Overview

- Last Lecture
  - Network Hardware and Protocols
- This Lecture
  - IPv6 Bootcamp
  - Reference:
- Next Lecture
  - Scripting Techniques

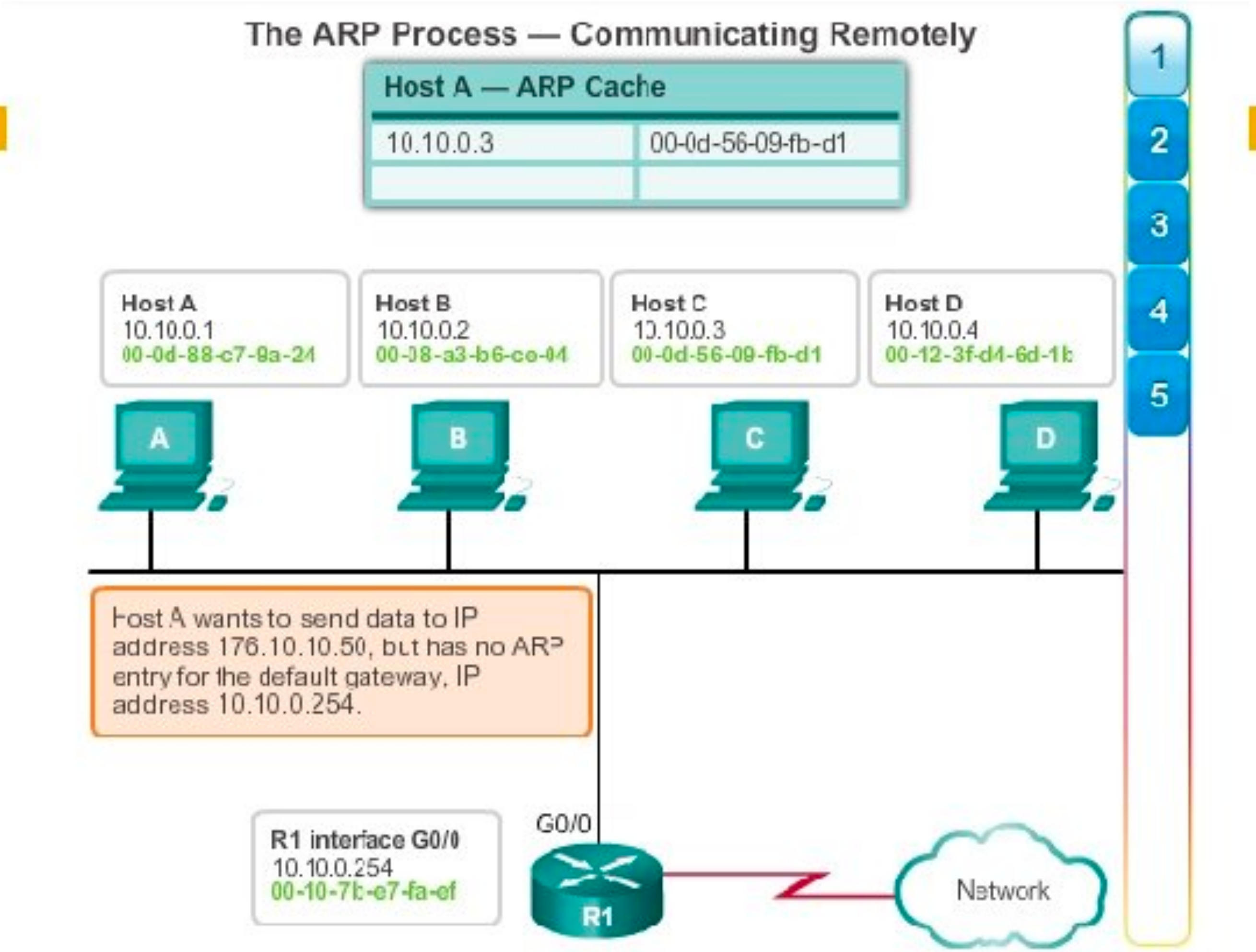
Outline

- Overview of IPv4
- Common IPv6 addresses
- Basic mechanisms of IPv6
- AutoConfiguration (SLAAC and DHCPv6)
- Neighbour Discovery (similar to ARP)
- Co-existing/Tunnelling (6in4, 6to4, Toredo)
- Security issues

How IPv4 works?

- Address: 32 bits
- DHCP is used for dynamic address allocation
- ARP is used to resolve MAC address given IP address
- Use a human simulation to show how it works

ARP



The ARP Process — Communicating Remotely

Host A — ARP Cache	
10.10.0.3	00-0d-56-09-fb-d1

Host A wants to send data to IP address 176.10.10.50, but has no ARP entry for the default gateway, IP address 10.10.0.254.

IPv6 Brief Recap

- Much enlarged address space (128 bits)
  - More addresses and more network IDs (no NAT needed)
  - Smaller routing tables via address aggregation
  - Everyone in the world could be online (directly)
- Autoconfiguration using DHCPv6 or SLAAC
  - Easier to have more devices (in-car networks, etc.)
- Streamlined packet header (easier routing)
- Advanced features
  - QoS, Mobility, (optional) IPSec

Address Notation

- 8 groups of 16 bits in hex, can be compressed
  - fe80:0000:0000:0000:0226:5eff:fe00:8242
  - fe80:**0**:**0**:**0**:**226**:5eff:fe00:8242
  - fe80**::**226:5eff:fe00:8242
  - fe80::226:5eff:fe00:8242%6 (or %eth0) zone index
- Some addresses have embedded IPv4
  - ::ffff:192.168.0.2 ≡ ::ffff:c0a8:2
- What about addresses with ports? (colon use)
  - [fe80::226:5eff:fe00:8242]:8081
  - http://[fe80::226:5eff:fe00:8242]:8081/

Prefix Notation

- Functionally equivalent to network mask or IPv4 Classless Inter-Domain Routing (CIDR) prefix
  - but much easier to work with because IPv6 uses hex notation, which is easier to convert to binary
- Trailing /n means that the network ID ends after the nth bit
  - e.g. fe80::/10 or 2002::/3
  - Exercise: is 3001::1 in 2002::/3 ?
  - Exercise: is fd6b:4104:35ce:0:a00:fed9 in fc00::/7 ?

Host Address Formulation

- 128 bits: 64-bit prefix & 64-bit interface identifier
- Interface IDs can be formed by hosts themselves
  - e.g. may base on their EUI-64 interface identifier.
  - For Ethernet, this is based on MAC address
    - 00-26-5E-00-82-42 → 0**2**26:5e**ff**:**fe**00:8242
    - insert ff:fe and swap universal/local bit (a MAC like this that is universal will be manufacturer-assigned)
- This interface identifier is appended to the prefix of the network advertised by the router.
- "Privacy extensions": random temporary interface IDs can be generated for outgoing traffic

Common Unicast Addresses

- See RFC4291
- ::1 and :: Loopback and Unspecified
- fe80::/10 Link-local
  - append *%zone index*: %eth0 (Linux) or %6 (MS)
- fc00::/7 Unique-local RFC4193
  - Like deprecated site-local, but with fewer problems, e.g. since RFC4193 addresses require good pseudo-random parts, organisations can most likely aggregate without conflict in their unique-local addresses.

Common Unicast Addresses (cont.)

- 2000::/3 Global unicast RFC3513 RFC4291
- 2001:0000::/32 Teredo RFC4380
- 2002::/16 6to4 tunnelling RFC3056
- 2001:db8::/32 Documentation only RFC3849
- Others …
  - These allocations are made by Internet Assigned Numbers Authority (IANA)
  - http://www.iana.org/numbers/

Common Multicast Addresses

- ff00::/8 is multicast, but we also encode scope:
  - ff + 4 bits of flags + 4 bits of scope + 112 bits of group ID
- There is no broadcast: special case of multicast
  - ff02::1   Link local ‘all-nodes’
  - ff02::2   Link local ‘all-routers’
- These are generally never used by *applications*
  - Scopes: e.g. 1 = node-local, 2 = link-local, 5 = site-local, 8 = organisation-local, E = global scope.
  - ff05::1   ‘Site’ local ‘all-nodes’

Lots of Addresses

- Unicast addresses have a particular scope
  - Node-local, Link-local, Global (Universal)
- Hosts have multiple addresses
  - must have link-local
  - plus any number of advertised prefixes (e.g. unique-local + global)
  - plus any static addresses
  - addresses have a lifetime (preferred, deprecated)
  - addresses can be temporary (privacy addresses)
  - plus multicast addresses (solicited node and all-nodes + ...)

Default Address Selection

- Choice of source address
  - Varying in version, scope, state
- Choice of destination address
  - Varying in version, scope, state
  - Could get multiple results during name lookup
- How to choose appropriate pairing?
  - Source: global v4 or link-local v6
  - Destination: global v4 or global v6
  - Not simple, so RFC3484 defines algorithm

What to get from IPv6 ISP?

- Smallest practical subnet size is /64
- RFC3177 contains recommendations
- Home network subscribers /48
  - In reality, some ISPs will give a /56, but a /64 is too small. You might give a /64 to a mobile network when you know no subnets are needed.
  - Remember that a /48 allows for $2^{64-48}=2^{16}$ subnets.
- Small and large enterprises /48
- Very large /47 or many /48s

How interfaces get configured

- Link-local address formulated and tested
- StateLess Address AutoConfiguration (SLAAC)
  - Nodes send out a Router Solicitation
  - Routers send out Router Advertisements informing nodes on the link of prefixes and lifetimes.
- DHCPv6 (either stateful or stateless)
  - **Stateful**: gives out static addresses that you might give to a server, for example (think DHCP for IPv4)
  - **Stateless**: augments SLAAC with extra info
- Manual/Static
  - Useful for routers and servers

Router Advertisement

- Multicast ICMPv6 message to ff02::1
  - or to the solicited node multicast address for the addr.
- Contents include at least these bits:
  - **M**anaged address config flag
    - If 0: use stateless autoconfiguration
    - If 1: use stateful configuration (DHCPv6)
  - **O**ther stateful config flag
    - If 1: use DHCPv6 for other information
- Router lifetime (>0 means default router)
- Contains a list of prefixes advertised on this link

Neighbour Discovery

- Replaces ARP
  - Implemented with ICMPv6
- Includes MTU and reachability information
  - Caching Path MTU
- Neighbour Solicitation & neighbour advertisement
  - Sent to the solicited node's multicast address. This is formulated based on the queried address to reduce traffic to all nodes.
- SEcure Neighbour Discovery (SEND)
  - See also: IPSec

Duplicate Address Detection

- Since some addresses are formed by hosts themselves, there could be duplicate addresses.
- Duplicate Address Detection (DAD)
  - uses Neighbour Discovery to query if generated address is used (if it is, abort this address)
- Generate link-local address, then “DAD” it
- Generate global addresses by adding interface ID to advertised prefixes, then “DAD” it.

Transition mechanisms–statuses

- **6in4** (Proto-41): statically configured tunnel
- E.g. as used by tunnel brokers
- **6to4**–more flexible; support relay routers
- **Teredo**–even more flexible; can tunnel through NAT over UDP
- ISATAP–Intra-Site Automatic Tunnel Addr. Prot.
- NAT64 & DNS64–Allow only IPv6 → IPv4
- Ignore: NAT-PT, 6*over*4 (note, not "6to4"), IPv4-*compatible* IPv6 addresses (not "-mapped"), 6Bone

Security Threats

- IPv6 might be on by default, and preferred...
  - You might not even realise or know how to manage it
- Autoconfiguration and rogue advertisements
- Routing header 0 (“loose source routing”)
- Firewalls for IPv6 generally neglected
  - if thought of at all yet ...
- Tunnelling mechanisms hide traffic
- Claims of “IPv6 support”

Summary

- Remember formats of various IPv6 addresses
  - link local, global unicast, multicast, loopback, unspecified, etc.
- How to detect duplicate link local address in SLAAC? use DAD protocol
- How to create an EUI-64 identifier based on the MAC address of a network interface card?

References

- IPv6 Essentials, Second Edition, by Silvia Hagan. **2006**. Published by O'Reilly, also available from Apple's AppStore
- http://rfc-editor.org/
  - Great for checking if particular RFCs have been deprecated (useful when checking book content!)
- http://www.iana.org/
- Wikipedia
  - Useful for checking up-to-date status and references

Experimenting IPv6

- On MacOS/Linux
  - $ ifconfig
  - $ netstat -rn
  - $ ping6, etc
- http://test-ipv6.com/
- host -a www.cs.otago.ac.nz ipv6.test-ipv6.com
  - Note: use the IP address of ipv6.test-ipv6.com
- telnet ipv4.test-ipv6.com 79
- telnet ipv6.test-ipv6.com 79
- telnet ds.test-ipv6.com 79

IPv6 addresses assignment

```
2001:0db8:0123:4567:89ab:cdef:1234:5678
|||| |||| |||| |||| |||| |||| |||| ||||
|||| |||| |||| |||| |||| |||| |||| |||128   /128 Single end-points and loopback
|||| |||| |||| |||| |||| |||| |||| ||124
|||| |||| |||| |||| |||| |||| |||| |120
|||| |||| |||| |||| |||| |||| |||| 116
|||| |||| |||| |||| |||| |||| |||112
|||| |||| |||| |||| |||| |||| ||108
|||| |||| |||| |||| |||| |||| |104
|||| |||| |||| |||| |||| |||| 100
|||| |||| |||| |||| |||| |||96
|||| |||| |||| |||| |||| ||92
|||| |||| |||| |||| |||| |88
|||| |||| |||| |||| |||| 84
|||| |||| |||| |||| |||80
|||| |||| |||| |||| ||76
|||| |||| |||| |||| |72
|||| |||| |||| |||| 68
|||| |||| |||| |||64   /64 Single end-user LAN subnet (required prefix size for SLAAC)
|||| |||| |||| ||60   /60 Some (very limited) 6rd deployments
|||| |||| |||| |56   /56 recommended Minimal end-site assignment
|||| |||| |||| 52   /
|||| |||| |||48    /48 recommended Typical assignment for home sites
|||| |||| ||44    /
|||| |||| |40    /
|||| |||| 36    /36 possible future local Internet registry (LIR) extra-small allocation
|||| |||32     /32 LIR minimum allocation
|||| ||28     /28 LIR medium allocation
|||| |24     /24 LIR large allocation
|||| 20     /20 LIR extra large allocation
|||16      /
```