Internet Protocols — Part 1
Low Level Communication Protocols

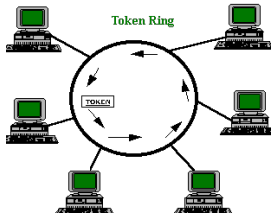
Token Ring Protocol

Lets look at a fairly simple protocol before we progress to real Internet ones.

- Simple Local Area Network Protocol
- Avoids complexities of Internetwork Networking

Token Ring Protocol - Simple Local Area Network Protocol

- Token Ring is a Local Area Network (LAN) protocol.
- First developed by IBM (1970s) and standardised 1985.
- A first and second layer protocol in the OSI 7 layer model.
- First release of Token Ring version was capable of 4Mbs data transmission rate,
  - improved later to 16Mbs.
- Token Ring operates on many cable types.
- **The protocol deals with the problem of collision**:
  - **Collision**— a state were two stations transmit at the same time.



Collision Avoidance

To avoid collision:

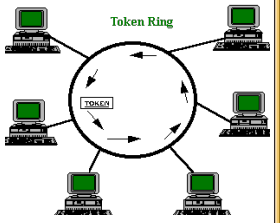
- need to control the access to the network.

How to Control?

- Need to use a control (permission) structure called **Token**.
- Token passed to stations via a set of rules (protocol).

Token Ting Configuration and Transmission

- Ring consists of
  - ring stations and
  - transmission medium.
- Data travels sequentially from station to station.
- Only the station in possession of the token is allowed to transmit data.
- Each station then :
  - repeats the data,
  - checks for errors, and
  - copies the data if appropriate.
- When the data is returned to the sending station, it removes it from the ring.



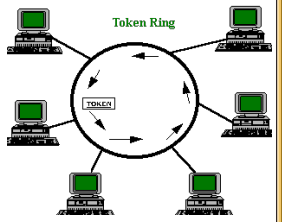
Token Ring Advantages

- High reliability, the Ring can continue normal operation despite any single fault.
- Bypassing inactive stations.
- Excellent traffic handling (17.8 kb in TR, only 15kb in Ethernet.).
- Large maximum frame length .
- High bandwidth efficiency. 70% in Token Ring, 30-40% in Ethernet.
  - But free from more complex network operations
  - So direct comparison not always valid?
- Many media choices: UTP STP coax fibre.
- Supports transmission priority.

Token Ring Mechanism

Whenever a station wishes to send a frame:

- It first waits for the token.
- As soon as it receives the token,
  - it initiates transmission of the frame,
  - New transmission includes the destination station address at its head.
  - The frame is repeated (received and retransmitted) by each station on the network until it circulates back to the source station, where it is removed .



Early Token Release mechanism (ETR)

A station releases the token in one of the two ways depending on the ring rate:

- With slower rings(4Mbps), the token is released only after the response bits have been received.
- With higher speed speed rings (16Mbps), it is released after transmitting the last bit of the frame.
- This is known as **Early Token Release mechanism (ETR)**

Early Token Release mechanism (ETR):

- enables multiple frames on the ring,
- makes the ring is more effective.
- When working in a large ring it improves performance,
  - enabling a mixture of stations with ETR and stations without ETR.

Token Ring Media

Token ring is a logical ring topology, but can physically implemented as :

- Ring
- Bus
- Star
- Token Ring can be operated on the following media:
  - Unshielded Twisted Pair (UTP).
  - Shielded Twisted Pair (STP): Allowing a Max. of 260 stations at 16Mps rings.
  - Coaxial cable (Thin/Thick/Broadband).
  - Fiber Optics.

TCP/IP — The Internet Protocol Suite

TCP/IP:

- (Recall) stands for **Transmission Control Protocol/Internet Protocol**.
- The Protocol upon which the whole Internet is based
  - Each node must be configured for TCP/IP to function properly.
- A software-based protocol

What is TCP/IP?

- TCP/IP is basically the binding together of Internet Protocols used to connect hosts on the internet- Main ones are IP and TCP
- TCP and IP have special packet structure (see next)
- IP (Internet Protocol) is responsible for delivering packets of data between systems on the internet + specifies their format. Packets forwarded based on a four byte destination IP address (IP number) . More on IP numbers later.
- IP DOES NOT MAKE GUARANTEES! It is very simple - essentially: *send and forget*.
- TCP (Transmission Control Protocol) is responsible for verifying the correct delivery of data/packets from client to server. Data can be lost - so TCP also adds support to detect errors + retransmit data until completely received.
- Together these help form TCP/IP - a means of specifying packets, and delivering them safely.

- There are other protocols in TCP/IP - such as User Datagram Protocol UDP.
- UDP is a simpler alternative to TCP for aiding the delivery of packets. It makes no guarantees regarding delivery - but does guarantee *data integrity*
- UDP also has no flow control, i.e. if messages are sent too quickly data may be lost.

IP Packet Structure (1)

IP uses a *Datagram* to transfer *packets* between *end systems* (usually computers) using *routers*. There are fourteen fields in an IP Packet (Network Level 3):

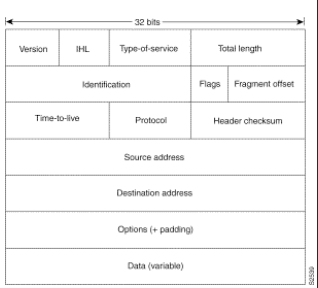
Version — Indicates the version of IP currently used.

IP Header Length (IHL) — Indicates the datagram header length in 32-bit words.

Type-of-Service — Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.

Total Length — Specifies the length, in bytes, of the entire IP packet, including the data and header.

Identification — Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

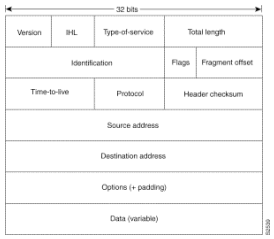


IP Packet Structure (2)

Flags — Consists of a 3-bit field of which the two low-order (least-significant) bits control fragmentation. The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

Fragment Offset — Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

Time-to-Live — Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.



IP Packet Structure (3)

Protocol — Indicates which upper-layer protocol (More Later) receives incoming packets after IP processing is complete.

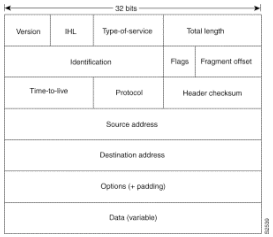
Header Checksum — Helps ensure IP header integrity.

Source Address — Specifies the sending node.

Destination Address — Specifies the receiving node.

Options — Allows IP to support various options, such as security.

Data — Contains upper-layer sent in packet.



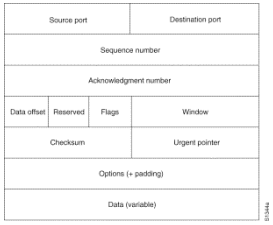
TCP Packet Structure(1)

There are 12 fields in TCP Packet (Transfer Level 4):

Source Port and **Destination Port** — Identifies points at which upper-layer source and destination processes receive TCP services.

Sequence Number — Usually specifies the number assigned to the first byte of data in the current message. In the connection-establishment phase, this field also can be used to identify an initial sequence number to be used in an upcoming transmission.

Acknowledgment Number — Contains the sequence number of the next byte of data the sender of the packet expects to receive.



TCP Packet Structure(2)

Data Offset — Indicates the number of 32-bit words in the TCP header.

Reserved — Remains reserved for future use.

Flags — Carries a variety of control information, including the SYN and ACK bits used for connection establishment, and the FIN bit used for connection termination.

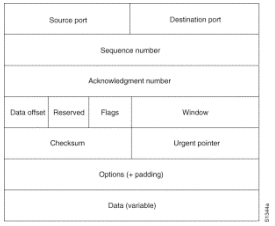
Window — Specifies the size of the sender's receive window (that is, the buffer space available for incoming data).

Checksum — Indicates whether the header was damaged in transit.

Urgent Pointer — Points to the first urgent data byte in the packet.

Options — Specifies various TCP options.

Data — Contains upper-layer sent in packet.



UDP Packet Structure

Source and **destination ports** – contain the 16-bit UDP protocol port numbers used to demultiplex datagrams for receiving application-layer processes.

Length field — specifies the length of the UDP header and data.

Checksum – provides an (optional) integrity check on the UDP header and data.

Layered Protocols — Encapsulation

Data from higher layers is **encapsulated** in lower layers (demonstrated in a moment!)

Layered protocol models rely on encapsulation:

- allows one protocol to be used for relaying another's messages.
- refers to the practice of enclosing data using one protocol within messages of another protocol.
- The encapsulating protocol must be open-ended,
  - Allowing for arbitrary data to be placed in its messages.
  - Another protocol can then be used to define the format of that data.

Encapsulation Example

Consider an Internet host that requests a hypertext page over a dialup serial connection.
The following scenario is likely:

1. HyperText Transfer Protocol (HTTP) is used to construct a message requesting the page. The message, (exact format not relevant here), is represented as follows:

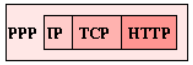
   HTTP

2. Transmission Control Protocol (TCP) is used to provide the connection management and reliable delivery that HTTP requires, but does not provide itself. TCP defines a message header format, which can be followed by arbitrary data. So, a TCP message is constructed by attaching a TCP header to the HTTP message, as follows:

   TCP | HTTP

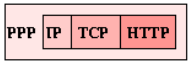
3. TCP does not provide any facilities for actually relaying a message from one machine to another in order to reach its destination. This feature is provided by the Internet Protocol (IP), which defines its own message header format. An IP message is constructed by attaching an IP header to the combined TCP/HTTP message:

4. Although IP can direct messages between machines, it can not actually transmit the message from one machine to the next. This function is dependent on the actual communications hardware. In this example, we're using a dialup modem connection, so it's likely that the first step in transmitting the message will involve the Point-to-Point Protocol (PPP):



Encapsulation Example Notes

Note: the PPP encapsulation drawn a little differently, by enclosing the entire message, not just attaching a header.



- This is because PPP may modify the message if it includes bytes that can't be transmitted across the link.
- The receiving PPP reverses these changes, and the message emerges intact.
- The point to remember is that the **encapsulating protocol can do anything it wants to the message**:
  - expand it,
  - encrypt it,
  - compress it

  so long as the original message is extracted at the other end.

PPP and PPPoE

Point-to-Point Protocol (PPP) is the Internet Standard for transmission and encapsulation of IP packets over serial lines (or point to point links).

Common ISPs (NTL,BT,etc) now typically use PPPoE - Point-to-Point Protocol over Ethernet

PPPoE merges PPP and Ethernet standards

This allows computers connected on a LAN (ethernet based) to connect to the internet through a broadband medium such as DSL (Digital Subscriber Line).

PPP supersedes SLIP (Serial Line Internet Protocol)

- faster
- more reliable
- supports error detection dynamic assignment of IP addresses
- data compression.

Routing

When one network is connected to another, a device called a router connects both networks and passes data between them.

- In a simple ring network a packet may be routed around the ring until it gets "caught" or gets back to the sender.
- The Internet is not that simple — multiple topologies.
- A router can be connected to more than one network.

- The route selected depends on traffic loads, what backbones are working *etc.*. **Not all** packets may be routed over the same route.

Some Routing Methods

- Routing Tables — specific maps (a list of routes) on how to get somewhere. Try one route from list until you succeed
- Centralised point — send all traffic through a centralised node in a network.
- Nearest Neighbour (Centralised adaptive routing) — a central node within each network knows only about its direct connections to the outside world. Send to nearest connection.

Sample Traceroute

```
traceroute to www.netscape.com (198.95.251.30), 30 hops max, 40 byte packets
 1  cr1.cf.ac.uk (131.251.1.42) 2.382 ms 1.868 ms 1.952 ms
 2  smds-gw.ulcc.ja.net (193.63.203.33) 8.456 ms 7.820 ms 8.711 ms
 3  193.63.94.8 (193.63.94.8) 12.064 ms 9.730 ms 15.122 ms
 4  icm-dc-1-S3/2-1984k.icp.net (192.157.65.113) 172.080 ms 168.902
    ms 175.264 ms
 5  icm-mae-e-H1/0-T3.icp.net (198.67.131.9) 162.964 ms *
 6  192.41.177.180 (192.41.177.180) 167.341 ms * 156.772 ms
 7  borderx2-hssi2-0.Washington.mci.net (204.70.74.117) 166.732 ms
 8  core-fddi-1.Washington.mci.net (204.70.3.1) 246.142 ms 320.413
    ms 301.374 ms
 9  core1-hssi-4.LosAngeles.mci.net (204.70.1.177) 234.920 ms
 10 core-hssi-2.SanFrancisco.mci.net (204.70.1.153) 495.669 ms *
 11 border2-fddi0-0.SanFrancisco.mci.net (204.70.3.162) 383.403
    ms
 12 netscape.SanFrancisco.mci.net (204.70.33.10) 250.367 ms * *
 13 www1.netscape.com (198.95.251.30) 243.961 ms
```

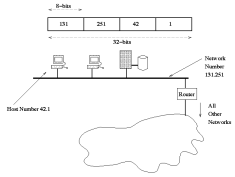
Note that on different runs of this at different times the routes will vary.

You can trace your own route by using the Mac OS X Application **Network Utilities** (Application/Utilities sub-folder)

IP Adresses

Every Computer Must have a unique IP address to be connected to a network — *a bit like a telephone number*

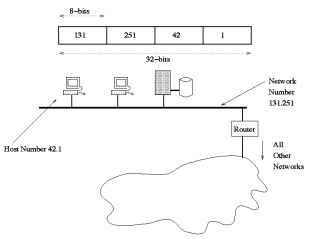
- An IP address is 32 bits (4 Bytes) wide,
- it is composed of two parts:
  - the network number,
  - the host number



IP Address Format

By convention IP Address is

- expressed as four decimal numbers separated by periods, such as `131.251.42.1`
- one decimal value of each of the four bytes.
  - First two numbers specify Network
  - Last two numbers specify Host
- Valid addresses thus range from 0.0.0.0 to 255.255.255.255,
- A total of about 4.3 billion addresses —- some are begininng to run out.
- Network Address part used to check valid hosts when connecting to Network.



Domain Names and IP Addresses

Internet nodes can have names as well as numbers

- Which is easier to remember?

  123.456.78.90 or www.mysite.com

- As Internet Grew Larger (early 1980s) so did number of nodes
- Bit like a Phone Book
  - We remember Peoples Names
  - We do not remember that many Phone Numbers
  - How many mobile numbers do you have to look up before you dial?
- Domain Name Service (DNS) Provides us with the look up of easier to remember (re guess?) names with the exact IP address needed to route over the internet

Domain Name Service (DNS)

However we need to map between names and numbers

- *Domain Name Service* (DNS)
- distributed database of names and numbers
- hosts run DNS program or know where to find one
- local mapping done by local DNS host
- remote hosts associated with remote DNS

The DNS system is, in fact, its own network. If one DNS server doesn't know how to translate a particular domain name, it asks another one, and so on, until the correct IP address is returned.

Domain Names Explained

Now we'll examine the name itself — compare to IP address format

- Domain names have at least two parts, separated by a dot or period.
- The rightmost part after the dot is called the **Top Level Domain** (TLD).

  The Top Level Domain serves to broadly categorize the name as to its type or purpose.
- Common TLDs include:

  .com — Commercial organisation worldwide (mainly US),
  .org — Organizations worldwide (mainly US) ,
  .edu — US educational institutions,
  .net — US networks, Public service, state-run offices in general
  .gov — U.S. government
  .mil — U.S. military.
  .int — International organisations.

  For example *apple.com*, *mit.edu*, *nasa.gov*.

Some More TLDs

- There are also hundreds of country TLDs, such as:

  .uk — UK,
  .us — USA,
  .fr — France,
  .de — Germany, *etc.*

- Additional *generic* TLDs (gTLDs) have been proposed. Domain names ending in .firm, .store, .web, .arts, .rec, .info, .nom and possibly many others may become more widespread on the Net.

Second Level Domains (SLDs)

- The part of the domain name before the dot is the *Second Level Domain (SLD)*.
- In the UK you can categorise institutions via the last but one rightmost field (`.uk`) will be the rightmost:

  .ac.uk — Academic institutions.
  .co.uk — Commercial organisations.

- Further left SLDs identify locations and institutions and maybe host machines. For example:

  cs.cf.ac.uk — the Computer Science (`cs`) dept and Cardiff (`cf`) University (`ac`) in the U.K. (`uk`).
  bbc.co.uk — BBC `bbc`, a company (`co`) in the U.K. (`uk`).
  albia.museo.csic.es — Host machine (`albia`) at Natural History Museum `museo`, Spain(`es`).
  spacelink.msfc.nasa.gov — Host machine called `spacelink` at Marshall Space Flight Center (`msfc`, which part of NASA (`nasa`) which is part of the U.S. federal government `gov`.

Who Controls Domain names?

- Domain Name Databases distributed over the Internet for ready access.
- Databases of domain records are maintained by a variety of agencies
  - InterNIC, the primary name registry on the Internet in the US, and
  - by a variety of agencies throughout the world.
- In 1998 the Internet more or less collapsed for a day when InterNIC was hacked.
- These databases are easily accessed from throughout the Net.
  - Accessed through a utility program called WHOIS.

Why do You need to know about Domain Names?

- Nearly all the time you use the internet you will use or need to refer to domain names.
- You will call domains up directly via

  Telnet — E.g. `telnet thrall.cs.cf.ac.uk`

  Ftp — E.g. `ftp ftp.cs.cf.ac.uk`

  WWW — you will either call domains

  explicitly via URL — E.g. `www.cs.cf.ac.uk` or

  implicitly via hypertext links.

  Email — User at (`@`) some domain. E.g. `dave@cs.cf.ac.uk`

Formal Email Address Make Up

On email you will mail `someuser` at (`@`) `somedomain`:

A typical email address looks like `user-name@domain.name`.

For example `dave@cs.cf.ac.uk` where

- `dave` is my user-name
- `@` is the standard email separator of name and domain-name
- `cs.cf.ac.uk` is the domain of this department.

Formal URL Make Up

URL stands for **Uniform Resource Locator**. The term URI (Uniform Resource Indicator) is also common, as is URN (Uniform Resource Name). Specifically, URI is a generic term which may describe both a URL and URN (essentially the same things).

A typical WWW URL is of the form

`IPtype://domain.name`.
where:

- The `IPtype` is the internet protocol type used for storage and transmission.
  - `IPtype` is typically Hypertext Transmission Protocol, `http`,
  - but can be `telnet`, `ftp`, `news` (USENET news) or even `mailto` (email) and some others.
- `://` is standard URL punctuation to separate `IPtype` from domain.

For example: `http://www.cs.cf.ac.uk`

Some More URLs

- The Domain name of a web site is commonly domain name you are used to will be prefixed by www (as in `www.cs.cf.ac.uk`, `www.bbc.co.uk`),
- Sometimes the host is named explicitly named by IP Address.

  For example: `http://www.cs.cf.ac.uk`
- Following the domain name the directory path and name of file can be specified.

  For example: `http://www.cs.cf.ac.uk/Dave/Internet`

  which accesses the `Dave/Internet` subdirectories on the server `www.cs.cf.ac.uk`

URLs for FTP

You can use a web browser to FTP (More on FTP soon)

- The format of the URL is

  ```
  ftp://myname@ftp.site.com
  ```

  where `myname` is the account login name and `ftp.site.com` is the domain name of the FTP site.
- Following this you will be asked to supply a password in a dialog box. (Some browsers allow you to configure an external FTP client application to perform subsequent FTP tasks)
- For anonymous FTP (more on this soon) you need not supply an user name, *e.g.*

  ```
  ftp://ftp.site.com
  ```
- You can also specify the path to a subdirectory or file after the site name, *e.g.*

  ```
  ftp://myname@ftp.site.com/dir/subdir/
  ```


`mailto` URL

We will see this later

- More on email protocol soon
- We frequently embed this in an HTML web page (a little while yet)
- Not a common entry as a location in a web browser (though it will work)

For completeness in our study on URLs we introduce the `mailto` URL:

- Basic format: `mailto:email address`, E.g.

  ```
  mailto:dave@cs.cf.ac.uk
  ```