Advanced Network Security

Network Layers

Thomas Martin

t.martin@mmu.ac.uk

March 6, 2019

Outline

Introduction

Networks are complex, with many components:

- hosts
- routers
- links
- applications
- protocols
- hardware
- software

The task of getting everything to work together is extremely difficult and complex. It requires some means of organising the structure of the network.

Layers

By breaking the problem down into a series of well-defined layers, we simplify the problem.

- Modularisation: it is easier to solve a number of small problems than one large problem.
- The layers are explicitly defined with specific purposes, and the interactions between them are limited (each layer relies on the layer below and provides services to the layer above).
- A change of implementation of one layer's service is transparent to the rest of system.
- This eases maintenance, updates (and teaching).

Definitions

Network: A set of computers and devices connected to each other for data exchange and resource sharing.

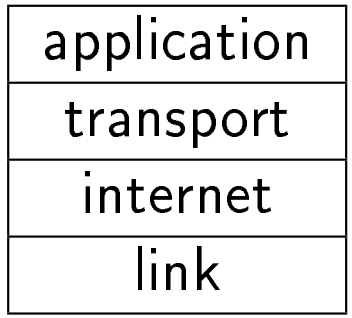
Nodes/Hosts/End systems: Providing services (servers) or receiving services (clients).

Intermediate systems: Forward received data to the intended recipients (e.g. switches/routers/hubs).

Protocol: A set of rules that governs communications between nodes.

Internet Protocol Stack

- Application: supporting network applications, e.g. FTP, SMTP, HTTP
- Transport: process-process data transfer, e.g. TCP, UDP
- Internet (or Network): routing of datagrams from source to destination IP, routing protocols
- Link: data transfer between neighbouring network elements, e.g. ethernet, 802.111 (WiFi), PPP

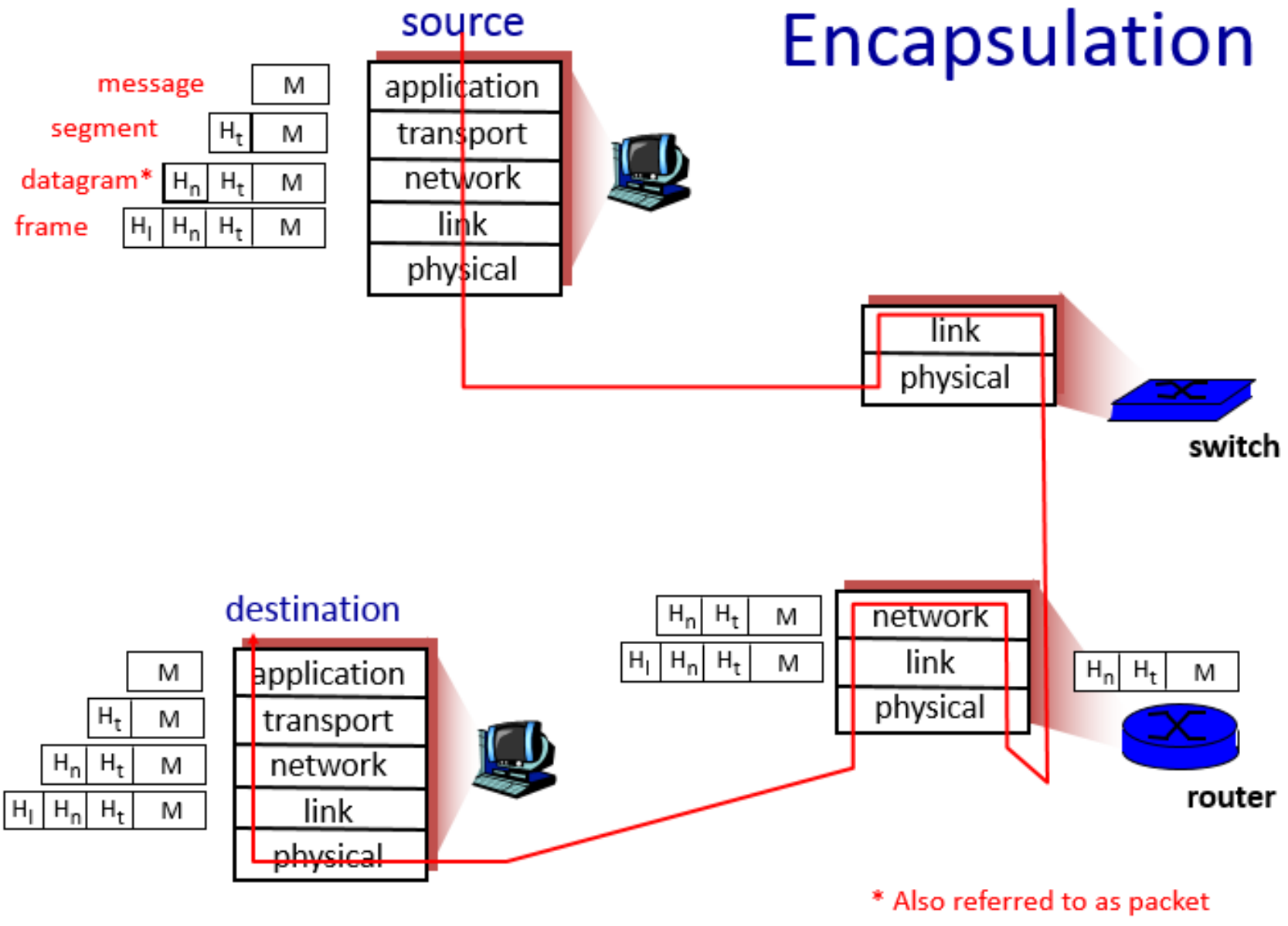


ISO/OSI reference model

- Presentation: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions.
- Session: synchronization, checkpointing, recovery of data exchange.
- Internet stack "missing" these layers! These services, if needed, must be implemented in application.
- Physical: bits "on the wire"

application
presentation
session
transport
network
data link
physical

Encapsulation



Link Layer

The link layer has responsibility for transferring datagrams from one node to a *physically adjacent* node over a link.

It is possible to use a number of different media on the Internet, and the different links may offer different services. **Error detection/-correction** may or may not be available. Using CRC's can help the receiver identify any modified bits due to interference or weak signal, and possibly even correct without needing re-transmission.

A related concept is **reliable transfer**, where the sender gets assurance that sent data has been received.

Link Layer - Multiple Access

Most links require **Multiple Access Protocols**. If two or more nodes transmitting simultaneously will cause interference (sent data not received), then a protocol is required to avoid collisions.

- Channel partitioning: divide the channel into smaller "pieces" (based on time slots or available frequencies), and allocate pieces to nodes for exclusive use.
- Random Access: allow collisions but provide a mechanism for recovery.
- Taking turns: controlled by either a central node or a token passed among nodes.

Link Layer - Addressing

Addressing at the link layer is performed with a 48-bit MAC[1] address. The function of the address is to get a frame from one interface to another physically-connected interface (in the same network). MAC addresses are generally burned into the NIC ROM, but can sometimes be set by software.

MAC address allocation is administered by the IEEE. A manufacturer buys a portion of the MAC address space (to ensure uniqueness). Generally, this determines the first 24 bits, and they are then free to use the remaining 24 bits to index their devices. MAC addresses are flat and portable (unlike IP which is hierarchical and depends on the IP subnet the node is attached to).

[1] Media Access Control

Link Layer - ARP

How can a node determine the MAC address of another node if they only know the IP address? Each IP node (e.g. host, router) on a LAN has an ARP table. This table contains IP-MAC address mappings for LAN nodes (along with a time-to-live, typically 20 minutes). The ARP protocol works as follows:

- A wants to send a datagram to B, and B's MAC address is not in A's ARP table.
- A broadcasts an ARP query packet, containing B's IP address. The destination MAC address is set to `FF-FF-FF-FF-FF-FF`, and all machines on the LAN will receive.
- B receives the ARP packet, and replies to A with its MAC address (unicast back to A).
- A can cache the MAC address.

Internet (Network) Layer

The network layer is responsible for delivering datagrams from the sending host to the receiving host. The network layer is understood and processed by every host and router. Each router needs to examine the header fields of all IP datagrams that pass through it. The two key functions of the network layer are:

- *Routing:* determine the route taken by datagrams from the source to the destination.
- *Forwarding:* move datagrams from the router's input to the appropriate output.

Internet (Network) Layer Services

Some network links have limits on the size of packets, and these limits will not be known to the original sender in advance. To cope with this, fragmentation can be performed on datagrams, with reassembly at the final destination.

Addresses at the network layer are 32-bit, and are identifiers for the host or router interface. IP addresses are written in the form: `149.170.133.128`. The address can be combined with a mask (e.g. `255.255.255.0`) where the high order bits identify the subnet (`149.170.133.0`), and the low order bits identify the host (128). The same can be specified with CIDR notation (/24).

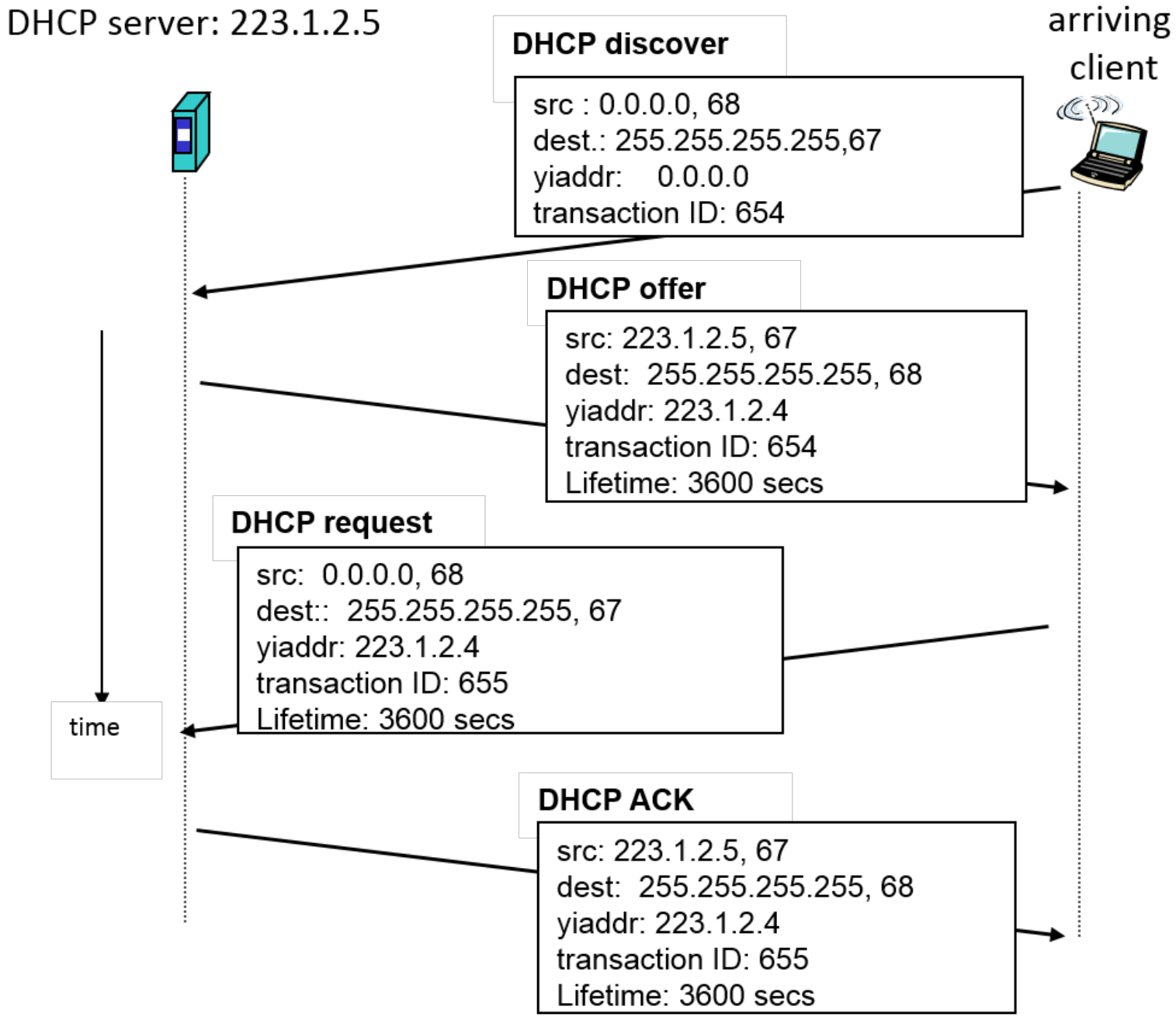
Internet (Network) Layer Addressing

IP addresses can be manually configured, but this is time-consuming. Instead the Dynamic Host Configuration Protocol (DHCP) can be used to request[2] an IP address from a server when joining a new network.

DHCP can return more than just an allocated IP address on the subnet:

- Address of first-hop router for client
- Name and IP address of DNS server
- Network mask (indicating network versus host portion of address)

[2] DHCP uses UDP segments


DHCP server: 223.1.2.5
arriving client
DHCP discover
src : 0.0.0.0, 68
dest.: 255.255.255.255,67
yiaddr: 0.0.0.0
transaction ID: 654
DHCP offer
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 654
Lifetime: 3600 secs
DHCP request
src: 0.0.0.0, 68
dest:: 255.255.255.255, 67
yiaddr: 223.1.2.4
transaction ID: 655
Lifetime: 3600 secs
time
DHCP ACK
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 655
Lifetime: 3600 secs

Internet (Network) Layer: NAT

Since addresses use 32-bits, that limits the Internet to 4 billion addresses. As they became more scarce, Network Address Translation (NAT) became a popular way to allow multiple private addresses (`10.0.0.0/8` or `192.168.0.0/16`) to connect from a single public address. The NAT router keeps a table of all connections, changing the source IP address to ensure replies get back and manipulates the source port number to ensure uniqueness.

NAT provides a number of security benefits, but breaks the original intention of the Internet. Address space exhaustion is solved in IPv6, along with a number of other improvements.

Internet (Network) Layer: ICMP

The Internet Control Message Protocol (ICMP) is used by hosts and routers to communicate network-level information, such as error reporting (host/network/port/protocol), echo request/replies.

The path packets take is not known by the sender. One way to discover is with Traceroute. Traceroute sends a series of UDP segments to the destination with TTL starting at 1 and increasing. TTL decreases by 1 each hop, and when it reaches 0 results in an ICMP error. The sources of these error indicate the path the packets take in the network.

Internet (Network) Layer: Routing

The most important task at the network layer is getting the packets to where their destinations. This is achieved with two separate processes: Routing and Forwarding.

Routing: Taking a view of the entire network from the point of view of one router and determining the least cost path to each possible destination. The results are saved in the *Forwarding Table*.

Forwarding: On receipt of a datagram by a router, comparing the destination with the forwarding table to determine which of the outputs should be used to pass the packet onto the next hop.

Internet (Network) Layer: Routing

There are two general approaches to finding the least cost path in a network: Link State (Dijkstra's algorithm) and Distance Vector (Bellman-Ford algorithm). Dijkstra has a high communication cost as each node needs to the costs of every link in the network. Bellman-Ford takes longer to converge the more nodes there are in the network

To simplify routing, destinations are aggregated. Prefixes are essentially IP addresses written in binary with a certain number of least-significant bits wild-cards. The ease with which these can be combined greatly reduces the number of entries needed in the table from 2^{32}. The down side is all subnets sizes must be a power of 2. The other fix is dividing the Internet into Autonomous Systems, with different systems used for inter-AS and intra-AS routing.

Transport Layer

Where the network layer provides host-to-host communication, the transport layer provides process-to-process communication. The main service is to make sure different segments are not mixed up. Port numbers (in the range 0 . . . 65535) facilitate the multiplexing/demultiplexing.

There are two important protocols at this layer:

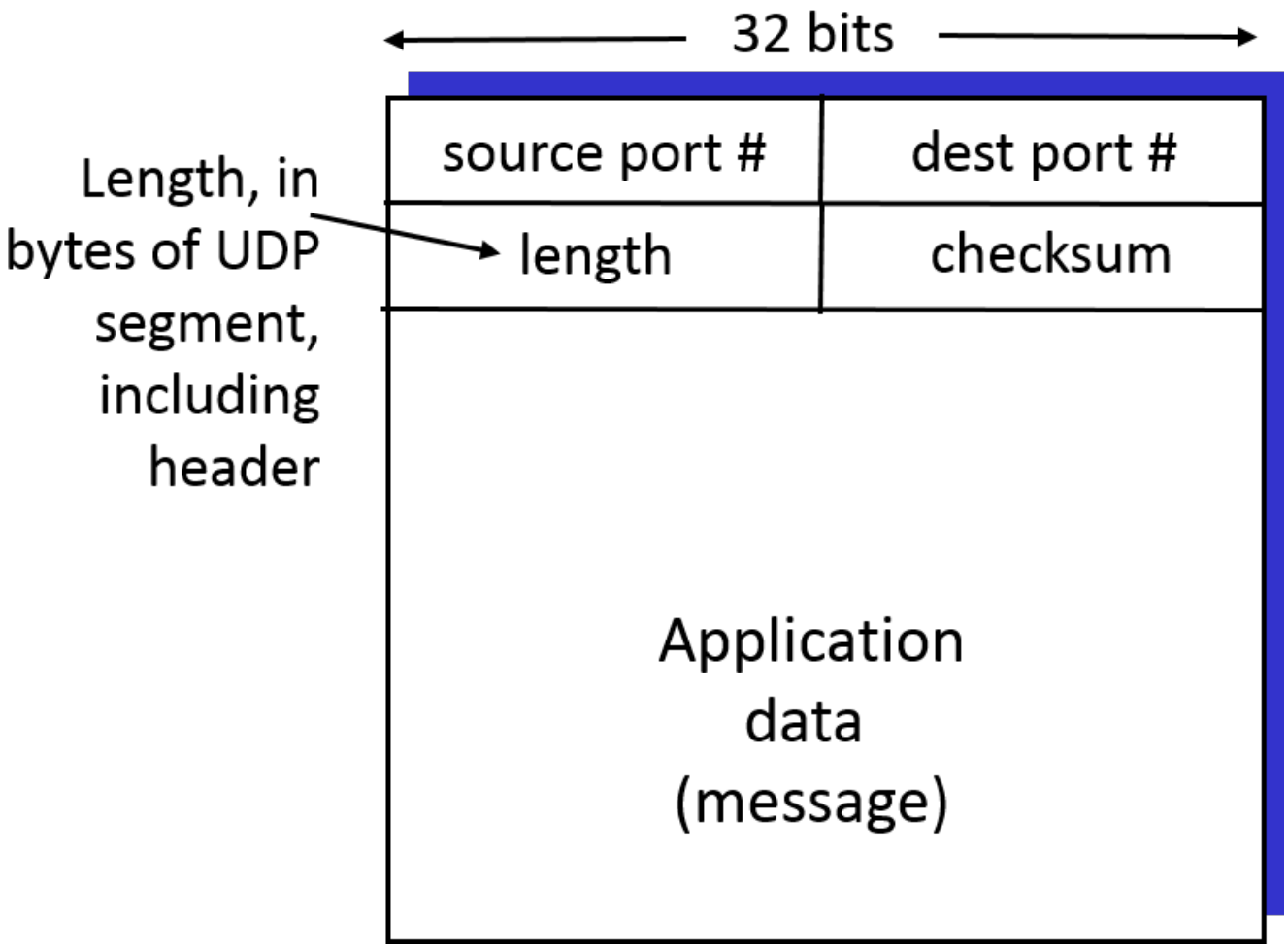
- TCP - Transmission Control Protocol
- UDP - User Datagram Protocol

Transport Layer: UDP

UDP provides a "best effort" service. Segments may be lost or delivered out of order to the application layer. It is entirely stateless, connectionless, there is no handshaking setup, no congestion/flow control. All this makes it very simple, but unreliable.

UDP is often used for streaming multimedia applications. Voice and video conferencing can tolerate some loss of segments if it means lower latency. Other uses for UDP include DNS and SNMP.

Transport Layer: UDP Segment Structure

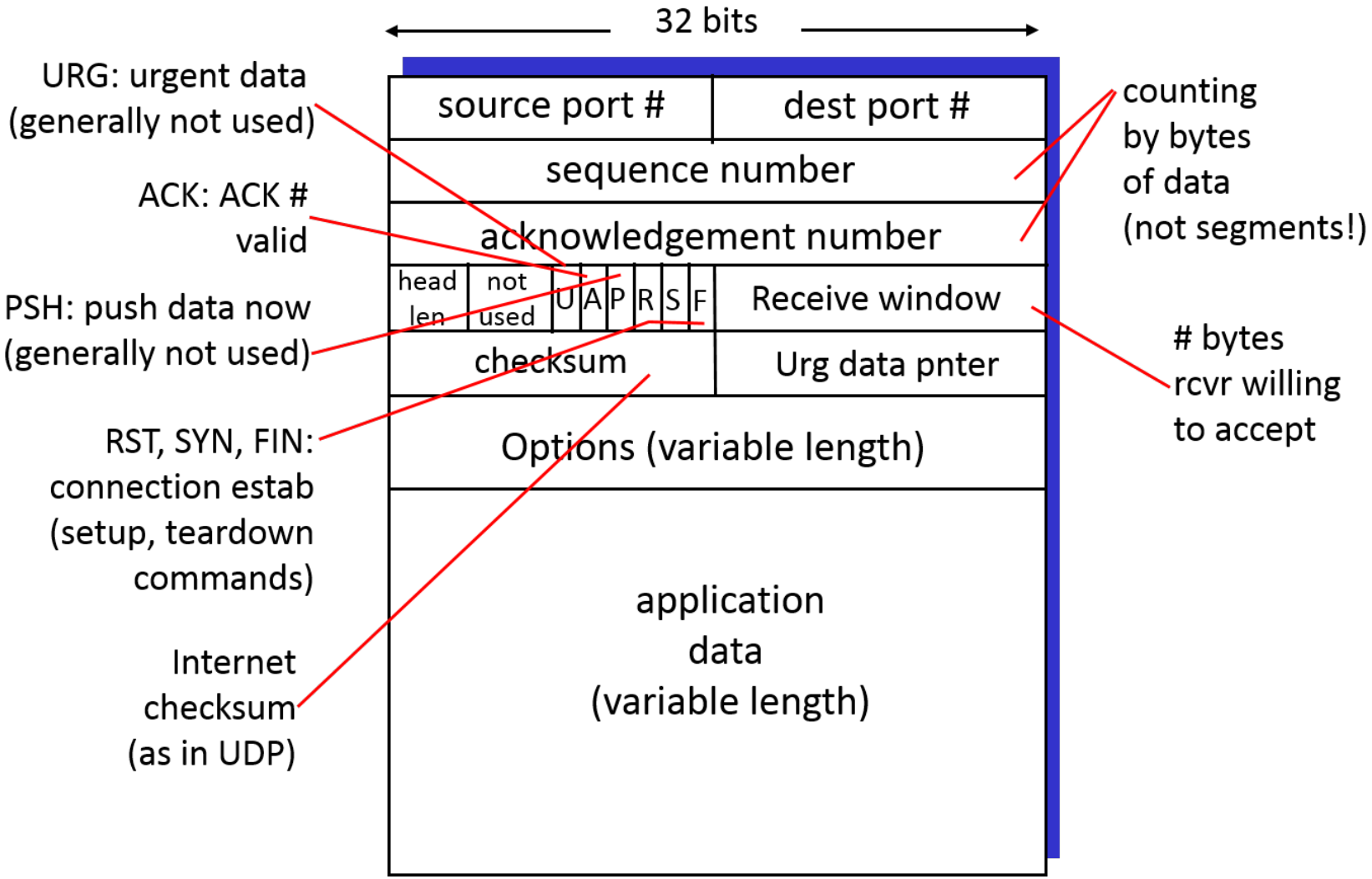


Transport Layer: TCP

TCP provides:

- Reliable, in-order byte stream
- Pipe-lined: multiple segments are sent in parallel
- Uses buffers at both sender and receiver end
- Full duplex data: bi-directional flow of data
- Connection-oriented: handshake (SYN - SYN/ACK - ACK) before data exchange, and close on completion (FIN - ACK x2)
- Flow controlled: sender will not overwhelm receiver
- Congestion controlled: traffic responds to delays in the network

Transport Layer: TCP Segment Structure



Transport Layer: TCP Reliability

To manage the reliable delivery of data, TCP uses sequence and acknowledgement numbers in every segment. These numbers relate to byte streams going in both directions.

- **Sequence number:** Byte stream "number" of first byte in this segment's data
- **Acknowledgement number:** Sequence number of the next byte expected from other side (a cumulative acknowledgement)

Each sender keeps a timer, and segments will be resent on either a timeout or duplicate ACKs.

Application Layer

There are three main possibilities for an application architecture:

- Client/server
- Peer-to-peer
- Hybrid of client/server and p2p.

Client/server is the most common of the three.
Server: Is always on, and has a permanent IP address.
Clients: Communicate with the server. They may be intermittently connected, may have dynamic IP addresses, and do not communicate with each other.

Application Layer

A **process** is a program running within a host. Within the same host, two processes communicate using inter-process communication (which is managed by the OS). Processes in different hosts communicate by exchanging messages.

- *Client process:* Process that initiates communication
- *Server process:* Process that waits to be contacted

An application protocol facilitates the exchange of these messages. Each protocol must use either TCP or UDP, depending on their needs.

Application Layer

An application layer protocol defines:

- Types of messages exchanged, e.g. request, response
- Message syntax, what fields are in messages and how fields are formatted
- Message semantics, meaning of information in fields
- Rules for when and how processes send and respond to messages

There are public-domain protocols, usually defined in RFCs. This simplifies interoperability. Examples include HTTP, FTP, SMTP, SSH.

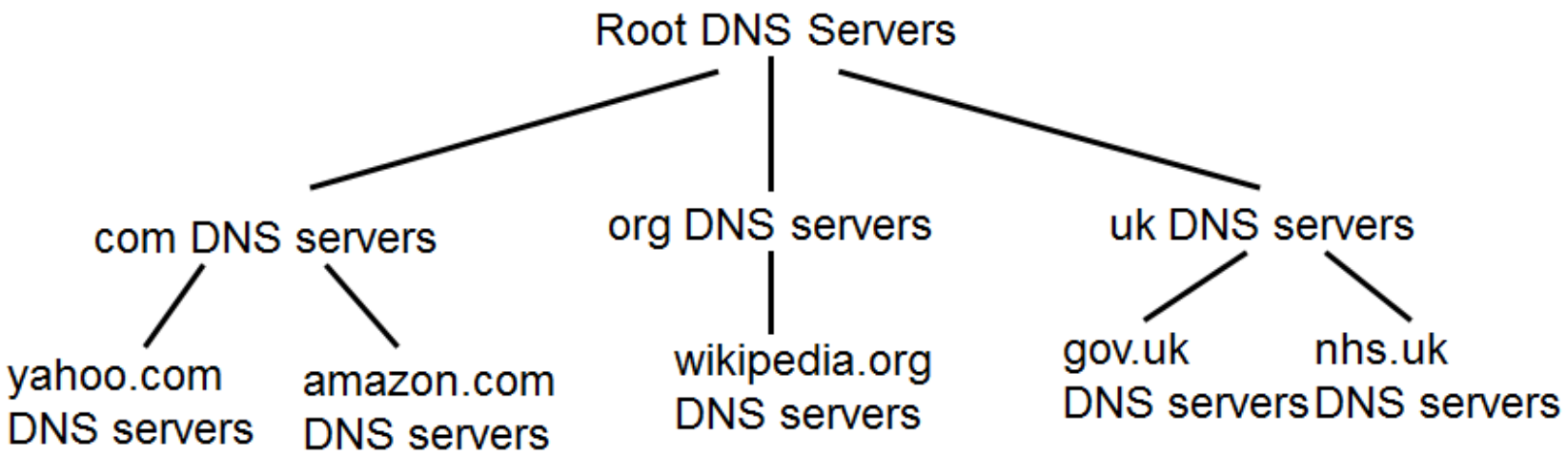
There are also proprietary protocols that commercial companies develop and use, such as Skype.

DNS

DNS (Domain Name System) is an application layer protocol. But it is an unusual one because it is necessary for the correct operation of the network layer. The network layer uses IP addresses, which have necessary properties of being fixed size with a hierarchical structure. But they are not user-friendly, i.e. easily remembered.

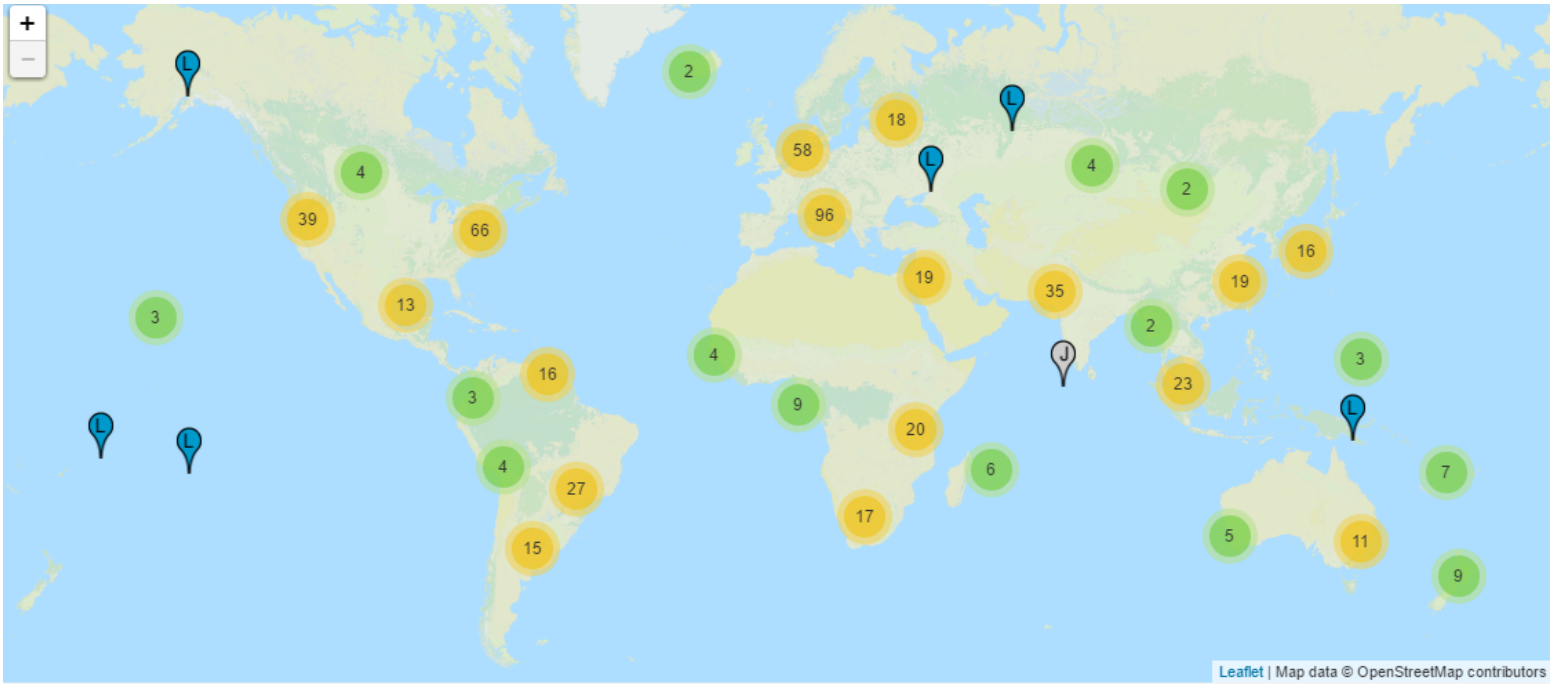
DNS provides a way to resolve domain name to IP address (and vice versa) translation. It exists as a distributed database implemented in a hierarchy of many name servers. As well as name translation, it can also be used for aliasing and load distribution.

DNS Hierarchy



A centralized name system would have many problems: single point of failure, massive traffic volume, latency for many users, inability to take down for maintenance. Distributing DNS takes care of many of these issues. Having a hierarchy allows for efficient resolution of queries.

DNS Root Servers



The root servers are named *letter*.root-servers.net (a-m), but as you can see from above, there are far more than 13 actual servers[3].

[3] `http://www.root-servers.org/`

TLD and Authoritative Servers

Top-level domain (TLD) servers are responsible for com, org, net, edu, jobs, etc. and all top-level country domains, e.g. uk, ie, ae, fr, ca, jp, etc. Root servers and TLD servers do not primarily perform name translation mapping. Their function is to redirect the query to name server for the next subdomain (domain names get processed from right to left) until eventually it gets sent to the Authoritative server.

An organization's authoritative DNS server provides the definitive hostname to IP mappings for their servers. It can be maintained by the organization or outsourced to a service provider. There can be many levels of sub-domains between the TLD and the authoritative servers, not just as simple as `amazon.com` but `moodle.mmu.ac.uk` or `www.aerospace.manchester.ac.uk`.

DNS Records

DNS stores and responds to queries with a number of different resource records (RR).

`RR format: (name, value, type, ttl)`

- `Type=A` - name is hostname, value is IP address
- `Type=NS` - name is domain, value is the hostname of the authoritative name server for this domain
- `Type=CNAME` - name is alias name for some "canonical" name, value is the canonical (real) name
- `Type=MX` - value is the name of the mailserver associated with name

DNS queries and responses use UDP over port 53.

Thank you

Relevant chapter: Erickson, Hacking: The Art of Exploitation, Chapter 4.

Any Questions?