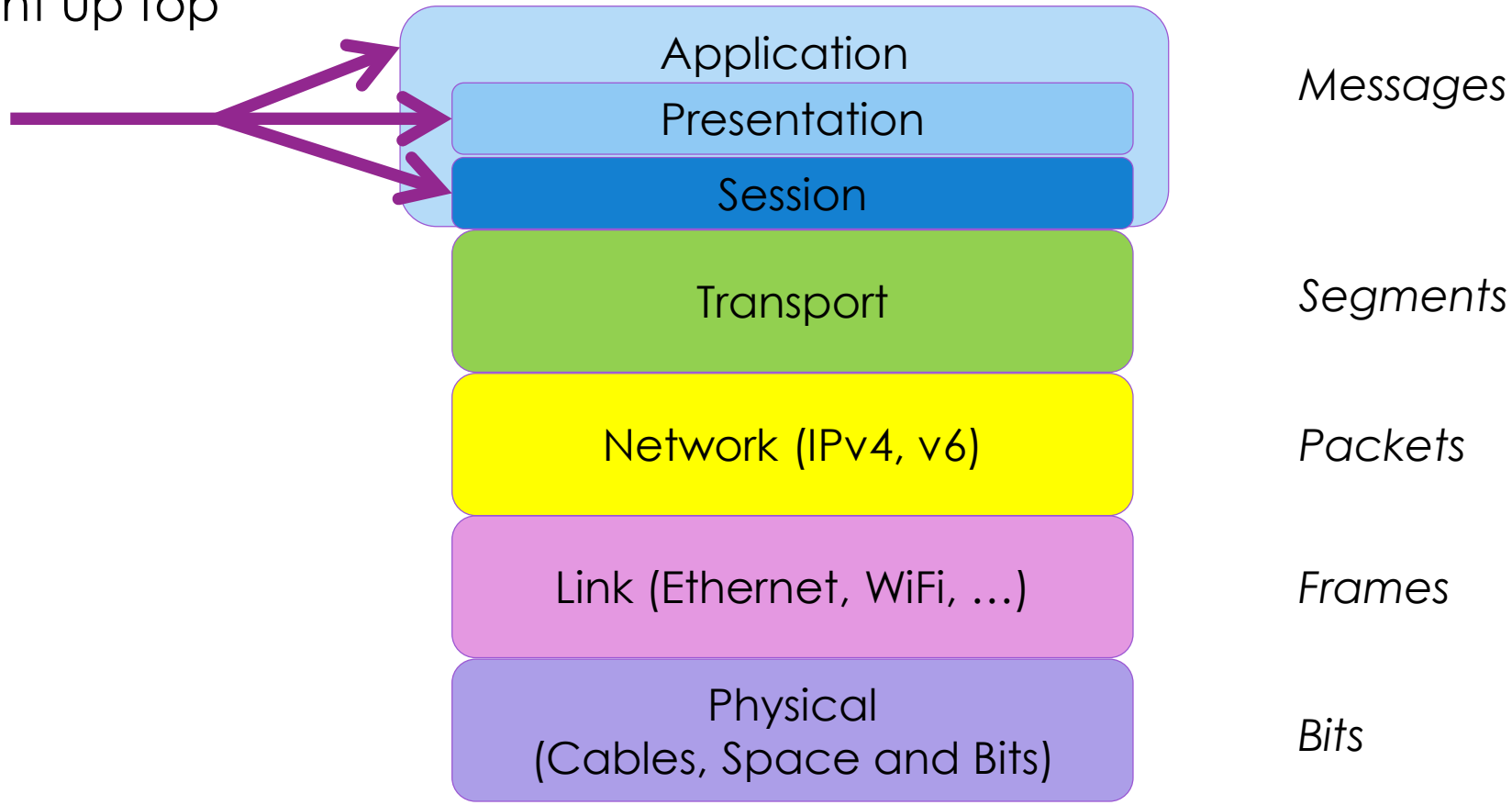
COMP3310/6331 – #13-14

Application Layer, DHCP, DNS

Dr Markus Buchhorn: *markus.buchhorn@anu.edu.au*

Where are we?

- Right up top



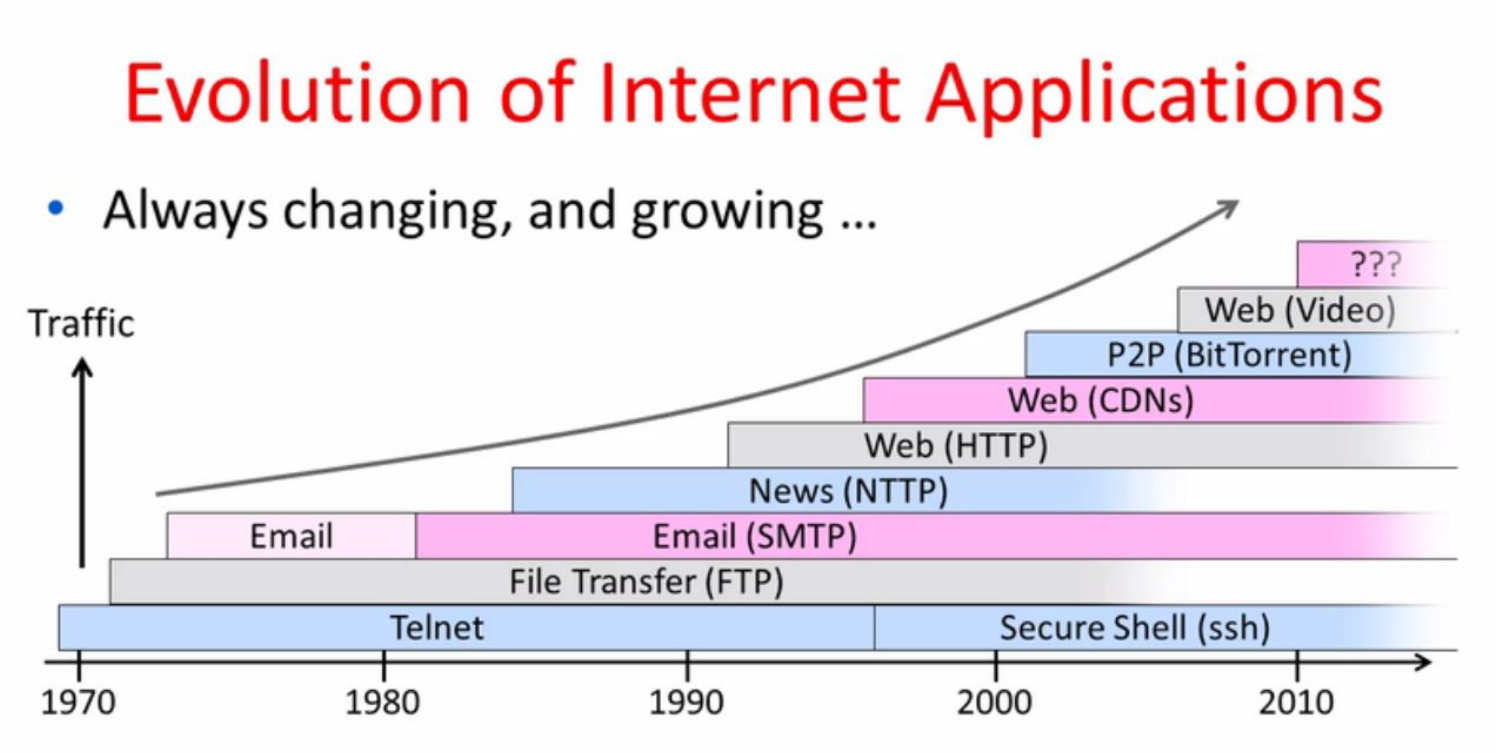
Some application context

- Good summaries to read
  - Akamai 'State of the Internet' (quarterly)
  - Cisco Visual Networking Index (annual)
  - Mary Meeker Internet Trends (annual)

- 3.6B users
- 2.8B smartphones – *using 10,000PB/month*
- Heading towards 27B devices
- 3h/day mobiles (+), 2h/day desktop (=)
- ~80% of traffic is video
- Best national average broadband ~28Mb/s (KR) – global is 7Mb/s
- 5G can bring 30Gb/s, as long as nobody is sharing…

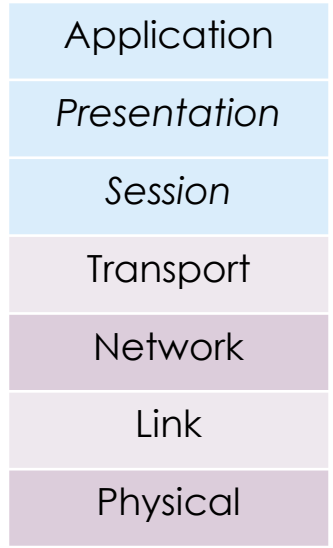
New applications all the time

- And each brings more traffic, to more users/devices
- D. Wetherall (2012):



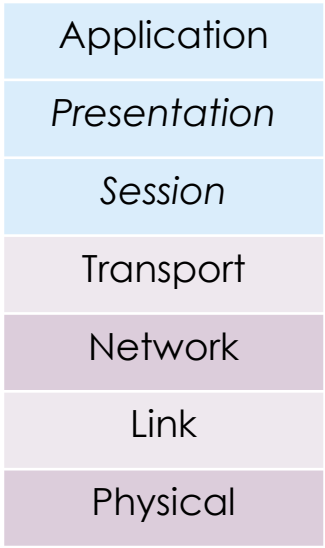
Application space

- Build sessions (a series of interactions)
  - E.g. a web page with multiple resources, multiple sources
  - A videoconference between particular endpoints
- Build on top of TCP (reliable byte-stream) or UDP (unreliable messages)
  - And add whatever functionality they require – e.g. reliable UDP sessions?
- Applications have one or more application-layer protocols
  - E.g. http/https for webpages



Applications space

- Also handle Presentation
- Manage:
  - Content-types (images, video, audio, text, …)
  - Content-encodings (compression, uuencode, mime, …)
  - Content-packaging (file formats, message types, …)
  - Content-selection (receiver capability negotiation)
- Deal with command and control between two endpoints
  - "I want X"
  - "You are about to receive Y"
- Often see plain-English application protocols
  - Efficiency is for geeks, debugging is much easier
  - Overheads are low (command headers vs data and lower-layers)



Helper protocols (are applications too!)

- *ARP – translate between layer 3 (IP) and layer 2 (MAC)*
- *ICMP, IGMP – network control and feedback*

- So (1) how do I get my IP address?
  - I need a routable/forwardable address to participate

- And (2) how do I get my name?
  - **150.203.56.47** or **3018:ae8::ae00:98:8ac2** are not memorable, nor guessable
  - *www.anu.edu.au* is

Dynamic Host Configuration Protocol…

- Problem: node wakes up, knows nothing.
- "What's my IP, mask, router/gateway?"
  - Needed to join the internet!
  - At least I have my MAC address.

- Solution 1: Manual configuration. Depends on local needs. Doesn't scale.
- Solution 2: Automatic configuration, service from IT

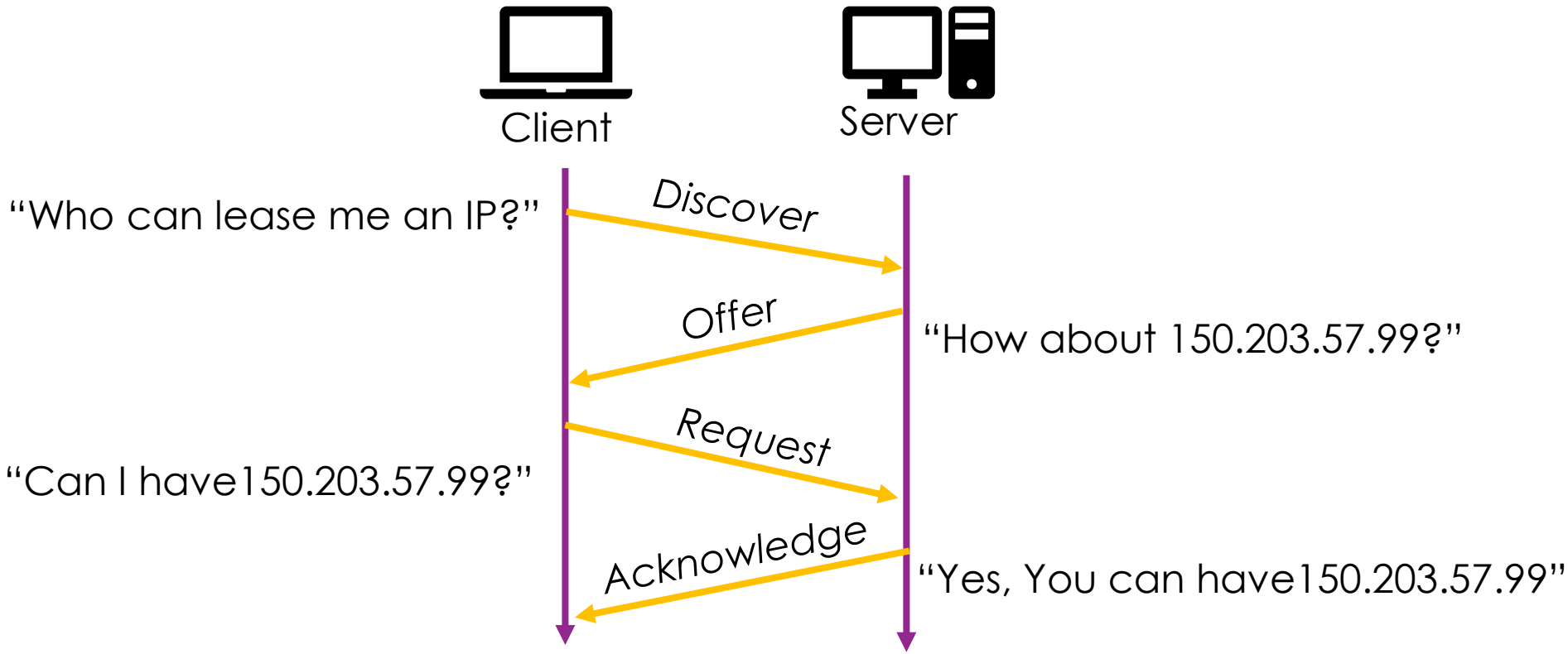
- DHCP (1993 – ex BOOTP) – gives/leases you your IP address

DHCP application

- Client/server application,
- UDP, client port:68, server port:67 – just ARQ if no reply
- Bootstrap:
  - How to send IP packets before IP is configured?
  - How to send them to DHCP server when you don't know where it is?
  - Broadcast to the rescue! IP:255.255.255.255 => Ethernet ff:ff:ff:ff:ff:ff
  - Source = 0.0.0.0
  - DHCP server should be on the same LAN (broadcast domain)
    - Or somebody needs to do some more work…

DHCP messages

- Really simple: DORA…



DHCP cont.

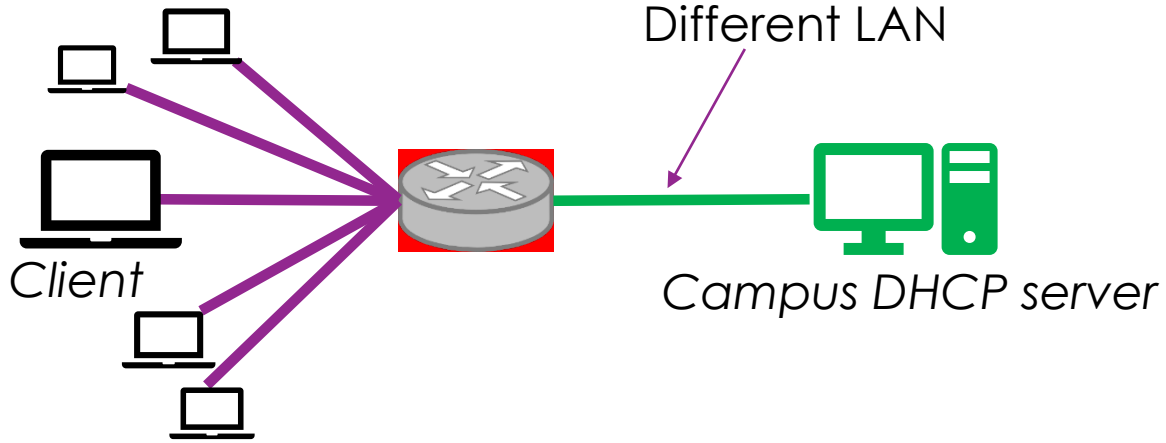
- Lease renewal:
  - Just REQUEST (can I please have) and ACK (yes you can)
    - unicast
  - If server disagrees:
    - Rejected (authoritative)
    - Ignored (passive) and timeout

- With new IP address, clients SHOULD *(gratuitous) ARP* to make sure it's ok…
  - Two DHCP servers; A manual/dynamic overlap;

- Actually a little more complex, due to BOOTP inheritance
  - Transition from BOOTP to DHCP with backwards compatibility
  - Packet format was kept, but purposes shuffled

DHCP does more

- DHCP relays



- Multiple DHCP servers (failover, performance)
- DHCP release – tell server to free up the address (optional) (*)
- 50+ features/records
  - Subnet mask, router, time server, dns server, log server, boot files, smtp, …
- Also allow for fixed ('static') MAC<->IP mapping

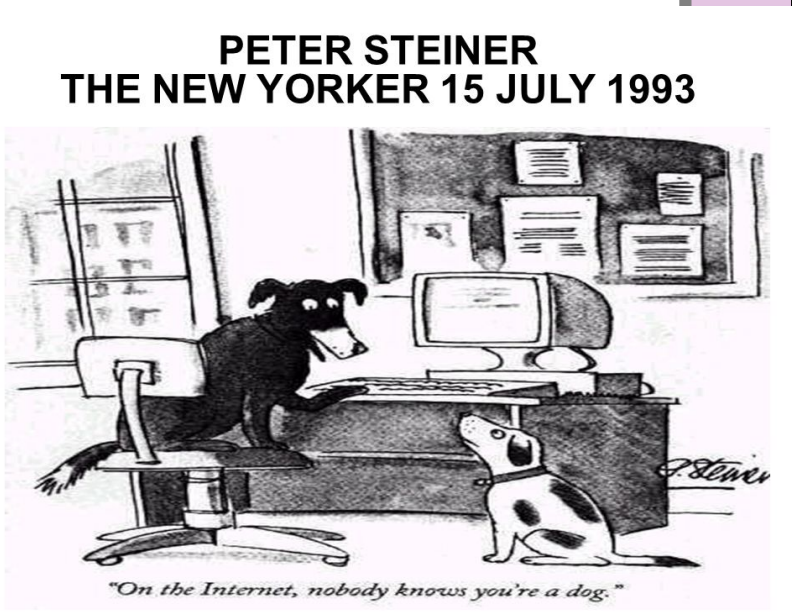
How does the DHCP server know?

- Manually configured, or
- Built off reasonable defaults
- Maintains database of who has what for when

- E.g. Home modem/router acting as DHCP server:
  - 192.168.x.y/24 subnet
  - DHCP server is the Default Route (to the Internet)
  - DHCP server is the DNS server

Domain Name System (DNS)

- Memorable, or guessable, names
  - www.anu.edu.au instead of 32-128 bits of addresses
  - A fixed name, rather than a variable address
- And a whole lot more!
  - Key service endpoints
  - Redirection, load balancing, dynamic allocation
  - Service metadata (priority)
  - Trust – somebody is in charge
    - Trust the device, if not the application, or the other user



Domain Name System (DNS)

- IP addresses and service endpoints change

- Why does an IP address change?
  - At home – ISP reallocation of your router
  - Organisational renumbering
    - Sold their block of IP addresses,
    - Relocating equipment, new server, …
  - Mobile devices

- Having multiple devices that failover/share a service as needed
  - Web servers, email servers, directory servers, file servers, …

Definitions

- Names (for humans)
  - not just devices/services, e.g. email address, social-media accounts, …
- Addresses (for protocols)
  - not just TCP or IP or MAC, e.g. URLs
- **Resolution** maps between them
  - Definitively/unambiguously
  - Mostly downwards, but lookups can also be 'reversed'

- *Note*
  - *a Name can have multiple Addresses*
  - *an Address can have multiple Names*

DNS Design

- Provide a Resolution Service
  - Mostly to convert names to IP addresses (www.anu.edu.au = 130.56.66.152)

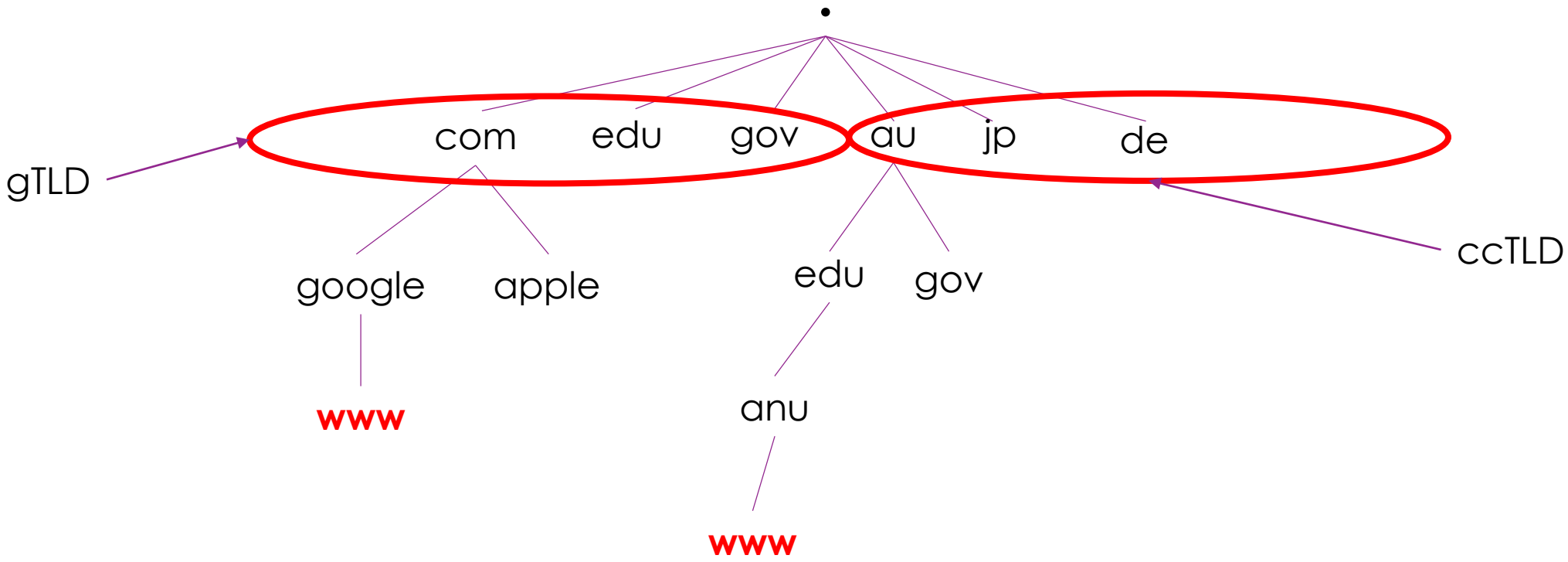
- Need to be
  - Easy to manage: many parties may be involved
  - Efficient: high data volumes, low-delays, low-load

- Build it:
  1. Distributed Directory *(no central database)*
  2. Hierarchical Namespace *(delegate to authorities)*
  3. Automated protocol/processes for running it *(set and forget(!))*

DNS Namespace

- Everything starts from '.' – the ROOT
- Add a 'TOP LEVEL DOMAIN' (TLD)
  - Which may be 'generic' (gTLD) = com, edu, org, net, mil, gov, …
  - Or a Country Code (ccTLD) = au, uk, us, it, fm, tv, to, …
- And keep building up from there towards your hostname
- A Fully Qualified Domain Name
- *Like www.anu.edu.au.*
- *Or www.google.com. (or goo.gl.)*

Typical DNS hierarchy view



How many TLDs?

- TLDs carry a **lot** of politics, and money, and culture, and …
- Defined by IANA, implemented by ICANN

- 6 originals, notionally for defined purposes (com = commercial, …)

- 7 new in 2000, *.museum, .aero, .coop, .name, .info, .pro, .biz*
  - Anger and confusion with .com and .biz!!

- 8 more from 2004-2012

How many TLDs?

- In 2008 new rules: No rules! Ok, some rules.
  - Financial model ($US185k),
  - Policies for each domain
  - Support for internationalisation (e.g. Chinese, Arabic, Cyrillic, …)
  - Sponsored TLDs (industry sectors, like .aero)
  - Geographic TLDs that aren't countries (.kiwi, .asia, .paris, …)

- In March 2018 – **1200 gTLDs**!
  - Lots of competition for the same names
  - Some very/too close
    - .hotels and .hoteis .unicorn and .unicom

This creates jobs (for lawyers and marketers) but little extra value

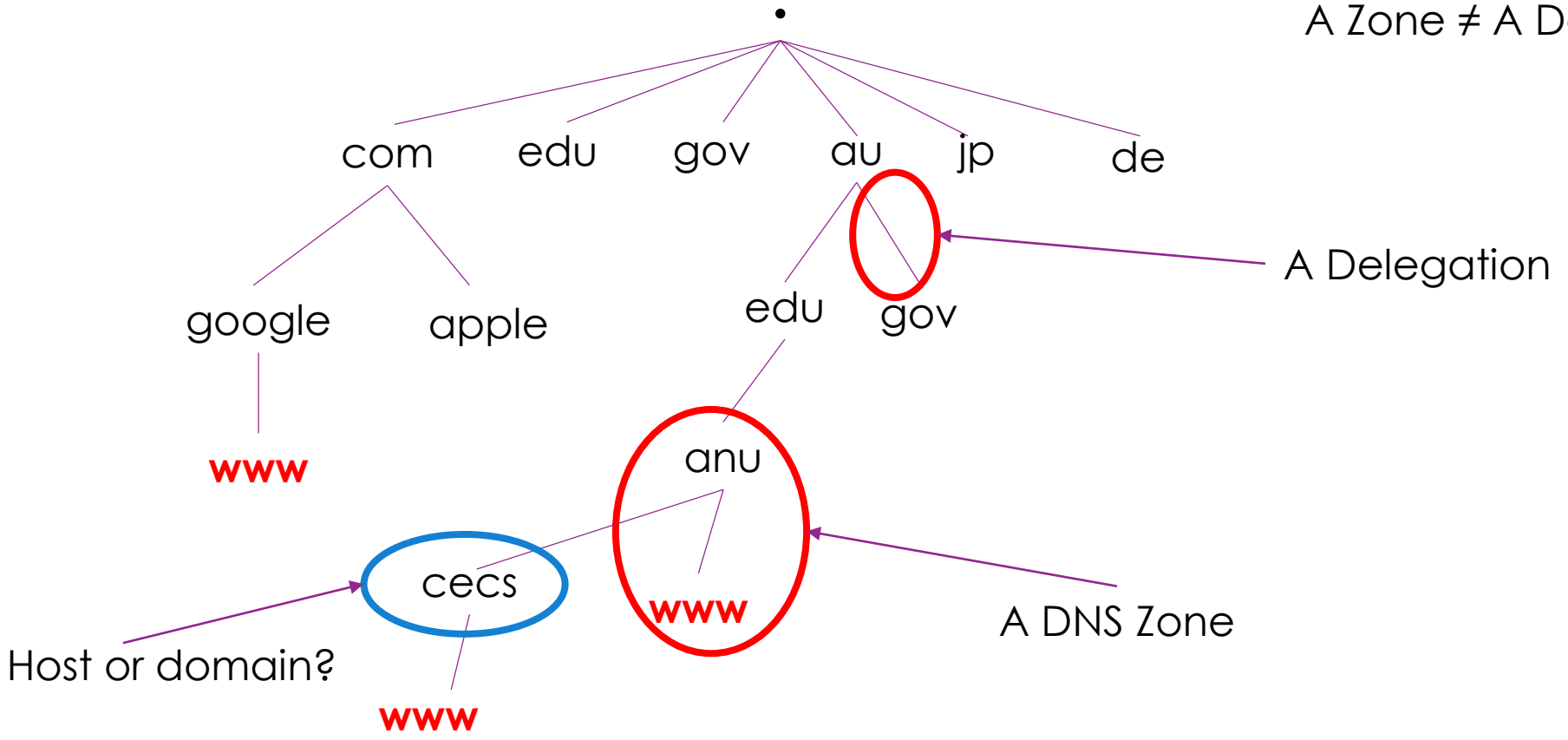
ccTLDs

- Based on ISO 3166 two-letter country codes
  - Yet more politics!
  - "Country" can be a disputed topic…
  - Countries come and go too…

- Own sub-domain rules within ccTLDs
  - .edu.au (like US, and added .asn.au and .id.au)
  - .ac.jp
  - .uniX.de

Back to tech!

A Domain ≠ A Zone
A Zone ≠ A Domain



Delegations = relationships = ownership

- Domains are what gets delegated - through legal entities – start from ICANN
  - AU Registrar (auda.org.au) administers second-level-domains in **.au**
  - Education Services Australia administers domains in **.edu.au**
  - ANU administers domains (and hosts) in **.anu.edu.au**
  - Colleges can have sub-domains, etc.

- Zones are shared pieces of the DNS database – through technology
  - Each zone identifies an authoritative nameserver
  - Each zone records delegations and their nameservers

What's in a zone?

- Information about
  - The zone, responsibilities
  - Further relationships (delegations)
  - And lots of addresses, services, etc.
  - And metadata about records (timeouts, etc.)
  - Through 'resource records'

RR Type	What it carries
SOA	Start of Authority – who's the boss
A	IPv4 address of a host
AAAA	IPv6 address of a host
CNAME	Canonical name, an alias
MX	eMail exchange for domain
NS	Nameserver of this or delegated domain

Zone example

- ANU examples:

1. `anu.edu.au.  35619  IN  SOA  ns1.anu.edu.au. hostmaster.anu.edu.au. 2019032016 3600 1800 1800000 36000`

2. `anu.edu.au.  150  IN  MX  10 mail.anu.edu.au.`

3. `www.anu.edu.au.  130  IN  CNAME  gaia-proxy.anu.edu.au.`

4. `gaia-proxy.anu.edu.au.  132  IN  A  130.56.66.152`

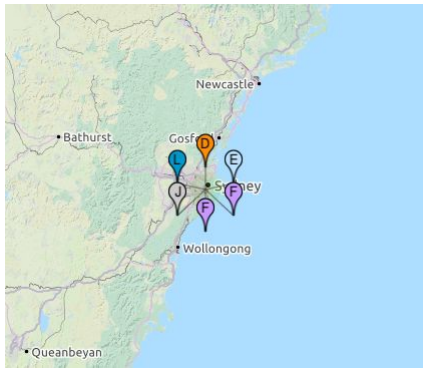
DNS resolution

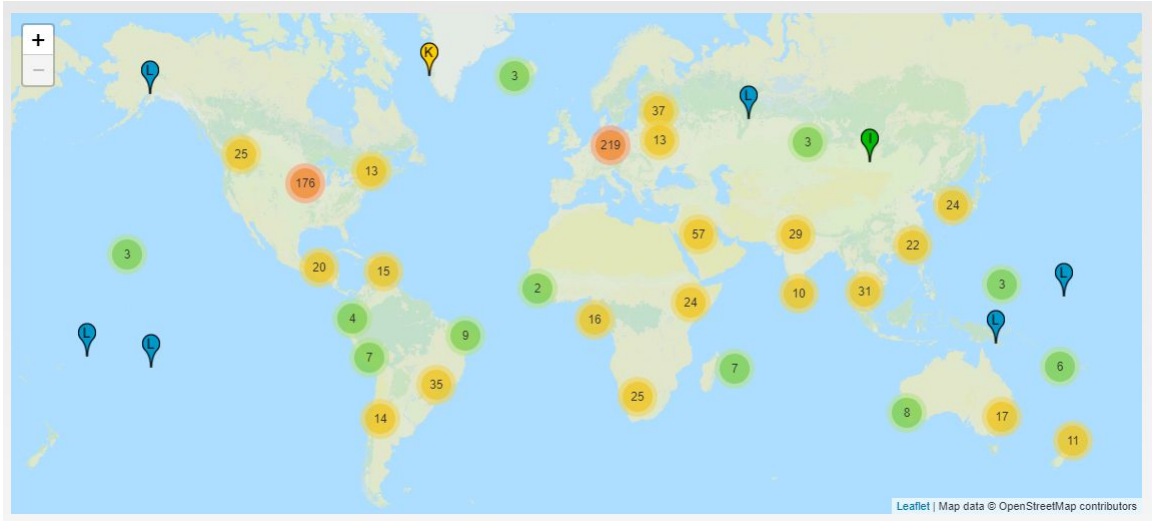
- Depends on the query…
- Let's start with "What is the IP address of host X?"

- Without anything to go by, go to the root!
  - It knows everything?
  - It knows who might know more…

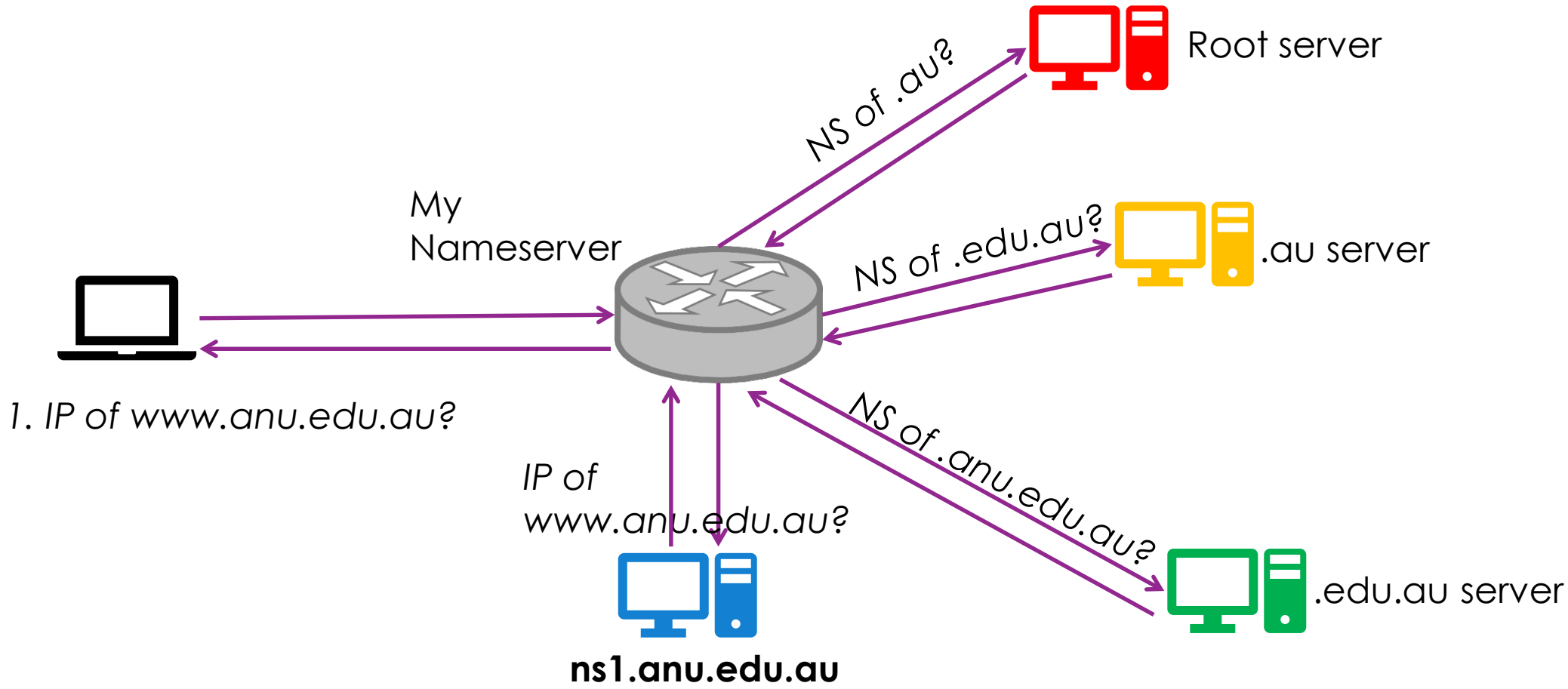
DNS root servers

- https://www.iana.org/domains/root/servers
- 13 important (and tempting) boxes on the Internet (a..m.root-servers.org)
  - Actually, several hundred replicas
- Every nameserver knows about them
  - Default route is the root
- Reachable via 'anycast'
  - (advertise the same IP address)





Resolving down the tree



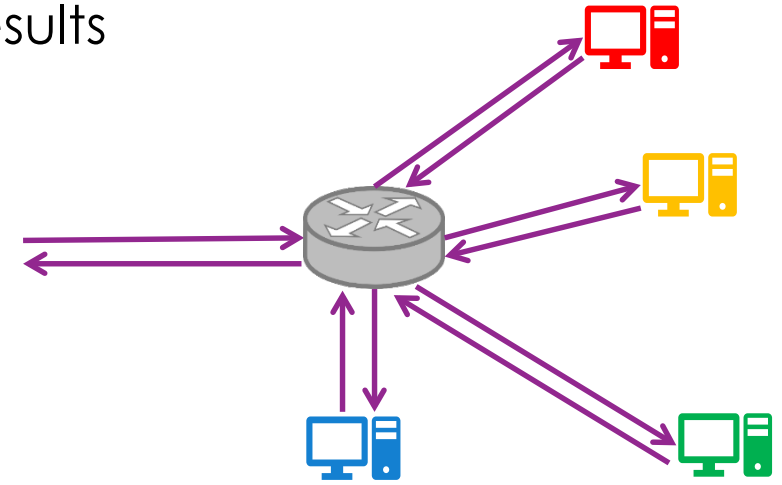
Recursive and Iterative

- Iterative: “Hey NS, who is next in the tree?”, then repeat
  - High performance, low delay
  - Provides a service

- Recursive: “Hey NS, you work it out, just give me the answer!”
  - Low performance, low impact
  - Good for the end client

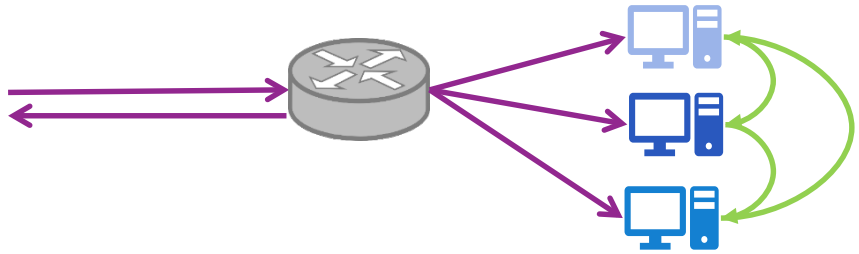
Caching

- Performance of this doesn't scale
  - A web page can have hundreds of resources from unique servers
  - Client needs to contact all of them.
  - Many lookups for a single session!
  - Need a shortcut – only need the last one/two?
- Nameservers can cache iterative-query results
  - **.au** won't change often
  - **.edu.au** won't change often
  - **.anu.edu.au** won't change often
- But they will – so need a Time-to-live (*)



Nameserver replication

- When one authoritative nameserver isn't enough…



- Register multiple nameservers
  - Spread the load, and the risk

- Client picks one
  - `anu.edu.au.  29112  IN  NS  ns1.anu.edu.au.`
  - `anu.edu.au.  29112  IN  NS  ns.adelaide.edu.au.`
  - `anu.edu.au.  29112  IN  NS  una.anu.edu.au.`

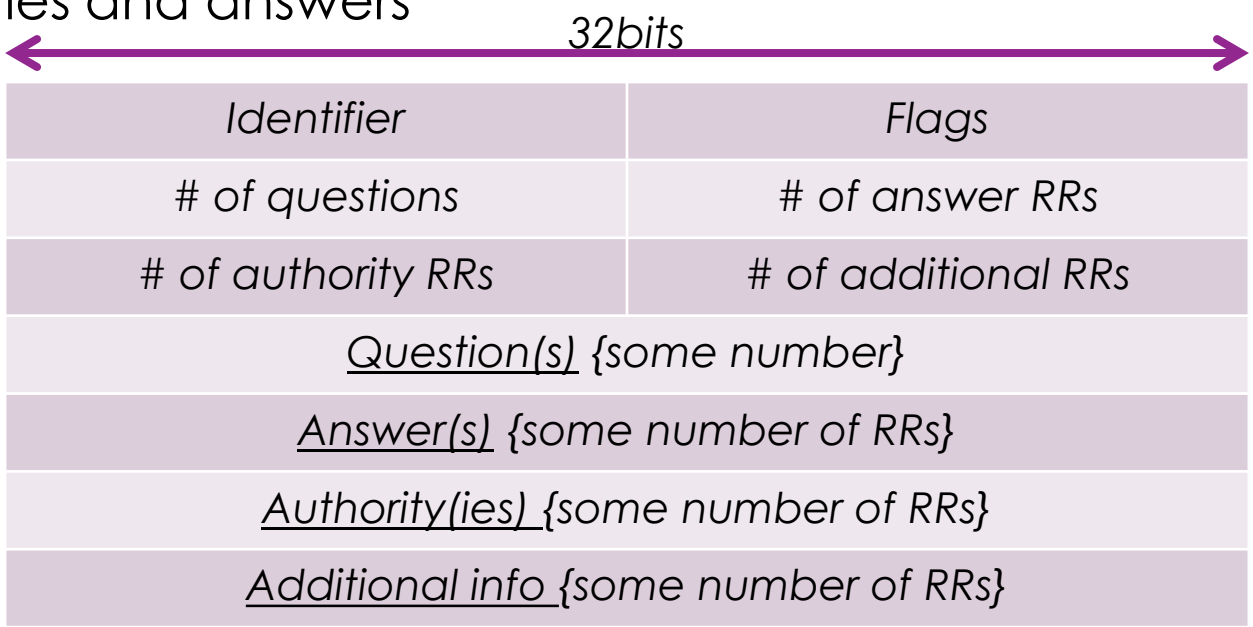
- **Zone transfers** – master/slave replication
  - Another type of DNS query/response

ANU returns the favour…?

- `adelaide.edu.au.  85674  IN  NS  ns2.adelaide.edu.au.`
- `adelaide.edu.au.  85674  IN  NS  authdns2.netcom.duke.edu.`
- `adelaide.edu.au.  85674  IN  NS  authdns1.netcom.duke.edu.`
- `adelaide.edu.au.  85674  IN  NS  authdns3.netcom.duke.edu.`
- `adelaide.edu.au.  85674  IN  NS  ns1.adelaide.edu.au.`
- `adelaide.edu.au.  85674  IN  NS  authdns4.netcom.duke.edu.`
- `adelaide.edu.au.  85674  IN  NS  ns.adelaide.edu.au.`
- `adelaide.edu.au.  85674  IN  NS  ns1.anu.edu.au.`

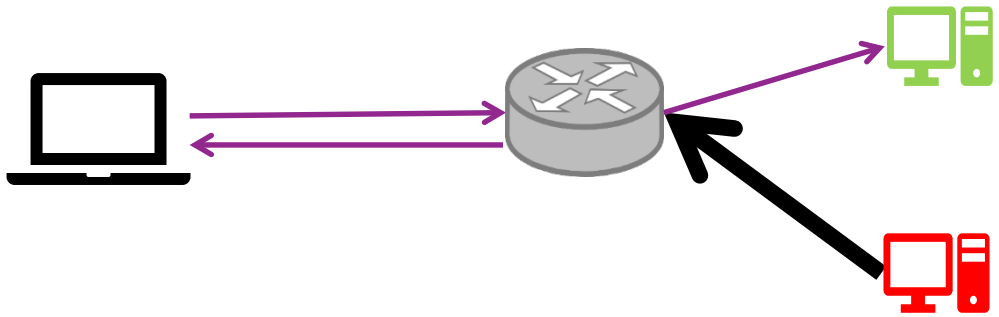
DNS Messages

- Simple, lightweight, UDP, port 53
  - ARQ – stateless servers
  - UDP: Need high-performance, minimise (TCP) load on the server
    - However, there is a TCP option… (for really large responses)
- Same packet structure for queries and answers
  - Just flags are changed
    - *Query or answer*
    - *Recursion desired*
    - *Recursion available*
    - *Reply is authoritative*
- Messages carry a 16-bit ID



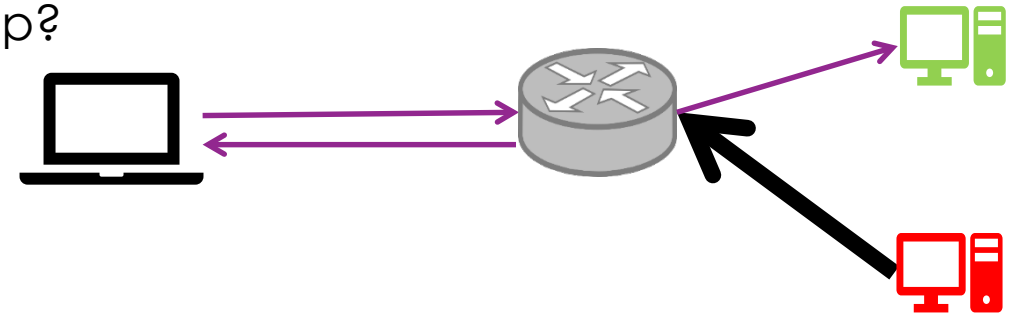
Of course this is secure. Right?

- Uhm – no.
- Villain-in-the-middle can corrupt/tamper/interfere with DNS queries
- Can redirect anybody, e.g. your connection to your bank's server…
  - Hack the authoritative nameserver?
  - "Hack" the caches/intermediary nameservers?
    - Actually spoofing - poison the cache – *get in first*



DNS (in)security

- Must be tricky?
  1. How does villain know what to send?
  2. How does villain make it look real?
  3. What happens when real reply turns up?



- Actually, not as hard as we'd like
  - Not that it's "easy"
- Don't try this at home, or anywhere, **ok?**

DNS (in)security

- What to send?
  - *Make the query yourself! Villain is just another client…*
- Make it real? *Circumvent DNS checks.*
  - Nameserver just checks headers:
    1. Is it from a known server?
    2. Does ID match?
    3. Does it help an outstanding-query?
  - but not the content

1. *Make source-IP the IP of an authority*
2. *Sends lots of replies with guessed/snooped ID (16-bit)*
3. *Send (flood!) the reply immediately after a query*

And third?

- What happens when the real response arrives?
  - Remember: Nameserver just checks
    - Is it from a known server?
    - Does ID match?
    - Does it help an outstanding-query?
  - But there's no longer an outstanding query…
  - And so that response gets ignored
  - And the DNS server is now caching your poisoned record…

Bring on DNS Security!

- Easy? DNSSEC…
  - Integrity and authenticity – it just adds **authentication**
  - Not about confidentiality (quite the opposite!)

- Extend DNS with new resource records

- Been discussed since 1997,
- Reasonably final by 2005,
- Root servers upgraded in 2010,
- but the rest, and the clients…?

New RRs

- RRSIG
  - Digital signatures of a set of domain records
    - Clusters of all your A, AAAA, MX, …
- DNSKEY
  - Public key for RRSIG signatures
  - Actually, two – Zone Signing Key (ZSK) and Key Signing Key (KSK).
    - KSK >> ZSK, reduces load on nameservers for key-validation. Need to trust the key!
- DS
  - Delegation Server key – for delegated zones
  - And CDNSKEY and CDS for delegated zone servers to propagate upwards

DNSSEC needs

- Try to minimise encryption overheads
  - DNS is a **very** popular transactional protocol – every transaction begins here!
  - Delays are bad.
  - Allow for new encryption techniques to be swapped in
    - And keys to be rolled-over

- Other RRs such as NSEC/NSEC3 – authenticated "no such name"
  - Unfortunately, this leaks zone information.
    - People like to probe networks…
  - Quote: "Either lie, or don't trust DNS to hold your secrets."
    - Avoid highlighting interesting endpoints.

So what changes?

- Query Nameservers as before, AND
- Validate replies for authenticity
  - From the top down, PKI chain of trust
  - Anchor is the root public key
  - Every reply carries the necessary keys

1. Use **key(root)** to check real-NS(.au)
2. Use **key(.au)** to check real-NS(.edu.au)
3. Use **key(.edu.au)** to check real-NS(.anu.edu.au)
4. Use **key(.anu.edu.au)** to confirm-IP(www.anu.edu.au)

Today?

- DNSSEC requires both clients and servers to update
- gTLDs (common ones) approaching 90%
- ccTLDs approaching 50%
- Lower domain levels from 2-90%

- Applications… maybe 10-15%?
- Don't even think about 'smart devices'
  - Web-cameras, baby monitors, home-security systems, …

Other DNS features

- Multiple names can point to one IP
  - One physical server hosting multiple virtual webservers
- One name can point to multiple IPs
  - Failover/load-balance
- Reverse lookups
  - Ensure connection from IP is from a domain, e.g. email spoofing, site validation
  - Uses a PTR record, in the *.in-addr.arpa* domain
  - Query for the PTR of *D.C.B.A.in-addr.arpa* points to the A record (the forward)

Other DNS features

- Sort-list:
  - Can prioritise from a list of response – e.g. 'in your prefix' vs 'not'
  - Useful for e.g. 'nearest' server, or for multi-interface servers
- Geopolitical-sensitivities – split DNS
  - What you get back depends on *where* you ask from
    - E.g. within some countries you can't get to some domains…
- Round-robin/"load-balancing"
  - Send a list, in different order each time
  - Broken a little by caching, and not knowing the actual load

Other DNS features

- LOC records
  - Latitude, longitude
  - and Altitude - from -100km up to +42000km
  - along with ‘precision’ of 1cm to 90000km

- SRV records
  - Identify service endpoints
    - That aren’t email (MX)
  - by Protocol Name and Type, and priority and weight
  - e.g. SIP, XMPP, STUN, Minecraft, …

"Dynamic DNS"

- Remember your NAT box at home?
  - With its changing IP address?
  - And that webserver running behind it?