

Universidade Estadual de
Mato Grosso dos Sul
Curso de Ciência da
Computação
Disciplina de Algoritmos
Paralelos e Distribuídos

Aula 5 – Nomeação: Nomes, Identificadores,
Endereços, Nomeação Estruturada,
Nomeação Baseada em Atributos.

Nomes, Identificadores e Endereços

- **Entidades:** máquinas, impressoras, discos, processos, usuários, páginas web, janelas gráficas, mensagens, etc.
 - São acessadas por meio de um **ponto de acesso**, ou simplesmente, um **endereço**.
 - Exemplo: Servidor e seu número IP.
- Um endereço pode ser usado como uma maneira de nomear ou identificar uma entidade.
 - Porém existe um problema: O que acontece quando a entidade muda o ponto de acesso?
 - Exemplo: **Mudança de um servidor Web para outra rede**.

Nomes, Identificadores e Endereços

- Pergunta: Como nomear entidades sem utilizar especificamente seu endereço, ou seja, nomeá-las independente da sua posição física (localização)?
- Usar Identificadores ou Nomes Amigáveis a seres humanos!!

Identificadores

- Em muitos casos, são cadeias aleatórias de bits com as seguintes propriedades:
 - Um identificador referencia no máximo **uma entidade**.
 - Cada entidade é referenciada por no máximo **um identificador**.
 - Um identificador sempre referencia a mesma entidade, isto é, nunca é reutilizado.
 - Exemplo: Identificadores de entidades em sistemas P2P baseados no sistema Chord.

Nomes Amigáveis

- Nomes representados por um cadeia de caracteres.
 - *Pathnames*, domínios na internet, números de processos.
 - Exemplo: www.comp.uems.br
- Como esses nomes e identificadores são **resolvidos** para endereços?

Sistemas de Nomeação

- Mantém uma vinculação **nome-endereço**.
- Na forma mais simples:
 - Tabelas de pares (nomes, endereços)
 - Sistemas que abrangem redes de grande porte, uma tabela centralizada não funciona.
- Neste caso, há três classes:
 1. Nomeação simples;
 2. Nomeação estruturada;
 3. Nomeação baseada em atributo

Nomeação Simples

- Aplicada a identificadores
 - Cadeias aleatórias de bits → Nomes simples.
 - Não contém sequer uma informação sobre como localizar o ponto de acesso de uma entidade associada.
- Problema: Dado um identificador, como localizar o ponto de acesso (endereço)?
 - Algumas soluções:
 - Broadcasting
 - Localização nativa
 - Tabelas de Hash Distribuídas (DHT)

Nomeação Simples: Broadcasting e Multicasting

- São consideradas soluções simples.
- Aplicadas somente a redes locais

Nomeação Simples: Broadcasting

- Recursos oferecidos por redes locais nas quais todas as máquinas estão conectadas a um único cabo ou seu equivalente lógico.
- Funcionamento:
 - Mensagem que contém o identificador da entidade é enviada a todas as máquinas da rede.
 - Cada uma das máquinas verifica se tem essa entidade;
 - Máquinas com ponto de acesso para a entidade, enviam uma mensagem que contém o endereço.

Nomeação simples: Broadcasting

- É ineficiente quando a rede cresce, pois:
 - A largura de banda da rede é desperdiçada, com grande número de mensagens de requisição;
 - Aumento da probabilidade de colisões de mensagens, diminuindo a vazão do sistemas;
 - Um grande número de máquinas podem ser interrompidas por requisições que não podem responder.

Nomeação simples: Multicasting

- Somente um número restrito de máquinas recebe a requisição.
- Banco de dados replicado:
 - O endereço multicast é associado a uma entidade replicada.
 - Multicasting é usado para localizar a réplica mais próxima.
 - Na requisição para o endereço multicast cada réplica responde com seu endereço IP
 - Réplica mais próxima → aquela cuja resposta chega antes.

Nomeação simples: Localização Nativa

- Abordagem para suportar entidades móveis em redes de grande escala.
- Monitora a localização corrente de uma entidade.
- Localização nativa costuma ser escolhida como o lugar em que a entidade foi criada.

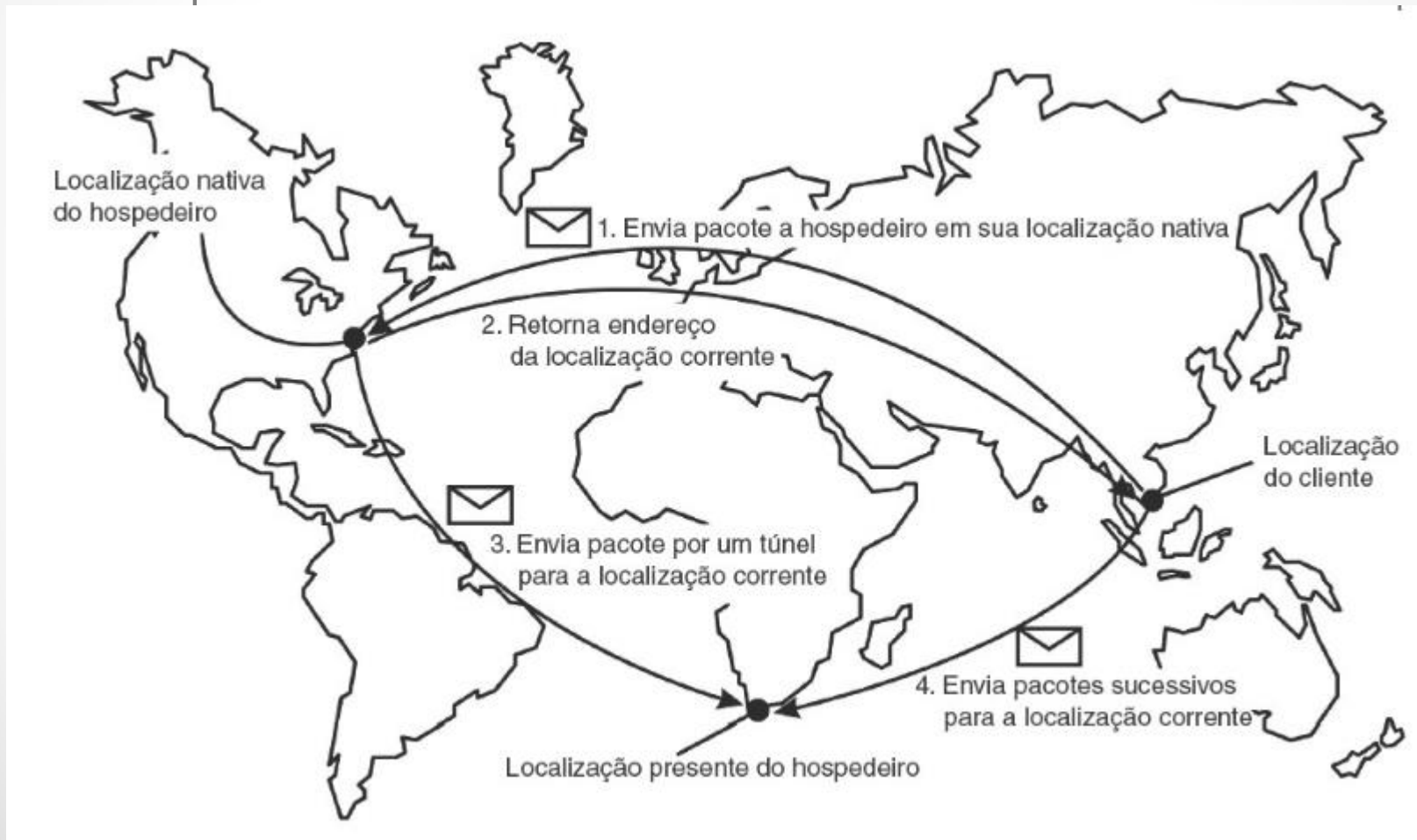
Nomeação simples: Localização Nativa

- **MOBILE IP**

- Cada host móvel usa um endereço fixo.
- Toda a comunicação é dirigida inicialmente ao agente nativo do host móvel (situado na rede local do endereço do host).
- Ao mudar de rede, este host recebe um endereço externo (care-of-address) e registra no agente nativo.
- Quando o agente nativo recebe um pacote para o host móvel
 - Se na rede local → pacote repassado.
 - Senão → túnel até a localização corrente.

Nomeação simples: Localização Nativa

- Princípio do Mobile IP



Nomeação simples: Localização Nativa

- Desvantagens:
 - Para se comunicar com uma entidade móvel, em primeiro lugar um cliente tem que contatar a localização nativa, que pode estar em um lugar completamente diferente.
 - Isto gera latência de comunicação.
 - Assegurar que a localização nativa sempre exista.
 - Entidade decide mudar permanentemente para outra localização → a localização nativa deve mudar também.

Tabelas de Hash Distribuídas (DHT)

- Os nós são organizados logicamente em um anel (**chord**).
 - Usa um espaço de identificadores de m bits para designar **nós** e **entidades** específicas (**arquivos, processos**).
 - Número m de bits é usualmente 128 ou 160.
 - Entidade com chave k cai sob a jurisdição do nó que tenha o menor identificador $id \geq k \rightarrow succ(k)$.
- Como resolver com eficiência uma chave k para o endereço de **succ(k)**?
- Duas abordagens:
 1. Abordagem Linear;
 2. Tabela de Derivação.

Tabelas de Hash Distribuídas - Chord

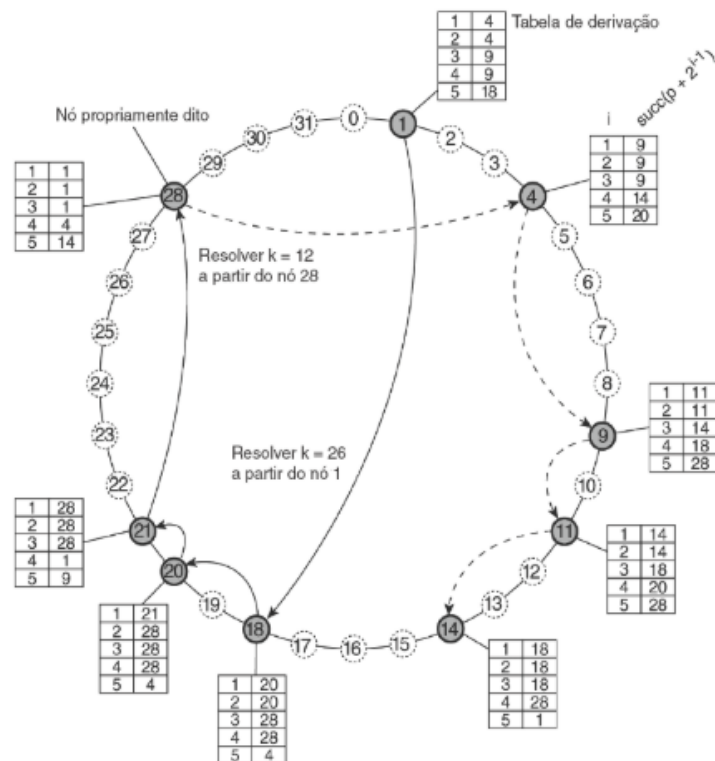
- Abordagem linear
 - Cada nó p monitora o sucessor $\text{succ}(p+1)$ e o predecessor $\text{pred}(p)$
 - Ao receber uma requisição para a chave k , p repassa a requisição para os seus vizinhos, a menos que $\text{pred}(p) < k \leq p \rightarrow p$ retorna o próprio endereço.
 - Não é escalável.

Tabelas de Hash Distribuídas - Chord

- Resolução da chave 26 a partir do nó 1 e da chave 12 a partir do nó 28 em um sistemas chord.

a) Suponhamos que $p = 4$ receba uma requisição para $k = 7 \rightarrow \text{succ}(p+1) \rightarrow$ repassa a requisição ao nó $= 9$

a) Suponhamos que $p = 4$ receba uma requisição para $k = 3 \rightarrow$ como $\text{pred}(4) = 1 < 3 \leq 4 \rightarrow$ retorna o próprio endereço.



Tabelas de Hash Distribuídas - Chord

- Tabela de Derivação (*finger table*)
 - Possui no máximo **m** entradas
 - Denota a tabela de derivação de **p** por **Ft_p**
 - **Ft_p [i] = succ(p + 2ⁱ⁻¹)**
 - a i-ésima entrada aponta para o primeiro nó que sucede **p** por no mínimo **2ⁱ⁻¹**.

Tabelas de Hash Distribuídas - Chord

- Como encontrar um entidade **K**?
 - Referências na tabela de derivação são atalhos para nós existentes no espaço de identificadores.
 - Distância do atalho em relação ao nó **p** aumenta exponencialmente à medida que o índice da tabela de derivação cresce.
 - Para consultar uma chave **K**, o nó **p** repassa a requisição a um nó **q** com índice **j** na tabela de derivação de **p**.
 - $q = Ft_p [j] \leq K \leq Ft_p [j+1]$

Explorando a Proximidade na Rede

- Problema:
 - Organização lógica dos nós em uma rede de sobreposição (*overlay*) pode levar a uma escolha errada no roteamento de mensagens.
 - Tanto **K** quanto **succ(k + 1)** podem estar muito longe fisicamente.

Explorando a Proximidade na Rede

- **Identificar nós com base na topologia:**
 - Ao atribuir um ID a um nó deve-se ter certeza que nós próximos no espaço de endereçamento também estejam próximos fisicamente.
- **Roteamento por Proximidade:**
 - Manter mais de um sucessor e repassar a requisição para o mais próximo.
- **Seleção de Vizinho por Proximidade:**
 - Ao escolher um vizinho (não em chord) pegue o mais próximo.

Nomeação Estruturada

- Nomes simples são bons para máquinas, mas não são convenientes para a utilização de seres humanos.
- Sistemas de nomeação comumente suportam nomes estruturados.
 - Exemplo: Nomeação de arquivos e hosts na internet.

Espaço de Nomes

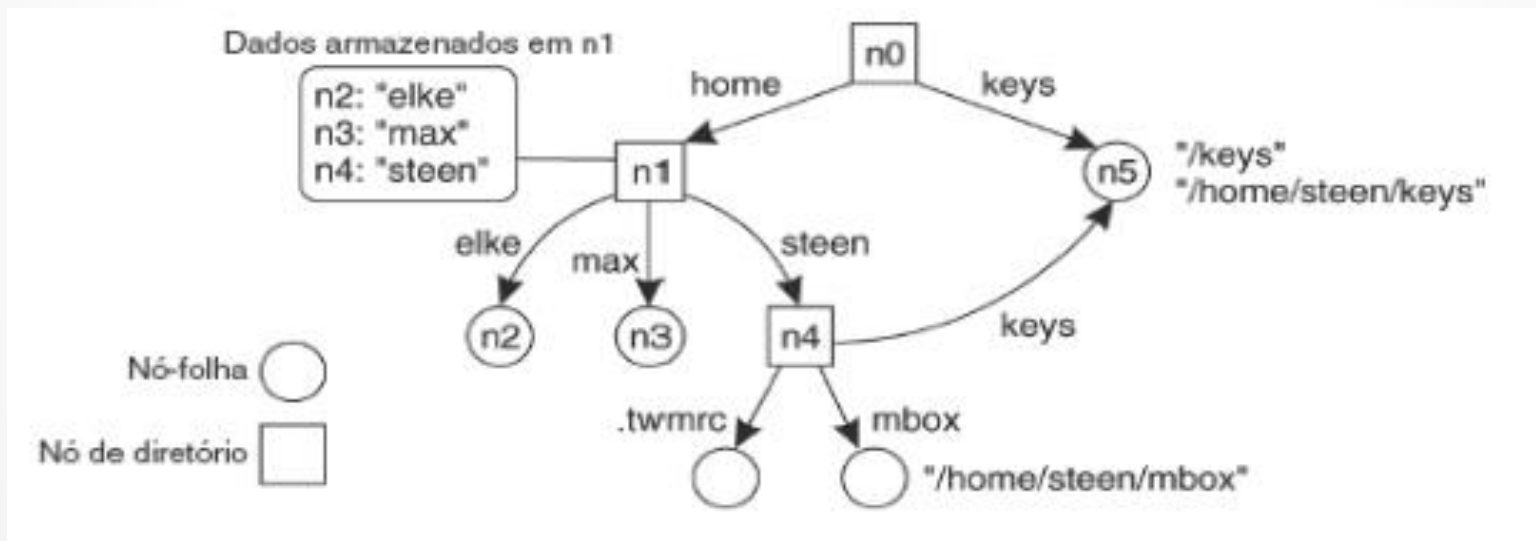
- Nomes são organizados em um espaço de nomes.
- Espaços de nomes podem ser representados como um grafo dirigido, com dois tipos de nós:
 - **Nó-folha**: entidade.
 - **Nó-de-diretório**: entidade que se refere a outros nós.
- O nó-de-diretório possui uma tabela de diretório.
 - **<nome aresta, nome nó>**

Espaço de Nomes

- Sistemas de nomeação possuem, na maioria, um nó raiz.
- Cada caminho no grafo de nomeação pode ser referenciado pela sequência dos labels nas arestas:
 - $N: \langle \text{label 1}, \text{label 2}, \dots, \text{label } n \rangle$
- **Nome de caminho absoluto**: o primeiro nó no caminho é a raiz.
- **Nome de caminho relativo**: o primeiro nó pode ser qualquer nó.

Espaço de Nomes

- Gráfico de nomeação geral com um único nó raiz.



Resolução de Nomes

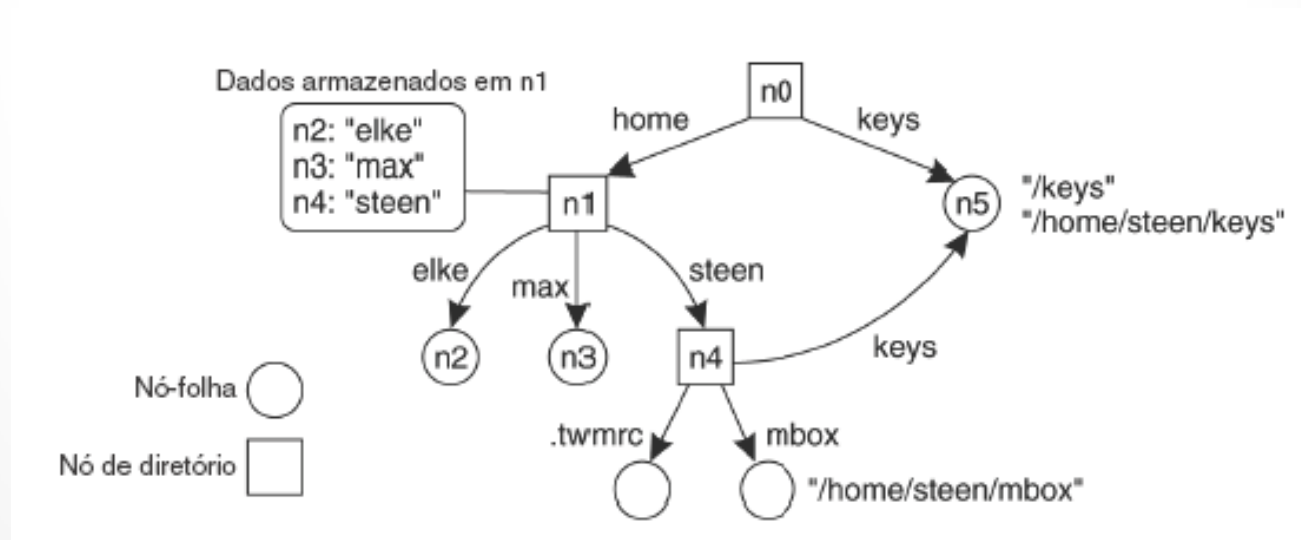
- Espaços de nomes oferecem um mecanismo para armazenar e recuperar informações sobre entidades por meio de nomes.
- Dado um nome de caminho, deve ser possível consultar qualquer informação armazenada no nó referenciado por aquele nome.
- Problema: Para resolver um nome, precisamos de um nó-de-diretório. Como escolher este nó inicial?

Resolução de Nomes

- Mecanismo de fechamento:
 - Trata da seleção do nó inicial em um espaço de nomes a partir do qual a resolução de nomes deve começar.
- São implícitos ao contexto em que a resolução de nomes está se aplicando.
 - www.comp.uems.br → o início da resolução é feita através do servidor de nome DNS.
 - /home/steen/mbox → início da resolução ocorre no servidor local NFS.

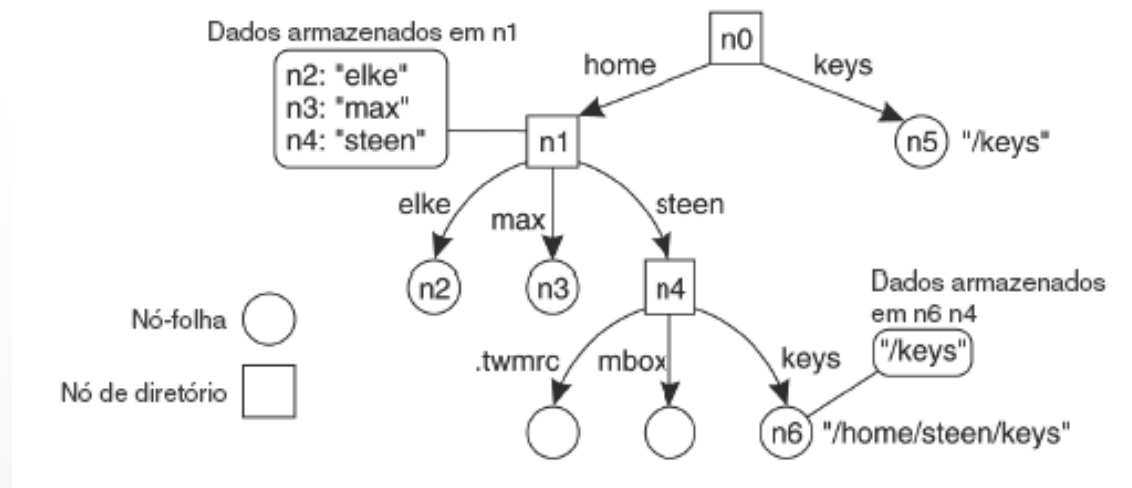
Resolução de Nomes Alias

- Um outro nome para a mesma entidade.
- Vários nomes absolutos para o mesmo nó (hard link)
- Gráfico de nomeação geral com um único nó-raiz.



Resolução de Nomes Alias

- *Symbolic link*: permite a um nó-folha ter uma associação a um caminho absoluto.
- Exemplo: conceito de um ponteiro simbólico explicado em um gráfico de nomeação:



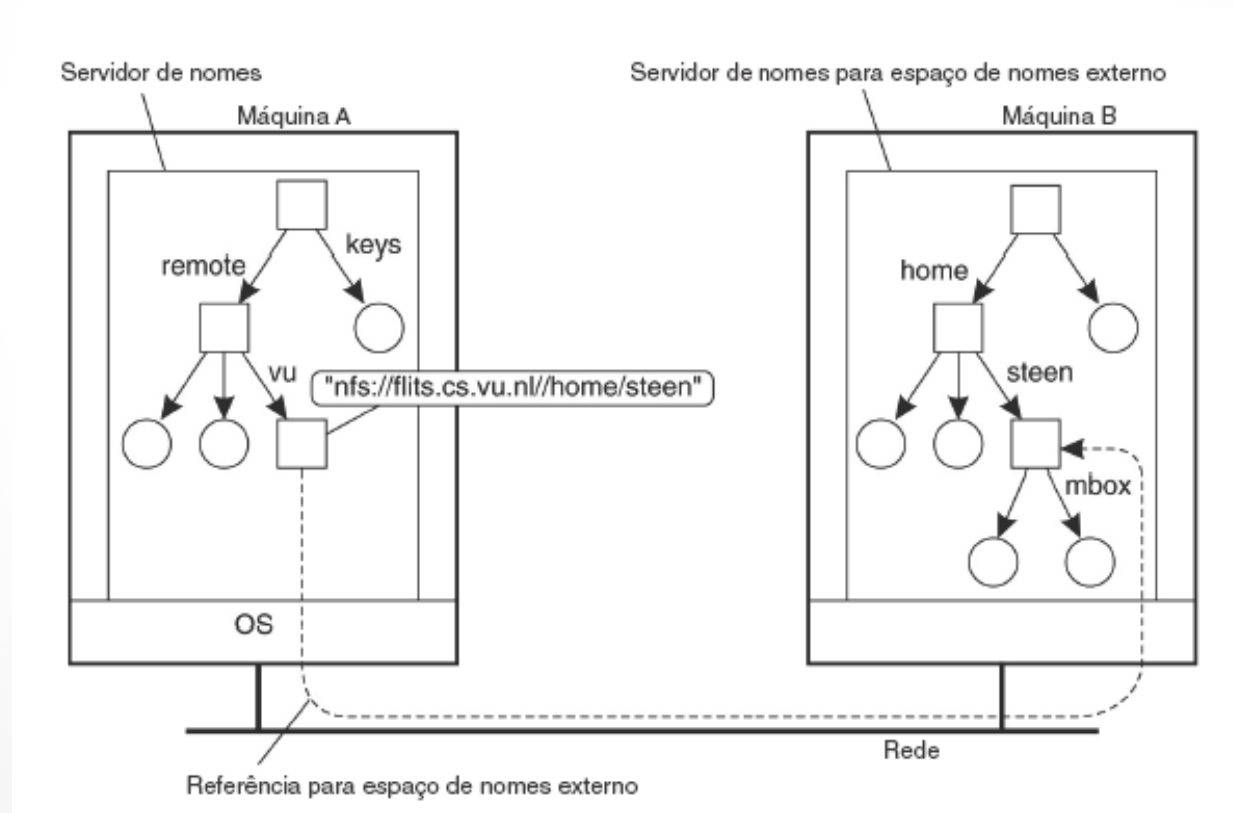
Resolução de Nomes

Diversos Espaços de Nomes

- Como fundir diferentes espaços de nomes de maneira transparente?
- Dados dois espaços de nomes A e B, como A acessa B e como B acessa A?
- **Uma solução: Montagem (Mounting)**
- *Mount point*: É o nó-de-diretório que armazena o identificador de nó do espaço de nome externo.
- *Mounting Point*: É o nó de diretório no espaço de nomes externos.

Diversos Espaços de Nomes - *Mounting*

- Montagem de espaços de nomes remotos por meio de um protocolo de acesso específico.



Implementação de um Espaço de Nomes

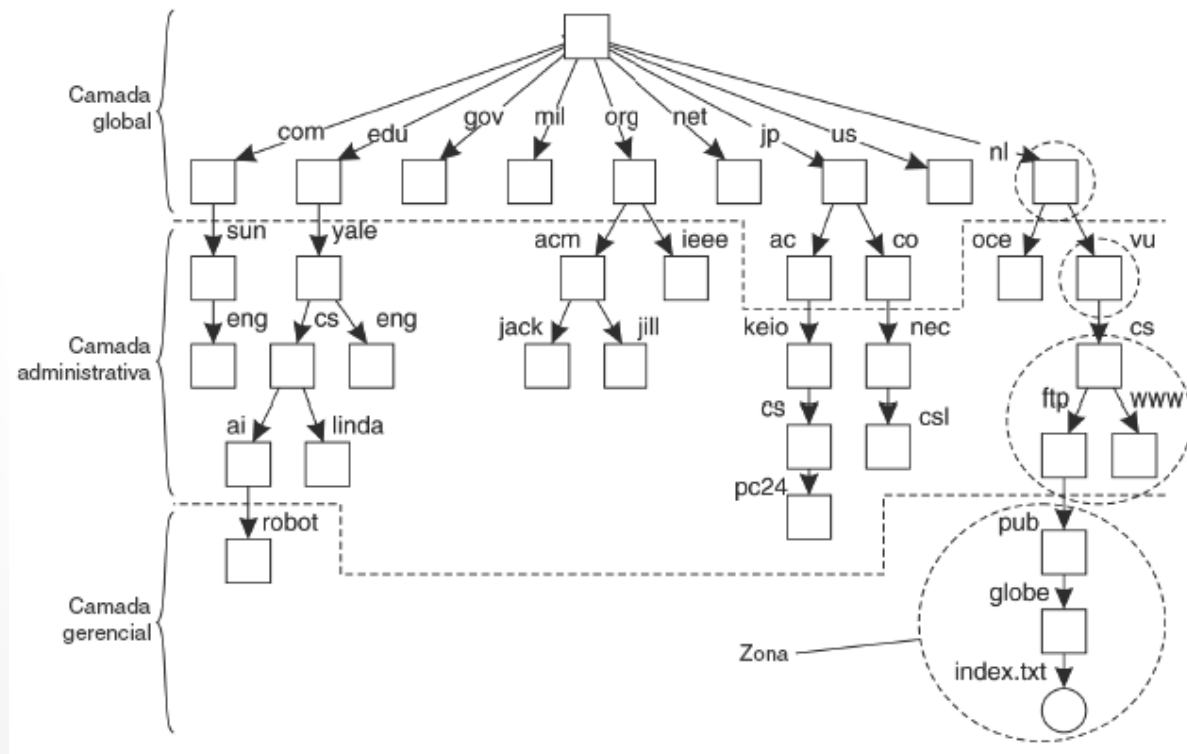
- Serviço que permite a usuários e processos adicionar, remover e consultar nomes.
- **Serviço de nomeação** é implementado por **servidores de nomes**.
- Servidores de nomes devem prover:
 - Escalabilidade
 - Manutenção descentralizada
 - Tolerância a Falhas
 - Robustez
 - Escopo global:
 - Os nomes possuem o mesmo significado em todos os lugares.

Implementação de um Espaço de Nomes

- Costumam ser organizados em hierarquia.
- Geralmente o espaço de nomes são divididos em três camadas:
 1. Camada Global
 - Raiz e seus filhos
 - Principal característica: Estabilidade
 - Podem representar organizações
 2. Camada Administrativa
 - Nós de diretórios
 - Gerenciados por uma única organização
 - Relativamente Estáveis
 3. Camada Gerencial
 - Nós cujo comportamento típico é a mudança periódica
 - Mantidos por administradores de sistemas e usuários finais

Implementação de um Espaço de Nomes

- Exemplo de repartição do espaço de nomes DNS, incluindo arquivos acessíveis pela internet, em três camadas.



Comparação entre Servidores de Nomes

- Comparação entre servidores de nomes para implementar nós de espaço de nomes de grande escala repartidos em uma camada global, uma camada administrativa e uma camada gerencial.

Item	Global	Administrativa	Gerencial
Escala geográfica da rede	Mundial	Organização	Departamento
Número total de nós	Poucos	Muitos	Grandes quantidades
Capacidade de resposta a consultas	Segundos	Milissegundos	Imediata
Propagação de atualizações	Lerda	Imediata	Imediata
Quantidade de réplicas	Muitas	Nenhuma ou poucas	Nenhuma
É aplicada cache do lado do cliente?	Sim	Sim	Às vezes

Como resolver nomes?

- **Resolução iterativa**

- Servidor responde somente o que sabe: o nome do próximo servidor que deve ser buscado.
- Cliente procura iterativamente os outros servidores.

- **Resolução Recursiva**

- Servidor passa o resultado para o próximo servidor que encontrar.
- Para o cliente, somente existe uma mensagem de retorno:
 - O endereço do nome ou “não encontrado”.

Como resolver nomes?

- Exemplos:

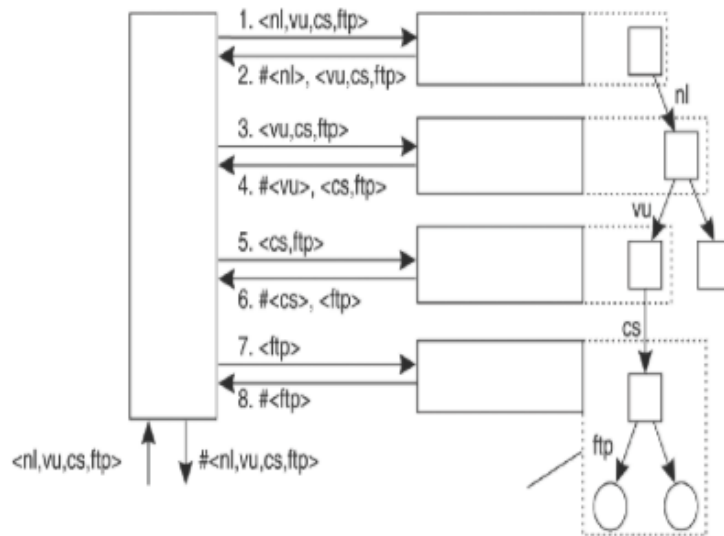


Figura 5.14 Princípio da resolução iterativa de nomes.

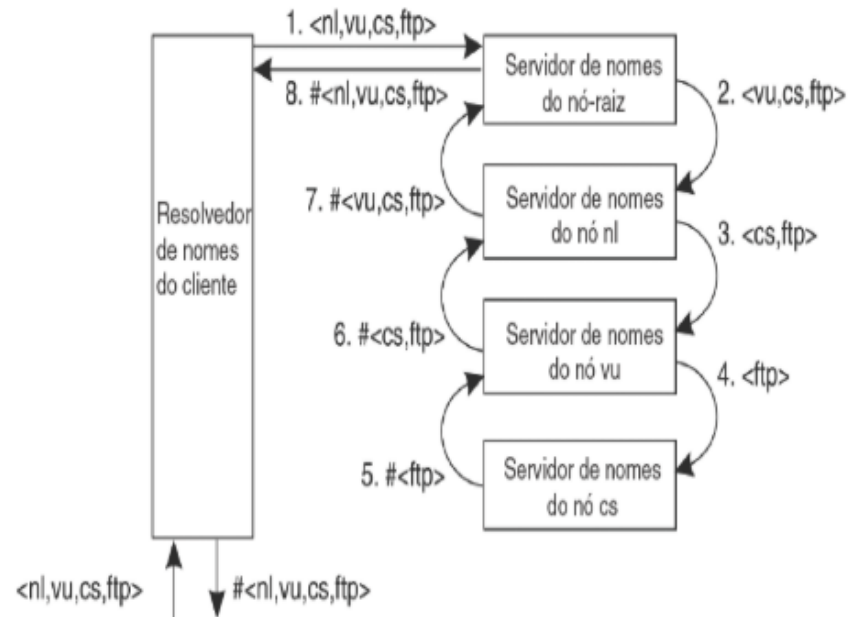


Figura 5.15 Princípio da resolução recursiva de nomes.

Resolução Iterativa

Versus

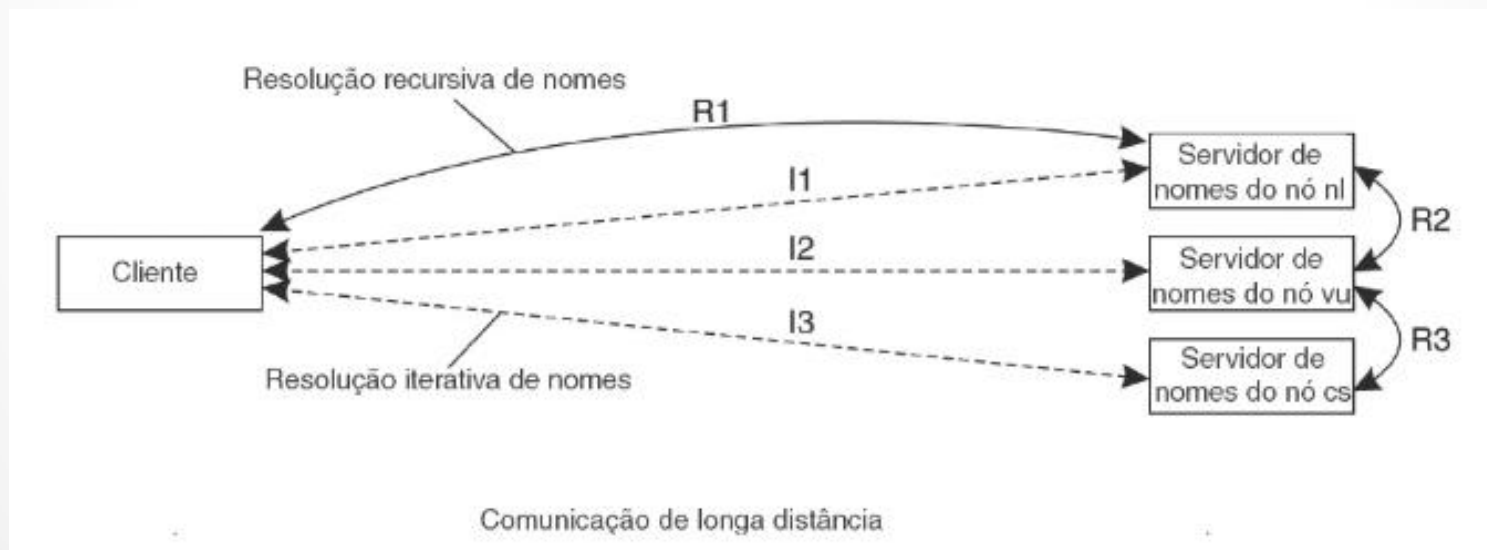
Resolução Recursiva

- Quadro comparativo:

Tipo	Carga Camada Global	Cache	Comunicação
Iterativa	Baixa	Menos Eficiente	Custosa
Recursiva	Alta	Mais Eficiente	Menos custosa

Resolução Iterativa *Versus* Resolução Recursiva

- Comparação entre resolução iterativa e resolução recursiva de nomes no que diz respeito aos custos de comunicação.



DNS: *Domain Name System*

Pessoas: muitos identificadores:

- CPF, nome, no. da Identidade

hospedeiros, roteadores Internet :

- endereço IP (32 bit) - usado p/ endereçar datagramas
- “nome”, ex., jambo.ic.uff.br - usado por gente

P: como mapear entre nome e endereço IP?

Domain Name System:

- *base de dados distribuída* implementada na hierarquia de muitos *servidores de nomes*
- *protocolo de camada de aplicação* permite que hospedeiros, roteadores, servidores de nomes se comuniquem para *resolver* nomes (tradução endereço/nome)
 - nota: função imprescindível da Internet implementada como protocolo de camada de aplicação
 - complexidade na borda da rede

DNS: *Domains Name System*

Serviços DNS

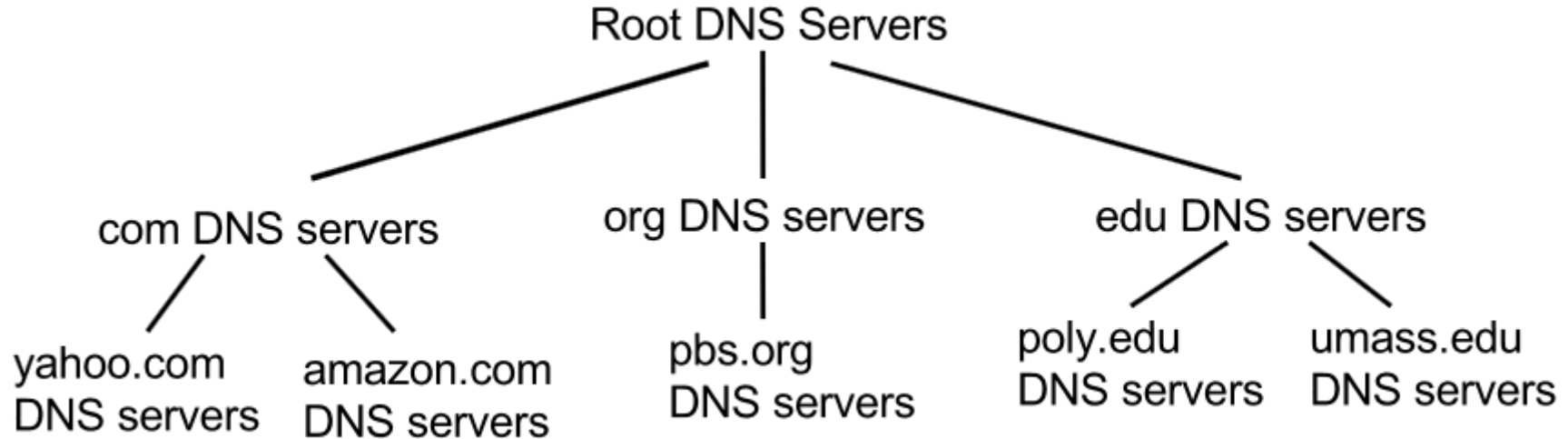
- Tradução de nome de hospedeiro para IP
- Apelidos para hospedeiros (aliasing)
 - Nomes canônicos e apelidos
- Apelidos para servidores de e-mail
- Distribuição de carga
 - Servidores Web replicados: conjunto de endereços IP para um nome canônico

Por que não centralizar o DNS?

- ponto único de falha
- volume de tráfego
- base de dados centralizada e distante
- manutenção (da BD)

Não é escalável!

Base de Dados



Cliente quer IP para www.amazon.com; 1ª aprox:

- Cliente consulta um servidor raiz para encontrar um servidor DNS .com
- Cliente consulta servidor DNS .com para obter o servidor DNS para o domínio amazon.com
- Cliente consulta servidor DNS do domínio amazon.com para obter endereço IP de www.amazon.com

DNS: Servidor Raiz

- Procurado pelo servidor local que não consegue resolver o nome
- Servidor raiz:
 - Procura servidor oficial se mapeamento desconhecido
 - Obtém tradução
 - Devolve mapeamento ao servidor local



Servidores TLD e Oficiais

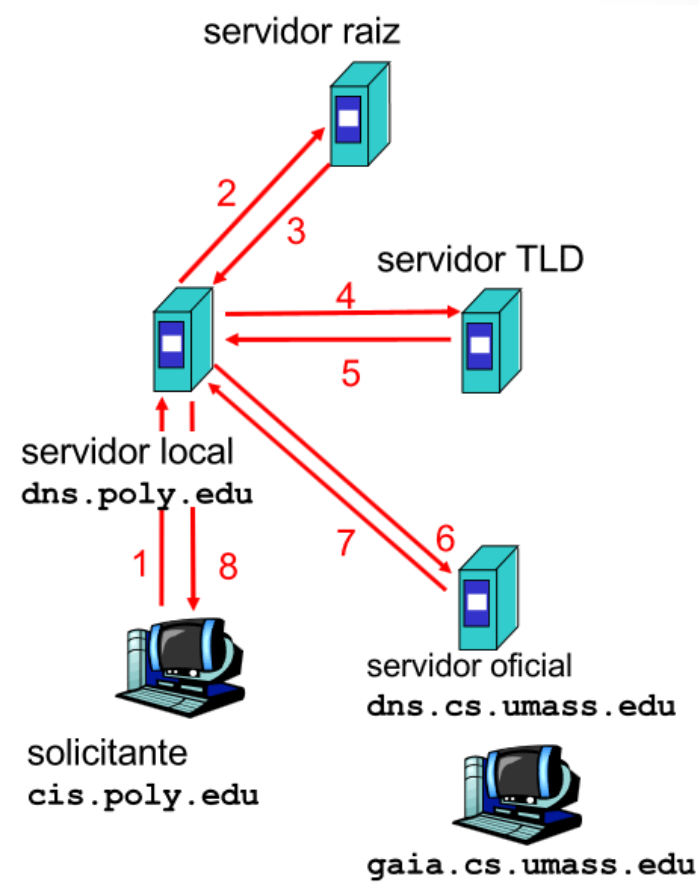
- **Servidores *Top Level Domain* (TLD)**
 - São servidores DNS responsáveis por domínios .com, .org, .net, .edu e todos os domínios responsáveis por países com br, fr, jp, uk, ca.
 - *Network Solutions* mantém servidores para domínio .com
 - NIC.br (Registro .br) para domínios .br
- **Servidores Oficiais**
 - Servidores DNS das organizações, provendo mapeamentos oficiais entre nomes de hospedeiros e endereços IP para os servidores da organização (Exemplo: Web e Corrêio).
 - Podem ser mantidos pelas organizações ou pelo provedor de acesso.

Servidor de Nomes Local

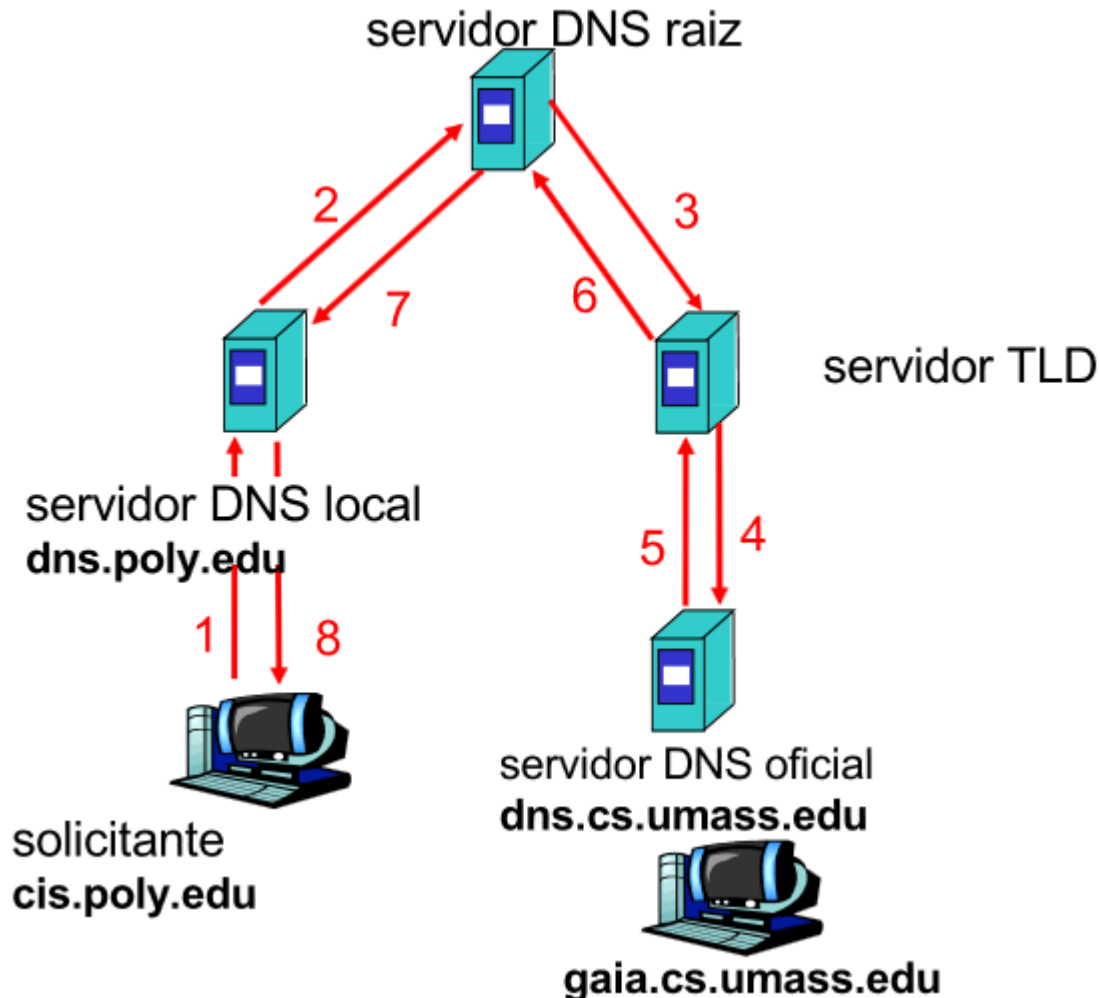
- Não pertence necessariamente à hierarquia.
- Cada ISP (residencial, universidade, companhia) possui um servidor de nome.
 - *Servidor de nomes default*
- Quando um hospedeiro faz uma consulta DNS, a mesma é enviada para o seu servidor DNS local.
 - Atua como um intermediário, enviando consultas para a hierarquia.

Exemplo de Resolução de Nomes pelo DNS

- Hospedeiro em cis.poly.edu quer endereço IP para gaia.cs.umass.edu



Exemplo de Resolução de Nome pelo DNS



DNS: uso de cache e atualização de dados

- Uma vez que um servidor qualquer aprende um mapeamento, ele o coloca em uma **cache** local.
 - Entradas na cache são sujeitas a temporização – desaparecem depois de um certo tempo.
 - Servidores TLD tipicamente armazenados no cache de servidores de nomes locais.
 - Servidores raiz não costumam ser visitados com muita frequência.

Nomeação Baseada em Atributo

- Fornece uma descrição da entidade que está sendo procurada.
- Descreve uma entidade em termos de pares `<atributo, valor>`
- Serviços de diretório retornam uma ou mais entidades que atendam a descrição do usuário.

Leitura, Estudo e Exercícios

- Este material tem sua fonte no livro:
 - Sistemas Distribuídos: Princípios e Paradigmas.
 - Andrew Tanenbaum e Marteen Steen
 - Capítulo 5

- Distributed System – Concepts and Design
- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair.
- Capítulos 13.