

LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RIO GRANDE DO
NORTE

PEER-TO-PEER SOBRE MANETS COM JXTA

HENRIQUE ANDRÉ BARBOSA BITTENCOURT DUTRA

PEER-TO-PEER SOBRE MANETS COM JXTA

Orientador:

Cleydon de A. F. Lages M. Jr.

HENRIQUE ANDRÉ BARBOSA BITTENCOURT DUTRA

PEER-TO-PEER SOBRE MANETS COM JXTA

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito obrigatório para obtenção do grau de Bacharel em Sistemas de Informação.

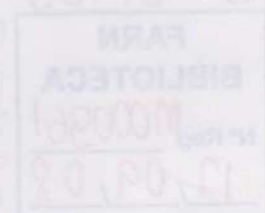
Orientador:

Gleydson de A. F. Lima, M.Sc.

Aluizio Ferreira da Rocha Neto, M.Sc.
Membro Examinador

Edgard de Faria Correa, Dr.
Membro Examinador

NATAL / RN
2007



Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Dutra, Henrique André Barbosa Bittencourt

Peer-to-peer sobre MANETS com JXTA / Henrique André Barbosa Bittencourt

Dutra. – Natal: FARN, 2007.

45f.

Orientador: Gleydson de A. F. Lima (M. Sc.)

Monografia (Bacharelado em Sistemas de Informação) – Faculdade Natalense
para o Desenvolvimento do Rio Grande do Norte.

1. Computação Ubiqua – Monografia. 2. Arquitetura JXTA – Monografia.
3. Mobile Ad-hoc Network – Monografia. I. Título.

RN/FARN/BC

CDU 004.7



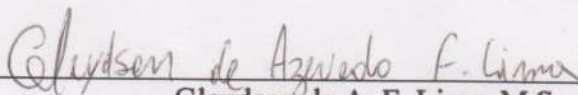
004.7
D975P
2007
ex 01

NATAL/RN
2007

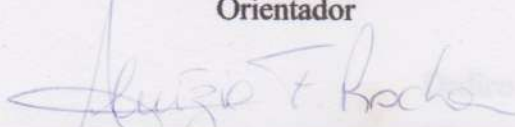
HENRIQUE ANDRÉ BARBOSA BITTENCOURT DUTRA

PEER-TO-PEER SOBRE MANET COM JXTA

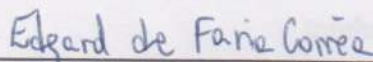
Monografia apresentada ao curso de Bacharelado em Sistemas de Informação da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte e aprovada em ____ de ____ de ____, pela banca examinadora composta pelos seguintes membros:



Gleydson de A. F. Lima, M.Sc.
Orientador



Aluizio Ferreira da Rocha Neto, M.Sc.
Membro Examinador



Edgard de Faria Correa, Dr.
Membro Examinador

RESUMO

Os dispositivos móveis sem fio e as redes *Peer-to-Peer* apresentam um grande crescimento na quantidade de usuário. Existe um ambiente altamente heterogêneo devido a diversidade de aparelhos com diferentes capacidades e grande variedade para comunicação sem fio, e a convergência entre essas tecnologias é um dos desafios para a chamada computação ubíqua, proposta por Mark Weiser. Este trabalho faz um estudo das *Mobile Ad-hoc Networks* (MANET) e as arquiteturas *Peer-to-Peer*, e como elas podem combinar para gerar uma arquitetura que provê a integração entre esses dispositivos em ambientes sem fio heterogêneos utilizando a plataforma JXTA.

Palavras-chave: Computação ubíqua, *Mobile Ad-hoc Network*, *Peer-to-Peer*, JXTA.

Dedico este trabalho ao meu avô, Vicente Dutra, que esteve sempre presente em todas as etapas da minha formação.

Ao meu avô, Heitor Barbosa (in memoriam) que mesmo estando do outro lado da vida certamente nunca deixou de estar ao meu lado.

Aos meus pais pelo apoio e paciência incondicional de sempre.

RESUMO

Os dispositivos móveis sem fio e as redes *Peer-to-Peer* apresentam um grande crescimento na quantidade de usuário. Existe um ambiente altamente heterogêneo devido a diversidade de aparelhos com diferentes capacidades e grande variedade para comunicação sem fio, e a convergência entre essas tecnologias é um dos desafios para a chamada computação ubíqua, proposta por Mark Weiser. Este trabalho faz um estudo das *Mobile Ad-hoc Networks* (MANET) e as arquiteturas *Peer-to-Peer*, e como elas podem combinar para gerar uma arquitetura que provê a integração entre esses dispositivos em ambientes tão heterogêneos utilizando a plataforma JXTA.

Palavras-chave: Computação ubíqua. Mobile Ad-hoc Network. Peer-to-Peer. JXTA.

ABSTRACT

The mobile devices wireless and the Peer-to-Peer network present a great growth in the amount of user. A heterogeneous environment due exists highly the diversity of devices with different capacities and great variety for communication wireless, and the convergence between these technologies is one of the challenges for the named ubiquitous computing, proposal for Mark Weiser. This work makes a study of Mobile Ad-hoc Networks (MANET) and the Peer-to-Peer architectures, and as they can combine to generate an architecture that to provide the integration enters these devices in so heterogeneous environments using platform JXTA.

Keywords: Ubiquitous Computing. Mobile Ad-hoc Network. Peer-to-Peer. JXTA.

Figura 10 Rede virtual JXTA versus Rede física.....	38
Figura 11 - Dispositivos utilizando JXTA para acessar a rede JXTA.....	39
Figura 12 Arquitetura da comunidade JXTA MED.....	40

LISTA DE FIGURAS

1 INTRODUÇÃO	9
Figura 1 - Relação entre computação ubíqua, pervasiva e móvel	15
Figura 2 - Classificação segundo IEEE	16
Figura 3 - Dois tipos de redes ad-hoc: (a) comunicação direta, (b) múltiplos saltos	17
Figura 4 - As camadas JXTA	27
Figura 5 - Mostra o comportamento da conexão de um unicast pipe e de um propagate pipe	31
Figura 6 - Rendezvous super-peers	32
Figura 7 - Mensagem sendo roteado sobre um firewall	33
Figura 8 - Os protocolos JXTA	33
Figura 9 - Peer-to-Peer versus Rede Física	37
Figura 10 Rede virtual JXTA versus Rede física	38
Figura 11 - Dispositivos utilizando JXME para acessar a rede JXTA	39
Figura 12 Arquitetura da comunidade JXTA MED.	40
4.3 ALGORITMOS DE ROTEAMENTO	22
5 PROJETO JXTA	26
5.1 OBJETIVOS	26
5.2 ARQUITETURA DA PLATAFORMA	27
5.3 CONCEITOS	28
5.3.1 Jax Id	28
5.3.2 Pwr	28
5.3.3 Peer Group	28
5.3.4 Pipes	29
5.3.5 Messages	31
5.3.6 Advertisements	31
5.4 ARQUITETURA DA REDE	31
5.4.1 Rendezvous Peer	31
5.4.2 Relay Peer	32
5.5 PROTOCOLOS	33
5.5.1 Peer Discovery Protocol (PDP)	33
5.5.2 Peer Information Protocol (PIP)	34
5.5.3 Peer Resolver Protocol (PRP)	34
5.5.4 Pipe Blinding Protocol (PBP)	34
5.5.5 Endpoint Routing Protocol (ERP)	34
5.5.6 Rendezvous Protocol (RVP)	35
6 INTEGRANDO MANET E JXTA	36
7 CONCLUSÃO	41
8 REFERÊNCIAS	43

1 INTRODUÇÃO	9
1.1 PROBLEMA DE PESQUISA	11
1.2 OBJETIVOS	11
1.2.1 Objetivos Gerais	11
1.2.2 Objetivos Específicos	12
1.3 JUSTIFICATIVA	12
1.4 ORGANIZAÇÃO DO TRABALHO	12
2 COMPUTAÇÃO UBÍQUA	13
3 MOBILE AD-HOC NETWORK	16
4 PEER-TO-PEER	18
4.1 DEFINIÇÃO	18
4.2 VANTAGEM	19
4.3 MODELOS DE ARQUITETURA	20
4.4 RESTRIÇÕES	21
4.5 ALGORITMOS DE ROTEAMENTO	22
5 PROJETO JXTA	26
5.1 OBJETIVOS	26
5.2 ARQUITETURA DA PLATAFORMA	27
5.3 CONCEITOS	28
5.3.1 Jxta id	28
5.3.2 Peer	28
5.3.3 Peer Group	28
5.3.4 Pipes	30
5.3.5 Messages	31
5.3.6 Advertisements	31
5.4 ARQUITETURA DA REDE	31
5.4.1 Rendezvous Peer	31
5.4.2 Relay Peer	32
5.5 PROTOCOLOS	33
5.5.1 Peer Discovery Protocol (PDP)	33
5.5.2 Peer Information Protocol (PIP)	34
5.5.3 Peer Resolver Protocol (PRP)	34
5.5.4 Pipe Binding Protocol (PBP)	34
5.5.5 Endpoint Routing Protocol (ERP)	34
5.5.6 Rendezvous Protocol (RVP)	35
6 INTEGRANDO MANET E JXTA	36
7 CONCLUSÃO	41
REFERÊNCIAS	43

1 INTRODUÇÃO

A massificação da internet criou um amplo potencial de processamento e que na maioria das vezes encontram-se subutilizado, diante deste prisma diversas soluções estão surgindo, outras sendo reutilizadas para tentar extrair e reutilizar dos usuários esses recursos desperdiçados, dentre estas soluções a arquitetura *peer-to-peer* (P2P) tem sido bastante utilizada.

Peer-to-peer é uma tecnologia escalável que provê uma rede virtual (*overlay*) entre os participantes, de tal maneira que a interação entre os membros é realizada de forma descentralizada, ou seja, sem a necessidade de um servidor central para estabelecer ou monitorar conexões entre os participantes. Nessa arquitetura há uma diminuição dos pontos críticos da rede, uma vez que um membro pode sair ou entrar e a topologia da rede se auto-ajusta as novas condições do sistema.

Embora atualmente o modelo predominante no funcionamento da internet seja cliente/servidor, ela foi concebida originalmente como um sistema *peer-to-peer*, por volta dos anos 60. A meta original da ARPANET (*Advanced Research Projects Agency Network*) era compartilhar recursos computacionais ao redor dos Estados Unidos. O desafio deste esforço era integrar diferentes redes, assim como as futuras tecnologias em uma arquitetura de rede comum, onde os novos *hosts* interagissem de igual para igual com os membros já pertencentes à rede (OREILLY, 2007).

Aos poucos, a internet foi perdendo essa característica, por motivos como especialização de funções, necessidade de robustez e alta disponibilidade, problemas de segurança e limitação de recursos computacionais ou capacidade de enlaces de comunicação.

Ainda, no fim dos anos 70 e início dos anos 80, muita pesquisa foi feita sobre o uso de sistemas distribuídos para aplicações específicas. As idéias de autonomia, transparência, execução concorrente, tolerância a falhas e heterogeneidade de plataformas foram organizadas no modelo distribuído e postas em prática, uso que foi consolidado na década seguinte.

Com o amplo investimento no desenvolvimento das tecnologias de rede para dispositivos móveis fez surgir um novo conceito em redes sem fios: a *Wireless Personal Area Network* cujo IEEE (*Institute of Electrical and Electronics Engineers*) define como padrão 802.15. Nessa categoria encaixam-se tecnologias como *bluetooth*, *zigBee*, UWB (*Ultrawideband*).

Seguindo a evolução das tecnologias de redes, a evolução na tecnologia de fabricação de semicondutores, sobretudo ao longo da última década, esta viabilizando o desenvolvimento

da miniaturização dos componentes eletrônicos como memórias e processadores. Esse progresso tem permitido a construção de dispositivos móveis cada vez menores, como PDAs (*Personal Digital Assistant*), *laptops* e celulares. Não só o tamanho, mas também o poder de processamento destes dispositivos vêm aumentando de forma rápida e consistente, permitindo a inclusão de novas funcionalidades além das suas funções básicas como os atuais telefones celulares.

Os avanços nos últimos anos da computação móvel e das tecnologias de rede estão proporcionando uma revolução com o surgimento de uma nova era da internet marcada pela mobilidade, esta nova era vem sendo chamada de *Any Era*, na qual bilhões de usuários demandam acesso a comunicação, informação e entretenimento em qualquer lugar, a qualquer hora, em qualquer dispositivo.

O que se pode observar de todo esse movimento que vem surgindo é que de fato está ocorrendo uma migração da computação tradicional, baseada em *desktops*, para um ambiente onde os dispositivos estarão espalhados ao nosso redor de forma transparente e interligados ao nosso estilo de vida. Os pesquisadores chamaram esse novo ambiente de computação ubíqua.

A idéia desse novo paradigma foi proposta por Weiser (1991), então pesquisador da XEROX. Em seu artigo "The Computer for the 21st Century", publicado em 1991, vislumbrava um novo cenário onde os computadores estariam disponíveis no ambiente, mas invisível ou pelo menos não tão intrusivo e cada usuário possuiria um ambiente computacional capaz de prover recursos em qualquer lugar, a qualquer momento. Esse ambiente seria como uma áurea que nos acompanharia a todo o momento e a qualquer lugar.

Analisando o novo paradigma podemos perceber o surgimento de um ambiente totalmente dinâmico, onde qualquer dispositivo é um cliente e provedor de recursos em potencial. Uma característica importante neste novo paradigma é que o usuário não precisa lembrar ou nem mesmo saber que está interagindo com dispositivos computacionais. Na visão de Weiser, as facilidades computacionais devem ser incorporadas aos ambientes para auxiliar atividades humanas mudando minimamente a forma como as tarefas são originalmente realizadas.

A vertiginosa evolução da computação móvel esta permitindo que as idéias de Weiser (1991) publicadas há mais de uma década tornem-se cada vez mais uma realidade.

Com o crescimento da computação móvel e o surgimento de novas tecnologias tem tornado o ambiente da computação móvel cada vez mais heterogêneo e estes precisam ser integrados, de forma dinâmica, quase sempre transparente e muitas vezes de forma

imprevisível compondo sistemas robustos e persistentes. Mobilidade e adaptabilidade tem sido o grande desafio da computação ubíqua.

A tecnologia JXTA (*Juxtapose*) permite que qualquer dispositivo digital conectado a qualquer rede (*peer*), por si só, descubra outros *peers* e com eles se comunique, criando comunidades conectadas. Qualquer dispositivo, de telefones celulares a estações de trabalho e servidores, pode conversar com qualquer outro e, obviamente, compartilhar arquivos e imagens. As pessoas têm total acesso ao conteúdo dos inúmeros dispositivos, independentemente de local ou rede.

A especificação JXTA foi concebida para ser independente de uma linguagem específica de programação (como por exemplo, C ou Java), plataformas e sistemas operacionais (como Windows e LINUX), e mesmo dos protocolos de rede utilizados (como TCP/IP ou *Bluetooth*).

O JXTA cria comunidades interligadas. Isto permite que os utilizadores tenham um melhor acesso aos seus conteúdos, independentemente da sua localização e retira a complexidade das redes dos sistemas operacionais. A tecnologia JXTA permite um estilo de computação mais social, uma computação Pessoa-a-Pessoa.

1.1 PROBLEMA DE PESQUISA

Com a expansão da computação móvel, temos que estar preparados para um novo ambiente onde existe um dinamismo muito grande e dispositivos heterogêneos. Este novo cenário gera desafios que precisam ser estudados.

- Como permitir a colaboração entre os dispositivos móveis?
- Qual arquitetura distribuída pode ser usada para aproveitar o potencial destes dispositivos?
- Como fornecer mecanismos para que haja uma convergência tecnológica?
- Como aplicações ubíquas podem ser desenvolvidas com as tecnologias que temos atualmente?

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

Estudar um modelo que permita o desenvolvimento de aplicações ubíquas com base nas tecnologias que temos atualmente, permitindo que dispositivos móveis possam ser integrados de maneira ad-hoc a rede existente.

1.2.2 Objetivos Específicos

- Estudar o cenário atual aonde computação móvel e redes sem fio (MANET) vem ganhando destaque
- Estudar arquitetura *peer-to-peer*
- Estudar *Mobile Ad-hoc Network*
- Estudar plataforma JXTA

1.3 JUSTIFICATIVA

Existe um crescimento na quantidade de usuários de dispositivos móveis sem fio (MANET). É preciso encontrar maneiras para utilizar o potencial desses aparelhos. Novas arquiteturas estão sendo propostas e algumas já existentes estão sendo remodeladas para contribuir na colaboração entre tais dispositivos.

Cada dispositivo tem seu próprio sistema operacional, com APIs de desenvolvimento proprietárias e utilizando os mais diversos canais de comunicação aérea, o ambiente gerado de tal situação é extremamente heterogêneo. Existe a necessidade de integrar todos esses dispositivos para que possam interagir e criar comunidades interligadas.

1.4 ORGANIZAÇÃO DO TRABALHO

A organização deste trabalho esta seccionado em sete capítulos, sendo:

Capítulo 1 – Explana sobre o conteúdo deste trabalho de forma a introduzir alguns conceitos e idéias que será abordado no restante da monografia.

Capítulo 2 – Explana sobre o crescimento da computação móvel e a conseqüente necessidade da convergência tecnológica entre os dispositivos móveis. Apresenta a computação ubíqua e as idéias de Mark Weiser.

Capítulo 3 – Aborda a evolução das tecnologias sem-fio e da computação até chegar nas MANETS (Mobile Ad-hoc Network).

Capítulo 4 – Apresenta os conceitos e características da arquitetura *peer-to-peer*.

Capítulo 5 – É mostrado de forma detalhada todo o projeto JXTA.

Capítulo 6 – Apresenta as similaridades entre MANETS e *peer-to-peer*, e como elas podem fornecer uma boa solução para ambientes distribuídos onde a necessidade por integração é primordial.

Capítulo 7 – Explana as conclusões deste trabalho.

2 COMPUTAÇÃO UBÍQUA

A computação tem vivido nos últimos anos uma revolução devido aos dispositivos móveis estarem cada vez mais freqüente na nossa sociedade, passando a fazer a parte inclusive do nosso cotidiano como simples objeto do dia-a-dia. Basta dar uma boa olhada ao redor para se perceber, por exemplo, que grande parte das pessoas carrega consigo um telefone celular. De acordo com Loureiro Filho (2006) colocando isso em números:

[...] só as redes de telefonia móvel baseadas na tecnologia GSM contabilizam 1,67 bilhões de clientes em todo o mundo, espalhados por 213 países. O uso de dispositivos móveis como smart phones e PDAs também tem crescido bastante em alguns setores. Esses últimos podem ser encontrados em soluções de Supply Chain Management, suporte a hospitais, dentre outras.

Vivemos na Era da Sociedade da Informação, um estágio de desenvolvimento social caracterizado pela capacidade de obter e compartilhar qualquer informação, instantaneamente, de qualquer lugar e da maneira mais adequada.

Toda essa popularização da tecnologia e a necessidade da sociedade de estar em constante contato com a informação irá gerar grandes desafios para a Ciência da Computação nos próximos anos: Convergência Tecnológica.

O principal intuito da convergência tecnológica é fornecer ao usuário acesso a suas informações e aplicações em qualquer lugar, de qualquer rede, por qualquer canal de comunicação, através de uma interface homem máquina coerente (única), com qualidade adequada e de forma transparente. Ou seja, tudo que a sociedade de informação atual necessita.

Independente do tipo de dispositivo portátil, a maior parte desses equipamentos deverá ter capacidade de se comunicar com a parte fixa da rede e, possivelmente, com outros computadores móveis.

A convergência das tecnologias de rádio, dos microprocessadores e dos dispositivos eletrônicos digitais pessoais está levando ao conceito de ubiquidade no qual dispositivos inteligentes, móveis e estacionários, coordenam-se entre si para prover aos usuários acesso imediato e universal a novos serviços, de forma transparente, que visam aumentar as capacidades humanas. (ARAÚJO, 2003).

O visionário pesquisador Mark Weiser descreveu este cenário de integração de dispositivos heterogêneos em seu artigo no fim dos anos 80 (WEISER, 1991). Ele vislumbrou uma forma pouco intrusiva de utilizar dispositivos computacionais no cotidiano das pessoas.

Segundo ele facilidades computacionais devem ser incorporadas aos ambientes para auxiliar atividades humanas mudando minimamente a forma como as tarefas são originalmente executadas. (WEISER, 1991). Weiser denominou de computação ubíqua essa forma transparente de integrar tecnologias as atividades humanas diárias. (MAIA, 2006).

Publicado em 1991, o artigo de Weiser prevê que os computadores pessoais desaparecerão do nosso olhar e passarão a fazer parte de todos os objetos, de forma integrada e onipresente, atuando na surdina. Ele faz uma analogia entre os computadores e o desaparecimento dos motores, que encolheram até passarem a fazer parte de pequenos objetos do dia a dia. Sem no entanto se tornarem visível aos olhos.

Essa maior integração com o ambiente, possivelmente mutável, exige que aplicações ubíquas sejam capazes de se ajustar dinamicamente a diferentes condições do ambiente do usuário, tais como dispositivos de hardware (telas ou microfones) ou serviços de software (bancos de dados ou e-mail), ou ainda a qualidade da plataforma de execução como processador, memória ou recursos de rede. Por essa razão, muitas aplicações ubíquas utilizam técnicas de adaptação dinâmica para flexibilizar sua implementação de forma que esta possa ser ajustada dinamicamente a diferentes condições de execução. (MAIA, 2006).

Por ser uma área emergente de pesquisa, termos como computação ubíqua, computação pervasiva e computação móvel têm sido usadas como sinônimos, embora sejam conceitualmente diferentes e empreguem diferentes idéias de organização e gestão de serviços computacionais.

À medida que a área evolui, esses conceitos vão sendo mais bem compreendidos e as suas definições tornam-se mais claras.

Em Araújo (2003) é exposto à distinção de cada um desses termos:

A computação móvel baseia-se no aumento da nossa capacidade de mover fisicamente serviços computacionais conosco, ou seja, o computador torna-se um dispositivo sempre presente que expande a capacidade de um usuário utilizar os serviços que um computador oferece, independentemente de sua localização. Combinada com a capacidade de acesso, a computação móvel tem transformado a computação numa atividade que pode ser carregada para qualquer lugar.

O conceito de *computação pervasiva* implica que o computador está embarcado no ambiente de forma invisível para o usuário. Nesta concepção, o computador tem a capacidade de obter informação do ambiente no qual ele está embarcado e utilizá-la para dinamicamente construir modelos computacionais, ou seja, controlar, configurar e ajustar a aplicação para melhor atender as necessidades do dispositivo ou usuário. O ambiente também pode e deve ser capaz de detectar outros dispositivos que venham a fazer parte dele. Desta interação surge a capacidade de computadores

agirem de forma “inteligente” no ambiente no qual nos movemos, um ambiente povoado por sensores e serviços computacionais.

A computação ubíqua beneficia-se dos avanços da computação móvel e da computação pervasiva. A computação ubíqua surge então da necessidade de se integrar mobilidade com a funcionalidade da computação pervasiva, ou seja, qualquer dispositivo computacional, enquanto em movimento conosco, pode construir, dinamicamente, modelos computacionais dos ambientes nos quais nos movemos e configurar seus serviços dependendo da necessidade.

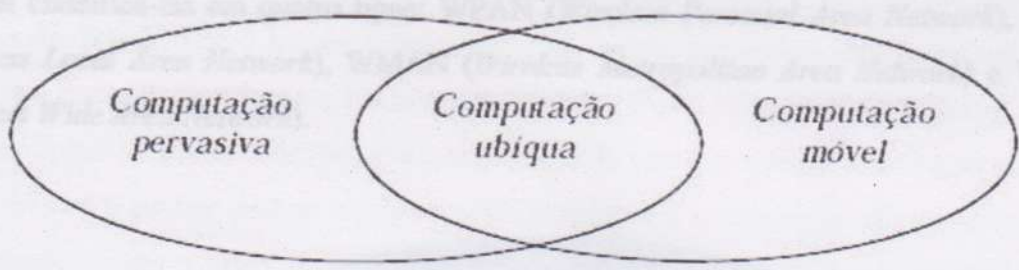


Figura 1 - Relação entre computação ubíqua, pervasiva e móvel
Fonte: Araújo (2003)

Tendo em vista todas as definições mencionadas acima, o termo computação ubíqua será usado aqui como uma junção da computação pervasiva e da computação móvel. A justificativa de se realizar uma diferenciação desses termos é que um dispositivo que está embutido em um ambiente, não necessariamente é móvel. Devido a isso, quando for utilizado o termo computação ubíqua, considerar-se-ão o alto grau de dispositivos embarcados da computação pervasiva juntamente com o alto grau de mobilidade da computação móvel. (ARAÚJO, 2003).

Figura 2 - Classificação segundo IEEE
Fonte: IEEE (2007)

Com a expansão da tecnologia wireless e da computação móvel, houve uma necessidade de um novo tipo de rede, que alcançasse maior mobilidade, capacidade de auto-organização e que fosse rapidamente estabelecida. Esse tipo de rede, que é independente de qualquer estrutura fixa, chama-se rede ad-hoc. Nesse contexto, a rede produzida a mobilidade e a capacidade por comunicação surgem as redes ad-hoc, ou MANET (Mobile Ad-Hoc Networking).

Consequentemente, redes ad-hoc caracterizam-se pela ausência de infraestrutura e pela ausência de capacidade de auto-organização, portanto são produzidas uma topologia dinâmica. Os nós em uma rede ad-hoc se comunicam com um outro nó, criando uma rede ad-hoc, na qual alguns dos dispositivos da rede fazem parte da rede de rede apenas durante o

3 MOBILE AD-HOC NETWORK

Nos últimos anos, acompanhando o desenvolvimento de dispositivos portáteis, tais como celulares, *notebooks* e *palmtops*, presenciamos também um grande crescimento no desenvolvimento de tecnologias e padrões para a comunicação entre tais dispositivos empregando tecnologias de comunicação *wireless* (sem fio).

Existem diferentes tipos de redes sem fio que variam em tecnologia e aplicação, sendo possível classificá-las em quatro tipos: WPAN (*Wireless Personal Area Network*), WLAN (*Wireless Local Area Network*), WMAN (*Wireless Metropolitan Area Network*) e WWAN (*Wireless Wide Area Network*).

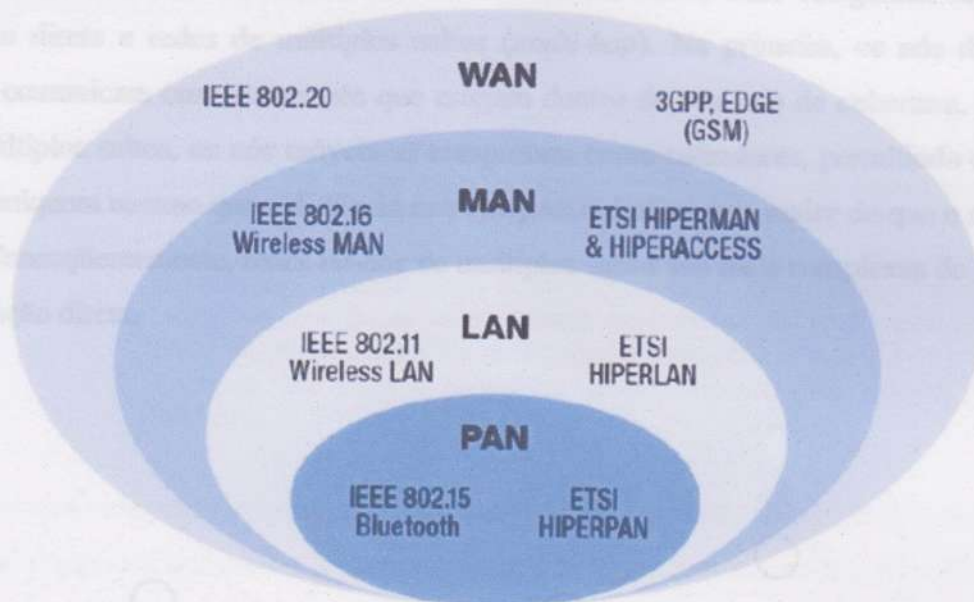


Figura 2 - Classificação segundo IEEE
Fonte: IEEE (2007)

Com a expansão da tecnologia *wireless* e da computação móvel, houve uma necessidade de um novo tipo de rede, que oferecesse maior mobilidade, capacidade de auto-organização e que fosse rapidamente estabelecida. Esse tipo de rede, que é independente de qualquer estrutura fixa, chama-se rede *ad-hoc*. Nesse contexto onde predomina a mobilidade e necessidade por comunicação surgem as redes móveis *ad-hoc*, ou MANET (*Mobile Ad-Hoc Networking*).

Geralmente, redes *ad-hoc* caracterizam-se pela ausência de infra-estrutura e pela conexão espontânea de seus nós, portanto não possuem uma topologia predeterminada. Os nós numa rede *ad-hoc* se comunicam sem conexão física entre eles criando uma rede *on the fly*, na qual alguns dos dispositivos da rede fazem parte da rede de fato apenas durante a

duração da sessão de comunicação, ou, no caso de dispositivos móveis ou portáteis, por enquanto que estão a uma certa proximidade do restante da rede.

Em Cordeiro e Agrawal (2002) define MANET como sendo: “[...] um sistema autônomo de *hosts* móveis (também atuando como roteadores) conectados via enlaces sem fio, cujo conjunto forma uma rede de comunicações modelada por um grafo arbitrário.”

A grande vantagem das redes *ad-hoc* sem fio está em oferecer alta flexibilidade e possuir um alto potencial de servir de estrutura sem fio ubíqua capaz de intercomunicar milhares de dispositivos. Elas são a única solução quando não existe infra-estrutura ou esta possui alto custo de instalação e baixa confiabilidade. Outra vantagem das redes *ad-hoc* é a robustez, já que os nós da rede resistem a catástrofes da natureza e situações de guerra.

As redes *ad-hoc* podem ainda ser subdivididas em outras duas categorias: redes de comunicação direta e redes de múltiplos saltos (*multi-hop*). Na primeira, os nós da rede somente se comunicam com outros nós que estejam dentro do seu raio de cobertura. Já nas redes de múltiplos saltos, os nós móveis se comportam como roteadores, permitindo que os nós se comuniquem mesmo que a distância entre origem e destino seja maior do que o raio de cobertura. Conseqüentemente, redes *ad-hoc* de múltiplos saltos são mais complexas do que as de comunicação direta.

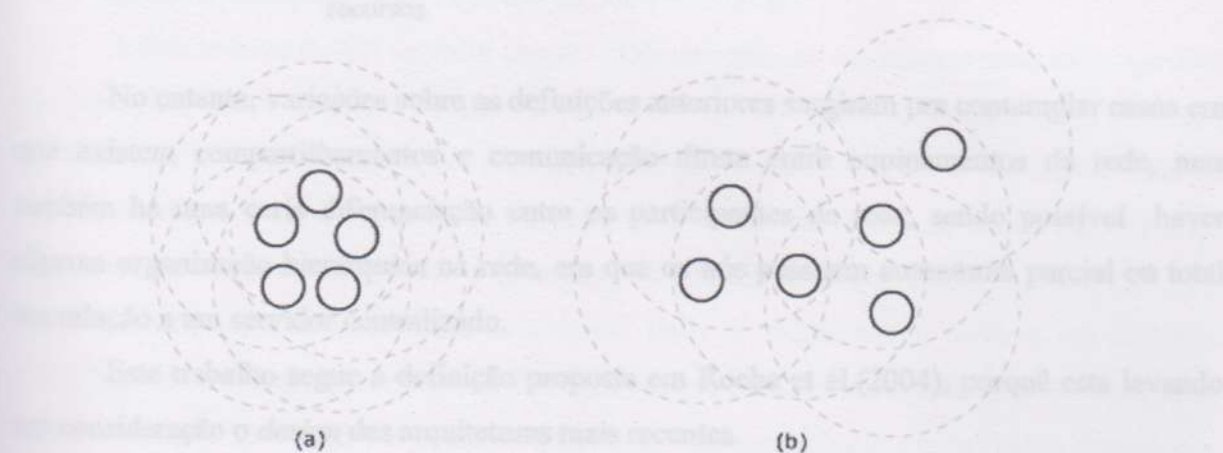


Figura 3 - Dois tipos de redes ad-hoc: (a) comunicação direta, (b) múltiplos saltos

4 PEER-TO-PEER

4.1 DEFINIÇÃO

Há inúmeras definições sobre redes Peer-to-Peer na literatura. É difícil obter consenso sobre definições quando uma tecnologia ou paradigma ainda é recente.

Segundo Stoica et al. (2001) "Sistemas e aplicações *peer-to-peer* são sistemas distribuídos sem qualquer forma de controle centralizado ou hierarquia organizacional, onde o software que está sendo executado em cada nó é equivalente em funcionalidades"

Para Rowstron; Druschel (2001), "Sistemas *peer-to-peer* podem ser caracterizados como sistemas distribuídos nos quais todos os nós possuem capacidades e responsabilidades idênticas e toda comunicação é simétrica".

De acordo com Kellerer (1998):

Uma arquitetura de rede distribuída pode ser chamada *Peer-to-Peer* (P2P) se os participantes compartilharem parte de seus próprios recursos de *hardware* (poder de processamento, capacidade de armazenamento, banda de rede, impressoras) . Esses recursos são necessários para prover os serviços e conteúdo oferecidos pela rede (ex. compartilhamento de arquivos ou espaços de trabalho para colaboração mútua). Os serviços e recursos são acessíveis por todos os pares sem necessidade de passar por nenhuma entidade intermediária. Os participantes dessa rede são tanto provedores de recursos (serviços e conteúdo) como demandadores desses mesmos recursos.

No entanto, variações sobre as definições anteriores surgiram pra contemplar casos em que existem compartilhamentos e comunicação direta entre equipamentos da rede, mas também há uma certa diferenciação entre os participantes da rede, sendo possível haver alguma organização hierárquica na rede, em que os nós possuem autonomia parcial ou total em relação a um servidor centralizado.

Este trabalho segue a definição proposta em Rocha et al.(2004), porquê esta levando em consideração o *design* das arquiteturas mais recentes.

Redes *peer-to-peer* são redes virtuais que funcionam na *Internet* com objetivo de compartilhar recursos entre os participantes, sendo que por principio não há diferenciação entre os participantes. O grupo de pesquisa sobre redes P2P da IRTF as define como o "o compartilhamento de recursos e serviços computacionais diretamente entre os sistemas". (ROCHA et al. 2004)

Para finalizar, Rocha et al. (2004) complementa:

Em geral, é aceito pela comunidade que sistemas P2P devem suportar os seguintes requisitos:

- Nós podem estar localizados nas bordas da rede;
- Nós com conectividade variável ou temporárias e endereços também variáveis;
- A capacidade de lidar com diferentes taxas de transmissão entre nós;
- Nós com autonomia parcial ou total em relação a um servidor centralizado;
- Assegurar que os nós possuem capacidades iguais de fornecer e construir recursos de seus *peers*;
- A rede deve ser escalável;

Tendo todas estas características citadas por Rocha *et al*, uma rede pode ser dita peer-to-peer, mesmo que algumas das funções de controle da rede estejam localizados em um servidor central (ponto de falha).

4.2 VANTAGEM

Existem diversos modelos, cada qual com suas vantagens e desvantagens, mas de maneira geral o que se busca nos sistemas peer-to-peer é pequena ou nenhuma centralização para aumentar a escalabilidade e diminuir os pontos de falhas.

A inexistência de um servidor central (com exceção do modelo centralizado) significa que é possível cooperar para formação de uma rede P2P sem qualquer investimento adicional em hardware de alto desempenho para coordená-la. Outra vantagem é a possibilidade de agregar e utilizar a capacidade de processamento e armazenamento que ficam subutilizadas em máquinas ociosas. Além disso, a natureza descentralizada e distribuída dos sistemas P2P puros torna-os inerentemente robustos a certos tipos de falhas muito comuns em sistemas cliente/servidor. Finalmente, o modelo P2P apresenta o benefício da escalabilidade, para tratar de crescimentos incontáveis no número de usuários e equipamentos conectados, capacidade de rede, aplicações e capacidade de processamento.

Podemos destacar como sendo relevantes as seguintes características que tornam a arquitetura P2P tão atrativa para o desenvolvimento de sistemas distribuídos:

- **Escalabilidade.** Não há gargalos para o crescimento em relação a capacidade do servidor. Em um sistema cliente/servidor, os servidores são os únicos responsáveis por toda a carga do sistema. Em determinados horários, estes servidores podem ficar sobrecarregados e o sistema como um todo tende a oferecer um serviço de baixa

qualidade. Em um sistema P2P, quando o número de clientes na rede aumenta, cresce também o número de servidores, aumentando na mesma proporção a quantidade de recursos compartilhados.

- **Robustez.** Não há um ponto único de falha. Em sistema cliente/servidor, se ocorrer a impossibilidade de acessar um servidor, todos os clientes necessitarão aguardar o restabelecimento do servidor, para então continuar com suas atividades. Em um sistema P2P, um outro *peer* poderá disponibilizar os mesmos serviços do *peer* que está fora da rede.
- **Flexibilidade.** Pode ser feita uma auto-configuração ou configuração dinâmica. Um exemplo é a rede automaticamente identificar um novo *peer* e realizar a redistribuição do conteúdo da rede.

4.3 MODELOS DE ARQUITETURA

Existem diferentes categorizações dos modelos de arquitetura P2P indicados na literatura.

Os sistemas *peer-to-peer* (P2P) podem ser implementados de maneira parcialmente centralizados ou totalmente descentralizados. (SCHOLLMEIER, 2002)

Consolidando este conceito, Schollmeier (2002) afirma:

Uma arquitetura de rede distribuída pode ser classificada como P2P “pura”, se em primeiro lugar for P2P [...] e em segundo lugar se qualquer entidade terminal da rede puder ser individualmente removida sem que a rede sofra qualquer perda em termos de serviço.

Uma arquitetura de rede distribuída deve ser classificada como P2P “híbrida” se em primeiro lugar for P2P [...] e em segundo lugar tiver a necessidade de uma entidade central para prover parte dos serviços da rede.

Nos sistemas P2P puros ou sistemas descentralizados, “ao invés de um índice central em um servidor, esses sistemas utilizam técnicas de buscas descentralizadas, onde uma pesquisa é propagada entre os *peers* e qualquer *peer* que possuir um arquivo que corresponda a pesquisa responde diretamente ao *peer* requerente”. (PANISSON, 2003)

Nessa rede [P2P Puro] não há um ponto central e cada nó tem o mesmo nível. Neste modelo todos os nós são autônomos e todos os nós são responsáveis por troca de recursos e por controle (gerenciamento) de recursos. Os nós podem comunicar-se de maneira direta ou através de vizinhos comuns (melhor escala de informações de controle, pior escala de tráfego de rede). (ROCHA et al, 2004).

Um sistema P2P híbrido centralizado é comumente chamado de sistema centralizado. Ele contém alguns elementos centralizados, que são nós com um papel diferenciado dos demais. Estes sistemas podem ainda ser subdivididos em totalmente centralizados ou hierárquicos. Nos sistemas híbridos totalmente centralizados possuem um nó central com todas as informações (status *online-offline* dos *peers*, conteúdos compartilhados, entre outros) de todos os *peers* da rede, um exemplo deste sistema é o Napster (NAPSTER, 2007). Os sistemas híbridos hierárquicos possuem nós especiais (denominados super-nós, *super-peers* ou refletoras).

Lv *et al.* (2002) propõem uma nova classificação para redes P2P e que esta sendo utilizada principalmente no mundo acadêmico, sendo assim este trabalho também utiliza esta definição.

Consiste em dividir a arquitetura em:

- **Centralizada.** A rede que mantém um índice central com informações atualizadas. Sistemas de compartilhamentos de arquivos como Napster (NAPSTER, 2007) e troca de mensagens utilizam esta arquitetura.
- **Descentralizada mas Estruturada.** Rede que não possui um servidor centralizado de diretório de informações, mas que tem uma estruturação significativa entre os nós. A topologia da rede é controlada e os documentos/arquivos são posicionados em locais que posteriormente tornam fácil sua localização. Esse tipo de arquitetura em geral utilizada busca baseada em *Dynamic Hash Tables* (DHT). Essa arquitetura é utilizada pelos sistemas Chord (CHORD, 2001), Pastry (ROWSTRON; DRUSCHEL, 2001), Tapestry (ZHAO; KUBIATOWICZ; JOSEPH, 2001) e CAN (RATNASAMY, 2001).
- **Descentralizada e Não Estruturada.** Rede que não possui servidor centralizado, nem controle preciso sobre a topologia e localização/busca de documentos. Exemplos da utilização desta arquitetura são as redes Gnutella (GNUTELLA, 2007) e KaZaA (KAZAA, 2007).

4.4 RESTRIÇÕES

Aplicações inerentemente P2P necessitam de conexões diretas entre os *peers*. Entretanto um problema preponderante a ser tratado no projeto desses sistemas são as barreiras de proteção dispostas nas redes, que impedem a comunicação direta entre os *peers*. Os tipos de proteção mais utilizados são os *firewalls*, servidores *proxy* e NAT (*Network Address Translation*).

Firewall é utilizado para filtragem de protocolos e portas específicas geralmente, permitem o fluxo de mensagens do protocolo HTTP (*HyperText Transfer Protocol*). Um *firewall* normalmente bloqueia conexões TCP (*Transmission Control Protocol*) provenientes de computadores externos, impedindo então que os *peers* externos se conectem ao computador protegido.

Um servidor *proxy* é um tipo de servidor que atua nas requisições dos seus clientes executando os pedidos de conexão a outros servidores. Um cliente conecta-se a um servidor *proxy*, requisitando algum serviço tal como, servidor de arquivos, *web*, ou outro recurso disponível em um servidor diferente.

Os dispositivos sob NAT, por sua vez, não possuem um endereço IP (*Internet Protocol*) globalmente roteável, ou seja, o *host* não é encontrado fora do domínio, sendo denominado IP inválido ou privado. Estes *hosts* ficam em redes denominadas privadas e, para cada rede dessas, há apenas um endereço público que é compartilhado pelos outros *hosts* contidos na rede, ou seja, o NAT fornece atribuição de endereço dinâmica aos dispositivos dentro do NAT, mas esses endereços não são endereços válidos publicamente na Internet. Qualquer *host*, dentro de um NAT, pode fazer conexões para fora (“*outgoing*”), mas conexões de fora para NAT (conexões “*incoming*”) apenas são aceitas, se a conexão estiver estabelecida entre os *hosts*. (OLIVEIRA, 2005)

Uma solução para transpor essas barreiras pode ser implementada utilizando o protocolo HTTP: *HTTP Tunneling*. Esse protocolo permite que um *web browser* inicie um pedido em um servidor *web* através da porta de comunicação 80. Isso efetivamente não se traduz em ameaça, uma vez que aplicações externas não podem iniciar uma conexão com o *web browser*. Por isso, administradores de rede normalmente permitem que essa porta não seja barrada pelo *firewall*. Essa solução também é aplicada para redes que utilizam servidores *proxy* e NAT (ROCHA et al, 2002)

4.5 ALGORITMOS DE ROTEAMENTO

Localizar informações numa rede volátil, de grande escala e altamente distribuída não é tarefa simples. A parte difícil é como encontrar o *peer*, de onde trazer a informação, e fazer isso de forma escalável. Para realizar este trabalho existem algoritmos de busca e roteamento que tentam otimizar o encaminhamento de uma mensagem de um *peer* a outro. Os três modelos mais comuns são: centralizado, inundação e *Dynamic Hash Tables* (DHT). O modelo de inundação é também classificado como modelo “descentralizado não-estruturado”,

enquanto que o modelo DHT é chamado de modelo “descentralizado estruturado”. Isso se deve ao mecanismo (estruturado ou não) de entrada de um nó na rede de busca.

No modelo centralizado a principal característica é conexão dos *peers* a um servidor central onde eles publicam seus recursos disponíveis para compartilhamento com o restante dos membros da rede. No servidor central é guardado apenas um índice contendo os recursos disponíveis a comunidade de usuários. Para localizar ou fazer *download* de um determinado recurso, o *peer* faz uma pesquisa no índice do servidor central e este retorna os identificadores de todos os *peers* que possuem o recurso procurado. Então é estabelecida uma conexão direta entre o *peer* que possui o recurso e o *peer* requerente. Este modelo requer uma infra-estrutura de gerenciamento (o servidor central), que armazena informações sobre todos os participantes da comunidade. Tal aspecto pode gerar limites de escalabilidade ao modelo, uma vez que requer servidores maiores quando o número de requisições aumenta, e mais espaço para armazenamento à medida que a quantidade de usuários cresce. Contudo, a experiência do Napster (NAPSTER, 2007) mostrou que, exceto por questões legais, esse modelo era bastante robusto e eficiente. (ROCHA et al, 2002; PANISSON, 2003)

Modelo de inundação surgiu como uma resposta ao modelo centralizado, pois essas soluções são altamente vulneráveis devido ao seu ponto de falha além de ser difícil escalabilidade quando se trata de sistemas da ordem de milhões de usuários. Esses problemas levaram a adoção de soluções descentralizadas. Essas soluções envolvem tipicamente a organização dos *peers* em uma *overlay network*, sobre a qual as requisições de procura são passadas de *peer* para *peer*. Funcionando dessa maneira, os sistemas atuais criam novos e diferentes problemas de escalabilidade. Por exemplo, no Gnutella (GNUTELLA, 2007), as mensagens de busca sofrem um processo de *flooding*, ou seja, ocorre um processo de inundação de mensagens dentro da rede. O processo de *flooding* em cada requisição certamente não é escalável — o número de mensagens circulando na rede cresce exponencialmente. Há também o fato de o processo de *flooding* ter de ser cortado em um determinado ponto, o que determina seu escopo, podendo assim falhar em encontrar o arquivo procurado em alguns sistemas grandes. (PANISSON, 2003)

Jovanovic (2001) em seu estudo sobre a rede Gnutella (GNUTELLA, 2007) observou que além do *flooding* gerado pelo excesso de mensagens ocorre também um fenômeno que ele chamou de “curto circuito”. Esse efeito pode ser descrito pela análise do comportamento de uma rede funcionando com algoritmo de *flooding* que tenha características de heterogeneidade nos atrasos de transmissão, como é o caso da internet.

O *flooding* nos sistemas P2P atuais funciona através de controle de *loops* e *time-to-live* (TTL). Para o controle de *loops*, cada *peer* guarda em *cache* os identificadores das mensagens recebidas recentemente. Se o *peer* receber uma mensagem que foi recentemente enviada a ele, a mensagem é automaticamente descartada. Isso evita que uma mensagem ocupe desnecessariamente recursos de rede. Para controle via TTL, cada mensagem enviada possui um campo denominado *time-to-live*, que é decrementado a cada vez que a mensagem é reenviada. Quando esse campo chegar à zero, a mensagem é descartada. (PANISSON, 2003)

Teoricamente, se o campo TTL de cada mensagem for atribuído o diâmetro da rede, todos os *peers* da rede receberão uma cópia da mensagem. Na prática, porém, quando a rede possui diferentes atrasos de transmissão entre os pontos, a mensagem pode não chegar a uma quantidade significativa de pontos.

O problema se agrava em redes de larga escala onde segundo Jovanovic (2001) a redução no número de *peers* alcançados chega a uma média de 55%, uma parte considerável dos *peers* na rede.

Comunidade entusiasta de aplicações peer-to-peer, tentando aprimorar a escalabilidade projetou um novo modelo de localização de recursos, O DHT (*Dynamic Hash Tables*), o resultado é um sistema altamente estruturado. Nesses sistemas a topologia da rede é controlada e os recursos são colocados em lugares específicos. Esses sistemas estruturados fornecem um mapeamento entre os identificadores de arquivos e sua localização, de forma que as buscas sejam eficientemente roteadas até os *peers* com os arquivos desejados.

O modelo de tabelas de *hash* distribuídas (DHT) é o mais recente. Nesse modelo, um ID randômico é associado a cada *peer* da rede que conhecem um determinado número de *peers*. Quando um documento é publicado (compartilhado) em tal sistema, um ID é associado ao documento baseado em uma *hash* de conteúdo dos documentos e no seu nome. Cada *peer* então encaminha o documento ao *peer* cujo ID do *peer* atual seja o mais próximo do ID do documento. Esse processo é repetido até que o ID do *peer* atual seja o mais próximo do ID do documento. Cada operação de roteamento também garante uma cópia local do documento seja mantida. Quando um *peer* solicita o documento de um sistema P2P, a requisição irá até o *peer* com ID mais semelhante ao ID do documento. Esse processo continua até que uma cópia do documento seja encontrada. Então o documento é transferido ao *peer* que originou a requisição, enquanto cada *peer* que participou do roteamento permanecerá com uma cópia local do documento. (ROCHA et al. 2002)

Esses sistemas, desta forma, oferecem uma solução realmente escalável para pesquisas de buscas. Porém é preciso considerar que estas buscas apenas poderão ser roteadas ao *peer*

responsável se for possível fazer uma busca exata, em que o identificador completo do objeto especificado é conhecido. Em outras palavras, é preciso conhecer o ID do documento antes de a pesquisa ser realizada, tornando a busca bem mais complexa que no modelo por inundação.

Essas buscas são chamadas de *exact match query*, que diferem consideravelmente das pesquisas feitas por aproximação, bastante comuns em sistemas de bancos de dados modernos. (PANISSON, 2003)

Segundo Rocha et al. (2002) existe um outro problema, "pode ocorrer a formação de "ilhas", onde a comunidade divide em sub-comunidades que não possuem nenhuma ligação (*links*) entre si".

Os principais algoritmos que implementam DHT são: Chord (CHORD, 2001), Pastry (ROWSTRON;DRUSCHEL, 2001), Tapestry (ZHAO;KUBIATOWICZ;JOSEPH, 2001) e CAN (RATNASAMY, 2001).

Esta situação lembra o que acontecia na Internet na era "pre-browser" onde ter acesso à Internet significava pertencer à comunidade de AOL (AOL, 2007). Prodigy ou CompuServ (COMPUSEV, 2007). Cada um desses provedores lançava um software que permitia navegação somente dentro de suas redes. O resultado era que o usuário estava fechado dentro de sua comunidade, e os serviços oferecidos os conteúdos eram específicos. "Projeto IXIA quer trazer ao "mundo P2P" o que o HTTP e os browsers trouxeram para Internet" (GONG, 2001).

• Independência da plataforma

Muitos sistemas P2P oferecem funcionalidades ou serviços através de um conjunto de APIs (Application Programming Interface) desenvolvidas sobre um sistema operacional particular usando protocolos de rede específicos, se o desenvolvedor quiser oferecer o mesmo serviço para duas comunidades distintas, ele vai ter que desenvolver o mesmo serviço duas vezes para duas plataformas P2P (GONG, 2001). Isto torna ineficiente e impraticável quando se usa diversas plataformas P2P existentes. A tecnologia IXIA é projetada para ser aplicada em qualquer sistema P2P existente, independente da linguagem de programação, independente do protocolo e do protocolo de rede.

5 PROJETO JXTA

5.1 OBJETIVOS

O projeto JXTA (*Juxtapose*) foi inicialmente concebido pela Sun Microsystems e projetado com a participação de um pequeno número de especialistas de instituições de ensino e da comunidade *open source*. O projeto definiu um conjunto de objetivos baseados no que foi percebido como falha em muitos sistemas *peer-to-peer*, existentes ou em desenvolvimento.

Atualmente o projeto define três objetivos principais: interoperabilidade, independência de plataforma e ubiquidade.

- **Interoperabilidade**

Muitos sistemas P2P são construídos sobre um único tipo de serviço. Por exemplo, o Napster (NAPSTER, 2007) fornecia compartilhamento de arquivos de música, Gnutella (GNUTELLA, 2007) fornece um compartilhamento para arquivos genéricos e AIM (AIM, 07) fornece um serviço de mensagens instantâneas. Dado as diversas características destes serviços e a falta de uma infra-estrutura P2P comum, os desenvolvedores criam suas próprias arquiteturas e sua própria comunidade de usuários, tornando-as incompatíveis com as já existentes (GONG, 2001).

Esta situação lembra o que acontecia na Internet na era “pre-browser” onde ter acesso a internet significava pertencer a comunidade da AOL (AOL, 2007), Prodigy ou CompuServ (COMPUSERV, 2007). Cada um desses provedores fornecia um *software* que permitia navegação somente dentro de suas redes. O resultado era que o usuário estava fechado dentro de sua comunidade, e os serviços fornecidos ou conteúdos eram específicos. “Projeto JXTA quer trazer ao “mundo P2P” o que o HTTP e os *browsers* trouxeram para Internet” (GONG, 2001).

- **Independência de plataforma**

Muitos sistemas P2P oferecem funcionalidades ou serviços através de um conjunto de APIs (*Application Programming Interface*) desenvolvida sobre um sistema operacional particular usando protocolos de rede específicos, se o desenvolvedor quiser oferecer o mesmo serviço para duas comunidades distintas, ele vai ter que desenvolver o mesmo serviço duas vezes para duas plataformas P2P (GONG, 2001). Isto torna ineficiente e impraticável quando se tem dezenas plataformas P2P existentes. A tecnologia JXTA é projetada para ser acolhida por todos os desenvolvedores, independente da linguagem de programação, independente de aplicação e de protocolos de rede.

- **Computação ubíqua**

JXTA foi projetado para ser implementado em qualquer dispositivo digital incluindo sensores, eletro eletrônicos, PDAs, roteadores, computadores de mesa, servidores e clusters.

A ubiqüidade é uma consequência de todas as características anteriores, a interoperabilidade e a independência de plataforma possibilita que JXTA execute nos mais variados dispositivos e se comunique através de redes heterogêneas.

5.2 ARQUITETURA DA PLATAFORMA

O projeto JXTA está criando um sistema peer-to-peer identificando um pequeno conjunto de funções básicas necessárias para atender aplicações P2P e fornecendo-as como funções de alto nível.

O projeto pode ser logicamente dividido em três camadas:

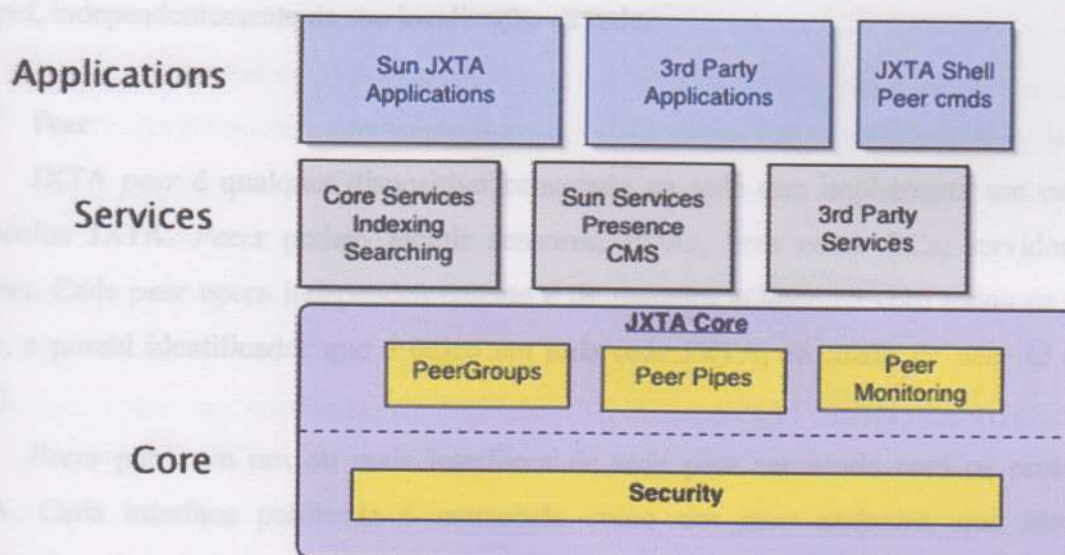


Figura 4 - As camadas JXTA

Fonte: SUN (2005)

Plataforma

Também chamada de core, essa camada encapsula as mínimas e essenciais primitivas que são comuns em redes P2P, incluindo *peers*, *peer groups*, comunicação, monitoramento e primitivas associadas à segurança.

Serviços

Essa camada inclui os serviços que não são absolutamente necessários para fazer uma rede P2P operar, mas são comuns ou desejáveis.

Aplicação

Nesta camada inclui as implementações que dão origem aos aplicativos como, por exemplo, *chats*, aplicações para entretenimento e muitos outros. A separação entre aplicação e serviços nem sempre é óbvia. Uma aplicação para um usuário pode ser entendida como um serviço para outro usuário.

5.3 CONCEITOS

5.3.1 Jxta id

Peers, *peer groups*, *pipes* e outros recursos do JXTA precisam possuir um identificador único em toda a rede. Um JXTA ID identifica unicamente uma entidade dentro da rede e provê a maneira mais simples e significativa de referenciar estas entidades.

URNs (*Uniform Resource Names*) são usados para representar JXTA ID. URNs foram concebidos para fornecer identificadores unívocos e persistentes para recursos publicados na internet, independentemente da sua localização na rede.

5.3.2 Peer

JXTA *peer* é qualquer dispositivo conectado na rede que implementa um ou mais protocolos JXTA. *Peers* podem incluir sensores, PDAs, bem como PCs, servidores ou clusters. Cada *peer* opera independentemente e de maneira assíncrona com todos os outros *peers*, e possui identificador que é único em toda rede JXTA, chamado de *peer id* (SUN, 2005).

Peers publicam um ou mais interfaces de rede para ser usada com os protocolos JXTA. Cada interface publicada é anunciada como um *peer endpoint*, que identifica unicamente a interface de rede. *Peer endpoints* são usados pelos *peers* para estabelecer uma conexão direta ponto-a-ponto entre dois nós, ou seja, correspondem as interfaces de redes usadas para enviar e receber dados (SUN, 2005).

Não é um requisito ter uma conexão ponto-a-ponto entre *peers* que querem comunicar-se. *Peers* intermediários podem ser usados para rotear mensagens para outros *peers* que estão separados, seja em redes físicas distintas ou configuração da rede (por exemplo, um NAT, *firewall* ou *proxies*) (SUN, 2005).

5.3.3 Peer Group

Um *peer group* é uma coleção de *peers* que concordam com um conjunto comum de regras. *Peers* se auto-organizam dentro de *peer groups*, identificado por um *peer group id* que

é único. Cada grupo pode estabelecer sua política de acesso, desde aberto (qualquer *peer* pode fazer parte) até altamente seguro e protegido (requer credenciais para fazer parte).

Peers podem pertencer a mais de um grupo simultaneamente e criar quantos grupos acharem necessários. Todos os *peers* fazem parte do grupo *Net Peer Group*, por padrão.

Os protocolos JXTA descrevem como os *peers* publicam, descobrem, entram e monitoram *peers groups*, mas eles não dizem quando ou por que grupos são criados, isto fica a cargo do desenvolvedor. (SUN, 2005).

Há diversos motivos para se criarem grupos:

- **Para criar um ambiente seguro.** Grupos criam um domínio local de controle em que uma política específica de segurança pode ser reforçada. A política de segurança pode ser simples como um texto plano nome/senha, ou sofisticado com chaves públicas de criptografia. A segurança de acesso permite um *peer* acessar e publicar conteúdos protegidos. *Peer groups* formam regiões lógicas que limitam o acesso aos recursos do grupo.
- **Para criar um ambiente com escopo.** Grupos permitem o estabelecimento de uma especialização no domínio local. Por exemplo, *peers* podem agrupar-se para fornecer um compartilhamento de documento ou compartilhar o uso dos seus processadores. Grupos servem para abstrair a rede em unidades lógicas fornecendo um escopo, ou seja, grupos definem um limite. Por exemplo, quando for realizada uma busca ela fica limitada somente ao escopo do grupo.
- **Para criar um ambiente monitorado.** Permite que *peers* monitorem outros *peers* para qualquer finalidade (monitorar tráfego, responsabilidades).

JXTA define em sua plataforma (*core*) o *peer group services*, que é um conjunto de serviços oferecido por um *peer group*. Para que dois *peers* possam interagir através de um serviço, eles necessariamente devem fazer parte do mesmo grupo.

Peer group services incluem os seguintes serviços:

- **Discovery Service.** Este serviço é usado pelos membros do grupo para procurar por recursos, tais como *peers*, *peer group*, *pipes* e serviços.
- **Membership Service.** É usado pelos membros para rejeitar ou aceitar novos *peers*. *Peers* que desejam entrar em um *peer group* primeiro tem que localizar um membro atual, e então enviar um pedido para entrar no grupo. O *peer* que requisitou a entrada pode ser rejeitado ou aceito pelos membros que formam uma coletividade. Este

serviço pode forçar os *peers* a votarem ou elegerem um representante dentro grupo para aceitar ou rejeitar novos pedidos.

- **Access Service.** É usada para validar pedidos (*request*) feitos de um *peer* para outro. O *peer* que recebeu o pedido fornece as credenciais do *peer* requisitante e informações sobre o pedido para determinar se o acesso é permitido. Em outras palavras, é uma camada de segurança para controlar o acesso a serviços e recursos dentro de um *peer group*.
- **Pipe Service.** É usado para criar e gerenciar *pipes* entre os membros do *peer group*.
- **Resolver Service.** É usada para enviar consulta (*query*) genérica para outros *peers*. *Peers* podem definir e trocar *queries* para localizar qualquer informação que seja necessária (exemplo, *status* de um serviço ou o estado de uma *pipe endpoint*).
- **Monitoring Service.** Este serviço é usado para permitir que um *peer* monitore outros membros do mesmo grupo.

Nem todos os serviços acima devem ser implementados. Um grupo é livre para implementar somente os serviços que forem úteis.

5.3.4 Pipes

Pipes são canais de comunicação virtuais usados para envio e recebimento de mensagens entre *peers*. *Pipes* não estão associados fisicamente com a localização/endereço de um *peer* e são identificados por um *pipe id*.

Pipes também possibilitam uma comunicação assíncrona, definindo *pipes* de entrada e saída. Por não estarem associados com o endereço físico de um *peer*, aplicações e serviços que se comunicam através de *pipes* não são afetados pela mudança de localização dos *peers* e eventuais mudanças de rota que possam acontecer durante a transmissão de dados.

Mensagens são enviadas pelos *pipes* e recebidas pelos ouvintes registrados neste *pipe*. *Pipes* podem restringir o conteúdo a ser trafegado e prover comunicação segura entre *endpoints* (TRAVERSAT, 2003; SUN, 2005).

5.4.1 Rendezvous Peer

Rendezvous Peers são *peers* especiais que indexam informações sobre outros *peers* que ele tem conhecimento. Sendo assim, este *peer* pode ajudar na busca por um *peer* específico ou no roteamento de mensagens propagadas na rede. Caso não haja nenhuma informação disponível em seu índice, a requisição é passada adiante para outro *rendezvous peer*, e assim por diante. *Rendezvous peers* são de fundamental importância na rede para facilitar o tráfego de mensagens enviadas e para otimizar o processo de buscas de recursos.

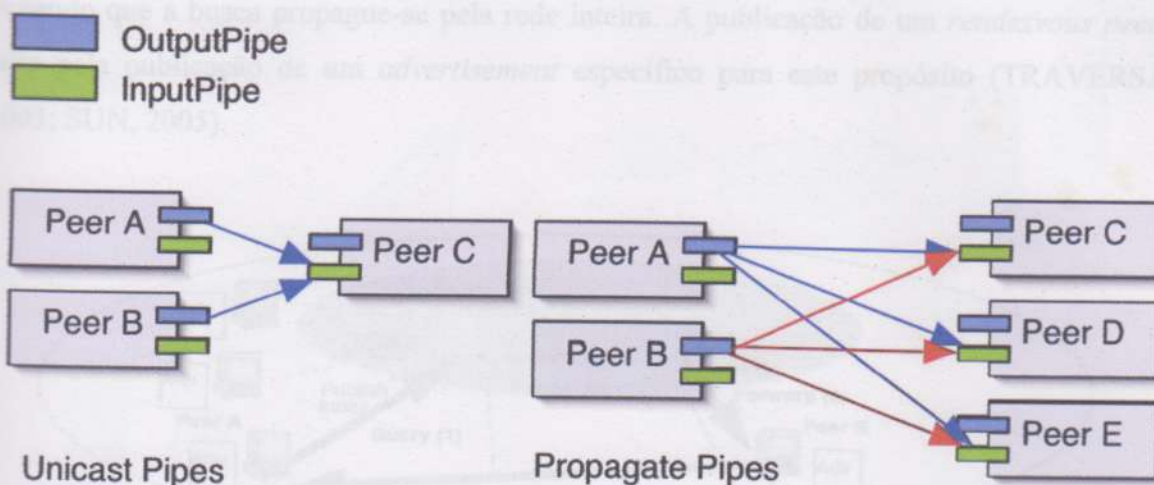


Figura 5 - Mostra o comportamento da conexão de um unicast pipe e de um propagate pipe
Fonte: SUN (2005)

5.3.5 Messages

Toda a rede se comunica por meio de JXTA *Messages*. Elas são a unidade básica de comunicação entre *peers*. Cada protocolo JXTA define um conjunto de *messages* que será usado pelo peer se implementar o protocolo. Elas podem ser enviada de diversas maneiras seja através de *endpoint service*, *pipe service*, *JxtaSocket* ou outra maneira.

5.4.2 Relay Peer

5.3.6 Advertisements

Todas as entidades participantes da rede P2P são representadas por *advertisements*. *Peers* publicam, isto é, disponibilizam dados na rede, mantêm um *cache* e trocam *advertisements* para descobrir novos recursos na rede. Os *advertisements* representam uma abstração de um determinado recurso disponível em uma rede JXTA. Toda e qualquer publicação e descoberta de objetos participantes na rede JXTA é feita através de publicações de *advertisements* específicos (TRAVERSAT, 2003; SUN, 2005).

5.4 ARQUITETURA DA REDE

5.4.1 Rendezvous Peer

Rendezvous Peers são *peers* especiais que indexam informações sobre outros *peers* que ele tem conhecimento. Sendo assim, este *peer* pode ajudar na busca por um *peer* específico ou no roteamento de mensagens propagadas na rede. Caso não haja nenhuma informação disponível em seu índice, a requisição é passada adiante para outro *rendezvous peers*, e assim por diante. *Rendezvous peers* são de fundamental importância na rede para diminuir o tráfego de mensagens enviadas e para otimizar o processo de buscas de recursos,

evitando que a busca propague-se pela rede inteira. A publicação de um *rendezvous peers* é feita pela publicação de um *advertisement* específico para este propósito (TRAVERSAT, 2003; SUN, 2005).

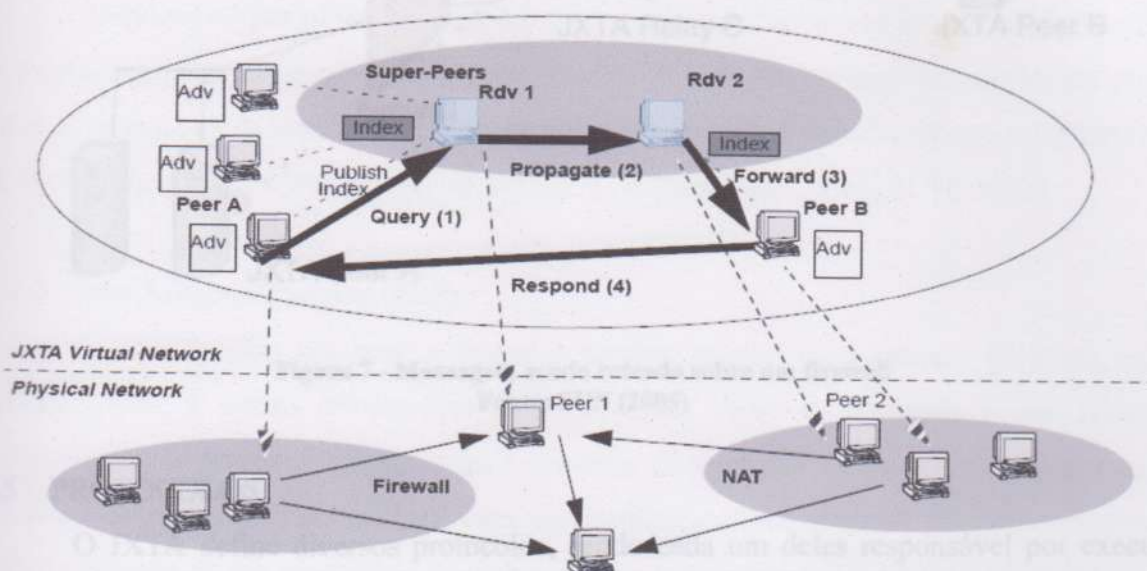


Figura 6 - Rendezvous super-peers
Fonte: TRAVERSAT (2003)

5.4.2 Relay Peer

Muito dos *peers* participantes da rede JXTA possuem conexões temporárias, e frequentemente tem o acesso direto à internet restrita por um firewall ou por uma conexão a ser realizada através de um NAT. Para conseguir acesso aos recursos da rede JXTA, *Relay Peers* podem efetuar o roteamento de mensagens entre *peers*, possibilitando que ambos se comuniquem. *Relay peers* ajudam o roteamento de mensagens entre *peers*, indicando os passos necessários (*hops*) para que uma determinada mensagem atinja seu destino. Qualquer *peer*, desde que tenha as devidas permissões em seu grupo, pode torna-se um *Relay Peer*. A publicação de *Relay Peers* é feita pela publicação de um *advertisement* específico para esta finalidade (TRAVERSAT, 2003; SUN, 2005).

5.5.1 Peer Discovery Protocol (PDP)

Este protocolo habilita um *peer* a procurar *advertisements* do tipo *peer*, pelo grupo, pelo e conteúdo, além de enviar os *advertisements* publicados pelo próprio *peer* para a rede. Este é o principal canal de comunicação do *peer* com a rede. A descoberta de novos *peers* na rede pode ser feita explicitamente, especificando o ID do *peer* ou do *peer group* desejado, neste último caso retornando todos os *peers* pertencentes ao

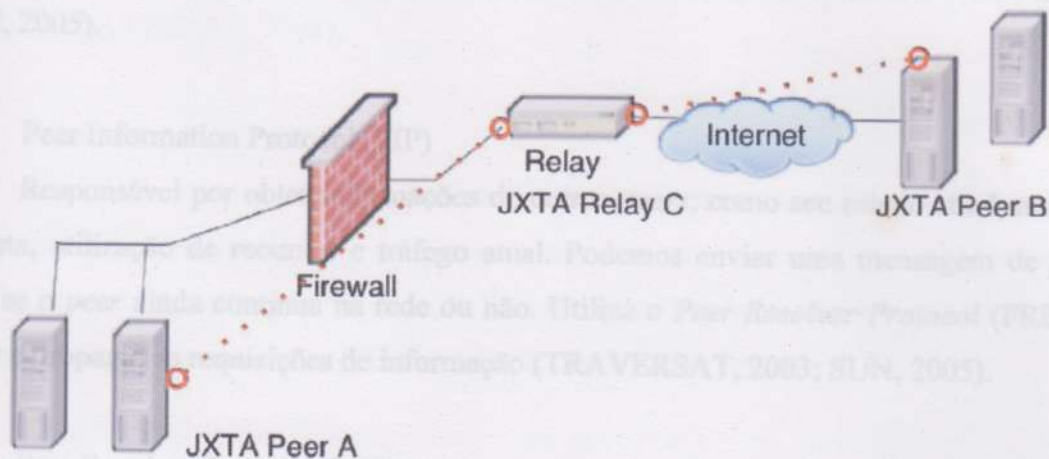


Figura 7 - Mensagem sendo roteado sobre um firewall
Fonte: SUN (2005)

5.5 PROTOCOLOS

O JXTA define diversos protocolos, sendo cada um deles responsável por executar uma tarefa específica na rede P2P como, por exemplo, realizar o roteamento de mensagens e possibilitar a comunicação entre *peers*. Cada protocolo é definido por uma ou mais mensagens em formato XML, trocadas entre os *peers* participantes da rede. Atualmente estão definidos seis protocolos.

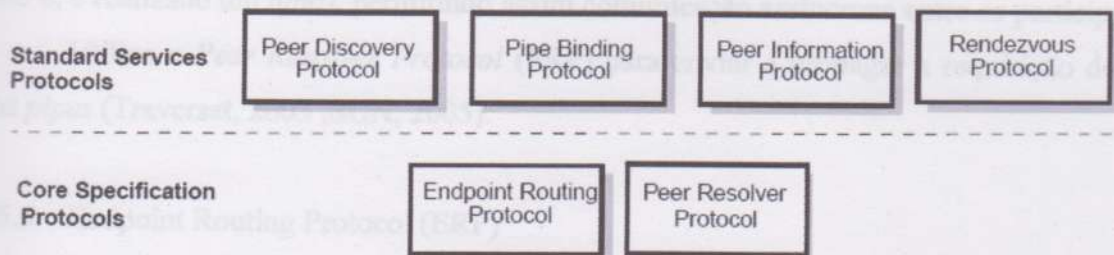


Figura 8 - Os protocolos JXTA
Fonte: TRAVERSAT (2003)

5.5.1 Peer Discovery Protocol (PDP)

Este protocolo habilita um *peer* a procurar *advertisements* do tipo *peers*, *peers groups*, *pipes* e conteúdo, além de enviar os *advertisements* publicados pelo próprio *peer* para o restante da rede. Este é o principal canal de comunicação do *peer* com o restante da rede. A descoberta de novos *peers* na rede pode ser feita explicitamente, especificando o UUID do *peer* ou do *peer group* desejado, neste último caso retornando todos os *peers* pertencentes ao

grupo. A descoberta de todos os *peers groups* também é possível (TRAVERSAT, 2003) (SUN, 2005).

5.5.2 Peer Information Protocol (PIP)

Responsável por obter informações de outros *peers*, como seu estado atual, tempo de resposta, utilização de recursos e tráfego atual. Podemos enviar uma mensagem de *ping* e saber se o *peer* ainda continua na rede ou não. Utiliza o *Peer Resolver Protocol* (PRP) para enviar e propagar as requisições de informação (TRAVERSAT, 2003; SUN, 2005).

5.5.3 Peer Resolver Protocol (PRP)

O PRP habilita um *peer* a enviar e receber *queries* para procura de *peers*, *peers groups*, *pipes* e outras informações na rede. Cada *query* é associada a um *Handler*, responsável por processar a mensagem e aguardar as respostas resultantes da *query* a serem enviadas pelos *peers*. *Peers* que provêm compartilhamento de arquivos podem oferecer uma busca avançada de seu repositório de dados para os outros *peers* (TRAVERSAT, 2003; SUN, 2005).

5.5.4 Pipe Binding Protocol (PBP)

Protocolo responsável por realizar a conexão de um *peer* com um ou mais *peers*. Para cada *peer* destino que a conexão virtual é estabelecida, um *pipe* de entrada e saída é aberto (isto é, é realizado um *bind*), permitindo assim comunicação assíncrona entre os participantes.

Utiliza o *Peer Resolver Protocol* (PRP) para enviar e propagar a requisição de *bind* dos *pipes* (Traversat, 2003 ;SUN, 2005).

5.5.5 Endpoint Routing Protocol (ERP)

Protocolo responsável por informar a rota para comunicação entre dois *peers* distintos quando uma conexão direta não é possível entre eles. A comunicação direta não será possível quando dois *peers* utilizarem redes de protocolos diferentes ou quando entre eles há um *firewall* ou a conexão é realizada por NAT. Neste caso o *Endpoint Routing Protocol* informa a rota necessária, indicando os *gateways* necessários para realizar a conexão. Caso a rota entre dois *peers* mude, ou a topologia da rede mude por qualquer motivo, este protocolo será utilizado para indicar a nova rota de comunicação entre eles (TRAVERSAT, 2003; SUN, 2005).

Qualquer *peer* pode ser um roteador deste tipo implementando o *Endpoint Routing Protocol* (TRAVERSAT, 2003).

5.5.6 Rendezvous Protocol (RVP)

Peers podem se tornar *peers* do tipo *Rendezvous*, indexando *advertisements* do grupo a que pertence e propagando mensagens para os ouvintes (*Listeners*) que estão registrados nele. Este protocolo controla a propagação de mensagens dos *peers Rendezvous* para os *peers* que pertencem ao mesmo grupo (TRAVERSAT, 2003; SUN, 2005).

As redes P2P e MANETs apresentam um grande crescimento no número de usuários. De Oliveira (2005) é apresentado alguns números a respeito deste crescimento desde o surgimento do Napster (NAPSTER, 2007).

Estatísticas apresentam que o tráfego referente aos aplicativos KaZaA (KAZAA, 2007) e Gnutella (GNUTELLA, 2007) tem sido de ordem de 40% a 60% de todo o tráfego da Internet. Em relação somente a redes Wi-Fi, o número de usuários, em 2003, era de 9,3 milhões e estudos apresentados no congresso Teleps 2004 prevêm que em 2008 serão cerca de 707 milhões de usuários, com uma taxa de crescimento anual de 137%.

As similaridades das Redes Móveis *Ad-hoc* e as Redes P2P chamam bastante atenção, ambas são auto-organizáveis, de natureza descentralizada e responsável por rotas requisições em ambientes distribuídos de topologia dinâmica. Além disso, em cada uma delas os segmentos de rede possuem características e funcionalidades equivalentes e podem enviar e receber requisições uns aos outros. Entretanto enquanto as MANETs são compostas de dispositivos com poucos recursos e, portanto, suscetíveis a falhas, as redes P2P são compostas pela sua tolerância a falhas. Essas características justificam o enorme sucesso nos últimos estatísticos apresentados acima.

O paradigma da comunicação *peer-to-peer* se adapta muito bem dentro de um ambiente móvel. Os *peers* podem entrar e sair a qualquer momento no ambiente das redes *ad-hoc*, uma vez que o nó encontra-se disponível na rede, todos os dispositivos móveis do ambiente podem se beneficiar e usar os serviços de infra-estrutura preparada. A infra-estrutura das redes *peer-to-peer* pode ser entendida como uma rede virtual (*overlay network*) sobre diferentes protocolos como TCP, UDP, Bluetooth e assim por diante.

Redes virtuais (*overlay network*), como P2P, têm a vantagem de incorporar diversos protocolos, permitindo desta forma que a aplicação opere entre elas de maneira transparente. Esta característica pode fazer com que ela possa ter uma infra-estrutura descentralizada,

6 INTEGRANDO MANET E JXTA

O surgimento da computação móvel e o avanço da Internet deram suporte ao surgimento de novas tecnologias e a quebra de paradigmas. Dentre tais tecnologias duas tem obtido especial atenção devido à vasta gama de aplicações que oferecem, *Peer-to-Peer* (P2P) e Redes Móveis Ad-hoc (MANET).

As aplicações são ditas P2P-Móvel, quando os aplicativos utilizam P2P (realiza, em algum momento, a comunicação direta entre *peers*) e são executadas em um dispositivo móvel. (OLIVEIRA, 2005).

As redes P2P e MANETs apresentam um grande crescimento no número de usuários. Em Oliveira (2005) é apresentado alguns números a respeito deste crescimento desde o surgimento do Napster (NAPSTER, 2007).

Estatísticas apresentam que o tráfego referente aos aplicativos KaZaA (KAZAA, 2007) e Gnutella (GNUTELLA, 2007) tem sido da ordem de 40% a 60% de todo o tráfego da Internet. Em relação somente a redes *wi-fi*, o número de usuários, em 2003, era de 9,3 milhões e estudos apresentados no congresso Telexpo 2004 prevêem que em 2008 serão cerca de 707 milhões de usuários, com uma taxa de crescimento anual de 127%.

As similaridades das Redes Móveis *Ad-hoc* e as Redes P2P chamam bastante atenção, ambas são auto-organizáveis, de natureza descentralizada e responsável por rotear requisições em ambientes distribuídos de topologia dinâmica. Além disso, em cada uma delas os integrantes da rede possuem características e funcionalidades equivalentes e podem enviar e responder requisições uns aos outros. Entretanto enquanto as MANETs são compostas de dispositivos com poucos recursos e, portanto, suscetíveis a falhas, as redes P2P são conhecidos pela sua tolerância a falhas. Essas características justificam o enorme sucesso nos números estatístico apresentados acima.

O paradigma da comunicação *peer-to-peer* se adapta muito bem dentro de um ambiente móvel. Os *peers* podem entrar e sair a qualquer momento no ambiente das redes *ad-hoc*, uma vez que o nó encontra-se disponível na rede, todos os dispositivos móveis do ambiente podem se beneficiar e usar os serviços da infra-estrutura preparada. A infra-estrutura das redes *peer-to-peer* pode ser entendida como uma rede virtual (*overlay network*) sobre diferentes protocolos como TCP, UDP, *Bluetooth* e assim por diante.

Redes virtuais (*overlay network*), como P2P, têm a vantagem de incorporar diversos protocolos, permitindo desta forma que a aplicação alterne entre elas de maneira transparente. Esta característica pode fazer com que ela possa ter uma infra-estrutura descentralizada,

dando desta forma a característica de se comunicar com outras redes também descentralizadas como CORBA (CORBA, 2007) ou Jini (JINI, 2007). Isto faz com que as redes peer-to-peer seja uma tecnologia adequada ao ambiente móvel (FREI, 2005).

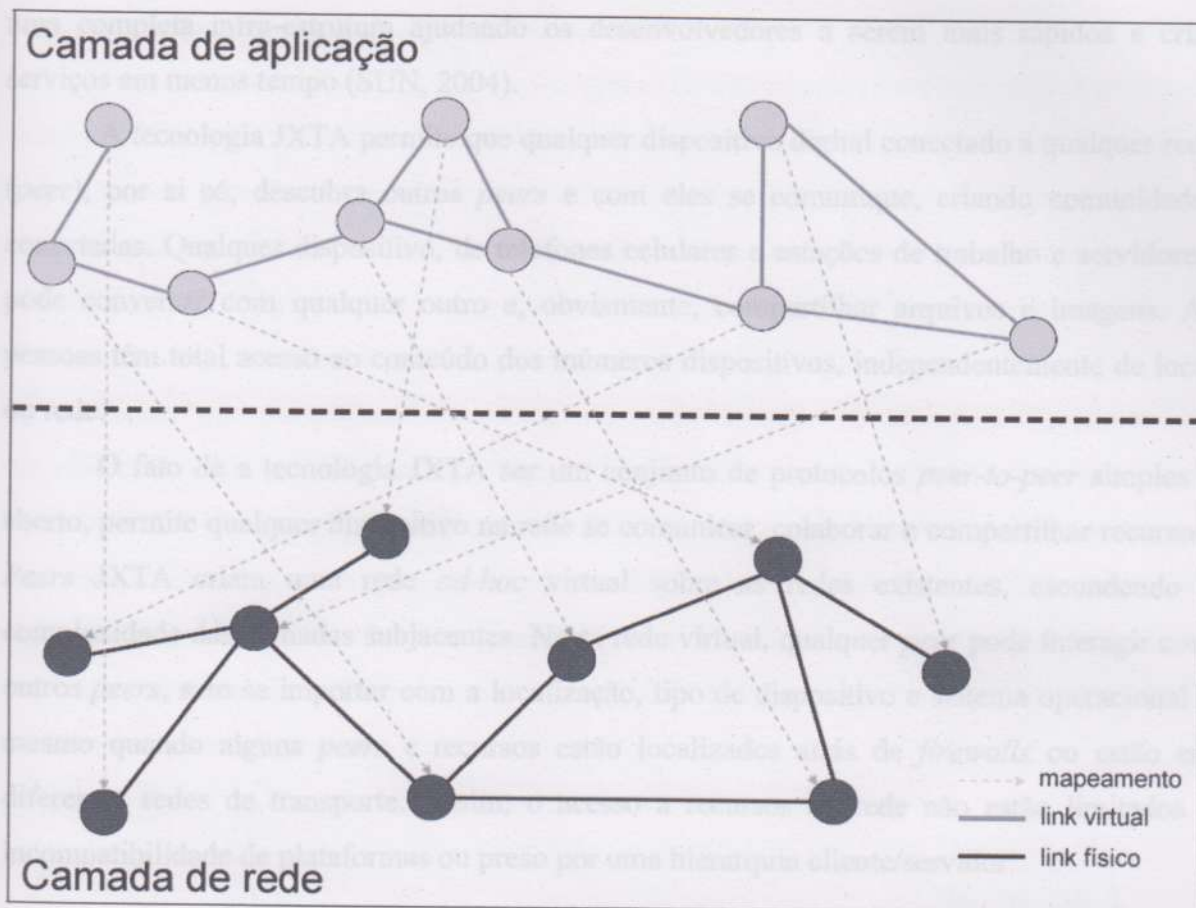


Figura 9 - Peer-to-Peer versus Rede Física

Neste novo cenário as aplicações não podem ser concebidas por paradigmas arquiteturais convencionais. Para este contexto onde a necessidade por mobilidade e integração é essencial, a arquitetura *peer-to-peer* (P2P) tem um ótimo resultado com as redes *ad-hoc*, por terem características muito parecidas.

Analisando o novo paradigma podemos perceber o surgimento de um ambiente totalmente dinâmico, onde qualquer dispositivo é um cliente e provedor de recursos em potencial, por isto *peer-to-peer* é naturalmente uma boa escolha para compor possíveis arquiteturas ubíquas.

A escolha do JXTA permite que problemas estruturais da MANET como a dinamicidade da rede causada pela entrada e saída de nós seja resolvida de forma satisfatória, permitindo o desenvolvimento de aplicações sem a preocupação sobre como esses problemas são resolvidos pelas camadas inferiores.

Computação P2P, em especial JXTA, permite que dispositivos distribuídos, serviços e redes cooperem juntos. Baseado em padrões abertos, possibilita que diversos dispositivos e redes sejam interoperáveis, e fornece flexibilidade porque as aplicações não estão amarradas a nenhum sistema proprietário. Removendo as complexidades da rede e do ambiente, fornece uma completa infra-estrutura ajudando os desenvolvedores a serem mais rápidos e criar serviços em menos tempo (SUN, 2004).

A tecnologia JXTA permite que qualquer dispositivo digital conectado a qualquer rede (*peer*), por si só, descubra outros *peers* e com eles se comunique, criando comunidades conectadas. Qualquer dispositivo, de telefones celulares a estações de trabalho e servidores, pode conversar com qualquer outro e, obviamente, compartilhar arquivos e imagens. As pessoas têm total acesso ao conteúdo dos inúmeros dispositivos, independentemente de local ou rede.

O fato de a tecnologia JXTA ser um conjunto de protocolos *peer-to-peer* simples e aberto, permite qualquer dispositivo na rede se comunicar, colaborar e compartilhar recursos. *Peers* JXTA criam uma rede *ad-hoc* virtual sobre as redes existentes, escondendo a complexidade das camadas subjacentes. Nesta rede virtual, qualquer *peer* pode interagir com outros *peers*, sem se importar com a localização, tipo de dispositivo e sistema operacional – mesmo quando alguns *peers* e recursos estão localizados atrás de *firewalls* ou estão em diferentes redes de transporte. Assim, o acesso a recursos da rede não estão limitados a incompatibilidade de plataformas ou preso por uma hierarquia cliente/servidor.

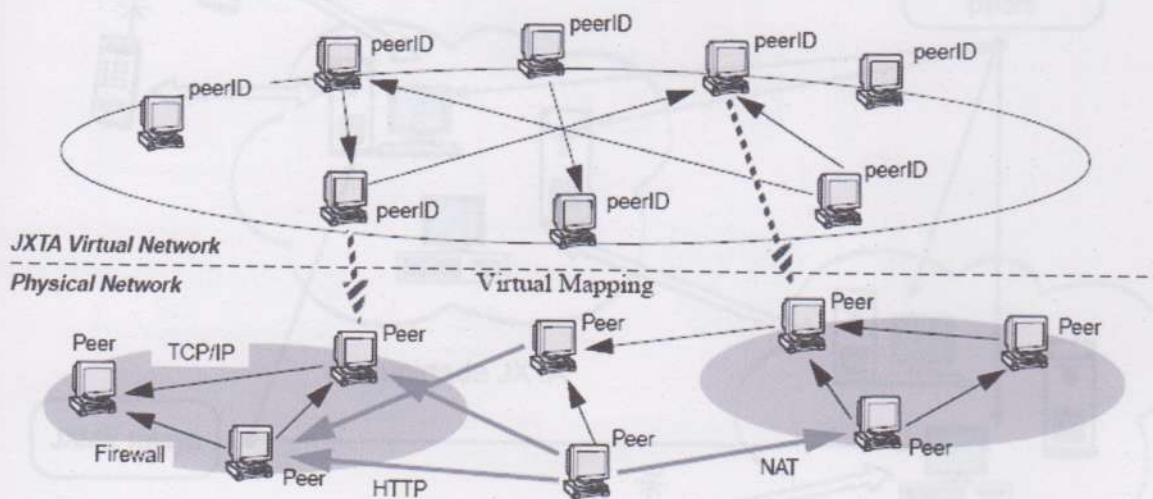


Figura 10 Rede virtual JXTA versus Rede física
Fonte: TRAVERSAT (2003)

Interoperabilidade é a principal meta do projeto (SUN, 2004), então JXTA foi projetado para possibilitar *peers* localizar e comunicarem-se uns com os outros, participar de comunidades e oferecer serviços através de diferentes plataformas e redes. Os mecanismos integrados de segurança como *Transport Layer Security* (TLS) e certificado digital, garantem segurança enquanto também facilita a comunicação.

Para desenvolvedores, fornece um conjunto de blocos que provê uma sólida base para computação distribuída e suporte as funções comuns de sistemas P2P. Aplicações distribuídas requerem uma grande infra-estrutura, e construir toda essa estrutura demanda tempo, recursos e programadores experientes. Usando JXTA, não é necessário inventar ou mesmo reinventar estruturas P2P, liberando o programador para criar novos recursos e agilizar o desenvolvimento de aplicações distribuídas.

Existe um subprojeto do JXTA chamado JXME, focado para funcionar em dispositivos muito limitados como PDA e *Smart Phones* com a Plataforma Java Micro Edition (JME). A principal desvantagem do JXME é a necessidade por um *relay peers*, o que torna uso ad-hoc impossível já que requer uma infra-estrutura na rede. Dois dispositivos que são colocados lado a lado não podem se comunicar diretamente. Para ambos os dispositivos poderem conversar necessitam de um servidor HTTP e isso é muito pesado para dispositivos com capacidades tão limitadas.

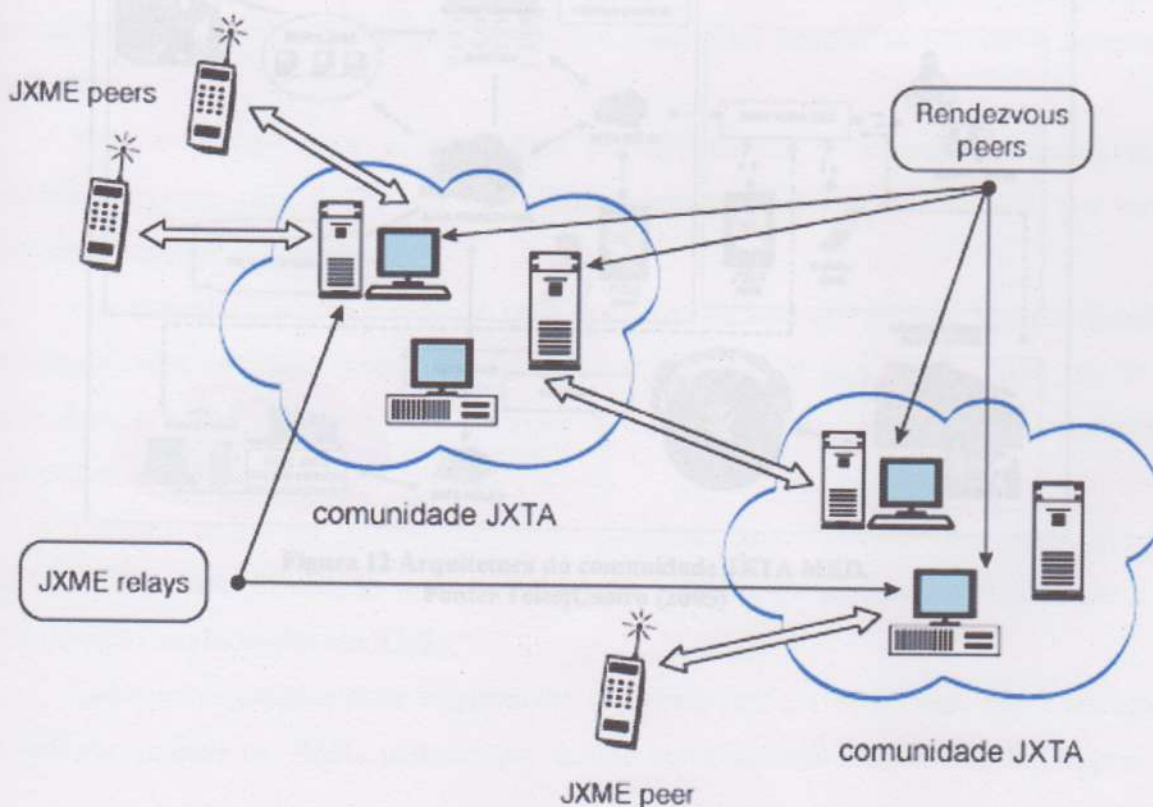


Figura 11 - Dispositivos utilizando JXME para acessar a rede JXTA

JMED é uma arquitetura de sistema peer-to-peer para serviços de assistência médica distribuída à distância que utiliza os conceitos abordados neste capítulo.

O projeto JMED é composto por dois subsistemas, JMED Controle e JMED Móvel, que operam de forma integrada no auxílio à prestação de assistência médica remota. O primeiro consiste de uma ferramenta computacional P2P cuja funcionalidade principal é suportar um ambiente de interação, em tempo real, entre médicos para: conversação, compartilhamento e visualização de imagens de exames de pacientes por meio de computadores interligados à Internet. O segundo subsistema complementa a funcionalidade do primeiro, ao possibilitar sua operação por meio de dispositivos de comunicação pertencentes a uma rede de telefonia móvel GSM/GPRS e que suportem a tecnologia J2ME (TELES;CASTRO, 2004).

O sistema JMED em si concebe a formação de uma topologia de rede P2P Híbrida baseada num estilo de comunicação ponto-aponto e multiponto, por meio dos quais os peers podem publicar e localizar recursos disponibilizados por/aos integrantes da rede (via advertisements JXTA) (TELES;CASTRO, 2004).

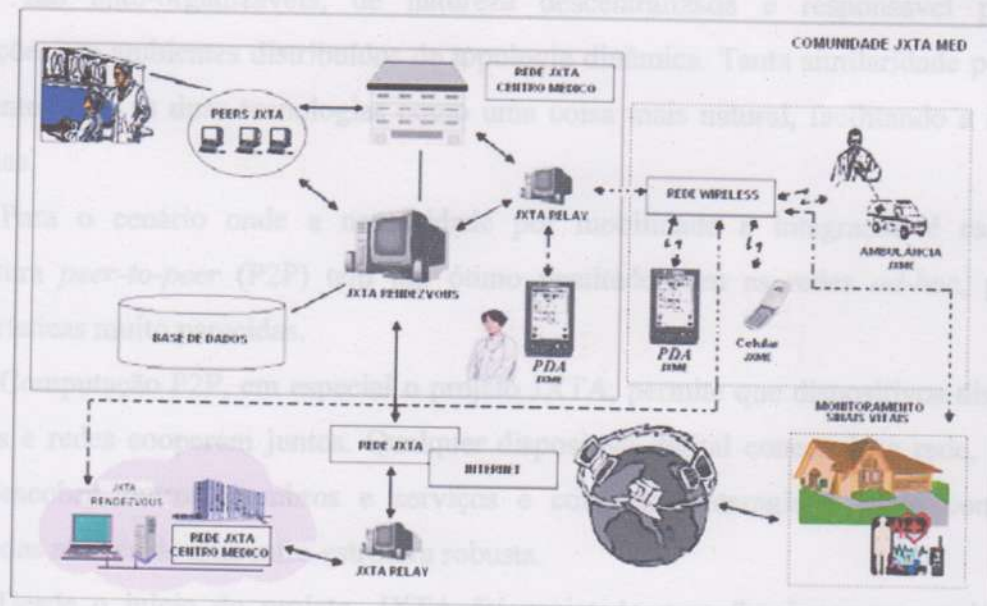


Figura 12 Arquitetura da comunidade JXTA MED.

Fonte: Teles;Castro (2005)

7 CONCLUSÃO

Aproveitando a crescente onda de dispositivos sem fio (MANET), este trabalho estudou os novos desafios que este ambiente gera e como integrar tais dispositivos em torno de uma única comunidade, caminhando rumo a computação ubíqua.

O ambiente formado por dispositivos móveis é caracterizado por ter uma alta dinamicidade e no contexto da computação ubíqua, cada dispositivo é um cliente e também um provedor de recursos em potencial, por isso peer-to-peer (é uma arquitetura distribuída) é nativamente uma boa escolha para compor arquiteturas ubíquas. Ela consegue agregar as características necessárias que um ambiente ubíquo necessita.

O modelo de arquitetura P2P não é um modelo novo, teve sua concepção formada ainda na criação da internet. Não tem a intenção de substituir a arquitetura cliente/servidor, mas é necessário ao desenvolvimento de aplicações escaláveis. Ainda não é um modelo maduro, muitas pesquisas sobre o assunto ainda deverão ser feitas, especialmente na área de algoritmos de roteamento para melhorar a eficiência das buscas por recursos.

Alem das características naturais que o peer-to-peer contempla para criação ambientes ubíquos, ficou bastante claro a sinergia que existe entre as MANETs e arquitetura P2P. Ambas são auto-organizáveis, de natureza descentralizada e responsável por rotear requisições em ambientes distribuídos de topologia dinâmica. Tanta similaridade permite um casamento entre as duas tecnologias como uma coisa mais natural, facilitando a integração entre elas.

Para o cenário onde a necessidade por mobilidade e integração é essencial, a arquitetura *peer-to-peer* (P2P) tem um ótimo resultado com as redes *ad-hoc*, por terem características muito parecidas.

Computação P2P, em especial o projeto JXTA, permite que dispositivos distribuídos, serviços e redes cooperem juntos. Qualquer dispositivo digital conectado a rede, por si só, pode descobrir outros membros e serviços e com eles interagir, criando comunidades conectadas através de uma infra-estrutura robusta.

Desde o início do projeto, JXTA foi projetado para funcionar nos mais diversos aparelhos. Por isso foi criada uma especificação e todos os protocolos e mensagens de comunicação são baseadas em XML.

Qualquer linguagem pode implementar a especificação, e como tudo que é trafegado na rede é com base em XML, permite que mesmo em diferentes plataformas, linguagens de

programação ou meio de comunicação, a rede formada seja interoperável. Essa interoperabilidade permite que haja a convergência tecnológica.

A escolha do JXTA para atuar na camada de aplicação traz grandes benefícios ao programador. Muito dos problemas que um desenvolvedor enfrenta quando esta criando sua própria rede P2P, o JXTA já possui a solução pronta. Como já foi dito ele é um *framework* genérico para criação de aplicações P2P, tem sua licença aberta para uso da comunidade de desenvolvimento, e é uma tecnologia independente de plataforma de sistema operacional, linguagem de programação e protocolo de comunicação de dados. Aqui pode ser implementado um ambiente que utilize apenas tecnologias não privadas, proporcionando a diminuição de custos em soluções a serem implementadas.

O caminho a ser percorrido pela computação ubíqua ainda é longo, porem com as tecnologias disponíveis como MANET, peer-to-peer e JXTA, já é possível vislumbrar cenários e aplicabilidades. Utilizando estas tecnologias já é possível criar modelos ubíquos.

BISIGNANO, Di Modica; Tomarchio. JIMobiPeers: A Middleware for Mobile Peer-to-Peer Computing in MANETs. In: ICDCS WORKSHOPS 2005, p. 743-751, 2005.

CORBA. Disponível em: <<http://www.omg.org>>. Acesso em: 26 nov. 2007.

CORDEIRO ; AGRAWAL, Mobile Ad hoc Networking. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 20., 2002, Búzios. Anais..., 2002.

COMPUSERV. Disponível em: <<http://www.compuserv.com>>. Acesso em: 26 nov. 2007.

FRÉL, Andreas Ralph. Jadedes: An adaptive pervasive middleware architecture. Tese Doutorado. SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, Suíça, 2005.

GNUTELLA. Disponível em: <<http://www.gnutella.com>>. Acesso em: 4 jan. 2007.

GONG, Li. JXTA: A Network Programming Environment. Maio/Junho 2001, Volume 5, p. 88 - 95.

IEEE. Disponível em: <<http://www.ieee.org>>. Acesso em: 26 nov. 2007.

REFERÊNCIAS

- AIM. Disponível em: <<http://www.aim.com>>. Acesso em: 4 jun. 2007.
- AOL. Disponível em: <<http://www.aol.com>>. Acesso em: 26 nov. 2007.
- ARAÚJO, Regina Borges de. **Computação Ubíqua: Princípios, Tecnologias e Desafios**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 21., 2003. Natal. **Anais...** Natal: SBRC, 2003. p. 45-115
- AUGUSTO, Carlos Henrique Pereira; REZENDE, José Ferreira de. **Uma Abordagem Adaptativa e Escalável para Descoberta de Serviços em Redes Ad Hoc**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 2006.
- BISIGNANO et al, Mario. **Expeerience: A Jxta middleware for mobile ad-hoc networks**. In: CONFERENCE ON PEER-TO-PEER COMPUTING, 2003.
- BISIGNANO; Di Modica; Tomarchio. **JMobiPeer: A Middleware for Mobile Peer-to-Peer Computing in MANETs**. In: ICDCS WORKSHOPS 2005: p.785-791, 2005.
- CORBA. Disponível em: <<http://www.omg.org>>. Acesso em: 26 nov. 2007.
- CORDEIRO ; AGRAWAL, **Mobile Ad hoc Networking**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 20., 2002. Búzios. **Minicursos** , 2002.
- COMPUSERV. Disponível em: <<http://www.compuserv.com>>. Acesso em: 26 nov. 2007.
- FREI, Andreas Ralph. **Jadabs: An adaptive pervasive middleware architecture**. Tese Doutorado. SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, Suíça. 2005.
- GNUTELLA. Disponível em: <<http://www.gnutella.com>>. Acesso em: 4 jun. 2007.
- GONG, Li. **JXTA: A Network Programming Environment**. Maio/Junho 2001, Volume 5, p. 88 - 95
- IEEE, Disponível em: <<http://www.ieee.org>>. Acesso em: 26 nov. 2007.

JINI. Disponível em: <<http://www.jini.org>>. Acesso em: 26 nov. 2007.

KAZAA. Disponível em: <<http://www.kazaa.com>>. Acesso em: 26 nov. 2007.

KELLERER, Dienstarchitekturen in der Telekommunikation: Evolution, Methoden und Vergleich, 1998.

JOVANOVIC, Mihajlo. **Modeling Large-Scale Peer-to-Peer Networks and a Case Study of Gnutella**. Dissertação (Mestrado em Ciência da Computação) - Division of Graduate Studies and Research of the University of Cincinnati, 2001.

LOUREIRO FILHO, Emerson Cavalcate. **Um Middleware Extensível para Disponibilização de Serviços em Ambientes Pervasivos**. Dissertação (Mestrado em Ciência da Computação). Universidade Federal De Campina Grande, Campina Grande, 2006.

LV et al. **Search and replication in unstructured peer-to-peer networks**. In: international Conference on Supercomputing, 16., 2002. New York

MAIA, Renato Figueiró. **Aplicações de Adaptação Dinâmica**. Monografia (Grupo de Sistemas Distribuídos) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2006.

NAPSTER. Disponível em: <<http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p4.html>>. Acesso em: 4 jun. 2007.

OLIVEIRA, Luciana Pereira. **Os dispositivos móveis e as redes Peer-to-peer (P2P)**. 2005. Dissertação (Graduação em Redes de Computadores) – Universidade Federal de Pernambuco, Recife.

OREILLY. **Peer-to-peer: Harnessing the Power of Disruptive Technologies**. Disponível em: <<http://www.OREILLY.com/catalog/peertopeer/chapter/ch01.html>> Acesso em: 4 jun. 2007.

PANISSON, André. **Implementação de um algoritmo de busca com controle de trafego distribuído e adaptacao de topologia para redes peer-to-peer baseados na plataforma jxta**. Monografia (Graduação em ciências da computação). Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.

RATNASAMY, Sylvia et al. **A Scalable Content-Addressable Network**. In: Proceedings of ACM SIGCOMM, 2001.

ROCHA et al. **Peer-to-Peer: Computação Colaborativa na Internet**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 22. **Minicursos**, 2004. Gramado. 2004 p. 3-46.

ROWSTRON ; DRUSCHEL, **Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems**, In: IFIP/ACM International Conference On Distributed Systems Platforms, 2001.

SCHOLLMEIER, Rüdiger. **A Definition of Peer-to-Peer Networking for the Classification of Peer to-Peer Architectures and Applications**, IEEE Internet Computing, 2002.

STOICA et al. **Chord: A scalable peer-to-peer lookup service for Internet applications**. In: PROCEEDINGS OF THE ACM SIGCOMM CONFERENCE. 2001.

SUN Microsystems, INC. **JXTA Technology: Creating Connected Communities**, 2004.

_____. **JXTA v2.3.x: Java Programmer's Guide**, 2005.

TELES e CASTRO. **Um Suporte de Transferência de Informações para Prontuários Eletrônicos: Experiência com JXTA**, 2005.

_____. **JMED – Uma Arquitetura de Sistema Peer-to-Peer para Serviços de Assistência Médica Distribuída à Distância**. 2004.

TRAVERSAT, Bernard. **Project JXTA 2.0 Super-Peer Virtual Network**, 2003.

WEISER, Mark. **The Computer for the 21st Century**. In: Scientific American, 1991.

ZHAO, Ben Y.; KUBIATOWICZ, John D.; JOSEPH, Anthony D.. **An Infrastructure for Fault-tolerant Wide-area Location and Routing**. U. C. Berkeley Technical Report UCB/CSD-01-1141, Abril, 2001