

Chapter

1

Introdução e Características da Rede P2P

Vitor Rosenbergre dos Santos Carmo

Abstract

Nowadays, sharing information and developing an effective communication system is of great importance for a computer architecture, and a peer-to-peer (P2P) network ensures the perfect sharing of files and resources for the improvement of communication. A P2P network is where two or more computers are connected in a system so that they can access and share data without the mediation of a third party. This architecture discards the traditional server and client model and embraces a system where both parties have the same capability. This presentation makes mention of the main features and information of P2P networks, along with famous applications like Napster, KaZaA, BitTorrent, Gnutella and Skype.

Resumo

Hodiernamente, compartilhar informação e desenvolver um sistema de comunicação efetivo é de grande importância para uma arquitetura de computadores, e a rede peer-to-peer (P2P) garante o compartilhamento perfeito de arquivos e recursos para o melhoramento da comunicação em rede. Uma rede P2P é tecnologia a qual dois ou mais computadores são conectados em um sistema de forma que eles podem acessar e compartilhar dados sem a mediação de terceiros. Essa arquitetura descarta o modelo tradicional de servidor e cliente e abraça um sistema onde ambas as partes possuem o mesmo recurso. Esta apresentação faz uma menção às principais características e informações das redes P2P, junto com aplicações famosas como Napster, KaZaA, BitTorrent, Gnutella e Skype.

1.1 Introdução

Existem várias definições de sistemas que seguem a arquitetura de redes de computadores Peer-to-Peer (P2P) na literatura e não existe uma resolução precisa sobre esse sistema e quais suas principais características. A primeira noção seria de uma arquitetura distribuída descentralizada onde cada ponto possui funcionalidades e funções semelhantes. No entanto, essa definição excluiria sistemas considerados peer-to-peer, como o KaZaA e Napster.

Segundo Theotokis et al. 2004, a arquitetura peer-to-peer é uma espécie de aplicação que leva vantagem de recursos - ciclos de CPU, conteúdo, presença humana - disponíveis na Internet. A definição também é cabível a alguns sistemas com servidores centralizados e outras aplicações baseadas na Computação em Grade. Ademais, os sistemas P2P são chamados assim pois dão a impressão de fornecer uma interação direta entre computadores.

De acordo com Sadok, existem características essenciais na arquitetura P2P, que as distinguem de uma organização central ou hierárquica. Sendo elas:

- Localizar nós em qualquer lugar da rede sem manter um grande nível de roteamento, permitindo aplicações mais complicadas do que o simples compartilhamento de arquivo, incluindo sistemas multicast, sistemas de comunicação anônimas e caches da web.
- A conexão de um nó à rede é por meio de outro nó que já pertence ao sistema. Os nós possuem conectividade variável ou temporária, endereços variáveis e autonomia parcial ou total em relação a um servidor centralizado. Eles são conectados de forma aleatória, sem restrição sobre o número de nós participantes da rede e podem se unir e sair da rede sem permissão e sem aviso prévio. Em relação a comunicação, os nós tem que ser capazes de se comunicar diretamente. Assegurar aos nós capacidades equivalentes de fornecer e consumir recursos de seus pares.
- Uma rede escalável com habilidade de tratar instabilidades e tolerância a falhas.

Uma rede pode ser considerada peer-to-peer mesmo que algumas funções de controle da rede estejam localizadas em um ponto central (sistemas com características diferentes).

Com as características descritas por Sadok e a base dada por Theotokis et al. 2004, temos que sistemas P2P são redes distribuídas que consistem de nós interconectados capazes de se estruturar em sistemas de rede overlay para compartilhar recursos, habilitados a se adaptar a falhas, sem necessitar da mediação de uma entidade central.

1.1.2 Redes P2P e Redes Overlay

Redes são consideradas overlay quando construídas sobre uma rede computacional já existente, normalmente a Internet. Elas são sistemas distribuídos por natureza, sem

nenhuma organização hierárquica ou controle centralizado. A peer-to-peer overlay permite acesso aos seus recursos por outros sistemas e suporta o compartilhamento de recursos, que requer tolerância a falhas, auto-organização e propriedades de escalabilidade massivas.

De acordo com Anderson et al. 2001, a Internet pratica o paradigma overlay quando usa o protocolo IP como solução internetworking sobre tecnologias de redes diversas como ATM, Frame Relay, etc. Isso ocorre porque a rede cria uma arquitetura com nível mais alto de abstração, solucionando vários problemas que são difíceis de serem tratados ao nível dos roteadores da rede subjacente.

Uma Rede P2P pode ser considerada uma rede overlay, já que funciona como uma rede "virtual", por conta da sua alta abstração, formada pela interconexão de nós, executando sobre uma infra-estrutura de uma rede física (Figura 1.1).

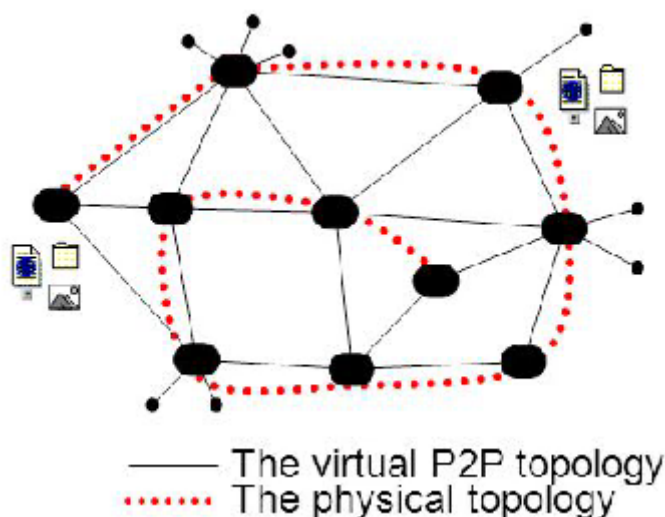


Figura 1.1: Topologia da Rede overlay

Como já mencionado na seção 1.1 e visto na Figura 1.1, a característica fundamental de uma rede peer-to-peer é a existência de um grupo de nós (peers) com interesses comuns conectados pelo mesmo sistema de comunicação.

1.1.3 Classificação de Modelos de Arquitetura P2P

Segundo Sadok, existem algumas categorizações e modelos de arquitetura de redes P2P. Na primeira categorização temos:

- Descentralizado - nesse modelo o controle e os dados são completamente distribuídos, ou seja, não existe um servidor central. Todos os nós são equivalentes, onde nenhum nó tem controle sobre outro conteúdo compartilhado entre outros usuários. O acesso à informação é executado diretamente entre os pares ou por meio de vizinhos em comum.

- Semi-centralizado - é uma combinação de arquiteturas centralizadas com distribuídas, onde existem os super-nós (*ultrapeers*, core nodes, gateways) e os nós ordinários. Os super-nós são normalmente mais poderosos, formam uma rede P2P descentralizada entre eles mesmos e cuidam dos pedidos de buscas no lugar dos clientes. Nós ordinários normalmente não possuem muita responsabilidade no sistema, já que apenas se conectam com os super-nós como clientes e carregam informações de índice sobre arquivos compartilhados para super-nós.

Numa outra classificação de redes P2P, conhecida por ser muito utilizada, temos três modelos. São eles:

- Busca centralizada (*centralized search*) - possui ao menos um servidor que fornece serviços de recursos. Eles mantêm a posição de listas de recursos disponíveis de todos os nós do sistema. Cada nó envia uma requisição pedindo o recurso desejado, classificando como uma das consultas. Para cada consulta, o servidor busca nas listas e responde um resultado para o nó requisitante. A resposta inclui a descrição dos recursos e dos nós responsáveis por eles. O nó requisitante escolhe um nó fornecedor e contata-o diretamente para realizar mais consultas.
- Busca por inundação (*flooding search*) - consiste em um grande número de nós independentes, cada um conectado a um pequeno subconjunto de nós e não a todos os nós da rede. Se um nó deseja encontrar um recurso na rede, que pode estar em um nó que não conhece, ele pode simplesmente transmitir sua consulta de pesquisa para seus vizinhos imediatos. Se os vizinhos não tiverem o recurso, ele solicitará que seus vizinhos encaminhem a consulta para seus vizinhos. Isso é repetido até que o recurso seja encontrado ou todos os nós tenham sido contatados, ou talvez um limite de salto imposto pela rede seja atingido.
- Busca por tabela *hash* distribuída (DHT) - tabela de hash distribuída armazena pares de valores-chave atribuindo chaves a nós autônomos diferentes para a busca. Cada nó armazenará os valores de todas as chaves pelas quais é responsável.

A categorização mais utilizada no ambiente acadêmico apresenta três modelos, sendo eles:

- Centralizada - a rede mantém possui um índice central com informações atualizadas. Sistemas de compartilhamento de arquivos como o Napster (seção 000) e de troca de mensagens usam esta arquitetura como base.

- Descentralizada e Estruturada - esta rede não possui um servidor centralizado, mas tem uma estruturação significativa entre os nós. Sua topologia de rede é controlada e os documentos e arquivos são postos em locais de fácil localização. Essa arquitetura é utilizada por vários sistemas, como o Chord, CAN e Pastry.
- Descentralizada e não Estruturada - diferente do outro modelo, esta rede não possui um servidor centralizado, e pouca estruturação e controle sobre a topologia e busca de documentos. A rede se baseia nos modelos descentralizado e semi-centralizado (da primeira categoria) e busca a inundação da segunda (nós independentes). As aplicações Gnutella e KaZaA são exemplos desta arquitetura de rede P2P.

Existem outras categorizações além das que foram apresentadas nesta seção, necessárias para outros casos de arquiteturas P2P. As aplicações que vão ser mostradas posteriormente ou utilizam algum desses modelos ou possuem características semelhantes.

1.2. Aplicações

De acordo com Sadok, os nós podem ser localizados, trazendo assim a conectividade para as bordas da rede e permitindo a comunicação e colaboração entre todos os computadores conectados. Com isso, a rede P2P possibilita um grande número de aplicações, como por exemplo o Gnutella, KaZaA, BitTorrent, Napster e Skype. Nesta seção, vamos descrever maiores detalhes sobre classes de aplicações responsáveis por compartilhamento de arquivos.

1.2.1 Compartilhamento de Arquivos

A transação de informações é uma das áreas onde a tecnologia P2P tem tido mais destaque. Várias aplicações são usadas por usuários da Internet para burlar as limitações de largura de banda dos servidores, que impedem a transferência de arquivos grandes.

Sistemas peer-to-peer que armazenam informação com a rede de P2P oferecem as várias vantagens, sendo algumas delas o fornecimento de uma área potencialmente ilimitada para o armazenamento de arquivos, a política de replicação múltipla do conteúdo, armazenando uma informação em mais de um nó na rede, a segurança das informações armazenados nos nós, evitando que seus dados sejam vazados e a localização e recuperação dos conteúdos armazenados na rede.

1.2.1.1 Napster

O Napster pode ser descrito como a aplicação gatilho que tornou as redes peer-to-peer web populares, já que os sistemas conhecidos como ftp e arquivos compartilhados não possuíam casos de uso, protocolos comuns, parâmetros e a escalabilidade de um sistema peer-to-peer.

Servidores Napster agem tanto como clientes quanto servidores na troca de arquivos de música. Um servidor primeiro se junta à rede pela conexão a um servidor central chamado broker. Após a conexão, o hospedeiro passa toda a informação para o broker. Essas informações são conhecidas como metadados. O broker armazena um

banco de dados dos metadados. Esses metadados contêm as informações de todos os hosts atualmente conectados ao broker. A figura 1.2 representa a conexão de sete computadores ao broker.

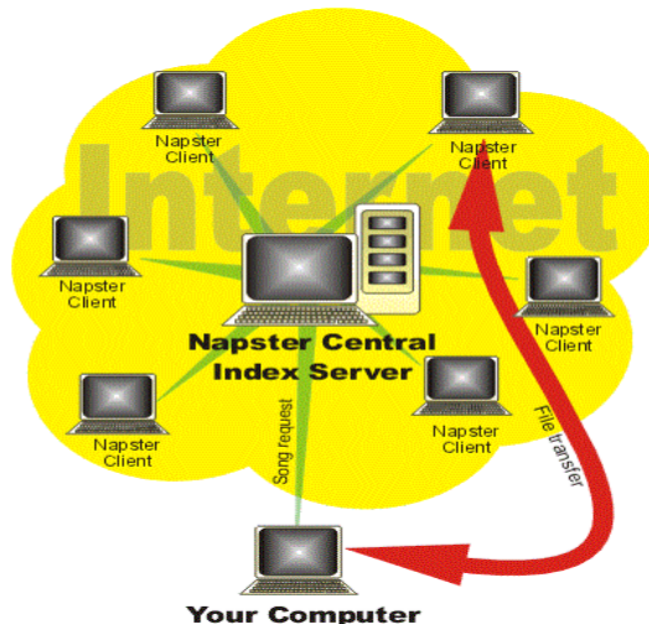


Figura 1.2: Arquitetura básica do Napster

Os clientes (*Napster Client*) consultam o banco de dados do broker em busca de arquivos de música específicos. O broker responde de volta com uma lista de músicas e pares correspondentes que as contêm. O cliente pode então coordenar com o broker a troca de um arquivo de um dos hosts remotos. Na Figura 1.2 o seu computador (*Your Computer*) está baixando um arquivo diretamente do *Napster Client* na nuvem. O broker participa na busca pelo arquivo e na configuração da troca de arquivos, no entanto, o download ocorreu diretamente entre os dois pares.

Além de pesquisar e compartilhar músicas, o Napster também oferece mensagens ponto a ponto, salas de bate-papo e listas de usuários quentes. As mensagens ponto a ponto permitem que um ponto fale para outro par. As salas de bate-papo permitem que grupos de usuários compartilhem informações. Uma mensagem postada em uma sala de bate-papo é vista por todos os usuários conectados à sala de bate-papo. As listas quentes contêm uma lista de pares populares com quem um cliente esteve em contato. Os pares podem adicionar uns aos outros para suas próprias listas quentes. Esta lista quente fornecerá informações sobre os metadados de um par, como bem como quando o par está online. O broker realiza toda a coordenação desses recursos extras. A interface do Napster é mostrado na Figura 1.3.

Devido à popularidade do Napster, muitos *brokers* do Napster estão disponíveis para lidar com milhões de usuários. Antes de agosto de 2000, esses *brokers* eram incapazes de compartilhar bancos de dados. Isso significava que cada broker continha uma rede Napster separada. Nesse mesmo ano, o Napster vinculou todos os *brokers* para que eles formassem uma grande rede, permitindo que qualquer usuário para consultar os metadados de todos os usuários conectados à rede Napster. Desde aquele tempo o Napster manteve seus *brokers* desvinculados a maior parte do tempo. Suspeita-se que

enquanto os brokers estavam interligados, a coordenação de informações entre os *brokers* era muita carga na rede.

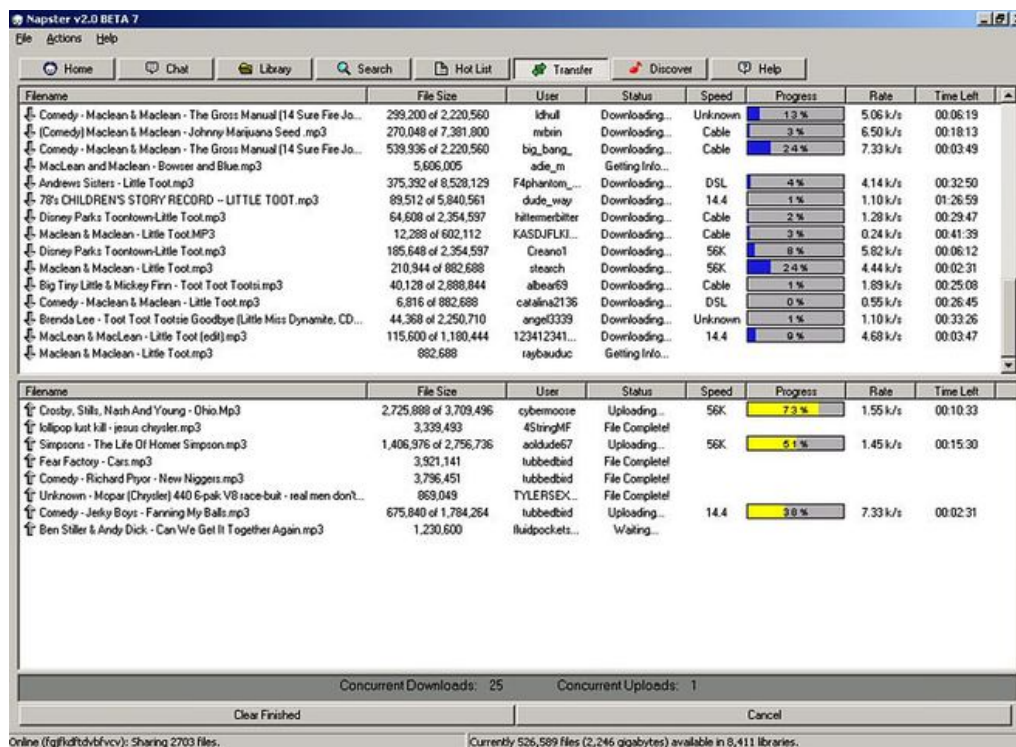


Figura 1.3: Interface do Napster

Em média, 871.000 usuários se mantinham na Rede Napster, com seu máximo de 1.179.433 usuários sendo atingido em 20 de novembro de 2000 às 16:00. Esse alto número de usuários contribuiu para a causa de desvincular todos os *brokers*.

1.2.1.4 Gnutella

O Gnutella é uma rede descentralizada que não possui um servidor central usado para coordenação. Uma vez que um hospedeiro Gnutella está conectado a uma rede Gnutella, ele atua como um nó coordenador que é livre para trocar arquivos com outros hospedeiros. Um hospedeiro Gnutella primeiro se junta a uma rede Gnutella conectando-se com ao menos um outro hospedeiro que se localiza na mesma rede. Para se obter arquivos, um cliente deve consultar a rede para descobrir onde os arquivos estão localizados.

Na Figura 1.4, o Computador (nó) A envia uma mensagem *search* para todos os seus computadores. Esta mensagem é propagada para outros computadores conectados. O *time-to-live* (TTL) em uma mensagem de consulta garantirá que uma mensagem não dure para sempre. O nó mais distante possível onde a mensagem de consulta morre é denominado como horizonte de pesquisa. Quando um computador contém arquivos que correspondem à mensagem *search*, ele envia uma mensagem de *search result* de volta

ao computador que requisitou o pedido. Depois que os resultados da pesquisa forem retornados, o cliente pode conectar-se diretamente e fazer download do ponto que carrega um arquivo desejado. Um exemplo está na Figura 1.4, quando o computador A baixa um arquivo diretamente do computador F.

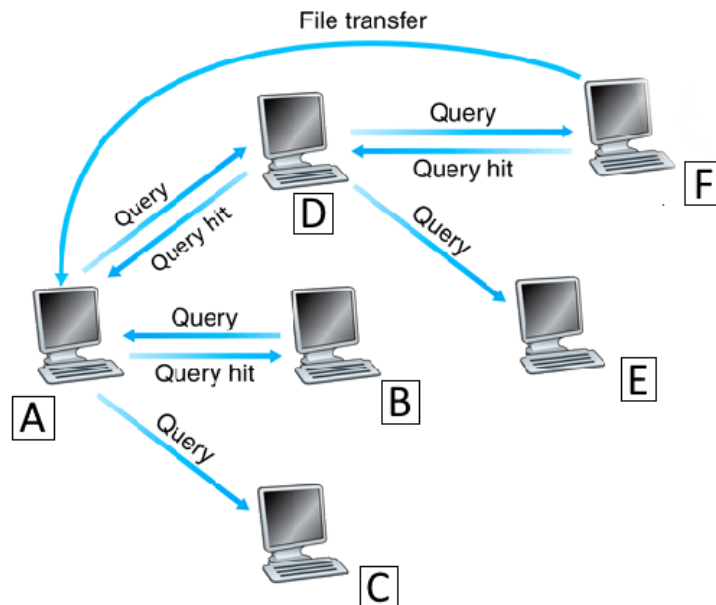


Figura 1.4: Uma Rede Gnutella

Existem cinco mensagens definidas para coordenação e descoberta de arquivos entre Gnutella Peers (nós): ping, resposta de ping, solicitação de ping do cliente, pesquisa e resultados de pesquisa. Todas as conexões entre peers Gnutella usam o protocolo de transporte TCP/IP. Um pequeno subconjunto de instruções HTTP são usadas para a inicialização de conexões de mesmo nível. Essas incluem as mensagens CONNECT, OK, GET e GIV. Suas características são:

- **CONNECT** - Com o endereço de outro servidor conectado à rede, uma conexão TCP é estabelecida com o mesmo através do envio de uma mensagem CONNECT. Esta mensagem é enviada na forma de uma string ASCII. Caso o computador que recebeu a mensagem deseja se conectar, o mesmo deve responder com a mensagem OK, qualquer resposta diferente indica que a conexão não será aceita. Exemplo da String: CONNECT/0.4\r\n.
- **OK** - O OK é uma mensagem que representa a aceitação da conexão com outro computador. Exemplo da String: OK\r\n.
- **GET** - O protocolo utilizado no Gutella para a transferência de arquivos é o HTTP. O servidor que está atuando como iniciador manda uma mensagem do tipo GET pedindo o arquivo, enviando o protocolo (HTTP) e a versão do protocolo para requisição. Exemplo da String: GET /get/<File Index>/<File Name>/HTTP/1.0\r\n.

- GIV - O GIV é a mensagem enviada pelo servidor que vai enviar um arquivo em resposta a uma requisição. O computador que responder estabelece uma conexão TCP no endereço IP e porta especificada. Enviando GIV <File>/<File Name>\n\n. Assim, o computador requisitando a informação vai enviar um GET e os dados vão ser transferidos.

O Gnutella é uma ideia simples de ser aplicada, mas possui pouca escalabilidade, desde que inundar um buffer com consultas desperdiça largura de banda. Algumas vezes, objetos existem mas não conseguem ser localizados devido a um limitado TTL. Dado a isso, várias estratégias de busca foram propostas para tentar solucionar estes problemas, como o *biased-random-walk* (passeio aleatório enviesado), onde cada computador (nó) grava o nível dos seus vizinhos, facilitando a busca em direção a nós de alto nível que possuem mais informações.

Essa crescente onda de popularidade revelou os limites da escalabilidade do protocolo inicial. No início de 2001, variações no protocolo melhoraram a escalabilidade. Em vez de tratar cada usuário como cliente e servidor, alguns usuários foram tratados como "ultra-pares" ou "supernós", roteando solicitações de pesquisa e respostas para usuários conectados a eles.

1.2.1.2 KaZaA

O KaZaA foi um dos primeiros sistemas P2P a explorar essa heterogeneidade organizando os pares em duas classes, Super Nodes (SNs) e Ordinary Nodes (ONs). Os SNs são geralmente mais poderosos em termos de conectividade, largura de banda, processamento e acessibilidade não NAT (Network address translation). Ademais, os SNs também têm maiores responsabilidades. Como visto na Figura 1.5, cada ON tem um SN pai. Quando um ON inicia a aplicação KaZaA, o ON escolhe um SN pai, mantém uma conexão TCP semipermanente com seu SN pai e carrega para este SN os metadados dos arquivos que está compartilhando.

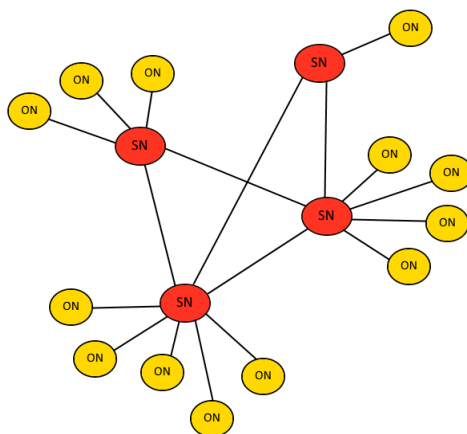


Figura 1.5: Uma rede KaZaA de duas camadas.

Como acontece com a maioria dos outros sistemas de compartilhamento de arquivos P2P, o KaZaA mantém um índice de arquivos que mapeia identificadores de arquivo para os endereços IP. Este índice de arquivo é distribuído entre os SNs. Em

particular, cada SN mantém um índice local para todos os seus ONs filhos, de modo que cada SN é semelhante a um hub Napster. Mas em contraste com o Napster, um SN não é um servidor dedicado, e sim um peer pertencente a um usuário individual. Para cada arquivo que um ON está compartilhando, os metadados que o ON envia para seu SN pai inclui: o nome do arquivo, o tamanho do arquivo, o *ContentHash*, e os descritores de arquivo. Os descritores de arquivo são usados para correspondências de palavras-chave durante a consulta. O ContentHash desempenha um papel importante na arquitetura, já que o KaZaA faz hashes a cada arquivo para uma assinatura de hash, que se torna o ContentHash do arquivo. O ContentHash é o único identificador usado para identificar um arquivo em uma solicitação de download HTTP. Se um download de um nó específico falhar, o ContentHash permite que o cliente KaZaA procure o arquivo específico, sem emitir uma nova consulta de palavra-chave.

Quando um usuário deseja localizar arquivos, o ON do usuário envia uma consulta com palavras-chave sobre a conexão TCP com seu SN pai. Para cada correspondência em seu banco de dados, o SN retorna o endereço IP, o número da porta do servidor e os metadados correspondentes à correspondência. Como mostrado na Figura 1.5, cada SN também mantém conexões TCP de longa duração com outros SNs, criando uma rede de sobreposição entre os SNs. Quando um SN recebe uma consulta, ele pode encaminhar a consulta para um ou mais dos SNs aos quais está conectado. Uma determinada consulta geralmente visitará um pequeno subconjunto dos SNs e, portanto, obterá os metadados informações de um pequeno subconjunto de todos os ONs.

1.2.1.3 BitTorrent

Nos últimos anos, o BitTorrent surgiu como um método popular de compartilhamento de dados. É um método bem efetivo, já que conta com grande porcentagem do tráfego de toda a Internet. Cada arquivo é distribuído em sua própria rede como um número de blocos de dados independentes. Um cliente pode compartilhar blocos de dados que foi totalmente baixado, mesmo que não tenha terminado de baixar o arquivo inteiro. Isso permite um paralelismo que é impossível se arquivos inteiros são tratados como blocos atômicos. O paralelismo é mais severamente testado em “flash crowd” (multidão instantânea) configurações quando um grande número de clientes se junta numa rede quase simultaneamente. Isso acontece na prática e cria uma grande demanda de downloads sem aumentar imediatamente a capacidade de upload do sistema.

Em geral, o BitTorrent distribui arquivos grandes, chamados torrents, cada um em uma rede separada. Um *torrent* é dividido em um grande número de blocos de dados menores, cada um geralmente entre 32K e 256K, e estes são então compartilhados de forma independente. Um cliente que não tenha baixado o arquivo inteiro ainda pode ter sido completamente baixado vários blocos, que podem ser enviados para outros clientes. Isso permite que os clientes compartilhem a carga de trabalho mesmo quando ainda estão baixando. Para cada torrent, há uma central correspondente chamada *tracker* (rastreador). Quando um cliente deseja baixar o torrent, ele entra em contato com o *tracker*. O *tracker* normalmente retorna um subconjunto aleatório de cinquenta clientes existentes, que chamamos de vizinhos do cliente. Um cliente também pode consultar o rastreador para novos vizinhos, o que geralmente acontece se o cliente ficar com menos

de vinte vizinhos (devido a desconexões, por exemplo). A qualquer momento, um cliente tenta fazer upload para cinco de seus vizinhos. Um desses vizinhos é escolhido aleatoriamente, enquanto os outros quatro são escolhidos usando um sistema *tit-for-tat*. Especificamente, um cliente tentará fazer upload para vizinhos dos quais está baixando mais rápido. Se o cliente terminou de baixar, ele carrega para o cliente para o qual ele pode carregar mais rapidamente. A Figura 1.6 mostra um pouco da arquitetura que consiste em um *tracker* centralizado e o arquivo *.torrent*.

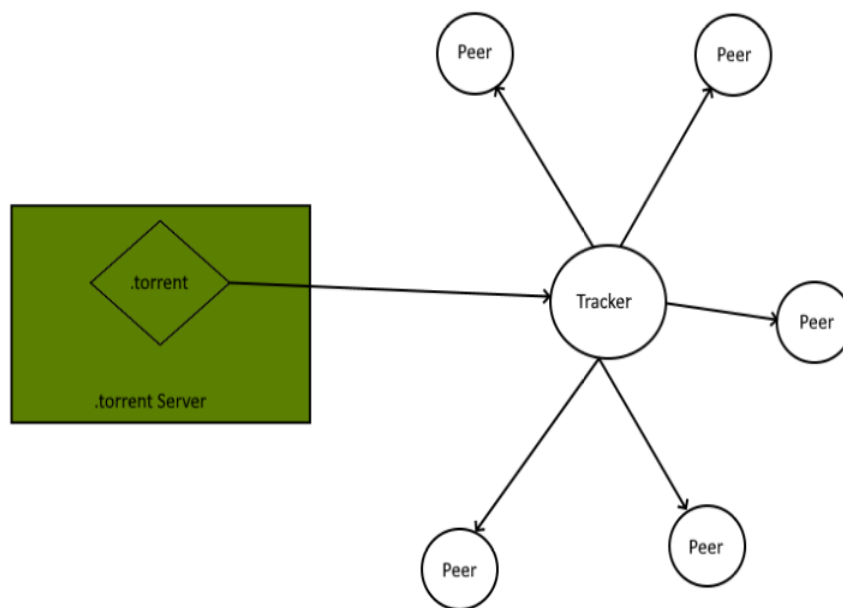


Figura 1.6: Arquitetura BitTorrent com *tracker* centralizado.

Uma vez que um cliente tenha decidido para quem deve fazer o upload, resta determinar quais blocos serão realmente transferidos. No BitTorrent, um cliente geralmente escolhe blocos de download que são menos replicados entre seus vizinhos.

Atualmente, o BitTorrent não suporta streaming de qualquer maneira. Ele usa uma política de download onde cada cliente escolhe um bloco de dados menos replicado entre seus vizinhos, o que em nada garante que os dados chegarão em qualquer aproximação do correto pedido. Por outro lado, todos os nossos resultados são válidos para qualquer política de download. Isso implica que se o BitTorrent mudou para a política intuitivamente inferior de sempre baixar blocos de dados em ordem. No entanto, surgiram alguns problemas práticos. Mais notavelmente, o último bloco se tornaria muito raro se os clientes saírem da rede assim que fizerem o download. A tentativa de resolver esses problemas pode auxiliar em trabalhos futuros.

1.2.1.5 Skype

O e-mail foi o aplicativo matador original para a Internet. Hoje, voz sobre IP (VoIP) e mensagens instantâneas (IM) estão complementando rapidamente o e-mail em empresas e redes domésticas. O Skype é um aplicativo que fornece esses serviços VoIP/IM em um pacote fácil de usar que funciona por trás de NAT/firewalls. Em outubro de 2005 o Skype se tornou um dos P2P mais populares.

O Skype oferece serviços pagos que permite usuários iniciarem e receberem chamadas por meio de números de telefone regulares por meio de gateways. Apesar de sua popularidade, pouco se sabe sobre os protocolos criptografados do Skype e a rede proprietária. Garfinkel conclui que o Skype está relacionado ao KaZaA, já que ambas as empresas foram fundadas pelos mesmos indivíduos, têm uma sobreposição de pessoal técnica, e que muito da tecnologia do Skype foi originalmente desenvolvido para o KaZaA. Análise de nível de pacote de rede do KaZaA e do Skype apoiam essa afirmação descobrindo semelhanças impressionantes em sua configuração de conexão e seu uso de uma rede peer-to-peer hierárquica baseada em supernós. As redes peer-to-peer baseadas em supernós organizam os participantes da rede em duas camadas – supernós e nós comuns. Normalmente, os supernós mantêm uma rede de sobreposição entre si, enquanto os nós ordinários escolhem um supernó para se associar. Supernós também funcionam como nós comuns e são eleitos entre eles com base em alguns critérios. Os nós comuns emitem consultas por meio do(s) supernó(s) aos quais estão associados. A Figura 1.7 ilustra a relação entre os nós ordinários (ON), supernós (SN) e o servidor de login. As arestas representam a relação entre os vizinhos.

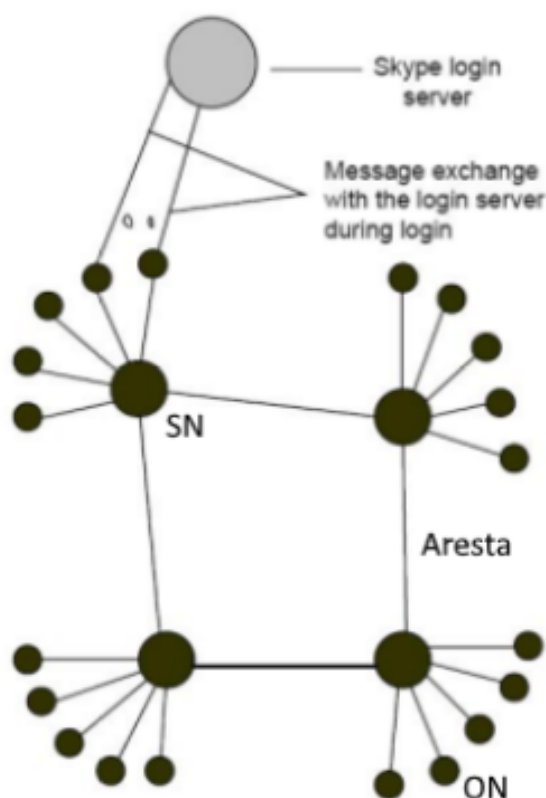


Figura 1.7: Rede Skype

No Skype, nós comuns enviam informações de disponibilidade, mensagens e solicitações de voice over IP (VoIP) e sessões de transferência de arquivos pela rede ponto a ponto do supernó. Se a solicitação de VoIP/transferência de arquivo é aceita, os clientes Skype estabelecem uma conexão direta entre si. Para determinar isso, executamos dois clientes Skype em hosts separados, e observamos os endereços IP de

destino e origem para pacotes enviados e recebidos em resposta a várias tarefas em nível de aplicativo. Repetimos o experimento para um único cliente por trás de um NAT, e ambos os clientes por trás de diferentes NATs. Observamos que, se um cliente está atrás de um NAT, o Skype usa a reversão de conexão em que o nó NAT inicia a sessão de mídia TCP/UDP (protocolo de transporte), independentemente de qual extremo solicitou o VoIP ou a transferência de arquivos sessão. Se ambos os clientes estiverem por trás de NATs, o Skype usa o percurso NAT tipo *Session Traversal Utilities for NAT* (STUN) para estabelecer a conexão. No caso de falha na travessia do NAT, o Skype volta para uma abordagem *Traversal Using Relays around NAT* (TURN) onde a sessão de mídia é retransmitida por um supernó, publicamente alcançável.

Um nó Skype atrás de um uplink de rede saturado e um nó atrás de um NAT não se tornaram supernós, enquanto um novo nó instalar em um host público com uma conexão de 10 mbps a Internet se juntou à rede do supernó em poucos minutos. A população de supernós selecionados pelo Skype, aparentemente baseada em acessibilidade e largura de banda sobressalente, tende a ser relativamente estável. O Skype, portanto, representa um ponto interessante no espaço de design P2P onde a heterogeneidade é aproveitada para controlar a rotatividade, não apenas lidar com ela.

1.3 Conclusão

As redes P2P não são novidade para os usuários mais antigos da Internet. A principal diferença das atuais redes P2P e do modelo original da Internet é a possibilidade de qualquer computador, inclusive aqueles instalados nas residências dos usuários, participarem ativamente da rede. Por isso, é esperado que com o amadurecimento da tecnologia, muitos tipos novos de redes P2P devam surgir. Aplicações críticas ainda não podem ser implementadas com o modelo P2P, uma vez que ele apresenta problemas de confiabilidade, devido à conectividade variável das máquinas participantes e à administração descentralizada. O modelo cliente/servidor permite que servidores de alta disponibilidade e segurança forneçam informações confiáveis, característica indispensável para o mundo corporativo. Para o usuário doméstico, ao contrário, frequentemente uma grande quantidade de informações é mais importante do que uma informação específica.

Considerando diferentes arquiteturas de sistemas peer-to-peer, os projetistas de sistemas devem avaliar os requisitos para suas aplicações particulares e escolher uma topologia para a plataforma que atende às suas necessidades. Para compará-los brevemente, podemos dizer que numa rede peer-to-peer cada peer recebe igual responsabilidade, independentemente de suas capacidades de computação/rede, levando a redução do desempenho à medida que os nós menos capazes são adicionados. Os sistemas P2P carecem de capacidade de gerenciamento, pois cada peer é seu próprio controlador. Esses sistemas quando não estruturados e com a busca cega de inundação não são vendáveis, pois em grandes sistemas, a quantidade de troca de mensagens limita a escalabilidade. Usando sistemas estruturados ou abordagens inteligentes de busca podem resolver a limitação de escalabilidade. A desvantagem dos sistemas estruturados padrão é que é difícil manter a estrutura necessária para a roteirização numa população transitória, na qual os nós se juntam e debandam em alta taxa.

Referências

- Andersen, D., Balakrishnan, H., Kaashoek, F. & Morris, R., “Resilient Overlay Networks”, 18th ACM Symposium on Operating Systems Principles, Banff/Canada, Outubro de 2001.
- Garfinkel, S. L. “VoIP and Skype Security”, Jan. 2005.
http://www.cs.columbia.edu/~salman/skype/SkypeSecurity_1_5_garfinkel.pdf.
Último acesso: 21/05/2022
- Napster Service Status. <http://www.napster.com/status/>. Último Acesso: 12/05/2022.
- Nilsson, Erik. Napster: “Popular Program Raises Devilish Issues”.
<https://www.semanticscholar.org/paper/Napster-and-Gnutella%3A-a-Comparison-of-two-Popular-Howe/d3d0b1783466ff951995df0af6e619e4d131d6e4>. Último acesso: 12/05/2022.
- N. Leibowitz, M. Ripeanu, and A. Wierzbicki, “Deconstructing the Kazaa Network” 3rd IEEE Workshop on Internet Applications (WIAPP’03), Santa Clara, CA, USA, June 2003.
- Sadok, Djamel, GT - “Computação Colaborativa (P2P)”, 2003.
https://memoria.rnp.br/_arquivo/gt/2003/p2p.pdf. Último acesso: 12/05/2022.
- Salman, A. Baset. and Henning G. Schulzrinne. “An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol”. Department of Computer Science Columbia University. New York, NY, USA. 2005.
http://www1.cs.columbia.edu/~salman/publications/skype1_4.pdf. Último acesso: 12/05/2022.
- Theotokis, S. A. and Spinellis, D. (2004). “A survey of peer-to-peer content distribution technologies”. ACM Computing Surveys.

Vídeo:

<https://drive.google.com/file/d/1a5p3XAZUfUobDVAISw7YBn-yZwIbhsnC/view?usp=sharing>