

30-769

Sistemas Distribuídos

MSc. Fernando Schubert

NOMEAÇÃO EM SISTEMAS DISTRIBUÍDOS

- Nomes desempenham papel importante em todos os sistemas de computação.
- São usados, por exemplo, para compartilhar recursos e identificar entidades
- Para resolver nomes é necessário implementar um Sistema de Nomes
- Em um sistema distribuído, a implementação do sistema de nomes costuma ser distribuída por várias máquinas
- A forma como a distribuição é feita influencia diretamente na escalabilidade e eficiência do sistema

NOMEAÇÃO EM SISTEMAS DISTRIBUÍDOS

- Modos diferentes e importantes pelos quais nomes são usados em SD
 - Implementação de nomes amigáveis para seres humanos
 - Nomes são usados para localizar entidades de modo independente de sua localização no instante em questão
 - Entidades podem ser descritas por meio de suas características
 - Mostra formas para resolver um nome de acordo com uma descrição de atributos

NOMES, IDENTIFICADORES E ENDEREÇOS

- Nome -> cadeia de bits ou caracteres usada para referenciar uma entidade.
- Entidade pode ser praticamente qualquer coisa: hosts, impressoras, discos, arquivos, etc
- Entidades são acessadas através de um ponto de acesso
 - O nome de um ponto de acesso é denominado endereço
 - Exemplo: servidor e seu endereço IP
 - Uma entidade pode fornecer mais de um ponto de acesso
 - Um ponto de acesso pode mudar ao longo do tempo

NOMES, IDENTIFICADORES E ENDEREÇOS

- O endereço pode ser usado como uma maneira de nomear, identificar uma entidade
- Problema
 - A entidade pode mudar facilmente de ponto de acesso
 - Nomes de endereço não são amigáveis para seres humanos
 - Como nomear entidades, sem utilizar especificamente seu endereço, ou seja, nomeá-las independentemente da sua posição física (localização)?

NOMES, IDENTIFICADORES E ENDEREÇOS

- Você considera que as URLs abaixo são independentes de localização?
http://www.acme.org/index.html <http://www.acme.nl/index.html>
- Um nome independente de localização significa que o nome da entidade é independente do seu endereço.
 - Apenas considerando o nome, nada pode ser dito sobre o endereço da entidade associada.
 - Logo, os dois exemplos são independentes de localização, embora o primeiro dê menos dicas de onde se encontra a localização da entidade

NOMES, IDENTIFICADORES E ENDEREÇOS

- Necessário definir um nome que seja independente dos endereços da entidade -> endereço independente de localização
 - Caso o sistema distribuído seja reorganizado e servidores mudem de endereço é importante que seja possível continuar acessando o serviço
 - Necessário definir um nome construído para ser usado por seres humanos, normalmente formados por cadeias de caracteres nome amigável a seres humanos

NOMES, IDENTIFICADORES E ENDEREÇOS

- Identificador -> um nome que identifica uma entidade e tem 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
 - Uso de identificadores permite que entidades sejam referenciadas sem ambiguidade

NOMES, IDENTIFICADORES E ENDEREÇOS

- Dê alguns exemplos de identificadores verdadeiros
 - ISBN de livros
 - Números de identificação de produtos de software e hardware
 - Número de um funcionário dentro de uma organização
 - Ethernet Addresses
 - CPF

NOMES, IDENTIFICADORES E ENDEREÇOS

- Um sistema de nomeação mantém uma vinculação nome – endereço que em sua forma mais simples é apenas uma tabela de pares (nome, endereço)
- Em sistemas distribuídos de grande escala é necessário usar tabelas descentralizadas
- Um nome é decomposto em várias partes e a resolução é feita por meio de consultas recursivas das partes

ftp.cs.vu.nl

NS(.) → NS(nl) → NS(vu.nl) → endereço de ftp.cs.vu.nl

NOMEAÇÃO SIMPLES

- Identificadores são, em muitos casos, cadeias aleatórias de bits (nomes não estruturados ou simples);
- Um identificador não contém informação de como localizar uma entidade
 - Como nomes simples podem ser resolvidos?
- Soluções simples aplicáveis somente em redes locais:
 - Broadcasting e multicasting
 - Ponteiros repassadores
 - Possuem problemas de escalabilidade!

NOMEAÇÃO SIMPLES

- Broadcasting
 - Recursos oferecidos a redes locais nas quais todas as máquinas estão conectadas a um único cabo ou seu equivalente lógico
 - Como funciona:
 - Uma mensagem que contém o identificador da entidade é enviada a cada máquina da rede
 - Cada uma delas deve verificar se tem essa entidade
 - Máquinas com ponto de acesso para a entidade, enviam uma mensagem que contém o endereço

NOMEAÇÃO SIMPLES

- Broadcasting
 - Se torna ineficiente quando a rede cresce
 - Largura de banda da rede é desperdiçada, com grande número de mensagens de requisição
 - Aumento da probabilidade de colisões de mensagens, diminuindo o throughput do sistema
 - Grande número de máquinas pode ser interrompido por requisições que não podem responder

NOMEAÇÃO SIMPLES

- Multicasting
 - Grupos restritos de hosts são identificados por um endereço multicast
 - Quando um host envia uma mensagem a um endereço multicast, a camada de rede fornece um serviço para entregar aquela requisição a todos os membros do grupo
 - Envia apenas para grupos específicos, limitando o tráfego
 - Também usado para associar réplicas e localizar a réplica mais próxima
 - A réplica mais próxima é aquela cuja a resposta chega primeiro

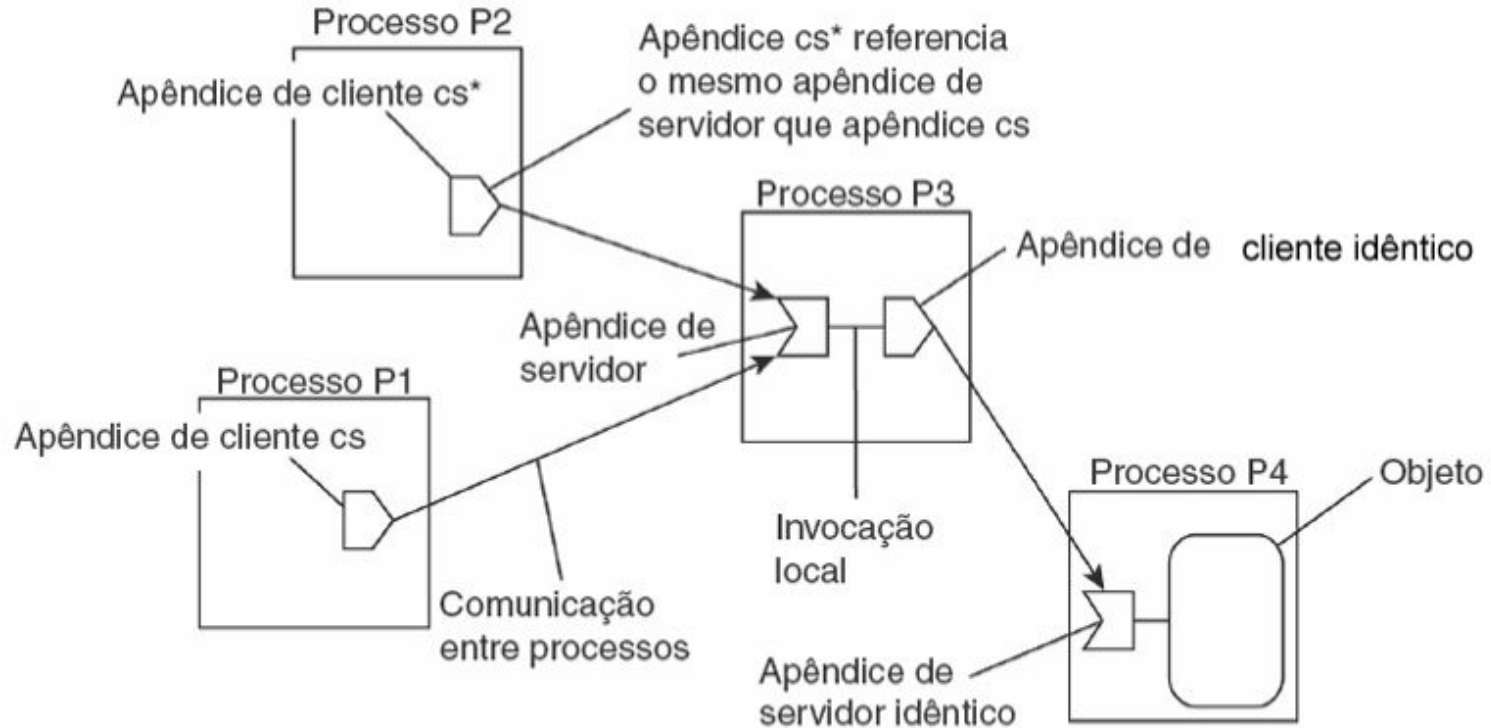
NOMEAÇÃO SIMPLES

- Ponteiros Repassadores
 - Usado para localização de entidades móveis
 - O princípio é bem simples:
 - Quando uma entidade se move de A para B, deixa em A uma referência à sua nova localização, em B.
 - Tão logo uma entidade é localizada, o cliente pode consultar o endereço corrente da entidade percorrendo uma cadeia de ponteiros repassadores

NOMEAÇÃO SIMPLES

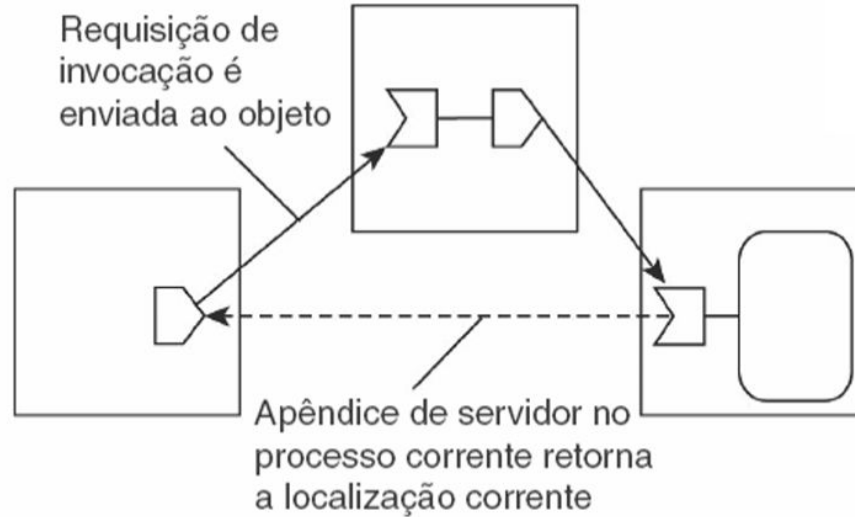
- Ponteiros Repassadores
 - Se não houver precauções, a cadeia para uma entidade de alta mobilidade torna o custo de localização proibitivo
 - Todas localizações intermediárias terão de manter sua parte da cadeia de ponteiros repassadores pelo tempo necessário
 - Enlaces rompidos geram vulnerabilidade
 - Se um ponteiro repassador for perdido, a entidade não poderá ser alcançada
 - É importante manter cadeias curtas e repassadores robustos

NOMEAÇÃO SIMPLES



NOMEAÇÃO SIMPLES

- Ponteiros Repassadores - redirecionamento



NOMEAÇÃO SIMPLES – LOCALIZAÇÃO NATIVA

- Quando uma entidade móvel se torna inalcançável por algum motivo uma solução é a localização nativa
- Localização Nativa
 - A máquina em que o objeto foi criado mantém sempre uma referência da sua localização corrente
 - Esta referência é mantida de modo tolerante a falha
 - Quando uma cadeia é rompida, pergunta-se à localização nativa do objeto qual a sua localização corrente

NOMEAÇÃO SIMPLES – 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
- Exemplo
 - Funcionamento do Mobile IP (mIPv6)
 - Cada host móvel usa um endereço IP fixo localização nativa

NOMEAÇÃO SIMPLES – LOCALIZAÇÃO NATIVA

- Mobile IP
 - Toda a comunicação é dirigida inicialmente ao agente nativo do host móvel (situado na rede local do endereço do host)
 - Ao mudar de rede, o host recebe um endereço externo e registra no agente nativo
 - Quando o agente nativo recebe um pacote para o host móvel ele consulta a localização do host
 - Se na rede local → pacote repassado
 - Senão → túnel até a localização corrente

NOMEAÇÃO SIMPLES – LOCALIZAÇÃO NATIVA



Figura 5.3 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 → Latência de comunicação
- Assegurar que a localização nativa sempre exista
- Entidade decide mudar permanentemente para outra localização → localização nativa deve também mudar

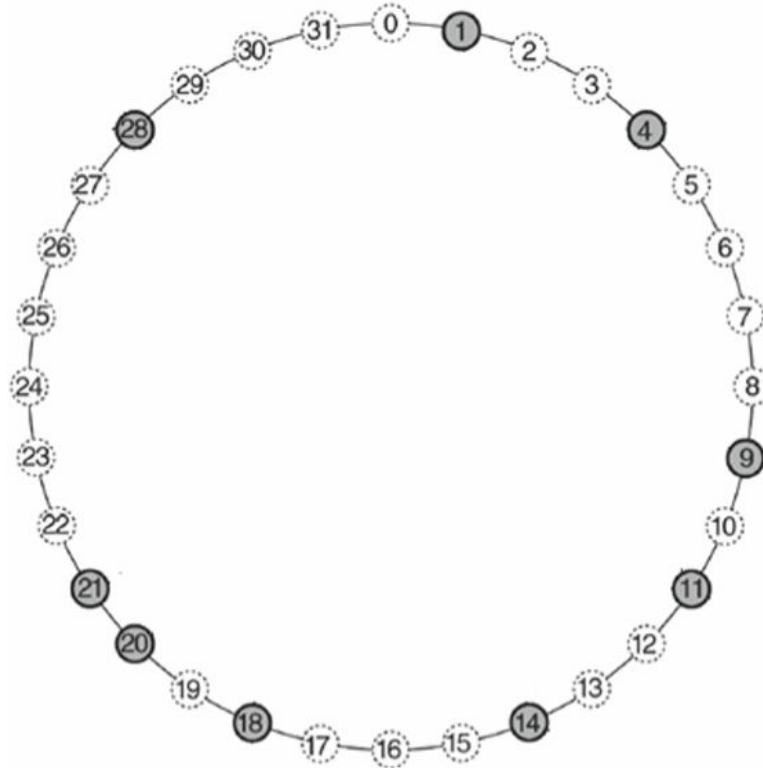
NOMEAÇÃO SIMPLES – DHT

- Tabelas de Hash Distribuídas (DHT)
 - Será considerado o sistema Chord nós são organizados logicamente em um anel
 - Usa um espaço de endereçamento de m bits para chaves para designar nós e entidades específicas (arquivos, processos)
 - Uma entidade com chave k cai sob a jurisdição do nó que tenha o menor identificador $id \geq k$, nó denominado sucessor de k $\text{succ}(k)$

NOMEAÇÃO SIMPLES – DHT

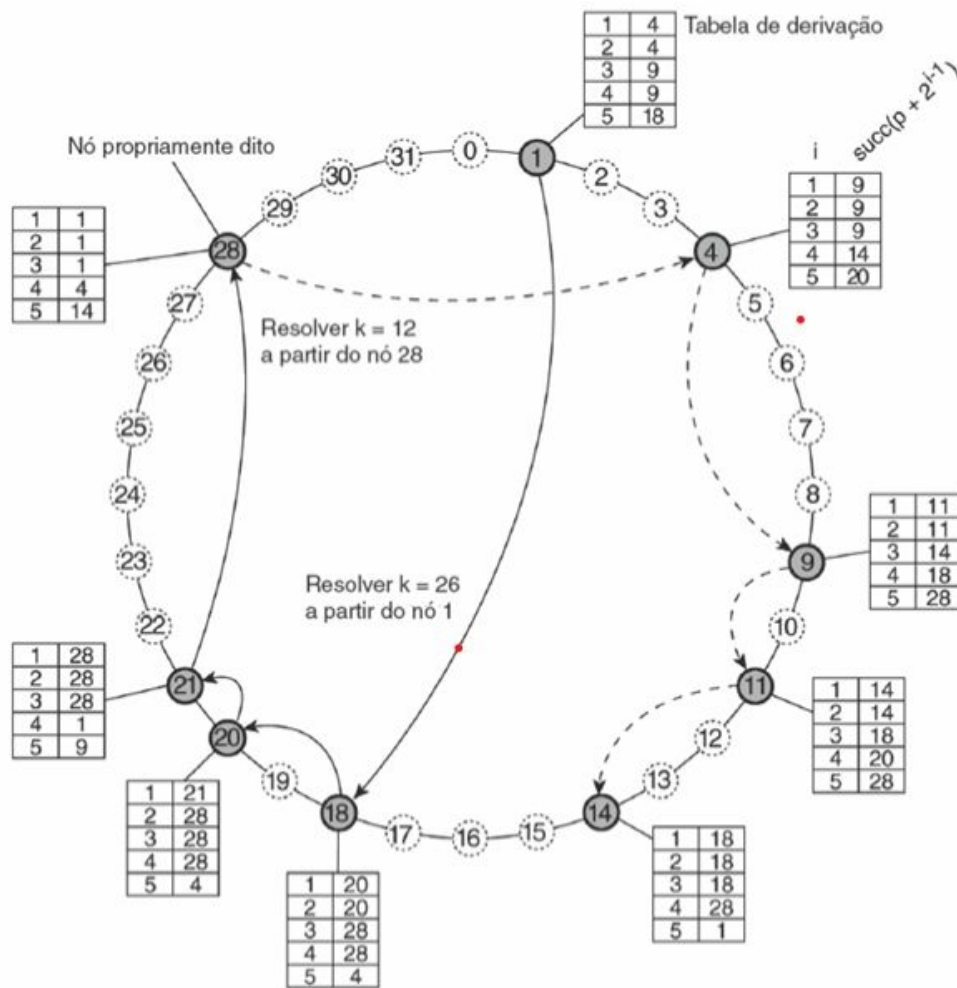
- Como resolver com eficiência uma chave k para o endereço de $\text{succ}(k)$?
 - Uma abordagem simples porém não escalável abordagem linear
 - Permite que cada nó p monitore o sucessor e o seu predecessor
 - Quando p receber uma requisição para a chave k :
 - Se $\text{pred}(p) < k \leq p$, então p retorna o próprio endereço
 - Senão p repassa a requisição para os seus vizinhos

NOMEAÇÃO SIMPLES – DHT



- a) Suponha que $p = 4$ receba uma requisição para $k = 7$
→ repassa a requisição ao nó = 9 que é sucessor de p
- b) Suponha que $p = 4$ receba uma requisição para $k = 3$
→ $\text{pred}(4) = 1$
→ Como $\text{pred}(p) < k \leq p$, então p retorna o próprio endereço

N



1) Considere a resolução de $k=26$, a partir do nó 1

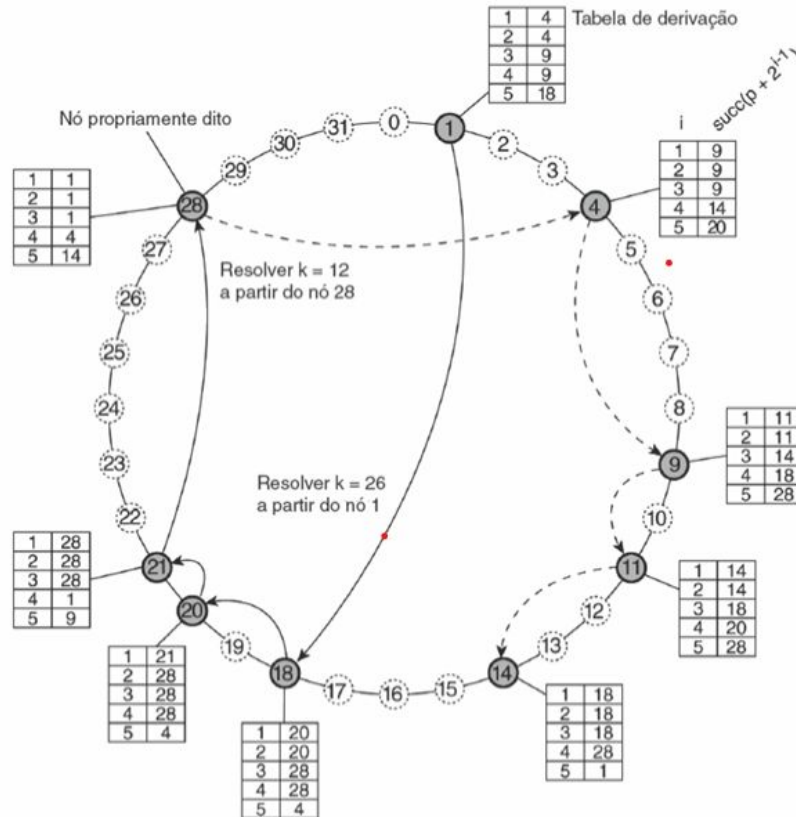
2) O nó 1 consultará $k=26$ em sua tabela de derivação
 \rightarrow verifica que o valor é maior do que $FT_1[5]$

3) A requisição será repassada para o nó 18

4) O nó 18 selecionará o nó 20, porque
 $FT_{18}[2] < k \leq FT_{18}[3]$

5) O nó 20 selecionará o nó 21, porque
 $FT_{20}[1] < k \leq FT_{20}[2]$

6) O nó 21 selecionará o nó 28, porque
 $k \leq FT_{21}[1]$



1) Considere a resolução
de $k=26$, a partir do nó 1

2) O nó 1 consultará $k=26$ em
sua tabela de derivação
→ verifica que o valor é maior do
que $FT_1[5]$

3) A requisição será repassada
para o nó 18

4) O nó 18 selecionará o nó 20,
porque
 $FT_{18}[2] < k \leq FT_{18}[3]$

5) O nó 20 selecionará o nó 21,
porque
 $FT_{20}[1] < k \leq FT_{20}[2]$

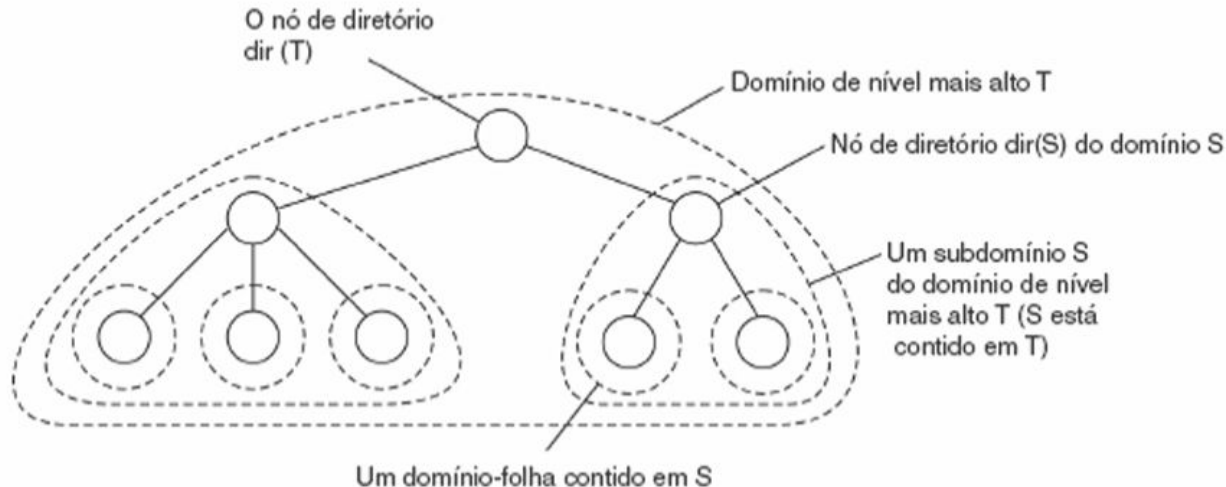
6) O nó 21 selecionará o nó 28,
porque
 $k \leq FT_{21}[1]$

NOMEAÇÃO SIMPLES – DHT

- Exploração de proximidade da rede
- Requisições podem ser roteadas erraticamente pela Internet
- Necessário levar em conta rede subjacente, nas formas:
 - 1. Identificadores de nós designados com base na topologia Nós próximos possuem identificadores próximos
 - 2. Roteamento por proximidade Nós mantêm lista de alternativas para repassar requisição Vários sucessores
 - 3. Seleção de vizinho por proximidade Otimiza tabelas de roteamento para o nó mais próximo ser selecionado como vizinho

NOMEAÇÃO SIMPLES – ABORDAGENS HIERÁRQUICAS

- Serão discutidas abordagens hierárquicas e posteriormente algumas otimizações
- Organização



NOMEAÇÃO SIMPLES – ABORDAGENS HIERÁRQUICAS

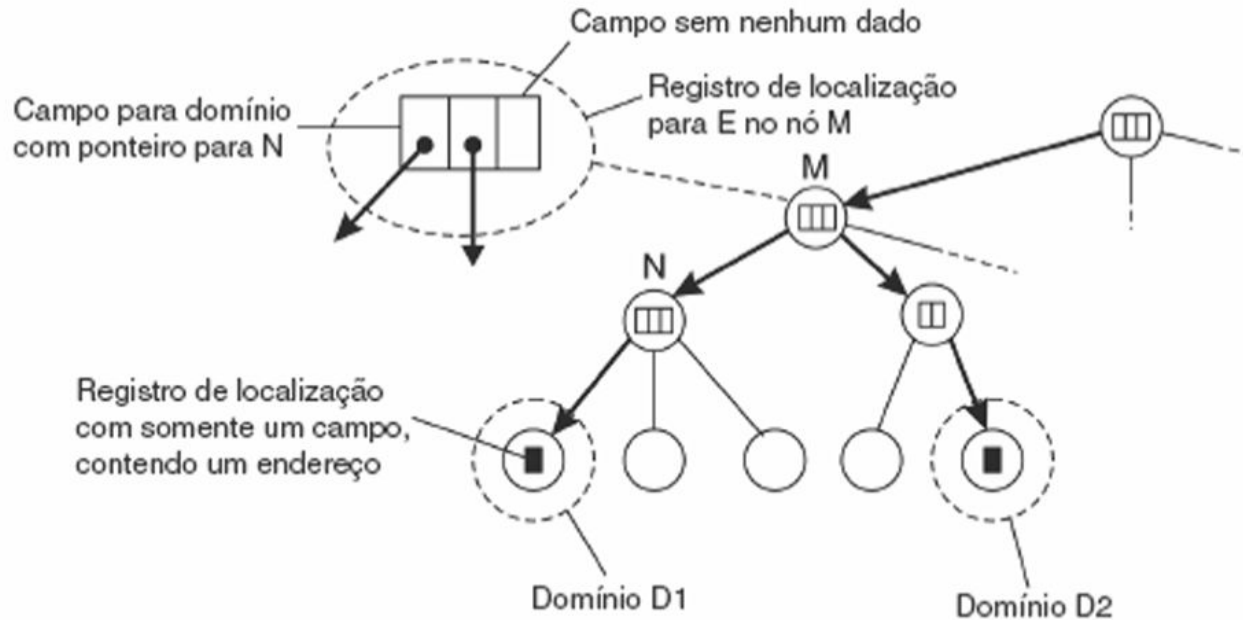
- Um esquema hierárquico:
 - Uma rede é dividida em um conjunto de domínios
 - Um domínio de nível mais alto abrange toda extensão da rede
 - Cada domínio é dividido em vários domínios menores
 - Um domínio de nível mais baixo é denominado domínio-folha normalmente corresponde a uma rede local
 - Cada domínio tem um nó diretório associado $dir(D)$ monitora as entidades no domínio
 - O nó de nível mais alto é o nó raiz conhece todas as entidades

NOMEAÇÃO SIMPLES – ABORDAGENS HIERÁRQUICAS

- Para monitorar o paradeiro de uma entidade, cada entidade possui um registro de localização
- Um registro de localização para uma entidade contém o endereço corrente da entidade naquele domínio
 - Os nós de diretório superiores conterão apontadores para seus nós imediatamente inferiores que levam em direção às entidades contidas no domínio
- Uma entidade pode ter vários endereços (ex.: replicação).
 - O nó de menor domínio contendo todos endereços mantém ponteiros para estes

NOMEAÇÃO SIMPLES – ABORDAGENS HIERÁRQUICAS

○ Replicação



NOMEAÇÃO SIMPLES – ABORDAGENS HIERÁRQUICAS

- Consultas são submetidas ao nó de diretório do domínio-folha D no qual o cliente reside
 - Caso não exista registro de localização para entidade, repassa a requisição para seu nó pai (entidade não localizada em D)
 - Quando um nó superior encontra registro da entidade, repassa a requisição para o nó filho correspondente ao endereço, até alcançar o nó que contém a entidade procurada
 - O endereço é então repassado ao nó requisitante
- Princípio da localidade: Busca primeiro nas proximidades, e no pior caso em toda a rede (através do nó raiz)

NOMEAÇÃO SIMPLES – ABORDAGENS HIERÁRQUICAS

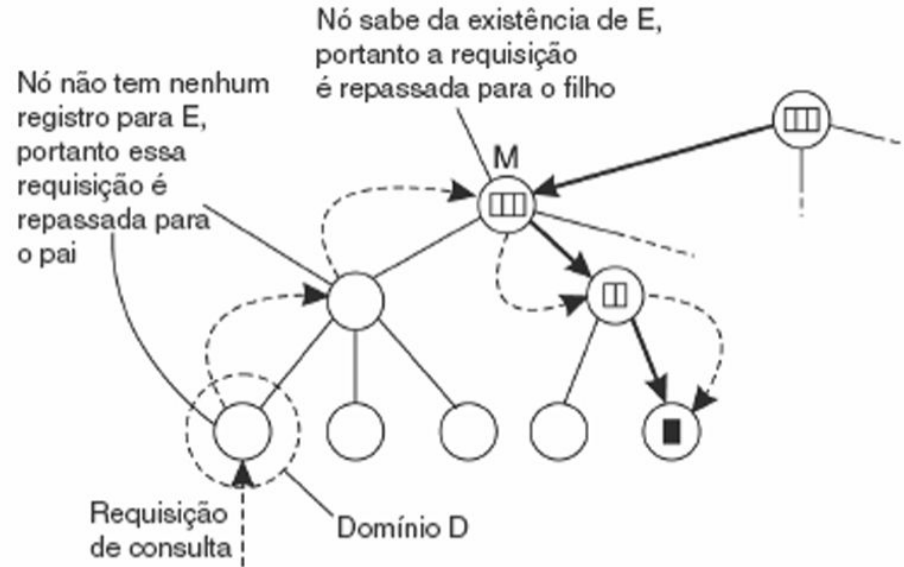


Figura 5.7 Consulta de uma localização em um serviço de localização organizado por hierarquia.

NOMEAÇÃO SIMPLES – ABORDAGENS HIERÁRQUICAS

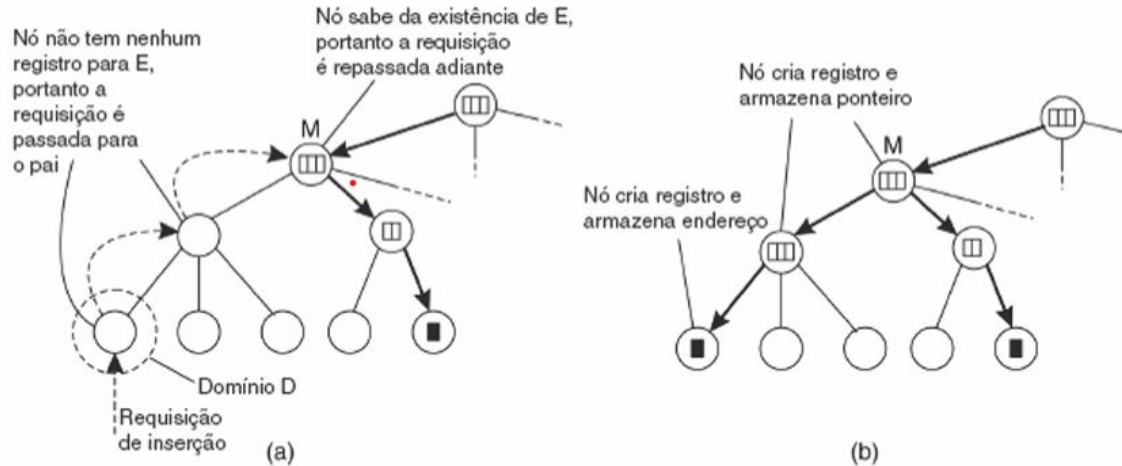


Figura 5.8 (a) Requisição de inserção é repassada para o primeiro nó, que sabe da existência da entidade E.
 (b) É criada uma cadeia de ponteiros repassadores até o nó-folha.