Computing Systems and Communications

- refs: [Null&Lobur, ch 12] [O'H&Bryant, sect 12.1–12.3]
- introduction
- communications model
- communications tasks
- communications architecture
- network addressing, protocols and routing
- internet protocol (IPv4): data transfer and future challenges

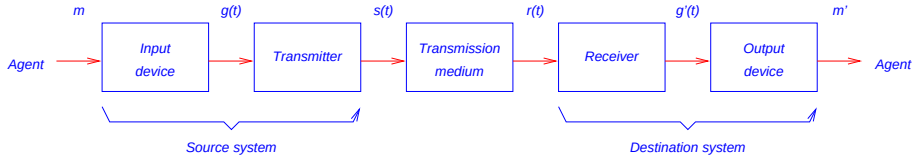
- other issues:
  - (finally . . .) Assignment 3: a taste of research in computer systems!
  - Lab 9: challenges?
  - week 12: Network Homework Exercise
  - week 12: Fri 11am lecture only: review topics (e.g. buffer overflow exploit), work though past Exam Questions
    - nominate your topics / questions!

Introduction

- need for communications
- what is communications?
  - communications allows us to transfer and exchange data
  - this allows access to much greater data than that available on one computer
- where are communications used?
  - financial systems: ATM networks, stock exchanges
  - airlines: reservation systems, aircraft control
  - manufacture: assembly line, design systems, stock control
  - academia: internet, sharing of ideas
- the computer-communications revolution: (*The network is the computer*)
  - there is no fundamental difference between data processing (computers) and data communications (transmission and switching equipment)
  - there are no fundamental differences among data, voice, graphics and video communications
  - the lines between single-processor computer, multi-processor computer, local network, metropolitan network, and long-haul network have blurred

A Communications Model

- communications: transmission of information between two agents
- data and information:
  - data: a representation of facts, concepts, or instructions in a formalised manner suitable for communication, interpretation, or processing by human beings or by computers
  - information: the meaning that a human being assigns to data by means of the conventions being applied to those data
- the information (message) is transformed (and recovered) over its transmission



Example 1 – Electronic Mail

1. the sender composes a message, m
2. through a keyboard, this is translated into a stream of bits, g
3. the bit stream is converted into a signal, $s(t)$, appropriate for transmission through the given medium
4. the signal $s(t)$ undergoes some degradation in transmission, the result is a potentially different received signal, $r(t)$
5. the receiver converts $r(t)$ into a bit stream, $g'(t)$, reversing the conversion used by the sender to produce $s(t)$
6. at this point, the receiving computer may examine the message for errors
7. if errors are detected, the receiving computer may cooperate with the sending computer, re-transmitting as necessary until the errors are resolved
8. the receiver eventually will receive a message m, that is usually identical to the transmitted message

Example 2 – Telephone Conversation

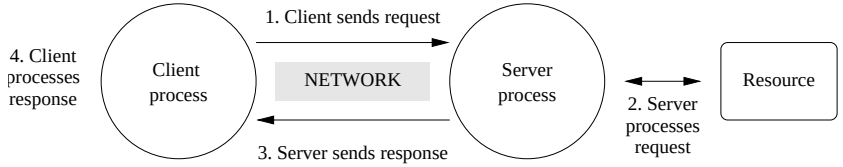
1. message m is generated as sound waves (by speaker's mouth)
2. telephone converts this to an electric signal, $s(t)$
3. the signal $s(t)$ may be directly transmitted, $g(t) = s(t)$
4. the transmitted signal will suffer some distortion, so that the received signal, $r(t)$, will not be identical to the transmitted signal, $s(t)$
5. the received signal is translated directly to sound waves, with no attempt to recover errors
6. the receiver hears a distorted version of the transmitted message m', but the message is generally understandable

Communications Tasks

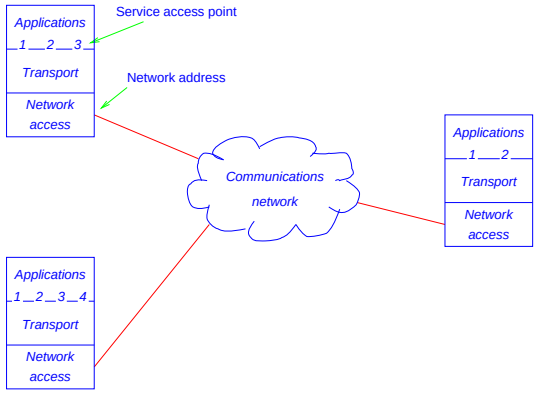
- reliable communication involves many different tasks
- some are listed below:
  - transmission system utilisation
  - interfacing
  - signal generation
  - synchronisation
  - exchange management
  - error detection and correction
  - addressing
  - routing
  - recovery
  - message formatting
  - protection
  - system management

Data Communication Networking

- in its simplest form, data communication takes place between two devices connected by some form of point-to-point transmission medium
- this is often impractical:
  - the devices are too far apart
  - there is a large number of devices, each of which require a link to many of the others at various times
- the solution is to attach each device to a communication network
- example: Client-Server Programming Model over a network
  - every network application is based on a client-server model
  - the fundamental operation in this model is the transaction

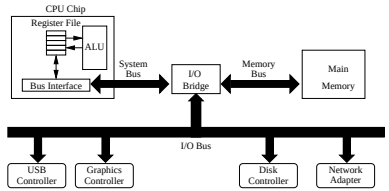


Computer Communications Architecture



on each node (computer), recall the network adapter is just another another I/O device:

- data copied from adapter to memory using DMA



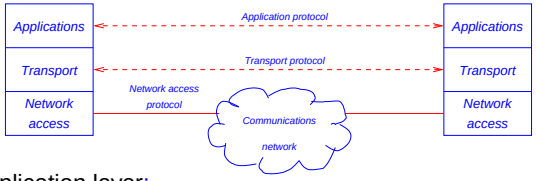
Abstraction Layers in the Computer Communications Architecture

- network access layer: deals with the physical network hardware
  - handles exchange of data between a computer and its attached network
  - the sending computer provides the network with a destination address for every message to be communicated
  - its software must deal with the specifics of the many kinds of network
- transport layer: ensures reliable transmission
  - due to the errors that are typically introduced during transmission, software is employed to ensure reliability
  - such error correction can be made independent of the application software and the particular network technology
  - may involve complex negotiations between sender and receiver
  - hides the details from the application layer and insulated from the hardware details by the network access layer
- application layer: holds the software for the various different network applications
  - different applications transfer information in different ways; e.g. telnet, ftp, email, web browser, etc.

Addressing and Protocols

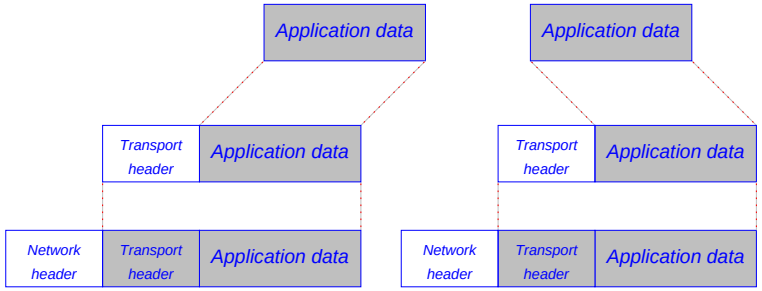
- a single machine may be host to many applications at a time
- each application on a machine will access the transport layer through an identifiable point called a Service Access Point (SAP, e.g. TCP/IP port)
- thus, an address must specify a computer and a SAP
- protocols are rules of communication that allow orderly, standardised communication between parties
- protocols are used in human communication
- in machine communication, protocols help to detect errors
- a breach of protocol is viewed as an error

Protocols in a Three Layer Architecture



- application layer:
  - data may consist of a message of *any* length
  - the data is given directly to the transport layer
- transport layer:
  - the message is typically broken into small units (packets) (issue: choosing size)
  - associated with each packet will be a header typically containing:
    - an address, a sequence number, and an error-detection code
- network access layer:
  - attaches its own header to each packet, typically containing:
    - the destination computer address, information on service requirements (priorities, time-stamps, etc.), and a further error detection code

Illustration of the Packets at Each Network Layer



(c.f. the communications model)

Network Routing

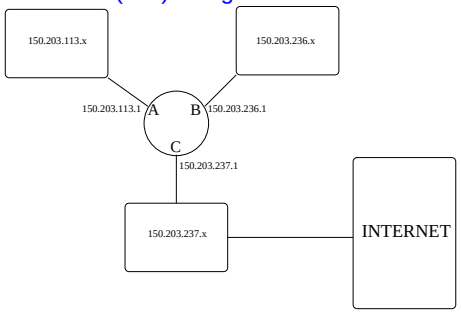
- how does a packet of information get from the PC on my desk to a computer across the campus or across the world?
  - direct connections between the billions of computers is impractical! (as well as inefficient use of resources)
- a local area network (LAN) connects computers within an organization (within a building)
- the computer passing the packets back and forth between the LAN is called:
  - a gateway (from the perspective of other computers in the network), or
  - a router (from the perspective of the network engineer)
  - can be anything from an old i386-based Linux box to a $10K+ CISCO box
  - as well as forward packets, may filter (security) and log (pay-per-MB) them, and monitor the network
- thus the routing problem can be broken into two parts:
  - communication within the the local area (LAN)
  - communication between LANs (inter-network communication)

Address Structure and Routing Decisions

- (in IPv4) computers are addressed with a unique 32-bit number
- split into four 8-bit numbers (dotted-decimal notation), e.g.
- `cs.anu.edu.au`: 150.203.164.35
  `www.anu.edu.au`: 150.203.99.8
  `www.abc.net.au`: 144.135.8.151
- the upper numbers addresses the network(s), the lower numbers (normally last) specifies the computer within the LAN
- how does a router know what to do with a packet?
- it makes a decision based upon the destination address of the packet and its given routing rules (tables)

example: packets for:

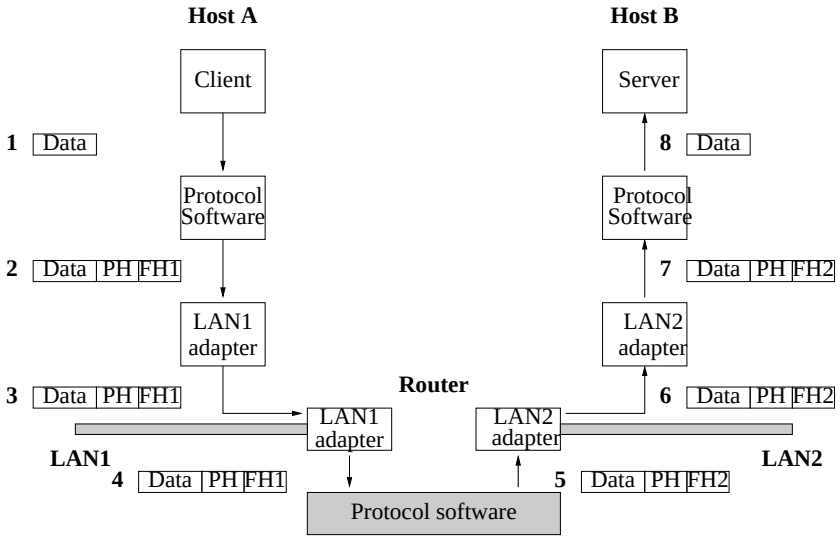
- 150.203.113.xx go out on link A
- 150.203.236.xx go out on link B
- 150.203.237.xx go out on link C
- other (sub)nets go out on link C



Internet Protocols

- developed in th 1970s (ARPANET, DoD) as a robust military network
- today: the Internet is based on TCP/IP
- many applications: WWW, E-Mail, FTP, Telnet, ...
- TCP/IP is a transport layer based on packet switching
- Internet Protocol (IP):
  - provides a non-reliable, connectionless network service
  - a packet consists of a header and user data (max. 64 KB)
  - header contains sender and receiver address, segmentation, information, maximal life time, checksum, etc.
- Transmission Control Protocol (TCP):
  - provides reliable, connection-oriented network service (connection has to be acknowledged)
  - TCP header contains source and destination port numbers, sequence- and acknowledge numbers, priority and checksum
  - port numbers refer to applications (e.g. FTP 21, Telnet 23, WWW 80)
  - the TCP packet is forwarded to the IP layer

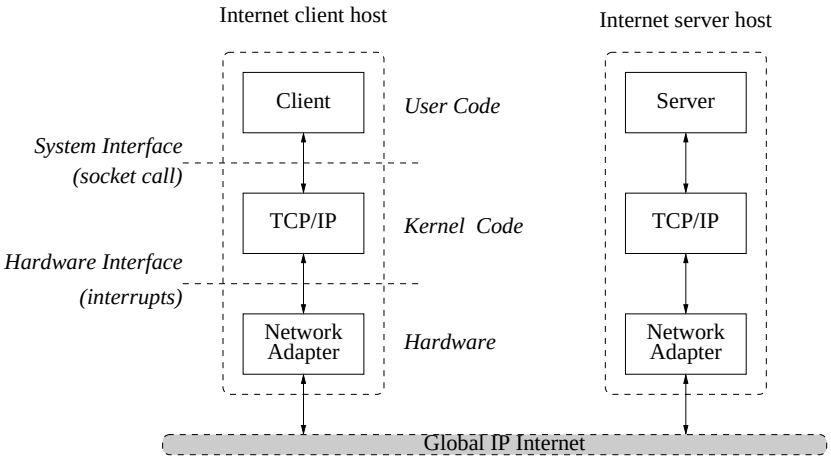
Data Transfer over the Internet



Data Transfer over the Internet: Steps

1. client on host A invokes a (socket `send()`) system call
2. host A creates a LAN1 frame (extended packet) by appending IP header and LAN1 frame header to data
3. LAN1 adapter sends frame to network
4. router reads frame from its LAN1 adapter and passes it to protocol software
5. router determines how to route frame and appends new LAN2 frame header as required
6. router passes frame though its LAN2 adapter
7. host B's LAN2 adapter reads frame and passes it to protocol software
8. protocol software on B strips off packet header and frame header and places data into server's virtual address space
9. the data is collected upon a socket `recv()` system call

Hardware/Software Organisation of Internet Application



The Internet and WWW: Major Challenges

- IPv4: problem: world is running out of 32-bit addresses
  - dotted-decimal structuring 202.125.14.192 causes wastage!
  - a solution: CIDR Hierarchical Addressing and Variable Length Subnet Masking
  - will eventually need to adopt IPv6 (64-bit addresses)
- internet routing remains an issue (e.g. distribution of info. on congestion / outage)
- how can a single web site deal with an intense period of (world-wide) demand?

  e.g. 2006 World Cup Soccer site
- how can the (exponentially increasing) WWW be so efficiently searched?

  e.g. Google
- approaches are similar to that used in other computer systems
  - content delivery networks: virtualization (of URLs) and caching (e.g. AKAMAI)
  - precomputation and parallelism (over and within queries)
- virtualization (of networks) also appears in the form of virtual private networks (VPNs)
- the future: semantic (rather than keyword) -based World Wide Web