



Programa Interdisciplinar de Pós-Graduação em
Computação Aplicada
Mestrado Acadêmico

Érico Maurício Santos Rocha

LP2P: Uma Abordagem Peer-to-Peer para o Compartilhamento de
Arquivos em Redes Locais

São Leopoldo, 2011

ÉRICO SANTOS ROCHA

**LP2P: Uma Abordagem Peer-to-Peer para o
Compartilhamento de Arquivos em Redes Locais**

Dissertação submetida à avaliação
como requisito parcial para a obten-
ção do grau de Mestre em Computação
Aplicada

Orientador: Prof. Dr. Rafael Bohrer
Ávila

SÃO LEOPOLDO
2011

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

R672l Rocha, Érico Santos

LP2P: Uma Abordagem Peer-to-Peer para o Compartilhamento de Arquivos em Redes Locais / por Érico Santos Rocha. — 2011.

103 f.: il. ; 30cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos - UNISINOS, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada - PIPCA, São Leopoldo, RS, 2011.

"Orientação: Prof. Dr. Rafael Bohrer Ávila, Ciências Exatas e Tecnológicas".

1. Peer-to-Peer. 2. Sistemas Distribuídos. 3. Compartilhamento de arquivos. 4. Redes Locais. I. Título.

CDU 004

Catálogo na publicação: Vanessa Borges Nunes - CRB 10/1556

ÉRICO MAURÍCIO SANTOS ROCHA

Dissertação apresentada como requisito parcial para a obtenção título de mestre, pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos.

Aprovado em 30 de Março de 2011

BANCA EXAMINADORA

Orientador: Prof. Dr. Rafael Bohrer Ávila – UNISINOS

(Examinador interno) Prof. Dr. Rodrigo da Rosa Rigui – UNISINOS

(Examinador externo) Prof. Dr. Philippe Oliver Alexandre Navaux – UFRGS

*Dedico este trabalho
a Flávia Almeida*

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Rafael Bohrer Ávila, por sua orientação, por sua exigência pela perfeição e dedicação, que contribuíram para a finalização deste estudo.

Aos amigos e colegas de mestrado, Janaína Lemos, Maurício Barreto, Jean Schmith e Fernando Artmann pelo companheirismo e colaboração prestados durante todo o período do curso.

A minha esposa, Flávia Almeida, por entender que o período de pouca atenção e tempo de minha parte era passageiro.

A todos os que não estão nesta lista e que de alguma forma contribuíram para que eu chegasse até aqui.

“The best way to predict the future is to create it”

- Peter Drucker

RESUMO

Nos últimos anos, o crescimento da Internet e o surgimento de novos serviços em rede contribuíram para o desenvolvimento de várias arquiteturas destinadas à computação distribuída. Neste contexto, a tecnologia P2P tem fomentado o interesse da comunidade científica em decorrência de suas características e funcionalidades inovadoras. Um dos fatores responsáveis por esta popularidade é proveniente das aplicações destinadas ao compartilhamento de arquivos. Este serviço também ocorre de forma expressiva em redes de menor abrangência (ex. LANs), onde recursos presentes nas estações de trabalho são muitas vezes desperdiçados enquanto as corporações investem maciçamente em infraestrutura específica para prover este serviço. Uma possível alternativa frente a este cenário, está na adaptação de alguns conceitos do paradigma P2P à infraestrutura de rede local. No entanto, há poucos esforços no sentido de explorar o potencial da comunicação peer-to-peer aliada aos padrões e vantagens disponíveis nas LANs. Em virtude de sua natureza distribuída, as abordagens P2P convencionais adotam este sistema principalmente em redes de longa distância (MANs e WANs) para formação da estrutura *overlay*. Contudo, estes aspectos não inviabilizam a utilização de algumas técnicas e características da tecnologia P2P em redes de menor abrangência. Este conjunto de fatores motivaram a proposta do LP2P—Local Peer-to-Peer-Protocol—, uma plataforma de comunicação distribuída e descentralizada que aplica a tecnologia P2P no compartilhamento de arquivos em redes locais.

Palavras-chave: Peer-to-Peer, Sistemas Distribuídos, Compartilhamento de arquivos, Redes Locais.

TITLE: “LP2P: Uma Abordagem Peer-to-Peer para o Compartilhamento de Arquivos em Redes Locais”

ABSTRACT

In recent years, the growth of the Internet and the emergence of new network services contributed to the development of various architectures for distributed computing. In this context, P2P technology has fostered the interest of the scientific community due to its features and innovative characteristics. One of the factors responsible for this popularity is coming from consumer applications and file sharing. This service also occurs in a significant way into networks of smaller scope (e.g. LANs), where features present in the workstations are often wasted while corporations invest heavily in infrastructure to provide this particular service. A possible alternative in the face of this scenario is the adaptation of some concepts of P2P paradigm to the local network infrastructure. However, there are few efforts to explore the potential of communication peer-to-peer coupled with the patterns and advantages available in LANs. Due to its distributed nature, conventional P2P approaches adopted this system mainly in long distance (MANs and WANs) to overlay structure formation. However, these aspects do not make use of some techniques and features of P2P networks smaller. These factors motivated the proposal of LP2P—Local Peer-to-Peer-Protocol—a distributed communications platform that applies technology in P2P file sharing on local networks

Keywords: Peer-to-Peer, Distributed Systems, File sharing, Local area network.

LISTA DE FIGURAS

Figura 1.1	Topologia heterogênea em Redes P2P	18
Figura 2.1	Classificação de sistemas computacionais	22
Figura 2.2	Níveis da arquitetura P2P	23
Figura 2.3	Descentralização de sistemas P2P	28
Figura 2.4	Rede <i>overlay</i>	31
Figura 2.5	Mapeamento de peers e recursos para estrutura <i>overlay</i>	32
Figura 2.6	Tipos de topologias peer-to-peer	34
Figura 2.7	Roteamento Centralizado	35
Figura 2.8	Roteamento por Inundação	36
Figura 2.9	Funcionamento do <i>Random Walk</i>	37
Figura 2.10	Algoritmo DHT	38
Figura 2.11	Tipos de Tráfego	39
Figura 4.1	Operação básica do LP2P	55
Figura 4.2	Arquitetura LP2P	58
Figura 4.3	Estrutura LP2P-BD	60
Figura 4.4	Cliente LP2P	61
Figura 4.5	Formato geral do pacote LP2P	62
Figura 4.6	Comando LIST	63
Figura 4.7	Comando SENDL	64
Figura 4.8	Comandos GET e SENDF	65
Figura 4.9	Comando ADD	66
Figura 4.10	Comando DEL	66
Figura 4.11	Mensagens AnL / RaL / ReqC / EstC	67
Figura 4.12	Formato padrão das notificações de erro	68
Figura 4.13	Comunicação interna - llist/llistu	70
Figura 4.14	Comunicação interna - lsendl	71
Figura 4.15	Comunicação interna - lget / lsendf / linf	72
Figura 4.16	Fases de Inicialização, Anúncio e Atualização	77
Figura 4.17	Comparação de processos restritos e públicos.	78

Figura 4.18	Fluxo de monitoramento	80
Figura 4.19	Fluxo de tratamento do Cache	81
Figura 5.1	Topologia de testes LP2P	85
Figura 5.2	Integração do LP2P e apt-get	88
Figura 5.3	Tempo para atualização do sistema	90
Figura 5.4	Custo da mensagem ADD em relação a quantidade de registros	93
Figura 5.5	Comparação entre custo dos comandos DEL e ADD	94
Figura 5.6	Custo SENDL - Relação de proprietários por arquivo	95
Figura 6.1	Arquitetura LP2P com descoberta de serviço integrada	98

LISTA DE TABELAS

Tabela 2.1	<i>Endereços Classe D conhecidos</i>	40
Tabela 3.1	Comparação entre sistemas pesquisados e LP2P	49
Tabela 4.1	Equivalência de recursos nos ambientes P2P e LAN	53
Tabela 4.2	Códigos de erros LP2P	69
Tabela 4.3	Mapeamento de erros para Comunicação Interna	72
Tabela 5.1	Resultados obtidos em arquivos de 1MB, 8MB e 16MB	86
Tabela 5.2	Tempo de Download - Conexão direta e LP2P	91
Tabela 5.3	Relação de Custo - Estrutura LP2P / Proprietários / Arquivos . . .	92

LISTA DE SIGLAS

ADSL - Asymmetric Digital Subscriber Line
ALM - Application Layer Multicast
AnL - Anuncio LP2P
AoE - Ata over Ethernet
ATM - Asynchronous Transfer Mode
DHT - Distributed Hash Table
EIGRP - Enhanced Interior Gateway Routing Protocol
FRC - Fator de Replicação do Compartilhamento
FTALM - Fault Tolerant Application Layer Multicast
HDLC - High-level Data Link Control
ISCSI - Internet Small Computer System Interface
LAN - Local Area Network
MAN - Metropolitan Area Network
MANET - Mobile Ad hoc Network
OSPF - Open Shortest Path First
P2P - Peer to Peer
QoS - Quality of Service
RaL - Resposta ao Anuncio LP2P
RIP - Routing Information Protocol
RSTP - Rapid Spanning Tree Protocol
SCTP - Stream Control Transmission Protocol
STP - Spanning Tree Protocol
TDMoE - Time Division Multiplexing over Ethernet
TTL - Time to Live
WAN - World Area Network
XMPP - Extensible Messaging and Presence Protocol
WSUS - Windows Server Update Services

SUMÁRIO

1	INTRODUÇÃO	16
1.1	MOTIVAÇÃO E DEFINIÇÃO DO PROBLEMA	17
1.2	DETALHAMENTO DO PROBLEMA	19
1.3	OBJETIVOS	19
1.4	ORGANIZAÇÃO DOS CAPÍTULOS	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	VISÃO GERAL DE SISTEMAS P2P	21
2.2	CLASSIFICAÇÃO DAS REDES PEER-TO-PEER	22
2.2.1	Nível 1: Infraestrutura	23
2.2.2	Nível 2: Aplicações	24
2.2.3	Nível 3: Comunidades	25
2.3	CARACTERÍSTICAS DO P2P	26
2.3.1	Anonimato	26
2.3.2	Auto-organização	26
2.3.3	Descentralização	27
2.3.4	Desempenho	28
2.3.5	Escalabilidade	29
2.3.6	Segurança	30
2.3.7	Tolerância a Falhas	30
2.4	REDES <i>OVERLAY</i>	31
2.4.1	Arquitetura	32
2.4.2	Topologias	33
2.4.3	Roteamento	34
2.4.3.1	Centralizado	35
2.4.3.2	Inundação	36
2.4.3.3	<i>Random Walk</i>	37
2.4.3.4	DHT (<i>Distributed Hash Table</i>)	38
2.5	COMUNICAÇÃO <i>MULTICAST</i>	38
3	TRABALHOS RELACIONADOS	41
3.1	PROTOCOLOS CONSOLIDADOS NA INTERNET	41

3.1.1	Napster	41
3.1.2	Gnutella	42
3.1.3	BitTorrent	42
3.1.4	Kademlia	43
3.1.5	Pastry	43
3.1.6	Tapestry	43
3.1.7	Ivy	44
3.1.8	Squirrel	44
3.1.9	OceanStore	45
3.1.10	xFS	46
3.2	PESQUISAS RECENTES	46
3.2.1	GnutellaBT	46
3.2.2	<i>JBPeer</i>	47
3.2.3	A-Kad	47
3.2.4	<i>ADC</i>	48
3.2.5	<i>OneHop DHT</i>	48
3.3	AVALIAÇÃO DE PROTOCOLOS	49
4	<i>LOCAL PEER-TO-PEER PROTOCOL</i>	51
4.1	VISÃO GERAL	51
4.2	MESCLANDO P2P E LANS	52
4.3	OPERAÇÃO	54
4.4	ARQUITETURA	57
4.4.1	LP2P-CORE	58
4.4.2	LP2P-BD	59
4.4.3	LP2P-MSG	60
4.4.4	Módulos Externos	60
4.5	PRIMITIVAS E SISTEMA DE MENSAGENS	62
4.5.1	Grupo 1 - Manipulação de arquivos	63
4.5.2	Grupo 2 - Notificações do sistema	65
4.5.3	Grupo 3 - Segurança	66
4.5.4	Grupo 4 - Erros do sistema	68
4.5.5	Grupo 5 - Comunicação Interna	69

4.6	FASES DE FUNCIONAMENTO DO PEER	73
4.6.1	Inicialização	73
4.6.2	Anúncio	75
4.6.3	Atualização	76
4.6.4	Estabelecimento de Chaves de Sessão	78
4.6.5	Transferência de dados	79
4.6.6	Monitoramento	79
4.6.7	Envelhecimento do Cache	80
4.6.8	Comunicação Interna	82
5	METODOLOGIA E VALIDAÇÃO	83
5.1	DESCRIÇÃO DAS MÉTRICAS	84
5.2	TOPOLOGIA DOS EXPERIMENTOS	85
5.3	CENÁRIO 1 - TESTES BÁSICOS	85
5.3.1	Avaliação do cenário 1	86
5.4	CENÁRIO 2 - APLICANDO O LP2P EM AMBIENTE REAL	87
5.4.1	Sistemas de atualização de <i>software</i>	87
5.4.2	Integrando LP2P e <i>apt-get</i>	88
5.4.3	Avaliação do cenário 2	89
5.5	AVALIAÇÃO DE CUSTO DO LP2P	90
6	CONCLUSÃO	96
6.1	TRABALHOS FUTUROS	97
	BIBLIOGRAFIA	99

1 INTRODUÇÃO

Nos últimos anos, o crescimento da Internet e o surgimento de novos serviços em rede contribuíram para o desenvolvimento de várias arquiteturas destinadas à computação distribuída. Neste contexto, a tecnologia peer-to-peer¹ tem fomentado o interesse da comunidade científica em decorrência de suas características e funcionalidades inovadoras.

Em sua essência, os sistemas P2P caracterizam-se pela formação de redes descentralizadas, tendo como propósito o compartilhamento de recursos computacionais. A partir desse objetivo, tal arquitetura promove a formação de redes virtuais entre seus participantes. Para essa estrutura de conexão, é atribuída a designação de topologia *overlay*², que de acordo com certas circunstâncias exprime conformidade com as conexões reais de seus integrantes (ABERER; HAUSWIRTH, 2002).

Segundo Oram (2001), a fase inicial da Internet exibiu aspectos de funcionamento similares ao da comunicação peer-to-peer. Nesse período, tal panorama permitia a troca de pacotes entre quaisquer *hosts*, pois o uso de mecanismos de segurança (ex.: *firewall*) era inexistente. Os protocolos TELNET (POSTEL; REYNOLDS, 1983) e FTP (HETHMON, 2007) representam um exemplo claro da comunicação P2P naquele cenário, onde qualquer *host* poderia atuar de forma irrestrita como cliente e servidor dessas aplicações (denominado *SERVENT*³ (BUFORD et al., 2009)).

Em decorrência das vulnerabilidades exploradas nesse ambiente, fez-se necessária a implementação de técnicas e mecanismos de segurança que reduziram drasticamente a comunicação P2P na Internet. Contudo, a exemplo do DNS (MOCKAPETRIS, 1987) e do TELNET (POSTEL; REYNOLDS, 1983), alguns protocolos e aplicações conseguiram manter seu comportamento peer-to-peer até os dias de hoje.

Com a evolução da Internet nos últimos anos, esse método de comunicação foi readaptado à nova realidade das redes e retomado na forma de diversas aplicações distribuídas. Dentre elas, destacam-se serviços de compartilhamento, sistemas de mensagens instantâneas, trabalho colaborativo, entre outras.

Por essas razões, a área de redes peer-to-peer tem sido um tópico de intensa pesquisa

¹Um peer é a representação de um dispositivo computacional que integra um sistema distribuído denominado peer-to-peer, sendo o ponto central deste trabalho. Na opinião do autor, uma eventual tradução do termo poderá prejudicar o entendimento do tema, sendo assim, esta expressão não será traduzida. No decorrer do trabalho os termos P2P e peer-to-peer fazem referência à mesma tecnologia.

²A designação de *overlay* nesta dissertação refere-se à tecnologia que constrói uma rede de forma sobreposta a outra, sendo abordada em maiores detalhes na Seção 2.4

³De forma geral, um *SERVENT* é um participante da estrutura P2P, que pode interagir com o sistema tanto no papel de servidor quanto de cliente. Esse *host* tem a propriedade de executar tais ações simultaneamente.

nos últimos anos (MCNAMARA; YANG, 2008), (ABIONA et al., 2009), (NAKAZAWA et al., 2010), (LI et al., 2010), (ROY et al., 2010), sendo o serviço de compartilhamento de arquivos uma das fontes de estudo dentre as mais exploradas.

1.1 MOTIVAÇÃO E DEFINIÇÃO DO PROBLEMA

Um dos fatores responsáveis pela popularidade do peer-to-peer é proveniente das aplicações destinadas ao compartilhamento de arquivos. Isto é evidenciado através da quantidade de sistemas voltados a esse serviço, sendo responsáveis por parcela significativa do tráfego inserido nos *backbones IP* (MCPHERSON, 2009).

Esse serviço também ocorre de forma expressiva em redes de menor abrangência (ex.: LANs), sendo aplicado no aperfeiçoamento de diversas atividades dentro e fora do ambiente corporativo. Contudo, a execução dessa funcionalidade em redes locais necessita de infraestrutura específica para armazenamento de dados, visando garantir a disponibilidade do conteúdo independentemente de sua criticidade ou função.

Essa característica torna-se nítida quando observada em grandes redes de corporações ou universidades, compostas por centenas de equipamentos interligados. Nesses cenários, ocorre simultaneamente a demanda constante por armazenamento aliado ao desperdício de *hardware* destinado ao mesmo propósito e disponível em cada estação de trabalho.

Além disso, no ambiente LAN o compartilhamento de arquivos é fornecido de forma centralizada, representando ponto único de falha e gargalo da rede. Conseqüentemente, existe a probabilidade de paralisação parcial ou total desse serviço. Outro problema decorrente dessa atividade nas LANs está na replicação do conteúdo. Tal fator é muitas vezes desnecessário ou mal-utilizado, exigindo, em muitos casos, investimentos em *software* ou *hardware* especializado para reduzir esse comportamento (Deduplication (CONSTANTINESCU et al., 2009)).

Uma possível alternativa para minimizar tais problemas está na adaptação de alguns conceitos do paradigma P2P à infraestrutura de rede local. No entanto, há poucos esforços no sentido de explorar o potencial da comunicação peer-to-peer aliada aos padrões e vantagens disponíveis nas LANs.

Em virtude de sua natureza distribuída, as abordagens P2P convencionais focalizam seu desenvolvimento e aplicação em redes geograficamente distribuídas (MANs e WANs) para formação da estrutura *overlay*. Esse comportamento implica na utilização de diversos métodos de acesso à rede, como apresentado na Figura 1.1. Contudo, tais fatores não inviabilizam a utilização de algumas técnicas e características da tecnologia P2P em redes de menor abrangência. Essa convergência permite a criação de sistemas peer-to-peer

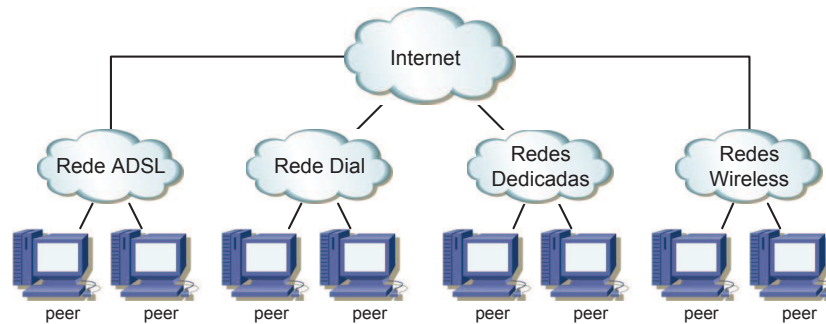


Figura 1.1: Topologia heterogênea em Redes P2P

independentes da infraestrutura *overlay*, o que não ocorre em outros protocolos destinados ao compartilhamento de arquivos (ex.: BitTorrent (DHUNGEL et al., 2010)).

O método de acesso padronizado, em conjunto com a maior largura de banda disponível nas redes locais, possibilita a formação de redes P2P descentralizadas (Pure P2P (OHZAHATA; KAWASHIMA, 2008)). Esse fator dispensa o uso de servidores centrais (ex.: Napster (HUANG et al., 2009)) ou de *super-peers* (ex.: Gnutella (BOUKERCHE et al., 2010)), bem como o emprego de algoritmos complexos, como o DHT (WANG et al., 2009), para tratar do gerenciamento de conteúdo compartilhado e peers conectados ao sistema.

Outra vantagem de um sistema que englobe tal conjunto de características é o fornecimento de compartilhamentos descentralizados, facilitando a criação de aplicações integradas a este recurso. Por exemplo, uma arquitetura de *backup* secundário funcionando paralelamente à solução principal proporcionaria maior agilidade na recuperação de informações em comparação ao sistema convencional. Além disso, esta solução seria capaz de concatenar logicamente todo recurso de armazenamento ocioso disponível em cada *host* para fornecer compartilhamentos públicos.

Uma consequência deste aspecto é a otimização dos recursos da rede, característica não contemplada pelos sistemas peer-to-peer tradicionais. Essa funcionalidade pode ser expandida através de ajustes finos, visando fornecer granularidade de replicação de acordo com a criticidade do recurso compartilhado. Logo, o grau de replicação influenciará diretamente na ocupação do espaço em disco, o qual está distribuído entre os *hosts* conectados ao sistema.

Nesse contexto, propor uma solução baseada na arquitetura peer-to-peer que consiga otimizar o uso do armazenamento disponível em redes locais é o problema central abordado neste trabalho.

1.2 DETALHAMENTO DO PROBLEMA

A definição do problema pode ser mais detalhada com a sua decomposição nas seguintes questões de pesquisa:

- É possível criar um modelo que agregue os conceitos encontrados em redes locais e na comunicação P2P?
- Quais aspectos inerentes a tecnologia P2P são factíveis de implementação no contexto das LANs?
- Como tal abordagem irá executar o gerenciamento dos recursos?
- Que aspectos de segurança poderiam ser garantidos nesta proposta?

1.3 OBJETIVOS

Com base nos problemas e cenários identificados, o objetivo fundamental desta dissertação está delimitado na definição e desenvolvimento do —Local Peer-to-Peer-Protocol—, sistema que utiliza técnicas P2P para prover compartilhamento de conteúdo em redes locais. Sendo assim, os objetivos específicos deste trabalho são:

- Definir uma arquitetura ao modelo LP2P, que permita desempenhar o serviço especificado;
- Estabelecer a relação entre cada módulo dessa arquitetura de comunicação, bem como a correlação entre as partes e respectivas funcionalidades dessa pilha de protocolos;
- Planejar a base de conhecimento necessária para o protocolo registrar a relação de peers, estados do sistema e compartilhamentos existentes na rede;
- Determinar o padrão de mensagens que será empregado entre distintos *hosts* LP2P e quais as regras para o envio e recebimento desse conjunto de pacotes;
- Criar conjunto de mensagens padronizado para comunicação interna do *host* com os demais agentes locais, módulos do *kernel*, ou servidores *web*;
- Definir os aspectos de segurança aplicados aos compartilhamentos, estabelecendo requisitos para o gerenciamento de chaves de sessão entre *hosts* que desejam trocar dados em áreas seguras;
- Validar o modelo proposto através do desenvolvimento e testes reais dessa pilha de protocolos, juntamente da análise do fluxo de mensagens envolvidas no provimento deste serviço.

1.4 ORGANIZAÇÃO DOS CAPÍTULOS

O restante desta dissertação está organizado da maneira descrita a seguir.

- *Capítulo 2:* Aborda os conceitos fundamentais para o entendimento deste trabalho, englobando as áreas de redes peer-to-peer, sistemas distribuídos, comunicação *multicast* e compartilhamento de recursos. Enfatiza-se o conjunto de informações que formam a base para definições posteriores deste estudo;
- *Capítulo 3:* Ilustra o estado da arte quanto a tecnologia peer-to-peer, através de literatura relacionada ao tema, dando enfoque às soluções de compartilhamento de arquivos e algoritmos correspondentes;
- *Capítulo 4:* Apresenta a proposta do LP2P, contemplando sua arquitetura, grupos de mensagens, sistema de comunicação, entre outros aspectos necessários ao desenvolvimento deste protocolo;
- *Capítulo 5:* Descreve a metodologia empregada na produção deste trabalho, reunindo as métricas de avaliação, os cenários de testes e a descrição completa dos resultados obtidos;
- *Capítulo 6:* Promove as considerações finais deste estudo, indicando novas possibilidades de pesquisa e trabalhos futuros na área.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo enfatiza os principais conceitos sobre sistemas P2P e comunicação *multicast*, os quais são base para a compreensão deste trabalho, abordando aspectos como definição, arquitetura, topologias, classificação e respectiva abrangência de suas funcionalidades. Na sequência, as características principais são apresentadas e discutidas em detalhes, visando fornecer os subsídios necessários para o desenvolvimento desta proposta.

Dessa forma, a Seção 2.1 aborda aspectos gerais das redes peer-to-peer – foco deste trabalho. A Seção 2.2 classifica as redes P2P em relação a funcionalidades e respectiva abrangência. As características dessa arquitetura são abordadas na Seção 2.3, de modo a familiarizar o leitor com as possibilidades decorrentes do uso dessa solução. Os aspectos de conectividade entre peers com o uso das redes *overlay* são descritos na Seção 2.4. Por fim, a Seção 2.5 abrange a tecnologia *multicast*, parte integrante da arquitetura foco deste estudo.

2.1 VISÃO GERAL DE SISTEMAS P2P

A respectiva abrangência de funcionalidades proporcionou uma série de definições para o termo peer-to-peer. Partindo das primeiras implementações dessa arquitetura, tal modelo é definido de forma totalmente descentralizada, no qual cada participante possui as mesmas atribuições, fornecendo e solicitando recursos computacionais. Contudo, o avanço dessa tecnologia teve como consequência a expansão desse conceito, tornando sua definição mais abrangente em relação ao anterior.

Dessa forma, visando englobar todos os níveis de independência desse sistema ao tradicional modelo cliente-servidor, a definição dada por Androutsellis-Theotokis e Spinellis (2004) será utilizada neste estudo em virtude de sua abrangência. Nesta, define-se que as redes peer-to-peer são uma classe de sistemas distribuídos, conforme a Figura 2.1, na qual a topologia de rede é auto-organizável em decorrência das atividades de seus participantes, em que tal aspecto objetiva o compartilhamento de recursos como dados, processamento, armazenamento e largura de banda, entre outros. Além disso, tais sistemas possuem resiliência suficiente contra falhas, de forma a proporcionar devida redução ou aumento de seus integrantes sem prejudicar a conectividade entre participantes ativos, mantendo desempenho aceitável sem necessitar de coordenação central para controle de suas ações.

Adicionalmente a comunidade acadêmica concorda que essa arquitetura deve prover conectividade variável ou temporária, bem como endereços variáveis em todos os peers, os

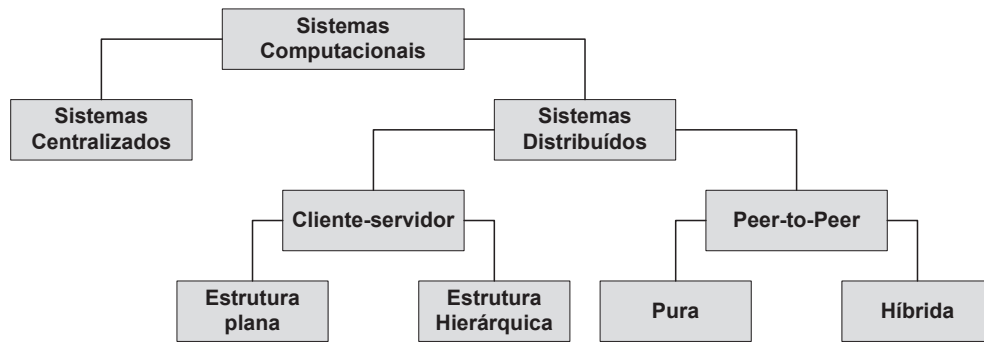


Figura 2.1: Classificação de sistemas computacionais

quais estão sempre localizados na camada de acesso da rede. Contudo, todo sistema deve ser capaz de tratar diferentes taxas de comunicação entre peers, provendo determinado grau de autonomia em relação a servidores centrais. Qualquer peer deve estar apto a fornecer e consumir recursos, conectando-se diretamente a outro peer e proporcionar escalabilidade da rede (DOMINGUES et al., 2004).

Dentre os diversos aspectos positivos que fizeram do P2P uma tecnologia promissora, estão a facilidade no compartilhamento de recursos e a inexistência de restrições quanto a quantidade de participantes do sistema. Em decorrência de tais fatores, esse paradigma pode oferecer várias vantagens, destacando-se:

- Maior poder computacional decorrente do aproveitamento de tempo ocioso em cada peer;
- Adaptatividade dos peers, permitindo que estes executem tarefas tanto como servidores quanto clientes;
- Redução de custos para o compartilhamento de recursos;
- Manutenção da autonomia dos peers participantes.

2.2 CLASSIFICAÇÃO DAS REDES PEER-TO-PEER

De acordo com Coulouris et al. (2005), a tecnologia peer-to-peer pode ser representada por um modelo de três níveis que classifica todos os aspectos técnicos correlatos ao tema. O conjunto dessas camadas, apresentado na Figura 2.2, engloba quesitos referentes a infraestrutura, aplicações e comunidades P2P.

Desses níveis é importante enfatizar que o termo peer é visto de forma distinta entre camadas 1 e 2 em relação à camada 3. No primeiro grupo o termo se refere basicamente às entidades técnicas que fornecem o aparato necessário para a troca de informações,

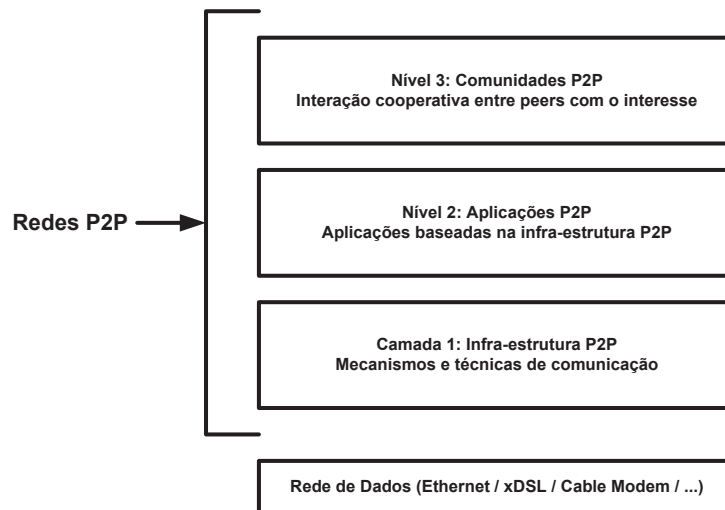


Figura 2.2: Níveis da arquitetura P2P

ao contrário da camada superior que está relacionada essencialmente com o usuário do sistema.

2.2.1 Nível 1: Infraestrutura

Esse nível trata dos mecanismos e técnicas primordiais para comunicação, integração e troca de informações do peer para a rede convencional, bem como do peer para a respectiva aplicação que faz uso da infraestrutura P2P. Essa camada atua como uma plataforma de serviço P2P padronizada, podendo ser empregada por qualquer aplicação e garantindo, conseqüentemente, a interoperabilidade entre diferentes sistemas peer-to-peer. Dentre seus principais serviços, destacam-se a identificação, localização e comunicação com os demais peers da rede aliados a quesitos de segurança como autenticação e autorização.

Atualmente, existem várias propostas para a padronização da infraestrutura P2P visando garantir sua interoperabilidade, dentre as quais destacam-se:

- **JXTA:** Seu projeto objetiva fornecer uma plataforma de colaboração que ofereça conectividade padronizada para sistemas distribuídos em larga escala, com total interoperabilidade e totalmente ubíqua, isto é, independente do tipo de dispositivo empregado na conexão (ABERER; HAUSWIRTH, 2002).
- **Jabber:** Esse projeto, também denominado de XMPP (*Extensible Messaging and Presence Protocol*), provê um sistema de mensagens instantâneas que já está em uso em provedores (Google) e outras plataforma de comunicação (Openfire¹, Tigase²

¹<http://www.igniterealtime.org/index.jsp>

²<http://www.tigase.org/>

e Ejabberd³). Jabber enquadra-se como infraestrutura P2P, pois objetiva fornecer uma camada de abstração para diferentes plataformas se comunicarem entre si, tais como MSN, AOL, Yahoo, GoogleTalk, entre outras (ORAM, 2001).

- **Magi:** Trata-se de uma plataforma destinada ao desenvolvimento de aplicações colaborativas. Emprega padrões WEB (HTTP, WebDAV, e XML), permitindo a comunicação entre aplicações sobre uma infraestrutura P2P (COULOURIS et al., 2005).
- **Groove:** É um sistema P2P colaborativo que também faz o papel de plataforma de comunicação. Seu principal objetivo é permitir a comunicação, compartilhamento de conteúdo e ferramentas para o desenvolvimento de atividades colaborativas de forma ubíqua (MILOJICIC et al., 2002a).

2.2.2 Nível 2: Aplicações

Um dos fatores responsáveis pela popularização das redes peer-to-peer esta aliado ao seu rol de aplicações, dentre os quais destacam-se os seguintes grupos:

Compartilhamento de arquivos: Este recurso é um dos responsáveis pelo sucesso e crescimento da fama das aplicações peer-to-peer, pois proporcionou de forma eficaz o intercâmbio de informações entre usuários, mantendo o respectivo anonimato das partes. Além dos pontos favoráveis mencionados anteriormente, o compartilhamento de arquivos fornece ilimitado espaço de armazenamento com total redundância ao manter o mesmo conteúdo em diversos peers. Essa ação denominada de replicação ocasiona inerentemente um fator de alta disponibilidade ao conteúdo. Não menos importante, essa arquitetura está apta a gerenciar de forma fácil e rápida os dados distribuídos na rede por meios de caches localizados em pontos estratégicos dessa estrutura (PIRES, 2007).

Computação Distribuída: Esse aspecto contempla o melhor empenho do *hardware*. Já foi comprovado que, em grande parte do tempo, estações de trabalho e até mesmo servidores permanecem ociosos. Atualmente, conceitos como processamento de alto desempenho, virtualização (BARHAM et al., 2003) e arquiteturas distribuídas são empregados para tratar desse problema. Visando essa demanda, a arquitetura P2P utiliza-se desses recursos para prover processamento similar ou até mesmo superior a computadores de grande porte com o atrativo de realizar tal tarefa a baixo custo de *hardware* e gerenciamento. Nesse sentido, uma aplicação P2P que tenha conformidade com esse quesito irá formar um grupo independente de peers e,

³<http://www.ejabberd.im/>

de forma transparente, aparecerá ao usuário apenas como um recurso, tendo o poder de processamento de todos os nós integrados em bloco único (COULOURIS et al., 2005).

Trabalho colaborativo: Segundo Pinheiro (2006), o trabalho colaborativo são aplicações que viabilizam a comunicação e colaboração entre diversos *hosts* para execução conjunta de atividades em comum, integrando para este fim várias aplicações como *e-mail*, calendário, gerenciamento de documentos e videoconferência. Essa iniciativa teve início nos ambientes locais de rede, sendo incorporada ao paradigma P2P de forma a prover novos horizontes na integração de usuários nesse cenário. O gerenciamento dinâmico dos grupos de trabalho, bem como a transposição de restrições quanto a localização dos usuários são consequências positivas dessa integração. Entretanto, existem questões que ainda merecem cuidado nesse contexto, como, por exemplo, a tolerância a falhas e a necessidade comunicações em tempo real, quesitos que apresentam diferentes comportamentos das redes LANs para a arquitetura P2P. Em ambos os casos, tais fatores podem prejudicar a sincronização de ações entre participantes, seja no acompanhamento de uma videoconferência ou na edição conjunta de documentos. Sendo assim, o sistema deve favorecer alta disponibilidade, bem como baixo retardo na comunicação para apresentar conformidade com a entrega confiável de mensagens na execução de tarefas colaborativas (AHMED; SHIRMOHAMMADI, 2006). Segundo Domingues et al. (2004), algumas aplicações bem difundidas dentro dessa área são o Microsoft NetMeeting e Groove.

Troca de mensagens: Essa categoria de aplicações está entre as mais utilizadas na Internet. Isso se deve ao seu mecanismo de funcionamento que promove a comunicação em tempo real por meio do envio de mensagens instantâneas, característica a qual os sistemas de correio eletrônico não atendem. Existe atomicidade nas ações de envio e recebimento de mensagens - fator que implica no uso de controles específicos para informar aos usuários quem está presente no sistema através de listas de contatos. Dessa forma, cada usuário pode constatar quem está *on-line* e iniciar uma comunicações instantânea, ou, ainda, nas plataformas atuais, existe a possibilidade de envio de mensagens *off-line* armazenadas em servidores centrais e entregues ao usuário desejado no momento de seu ingresso na rede. Em decorrência de sua popularidade, várias plataformas para esse fim estão disponíveis na rede, como ICQ, AOL Instant Messenger, JABBER e MSN Messenger (DOMINGUES et al., 2004).

2.2.3 Nível 3: Comunidades

De acordo com Coulouris et al. (2005), a representação de indivíduos que possuem interesses em comum, que independem de posição geográfica, bem como utilizam de

algum artifício tecnológico para integração é denominado uma comunidade virtual. Como consequência de sua natureza distribuída, a tecnologia peer-to-peer apresenta-se como uma plataforma ideal para criação de comunidades. Atualmente estão disponíveis na Internet uma variedade de comunidades com os mais variados propósitos. Nesse quesito destacam-se as soluções para troca de arquivos como BitTorrent (LEVIN et al., 2008) e o sistema SETI@HOME, que ilustram a relação do termo comunidade à forma de emprego da arquitetura peer-to-peer.

2.3 CARACTERÍSTICAS DO P2P

O paradigma peer-to-peer é composto por uma diversidade de características para seu devido funcionamento. Algumas dessas são altamente dependentes ou correlacionadas a outros aspectos da arquitetura. Esse grau de relacionamento dependerá do tipo de estrutura ou aplicação a ser implementada. Esta seção objetiva apresentar tais aspectos, base necessária para o desenvolvimento de novos protocolos desse modelo de comunicação.

2.3.1 Anonimato

O anonimato é um dos objetivos da maioria dos sistemas P2P, evitando que a censura digital seja possível nesse ambiente. Para que essa característica seja plenamente atendida, o conjunto de entidades que integram a comunicação entre os peers não deve ser divulgado (ANDROUTSELLIS-THEOTOKIS; SPINELLIS, 2004).

Uma arquitetura P2P em conformidade com essa característica não pode permitir que qualquer das partes que compartilham uma informação postando ou recebendo documentos seja identificada. Sendo assim, um peer na função de servidor não deverá ter conhecimento de quais documentos está armazenando, bem como este não poderá informar que documento utiliza em resposta a uma determinada consulta originada por outro peer.

Esse conjunto de prerrogativas converge para três tipos diferentes de anonimato promovidos pela arquitetura P2P. A identidade do remetente é omitida quando o sistema dá ênfase ao anonimato do remetente. De forma oposta, outros sistemas agem apenas em relação à privacidade do receptor, omitindo do sistema a identificação desta parte. O anonimato mútuo ocorre quando a identidade de todos os integrantes da comunicação é omitida das partes (MILOJICIC et al., 2002a).

2.3.2 Auto-organização

O fator de auto-organização na arquitetura P2P é de extrema importância, influenciando outras características como escalabilidade e tolerância a falhas. Basicamente, é

definida como o processo em que a organização do sistema aumenta ou reduz de forma espontânea como resposta às modificações na topologia da rede *overlay* (ZAHID et al., 2005).

A constante entrada e saída de peers do sistema, uma das características fundamentais deste paradigma, traz como consequência um fator de imprevisibilidade ao verdadeiro número de *hosts* participantes da rede. Tal comportamento exige frequentes reconfigurações para lidar de forma eficiente com essas mudanças, tarefa com elevado grau de complexidade e inviável de ser executada manualmente (BARCELLOS; GASPARY, 2006).

Consequentemente, todo projeto de sistema P2P em conformidade com a auto-organização não utilizará uma topologia estrela ou comunicação em *broadcast* para troca de informações com os demais peers. A topologia obtida nesse contexto será totalmente descentralizada e nenhum integrante terá conhecimento total dos peers conectados à rede. Em contrapartida, todos os integrantes cooperam na formação e manutenção dessa estrutura (BUFORD et al., 2009).

2.3.3 Descentralização

Segundo Coulouris et al. (2005), a descentralização, uma consequência imediata ao compartilhamento de recursos, implica em uma estrutura desprovida de ponto central, na qual cada participante exerce as mesmas funções atuando de forma autônoma. Nesse sentido, o modelo de comunicação peer-to-peer apresenta-se como uma alternativa viável frente às soluções centralizadas baseadas na arquitetura cliente-servidor.

No modelo tradicional, os dados são armazenados em servidores centrais e distribuídos mediante solicitação. Esse sistema é ideal para determinados cenários, facilitando aspectos como segurança e confiabilidade da informação. Entretanto, essa topologia é sujeita a picos de utilização da rede ou servidores. Mesmo com os avanços tecnológicos que proporcionaram melhor desempenho do *hardware* e maiores taxas de transmissão, essa abordagem permanece com um custo administrativo elevado. Adicionalmente, as soluções centralizadas ainda necessitam de interferência humana para sua respectiva implantação e manutenção (ZAHID et al., 2005).

Além da redução de custos operacionais, uma solução descentralizada proporciona maior interesse frente à abordagem tradicional, pois tal característica pode ser empregada para garantir alta disponibilidade da rede e evitar gargalos no sistema que venham a prejudicar seu respectivo desempenho. Gnutella (ANDROUTSELLIS-THEOTOKIS; SPINELLIS, 2004) e Freenet (ABERER; HAUSWIRTH, 2002) são exemplos de sistemas peer-to-peer totalmente descentralizados.

Uma das principais características referentes à descentralização da informação está

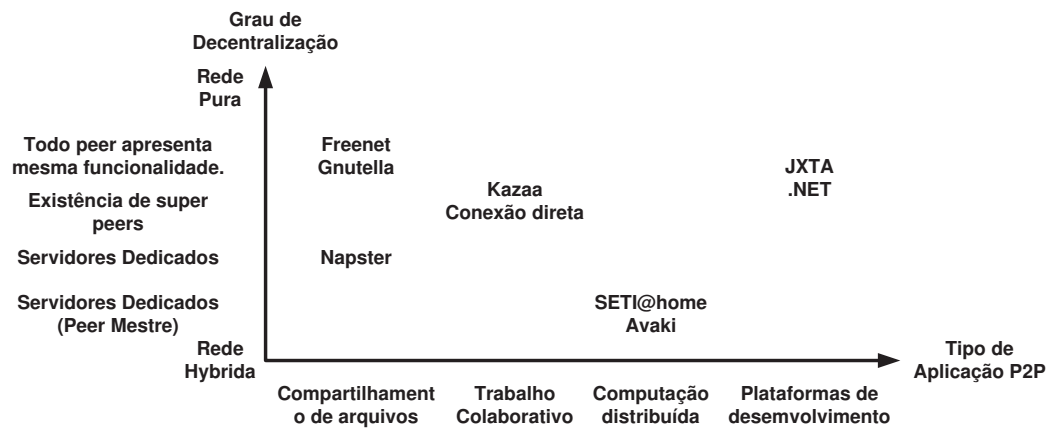


Figura 2.3: Descentralização de sistemas P2P

relacionada com a participação dos usuários no controle de recursos e dados do sistema. Ressalta-se que nos ambientes completamente descentralizados todo peer é um agente de funções equivalentes. Esse aspecto torna a implementação desse modelo de comunicação mais complexa, pois, em sua essência, a função de um dispositivo central, com visão global de toda estrutura, é descartada. Como resultado desse cenário, vários sistemas de comunicação P2P foram desenvolvidos de forma híbrida, unificando conceitos do modelo tradicional aos novos da comunicação peer-to-peer (MILOJICIC et al., 2002a).

É possível mensurar o grau de autonomia de um sistema P2P através da classificação deste frente aos modelos de comunicação descentralizados e híbridos. A Figura 2.3 relaciona o grau de autonomia de alguns sistemas em relação ao seu tipo de aplicação, escalabilidade, bem como o nível de dependência relativo a servidores.

2.3.4 Desempenho

De acordo com Milojicic et al. (2002a), uma consequência decorrente da abordagem P2P diz respeito ao desempenho que essa plataforma estabelece. Seu paradigma descentralizado melhora esse fator ao unificar recursos espalhados na rede para atendimento de uma determinada requisição, seja esta de armazenamento ou processamento de informações.

Ressalta-se que a essência da arquitetura é a completa descentralização que traz diversos benefícios e desafios para seu pleno desenvolvimento. Nesse contexto, o desempenho é influenciado pelo processamento, armazenamento e conexões de rede, sendo o último de maior impacto nos casos em que determinada aplicação distribuída necessita de maior vazão ou envio de mensagens - fatores que limitam a escalabilidade do sistema (MILOJICIC et al., 2002a).

Comparando as principais abordagens de rede P2P, aquelas que se utilizam de coordenação centralizada possuem as mesmas vulnerabilidades encontradas nas convencionais

redes do tipo cliente-servidor. Mesmo que esses peers atuem apenas na fase de inicialização, eles poderão sofrer sobrecarga devido ao grande número de requisições, latência da rede ou gargalo de algum componente desse agente.

Diferentemente da abordagem anterior, nas redes descentralizadas a comunicação é realizada individualmente - peer a peer. Entretanto, esse método de comunicação exige maior vazão da rede devido ao elevado número de mensagens de busca que trafegam na estrutura *overlay*. O consumo de recursos necessário para uma busca nesse cenário é proporcional ao volume de mensagens enviadas, que por sua vez, é proporcional à quantidade de peers na rede.

Para propiciar melhor desempenho, independentemente da abordagem adotada, foram propostos mecanismos de replicação, *cache* e roteamento otimizado. A replicação consiste em espalhar cópias o mais próximo possível do peer que solicitou determinado conteúdo, reduzindo assim a distância entre emissor e receptor dos dados. O *cache* objetiva reduzir o caminho necessário para procura de recursos, melhorando o desempenho através da minimização de mensagens de busca na rede e, conseqüentemente, reduzindo os problemas de latência do sistema. Finalmente, o roteamento inteligente visa agrupar de forma organizada as conexões entre os integrantes da rede, manipulando dinamicamente as conexões entre peers para agrupá-los de acordo com requisitos de interesse comum (BUFORD et al., 2009).

2.3.5 Escalabilidade

A escalabilidade da arquitetura P2P representa a propriedade de oferecer suporte a milhares de conexões simultâneas e relaciona-se com o fator de autoadaptação da rede, aumentando e reduzindo a quantidade de participantes sem proporcionar qualquer sobrecarga na estrutura. Tal flexibilidade é vista em diversos sistemas desenvolvidos para trabalhar de forma descentralizada que apresentam taxas de crescimento muito superiores em relação à quantidade de integrantes inicial (RATNASAMY et al., 2001).

Comparados à arquitetura cliente-servidor, os sistemas peer-to-peer apresentam vantagens neste quesito. Isso deve-se ao fato da arquitetura tradicional dispor de ponto único de falha e gargalo do sistema. Como consequência, essa rede poderá apresentar maior latência e menor qualidade para atendimento de requisições em momentos de pico ou nos casos de sobrecarga de seus servidores (ROUSE; BERMAN, 2006).

Entretanto, tal aspecto envolve certo custo adicional de desenvolvimento, pois, diferentemente do modelo tradicional no qual as necessidades de ampliação são executadas de forma coordenada por meio dos administradores do ambiente, nos sistemas P2P esse fator deve estar intrínseco em cada peer. Esse compartilhamento de responsabilidades garante o gerenciamento da conectividade, bem como da extensão da infraestrutura *overlay* e

respectivo balanceamento de carga.

Para atender a essa característica existem soluções desenvolvidas para cada tipo de topologia utilizada. Em redes híbridas, o tratamento é similar ao executado nos sistemas cliente-servidor. Nas topologias descentralizadas, existe a possibilidade de sobrecarga do sistema, ocasionada pelo aumento de nós na rede, gerando maior quantidade de mensagens entre os peers devido ao processo de descoberta e busca de conteúdo (MILOJICIC et al., 2002a).

2.3.6 Segurança

A frequente entrada e saída de integrantes da rede, juntamente com sua natureza distribuída, impactam em questões relacionadas à segurança da informação. Conforme Barcellos e Gasparly (2006), mesmo com a fundamental contribuição das redes peer-to-peer para compartilhamento e colaboração em ambientes distribuídos, sua implementação em larga escala de forma segura continua sendo um desafio a ser superado.

De acordo com Coulouris et al. (2005), o uso de redes peer-to-peer em larga escala deve estar vinculado à implementação de fatores de segurança que protejam esses ambientes contra ataques ou ações danosas ao sistema ou sua política de compartilhamento de recursos.

Essa arquitetura deve estar apta a identificar pacotes indevidos, como, por exemplo, respostas erradas a determinada consulta, bem como coibir os usuários que objetivam enganar o sistema por meio da disparidade entre recursos procurados em relação aos ofertados (MILOJICIC et al., 2002a). Um caso clássico dessa ação está em definir banda inferior ao que o peer dispõe para compartilhar, forçando que integrantes da rede busquem o referido recurso em outros pontos.

2.3.7 Tolerância a Falhas

Na arquitetura peer-to-peer o comportamento do sistema é influenciado pelo conjunto de ações que cada peer exerce na rede, tornando este altamente dependente do comportamento que cada membro exerce. Nesse contexto, uma entrada ou saída de peer da rede pode prejudicar o encaminhamento de dados, comprometendo a interação entre *hosts* localizados em extremos distintos da rede. Uma solução empregada frequentemente consiste em prover múltiplas rotas para o mesmo canal de comunicação. Em se tratando dos sistemas peer-to-peer, o gerenciamento das ações de entrada e saída de *hosts*, em conjunto com algoritmos de roteamento favorecem o estabelecimento de técnicas de redundância para prover tolerância a falhas nesse ambiente (ABERER et al., 2005).

Em sistemas P2P com topologias não estruturadas, a tolerância a falhas é atingida por

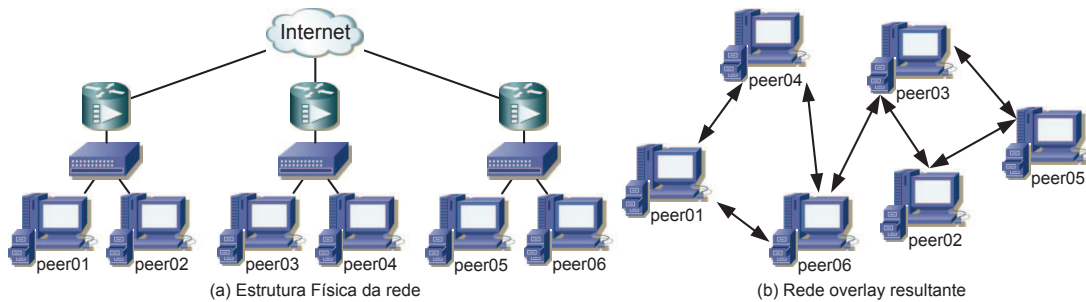


Figura 2.4: Rede *overlay*

meio do registro de enlaces dinamicamente tão logo essa relação de peers seja descoberta através de algum mecanismo de roteamento. Outra forma de dispor desse recurso é propiciar maior disponibilidade dos dados na rede através das várias técnicas de replicação desenvolvidas nos últimos anos (MILOJICIC et al., 2002a).

2.4 REDES *OVERLAY*

A infraestrutura *overlay* refere-se a qualquer rede de computadores construída de forma sobreposta a outra topologia existente. Segundo Domingues et al. (2004), um exemplo dessa relação de sobreposição está no próprio protocolo IP que monta sua estrutura de conexões sobre outras tecnologias como ADSL, ATM, Frame Relay, HDLC, entre outras. Conforme ilustra a Figura 2.4, a topologia, bem como o relacionamento de conexões dessa rede virtual não possui qualquer relação com sua estrutura real. As conexões entre peers são lógicas, necessitando em muitos casos de diversos enlaces físicos para seu estabelecimento.

Para a plataforma P2P, esse modelo de conectividade proporciona uma série de vantagens, através do armazenamento de dados de forma distribuída, dos mecanismos para localização de conteúdo e do roteamento de mensagens. De uma forma geral, o relacionamento de um peer com essa estrutura é dividido no seu respectivo ingresso na rede, bem como nas operações de consulta, acesso aos dados e, por fim, saída desse ambiente.

No momento de ingresso na rede, o peer deve se comunicar com outros *hosts* para informar que a topologia *overlay* possui um integrante adicional. Imediatamente este divulga sua base local de informações, compartilhando-as com elementos diretamente conectados a ele. Além disso, o novo integrante objetiva buscar alguma informação nessa estrutura, e, para este fim, mensagens contendo solicitações para busca de recursos são enviadas a seus vizinhos em conformidade com algum mecanismo de roteamento e busca preestabelecido para aquele sistema. Tão logo seja encontrado, a transferência desse recurso é iniciada.

Quando as atividades de busca e transferência se encerrarem, o peer pode arbitrariamente deixar esse sistema ou permanecer conectado - de forma a contribuir com o restante

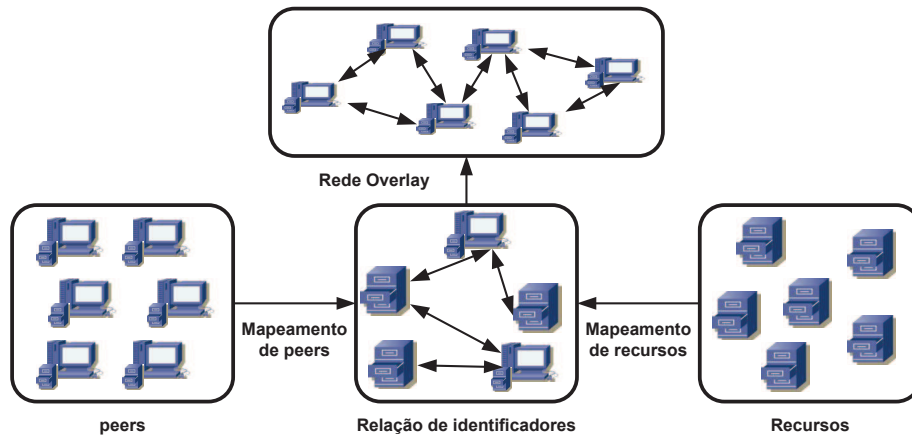


Figura 2.5: Mapeamento de peers e recursos para estrutura *overlay*

dessa comunidade, no sentido de aumentar a capacidade de compartilhamento e robustez da rede.

2.4.1 Arquitetura

Em decorrência da quantidade de sistemas peer-to-peer, se faz necessário determinar aspectos comuns entre essas soluções, de forma a padronizar por meio de um modelo de referência o desenvolvimento de novas aplicações dessa arquitetura. Segundo Aberer et al. (2005), o termo *overlay* refere-se a uma rede de computadores sobreposta em outra estrutura de conectividade. No que se refere à arquitetura peer-to-peer, a estrutura *overlay* é utilizada para o acesso a determinados recursos. A relação entre recursos disponíveis e respectivos nós que os detêm deve ser mapeada para um conjunto de identificadores. Além disso, é necessário definir uma forma coerente de categorizar identificadores de peers aos identificadores de recursos.

De acordo com a Figura 2.5, essa definição possibilita uma organização hierárquica dos dados, de forma a otimizar a topologia *overlay*, bem como as atividades de pesquisa na rede. No que se refere aos aspectos principais da estrutura *overlay*, além dos mecanismos de roteamento existem os seguintes fatores empregados em grande parte dessas estruturas:

- **Definição do espaço de identificadores:** Realiza o endereçamento de recursos e *hosts* da rede. Proporciona independência quanto à localização física dos nós e, dependendo da aplicação, pode manter informações de estado em virtude desse identificador possuir caráter lógico. Existe a necessidade de gerenciamento desse recurso para relacionar de forma hierárquica identificadores de recursos e peers (ABERER et al., 2005).
- **Relacionamento entre recursos, peers e identificadores:** No mapeamento de recursos e peers para a estrutura *overlay*, é normalmente utilizada uma função

específica para essa designação. Para isso algum identificador do *host*, como, por exemplo, o endereço IP, é aplicado a essa função, que apresenta como retorno um identificador único do *host* no espaço de endereçamento da rede *overlay*. No caso dos recursos, seus dados são aplicados a mesma função, gerando uma designação exclusiva ao mesmo. Dessa forma, é necessário que o espaço de armazenamento seja suficientemente grande para comportar a designação de identificadores para todos os *hosts* e recursos presentes (BUFORD et al., 2009).

- **Gerência de identificadores:** O gerenciamento do conjunto de identificadores é distribuído entre os integrantes do sistema. Cada peer tem conhecimento de seu identificador, bem como do conjunto de identificadores que o mesmo deve gerenciar. Devido à dinamicidade da rede em decorrência da constante entrada e saída de peers, ocorre de um determinado identificador de recurso estar sobre a responsabilidade de mais de um peer. Isso proporciona tolerância a falhas ao acesso do recurso, mas também acarreta em melhor desempenho por conta do surgimento de várias fontes ao mesmo item.

Com essa sistemática, tanto a designação quanto a busca de um determinado conteúdo estará condicionada à comparação do índice desejado em relação ao identificador do peer mais próximo desse valor. Um exemplo dessa relação é apresentado em maiores detalhes na Seção 2.4.3.

- **Estrutura da rede:** Em sua essência, uma rede peer-to-peer deve suportar um grande número de integrantes, inviabilizando que o conhecimento de toda topologia resida nos peers. Logo, um *host* necessita apenas conhecer um pequeno grupo de participantes para alcançar qualquer informação disponível no ambiente, pois dessa forma todos os nós estarão conectados mesmo que indiretamente. No que tange aos projetos de redes peer-to-peer, esse cenário está ligado ao fenômeno "small world", que, segundo Aberer e Hauswirth (2002), implica no fato de que em um determinado grupo existe um pequeno número de conexões intermediárias capaz de relacionar qualquer par de membros do grupo. Conseqüentemente, se houver maior número de conexões e pares envolvidos na rede, maior conectividade e desempenho existirá no ambiente devido à quantidade de caminhos disponíveis entre seus participantes.

2.4.2 Topologias

As topologias de rede que a arquitetura peer-to-peer adota podem ser classificadas de várias maneiras, de acordo com suas características. De uma forma genérica, as topologias de rede se dividem em pura, híbrida e superponto, sendo esta derivada dos primeiros modelos. Além disso, o modelo puro pode ser classificada como estruturada e não estruturada.

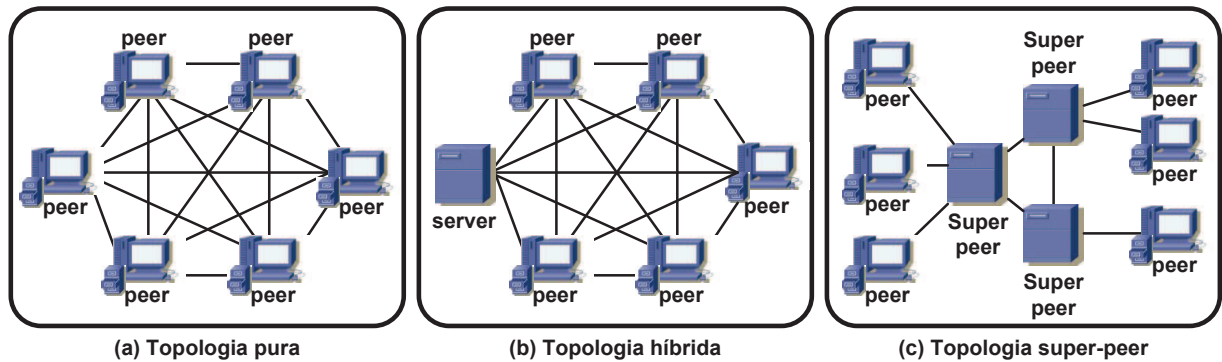


Figura 2.6: Tipos de topologias peer-to-peer

A topologia pura - também conhecida como descentralizada -, segue fielmente a arquitetura peer-to-peer. Conforme é ilustrado na Figura 2.6a, o funcionamento dessa rede é totalmente descentralizado, sendo essa a sua principal característica. Cada integrante é efetivamente um *servent*, responsável pela manutenção de suas informações, bem como por responder e executar requisições por conteúdo. No caso do modelo puro estruturado, existem mecanismos de controle para localização dos recursos, sendo que cada peer gerencia seus recursos e índices localizados em outros integrantes da rede. Com relação ao modelo puro não estruturado, as restrições quanto aos métodos de busca, distribuição e pesquisa de recursos são inexistentes. Dessa forma, uma abordagem do tipo *flooding* é normalmente empregada nesse cenário, acarretando um baixo desempenho da rede (ABERER; HAUSWIRTH, 2002).

Na topologia híbrida, conforme ilustra a Figura 2.6b, é empregado um servidor principal para armazenamento da localização de recursos na rede. O servidor central fica encarregado por receber e manter o catálogo de informações de recursos, cabendo aos outros integrantes consultar esse registro e, após, transferir os dados desejados diretamente (BUFORD et al., 2009).

A Figura 2.6c apresenta a topologia superponto, decorrente da fusão das topologias anteriores. Peers de maior capacidade computacional atuam como servidores centrais em determinadas áreas da rede, da mesma forma que ocorre com os servidores centrais da topologia híbrida. Em contrapartida, os demais elementos comportam-se como elementos das topologias puras, fazendo dentro daquele segmento de rede peer-to-peer pesquisas via *flooding* a todos os integrantes do seguimento (PINHEIRO, 2006).

2.4.3 Roteamento

O encaminhamento de mensagens e a busca do peer responsável por determinado identificador de dados são tarefas desempenhadas por algoritmos de roteamento aplicados a todos os *hosts* que integram a estrutura *overlay*. No contexto das redes peer-to-peer,

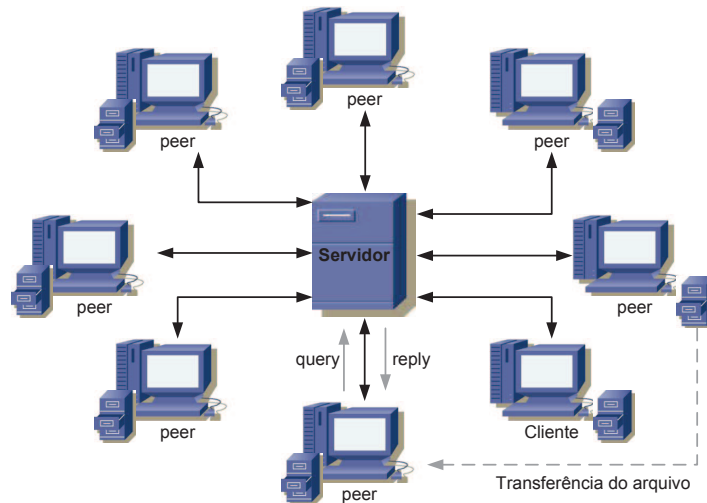


Figura 2.7: Roteamento Centralizado

essa função decorre de maior complexidade em comparação aos algoritmos tradicionais de roteamento (OSPF, RIP, EIGRP) (TANENBAUM, 2003), pois não existe estrutura fixa de nós, e sua topologia é de grande escala e altamente distribuída.

Da mesma maneira dos algoritmos tradicionais empregados no roteamento de pacotes IP, estes também objetivam encaminhar as mensagens de forma otimizada, buscando o caminho mais curto e de melhor desempenho entre origem e destino através de peers intermediários que integram a rede *overlay*. Nessa estrutura descentralizada, os algoritmos são executados em cada peer; cabe a ele escolher dentre outros peers, nos quais este possui conexão, o repasse das mensagens que estão transitando na rede. Atualmente, existem quatro algoritmos principais de roteamento, classificados da seguinte forma:

2.4.3.1 Centralizado

Também conhecido como modelo híbrido, é a abordagem em que o roteamento de mensagens é inexistente. Isso se deve a sua arquitetura, na qual cada peer possui uma conexão com determinado servidor central, atuando com repositório de índices. Esse servidor armazena a relação de arquivos que cada participante deseja compartilhar - isso ocorre no momento de ingresso do peer ou após atualização na base de arquivos compartilhados dos peers. Conforme ilustra a Figura 2.7, com essa base de índices montada, clientes enviam ao servidor solicitações de arquivo, caso exista o referido arquivo o cliente recebe como resposta a indicação do peer que o armazena. Na ocorrência de diversas fontes para o conteúdo solicitado, o servidor informa ao cliente o peer mais adequado, baseado em quesitos como disponibilidade e processamento.

Dos vários sistemas P2P existentes, o Napster adotou esse modelo tendo como resultado uma plataforma eficiente e robusta. No entanto, é necessário enfatizar que a utilização de

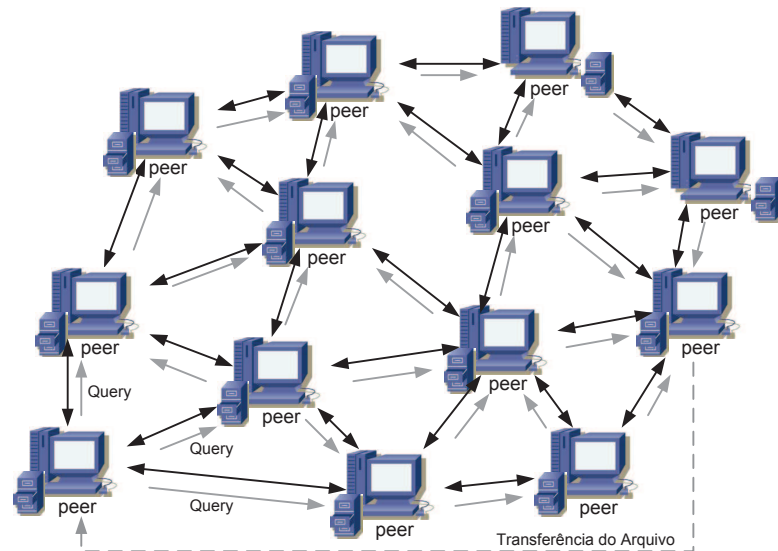


Figura 2.8: Roteamento por Inundação

servidor centralizado para o gerenciamento desses recursos pode acarretar em limites quanto à escalabilidade desse modelo, pois requer servidores de maior capacidade à medida que o volume de requisições e índices a gerenciar aumentam (ANDROUTSELLIS-THEOTOKIS; SPINELLIS, 2004).

2.4.3.2 Inundação

Diferentemente do modelo centralizado, nesse modelo não existe a função de servidor central de índices. Cada peer tem conhecimento apenas de seus dados locais. Dessa forma, uma requisição é enviada para todos os peers com os quais o requisitante possui uma conexão estabelecida. Esse processo é executado sucessivamente até que o arquivo seja encontrado ou que o número de encaminhamentos alcance o seu respectivo limite. Como ilustrado na Figura 2.8, quando o arquivo solicitado é finalmente encontrado, a transferência ocorre de forma direta entre solicitante e peer detentor do arquivo - logo após este responder à mensagem que deu início ao processo de busca (LUA et al., 2005).

Devido ao volume de mensagens, esse modelo impõe sobrecarga dos enlaces de comunicação, bem como de processamento em cada peer. No entanto, existem técnicas capazes de minimizar essa deficiência por meio de mecanismos de *cache* e utilização de *super-peers*, reduzindo o volume de mensagens na rede e ampliando sua aplicabilidade para ambientes com centenas de milhares de peers. Gnutella é um exemplo de sistema peer-to-peer que adota essa abordagem para encaminhamento de mensagens e busca de conteúdo (BUFORD et al., 2009).

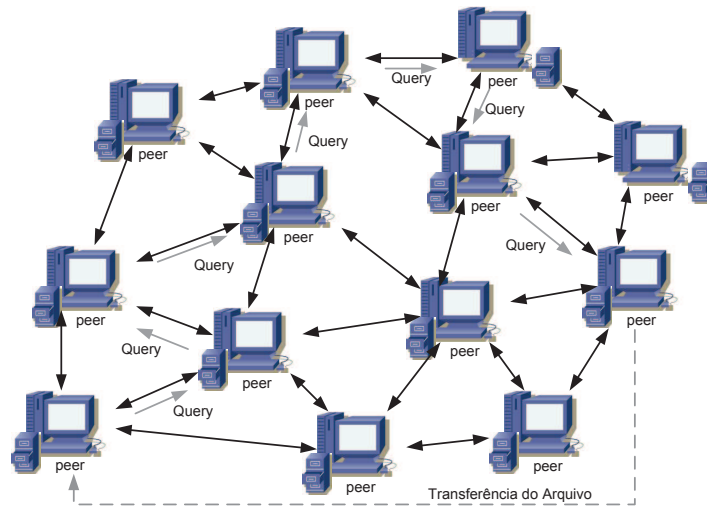


Figura 2.9: Funcionamento do *Random Walk*

2.4.3.3 *Random Walk*

Random Walk é uma alternativa em relação à abordagem de inundação que objetiva reduzir o consumo de recursos na rede causados pela abordagem anterior. Resumidamente, um peer que deseja procurar determinado conteúdo envia um número k de consultas para seus vizinhos, sendo estes selecionados de forma aleatória. Cada pacote de pesquisa possui em sua estrutura um campo do tipo TTL (*time-to-live*) inicializado com valor $T > 0$ para limitar o número de saltos que esse pacote irá percorrer dentro do referido caminho randômico. No momento em que um peer recebe o pacote, este verifica se dispõe do conteúdo solicitado, caso isso se confirme, a transferência ocorre de forma direta com o solicitante conforme ilustrado na Figura 2.9. Caso contrário, o peer intermediário avalia se o campo TTL continua maior que 0, decrementando seu valor em uma unidade e repassando o mesmo a outro vizinho - também escolhido de forma aleatória. Finalmente, se o campo TTL alcançar seu valor limite, esse pacote será descartado, finalizando a busca no referido caminho (BISNIK; ABOUZEID, 2007).

Ao contrário do modelo de inundação, o *overhead* do Random Walk não depende da topologia de rede e dos recursos disponíveis. O desempenho dessa busca e seu respectivo consumo de recursos estão diretamente relacionados com o número de pacotes k enviados e o valor do TTL empregado em sua fase de inicialização. A configuração de valores baixos para k e T , visando a procura de conteúdos de baixa popularidade, proporcionará maiores chances de insucesso nessa tarefa. Por outro lado, existirá maior *overhead* em uma configuração superestimada de k e T na busca de conteúdos com certo grau de disseminação entre os peers. Dessa maneira, é necessário que tais parâmetros sejam definidos de acordo com a popularidade do recurso a ser procurado (BUFORD et al., 2009).

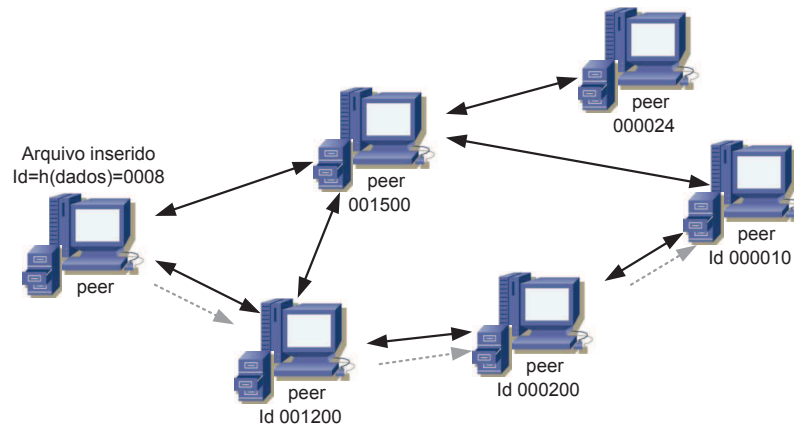


Figura 2.10: Algoritmo DHT

2.4.3.4 DHT (Distributed Hash Table)

É o modelo mais recente para o roteamento de mensagens nas redes *overlay*. Nele, cada peer recebe um identificador único - associado a este de forma aleatória. Quando um dado objeto é disponibilizado na rede, o peer recebe um identificador decorrente da aplicação de uma função *hash* ao nome e conteúdo desse objeto (BUFORD et al., 2009). Após essa definição, o peer no qual o arquivo foi compartilhado encaminha o referido documento ao peer de identificador mais próximo daquele atribuído ao objeto. Nesse processo de encaminhamento, ilustrado na Figura 2.10, todo *host* que participar do roteamento dessas mensagens executará cópia local do conteúdo. As requisições destinadas ao referido identificador também são encaminhadas ao peer de identidade mais próximo ao solicitado, sendo que esse processo se mantém ativo até que uma cópia do dado desejado seja encontrada (MILOJICIC et al., 2002a).

Em relação às demais técnicas de roteamento, os modelos baseados em DHT apresentam menor latência, além de proporcionarem efetivamente o caminho mais curto entre solicitante e dado requerido, tornando-o a melhor alternativa para grandes comunidades peer-to-peer. Contudo, o uso dessa abordagem exige mais controle relativo à localização dos dados, sua consulta apresenta pouca flexibilidade, permitindo apenas consultas simples, e, normalmente, apenas o compartilhamento de arquivos é possível em tais ambientes (PINHEIRO, 2006);

2.5 COMUNICAÇÃO *MULTICAST*

De acordo com os princípios abordados no Capítulo 1, existe atualmente uma onda crescente de aplicações que exigem maior largura de banda, em especial aquelas que apresentam integração com som, animações e vídeo. Entretanto, a função de gerenciar diversos tipos de aplicações para transmiti-las no mesmo meio, tal como o Ethernet (DOWNES et

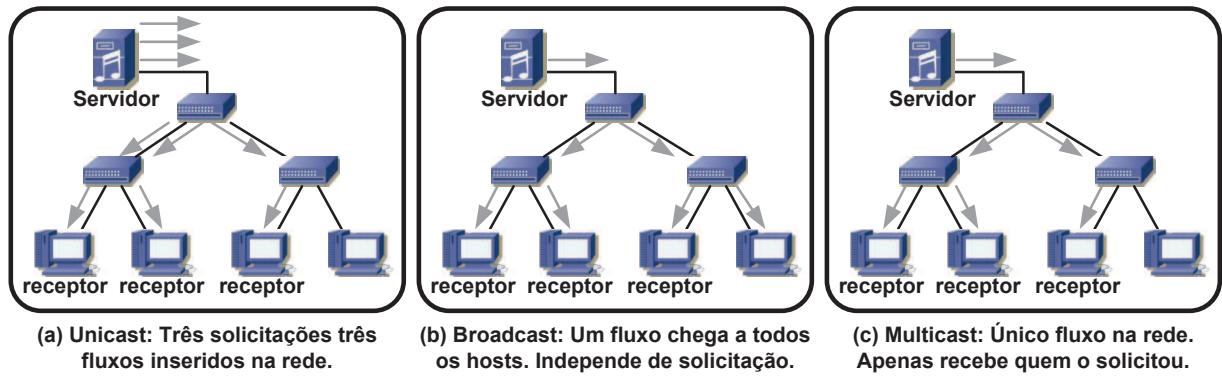


Figura 2.11: Tipos de Tráfego

al., 2000), é um processo de maior complexidade, pois potencializa o uso constantemente elevado da largura de banda da rede. Basicamente, todo esse cenário pode ser tratado na forma de pacotes de comunicação do tipo *unicast*, *broadcast* ou *multicast* (COMER, 2001).

Na transmissão *unicast*, o servidor envia uma cópia de cada pacote para todo endereço de cliente, tratando cada qual de forma independente através de conexões ponto a ponto. Em decorrência desse método de funcionamento, esse modelo possui significantes restrições de escalabilidade, sendo totalmente dependente do número de conexões de usuários, bem como da quantidade de mensagens replicadas na rede (TANENBAUM, 2003). Conforme ilustra a Figura 2.11a, um servidor de vídeo trabalhando nesse modelo de comunicação envia um fluxo em resposta a cada solicitação que recebe.

No modelo *broadcast*, um determinado *host* envia apenas uma cópia de cada pacote usando um endereço desse tipo. Em comparação ao sistema anterior, ele pode proporcionar menor carga na rede se os dados enviados são de interesse de todos os participantes, caso contrário, um *host* que não realizou qualquer solicitação será obrigado a processar esses pacotes, conforme ilustra a Figura 2.11b (DOWNES et al., 2000).

A solução mais eficiente para otimizar a largura de banda é representada pela comunicação *multicast*, na qual uma cópia de cada pacote é remetida para um endereço especial. Diferentemente do *unicast*, um servidor que utiliza o *multicast* envia apenas um fluxo de dados destinado a vários clientes. Diferentemente do *broadcast*, os *hosts* decidem escutar ou não o tráfego proveniente de um endereço *multicast*. Essa tecnologia proporciona economia da largura de banda e controla o tráfego na rede, forçando esta a replicar pacotes somente quando necessário. Conforme pode ser observado na Figura 2.11c, esse modelo de comunicação elimina a redundância de tráfego, propiciando, assim, a redução do processamento da rede e nos *hosts* (COMER, 2001).

Resumidamente, o IP *multicast* é uma transmissão de um pacote de dados IP para um grupo de *hosts* definido por um único endereço IP - também conhecido como endereço de grupo. Essa tecnologia visa reduzir o tráfego de rede distribuindo simultaneamente

Tabela 2.1: *Endereços Classe D conhecidos*

Endereço	Propósito
224.0.0.1	Todos os hosts em uma sub-rede
224.0.0.2	Todos os roteadores em uma sub-rede
224.0.0.4	Roteadores DVMRP
224.0.0.5	Roteadores OSPF
224.0.0.9	Roteadores RIP2
224.0.0.13	Roteadores PIM

uma única cópia de dados para múltiplos recipientes. É uma extensão do protocolo IP (DEERING, 1989), tendo as seguintes características:

- Facilita a transmissão de um datagrama IP para um grupo de *host*, englobando zero ou mais *hosts* identificados por um único endereço IP de destino;
- Entrega um datagrama para todos os membros do grupo com a mesma confiabilidade de melhor esforço presente nos datagramas *unicast*;
- Possibilita associação dinâmica de *hosts* a um determinado grupo;
- Não leva em consideração a localização ou número de membros do grupo;
- Possibilita associação de um *host* a vários grupos simultaneamente;
- Sustenta múltiplos fluxos de dados destinados ao mesmo endereço de grupo;
- Suporta múltiplas aplicações em um mesmo endereço de grupo.

Relativo à faixa de endereçamento destinado a esse modelo de comunicação, o *multicast* emprega o escopo de endereços da classe D. O endereço dessa classe consiste em 4 bits de alta ordem configurados em 1110, seguidos por 28 bits variáveis, de forma a gerar um endereço de grupo dentro da faixa de 224.0.0.0 até 239.255.255.255 (DOWNES et al., 2000).

Além do escopo de endereços, são necessários determinados algoritmos para controle da entrada e saída de membros do grupo, bem como a propagação das informações de grupo para redes que estão em roteadores distintos. No ambiente local, esse controle dos membros é coordenado pelo protocolo IGMP através de mensagens de ingresso e saída do grupo. Para a propagação das informações de grupo entre roteadores distintos, são utilizados protocolos específicos destinados ao roteamento do tráfego *multicast*, como, por exemplo, PIM (ESTRIN et al., 1997), DVMRP (WAITZMAN et al., 1988) e MOSPF (MOY, 1994).

3 TRABALHOS RELACIONADOS

O capítulo anterior apresentou os principais tópicos relacionados a comunicação peer-to-peer e tráfego *multicast*, descrevendo as classificações existentes, bem como características de cada tecnologia. Os requisitos mínimos e as dificuldades encontradas para o desenvolvimento de aplicações em conformidade com a arquitetura P2P também foram abordados nesse levantamento de conceitos básicos.

Visando explorar as funcionalidades e demais vantagens da comunicação P2P, diversas propostas encontram-se em desenvolvimento, cada qual com seu escopo de atuação e devidas restrições.

Este capítulo tem como objetivo apresentar uma visão de projetos já consolidados e de outras pesquisas recentes relacionadas ao tema. Este levantamento prioriza protocolos voltados para o compartilhamento de recursos em redes peer-to-peer, descrevendo sua aplicabilidade, funcionamento e principais características.

3.1 PROTOCOLOS CONSOLIDADOS NA INTERNET

As constantes pesquisas que referenciam ou tratam diretamente a arquitetura reverte-ram em diversas plataformas e modelos de comunicação, que em muitos casos alcançaram consolidação no meio acadêmico e na Internet. Esta seção visa abordar tais plataformas em virtude da efetiva contribuição que desempenharam nos últimos anos.

3.1.1 Napster

O Napster (MILOJICIC et al., 2002b) faz uso de um servidor central (o servidor de diretório) que mantém uma base de dados com informações dos clientes Napster (como, por exemplo, o nome do usuário e a lista de arquivos que ele disponibiliza). Na primeira vez que o *software* do cliente é executado, ele se registra na rede, sendo então incluído no banco de dados do servidor de diretório.

Quando um usuário faz uma busca, é enviada uma requisição ao servidor de diretório, que retorna uma listagem com todos os outros peers que estão conectados e que possuem o arquivo desejado. O usuário pode, dessa forma, estabelecer uma comunicação direta com algum dos participantes que disponibilizam o conteúdo. Comparando o Napster ao objeto desta pesquisa, o LP2P não emprega *hosts* centrais para armazenamento dos registros e controle de requisições, tendo em vista sua arquitetura descentralizada.

3.1.2 Gnutella

O Gnutella (RIPEANU et al., 2002) é um sistema descentralizado para compartilhamento de arquivos e foi criado por um grupo de desenvolvedores da Nullsoft. Diferentemente do Napster, o Gnutella não utiliza um servidor central para o rastreamento dos conteúdos disponibilizados pelos usuários. Para compartilhar arquivos, um peer A (que se comporta como um *servent*) se conecta a outro computador B na rede e avisa que está buscando um determinado arquivo. O peer B comunica então a todos os participantes com os quais ele está conectado sobre a busca que A está realizando. O alcance dessa rede é limitado por um (TTL - *time-to-live*), indicando o número de peers que o pedido atingirá. A maioria dos *servents* Gnutella rejeitam qualquer mensagem que possuir um TTL muito alto. Uma vez anunciado para diversos participantes da rede, A pode procurar pelo arquivo desejado junto aos outros peers. O pedido será enviado a todos os participantes da rede, começando com B, depois C, D... Z, que enviarão para os computadores que eles são conectados e assim por diante. Se um computador (por exemplo, F) possui o arquivo buscado por A, ele envia as informações (como nome e tamanho do arquivo) de volta, percorrendo todo o caminho feito por A, de forma que este possa se conectar diretamente com F para realizar o *download*.

Em relação ao LP2P, existe similaridade no processo de inundação realizado pelo Gnutella na busca do conteúdo. Entretanto, a plataforma P2P local emprega *multicast* para obter os registros de arquivos, e não utiliza *super-peers* na gestão de conexões e divulgação de informações.

3.1.3 BitTorrent

Criado em 2003, o BitTorrent (IZAL et al.,) é um protocolo P2P bastante utilizado na Internet. Nessa arquitetura, os arquivos são divididos em pequenas porções, chamadas de peças, que são trocadas entre os participantes da rede. A disponibilização de conteúdos é feita por meio de arquivos de metadados ou *torrents* obtidos através de portais específicos. Esses arquivos contêm informações como o *hash* e o nome do rastreador, que provê o serviço de descoberta de conteúdos e permite a interação entre os peers. De posse do *torrent*, o cliente BitTorrent se comunica com o rastreador informado nesse arquivo e envia os dados necessários para que os peers possam iniciar o processo de compartilhamento. O cliente dessa arquitetura se comporta como um "semeador", permitindo assim que outros peers se conectem a ele e realizem *downloads*. À medida que os utilizadores recebem partes de um arquivo, eles podem se tornar novos semeadores desse conteúdo.

3.1.4 Kademia

A rede Kademia (HEIDELBERG, 2002) apresenta uma arquitetura totalmente descentralizada, na qual as informações atribuídas aos participantes se disseminam automaticamente à medida que buscas por chaves são realizadas. A infraestrutura de roteamento faz uso de um mecanismo baseado em uma métrica de distância entre identificadores de peers. A topologia pressupõe que todas as mensagens trocadas carregam informações úteis para contato entre os peers. O sistema faz uso dessa informação para enviar mensagens de busca assíncronas e paralelas, a fim de evitar atrasos e *timeouts* decorrentes da presença de *hosts* desativados. Diversas aplicações de P2P utilizam o algoritmo Kademia, entre elas o eDonkey, o eMule e o BitTorrent, que emprega Kademia para permitir o uso de *torrents* sem um *tracker*. Em virtude de empregar conectividade apenas em rede local, o LP2P difere do Kademia ao não utilizar DHTs.

3.1.5 Pastry

Pastry é um protocolo de encaminhamento e formação de redes *overlay*, que implementa tabela *hash* distribuída. Nessa estrutura, cada *host* recebe de forma randômica um identificador de 128 *bits*, obtido com o auxílio de uma função *hash* aplicada ao endereço IP do peer. Essa identificação, denominada *nodeId* é utilizada para indicar o posicionamento do *host* em relação ao espaço circular de endereçamento de chaves com valores de 0 a $2^{128} - 1$. Para o propósito de roteamento, o *nodeId* e as chaves são pensados como uma sequência de dígitos com base 2^b , onde b é parâmetro de configuração normalmente definido com o valor 4. O Pastry roteia as mensagens para o *host* cujo *nodeId* está numericamente mais próximo da chave pesquisada (TAKESHITA et al., 2008).

Devido a sua natureza redundante e descentralizada, não existe um ponto único de falha, e qualquer *host* pode deixar a rede sem aviso prévio com poucas chances de perda de dados. O protocolo também é capaz de usar uma métrica de roteamento fornecida por um programa externo, como *ping* e *traceroute*, para determinar as melhores rotas para armazenar em sua tabela de roteamento. A diferença principal deste modelo frente a proposta P2P local, é o roteamento baseado em tabelas *hash*. Como pontos de similaridade estão a descentralização e alta disponibilidade objetivada nos dois casos.

3.1.6 Tapestry

Tapestry (BUFORD et al., 2009) é uma infraestrutura extensível que fornece a localização do objeto e encaminhamento, descentralizada, com foco na eficiência e baixa latência das mensagens. Sua estrutura *overlay* é muito similar ao Pastry, ambas são baseadas no algoritmo PRR (*Plaxton, Rajaraman, and Richa*) e compartilham dos mesmos

métodos de entrada e saída. Além disso, o processo de atribuição de identificadores na execução do roteamento e armazenamento de cada conteúdo é idêntico nos dois casos. A inexistência de vizinhos e *hosts*-folha representam os fatores que diferenciam o protocolo Tapestry de Pastry.

A baixa latência e eficiência proporcionadas por este protocolo são obtidas graças à construção de tabelas de roteamento otimizadas de forma a reduzir o encaminhamento entre solicitante e dado pesquisado. Junto dessas funcionalidades, Tapestry permite a distribuição dos objetos de acordo com as necessidades de uma determinada aplicação, e permite que aplicativos implementem *multicast* na rede *overlay*. De seu rol de características, foram incorporados ao modelo proposto os quesitos de baixa latência e eficiência, somado ao uso de *multicast* em determinadas situações.

3.1.7 Ivy

O Ivy (MUTHITACHAROEN et al., 2002) é um sistema de arquivos peer-to-peer que suporta leitura e escrita multi usuário. Este sistema de arquivos possui arquitetura totalmente descentralizada e dispõe uma imagem única do sistema de arquivos, tal como o NFS, não necessitando porém de um servidor dedicado. Os dados e metadados são armazenados no sistema peer-to-peer de armazenamento por blocos DHash (DABEK et al., 2001), que distribui e replica blocos com o objetivo de prover disponibilidade.

O Ivy faz uso de logs individuais para cada usuário com o objetivo de assegurar a consistência dos metadados com a existência de múltiplos usuários com permissão de escrita e para que o sistema possa conviver com a falta de confiança entre os seus usuários. No Ivy, um log consiste de uma lista encadeada de registros imutáveis que contém informações relacionadas a uma única alteração no sistema de arquivos.

Cada usuário com acesso de escrita mantém um log das suas modificações. Para ordenar os registros, o Ivy utiliza os vetores de versão usados contidos nos logs dos usuários. Um usuário pode escolher entre os outros usuários aqueles que lhe são confiáveis e devem ter portanto os seus logs considerados. A proteção contra ataques de não-usuários do Ivy é conseguida pela verificação criptográfica dos dados retornados via DHash.

3.1.8 Squirrel

O Squirrel (IYER et al., 2002) é um sistema peer-to-peer totalmente descentralizado cujo objetivo principal é permitir que os navegadores web em computadores *desktop* possam compartilhar as caches locais para formar um *web cache* eficiente e escalável, sem a necessidade de hardware dedicado e sem os custos administrativos associados, tendo como ambiente alvo as redes corporativas.

Este sistema utiliza o algoritmo Pastry para localização de objetos a fim de identificar e criar rotas para os nodos que possuem cópias de cache de um objeto requerido. Devido ao fato de o Pastry ser escalável, auto-organizável e resistente a falhas de nodos concorrentes, o Squirrel também é. Na ocorrência de falhas em múltiplos componentes da rede, o Squirrel apenas necessita buscar novamente uma pequena fração dos objetos em cache do navegador web de origem.

Conforme os autores, o Squirrel apresenta desempenho comparável a um *web cache* centralizado em termos de uso da largura de banda e latência, apresentando baixa sobrecarga nos nodos participantes.

3.1.9 OceanStore

OceanStore, um sistema de armazenamento distribuído em escala global, foi a primeira tentativa P2P de agregar *hosts* no papel de *super-peers*. OceanStore pode ser visto como um antecessor dos sistemas P2P, projetado para executar em um grande número de *hosts* distribuídos em todo o mundo e ser mantido por múltiplos participantes independentes. Em seu funcionamento, cada peer é classificado como confiável e não confiável, da mesma forma que ocorre com outras plataformas P2P (SACHA, 2009).

Para garantir a confiabilidade e desempenho, os dados armazenados são replicados e distribuídos uniformemente entre os *hosts* usando um algoritmo de replicação proativa. Cada objeto cadastrado tem uma réplica de dados primária, que ordena as versões, verifica credenciais de controle de acesso, e constrói uma árvore de difusão entre as réplicas secundárias. As réplicas primárias são hospedadas em um conjunto de *hosts* selecionados, denominados como anel interno. Esses *hosts* são comparados aos *super-peers* utilizados por outros sistemas P2P.

A principal camada da arquitetura OceanStore consiste em um pequeno número de réplicas localizadas em regiões da rede com largura de banda e conectividade elevada. No entanto, não está definido qualquer algoritmo para a eleição de *hosts* que integram o anel interno. De acordo com a especificação do protocolo de replicação, os *hosts* que formam o anel interno são selecionados manualmente pelos operadores do sistema.

A contribuição do OceanStore ao LP2P está na funcionalidade de replicação proativa. Esse recurso foi adaptado ao contexto do LP2P, descartando o uso de *super-peers* e agregando quesitos que consideram o interesse do usuário e a disseminação do conteúdo na rede.

3.1.10 xFS

o xFS (ANDERSON et al., 1996) é um sistema de arquivos em rede que não utiliza servidor central. Seu armazenamento em rede explora o desempenho e disponibilidade do RAID (*Redundant Arrays of Inexpensive Disks*) e faz uso de sistema de arquivos estruturado em logs para organizar o armazenamento. Para distribuir o controle em toda a rede o xFS faz uso de técnicas inspiradas naquelas utilizadas para prover consistência de cache com uso multiprocessadores.

De acordo com os autores, a co-localização da gerência de um arquivo com o cliente que cria esse arquivo, quando comparada com o emprego de um gerente centralizado, pode reduzir significativamente os problemas relacionados a localidade devido a redução do número de saltos de rede necessários para atender às solicitações do cliente.

No que diz respeito a segurança, o xFS mostra-se mais apropriado para ambientes restritos, no qual as máquinas que se comunicam através de uma rede rápida e onde a confiança entre os nodos pode prover certa segurança. Assim como outros sistemas de arquivos, o xFS confia no kernel para impor um firewall entre os processos de usuários não confiáveis e sistemas como o xFS. Os nodos podem ainda reforçar a segurança do sistema de arquivos se, por exemplo, os servidores armazenarem somente fragmentos fornecidos por clientes autorizados.

3.2 PESQUISAS RECENTES

3.2.1 GnutellaBT

Segundo Luo et al. (2009), o compartilhamento de arquivos é uma aplicação importante na Internet, reforçando a gama de aplicações P2P destinadas a esse propósito. Destas, a plataforma BitTorrent apresenta melhor eficiência, mas seu funcionamento exige a publicação de um arquivo do tipo *torrent* na Internet, dificultando o uso desse protocolo. Para solucionar esse problema, os autores propõem o uso da estrutura Gnutella como base para formar uma rede de *trackers*¹ dedicada, em que a publicação de arquivos *torrent* pela *web* é substituída pelo recurso de localização da rede Gnutella.

A escolha de vizinhos é executada por meio do algoritmo Random Walk, sobre uma rede BitTorrent específica. Esse sistema denominado GnutellaBT não requer o suporte de servidores *web* ou peers dedicados à função de *trackers*, sendo essas as maiores contribuições desse estudo para o compartilhamento de arquivos em redes peer-to-peer.

A arquitetura do GnutellaBT, é uma estrutura de duas camadas. A camada inferior

¹Servidor que auxilia na comunicação entre dois computadores que utilizam o protocolo BitTorrent

utiliza a rede Gnutella, enquanto a rede *torrent* localiza-se na camada superior onde estão peers relacionados com arquivos *torrent* específicos.

Os peers em uma rede GnutellaBT exercem três papéis distintos. O primeiro refere-se ao de consulta, em que o *host* suporta essa operação de forma colaborativa. A segunda função é de *seed*, na qual o *host* serve tanto para atender consultas quanto para o armazenamento de *chunks*². A última função é de *leech*, que atende consultas e armazena partes de *chunks*. Na plataforma GnutellaBT esses papéis são idênticos.

Este sistema insere sua parcela de contribuição ao projeto LP2P através do processo de consulta colaborativa dos dados. A função de *leech* também é inserida na sistema P2P local, pois objetiva que cada *host* atue efetivamente no papel de servidor e cliente dos recursos compartilhados.

3.2.2 *JBPeer*

O compartilhamento de arquivos utilizando métodos baratos de transmissão e recepção de dados foram os motivadores do estudo realizado por McNamara e Yang (2008). Esse trabalho culminou no desenvolvimento da arquitetura JBPeer (Java Bluetooth Peer-to-Peer), sistema de compartilhamento de arquivos peer-to-peer para redes móveis. De forma análoga ao LP2P, as informações de controle são transmitidas utilizando a rede GSM ao invés da tecnologia *multicast*, enquanto a troca de arquivos emprega a tecnologia Bluetooth no lugar da transmissão *unicast*.

3.2.3 *A-Kad*

O trabalho apresentado por Ni et al. (2009) propõe um protocolo peer-to-peer anônimo com base na rede Kad³, através do estabelecimento de dois canais anônimos onde são realizados processos de recuperação e publicação de arquivos, atingindo o anonimato mútuo na troca de arquivos na estrutura P2P.

O protocolo busca fornecer uma camada de comunicação que atenda os princípios do anonimato entre os envolvidos na troca de arquivos sobre a estrutura Kad. Com esse sistema, as identidades dos usuários serão bem-protegidas, não importando se eles são prestadores de arquivo ou solicitantes. Visando o anonimato, os autores consideram quatro aspectos relativos ao objetivo principal da rede Kad.

1. O servidor de arquivo pode anonimamente publicar as suas informações de arquivos sem se preocupar que sua identidade será revelada;

²Em redes BitTorrent, um *chunk* representa partes de um determinado arquivo compartilhado

³A rede Kad é uma rede P2P na qual implementa o protocolo *overlay* Kademlia

2. O solicitante pode enviar seus arquivos de consulta de forma anônima, sem se preocupar que seu conteúdo de consulta será exposto;
3. O fornecedor de segurança de arquivo pode transferir arquivos e sua privacidade estará bem-protegida;
4. O solicitante de arquivo também pode seguramente recuperar os arquivos que quiser, e ninguém pode conhecer o conteúdo do arquivo específico.

Para atingir os objetivos acima, foram adotadas três abordagens distintas com base no Kad original, estabelecendo dois canais de comunicação anônimos para fornecimento de arquivos, criptografia e metadados, aplicando uma função *hash* à chave de pesquisa do arquivo. Para que este conjunto de premissas e abordagens seja executado de acordo, os autores estabeleceram 4 fases de funcionamento: *publicação dos metadados*, *pesquisa*, *transferência de arquivo* e *estabelecimento de canal seguro entre transmissor e receptor*.

Com a implementação das premissas propostas, foi possível fornecer o anonimato desejado em dois canais de comunicação para o prestador de arquivos. Sendo assim, manteve-se a eficiência e a preservação da privacidade no decorrer da busca de arquivos, que passou a empregar o *hash* de consulta ao invés de uma palavra-chave. Adicionalmente, esse estudo propôs uma nova abordagem para avaliação do grau de anonimato.

3.2.4 *ADC*

Aliado à pesquisa anterior, o protocolo ADC (Advanced Direct Connect) (MOLIN, 2009) está relacionado ao presente estudo ao permitir conectividade local de rede. ADC oferece compartilhamento de arquivos utilizando estrutura P2P híbrida composta por clientes que conectam-se a um *hub* central. Esse dispositivo é responsável por fornecer e atualizar informações quanto à localização dos recursos e integrantes da rede, juntamente com um serviço de comunicação instantânea entre peers. A pesquisa por determinado conteúdo é realizada sempre com o auxílio do *hub* central, enquanto a transferência de dados ocorre de forma direta entre solicitante e dono do recurso pesquisado. Sua similaridade ao LP2P está na transferência de dados de forma direta, no entanto, o protocolo ADC faz uso da centralização para coordenar a entrada e saída dos peers, bem como para auxiliar no processo de pesquisa e organização da relação de conteúdos compartilhados, funções totalmente descentralizadas no modelo LP2P.

3.2.5 *OneHop DHT*

O protocolo OneHop DHT (Distributed Hash Table) (WANG et al., 2009) é aplicado em sistemas que necessitam de rápida convergência (ex.: sinalização HLR/HSS utilizada

Tabela 3.1: Comparação entre sistemas pesquisados e LP2P

Sistema P2P Característica	Organização	Descentralização	Arquitetura	Protocolo de pesquisa	Peers Entrada / Saída	Aplicação	Abrangência
Napster	Não estruturada	Servidores dedicados	Centralizada	Pesquisa de índices	Constante	Compartilhamento de arquivos	Internet
Gnutella	Não estruturada	Topologia plana com <i>peers</i> iguais	Rede plana e ad-hoc. Busca por inundação e transferência direta	Consulta por inundação.	Constante	Compartilhamento de arquivos	Internet
BitTorrent	Não estruturada	Modelo centralizado. Tracker mantém controle dos <i>peers</i> .	Peers solicitam informações de um Tracker central.	Tracker.	Constante	Compartilhamento de arquivos	Internet
OceanStore	Estruturada	Super Peers	Tabela hash distribuída	Tapestry	Constante	Armazenamento Distribuído	Internet
Tapestry	Estruturada	Funcionalidade DHT em escala da Internet	Rede mesh global <i>Plaxton-style</i> .	Sufixo do NodeID.	$\log_2 N$	infraestrutura P2P	Internet
Pastry	Estruturada	Funcionalidade DHT em escala da Internet	Rede mesh global <i>Plaxton-style</i> .	Chave e sufixo do NodeID	$\log_2 N$	infraestrutura P2P	Internet
Kademlia	Estruturada	Funcionalidade DHT em escala da Internet	Métrica XOR para distância entre pontos no espaço de chaves.	Roteamento baseado na chave e NodeID.	$\log_2 N + c$ onde $c =$ constante de baixo valor.	infraestrutura P2P	Internet
GnuTellaBT	Não estruturada	Super Peers	Arquitetura de dois níveis. Gnutella / GnutellaBT	Consulta por inundação.	Não definido	Compartilhamento de arquivos	Internet
JBPeer	Não estruturada	Servidores dedicados	Centralizada	Lista de recursos	Constante	Compartilhamento de arquivos	MAN / GSM / Bluetooth
A-Kad	Estruturada	Funcionalidade DHT em escala da Internet	Kademlia	Kademlia	Não definido	Comunicação anônima	Internet
ADC	Não estruturada	Super Peers	Formada por Hubs e clientes em modo passivo e ativo	Realizada através do <i>peers</i> na função de Hubs	Constante	Compartilhamento de arquivos. Mensagem Instantânea	LAN / Internet

em telecomunicações). Em redes P2P esse mecanismo proporciona desempenho na busca de conteúdo, além de propagar de forma eficiente as mudanças de estado na relação entre *peers* e dados disponíveis. Esse recurso auxilia o compartilhamento de arquivos, mantendo em cada *peer* uma tabela completa de roteamento - proporcionando a transferência de arquivos de forma direta. Essa rede é definida como uma estrutura P2P híbrida, na qual *peers* gerenciadores auxiliam no processo de entrada de novos integrantes, bem como na atualização e consistência das DHTs. Em comparação ao LP2P, ambos objetivam propagar a base de conhecimento da rede em todos os participantes. Contudo, o LP2P realiza essa função sem o uso tabelas *hash*, pois adota uma bordagem diferenciada, mesclando inundação com transmissão *multicast*.

3.3 AVALIAÇÃO DE PROTOCOLOS

Ao longo deste capítulo, foram revisadas diversas propostas para o uso da arquitetura P2P, das mais variadas formas. É notória a gama de funcionalidades providas por esse paradigma de conexões, bem como os benefícios que ele representa.

Contudo, no decorrer deste estudo, foram observadas poucas ações no sentido de levar o peer-to-peer a ambientes com maior controle e estabilidade na permanência dos *hosts*. Esse fato pode estar relacionado ao modelo inicial P2P - totalmente descentralizado, distribuído e sem limites para a formação de suas comunidades. Os esforços de pesquisa

focam principalmente em aplicações voltadas para redes *wireless*, jogos e multimídia. Entretanto, o escopo de aplicações que o peer-to-peer pode proporcionar vai muito além desse nicho de funcionalidades.

Tais fatores não impedem que a arquitetura P2P seja agregada ao cenário das redes locais. Da mesma maneira que aplicações multimídia se inserem ao escopo da comunicação P2P, é possível unificar os conceitos do peer-to-peer para compartilhar recursos em ambientes de maior estabilidade. Em consequência disso, sistemas de *backup* e armazenamento distribuído seriam projetados sob uma abordagem P2P com menor custo de implantação e gerenciamento.

Desta forma, os aspectos avaliados no andamento deste capítulo impulsionaram o surgimento da plataforma LP2P, influenciada por várias particularidades de cada protocolo pesquisado. Sua principal contribuição está na formação de unidades de armazenamento distribuídas mas representadas ao usuário de forma unificada. Neste sentido o LP2P se difere das demais soluções ao não empregar mecanismos complexos para gerenciamento de índices da rede (Ex. DHT). A Tabela 3.1 resume esta avaliação, relacionando alguns dos pontos inseridos ao projeto LP2P como resultado dessas pesquisas.

Sendo assim, este trabalho apresenta o LP2P como nova vertente de estudos que objetiva aplicar essa tecnologia para distribuição de conteúdo em redes locais. Os capítulos subsequentes tratam respectivamente da definição, desenvolvimento e avaliação deste modelo.

4 LOCAL PEER-TO-PEER PROTOCOL

O LP2P destina-se ao compartilhamento de arquivos entre equipamentos conectados à infraestrutura de rede local. Objetiva criar e gerenciar áreas de armazenamento distribuído em termos de aplicação, mas centralizado na visão do usuário. Esse sistema visa aproveitar de forma eficiente o espaço de armazenamento ocioso e reduzir o consumo de banda na rede, resultado da comunicação cliente/servidor. Junto com essas vantagens, é imperativo observar que esse serviço não aumenta os custos de gestão e contribui para a redução do investimento em *hardware* especializado para o armazenamento de dados em ambientes corporativos.

Seu desenvolvimento tem o propósito de agregar as vantagens do paradigma P2P às condições favoráveis de transmissão e recepção de dados disponíveis nas redes locais. Dentre elas, ressalta-se a estabilidade da infraestrutura LAN, oferecendo maior tempo de conectividade para cada participante. Além disso, fatores como *throughput*, baixa latência, homogeneidade de conexões e mecanismos de redundância possibilitam o desenvolvimento de aplicações P2P voltadas ao compartilhamento de arquivos em redes locais.

Consequentemente, o conjunto de fatores ilustrados no decorrer deste trabalho fomentou o desenvolvimento do LP2P, que visa oferecer repositórios de dados distribuídos. Dentre suas vantagens estão o aproveitamento do armazenamento inativo em cada estação de trabalho e o controle otimizado das réplicas de cada conteúdo. As seções subsequentes apresentam esse sistema, abordando seu funcionamento, arquitetura, relação entre módulos, conjunto de mensagens e demais aspectos que o compõem.

4.1 VISÃO GERAL

Normalmente, em ambientes corporativos, ocorre o baixo aproveitamento dos recursos de armazenamento presentes nas estações de trabalho. Aliado a isso, o custo para aquisição e manutenção de *hardware* específico, destinado ao armazenamento centralizado, é elevado. Tais deficiências impulsionaram o Grupo de Pesquisa em Redes de Computadores do PIPCA no sentido de pesquisar e sugerir soluções viáveis para o tratamento desses problemas.

Essa demanda deu origem a três diferentes frentes de pesquisas, que somadas apresentam-se como uma solução integrada para o cenário mencionado anteriormente. A primeira objetiva implementar um módulo genérico de descobertas de serviços para detectar os recursos disponíveis, minimizando eventual *overhead*. A segunda demanda de estudo tem como finalidade categorizar usuários e aplicar mecanismos de recompensa e confiança, técnicas muito utilizadas em sistemas peer-to-peer já consolidados na Internet.

O tema central deste trabalho representa o terceiro objeto de pesquisa, envolvendo a formulação, bem como o desenvolvimento e a avaliação do *Local Peer-to-Peer-Protocol* (LP2P).

Basicamente, a proposta do LP2P gerencia a comunicação e a troca de dados entre peers, empregando uma abordagem de conexões que descarta o uso de servidores centralizados. Obedecendo aos princípios do modelo peer-to-peer de gestão compartilhada, tal arquitetura conta com o envolvimento de todos os participantes da rede para formação e manutenção dos indexadores de conteúdo. Com essas características, o conjunto de peers proporciona o devido armazenamento distribuído dos conteúdos cadastrados na rede.

Enfatiza-se que são propriedades desejáveis desse protocolo a eficiência e o baixo *overhead*. Com esse objetivo, o projeto conta com um conjunto de mensagens categorizadas como *controle*, que emprega em sua maioria mensagens do tipo *multicast*, e *transferência de dados*, que utiliza mensagens na forma de *unicast*. A diferenciação no método de envio desses pacotes, além de melhorar o aproveitamento da largura de banda, também auxilia nas questões de escalabilidade do modelo. Contudo, antes da devida abordagem do modelo, se faz necessária a reflexão quanto à inserção dos principais conceitos P2P em redes locais. Na seção a seguir, tais ponderações são apresentadas e relacionadas ao LP2P.

4.2 MESCLANDO P2P E LANS

O peer-to-peer é uma abordagem robusta de conectividade, pois agrega diversas funcionalidades que juntas favorecem sua utilização nos mais variados cenários (ex.: processamento distribuído, trabalho colaborativo etc.). Mesclar suas propriedades com os recursos oferecidos pelas redes locais requer uma análise cuidadosa desses aspectos. Esta seção revisa e avalia a aplicabilidade de cada recurso dentro do ambiente LAN. Assim, o texto a seguir retoma algumas peculiaridades da comunicação peer-to-peer, investigando em cada caso a possibilidade de sua integração ao protocolo objeto deste estudo.

De acordo com as pesquisas realizadas, uma das questões que impulsionaram o crescimento da tecnologia P2P é dada pela troca de dados sem a necessidade da identificação do solicitante e proprietário do recurso em questão. O anonimato é de aplicação viável em redes locais que ofereçam acesso público. Entretanto, essa liberdade pode ocasionar algumas deficiências de segurança. Logo, o uso dessa funcionalidade deve ser controlado de acordo com o tipo de recurso compartilhado. Por exemplo, um repositório com acesso restrito não deve oferecer tal funcionalidade. De maneira oposta, pastas com acesso público dispensam o uso de autenticação e favorecem a comunicação de forma anônima.

As constantes mudanças de topologia exercem baixa influência no processo de comunicação nas LANs. O ambiente de acesso local apresenta maior estabilidade, ocasionando

Tabela 4.1: Equivalência de recursos nos ambientes P2P e LAN

	P2P	LAN
Anonimato	Peers não tomam conhecimento da identidade do solicitante e dono do recurso.	Mesmo funcionamento, porém aplicado somente em compartilhamentos públicos.
Auto-Organização	Rede se adapta dinamicamente as entradas e saída de peers.	Não existe adaptação, mas o controle dos peers ativos.
Descentralização	Topologias puras ou híbridas com super-peers ou servidores centrais de índices.	As questões de desempenho da LAN favorecem topologias de comunicação puras.
Desempenho	Obtido pelo particionamento do conteúdo solicitado entre diversos fontes.	Mesmo comportamento das redes P2P.
Multicast	Exige algoritmos específicos de roteamento.	Solução consolidada em redes LAN com rápida implementação.
Tolerância a Falhas	Utiliza replicação de dados.	Além da replicação de dados mecanismos com STP e RSTP auxiliam neste processo.
Roteamento	Principais algoritmos: DHT, Random Walk, Centralizado e Inundação.	Devido ao desempenho e banda disponível o roteamento de informações de controle pode utilizar abordagem do tipo inundação.

menor transição de estado dos *hosts* da rede P2P. Tal característica dispensa o uso de algoritmos que gerenciem o processo de auto-organização dos *hosts*. No entanto, um protocolo que aplique o paradigma P2P no meio local deve realizar tal controle, evitando assim inconsistência na base de informações que relaciona os recursos disponíveis na rede e sua localização correspondente.

O fator de descentralização inserido no modelo de comunicação local pode aproveitar a largura de banda e estabilidade da rede, oferecendo administração descentralizada. Consequentemente, a responsabilidade pelo gerenciamento desse ambiente seria compartilhada entre os membros conectados ao sistema, caracterizando a formação de redes P2P puras.

O desempenho encontrado nas redes peer-to-peer pode ser obtido de forma similar no cenário P2P local. O conteúdo é transferido ao destinatário utilizando uma ou várias fontes - neste caso, cada uma irá transmitir apenas parte do conteúdo solicitado. Esse processo apresenta melhor desempenho em comparação à comunicação P2P tradicional, devido ao método padronizado de acesso e banda disponível na LAN.

A característica de tolerância a falhas da arquitetura peer-to-peer é passível de integração ao sistema de comunicação local. Nesse ambiente, essa funcionalidade é alcançada utilizando os mesmos métodos empregados nas soluções tradicionais. Além disso, os recursos de conectividade nativos desse ambiente ampliam tal capacidade de recuperação da rede, evitando perda de dados na comunicação entre *hosts*.

Os principais algoritmos de roteamento em redes P2P são, respectivamente, o DHT, *Random Walk*, centralizado e inundação. Transpondo esse conjunto de técnicas para o contexto das redes locais, um protocolo projetado para tal finalidade pode adotar uma abordagem do tipo inundação, porque seu comportamento em LANs é equivalente ao de mensagens *unicast*, pelo menos no nível físico. Mesmo nas camadas mais altas, o impacto da técnica de inundação pode ser minimizado pelo uso de *multicast* em vez de *broadcast*.

A transmissão de dados utilizando *multicast* está consolidada nas LANs. Entretanto,

sua aplicabilidade em redes WAN depende de diversos fatores, como protocolos específicos (DVMRP e MOSPF), dificultando o emprego desse método na comunicação P2P. Portanto, é possível unificar *multicast* para prover envio de dados em conformidade com o paradigma P2P em redes locais. Esse recurso poderia ser empregado no envio de informações de controle, aproveitando-se de que o crescimento do tráfego em relação ao número de usuários apresenta comportamento linear, reduzindo de forma significativa o impacto deste protocolo na rede.

Segundo os aspectos abordados no decorrer desta seção, é possível agregar particularidades do peer-to-peer às redes locais. A Tabela 4.1 apresenta resumidamente a equivalência de cada recurso analisado, reforçando que sua respectiva migração deve obedecer às mudanças necessárias para fazer uso do novo ambiente de aplicação. As ponderações resultantes formam a base da proposta apresentada a seguir.

4.3 OPERAÇÃO

Em seu funcionamento, cada *host*, ao ingressar no ambiente LP2P, está apto a compartilhar dados com outros peers da rede. Todo membro divulga aos demais o conjunto de arquivos que está disponibilizando em determinado repositório utilizando troca de mensagens. Esse intercâmbio de informações cria uma base de conhecimento que consiste da relação de compartilhamentos, *hosts* e dados compartilhados, disponíveis para uso de aplicativos externos presentes em cada peer, e comunicando-se localmente com a plataforma LP2P. Assim, esses agentes atuam como clientes LP2P e utilizam as informações propagadas entre peers para fornecer aos usuários locais a lista de arquivos disponíveis em cada repositório.

Na perspectiva do usuário, esses conjuntos de arquivos são apresentados na forma de *uma única pasta compartilhada, entretanto, seu conteúdo é distribuído e cada arquivo pode estar armazenado em diversos hosts*. A Figura 4.1 ilustra essa operação. *Peer1* e *peer4* ingressam no compartilhamento *mapas*, enquanto *peer2* e *peer3* conectam-se ao recurso *pesquisa*. É importante ressaltar que um *host* pode ingressar em diversos compartilhamentos simultaneamente sem qualquer restrição - neste caso, cada um é apresentado ao usuário separadamente.

Um exemplo mais detalhado da operação do modelo que contempla genericamente essa troca de mensagens é ilustrado pelo uso do LP2P no compartilhamento de músicas em rede local. Nesse cenário, assumimos que quatro peers (**A**, **B**, **C** e **D**) estão conectados ao compartilhamento **MIDIAS**, cada um contribuindo com uma série de arquivos MP3. Após a fase de ingresso no sistema, os usuários visualizarão o conjunto mesclado de canções como parte do mesmo compartilhamento de rede. Então, suponhamos que um novo *host*

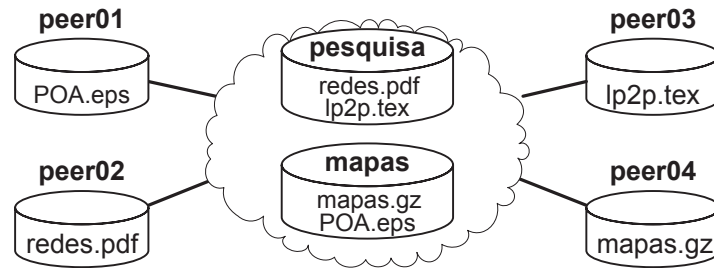


Figura 4.1: Operação básica do LP2P

E decida ingressar ao diretório **MIDIAS**. Para que essa ação obtenha êxito, as seguintes etapas são realizadas:

1. O usuário na máquina **E** inicia a aplicação LP2P (*daemon* associado à interface de usuário);
2. O *daemon* LP2P em execução no *host* realiza o processo de reconhecimento do ambiente, detectando a quais compartilhamentos irá ingressar. Além disso, esta fase também determina fatores como replicação do conteúdo e aspectos de segurança;
3. A partir dessa informação, o LP2P residente no *host* **E** envia pacotes de anúncio ao grupo *multicast* correspondente ao repositório **MIDIAS**, informando que um novo membro deseja ingressar no referido compartilhamento;
4. O pacote de anúncio é recebido por todos os membros já conectados à pasta LP2P, e, de acordo com um fator de probabilidade, apenas um peer responderá a referida mensagem (por exemplo, o *host* **A**), confirmando ao recém-chegado e aos demais membros a entrada de um novo componente. O fator de probabilidade considera todos os nós ativos e visa reduzir o envio de mensagens redundantes;
5. Após receber a confirmação de ingresso, *host* **E** envia novo pacote solicitando a lista de recursos disponíveis;
6. Novamente é aplicado o fator de probabilidade, e apenas um dos integrantes envia a relação de arquivos cadastrados no diretório **MIDIAS**;
7. *Host* **E** recebe a resposta correspondente e fornece esses dados para uma aplicação cliente, que atua como uma interface entre o usuário e LP2P (por exemplo, uma janela do gerenciador de arquivos, ponto de montagem no sistema de arquivos ou interface *web* contendo a relação dos recursos compartilhados);
8. Os dados são apresentados ao usuário como uma única unidade de armazenamento que contém o nome, tamanho e metadados adicionais de cada arquivo;

9. Paralelamente ao passo anterior, com o objetivo de atualizar a base de conhecimento da rede, o *host E* envia sua listagem de arquivos locais aos demais participantes através do mesmo grupo *multicast*. Nessa ação, arquivos são inseridos ao compartilhamento ou uma entrada existente é atualizada para conter a indicação de nova réplica do recurso.

A operação descrita anteriormente tem o objetivo de proporcionar a entrada de novos *hosts* ao sistema, bem como propagar a localização dos recursos compartilhados. Tais tarefas são primordiais no sentido de garantir a troca de arquivos em redes peer-to-peer com gerenciamento descentralizado. O comportamento ilustrado nessa sequência de ações exemplifica claramente uma rede P2P de gestão compartilhada. Tanto a união do *host* ao diretório **MIDIAS** quanto a entrega dos registros contou com a participação dos peers **A**, **B**, **C** e **D**. No entanto, até este ponto, os dados não foram efetivamente transferidos, tarefa que requer a execução das seguintes ações:

1. O usuário do *host E* decide ouvir determinado conteúdo, executando um duplo clique no ícone correspondente. Imediatamente, o cliente (*front-end* do usuário) solicita ao LP2P esse conteúdo;
2. Internamente, o LP2P verifica a localização do arquivo selecionado consultando sua base de conhecimento. Quando o recurso está presente localmente (ou seja, no disco rígido), o mesmo é carregado na memória e acessado diretamente pelo usuário;
3. Caso esteja localizado em repositório remoto, o LP2P determina uma fonte do conteúdo de acordo com as informações contidas em seus registros do compartilhamento. Com a escolha do referido peer, mensagens *unicast* são remetidas a esse *host* solicitando a transferência do conteúdo;
4. O *host* remoto, ao perceber esse pacote, transfere o arquivo ao peer **E**. Caso contrário, remete uma mensagem de erro informando que o objeto em questão não foi encontrado;
5. Quando detectada uma mensagem de erro, o processo de busca do arquivo é retomado e outra fonte selecionada. Se não houver mais fontes disponíveis, uma notificação de erro é repassada ao cliente LP2P que, por sua vez, remete ao usuário mensagem de falha para abrir o arquivo em questão;
6. Caso a transferência seja efetuada com sucesso, o conteúdo é entregue ao cliente LP2P, que fica incumbido da respectiva função de *cache*. Paralelamente, o LP2P controla o número de transferências do arquivo, fazendo sua replicação de acordo com os parâmetros definidos ao compartilhamento.

Podemos observar que o processo acima ocorre com interferência mínima do usuário, que necessita apenas acionar o módulo LP2P e, posteriormente, selecionar o arquivo desejado. Adicionalmente, este modelo de operação permite usabilidade significativa e conveniência de uso, pois não requer conhecimento técnico avançado para usufruir do serviço LP2P. Além disso, a facilidade de comunicação, bem como seu gerenciamento automatizado, não resultam em custos adicionais de apoio especializado para este sistema, impactando na redução de custos com mão de obra por parte das organizações.

Para executar os processos ilustrados anteriormente, o LP2P apresenta uma organização modular responsável pela comunicação entre peers e aplicativos do usuário, utilizando uma abordagem não estruturada de conexões entre os participantes da rede. Nas próximas seções, esses aspectos são explorados e apresentados detalhadamente.

4.4 ARQUITETURA

A definição da arquitetura deste modelo de comunicação é uma etapa fundamental do projeto. A proposta deve apresentar conformidade com as particularidades que englobam as redes locais, bem como o gerenciamento de recursos existentes e a capacidade de armazenamento de cada peer. Para este fim, faz-se necessária a definição dos seguintes aspectos:

- **Pilha de protocolos:** Modulariza essa arquitetura de comunicação visando facilitar seu desenvolvimento e respectiva escalabilidade;
- **Correlação:** Define a integração entre os módulos;
- **Sistemas de mensagens:** Estabelece um padrão de mensagens para uso local e externo do *host*, sendo responsável pela divulgação de informações de controle. Realiza a transferência de dados entre peers, bem como trata da entrega destes às aplicações locais;
- **Base de conhecimento:** Responsável por criar e manter uma relação de registros para cada recurso disponível na rede;
- **Acesso aos recursos LP2P:** Conjunto de regras para integração do LP2P a clientes externos. Um cliente fornece ao usuário uma interface amigável de acesso ao conteúdo compartilhado.

Seguindo os objetivos deste estudo, foi definida a pilha de protocolos ilustrada na Figura 4.2. Basicamente, o LP2P é formado por seu gerenciador central (LP2P-CORE), além do módulo de mensagens (LP2P-MSG) e a base de informações (LP2P-BD). Juntamente a

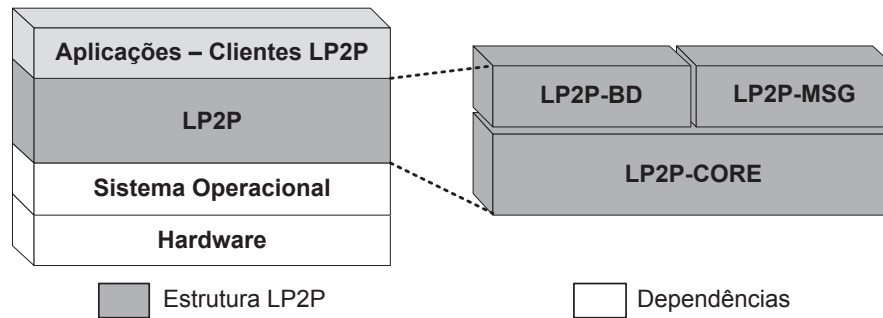


Figura 4.2: Arquitetura LP2P

esses componentes, o modelo oferece interface para conexão de módulos externos representados por aplicações clientes que exercem o papel de intermediador entre plataforma LP2P e usuários. Com relação às dependências, o LP2P necessita especificamente dos recursos de conectividade de rede oferecidos pelo sistema operacional. As seções subsequentes apresentam com maiores detalhes cada componente da arquitetura.

4.4.1 LP2P-CORE

A coordenação das funções de cada componente do modelo é de responsabilidade do LP2P-CORE. Este módulo oferece uma camada de abstração entre plataforma de comunicação, base de conhecimento e clientes. Dentre suas atribuições, estão a carga dos parâmetros de funcionamento relacionados a cada recurso, como o fator de replicação ou o tipo de acesso relacionado ao compartilhamento.

Seu funcionamento dispõe de vários estágios que dependem do tipo de ação a ser executada pelo peer. Dentre os quais, destacam-se o processo de inicialização, o estabelecimento de chaves de sessão, além do atendimento das requisições oriundas de outros participantes e dos clientes locais.

Outra atividade atribuída ao LP2P-CORE está relacionada à indexação dos arquivos inseridos no compartilhamento. Antes da inserção de um arquivo na base de conhecimento local, o gerenciador do peer aplica a este função *hash* proveniente de sua leitura. Só então esse identificador e os demais dados do recurso são repassados ao módulo LP2P-BD para o devido cadastramento. Logo, tal abordagem implica em alterações promovidas pelo usuário que modificam a estrutura do arquivo e, por consequência, seu identificador. No entanto, seu nome permanece o mesmo. Sendo assim, cabe ao LP2P-CORE coordenar essa ação em seu estado de monitoramento do sistema de arquivos local.

4.4.2 LP2P-BD

A gestão e troca de arquivos entre peers necessitam de uma base de conhecimento detalhada, que relacione cada elemento da rede ao compartilhamento e hospedeiro correspondente. A ausência desses registros impede que o peer busque os arquivos pertencentes a um determinado compartilhamento. Portanto, ele não será capaz de inserir seus próprios arquivos nesse ambiente, o que equivale a dizer que o peer encontra-se inoperante ao sistema. Mediante isso, foi atribuído ao módulo LP2P-BD a incumbência dessa tarefa. Através da sinalização recebida dos demais módulos do modelo, o LP2P-BD cria e mantém base específica para cada repositório ingressado pelo *host*.

Na busca dos conteúdos inseridos em cada pasta LP2P, o modelo recorre às informações cadastradas em sua base de conhecimento. Esse conjunto, ilustrado na Figura 4.3, é composto pelos registros de *arquivos*, *proprietários* e *peers*. Os seguintes campos formam o registro de cada arquivo:

- a) **ID Arquivo**: Identificador único derivado de uma função *hash* a partir da leitura do arquivo.
- b) **Nome**: Identificador de 512 *bytes* contendo o nome do arquivo.
- c) **Tamanho**: Espaço que o arquivo ocupa em disco, com suporte até o valor de 10GB.
- d) **TimeStamp**: Sequência que indica a data da primeira inserção do arquivo no diretório LP2P.
- e) **Relação de Proprietários**: Lista de participantes que possuem o arquivo.
- f) **Sinalizador Local**: *Nibble* com três valores válidos. Valor **0** indica que o arquivo não está presente localmente, **1** informa que o arquivo foi copiado para o armazenamento local, e **2** mostra que o arquivo está em disco, sendo o referido peer seu proprietário original (aquele que inseriu o arquivo no compartilhamento).
- g) **ID Evento**: Informa a última ação ocorrida com o arquivo, tendo o valor **1** para inserção e **2** para exclusão do arquivo no compartilhamento especificado. Este campo é utilizado na comunicação do LP2P-CORE com os módulos clientes, no processo de sincronização dos arquivos disponíveis na pasta LP2P.

Para efetuar o cadastro de proprietários, o LP2P-BD emprega o identificador (*ID Proprietário*), derivado do *hostid* da estação onde o LP2P foi ativado, juntamente da indicação do peer. No controle de *hosts* conectados estão envolvidos os seguintes parâmetros:

- a) **ID Peer**: Endereço IP do peer.

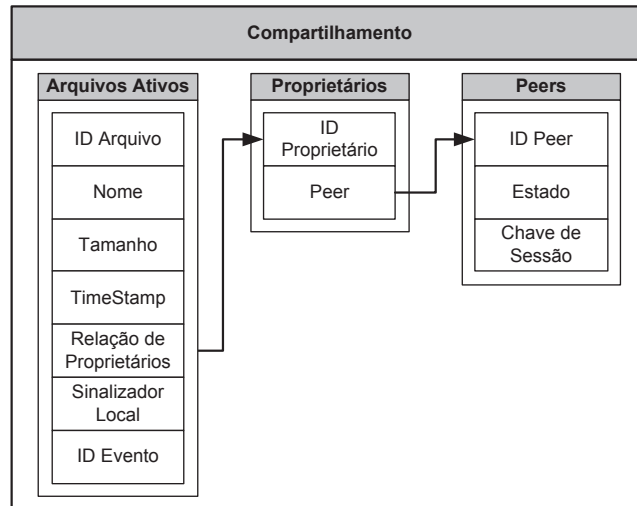


Figura 4.3: Estrutura LP2P-BD

- b) **Estado**: *Booleano* que indica se o *host* está ativo na rede.
- c) **Chave de sessão**: Informação cadastrada em compartilhamento seguro após processo de negociação.

O armazenamento desses registros pode ser executado em memória ou disco utilizando banco de dados local ou conjunto de arquivos. A quantidade de dados compartilhados, aliados ao número de integrantes da rede e abrangência do sistema, determinará o método de armazenamento mais apropriado ao LP2P-BD.

4.4.3 LP2P-MSG

Juntamente com a base de conhecimento, o módulo para gerenciamento de mensagem exerce função vital no funcionamento deste modelo de comunicação. De forma similar, ao LP2P-BD uma falha em LP2P-MSG impedirá o envio e recebimento de solicitações, ocasionando a inoperância do peer.

Seu formato padrão permite ao modelo enviar requisições para o ingresso do *host*, bem como solicitar e receber a lista de arquivos disponíveis. Além disso, é função deste módulo cuidar das notificações quanto às atualizações da rede, inclusão e exclusão de arquivos e o estabelecimento de chaves para envio de mensagens seguras. Juntamente com essas funções, o LP2P-MSG também está incumbido de receber e transmitir os arquivos - de forma parcial ou completa.

4.4.4 Módulos Externos

O presente estudo visa o desenvolvimento do *core* da arquitetura LP2P, abrangendo uma série de estados e primitivas para a união de computadores em torno de uma

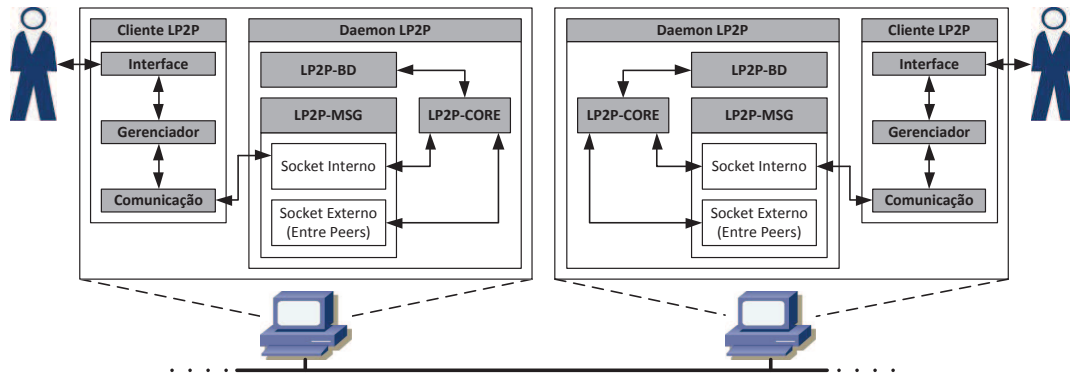


Figura 4.4: Cliente LP2P

abordagem peer-to-peer. Conforme mencionado ao longo deste capítulo, o LP2P comunica-se com outras aplicações para fornecer e solicitar serviços. A entrega de serviços é uma das principais atividades do protocolo, exemplificada pela apresentação dos arquivos compartilhados ao usuário.

Para que ocorra essa integração, é necessário o estabelecimento de padrões que possibilitem tal comunicação. Essas regras definem o escopo de ação das aplicações externas, juntamente com suas principais funcionalidades. Não menos importante, essa incorporação de normas determina como o LP2P irá atuar neste cenário. Logo, a presente seção tem como objetivo ilustrar as especificações necessárias ao seu desenvolvimento e correlação de aplicações externas com o LP2P.

Os clientes LP2P são responsáveis por apresentar os dados compartilhados aos usuários da rede. Para este fim, uma aplicação cliente comunica-se com o LP2P utilizando grupo específico de primitivas. Basicamente, o cliente demanda requisições ao LP2P-CORE, que executa a devida avaliação e responde de acordo com o serviço demandado ou reporta a ocorrência de uma falha.

De acordo com a figura 4.4, os dados compartilhados são gerenciados e propagados pelo módulo LP2P residente em cada participante. Cabe ao cliente realizar uma consulta via protocolo e apresentar os dados obtidos ao usuário. A transferência de informações ou arquivos do LP2P não depende do funcionamento ou de qualquer requisição proveniente dos módulos clientes. Contudo, as consultas e transferências demandadas pelo usuários utilizando essa interface necessitam que o LP2P esteja ativo. Para que isso ocorra de forma eficiente, o módulo LP2P-CORE se encarrega de ativar o LP2P-MSG para atender requisições direcionadas ao endereço de *loopback* do peer na porta 30000. O cliente, por sua vez, deve estar apto a atender pacotes direcionados ao mesmo endereço (*loopback*) na porta 30002.

Apesar de não existirem restrições quanto à arquitetura das aplicações clientes, é esperado que o mesmo apresente certas funcionalidades para correta integração com o

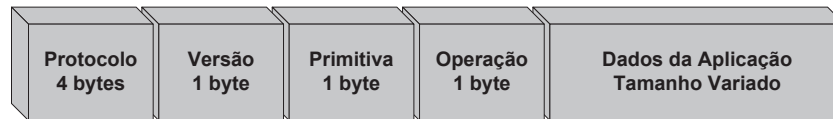


Figura 4.5: Formato geral do pacote LP2P

LP2P. Essencialmente, essa aplicação deve oferecer as funções de *interface*, *gerenciamento* e *comunicação*. A responsabilidade de prover facilidade de uso e apresentação dos dados ao usuário está inserida na camada de interface. Transferir arquivos e informações de controle entre cliente e *daemon* LP2P, obedecendo às regras de mensagens do protocolo, é incumbência da função de comunicação. *O papel de cache do conteúdo*, aliado à coordenação das demais funcionalidades, está a cargo da camada de gerenciamento.

Em Rocha et al. (2010) é apresentado um exemplo desse tipo de aplicação. O LP2PFS foi implementado para o *kernel* do Linux, atuando como um sistema de arquivos local. Estando em conformidade com os padrões de comunicação estabelecidos pelo LP2P, ele permite disponibilizar os arquivos presentes no compartilhamento, atuando como interface com o usuário. Isto significa que a partir do *shell* de sistemas operacionais Linux é possível relacionar uma pasta LP2P a um ponto de montagem específico. O LP2PFS foi desenvolvido com base na camada de abstração do *kernel* para sistemas de arquivos, o Virtual File System (VFS), utilizando o subsistema de Page Cache para o armazenamento dos dados na memória.

4.5 PRIMITIVAS E SISTEMA DE MENSAGENS

A comunicação presente na rede LP2P emprega estrutura de pacotes normatizada (Figura 4.5), onde os 7 primeiros *bytes* são fixos, enquanto o tamanho do *payload* restante dependerá do tipo de operação em execução. Os 4 *bytes* iniciais contemplam a identificação do protocolo, tendo como valor padrão LP2P, enquanto sua versão está localizada no quinto *byte*. As primitivas, que objetivam agrupar de forma hierárquica o conjunto de operações, estão localizadas no sexto *byte* da estrutura. Completando a categorização das mensagens, o sétimo *byte* carrega a informação referente à operação em execução. Dessa forma, o modelo proposto oferece suporte à expansão de suas funcionalidades através das possíveis combinações dos campos 6 e 7 do pacote padrão.

Essa categorização de operações em grupos de primitivas objetiva padronizar o método de transferência de arquivos, bem como a troca de dados para o controle dos compartilhamentos LP2P. Cada operação é classificada de acordo com um grupo, abrangendo ações de *manipulação de arquivos*, *notificações*, *segurança*, *erros do sistema* e *comunicação interna*. A seguir são descritos em maiores detalhes cada grupo, bem como seu respectivo conjunto de operações.

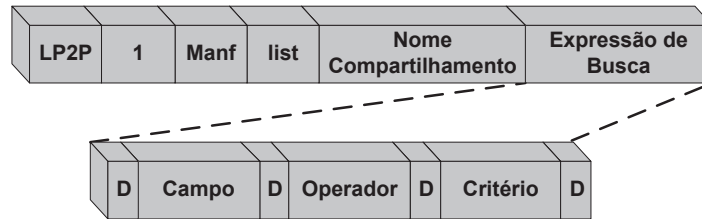


Figura 4.6: Comando LIST

4.5.1 Grupo 1 - Manipulação de arquivos

Este grupo engloba a divulgação dos arquivos disponíveis na rede, bem como seu transporte entre peers. Sendo assim, suas mensagens visam carregar instruções que solicitem ou retornem registros de arquivos inseridos no compartilhamento, ou seu conteúdo. A primitiva de manipulação de arquivos é composta pelo seguinte conjunto de operações:

- **list**

Os pacotes do tipo *list* têm como objetivo solicitar a lista de arquivos que determinado compartilhamento detém. Para este fim, este comando é encaminhado a todos os integrantes do ambiente LP2P na forma de pacote *multicast* direcionado ao grupo correspondente. Sua estrutura, apresentada na Figura 4.6, contém a parcela fixa do protocolo LP2P, juntamente com o nome do repositório acompanhado de um critério de busca. Esse parâmetro, enviado como parte da mensagem, tem a função de informar aos *hosts* remotos como proceder na filtragem dos arquivos que o peer tem interesse em receber.

Essa expressão é composta pelo campo a ser consultado (ex.: Nome_Arquivo ou Tamanho), um operador (`==`, `!=`, `>`, `>=`, `<`, `<=`) e o respectivo critério de avaliação (ex.: Nome_Arquivo `==` Artigo.pdf) separados por um delimitador de campo (na figura, o campo marcado com a letra D). No processo de anúncio, o peer não dispõe da lista de arquivos disponíveis. Por isso, essa expressão é enviada no formato `* == *`, solicitando todos os dados já cadastrados no diretório LP2P.

- **sendl**

Os pacotes da operação *sendl* são disparados em resposta à requisições do tipo *list*. Essa mensagem traz em seu cabeçalho os dados fixos do protocolo, o compartilhamento envolvido e uma listagem contendo todos os arquivos cadastrados no diretório LP2P. Da mesma forma que ocorre com o pacote *list*, *sendl* é transmitido utilizando *multicast*, auxiliando na replicação dos registros da base de conhecimento entre todos os *hosts* conectados ao recurso.

De acordo com a Figura 4.7, a indicação de cada arquivo inserido nessa mensagem compreende um identificador do recurso (*hash* resultante da leitura do arquivo), o

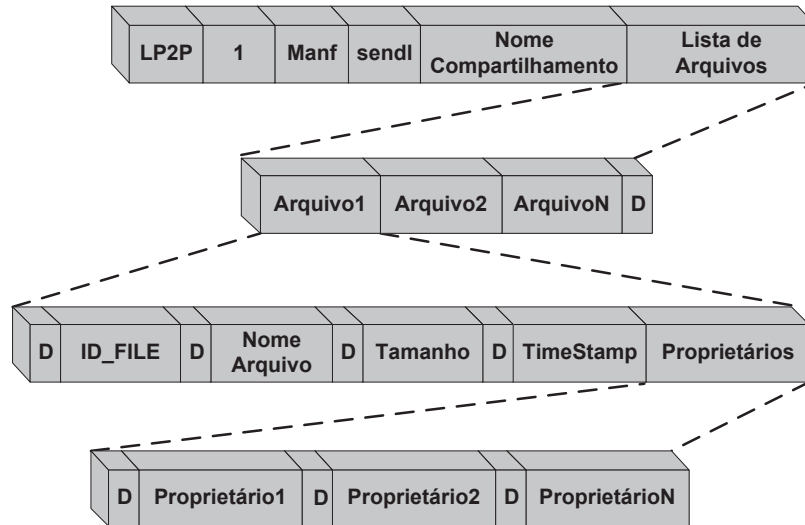


Figura 4.7: Comando SENDL

nome do arquivo, bem como seu tamanho e *timestamp*. A relação de proprietários completa as características do arquivo compartilhado.

É importante enfatizar que o tamanho dessa listagem é variável, dependendo diretamente da relação entre conteúdos disponíveis na pasta LP2P e peers conectados. Conseqüentemente, características como quantidade de registros e proprietários, além do fator de replicação irão influenciar no tamanho dessa mensagem.

- **get**

A mensagem *get* requisita o envio de arquivos, empregando uma ou diversas fontes conectadas à pasta LP2P. Nessa operação (Figura 4.8a) é solicitado o envio de um arquivo ou de uma parte de seu conteúdo para o *host* que originou esse pedido. Logo, sua execução pode ser realizada de forma distribuída ou ponto-a-ponto, em que o uso de cada um desses métodos dependerá da quantidade de fontes do arquivo a ser buscado.

A diferença fundamental entre os dois métodos está na forma como o peer irá realizar essa requisição. No caso distribuído, pacotes com essa mensagem são emitidos de acordo com a quantidade de fontes a serem acionadas. Em cada mensagem, os campos *índice* e *valor* são configurados para proporcionar a cópia distribuída. *Índice* define o *byte* inicial para a cópia, enquanto *Valor* determina a parcela de dados que será transferida a partir do ponto delimitado pelo campo *Índice*. No modelo ponto-a-ponto, ambos os campos são configurados com o valor zero. Com isso, o *host* que receber a requisição *get* nesse formato reconhece que todo o conteúdo do arquivo deve ser remetido ao demandante desse comando.

- **sendf**

Em resposta à solicitação *get*, o pacote *sendf* é gerado. Nessa mensagem, ilustrada

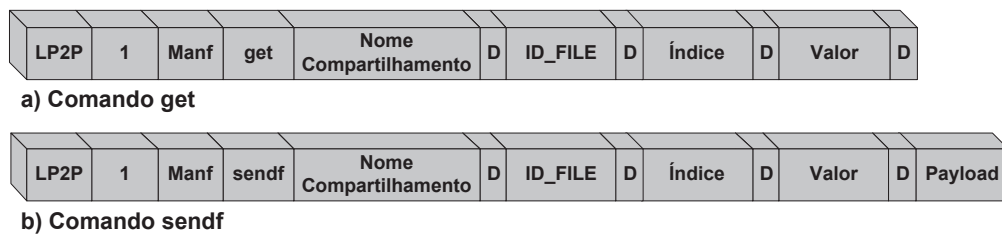


Figura 4.8: Comandos GET e SENDF

na Figura 4.8b, o arquivo solicitado é inserido no campo *payload* de forma parcial ou completa. As definições entregues junto à mensagem *get* determinarão o método de envio a ser empregado.

Sendf dispõem de estrutura similar à mensagem *get*, com exceção do campo *payload*, reservado para o conteúdo que será transportado. As funções dos campos *Índice* e *Valor* são idênticas à mensagem de solicitação de arquivos, entretanto, neste caso, tais informações servem para auxiliar o requerente do arquivo no processo de remontagem das partes recebidas.

4.5.2 Grupo 2 - Notificações do sistema

Da mesma forma que ocorre em redes peer-to-peer tradicionais, os usuários do sistema LP2P dispõem da prerrogativa de alteração do conteúdo armazenado localmente. Mediante este fato, cabe a primitiva de *Notificações do Sistema* oferecer um conjunto de métodos capaz de disseminar tais mudanças que ocorrem nos diretórios LP2P de cada participante. Portanto, o envio desses pacotes é acionado de acordo com as modificações detectadas pelo peer. Tendo em vista que a manutenção da base de informações é uma ação compartilhada, torna-se necessário que cada participante notifique aos demais essas ocorrências utilizando o seguinte conjunto de mensagens:

- **add**

Para adicionar arquivos à pasta LP2P, uma mensagem do tipo *add* é gerada e remetida ao endereço *multicast* correspondente ao repositório envolvido nessa operação. Esse pacote, ilustrado na Figura 4.9, é formado pelos *bytes* obrigatórios e pelos dados de aplicação. Dentre as características da rede, estão o nome do compartilhamento e o identificador do proprietário, ambos de *255 bytes*. Com relação às informações de cada arquivo, esta mensagem repassa os campos *nome* e *identificador* de *512 bytes* juntamente de seu *tamanho* e *timestamp*.

- **del**

Na operação de exclusão, além dos campos fixos, apenas o nome do compartilhamento e o identificador de arquivo são necessários para que esta operação seja efetivada.

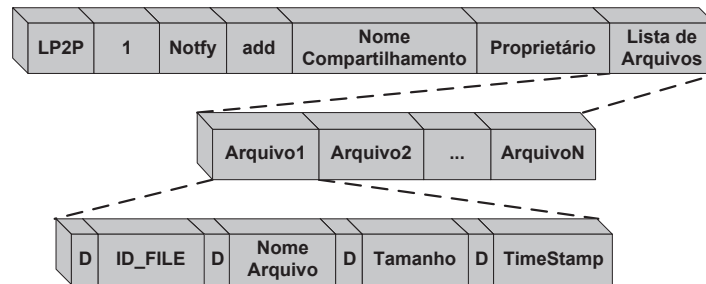


Figura 4.9: Comando ADD

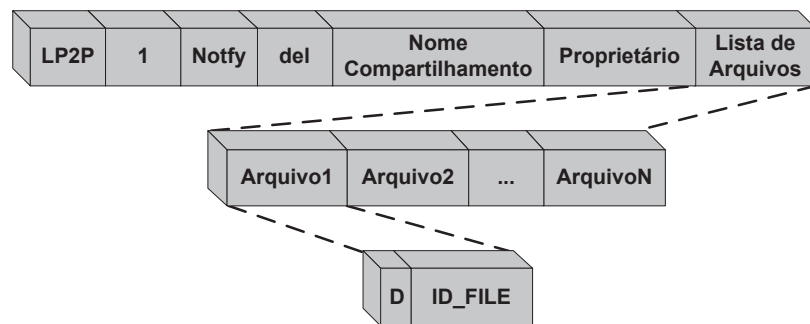


Figura 4.10: Comando DEL

Neste âmbito, ao receber uma mensagem *del*, cada *host* exclui de sua base de conhecimento o referido apontamento contido nesta operação. Sua estrutura (Figura 4.10) possibilita a exclusão de um ou mais arquivos, cabendo ao peer que receber tal notificação vasculhar sua base de conhecimento realizando os ajustes necessários.

A execução dessas mensagens exerce influência direta na manutenção dos índices de cada área compartilhada. Sua incorporação ao modelo LP2P se justifica ao evitar falsos positivos no processo de busca e transferência de arquivos. Ou seja, um peer terá maior probabilidade de sucesso nas operações de busca e transporte dos conteúdos cadastrados em seus registros.

4.5.3 Grupo 3 - Segurança

O processo de entrada do *host*, bem como a sincronização do fator de replicação, são funções das mensagens inseridas neste grupo. Além disso, os aspectos de segurança também são abordados por este grupo de mensagens, promovendo fluxos para troca de chaves visando a comunicação segura entre peers LP2P. Sendo assim, este grupo conta com as seguintes mensagens:

- **AnL**

Um peer que ingresse na rede deve anunciar sua presença aos demais. Para atender essa necessidade, o pacote *AnL* (*Anúncio LP2P*), ilustrado na Figura 4.11a, é

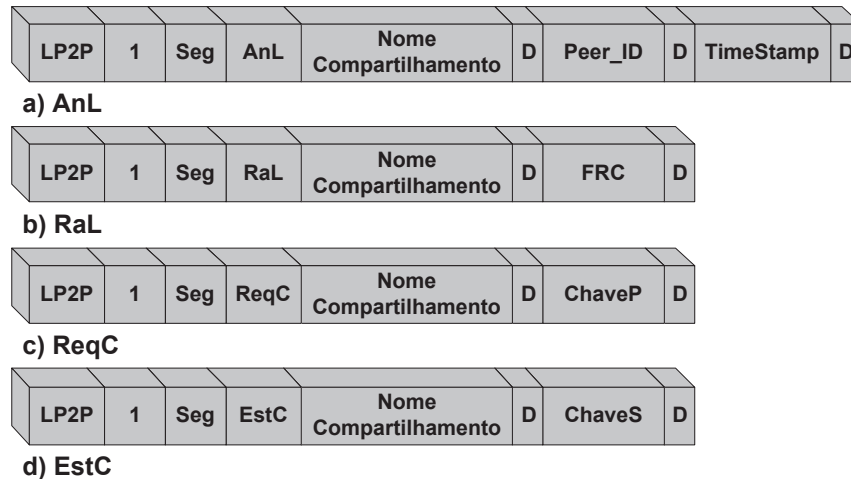


Figura 4.11: Mensagens AnL / RaL / ReqC / EstC

remetido ao endereço *multicast* do compartilhamento. O envio desta mensagem pode variar de acordo com o tipo de compartilhamento (público/restrito). No acesso restrito os campos `Peer_ID` e `TimeStamp` são inseridos na mensagem, do contrário, esses campos são configurados com valor nulo.

- **RaL**

O pacote do tipo *RaL* (*Resposta ao Anúncio LP2P*) (Figura 4.11b) é enviado via *unicast* ao *host* que originou a mensagem de anúncio. Além de confirmar a entrada do peer, *RaL* informa o fator de replicação (Campo `FRC`) aplicado ao compartilhamento.

- **ReqC**

Utilizado em compartilhamentos seguros, a mensagem *ReqC* (*Requisição de chave*) dá início ao processo de negociação da chave de sessão. Como essa chave é aplicada para garantir segurança em mensagem ponto-a-ponto, esta solicitação é enviada na forma de *unicast* ao *host* no qual se deseja estabelecer tal fator criptográfico. A mensagem *ReqC* (Figura 4.11c) utiliza o *hostid* e o *timestamp* do sistema para gerar uma chave primária, sendo esta criptografada com o auxílio da chave de grupo associada à pasta LP2P. O resultado final dessa ação é então inserido no campo **ChaveP** e enviado ao destino com a garantia de que a confidencialidade desses dados será preservada.

- **EstC**

Ao receber uma mensagem *ReqC*, o peer emprega a chave de grupo visando obter a chave primária (*ChaveP*). Essa informação é então utilizada para cifrar a chave secundária (*ChaveS*), também derivada do *hostid* e do *timestamp* do sistema. A chave secundária é então inserida no campo correspondente do pacote *EstC* (*Estabelecimento de Chave*) (Figura 4.11d) e remetido ao *host* que emitiu *ReqC*.



Figura 4.12: Formato padrão das notificações de erro

4.5.4 Grupo 4 - Erros do sistema

Os grupos de primitivas apresentados anteriormente executam apenas ações de solicitação e resposta a determinado evento ou estado exercido pelo peer. Contudo, todo sistema está sujeito a algum tipo de falha, necessitando o devido tratamento e notificação das partes envolvidas. Por definição, o modelo LP2P realiza a notificação de falhas aplicando primitiva específica para tal finalidade.

O grupo de *Erros do Sistema* executa esse serviço dispondo de uma mensagem de formato padrão (Figura 4.12), contendo as informações obrigatórias do protocolo, além de uma sequência de três *bytes* destinados à especificação da falha. Esses campos definem respectivamente o identificador da primitiva, sua operação além do tipo de erro. O formato adotado nesta mensagem permite a ampliação desse conjunto de códigos em futuras versões do modelo LP2P. A Tabela 4.2 relaciona os códigos mapeados na primeira versão do modelo-tema do presente estudo.

A relação de erros abrange respectivamente falhas gerais e específicas. A primeira situação ocorre quando o erro encontrado não corresponde às possibilidades até então mapeadas (ex.: Código 111 - Erro geral na operação *list*). Neste caso, nenhuma notificação ao emissor será efetivada, visto que esses erros são de caráter informativo. Com relação aos erros específicos, eles já estão delineados e, por consequência, associados ao devido código de erro (ex.: Código 142 - ID_FILE inválido na mensagem *sendf*).

Quanto à detecção de tais eventos, esta tem por objetivo gerar uma notificação ao sistema. No contexto do modelo LP2P, a retransmissão desse aviso dependerá do tipo de mensagem, bem como do erro associado. Para o grupo de manipulação de arquivos, os erros 112 (*list* - Expressão de busca inválida) e 113 (*list* - Solicitante não ingressou na pasta LP2P) serão remetidos ao *host* que enviou o pacote *list*. Contudo, o envio desse aviso respeitará o fator de probabilidade, evitando múltiplas respostas.

Ao receber o erro 112, o peer refaz sua solicitação. Este tratamento também ocorre com os códigos 132 a 134, 142 a 144, 212 e 213, 222. No caso da notificação 113, ela indica que o *host* não realizou o processo de anúncio ou foi definido como desconectado do sistema. Tal estado faz com que o *host* reinicie sua participação no compartilhamento desde a fase de anúncio do peer. De maneira similar ao erro 113, a mesma abordagem será adotada no recebimento das notificações do tipo 113, 135, 214, 223, 332 e 342.

Tabela 4.2: Códigos de erros LP2P

ID_PRIMITIVA	Nome Primitiva	ID_OPERAÇÃO	Nome Operação	ID_ERRO	Significado
1	Manipulação de Arquivos	1	list	1	Erro geral na operação list
1		1	list	2	Expressão de busca inválida
1		1	list	3	Solicitante não ingressou na pasta LP2P
1		2	sendl	1	Erro geral na operação sendl
1		2	sendl	1	Compartilhamento Inválido
1		2	sendl	1	Emissor não ingressou na pasta LP2P
1		3	get	1	Erro geral na operação get
1		3	get	2	ID_FILE não encontrado
1		3	get	3	Erro Campo Índice
1		3	get	4	Erro Campo Valor
1		3	get	5	Solicitante não ingressou na pasta LP2P
1		4	sendf	1	Erro geral na operação sendf
1		4	sendf	2	ID_FILE inválido
1		4	sendf	3	Erro Campo Índice
1		4	sendf	4	Erro Campo Valor
1		4	sendf	5	Payload inválido
1		4	sendf	6	Emissor não ingressou na pasta LP2P
2		Notificações do sistema	1	add	1
2	1		add	2	Erro Campo proprietário
2	1		add	3	Lista de arquivos inválida
2	1		add	4	Emissor não ingressou na pasta LP2P
2	2		del	1	Erro geral na operação del
2	2		del	2	Lista de arquivos inválida
2	2	del	3	Emissor não ingressou na pasta LP2P	
3	Segurança	1	AnL	1	Erro geral na operação AnL
3		1	AnL	2	Compartilhamento Inválido
3		2	RaL	1	Erro geral na operação RaL
3		2	RaL	2	Compartilhamento Inválido
3		3	ReqC	1	Erro geral na operação ReqC
3		3	ReqC	2	Solicitante não ingressou na pasta LP2P
3		3	ReqC	3	Compartilhamento Inválido
3		4	EstC	1	Erro geral na operação EstC
3		4	EstC	2	Emissor não ingressou na pasta LP2P
3	4	EstC	3	Compartilhamento Inválido	

Para o tratamento das falhas decorrentes da primitiva de segurança, o LP2P executará os seguintes procedimentos:

Código 311 e 321) O emissor de AnL ou RaL não são notificados.

Código 331 e 341) O estabelecimento de chaves é reiniciado, sendo os dados do atual processo descartados.

Código 312, 322, 333 e 343) Essas falhas podem ocorrer quando o mesmo grupo *multicast* for atribuído a compartilhamentos de nomes diferentes. Neste caso, as mensagens contendo o nome de outro repositório são apenas ignoradas.

4.5.5 Grupo 5 - Comunicação Interna

A comunicação interna é um processo que ocorre entre o *daemon* LP2P e seus clientes, exigindo um tratamento diferenciado daquele aplicado na interação com outro *daemon* LP2P. Aspectos como portabilidade e limitações podem variar de acordo com diversos fatores, desde limitações do *kernel* até o tipo de linguagem empregada em seu desen-

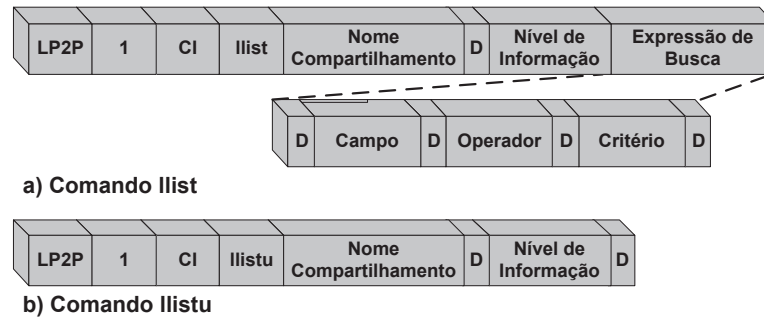


Figura 4.13: Comunicação interna - llist/llistu

volvimento. Tendo em vista essas particularidades, este grupo de primitivas oferece um conjunto de operações mais flexível. Tal abordagem pretende atender esse tipo de aplicação, independentemente de implementação, empregando o seguinte conjunto de operações:

- **llist**

Local List (llist), ilustrada na Figura 4.13a, executa tarefa similar à mensagem *list*. O **Nível de Informação** é o fator preponderante que diferencia esta mensagem do pacote *list*. O uso deste parâmetro possibilita que o cliente informe a quantidade campos que a resposta de *llist* deve conter, proporcionando maior flexibilidade no desenvolvimento de clientes LP2P. Basicamente, a configuração deste campo obedece à seguinte categorização:

- Nível 1)** Para cada registro requer que a resposta informe apenas os campos *ID_FILE* e *Nome Arquivo*.
- Nível 2)** Neste caso, são remetidos além dos campos do nível anterior as informações de *Tamanho* e *TimeStamp*.
- Nível 3)** Cada registro deve ser enviado com sua totalidade de informações cadastradas na base de conhecimento.

Esta mensagem pode ser empregada em dois momentos distintos: primeiro, na inicialização do cliente, pois o mesmo desconhece os arquivos disponíveis e, portanto, está impossibilitado de apresentar algo ao usuário; segundo, no momento em que o cliente queira reiniciar toda sua relação de registros, ele envia tal requisição para receber novamente a lista completa de registros.

Neste cenário, é possível atender requisições que necessitem de um conjunto menor de informações. Um exemplo seria um sistema de arquivos LP2P, no qual a quantidade de fontes por arquivo é irrelevante para seu funcionamento. Em contrapartida, uma aplicação com o propósito de gerar estatísticas dos compartilhamentos LP2P utilizaria todos os dados cadastrados. Logo, seria viável obter resultados exatos

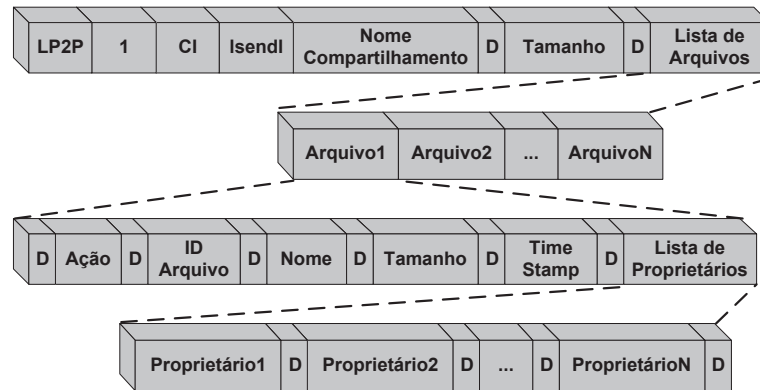


Figura 4.14: Comunicação interna - lsendl

quanto à ocupação de disco e disseminação de arquivos no ambiente LP2P, gerando assim relatórios que relacionassem esses aspectos.

- **llistu**

A definição da mensagem *local get update (llistu)*, ilustrada na Figura 4.13b, também está relacionada ao atendimento de diversos tipos de cliente. Aplicações com requisitos específicos de memória (ex.: módulo do *kernel* para sistema de arquivos LP2P) teriam seu desempenho prejudicado caso mensagens do tipo *add* e *del* fossem atribuídas às primitivas de comunicação interna.

Neste contexto, o envio de atualizações em lote apresenta melhor funcionamento em comparação às mensagens *del* e *add*. Logo, *llistu* é enviada da aplicação ao LP2P, solicitando as modificações que ocorrem nos registros do compartilhamento desde a última sincronização dada por uma mensagem do tipo *sendl* ou *lsendu*;

- **lsendl**

Local sendl(Figura 4.14) é uma adaptação do comando *sendl* para o contexto da primitiva de comunicação interna. Atua como resposta às mensagens *llist* e *llistu*, possibilitando o envio total ou parcial dos dados de cada arquivo. O campo **Ação** indica se um arquivo foi adicionado ou removido do compartilhamento, possibilitando uma resposta adequada às mensagens *llist* e *llistu*.

Na resposta ao comando *llist*, o campo **Ação** é configurado de forma a indicar que o conteúdo listado está sendo adicionado ao recurso compartilhado. Com relação à resposta do tipo *Local List Update*, o campo Ação possibilita informar ações de exclusão, inclusão e modificação de arquivos.

- **lget**

Desempenha a mesma função do comando *get*, tendo como diferencial o fato de não especificar o transporte parcial do arquivo. Sendo assim, a mensagem *Local get* (Figura 4.15a) emprega uma estrutura de maior simplicidade, bastando ao cliente informar o identificador referente ao arquivo que deseja receber.

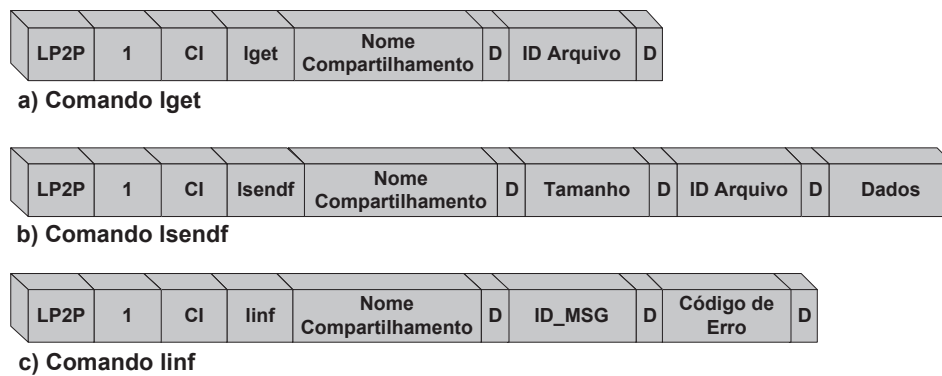


Figura 4.15: Comunicação interna - lget / lsendf / linf

- **lsendf**

Local Send File (sendf) entrega ao cliente todo o arquivo requisitado. Não executa entrega parcial dos arquivos, e informa ao módulo solicitante o tamanho do arquivo que será entregue. Como ilustra a Figura 4.15b, o formato de *lsendf* difere da mensagem *sendf* ao não empregar os campos *Índice* e *Valor*.

- **linf**

Local Info relaciona os erros ocorridos nas mensagens desta primitiva, empregado para notificar *daemon* e cliente LP2P quanto ao surgimento de erros. A Figura 4.15c relaciona cada campo desta mensagem e sua respectiva posição dentro do pacote. A relação de erros utiliza dois *bytes*, identificando respectivamente a operação e o código de erro associado. Referente à lista de erros mapeados na presente versão do LP2P, a Tabela 4.3 apresenta essa relação. O valor **0** atribuído ao campo *ID_MSG* indica um erro geral detectado antes mesmo de ocorrer a verificação da operação inserida nas mensagens de comunicação interna.

Tabela 4.3: Mapeamento de erros para Comunicação Interna

ID_MSG	Operação	ID_ERRO	Significado
0	Primitiva	1	Compartilhamento não encontrado
1	llist	1	Erro geral na operação llist
1	llist	2	Nível de informação desconhecido
1	llist	3	Expressão de busca inconsistente
2	llistu	1	Erro geral na operação llistu
2	llistu	1	Nível de informação desconhecido
3	lsendl	1	Erro geral na operação lsendl
3	lsendl	2	Tamanho da mensagem inválido
3	lsendl	3	Erro na leitura dos índices
4	lget	1	Erro geral na operação lget
4	lget	2	ID_Arquivo inválido
5	lsendf	1	Erro geral na operação lsendf
5	lsendf	2	Tamanho da mensagem inválido
5	lsendf	3	Erro na leitura dos Dados

4.6 FASES DE FUNCIONAMENTO DO PEER

Os grupos de primitivas, aliados aos módulos da arquitetura LP2P, possibilitam um conjunto de ações e estados desempenhados em cada participante. Tais fatores proporcionam um ciclo de vida referente à participação de cada peer junto ao sistema LP2P. Este processo contempla desde ações básicas, como a leitura de arquivos de inicialização, passando ainda pela adesão ao compartilhamento, a troca de registros, recebimento e envio de arquivos.

Em cada etapa, cabe ao *host* solicitar serviços dos demais integrantes da rede, ou ser solicitado para atender requisições. Em ambos os casos, os processos poderão ser efetuados de forma distribuída (ex.: na descoberta ou busca de arquivos) ou mediante comunicação ponto-a-ponto (ex.: estabelecimento de chaves). O restante desta seção explora ambas as possibilidades, apresentando suas ações e devidas características.

4.6.1 Inicialização

Para iniciar suas funções, o LP2P-CORE realiza a carga de um conjunto de parâmetros antes do peer tentar ingressar na rede. Nesses dados estão a relação de compartilhamentos que o *host* realizará no processo de entrada, além do tipo de acesso e os aspectos de segurança vinculados a cada recurso. O LP2P-CORE obtém esse conjunto de informações a partir da leitura dos dados contidos em arquivos de configuração. Outra atribuição relacionada à fase de inicialização é a determinação do fator de replicação que o peer utilizará como referência no processo de anúncio.

Após o levantamento dessas informações, LP2P-CORE aplica ao nome de cada repositório uma função *hash* de forma a obter um endereço IP *multicast* correspondente. Em virtude de existir a possibilidade da atribuição do mesmo endereço a diferentes recursos, cada mensagem do modelo carrega o nome do compartilhamento juntamente com os dados da operação.

O tipo de acesso é classificado como público ou restrito, tendo como diferença o uso de criptografia dos dados (após o sétimo *byte*) vinculados ao compartilhamento de acesso restrito. Para este fim, é utilizada uma chave de grupo obtida no processo de carga da configuração. Após, LP2P-CORE interage com LP2P-MSG para informá-lo de que os pacotes destinados ao compartilhamento em questão deverão ser criptografados com o auxílio da chave de grupo.

O fator de replicação no sistema LP2P define o momento da execução da cópia permanente dos arquivos - quando será realizada a transferência definitiva destes. Conforme relatado na Seção 4.4.4, o *cache* de conteúdo é de responsabilidade do módulo cliente, cabendo ao LP2P a transferência definitiva (quando ocorre armazenamento local) apenas

quando as definições de replicação estiverem atendidas. Sendo assim, esse coeficiente estabelece uma relação entre transferências do arquivo e sua disseminação na rede, obedecendo às seguintes especificações:

$$FrL = \left(\frac{NmR}{Np} \right) \quad (4.1)$$

$$FrE = \left(\frac{NaR/Np}{NmT} \right) \quad (4.2)$$

$$FrC = \left(\frac{NmR/Np}{NaT/NmT} \right) \quad (4.3)$$

$$Ct \longleftrightarrow \left[\left((FrE \leq FrL) \wedge (FrE = FrC) \right) \vee \left(NmT = \frac{NaT}{2} \right) \right] \quad (4.4)$$

Onde:

FrL: Fator de Replicação LP2P;

FrE: Fator de Replicação Esperado;

FrC: Fator de Replicação Calculado;

Ct: Condição de transferência;

NmR: Número máximo de réplicas;

Np: Número de peers conectados;

NaR: Número atual de réplicas;

NmT: Número máximo de Transferências;

NaT: Número atual de Transferências;

FrL representa o percentual de replicação atribuído ao diretório LP2P, sendo aplicado a cada arquivo compartilhado. Sua faixa de valores varia de 0 (sem replicação) a 1 (replicação total). Atribuir a esse coeficiente seu limite inferior implica na desativação da replicação. Sendo assim, mesmo que diversas solicitações sejam realizadas por determinado peer a igual conteúdo (independente de sua origem), nenhuma réplica será efetuada. De maneira análoga, ao configurar o fator de replicação em seu limite superior, as cópias serão criadas após a conclusão da primeira transferência.

Para os demais casos, localizados entre os limites superior e inferior de *FrL*, o gerenciador do peer empenha o rol de equações definidas anteriormente para deliberar quando a condição de transferência (*Ct*) será atingida. Nesta análise, a relação de peers e réplicas por arquivo é apurada em conjunto do volume de cópias efetuadas do referido recurso. Dessa observação o LP2P-CORE extrai as medidas pertinentes ao fator de replicação esperado e calculado.

FrE é estabelecido em função da quantidade de fontes e peers conectados no momento

de nova transferência. Mantendo-se inferior a FrL , indica que o percentual de disseminação do conteúdo não atingiu o patamar estipulado ao diretório LP2P. Ao superar o referido delimitador, informa que existem fontes excedentes para o referido arquivo. A taxa atual e mínima de solicitações para tornar um arquivo elegível a replicação, juntamente com FrE , estabelecem o fator de replicação calculado FrC . Sendo assim, o critério de cópia poderá ser atingindo nos seguintes momentos:

1. Enquanto o fator de replicação não for superado, e o número de solicitações atingir o valor mínimo estipulado ao compartilhamento;
2. Ao existir cópias excedentes, e a quantidade de solicitações for o dobro do limite associado ao diretório LP2P. Isso permite que conteúdos amplamente requisitados sejam copiados, reduzindo o tráfego na rede.

De toda esta abordagem, surgem três cenários quanto à disponibilidade de recursos e arquivos do diretório LP2P. No primeiro, ocorre alta disponibilidade dos arquivos devido à quantidade de réplicas existentes, mas o espaço total de armazenamento é limitado. Por outro lado, a baixa disponibilidade dos arquivos implica em menor quantidade de réplicas e maior espaço possível para inserir arquivos na pasta LP2P. Como terceira alternativa, uma política de equilíbrio poderia ser adotada ao fator de replicação, permitindo maior espaço e relação de arquivos disponíveis no compartilhamento LP2P.

4.6.2 Anúncio

Ao concluir a fase de inicialização, o gerenciador coordena o módulo LP2P-MSG para enviar mensagens de anúncio via *multicast* ao endereço do compartilhamento calculado na fase de inicialização. Esta mensagem, entregue a todos os *hosts*, informa a tentativa de ingresso de novo peer ao sistema LP2P. Ao receber o anúncio, os demais integrantes utilizam um fator de probabilidade para definir qual deles irá assumir o processo de entrada do novo *host*. Esta função é dada ao módulo LP2P-CORE, que faz a leitura dos registros inseridos em LP2P-BD e calcula esse parâmetro.

O peer que estiver coordenando essa ação de ingresso deve enviar uma resposta à mensagem de anúncio, sendo esta endereçada ao grupo *multicast* do compartilhamento. Desta forma, o novo peer é notificado de sua adesão ao LP2P, enquanto os demais membros são informados da alteração na composição de integrantes da rede. Assim, em cada peer o LP2P-CORE correspondente executa a inserção de um novo registro em sua base de conhecimento. Essa ação confirma o reconhecimento dos participantes ativos e possibilita o recebimento de pacotes de atualização e requisição enviados por esse *host*.

O anúncio do *host* é repetido 4 vezes em intervalos de 30 segundos. Ao final desse ciclo, se a notificação de ingresso não for recebida, o LP2P-CORE constata que o sistema

LP2P encontra-se desativado. Conseqüentemente, esse peer é o primeiro a ingressar no compartilhamento, reativando essa pasta.

A probabilidade empregada nesta fase objetiva evitar que respostas redundantes, e, por sua vez, desnecessárias sejam inseridas na rede. O cálculo deste fator dependerá do tipo de pacote para o qual o mesmo está sendo calculado e obedecerá às seguintes relações:

$$Pe = \left(\frac{1}{Npeers} \right) \quad (4.5)$$

$$Pe = \left(\frac{1}{Npeers - Nsolicitantes} \right) \quad (4.6)$$

Onde:

Pe: Probabilidade de envio;

Npeers: Número de peers conectados ao sistema;

Nsolicitantes: Número de solicitantes conectados ao sistema;

Na mensagem de anúncio, este fator é obtido utilizando a equação 4.5, pois a resposta deste pacote deve considerar todos os peers ativos no ambiente LP2P. Nesta relação, o *host* que envia a mensagem de anúncio não é considerado um membro ativo na rede. Nos demais casos, as mensagens envolvem apenas *hosts* conectados, logo, a equação 4.6 apresenta melhor conformidade, visto que exclui do cálculo os membros que enviaram uma solicitação.

4.6.3 Atualização

Após a confirmação de acesso, o novo integrante remete mensagem pedindo a lista de recursos disponíveis (*list*). Novamente, o fator de probabilidade é empregado e apenas um *host* responderá a essa requisição (*sendl*). Essa sequência de mensagens também ocorre na forma de *multicast*, proporcionando que a lista de arquivos disponíveis esteja consistente em toda rede.

Tendo recebido a relação de recursos, o LP2P-CORE do solicitante atualiza sua base de conhecimento local com este rol de informações. No mesmo instante, também é executado a verificação dos arquivos locais que estão relacionados ao ambiente compartilhado. Então, os registros são inseridos na base de conhecimento local e, posteriormente, propagados para os demais peers por uma mensagem de adição de conteúdo (*add*).

Com o recebimento de uma mensagem de adição, o LP2P-CORE repassa seu conteúdo ao módulo LP2P-BD para a devida atualização da base de conhecimento. Nesse processo, um arquivo poderá ser alterado ou adicionado ao compartilhamento. A alteração ocorre

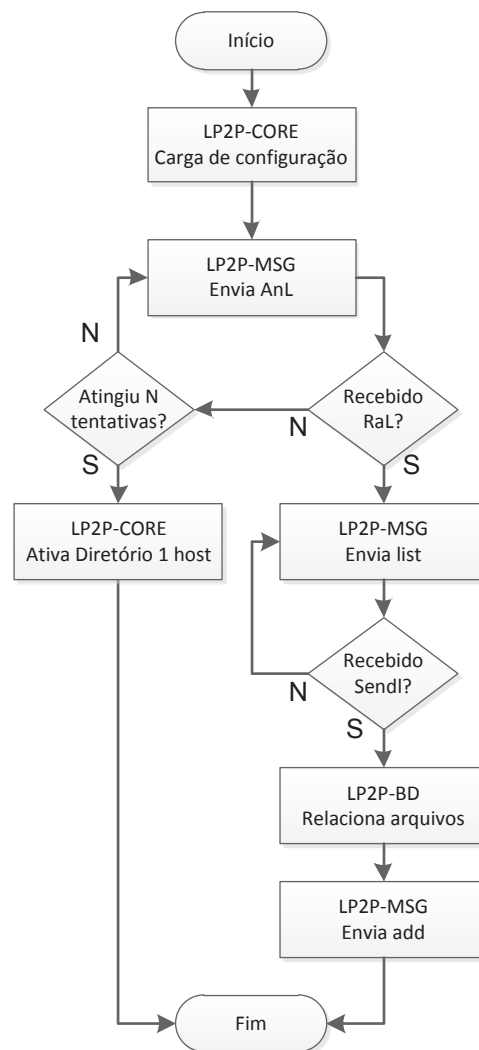


Figura 4.16: Fases de Inicialização, Anúncio e Atualização

quando o arquivo já estiver disponível, implicando na existência de uma nova fonte ao referido conteúdo. Neste caso, a listagem de proprietários do arquivo é alterada com a inserção do novo peer.

No caso da reativação do compartilhamento, cabe ao LP2P-CORE do peer apenas realizar a verificação dos arquivos locais e inseri-los na base de conhecimento coordenada pelo módulo LP2P-BD.

Com o término das fases de *inicialização*, *anúncio* e *atualização*, ilustradas na Figura 4.16, o peer passa a ter uma visão completa do diretório LP2P, bem como dos arquivos inseridos nessa pasta distribuída. O *host* efetivamente está conectado ao sistema e pode agora participar do processo de ingresso de novos participantes. No entanto, demais funcionalidades ainda não foram devidamente ativadas. Dentre elas, destacam-se o monitoramento dos arquivos locais e tratativas para descarte de arquivos em *cache*. Sendo assim, o restante desta seção aborda os seguintes estados do peer:

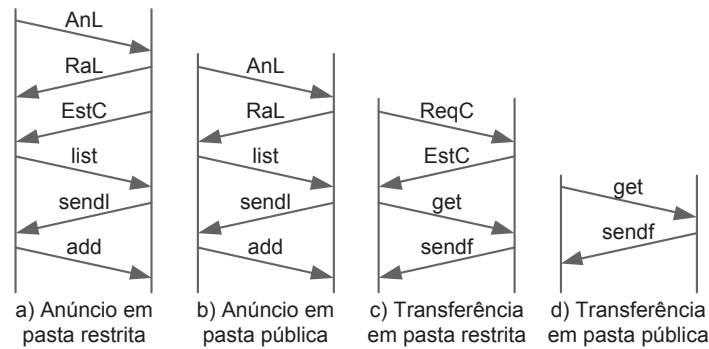


Figura 4.17: Comparação de processos restritos e públicos.

4.6.4 Estabelecimento de Chaves de Sessão

Na ocorrência de compartilhamentos seguros, é necessária a negociação de uma chave de sessão entre *hosts* LP2P. Essa chave será aplicada na respectiva transferência de arquivos entre participantes de um mesmo compartilhamento restrito. Para este fim, o LP2P-CORE, ao identificar que necessita realizar uma comunicação criptografada, deve garantir esse recurso antes de realizar qualquer envio ou recebimento de arquivos.

Primeiramente, o gerenciador do peer avalia para qual peer será realizada a transação. Após, ele confirma em sua base de conhecimento a disponibilidade do recurso criptográfico associado ao referido *host*. Caso esse recurso exista, a comunicação transcorre normalmente utilizando a chave de sessão. Do contrário, o peer inicia a negociação dessa chave com a sua contraparte.

Neste processo, o peer, na função de inicializador da chave, coleta as informações de *hostid* e *timestamp* do sistema para gerar uma chave primária. Essa chave então é criptografada com o auxílio da chave de grupo e inserida no pacote *ReqC*. Então, esse pacote é remetido ao *host* correspondente. O LP2P-CORE do peer que recebeu o pacote *ReqC* extrai a chave primária e reutiliza para cifrar seu próprio *hostid* e *timestamp* no formato da chave secundária. O resultado dessa operação é inserido na mensagem *EstC* e devolvido ao *host* que enviou o pacote *ReqC*.

Após, com ambas as partes de posse das duas chaves parciais, é gerada a chave de sessão a partir da união da *ChaveP* e *ChaveS*. Essa chave é então armazenada na base de conhecimento, e as comunicações subsequentes entre as partes utilizarão esse recurso de segurança em suas transações. No caso de erros nas operações *ReqC* e *EstC*, a devida notificação dessa falha é remetida ao *host* responsável por tal pacote, as chaves parciais até então trocadas são descartadas e a negociação é reiniciada.

Uma outra alternativa ao estabelecimentos de chaves é decorrente da fase de anúncio do peer. Neste caso, o peer tem ciência de que tentará ingressar em compartilhamento seguro. Por esse motivo, a mensagem de anúncio é remetida contendo os dados de *hostid*

e *timestamp*. Dessa forma, o peer que enviar a resposta *RaL*, remete na sequência uma mensagem do tipo *EstC*. Cabe ao LP2P-CORE do anunciante aceitar apenas os pacotes *RaL* e *EstC* oriundos do mesmo peer. A Figura 4.17 ilustra as diferenças nos processos de ingresso do peer e transferência de arquivos em compartilhamentos restritos e de acesso público.

4.6.5 Transferência de dados

Para efetuar esta função, o LP2P-CORE realiza uma consulta junto ao LP2P-BD com objetivo de determinar uma fonte do conteúdo. Essa definição ocorre com base no menor identificador (Endereço IP do peer) não acionado até o momento dessa nova solicitação - de forma similar a uma fila circular contendo os peers ativos. A partir dessas informações, o módulo gerenciador define para qual peer será solicitada a transferência do recurso desejado, fazendo o envio dessa requisição por meio do módulo LP2P-MSG.

A transferência do arquivo ocorre em decorrência da interação do usuário através de um módulo cliente LP2P. Dessa forma, o LP2P-CORE tem a incumbência de coordenar o transporte do arquivo entre o peer de origem a o cliente LP2P, realizando a entrega ao módulo cliente responsável pela solicitação.

Quando a relação de transferências atinge os parâmetros de replicação preestabelecidos, o arquivo é gravado de forma definitiva no disco local do peer. Adicionalmente, o módulo LP2P-BD é acionado para alterar a respectiva entrada na base de conhecimento, informando que esse conteúdo também está presente localmente. Na ocorrência de transferências com diversas fontes, os campos *índice* e *valor* são inseridos para delimitar o *byte* inicial, além da parcela que deverá ser transferida a partir daquele ponto.

4.6.6 Monitoramento

Alterações no catálogo de arquivos ocorrem constantemente no decorrer das atividades efetuadas pelo usuário. Ações como exclusão, inclusão e alteração de arquivos demandam constante observação para garantir a consistência dos indexadores presentes em cada peer. Tal conformação determina que o LP2P-CORE monitore o sistema de arquivos local, emitindo uma resposta adequada a cada evento constatado.

Primeiramente, é efetivada a atualização da própria base de conhecimentos, na qual transcorre a interação do LP2P-CORE com o mantenedor desse recurso (módulo LP2P-BD). Em seguida, o gerenciador de mensagens é acionado para notificar os demais integrantes da ocorrência dessa alteração. Essa ação resulta no envio de mensagem ao grupo *multicast* apropriado.

Na outra extremidade da rede, cada participante receberá uma mensagem do tipo

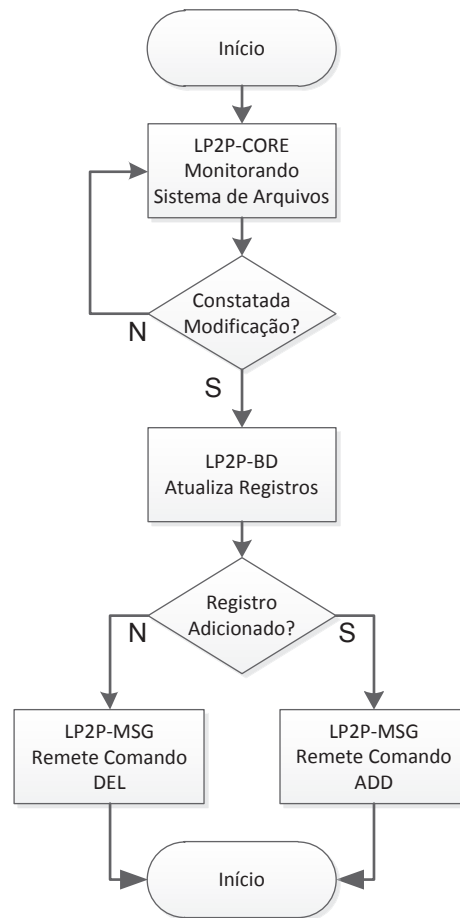


Figura 4.18: Fluxo de monitoramento

add ou *del*, dependendo da ação do *host* de origem. Em cada LP2P remoto associado ao compartilhamento, a mensagem é repassada a cada LP2P-CORE que, por sua vez, faz a *interface* junto ao LP2P-BD para consolidar essa atualização.

Nas ações relacionadas acima (ilustradas na Figura 4.18), apenas a quantidade de registros envolvidas pode variar entre cada notificação. O restante do processo é executado de forma idêntica, tanto na exclusão quanto na inclusão de arquivos. No entanto, a modificação do conteúdo demanda um tratamento específico. Neste caso, o identificador antigo é retirado da base local, ou seu conteúdo é alterado, de forma a retirar o referido peer da relação de usuários. Após, o novo conteúdo é registrado. E, finalmente, é enviada em sequência a mensagem do tipo *del* e *add*.

4.6.7 Envelhecimento do Cache

Proporcionar disponibilidade ao conteúdo ou apenas tornar seu acesso mais eficiente são os benefícios diretos obtidos pelo sistema LP2P ao empregar técnicas de replicação. A flexibilidade apresentada por este fator permite que determinada cópia do arquivo ocorra mesmo fora das condições normais de uso. Como observado no processo de anúncio, os

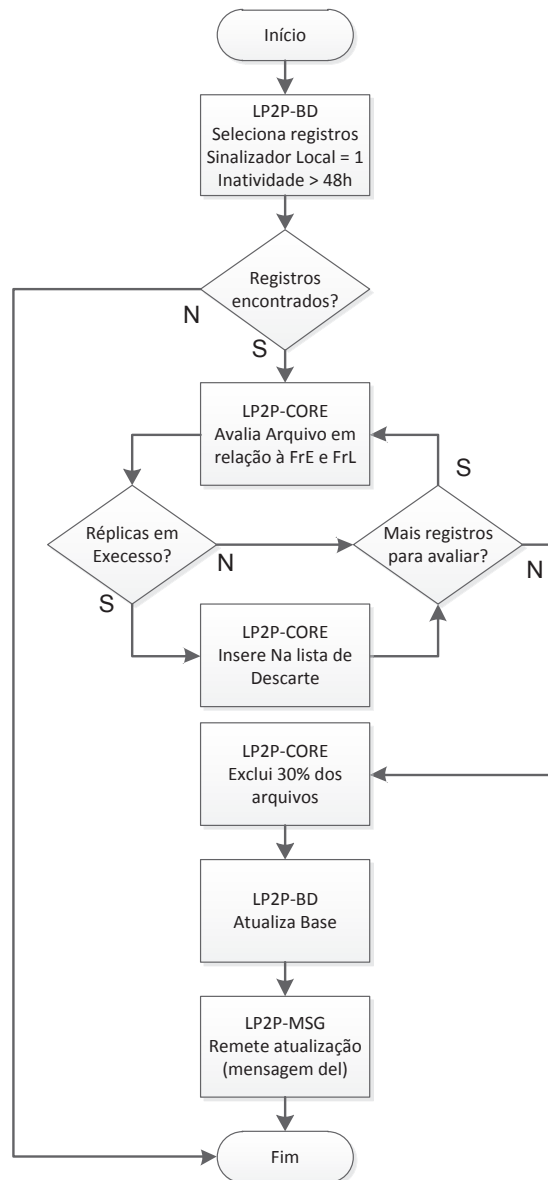


Figura 4.19: Fluxo de tratamento do Cache

fatores limitantes são determinados pelo coeficiente máximo de replicação (FrL) associado à condição de transferência (Ct).

O LP2P-CORE realiza esse reconhecimento, validando após cada recebimento de arquivo os quesitos de replicação do diretório. Esta metodologia prioriza o atendimento do usuário com o intuito de reduzir o consumo de banda ao evitar que determinado conteúdo trafegue inúmeras vezes para atingir o mesmo destino.

A preeminência do usuário implica, na maioria das vezes, na criação de réplicas acima dos valores estabelecidos ao repositório LP2P. Várias situações podem ocasionar um comportamento predatório na busca de arquivos, o que justificaria uma replicação acima do esperado. Por exemplo, um novo conteúdo inserido na rede desperta o interesse imediato dos usuários, as réplicas são criadas e, com o passar do tempo, o acesso diminui e novas

réplicas não são mais realizadas - a não ser em casos especiais (relatado na fase de anúncio).

Consequentemente, cabe ao LP2P-CORE lidar com tal situação, evitando uso de armazenamento dos peers de forma desnecessária. Neste contexto se insere o mecanismo de envelhecimento do *cache*, apresentado na figura 4.19. Primeiramente, o gerenciador do LP2P seleciona da base de conhecimento todos os arquivos configurados com valor **1** no campo *Sinalizador Local* e com inatividade maior a 48 horas. A partir dessa lista, cada arquivo é ponderado de acordo com os valores de replicação esperados (*FrE*) e do compartilhamento (*FrL*), resultando uma nova tabela contendo aqueles que estão acima do valor de referência.

Desse rol de registros, o peer executa o descarte de 30% dos arquivos, todos escolhidos aleatoriamente. Para encerrar, a relação de itens excluídos é repassada ao gerenciador de mensagens, enviando uma notificação do tipo *del* ao grupo *muticast* apropriado.

4.6.8 Comunicação Interna

Trata-se da última fase do processo de ativação do peer. Após ingressar no compartilhamento, receber e divulgar seus registros, o peer aciona sua interface de comunicação interna para atender às solicitações oriundas dos módulos clientes. Nesse instante, toda estrutura LP2P está ativa e apta ao atendimento do usuário.

Este capítulo apresentou o conjunto de definições e módulos pertencentes ao LP2P. Conforme enfatizado ao longo do mesmo, este modelo está em constante aperfeiçoamento, e permite a introdução de novas funcionalidades através de sua estrutura flexível de primitivas e mensagens. Nos próximos capítulos são relatadas avaliações reais, bem como teóricas de seu funcionamento.

5 METODOLOGIA E VALIDAÇÃO

Ao longo deste trabalho, foi enfatizado que sua principal meta é definir e desenvolver um modelo de compartilhamento de arquivos que incorpore características inerentes à comunicação P2P em um sistema focado em ambientes locais. A fim de avaliar este modelo, foram realizados testes do LP2P em ambiente de produção, juntamente com análise do impacto que o conjunto de mensagens do protocolo causa na rede.

Para executar os experimentos práticos, foi necessário implementar o protótipo do LP2P seguindo o cerne de suas especificações. Visando tornar os testes mais completos sob o aspecto da arquitetura LP2P, também foi implementado uma *interface* de usuário seguindo as premissas destinadas ao desenvolvimento do módulo cliente. Dessa maneira, foi possível acompanhar nos experimentos todo processo de requisição originado pelo usuário e o atendimento correspondente.

Na implementação do protótipo foi adotada a linguagem JAVA. Essa definição objetiva atender aos requisitos de portabilidade definidos na fase de planejamento do sistema LP2P. No módulo cliente, que provê a interação entre usuário e compartilhamento, foi utilizada a linguagem C. Tal escolha ocorreu em virtude do uso desse cliente como módulo do *kernel* em sistemas Linux, que empregam essa linguagem.

Os testes do protótipo foram realizados em dois cenários distintos, abrangendo desde pequenas transferências de arquivos até a interação do LP2P com aplicações reais para distribuição de conteúdo. Em cada ambiente, o módulo LP2P realizou a carga e divulgação dos arquivos, ilustrado pelas fases de *Inicialização*, *Anúncio* e *Atualização*. Esse comportamento proporcionou que, ao final do processo, todos os componentes da rede visualizassem o conjunto completo de arquivos compartilhados.

Ao módulo cliente, agregado ao cenário de testes como *front-end* entre usuário e LP2P, foi atribuída a função de obter os registros inseridos na base de conhecimento do diretório LP2P e apresentá-los ao usuário. Além disso, todo o processo de transferência de arquivos teve início a partir de requisições originadas na *interface* desse módulo. Essas tarefas seguiram exclusivamente a relação de mensagens inseridas no grupo de primitivas de comunicação interna.

Para fins de avaliação, o sistema de mensagem foi depurado detalhadamente, verificando o custo total e de cada etapa da comunicação LP2P. Essa análise teve como resultado um conjunto de cálculos que visam extrair o consumo de banda relacionado à execução do LP2P. Logo, nesses cenários estão contempladas medidas que envolvem desde pontos específicos do protocolo até o conjunto completo de mensagens necessárias para a transferência de arquivos.

5.1 DESCRIÇÃO DAS MÉTRICAS

A relação de experimentos e avaliações transcorridas neste capítulo visa analisar o protocolo LP2P diante dos seguintes aspectos:

- **Vazão Individual:** Define-se como vazão individual $V(x)$, o número médio de pacotes entregues com sucesso ao destino por unidade de tempo.
- **Vazão Agregada:** É o somatório da vazão de dados de todos os *hosts* do sistema em análise, sendo representada por:

$$V_{agg} = \sum_{x=1}^H [N(x) \cdot V(x)] \quad (5.1)$$

Onde:

H: *Hosts* conectados ao sistema.

- **Tempo de resposta:** Mede o tempo transcorrido desde o início de uma solicitação até sua finalização.
- **Overhead do protocolo:** Este fator é constituído pela quantidade de dados que são necessários na gestão e controle de uma ação, mas que são invisíveis para o usuário. No LP2P, as informações de propagação de arquivos ou anúncio do *peer*, que não são de interesse dos usuários, são classificadas como *overhead*, assim como ocorre com dados das camadas 4 (*sessão*), 3 (*rede*) e 2 (*enlace*).

As premissas para avaliação do modelo, representadas no conjunto de métricas relacionadas acima, obedeceram aos seguintes métodos de comunicação e transferência:

- **Comunicação ponto a ponto LP2P:** O módulo LP2P é executado em dois *peers*. Neste caso, uma das máquinas disponibiliza o conjunto de arquivos na pasta compartilhada, enquanto a outra realiza consultas e solicitações de forma similar à comunicação cliente-servidor.
- **Comunicação multiponto LP2P:** Neste método de conexão o módulo LP2P é acionado em todos os participantes, gerando uma topologia de conexão lógica fortemente conectada, similar às redes *full-mesh*. Todos os *hosts* atuam como clientes e servidores simultaneamente.

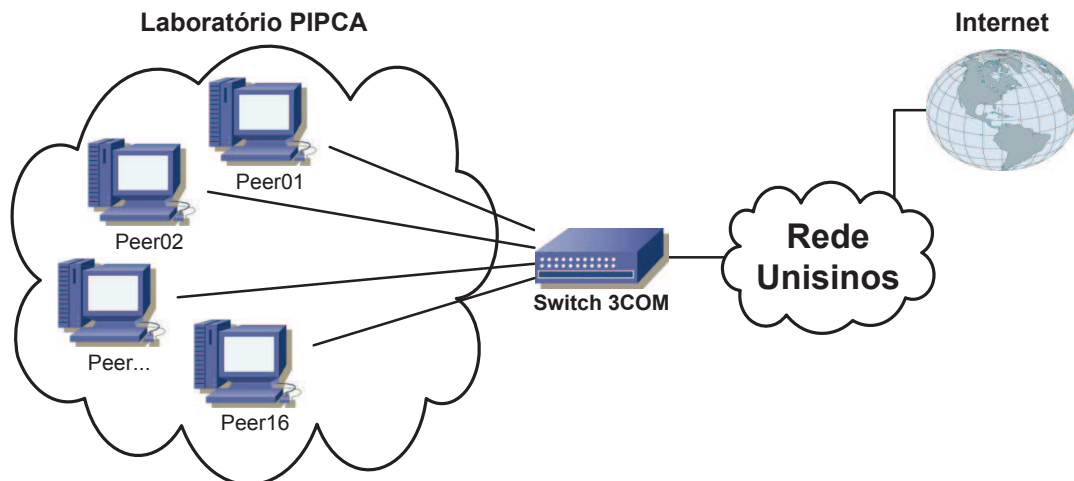


Figura 5.1: Topologia de testes LP2P

5.2 TOPOLOGIA DOS EXPERIMENTOS

Para avaliar o protocolo em ambiente de produção, foi montada a topologia de testes ilustrada na Figura 5.1, englobando a verificação das métricas de vazão individual e agregada ao tempo de resposta. Esta estrutura de rede conta com tecnologia FastEthernet e *switch* 3COM 10/100/1000, viabilizando a conexão dos *peers* a uma velocidade de 100Mbps na LAN e dispendo de uma saída para Internet através de conexão dedicada.

Um conjunto de 16 computadores, localizados no laboratório de uso geral do PIPCA, foi conectado ao *switch* 3COM para completar a topologia de testes LP2P. Cada equipamento utiliza sistema operacional Linux (distribuição Debian Lenny com *kernel* 2.6.26-2-686) e conta com 4GB de memória RAM, 250GB de HD e processadores Intel Core2 Duo de 1.8GHZ.

A escolha da distribuição de Linux Debian ocorreu como consequência de sua estabilidade. Visando obter maior confiabilidade na execução dos procedimentos de avaliação, optou-se por empregar a versão de *kernel* 2.6.26-2-686, pois pertence à distribuição estável¹ do Debian.

5.3 CENÁRIO 1 - TESTES BÁSICOS

O contexto deste primeiro cenário de testes envolve a transferência de arquivos de forma direta entre os *peers* da topologia. O cliente LP2PFS (ROCHA et al., 2010) foi empregado para interagir com o sistema LP2P, enviando solicitações para o recebimento

¹As versões da distribuição Linux Debian são classificadas em estável, teste e instável. Para ambientes de produção, os desenvolvedores desta distribuição recomendam o uso da versão estável. Maiores informações em: <http://www.debian.org/releases/index.pt.html>

Tabela 5.1: Resultados obtidos em arquivos de 1MB, 8MB e 16MB

Tamanho do Arquivo	Tempo de Execução	Vazão	
		Individual ($V(x)$)	Agregada (V_{agg})
1 MB	2,543s	6,291 MB/s	106,950 MB/s
8 MB	15,178s	8,041 MB/s	136,703 MB/s
16 MB	30,769s	8,320 MB/s	141,442 MB/s

de arquivos da rede. Tendo em vista que o objetivo era avaliar o protocolo LP2P, o módulo cliente foi configurado de forma a não influenciar esta análise.

Na sequência de testes, foram executadas transferências ponto a ponto e multiponto para multiponto de arquivos com tamanhos de 1MB, 8MB e 16MB, sendo realizados da seguinte maneira: cada *peer* presente na rede lê todos os arquivos de um mesmo tamanho inseridos no compartilhamento, entretanto em ordens diferentes. Por exemplo, o *peer1* da rede lê os arquivos do *peer2* até o *peer16* e, por fim, lê o seu próprio arquivo; já o *peer2* lê o arquivo do *peer3* até o *peer16*, lê do *peer1* e, por último, lê o seu próprio arquivo, e assim sucessivamente.

Essa linha de ação proporcionou que, em qualquer instante de tempo do teste, cada *peer* estivesse simultaneamente recebendo um arquivo solicitado e remetendo outro para atender uma requisição, evitando assim o comportamento do tipo exame (ROCHA et al., 2009) nesta topologia de avaliação. É importante salientar que, no momento de cada execução, o diretório LP2P já estava ativo e com todos os *hosts* conectados.

Além das operações de transferência, ao final de cada execução verificou-se a integridade dos dados com o algoritmo de *hash* MD5. Desse modo, a integridade dos arquivos presentes no compartilhamento foi conferida. Ressalta-se que esse procedimento foi executado com objetivo de fornecer mais subsídios para comprovar a transferência correta dos dados, podendo ser dispensado em termos práticos uma vez que o LP2P usa TCP nas operações de transporte de arquivos.

5.3.1 Avaliação do cenário 1

A vazão individual obtida por cada *peer* da rede, juntamente à vazão agregada do sistema e ao tempo de *download* transcorrido em cada tamanho de arquivo são ilustrados na Tabela 5.1.

Nos testes com arquivos de 16MB/s, foi atingido o maior valor de vazão individual de um *peer*: 8.320 MB/s. Com os resultados dos 3 testes, é possível comprovar que a

vazão também é influenciada pelo tamanho do arquivo requisitado, já que em arquivos de 1 *megabyte*, a vazão máxima foi de 6.291MB/s, chegando a 8.041MB/s com arquivos de 8 *megabytes* e atingindo o máximo de 8.320MB/s com arquivos de 16 *megabytes*.

Não foram mostrados testes para arquivos maiores que 16 *megabytes*, devido ao fato de que com arquivos de 16 *megabytes* já é possível saturar a rede com dados. Aliado a isso, outro parâmetro relevante presente na Tabela 5.1 é a vazão agregada do sistema, também influenciada pelo tamanho do arquivo utilizado nos testes.

Entretanto, nos testes do sistema com 16 *peers*, não foi possível atingir uma vazão individual de cada *peer* acima de 10 MB/s, o que seria o esperado neste cenário que utiliza conexões de 100Mbps. Após investigação, concluiu-se que esse comportamento foi decorrente do uso da versão 7 do Java, que não estava estável, além do módulo LP2PFS não executar o conjunto de funcionalidades que o cliente deve oferecer. Esses fatores resultaram em maior tempo de atendimento e reduziram, dessa maneira, o valor da vazão individual.

5.4 CENÁRIO 2 - APLICANDO O LP2P EM AMBIENTE REAL

Nesta seção, um cenário possível para utilização da plataforma proposta é contemplado, com o objetivo de ilustrar o potencial do sistema e suas respectivas vantagens. Dentre as aplicações possíveis deste modelo, a integração do LP2P a plataformas de atualização de *software* é apresentada, abordando as técnicas atuais para este fim e como o modelo pode facilitar esta operação.

5.4.1 Sistemas de atualização de *software*

Entre as várias ferramentas destinadas a reduzir ações prejudiciais no ambiente computacional, uma política clara de atualização dos sistemas operacionais desempenha um papel de extrema importância em face das constantes vulnerabilidades descobertas diariamente.

Hoje, muitos sistemas oferecem a seus usuários a capacidade de baixar atualizações de *software* automaticamente da Internet. Por exemplo, a família Windows da Microsoft executa tal operação de forma *on-line* através da plataforma Windows Update, enquanto a distribuição Linux Debian utiliza a ferramenta *apt-get* para esta finalidade.

Ao executar uma atualização por intermédio dessas ferramentas, não existem preocupações significativas decorrentes deste procedimento. Contudo, executar tal ação em um grande parque de máquinas de um laboratório ou rede empresarial impactaria de forma significativa no consumo dos *links* de Internet instalados nessas redes.

A fim de minimizar tal efeito, cada sistema tem a opção de formar espelhos na rede

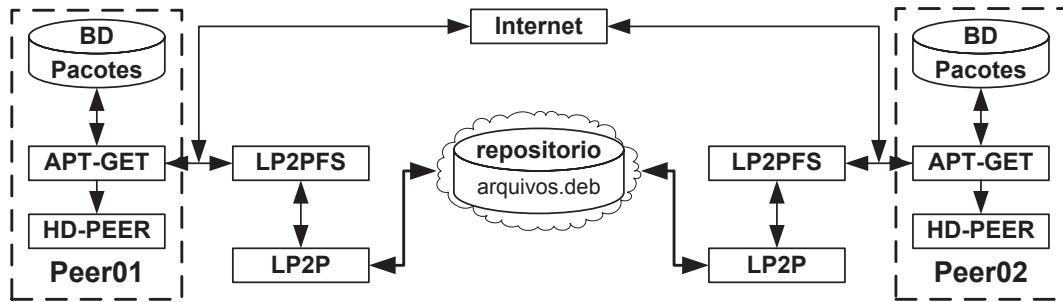


Figura 5.2: Integração do LP2P e apt-get

interna, realizando o *download* externo apenas uma vez. Exemplos dessas aplicações são o WSUS² para a plataforma Microsoft e *apt-cacher*³ empregado em sistemas Linux. A desvantagem dessas soluções é o uso de servidores dedicados que representam pontos centrais de falha e requerem investimentos tanto no *hardware* de servidores quanto na infraestrutura de rede.

Neste contexto, o LP2P oferece uma alternativa que proporciona redução no consumo da largura de banda combinado a um sistema de atualização distribuído e descentralizado, em conformidade com os objetivos do presente modelo de comunicação. Ao utilizar o LP2P, qualquer computador na rede que realiza uma atualização via Internet disponibiliza automaticamente este conteúdo aos demais integrantes da rede. Visando avaliar este processo detalhadamente, as próximas seções descrevem a utilização do *apt-get* integrado ao modelo LP2P.

5.4.2 Integrando LP2P e apt-get

Como ilustra a Figura 5.2, o LP2P está habilitado em cada estação de trabalho com a pasta `/var/cache/apt/archives/` como a fonte de arquivos a serem compartilhados. Além disso, por meio de um cliente dedicado VFS do Linux, *LP2PFS*, cada participante tem a pasta `/DEBS/repositorio` como ponto de montagem do sistema de arquivos local onde o LP2P disponibiliza os arquivos compartilhados. Esse ponto de montagem é inserido no arquivo de configuração do sistema Debian APT como um dos repositórios de *software* a ser consultado (usando a sintaxe `file:/// URI`). Após a ativação deste cenário, no qual todos os participantes visualizam os pacotes disponíveis no repositório compartilhado, o processo de atualização segue a seguinte ordem de execução:

- **Inicialização:** O *daemon* LP2P é executado em cada *peer*;
- **Criação da base de informações:** Cada *peer* publica aos demais os pacotes que dispõem localmente (por exemplo, como resultado de atualizações anteriores), a

²<http://technet.microsoft.com/en-us/wsus/default.aspx>

³<http://freshmeat.net/projects/aptcacher/>

fusão de todas as listas publicadas forma o conjunto de pacotes disponíveis no compartilhamento;

- **Atualização de pacotes:** Quando um *host* tenta instalar um certo pacote, duas situações podem ocorrer:
 - O pacote solicitado não está disponível localmente (porque nenhum *peer* fez essa instalação anteriormente). Neste caso o APT não irá encontrar o pacote no *cache* local (*/var/cache/apt/archives*) e, conseqüentemente, irá fazer o *download* diretamente da Internet. Uma vez que os pacotes são armazenados no *cache* local, o arquivo baixado será adicionado ao compartilhamento LP2P. Na ocorrência de uma nova tentativa de instalar o mesmo pacote (a partir de qualquer outro *host*), este será encontrado na rede local, evitando novo *download* de uma fonte externa.
 - O arquivo de atualização solicitado já está disponível no repositório LP2P: Neste cenário, o pacote solicitado é copiado do ponto de montagem (como definido na configuração pela sintaxe `file:/// URI`) para o *cache* local e torna-se disponível para instalação.

Deve-se observar que cada *host*, ao executar o processo de atualização ou instalação de novos pacotes, realiza cópia local do arquivo de instalação. Esse comportamento redundante é intrínseco ao funcionamento da aplicação APT, tornando-se muito útil ao sistema compartilhado. Se algum usuário decidir excluir os arquivos presentes em seu *cache* local, outros membros da rede ainda poderão fornecer o mesmo conjunto de pacotes. Apenas no caso de pacotes presentes em 1 *peer* essa proteção não será possível, sendo necessário baixar novamente o referido pacote da Internet.

O cenário apresentado acima evidencia o potencial do sistema LP2P para a redução da utilização da rede. Na próxima seção, são inseridos alguns resultados decorrentes dessa integração (LP2P e *apt-get*), aplicados à topologia de testes da Figura 5.1.

5.4.3 Avaliação do cenário 2

As implementações dos protótipos do LP2P e do módulo LP2PFS foram inseridas neste cenário, visando obter uma prova de conceito, bem como uma avaliação quantitativa referente ao desempenho do LP2P no cenário descrito anteriormente.

Foram avaliadas duas situações: *i*) atualizações de pacotes em único *host* (operação *dist-upgrade*), não dispondo de arquivos armazenados previamente no *cache*, o que corresponde a uma situação normal do *apt-get*; *ii*) um segundo *host* também executa uma atualização, mas se beneficia dos *downloads* realizados pelo primeiro *peer*.

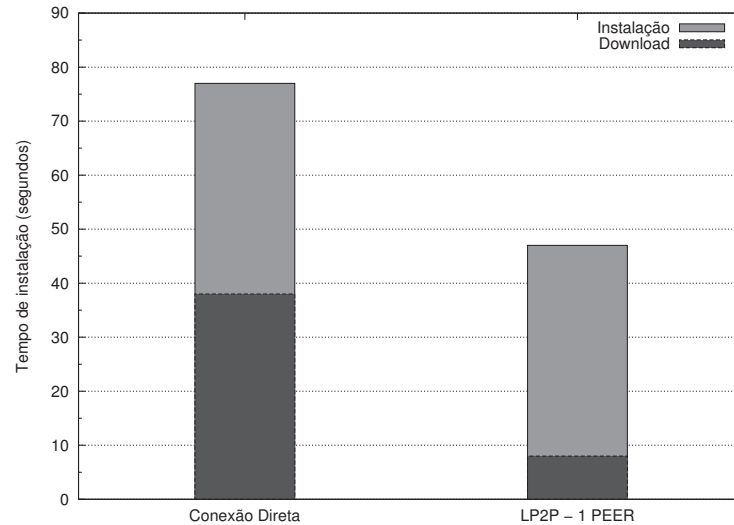


Figura 5.3: Tempo para atualização do sistema

Como ilustrado na Figura 5.3, a sequência de testes mediu o tempo para atualizar as estações Debian nos dois cenários descritos anteriormente. A barra *Conexão Direta* corresponde à primeira situação, e, como esperado, apresenta o pior desempenho, levando cerca de 77 segundos para sua conclusão (contemplando tempo de *download* de 39 segundos e instalação dos pacotes de 38 segundos). Na segunda situação, o tempo de *download* é reduzido em cerca de 79,49%, pois os pacotes podem ser obtidos localmente (tempo médio de 8 segundos), representando economia do *link* e agilidade no processo de atualização.

5.5 AVALIAÇÃO DE CUSTO DO LP2P

Com o objetivo de complementar esta prova de conceito, foram realizados testes de *download* com arquivos de tamanhos variados. Também foram feitos os cálculos do *overhead* de cada mensagem do modelo. Essas ações têm como objetivo, respectivamente, avaliar o tempo gasto na operação de transferência e verificar a eficiência de cada mensagem enviada através da rede. A eficiência, neste caso, é a *relação entre dados de controle e informação útil*.

Quanto ao *download*, foram realizados testes com arquivos de tamanhos variados de forma direta com 1 *host*, um conjunto de 5 *hosts* e através da rede LP2P composta por 5 *peers*. Os tempos médios de cada *download* são apresentados na Tabela 5.2, para arquivos de 25MB a 100MB.

Os resultados obtidos representam o primeiro *download* realizado da Internet mais as transferências subsequentes que ocorreram na comunicação entre *peers* LP2P. No caso do grupo de 5 *hosts*, o conteúdo foi transferido 5 vezes, o que resultou em maior tempo de consumo da rede. Comparativamente, o LP2P obteve o melhor custo-benefício entre os

Tabela 5.2: Tempo de Download - Conexão direta e LP2P

Tamanho do Arquivo	Tempo de <i>Download</i>		
	1 Host	5 Hosts	LP2P
25 MB	24,35s	121,75s	32,35s
50 MB	48,71s	243,55s	64,71s
75 MB	73,07s	365,35s	97,07s
100 MB	97,43s	487,15s	129,43s

cenários, pois disseminou o conteúdo para o maior quantidade de máquinas, dispensando de tempo proporcional de consumo do *link* externo.

Para definir o *overhead* do sistemas, avaliou-se primeiramente os quesitos gerais de custo do protocolo. Estão incluídos neste conjunto a estrutura básica do pacote LP2P, o nome do compartilhamento e informações referentes ao registro de arquivos e proprietários. Sendo assim, segue a relação de parâmetros genéricos juntamente de suas avaliações:

- **Fórmula geral custo**

Relação que verifica a parcela de bytes fixos e variáveis de uma mensagem LP2P para validar seu respectivo custo, sendo definida da seguinte maneira:

$$CmL = CfP + CvP + CfOp + CvOp \quad (5.2)$$

Onde:

CmL: Custo da mensagem LP2P;

CfP: Custo fixo do protocolo;

CvP: Custo variável do protocolo;

CfOp: Custo fixo da mensagem;

CvOp: Custo variável da mensagem

- **Estrutura do Protocolo**

Contabiliza apenas as informações fixas do formato geral de mensagem, mais o nome do compartilhamento envolvido. Sendo assim, este conjunto de informações poderá variar de 17 a 520 bytes. Destes, 7 carregam as informações fixas do LP2P (*Identificador, versão, identificação da primitiva e operação*), enquanto que os demais representam o nome do compartilhamento;

- **Informações do proprietário:**

O registro de um proprietário poderá ocupar 15 ou 16 bytes de um pacote LP2P.

Tabela 5.3: Relação de Custo - Estrutura LP2P / Proprietários / Arquivos

Recurso	Custo (Bytes)		
	Otimista	Médio	Pessimista
Estrutura Protocolo	17	260	520
Informação Proprietário	15	15	16
Registro do Arquivo	58	280	560

Nas mensagens de adição (*ADD*) e exclusão (*DEL*) de conteúdo, são utilizados 15 bytes para carregar a identificação do proprietário. Por outro lado, a mensagem *SENDL* insere ao conteúdo de cada arquivo a lista de usuários que detêm este recurso, tornando necessário o uso de um byte adicional como delimitador entre os diferentes donos do arquivo. Logo, o custo máximo que este recurso irá inserir na rede será de 16 bytes.

- **Registro de arquivo**

Dentre os campos que compõem o registro de um arquivo na base de conhecimento LP2P, apenas o nome do arquivo apresentará variação de 10 a 512 bytes. Os demais campos (4 delimitadores de 1 byte, *ID_FILE* de 20 bytes, Tamanho e Time-Stamp de 12 bytes) somados necessitam de 48 bytes para execução de suas funções;

Com base nestes dados foi possível verificar que as mensagens LP2P podem apresentar três tipos de comportamento – *otimista*, *médio* e *pessimista*. A Tabela 5.3 relaciona este aspecto aos fatores avaliados anteriormente, fornecendo assim as premissas iniciais para dar continuidade ao processo de avaliação que é apresentado a seguir:

- **NEPL**

O pacote responsável pela notificação de erros no protocolo LP2P dispõe de estrutura fixa de 4 bytes, destinados ao delimitador da mensagem, *ID_Primitiva*, *ID_Operação* e *ID_Erro*. Sendo assim, o tamanho das mensagens deste tipo é fortemente influenciado pelo nome do compartilhamento.

- **GET**

A estrutura desta mensagem não dispõem de campos variáveis, inserindo 48 bytes para execução de suas tarefas;

- **SENDF**

Os dados do usuário, indicados no campo Payload de *SENDF*, não estão incluídos no custo da mensagem. Sendo assim *SENDF* apresenta a mesma carga imposta por *GET*;

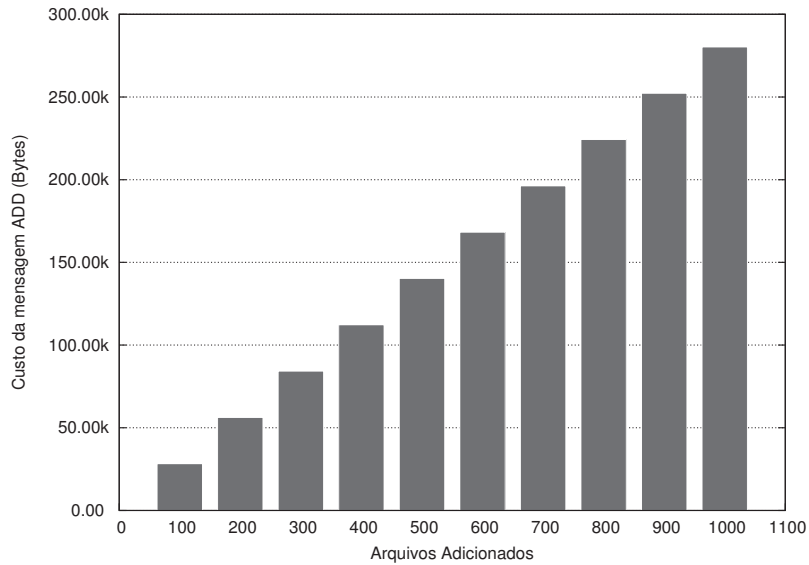


Figura 5.4: Custo da mensagem ADD em relação a quantidade de registros

- **ADD**

Esta mensagem tem como valor fixo apenas os dados referentes ao proprietário que notifica a inserção de novo conteúdo. Juntamente com este fator, a quantidade de arquivos afetará o tamanho deste pacote a medida que uma quantidade maior de arquivo for inserida no sistema. Adaptando-se a equação 5.2 e considerando o caso médio para os dados referentes ao protocolo e registros de cada arquivo, é possível estimar o custo da mensagem ADD através da seguinte equação:

$$CmADD = CfP + CvP + CfOp + \sum_{x=1}^{Qa} [R(x)] \quad (5.3)$$

Onde:

CmADD: Custo da mensagem ADD;

Qa: Quantidade de arquivos a inserir

R(x): Registro a inserir

A Figura 5.4 ilustra diversos cenários de inserção de arquivos, considerando o valor médio de cada registro a ser inserido no diretório LP2P.

- **DEL**

A estrutura da mensagem de exclusão possui similaridade com o comando destinado a adição de arquivos. No entanto, o conjunto de informações de cada arquivo, inseridos no pacote do tipo *DELL*, é muito menor em relação a mensagem *ADD*. Para notificar cada exclusão, o LP2P insere 21 bytes no pacote de notificação. Conforme pode ser observado na Figura 5.5, uma mensagem do tipo *DEL* é em média 10 vezes menor do que uma mensagem do tipo *ADD*;

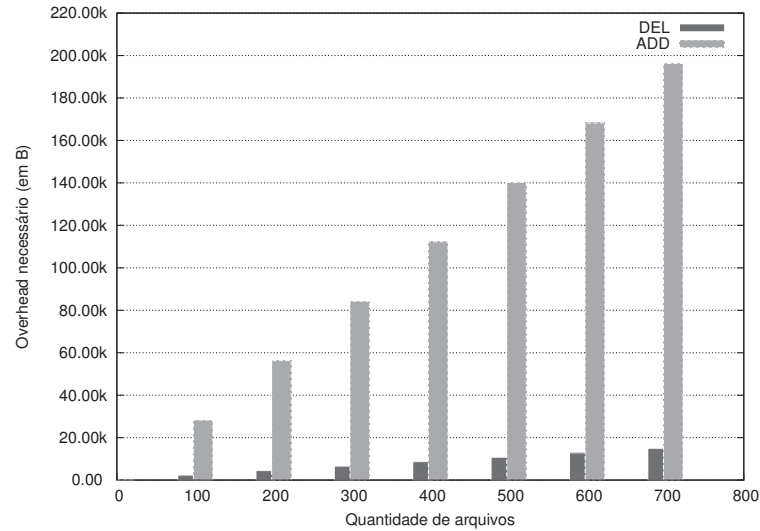


Figura 5.5: Comparação entre custo dos comandos DEL e ADD

- **LIST**

Nesta mensagem, apenas o campo contendo o critério de busca terá valor variável de 1 a 512 bytes. Os demais campos somados representam um total de 6 bytes;

- **SENDL**

Trata-se da mensagem de maior complexidade ao sistema LP2P, influenciada pela relação entre arquivos e quantidade de proprietários por arquivo. Esta dependência aliada ao restante de dados fixos do protocolo pode ser normatizada pela seguinte expressão:

$$CmSendl = Cp + \sum_{m=1}^{Qa} [CfA + (CfP * Qp)] \quad (5.4)$$

Onde:

CmSendl: Custo da operação *SENDL*;

Cp: Custo do protocolo. Tendo em vista sua baixa influência na mensagem *SENDL* este fator concatena a parcela fixa e variável do protocolo.

Qa: Quantidade de arquivos para inserir;

CfA: Custo fixo por arquivo;

CfP: Custo fixo por proprietário;

Qp: Quantidade de proprietários por arquivo;

Desta forma, o pacote *SENDL* contempla o conjunto de arquivos presentes na rede, juntamente com relação de proprietários atribuída a cada arquivo. Logo, os fatores que vão influenciar no custo desta mensagem são respectivamente o número de proprietários por arquivo (múltiplo de 16 bytes) e o custo por arquivo, tendo este valor médio de 280 bytes. A figura 5.6 ilustra o impacto destes fatores em *SENDL*.

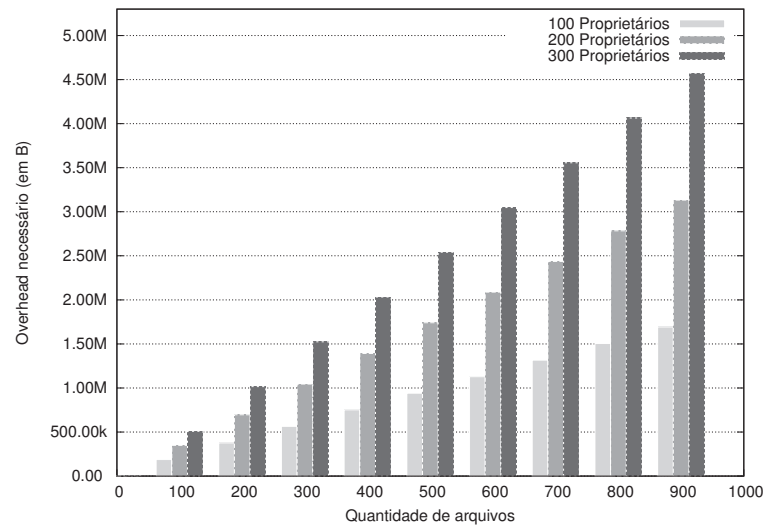


Figura 5.6: Custo SENDL - Relação de proprietários por arquivo

Conforme relatado ao longo do capítulo, o *overhead* relaciona apenas os dados de controle do sistema de mensagens LP2P. Sob o aspecto do usuário, este cálculo remete a soma dos 7 bytes fixos do protocolo, delimitadores de mensagem, informações do compartilhamento, arquivos e proprietários.

Dessa forma, existem duas situações de interesse do usuário. A primeira contempla os passos necessários para ingresso do peer, atualização da base de informações do compartilhamento, a requisição e o recebimento de arquivos. Na segunda, o peer já está conectado ao diretório LP2P e necessita apenas solicitar e receber arquivos.

Nestes cenários, o maior índice de *overhead* irá sempre ocorrer no processo de inicialização do peer, devido a sincronização de arquivos que já estão cadastrados na pasta compartilhada e os novos inseridos pelo peer que ingressa no sistema.

Considerando o custo das mensagens, o tamanho dos arquivos que serão transferidos ao longo da participação do peer e o meio de comunicação da rede local, é possível afirmar que o custo da mensagem é mínimo frente a parcela de dados de interesse do usuário.

6 CONCLUSÃO

O LP2P *Local Peer-to-Peer Protocol*, cuja especificação e desenvolvimento foram tratados neste trabalho é uma proposta para compartilhamento P2P de arquivos voltada aos ambientes locais. O desenvolvimento deste sistema foi motivado pela popularidade dos sistemas P2P em redes geograficamente distribuídas e do conseqüente interesse do meio acadêmico por este tipo de tecnologia aliados ao alto custo de aquisição e manutenção das soluções de armazenamento centralizado em redes locais presentes em ambientes como universidades e empresas.

Nestes ambientes, existe uma demanda constante por armazenamento que contrasta com o desperdício da capacidade de armazenamento disponível nas estações de trabalho. É importante destacar que além disso, o compartilhamento de arquivos fornecido de forma centralizada representa ponto único de falha e pode ocasionar gargalos da rede, impactando na disponibilidade do serviço.

Neste contexto, o LP2P representa uma alternativa ao uso de *hardware* específico para armazenamento em ambientes LAN, uma vez que este sistema adapta alguns conceitos do paradigma P2P à infraestrutura de rede local, aproveitando-se das características específicas destas redes, tais como alta vazão e baixas latências de comunicação.

Para fins de validação, o protótipo do LP2P e o módulo cliente foram implementados com uso das linguagens de programação JAVA e C, respectivamente. Estes dois softwares foram utilizados na verificação do custo total e de cada etapa da comunicação via LP2P, tendo como resultado um conjunto de cálculos que fornecem o consumo de banda associado ao uso sistema.

Com o objetivo de avaliar o protocolo LP2P em ambiente de produção, foi montada uma topologia constituída de 16 computadores localizados no laboratório de uso geral do PIPCA, conectados a um *switch* e fazendo uso da tecnologia FastEthernet. Neste teste foi feita a verificação das métricas de vazão individual e agregada ao tempo de resposta.

Os testes realizados abrangeram dois cenários: no primeiro deles, o LP2P foi utilizado para a transferência de arquivos de forma direta entre os peers da topologia, onde em todos os instantes cada um dos peers recebia um arquivo solicitado e enviava outro arquivo para atender uma requisição, a fim de evitar o comportamento conhecido como enxame. No segundo cenário, o LP2P foi testado integrado a ferramenta *apt-get* (que é utilizada em sistemas Linux), com base na premissa de que ao utilizar o LP2P, qualquer computador na rede que faz uma atualização através da Internet disponibiliza automaticamente este conteúdo aos demais peers. Assim, para um pacote solicitado que não está disponível localmente (no diretório LP2P) o APT realiza o *download* diretamente da Internet e,

posteriormente, quando outro peer solicitar o mesmo pacote, o APT irá copiá-lo diretamente do diretório LP2P.

Na comparação entre os casos em os pacotes solicitados por um host estavam presentes no diretório LP2P (pelo fato de que algum peer já havia realizado o download da Internet anteriormente) e aqueles em que o APT necessitou fazer o download de pacotes via Internet, o uso do LP2P possibilitou a redução no tempo de instalação de 79,49% em relação ao tratamento convencional. Além disso, frente a outras soluções de cache, como por exemplo o apt-cacher, o LP2P apresenta a vantagem de ser distribuído, o que contribui significativamente para o aumento da disponibilidade dos conteúdos.

O pico do custo protocolo LP2P ocorre no processo de ingresso do peer na rede, onde é realizada a sincronização das bases de conhecimento, a fim de unificá-las. No entanto, este custo é reduzido nas demais solicitações, o que significa que em média o LP2P exerce baixo *overhead* na rede.

No que diz respeito aos custos econômicos, o LP2P permite o aproveitamento do *hardware* ocioso existente nas estações de trabalho, reduzindo assim a necessidade de investimento em servidores e repositórios centrais de armazenamento (ex: storage). Como o LP2P permite que o fator de replicação seja configurado por compartilhamento, os conteúdos mais críticos disponibilizados, por exemplo, em um ambiente corporativo podem ter sua disponibilidade assegurada pelo uso de um alto fator de replicação, enquanto que os arquivos menos críticos podem ter um grau de replicação menor, possibilitando dessa forma que o administrador da rede possa gerenciar a ocupação do disco nas estações de trabalho.

Com as características apresentadas ao longo deste trabalho, é possível concluir que o LP2P constitui uma alternativa de baixo custo ao emprego de *hardware* dedicado ao armazenamento em ambientes LAN. Este sistema pode fornecer uma arquitetura de *backup* secundário funcionando paralelamente à solução principal, proporcionando agilidade na recuperação de informações em comparação ao sistema convencional. Além disso, esta solução seria capaz de concatenar logicamente os recursos de armazenamento ociosos presentes em cada máquina para fornecer compartilhamentos públicos.

6.1 TRABALHOS FUTUROS

Como trabalhos futuros, podem ser desenvolvidos mecanismos para controlar os direitos de acesso (leitura e escrita) e a exclusão permanente de arquivos, visto que atualmente, um arquivo só é totalmente excluído se ele é apagado em todos os peers que o contêm.

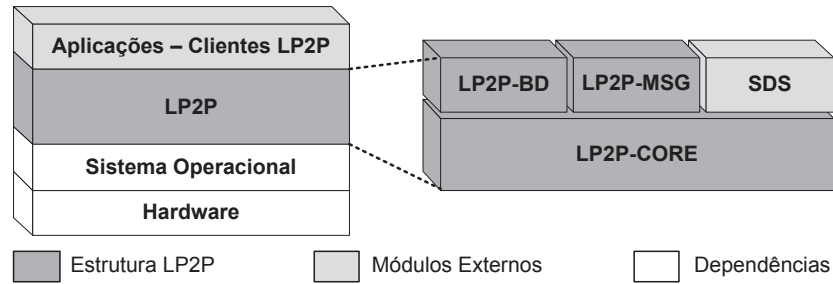


Figura 6.1: Arquitetura LP2P com descoberta de serviço integrada

Adicionalmente, o LP2P podem ser integrados a sistemas para descoberta de serviços (conforme ilustra a Figura 6.1), como por exemplo o Jini ¹, que é uma solução baseada em JAVA ou ainda, ferramentas baseadas em DNS multicast (mDNS) (CHESHIRE; KROCHMAL, 2011b) e DNS-Based Service Discovery (DNS-SD) (CHESHIRE; KROCHMAL, 2011a), como é o caso do Bonjour (BONJOUR... , 2006) e do Avahi².

¹<http://www.jini.org>

²<http://avahi.org/>

BIBLIOGRAFIA

- ABERER, K. et al. The essence of P2P: A reference architecture for overlay networks. In: *PROC. OF THE 5TH IEEE INTERNATIONAL CONFERENCE ON PEER-TO-PEER COMPUTING – P2P2005*. Konstanz, Germany: IEEE, 2005. p. 11–20.
- ABERER, K.; HAUSWIRTH, M. An Overview on Peer-to-Peer Information Systems. In: *WDAS*. Paris, France: Carleton Scientific, 2002.
- ABIONA, O. et al. Architectural model for wireless peer-to-peer (wp2p) file sharing for ubiquitous mobile devices. In: *Electro/Information Technology, 2009. eit '09. IEEE International Conference on*. [S.l.: s.n.], 2009.
- AHMED, D. T.; SHIRMOHAMMADI, S. A hybrid P2P protocol for real-time collaboration. In: *Proc. IEEE Workshop on Collaborative P2P Information Systems*. Manchester, UK: IEEE Computer Society, 2006. p. 73–78. ISBN 0-7695-2623-3.
- ANDERSON, T. E. et al. Serverless network file systems. *ACM Trans. Comput. Syst.*, ACM, New York, NY, USA, v. 14, p. 41–79, February 1996. ISSN 0734-2071. Disponível em: <<http://doi.acm.org/10.1145/225535.225537>>.
- ANDROUTSELLIS-THEOTOKIS, S.; SPINELLIS, D. A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys*, ACM, New York, NY, USA, v. 36, n. 4, p. 335–371, 2004. ISSN 0360-0300.
- BARCELLOS, M. P.; GASPARY, L. P. Fundamentos, tecnologias e tendências rumo a redes p2p seguras. In: PUC-RIO (Ed.). *Jornada de Atualização em Informática*. Rio de Janeiro, RJ, Brasil: [s.n.], 2006. v. 1, p. 187–243. Minicursos.
- BARHAM, P. et al. Xen and The art of Virtualization. In: *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2003. p. 164–177. ISBN 1-58113-757-5.
- BISNIK, N.; ABOUZEID, A. A. Optimizing random walk search algorithms in p2p networks. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 51, n. 6, p. 1499–1514, 2007. ISSN 1389-1286.
- BONJOUR Overview. 2006. Informação técnica sobre o sistema Bonjour. Disponível em: <<http://developer.apple.com/mac/library/documentation/cocoa/Conceptual/NetServices/Articles/about.html>>. Acesso em: janeiro de 2011.
- BOUKERCHE, A.; ZARRAD, A.; ARAUJO, R. B. A cross-layer approach-based gnutella for collaborative virtual environments over mobile ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, IEEE Computer Society, Los Alamitos, CA, USA, v. 21, p. 911–924, 2010. ISSN 1045-9219.
- BUFORD, J. F.; YU, H.; LUA, E. K. *P2P Networking and Applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 2009. ISBN 978-0-12-374214-8.

CESHIRE, S.; KROCHMAL, M. *DNS-Based Service Discovery*. fev. 2011. Internet Draft, Internet Engineering Task Force (IETF). <http://files.dns-sd.org/draft-cheshire-dnsext-dns-sd.txt>.

CESHIRE, S.; KROCHMAL, M. *Multicast DNS*. fev. 2011. Internet Draft, Internet Engineering Task Force (IETF). <http://files.multicastdns.org/draft-cheshire-dnsext-multicastdns.txt>.

COMER, D. E. *Interligação em rede com TCP/IP volume 1*. 3rd. ed. [S.l.]: Editora Campus, 2001. 672 p. ISBN 85-352-0270-6.

CONSTANTINESCU, C.; PIEPER, J.; LI, T. Block size optimization in deduplication systems. In: *Data Compression Conference, 2009. DCC '09*. [S.l.: s.n.], 2009. p. 442–442. ISSN 1068-0314.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. *Distributed Systems: Concepts and Design*. Boston, MA, USA, 4th: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321263545.

DABEK, F. et al. Wide-area cooperative storage with cfs. *SIGOPS Oper. Syst. Rev.*, ACM, New York, NY, USA, v. 35, p. 202–215, October 2001. ISSN 0163-5980. Disponível em: <<http://doi.acm.org/10.1145/502059.502054>>.

DEERING, S. *RFC 1075: Host extensions for IP multicasting*. [S.l.], ago. 1989. Disponível em: <<http://www.rfc-editor.org/rfc/rfc1112.txt>>.

DHUNGEL, C. Z.; ZHENGYE, P. D. W.; K.W., L. R. Bittorrent darknets. In: *INFOCOM, 2010 Proceedings IEEE*. [S.l.: s.n.], 2010.

DOMINGUES, A. et al. *Peer-to-Peer: Computação Colaborativa na Internet*. Gramado, RS, Brasil: [s.n.], Maio 2004. SBRC. Minicursos.

DOWNES, K. et al. *Internetworking Technologies Handbook, Second Edition*. [S.l.]: Cisco Press, 2000. ISBN 85-352-0663-9.

ESTRIN, D. et al. *RFC 2117: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*. [S.l.], jun. 1997. Disponível em: <<http://www.rfc-editor.org/rfc/rfc2117.txt>>.

HEIDELBERG, S. B. . (Ed.). *Kademlia: A Peer-to-Peer Information System Based on the XOR Metric*. [S.l.: s.n.], 2002. ISBN 978-3-540-44179-3.

HETHMON, P. *RFC 3659: Extensions to FTP*. [S.l.], mar. 2007. Disponível em: <<http://www.rfc-editor.org/rfc/rfc3659.txt>>.

HUANG, H.; WU, H.; WANG, G. Study of distributed p2p information sharing system. *Intelligent Information Technology Applications, 2007 Workshop on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 2, p. 300–303, 2009.

IYER, S.; ROWSTRON, A.; DRUSCHEL, P. Squirrel: a decentralized peer-to-peer web cache. In: *Proceedings of the twenty-first annual symposium on Principles of*

distributed computing. New York, NY, USA: ACM, 2002. (PODC '02), p. 213–222. ISBN 1-58113-485-1. Disponível em: <<http://doi.acm.org/10.1145/571825.571861>>.

IZAL, M. et al. Dissecting bittorrent: Five months in torrent's lifetime. In: *Lecture Notes in Computer Science*. [S.l.: s.n.].

LEVIN, D. et al. Bittorrent is an auction: analyzing and improving bittorrent's incentives. In: *In SIGCOMM*. [S.l.: s.n.], 2008.

LI, Z.; XU, Z.; FENG, X. Design and implement of file sharing system with identity certification based on cl-pkc in p2p networks. In: *Anti-Counterfeiting Security and Identification in Communication (ASID), 2010 International Conference on*. [S.l.: s.n.], 2010.

LUA, E. K. et al. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, v. 7, p. 72–93, 2005.

LUO, X. et al. An interest-based P2P file sharing system. In: *Proc. of Communications and Networking in China, 2009. ChinaCOM 2009. Fourth International Conference. CHINACOM*. Xian, NJ, USA: IEEE Press, 2009. p. 1–5. ISBN 978-1-4244-4337-6.

MCNAMARA, G.; YANG, Y. Creating a mobile p2p file sharing environment over bluetooth. In: *Pervasive Computing and Applications, 2008. ICPCA 2008. Third International Conference on*. [S.l.: s.n.], 2008. v. 2, p. 863–868.

MCPHERSON, D. *ATLAS Internet Observatory 2009*. 2009. Relatório Técnico disponibilizado pelo grupo Atlas. Disponível em: <http://www.nanog.org/meetings/nanog47/presentations/Monday/Labovitz_ObserveReport_N47_Mon.pdf>. Acesso em: nov. 2010.

MILOJICIC, D. et al. *Peer-to-Peer Computing*. Palo Alto, CA, USA, 2002.

MILOJICIC, D. S. et al. *Peer-to-Peer Computing*. [S.l.], 2002.

MOCKAPETRIS, P. *RFC 1035: Domain names - implementation and specification*. [S.l.], nov. 1987. Disponível em: <<http://www.rfc-editor.org/rfc/rfc1035.txt>>.

MOLIN, K. *Measurement and Analysis of the Direct Connect Peer-to-Peer File Sharing Network*. Tese (Doutorado) — University of Gothenburg, June 2009.

MOY, J. RFC, *RFC 1584: Multicast Extensions to OSPF*. mar. 1994. Disponível em: <<http://www.rfc-editor.org/rfc/rfc1584.txt>>.

MUTHITACHAROEN, A. et al. Ivy: a read/write peer-to-peer file system. *SIGOPS Oper. Syst. Rev.*, ACM, New York, NY, USA, v. 36, p. 31–44, December 2002. ISSN 0163-5980. Disponível em: <<http://doi.acm.org/10.1145/844128.844132>>.

NAKAZAWA, H. et al. A method for controlling search in a pure p2p file sharing network. In: *Information and Telecommunication Technologies (APSITT), 2010 8th Asia-Pacific Symposium on*. [S.l.: s.n.], 2010.

NI, Y.; NYANG, D.; WANG, X. A-Kad: an anonymous P2P protocol based on Kad network. IEEE, p. 747–752, 2009.

OHZAHATA, S.; KAWASHIMA, K. A study on traffic characteristics evaluation for a pure p2p application. In: *Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on*. [S.l.: s.n.], 2008. p. 483–490. ISSN 1066-6192.

ORAM, A. (Ed.). *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2001. ISBN 0-596-00110-X.

PINHEIRO, M. *Uma Arquitetura P2P Baseada na Hierarquia do Endereçamento IP com Roteamento Unificado*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, Janeiro 2006.

PIRES, C. *Um Sistema P2P de Gerenciamento de Dados com Conectividade Baseada em Semântica*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Abril 2007.

POSTEL, J.; REYNOLDS, J. *RFC 0854: Telnet Protocol Specification*. [S.l.], maio 1983. Disponível em: <<http://www.rfc-editor.org/rfc/rfc854.txt>>.

RATNASAMY, S. et al. A scalable content-addressable network. In: *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2001. p. 161–172. ISBN 1-58113-411-8.

RIPEANU, M.; FOSTER, I.; IAMNITCHI, A. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system. *IEEE Internet Computing Journal*, v. 6, p. 2002, 2002.

ROCHA, E. S. et al. Análise matemática para avaliação de desempenho em ambientes peer-to-peer. In: *ERRC 2009 - PG*. [S.l.: s.n.], 2009.

ROCHA, É. S.; MARCON, D. S.; ÁVILA, R. B. Comunicação peer-to-peer aplicado a redes locais. In: *ESCOLA REGIONAL DE REDES DE COMPUTADORES, ERRC, 8.*, 2010, Alegrete, RS. *Anais...* [S.l.], 2010.

ROUSE, C.; BERMAN, S. A scalable p2p database system with semi-automated schema matching. In: *ICDCSW '06: Proceedings of the 26th IEEE International Conference Workshops on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2006. p. 78. ISBN 0-7695-2541-5.

ROY, S. et al. Design issues of the prtorrent file sharing protocol. In: *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*. [S.l.: s.n.], 2010.

SACHA, J. *Exploiting heterogeneity in peer-to-peer systems using gradient topologies*. 2009.

TAKESHITA, K.; SASABE, M.; NAKANO, H. Mobile p2p networks for highly dynamic environments. *Pervasive Computing and Communications, IEEE International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 453–457, 2008.

TANENBAUM, A. *Redes de Computadores*. 4th. ed. Rio de Janeiro, RJ, Brasil: Campus, 2003.

WAITZMAN, D.; PARTRIDGE, C.; DEERING, S. *RFC 1075: Distance Vector Multicast Routing Protocol*. [S.l.], nov. 1988. Disponível em: <<http://www.rfc-editor.org/rfc/rfc1075.txt>>.

WANG, G. ji; XUE, H. qiang; WEI, B. A self-stabilizing leader election algorithm in onehop dht. In: *Communications and Networking in China, 2009. ChinaCOM 2009. Fourth International Conference on*. [S.l.: s.n.], 2009. p. 1–5.

ZAHID, R. et al. A Survey of Peer-to-Peer Storage Techniques for Distributed File Systems. In: *Systems", International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II - Volume 02*. [S.l.]: IEEE Computer Society, 2005. p. 4–6.