

MC714

Sistemas Distribuídos

2º semestre, 2014

Nomeação

Nomeação

- Compartilhar recursos, identificar entidades de maneira única, fazer referência a localizações...
- Resolução de nomes
- Espaço de nomes e implementação

Nomeação

- Nome: cadeia de bits ou caracteres usada para referenciar uma entidade (recurso, hospedeiro, impressora, disco, arquivo ...)
- Acessar uma entidade: via **ponto de acesso**
 - Ponto de acesso: entidade nomeada por um **endereço**
 - Entidade pode ter mais de um ponto de acesso
- Nome: mais fácil de lembrar que endereço, mais flexível, independente de localização.

Nomeação

- **Identificador** → nome que possui as seguintes propriedades:
 1. 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.
- Como traduzir nomes e identificadores para endereços (pontos de acesso)?
 - Relação com roteamento/como encontrar ponto de acesso.

Nomeação

- Nomeação simples
 - Soluções simples (broadcast e multicast)
 - Soluções baseadas em localização nativa (home)
 - Tabelas hash distribuídas (DHT)
 - Abordagens hierárquicas
- Nomeação estruturada
- Nomeação baseada em atributos

Nomeação Simples

- Nomes simples: não contêm nenhuma informação sobre como localizar o ponto de acesso de sua entidade associada.
- Solução 1: broadcasting
 - Não é escalável; útil em redes locais.
 - Todos devem estar esperando requisições de nome.
 - Ex.: ARP – Address Resolution Protocol – IP → MAC address.
 - Alternativa: multicasting

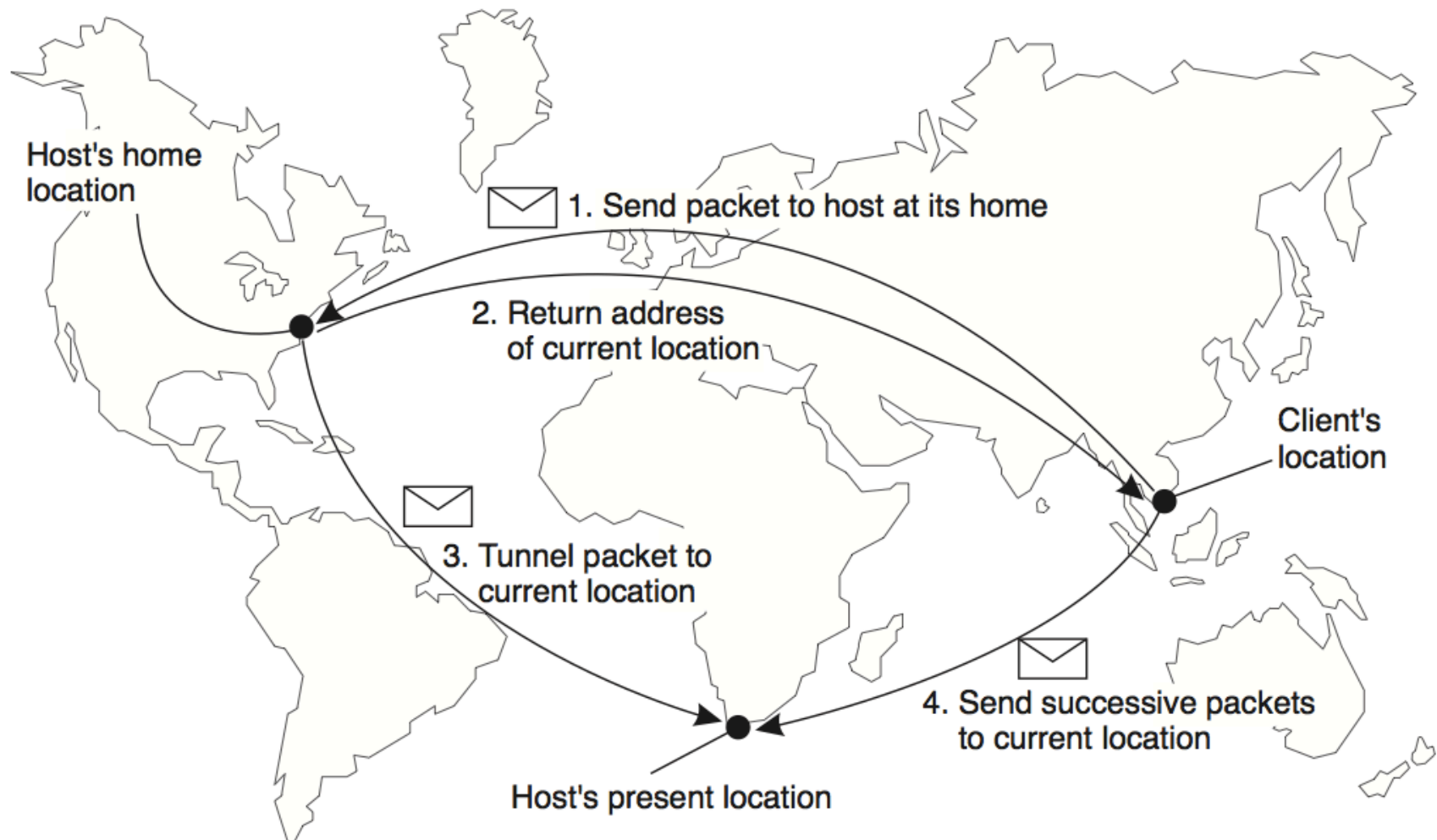
Nomeação Simples

- Solução 2: ponteiros repassadores
 - Entidade se move de A para B → deixa em A uma referência para B.
 - Simples e pode ser transparente
 - Cadeia pode tornar-se longa demais → latência
 - Cadeias podem sofrer com falhas → quanto mais longa, pior
- Ex.: cadeias SSP (stub scion pairs)
 - Ponteiro repassador: par (apêndice de cliente, apêndice de servidor)
 - Cada apêndice de servidor referencia objeto local ou um apêndice de cliente remoto do objeto.
 - Fig. 80.

Localização nativa

- Entidade possui um agente nativo (home address).
- Requisições dirigidas ao agente nativo.
- Home address pode ser registrado em serviço de nomes.
- Agente nativo encaminha requisição ao endereço atual da entidade e informa requisitante da localização.

Localização nativa



Tabelas Hash Distribuídas (DHTs)

- Chord
- Identificador de m bits designado aleatoriamente
- Entidade com chave k sob jurisdição de nó com nó com $\text{id} \geq k$ ($\text{succ}(k)$).
- Questão: resolver com eficiência chave k para endereço $\text{succ}(k)$.
- Abordagem 1:
 - Cada nó p monitora $\text{succ}(p+1)$ e $\text{pred}(p)$.
 - Nó p repassa requisição de chave k para vizinho mais adequado, exceto se $\text{pred}(p) < k \leq p$

Tabelas Hash Distribuídas (DHTs)

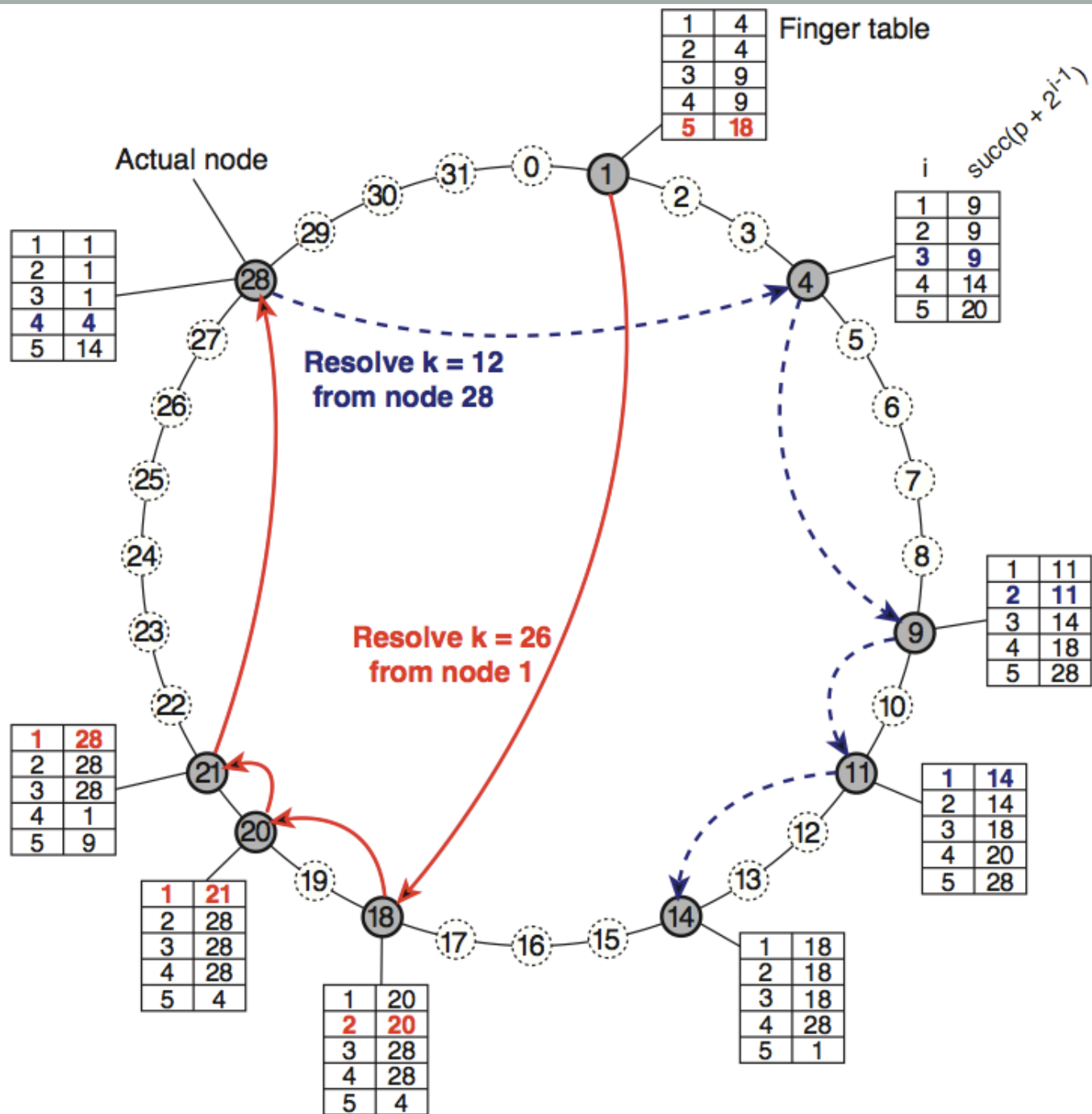
- Abordagem 2: tabela de derivação (finger table).
 - Cada nó mantém m entradas de tal forma que a i -ésima entrada aponta para o primeiro nó que sucede p por no mínimo 2^{i-1} .

$$FT_p[i] = succ(p + 2^{i-1})$$

- Para consultar chave k , repassa para nó q com índice j :

$$q = FT_p[j] \leq k \leq FT_p[j + 1]$$

- Finger tables devem ser mantidas atualizadas
 - $q == pred(succ(q+1))$.
 - para todo i , achar $succ(q+2^{i-1})$.



DHT - Exploração de proximidade

- Requisições podem ser roteadas erraticamente, pois overlay não considera localização.
 - Levar em conta rede subjacente
- Identificadores baseados na topologia
- Roteamento por proximidade
 - Faixa de opções em cada entrada da finger table
- Seleção de vizinhos por proximidade
 - Similaridade com roteamento por proximidade

Abordagens hierárquicas – visão geral

- Rede dividida em conjunto de domínios
 - Podem ser divididos em subdomínios
- Domínio mais alto: abrange a rede toda
- Domínio-folha: mais baixo na rede
- $\text{Dir}(D)$: nó diretório que monitora entidades dentro de um domínio D .
- $\text{Dir}(D)$, com D sendo domínio mais alto: nó (de diretório) raiz
- Fig. 81

Abordagens hierárquicas – visão geral

- Cada entidade tem registro de localização no nó de diretório do domínio a que pertence.
 - Registro de localização para entidade E no nó de diretório N para um domínio-folha D contém endereço corrente de E em D.
 - Em um nível mais alto: nó de diretório N', domínio D' (que contém D) contém registro de localização de E como um ponteiro para N.
 - Nó raiz: registro de localização para cada entidade com ponteiro para nó de diretório do próximo subdomínio onde entidade está localizada.

Abordagens hierárquicas – visão geral

- Entidade pode ter vários endereços
 - Entidade replicada
 - Endereços no domínio folha D1 e D2
 - Nó de diretório que contém D1 e D2: dois ponteiros para entidade
- Fig 82.

Abordagens hierárquicas – consulta

- Cliente C deseja localizar E
 - Emite requisição de consulta ao nó de diretório do domínio folha D , onde C reside.
 - D tem registro de E ?
 - Se sim, E está no domínio de D . Pode responder.
 - Se não, repassa requisição ao seu pai D' .
 - D' tem registro de E ?
 - ...
 - Chega a M , que retorna registro de $E \rightarrow E$ está em $\text{dom}(M)$.
 - Registro do nó folha sob M tem endereço de E . Retorna.
- Fig 83.

Abordagens hierárquicas – atualização

- Entidade E cria réplica em domínio folha D.
- Insere seu endereço em $\text{dir}(D)$.
- $\text{dir}(d)$ repassa ao nó pai D' .
- D' repassa ao nó pai
- Até chegar a nó M que já tem registro de outra réplica de E.
- Fig. 84
- Remoção: análoga à inserção.
 - Sobe até encontrar registro para E que fique não vazio após a remoção, ou até a raiz.

Nomeação estruturada

- Nomes simples são bons para máquinas, mas não são convenientes para seres humanos.
- Nomeação estruturada: composição de nomes simples.
 - Arquivos, hospedeiros na Internet
- Espaço de nomes
 - Grafo dirigido rotulado, com dois tipos de nós: nós folha e nós de diretório

Nomeação estruturada

- Nó folha: entidade nomeada, sem saídas.
- Nó de diretório: vários ramos de saída, rotulados.
 - Armazena tabela de entradas (rótulo do ramo, identificador do nó)
- Nó raiz: somente saídas
- Fig. 85
- Caminhos referenciados pela sequência de rótulos
- $N: \langle \text{label-1}, \text{label-2}, \dots, \text{label-n} \rangle \rightarrow \text{nome de caminho}$
- Caminho absoluto: 1º nó do nome é raiz
- C.C.: caminho relativo

Resolução de nomes

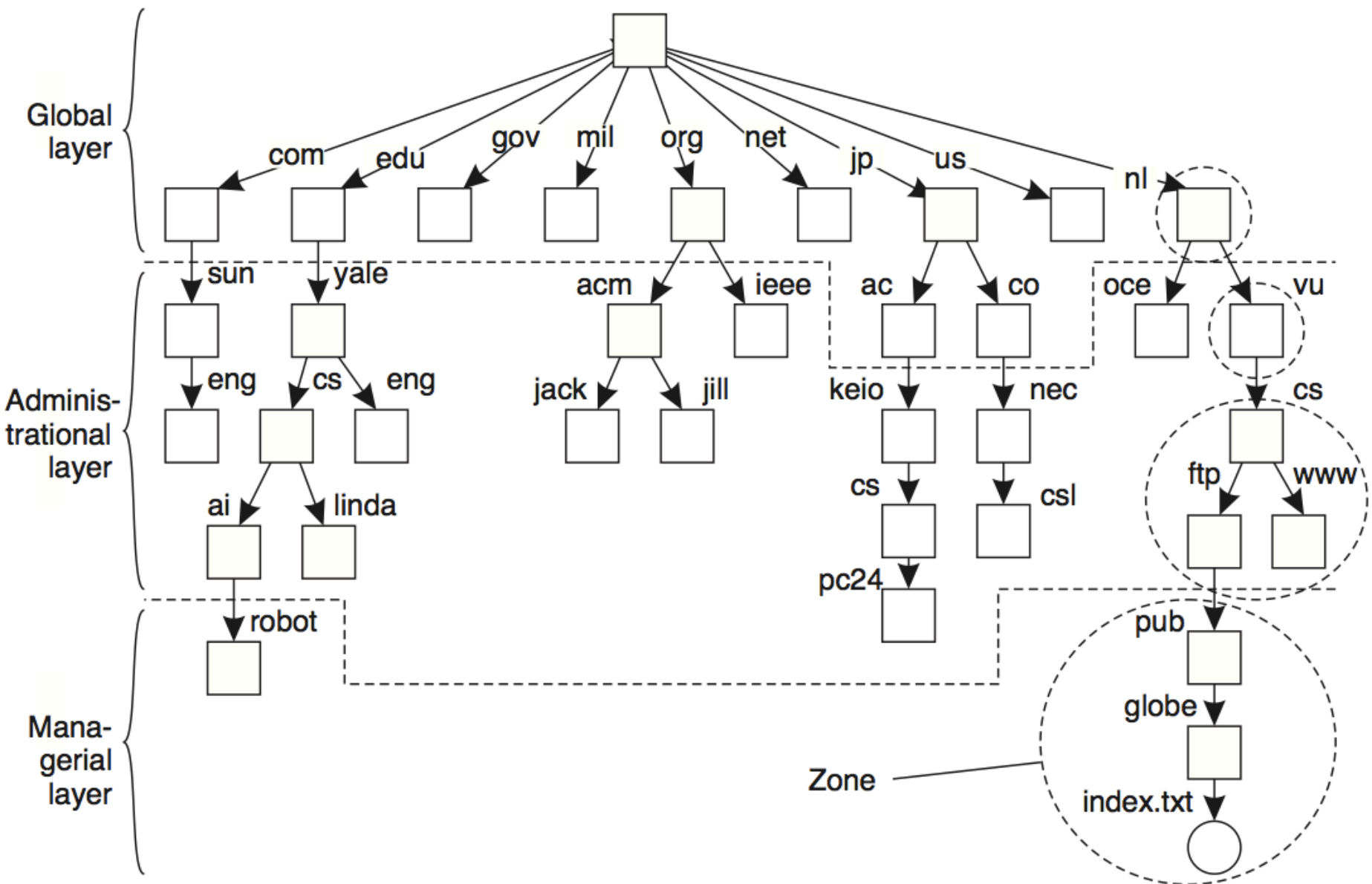
- Processo de busca de um nome: resolução de nome
- Mecanismo de fechamento: seleção do nó inicial de um espaço de nomes onde resolução começa.
- Apelidos (aliases)
 - Ponteiros estritos
 - Ponteiros simbólicos
- Resolução de nomes pode ocorrer em mais de um espaço de nomes
 - Sistema de arquivo montado
 - Espaço de nomes externo (ex.: NFS).

Espaço de nomes - implementação

- SDs: distribuir implementação do espaço de nomes por vários servidores de nomes.
 - Distribuir nós do grafo de nomeação.
- Grande escala → comum em hierarquia
 - Camada global
 - Camada administrativa
 - Camada gerencial

Espaço de nomes - implementação

- Camada global
 - Nós de nível mais alto (raiz e nós próximos)
 - Mais estáveis → tabelas de diretório raramente mudam
- Camada administrativa
 - Nós de diretório gerenciados por uma única organização
 - Representam grupos de entidades que pertencem à mesma organização ou unidade administrativa
 - Mudanças com maior frequência que na camada global
- Camada gerencial
 - Mudança periódica
 - Nós mantidos por administradores e usuários



Espaço de nomes - implementação

- Camada global

- Alta disponibilidade: se falha, grande parte do espaço fica inalcançável.
- Desempenho: Baixa taxa de mudança; Cache local é útil. Não precisam responder tão rapidamente.
- Replicação pode ser aplicada.

- Camada administrativa

- Se falhar, muitos recursos dentro da organização podem ficar inalcançáveis.
- Deve responder mais rapidamente que camada global.

- Camada gerencial

- Indisponibilidade temporária afeta poucos usuários
- Desempenho é crucial;
- Muda com frequencia → cache pode não funcionar muito bem.

Espaço de nomes - implementação

Item	Global	Administrational	Managerial
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

Figure 5-14. A comparison between name servers for implementing nodes from a large-scale name space partitioned into a global layer, an administrative layer, and a managerial layer.

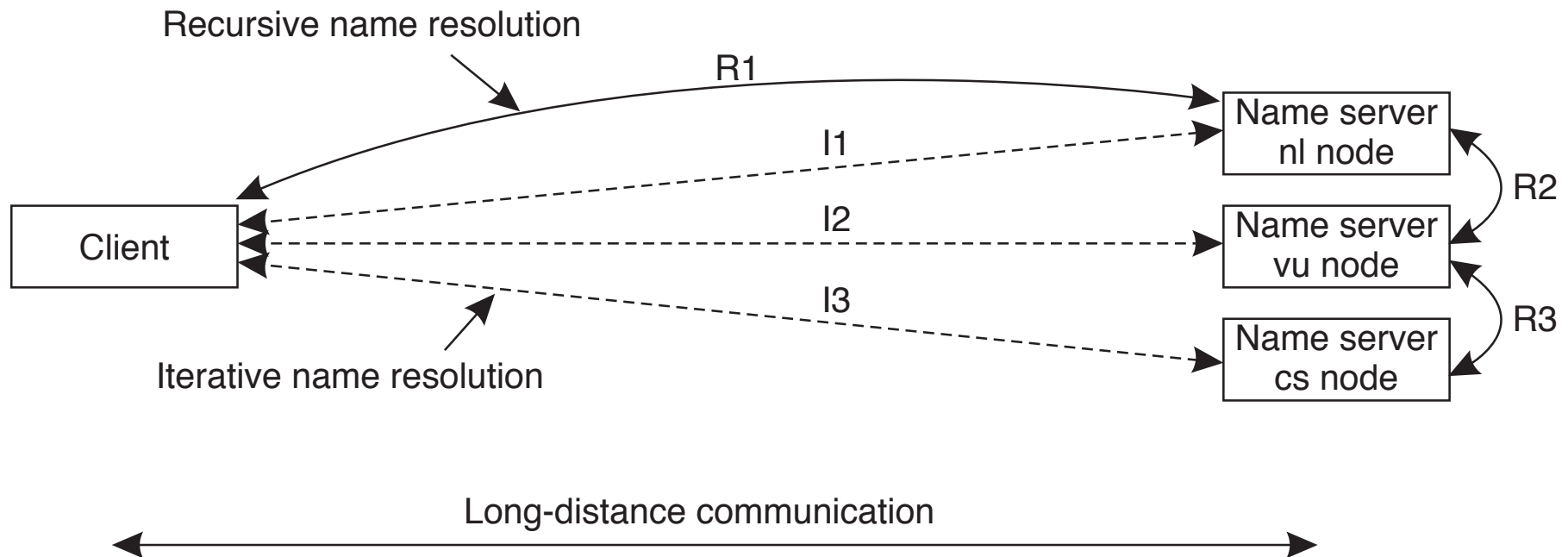
Resolução de nomes - implementação

- Resolução iterativa
 - Resolvedor entrega nome completo ao servidor-raiz
 - Servidor resolve nome até onde conhece e retorna endereço do servidor de nome associado
 - Resolvedor de nome do cliente entra em contato com servidor retornado
 - ...
 - Fig. 86

Resolução de nomes - implementação

- Resolução recursiva
 - Ao invés de retornar resultado intermediário, o próprio servidor consultado realiza consulta ao próximo nível.
 - Carga maior aos servidores de nomes.
 - Em geral servidores na camada global suportam somente resolução iterativa.
 - Cache mais eficiente que na iterativa.
 - Pode reduzir custo de comunicação.
 - Fig. 87

Resolução de nomes - implementação



Comunicação iterativa versus recursiva.

Resolução recursiva e cache

Servidor para nós	Deve resolver	Busca	Passa para filho	Recebe faz cache	Responde para cliente
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>

Exemplo - DNS

- Domain Name System – DNS
- Resolve endereços IP a partir de nomes na Internet
- Espaço de nomes hierárquico: listagem de rótulos separados por pontos.

Name	Record type	Record value
cs.vu.nl	SOA	star (1999121502,7200,3600,2419200,86400)
cs.vu.nl	NS	star.cs.vu.nl
cs.vu.nl	NS	top.cs.vu.nl
cs.vu.nl	NS	solo.cs.vu.nl
cs.vu.nl	TXT	"Vrije Universiteit - Math. & Comp. Sc."
cs.vu.nl	MX	1 zephyr.cs.vu.nl
cs.vu.nl	MX	2 tornado.cs.vu.nl
cs.vu.nl	MX	3 star.cs.vu.nl
star.cs.vu.nl	HINFO	Sun Unix
star.cs.vu.nl	MX	1 star.cs.vu.nl
star.cs.vu.nl	MX	10 zephyr.cs.vu.nl
star.cs.vu.nl	A	130.37.24.6
star.cs.vu.nl	A	192.31.231.42
zephyr.cs.vu.nl	HINFO	Sun Unix
zephyr.cs.vu.nl	MX	1 zephyr.cs.vu.nl
zephyr.cs.vu.nl	MX	2 tornado.cs.vu.nl
zephyr.cs.vu.nl	A	192.31.231.66
www.cs.vu.nl	CNAME	soling.cs.vu.nl
ftp.cs.vu.nl	CNAME	soling.cs.vu.nl
soling.cs.vu.nl	HINFO	Sun Unix
soling.cs.vu.nl	MX	1 soling.cs.vu.nl
soling.cs.vu.nl	MX	10 zephyr.cs.vu.nl
soling.cs.vu.nl	A	130.37.24.11
laser.cs.vu.nl	HINFO	PC MS-DOS
laser.cs.vu.nl	A	130.37.30.32
vucs-das.cs.vu.nl	PTR	0.26.37.130.in-addr.arpa
vucs-das.cs.vu.nl	A	130.37.26.0

Nomeação baseada em atributos

- Entidade tem conjunto associado de atributos
- Podem ser usados para buscar entidades
- “Páginas amarelas”
- Também chamados “serviços de diretório”
- Atributos podem ser descritos de forma diferente por pessoas diferentes
 - Padronização de descrição de atributos.
 - Estrutura de descrição de recurso – resource description framework – RDF
 - Sujeito, predicado, objeto – (Pessoa, nome, Alice)