CSCU9A1: Systems 5
How networks work: Network protocols

Leslie Smith

Last updated October 2016

UNIVERSITY OF STIRLING

Content

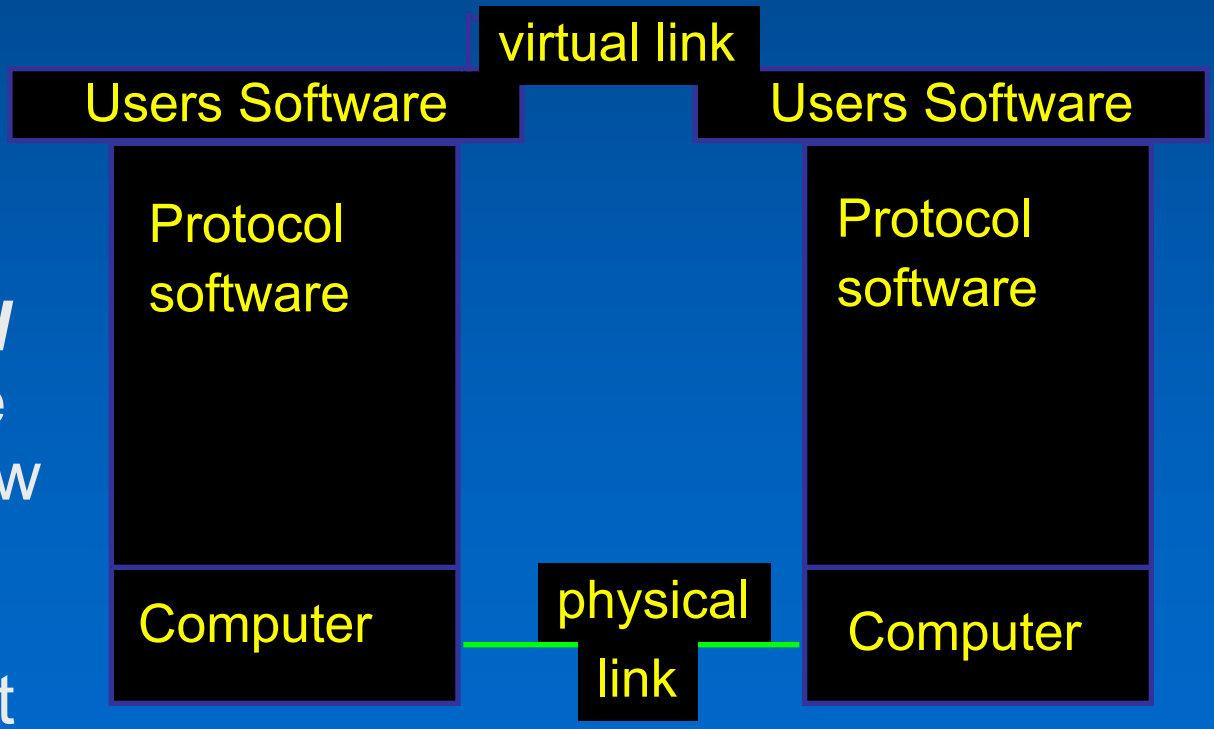
- Protocols: basic concepts
- Packet switching
- TCP/IP
- Higher level protocols
- Portal concept.

Overview: protocols

- At the lowest level, networks consist of nodes sending data to each other.
- At the highest level, networks enable seamless interoperation of distributed systems/ distributed computers
  - Many different types of computer
  - Able to function in the presence of errors
    - Why is this important?
  - Able to support different types of functionality
    - Server-based systems
    - Peer-to-peer systems

Basic concepts

- Networks consist of a number of computers connected to each other
- Software (***protocol software***) uses the connections to allow communication between software running on different machines



Physical and virtual links

Physical link

The actual (physical) connection:

wires,
radio-based connection,
light fibre
infra-red, light, etc

Virtual link

The connection as it appears to the casual user

- being connected to web-sites
- being able to use files etc which are not local to the machine being used
- generically: the connection between processes running on the machines

Packet Switching

- Packet switching vs circuit switching
  - Historical background, old-fashioned telephone systems and copper wire links. Actual mechanical, then electronic switches.
- The basic technology on all digital data networks is packet-switching
  - Data is divided into small packets
  - Each packet is sent individually
  - Received packets are then reassembled to recreate the original data
- Packets can get lost or damaged
  - (damaged? 1 => 0 or 0=> 1)
- Packets may take different routes and arrive out of order.

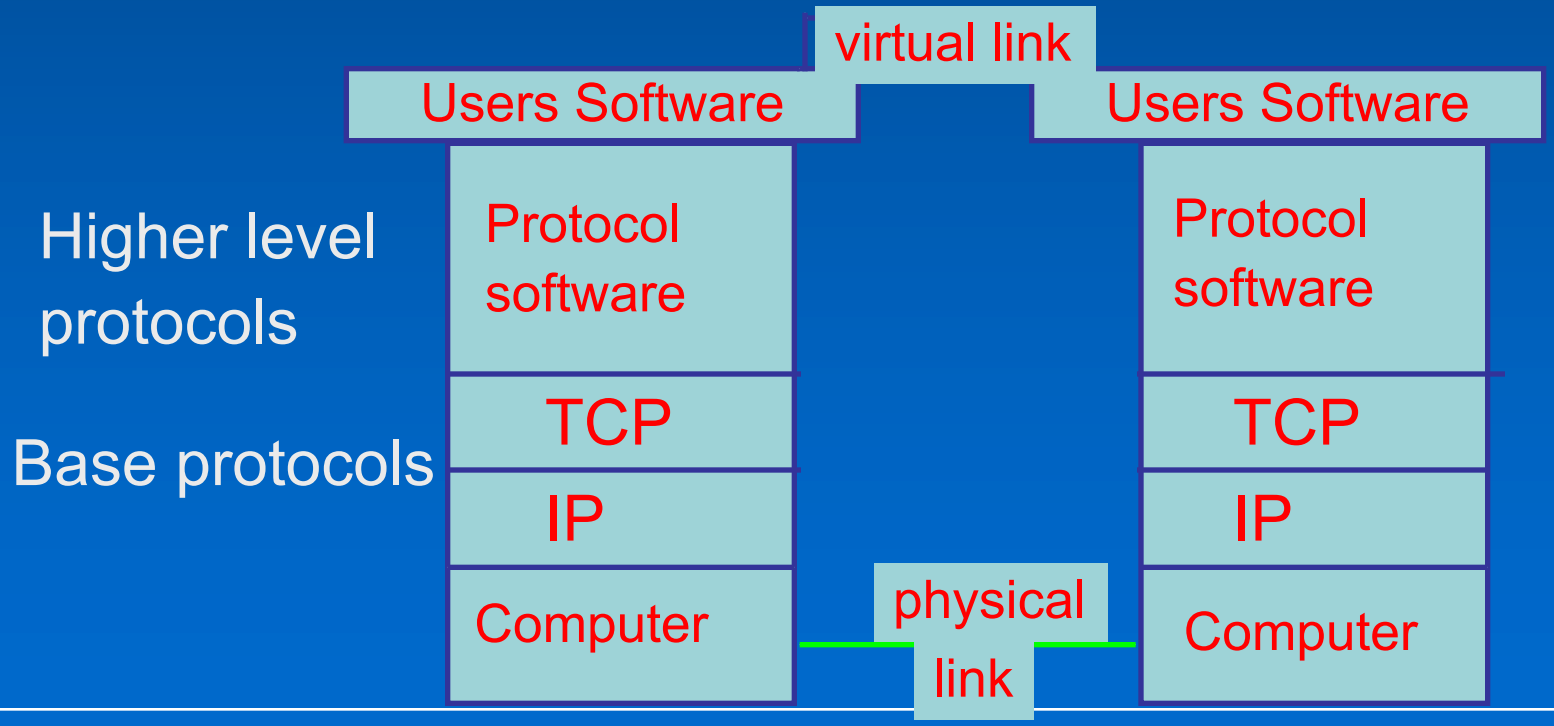
Protocols

- A communications protocol is a set of rules for communications
  - True for both machines and humans
- Humans have their own protocols
- These are generally culturally defined, and describe the way we interact with each other
  - in particular the way we use language and gestures
    - these protocols evolve over time, and are interesting in their own right!
    - We can (generally) cope with people ignoring parts of these protocols
      - though we often don't like it!
        - » (and we train children to use them: say *please* and *thank-you*…)

- Computers need stricter protocols than humans, as they're much less flexible
  - The word *protocol comes from the Greek word protokollon, meaning first glue. It referred to a leaf glued to a manuscript which described its contents.*

Protocols used on the Internet

- The Internet uses **a base protocol set** for machines to talk to each other
  - (a base protocol works directly on the underlying (physical) connection)
- It's called TCP/IP
- Which stands for *Transmission Control Protocol/Internet Protocol*



TCP/IP: the low-level protocols

- TCP/IP provides the capability of sending a sequence of bits from one machine to another
  - It can cope with errors being introduced
  - It can cope with network problems
- It used throughout the (static) internet
  - It is utilised by many other protocols (higher-level protocols)

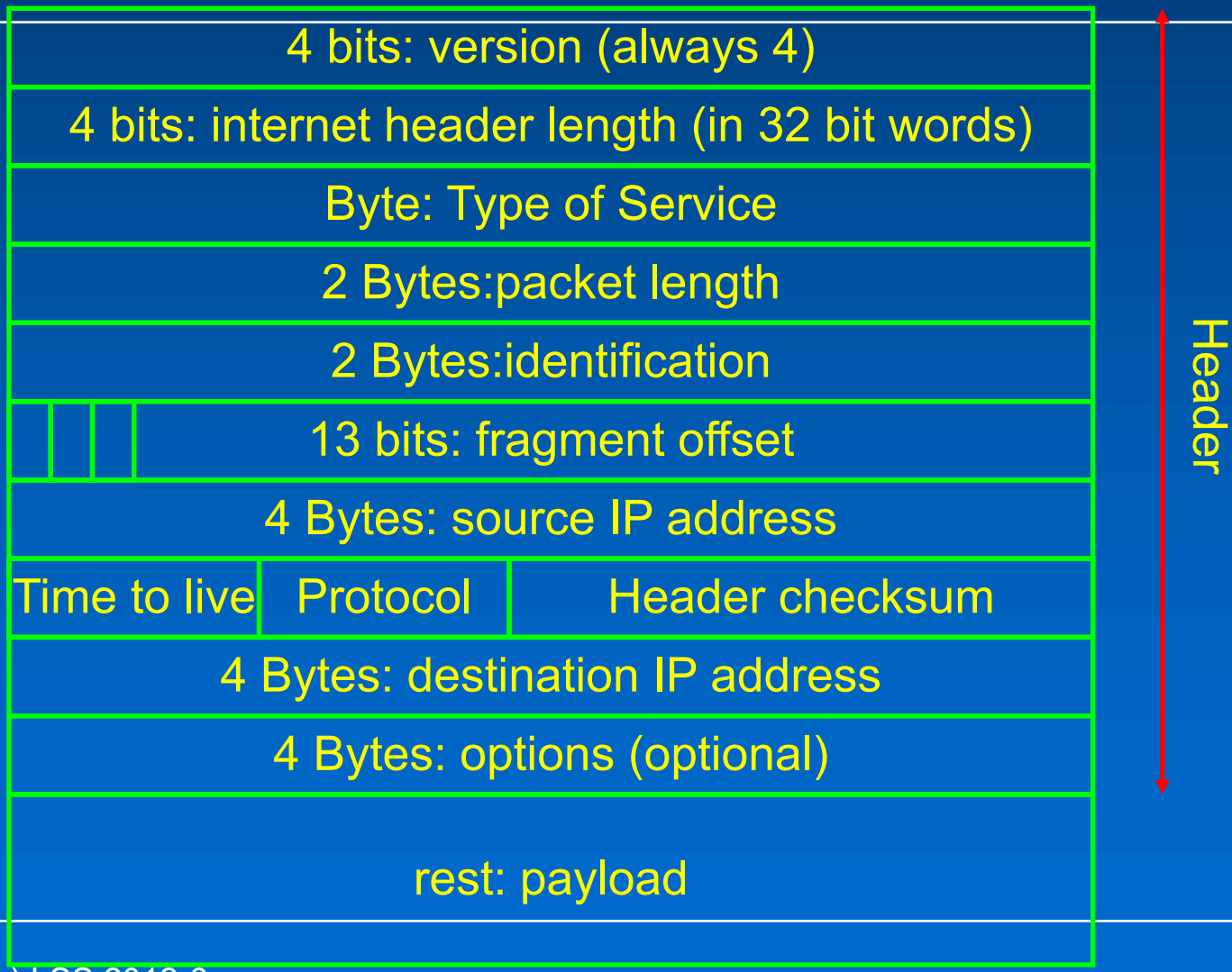
- Because is provides error detection and correction other protocols that use its facilities can assume that the data has been delivered correctly
- We look briefly at IP and then at TCP

IP: the lowest-level protocol

- Internet Protocol is the lowest-level packet-based protocol
  - Individual packets are sent from node to node.
  - Delivery may or may not occur
  - Packets may be lost or damaged
  - Packets may arrive out of order.
  - (of course, mostly, they do arrive, correctly and in order)

IP packets (IPv4)

1011010100010110100111000101111000 Etc.



Running out of addresses?

- Internet Protocol version 4 (known as IPv4)
- Has 4 bytes for source and destination addresses
  - Implies 2^{32} addresses maximum
  - (actually, some are reserved, and most are allocated in blocks)
    - Addresses are running out (why?)
- IPv6 is the replacement for IPv4
  - Much larger number of addresses
    - Uses 16 bytes (128 bits) for addresses
      - Logical maximum of 2^{128}

TCP and data integrity

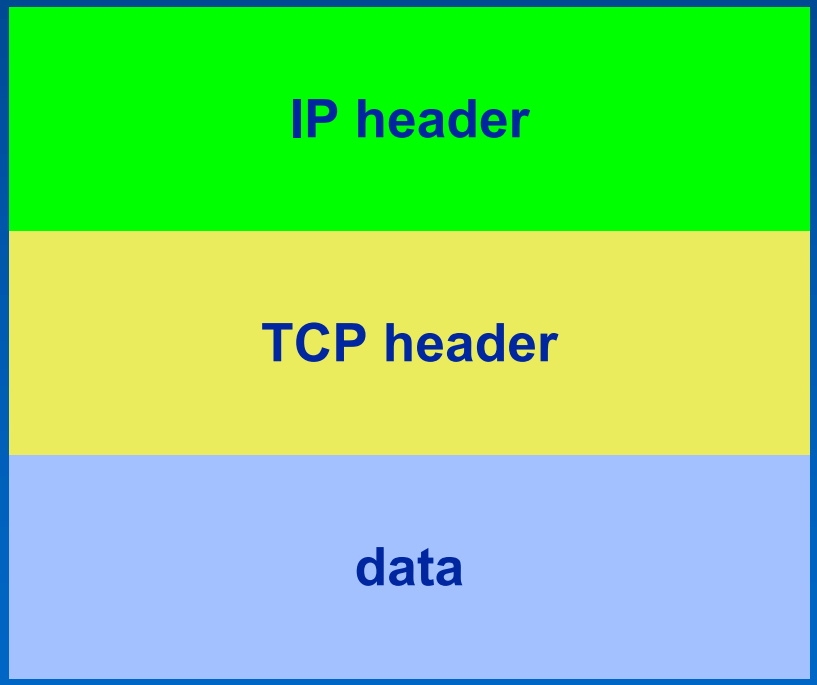
- For most purposes we need the data to be reassembled correctly
  - When might it not matter?
  - When might this be absolutely critical?
- Mostly, TCP is used to reassemble packets correctly
  - There are other protocols that sometimes get used instead:
    - User Datagram Protocol (UDP)

TCP packets

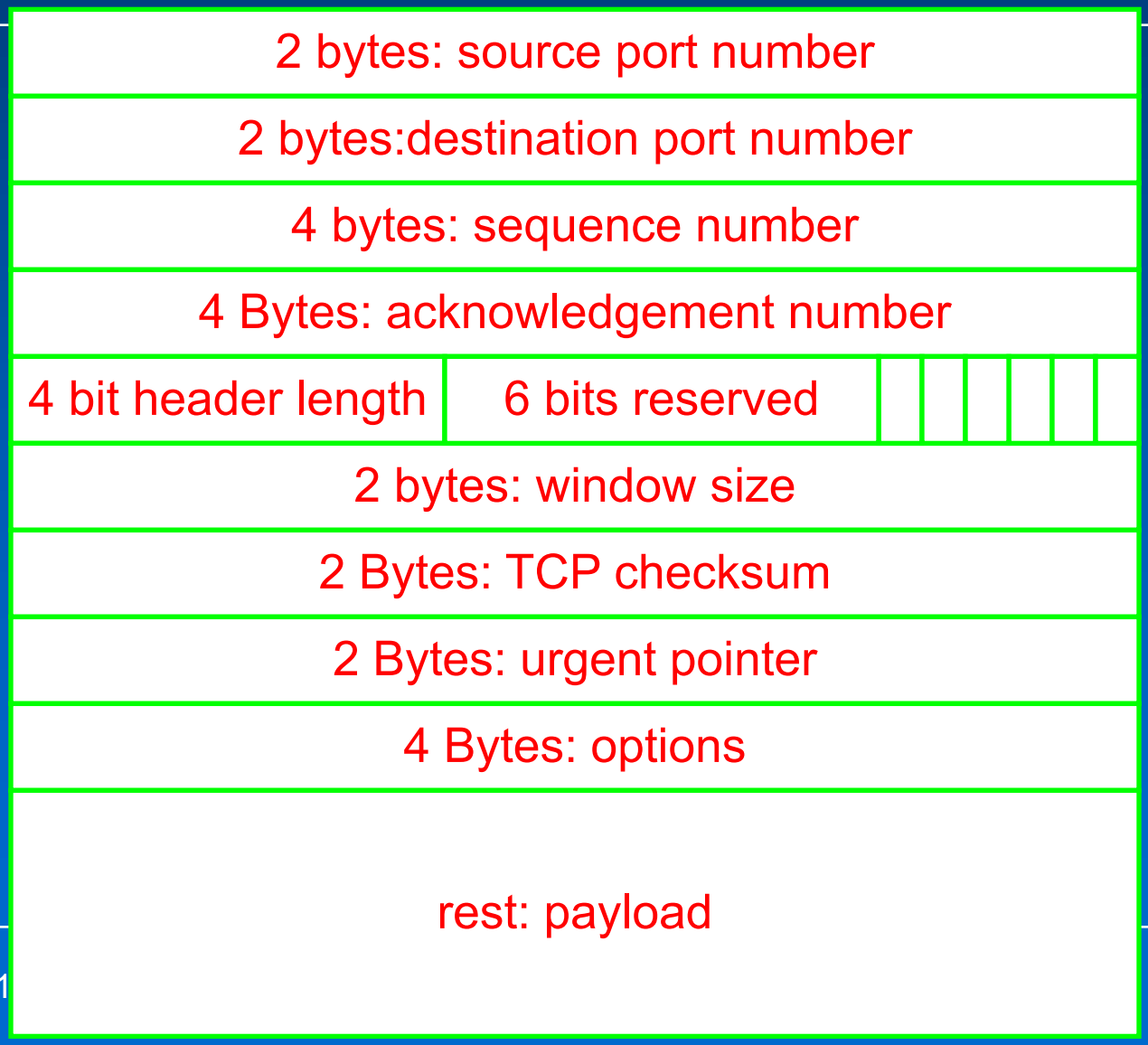
–Note that what is actually sent (physically) is ***always*** an IP packet

–The TCP header is part of the IP packet

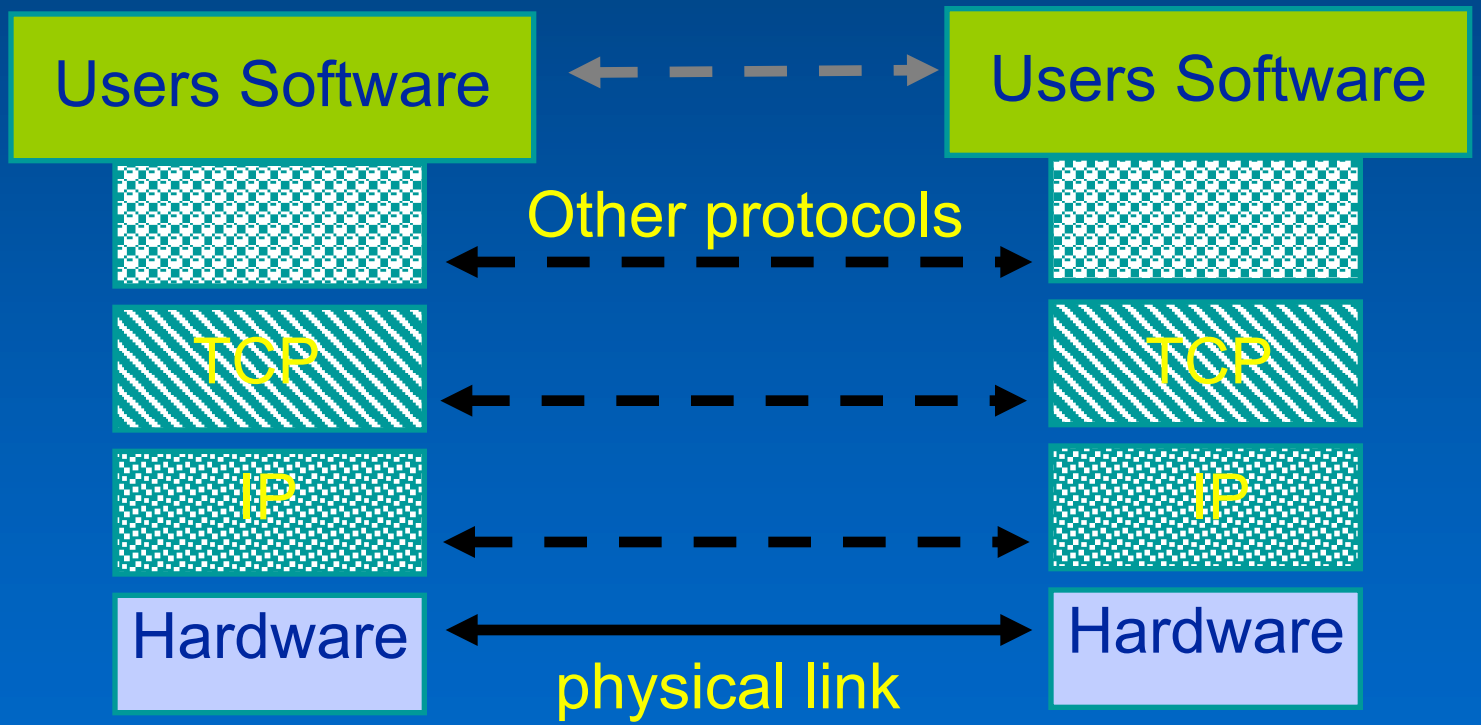
–As far as IP is concerned, it's just more data



TCP packet format (for information only!)



TCP, IP and the protocol stack



What's above TCP/IP?

- Higher level protocols
  - i.e. protocols that use the facilities supplied by TCP & IP
- Relates to what networks are used *for*
  - Distributed computers
    - Need to ensure correct data transmission always
      - Essentially using TCP/IP to enable file transmission, authentication, printing etc.
  - Access to specifically network based applications
    - E.g. WWW, email,
      - (need accuracy)
    - Streamed services (e.g. radio, video)
      - Can cope with drop-out.
- Higher-level protocols are generally specific to particular applications.

The WWW Protocol

- The World Wide Web uses a *fairly* simple protocol for transferring information
  - HyperText Transfer Protocol (HTTP)
- This sits ‘on top of’ the TCP/IP protocol the Internet uses for low-level data transfer
  - In fact, most web browsers can also use ftp and mail protocols too, but HTTP is the specific WWW protocol
- HTTP is a simple ‘client-server’ protocol
  - The *client* is Web Browser you use on the local machine
  - The *server* is the Web Server, on this or another site

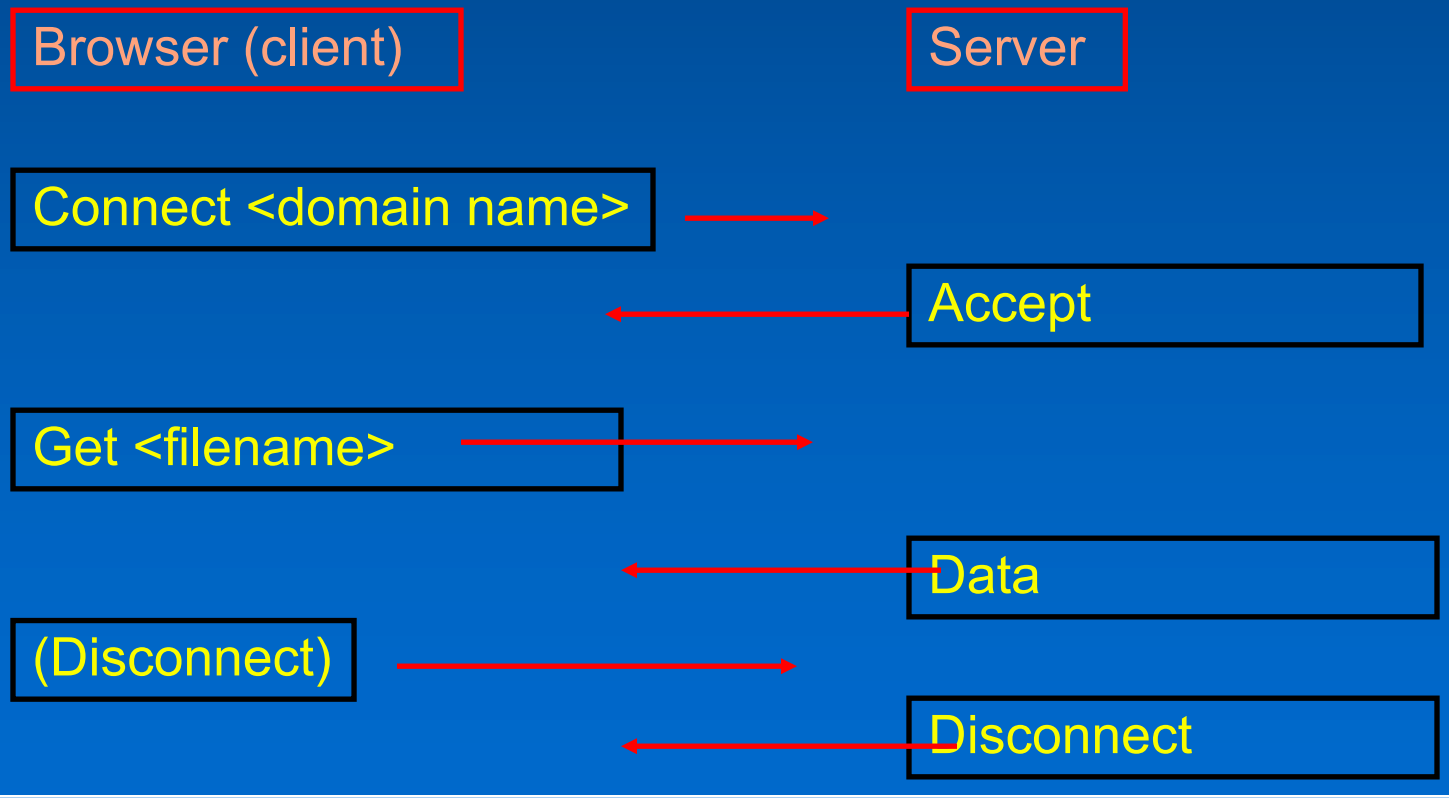
Uniform Resource Locators

- Each document on the World Wide Web has a unique name
  - known as a Uniform Resource Locator (or URL)
- This is the address of the machine, with an extension for a file name
- So:
  - **`http://www.cs.stir.ac.uk/~lss`**
  - is the address of a page (my 'Home Page')

- The first part (before the :///) describes the protocol in use
- The second part (after the :// and before the single /) is the name of the machine
- The third part (after the single /) is the address of the file on that machine
  - With some shortcuts … (e.g. ~ introduces the top level WWW directory of a user)
- The program you use to browse the Web is called a browser
  - Internet Explorer or Firefox or Opera or Safari or Chrome or ….

HTTP Basics

- HTTP is a simple **connection-oriented** protocol which assumes a robust error-free service is provided (generally by TCP and IP)
- In normal operation, the interchange is very simple:



But what if there's a problem?

- The protocol needs to take care of (for example)
  - the domain address being invalid
  - the domain address being correct, but the document address (file) being invalid, or the file being protected.

- and in addition, it should
  - provide the client with useable error messages for **anything** that can go wrong.

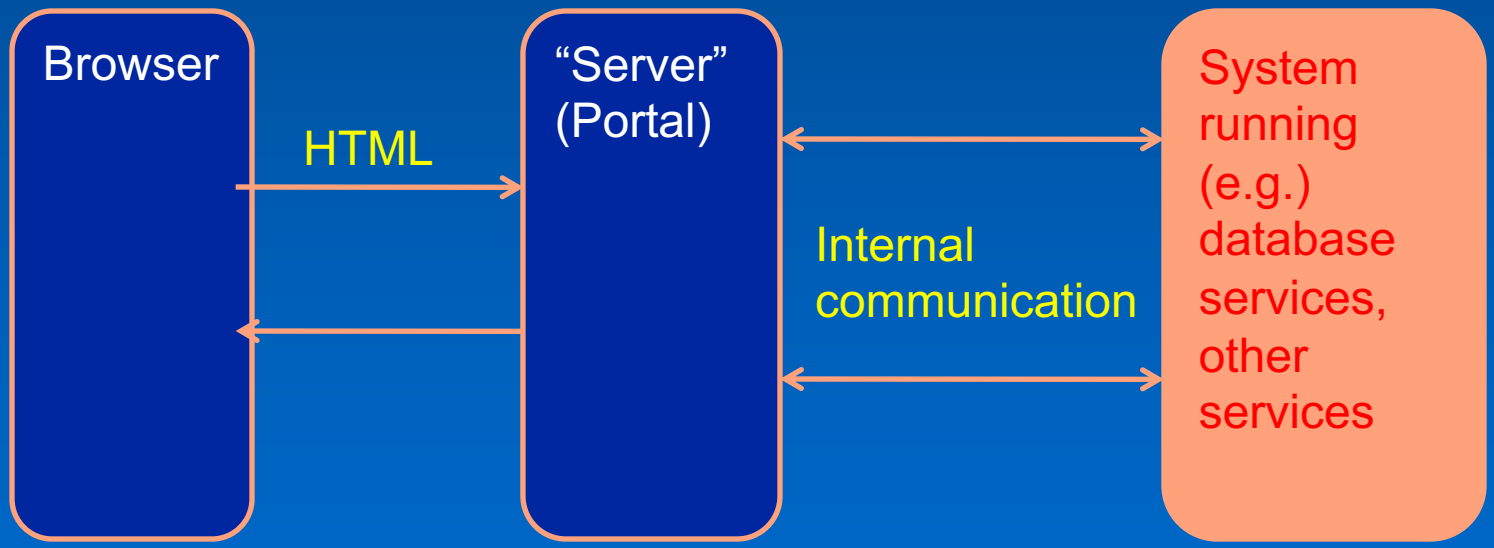
  (Can we think up what can go wrong?)
- The result is that the HTTP 1.1 protocol specification is actually 109 pages long!

Other high level protocols

- SMTP: simple mail transfer protocol
  - Used for email
- FTP: file transfer protocol
  - Used for transferring files
    - These use TCP/IP
- RTP: Real-time transfer protocol
  - Used for streaming video or audio
    - Built on to of UDP

Web portals

- Often the web is used in a more sophisticated way than simply using the browser to get data from a server
- The server is a door (portal) on to a complex system.



The HTML (etc) sent back from the portal to the browser is the result of information processing "behind" the server.

Example portals

- All E-Commerce sites are portals
  - The server communicates with a "back-office" machine which keeps track of
    - Goods being sold
    - Customers and their orders
  - And the pages sent to the browser are often dynamically generated
    - And not simply files on the server
- Many other types of services are possible
  - WebCT
  - Stirling staff/student portal
  - Mapping systems (Google maps, multimap, etc)

  - Some are highly sophisticated portals onto complex systems.
  - For example CARMEN: An e-science virtual laboratory supporting collaboration in neuroinformatics: http://portal.carmen.org.uk

Portal example: CARMEN