



# BCC 362 – Sistemas Distribuídos

Joubert de Castro Lima – [joubertlima@gmail.com](mailto:joubertlima@gmail.com)  
Professor Adjunto – DECOM

**UFOP**

# ANTES DE PRATICARMOS OS TEMAS

- 1- NOMEAÇÃO,
- 2- LOCALIZAÇÃO,
- 3- SINCRONIZAÇÃO,
- 4- TOLERÂNCIA A FALHAS,
- 5- RECOVERY.



# LISTA 1

1) Explique os tipos de transparência:

Transparência de acesso

Transparência de localização

Transparência de migração

Transparência de relocação

Transparência de replicação

Transparência de concorrência

Transparência à falha



2) quanto mais transparente melhor o sistema ??? justifique.

- 3) Um sistema aberto é desejável, mas para que isto aconteça precisamos garantir Interoperabilidade e portabilidade. Redefina um sistema distribuído aberto usando o que vem a ser interoperabilidade e portabilidade.
- 4) o que significa ser um sistema escalável em relação ao tamanho?
- 5) Explique e exemplifique 3 técnicas que auxiliam na melhora da escalabilidade?
- 6) Um amigo seu chega com a seguinte inquietação: qual a diferença entre computação em cluster e em grade ? Explique o que entende sobre ambas, mostrando as diferenças.
- 7) hoje em dia está em foco sistemas pervasivos. Geralmente, tais sistemas são distribuídos, portanto defina o que entende sobre sistemas pervasivos.
- 8) Arquiteturas de sistemas mostram como componentes de software devem ser organizados e como se comunicam. Defina componentes de software.
- 9) Monte a arquitetura do sistema sendo feito no trabalho prático 2. Detalhes, detalhes...

10) Além de componentes, precisamos de conectores em nossas arquiteturas. Defina conectores e redesenhe sua solução, mostrando os conectores e componentes reunidos.

11) Explique uns 3 estilos arquitetônicos. Qual você escolheu em seu trabalho prático 2 e por que?

12) Autores sugerem que trabalhar com os conceitos cliente e servidor é uma boa. De forma análoga, dividir um sistema em camadas também é recomendado. Quais as camadas mínimas? Descreva cada uma delas.

13) Clientes podem ser magros ou gordos. Explique esta afirmação usando o modelo de camadas da arquitetura cliente servidor.

14) Explique distribuição vertical e horizontal.

15) Explique uma solução para o trabalho 2 em forma de camadas (o cliente e o servidor). Descreva brevemente a funcionalidade de cada uma das camadas citadas.

16) Como um sistema em camadas funciona?

17) O que é um processo? Diferencie processo de código.

18) Explique o seu trabalho prático 2 usando processos. Desde o processo cliente que pode ou não chamar outros processos clientes (threads). Um processo cliente se comunica com.....

19) o que são threads? Quais as diferenças entre um processo e uma thread, também chamada light process.

20) O que são seções críticas? Como resolver tal problema?

21) Explique o problema do produtor consumidor. Detalhe ao máximo o que entendeu sobre tal estratégia de paralelização. Por que também é chamado pipeline.

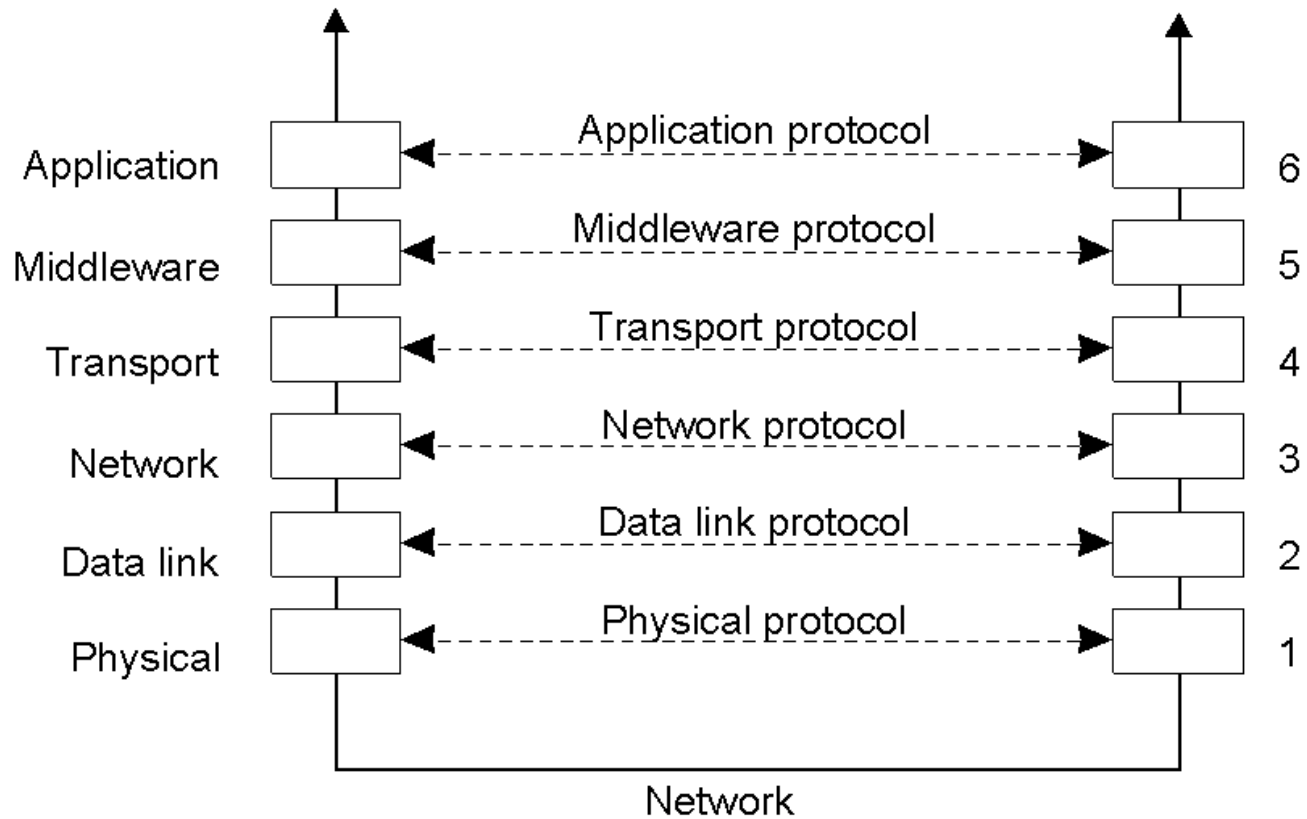
22) Porque é interessante criarmos múltiplos processos, mesmo com um único Processador ?

23) Explique o que entende sobre virtualização. Por que podemos afirmar que em nosso curso estamos a virtualizar.

24) Servidores podem ser iterativos ou concorrente, com ou sem estado. Explique estes tipos, exemplifique cada tipo e classifique o tipo de servidor implementado em seu trabalho prático 2.

25) O que significa vinculação processo-recurso?? Quais tipos de vinculação existem? Qual você recomenda e por que?

26) Explique as camadas e os protocolos da figura abaixo





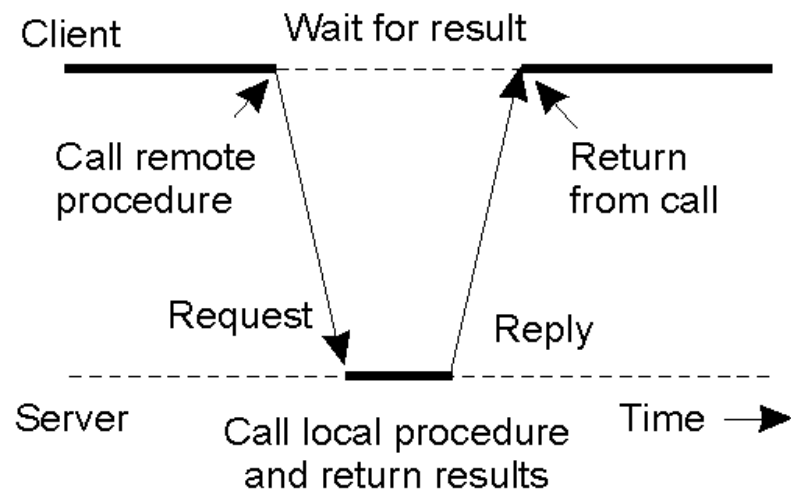
27) quais os 4 tipos de comunicação poderemos oferecer em nosso middleware ?  
Detalhe cada uma delas.

28) o que você entende sobre o conceito de pilha em arquitetura de computadores?  
Exemplifique a pilha de um programa qualquer.

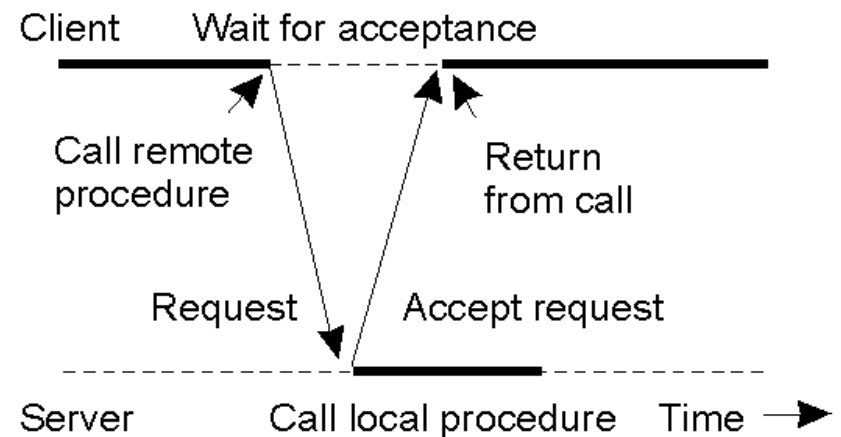
29) Como uma chamada remota de procedimento ocorre. Utilize em sua explicação o conceito de stubs no cliente e no servidor.

30) você utilizou stubs em seu trabalho prático 2? Justifique.

31) Diferencie e explique cada tipo de comunicação na figura abaixo

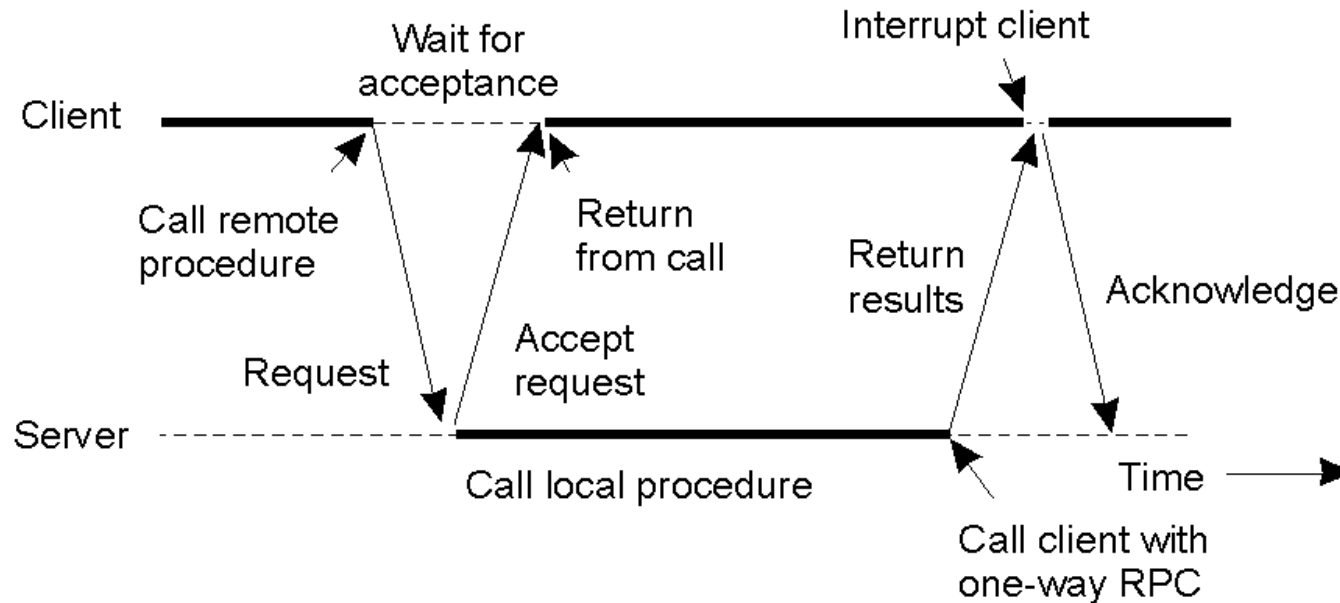


(a)

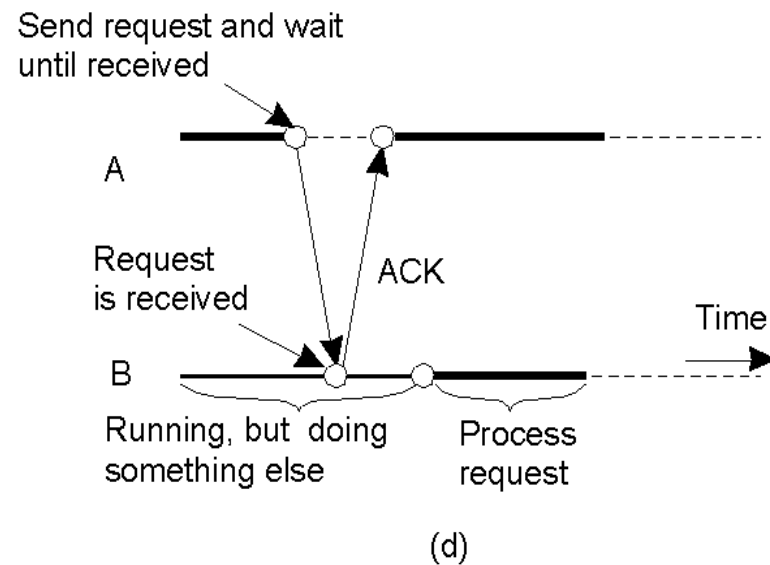
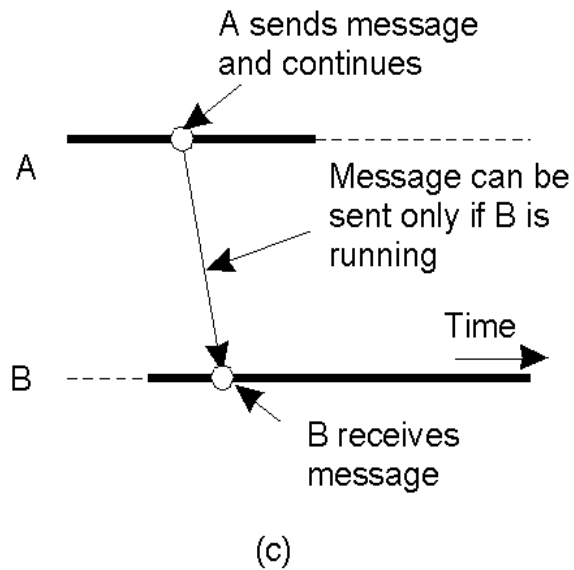
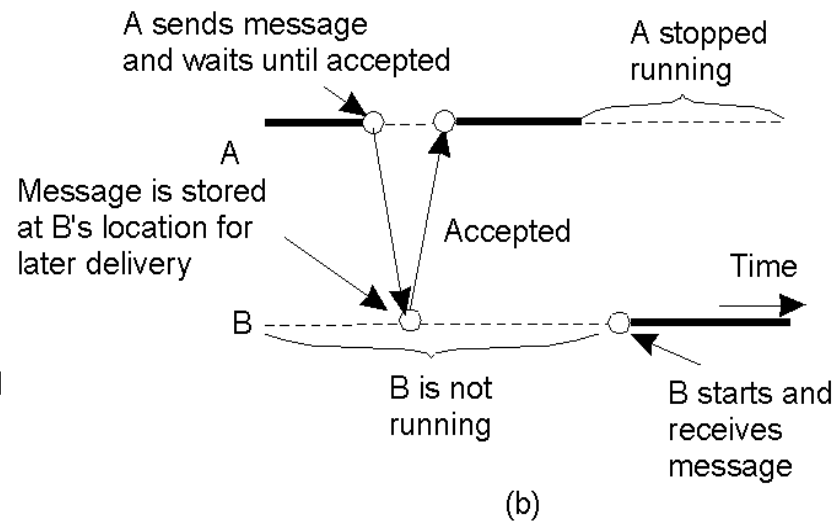
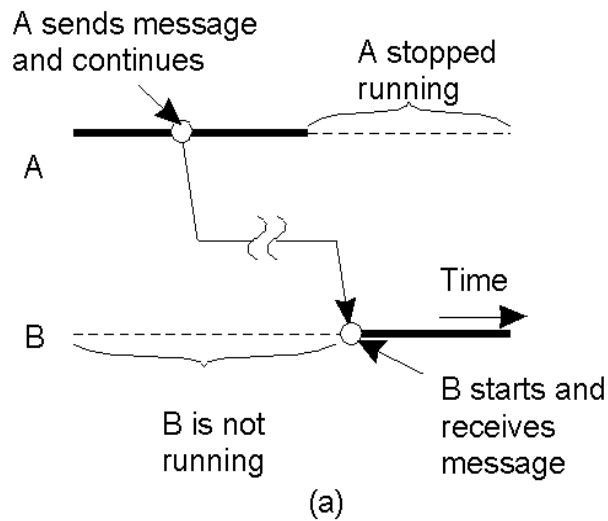


(b)

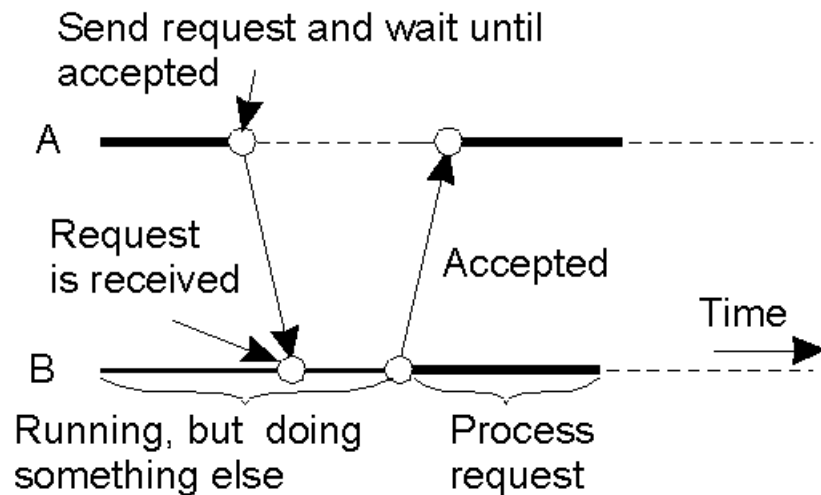
32) Relate o que se passa na figura abaixo. Comece a explicar da esquerda para a direita na figura e sempre pelo cliente, pois este inicia todo o processo. Não esqueça de relatar qual o tipo de comunicação ocorre (síncrona ou assíncrona).



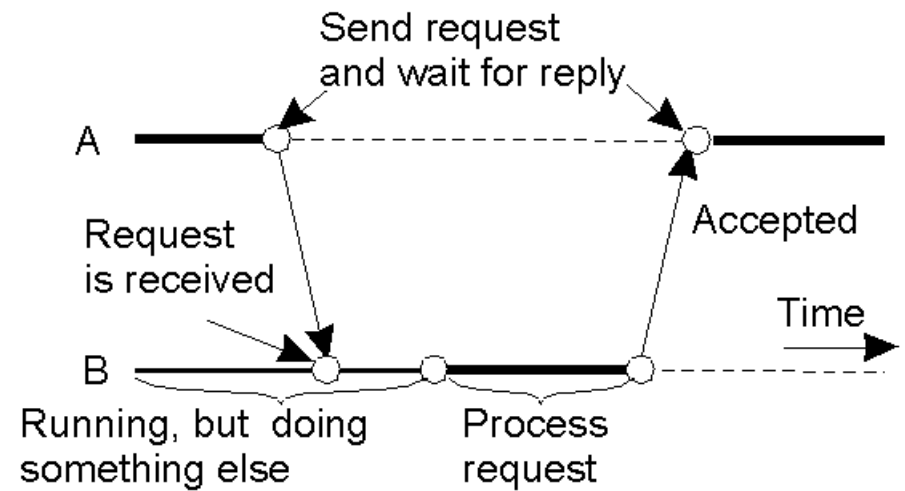
### 33) Diferencie e explique os tipos de comunicação das figuras abaixo



34) Diferencie e explique os tipos de comunicação da figura abaixo. O que diferem dos tipos apresentados na questão 33?



(e)



(f)

35) qual o tipo de comunicação você implementou em seu trabalho prático 2? Quais outros são possíveis de serem implementados em trabalhos futuros?? Utilize as questões 33 e 34 como base.



FIM DA LISTA 1

# Nomeação

**Slides do livro Sistemas Distribuídos – Andrew S. Tanenbaum**

# Nomeação

A name in a distributed system is a string of bits or characters that is used to refer to an entity.

To operate an entity we need an access point.  
The name of an access point is called an address.

Example: a telephone can be viewed as an access point of a person and the number corresponds the address

Example: In a distributed system an access point is a host running a specific server, with its address formed by the combination of IP : port.

# Name spaces

Names in a distributed system are organized into what is commonly referred to as a name space.

Name space can be represented as a labeled, directed graph with two types of nodes.

**Leaf node** represents a named entity and has the property that it has no outgoing edges.

**Directory node** has a number of outgoing edges each labeled with a name. It stores a table in which an outgoing is represented as a pair (edge label, node identifier)



# Name spaces

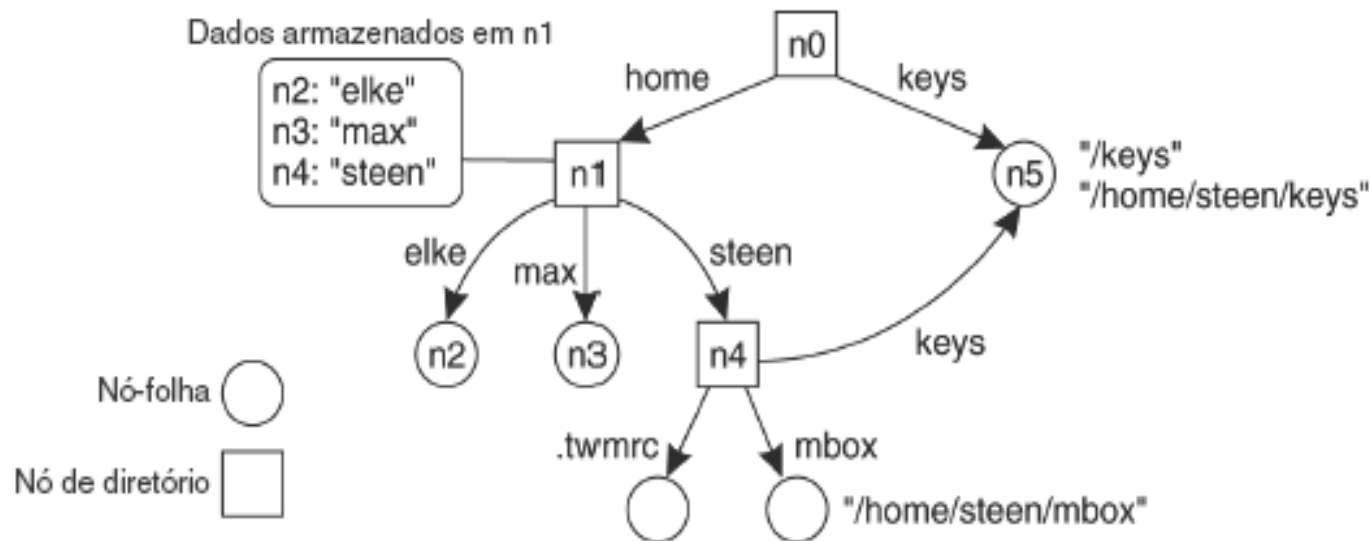


Figura 5.9 Gráfico de nomeação geral com um único nó-raiz.

# Name spaces

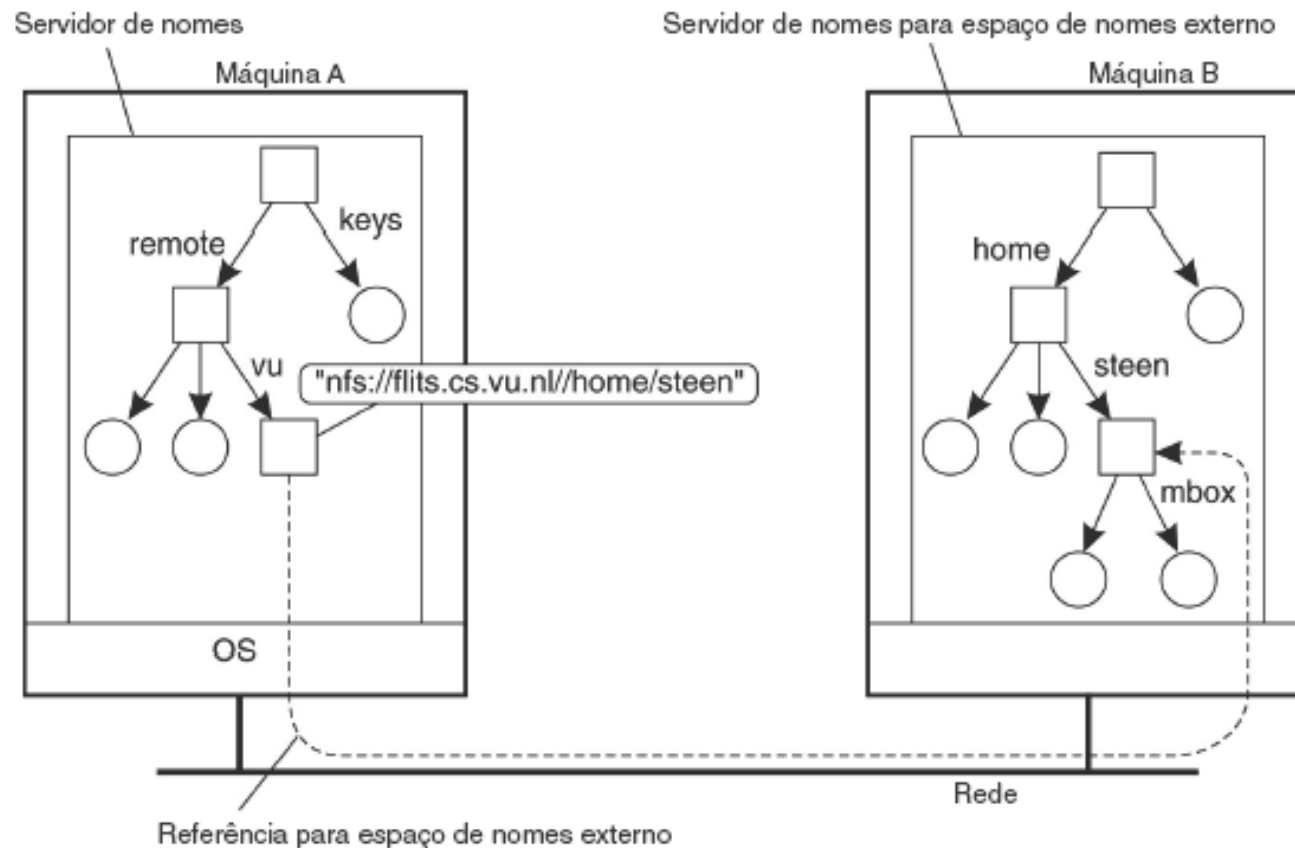


Figura 5.12 Montagem de espaços de nomes remotos por meio de um protocolo de acesso específico.

# **\*\* Name Space Distribution**

To effectively implement such a name space, it is convenient to partition it into logical layers.

**Global layer** is formed by the highest level nodes.

**Administrational layer** is formed by directory nodes that together are managed within a single organization.

**Managerial layer** consists of nodes that may typically change regularly.

# Name Space Distribution

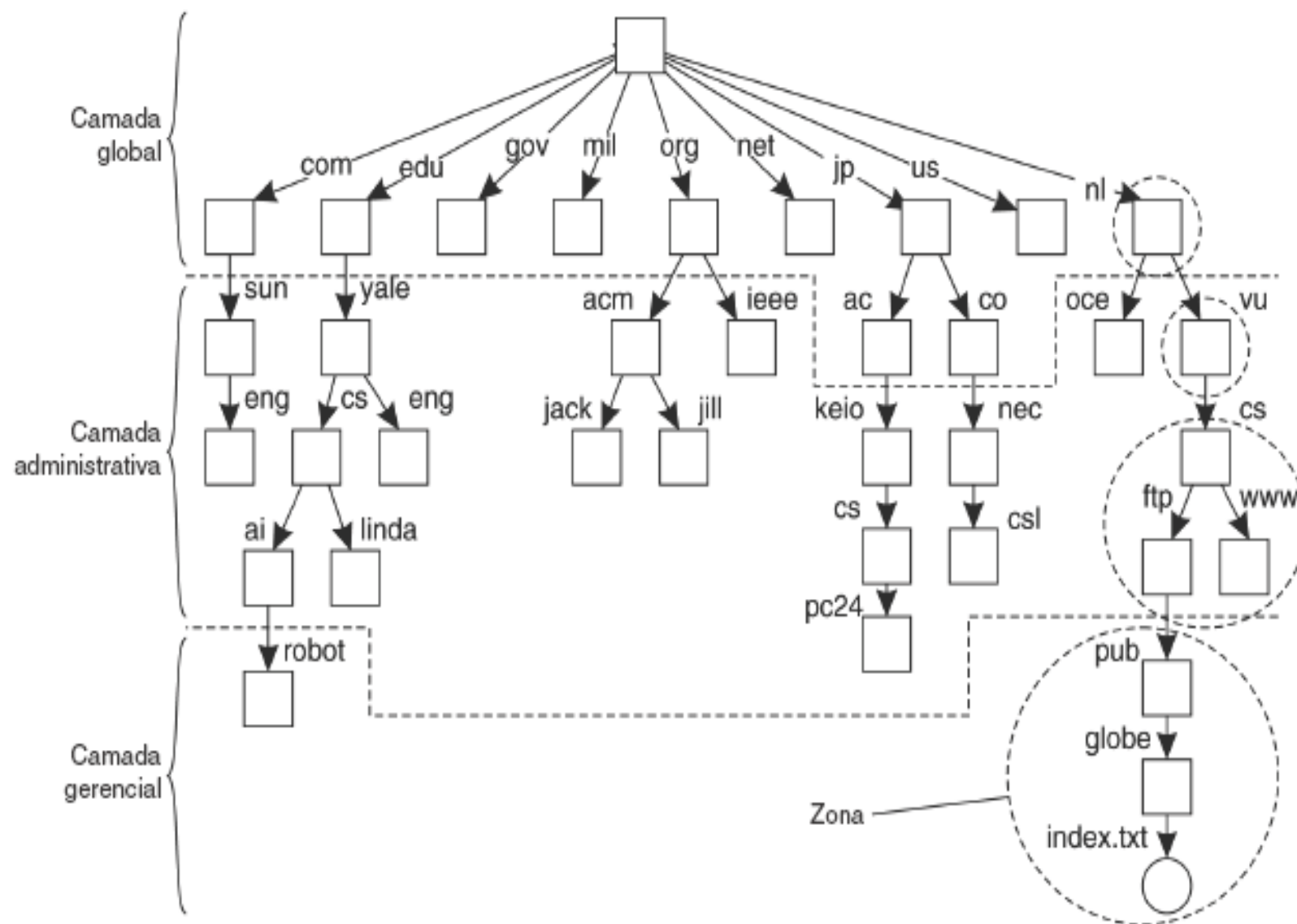


Figura 5.13 Exemplo de repartição do espaço de nomes DNS, incluindo arquivos acessíveis pela Internet, em três camadas.

# **\*\* Name Space Distribution**

<b>Item</b>	<b>Global</b>	<b>Administrational</b>	<b>Managerial</b>
Geographical scale of network	Worldwide	Organization	Department
Total number of nodes	Few	Many	Vast numbers
Responsiveness to lookups	Seconds	Milliseconds	Immediate
Update propagation	Lazy	Immediate	Immediate
Number of replicas	Many	None or few	None
Is client-side caching applied?	Yes	Yes	Sometimes

A comparison between name servers for implementing nodes from a large-scale name space partitioned into a global layer, as an administrational layer, and a managerial layer.

# Implementation of name resolution

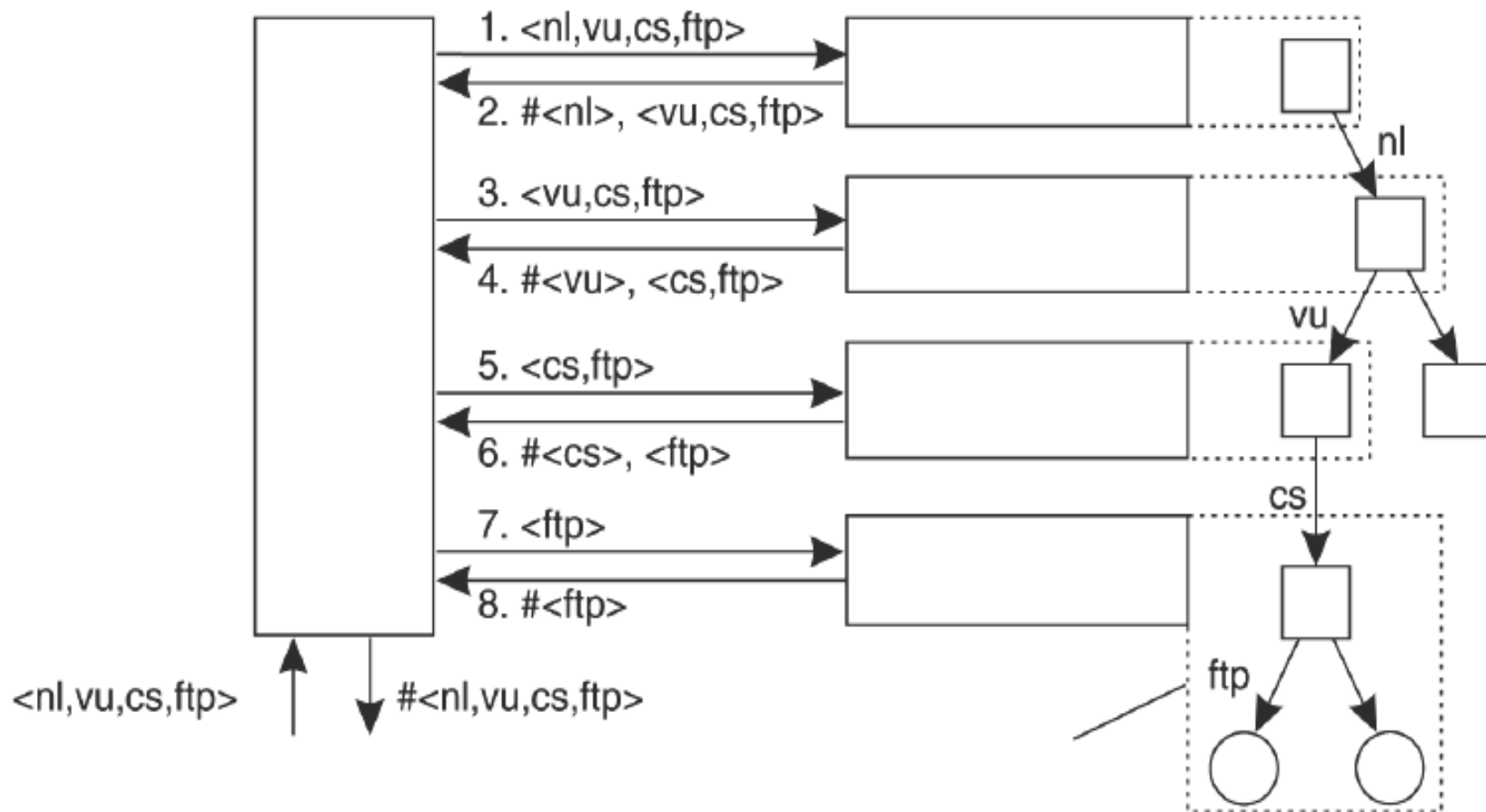


Figura 5.14 Princípio da resolução iterativa de nomes.

# Implementation of name resolution

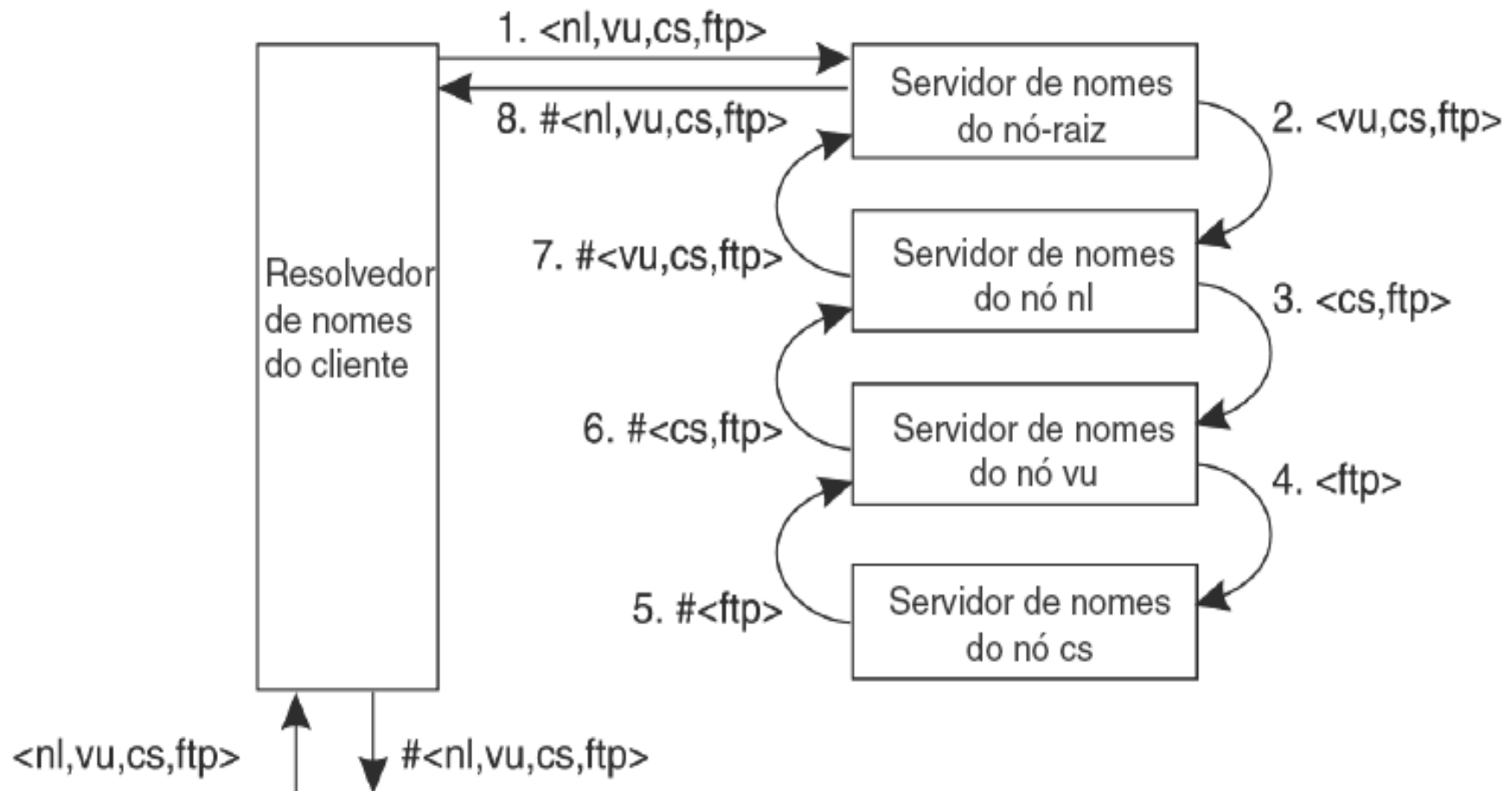
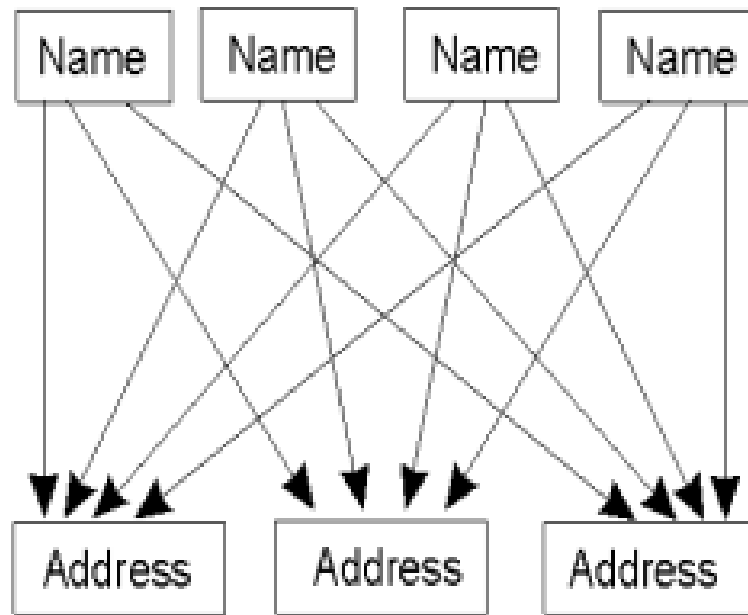
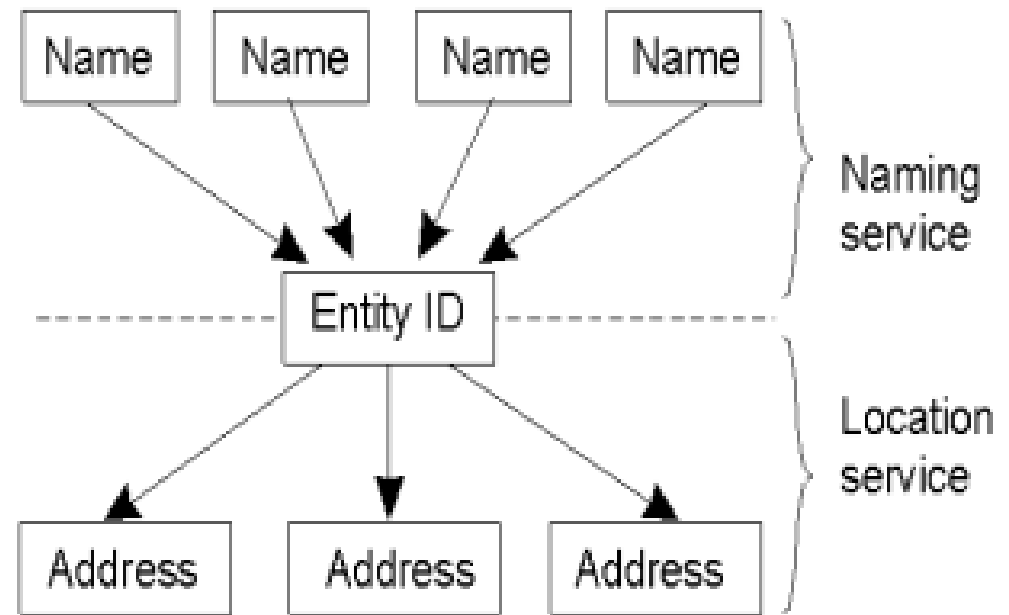


Figura 5.15 Princípio da resolução recursiva de nomes.

# Naming versus Locating Entities



(a)

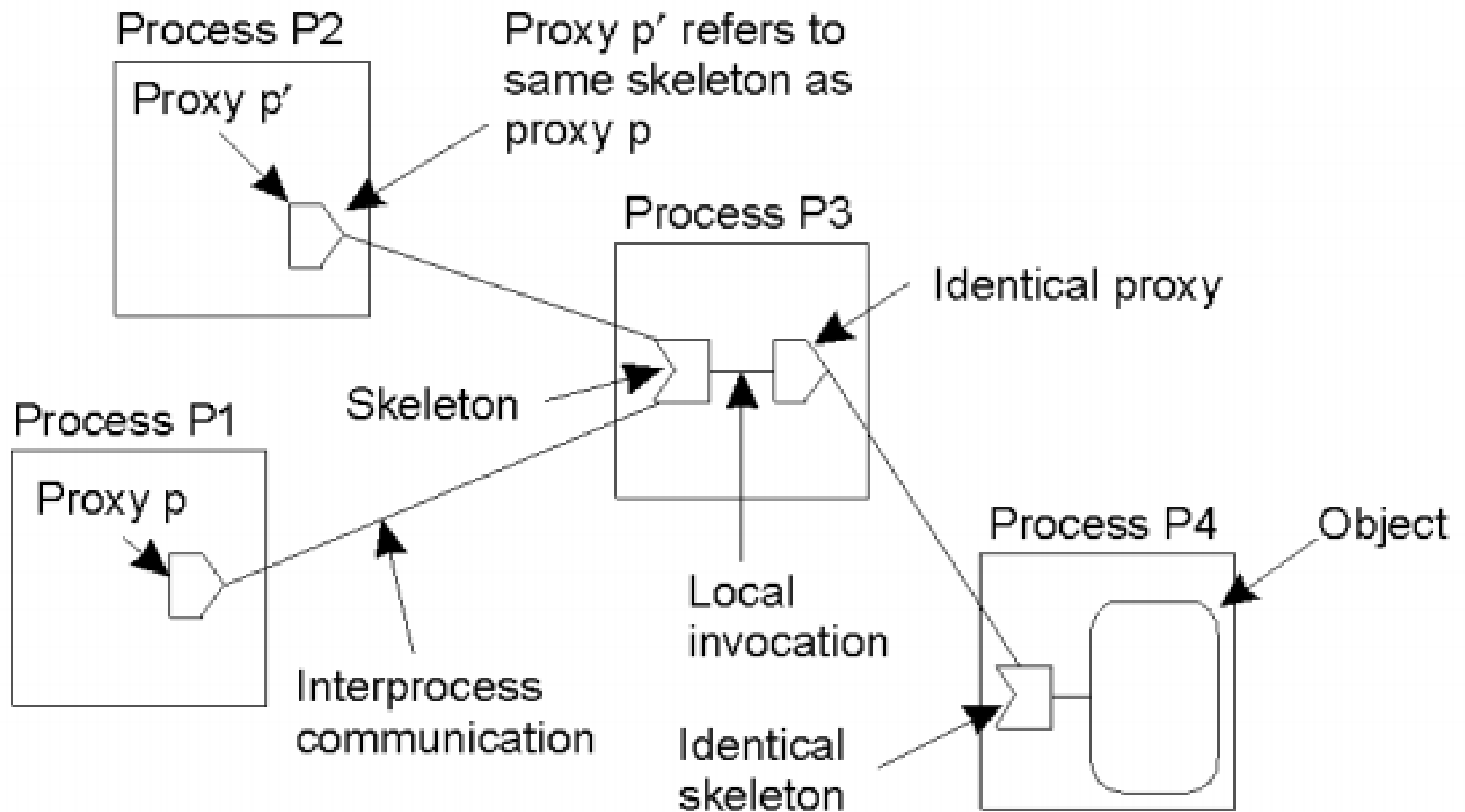


(b)

- a) Direct, single level mapping between names and addresses.  
(Traditional DNS)
- b) T-level mapping using identities. (More adequate approach for mobile entities, used in location service problem)

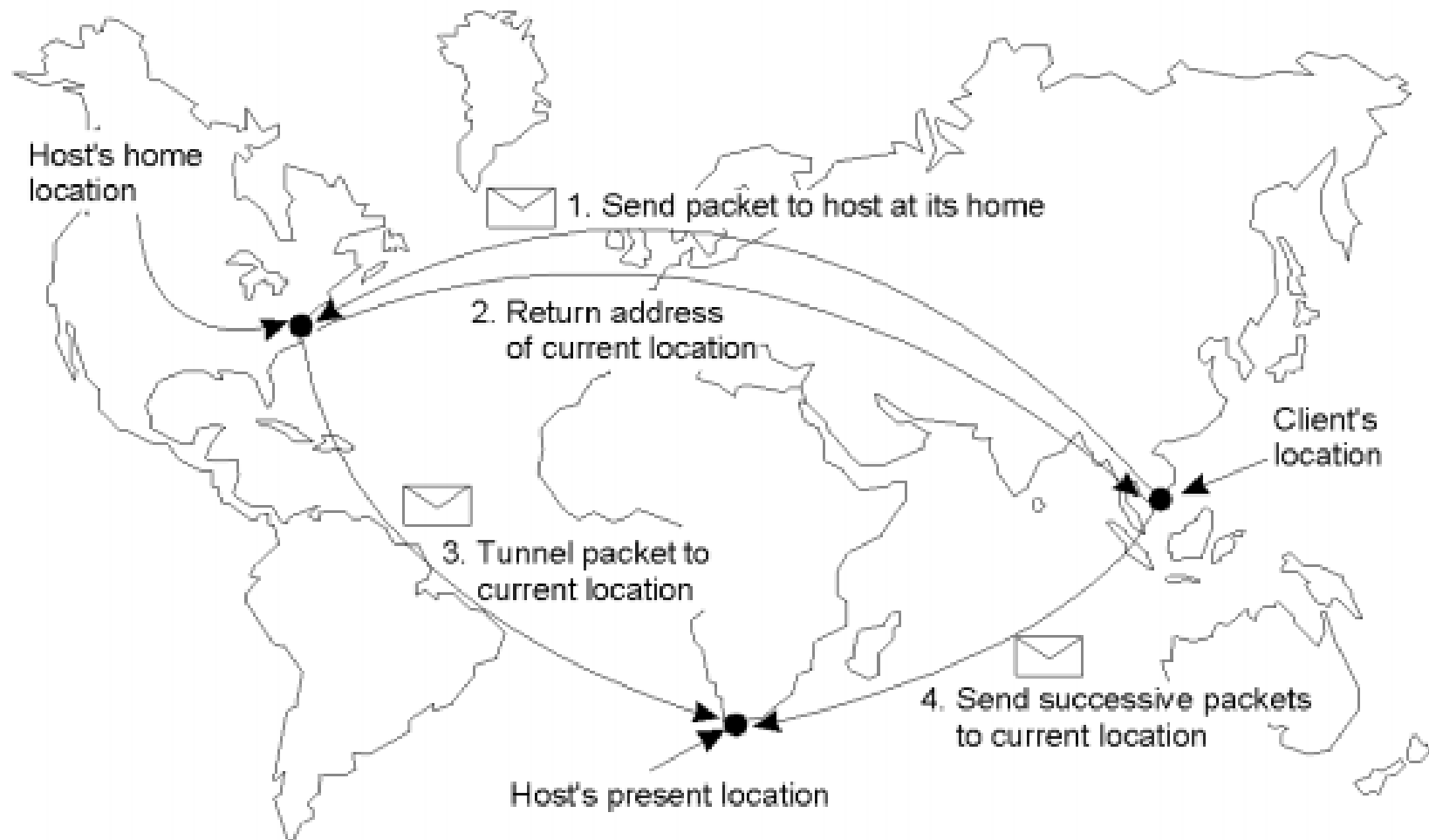


# Locating



The principle of forwarding pointers using *(proxy, skeleton)* pairs. This example considers distributed objects locating.

# Home-Based Approaches



The principle of Mobile IP.

# Hierarchical approaches

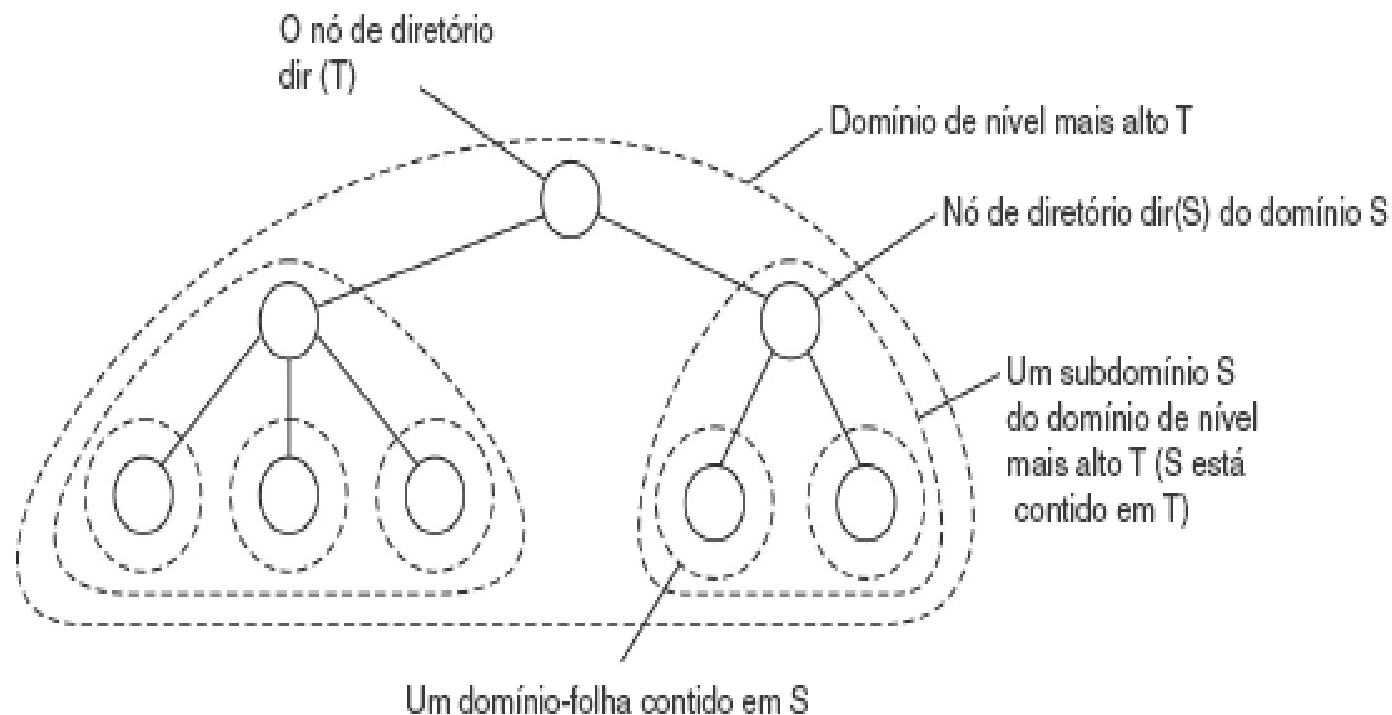


Figura 5.5 Organização hierárquica de um serviço de localização em domínios, cada um com um nó de diretório associado.

# Hierarchical approaches

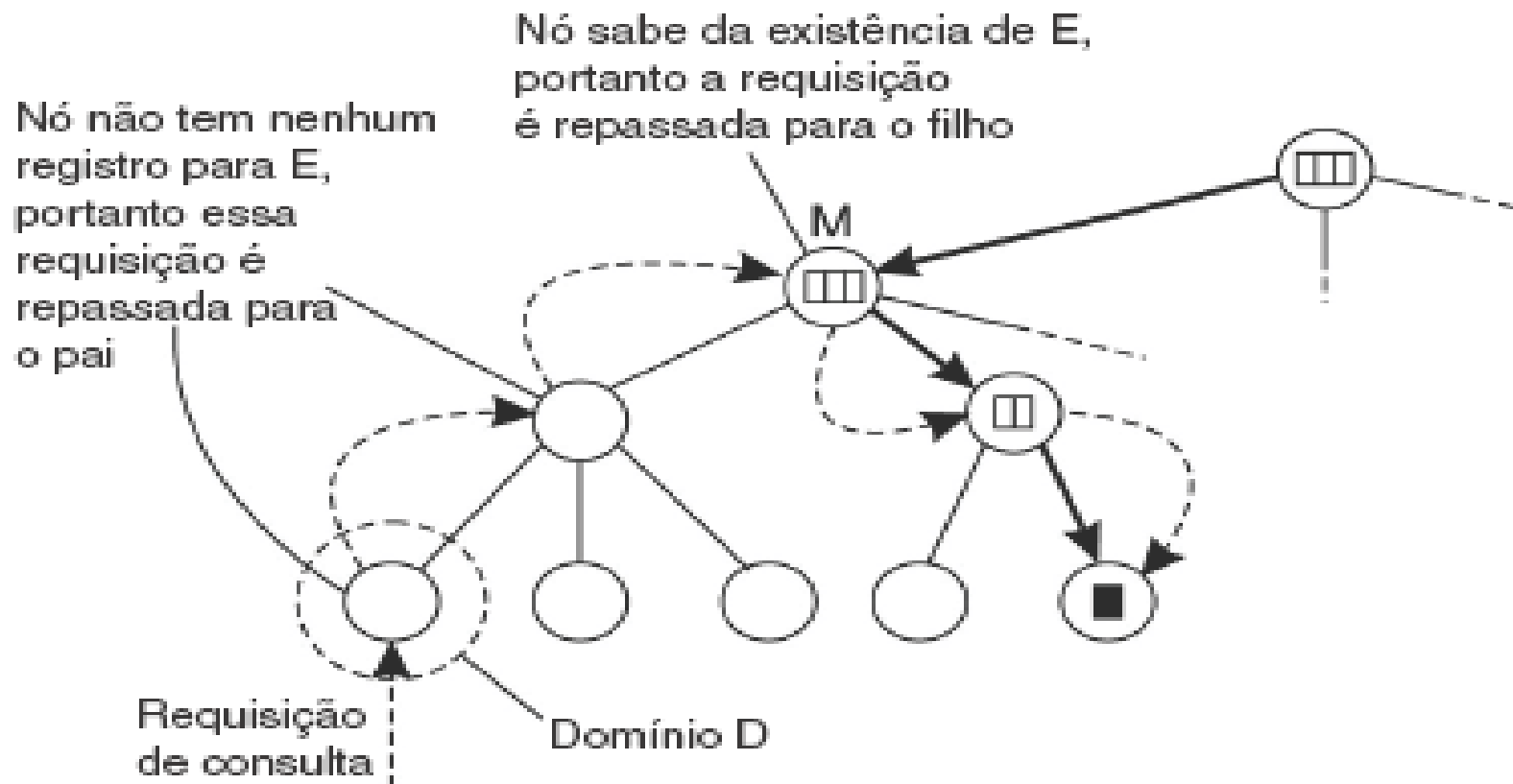


Figura 5.7 Consulta de uma localização em um serviço de localização organizado por hierarquia.

# Problemas de escalabilidade e referência perdida

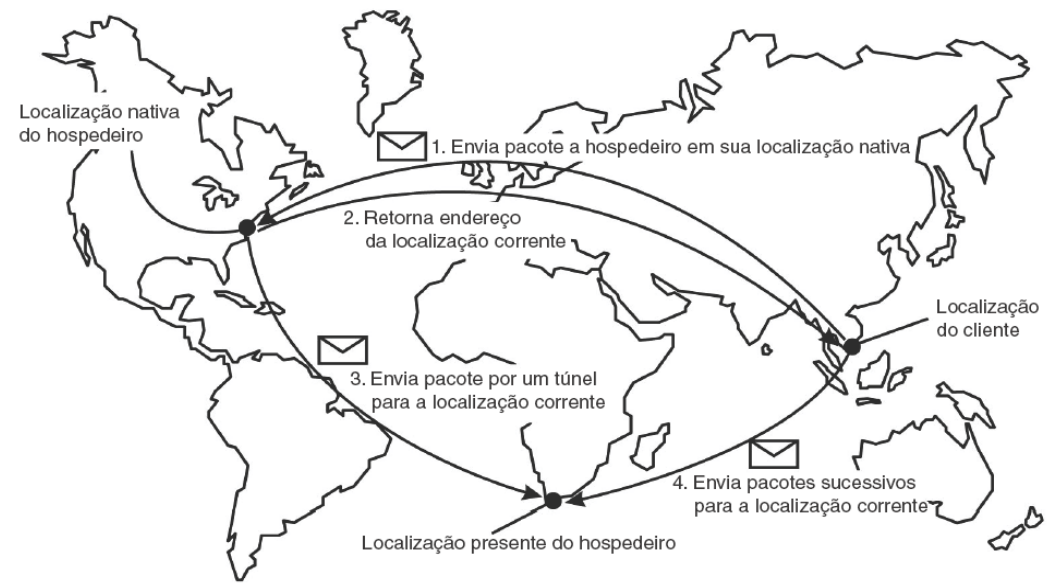


Figura 5.3 Princípio do Mobile IP.

