interface</i>
WSCL	<i>Web Service Conversation Language</i>
WSDL	<i>Web Service Definition Language</i>
WSInspection	<i>Web Services Inspection Language</i>
WS-Coordination	<i>Web Service Coordination</i>
WS-Transaction	<i>Web Service Transaction</i>
X	
Xlink	<i>XML Linking Language</i>
XML	<i>Extensible Markup Language</i>
Xpath	<i>XML Path Language</i>
XPointer	<i>XML Pointer Language</i>
XSL	<i>eXtensible StyleSheet Language</i>
XSLT	<i>eXtensible StyleSheet Language Transformation</i>
XSL-FO	<i>XSL Formatting Objects</i>

LISTA DE FIGURAS

FIGURA 2.1 PILHA DA ARQUITETURA <i>WEB SERVICES</i>	6
FIGURA 2.2 ENTIDADES RELACIONADAS.....	7
FIGURA 3.1 REDE INFRA-ESTRUTURADA.	28
FIGURA 3.2 REDE <i>AD HOC</i>	28
FIGURA 3.3 COMUNICAÇÃO ÚNICO-SALTO.	29
FIGURA 4.1 CLIENTE <i>WEB SERVICE</i>	39
FIGURA 4.2 ARQUITETURA BASEADA EM MULTI-CANAIS.....	43
FIGURA 4.3 DISPOSITIVO <i>PROXY</i>	44
FIGURA 5.1 ARQUITETURA GERAL DO MODELO.....	51
FIGURA 5.2 MODELO PROPOSTO.	55
FIGURA 5.3 MODELO PROPOSTO DETALHADO.....	56
FIGURA 5.4 DISPOSITIVO MÓVEL ADOTADO NO AMBIENTE EXPERIMENTAL.....	59
FIGURA 5.5 PILHA NECESSÁRIA AO DESENVOLVIMENTO DE UM <i>WEB SERVICE</i>	61
FIGURA 5.6 AMBIENTE EXPERIMENTAL.....	62
FIGURA 5.7 TESTE REALIZADO ATRAVÉS DO SIMULADOR.....	63

LISTA DE QUADROS

QUADRO 2.1 DOCUMENTO WSDL.....	15
QUADRO 2.2 MÁQUINA VIRTUAL.....	21
QUADRO 3.1 MÉTRICAS PARA OS PROTOCOLOS DE ROTEAMENTO.....	31
QUADRO 3.2 RESUMO DE PROTOCOLOS – ABORDAGEM PRÓ-ATIVA.....	32
QUADRO 3.3 RESUMO DE PROTOCOLOS – ABORDAGEM REATIVA.....	33
QUADRO 3.4 RESUMO DE PROTOCOLOS – ABORDAGEM HÍBRIDA.....	33
QUADRO 5.1 ETAPAS DO FUNCIONAMENTO DO MODELO.....	54

RESUMO

A arquitetura *web services* permite que aplicações desenvolvidas em diferentes plataformas interajam com o intuito de trocar dados e realizar tarefas sobre a rede mesmo em ambientes heterogêneos. A arquitetura utiliza uma pilha de padrões entre os quais destacam-se três: WSDL, linguagem utilizada para descrever o serviço, SOAP, protocolo que encapsula as mensagens XML trocadas pela rede e UDDI, mecanismo que age como um repositório de *web services*, todos com base na linguagem XML.

As redes *Ad hoc* possuem como principal característica uma formação rápida e temporária. Por não necessitar de infra-estrutura fixa, essas redes são estabelecidas rapidamente em locais onde as redes fixas teriam dificuldade ou até mesmo a impossibilidade de serem instaladas. Nessas redes, os nodos comunicam-se diretamente sem a necessidade de infra-estrutura fixa, sendo eles próprios responsáveis pelo roteamento dos pacotes.

Este trabalho de pesquisa apresenta um modelo que visa possibilitar a execução de *web services* em redes *Ad hoc*. Tendo em vista a usual carência de serviços que os usuários dessas redes possuem, nossa contribuição permite a tais usuários dispor do acesso a alguns serviços, que geralmente estão disponíveis apenas na rede fixa, restringindo o acesso aos serviços móveis à área de cobertura alcançada por essas redes.

Tendo em vista que os dispositivos móveis tiveram nos últimos tempos suas capacidades computacionais aumentadas, possibilitando com isso, a execução de um número cada vez maior de serviços, nosso objetivo é tornar possível a qualquer dispositivo em uma rede móvel *Ad hoc* oferecer um conjunto de serviços a qualquer outro dispositivo na mesma rede, de modo que, o próprio dispositivo seja um servidor móvel de serviços aos seus pares na rede.

O funcionamento do modelo tem como principal característica o fato de disponibilizar serviços sem que o usuário dependa do auxílio de qualquer infra-estrutura fixa no momento da execução desses serviços. Faz-se necessário à interação com a rede fixa apenas para que o dispositivo móvel armazene os módulos necessários para dispor do serviço aos seus pares. Uma vez concluída essa etapa, o usuário móvel tem autonomia suficiente para coordenar sua função de fornecer os serviços em momentos e locais apropriados.

Nossos resultados experimentais indicam que o modelo possui claras vantagens ao ser utilizado em redes *Ad hoc*. Dentre elas, destacamos o fato de explorar as características desses ambientes com a execução do serviço de modo *Ad hoc*, levando de fato, serviços a qualquer lugar.

ABSTRACT

The web services architecture allows applications, developed in different platforms, to exchange data and execute tasks on heterogeneous environments over computer networks. This architecture uses standard software layers to ensure the interoperability among applications written in XML, using packages such as: a language to describe the service (WSDL); a protocol that encapsulates XML messages sent through the network (SOAP); and a mechanism that acts as a repository for the web services (UDDI).

Ad hoc networks have as their main feature a fast and temporary formation, because unlike conventional networks, this kind of network does not require a fixed infrastructure. Therefore, it is possible to have an Ad hoc network in places where it is difficult or impossible to establish a traditional computer network. In Ad hoc networks, nodes communicate directly without the need of a fixed infrastructure, and these nodes can also act as routers. Nowadays, one may verify the absence of service variety to Ad hoc configuration when compared to networks with infrastructure.

In this research work we present a model which aims at the execution of web services in Ad hoc networks. The purpose of this work is to provide a set of services for mobile devices in the same Ad hoc network.

In the last few years, mobile devices have increased their processing and storage capacity, what makes possible, now, to execute a relatively big set of services. For this reason, we assume as the main purpose of this work to make it possible for any mobile device connected to an Ad hoc network, to offer services to any other devices on the same network, turning it into a mobile server.

The model's operation has as main characteristic offering many services in a way the user doesn't need support from any fixed infrastructure at the moment these services are being executed. The interaction with the fixed network becomes necessary only for the mobile device to store the modules that will allow the services to be offered to the other devices. After this stage is finished, the mobile user has enough autonomy to coordinate its functions in order to supply the services at appropriate moments and places.

Our experimental results indicate the model has clear advantages when used in Ad hoc networks. Among them, we emphasize the fact of exploring these environments

characteristics by executing the service on Ad hoc mode, taking services, in fact, anywhere.

1 INTRODUÇÃO

A arquitetura *web services* vem sendo muito pesquisada nos últimos anos e várias propostas e padrões surgiram com o propósito de possibilitar a execução de serviços sem a intervenção do usuário (TSAI et al., 2003), (ZHU et al., 2003), (BENATALLAH et al., 2004), integrando aplicações visando a troca de informações em ambientes homogêneos ou heterogêneos.

Como diferencial perante outras arquiteturas como, por exemplo, CORBA (OMG, 1995), a arquitetura *web services* utiliza padrões *de facto*, permitindo que qualquer aplicação, independente da ferramenta utilizada para seu desenvolvimento, possa ser disponibilizada como um *web service* aos usuários do ambiente *world wide web*, de uma *intranet*, ou ainda, de uma rede local sem fio (WLAN). Os principais padrões relacionados à arquitetura são: WSDL (*Web Service Definition Language*), linguagem utilizada para descrever o serviço; SOAP (*Simple Object Access Protocol*), protocolo que encapsula as mensagens XML (*eXtensible Markup Language*) e UDDI (*Universal Description Discovery & Integration*), mecanismo que age como um repositório de *web services*.

Redes móveis *Ad hoc* são caracterizadas por sua formação temporária e dinâmica, onde serviços e dispositivos podem estar disponíveis transitoriamente. Nessas redes, a comunicação entre os nodos é realizada sem infra-estrutura fixa, sendo os próprios nodos responsáveis pelo roteamento dos pacotes.

A capacidade dos dispositivos que operam nessas redes, como *PDA's* (*Personal Digital Assistants*) e *notebooks* foi objeto de um avanço considerável nos últimos tempos, possibilitando com isso a execução de um número cada vez maior de serviços. Seguindo esse avanço, fabricantes de plataformas de desenvolvimento, aprimoram cada vez mais suas ferramentas com o intuito de suportar a criação de serviços para estes ambientes.

Tendo em vista a usual carência de serviços nessas redes, nossa contribuição busca tornar possível a qualquer dispositivo em uma rede móvel *Ad hoc* oferecer um conjunto de serviços a qualquer outro dispositivo na mesma rede.

Web services utilizados com a computação móvel resultarão em ganhos significativos para consumidores e fornecedores de aplicações, tornando factível aos usuários dessas redes o acesso a uma gama de aplicações que geralmente estão disponíveis somente aos usuários de redes infra-estruturadas, redes fixas, ou ainda, exigem que os usuários móveis acessem uma rede fixa no momento da utilização do serviço requerido, mantendo os serviços disponíveis apenas dentro da área de cobertura alcançada pela rede. Nossa principal contribuição está ligada ao fato de permitir o acesso ao serviço de modo *Ad hoc*, ou seja, o usuário acessa o serviço sem o auxílio da rede fixa, explorando assim, em sua totalidade, as características da rede *Ad hoc*.

Neste cenário, são identificados diversos desafios, tanto por parte dos dispositivos, como também, no que se refere ao ambiente em que eles operam, ou seja, a rede *Ad hoc*. Dentre os desafios, podemos citar as limitações de energia e largura de banda e a freqüente perda de conexão quando comparada às redes fixas. Embora não sejam estes os únicos problemas, nosso trabalho objetiva tratar prioritariamente os anteriormente mencionados.

1.1 Objetivo

O objetivo principal deste trabalho de pesquisa é a definição de um modelo que permita disponibilizar a arquitetura *web services* aos usuários das redes móveis *Ad hoc*, de modo que tais usuários disponham do acesso a serviços sem o auxílio da rede fixa no momento da utilização desses serviços.

Como objetivo específico, pretende-se demonstrar que a adoção do modelo proposto permite suprir os usuários desses ambientes com uma gama de serviços que atualmente estão disponíveis somente aos usuários de redes infra-estruturadas ou das redes fixas.

1.2 Metodologia

- Definir e apresentar o modelo genérico para o provimento de *web services* em redes *Ad hoc* de modo que o dispositivo móvel seja o fornecedor de serviços a seus pares na rede e que tais serviços possam ser

disponibilizados sem o auxílio da rede fixa no momento da execução desse serviço.

- Desenvolver um serviço com base no modelo proposto a fim de testar e validar tal modelo.

1.3 Justificativa

A arquitetura *web services* traz consigo a possibilidade de integração de aplicações de modo que elas troquem informações de maneira padronizada.

As redes *Ad hoc* são formadas dinamicamente, sem a necessidade de infraestrutura fixa, chegando a lugares de difícil ou impossível acesso às redes infra-estruturadas.

O assunto M-Services, ou serviços móveis são o que muitos autores caracterizam como serviços disponíveis a qualquer hora e em qualquer lugar. Há um grande interesse da comunidade científica em disponibilizar serviços aos usuários das redes móveis. Esses usuários possuem uma grande carência de serviços, uma vez que tais serviços encontram-se geralmente na rede fixa, ou dependem dela no momento de sua execução. O principal diferencial da nossa pesquisa está no fato de tornar os serviços disponíveis de modo *Ad hoc*, independente da rede fixa.

1.4 Limitações

Em nossa pesquisa não foram abordados os tópicos a seguir, os quais poderão ser explorados em pesquisas futuras.

- Os serviços a serem desenvolvidos à partir do modelo proposto estarão limitados a opção de *read-only* já que o modelo não trata questões referentes à consistência dos dados;

- Apesar de serem considerados importantes, os aspectos relativos à qualidade de serviço e à segurança não serão abordados explicitamente;
- Não foram realizadas medições referentes à qualidade do enlace e ao consumo de energia dos dispositivos durante a execução da aplicação desenvolvida com base no modelo proposto.

1.5 Estrutura da dissertação

Esta dissertação está dividida em 6 capítulos, descritos a seguir.

Após este primeiro capítulo, de Introdução, o capítulo 2 apresenta a arquitetura *web services* e os principais padrões associados.

O capítulo 3 apresenta os principais conceitos relacionados às redes WLANs, suas principais características e desafios, topologias e modelos de comunicação. Nesse capítulo as redes *Ad hoc* foram esboçadas de modo mais extenso, já que essas redes fazem parte do cenário da nossa pesquisa.

O capítulo 4 descreve os trabalhos que envolvem o tema *M-Services*, fornecendo-nos uma visão geral das pesquisas desenvolvidas na nossa área de estudo. Os trabalhos exploram temas como: acesso, armazenamento, descobrimento e entrega dos serviços, entre outros.

O capítulo 5 ilustra a definição do modelo proposto e demonstra o funcionamento do mesmo. Adicionalmente apresentamos o ambiente experimental utilizado para a validação do modelo.

Finalmente, o capítulo 6 apresenta as conclusões do trabalho e sugere alguns tópicos a serem explorados na continuidade desta pesquisa.

2 WEB SERVICES

Este capítulo aborda a arquitetura *web services*. Detalhamos os principais padrões relacionados com a arquitetura, ou seja, o protocolo SOAP, a linguagem WSDL e o repositório UDDI. Além disso, apresentamos a linguagem XML. Embora existam outros padrões e também novas propostas, escolhemos não detalhá-los por eles não estarem dentro do contexto da nossa pesquisa.

2.1 Introdução

Em janeiro de 2002 o *Web Services Activity* foi iniciado tendo como objetivo desenvolver um conjunto de tecnologias para explorar todo o potencial dos *web services*. A *Web Services Activity* é composta por três grupos de trabalho, um grupo de interesse e um grupo de coordenação, o qual coordena o trabalho da *Activity*.

O *Web Services Architecture Working Group* publicou em fevereiro de 2004 um documento que especifica a arquitetura *web services*, no qual define os componentes funcionais e o relacionamento entre esses componentes.

A arquitetura *web services* vem sendo muito pesquisada nos últimos anos e como outras propostas, surge com o intuito de possibilitar a execução de serviços sem a intervenção do usuário. Adicionalmente, torna realidade a troca de informações sobre a rede, até mesmo em ambientes heterogêneos (TSAI et al., 2003), (ZHU et al., 2003), (BENATALLAH et al., 2004). Como diferencial perante outras arquiteturas como, por exemplo, CORBA (OMG, 1995), a arquitetura *web services* utiliza padrões de *facto*, e permite que qualquer aplicação, independente da ferramenta utilizada para seu desenvolvimento, possa ser disponibilizada como um *web service* aos usuários do ambiente *world wide web*, de uma *intranet*, ou ainda, de uma WLAN.

“*Web services* podem ser implementados em várias linguagens de programação e sobre diferentes plataformas” (TSAI et al., 2003). Além disso, se forem adequados aos padrões já estabelecidos, serviços existentes, ou seja, aplicações anteriormente desenvolvidas podem ser disponibilizadas como *web services*.

Web services são formados por diversos padrões desenvolvidos pelo W3C (*World Wide Web Consortium*), ou ainda, por consórcios formados pela indústria de tecnologia.

A figura 2.1 ilustra a pilha da arquitetura *web services*, demonstrando de modo geral, como os padrões se relacionam. Os principais são: SOAP, UDDI e WSDL, todos com base na linguagem XML, e padrões *de facto* (PILIOURA et al., 2002), (BENATALLAH et al., 2003), (CHUNG et al., 2003).

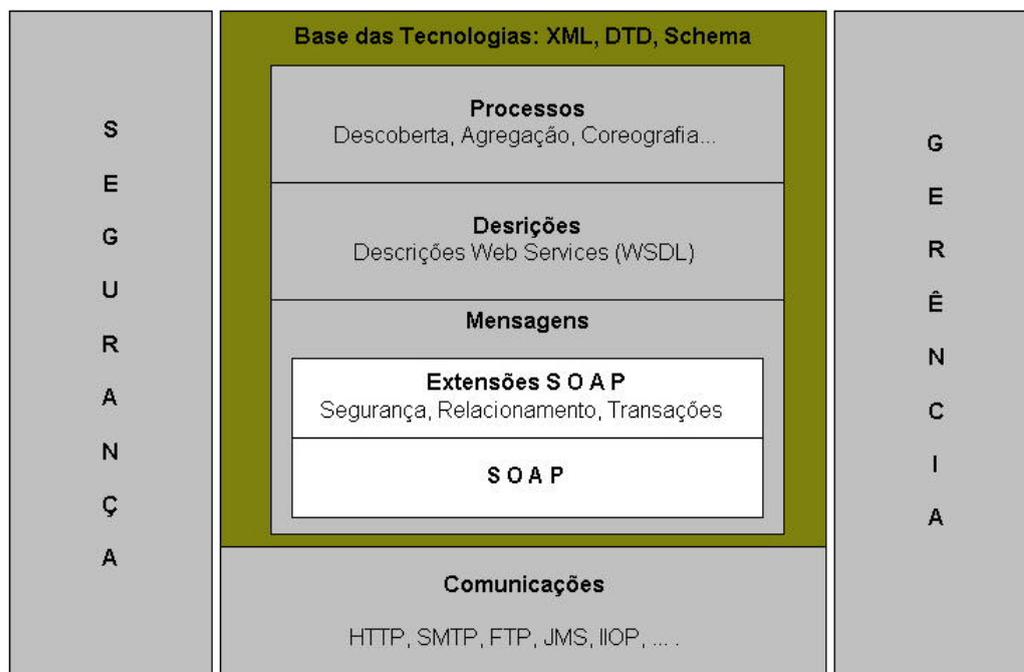


FIGURA 2.1 Pilha da Arquitetura *Web Services*.

Fonte: (W3C, 2004, A)

A linguagem XML é a chave principal da arquitetura *web services*, permitindo a troca de dados em ambientes heterogêneos. Mensagens XML são empacotadas pelo protocolo SOAP e podem ser transmitidas através de diversos protocolos de rede.

Segundo (W3C, 2004, A) *web services* são definidos:

“... como um sistema de software projetado para suportar interação interoperável máquina-a-máquina sobre a rede. Ele possui uma interface descrita em um formato processável por máquina (especificamente WSDL). Outros sistemas interagem com o *web service* da maneira fixada pela descrição usando mensagens SOAP, tipicamente transportadas usando HTTP com uma publicação serial XML em conjunto com outros padrões *web* relacionados...”.

A arquitetura *web services* utiliza o paradigma da arquitetura orientada a serviço (SOA). SOA é uma forma de arquitetura de sistemas distribuídos (W3C, 2004, A). Segundo os autores (CURBERA et al., 2003) “para operar em um ambiente de arquitetura orientada a serviço, aplicações devem declaradamente definir seus requisitos funcionais e não funcionais e suas capacidades, de forma legível para a máquina”.

A arquitetura orientada a serviço estabelece três principais papéis: o fornecedor, o cliente e o repositório de serviços (DELICATO et al., 2003). A figura 2.2 ilustra de forma resumida como se relacionam essas entidades.

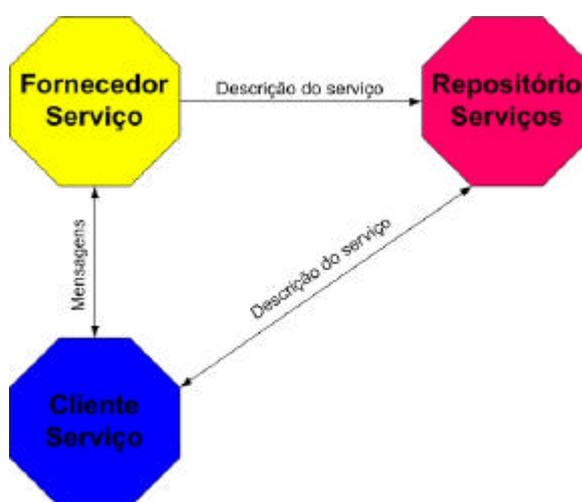


FIGURA 2.2 Entidades Relacionadas.

A partir do momento que o fornecedor do serviço deseja tornar público um *web service*, ele publica a interface desse serviço em um repositório central. A linguagem WSDL é utilizada para escrever essa interface e descrever o *web services* de forma padronizada (STEELE 2003).

Para a interação entre fornecedor e repositório de serviços a interface WSDL é encapsulada pelo protocolo SOAP. O SOAP aceita diversos protocolos para o transporte das mensagens, como HTTP, FTP, SMTP, entre outros. Geralmente essa transação é efetuada sobre HTTP.

O papel do cliente é procurar em um ou mais repositórios, serviços que satisfaçam suas necessidades. Uma vez encontrado, o cliente solicita a interface WSDL desse serviço e fará uso dela para descobrir como invocar o serviço. Exemplos de

informações encontradas nesse documento são: a URL onde serviço esta localizado, quais os parâmetros são necessários para a invocação de um determinado método, protocolo utilizado para o transporte das mensagens, etc.

Finalmente o papel do repositório é possibilitar que fornecedores publiquem informações sobre seus serviços de modo que estes possam ser descobertos e acessados pelos clientes. O repositório de serviços não contém o serviço em si; apenas a descrição dos serviços contendo informações de onde localizá-lo e como invocá-lo.

De forma genérica SOA e *web services* são mais apropriados às seguintes aplicações (W3C, 2004, A):

- Aplicações que irão rodar sobre a internet, onde confiança e velocidade não poderão ser garantidas;
- Onde componentes de sistemas distribuídos executam em diferentes plataformas e produtos de diferentes fabricantes;
- Onde uma aplicação existente necessita ser disponibilizada para uso pela rede como um *web service*.

Nas palavras dos autores (BENATALLAH et al., 2003) “uma aplicação desenvolvida usando um *middleware* baseado em componente pode ser exportada como um *web service* e pode ser composta com outro *web service* usando composição de *web services*”. A composição de serviços pode ser estática (em tempo de criação), ou dinâmica (em tempo de execução). O mesmo autor apresenta várias pesquisas na área de composição de serviços (BENATALLAH et al, 2004, 2003, 2002). Os autores (PIRES et al., 2003) igualmente apresentam um trabalho nessa área.

Segundo o mesmo autor (BENATALLAH et al., 2004), diversas tecnologias têm surgido para estender os atuais padrões citados anteriormente, com o objetivo de suportar a integração de aplicações. Algumas são *Business Process Execution Language* (BPEL4WS), *Web Services Conversation Language* (WSCL), *Web Service Choreography Interface* (WSCI), *WS-Coordination*, *WS-Transaction*, entre outras.

Além destas, outras tecnologias como *ontology*, *workflow* e *semantic web* também estão sendo pesquisadas (TSAI et al., 2003).

2.2 Padrões Relacionados

Inicialmente descrevemos a linguagem XML por ela ser a base dos três principais padrões relacionados com a arquitetura *web service*. Posteriormente descrevemos a linguagem WSDL, o repositório UDDI e finalmente o protocolo SOAP, todos esses, detalhados respectivamente a seguir.

2.2.1 XML (*eXtensible Markup Language*)

Em 1986, a SGML (*Standard Generalized Markup Language*) foi adotada pela ISO (*International Organization for Standardization*) como um padrão para a troca e armazenagem de dados (ISO 8879). É caracterizada por ser uma linguagem extremamente poderosa, mas também muito pesada e complexa. Devido a esta complexidade, seu uso no ambiente *world wide web* tornou-se inviável (ANDERSON et al., 2001).

Com o aumento da utilização da *world wide web* surgiu a necessidade de uma abordagem mais simples que a complexa SGML. Sendo assim, em 1996 o *XML Working Group* deu início ao projeto que resultaria, em 1998, na recomendação XML 1.0.

Segundo o (W3C, 2004, C) o projeto da especificação XML tinha os seguintes objetivos:

- A linguagem deveria ser usada sobre a internet;
- A linguagem deveria suportar uma grande variedade de aplicações;
- A linguagem deveria ser compatível com a SGML;
- Deveria ser fácil escrever programas que processassem documentos formatados na linguagem;
- O número de atributos opcionais deveria ser o mínimo;
- Pessoas deveriam ler de forma clara os documentos;
- O projeto deveria ser preparado rapidamente;

- O projeto deveria ser formal e conciso;
- Deveria ser fácil gerar os documentos;
- Resumir as marcações teria mínima importância.

A XML (*eXtensible Markup Language*) é hoje utilizada amplamente para a troca de dados entre aplicações sob plataformas homogêneas ou heterogêneas de maneira independente, porque permite que se descubra a estrutura dos documentos, tendo como uma das principais características positivas, ser de domínio público.

Os pontos chave de XML dentro da arquitetura *web services* são sua sintaxe, arquitetura, e os conceitos de XML *Infoset*, XML *Schema* e XML *Namespaces* (W3C, 2004, A).

Sintaxe XML

O conteúdo de um documento XML constitui-se de dados de caracteres e marcações (*markup*). As marcações compreendem *tags* iniciais, *tags* finais, *tags* de elemento vazio, referência à entidade, comentários, seções CDATA, delimitador, declaração de tipo de documento (DTD), instruções de processamento, declarações XML, declarações de texto e espaços em branco (W3C, 2004, C).

XML Infoset

Fornece um vocabulário para os parsers descreverem o conteúdo de um documento XML (ANDERSON et al., 2001). Um XML Infoset ou XML *Information Set* consiste de vários *information items*, cada um, contendo uma propriedade. Um *information item* é uma descrição abstrata de alguma parte de um documento XML (W3C, 2004, J).

XML Schema

XML *Schema* descreve a estrutura de um documento XML. É uma recomendação do W3C, dividida em duas partes: *structures* e *datatypes* (W3C, 2001, E). A primeira parte define a linguagem XML *Schema*, a qual oferece facilidades para descrever a estrutura e restringe o conteúdo de documentos XML. A segunda parte da recomendação define os mecanismos para validar o conteúdo dos elementos XML especificando os tipos de dados para cada elemento.

Além do padrão XML *Schema* existem outras propostas, como por exemplo: RELAX NG (*Regular Language description for XML*), TREX (*Tree Regular Expressions for XML*), SOX (*Schema for Object-oriented XML*), Schematron (*Schema for Object-oriented XML*).

XML Namespace

XML *Namespace* é uma coleção de nomes, identificados por um URI (RFC 2396). São usados em documentos XML como tipos de elementos e nomes de atributos (XML, 2004).

Arquitetura XML

Um documento XML possui uma estrutura física e uma estrutura lógica. O documento é composto fisicamente por unidades chamadas entidades. Logicamente, o documento é composto de declarações, elementos, comentários, referências de caractere e instruções de processamento, todos estes indicados com marcações (W3C, 2004, C).

Para se ler e acessar a estrutura e o conteúdo de um documento XML, é utilizado um parser ou processador. Existem dois tipos de parsers: sem validação e com validação. Parsers com validação utilizam um DTD (*Document Type Definition*) que poderá ser interno, externo ou ambos.

As APIs DOM (*Document Object Model*) e SAX (*Simple API for XML*) são exemplos de parsers utilizados para validar documentos XML bem formados. (W3C, 2004, D). Documentos XML serão bem formados se (ANDERSON et al., 2001):

- Obedecerem à sintaxe XML, podendo ser usados sem um DTD ou Schema;
- Formados por um único nó raiz, constituindo-se de uma árvore hierárquica;
- As únicas referências a entidades externas forem a um DTD fornecido juntamente com o documento.

A interface DOM tem como objetivo facilitar o trabalho dos programadores para acessar componentes, adicionar ou editar seu conteúdo, atributos e estilo. Ela é neutra no que se refere à linguagem e plataforma. O DOM é a API que mais oferece suporte para trabalhar com documentos XML, tratando-os na forma de uma árvore, na qual cada elemento é tratado como um nó. Ele define a estrutura lógica dos documentos e a forma através da qual um documento é acessado e manipulado (W3C, 2004, D). O DOM carrega o arquivo na memória, aumentando o consumo dessa de acordo com o tamanho do arquivo.

Diferentemente do DOM, o SAX não foi desenvolvido pelo W3C e sim por um grupo informal de participantes do XML-DEV. A lista XML-DEV foi criada em janeiro de 1997 pelo professor Peter Murray-Rust da Universidade de Nottingham (UK) e pelo Dr. Henry Rzepa do Colégio Imperial (UK) e foi gerenciada e operada por eles com a ajuda de generosos recursos doados pelo Colégio Imperial até fevereiro de 2000, quando a operação da lista foi transferida ao OASIS (XML org, 2004). O SAX é orientado a eventos e têm a desvantagem de não suportar manipulação de acesso randômico sobre o documento, sendo ele, um mecanismo de acesso serial. O SAX permite que se trabalhe com documentos XML bastante grandes, requerendo poucos recursos do sistema. Além disso, possui a característica de ser de domínio público (SAX, 2004).

A XSL (*eXtensible StyleSheet Language*) é a linguagem de estilos para XML que torna possível que um mesmo dado seja representado de diversas formas. Segundo (W3C, 1999, F), XSL consiste de três partes para a transformação e apresentação de documentos XML: XSLT (*eXtensible StyleSheet Language Transformation*), XPath (*XML Path Language*) e XSL-FO (*XSL Formating Objects*).

Outras recomendações do W3C, como, XLink, XML Base, XPointer utilizadas em conjunto, fornecem grande poder ao XML. O XLink (*XML Linking Language*), permite que elementos sejam inseridos dentro de documentos XML para criar e descrever links entre recursos (W3C, 2001, G). Os links poderão ser unidirecionais ou até mesmo links mais complexos.

Os links são utilizados para definir relacionamentos entre documentos, assim como, para definir uma seqüência no qual os documentos devem ser lidos ou até mesmo para inserir conteúdo não XML em um documento XML (Anderson et al., 2001).

O XPointer (*XML Pointer Language*) permite apontar para uma determinada área em um documento XML. Utiliza expressões XPath como identificadores URI permitindo referência de elementos em documentos externos (W3C, 2001, G).

Um dos aspectos mais importantes da linguagem é a separação dos dados com sua estrutura e apresentação. Além disso, a independência dos diversos padrões relacionados com XML como exemplo, XPointer, XLink, XBase (W3C, 2001, G), entre outros, em constante desenvolvimento, adicionam poder à linguagem por não estarem intrinsecamente ligados a ela.

2.2.2 WSDL (*Web Services Description Language*)

O W3C publicou em 21 de dezembro de 2004 a última modificação, até o presente momento, do rascunho da especificação que se refere a uma linguagem para a descrição de serviços, sob a denominação de *Web Services Description Language*.

A linguagem WSDL define uma gramática para descrever serviços na rede como coleções de pontos finais (portas) de comunicação, ajudando a automatizar a comunicação entre as aplicações (W3C, 2001, B).

“WSDL é uma linguagem baseada em XML usada para descrever *web services* de forma padronizada” (STEELE, 2003).

Nas palavras dos autores (PIRES et al., 2003) “através da WSDL, um cliente remoto entende como interagir com o *web service*”. Exemplos de informações obtidas com esse documento são métodos, parâmetros, *data types*, URL, o protocolo pelo qual o serviço poderá ser acessado, etc (FRIEDMAN, 2002). Estas informações serão usadas entre o *requester agent* e o *provider agent*. Além dessas, a especificação **WSDL**

especifica também uma ou mais localizações na rede nas quais um *provider agent* poderá ser invocado e fornece informações sobre o padrão de troca de mensagem esperado.

Nas palavras do autor (BENATALLAH et al., 2003) “WSDL suporta a definição de pontos de entrada e tipos de mensagens fornecidas pelo *web service*”.

Para definir um serviço, a linguagem WSDL utiliza os seguintes elementos (W3C, 2001, B):

- **Serviço (*service*)** – os pontos finais (*endpoints*) relacionados ao serviço;
- **Porta (*port*)** – define o ponto final. Especifica uma única ligação (*binding*);
- **Ligação (*binding*)** – *binding* refere-se ao processo de associar a informação de um protocolo ou formato de dados com uma entidade abstrata, (mensagem, operação, ou *portType*);
- **Tipo de porta (*port type*)** – um conjunto abstrato de operações suportadas por uma ou mais portas;
- **Tipos (*types*)** – um container para definição de tipo de dados usando algum sistema de tipo, por exemplo, XSD, também conhecido como XML Schema;
- **Mensagem (*message*)** – uma ou mais partes (*parts*) lógicas. As partes são mecanismos flexíveis para descrever o conteúdo lógico abstrato de uma mensagem;
- **Operação (*operation*)** – descrição abstrata de uma ação suportada pelo serviço.

A linguagem WSDL define quatro operações que um ponto final (*endpoint*) suporta (W3C, 2001, B):

- **Único-caminho (*One-way*)** - o ponto final recebe uma mensagem e nada será retornado;
- **Requisição-resposta (*Request-response*)** - O ponto final recebe uma requisição e retorna uma mensagem;
- **Solicitação-resposta (*Solicit-response*)** - ponto final envia e recebe uma mensagem;
- **Notificação (*Notification*)** - O ponto final envia uma mensagem.

Podemos observar através do quadro 2.1 o exemplo de um documento WSDL. Dentre as informações contidas nele, destacam-se: declarações de *namespace* (linhas 3, 6, 7, 8, 9), nome do serviço, no qual será único no documento (linha 37), protocolo a ser utilizado para o transporte das mensagens, tipo de dados dos parâmetros (linhas 14, 15), endereço físico onde o serviço poderá ser acessado (linha 39), entre outras que podem ser observadas na figura abaixo.

QUADRO 2.1 Documento WSDL.

01	<?xml version="1.0" encoding="UTF-8"?>
02	<wsdl:definitions targetNamespace="http://localhost:8080/axis/Servidor.jws"
03	xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-
04	soap" xmlns:impl="http://localhost:8080/axis/Servidor.jws"
05	xmlns:intf="http://localhost:8080/axis/Servidor.jws"
06	xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
07	xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
08	xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
09	xmlns:xsd="http://www.w3.org/2001/XMLSchema">
10	<wsdl:message name="calculaResponse">
11	<wsdl:part name="calculaReturn" type="xsd:int"/>
12	</wsdl:message>
13	<wsdl:message name="calculaRequest">
14	<wsdl:part name="A" type="xsd:int"/>
15	<wsdl:part name="B" type="xsd:int"/>
16	</wsdl:message>
17	<wsdl:portType name="Servidor">
18	<wsdl:operation name="calcula" parameterOrder="A B">
19	<wsdl:input message="impl:calculaRequest" name="calculaRequest"/>
20	<wsdl:output message="impl:calculaResponse" name="calculaResponse"/>
21	</wsdl:operation>
22	</wsdl:portType>
23	<wsdl:binding name="ServidorSoapBinding" type="impl:Servidor">
24	<wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

25	<wsdl:operation name="calcula">
26	<wsdlsoap:operation soapAction=""/>
27	<wsdl:input name="calculaRequest">
28	<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
29	namespace="http://DefaultNamespace" use="encoded"/>
30	</wsdl:input>
31	<wsdl:output name="calculaResponse">
32	<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
33	namespace="http://localhost:8080/axis/Servidor.jws" use="encoded"/>
34	</wsdl:output>
35	</wsdl:operation>
36	</wsdl:binding>
37	<wsdl:service name="ServidorService">
38	<wsdl:port binding="impl:ServidorSoapBinding" name="Servidor">
39	<wsdlsoap:address location="http://localhost:8080/axis/Servidor.jws"/>
40	</wsdl:port>
41	</wsdl:service>
42	</wsdl:definitions>

2.2.3 UDDI (*Universal Description, Discovery and Integration*)

O propósito geral do UDDI é a representação de dados e *meta dados* de modo a disponibilizar informações sobre um negócio ou entidade e suas interfaces técnicas. Ele oferece um mecanismo padrão para catalogar *web services*, de modo que eles possam ser descobertos e utilizados. Sua interface de programação é baseada em XML (OASIS, 2003).

A especificação UDDI define um repositório para *web services* e para outros serviços eletrônicos e não eletrônicos (OASIS, 2003). Estes serviços são registrados através de um documento WSDL (STEELE, 2003).

Fornecedores publicam seus serviços em repositórios públicos ou privados e consumidores utilizam esses repositórios para descobrir serviços que satisfaçam seus requisitos, obtendo meta dados necessários para o consumo desses serviços. Um registro privado somente permite acesso a usuários autorizados. Um registro público não restringe acesso ao serviço (OASIS, 2003). Para publicar um serviço no repositório é necessário disponibilizar a URL que contém a interface WSDL, fornecer informações sobre a empresa ou sobre a pessoa responsável pelo serviço e ainda, categorizar esse serviço para que ele possa ser encontrado de forma mais ágil. Vale ressaltar que esse serviço de publicação e descoberta do UDDI não possui custo nem para o fornecedor e nem para o cliente.

Para fornecer um mecanismo simples para a transmissão de mensagens, o UDDI utiliza o protocolo SOAP. Se nenhum outro protocolo for especificado, as mensagens XML empacotadas pelo protocolo SOAP serão transmitidas pelo protocolo HTTP. As seguintes entidades constituem um registro de serviço (OASIS, 2003):

- *businessEntity*: descreve um negócio ou organização que fornece o *web service*;
- *businessService*: descreve uma coleção de *web services* relacionados oferecidos por uma organização descrita pelo *businessEntity*;
- *bindingTemplate*: descreve informações técnicas necessárias para usar o *web service*. Além disso, fornece suporte para determinar um ponto de entrada técnico, suporte para serviços hospedados remotamente ou, ainda, a descrição de características únicas de um determinado serviço;
- *tModel*: possui dois principais objetivos. O primeiro é descrever um *web service* e utilizar a descrição para sua pesquisa. O segundo é tornar as descrições úteis para que se possa, a partir delas, aprender a interagir com o serviço;
- *publisherAssertion*: descreve, na visão de um *businessEntity*, o relacionamento que um *businessEntity* têm com outro *businessEntity*.

Soluções de diversos fabricantes implementam o UDDI (OASIS, 2003). Dentro da arquitetura *web services*, atualmente, ele é o padrão de repositório de serviços. Ainda assim, existem diversas alternativas ao padrão UDDI. Um exemplo é a tentativa conjunta da Microsoft e da IBM chamada *Web Services Inspection Language* (WSInspection).

Adicionalmente, o repositório de serviços UDDI é mais adequado para redes fixas. Nas redes móveis, são necessários mecanismos mais dinâmicos para lidar com a mobilidade dos usuários e todas as outras características inerentes ao ambiente móvel. A

seção trabalhos correlatos apresenta resultados de pesquisa sobre descoberta de serviços em redes dinâmicas.

2.2.4 SOAP (*Simple Object Access Protocol*)

A especificação SOAP ou XMLP (*XML Protocol*) é uma recomendação do W3C de 24 de junho de 2003, produzida pelo *XML Protocol Working Group*.

O grupo projetou quatro componentes, nos quais caracterizam o protocolo SOAP (W3C, 2003, I):

- Um envelope para encapsular os dados XML a serem transferidos;
- Uma convenção para o conteúdo do envelope, quando usado para aplicações de RPC;
- Um mecanismo para serialização de dados, tais como objetos gráficos;
- Um mecanismo para usar transporte HTTP. As plataformas de desenvolvimento não são obrigadas a suportar HTTP por ele não ser o único mecanismo de transporte suportado pelo SOAP.

Segundo o autor (BENATALLAH et al., 2003), o SOAP divide a mensagem em duas partes, cabeçalho e corpo. Enquanto o elemento cabeçalho é opcional, o corpo é um elemento obrigatório dentro do envelope SOAP. Se ambos forem utilizados, conterão múltiplos blocos de informações. A escolha dos dados que serão colocados no cabeçalho e quais vão para o corpo são decisões tomadas em tempo de aplicação.

O SOAP pode ser usado para implementar vários modelos para a troca de mensagens distribuídas, baseadas em XML (O'Tuathail & Rose, 2002). Mensagens SOAP transmitem informações na forma de um XML *Infoset* entre nodos SOAP, ou seja, um transmissor e um receptor SOAP (W3C, 2003, I).

Nas palavras dos autores (WOMBACHER & MAHLEKO, 2003) “o protocolo SOAP é um padrão que têm como base a linguagem XML e dentro da arquitetura *web*

services, é utilizado para a troca de mensagens”. Um documento XML se torna um documento SOAP depois de ser empacotado dentro de um envelope SOAP contendo cabeçalho e corpo, declarações de *Namespace*, ligação com um protocolo e estilos de codificação para serialização dos dados.

Mensagens SOAP podem ser transmitidas através de diversos protocolos. Adicionalmente, pode ser usada em uma variedade de padrões de troca de mensagens. Alguns dos protocolos aceitos pelo SOAP são HTTP, SMTP, TCP, FTP, Jabber, entre outros (W3C, 2003, I). O HTTP é o mais utilizado por passar pelos *firewalls*, mas não fornece suporte para comunicação assíncrona.

O protocolo SOAP é aceito como um padrão *web de facto*. Caracteriza-se por ser um protocolo de comunicação de peso leve capaz de efetuar a troca de informações em ambientes distribuídos, sendo independente de linguagem e plataforma. Seus maiores objetivos são simplicidade e expansibilidade.

A especificação SOAP fornece um mecanismo para o tratamento de erros. Uma falha é usada para transmitir uma informação de erro dentro de uma mensagem SOAP e será utilizada sempre que ocorrerem erros durante o processamento das mensagens SOAP (W3C, 2003, I).

As duas principais implementações do protocolo SOAP são o Apache SOAP da *Apache Foundation*, o qual é uma implementação Java, e o NuSOAP da *NuSphere Corporation*, que foi implementado em PHP.

2.3 Sobre as plataformas para *Web Services*

Atualmente, no que se refere à implementação de *web services* dois fabricantes lideram o mercado: *Microsoft* e *SUN Microsystems*, com suas plataformas .NET e Java, respectivamente. As seções a seguir apresentam as principais características de cada uma dessas plataformas.

2.3.1.1 Plataforma .NET

A plataforma .Net é uma solução que permite migrar códigos escritos em diversas linguagens de programação e transformá-los em código *microsoft*, ou seja, códigos ASP, VB, C++, etc (MICROSOFT, 2005).

A plataforma possui suporte completo para o desenvolvimento de *web services*. Para isso se faz necessário os seguintes elementos:

- *Microsoft Visual Studio .Net*: plataforma de desenvolvimento;
- *Windows Server System*: servidor *web*;
- Máquina Virtual: há duas versões da máquina virtual para a plataforma .Net intituladas de *Microsoft .Net Framework* e *Microsoft .Net Compact Framework*, esta última projetada para rodar em dispositivos móveis com capacidades computacionais reduzidas;
- *Web Services Enhancements (WSE)*: é um módulo de extensão ao *Microsoft Visual Studio* para o desenvolvimento de *web services*;
- *Microsoft SOAP toolkit*: uma implementação da especificação do protocolo SOAP.

Embora seu suporte ao desenvolvimento de *web services* seja satisfatório, incluindo cada vez mais facilidades, inclusive para o desenvolvimento voltado ao ambiente móvel, a plataforma .Net apresenta a desvantagem de ser proprietária e obrigar que as máquinas utilizadas para a execução dos *web services* rodem um sistema operacional *Microsoft* como, por exemplo, windows 2000, windows XP, windows CE, etc.

2.3.1.2 Plataforma Java

A plataforma Java utilizada no suporte ao desenvolvimento de *web services* invocados através da rede fixa é a Java 2 Platform, Enterprise Edition (J2EE).

Para o desenvolvimento de serviços que irão operar em dispositivos com capacidades computacionais minimizadas a SUN desenvolveu a Java 2 Platform, Micro

Edition (J2ME). A especificação J2ME fornece dois pacotes opcionais baseados em XML que podem ser utilizados no desenvolvimento de *web services* para dispositivos como PDAs, telefones celulares, etc, são eles (SUN, 2005):

- Java API for XML Processing (JAXP): o propósito desse pacote é habilitar as aplicações a parsear e transformar documentos XML. O JAXP é uma versão reduzida do JAXP da plataforma J2SE (Java 2 Platform, Standard Edition);
- Java API for XML-based RPC (JAX-RPC): essa API é uma implementação da tecnologia RPC para a linguagem Java. É um subconjunto do JAX-RPC da plataforma J2EE. Além disso, permite que seja utilizada a funcionalidade de RPC baseada em XML utilizando o protocolo SOAP.

A máquina virtual a ser instalada nos equipamentos para executar o serviço dependerá da plataforma escolhida para o desenvolvimento desse. O quadro 2.2 ilustra as duas opções existentes até o presente momento:

QUADRO 2.2 Máquina Virtual

Plataforma	Máquina Virtual
J2EE	<i>Java Virtual Machine (JVM)</i>
J2ME	<i>Kilobyte Virtual Machine (KVM)</i>

Adicionalmente, independente da plataforma Java utilizada o servidor *web* poderá ser o *Java Web*, o *Apache HTTP Server*, dentre outros.

Uma das principais vantagens envolvidas com a plataforma Java é que os serviços desenvolvidos a partir dela tornam-se operáveis em qualquer dispositivo, independentemente da arquitetura ou ainda, do sistema operacional desse dispositivo. Além disso, existem implementações disponíveis para Java da maior parte dos padrões envolvidos com a arquitetura *web service*, incluindo HTTP, SOAP, XML, e as versões reduzidas kHTTP, kSOAP, kXML, dentre outros.

Há um conjunto de IDEs que podem ser utilizadas por um desenvolvedor Java, NetBeans da SUN, a IDE Eclipse da Eclipse Foundation, o Gel da GExperts Inc., entre outras. Utilizamos no desenvolvimento do nosso serviço a IDE Eclipse por ela ter uma

variedade de funcionalidades que ajudam o desenvolvimento. Além disso, o Eclipse foi desenvolvido em Java e há diversos plug-ins que são adicionados conforme a necessidade da aplicação que esta sendo desenvolvida.

Com o intuito de tornar factível o projeto do J2ME, o conjunto de bibliotecas da plataforma que foi herdado do J2SE é bastante reduzido, diversas bibliotecas não estão disponíveis para o desenvolvimento de serviços, restringindo assim, o número de aplicações que poderiam ser implementadas e tornando isso, uma desvantagem perante outras plataformas.

2.4 Conclusões

Este capítulo abordou a arquitetura *web services* e os três principais padrões relacionados, WSDL, UDDI e SOAP. Em adição, apresentamos a linguagem XML por ela ser extensivamente utilizada pelos padrões que compõe essa arquitetura. Finalmente esboçamos as principais ferramentas para o desenvolvimento de *web services* existentes no mercado, comentando brevemente suas principais características.

A utilização de *web services* torna factível a troca de informações mesmo em ambientes heterogêneos. Sua adoção ganha adeptos em diversas áreas além de ser alvo de intensas pesquisas para disponibilizar soluções mais robustas e voltadas aos mais diversos ambientes.

3 REDES LOCAIS SEM FIO (WLAN)

3.1 Introdução

Nos últimos anos, tivemos uma evolução das tecnologias de comunicação móvel. As redes móveis inicialmente eram utilizadas em campos de batalha, mas, atualmente, ganham adeptos em todas as áreas: hospitais, clínicas, universidades, comércios em geral, edifícios, etc.

Além dos avanços nas tecnologias de comunicação, o número de usuários de dispositivos móveis tem crescido constantemente. A mobilidade concedida a tais usuários torna factível o acesso a serviços fora de seus escritórios ou residências, concedendo as habilidades de acessar e-mails e internet, transferir arquivos, utilizar serviços diversos, entre outras funções que cada dia tornam-se disponíveis.

Caminhamos ao encontro do que alguns pesquisadores chamam de “serviços disponíveis a qualquer hora e em qualquer lugar”, embora ainda existam sérias limitações para que isso seja uma realidade.

A comunicação nas WLANs se dá através de duas modalidades de propagação de sinal, rádio-frequência e luz infravermelha. A luz infravermelha tornou-se menos utilizada por possuir alguns aspectos que dificultam muito sua comunicação. Um desses, é o fato da luz não ultrapassar corpos opacos.

Neste capítulo explanaremos as redes locais sem fio, suas principais vantagens, desafios, topologias e principais problemas encontrados com a tecnologia de comunicação e com os dispositivos que operam nesses ambientes. Serão descritos também os modelos de comunicação que atendem ao ambiente móvel. Além disso, as redes *Ad hoc* serão apresentadas de forma mais ampla por serem o cenário da nossa pesquisa.

3.2 Desafios

Os usuários de ambientes *wireless* esperam alcançar, neste ambiente, as mesmas características das redes fixas e se deparam com uma série de restrições e desafios.

Alguns dos desafios da comunicação móvel *wireless* são (FORMAN & ZAHORJAN, 1994):

- Alta variação da largura de banda disponível;
- Altas taxas de erro;
- Custo da transmissão;
- Indisponibilidade temporária;
- Perda freqüente de conexão;
- Redes heterogêneas;
- Segurança.

Quando comparada com a das redes fixas, a largura de banda das redes móveis ainda é vista como uma desvantagem, embora a variação do padrão 802.11g tenha minimizado essa disparidade, tornando factível executar uma maior gama de aplicações sobre esses ambientes.

Dentre outras adaptações, os protocolos da arquitetura TCP/IP devem ser adaptados às características desses ambientes de forma que possam pelo menos amenizar alguns dos desafios citados anteriormente.

Nos últimos tempos testemunhamos um avanço considerável dos equipamentos que operam nesses ambientes. PDAs (*Personal Digital Assistants*), *laptops*, *paggers*, freqüentemente chamados de *mobile hosts*. Eles tiveram suas capacidades computacionais aumentadas. Tal característica possibilita que a carga dos serviços seja parcialmente ou em alguns casos totalmente removida dos servidores, até então, sempre localizados na rede fixa. Ainda assim, tais dispositivos possuem diversos desafios e desvantagens (PILIOURA et al., 2003), (MOHAPATRA et al., 2004), (STEELE, 2003), tais como:

- Custo;
- Display limitado;
- Entrada de dados restrita;
- Desempenho;
- Baixa capacidade de processamento (CPU) e memória;
- Carga finita de bateria;
- Heterogeneidade.

Em (SATYANARAYANAN, 1996) são apresentadas quatro restrições da computação móvel: os elementos são pobres de recursos, o nível de segurança é baixo, o desempenho é variável, e a energia é finita. Com isso, concluímos que, mesmo depois de anos de pesquisa e avanços significativos em hardware e software os mesmos desafios ainda persistem e várias adaptações surgem a todo o momento, visando superá-los.

3.3 Padrão IEEE 802.11

O grupo de trabalho IEEE 802.11 tem como objetivo desenvolver o padrão internacional para as redes WLANs. O padrão IEEE 802.11 define a camada física e a camada MAC (*Medium Access Control*) para redes locais *wireless* (WLAN) e opera na frequência de 2.4 GHz.

O protocolo MAC adotado é o CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*) que tem como característica escutar o meio *wireless* antes de iniciar uma transmissão com o intuito de evitar colisões. A versão 802.11e MAC

propõe suportar QoS e diferenciação de serviços, característica até então não suportada pelo 802.11 (FERRE et al., 2004).

Várias especificações do padrão 802.11 foram desenvolvidas pelo grupo visando a melhoria do padrão e buscando contemplar as deficiências encontradas. As duas mais utilizadas pelos equipamentos atualmente são 802.11b e 802.11g. (IEEE 802.11, 2004):

- **802.11 (1997)** – primeira especificação para redes WLAN, operando em uma ou mais faixas de frequência, com taxas de transmissão de até 2Mbps;
- **802.11a (1999)** – extensão do padrão, maior velocidade, operando na faixa de 5GHz com taxas de transmissão de até 54Mbps. Possui como características não ser indicado para dispositivos móveis e possuir um alto consumo de energia;
- **802.11b (1999)** – Opera na faixa de 2.4GHz, com taxas de transmissão até 11 Mbps. As melhorias dessa especificação concentram-se na camada física;
- **802.11c** – fornecido com a documentação do 802.11- especifica os processos da camada MAC para o padrão 10038 ISO/IEC (IEEE 802.1D);
- **802.11d (2001)** – publicação das definições e requisitos para permitir operar em domínios controlados e vaguear entre esses domínios;
- **802.11e** – teve como meta adicionar QoS à camada MAC 802.11, visando permitir aplicações como voz, áudio e vídeo sobre redes *wireless*;
- **802.11f (2003)** – recomendação de práticas para a interoperabilidade de pontos de acesso de diferentes fabricantes usando um *Inter-Access Point Protocol* (IAPP) em um sistema distribuído, suportando links WLAN;
- **802.11g (2003)** – extensão para a camada física (PHY) do padrão 802.11b. Preserva a compatibilidade com os atuais dispositivos 802.11b, ou seja, a rede

poderá operar com equipamentos que suportam 11 Mbps assim como 54 Mbps;

- **802.11h (2003)** – estende as camadas MAC e física para fornecer gerenciamento do poder de transmissão e área de alcance na faixa de 5 GHz, isso deverá regulamentar o padrão em alguns países europeus;
- **802.11i** – Aumentar a segurança e os mecanismos de autenticação do padrão 802.11.

3.4 Topologias

Segundo o grupo de pesquisa 802.11 do IEEE, uma rede móvel *wireless* é classificada sob uma das seguintes topologias de comunicação: *Ah hoc (Independent Basic Service Set - IBSS)* ou infra-estruturada (*Extended Service Set - ESS*).

3.4.1 Infra-estruturada

A topologia ESS alcança uma área maior por interligar redes BSS expandindo a rede. Uma rede *Basic Service Set (BSS)* é formada por um conjunto de estações e pelo menos um ponto de acesso, no qual as estações estarão associadas. Eles são controlados por uma única função de coordenação no qual determina quando uma estação dentro da BSS pode transmitir e receber dados no canal *wireless* (IEEE 802.11, 2004).

No modo infra-estruturado um ponto de acesso age como ponte ligando a rede móvel *wireless* à rede fixa (IEEE 802.3). Adicionalmente, os dispositivos não se comunicam diretamente e sim através do ponto de acesso que encaminha os pacotes ao destino utilizando o endereço MAC dos dispositivos. A figura 3.1 apresenta os componentes que formam uma BSS.

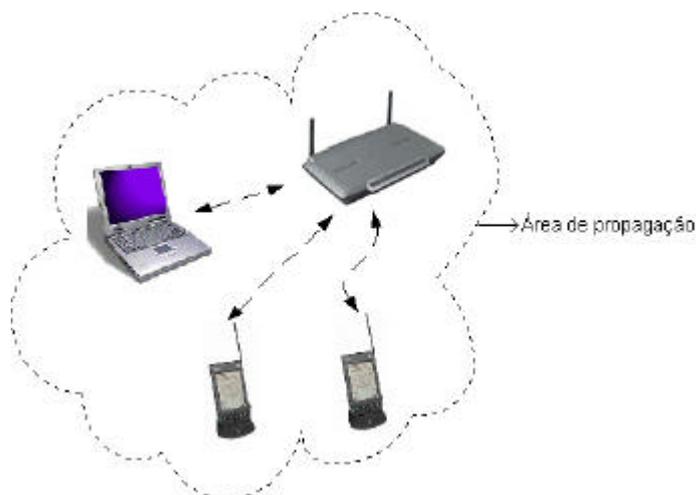


FIGURA 3.1 Rede Infra-Estruturada.

3.4.2 *Ad hoc*

Redes *Ad hoc* ou MANET (*Mobile Ad hoc Networking*) são redes *wireless* formadas por nodos móveis que se comunicam sem a necessidade de infra-estrutura fixa ou administração centralizada (MOHAPATRA et al., 2004). Devido a essa característica, essas redes são rapidamente estabelecidas e altamente flexíveis, podendo chegar a lugares difíceis ou até mesmo inacessíveis às redes infra-estruturadas e às redes fixas (GAERTNER & CAHILL, 2004). Podemos verificar através da figura 3.2 uma rede *Ah hoc* formada por PDAs e *notebooks*, onde a comunicação é peer-to-peer e ocorre sem o auxílio de qualquer entidade centralizadora.

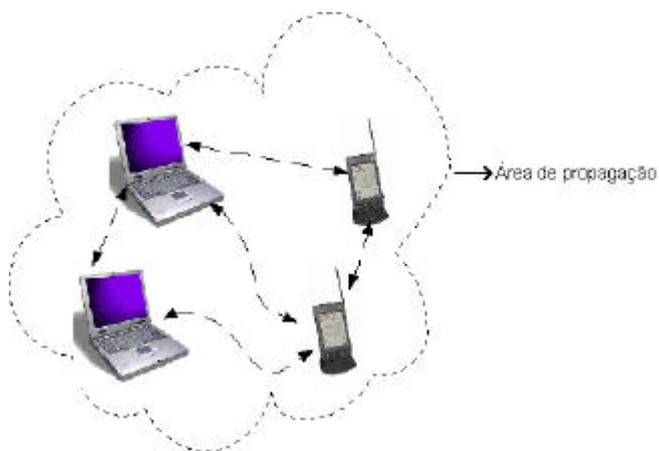


FIGURA 3.2 Rede *Ad hoc*.

A rápida formação é uma das principais características dessas redes, tornando factível aos usuários, estabelecerem uma rede local altamente dinâmica. Outra importante característica é que se um determinado nodo falhar, a rede seguirá operando normalmente.

Devido à constante movimentação dos nodos em uma rede móvel *Ad hoc*, a topologia é dinâmica, podendo consistir de links bidirecionais e unidirecionais (CORSON & MACKER, 1999).

Se desejado, a rede *Ad hoc* pode interagir com a rede fixa através de um ponto de acesso (STEELE, 2003), que agirá como ponte ligando a rede *wireless* a rede fixa.

Devido à falta de infra-estrutura fixa, os nodos móveis, agem como roteadores. Estando dois nodos dentro da mesma área de transmissão, eles se comunicam diretamente, sem a necessidade de nodos roteadores. Essa forma de comunicação é chamada de único-salto. Quando um nodo deseja transmitir pacotes de dados para outro nodo que não esteja dentro da sua área de transmissão, nodos intermediários agem como roteadores, fazendo o encaminhamento desses pacotes (CAO et al., 2004), (GAERTNER & CAHILL, 2004), (CORSON & MACKER 1999). Nesse caso, a forma de comunicação é conhecida como múltiplos-saltos.

Podemos observar através da figura 3.3 uma rede *Ad hoc* formada por dispositivos heterogêneos. A comunicação ocorre diretamente entre um PDA (Nodo A) e um *notebook* (Nodo B).



FIGURA 3.3 Comunicação único-salto.

As aplicações destinadas a estas redes e os próprios dispositivos que nela operam devem ser concebidos tendo como preocupação preponderante o consumo de energia, visto que, estes dispositivos são compostos de baterias com cargas finitas de energia.

A comunicação nesta classe de redes está sempre sujeita às interferências do ambiente, por exemplo, carros, prédios, o próprio corpo do usuário, entre outras.

Segundo (CORSON & MACKER, 1999) dentre as características das redes *Ad hoc* destaca-se:

- Topologia dinâmica: os nodos movimentam-se aleatoriamente, fazendo com que estas redes necessitem de poderosos mecanismos de roteamento que descubram a localização real de um determinado nodo;
- Limitada largura de banda: com a tecnologia disponível atualmente (padrão 802.11), a largura de banda destas redes alcança no máximo 54 Mbps;
- Segurança: os dados podem ser propagados para fora dos limites da rede, ficando mais vulneráveis à escuta e invasões.

As características acima mencionadas, quando comparadas às das redes fixas apresentam-se como pontos de desvantagem.

Em seu recente trabalho (GAERTNER & CAHILL, 2004), afirmam que “a qualidade nas atuais redes *Ad hoc* 802.11 ainda é baixa e os usuários podem experimentar fortes flutuações na qualidade do link”.

Em se tratando de QoS, algumas garantias como tempo limite para a entrega de pacotes, disponibilizar largura de banda e garantia probabilística de perda de pacotes, ainda não são possíveis. A reserva de recurso, por enquanto, é a escolha para tais garantias (GAERTNER & CAHILL, 2004).

Adicionalmente, essas redes ainda não possuem um mecanismo eficiente de endereçamento. Diversas abordagens têm surgido com o objetivo de suprir tal necessidade, como por exemplo, arquiteturas livres de endereço (ELSON & ESTRIN 2000), IP móvel, protocolos de auto-configuração de endereços IP, entre outros.

3.4.2.1 Roteamento

Por não ter uma entidade de coordenação, a mobilidade dos nodos em uma rede *Ad hoc* torna a descoberta da localização e a escolha da melhor rota para a entrega de um pacote do nodo origem para o nodo destino uma tarefa árdua. Essa característica faz com que se necessite de mecanismos de roteamento poderosos.

Dentre as áreas de pesquisa dentro das redes *Ad hoc*, o roteamento é a área que mais tem sido pesquisada (CAO et al., 2004). Até aqui diversos algoritmos de roteamento foram propostos, mas ainda não se dispõe de um mecanismo que seja eficiente em todos os momentos da vida dessas redes, ou seja, os algoritmos que se mostram eficientes quando a rede é pequena, perdem essa característica à medida que a dimensão da rede aumenta.

O grupo MANET (*Mobile Ad hoc Networking*) estabeleceu algumas métricas que os protocolos de roteamento projetados para esses ambientes devem satisfazer (CORSON & MACKER, 1999):

QUADRO 3.1 Métricas para os protocolos de roteamento.

Métrica	Descrição
Atraso e throughput de dados fim a fim	Medidas estatísticas como média, variância, distribuição.
Tempo para a descoberta de rota	O tempo necessário para estabelecer as rotas. Nos algoritmos de roteamento sob demanda, essa propriedade mede o atraso do pacote fim a fim.
Ordem de entrega dos pacotes	A porcentagem dos pacotes entregue fora de ordem é uma medida externa para avaliar a performance dos protocolos de roteamento.
Eficiência	Tráfego de dados e de controle maior do que o suportado pelo canal de comunicação, degrada a eficiência do roteamento. Esta medida será obtida através dos números médios de bits de dados transmitidos / bits de dados entregues, bits de

	controle transmitidos / bits de dados entregues, pacotes de controle transmitidos / pacotes de dados entregues.
--	--

Tendo em vista o grande número de protocolos propostos, eles foram classificados quanto ao processo de descoberta de rotas em três abordagens: abordagem reativa, abordagem pró-ativa e abordagem híbrida (ABOLHASAN et al., 2003).

- Abordagem Pró-ativa: protocolos que se enquadram nesta abordagem mantêm uma tabela sobre a posição dos nodos, assim, quando for feita uma requisição, o endereço do destinatário ou do próximo salto já é conhecido. Atualizações sobre a posição dos nodos são realizadas periodicamente pelo envio de mensagens, o que acarreta um *overhead* na rede. O quadro 3.2 apresenta alguns protocolos pró-ativos.

QUADRO 3.2 Resumo de Protocolos – Abordagem Pró-Ativa.

Protocolos – Abordagem Pró-Ativa
<i>Destination-sequence distance vector (DSDV)</i>
<i>Wireless routing protocol (WRP)</i>
<i>Global state routing (GSR)</i>
<i>Fisheye state routing (FSR)</i>
<i>Source-tree adaptive routing (STAR)</i>
<i>Distance routing effect algorithm for mobility (DREAM)</i>
<i>Multimedia support in mobile wireless networks (MMWN)</i>
<i>Cluster-head gateway switch routing (CGSR)</i>
<i>Hierarchical state routing (HSR)</i>
<i>Optimised link state routing (OLSR)</i>
<i>Topology broadcast reverse path forwarding (TBRPF)</i>

- Abordagem Reativa: protocolos que utilizam essa abordagem descobrem a rota somente após uma requisição para a entrega de um determinado pacote. Estes algoritmos apresentam um maior atraso quando comparados aos reativos por sua característica de descoberta de rotas sob demanda. Vários protocolos reativos

surgiram para suprir as deficiências dos protocolos pró-ativos. Alguns deles são mostrados na tabela 3.3.

QUADRO 3.3 Resumo de Protocolos – Abordagem Reativa.

Protocolos – Abordagem Reativa
<i>Ad hoc on-demand distance vector (AODV)</i>
<i>Dynamic source routing (DSR)</i>
<i>Routing on-demand acyclic multi-path (ROAM)</i>
<i>Light-weight móbile routing (LMR)</i>
<i>Temporally ordered routing algorithm (TORA)</i>
<i>Associativity-based routing (ABR)</i>
<i>Signal stability adaptive (SSA)</i>
<i>Relative distance micro-discovery ad hoc routing (RDMAR)</i>
<i>Location-aided routing (LAR)</i>
<i>Ant-colony-based routing algorithm (ARA)</i>
<i>Flow oriented routing protocol (FORP)</i>
<i>Cluster-based routing protocol (CBRP)</i>

- Abordagem Híbrida: protocolos híbridos unem as características das duas abordagens acima mencionadas, utilizando as características pró-ativas e reativas em um único protocolo. Estes protocolos geralmente dividem a rede em zonas, ou agrupam os nodos dentro de árvores, ou clusters. O quadro 3.4 apresenta cinco protocolos híbridos.

QUADRO 3.4 Resumo de Protocolos – Abordagem Híbrida.

Protocolos – Abordagem Híbrida
<i>Zone routing protocol (ZRP)</i>
<i>Zone-based hierarchical link state (ZHLS)</i>
<i>Scalable location update routing protocol (SLURP)</i>
<i>Distributed spanning trees based routing protocol (DST)</i>
<i>Distributed dynamic routing (DDR)</i>

Tratando-se dos métodos de comunicação, o modo de envio de pacotes *broadcast* é bastante utilizado nas redes *Ad hoc*. Ele tem sido pesquisado no intuito de minimizar alguns problemas como colisão, contenção e *re-broadcast* redundante, que aparecem

quando utilizamos comunicação *broadcast* (ZHU et al., 2004), (MOHAPATRA et al., 2004).

“Quando um novo nodo entra em uma MANET, ele não tem conhecimento sobre os outros nodos e serviços dentro da rede. Para esta descoberta, geralmente é utilizado o modo *broadcast*” (ZHU et al., 2004).

Nas palavras do autor (ZHU et al., 2004), “*broadcasting* é um componente chave para muitos protocolos de roteamento”. Muitas vezes, um mesmo protocolo implementa uma ou mais técnicas de comunicação. O trabalho de pesquisa de (MOHAPATRA et al., 2004) apresenta as vantagens e desvantagens de diversas técnicas como *multicasting*, *broadcasting*, *geocasting* e *anycasting*.

3.5 Modelos de Comunicação Móvel

Existem diversos modelos de comunicação usados pelas aplicações em sistemas distribuídos. Dentre eles, destacam-se três principais modelos: o tradicional modelo cliente / servidor, modelo *peer-to-peer* e modelo agentes móveis, nos quais, demonstramos a seguir (MATEUS & LOUREIRO, 1998):

3.5.1 Modelo Cliente / Servidor

Em uma rede móvel *wireless*, o usuário portando um dispositivo móvel representa o cliente. Este cliente requisita informações de um ou mais servidores que normalmente estão na rede fixa. Em alguns casos, o mecanismo de replicação de servidor é utilizado para aumentar a disponibilidade do serviço.

Dois abordagens são utilizadas para adequar o modelo cliente/servidor à comunicação móvel com o objetivo de lidar com as restrições do ambiente móvel *wireless*: modelo cliente móvel/agente/servidor e modelo cliente móvel/interceptor/servidor; elas são descritas nos próximos itens.

3.5.1.1 Modelo Cliente Móvel / Agente / Servidor

Neste modelo, o agente fica na rede fixa e é o representante do dispositivo móvel. O servidor, no qual recebe as requisições dos clientes permanece na rede fixa e tem sua carga de trabalho minimizada, pois o agente realiza algumas tarefas no lugar desse servidor como, por exemplo, compressão de dados.

O agente pode ser projetado para tratar de clientes ou de serviços. Se for um agente de serviço, este poderá atender mais de um cliente simultaneamente oferecendo um serviço a um grupo de clientes.

Nesse modelo, o roteamento dos pacotes sofre alteração. No caso do agente representar um serviço, apenas o tráfego referente a esse serviço deverá passar pelo agente. Em contrapartida, se o agente representar um cliente, todo o tráfego em direção a esse cliente deverá passar pelo seu agente.

Este modelo é apropriado para dispositivos com capacidades computacionais limitadas e redes que possuem pouca demanda computacional. Mesmo sendo uma evolução do modelo cliente / servidor, esse modelo apresenta algumas desvantagens como, por exemplo, o fato de obrigar as aplicações clientes a interagirem não mais com o servidor e sim com o agente.

3.5.1.2 Modelo Cliente Móvel / Interceptor / Servidor

O modelo cliente móvel / interceptor / servidor, apresenta-se como uma solução aos problemas apresentados no modelo anterior. Neste modelo, o agente é dividido em duas partes que agora chamamos de interceptores. Uma parte fica na rede fixa e a outra na rede móvel, ou seja, com o cliente. Eles são transparentes para o usuário e têm a função de minimizar esforços em ambos os lados, com o objetivo de amenizar alguns problemas inerentes ao ambiente.

O interceptor do lado do cliente móvel intercepta as chamadas desse cliente e juntamente com o interceptor do lado do servidor executam otimizações com a finalidade de reduzir a comunicação no canal sem fio.

Este modelo é adequado para dispositivos com capacidades computacionais de processamento e memória secundária suficientes e redes com bastante demanda computacional. Como desvantagem, obriga o desenvolvimento de um par de interceptores para cada tipo de aplicação.

3.5.2 Modelo Peer-to-Peer

Neste modelo, os dispositivos agem tanto como clientes quanto como fornecedores para seus pares, permitindo que os dispositivos trabalhem de modo cooperativo.

Este modelo é adotado em situações em que a ligação com a rede fixa não é necessária.

3.5.3 Modelo Agentes Móveis

Os agentes, com o intuito de realizar sua tarefa podem migrar de uma estação para outra, disparar a execução de novos agentes na rede ou ainda, interagir com agentes existentes. Adicionalmente, o agente executa sua tarefa de maneira autônoma e independente da aplicação que o invocou (MATEUS & LOUREIRO, 1998). Ao terminar a tarefa o agente retorna o resultado para a aplicação solicitante.

Uma importante característica dos agentes é sua capacidade de operar em modo desconectado. Dessa forma, depois de executar uma determinada operação, o agente aguarda até que o dispositivo esteja conectado novamente para enviar o resultado.

Os agentes são capazes de rodar em ambientes heterogêneos, o que permite a migração de uma estação para outra. Essa é uma importante característica no ambiente móvel *wireless*, já que, tais ambientes são compostos por equipamentos de diferentes fabricantes que executam sobre plataformas de software distintas.

Os principais problemas do modelo agentes móveis estão relacionados às questões de segurança: autenticação, privacidade, etc. Exemplo: agentes maliciosos, ou ainda, nodos maliciosos.

3.6 Conclusões

Este capítulo apresentou as redes WLAN, os elementos que compõe essas redes, principais características, topologias, a padronização existente para a tecnologia de

comunicação, etc. Além disso, procuramos demonstrar os desafios encontrados nesses ambientes.

Adicionalmente, as redes *Ad hoc* foram apresentadas de maneira mais extensa por fazer parte do cenário de nossa pesquisa.

4 Serviços Móveis: Principais Abordagens

4.1 Introdução

Serviços móveis têm despertado a atenção de diversos pesquisadores que buscam solucionar os diversos desafios identificados nessa área. Necessita-se de alternativas mais eficientes para o descobrimento dinâmico dos serviços, segurança, composição de serviços, entre outras. A seguir apresentamos os principais desafios relacionados a serviços móveis e alguns trabalhos que buscam solucioná-los.

4.2 Principais Desafios

Até pouco tempo atrás todos os serviços ou aplicações eram projetados para uma estrutura de rede fixa. Essas redes operam com taxas de transmissão altíssimas, onde o usuário dificilmente perderá a conexão com o servidor. Esse mesmo usuário poderá ser facilmente localizado, pois sua localização física não muda, podendo ser descoberto rapidamente pelos mecanismos de endereçamento utilizados nesses ambientes.

Atualmente, os usuários acostumados com as características acima mencionadas, deparam-se com a mobilidade a eles concedida através da evolução dos dispositivos móveis, assim como, da evolução das tecnologias de comunicação sem fio. A partir disso, surge uma série de desafios a serem superados para essa nova modalidade de serviços chamada serviços móveis. Certamente vamos utilizar a experiência adquirida com a evolução tecnológica alcançada nas mais diversas áreas da computação, mas devemos nos desprender um pouco dos conceitos de usuário fixo e pensarmos em um usuário móvel que deseja que os serviços móveis funcionem tão bem como os serviços já consagrados, ainda que, para isso, existam diversos desafios a serem superados. Entendemos serviço móvel como qualquer serviço a ser disponibilizado em redes móveis *wireless*, inclusive *web services*.

Há várias formas de tornar possível o uso de *web services* em ambientes *wireless*, o autor (PILIOURA et al., 2002) apresenta algumas delas, e destaca as principais vantagens e desvantagens:

- Dispositivo móvel agindo como cliente: neste cenário, encontramos desafios como a descoberta e execução automática dos *web services* e também a personalização dos mesmos, no que se refere ao perfil do usuário. Aqui, o dispositivo móvel terá uma aplicação *web service* cliente e se comunicará com o repositório de serviços e com o fornecedor do serviço através da rede *wireless*. Através da figura 4.1 podemos visualizar as entidades envolvidas neste cenário. No próximo exemplo, temos um PDA acessando o *proxy*, este dispositivo poderia ser um celular, já que, a rede *wireless* pode ser uma WLAN ou GSM/GPRS.

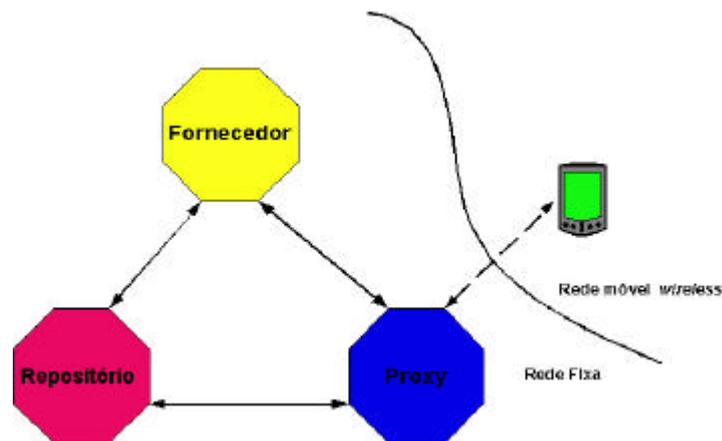


FIGURA 4.1 Cliente *web service*.

Podemos observar a partir da figura 4.1 que há um terceiro elemento localizado na rede fixa, o *proxy*. O *proxy* interage com o repositório de serviços e com o fornecedor do serviço, retornando os resultados ao dispositivo móvel. Assim, o dispositivo móvel não precisa suportar a carga do *web service* cliente. Uma das vantagens da utilização do *proxy* é que, no caso de perda de conexão com a rede, ele guarda os resultados da invocação do serviço até que a conexão seja restabelecida. Este cenário apresenta muitas vantagens e é aplicável no caso onde os usuários se movem dentro de um ambiente determinado. Uma desvantagem deste modelo é o fato do *proxy* se tornar um único ponto de falha, expondo a rede a ataques.

- Dispositivo móvel como fornecedor de serviços: este segundo cenário, permite levar serviços a qualquer lugar. Este cenário apresenta diversos desafios decorrentes da mobilidade do dispositivo fornecedor, a qual é bastante difícil de ser tratada. Mostra-se uma opção aplicável quando o número de clientes é pequeno. Uma maneira de gerenciar a localização do fornecedor seria manter uma tabela com o endereço dos clientes e notificá-los do novo endereço do fornecedor. Outras opções giram em torno de IP Móvel e APIs de atualização automática.

4.2.1 Descoberta de Serviços

Para tornar possível a descoberta de serviços em ambientes dinâmicos há diversos desafios a serem superados. Essa é uma área de pesquisa em ascensão, tendo em vista que os protocolos de descoberta de serviços projetados para as redes fixas não satisfazem as necessidades das redes *Ad hoc*, por estas redes precisarem de descoberta de serviços mais dinâmicas, como, por exemplo, descoberta de serviços baseada em proximidade, mínima administração e maior tolerância à falhas. Um protocolo de descoberta de serviços deverá ser capaz de descobrir os serviços com tempo de resposta satisfatório e não poderá inundar a rede com informações de controle. Tais objetivos são conflitantes e deverão ser balanceados.

Dentro desse enfoque, os autores (LEE et al., 2003) desenvolveram Konark, um protocolo distribuído projetado para redes *Ad hoc, peer-to-peer*. Este protocolo tem duas principais características, a descoberta e a entrega de serviços independentemente de dispositivo.

Konark usa HTTP e SOAP para lidar com a entrega de serviços; ele fornece um servidor HTTP em cada dispositivo que pode lidar com as requisições de serviços. As requisições e respostas são baseadas em SOAP. Além disso, cada dispositivo deverá ter um repositório de serviços com informações sobre serviços oferecidos por outros dispositivos. Os dispositivos têm a capacidade de hospedar e fornecer seu serviço a outros usuários usando o servidor local HTTP, além de consultar a rede por serviços

oferecidos por outros participantes e utilizar os serviços descobertos, permitindo assim que cada dispositivo possa agir como cliente e servidor simultaneamente.

Como ponto de desvantagem, todos os dispositivos que queiram utilizar a proposta deverão ter previamente instalado uma aplicação Konark para iniciar e controlar a publicação, a descoberta e uso do serviço. As funcionalidades do Konark são disponibilizadas através de um conjunto de APIs.

4.2.2 Contexto do Usuário

Outro tema bastante pesquisado atualmente é a propriedade de disponibilizar serviços de acordo com o contexto do usuário. O Contexto pode ser entendido como o tipo do dispositivo, a localização física do usuário, as preferências pessoais ou preferências do grupo no qual o usuário está inserido. Essa propriedade é de extrema valia, visto que, dentre as inúmeras vantagens obtidas através da sua utilização, podemos citar o fato de que com sua adoção, não inundaríamos a rede oferecendo serviços a um determinado usuário dos quais ele nunca teria interesse.

Em sua pesquisa, (CABRI et al., 2003) definem um framework para disponibilizar *M-services* adaptados ao contexto do usuário. O trabalho consiste em adaptar o serviço dinamicamente a estas características, estando focado principalmente nas características de localização física do usuário. Os serviços móveis poderão ser acessados por usuários móveis ou por agentes de software que se movem sobre a rede e possuem a capacidade de interagir com outros agentes.

Os autores estudam como adaptar o modelo apresentado às redes móveis *Ad hoc* e deixá-lo acessível através de SOAP, WSDL e UDDI.

Adicionalmente, devemos mencionar que a utilização de agentes dentro do cenário de serviços móveis é um paradigma importante a ser ainda mais explorado.

4.2.3 Acessar Serviços em Ambientes Móveis

Visto que as peculiaridades dos ambientes móveis obrigam que sejam reformulados os mecanismos que foram projetados para os ambientes fixos para que o

acesso e a conseqüente utilização dos serviços móveis sejam realizados de forma satisfatória, devemos pesquisar meios que torne eficiente o acesso a esses serviços, assunto esse que engloba um pouco de cada um dos itens mencionados anteriormente. Esses mecanismos deverão de algum modo lidar com os desafios que encontramos nos ambientes móveis quando tentamos portar, por exemplo, a arquitetura *web services* ao mundo *wireless* de serviços móveis. Seguindo essa linha descrevemos a seguir três trabalhos na área de acesso a serviços móveis: Devemos ressaltar que o trabalho dos autores (YANG et al., 2003) é muito interessante e nossa proposta apresenta-se bastante similar em alguns aspectos. Como diferencial, os autores aprofundam a pesquisa demonstrando três técnicas para o acesso ao canal sem fio: *B+Tree*, *Signature Indexing e Hashing*. *Signature Indexing* é o mais adequado método de acesso para o canal *wireless*, seguido do método *hashing* para aplicações onde energia é um fator crítico. Veremos a seguir os três trabalhos ora mencionados:

Os autores (YANG et al., 2003) descrevem uma estrutura para organizar e acessar *web services* eficientemente em ambientes *broadcast*. Foi definido um modelo multicanal para transmitir as informações sobre *M-Services* dentro de uma determinada área geográfica. O Canal UDDI inclui informação de registro sobre os *m-services*. O canal *m-services* possui a descrição e o código executável de cada *m-service*. O canal de dados contém os dados atuais necessários para execução do *m-service* no dispositivo móvel. A figura 4.2 apresenta a interação do usuário móvel com esses canais e do registro local UDDI com o canal UDDI com o intuito de manter os serviços atualizados.

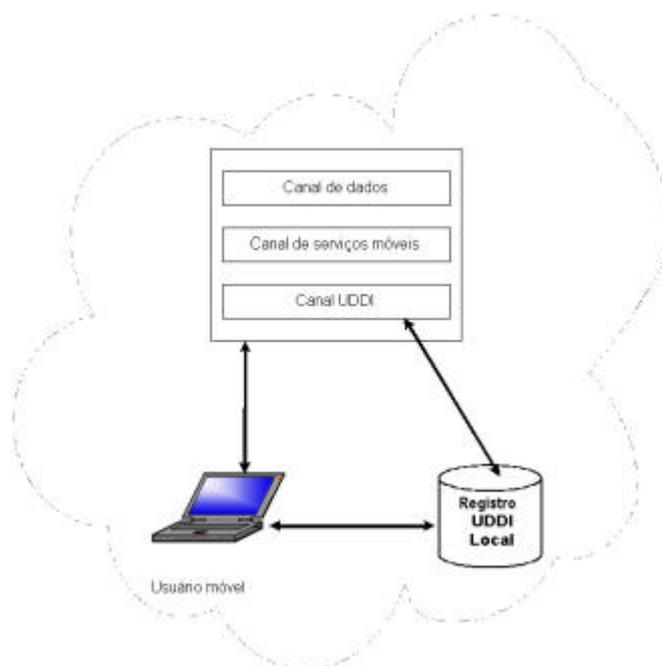


FIGURA 4.2 Arquitetura baseada em multi-canais.

A pesquisa apresenta três modos para acessar *m-services*: *at-hand*, *on-demand* e *broadcast*.

No modo *broadcast*, no qual foi fundamentada a pesquisa, um *service broker* periodicamente transmite os serviços disponíveis sobre o canal *wireless*. Clientes escutam o canal, identificando o serviço de interesse e baixam o serviço para execução local. Este modo mostra-se o mais adequado em ambientes onde temos um grande número de clientes em movimentação, a performance não depende do número de clientes e também não sobrecarrega o servidor com requisições de clientes.

O segundo trabalho é a pesquisa apresentada em (CAO et al., 2004) no qual define um *framework* para o acesso aos dados, permitindo que os nodos armazenem dados ou apenas o caminho para estes dados, com o objetivo de reduzir o atraso e tornar os dados disponíveis. Os nodos devem armazenar dados diferentes daqueles armazenados por seus vizinhos, aumentando assim a disponibilidade a esses dados, embora isso cause um maior atraso, pois os nodos devem acessar os dados de seus vizinhos no lugar de acessar os dados localmente.

O terceiro trabalho têm seu foco principal no armazenamento de *web services* em redes *Ad hoc* para tornar os serviços mais acessíveis a dispositivos móveis. O autor (FRIEDMAN, 2002) define que cada serviço deverá ter um único *proxy* dentro de cada

rede *Ad hoc* no qual o serviço está sendo usado, ou seja, todas as invocações a um determinado serviço dentro desta rede, deverão ser feitas a este *proxy*.

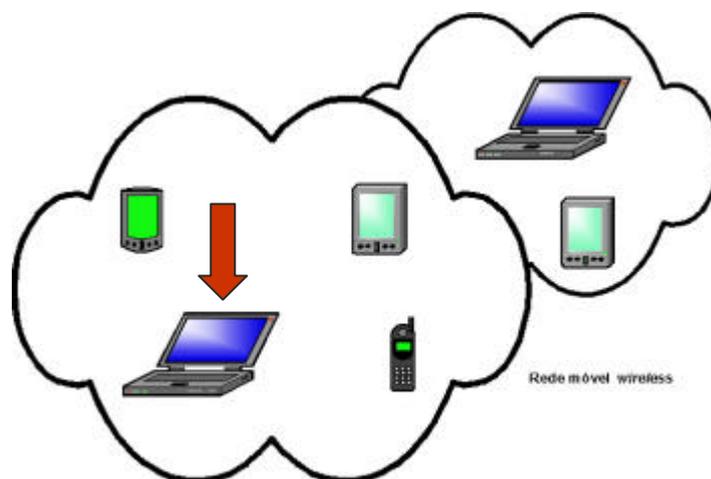


FIGURA 4.3 Dispositivo *Proxy*.

Podemos observar através da seta na figura 4.3 o nodo escolhido para ser o *proxy* nessa rede. O autor lida com alguns desafios como, por exemplo, escolher a melhor localização para o *proxy*, garantir que pelo menos um dispositivo atuará como *proxy*, além de garantir que ele continuará a atuar como *proxy* até que outro dispositivo seja escolhido, no caso de desistência, por exemplo.

Outra consideração importante assumida pelo autor é que o primeiro nodo a acessar um serviço, se tornará o *proxy*. Caso o nodo fique sobrecarregado ele poderá repassar alguns serviços para outro nodo.

4.2.4 Composição de Serviços

A composição de serviços é um assunto bastante pesquisado dentro da arquitetura *web services* numa projeção para as redes fixas, permitindo que diversas aplicações interajam com a finalidade de executar tarefas diversas. Para a realização dessas, enquanto que algumas aplicações realizam pequenas tarefas, outras poderão realizar tarefas bem mais complexas. A composição de serviços é utilizada para aderir os elementos participantes a fim de obter o resultado final numa computação cooperativa.

O trabalho do autor (STEELE, 2003) propõe a composição de serviços em redes *Ad hoc*. Para isso, é apresentado um sistema baseado em *web services* que permite uma integração dinâmica e temporária de módulos de software ou aplicações em redes *Ad hoc*. O trabalho adota a arquitetura *wireless extended internet*, onde existe um servidor com quem os dispositivos móveis se comunicam. O serviço poderá ser acessado por qualquer usuário que esteja na área de cobertura do ponto de acesso. Neste ambiente, o usuário móvel utiliza o ponto de acesso para acessar a rede e pesquisar por serviços disponíveis em um UDDI privado. O repositório UDDI estará armazenado em um servidor. Este servidor terá também, o código de alguns dos serviços oferecidos pelo UDDI.

O autor introduz brevemente o assunto de agentes inteligentes, sendo esta, conforme mencionado anteriormente na seção 4.2.2, uma área para pesquisas futuras relacionadas com *web services* e mobilidade.

O principal enfoque do trabalho esta na composição de serviços automatizada para redes *Ad hoc*, utilizando para isso, os padrões UDDI, WSDL, BPEL4WS.

Devemos observar que embora o trabalho seja projetado para redes *Ad hoc*, ele utiliza para disponibilizar os serviços e mais especificamente a composição dos serviços uma infra-estrutura fixa que irá limitar a utilização dos mesmos, portanto, o serviço não será disponibilizado de modo *Ad hoc*.

4.2.5 Segurança

Não poderíamos deixar de mencionar o assunto segurança, indispensável para qualquer ambiente computacional, seja ele móvel ou fixo. Esse é também um tema bastante pesquisado nas redes móveis.

O trabalho dos autores (WANG et al., 2003), apresenta uma técnica para o controle de acesso seguro e flexível para *M-Services* baseado em RBAC (controle de acesso baseado em papéis). A arquitetura para o controle de acesso é formada por algumas entidades, um TCC (*trusted credential center*), um *trusted authentication*, um *registration center* (TARC), e um mecanismo baseado em *ticket* seguro para o acesso a

serviços. Nessa arquitetura, *tickets* são emitidos pela entidade TCC e estes, carregam informações de autorização, necessárias para a utilização dos serviços.

Os *tickets* podem especificar regras de controle de acesso para diferentes tipos de serviços. Quando um usuário apresenta um *ticket* para o fornecedor do serviço, ele verifica a validade do *ticket* e o serviço é fornecido baseado nos privilégios de controle de acesso.

4.3 Conclusão

Através da revisão bibliográfica realizada concluímos que ainda há muito trabalho a ser realizado antes que os usuários dos ambientes móveis possam dispor de serviços disponíveis a qualquer hora e em qualquer lugar.

Atualmente os serviços estão disponíveis dentro do alcance das redes móveis, alcance esse determinado pelas tecnologias de comunicação atuais.

Necessitamos de mecanismos mais eficientes para a busca dinâmica de serviços, para prover segurança para as transações realizadas sob esses ambientes, garantias de qualidade de serviço, técnicas eficientes de replicação de dados, mecanismos de roteamento poderosos, etc.

A principal diferença da nossa proposta com os trabalhos aqui apresentados é que o nosso modelo disponibiliza o serviço móvel de modo *Ad hoc*, não forçando o dispositivo a estar conectado a rede fixa para acessar o serviço. Alguns outros trabalhos vistos nesse capítulo poderão ser utilizados para complementar o modelo proposto em pesquisas futuros.

5 UM MODELO PARA O PROVIMENTO DE *WEB SERVICES* EM REDES *AD HOC*

5.1 Introdução

Apesar da grande facilidade em implementar redes de comunicação fixas nos dias atuais, com a grande diversidade de produtos de boa qualidade e a custos acessíveis, a adoção de redes sem fio e, particularmente, das redes *Ad hoc* surge como uma tendência natural, graças ao desenvolvimento e aprimoramento das diversas modalidades de equipamentos móveis, aos avanços das tecnologias de comunicação presenciados nos últimos anos, e das diversas vantagens trazidas com a mobilidade concedida aos usuários dessas redes.

Evidentemente, a conectividade entre redes móveis e redes fixas é uma necessidade para grande parte das aplicações, o que já é suportado por um grande número de equipamentos e softwares desenvolvidos para este fim.

Por outro lado, a operação, de modo completamente autônomo (a saber, sem interação com alguma rede fixa ou entidade de coordenação) de redes *Ad hoc*, é um ponto a ser explorado, o qual se justifica pelo crescimento de aplicações que apresentam como requisito esta autonomia. Alguns exemplos de aplicações onde a autonomia é um fator importante são aplicações militares, de sobrevivência, desastres, comunicação entre ilhas, etc (MOHAPATRA et al., 2004).

Outro ponto importante a ser tratado é a possibilidade de diversos equipamentos portáteis poderem disponibilizar serviços aos demais equipamentos dentro de uma mesma rede, de modo que esses serviços estejam operantes durante grande parte do período de funcionamento de uma rede temporária para que as diferentes aplicações que a utilizam possam ser acessadas à medida que haja necessidade.

Este capítulo apresenta um modelo que permite a execução de *web services* em redes *Ad hoc*. O modelo visa tornar possível a qualquer dispositivo dessa rede oferecer um conjunto de serviços a qualquer outro dispositivo na mesma rede. Tendo em vista a usual carência de serviços que os usuários de redes *Ad hoc* possuem, nossa contribuição

busca permitir que tais usuários disponham do acesso a alguns serviços, que geralmente estão disponíveis apenas na rede fixa, ou ainda, necessitam dela no momento da utilização desse serviço.

5.2 Arquitetura

Uma primeira preocupação ao se pensar em termos de modelo, é qual arquitetura adotar, particularmente no que diz respeito à interatividade da rede *Ad hoc* com redes fixas.

Em seu trabalho, para o desenvolvimento de serviços *wireless*, os autores (YUAN et al., 2002) sugerem três alternativas de arquitetura as quais serão apresentadas a seguir.

5.2.1 *Wireless Portal Network*

Nessa arquitetura, os clientes *wireless* acessam os serviços da internet através de pontos de acesso ao portal. O portal cria uma espécie de “jardim gradeado” cujo acesso é controlado nos pontos de entrada. Esta arquitetura é adequada para o suporte à comunicação de dispositivos do tipo *thin-client*, para uso de tecnologias, como por exemplo, WAP.

De forma geral, um portal de *web services* WAP funciona da seguinte forma:

- O cliente *wireless* encaminha uma solicitação de informação na forma de um formulário WML (*Wireless Markup Language*). WML é uma linguagem de marcação baseada em XML usada para descrição de conteúdo e interface de usuário para dispositivos com restrições computacionais.
- O portal recebe a solicitação, verifica os privilégios de usuário do cliente e encapsula a solicitação numa mensagem SOAP para o servidor que contém o *web service*;

- Após a execução do *web service*, a resposta obtida é traduzida na forma de um documento formatado em WML, o qual é enviado para o cliente para ser exibida no display.

Esta alternativa de arquitetura revela-se interessante para situações em que se necessite acesso rápido num ambiente *wireless*. Pelo fato de suportar clientes leves (*thin-clients*), não requer alto poder de processamento dos dispositivos envolvidos. Além disso, determinadas tarefas que exigem maior poder computacional só poderão ser realizadas por computadores mais robustos, como, por exemplo, servidores de uma rede fixa.

Esta arquitetura, do ponto de vista do uso dinâmico de *web services*, pode apresentar alguns inconvenientes, os quais são relacionados a seguir:

- O fato de se basear num componente centralizado caracteriza-se como um ponto passível de falha, podendo expor a rede como um todo a ataques;
- Clientes leves são limitados do ponto de vista da interface;
- Clientes leves não possuem recursos para memorizar o estado de aplicações mais complexas, o que pode forçar o usuário a ter de navegar por inúmeras páginas até poder completar uma tarefa complexa (como, por exemplo, transações);
- Clientes leves suportam um conjunto limitado de protocolos de segurança, não permitindo o estabelecimento de políticas de segurança específicas da aplicação; no caso de um portal WAP, um *gateway* mal configurado pode gerar falhas de segurança.

5.2.2 Wireless Extended Internet

Essa arquitetura estende os recursos disponíveis na internet aos dispositivos móveis. Tem como característica não necessitar de uma entidade de controle. Para

fornecer acesso aos serviços utiliza *hubs web services* descentralizados que fornecem suporte para os *wireless web services*. Os *hubs* permitem levar a esses ambientes serviços que busquem minimizar as disparidades entre o ambiente fixo e móvel, como por exemplo, os diversos tipos de *displays* dos dispositivos móveis ou ainda, esquemas de segurança. A comunicação é XML sobre HTTP e uma conexão TCP/IP do dispositivo móvel pode ser ativada.

Segundo essa arquitetura, os dispositivos móveis caracterizam-se por um maior poder de processamento, suportando interfaces mais sofisticadas, processamento *offline* e a execução automática de transações. Também oferece suporte à implementação de políticas de segurança flexíveis, dependentes da aplicação, o que é um aspecto positivo quando comparada à alternativa baseada em clientes leves.

Ainda, como esta arquitetura sugere uma extensão da internet, a comunicação é descentralizada, o que evita a existência de um ponto de vulnerabilidade no que diz respeito ao acesso não autorizado a informações e outros recursos.

5.2.3 Wireless Ad hoc Network

Essa arquitetura permite que os dispositivos *wireless* tornem-se servidores para seus pares, habilitando-os a fornecer conteúdo, roteamento do tráfego na rede, e muitos outros serviços. Por ser uma tecnologia relativamente recente, esta proposta apresenta ainda alguns aspectos a serem investigados, como as questões de desempenho e segurança.

O modelo proposto neste trabalho vai adotar como referência a arquitetura *wireless Ad hoc network*, como já discutido em (FARIA et. al., 2004).

Escolhemos este perfil de arquitetura por ele revelar-se ideal para as redes *Ad hoc*, pois não utiliza nenhuma infra-estrutura fixa para disponibilizar o serviço aos clientes, que é um aspecto que consideramos importante no contexto deste trabalho. Observamos ao longo da pesquisa que usuários de redes *Ad hoc* utilizam os serviços sempre com o auxílio de alguma infra-estrutura fixa, invalidando de certo modo as características intrínsecas ao seu ambiente que seria executar esses serviços realmente de modo independente.

A figura 5.1 ilustra a arquitetura do modelo de forma geral. Como ilustrado nessa figura, nosso modelo foi projetado para usuários de *notebooks* e PDA's, dando ênfase ao aspecto de autonomia e disponibilidade a aplicações onde estes requisitos sejam considerados fundamentais.

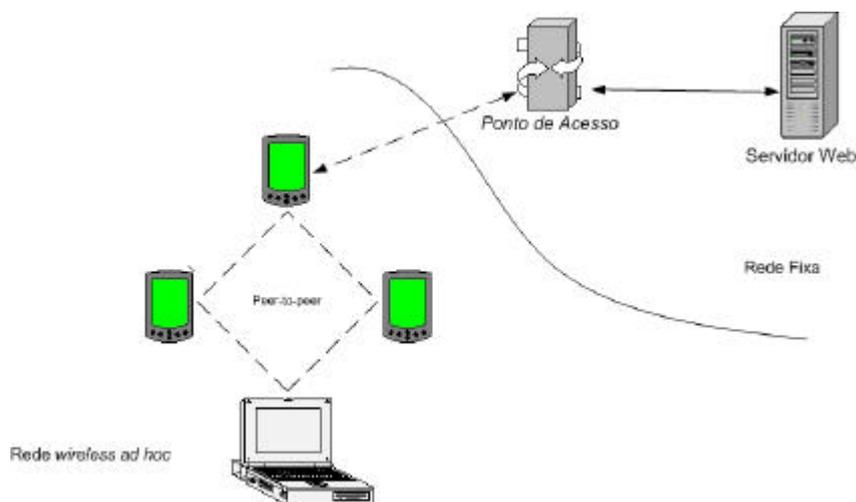


FIGURA 5.1 Arquitetura geral do modelo.

5.3 Conectividade

Com o intuito de lidar com alguns dos desafios do ambiente móvel como, por exemplo, a perda freqüente de conexão e observando que a capacidade dos dispositivos móveis vem aumentando consideravelmente, o modelo proposto executará o *web service* localmente.

De modo a ilustrar um serviço implementado de acordo com o modelo proposto descrevemos seu funcionamento:

- **Conexão com a rede fixa**

O servidor conectado à rede fixa contém os módulos de software necessários para suprir um dispositivo móvel que queira tornar-se fornecedor de serviço para seus pares em uma rede *Ad hoc*. Denominamos esses módulos de módulo móvel e módulo fixo.

O módulo móvel do servidor conectado à rede fixa contém um repositório UDDI privado contendo todos os serviços que esse servidor pode disponibilizar, e mais a aplicação fornecedora e a aplicação cliente de cada serviço publicado nesse repositório UDDI.

O módulo fixo contém a máquina virtual e a base de dados. Partimos do princípio que a máquina virtual e a base de dados, ou seja, o módulo fixo estará previamente instalado nos dispositivos que formam a rede. Por tal motivo, denominamos este módulo de módulo fixo. Ressaltamos que a escolha da máquina virtual assim como da base de dados utilizada estão diretamente vinculadas ao serviço.

O usuário do dispositivo móvel, ou seja, o usuário que será o fornecedor do serviço para seus pares na rede *Ad hoc* acessa a rede fixa apenas uma vez para pegar o serviço, depois disso ele poderá na rede *Ad hoc* fazer o papel do fornecedor tornando possível o acesso aos serviços sem a necessidade de qualquer outra comunicação com a rede fixa.

Deve-se observar que a interação do dispositivo móvel com a rede fixa ocorre apenas neste momento, a partir disso, o usuário móvel têm autonomia suficiente para ser um fornecedor de serviços em uma rede *Ad hoc*.

Uma vez estabelecida a comunicação entre o dispositivo móvel e o servidor conectado a rede fixa, será realizada a transferência do módulo móvel. Essa comunicação poderá ser estabelecida através de duas opções. Na primeira, o dispositivo móvel é conectado fisicamente ao servidor. Já na segunda opção o dispositivo móvel acessa a rede fixa através de um ponto de acesso.

Haja visto que o dispositivo móvel foi suprido com o serviço de interesse, ele torna-se um DMF (Dispositivo Móvel Fornecedor), estando apto a fornecer tal serviço a seus pares em uma rede *Ad hoc*, não necessitando se conectar em qualquer outro momento ao servidor na rede fixa.

- **DMF**

Concluída a etapa anterior, o usuário do DMF dispara na rede no qual ele esta localizado, uma mensagem *broadcast* oferecendo a seus pares o serviço anteriormente selecionado e adquirido através da rede fixa. Limitamos os serviços à opção *read-only*,

pois não lidamos com as questões de consistência dos dados, sendo isso, outra fonte de pesquisa que têm atraído a atenção de diversos pesquisadores. A consistência dos dados embora seja um tema a ser bastante explorado em serviços móveis está fora do escopo do nosso trabalho de pesquisa.

Os dispositivos que compõe a rede recebem a mensagem enviada pelo DMF e optam entre aceitar ou recusar o serviço. Os dispositivos interessados retornam a mensagem ao DMF e tornam-se nesse momento dispositivos clientes, estabelecendo assim, uma conexão com o DMF requisitando o módulo móvel.

O intervalo entre essas mensagens *broadcast* enviadas à rede pelo DMF é controlado pelo próprio usuário. Dessa forma, um serviço não será oferecido em um momento inadequado, como, por exemplo, num dado instante em que ele está utilizando seu dispositivo para uma determinada tarefa que necessite total prioridade de seus recursos computacionais.

Uma vez terminada a transferência dos arquivos necessários, a execução do serviço será realizada localmente, minimizando assim, alguns dos desafios inerentes ao ambiente móvel como a freqüente perda de conexão, a energia limitada dos dispositivos, entre outros que foram mencionados anteriormente. Nesse momento, o dispositivo cliente que efetuou a transferência dos módulos, além de utilizar o serviço, se desejar, também poderá agir como DMF em qualquer rede *Ad hoc*. Dessa forma, poderemos num dado momento ter em uma mesma rede vários fornecedores e vários clientes, aumentando com isso, a disponibilidade dos serviços. No caso de um dispositivo (DMF) deixar a rede, um outro dispositivo que tenha requisitado um serviço anteriormente poderia oferecê-lo nessa mesma rede ou eventualmente, em outra rede.

Devemos observar que há então dois cenários para um dispositivo móvel cliente se tornar um DMF com a única diferença que o DMF que acessar a rede fixa para a transferência do módulo móvel terá disponível o diretório UDDI com a opção de escolher entre todos os serviços que estão disponíveis enquanto que o DMF que adquiriu o módulo móvel através da rede *Ad hoc*, ou seja, através de um DMF terá apenas um único serviço.

Adicionalmente, devemos complementar que o usuário terá a responsabilidade de gerenciar os recursos computacionais do seu dispositivo, controlando dessa forma, o número de vezes que ele vai oferecer o serviço à rede, ou seja, caso o usuário esteja com

muitas conexões ativas ou ainda com uma carga mínima da bateria, ele não enviará mais nenhuma mensagem *broadcast* para a rede oferecendo o serviço aos seus pares até que o seu status atual volte a uma opção confiável, onde ele possa fornecer o serviço de forma satisfatória.

- **Funcionamento**

Conforme descrito no quadro 5.1, o funcionamento do modelo proposto se dá através das etapas A, B, C, D:

QUADRO 5.1 Etapas do funcionamento do modelo.

Etapa	Ação
A	O DMF dispara uma mensagem broadcast.
B	Os dispositivos A e B solicitam o serviço ao DMF.
C	Será estabelecida uma conexão entre DMF e dispositivos A e B para a transferência do módulo móvel.
D	Execução local do serviço.

No caso de queda de conexão, ou a bateria do equipamento descarregue durante uma transferência, o cliente aguardará por uma nova mensagem *broadcast* que deverá ser enviada pelo DMF.

Como extensão a essa pesquisa, devemos implementar um mecanismo que controle essa interação a fim de dar continuidade na transferência interrompida do módulo requerido ao invés de iniciá-la novamente, minimizando dessa forma, a utilização dos recursos da rede. Adicionalmente, deverá ser medido o custo desse controle para avaliar sua real viabilidade. Além disso, devemos desenvolver uma ferramenta que controle e informe ao DMF alguns dados como, por exemplo, a qualidade do link, o número de usuários conectados, dentre outras informações que sejam importantes para evitar sobrecarga tanto no dispositivo, como na rede.

O DMF guarda uma tabela contendo informações sobre a etapa c, com isso, evitamos que os dispositivos clientes peguem mais de uma vez o mesmo serviço, tornando possível realizar o controle de versões.

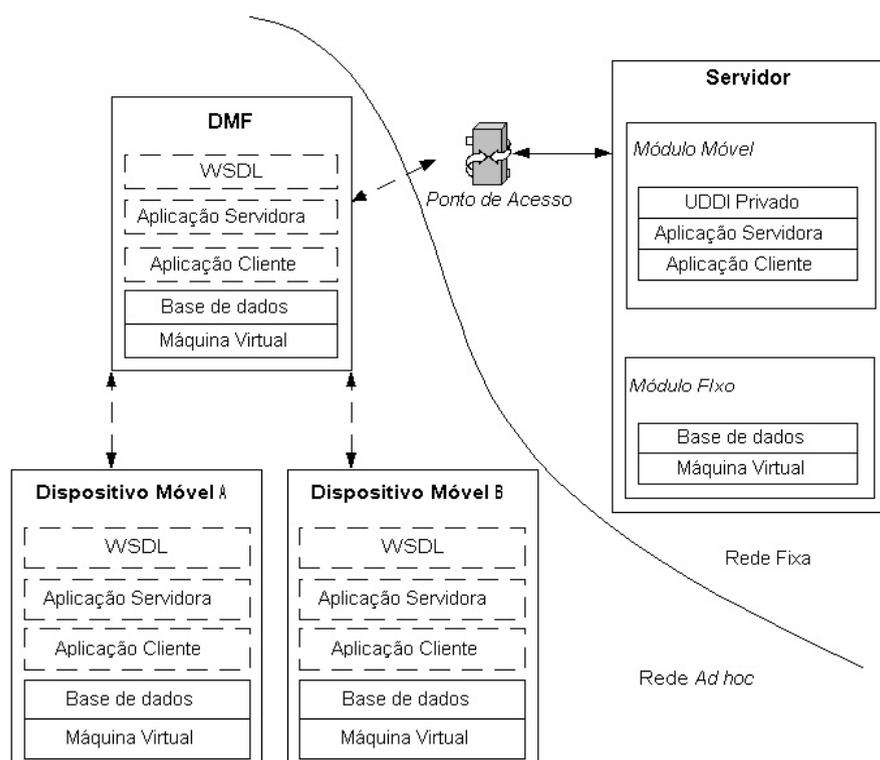


FIGURA 5.2 Modelo Proposto.

A figura 5.3 apresenta o modelo de forma um pouco mais detalhada. Através dessa figura observamos a interação com a rede fixa. Podemos visualizar as duas opções existentes para a conexão com o servidor *web*, Opção A, através de um meio físico e Opção B, através do link *wireless*.

Posteriormente, o dispositivo móvel requisita o módulo móvel através de uma das opções A ou B e se torna um DMF. Concluída essa etapa, o serviço pode ser fornecido de modo *Ad hoc* aos usuários da rede.

Possuindo acesso físico ao servidor na rede fixa (opção A), ou ainda, autenticando-se na rede *wireless* (opção B), qualquer dispositivo móvel poderá se tornar um DMF.

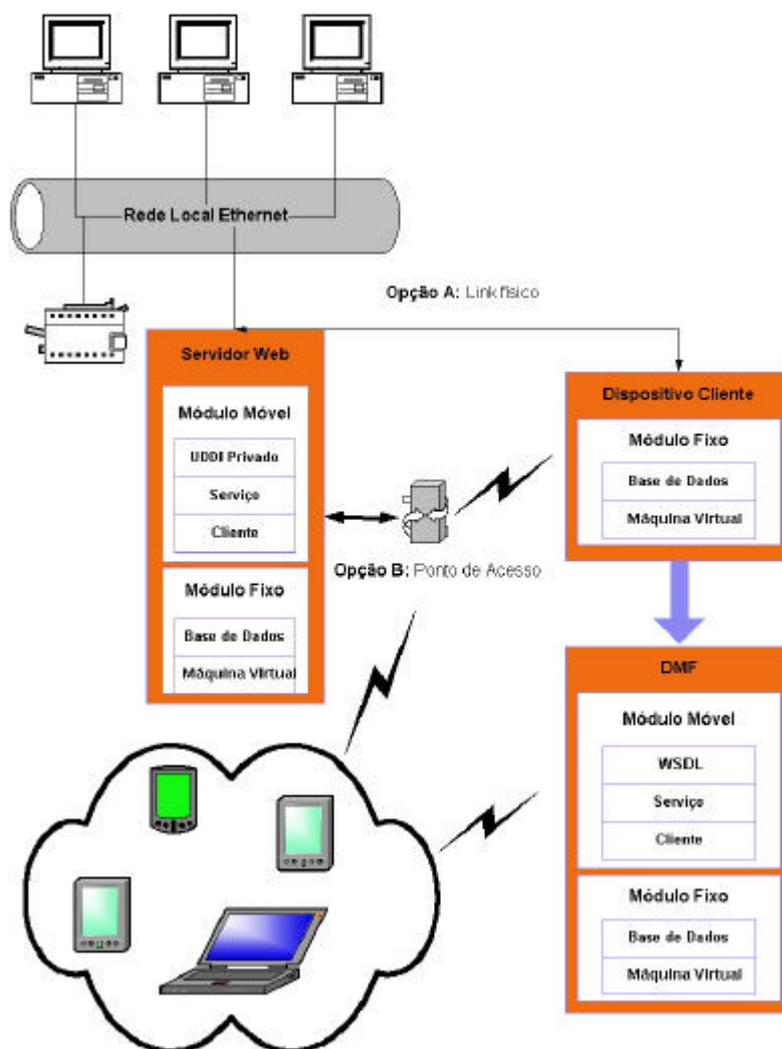


FIGURA 5.3 Modelo Proposto detalhado.

Já que estamos trabalhando na camada de aplicação e não na camada de rede, o problema de roteamento não foi abordado no modelo, tendo em vista que, esse assunto não faz parte do enfoque deste trabalho.

Para a realização dos testes, devido à falta de padronização quanto ao mecanismo de endereçamento, foram atribuídos endereços IP aos dispositivos. Adicionalmente, partiu-se do princípio que os dispositivos móveis na rede *Ad hoc* não possuem disparidades quanto ao mecanismo de endereçamento utilizado. Esta ainda é uma questão de pesquisa extremamente pesquisada nas redes *Ad hoc*.

5.4 Aplicabilidade

Os tipos de serviços que se beneficiariam com a utilização do modelo são aqueles que se fazem necessários em lugares onde utilizar uma infra-estrutura de rede fixa não é viável, seguindo a motivação para a utilização das redes *Ad hoc* e tornando realística a utilização de serviços sobre essas redes. Alguns exemplos como, por exemplo, consulta a informações necessárias durante a pulverização de agrotóxicos em uma lavoura de arroz, ou ainda, informações sobre a área de replantio de soja. Além dessas, apresentamos um estudo de caso logo abaixo que tem como objetivo demonstrar a utilização de um serviço conforme o modelo proposto.

5.4.1 Estudo de caso

Uma dificuldade usual encontrada pelo corpo de bombeiros é a falta de um serviço que informe, por exemplo, onde estão localizados os hidrantes mais próximos do local onde esta acontecendo um incêndio. A busca por essas informações é de extrema importância, mas toma um tempo precioso que em muitos casos, vale a vida das pessoas envolvidas numa tragédia.

Uma vez aberta uma chamada para atender a um incêndio, o usuário portando um dispositivo móvel faz uma conexão com servidor que contém o serviço de localização de hidrantes e adquire o módulo móvel se tornando um DMF. Estabelecida a rede *Ad hoc*, o serviço poderá ser oferecido por esse usuário aos seus pares na rede. A partir disso, podemos verificar que o modelo torna factível levar serviços a lugares inacessíveis a rede fixa, lugares onde a necessidade é temporária, buscando de fato, uma computação disponível a qualquer hora e em qualquer lugar.

Deve-se observar que a conexão com a rede fixa ocorre apenas em um momento, para que um dispositivo se torne um DMF, posterior e essa etapa, o serviço será oferecido e conseqüentemente utilizado sem o auxílio de qualquer infra-estrutura fixa.

5.5 Avaliação da Proposta

Devido às limitações de tempo decorrentes da realização desta dissertação e considerando as dificuldades de infra-estrutura para definir um ambiente experimental adequado a uma completa validação do modelo, decidiu-se realizar uma implementação de modo a se fazer uma avaliação preliminar do modelo.

A definição de uma metodologia de validação mais apurada fica, então, como uma proposta de continuidade desta pesquisa, fora, porém, do contexto da dissertação de mestrado.

5.5.1 Elementos utilizados na parte experimental

A implementação realizada foi baseada no uso de 2 PDAs, sendo que a plataforma de desenvolvimento adotada foi baseada na linguagem Java. Uma descrição mais detalhada destes elementos é realizada nas seções que seguem.

5.5.2 Dispositivos móveis

Como dispositivos móveis, adotou-se os equipamentos Palm Tungsten C (Figura 5.4) disponíveis no LabWEB da UFSC, os quais são equipados com processador Intel Xscale, de frequência 400 MHz, com 64 Mb de memória RAM, com interface WiFi (IEEE 802.11b) e infravermelho, rodando o sistema Operacional PalmOS, versão 5.2.1.

Estes equipamentos foram escolhidos pelo fato de encontrarem-se disponíveis no momento da implementação, por apresentarem uma excelente relação custo/benefício e também por representarem de modo fiel o padrão de equipamentos utilizados nas aplicações típicas de redes *Ad hoc*. A autonomia deste equipamento em termos de energia é estimada em aproximadamente 7 horas.



FIGURA 5.4 Dispositivo móvel adotado no ambiente experimental.

5.5.3 A Rede Fixa

Em nosso ambiente, a rede fixa utilizada é aquela instalada no LSD (Laboratório de Sistemas Distribuídos), a qual é composta de 6 microcomputadores com processadores AMD, rodando windows 2000 e windows XP. A rede é baseada no padrão Ethernet (IEEE 802.3) e é conectada à rede do Departamento de Informática e de Estatística da UFSC, que possui acesso permanente à internet.

O servidor *web* conectado a rede fixa, contendo os módulos necessários para a disponibilizar o serviço, está equipado com processador AMD Athlon, com frequência de 2,4GHz, com 512 Mb de memória RAM, 40Gb de disco rígido rodando windows XP.

5.5.4 Protocolos e outros elementos

No contexto do nosso modelo, a interação entre os principais padrões relacionados com a arquitetura *web services*, WSDL, SOAP e UDDI, mantêm-se a mesma.

As implementações tradicionais para SOAP e XML possuem um número de classes muito grande, o que dificulta portar os serviços desenvolvidos sobre essas bibliotecas para os dispositivos móveis. Essa é a principal motivação e justificativa para a adoção dos pacotes reduzidos kSOAP e kHTML com o objetivo de dar suporte a comunicação entre os dispositivos para fins de troca de informações sobre os *web services* (KOBJECTS, 2004).

O pacote kSOAP é uma API SOAP, baseada em kXML. A implementação kXML é um parser XML indicado para ambientes com restrições computacionais (KOBJECTS, 2004).

As bibliotecas kSOAP, kHTML e kXML são implementações Java totalmente abertas, com código fontes disponíveis. São bibliotecas reduzidas projetadas para o desenvolvimento J2ME.

5.5.5 Comunicação

A comunicação foi realizada através de *sockets*. Devido à falta de padronização quanto ao mecanismo de endereçamento, para a realização dos testes, foram atribuídos endereços IP aos dispositivos, tendo em vista que, esse assunto não faz parte do enfoque deste trabalho.

5.5.6 Escolha da Plataforma e opções de implementação

Como indicado anteriormente no capítulo 2, uma das principais vantagens envolvidas com a plataforma Java é que os serviços desenvolvidos a partir dela tornam-se operáveis em qualquer dispositivo, independentemente da arquitetura ou ainda, do sistema operacional desse dispositivo. Além disso, existem implementações disponíveis para Java da maior parte dos padrões envolvidos com a arquitetura *web service*, incluindo HTTP, SOAP, XML, e as versões reduzidas kHTTP, kSOAP, kXML, dentre outros.

O desenvolvimento do nosso serviço para as validações preliminares ao modelo proposto foi baseado na plataforma Java, particularmente no ambiente J2ME. O J2ME é voltado ao desenvolvimento de serviços projetados para dispositivos móveis que possuem restrições quanto ao poder de processamento, memória, capacidade de armazenamento, etc.

A SUN disponibiliza implementações independentes para o desenvolvimento de serviços móveis, neste caso, mais especificamente *web services*: *J2ME Wireless Toolkit* e as APIs CLDC (*Connect Limited Device*) e MIDP (*Mobile Information Device Profile*). As versões utilizadas para o desenvolvimento do serviço que foi usado para a realização dos testes foram MIDP 2.0, CLDC 1.1 e *Wireless Toolkit 2.2*.

A figura 5.5 ilustra a pilha de elementos necessárias ao desenvolvimento de um *web service* construído com base na plataforma J2ME.



FIGURA 5.5 Pilha necessária ao desenvolvimento de um *web service*.

É importante ressaltar aqui, que nossa arquitetura não está presa à plataforma de desenvolvimento. A escolha foi baseada no ambiente disponível para a realização dos testes. Na continuidade deste trabalho, deverão ser utilizadas outras plataformas para medir a relevância que a essa escolha exerce sobre o modelo.

Algumas limitações de tecnologia foram identificadas. Primeiramente, não foi possível a transferência do módulo móvel pelo enlace *wireless*, pois o J2ME não possui suporte para objeto serializável. Para contornar essa limitação, transmitimos pela rede a mesma quantidade de bytes do módulo em *string*. Tal substituição nos levou a partir do princípio que a transferência foi realizada satisfatoriamente.

A partir da figura 5.6 notamos o envio da mensagem broadcast e a requisição do serviço sendo realizada por um dispositivo na rede.

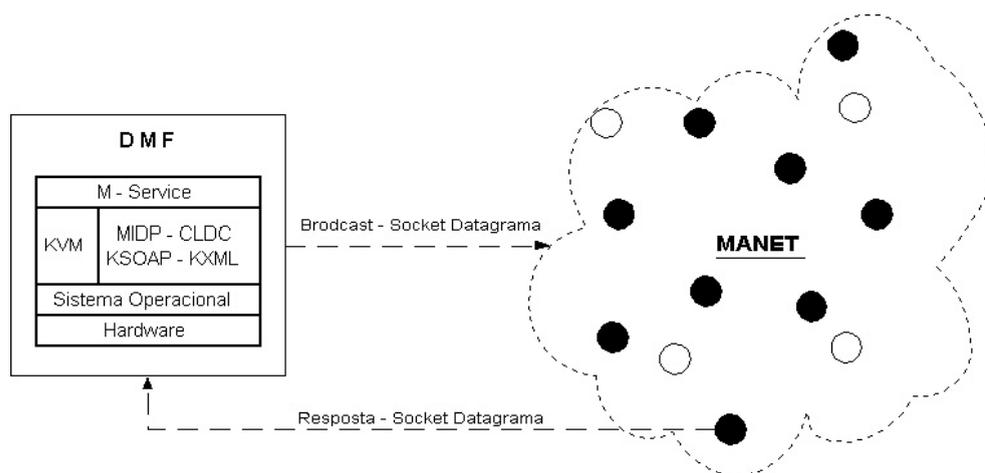


FIGURA 5.6 Ambiente experimental.

Podemos observar que para serviços que utilizam pouco processamento, o custo da transferência do serviço é maior do que executar o serviço no modo cliente-servidor, por isso, o DMF deverá escolher automaticamente a maneira de execução mais apropriada, adequando-se as características atuais do ambiente e do serviço.

5.5.7 Resultados da Avaliação Preliminar

Desenvolvemos uma aplicação provedora com alguns métodos públicos. Esses métodos foram descritos em uma interface WSDL que foi gerada automaticamente. A aplicação cliente invoca tais métodos, passando os parâmetros necessários para obter o resultado da requisição. O transporte foi feito sobre o protocolo HTTP. O serviço desenvolvido para a realização dos testes não utilizava base de dados.

De fato, há uma conexão com a rede fixa, mas ela está limitada somente ao momento em que o usuário móvel pega o serviço no servidor conectado a essa rede que contém um repositório UDDI privado com a descrição dos diversos serviços a serem disponibilizados.

Inicialmente realizamos alguns testes através de simulador. Em uma segunda etapa transferimos o serviço para o dispositivo móvel para realizar as transações em uma rede *Ad hoc* formada por dois dispositivos que agiram ora como cliente e ora como servidor do serviço. A figura 5.7 mostra o teste do serviço realizado sobre o simulador. É apresentado no *display* o resultado da invocação de um dos métodos do *web service*.



FIGURA 5.7 Teste realizado através do simulador.

Os testes foram satisfatórios. O modelo mostra-se uma alternativa interessante para o fornecimento de serviços a usuários móveis, preferencialmente, usuários das redes *Ad hoc*, já que os usuários de redes com infra-estrutura fixa contam com pelo menos uma entidade centralizadora facilitando o acesso aos serviços.

Embora existam modificações a serem realizadas no modelo, a aplicação utilizada como teste demonstra que é possível realizar a transferência e a execução de *web services* em redes *Ad hoc*. Posteriormente seria interessante o desenvolvimento de uma API que monitorasse o enlace e pudesse agir de modo inteligente juntamente com o DMF a fim de controlar o número de interações ativas entre DMF e dispositivos

clientes. Outros controles podem ser adicionados maximizando dessa forma, a utilização do modelo.

5.6 Conclusão

Este capítulo apresentou de forma detalhada a nossa proposta para o provimento de *web services* em redes *Ad hoc*. Demonstramos o funcionamento do modelo proposto, suas características e como ele torna factível disponibilizar serviços aos usuários dessas redes. Adicionalmente, apresentamos os dispositivos, protocolos, plataformas, dentre outros elementos utilizados para o desenvolvimento de um *web service* que teve como objetivo nos dar uma avaliação preliminar para a validação do modelo.

Concluimos que embora tenhamos muitas limitações e uma série de melhorias e recursos adicionais a serem aderidos a nossa proposta, a adoção do modelo para o desenvolvimento de serviços móveis minimiza os diversos desafios encontrados quando desejamos levar serviços aos usuários destes ambientes tornando-se um diferencial dentro da topologia *Ad hoc* por permitir a execução dos serviços sem auxílio de qualquer infra-estrutura fixa.

6 CONCLUSÕES E TRABALHOS FUTUROS

Os avanços nas tecnologias de comunicação sem fio, o aumento da performance dos dispositivos móveis e ainda o crescente número de usuários dessas tecnologias nos remetem para uma computação que necessita estar disponível no momento que o usuário desejar. Atualmente, assegurar a conectividade com a rede a qualquer hora e em qualquer lugar não é uma realidade já que os serviços ficam disponíveis apenas dentro de uma determinada área de cobertura. Tal fato deixa diversas lacunas a serem exploradas pela comunidade científica que busca constantemente minimizar ou resolver os diversos desafios que essa nova computação impõe.

A principal contribuição do trabalho de pesquisa aqui apresentado está na proposta de um modelo que possibilite disponibilizar serviços sem a necessidade de infra-estrutura fixa, possibilitando aos usuários o acesso a aplicações realmente de modo *Ad hoc*, explorando as características desses ambientes.

Tal modelo baseia-se nos padrões existentes para *web services* e leva em conta as restrições e dificuldades intrínsecas às redes *Ad hoc*, conforme mencionadas em capítulos anteriores. Além disso, torna o fornecimento de serviços na topologia *Ad hoc* o mais independente possível do acesso a redes fixas.

A validação da proposta foi realizada através do desenvolvimento de uma aplicação, projetada com base no modelo em questão, que demonstrou, de forma inegável, a viabilidade de utilizar servidores móveis. Os testes iniciais foram realizados utilizando dois dispositivos móveis formando a rede *Ad hoc*. Os dispositivos atuaram como clientes e também como fornecedores do serviço.

Comparada a modelos voltados ao oferecimento de *web services* encontrados na literatura, o modelo aqui proposto oferece mais flexibilidade e independência da rede fixa, adaptando-se de modo mais consistente à dinâmica das redes *Ad hoc*.

Evidentemente, por ser uma pesquisa que caracterizou a escolha do modelo, muitos aspectos restam a definir, o que poderá ser exposto como futuros trabalhos nesta área.

Extrapolando os objetivos atingidos com a realização desta pesquisa no contexto da dissertação de mestrado, pretendemos dar continuidade dentro deste tema, através da implementação de uma ferramenta que deverá monitorar a qualidade do enlace com o intuito de controlar o número de conexões ativas com o DMF. A ferramenta em desenvolvimento deverá permitir que o usuário permaneça na rede, em situações mais críticas, até o final de uma operação. Além disso, controlará o número máximo de conexões ativas de modo automatizado.

Uma outra direção de contribuição de trabalhos futuros incluem a questão da segurança e garantias de QoS para os serviços disponibilizados a partir do modelo aqui proposto.

REFERÊNCIAS BIBLIOGRÁFICAS

ABOLHASAN, M.; WYSOCKI, T.; DUTKIEWICZ, E. **A review of routing protocols for mobile ad hoc networks.** Elsevier Computer Science, <www.elsevier.com/locate/adhoc>. Disponível on-line, acesso em 21/09/2004, p. 1-22, 2003.

ANDERSON, Richard et al. **Professional XML**Rio de Janeiro: Editora Ciência Moderna Ltda, 2001.

BENATALLAH B.; CASATI, F., TOUMANI F. **Web Service Conversation Modeling: a cornerstone for e-business automation.** Internet Computing, IEEE, p. 46-54, Jan. 2004.

BENATALLAH B.; SHENG Q.Z.; DUMAS M. **The Self-Serv environment for Web services composition.** Internet Computing, IEEE, p. 40-48, Jan. 2003.

BENATALLAH B.; DUMAS M.; SHENG Q.Z.; et al. **Declarative composition and peer-to-peer provisioning of dynamic web services.** In: International Conference on Data Engineering, 18, San Jose, CA USA, p. 297-308, 2002.

CABRI, G.; LEONARDI, L.; MAMEI, M.; et al. **Location-dependent services for mobile users.** Transactions on Systems, Man and Cybernetics, Part A, IEEE, Volume: 33, Issue: 6, p. 667-681, Nov. 2003.

CAO, G., YIN, L., DAS, C. R. **Cooperative Cache-Based Data Access in Ad Hoc Networks.** IEEE Computer, p. 32-39, Fev. 2004.

CHUNG, J.; HIN, K.; MATHIEU, R. G. **Web Services Computing: Advancing Software Interoperability.** IEEE Computer Society, p. 35-37, Oct. 2003.

CORSON, S.; MACKER, J. **Mobile Ad hoc Networking (MANET)**, RFC 2501, Jan. 1999. Disponível on-line, acesso em 02/05/2004.

CURBERA, F.; KHALAF, R.; MUKHI, N.; et al. **The Next Step in Web Services**. Communications of the ACM. Vol. 46, Nº 10, p. 29-34, Oct. 2003.

DELICATO, F.C.; PIRES, P.F.; PIRMEZ, L.; et al. **A flexible web service based architecture for wireless sensor networks**. International Conference on Distributed Computing Systems Workshops, 2003. Proceedings. 23rd, p. 730-735, May. 2003.

ELSON, J.; ESTRIN D; **An Address-Free Architecture for Dynamic Sensor Networks**. Tech. Rep. 00-724, Computer Science Department USC, January 2000.
<<http://citeseer.ist.psu.edu/elson00addressfree.html>>

FARIA, F.; MAZZOLA, V.; DANTAS M.A.R. **Servidores Móveis em Redes Ad hoc**. II Escola Regional de Redes de Computadores, 129-134, Canoas - RS. Julho 2004

FERRE, P.; DOUFEXI, A.; NIX, A.; et al. **Throughput analysis of IEEE 802.11 and IEEE 802.11e MAC Wireless**. Communications and Networking Conference, 2004. WCNC. 2004 IEEE, Volume: 2, 21-25 p. 783-788. Mar 2004.

FORMAN, G.H.; ZAHORJAN, J.; **The challenges of mobile computing**. Computer, Volume: 27, Issue: 4, 1994, p. 38 – 47, April 1994

FRIEDMAN, R. **Caching Web Services in Mobile Ad-Hoc Networks: Opportunities and Challenges**, ACM Workshop On Principles Of Mobile Computing, p. 90-96, Oct 2002.

GAERTNER, G.; CAHILL, V.; **Understanding link quality in 802.11 mobile ad hoc networks**. Internet Computing, IEEE , Volume: 8 , Issue: 1 , p. 55-60, Jan-Feb 2004.

IEEE 802.11, Working Group for Wireless Local Area Networks. **IEEE 802.11 Technical report**. <<http://standards.ieee.org/wireless/overview.html#802.11>>. Disponível on-line, acesso em 27/09/2004.

KOBJECTS. **KSOAP, KXML** <<http://kobjects.org/>> Disponível on-line, acesso em 22/01/2005.

LEE, C.; HELAL, A.; DESAI, N.; et al. **KONARK: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services**. Transactions on Systems, Man and Cybernetics, Part A, IEEE, Volume: 33, Issue: 6. p. 682-696, Nov 2003.

MATEUS, G. R.; LOUREIRO, A. A. F. **Introdução a Computação Móvel**. DCC/IM, COPPE/Sistemas, NCE/UFRJ, 11a. Escola de Computação, 1998.

MICROSOFT CORPORATION. **Microsoft NET**. <<http://www.microsoft.com/Net/>> Disponível on-line, acesso em 20/01/2005.

MILENKOVIC, M.; ROBINSON, S.H.; KNAUERHASE, R.C.; et al. **Toward Internet Distributed Computing**, IEEE Computer, p. 38-46, May 2003.

MOHAPATRA, P.; CHAO GUI; JIAN LI. **Group communications in mobile ad hoc networks**. **Computer**, Volume: 37 , Issue: 2 , p. 52 – 59, Feb 2004.

OASIS Member Section, **Universal Description, Discovery e Integration** <<http://uddi.org/pubs/uddi-v3.0.1-20031014.htm>>. Disponível on-line, acesso em 12/06/2004.

OMG (Object Management Group). **The Common Object Request Broker: Architecture and Specification** Revision 2.0, July 1995.

PILIOURA, T.; TSALGATIDOU, A.; HADJIEFTHYMIADES S; **Scenarios of using web services in M-commerce**. ACM SIGecom Exchanges Volume 3, Issue 4 Winter, Mobile commerce, p. 28-36, Dec. 2002.

PIRES, P.F.; BENEVIDES, M.R.F.; MATTOSO, M. **Mediating Heterogeneous Web Services**. Symposium on Applications and the Internet, 2003. p. 344-347, Jan. 2003.

O'TUATHAIL E.; ROSE M. **Internet Official Protocol Standards**. RFC 3288, Jun 2002. Disponível on-line, acesso em 14/09/2004.

(SAX) **Simple API for XML**. <<http://sax.sourceforge.net/>>. Disponível on-line, acesso em 10/09/2004.

SATYANARAYANAN, M. **Fundamental challenges in mobile computing**. Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing. p. 1-7, May 1996.

STEELE, R. **A Web Services-based System for Ad-hoc Mobile Application Integration**. Coding and Computing ITCC International Conference. IEEE, p. 248-252. Apr. 2003

SUN Microsystems. <<http://www.sun.com/>> Disponível on-line, acesso em 22/01/2005.

TSAI T. M.; YU H.K.; LIAO P.Y.; et al. **Semantic Modeling among Web Services Interfaces for Services Integration – SOTA (Smart Office Task Automation) platform**. In: International Workshop on Database and Expert Systems Applications IEEE, 14. p. 579-583, Sept. 2003.

WANG, H; ZHANG, Y.; CAO, J.; et al. **Achieving secure and flexible M-services through tickets**. IEEE Transactions on Systems, Man and Cybernetics, Part A, Volume: 33, Issue: 6, p. 697-708, Nov. 2003.

WOMBACHER, A.; MAHLEKO, B.; **Ad-hoc business processes in web services**. Symposium on Applications and the Internet Workshops, 2003. Proceedings. 2003, 27-31 Jan. 2003 p. 101-105

(W3C, A), **Web Services. Technical report**. The World Wide Web Consortium <<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211>>. Disponível on-line, acesso em 19/02/2004.

(W3C, B), **Web Services Description Language (WSDL). Technical report**. The World Wide Web Consortium <<http://www.w3.org/TR/wsdl>>. Disponível on-line, acesso em 21/06/2004.

(W3C, C), **Extensible Markup Language (XML). Technical report**. The World Wide Web Consortium <<http://www.w3.org/TR/2004/REC-xml-20040204/>>. Disponível on-line, acesso em 15/03/2004.

(W3C, D), **Document Object Model (DOM). Technical report**. The World Wide Web Consortium < <http://www.w3.org/DOM/>>. Disponível on-line, acesso em 11/08/2004.

(W3C, E), **XML Schema. Technical report**. The World Wide Web Consortium <<http://www.w3.org/XML/Schema>>. Disponível on-line, acesso em 10/08/2004.

(W3C, F), **Extensible Stylesheet Language (XSL). Technical report**. The World Wide Web Consortium < <http://www.w3.org/Style/XSL/>>. Disponível on-line, acesso em 11/08/2004.

(W3C, G), **XML Pointer, XML Base and XML Linking. Technical report**. The World Wide Web Consortium <<http://www.w3c.org/XML/Linking>>. Disponível on-line, acesso em 17/08/2004.

(W3C, H), **Voice Extensible Markup Language (VoiceXML). Technical report.** The World Wide Web Consortium <<http://www.w3.org/TR/voicexml20/>>. Disponível on-line, acesso em 19/08/2004.

(W3C, I), **Simple Object Access Protocol (SOAP). Technical report.** The World Wide Web Consortium <http://www.w3c.org/TR/2003/REC-soap12-part0-20030624>. Disponível on-line, acesso em 13/09/2004.

(W3C, J), **XML Information Set. Technical report.** The World Wide Web Consortium < <http://www.w3.org/TR/xml-infoset/>>. Disponível on-line, acesso em 04/01/2005.

XML ORG. **Extensible Markup Language (XML).** Technical report <<http://www.xml.org/xml/xmldev.shtml>>. Disponível on-line, acesso em 10/09/2004.

YANG, X.; BOUGUETTAYA, A.; MEDJAHED, B.; et al. **Organizing and Accessing Web Services on Air**, Systems, Man and Cybernetics, IEEE Transactions, p. 742-757, Nov. 2003.

YUAN, M. J.; LONG, J. **Java Readies Itself for Wireless Web Services.** <<http://www.javaworld.com/javaworld/jw-06-2002/jw-0621-wireless.html>>, 2002. Disponível on-line, acesso em 18/05/2003.

ZHU, C.; LEE, M.J.; SAADAWI, T. A **Border-aware Broadcast Scheme for Wireless Ad Hoc Network.** Consumer Communications and Networking Conference First IEEE, p. 134-139, Jan. 2004.

ZHU, J. **Web Services Provide the Power to Integrate.** Power and Energy Magazine, IEEE, p. 40-49, Nov.-Dec. 2003.