



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA - CIn
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
2005.1

OS DISPOSITIVOS MÓVEIS
E
AS REDES PEER-TO-PEER (P2P)

Por
LUCIANA PEREIRA OLIVEIRA
Trabalho de Graduação em Redes de Computadores

Recife
Agosto, 2005



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA - CIn
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
2005.1

LUCIANA PEREIRA OLIVEIRA

Os Dispositivos Móveis e As Redes
Peer-To-Peer (P2P)

Este trabalho foi apresentado ao programa de graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Djamel Fawzi Hadj Sadok

Recife
Agosto, 2005

Dedicatória

Dedico aos meus pais, a minha irmã e a minha avó pelo apoio, incentivo, amor e carinho que recebi em todos os momentos da minha vida.

Agradecimentos

Aos meus pais, Vicente e Maria das Graças, à minha irmã, Ana Lúcia e meu cunhado Túlio por todo amor, carinho, companhia, dedicação e apoio nas horas mais difíceis da minha vida. Obrigada por me ajudarem a não deixar de realizar meus sonhos.

Ao meu namorado, Luis Ricardo, pelo amor, pela companhia e amizade. Também agradeço a sua família, em especial, a sua mãe, Inês, pelo apoio.

Aos professores Judith Kelner e Djamel Sadok por fazer parte do Grupo de Pesquisa em Redes e Telecomunicações (GPRT), como também pela orientação.

Ao GPRT pela excelente infra-estrutura e o apoio necessário em termos físicos para a realização deste trabalho.

Ao grupo de trabalho em computação colaborativa (GT-P2P) pela motivação ao estudo dos assuntos abordados nesta monografia.

A Joseane Fidalgo e a Ramide por permitirem a dedicação exclusiva a este trabalho de graduação nas últimas semanas.

Ao professor Carlos Kamienski pela orientação e, principalmente, pelo o acompanhamento das avaliações de desempenho.

A Gabriel pela ajuda e revisão desta monografia.

Sumário

Dedicatória	3
Agradecimentos	4
Lista de Figuras	7
Lista de Tabelas	9
Resumo	10
Abstract.....	11
1 Introdução.....	12
1.1 Motivação	13
1.2 Objetivos.....	14
1.3 Trabalhos Relacionados.....	15
1.4 Estrutura da Monografia.....	16
2 Restrições e Requisitos para P2P-Móvel.....	18
2.1 Restrições Existentes em Redes P2P.....	18
2.2 Restrições dos Dispositivos Móveis	19
2.3 Tecnologia de Acesso	20
2.3.1 Taxa de Transferência	23
2.3.2 Estrutura da Rede Celular.....	25
2.4 Aspectos Econômicos.....	26
2.5 Considerações Finais	28
3 Aplicações	29
3.1 Monitorando Equipamento de uma Rede P2P.....	29
3.2 Voz sobre IP em uma Rede <i>Peer-to-Peer</i>	30
3.3 Compartilhamento de Arquivos	30
3.4 Blog	31
3.5 Troca de Mensagens	31
3.6 Jogos	32
3.7 Ferramenta Educacional com Entretenimento.....	32
3.8 Considerações Finais	34
4 Arquiteturas P2P.....	35
4.1 Arquitetura P2P-Móvel Pura	37
4.1.1 Mobile CHEDAR	38
4.1.2 Proem.....	41
4.1.3 Proem X Mobile CHEDAR.....	43
4.2 Arquitetura P2P-Móvel Híbrida	43
4.2.1 Arquitetura JXTA para Dispositivos Móveis (JXME).....	43
4.2.2 Arquitetura X-Peer	47
4.2.3 X-Peer x JXTA-JXME	50
4.3 Arquitetura para Comunicação entre as Redes P2P Pura e Híbrida	51
4.4 Arquitetura P2P-Móvel de Propósito Geral e Específico.....	54
4.4.1 Arquitetura P2P-Móvel de Propósito Geral (PnPAP)	54
4.4.2 Arquiteturas para Compartilhamento de Arquivos em Dispositivos Móveis	57
4.4.2.1 Utilizando Serviço MMS.....	57
4.4.2.2 Utilizando um <i>Peer</i> Intermediário.....	59
4.4.2.3 Extensão do eDonkey	61
4.4.2.4 Comparações entre as Arquiteturas para Compartilhamento de Arquivos.....	63
4.5 Considerações Finais	63
5 Modificações na Arquitetura X-Peer.....	65

5.1 Proposta e Implementação da Transmissão de Dados com UDP	65
5.2 Proposta de Novos Serviços para P2P-Móvel	67
5.3 Proposta LBS (Location Based Services).....	69
5.4 Contribuições.....	72
6 Avaliação de Desempenho	73
6.1 Motivação e Trabalhos Relacionados.....	73
6.2 Objetivo	73
6.3 Ambiente de Avaliação	74
6.3.1 Metodologia de Avaliação.....	74
6.3.2 Ambiente de Medição.....	75
6.4 Parâmetros e Métricas	77
6.5 Aplicativo da Avaliação de Desempenho.....	78
6.5.1 Requisito de Coleta da Métrica RTT.....	79
6.5.2 Requisito de coleta da Métrica Vazão	81
6.6 Cenários e Resultados.....	82
6.6.1 Cenário 1: C1.....	82
6.6.2 Cenário 2: C2.....	86
6.6.3 Cenário 3: C3.....	89
6.6.4 Cenário 4: C4.....	90
6.7 Avaliação dos Resultados	94
6.8 Dificuldades Encontradas	96
6.9 Considerações Finais	97
7 Conclusão	98
7.1 Considerações Finais	98
7.2 Principais Contribuições.....	99
7.3 Trabalhos Futuros	100
Referências	102
Apêndice A – Glossário.....	106

Lista de Figuras

Figura 1 Redes GPRS e WLAN utilizando técnica relay server e SIP.....	19
Figura 2 Padronizações da tecnologia de redes sem fio	21
Figura 3 Percurso de uma mensagem em uma rede ad-hoc	22
Figura 4 Estrutura simplificada da rede para transporte de pacotes de dados.....	25
Figura 5 Aparelho Firefly da empresa GadgetMadness	32
Figura 6 Aparelho i-Care da empresa Sogi	32
Figura 7 Interface gráfica baseada em teoria dos conjuntos do software FolkMusic.....	33
Figura 8 Arquitetura híbrida com super-peers.....	37
Figura 9 Topologias de redes Bluetooth.....	38
Figura 10 Topologia estrela numa rede Mobile CHEDAR	39
Figura 11 Entrega de dados entre os nós Mobile CHEDAR e CHEDAR.....	40
Figura 12 Arquitetura do Middleware Proem	42
Figura 13 Arquitetura JXTA	44
Figura 14 Entidades de uma rede JXTA.....	45
Figura 15 Arquitetura JXTA com JXME 1.0	46
Figura 16 Arquitetura proxyless JXME	46
Figura 17 Infra-estrutura X-Peer na rede da RNP	47
Figura 18 Arquitetura X-Peer	48
Figura 19 Tamanho médio das mensagens referentes aos serviços X-Peer	50
Figura 20 Arquitetura para comunidade P2P	52
Figura 21 Modelagens do Proxy Mobile	53
Figura 22 Multi-hop X Multi-destination.....	54
Figura 23 Arquitetura PnPAP.....	55
Figura 24 Exemplo da comunicação entre o ME e o Motor.....	56
Figura 25 Busca e compartilhamento de arquivo MMS.....	58
Figura 26 Exemplo de transação MMS	59
Figura 27 Peer intermediário	60
Figura 28 Visão geral da arquitetura P2P-Móvel	61
Figura 29 Comunicação direta com TCP	66
Figura 30 Comunicação indireta com TCP	66
Figura 31 Componente refletor na rede X-Peer.....	67
Figura 32 Modificação da arquitetura X-Peer para atender a mobilidade.....	68
Figura 33 Modificação na arquitetura X-Peer para atender LBS	71
Figura 34 Uso do serviço XLB com GPS	71
Figura 35 Diagrama do ambiente de medição	76
Figura 36 Tempo de resposta ou RTT	78
Figura 37 Execução de uma comunicação indireta	79
Figura 38 Execução de uma comunicação direta	79
Figura 39 Diagrama de seqüência simplificado do aplicativo para coletar RTT	80
Figura 40 Diagrama de seqüência simplificado do aplicativo para coletar RTT	81
Figura 41 RTT e Vazão da comunicação direta entre peer móveis.....	83
Figura 42 RTT e Vazão da comunicação indireta entre peer	84
Figura 43 Comportamento do tempo de resposta com o aumento das mensagens	84
Figura 44 Latência no X-Peer na comunicação peer fixo - peer fixo.....	85
Figura 45 Descrição do diagrama de seqüência da troca de mensagens DHTDeliver...	86
Figura 46 RTT (ms) e vazão (msg/s) da comunicação direta entre um peer móvel e um peer fixo.....	87
Figura 47 RTT (ms) e vazão (msg/s) da comunicação indireta entre um peer móvel e	

um peer fixo.....	88
Figura 48 RTT e vazão na comunicação entre um dispositivo móvel e um desktop	88
Figura 49 Comparando os cenários 2 e 3	90
Figura 50 Comparando o tempo de resposta UDP e TCP	91
Figura 51 Comparação do RTT usando apenas J2ME e Módulo P2P	92
Figura 52 Comparação do RTT usando apenas J2ME e Módulo P2P no emulador	93
Figura 53 RTT X gênero do jogo X rede sem fio.....	95
Figura 54 Problema com coletas automatizadas	96

Lista de Tabelas

Tabela 1 Taxa de transferência das tecnologias de acesso	23
Tabela 2 Canais utilizados no padrão 802.16	24
Tabela 3 Estimativa de custos de conteúdo download	27
Tabela 4 Jogadores X Arquitetura	32
Tabela 5 Resumo das características de redes Cliente/Servidor e Peer-to-Peer.....	36
Tabela 6 Parâmetros para o cenário 1.....	82
Tabela 7 Parâmetros para o cenário 2.....	87
Tabela 8 Parâmetros para o cenário 3.....	91
Tabela 9 RTT em diferentes jogos	94

Resumo

Os dispositivos móveis e as redes *peer-to-peer* (P2P), se analisados separadamente, apresentam um grande crescimento no número de usuários. No entanto, existe uma diversidade de aparelhos com diferentes capacidades e grande variedade de tecnologias para a comunicação sem fio (Wi-Fi, Wi-Max, Bluetooth, GPRS, UMTS), e a combinação de aparelhos e interfaces de rede é apenas um dos novos desafios para os ambientes P2P.

Atualmente, existem aplicações que utilizam a tecnologia Wi-Fi e a arquitetura para P2P-Móvel. O grupo desenvolvedor do aplicativo Skype, que permite a transmissão de voz utilizando VoIP (*Voice over Internet Protocol*) e a arquitetura P2P, está disponibilizando uma versão para Pocket PC, denominada PocketSkype. O grupo eMule também está explorando a arquitetura P2P com dispositivos móveis que resultou, inicialmente, no lançamento de uma aplicação para celulares denominada MobileMule. Este aplicativo permite aos usuários terem um maior controle do compartilhamento de arquivos através dos celulares, principalmente, quando não estiverem próximos de seus *desktops* que utilizam o eMule para compartilhar arquivos.

Outro fato que está ocorrendo no campo de redes P2P é a criação de novo protocolos e modificação de antigos protocolos para atender aos *peers* móveis. A arquitetura P2P híbrida possibilita a existência de dois tipos de *peers* (*peers* e *super-peers*), existindo uma diferença de hierarquia em que os *super-peers* possuem o nível superior aos *peers*. JXTA é um exemplo deste tipo de arquitetura P2P. O grupo que desenvolveu essa arquitetura criou uma versão dos protocolos do *framework* JXTA para atender especificamente dispositivos móveis, denominada JXME.

Um exemplo de modificação de protocolos para atender aos *peer* móveis foi realizado pelo grupo de trabalho em computação colaborativa (GT-P2P) que desenvolveu a infra-estrutura X-Peer. Em 2004, o grupo identificou que o tempo para realizar o processamento dos protocolos definidos em XML (*Extensible Markup Language*) era muito elevado. Por isso, em 2005, o grupo decidiu que todos os protocolos utilizados no X-Peer fossem substituídos por protocolos no formato binário. Esses protocolos são menores e requerem pouco processamento, o que é desejado pelas aplicações P2P em aparelhos móveis, tais como *handhelds*.

Desse modo, este trabalho descreve o estudo realizado sobre P2P-Móvel, considerando os novos desafios aos ambientes P2P, as aplicações e as arquiteturas. Em especial, a arquitetura X-Peer foi avaliada, o que resultou na apresentação de propostas para que essa arquitetura atenda melhor aos dispositivos móveis. Para realizar esta avaliação, uma aplicação foi desenvolvida com o objetivo de coletar amostras do tempo de resposta e vazão na camada de aplicação, de forma a permitir o melhor estudo dos *peers* utilizando uma rede X-Peer em ambiente real.

Palavras-Chave: P2P-Móvel, *peer-to-peer*, *peers* móveis e X-Peer.

Abstract

The mobile devices and the peer-to-peer (P2P) networks, if analyzed separately, presents a great growth in their number of users. However, there is a wide diversity of devices, with different capacities and great variety of technologies for wireless communications (Wi-Fi, Wi-Max, Bluetooth, GPRS, UMTS), and the combination of devices and network's interfaces is only one of the new challenges for P2P environments.

Nowadays, there are several applications that use Wi-Fi technology and Mobile-P2P architecture. The developer group of Skype provides internet telephony service that is based on the P2P principle and already launched a version of their product for Pocket PC, named PocketSkype. The eMule group is also exploring P2P architecture for mobile devices and launched an application called MobileMule. This application allows the users to have a bigger control of their shared files through cell phones, mainly when they are not close to their desktops, that use emule to share files.

Another fact that is occurring in the field of P2P networks, is the creation and modification of protocols to assist the mobile peers. Hybrid P2P architecture allows the existence of two types of peers (super-peers and peers), thus existing a hierarchy difference where the super-peers are in a superior level compared to the peers. JXTA is an example of this type of P2P architecture. The group that developed this architecture created a different version of the JXTA framework protocols to assist mobile devices, called JXME.

An example of protocol modification to assist mobile peers was developed by the work group in collaborative computation - GTP2P that developed the X-Peer infrastructure. In 2004, the group identified that the time to accomplish the processing of protocols defined in XML (Extensible Markup Language) was very high. Therefore, in 2005, the group decided that all the protocols used in the X-Peer would be replaced by protocols in the binary format. Since those protocols are smaller and require less processing time, they are exactly what is desired by P2P applications in mobile devices, such as handhelds.

This work describes the study carried through on the Mobile-P2P, considering the new challenges to P2P environments, the applications and the architectures. In especial, the X-Peer architecture was evaluated, what resulted in modification proposals so that this architecture assists better the mobile devices. To accomplish the evaluation work, an application was developed with the objective of collecting samples of response time and dataflow in the application layer, to allow the study of peers using a X-Peer network in a real environment.

Key Works: Mobile-P2P, peer-to-peer, mobile peers and X-Peer.

1 Introdução

Na literatura existem diversas definições para *peer-to-peer* (P2P). Algumas dessas definições restringem o conceito, que classifica um sistema ou aplicação, como P2P ou cliente/servidor. As duas definições a seguir, encontradas na literatura, exemplificam o caso, em que o conceito P2P é restrito.

1. Stoica e Balakrishnan [1] definiram P2P como: “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 funcionalidade”;
2. Rowstron e Druschel [2] também reforçam essa restrição com a seguinte frase: “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”.

No entanto, variações sobre as definições anteriores surgiram para 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.

A vantagem das redes *peer-to-peer* é quanto à distribuição que oferece:

- Escalabilidade, porque não há gargalos para o crescimento em relação à 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, uma vez que todos podem atuar como clientes e servidores, aumentando na mesma proporção à quantidade de recursos compartilhados.
- Robustez, porque não há um ponto único de falha. Em um sistema cliente/servidor, se ocorrer a impossibilidade de acessar um servidor, todos os clientes necessitam aguardar o restabelecimento do servidor, para então continuar com suas atividades. Em um sistema P2P, um outro *peer* poderá disponibilizar os mesmo serviços do *peer* que está fora da rede.
- Flexibilidade, porque 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.

Neste contexto, os dispositivos móveis também podem representar um *peer* na rede P2P, denominado *peer* móvel ou nó móvel. Este é caracterizado, principalmente, pela mobilidade proporcionada pela rede sem fio, e pelas diversas restrições que serão detalhadas no segundo capítulo desta monografia.

As redes *peer-to-peer* (P2P) e os dispositivos móveis que acessam redes sem fio, se analisados separadamente, apresentam um grande crescimento no número de usuários. Sabe-se que a tecnologia P2P tem crescido exponencialmente desde o surgimento do Napster [3]. Estatísticas apresentam que o tráfego referente aos aplicativos KaZaa [4] e Gnutella [5] tem sido da ordem de 40% a 60% de todo o tráfego da Internet [6]. Em relação somente a redes Wi-Fi¹, o número de usuário, 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%[7]. Portanto, qual a viabilidade do uso de dispositivos móveis em redes *peer-to-peer*? A edição especial da veja de julho de 2005 [8] publicou: “A empresa de análise e pesquisa britânica Datamonitor prevê que, até 2009, pelo menos 13 milhões de consumidores europeus usarão a transmissão *wireless* para compartilhar arquivos de áudio e vídeo...”.

Nesta visão, em que os dispositivos móveis poderão participar das redes P2P, são realizadas proposta de novas arquiteturas P2P (publicadas em [31], [32] e [36]) e de modificação de antigas arquiteturas P2P (publicadas em [13], [28], [29], [34], [35] e [39]) para fornecer um melhor ambiente a esses equipamentos restritos em capacidade de processamento, de memória entre outros recursos. As redes P2P que permitem a existência e participação de *peers* móveis podem ser denominadas P2P-Móvel. Portanto, os *peers* de uma rede P2P-Móvel podem ser apenas aparelhos móveis, ou aparelhos fixos e móveis. As aplicações são denominadas aplicações P2P-Móvel, quando os aplicativos utilizam o conceito P2P (realiza, em algum momento, a comunicação direta entre *peers*) e são executadas em um dispositivo móvel.

1.1 Motivação

Como foi identificado na introdução deste capítulo, existe um grande crescimento do número de usuário de rede sem fio, no entanto Wi-Fi é apenas um elemento de um grande conjunto formado por padrões para comunicação sem fio. Outros padrões que fazem parte desse conjunto são: Bluetooth², UWB³, ZigBee⁴, Wi-Max⁵, assim como os padrões da telefonia celular, tais como GPRS (*General Packet Radio Service*) e UMTS (*Universal Mobile Telecommunications System*).

¹ Wi-Fi – É um nome comercial para um padrão de rede *Wireless* chamado de 802.11b. A velocidade e alcance são tipicamente de 12Mbps e 100m respectivamente.

² Bluetooth – Apresenta conexão via rádio e frequência com alcance de 10m e capacidade de transmitir 720 Kbps.

³ UWB (*Ultra Wide Band*) – Esta é uma tecnologia de transmissão de dados sem fio que ao invés de operarem numa frequência fixa, os transmissores UWB utilizam um número quase infinito de frequências entre 0 e 60 GHz, sem permanecer em uma única frequência por mais do que algumas frações de segundo. Apenas as duas partes envolvidas conhecem o padrão de frequências utilizado, o que ajuda a manter a segurança dos dados. Tem a capacidade de transmitir dados até 500Kbps, mas com um alcance de apenas 10 metros.

⁴ ZigBee – Tem como proposta tornar-se padrão *wireless* de baixo consumo e curto alcance para monitoração e automação de aplicações industriais, comerciais ou urbanas, sua taxa de transferência é de 200Kbps e tem o alcance de 75m.

⁵ Wi-Max (*Worldwide Interoperability for Microwave Access*) – É o padrão 802.16 também conhecido como *Air Interface for Fixed Broadband Wireless Access Systems* com alcance de até 40Km, possibilitando taxas de transferência de até 75Mbps.

A combinação dessas interfaces de comunicação, e das diversas capacidades e restrições dos dispositivos móveis gera novos desafios para o desenvolvimento de arquiteturas e aplicações *peer-to-peer*.

Esses novos *peers* são caracterizados pela grande mudança de localização e modificação de estado (*online* ou *offline*). Diante dessa descrição, é possível identificar um novo tipo de dado – localização geográfica dos usuários – que poderá ser interessante para uma comunidade P2P.

A publicação e compartilhamento da posição geográfica pelos *peers* móveis e fixos, assim como as buscas de informações relacionadas com a localização, proporcionam novos serviços e aplicações. Um exemplo é o cenário em que turistas de uma excursão formam uma rede P2P-Móvel. Os *peers* móveis, que representam os turistas, poderão não só compartilhar as fotos, músicas e agenda da viagem, mas também localizações de pontos turísticos e localizações de cada turista, durante a viagem. Esse compartilhamento facilitará o inter-relacionamento de todo o grupo, provavelmente, evitando atrasos causados por desencontros. Desse modo, P2P-Móvel pode proporcionar ambientes mais dinâmicos aos usuários de dispositivos móveis.

Para viabilizar o atendimento aos *peers* móveis, novas arquiteturas e protocolos estão sendo propostos, assim como antigas arquiteturas e protocolos P2P estão sendo modificados. Um exemplo de modificação de protocolos, para levar em consideração os *peer* móveis, foi realizado pelo grupo de trabalho em computação colaborativa (GT-P2P), que desenvolveu a infra-estrutura X-Peer. No final de 2004, foi identificado que o tempo, para realizar o processamento dos protocolos do X-Peer definidos em XML (*Extensible Markup Language*), era muito elevado. Por isso, em janeiro de 2005, o grupo decidiu que todos os protocolos utilizados no X-Peer seriam substituídos por protocolos no formato binário. Essa modificação proporcionou protocolos menores e um menor processamento, o que é requerido e desejado por aplicações P2P-Móvel em aparelhos com restrições de recursos, tais como *handhelds*.

1.2 Objetivos

A participação no grupo de trabalho em computação colaborativa resultou no desenvolvimento de módulos que facilitam a implementação de aplicações P2P para a arquitetura X-Peer. Assim como, o desenvolvimento de aplicações para equipamentos com baixas restrições (por exemplo, *desktops* e *notebooks*) e para dispositivos com altas restrições (por exemplo, *handhelds* e *PocketPCs*).

O conhecimento obtido nesse projeto motivou no estudo aprofundado dos desafios que os ambientes sem fio trazem para as aplicações P2P, assim como a pesquisa de outras arquiteturas P2P que permitam a participação de *peers* móveis.

Dessa forma, considerando-se redes *peer-to-peer* e dispositivos móveis, o objetivo principal deste trabalho de graduação é estudar aspectos relacionados às redes P2P-Móvel, propor modificações na arquitetura X-Peer para que ela ofereça um melhor ambiente aos dispositivos móveis, assim como realizar uma avaliação de desempenho da atual arquitetura X-Peer, por meio de medições em ambiente real. Desse modo,

verificar a viabilidade de aplicações P2P-Móvel. Para isso, os objetivos específicos deste trabalho são descritos abaixo:

- 1 Compreender problemas e requisitos dos dispositivos móveis para a participação de uma rede P2P-Móvel.
- 2 Identificar as aplicações existentes para dispositivos móveis.
- 3 Identificar arquiteturas P2P que permitem a participação de *peers* móveis, e em seguida realizar comparações.
- 4 Obter maior conhecimento sobre a arquitetura X-Peer para descrever uma proposta de modificação.
- 5 Desenvolver uma aplicação utilizando o X-Peer, para ser utilizada na avaliação de desempenho. Esta aplicação tem como objetivo permitir a realização de coletas para as medições da avaliação.
- 6 Realizar avaliação de desempenho com a finalidade de verificar a viabilidade de aplicações P2P-Móvel, utilizando a arquitetura X-Peer.

1.3 Trabalhos Relacionados

P2P-Móvel é um assunto recente (a referência mais antiga é de 2001), por isso não foram encontrados trabalhos que descrevessem um levantamento bibliográfico de aspectos dessas redes P2P. Portanto, serão apresentados os trabalhos relacionados para cada assunto abordado nesta monografia, exceto ao assunto de identificação de problemas e requisitos dos dispositivos móveis, porque os trabalhos relacionados estão em quase todas as referências neste documento.

Um dos assuntos abordados nesta monografia é a identificação de aplicações em que os equipamentos móveis utilizam uma rede P2P e aplicações P2P-Móvel. Para isso, são utilizadas as seguintes pesquisas. Em [16] e [14] os autores apresentam aplicações que são P2P-Móvel, assim como aplicações que não são P2P-Móvel, mas que utiliza uma rede P2P. O artigo, em [21], descreve uma breve comparação de arquitetura P2P versus número de jogadores, assim como descreve aspectos relacionados com jogos vários usuários utilizando dispositivos móveis. Em [24], Wiberg inova com a proposta de uma aplicação educativa P2P-Móvel. Esses foram os trabalhos mais relevantes em relação à descrição de aplicações em rede P2P com dispositivos móveis.

Em relação a arquiteturas, um outro conjunto de pesquisas e trabalhos correlacionados será apresentado. Em primeiro lugar esta pesquisa se baseia em [27], esta referência descreve a arquitetura X-Peer que possui um *framework* semelhante ao de JXTA. Os documentos [34] e [35] apresentam JXTA e sua extensão para P2P-Móvel. Em [28] e [29], a arquitetura CHEDAR e sua extensão são apresentadas. CHEDAR também se inspira em conceitos e mecanismos para fornecer uma arquitetura P2P, no entanto é classificada como uma arquitetura diferente das duas anteriores. Em [31] e [32], outra arquitetura semelhante a extensão de CHEDAR, denominada Proem, é apresentada. Em [36] é descrita uma proposta para que possa existir a comunicação de *peers* que utilizam

as arquiteturas híbridas (JXTA e X-Peer) e puras (CHEDAR e Proem). Em [13] e [39], a extensão de uma arquitetura específica para compartilhamento de arquivos (eDonkey) é apresentada utilizando conceitos *caching* e *crawler*. Diferente desta última, em [9], é proposta uma arquitetura genérica para o desenvolvimento de aplicações P2P, em que abstrai o tipo da interface de comunicação e do protocolo P2P e P2P-Móvel.

Além dos documentos enumerados anteriormente, outros trabalhos relevantes foram utilizados para descrever a proposta de modificação da arquitetura X-Peer. A pesquisa em [38] aborda uma técnica para permitir a comunicação direta utilizando UDP, mesmo quando os *peer* estão sob NAT. Esta foi a principal referência para não só propor, mas também implementar um componente que facilita a implementação de aplicações P2P e P2P-Móvel, mesmo sob NAT. Em [22], é descrito um trabalho sobre LBS (*Location Based Services*), que contribui na proposta para permitir que exista este serviço em uma rede X-Peer.

Foram encontradas poucas pesquisas sobre avaliação de desempenho de arquiteturas P2P-Móvel, por isso apenas as seguintes referências foram utilizadas. Em [13] e [39], é apresentado um modelo de simulação para avaliar a proposta de modificação da arquitetura eDonkey. A referência [13] analisa apenas as estratégias de *caching* para o *peer caching*. Em [39], são avaliados o *peer crawler* e as modificações no servidor de índices. O artigo [15] é outra referência que avalia a questão de aplicações de compartilhamento de arquivos utilizando dispositivos móveis. Esse trabalho realizou uma medição por meio de um software para verificar a viabilidade de compartilhamento de arquivos em dispositivos móveis. Pode-se perceber que poucos artigos (apenas [13], [15] e [39]) foram publicados em relação à avaliação de desempenho no ambiente P2P-Móvel. Além disso, esses trabalhos focam principalmente na análise da viabilidade de aplicações de compartilhamento de arquivos. Portanto, a avaliação realizada neste trabalho de graduação tem como objetivo geral contribuir na avaliação de uma arquitetura diferente da eDonkey, assim como verificar a viabilidade da utilização de outras aplicações P2P em ambiente móveis sem fio. O diferencial deste trabalho é a realização da medição de P2P-Móvel com tecnologia 802.11 e o estudo da viabilidade de aplicações P2P utilizando a arquitetura X-Peer que permite o desenvolvimento de diversas aplicações, ou seja, não restringe a avaliação a aplicações de compartilhamento de arquivos.

1.4 Estrutura da Monografia

O documento é composto por sete capítulos, cada um apresentado da seguinte forma:

Capítulo 1 – Apresenta uma introdução a respeito de conceitos sobre redes *peer-to-peer*, *peers* móveis, aplicações e redes P2P-Móvel. Descreve a motivação para o estudo de P2P-Móvel e para a elaboração deste trabalho de graduação, assim como o objetivo da monografia e trabalhos relacionados a este documento.

Capítulo 2 – Identifica as restrições dos aparelhos móveis e os requisitos para viabilidade de arquiteturas e aplicações P2P-Móvel.

Capítulo 3 – Apresenta aplicações que utilizam uma rede P2P e aplicações P2P-Móvel para dispositivos móveis.

Capítulo 4 – Apresenta arquiteturas para P2P-Móvel e finaliza comparando as arquiteturas citadas neste capítulo.

Capítulo 5 – Descreve a proposta de modificação da arquitetura X-Peer. Esta arquitetura é apresentada no capítulo 4.

Capítulo 6 – Descreve a aplicação que foi implementada para realizar a medição, assim como a avaliação de desempenho, seus resultados e dificuldades encontradas. O capítulo é finalizado com conclusões obtidas na avaliação.

Capítulo 7 – Apresenta as conclusões sobre o trabalho, enfatizando as principais contribuições e sugestões para trabalhos futuros.

2 Restrições e Requisitos para P2P-Móvel

O crescimento do uso de dispositivos móveis – *smartphone e PDAs*⁶ – trouxe novos desafios para a construção de aplicações P2P-Móveis [9]. Por isso, o uso de aplicações *peer-to-peer* em ambientes móveis requer a compreensão não só dos problemas encontrados em redes P2P, mas também das restrições intrínsecas dos dispositivos móveis, das redes sem fio e aspectos econômicos. Esses tópicos serão abordados nas seções a seguir.

2.1 Restrições Existentes em Redes P2P

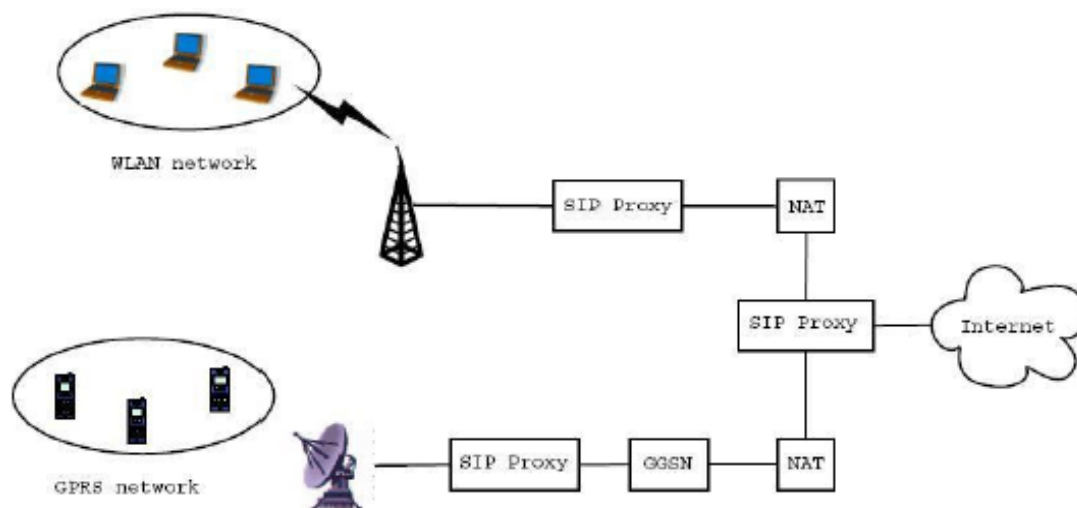
Um das dificuldades existentes em redes P2P, independentes de a rede ser fixa ou móvel, é a questão de máquinas sob *Network Address Translation* (NAT) e *firewalls*⁷ que estão cada dia mais comuns, mas restringido a alcançabilidade entre os computadores. Um *firewall* normalmente bloqueia conexões *Transmission Control Protocol* (TCP) provenientes de computadores externos, impedindo então que os *peers* externos se conectem ao computador protegido.

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 uma 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*.

Para solucionar esses problemas, existem técnicas que, entretanto prejudicam o tempo de resposta. Uma possível técnica para solucionar problemas com NAT em P2P-Móvel é o uso de um refletor central (*relay server*) com o protocolo SIP (*Session Initiation Protocol*) [10] e [12]. Essa última técnica pode ser observada na Figura 1, que apresenta celulares de uma rede GPRS se comunicando com *notebooks* de uma rede *Wireless Local Area Networks* (WLAN). Nestes cenários, os celulares se comportam como máquina sob NAT, porque mesmo não possuindo IP na tecnologia 2/2,5G, eles podem obter um identificador por meio do *Gateway GPRS Support Node* (GGSN). A estrutura de rede celular e seus problemas estão detalhados na seção 2.3.2.

⁶ PDA - *Personal Digital Assistant*, termo que designa pequenos aparelhos de mão, ou *palm tops*, com funcionalidade de computador. Alguns lançamentos incorporam celulares - e estes a funcionalidade de um PDA.

⁷ *Firewall* – *Software* para gerenciamento de entrada e saída de informações pela rede.



Fonte: referência [10]

Figura 1 Redes GPRS e WLAN utilizando técnica *relay server* e SIP

Outra restrição, que não é particular para P2P-Móvel, é a interoperabilidade entre os protocolos P2P existentes. Atualmente, existe uma grande quantidade de protocolos e mensagens P2P disponíveis, portanto um requisito desejável, não só para a comunicação P2P entre os dispositivos móveis, é a interoperabilidade entre os protocolos e mensagens P2P existentes. Os protocolos JXTA⁸ e JXME⁹ são interoperáveis, mas JXTA não é nativamente interoperável com outros protocolos P2P. Ao contrário desses, os protocolos Jabber e SIP são facilmente interoperáveis com outros protocolos e mensagens P2P, devido à estrutura modular e a disponibilidade de *plugins* [9].

Um dos problemas na interoperabilidade entre os protocolos P2P é a questão do endereçamento. Basicamente todo protocolo P2P tem seu próprio método de nomear e endereçar os *peers*, por isso existe a necessidade de tradução de endereços [9].

2.2 Restrições dos Dispositivos Móveis

Os dispositivos móveis, quando comparados com os computadores *desktops*, possuem várias restrições de recursos. A memória, a capacidade de processamento e o tempo de vida das baterias são limitados; as interfaces aéreas são relativamente estreitas e caras; existe também uma alta variação da presença ou desconexão (*online-offline*) do dispositivo na rede. Porém, a evolução das tecnologias de redes e dos dispositivos móveis diminui essas limitações, e deste modo surgiu a possibilidade de usar aplicações P2P também em dispositivos móveis [13] e [9].

Apesar das limitações, a capacidade de armazenamento em memória está aumentando continuamente. Cartões de memória com capacidade de até 4GB¹⁰, atualmente estão disponíveis no mercado. Isso deixa claro que o tamanho de memória não é um

⁸JXTA: plataforma para desenvolvimento de aplicações P2P. Maiores detalhes na seção de arquiteturas.

⁹JXME: extensão de JXTA que permite o desenvolvimento de aplicações P2P para dispositivos móveis. Maiores detalhes na seção de arquiteturas.

¹⁰Cartões de memória com capacidade de 4GB podem ser encontrados à venda no link: <http://froogle.google.com/froogle?q=MemoryStick+PRO+de+4+GB+%24&hl=en&lr=&sa=N&tab=ff&oi=froogler>

empecilho hoje em dia. Entretanto, vale a pena lembrar que a maioria dos celulares possui somente um pequeno chip de memória embutido.

O poder de processamento também está crescendo. Os processadores podem manipular com as várias larguras de banda no acesso sem fio e realizar processamentos que permitem ao usuário utilizar vídeo conferência e jogos cheios de interações [14].

O tempo de vida das baterias também cresce, mas em um ritmo menor que os dois itens anteriores, por isso continuará a ser o grande gargalo da tecnologia dos dispositivos móveis. Conseqüentemente, os usuários desses equipamentos móveis continuarão a estar menos *online*, se comparados aos dispositivos não móveis, em que a maioria das aplicações P2P roda hoje em dia. A redução do tempo *online* dos *peers* afeta extremamente o tempo de *download*, o que compromete a utilização de aplicações de compartilhamento de arquivos em redes P2P-Móveis [13]. Juntamente com a capacidade de bateria, as limitações da potência de transmissão fazem com que a largura de banda de *uplink*¹¹ seja significativamente mais cara que a de *downlink*¹².

A possibilidade de utilizar aplicações que permitem aos usuários uma grande interação é prometida pelas operadoras da 3G, no entanto, essas aplicações requerem um alto gasto da bateria, ou seja, o tempo disponível para utilizar os prometidos serviços da 3G será menor do que usar os serviços da 2G [14].

A escolha da linguagem ou API (*Application Programming Interface*) para programação de aplicações para dispositivos móveis também pode ser considerada como um obstáculo, porque ela pode apresentar restrições de acesso a certos recursos. Um exemplo é a API J2ME (*Java 2 Micro Edition*) CLDC/MIDP¹³ que não permite um componente externo acessar o sistema de arquivo dos celulares. Porém, esta prevenção criada pelas aplicações MIDP pode não permitir a transferência de arquivos entre os *peers* em um sistema de compartilhamento de arquivos. Outro exemplo é a API SuperWaba¹⁴ que não permite a implementação de um aplicativo P2P (cliente/servidor), uma vez que não disponibiliza a funcionalidade de abrir um *socket* para escutar o pedido de abertura de conexões. SuperWaba permite apenas que a aplicação solicite a abertura de conexões a uma máquina *desktop*, conseqüentemente só é possível desenvolver aplicações que se comportem como cliente. Algumas APIs de linguagem de programação, tal como Symbian C++, fornece quase acesso ilimitado para os recursos móveis [9].

2.3 Tecnologia de Acesso

As redes sem fio geram uma série de vantagens, entre elas estão: mobilidade, flexibilidade e custo. A mobilidade do usuário pode ser muito rápida e freqüente. No

¹¹ *Uplink*: termo técnico para a transmissão de dados no sentido do usuário para a rede ou provedor de serviço da Internet. Também designado por “canal de retorno”.

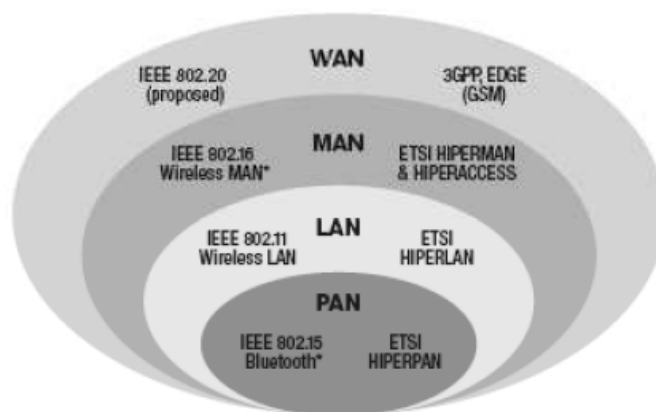
¹² *Downlink*: Termo técnico usado para definir a transmissão de dados na seqüência rede/operadora ou provedor de serviço/Internet ao usuário.

¹³ CLDC/MIDP - (*Connected Limited Device Configuration*)/ (*Mobile Information Device Profile*) fornecem um ambiente de execução completo para uma aplicação de J2ME que tem como alvo os dispositivos móveis, como celulares, *paggers* e PDA's.

¹⁴ SuperWaba é uma plataforma para desenvolvimento de software para aplicações cujo alvo são os PDAs. Maiores detalhes em <http://www.superwaba.com.br>

cenário sem fio, “rápida” significa que um *peer* pode se mover várias vezes durante uma sessão de comunicação, e “frequente” significa que existe uma alta probabilidade que um *peer* não seja encontrado em uma mesma localização. A maioria dos protocolos de descoberta P2P normalmente suporta a rara mobilidade, devido ao fornecimento de um conjunto de endereços *peer* para se conectar. No entanto, isto pode se tornar muito ineficiente se um *peer* tem de usar a lista de endereços frequentemente [10].

Existem diferentes tipos de redes sem fio que variam em tecnologia e aplicação, sendo possível classificá-las em quatro tipos: WPANs (*Wireless Personal Area Network*), WLANs (*Wireless Local Area Network*), WMANs (*Wireless Metropolitan Area Network*) e WWANs (*Wireless Wide Area Network*). A Figura 2 apresenta estes tipos de redes que serão exemplificadas nos próximos parágrafos [44] e [45].



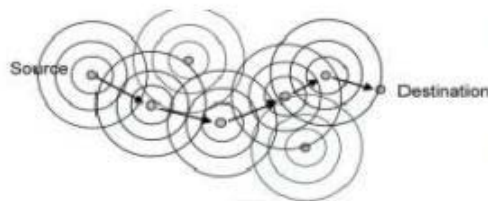
Fonte: referência [44]

Figura 2 Padronizações da tecnologia de redes sem fio

Bluetooth é um exemplo de tecnologia de redes WPANs as quais cobrem pequenas distâncias e oferecem baixas velocidades, se comparada a outras tecnologias sem fio.

A tecnologia 802.11, também conhecida como Wi-Fi, é um exemplo de rede WLAN a qual oferece grande flexibilidade para seus usuários, principalmente os que utilizam laptops e PDAs. Este tipo de rede define duas formas de comunicação: ponto a ponto (*ad-hoc*) ou modo infra-estrutura.

O modo *ad-hoc* permite que um pequeno grupo de máquinas se comunique diretamente, sem a necessidade de um *Access Point* (AP ou Ponto de Acesso) e de uma rede física para conectar as estações. Neste modo todas as estações são conectadas diretamente umas às outras. Outras características da rede *ad-hoc* incluem um modo de operação ponto a ponto distribuído, roteamento *multi-hop*, e mudanças relativamente frequentes na concentração dos nós da rede. A responsabilidade por organizar e controlar a rede é distribuída entre os próprios terminais. Em redes *ad-hoc*, alguns pares de terminais não são capazes de se comunicar diretamente entre si, então alguma forma de re-transmissão de mensagens, através de vários saltos, é necessária, para que estes pacotes sejam entregues ao seu destino, como pode ser visto na Figura 3.



Fonte: referência [11]

Figura 3 Percurso de uma mensagem em uma rede *ad-hoc*

No modo de infra-estrutura, a rede possui pontos de acessos fixos que conectam a rede sem fio à rede convencional e estabelecem a comunicação entre os diversos clientes. Dessa forma, uma rede sem fio também pode ser caracterizada pela existência de nós fixos (tal como estações base e pontos de acesso) que servem a uma área ou a célula particular. Desse modo, a rede pode ser formada por *peers* com alta mobilidade e por *peers* fixos que possuem localização fixa [10] e [11].

A rede WMAN se aplica a regiões urbanas. A tecnologia 802.16, também denominada Wi-Max está classificada neste tipo de rede que oferece uma cobertura geográfica consideravelmente maior que as WLANs, chegando a distâncias de até 50 km.

As redes WWAN são redes com grande dispersão geográfica, voltadas para aplicações móveis que utilizem telefones celulares e PDAs. Elas apresentam um crescente uso de conexões de banda larga e serviços de comutação de pacote de dados para permitir a transmissão de e-mails, textos, imagens, som e vídeo, com a mesma qualidade e velocidade que os dispositivos ligados por fios. As seguintes tecnologias se enquadram neste tipo de rede: GPRS, EDGE (*Enhanced Data rates for Global Evolution*) e UMTS.

Neste contexto, atualmente um dispositivo pode ter interfaces conectadas a vários tipos de redes (possivelmente ao mesmo tempo, desde que estas redes possam se sobrepor). Eles também podem ter múltiplos endereços IP (*multihoming*) [10].

Essas tecnologias de acesso diferem em termos de interface aérea, capacidade de QoS (*Quality of Service*), disponibilidade da taxa de transferência e mecanismos de transporte na rede [13]. Comparações e detalhes relacionados a taxas de transferência serão apresentados na seção 2.3.1.

As duas principais restrições da interface aérea são: uma largura de banda relativamente baixa, e alta latência. Por isso, é essencial reduzir o *overhead* da sinalização até que seja possível registrar uma *performance* aceitável. Caso contrário, as aplicações com alto tráfego de sinalização são consideradas muito caras, principalmente, em rede celular. Uma maneira de otimizar a comunicação P2P seria utilizar estratégias como *caching* para reduzir a banda e o tráfego de sinalização. Deve-se diminuir o tráfego direto entre os *peers* e entre os dispositivos móveis, porque na comunicação P2P entre dois dispositivos móveis, os recursos da interface aérea é utilizada duas vezes, se comparada com as transmissões entre o dispositivo móvel e a rede fixa [13] e [15]. No entanto, *caching* ainda não é uma abordagem utilizada em rede P2P.

2.3.1 Taxa de Transferência

A interface aérea é comumente vista como o gargalo em sistemas de comunicação sem fio, embora o recente desenvolvimento de tecnologias de acesso a dados tem aumentado a velocidade de 9,6Kbps até 54Mbps. A típica taxa de transferência é apresentada na Tabela 1[13]e [14].

Tabela 1 Taxa de transferência das tecnologias de acesso

Tecnologia de Acesso	Taxa de Transferência
HSCSD	Acima de 43,2 Kbps
GPRS	43,2 Kbps
EDGE (4+2)	236KBps DL(<i>downlink</i>)/ 118KBps UL(<i>uplink</i>)
WCDMA	Acima de 384Kbps DL/ 128Kbps UL
HSDPA/HSUPA	Máximo 13,3Mbps, esperado 3,36Mbps / máximo 5,671Mbps, esperado 1,46Mbps
WLAN	Máximo 112Mbps, tipicamente 11 Mbps.

Fonte: referência [14]

A taxa máxima de dados nos serviços *peer-to-peer* tem sido calculada levando em consideração a conexão *uplink* do conteúdo disponibilizado e a conexão *downlink* do conteúdo consumido, na qual a menor taxa de transferência de dados será a taxa usada na comunicação entre os *peers*. Um exemplo é ambos os *peers* estiverem usando HSDPA e HSUPA como tecnologia de acesso. Neste caso a HSUPA representa o gargalo da tecnologia de acesso, uma vez que a taxa máxima de dados com que os móveis podem trocar é de HSUPA, ou seja, a taxa de dados é de 1,46 Mbps [14].

Uma observação interessante é que o tráfego de redes P2P tradicionais é caracterizado pelo uso simétrico das tecnologias de acesso, em termos de consumo de banda. Isso é diferente em relação à Web que fornece consumo de banda assimétrico por natureza. Devido ao interesse em diminuir o tráfego direto entre *peers* móveis e evitar um cenário como o do parágrafo anterior, deve-se ser analisada a simetria ou assimetria em rede P2P-Móvel com o uso de *caching*.

Ainda em relação à Tabela 1, pode-se afirmar que o *downloading* em tempo real pode ser facilitado apenas no caso da WCDMA e suas extensões HSDPA/HSUPA, no entanto a tecnologia mais adequada para aplicações de compartilhamento de arquivos é a WLAN. O trabalho na tecnologia de acesso a dados 3G UTRAN (*Terrestrial Radio Access Network*) está avançando em 3GPP (*Third Generation Partnership Project*). Em um futuro próximo, uma tecnologia de alta capacidade é esperada, denominada Wi-Max [14].

O padrão IEEE 802.16 (Wi-Max) é caracterizado pela arquitetura totalmente centralizada que suporta uma topologia ponto-multiponto em que cada estação base se comunica com até centenas de estações estacionárias de usuários. Um ponto diferencial do padrão 802.16 é que a interface aérea foi projetada para transmitir dados ou tráfego multimídia que necessitam de alto suporte de qualidade de serviço. O padrão 802.16 é completamente orientado a conexões a fim de garantir qualidade de serviço para comunicação de telefonia e multimídia, as quais não admitem atrasos.

Ao contrário do IEEE 802.11 (Wi-Fi), o padrão IEEE 802.16 utiliza um espectro variável, ou seja, emprega um sistema de modulação adaptativa, com a utilização de três esquemas de modulação diferentes: QAM-64, QAM-16 e QPSK. Quando a intensidade do sinal na banda milimétrica cai com o aumento da distância da estação base, o esquema de modulação é modificado dependendo da distância que a estação do assinante se encontra em relação à estação base. Para assinantes próximos é usado o QAM-64, com 6 bits/ baud, no caso de assinantes situados a uma distância média é usado o QAM-16 com 4 bits/ baud, e para assinantes distantes é usado o QPSK com 2 bits/ baud. Os esquemas QAM-16 e QPSK permitem um aumento no alcance do sinal, mas trazem, como consequência, a redução da vazão. Na especificação 802.16, são definidos canais de 20MHz, 25MHz (tipicamente dos EUA) e 28MHz (tipicamente europeu), conforme mostrado na Tabela 2 que mostra quanto mais distante estiver o assinante em relação à estação base, mais baixa será a taxa de dados. [44] e [45].

Tabela 2 Canais utilizados no padrão 802.16

Largura de banda do canal (MHz)	Taxa de transferência (Mbps) QPSK	Taxa de transferência (Mbps) 16-QAM	Taxa de transferência (Mbps) 64-QAM
20	32	64	96
25	40	80	120
28	44.8	89.6	134.4

Fonte: referência [44]

Apesar das vantagens prometidas pelo padrão 802.16, este possui certa semelhança com a estrutura da rede celular. Essas duas possuem topologia muito centralizada e hierárquica. Portanto, o padrão 802.16 também apresenta um forte contraste à proposta da maioria das topologias das redes *overlay* para sistemas P2P.

Nos sistemas da terceira geração (3G), como UMTS e a tecnologia HSDPA, o custo da transmissão de dados sobre a interface aérea é geralmente alto, se comparado com redes fixas. Isso atualmente é verdade para os sistemas da segunda geração (2G) e 2,5G, como GSM/GPRS, que teoricamente possuem taxas de transferência de até 171Kbps e tipicamente registram taxas de transferência entre 28 e 50Kbps. Além disso, a média dos tempos de resposta é significativamente mais alta do que em sistemas com fio devido ao grande *overhead* dos protocolos e esquemas de correção de erros complexos, acarretando em uma baixa performance especialmente do TCP [13]. O uso da transmissão de dados com TCP sofre retransmissões de pacotes devido às perdas de pacotes [15]. Estes resultados também foram confirmados em [13] pela medição do eDonkey utilizando GPRS.

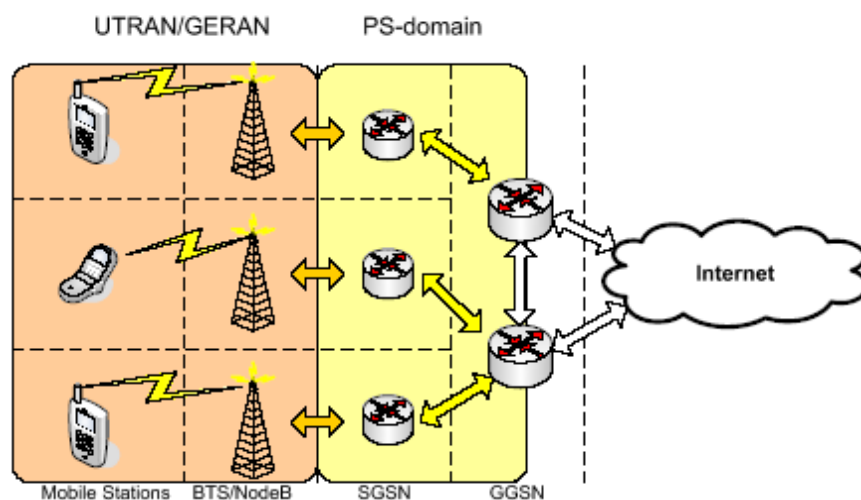
O GPRS é uma tecnologia de comutação de pacotes, nela os canais físicos podem ser compartilhados entre diferentes usuários móveis e somente é atribuído um canal físico quando for necessário transmitir ou receber dados. Por isso, as taxas de dados usando GPRS dependem do número total da relação voz/dados dos usuários em uma célula e do suporte a taxa de dados de uma estação móvel (*Mobile Station* - SM). Na rede GSM/GPRS, a comutação de circuitos (voz) e a comutação de pacotes dos usuários existem em paralelo. A combinação dos canais *uplink/downlink* depende do terminal móvel e é referenciada como uma classe *multi-slot*. Consequentemente, eles competem

por recursos existentes na interface aérea. No entanto, a alocação dinâmica da largura de banda é baseada em conceder prioridade ao tráfego de voz, incluindo a opção de parar as transmissões de dados em favor a chamadas de voz [15].

2.3.2 Estrutura da Rede Celular

Em geral, a estrutura de um sistema de comunicação móvel é projetada de forma hierárquica. Para GPRS ou UMTS, o tráfego do fluxo de dados de cada dispositivo móvel atravessa a estrutura da rede, da UTRAN da UMTS através do domínio da comutação de pacote e volta [13]. Até no caso de dois MSs trocando dados e presentes na mesma GGSN, o caminho entre os MS e a GGSN é sempre percorrido duas vezes [15].

Na GGSN, os *hosts* dos dispositivos móveis obtêm determinado identificador (chamado nesta seção 2.3.2 de endereço IP), semelhante ao endereço IP de uma máquina sob NAT. Então, a GGSN é a interface (ou roteador especial) para a Internet e para os outros dispositivos móveis no domínio móvel, fazendo também o ponto na estrutura da rede onde todo tráfego de pacotes é concentrado, visto na Figura 4. Depois de passar no GGSN, a operadora irá rotear o dado para um celular de acordo com sua especificação [16]. Semelhante à Figura 4, geralmente, existem vários GGSN na estrutura da rede, cada um servindo como um *gateway* para uma grande parte da rede móvel. Esta hierarquia, topologia muito centralizada, é um forte contraste a proposta da maioria das topologias das redes *overlay* para sistemas P2P [13].



Fonte: referência [13]

Figura 4 Estrutura simplificada da rede para transporte de pacotes de dados

Dessa forma, para transferir um dado da Internet para o celular, os dados passam por um túnel que é finalizado no GGSN, *gateway* GPRS [16]. Uma das principais consequências dessa estrutura é o alto tempo de atraso na transferência de um pacote IP em um dado caminho entre dois terminais [15].

Essa estrutura possui várias limitações para aplicações P2P. Primeiramente, o GGSN pode variar durante a transferência de dados de um terminal móvel em movimento e principalmente o SGSN (*Serving GPRS Support Node*). Este último irá variar mais, porque atende a uma menor área de cobertura. Isto causa a mudança do endereço IP

dos terminais móveis e como consequência não é possível manter conexões ativas durante movimento. Uma forma de manter conexões ativas, seria implementar esta característica na rede, por exemplo, com SIP, que cria, modifica e encerra conexões entre usuários. Com este protocolo a operadora sempre saberá qual é a localização atual (endereço IP) do usuário e então a conexão ativa poderá ser mantida [16].

No ambiente 2/2,5G existe a restrição de que dados e voz não podem ser enviadas simultaneamente, a menos que o telefone móvel pertença na classe A ou a operadora adicione novas características a rede. A primeira opção, telefones móveis na classe A, exigirá da CPU muito processamento. A segunda opção, modificações na rede, exigirá um alto investimento e atualmente não parecem existir operadoras que concordem em investir nesta tecnologia [16].

Atualmente as operadoras não permitem que os terminais móveis ajam como terminais independentes com um endereço IP, elas não dão suporte a conexões fim-a-fim e possuem total controle sobre a rede, incluindo portas TCP/IP, por isso as operadoras só permitem enviar dados através de porta que sempre estão abertas. Dois exemplos são as operadoras Sonera e Elisa (maiores operadoras da Finlândia) que controlam cada dado transmitido pelas aplicações para um celular de acordo com [16].

Além disso, outros dos mais importantes requisitos operacionais das operadoras de redes de celulares são manter o controle da cobrança dos serviços fornecidos e do tráfego em seu próprio domínio, para evitar o custo, devido ao tráfego do interdomínio. Isto é verdade para ambas operadoras, linha fixa e celular. Se o P2P-Móvel é integrado dentro da estrutura de serviços, será, então, necessário fornecer meios para controlar e para debitar. Por outro lado, o mecanismo de controle para um sistema P2P-Móvel deve ser cuidadosamente selecionado, a fim evitar a total degeneração do sistema centralizado. Os mecanismos do controle não devem alterar conceitos fundamentais do P2P, tais como a descentralização. O modelo do negócio usado para debitar também deve cumprir com aplicações P2P, um exemplo seria recompensar os usuários que compartilham. Por outro lado, um sistema P2P-Móvel pode ser beneficiado pela existência da infra-estrutura de um sistema de comunicação móvel. Isso porque a rede fornece conhecimentos de localização, status *online-offline* e o contrato de serviços dos usuários de celulares, que pode ser utilizado para sinalizar *overhead* e o crescimento da qualidade de serviço [13].

2.4 Aspectos Econômicos

Nas redes de celular, a questão financeira ainda é um problema para aplicações P2P,. Porém existe operadora demonstrando o interesse em oferecer serviços *peer-to-peer* em ambientes móveis. Um exemplo é a operadora coreana SK Telecom que decidiu oferecer serviços P2P baseados em compartilhamento de conteúdo. Estes permitirão aos usuários trocar figuras, músicas e vídeo, sem levar em consideração se o conteúdo é *copyright* ou não. Dessa forma, nesta seção, serão ressaltados problemas relacionados ao preço e regulamentação do conteúdo compartilhado.

Primeiramente, o preço dos serviços está entrelaçado com a viabilidade dos serviços P2P. De acordo com um simples cálculo da Tabela 3, o esquema de preços dos serviços de dados oferecido pelas operadoras 2G/3G não permite a total adoção dos serviços de compartilhamento de arquivos. Na maioria dos casos, o custo da transmissão de dados

de um arquivo de música entre dois celulares é mais caro, do que o preço de um serviço de conteúdo da internet que oferece um arquivo de música por volta de \$1. A menor taxa, fornecida pela operadora PTC 2G/3G para clientes poloneses, é 50% do preço do conteúdo anterior, considerando um modelo muito otimista, que provavelmente adequará consumidores de alta capacidade de consumo [14].

Tabela 3 Estimativa de custos de conteúdo download

Serviço / Custo do download em Euro	Custo por Mês	Custo por MB sobre o limite	Custo de download de arquivo MP3 (3MB)*	Custo de download de foto (60kb)*
PTC Poland: Blue connect GPRS/ EDGE/ 3G/ WLAN	Taxa do plano: 23.6e	2GB garantido	0.472e	0.0009e
Three Sweden: NetConnect Free GPRS/3G	87,1e/GB	0.4e	1.742e	0.0033e
Three Sweden: NetConnect Flex GPRS/3G	5.3e	0-25MB 0.9e 26-100MB 0.7e 100MB<0.4e	2.506	0.0048e
VodaFone Swede: Data GPRS/3G	61e/150MB	0.7e (mudança através de 1kb)	3.32e	0.0064e
Orange UK: World Access 400 GPRS/3G	400MB/77.3e	0.73e	1.314e	0.0025e
Orange UK: World Access Max GPRS/3G	1000MB/128.9 e	(política de uso justo)	2.19e	0.0042e
* Considerando: usuários baixam 300MB de dados por mês (100MB de dados sobre P2P, 50MB navegando pela WEB, 50MB e-mail). A taxa de dados P2P é calculada considerando a saída de dados (dados produzidos) e tráfego de entrada (conteúdo consumido). A sinalização P2P e overhead dos protocolos não são incluindo.				

Fonte: referência [14]

Uma das visões futuras para esta questão de custo, é que provavelmente, as operadoras poderão oferecer “pacotes” para serviços P2P com serviços de acesso a dados, em vez de baixar os preços, a fim de estimular a adoção do serviço. Atualmente, de acordo com a análise de [14], a maioria das operadoras, consideradas nas Tabela 3, permite livre acesso para seus serviços de conteúdo através de WAP (Wireless Applications Protocol).

Outro problema é a regulamentação. Frequentemente, os celulares possuem limitações para enviar uma música (*ring-tone*), símbolos e figuras, mas não devido à tecnologia. O motivo para estas limitações está relacionado a questões econômicas das operadoras, que estão interessadas em proteger as suas vendas de músicas (*ring-tone*), símbolos e figuras. Isso dificulta ou mesmo impossibilita a existência de compartilhamento de arquivos em redes móveis [16]. A indústria da música também é uma barreira para o fornecimento de compartilhamento de arquivos em rede de celular, sendo por isso realizados trabalhos, com a finalidade de proteger a propriedade intelectual. Um dos resultados de estudos para proteger a propriedade intelectual é o DRM (*Digital Rights Management*), que é um conceito avançado para esta proteção. O DRM permite que os varejistas possam facilmente determinar as condições da transação, tais como o pagamento por preço único, pelo pacote, ou para executar um software ou uma música.

2.5 Considerações Finais

Neste capítulo, foi possível observar que as dificuldades encontradas em sistemas P2P tradicionais também podem estar presentes em P2P-Móvel. Este fato foi exemplificado através da dificuldade de permitir a interoperabilidade entre protocolos P2P e da alcançabilidade de *peers* sob NAT e/ou *firewall*. Também foram identificados desafios devido à escolha da linguagem e a API para implementação de aplicações P2P-Móvel, e às restrições quanto à memória, ao processamento e à bateria dos diversos dispositivos móveis.

A característica de mobilidades dos *peers* móvel, proporcionada pelas redes sem fio, foi apresentada como outro desafio relacionado à tecnologia de acesso ou estrutura da rede. Um exemplo é a presença da característica de forte centralização na rede Wi-Max e na estrutura da rede celular, porque representando o oposto da proposta das topologias das redes P2P.

Foi identificado que em uma rede P2P-Móvel pode conter uma variedade de tecnologias de acesso. No entanto, a menor taxa de transferência determina o valor máximo para transferência de dados entre dois *peers*. Dessa forma, estes dois fatos motivam a utilização de assimetria na comunicação P2P por meio de *caching*.

Por fim, as questões econômicas foram apresentadas como relevantes em relação à viabilidade de P2P-Móvel. Observou-se que os valores cobrados pelas operadoras e a questão de propriedade intelectual resultam em grandes barreiras para as aplicações P2P-Móvel, principalmente na estrutura de rede celular.

No próximo capítulo, serão apresentados tipos de aplicações, exemplificados por meio de aplicativos, no contexto de redes P2P e dispositivos móveis.

3 Aplicações

Este capítulo tem como principal objetivo apresentar aplicações que utilizam arquiteturas *peer-to-peer* e são executadas em dispositivos móveis. O capítulo também apresentará aplicativos que podem ser classificados como aplicações P2P-Móvel, porque eles utilizam em algum momento a comunicação direta com a intenção de compartilhar alguma informação.

3.1 Monitorando Equipamento de uma Rede P2P

eMule é um aplicativo que utiliza uma rede *peer-to-peer* para compartilhar arquivos. Este *software* deve ser executado em computadores pessoais e age como um P2P-Client. Isso porque o eMule permite que o usuário possa está conectado diretamente a vários outros usuários para baixar e disponibilizar músicas, filmes entre outros recursos. A partir do crescimento do numero de pessoas que acessam uma rede sem fio com celular, existiu a motivação de criar uma aplicação (MobileMule¹⁵) para controlar de maneira remota uma máquina com eMule executando.

O MobileMule permite que qualquer dispositivo móvel possa obter informações de equipamentos participantes de uma rede *peer-to-peer*. No entanto, esta aplicação não é um P2P-Client, mas sim um exemplo de controle remoto de um computador, onde o celular é o controle.

O MobileMule conecta-se através de redes de telecomunicações, como um cliente para um servidor eMule em um computador na porta 80. O MobileMule permite que o usuário solicite o início de uma busca, selecione arquivos para realizar *download* no computador, checar status dos arquivos que estão sendo baixados, checar propriedade dos arquivos (nome, tamanho do arquivo e quantidade de fontes, que corresponde ao número de usuários compartilhando este arquivo) e realizar o *preview* que permite visualizar o primeiro frame de um arquivo de vídeo. Também estão disponíveis os comandos de fechar a aplicação eMule, desligar e ligar o computador remotamente.

Quando se solicita a busca de um arquivo, no máximo, são retornados quinze resultados ordenados pela maior quantidade de fontes disponíveis para o arquivo procurado. O MobileMule também mostra o número total de arquivos disponíveis para a determinada busca, no formato de dois dígitos, ou seja, mesmo que existam 500 arquivos, a interface gráfica do *software* apresentará 99 [16].

Existe uma proposta de modificação da rede eDonkey para permitir a participação de dispositivos móveis no compartilhamento de arquivos, ou seja, os dispositivos móveis seriam mais do um controle remoto, eles também poderiam agir como *peers* que disponibilizam recursos. Essa e outras propostas de arquiteturas para compartilhamento de arquivos estão na seção 4.4.2.

¹⁵ O *software* MobileMule está disponível em <http://mobil.emule-project.net/index.php?page=download&language=en>

3.2 Voz sobre IP em uma Rede *Peer-to-Peer*

O Skype foi criado por Niklas Zennström e por Janus Friis, fundadores do KaZaA. Este programa, baseado no princípio P2P, fornece serviço de telefonia na internet – VoIP (*Voice over Internet Protocol*). Todas as aplicações do Skype agem como *peers* em uma rede P2P distribuída. O único elemento central do sistema é o servidor de *login*.

A tecnologia VoIP, permite que a voz seja codificada em tempo real e enviada para um *peer* que decodifica o *stream* de áudio e envia para a saída de som. Todos os *peers* participantes em uma sessão VoIP têm a capacidade de simultaneamente de enviar e receber *stream* de áudio em tempo real [18].

Inicialmente o Skype fornecia aplicativos para *desktops* e em 2005 lançou o primeiro produto para dispositivo móvel, denominado PocketSkype. Esse primeiro produto atende a *PocketPC* com interface aérea WLAN. A empresa também fez parceria com a Motorola para desenvolver novas aplicações Skype, a fim de atender aos usuários dos dispositivos da Motorola e incluir uma variedade de dispositivo com tecnologia de acesso *Bluetooth* [14].

Em relação à performance do PocketSkype, é necessária uma conexão rápida. Em conexões ADSL de 256kbps, através de Wi-Fi ou *Ethernet*, as denominadas apresentam pequenos e eventuais cortes, nada que recrimine o uso, mesmo nesta velocidade a qualidade é razoável. Em conexões compartilhadas por *ActiveSync* via USB ou *Bluetooth*, foi possível estabelecer chamadas, no entanto as conversas apresentavam muitas interrupções, tornando inviável qualquer diálogo. Via GPRS, não é recomendado o seu uso [17].

XVoice é uma outra aplicação que utiliza uma arquitetura P2P e VoIP. Ela foi desenvolvida pelo grupo de trabalho em computação colaborativa (GTP2P), que tem o objetivo de fornecer este aplicativo para equipamentos restritos, tais como *PocketPC*.

3.3 Compartilhamento de Arquivos

A empresa NewBay lançou, em 26 de outubro de 2004, o software FoneShare, a primeira aplicação para compartilhamento de conteúdo para dispositivos móveis. Usando o FoneShare, os usuários podem procurar, comprar, armazenar, compartilhar e recomendar conteúdo (tais como *ringtones*, jogos, música, papel de parede e *trailers*) para outros usuários de dispositivos móveis.

Esta aplicação foi construída conceitualmente através de comunidades. O sistema tem o objetivo de incentivar os usuários móveis, que estão satisfeitos com uma compra, a recomendar conteúdos a seus grupos de amigos. Para esta aplicação, também existe critério de reputação e recompensa em que os usuários do aplicativo FoneShare podem formar identidades *online*, comprar conteúdo e sugerir compras a outros usuários do FoneShare. A cada usuário são concedidos “os pontos de mérito” baseados em sua atividade, e os super-usuários podem ser recompensados com conteúdos ou descontos em serviços.

Dessa forma, a aplicação FoneShare oferece a operadoras de dispositivos móveis o aumento das vendas de conteúdo, das vendas dos serviços de mensagens e redução nos custos de marketing [14] e [19].

3.4 Blog

Algumas aplicações P2P permitem que o usuário possa escrever diretamente em páginas WEB denominadas de “Blog”, e nelas também disponibilizar arquivos, tais como MP3. Os usuários podem criar seus Blogs de acordo com um tema. Por exemplo, um Blog privativo para uma equipe de trabalho discutir projetos e apresentar soluções. Um sistema Blogger é encontrado em (www.blogger.com.br) [20].

A empresa NewBay também desenvolveu uma aplicação do tipo Blog, denominada PhoneBlog, que permite, aos usuários de equipamentos móveis, colocar fotografias, áudio, texto e vídeo em seu Blog ou álbum. Todo esse conteúdo é imediatamente visível usando WAP ou um *web browser* padrão. PhoneBlog tem sido oferecido por algumas das operadoras T-Mobile EUA, O2 Ireland e SETAR NV em Aruba sob o serviço denominado Waw!Blog. Em média, são enviadas quatro mensagens MMS por o mês, utilizando esse serviço. O serviço tem gerado cinco milhões de mensagens MMS em um período de seis meses e a média do tráfego WAP é de 4.8 MB/ativação usuário/mês.

Outra aplicação para dispositivos móveis do tipo Blog criada pela Futurice é a denominada Futublog. A Futublog tem a finalidade de permitir, aos usuários de celulares, compartilhar e armazenar fotos que foram tiradas pela câmera do seu aparelho.

Semelhante ao que foi informado no início desta seção, o desenvolvimento de aplicações do tipo Blog para dispositivos móveis, também poderia utilizar o conceito de P2P em que os *peers* compartilham as informações, que posteriormente serão disponibilizadas no formato Blog, geralmente apenas HTML [14].

3.5 Troca de Mensagens

A possibilidade de poder observar as pessoas entrando na rede e enviar uma mensagem em tempo real, tem tornado as aplicações de mensagem instantânea (IM – *Instant Messaging*) uma das mais populares da Internet.

As conhecidas aplicações para *desktop* (AIM¹⁶, ICQ¹⁷, MSN¹⁸ e Yahoo¹⁹) atualmente estão disponíveis para diversos tipos de dispositivos móveis.

Muitas arquiteturas, como JXME, X-Peer e outras, que permitem P2P-Móvel, também disponibilizam simples aplicativos IM para realizar testes da arquitetura.

¹⁶ Aplicativo AIM para dispositivo móvel disponível em <http://www.aol.ca/aim/mobile/index.adp>

¹⁷ Software ICP para dispositivo móvel disponível em <http://www.icq.com/channels/wireless/>

¹⁸ Aplicativo MSN para dispositivo móvel disponível para dispositivos móveis em <http://mobile.msn.com/>

¹⁹ Aplicação Yahoo para dispositivos móveis disponível em <http://mobile.yahoo.com/messenger>

3.6 Jogos

As aplicações de Jogos com vários usuários também podem extrair vantagens da arquitetura P2P, que permitirá encontrar esses usuários e estabelecer comunicações P2P para troca de mensagens durante o jogo, o que é muito comum em jogo em rede.

Em Setembro de 2003, a Nokia disponibilizou uma plataforma denominada N-Gage para jogos multi-usuário em dispositivos móveis. A plataforma permite que os usuários possam jogar com vários usuários conectados localmente via *Bluetooth* e/ou remotamente sobre uma rede móvel. Leisure Suit Larry é um exemplo de jogo para esta plataforma. Este jogo oferece a oportunidade de testar capacidades sociais entre os jogadores via *Bluetooth*. B'ngo é um celular com funcionalidade de *handheld* e possibilita jogos utilizando *Bluetooth* com capacidade máxima de oito jogadores [21].

A vantagem de se utilizar arquitetura P2P para construção de jogos pode ser observada na Tabela 4, em que P2P se enquadra tanto em jogos com pouco jogadores, como em jogos com vários jogadores.

Tabela 4Jogadores X Arquitetura

Número de Jogadores	Arquitetura
Pequeno 1-16	P2P Puro
Grande 16-200	Cliente/Servidor e P2P Híbrido
Muito grande acima de 200	P2P Híbrido

Fonte: referência [21].

No entanto, não foram encontradas propostas de arquiteturas P2P, específicas para o desenvolvimento de jogos.

3.7 Ferramenta Educacional com Entretenimento

O uso de dispositivos móveis, atualmente, não atende apenas os adultos. Existem aparelhos no mercado dedicados a atender crianças e adolescentes. Atualmente é possível comprar um celular com teclados específicos para realizar chamadas programadas para o pai e a mãe da criança. As Figura 5 e Figura 6 mostram exemplos destes aparelhos.



Figura 5 Aparelho Firefly da empresa GadgetMadness
(<http://www.gadgetmadness.com>)

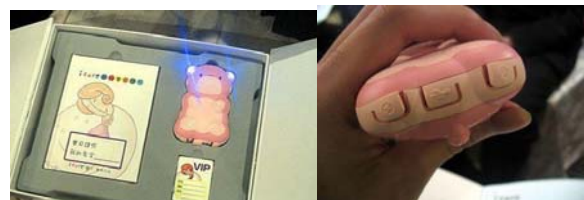


Figura 6 Aparelho i-Care da empresa Sogi
(<http://www.sogi.com.tw>)

Devido esta disponibilidade de acesso a dispositivos móveis por crianças e adolescente, surge o interesse em desenvolver softwares educativos para os aparelhos móveis. O protótipo FolkMusic é um exemplo de ferramenta que mescla diversão e aprendizado. Essa aplicação explora os fundamentos da teoria dos conjuntos (incluindo elementos, conjuntos, uniões e interseções de diferentes conjuntos), onde os usuários que estão em uma determinada localização representam um conjunto que contém elementos (por exemplo, músicas MP3) armazenados em um dispositivo móvel (por exemplo, *smartphone* ou PDA). Quando duas pessoas se encontram, a união de dois conjuntos pode ser realizada. Além disso, a interseção desses dois conjuntos indica quais os arquivos MP3 eles possuem em comum. Também se pode considerar como conjunto, membros de uma comunidade, dispositivos móveis de uma pessoa, um local, ou uma posição geográfica. Isso significa que um lugar pode ser carregado com arquivos de música, os elementos música podem ser posicionados geograficamente.

A outra funcionalidade do FolkMusic é permitir, ao mesmo tempo, a diversão durante o aprendizado, porque após encontrar os arquivos de músicas, elas podem ser escutadas, antes mesmo de serem completamente baixadas para o dispositivo móvel.

A versão atual desse protótipo foi implementada utilizando a arquitetura *peer-to-peer*, com possibilidade de uso em rede *ad-hoc*. O código está disponível para *Laptop* com tecnologia de acesso WLAN (IEEE 802.11b) e um equipamento receptor GPS para traçar a posição geográfica e associar os arquivos de músicas com a localização. Para permitir que os usuários explorem a teoria dos conjuntos, enquanto escutam música, a interface gráfica de busca de MP3 foi desenvolvida baseada na teoria dos conjuntos como pode ser vista na Figura 7. E tem como trabalho futuro porta o código para *PocketPC* [24].

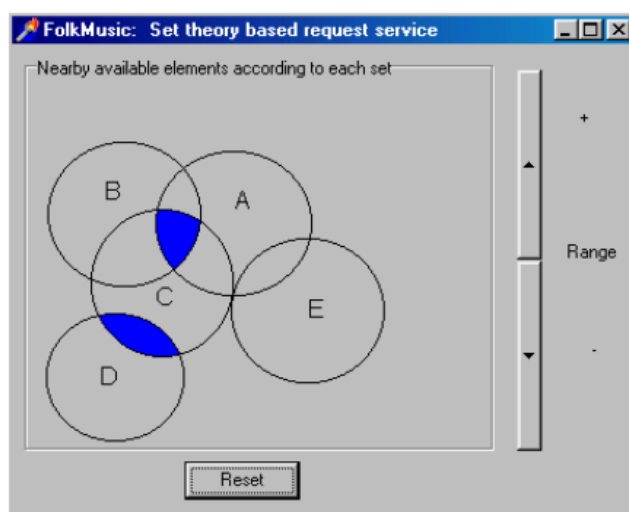


Figura 7 Interface gráfica baseada em teoria dos conjuntos do software FolkMusic

3.8 Considerações Finais

Neste capítulo, foram apresentadas diferentes aplicações em termos de requisitos de banda e de qualidade de serviço em geral. Esses diversos aplicativos estão disponíveis não só para equipamento com baixa restrição de hardware, mas também para dispositivos com alta restrição de recursos.

Também foi identificada uma tendência em portar aplicações dos tradicionais sistemas P2P para ambientes móveis, e um exemplo disto, é o desejo do grupo GT-P2P em disponibilizar a aplicação XVoice para dispositivos móveis.

Esse interesse geral em desenvolver aplicações P2P para ambientes móveis, juntamente com os desafios abordados no capítulo 2, motivam o estudo aprofundado dos sistemas P2P-Móvel. Assim como, é possível identificar a importância de uma avaliação sobre a viabilidade destas aplicações.

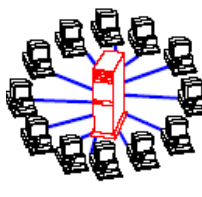
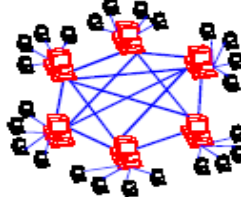
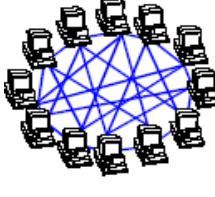
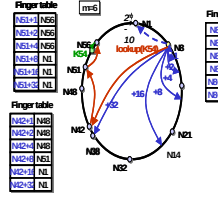
No próximo capítulo, será apresentado um estudo detalhado de arquiteturas para dar suporte à construção dessas aplicações.

4 Arquiteturas P2P

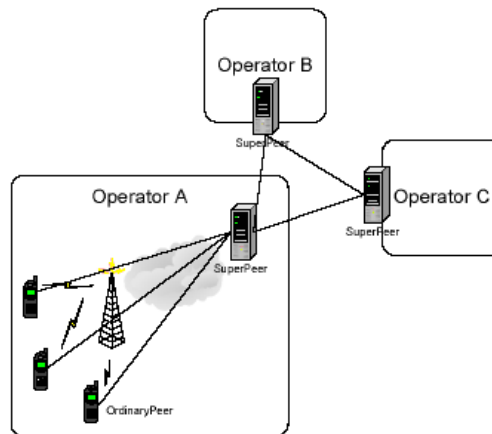
Os sistemas P2P tradicionais podem ser implementados de maneira híbrida ou totalmente descentralizada (sistema puro). Um sistema P2P tradicional puro se subdivide em sistemas puros estruturados e não estruturados, que se diferenciam, respectivamente, pela capacidade de recuperar todas as informações disponíveis na rede ou apenas parte delas. Exemplos de sistemas estruturados são os sistemas baseados em tabelas *hash* distribuídas (DHT – *Distributed Hash Tables*), como Chord [1], Pastry [2], Tapestry [25] e CAN [26]. O exemplo mais característico de sistema não estruturado é o Gnutella.

Sistemas híbridos também podem ser subdivididos em totalmente centralizados e hierárquicos. Os 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. Os sistemas híbridos hierárquicos possuem nós especiais (denominados super-nós, *super-peers* ou refletores), que servem como servidores para um pequeno grupo de nós. A comunicação entre esses super-nós, por sua vez, é realizada através de um mecanismo dependente de implementação. Por exemplo, os super-nós do KaZaA se comunicam entre si de maneira totalmente não estruturada, o que dificulta a recuperação de todas as informações existentes na rede [27]. O resumo destes conceitos e comparação com a arquitetura cliente/servidor está apresentado na tabela 4.

Tabela 5 Resumo das características de redes Cliente/Servidor e Peer-to-Peer

Cliente-Servidor	Peer-to-Peer			
1. Servidor é uma entidade central e só ele fornece serviços e conteúdos. -> Rede gerenciada pelo servidor. 2. Servidor tem sistema com alto desempenho. 3. Cliente possui sistema com baixo desempenho Exemplo: WEB	1. Recursos compartilhados entre os <i>peers</i>. 2. Recursos podem ser acessados diretamente de outros peers. 3. <i>Peer</i> é ao mesmo tempo provedor e solicitador. (conceito de <i>Servent</i>).			
	P2P Híbrido		P2P Puro	
	P2P Centralizada	P2P Hierárquico	Não Estruturado	Estruturada Baseada em DHT
	1. Inclui todas as características de P2P. 2. É necessária uma entidade central para fornecer serviços. 3. A entidade central é um tipo de banco de dados de índice/grupo Exemplo: Napster	1. Inclui todas as características de P2P. 2. Qualquer entidade terminal pode ser removida sem perdas de funcionalidade. 3. -> Entidade central dinâmica Exemplo: JXTA	1. Inclui todas as características de P2P. 2. Qualquer entidade terminal pode ser removida sem perdas de funcionalidade. 3. -> Não existe entidade central. Exemplo: Gnutella 0.4 e Freenet.	1. Inclui todas as características de P2P. 2. Qualquer entidade terminal pode ser removida sem perdas de funcionalidade. 3. -> Não existe entidade central. 4. Conexões na <i>overlay</i> são fixas. Exemplo: Chord e CAN.
				

Neste contexto, essas classificações de P2P, também podem ser aplicadas a P2P-Móvel. Em geral, a arquitetura híbrida hierárquica exemplificada na Figura 8 e na Tabela 5 tem se apresentado como a mais eficiente em termos de uso de recursos móveis e controle, uma vez que o uso da arquitetura híbrida centralizada não é eficiente quando os usuários P2P estão distribuídos em diferentes operadoras [14]. No entanto, também existem implementações de arquitetura pura para P2P-Móvel.



Fonte: referência [14]

Figura 8 Arquitetura híbrida com *super-peers*

Desse modo, este capítulo está estruturado da seguinte forma. Na seção 4.1, serão descritas arquiteturas P2P-Móvel pura que utilizam rede ad-hoc na camada de rede. Na seção 4.2, serão apresentadas arquiteturas híbridas para P2P-Móvel. A seção 4.3 apresentará uma arquitetura que pretende solucionar a comunicação entre as arquiteturas anteriores (P2P-Móvel Pura e Híbrida) e na seção 4.4 são apresentadas arquiteturas para propósito geral e específico em relação às arquiteturas P2P-Móvel. A seção 4.5 finaliza este capítulo descrevendo considerações finais.

4.1 Arquitetura P2P-Móvel Pura

Enfatizando a descrição da topologia MANET (*mobile ad-hoc network*), apresentada na seção 2.3, esta não possui uma estrutura predeterminada consistindo, geralmente, de nós móveis em uma rede sem fio, que podem se comunicar através de nós intermediários. Os nós de uma rede MANET se comportam semelhante a uma rede P2P pura, a diferença é que elas trabalham em diferentes camadas. MANET está especificada até a camada de rede, enquanto o P2P se limite à camada de aplicação. Desse modo, a combinação dessas camadas também pode ser denominada P2P-Móvel (*peer-to-peer móvel*).

Existem algumas dificuldades identificadas nessa combinação, tais como os protocolos P2P tradicionais são incapazes de determinar se está utilizando uma infra-estrutura de rede fixa ou uma MANET. Desse modo, quando se utiliza MANET e ocorrem desconexões devido à mobilidade dos nós, uma aplicação P2P tradicional tentará re-estabelecer a conexão usando a mesma informação de roteamento. Porém poderia haver outras fontes que forneceria um melhor serviço, devido às mudanças na topologia da rede.

Nesta seção, serão apresentadas duas arquiteturas que podem ser classificadas como P2P-Móvel pura e que utilizam uma rede MANET. A arquitetura Mobile CHEDAR será apresentada na seção 4.1.1, a Proem na seção 4.1.2 e na seção 4.1.3 essas duas arquiteturas são comparadas.

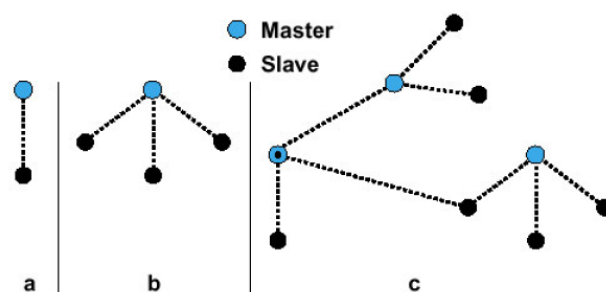
4.1.1 Mobile CHEDAR

CHEDAR²⁰ (*CHEap Distributed ARchitecture*) é um *middleware peer-to-peer* projetado em J2SE (*Java 2 Platform, Standard Edition*) e comunicação através de TCP para o fácil desenvolvimento de aplicações P2P. CHEDAR permite encontrar e manipular qualquer recurso de uma rede, tais como dados (arquivos), *software* (sistemas operacionais ou aplicações específicas) e *hardware* (computadores, impressoras e monitores). Dessa modo, CHEDAR pode ser usado para encontrar computadores inativos e com certas características a fim de executar processamento intenso de dados.

Cada nó de CHEDAR é identificado por um pseudo-identificador (CHEDAR ID). Os nós mantêm também uma base de dados dos recursos localmente disponíveis e compartilhados pelo proprietário do dispositivo. Os recursos remotos, descobertos na rede, também podem ser adicionados à base de dados, combinados com a informação sobre seu proprietário e identificados por CHEDAR ID e meta-informação. A meta-informação pode conter diversas informações, tais como o tipo, o nome e a versão dos dados para aplicações, ou toda a descrição útil para o *hardware*. A base de dados de recursos é armazenada no formato XML usando um DTD²¹ (*Document Type Declaration*) específico. Esta organização dos dados em XML permite a realização de complexas consultas à base de dados utilizando XPath²².

Para satisfazer os computadores móveis e as propriedades inerentes de P2P em rede sem fio com configuração *ad hoc*, o projeto CHEDAR está sendo estendido à plataforma móvel sob o nome de Mobile CHEDAR. Este fornece funcionalidades de registrar recursos no dispositivo móvel e consultar outros *peers*. Esta extensão foi implementada usando J2ME, que é apropriada para dispositivos móveis sem fio. Mobile CHEDAR usa *Bluetooth*, como tecnologia de transmissão de dados, no entanto aplicou certa restrição sobre essa tecnologia de acesso.

A Figura 9 mostra as possíveis topologias quando se utiliza *Bluetooth*.



Fonte: referência [47].

Figura 9 Topologias de redes Bluetooth

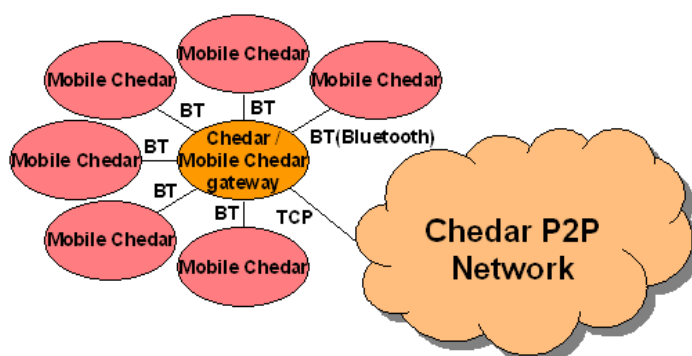
²⁰ Mais informações sobre o projeto CHEDAR podem ser encontradas em <http://tisui.it.jyu.fi/cheesefactory/>

²¹ DTD – É uma forma de validar o documento XML, por meio da especificação de elementos ou atributos que são permitidos em um documento XML, e em que local do documento eles podem aparecer.

²² XPath (*XML Path Language*) - é uma linguagem para manipulação de documento XML, formada por um conjunto de regras de sintaxe para definir partes de um documento XML. Esta linguagem usa uma notação de caminhos para navegar através da estrutura hierárquica de um documento XML, de forma semelhante ao que se faz para referenciar um diretório no disco. Para isso XPath define uma biblioteca de funções padrão.

Em **a**, tem-se uma *piconet* com um único escravo. Em **b**, tem-se uma *piconet* com múltiplos escravos. Em **c**, tem-se uma possível configuração de uma *scatternet* onde cada *piconet* tenha apenas um *master*, porém, escravos podem participar de diferentes *piconets* inclusive o *master* de uma *piconet*, pode ser *slave* de outra *piconet*.

No entanto, o trabalho sobre Mobile CHEDAR informa que atualmente existe uma restrição das implementações da tecnologia *Bluetooth* em relação à topologia c da Figura 9, ou seja, hoje em dia um *peer* não pode se conectar, ao mesmo tempo, a mais de uma *piconet*²³. Então, é possível apenas a implementação de uma única topologia disponível, a estrela, em que um dispositivo tem a função de *master* e os outros a função de *slave*. A Figura 10 exemplifica a topologia estrela, em que o *peer gateway* possui a função máster e os nós Mobile Chedar possuem a função de *slave*.

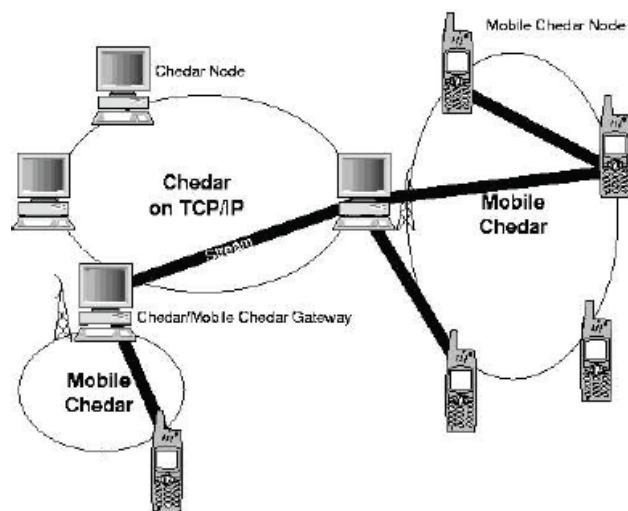


Fonte: referência [46].

Figura 10 Topologia estrela numa rede Mobile CHEDAR

Esta arquitetura se destaca por permitir uma maior alcançabilidade dos *peers* móveis que utilizam *Bluetooth*, devido à adição de um nó *gateway* Mobile CHEDAR/CHEDAR. Este dispositivo além de possui a funcionalidade de *master*, também representa uma estação adaptada com *Bluetooth* e TCP. Isto permite que os *peers* Mobile CHEDAR possam se comunicar com outro grupo de *peers* móveis, assim como existir a comunicação entre *peer* móveis e os *peers* de uma rede fixa. A Figura 11 exemplifica a alcançabilidade.

²³ *Piconet* - Micro-redes de dispositivos. É um termo utilizado na tecnologia *Bluetooth*, onde os dispositivos que estão próximos uns dos outros automaticamente estabelecem contato entre si, formando pequenas redes.



Fonte: referência [46].

Figura 11 Entrega de dados entre os nós Mobile CHEDAR e CHEDAR.

A descoberta de nós vizinhos é um pré-requisito para realizar consultas de recursos. Utiliza-se o protocolo *Bluetooth's Service Discovery* (SDP), desde que os nós estejam disponíveis a se comunicarem utilizando o canal sem fio. Para disponibilizar uma informação nesta rede, o nó Mobile CHEDAR anuncia a outros nós usando o SDP e, ao buscar os vizinhos, as mensagens recebidas são adicionadas à base de dados de recursos, caso a base de dados não esteja atualizada em relação à nova informação. Essa informação armazenada na base de dados de pelo menos um nó deve ao menos conter o CHEDAR ID e seu endereço MAC de *Bluetooth*.

A descoberta dos recursos é executada com uma consulta de um salto, que é identificada pelo identificador único da mensagem. Este salto deve atingir todos os nós dentro do alcance de *Bluetooth*. Desse modo, quando a consulta chega a um nó *gateway* CHEDAR/Mobile CHEDAR, o nó verifica se a consulta foi recebida anteriormente: se não foi, o nó *gateway* envia a mensagem a todos os vizinhos do nó CHEDAR com um *timeout*; se foi recebida anteriormente, a consulta é descartada.

Nessa consulta, se a mensagem recebida combinar com um dos recursos possuídos pelo nó, o nó responde ao vizinho, de que recebeu a pergunta (a mensagem da resposta deve ser marcada com o mesmo identificador da mensagem da consulta). Desse modo, a mensagem de resposta vai até ao nó de origem da consulta, percorrendo um caminho contrário à trajetória da consulta. Após a localização do recurso (ou localizações, se o recurso existir em exemplos múltiplos na rede) ser descoberta, o nó Mobile CHEDAR informa a aplicação, que decide como adquirir ou usar o recurso [28] e [29].

4.1.2 Proem

Semelhante a arquitetura Mobile CHEDAR, Proem²⁴ leva em consideração que um sistema P2P-Móvel é um sistema descentralizado, em que não existe um nó central e todos os *peers* possuem as mesmas funções e responsabilidade; os *links* de comunicação são altamente transientes, ou seja, conexões e desconexões ocorrem freqüentemente e de forma não esperada; os *hosts* móveis podem freqüentemente se mover e independentemente um do outro, sendo possível mudar de topologia de forma não esperada.

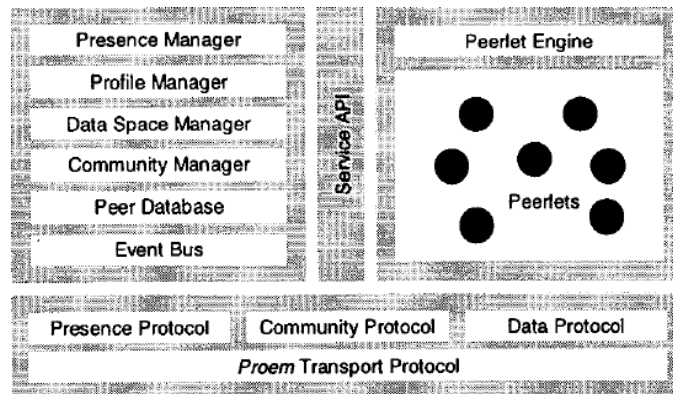
Dessa forma, Proem é um *middleware* que possibilita desenvolver aplicações colaborativas, com mobilidade altamente dinâmica e com dispositivos heterogêneos. Ele suporta conexão 802.11b no modo *ad hoc*. Este *middleware* consiste em três componentes: uma aplicação em ambiente *runtime*, conjunto de serviços e uma pilha de protocolos e seus principais objetivos foram:

- Permitir o desenvolvimento em alto nível de aplicações P2P-Móvel.
- Ser independente de plataforma.
- Permitir interoperabilidade e extensão.
- Dar suporte a conexões intermitentes.
- Fornecer funções para nomeação, descoberta e identificação da presença de serviços, comunicação, gerenciamento de identificadores, de espaço de dados e de comunidades, segurança e privacidade.

Em uma rede com Proem, podem existir quatro entidades: *peer* que é o dispositivo, usuário, espaço de dados e comunidade, que corresponde a um conjunto de entidades. Essas entidades podem ser referenciadas por um ou mais nomes expressos em *Uniform Resource Identifiers* (URL). No entanto, cada nome deve ser único e apenas referenciar a uma entidade. Proem também permite indiretamente referenciar as entidades por meio de *profile* baseado em dados XML para descrever a entidade.

Nesta arquitetura (ver Figura 12) as aplicações são denominadas *peerlets* e são executadas em ambiente *runtime* pelos *peerlet engine*. Os *peerlets* são baseados em eventos, que são enviados na comunicação. O ambiente *runtime* emite os eventos para os *peerlets*, como reação a mudanças no estado interno ou externo, assim como reação a mensagens recebidas por *peers* próximos. Os *peerlets* também podem ser adicionados ou retirados de um *peerlet engine*.

²⁴ Mais informações sobre o projeto Proem podem ser encontradas em <http://wearables.cs.uoregon.edu/proem/index.html>



Fonte: referência [30]

Figura 12 Arquitetura do *Middleware Proem*

O Proem definiu a sintaxe e semântica das mensagens trocadas pelos *peers* por meio de quatro protocolos – um protocolo de transporte de baixo nível e três protocolos de alto nível. Essa definição garante a interoperabilidade entre implementações do Proem em diferentes *hardware* e *software*. Isso porque os *peers* Proem podem ser implementados em qualquer linguagem e não requerem um protocolo de transporte específico. No entanto, é fornecido apenas um ambiente de desenvolvimento para aplicações com Java.

O protocolo de transporte dá suporte às conexões básicas entre os *peers* e utiliza XML para representar seu conjunto de mensagens. Este protocolo pode ser implementado sobre qualquer protocolo existente, tais como TCP/IP, UDP e HTTP. O protocolo de presença permite que o usuário anuncie sua presença e descubra a presença de outros usuários; o protocolo de dados permite aos *peers* compartilhar e sincronizar os dados; e o protocolo de comunidade permite que os *peers* estabeleçam relacionamentos de confiança pela formação de grupos de *peers* mutuamente confiáveis.

O conjunto de serviços fornece funcionalidades comuns para os *peerlets*. Esses serviços são apresentados em uma API de alto nível. O serviço gerenciamento de presença é responsável por anunciar a presença de usuários e por descobrir usuários próximos (a proximidade depende da topologia da rede e inclui os usuários alcançáveis diretamente ou indiretamente); o gerenciador de perfil mantém informações sobre identificação de usuários e recursos compartilhados; gerenciador de espaço de dados é responsável pela persistência do armazenamento do espaço de dados, assim como controle de acesso; gerenciador da comunidade mantém os rastros dos *peers* em um grupo e também valida os *peers* no grupo; o banco de dados do *peer* mantém um persistente *log* de encontros com outros *peers* (e usuários) e permite *peerlets* adicionar meta-informações customizadas, o que permite determinar quando e qual a frequência de um particular *peer* ou usuários tem sido encontrado no passado; e o barramento de eventos permite a comunicação baseada em eventos entre os *peerlets* [28], [30], [31] e [32].

4.1.3 Proem X Mobile CHEDAR

As arquiteturas Proem e Mobile CHEDAR podem ser classificadas como arquitetura P2P-Móvel pura e as mesmas apresentam certas vantagens e desvantagens. Ambas possibilitam o desenvolvimento em alto nível de aplicações P2P-Móvel que serão executadas em ambiente caracterizado pela grande mobilidade, pela existência de dispositivos heterogêneos e pelos *links* de comunicação muito transientes. Essas características causam a mudança de topologia de forma não esperada.

A desvantagem de ambas arquiteturas P2P-Móvel pura é a provável ineficiência em relação à performance, devido ao uso de XML. Proem usa XML nos protocolos de comunicação, enquanto Mobile CHEDAR utiliza XML no formato dos dados armazenados.

Quanto à identificação dos *peers*, na arquitetura Mobile CHEDAR, cada *peer* é referenciado por um identificador único denominado CHEDAR ID, enquanto na Proem um *peer* pode ser referenciado por vários nomes únicos nos formatos URL ou XML. Isto facilita a interoperabilidade na Internet, que por outro lado, adiciona complexidade.

Mobile CHEDAR permite apenas a tecnologia de acesso *Bluetooth* (WPAN) e atualmente não dar suporte a transferência de dados de áudio/vídeo. A Proem permite a tecnologia de acesso 802.11b (WLAN) e possibilita a transferência de dados em diversas aplicações, tais como uma aplicação de *chat* e compartilhamento de MP3.

Mobile CHEDAR destaca-se em relação à integração de rede fixa e móvel, assim como em relação à alcançabilidade entre *peers*. Provavelmente a alcançabilidade entre *peers* de uma rede Proem será menor que em uma rede CHEDAR/Mobile CHEDAR. Isso porque só é possível utilizar a estrutura *ad-hoc* na rede Proem. Enquanto na outra rede, podem existir nós com funcionalidade adicional de *gateway* entre redes CHEDAR formadas por *peers* móveis e fixos.

4.2 Arquitetura P2P-Móvel Híbrida

Nesta seção, serão apresentadas arquiteturas que permitem o desenvolvimento de aplicações P2P-Móvel, em ambientes onde os *peers* podem formar uma rede P2P-Móvel híbrida. Na seção 4.2.1, a arquitetura JXME será apresentada, a seção 4.2.2 descreve a arquitetura X-Peer e a seção apresenta comparações entre essas arquiteturas em relação a P2P-Móvel.

4.2.1 Arquitetura JXTA para Dispositivos Móveis (JXME)

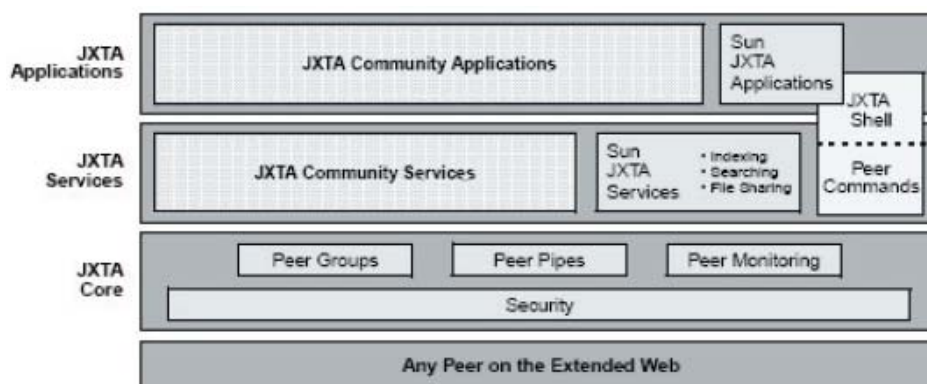
JXTA²⁵ é uma arquitetura P2P que disponibiliza um conjunto de protocolos genéricos para programação avançada de aplicações P2P. Ela tem sido testada em redes fixas e possui

²⁵ Detalhes sobre o projeto JXTA podem ser encontrados em <http://www.jxta.org>

atualmente duas versões para *peers* móveis, desenvolvidas pelo projeto JXTA-J2ME (JXME²⁶).

Os *peers* JXTA criam uma rede virtual e *ad-hoc* sobre a rede existente. Esses *peers* se comunicam por meio de um modelo estruturado DHT. Ela disponibiliza tecnologias WEB tais como HTTP (*HyperText Transfer Protocol*), TCP/IP e XML.

A arquitetura JXTA pode ser apresentada em três camadas: camada principal, camada de serviços e camada de aplicação como visto na Figura 13. A camada principal está dividida em *peer* para estabelecimento, gerenciamento da comunicação, tais como roteamento, e outras tarefas do nível de redes. A camada de serviço está dividida em conceitos de alto nível tais como indexação, busca e compartilhamento de arquivos. A camada de aplicação inclui aplicações como sistemas de *e-mail*, de leilão e armazenamento.



Fonte: referência [34]

Figura 13 Arquitetura JXTA

O modelo de endereçamento do JXTA é baseado em um modelo uniforme e de localização independente do endereçamento lógico. Todos os recursos da rede possuem um *peer ID*, independente do endereço físico. A arquitetura padrão é baseada em *super-peers Rendezvous* e *Relays*. *Rendezvous* são *peers* que são responsáveis por armazenar índices dos *peers* ligados aos *super-peers*, eles fornecem o serviço de busca de *peers* para depois ser realizada a comunicação P2P entre os *peers*, mesmo quando estes estiverem conectados a *super-peers* diferentes, mas não possui a capacidade de atravessar *firewalls*. *Relays* são *peers* que possuem um mecanismo para comunicação com outros *peers* separados por *firewalls* ou NAT. Estas entidades podem ser vistas na Figura 14.

²⁶ Mais informações sobre o projeto JXME estão disponibilizadas em <http://jxme.jxta.org/>

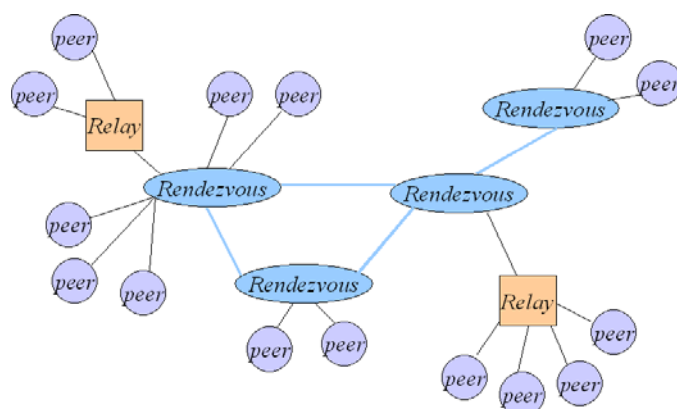


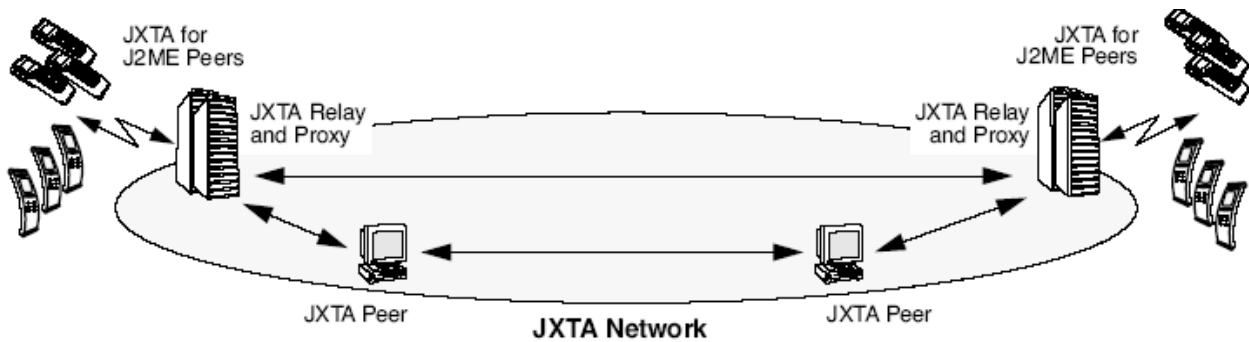
Figura 14 Entidades de uma rede JXTA

Em novembro de 2001, foi divulgado que a expansão dos protocolos de JXTA para permitir o desenvolvimento de aplicações P2P em dispositivos móveis, ou seja, a primeira versão do JXME, seria realizada [33]. Atualmente JXME possui duas versões: *proxied* (JXME 1.0) e *proxyless* ou sem *proxy* (JXME 2.0). O propósito destas versões é permitir que JXTA forneça funcionalidades compatíveis com J2ME, usando o *Connected Limited Device Configuration* (CLDC) e o *Mobile Information Device Profile* (MIDP). Portanto, teoricamente, qualquer dispositivo MIDP é capaz de participar de atividades P2P com outro dispositivo MIDP [9].

Alguns dos objetivos iniciais do JXME foram: interoperabilidade com JXTA em *desktops* e *workstations*, fornecer uma infra-estrutura para pequenos dispositivos, ser pequeno o suficiente para poder rodar em celulares e PDAs, assim como ser compatível com MIDP 1.0. No entanto, a versão MIDP 1.0 possui limitações como: a ausência de um *parser* para XML e suporta apenas HTTP. A versão *proxied* atendeu a esses objetivos, mas possui as limitações do MIDP 1.0 que impossibilita que os *peers* móveis oferecessem serviços aos outros *peers* [16] e [34].

A versão *proxyless* utiliza XML e possui quase todas as funcionalidades do JXTA nativo, ao passo que a versão *proxied* é mais leve e precisa de um *peer* JXTA nativo para agir como seu *proxy*. Isso porque a versão *proxied* utiliza comunicações com codificação binária (mais leve) com seu *proxy*. Este tem a função de conversão entre formatos XML e binário, para permitir a interoperabilidade entre a versão *proxied*, *proxyless* e JXTA. Analogamente, as duas versões de JXME não podem ser executadas como *super-peers*. Uma das restrições de JXME é que ele trabalha atualmente apenas no ambiente Java e ainda não existem versões JXTA portadas para a linguagem *Symbian C++* [9].

Como descrito no parágrafo anterior, a versão *proxied* necessita de um *peer* com função de *proxy*, denominados JXTA *relay*. Dessa forma, os *peers* móveis agem praticamente como pontos de visão (clientes). Os *relays* ficam encarregados de realizar tarefas com intenso processamento, busca de recursos, fornece a interoperabilidade com os protocolos JXTA entre outras funcionalidades para otimizar e permitir a participação dos *peers* móveis. Esta arquitetura pode ser vista na Figura 15, em que não existiu a comunicação P2P entre *peers* móveis.



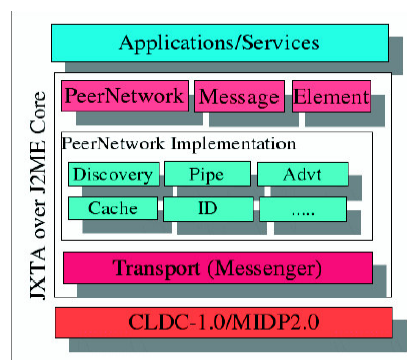
Fonte: referência [34]

Figura 15 Arquitetura JXTA com JXME 1.0

Na Figura 15, os dispositivos móveis solicitam uma busca ao *relay* JXTA que encaminha para rede JXTA, as respostas para os *peers* móveis passam pelo *proxy* que “enxuga”, otimiza a mensagem e transfere para o dispositivo, uma das suas vantagens é que a redução da mensagem lida a uma redução do volume do tráfego [16]e [34].

É importante salientar que o uso de *relays* JXTA não é uma nova idéia para JXTA. Eles existiam anteriormente, quando dois *peers* não eram alcançáveis, devido à existência de *firewall* e, ou NAT. Diferente do modelo cliente-servidor, os *peers* móveis não precisam estabelecer e manter um relacionamento estático com os *relays* JXTA. Dois *peers* móveis podem se conectar a diferentes *relays*, descobrirem-se e comunicarem-se; também podem dinamicamente mudar de *relay*, ou ter vários *relays*. A referência [10] informa que no futuro, os *peers* móveis poderão procurar por *relays* de JXTA e configurar então um deles para ser seu representante.

A versão JXME 2.0 tem o intuito de: dá suporte a descoberta de grupos JXTA e conteúdos; criar grupos e conteúdos; participar e entrar em grupos; e realizar o estabelecimento de comunicações com outros *peers*. Atualmente, ela apenas permite a abstração de NAT e *firewall* quando a comunicação utiliza TCP. A arquitetura desta nova versão (*proxyless*) pode ser vista na Figura 16[16].



Fonte: referência [16]

Figura 16 Arquitetura *proxyless* JXME

O próximo objetivo do projeto JXME, informado em 2005 na referência [35], é atualizar JXME para que seja retirada sua dependência a um *proxy*.

4.2.2 Arquitetura X-Peer

Com o objetivo de extrair as maiores vantagens de cada um dos modelos citados na seção 4, foi proposto o *middleware* X-Peer²⁷, que utiliza o conceito de super-nós executando sobre um modelo estruturado DHT para garantir escalabilidade e robustez. Dessa forma, o X-Peer pode ser configurado para ser desde um sistema P2P centralizado, caso exista apenas um nó X-Peer, até um sistema totalmente descentralizado, caso um X-Peer execute em cada nó da rede. Se o X-Peer for configurado para trabalhar com um número intermediário de nós X-Peer, ele se comporta como o modelo híbrido do KaZaA, sendo que os super-nós se comunicam de maneira estruturada, através de um sistema DHT.

Cada nó X-Peer mantém um conjunto de usuários (aplicações que utilizam serviços do X-Peer) conectados a ele, onde as meta-informações referentes a cada usuário ficam distribuídas entre os super-nós que compõem a rede X-Peer. Essas informações podem ser acessadas por qualquer outro super-nó X-Peer em no máximo $\log(n)$ saltos, uma vez que o X-Peer é modelado sobre um DHT [36], onde n é o número de super-nós que a rede X-Peer contém.

Atualmente esta arquitetura está em execução em três pontos de presença (POP) da RNP (Rede Nacional de Pesquisa). A Figura 17 mostra os locais em que os super-nós que formam a rede X-Peer estão sendo executados: Pernambuco (PE), Minas Gerais (MG) e Paraná (PR). Como pode ser visto, atende a *peers* fixos, assim como *peers* móveis.

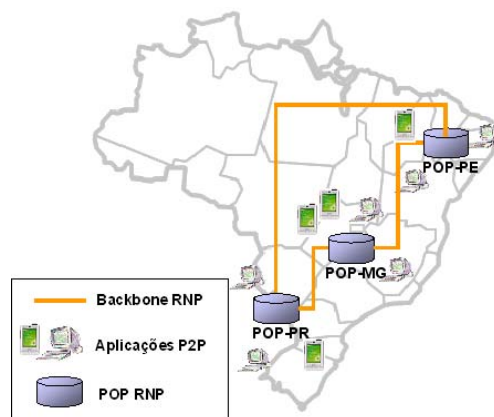


Figura 17 Infra-estrutura X-Peer na rede da RNP

Através do uso do X-Peer, qualquer super-nó da rede pode deixar de funcionar e mesmo assim as aplicações conectadas continuam em execução. Outra característica inerente às aplicações P2P, presente nesse *middleware*, é a capacidade de um super-nó acessar diretamente qualquer outro super-nó da rede.

²⁷ Informações adicionais sobre X-Peer estão disponibilizadas em <http://www.gprt.ufpe.br/gtp2p/>

Nesta arquitetura podem existir dois tipos de comunicação direta (P2P) e indireta. Uma comunicação indireta ocorre, quando os usuários não conseguem estabelecer uma comunicação, devido à presença de *firewall* e/ou NAT entre os *peers*. Todo super-nó tem papel de cliente e de servidor, operando como servidor, quando compartilha meta-informações, e como cliente, quando recupera meta-informações na rede. As aplicações por sua vez também possuem o papel de cliente e servidor quando compartilham dados entre si.

A arquitetura do X-Peer é dividida em três camadas principais, ilustradas pela Figura 18.

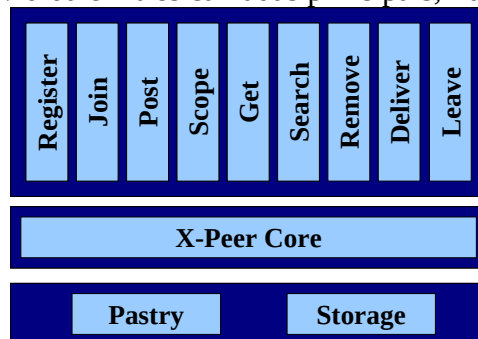


Figura 18 Arquitetura X-Peer

- Camada de Serviços (Register, Join, etc) – Contém os serviços disponibilizados pelo X-Peer, úteis para a construção de aplicações P2P;
- Camada de Aplicação (X-Peer Core) – É uma aplicação distribuída que atende solicitações de diversas aplicações conectadas simultaneamente;
- Camada de Módulos Úteis (Pastry e Storage) – Fornece alguns componentes para prover serviços à camada de aplicação.
- Pastry: solução DHT adotada, que fica encapsulada sob uma interface genérica DHT, e, portanto, pode ser substituída por outra solução DHT que forneça a mesma API, como o Chord.
- Storage: sistema de armazenamento em arquivo desenvolvido especificamente para o X-Peer.

Semelhante a arquitetura JXTA, o X-Peer foi projetada com a intenção de ser um *framework* flexível para o desenvolvimento de aplicações P2P. Este é o diferencial em relação às outras arquiteturas P2P tradicionais como KaZaA e Skype. Estas redes P2P são dedicadas a atender apenas um tipo de aplicação P2P.

Os serviços que a infra-estrutura X-Peer disponibiliza para auxiliar aplicações são descritos abaixo:

- **Join** – Através desse serviço o usuário solicita a sua participação na rede.

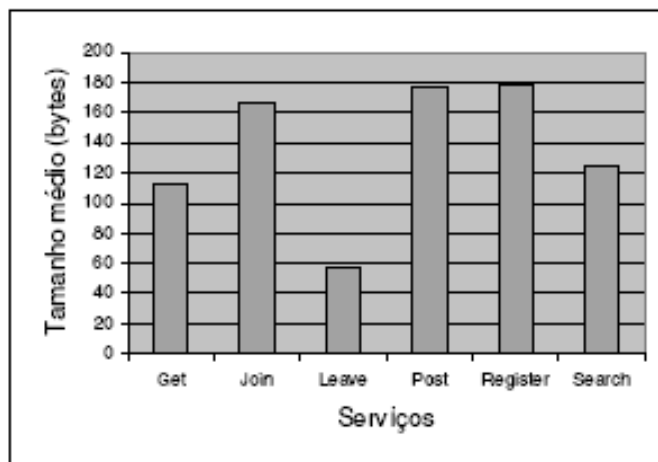
- **Leave** – Esse serviço faz com que o usuário deixe a rede requisitando à rede X-Peer que remova as informações publicadas pelo usuário. No entanto, este serviço é chamado automaticamente pela rede X-Peer, quando a rede identifica que o usuário se desconectou sem solicitar o serviço Leave.
- **Scope** – Especifica a rede visível por um usuário, ou seja, define os usuários que são acessíveis durante uma busca.
- **Post** – Cada usuário pertencente à rede X-Peer mantém uma estrutura de dados com informações públicas, que servem para localizar o usuário na rede. O Post é o serviço que permite adicionar informações nessa estrutura.
- **Get** – Recupera informações públicas através do serviço Post.
- **Remove** – Remove um campo publicado através do Post.
- **Search** – Esse serviço permite localizar um usuário na rede X-Peer. Essa localização pode ser feita através do nome do usuário ou de uma informação publicada por ele. O nome do campo, publicado através do Post, também pode ser utilizado para recuperar um usuário.

Este *middleware* possui dois protocolos *XApplicationProtocol* e o *XDHTProtocol*. O primeiro é formado por um conjunto de mensagens trocadas entre as aplicações que estão conectadas e o X-Peer. O segundo é formado por um conjunto de mensagens trocadas entre os X-Peers para atenderem um serviço. Inicialmente os protocolos deste *middleware* foram implementados utilizando XML e Java com a API J2SE.

Em 2004, o conjunto de mensagens do protocolo *XApplicationProtocol* foi reestruturado para atender dispositivos móveis e por isso implementadas com a API J2ME. Após a implementação, foram realizadas avaliações de desempenho utilizando a aplicação desta monografia (descrita na seção 6.0). Os resultados das avaliações apresentaram tempos de respostas inviáveis para qualquer aplicação que utilizasse o protocolo *XApplicationProtocol* em equipamentos restritos como celulares e *handhelds*.

Após a identificação da baixa performance com os protocolos em XML, devido principalmente a execução do *parser* do XML nos dispositivos, os conjuntos de mensagens dos protocolos foram reimplementados utilizando o formato binário.

A Figura 19 ilustra o tamanho das mensagens em *bytes* quando possuíam o formato XML. Pode-se observar que o valor médio obtido ficava em torno de 135 *bytes* e o tamanho da mensagem varia de serviço para outro, mas a maioria das mensagens é menor que 180 *bytes*. Após a mudança para o formato binário, as mensagens passaram a ser muito pequenas, quando comparadas com as do formato XML. Um exemplo é a mensagem Leave que possui atualmente o tamanho de 3 byte, enquanto a mesma no formato XML possuía um pouco menos de 60bytes. Portanto, as modificações resultaram em mensagens menores e mais eficientes, permitindo um melhor uso de aplicações P2P-Móveis na rede X-Peer.



Fonte: referência [27]

Figura 19 Tamanho médio das mensagens referentes aos serviços X-Peer

Desse modo, esta arquitetura permite que aplicações P2P-Móveis possam se comunicar diretamente independente do hardware que esteja sendo executada. Uma das vantagens desta arquitetura P2P-Móvel é a possibilidade de realizar a comunicação entre diferentes equipamentos utilizando TCP, sem a necessidade de *proxys* para realizar conversões das mensagens XML e binária. Essa conversão acrescentaria um *overhead* adicional de comunicação P2P entre dispositivos móveis.

4.2.3 X-Peer x JXTA-JXME

Os projetos JXTA e X-Peer inicialmente foram desenvolvidos para permitir a construção de aplicações em rede P2P do tipo híbrida e com configuração de conexão no modo de infraestrutura. No princípio, também estavam caracterizadas como um *framework* flexível para o desenvolvimento de aplicações P2P tradicionais que seriam executadas em computadores com baixas restrições de recursos.

O crescimento no uso de dispositivos móveis motivou estes projetos a reestruturarem suas arquiteturas para permitir a implementação de aplicações P2P-Móvel. No entanto, essas reestruturações foram realizadas de maneiras diferentes.

JXTA optou por continuar a usar protocolos em XML e, para tal, foi necessário fornecer um menor protocolo em XML para a segunda versão de JXME. Além disso, o grupo adicionou *proxys* na rede JXTA para realizar conversões entre os protocolos no formato XML e formato binário. Desse modo, a rede JXTA utiliza conversões para atender aos dispositivos móveis com alta restrição de hardware, os quais utilizam a primeira versão de JXME (protocolos no formato binário). Já o projeto X-Peer optou por modificar todo o conjunto de mensagens dos protocolos do formato XML para a estrutura binária, porque prioriza o requisito de performance.

De maneira semelhante, as duas arquiteturas fornecem APIs para que os desenvolvedores construam aplicações sem se preocupar com NAT e *firewall*, comuns em grandes redes,

como a Internet. Essas APIs só abstraem esses elementos, quando se utiliza o protocolo de transporte TCP para realizar as comunicações P2P.

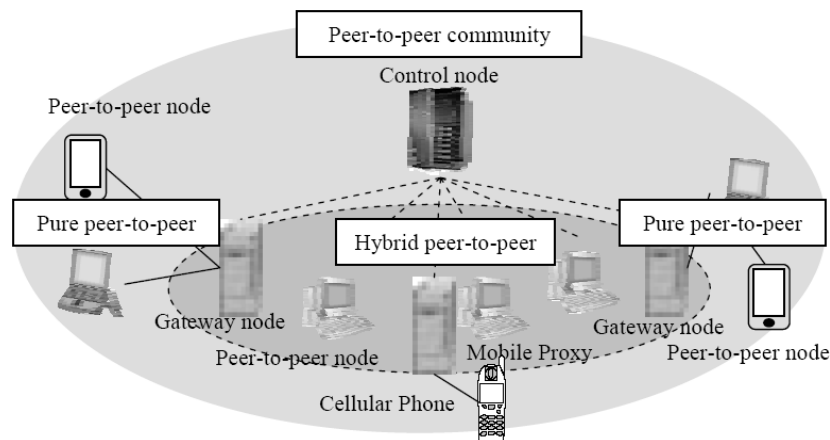
A vantagem de J2ME é que o mesmo permite desenvolver aplicações utilizando o protocolo HTTP e programar com todas as APIs de J2ME, mas a comunicação P2P é impossível na primeira versão de J2ME.

A desvantagem da arquitetura X-Peer é que ela restringe a implementação de aplicações para dispositivos móveis, porque a API dependente da versão de J2ME 2.0 ou superior, uma vez que as anteriores a 2.0 de J2ME não permitem abrir conexões com *Socket*. Isto impossibilita a comunicação P2P. No entanto, esta arquitetura possui a vantagem de possuir protocolos simples, principalmente, quando comparados com mensagens em XML, e não possui entidade intermediária para realizar traduções de protocolos, que causam um atraso adicional.

4.3 Arquitetura para Comunicação entre as Redes P2P Pura e Híbrida

Todas as entidades de uma comunicação *peer-to-peer* têm um conjunto de interesses comuns e obedecem a um conjunto comum de políticas, construindo uma comunidade P2P. A arquitetura da Figura 20 propõe a possibilidade de comunicação entre as duas redes P2P puras e uma híbrida. Estas redes foram apresentadas na introdução deste capítulo e nas seções anteriores. Para esta integração, foram definidas as seguintes três novas entidades:

- *Mobile proxy* oferece funcionalidades adicionais para dispositivos de capacidade restrita;
- *Control node* é uma entidade administrativa. Esta entidade gerencia a comunidade P2P na rede P2P. Ela fornece várias funções independentes de aplicações, tais como resolução de nome, disponibilização de informações para roteamento, otimização da topologia da rede, autenticação de nós e gerenciamento de grupos *multicast*;
- *Gateway node* é a entidade intermediária entre as redes P2P pura e híbrida. As três principais funções são: traduzir a “ordem” imposta pelo *control node*, coletar informações da topologia das redes P2P puras, em seguida reportar para o *control node* e dar suporte a comunicação entre rede P2P híbrida e pura.

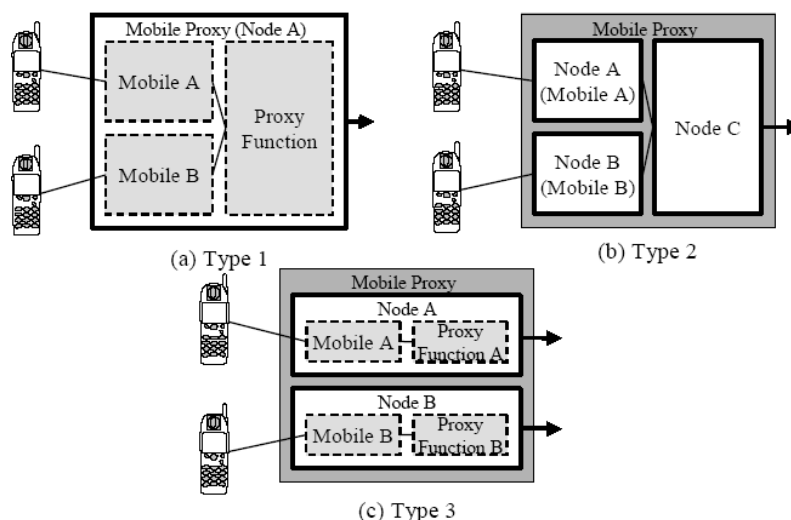


Fonte: referência [36]

Figura 20 Arquitetura para comunidade P2P

O *proxy mobile* permite que um dispositivo móvel possa agir virtualmente como um nó *peer-to-peer*, entidade independente, e tenha uma performance necessária para funcionalidades da arquitetura P2P. Para isso, este *proxy* pode ser modelado de três formas, vistas na Figura 21:

- Tipo (a) Os dispositivos móveis compartilham o mesmo *proxy*, sendo assim considerados juntamente um nó na rede P2P.
- Tipo (b) Cada dispositivo móvel é virtualmente um nó e identificado separadamente por nomes diferentes. Para a realização deste tipo de *proxy mobile*, algumas funcionalidades devem ser implementadas no nó C, tais como transformação de mensagens recebidas de um dispositivo móvel para uma mensagem do protocolo *peer-to-peer*.
- Tipo (c) Este *proxy mobile* é formado por um par de um dispositivo móvel e uma função *proxy*. Neste caso, um dispositivo móvel tem seu próprio nome e age como nós separadamente através do *proxy mobile* que não age como um nó separadamente.



Fonte: referência [36]

Figura 21 Modelagens do *Proxy Mobile*

Pode-se observar que este trabalho inova a abordagem de uso de *proxys* para atender aos dispositivos móveis. O cenário para compartilhamento de arquivos é um exemplo em que esses *proxys* podem se comportar de maneiras diferentes. Nos próximos parágrafos, será exemplificado o comportamento desses *proxys* para atender a uma aplicação de compartilhamento de arquivos.

Para o tipo de *proxy* **(a)**, o equipamento com a funcionalidade de *proxy* existiria para atender uma rede WPAN. Esta rede se comportaria como um único *peer* na rede P2P para compartilhar arquivos. A rede P2P poderia solicitar ao *proxy* uma certa música, em seguida o *proxy* solicita diferentes fragmentos a cada *peer* da sua rede WPAN e retornando um único resultado.

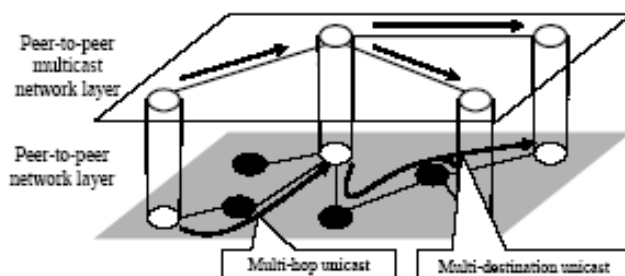
Para o tipo de *proxy* **(b)**, o equipamento com a funcionalidade de *proxy* não existiria apenas para atender uma rede WPAN. Ao chegar uma solicitação de arquivo para o *proxy*, ele identificaria se a mensagem é para o próprio *proxy* (nó C) ou para um dos *peers* da rede WPAN.

Para o tipo de *proxy* **(c)**, o equipamento com a funcionalidade de *proxy* existiria apenas para atender uma rede WPAN. No entanto, apresentaria comportamento diferente do *proxy* tipo **(a)**. Ao chegar uma solicitação de arquivo para o *proxy*, ele identificaria qual dos *peers* da rede WPAN está sendo solicitado para entregar um arquivo.

Portanto, esses três tipos de modelagens devem existir, para que os *proxys* se adequem aos requisitos da aplicação *peer-to-peer*.

O modelo de comunicação, para a arquitetura desta seção, fornece a possibilidade de três tipos de comunicação: *unicast*, *multicast* e *broadcast*. Uma mensagem *unicast* é enviada ao nó de destino diretamente, usando *multi-hop unicast* ou *multi-destination unicast*, a diferença pode ser observada na Figura 22. Quando um nó recebe uma mensagem *multicast* de um nó de um grupo *multicast*, o nó receptor encaminha a mensagem recebida ao grupo

de nós adjacentes restantes usando *multi-hop unicast*. O tipo de mensagem *broadcast* ocorre, quando um nó envia uma mensagem broadcast a todos os nós adjacentes, mas o encaminhamento de uma mensagem broadcast é controlado por sua contagem de saltos. Isso porque, os mecanismos de nomeação e de roteamento de mensagens do *P2P Core Protocol* são definidos de forma independente do protocolo de transporte. As mensagens de pedido, de resposta e de anúncio de comunicação permitem realizar a comunicação *peer-to-peer*.



Fonte: referência [36]

Figura 22 Multi-hop X Multi-destination

Os protocolos definidos para esta arquitetura foram implementados com XML e desenvolvidos usando HTTP, TCP e *Bluetooth* [36]. Uma camada *P2P Core Protocol* foi definida para processar as mensagens *peer-to-peer* que podem ser *multi-hop*, *broadcast* e *unicast*, a fim de permitir essas mensagens nas redes *peer-to-peer* puro e híbrido. Uma vez, que no modelo puro é caracterizado por existir *multi-hop* e *broadcast*, enquanto o modelo híbrido é caracterizado pelas mensagens *unicast*.

Apesar da definição dos protocolos em XML, é possível que as aplicações P2P para *peers* móveis sejam eficientes, devido às três abordagens de *proxy* na rede. Portanto, o uso da arquitetura JXTA-JXME, juntamente com esses tipos de *proxys* é uma solução para implementação de aplicações mais eficientes em relação à presença de *peers* moveis em uma rede P2P que utiliza XML.

4.4 Arquitetura P2P-Móvel de Propósito Geral e Específico

A seção 4.4.1 apresentará uma arquitetura P2P-Móvel de propósito geral a fim de solucionar o problema de interoperabilidade descrito na seção de problemas do capítulo 2. A seção 4.4.2 apresentará arquiteturas que tem um propósito específico de tratar questões de compartilhamento de arquivos em P2P-Móvel.

4.4.1 Arquitetura P2P-Móvel de Propósito Geral (PnPAP)

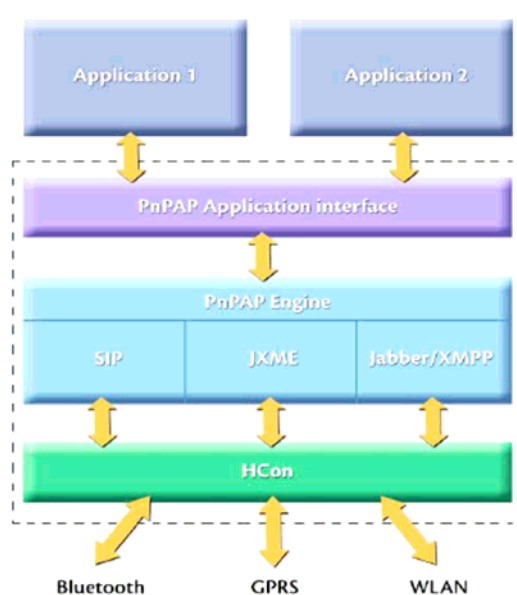
Atualmente, os desenvolvedores de aplicativos P2P deparam-se com diversos protocolos e tecnologias de acesso. No entanto, eles precisam decidir a escolha de um antigo protocolo que permite interoperabilidade, ou novo protocolo que fornece maior número de requisitos

necessários para a aplicação. Ocorrem casos em que é necessário escolher um protocolo com a maior compatibilidade, mesmo possuindo menor número de características necessárias para a aplicação, o que acarreta em dependências entre a aplicação e um conjunto de protocolos.

A arquitetura PnPAP (*Plug-and-Play Application Platform*) foi projetada para abstrair a situação anterior. Ela é uma plataforma que tem o objetivo de fornecer a comunicação entre aplicações móveis que utilizam diferentes protocolos P2P e diferentes tecnologias de conexão de maneira flexível e eficiente. A PnPAP resolve um dos problemas da interoperabilidade, porque possui métodos para traduzir as diferentes maneiras de endereçamento de cada protocolo P2P. Dessa forma, ela permite a construção de aplicações sem se preocupar com a interoperabilidades de protocolos e tecnologias de acesso.

PnPAP permite ser utilizada por todas as linguagens de programação em rede que possam rodar em dispositivos móveis, assim como abstrai a tecnologia de acesso utilizada para conexão. Isso porque foi acoplada com a camada de gerenciamento Hcon (*Holistic Connectivity*), que pode dinamicamente usar diferentes tecnologias de conexão de rede, tais como WLAN, *Bluetooth*, EDGE e GPRS.

A Figura 23 ilustra a estrutura da PnPAP que consiste de uma interface e de um motor, que reage a requisições recebidas através da interface. A seguir serão detalhadas a camada HCon e o motor da arquitetura PnPAP.



Fonte: referência [9]

Figura 23 Arquitetura PnPAP

HCon tem o conhecimento e controle das diferentes tecnologias de conexão de rede. PnPAP utiliza essa camada para mudar o tipo de conexão de maneira transparente, sem que o usuário ou aplicação requirite.

A referência [9] informa que a rede de *peers* pode se conectar usando a melhor tecnologia para uma determinada tarefa. Considerando o cenário em que existe um equipamento A com apenas *Bluetooth* e um equipamento B com *Bluetooth* e *WLAN*, PnPAP escolherá a conexão utilizando *Bluetooth* no equipamento A e *WLAN* no equipamento B. De acordo com o capítulo 2, em uma comunicação que utiliza *Bluetooth* e *WLAN*, a taxa de transferência utilizada na comunicação P2P será a menor (*Bluetooth*). Portanto, não existe a vantagem de otimização da tecnologia de acesso.

Para realizar esta otimização na escolha do protocolo, existe um motor na plataforma PnPAP que é controlado por uma máquina de estados (ME). Neste modelo, o motor e a ME se comunicam usando eventos em que o motor pode enviar um evento para a ME disparar a transição de estado. A transição pode gerar um evento que define uma ação a ser melhorada no motor. Um exemplo, visto na Figura 24, é o motor gera o evento “SIP disponível”. Isto pode gerar a transição de estado e gerar o evento “trocar para SIP” no motor.

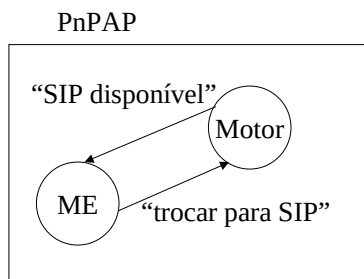


Figura 24 Exemplo da comunicação entre o ME e o Motor

A inovação deste acesso é na representação da ME. A representação é explícita e baseada em um XML denominado RDF (*Resource Description Framework*). A ME é executada pelo interpretador de máquinas de estados que atualmente se comunica com o motor. O artigo também indica que a vantagem é que o comportamento do motor pode ser modificado sem mudar o código – não é necessário re-compilar ou instalar nova versão. Em vez disso, o motor passa a descrição da ME para o interpretador como um dado. Essa descrição é passada quando o motor é inicializado e novas descrições podem ser passadas, quando a aplicação está em execução.

As vantagens listadas no artigo [9] em relação à existência de uma ME controlando a PnPAP e HCon foram:

- Ambiente uniforme para todas as aplicações P2P.
- Fácil desenvolvimento e manutenção das aplicações.
- O melhor protocolo P2P e conectividade são sempre escolhidos.
- Melhor interoperabilidade entre protocolos, dispositivos e redes.
- O comportamento da PnPAP pode ser modificado, devido à interpretação da ME que é baseada em RDF.

No entanto, foi identificado no capítulo 2, que as aplicações móveis são caracterizadas por diversas restrições, em que se destaca o processamento e energia. Desse modo, pode-se chegar a uma conclusão contrária ao do artigo. Provavelmente, a flexibilidade do motor é uma desvantagem para dispositivos móveis. Isso fica claro como desvantagem, quando é indicada, no trabalho, a necessidade da mudança da representação do motor de RDF para um formato binário, pois a atualização da ME não é muito freqüente.

É possível concorda com o trabalho da arquitetura PnPAP, quando este é considerado como uma solução para otimizar o consumo de memória e processamento em relação ao *overhead* e à baixa performance causados por diferentes protocolos. Isso porque o SIP, por exemplo, é mais leve que o JXTA.

4.4.2 Arquiteturas para Compartilhamento de Arquivos em Dispositivos Móveis

Uma das áreas, em que a tecnologia P2P tem se mostrado de grande sucesso na Internet, é a troca e armazenamento de conteúdo, por isso esta seção apresentará arquiteturas de propósito específico para o compartilhamento de arquivos.

As aplicações P2P, para compartilhar recursos, necessitam ter suporte a dois requisitos fundamentais: coordenação ou mecanismos de mediação de recursos e funções de controle. Um exemplo para o primeiro requisito seria a disponibilização de uma função que permita localizar recursos ou entidades. Exemplos para mecanismos de controle são funções que permitem, priorizam e agendam os acessos aos recursos. A arquitetura P2P Pura implementa ambos os mecanismos nos nós da rede, enquanto sistemas P2P Híbridos possuem estas funções em entidades centrais.

As seções 4.4.2.1 e 4.4.2.2 apresentam cenários em que a troca de arquivos atualmente pode ser realizada entre dispositivos móveis utilizando a rede celular com a tecnologia 2/2,5G. A seção 4.4.2.3 descreve a proposta de modificação da arquitetura eDonkey para permitir que dispositivos móveis possam compartilhar arquivos utilizando P2P-Móvel. Na seção 4.4.2.4, uma comparação entre as arquiteturas de compartilhamento de arquivos é realizada.

4.4.2.1 Utilizando Serviço MMS

Esta arquitetura híbrida apresenta a possibilidade de compartilhamento de arquivos para dispositivos móveis utilizando um ou mais servidores (*super-peers*) dedicados. Os servidores permitem o cadastrado e armazenamento dos números (MSISDN²⁸) dos celulares e listas de arquivos vinculados ao número do usuário. O servidor armazena estas

²⁸ MSISDN - *Mobile Station ISDN (Integrated Services Digital Network)* é o número que permite realizar um conjunto de chamadas para um assinante correspondente. Um assinante pode ter vários MSISDNs, correspondentes a diferentes serviços.

informações em um banco de dados. A interface com o banco de dados pode ser utilizando um WAP ou uma página de buscas XHTML (*Extensible HyperText Markup Language*).

Esta arquitetura está exemplificada na Figura 25. Inicialmente, o dispositivo móvel utiliza a interface com o banco de dados, para saber se existe um usuário que possui um determinado arquivo para ser compartilhado. Se o usuário, possuidor do arquivo desejado, for encontrado, o número do celular é retornado. Após o dispositivo obter o número, ele solicita um pedido de compartilhamento usando mensagens do serviço MMS (*Multimedia Messaging System*) e finalmente a transferência do arquivo é realizada de maneira P2P. Portanto, esta arquitetura, admite a existência da comunicação direta com MMS entre os celulares, após o encontro do número do celular que contém um arquivo [16].

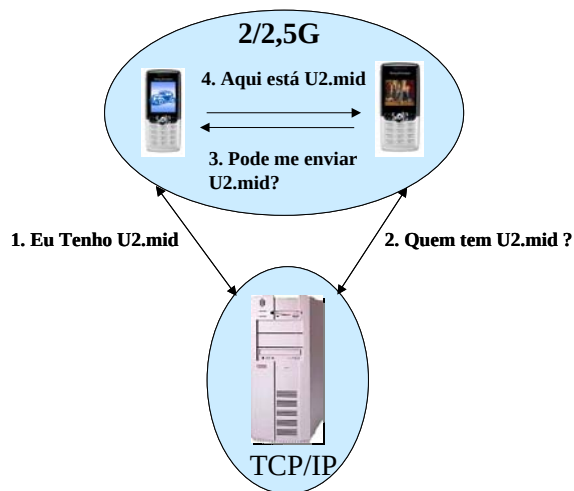


Figura 25 Busca e compartilhamento de arquivo MMS

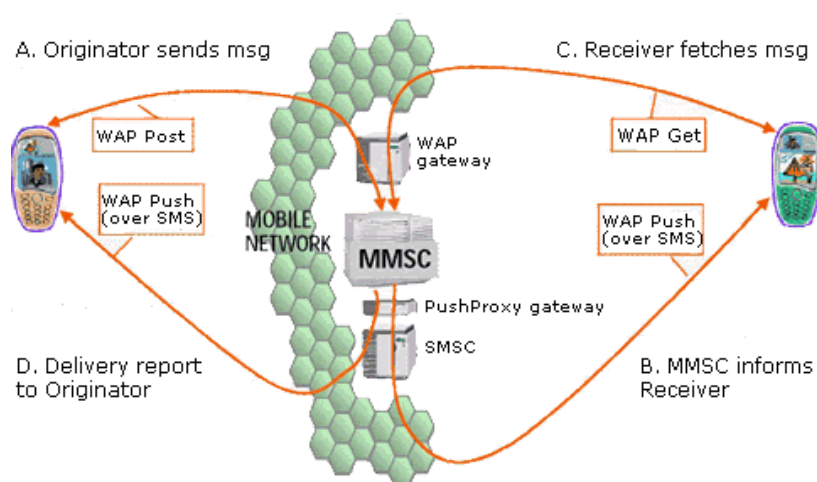
Pode-se observar que uma desvantagem desta arquitetura é a questão de privacidade dos usuários. Semelhante a arquitetura Napster, os usuários podem ser identificados. Neste caso, o número do celular leva ao usuário.

As mensagens do serviço MMS foram escolhidas para transferência dos arquivos, devido ao rápido crescimento do uso dessas mensagens nas operadoras de celulares. Essas mensagens são semelhantes a do serviço SMS (*Short Messaging Service*). A diferença é que o usuário pode emitir uma mensagem MMS com várias informações e o tamanho não é limitado, permitindo voz e texto. O receptor é reconhecido usando o número de MSISDN ou um endereço de e-mail. Apesar do MMS ser considerado como tendo tamanho ilimitado, as limitações de capacidade e de tamanho máximo da memória do telefone móvel restringem o tamanho da mensagem. Entretanto, poderia se usar um software que converteria arquivos grandes em menores, para reverter esta limitação.

Uma das conclusões obtidas, no capítulo 2, foi a necessidade de menores preços para motivar o uso da estrutura de rede celular para acessar serviços P2P-Movél. Dessa forma, considerando a existência de bons preços, esta arquitetura seria boa para as operadoras, quando se avalia o crescimento no uso de MMS gerado pelo compartilhamento de arquivos.

No entanto, seria ruim para os proprietários de conteúdo, se não for utilizada alguma maneira de garantir a propriedade intelectual (uso de DRM, por exemplo).

Apesar da referência [16] apresentar como uma arquitetura P2P para compartilhamento de arquivos utilizando a estrutura da rede celular, a Figura 26 contradiz a comunicação P2P, por meio de um exemplo de mensagem MMS enviada de um usuário para outro. É possível observar que a comunicação não ocorre diretamente, porque as centrais SMS e MMS aparecem como entidades intermediárias durante a comunicação. O receptor da mensagem primeiramente recebe uma notificação da mensagem MMS (indicado como B, na figura) e daí, se usuário desejar baixar o arquivo, conecta-se ao MMSC (Central MMS) e efetua o *download* do arquivo (indicado como C). Em seguida a origem recebe uma confirmação da entrega (indicado como D) [37].



Fonte: referência [37]

Figura 26 Exemplo de transação MMS

4.4.2.2 Utilizando um *Peer* Intermediário

Inicialmente esta arquitetura pode parecer uma contradição do paradigma P2P, mas o *peer* intermediário é uma solução para permitir que dispositivos móveis com baixa restrição de recursos, possam compartilhar arquivos indiretamente. Isso porque o *peer* intermediário deverá ser responsável por gerenciar a transferência de arquivos da rede P2P híbrida, assim como ser responsável por escolher o protocolo e tecnologia de acesso adequado para a transmissão dos dados.

Nesta arquitetura, os *peers* móveis precisam estar associados a um *peer* intermediário. O *peer* intermediário, na Internet, compartilha e busca recursos entre todos os outros *peers* intermediários. Quando um dispositivo móvel deseja compartilhar arquivos, ele deve transferir seus arquivos para seu *peer* intermediário, que irá compartilhar com os outros. Pode-se observar que existe a comunicação direta entre os *peers* intermediários, mas não existe a comunicação direta entre os *peer* móveis.

O *peer* intermediário é caracterizado por possuir um software capaz de abrir qualquer tipo de conexão com um dispositivo móvel, tais como infravermelho ou *Bluetooth*, e de disponibilizar funcionalidade de busca e de compartilhamento de arquivos, que também serão intermediadas pelo software [16]. Adicionalmente, é importante que o *peer* intermediário seja capaz de realizar conversões de protocolos fornecidos para os aparelhos restritos (tais como HTTP, MMS, SMS) e os protocolos da rede fixa (tais como TCP/IP, UDP) utilizados pelos *peers* intermediários para transferir informações de controle e de coordenação.

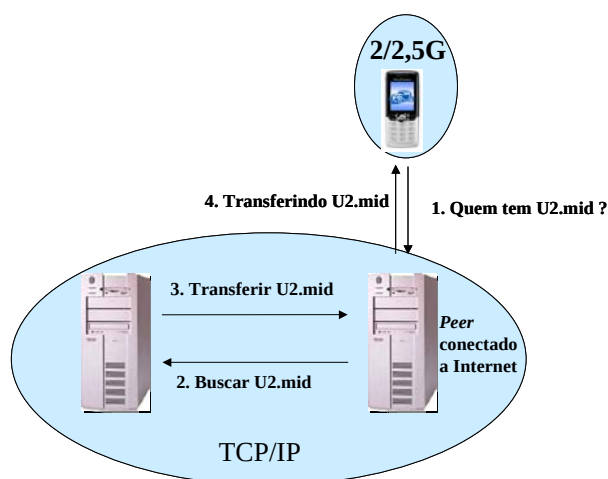


Figura 27 *Peer* intermediário

Na Figura 27, é exemplificada a busca de arquivos por um *peer* móvel utilizando a rede de *peers* intermediários os quais utilizam, entre eles, o protocolo TCP/IP para realizar a busca e transferência de arquivo. Primeiramente, o *peer* intermediário de um *peer* móvel solicita um arquivo, em seguida o *peer* intermediário se comunica de maneira P2P com outros *peers* intermediários que após encontrar o arquivo, transfere o arquivo para o seu *peer* móvel, utilizando a tecnologia de acesso e protocolo adequados. A entidade intermediária, também pode realizar alterações no arquivo, antes de transferi-lo. Um exemplo seria a redução da resolução de uma figura. Dessa forma, o arquivo seria adaptado para as restrições de recursos de um determinado equipamento.

Algumas desvantagens presentes nesta arquitetura são: possivelmente o *peer* intermediário poderá apresentar um atraso significativo, poderá apresentar falta de privacidade, visto que os *peers* dependerão existencialmente dele, esta arquitetura também não apresenta escalabilidade, porque o crescimento no número de *peers* móveis dependerá da capacidade dos *peers* intermediários e a perda da autenticidade, devido à possibilidade do *peer* intermediário modificar o arquivo, no entanto, esta última também pode ser visto como uma vantagem para melhorar a performance.

4.4.2.3 Extensão do eDonkey

Os sistemas P2P para compartilhamento de arquivos em ambiente móvel das seções 4.4.2.1 e 4.4.2.2 preservam os dois requisitos da seção 4.4.2 (coordenação ou mecanismos de mediação de recursos e funções de controle), mas não atendem o mínimo desejo das operadoras de rede móvel, que é ter o controle do tráfego e conteúdo dos dados. Dessa forma, esta seção descreve uma proposta de uma arquitetura para melhorar a performance de uma rede P2P com dispositivos móveis, assim como atender também esse requisito das operadoras.

Para isso, foi proposta em [13] a criação de uma arquitetura P2P baseada no protocolo de compartilhamento de arquivos eDonkey, devido a robustez e popularidade dessa rede. A extensão da arquitetura eDonkey foi realizada por meio da modificação dos servidores de índices, adição de *peers caching* e de *peers crawler*, como pode ser isto na Figura 28.

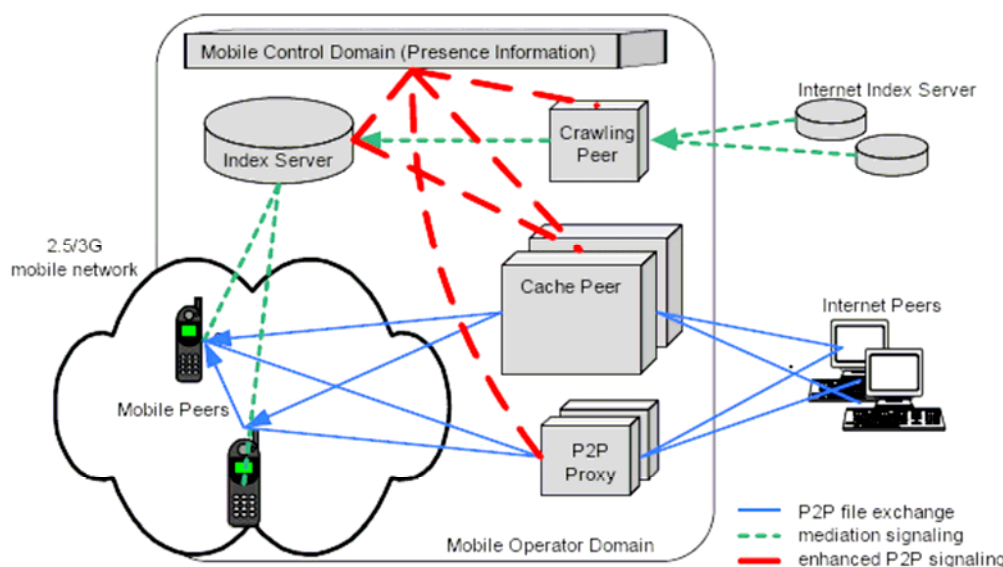


Figura 28 Visão geral da arquitetura P2P-Móvel

O protocolo eDonkey foi projetado como um protocolo de rede P2P híbrida. Esse protocolo permite que um recurso (filme, música e outros) possa ser baixado utilizando vários recursos (várias máquinas, assim como vários fragmentos do filme). A coordenação entre as várias fontes é um tanto difícil, uma vez que os arquivos podem ser modificados ou renomeados. Também é necessário o controle das cópias e fragmentos dos arquivos.

Os servidores de índices, na arquitetura eDonkey, fornecem dois principais serviços: buscar por nome e requisitar uma fonte. Uma das modificações nesta arquitetura foi adição de novas funcionalidades aos servidores de índices: monitorar todos os recursos conhecidos no domínio da rede sem fio, identificar os arquivos mais populares, emitir uma sinalização para o *peer cache* informando a frequência de acesso a um recurso e alterar a mensagem de requisição de recurso, se for identificado que o recurso está disponível em um *peer cache*.

Quando os servidores de índices são requisitados para buscar um recurso, eles identificam a frequência de acesso ao recurso pelo identificador (ID) e decide considerar a popularidade desse recurso, ou seja, é mais relevante a quantidade de diferentes IDs dos *peers* que estão em busca do recurso, do que a quantidade total de requisições de um recurso. Isso porque o tempo de transmissão e tamanho do arquivo pode influenciar na quantidade de solicitações de requisições de um *peer* para um mesmo recurso.

A sinalização enviada pelo servidor de índice para o *peer cache* permite que este *peer* decida, se o recurso deve ser inserido, ou retirado do conjunto de arquivos mais populares na rede. Dessa forma, o *peer cache* armazena as informações mais populares da rede, para reduzir o uso da cara interface aérea dos dispositivos móveis, assim como diminuir o tráfego na rede P2P.

O *peer cache* foi projetado para fornecer rápidas respostas ao *peers* e menores filas de espera por recursos. Então, o *cache* irá permitir um número grande e significativo de *uploads* simultâneos da rede para os *peers* móveis que estão na margem da rede P2P. Isso porque os *peers* móveis irão receber fragmentos das instâncias do *peers caching* que também estão na rede. As simulações de [13] mostram que isto pode reduzir o tráfego da rede com dispositivos móveis e diminuir o tempo para baixar um arquivo.

Os *peers crawling* dão suporte aos servidores de índices com os recursos não conhecidos no domínio da rede móvel, o que garante o controle de todo o conteúdo que está sendo compartilhado pelos *peers* de uma operadora. Se primeiro servidor de índice não encontrar o resultado de uma consulta (busca de um recurso), o *peer crawling* automaticamente conecta a outro servidor de índice e realiza uma nova consulta no outro servidor. Dessa forma, a entidade *peer crawling* é essencial para permite que um *peer* móvel acesse o conteúdo de *peer caching* de outro domínio, uma vez que os outros servidores de índice não podem distinguir os *peers caching*. Portanto, esta extensão do eDonkey atende os requisitos de controle do tráfego e conteúdo dos dados na rede das operadoras, para o caso de cada servidor de índice ser uma operadora de celular.

Para maximizar os benefícios das modificações da arquitetura eDonkey, os *peers* móveis devem se conectar ao melhor servidor de índice. A entidade *crawler* é usada para realizar a coordenação entre o servidor de índice de um domínio móvel, com os servidores de índices na Internet. O servidor de índice solicita um recurso não conhecido ao *crawler*, que encaminha o meta-dado para os servidores de índices na Internet. Dessa forma, qualquer recurso disponível dentro da comunidade global eDonkey pode ser localizado e acessado.

A busca por recursos a partir de *peers* móveis, geralmente possui uma pequena alcançabilidade e taxa de transferência. Além disso, os *peers* móveis geram tráfego duplicado (a informação percorre o caminho GGSN para rede eDonkey na Internet e o caminho inverso) comparado com a transferência de um recurso a partir dos super-nós da rede. Portanto, essa extensão da arquitetura eDonkey melhora muito a performance, porque evita baixar os recursos a partir dos *peers* móveis, que estão nas margens da rede [13].

A proposta desse trabalho acaba por permitir, assim como priorizar o uso assimétrico das tecnologias de acesso, o que é comum na Web. Provavelmente, é interessante utilizar esta

estratégia em benefício da performance, mesmo de certa forma descaracterizando a tradicional rede P2P que utiliza a comunicação simétrica.

4.4.2.4 Comparações entre as Arquiteturas para Compartilhamento de Arquivos

Todas as arquiteturas de propósito específicas para aplicação de compartilhamento de arquivos descritas, na seção 4.4.2, podem ser classificadas como arquitetura P2P-Móvel híbrida. Isso não implica que o compartilhamento de arquivos só pode ocorrer neste tipo de arquitetura, porque a arquitetura Proem que é P2P-Móvel pura também permite a implementação de aplicações de compartilhamento de arquivos em uma rede móvel *ad-hoc*.

A desvantagem das arquiteturas da seção 4.4.2.1 e 4.4.2.2 é a quebra quase completa do paradigma P2P, porque não permite a comunicação direta entre as aplicações móveis. Essas arquiteturas não permitem a comunicação P2P entre os *peers* móveis, porque priorizam atender os requisitos das operadoras, especificamente, a arquitetura da seção 4.4.2.2, quando utiliza HTTP, como protocolo da camada de transporte, apresenta estrutura semelhante à arquitetura da primeira versão de JXME.

A extensão da arquitetura eDonkey (seção 4.4.2.3) é uma abordagem inovadora para as arquiteturas de P2P-Móvel híbrida e permite a comunicação P2P entre os *peers* móveis. Esta extensão também não permite a comunicação direta entre *peer* móvel e fixo, devido à existência de *proxy*. A comunicação direta entre os *peers* são evitadas, por motivo da adição de *peers caching* e o *peer crawler*. Desse modo, esta proposta também quebra parcialmente o conceito P2P. No entanto, existe a contribuição para melhorar a performance de aplicações P2P que utilizam arquiteturas P2P-Móveis. Além disso a contribuição é mais relevante para cenários em que se utilizam dispositivos móveis e aplicações compartilhamento de arquivos.

Portanto, a quebra do conceito P2P pode ser considerada como um preço aceitável, visto que a maior preocupação de P2P-Móvel é lidar com os problemas inerentes a ambientes sem fio.

4.5 Considerações Finais

Neste capítulo, foram apresentadas arquiteturas P2P-Móvel existentes. Esta pesquisa teve como resultado o levantamento de vantagens e desvantagens das abordagens utilizadas pelas arquiteturas, por meio de comparações. Assim como, novas entidades foram identificadas para melhorar a performance dos *peers* em uma rede P2P-Móvel, tais como:

- *Proxys* para converter protocolos, geralmente, para um formato binário;
- *Node gateway* para aumentar a alcançabilidade de *peers* e integração de redes P2P;

- *Peers caching* para reduzir o tráfego na rede P2P e diminuir o uso da interface aérea.
- *Peers crawler* para controlar o tráfego e o conteúdo da rede P2P;

A entidade *proxy* foi encontrada na arquitetura da seção 4.3, na arquitetura de JXTA, quando os *peers* móveis utilizam a versão 1.0 de JXME, e em arquiteturas específicas para compartilhamento de arquivos (da seção 4.4.2.2 e 4.4.2.3).

A entidade *node gateway* foi encontrada na arquitetura da seção 4.3 e da seção 4.1.1 (Mobile CHEDAR). Nesta última, essa entidade é denominada *gateway* CHEDAR/Mobile CHEDAR. A arquitetura Mobile CHEDAR foi considerada em [28] e [29] como uma arquitetura P2P-Móvel pura, mas seria possível contradizer esta classificação, apenas quando existir pelo menos uma entidade *node gateway* na rede. Isso porque, no início deste capítulo, uma arquitetura será P2P-Móvel, se todos os nós possuírem a mesma funcionalidade. No entanto, o *node gateway* possui funcionalidade diferente de todos os outros *peers*. Dessa forma, quando existir um *node gateway*, a arquitetura Mobile CHEDAR deveria ser classificada como P2P-Móvel híbrida, apesar de está utilizando uma rede MANET na camada física.

Os super-nós das arquiteturas P2P híbridas (tais como, JXTA e X-Peer) também poderiam ser vistos como entidades *gateway*, visto que eles, geralmente, possuem o objetivo de permitir a alcançabilidade dos *peers* que estão sob NAT e/ou *firewall*.

As entidades *peers caching* e *peers crawler* foram encontradas apenas na extensão da arquitetura eDonkey da seção 4.4.2.4.

Também foi identificado que a maioria das arquiteturas utiliza XML, o que leva a pensar na provável ineficiência dessas arquiteturas, visto que os trabalhos sobre X-Peer e PnPAP relataram problemas relacionados a performance, quando se utiliza XML em equipamentos restritos. No entanto, a proposta de três possibilidades de implementação para *proxy* da seção 4.3 pode ser uma alternativa para as arquiteturas que utilizam XML.

Em relação à camada de rede, foi possível identificar que sistemas P2P-Móvel puro utilizam o modo de conexão *ad-hoc*, enquanto sistemas P2P-Móvel híbridos utilizam o modo de conexão de infra-estrutura. É possível que isto seja uma característica desses sistemas, porque a performance de um sistema P2P-Móvel puro diminui à medida que a rede cresce, enquanto P2P-Móvel híbrido tem objetivo de transmitir dados a longa distância.

No próximo capítulo, serão apresentadas propostas para modificação da arquitetura X-Peer.

5 Modificações na Arquitetura X-Peer

Para esta monografia, foram identificadas três propostas para serem realizadas na arquitetura X-Peer, com diferentes níveis de dificuldades:

1. A primeira alteração faz com que aplicações P2P e P2P-Móvel da infra-estrutura X-Peer possam se comunicar utilizando UDP, mesmo quando estiverem sob NAT e *firewall*, permitindo o desenvolvimento de aplicações em tempo real. Um exemplo é VoIP;
2. A segunda proposta corresponde à adição de novos serviços que permitam desconexões (temporárias) dos *peer* móveis. Desconexões ocorrem com frequência em um ambiente móvel, mas a atual rede X-Peer prejudicava a performance, porque a solicitação do serviço Join sempre é necessária para continuar a usar cada aplicação em que os *peer* móveis anteriormente estavam conectados;
3. A terceira proposta modifica a arquitetura X-Peer para que ela permita o desenvolvimento de aplicações que utilizam LBS.

Respectivamente, cada uma destas propostas será apresentada nas próximas seções.

5.1 Proposta e Implementação da Transmissão de Dados com UDP

A arquitetura Proem (seção 4.1.2) permite a comunicação entre os *peers* móveis utilizando qualquer protocolo da camada de transporte, no entanto, não foi identificada a resolução de problemas relacionados a *peers* sob NAT, de forma que seja transparente para a aplicação.

Até então, a arquitetura X-Peer fornece um módulo denominado Módulo P2P. Este módulo abstrai o problema de comunicação direta (presença de *firewalls* e NATs) utilizando o protocolo TCP. Isso porque, o módulo se encarrega de testar a conectividade entre os *peers* e aplicar algumas técnicas para permitir a conexão entre eles. Dependendo da situação de conexão de cada *peer* a comunicação entre eles pode ocorrer de duas formas diferentes.

As possíveis situações de conectividades de um *peer* A e outro *peer* B são:

1. **B está acessível por A.** Esta situação pode ocorrer para qualquer configuração em A (NAT e/ou *firewall* ou nenhum) e para B não estando sob *firewall* nem NAT.
2. **B não está acessível por A, mas A está acessível por B.** Esta situação pode ocorrer para B estando sob *firewall* e/ou NAT e A não estando sob *firewall* nem NAT.
3. **B não está acessível por A nem A está acessível por B.** Esta situação pode ocorrer quando tanto o *peer* A quanto o *peer* B estão sob *firewall* e/ou NAT.

Esse módulo original ao identificar as situações 1 e 2 permite que a troca de mensagens seja realizada diretamente por meio de conexões *peer-to-peer* com TCP (Figura 29). Na situação 3, A e B não conseguem trocar mensagens de forma direta, portanto o módulo é responsável por controlar a comunicação indireta, por meio de serviço Deliver disponibilizados pelo X-Peer (Figura 30).

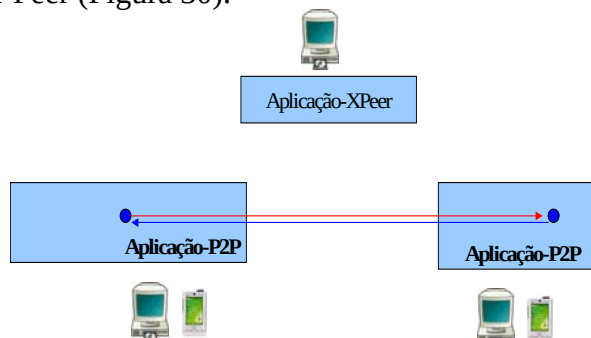


Figura 29 Comunicação direta com TCP

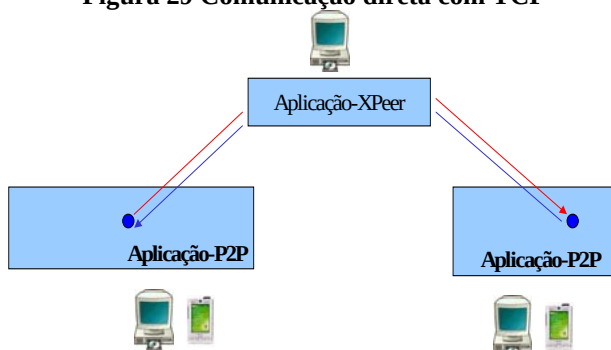


Figura 30 Comunicação indireta com TCP

No entanto, existem aplicações que requerem o uso do protocolo UDP. Por isso, a extensão deste módulo P2P foi proposta e implementada, para que este abstraia a presença de NAT, quando se estiver desenvolvendo um aplicativo que utiliza o protocolo UDP. Outra função deste novo módulo P2P é possibilitar a transmissão de dados para o X-Peer utilizando UDP. Nos parágrafos a seguir, a implementação da solução será detalhada.

Em [38] são identificadas duas maneiras de permitir a comunicação P2P entre *peers* que estão sob NAT: *hole punching* implementado neste trabalho e *relay* que são *peers* responsáveis por encaminhar mensagens. Esta última é uma boa técnica a ser utilizada quando a primeira falhar por motivos de *firewall*.

A implementação da técnica *hole punching* permite que possa existir a comunicação direta entre os *peers*, quando estes estiverem sob NAT (Figura 31). Isso evita a sobrecarga dos nós X-Peer, o consumo da largura da banda e reduz atraso na comunicação.

A técnica de *hole punching* corresponde a troca de mensagens UDP utilizando ip:porta públicos e privados entre dois *peers* e ao mesmo tempo.

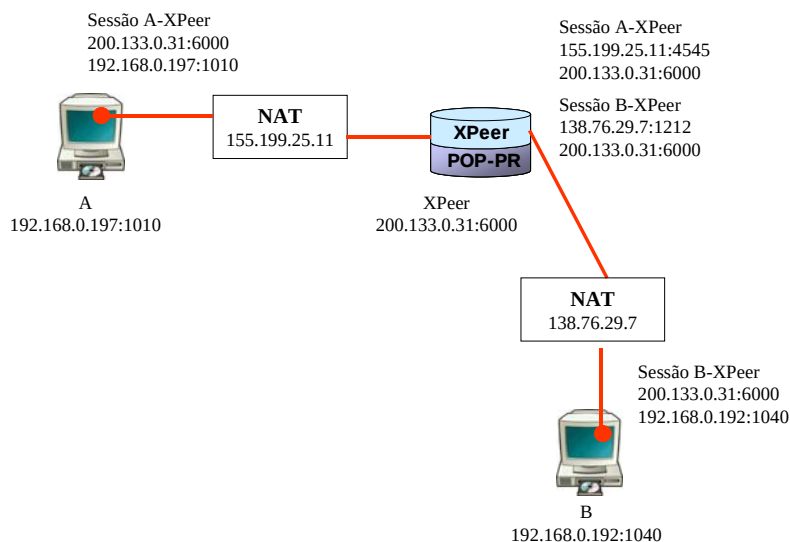


Figura 31 Componente refletor na rede X-Peer

Para implementar a técnica *hole-punching*, no Módulo P2P, foi necessário adicionar um componente acoplado em cada X-Peer. Este componente foi denominado RemoteUDP, que tem a função de enviar mensagens Return para o *peer* que enviou uma mensagem UDP. A mensagem Return contém o host:porta público e privados do *peer* que enviou a mensagem UDP.

Outro componente denominado P2PUDP foi implementado para ser utilizado pelas aplicações. Este utiliza a técnica *hole-punching* para permitir a abstração da existência de NAT na comunicação entre os *peers*. O componente foi estruturado de forma a permitir a implementação de aplicações P2P tanto para *peers* fixos, como móveis. O P2PUDP utiliza inicialmente o RemoteUDP para saber o ip:porta público e privado de cada aplicação. Posteriormente, cada *peer* envia uma rajada de mensagens UDP solicitando *hole-punching*. No momento em que um *peer* identifica o recebimento de uma mensagem, a técnica é finalizada. Após essa identificação, os *peers* podem realizar a comunicação P2P utilizando UDP, mesmo existindo NAT entre os *peers*.

Pode-se observar que esta proposta contempla apenas a abstração de cenários em que os *peers* estão sob NAT. A técnica de *relays* identificada anteriormente poderia ser utilizada em cenário em que existem *firewall*. No entanto, na seção 6.5.3 foi identificada uma provável ineficiência de UDP da API de J2ME. Por isso, a implementação da técnica de *relay*, que irá abstrair a comunicação entre *peers* sobre *firewall*, foi adicionada como trabalho futuro.

5.2 Proposta de Novos Serviços para P2P-Móvel

Durante a avaliação de desempenho (detalhada no capítulo 6), foi identificado que o X-Peer não oferece um adequado ambiente para aplicações P2P-Móvel, quando os dispositivos móveis que possuam alta ocorrência de desconexões, porque a arquitetura realiza automaticamente o serviço Leave, no momento em que ocorre a desconexão de uma

aplicação com a rede X-Peer. Esse chamado automático do serviço Leave é realizado com a finalidade de evitar a permanência de conexões abertas, causada pela implementação inadequada de aplicações, ou mesmo a desconexões de usuários que não re-conectaram para solicitar o serviço Leave.

Como consequência da automática chamada do serviço Leave, as aplicações P2P-Móvel também precisam solicitar o serviço Join, mesmo quando ocorrem desconexões por questões de mobilidade, agravando a performance de qualquer aplicação. Portanto, esta seção tem como objetivo descrever novos serviços para o X-Peer, a fim de que as aplicações P2P-Móvel possam explorar essa arquitetura de maneira mais eficiente, principalmente quando existir alta ocorrência de desconexões. Esses novos serviços estão destacados na Figura 32.

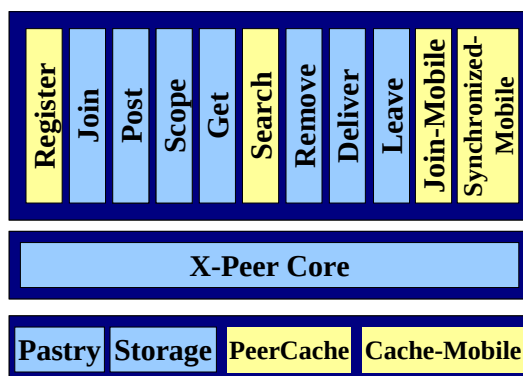


Figura 32 Modificação da arquitetura X-Peer para atender a mobilidade

O componente adicional (*Cache-Mobile*) é diferente da proposta de *peer cache* da seção 4.4.2.3, pois ele não tem o objetivo de evitar as comunicações P2P. O *Cache-Mobile* armazenará apenas as informações transmitidas para o *peer* móvel, durante um intervalo de tempo em que o *peer* móvel estiver ausente da rede. Um exemplo é o cenário em que *peer* está fazendo *download* de um arquivo e ocorrer sua desconexão devido à mobilidade. Neste cenário, o *Cache-Mobile* receberá os dados que seriam enviados para o *peer* móvel.

O segundo novo componente (*PeerCache*) tem a funcionalidade semelhante ao *peer cache* da seção 4.4.2.3. No entanto, o *PeerCache* dependerá do interesse da aplicação em utilizá-lo ou não para melhorar a performance. Por isso, é proposto que os serviços Register e Search sejam modificados. A aplicação utilizará Register para informe a rede X-Peer que poderá utilizar o *PeerCache*. Caso a aplicação tenha interesse em utilizar o *PeerCache* na busca, um campo da mensagem Search é preenchido.

Desse modo, quando o X-Peer identifica a necessidade de busca com *PeerCache*, primeiramente ele verificará se a aplicação permite utilizar o novo componente. Em seguida, se o conteúdo estiver no *PeerCache*, a rede X-Peer retorna o local do *PeerCache*, caso contrário o X-Peer retornará o local do *peer* que possui o conteúdo. Neste último caso, a informação será repassada para o *peer* solicitante do serviço Search e para o *PeerCache*, ou seja, este componente também irá buscar o novo conteúdo.

O uso do *PeerCache* é proposto como opcional, porque não faz sentido o uso deste componente para todas aplicações. Aplicação de troca de mensagem é um exemplo em que o *PeerCache* nunca deve ser usado, mas a aplicação de compartilhamento de arquivos deveria utilizar o *PeerCache*.

A justificativa para o uso de *PeerCache* está relacionada com um dos problemas atuais do paradigma P2P que sempre realiza a transferência fim-a-fim até em situações em que o arquivo se encontra no *cache* local. Os sistemas de *cache* existentes foram desenvolvidos para trabalhar com objetos Web. Por isso poderia-se utilizar o novo componente, dependendo da aplicação, para melhorar o desempenho do X-Peer. Objetivo é procurar soluções para minimizar o tráfego.

O serviço *Join-Mobile* permite que o usuário informe ao X-Peer que esta executando uma certa aplicação em um dispositivo móvel. Isto significa que o usuário está informando previamente que poderão ocorrer desconexões devido à mobilidade ou falta de energia fornecida pela bateria. A proposta é que a mensagem, enviada pelo dispositivo para solicitar este serviço, seria semelhante à mensagem que permite a solicitação do serviço Join, a diferença é que o usuário, de maneira complementar, informar também um *timeout*, ou seja, o tempo que poderá ficar ausente devido à mobilidade ou falta de energia fornecida pela bateria. Dessa forma, a rede X-Peer não irá desconectar o usuário da aplicação de forma automática. Além disso, se o dispositivo conseguir entrar na rede X-Peer dentro do intervalo de tempo informado pelo *Join-Mobile*, a aplicação não precisará enviar uma nova mensagem *Join-Mobile* ou Join.

O *timeout* informado na mensagem *Join-Mobile* é utilizada pelo componente *Cache-Mobile* para que ele atue como receptor de mensagens para o *peer* móvel desconectado, durante esse intervalo de tempo.

O serviço *Synchronized-Mobile* é uma opção que permitirá o usuário da aplicação recuperar toda informação armazenada no *Cache-Mobile*. Uma observação importante é que este serviço só poderá ser utilizado para aqueles usuários que entraram na rede X-Peer utilizando o serviço *Join-Mobile*. Outra informação relevante é que só existirá informação no componente *Cache-Mobile*, se o tempo de ausência do *peer* móvel não exceder o *timeout* informado pela mensagem correspondente ao serviço *Join-Mobile*.

5.3 Proposta LBS (Location Based Services)

LBS é um tipo de SIG (*Sistema de Informação Geográfica*) que tem a sua principal função de prover conteúdo baseado em localização.

A motivação, para a disponibilização deste serviço, está relacionada com o crescimento do uso de LBS em arquiteturas cliente/servidor. Em São Paulo, é comum, táxis estarem munidos com equipamentos de navegação capazes de traçar rotas por menor caminho e, até mesmo, de determinar rotas alternativas para desviar de tráfegos congestionados. Segundo

o *International Data Corporation* (IDC), a receita gerada somente pelos serviços de localização deve ser superior a US\$ 5 bilhões em 2004 [22].

A VIVO, empresa prestadora de serviços de telecomunicações móveis, está disponibilizando aplicações com serviço de localização de alta precisão que utilizam tecnologia sem fio para clientes de São Paulo e do Rio de Janeiro. Baseado na tecnologia gpsOne, da Qualcomm, que faz utilização de satélites GPS e Estações Rádio Base (ERBs), da rede CDMA2000 1x da VIVO, o sistema oferece resultado com precisão de 5 a 50 metros. Atualmente ela fornece três aplicativos: **VIVO Localiza**, **VIVO Aqui Perto** e **VIVO Onde Estou?**:

- **VIVO Localiza** é um serviço de localização de pessoas, que identifica com precisão a posição de um usuário ou a localização de um ponto de interesse a partir do próprio celular.
- **VIVO Aqui Perto** é um aplicativo desenvolvido em parceria com a Webraska para procurar estabelecimentos comerciais, como bares, restaurantes e cinemas.
- O aplicativo **VIVO Onde Estou?** indica a localização do celular, com logradouro, número, bairro e cidade do próprio usuário.

Em relação ao preço, o aplicativo **VIVO Localiza** estipula valores variados (R\$ 9,99 por pacote de utilização - para cinco localizações; três localizações custam R\$ 7,99 e uma localização por R\$ 5,99) mais tráfego de dados. **VIVO Aqui Perto** tem a tarifa mensal que possibilita a utilização do serviço para 5 localizações por R\$ 9,99 [23].

Existem basicamente duas maneiras de implementar o LBS: utilizar soluções baseadas em *handset* (*handset-based*) ou rede (*network-based*), que dependem da base de informações de clientes e da base de informações geográfica. As soluções *handset-based* especificam que o dispositivo terá que ter um receptor GPS (*Global Positioning System*) para fazer o processo de localização. As soluções *network-based* fazem com que a localização seja feita de acordo com a área de cobertura das células. Esta última solução é disponibilizada por operadoras de telefonia celular (por exemplo, a empresa VIVO). A união de soluções *handset-based* com *network-based* proporcionou o surgimento de uma terceira solução denominada híbrida.

Diante deste crescimento de aplicativo e serviços LBS que utilizam este tipo de SIG, seria interessante adicionar a funcionalidade de compartilhamento de informação geográfica no X-Peer. Portanto, esta seção apresentará a proposta de modificação da arquitetura, no entanto, será dependente de uma rede celular ou equipamento GPS, porque seria difícil sem dados mais precisos, como os fornecidos pelas operadoras ou GPS.

Para atender este novo requisito, seria adicionado um novo protocolo denominado *XLocalizationProtocol* e formado por um conjunto de mensagens para atender o novo serviço denominado XLBS – serviço de localização no X-Peer. Este protocolo seria formado por três mensagens: *XPeerAround*, *RegisterLBS* e *SearchLBS*. A primeira seria utilizada por dispositivos móveis, para qual o X-Peer mais próximo dele; o segundo tipo de

mensagem permitirá que as aplicações informem o interesse em compartilhar a localização do dispositivo; a terceira mensagem permitirá que o dispositivo recupere serviços, aplicações e *peers* baseados na localização dos outros *peers* que também estão compartilhando a localização.

Quanto à estrutura do X-Peer, seria adicionado um componente de banco de dados geográfico (BDGEO) em um ou mais nós X-Peer para armazenar e consultar as informações geográficas coletadas por qualquer nó X-Peer. Dessa forma, a Figura 33 mostra a modificação na arquitetura X-Peer com a adição do serviço XLBS (serviço de localização realizado pelo X-Peer) e o componente BDGEO que está destacado, porque ele pode estar presente em apenas um X-Peer.

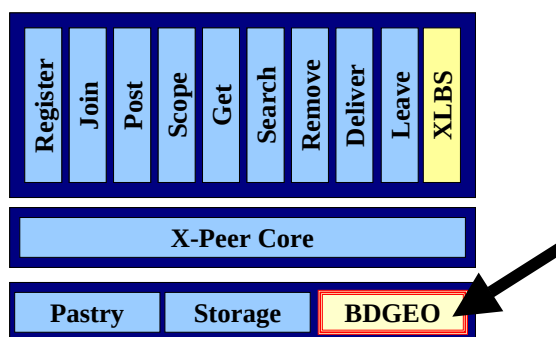


Figura 33 Modificação na arquitetura X-Peer para atender LBS

A Figura 34 exemplifica o uso da mensagem *RegisterLBS*. Os *peers* poderão utilizar, por exemplo, um receptor GPS para obter a localização. Em seguida, os *peers* enviarão a mensagem *RegisterLBS* com as informações da posição geográfica, para qualquer nó X-Peer. Este último encaminhará para o nó X-Peer com o componente BDGEO.

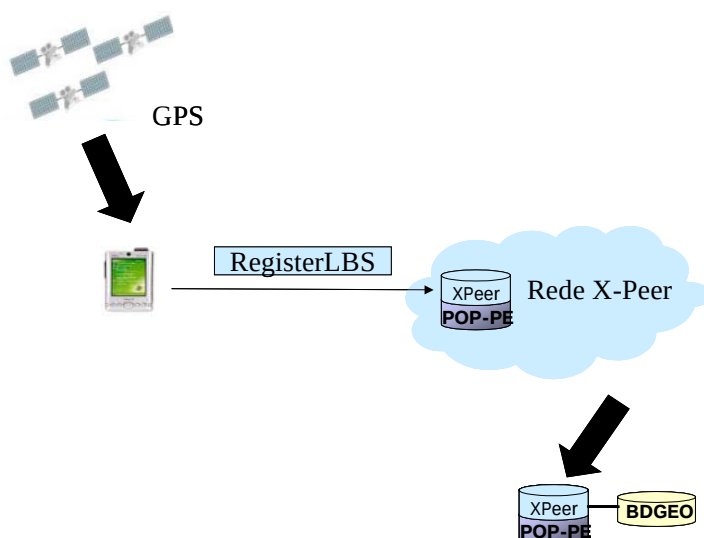


Figura 34 Uso do serviço XLB com GPS

5.4 Contribuições

Neste capítulo, foram apresentadas propostas para melhorar a arquitetura X-Peer em relação a *peer* móveis. Ele foi resultado do estudo detalhado sobre P2P-Móvel, por meio do levantamento de restrições e pré-requisitos dos dispositivos móveis, das aplicações, das arquiteturas e da avaliação de desempenho que será detalhado no próximo capítulo.

É possível identificar as seguintes contribuições mais relevantes, deste capítulo:

- Proposta e implementação da extensão do módulo P2P. Esta extensão teve como objetivo permitir a alcançabilidade de aplicações que estão sob NAT e utilizam UDP na camada de transporte.
- Proposta de modificação da arquitetura X-Peer, para melhor atender aos *peers* móveis, por meio de dois novos componentes (*Cache-Mobile* e *Peer-Cache*) e adição de serviços para solucionar a dificuldade de aplicações P2P-Móvel em utilizar freqüentemente o serviço Join.
- Proposta de modificação da arquitetura X-Peer para disponibilizar um serviço de LBS. Este novo serviço permitirá que os *peers* registrem e compartilhem dados geográficos, assim como a busca de nós X-Peer, aplicações, *peers*, entre outros recursos por meio da localização geográfica.

No próximo capítulo, serão apresentadas a avaliação de desempenho e a aplicação desenvolvida para realizar as medições nessa avaliação.

6 Avaliação de Desempenho

Este capítulo apresenta a avaliação de desempenho de aplicações P2P em dispositivos móveis. A seção 6.1 descreve a motivação e os trabalhos relacionados. A seção 6.2 apresenta o objetivo geral e específico da avaliação de desempenho. A seção 6.3 apresenta o ambiente da avaliação. A seção 6.4 indica os parâmetros e as métricas. A seção 6.5 descreve o aplicativo desenvolvido para coletar as métricas. A seção 6.6 fornece os resultados de cada cenário definido na mesma e a seção 6.7 avalia os resultados em relação a aplicações P2P-Móvel. A seção 6.8 descreve as dificuldades encontradas durante a avaliação e a seção 6.9 conclui a avaliação apresentada neste capítulo.

6.1 Motivação e Trabalhos Relacionados

Muitas das arquiteturas analisadas (no capítulo 4) não apresentam avaliações sobre desempenho e experimentos para validar o uso dessas abordagens inovadoras. Foram encontradas referências relacionadas apenas a arquitetura eDonkey.

Em [13] e [39], é apresentado um modelo de simulação para avaliar a proposta de modificação da arquitetura eDonkey. A extensão desta arquitetura (seção 4.4.2.3) adiciona *peers caching*, *peers crawler* e modifica os servidores de índices. A referência em [13] analisa apenas as estratégias de *caching* para o *peer caching*. Em [39], são avaliados o *peer crawler* e as modificações no servidor de índices.

O artigo [15] é outra referência que avalia a questão de aplicações de compartilhamento de arquivos utilizando dispositivos móveis. Esse trabalho realiza uma medição por meio de um software que captura pacotes na comunicação entre dispositivos móveis. O cenário avaliado utiliza a atual arquitetura eDonkey e a tecnologia GPRS durante a medição.

Pode-se perceber que poucos artigos (apenas [13], [15] e [39]) foram publicados em relação à avaliação de desempenho no ambiente P2P-Móvel. Estes trabalhos focam principalmente na análise da viabilidade de aplicações de compartilhamento de arquivos. Assim como, concluíram que este tipo de aplicação é viável em redes sem fio e apresentaram a importância do uso de *caching* para reduzir o atraso na transmissão dos arquivos.

A restrição das avaliações anteriores (apenas o uso da arquitetura eDonkey) foi uma motivação para a realização desta avaliação utilizando outra arquitetura P2P. Outra motivação para esta avaliação de desempenho é saber que os resultados a serem apresentados poderão direcionar e despertar o interesse na criação de novas aplicações, assim como serviços para P2P-Móveis.

6.2 Objetivo

O objetivo geral desta avaliação é estudar a viabilidade da utilização de aplicações P2P em ambientes móveis sem fio utilizando a arquitetura X-Peer.

Para responder ao objetivo geral, foram identificados objetivos específicos enumerados abaixo, e que este capítulo se propõe a responder.

- 1) Comunicação móvel-móvel via TCP: Qual é o impacto no tempo de resposta e vazão se o tamanho da mensagem a ser enviada for aumentada na comunicação direta entre *peers* e na comunicação indireta (usando o X-Peer), quando apenas dispositivos móveis são utilizados? Qual a diferença entre os tempos de respostas, quando a comunicação for direta e indireta?
- 2) Comunicação móvel-fixo via TCP: Qual é o impacto no tempo de resposta e vazão se o tamanho da mensagem a ser enviada for aumentada na comunicação direta entre *peers* e na comunicação indireta, quando se utiliza um dispositivo móvel e uma máquina fixa? Qual a diferença entre os tempos de respostas, quando a comunicação for direta e indireta?
- 3) Comparação dos itens 1 e 2: Comparar os tempos de respostas quando existe a participação de dois *peers* móveis e quando existe a comunicação entre *peer* móvel e fixo.
- 4) Aplicação utilizando UDP: Verificar a viabilidade das aplicações P2P-Móvel que utilizam o novo módulo P2P desenvolvido neste trabalho de graduação (ver seção 5.1).
- 5) Comparação entre TCP e UDP: Realizar a comparação dos tempos de resposta, quando se utiliza UDP e TCP na comunicação direta entre *peers*.
- 6) Avaliar atrasos: Verificar se os atrasos encontrados nos experimentos permitirão usar certas aplicações.

6.3 Ambiente de Avaliação

Esta seção tem a finalidade de descrever como e onde foram realizadas as análises de desempenho. A seguir serão descritas: a metodologia de avaliação e o ambiente disponibilizado para realizar a avaliação.

6.3.1 Metodologia de Avaliação

Modelagem analítica, simulação e medição são técnicas de avaliação de desempenho conhecidas. Dentre elas, a medição foi escolhida, apesar de proporcionar resultados mais lentos e um custo mais alto, quando comparada com um processo de simulação. Porém através da medição será possível responder as perguntas da seção 6.2 com valores reais.

A desvantagem da medição é quanto aos recursos físicos e ao tempo para realização dos experimentos que restringem a quantidade de verificações do desempenho, dificultando a variação da carga e ambientes. No entanto, a medição é considerada uma técnica útil para a análise de sistemas computacionais e requer que o sistema a ser analisado esteja disponível. Para atender a este requisito, um aplicativo (descrito na seção 6.5) foi implementado e a arquitetura X-Peer foi escolhida, sobretudo, devido aos recursos que ela fornece (ver seção 4.2.2). Outro fato que influenciou na escolha da arquitetura X-Peer foi a participação no GT-P2P. A familiaridade com o código fonte facilitou a extensão da arquitetura e também sua avaliação.

6.3.2 Ambiente de Medição

Na avaliação, foram utilizados os recursos operacionais do Grupo de Pesquisa de Redes de Telecomunicações (GPRT). A composição da rede interna do GPRT é heterogênea. Esta rede é formada por microcomputadores, computadores sem fio, estações de trabalho, dois pontos de acesso de rede sem fio, vários servidores com diferentes plataformas operacionais, tais como Windows 2000, Windows XP e Linux.

Quatro máquinas foram disponibilizadas para o desenvolvimento e teste da aplicação, assim como para coleta do experimento. Dentre esses equipamentos, dois dispositivos móveis de mão, CLIE da Sony, com sistema operacional PalmOs v.5.2, CPU i.MXL, memória RAM de 32MB, interface de rede *Wireless LAN* (IEEE802.11b), frequência da banda 2.4 GHz.

O aplicativo para avaliação de desempenho (seção 6.4) foi instalado nos dispositivos móveis e em um computador Pessoal, com sistema operacional Microsoft Windows 2000, processador AMD *Athlon* (TM) XP 2000, memória RAM de 512 MB, interface de rede *Ethernet* e frequência da banda de 2.4 GHz. Em outro computador pessoal com mesma configuração anterior foi instalado o *middleware* X-Peer representando um super-nó da rede X-Peer.

Foi utilizando apenas um ponto de acesso sem fio da marca *DLink AirPlus Xtreme DI-624*. Este equipamento disponibiliza a comunicação sem fio IEEE 802.11g que é compatível com o padrão IEEE 802.11b (utilizado pelos dispositivos móveis, CLIE), frequência da banda 2.4 GHz e taxa de transferência de 108Mbps.

O ambiente de medição utilizado está descrito no diagrama da **Figura 35**.

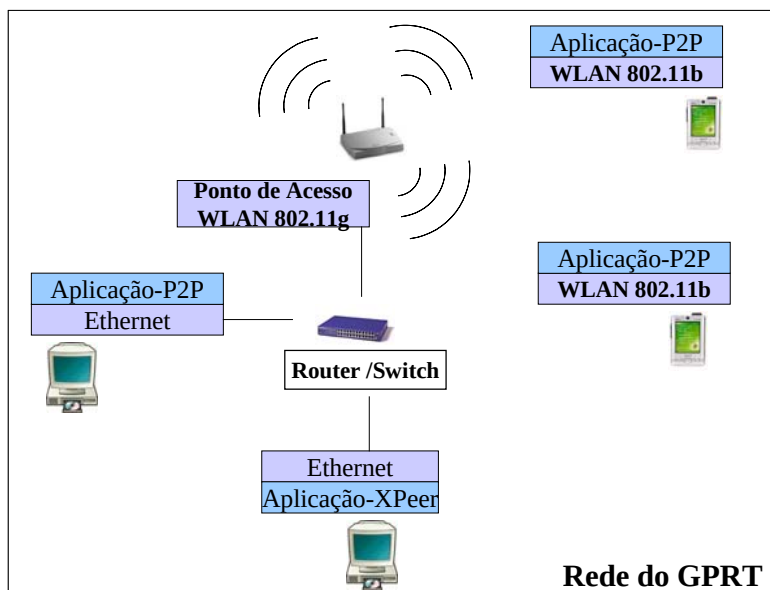


Figura 35 Diagrama do ambiente de medição

Uma breve descrição dos componentes da **Figura 35** pode ser observada a seguir:

- **Aplicação-P2P:** Componente que representa o *software* desenvolvido para medir, coletar e armazenar as métricas. Ele é responsável por enviar e receber as mensagens, assim como ligar e desligar o contador de uma determinada métrica. A aplicação oferece uma interface para configurar os parâmetros de entrada que estão descritos na seção 6.4. Os dados coletados por este componente são posteriormente carregados em uma planilha Excel para gerarem gráficos. Portanto, como pode ser visto na **Figura 35**, o componente foi instalado em todos equipamentos. Cada medição sempre é realizada entre dois determinados equipamentos (entre dois *peers* móveis, entre dois *peers* fixos ou entre um *peer* fixo e um *peer* móvel).
- **Ethernet:** Componente que representa o tipo da tecnologia da interface de comunicação utilizado nos equipamentos do tipo *desktop*.
- **WLAN 802.11b:** Componente que representa o tipo da tecnologia da interface de comunicação utilizado nos dispositivos móveis.
- **Ponto de acesso sem fio WLAN 802.11g:** Componente responsável por distribuir uma banda de conexão *wireless* na rede GPRT. Este componente é compatível com o 802.11b.
- **Router/Switch:** É o equipamento responsável por interligar o ponto de acesso sem fio e as interfaces de rede *Ethernet*.

Neste ambiente, também foram instalados alguns *softwares* necessários para execução, desenvolvimento e testes do aplicativo da seção 6.4. Nos computadores pessoais, foram instaladas as máquinas virtuais (VM – *Virtual Machine*), fornecidas pela SUN, para a execução da aplicação da seção 6.4 e X-Peer que foram desenvolvidas em Java e utilizam a

API J2SE; a ferramenta J2ME *Wireless Toolkit* 2.0 foi instalada para permitir o desenvolvimento da aplicação da seção 6.5 em J2ME e um ambiente de testes por meio de um emulador. A KVM (*Kilobyte Virtual Machine*) fornecida pela IBM foi instalada nos PDAs, para ser possível executar o código desenvolvido com a ferramenta J2ME *Wireless Toolkit* 2.0.

6.4 Parâmetros e Métricas

Parâmetro é tudo aquilo que pode ser configurável em uma avaliação de desempenho. Fatores são os parâmetros que variam entre os experimentos. Níveis são os valores que os fatores podem assumir durante os experimentos. Dessa forma, abaixo serão listados os parâmetros. Apenas o último foi um parâmetro fixo, enquanto os níveis e fatores dos experimentos serão definidos na seção de cenários (seção 6.6).

- Tipo de comunicação: representa o tipo de comunicação entre os *peers*. Quando os dispositivos estão sob *NAT* ou *firewall*, os *peers* utilizam o X-Peer para realizar a comunicação indireta, caso contrário o módulo P2P (ver seção 5.1) permite que os *peers* realizem a comunicação direta.
- Tamanho da mensagem: representa o tamanho em *bytes* do conteúdo da mensagem a ser enviada e recebida entre os dispositivos.
- Equipamento: indica qual o hardware utilizado no experimento.
- Tipo do protocolo: indica qual o protocolo da camada de transporte utilizado.
- Número de super-nós: representa o número de nós X-Peers que formarão um rede P2P. Este foi definido como um parâmetro fixo, existindo apenas um X-Peer na rede local do GPRT. Essa escolha foi limitada por questões físicas para realizar o experimento.

As seguintes métricas são utilizadas para obter resultados das medições através da aplicação para análise de desempenho: tempo de resposta ou RTT (*Round Trip Time*), vazão e latência no X-Peer. Cada uma dessas definições será descrita nos próximos parágrafos.

Tempo de resposta ou RTT. O tempo de resposta $T_{RTT}^{iAB} = T_{final}^{iA} - T_{inicial}^{iA}$ é a medição na camada de aplicação do experimento *i* entre os dispositivos A e B, onde $T_{inicial}^{iA}$ corresponde ao valor do instante anterior ao envio de uma mensagem pelo dispositivo A para o dispositivo B, e T_{final}^{iA} representa o valor do instante da chegada de uma mensagem no dispositivo A. O dispositivo A receberá a mensagem quando o dispositivo B responder a chegada da mensagem de A, por meio do envio de uma mensagem para o dispositivo A. Esta definição pode ser vista na Figura 36.

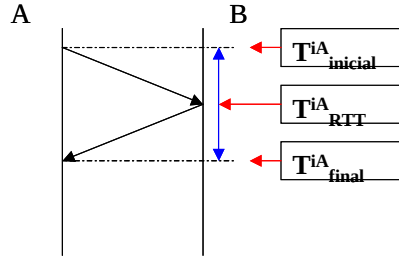


Figura 36 Tempo de resposta ou RTT

Latência no X-Peer: é o tempo que um pacote demora a sair de um nó X-Peer, após a chegada do mesmo pacote. Nesta avaliação, esta métrica é apenas uma estimativa, porque o tempo de transmissão é desprezível, em uma rede local e cabeada, visto que o tempo de transmissão nessa rede é por volta de 1 a 2 ms. Portanto a latência do experimento i é $T^{i}_{Latência} = T^{iXPeer}_{RTT} - T^{iP2P}_{RTT}$, onde T^{iXPeer}_{RTT} é o valor da medição do tempo de resposta quando a comunicação é indireta *peers* fixos, ou seja, utilizam o X-Peer, e T^{iP2P}_{RTT} é o valor da medição do tempo de resposta da comunicação direta entre *peers* fixos.

Vazão nos *peers* móveis: mede a quantidade de mensagens em *bytes* recebida pela aplicação do dispositivo móvel em intervalos de k segundos. A vazão de um experimento i , em um *peer* móvel, é $V^i_x = \{ (B^i_1 / X) / k, (B^i_2 / X) / k, ..., (B^i_{n-1} / X) / k, (B^i_n / X) / k \}$ **mensagens/segundo (msg/s)**, onde B^i_1 é o número de *bytes* coletados no primeiro k segundos, X é o tamanho da mensagem em *bytes*. Desse modo, (B^i_1 / X) representa o número de mensagens recebidas.

É importante ressaltar que cada amostra representa o resultado de uma métrica coletada durante um intervalo de tempo.

6.5 Aplicativo da Avaliação de Desempenho

Atualmente, o GT-P2P disponibiliza uma aplicação para dispositivos móveis denominada *XatMobile*. Esta aplicação não é adequada para essa avaliação, porque não permite o controle das mensagens enviadas. Dessa forma, a falta de controle poderia influenciar nos resultados.

Por isso, uma aplicação, denominada Aplicação-P2P, foi implementada para realizar a avaliação de desempenho. Este aplicativo se comporta de maneira semelhante ao comando **ping**²⁹, porém implementado na camada de aplicação.

A aplicação implementa um protocolo simples, composto de duas mensagens: PING, que representa a mensagem enviada para um equipamento, e PONG, que representa a resposta a mensagem PING.

²⁹ O comando **ping** utiliza mensagens ICMP, portanto é na camada de rede. Dessa forma, não é adequada para esta avaliação porque se pretende avaliar o tempo de resposta no nível da camada de aplicação. Dessa forma, serão considerados os atrasos causados pela máquina virtual e pelos componentes disponibilizados pelo GT-P2P para implementar aplicações P2P que utilizam uma rede X-Peer.

Este aplicativo teve como objetivo coletar as métricas RTT e vazão. Para atender a esse objetivo, foram identificados dois requisitos: o requisito de coleta da métrica RTT (detalhado na seção 6.5.1) e o requisito de coleta da métrica vazão (detalhado na seção 6.5.1).

A aplicação desenvolvida utilizou a arquitetura X-Peer, que foi escrito totalmente em Java. Também foi usada a extensão do módulo de comunicação P2P, implementado neste trabalho de graduação. Este módulo também foi implementado apenas na linguagem Java e restrito a APIs (*Application Programming Interfaces*) J2SE e J2ME. Portanto, foram utilizadas as APIs J2SE, J2ME e do X-Peer, especificamente o pacote que permite utilizar a extensão do módulo de P2P (ver seção 5.1) e módulo de comunicação com X-Peer.

6.5.1 Requisito de Coleta da Métrica RTT

Este é um dos requisitos do aplicativo para avaliação de desempenho. Este requisito tem a intenção de realizar a coleta de várias amostras da métrica RTT ou tempo de resposta. Cada amostra é obtida no instante inicial de saída de uma mensagem PING e o tempo final correspondente à chegada da mensagem PONG, a qual representa a resposta à mensagem PING. Cada uma das mensagens está definida em um byte e elas fazem parte do protocolo de comunicação da aplicação implementada. A troca de mensagens pode ser observada indireta e direta estão representadas respectivamente nas **Figura 37** e **Figura 38**.

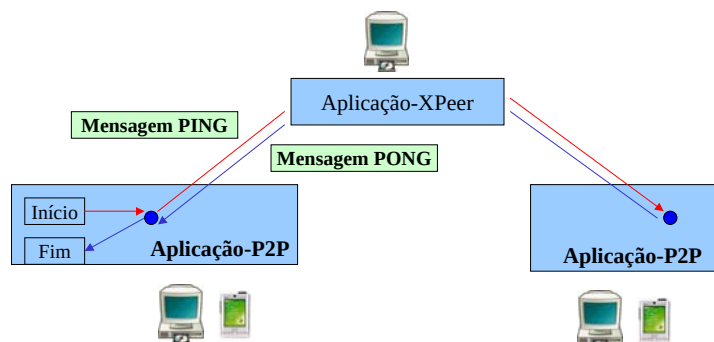


Figura 37 Execução de uma comunicação indireta

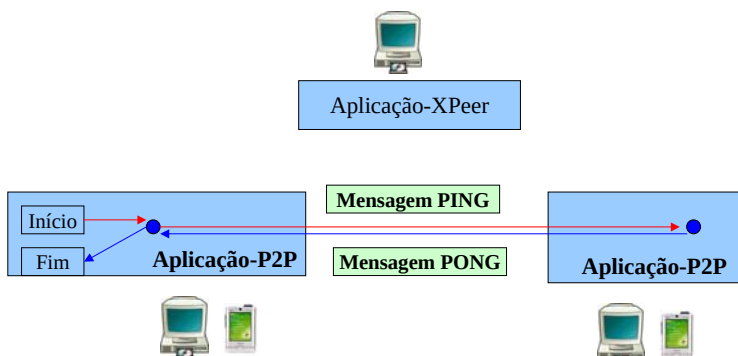


Figura 38 Execução de uma comunicação direta

A sequência da Figura 39 mostra como é realizada a coleta da métrica RTT. Quando se inicia uma medição, a aplicação solicita ao usuário a configuração dos parâmetros de entradas. Após realizar a configuração, a aplicação se conecta a um nó X-Peer e solicita serviços Register para registrar a aplicação e um usuário. Em seguida, o aplicativo solicita o serviço Join à infra-estrutura X-Peer, para informar que o usuário está *online* na aplicação de medição. Após essa configuração do ambiente de medição, é iniciada a medição de acordo com o cenário a ser avaliado. Também é necessário configurar o aplicativo do segundo equipamento que segue a mesma sequência da Figura 39, exceto os últimos passos. Isso porque o segundo equipamento aguarda mensagens PONG e, quando recebe PONG, envia uma mensagem PING.

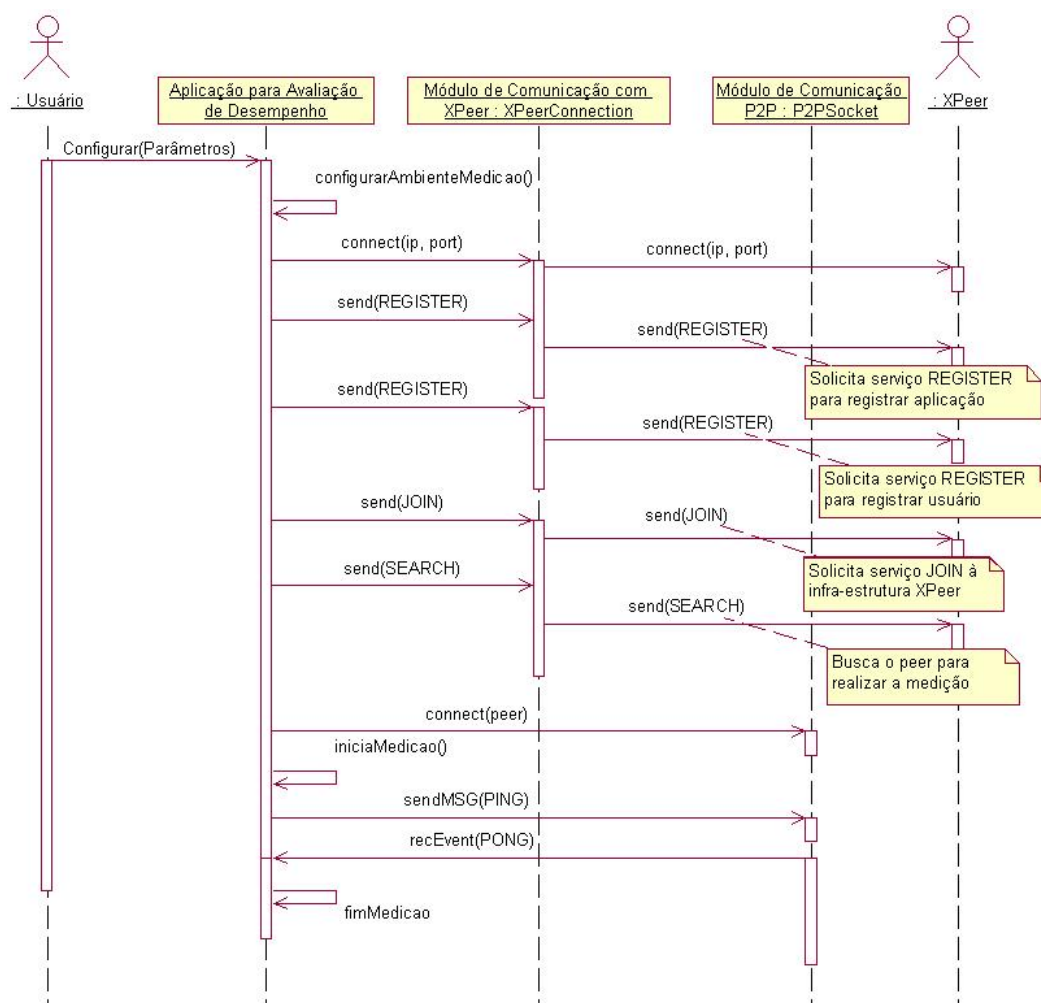


Figura 39 Diagrama de sequência simplificado do aplicativo para coletar RTT

É importante ressaltar que as coletas realizadas consideram o atraso desde a camada física até a camada de aplicação, uma vez que o foco deste trabalho está em apresentar as características viáveis de aplicações P2P-Móveis.

6.5.2 Requisito de coleta da Métrica Vazão

Este é o segundo requisito da aplicação de medição. Este requisito tem o objetivo de realizar a coleta de várias amostras da métrica vazão. Cada amostra é obtida a cada intervalo de tempo definido na interface de configuração da aplicação. A coletado ocorre durante a troca de mensagens PING e PONG, ou seja, a medição da vazão obtém valores para uma aplicação caracterizada por uma sincronização da troca de mensagens.

A seqüência da Figura 40 mostra como é realizada a coleta da métrica Vazão. Esta métrica é capturada por apenas um equipamento, que executa a aplicação configurada para aguardar mensagens PING. Isso, porque o equipamento, configurado para aguardar mensagens PONG, segue o diagrama de seqüência da Figura 39.

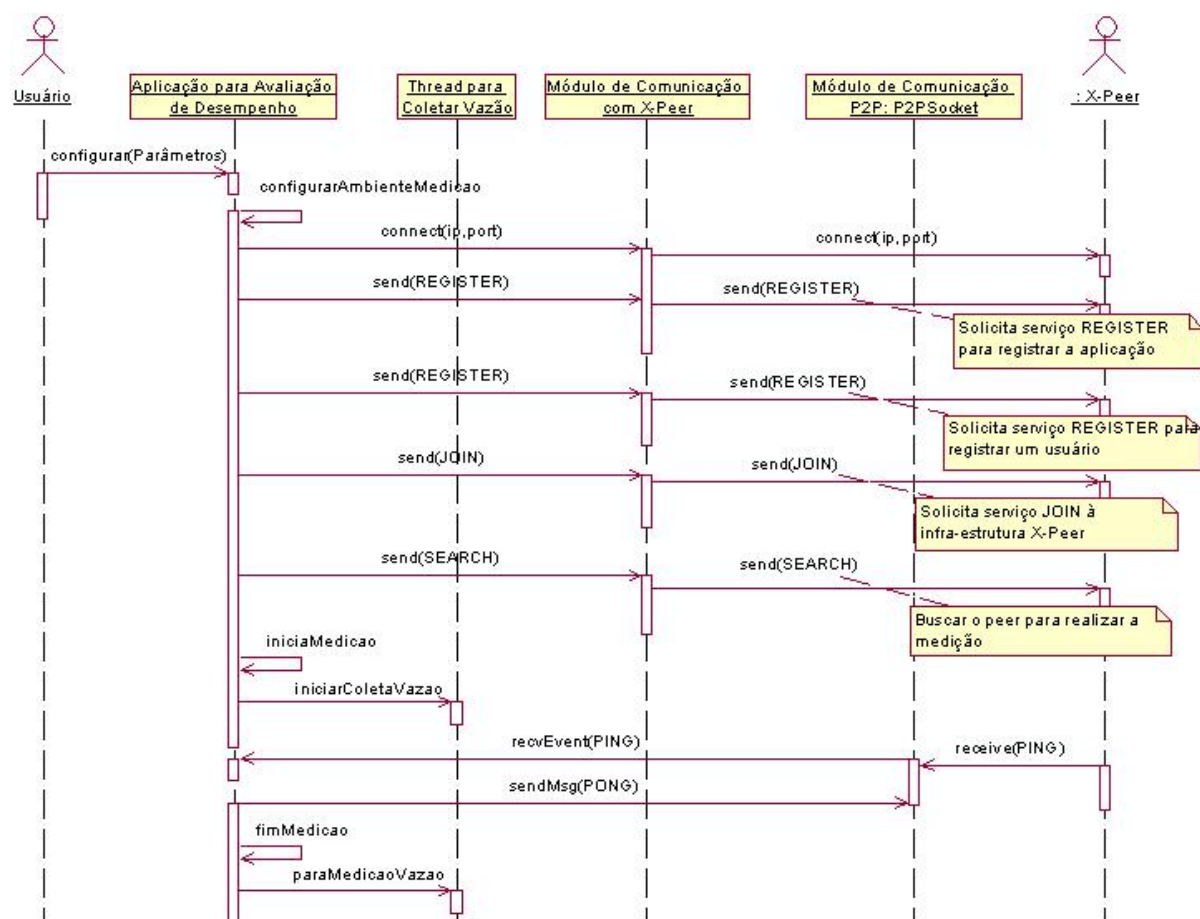


Figura 40 Diagrama de seqüência simplificado do aplicativo para coletar RTT

Quando se inicia uma medição da Vazão, a aplicação solicita ao usuário a configuração dos parâmetros de entrada. Após realizar a configuração, a aplicação se conectar a um nó X-Peer e solicita serviços Register para registrar a aplicação e um usuário. Em seguida, o aplicativo solicita o serviço Join à infra-estrutura X-Peer, para informar que o usuário está *online* na aplicação de medição. Após essa configuração do ambiente de medição, é iniciada a medição de acordo com o cenário a ser avaliado. Durante esta medição são realizadas as ações: responder mensagens PING e coletar a quantidade de *bytes* recebida

pela aplicação em intervalos de tempo. O valor deste parâmetro (intervalo de tempo) é determinado pelo usuário na interface de configurações.

6.6 Cenários e Resultados

Os objetivos desta análise foram citados na seção 6.2 e estão relacionados com a transmissão de dados entre usuários e a arquitetura X-Peer, considerando a participação de *peers* móveis.

Devido ao grande número de combinações possíveis para os parâmetros da seção 6.4, optou-se por definir os fatores e níveis dos parâmetros em cada cenários.

Em todos os cenários, foram realizadas 3 repetições para cada conjunto de 32 medições (envio de 32 mensagens PING), porque foi identificado um elevado desvio padrão para um conjunto de 300 ou mais medições e devido a outras restrições detalhadas na seção de dificuldades. Foram rotulados intervalos de confiança ao nível de 95%.

6.6.1 Cenário 1: C1

As aplicações em geral transmitem mensagens com tamanhos variados, portanto este cenário tem como objetivo responder o segundo questionamento da seção 6.2: saber a variação do tempo de resposta à medida que o tamanho das mensagens cresce, assim como avaliar o tempo de resposta na comunicação direta e indireta entre *peers* móveis.

Foram escolhidos diferentes tamanhos de mensagens, com crescimento exponencialmente o que pode ser visto na Tabela 6. Estes valores foram atribuídos para saber o impacto no tempo de resposta nas comunicações diretas e indiretas, quando o tamanho da mensagem a ser enviado cresce.

Tabela 6 Parâmetros para o cenário 1

Parâmetros	
Fatores	Níveis
Tipo de comunicação	Direta e Indireta
Tamanho da mensagem	32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 <i>bytes</i>
Equipamento	Comunicação entre dois dispositivos móveis
Tipo de protocolo	TCP

A Figura 41 mostra os tempos de respostas e as taxas de mensagens recebidas por um *peer* móvel (vazão) à medida que o tamanho das mensagens cresce exponencialmente, na comunicação direta. Pode-se observar que, a partir do tamanho de mensagem 8192 *bytes*, inicia-se um crescimento do RTT e a queda da vazão. Ainda não existe uma justificativa para isso.

De acordo com [42] cada rede impõe um tamanho máximo a seus pacotes - MTU (*Maximum Transmission Unit*). Dentre as principais causas estão:

1. Hardware (por exemplo, diferentes equipamentos de ponto de acesso sem fio podem determinar diferentes tamanhos de pacotes);
2. Sistema operacional (por exemplo, todos os *buffers* têm 512 bytes);
3. Protocolo (o tamanho máximo para TCP pode ser maior ou menor que para UDP);
4. Compatibilidade de algum padrão (por exemplo, *Ethernet* e 802.11b);
5. Desejo de reduzir de alguma forma as retransmissões provocadas por erro;
6. Desejo de evitar que um pacote ocupe o canal por muito tempo.

De acordo com [43], o tamanho MTU em uma rede 802.11 é de 8191 bytes, por isso, a curva de crescimento do RTT é iniciada quando a mensagem possui tamanho de 8192 bytes, devido ao aumento no número de pacotes a serem transmitidos.

Portanto, provavelmente, a curva de crescimento do RTT é iniciada quando a mensagem possui tamanho de 8192 bytes, devido à necessidade da fragmentação dos quadros em vários pacotes na camada de rede [42], ou seja, devido ao aumento no número de pacotes a serem transmitidos.

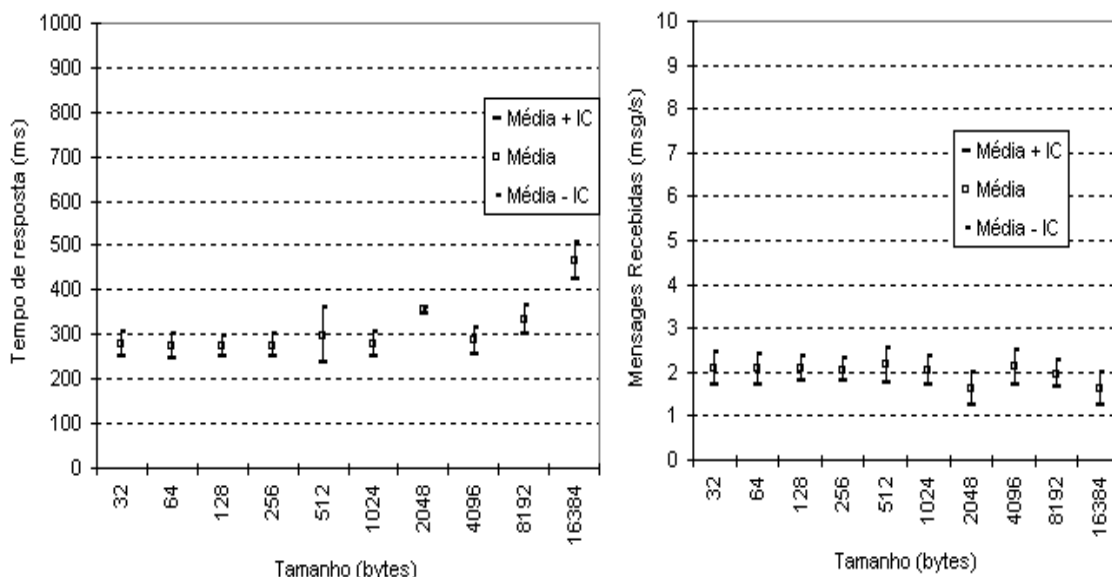


Figura 41 RTT e Vazão da comunicação direta entre peer móveis

A Figura 42 mostra a variação do tempo de resposta e da vazão à medida que a mensagem cresce exponencialmente, na comunicação indireta. Pode-se observar que, a partir do tamanho de mensagem 2048 bytes, é apresentado o início de um crescimento do RTT e a queda da vazão.

Ainda não existe uma justificativa para isso, semelhante a explicação da Figura 41. É necessário realizar mais estudos sobre este resultado, principalmente, para responder o seguinte questionamento: Porque à medida que as mensagens crescem acima de 2048 *bytes*, não ocorrem mudanças significativas na curva de crescimentos?

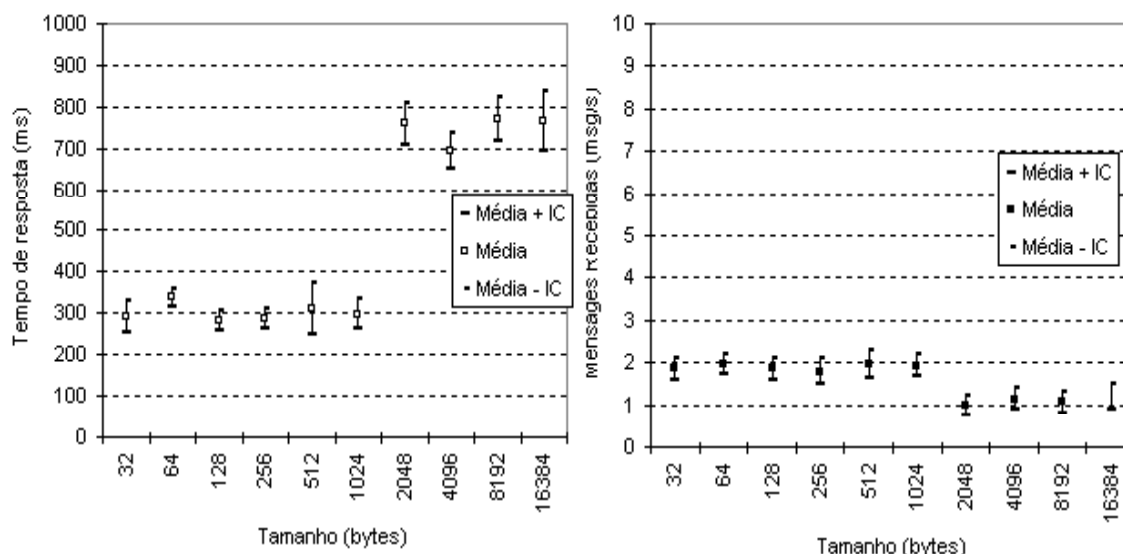


Figura 42 RTT e Vazão da comunicação indireta entre peer

A Figura 43 mostra as médias dos resultados anteriores relacionados com o tempo de resposta e vazão da comunicação direta e indireta. Os tempos de respostas da comunicação direta e indireta são muito próximos quando a mensagem é inferior a 2048 *bytes*, porque , provavelmente, as mensagens com estes tamanhos não precisam ser fragmentadas.

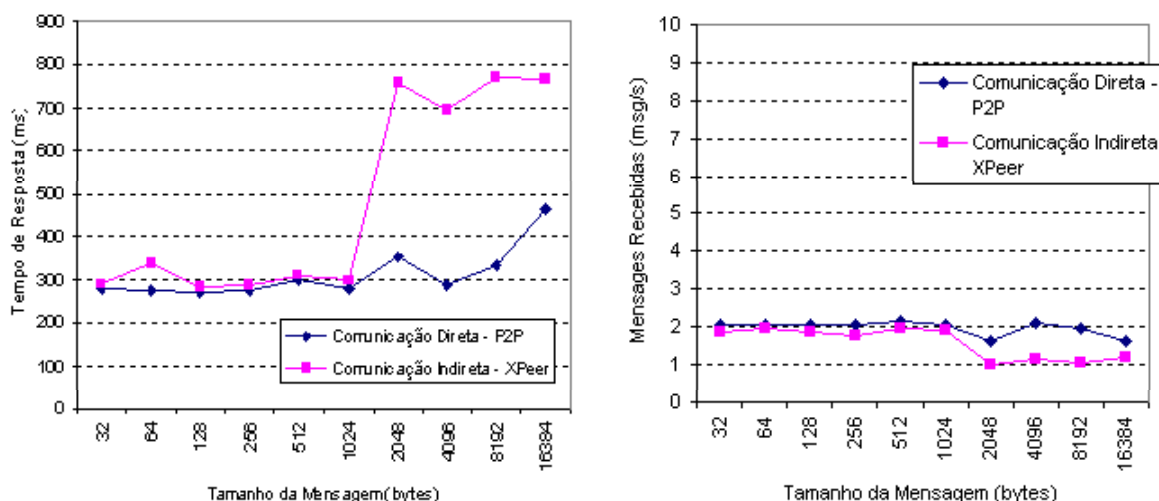


Figura 43 Comportamento do tempo de resposta com o aumento das mensagens

Ainda não existe uma justificativa para o comportamento da curva da Figura 43. De acordo com [43] as redes *Ethernet* e 802.11 possuem tamanhos MTU diferentes e de acordo [42] uma das causas da necessidade de fragmentação devido a compatibilidade dos padrões

Ethernet e 802.11. Portanto, provavelmente, a diferença do início da curva de crescimento ocorre devido aos diferentes tamanhos de MTU para os padrões de rede *Ethernet* e 802.11.

Também é necessário realizar mais estudos sobre o comportamento da curva da **Figura 43**, para saber por que o atraso presente na comunicação indireta é muito elevado. Existem duas prováveis suspeitas que precisam ser melhor analisados:

- Duas fragmentações: entre as redes 802.11 e *Ethernet*, pois os pacotes percorrem duas vezes o caminho *peer* móvel - ponto de acesso sem fio - X-Peer - ponto de acesso sem fio - *peer* móvel;
- Latência na rede X-Peer, mesmo sendo formado por apenas um nó, neste ambiente de avaliação. Ela também cresce à medida que é necessário fragmentar mais pacotes. Os próximos parágrafos explicam a causa deste crescimento.

O crescimento da latência no X-Peer pode ser observado na **Figura 44**.

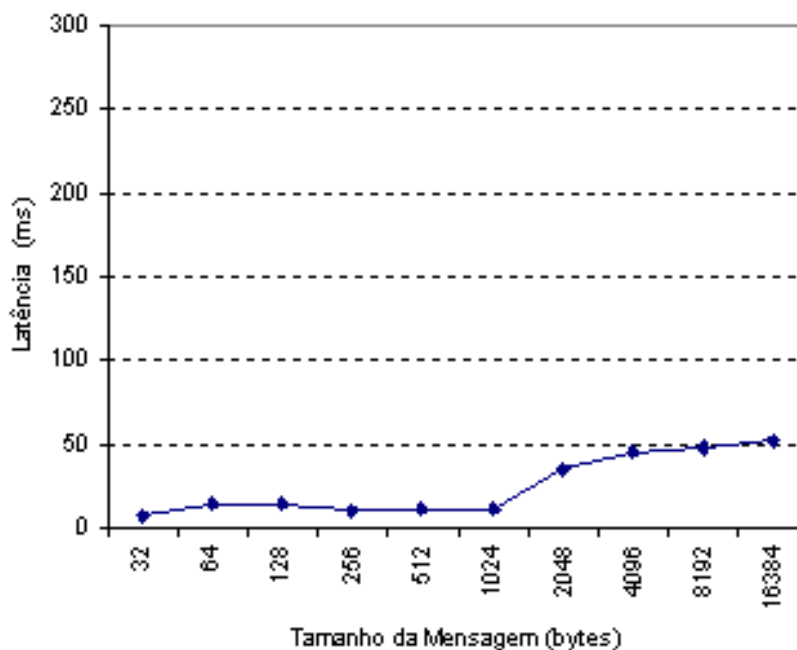


Figura 44 Latência no X-Peer na comunicação *peer* fixo - *peer* fixo

Existe o crescimento da latência no X-Peer à medida que as mensagens crescem, porque, quando uma aplicação deseja solicitar a execução do serviço Deliver, ocorre uma maior quantidade de trocas de mensagens denominadas DHTDeliver utilizando UDP. Isso representa o crescimento no atraso da comunicação indireta. A explicação para o crescimento da latência no X-Peer vista na **Figura 45** é apresentada em seguida, por meio do diagrama de seqüência da troca de mensagens na rede X-Peer. Esse aumento da latência ocorre mesmo quando existe apenas um nó na rede X-Peer.

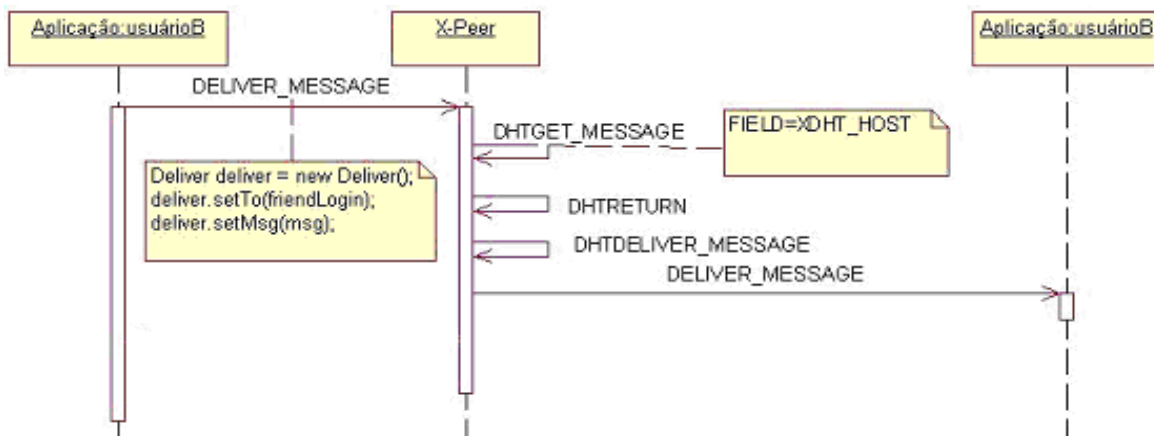


Figura 45 Descrição do diagrama de seqüência da troca de mensagens DHTDeliver

A **Figura 45** apresenta a seqüência de trocas de mensagens na rede X-Peer formada por apenas um nó. A aplicação de medição cria uma mensagem Deliver para solicitar a execução da transmissão de dados em uma comunicação indireta.

O nó X-Peer ao recebe a mensagem Deliver cria uma mensagem DHTGet para saber o host do X-Peer em que a aplicação do usuárioB está conectado. Após obter essa informação, o nó X-Peer envia para ele mesmo uma mensagem DHTDeliver e para cada DHTDeliver enviada, um DHTReturn é recebida. É importante salientar que os nós da rede X-Peer utilizam UDP para transmitir as mensagens DHTGet, DHTDeliver e DHTReturn; e utilizam TCP para transmitir a mensagem Deliver.

Portanto, à medida que cresce o tamanho da mensagem enviada pela aplicação de medição, também cresce o número de pacotes UDP enviados pelo nó X-Peer para ele mesmo e o número de pacote utilizando TCP enviado para as aplicações de medição.

6.6.2 Cenário 2: C2

A arquitetura X-Peer, diferente das outras do capítulo de arquitetura (capítulo 4), permite a comunicação entre um *peer* fixo e um *peer* móvel sem a presença de um *Proxy*. Este cenário tem como objetivo dá continuidade da resposta do segundo questionamento da seção de objetivos, assim como validar a comunicação direta e indireta entre um dispositivo móvel e um *desktop*.

Foram escolhidos diferentes tamanhos de mensagens, com crescimento exponencialmente o que pode ser visto na Tabela 7. Este valores foram atribuídos para saber o impacto no tempo de resposta nas comunicações diretas e indiretas, quando o tamanho da mensagem a ser enviado cresce.

Tabela 7 Parâmetros para o cenário 2

Parâmetros	
Fatores	Níveis
Tipo de comunicação	Direta e Indireta
Tamanho da mensagem	32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 <i>bytes</i>
Equipamento	Comunicação entre <i>desktop</i> e um dispositivo móvel
Tipo de protocolo	TCP

A **Figura 46** mostra os tempos de respostas e as taxas de mensagens recebidas por um *peer* fixo (vazão) à medida que as mensagens crescem exponencialmente, na comunicação direta. Pode-se observar que, a partir do tamanho de mensagem 2048*bytes*, inicia-se um crescimento do RTT. Este crescimento é mais claro quando se observa a queda da vazão em 2048 *bytes*. Ainda não existe uma justificativa para isso, mas possivelmente ocorre devido à fragmentação, semelhante a explicação do cenário 1. Portanto, é necessário realizar novos experimentos para confirmar a hipótese sobre a questão da fragmentação.

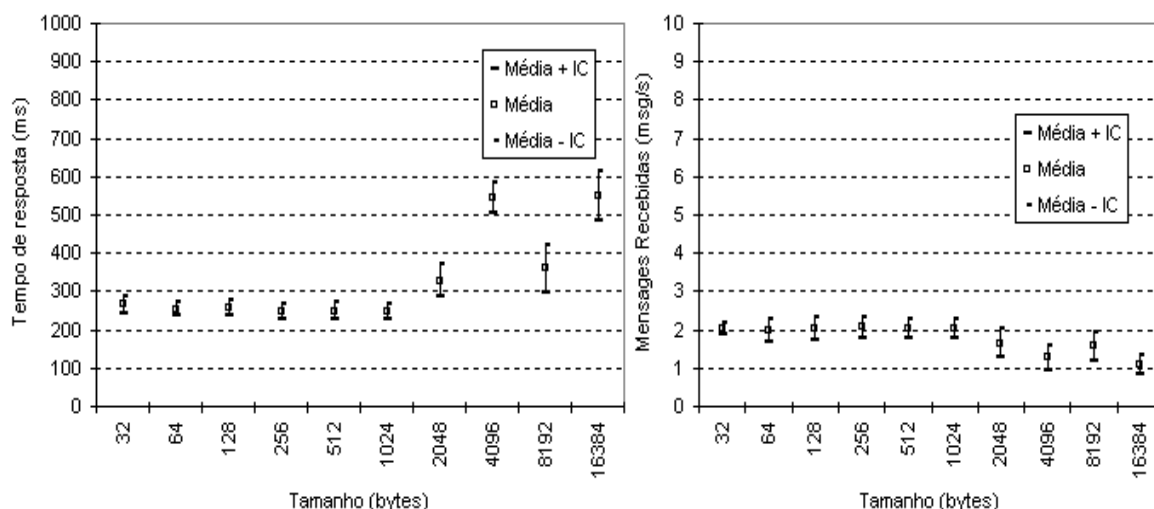


Figura 46 RTT (ms) e vazão (msg/s) da comunicação direta entre um *peer* móvel e um *peer* fixo

A **Figura 47** mostra os tempos de respostas e as taxas de mensagens recebidas por um *peer* fixo (vazão) à medida que as mensagens crescem exponencialmente, na comunicação indireta. Pode-se observar que, a partir do tamanho de mensagem 2048*bytes*, inicia um crescimento do RTT, este crescimento é mais claro quando se observa a queda da vazão em 2048 *bytes*. Ainda não existe uma justificativa para isso, mas possivelmente ocorre devido à fragmentação, semelhante a explicação do cenário 1.

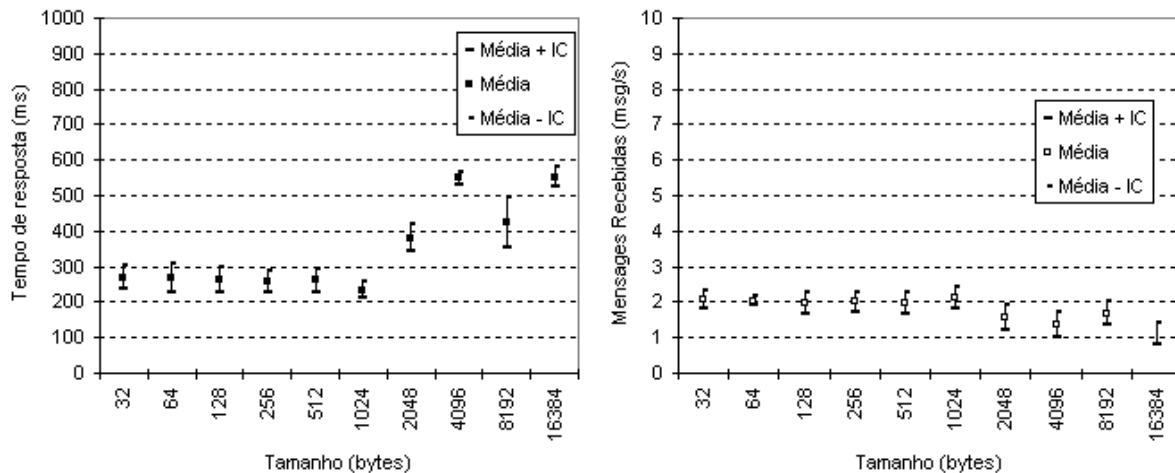


Figura 47 RTT (ms) e vazão (msg/s) da comunicação indireta entre um *peer* móvel e um *peer* fixo

A Figura 48 mostra as médias dos resultados relacionados com o tempo de resposta e vazão da comunicação direta e indireta. O tempo de resposta da comunicação direta é inferior ao RTT da comunicação indireta, existindo uma pequena diferença entre a comunicação direta e indireta.

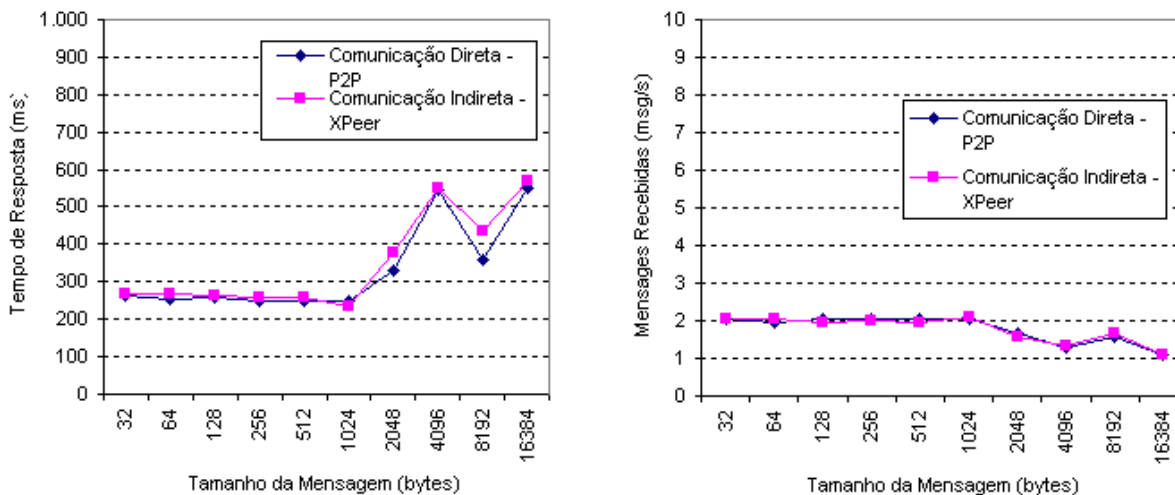


Figura 48 RTT e vazão na comunicação entre um dispositivo móvel e um *desktop*

Os tempos de respostas da comunicação direta e indireta são muito próximos independente do tamanho da mensagem. Ainda não existe uma justificativa para o comportamento da curva da Figura 44. Observado na Figura 44 da seção 6.6.1, pode-se verificar que a latência no X-Peer é muito pequena. Neste caso o caminho *peer* móvel – ponto de acesso sem fio – X-Peer – ponto de acesso sem fio – *peer* móvel é percorrido apenas uma vez, porque o *desktop* está na rede mesma rede do X-Peer.

Os mesmos experimentos foram realizados com o ponto de acesso sem fio *Trendnet* 802.11g modelo *Tew-410 APB Plus* com taxa de transferência de 54 Mbps. Essa mudança de ponto de acesso sem fio representou em um RTT duas vezes maior.

Portanto, utilizando o resultado anterior, os percursos dos pacotes entre os peers e a comparação entre a Figura 44 e a **Figura 43**, pode-se supor que o atraso mais significativo para determinar o RTT na comunicação entre os *peers* é o *overhead* causado pela fragmentação entre as duas redes e o hardware do ponto de acesso sem fio. No entanto, ainda seria necessário realizar mais experimentos.

6.6.3 Cenário 3: C3

O terceiro objetivo desta avaliação é comparar os cenários anteriores C1 e C2 que pode ser observado na Figura 49. É possível observar que os RTTs, no intervalo de tamanho 32 à 1024 *bytes*, são muito próximos. Também se pode ver a escala do tempo de resposta de resposta do menor para o maior:

- 1) O menor RTT é representado pela comunicação direta entre um *peer* fixo e um *peer* móvel;
- 2) O segundo menor tempo de resposta é encontrado na comunicação indireta entre um *peer* fixo e um *peer* móvel utilizando o X-Peer;
- 3) O terceiro menor RTT está na comunicação direta entre *peers* móveis.
- 4) O maior tempo de resposta é encontrado na comunicação indireta entre os *peers* móveis utilizando o X-Peer.

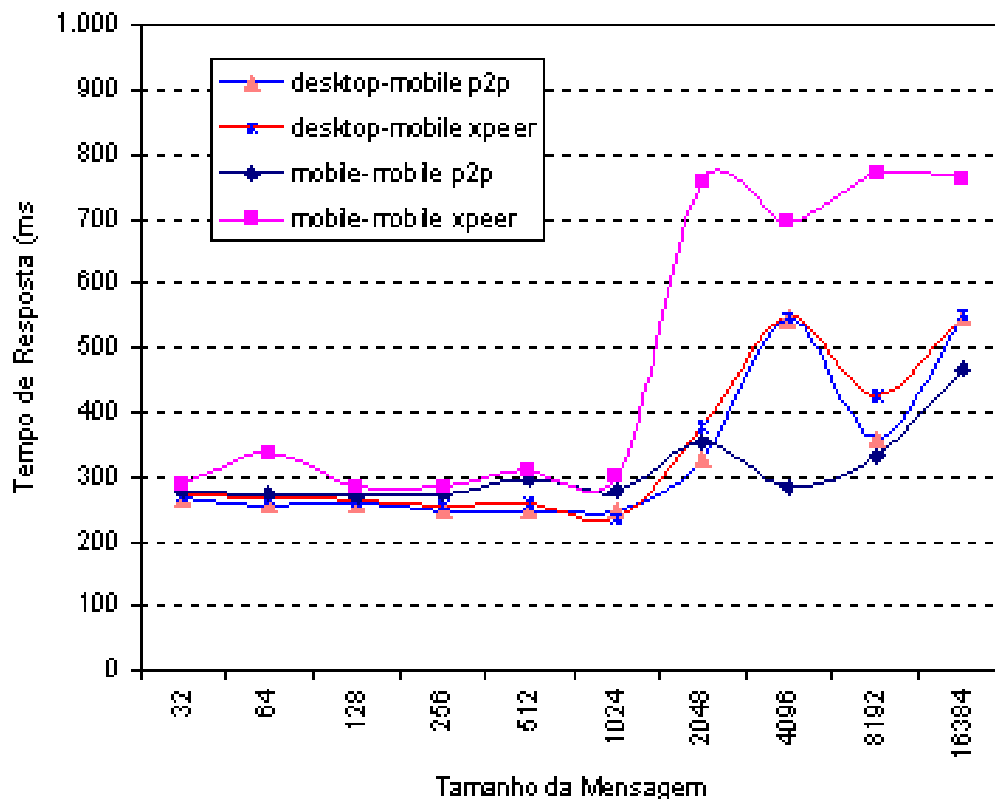


Figura 49 Comparando os cenários 2 e 3

A partir do tamanho de mensagem 1024bytes, existe uma maior variação entre os tempos de resposta. Isso ocorre provavelmente devido a diferentes fragmentações da rede 802.11 e Ethernet entre os *peer* móveis e o ponto de acesso sem fio, uma vez que existe a suspeita de que o RTT é determinado pela comunicação em que *peer* móveis estão presentes. Esta última suposição foi apresentada no cenário C2.

6.6.4 Cenário 4: C4

Este cenário tem o objetivo de responder o questionamento dos itens 3 e 4 da seção 6.2: verificar a viabilidade de aplicações P2P-Móvel que utilizam o novo Módulo P2P da seção 5.1 e comparar o tempo de resposta, quando se utiliza UDP e TCP na comunicação direta entre *peers*.

O ambiente de medição foi configurado de acordo com a Tabela 8, para realizar este experimento.

Tabela 8 Parâmetros para o cenário 3

Parâmetros	
Fatores	Níveis
Tipo de comunicação	Direta
Tamanho da mensagem	32 bytes
Equipamento	Comunicação entre <i>peer</i> móveis
Tipo de protocolo	TCP e UDP

A literatura mostra que existe um maior atraso quando se utiliza o protocolo TCP, devido à sinalização utilizada por esse protocolo para garantir a entrega dos pacotes, visto que TCP é um protocolo orientado a conexão. Portanto era esperado um menor tempo de resposta para as mensagens que utilizaram UDP. No entanto, os resultados deste cenário não se comportaram de maneira esperada. A Figura 50 mostra o tempo de resposta quando se utiliza UDP e TCP por meio do novo Módulo P2P.

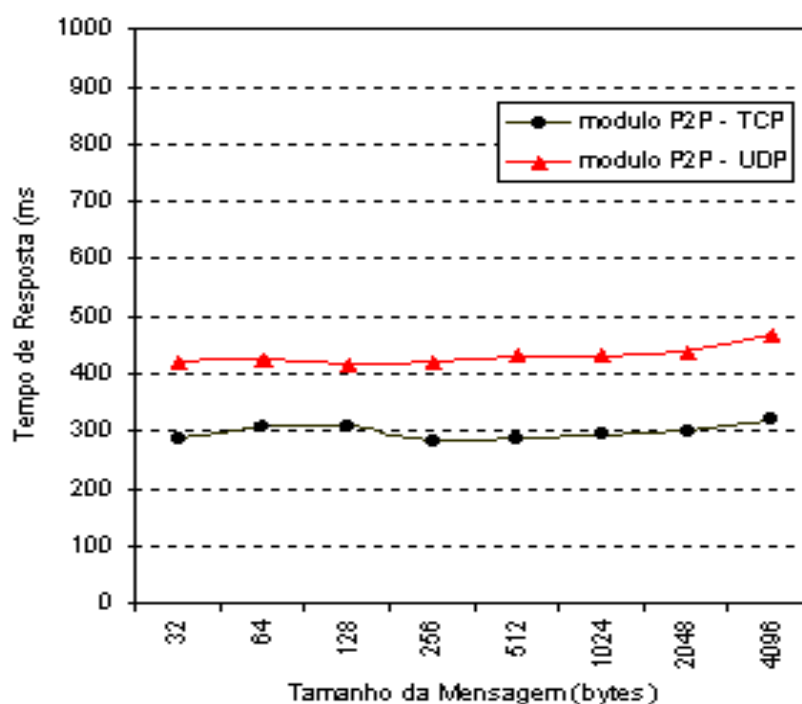


Figura 50 Comparando o tempo de resposta UDP e TCP

Após identificar que o RTT da transmissão com o protocolo UDP é maior que o RTT da transmissão com o protocolo TCP, foram realizados novos experimentos isolando a API J2ME do código do novo Módulo P2P. Desse modo, pretendia-se saber se a diferença do RTT está relacionada com a implementação do novo módulo P2P. Para isso, foi necessário implementar uma aplicação que utiliza apenas *Socket* (Apenas TCP) e *Datagram* (Apenas UDP).

Os resultados presentes, na Figura 51, mostram os tempos de resposta, quando se realiza a comunicação direta entre *peers* móveis utilizando mensagem de tamanho 32 bytes com

TCP e UDP. Pode-se observar que o RTT da transmissão com o protocolo TCP é inferior ao tempo de resposta das mensagens enviadas com protocolo UDP. Portanto, os resultados continuam a contradizer a literatura e se pode concluir que o problema não está relacionado com a implementação do novo Módulo P2P.

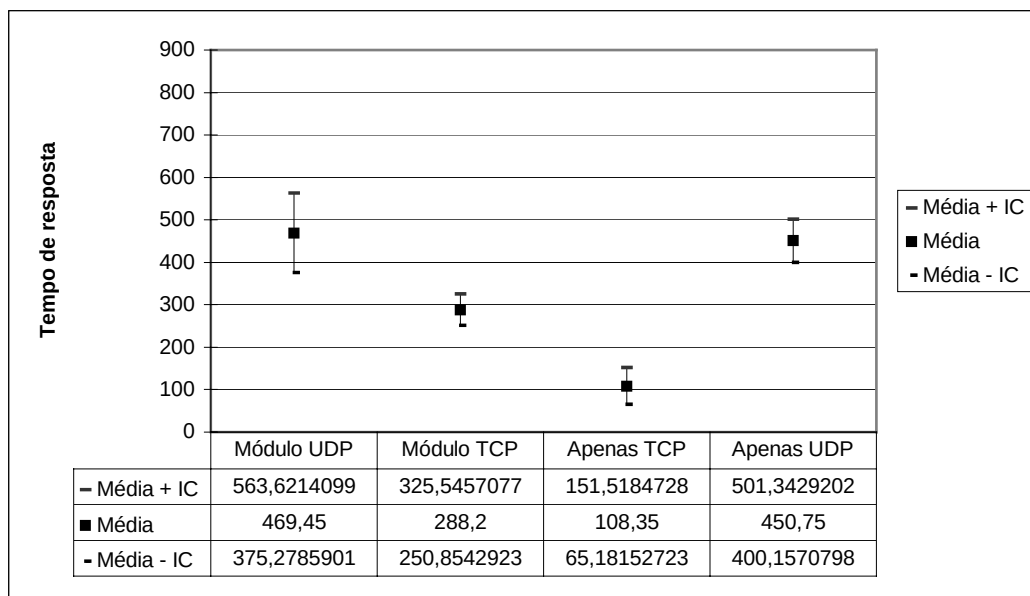


Figura 51 Comparação do RTT usando apenas J2ME e Módulo P2P

Novos experimentos foram realizados para isolar a possibilidade de que o problema esteja relacionado com a máquina virtual dos dispositivos móveis. Para isso, as aplicações foram executadas com emulador de J2ME em dois *desktops*. Durante a medição, estes computadores utilizaram a rede local do GPRT e dessa maneira isolando apenas a questão da máquina virtual. Neste ambiente, o RTT das mensagens que utilizam UDP continuou a ser maior que as mensagens que utilizavam TCP.

Dessa forma, o passo seguinte foi também isolar a rede de comunicação. As aplicações foram executadas em um único computador sem utilizar a rede local. Os resultados dos tempos de respostas apresentaram valores próximos. No entanto, como pode ser visto na Figura 53, os resultados continuam a contradizer a literatura.

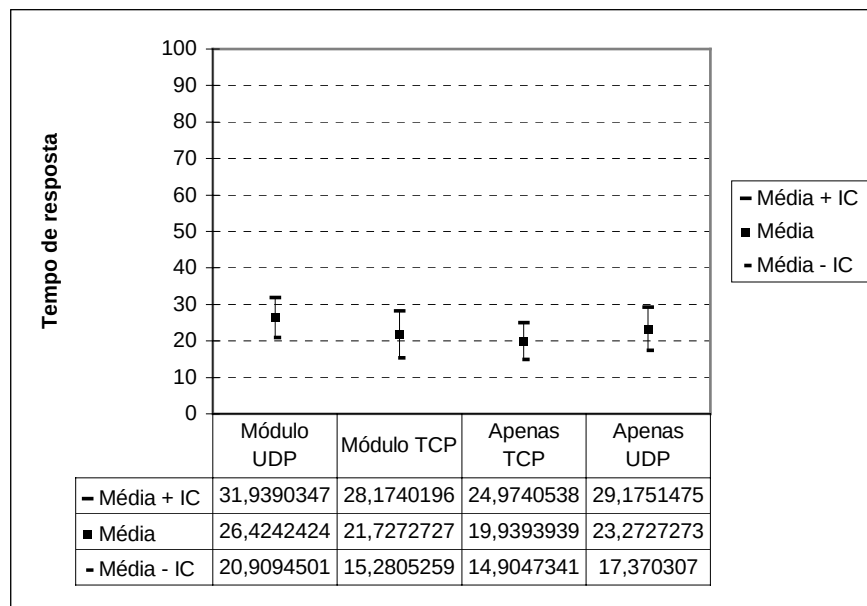


Figura 52 Comparação do RTT usando apenas J2ME e Módulo P2P no emulador

Dessa forma, surgiu a suspeita de que o problema poderia estar na implementação de J2ME em relação ao uso de UDP. Para isso, foi implementada uma aplicação de medição que utilizasse *Socket* e *Datagram* da API J2SE. Esta aplicação transmitiu mensagens de 32 bytes na rede local do GPRT entre dois *desktops*. Verificou-se que a média do RTT de mensagens que utilizam UDP é de 0,26ms e o tempo de resposta das mensagens que utilizaram TCP foi em média de 32 ms. Portanto, a aplicação que coleta RTT e o novo Módulo P2P não estão implementados incorretamente, mas provavelmente o problema está na implementação de J2ME, ou máquina virtual para PalmOS, ou é problema com a pilha TCP do PalmOS.

Devido aos valores não esperados, foi realizada uma pesquisa sobre a ineficiência de J2ME quanto à comunicação com UDP. Nenhum artigo comprovou essa ineficiência. O artigo [40] também identificou resultados não esperados com UDP e TCP, quando avaliava aplicações em C e J2ME. Esse artigo apenas suspeita que possa existir algum problema com a máquina virtual para o dispositivo da Palm.

Foi identificado em alguns *fóruns* que algumas pessoas também se depararam com este problema, mas nenhuma apresentou a causa do problema. Também foi adicionada a identificação deste problema no *fórum* da Sun para desenvolvedores que utilizam J2ME, mas nenhuma resposta foi postada até este momento (10/08/2005).

Portanto, seria um trabalho futuro implementar uma aplicação de medição em C para verificar o tempo de resposta utilizando *Socket* e *Datagram*, assim como instalar a aplicação em J2ME em outro dispositivo para verificar o RTT. Após a realização desses experimentos, seria possível considerar que o atraso é causado pela máquina virtual, implementação de J2ME, ou pela pilha TCP do sistema operacional PalmOS.

6.7 Avaliação dos Resultados

Esta seção tem como objetivo avaliar os resultados obtidos, por meio dos experimentos, e verificar se eles permitem utilizar aplicações como jogos, aplicação que exija mobilidade semelhante ao aplicativo da seção 3.6, troca de mensagens e transmissão de voz.

A referência [41] disponibiliza informações sobre viabilidade de jogos de acordo com atrasos encontrados em redes sem fio. Esse trabalho classifica, em três categorias, os jogos para serem analisados em relação ao RTT no ambiente em rede sem fio:

1. *Action games* (Jogos de ação): Quake II.
2. *Real-time strategy games* (Estratégia em tempo real): Age of Kings.
3. *Turn-based strategy games* (Jogos baseados em estratégia): Panzer General 3D.

Esta referência considera que existem dois atrasos máximos: o atraso para iniciar o jogo (primeira fase) e o atraso durante o jogo (segunda fase). O atraso máximo é de 900ms para a primeira fase do jogo Panzer General 3D e 1400ms para Quake II e Age of Kings.

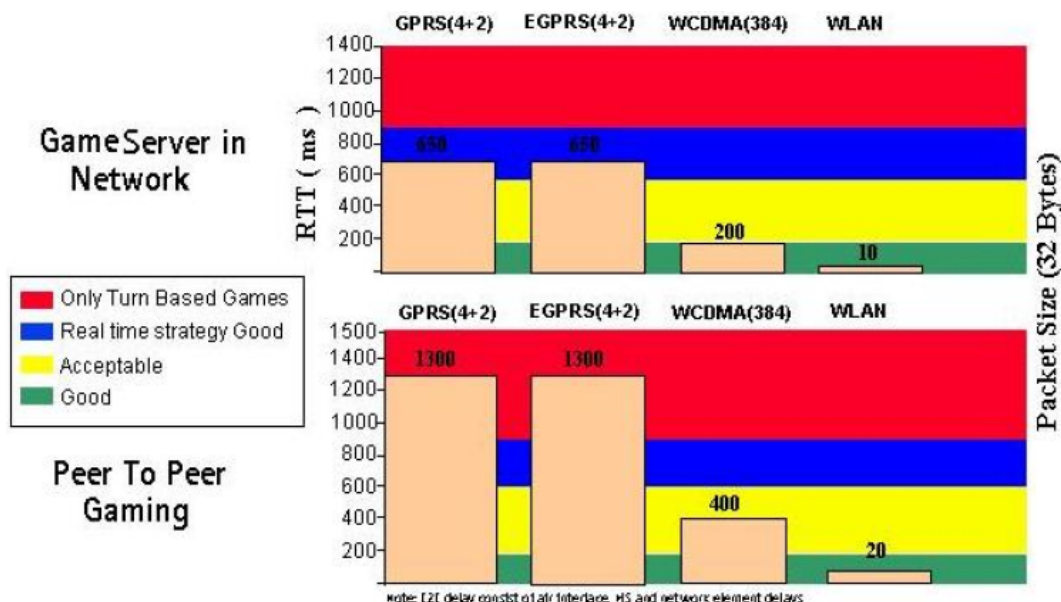
Durante o jogo, o RTT afeta a performance do jogo de acordo com a classificação do jogo. Os atrasos máximos estão na Tabela 9.

Tabela 9 RTT em diferentes jogos

Game	Quality	Max Delay	Max Delay for Experienced Players
Action Games	Good	320 ms	200 ms
	Acceptable	1000 ms	600 ms
Real-Time Strategy Game	Good	900 ms	
	Acceptable	1800 ms	
Turn-Based Games		40 sec*	

Fonte: referência [41]

A Figura 53 apresentar os tempos de respostas classificados como aceitáveis ou não para determinadas classificações de jogos em diferentes redes sem fio. Pode-se perceber que todos os jogos que se enquadram na classificação da referência [41], provavelmente apresentarão boa ou aceitável performance, independente da combinação da comunicação (direta ou indireta) x tipo do *peer* (*peer* móvel ou *peer* fixo), uma vez que o intervalo do atraso dessas combinações está entre 200 e 400.



Fonte: referência [41]

Figura 53 RTT X gênero do jogo X rede sem fio

A partir desta mesma referência [41] pode-se concluir que aplicações de trocas de mensagens (por exemplo, *xatMobile*, disponível pelo GTP2P) também são viáveis. Isso porque todos os jogos listados anteriormente possuem a funcionalidade de troca de mensagens entre os jogadores. Desse modo, uma aplicação que apenas troca mensagens provavelmente será eficiente.

Durante a avaliação de desempenho, foi possível identificar que uma aplicação semelhante a da seção 3.6 não será eficiente. Ela é uma aplicação em que a mobilidade é fundamental, porém a arquitetura X-Peer não fornecer um ambiente favorável a desconexões que ocorrem geralmente durante a mobilidade, ou mesmo falta de energia da bateria do dispositivo móvel.

Também foi identificado que não é viável implementar qualquer aplicação que utilize UDP para transmissão de dados. Isso porque provavelmente existe a ineficiência de UDP da API de J2ME. Logo, seria interessante implementar a mesma aplicação de medição em C ou C++ e em seguida, verificar se o RTT com UDP e TCP irão se comportar de maneira adequada ($T_{RTT}^{UDP} < T_{RTT}^{TCP}$). Após essa verificação, poder-se-ia fornecer o componente P2P em C ou C++. Dessa forma, a aplicação XVoice, disponível pelo GTP2P para executar em computadores locais, precisará aguardar o novo componente P2P, para então está disponível para dispositivos móveis.

6.8 Dificuldades Encontradas

Existiu a intenção de tornar mais automática possível a aplicação desenvolvida para coletar informações do desempenho de aplicações para P2P-Móveis. No entanto, identificou-se a impossibilidade da execução de um grande número de medições seqüencialmente. Primeiramente, porque após 300 medições, o tempo de resposta varia muito. Essa variação no tempo de resposta pode ser visto na Figura 54.

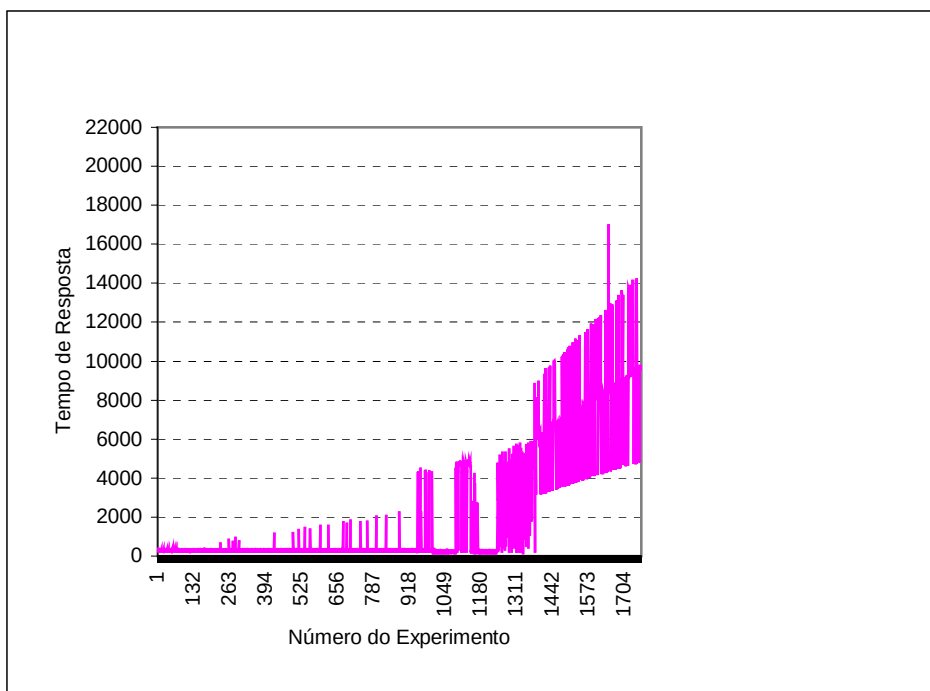


Figura 54 Problema com coletas automatizadas

Além dessa variação, o equipamento é bloqueado e apresenta o erro em “MemoryMgr.c, Line:4613”, após um indeterminado número de coletas. Esse erro obriga o usuário a reiniciar o sistema operacional. A Figura 54 mostra um exemplo em que a aplicação foi configurada para realizar 100000 coletas. Porém, o dispositivo móvel parou de executar na coleta 1850 e o tempo de resposta começou a variar intensamente acima de 500 coletas executadas.

Inicialmente se imaginou que a aplicação estava criando vários objetos dinamicamente o que levaria a falta de memória, visto que são aparelhos restritos em recursos. Após a melhor análise da aplicação desenvolvida, observou-se que os números de objetos criados não variavam com o tempo e, na Web, o mesmo erro foi encontrado em uma outra situação.

Para contornar este problema, foi definido um número fixo de 32 experimentos para cada configuração de parâmetro. Após o fim da medição, o aparelho e todo o ambiente, inclusive o X-Peer, são reiniciados. Realizando os experimentos dessa forma, não foi identificado o erro de performance apresentado anteriormente. No entanto, esta maneira não é totalmente automatizada, e por isso gerando um longo tempo para realização da medição, assim como uma maior monitoração do dispositivo móvel por um ser humano.

Também foi definido que cada conjunto de 32 medições seria repetido três vezes, devido à restrição da automatização e do tempo para realizar a avaliação de desempenho para este trabalho de graduação.

Um problema identificado na arquitetura X-Peer é a questão de permitir um ambiente viável para a mobilidade. O X-Peer restringe que os aplicativos só possam solicitar os serviços se eles estiverem registrados e enviado uma mensagem Join. Quando o aplicativo fecha a conexão com a infra-estrutura X-Peer, essa rede realiza automaticamente o serviço Leave, por isso obrigando o envio de uma nova mensagem Join.

A identificação dos inesperados valores dos tempos de repostas inviabilizou a análise do novo componente P2P e resultou em uma análise mais detalhada sobre o problema identificado.

A existência de um ambiente muito heterogêneo, também foi uma dificuldade, porque exigia uma maior atenção para obter um certo controle do ambiente. Modificações nos resultados poderiam ser causadas por diversos fatores, e muitas vezes imaginava-se que poderia ser a versão da aplicação de medição. Um exemplo em que a heterogeneidade representou uma dificuldade foi a identificação de um elevada RTT, quando comparado com medições anteriores. Inicialmente se supôs que poderia ser um problema de versão ou mesmo da implementação da Aplicação-P2P, mas posteriormente identificou-se que o uso de um diferente ponto de acesso sem fio estava causando o elevado valor do RTT.

6.9 Considerações Finais

Neste trabalho, são fornecidos valores reais da medição de atraso em comunicações diretas e indiretas utilizando uma rede X-Peer, *peers* fixos e móveis. Foi demonstrada uma provável ineficiência de aplicações que utilizam UDP da API J2ME e conseqüentemente uma certa “impossibilidade” de implementação do XVoice utilizando o novo Módulo P2P (seção 5.1). No entanto, foi identificada a viabilidade do uso de jogos que utilizam o protocolo TCP nesta rede P2P-Móvel, assim como a implementação de aplicações de troca de mensagens.

Também foi observado que se deve evitar a troca de grandes mensagens entre *peer* móveis e *peer* fixo, devido ao alto *overhead*, causado, provavelmente, pela a fragmentação. Assim como, deve-se evitar a comunicação indireta entre *peers* móveis e *peers* fixos. Além disso, é interessante priorizar a comunicação direta entre um *peers* móveis e um *peers* fixos, no intervalo entre 32-1024 *bytes*. Portanto, a existência de uma entidade *peer caching* na rede X-Peer também beneficiaria a transferência de mensagens para *peer* móvel e *peer* fixo.

Em relação apenas a questão de avaliações de desempenho, os trabalhos futuros seriam a avaliação do novo componente implementado em outra linguagem; avaliar o X-Peer após a implementação da proposta para permitir um ambiente mais favorável a mobilidade. Assim como dá continuidade as investigações sobre o questionamento da existência de fragmentação entre a rede sem fio e a rede local *Ethernet* e investigar a possibilidade de mudança de tamanho MTU nas rede analisadas.

7 Conclusão

Neste capítulo, serão apresentados as considerações finais, as principais contribuições e os possíveis trabalhos futuros.

7.1 Considerações Finais

Esta monografia teve como objetivo o estudo dos novos desafios que os ambientes sem fio trazem para as aplicações P2P, considerando as diversas restrições de recursos para dispositivos móveis, aplicações e arquiteturas P2P que permitem a participação de *peers* móveis.

Como resultado da pesquisa de P2P-Móvel, inicialmente, foram apresentados as restrições e os pré-requisitos para *peers* móveis. Foi identificado que a presença dos *peers* móveis na rede P2P poderá diminuir a taxa de transferência, uma vez que a menor taxa de transferência entre dois *peers* será utilizada nesta comunicação P2P. Isso motivou a utilização de assimetria na comunicação P2P, por meio de *caching*. Também se observou que as questões econômicas são relevantes em relação à viabilidade de P2P-Móvel, uma vez que os valores cobrados pelas operadoras e a questão de propriedade intelectual resultam em grandes barreiras para as aplicações P2P-Móvel, principalmente na estrutura de rede celular.

Posteriormente foi possível verificar que aplicações atualmente existentes no tradicional P2P estão sendo disponibilizadas para dispositivos móveis, assim como surgem novas aplicações. Um exemplo é o aplicativo com o intuito educativo da seção 3.7.

Para finalizar o levantamento bibliográfico, foram identificadas arquiteturas P2P-Móvel. Foi possível observar que as arquiteturas das redes P2P-Móvel podem ser classificadas de maneira semelhante à classificação das tradicionais arquiteturas P2P (pura e híbrida). O uso de XML é algo comum entre as arquiteturas, o que garante a flexibilidade, no entanto existe a perda de performance. Também é comum a proposta de inclusão de *proxys*, quando a arquitetura da rede P2P-Móvel é classificada como híbrida. A arquitetura da seção 4.3 se destaca por apresentar três maneiras de implementar *proxys* na rede P2P-Móvel. Dentre as arquiteturas identificadas, foram encontradas várias arquiteturas específicas para compartilhamento de arquivos, em que a maioria utiliza a estrutura híbrida. A extensão da arquitetura eDonkey se destaca pelas propostas de *peers caching* e *peers crawler* que quebram o conceito P2P de simetria em benefício da performance. Isso porque essas entidades minimizam o uso da interface aérea a fim de reduzir o atraso na transmissão, a banda e o tráfego de sinalização.

O objetivo deste trabalho de graduação também foi realizar uma avaliação de desempenho para verificar a viabilidade de aplicações P2P-Móvel. A arquitetura X-Peer foi escolhida, sobretudo, devido ao destaque entre as arquiteturas P2P-Móvel pela inexistência de *proxys* e não utilizar XML nos seus protocolos. Isso porque o grupo, que a desenvolveu, priorizou a performance reduzindo o *overhead* na comunicação P2P. Os resultados obtidos pela aplicação implementada neste trabalho foram animadores em relação à viabilidade da construção de aplicações P2P-Móvel utilizando o protocolo TCP.

No entanto, a avaliação verificou que a arquitetura X-Peer dificulta a mobilidade dos dispositivos. Assim como, identificou uma provável ineficiência do uso do protocolo UDP da API J2ME. Este possível problema com J2ME teve como consequentemente a inviabilidade da proposta e implementação da extensão do Módulo P2P, pois o componente foi implementado utilizando a API de J2ME.

O levantamento bibliográfico e avaliação de desempenho resultaram em duas modificações mais relevantes para a arquitetura X-Peer:

- Proposta de modificação da arquitetura X-Peer, para melhor atender aos *peers* móveis, por meio dois diferentes componentes (*Cache-Mobile* e *PeerCache*) e adição de serviços para solucionar a dificuldade de aplicações P2P-Móvel em utilizar frequentemente o serviço Join.
- Proposta de modificação da arquitetura X-Peer para disponibilizar um serviço de LBS. Este serviço permitirá o registro e compartilhamento de dados geográficos, assim como a busca de nós X-Peer, aplicações, *peers*, entre outros recursos por meio da localização geográfica.

7.2 Principais Contribuições

O trabalho realizado resultou em um estudo detalhado sobre P2P-Móvel que pode ser enumerado em: levantamento de restrições e pré-requisitos dos dispositivos móveis, aplicações, arquiteturas, avaliação de desempenho e propostas para melhorar a arquitetura X-Peer. Adicionalmente, é importante destacar algumas contribuições relevantes, dentre as quais se podem citar:

- O levantamento bibliográfico sobre P2P-Móvel resultou na descrição detalhada do estudo sobre P2P-Móvel. Além disso, representou a contextualização para compreensão das propostas de modificação da arquitetura X-Peer para que esta forneça um melhor ambiente para esses novos *peers*.
- A avaliação da arquitetura X-Peer em relação à participação de dispositivos móveis resultou em:
 - a. Verificação da viabilidade de aplicações P2P-Móvel utilizando o protocolo TCP e identificação de um provável problema com a API J2ME. Foi identificado que todas as arquiteturas P2P-Móvel listadas neste documento utilizam J2ME. Portanto esta suspeita de ineficiência de J2ME é extremamente importante para todas essas arquiteturas, assim como para qualquer tipo de aplicação que utilize o protocolo UDP por meio da API J2ME.
 - b. Identificação da importância em evitar a troca de grandes mensagens entre *peer* móvel e *peer* fixo devido ao, provável, *overhead* durante a fragmentação entre as rede sem fio e rede local *Ethernet*.
 - c. Identificação da importância em priorizar a comunicação direta entre um *peer* móvel e um *peer* fixo, no intervalo entre 32-1024 *bytes*. Portanto, a

existência de uma entidade *peer caching* na rede X-Peer também beneficiaria a transferência de mensagens para *peer* móvel e *peer* fixo.

- d. Levantamento do questionamento sobre a existência de fragmentação entre a rede sem fio e a rede local *Ethernet* para realizar uma melhor investigação.
- A proposta e implementação da extensão do Módulo P2P foram mais relevantes para a arquitetura X-Peer em ambientes P2P tradicional. Isso porque a provável ineficiência de J2ME, em relação ao protocolo UDP, acarretou na inviabilidade do uso da extensão deste módulo.
- A proposta de modificação da arquitetura X-Peer foi um dos resultados da avaliação de desempenho, assim como do levantamento bibliográfico. Este último apresentou uma arquitetura P2P-Móvel que propõe a adição de *peers caching* na rede, enquanto a avaliação identificou a dificuldade de aplicações P2P-Móvel em utilizar frequentemente o serviço Join. As sugestões de modificações tiveram como finalidade mostrar uma solução para que a arquitetura X-Peer forneça um melhor serviço aos *peers* móveis.
- A proposta de modificação da arquitetura X-Peer para disponibilizar um serviço de LBS. Isso permitirá o compartilhamento de recursos e recuperação desses baseados em localização.

7.3 Trabalhos Futuros

- Continuar a avaliação de J2ME: finalizar a identificação da suspeita de ineficiência da API J2ME em relação à transmissão de UDP. Para isso seria interessante testar a aplicação de medição em outros dispositivos móveis, a fim de verificar a existência de algum problema quanto a máquina virtual da IMB fornecida para PalmOS. Também se deve desenvolver uma aplicação em C/C++ para verificar se existe algum problema com a pilha TCP do sistema PalmOS.
- Implementar Módulo P2P e sua extensão: Se for confirmado o problema com J2ME, uma sugestão de trabalho futuro seria a re-implementação do Módulo P2P e sua extensão em C ou C++ para dispositivos móveis.
- Finalizar implementação da extensão do componente P2P: a implementação desta extensão não foi finalizada, porque não abstrai cenários em que os *peers* estão sob *firewall*. Na seção 5.1, é proposto o uso da técnica com *relay* para finalizar a implementação da extensão do componente P2P.
- Verificar a viabilidade da aplicação XVoice em *peers* móveis: atualmente, é inviável implementar qualquer aplicação P2P-Móvel utilizando UDP por meio do novo Módulo P2P. Portanto, após identificar o problema ao se usar o protocolo UDP da API J2ME e solucionar este problema, seria interessante avaliar aplicações que utilizam UDP. Essa avaliação teria como objetivo verificar a viabilidade de aplicações semelhantes à aplicação XVoice em *peers* móveis em uma rede X-Peer.

- Avaliar outras tecnologias de rede sem fio: neste trabalho foram realizadas medições em cenários utilizando apenas a tecnologia 802.11 e *Ethernet*. Portanto, poderia-se medir os mesmos cenários modificando a tecnologia de acesso, uma dessas poderia ser GPRS.
- Investigar melhor o RTT da comunicação com TCP: foram levantadas suspeitas em relação aos resultados dos tempos de respostas da comunicação utilizando UDP. Portanto, é necessário finalizar esta investigação, por meio de um maior controle do ambiente de medição. Verificar o número de pacotes trocados durante o experimento. Verificar se o tamanho MTU é realmente diferente nas redes sem fio Wi-Fi e rede *Ethernet*. Assim como, investigar a questão da fragmentação entre as duas redes.
- Implementar a proposta da seção 5.2: implementar a proposta de modificação do X-Peer para atender o requisito de mobilidade e avaliação de desempenho voltada para a questão de mobilidade.
- Implementar a proposta da seção 5.3: implementar a proposta para disponibilizar o compartilhamento de dados relacionados à localização dos *peers*, através do novo serviço XLBS e a adição de um ou vários bancos de dados geográficos na rede X-Peer.

Referências

- [1] STOICA, I.; MORRIS, R.; KARGER, D. R.; KAASHOCK, M; DABEK, F.; BALAKRISHNAN, H. **Chord: A scalable peer-to-peer lookup protocol for internet applications**. In Proceedings of the ACM SIGCOMM, pages 149--160, San Diego, California, Agosto de 2001.
- [2] ROWSTRON, A.; DRUSCHEL, P. **Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems**. In Proc. 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, Outubro de 2001.
- [3] NAPSTER. **Página do Napster**. Disponível em: <<http://www.napster.com>>. Acesso em: julho de 2005.
- [4] KAZAA. **Página do Kazaa**. Disponível em: <<http://www.kazaa.com>>. Acesso em: julho de 2005.
- [5] GNUTELLA. **Página do Gnutella**. Disponível em: <<http://www.gnutella.com>>. Acesso em: julho de 2005.
- [6] DURANTE, G. B. **Redes Peer-to-Peer**. Disponível em: <www.gta.ufrj.br/grad/04_1/p2p/>. Acesso em: julho de 2005.
- [7] **A explosão do Wi-Fi**. Disponível em: <<http://www.cpqd.com.br/site/ContentView.php?cd=1431>>. Acesso em: maio de 2005.
- [8] **Tecnologia: eles chegaram!**. Revista Veja, edição especial nº 46, julho de 2005. p. 60 –61.
- [9] HARJULA, E.; YLIANTTILA, M.; ALA-KURIKKA, J.; RIEKKI, J.; SAUVOLA, J. **Plug-and-Play Application Platform: Towards Mobile Peer-to-Peer**, in the 3rd International conference on Mobile and Ubiquitous multimedia (MUM2004), College Park, Maryland, USA, October 2004. p. 63 – 69.
- [10] DELGADO, D. U. **Implementation and Evaluation of the Service Peer Discovery Protocol**. Maio de 2004. Tese (Doutorado em Ciência da Computação) — Dept. of Microelectronics and Information Technology (IMIT). p. 4-10.
- [11] **Tutorial Wireless**. Disponível em: <<http://www.babooforum.com.br/ideallbb/view.asp?topicID=269602>>. Acesso em: agosto de 2005.
- [12] B. Ford. **Network Address Translation and Peer-to-Peer Applications (NATP2P)**. Disponível em <<http://pdos.csail.mit.edu/~baford/nat/draft-ford-natp2p-00.txt>>. Acesso em: agosto de 2005.
- [13] ANDERSEN, F.; MEER, H.; DEDINSKI, I.; HOßFELD, T.; KAPPLER, C.; MÄDER, A.; OBERENDER, J. O.; TUTSCHKU, K. **An Architecture Concept for Mobile P2P File Sharing Services**.
- [14] MATUSZEWSKI, M. **Business Study of Mobile Peer-to-Peer Content Distribution**. 2005. Disponível em: <http://www.tml.tkk.fi/Opinnot/T-109.551/2005/reports/Mobile_P2P.pdf>. Acesso em agosto de 2005.
- [15] HOßFELD, T.; TUTSCHKU, K.; TRAN-GIA, P.; ANDERSEN, F. **Mapping of File-Sharing onto Mobile Environments: Enhancement by UMTS**. Mobile Peer-to-Peer Computing MP2P, in conjunction with the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom'05), Kauai Island, Hawaii. Maio 2005.

- [16] BISTRÖM, J.; PARTANEN, V. **Mobile P2P – Creating a mobile file-sharing environment**. Relatório técnico, nº 344, novembro de 2004. Disponível em: <<http://www3.informatik.uni-wuerzburg.de/staff/hossfeld>>. Acesso em: agosto de 2005.
- [17] IZZO, C. A. **Skype: Voz sobre IP em Pocket PC**. Disponível em: <<http://www.mundosemfio.com.br/news/000022.shtml>>. Acesso em: agosto de 2005.
- [18] HOWIE, D.; YLIANTTILA, M.; HARJULA, E.; SAUVOLA, J. **State-of-the-art SIP mobile application supernetworking**, proc. Nordic Radio Symposium 2004, Oulu, Finland. Disponível em: <<http://www.mediateam oulu.fi/projects/allip/?lang=en>>. Acesso em: agosto 2005.
- [19] <http://www.wirelessdevnet.com/news/2004/oct/27/news3.html> 24/06/2005
- [20] CALLADO, A.; KAMIENSKI, C.; SADOK, D. H.; SOUTO, E.; JÚNIOR, J.; DOMINGUE, M. A. O.; SILVESTRE, G. **Peer-to-Peer: Computacao colaborativa na Internet**. Minicurso apresentado no Simpósio Brasileiro de Redes de Computadores – SBRC 2004. Disponível em: <www.cin.ufpe.br/~cak/publications/sbrc2004_minicurso_p2p.pdf>. Acesso em: agosto 2005.
- [21] AKKAWI, A.; SCHALLER S.; WELLNITZ O.; WOLF, L. C. **Networked Mobile Gaming for 3G-Networks**, ICEC 2004: Third International Conference, Eindhoven, The Netherlands, September 1-3, 2004. p. 457 – 467.
- [22] KUHNEN, A. **Protótipo de uma aplicação LBS utilizando GPS conectado em celular para consultar dados georeferenciados**. 2003. p. 11 – 23. Monografia (Trabalho de conclusão de curso em ciência da computação) - Universidade Regional de Blumenau.
- [23] **Vivo lança serviços de localização no celular**. Setembro de 2004. Disponível em: <http://www.vivo.com.br/portal/press_releases_-_vivo_lanca_servicos_de_localizacao_no_celular.php>. Acesso em: agosto 2005.
- [24] WIBERG, M. **FolkMusic: A Mobile Peer-to-Peer Entertainment System**, Proceedings of the 37th Hawaii International Conference on System Sciences, USA: Big Island, 2004. p. 90290.2.
- [25] STRIBLING, J.; RHEA, S.; JOSEPH, A.; KUBIATOWICZ, J. **Tapestry: A Resilient Global-scale Overlay for Service Deployment**, IEEE Journal on Selected Areas in Communications, Janeiro 2004.
- [26] RATNASAMY, S.; FRANCIS, P.; HANDLEY, M.; KARP, R.; SHENKER, S. **A Scalable Content-Addressable Network**. ACM SIGCOMM 2001, Agosto de 2001.
- [27] JÚNIOR, J. R.; FIDALGO, J.; DANTAS, R.; OLIVEIRA, L. KAMIENSKI C.; SADOK, D. H. **X-Peer: Um Middleware para Aplicações Peer-to-Peer**. Workshop de Peer-to-Peer (WP2P). SBRC 2005 - Simpósio Brasileiro de Redes de Computadores. 2005.
- [28] KURHINEN, J.; VAPA, M.; WEBER, M.; KOTILAINEN, N.; VUORI, J. **Short Range Wireless P2P for Co-operative Learning**, 3rd International Conference on Emerging Technologies and Applications (ICETA 2004), Kosice, Slovakia, 2004.
- [29] KOTILAINEN, N.; WEBER, M.; VAPA, M.; VUORI, J. **Mobile Cheddar - A Peer-to-Peer Middleware for Mobile Devices**, Workshops Proceedings of the Third IEEE Conference on Pervasive Computing and Communications (Percom 2005), pp. 86-90, Kauai Island, Hawaii, USA, 2005.

- [30] KORTUEM, G. **Proem: A Middleware Platform for Mobile Peer-to-Peer Computing**, ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, no. 4. p. 62–64, 2002.
- [31] DING, G.; BHARGAVA, B. **Peer-to-peer File-sharing over Mobile Ad hoc Networks**, in proc. 2nd IEEE Conf. on Pervasive Computing and Communications Workshops. Orlando, Florida, 2004.
- [32] KORTUEM, G.; SCHNEIDER, J.; PREUITT, D.; THOMPSON, T. G. C.; FICKAS, S.; SEGALL, Z. **When peer-to-peer comes Face-to-face: collaborative peer-to-peer computing in mobile ad hoc Networks**. IEEE Computer Society 2001. 1st International Conference on Peer-to-Peer Computing (P2P 2001), Linköping, Sweden, 2001. p. 27-29.
- [33] SIEBES, R. **Peer to Peer solutions in the Semantic Web context: an overview**
- [34] AURORA, A.; HAYWOOD, C.; PABLA, K. S. **JXTA for J2ME – Extending the Reach of Wireless With JXTA Technology**. 2002. Sun Microsystems. Disponível em: < <http://whitepapers.zdnet.co.uk/0,39025945,60062410p-39000518q,00.htm>>. Acesso em : agosto de 2005. p. 1 – 5.
- [35] JXME. **Página do projeto JXME**. Disponível em: <<http://jxme.jxta.org/>>. Acesso em: agosto 2005.
- [36] Kato, T.; ISHIKAWA, N.; SUMINO, H.; HJELM, J.; YU, Y.; MURAKAMI, S. **A Platform and Applications for Mobile Peer-to-Peer Communications**. Disponível em: <http://www.research.att.com/~rjana/Takeshi_Kato.pdf>. Acesso em: agosto de 2005.
- [37] MEIRELES, R. R. **Tutoriais Telefonia Celular**. Disponível em: <http://www.teleco.com.br/tutoriais/tutorialmwm/pagina_4.asp>. Acesso em: agosto de 2005.
- [38] **Peer-to-Peer Communication Across Network Address Translators**. Disponível em: <<http://www.brynosaurus.com/pub/net/p2pnat/>>. Acesso em: agosto de 2005.
- [39] HOßFELD, T.; MÄDER, A.; TUTSCHKU, K.; TRAN-GIA, P.; ANDERSEN, T.; MEER, H.; DEDINSKI, I. **Comparison of Crawling Strategies for an Optimized Mobile P2P Architecture**. Será publicado na 19th International Teletraffic Congress (ITC19), Beijing, China, September 2005. Disponível em: <<http://www3.informatik.uni-wuerzburg.de/staff/hossfeld/>> Acesso em: agosto de 2005.
- [40] Forum Nokia. **Multiplayer Game Performance over Cellular Networks**
- [41] BANSAL, V.; DALTON, A. **A Performance Analysis of Web Services on Wireless PDAs**
- [42] TUNEMBAUM, A. S. **Redes de computadores**. Tradução 3º edição, editora campus. p. 463 – 467.
- [43] Theoretical Maximum Throughput of IEEE 802.11 and its Applications
- [44] CÂMARA, J. M. O. **Redes sem fio metropolitanas baseadas no padrão 802.16: um estudo de caso para Belém**. 2005. p.12 – 49. Dissertação para obtenção do grau de mestre em ciências em engenharia de sistemas e computação - Universidade Federal do Rio de Janeiro.
- [45] JÚNIOR, P. D. M. **Modelagem e análise de um protocolo de acesso alternativo para o padrão IEEE 802.16 de redes metropolitanas sem fio**. Abril de 2005. p.12 – 49. Monografia (Trabalho de conclusão de curso em ciência da computação) - Universidade Federal do Pará.

- [46] KOTILANINEN, N. **Mobile Chedar – A Peer-to-Peer Middleware for Mobile Devices.** Março de 2005. Apresentação para o workshop internacional em MP2P'05 (Mobile Peer-to-Peer Computing).
- [47] BILLO, E. A. **Uma pilha de protocolos Bluetooth adaptável à aplicação.** Fevereiro de 2003. p. 13 – 14. Monografia (Trabalho de conclusão de curso em ciência da computação) - Universidade Federal de Santa Catarina.

Apêndice A – Glossário

802.11b	Padrão de conexão <i>wireless</i> , nomeado comercialmente como Wi-Fi, que utiliza frequência de 2.4 Ghz (aproximadamente) tendo uma velocidade de 11 Mbps, e uma Cobertura Nominal de 100m.
3GPP	<i>Third Generation Partneship Project.</i>
Ad-hoc	Tipo de configuração da camada de rede para realizar comunicação entre as estações de trabalho diretamente, sem a necessidade de um ponto de acesso (AP) e de uma rede física para conectar as estações.
API	<i>Application Programming Interface.</i>
Aplicação P2P-Móvel	Aplicação que utiliza o conceito P2P (realiza, em algum momento, a comunicação direta entre <i>peers</i>) e é executada em um dispositivo móvel.
Bluetooth	Conexão via rádio frequência com alcance de 10m, utilizados para conexões em uma PAN, tal como interconectar PDAs, celulares, etc.
CHEDAR	<i>CHEap Distributed ARchitecture</i>
CLDC	<i>Connected Limited Device Configuration.</i>
Downlink	Termo técnico usado para definir a transmissão de dados na sequência rede/operadora ou provedor de serviço/Internet ao usuário.
DRM	<i>Digital Rights Management</i>
DTD	<i>Document Type Declaration.</i>
EDGE	<i>Enhanced Data rates for Global Evolution</i>
Firewall	<i>Software</i> para gerenciamento de entrada e saída de informações pela rede.
GGSN	<i>General Packet Radio Service.</i>
GPRS	<i>General Packet Radio Service.</i>
Hot Spots	Pontos de acesso público, que utilizam um <i>Acces Point</i> para fornecer a distribuição de sinal.
HTTP	<i>HyperText Transfer Protocol</i>

Infra-estrutura	Sistema de Conexão Wireless baseado em acesso de vários clientes a um ponto de acesso, que gerencia as informações da rede com toda a política de diretrizes e segurança Cabível.
Infrared	Conexão que utiliza raios infravermelhos para transmissão de dados. Apesar de barata, é uma conexão bem lenta. Encontrado comumente em controles remotos, PDAs ,etc.
IP	<i>Internet Protocol.</i>
ISDN	<i>Integrated Services Digital Network</i>
J2ME	<i>Java 2 Micro Edition.</i>
J2SE	<i>Java 2 Platform, Standard Edition</i>
JXTA	Plataforma para desenvolvimento de aplicações P2P.
JXME	Extensão de JXTA que permite o desenvolvimento de aplicações P2P para dispositivos móveis.
LAN	<i>Local Area NetWork.</i>
MAC Address	Endereço Físico de um componente de rede (como <i>Access Point</i> e Placa de rede) que é único e imutável.
MANET	<i>mobile ad-hoc networks</i>
MIDP	<i>Mobile Information Device Profile.</i>
MMS	<i>Multimedia Messaging System</i>
MS	<i>Mobile Station</i> , ou estação móvel.
MSISDN	<i>Mobile Station ISDN</i> é o número que permite realizar um conjunto de chamadas para um assinante correspondente. Um assinante pode ter vários MSISDNs, correspondentes a diferentes serviços.
MTU	<i>Maximum Transmission Units</i>
NAT	<i>Network Address Translation.</i>
PAN	<i>Personal Area Network.</i> Prove acesso aos aparelhos proximos ao utilizador como celulares, PDAs, notebooks entre outros.
PDA	<i>Personal Digital Assistant.</i> Termo que designa pequenos aparelhos de mão, ou <i>palm tops</i> , com funcionalidade de computador. Alguns lançamentos incorporam celulares - e estes a funcionalidade de um PDA.

Piconet	Micro-redes de dispositivos. É um termo utilizado na tecnologia <i>Bluetooth</i> , onde os dispositivos que estão próximos uns dos outros automaticamente estabelecem contato entre si, formando pequenas redes.
RDF Ponto de Acesso	<i>Resource Description Framework</i> . Conhecido no inglês por <i>Acces Point</i> , sendo responsável por distribuir uma banda de conexão <i>wireless</i> em um ambiente.
QoS	<i>Quality of Service</i>
Rede P2P-Móvel	Rede P2P que permitem a existência e participação de <i>peers</i> móveis.
SuperWaba	Plataforma para desenvolvimento de software para aplicações cujo alvo são os PDAs.
SIP	<i>Session Initiation Protocol</i>
SMS	<i>Short Messaging Service</i> .
TCP	<i>Transmission Control Protocol</i>
UMTS	<i>Universal Mobile Telecommunications System</i> .
Uplink	Termo técnico para a transmissão de dados no sentido do usuário para a rede ou provedor de serviço da Internet. Também designado por “canal de retorno”.
UTRAN	<i>Terrestrial Radio Access Network</i> .
UWB	<i>Ultra Wide Band</i> . Esta é uma tecnologia de transmissão de dados sem fio que ao invés de operarem numa frequência fixa, os transmissores UWB utilizam um número quase infinito de frequências entre 0 e 60 GHz, sem permanecer em uma única frequência por mais do que algumas frações de segundo. Apenas as duas partes envolvidas conhecem o padrão de frequências utilizado, o que ajuda a manter a segurança dos dados. Tem a capacidade de transmitir dados até 500Kbps, mas com um alcance de apenas 10 metros.
WAN	<i>Wide Area Network</i> . Rede Para utilização em prédios corporativos e interligações não próximas fisicamente, mas não tão distantes como a WMAN.
WAP	<i>Wireless Applications Protocol</i>

Wi-Fi	Conexão sem fio, também conhecido como 802.11, que trabalha em frequências de rádios para transportar dados. Velocidade, cobertura nominal e frequência variam de acordo com o padrão.
WiMax	<i>Worldwide Interoperability for Microwave Access</i> . É o padrão 802.16 com alcance de até 40Km, possibilitando taxas de transferência de até 75Mbps.
WLAN	<i>Wireless Local Area Network</i> . Rede interna para uso, na maioria das vezes, doméstica. Conhecida também como Rede local.
WMAN	<i>Wireless Metropolitan Area Network</i> . Rede externa de uso público em acesso as grandes metrópoles e centros urbanos.
XHTML	<i>Extensible HyperText Markup Language</i> .
XML	<i>Extensible Markup Language</i>
XPath	<i>XML Path Language</i>
ZigBee	Tem como proposta tornar-se padrão <i>wireless</i> de baixo consumo e curto alcance para monitoração e automação de aplicações industriais, comerciais ou urbanas, sua taxa de transferência é de 200Kbps e tem o alcance de 75m.