DHCP and DNS

Nick Urbanik <nicku@nicku.org>

Copyright Conditions: GNU FDL (see http://www.gnu.org/licenses/fdl.html)

A computing department
Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS)

Organising computers in a large network
Reference books:

*The DHCP Handbook*, Ralph Droms & Ted Lemon, 2nd edition,

*DNS and Bind*, Paul Albitz and Cricket Liu, 4th edition
DHCP: Why?

- Manually assigning IP addresses (the alternative to DHCP) causes:
  - More work to set up
  - Much more work to change
  - IP address conflicts
  - Unsatisfied users who configure their own machines to cause more conflicts
DHCP: Why not?

- Last year, on many Tuesday afternoons, our laboratories were disrupted by “network failure”
- This was caused by project students running DHCP servers on our network,
- ...and also, by a small router running a DHCP server accidentally plugged into our campus network
- Solution: when detect this, run Ethereal listening on ports 67 and 68
- identify culprit, and turn off rogue server
What can DHCP do?

- Current standard DHCP servers can:
  - Allocate all IP parameters
  - Divide hosts into classes, based on many criteria, such as:
    - Manufacturer
    - Explicitly putting individual machines into different classes
    - Whether the machine is registered
  - Offer different parameters to machines in different classes
  - Dynamically update DNS servers
  - Support a DHCP failover protocol
Internet Software Consortium: ISC DHCP

- ISC makes *reference implementations* of DNS, DHCP
- Implemented by people directly involved with the standardisation process
- Provide the most standards compliant, most feature-rich implementations
- ISC DHCP server very robust
  - See experience with Tsing Yi Computer Centre
Experience at Tsing Yi CC

- At Tsing Yi Computer Centre:
  - Computer Centre in TY used MS DHCP on NT 4
  - Crashed twice, with complete loss of database containing MAC addresses of all computers on campus
  - Out of action for two days at a time, long sessions of manual retyping of all the data again
  - Replaced with system based on ISC DHCP server on a 486
  - Has worked well ever since (no down time)
Characteristics of DHCP

- All communication initiated by the client
- Uses UDP on port 68 for client, port 67 for server
- One DHCP session has a common xid ("transaction ID" in Ethereal), randomly selected by the client
- Uses unicast when client has IP address, [and client is not in REBINDING state — see later; broadcast otherwise
- Addresses offered from
  - address pools, or
  - Fixed addresses allocated to particular computers
Leases

- Server offers IP address and network parameters for a limited time (called a *lease*).
- In practice, leases may vary from 30 minutes to a week or so.
- Short lease:
  - Clients get updated parameters quickly.
  - Essential if you have more clients than addresses.
  - Requires more processing power on the server.
- Long lease:
  - More reliable (clients may continue to operate for a week after DHCP server fails).
  - But takes longer for all clients to get new settings if they change.
(Some) Standards for DHCP

- RFC 2131 — Basic DHCP operation
  - excerpts from this appear in exams!
- RFC 2132 — DHCP options: a list of the kinds of things a client can ask a DHCP server for

IETF Drafts:
- draft-ietf-dhc-authentication-14.txt
  - supports authentication between clients and servers
- draft-ietf-dhc-dhcp-dns-12.txt
  - interaction between DHCP and DNS servers
- draft-ietf-dhc-failover-07.txt
  - supports failover between 2 DHCP servers
DHCP Messages — 1

- **DHCPDISCOVER** — from *client*
  - client has no address, asking for a new one

- **DHCPOFFER** — from *server*
  - Offer of address and other parameters

- **DHCPREQUEST** — from *client*
  - Client asks if can use the offered address and parameters

- **DHCPACK** — from *server*
  - Server says “yes, go ahead, this address and these parameters are yours; the lease starts now.”
DHCP Messages — 2

- **DHCPNAK** — from server
  - “no, you may not have that address; go to the INIT state”

- **DHCPDECLINE** — from client
  - Client has detected another machine is using the offered address, and tells the server about this problem

- **DHCPRELEASE** — from client
  - Server expires the lease immediately

- **DHCPINFORM** — from client
  - Client already has an IP address, but wants other network settings from the server
State Diagram for DHCP protocol

See page 34 of RFC 2131 for a more complete state diagram.
DHCP Client States — 1

- **INIT-REBOOT**
  - Boot before lease expires

- **INIT**
  - Boot after lease expires
  - No response from server, lease expired

- **SELECTING**
  - Has been offered an address
  - DHCPDISCOVER/DHCP OFFER broadcast

- **BOUND**
  - Has IP address
  - DHCPREQUEST/DHCPACK broadcast
  - DHCPREQUEST/DHCPACK unicast

- **RENEWING**
  - At T1, renew using unicasts
  - DHCPREQUEST/DHCPACK broadcast
  - DHCPREQUEST/DHCPACK unicast

- **REBINDING**
  - At T2, begin broadcasting requests to all DHCP servers
  - DHCPREQUEST broadcast
  - DHCPACK unicast
  - DHCPNAK

Client States:
- INIT-REBOOT
- INIT
- SELECTING
- BOUND
- RENEWING
- REBINDING

Server Responses:
- DHCPREQUEST/DHCPACK
- DHCPACK
- DHCPNAK
DHCP Client States — 2

- **INIT** (client is booting)
  - no IP address yet.
  - next message from client will be a broadcast **DHCPDISCOVER**.

- **INIT-REBOOT** (has unexpired lease)
  - has IP address, but is not using it
  - client will next broadcast **DHCPREQUEST**
  - Will move to **BIND** state if no response

- **SELECTING** (has received at least one **DHCPOFFER**)
  - Waiting for any other **DHCPOFFERS**
 DHCP Client States — 3

- **BOUND** (Client has an address)
  - Initiated by client receiving **DHCPACK** to **DHCPREQUEST**
  - Send no more messages until $T_1$ (renewal time, configured in client by the server)

- **RENEWING** (client has reached **renewal time** $T_1$ in **BOUND** state)
  - Client *unicasts* **DHCPREQUEST** to server
  - Server *unicasts* **DHCPACK** to client
  - $T_1 = \text{lease time}/2$
DHCP Client States — 4

- **REBINDING** (client has reached *rebinding time* $T_2$ without *DHCPACK* from server)
  - client broadcasts *DHCPREQUEST*
  - client is looking for another server
  - $T_2 = \text{lease time} \times \frac{7}{8}$
  - If lease expires, client goes back to *INIT* state
  - Any network connections lost—bad for users!! Don’t let it happen to them!
The client is booting, with no IP lease
Confirming an IP Address when restarting

The client’s lease has not expired

DHCPREQUEST

DHCPACK

server

time
Extending a lease

- Lease is extended at $T_1$ before expires
- Unicast, because address is valid
- only case of unicast in DHCP protocol
- $T_1 = \text{leasetime} / 2$
Moving a computer to new subnet

- Refuse old address, issue a new one
Problems on the Network

- Often a computer has a bad configuration
- Faulty hardware may also cause excessive resending of bad packets
- Less often, a person may be doing something naughty on purpose!

Need some way to:
- track the location of a computer on the network
- determine if a computer is managed by the company or is a notebook brought in by a visitor

Want some way to register company machines
Ways of using DHCP

- There are two fundamentally different ways of using DHCP
- Typified by implementation in Campus, and ICT (till last week)
  - (both implemented by Nick!)
  - Fixed addresses for registered clients (Campus network)
  - Dynamic addresses for all comers (ICT till recently)
- Better: can provide automatic registration for clients: see chapter 20 of The DHCP Handbook
/etc/dhcppd.conf

- This plain text configuration controls behaviour of ISC DHCP server
- ISC DHCP server supports conditional statements, switch statements, substring expressions
- Almost a complete programming language!
- This text file can be generated by software (Perl programs often used)
**dhcpd.leases**

- This plain text file is generated by the DHCP server.
- Can be parsed by a Perl program.
- Can be used to determine the MAC address of an unregistered computer.
Advantage of Text Configuration

- Text can be easily generated by a program
- Can be easily checked by a human
- Microsoft DHCP server configuration and lease information is in an undocumented binary format
- reduces what can be done with it
- makes it hard to enter large amounts of information about many computers
  - experience at Tsing Yi Computer Centre
Host Records with Fixed Address

Can specify a fixed address for particular hosts:

```
# Machine type = COMPAQ DESKPRO  Laboratory = A204c
host a204c-03 {
    hardware ethernet 00:01:03:44:1d:62;
    fixed-address 172.19.80.003;
}

# Machine type = COMPAQ DESKPRO  Laboratory = A204c
host a204c-04 {
    hardware ethernet 00:01:03:45:2d:8f;
    fixed-address 172.19.80.004;
}
```

Can generate these with a Perl program
Method used by Computer Centre

- Uses Samba, ISC DHCP
- Documented on our web site; see the link to “DHCP and DNS System”
  
  http://nicku.org/snm/dhcp-dns-system/
Method Currently used by ICT

- Fixed DHCP and DNS records generated from an Excel spreadsheet
- Same as older method used by Computer Centre
- ...but also use the Perl module Spreadsheet::ParseExcel, which can read an Excel Spreadsheet directly — see parse-excel.pl at the URL in slide 28
- Generates DNS records also, using h2n
h2n—not a bird flu

According to [http://www.menandmice.com/6000/61_recent_survey.html](http://www.menandmice.com/6000/61_recent_survey.html), 68% of DNS servers in .com domain are misconfigured.

System administrators can make many mistakes

Best to generate DNS resource records with a program rather than by hand


input: a file in host table format (of /etc/hosts)

output is all the resource records, and DNS server configuration file.
cron job runs every 2 minutes, and does the following:

if ⟨excel spreadsheet⟩ is newer than /etc/dhcpd.conf
  parse ⟨excel spreadsheet⟩ into a ⟨hostfile⟩
  append any other required host files to this ⟨hostfile⟩
  if generate /tmp/dhcpd.conf from ⟨hostfile⟩
    move /tmp/dhcpd.conf to /etc/dhcpd.conf
    restart DHCP server
  ensure ⟨excel spreadsheet⟩ is not newer than /etc/dhcpd.conf
  stop DNS server
  wait for it to stop
  generate DNS resource records from ⟨hostfile⟩
  remove DNS journal files
  start DNS server
Older method used in ICT: free for all!

- Each client is offered:
  - an address in range 172.19.123.1 to 172.19.127.200
  - netmask /18
  - default gateway 172.19.127.254
  - domain name, tyict.vtc.edu.hk
  - name servers 172.19.64.52, 202.40.209.220
  - WINS servers 192.168.68.240, 202.20.100.226
  - NTP server ntp.tyict.vtc.edu.hk
  - a lease of 2 hours \((2 = 7200 \text{ seconds}/3600)\)
- The DHCP server attempts to create a DNS record for the client
- A separate log file will be created (see man syslog)
Older method used in ICT: free for all!

authoritative;
log-facility local1;

option domain-name "tyict.vtc.edu.hk";
ddns-update-style interim;
option netbios-name-servers 192.168.68.240, 202.20.100.226;
option domain-name-servers 172.19.64.52, 202.40.209.220;
option ntp-servers ntp.tyict.vtc.edu.hk;
subnet 172.19.64.0 netmask 255.255.192.0 {
    option routers 172.19.127.254;
    max-lease-time 7200;
    default-lease-time 7200;
    range 172.19.123.1 172.19.127.200;
}
Troubleshooting DHCP 1

- Our major problem: unauthorised DHCP servers giving **DHCPNAK** to all requests

- **Solution:** use **ethereal** in promiscuous mode with filter **port 67 or port 68**

- Examine packets from rogue server

- Use **xnmap** to gather more information about the rogue server

- Now go and talk with the person responsible
Troubleshooting DHCP 2

- Other problems:
  - Examine the DHCP server log using `tail -f` shows all DHCP messages received and sent by the server
  - Examine log on the client
  - Use `tcpdump` or `ethereal` to collect data analyse it in Ethereal
  - Compare with the `client state diagram`
  - Compare with normal, expected behaviour
Automatic Client Registration

Making it easy for customers to register their computers

Avoiding manual misconfigured settings
Automatic Client Registration

- It is good to be able to map IP addresses to particular computers (and users)

- Often computers cause trouble without the user being aware
  - e.g., project students with rogue DHCP servers

- Want convenience for user and sysadmin

- Can use the ISC DHCP server to implement such an automatic registration system.

- Depends on dividing IP hosts into two classes: known and unknown.
The file /etc/dhcpd.conf controls the behaviour of the ISC DHCP server.

It may be edited by external programs and host statements may be added:

Examples:

```plaintext
host a204-16 {
    hardware ethernet 00:08:02:1d:87:72;
}
host a204-17 {
    hardware ethernet 00:08:02:1d:87:02;
}
host a204-18 {
    hardware ethernet 00:08:02:1c:1c:43;
}
```
Known and unknown hosts

A host is *known* if it has a host declaration

```
subnet 172.19.64.0 netmask 255.255.192.0 {
    option routers 172.19.127.254;

    # Unknown clients get this pool.
    pool {
        option domain-name-servers bogus.tyict.vtc.edu.hk;
        max-lease-time 120;
        range 172.19.120.0 172.19.122.255;
        allow unknown clients;
    }

    # Known clients get this pool.
    pool {
        option domain-name-servers ns.tyict.vtc.edu.hk;
        max-lease-time 28800;
        range range 172.19.123.1 172.19.127.200;
        deny unknown clients;
    }
```

OSSI—DHCP and DNS – p. 39/44
Known and unknown hosts

- So the hosts a204-16, a204-17 and a204-18 get full parameters
- Others (without a hosts declaration) get
  - a short lease
  - a bogus name server that redirects all web access to a registration server
- Block the IP addresses from unknown hosts at the firewall
- they get no Internet access
- users are motivated to register
The registration server

- All unregistered hosts get a “bogus” name server that maps all hostnames to itself
- The web browser will go to the registration application, no matter URL entered
- Registration application edits `/etc/dhcpd.conf` on DHCP server
- Adds the host as a *known host*
- Gets the information from the DHCP lease
- User just needs to enter their user name and LDAP password
Registration Application

- A web application
- User interface is very simple — enter only:
  - user name
  - password
- Application knows IP address from web server
- Looks up MAC address from DHCP leases file
- Edits `/etc/dhcpd.conf`, adds a host record
- Can assign a fixed or dynamic address
Registered computer

- Now the client can either reboot, or wait 60 seconds to $T1$, and get a long term lease
- The machine becomes a “known host”
- Client can now access Internet conveniently
- Could extend this by adding MAC address to access control list of the appropriate port on the main switch
- Unregistered computers blocked by switch
- Enforces limiting access to registered computers only
Fixed or Dynamic Addresses

Would it be better to have known host records for registered computers and *dynamic addresses*, registered *dynamically* with the DNS server. . .

Or is it better to have *fixed addresses* and *fixed DNS records*?

I think that dynamic updates to DNS provide no additional benefit, and simply make the system more complex.

I recommend making the system as simple as possible

  - both for the system administrator and the users
  - . . . but no simpler.