Troubleshooting Addressing Services

CCNP TSHOOT 6

Agenda

- NAT
- DHCP
- IPv6 SLAAC/DHCPv6

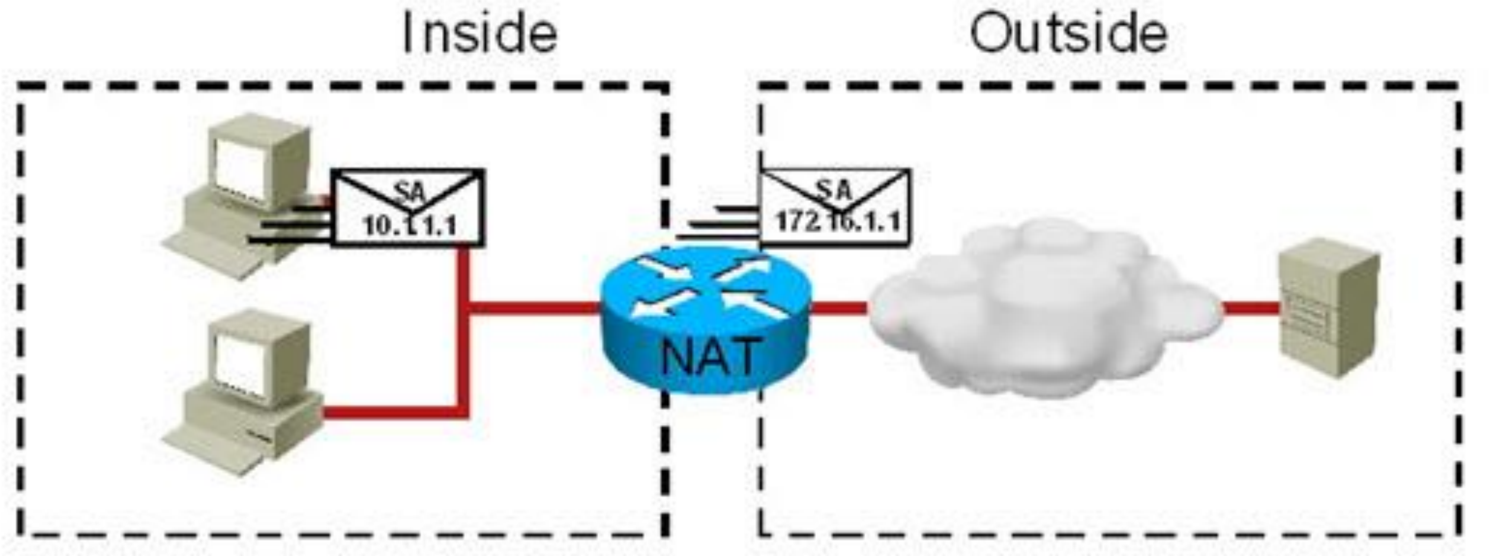
Common IPv4 Addressing Service Issues

NAT/PAT Operation Review

- Network Address Translation (NAT) was designed for IP version 4 (IPv4) address conservation.
  - Also used for address hiding, with security implications.
- Usually operates at the border of a network and translates source address of exiting IP packets that are private addresses to public addresses before packets are forwarded out
  - The packet header information and the corresponding translated IP address are kept in a NAT table
  - NAT does the reverse for the destination address of the responding IP packets based on the content of the NAT table.
- NAT can also be used to renumber global address space when switching between service providers.
- In VPN connectivity situations, NAT can overcome the connectivity issues that arise by translating overlapping address spaces to non-overlapping addresses.

NAT Addresses

- NAT operates at the border of the network where its interfaces are divided as follow
  - NAT Inside uses internal private address
  - NAT Outside uses outside public address
- NAT translates according to the direction
  - SRC NAT
  - DST NAT



NAT Types

- **Static NAT**
  - Inside local (locally significant) and inside global (globally significant) addresses are mapped one to one.
  - Useful when an inside device must be accessible from the outside network (such as a web server).
  - In troubleshooting, IP address changes might affect an existing static configuration.
- **Dynamic NAT**
  - Translates addresses utilizing the same basic technology as static NAT.
  - Local addresses are translated to a group or pool of global addresses.
  - Issues can be related to the size of the global pool (requires one-to-one translation).
  - Some inside hosts may not obtain a valid global address, causing connectivity problems.
  - Subject to management, tracking, and audit issues due to the dynamic nature of the translation.
- **NAT overloading**
  - Special type of dynamic NAT in which addresses are translated in a many-to-many fashion.
  - Also known as PAT, or Port Address Translation, because global addresses can be reused.
  - The differentiator for multiple inside local addresses sharing the same global address is a port number.
  - NAT overloading suffers from some application support issues.

NAT-related Caveats

- **Some applications are not NAT friendly**
  - Certain applications such as IP telephony call-setup protocols make a reference to the host IP address (before translation). This will cause the Voice over IP (VoIP) traffic (the actual call) to be dropped, because the IP packet destination addresses are private addresses and unreachable. Usage of these applications with NAT requires special configurations and workarounds.
- **NAT contributes to the total end-to-end delay**
  - Packets subject to NAT experience more delay than they would without NAT. If you experience significant delays due to translations, it is possible that the NAT device is doing excessive NAT translations.
- **Using NAT over a VPN**
  - IP address translation performed by NAT modifies the IP header checksum as well. If the IP header checksum is used in the integrity check performed by a security protocol, the IP packet will be rejected. There are workarounds for these cases.
- **NAT will hide the IP address information**
  - End-to-end troubleshooting can be challenging with NAT. A good understanding of the NAT process is crucial before starting the troubleshooting process.

Pros and Cons

Advantages and disadvantages of implementing NAT:

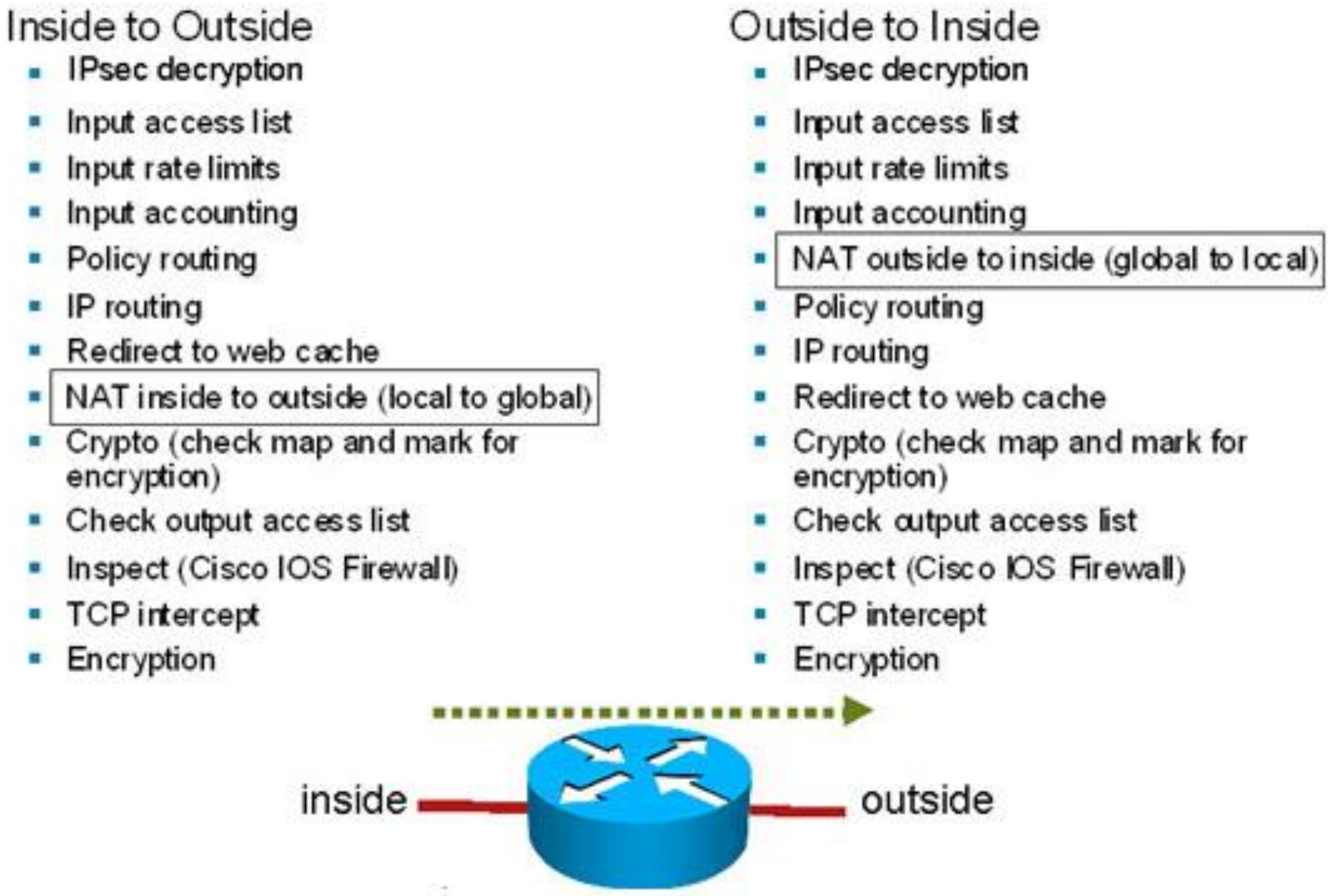
Advantages	Disadvantages
Conserves registered addresses	Translation introduces processing delays
Hides the actual address of internal hosts and services	Loss of end-to-end IP reachability
Increases flexibility when connecting to Internet	Certain applications will not function with NAT enabled
Eliminates address renumbering as the network changes from one ISP to another	Considerations are needed when working with VPNs

NAT-Sensitive Protocols

Protocol	Behavior
IPsec	NAT changes certain IP header fields such as the IP address and the IP header checksum. This can conflict with IPsec integrity.
ICMP	Many ICMP packets, such as Destination Unreachable, carry embedded IP header information inside the ICMP message payload, not matching IP packet’s translated address.
Session Initiation Protocol (SIP)	Protocols such as SIP negotiate address and ports numbers at the application layer, which can become invalid through a NAT device.

IOS Processing

- More about order of operations in Cisco Document ID: 6209, „NAT Order of Operation“



Inside to Outside

- IPsec decryption
- Input access list
- Input rate limits
- Input accounting
- Policy routing
- IP routing
- Redirect to web cache
- NAT inside to outside (local to global)
- Crypto (check map and mark for encryption)
- Check output access list
- Inspect (Cisco IOS Firewall)
- TCP intercept
- Encryption

Outside to Inside

- IPsec decryption
- Input access list
- Input rate limits
- Input accounting
- NAT outside to inside (global to local)
- Policy routing
- IP routing
- Redirect to web cache
- Crypto (check map and mark for encryption)
- Check output access list
- Inspect (Cisco IOS Firewall)
- TCP intercept
- Encryption

Troubleshooting NAT Issues

- NAT configuration should be documented and ideally accompanied with diagram
- NAT components include
  - ACLs are used to tell the NAT device "what source IP addresses are to be translated“
  - IP NAT pools are used to specify "to what those addresses translate", as packets go from IP NAT inside to IP NAT outside.
  - Marking the IP NAT inside interfaces and the IP NAT outside interfaces correctly is important.
- NAT packets still have to obey routing protocols and reachability rules
  - Make sure that every router knows how to reach the desired destinations.
  - Make sure the public addresses to which addresses translate are advertised to the outside neighbors and autonomous systems.

Troubleshooting Commands

`clear ip nat translation`

- Removes NAT entries from the NAT table.
- Specific entries can cleared with additional parameters.
- Clearing all translations can cause disruption until new translations are re-created.

`show ip nat translations`

- Displays all the translations (static and dynamic) that are currently installed and active on the router.

`show ip nat statistics`

- Displays NAT statistics such as number of translations (static, dynamic, extended), number of expired translations, number of hits (match), number of misses (no match).

`debug ip nat`

- Displays information about each packet that the router translates.

`debug ip nat detailed`

- Generates a description of each packet considered for translation.
- Also displays information about certain errors or exception conditions, such as the failure to allocate a global address.

`debug ip packet [`*`access-list`*`]`

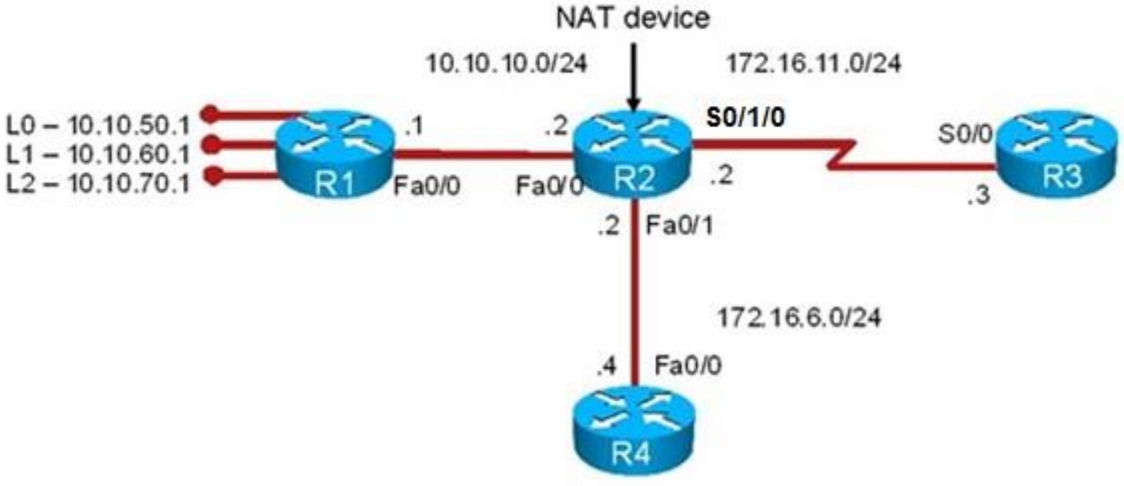
- Displays general IP debugging information and IP security option (IPSO) security transactions.
- If a communication session is closing when it should not be, an end-to-end connection problem can be the cause.
- Useful for analyzing messages traveling between the local and remote hosts.
- Captures packets that are process switched including received, generated, and forwarded packets.
- IP packets that are switched in the fast path are not captured.
- The *`access-list`* option allows you to narrow down the scope of debugging.

The `debug conditional` Command

- **`debug condition interface`** *`interface`*
  - Called conditionally triggered debugging.
  - Generates debugging messages for packets entering or leaving on the specified interface.
  - Will not generate debugging output for packets for a different interface.
  - First define the condition with the **`debug condition`** command. For example, define a condition of **`interface serial 0/0`**.
  - This definition means that all debug output will be limited to that particular interface.
  - The condition remains defined and applied until it is removed.
  - Check the active debug conditions using the **`show debug condition`** command.

NAT/PAT 1st Example ①

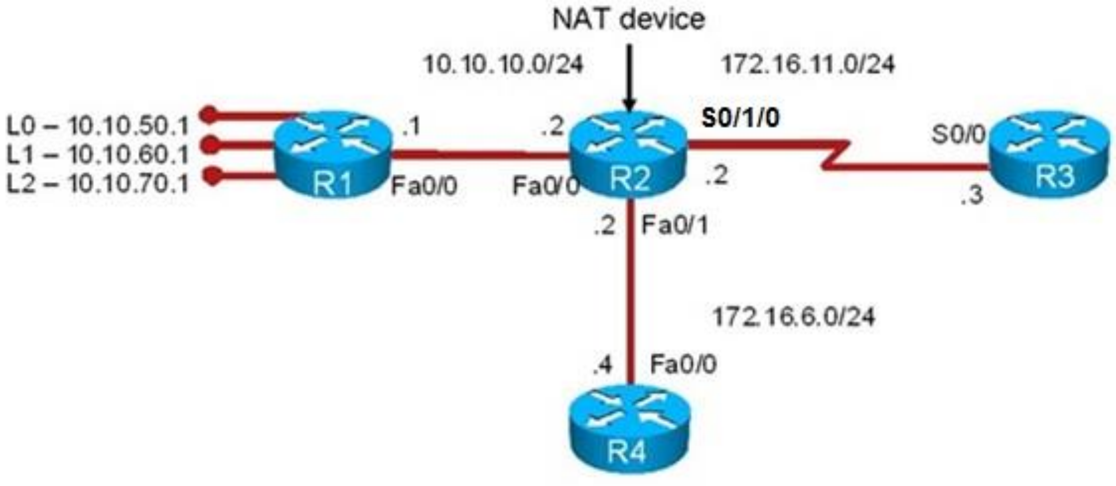
- Router R1 can ping R4, but router R1 cannot ping R3.
- There are no routing protocols running in any of the routers.
- R1 uses R2 as its gateway of last resort.
- The objective is to restore end-to-end connectivity from R1 to all destinations.



NAT/PAT 1st Example ②

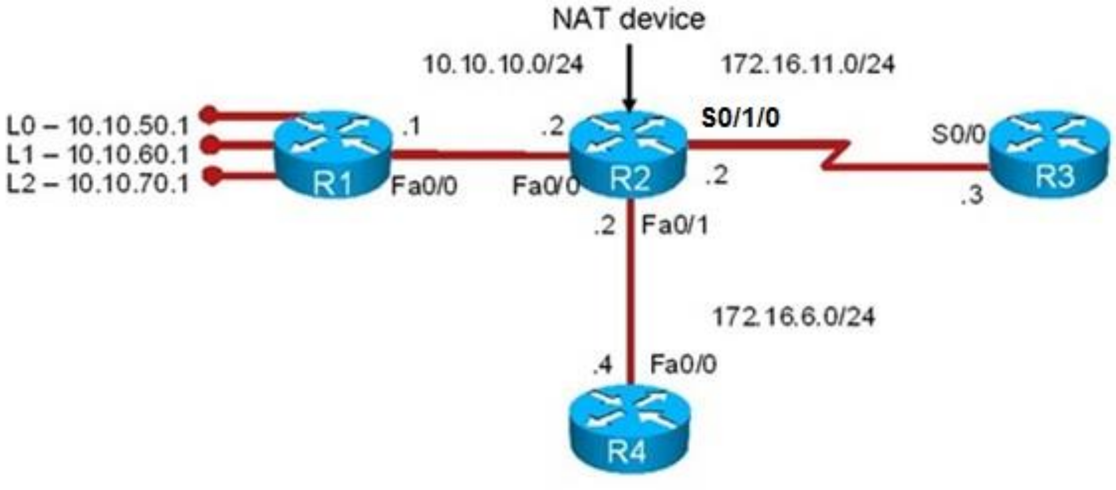
```
R2# sh ip nat statistics
Total active translations: 1 (1 static, 0 dynamic, 0 extended)
Outside interfaces:
  FastEthernet0/1, Serial0/1/0

Inside interfaces:
  FastEthernet0/0

Hits: 39  Misses: 6
CEF Translated packets: 45, CEF Punted packets: 49
Expired translations: 6
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 10 pool NAT_OUT refcount 0
 pool NAT_OUT: netmask 255.255.255.0
        start 172.16.6.129 end 172.16.6.240
        type generic, total addresses 112, allocated 0 (0%), misses 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

NAT/PAT 1st Example ③



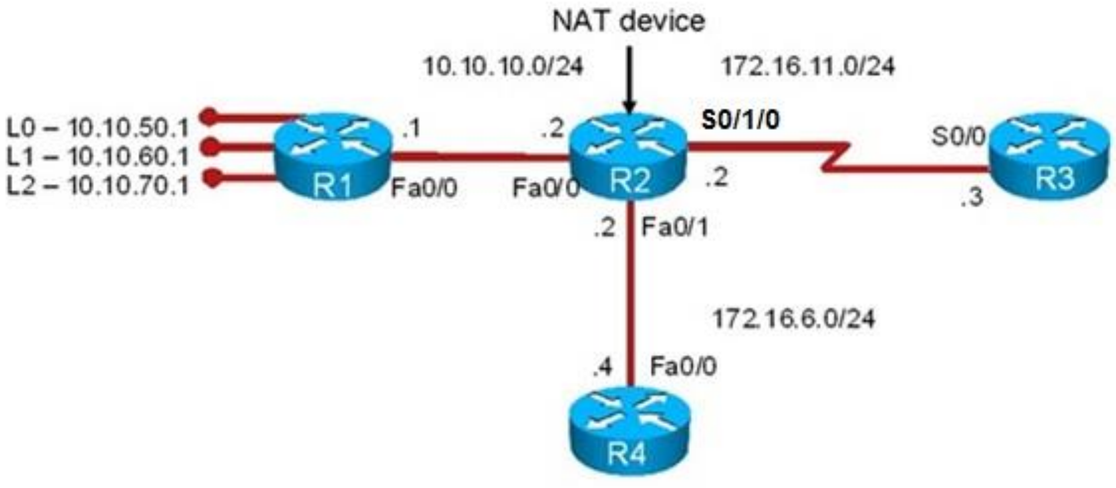
```
R2# sh ip nat translations
Pro   Inside global   Inside local     Outside local     Outside global
---   172.16.6.1      10.10.10.1         ---               ---
```

NAT/PAT 1st Example ④



```
R3# debug ip icmp
ICMP packet debugging is on

R1# ping 172.16.11.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R3#
*Aug 23 13:54:00.556:  ICMP:  echo reply sent, src 172.16.11.3,  dst 172.16.6.1
*Aug 23 13:54:02.552:  ICMP:  echo reply sent, src 172.16.11.3,  dst 172.16.6.1
*Aug 23 13:54:04.552:  ICMP:  echo reply sent, src 172.16.11.3,  dst 172.16.6.1
*Aug 23 13:54:06.552:  ICMP:  echo reply sent, src 172.16.11.3,  dst 172.16.6.1
*Aug 23 13:54:07.552:  ICMP:  echo reply sent, src 172.16.11.3,  dst 172.16.6.1
```

NAT/PAT 1st Example ⑤



```
R3# show ip route 172.16.6.0 255.255.255.0
% Subnet not in table

R3# configure terminal
R3(config)# ip route 172.16.6.0 255.255.255.0 172.16.11.2
R3(config)# exit

R1# ping 172.16.11.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R1#
```

Common NAT Issues

- Incorrect or missing identification of inside/outside interface
  - Application of `ip nat [inside|outside]` on physical interface instead of subinterface
- Incorrect ACL configuration targeting source addresses
  - Statement `permit any` is not supported and may lead to multiple translation problems
  - Statical translations should be explicitly excluded from dynamic translation rules
- Missing keyword `inside` in `ip nat` config
  - `ip nat source` is intended for different NAT use-case (see Command Reference for more) and it is NOT compatible with `ip nat inside source`
- Public address space of NAT pools is not known to routing tables
  - Create at least Null0 static and redistribute it
- ACL on interfaces that does not take into account packet after NAT
  - In ACL is checked before NAT and out ACL after NAT on **inside** interface
  - In ACL is checked after NAT and out ACL before NAT on **outside** interface

Reviewing DHCP Operation

Overview

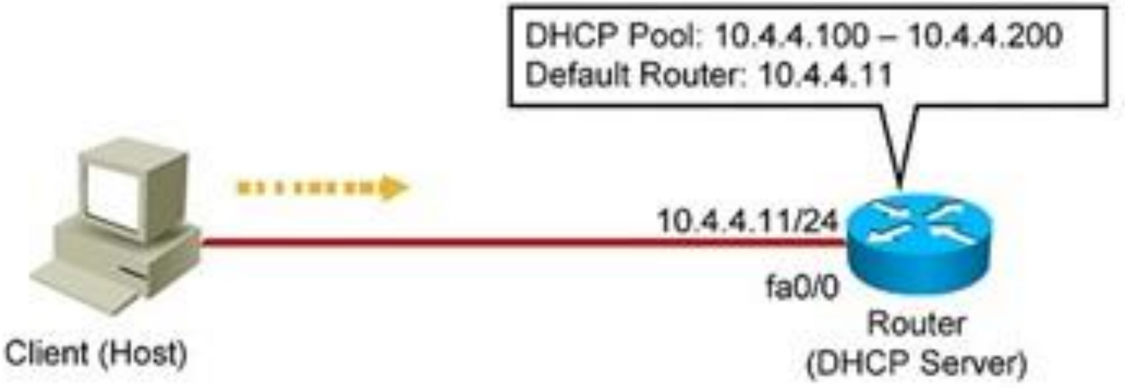
- DHCP is a dominant client/server protocol for automatic host configuration
- Initially it was used just for dynamic IP address assignment, however today it is capable of providing various network parameters
  - IP address of WLC controller
  - TFTP server adddress
  - WINS server address
  - NTP address
  - web proxy
  - Boot server for PXE environment
- DHCP parameters are carried as additional DHCP options
  - some of them standardized
  - others could be freely introduced if added to DHCP server implementation

Options

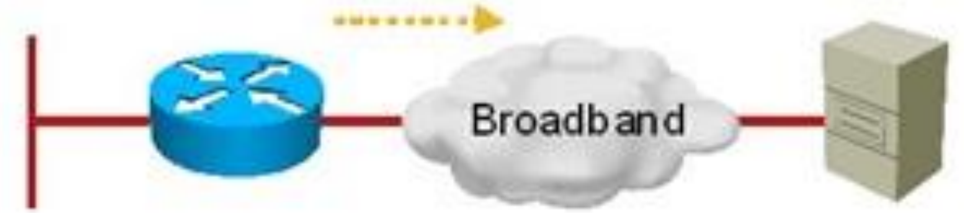
DHCP Option	Code	Description
Subnet Mask	1	Specifies the subnet mask for the client to use (as per RFC 950)
Router	3	The list of routers the client can use (usually, in order of preference)
Domain Name Server	6	The list of DNS servers the client can use (usually, in order of preference)
ARP Cache Timeout	35	Specifies the timeout (seconds) for ARP cache entries
IP Address Lease Time	51	Specifies the period over which the IP address is leased (it must be renewed)
Relay Agent Information	82	Information about the port from which the DHCP request originates
TFTP Server IP Address	150	Typically used by devices such as IP Phones to download their configuration files

Cisco Router DHCP Roles

Router acting as DHCP server

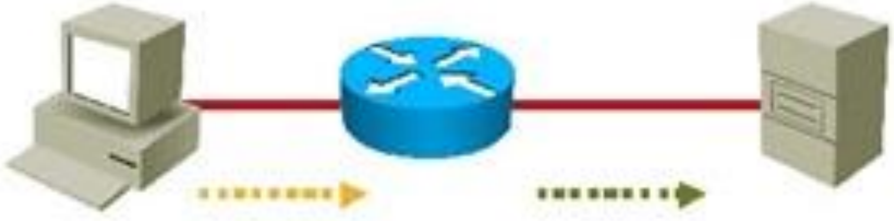


Router acting as DHCP client

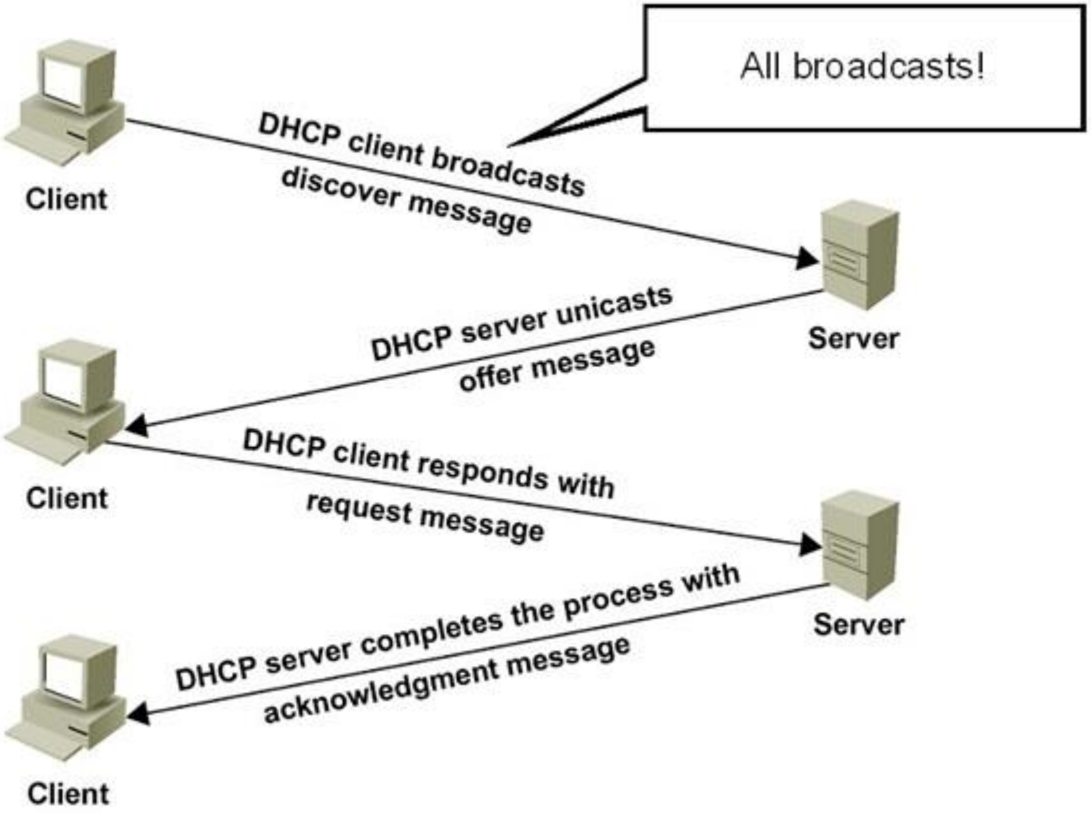


Router brokering DHCP transactions

(DHCP relay agent)



Illustrative Communication


All broadcasts!
Client
DHCP client broadcasts discover message
Server
DHCP server unicasts offer message
Client
DHCP client responds with request message
Server
DHCP server completes the process with acknowledgment message
Client

DHCP Messages

Packet Type	Description
Discover	Client looking for available DHCP servers. It is a UDP broadcast (source port is 68, and the destination port is 67).
Offer	This is the server's response to the client's discover message. This is also a UDP broadcast (source port is 67, and the destination port is 68).
Request	This is client's response to one specific DHCP offer.
Decline	Client-to-server communication, indicating that the IP address is already in use.
Ack	Server-to-client communication. This is the server's response to a client request. This message includes all configuration parameters.
Nack	Server-to-client communication. This is the server's negative response to a client's request, indicating the original offer is no longer available.
Release	Client-to-server communication. The client relinquishes its IP address and other parameters.
Inform	Client-to-server communication. Using this message, the client asks for local configuration parameters such as DNS server's IP address, but it has its IP address externally configured.

Motivation for Relay Agent

- DHCP services hugely leverages broadcasts
  - It allows client without legit IP address (0.0.0.0) to send traffic
  - UDP transport protocol, server listens on port 67, client listens on port 68
- However, broadcast nature limits DHCP only to a single broadcast domain
  - Then each domain should have own DHCP server which apparently does not scale very well
  - DHCP Relay agent is implemented as solution to previous use-case
  - **Relay agent** delegates undirected broadcast received on a particular interface and sends them as unicast towards selected server (IP address). Responses from server are backpropagated to original sender.

DHCP Relay Agent

- The Cisco IOS command that makes a router a DHCP relay agent is **`ip helper-address`**.
  - This is an interface configuration command that makes the router forward the BootP/DHCP requests from clients to the DHCP server.
  - `IF` the DHCP server's IP address changes `THEN` all interfaces of all routers MUST be reconfigured with the new IP helper-address (DHCP server's new IP address)
  - Command could be present even multiple times on a single interface
- Enabling a router interface with the **`ip helper-address`** command makes the interface forward UDP broadcasts for six protocols (not just DHCP) to the IP address
  - TFTP (port 69)
  - DNS (port 53)
  - Time Service (port 37)
  - NetBIOS Name Service and Datagram Service (ports 137 and 138)
  - TACACS (port 49)
  - DHCP/BOOTP Client and Server (ports 67 and 68)
- `IF` other protocols do not require this service `THEN` forwarding their requests must be disabled manually on all routers using the Cisco IOS **`no ip forward-protocol udp`** *`port-number`* command in global config mode

DHCP Issues ①

- **Configuration issues** can result in many symptoms
  - Clients not obtaining IP information from the server
  - Client requests not reaching the server across a DHCP relay agent
  - Clients failing to obtain DHCP options and extensions
- **Address pool issues**
  - Poor capacity planning and security issues might result in DHCP scope exhaustion.
  - When using static and dynamic IP address assignments, an IP address that is already in use can be granted.
  - Multiple DHCP servers, or even rogue DHCP servers can result in duplicate IP addresses assigned to hosts.
- **Management issues**
  - Due to the “pull” nature of DHCP.
  - There are no provisions in the protocol to allow the DHCP server to push configuration parameters or control messages to DHCP clients.
  - A good example, with critical implications in IP address renumbering, is that IP addresses must be renewed from the client side. There is no server-side, push-type renewal process.
  - This means that during renumbering, all clients would need to reboot or manually renew their IP addresses. Otherwise, you need to wait until the clients leases expire, which might not be a viable option.

DHCP Issues ②

- **DHCP Snooping Issues**
  - DHCP Snooping protects network against
    - …rogue DHCP servers
    - …sniffing of DHCP traffic
    - …spoofing of DHCP messages
    - …DoS attacks
  - When deployed following should be taken into account
    - Proper trust boundaries configuration
    - Proper activation of DHCP Snooping (per VLAN or globally)
    - Configuration of rate limiting
    - Acceptance of DHCP packets with Option-82 present but without Relay agent address by DHCP server
  - Improper DHCP Snooping deployment could lead to service perform degradation or even DoS

Troubleshooting Questions

- *Where are the DHCP servers and clients located?*
- *Are DHCP relay agents configured?*
- *What are the DHCP pool sizes? Are they sufficient?*
- *Are there any DHCP option compatibility issues?*
- *Are there any ACLs or firewalls filtering UDP port 67 or UDP port 68?*
- *Are there any active DHCP DoS attacks?*
- *Is forwarding disabled on the router acting as DHCP Relay Agent for any UDP ports (using the Cisco IOS* **`no ip forward-protocol udp`** *`port`* *command)?*
- *Is the* **`ip`** ***`helper-address`*** *command applied to correct router interfaces?*
- *Is DHCP snooping configured?*

The `show` Commands

`show ip dhcp server statistics`

- Displays counts for server statistics and messages sent and received for an IOS-based DHCP server.

`show ip dhcp binding`

- Displays DHCP binding information for IP address assignment and subnet allocation.

`show ip dhcp conflict`

- Displays address conflicts found by a Cisco IOS DHCP server when addresses are offered to the client.

`show ip dhcp pool` *`name`*

- Displays the subnets allocated and the current utilization level for the pool or all the pools if the name argument is not used.

`show ip dhcp database`

- **URL:** Specifies the remote file used to store automatic DHCP bindings
- **Read/written:** The last timestamp bindings were read/written from the file server
- **Status:** Indication of whether the last read/write of host bindings was successful
- **Delay:** The amount of time (in seconds) to wait before updating the database
- **Timeout:** The amount of time (in seconds) before the file transfer is aborted
- **Failures/Successes:** The number of failed/successful file transfers

The `debug` Commands

`debug ip udp`

- Displays UDP packets sent and received.
- Can use considerable CPU cycles on the device.

`debug ip dhcp server [packets | events]`

- Enables DHCP server debugging.
- The **`events`** option reports server events such as address assignments and database updates.
- The **`packets`** option decodes DHCP receptions and transmissions.

`clear ip dhcp binding {* |` ***`address`*** **`}`**

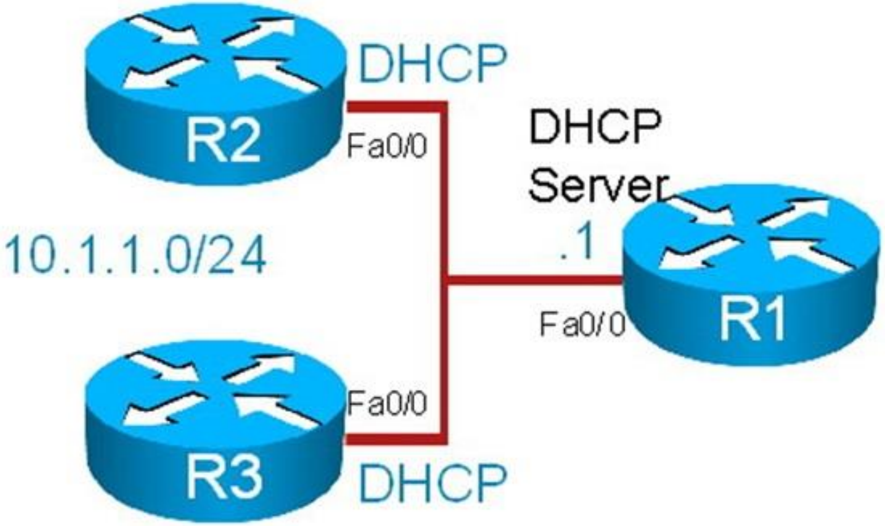
- Deletes an address binding from the DHCP server database.
- The address denotes the IP address of the client.
- If the asterisk (*) character is used as the address parameter, DHCP clears all automatic bindings.

`clear ip dhcp conflict {* |` ***`address`*** **`}`**

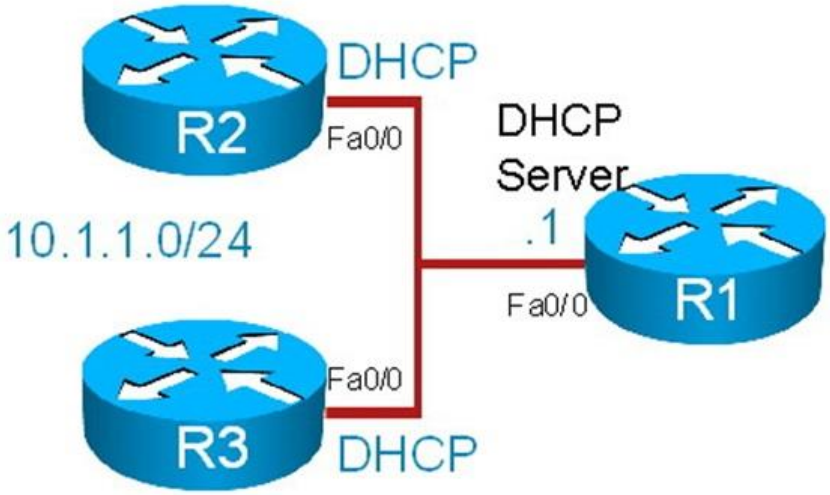
- Clears an address conflict for a specific entry with the *`address`* option.
- Clears all address conflicts with the asterisk (**`*`**) option.

DHCP 1st Example ①

- Router R1 provides DHCP services to clients in the 10.1.1.0 subnet.
- The DHCP clients are R2 and R3.
- A security audit has been recently performed in router R1.
- It is reported that R1 is no longer providing reliable DHCP services.
- The clients are unable to renew their IP addresses.



DHCP 1st Example ②

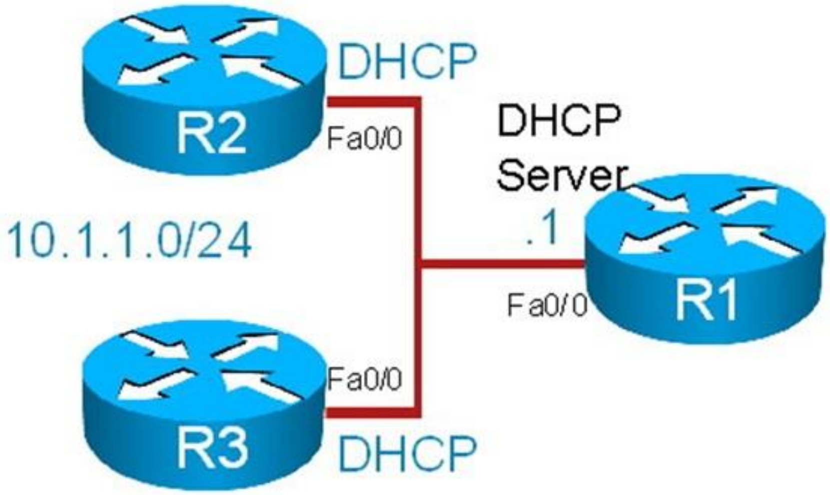


```
R2# show ip int brief
Interface          IP-Address   OK? Method Status                Protocol
FastEthernet0/0    unassigned   YES DHCP   up                    up
FastEthernet0/1    unassigned   YES NVRAM  administratively down down
Serial0/0/0        unassigned   YES NVRAM  administratively down down
Serial0/0/1        unassigned   YES NVRAM  administratively down down

R3# show ip int brief
Interface          IP-Address   OK? Method Status                Protocol
FastEthernet0/0    unassigned   YES DHCP   up                    up
FastEthernet0/1    unassigned   YES NVRAM  administratively down down
Serial0/0/0        unassigned   YES NVRAM  administratively down down
Serial0/0/1        unassigned   YES NVRAM  administratively down down
```

DHCP 1st Example ③

```
R3# debug dhcp detail
DHCP client activity debugging is on (detailed)
R3#

*Aug 23 17:32:37.107: Retry count: 1 Client-ID: cisco-0019.5592.a442-Fa0/0
*Aug 23 17:32:37.107: Client-ID hex dump: 636973636F2D303031392E353539322E
*Aug 23 17:32:37.107: 613434322D4551302F30
*Aug 23 17:32:37.107: Hostname: R3
*Aug 23 17:32:37.107: DHCP: SDiscover: sending 291 byte length DHCP packet
*Aug 23 17:32:37.107: DHCP: SDiscover 291 bytes
*Aug 23 17:32:37.107: B cast on FastEthernet0/0 interface from 0.0.0.0
*Aug 23 17:32:40.395: DHCP: SDiscover attempt #2 for entry:
*Aug 23 17:32:40.395: Temp IP addr: 0.0.0.0 for peer on Interface: FastEthernet0/0
*Aug 23 17:32:40.395: Temp sub net mask: 0.0.0.0
*Aug 23 17:32:40.395: DHCP Lease server: 0.0.0.0, state: 1 Selecting
*Aug 23 17:32:40.395: DHCP transaction id: 13BA
*Aug 23 17:32:40.395: Lease: 0 secs, Renewal: 0 secs, Rebind: 0 secs
*Aug 23 17:32:40.395: Next timer fires after: 00:00:04
*Aug 23 17:32:40.395: Retry count: 2 Client-ID: cisco-0019.5592.a442-Fa0/0
*Aug 23 17:32:40.395: Client-ID hex dump: 636973636F2D303031392E353539322E
*Aug 23 17:32:40.395: 613434322D4551302F30
<output omitted>
*Aug 23 17:32:44.395: Hostname: R3
*Aug 23 17:32:44.395: DHCP: SDiscover: sending 291 byte length DHCP packet
*Aug 23 17:32:44.395: DHCP: SDiscover 291 bytes
*Aug 23 17:32:44.395: B cast on FastEthernet0/0 interface from 0.0.0.0
*Aug 23 17:32:48.395: DHCP: Qscan: Timed out Selecting state
%Unknown DHCP problem... No allocation possible
*Aug 23 17:32:57.587: DHCP: waiting for 60 seconds on interface FastEthernet0/0
```

DHCP 1st Example ④

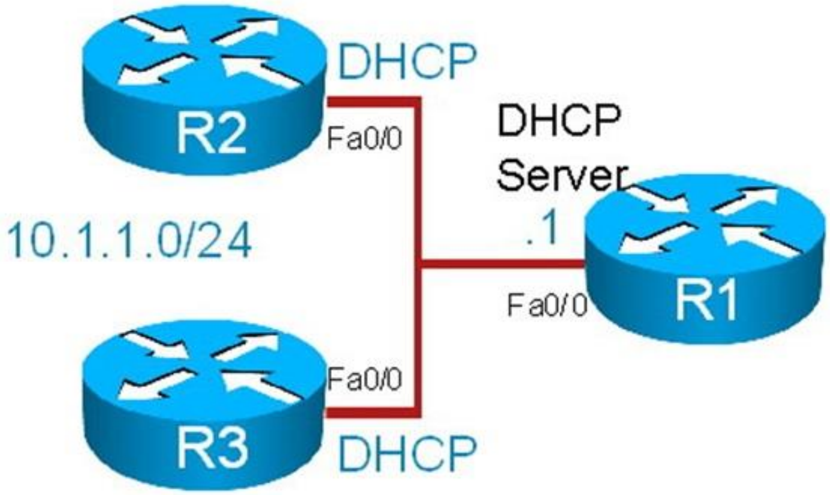


```
R1# show ip int brief
Interface          IP-Address   OK? Method Status                Protocol
FastEthernet0/0    10.1.1.1     YES manual up                    up
FastEthernet0/1    unassigned   YES NVRAM  administratively down down
Serial0/0/0        unassigned   YES NVRAM  administratively down down
Serial0/0/1        unassigned   YES NVRAM  administratively down down
```

DHCP 1st Example ⑤

```
R1# show ip dhcp server statistics
Memory usage        9106
Address pools       1
Database agents     0
Automatic bindings  0
Manual bindings     0
Expired bindings    0
Malformed messages  0
Secure arp entries  0

Message             Received
BOOTREQUEST         0
DHCPDISCOVER        1
DHCPREQUEST         1
DHCPDECLINE         0
DHCPRELEASE         0
DHCPINFORM          0
Message Semt
BOOTREPLY           0
DHCPOFFER           1
DHCPACK             1
DHCPNAK             0

R1# sh ip dhcp pool
Pool vlan10 :
Utilization mark (high/low) : 100/0
Subnet size (first/next)    : 0/0
Total addresses             : 254
Leased addresses            : 0
Pending event               : none
1 subnet is currently in the pool :
Current index    IP address range            Leased addresses
10.1.1.12        10.1.1.1 -10.1.1.254        0
```

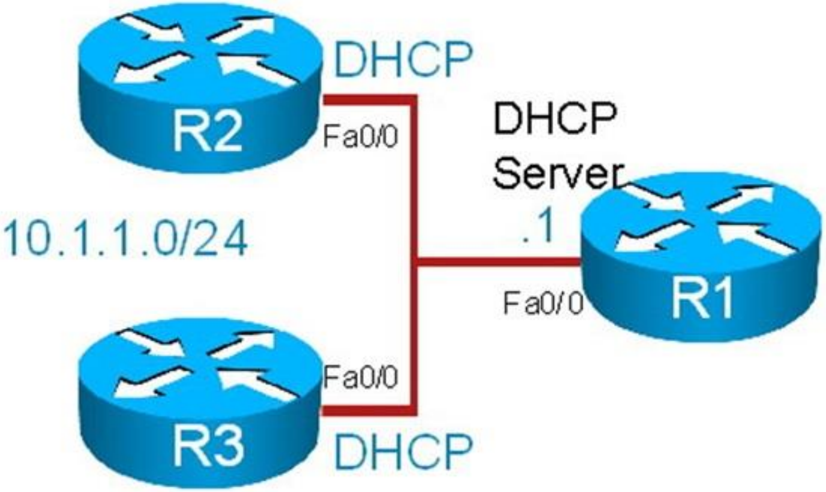
DHCP 1st Example ⑥



```
R1# show ip sockets
Proto  Remote    Port   Local      Port  In Out   Stat  TTY  OutputIF
88   --listen--      10.1.1.1      10  0     0      0    0
17   --listen--      10.1.1.1     161  0     0   1001    0
17   --listen--      10.1.1.1     162  0     0   1011    0
17   --listen--      10.1.1.1   57767  0     0   1011    0
17   --listen--      --any--      161  0     0   20001   0
17   --listen--      --any--      162  0     0   20011   0
17   --listen--      --any--    60739  0     0   20011   0
R1#
```

Note: There is no entry for UDP port 67 (DHCP server)

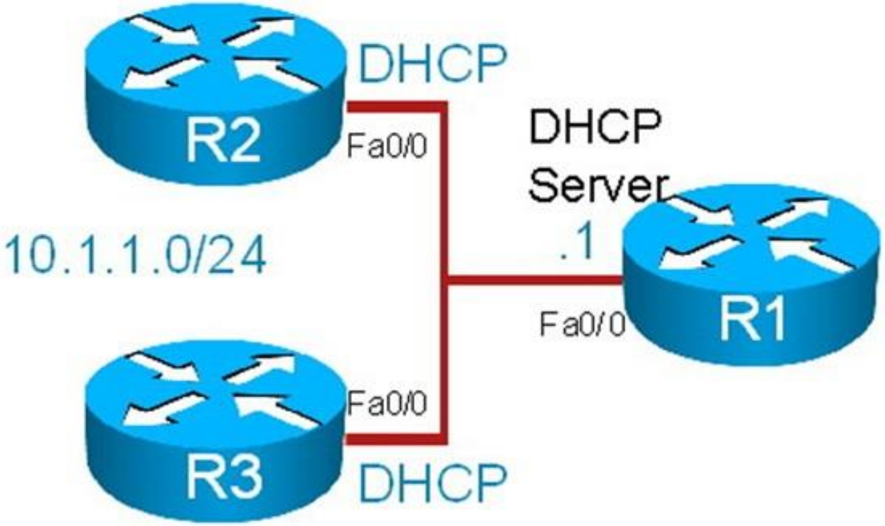
DHCP 1st Example ⑦



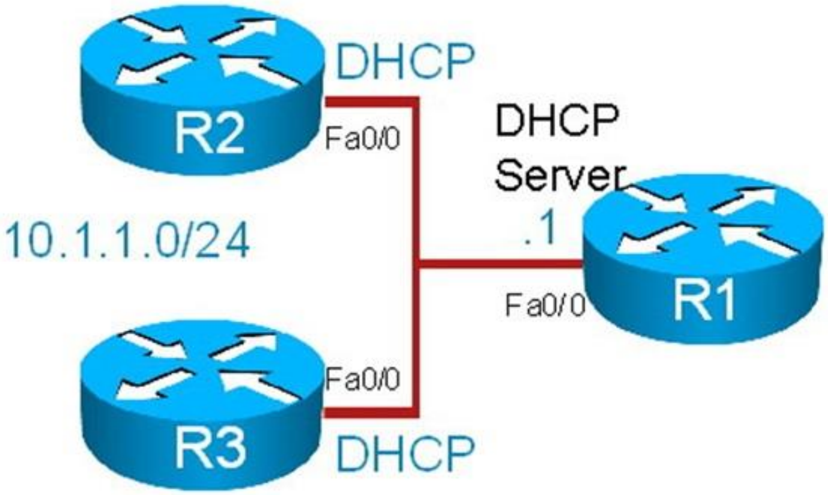
```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# service dhcp
R1(config)# end
R1#

R1# show ip sockets
Proto  Remote   Port    Local    Port  In Out   Stat  TTY  OutputIF
88   --listen--      10.1.1.1      10  0    0      0    0
17   --listen--      10.1.1.1     161  0    0   1001    0
17   --listen--      10.1.1.1     162  0    0   1011    0
17   --listen--      10.1.1.1   57767  0    0   1011    0
17   --listen--      --any--      161  0    0   20001   0
17   --listen--      --any--      162  0    0   20011   0
17   --listen--      --any--    60739  0    0   20011   0
17 0.0.0.0         0 10.1.1.1      67  0    0    2211   0
R1#
```

DHCP 2nd Example ①

- In this scenario, the IP address of router R1 Fa0/0 was previously 10.1.1.100.
- It has been changed to 10.1.1.1 to comply with a new network policy. This policy states that all branch routers will have the first IP address on any subnet
- After the change, some DHCP clients are reporting duplicated IP addresses. Users state that this happens sporadically, a few times a week.

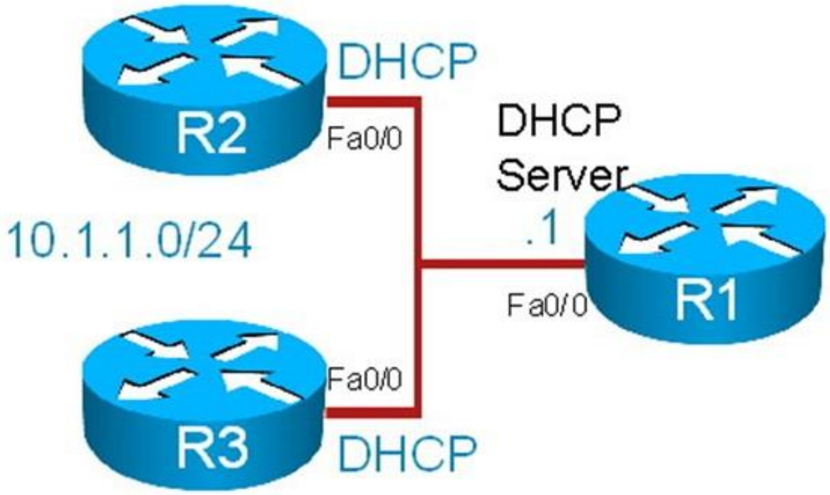


DHCP 2nd Example ②



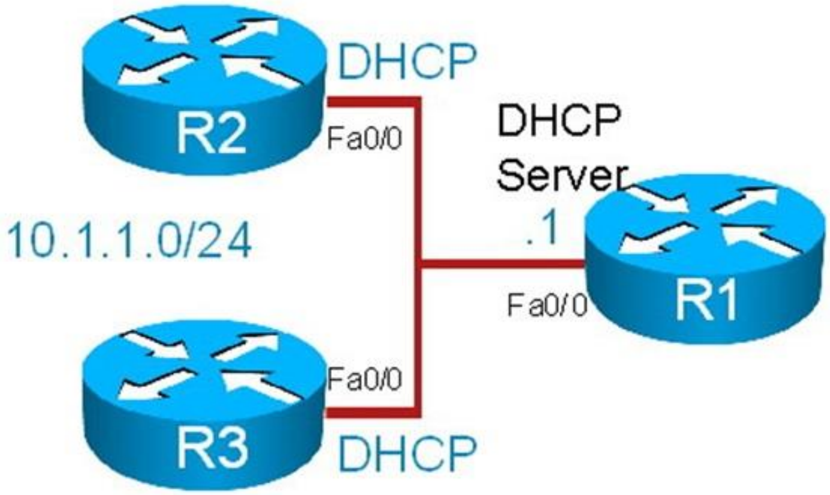
```
R1# show running-config | beg ip dhcp pool

ip dhcp pool vlan10
network 10.1.1.0 255.255.255.0
default-router 10.1.1.1
lease 3
```

DHCP 2nd Example ③



```
R1# show ip dhcp conflict
IP address         Detection method     Detection time           VRF
10.1.1.1           Gratuitous ARP       Aug 23 2009 06:28 PM
10.1.1.3           Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.3           Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.4           Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.5           Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.6           Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.7           Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.8           Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.9           Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.10          Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.11          Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.12          Gratuitous ARP       Aug 23 2009 06:29 PM
10.1.1.13          Gratuitous ARP       Aug 23 2009 06:29 PM
--More--
```

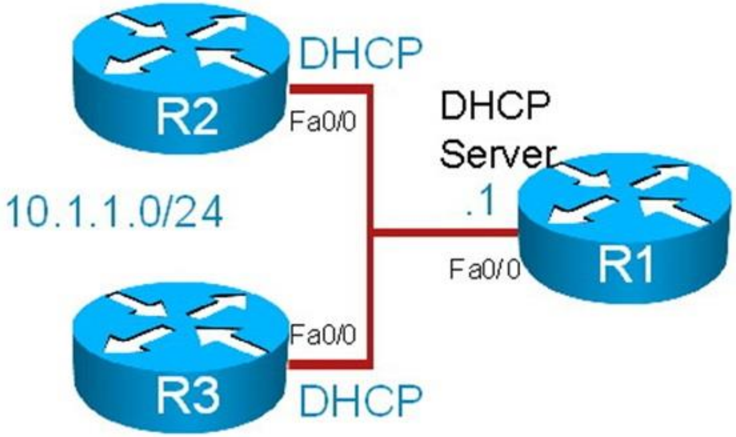
DHCP 2nd Example ④



```
R1# sh run | inc excluded

ip dhcp excluded-address 10.1.1.100

R1#
```

DHCP 2nd Example ⑤

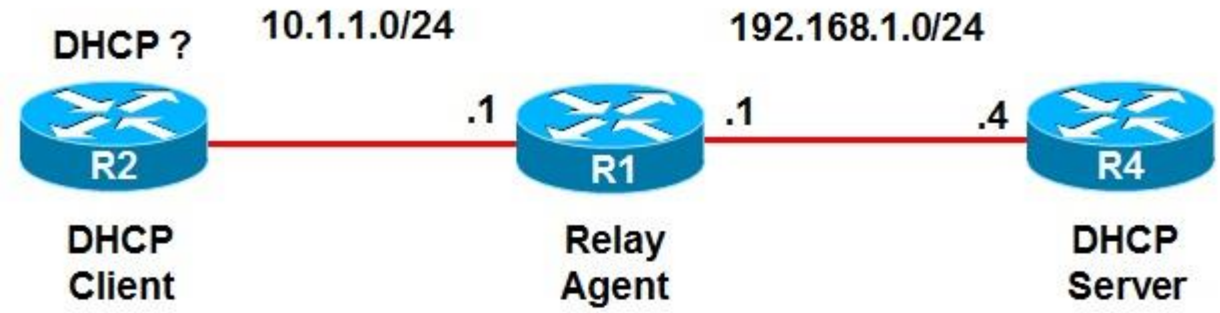
Note: Configure R1 to exclude the range of addresses that are to be reserved for static assignment.



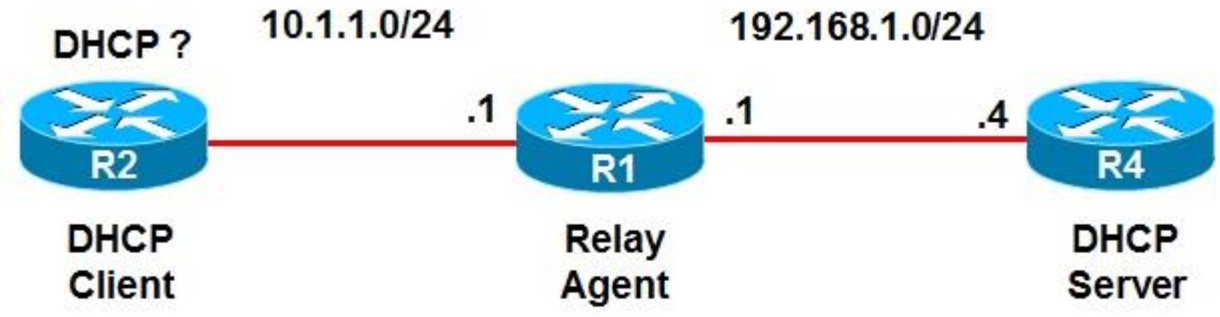
```
R1# conf t
Enter configuration commands, one per line. End with CNTL/Z.

R1(config)# no ip dhcp excluded-address 10.1.1.100
R1(config)# ip dhcp excluded-address 10.1.1.1 10.1.1.20
R1(config)# end
R1#
```

DHCP 3rd Example ①

- R4 is a centrally located DHCP server.
- The DHCP clients in network segment 10.1.1.0 are unable to obtain IP address and other parameters.
- R2 is a DHCP client that is having trouble acquiring ip address.
- R1 is supposed to act as a relay agent and forward DHCP messages between local clients and the DHCP server (R4).



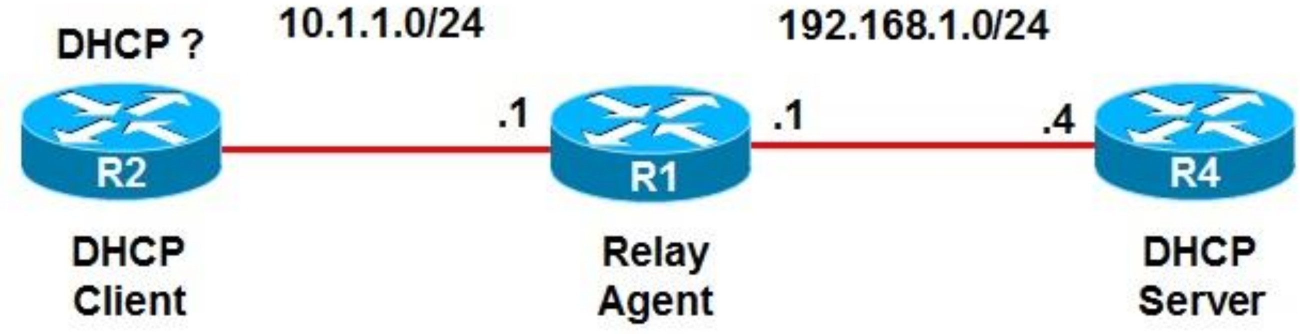
DHCP 3rd Example ②



```
R1# debug ip udp
UDP packet debugging is on
R1#
R1#
*Aug 23 19:01:05.303: UDP: rcvd src-0.0.0.0(68), dst=255.255.255.255(67),
length=584
*Aug 23 19:01:05.303: UDP: broadcast packet dropped, src=0.0.0.0,
dst=192.168.1.255
*Aug 23 19:01:08.911: UDP: rcvd src-0.0.0.0(68), dst=255.255.255.255(67),
length=584
*Aug 23 19:01:08.911: UDP: broadcast packet dropped, src=0.0.0.0,
dst=192.168.1.255
*Aug 23 19:01:12.911: UDP: rcvd src-0.0.0.0(68), dst=255.255.255.255(67),
length=584
*Aug 23 19:01:12.911: UDP: broadcast packet dropped, src=0.0.0.0,
dst=192.168.1.255
<output omitted>
```

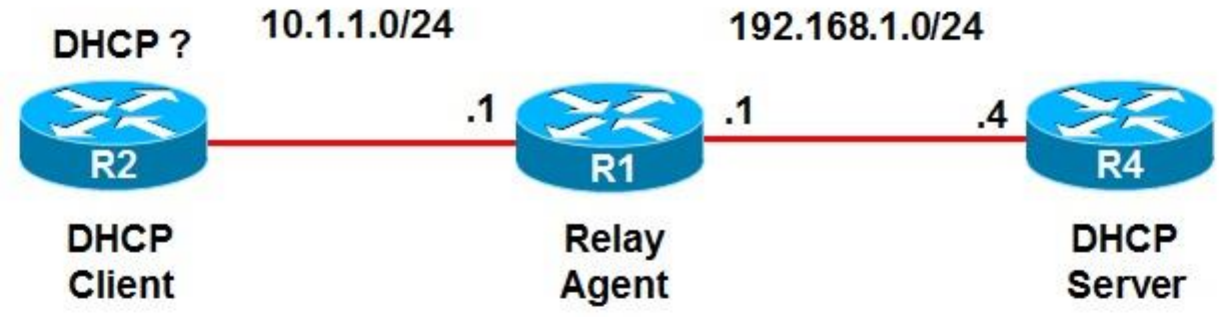
DHCP 3rd Example ③

Note: Configure R1 with a helper address to forward DHCP requests to R4.



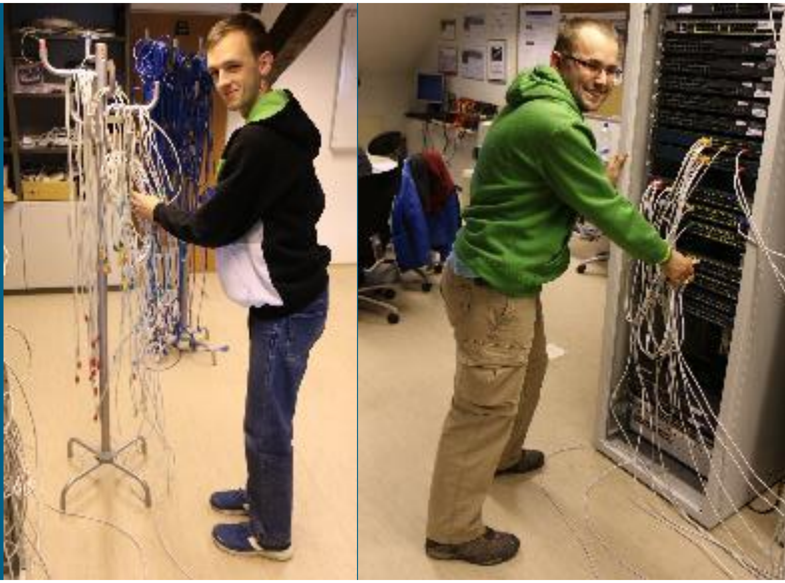
```
R1(config)# int fa0/0
R1(config-if)# ip helper-address 192.168.1.4
R1(config-if)# end
```

DHCP 3rd Example ④



```
R4# debug ip udp
UDP packet debugging is on
R4#
*Aug 23 19:31:39.303: UDP: sent src=0.0.0.0(67), dst=255.255.255.255(68),length=308
*Aug 23 19:31:39.303: UDP: rcvd src=0.0.0.0(68), dst=255.255.255.255(67),length=584
*Aug 23 19:31:39.303: UDP: sent src=0.0.0.0(67), dst=255.255.255.255(68),length=308
*Aug 23 19:31:40.159: UDP: rcvd src=0.0.0.0(68), dst=192.168.1.4(67), length=584
*Aug 23 19:31:44.159: UDP: rcvd src=0.0.0.0(68), dst=192.168.1.4(67), length=584
*Aug 23 19:31:46.307: UDP: rcvd src=10.1.1.11(53470), dst=255.255.255.255(69),length=30
*Aug 23 19:31:49.307: UDP: rcvd src=10.1.1.11(53470), dst=255.255.255.255(69),length=30
<output omitted>
*Aug 23 19:32:28.439: UDP: rcvd src=10.1.1.11(53470), dst=255.255.255.255(69),length=29
*Aug 23 19:32:31.439: UDP: rcvd src=10.1.1.11(53470), dst=255.255.255.255(69),length=29
*Aug 23 19:32:35.439: UDP: rcvd src=10.1.1.11(53470), dst=255.255.255.255(69),length=29
*Aug 23 19:32:37.011: UDP: rcvd src=0.0.0.0(68), dst=192.168.1.4(67), length=584
```

SLAAC, DHCPv6



IPv6 Routing Overview – IPv6 vs IPv4

- IPv6 has many similarities with IPv4 and that makes information gathering and analysis processes similar as well.
- Commands that contain the word “IP” maintain most of their syntax in terms of structure.
- IPv6 and IPv4 similarities represent a benefit but their differences can affect the troubleshooting process.
- There are no broadcasts in IPv6. Neighbors are discovered through ICMPv6 multicasts.
- The addressing structure is still hierarchical, and CIDR, or classless inter-domain routing rules and nomenclature still apply.
- The IPv6 subnet mask is potentially longer, for example /96, or even /128, as compared to /32 (at the most) with IPv4.
- OSPFv3 is still a link state protocol and neighbor sessions, LSAs, hierarchical areas, etc. still exist.

IPv6 Routing Overview – IPv6 vs IPv4 – Cont.

- For almost every IPv4 command there is an IPv6 counterpart
  - The `show ip route` command is becomes `show ipv6 route`
  - The `show ip interface` command becomes `show ipv6 interface`
  - Testing commands, `ping` and `trace` maintain syntax and outcome consistency with their IPv4 counterparts
- IPv6 is much more than a simple expansion in the address space
  - It does away with broadcasts, which affects protocols such as DHCP
  - ARP does not exist
  - Layer 2 addresses are gathered by hosts using the ICMPv6-based neighbor discovery process
  - This process serves other purposes as well, including DAD or Duplicate Address Detection, stateless auto-configuration, and others

Comparison

	IPv4	IPv6
Address Resolution Protocol	Used to find Layer 2 address mappings	Does not exist. ICMPv6 neighbor discovery is used instead
Secondary IP Addresses	Available, but the main IP address is used as packet source	Do not exist. Interfaces can have multiple concurrent IPv6 addresses of different types
Routing Protocols	Use interface IP address to exchange routing information	Use the link local address to create neighbor sessions and to assign as next-hop
Address Assignment	Static, or Dynamic (using DHCP)	Static, or Dynamic (using DHCP or stateless auto-configuration)

Troubleshooting IPv6 Issues - Cont.

- With IPv6 there are common configuration mistakes:
  - A misconfigured auto-configuration router that is not advertising network information to hosts will prevent IPv6 hosts from establishing full connectivity as they lack global unicast addresses.
- Other typical problem areas are related to IPv6 routing protocols and include:
  - Suboptimal routing due to improper summarization
  - Parameter mismatches on protocols such as OSPF that negotiate parameters.
- For tunnel scenarios, other components such as routing protocols, may need to change when the specific migration or tunneling method changes.
  - When using 6to4 tunnels, not all IGPs will function properly, due to the fact that when multicast addresses are used to establish adjacencies, those addresses are not properly mapped to a tunnel destination.

The `show` Commands

`show ipv6 interface`

- Displays the usability status of interfaces configured for IPv6.
- Validates the IPv6 status of an interface and its configured addresses.
- If the interface’s hardware is usable, the interface is marked as up.
- If the interface can provide two-way communication for IPv6, the line protocol is marked as up.

`show ipv6 routers`

- IPv6-specific command (does not have an IPv4 counterpart).
- Displays IPv6 router advertisement (RA) information received from onlink routers.

`show ipv6 route`

- Displays the contents of the IPv6 routing table.

`show ipv6 protocols`

- Displays the parameters and current state of the active IPv6 routing protocol processes.
- The information displayed is useful in troubleshooting routing operations.

The debug Commands

`debug ipv6 routing`

- Displays debugging messages for IPv6 routing table updates and route cache updates.
- Displays messages whenever the routing table changes.

`debug ipv6 nd`

- Displays debugging messages for IPv6 ICMP neighbor discovery (ND) transactions.
- Can help determine whether the router is sending or receiving IPv6 ICMP ND messages.

`debug ipv6 packet`

- Use this command to display debugging messages for IPv6 packets.
- The debugging information includes packets received, generated, and forwarded.
- Note that fast-switched packets do not generate messages.

Manaul IPv6 Address Assigment

- Equipment interface as client

```
R1(config)# ipv6 unicast-routing
R1(config)# interface Fastethernet 0/0
R1(config-if)# ipv6 address 2001:abcd: 1234::1/64
R1(config-if)# ipv6 address 2001:1234:abcd::/64 eui-64
R1(config-if)# ipv6 address fe80::1 link-local
```

Stateless Address Autoconfiguration (SLAAC)

- Equipment interface as client

```
R1(config)# ipv6 unicast-routing
R1(config)# interface Fastethernet 0/0
R1(config-if)# ipv6 address autoconfig
```

- Server Configuration

```
R2(config)# interface fastethernet 0/0
R2(config-if)# no ipv6 nd other-config-flag
R2(config-if)# no ipv6 nd managed-config-flag
```

Automatic configuration with DHCPv6

- Equipment interface as client

```
R1(config)# ipv6 unicast-routing
R1(config)# interface Fastethernet 0/0
R1(config-if)# ipv6 address dhcp
```

- Server Configuration

```
R2(config)# ipv6 dhcp pool test1
R2(config-dhcpv6)# dns-server 2001:1000::1
R2(config-dhcpv6)# domain-name example.com
R2(config-dhcpv6)# address prefix 2001:1000::/64
lifetime infinite
R2(config)# interface fastethernet 0/0
R2(config-if)# ipv6 dhcp server test1 rapid-commit
R2(config-if)# ipv6 nd managed-config-flag
```

Stateful autoconfiguration using DHCPv6

- Equipment interface as client

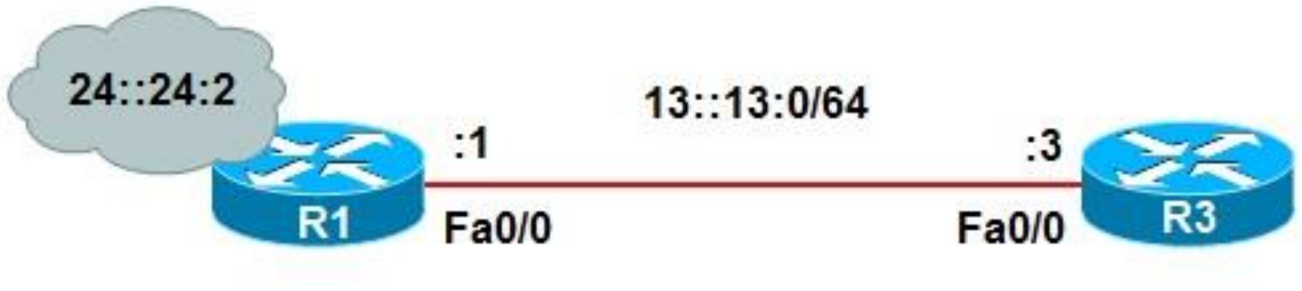
```
R1(config)# ipv6 unicast-routing
R1(config)# interface Fastethernet 0/0
R1(config-if)# ipv6 address autoconfig
```

- Server Configuration

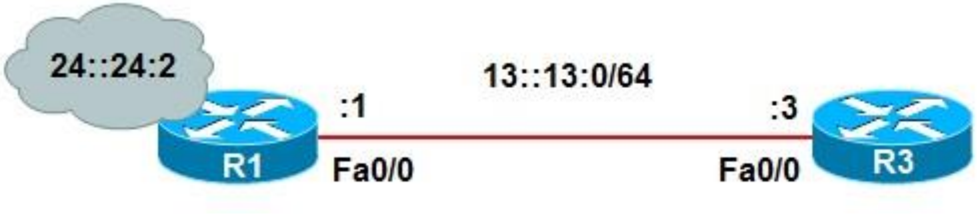
```
R2(config)# ipv6 dhcp pool test1
R2(config-dhcpv6)# dns-server 2001:1000::1
R2(config-dhcpv6)# domain-name example.com
R2(config)# interface fastethernet 0/0
R2(config-if)# ipv6 nd other-config-flag
```

IPv6 Troubleshooting Example 1: IPv6 Routing Problems

- Recent changes in the network have rendered router R3 isolated and with no connectivity outside the fast Ethernet segment connected to R1.
- This is verified by performing a ping from R3 to a remote destination (24::24:2), and it does not succeed.
- The changes were aimed at providing automatic IP address assignment and configuration to certain devices.
- After the change, R3 lost connectivity to the rest of the network.
- A bottom-up troubleshooting approach will be applied to resolve this problem, starting at R3.

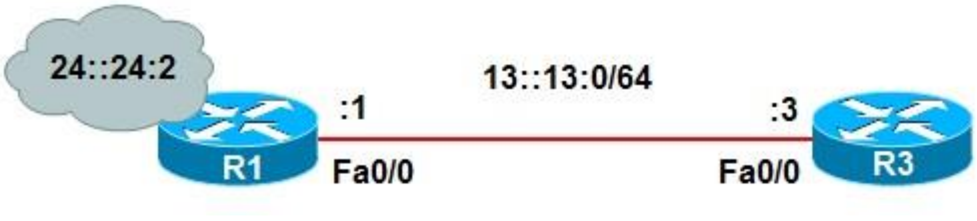


IPv6 Troubleshooting Example 1 – Cont.



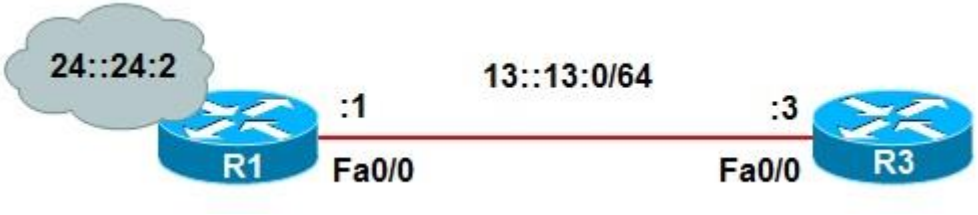
```
R3# show ipv6 interface Fa0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::219:55FF:FEF0:B7D0
  Global unicast address(es):
    13::13:3, subnet is 13::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF13:3
    FF02::1:FFF0:B7D0
  MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

IPv6 Troubleshooting Example 1 – Cont.



```
R3# show run int Fa0/0
Building configuration...
Current configuration : 111 bytes
!
Interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 ipv6 address autoconfig
 ipv6 enable
end
```

IPv6 Troubleshooting Example 1 – Cont.

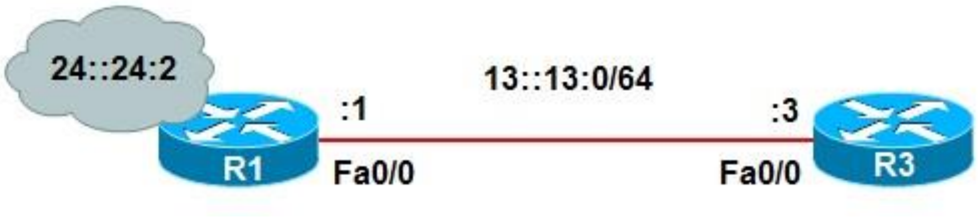


```
R3# show ipv6 route
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static Route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   13::/64 [0/0]
     via ::, Fast Ethernet0/0
L   13:219:55FF;FEF0:B7D0/128 [0/0]
     via ::, FastEthernet0/0
C   103::/64 [0/0]
     via ::, Loopback0
L   103::3/128 [0/0]
     via ::, Loopback0
L   FE80::/10 [0/0]
     via ::, Null0
L   FF00::/8 [0/0]
     via ::, Null0
```

IPv6 Troubleshooting Example 1 – Cont.

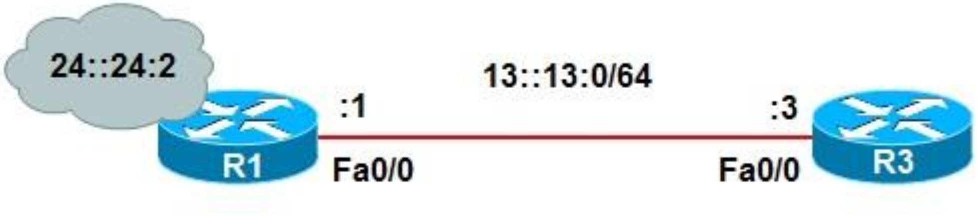
```
R3# debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
R3#
R3#
R3# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)# int Fa0/0
R3(config-if)# shutdown
R3(config-if)# no shutdown
R3(config-if)#
*Aug 23 21:44:18.491: ICMPv6-ND: Sending Final RA on FastEthernet0/0
*Aug 23 21:44:18.491: ICMPv6-ND: Address FE80::219:55FF:FEF0:B7D0/10 is down on
FastEthernet0/0
*Aug 23 21:44:20.491: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to
up
*Aug 23 21:44:20.971: ICMPv6-ND: Sending NS for FE80::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:44:21.971: ICMPv6-ND: DAD: FE80::219:55FF:FEF0:B7D0 is unique
*Aug 23 21:44:21.971: ICMPv6-ND: Sending NA for FE80::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:44:21.971: ICMPv6-ND: Address FE80::219:55FF:FEF0:B7D0 is up on
FastEthernet0/0
*Aug 23 21:44:23.971: ICMPv6-ND: Sending RS on FastEthernet0/0
*Aug 23 21:44:27.971: ICMPv6-ND: Sending RS on FastEthernet0/0
*Aug 23 21:44:31.971: ICMPv6-ND: Sending RS on FastEthernet0/0
```

IPv6 Troubleshooting Example 1 – Cont.



```
R1# show running-config interface f0/0
Building configuration...
Current configuration : 112 bytes
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 ipv6 address 13::13:1/64
 ipv6 enable
end
```

IPv6 Troubleshooting Example 1 – Cont.



```
R1# show ipv6 interface f0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::219:56FF:FE2C:9856
  Global unicast address(es):
    13::13:1, subnet is 13::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF13:1
    FF02::1:FF2C:9856
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  Default router is FE80::219:55F:FEF0:B7D0 on FastEthernet0/0
```

IPv6 Troubleshooting Example 1 – Cont.

- What else does R1 need to become a proper autoconfiguration router?
- How can it be that R3 has a working IPv6 address if the autoconfiguration process is not working?
- Why is it that R3 cannot access the rest of the network, even with a working IPv6 address and no noticeable physical issues?
- One requirement for autoconfiguration is explicitly enabling IPv6 unicast routing.

```
R1# show run | inc unicast-routing
R1#
R1# debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
R1#
R1# conf t
Enter configuration commands, one per line. End with CNTL/A.
R1(config)# ipv6 unicast-routing
R1(config)# end
R1#
*Aug 23 22:01:45.175: ICMPv6-ND: Sending RA to FF02::1 on FastEthernet0/0
*Aug 23 22:01:45.175: ICMPv6-ND: MTU = 1500
*Aug 23 22:01:45.175: ICMPv6-ND: prefix = 13::/64 onlink autoconfig
*Aug 23 22:01:45.175: ICMPv6-ND: 2592000/604800 (valid/preferred)
```

IPv6 Troubleshooting Example 1 – Cont.

The **debug** output on R3 upon enabling IPv6 routing on R1

```
R3# debug ipv6 nd
ICMP Neighbor Dicovery events debugging is on
R3#
R3# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)# int Fa0/0
R3(config-if)# shutdown
R3(config-if)# no shutdown
R3(config-if)#
*Aug 23 21:57:47.547: ICMPv6-ND: Sending Final RA on FastEthernet0/0
*Aug 23 21:57:47.547: ICMPv6-ND: Address 13::219:55FF:FEF0:B7D0/64 is down on
FastEthernet0/0
*Aug 23 21:57:47.547: ICMPv6-ND: Address FE80::219:55FF:FEF0:B7D0/10 is down on FastEthernet0/0
R3#
*Aug 23 21:57:48.003: %SYS-5-CONFIG_I: Configured from console by console
*Aug 23 21:57:53.279: ICMPv6-ND: Sending NS for FE80::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:57:54.279: ICMPv6-ND: DAD: FE80::219:55FF:FEF0:B7D0 is unique
*Aug 23 21:57:54.279: ICMPv6-ND: Sending NA for FE80::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:57:56.279: ICMPv6-ND: Sending RS on FastEthernet0/0
*Aug 23 21:57:56.279: ICMPv6-ND: Received RA from FE80::219:56FF:FE2C:9856 on
FastEthernet0/0
*Aug 23 21:57:56.279: ICMPv6-ND: Sending NS for 13::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:57:56.279: ICMPv6-ND: Autoconfiguring 13::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:57:57.279: ICMPv6-ND: DAD: 13::219:55FF:FEF0:B7D0 is unique
*Aug 23 21:57:57.279: ICMPv6-ND: Sending NA for 13::219:55FF:FEF0:B7D0 on
FastEthernet0/0
*Aug 23 21:57:57.279: ICMPv6-ND: Address 13::219:55FF:FEF0:B7D0 is up on
FastEthernet0/0
```

IPv6 Troubleshooting Example 1 – Cont.

The debug output on R3 upon enabling IPv6 routing on R1

```
R1# sh ipv6 int f0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::219:55FF:FEF0:B7D0
  Global unicast address(es):
    13::219:55FF:FEF0:B7D0, subnet is 13::/64 [PRE]
      Valid lifetime 2591941 preferred lifetime 604741
  Joined group address(es):
   FF02::1
   FF02::2
   FF02::1:FFF0:B7D0
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 0 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses
R1#

R3# ping 24::24:2
Type escape sequence to abort
Sending 5, 100-byte ICMP Echos to 24::24:2, timeout is 2 seconds:
!!!!!
Success reate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
R3#
```

Useful Commands

```
clear ip nat translation
show ip nat translations
show ip nat statistics
debug ip nat
debug ip packet [access-list]
debug condition interface interface
show debug condition
show ip dhcp server
show ip dhcp binding
show ip dhcp conflict
show ip dhcp database
show ip dhcp pool
show ip socket (or show sockets and show udp)
debug ip udp
debug dhcp detail
debug ip dhcp server [packet | event]
clear ip dhcp binding
clear ip dhcp conflict
debug ipv6 routing
debug ipv6 nd
debug tunnel
debug ipv6 packet
show ipv6 interface
show ipv6 routers
show ipv6 route
show ipv6 protocols
```

Slides adapted by Vladimír Veselý and Matěj Grégr
partially from official course materials
but the most of the credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

Last update: 2017-03-10