Deploying DNS and DHCP Services
Session NCM-240

Why Should I Care?

• If DNS has problems, most users will attribute them to the network
• Want to understand the users’ view of the network
• Addresses are connected to both the network topology and names
• Administration of addresses and names is shared between host and network managers
Agenda

- Addresses
  - Hierarchy and topology
  - Assignment and reliability
- Naming
  - Structure
  - Protocol
  - Reliable operation
  - New things

Deploying DNS and DHCP Services

Addresses
Address Fundamentals

- IPv4 address 32 bits (e.g. 128.9.0.33)
- Dotted decimal ~ 8-bit fields
- IPv6 128 bits ~ colon-separated hexadecimal 16-bit fields

Address Hierarchy

- Best if topology matches hierarchy
Subnet Mask

Address 128.9.0.33

Mask 255.255.255.0

- Mask separates network (1) from host (0) part of the address
- Prefix (longest match) routing ~ contiguous “1” bits to the left

Subnets

- Each range of addresses for hosts defines a subnet e.g. 128.9.0.0/24
- Within the subnet, hosts communicate directly, using layer 2
- Special meaning for certain addresses
  - all-ones—broadcast
  - all-zero—network
Special Addresses

• Multicast
  - IPv4 ~ 224-239.d.d.d [RFC 2365]
  - IPv6 ~ FFxx:x:x:x:x:x:x

• Any-cast [RFC 1546]
  looks like unicast, but with multiple advertisers

• Link-local

Packet Filtering

• Why?
  - Security—blocking traffic
  - Selecting traffic for tunneling and differential service

• Facilitate filtering
  - Shorten ACLs
  - Align address ranges on bit boundaries
Address Translation

- Where addresses are not valid either for privacy (not safe) or because they are not unique
- Pool of addresses (ports) is assigned to “internal” in the translator when sessions initiate from “inside”
- RFC 3022

NAT, PAT, and Dynamic NAT

- **Static NAT**: Permanent—1 to 1
  - Permanent Mappings between Internal Servers to External Addresses
- **Dynamic NAT**: Dynamic—1 to 1
  - Pool of External Addresses Dynamically Assigned to Internal Clients for Duration of Session
- **PAT**: Dynamic—Many to 1
  - Multiple Internal Clients Share Single External Address
DHCP Bootstrapping

How It Works

• How to get configuration before an address enables communication?
• Local broadcast
• Relay agent

Dynamic Host Configuration Protocol (DHCP)

Local Broadcast to Acquire a Lease

• Allocate address from a range
• Address can be tied to MAC (layer 2)
• Subnet mask
• Default route
• DNS server address
How DHCP Works

Obtaining a Lease

- Dynamically assigns configuration information
- Creates IP address pools to conserve addresses and support mobile users
- Clients broadcast DHCP DISCOVER packet on local subnet
- Multiple servers can respond
- Client chooses first or best response

Here is your configuration:
IP Address: 192.204.18.7
Subnet Mask: 255.255.255.0
Default Routers: 192.204.18.1, 192.204.18.3
DNS Servers: 192.204.18.8, 192.204.18.9
Lease Time: 5 days

How DHCP Works

DHCP Discover Process

- DHCP client broadcasts DHCP DISCOVER packet on local subnet
- DHCP servers send OFFER packet with lease information
- DHCP client selects lease and broadcasts DHCP REQUEST packet
- Selected DHCP server sends DHCP ACK packet
DHCP Relay

- DHCP clients broadcast a DHCP discover packet
- DHCP relay (ip helper address) on the router hears the DHCP Discover packet and forwards (unicast) the packet to the DHCP server
- DHCP relay fills in the GIADDR field with IP address of the primary interface of router
- DHCP relay can be configured to forward the packet to multiple DHCP servers. Client will choose the “best” server
- DHCP servers use GIADDR field of DHCP Discover packet as an index into the list of address pools

DHCP Attributes for Applications

- Server location [RFC 2610]
- Novell directory services [RFC 2241]
- Time, NIS, TCP and IP parameters… [RFC 2131]
DHCP Reliability

- **Failover**
  
  Draft based on our design
  
  Coordinating the name in the DNS uses an identifier stored in DNS

- **Loadsharing**
  
  Servers answer only for configured hash (MAC)
  
  RFC 3074

---

DHCP Safe Failover Protocol

- All DHCP requests are sent to both servers
- Primary updates backup with lease information
- Backup takes over when primary fails
- Backup server uses dedicated pool of addresses allocated by the primary to prevent duplicate IP address
- Servers synchronize when primary is up
- IETF Internet Draft
## How DHCP Works DHCP Packet

<table>
<thead>
<tr>
<th>OP Code</th>
<th>Hardware Type</th>
<th>Hardware Length</th>
<th>HOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Transaction ID (XID)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Seconds</td>
<td>Flags</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Client IP Address (CIADDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Your IP Address (YIADDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server IP Address (SIADDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gateway IP Address (GIADDR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Client Hardware Address (CHADDR)—16 bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Server Name (SNAME)—64 bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Filename—128 bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DHCP Options</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Summary

- DHCP
- Questions?
Deploying DNS and DHCP Services

Naming

Domain Name Service

- DNS is a database
- Distinct features
  - Only look-up queries
  - Replicated content
  - Distributed control (zones)
Name Hierarchy

- Independent of address hierarchy
- Names length not limited by address size

Terminology

- Label
- Resource record
- Value
### Record Format

<table>
<thead>
<tr>
<th>Label</th>
<th>RR-type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAXA.ISI.EDU.</td>
<td>IN</td>
<td>A 10.2.0.27</td>
</tr>
<tr>
<td>VAXA.ISI.EDU.</td>
<td>IN</td>
<td>A 128.9.0.33</td>
</tr>
</tbody>
</table>

Optional fields:
We only care about class = IN
ttl will get attention later

### Address Examples

<table>
<thead>
<tr>
<th>Label</th>
<th>RR-type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAXA.ISI.EDU.</td>
<td>A</td>
<td>10.2.0.27</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>128.9.0.33</td>
</tr>
</tbody>
</table>

Empty label is the same as above
Address and Canonical Name

- ‘A’ resource record (RR)
- ‘CNAME’ RR
  The value of a canonical name is ALWAYS the label of an A record
- Service alias using CNAME [RFC 2219] Best Current Practice

Delegation Zone

- Hierarchical name space
- Each node in tree represents domain/subdomain
- Some subdomains are defined as zones
- Each zone has a “primary” name server responsible for all lower nodes
- Resource records (RR) are defined for each node
Delegation Records

- Distributes database administration
- Name Server (NS) RR
- Zone Start of Authority (SOA)

Delegation: NS and “Glue”

- NS resource record (RR)
- “Glue” entries in parent and child

```
<domain>   NS   <server>
SRI.COM.   NS   KL.SRI.COM.
KL.SRI.COM. A   10.1.0.2
```
Delegation: SOA

\[
\begin{align*}
{name} & \ [\ttl] & [\texttt{<class>}] & \texttt{SOA} & \texttt{<origin>} & \texttt{<person>} \\
( & \texttt{<serial>} \\
& \texttt{<refresh>} \\
& \texttt{<retry>} \\
& \texttt{<expire>} \\
& \texttt{<minimum>} & )
\end{align*}
\]

\$\text{ORIGIN ARPA.}

@ IN SOA SRI-NIC.ARPA HOSTMASTER.SRI-NIC.ARPA. ( \\
45 ;serial (sequential) \\
3600 ;refresh (1 hour regular check) \\
600 ;retry (10 minutes until check) \\
3600000 ;expire (42 days until refresh) \\
86400 ) ;minimum (a day)

Reverse

- Another hierarchy for in-addr.ARPA

Reverse the order in the label because names aggregate within suffixes rather than (address) prefixes

\[
\begin{align*}
27.0.2.10.IN-ADDR.ARPA. & \text{ PTR } VAXA.ISI.EDU. \\
33.0.9.128.IN-ADDR.ARPA. & \text{ PTR } VAXA.ISI.EDU.
\end{align*}
\]
"For every IP address, there should be a matching PTR record in the in-addr.arpa domain."

RFC 1912. Common DNS Operational and Configuration Errors

RFC 1033. Domain Administrators Operations Guide

"PTR's should use official names and not aliases."

Reverse not CNAME

Reverse Requirement
Reverse Complication

- Address field separations not on dotted decimal boundaries
- Create CNAMEs in IN-ADDR.ARPA
  - Labels are just text—describing addresses
  - Parent creates labels: 1.2/25...in-addr.arpa
  - Delegates zone for 2/25...in-addr.arpa

DNS Servers and Resolvers

- When an application connects to a host by name, the OS asks the resolver what address to use
  - Most applications use addresses in the order provided by the resolver
- DNS client is a name resolver
- DNS server answers queries
**DNS Servers and Resolvers**

- Application connects by name, the OS gets the address from the resolver.
- Most applications use addresses in the order provided by the resolver.

**TCP and UDP Ports**

- Port 53 for both TCP and UDP.
- UDP for queries if small enough.
- TCP for zone transfer.
- Server uses source of 53 when "forwarding".
Redirection and Recursion

- Redirection: “Take your question over there.”
- Recursion: “I’ll check and get back to you.”

DNS First Query

- Clients (stub resolvers) query local DNS server for IP addresses
- Local server starts with the root name server and follows referrals until it finds a server that has the answer
- Local servers send answers back to the clients and cache the answers
DNS Subsequent Queries

- Clients (stub resolvers) query local DNS server for IP addresses
- After the first time, the answer is found in the cache
- Local servers send answers back to the clients and cache the answers

Caching and Forwarders

- Caching is controlled by the time to live (TTL) in RR
- Negative caching required by RFC 2308
- The “minimum” ttl parameter in the SOA is the ttl for caching negative answers
- Sending a recursive query to a forwarder builds a cache for the site
Time to Live

- Changing host addresses
- Reduce TTL prior to change
- Then restore to manage the load
- CNR dynamically updates DNS ttl with 1/3 DHCP lease time

Secondary Servers

- Reliability depends on separation
- Location—physical and subnet
- Independent fate—separate power
- Separate administration if possible
- RFC 2182—best current practice
Replication

- Transfer zone contents
- Transfer controlled by serial number and refresh parameters in SOA
- Incremental transfer

Replication Efficiency

- Notify (new protocol operation) enables primary to inform secondary when RRs have changed [RFC 1996]
- Incremental transfer (IXFR) sends just changes to the zone [RFC 1995]
Dynamic Update

- Atomic update of RR-set
- Only if conditions are met
  - RR exists
  - RR has specified value
- Trusting version—RFC 2136
- Secure version—RFC 3007
- Created so that DHCP can update DNS

Securing Queries

- TSIG—Transaction Signature
  - RFC 2845
- Secret-key hash of the transaction
  (HMAC-MD5) to the forwarder
- Pseudo RR, not cached or saved
- Only useful with local forwarders
Securing Zone Transfer

- TSIG RR
- Secondary servers have an administrative relationship that can support secret keys
- Don’t need the overhead of public keys

TKEY

- Transaction KEY—not stored
- Use DNS to establish secret keys alternative to manual keys
- Modes include
  - Diffie-Hellman
  - GSS-API
  - Server or Resolver assigned encrypted (encrypted using KEY RR)
- RFC 2930
Securing Zone Contents

- DNS security
- Key RR—distributes public keys
- SIG RR—authenticates (signs) RR set
- NXT RR—“next” record enables authentication of non-existence
- RFC 2535

Deployment of DNS Sec

- Experimental only now
- Trust depends on the entire path to the root
- Signing all the RRsets in .com is an unresolved problem
Split DNS

• External holds limited contents for public

• Internal
  - Isolated clients query DNS servers configured as root
  - Internal (secondary) servers forward to external caching server for other domains

Support for Renumbering

• DNAME—RFC 2672
  - Map an entire subtree of the DNS name space to another domain

• Suffix substitution
  - If looking up a QNAME a.b.c.d.e.f encounters a DNAME record d.e.f. DNAME w.xy.
  - It will look for a.b.c.w.xy.
IP Version 6

• AAAA—RFC 1886—IP6.INT
  Nibbles—4-bit fields in IP6.INT
• A6—RFC 2874—IP6.ARPA
  Binary label—RFC 2673
• Which to use will be decided soon

Binary Label

• RFC 2673
• Example—these are equivalent!

\[x3FFE07C0004000090A0020FFFE812B32/128].IP6.ARPA.
\[x0A0020FFFE812B32/64].\[x0009/16].\[x3FFE07C00040/48].IP6.ARPA.
Records for Applications

- MX
- SRV
- NAPTR

MX

- Mail eXchange RR
- Where the mail for the host is to be sent
- Round-robin within equal preferences
- Mailers send only to lower preference numbers

Name   TTL   Class MX   Preference  Target

BAZ.FOO.COM.  MX  10  PO1.FOO.COM.
MX  20  PO2.FOO.COM.
MX  30  PO3.FOO.COM.
Wildcards

- Special treatment for ‘*’ in the label
- Any name in the query matches, and the answer is synthesized
- Most often used in mail exchange

```
FOO.COM.       MX   10   RELAY.CS.NET.
* .FOO.COM.     MX   20   RELAY.CS.NET.
```

SRV

- Generalize the MX idea
- Find hosts offering service in a domain
- Add structure to the name
- Add fields to the RR—specialize Priority and Weight (replace Preference)
- Target must NOT be an alias (CNAME)
SRV

Format of RR and Example

_Service._Proto.Name   SRV Priority Weight Port Target
_ldap._tcp.example.com  SRV 1 10 389 ldap1.example.com
                 SRV 1 20 389 ldap2.example.com

NAPTR

- Naming authority pointer
- Universal resource identifier
- Regular expressions
- Replacement strings
- RFC 2915
NAPTR

Format:
DomainName TTL Class NAPTR
Order Preference Flags Service Regexp Replacement

Examples:
cid.urn.arpa. IN NAPTR
"; order pref flags service regexp replacement
100 10 "" "" /urn:cid:.+@([^\./]+\.)\{\}*$/\2/i" .

http.uri.arpa. IN NAPTR
"; order pref flags service regexp replacement
100 90 "" "" !http://([^:/]+)!\1!i" .

Load Sharing

• Resolvers use addresses in the order received, although the original concept was that they choose randomly

• DNS server can rotate the order of the (multiple) addresses of a hostname to distribute the load
Source-Dependent Answers

- Return addresses in the order of “closeness” to the resolver
- Same subnet is close, but requires knowing the subnet mask
- Can look into the routing structure
- DNS support for content networking uses other metrics for which answer to give

Summary

- Addresses can be allocated automatically
- DNS can support more than just name to address lookup
Deploying DNS and DHCP Services
Session NCM-240

Please Complete Your Evaluation Form
Session NCM-240
Reverse Delegation

- RFC 2317
  2.3/12 CNAME xxx.IN-ADDR.ARPA

For delegation of
192.0.2.0/25 to organization A
192.0.2.128/26 to organization B
192.0.2.192/26 to organization C

Reverse Delegation Problem

$ORIGIN 2.0.192.in-addr.arpa.
;
1       PTR     host1.A.domain.
2       PTR     host2.A.domain.
3       PTR     host3.A.domain.
;
129     PTR     host1.B.domain.
130     PTR     host2.B.domain.
131     PTR     host3.B.domain.
;
193     PTR     host1.C.domain.
194     PTR     host2.C.domain.
195     PTR     host3.C.domain.
Reverse Delegation Solution

$ORIGIN 2.0.192.in-addr.arpa.

@ IN SOA my-ns.my.domain.
hostmaster.my.domain. (...)

;/...
;  <<0-127>> /25
0/25 NS ns.A.domain.
0/25 NS some.other.name.server.
;
1 CNAME 1.0/25.2.0.192.in-addr.arpa.
2 CNAME 2.0/25.2.0.192.in-addr.arpa.
3 CNAME 3.0/25.2.0.192.in-addr.arpa.

;  <<128-191>> /26
128/26 NS ns.B.domain.
128/26 NS some.other.name.server.too.
;
129 CNAME 129.128/26.2.0.192.in-addr.arpa.
130 CNAME 130.128/26.2.0.192.in-addr.arpa.
131 CNAME 131.128/26.2.0.192.in-addr.arpa.

$ORIGIN 0/25.2.0.192.in-addr.arpa.

@ IN SOA ns.A.domain. hostmaster.A.domain. (...)
@ NS ns.A.domain.
@ NS some.other.name.server.
;
1 PTR host1.A.domain.
2 PTR host2.A.domain.
3 PTR host3.A.domain.