University of Glasgow | School of Computing Science

Networks & Operating Systems Essentials

Dr Angelos Marnerides

<angelos.marnerides@glasgow.ac.uk>

School of Computing Science, Room: S122

Today, on NOSE2…

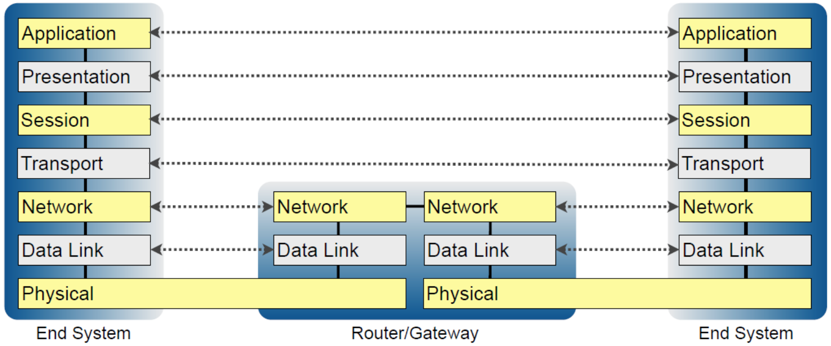
Source: https://xkcd.com/742/

Based on slides © 2017 Colin Perkins

NETWORK LAYER (L3)

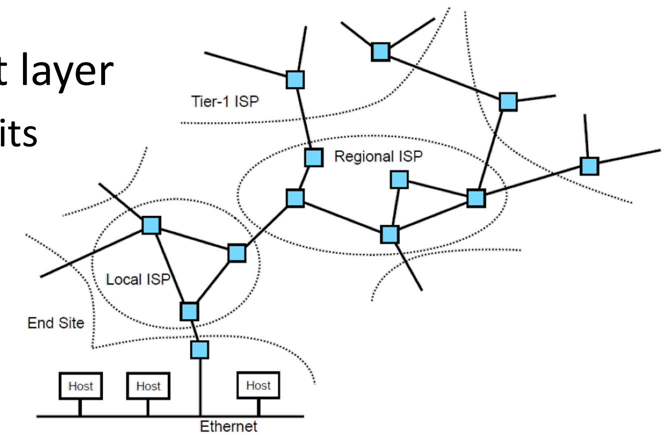
The Network Layer

- First end-to-end layer in the OSI reference model
  - Responsible for end-to-end delivery of data:
    - Across multiple link-layer hops and technologies
    - Across multiple *autonomous systems*
  - Building an Internet: a set of **inter**connected **net**works
- An *internet* comprises a set of interconnected networks
  - Each network administered separately
    - An autonomous system (AS)
    - Making independent policy and technology choices



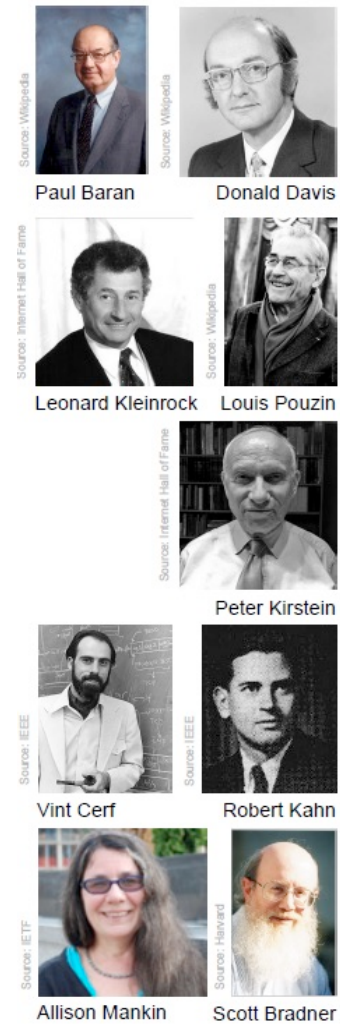
The Network Layer

- Components of *an* internet
  - A common end-to-end network protocol
    - Provide a single seamless service to transport layer
      - Delivery of data packets/provisioning of circuits
      - Addressing of end systems
  - A set of gateway devices (a.k.a. routers)
    - Implement the common network protocol
    - Hide differences in link layer technologies
      - Framing, addressing, flow control, error detection and correction
      - Desire to perform the least amount of translation necessary

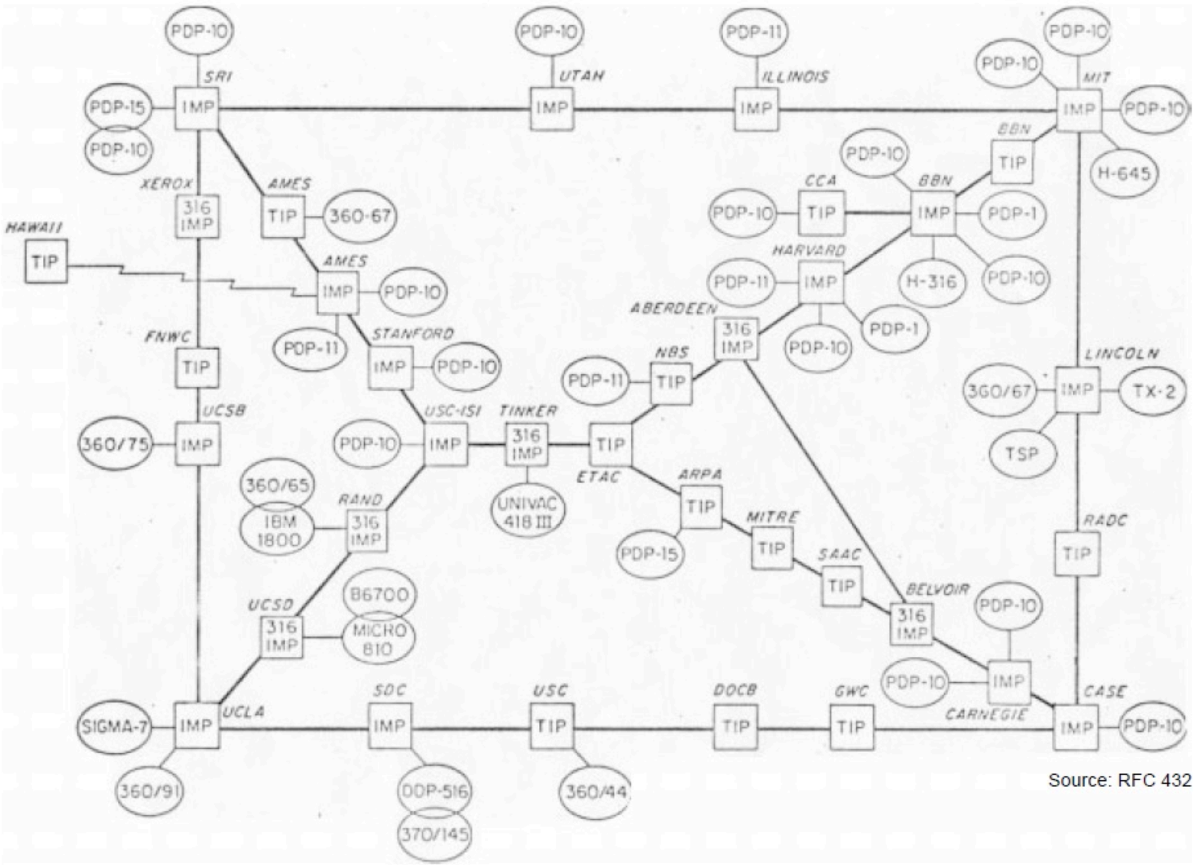


The Internet

- The globally interconnected networks running the *Internet Protocol* (IP)
  - 1965: Concept of packet switching
    - Paul Baran (RAND, USA), Donald Davies (NPL, UK)
  - 1969: Wide-area packet networks
    - ARPANET (US), CYCLADES (France)
  - 1973: First non-US ARPANET sites
    - UCL
  - 1974: Initial version of the Internet Protocol
    - Vint Cerf and Robert Kahn
  - 1981: Access to ARPANET broadened to non-DARPA-funded sites
    - NSF funds access for universities; production internetworking starts
  - 1983: Network switched to IPv4
  - 1992: Development of IPv6 starts
    - Initial IETF IPng effort led by Allison Mankin and Scott Bradner



The Internet



Source: RFC 432

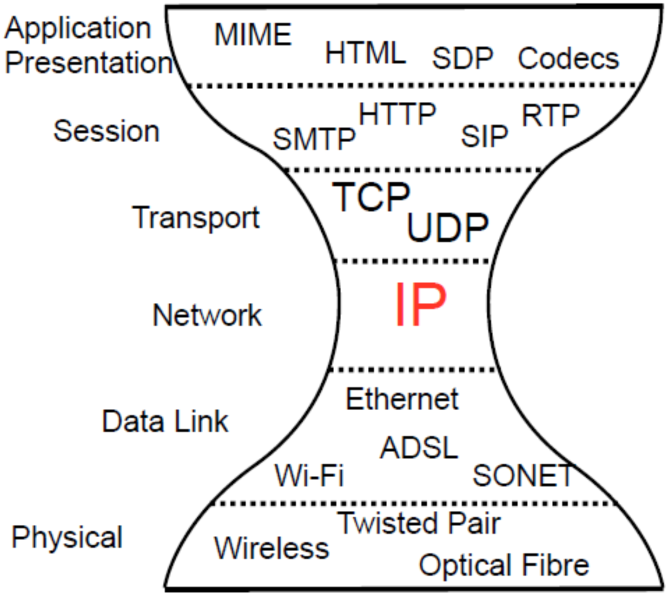
ARPA Network Map, December 1972

THE INTERNET PROTOCOL (IPV4 AND IPV6)

The Internet Protocol

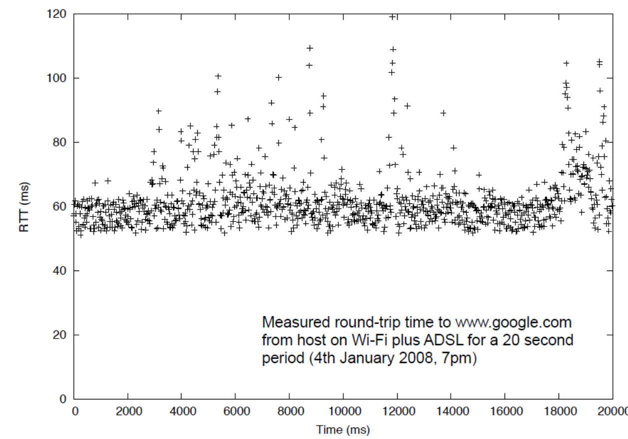
- IP provides an abstraction layer
  - Transport protocols and applications above
  - Assorted data link technologies and physical links below
  - A simple, best effort, connectionless, packet delivery service
  - Addressing, routing, fragmentation and reassembly

- Basic concepts:
  - Global inter-networking protocol
  - Hour glass protocol stack
    - Many transport & application layer protocols
    - Single standard network layer protocol (IP)
      - Packet switched network, best effort service
      - Uniform network and host addressing
      - Uniform end-to-end connectivity (subject to firewall policy)
    - Range of link-layer technologies supported

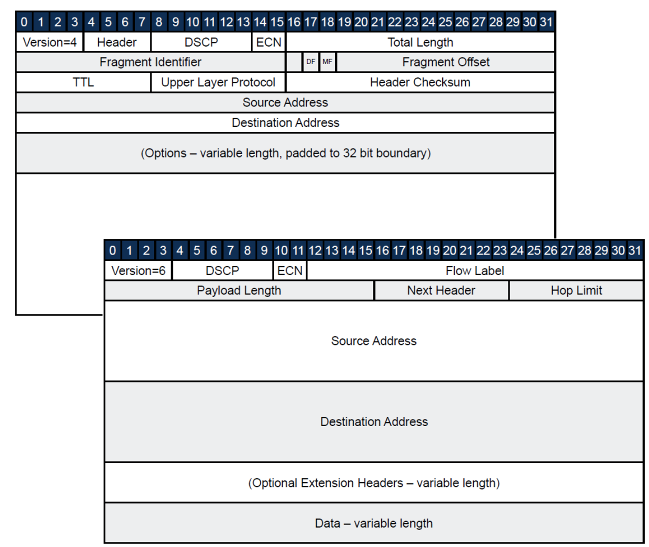


The Internet Protocol

- IP Service Model: Best effort, connectionless, packet delivery
  - Just send – no need to setup a connection first
  - Network makes its best effort to deliver packets, but provides **no** guarantees
    - Time taken to transit the network may vary
    - Packets may be lost, delayed, reordered, duplicated or corrupted
    - The network discards packets it can't deliver
  - Easy to run over any type of link layer
  - Fundamental service: can easily simulate a circuit over packets, but simulating packets over a circuit difficult

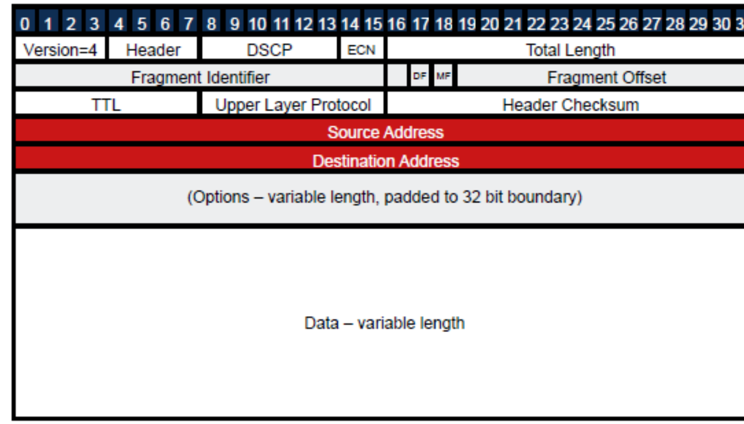


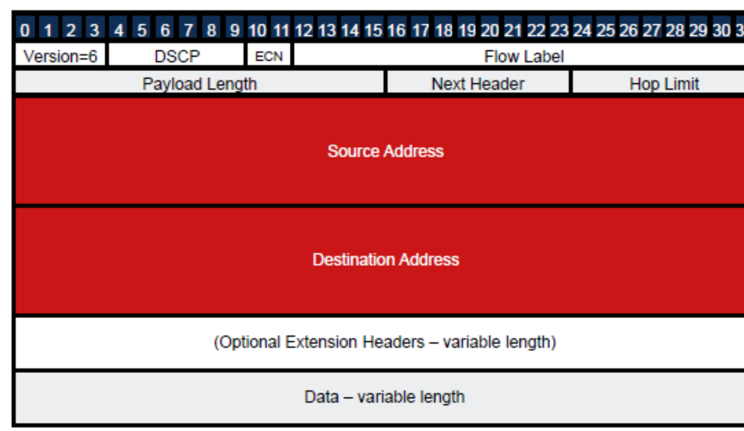
- Two versions of IP in use
  - IPv4 – the current production Internet
  - IPv6 – the next generation Internet
    - Compared to IPv4: simpler header format, larger addresses, removes support for fragmentation, adds flow label



Addressing

- Every network interface on every host is intended to have a unique address
  - Hosts may change address over time to give illusion of privacy
  - Addressable ≠ reachable: firewalls exist in both IPv4 and IPv6
- IPv4 addresses are 32 bits
  - Example: 130.209.241.197
  - Significant problems due to lack of IPv4 addresses → details later
- IPv6 addresses are 128 bits
  - Example: 2001:4860:4860::8844

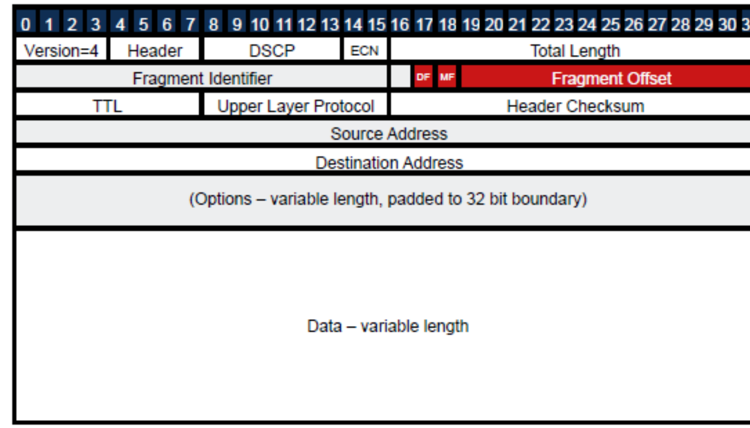


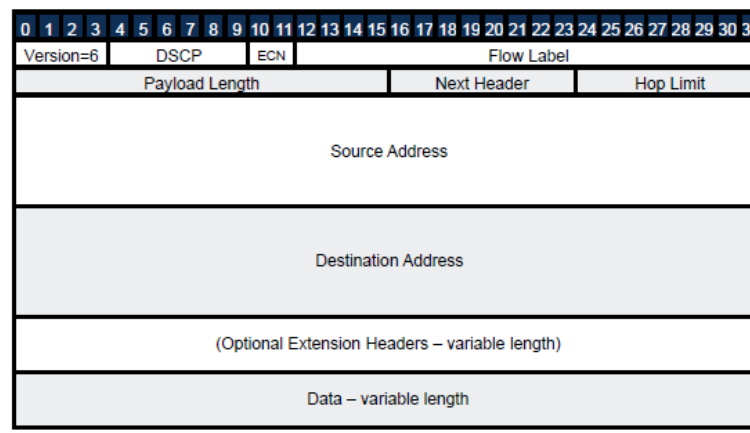


Fragmentation (placeholder)

- Link layer has a maximum packet size (MTU)
  - What is this?
  - Why is it needed?
  - How is it used?
  - Is it a good solution?

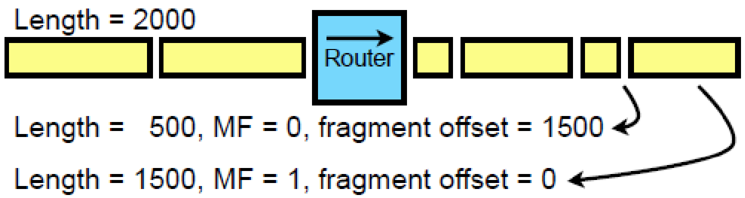
(you can post answers on Slido ☺)





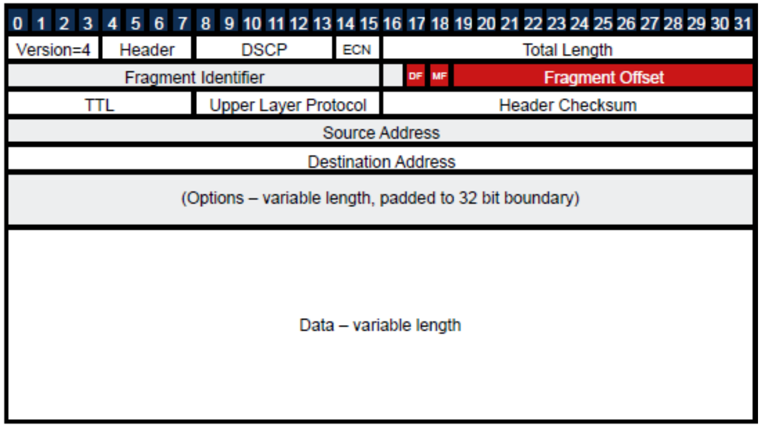
Fragmentation

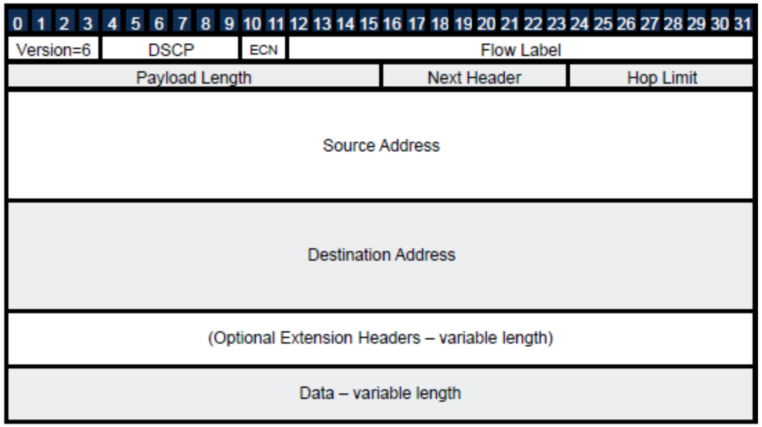
- Link layer has a maximum packet size (MTU)
  - Ethernet: 1500 bytes by default
- IPv4 routers will fragment packets that are larger than the MTU
  - MF bit is set if more fragments follow: reconstruct using fragment offset and fragment



  - DF bit is set → routers shouldn't fragment, must discard large packets

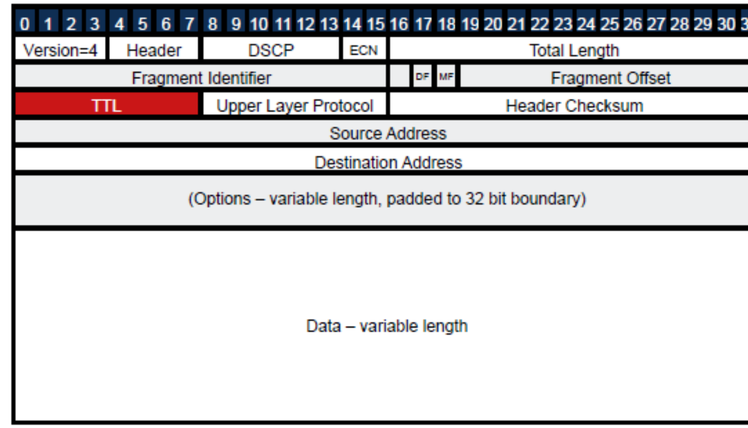
- IPv6 doesn't support fragmentation
  - Hard to implement for high rate links
  - End-to-end principle

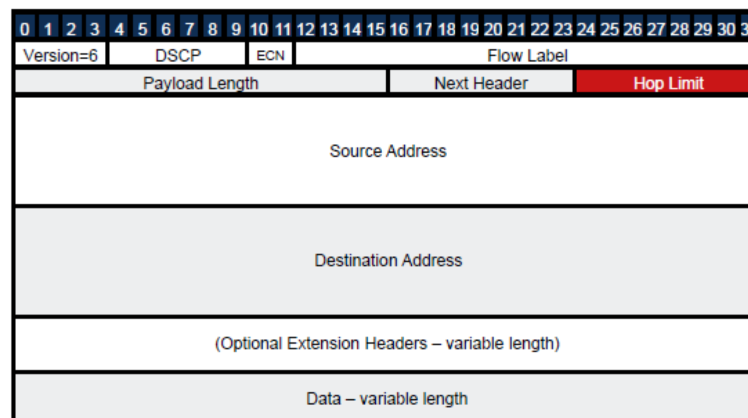




Loop Protection

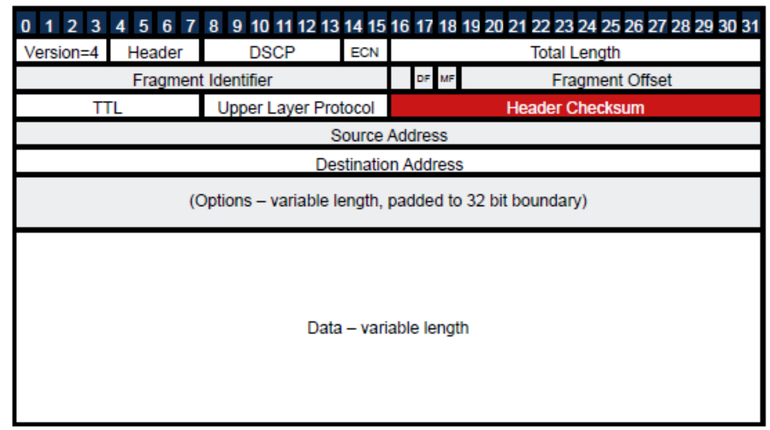
- Packets include a forwarding limit:
  - Set to a non-zero value when the packet is sent (typically 64 or 128)
  - Each router that forwards the packet reduces this value by 1
  - If zero is reached, packet is discarded
- Why is it needed?
  - Stops packets circling forever if a network problem causes a loop
    - Assumption: network diameter is smaller than initial value of forwarding limit

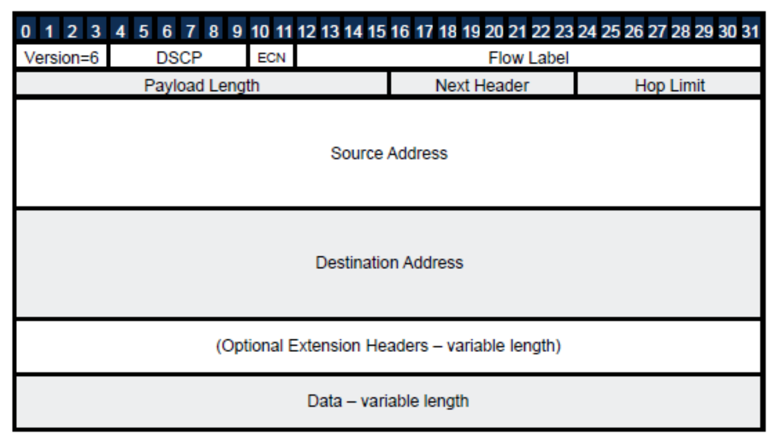




Header Checksum

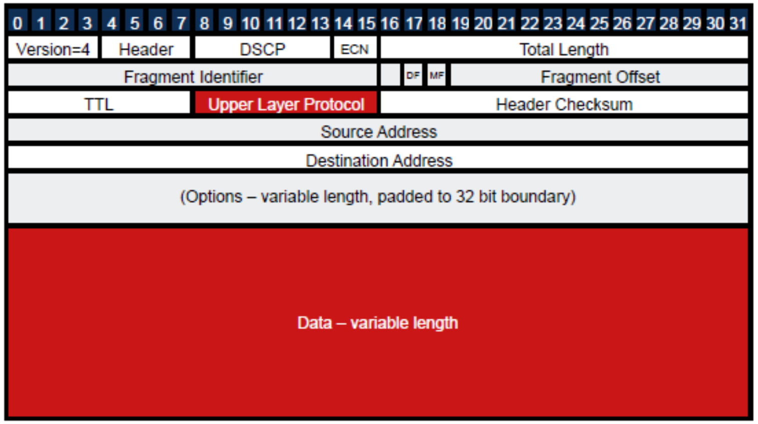
- IPv4 header contains a checksum to detect transmission errors
  - Conceptually similar to link-layer checksum, although uses a different algorithm
  - Protects the IP header only, not the payload data protected
    - Payload data must be protected by upper layer protocol, if needed
  - Isn't the checksum part of the IP header? (you can tell me what you think during the Q&A session ☺)
- IPv6 does not contain checksum
  - Assumes the data is protected by a link layer checksum

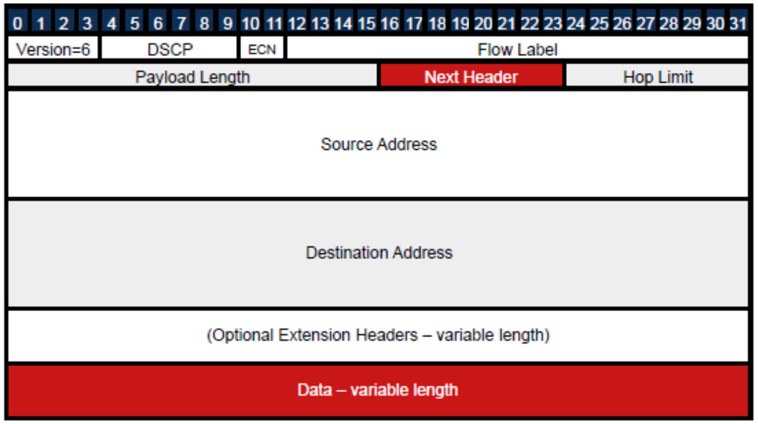




Transport Layer Protocol Identifier

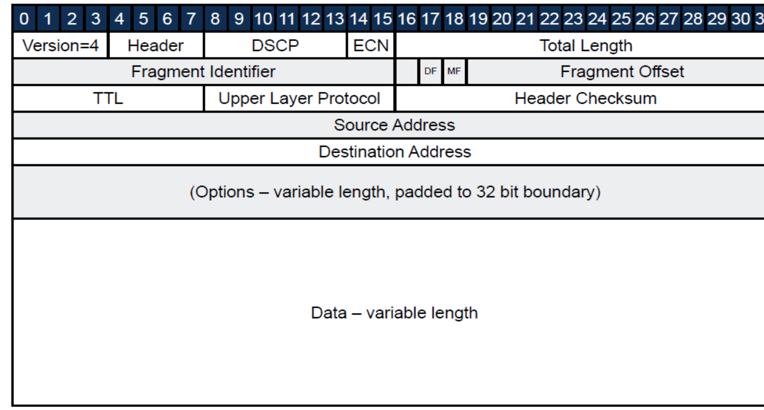
- Network layer packets include the transport layer data as payload
- Must identify what transport layer protocol is used, to pass the data to the correct upper-layer protocol
  - TCP = 6
  - UDP = 17
  - DCCP = 33
  - ICMP = 1
- Protocols managed by the IANA
  - http://www.iana.org/assignments/protocol-numbers/

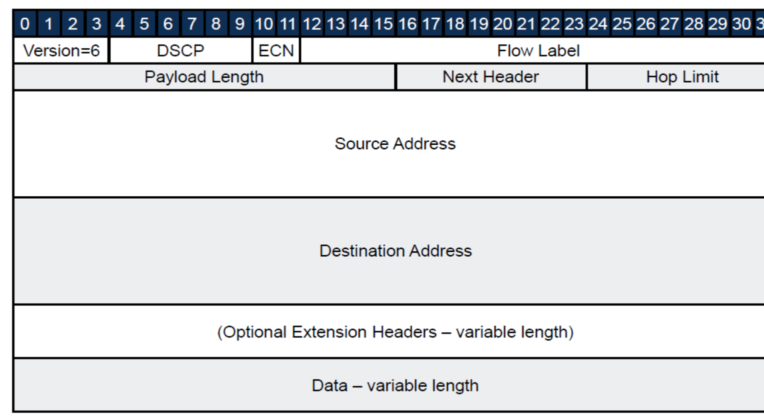




Aside: Misc

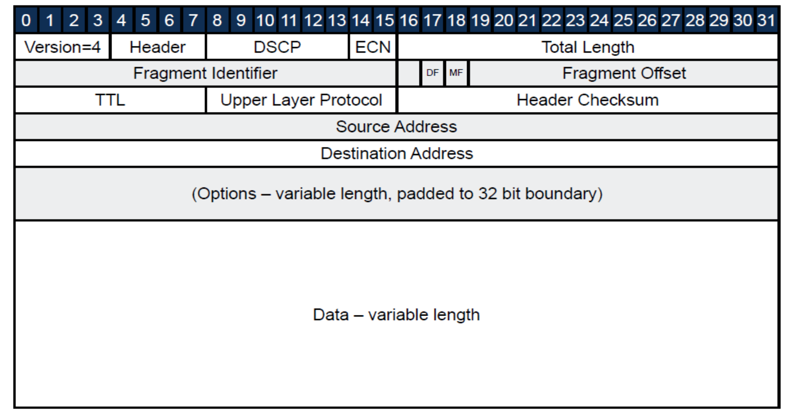
- Differentiated services (DiffServ)
  - End systems can request special service from the network
    - Telephony or gaming might prefer low latency over high bandwidth
    - Emergency traffic could be prioritised
    - Background software updates might ask for low priority
  - Signalled by differentiated service code point (DSCP) field in header
  - Provides a hint to the network, not a guarantee
    - Often stripped at network boundaries
    - Difficult economic and network neutrality issues
      - Who is allowed to set the DSCP and what are they charged for doing so?
  - IPv6 provides a flow label to group related traffic flows together

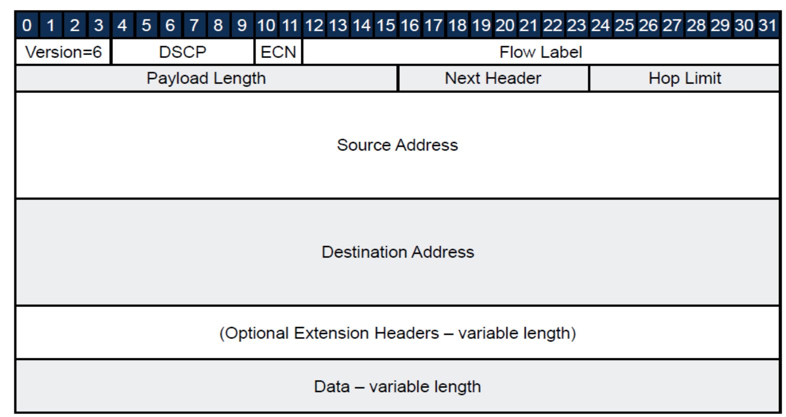




Aside: Misc

- Explicit Congestion Notification
  - Routers typically respond to network congestion by dropping packets
    - A “best effort” packet delivery service
    - Transport protocols detect the loss, and can request a retransmission if necessary
  - Explicit congestion notification gives routers a way to signal congestion is approaching
  - If a sending host enables ECN, routers monitor link usage and can indicate congestion is imminent
  - A host seeing that congestion is imminent needs to reduce it’s sending rate – or the congested router will start dropping packets





IPv4 or IPv6?

- IPv4 has reached end-of-life: insufficient addresses
- IPv6 intended as long term replacement for IPv4
  - Primary goal: increase the size of the address space, to allow more hosts on the network
  - Also simplifies the protocol, makes high-speed implementations easier
- Not yet clear if IPv6 will be widely deployed
  - But, straight-forward to build applications that work with both IPv4 and IPv6
  - DNS query will return IPv6 address if it exists, else IPv4 address; all other communication calls use the returned value
  - Write new code to support both IPv6 and IPv4

- To get a better idea on the IPv6 deployment status check : https://www.worldipv6launch.org/measurements/

IP Addresses

- How to name hosts in a network?
  - Is the address an identity or a location?
    - Does it name the host, or the location at which it attaches to the network
  - How should addresses be allocated?
    - Hierarchical or flat?
  - What is the address format?
    - Human or machine readable?
    - Textual or binary? Structured or unstructured?
    - Fixed or variable length? How large?

Identity and Location

- Addresses can denote host identity
  - Give hosts a consistent address, irrespective of where or when they attach to the network
  - Simple upper-layer protocols
    - Transport layer and applications unaware of multi-homing or mobility
  - Puts complexity in network layer
    - Network must determine location of host before it can route data
    - Often requires in-network database to map host identity to routable address
    - E.g., mobile phone numbers

- Alternatively, an address can indicate the location at which a host attaches to the network
  - Address structure matches the network structure
    - Network can directly route data given an address
    - E.g., geographic phone numbers: +44 141 330 4256
  - Simplifies network layer, by pushing complexity to the higher layers
    - Multi-homing and mobility must be handled by transport layer or applications – transport layer connections break when host moves

Address Allocation & Format

- Are addresses allocated hierarchically?
  - Allows routing on aggregate addresses
    - E.g., phone call to +1 703 243 9422
  - Forces address structure to match network topology
  - Requires rigid control of allocations
- Or is there a flat namespace?
  - Flexible allocations, no aggregation → not scalable
- How about format? Textual or binary? Fixed or variable length?
  - Fixed length binary easier (faster) for machines to process
  - Variable length textual easier for humans to read
  - Which are you optimising for?

IP Addresses

- IP addresses have the following characteristics:
  - They specify location of a network interface
  - They are allocated hierarchically
  - They are fixed length binary values
    - IPv4: 32 bits
    - IPv6: 128 bits
- Domain names are a separate *application level* namespace
- Both IPv4 and IPv6 addresses encode location
  - Addresses are split into a network part and a host part
    - A netmask describes the number of bits in the network part
    - The network itself has the address with the host part equal to zero
    - The broadcast address for a network has all bits of host part equal to one
      - Allows messages to be sent to all hosts on a network
  - A host with several network interfaces will have one IP addresses per interface
    - E.g., laptop with an Ethernet interface and a Wi-Fi interface will have two IP addresses
- Example:
  - IP address: 130.209.241.197 => `10000010 11010001 11110001 11000101`
  - Netmask: 255.255.240.0 => **11111111 11111111 1111**0000 00000000
  - Network = 130.209.240.0/20 => **10000010 11010001 1111**0000 00000000
  - Broadcast: 130.209.255.255 => **10000010 11010001 1111**1111 11111111

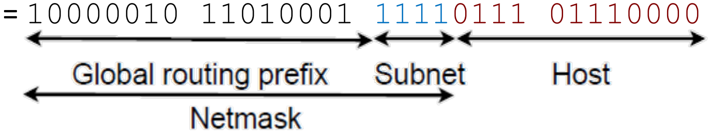
Aside: Classes of IP Addresses

- IP addresses used to be allocated so the netmask was a multiple of 8 bits
    - Class A → a /8 network (~16 million addresses)
    - Class B → a /16 network (65536 addresses)
    - Class C → a /24 network (256 addresses)
  - Old terminology, still used sometimes
  - Inflexible, and wasted addresses

- Arbitrary length netmask allowed since 1993:
  - The Glasgow SoCS network is a /20

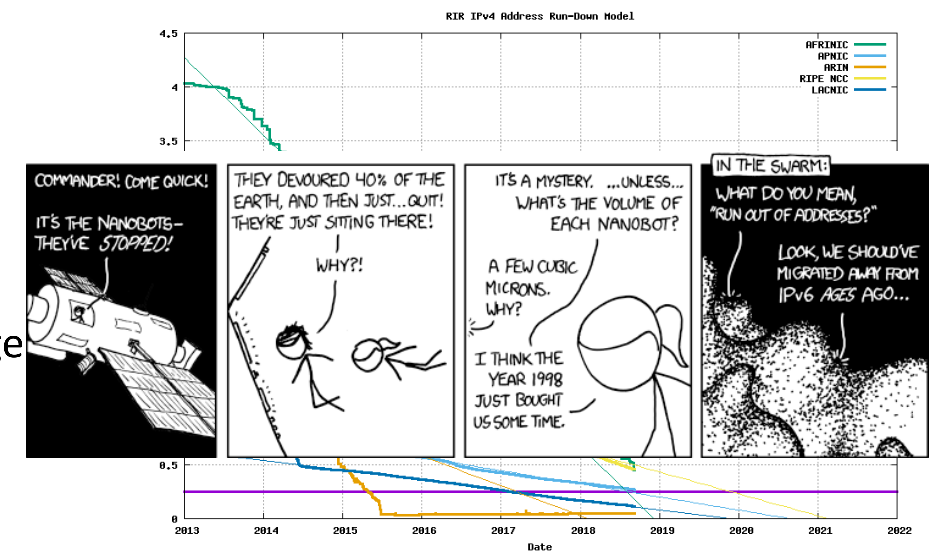
Aside: IPv6 Addresses

- IPv6 uses 128 bit binary addresses, written as 8 ":" separated 16 bit hexadecimal fields
    - `2a00:1098:0000:0086:1000:0000:0000:0010`
- Usually written in a shortened form [RFC 5952]
  - Leading zeros in each 16 bit field are suppressed
  - A run of more than one consecutive 16 bit field that is all 0 is omitted and replaced with a "::"
    - If there is more than one such run, the longest is replaced
    - If there are several runs of equal length, the first is replaced
  - The "::" must not be used to replace a single 16 bit field
    - `2a00:1098:0:86:1000::10`
- Local identifier part of IPv6 address is 64 bits
    - `2001:0db8:85a3:08d3:1319:8a2e:0370:7334`
  - Can be derived from Ethernet/Wi-Fi MAC address
    - 48-bit IEEE MAC: `0014:5104:25ea`
    - Expand to 64 bits: `0014:51ff:fe04:25ea`
    - Invert bit 6: `0214:51ff:fe04:25ea`
  - Or randomly chosen, with bit 6 set to zero, to give illusion of privacy
- Routers advertise network part, hosts auto-configure address
    - `2001:0db8:85a3:08d3:1319:8a2e:0370:7334`
- Network part is split into global routing prefix (up to 48 bits) and a subnet identifier
    - `130.209.247.112 = 10000010 11010001 11110111 01110000`



IP Address Management

- IPv4 has 2^{32} = 4,294,967,296 addresses
  - IANA administers the pool of unallocated addresses
    - Historically would assign addresses directly to ISPs, large enterprises, etc.
    - Now, addresses assigned to regional Internet registries (RIRs) as needed:
      - Allocations made one /8 (224 = 16,777,216 addresses) at a time
  - RIRs allocate addresses to ISPs and large enterprises within their region; ISPs allocate to their customers
- IANA has allocated **all** available addresses to RIRs
  - Last allocation on 3 February 2011
- In practical terms, we have run out of IPv4 address space
- IPv6 provides 128 bit addresses
  - If deployed it will solve address shortage for a long time
    - 2^{128} = 340,282,366,920,938,463,463,374, 607,431,768,211,456 addresses
    - Approx. 665,570,793,348,866,943,898,599 addresses per m^2 of the Earth's surface



Source: http://ipv4.potaroo.net/

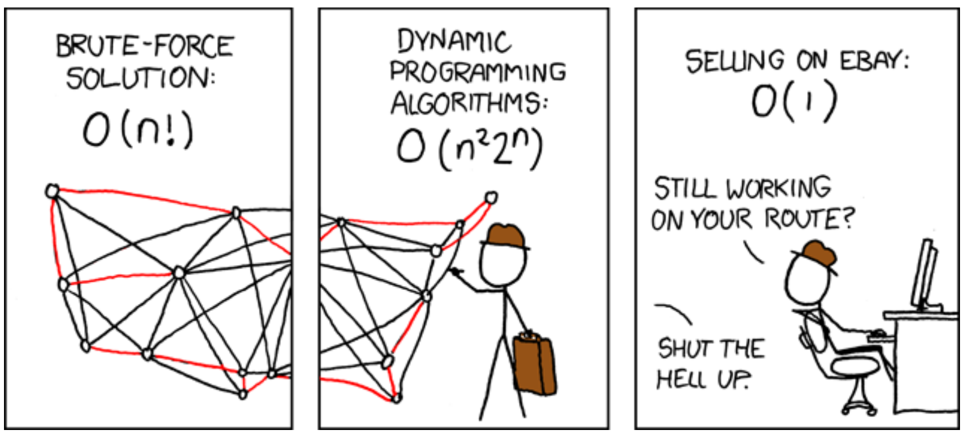
IPv6 Deployment Issues

- IPv6 requires changes to every single host, router, firewall, and application...
  - Significant deployment challenge!
  - Host changes done: MacOS X, Windows, Linux, FreeBSD, Symbian, iOS, Android, etc.
  - Backbone routers generally support IPv6, home routers and firewalls are starting to be updated
  - Many applications have been updated

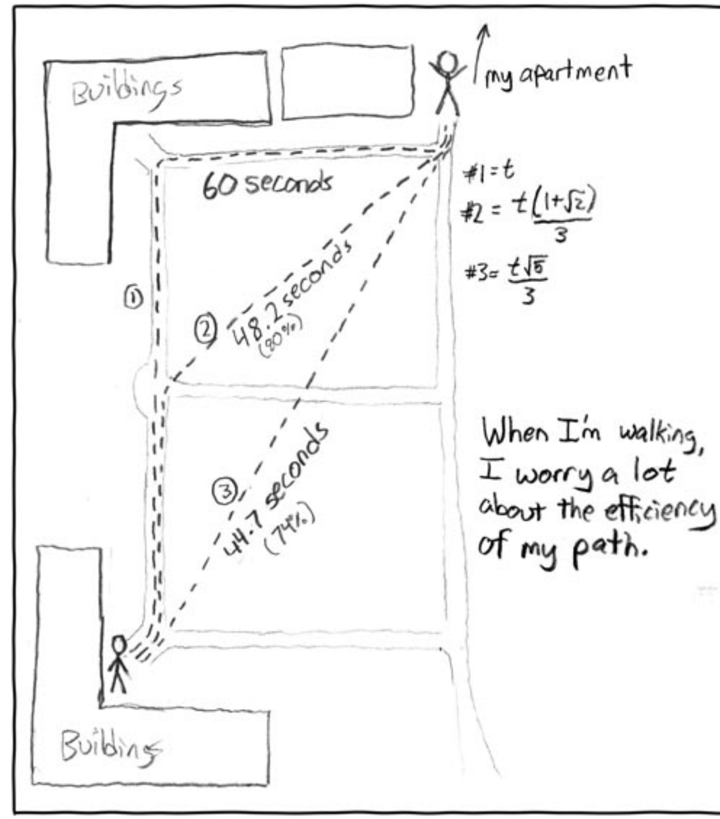
Reading Material

- **Peterson & Davie “Computer Networks: A systems approach”**: Chapter 3, sections 3.2, Chapter 4, section 4.1
- **Kurose & Ross “Computer Networking: A top-down approach”** : Chapter 4, sections 4.1, 4.4, 4.6
- **Tanenbaum & Wetherall “Computer Networks” 5th edition**: Chapter 5, sections 5.1, 5.5, 5.6
- **Bonaventure “Computer Networking” : https://www.computer-networking.info/1st/html/network/network.html**

Coming up next...



Source: https://xkcd.com/399/



Source: https://xkcd.com/85/