The Internet Ecosystem and Evolution

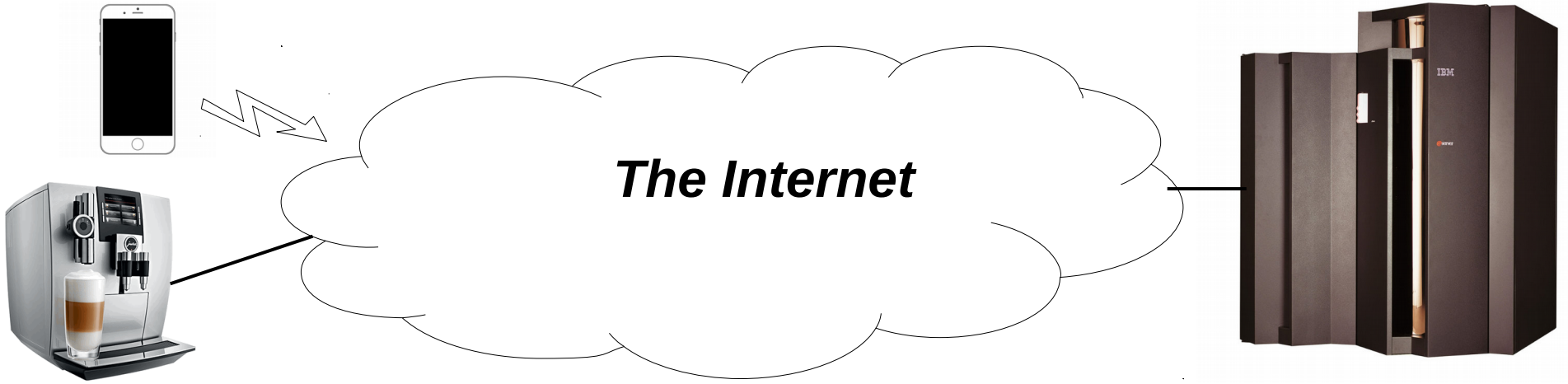
Contents

- The TCP/IP protocol suite
- Layered protocol stacks
- The link layer
- The network layer: IP
- The transport layer: TCP/UDP

Network layers

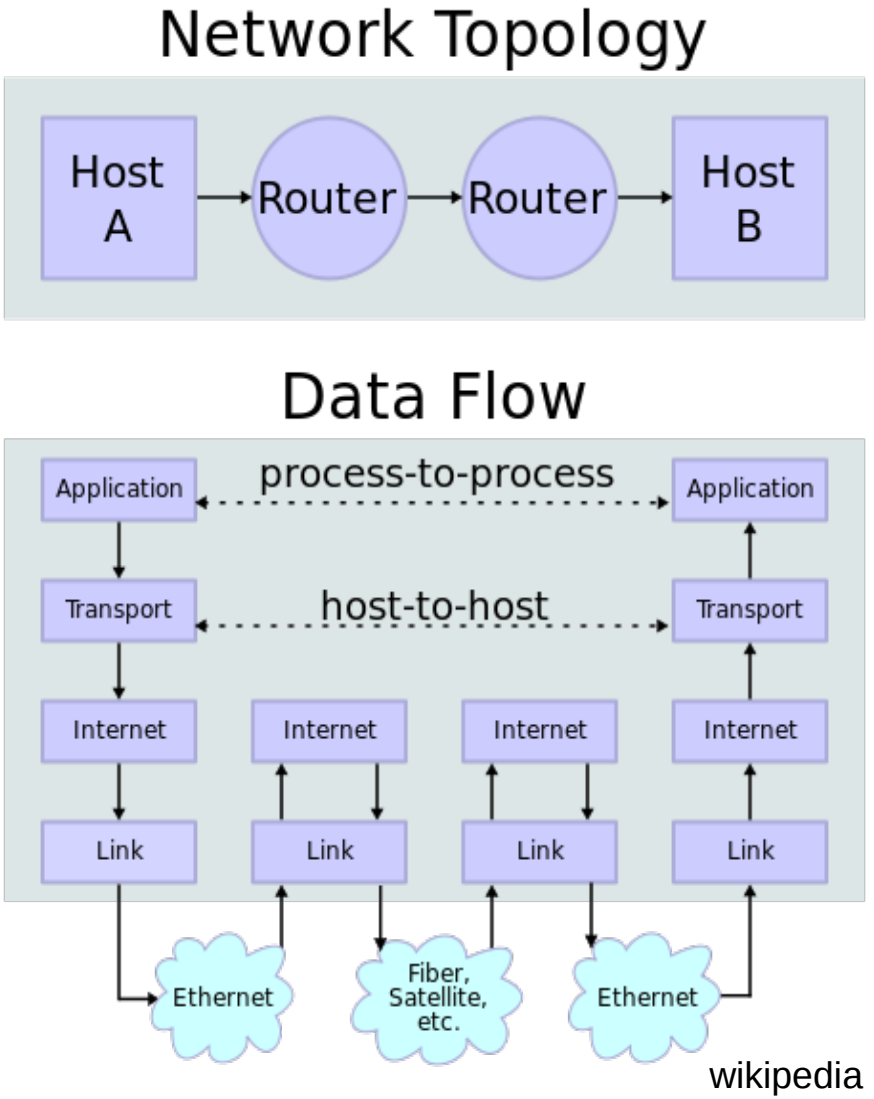
Protocols and layers

- The Internet is a complex system, connecting
  - everthing to everything, from intelligent IoT coffee machines to supercomputers and iPhones
  - every country to every other country (more or less), e.g., Palestine to Israel, the USA to China, etc.
- Standard **network protocols** ensure that any two remote hosts can talk to each other



Protocols and layers

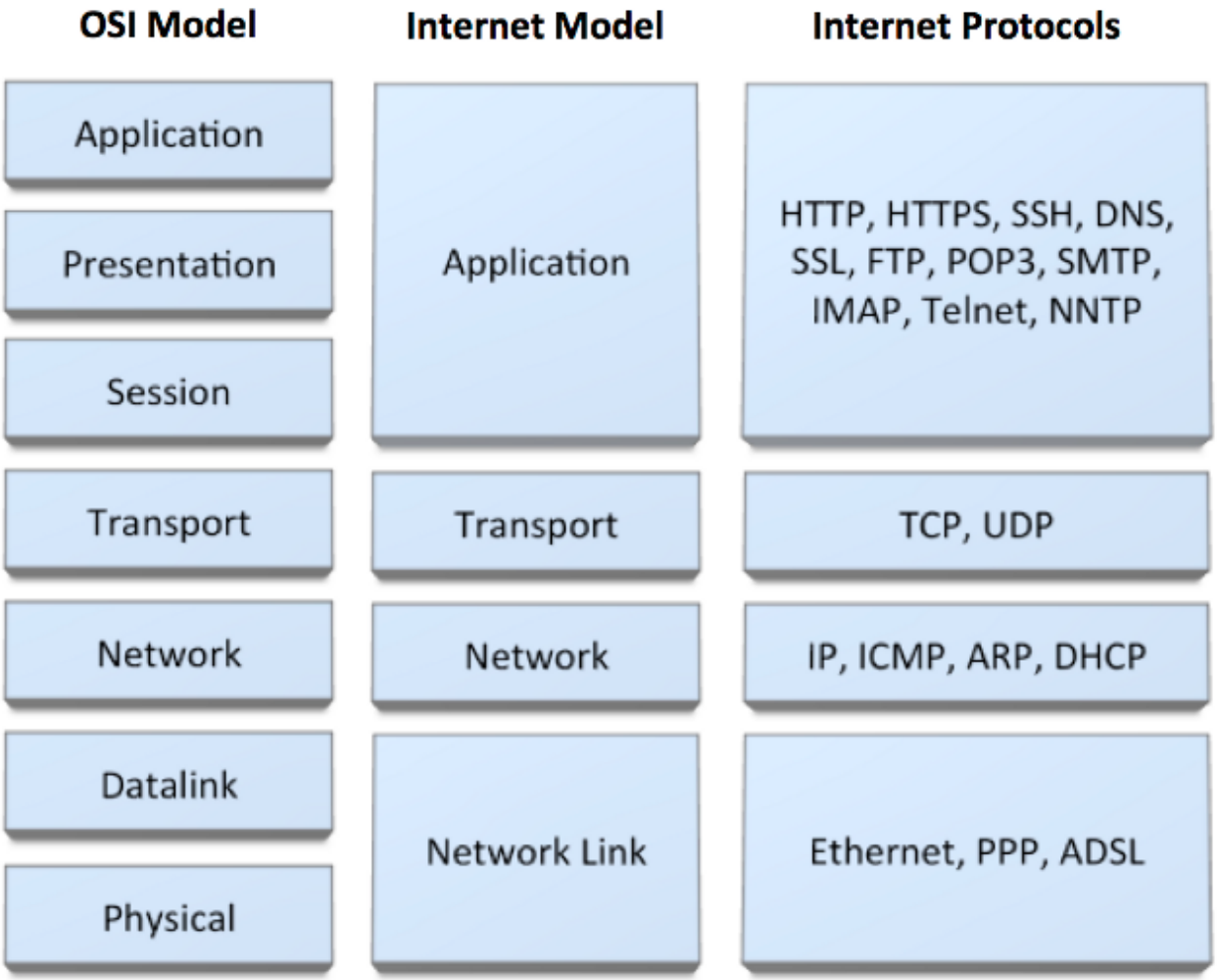
- **Protocol stack:** collection of all the functions a host must implement to speak to other hosts
- **Protocol layer:** a module of the protocol suite responsible for a well-defined subset of the protocol stack's functionality



wikipedia

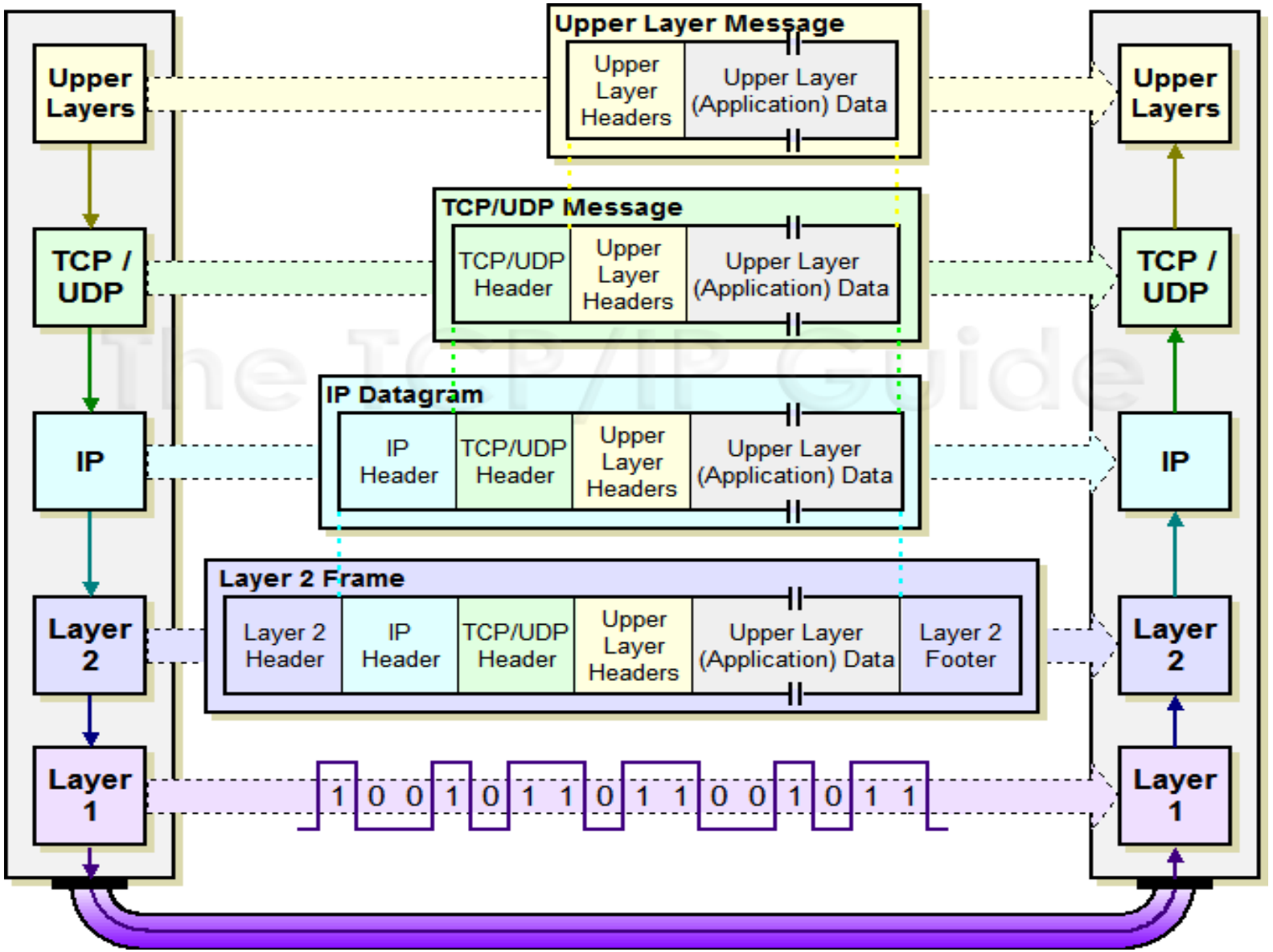
The TCP/IP protocol suite

- ISO/OSI: an “ideal” design, TCP/IP is different



http://vichargrave.com/network-programming-design-patterns-in-c

Encapsulation



wikipedia

TCP/IP: Principles

- **End-to-end:** per-application “state” only at hosts, the core does not store micro-state (scalability)
- **Connectionless design:** this is a consequence of the E2E principle, the core scales with network size *N*, not with the number of end-users and applications *M (M>>N)*
- **Resilience:** resilience to large-scale failures (a “cold war” design, under nuclear threat)
  - no central control: completely distributed design
  - dynamic routing: adapt to (changing) topology
  - packets of a single flow may take different routes
- **Robustness:** be conservative in what you send, be liberal in what you accept (Postel's law on extreme interoperability)

Connectionless vs. connection-oriented protocols

	Connectionless	**Connection-oriented**
Circuit setup	No	Before communication
Addressing	Every packet contains the fully-specified destination address	Packets only contain the virtual circuit identifier (shorter than address)
Packet forwarding	Per-packet forwarding decision	(Circuit-)switching
State information	Forwarding tables: *O(N)*, where *N* is the number of hosts/addresses	Virtual circuit switching tables: *O(M)*, *M* is the number of applications in talk: *M >> N*
Admission control and congestion control	Complex (but doable)	Simple

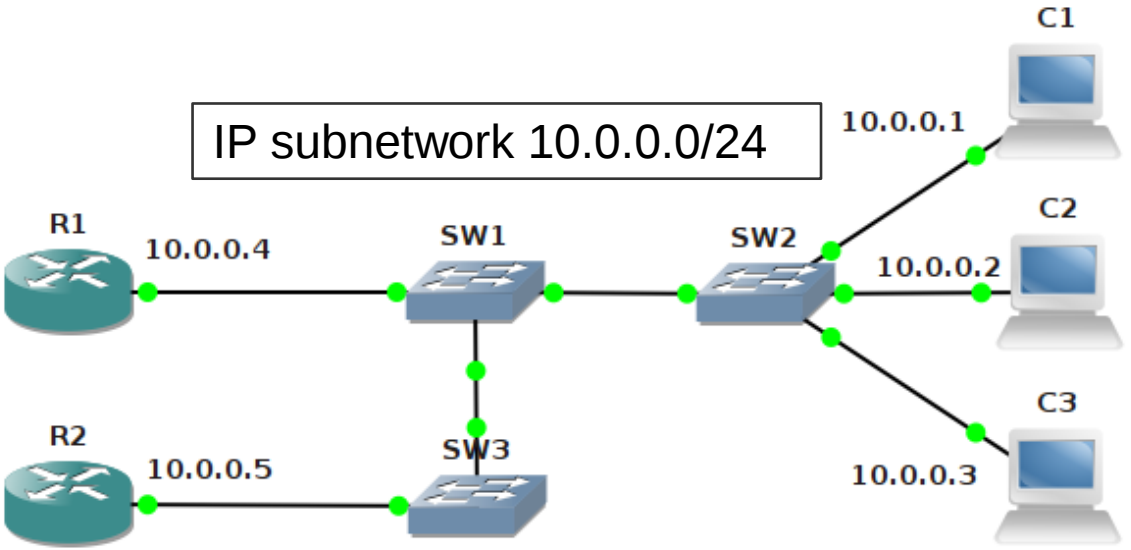
The link layer

Terminology: Hosts and links

- The Internet is made up by an astronomical number of **local-area networks** (LANs, like (Ethernet, WiFi, etc.), leased lines, etc., connected into a common network
- **Host:** a computer with one or more network interfaces/ports endued with a **unique IP address**
- **Link:** a LAN infrastructure connecting a **subnetwork of hosts**

Terminology: Hosts and links

- Interfaces at routers `R1` and `R2` and terminals `C1`, `C2` and `C3` are set with an IP address
- Ethernet switches `SW1`, `SW2`, and `SW3` do not have an IP addresses
- The LAN that is transparent to the IP hosts

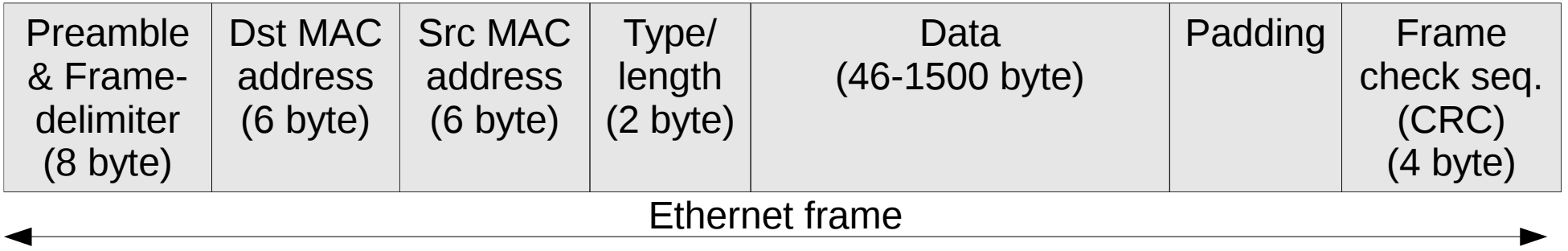


Link layer

- **Layer-2:** usually a local-area network (LAN) protocol (Ethernet, PPP, WiFi,...), but can also be a point-to-point (P2P) serial link, a long-haul leased line, etc.
- **Function: transport of IP packets between neighboring IP hosts** (connected to the same subnet)
- Service model:
  - re/assembly of IP packets to/from L2 frames
  - ordered, reliable (acknowledged) transmission
  - medium access control (MAC): access to the shared the transmission medium is controlled by a MAC
  - potentially, the link layer protocol also provides error detection, error correction, etc.

The Ethernet

- The IEEE 802.3 protocol suite
  - 3 Mbit/sec ↔ 100 Gbit/sec, 48 bit flat address space
- CSMA/CD medium access by default
- Ethernet segments can be interconnected:
  - **hub/repeater:** all Ethernet frames are “blindly” repeated to all connected segments
  - **switch:** just frames whose destination is on segment (Spanning Tree Protocol, SP Bridging)



The network layer

Network layer: Internet Protocol

- The “lingua franca” of all hosts connected to the Internet: IPv4 (slow transition on the way to IPv6)
- **Functions: unreliable, connectionless, best-effort datagram service for the transport layer**
  - **unreliable:** no error detection/correction (only header chksum!)
  - **connectionless:** no connection setup/tead-down before/after communication
  - **datagram:** every packet's IP header contains the destination IP address, packets are routed individually
  - **best-effort:** „all packets are created equal” (?!)

Network layer: Services

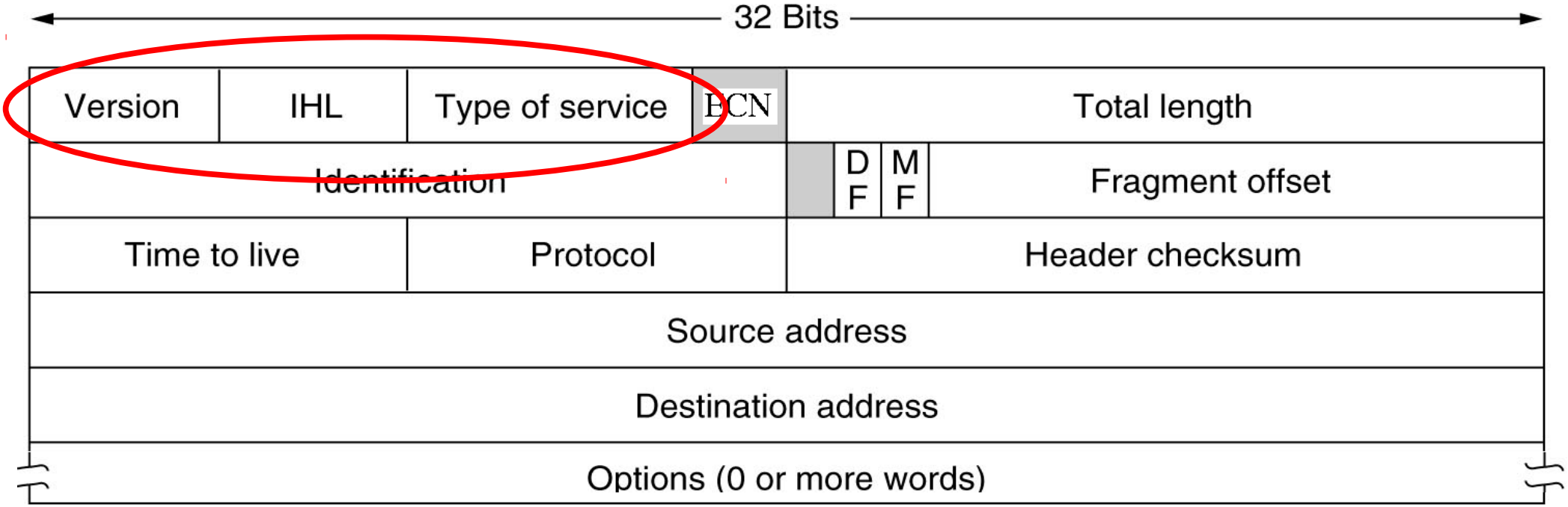
- Internetworking: world-wide connectivity
  - global routing of traffic between heterogeneous devices, subnets, operating systems, link later protocols, etc.
- Addressing: global location/identification of hosts
  - host ports are endued with unique IP addresses
- Routing & forwarding: between any(?) two hosts
  - **forwarding:** done one-by-one at routers, by looking up the destination address in the forwarding tables, to the next-hop IP address
  - **routing:** maintenance of the forwarding tables
- Other: fragmentation (IPv4: R2R, IPv6: E2E), etc.

IPv4 and IPv6

- Conventionally, IP as of version 4 uses 32-bit addresses
- Allows to connect roughly 4 billion hosts to the Internet
- Internet registries have long run out of new IP address ranges that can be handed out to hosts
- IPv6 introduced a 128-bit address space (enough to address any particle in the Universe)
- Transition to IPv6 is ongoing, use of IPv4 is still pervasive and is not expected to go away soon

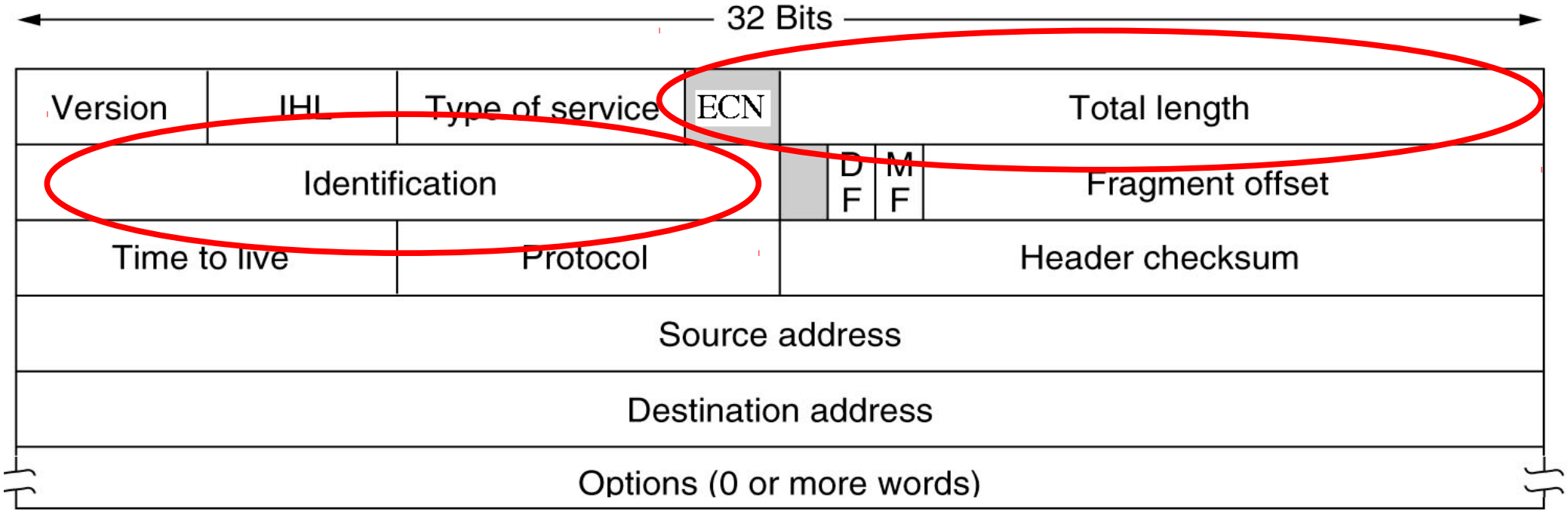
IPv4: Header

- Version (4 bit): always set to `0100` (binary 4)
- Internet Header Length (IHL, 4 bit): header length of 4 byte words (min 20, max 60 byte)
- Type of Service (ToS/DiffServ codepoint, 6 bit)



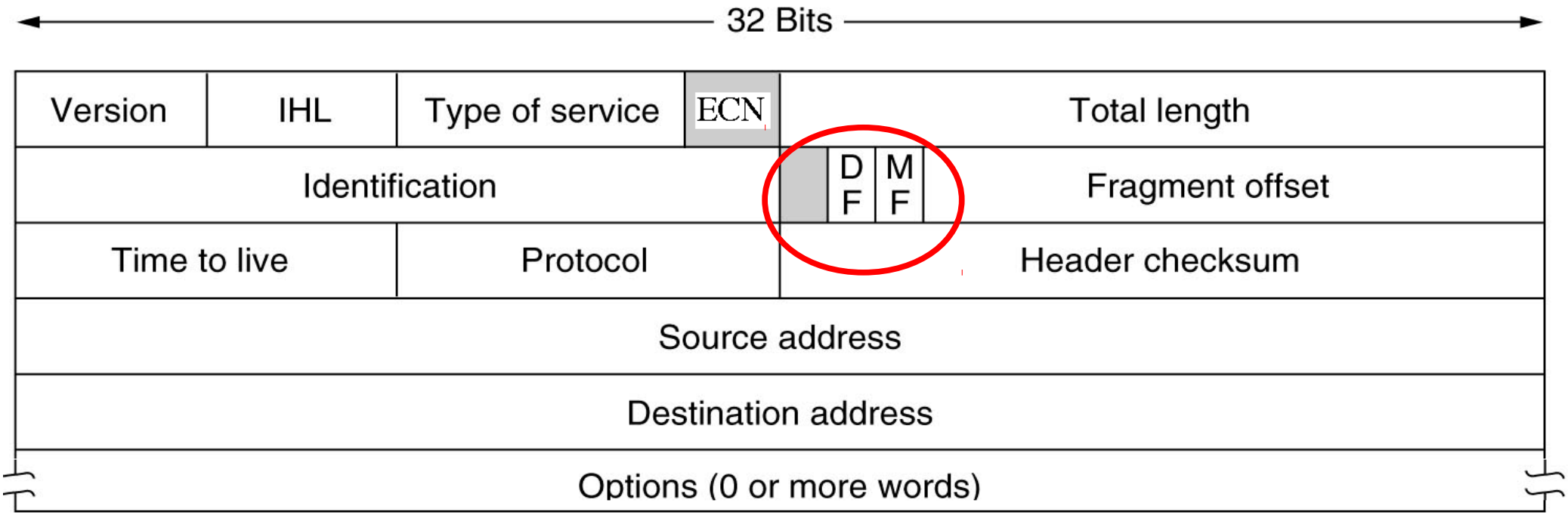
IPv4: Header

- Explicit Congestion Notification (ECN, 2 bits): congestion signal from network, rarely used
- Total length (16 bit): length of pkt in bytes
- Identification (16 bit): identify fragmented packets



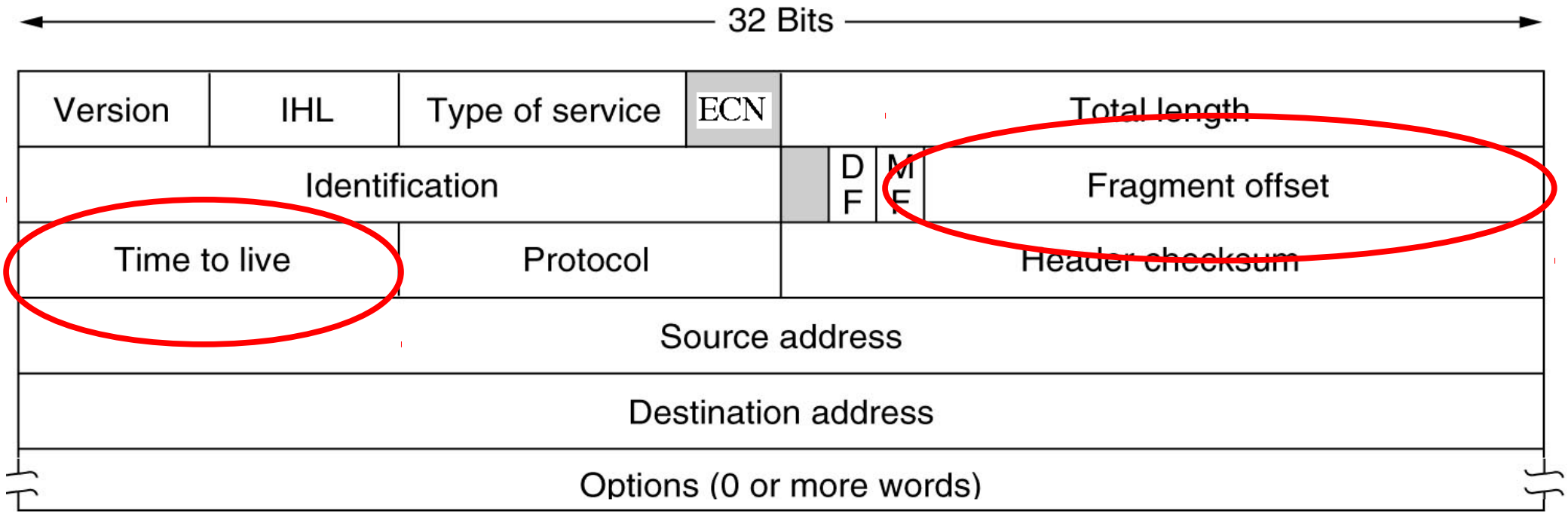
IPv4: Header flags

- bit 0: reserved, always 0
- bit 1: „Don't fragment” (DF)
- bit 2: „More fragments” (MF)



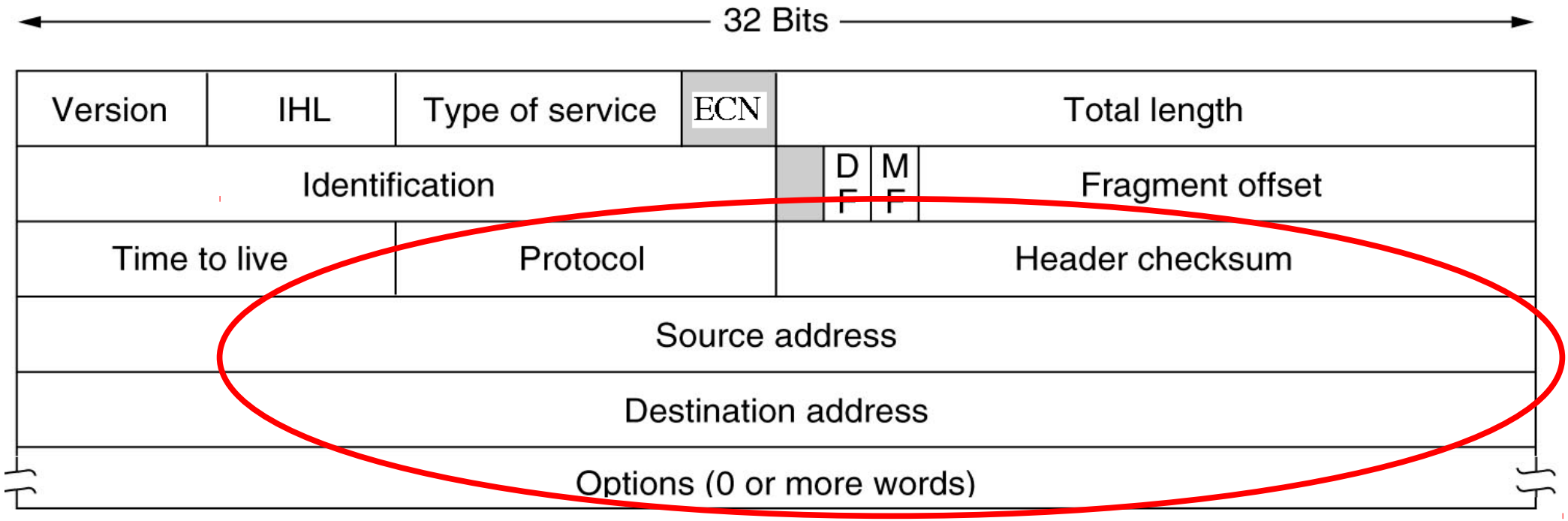
IPv4: Header

- Fragment offset (13 bit): offset into original payload for the actual fragment
- Time to live (TTL, 8 bit): routers decrement TTL and drop pkt when TTL reaches zero (avoid loops)



IPv4: Header

- Protocol (8 bit): id of the upper layer protocol the payload is to sent to (TCP, UDP, ICMP, ...)
- Header checksum (16 bit)
- Source and destination addresses (32 bit each)



IPv4: Adressing & subnets

- IPv4 address: 32 bit unsigned integer, in total 4294967296 (2^{32}) distinct addresses
- But the form `2554524783` is hard to remember
  - decimal notation: `2554524783`
  - binary: `10011000 01000010 11110100 01101111`
  - „dotted decimal”: broken to four 1-byte decimal numbers, separated by a dot: `152.66.244.111`

152	66	244	111
10011000	01000010	11110100	01101111
2554524783			

IPv4: Addressing & subnets

- “Classful” until 1993, since then “classless” (CIDR)
- Sets of consecutive IP addresses make up **subnets**
  - **physical subnet:** for hosts connected to the same link
  - **logical subnet:** created by the operator for routing purposes
- Subnets appear to the rest of the Internet as a single IP address prefix instead of separate per-host IP addresses
- **Aggregation:** one prefix for potentially thousands of hosts!
- The two parts of the IPv4 address:
  - first *X* bits (the **prefix**): **subnet identifier**
  - remaining *32-X* bits: **host identifier** (withing subnet)
  - *X* is called the **prefix length** (e.g., `/18`) or the **netmask** (in dotted decimal notation, e.g., `255.255.192.0` = `/18`)

IPv4: Classful addresses

- **Classful:** address communicates prefix length

Class	Address domain	Mask/CIDR	Example
Class A	`0*******/8` `0.0.0.0/8 – 127.0.0.0/8`	`255.0.0.0 (/8)`	`17.0.0.0/8` (Apple Inc.)
Class B	`10****** ********/16` `128.0.0.0/16 – 191.255.0.0/16`	`255.255.0.0 (/16)`	`152.66.0.0/16` (BMENET)
Class C	`110***** ******** ********/24` `192.0.0.0/24 – 223.255.0.0/24`	`255.255.255.0 (/24)`	`192.160.172.0/24` (SOTE)
Class D multicast	`1110*...` `224.0.0.0 – 239.255.255.255`	–	`224.0.0.5` (All OSPF Routers)
Class E foglalt	`1111*...` `240.0.0.0 – 255.255.255.255`	–	–

- Historical: `195.1.0.0/16` would be Class C, but used as Class B

IPv4: CIDR

- **Classless Inter-domain Routing:** any address range goes with any prefix length
- Variable Length Subnet Masking (VLSM)

CIDR notation	192.168.192.0/18
Prefix length	18 bit (counted from the Most Significant Bit (MSB))
binary	11000000 10101000 11000000 00000000
Subnet mask (binary)	11111111 11111111 11000000 00000000
Subnet mask (dotted)	255.255.192.0
Number of unique IP addresses in subnet	$2^{32-18}=2^{14}=16384$ (in fact, only $2^{14}-2$: first address is reserved for subnet id and last addess for subnet multicast address)
First IP address	192.168.192.1
binary	11000000 10101000 11000000 00000001
Last IP address	192.168.255.254
binary	11000000 10101000 11111111 11111110

IPv4: Masking

- Does address `192.168.199.100` belong to the subnet `192.168.192.0/18`?

Subnet	`192.168.192.0/18`
Binary	`11000000 10101000 11000000 00000000`
Subnet id 18 bit from the MSB	`11000000 10101000 11`
IP address	`192.168.199.100`
Binary	`11000000 10101000 11000111 01100100`
First 18 bit equals subnet id	`11000000 10101000 11`

- Subnet ids match (first 18 bits = **prefix**): yes, address belongs to the subnet
- Can be tricky!

Terminology

- Routing table = Routing Information Base (RIB)
  - unique for each running routing protocol
- Forwarding table = Forwarding Information Base (FIB)
  - by distilling multiple RIBs into a single FIB
  - consulted for each packet passing the router
- **Routing ≠ Forwarding**
  - **forwarding:** pass packet to the next-hop
  - **routing:** compute forwarding paths/tables and find next-hop (fill the FIB!)

IPv4 Forwarding

- **IP Router:** devices dedicated to IP-level packet processing and forwarding
  - passing packets between subnets: router ports connected to distinct subnets, each with unique IP
  - forwards packets based on the FIB
  - FIB maintenance: statically or dynamically, running a dedicated routing protocol for filling the FIBs
  - miscellaneous services: management (SNMP, CLI), monitoring (SNMP), misc. protocols (IGMP, CDP), access control, NAT, etc.

IPv4: Routing

1) Routing protocols (optionally more than one per router: OSPF + BGP) **exchange topology descriptors** between neighboring routers
2) Each routing protocol **sets up its own RIB** based on the information learned
3) Router **distills a FIB** by selecting the “best” next-hop to each subnet prefix from each RIB
4) Downloads the **(prefix → next-hop) pairs** to the FIB and constantly updates these assotiations whenever a topology change is learned

IPv4: Forwarding

1) Checks packet's validity (version, options, etc.)

2) Performs FIB lookup for each received IP packet

- based on the destination address in the IPv4 header
- **longest prefix match (LSB):** the smallest known (to the router) subnet that still contains the address

3) Handles packet (decrement TTL-t, set chksum)

4) Forward packet to the next-hop as found in FIB

- **Hop-by-hop routing:** routers know only the next-hop along the forwarding path, not the forwarding path itself (explicit routing)

Longest prefix match

- If multiple subnets match IP address in the FIB
- The **most specific match is preferred**: the subnet whose subnet id matches IP address on the most bits (counted from the MSB)
- Smallest subnet still containing the address
- **Longest Prefix Match (LPM):** key to IP!!
  - can be used to implement a lot of useful tricks
  - but makes IP packet forwarding pretty complex, as FIB lookup is nontrivial due to LPM

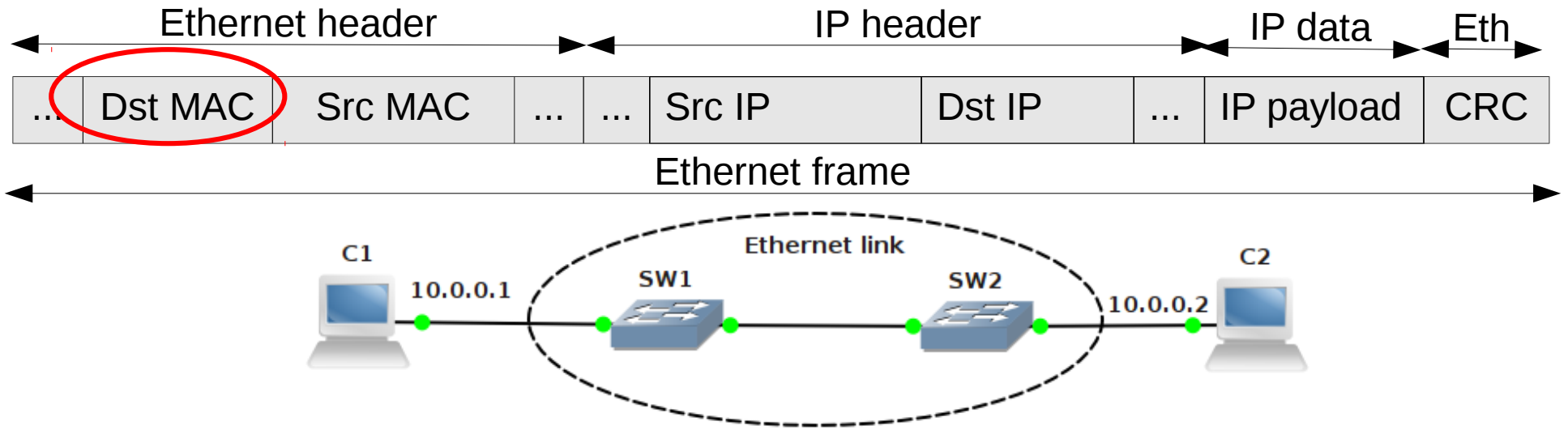
LPM: Example

Router FIB		
IP prefix/prefix len.	**Subnet identifier**	**Next-hop IP addr.**
192.168.0.0/16	11000000 10101000	10.0.0.1
192.168.0.0/17	11000000 10101000 0	10.0.0.2
192.168.64.0/18	11000000 10101000 01	10.0.0.3
192.168.96.0/19	11000000 10101000 011	10.0.0.4

- Address 192.168.1.1=x.x.00000001.000000001 matches first two entries, entries 3 and 4 differ on bit positions in marked red: entry 2 preferred
- Address 192.168.95.2=x.x.01011111.000000010 LPM is entry 3, for 192.168.97.3=x.x. 01100001.000000011 LPM is entry 4

IP over Ethernet

- IP usually runs in top of an Ethernet link layer



- Hosts `C1` and `C2` are on the same link: switches `SW1` and `SW2` transparent to these hosts
- **Address Resolution Protocol (ARP):** `C1` queries all hosts on the link, which Ethernet MAC address belongs to IP address `10.0.0.2`?

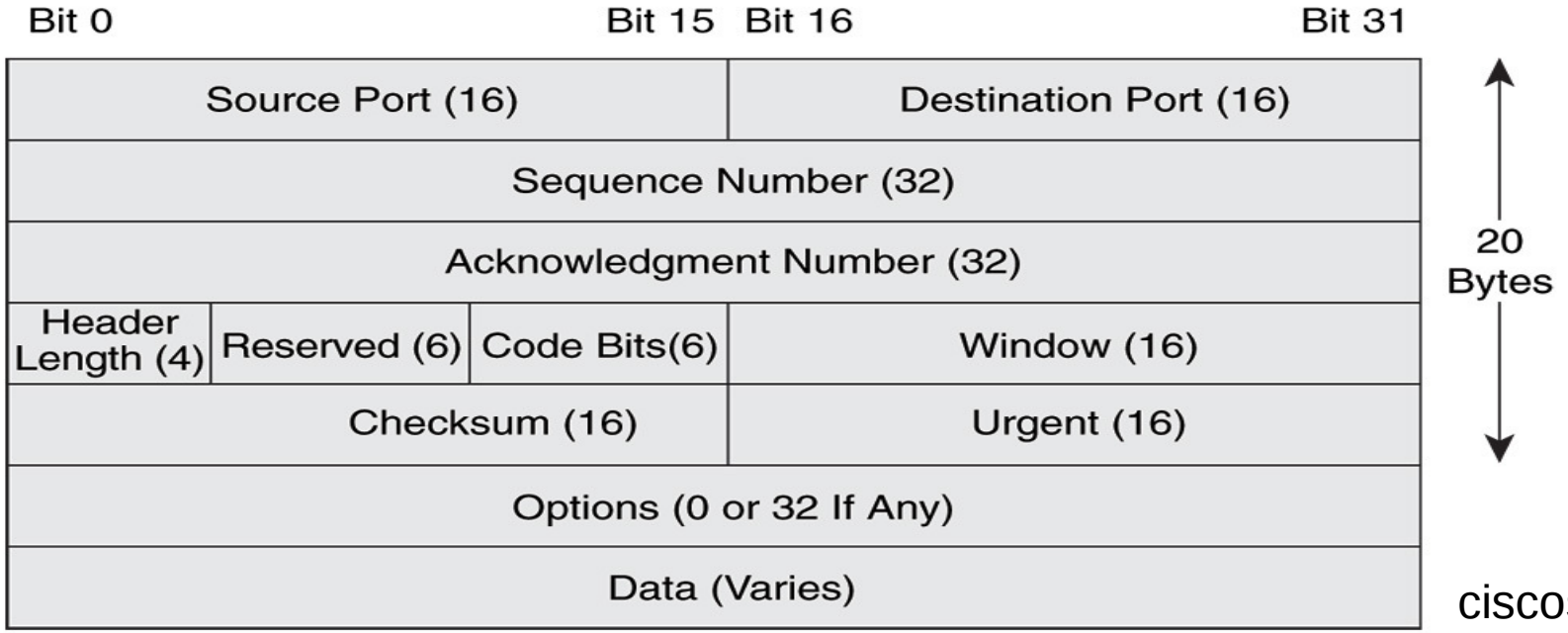
Transport layer

Transport layer

- For communication between specific users/ apps on the hosts: apps can originate/send traffic on a specific **port** (UDP/TCP port)
- TCP/IP: two popular transport protocols
- **Transmission Control Protocol (TCP):** connection-oriented, reliable stream transport between two unique TCP ports
- **User Datagram Protocol (UDP):** connectionless, non-reliable datagram service between UDP ports

Transmission Control Protocol

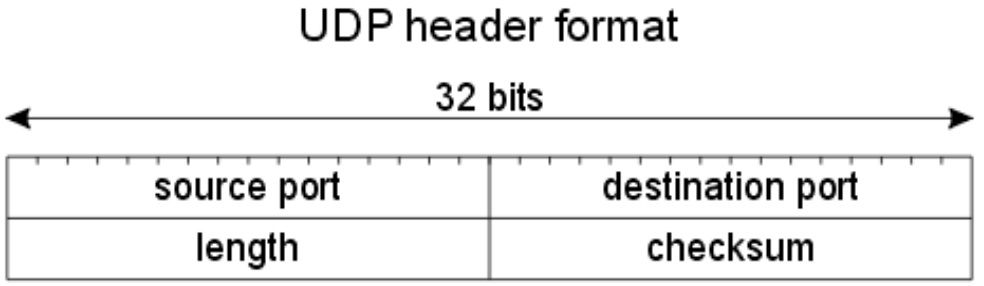
- Connection-oriented, reliable data stream
  - flow control (avoid flooding slow receiver with data)
  - congestion control by rate-control at sending host
  - multiplexing multiple connections to a single TCP connection (used extensively in HTTP, for instance)



ciscoskills.net

User Datagram Protocol

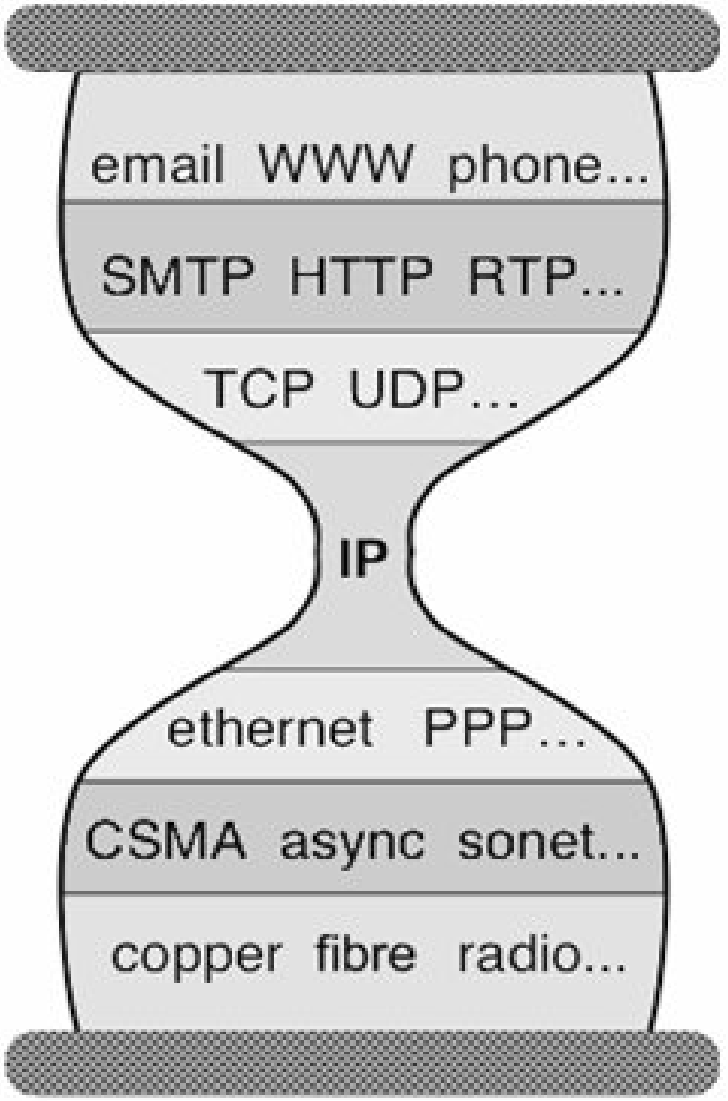
- Connectionless, datagram service
  - error detection (CRC)
  - but not reliable in the face of packet loss, packets may not arrive to receiver in the same order as sender sent, no protection to packet duplication
  - no connection setup (handshake, etc.)



wenk.be

IP „hourglass” model

- IP: largest common divisor
  - every packet passes the IP layer
  - every host speaks IP: true „internetworking”
- But next to impossible to alter/change/innovate:
  - IP multicast, IPv6,...
  - „internet ossification”



Trilogy project