

第五章 命名系统

- 名称、标识符和地址
- 无层次命名
- 结构化命名



名称、标识符和地址

- **名称**：由位或字符串组成的串，用来指向一个**实体**
- 一个实体的**访问点**的地址称为该实体的**地址**：特定服务器的地址：**IP地址+端口**
- **标识符**：具有以下属性的名称：
 - 一个标识符最多指向一个实体
 - 每个实体最多由一个标识符指向
 - 一个标识符始终指向一个实体
 - 使用位串来表示
- 易于理解的**名称**：为用户制定，字符串表示

无层次命名

- 问题：给定实体的一个无结构的名称（如标识符），如何定位该实体？
- 解决方案：
 - 简单方法
 - 基于起始位置的方法
 - 分层方法



简单方法

- 广播和多播
- 转发指针

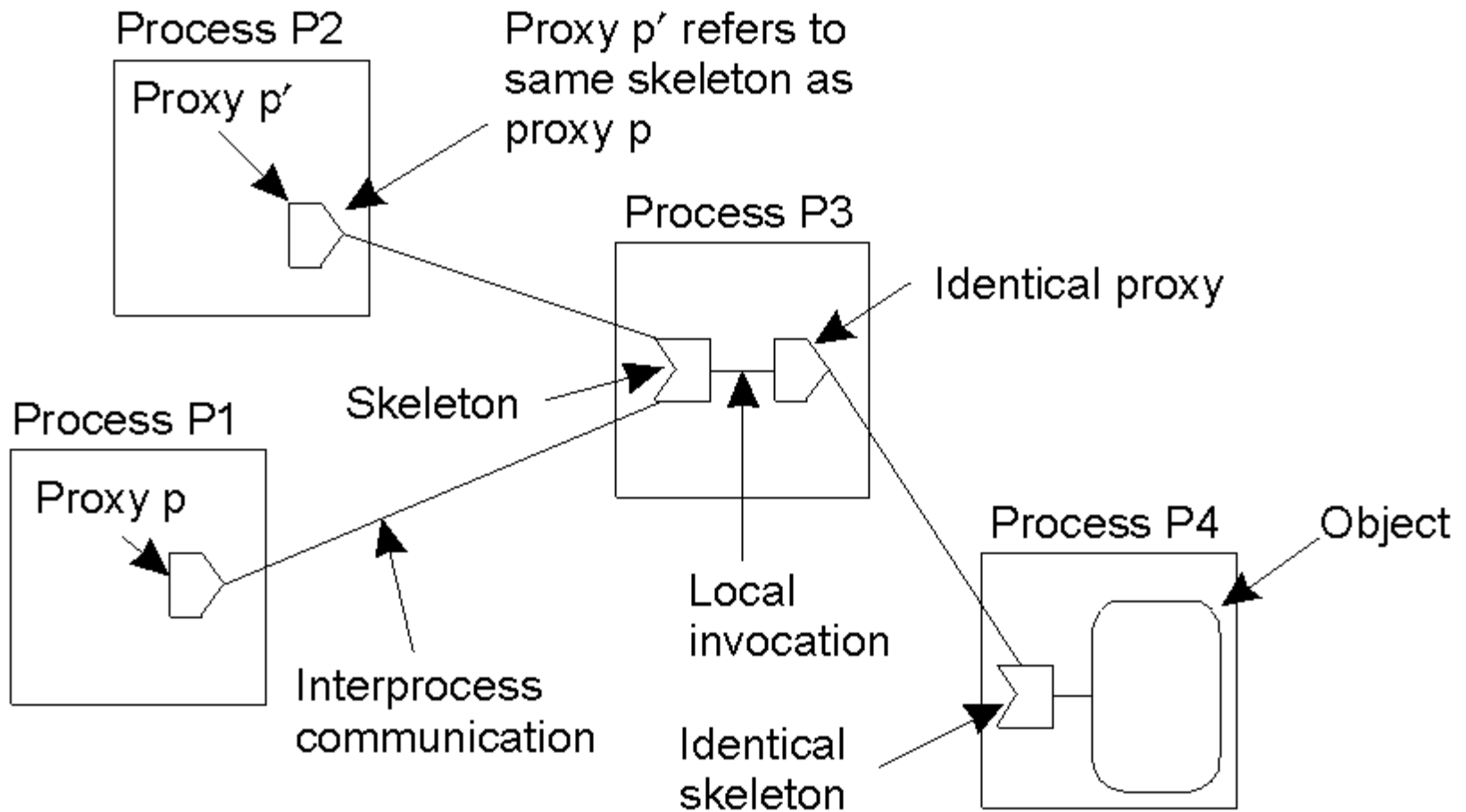


广播和多播

- 广播
 - 适用于局域网
 - 在广域网内变得低效
- 多播
 - 只发送给一组符合条件的主机
 - 可进行多播实体的定位服务
 - 可用于定位最近副本

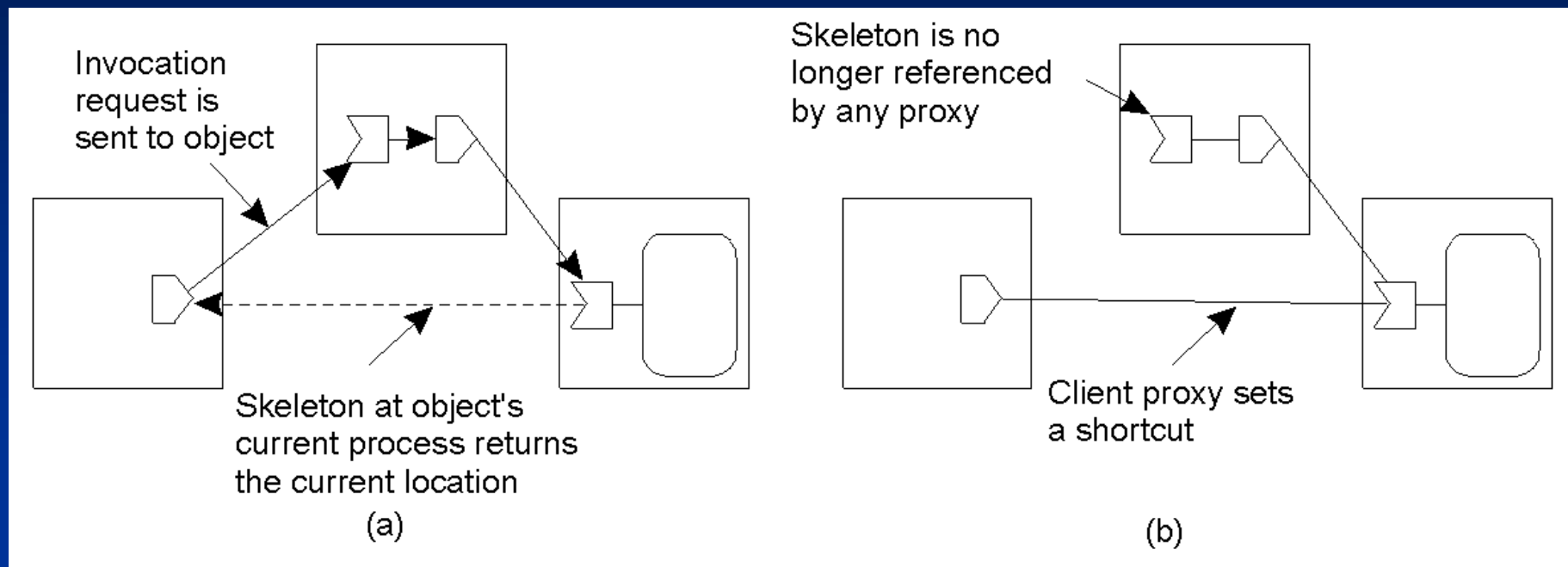


转发指针 (1)



使用 (*proxy*, *skeleton*) 对的转发指针

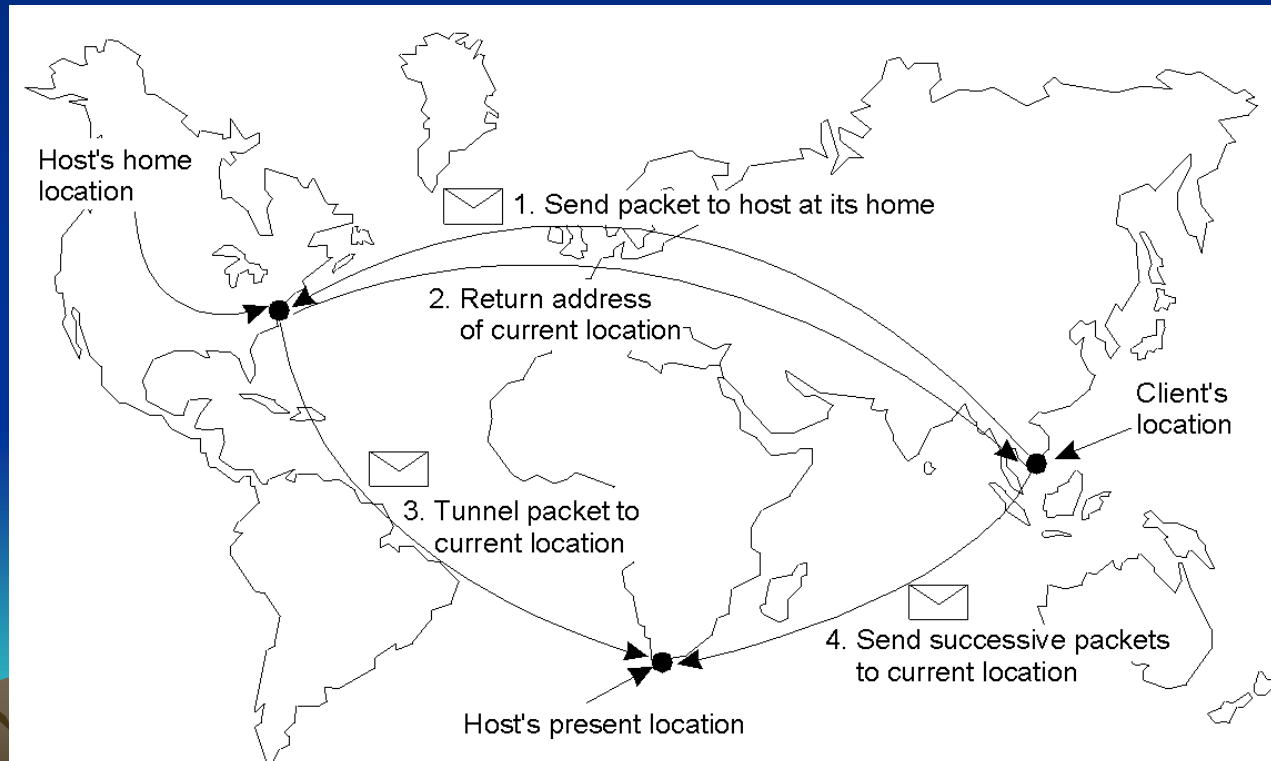
转发指针 (2)



通过在代理中存储一个捷径来重定向转发指针

基于起始位置的方法

- 起始移动代理
- 当移动主机转移到另一个网络时，将转交地址在起始移动代理中注册
- 缺点：
 - 客户必须首先与起始位置联系，而这一位置可能与实体相邻很远
 - 使用了固定的起始位置



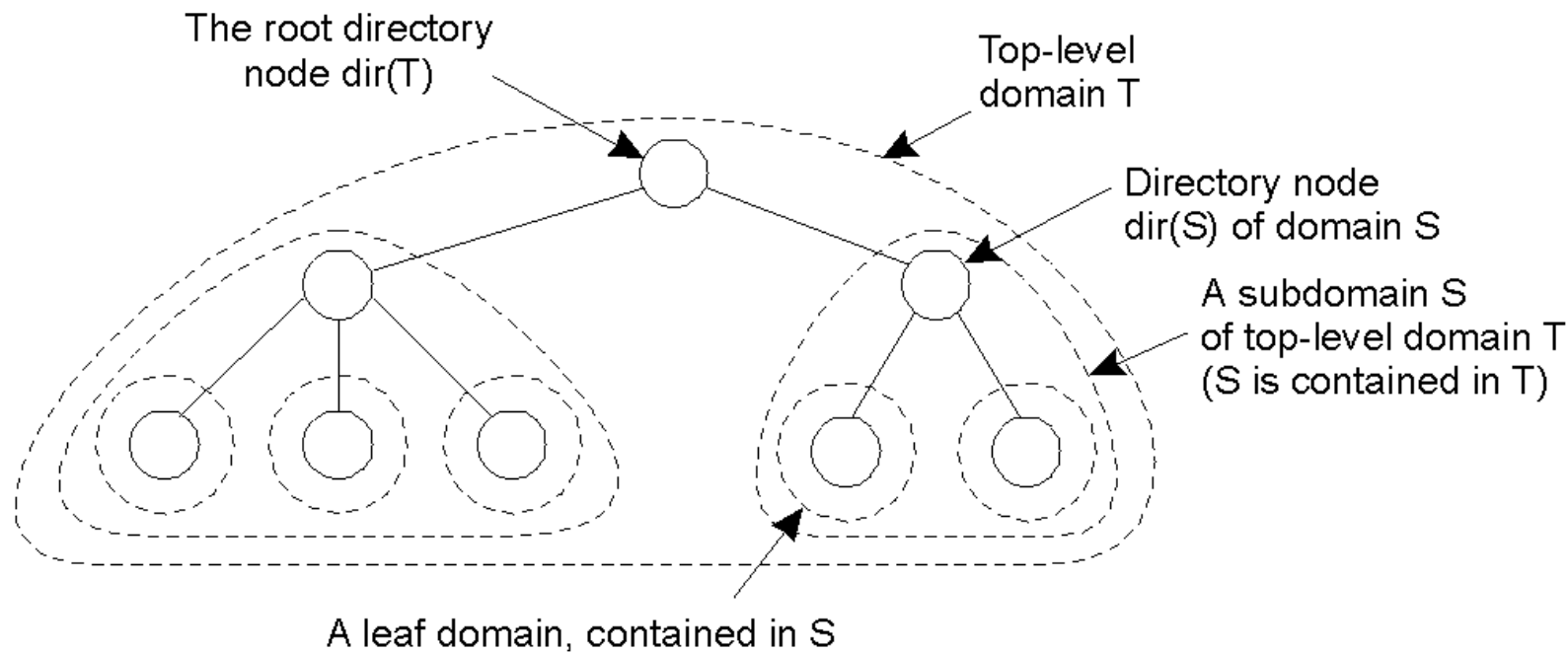
移动 IP 原理

分层方法

- 类似**DNS**，网络被划分为一组域
- 目录节点：记录域包含的实体
 - 叶域的目录节点**N**记录实体**E**在域中的位置
 - 更高一层域的目录节点**N'**记录实体**E**的位置，包含指向**N**的指针

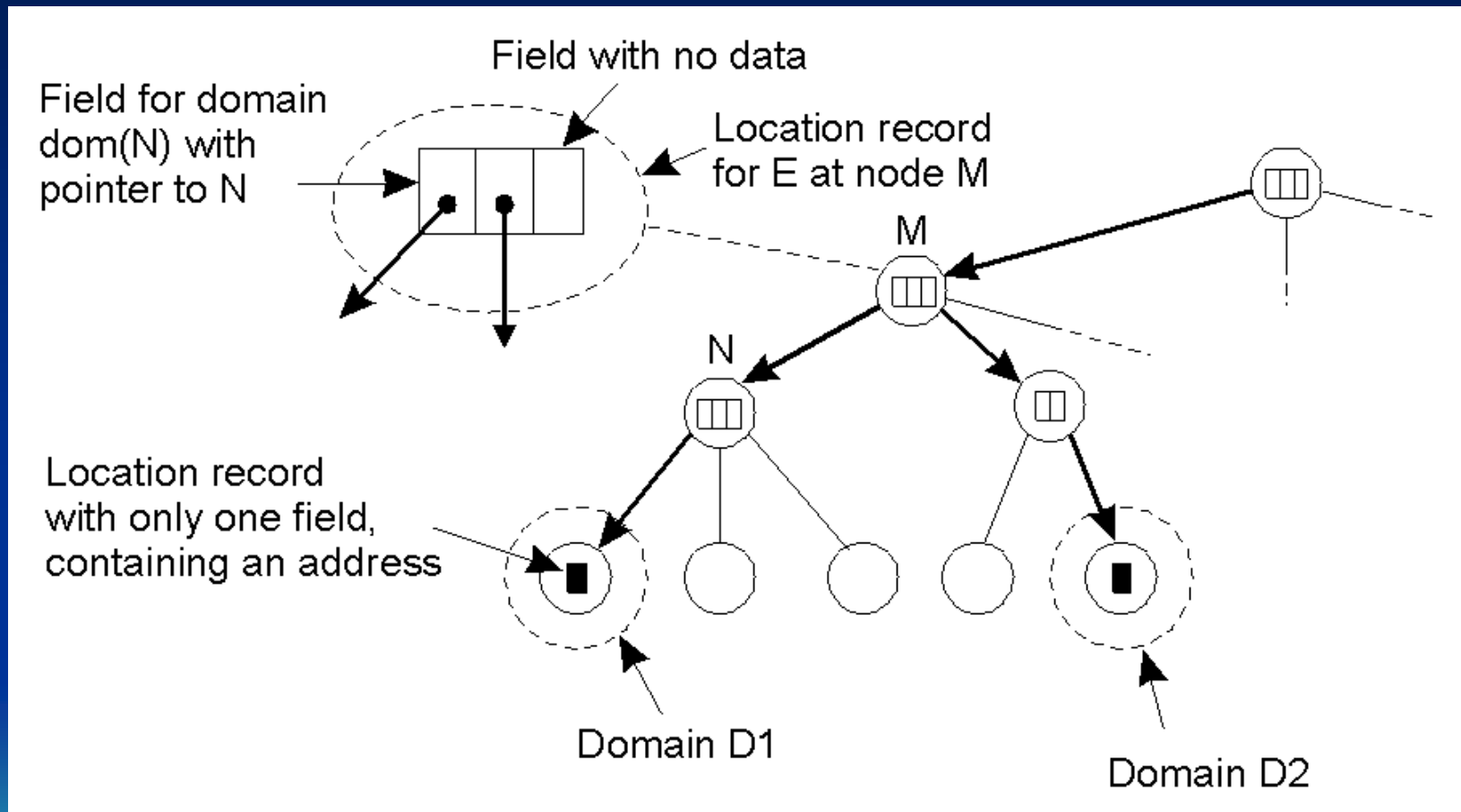


分层方法 (1)



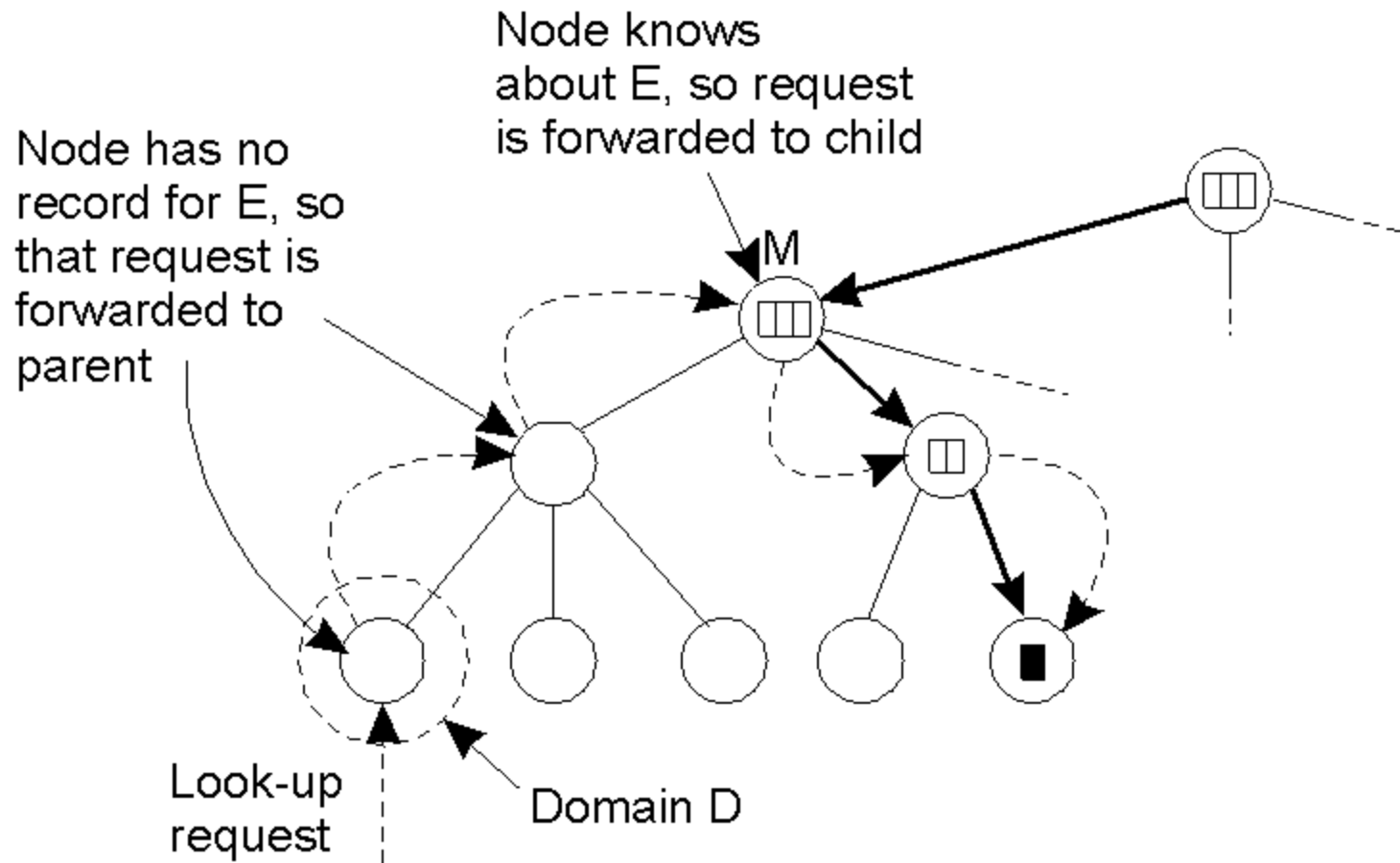
把定位服务划分为域的分层组织，每个域都有关联的目录节点

分层方法 (2)



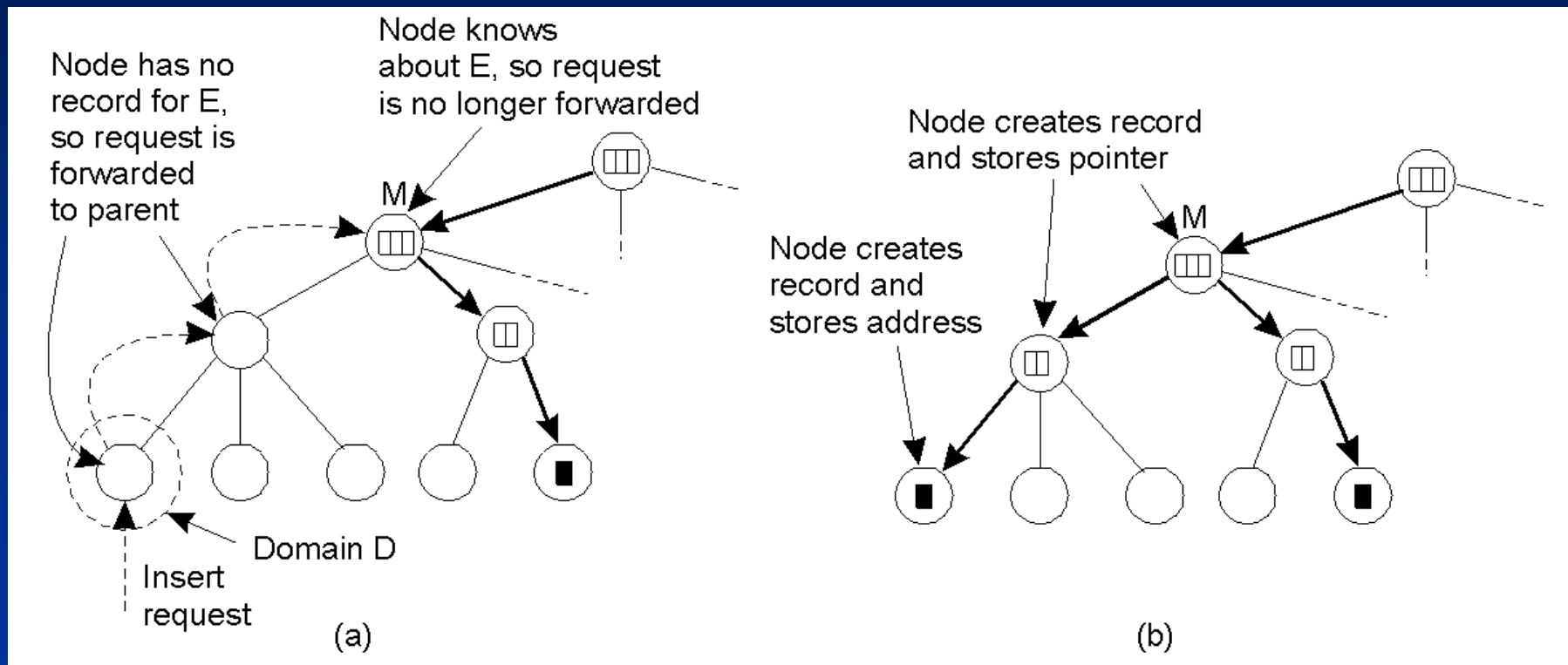
实体信息存储示例，这里的实体在不同的叶域中拥有两个地址

分层方法 (3)



在分层组织的定位服务中的位置查找

分层方法 (4)



更新操作

- 插入请求被转发到第一个知道实体E的节点
- 转发指向叶节点的指针所形成的链

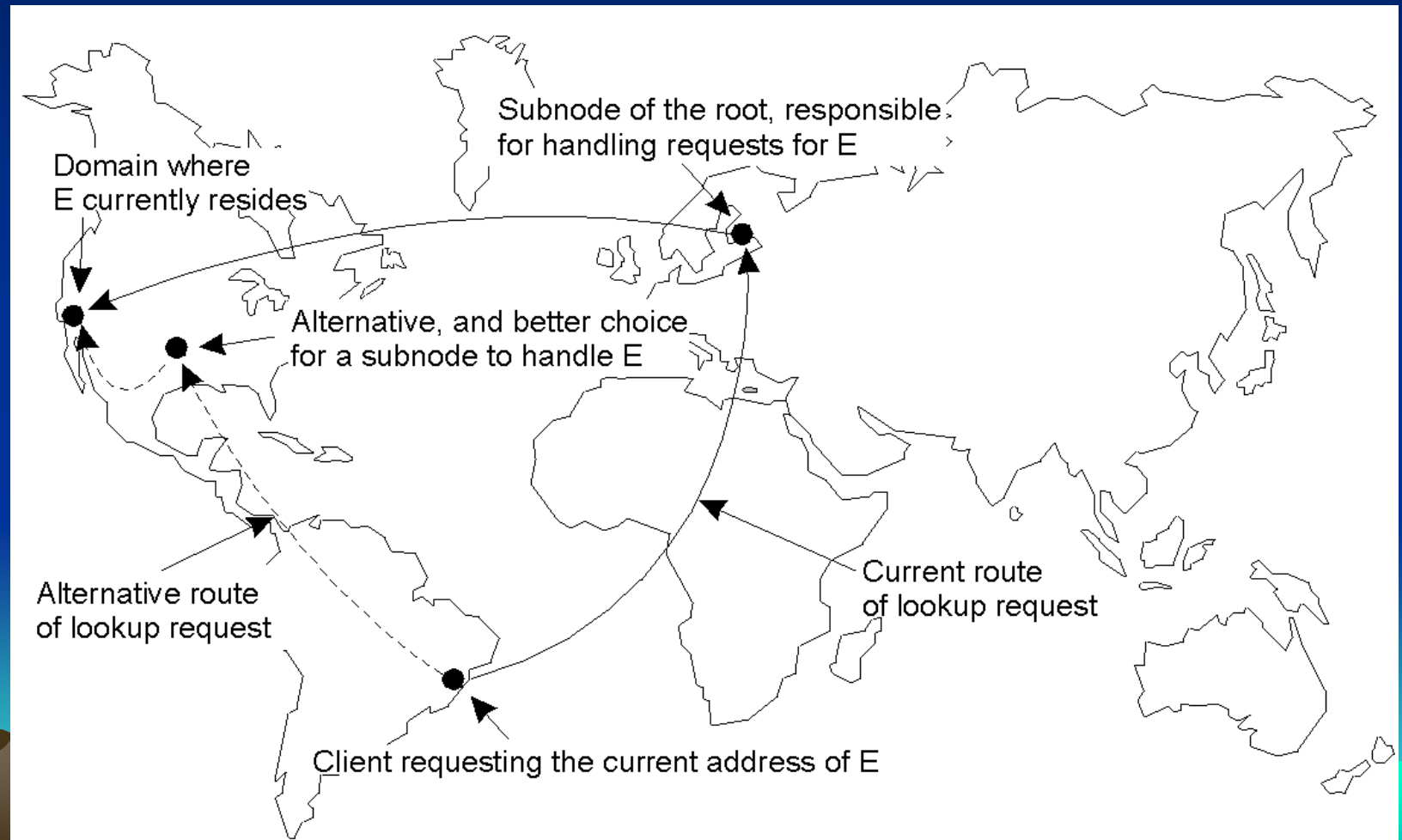
可扩展性问题(1)

- 分层定位服务存在的一个主要问题是根节点存储所有实体的位置并为每个实体处理请求，容易成为瓶颈
- 解决办法：将根节点和其他高层目录节点划分为多个子节点
- 问题：这些子节点的物理放置
 - 集中放置
 - 均匀放置
 - 实体的创建地相邻的节点



可扩展性问题(2)

在定位服务覆盖的网络中均匀放置根节点的子节点所存在的可扩展性问题



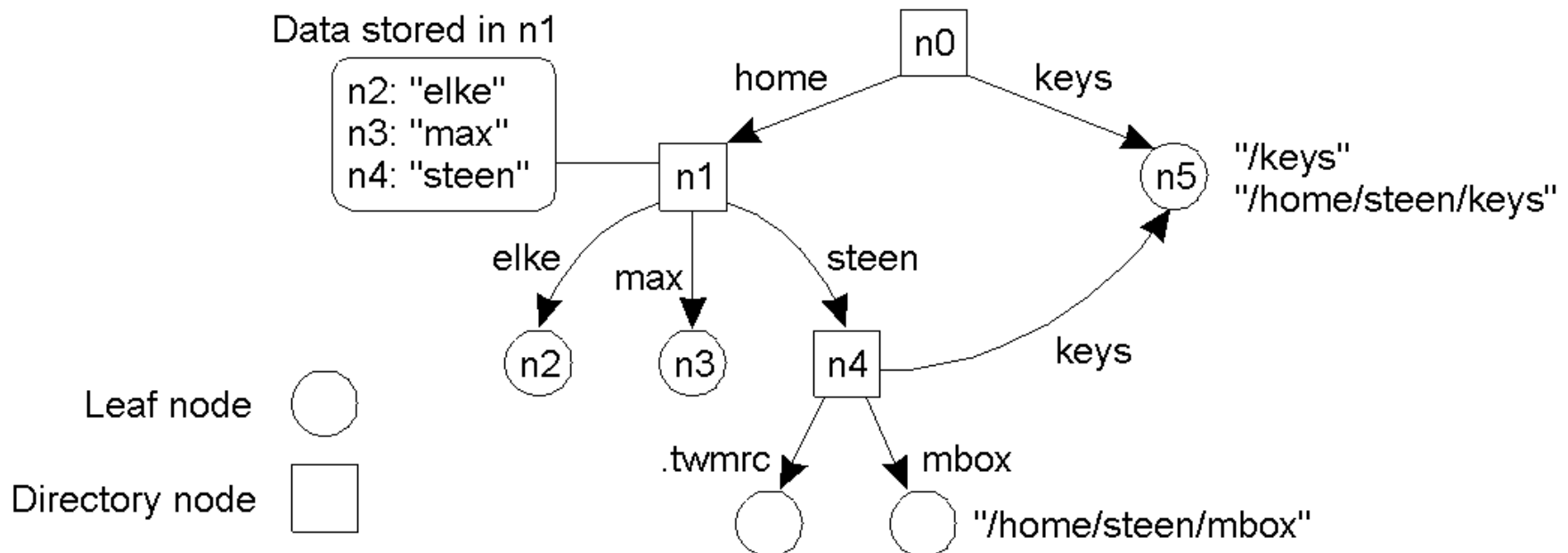
结构化命名

- 名称空间
- 名称解析
- 名称空间的实现



名称空间 (1)

- 名称组织在名称空间 (Name Space) 中
- 节点的有向图:
 - 根节点
 - 目录节点: 存储目录表 (<边标签, 节点标识符>对)
 - 叶子节点: 存储所表示实体的信息
 - 路径: 边标签序列, 如 “/home/steen/mbox”

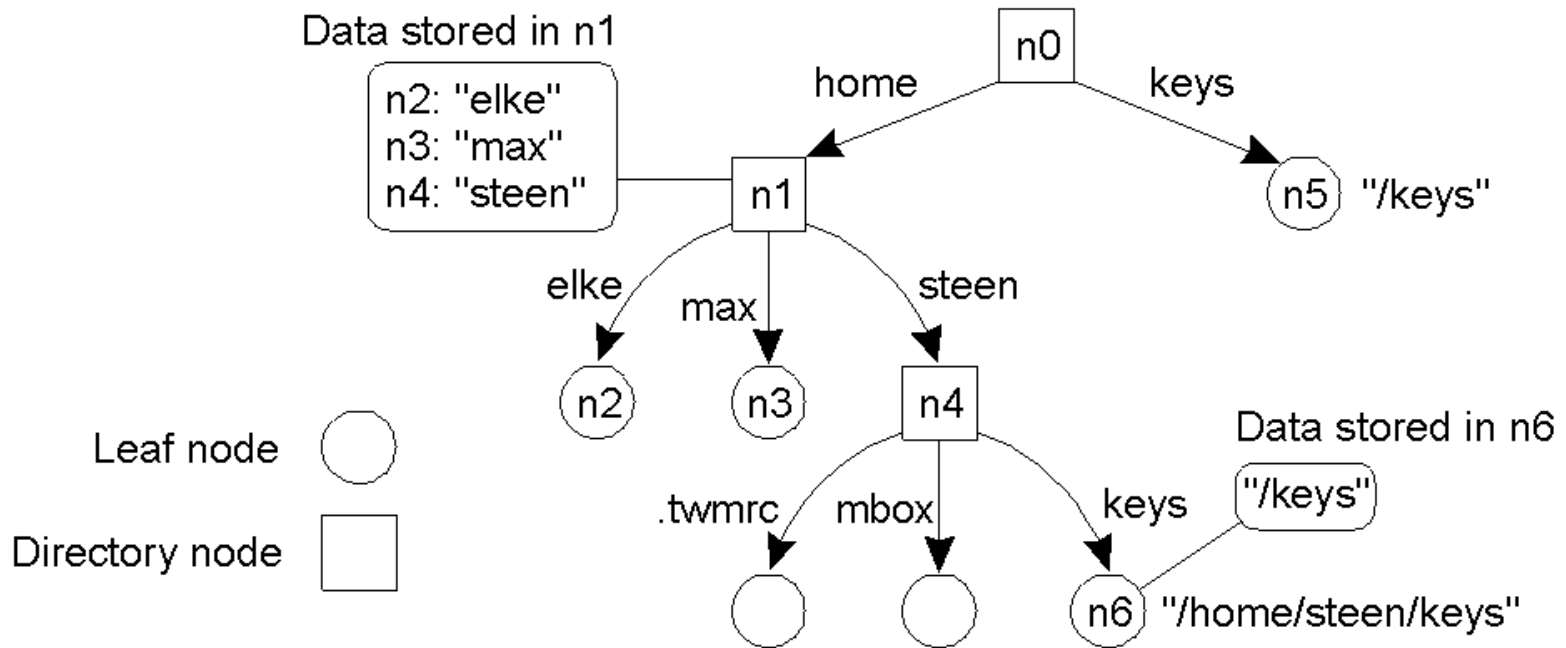


名称解析

- 定义:给定一个路径名，应该能够查找出存储在由该名称指向节点中的任何信息
- 别名：同一实体的另一名称
 - 硬链接：多个绝对路径名指向同一结点
 - 符号链接：节点存储绝对路径名



链接和挂载 (1)



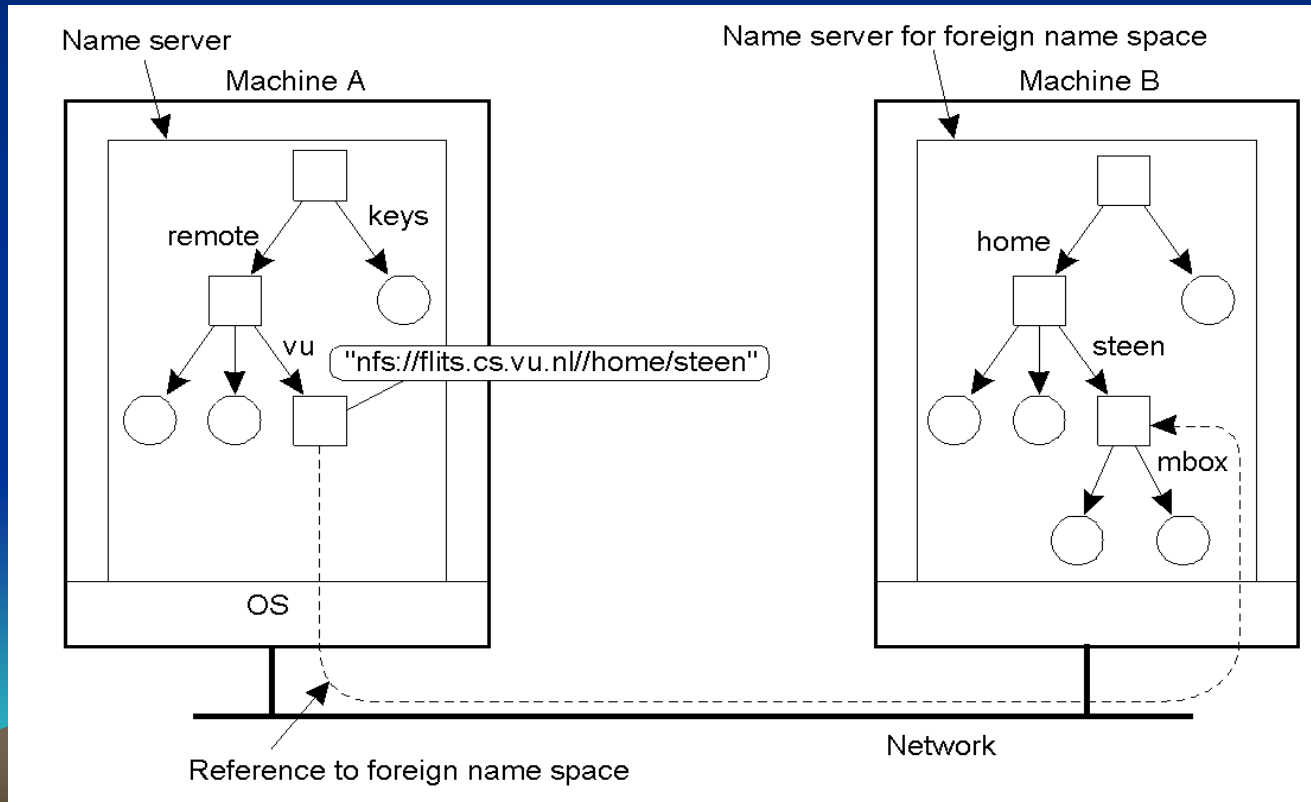
命名图中的符号链接的概念

链接和挂载 (2)

挂载：以透明的方式合并不同的名称空间；**挂载点（mount point）**和**挂载点（mounting point）**

需要的信息：访问协议的名称、服务器的名称和外部名称空间的挂载点的名称，可以表示成一个**URL**。

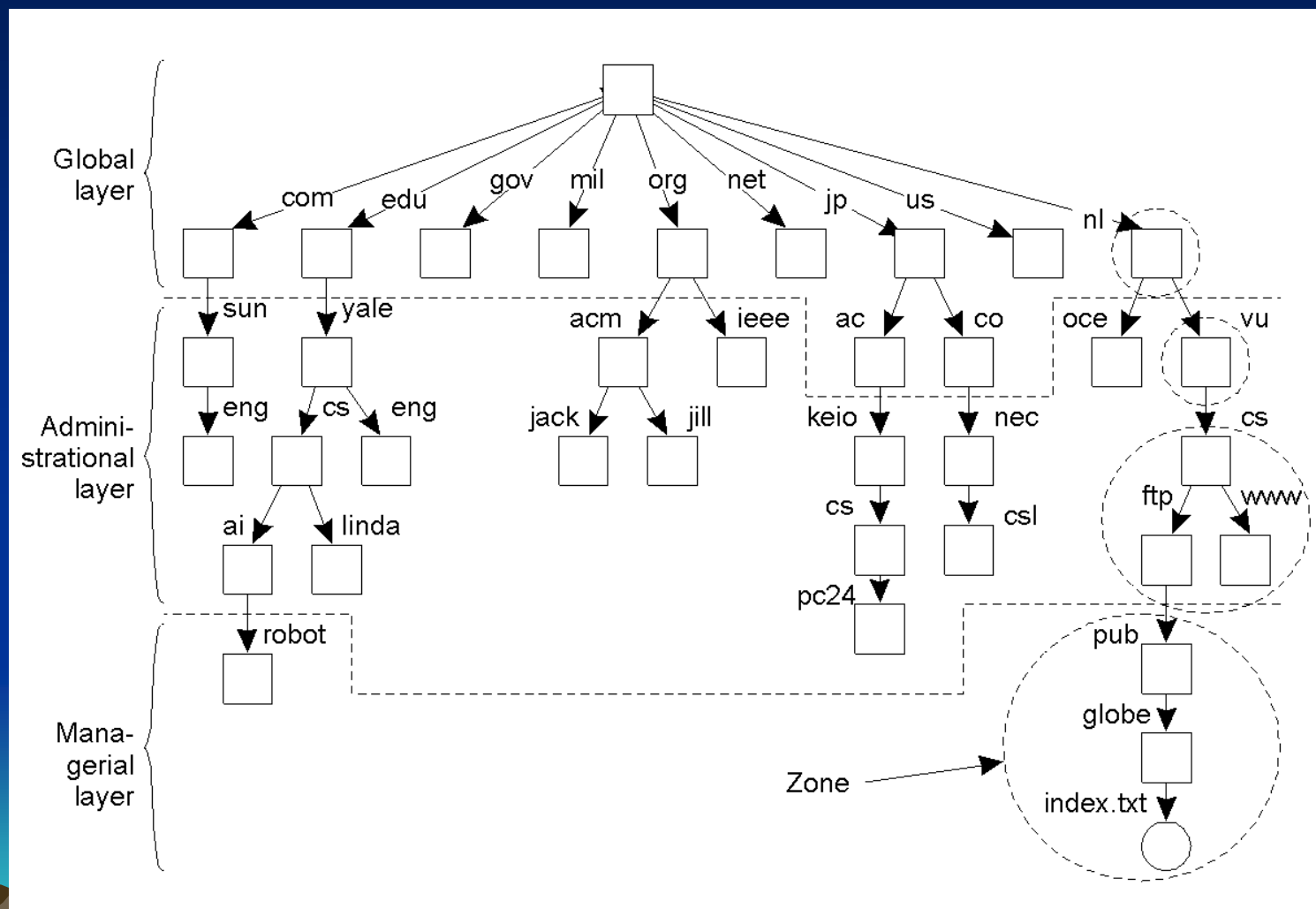
对/remote/vu/mbox的名称解析



通过指定的访问协议装配远程名称空间

名称空间的实现--名称空间划分(1)

命名服务是一种允许用户和进程添加、删除和查找名称的服务，由名称服务器实现



DNS name space 划分示例

名称空间划分(2)

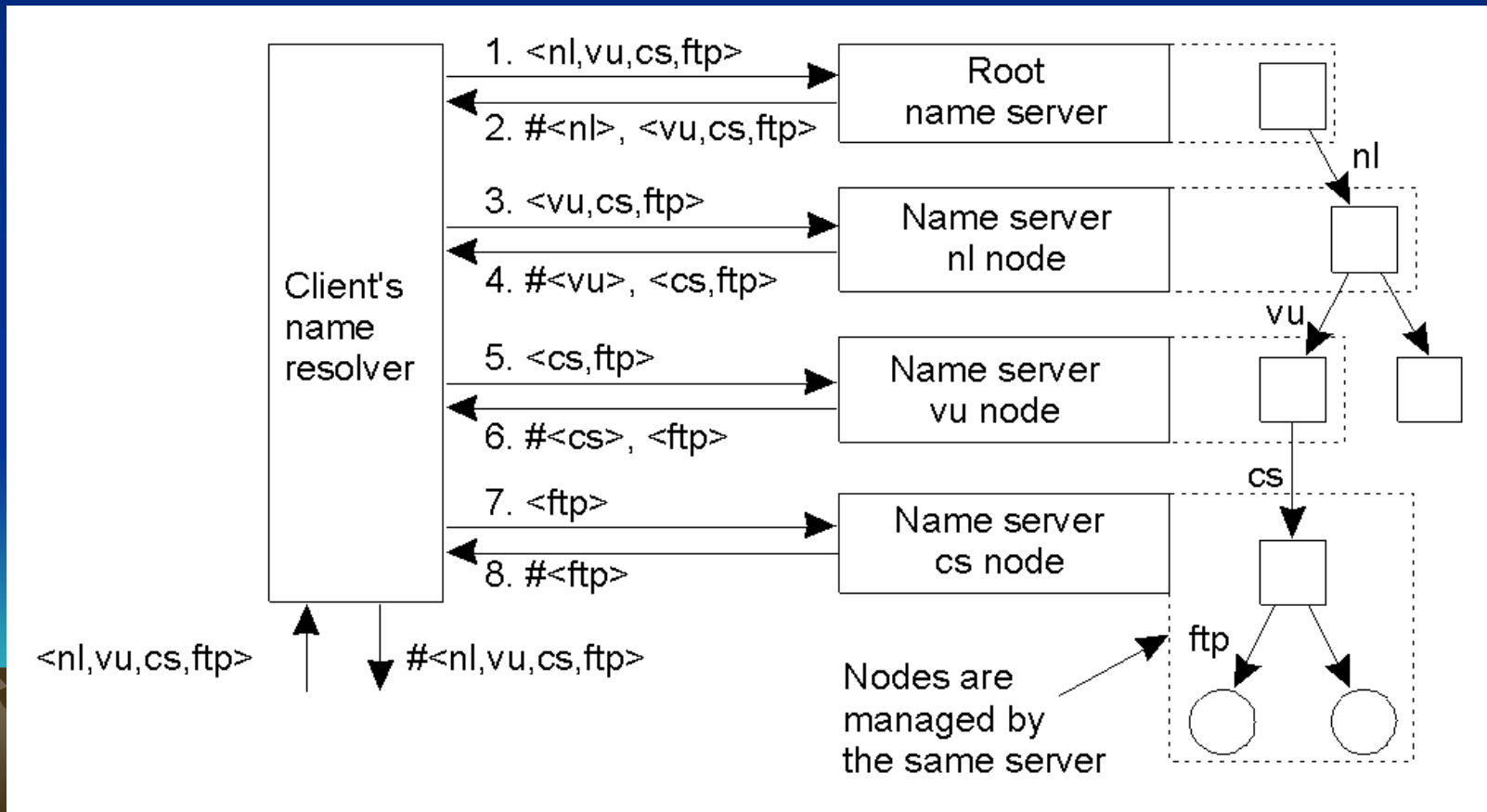
内容	全局层	行政层	管理层
网络的地理范围	世界范围	组织	部门
节点数目	少	许多	极多
查询响应	秒	毫秒	立即
更新的传播情况	延迟	立即	立即
复制数目	许多	没有或很少	没有
是否采用客户端缓存	是	是	有时

为全局层、行政层、管理层实现节点的名称服务器之间的比较



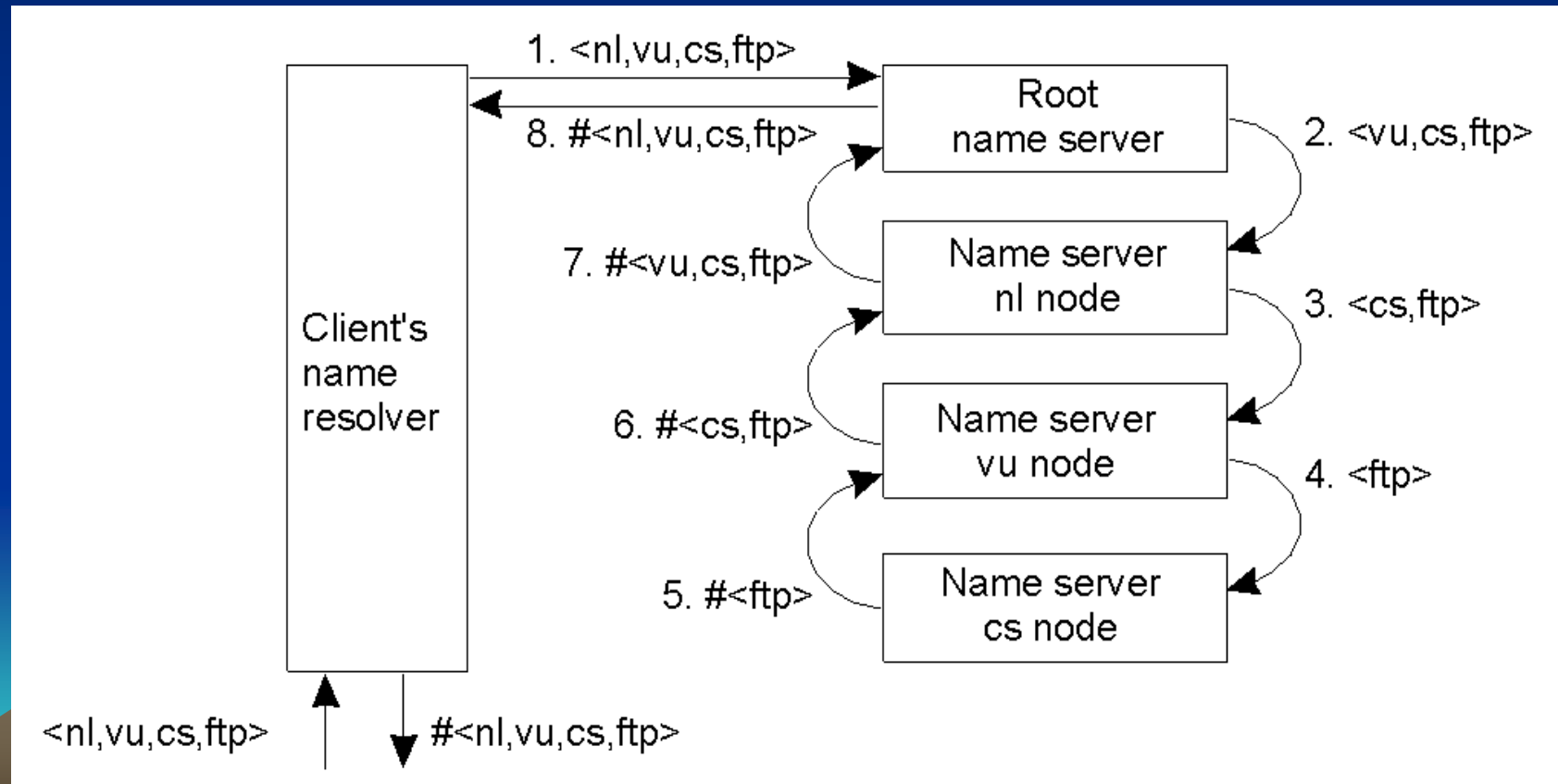
名称解析实现(1)

- 迭代名称解析原理 (iterative name resolution)
- 解析root:<nl,vu,cs,ftp,pub,globe,index.txt>,对应于
<ftp://ftp.cs.vu.nl/pub/globe/index.txt>
- #<nl>用来指明一台服务器的地址, 该服务器负责处理<nl>涉及到的节点
- 最后, 客户端会与该FTP服务器联系, 后者发送会被请求的文件



名称解析实现(2)

- 递归名称解析原理 **recursive name resolution**.
- 解析root:<nl,cs,pub,globe,intex.txt>,对应于ftp.cs.vu.nl/pub/globe/index.txt
- 缺点：要求名称服务器有较高性能
- 优点：缓存效果更有效；减少通信开销

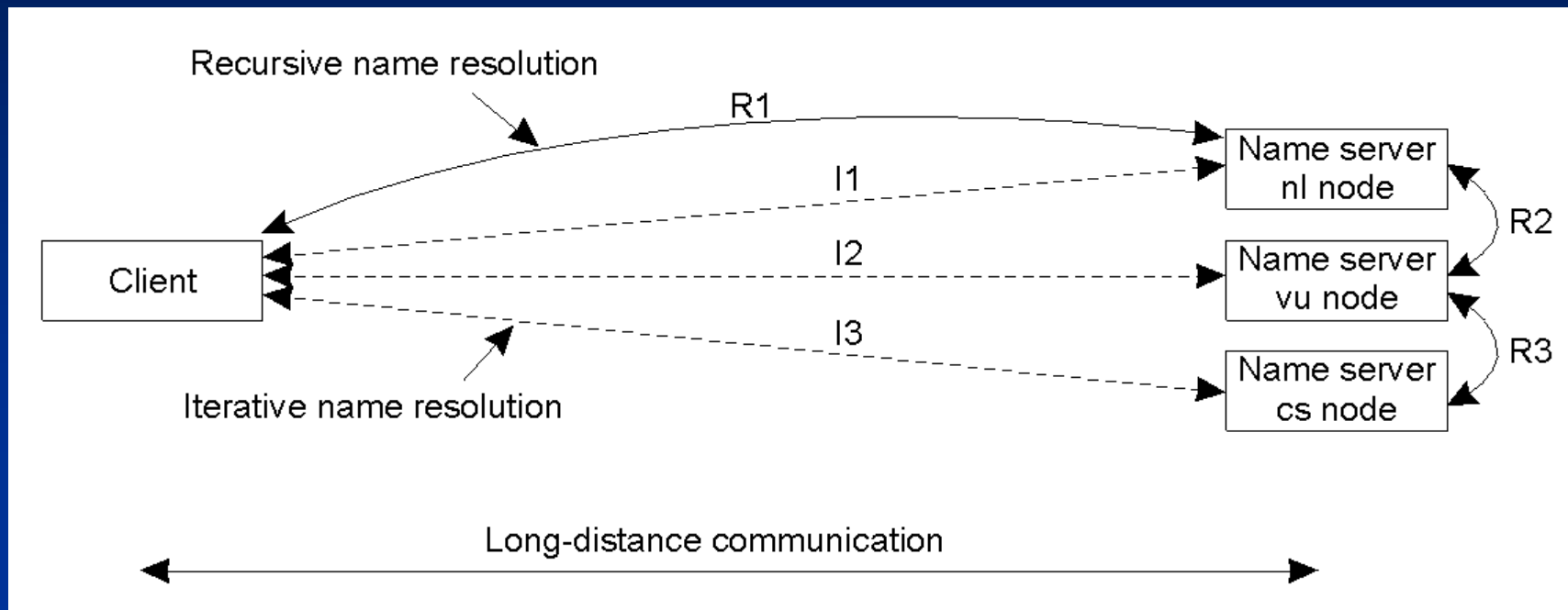


名称解析实现(3)

服务器所在节点	需要解析的标识符	查询	传递给下一个服务器	递归和缓存	返回请求者
cs	<ftp>	#<ftp>	--	--	#<ftp>
vu	<cs,ftp>	#<cs>	<ftp>	#<ftp>	#<cs> #<cs, ftp>
nl	<vu,cs,ftp>	#<vu>	<cs,ftp>	#<cs> #<cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>
root	<nl,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

对 *<nl, vu, cs, ftp>* 进行递归名称解析，名称服务器缓存用于后续查询的中间结果

名称解析实现(4)



迭代名称解析与递归名称解析通信代价比较

小结

- 名称、标识符和地址
- 无层次命名
 - 简单方法
 - 基于起始位置的方法
 - 分层方法
- 结构化命名
 - 名称空间
 - 名称解析
 - 名称空间的实现



习 题

- 假设某个移动实体几乎从不会离开域D，即使离开也很快返回。如何利用该信息在分层定位服务中加快查询操作的速度？
- 假设一个实体从位置A转移到位置B，期间经过了几个中间位置（停留时间都很短），最终到达B。在分层定位服务中更改地址可能花费较长时间，因此在经过中间位置时，应避免更改地址，那么经过中间位置时应该如何查找该实体？
- 在深度为k的分层定位服务中，当移动实体改变它的位置时，最少需要更新多少条位置记录？最多多少条位置记录？