



**The Abdus Salam
International Centre for Theoretical Physics**



2057-15

**First Workshop on Open Source and Internet Technology for
Scientific Environment: with case studies from Environmental
Monitoring**

7 - 25 September 2009

TCP/IP Networking

Abhaya S. Induruwa
*Department of Computing
Canterbury Christ Church University
North Holmes Road
Canterbury CT1 1 QU
U.K.*



An Overview of TCP/IP Protocol Stack

Abhaya Induruwa
Department of Computing
Canterbury Christ Church University
United Kingdom



TCP/UDP – IP Protocol Suite



Developed in the late 60s early 70s

➤ **Transmission Control Protocol**

➤ TCP – IEN5 → RFC 761 → RFC 793

➤ **Internet Protocol**

➤ IP – IEN2 → RFC 760 → RFC 791

➤ **User Datagram Protocol**

➤ UDP – RFC 768



TCP



- connection oriented
- reliable
- stream transport protocol



How does TCP achieve this?



- connection oriented
- uses RST, SYN, ACK, FIN flags to establish, maintain and tear down connections
- checksum for both header and application data
- sequence numbers to put the packets in the correct order
- acknowledgement numbers to let the sender know that a packet has been received
- retransmission if packets are not acknowledged.



UDP



- connection-less
- inherently unreliable
- fast
- message transport protocol



Why do we need UDP?



- inherently unreliable but fast
- no connection establishment
- no acknowledgements
- no retransmissions
- little transport protocol overhead (8 bytes only)
- messages are delivered on a best effort basis
- checksum is used to detect errors. If errors are detected, UDP can discard or deliver to the application with a warning



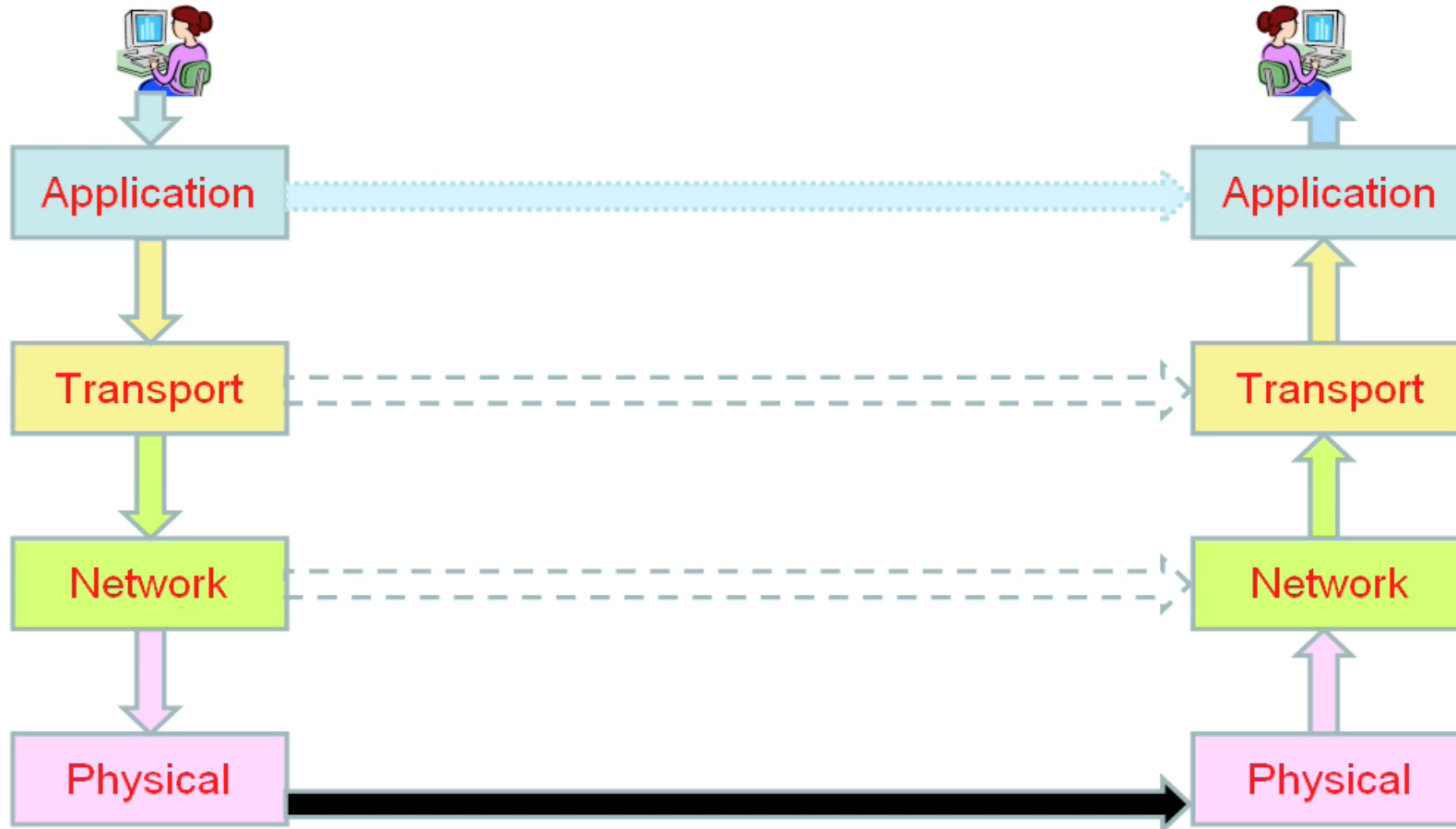
The role of IP



- Best effort delivery
- based on IP addresses
- Inter-domain and Intra-domain routing
- Header checksum to ensure correct header on delivery
- Routers can drop packets resulting packet loss
- IP does not do anything about it.

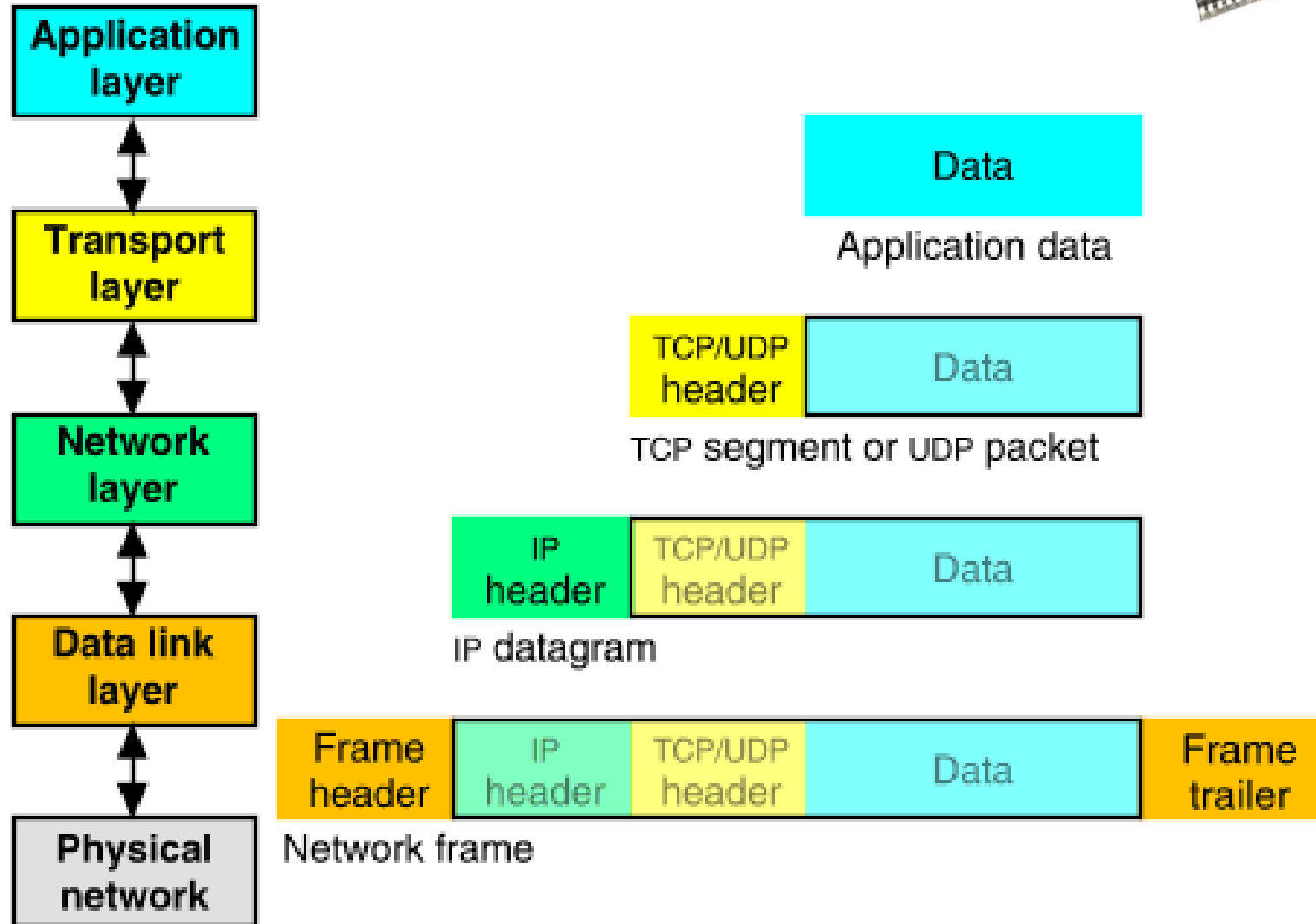


TCP/IP in action



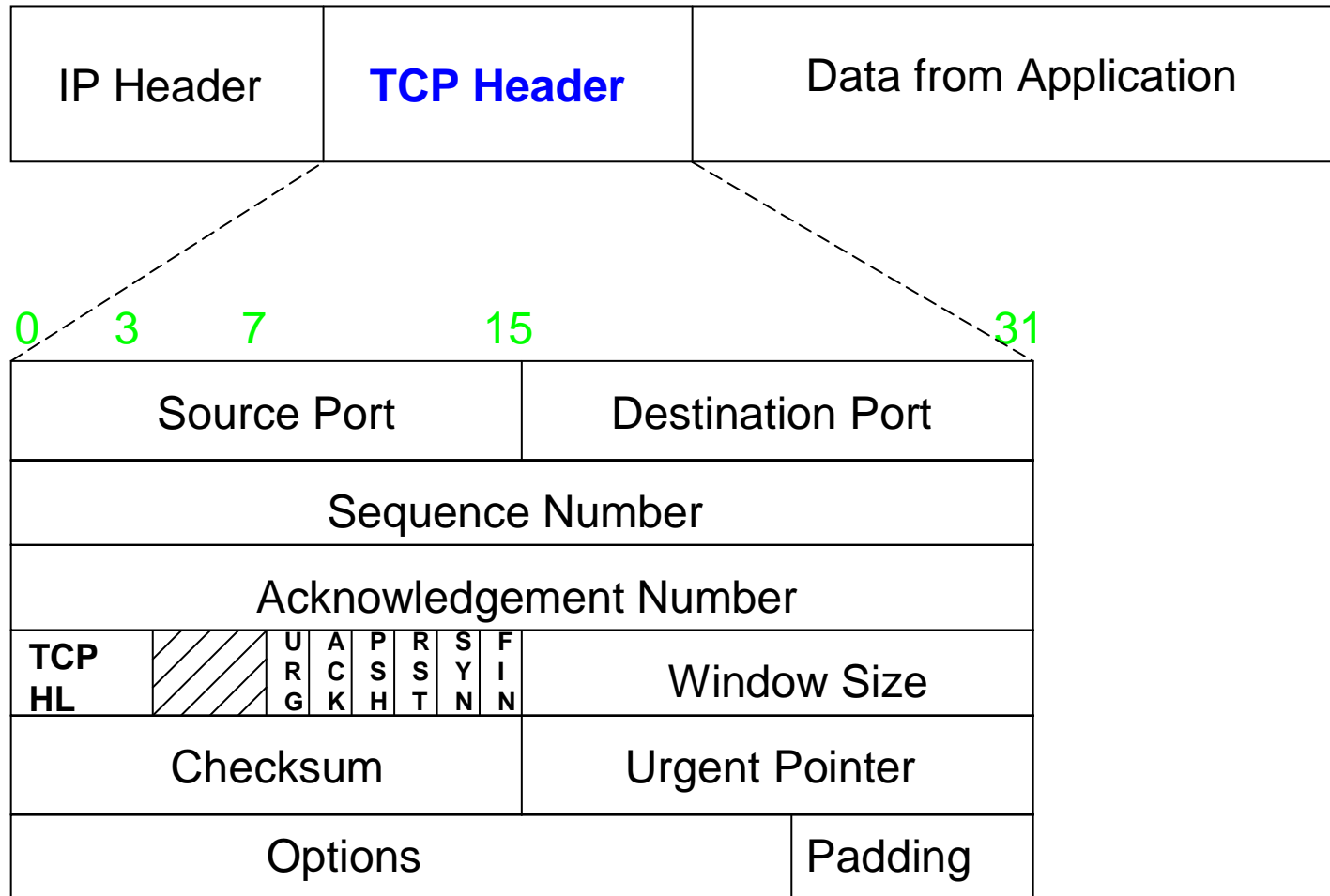


Data Encapsulation



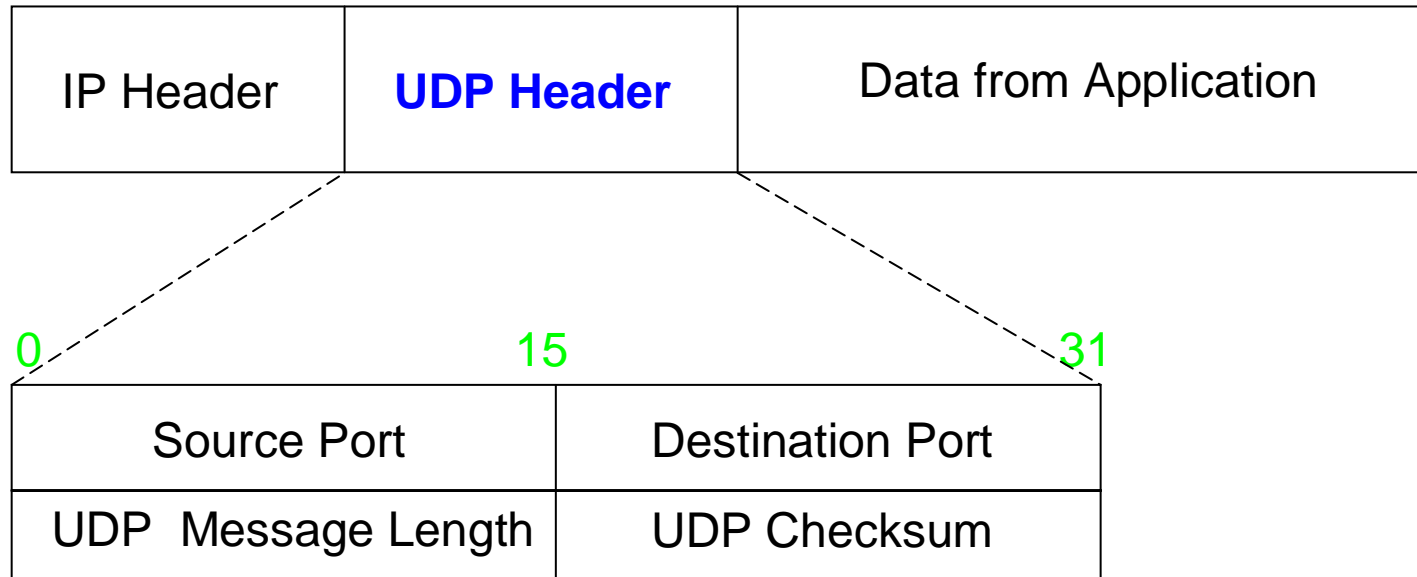


TCP Segment Header





UDP Packet Header





What is a Port?



- A port is a transport address at which processes can listen for connection requests
- 16 bit port address giving 65536 ports
- Well known ports
 - Below 1024 (0 – 1023)
 - Provide standard services
 - Known port numbers?



Well Known Port Numbers



- Port numbers below 1024
- Reserved for standard services

FTP	21	File Transfer	TCP
SSH	22	Secure Shell Login	TCP
Telnet	23	Remote login	TCP
SMTP	25	Simple Mail Transfer	TCP
DNS	53	Domain Name Service	TCP/UDP
TFTP	69	Trivial FTP	UDP
Finger	79	user lookup	TCP
HTTP	80	World Wide Web	TCP
POP3	110	Post Office Protocol V3	TCP
SNMP	161	Network Management	UDP
SNMP Traps	162	Network Management Traps	UDP
HTTPS	443	Secure HTTP	TCP
IMAP4	993	Message Access over TLS/SSL	TCP

<http://www.iana.org/assignments/port-numbers>



Registered Port Numbers

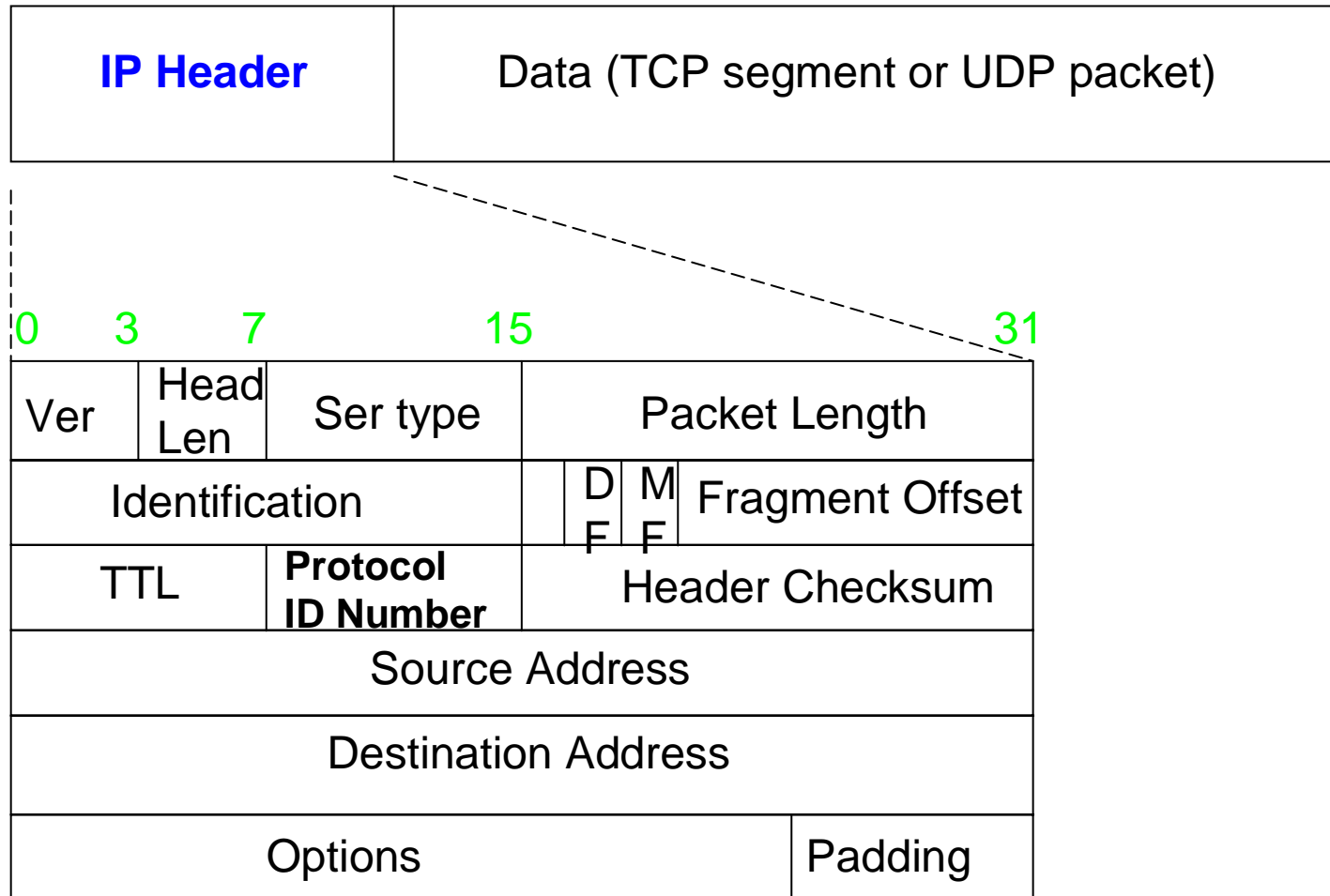


- Port numbers from 1024 to 49151
 - L2TP 1071 Layer 2 Tunnelling Protocol
 - PPTP 1723 Point to Point Tunnelling Protocol
- Port numbers above 49151 are unassigned.
- These are for Dynamic/Private services.

<http://www.iana.org/assignments/port-numbers>



IPv4 datagram





IPv4 Header



- IP Header has a 20 byte fixed part
- A variable length optional part
- Padding is used to align with a byte boundary
- IP Header is transmitted in big endian order. ie. From left to right with the high order bit of the version field going first
- SUN Sparc is big endian architecture
- Intel Pentium is little endian architecture



IPv4 Header (contd ...)



- Fragmentation is controlled by two bits
 - DF – Do not Fragment
 - instructs the router not to fragment this packet
 - MF – More Fragments
 - when set to one indicates that this packet is fragmented and this fragment is not the last one
- Fragmentation Offset
 - 13 bit field to specify in bytes the offset of the data field of this packet from the starting point of the source packet.
- Identification
 - 2 byte field used to identify the packets created as a result of fragmentation. All fragments of the same packet must have the same value in this field.



IPv4 Header (contd ...)



- Protocol Identification Number

- Specify the higher layer protocol for which the payload is intended (using Assigned Internet Protocol Numbers).

1 – ICMP

6 – TCP

8 – EGP

17 – UDP

41 – IPv6

89 – OSPFIGP

115 – L2TP

255 – Reserved

<http://www.iana.org/assignments/protocol-numbers>

- Header Checksum

- 2 bytes; calculated for header only
- Since some header fields get changed in the course of packet transmission across the network, the header checksum is recomputed at each router node.



IPv4 Addresses



- 32 bit source address
- 32 bit destination address
- Completely different from any hardware address (eg. 48 bit Ethernet Address)
- Every node (host or router) on an internetwork should have at least one unique IP address
- Hosts connected to more than one network will have a different IP address to every connection (multi homed networks)



IPv4 Address Classes



- 5 different classes – known as Classful

		0	7	15	23	31
Class	A	0	Net ID	Host ID		
Class	B	1 0	Net ID		Host ID	
Class	C	1 1 0	Net ID		Host ID	
Class	D	1 1 1 0	Multicast address			
Class	E	1 1 1 1 0	Reserved for future use			





IPv4 Addressing



	Network ID	No of networks	No of Hosts
Class A	1-127	126	16, 777,214
Class B	128-191	16,384	65,534
Class C	192-223	2,097,151	254



IPv4 Address Ranges



	Address Range
Class A	1.0.0.0 - 127.255.255.255
Class B	128.0.0.0 - 191.255.255.255
Class C	192.0.0.0 - 223.255.255.255



IPv4 Addresses



- A total of 4 Giga addresses
- Usually written in dotted decimal notation
- IP address space is managed by ICANN
(Internet Corporation for Assigned Names and Numbers)
- Every Internet Host and Router has a unique IP address which encodes its network number and host number



Special IPv4 Addresses



0.0.0.0 This host on this network – used during boot up

255.255.255.255 Direct Broadcast on LAN

127.x.y.z Local loopback address

- eg. 127.0.0.0
- Data sent to this address will not be transmitted into the network
- Packets received with this address are returned to the upper layer protocol of the same computer
- Useful for testing programme



Network Layer Supporting Protocols



- IP is the main network layer protocol used for data transfer.
- Several control protocols exist to assist IP
 - ICMP – RFC 792
 - ARP – RFC 826 (IP to hardware address)
 - RARP – RFC 903 (H/W to IP address)
 - BOOTP – RFCs 951, 1048 and 1084
 - DHCP – RFCs 2131 and 2132



Network Sockets



- Socket is an end point of a communication channel
- identified by a socket address
- in IPv4 socket addresses are 48 bits
- not to be confused with 48 bit Ethernet addresses
- obtained by combining IP address (32 bits) with port address (16 bits)
- TCP sockets and UDP sockets



TCP/IP Client Server Example



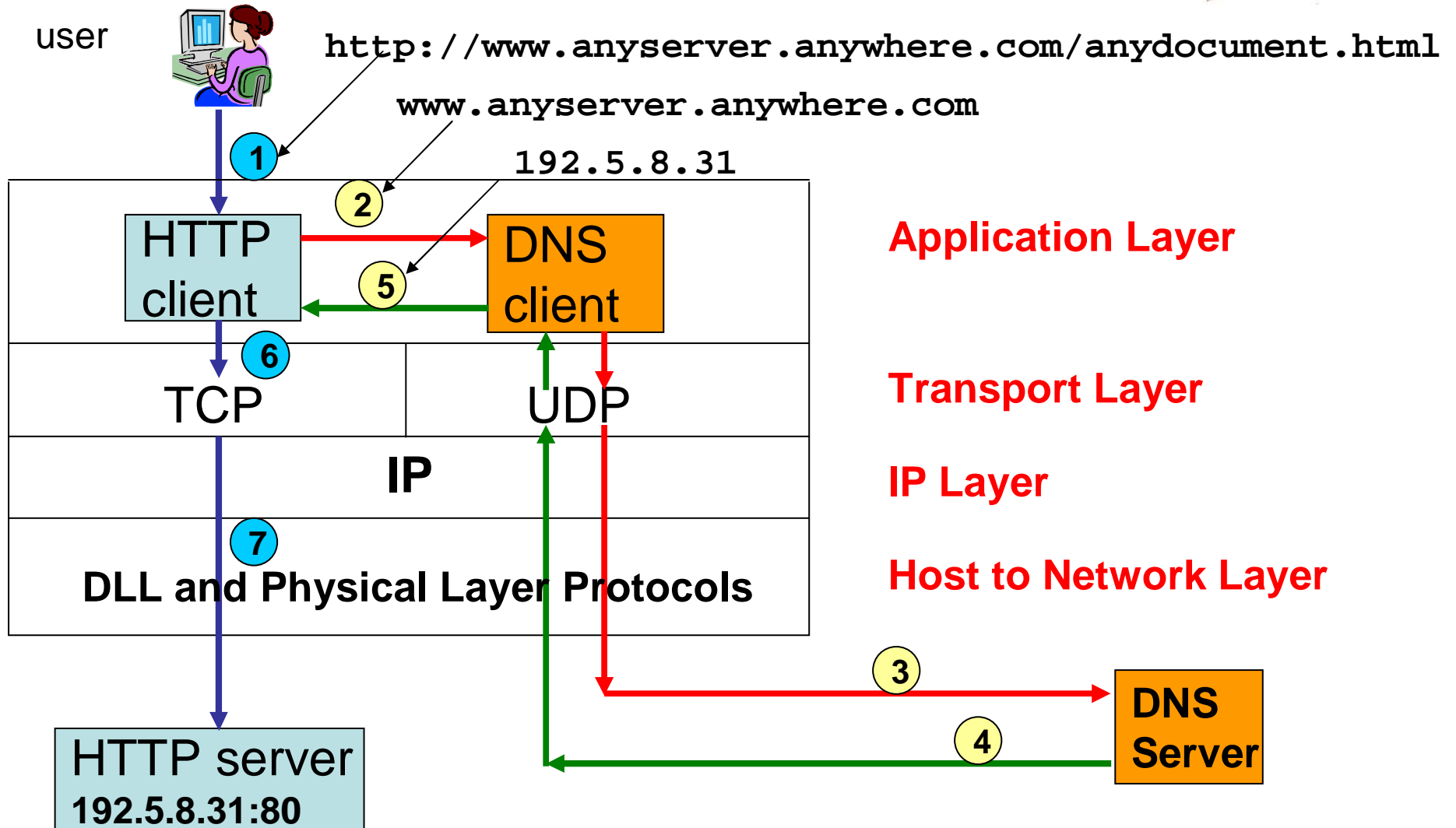
A user enters a URL on a browser to access a web server.

How many different protocols are used?

- **HTTP**
- **DNS**
- **TCP**
- **UDP**
- **IP**
- **Physical layer, eg. Ethernet**



HTTP Client-Server Application





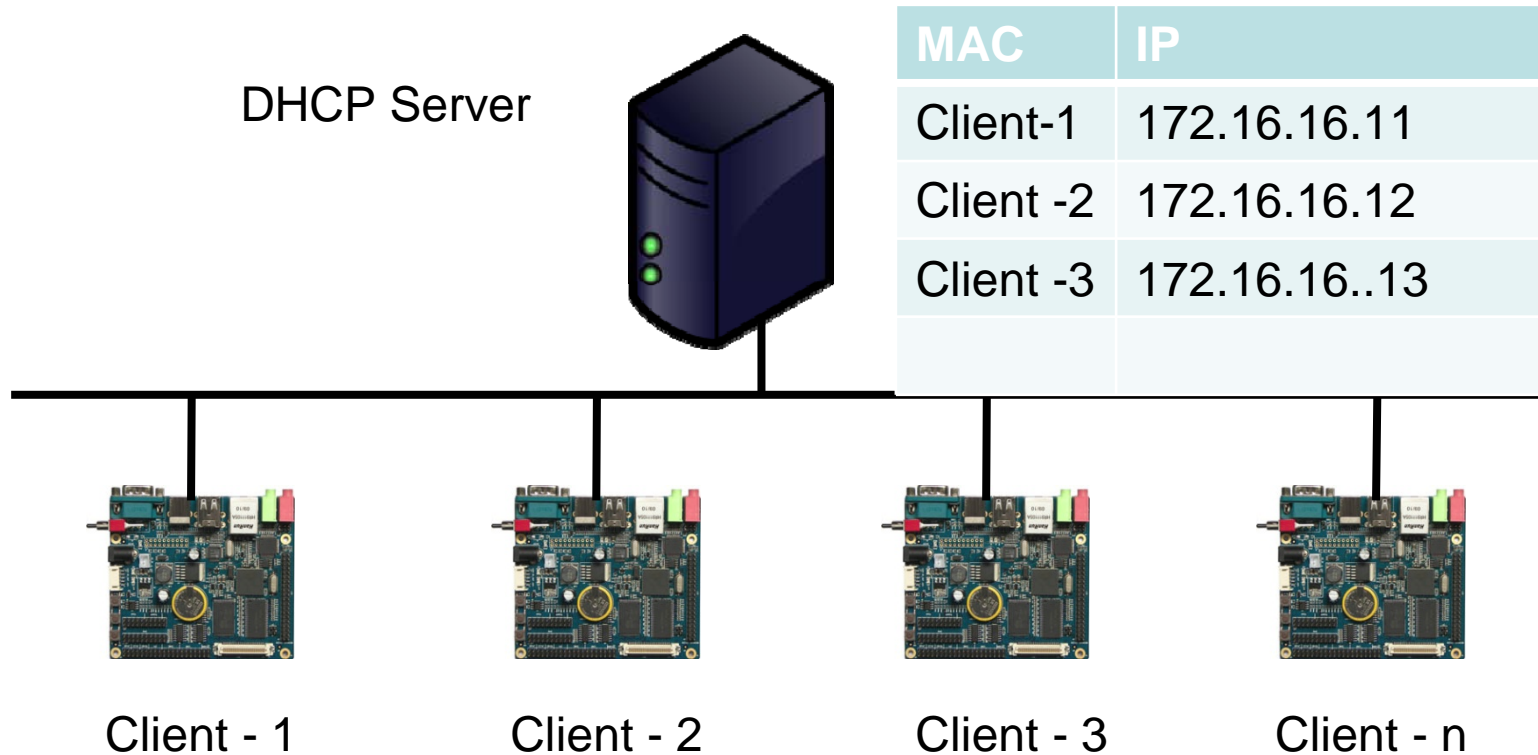
DHCP Protocol



- Dynamic Host Configuration Protocol – RFC 1541 superseded by RFC 2131
- on booting the device broadcasts its MAC address
- the DHCP server listens to these broadcasts and replies with an IP address
- IP address can be allocated from a pool – hence dynamic
- IP address can also be fixed – kept in a table



DHCP Protocol



Clients get the IP address corresponding to their MAC address from the table.



TFTP Protocol



- Trivial File Transfer Protocol – RFC 1350
- uses UDP over port 69
- no security
- Typically used by disk-less workstations
- used by embedded systems, such as Geode and ARM boards
- On booting the device sends a DHCP request for the IP address
- Once it gets its IP address, it sends a RRQ using TFTP to read the file



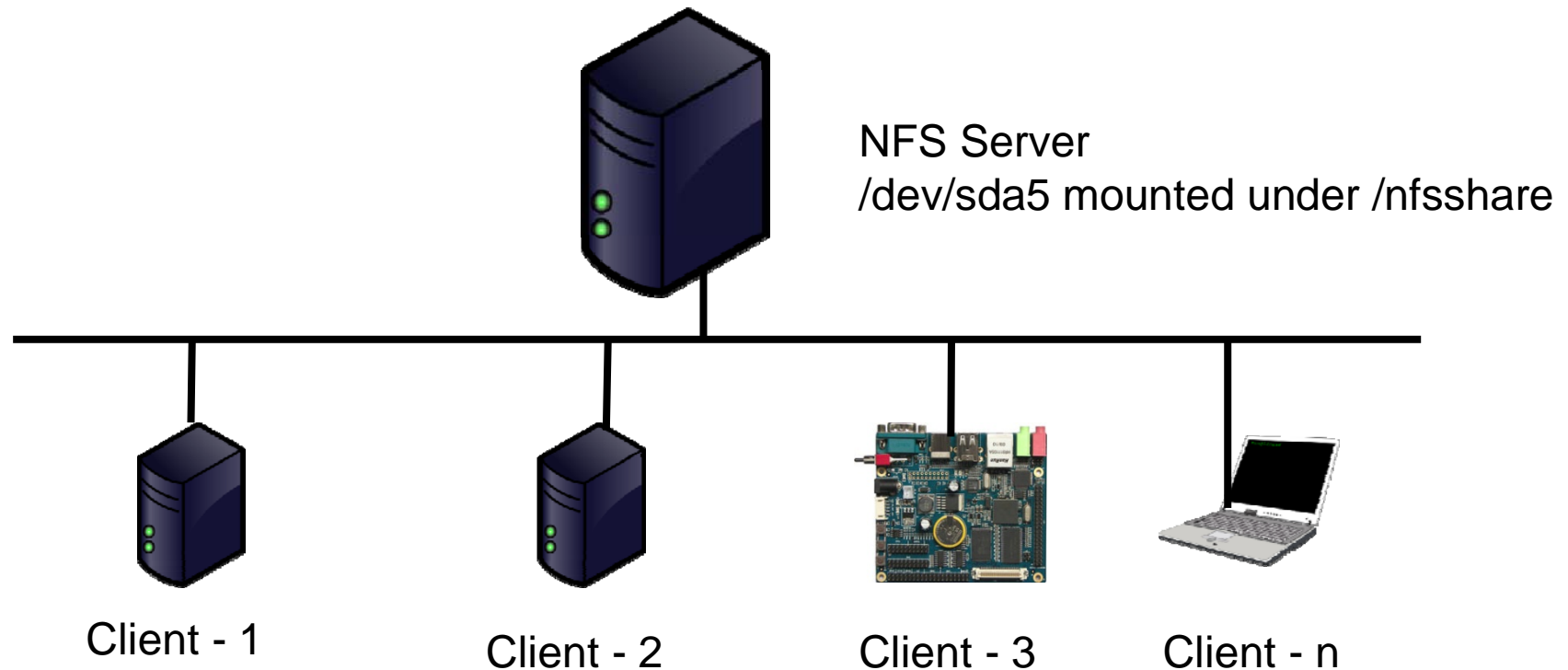
NFS Protocol



- **Originally developed by Sun Microsystems in 1984**
- **allows a client to share files and directory over a network in a manner similar to how local storage is accessed**
- **used UDP**
- **NFSv2 released as Open Source in 1989 – RFC 1094**
- **Now NFSv4 (2003), version developed by the IETF – TCP based**
- **Next revision is 612 pages!**



NFS Protocol



NFS clients can mount `/nfsshare` and use as if they are accessing local hard disk



Network Programming



- Network programming can be done in many popular languages
- C, C++, Java, C#, VB.NET all have APIs to programme network functions
- PHP is an alternative



Hello World in PHP



```
<!--
```

This example shows how PHP code is embedded in HTML code. The browser will display the header from the HTML part of the code. Then it will display the output by executing PHP code.

```
-->
```

```
<html><body><h1>Hello World Example</h1></body></html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
// echo will display the string argument
```

```
echo "Hello World";
```

```
?>
```

```
</body>
```

```
</html>
```



Network Programming in PHP



```
<!--
```

This example shows how PHP programming can be used to deal with IP addresses and host names.

```
-->
```

```
<html><body><h1>IP Address Example</h1></body></html>
```

```
<?php
```

```
// get host by address
```

```
$hostaddress = gethostbyaddr('127.0.0.1');
```

```
print $hostaddress;
```

```
print "<p>The host name is: $hostaddress". "</p>\n";
```

```
// get host by name
```

```
$hostip = gethostbyname("www.ictp.it");
```

```
print "The host address is: $hostip". "<p/>\n";
```

```
// and re-check
```

```
echo gethostbyaddr("$hostip");
```

```
echo "<br/>";
```

```
// convert dotted decimal IPv4 address to IP proper address
```

```
$longip = ip2long($hostip);
```

```
print ("The long address is: $longip". "<p/>\n");
```

```
// convert to packed notation
```

```
$packedip = inet_pton($hostip);
```

```
printf ('%$s %$b', "The packed address is: " $packedip);
```

```
echo inet_ntop($packedip);
```

```
echo long2ip(-1939271547);
```

```
?>
```



Network Programming in PHP



Some of the available PHP commands:

`checkdnsrr` – Check DNS records corresponding to a given Internet host name or IP address

`dns_check_record` – Alias of `checkdnsrr`

`dns_get_mx` – Alias of `getmxrr`

`dns_get_record` – Fetch DNS Resource Records associated with a hostname

`gethostbyaddr` – Get the Internet host name corresponding to a given IP address

`gethostbyname` – Get the IPv4 address corresponding to a given Internet host name

`gethostbynameel` – Get a list of IPv4 addresses corresponding to a given Internet host name

`gethostname` – Get the host name

`getprotobyname` – Get protocol number associated with protocol name

`getprotobynumber` – Get protocol name associated with protocol number



Network Programming in PHP



References:

<http://it2.php.net/manual/en/book.network.php>

<http://it2.php.net/manual/en/ref.sockets.php>

<http://www.devshed.com/c/a/PHP/An-Introduction-to-Sockets-in-PHP/>