Redes e Sistemas Autónomos
Apontamentos
Parte I

Universidade de Aveiro

Sebastian D. González

Redes e Sistemas Autónomos
Apontamentos
Parte I

Dept. de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro

sebastian.duque@ua.pt(103690)

25 de junho de 2024

Warning!!

Isto são apenas uns apontamentos realizados por uma pobre alma de MIECT, feitas a partir dos slides da disciplina e outras fontes 😈. Por favor, não usem apenas estes apontamentos como material de estudo.

Dito isto, boa sorte a todos e ámen CT 🙏.

Agradecimentos à Prof. Susana Sargento susana@ua.pt por todo o material fornecido nas aulas.

Conteúdo

Glossário

CDM Centralized Directory Model.

CDN Content Delivery Network.

CHORD Circular Hashing of Resource Distribution.

DHT Distributed Hash Tables.

FRM Flooded Requests Model.

IPFS InterPlanetary File System.

NAT Network Address Translation.

VOIP Voice Over IP.

Peer-to-Peer Systems and Networks

As redes baseadas em IP tornaram-se a norma para as aplicações web-based em redes corporativas internas e muitas interações entre empresas. Houve uma grande aceitação e crescimento explosivo, o que resultou em sérios problemas de desempenho e experiência do utilizador.

Para um grande conjunto de aplicações, incluindo o acesso a vídeos, é necessário melhorar o desempenho das aplicações em rede, utilizando muitos sites(servidores independentes e routers) em diferentes pontos dentro da rede.

1.1 CDN

Content Delivery Network é uma rede de servidores interligados que aceleram o carregamento de páginas web, armazenando conteúdo perto dos utilizadores para diminuir o tempo de resposta na comunicação. Esta tecnologia tem evoluído ao longo do tempo, focando-se em acelerar a entrega de conteúdo, suportar serviços de streaming de áudio e vídeo, e resolver desafios na distribuição de conteúdo em dispositivos móveis. As CDNs são fundamentais para reduzir a latência, melhorar a eficiência na comunicação entre servidores e utilizadores, e aliviar o tráfego nos servidores originais, proporcionando uma experiência web mais rápida e fiável.

⇒ Evita ter que aceder ao servidor principal para obter a informação/aplicação

⇒ Evitar o congestionamento no caminho para o servidor principal

⇒ Conjunto de sites usados para melhorar o desempenho de aplicações web-based coletivamente

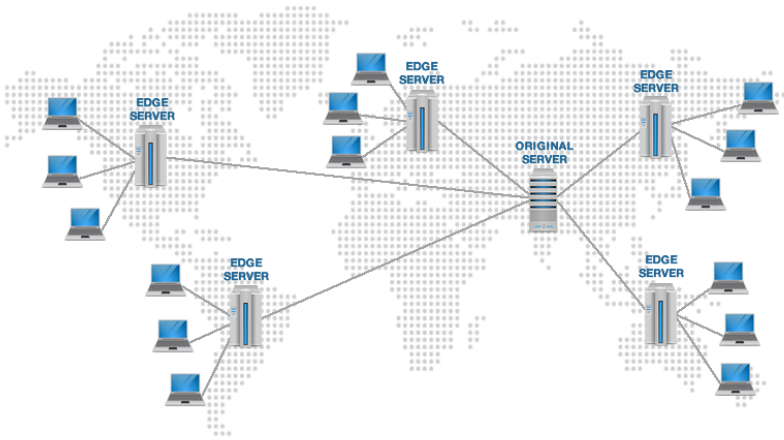


Figura 1.1: CDN [1]

1.1.1 Componentes da CDN

- **Content Delivery Infrastructure**: Entregar conteúdo aos clientes através de `surrogates`[1]

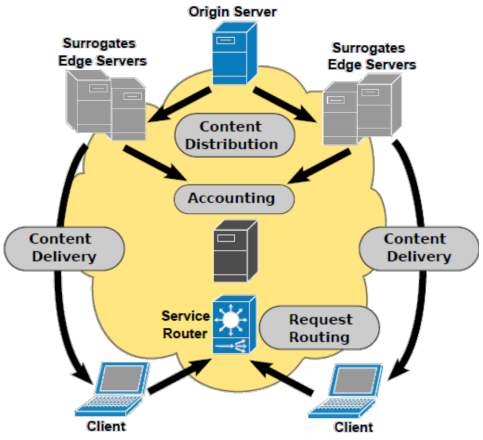


Figura 1.2: CDN Infrastructure

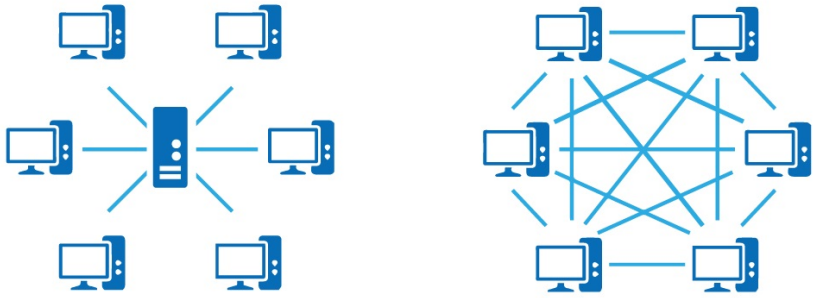
[1]Neste contexto, "surrogates"refere-se aos servidores distribuídos numa CDN que atuam como substitutos do servidor de origem

- **Request Routing Infrastructure**: Reencaminhar pedidos de conteúdo de um cliente para o `surrogate` adequado
- **Distribution Infrastructure**: Mover ou replicar conteúdo da fonte de conteúdo (servidor de origem, provedor de conteúdo) para os surrogades (acesso ao conteúdo: por exemplo, redirecionamento DNS, anycast - servidores replicados no mesmo domínio de rede)
- **Accounting Infrastructure**: Registar e reportar atividades de distribuição e entrega

1.2 Redes *peer-to-peer*

As redes *peer-to-peer* exploram a conectividade entre os participantes de uma rede, aproveitando a largura de banda cumulativa dos participantes e são comumente usadas para partilhar ficheiros de conteúdo, como áudio, vídeo e dados, incluindo dados em tempo real, como tráfego telefónico(VOIP). Numa rede puramente *peer-to-peer*, não há distinção entre clientes e servidores, com nós pares atuando simultaneamente como "clientes"e "servidores"para outros nós.

⇒ À medida que os nós chegam e a demanda no sistema aumenta, a capacidade total do sistema também aumenta.



Os desafios associados às redes *peer-to-peer* incluem a descoberta de pares e gestão de grupos, localização, busca e colocação de dados, entrega confiável e eficiente de arquivos, segurança, privacidade, anonimato e confiança.

1.2.1 Tipos de rede P2P

1.2.1.1 Pure P2P

Pure P2P refere-se a um ambiente em que todos os nós participantes são pares, sem a presença de um sistema central de controlo, coordena ou facilita

as trocas entre os peers. Neste tipo de rede, não há distinção entre clientes e servidores, pois todos os nós atuam como pares iguais que simultaneamente funcionam como ”clientes”e ”servidores”para os outros nós na rede.

1.2.2 Hybrid P2P

Hybrid P2P existem servidores que permitem que os pares interajam entre si. A quantidade de envolvimento do sistema central varia com a aplicação, e diferentes pares podem ter funções diferentes, como nós simples, routers, `rendezvous` 1.2.3.2. Esta abordagem combina a arquitetura *peer-to-peer* com o modelo client-server, proporcionando benefícios específicos para certos cenários de rede.

1.2.3 Tipos de Peers

1.2.3.1 Simple Peers

Simple Peers - são utilizadores finais individuais que fornecem serviços a partir dos seus dispositivos e consomem serviços fornecidos por outros pares na rede. Geralmente, eles estão localizados atrás de um firewall, separados da rede em geral, o que limita sua acessibilidade de rede. Devido a essa limitação, os ”Simple Peers”têm a menor responsabilidade em qualquer rede *peer-to-peer*, não sendo responsáveis por lidar com a comunicação em nome de outros pares

1.2.3.2 Rendez-vous Peers

Rendez-vous Peers - são pontos de encontro em redes *peer-to-peer*, onde os pares podem descobrir outros pares e recursos. Eles facilitam a comunicação entre os pares, armazenam informações para uso futuro e ajudam a melhorar a eficiência e responsividade da rede. Esses pontos de encontro podem estar localizados dentro ou fora da firewall, desde que possam acessar a rede de forma segura e autorizada.

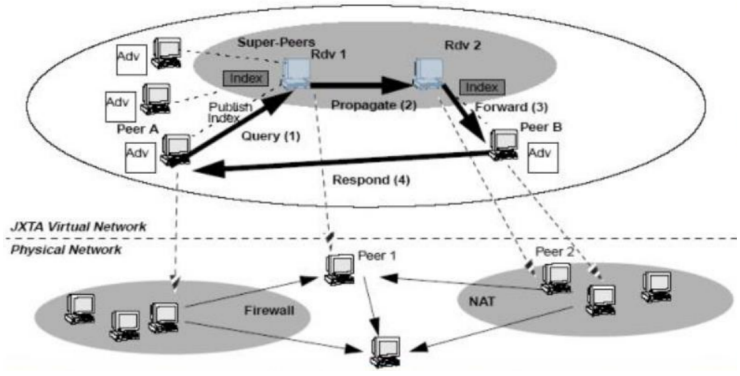


Figura 1.3: Rendez-vous Peers

1.2.3.3 Router (Relay) Peers

Router (Relay) Peers - Um router peer fornece um mecanismo para que os peers possam comunicar com outros peers separados da rede por firewall ou NAT. Pares fora da firewall podem comunicar com um peer atrás da firewall, e vice-versa. Um relay não é necessariamente um rendez-vous peer.

⇒ O relay está na transmissão de dados, enquanto o rendez-vous está sempre no caminho de descoberta (e talvez na transmissão de dados também)

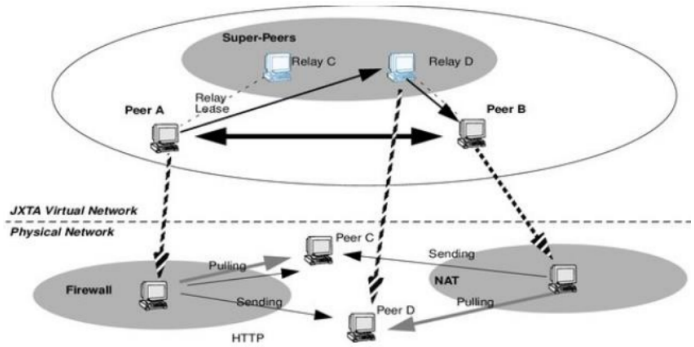


Figura 1.4: Router (Relay) Peer Network

1.2.4 Estruturada VS Não estruturada

- **P2P não Estruturadas -** são redes em que os links da overlay são estabelecidos de forma aleatória. Quando um par quer encontrar um determinado dado na rede, a consulta é disseminada pela rede para encontrar o maior número possível de pares que partilhem esses dados. Contudo, as consultas nem sempre são resolvidas, principalmente se os dados forem raros e partilhados por poucos pares. Este método de disseminação gera um elevado tráfego de sinalização na rede. Exemplos de redes P2P não estruturadas são Gnutella 1.2.5, FastTrack/KaZaa 1.2.7 e BitTorrent 1.2.8.

- **P2P Estruturadas -** existe um protocolo logicamente consistente a nível global para assegurar que qualquer nó consiga fazer routing eficiente de uma busca para um par que tenha o ficheiro desejado, mesmo que o ficheiro seja extremamente raro. O tipo mais comum de rede P2P estruturada é a DHT (1.2.9.3) e outros exemplos de redes P2P estruturadas incluem Chord, Pastry, Tapestry, CAN, Tulip, Kadmelia, BitTorrent (sem rastreador) e IPFS 1.2.9.

1.2.5 Gnutella

O protocolo Gnutella é uma rede aberta **Hibryd** 1.2.1.1 em que cada cliente também atua como servidor e é usada principalmente para partilha de arquivos, especialmente ficheiros de música, e cria uma camada de aplicação sobre a Internet, operando de forma descentralizada e mudando constantemente sua infraestrutura.

Type	**Description**	**Contained Information**
Ping	Announce availability and probe for other servents	None
Pong	Response to a ping	IP address and port# of responding servent; number and total kb of files shared
Query	Search request	Minimum network bandwidth of responding servent; search criteria
QueryHit	Returned by servents that have the requested file	IP address, port# and network bandwidth of responding servent; number of results and result set
Push	File download requests for servents behind a firewall	Servent identifier; index of requested file; IP address and port to send file to

Tabela 1.1: Gnutella: Protocol Message Types

1.2.5.1 Searching in Gnutella (structureless)

As queries são disseminadas para os vizinhos, têm um Tempo de Vida (TTL) e são encaminhadas apenas uma vez. Uma querie pode obter várias respostas indicando quais pares fornecem o ficheiro solicitado. Entre essas respostas, seleciona uma e entra em contacto diretamente com ela para descarregar o ficheiro.

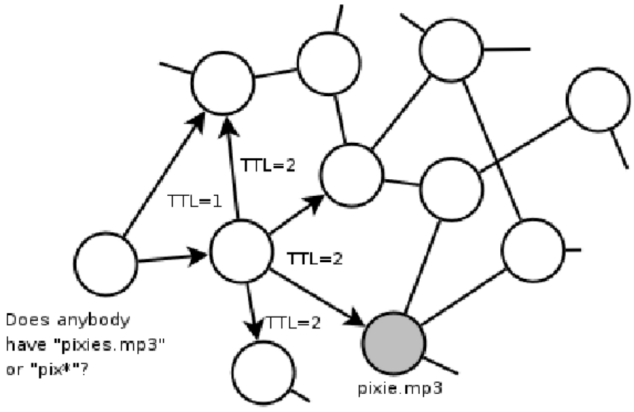


Figura 1.5: Gnutella Searching

- **Anel Expansível**
  - Iniciar a pesquisa com um TTL pequeno (por exemplo, TTL = 1).
  - Se não houver sucesso, aumentar iterativamente o TTL (por exemplo, TTL = TTL + 2).
- **K-Caminhantes Aleatórios**
  - Encaminhar a consulta apenas para um vizinho escolhido aleatoriamente, com um TTL grande.
  - Iniciar k caminhantes aleatórios.
  - O caminhante aleatório verifica periodicamente com o solicitante se deve continuar.

1.2.5.2 Hybrid Gnutella: "Ultrapeers"

- Ultrapeers podem ser instalados (KaZaA, 2001-2006) ou promovidos automaticamente (Gnutella v.2, 2003-...)
- Topologias baseada em Ultrapeers:
  - Consultas espalhadas entre ultrapeers
  - Nós folha protegidos do tráfego de consultas
  - Baseado em multiplos crawlers

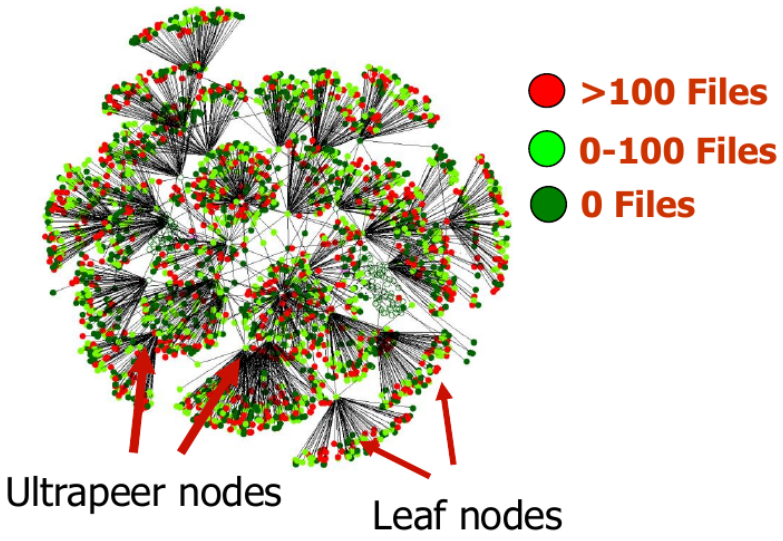


Figura 1.6: Oct 2003 Crawl of public gnutella (v.2)

1.2.6 OpenNAP/Napster

O OpenNAP é uma extensão de outros tipos de servidores que fornecem pesquisa e facilitam transferências diretas entre clientes. Opera dentro de uma arquitetura de rede **Híbrida 1.2.1.1 e Não estruturada 1.2.4** e utiliza o algoritmo Centralized Directory Model (CDM).

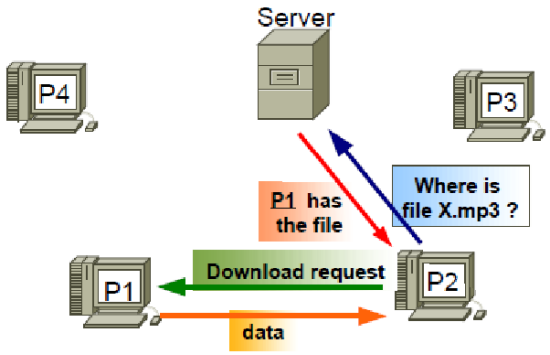


Figura 1.7: OpenNAP/Napster

1.2.7 FastTrack/KaZaA

É uma extensão do protocolo Gnutella que adiciona super-nós para melhorar a escalabilidade. Uma aplicação peer hospedada por uma máquina poderosa com uma ligação de rede rápida torna-se automaticamente um super-nó, atuando eficazmente como um servidor de indexação temporário para outros peers mais lentos. Os super-nós comunicam entre si para satisfazer pedidos de pesquisa.

A arquitetura de rede é híbrida **Híbrida 1.2.1.1 e Não estruturada 1.2.4**, com o algoritmo Flooded Requests Model (FRM).

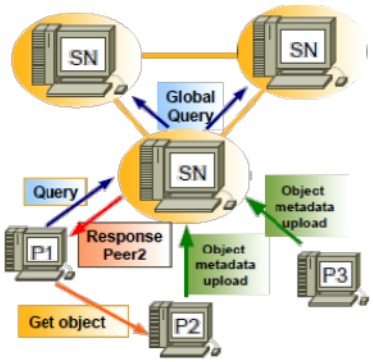


Figura 1.8: /KaZaA

1.2.8 BitTorrent

O BitTorrent é uma tecnologia de partilha de ficheiros que revolucionou a forma como partilhamos conteúdo online.

- **Tracker Central:** O BitTorrent transfere parte do trabalho de rastreamento de ficheiros para um servidor central conhecido como **tracker**. Este tracker identifica o **swarm**, que é um grupo de computadores envolvidos na transferência do mesmo arquivo, e auxilia o software do cliente na troca de partes do arquivo entre os computadores no mesmo swarm.
- **Princípio do "tit-for-tat":** Uma característica fundamental do BitTorrent é o princípio do "tit-for-tat", onde para receber arquivos, é necessário partilhá-los também. Isto resolve o problema do `leeching` [2], garantindo que todos contribuam para a partilha de ficheiros. Como resultado, é possível realizar downloads rápidos de ficheiros grandes utilizando o mínimo de largura de banda.
- **.torrent:** Os ficheiros .torrent funcionam como ponteiros, direcionando o computador para o ficheiro que se deseja fazer o download. Eles contêm informações sobre o arquivo e os trackers que ajudam na localização dos peers no swarm.
- **Download contínuo:** Mesmo após o download completo de um ficheiro, enquanto estiver a ser executado o BitTorrent, o computador pode continuar a partilhar o .torrent com outros utilizadores.

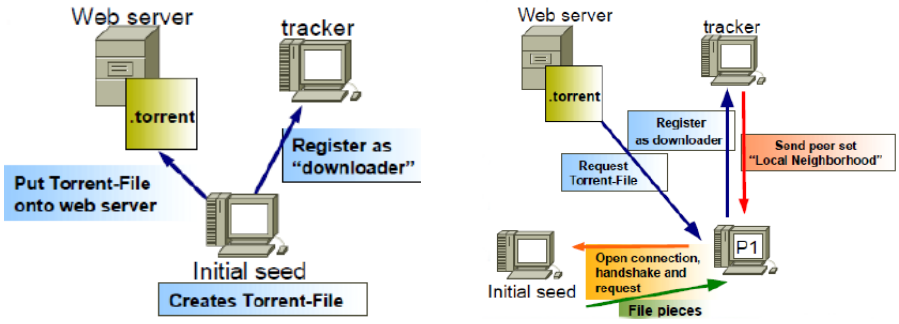


Figura 1.9: BiTorrent

[2] Acontece quando computadores apenas fazem o download de ficheiros sem dar nada em troca nem contribuindo com a rede, ou seja, sem partilhar ficheiros de volta.

1.2.9 IPFS

InterPlanetary File System (IPFS) representa uma inovação significativa, focalizada na distribuição descentralizada de dados.

- **Identificação baseada em conteúdo**: O IPFS utiliza um sistema de identificação baseado em hash seguro para garantir a integridade e a autenticidade dos conteúdos, atribuindo a cada ficheiro um hash único que permite localizar e verificar a sua integridade de forma eficiente.
- **Resolução de localizações:** As localizações dos conteúdos no IPFS são resolvidas através da Distributed Hash Tables (DHT) 1.2.9.3, um sistema descentralizado que mapeia chaves de conteúdo para os nós que o possuem.
- **Troca de blocos**: Para a distribuição de arquivos, o IPFS utiliza o protocolo *peer-to-peer* Bittorrent.

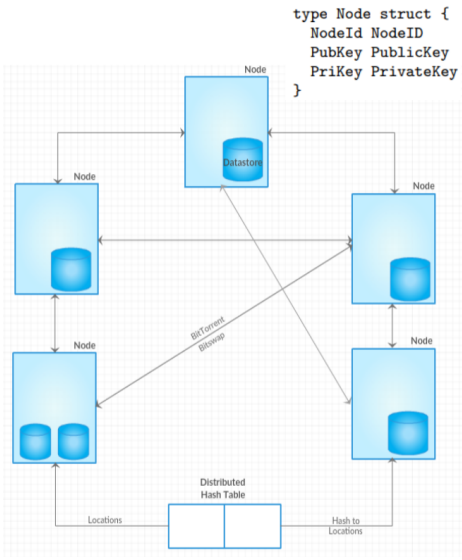


Figura 1.10: IPFS

- **Protocolo Bitswap**: Além do Bittorrent, o IPFS incentiva a troca de blocos através do protocolo Bitswap 1.2.9.2, que otimiza a transferência de dados entre os nós, priorizando a obtenção de blocos que o nó ainda não possui.
- **CID:** No IPFS, cada ficheiro ou diretório tem um **CID**(Content Identifier)(Um hash SHA256 único usado para identificá-lo).
- **Organização baseada em versões**: Uma característica fundamental do IPFS é sua organização baseada em versões do `Merkle DAG`, similar ao sistema de controlo da versão Git.

- **Segurança**: Para garantir a segurança, o IPFS utiliza servidores de auto-certificação para os nós de armazenamento. Isto ajuda a proteger os dados contra alterações maliciosas e garante a autenticidade dos nós na rede.
- **Robustez e eficiência**: Todas essas características combinadas tornam o IPFS uma solução robusta e segura para o armazenamento e distribuição descentralizada de dados. Promove a resiliência e eficiência na troca de informações na rede, sem depender de servidores centralizados.
- **Split Factor:** É uma técnica adaptativa para controlar a eficiência da rede IPFS, garantindo que os blocos sejam distribuídos de forma eficiente e minimizando duplicatas desnecessárias.

1.2.9.1 Passo a passo:

- Os ficheiros estão em armazenamento distribuído
- A DHT utiliza o hash do ficheiro como chave para devolver a localização do mesmo.
- Uma vez determinada a localização, a transferência ocorre de forma *peer-to-peer* como uma transferência descentralizada.

1.2.9.2 Bitswap

O protocolo Bitswap é utilizado para incentivar nós pares que mantêm blocos de dados. Esses nós possuem listas de desejos (want_list) e listas de posses (have_list). Qualquer desequilíbrio é registado na forma de créditos e dívidas de Bitswap.

Além disso, o protocolo Bitswap possui disposições para lidar com exceções, como nós que não contribuem, nós que não desejam nada e nós que não possuem nada.

1. **Open**: os pares enviam `ledgers` até chegarem a um acordo.
2. **Envio**: os pares trocam listas de desejos (*want_lists*) e blocos.
3. **Fechar**: os pares desativam uma conexão.
4. **Ignorado**: (especial) um par é ignorado (por um determinado tempo limite) se a estratégia de um nó evitar o envio.

```
// Protocol interface:
interface Peer {
    open (nodeid: NodeId, ledger : Ledger);
    send_want_list (want_list :WantList);
    send_block (block :Block) -> (complete : Bool);
    close (final : Bool);
}
```

1.2.9.3 DHT

1.2.9.3.1 Searching in DHTs

A pesquisa em DHTs estruturadas requerem que seja conhecido o nome exato do ficheiro que se procura. Os nomes dos ficheiros (chaves) estão mapeados para identificadores de nós. Se houver uma alteração no nome do ficheiro, a pesquisa terá de ser feita em nós diferentes. Não é possível fazer correspondência com `wildcard matching` [3]: não é possível procurar por um ficheiro que comece com "pix*".

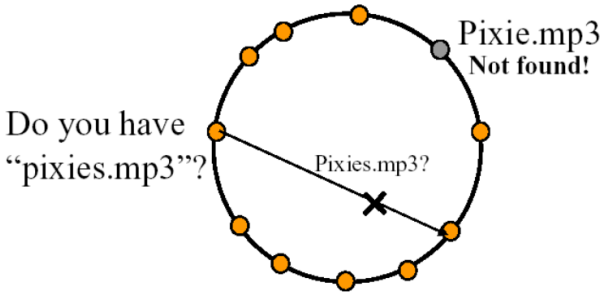


Figura 1.11: DHT searching

1.2.9.3.2 CHORD

Circular Hashing of Resource Distribution (CHORD) é um algoritmo de Searching em DHTs permite que os nós numa rede *peer-to-peer* localizem e acessem aos recursos de forma eficiente, aproveitando uma estrutura de anel e funções de hash para organizar e distribuir esses recursos na rede.

1. Cada ficheiro recebe um identificador, que é convertido num código através de um processo chamado "`hashing`". Esse código é então usado como chave para esse recurso específico.

2. Quando um nó precisa de um ficheiro ou dado, ele também aplica a função de hash ao nome desse recurso, utilizando SHA-1, e envia um pedido usando essa chave gerada.

3. Todos os nós na rede também utilizam a mesma função de hash para converter os seus endereços IP em códigos. Conceptualmente, estes nós organizam-se num anel, com os endereços IP hashados ordenados de forma crescente ao longo deste anel.

[3] Caracteres wildcard são caracteres tipo o asterisco (*) ou o ponto de interrogação (?) que são comumente usados para representar um ou mais caracteres desconhecidos numa busca.

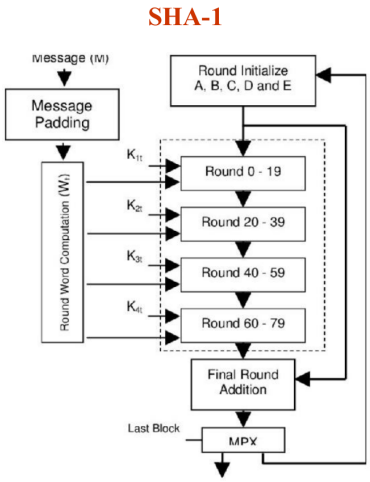


Figura 1.12: CHORD: DHT Algorithm

1.2.9.3.3 Search Information

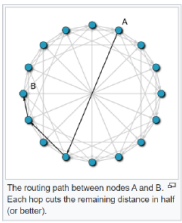
Quando um nó deseja obter um conteúdo:

a) Ele converte o identificador do dado num código hash e envia um pedido para o sucessor desse código.

b) O sucessor responde com o endereço IP do nó que contém o dado real.

c) Para solicitar informações do sucessor, quando não conhece o seu IP, mas apenas a chave:

1. Cada nó mantém uma tabela de referência chamada de `finger table` 1.2.9.3.4.
2. Essa tabela contém uma lista de chaves e os IPs dos sucessores correspondentes.
3. Cada nó possui o IP de uma sequência exponencial de nós que o seguem, ou seja, a entrada i da `finger table` do nó k contém o IP do nó $k + 2^i$.



1.2.9.3.4 Finger Table

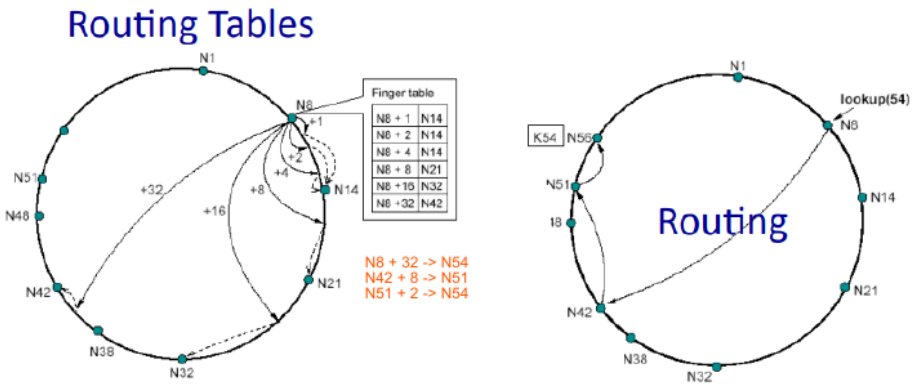
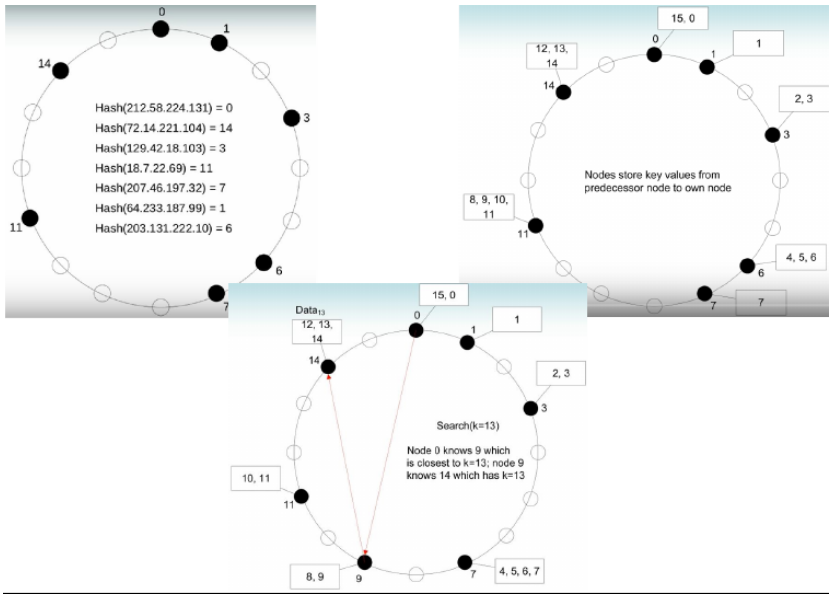


Figura 1.13: Finger Table

Examplo:



1.2.9.3.5 Flooding vs. DHTs

Flooding	DHTs
Pode falhar em encontrar alguns ficheiros.	Pode não ser eficiente em seleções complexas como crac. wildcard.
Usa muita largura de banda.	Mais eficiente em encontrar coisas.
Utiliza lógica arbitrária de um único local.	Pode realizar junções equi, seleções, agregações, etc.

Tabela 1.2: Comparação entre Flooding e DHTs

Solução?? 🤔🤔🤔

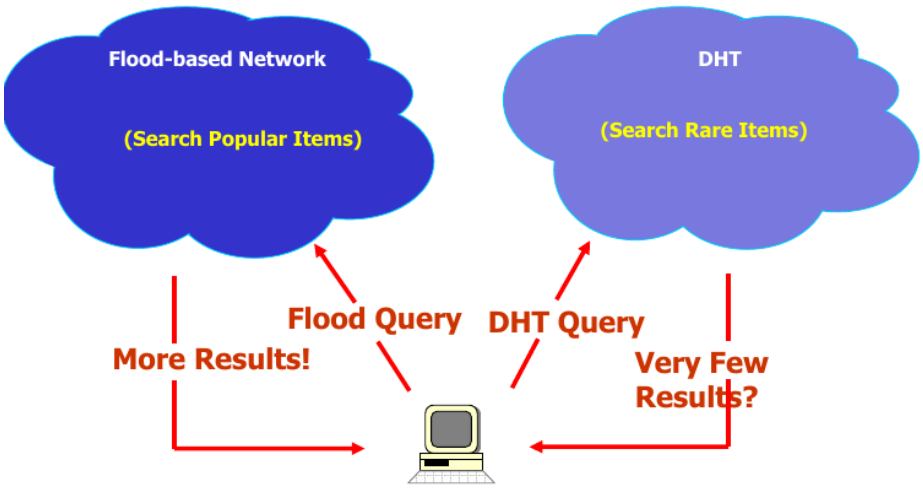


Figura 1.14: Hybrid = "Best of both worlds"

1.2.10 Security

- **Poisoning Attacks💀:** Por exemplo, fornecer ficheiros cujos conteúdos são diferentes da descrição.
- **Polluting Attacks:** Por exemplo, inserir ”pedaços”ou pacotes ”maus”😠 num ficheiro válido na rede.
- **Freeloaders:** Utilizadores ou software que utilizam a rede sem contribuir com recursos para a mesma.
- **Inserção de Vírus nos Dados Transportados:** Por exemplo, ficheiros descarregados ou transportados podem estar infetados com vírus ou outro malware.
- **Malware no Software da Rede Peer-to-Peer em Si:** Por exemplo, software distribuído pode conter spyware.
- **Denial of Service Attacks:** Ataques que podem fazer a rede funcionar muito lentamente ou parar completamente.
- **Filtering:** Operadores de rede podem tentar evitar que os dados das redes *peer-to-peer* sejam transportados.
- **Identity Attacks:** Por exemplo, rastrear os utilizadores da rede e assediá-los ou atacá-los legalmente.
- **Spamming:** Por exemplo, enviar informações não solicitadas pela rede, não necessariamente como um ataque de negação de serviço.

Solução?? 🤔🤯

- **A maioria dos ataques pode ser derrotada ou controlada por meio do design cuidadoso da rede *peer-to-peer* e pelo uso de criptografia.** No entanto, quase qualquer rede falhará quando a maioria dos pares estiver a tentar danificá-la.
- **Anonimato:** Alguns protocolos *peer-to-peer* (como o Freenet) tentam ocultar a identidade dos utilizadores da rede, passando todo o tráfego por nós intermediários.
- **Criptografia:** Algumas redes *peer-to-peer* criptografam os fluxos de tráfego entre pares. Isso dificulta um ISP detetar se a tecnologia *peer-to-peer* está a ser utilizada (pois alguns limitam artificialmente a largura de banda), esconde o conteúdo do ficheiro de espiões, impede esforços de aplicação da lei ou censura de certos tipos de material, autentica utilizadores e previne ataques ”man in the middle”em protocolos, e ajuda a manter o anonimato.

Bibliografia

[1] aaplha net solutions, *CDN – Content Delivery Network*, https://www.aalphanetsolution.com/blog/what-is-cdn/.