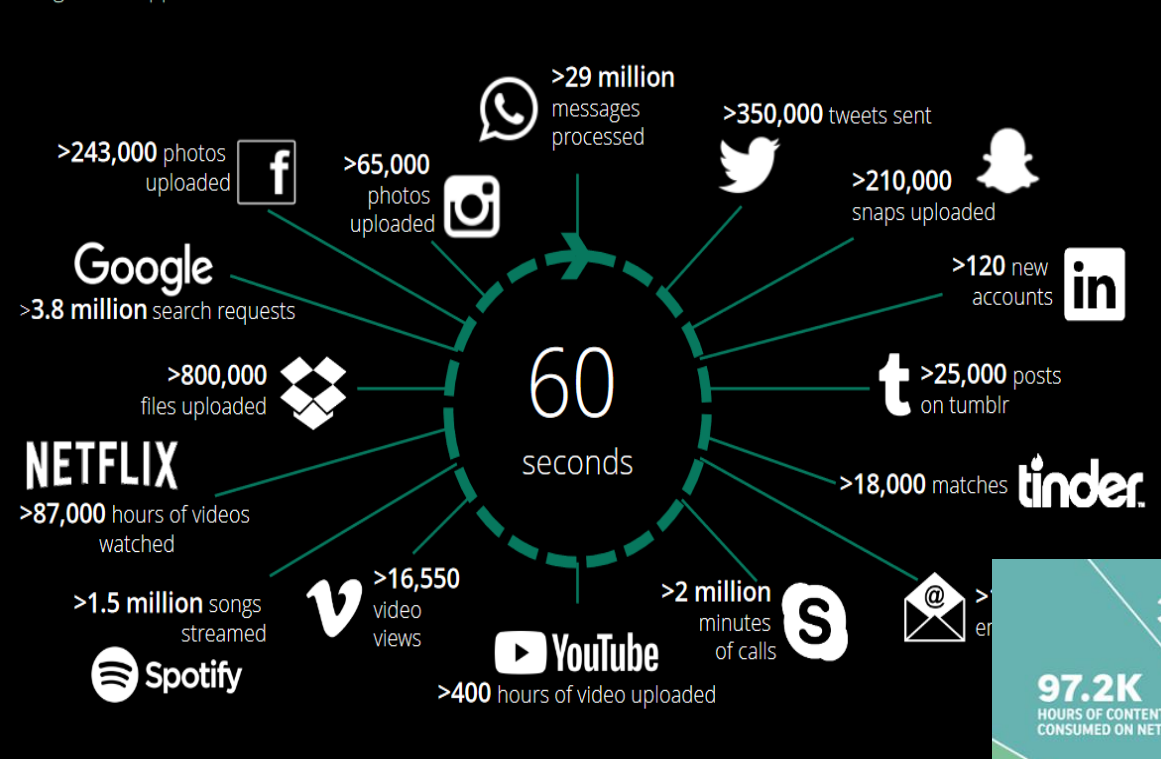


Sistemas ponto a ponto e Redes

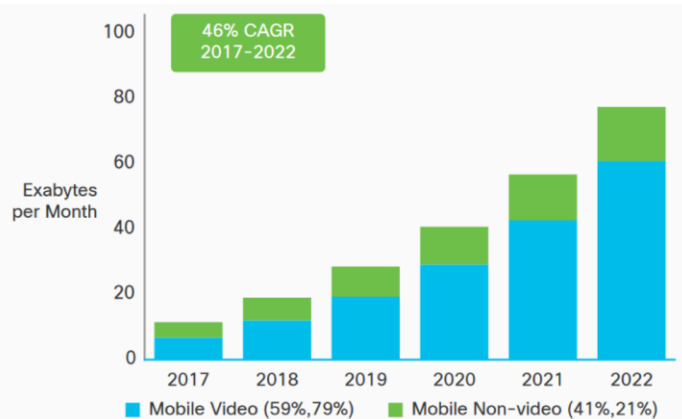
**Mestrado em Engenharia de
Computadores e Telemática
2023/2024**



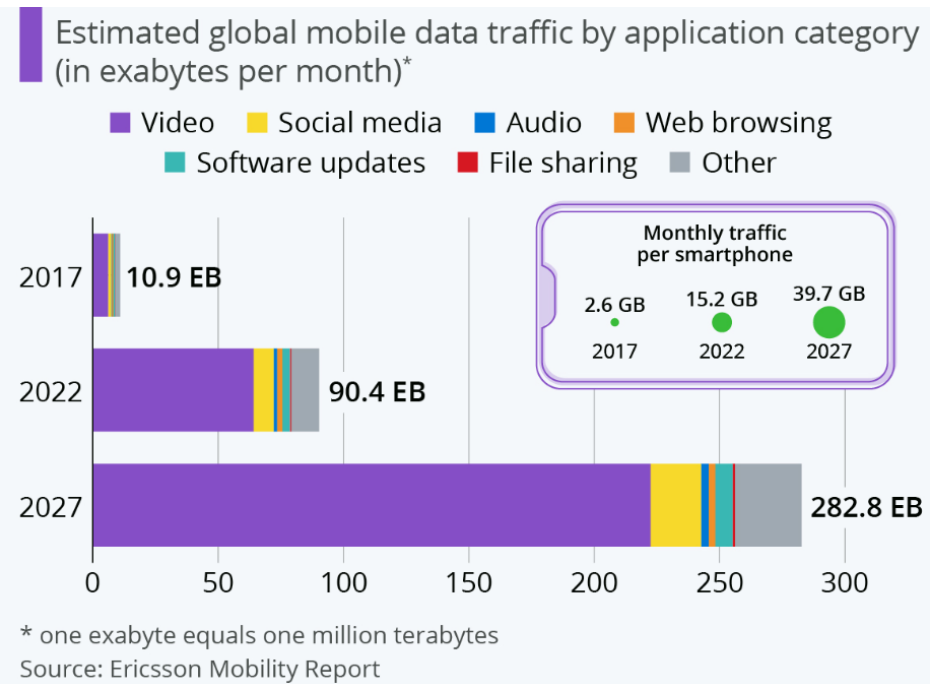
Motivação

- Redes baseadas em IP
- Aplicativos baseados na Web tornaram-se a norma para redes internas corporativas e muitas interações entre empresas
- Grande aceitação e crescimento explosivo
 - Sérios problemas de desempenho
 - Experiência do usuário degradada
 - Para um grande conjunto de aplicações, incluindo acesso a VÍDEO
- Melhorar o desempenho de aplicativos em rede
 - Use muitos sites em diferentes pontos da rede
 - Servidores autônomos
 - Roteadores

Vídeo móvel



Note: Figures in parentheses refer to 2017 and 2022 traffic share.
Source: Cisco VNI Mobile, 2019



Redes de distribuição de conteúdo

- O cliente tenta acessar o site do servidor principal de um aplicativo
- É redirecionado para um dos outros sites
- Cada site armazena informações em cache
 - Evite ir ao servidor principal para obter a informação/aplicação
- Acesse um local próximo
 - Evite congestionamento no caminho para o servidor principal
- Conjunto de sites usados para melhorar o desempenho de aplicativos baseados na Web coletivamente
 - Rede de distribuição de conteúdo

O que é um CDN?

- Rede de distribuição de conteúdo

- Às vezes também chamada de Rede de Distribuição de Conteúdo
- Pelo menos metade dos bits do mundo são entregues por uma CDN

- Provavelmente mais próximo de 80/90%

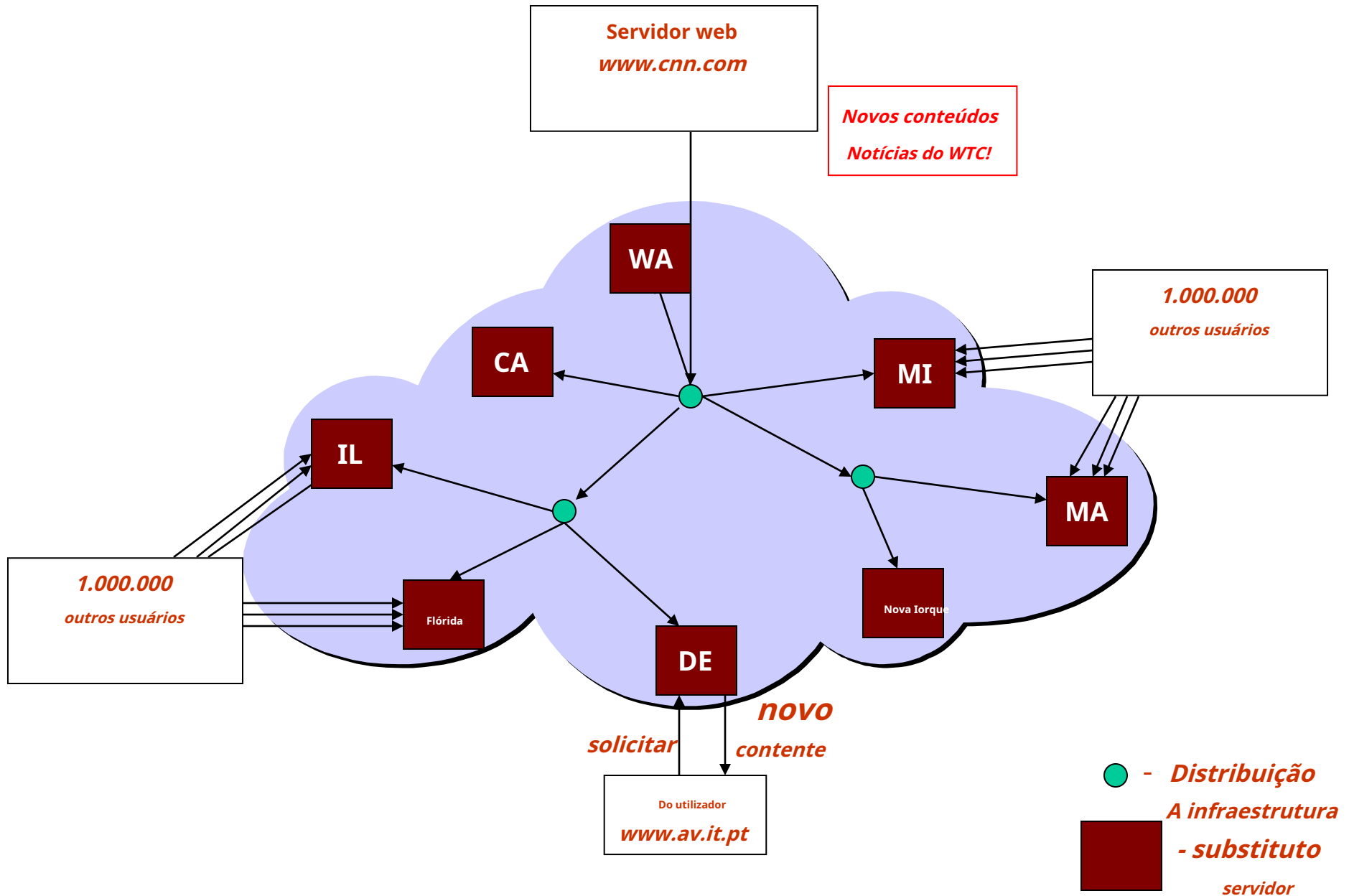
- Metas Primárias

- Crie réplicas de conteúdo em toda a Internet
- Garantir que as réplicas estejam sempre disponíveis
- Direcionar clientes para réplicas que darão bom desempenho

Principais componentes de um CDN

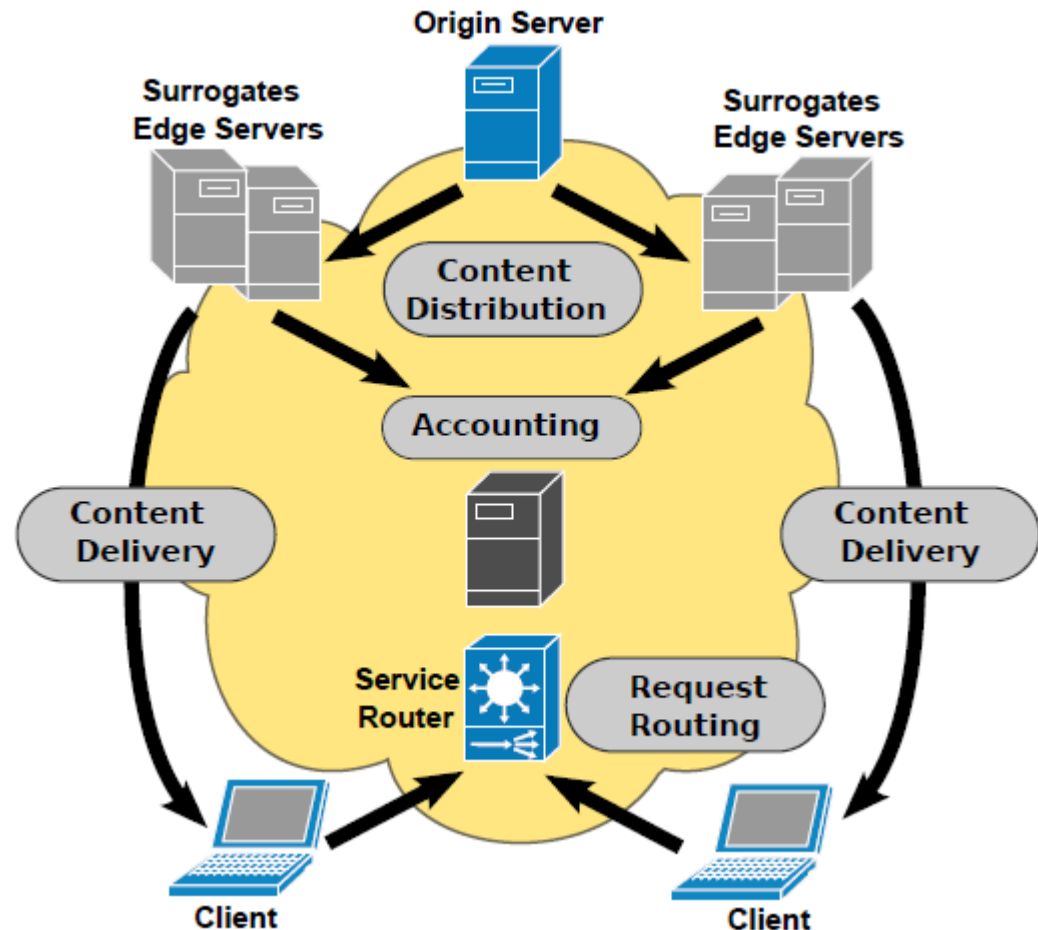
- Servidores distribuídos
 - Geralmente localizado dentro de outros ISPs
- Rede de alta velocidade conectando-os
- Clientes
 - Pode estar localizado em qualquer lugar do mundo
 - Eles querem desempenho rápido na Web
- Vinculação de clientes e servidores distribuídos
 - Algo que liga os clientes a servidores de réplica “próximos”

Arquitetura CDN



Componentes CDN

- *Infraestrutura de entrega de conteúdo:* Entrega de conteúdo aos clientes a partir de substitutos
- *Solicitar infraestrutura de roteamento:* Orientar ou direcionar a solicitação de conteúdo de um cliente para um substituto/substituto adequado
- *Infraestrutura de Distribuição:*
Mover ou replicar conteúdo da fonte de conteúdo (servidor de origem, provedor de conteúdo) para substitutos (acesso ao conteúdo: por exemplo, redirecionamento de DNS, anycast – servidores de réplica no mesmo domínio de rede)
- *Infraestrutura Contábil:*
Registro e relatórios de atividades de distribuição e entrega



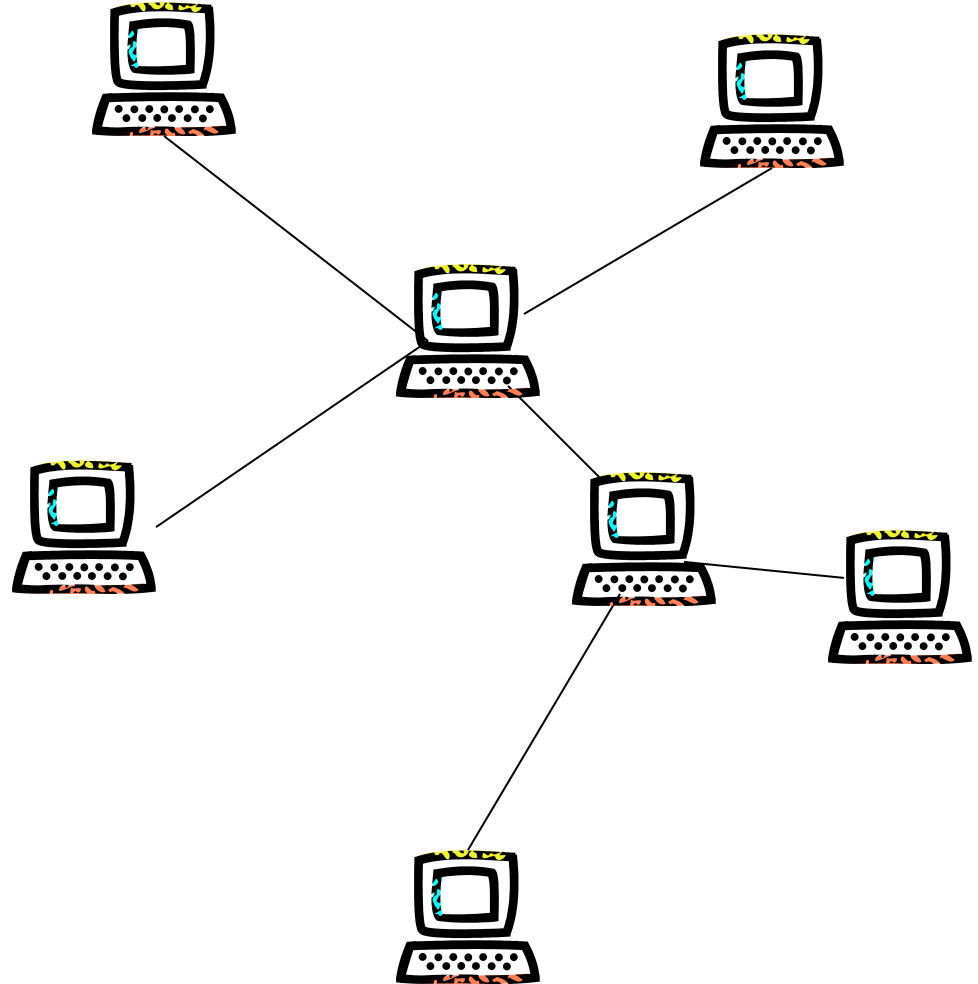
Redes ponto a ponto

Redes ponto a ponto

- Explora conectividade diversificada entre participantes de uma rede
- Explora a largura de banda cumulativa dos participantes da rede
- Normalmente usado para conectar nós por meio de conexões grandes
 - Compartilhamento de arquivos de conteúdo contendo áudio, vídeo, dados
 - Mesmo dados em tempo real, como tráfego de telefonia, também são transmitidos usando tecnologia P2P (Skype)
- Rede ponto a ponto pura
 - Não há noção de clientes ou servidores
 - Nós pares iguais que funcionam simultaneamente como "clientes" e "servidores" para os outros nós da rede

O modelo P2P

- Os recursos de um par são semelhantes aos recursos dos outros participantes
- P2P – pares comunicando diretamente com outros pares e compartilhando recursos
- Serviços P2P
 - Computação distribuída
 - Compartilhamento de arquivos
 - Colaboração



Vantagens

- Os clientes fornecem recursos, incluindo largura de banda, espaço de armazenamento e poder de computação
- À medida que os nós chegam e a demanda do sistema aumenta, a capacidade total do sistema também aumenta
- A natureza distribuída também aumenta a robustez em caso de falhas, replicando dados em vários pares
 - Permita que os pares encontrem os dados sem depender de um servidor de indexação centralizado

Aplicativos P2P

- Compartilhamento de arquivos
 - Usando protocolos da camada de aplicação
 - DirectConnect (centralizado), Gnutella (inundação), BitTorrent (híbrido), IPFS
- VoIP
 - Usando protocolos da camada de aplicação
 - TRAGO
- Streaming de mídia
- Mensagem instantânea
- Publicação e distribuição de software
- Publicação e distribuição de mídia
 - rádio, vídeo

Desafios

- Descoberta de pares e gerenciamento de grupo
- Dados localização, pesquisa e posicionamento
 - Pesquisa e roteamento
- Entrega de arquivos confiável e eficiente
- Segurança/privacidade/anonimato/confiança

Tipos P2P

- **P2P puro** refere-se a um ambiente onde todos os nós participantes são pares
 - Nenhum sistema central controla, coordena ou facilita as trocas entre pares
- **P2P híbrido** refere-se a um ambiente onde existem servidores que permitem que os pares interajam entre si
 - O grau de envolvimento do sistema central varia de acordo com a aplicação
 - Diferentes peers podem ter funções diferentes (nós simples, roteadores, rendezvous)

Usado dependendo da aplicação

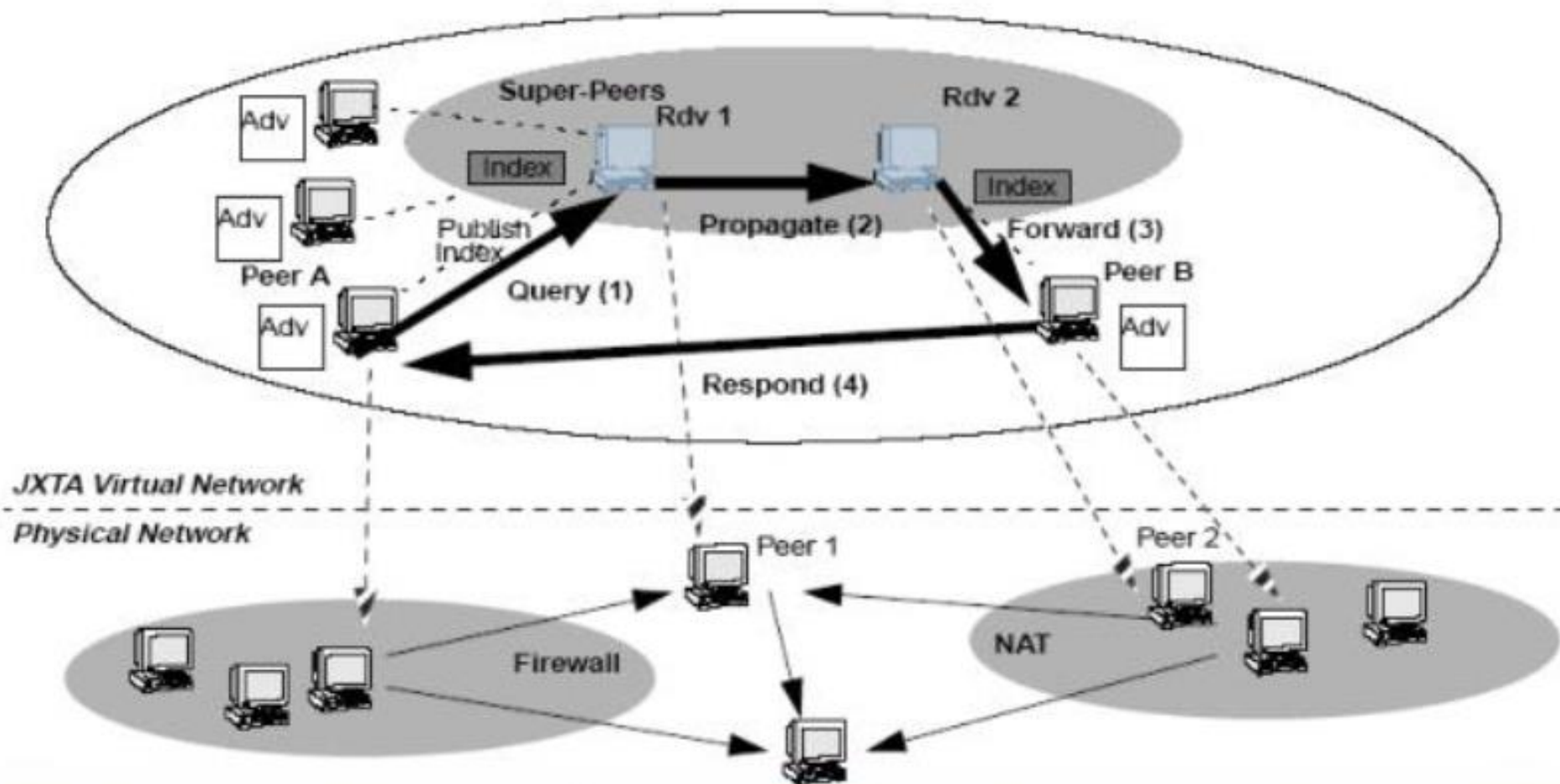
Pares simples

- Usuário final único, permitindo que esse usuário forneça serviços a partir de seu dispositivo e consuma serviços fornecidos por outros pares na rede
 - Geralmente estará localizado atrás de um firewall, separado da rede como um todo; peers fora do firewall provavelmente não serão capazes de se comunicar diretamente com o peer simples localizado dentro do firewall.
 - Devido à sua acessibilidade limitada à rede, os pares simples têm a menor responsabilidade em qualquer rede P2P.
- Eles não são responsáveis por lidar com a comunicação em nome de outros pares.

Encontro de pares

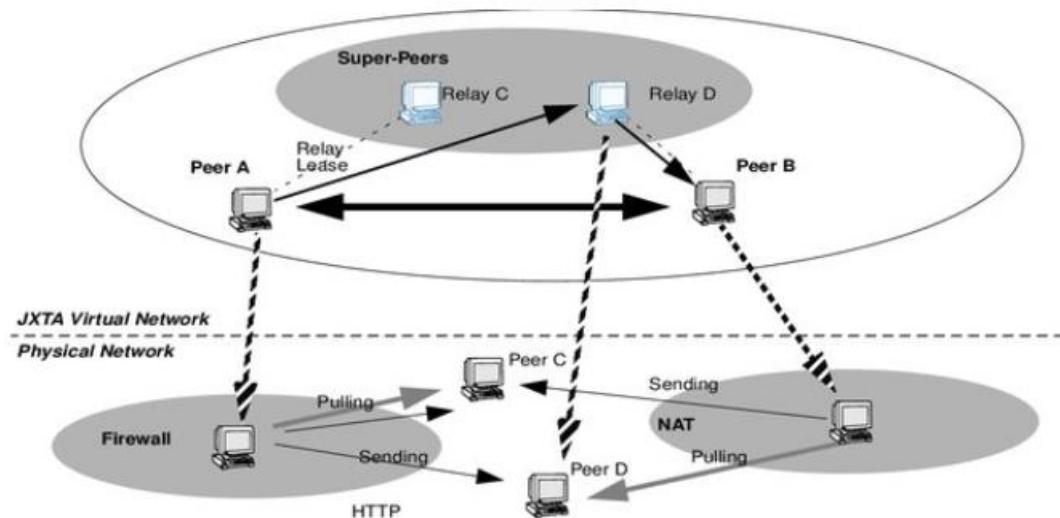
- Local de reunião ou encontro
 - Fornece aos peers um local de rede para usar para descobrir outros peers e recursos de peer.
- Os peers emitem consultas de descoberta para um peer de encontro, e o encontro fornece informações sobre os peers que ele conhece na rede.
- Pode armazenar informações em cache sobre pares para uso futuro ou encaminhando solicitações de descoberta para outros pares de encontro
 - Melhore a capacidade de resposta, reduza o tráfego de rede e forneça um melhor serviço a pares simples.
- Geralmente fora do firewall de uma rede interna privada. Um ponto de encontro poderia existir atrás do firewall, mas precisaria ser capaz de atravessá-lo usando um protocolo autorizado pelo firewall ou um ponto de roteador fora do firewall.

Encontro de pares



Pares de roteador (retransmissão)

- Um peer roteador fornece um mecanismo para que os peers se comuniquem com outros peers separados da rede por firewall ou equipamento de Network Address Translation (NAT).
- Pontos fora do firewall para se comunicar com um ponto atrás do firewall e vice-versa.
- Um **relé** não é necessariamente um ponto de encontro
 - O relé está no fluxo de dados
 - O Rendez-vous está sempre no caminho da descoberta (e talvez no fluxo de dados).



Estruturado vs Não Estruturado

- Redes P2P não estruturadas

- Formado quando os links de sobreposição são estabelecidos arbitrariamente.
- Se um par quiser encontrar um dado desejado na rede, a consulta será inundada pela rede para encontrar o maior número possível de pares que compartilhem os dados

- As **dúvidas** nem sempre podem ser resolvidas

- Se um peer estiver procurando por dados raros compartilhados apenas por alguns outros peers, então é altamente improvável que a busca seja bem-sucedida.

- Inundações causam uma grande quantidade de tráfego de sinalização na rede

- Gnutella e FastTrack/KaZaa, BitTorrent

- Redes P2P estruturadas

- Protocolo (lógica) globalmente consistente para garantir que qualquer nó possa rotear eficientemente uma pesquisa para algum ponto que tenha o arquivo desejado, mesmo que o arquivo seja extremamente raro

- O tipo mais comum de rede P2P estruturada é a Distributed Hash Table (DHT)

- Uma variante de hashing consistente é usada para atribuir a propriedade de cada arquivo a um peer específico
- Chord, Pastry, Tapestry, CAN, Tulip, Kadmelia, BitTorrent (trackerless), IPFS

Informações totalmente descentralizadas

Sistemas

- Compartilhamento de arquivos P2P

- Aplicação em escala global

- Exemplo: Gnutella

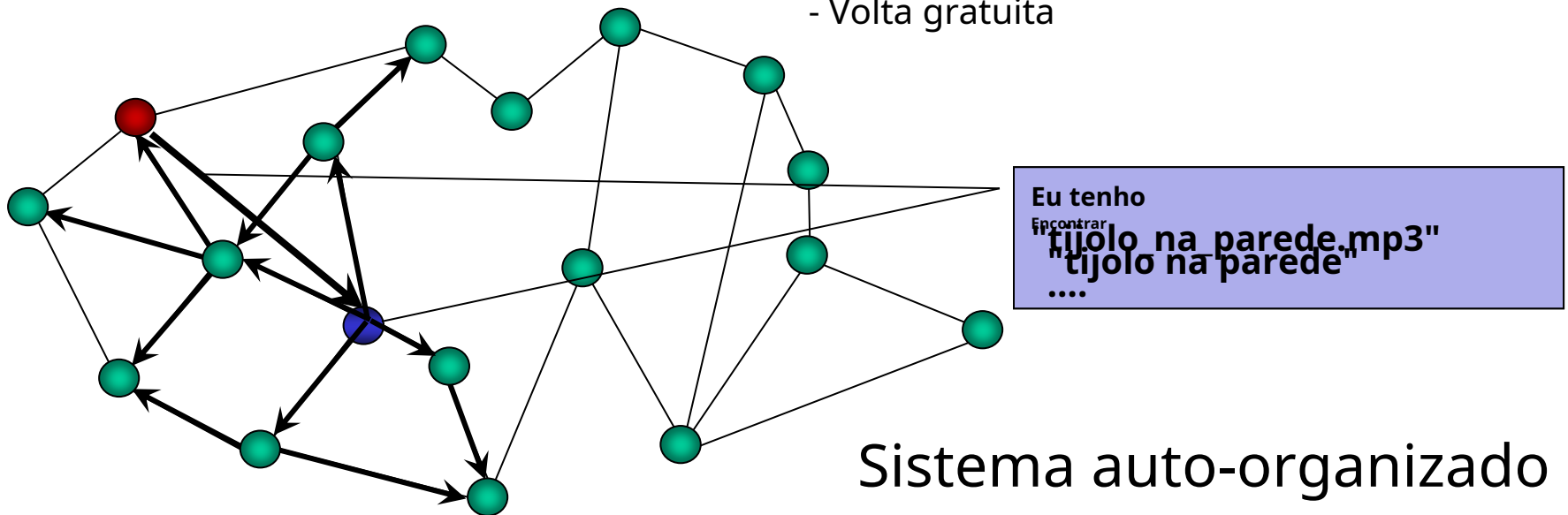
- 40.000 nós, 3 milhões de arquivos (agosto de 2000)
 - Nós 3M (janeiro de 2006)

- Forças

- Bom tempo de resposta, escalável
 - Sem infraestrutura, sem administração
 - Nenhum ponto único de falha

- Fraquezas

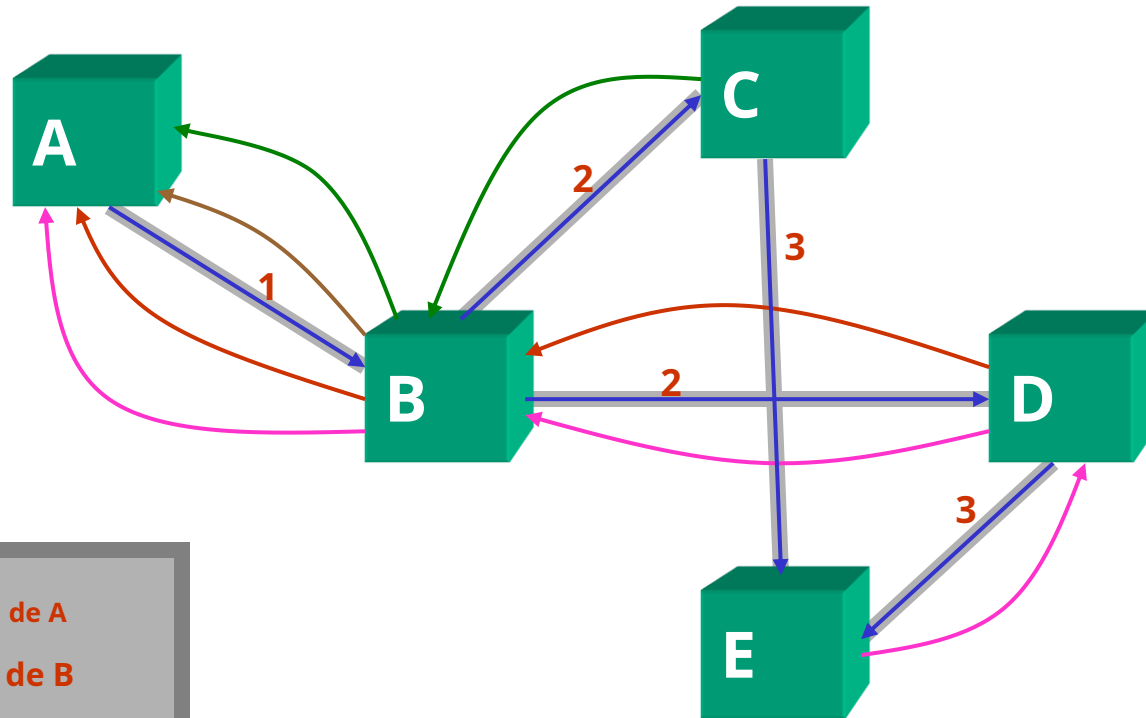
- Alto tráfego de rede
 - Sem pesquisa estruturada
 - Volta gratuita



Gnutella: sem servidores

Gnutella: Conhecendo Pares

(Pingue-pongue)



- O ping de A
- Pong de B
- Pong de C
- Pong do D
- E-pong

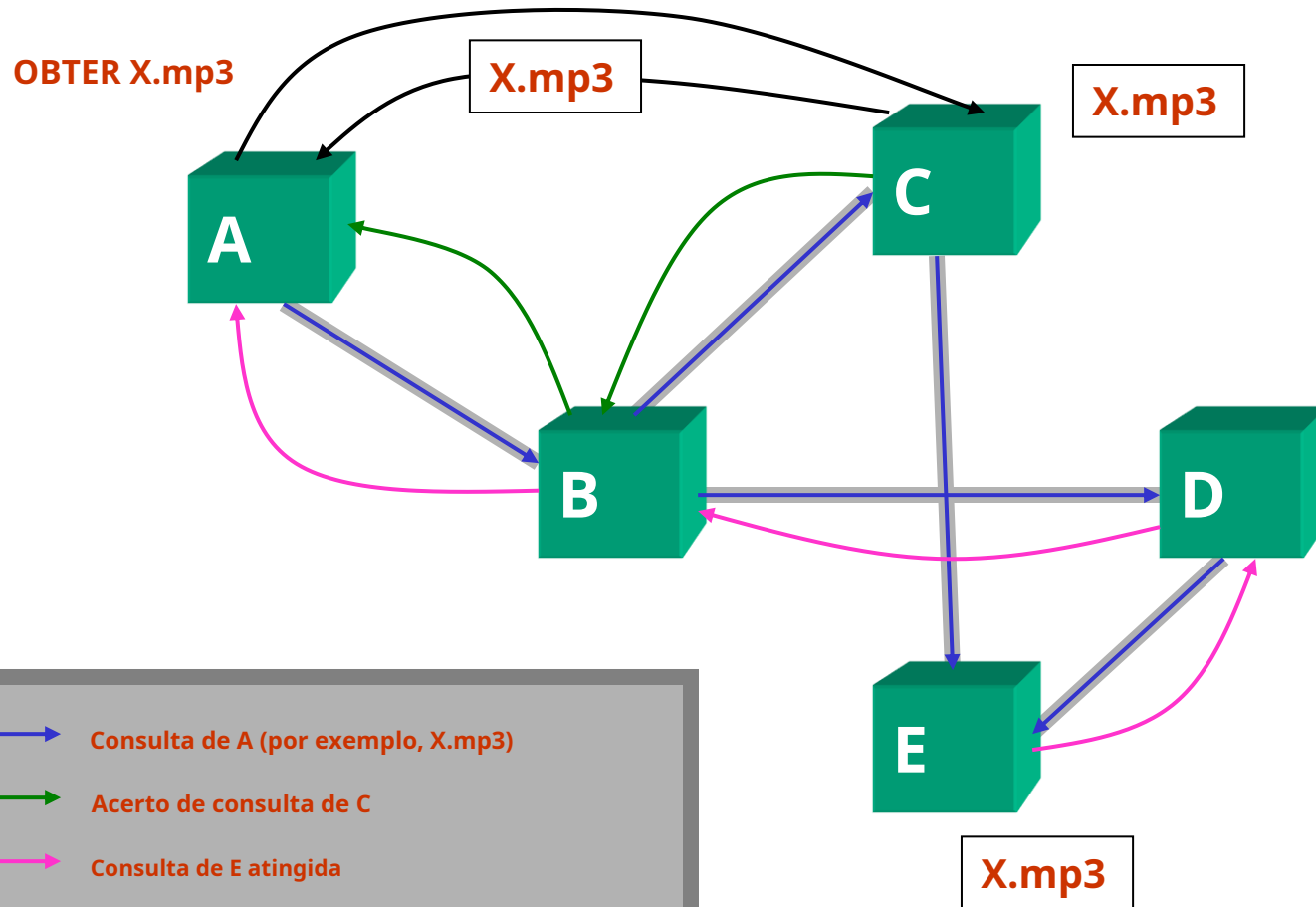
Gnutella: Mensagem de Protocolo

Tipos

Tipo	Descrição	Informações contidas
Pingar	Anuncie a disponibilidade e investigue outros servidores	Nenhum
Pong	Resposta a um ping	Endereço IP e número da porta do serviço de resposta; número e total de KB de arquivos compartilhados
Consulta	Solicitação de pesquisa	Largura de banda mínima da rede do serviço de resposta; critérios de pesquisa
ConsultaHit	Retornado por servidores que possuem o arquivo solicitado	Endereço IP, número da porta e largura de banda da rede do serviço de resposta; número de resultados e conjunto de resultados
Empurrar	Solicitações de download de arquivos para servidores atrás de um firewall	Identificador de serviço; índice do arquivo solicitado; Endereço IP e porta para enviar o arquivo

Gnutella: Procurando

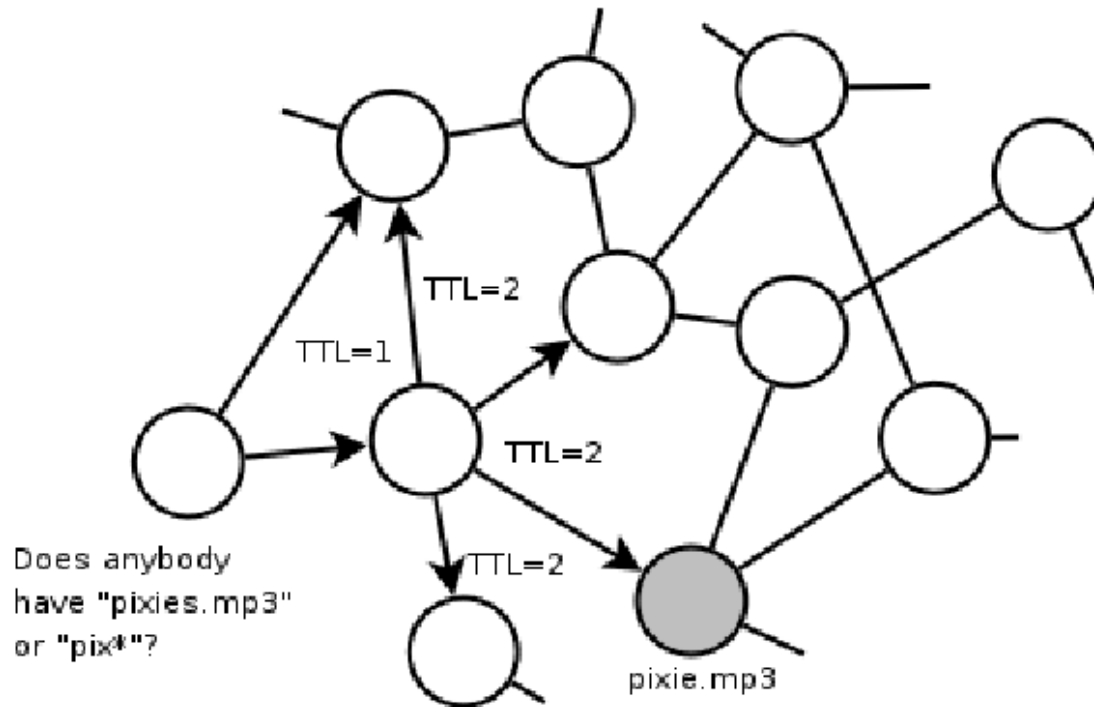
(Consulta/QueryHit/GET)



Procurando em Gnutella (sem estrutura)

- As consultas são enviadas aos vizinhos, têm um TTL e são encaminhadas apenas uma vez
- A consulta pode obter diversas respostas indicando quais peers fornecem o arquivo solicitado. Entre eles seleciona um e o contata diretamente para baixar o arquivo.

– Podemos pesquisar usando menos pacotes?



Melhorias de mensagem

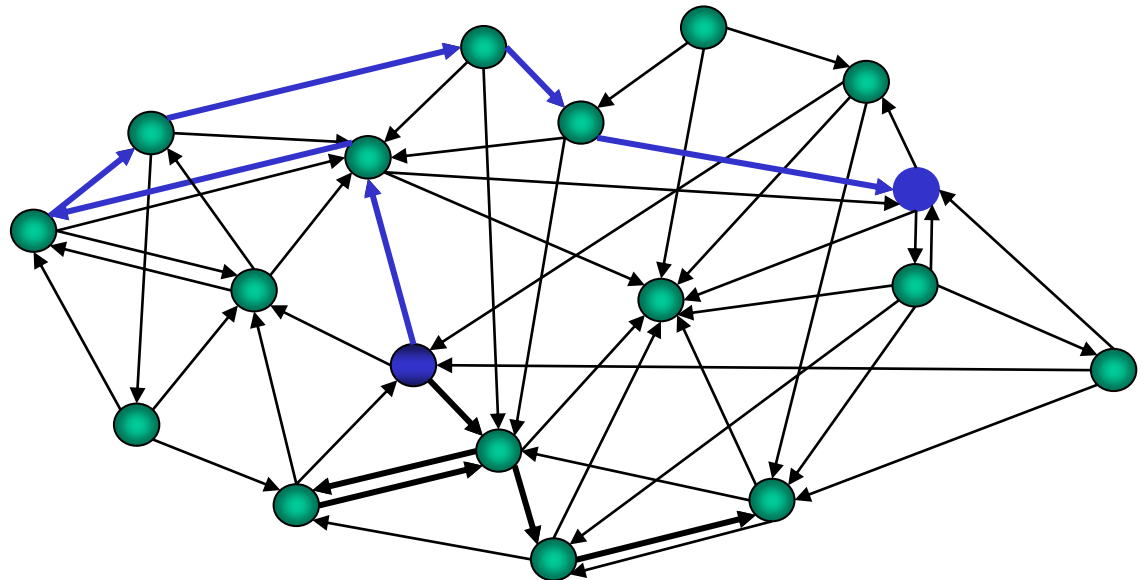
Inundações

- Anel Expansível

- iniciar a pesquisa com TTL pequeno (por exemplo, $TTL = 1$)
- se não houver sucesso, aumente iterativamente o TTL (por exemplo, $TTL = TTL + 2$)

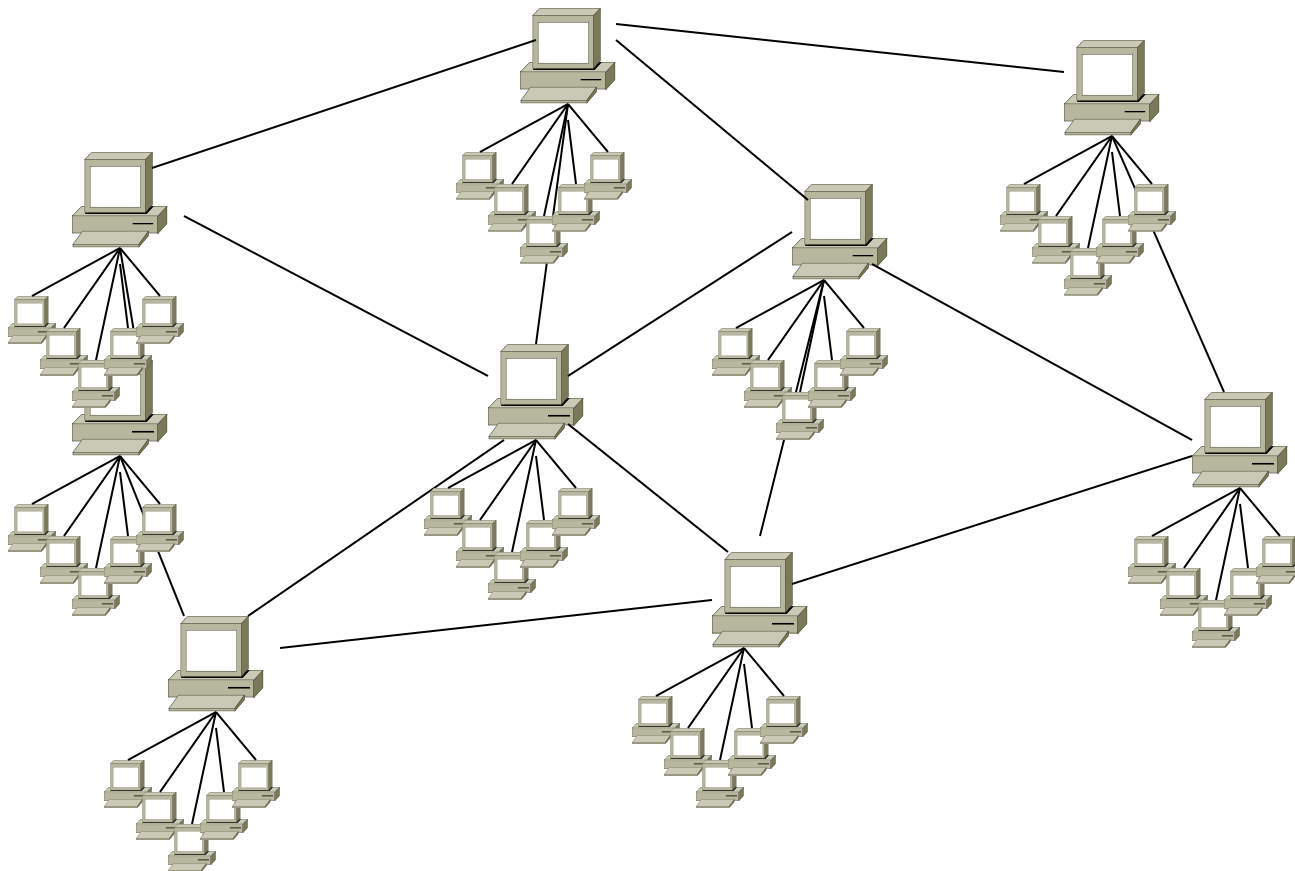
- k-caminhantes aleatórios

- encaminha a consulta apenas para um vizinho escolhido aleatoriamente, com TTL grande
- iniciar k caminhantes aleatórios
- o caminhante aleatório verifica periodicamente com o solicitante se deve continuar



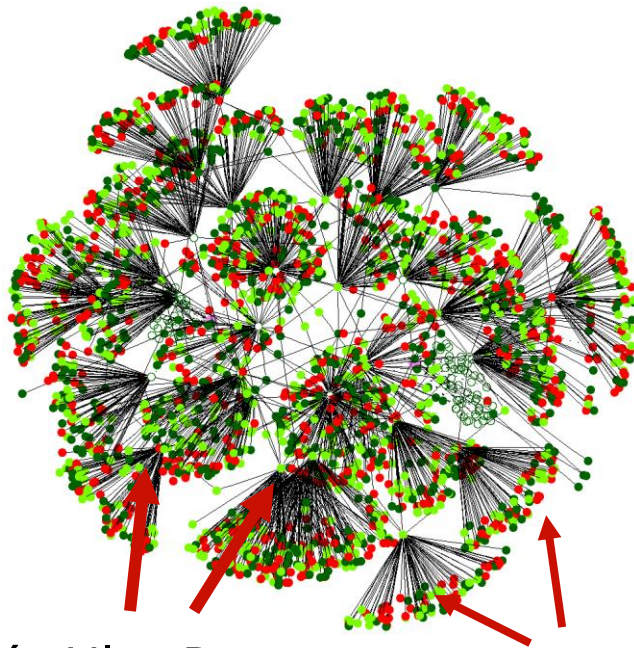
Gnutella Híbrida: “Ultrapeers”

- Ultrapeers podem ser instalados (KaZaA, 2001-2006) ou autopromovidos (Gnutella v.2, 2003-...)



Rede Real Gnutella

Outubro de 2003 Rastreamento de gnutella pública (v.2)



Nós UltraPeer

Nós folha

● > 100 arquivos

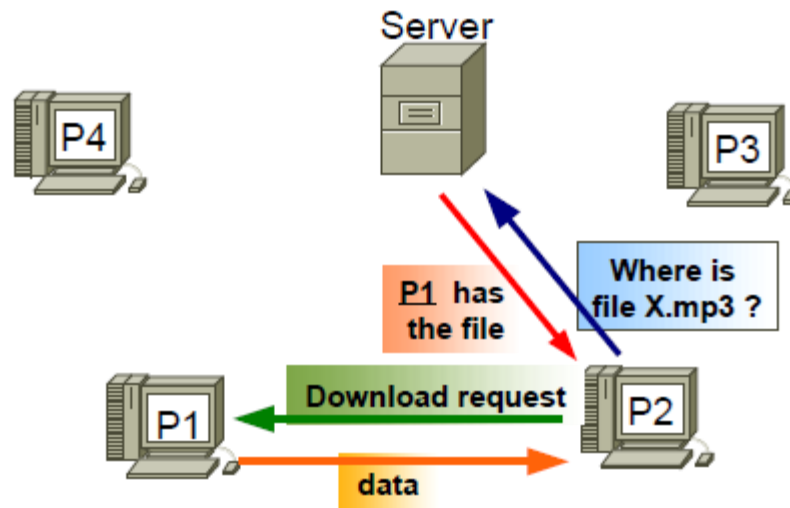
● 0 arquivos

● 0-100 Arquivos

- Rede popular de compartilhamento de arquivos de código aberto
 - ~450.000 usuários em 2003
 - ~2.000.000 de usuários em 2022
- Topologia baseada em Ultrapeer
 - Consultas inundadas entre ultrapeers
 - Nós folha protegidos do tráfego de consulta
 - Baseado em vários rastreadores

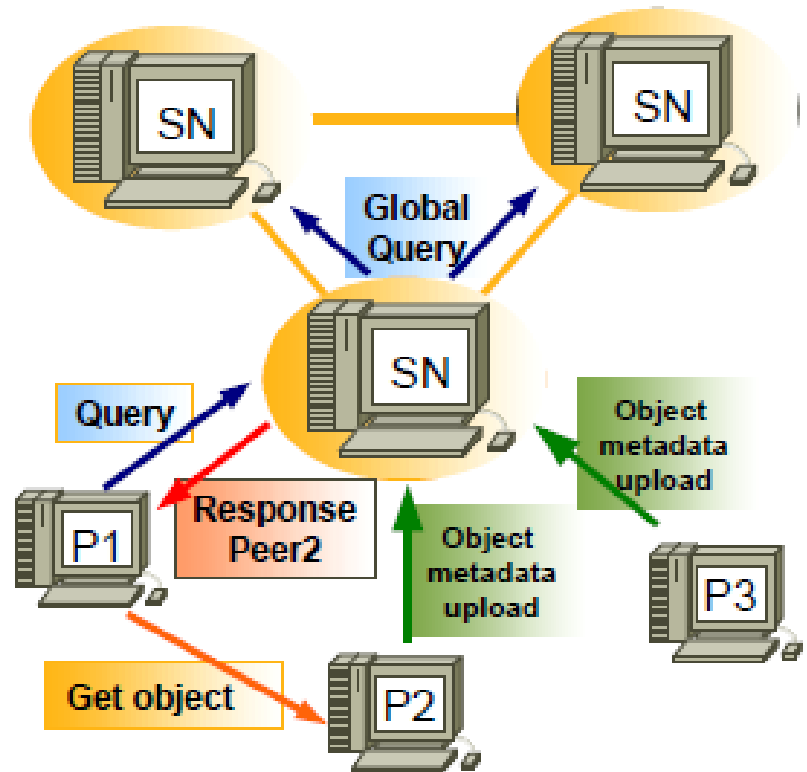
OpenNAP/Napster

- Os arquivos (música) estão na máquina cliente
- Servidores fornecem pesquisa (rendez-vous) e iniciam transferências diretas entre clientes
- OpenNAP é uma extensão para outros tipos e servidores de ligação.
- Arquitetura de Rede: Híbrida Não Estruturada.
- Algoritmo: Modelo de Diretório Centralizado (CDM)



FastTrack/KaZaA

- É uma extensão do protocolo Gnutella que adiciona supernós para melhorar a escalabilidade (~gnutella v.2)
 - Uma aplicação peer hospedada por uma máquina poderosa com uma conexão de rede rápida torna-se automaticamente um supernó, agindo efetivamente como um servidor de indexação temporário para outros peers mais lentos.
 - Comunicar-se entre si para satisfazer solicitações de pesquisa
- Arquitetura de Rede: Híbrida Não Estruturada.
- Algoritmo: Modelo de Solicitações Inundadas (FRM)



BitTorrent

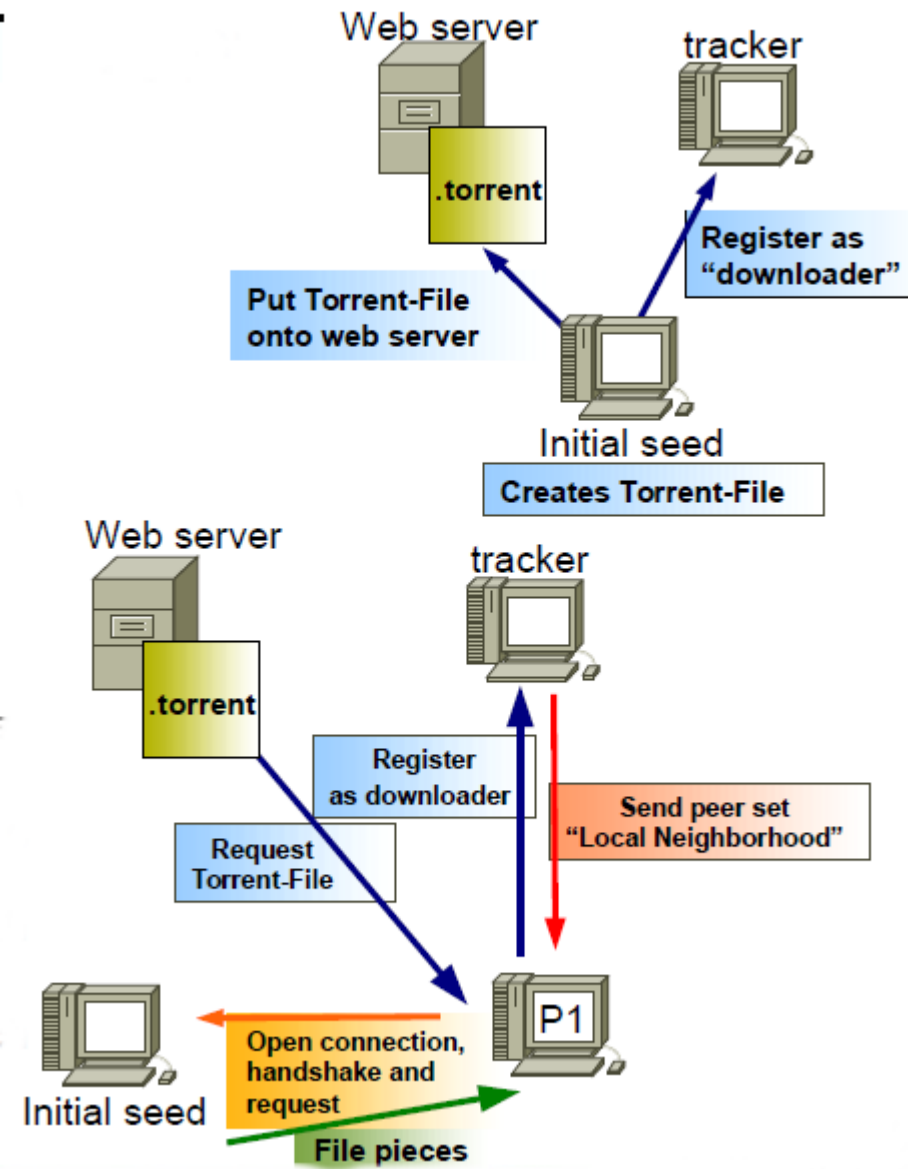
- O BitTorrent transfere parte do trabalho de rastreamento de arquivos para um servidor central denominado rastreador
- Usa um princípio chamado olho por olho
 - Para receber arquivos, você deve entregá-los
 - Resolve o problema da sanguessuga
- Permite download rápido de arquivos grandes usando largura de banda mínima da Internet
- . torrent: arquivo ponteiro que direciona o computador para o arquivo que deseja baixar
- Swarm: grupo de computadores baixando ou enviando simultaneamente o mesmo arquivo
- Tracker: servidor que gerencia o processo de transferência de arquivos BitTorrent

BitTorrent

- O software cliente BitTorrent se comunica com um rastreador para encontrar outros computadores executando o BitTorrent que possuem o arquivo completo (semeadores) ou que possuem uma parte do arquivo (atualmente baixando o arquivo)
- O rastreador identifica o enxame: este grupo de computadores
- O rastreador ajuda o software cliente a trocar partes do arquivo com outros computadores do enxame
- O computador recebe várias partes do arquivo simultaneamente
- Ao executar o software BitTorrent após a conclusão do download, outras pessoas podem receber o arquivo .torrent deste computador
 - Classificação mais elevada no sistema olho por olho

BitTorrent

- Rastreadores: monitoram o número de sementes/ pares; responsável por ajudar os downloaders a se encontrarem, usando um protocolo simples sobre HTTP.
- O Downloader envia informações de status para rastreadores, que respondem com listas de contatos informações para pares que estão baixando o mesmo arquivo.
- Os servidores Web não possuem informações sobre a localização do conteúdo
 - Armazenar apenas arquivos de metadados descrevendo os objetos (comprimento, nome, etc.) e associando a cada um deles a URL de um rastreador
- Arquitetura de rede: híbrida não estruturada
- Algoritmo: Modelo de Diretório Centralizado (CDM)
- torrents "trackerless" através de um sistema chamado "banco de dados distribuído", através de DHT (Distributed Hash Tables)



Sistema de arquivos interplanetário (IPFS)

<https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf>

https://ria.ua.pt/bitstream/10773/31279/1/Documento_Ricardo_Chaves.pdf

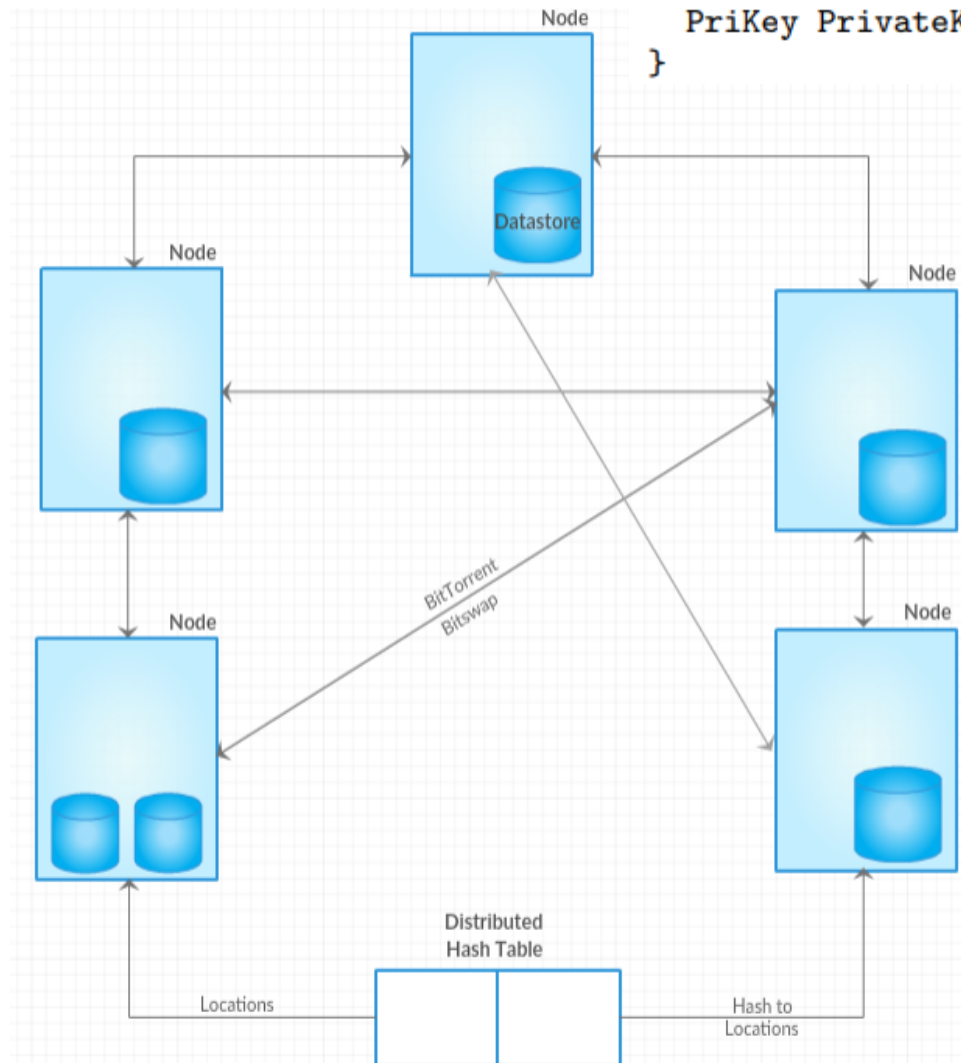
IPFS

- Sistema de arquivos distribuído globalmente: IPFS trata de descentralização de “distribuição”
- Identificação baseada em conteúdo com hash seguro de conteúdo
- Resolver locais usando Distributed Hash Table (DHT)
- Bloqueie trocas usando o popular protocolo de distribuição de arquivos peer-to-peer Bittorrent
- Troca de blocos incentivada usando *Troca de bits* protocolo
- Organização de arquivos baseada em versão Merkle DAG (Directed Acíclico Graph), semelhante ao sistema de controle de versão Git
- Servidores de autocertificação para nós de armazenamento para segurança

IPFS

- Arquivos em armazenamento distribuído
- Tabela hash distribuída, usa o hash do arquivo como chave para retornar o localização do arquivo.
- Uma vez determinada a localização, a transferência ocorre peer-to-peer como uma transferência descentralizada.

```
type Node struct {  
    NodeId NodeID  
    PubKey PublicKey  
    PriKey PrivateKey  
}
```



IPFS Architecture

IPFS: Troque os blocos

Os nós pares que contêm os blocos de dados são incentivados por um protocolo chamado BitSwap.

Os nós pares têm um *lista_de_queridos_e_tenho_lista*

Qualquer desequilíbrio é observado na forma de crédito e dívida

BitSwap

O protocolo BitSwap gerencia as trocas de blocos envolvendo os nós de acordo

Os nós da rede devem, portanto, fornecer valor na forma de blocos.

Se você enviar um bloco, receberá um token IPFS que pode ser usado quando precisar de um bloco.

O protocolo BitSwap tem disposições para lidar com exceções, como nó de carregamento livre, nó sem querer nada, nó sem nada.

Cálculo de troca de bits

- Índice de dívida $r = \frac{\text{bytes_sent}}{\text{bytes_recv} + 1}$
- Probabilidade de envio para um devedor $P(\text{send} | r) = 1 - \frac{1}{1 + \exp(6 - 3r)}$
 - A função cai rapidamente à medida que o rácio de endividamento dos nós ultrapassa o dobro do crédito estabelecido
- Os nós BitSwap mantêm livros contábeis contabilizando as transferências com outros nós
 - Ao ativar uma conexão, os nós BitSwap trocam suas informações contábeis. Se não corresponder exatamente, o razão é reinicializado do zero, perdendo o crédito ou dívida acumulada.

```
type Ledger struct {  
    owner      NodeId  
    partner    NodeId  
    bytes_sent int  
    bytes_recv int  
    timestamp  Timestamp  
}
```

Cálculo de troca de bits

- Esboço do tempo de vida de uma conexão peer:
 - 1. Aberto: os pares enviam livros até concordarem.
 - 2. Envio: peers trocam listas de desejos e blocos.
 - 3. Fechar: peers desativam uma conexão.
 - 4. Ignorado: (especial) um par é ignorado (durante um tempo limite) se a estratégia de um nó evitar o envio

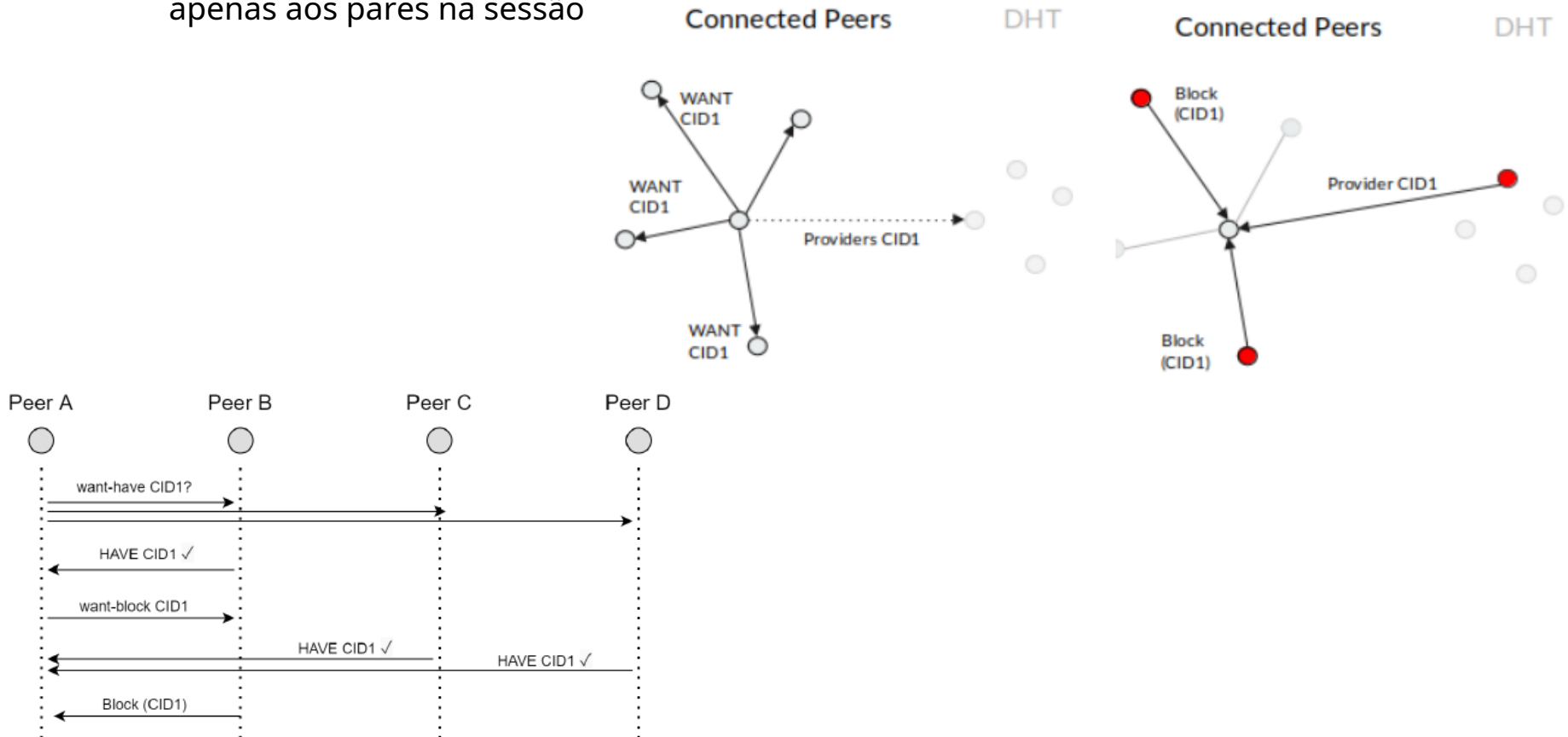
```
// Protocol interface:  
interface Peer {  
    open (nodeid :NodeId, ledger :Ledger);  
    send_want_list (want_list :WantList);  
    send_block (block :Block) -> (complete :Bool);  
    close (final :Bool);  
}
```


ID do conteúdo

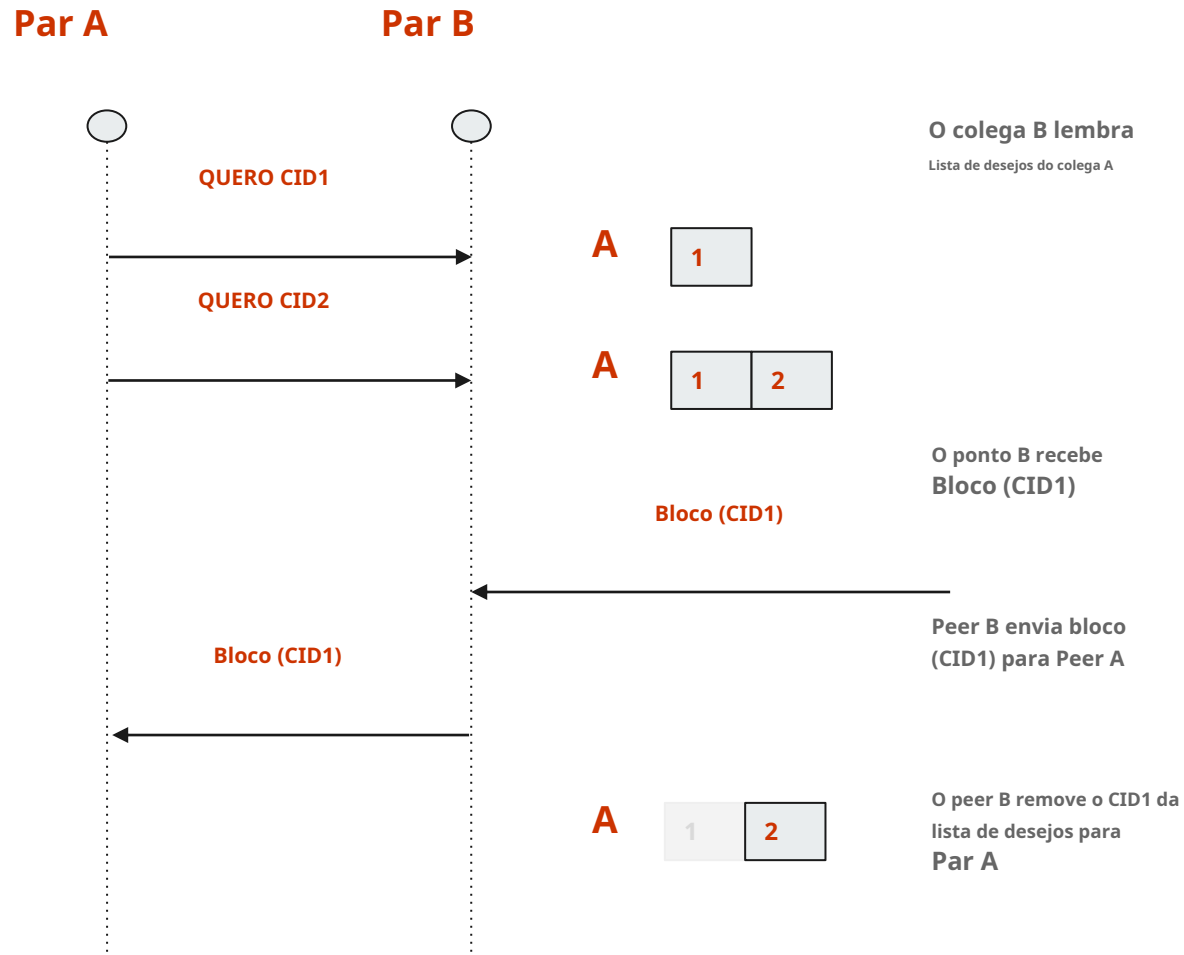
- No IPFS, todo arquivo ou diretório possui um CID
 - Hash SHA256 exclusivo usado para identificar o arquivo.
- Sempre que o conteúdo muda, o CID também muda.
- Para acompanhar os arquivos após serem alterados, o IPFS utiliza o InterPlanetary Name System (IPNS) onde o nome é o hash de uma chave pública, armazenada no DHT, apontando para o CID da versão mais recente

Exemplos de troca de bits

- Quando um peer deseja um bloco, ele transmite um Want para todos os peers conectados
- Se não houver resposta, pergunta ao DHT quem possui o CID raiz. Os pares que respondem são adicionados à sessão e as solicitações subsequentes são enviadas apenas aos pares na sessão



Exemplos de troca de bits

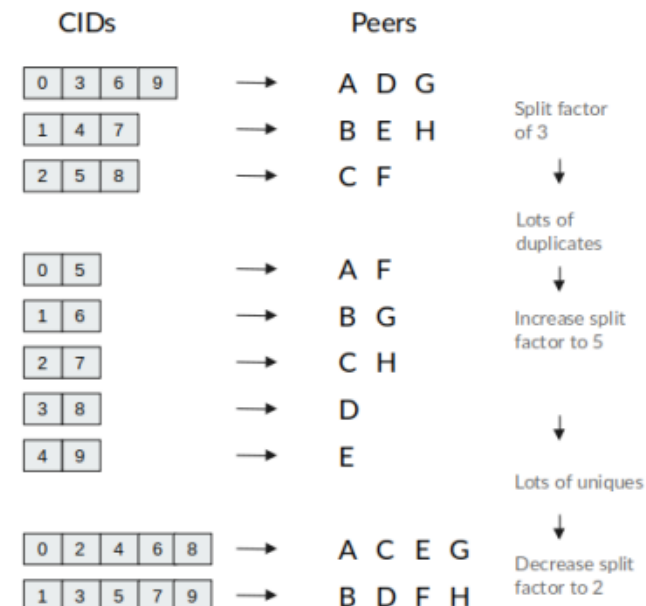


IPFS: fator de divisão

- Quando o nó local recebe um bloco, ele transmite uma mensagem de cancelamento do CID desse bloco para todos os pares conectados.
- No entanto, o cancelamento não pode ser processado pelo peer destinatário antes de enviar o bloco → duplicatas
- O nó local monitora a proporção de duplicatas/ blocos recebidos e ajusta a divisão

fator.

- Se a proporção for superior a 4 (um grande número de duplicatas), o fator de divisão será aumentado - o mesmo CID será enviado para menos pares.
- Se a proporção for inferior a 2 (poucas duplicatas) o fator de divisão é diminuído - o mesmo CID será enviado para mais pares.



IPFS: Cluster

- O cluster facilita a replicação de conteúdo em vários nós
- Todos os pares do cluster precisam compartilhar o mesmo segredo do cluster para fazer parte do mesmo cluster
 - Cada peer do cluster possui seu próprio ID exclusivo.
- Quando novos dados são adicionados e fixados em um dos pares do cluster, todos os outros pares desse cluster recebem os dados.
- O par responsável por iniciar o cluster é aquele que cria o segredo do cluster e é selecionado como líder do cluster.
- Cada peer do cluster é capaz de modificar, adicionar ou remover dados do cluster.
- Quando um par é adicionado ou removido do cluster, o cluster continua funcionando normalmente.
- Caso o líder do nó caia, um novo líder é eleito com base em um algoritmo de consenso RAFT.
- Facilita o gerenciamento de grupos que receberão o mesmo conteúdo, por exemplo, atualizações de software, conteúdo multicast, conteúdo baseado em localização

Algoritmo de consenso RAFT

- Uma eleição de líder é iniciada por um servidor candidato.
- Um servidor torna-se candidato se não receber nenhuma comunicação do líder durante um período chamado tempo limite de eleição (normalmente $3 \times 100\text{ms}$), portanto ele assume que não há mais nenhum líder em exercício.
- Inicia a eleição aumentando o contador de mandatos, votando em si mesmo como novo líder e enviando mensagem a todos os demais servidores solicitando seu voto.
- Um servidor votará apenas **uma vez por período**, com um **primeiro a chegar, primeiro a ser servido** base.
- Se um candidato receber uma mensagem de outro servidor com um número de mandato maior que o mandato atual do candidato, a eleição do candidato será derrotada e o candidato se transformará em seguidor e reconhecerá o líder como legítimo.
- Se um candidato obtiver a maioria dos votos, ele se tornará o novo líder.
- Se nada disso acontecer, por exemplo, devido a uma votação dividida, então inicia-se um novo mandato e inicia-se uma nova eleição.

IPFS: localizando nós

Os nós são identificados por hashes criptográficos de chave pública

Eles guardam os objetos que formam os arquivos a serem trocados

Os objetos são identificados por um hash seguro e um objeto pode conter subobjetos, cada um com seu próprio hash, que é usado na criação do hash raiz do objeto.

```
type Node struct {  
    NodeId NodeID  
    PubKey PublicKey  
    PriKey PrivateKey  
}
```

```
n.PubKey, n.PrivKey = PKI.genKeyPair()  
n.NodeId = hash(n.PubKey)
```

IPFS: Localizando objetos

O IPFS identifica os recursos por um hash.

Em vez de identificar o recurso pela sua localização como no HTTP, o IPFS identifica-o pelo seu conteúdo ou pelo hash seguro do seu conteúdo.

Como resolver a localização?

Envie uma solicitação para qualquer pessoa com um recurso com o identificador de hash

A parte de roteamento do protocolo IPFS mantém um DHT para localizar os nós, bem como para objetos de arquivo.

Um DHT simples mantém o hash como chave e a localização como valor.

A chave pode ser hash diretamente no local. O DHT resolve para o local mais próximo do valor-chave.

IPFS: Objetos

- Fixação de objetos: os nós que desejam garantir a sobrevivência de objetos específicos podem fazê-lo fixando os objetos.
 - Os objetos são mantidos no armazenamento local do nó.
- Publicação de objetos: DHT, com endereçamento hash de conteúdo, permite publicar objetos de forma distribuída
 - Qualquer pessoa pode publicar um objeto simplesmente adicionando sua chave ao DHT, adicionando-se como um par e fornecendo a outros usuários o caminho do objeto.
 - Novas versões têm hash diferente e, portanto, são novos objetos. Rastrear versões é tarefa de objetos de controle de versão adicionais

Como conectar um nó IPFS para a rede p2p?

- O arquivo de configuração (\$IPFS_PATH/config) de cada nó IPFS possui uma lista de endereços de bootstrap

O IPFS vem com uma lista padrão de peers confiáveis, mas pode ser modificado para atender a outras necessidades. Um uso popular para uma lista de bootstrap personalizada é criar uma rede IPFS pessoal.

```
"Bootstrap": [  
  "/dnsaddr/bootstrap.libp2p.io/p2p/QmcZf59b...gU1ZjYZcYW3dwt",  
  "/ip4/104.131.131.82/tcp/4001/p2p/QmaCpDMG...UfsmvsqQLuvuJ",  
  "/ip4/104.131.131.82/udp/4001/quic/p2p/Qma...UfsmvsqQLuvuJ",  
  "/dnsaddr/bootstrap.libp2p.io/p2p/QmNnooD5...BMjTezGAJN",  
  "/dnsaddr/bootstrap.libp2p.io/p2p/QmQCU2Ec...J16u19uLTa",  
  "/dnsaddr/bootstrap.libp2p.io/p2p/QmbLHANM...Ucqanj75Nb"  
],
```

- Lista de peers com os quais o daemon IPFS aprende sobre outros peers na rede
 - Executando o comando daemon IPFS
 - Primeiro estabeleça uma conexão p2p com nós de bootstrap do Protocol Labs (empresa por trás do IPFS)
 - Através desses nós de bootstrap, ele encontrará ainda centenas de outros pares
 - Os peers conversarão através de TCP, UDP na porta: **4001**

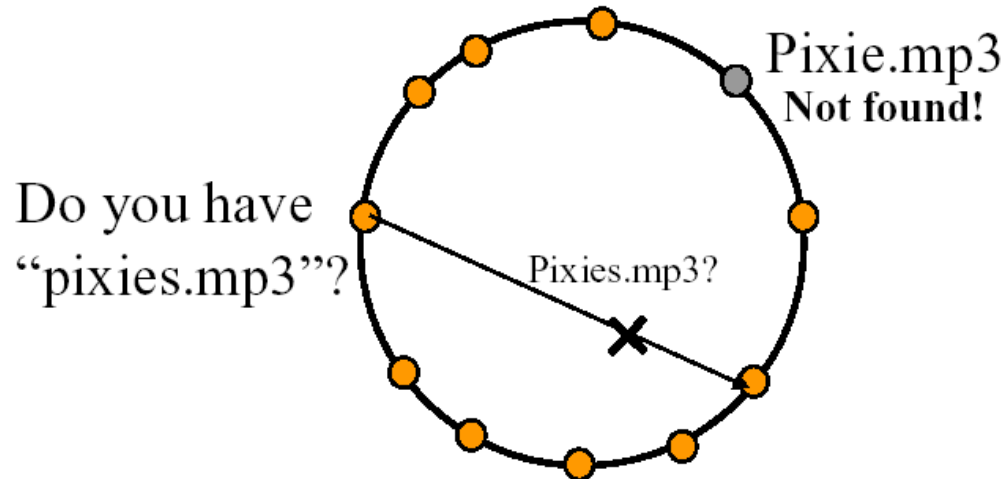
Tabelas Hash Distribuídas (DHT)

<https://www.cs.cmu.edu/~dga/15-744/S07/lectures/16-dht.pdf>

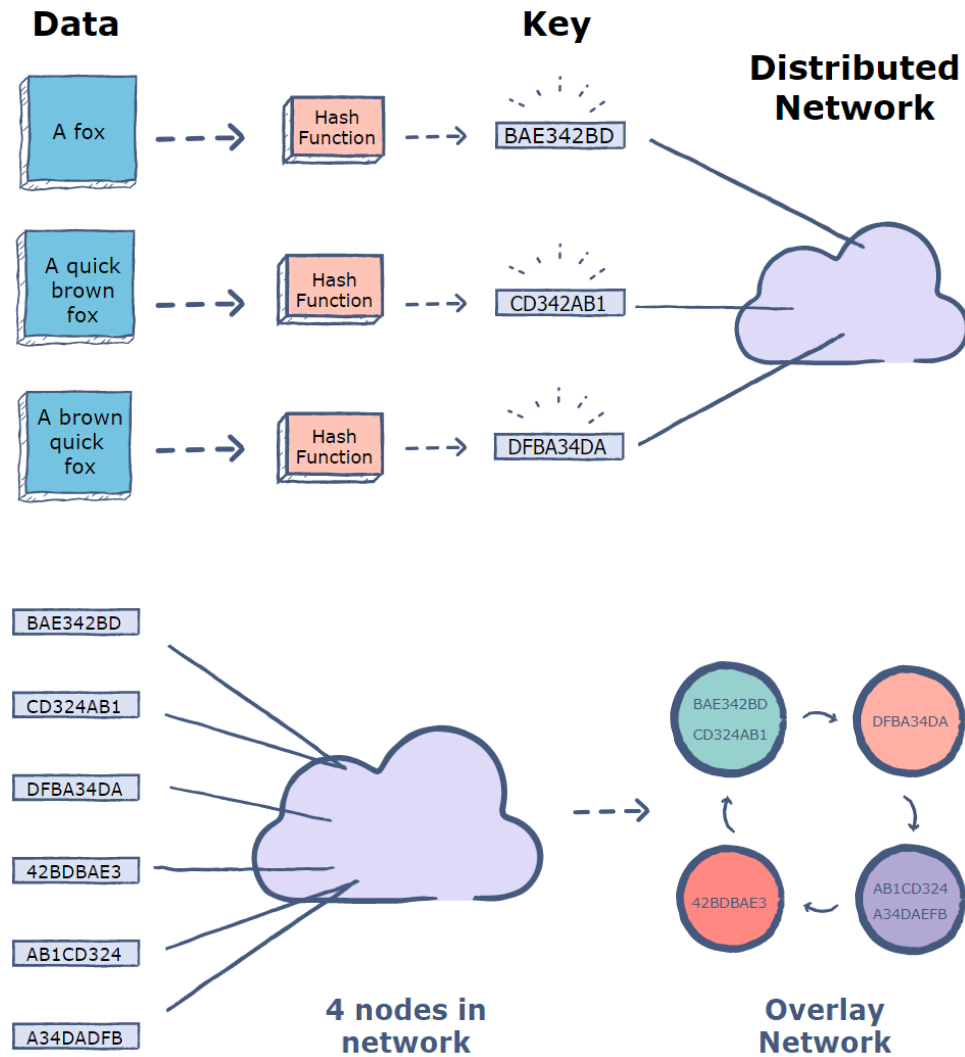
<https://pdos.csail.mit.edu/papers/ton:chord/paper-ton.pdf>

Pesquisando em DHTs (estruturada)

- Precisa saber o nome exato do arquivo
 - Chaves (nomes de arquivos) mapeadas para ids de nó
 - Alteração no nome do arquivo → pesquisar em nós diferentes
 - Sem correspondência de caracteres curinga: não é possível solicitar o arquivo “pix*”

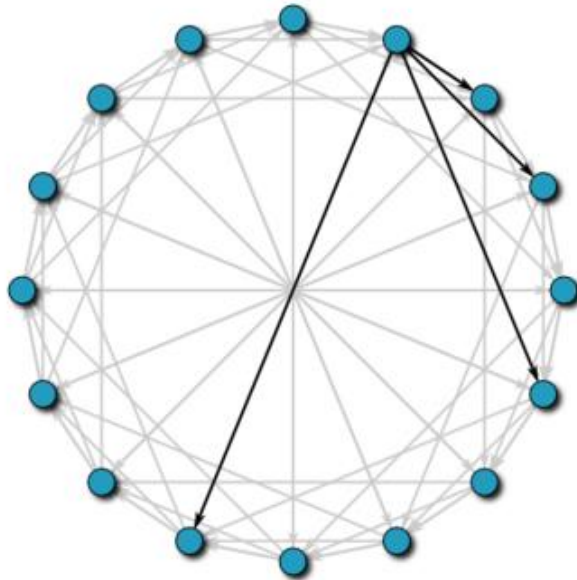


DHT

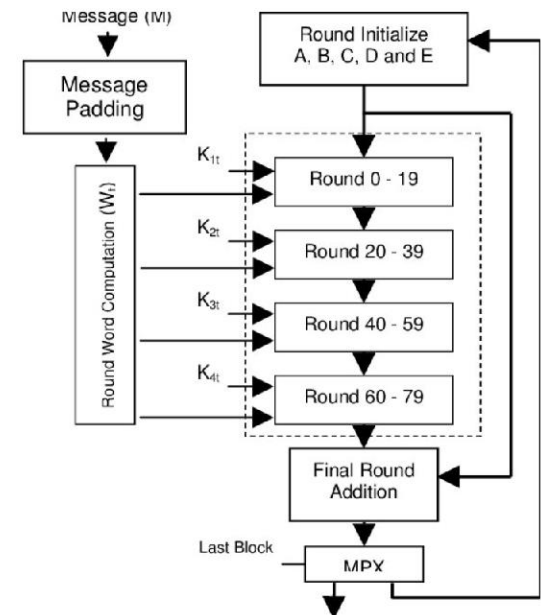


Acorde: Algoritmo DHT

- Todos os arquivos/itens de dados na rede terão um identificador, que será criptografado para fornecer uma chave para aquele recurso específico
- Se um nó precisar de um arquivo/dados, ele fará um hash de seu nome e enviará uma solicitação usando esta chave.
- Todos/nos nós também usam a função para fazer hash de seus endereços IP e, conceitualmente, os nós formarão um anel em ordem crescente de seu IP com hash



SHA-1

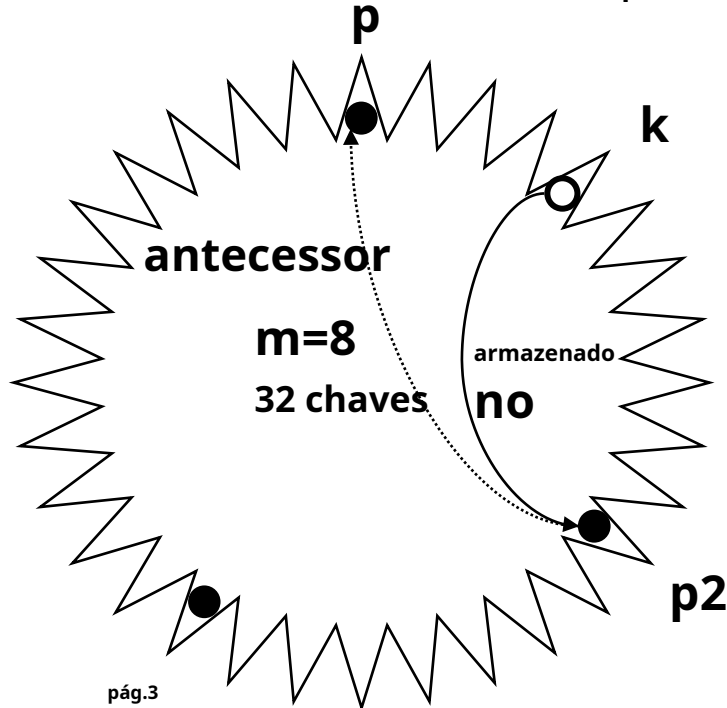


Acorde: Algoritmo DHT

- O nó sucessor de uma chave k é o primeiro nó cujo ID é igual a k ou segue k no círculo identificador, indicado por $\text{sucessor}(k)$
- Cada chave é atribuída ao seu nó sucessor, portanto, procurar uma chave k é consultar o sucessor (k) .
- Quando um nó deseja compartilhar um arquivo ou alguns dados
 - Faz hash do identificador para gerar um **chave k** , e envia seu **PI** e a **identificador de arquivo** para **$\text{sucessor}(k)$**
 - Estes são então armazenados neste nó
 - Todos os recursos são indexados em um grande DHT em todos os nós participantes
 - Se houver dois ou mais nós que contêm um determinado arquivo ou recurso, **as chaves serão armazenadas no mesmo nó** no DHT, dando ao nó solicitante a opção de solicitar o arquivo em um ou outro nó, ou em ambos

DHT: Armazenar informações

- Hashing de chaves de pesquisa E endereços de pares em chaves binárias de comprimento m
 - por exemplo, $m=8$, $\text{key}(\text{"jingle-bells.mp3"})=17$, $\text{key}(196.178.0.1)=3$
- Calcula hash de dados para obter k
- O roteamento é usado para encontrar o nó que armazena a chave k (nó_k)
- As chaves de dados são armazenadas na próxima chave de nó maior

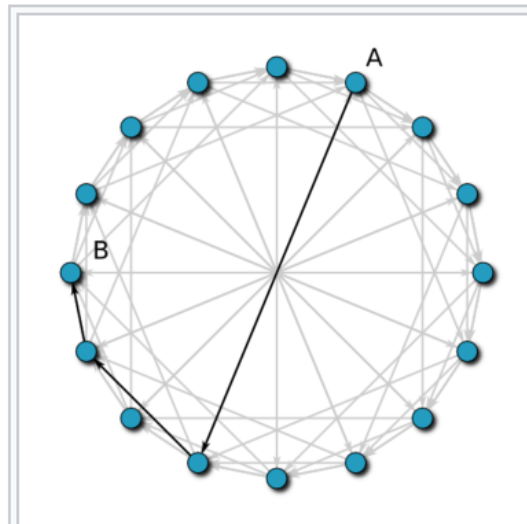



Possibilidades de pesquisa

1. cada peer conhece todos os outros tamanhos de tabela de roteamento $O(n)$
2. Os pares conhecem o custo de pesquisa do sucessor $O(n)$

DHT: Informações de pesquisa

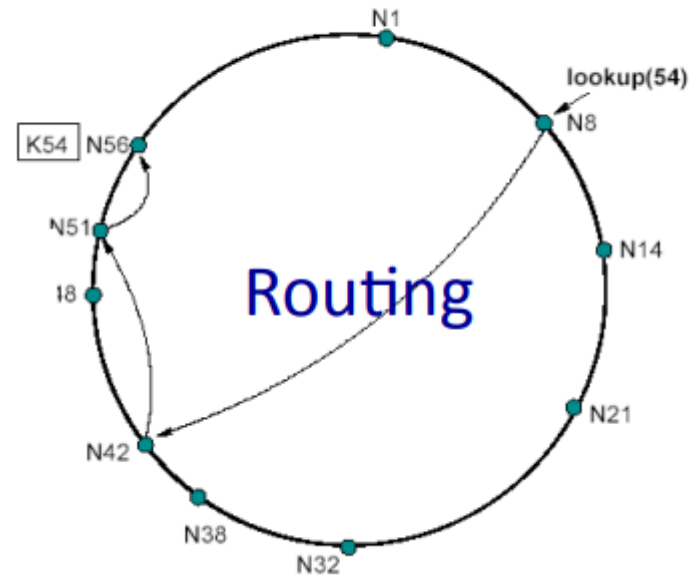
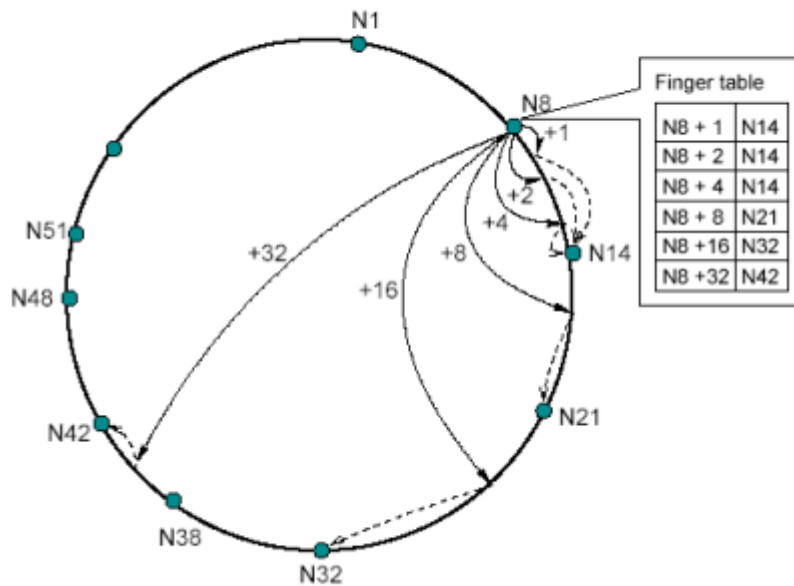
- Quando um nó deseja um conteúdo
 - Faz hash do identificador de dados e envia uma solicitação ao sucessor (k)
 - Responder com o IP do nó que contém os dados reais
 - Como um nó solicita informações do sucessor (k), quando não sabe seu IP, mas apenas a chave?
 - Cada nó contém o que é conhecido como tabela de dedos
 - Contém uma lista de chaves e seus IPs sucessores
 - Cada nó contém o IP de uma sequência exponencial de nós que o seguem, ou seja, entrada e do nó k a tabela de dedos contém o IP do nó $k+2^e$



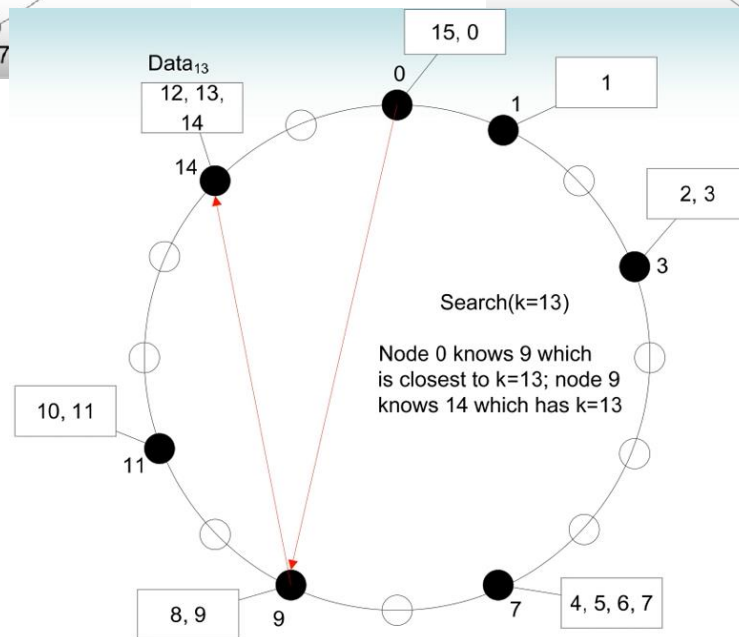
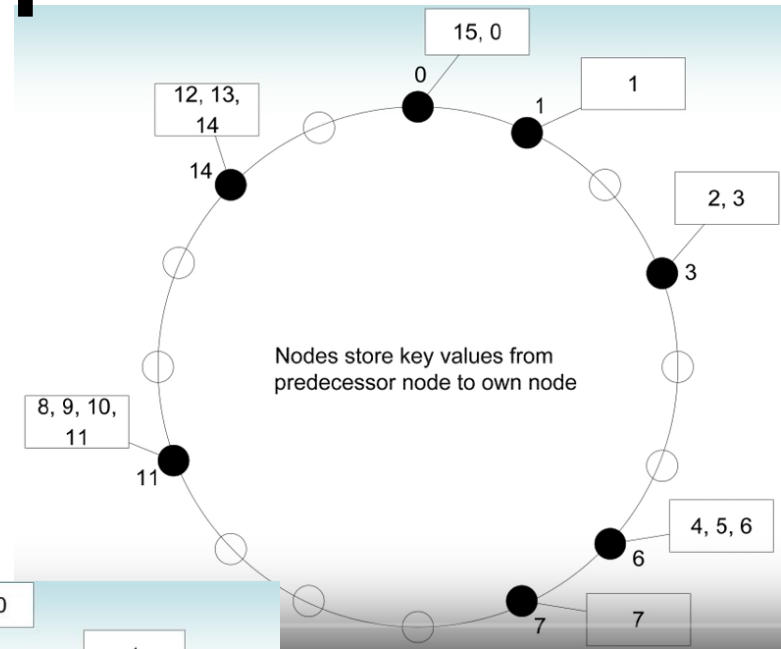
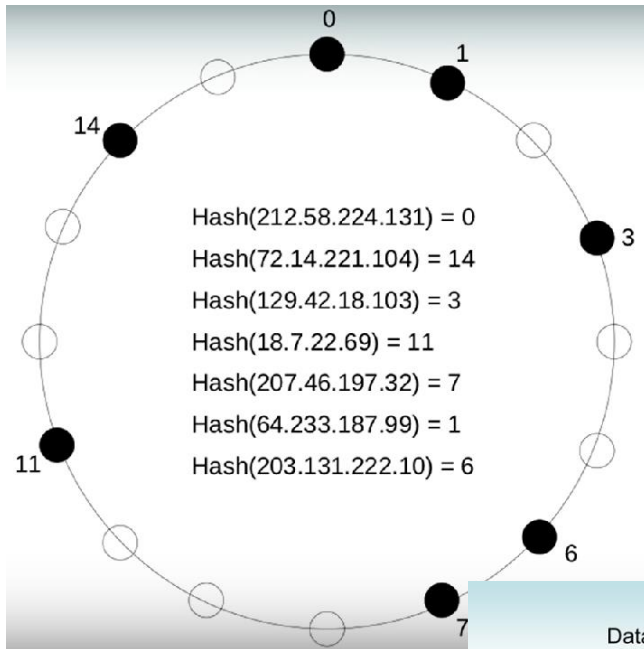
The routing path between nodes A and B. 
Each hop cuts the remaining distance in half (or better).

58 Informações de pesquisa: Dedo Mesa

Routing Tables



Exemplo



Pesquisa de arquivos: inundação vs. DHTs

- Lembrar
 - Inundações podem perder arquivos
 - DHTs nunca devem
- Complexidade da consulta
 - Flooding pode lidar com lógica arbitrária de site único
 - DHTs podem fazer equijoins, seleções, agregações, etc.
 - Mas não tão bom em seleções sofisticadas como curingas
- Desempenho de consulta
 - As inundações podem ser lentas para encontrar coisas, consomem muito BW
 - DHTs: caro para publicar documentos com muitos termos
 - DHTs: caro para cruzar listas de longo prazo
 - Mesmo que a produção seja muito pequena!
- Solução híbrida!

Pesquisa Híbrida

Híbrido = “O melhor dos dois mundos”



Segurança

Segurança - ataques

- Ataques de envenenamento
 - por exemplo, fornecer arquivos cujo conteúdo é diferente da descrição
- Ataques poluentes
 - por exemplo, inserir pedaços/pacotes "ruins" em um arquivo válido na rede
- Sanguessuga
 - Usuários ou software que fazem uso da rede sem contribuir com recursos para ela
- Inserção de vírus nos dados transportados
 - por exemplo, arquivos baixados ou transportados podem estar infectados com vírus ou outro malware
- Malware no próprio software de rede ponto a ponto
 - por exemplo, software distribuído pode conter spyware
- Ataques de negação de serviço
 - Ataques que podem fazer com que a rede funcione muito lentamente ou quebre completamente
- Filtragem
 - Os operadores de rede podem tentar impedir que dados de rede peer-to-peer sejam transportados
- Ataques de identidade
 - por exemplo, rastrear os usuários da rede e assediá-los ou atacá-los legalmente
- Spam
 - por exemplo, envio de informações não solicitadas pela rede - não necessariamente como um ataque de negação de serviço

Segurança

- A maioria dos ataques pode ser derrotada ou controlada pelo design cuidadoso da rede peer-to-peer e pelo uso de criptografia.
 - No entanto, quase todas as redes falharão quando a maioria dos pares tentar danificá-la.
- Anonimato
 - Alguns protocolos peer-to-peer (como o Freenet) tentam ocultar a identidade dos usuários da rede, passando todo o tráfego através de nós intermediários.
- Criptografia
 - Algumas redes peer-to-peer criptografam os fluxos de tráfego entre peers
 - Tornar mais difícil para um ISP detectar que a tecnologia peer-to-peer está sendo usada (pois alguns limitam artificialmente a largura de banda)
 - Ocultar o conteúdo do arquivo de bisbilhoteiros
 - Impedir esforços de aplicação da lei ou censura de certos tipos de material
 - Autenticar usuários e evitar ataques "man in the middle" em protocolos
 - Ajuda na manutenção do anonimato