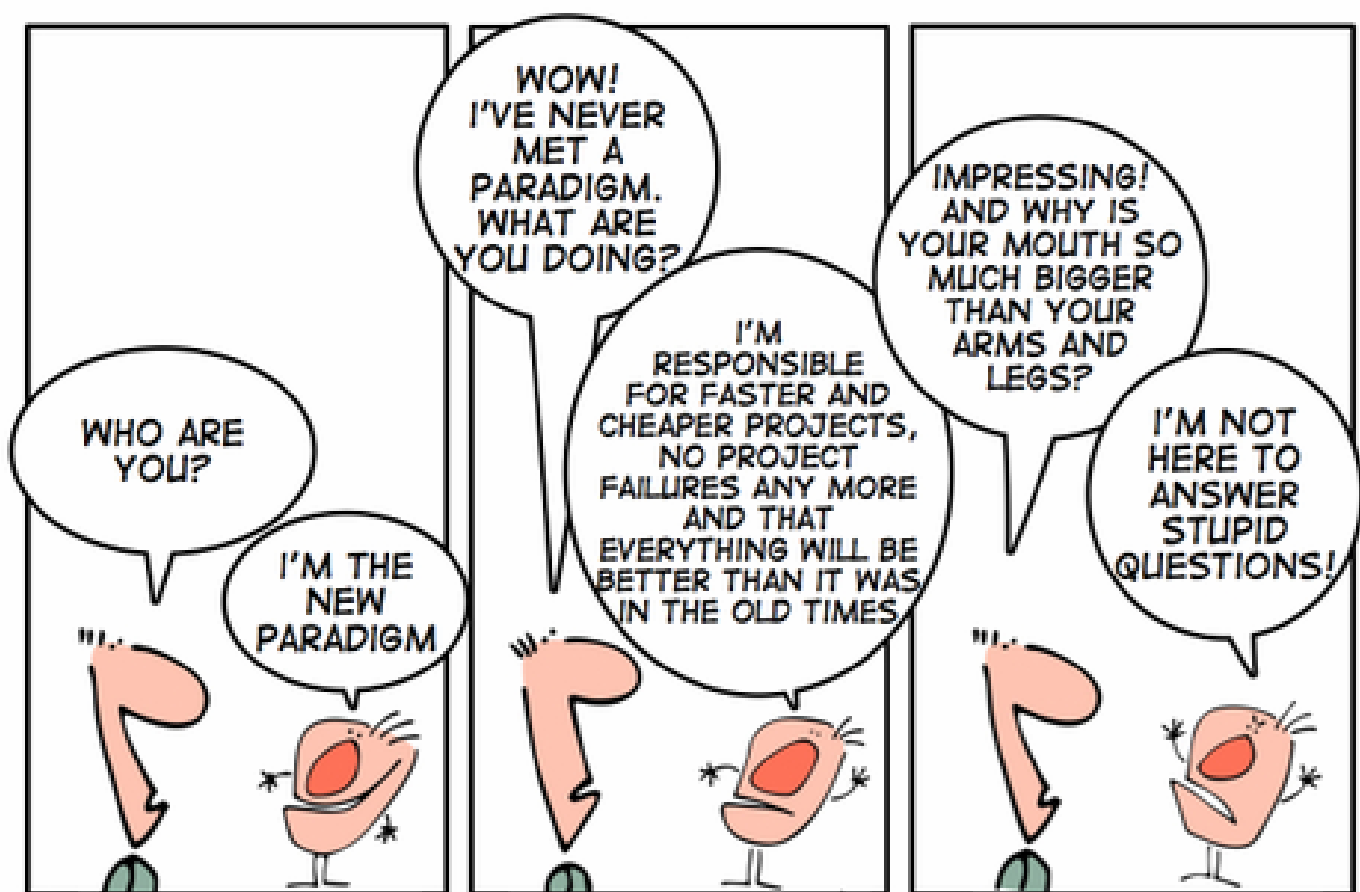


# **Projeto de Sistemas Distribuídos**

Introdução à Paradigmas  
Nomes e Endereços

*Prof. Msc. Marcelo Iury de Sousa Oliveira*  
*marceloiury@gmail.com*  
*<http://sites.google.com/site/marceloiury/>*



THE NEW PARADIGM - PART 1

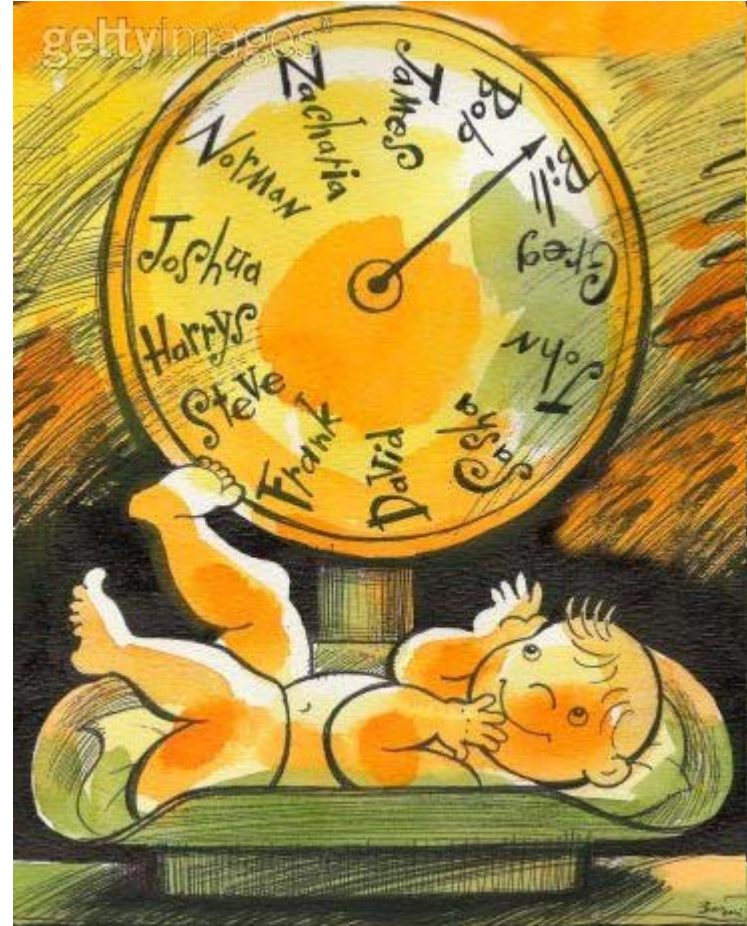
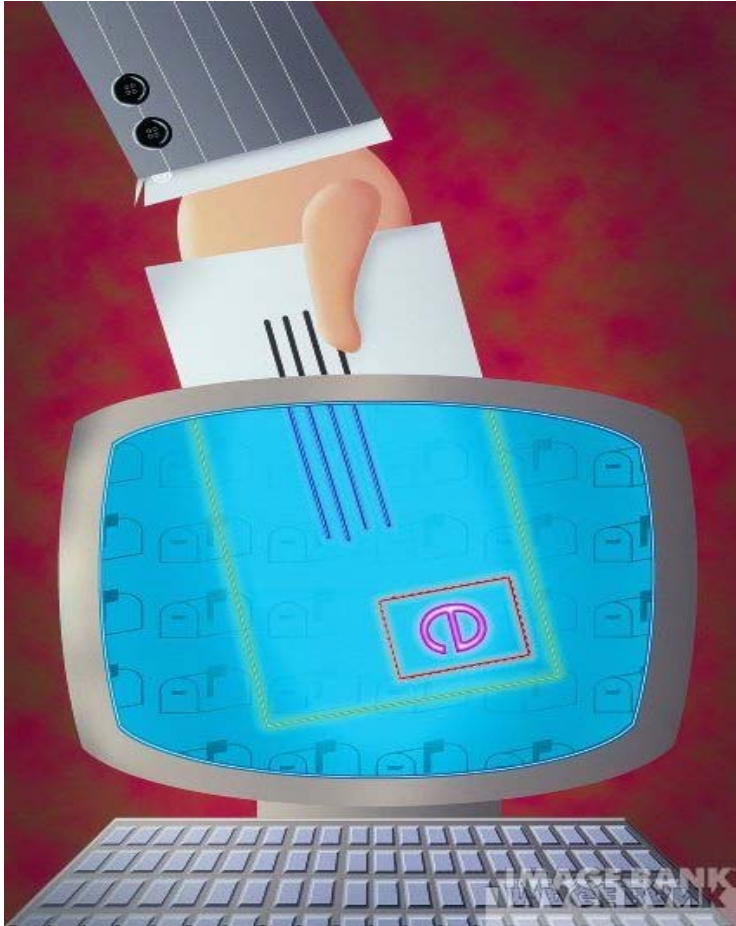
*geek and probe*

- Definições de **paradigmas** na internet:
  - Paradigma (do grego Parádeigma) literalmente modelo, é a representação de um padrão a ser seguido. ...  
[pt.wikipedia.org/wiki/Paradigmas](http://pt.wikipedia.org/wiki/Paradigmas)
  - paradigma - **um modelo ou exemplo; as inflexões de uma palavra; um modo de se pensar sobre, encarar algo ou agir de acordo com algum contexto; um sistema de ...**  
[pt.wiktionary.org/wiki/paradigma](http://pt.wiktionary.org/wiki/paradigma)
  - paradigma - Um exemplo que serve como modelo; padrão  
[vocesa.abril.com.br/informado/aberto/ar\\_164940.shtml](http://vocesa.abril.com.br/informado/aberto/ar_164940.shtml)
  - **Modelo, padrão de regras, segundo as quais as pessoas procuram solucionar seus problemas e obter sucesso.**  
[www.eps.ufsc.br/disserta99/berger/glossario.html](http://www.eps.ufsc.br/disserta99/berger/glossario.html)
  - paradigma - sm 1. Modelo, protótipo. 2. Gram. Conjunto de fórmulas de palavras variáveis que servem de modelo para as demais do mesmo grupo mórfico.  
[www.guia.heu.nom.br/dicion%C3%A1rio.htm](http://www.guia.heu.nom.br/dicion%C3%A1rio.htm)

# Principais Paradigmas

- **Nomes e Endereços**
- Trocas de Mensagens
- Operações Remotas
- Comunicação em Grupo
- Time e Clocks
- Sincronismo
- Ordenação
- Coordenação
- Consistência
- Concorrência
- Atomicidade

# Nomes e Endereços



# Nomes e Endereços



**Qual a diferença entre um  
e outro? E como é a  
relação entre eles?**



## CNAME RECORDS WWW.GOOGLE.COM

canonical name = [www.l.google.com](http://www.l.google.com)

## ANDERE RECORDS WWW.GOOGLE.COM

a.l.google.com internet address = 74.125.53.9

b.l.google.com internet address = 74.125.45.9

e.l.google.com internet address = 209.85.137.9

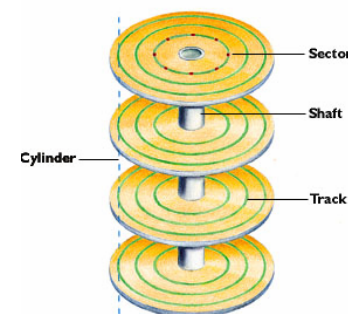
d.l.google.com internet address = 74.125.77.9

f.l.google.com internet address = 72.14.203.9

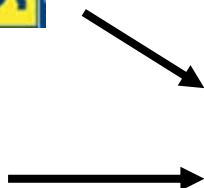
g.l.google.com internet address = 74.125.95.9



*Tracks, Cylinders, and Sectors*







## CNAME RECORDS WWW.GOOGLE.COM

canonical name = [www.l.google.com](http://www.l.google.com)

## ANDERE RECORDS WWW.GOOGLE.COM

a.l.google.com internet address = 74.125.53.9

b.l.google.com internet address = 74.125.45.9

e.l.google.com internet address = 209.85.137.9

d.l.google.com internet address = 74.125.77.9

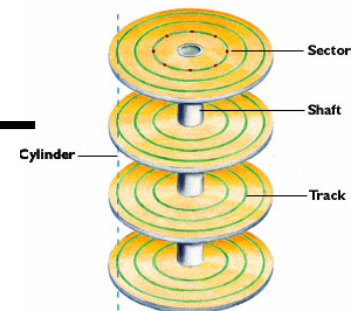
f.l.google.com internet address = 72.14.203.9

g.l.google.com internet address = 74.125.95.9

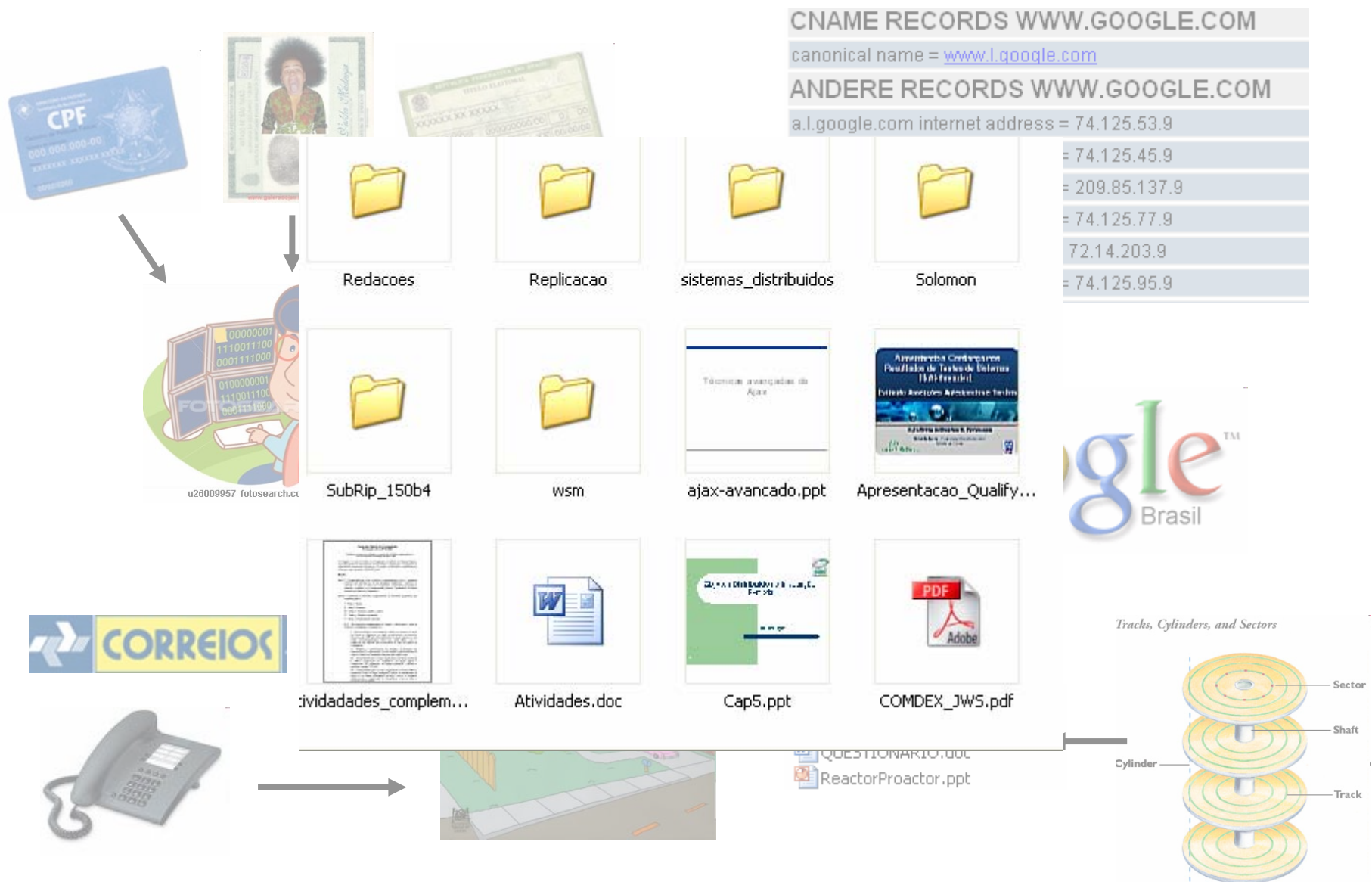


Tracks, Cylinders, and Sectors

paper obe-v5.doc  
Proj CommuneBackup- v2.doc  
QUESTIONÁRIO.doc  
ReactorProactor.ppt







# Nomes e Endereços - Para quê?

- Endereços são úteis para localizar componentes
  - São usados para interagir com uma entidade com a qual o nome refere-se
- Nomes, além de serem mais fáceis de serem lembrados
  - São independentes do protocolo e da localização do componente
  - Podem ter seu significado mudado ao longo do tempo
- Nomes únicos globalmente e contexto de nomes
- Nomes “puros” vs “impuros”
- Níveis de abstração
  - [marceloiury@gmail.com](mailto:marceloiury@gmail.com) é um nome ou um endereço?
- Pode-se usar endereços como nomes?

# Nomes, Identificadores e Endereços

- **Nome:** cadeia de bits ou caracteres usada para referenciar uma entidade.
- **Entidade:** “praticamente qualquer coisa”: hospedeiros, impressoras, discos, arquivos.
- **Ponto de Acesso:** um tipo de entidade especial que possibilita acesso para agir sobre uma entidade.
- **Endereço:** Nome de um ponto de acesso.

Exemplo: Google

Nome: *www.google.com*

Entidade: *serviço de busca da Google*

Ponto de acesso: *www*

Endereço: *64.233.169.104*

# Identificadores

- Um nome que tem as seguintes propriedades:
  1. Um identificador referencia, no máximo, uma entidade;
  2. Cada entidade é referenciada por, no máximo, um identificador;
  3. Um identificador sempre referencia a mesma entidade, isto é, nunca é reutilizado.
    - Endereço Ethernet: cadeia aleatória de 48 bits;
    - Endereços de memória: cadeias de 32 ou 64 bits.

# Tipos de Nome

- Nome Orientados a Humanos
  - Nomes de alto nível
  - Nome de arquivos
- Nomes Orientados a Sistemas
  - Nomes de baixo nível
  - ID
- Nomes de alto nível não precisam ser únicos, os de baixo nível, sim
  - Tabelas de Vinculação Nome-Endereço
    - Nome -> Endereço
- Alto nível tem que ser prático para uso humano ao passo que baixo nível tem que ser prático (desempenho) para o sistema

# Tipos de Endereço

- Endereço Lógico de Grupo:
  - permite interagir com o grupo como um todo sem necessidade de conhecer endereços individuais
  - requer o uso de protocolos de comunicação em grupo -capazes de reconhecer o endereço
- Endereços específicos de protocolos
  - IP - endereço independente da arquitetura ou sistema operacional da máquina
  - Multicast-IP - o emissor não precisa se preocupar se os receptores estão ativos



**De Nome e Endereços já  
nós entendemos, mas  
qual a diferença em  
relação ao mundo não  
distribuído?**



# Mapeamento de nomes

- Serviços de nomes traduzem nomes em endereços
  - Binding
    - Associa um nome a um endereço
  - Lookup
    - Recupera (resolve) um endereço a partir de um nome
  - Unbiding
    - Desassocia um nome a um endereço
- Como nomes simples podem ser resolvidos?
  - **Broadcasting e multicasting**
  - **Ponteiros repassadores**

# Broadcasting e Multicasting

- **Broadcasting** – uma mensagem que contém o identificador da entidade é enviada por broadcast a cada máquina da rede e cada uma delas deve verificar se tem essa entidade. Somente as máquinas que podem oferecer AP para a entidade respondem com mensagem que contém o endereço daquele AP. (Usado nos protocolos **ARP**)
  - Ineficiente quando a rede cresce
- **Multicasting** – Envia apenas para grupos específicos, limitando o tráfego;
  - Também usado para associar réplicas e localizar a réplica mais próxima.

# Ponteiros e Repassadores

- Usada para localização de entidades móveis:
  - Quando uma entidade se move de A para B, deixa em A uma referência à sua nova localização, em B.
  - Uma entidade é localizada por exemplo, usando um serviço tradicional de nomeação, e o cliente pode consultar o endereço corrente da entidade percorrendo uma cadeia de ponteiros repassadores.
    - Se não houver precauções, a cadeia para uma entidade de alta mobilidade torna o custo de localização proibitivo;
    - Todas localizações intermediárias terão de manter sua parte da cadeia de ponteiros repassadores pelo tempo necessário;
    - Enlaces rompidos geram vulnerabilidade.
- O que acontece em caso de queda de um repassador?

# Abordagens baseadas na localização nativa

- Uma **localização nativa** monitora a localização corrente de uma entidade.
  - Usada como mecanismo de emergência para localização de serviços baseada em ponteiro de repassadores.
  - Ex.: Funcionamento do Mobile IP (mIPv6), onde o HoA é o IP fixo (localização nativa) que dirige os pacotes ao **agente nativo**, que monitora o hospedeiro móvel sempre que o mesmo pede um novo CoA à sua rede local, sendo o mesmo devidamente registrado.
  - Ex: Relação Pais e Filhos

# Localização Nativa

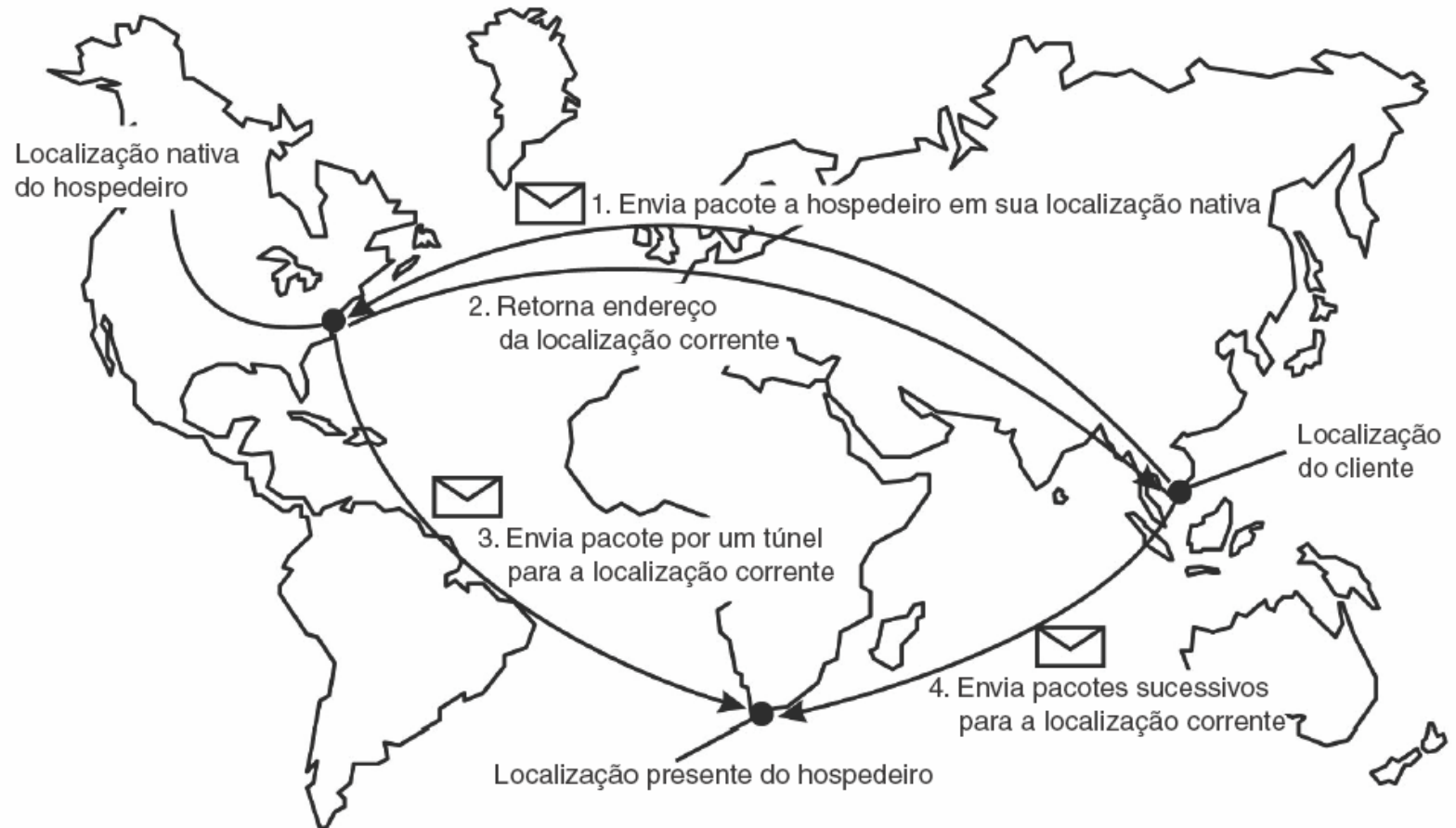
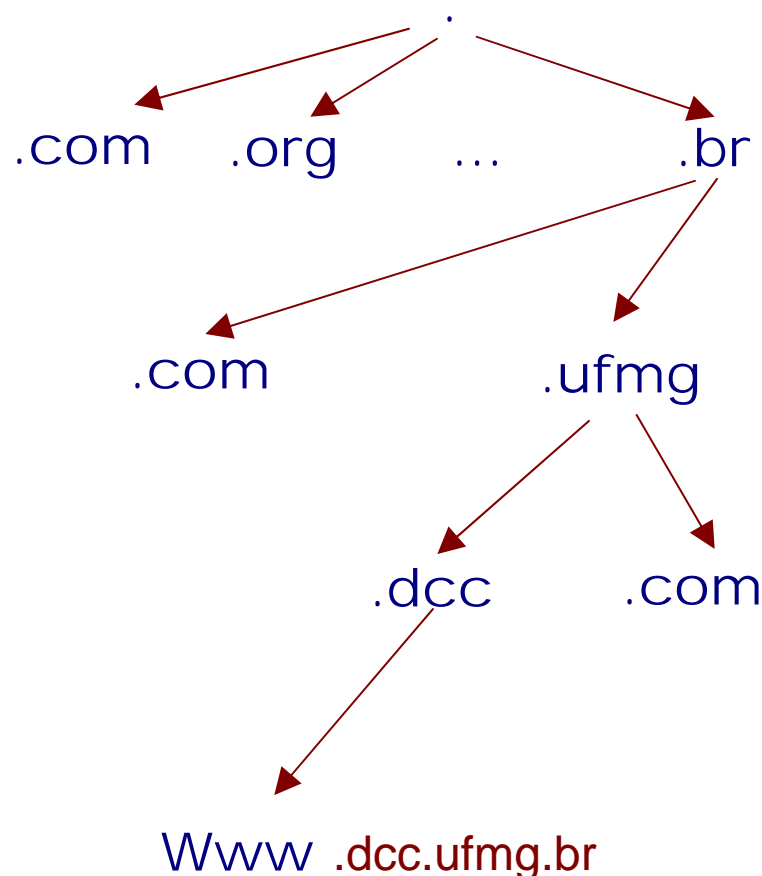


Figura 5.3 Princípio do Mobile IP.

# Sistema Hierárquico

- Nome e Endereço compostos por prefixos
- Características:
  - Escalabilidade;
  - Balanceamento de carga desigual;
  - Pontos críticos de falha: dependência dos elementos próximos à raiz.
- Ex.: DNS.



# Resolução de nomes iterativa

ftp.

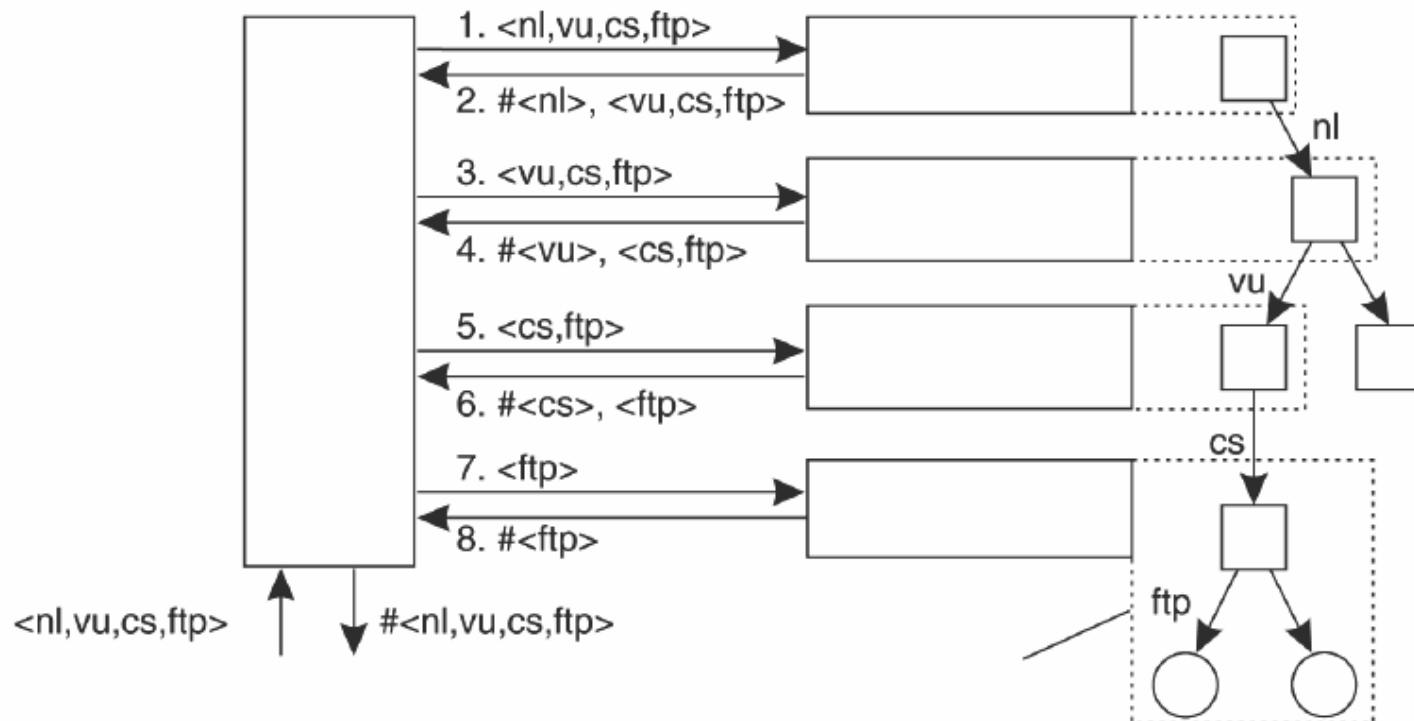


Figura 5.14 Princípio da resolução iterativa de nomes.



# Nomes DNS

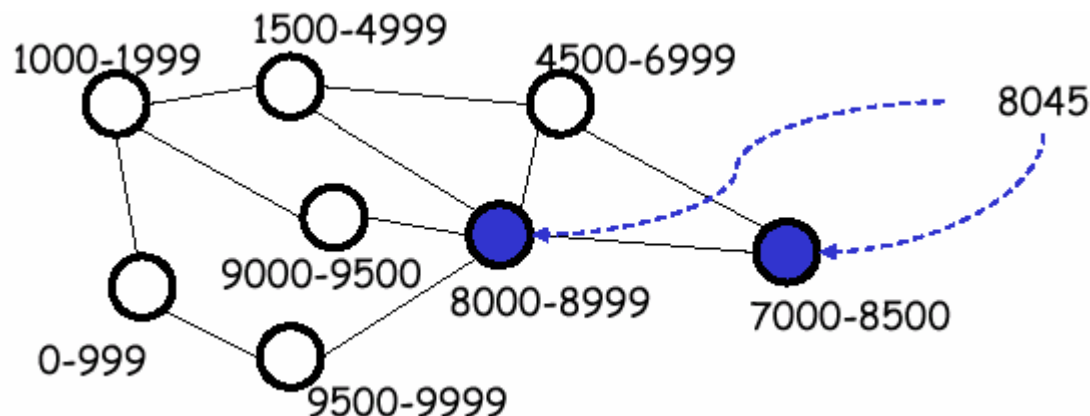
- Nomes hierárquicos
- Gestão de topo efectuada pelo Internet Network Information Center
- Gestão dos subdomínios delegada em autoridades regionais
  - Portugal: Fundação para o Cálculo Científico Nacional (FCCN)
- Gestão dos subdomínios locais delegada nas organizações
- Entradas no DNS associam a cada nome informação sobre o objecto (registos **RR**, Resource Register)

# Modelo DHT

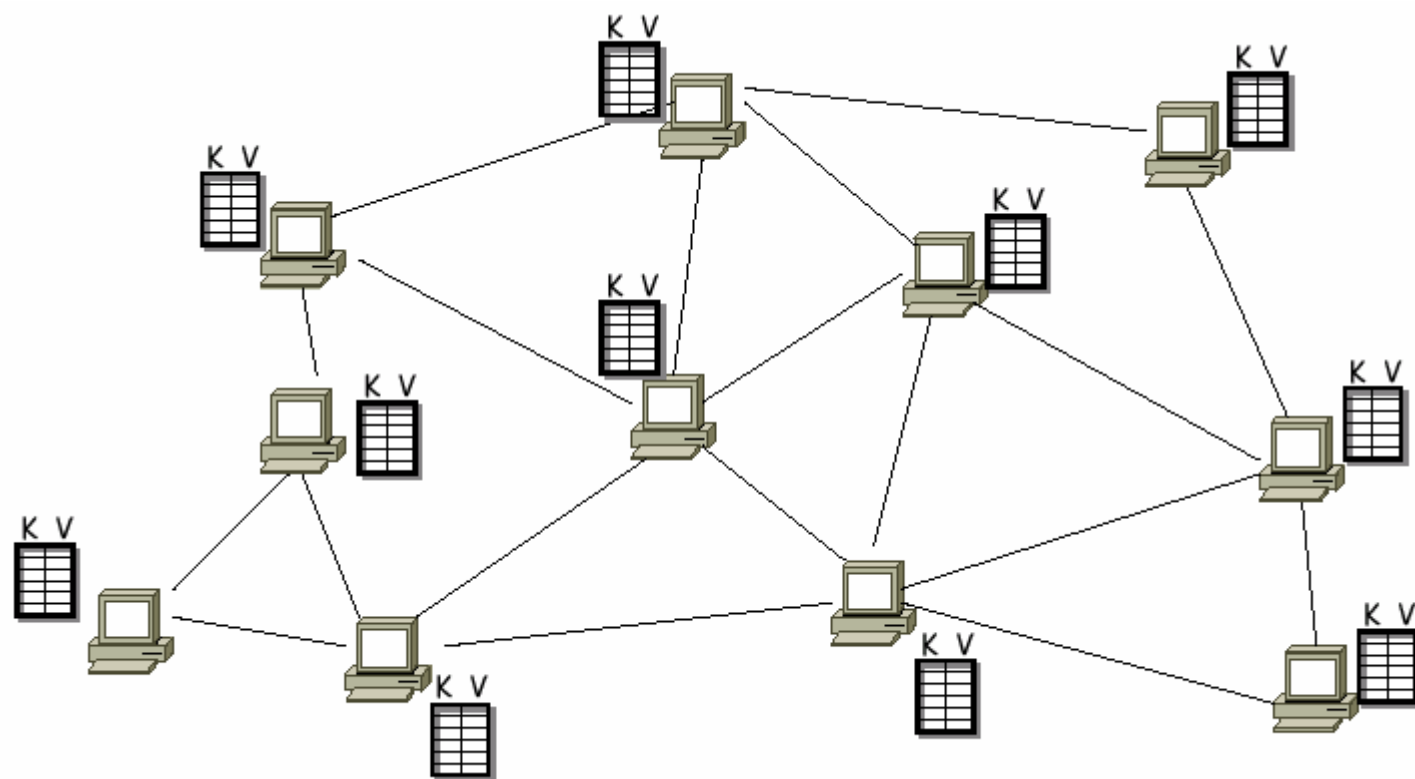
- Hash
  - Estrutura de dados importantes para desenvolvimento
- Hash distribuído na escala da Internet
- Importante para sistemas distribuídos grandes
- Sistemas P2P
  - Chord, CAN, Tapestry, Pastry, ...
- Sistemas muito grandes de gerenciamento de armazenamento de dados
  - OceanStore, CFS, Publius, ...
- Espelhamento de servidores Web

# DHT: Funcionamento

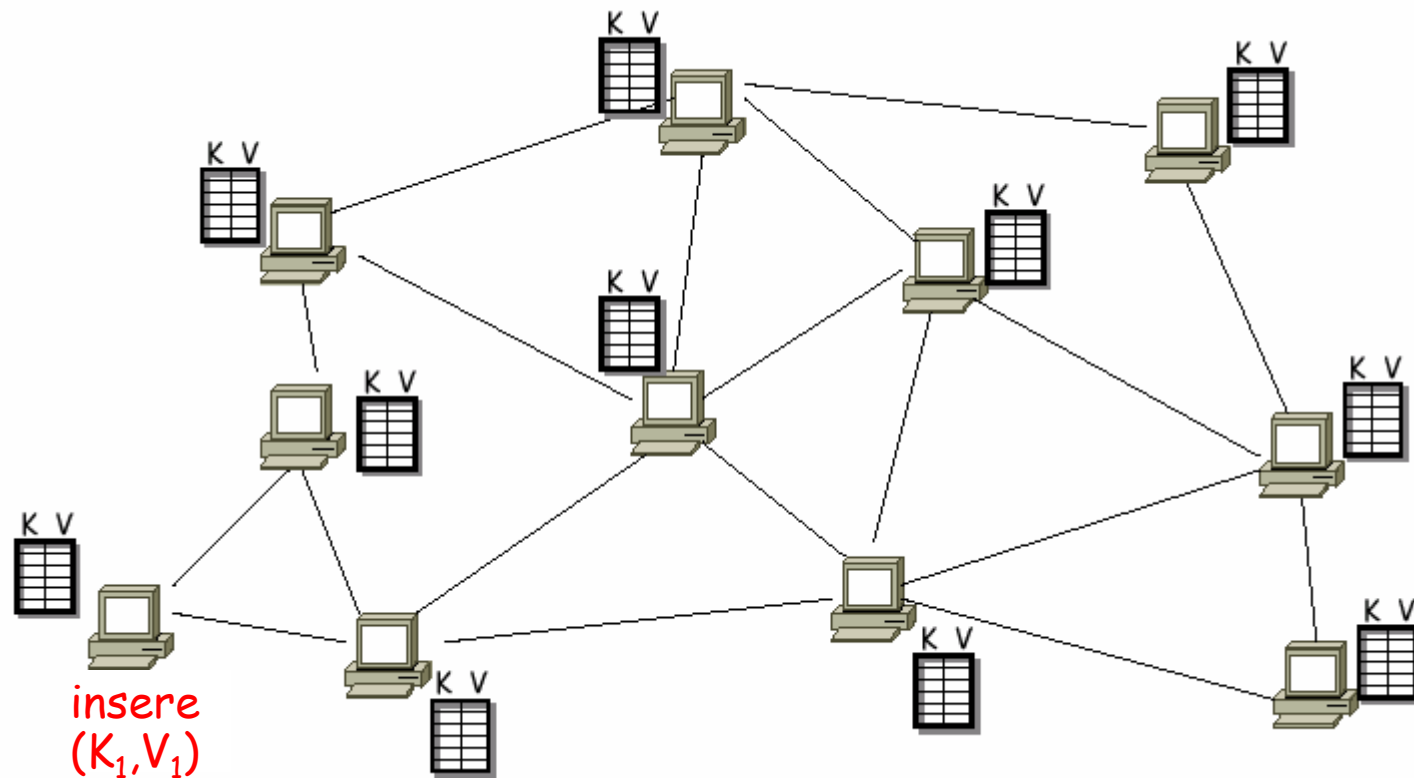
- Função de hash mapeia objeto a identificador único
  - Ex:  $h(\text{"Aquarela do Brasil"}) \rightarrow 8045$
- Faixa de resultados da função de hash é distribuída pela rede



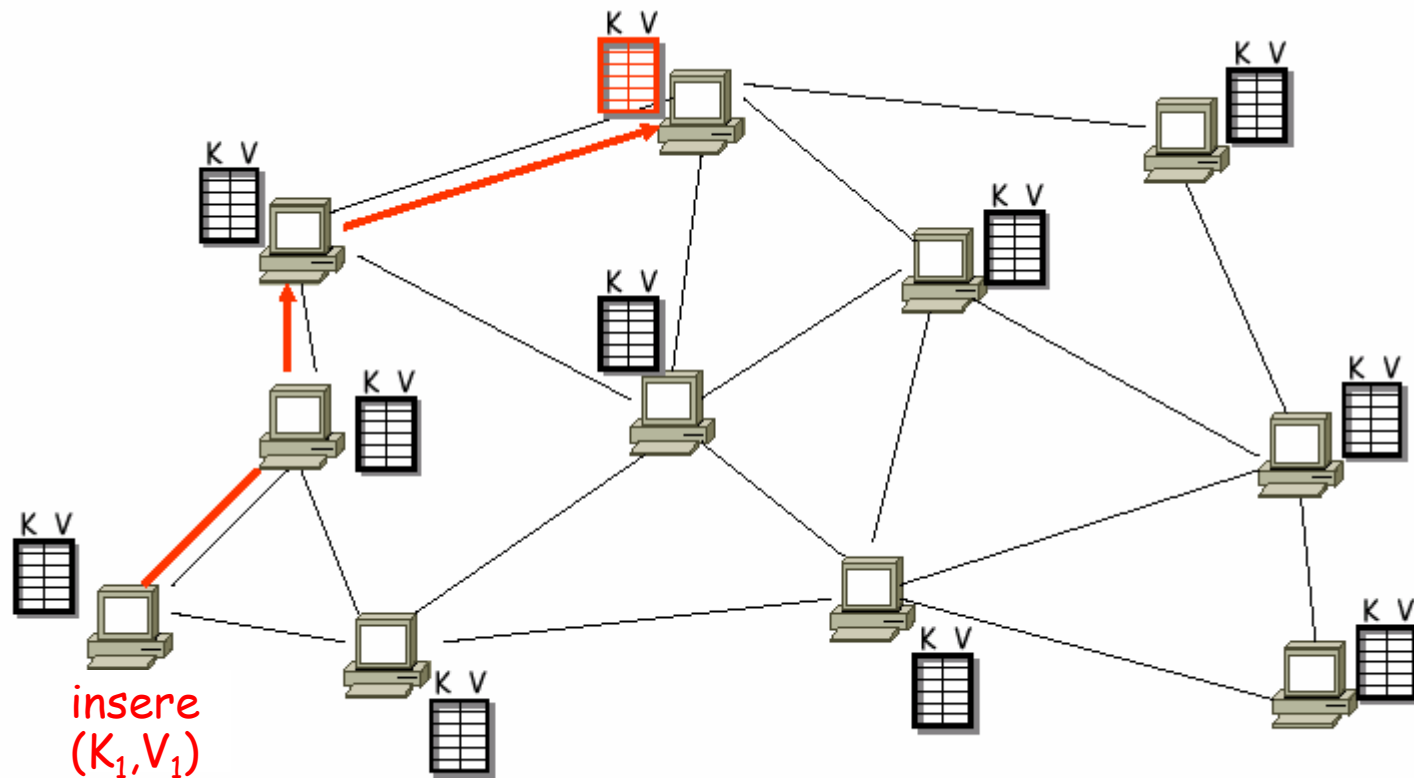
# DHT: Idéia Básica



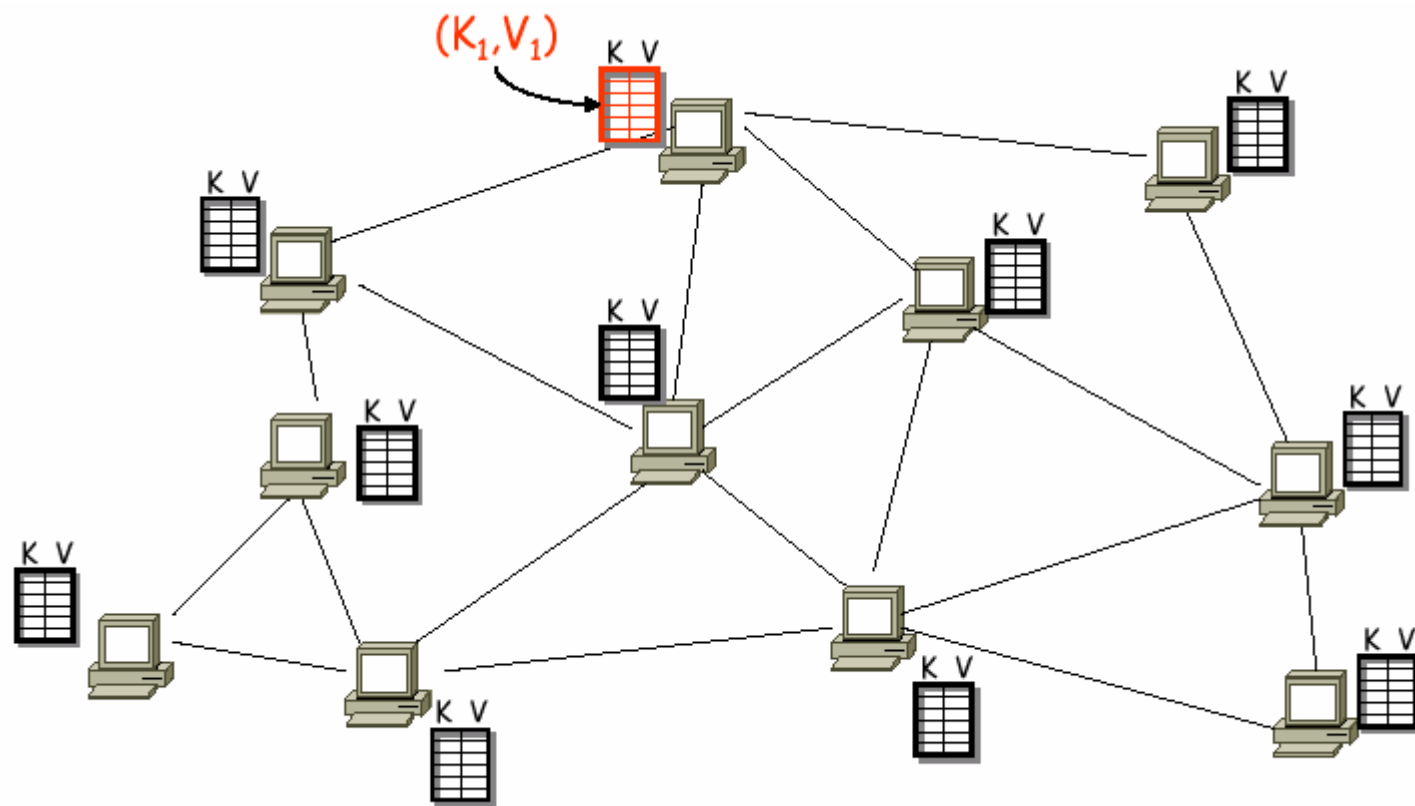
# DHT: Idéia Básica



# DHT: Idéia Básica



# DHT: Idéia Básica





# DHT: Idéia Básica

