Networking

G54CCS – Lecture 5

Richard Mortier

http://www.cs.nott.ac.uk/~rmm/teaching/2011-g54ccs/

Overview

- Connectivity
- Protocols
- Addressing
- Naming

Overview

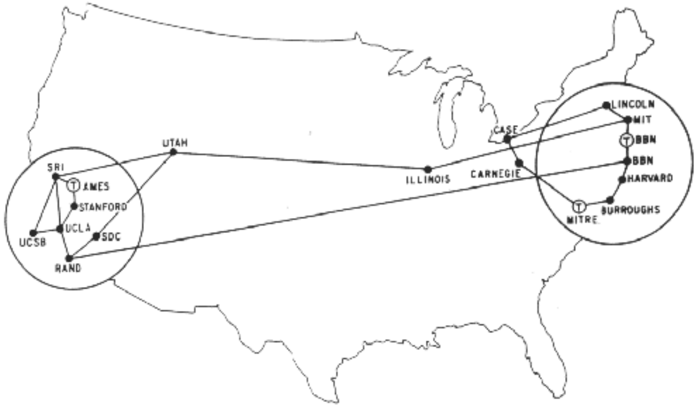
- Connectivity
  - Network Hierarchy
  - Network Resources
  - Security
- Protocols
- Addressing
- Naming

Connectivity

- What do networks do?
  - Transfer data between hosts (computers)
- What does the *Internet* do?
  - Transfer data between *networks*
- Cloud services rely on network connectivity
  - Other networks
  - Fixed and mobile devices
  - Wired and wireless access

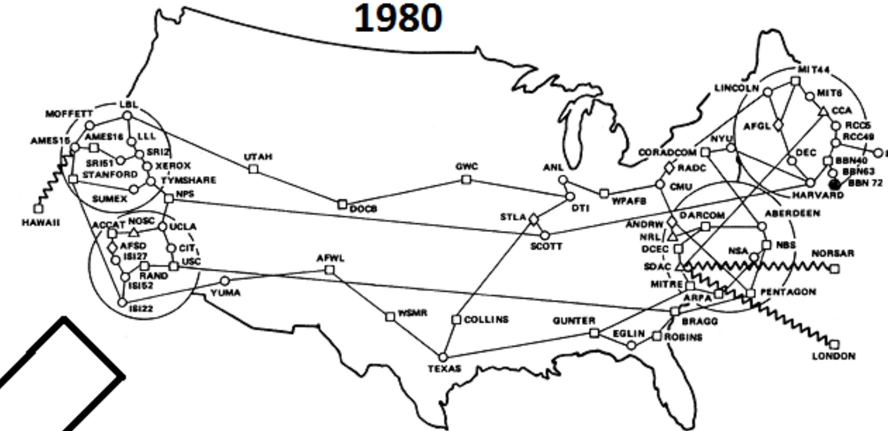
Networks

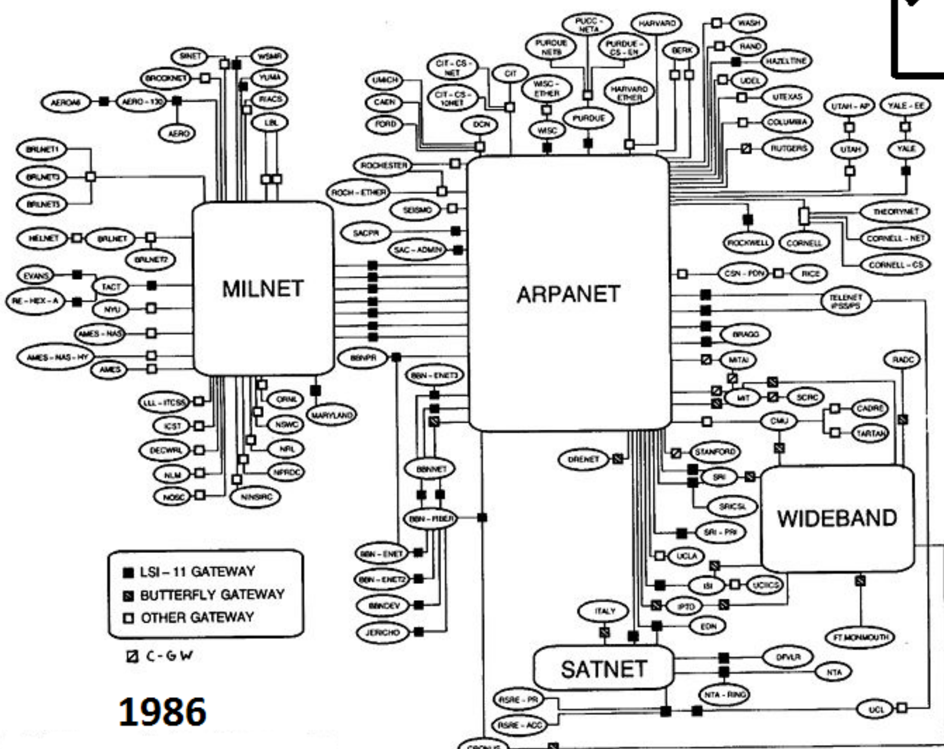
- A hierarchy of providers
- Local Area Network (LAN)
  - School, University
- Metropolitan Area Network (MAN)
  - University, EMMAN
- Wide Area Network (WAN)
  - National: JANET
  - International: Sprint, AT&T



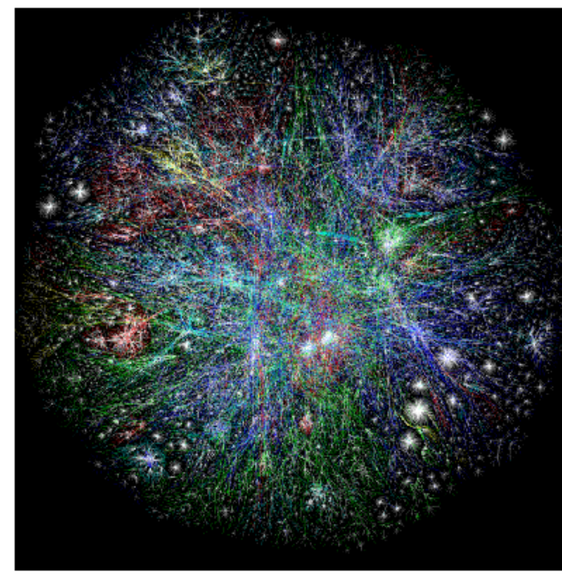
1971

1980





1986



2003

Network Resources

- Reliability
  - How much data is lost in transfer?
- Latency
  - How long does it take to get there?
- Bandwidth
  - How fast can data be transferred?
- Bandwidth *vs*. Throughput *vs*. Goodput
  - Raw signal *vs*.
  - Impact of encoding *vs*.
  - Impact of loss

Performance Variability

- Loss
  - Generally zero until something goes wrong
  - Loss due to overload *vs.* error
- Latency
  - Speed of light, switching, queuing
  - San Francisco—New York, Transatlantic ~ 75ms
- Bandwidth http://bit.ly/u6lzpQ
  - 2G = 14.4 kb/s ; 2.5G = 57.6 kb/s ; 3G = 384 kb/s
  - ADSL ~ 8 Mb/s ; Cable modem (DOCSIS) ~ 50 Mb/s
  - Wireless Ethernet = 2 – 600 Mb/s
  - Wired Ethernet = 100 Mb/s – 100 Gb/s
  - Disk ~ 3 Gb/s ; HDMI = 10.2 Gb/s ; RAM ~ 256 Gb/s

Network Security

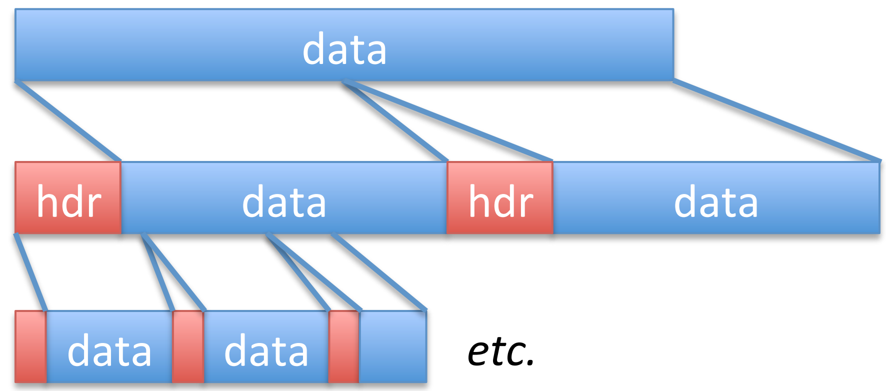
- Network security is *very* hard to get right
  - Too much software, too many attack vectors
- If you connect, anyone can send you anything
  - Firewalling helps
  - But can't stop malicious data
  - Or prevent careless users
- In any case, resources are remote
  - This means they can be consumed remotely
  - Your connectivity can be removed
  - Distributed Denial of Service

Overview

- Connectivity
- Protocols
  - Internet
  - Transport
  - Application
- Addressing
- Naming

Encapsulation

- Data comes down from higher layer in chunks
- Packetize to generate suitable sized chunks
- Encapsulate by prepending header
- Example: IP header




```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            data                            ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Transport

- IP provides packets with addresses
  - Hosts (interfaces) have addresses…
  - But we need to get data between programmes!
- Job of the *transport* protocol
  - TCP (reliable, ordered, bytestream) *vs*. UDP (not)
  - Both use *port numbers* to address *within the host*
  - Identify a particular process
  - E.g., ports 80 and 443 are (usually) a web server

HTTP

- On top of all that, we have HTTP
  - Originally for transferring static content
  - Webpages, images
  - Now the *de facto* standard application protocol
- Simple set of verbs
  - GET, POST, PUT, DELETE, HEAD, …
- Uniform resource addressing
  - `scheme://domain:port/path?query_string#fragment_id`
  - http://www.nottingham.ac.uk/search.aspx?q=mortier
  - Cf. labs, particularly #3

Overview

- Connectivity
- Protocols
- Addressing
  - Routing & Forwarding
  - Aggregation
  - Management
- Naming

Internet Addressing

- Host addresses in IPv4 are 32 bits long
  - Intended to be globally unique
  - Really it's *interfaces* that get allocated addresses
  - 32 bits is no longer enough
- Commonly depicted in 4 parts
  - E.g., `128.243.35.39` == `0x80f32327` == `2163417895`
- How does traffic get where it needs to?

Routing and Forwarding

- The Internet is connected by *routers*
  - Computers with many network interfaces (*ports*)
  - Running specialized software

Routing and Forwarding

Cisco CRS-1
Multi-shelf system

Routing and Forwarding

- The Internet is connected by *routers*
  - Computers with many network interfaces (*ports*)
  - Running specialized software
- *Routing* builds up information to figure out the correct port on which to *forward* a packet
  - Packets may be replicated, reordered, dropped
  - *Scalability*: networks may become large
  - *Dynamics*: need to handle host and link failures

Address Aggregation

- Need to determine the *route* from the *address*
  - Group addresses for efficiency
- Use *prefixes* with explicit *prefix lengths*
  - E.g., 172.16/12; 10/8; 192.168/16; 128.243/16
- Routing generates (prefix, port) mapping
  - The *forwarding table*
- Map *address* to *prefix* via *longest prefix match*
  - Means the *most specific* entry is used
  - If no match, then use *default route*; else drop

Longest Prefix Matching

192	168	10	12	/**32** – Host
1100 0000	1010 1000	0000 1010	0000 1100	
192	168	0	0	/**16**
1100 0000	1010 1000	0000 0000	0000 0000	
192	168	8	0	/**21**
1100 0000	1010 1000	0000 1000	0000 0000	
192	168	10	0	/**23**
1100 0000	1010 1000	0000 1010	0000 0000	
192	168	10	0	/**24**
1100 0000	1010 1000	0000 1010	0000 0000	
192	168	4	0	/**24**
1100 0000	1010 1000	0000 0100	0000 0000	

Address Management: Macro

- Internet Assigned Numbers Authority (IANA)
  - Co-ordinates number spaces
- Delegates *netblocks* to Regional Internet Registries (RIRs)
  - Africa, Asia/Pacific, North America, Latin America, EMEA
  - ...and down to National and Local registries (NIRs, LIRs)
- Approach appropriate registry for an allocation
  - Must provide suitable justification
  - Much harder to get a large allocation (== short prefix)
- Deeply manual process involving much politics
  - http://www.iana.org/numbers/ and http://www.iana.org/

Address Management: Micro

- Used to also be very manual
  - A big text file containing (IP, Ethernet) address map
  - Also needed to maintain subnet allocations for routing protocols
- Dynamic Host Configuration Protocol, DHCP
  - RFC 2131
  - "Can anyone give me an address?"
  - Usually provides a pile of other configuration information
- Address Resolution Protocol, ARP
  - "Does anyone know who's at <ipaddr>?"
- Both utilise IP broadcast address, 255.255.255.255

Network Address Translation

- Private Addressing, RFC1918
  - 172.16/12, 192.168/16, 10/8
  - Should never be externally routed
- Traditional NAT, RFC3022; see also RFC2663
  - Use private addresses internally
  - Map into a (small) set of routable addresses
  - Use source ports to distinguish connections
  - Requires IP, TCP/UDP header rewriting
    - Addresses, ports, checksums at least
- Not a security mechanism!

Address Shortages

- IPv4 supports 32 bit addresses
  - Advertised as nearly 400,000 netblocks
  - IANA pool exhausted 3/Feb/2011
  - First RIR pool exhausted 15/Apr/2011 (APNIC)
- Complete exhaustion in 2013—2014
  - Virtualization (cloud!) is accelerating this
- IPv6 supports 128 bit addresses
  - So not a problem?
  - ...except for the routing protocols
  - ...and all the associated services needing to move

Overview

- Connectivity
- Protocols
- Addressing
- Naming
  - DNS
  - Queries & Responses
  - Security

Naming

- IP addresses are all very well but
  - Not especially human-readable
  - Not always appropriate granularity
- HOSTS.TXT
  - A file (/etc/hosts) mapping names-numbers
  - Originally transferred to all hosts using FTP
  - Simple, but not terribly automatic or scalable
  - Scale via distributed hierarchical set of servers

DNS

- Domain Name Service, RFC1034/1035/2181
  - Client-Server protocol returning variety of records
  - Commonly uses UDP for queries but can use TCP
  - TCP used for bulk transfers between servers
- Hierarchy is "baked in"
  - Namespace divides into *zones*
  - Top Level Domains usually professionally managed
  - Root servers know how to get everywhere
- Not a 1:1 mapping between names and numbers!
  - E.g., Round-robin load-balancing

Name Service

- TLDs operated by registrars
- Delegate sub-domains to other registrars
  - ...and on down the hierarchy
- Eventually customer rents a subdomain/name
  - I.e., registrar installs appropriate records
- Setup primary and secondary servers
  - For subdomains
  - Separate IP netblocks, physical networks, *&c*
  - DNS is a *very* common single-point-of-failure

Queries

- Queries either *recursive* or *iterative*
  - A-B-C-D-A; or A-B-A, A-C-A, A-D-A
- Server either *authoritative* or *caching*
  - To discover authoritative requires query to root
  - Thus load on root servers is very high
- Caching server locally
  - Caches records each with an expiry time: *soft-state*
- Acquire zone's complete set via *zone transfer*
  - Often access controlled

Responses

- Name lookup uses following record types:
  - CNAME: name --> canonical name
    - `www.cs.nott.ac.uk. 61272 IN CNAME pat.cs.nott.ac.uk.`
  - A: name --> number
    - `pat.cs.nott.ac.uk.    68622IN   A    128.243.20.9`
    - `pat.cs.nott.ac.uk.    68622IN   A    128.243.21.19`
  - PTR: name (or number) --> name
    - `9.20.243.128.in-addr.arpa.  39617 IN PTR pat.cs.nott.ac.uk.`
  - NS: domain --> authoritative name server
    - `cs.nott.ac.uk.   10585IN   NS   ns1.nottingham.ac.uk.`
    - `cs.nott.ac.uk.   10585IN   NS   ns2.nottingham.ac.uk.`
    - `cs.nott.ac.uk.   10585IN   NS   marian.cs.nott.ac.uk.`
    - `cs.nott.ac.uk.   10585IN   NS   extdns1.warwick.ac.uk.`
    - `cs.nott.ac.uk.   10585IN   NS   extdns2.warwick.ac.uk.`
  - MX: domain --> mail exchange
    - `nott.ac.uk.  3600 IN   MX   1 mx191.emailfiltering.com.`
    - `nott.ac.uk.  3600 IN   MX   2 mx192.emailfiltering.com.`
    - `nott.ac.uk   3600 IN   MX   3 mx193.emailfiltering.com.`

DNS Security

- DNS is quite insecure
  - Cache poisoning
    - Caching and soft-state mean bad data propagates and can persist for some time
    - Even if through a simple mistake
  - Man-in-the-middle attacks
    - Iterative/Recursive queries almost demand this
  - Name spoofing
    - How clear is *your* font?
    - How well can *your* users spell?

Overview

- Connectivity
  - Network Hierarchy
  - Network Resources
  - Security
- Protocols
  - Internet
  - Transport
  - Application
- Addressing
  - Routing & Forwarding
  - Aggregation
  - Management
- Naming
  - DNS
  - Queries & Responses
  - Security