

DHCP & NAT

tsaimh (2024, CC BY-SA)
wangth (2018-2023, CC BY-SA)
? (2009-2017)

國立陽明交通大學資工系資訊中心

Information Technology Center of Department of Computer Science, NYCU

DHCP – Dynamic Host Configuration Protocol

DHCP Motivation

- BOOTP (Bootstrap Protocol)
 - Support sending extra information beyond an IP address to a client to enable customized configuration
 - Effectively solve one of the major problems that administrators have with manual configuration
- Problems of BOOTP
 - BOOTP normally uses a **static method** of determining what IP address to assign to a device
- **Dynamic** Host Configuration Protocol (DHCP)
 - DHCP is an extension of the BOOTP. The first word describe the most important new capability added to BOOTP
 - Assign IP dynamically
 - Move away from **static**, **permanent** IP address assignment
 - Compatible with BOOTP

DHCP introduction

- DHCP
 - Dynamic address assignment
 - A pool of IP address is used to dynamically allocate addresses
 - Still support static mapping of addresses
 - Enable a DHCP client to “lease” a variety of network parameters
 - IP, netmask
 - Default router, DNS servers
 - 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
- Two main function of DHCP
 - Provide a mechanism for assigning addresses
 - A method by which clients can request addresses and other configurations

DHCP Address Assignment

- Address allocation mechanisms
 - Provide flexibility for configuring addresses on different types of clients
 - Three different address allocation mechanisms
 - Manual allocation (compatible with bootp)
 - IP address is pre-allocated to a single device
 - Automatic allocation (without lease time)
 - Assign an IP address permanently to a device
 - Dynamic allocation (with lease time)
 - Assign an IP address from a pool for a limited period of time
- Manual allocation
 - Equivalent to the method BOOTP used
 - For servers and routers
 - Administrative benefit

Dynamic allocation

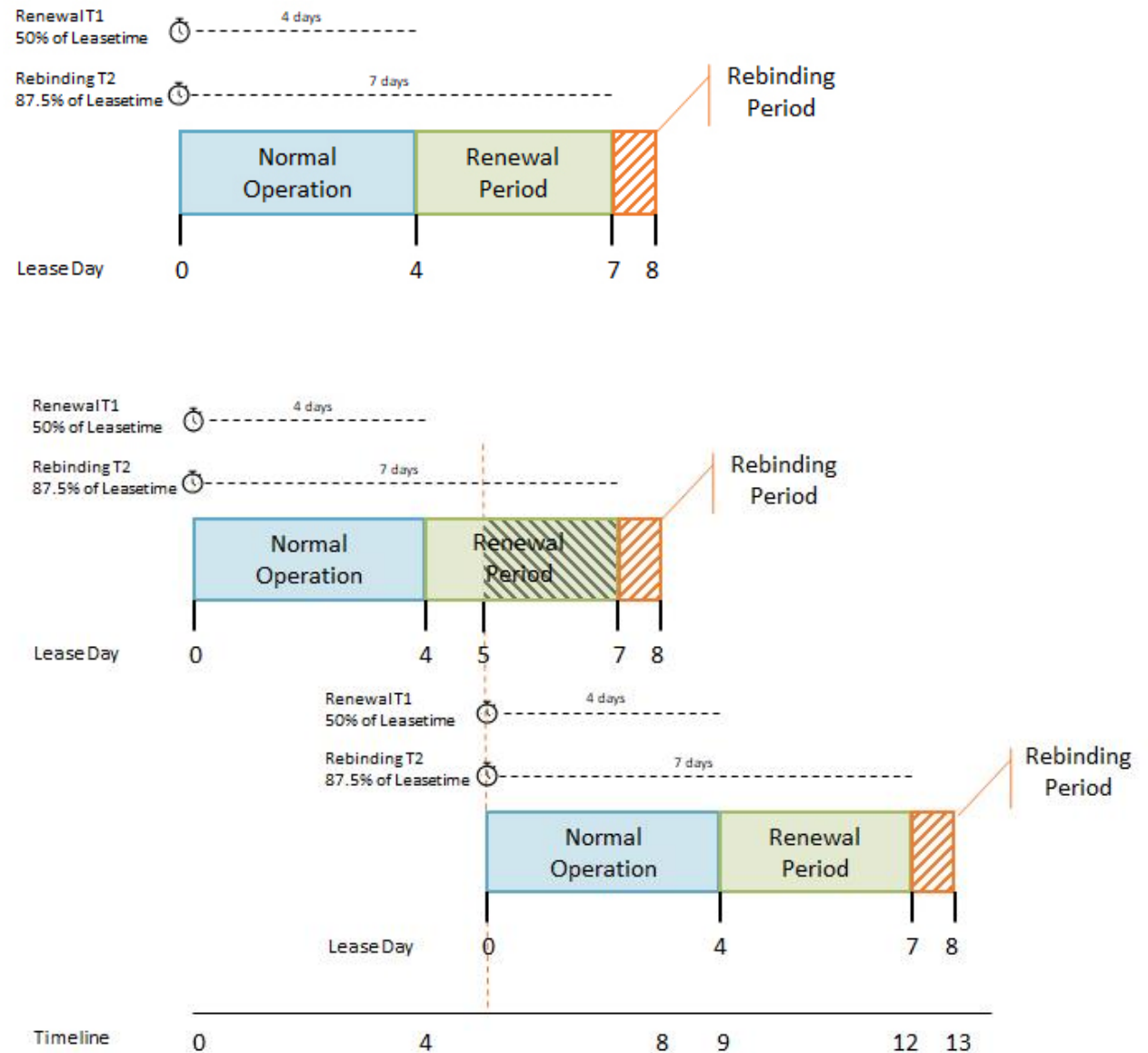
- Benefits for dynamic allocation
 - Automation
 - No intervention for an administrator
 - Centralized management
 - An administrator can easily look to see which devices are using which addresses
 - Address reuse and sharing
 - Portability and universality
 - Do NOT require DHCP server know the identity of each client
 - Support mobile devices
 - Conflict avoidance

DHCP Leases

- Dynamic address allocation is by far the most popular
 - Hosts are said to “lease” an address instead of “own” one
- DHCP lease length policy
 - A trade-off between stability and allocation efficiency
 - The primary benefit of using long lease is that the addresses of hosts are relatively stable
 - Servers
 - The main drawback of using long leases is to increase the amount of time that an IP can be reused
- Assigning lease length by client type
 - Use long lease for desktop computers
 - Use short lease for mobile devices
- Factoring lease renewal into lease length selection

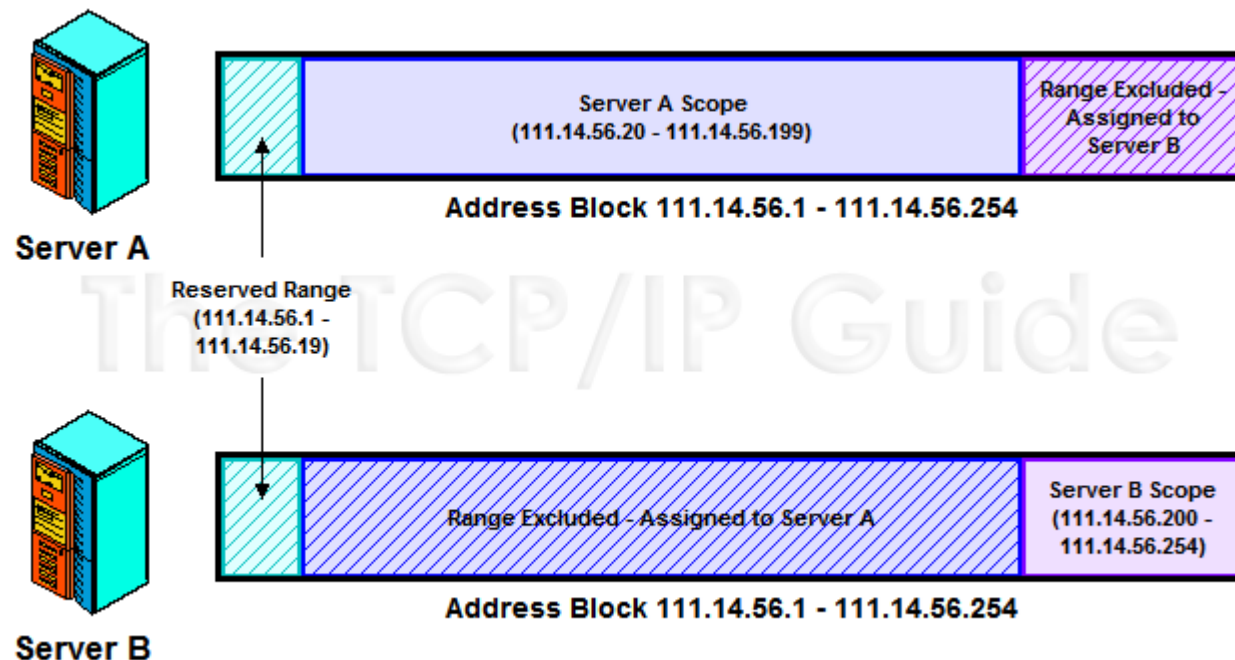
DHCP Lease “Life Cycle”

- Life cycle
 - Allocation
 - Reallocation
 - Normal operation
 - Renewal
 - Rebinding
 - Release



DHCP Lease Address Pools

- Each DHCP server maintains a set of IP addresses
 - Use to allocate leases to clients
 - Most of clients are equals
 - A range of addresses is normally handled as a single group defined for a particular network

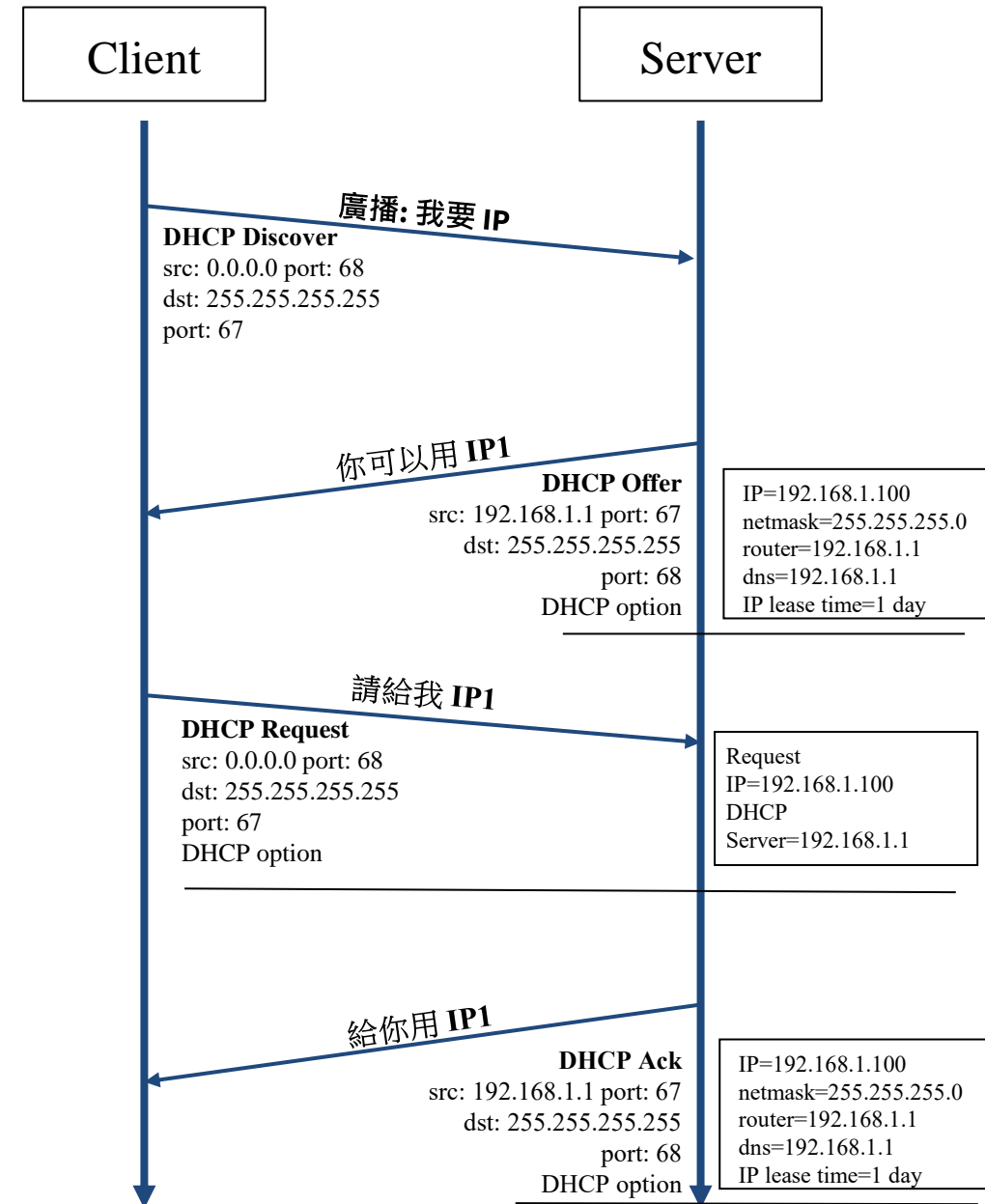


DHCP Protocol (1)

- 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. browsers request web proxy settings from server
 - It does **not** refresh the IP expiry time in server's database
- DHCP Release
 - Client send this request to server to release the IP, and the client will un-configure this IP
 - Not mandatory

DHCP Server on FreeBSD (1)

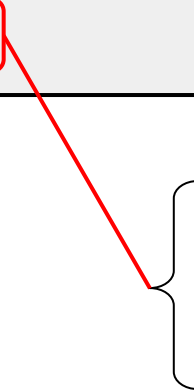
- Kernel support
 - device bpf (FreeBSD 5.x↑)
 - pseudo-device bpf (FreeBSD 4.x↓)
- Install DHCP server
 - `/usr/ports/net/isc-dhcp44-server/`
 - `pkg install isc-dhcp44-server`
- Enable DHCP server in `/etc/rc.conf`
 - `dhcpd_enable="YES"`
 - `dhcpd_flags="-q"`
 - `dhcpd_conf="/usr/local/etc/dhcpd.conf"`
 - `dhcpd_ifaces=""`
 - `dhcpd_withumask="022"`

DHCP Server on FreeBSD (2)

Three-way handshake

- 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 140.113.17.5, 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.leases` (leases issued)
 - `/usr/local/etc/rc.d/isc-dhcpd`

NAT – Network Address Translation

IP Address Crisis

- IP address crisis
 - Run out of class B address
 - The most desirable ones for moderately large organizations
 - IP address were being allocated on a FCFS
 - With no locality of reference
- Solutions
 - Short term
 - Subnetting and CIDR (classless inter-domain routing)
 - NAT (network address translation)
 - Long term
 - IPv6

Network Address Translation (NAT)

- Some important characteristics of how most organizations use the internet
 - Most hosts are client
 - Few hosts access the internet simultaneously
 - Internet communications are routed
- Network Address Translation
 - RFC 1631, in May 1994
 - A basic implementation of NAT involves
 - Using one of the private addresses for local networks
 - Assigned one or more public IP addresses
 - The word ‘translator’ refers to the device that implements NAT

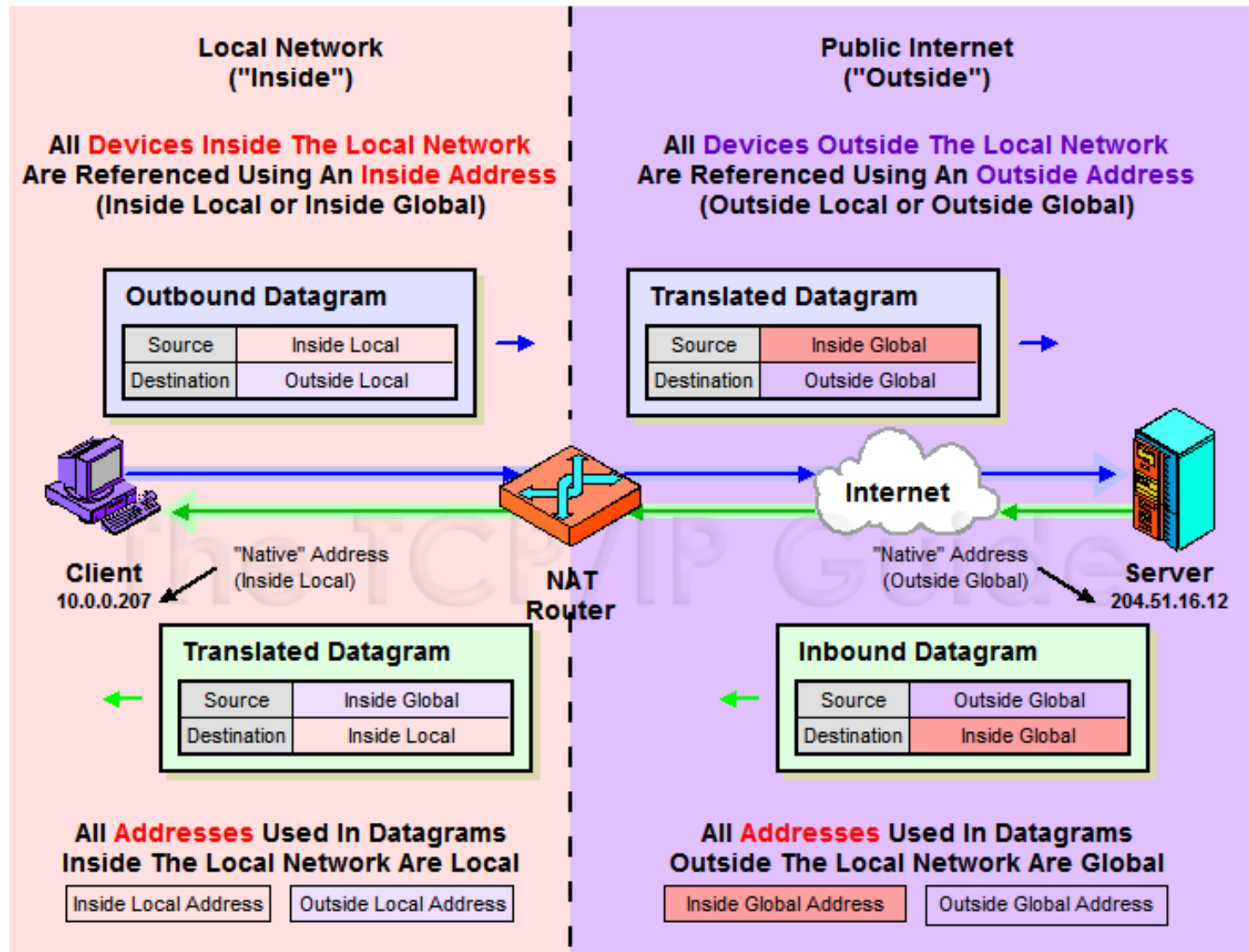
Private Address Space

- 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

Network Address Translation (NAT)

- 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)

NAT Terminology

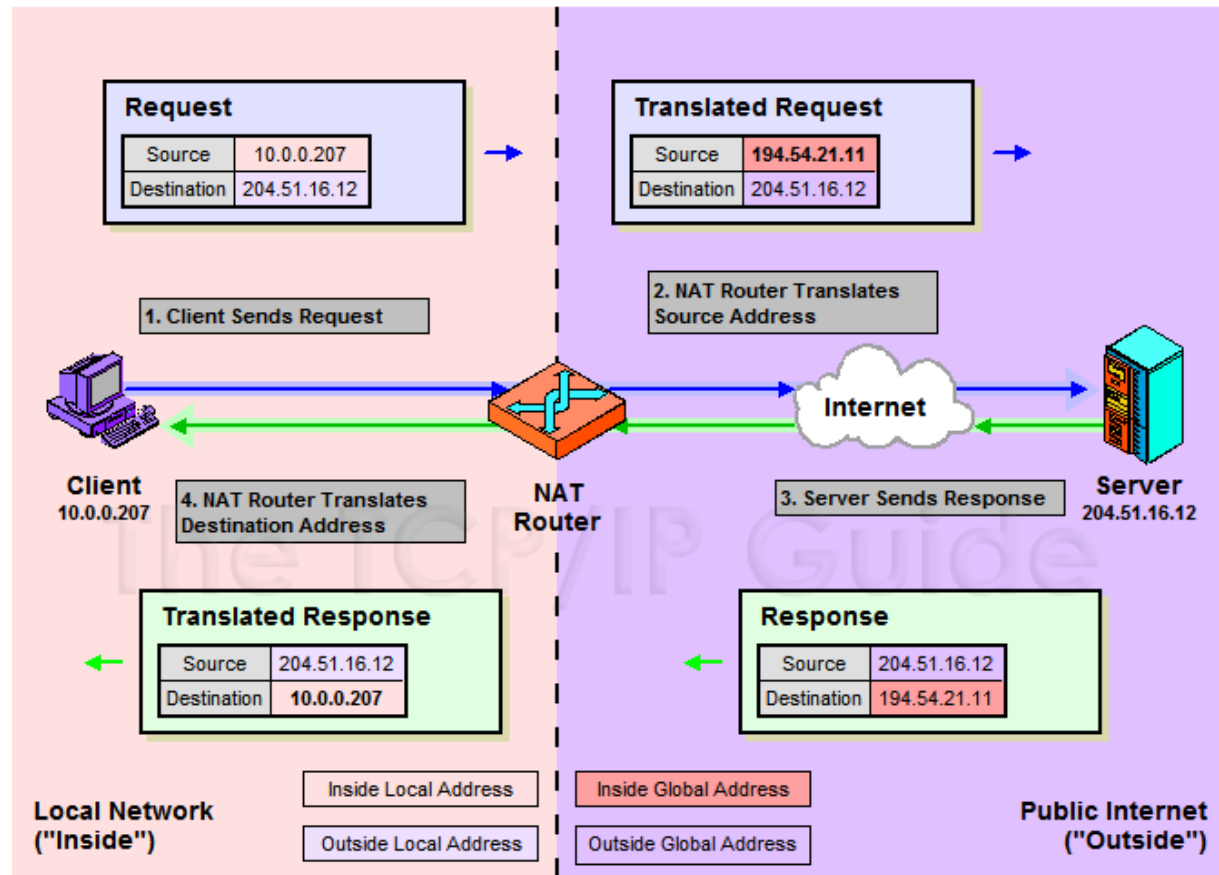


NAT Address Mappings

- Each time a NAT router encounters an IP datagram
 - It must translate addresses
 - BUT, how does it know what to translate, and what to use for the translated addresses
- Translation table
 - Maps the inside local address to the inside global address
 - Also contains mappings between outside global address and outside local address for inbound translations
- Two address mappings
 - Static mappings
 - Allow the inside host with an inside local address to always use a inside global address
 - Dynamic mappings
 - Allow a pool of inside global addresses to be shared by a large number of inside hosts

NAT Unidirectional Operation

- NAT Unidirectional Operation
 - Traditional/Outbound operation
 - The original variety of NAT in RFC 1631
 - The simplest NAT
 - The client/server request/response communication would sent from the inside to outside network



[The TCP/IP Guide - IP NAT Unidirectional \(Traditional/Outbound\) Operation](#)

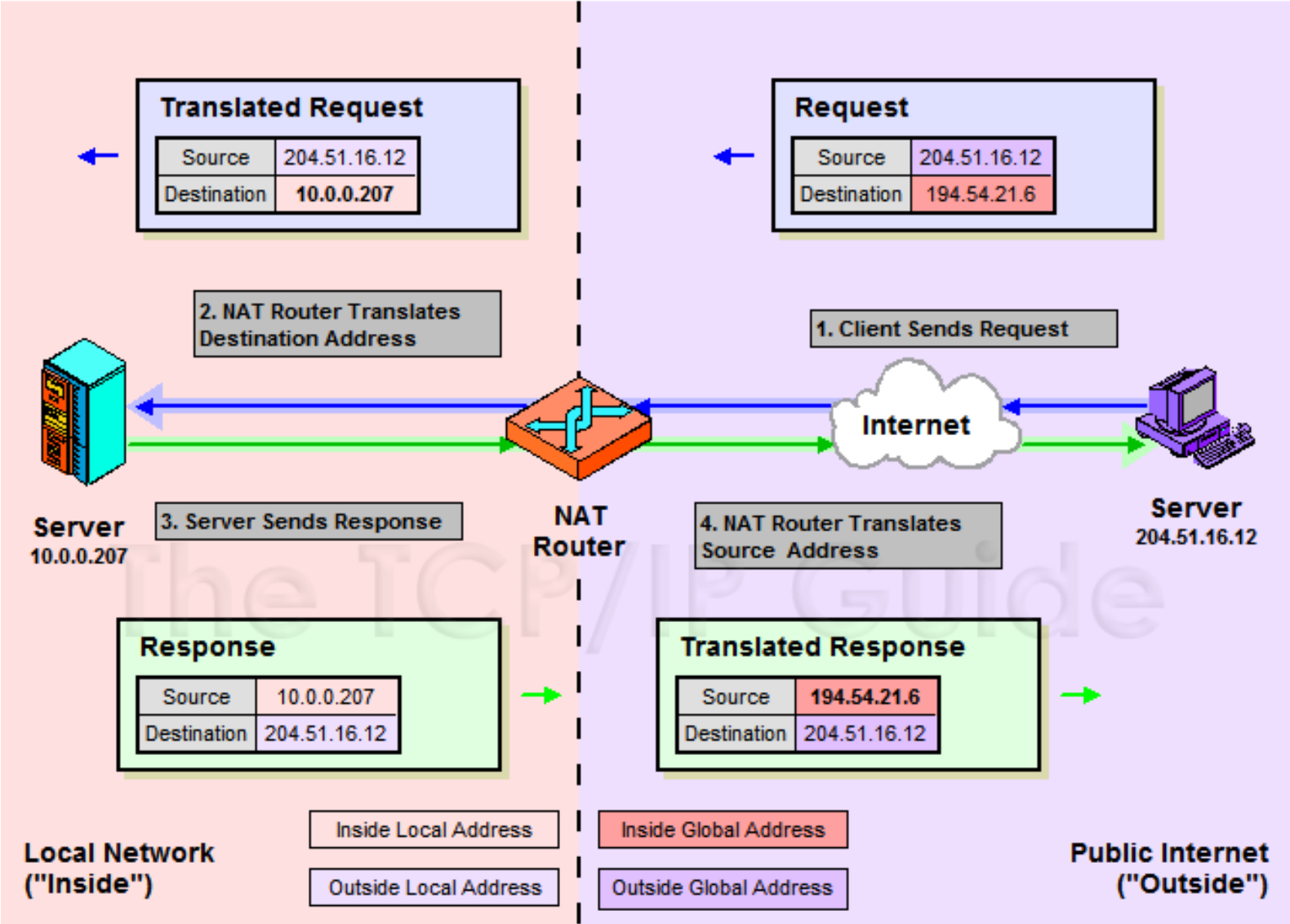
NAT Bidirectional Operation

- NAT Bidirectional Operation
 - Two-Way/Inbound operation
 - A host on the outside network initiate a transaction with one on the inside
- The problem with inbound NAT
 - NAT is inherently asymmetric
 - The outside network does not know the private addresses of the inside network
 - Hidden addresses are not routable
 - The outbound hosts DO NOT know the identity of the NAT router
 - NAT mapping table

NAT Bidirectional Operation

- Two methods to resolve the hidden address problem
 - Static mapping
 - DNS
 - RFC 2694, DNS extensions to NAT
- The basic process is as follows
 - The outside host sends a DNS request using the name of the private host
 - The DNS server for the internal network resolves the name into an inside local address
 - The inside local address is passed to NAT and used to create a dynamic mapping
 - DNS server sends back the name resolution with the inside global address

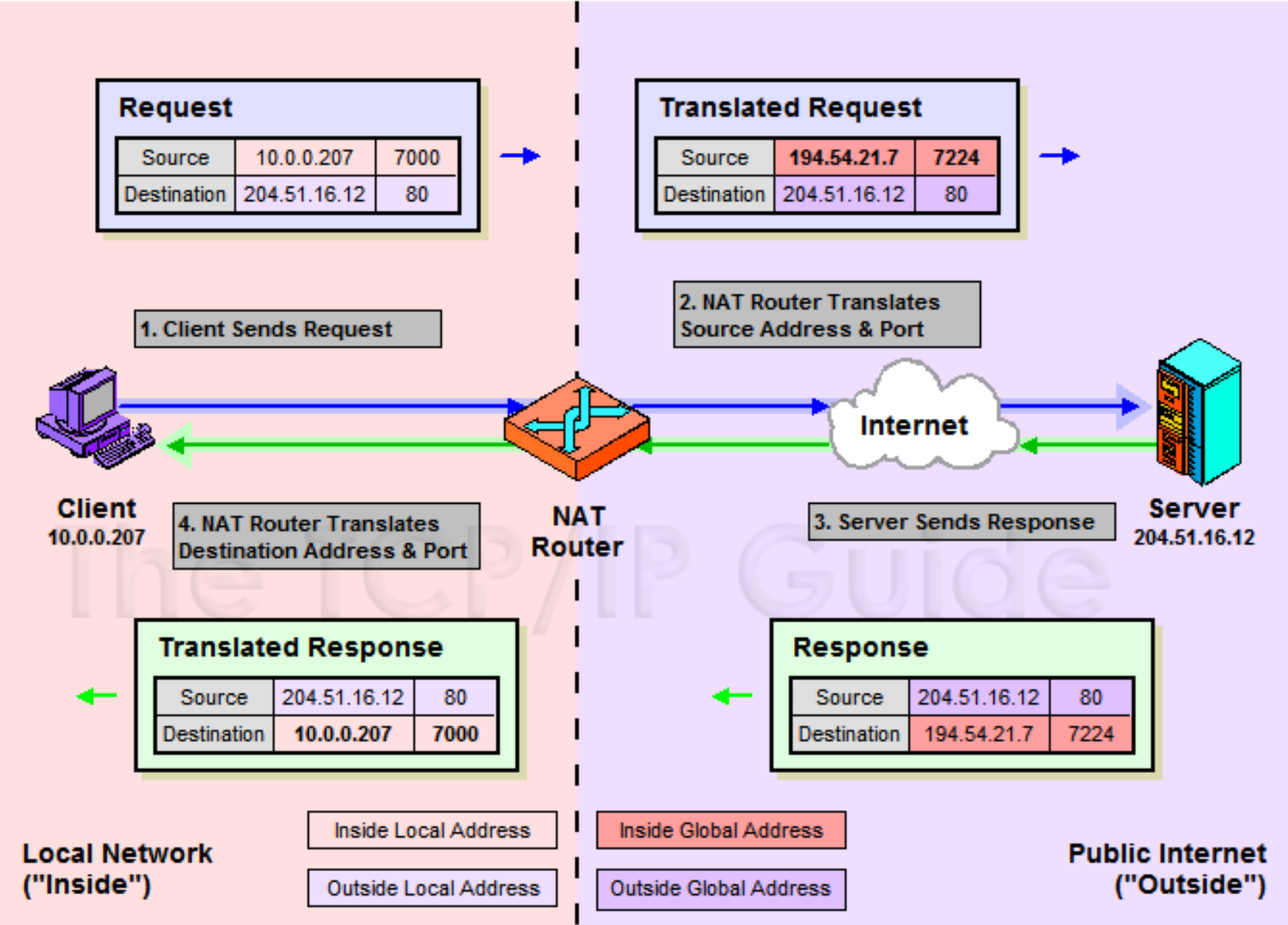
NAT Bidirectional Operation



NAT Port-Based Operation

- NAT Port-Based Operation
 - Overloaded operation
 - Network Address Port Translation (NAPT)/Port Address Translation (PAT)
 - Both traditional NAT and bidirectional NAT work by swapping inside network and outside network addresses
 - One-to-one mapping between inside local address and inside global address
 - Use dynamic address assignment to allow a large number of private hosts to share a small number of registered public addresses
- Using ports to multiplex private addresses
 - Also translate port addresses
 - Allow 250 hosts on the private network to use only 20 IP address
 - Overloading of an inside global address

NAT Port-Based Operation

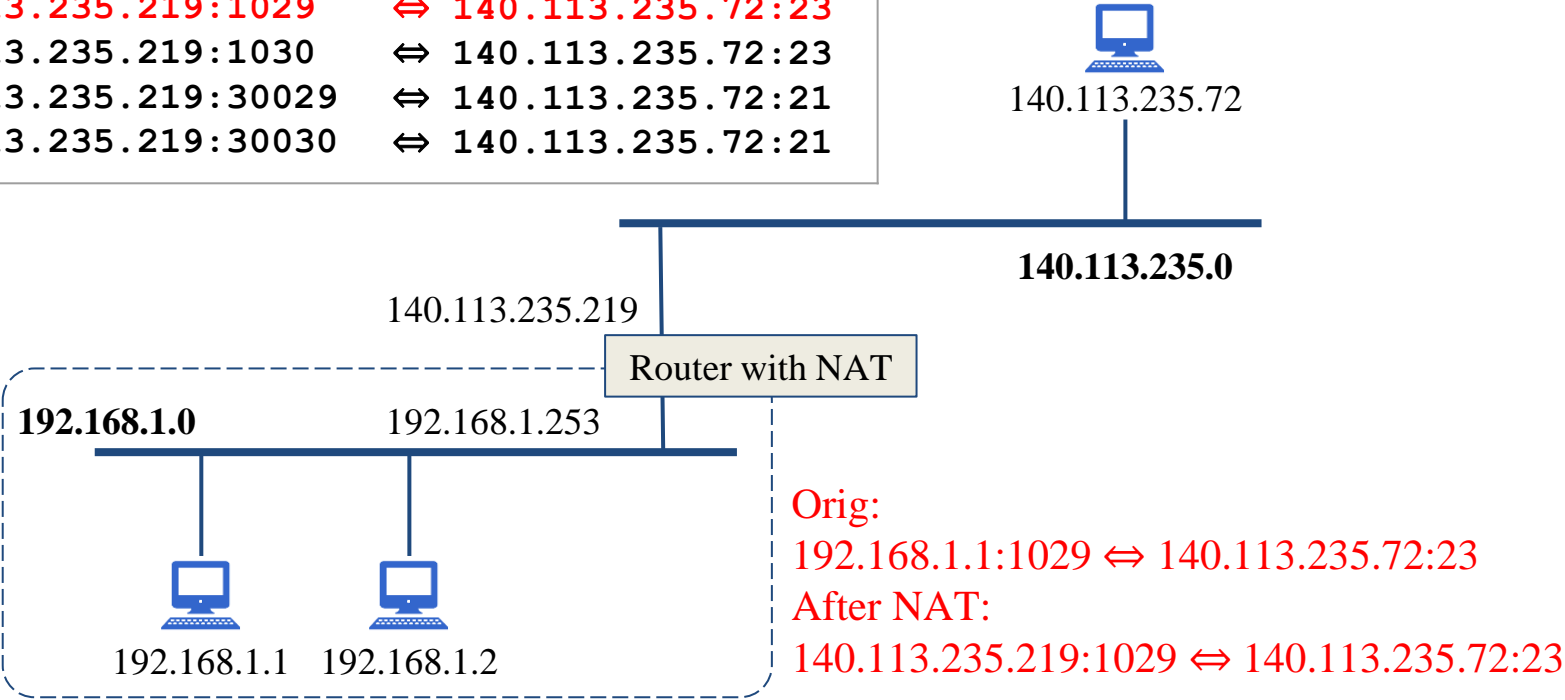


NAT Port-Based Operation

- NAT example:

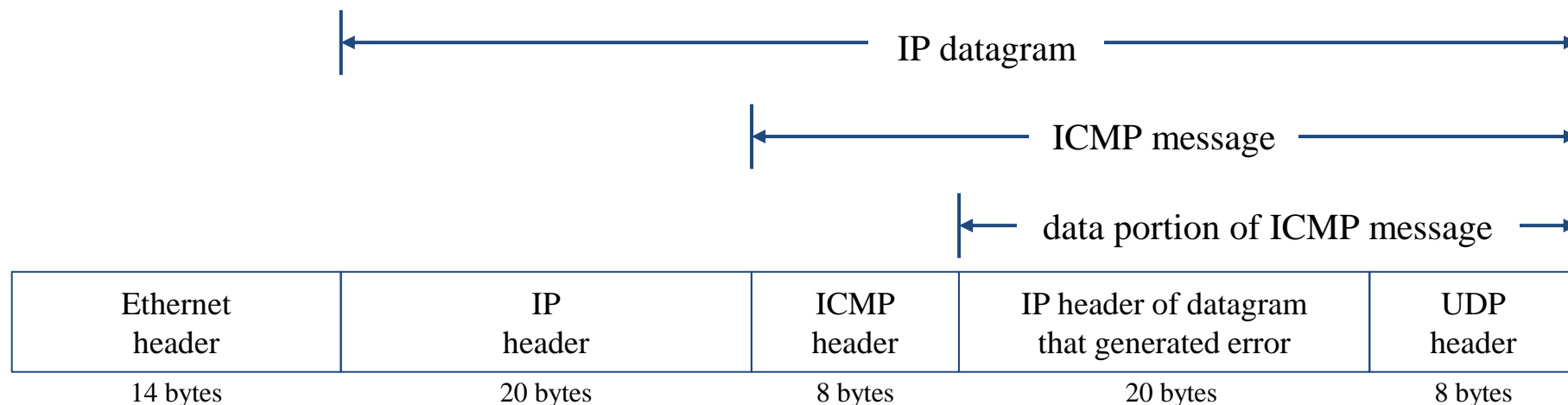
NAT mapping table

Orig	Alias	Remote
192.168.1.1:1029	140.113.235.219:1029	⇔ 140.113.235.72:23
192.168.1.1:1030	140.113.235.219:1030	⇔ 140.113.235.72:23
192.168.1.2:1029	140.113.235.219:30029	⇔ 140.113.235.72:21
192.168.1.2:1030	140.113.235.219:30030	⇔ 140.113.235.72:21



NAT Compatibility Issues

- It is NOT possible for NAT to be completely transparent to the hosts that use it
 - ICMP Manipulations
 - Applications that embed IP address
 - FTP
 - Additional issues with port translation
 - The issues applying to addresses now apply to ports as well
 - Problems with IPsec



Source NAT (SNAT) and Destination NAT (DNAT)

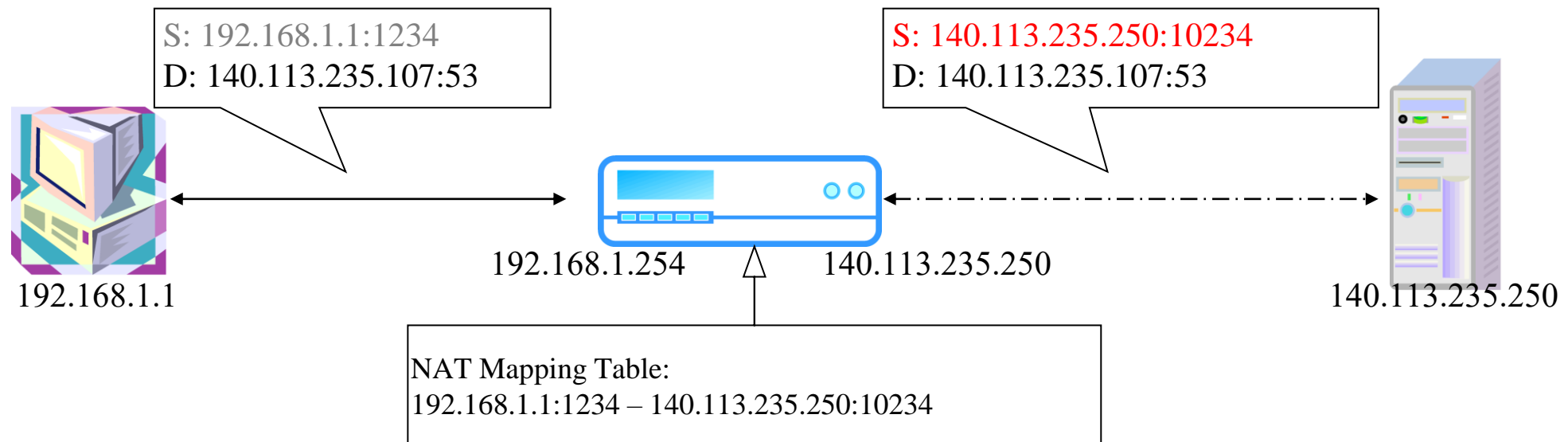
- SNAT & DNAT

- S: Source D: Destination

- SNAT

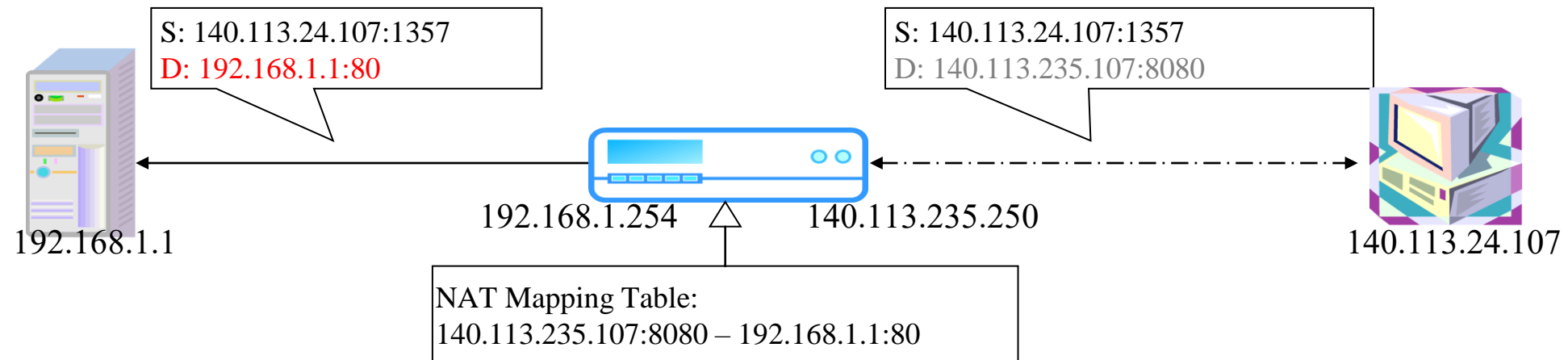
- Rewrite the source IP and/or Port.

- The rewritten packet looks like one sent by the NAT server.



DNAT

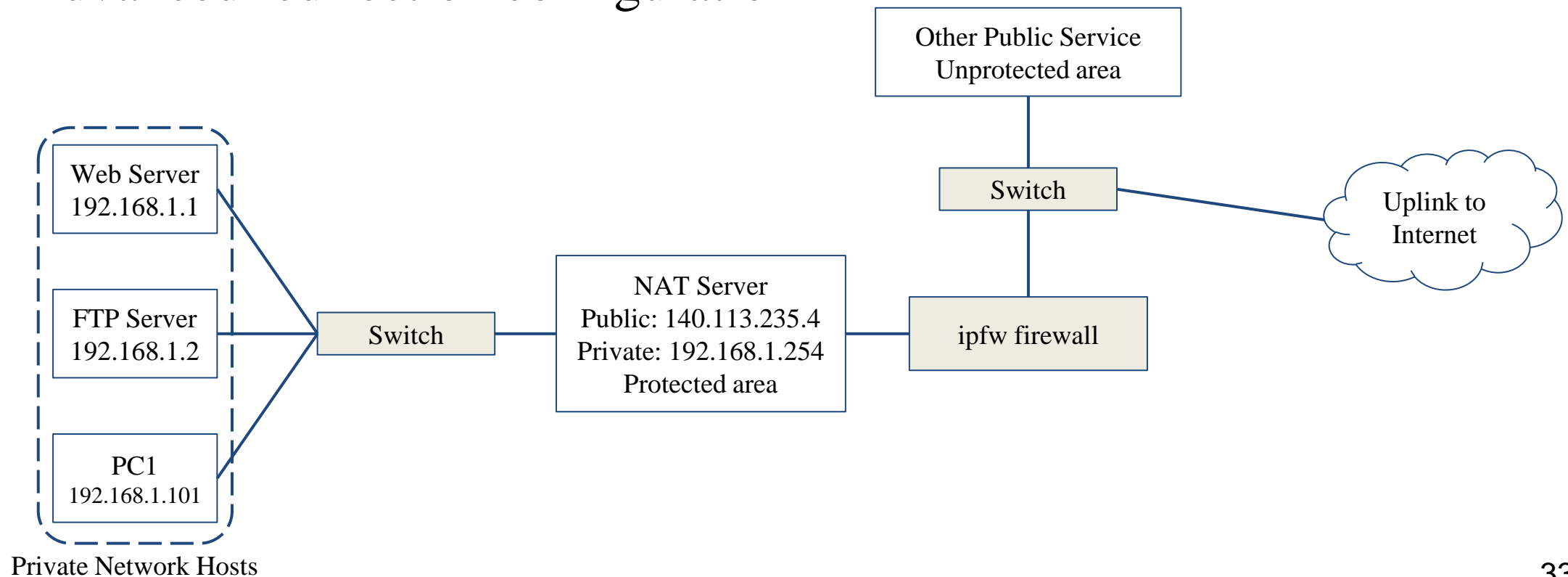
- DNAT
 - Rewrite the destination IP and/or Port.
 - The rewritten packet will be redirected to another IP address when it pass through NAT server.



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

NAT 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 21
```

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
```