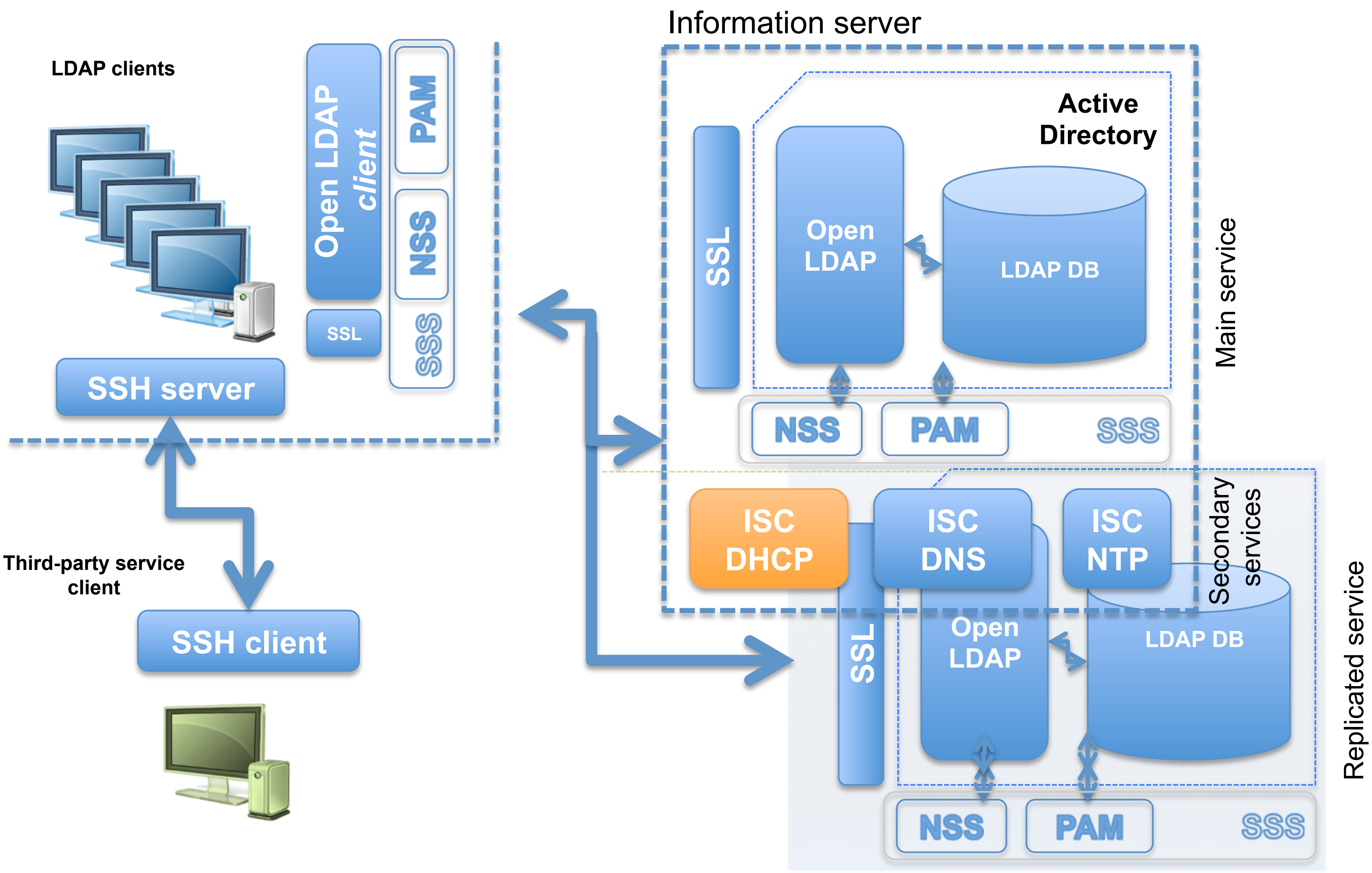
Computer System Design and Administration

Topic 4. Network configuration service: ISC DHCP

José Ángel Herrero Velasco

Department of Computer and
Electrical Engineering

Secure information service: Puzzle



Target: … server convergence

- Installation, configuration and deployment of ***third-party* network services** for local *networking* management on the INTRANET:
  - Dynamic configuration service (**DHCP**): *ISC dhcpd:*
    - **Dynamic host configuration** of network parameters in local hosts.
  - Domain name service (DNS): *ISC bind9.*
  - Network time service (NTP): *ISC ntpd.*

DHCP: Dynamic Host Configuration Protocol

- It allows hosts from a TCP/IP network to "***lease***" their network and administrative configuration:
  - Hosts don't need to know that configuration previously.
- It is suitable for *"dynamic"* environments (ISPs):
  - When a connected host boots, DHCP automatically assign (**rents**) a *full* network configuration:
    - This can be **reused** by other hosts when this is *off-line.*
- It is suitable for *"static"* environments too (LANs):
  - **Centralized** network configuration:
    - It simplifies the global network configuration.
    - It makes the system administrator's life easier.
- It is an evolution of **BOOTP** (67/UDP port):
  - Initially it was deployed to boot *diskless* UNIX hosts:
    - In this case, DHCP service should send to clients a full network configuration:
      - **Network configuration and kernel + initrd (boot SO ramdisk) included.**
    - DHCP service should provide everything.
  - **DHCP** can operate with **BOOTP.**

DHCP: “Leased” networking parameters

- **IP address** and **netmasks.**
- **DNS** name servers.
- **NTP** servers.
- **Gateways** (default network routes).

- Remote **Syslog** servers.
- WINS, proxy and X Servers (if applicable).
- **TFTP (+ PXE)** network boot servers:
  - Diskless boot.
- … There are dozens more (RFC2131/2):
  - http://www.rfc-base.org/rfc-2132.html.

DHCP: Parameter assignment

- The leased parameters (*lease*) **must be renewed** by client hosts:
  - **Periodically** (when *lease time is half over).*
  - If **lease time** is over and the *lease* is not renewed:
    - The lease expires → DHCP server "removes" them!!!
    - Service is free to be reused by other clients.
  - Lease time is configurable:
    - From hours to days... endless even.

- Service can assign network parameters in 2 ways:
  - **Dynamically →**
    - Regardless of who the client is:

      → *floating* IP.
  - **Statically →**
    - Settings are pre-assigned for each client:
      - **Uniquely.**
      - **Through interface MAC address.**

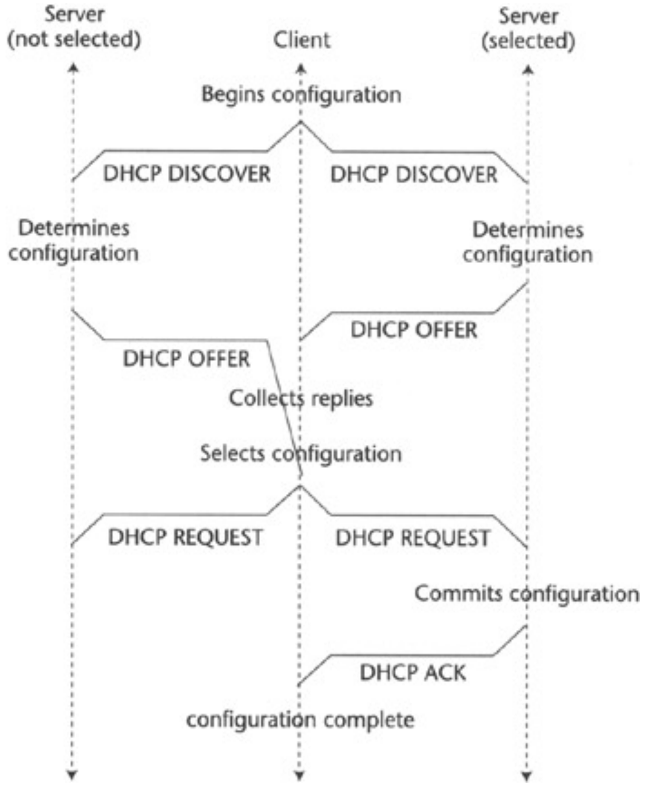
- More than one operative DHCP service could even exist in a LAN:
  - Conflicts???

DHCP: Operation

CLI

- When client-host boots, it sends a **DISCOVER** message → DHCPDISCOVER:
  - In *broadcast* mode (IPv4):
    - To the whole network.
  - This message contains client data, such as:
    - MAC address.
    - …

(**) → **DISCOVER** message can be relayed out of its subnet, using a **“relay agent”…**



DHCP: Operation

SERV

- The DHCP Server(s) responds with an **OFFER** message → `DHCPOFFER`:
  - It contains the **IP address** and other network parameters:
    - If there were more than two DHCP servers running on the subnet, any of them could answer the client request simultaneously:
      - The client takes:
        - » The first reply.
        - » Preset.

CLI

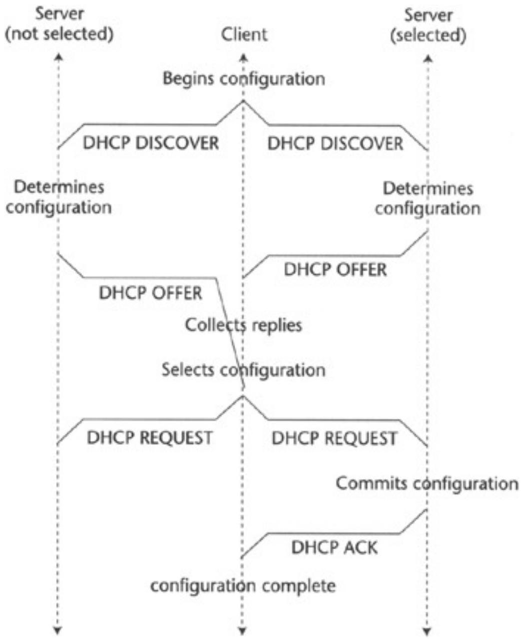
- The client replies to DHCP server with a **REQUEST** message → `DHCPREQUEST`:
  - In *broadcast* mode (IPV4):
    - The message contains the *“winner”* DHCP server data.

SERV

- The DHCP server responds with an **ACK** message → `DHCPACK`:
  - … and **“blocks”** the assigned IP address.

CLI

- Before using it, the client **checks** the IP address:
  - That is, if it is not in use!!:
    - According to its **ARP** table.
  - If it is, the client responds with a **DECLINE** message → `DHCPDECLINE`:
    - The dialog **restarts**!!!



Server (not selected)

Client

Server (selected)

Begins configuration

DHCP DISCOVER

DHCP DISCOVER

Determines configuration

Determines configuration

DHCP OFFER

DHCP OFFER

Collects replies

Selects configuration

DHCP REQUEST

DHCP REQUEST

Commits configuration

DHCP ACK

configuration complete

DHCP: Operation

- You can see that “talk” between DHCP server and client in the `log` files:

```
server-01 :~# tail /var/log/syslog
…
Jan 15 11:39:38 server-01 dhcpd: DHCPDISCOVER from 00:0c:29:5f:e6:8d via eth0
Jan 15 11:39:38 server-01 dhcpd: DHCPOFFER on 192.168.155.201 to 00:0c:29:5f:e6:8d via eth0
Jan 15 11:39:38 server-01 dhcpd: Dynamic and static leases present for 192.168.155.201.
Jan 15 11:39:38 server-01 dhcpd: Remove host declaration localdomain or remove 192.168.155.201
Jan 15 11:39:38 server-01 dhcpd: from the dynamic address pool for 192.168.155/24
Jan 15 11:39:38 server-01 dhcpd: DHCPREQUEST for 192.168.155.201 (192.168.155.101)
      from 00:0c:29:5f:e6:8d via eth0
Jan 15 11:39:38 server-01 dhcpd: DHCPACK on 192.168.155.201 to 00:0c:29:5f:e6:8d via eth0
…
server-01 :~#
```

DHCP: Operation

- Network data **expiration:**
  - The network configuration has a defined **"time to live"** parameter:
    - **"Lease Time".**
  - When this value is half over, the client attempts to **renew its lease:**
    - Sends a **REQUEST…**
    - If it doesn't do so, DHCP server will revoke the client network configuration.
- Client runs a *Graceful shutdown:*
  - When client host shutdowns, it sends a **RELEASE** message to notify the server that its network configuration must be discarded.
- The DHCP server is obliged to keep track of the configurations:
  - Keep the same IP address for the client.
  - Even if client *reboots.*
- If DHCP server fails **and…:**
  - "Lease time" is running out or client reboots:

    →Clients won't be able to connect again.

  → Unless we have **dhcpclient** properly configured. ☺

DHCP: Service installation (ISC DHCP)

- DHCP from www.ISC.org:
  - The most stable version for DHCP servers.

Source: www.isc.org.

- In debian, by default...:
  - Installation from **sources** (**):

```
$ wget ftp.isc.org…
$ ./configure; make; make install.
```

  - Installation from DEBIAN repository (mirrors):

Server

```
$ apt-get install isc-dhcp-server.
```

Client

```
$ apt-get install isc-dhcp-client.
```

  - Checking:

```
$ vi /etc/dhcp/dhcpd.conf.
$ cat /var/lib/dhcp/dhcpd.leases (log).
```

→ ** Debian repositories usually have older versions (but more stable).

DHCP: Daemon and service configuration

- **Server (*daemon*)** configuration:

```
$ vi /etc/default/isc-dhcp-server:
```

  - `DHCPd` *daemon* relative options:
    - `DHCPD_CONF`: main configuration file for DHCP service.
    - `OPTIONS`: secondary *daemon* options.
    - `INTERFACES`: Ethernet interfaces which `DHCPd` will operate.

- **Service** configuration:

```
$ vi /etc/dhcp/dhcpd.conf:
```

  - This file is very syntax sensitive (as the rest of the config file… 🙂):
    - If an (syntax) error exists, the service doesn't start.
  - When this file is modified, we must restart DHCP service.

- **Options:**
  - Domain name which DHCP service will manage:
    - **`option domain-name`** `“domain name”.`
  - Maximum and initial *“lease time”* for network parameters:
    - **`max-lease-time`** `24000.`
    - **`default-lease-time`** `3600.`

DHCP: Service configuration

– Network parameters for all clients: **netmask**, **gateways**, **DNS servers**, etc.:

- `option domain-name-servers <IP1>, <IP2>, …`
- `option routers <router IP>.`
- `option subnet-mask <Network mask IP>.`
- `option broadcast-address <broadcast IP>.`

– Subnets managed:

- Defined by address ranges: (from… to…):
  - *Dynamic* **(floating IPs).**
  - *Static* **(according to client MAC address).**
```
subnet 192.168.0.0 netmask 255.255.255.0
 {range 192.168.0.20 192.168.0.30; }
```

– Both within and outside the subnet definitions, we can define **hosts** and host **groups:**

- These host definitions enable static network parameters for each host (or host group).
```
group {
    <global parameter for every host in the group>

    host <hostname (FQDN)> {
           <specific network parameters>
    }
}
```

DHCP: Service configuration

- **TFTP/BOOTP** configuration:
  - That configuration will be useful for booting **diskless** (or not installed) hosts:
    - **Header** parameters:

      ```
      allow booting;
      allow bootp;
      ```

    - **Global** parameters:

      ```
      option imageserver  code 140 = text;
      option imageserver  "<IP servidor systemimager>";
      ```

    - **Particular** (group and **host)** parameters:

      - Both can be used jointly -

      ```
      – next-server <IP servidor systemimager>;
      – filename "pxelinux.0";
      ```

Network **boot loader** image:
located on system images server → `/tftpboot/`
…

DHCP: Service configuration

/etc/dhcp/dhcpd.conf

Sample

```
# Overall config options
allow booting;
allow bootp;
option domain-name “localdomain”;
option domain-name-servers 192.168.0.11, 193.194.193.22;
option subnet-mask 255.255.255.0;
max-lease-time 7200;

# Dynamic IP range (Floating Ips - dynamic assignment)
subnet 192.168.0.0 netmask 255.255.255.0{
    range 192.168.0.100 192.168.0.120;}

# Static IP range (assigned according to the client MAC address)
subnet 192.168.0.0 netmask 255.255.255.0{
    range 192.168.0.20 192.168.0.40;
    option broadcast-address 192.168.0.255;
    option routers 192.168.0.2;
    host client {
         hardware ethernet 08:00:07:12:34:56;
         fixed-address 192.168.0.25;
    }
…
}
```

DHCP: Is it Flexible? Is it Safe?

- **Is it flexible?:**
  - It allows a **centralized** network management:
    - **Dynamic (floating) hosts:**
      - **Laptops, temporal *ad-hoc networks, guest hosts.***
    - **Static (permanent) hosts:**
      - **@MAC.**
  - Any network change that occurs will be easily solved:
    - For instance, any change on router or DNS IP.
    - All my network configuration resides in a single file.
- **Is it safe?:**
  - Initially, when we assign *static* parameters (IP) to clients, we know exactly what host it is for:
    - We can keep control.
  - But…:
    - MAC address is recorded in a ROM of the network interface.
    - → **Impossible to modify** ??!?!??!:
      - **It doesn't prevent *"unauthorized"* hosts using our subnet:**
        - » They can assign a MAC address themselves ← MAC spoofing.
  - If a single running DHCP service fails, our host clients become *network-less:*
    - So, at least it is very important to have *dhcpclient service* on clients or more DHCP servers.
  - It doesn't include any security mechanism (by default):
    - Typical attacks (… and very dangerous!!):
      - **Authentication:**
        - » Unauthorized DHCP servers providing false information to clients.
        - » Unauthorized clients gaining access to resources.
      - **Attacks:**
        - » **DHCP man in the middle:**
          - **ARP spoofing.**
          - **MAC/IP spoofing.**
        - » **DHCP starvation:**
          - Unauthorized DHCP server attacks.
          - → Deny of Service (DoS) from client side.
    - Authentication mechanism → RFC 3046, RFC 3118, EAPoDHCP…

Attention!!
One Point of Failure
(PoF)