mas Distribuidos

Servicio de Nombres

Índice

- Introducción
- Servicio de nombres
  - Estudio de un ejemplo práctico: DNS
- Servicio de directorio
  - Estudio de un ejemplo práctico: LDAP
- Descubrimiento de servicios

a basada en hechos reales

r con persona en un contexto para pedirle algo
organización, una ciudad, un país, el mundo, ...
ección de contacto (p.e. nº teléfono en ese contexto)
) [permanente] → *Dirección* (*dónde*) [transitorio]
e nombres y direcciones no son tan diferentes…
co "páginas blancas" (Servicio de nombres)
er nº teléfono de servicio de guía del contexto dado
a persona con "nombre" unívoco en ese contexto
llidos | nº empleado | nº DNI
na nivel de indirección respecto a dirección contacto
persona cambie nº tfno (cuidado con agenda-caché)
se cadena de consultas; ¿nº tfno empleado?:
º tfno empresa; 2º centralita empresa me da tfno empleado

Una historia basada en hechos reales

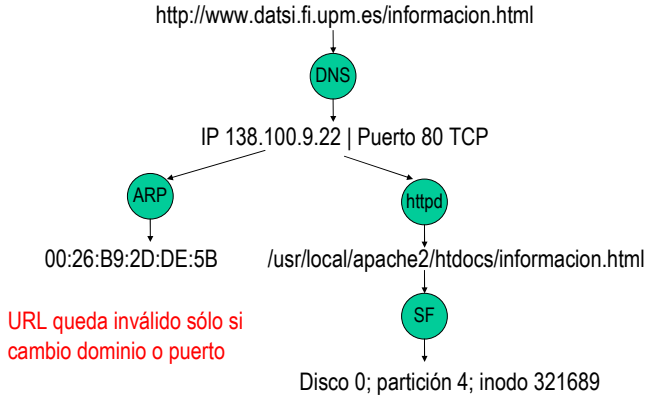
- Nombres y direcciones suelen tener carácter jerárquico
  - Facilita su administración y gestión
  - Ejs. Nombres: ID empleado internacional (ISBN, cuenta bancaria, ...)
    - Cambio de recurso en jerarquía puede invalidar el nombre
  - Ejs. Direcciones: nº teléfono o dirección postal
  - Encaminamiento jerárquico
- A veces quiero contactar con cualquiera que dé un servicio
  - Necesito conocer condiciones de servicio para elegir
- Servicio telefónico "páginas amarillas" (Servicio de directorio)
- ¿Y si ni siquiera sé nº tfno. de servicios de guía (o no los hay)?
  - Quizás debería gritar pidiendo ayuda
  - Descubrimiento de servicios

orm Resource Identifier

ficadores de recursos URIs en Internet:
s y direcciones URLs
ce Name: *Nombre* (*qué*) [permanente]
urso sin incluir información de localización
oceso de traducción
ce Locator: *Dirección* (*dónde*) [¿transitorio?]
afectados si recurso "se mueve"
e pueden considerar permanentes
edia URN vs. URL
7
org/html/rfc3187.html

5 Fernando Pérez Costoya

Ejemplo: Niveles de traducción de URL



Sistemas Distribuidos 6 Fernando Pérez Costoya

rvicio de nombres

dad en SD → punto(s) de acceso a la entidad
IP+ puerto(s)+ protocolo(s)
: referencia(s) a objeto(s)
referirse a una entidad única en SD
estar replicada (p.e. fichero en Coda)
varios nombres para la misma entidad (alias)
os de entidades en SD
ios, grupos, procesos, dispositivos, máquinas, ...
es específicos para algunos tipos de entidades
SFD), para máquinas (DNS), ...
e nombres integral para todas las entidades
os por gran volumen y frecuencia de actualizaciones

7 Fernando Pérez Costoya

Jerarquía de nombres

- SD incluye muchas entidades muy diversas
  - Como SFD, organización jerárquica facilita asignación y gestión
    - Impresoras de distintos departamentos con el mismo nombre
- Espacio de nombres jerárquico
  - Entidades contenedoras de otras entidades (directorios)
- Traducción de nombres (*pathnames*):
  - Proceso iterativo que parte de un nodo inicial
    - Necesidad de conocer traducción de nodo inicial
    - Absoluta (nodo raíz) vs. Relativa (nodo intermedio)
  - Proceso costoso: Uso de caché en traducción
    - Información inválida si migración
    - Garantía de validez vs. coste de mantener la coherencia de la caché
    - Muchos servicios de nombres pueden devolver información obsoleta

Sistemas Distribuidos 8 Fernando Pérez Costoya

ción de servicio de nombres

l servicio de nombres
ociar nombre con una entidad; Crear directorio
re

de servidor de nombres único:
escala, rendimiento y fiabilidad

ara paliarlos:
ucciones
espacio de nombres
espacio de nombres

cciones:
oherencia
ay garantía
orna TTL ("tiempo de vida") de información

Distribución y replicación

- Espacio de nombres partido y distribuido entre servidores
  - Se requiere info. que "monte" particiones para formar árbol único
  - Cada partición gestionada por (al menos) un servidor
    - Posibilita administración distribuida
  - Mismas alternativas de navegación que en SFD
    - Iterativa, Transitiva y Recursiva
- Partición replicada en varios servidores
  - Fiabilidad y rendimiento, pero hay que asegurar coherencia
    - Fiabilidad: mejor réplicas en distintas subredes
  - Esquema simétrico:
    - Consulta a cualquier réplica
    - Actualización simultánea en todas las réplicas
  - Esquema asimétrico: 1 primario/maestro y N secundarios/esclavos
    - Consulta a cualquier réplica
    - Actualización en primario con propagación a réplicas (Modo *push* o *pull*)

entaciones alternativas

st/multicast:
ecurso de nombre N?
SD propósito general (mala escalabilidad y eficiencia)
rimiento servicios o para aspectos específicos (ARP)

es un mecanismo de traducción

s de ubicación probable
temas con recursos móviles
o migra, antigua ubicación almacena la nueva
nas de ubicación probable de un objeto
gue cadena hasta encontrar ubicación actual
traviesa cadena, intermediarios saben nueva ubicación
de tener un nodo *home*

Domain Name System (DNS)

- Servicio de nombres de máquinas en Internet: nombre → IP
  - No es un serv. nombres general pero ilustrativo por escalabilidad
  - Diseño genérico: aunque uso habitual nombre de máquinas Internet
  - Inicios de Internet: fichero HOST que se actualizaba periódicamente
- Espacio de nombres de DNS jerárquico
  - Nombre: secuencia de dominios (≈directorios) de dcha. a izda.
    - www.datsi.fi.upm.es. → . + es + upm + fi + datsi
  - Dominio raíz: . → Caminos absolutos (FQDN) terminan con .
  - Dominios nivel superior (TLD)
    - gTLDs: genéricos (com, org, ...)
    - ccTLDs: por país (¿qué pasa con el de Tuvalu?)
  - De segundo nivel, de tercero, ...
- Implementación más usada BIND

nombres distribuido: Zonas

ción del árbol global (zona ≠ dominio)
ursos de un dominio y sus subdominios no delegados
dominios
io puede tener su propia zona
e incluye "punto de montaje" a esa zona subordinada
: delegar todos los subdominios
a cada dominio (zona ≈ dominio)
es a los mismos servidores que el dominio del que cuelgan

replicada:
tro/primario y *N* (al menos 1) esclavos/secundarios
r réplicas en distintas subredes

tenida en una zona:
esource Records (RR) que describen sus recursos

13 Fernando Pérez Costoya

o SOA (*Start of Authority*)

ición de una zona
ón en fichero de zona (wikipedia)
ple.com. username.example.com. (
ber of this zone file
day)
in case of a problem (2 hours)
time (4 weeks)
ing time in case of failed lookups (1 hour)

a: *dig fi.upm.es. SOA*
SOA chita.fi.upm.es. hostmaster.fi.upm.es. 2013102101 28800 7200 2419200

tart Of Authority; S. maestro; responsable; nº serie (incrementar si cambio);
cundario; Tiempo de reintento de secundario antes actualización fallida;
le secundario ante actualización fallida; TTL para cache negativa (tiempo en as)

15 Fernando Pérez Costoya

Resource Record

- Definición de un recurso: *Nombre Tipo Clase TTL Datos*
  - Clase *IN* para Internet (otros *HS*, para Hesiod, y *CH*, para Chaos)
  - NOTA: *Nombre* puede tener * a la izqda. (*wildcard RR*; no lo tratamos)
- Fichero de zona:
  - Fichero de texto en primario define RRs de una zona: 1 RR/línea
    - Aunque RRs se transmiten en binario
  - Incluye RRs de recursos del dominio y de subdominios no delegados
  - Sintaxis definida para facilitar introducción de datos en fichero de zona
    - Macros, caracteres especiales, caminos relativos, omisión de campos,...
- Diversos tipos de RRs
  - Nos centramos en SOA, A, AAAA, PTR, CNAME, MX, SRV, TXT y NS
  - No tratamos los RRs relacionados con la extensión DNSSEC
    - Proporciona autenticación e integridad en DNS

Sistemas Distribuidos 14 Fernando Pérez Costoya

RR de tipo A o AAAA

- Dirección de máquina: A (IPv4) y AAAA (IPv6)
- Ejemplo de definición en fichero de zona (wikipedia)

```
www.example.com.   A      192.0.2.1      ; IPv4 address for example.com
www.example.com.   AAAA   2001:db8:10::1  ; IPv6 address for example.com
```

- Ejemplo de consulta: *dig www.fi.upm.es. A*

```
www.fi.upm.es.     86400  IN   A    138.100.243.10
```

- Múltiples recursos con mismo nombre (reparto de carga)

```
www.google.es.     300    IN   A    130.206.193.48
www.google.es.     300    IN   A    130.206.193.59
www.google.es.     300    IN   A    130.206.193.26
..... Hasta 16 .....
```

  - Nótese TTL bajo en RR para favorecer el reparto de carga

Sistemas Distribuidos 16 Fernando Pérez Costoya

RR de tipo PTR

- sa dirección IP → Nombre
- ión por DNS: mediante dominios especiales
  - *dr.arpa*.
  - 0.243.10 → 10.243.100.138.in-addr.arpa.
  - *rpa*.
  - 20:41c:40:12:100:4:4 → 0.0.1.0.2.1.0.0.0.4.0.0.c.1.4.0.0.2.7.0.1.0.0.2.ip6.arpa.
- sulta:
  - *138.in-addr.arpa. PTR*
  - 86400 IN PTR www.fi.upm.es.
  - *0.0.1.0.2.1.0.0.0.4.0.0.c.1.4.0.0.2.7.0.1.0.0.2.ip6.arpa. PTR*
  - .0.4.0.0.c.1.4.0.0.2.7.0.1.0.0.2.ip6.arpa. 86302IN PTR

RR de tipo CNAME (Canonical NAME)

- Alias: Nuevo nombre para mismo recurso

```
www.datsi.fi.upm.es.      86400  IN   CNAME  avellano.datsi.fi.upm.es.
avellano.datsi.fi.upm.es. 86400  IN   A      138.100.9.22
```

- Frente a:

```
www.datsi.fi.upm.es.      86400  IN   A      138.100.9.22
avellano.datsi.fi.upm.es. 86400  IN   A      138.100.9.22
```

- Más flexibilidad ante cambios pero ineficiencia por indirección
- Pueden encadenarse:

```
www.elpais.com.          967    IN   CNAME  elpais.es.edgesuite.net.
www.elpais.es.           1456   IN   CNAME  elpais.es.edgesuite.net.
elpais.es.edgesuite.net. 10467  IN   CNAME  a1749.g.akamai.net.
a1749.g.akamai.net.      20     IN   A      130.206.192.24
a1749.g.akamai.net.      20     IN   A      130.206.192.49
```


RR de tipo MX

- rreo para dominio con orden de preferencia
- ass MX priority target
- ulta: *dig upm.es. MX*
  - MX 10 relay.upm.es.
  - MX 30 relay4.upm.es.
  - MX 50 correo.upm.es.
- orden de preferencia:↓ prioridad → ↑ preferencia
- rreo debe contactar con servidor de menor nº siguiente, …

RR de tipo SRV

- Permite especificar qué máquinas dan un servicio en el dominio
- Formato:
  - *_service._proto.name TTL class SRV priority weight port target*
- Permite especificar prioridades y reparto entre misma prioridad
- Ejemplo de wikipedia:

```
_sip._tcp.example.com. 86400 IN SRV 10 60 5060 bigbox.example.com.
_sip._tcp.example.com. 86400 IN SRV 10 20 5060 smallbox1.example.com.
_sip._tcp.example.com. 86400 IN SRV 10 10 5060 smallbox2.example.com.
_sip._tcp.example.com. 86400 IN SRV 10 10 5066 smallbox2.example.com.
_sip._tcp.example.com. 86400 IN SRV 20 0 5060 backupbox.example.com.
```

- ¿Por qué se usan tan poco? ¿Por qué no se usan para la Web?

RR de tipo TXT

texto con un nombre

iadir funcionalidad a DNS sin nuevos RRs

icación:

Framework (SPF):

náquinas de un dominio pueden enviar correo

xiste también un RR de tipo SPF

iedad de un dominio para Google

ulta: *dig fi.upm.es. TXT*

```
XT   "v=spf1 ip4:138.100.8.0/24 ip4:138.100.198.0/24 ip4:138.100.4.67 -all"
XT   "google-site-verification=rJT2Yatvyyg4HepVHZ-
gU"
```

21 Fernando Pérez Costoya

RR de tipo NS (*Name Server*)

- Primer uso: especificar servidores de nombres para un dominio
- Ejemplo: *dig fi.upm.es. NS*

```
fi.upm.es.          86400  IN  NS  chita.fi.upm.es.
fi.upm.es.          86400  IN  NS  zape.fi.upm.es.
fi.upm.es.          86400  IN  NS  tarzan.fi.upm.es.
fi.upm.es.          86400  IN  NS  galileo.ccupm.upm.es.
fi.upm.es.          86400  IN  NS  ns.fi.upm.es.
```

- Ejemplo: *dig 8.100.138.in-addr.arpa. NS*

```
8.100.138.in-addr.arpa. 86400  IN  NS  zape.fi.upm.es.
8.100.138.in-addr.arpa. 86400  IN  NS  chita.fi.upm.es.
8.100.138.in-addr.arpa. 86400  IN  NS  galileo.ccupm.upm.es.
8.100.138.in-addr.arpa. 86400  IN  NS  tarzan.fi.upm.es.
8.100.138.in-addr.arpa. 86400  IN  NS  ns.fi.upm.es.
```

Sistemas Distribuidos 22 Fernando Pérez Costoya

po NS para delegación

legar subdominio a s.nombres (pto. montaje)

nombre del RR el del subdominio

ios delegados no se puede obtener mediante consulta

delega administración de sus recursos DNS a FI:

na de *upm.es.* debe incluir:

```
chita.fi.upm.es.
zape.fi.upm.es.
tarzan.fi.upm.es.
galileo.ccupm.upm.es.
ns.fi.upm.es.
```

na de *100.138.in-addr.arpa.* debe incluir:

```
NS   zape.fi.upm.es.
NS   chita.fi.upm.es.
NS   galileo.ccupm.upm.es.
NS   tarzan.fi.upm.es.
NS   ns.fi.upm.es.
```

legación y CIDR (no lo tratamos)

23 Fernando Pérez Costoya

Ejemplo hipotético

- Empresa con sede central y tres departamentos
  - Un administrador gestiona sede central, dep1 y dep2
  - Dep3 con administrador propio (y servidor de correo propio)
- Detalles de servidores de nombres de cada dominio:
  - Sede central de la empresa (emp.es.): 2 s. de nombres
    - maestro en dominio (ns.emp.es.);esclavo externo (ns.isp.com.)
  - Dep1 (dep1.emp.es.): subdominio no delegado
  - Dep2 (dep2.emp.es.): subdominio delegado a mismos servidores
  - Dep3 (dep3.emp.es.): 3 s. de nombres
    - Maestro (ns1.dep3.emp.es.); esclavo interno (ns2) y externo (ns.isp.com.)
- Empresa tiene asignada red clase B 138.99.0.0
  - 138.99.1 central; 138.99.2 dep1; 138.99.3 dep2; 138.99.4 dep3
  - Sólo está delegado subdominio de 138.99.4

Sistemas Distribuidos 24 Fernando Pérez Costoya

zona emp.es. (ns.emp.es.)

```
    IN    SOA    ns.emp.es ....
    IN    NS     ns.emp.es. ; servidor maestro del dominio
    IN    NS     ns.isp.com. ; servidor esclavo externo

    IN    NS     ns.emp.es. ; servidor maestro en el dominio padre
    IN    NS     ns.isp.com. ; servidor esclavo externo
! subdominio
    IN    NS     ns1.dep3.emp.es. ; servidor maestro en su propio subdominio
    IN    NS     ns2.dep3.emp.es. ; servidor esclavo en su propio subdominio
    IN    NS     ns.isp.com. ; servidor esclavo externo
    IN    MX     10 mail1.emp.es. ; servidor de correo preferente para la empresa
    IN    MX     20 mail2.emp.es. ; servidor de correo de reserva para la empresa
    IN    MX     10 mail1.emp.es. ; servidor de correo preferente para dep1
    IN    MX     20 mail2.emp.es. ; servidor de correo de reserva para dep1

    IN    A      138.99.1.1
    IN    A      138.99.1.2
    IN    A      138.99.1.3
    IN    A      138.99.1.4
    IN    A      138.99.2.1; RR de subdominio no delegado en misma zona

    IN    A      138.99.4.1
    IN    A      138.99.4.2
```


F. zona dep2.emp.es. (ns.emp.es.)

```
dep2.emp.es.                IN   SOA   ns.emp.es ....
dep2.emp.es.      86400     IN   NS    ns.emp.es. ; servidor maestro del dominio (=padre)
dep2.emp.es.      86400     IN   NS    ns.isp.com. ; servidor esclavo externo
; Correo
dep2.emp.es.      86400     IN   MX    10 mail1.emp.es. ; s. correo preferente para dep2
dep2.emp.es.      86400     IN   MX    20 mail2.emp.es. ; s. correo de reserva para dep2
; Máquinas en el dominio de dep2
www.dep2.emp.es.  86400     IN   A     138.99.3.1
backup.dep2.emp.es. 86400   IN   CNAME www.dep2.emp.es.
```


F. zona dep3.emp.es. (ns.dep3.emp.es.)

```
    IN    SOA    ns1.dep3.emp.es ....
    IN    NS     ns1.dep3.emp.es.; servidor maestro del dominio (!=padre)
    IN    NS     ns2.dep3.emp.es.; servidor esclavo en el propio dominio
    IN    NS     ns.isp.com. ; servidor esclavo externo

    IN    MX     10 mail.dep3.emp.es. ; s. correo preferente para dep3
    IN    MX     20 mail2.emp.es. ; s. correo de reserva para dep3
ep3
    IN    A      138.99.4.1
    IN    A      138.99.4.2
    IN    A      138.99.4.3
    IN    A      138.99.4.4; reparto de carga en servicio web
    IN    A      138.99.4.5 ; reparto de carga en servicio web
```


F. zona 99.138. (ns.emp.es.)

```
99.138.in-addr.arpa.              IN   SOA   ns.emp.es ....
99.138.in-addr.arpa.      86400   IN   NS    ns.emp.es.
99.138.in-addr.arpa.      86400   IN   NS    ns.isp.com.
; dir. IP de dep3 delegadas a servidor maestro en el subdominio (no se necesitan glue records)
4.99.138.in-addr.arpa.    86400   IN   NS    ns1.dep3.emp.es.
4.99.138.in-addr.arpa.    86400   IN   NS    ns2.dep3.emp.es.
4.99.138.in-addr.arpa.    86400   IN   NS    ns.isp.com.
; Máquinas de sede central, dep1 y dep2
1.1.99.138.in-addr.arpa.  86400   IN   PTR   ns.emp.es.
2.1.99.138.in-addr.arpa.  86400   IN   PTR   mail1.emp.es.
3.1.99.138.in-addr.arpa.  86400   IN   PTR   mail2.emp.es.
4.1.99.138.in-addr.arpa.  86400   IN   PTR   www.emp.es.
1.2.99.138.in-addr.arpa.  86400   IN   PTR   www.dep1.emp.es.
1.3.99.138.in-addr.arpa.  86400   IN   PTR   www.dep2.emp.es.
```

F. zona 4.99.138. (ns.dep3.emp.es.)

```
        IN   SOA    ns1.dep3.emp.es. ....
6400    IN   NS     ns1.dep3.emp.es.
6400    IN   NS     ns2.dep3.emp.es.
6400    IN   NS     ns.isp.com.

6400    IN   PTR    ns1.dep3.emp.es.
6400    IN   PTR    ns2.dep3.emp.es.
6400    IN   PTR    mail.dep3.emp.es.
6400    IN   PTR    www.dep3.emp.es.
6400    IN   PTR    www.dep3.emp.es.
```


Glue records

- Posibles círculos viciosos en la traducción de nombres
- Si s. nombres de subdominio (Ssub) pertenece a subdominio
  - En el ejemplo: ns1.dep3.emp.es. y ns2.dep3.emp.es.
  - Para obtener IP de cualquier máq. subdominio → contactar con Ssub
    - ¿IP de www.dep3.emp.es? → contactar con ns1.dep3.emp.es.
  - Pero para hacerlo necesito IP de Ssub → contactar con Ssub
    - ¿IP de ns1.dep3.emp.es.? → contactar ns1.dep3.emp.es.
- *Glue record (GR)*
  - RR de tipo A/AAAA que se incluye en un dominio ajeno
  - Solución c. vicioso: padre debe incluir RR tipo A con dir. IP de Ssub
  - Aumenta problemas de coherencia
    - Cambios en IP de Ssub deben reflejarse también en dominio padre
  - No necesario *glue record* para servidor externo o en dominio padre
    - Siempre se puede obtener su traducción

: delegaciones en UPM

```
n.ccupm.upm.es. hostmaster.upm.es. 2013101401 86400 7200 1209600 3600

einstein.ccupm.upm.es. hostmaster.upm.es. 2011060701 86400 7200 1209600 7200

chita.fi.upm.es. hostmaster.fi.upm.es. 2013102101 28800 7200 2419200 3600

A   chita.fi.upm.es. hostmaster.fi.upm.es. 2012121801 28800 7200 2592000 3600
```

egaciones existen
esitan *glue records*?

Servidores de nombres raíces

- Hay "13" servidores de dominio raíz (.) replicados
  - Desde *a.root-servers.net* hasta *m.root-servers.net*
  - "13" porque esa información cabe en paquete UDP
    - DNS usa UDP (53); y sólo TCP(53) cuando tamaño lo aconseja
  - ¿Problemas de escalabilidad?
    - Detrás de cada uno hay múltiples servidores (uso de *anycast*)
  - Incluyen NS y *glue records* de dominios de nivel 1º (TLDs)
    - Aunque también gestionan algunos dominios de primer nivel → *arpa*.
  - Cada serv. DNS tiene dir. de servidores raíz (fichero *root.servers*)
    - Se debe actualizar periódicamente
- Lista y localización: *http://root servers.org*

Servidores DNS

- …na (*authoritative*) *N* (≥0) zonas directas/inversas
  - …ede ser maestro de otras esclavo
- …uede tener doble rol (algo confuso):
  - …ceso a sus zonas
  - …omo cliente en navegación recursiva
- …frecer navegación iterativa; recursiva opcional
  - …cursiva (no caché) → sólo acceso a sus zonas
  - …hé)→por seguridad sólo peticiones de ciertas máquinas
- …ersos tipos de servidores de nombres:
  - …aché (*nonauthoritative*), recursivo para clientes internos
  - …tes de una organización actuando de proxy
  - …ritative* y recursivo sólo para clientes internos
  - …tes de una organiz. y da acceso a sus zonas a todo el mundo
  - …ritative* no recursivo → p.e. servidor raíz o de TLD

Resolución de consultas

- …e una consulta *C* de *N*:
  - …RRs de todas sus zonas y de su caché (si usa)
  - …jor encaje → RR (*RRX*) que sea sufijo más largo de *C*
  - …pleto → envía a *N* consuelta resuelta
  - …mo *authoritative* si no proviene de la caché
  - …s satisfacen consulta, se envía a *N* lista con todos
  - … rota la lista cada vez (*Round-robin DNS*) para reparto de carga
  - …nformación adicional para agilizar la operación
  - …sulta MX puede retornar los RRs de tipo A de servidores de correo
  - …ompleto, *RRX* → NSs de dominio por donde continuar
  - …aso, los NSs de servidores raíz
- …siva: S envía consulta a uno de los NSs encontrados
- …cursiva: *S* envía a *N* los RRs de los NSs encontrados
- … direcciones de NSs encontrados
- …cursiva: las incluye como info. adicional en mens. de respuesta a *N*
- …siva: *S* las usa para contactar; sino tiene que obtenerlas

Resolver

- Parte cliente de DNS: da servicio a aplicaciones en un nodo
- Implementado habitualmente como biblioteca (⊂ en *libc*)
- Proporciona API para traducción directa e inversa:
  - UNIX: *gethostbyname/getaddrinfo* y *gethostbyaddr/getnameinfo*
- Configurado con servidores de nombres a los que consulta
  - Requiere también las direcciones IP de todos esos s. de nombres
  - En UNIX: */etc/resolv.conf*
- Esos servidores de nombres deben ser recursivos
  - *Resolver* no sabe navegar
- Puede usar caché (las aplicaciones también)
- UNIX permite configurar mecanismo de traducción de *hosts*
  - */etc/hosts*, DNS, NIS, LDAP (*/etc/nsswitch.conf*)

Ejemplos reales de traducción

- Operación de traducción directa: *www.fi.upm.es.*
- Operación de traducción inversa: *138.100.243.10.*
- *Resolver* tiene configurado como SN (3 opciones):
  - SN1: *ns.miempresa.com.*; IP 139.100.1.1
    - Zonas gestionadas: *miempresa.com.* y *100.139.in-addr.arpa., …*
  - SN2: *einstein.ccupm.upm.es.*; IP 138.100.4.8
    - Zonas gestionadas: *upm.es.* y *100.138.in-addr.arpa., …*
  - SN3: *zape.fi.upm.es.*; IP 138.100.8.1
    - Zonas gestionadas: *fi.upm.es.* y *243.100.138.in-addr.arpa., …*
- Supuestos:
  - traducción recursiva resolver-SN e iterativa desde SN
  - cachés vacías

ión directa usando SN1

- *gethostbyname("www.fi.upm.es.")* de *resolver*
- tición DNS de tipo A a dir. de SN1: 139.100.1.1
- : . → elige un s. raíz: *a.root-servers.net.* (198.41.0.4)
- *net.* mejor encaje: *es.* → envía a SN1 los NSs de *es.*
- *ecords* como información adicional
- . (194.69.254.1)
- encaje: *upm.es.* → envía a SN1 los NSs de *upm.es.*
- *ecords* como información adicional
- *.ccupm.upm.es.* (138.100.4.8)
- *.upm.es.* mejor encaje: *fi.upm.es.*
- los NSs de *fi.upm.es.* y sus *glue records*
- *upm.es.* (138.100.8.1)
- . Encaje completo: *www.fi.upm.es.*
- el NS de tipo A → *www.fi.upm.es.* | 138.100.243.10
- o envía al *resolver* y éste retorna la IP a la aplicación

Traducción directa usando SN2 y SN3

- Aplicación llama a *gethostbyname("www.fi.upm.es.")* de *resolver*
- *Resolver* envía petición DNS de tipo A a dir. de SN2: 139.100.4.8
- SN2 mejor encaje: *fi.upm.es.* → elige *zape.fi.upm.es.* (138.100.8.1)
  - *zape.fi.upm.es.* encaje completo: *www.fi.upm.es.*
    - envía a SN2 el NS de tipo A → *www.fi.upm.es.* (138.100.243.10)
      - SN2 se lo envía al *resolver* y éste retorna la IP a la aplicación

---

- Aplicación llama a *gethostbyname("www.fi.upm.es.")* de *resolver*:
- *Resolver* envía petición DNS de tipo A a dir. de SN3: 138.100.8.1
- SN3 encaje completo: *www.fi.upm.es.* (138.100.243.10)
  - SN3 se lo envía al *resolver* y éste retorna la IP a la aplicación

ión inversa usando SN1

- *gethostbyaddr(...138.100.243.10...)* de *resolver*
- TR *10.243.100.138.in-addr.arpa.* a SN1: 139.100.1.1
- : . → elige un s. raíz: *a.root-servers.net.* (198.41.0.4)
- *net.* mejor encaje: *in-addr.arpa.*
- 1 los NSs de *in-addr.arpa.* y sus *glue records*
- *lr-servers.net.* (199.212.0.73)
- *ers.net.* mejor encaje: *138.in-addr.arpa.*
- 1 los NSs de *138.n-addr.arpa.* (no sus *glue records*)
- *et.* → tiene que hacer la traducción directa completa
- el detalle por ya conocido; Obtiene 199.180.180.63
- r encaje: *100.138.in-addr.arpa.*
- 1 NSs de *100.138.n-addr.arpa.* (no sus *glue records*)

Traducción inversa usando SN1 (cont.)

- SN1 elige *einstein.ccupm.upm.es.* → necesita trad. directa completa
  - No se muestra el detalle por ya conocido; Obtiene 138.100.4.8
  - *einstein.ccupm.upm.es.* mejor encaje: *243.100.138.in-addr.arpa.*
    - envía a SN1 NSs de *243.100.138.in-addr.arpa.* y sus *glue records*
- SN1 elige *zape.fi.upm.es.* (138.100.8.1)
  - *zape.fi.upm.es.* encaje completo: *10.243.100.138.in-addr.arpa.*
    - envía a SN1 el NS de tipo PTR →
      - *10.243.100.138.in-addr.arpa* | *www.fi.upm.es.*
      - SN1 se lo envía al *resolver* y éste retorna nombre del *host* a la aplicación

inversa usando SN2 y SN3

gethostbyaddr(...138.100.243.10...) de *resolver*
R *10.243.100.138.in-addr.arpa.* a SN2: 139.100.4.8
243.100.138.in-addr.arpa. → elige *zape.fi.upm.es.*
encaje completo: *10.243.100.138.in-addr.arpa.*
el NS de tipo PTR →
00.138.in-addr.arpa | www.fi.upm.es.
lo envía al *resolver* y éste retorna nombre del *host* a la aplicación

gethostbyaddr(...138.100.243.10...) de *resolver*
R *10.243.100.138.in-addr.arpa.* a SN3: 138.100.8.1
eto: *10.243.100.138.in-addr.arpa.*
8.in-addr.arpa | www.fi.upm.es.
a al *resolver* y éste retorna nombre *host* a la aplicación

41 Fernando Pérez Costoya

Mantenimiento de info. de zona

- Sincronización de esclavo
  - Esclavo pide info. zona a maestro
    - Periódicamente (tal como lo especifica SOA)
    - O cuando maestro avisa de cambios (*NOTIFY*)
  - Si cambio: transferencia zona completa (*AXFR*) o incremental (*IXFR*)
  - Sólo se debe permitir transferencia de zona entre maestro y esclavos
- Actualización de DNS:
  - Cambio en fichero zona, incrementa nº en SOA y aviso a maestro
  - *Dynamic* DNS: Protocolo DNS incluye ops. para actualizar zona
    - Añadir, modificar y borrar RR pero no crear nuevas zonas
    - Mucho más flexible pero menos seguro
    - Algunas aplicaciones:
      - Permitir que máquinas mantengan mismo nombre en sistemas con DHCP
      - Servidor elige cualquier puerto y usa SRV (requerido por *Active Directory*)

Sistemas Distribuidos 42 Fernando Pérez Costoya

Índice

bres
ejemplo práctico: DNS
ctorio
ejemplo práctico: LDAP
de servicios

43 Fernando Pérez Costoya

Servicio de directorio

- Punto de acceso es sólo uno de los atributos de una entidad
  - Nombre impresora → modelo, color, ubicación, formatos soportados, ...
  - Pueden gestionarlos servidores específicos
    - Servicio de impresión gestiona información de impresoras
  - Problema: Duplicidad de funcionalidad
- Solución: Generalización del servicio de nombres
  - Nombre → conjunto de atributos de la entidad
- Aplicable a entidades de infraestructura del SD y de "negocio"
  - En FI: alumnos, profesores, títulos, asignaturas, dptos., servicios, ...
- Servicio de directorio (Sdir):
  - Repositorio de información de entidades de SD
    - Sun: "globalización" de */etc* → *Network Information System* (NIS)
  - No todo atributo ⊂ Sdir: no incluir atributos muy dinámicos
    - Tamaño cola de trabajos debería gestionarlo el servicio de impresión

Sistemas Distribuidos 44 Fernando Pérez Costoya

resolución de nombres

Páginas blancas): nombre → atributos
áginas amarillas): atributos → entidades
ir fichero en impresora en color cerca de mi despacho"
os en resolución por atributos:
de búsqueda (base)
scar entidades en determinada sucursal de la empresa
la búsqueda (sub-árbol, hijos directos, sólo base, ...)
búsqueda:
ca que deben satisfacer las entidades buscadas
ras en color que estén ubicadas en el tercer piso
o de búsqueda
entidades que se retornarán
e retornarán de las entidades seleccionadas

Tipos de entidades gestionadas

- ¿Qué tipos de entidades gestiona un servidor de nombres?
- Predeterminado:
  - Tipos de entidades predefinidas
- Configurable:
  - Existe un mecanismo para definir los tipos de las entidades
    - hay que definir: nombre del tipo, atribs., tipos de los atribs., etc.
  - Separación entre definición de tipos de entidades y de entidades
    - Similitud con base de datos: esquemas y datos
    - Similitud con POO: clases y objetos
  - Extensible: tipos predefinidos pero se pueden definir adicionales
    - Puede ser útil la herencia (simple o múltiple)

rectorio vs. Base de datos

ilitud
información
eda sofisticada
ferencias. Servicio de directorio:
ltas pero muy pocas modificaciones
muy simples
mas estándar, siempre que sea posible
para cambiar esquemas para datos ya creados
a profesor FI asignaturas que imparte
para datos jerárquicos
tiples valores para cada atributo
ados, no requiere coherencia estricta
s Sdir implementados con una base de datos

Lightweight Directory Access Protocol

- Precedente: X.500 servicio de directorio de ISO
  - Concebido para ser un directorio mundial
  - Complejo
  - Pesado: Ejecuta sobre la pila OSI
  - Protocolo de acceso DAP (*Directory Access Protocol*)
- LDAP (*Lightweight Directory Access Protocol*, RFC 4510)
  - Basado en X.500
  - Más sencillo
  - Más ligero: ejecuta sobre la pila TCP/IP
  - Es un protocolo pero define implícitamente un modelo de datos
    - No define aspectos de implementación
  - Distintos sistemas ofrecen una interfaz LDAP (p.e. *Active Directory*)
  - Actualmente versión 3

Objetos y clases

to (entrada) en LDAP
etos: Objeto ∈ Clase (atributo *objectClass*)
junto de atributos del objeto
o | obligatorio(ob) u optativo(op) | valor único o múltiple
s forman una jerarquía (*top* raíz de jerarquía)
hereda atributos de superclases

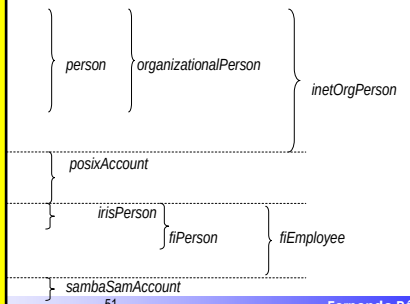
no pueden definirse objetos de esa clase (p.e. *top*)
): Objeto ∈ Una y solo una clase estructural
mbiar la clase estructural de un objeto
bjeto puede estar asociado a varias clases auxiliares
dirse dinámicamente: Facilitan extensión de objetos
)=ES|AB; Superclase(AU)=AU|AB

Ejemplos de clases

- *top*: raíz; AB; ob: *objectClass*
- *person*: ↓*top;* ES; ob: *cn*, *sn*; op: *telephoneNumber*, ...
- *residentialPerson*: ↓*person;* ES; ob: *l;* op: *postalAddress*, ...
- *organization*: ↓*top;* ES; ob: *o;* op: *postalAddress*, ...
- *organizationalUnit*: ↓*top;* ES; ob: *ou;* op: *postalAddress*, ...
- *dcObject*: ↓*top;* AU; ob: *dc* (valor único)
- *device*: ↓*top;* ES; ob: *cn*; op: *serialNumber*, *o*, *ou*, *owner*, ...
- *groupOfNames*: ↓*top;* ES; ob: *cn*, *member*; op: *o*, *ou*, ...
- *alias*: ↓*top;* ES; ob: *aliasedObjectName*
- *referral*: ↓*top;* ES; ob: *ref*

e mi entrada en LDAP de FI

ata Interchange Format): protocolo LDAP es binario
← estructural (*top*→*person*→*organizationalPerson*→*inetOrgPerson*)
← auxiliar (*top*→ *posixAccount*)
← auxiliar (*top* → *irisPerson* → *fiPerson* → *fiEmployee*)
← auxiliar (*top* → *sambaSamAccount*)



Extracto de entrada FI en LDAP de FI

objectClass: dcObject ← auxiliar (*top* → *dcObject*)
objectClass: organization ← estructural (*top* → *organization*)
objectClass: labeledURIObject ← auxiliar (*top* → *labeledURIObject*)
dc: fi ← atributo específico de *dcObject*
o:: RmFjdWx0YWQgZGUgSW5mb3Jtw6F0aWNhIC0gVVBN
postalCode: 28660
l: Boadilla del Monte
st: Madrid
labeledURI: http://www.fi.upm.es ← atributo específico de *labeledURIObject*
telephoneNumber: +34 913367399

Decodificación de base 64
o: Facultad de Informática – UPM

odelo de nombres

- nombre: *Relative Distinguished Name* (RDN)
  - os de la entrada que la hacen única entre "hermanos"
  - ez (ej. múltiples: *cn=Fernando Perez Costoya+dni=76543210)*
- mbres (*Directory Information Tree*, DIT)
  - eto (*path*): *Distinguished Name* (DN)
    - ntrada + DN del padre (separados por comas)
    - *fperez,ou=personal,dc=fi,dc=upm,dc=es*
  - con jerarquía de clases
  - pero directorios también tienen información asociada
  - eto raíz (sufijo o base): a discreción
    - partir de dominio DNS usando clase auxiliar *dcObject*
    - *fi.upm.es → dn: dc=fi,dc=upm,dc=es*
  - gestiona 1 ó más DIT
  - lve metainformación en objetos/atrib. operacionales
  - dos por el servidor, esquemas soportados, ...

rama del DIT del LDAP de FI

Extracto de rama del DIT del LDAP de FI

```
# fi.upm.es
dn: dc=fi,dc=upm,dc=es
dc: fi
objectClass: dcObject
objectClass: organization
.....
# personal, fi.upm.es
dn: ou=personal,dc=fi,dc=upm,dc=es
ou: personal
objectClass: organizationalUnit

dn: uid=fperez,ou=personal,dc=fi,dc=upm,dc=es
uid: fperez
.....
```


Distribución y replicación

- Espacio de nombres distribuido usando *referrals*
  - Objeto en DIT especifica punto de montaje
  - No definido el modelo de navegación
    - Implementación más habitual iterativa
    - Aunque también recursiva (*chaining*)
- Replicación de espacio de nombres no definida por estándar
  - OpenLDAP admite dos esquemas:
    - Maestro-esclavo: asimétrico
    - Multi-maestro: simétrico
  - OpenLDAP no garantiza coherencia

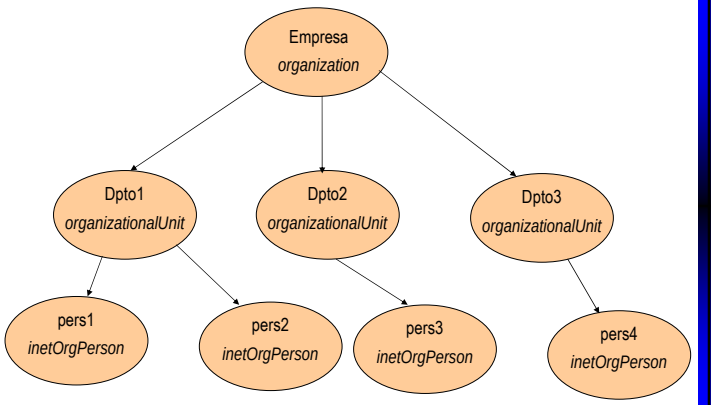
Diseño del DIT

ere experiencia
de info. del SD y cómo evolucionará
a evitar que cambios previstos en info. modifiquen DIT
ería afectar a atributos en vez de a estructura de DIT
co profundo
nde personal cambia de dpto. con frecuencia

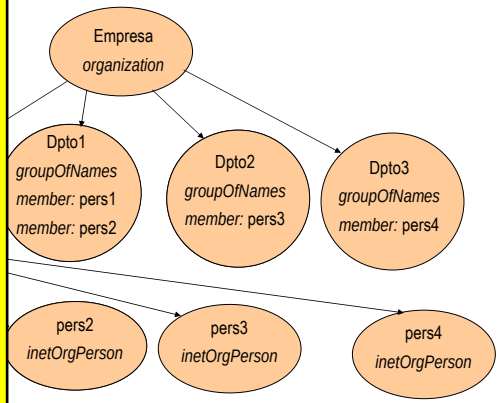
onalUnit/dpto. + 1 *inetOrgPerson*/persona
persona hija de entrada de su departamento

onalUnit para todo el personal + 1 *inetOrgPerson*/persona
ames/dpto. con 1 atributo *member*/persona
mbia de departamento: cambio atributos, no cambio DIT
ciertas búsquedas pueden ralentizarse

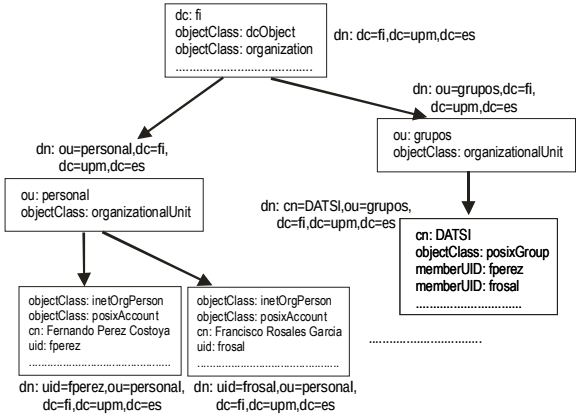
Diseño 1



Diseño 2



Extracto de jerarquía de LDAP de FI



eraciones de LDAP

necta y autentica/desconecta

una búsqueda basada en los parámetros:
búsqueda
entrada base, sólo hijos o todo el sub-árbol
eda
e devuelven (además, si valores o sólo tipos)
s alias o no durante la búsqueda
o y máximo nº de entradas retornadas

rueba si DN dado tiene un valor en atributo

ade/Elimina la entrada del DN dado

atributos (añade, elimina o cambia) de un DN

bia DN de una entrada
ólo cambia RDN final; mueve en DIT en caso contrario

61 Fernando Pérez Costoya

Acceso a operaciones de LDAP

- API de programación en C
  - *ldap_bind()*,*ldap_search()*,*ldap_add()*,*ldap_delete()*,*ldap_modify()*, ...
- Mandatos
  - *ldapsearch*, *ldapadd*, *ldapdelete*, *ldapmodify*, *ldapmodrdn*, ...
    - La mayoría usan el formato LDIF como entrada o salida
- Formato URL estándar para LDAP
  - *ldap://máquina:puerto/DNbase?atributos?ámbito?filtro*
    - *ldaps* si usa comunicación segura

Sistemas Distribuidos 62 Fernando Pérez Costoya

de búsquedas (en triqui)

nfo.fi.upm.es -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es'
al,dc=fi,dc=upm,dc=es'

esores que comparten un despacho dado

info.fi.upm.es -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es'
upm,dc=es' '(roomNumber=4201)' cn sn

nal de nombre Fernando y no sean del DATSI

info.fi.upm.es -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es'
upm,dc=es' '(&(!(departmentNumber=DATSI))(cn=*Fernando*))' cn

ciones" de la FI

nfo.fi.upm.es -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es'
s one '(objectClass=organizationalUnit)' ou

63 Fernando Pérez Costoya

Esquema

- Paquete que incluye definiciones en ASN.1 y que usan OIDs
- Esquema incluye varios tipos de definiciones:
  - *ldapsyntax*: Define tipos básicos de LDAP
  - *matchingRule*: Op. de comparación sobre tipos básicos
  - *attributetype*: Definición de atributo
  - *objectclass*: Definición de clase
  - *matchingRuleUse*: Para qué atributo se usa una regla de comparación
  - *dITContentRule*: qué clases auxiliares permitidas para una c. estruct.
  - *dITStructureRule*: qué clases pueden ser padres de una c. estructural
  - *nameForm*: qué atributos pueden usarse como RDN de c. estructural
- Se usa herencia tanto en defs. de clases como de atributos
- Hay esquemas estandarizados:
  - *core*, *cosine*, *inetorgperson*, *nis*, ...

Sistemas Distribuidos 64 Fernando Pérez Costoya

tipos de datos de LDAP

r, por interoperabilidad no deberían definirse nuevos tipos

.fi.upm.es -x -b cn=subschema -s base ldapsyntaxes

1.1466.115.121.1.44 DESC 'Printable String')
1.1466.115.121.1.11 DESC 'Country String')
1.1466.115.121.1.26 DESC 'IA5 String')
1.1466.115.121.1.40 DESC 'Octet String')
1.1466.115.121.1.41 DESC 'Postal Address')
1.1466.115.121.1.50 DESC 'Telephone Number')
1.1466.115.121.1.36 DESC 'Numeric String')
1.1466.115.121.1.27 DESC 'Integer')
1.1466.115.121.1.24 DESC 'Generalized Time')
1.1466.115.121.1.7 DESC 'Boolean')
1.1466.115.121.1.6 DESC 'Bit String')
.....

65 Fernando Pérez Costoya

Reglas de comparación de tipos

- Definidas por estándar, por interoperabilidad no deberían definirse nuevas reglas

ldapsearch -H ldaps://info.fi.upm.es -x -b cn=subschema -s base matchingRules

matchingRules: (2.5.13.4 NAME 'caseIgnoreSubstringsMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.58)
matchingRules: (2.5.13.2 NAME 'caseIgnoreMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)
matchingRules: (1.3.6.1.4.1.1466.109.114.3 NAME 'caseIgnoreIA5SubstringsMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
matchingRules: (1.3.6.1.4.1.1466.109.114.2 NAME 'caseIgnoreIA5Match' SYNTAX 1.3.6.1.4.1.1466.115.121.1.26)
matchingRules: (2.5.13.14 NAME 'integerMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.27)
matchingRules: (2.5.13.13 NAME 'booleanMatch' SYNTAX 1.3.6.1.4.1.1466.115.121.1.7)
.....

Sistemas Distribuidos 66 Fernando Pérez Costoya

inición de atributos

.fi.upm.es -x -b cn=subschema -s base attributetypes

AME 'name'
eMatch
ubstringsMatch
466.115.121.1.15{32768})

ME ('cn' 'commonName') SUP name)

9200300.100.1.25
Component')
7: domain component'
elA5Match
A5SubstringsMatch
466.115.121.1.26 SINGLE-VALUE)

67 Fernando Pérez Costoya

Definición de clases

ldapsearch -H ldaps://info.fi.upm.es -x -b cn=subschema -s base objectClasses

objectclass (2.5.6.0 NAME 'top' ABSTRACT MUST objectClass)

objectclass (2.5.6.6 NAME 'person' SUP top STRUCTURAL
MUST (sn $ cn)
MAY (userPassword $ telephoneNumber $ seeAlso $ description))

objectclass (2.5.6.7 NAME 'organizationalPerson' SUP person STRUCTURAL
MAY (title $ x121Address $ registeredAddress $ destinationIndicator $
preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifier $
telephoneNumber $ internationaliSDNNumber $
facsimileTelephoneNumber $ street $ postOfficeBox $ postalCode $
postalAddress $ physicalDeliveryOfficeName $ ou $ st $ l))

objectclass (1.3.6.1.4.1.1466.344 NAME 'dcObject'
DESC 'RFC2247: domain component object'
SUP top AUXILIARY MUST dc)

Sistemas Distribuidos 68 Fernando Pérez Costoya

eglas de comparación

```
.fi.upm.es -x  -b cn=subschema -s base matchingRulesUse

.1.4.1.1466.109.114.2 NAME 'caseIgnoreIA5Match' APPLIES (
S associatedDomain $ email $ aRecord $ mDRecord $ mXRecord
cord $ cNAMERecord $ janetMailbox $ gecos $ homeDir.... ) )
3.13 NAME 'booleanMatch' APPLIES ( hasSubordinates $
stMod $ olcReadOnly $ olcReverseLookup $ olcDbNoSync $
DbLinearIndex $ olcChainCacheURI $ olcChainReturnError $
olcDbChaseReferrals $ olcDbProxyWhoAmI $ olcDbSingleConn
yConn $ pwdLockout $ pwdMustChange $ pwdAllowUserChange
mbaBoolOption $ pwdReset $ olcPPolicyHashCleartext $
t $ olcSpNoPresent $ olcSpReloadHint ) )
```


e esquema del LDAP de FI

```
47.1.19.10.4.2.4 NAME 'fiRelationShip' DESC 'Relacion del usuario con
aseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch SYNTAX
1.1.15 )
47.1.19.10.4.2.1 NAME 'fiGender' DESC 'Sexo de  la persona (ISO
rMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
7547.1.19.10.4.2.5 NAME 'fiTeaching' DESC 'Asignaturas impartidas por
aseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch SYNTAX
1.1.15{20} )
7.4.3.1.2 NAME 'irisPerson' DESC 'Persons inside the IRIS community'
Y ( sn1 $ sn2 $ irisPersonalTitle $ irisPersonalUniqueID $
UserPrivateAttribute $ irisUserStatus $ irisMailHost $
irisMailbox $ irisMailMainAddress $ irisMailAlternateAddress $
ClassifCode ) )
7.1.19.10.4.1.1 NAME 'fiPerson' DESC 'Persona perteneciente a la
(UPM)' SUP irisPerson AUXILIARY MUST ( uid $ mail )
erTime $ fiMailQuotaSize $ fiGender $fiRelationShip ) )
7547.1.19.10.4.1.3 NAME 'fiEmployee' DESC 'Empleado de la Facultad
UP fiPerson AUXILIARY MAY fiTeaching)
```


Creación de un nuevo esquema

- Sólo si es estrictamente necesario
  - Nunca cambiar comportamiento de objetos/atrib. estándar
- 2 alternativas para extender clase ya existente
  - Crear nueva clase estructural derivada de clase existente
    - Permite mejor control: se pueden definir reglas de contenido/estructura
    - Pero requiere eliminar y reinsertar todos los objetos existentes
  - Crear clase auxiliar derivada de *top* e incluirla en definición de objetos
    - Se puede añadir directamente usando *Modify*
- C. auxiliar también permite incluir atrib. en objetos de ≠ clases
  - p.e. fecha de alta en organización, tanto personas como dispositivos

Modelo de seguridad

- 3 métodos de autenticación
  - Sin autenticación: se considera usuario anónimo
  - Autenticación básica: DN del usuario + contraseña
  - *Simple Authentication and Security Layer* (SASL)
    - Entorno genérico de autenticación y seguridad de datos
    - Permite usar múltiples mecanismos (p.e. SASL DIGEST-MD5)
    - SASL EXTERNAL: protocolo nivel inferior proporciona autenticación
      - Como cuando se usa *Transport Layer Security* (TLS)
- Protección de entradas no definida por el estándar
  - Habitualmente se usan listas de control de acceso (ACL)
    - Controlan acceso a cada atributo de una entrada

Índice

Gestión de nombres en SD convencional

- Estructura del SD bastante estática
- Cada nodo se configura con (suponiendo uso de IP)
  - Su dir. IP, máscara de red, dir. *router*, e info. de encaminamiento
  - Su nombre, dominio DNS al que pertenece y direc. servidores DNS
  - Nombre servidor(es) LDAP y conocimiento del esquema usado
  - Los manejadores requeridos para interacción con dispositivos en SD
- Incluso en SD convencional, op. configuración no escalable
  - Uso de DHCP (*Dynamic Host Configuration Protocol*)
- Necesidad limitada de "descubrimiento" de servicios/dispos.
  - Una vez instalada nueva impresora se la da de alta en LDAP
  - Próxima búsqueda de impresoras en LDAP la encontrará
  - Dar de baja impresora (poco frecuente): basta con actualizar LDAP
  - No se requiere "*Plug* & *Play*" en el nivel del SD

…nombres en SD móvil/ubicuo

…bicua → "invisible" → auto configuración
…icos, "espontáneos" y volátiles:
…y salen de un SD: de un "espacio inteligente" (EI)
…digital y yo entramos/salimos en habitación de hotel
… entra/sale de EI controlado por un semáforo inteligente
… de servicios clave para computación ubicua
…EI:
…ura y descubre, y es descubierto, por nodos restantes
…e servicios, hace conocerlos a quiénes le interesen
… de serv., descubre los de otros nodos que le interesen
…"lenguaje" de definición y búsqueda de servicios
…lores (similar a LDAP), basado en XML, ontologías, ...
…exibilidad para incorporar nuevos tipos de serv./dispos.

Gestión de nombres en SD móvil/ubicuo

- *Plug & play* de servicios/dispositivos en SD
  - Además de descubrirlos, hay que saber "hablar" con ellos
  - Nuevo tipo puede requerir nuevo manejador
- Nodo abandona EI: servicio/dispositivo desaparece
  - Abandono abrupto → uso de *leases*
- Problema de frontera del espacio inteligente:
  - Delimitación precisa de confines de un espacio inteligente
    - ¡Espero que mis fotos no se impriman en habitación contigua!
  - Necesidad de crear ámbitos (*scopes*)
- Limitación de recursos y volatilidad pueden condicionar:
  - Estrategias de auto-configuración y descubrimiento de servicios
- Ej. Jini, UPnP, Zeroconf, *Service Location Protocol* (RFC 2608)

lantilla de servicio de SLP

.tum.de:1020/queue1
administrator
J4050 N
m 0409
se
e-sided, two-sided

arison Of Service Discovery Protocols And Implementation Of The ocol", Christian Bettstetter y Christoph Renner

77 Fernando Pérez Costoya

Auto-configuración

- Obtención de dirección IP (e info asociada: máscara, *router*,...)
  - Uso de DHCP:
    - Nodo *broadcast* petición de dirección IP
    - Servidor DHCP asigna dirección IP con *lease* asociado
  - Si DHCP no disponible (por volatilidad o limitación de recursos)
    - *Dynamic Configuration of IPv4 Link-Local Addresses* (RFC 3927)
    - Nodo elige su dir. IP y usa ARP para comprobar que no está en uso
    - Si conflicto, selecciona otra
- Obtención de nombre DNS (si requerido)
  - Uso de DNS con protocolo de actualización: *Dynamic-DNS*
  - Si DNS no disponible (por volatilidad o limit. recursos): *Multicast*-DNS
    - Consultas a dominio `.local.` usan *multicast* dir. fija
    - Nodo correspondiente responde (con *unicast* o *multicast*)

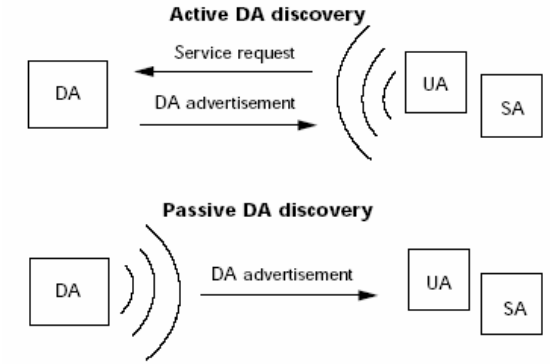
Sistemas Distribuidos 78 Fernando Pérez Costoya

descubrimiento de servicios

ndo terminología SLP):
roveedor servicio (SA), servidor descubrimiento (DA)
cipal: con o sin DA
últiples DA: replicación y/o info. de distintos ámbitos
PnP):
localizar DAs (posible filtro por ámbitos)
activa: UA/SA *multicast* a dirección fija
pasiva: DA *multicast* a dirección fija
ncorporación de nuevos DAs al sistema
den registrar servicios y UAs realizar consultas
vicio mediante *unicast* en DAs localizados
ediante *unicast* a alguno de los DAs localizados
r a DA notificación si aparece un tipo de SA → evento

79 Fernando Pérez Costoya

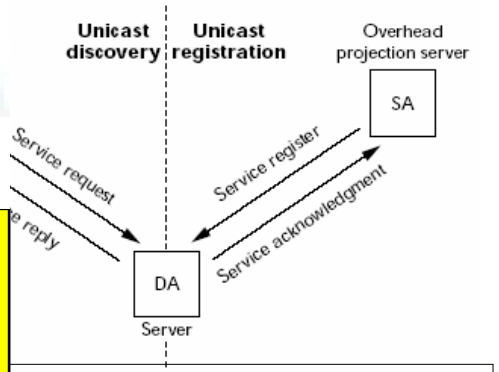
Descubrimiento de DA en SLP



Service Location Protocol: Automatic Discovery of IP Network Services. Erik Guttman

Sistemas Distribuidos 80 Fernando Pérez Costoya

úsqueda de servicio en SLP



ation Protocol: Automatic Discovery of IP Network Services. Erik Guttman

81 Fernando Pérez Costoya

Servicios de descubrimiento de servicios

- Sin DA: *pull* versus *push*
  - *Pull*: UA *multicast* petición; SA la recibe y responde
  - *Push*: SA *multicast* anuncio de servicio; UAs guardan esa info.
  - *push* descubrimiento automático nuevo SA; *pull* uso de *polling*
- Esquema híbrido (SLP)
  - Mientras no haya ningún DA (suponiendo modelo *pull* como SLP):
    - SA escucha dir. *multicast* peticiones de servicio
    - UA envía a dir. *multicast* peticiones de servicio
    - SAs/UAs escuchando dir. *multicast* posibles altas de DAs
  - Cuando aparece un DA no habiendo ninguno antes
    - SAs registra servicio en DA mediante *unicast*
    - UAs consultan DA usando *unicast*
  - Si desaparecen todos los DAs: vuelta al primer punto

Sistemas Distribuidos 82 Fernando Pérez Costoya

conf (Bonjour, Avahi)

utomática máquinas/servicios en red IP

rección IP

disponible, direcciones *Link-Local*

ombres de máquina

onal y, si no disponible, *Multicast*-DNS

de servicios:

→ DNS-SD: una extensión de DNS

y TXT para especificación de servicios

para detectar servidor ya no presente

eries para solicitar notificación si aparece nuevo servicio

83 Fernando Pérez Costoya