Networking 9: Address Allocation

i.g.batten@bham.ac.uk

Ethernet Addresses

- Ethernet Addresses (MAC addresses) are normally inherent to the machine.
- 48 bits (6 bytes)
  - 3 bytes identify manufacturer (manufacturers can get multiple blocks)
  - 3 bytes identify device
- Can be changed for good-ish reasons (some old protocols altered the MAC address to match the higher layer address), bad reasons (“MAC spoofing” to evade access controls) and unusual reasons (duplicates are not unheard of, although you would be amazingly unlucky).
- Debate as to whether machines with two interfaces should have one or two addresses: standard says one, reality (and requirements of 2017) says two. Issue only arises now with elderly Sun hardware.

Network Numbers

- Where do you get an IP network from?
  - Small allocations come from your ISP, and belong to your ISP. You have to renumber your network if you use them and change ISP.
  - Larger “provider independent” allocations can be obtained from **regional registries.**
  - You need to make a very good case to get these now
    - We will discuss other issues with PI towards the end of the course.

Where do local IP numbers come from?

- Static Allocation
- bootp (obsolete)
- DHCP/DHCPv6
- SLAAC (IPv6 only)

Static Allocation

- The end point is given an IP number in some sort of configuration file / memory / register.
- Each time the device boots, it gets exactly that IP number, even if it is wrong for the network, clashes with other devices, etc.
- The machine might notice that it is a duplicate, but otherwise has very little protection.

Static Allocation

- Essential for routers and infrastructure devices that need to be up in the very early stages after a power failure.
- Often used for servers that need to have fixed addresses (www.my.domain, mail.my.domain).
  - But there are alternatives for this.

bootp

- Device broadcasts its ethernet address
- “bootp server” hands back an IP number and some other stuff (DNS servers, default router).
- What is the problem with this?

bootp limitations

- RFC951
- Ironically, written by Sun employee number one, but never actually used by Sun as bootp not powerful enough for their requirements.
- Only really works for static assignments by another route (perhaps to machines too limited to be able to store a static address), because no means to reclaim addresses that are no longer used
- Normally configured with a large table statically mapping MAC addresses to IP numbers.
- Obsolete for this reason. Some DHCP servers offer bootp for backwards compatibility: usually now switched off.

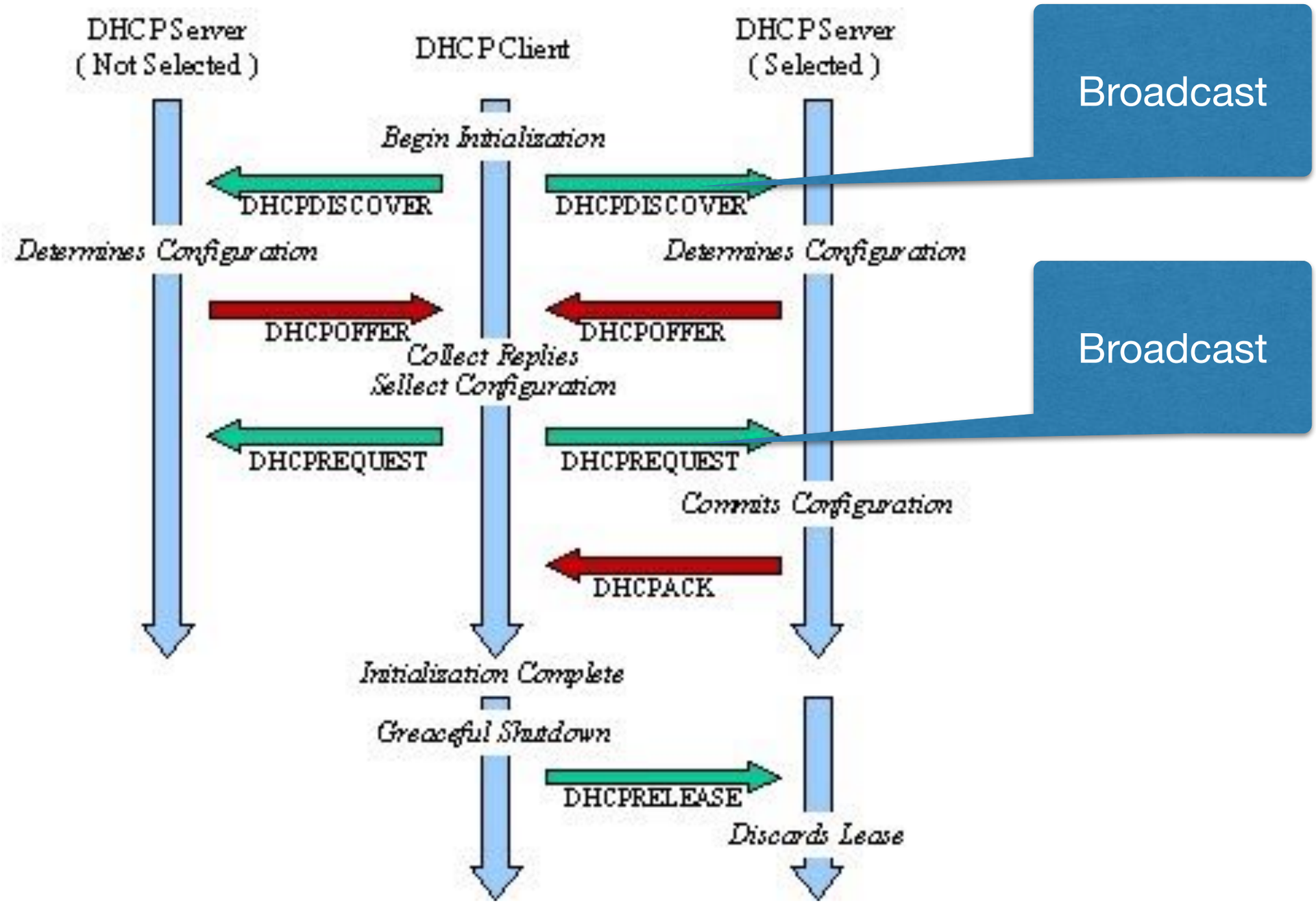
DHCP

- Dynamic Host Configuration Protocol (RFC2131)
- Provides a means to **lease** a temporary IP number for a specified duration, as well as obtaining addresses mappings for routers, DNS, time servers, etc.
- Can also be used to inform devices of a statically allocated IP number
- Main means of allocating IPv4 addresses on LANs in 2017

DHCP Initial Operation

- Client broadcasts a request for an IP number, including its MAC address or some other identifier (DHCPDISCOVER)
- Server(s) reserve an available IP number, and broadcast an offer of it with a lease time (how long IP number is valid for)(DHCPOFFER)
- Client chooses from amongst offers, and broadcasts a reply containing chosen IP number (DHCPREQUEST)
- Server that offered the IP number finishes reserving it and acknowledges (DHCPACK); other servers see that their offer has been declined and unreserve their offer (or wait a decent interval and free the reservation unilaterally)

DHCP Initial Operation



DHCPREQUEST

- Initially, a broadcast containing the IP number (and other information) that the client has decided to use.
- Received by DHCP servers:
  - If it matches what they offered, they know the client is using the offered IP number, so locks the offered IP number for the duration of the lease
  - If it does not match (eg, MAC address is no one they've made an offer to, or claimed IP number is different) then they do nohing

DHCP Request

- But a client can also directly send a request to a known server, in order to renew a lease.
- Conventionally, renewal attempts start after half the lease has passed.

DHCP Static v Pools

- DHCP can work like bootp, always handing out the same IP number for the same MAC address, to configure static machines
- Or it can manage a pool of temporary addresses.
- Smart DHCP servers store the assignments so that if you ask for an address, you always get the same one from the pool unless it has been allocated to someone else in the meantime.
- Very Smart DHCP servers can update DNS servers to record the name to IP binding.

DHCP Redundancy

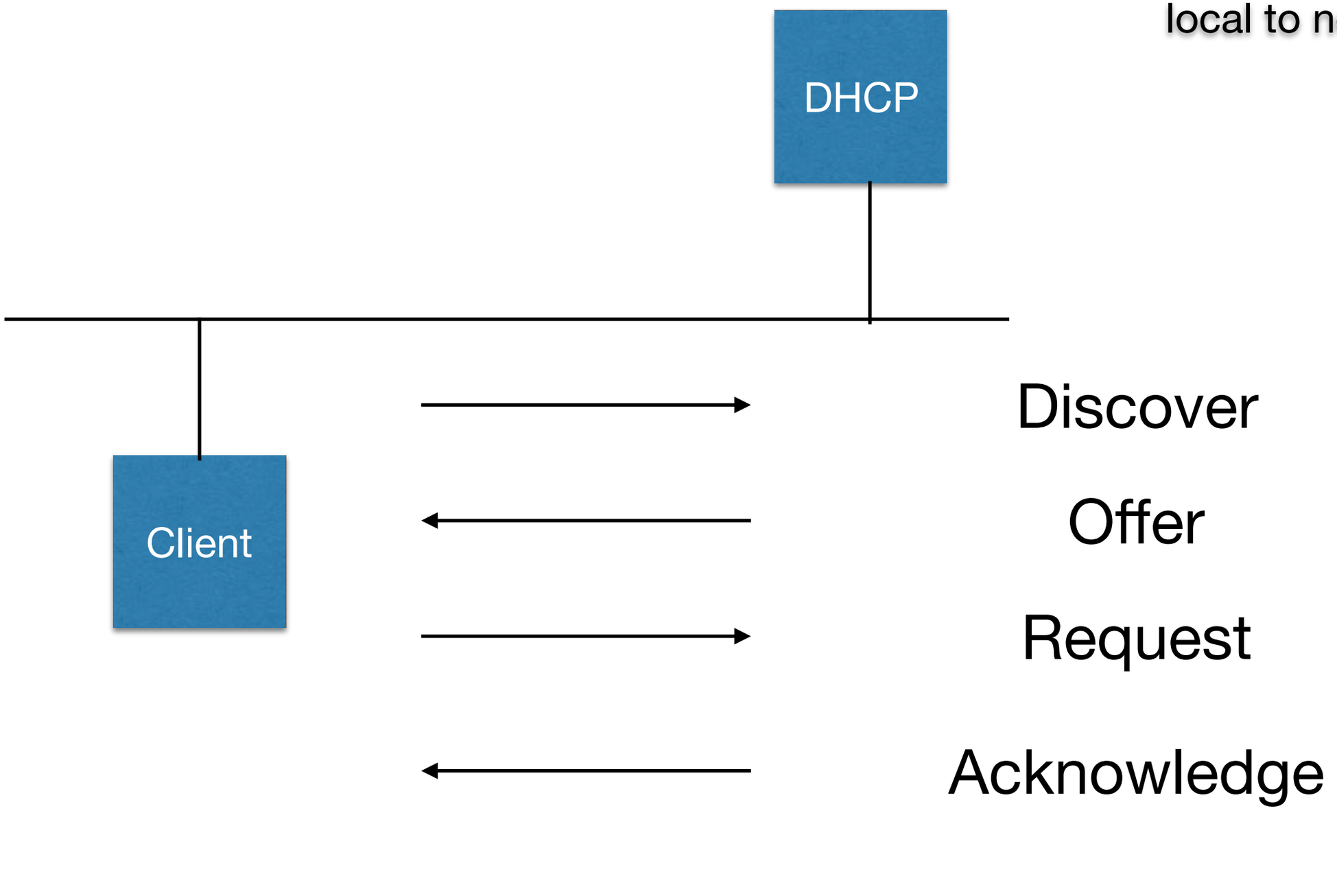
- Loss of DHCP server will wipe out your network.
- But DHCP servers contain delicate state
- Simple home routers can be careless about this, and restart of router may require restart of client devices to get all state into agreement: hence typically use short lease times
- Choices include:
  - two DHCP servers managing disjoint pools and let client select (client IP numbers will change at random intervals)
    - you can have one server lag the other by a second to only be used in an emergency, which is a common trick
  - complex failover protocol so that both servers make the same offer and commit to the same lease (no standards, and implementations are rare)
  - Using higher-level redundancy protocols to have one virtual server (probably best).

DHCP is a mess!

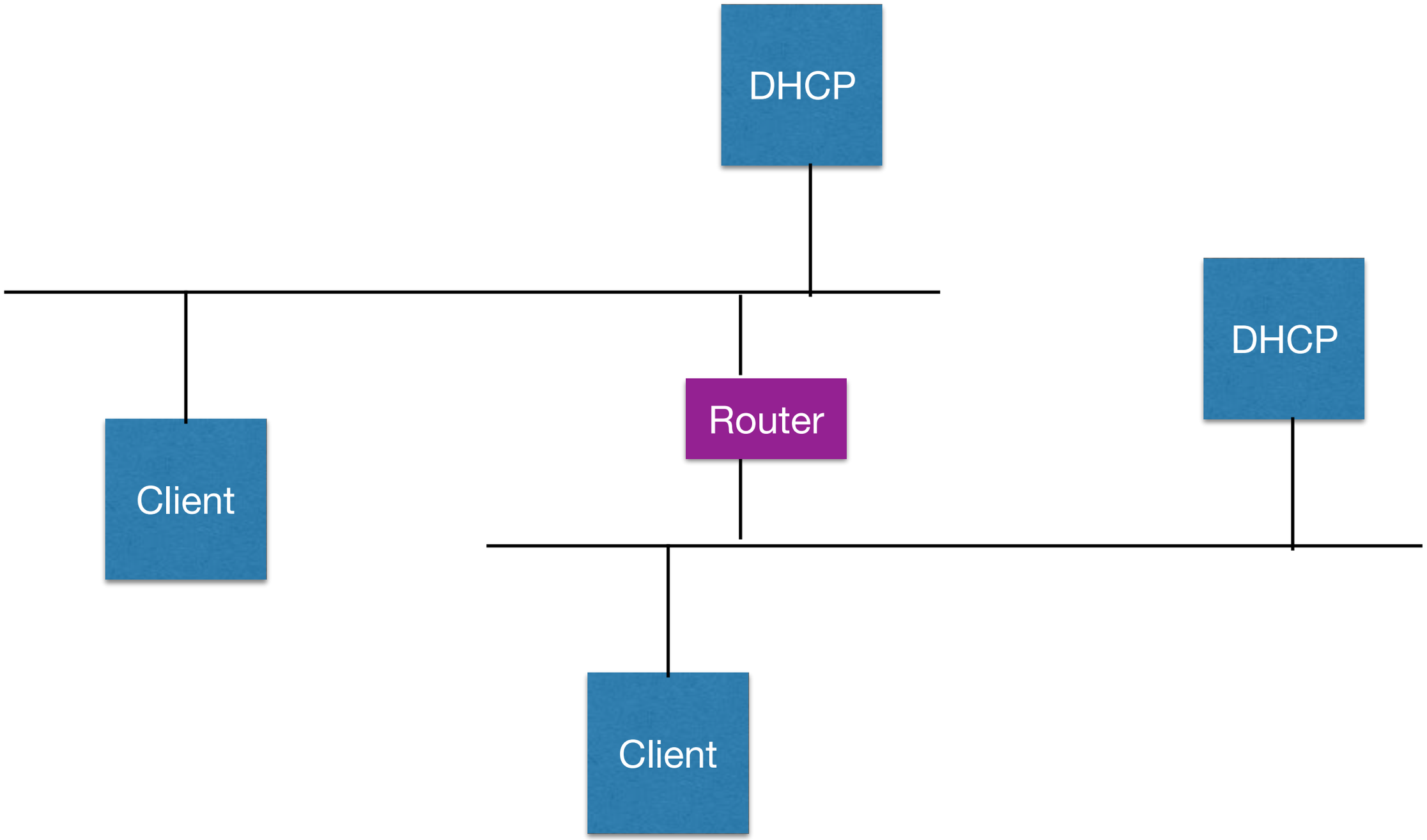
- DHCP failure is a serious problem
- DHCP is a nightmare to debug
- DHCP relays in complex networks are also very complex (look them up! take a headache tablet first!)

DHCP Relaying

All done as Broadcast, so local to network



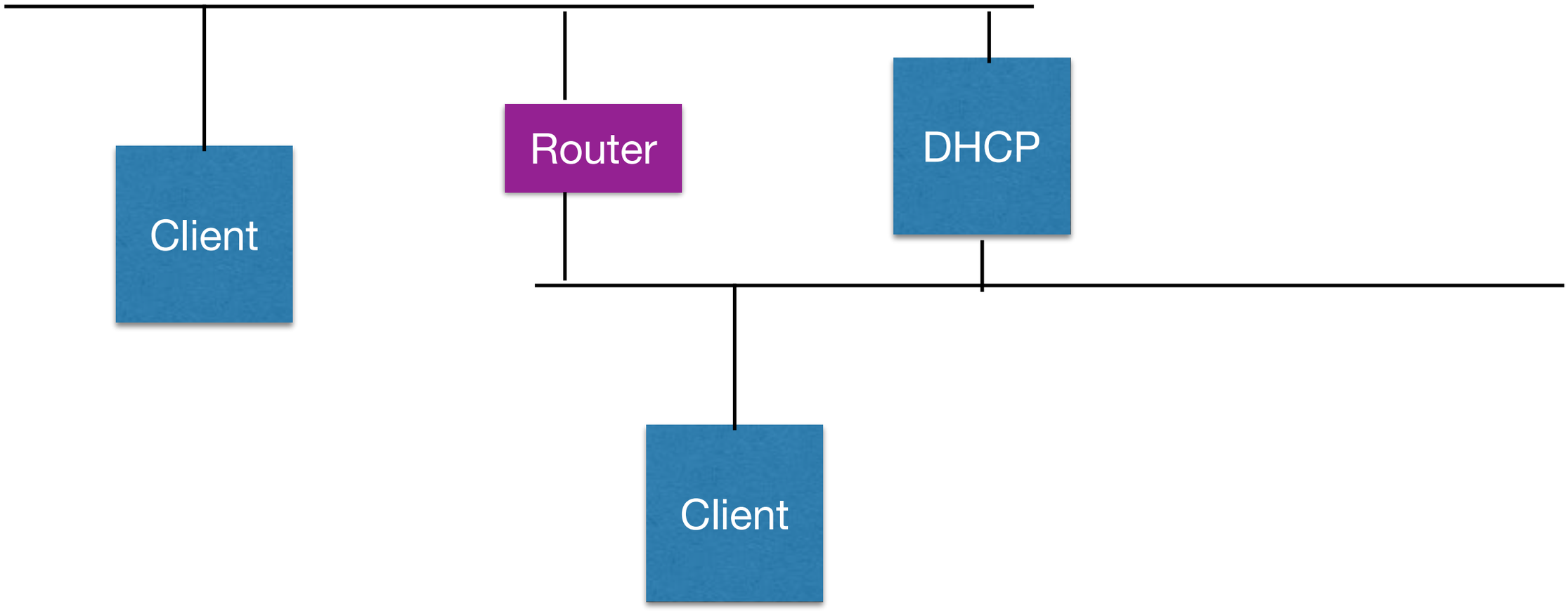
Multiple Networks?
Option1: Lots of DHCP



Problems

- We discussed how vital DHCP servers are
- Now you have the problem of making all of them reliable, rather than just one
-

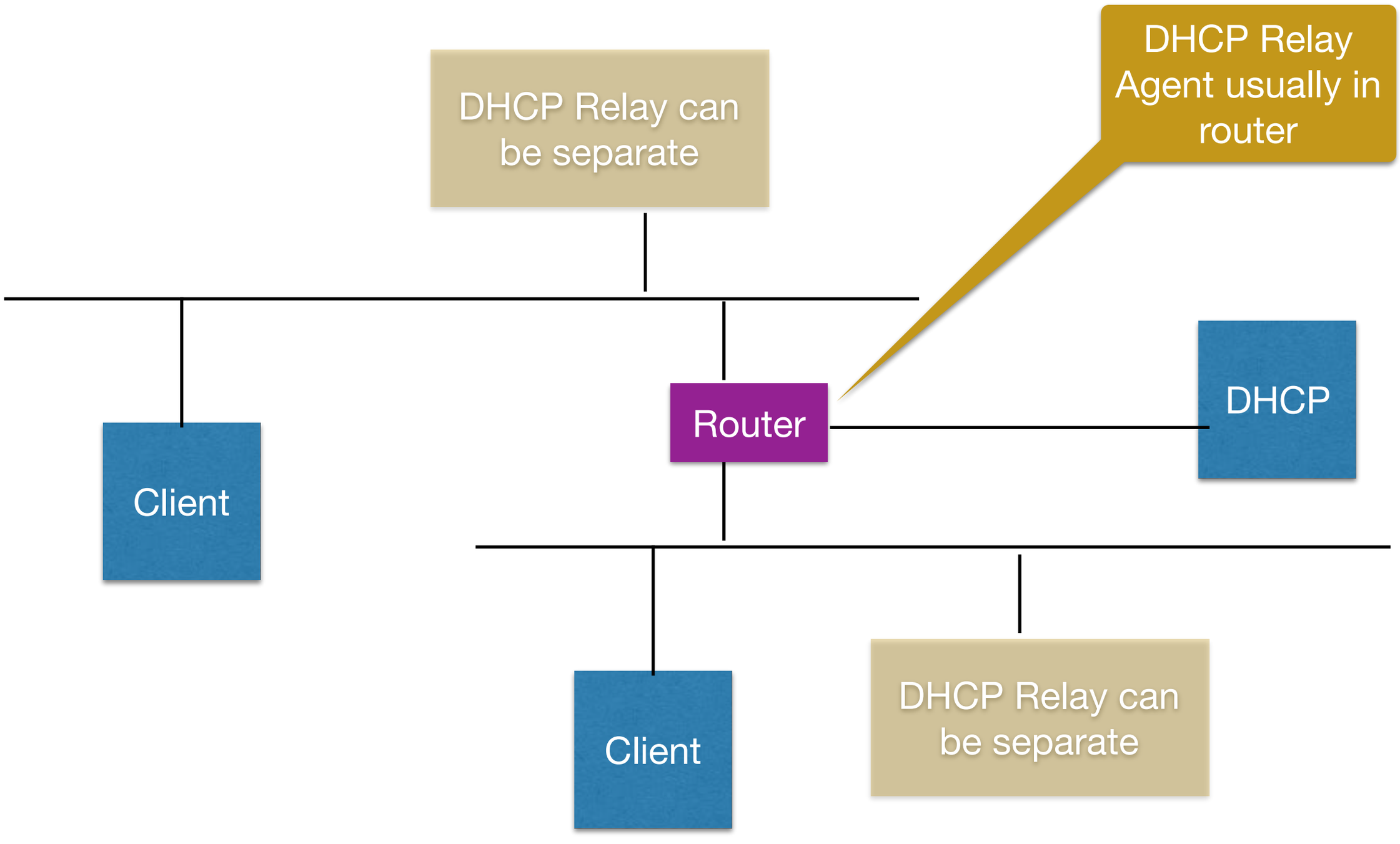
Option 2: Multihome



Problems

- Not scalable into very large networks
- Security nightmare (DHCP server is an implicit by-pass for any filtering that firewalls might do)

Option 3: Relaying



DHCP Relaying

- Relay agent on each network, usually embedded in router (but can be standalone)
  - Contains no state, can safely be replicated and have multiple running at the same time
- Operation is messy

DHCP Relaying

- Relay agent hears a broadcast packet
- Is configured to send all requests to a known DHCP server, after filling in its own address
- DHCP server receives unicast DHCP packets, containing a relay address.
- Server sends responses back to the relay
- The relay then broadcasts them on the local network

DHCP Relaying

- Relaying to a single server is the right solution in complex networks
  - DHCP server can be industrial-strength in a secure (power, aircon) data centre
- Makes complying with legal and regulatory requirements easier
- Nasty to debug

DHCP Security

- Obvious DoS: rapidly request leases, and then abandon them, so that the servers have no more addresses to hand out.
    - Some servers have no, or unrealistic, release periods
- Colleague reports a ***** is doing this regularly (either deliberately or through messed-up laptop) on Chiltern Line trains south of Banbury

DHCP Security

- And it's un-authenticated, so running a rogue DHCP server on a network which provides addresses of hacked DNS and router is powerful.
- And hard to stop: more in Network Security next semester.

IPv6 address allocation

- DHCPv6: broadcast your MAC address, get an IPv6 configuration, as with DHCP (v4)
  - All the usual problems with DHCP
    - Much debate about logging requirements, however
- Preferred methods are static for servers and SLAAC, Stateless Address Auto Configuration, for everything else
  - Limited experience, so “preferred” may not be “right”

IPv6 Router Advertisements

- In general, IPv4 routers (where a machine should send packets which are not destined for the local network) are configured either statically or by DHCP.
- IPv6 routers send router advertisements, which announce their existence and provide basic information about connectivity, both as periodic broadcasts and in response to solicitation messages
  - ICMP v6 messages 133 (solicitation) and 134 (advertisement)
  - At last support for DNS server information is starting to be discussed

SLAAC for IPv6

- Routers broadcast router advertisement packets
- From these, a device can deduce the prefix, the /64 that identifies the network
- They then put their 48 bit MAC address, plus some other stuff, into the address and just use it, without any more formalities
    - Full details, including dealing with clashes, in RFC4862
- But…

Problem #1 with SLAAC: DNS

- The device now has an address, but no details on things like DNS servers
- In dual-stack world, it can just use the IPv4 devices discovered with DHCP (dirty, but works)
- In single-stack world, intention is that there be standard multicast addresses for standard services, or that it's provided via NDP, or via DHCP "O" mode (next slide), or something.
    - Sadly none of these are universally implemented: another proof IPv6 deployment is still not really ready, making dual-stack (at least via RFC1918) almost essential

IPv6 DHCP for configuration

- Router Advertisement contains options field
- Amongst other options are “M” and “O”.
- These control how addresses are allocated and managed

IPv6 “Managed”

- If the router advertisement has the “M” flag set, the network is “Managed”.
- This means that clients must not use IPv6 SLAAC, and instead must use DHCPv6.
  - DHCPv6 will include DNS information

IPv6 “Other”

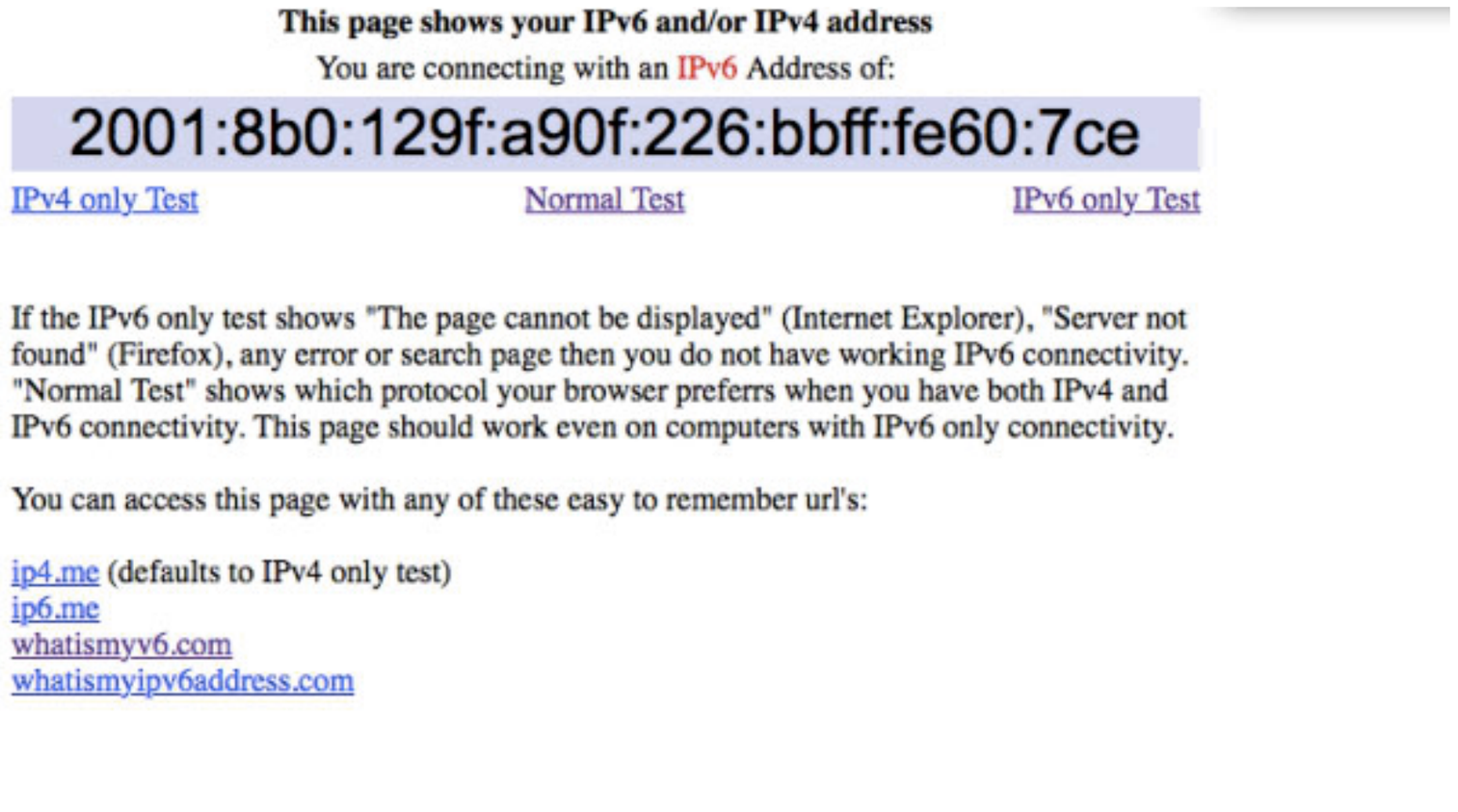
- A nasty hack.
- If the “O” or “Other Information” flag is set, the IPv6 node gets its own IPv6 address (by SLAAC, or statically allocated, or some other mechanism) but still goes to the DHCPv6 server for other configuration data.
- DHCPv6 server doesn’t need to manage leases, so is essentially static.
- Best solution until DNS multicast is reliably available.
  - But apparently (hot off the press!) now broken in Android, just as it started working in iOS.

Problem #2 with SLAAC

- In IPv4, your MAC address (which is unique to your machine) never leaves the local network
- In IPv6 with SLAAC, your MAC address is embedded in your IP number.
- This allows an observer to track you from network to network just by your temporary IP number.
- Arguably a privacy risk

IPv6 Privacy

```
en0: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST>
mtu 1500
  options=27<RXCSUM,TXCSUM,VLAN_MTU,TSO4>
  ether 00:26:bb:60:07:ce
  inet6 fe80::226:bbff:fe60:7ce%en0 prefixlen 64 scopeid 0x4
  inet 10.92.213.177 netmask 0xffffff00 broadcast 10.92.213.255
  inet6 2001:8b0:129f:a90f:0226:bbff:fe60:7ce prefixlen 64 autoconf
```



This page shows your IPv6 and/or IPv4 address

You are connecting with an IPv6 Address of:

2001:8b0:129f:a90f:226:bbff:fe60:7ce

IPv4 only Test Normal Test IPv6 only Test

If the IPv6 only test shows "The page cannot be displayed" (Internet Explorer), "Server not found" (Firefox), any error or search page then you do not have working IPv6 connectivity. "Normal Test" shows which protocol your browser preferrs when you have both IPv4 and IPv6 connectivity. This page should work even on computers with IPv6 only connectivity.

You can access this page with any of these easy to remember url's:

ip4.me (defaults to IPv4 only test)
ip6.me
whatismyv6.com
whatismyipv6address.com

IPv6 Privacy

- In fact, this worry is probably over-stated
  - Implies an attacker who can observe packets on all the networks you visit, but cannot observe any authentication exchange which identify you.
  - If they know which networks to look at, they already know where you are
  - If they can observe packets on those networks, they can see your machine by MAC address
  - If they have access to authentication logs or similar, they can track you by network mumber (first 64 bits) plus your authentication information.

RFC4941 Privacy

- Like SLAAC, but instead a random IPv6 address is chosen each time one is required (with mechanism for dealing with very unlikely clashes)
  - iPhones, Android and the like do this by default (which can be really annoying), getting a new address on every wake up from sleep
  - Linux, Windows, OSX have it as a choice (typically “on” for laptops, “off” for fixed machines) and acquire a new address whenever they change network.

Privacy Addresses

```
options=27<RXCSUM,TXCSUM,VLAN_MTU,TSO4>
ether 00:26:bb:60:07:ce
inet6 fe80::226:bbff:fe60:7ce%en0 prefixlen 64 scopeid 0x4
inet 10.92.213.177 netmask 0xffffff00 broadcast 10.92.213.255
    inet6 2001:8b0:129f:a90f:3912:c2fc:cfd1:6434 prefixlen 64
autoconf temporary
```

Note same prefix, different suffix

RFC7217

- Designed to produce stable addresses on each network to which you connect.
- Hashes the network's prefix, the interface identity (name, MAC, etc), the optional Network ID (SSID, etc), a counter in case there is a clash and a secret key initialised at OS installation.
- RID = F(Prefix, Net_Iface, Network_ID, DAD_Counter, secret_key)
- Preferable to RFC4941, because addresses remain the same on the same network (good for logging)ˆ

On latest OSX bits

- Probably some other new OSes as well.
- Means that

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
   ether 78:4f:43:90:d9:f8
   inet6 fe80::145a:76e7:d93a:7f92%en0 prefixlen 64 secured scopeid 0x5
```

Need to check iOS

- My iOS11 iPhone had 2001:8b0:129f:a90f:21f5:affa:eef:cbb1 for 24 hours, which is longer than it used to manage (a year ago, the address changed each time phone slept/woke). Similar longevity for addresses on wife's iPad (still iOS 10).
- But only 24 hours, to "secret" probably re-initialised on reboot, or something.