Week 13

Networks (cont)

Network Architecture

2/33

Five layer model:

- lowest (least abstract) level
- *physical layer* ... bits on wires
- *link layer* ... e.g. ethernet, MAC
- *network layer* ... e.g. IP
- *transport layer* ... e.g. TCP/UDP
- *application layer* ... e.g. HTTP, email
- highest (most abstract) level

Packets at each level incorporate headers from lower levels

Software at each level uses headers appropriate for its purpose

Transport Layer

3/33

Transport layer deals with ...

- *data integrity*
  - some apps (e.g. file transfer) require 100% reliable transfer
  - other apps (e.g. audio streaming) can tolerate some loss
- *timing*
  - some apps (e.g. networked games) require low transmission delay
- *throughput*
  - some apps (e.g. multimedia) require minimum throughput
  - other apps ("elastic apps") can use whatever is available
- *security*
  - some apps (e.g. web services) require encrypted transmission

... Transport Layer

4/33

Properties of some common apps ...

- file transfer: no loss, elastic, not time sensitive
- email: no loss, elastic, not time sensitive
- web/http: no loss, elastic, not time sensitive
- audio: loss-tolerant, 5Kbps-1Mbps, few ms delay ok
- video: loss-tolerant, 10Kbps-5Mbps, few ms delay ok
- games: loss-tolerant, 5Kbps-5Mbps, few secs delay ok
- texting: no loss, elastic, few ms delay ok

Transport Layer Protocols

5/33

Transport layer protocols provide

- *logical communication* between processes on different hosts
- transport protocols run within end-point processes
  - sender: splits messages into *segments*, passes to network layer
  - receiver: reassembles segments into messages, passes to app layer

Two main transport layer protocols on Internet

- TCP ... reliable, connection-oriented protocol, byte-stream
- UDP ... unreliable, simple, connectionless protocol, segments

... Transport Layer Protocols

6/33

TCP (Transmission Control Protocol) provides ...

- *reliable transport*: data flow between sender/receiver
- *flow control*: sender doesn't overwhelm receiver
- *congestion control*: slow sender if network congested
- *connection-oriented*: setup required between client/server

Does not provide: timing/throughput guarantees, security

UDP (User Datagram Protocol) provides ...

- fast (for sender), but unreliable data transfer

Does not provide: reliability, flow control, timing/throughput guarantees, security

---

... Transport Layer Protocols

7/33

TCP is typically layered on top of IP protocol

- IP is an unreliable network-layer protocol
- TCP provides reliable stream of data on top of IP

How TCP works ...

- set up connection between sender and receiver
- sender transmits a pipeline of segments
- expect ACK for each segment
- retransmissions triggered by timeouts and duplicate ACKs
- receiver manages receipt and collation of segments

... Transport Layer Protocols

8/33

Above comments makes UDP sound "sub-standard"

However, it has a number of advantages (over TCP) ...

- small segment headers, no connection setup costs
- UDP senders can transmit segments as fast as they like
- segments are handled independently of each other

Effective for low-latency apps that can tolerate lost/damaged packets

Error detection requires use of *checksum*

- sender and receiver treat segment data as sequence of ints
- sender: compute sum of ints, store in header
- receiver: compute sum of ints, compare to checksum

Important applications that use UDP: DNS, TFTP, RTSP

Network Layer

9/33

Transport layer provides a way for app processes to communicate

Network layer provides communication between *hosts*

- hosts specified by IP addresses (e.g. 129.94.242.19)

Basic functions of network layer (Internet layer)

- for outgoing packets:
  - select the next-hop host
  - pass packet to link layer to transmit to host
- for incoming packets:

    - if reached destination: extract payload, pass to transport layer
    - if not reached destination, treat as outgoing packet
- for all packets/transmissions: error detection, diagnostics
- may also split "oversize" segments into smaller packets

Network Layer Protocol

10/33

IP (Internet Protocol) is a network layer protocol that provides ...

- host addressing and routing of packets
- splitting and reassembly of large packets

Routing is one of the most important functions

- each host maintains a *routing table* (maps address→next-hop)
- uses *subnets* to reduce table size
    - all hosts in a subnet have common prefix (e.g. CSE 129.94.2xx.xxx)
    - all IP addresses with common prefix sent to same host (gateway)
- routing table maintained dynamically
    - hosts transmit "active" signals to each other periodically

---

... Network Layer Protocol

11/33

Simplified routing algorithm (IP forwarding)

```
Inputs: D = destination IP address
        N = network prefix (of IP address)

if (N matches a directly connected network address)
   send packet over link to D
else if (routing table contains a route for N)
   send packet to next-hop address given in routing table
else if (default route exists in routing table)
   send packet to the default route
else
   can't find route; transmit error message to sender
```

Link Layer

12/33

Link layer takes packets from network layer and transmits them

- every host on network has network layer implementation
- implemented as a combination of hardware/software
- each host contains a *network interface card* (NIC)
- connected to system bus as i/o device

Services provided by link layer

- *flow control* ... pacing between adjacent sending and receiving nodes
- *error detection* ... detects transmission errors; flags error to network layer
- *error correction* ... can identify and correct single bit errors

If error corrected, no retransmission; if not correctable, request retransmission

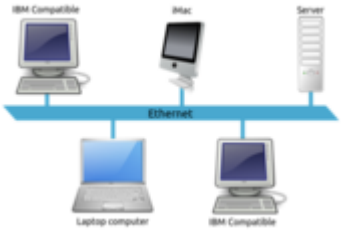
---

Ethernet

13/33

Ethernet is an example of link layer implementation

- ethernet is a cable physically connecting multiple hosts
- data broadcast onto cable, tagged with receiver MAC address
- devices recognise their own data using MAC address



MAC address = Media Access Control address, stored in NIC

... Ethernet

14/33

Ethernet is a shared broadcast medium, and so ...

- *interference*: two different packets broadcast at same time
- *collision*: node receives two or more signals at same time

Multiple access protocols handle this, but cannot also use ethernet

Example multiple access protocols

- *channel partitioning*
  - partition channel based on time-slices/frequency-bands/...
  - allocate one partition to each node for exclusive use
- *random access*
  - allow collisions; need mechanisms to recover from collisions
- *taking turns*
  - nodes take turns; nodes with more to send get longer turns

---

... Ethernet

15/33

A random access transmission protocol (CSMA)

1. NIC receives packet from network layer and creates frame
2. if NIC senses channel idle, start frame transmission
3. if channel busy, wait until channel idle and go to step 2
4. if entire frame transmitted without interference, go to step 8
5. if NIC detects interference while transmitting, abort transmission
6. after abort, choose "random" delay time (longer if more collisions)
7. after waiting, go to step 2
8. mission accomplished (frame transmitted)

Course Review

Course Goals

17/33

At the end of COMP1521, we hope that you ...

- understand the structure of computer systems
- can describe how computers/programs work at a low-level
- are better able to reason about and debug your C programs

Major topics ...

- components of modern computer systems
- how C programs execute (at the machine level)
- how to write (MIPS) assembly language
- Unix/Linux system-level programming
- how operating systems and networks are structured
- introduction to concurrency, concurrent programming

Detailed Topics

18/33

- Processors
  - data representation, instruction set
  - *assembler programming*
- Program execution (mapping C to assembler)
  - memory layout: stack, heap, data, code
  - control structures, function calls
- Operating system architecture
  - memory, devices, buffers, i/o, interrupts, signals
  - virtual memory, processes, file systems, *system calls*
- Concurrency
  - synchronisation, coordination, communication
- Network architecture
  - layers, protocols, addressing, transmission, sockets

Course Assessment

19/33

```
quizzes   = mark for online quizzes        (out of 10)
labs      = mark for lab exercises         (out of 10)
ass1      = mark for assignment 1          (out of  7)
ass2      = mark for assignment 2          (out of 13)
courseTot = (quizzes+labs+ass1+ass2)       (out of 40)

pracExam  = mark for Prac part of exam     (out of 30)
theoExam  = mark for Theory part of exam   (out of 30)
examTot   = (pracExam+theoExam)            (out of 60)

examOK    = (examTot > 22/60)              (after scaling)

mark      = courseTot + examTot            (out of 100)

grade     = HD|DN|CR|PS  if examOK && mark ≥ 50
          = FL           if examOK && mark < 50
          = UF           if !examOK
```

Final Exam

Final Exam

21/33

3-hour exam on Wed 8 November, worth 60% of course mark.

Held in CSE labs (allocations posted on web site)

Exam runs in two sessions 9:15-12:30 and 12:55-4:10 (incl reading)

Like COMP1511, afternoon people assemble early (12:20)

Too many people opted for afternoon on the Poll! No help.

- send me email if you have a *compelling* reason for either session
- compelling = another exam that day, DSU requirements, ... not "I want to sleep in"

Exam has two parts, but all answered and submitted online

- 30/60 based on Practical Part, 30/60 based on "Theory" Part

---

... Final Exam

22/33

Bring: your student card, a pen, that's all

What's available to you (in the exam and right now):

- online access to Unix Programmers Manual (`man`)

- a C quick-reference sheet (accessible via menu)
- a MIPS quick-reference sheet (accessible via menu)
- a sheet of paper for rough working (not to be removed)

What you do **not** have access to:

- no access to the COMP1521 web site
- no access to your files (labs, assignments, etc.)
- no access to Web, Google, Facebook, Stack Overflow, etc.

... Final Exam

Practical Part (aka "Prac Exam")

- three small(ish) programming tasks
- aim: check whether you can program in MIPS and C
- level-of-difficulty: two easy (MIPS and C), one not-so-easy (C)
- supplied with test data and `check` script
- once it passes all `check` tests, submit and move on
- partial marks available **if** submitted program compiles/loads
- zero marks if no submission or submission has compile errors
- zero marks for "table look-up" solutions (extra tests in marking)

... Final Exam

Theory Part (aka "Theory Exam")

- short-answer questions (about 7, with varying marks)
- aim: check how much you know about course material
- some calculation may be required; you have on-screen calculator
- cover a wide range of topics from the course
  - e.g. what is the output of some program or other?
  - e.g. which page is next for replacement under LRU?
  - e.g. which edges are in the minimum spanning tree? :-)
- think: tutorial-like questions, quiz-like questions (but not M/C)

... Final Exam

Some exam strategy tips:

- 180 mins, 90 marks ⇒ 1 mark ≅ 2 mins
- partition time between theory and prac as you like/need
- but don't spend more than 40 mins on any one Prac question
- if stuck with debugging, work on the next question
- allow at least one hour for theory questions

Revision Strategy

How to revise?

- re-read lecture slides and example programs (see web)
- take a look at old exams (on web site soon)
- review tute and lab exercises and assignments
- come to a *StuVac consultation* to resolve problems
- write some programs
  (programming is a *skill* that improves with practice)

No questions from past exams/labs/assignments will be in the exam.

Supplementary Exams

Supplementary exams are only available to students who

- do *not* attend the exam on November 8

- have a serious documented reason for not attending
  (must convincingly show that your ability to study was significantly affected)
- score ≥ 18/40 for quizzes+labs+assignments

If you attend the final exam

- you are making a statement that you are "fit and healthy enough"
- it is your only chance to pass (i.e. no second chances)

Supp Exam will be held on Tuesday 21 November

- don't leave the country if you have a Supp and still want to pass

---

Assessment

28/33

Assessment is about determining how well *you* understand the syllabus of this course.

If you can't *demonstrate your understanding*, you don't pass.

In particular, I don't pass people just because ...

- please, please, ... my parents will be ashamed of me
- please, please, ... I tried *really hard* in this course
- please, please, ... I'll be excluded if I fail COMP1521
- please, please, ... this is my final course to graduate (unlikely)
- etc. etc. etc.

... Assessment

29/33

Of course, assessment isn't a "one-way street" ...

- I get to assess you in the final exam
- you get to assess me in the Course Evaluation

Available via myUNSW until November 3. Please fill out.

Telling me good things is fine ...

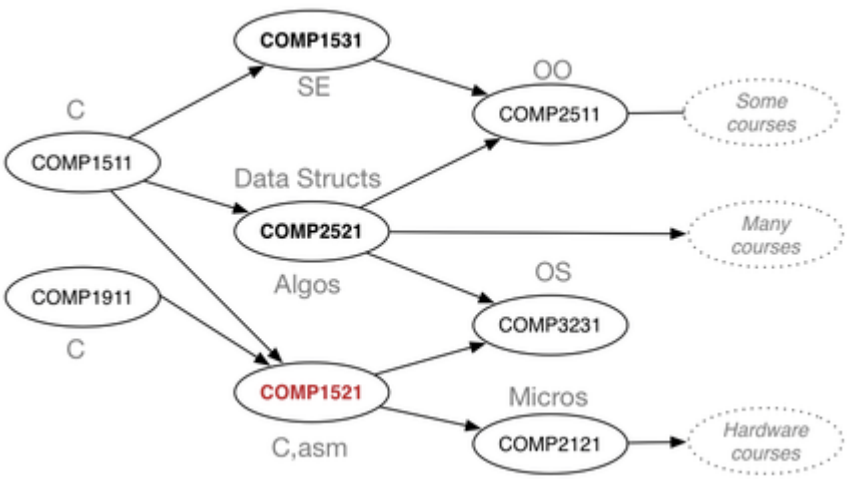
Telling me things I did wrong is better ...

(If I don't know what's wrong, I don't know what to fix)

The Future

Course Context

31/33



Course Offerings

COMP1531 Software Engineering Fundamentals

- 2018: sem1, sem2, 2019: term1, term3

COMP2511 Object-oriented Programming

- 2018: sem1, sem2, 2019: term2, term3

COMP2521 Data Structures and Algorithms

- 2018: summer, sem1, sem2, 2019: term1, term2, term3

COMP2121 Microprocessors and Interfacing

- 2018: sem1, sem2, 2019: term1, term2

COMP3231 Operating Systems

- 2018: sem1, 2019: term1

And Finally ...

Good Luck with your Exams!

and with your future computing studies

Produced: 23 Oct 2017