EEE116 Multimedia Systems

Topic 8 Internetworking (The final one)

- Preparation of multimedia for streaming

- Network service paradigm
- Protocols and layering

- Internetworking
  - Internet architecture, concepts and protocol
  - Internet Protocol (IP)
    - Internet protocol address
    - IP PDU datagram forwarding
    - IP PDU fragmentation and reassembly
  - Transport layer protocols.
    - User Datagram Protocol (UDP).
    - Transmission Control Protocol (TCP).
- Internet Applications.

- Internet multimedia streaming

Tutorial Problems:
Q17, Q18, Q19.

Dr Charith Abhayaratne
c.abhayaratne@sheffield.ac.uk

Multimedia File Types

Multimedia Types:

?

We have learned how to

?

Usually, when they are compressed and stored in media servers (or computer hard drives) they are stored as files. Often these files are identified by a specific extension at the end of their file name.

Any known multimedia file extensions ?

Multimedia File Types

Image Formats:

JPEG -
JPEG 2000 -
BMP-
GIF
TIFF
PNG
PGM

?

Video formats:

MPEG –
H.26x –

?

Audio formats:

MPEG -

?

Any other file types?

? }

Systems Layer

Multimedia codecs are rarely used in isolation: Instead it is part of a communication system that involves:

1. coding audio visual content and metadata
2. combining the coded data
3. storing and transmitting the combined stream.

The choice of options for combining, storing and transporting varies with the application.

Usually the “systems layer” in multimedia standards specifies these options.

Audio and video coders deliver as their output elementary streams (ES) from the “compression layer”

There are 3 types of Elementary Streams:

1. video
2. audio
3. private data

Systems Layer

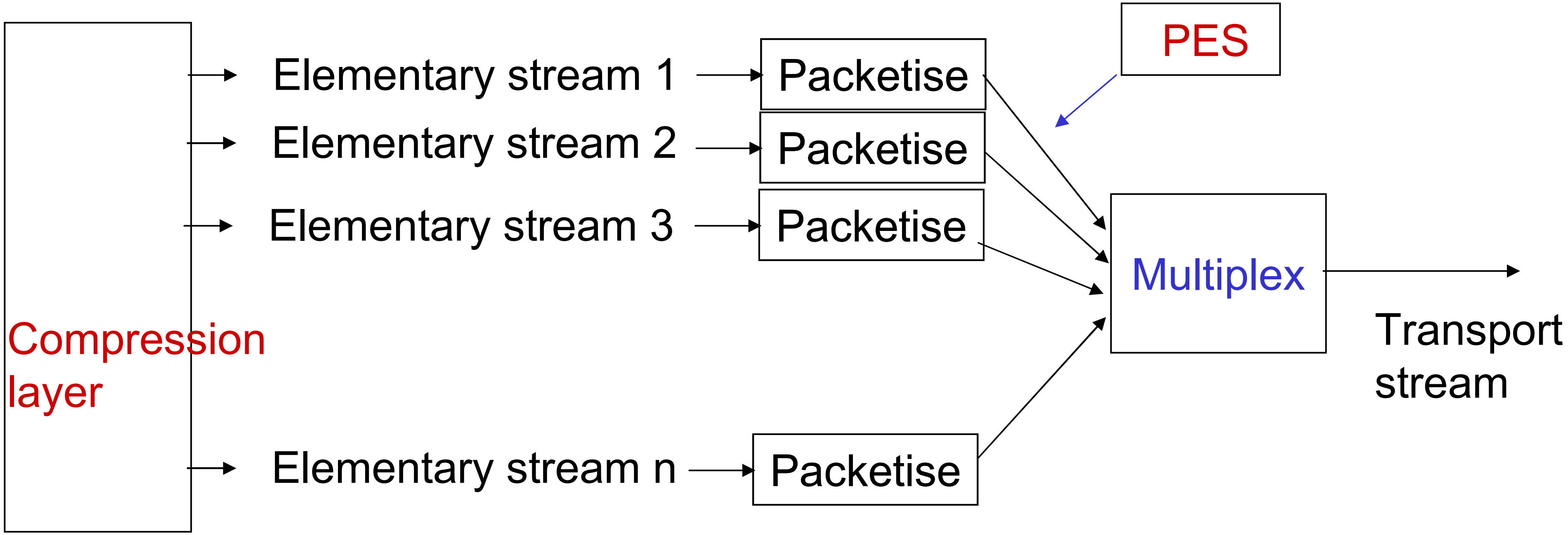
Main functions of the system layer which surrounds (or packetises) the compression layer are as follows:

1. Packetisation and combination of multiple streams into one single bit steam.
2. Addition of time stamps on elementary streams for synchronisation at play back (e.g., Lip synchronisation)
3. Initialisation and management of buffers required to decode the elementary streams.

Each elementary stream is cut into packets to form a packetised elementary stream (PES)

A packet starts with a header followed by the elementary stream’s data. Fields in the packet header vary according to the Standard.

Systems Layer



INTERNETWORKING

Service Paradigms

- Networks accept and deliver individual packets of data.

- Each packet has to be represented in the format defined by the network.

- Many network systems provide the ability to hide details of packets and provide an interface for computers specify remote destinations and send/receive data.

- The detailed interface mechanisms are called service paradigms.

-There are two types:
  - Connection-oriented service
  - Connectionless service

Service Paradigms

- Connection-oriented :

  - Two applications must establish a connection
  - Then send data across the connection
  - Terminate the connection after the data transmission ends
  - (Analogous to telephone conversation)

- Connectionless :

  - Permits an application to send a message to any destination at anytime
  - Sending application must specify a destination with each message.
  - (Analogous to sending a letter by post)

  – Which is more reliable ?

Reliability

- Reliable communication

  Guarantee delivery of every packet

  Need to confirm arrival

- Unreliable communication

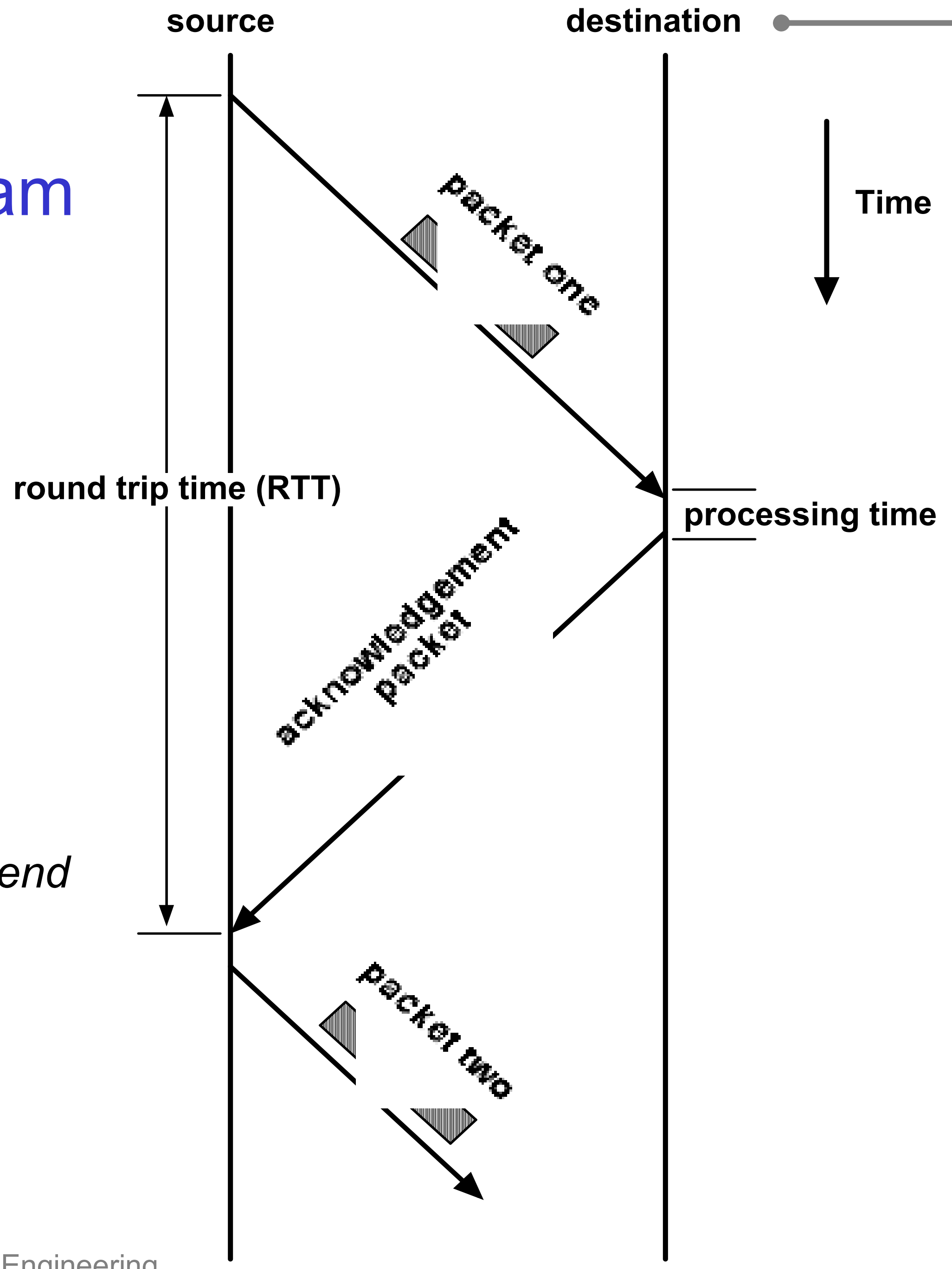
  “Best effort” approach

  Tolerate lost packets

Most computer networking uses reliable communication running on unreliable physical networks

(This is what we discussed in “Reliability” in Topic 2)

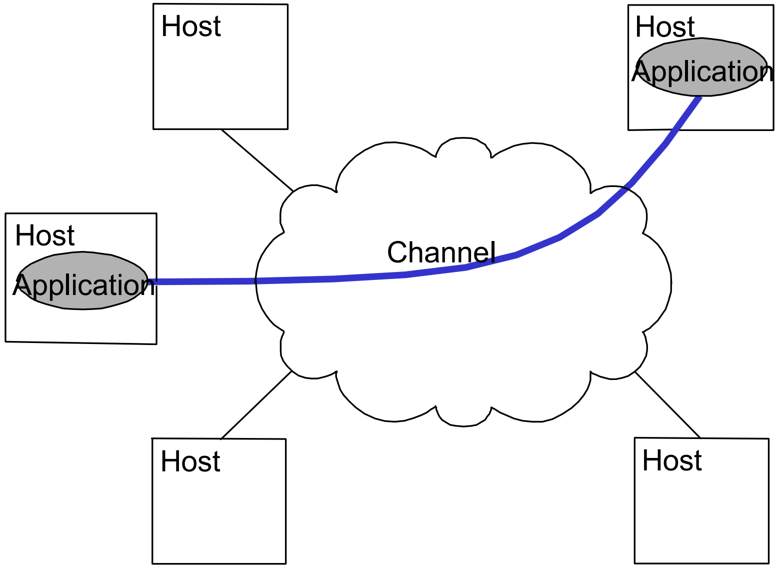
Timeline diagram



Estimate how long to send packets reliably via a geostationary satellite

Supporting Common Services

- Networks do more than just deliver data packets
- Provide for communication between applications running on distributed computers
- Provide logical channels for application-level processes (programmes)



Overall Network Architecture

- Communication hardware consists of mechanisms to transfer bits from one point to another.

- These require instructions in binary

- Usually a standardised common network software interface is used by the applications to interact with the network hardware.

- Communication protocol

    - An agreement that specifies the format and meaning of messages exchange

    -An application interacts with protocol software that follows the rules specified in the protocol when communicating.

Elements of Protocols

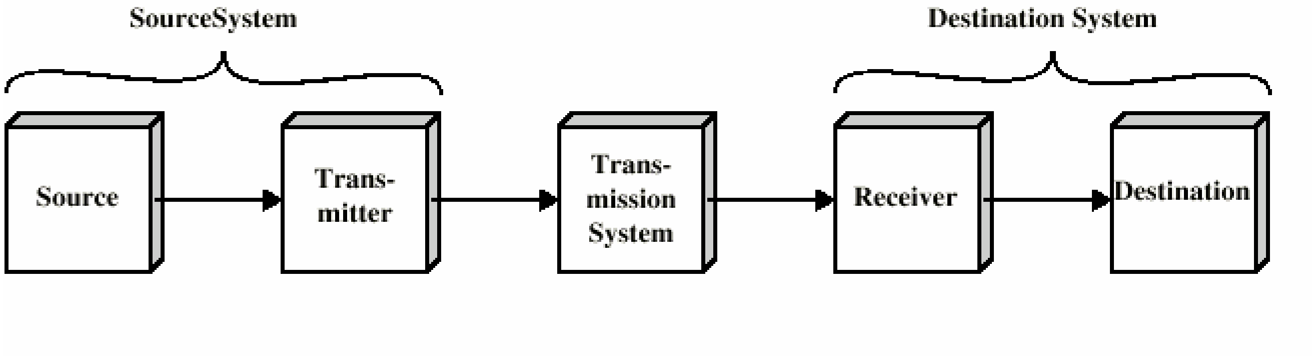
- Syntax
  - –Format of data (e.g., maximum number of bytes in a packet)
  - –Signal levels (e.g., 5 V = ‘1’ and 0 V = ‘0’)

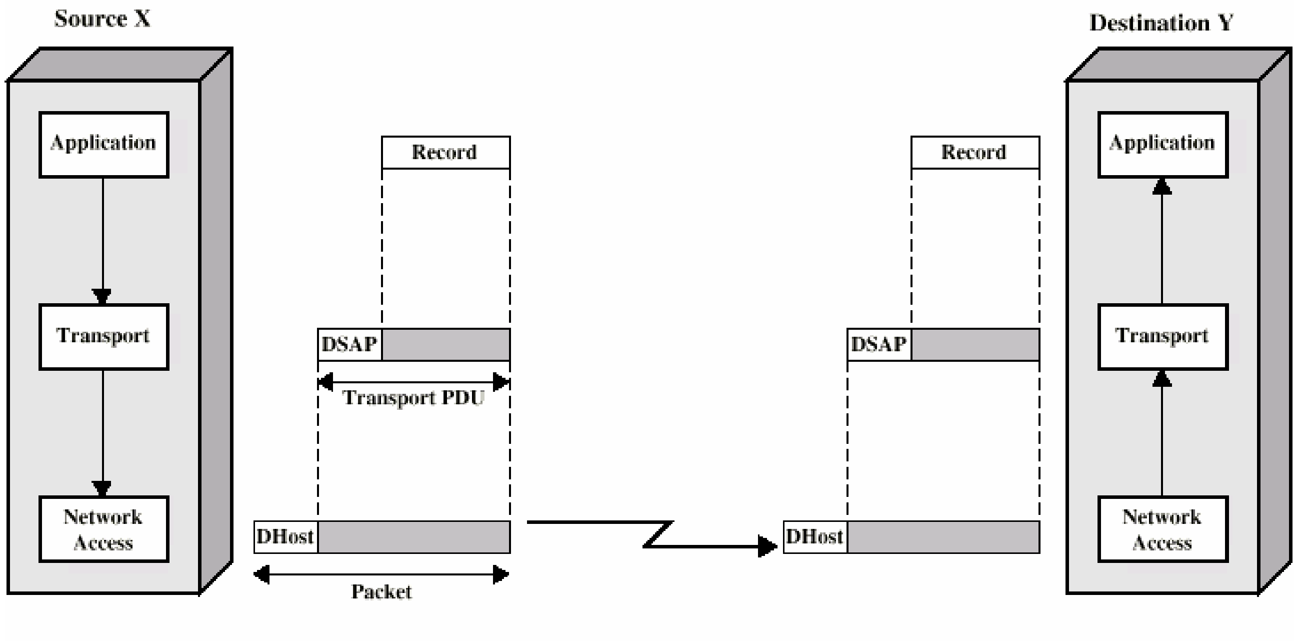
- Semantics
  - –Control information (e.g., “acknowledgement package sent only after successful receipt of data packet”)
  - –Error handling (e.g., “if next data packet not received within time, *T*, then send not-acknowledgement packet”)

- Timing
  - –Matching the speed of activities (e.g., otherwise memory overflows, etc.)
  - –Sequencing (e.g., correct of order of activities - e.g., open connection first, then …)

Task of communication broken up into modules - as a number of **layers**

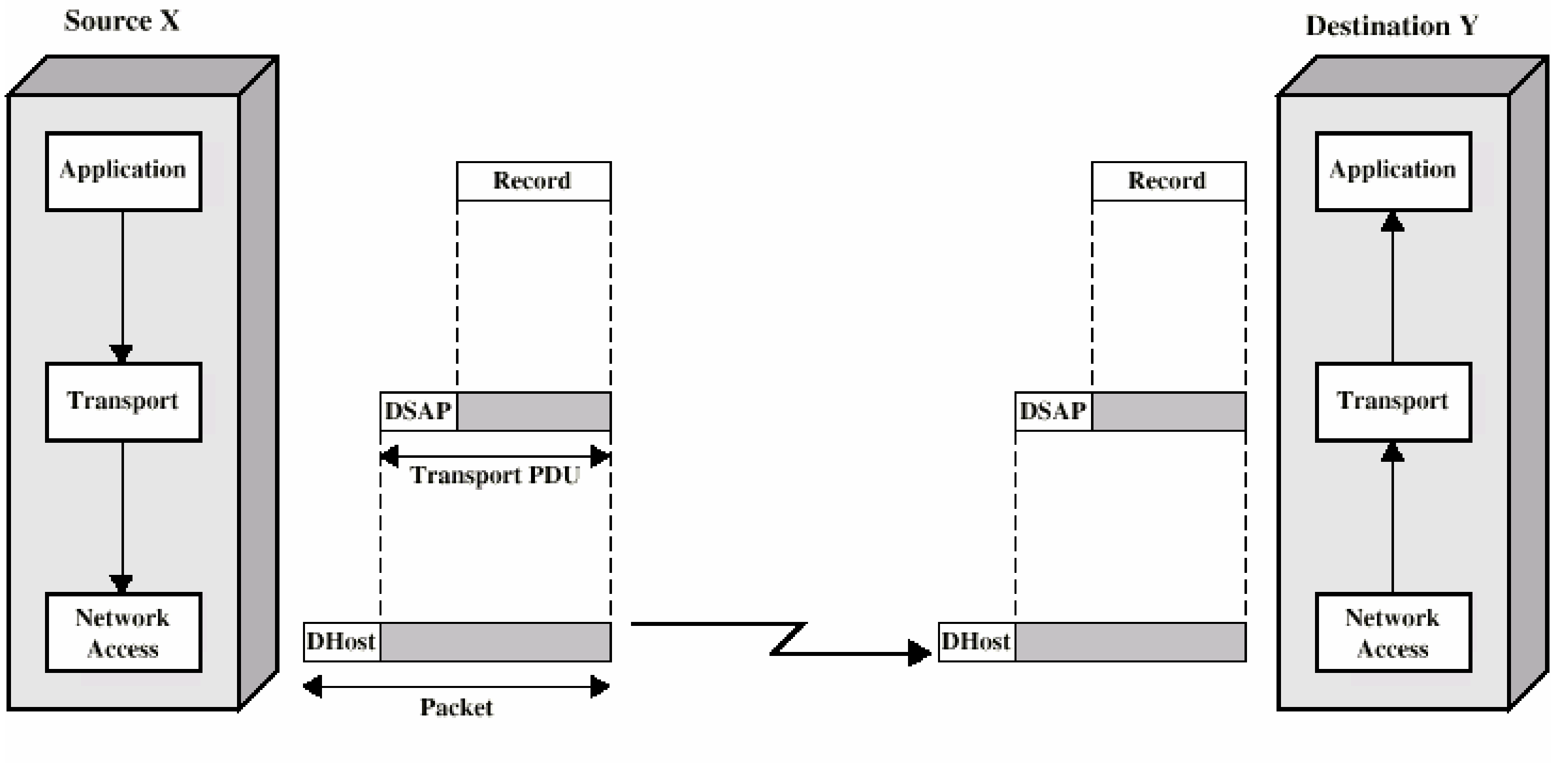
Common language between all modules using the **protocol.**



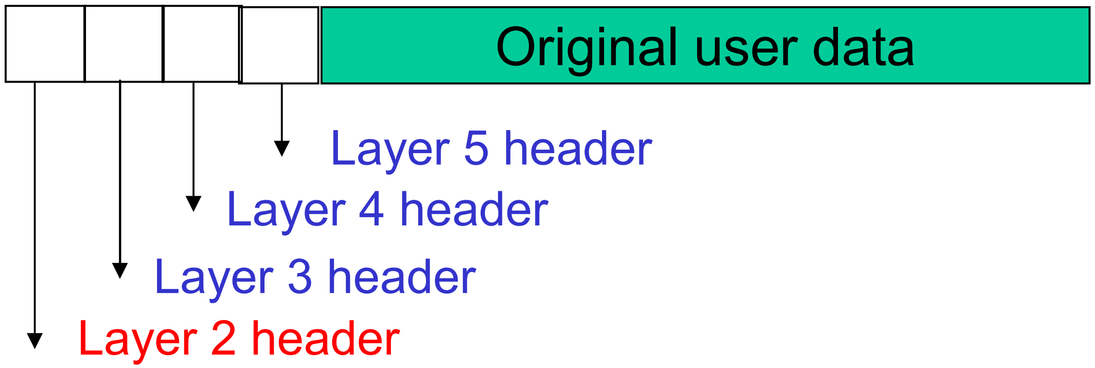


Operation of a Layered Protocol Architecture

- Break the task of communicating from one application to another into layers
- Data/instructions are passed as packets from the application to a transport layer
- Additional information added to the packet, then passed to the network access layer



Nested headers for layers



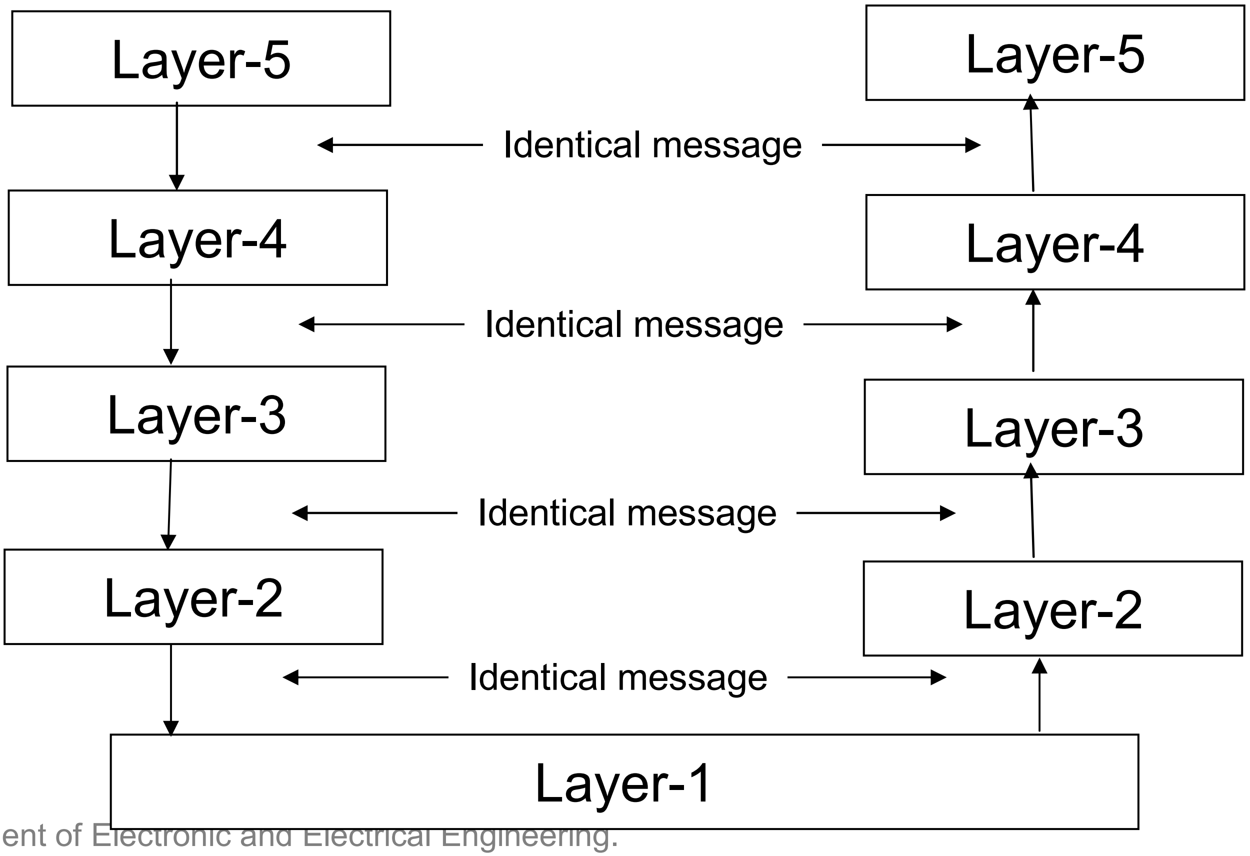
Original user data

Layer 5 header

Layer 4 header

Layer 3 header

Layer 2 header



Layer-5

Layer-5

Identical message

Layer-4

Layer-4

Identical message

Layer-3

Layer-3

Identical message

Layer-2

Layer-2

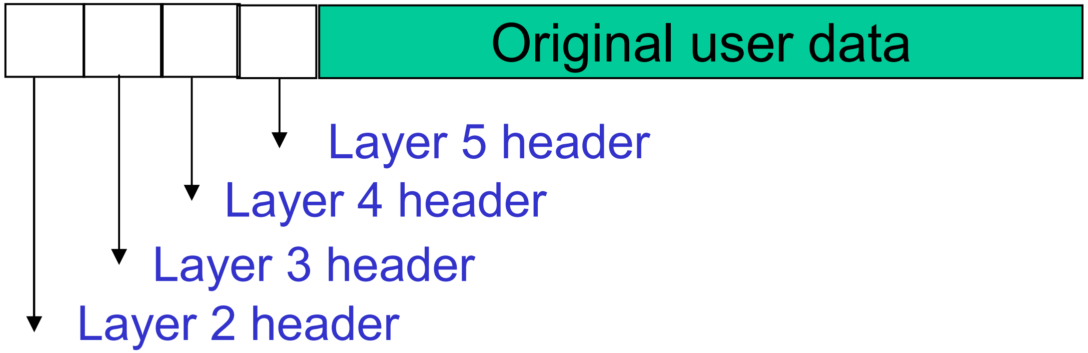
Identical message

Layer-1

Layer N software on the destination computer must receive the exact message sent by layer N software on the sending computer

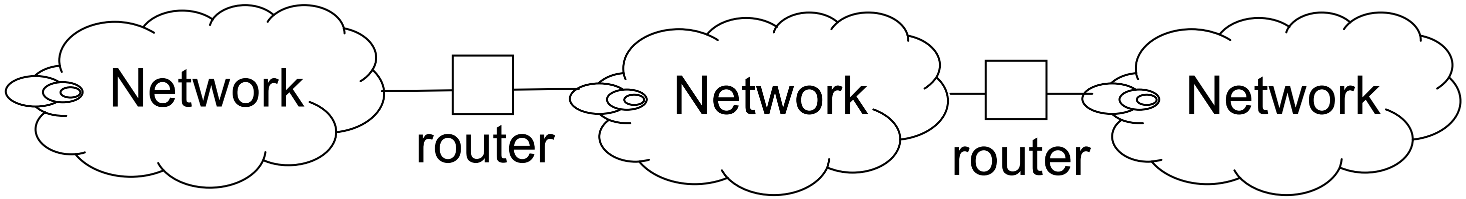
Protocol Data Units (PDU)

- At each layer, protocols are used to communicate
- Control information is added to data at each layer
- Each packet (or fragment of a packet) has a transport header added (in the transport layer)
  - Destination address
  - Sequence number
  - Error detection, etc.



These packets are called **protocol data units (PDU)**

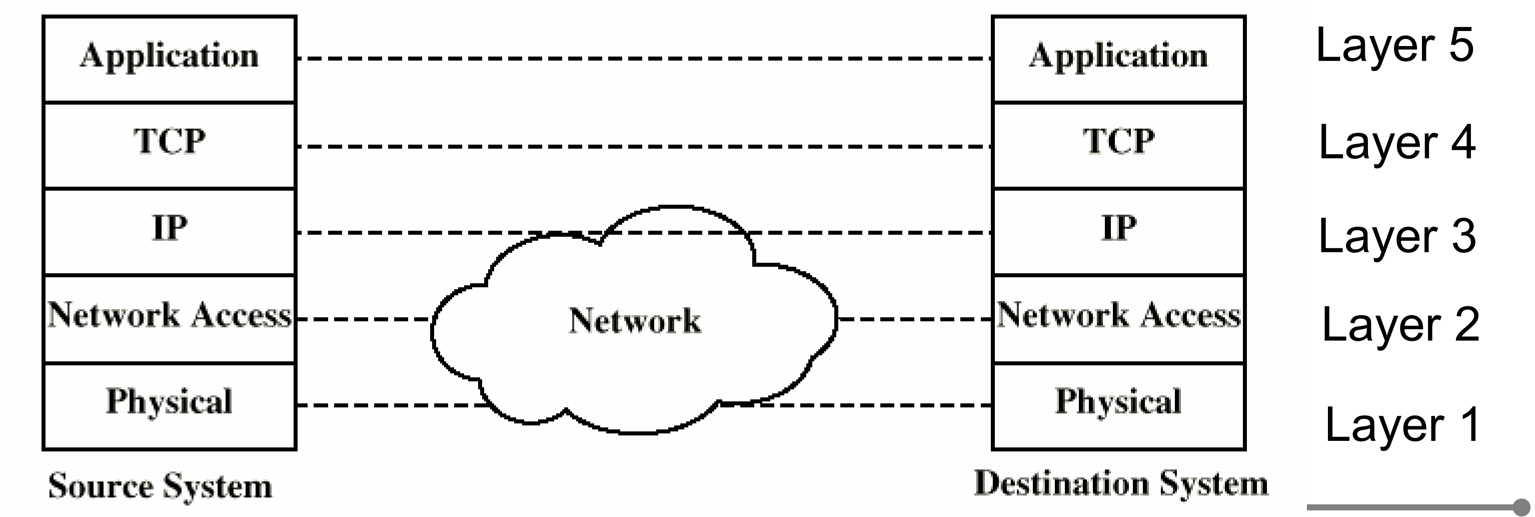
Internetworking



Internetworking is interconnecting a set of different physical networks together. The resulting system is known as Internet.

Interconnecting is via a router, which is a special purpose computer.

Internetworking uses 5-layer architecture and TCP/IP protocol suite.



Source System	Destination System	
Application	Application	Layer 5
TCP	TCP	Layer 4
IP	IP	Layer 3
Network Access	Network Access	Layer 2
Physical	Physical	Layer 1

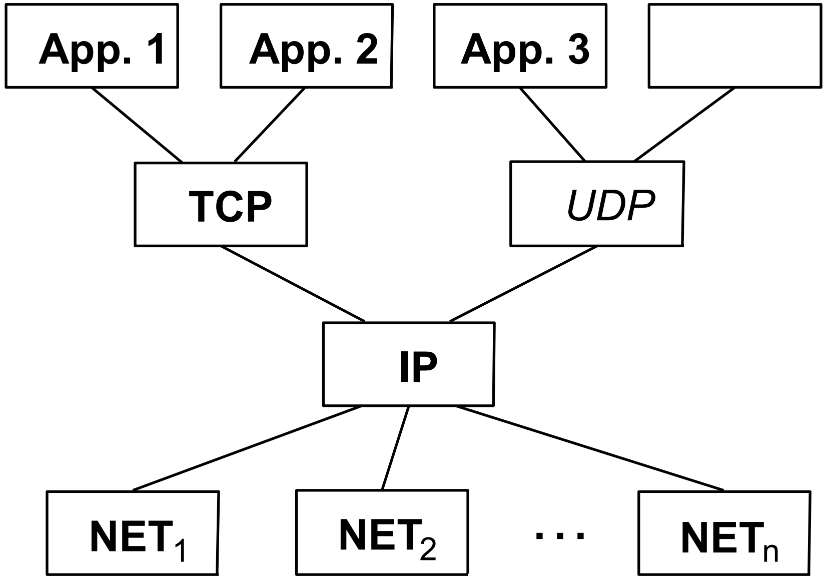
- **Network Layer**
  Concerned with handling packets along individual links, access to local network, the physical aspects of transmitting and receiving bit streams.

- **Internet Protocol (IP) Layer**
  Concerned with locating destination node and overall routing of packets across networks

- **Transport Control Protocol (TCP) Layer**
  Concerned with the reliable transmission of complete message (collection of packets) across networks from specified source and destination

- **Application Layer**
  Concerned with receiving and delivering messages between remote applications.

Internet Protocols - TCP/IP



Internet Protocol Graph

Five layer model

Transmission Control Protocol
Internet Protocol

$NET_{1..n}$ - types of physical network protocols

Implemented by combination of hardware (network adaptor) and software (network device driver)

TCP - reliable communication protocol

UDP - unreliable communication protocol

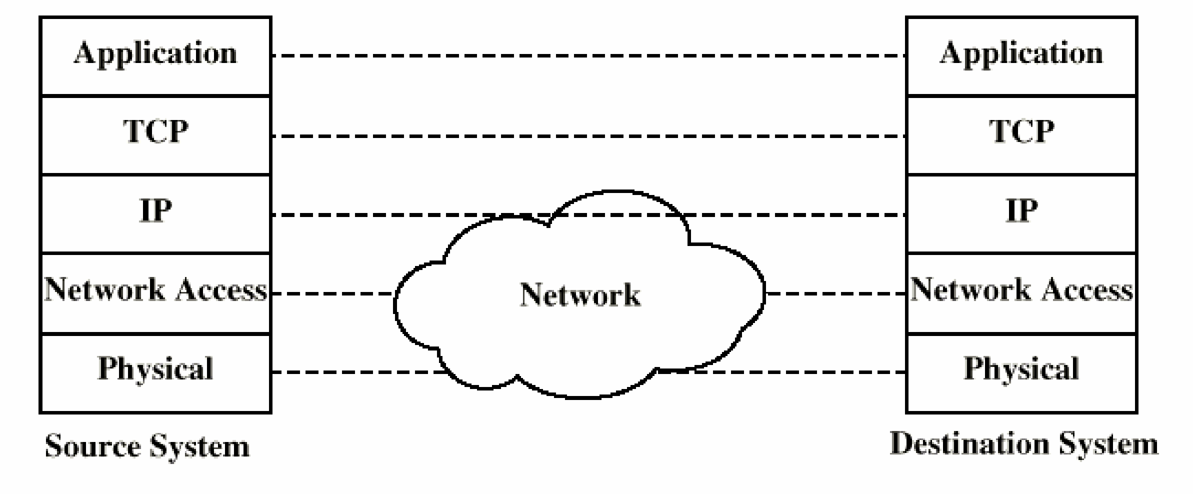
(User Datagram Protocol)

Internet Protocol

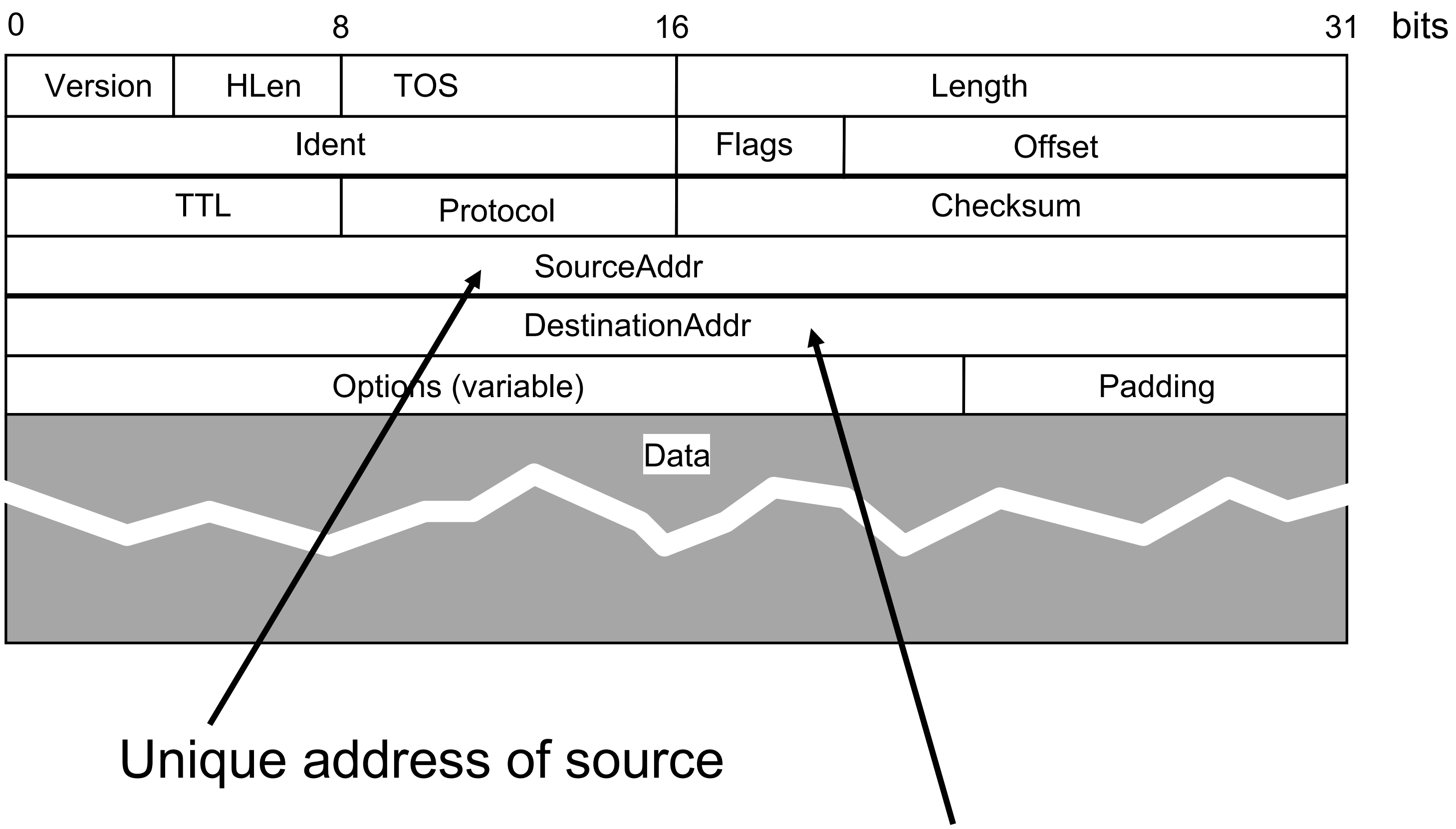
Provides an **addressing scheme** that can uniquely identify all hosts on the Internet.

Provides a **best-effort delivery** service of packets (called **PDU**)

If a packet gets corrupted, lost, etc. nothing is done about it.
This is called an **unreliable service**



Datagram Format for IP version 4



<table>
<tr><td>0</td><td></td><td>8</td><td>16</td><td></td><td>31 bits</td></tr>
<tr><td>Version</td><td>HLen</td><td>TOS</td><td colspan="3">Length</td></tr>
<tr><td colspan="3">Ident</td><td>Flags</td><td colspan="2">Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="3">Checksum</td></tr>
<tr><td colspan="6">SourceAddr</td></tr>
<tr><td colspan="6">DestinationAddr</td></tr>
<tr><td colspan="5">Options (variable)</td><td>Padding</td></tr>
<tr><td colspan="6">Data</td></tr>
</table>

Packet header

Application data (Payload)

Unique address of source

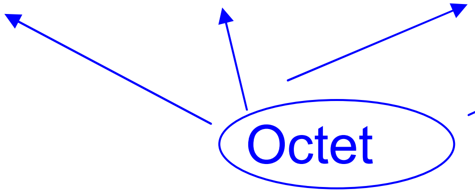
Unique address of destination

What does Payload region contain?

IP Address - 32 bits in length (2^{32} addresses)

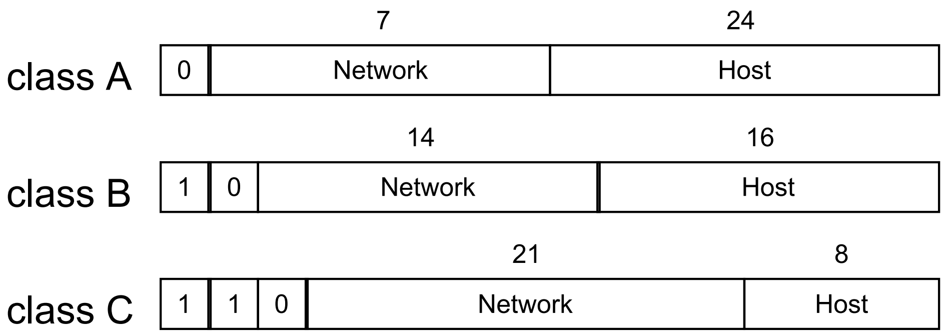
Range 0 – 2^{32} -1 = 0 - 4,294,967,295 - *sounds a big number!*

Usually expressed not as 32-bit binary number but as dotted decimal
For example, 10000000 01100000 00010001 00101001 expressed as


Octet

128.96.33.81

Useful if a network contains consecutive IP numbers - so first part of number indicates the network. In trying to locate particular host from a distance - first locate the network - then the individual host.


class A: 0 | Network (7) | Host (24)
class B: 1 | 0 | Network (14) | Host (16)
class C: 1 | 1 | 0 | Network (21) | Host (8)

First part of address specifies the network;

Second part identifies hosts within the network.

IP Address Classes

Address Class	Octets for prefix	Fixed bit values in prefix	Variable bits number in prefix	Maximum number of networks	Range of values for the first octet	Bits in suffix	Maximum number of hosts per network
A	1	0					
B	2	10					
C	3	110					

Two more classes

D - First 4 bits = 1110 :::: 2^{28} addresses for Multicast address
E - First 4 bits = 1111 :::: 2^{28} addresses (reserved for future use)

As cannot assign every IP number - there are not sufficient numbers
So:

- Reuse numbers by dynamically assigning a number just for the duration of a session

- Increase the size of the IP number - **IP version 6** uses 128 bit number
  - **Provides ~3.4 x 10^{38} nodes**
  - **Or ~10^{23} unique addresses per square metre of the Earth's surface**

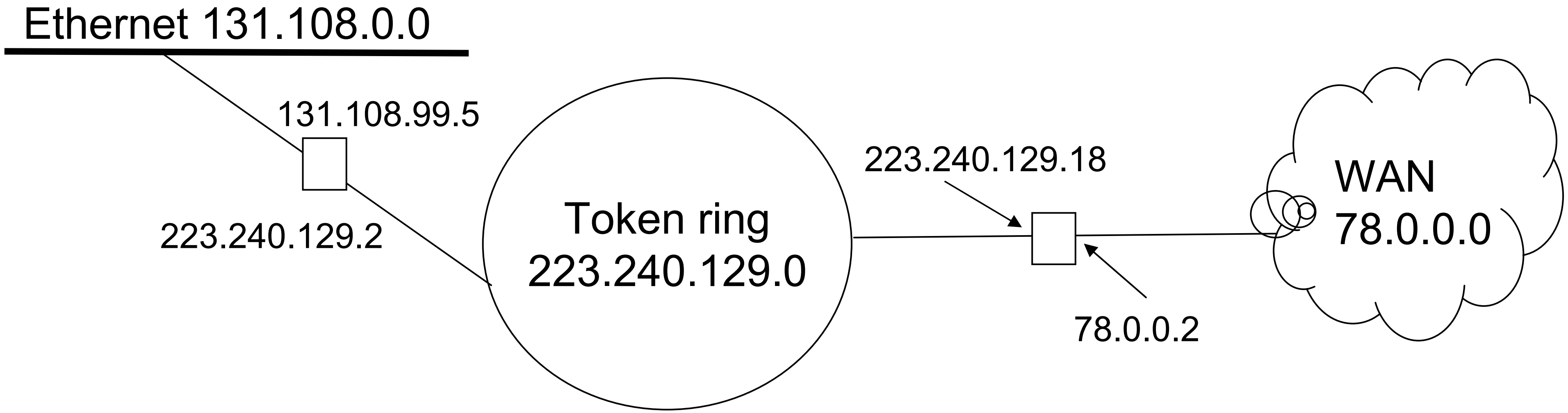
Homework Exercise:

Find the IP address of a computer connected to the University network and determine the class of IP address.

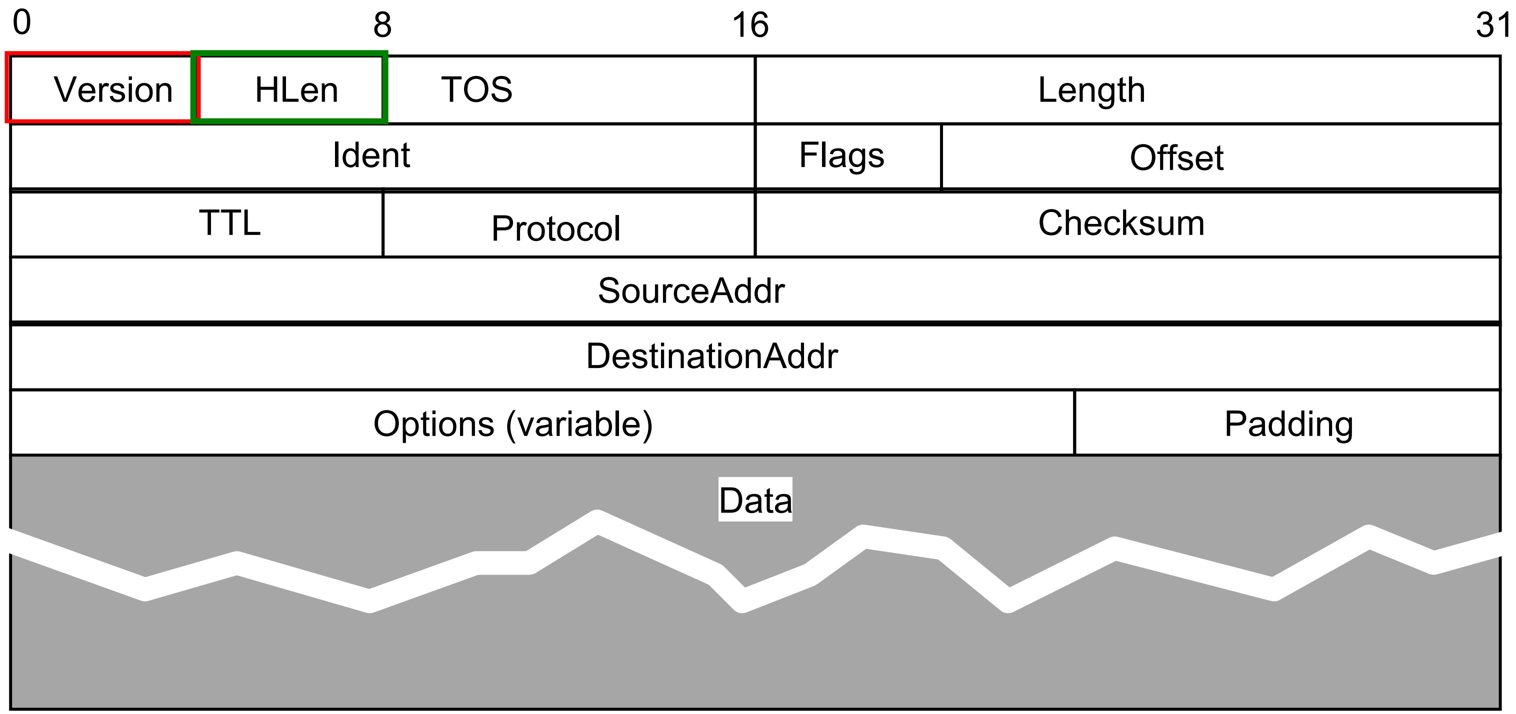
IP addresses

- An IP address
  - does not identify a specific computer
  - only identifies connection between a computer and a network.

- A computer with multiple network connections (e.g., a router) must be assigned one IP address for each connection.



IP Datagram Header



0 8 16 31

Version | HLen | TOS | Length

Ident | Flags | Offset

TTL | Protocol | Checksum

SourceAddr

DestinationAddr

Options (variable) | Padding

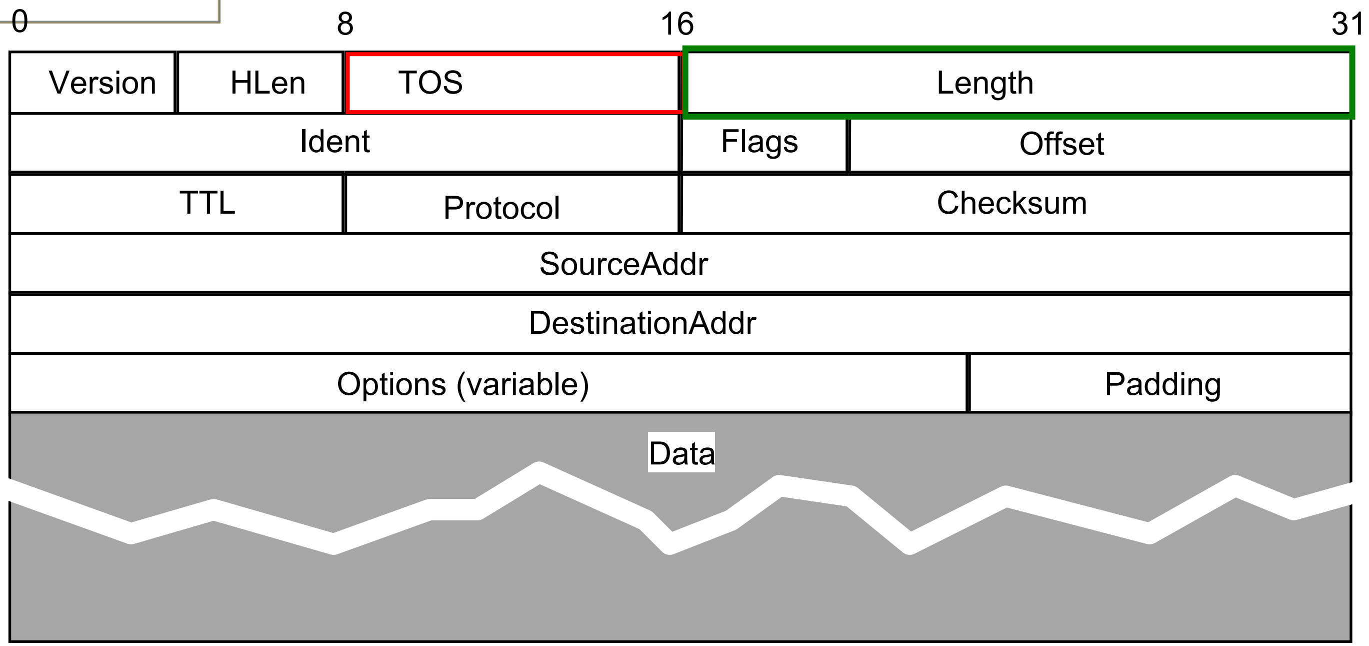
Data

Header represented as 32-bit words
Brief description of fields:

Version: identifies which version of IP

Hlen: length of header (in 32-bit words)
what is the HLen value for the above example?

IP Datagram Header



TOS: type of service (allows packets to be treated differently depending on application needs)

Length: length of datagram including header (uses 16 bits)
What is the largest datagram size?

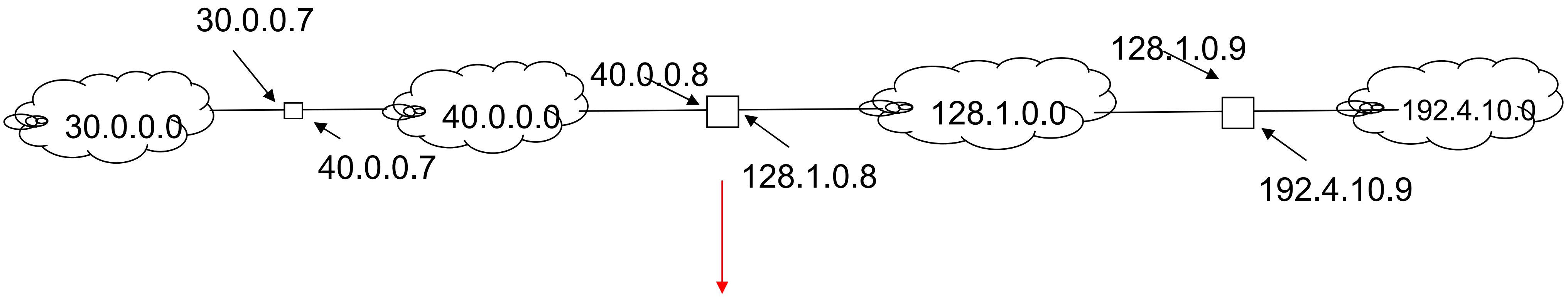
Datagram Forwarding

- The size of a datagram is determined by the application that sends data.
- Each datagram consists of a header followed by data
- Source and destination addresses in the datagram are IP addresses.

- Forwarding
    - When a router receives a datagram, it extracts its network address:
        - -First identifies the address type
        - -Then choose the appropriate network/subnet mask to extract the network address
        - -Then reads the routing table and find the next hop.

    -This is called hop-by-hop forwarding.

Routing Table Entries



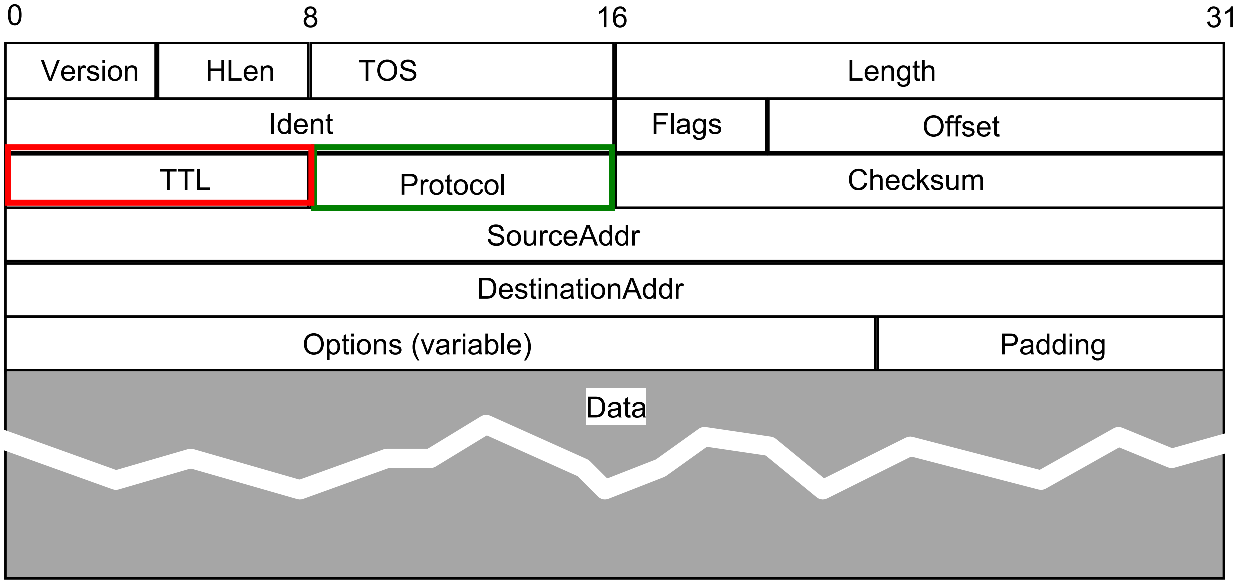
Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.8 (next router)
40.0.0.0	255.0.0.0	Deliver direct
128.1.0.0	255.255.0.0	Deliver direct
194.4.10.0	255.255.255.0	128.1.0.9 (next router)

Table contains an entry per different network.

How many rows are there in a routing table?

$2^7+2^{14}+2^{21}+1$

IP Datagram Header



TTL: time to live (historically time a datagram was allowed to exist on network; now each router decrements field by ‘1’; when field is ‘0’ datagram discarded - I.e., sets hop limit (current default is 255)

Protocol: demultiplexing key - identifies higher-level protocol (e.g., TCP = 6, UDP = 17) - ***see later***

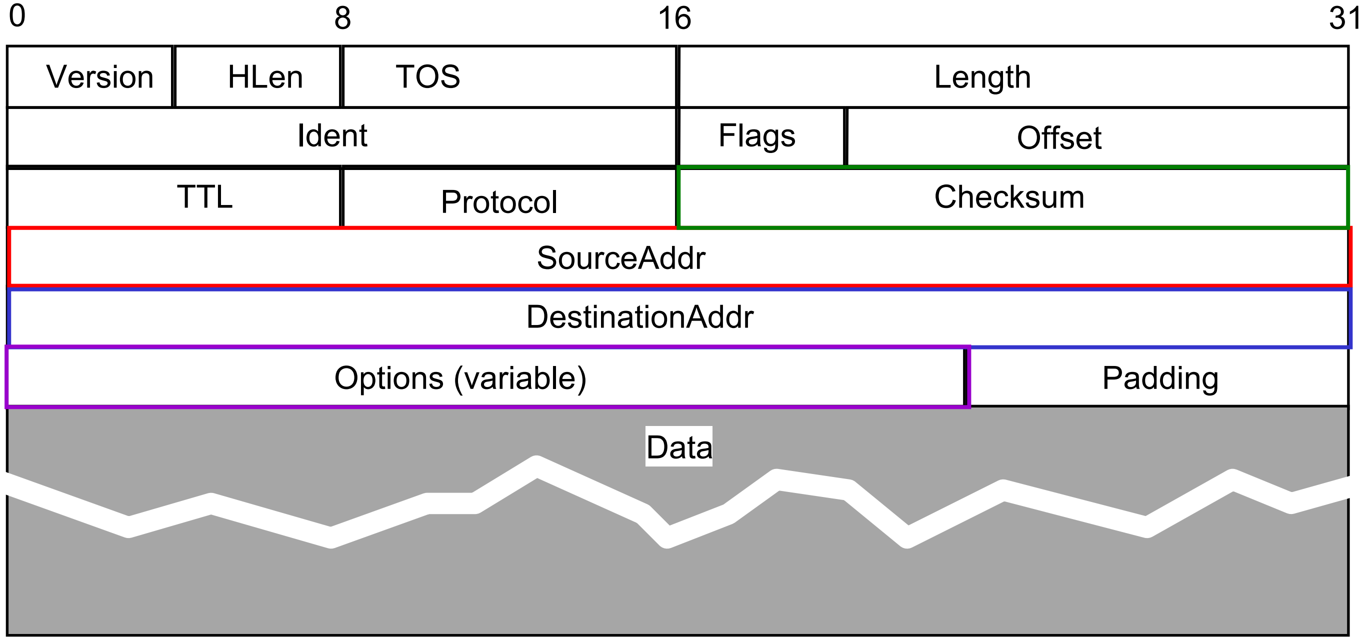
Experiments

- traceroute
  - Visit http://www.net.cmu.edu/cgi-bin/netops.cgi
  - Use “traceroute” command to trace the hops to www.shef.ac.uk from http://www.net.cmu.edu.
  - Also http://www.visualroute.com/support/newrelease.html for a visual route of packet transfer.
- ping
  - Use DOS prompt

```
   • U:\>ping www.bbc.co.uk
   • Pinging www.bbc.net.uk [212.58.224.84] with 32 bytes of data:
   • Reply from 212.58.224.84: bytes=32 time=6ms TTL=246
   • Reply from 212.58.224.84: bytes=32 time=6ms TTL=246
   • Reply from 212.58.224.84: bytes=32 time=7ms TTL=246
   • Reply from 212.58.224.84: bytes=32 time=6ms TTL=246

   • Ping statistics for 212.58.224.84:
–          Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   • Approximate round trip times in milli-seconds:
–          Minimum = 6ms, Maximum = 7ms, Average = 6ms
```

IP Datagram Header



0 8 16 31

Version | HLen | TOS | Length

Ident | Flags | Offset

TTL | Protocol | Checksum

SourceAddr

DestinationAddr

Options (variable) | Padding

Data

Checksum: 1’s complement sum of entire header (as 16-bit words)
– to check whether any data in header is corrupted

SourceAddr: Source address (32-bit)

DestinationAddr: Destination address (32-bit)

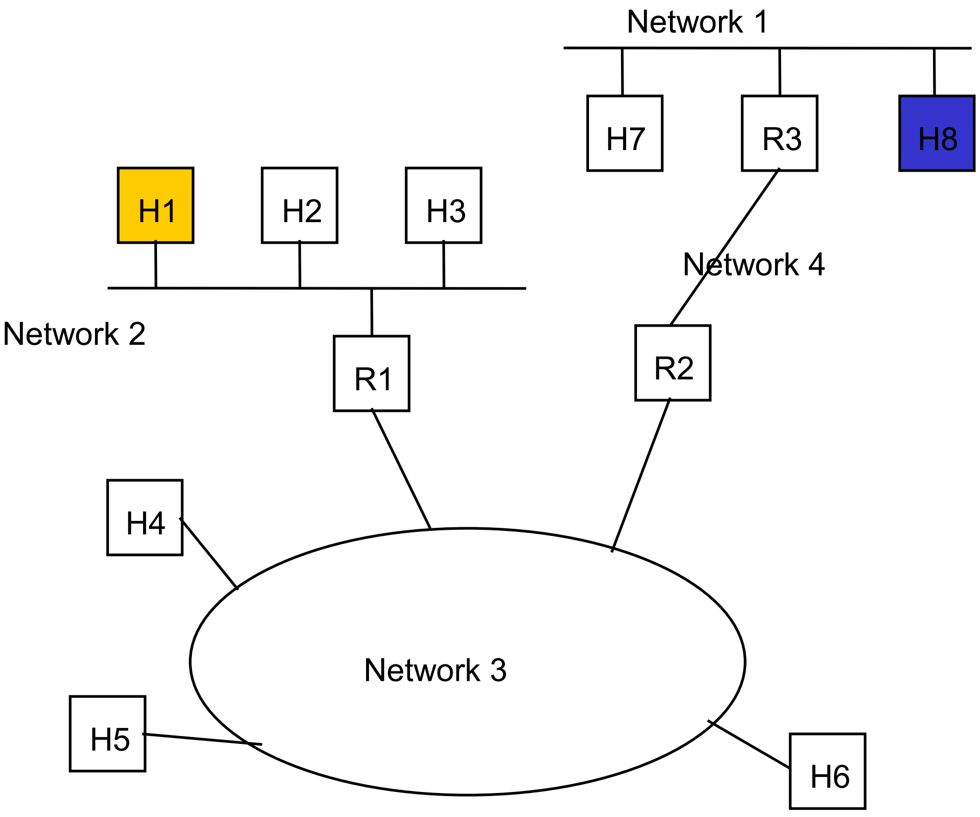
Options: rarely used - presence detected by examining HLen

Datagram Forwarding

- Each datagram contains the full IP address of the destination
- Each router reads this to determine where to send datagram
- A hop-by-hop approach

Suppose H1 –> H8

- Different network so sends to its router, R1
- R1 cannot deliver directly as not Network 3
- Sends to its master router R2
- R2 has forwarding table, which notes H8 address is for network number 1
- Forwards to R3 - which delivers datagram directly

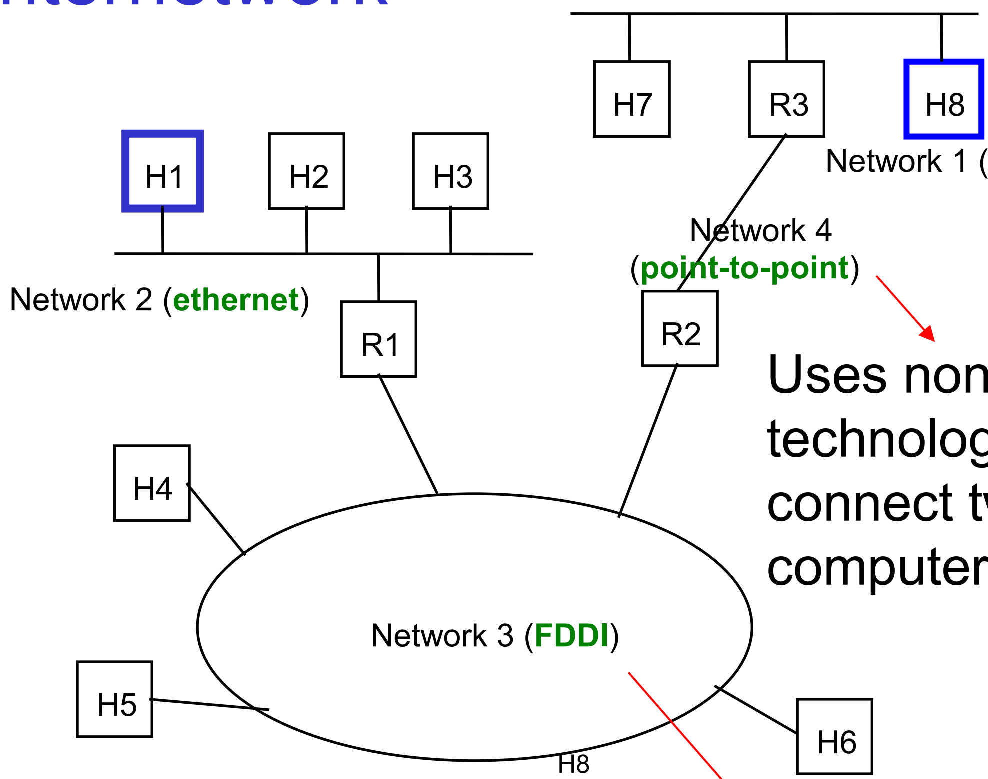


A Simple Internetwork

R - Router: network node connected to two or more networks

H - Host: individual computer

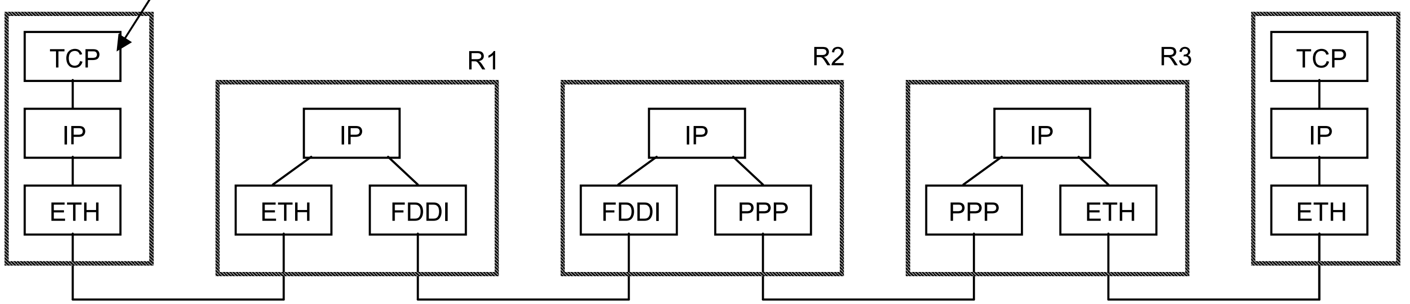
Types of physical network



Uses non-shared technology to connect two computers.

Fibre Distributed Data Interconnect

Higher-level protocols (TCP)



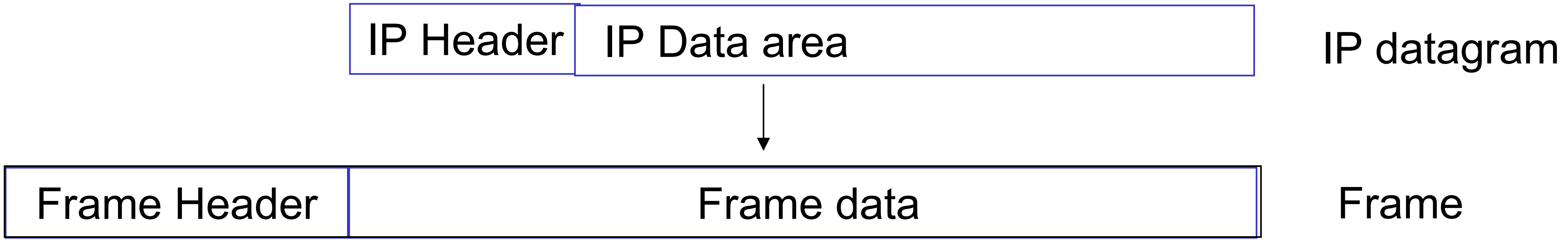
Internet Protocol (IP) runs on all nodes - enables the different physical networks to operate as a single logical network

Encapsulation

-There are several types of network technology

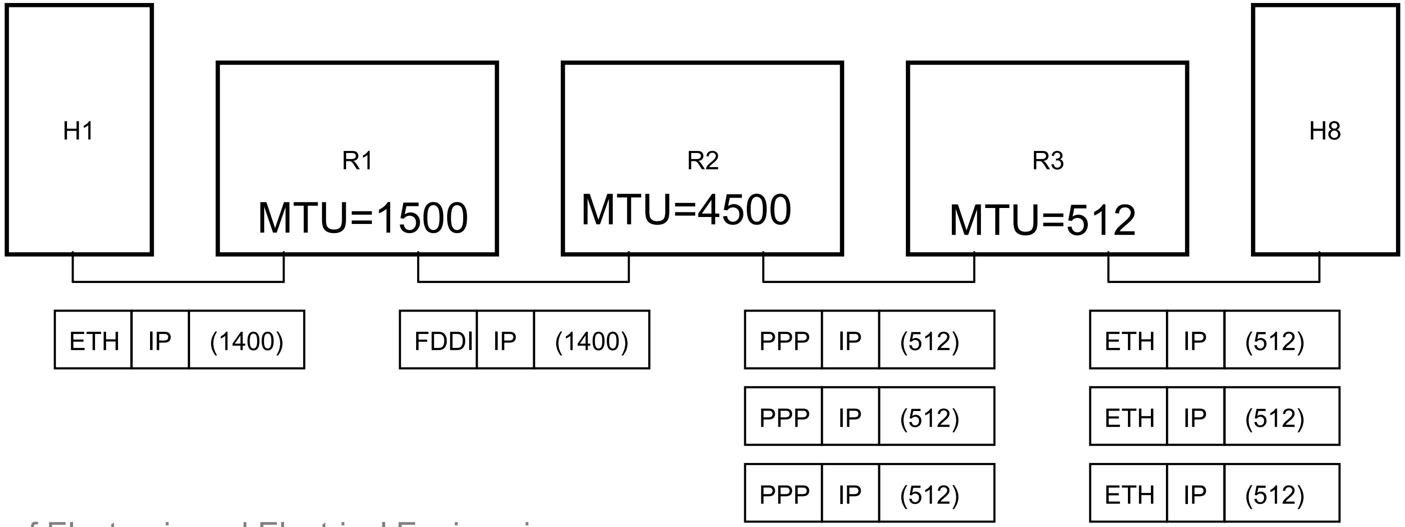
-E.g., FDDI, point-to-point (ppp), Ethernet, etc

- The technique known as “encapsulation” is used to transmit a datagram, in a common format, across different types of physical networks, that do not understand the datagram format.
- In this process, an IP datagram is encapsulated in a frame.
- A frame is the form of a data packet that the underlying hardware accepts and delivers.
- The entire datagram is placed in the data part of a frame.
- Frame header records the destination address, which is the same as the address of the next hop to which the datagram should be sent.

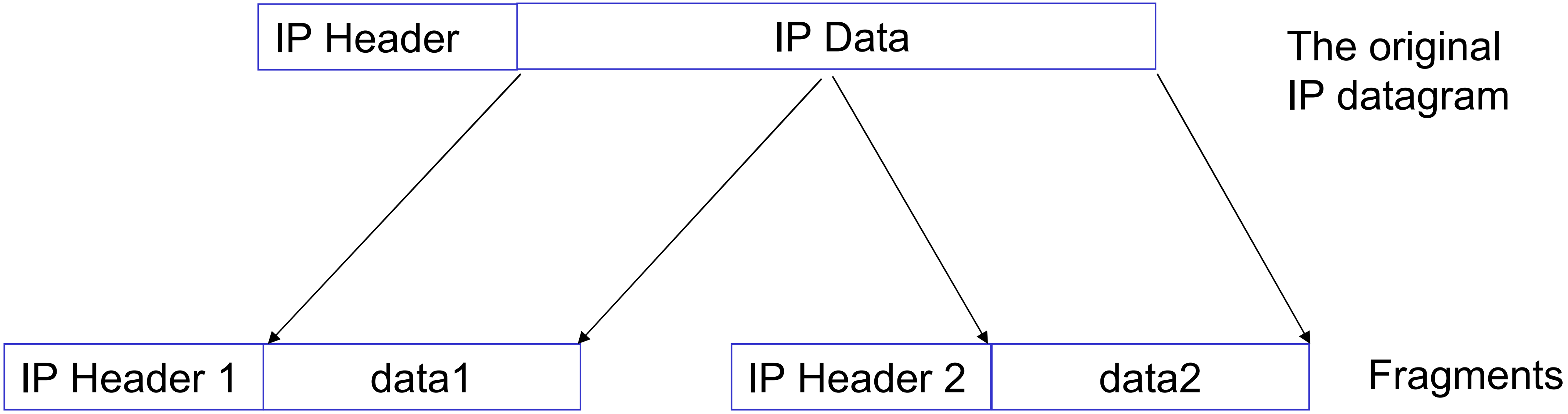


Maximum Transmission Unit (MTU)

- Each network technology specifies the maximum amount of data that a frame can carry.
- This limit is called **maximum transmission unit (MTU).**
- The maximum packet size varies with the type of network
  - E.g., - Ethernet = 1,500 bytes; FDDI = 4,500 bytes
- Hence IP service model must take this into account
- **Fragmentation** occurs when a router receives a datagram larger than the MTU of the outgoing network.
  - i.e., the router divides the datagram into small fragments and send each fragment independently.
- **Reassembly** (usually) only occurs at receiving host



Fragmentation

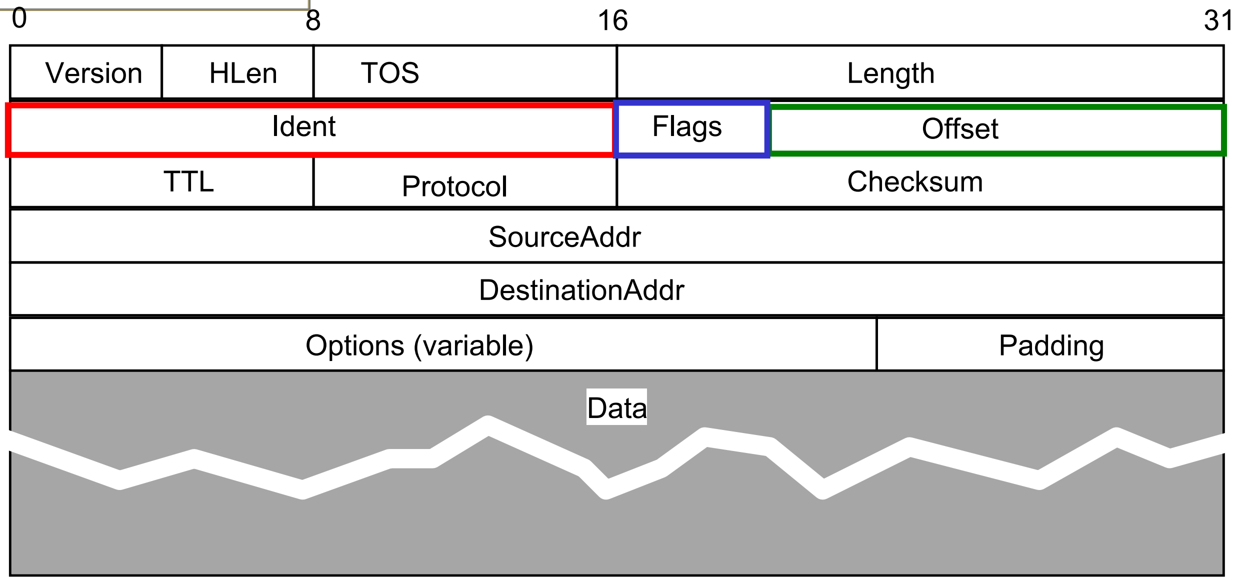


Router fragments data into N components.
It copies the original data header into new datagram headers.

The reverse process is called reassembly.

How does the destination know whether the received datagram is a fragment or not?

Fragmentation information in IP datagram


0 8 16 31

Version | HLen | TOS | Length

Ident | Flags | Offset

TTL | Protocol | Checksum

SourceAddr

DestinationAddr

Options (variable) | Padding

Data

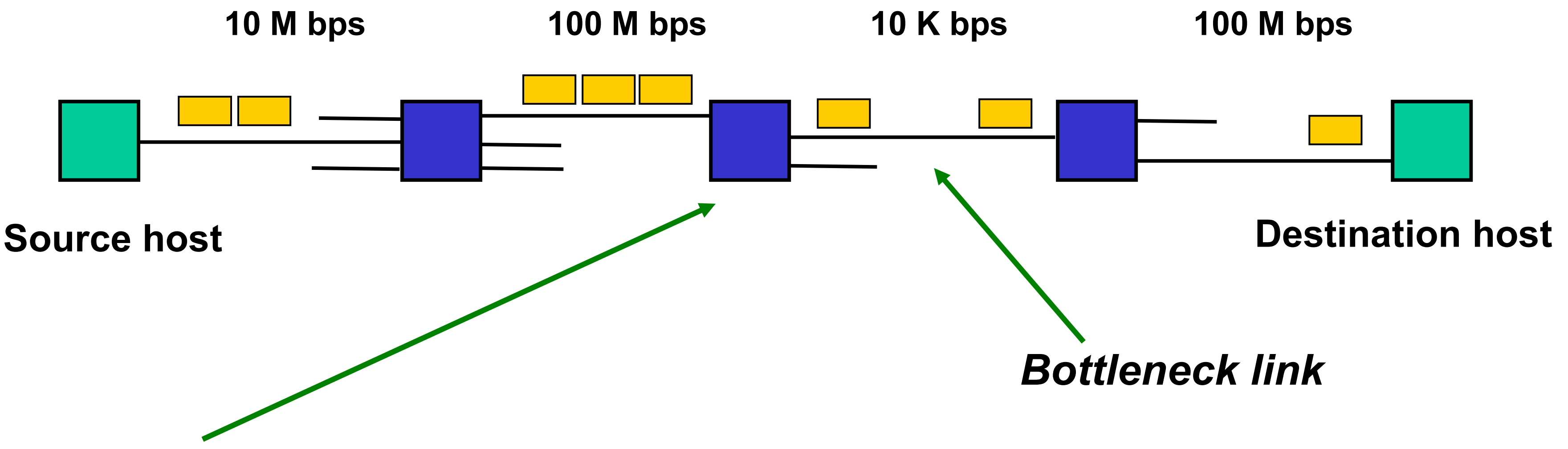
Fields Identification (Ident), Flags and Offset in the IP datagram represent the fragmentation information.

Flags : A 4 bit field to give specific information. A specific bit in the Flags is set to indicate that the datagram is a fragment.

Identification : This is the unique identification number placed in an outgoing datagram by the sender host. Therefore all fragments of a given datagram has the same identification number.

Offset : This specifies where in the original datagram the fragment belongs. i.e., it tells the receiver how to order fragments for reassembly.

Best-effort delivery



- Packets pile up at this router
- When buffer memory is full, packets eliminated (dropped)
- Source just keeps sending packets
- So communication is unreliable - packets are lost.
- And “wasted” packets just add to congestion on the network
- Also the source does not know if the destination is available to accept packets (or even that it exists!)

Common errors in best-effort delivery

- Datagram duplication
- Delayed or out-of order delivery
- Corruption of data
- Datagram loss

- IP is designed to work over all types of network hardware
  - The undelying hardware may misbehave.
  - This results in errors.

- **Higher layers of protocols are required to handle these**

- So far we considered **host-to-host** packet services.
  - Pass packets from one host to another specified host.
- Now need to consider **process-to-process** communication.
  - Often called **end-to-end protocols**.
- This is the main functionality of the transport layer.

- General properties of **transport layer**

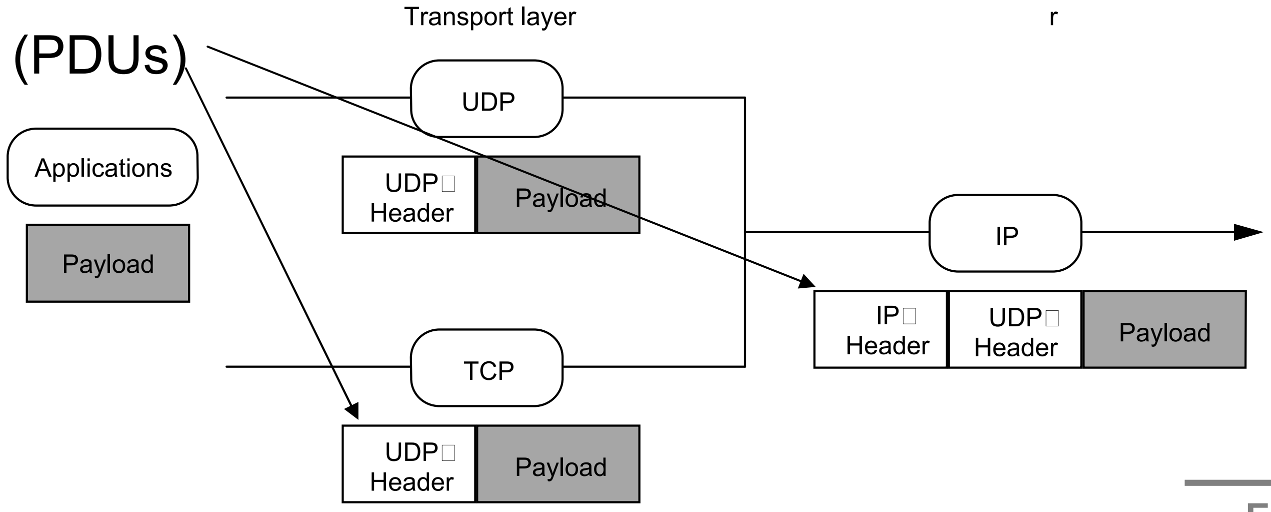
- Guarantees message delivery
- Delivers messages in the order sent
- Delivers at most one copy of the message
- Supports arbitrarily large messages

- Allows synchronisation of sender and receiver
- Supports multiple application processes on same host
- Permits receiver to use flow control

Connection oriented and Connectionless transport

- Transport protocols support two forms of communication
  - Connection-oriented :
    - Two applications must establish a connection
    - Then send data across the connection (e.g., TCP)
  - Connectionless :
    - Permits an application to send a message to any destination at anytime
    - Sending application must specify a destination with each message (e.g., UDP)

Protocol Data Units (PDUs)

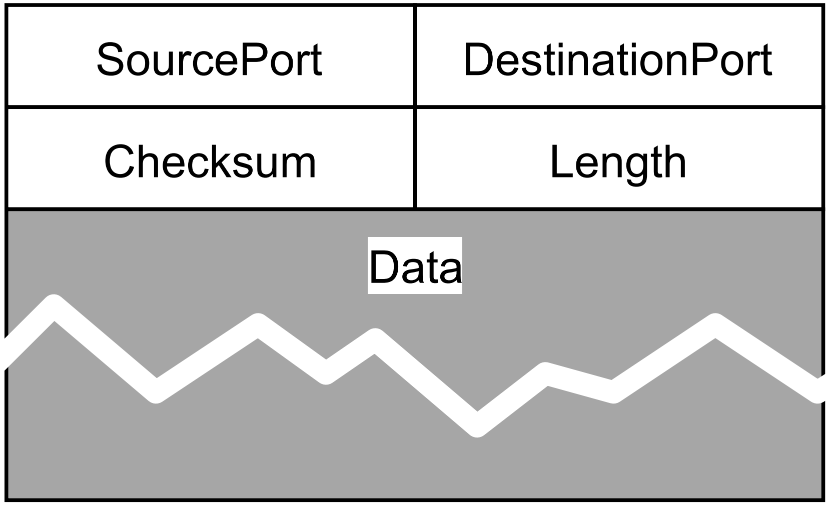


User Datagram Protocol (UDP)

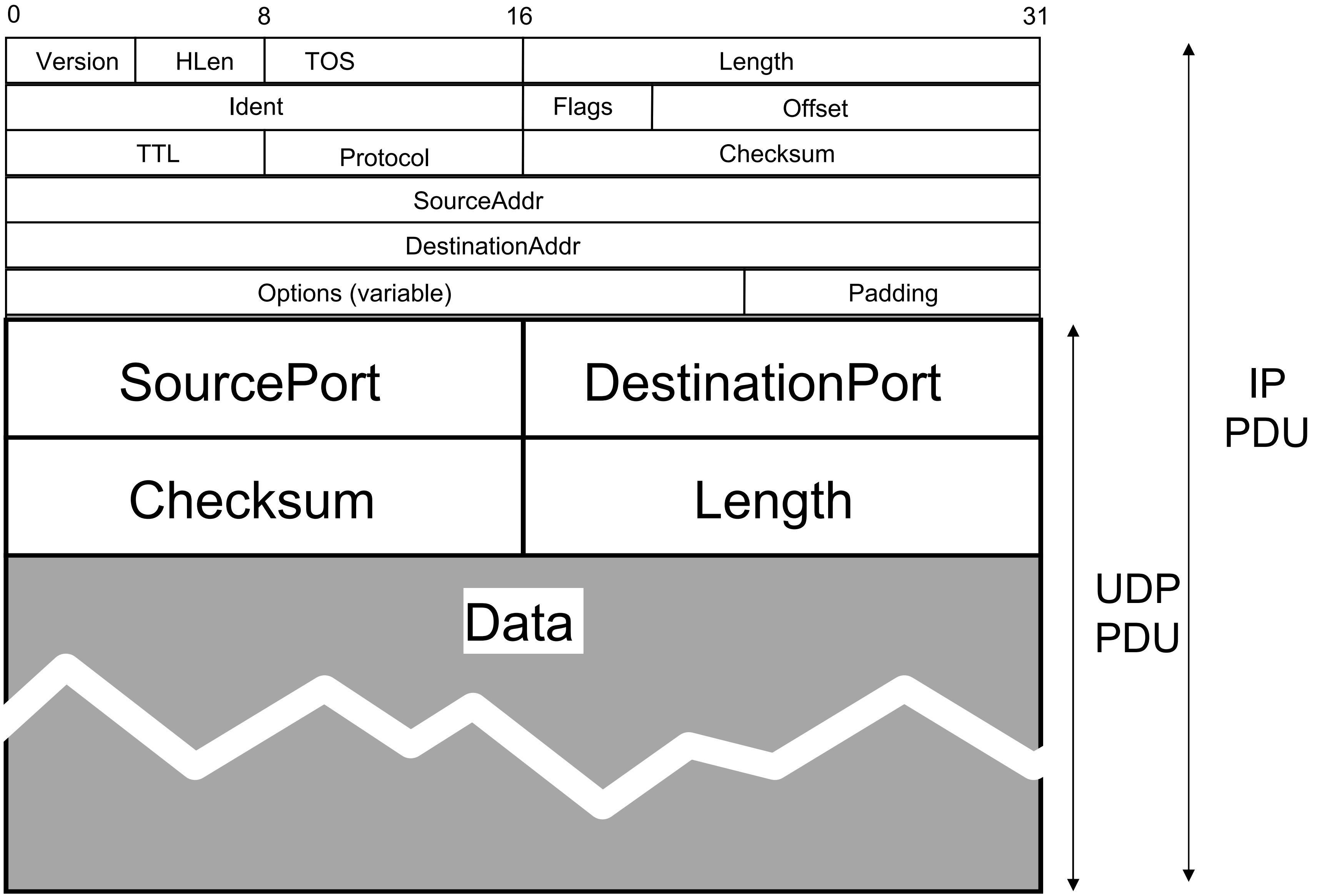
- Only additional function over IP is to provide demultiplexing for multiple processes

- Achieved using an identifier for each process - a *port number*
  - *Example port numbers:*
    - *20 = File Transfer*
    - *25 = Email*
    - *80 = WWW*

- Source process sends message to a port;
- Destination process receives message from a port



SourcePort	DestinationPort
Checksum	Length



0
8
16
31
Version
HLen
TOS
Length
Ident
Flags
Offset
TTL
Protocol
Checksum
SourceAddr
DestinationAddr
Options (variable)
Padding
SourcePort
DestinationPort
Checksum
Length
Data
UDP
PDU
IP
PDU

Transmission Control Protocol (TCP)

Reliable, Connection-Oriented, Byte-Stream Protocol
{Not the only service operating on IP}
(Why and) What TCP has to do:

- Needs to make explicit connections - establishment and teardown phases.
- Cope with very variable RTTs.

  (a few ms across campus, 100 ms across Europe (morning) or 400 ms (afternoon))
- Packets can get reordered across Internet.
- Cope with very old packets that are received.
- Variable amount of data in the “pipe” - need to establish resources at destination.
- Cope with network congestion - packets sent faster than can be received.

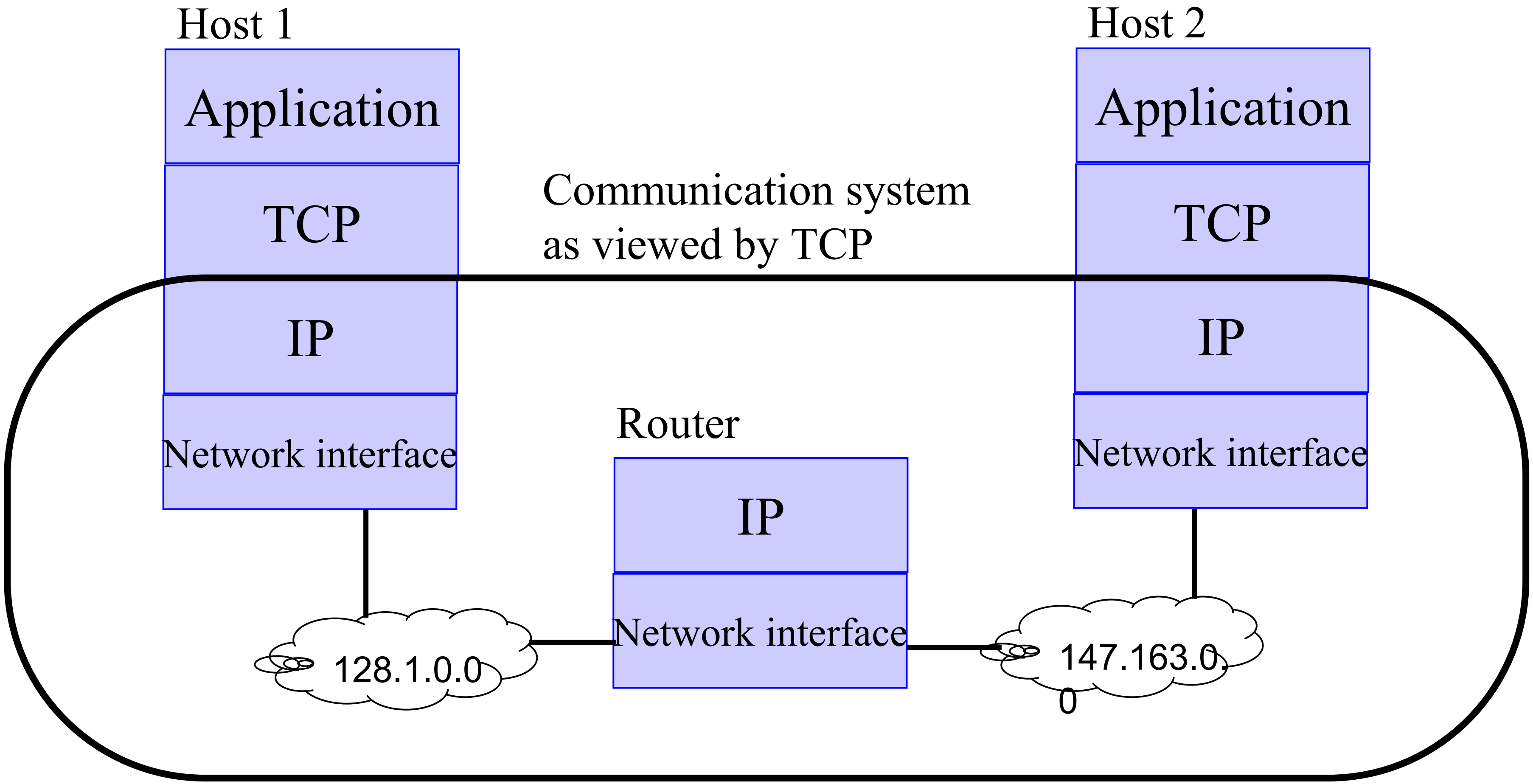
TCP services to applications

- TCP provides
  - Complete reliability (no loss or duplication of data)
  - Connection-oriented full-duplex stream transport service
- Allows two application programmes to
  - Form a connection
  - Send data (in either direction)
  - Terminate the connection.
- Each connection is
  - Started reliably
  - Terminated gracefully

How TCP works

- TCP is an end-to-end protocol
  - i.e., provides a connection directly from an application in one computer to an application on another computer.
- These are virtual connections achieved by software.
  - Two machines exchange messages to achieve the illusion of a connection.
- TCP uses IP to carry messages
  - As a packet communication system that connects hosts at two end points of a connection
- IP treats each TCP message as data to be transferred.

An Example

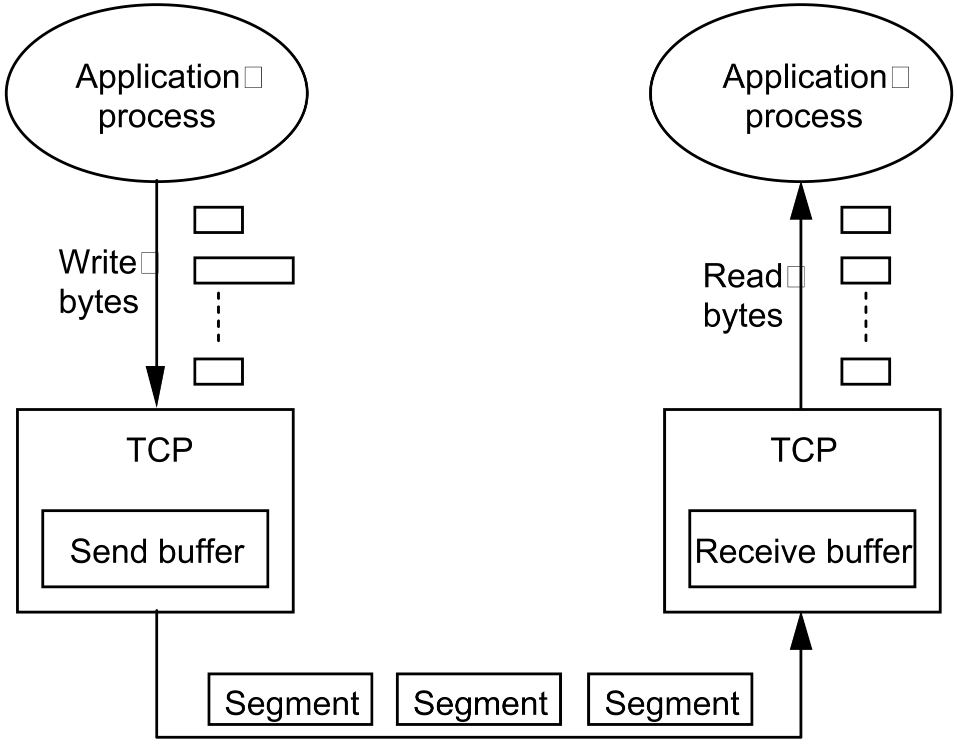

Host 1
Application
TCP
IP
Network interface
Host 2
Application
TCP
IP
Network interface
Communication system
as viewed by TCP
Router
IP
Network interface
128.1.0.0
147.163.0.
0

- TCP buffers “application” bytes until it has sufficient data to fill a packet - called segments

- Sends a segment when:
  - Buffered bytes = maximum segment size (MSS)
    - MSS set so that local IP does not have to fragment

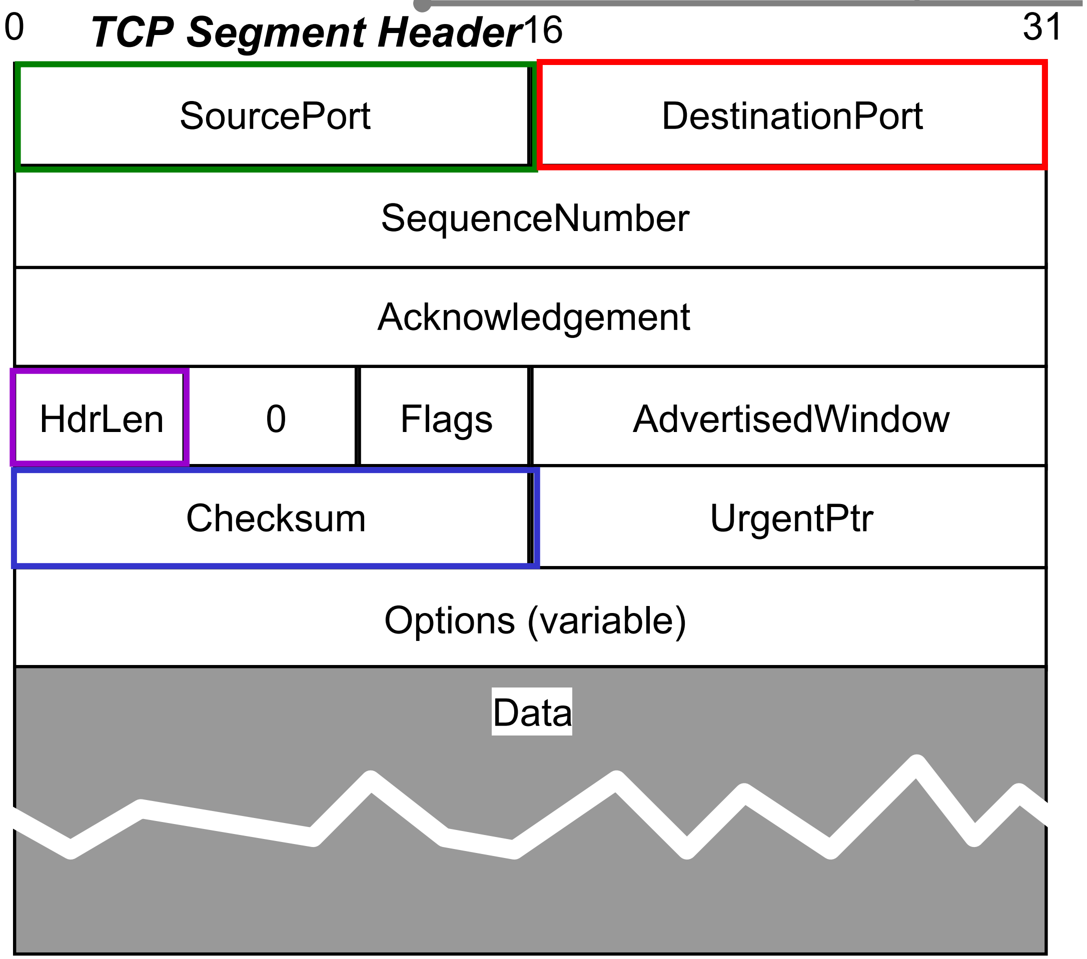
    OR
  - When issued a push operation from application

    OR
  - After a certain time has elapsed.



Each segment contains TCP header

- **SourcePort** and **DestinationPort** are demultiplexing keys as in UDP
- **SourcePort**: identifies the programme that sent data
- **Destination port:** identifies which programme in the receiving computer should receive the data.
- Together with source and destination IP addresses they uniquely define each TCP connection.
- **Checksum**: contains a checksum that covers TCP segment header.
- **HdrLen:** the header length as in IP datagram (4 bits)



Flags contain 6 bits and specifies the segment type.

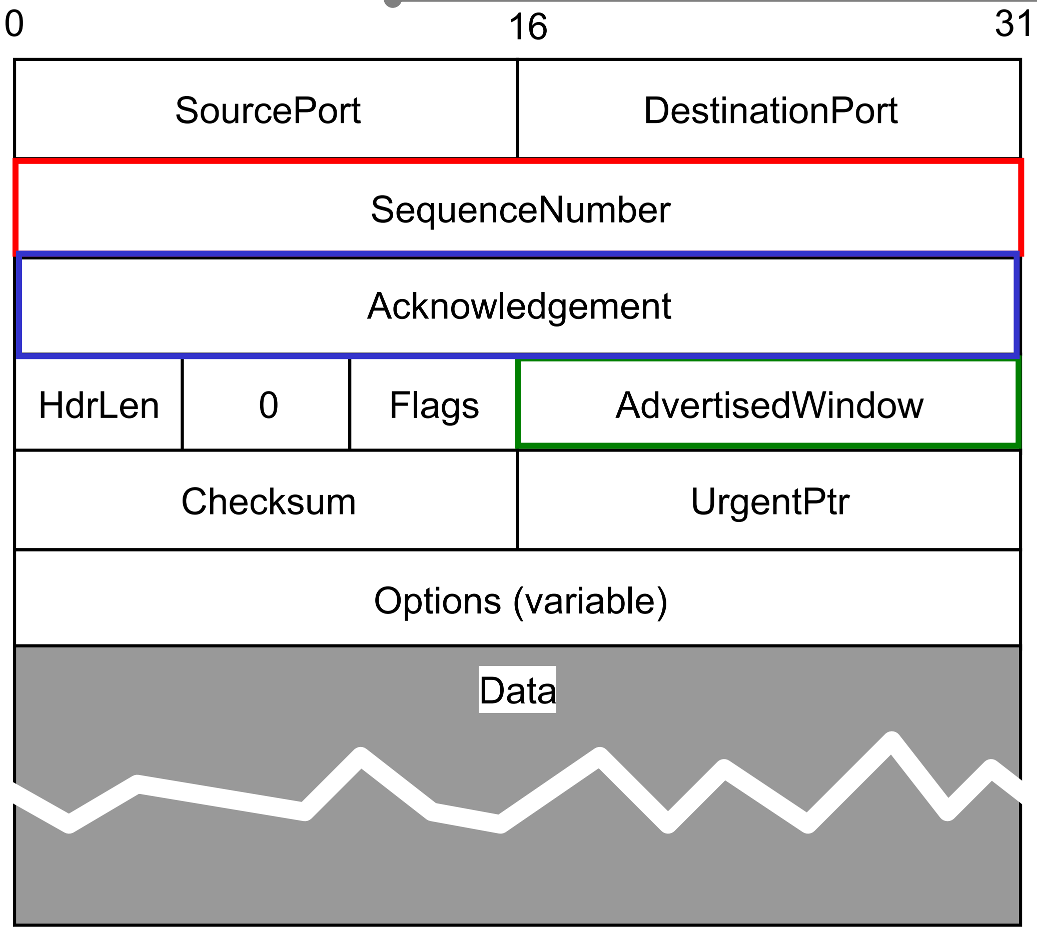
SequenceNumber – refers to the outgoing data.

It gives the sequence number of first byte of data in segment

AcknowledgementNumber and **AdvertisedWindow** carry information about the flow of data from the receiver side.

Acknowledgement Number specifies the sequence number of data that has been received.

Advertised window specifies how much buffer space is available for more data at the destination.



0 16 31

SourcePort | DestinationPort

SequenceNumber

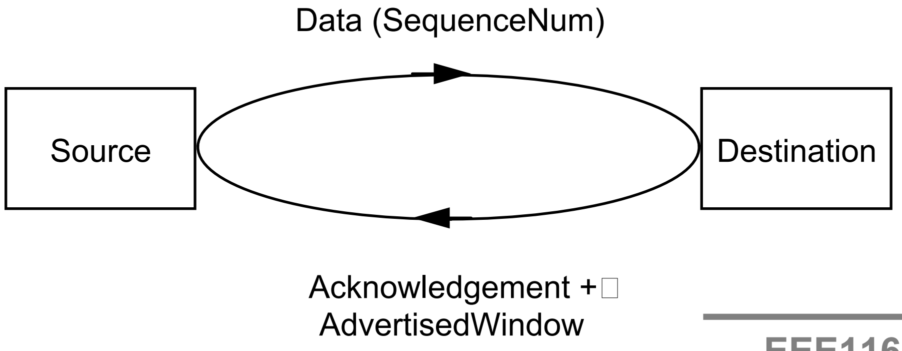
Acknowledgement

HdrLen | 0 | Flags | AdvertisedWindow

Checksum | UrgentPtr

Options (variable)

Data



Data (SequenceNum)

Source

Destination

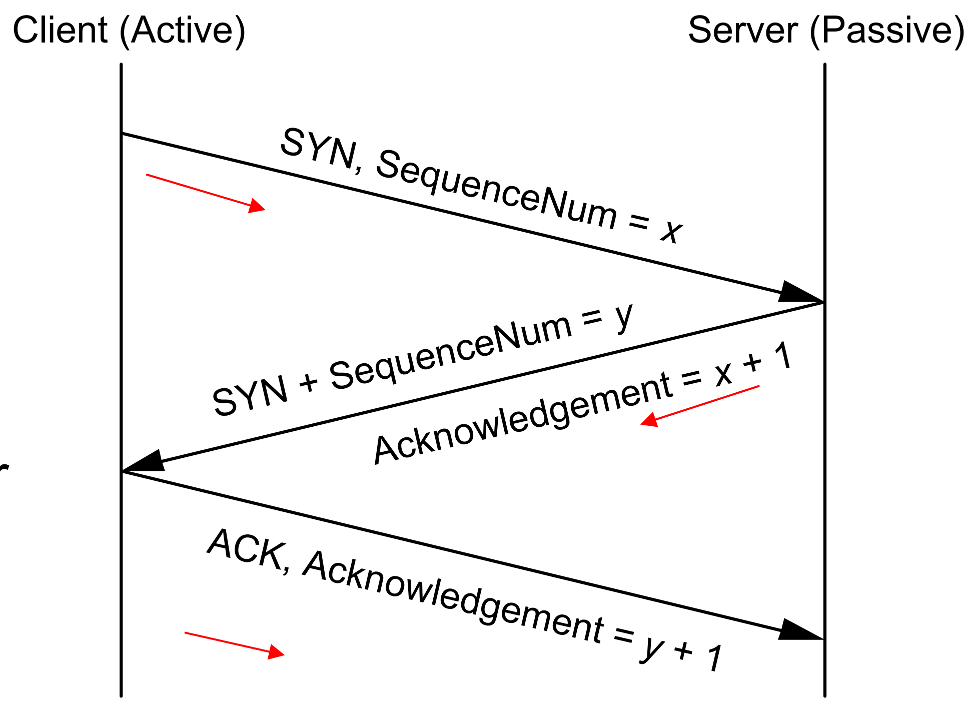
Acknowledgement + AdvertisedWindow

Establishing Connection *Three-Way Handshake*

- Client (active participant) sends a segment to server (passive) with the initial sequence number and flags = SYN

- Server responses (if available) with a segment acknowledging the client with flags = ACK and states its own sequence number and flags = SYN

- Client responds to acknowledge to server sequence number



Note:

- Returned sequence numbers are incremented as it represents the next expected segment.
- Both x and y generated randomly to prevent confusion between connections.

Breaking a connection is more complicated ☹

- Sequence numbers ensure that no fragment is missed (if it is not received within a certain time, the destination asks for it to be sent again) and that fragments can be assembled in the correct order at the destination.

- The time to send a complete packet is set by the **Round Trip Time** - as the source has to receive an acknowledgement of the previous packet before sending the next. With a RTT of 300 ms to the USA, this could make the Internet very slow. **(E.g., Tutorial Problem - Q18 –ii)**

- Hence, send a burst of packets up to the ***advertisedwindow*** (the maximum packets the destination can cope with) but still respond to individual acknowledgements. **(E.g., Tutorial Problem** - **Q18 –iii)**

- Remember the intervening network is shared with lots of other traffic - how do you maintain maximum flow of data without causing congestion?
    - by estimating the effective window size available at the receiver from the information received.

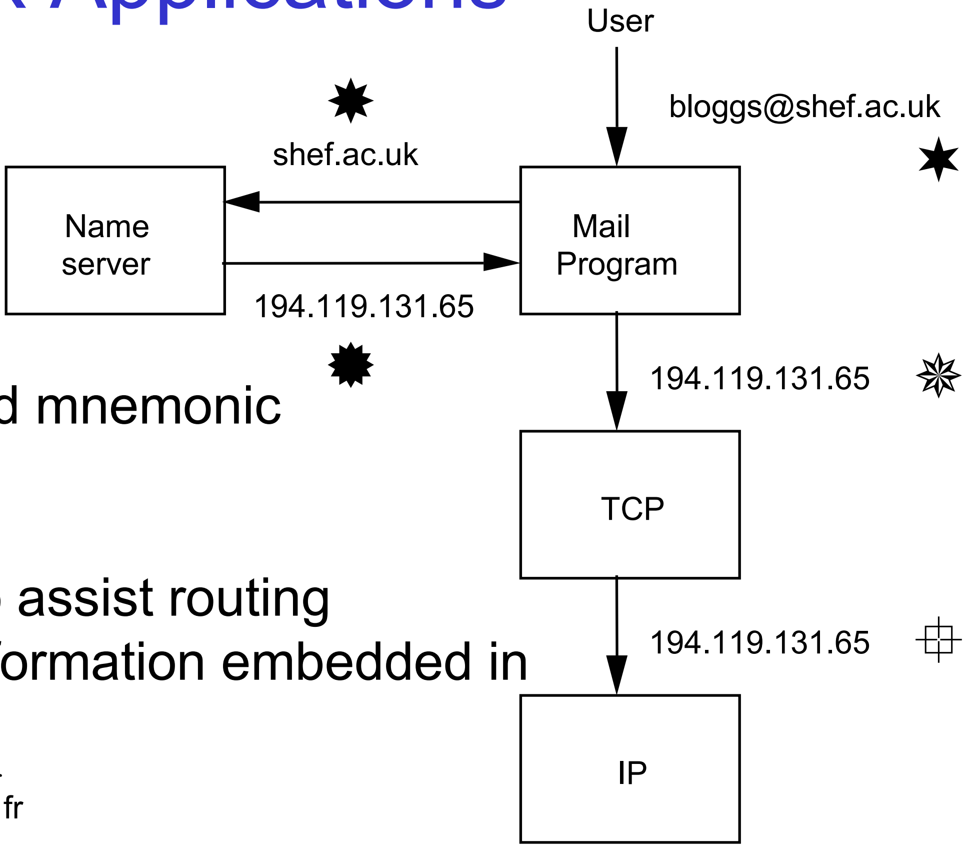
Network Applications

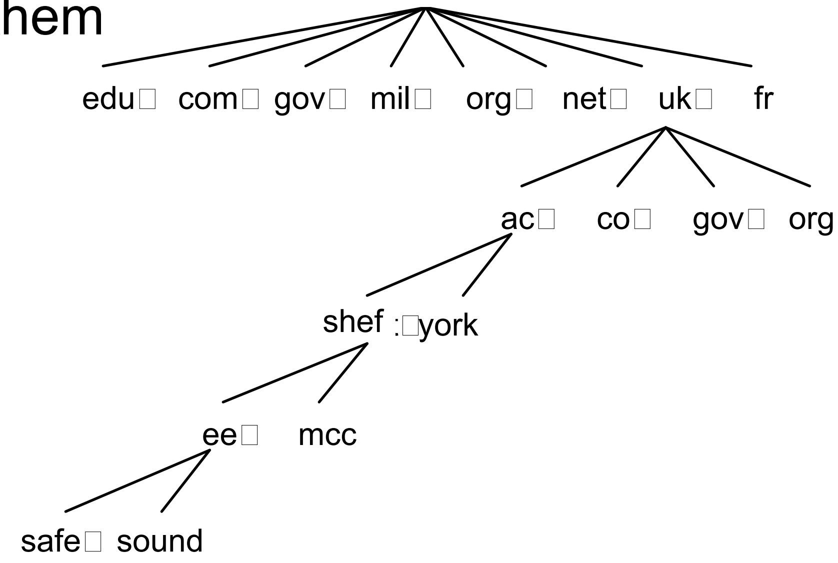
Domain Name System

Host names are for our benefit
All routing uses IP addresses

Names are of variable length and mnemonic
Addresses are of fixed length

Names contain no information to assist routing
Addresses often have routing information embedded in them





Domain Hierarchy

Network Applications

Network applications are the *reasons* for the existence of a computer network.

An application-layer protocol defines how an application's processes pass messages to each other

Applications:

The Web

Hyper text transfer protocol (HTTP),

File transfer

File transfer protocol (FTP)

Email

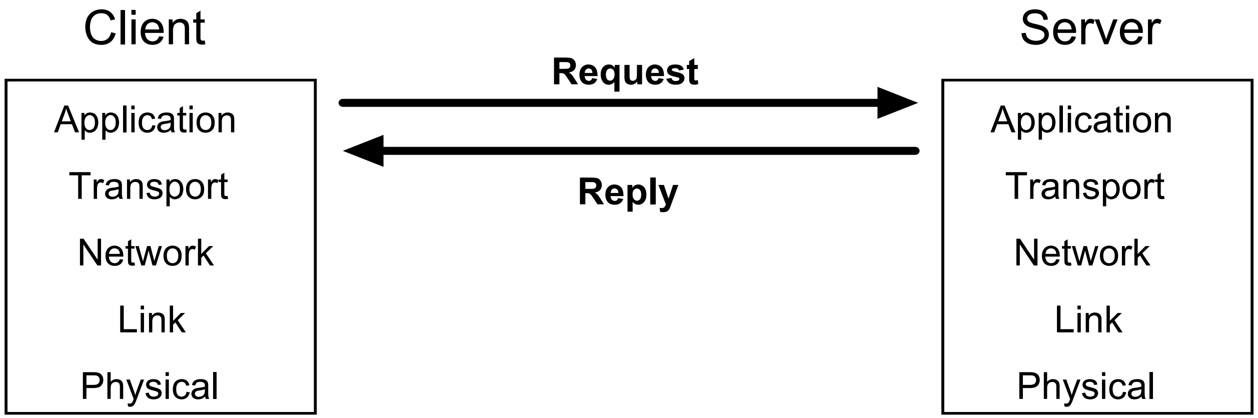
Sending

Simple Mail Transfer Protocol (SMTP)

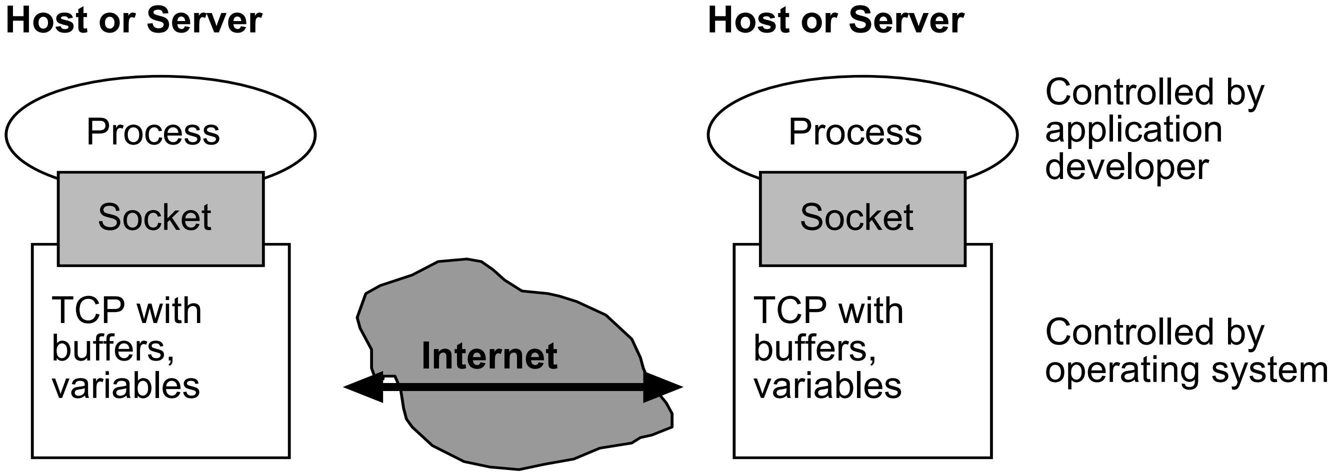
Mail access

Post Office Protocol - version 3 (POP3)

Internet mail access protocol (IMAP)



The host that initiates a connection is labeled the client


Host or Server
Host or Server
Process
Socket
TCP with buffers, variables
Internet
Process
Socket
TCP with buffers, variables
Controlled by application developer
Controlled by operating system

Socket is the interface between the application layer and the transport layer
Also called the *API* (*Application programmers' interface*)

Multimedia streaming over Internet

Internetworking: revision

Host-to-host communication & process-to-process communication.

Relevant protocols: IP/UDP/TCP

Reliable vs. Unreliable communication.

Connectionless vs. Connection-based communication.

Congestion control?

Reliability vs. timely delivery?

Which is more important for video streaming?

Transport Layer Protocols (Summary)

Transmission Control Protocol (TCP)

- Reliable delivery (no loss or no duplication of packets)
    - Using Sequence numbers and acknowledgements
- Connection oriented (Full duplex stream transport)
- Congestion control mechanisms

- Reliability requires – retransmissions – This causes delays.
    - Not acceptable for real-time streaming.

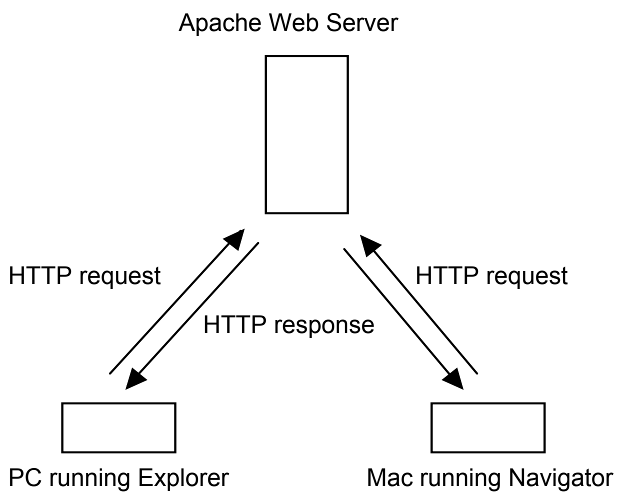
User Datagram Protocol (UDP)

- Unreliable (no guaranteed delivery, duplication of packets).
- Unordered datagram service.
- Messages are sent with minimum protocol mechanism.
- User have more control over when data is sent.

Reliability vs. Timeliness

Multimedia over Internet

1. File transfer – HTTP or FTP
   Involves TCP

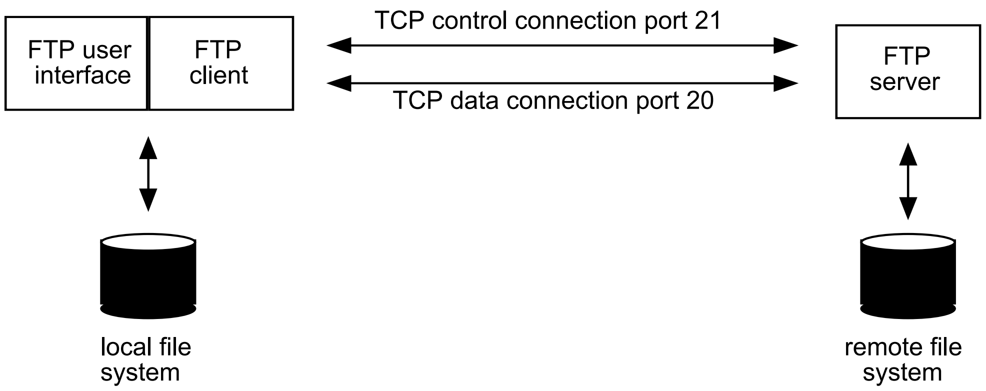


Transmit time

T_{trans} = (Data size) / (Channel data rate)

Playing back time

T_{play}=(Data size) / (Encoded data rate)



For this type of transmissions

Uaually T_{trans}> T_{play}

Not suitable for real-time applications.

Multimedia over Internet

2. TCP based streaming

- Use of buffers
- Start playing out from a buffer while receiving data.

- Transmit time from the start of playing out the buffer

T_{trans}= ((Data size) – (buffered data size)) / (Channel data rate)

Uaually $T_{trans} < T_{play}$

- Congestion causes reduction in channel data rate
- This leads to pauses in play out.

3. UDP based streaming

- Playing out from buffer while data is being received
- Start play out after small amount of data is in the buffer (Small delay)
- $T_{trans} = T_{play}$

Therefore the preferred method is UDP/IP.

Requirements of Multimedia over Internet

1. Sequencing – To order packets at the receiver
- To detect packet losses.

2. Synchronization
– Intra-media (timing information for play out of successive packets)
- Inter-media (to synchronize multiple media streams)

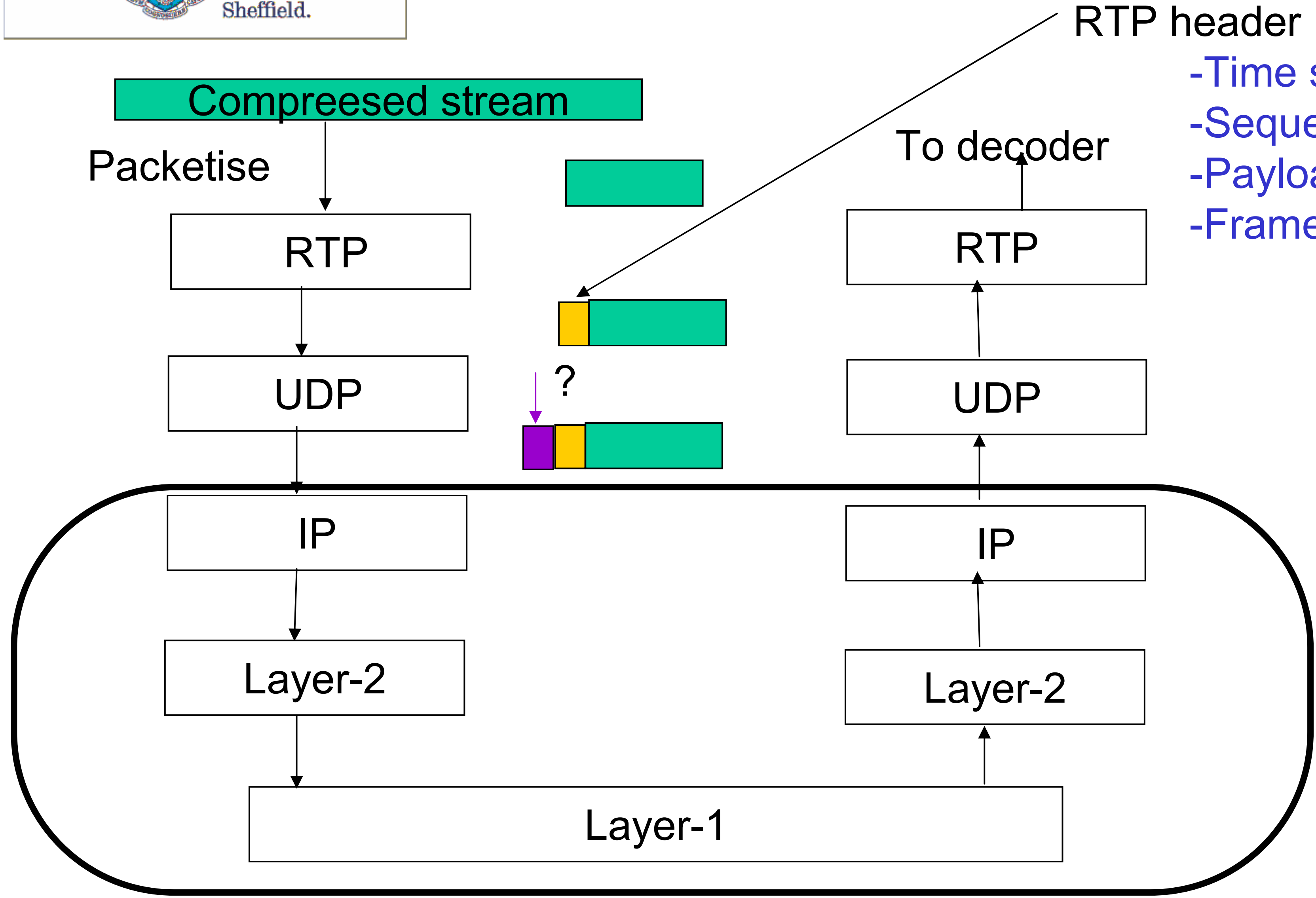
3. Payload identification
- for use of different codecs for different data types.

4. Frame indication
- indicates start and end of frames.

Solution: Streaming Protocols (to address these requirements)
1. Real-time Transport Protocol (RTP)
2. Real-time Transport Control Protocol (RTCP)

- Defined in a generic way to be independent of underlying transport layer.

RTP



RTP header

- -Time stamping
- -Sequence numbering
- -Payload ID
- -Frame indication

RTP

The requirements for Multimedia over Internet are fulfilled by

-Mixers

-Combining several media streams to one new stream

-Translators

-Change encoding bit rates (transcoding for low bit rates)

RTP/UDP facilitates transport of multimedia data.

However, parameters like packet loss rates, available bandwidths and delays are required for de-packetisation and decoding processes. i.e., need some feedback on network performance.

Real Time Control Protocol (RTCP)

- provides feedback on quality of data distribution.

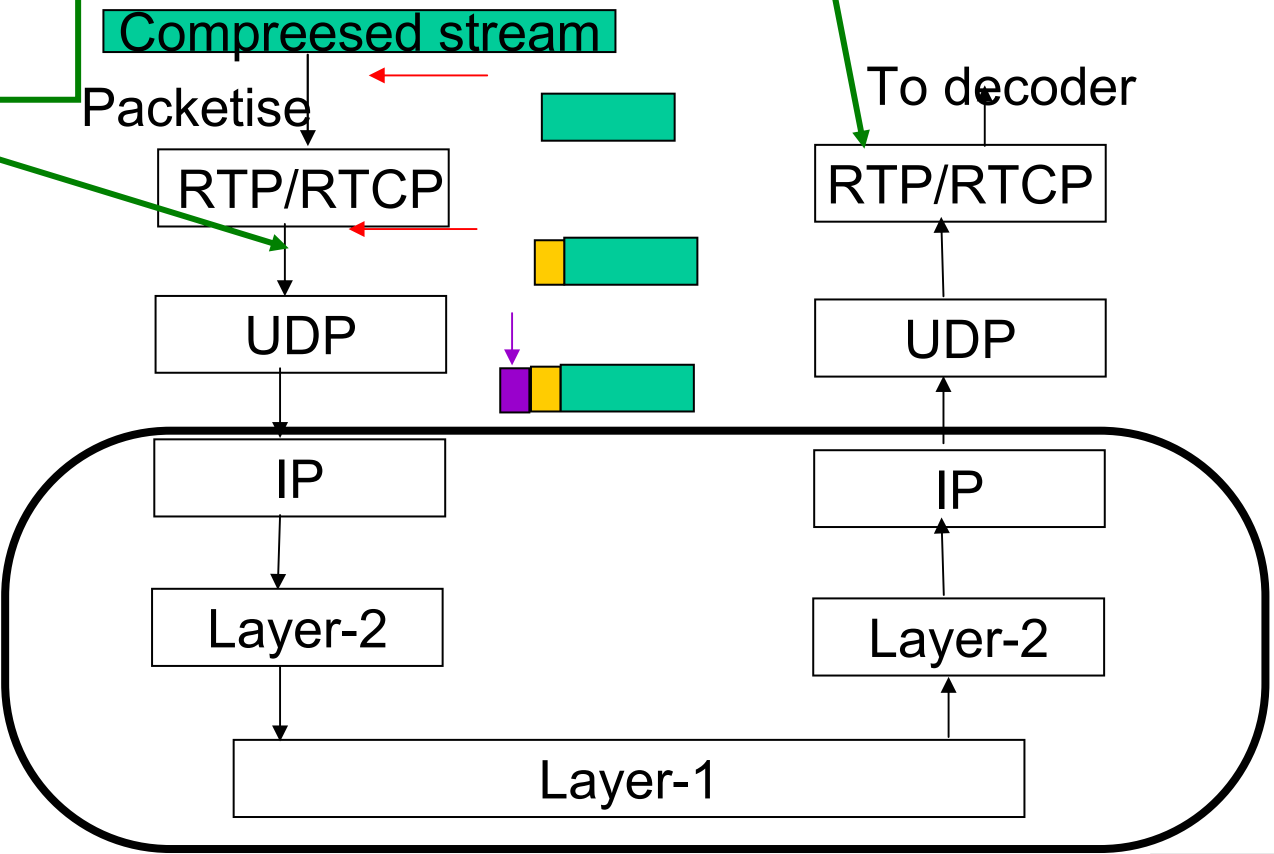
RTP & RTCP

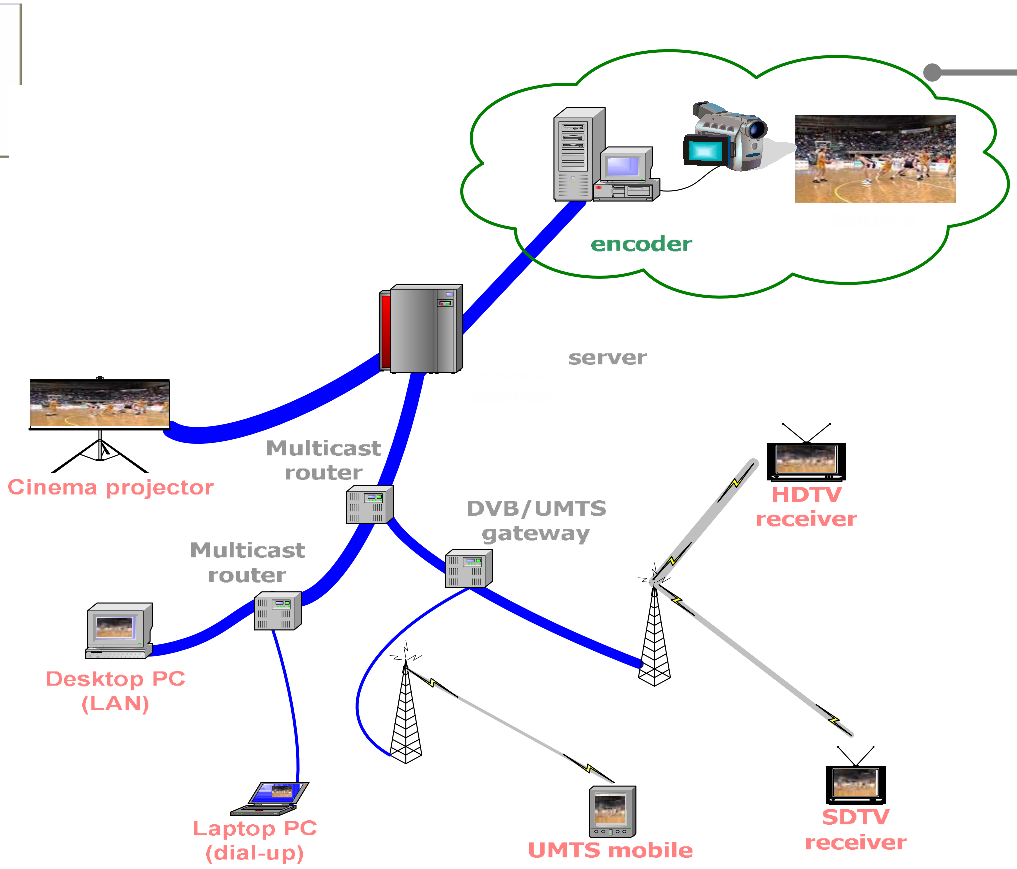
RTP header is used for
-Reorder packets
-identify lost packets
-detect payload type
-detect participants
-packets to frames

RTP header
-Time stamping
-Sequence numbering
-Payload ID
-Frame indication

Transport
RTP/UDP/IP

Control
RTCP/UDP/IP




encoder
server
Cinema projector
Multicast router
Multicast router
DVB/UMTS gateway
HDTV receiver
Desktop PC (LAN)
Laptop PC (dial-up)
UMTS mobile
SDTV receiver

The End

Find the IP address of a computer connected to the University network and determine the class of IP address.

- How?
  - U:\>ipconfig
    - Windows IP Configuration
    - Ethernet adapter Local Area Connection:
    - Connection-specific DNS Suffix . : shef.ac.uk
    - IP Address. . . . . : 143.167.62.11
    - Subnet Mask . . . . : 255.255.255.0
    - Default Gateway . . . . : 143.167.60.254

- Address Class ?

  [redacted]

  - Network address?

    [redacted]

  - How many possible connections?

    [redacted]

Before the exam ….

1. Revise your lecture notes
2. Practice your tutorial questions.
3. Use revision quiz to test what you have already revised.
4. Attempt past exams.

the exam ….

EEE116 Exam - 2 hours

Answer 3 questions (out of 4)

DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING
Autumn Semester 2008-2009 (2 hours)
Multimedia Systems 1

Answer **THREE** questions. **No marks will be awarded for solutions to a fourth question.** Solutions will be considered in the order that they are presented in the answer book. Trial answers will be ignored if they are clearly crossed out. **The numbers given after each section of a question indicate the relative weighting of that section.**

Use the relative weighting of a section to determine how much time you spend for that section.

the exam ….

1. Read the exam paper
2. Choose the questions you want to attempt.
3. Allocate time for each question.

4. Read the question carefully
5. Answer what is asked in the question

   Take notice of the phrases

   ‘What are’
   ‘define’
   ‘list’
   ‘explain briefly’
   ‘give AN example’
   state
   describe etc..
6. For computational answers, show all steps : you can get marks for intermediate steps even if the final answer is wrong.
7. **Don’t forget units**.

Contact Details:

Room F176 Phone: ext. 25893

Email: c.abhayaratne@sheffield.ac.uk