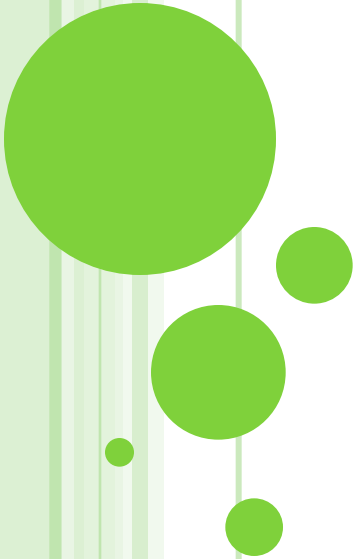
DHCP, Firewall and NAT



DHCP –
Dynamic Host Configuration Protocol

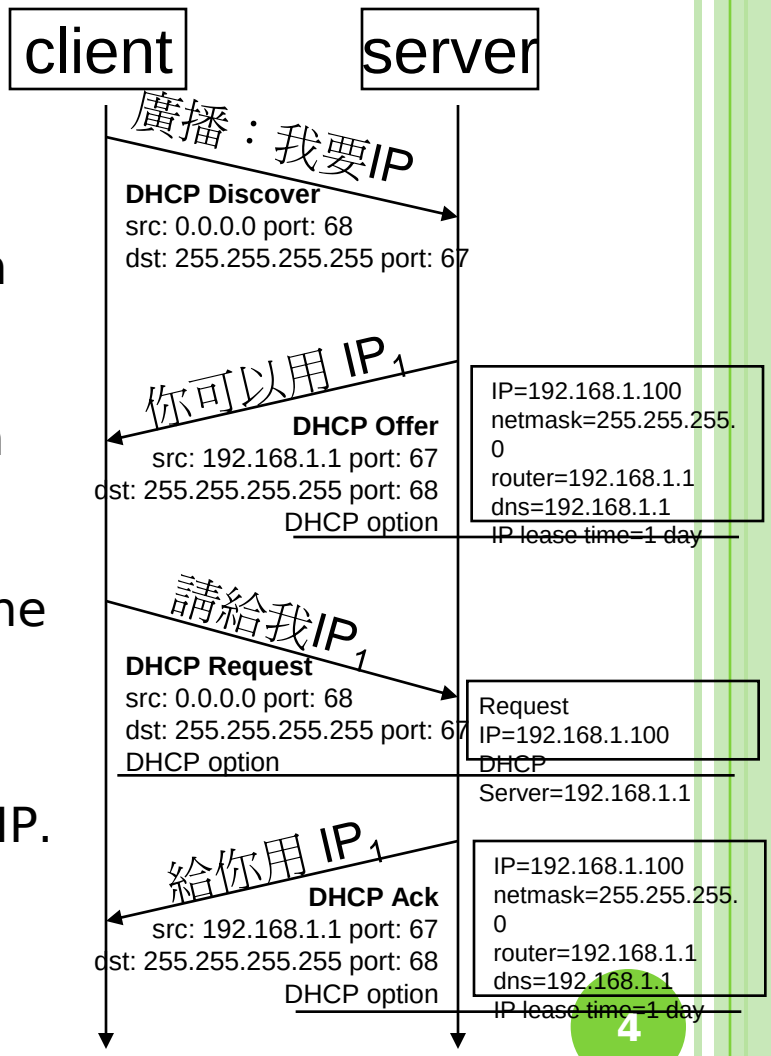
DHCP introduction

- DHCP
  - Dynamic Host Configuration Protocol
  - A system can connect to a network and obtain the necessary information dynamically
- Client-Server architecture
  - DHCP client broadcasts request for configuration info.
    - UDP port 68
  - DHCP server reply on UDP port 67, including
    - IP, netmask, DNS, router, IP lease time, etc.
- RFC
  - RFC 2131 – Dynamic Host Configuration Protocol
  - RFC 2132 – DHCP Options

DHCP Protocol (1)

- DHCP Discover
  - Broadcasted by client to find available server.
  - Client can request its last-known IP, but the server can ignore it.
- DHCP Offer
  - Server find IP for client based on clients hardware address (MAC)
- DHCP Request
  - Client request the IP it want to the server.
- DHCP Acknowledge
  - Server acknowledges the client, admit him to use the requested IP.

※ Question
  - Why not use the IP after DHCP offer?



DHCP Protocol (2)

- DHCP inform
  - Request more information than the server sent.
  - Repeat data for a particular application.
  - ex. browser request proxy info. from server.
  - It does not refresh the IP expiry time in server's database.
- DHCP Release
  - Client send this request to server to releases the IP, and the client will un-configure this IP.
  - Not mandatory.

DHCP server on FreeBSD (1)

- Kernel support (in GENERIC)

```
# The `bpf' device enables the Berkeley Packet Filter.
# Be aware of the administrative consequences of enabling this!
# Note that 'bpf' is required for DHCP.
device          bpf             # Berkeley packet filter
```

- Install DHCP server
  - cd/usr/ports/net/isc-dhcp3-server/
  - cd /usr/local/etc
  - cp dhcpd.conf.sample dhcpd.conf
- Enable DHCP server in /etc/rc.conf

```
dhcpd_enable="YES"
#dhcpd_flags="-q"
#dhcpd_conf="/usr/local/etc/dhcpd.conf"
#dhcpd_ifaces=""
#dhcpd_withumask="022"
```

DHCP server on FreeBSD (2)

- Option definitions

```
option domain-name "cs.nctu.edu.tw";
option domain-name-servers 140.113.235.107,
    140.113.1.1;

default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
log-facility local7;
```

/etc/syslogd.conf
/etc/newsyslog.conf

DHCP server on FreeBSD (3)

- Subnet definition

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.101 192.168.1.200;
    option domain-name "cs.nctu.edu.tw";
    option routers 192.168.1.254;
    option broadcast-address 192.168.1.255;
    option domain-name-servers 140.113.235.107, 140.113.1.1;
    default-lease-time 3600;
    max-lease-time 21600;
}
```

- Host definition

```
host fantasia {
    hardware ethernet 08:00:07:26:c0:a5;
    fixed-address 192.168.1.30;
}
host denyClient {
    hardware ethernet 00:07:95:fd:12:13;
    deny booting;
}
```

DHCP server on FreeBSD (4)

- Important files
  - /usr/local/sbin/dhcpd
  - /usr/local/etc/dhcpd.conf
  - /var/db/dhcpd/dhcpd.leases (leases issued)
  - /usr/local/etc/rc.d/isc-dhcpd

http://www.freebsd.org/doc/en/books/handbook/network-dhcp.html

PXE (Preboot Execution Environment)

- /usr/local/etc/dhcpd.conf

```
subnet 192.168.7.0 netmask 255.255.255.0 {
        option subnet-mask 255.255.255.0;
        range dynamic-bootp 192.168.7.100 192.168.7.109;
        option root-path "/home/tftproot";
        next-server 192.168.7.254;
        server-identifier 192.168.7.254;
        filename "/boot/pxeboot";
        option routers 192.168.7.254;
}
```

- /etc/inetd.conf

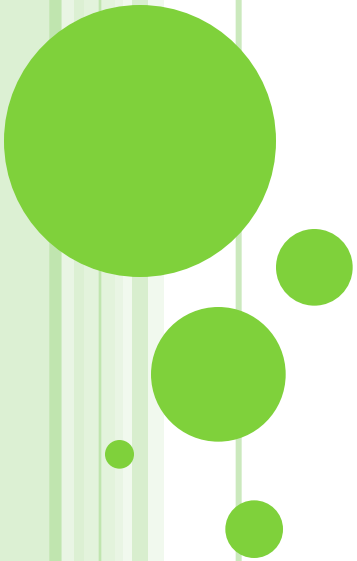
```
tftp  dgram  udp  wait  root  /usr/libexec/tftpd  tftpd -l -s /home/tftproot
```

- /etc/exports

```
/home/tftproot -ro -maproot=nobody -network 192.168.7.0 -mask 255.255.255.0
```

- /home/tftproot
  - What in the CD
  - gzip -d boot/mfsroot.gz

http://www.freebsd.org/doc/en/articles/pxe/article.html



Firewalls

Firewalls

- Firewall
  - A piece of hardware and/or software which functions in a networked environment to prevent some communications forbidden by the security policy.
  - Choke point between secured and unsecured network
  - Filter incoming and outgoing traffic that flows through your system
- How can it be used to do
  - To protect your system from unwanted traffic coming in from the public Internet
    - Such as telnet, NetBIOS
  - To limit or disable access from hosts of the internal network to services of the public Internet
    - Such as MSN, ssh, ftp
  - To support NAT (Network Address Translation)

Firewalls – Layers of Firewalls

- Network Layer Firewalls
  - Operate at a low level of TCP/IP stack as IP-packet filters.
  - Filter attributes
    - Source/destination IP
    - Source/destination port
    - TTL
    - Protocols
    - …
- Application Layer Firewalls
  - Work on the application level of the TCP/IP stack.
  - Inspect all packets for improper content, a complex work!
- Application Firewalls
  - The access control implemented by applications.

Firewall Rules

- Two ways to create firewall rulesets
  - Exclusive
    - Allow all traffic through except for the traffic matching the rulesets
  - Inclusive
    - Allow traffic matching the rulesets and blocks everything else
    - Safer than exclusive one
      - reduce the risk of allowing unwanted traffic to pass
      - Increase the risk to block yourself with wrong configuration

Firewall Software

- FreeBSD
  - IPFIREWALL (known as IPFW)
  - IPFILTER (known as IPF)
  - *Packet Filter (known as PF)*
- Solaris
  - IPF
- Linux
  - ipchains
  - iptables

Packet Filter (PF)

- Introduction
  - Firewall migrated from OpenBSD
  - NAT, Bandwidth limit (ALTQ) support
  - Load balance
  - http://www.openbsd.org/faq/pf/

PF in FreeBSD (1)

- Enable PF in /etc/rc.conf

  pf_enable="YES"

  pf_rules="/etc/pf.conf"

- Rebuild Kernel (if ALTQ is needed)

  device pf

  device pflog

  device pfsync

  options ALTQ

  options ALTQ_CBQ

  options ALTQ_RED

ALTQ -- alternate queuing of network packets

PF in FreeBSD (2)

- PF command
  - pfctl –s <rules|nat|queue|tables> -v
  - pfctl /etc/pf.conf
  - pfctl –t <table> -T <add|delete> <ip>
  - pfctl –t <table> -T show

PF in FreeBSD (3)

- PF Configuration File
- The last matching rule "wins“
  - "quick" keyword
- /etc/pf.conf
  - Macros
    - define common values, so they can be referenced and changed easily.
  - Tables
    - similar to macros, but more flexible for many addresses.
  - Options “set”
    - tune the behavior of pf, default values are given.
  - Normalization “scrub”
    - reassemble fragments and resolve or reduce traffic ambiguities.
  - Queueing “altq”, “queue”
    - rule-based bandwidth control.
  - Translation (NAT) “rdr”, “nat”, “binat”
    - specify how addresses are to be mapped or redirected.
  - Filtering “antispoof”, “block”, “pass”
    - the implicit first two rules are

PF in FreeBSD (4)

- Ex.

```
# macro definitions
extdev='fxp0‘
intranet='192.168.219.0/24‘
winxp=‘192.168.219.1’
server_int=‘192.168.219.2’
server_ext=‘140.113.214.13’

# options
set limit { states 10000, frags 5000 }
set loginterface $extdev
set block-policy drop

# tables
table <badhosts> persist file “/etc/badhosts.list”

# filtering rules
pass in  all
pass out all
block log in on $extdev proto tcp from any to any port {139, 445}
block log in on $extdev proto udp from any to any port {137, 138}
block on $extdev quick from <badhosts> to any
pass in on $extdev proto tcp from 140.113.0.0/16 to any port {139, 445}
pass in on $extdev proto udp from 140.113.0.0/16 to any port {137, 138}
```

PF in FreeBSD (5)

- Logging
  - pflogd
    - /etc/rc.conf

    pflogd_enable="YES"

    pflogd_flags="-f <filename>“
  - pflog(4)
    - /dev/pflog
    - A pseudo-device which makes visible all packets logged by the packet filter, pf(4).

NAT – Network Address Translation

NAT (1)

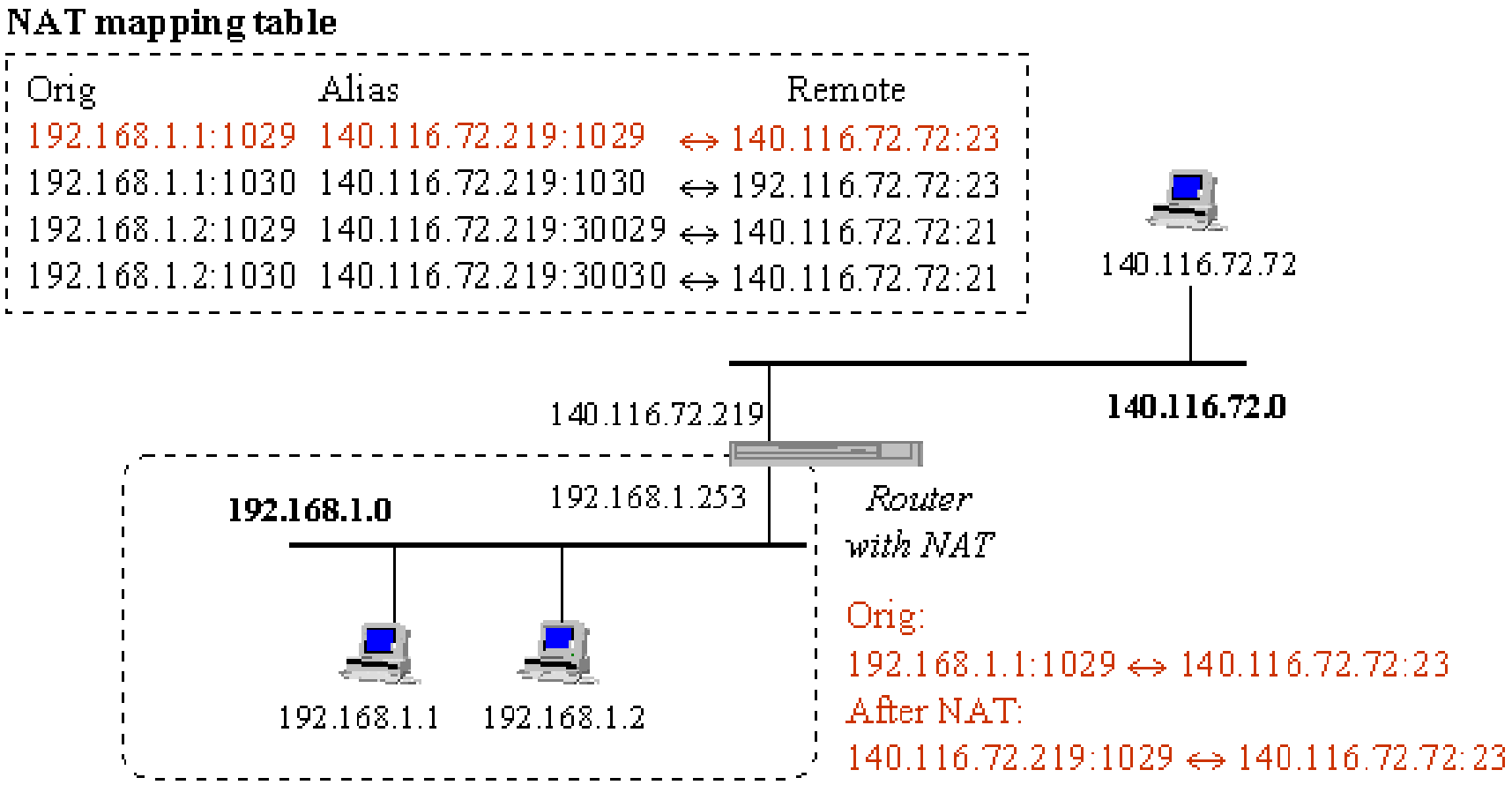
- What is NAT?
  - Network Address Translation
  - Re-write the source and/or destination addresses of IP packets when they pass through a router or firewall.
  - What can be re-written?
    - Source/destination IPs
    - Source/destination ports
- What can NAT do?
  - Solve the IPv4 address shortage. (the most common purpose)
  - Kind of firewall (security)
  - Load balancing
  - Fail over (for service requiring high availability)
  - Transparent proxy

NAT (2)

- Address shortage of IPv4
- Private addresses space defined by RFC1918
  - 24-bit block (Class A)
    - 10.0.0.0/8
  - 20-bit block (16 contiguous Class B)
    - 172.16.0.0/12 ~ 172.31.0.0/12
  - 16-bit block (256 contiguous Class C)
    - 192.168.0.0/16 ~ 192.168.255.0/16
- Operation consideration
  - Router should set up filters for both inbound and outbound private network traffic

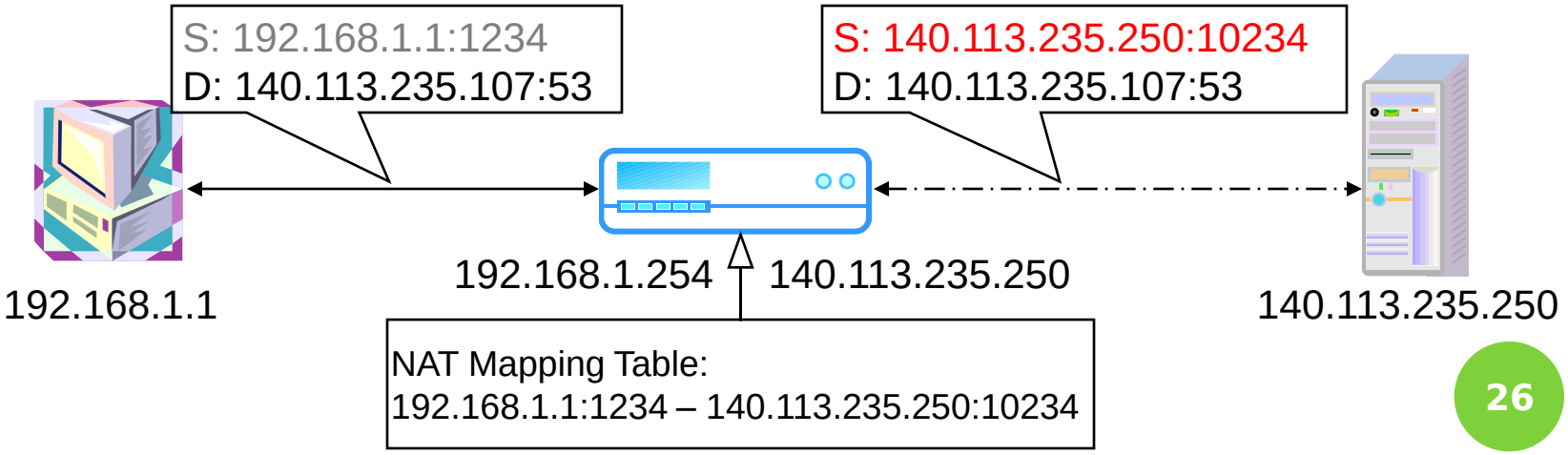
NAT (3)

- NAT example:



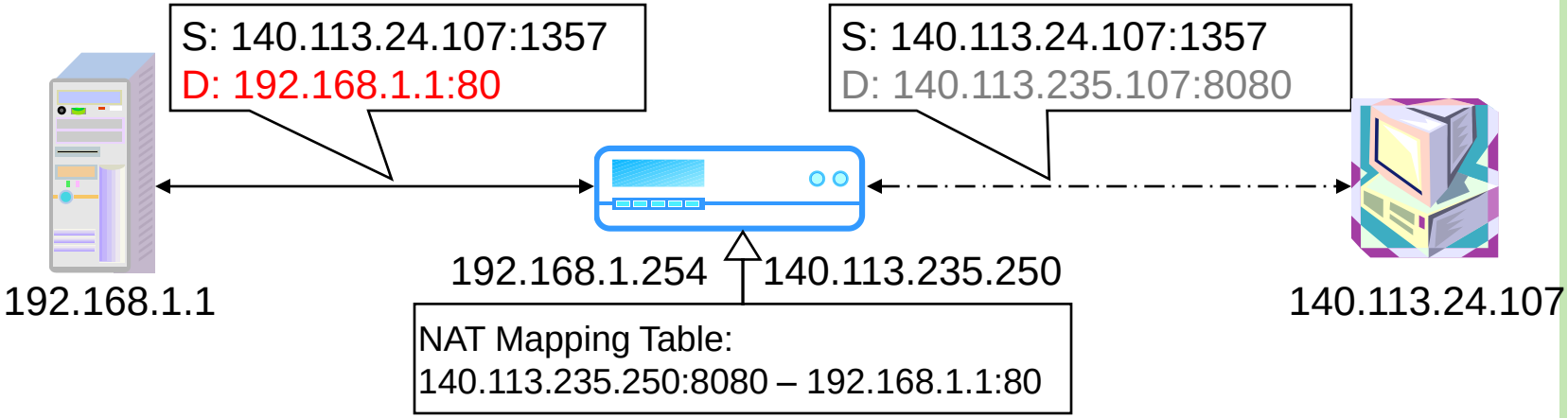
NAT (4)

- SNAT & DNAT
  - S: Source D: Destination
  - SNAT
    - Rewrite the source IP and/or Port.
    - The rewritten packet looks like one sent by the NAT server.



NAT (5)

- DNAT
  - Rewrite the destination IP and/or Port.
  - The rewritten packet will be redirect to another IP address when it pass through NAT server.



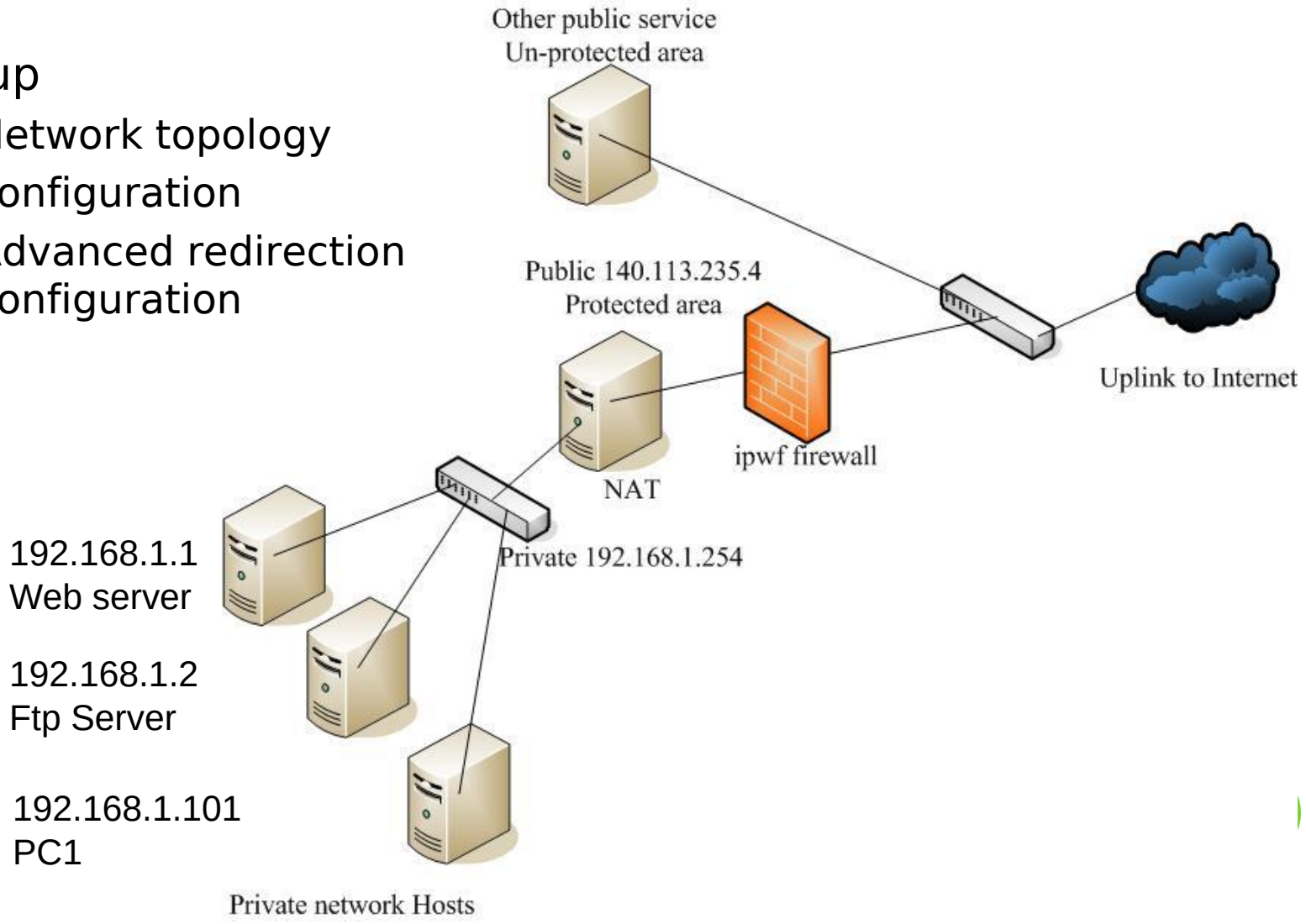
- Both SNAT and DNAT are usually used together in coordination for two-way communication.

NAT (6)

- Types of NAT
  - Full cone NAT
    - map an internal IP and port to a public port
  - A restricted cone NAT
    - Full Cone with IP filtering
  - A port restricted cone NAT
    - Full Cone with IP and port filtering
  - A symmetric NAT
    - Build IP and port mapping according to a session ID
- Problem of NAT

NAT on FreeBSD (1)

- Setup
  - Network topology
  - configuration
  - Advanced redirection configuration



NAT on FreeBSD (2)

- IP configuration (in /etc/rc.conf)

  ifconfig_fxp0="inet 140.113.235.4 netmask 255.255.255.0 media autoselect"
  ifconfig_fxp1="inet 192.168.1.254 netmask 255.255.255.0 media autoselect“
  defaultrouter="140.113.235.254“

- Enable NAT
  - Here we use Packet Filter (PF) as our NAT server
  - Configuration file: /etc/pf.conf
    - nat
    - rdr
    - binat

```
# macro definitions
extdev='fxp0‘
intranet='192.168.1.0/24‘
webserver=‘192.168.1.1’
ftpserver=‘192.168.1.2’
pc1=‘192.168.1.101’

# nat rules
nat on $extdev inet from $intranet to any -> $extdev
rdr on $extdev inet proto tcp to port 80 -> $webserver port 80
rdr on $extdev inet proto tcp to port 443 -> $webserver port 443
rdr on $extdev inet proto tcp to port 21 -> $ftpserver port 80
```

NAT on FreeBSD (3)

```
# macro definitions
extdev='fxp0‘
intranet='192.168.219.0/24‘
winxp=‘192.168.219.1’
server_int=‘192.168.219.2’
server_ext=‘140.113.214.13’

# nat rules
nat on $extdev inet from $intranet to any -> $extdev
rdr on $extdev inet proto tcp to port 3389 -> $winxp port 3389
binat on $extdev inet from $server_int to any -> $server_ext
```