Elosztott rendszerek: Alapelvek és paradigmák
Distributed Systems: Principles and Paradigms

Maarten van Steen[1] Kitlei Róbert [2]

[1]VU Amsterdam, Dept. Computer Science
[2]ELTE Informatikai Kar

5. rész: Elnevezési rendszerek

2015. május 24.

Tartalomjegyzék

Elnevezési rendszerek

Elnevezési rendszerek

Az elosztott rendszerek entitásai a kapcsolódási pontjaikon (access point) keresztül érhetőek el. Ezeket távolról a címük azonosítja, amely megnevezi az adott pontot.
Célszerű lehet az entitást a kapcsolódási pontjaitól függetlenül is elnevezni. Az ilyen nevek helyfüggetlenek (location independent).
Az egyszerű neveknek nincsen szerkezete, tartalmuk véletlen szöveg.
Az egyszerű nevek csak összehasonlításra használhatóak.

Azonosító

Egy név azonosító, ha egy-egy kapcsolatban áll a megnevezett egyeddel, és ez a hozzárendelés maradandó, azaz a név nem hivatkozhat más egyedre később sem.

Strukturálatlan nevek

Strukturálatlan nevek feloldása

Milyen lehetőségek vannak strukturálatlan nevek feloldására? (Azaz: hogyan találjuk meg a hozzárendelt kapcsolódási pontot?)

- egyszerű megoldások (broadcasting)
- otthonalapú megoldások
- elosztott hasítótáblák (strukturált P2P)
- hierarchikus rendszerek

Névfeloldás: egyszerű megoldások

Broadcasting

Kihirdetjük az azonosítót a hálózaton; az egyed visszaküldi a jelenlegi címét.

- Lokális hálózatokon túl nem skálázódik
- A hálózaton minden gépnek figyelnie kell a beérkező kérésre

Továbbítómutató

Amikor az egyed elköltözik, egy mutató marad utána az új helyére.

- A kliens elől el van fedve, hogy a szoftver továbbítómutató-láncot old fel.
- A megtalált címet vissza lehet küldeni a klienshez, így a további feloldások gyorsabban mennek.
- Földrajzi skálázási problémák
  - A hosszú láncok nem hibatűrőek
  - A feloldás hosszú időbe telik
  - Külön mechanizmus szükséges a láncok rövidítésére

Otthonalapú megközelítések

Egyrétegű rendszer

Az egyedhez tartozik egy otthon, ez tartja számon az egyed jelenlegi címét.

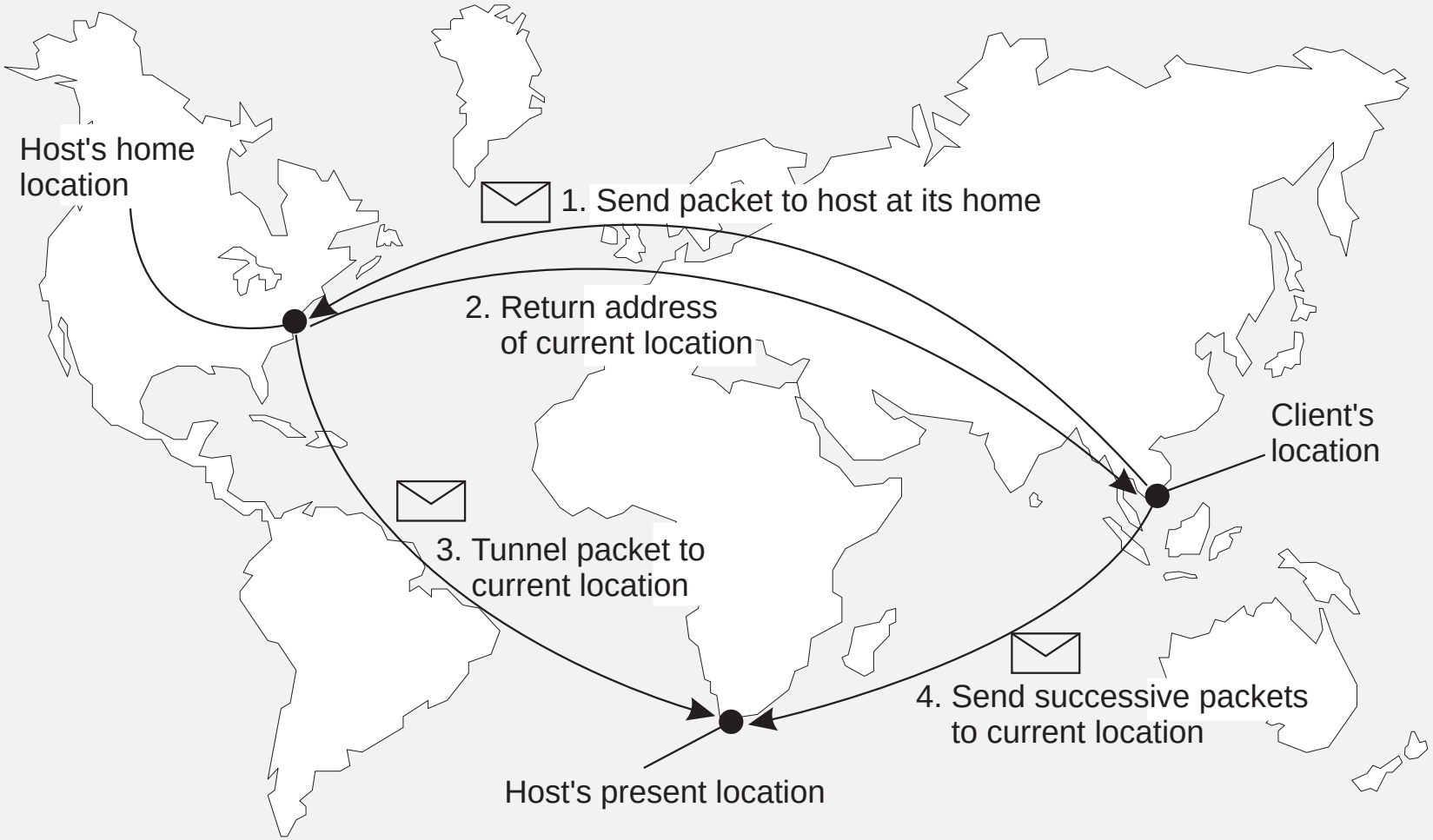
- Az egyed otthoni címe (home address) be van jegyezve egy névszolgáltatásba
- Az otthon számon tartja az egyed jelenlegi címét (foreign address)
- A kliens az otthonhoz kapcsolódik, onnan kapja meg az aktuális címet

Kétrétegű rendszer

Az egyes (pl. földrajzi alapon meghatározott) környékeken feljegyezzük, hogy melyik egyedek tartózkodnak éppen arrafelé.

- A névfeloldás először ezt a jegyzéket vizsgálja meg
- Ha az egyed nincsen a környéken, csak akkor kell az otthonhoz fordulni

Otthonalapú megközelítések



Otthonalapú megközelítések

Problémák

- Legalább az egyed élettartamán át fenn kell tartani az otthont
- Az otthon helye rögzített $\Rightarrow$ költséges lehet, ha az egyed messze költözik
- Rossz földrajzi skálázódás: az egyed sokkal közelebb lehet a klienshez az otthonnál

Eloszott hasítótábla

Chord eloszott hasítótábla

Elosztott hasítótáblát (distributed hash table, DHT) készítünk (konkrétan Chord protokoll szerintit), ebben csúcsok tárolnak egyedeket. Az N csúcs gyűrű overlay szerkezetbe van szervezve.

- Mindegyik csúcshoz véletlenszerűen hozzárendelünk egy m bites azonosítót, és mindegyik entitáshoz egy m bites kulcsot. (Tehát $N \leq 2^m$.)
- A k kulcsú egyed felelőse az az id azonosítójú csúcs, amelyre $k \leq id$, és nincsen köztük másik csúcs. A felelős csúcsot a kulcs rákövetkezőjének is szokás nevezni; jelölje $succ(k)$.

Rosszul méreteződő megoldás

A csúcsok eltárolhatnák a gyűrű következő csúcsának elérhetőségét, és így lineárisan végigkereshetnénk a gyűrűt. Ez $\mathcal{O}(N)$ hatékonyságú, rosszul skálázódik, nem hibatűrő...

DHT: Finger table

Chord alapú adattárolás

- Mindegyik p csúcs egy FT_p „finger table”-t tárol m bejegyzéssel:

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

  Bináris (jellegű) keresést szeretnénk elérni, ezért minden lépés felezi a keresési tartományt: $2^{m-1}, 2^{m-2}, \ldots, 2^0$.

- A k kulcsú egyed kikereséséhez (ha nem a jelenlegi csúcs tartalmazza) a kérést továbbítjuk a j indexű csúcshoz, amelyre

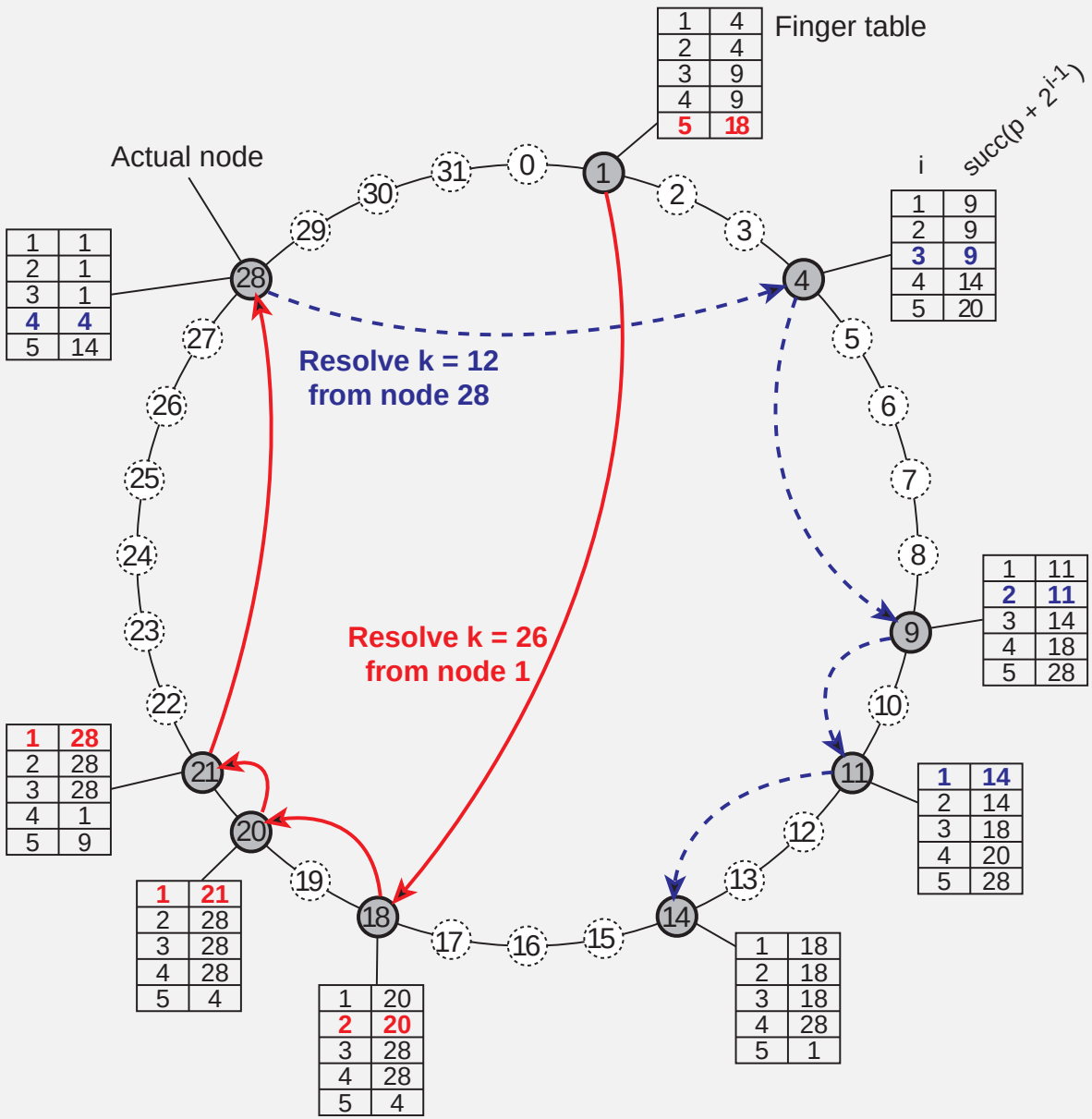
$$FT_p[j] \leq k < FT_p[j+1]$$

  illetve, ha $p < k < FT_p[1]$, akkor is $FT_p[1]$-hez irányítjuk a kérést.

Jól méreteződő megoldás

Ez a megoldás $\mathcal{O}(m)$, azaz $\mathcal{O}(log(N))$ hatékonyságú.

DHT: Finger table



A hálózati közelség kihasználása

Probléma

Mivel overlay hálózatot használunk, az üzenetek sokat utazhatnak két csúcs között: a k és a $succ(k+1)$ csúcs messze lehetnek egymástól.

Azonosító topológia szerinti megválasztása: A csúcsok azonosítóját megpróbálhatjuk topológiailag közeli csúcsokhoz közelinek választani. Ez nehéz feladat lehet.

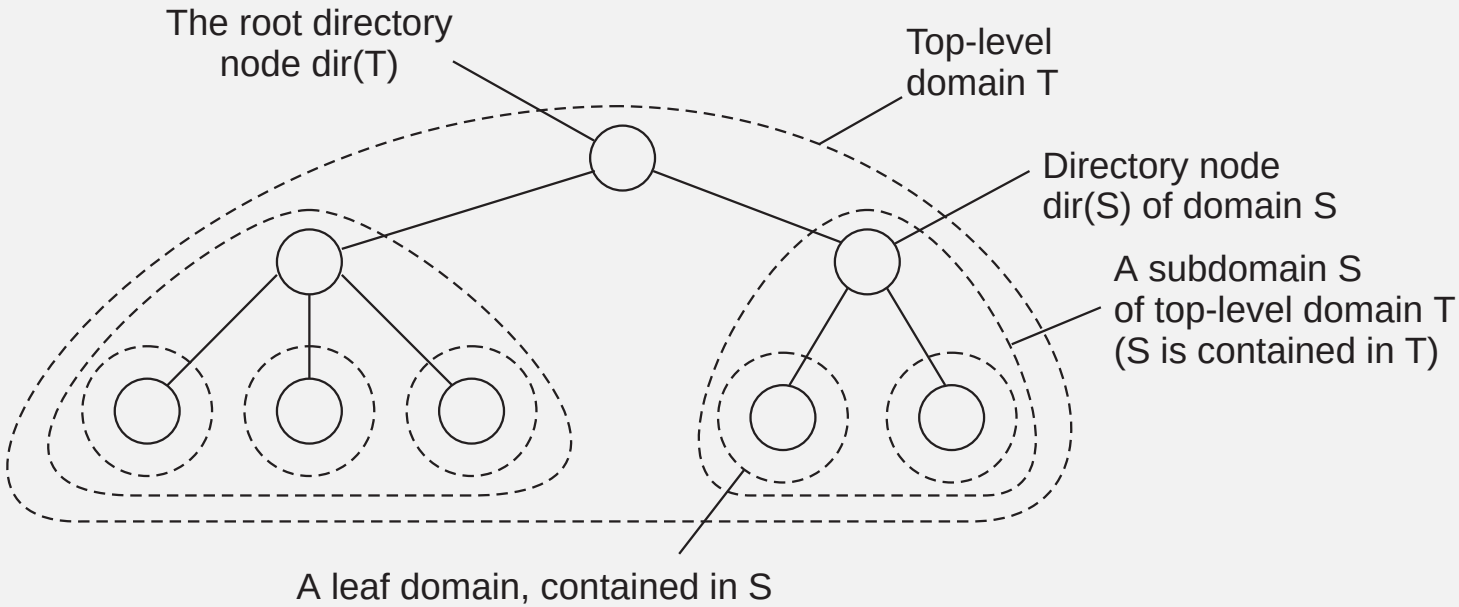
Közelség szerinti útválasztás: A p csúcs FT_p táblája m elemet tartalmaz. Ha ennél több információt is eltárolunk p-ben, akkor egy lépés megtételével közelebb juthatunk a célcsúcshoz.

Szomszéd közelség szerinti megválasztása: Ha a Chordtól eltérő ábrázolást követünk, a csúcs szomszédainak megválasztásánál azok közelségét is figyelembe lehet venni.

Hierarchikus módszerek

Hierarchical Location Services (HLS)

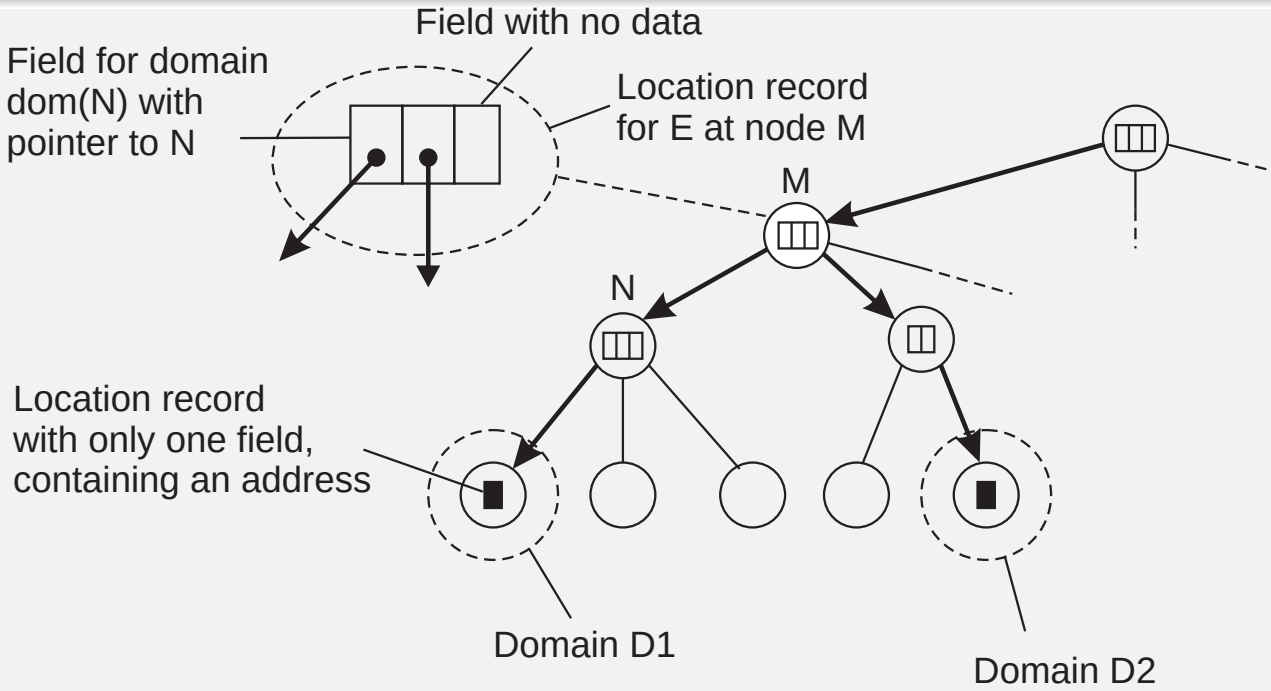
A hálózatot osszuk fel tartományokra, és mindegyik tartományhoz tartozzon egy katalógus. Építsünk hierarchiát a katalógusokból.



HLS: Katalógus-csúcsok

A csúcsokban tárolt adatok

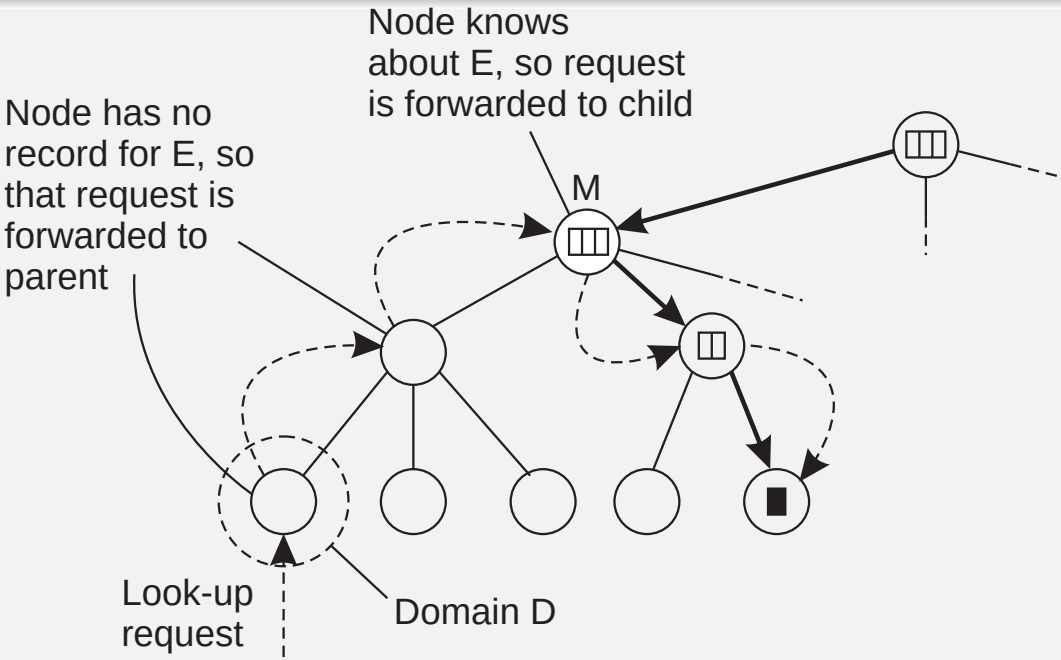
- Az *E* egyed címe egy levélben található meg
- A gyökértől az *E* leveléig vezető úton minden belső csúcsban van egy mutató a lefelé következő csúcsra az úton
- Mivel a gyökér minden út kiindulópontja, minden egyedről van információja



HLS: Keresés a fában

Keresés a fában

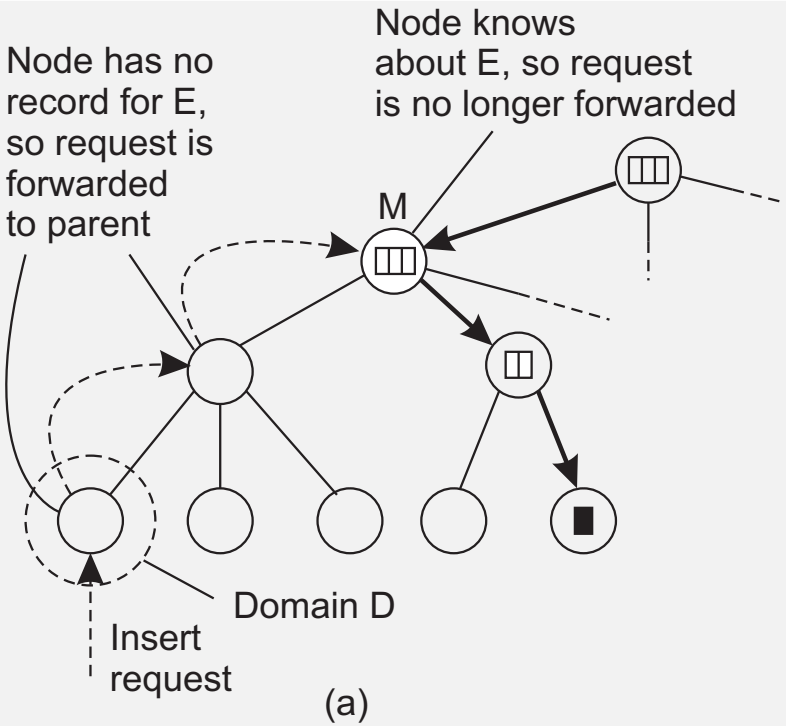
- A kliens valamelyik tartományba tartozik, innen indul a keresés
- Felmegyünk a fában addig, amíg olyan csúcshoz nem érünk, amelyik tud *E*-ről, aztán követjük a mutatókat a levélig, ahol megvan *E* címe
- Mivel a gyökér minden egyedet ismer, az algoritmus terminálása garantált

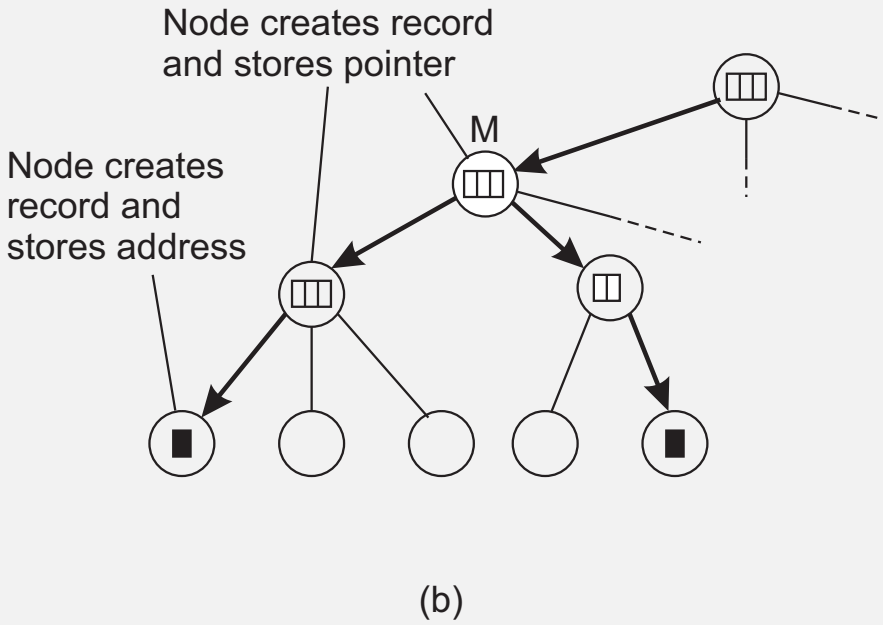


HLS: Beszúrás

Beszúrás a fában

- Ugyanaddig megyünk felfelé a fában, mint keresésnél
- Az érintett belső csúcsokba mutatókat helyezünk
- Egy csúcsban egy egyedhez több mutató is tartozhat

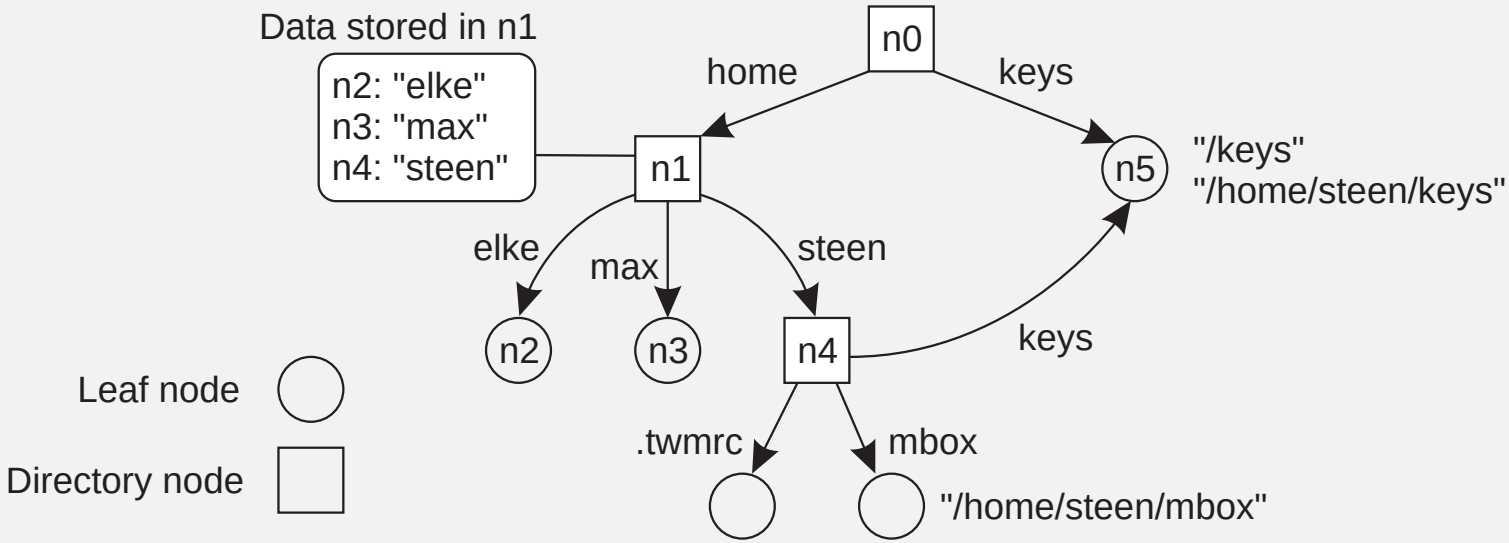




Névtér

Névtér

névtér: gyökeres, irányított, élcímkézett gráf, a levelek tartalmazzák a megnevezett egyedeket, a belső csúcsokat katalógusnak vagy könyvtárnak (directory) nevezzük
Az egyedhez vezető út címkéit összeolvasva kapjuk az egyed egy nevét. A bejárt út, ha a gyökérből indul, abszolút útvonalnév, ha máshonnan, relatív útvonalnév. Mivel egy egyedhez több út is vezethet, több neve is lehet.



Névtér

Attribútumok

A csúcsokban (akár a levelekben, akár a belső csúcsokban) különféle attribútumokat is eltárolhatunk.

- Az egyed típusát
- Az egyed azonosítóját
- Az egyed helyét/címét
- Az egyed más neveit
- ...

Névfeloldás

Gyökér szükséges

Kiinduló csúcsra van szükségünk ahhoz, hogy megkezdhessük a névfeloldást.

Gyökér megkeresése

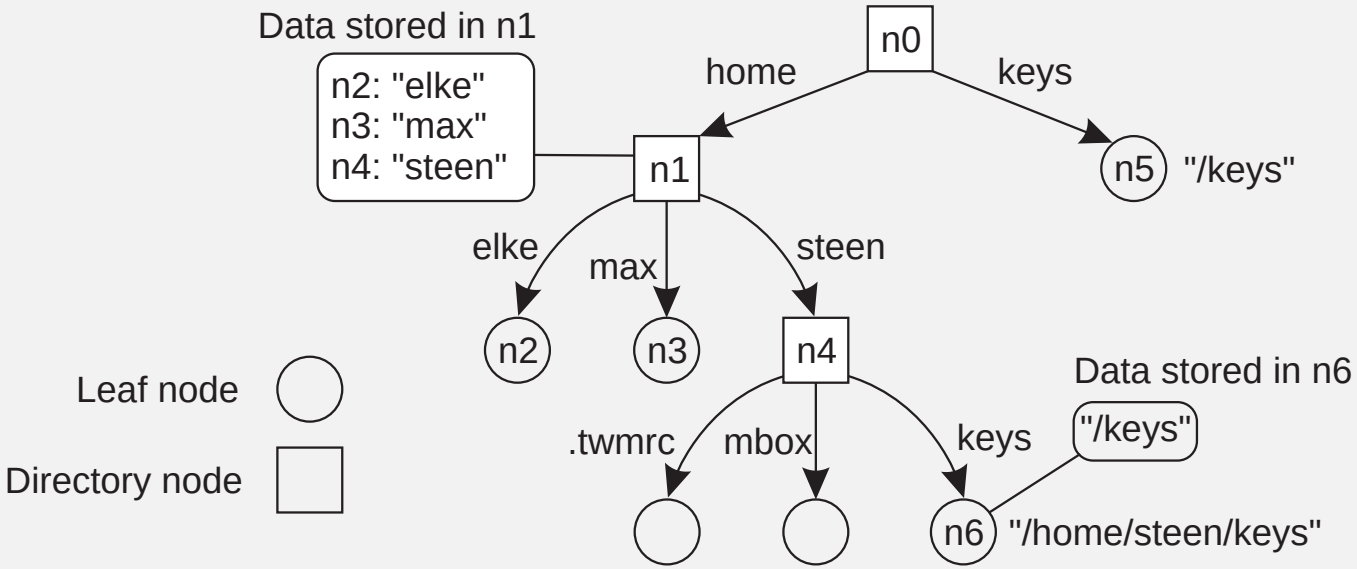
A név jellegétől függő környezet biztosítja a gyökér elérhetőségét. Néhány példa név esetén a hozzá tartozó környezet:

- *www.inf.elte.hu*: egy DNS névszerver
- */home/steen/mbox*: a lokális NFS fájlszerver
- *0031204447784*: a telefonos hálózat
- *157.181.161.79*: a www.inf.elte.hu webszerverhez vezető út

Csatolás (linking)

Soft link

A gráf csúcsai valódi csatolások (hard link), ezek adják a névfeloldás alapját.
soft link: a levelek más csúcsok álneveit is tartalmazhatják. Amikor a névfeloldás ilyen csúcshoz ér, az algoritmus az álnév feloldásával folytatódik.



A névtér implementációja

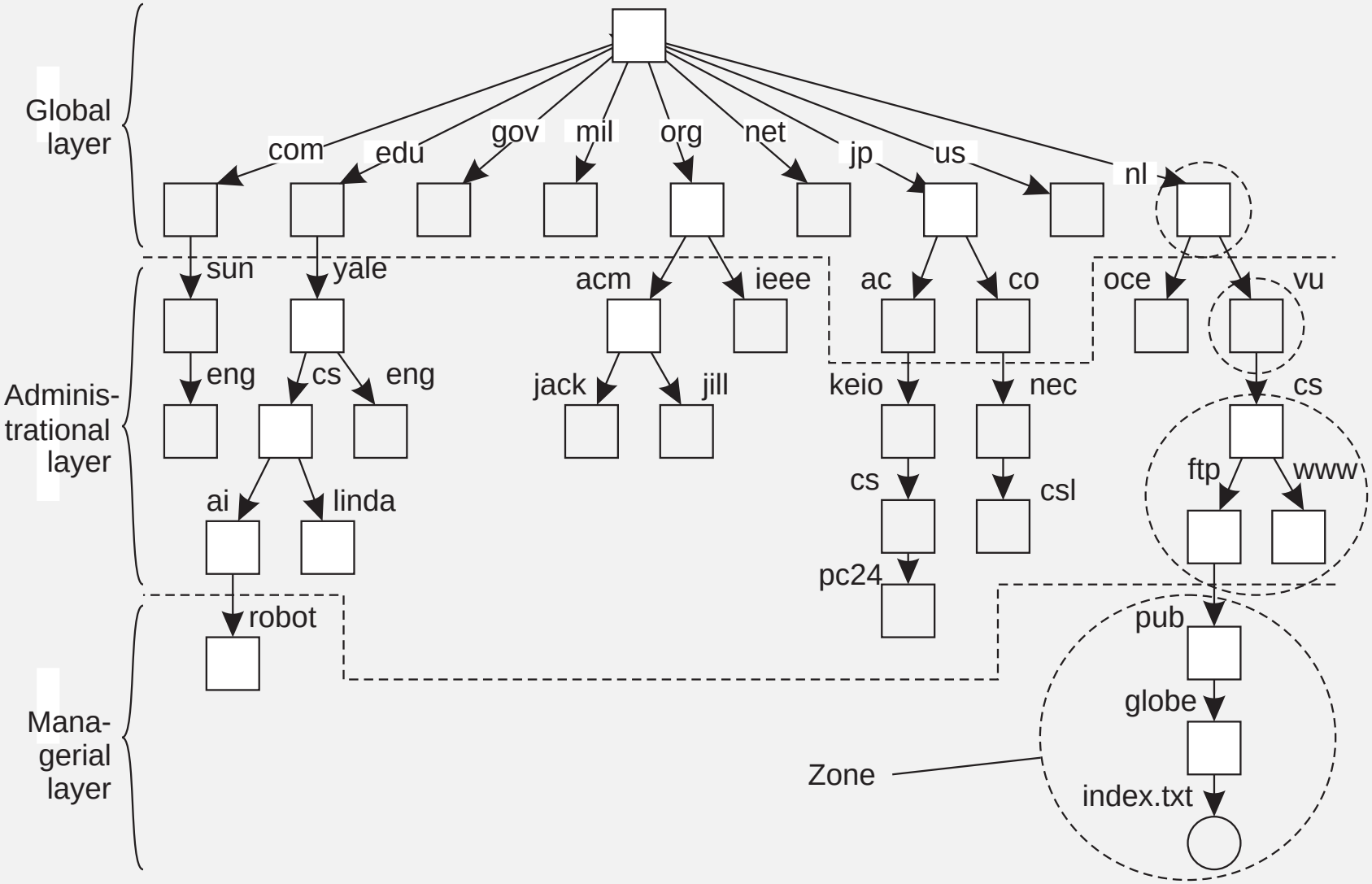
Nagyméretű névtér tagolása

Ha nagy (világméretű) névterünk van, el kell osztanunk a gráfot a gépek között, hogy hatékonnyá tegyük a névfeloldást és a névtér kezelését.
Ilyen nagy névteret alkot a DNS (Domain Name System).

- Globális szint: Gyökér és felső csúcsok. A szervezetek közösen kezelik.
- Szervezeti szint: Egy-egy szervezet által kezelt csúcsok szintje.
- Kezelői szint: Egy adott szervezeten belül kezelt csúcsok.

Szempont	Globális	Szervezeti	Kezelői
Földrajzi méret	Világméretű	Vállalati	Vállalati alegység
Csúcsok száma	Kevés	Sok	Rendkívül sok
Keresés ideje	mp.	ezredmp.	Azonnal
Frissítés terjedése	Ráérős	Azonnal	Azonnal
Másolatok száma	Sok	Nincs/kevés	Nincs
Kliens gyorsítótáraz?	Igen	Igen	Néha

A névtér implementációja: DNS



A névtér implementációja: DNS

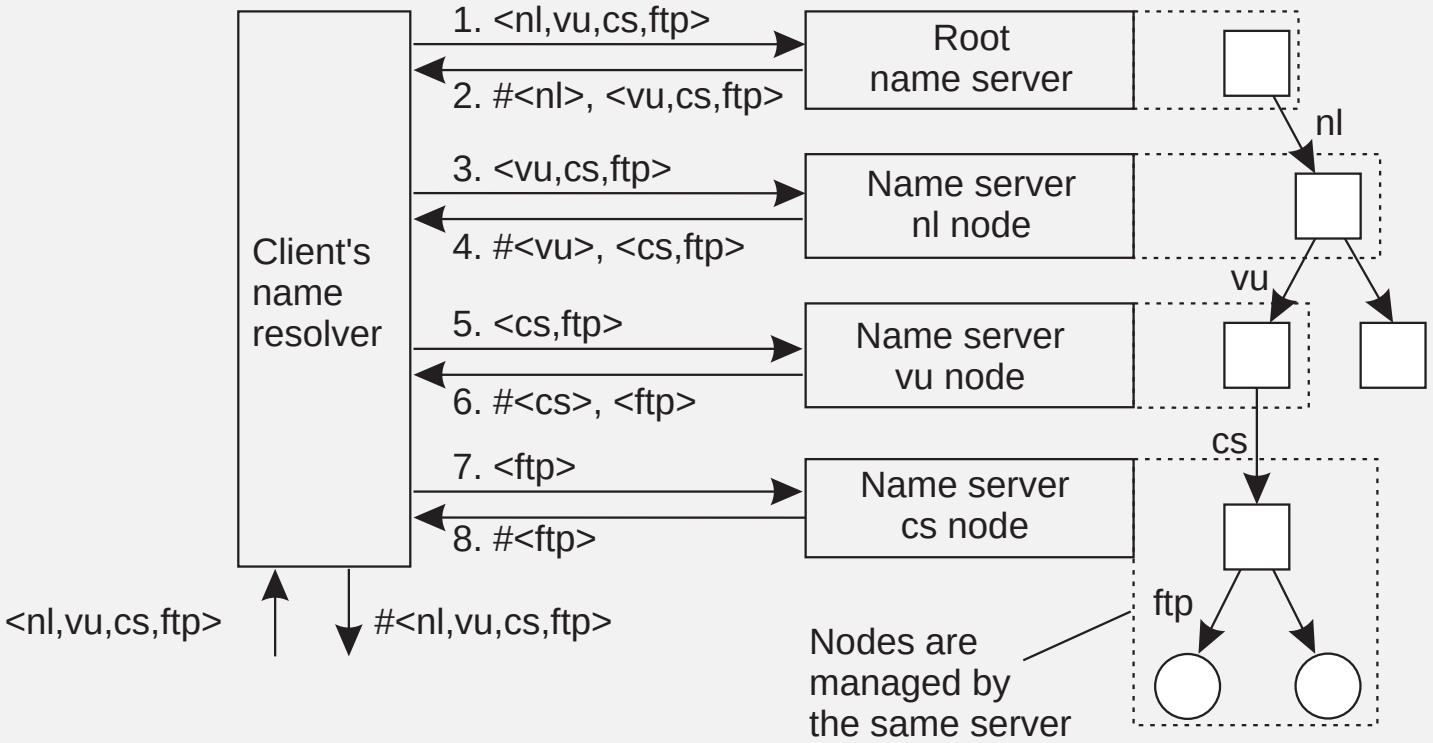
A DNS egy csúcsában tárolt adatok

Legtöbbször az A rekord tartalmát kérdezzük le; a névfeloldáshoz feltétlenül szükséges az NS rekord.
Egy zóna a DNS-fa egy összefüggő, adminisztratív egységként kezelt része, egy (ritkábban több) tartomány (domain) adatait tartalmazza.

Rekord neve	Adat jellege	Leírás
A	Gazdagép	A csomópont gazdagépének IP címe
NS	Zóna	A zóna névszervere
SOA	Zóna	A zóna paraméterei
MX	Tartomány	A csomópont levelezőszervere

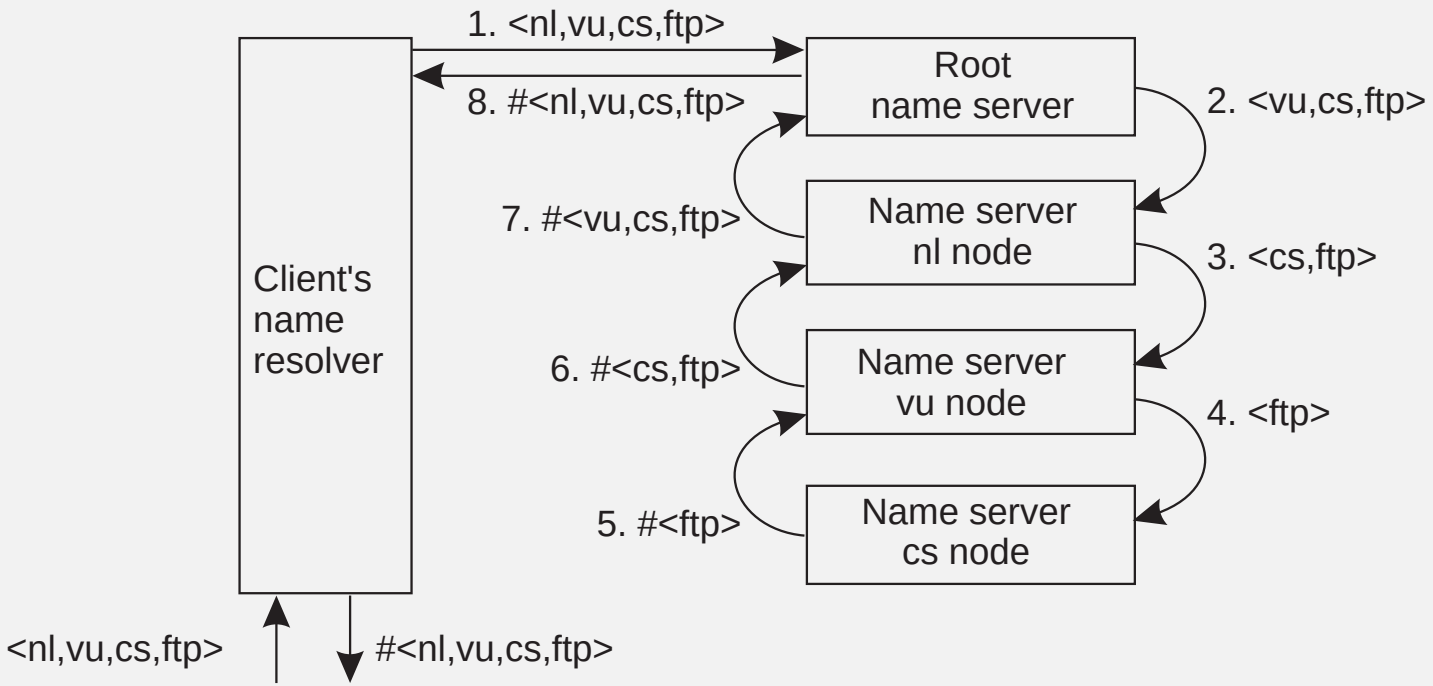
Iteratív névfeloldás

A névfeloldást a gyökér névszerverek egyikétől indítjuk.
Az iteratív névfeloldás során a névnek mindig csak egy komponensét oldjuk fel, a megszólított névszerver az ehhez tartozó névszerver címét küldi vissza.



Rekurzív névfeloldás

A rekurzív névfeloldás során a névszerverek egymás közt kommunikálva oldják fel a nevet, a kliensoldali névfeloldóhoz rögtön a válasz érkezik.



Rekurzív névfeloldás: cache-elés

A névszerver ezt kezeli	Feloldandó cím	Feloldja	Átadja lefele	Fogadja és cache-eli	Visszaadja
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>
(gyökér)	<nl,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

Névfeloldás: átméretezhetőség

Méret szerinti átméretezhetőség

Sok kérést kell kezelni rövid idő alatt $\Rightarrow$ a globális szint szerverei nagy terhelést kapnának.

Csúcsok adatai sok névszerveren

A felső két szinten, és sokszor még az alsó szinten is ritkán változik a gráf. Ezért megtehetjük, hogy a legtöbb csúcs adatairól sok névszerveren készítünk másolatot, így a keresést várhatóan sokkal közelebbről indítjuk.

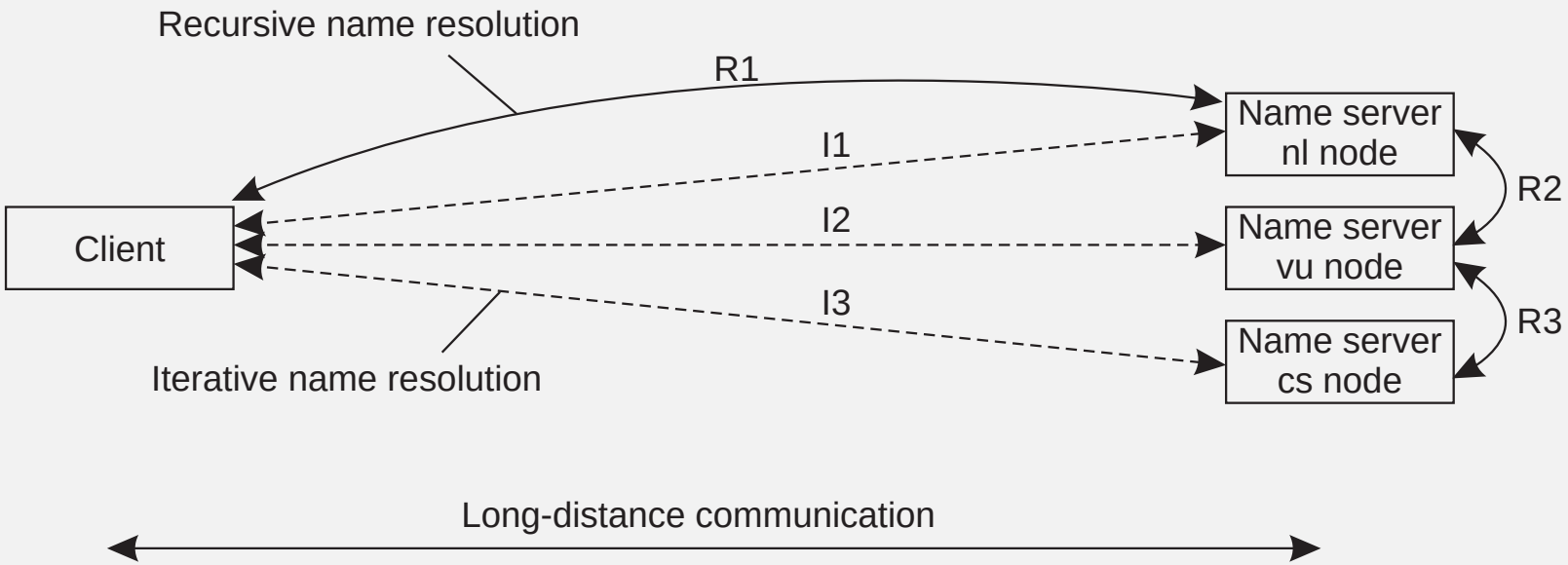
A keresett adat: az entitás címe

A legtöbbször a névfeloldással az entitás címét keressük.
A névszerverek nem alkalmasak mozgó entitások címeinek kezelésére, mert azok költözésével gyakran változna a gráf.

Névfeloldás: átméretezhetőség

Földrajzi átméretezhetőség

A névfeloldásnál a földrajzi távolságokat is figyelembe kell venni.



Helyfüggés

Ha egy csúcsot egy adott névszerver szolgál ki, akkor földrajzilag oda kell kapcsolódnunk, ha el akarjuk érni a csúcsot.

Attribútumalapú nevek

Attribútumalapú keresés

Az egyedeket sokszor kényelmes lehet a tulajdonságaik (attribútumaik) alapján keresni.

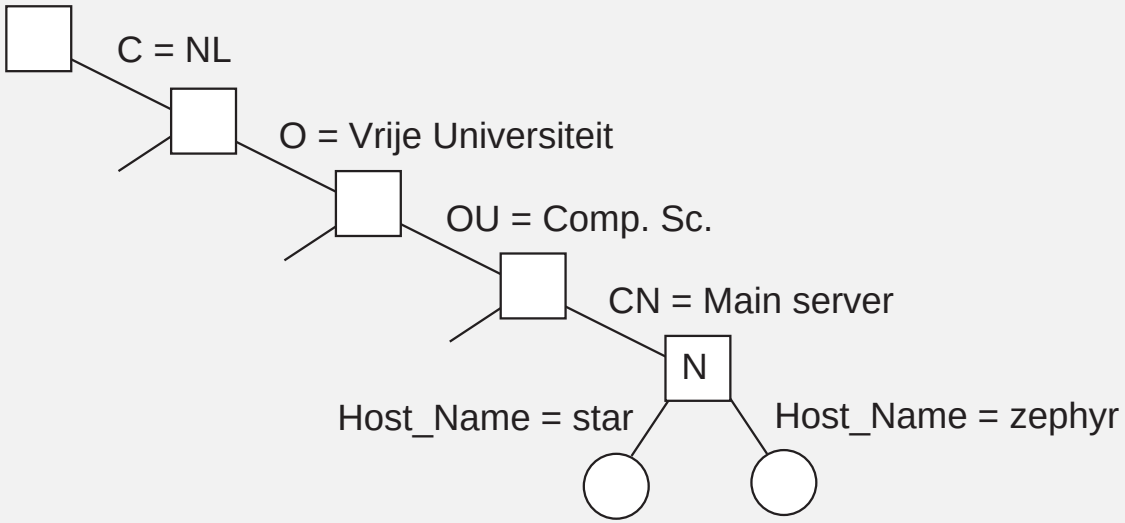
Teljes általánosságban: nem hatékony

Ha bármilyen kombinációban megadhatunk attribútumértékeket, a kereséshez az összes egyedet érintenünk kell, ami nem hatékony.

X.500, LDAP

A katalógusszolgáltatásokban (directory service) az attribútumokra megkötések érvényesek. A legismertebb ilyen szabvány az X.500, amelyet az LDAP protokollon keresztül szokás elérni.
Az elnevezési rendszer fastruktúrájú, élei névalkotó jellemzőkkel (attribútum-érték párokkal) címzettek. Az egyedekre az útjuk jellemzői vonatkoznak, és további párokat is tartalmazhatnak.

Példa: LDAP



Attribute	Value
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Comp. Sc.
CommonName	Main server
Host_Name	star
Host_Address	192.31.231.42

Attribute	Value
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Comp. Sc.
CommonName	Main server
Host_Name	zephyr
Host_Address	137.37.20.10

```
answer =
search("&(C = NL) (O = Vrije Universiteit) (OU = *) (CN = Main server)")
```