University of Glasgow | School of Computing Science

Networks & Operating Systems Essentials (NOSE – Level 2)

Dr Angelos Marnerides
<angelos.marnerides@glasgow.ac.uk>
School of Computing Science

Why ?

- An intro to:
  - Computer Networks
    - How are systems connected together to form a network?
    - How are networks connected together to form an internet?
    - How do data find their route through this maze?
    - How can we have reliable communications through unreliable links?
    - What are the repercussions for programs that communicate over this infrastructure?
  - Operating Systems
    - Why do we need them?
    - What is a process? How is it different to a thread? How can we have multiple of them running concurrently?
    - How can we partition/share the available memory among them?
    - What if the computer's RAM is not enough?
    - How do we store and retrieve data from files?

Intended Learning Outcomes

- By the end of this course, you should be able to:
  - Understand the principles underlying networked systems, such as the Internet, including the importance of layering in protocol design
  - Describe the key features of a local-area network technology, such as Ethernet
  - Explain the concept of internetworking, and how the Internet is built as a network of networks; describe key features of the network layer
  - Describe what is meant by a best-effort service, and explain why the Internet offers such a service; discuss how the transport layer can enable reliable connections on this substrate
  - Understand the difference between the Web and the Internet, and be able to describe, in outline, the operation of an application layer protocol such as HTTP
  - Be familiar with issues around privacy and security of Internet traffic
  - Use appropriate terminology in correctly explaining the functions of an operating system
  - Contrast the low-level hardware facilities (e.g., physical memory frames) with the higher-level abstractions provided by an operating system (e.g., virtual memory pages)
  - Evaluate the relative merits of a range of simple resource allocation algorithms (e.g., for process scheduling)
  - Implement simple operating system components and basic algorithms (e.g., scheduling, free space management)

Course syllabus & outline

Networks

- Basic concepts
- Protocols & layers
- Physical and data-link layers
  - Case study: Ethernet
- Network layer
  - Case study: IP
  - Inter-/Intra-Domain Routing
- Transport layer
  - Case study: TCP & UDP
- Application layer
  - Case study: DNS
  - Case study: HTTP
- Security and privacy

Operating systems

- Processes
  - Process management
  - Scheduling
- Memory
  - Virtual memory
  - Page tables
- Storage
  - Block storage
  - File systems
  - Case study: MSDOS FS
  - Case study: UNIX FS

Lectures and Tutorials

- **Lectures**: slides + videos to be uploaded in advance on Moodle
  - 2x 1-hour Q&A sessions on Tuesdays and Thursdays (11:00-12:00)
  - **Advise**: **view lecture videos in advance and post questions in advance**
    - **Lectures are chunked into 10-15min short videos**
  - Questions can be pre-posted on **Slido** and answered during Q&A slots.
  - Slido links to be provided on a weekly basis on Moodle

- **Labs:** 1x 1-hour Team—based lab/tutorial
    - **Important: you should work on your lab material before your pre-assigned lab session (lab sessions are used for Q&A)**
    - **Keep an eye out on Moodle and your email inbox!**
    - Spoiler: no lab this week
  - **NOSE2 Teams channel** : keep an eye for instructions on Moodle how to join the MS Teams group: this is how you should approach tutors and myself for Q&A during your allocated lab session.

Lectures and Tutorials (cont..)

- Plus one-on-one (virtual –MS Teams- for now) meetings when needed
  - Please check the Moodle page on my availability
  - Please contact me at least 4 working days in advance in order to arrange a meeting

- Please make sure before hand:
  1. you have tried your best on understanding the material at your own time
  2. you have tried even more than your best on understanding the material through additional sources
  3. you have tried to ask your lab tutors during lab sessions
  4. If 1) and 2) didn't work then email me to arrange a slot.

Lectures and Tutorials (cont..)

- Pre-session reading **vital** for the course
  - To be uploaded on Moodle every Thursday/Friday, for the lectures of the following week
  - Advisable to study the material before watching the lecture video.
  - *If no such material has been uploaded to Moodle by Friday, then there will be none for the coming week*
  - ***Please use the allocated lecture slots for your advantage by asking questions on MS Teams (either in the main group or lab subgroups)***

- Extra reading material to be provided as appropriate
  - This will not be assessed or examined; only provided for those wanting to dig deeper…

- Books are recommended reading
  - Do your pre-session reading, attend Q&A lecture slots, take notes
  - Resort to the books when something is unclear or you want to know more

Assessment

- **Final exam**
  - 80% of the grade
- **Assessed Exercises**
  - 1 AE on networks – 10%
    - Hand out: Week 4
    - Hand in: Week 6
  - 1 AE on operating systems – 10%
    - Hand out: Week 8
    - Hand in: Week 10

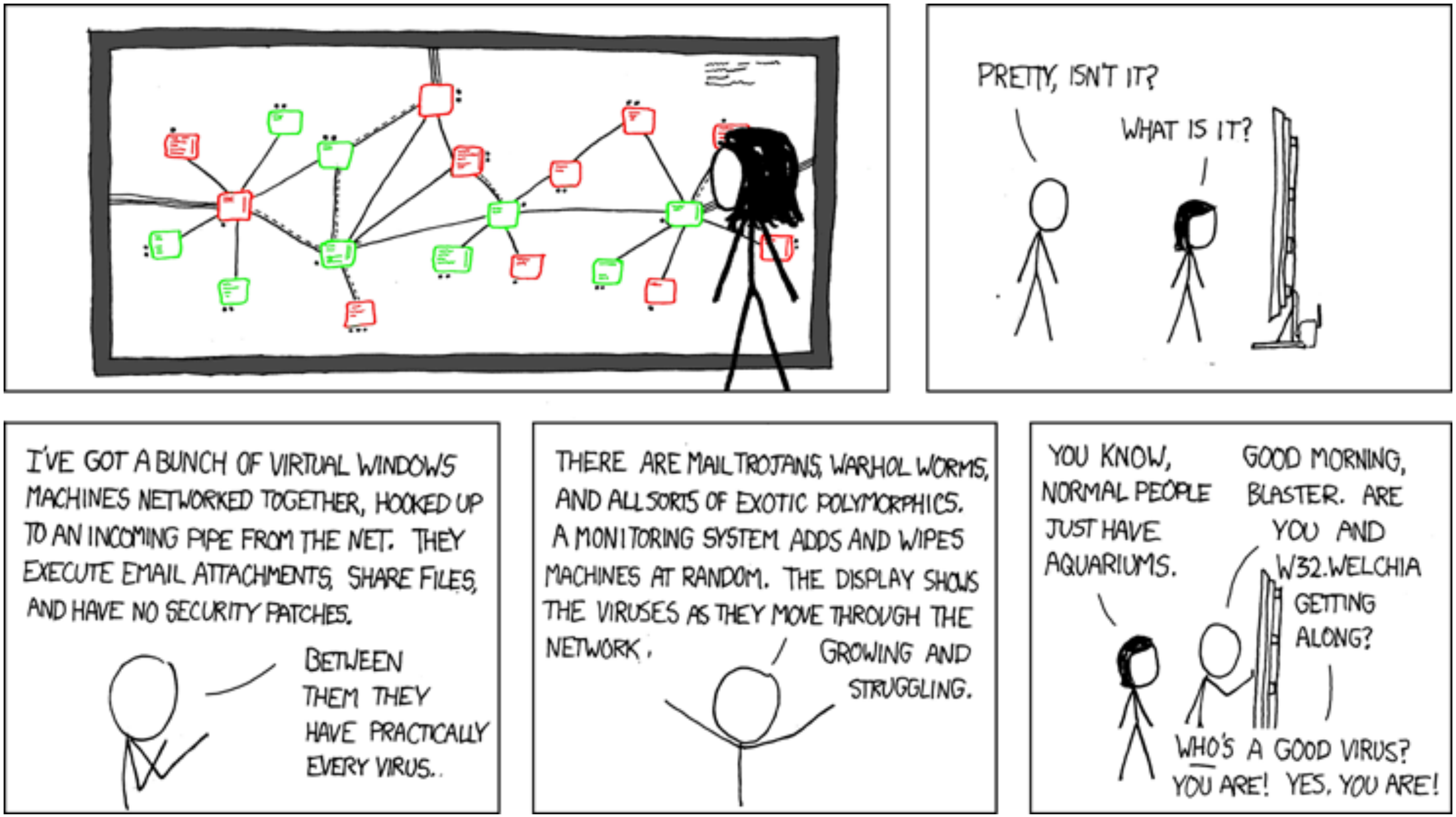
Miscellania

- You are assumed to know how to interact with a command line interface
  - A large part of your lab work will be based on this
- You are assumed to know how to write code in Python
  - Assessed exercises will be based on this
- Unless explicitly stated otherwise, lab work will be formative
  - If you submit an answer, you will receive feedback but your answer is not marked; treat labs as a way of being exposed to practical aspects of the course and/or as a "conversation starter"
- Assessed coursework will be summative
  - I.e., marked
- Q&A forums on MS Teams
  - Use freely but use responsibly…
- **Beware of self-study!**
  - Not just about spending time on your own reading up
  - You are in charge of **what**, **how** and **when/for how long** you study

Caveats

- Lecture notes are **not** sufficient; you **must** study the pre-session reading material and keep **your own notes**
- You **must** schedule time **each week** to make sure you don't lose track
- You **must** come prepared to the labs and tutorials and **work on the material in advance**
- You **must** pay attention to the lab/assessment handouts and instructions
- You **must not** leave summative assessment till the last minute
- Proper time management will be your friend throughout the semester…
- **Budget your time**
  - 10-credit course = 100 hours of work
    - ~20 hours to revise for the exam + 1.5 hours to sit the exam
      - Assuming attendance/participation in the course throughout the semester!
    - 20% assessed exercises = 20 hours, 10 hours each (~3.3 hours per week)
    - 5.5 hours weekly (on average) for
      - Attending lectures: 2 hours
      - Attending lab/tutorial:1 hour
      - **Self-study: ~2.5 hours** (**NOT including time** for **AEs**)
  - **~5.5 hours** per week **per 10-credit course**
  - **~8.8 hours** per week when AEs are due
  - **Now do that 6x!**

Today, on NOSE2…



Source: https://xkcd.com/350/

Based on slides © 2017 Colin Perkins

INTRODUCTION TO NETWORKED SYSTEMS

Networked Systems

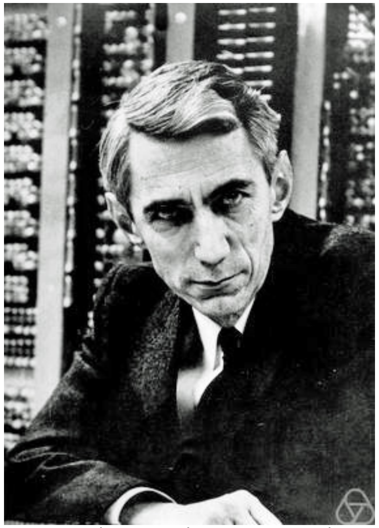
- Autonomous computing devices that exchange data to perform some application goal
  - The exchange of data is explicitly visible to the application – the system is aware of the network
  - Applications using the Internet is one example, but other networks in widespread use:
    - Digital broadcast TV (e.g., FreeView in the UK)
    - Mobile voice telephony
    - Controller area networks connecting sensors and other components within vehicles or aircraft
    - Sensor networks
    - …

- Key aspects:
  - Communication: how is information exchanged across a single link?
  - Networking: how are links interconnected to build a wide area network?
  - Networked system: how do systems communicate across the network?

Networked Systems

- Communication:
  - Messages transferred from source to destination(s) via some communication channel
    - Size of messages might by bounded
    - Communication might be *simplex*, *half-duplex* or *full-duplex* (?)

- Information:
  - Messages convey information
    - The amount of information in a message can be characterised mathematically (*Information Entropy*)
      - $\mathrm{H(X)} = \mathrm{E}[-\ln(\mathrm{P(X)})] = -\sum_{i=1}^{n} P(x_i) \log_b(P(x_i))$
        - » $b = 2$ → *bits* or *"shannons"*
  - Capacity of channels to convey information can also be modelled
    - How much? How fast? How much power used?
    - Physical limits exist on the capacity of a channel



Signals

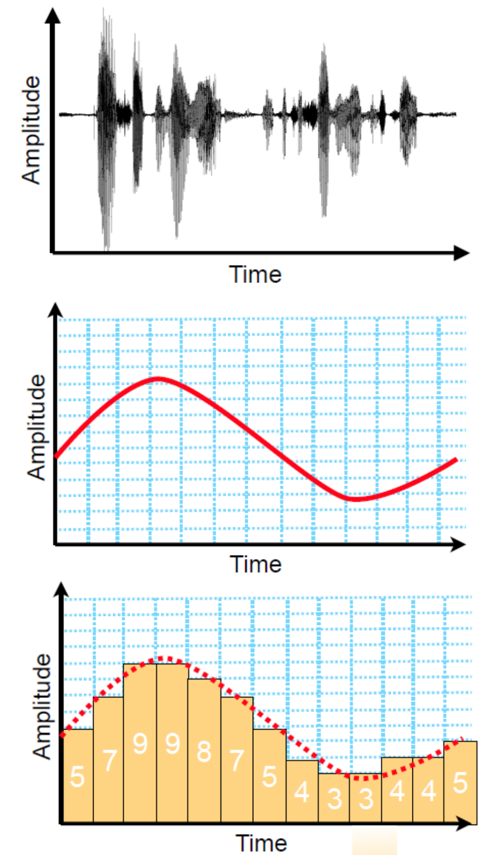
- Physical form of a message is a *signal*
  - May be a material object (carrier pigeon, CD, …)
  - Usually a wave (sound, electrical signal, light, radio)

- Signal may be analogue or digital
  - Analogue: a smooth continuum of values
  - Digital: a sequence of discreet symbols
    - Mapping information to symbols is known as *coding*

Analogue Signals

- Simplest analogue signal: amplitude directly codes value of interest
  - AM Radio, analogue telephones
- Can be arbitrarily accurate
- Susceptible to noise and interference on channel
- Difficult to process with digital electronics

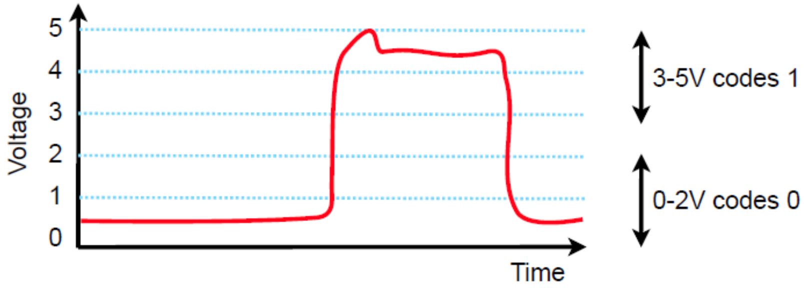
- Any analogue signal can be represented digitally:
  1. Sample the signal at a suitable rate
  2. Quantise to nearest allowable discrete value
  3. Convert to digital representation
- The *sampling theorem* determines the rate at which the signal must be sampled for accurate reconstruction



01010111**1001**1001**1000**0111**0101**0100…

Digital Signals

- Digital signals comprise a sequence of discrete symbols – fixed alphabet, not arbitrary values
  - But underlying channel is almost always analogue
  - Modulation used to map a digital signal onto the channel
  - Example: non-return to zero (NRZ) modulation



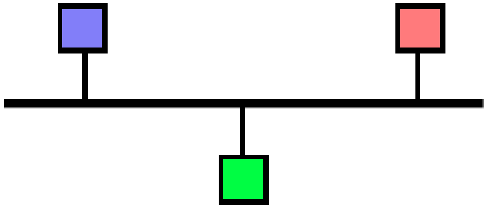
- Computing systems use binary encoding
  - The digital signal comprises two symbols: 0 and 1
- Networked systems often use non-binary encoding
  - Example: wireless links frequently use complex modulation schemes with either 16, 64, or 256 possible symbols
  - Number of symbols transmitted per second is the *baud rate* (not to be confused with the *bit rate*)

Émile Baudot (1845-1903)

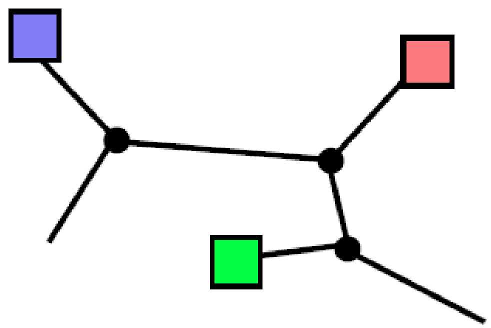
Channels, Network Links, Networks

- A *signal* is conveyed via a *channel*
  - May be directly conveyed
    - Electrical signals in a cable
  - May be modulated onto an underlying carrier
    - Radio
- The combination of signal and channel forms a *link*

- A link directly connects one or more hosts

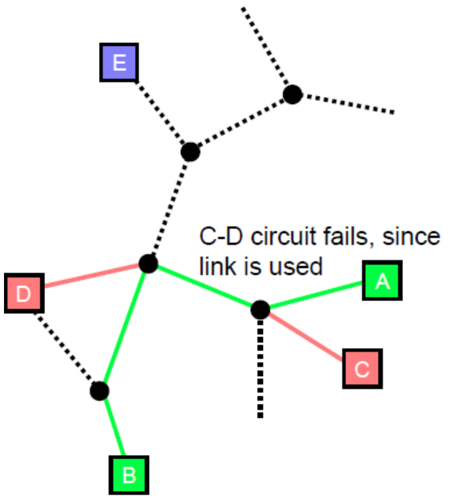


- A network comprises several links connected together
  - The devices connecting the links are called either *switches* or *routers* depending on the type of network

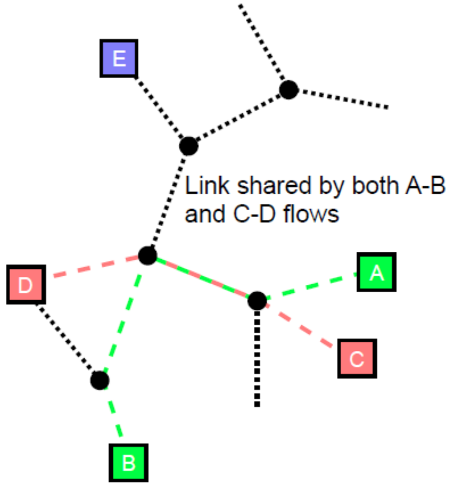


Network Switching

- Circuit Switched Networks
  - A dedicated circuit can be set up for A and B to communicate
    - A and B exchange arbitrary length messages
    - Guaranteed capacity once circuit is created
    - But – the dedicated circuit can block other communications (e.g. the C to D path); the capacity of the network gives the blocking probability
    - Example: traditional telephone network

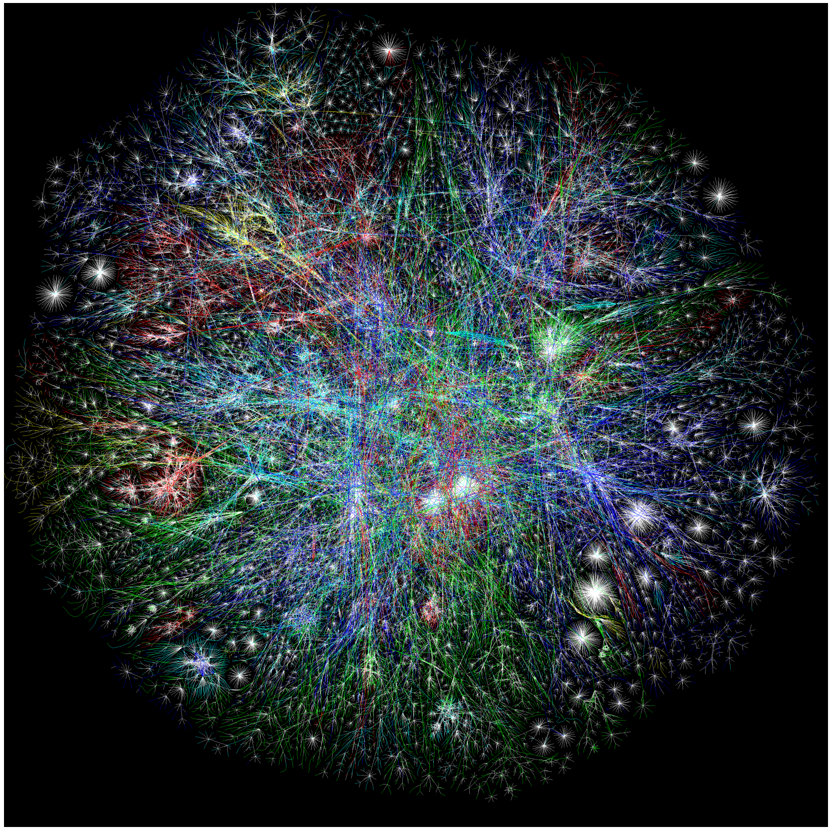


- Packet Switched Networks
  - Messages are split into small packets before transmission
    - Allows A-B and C-D to communicate at the same time, sharing the bottleneck link
    - Connectivity guaranteed, but the available capacity varies depending on how many other people are using the network
    - Packets are small, and have a size constraint; a message can consist of many packets
    - Example: the Internet



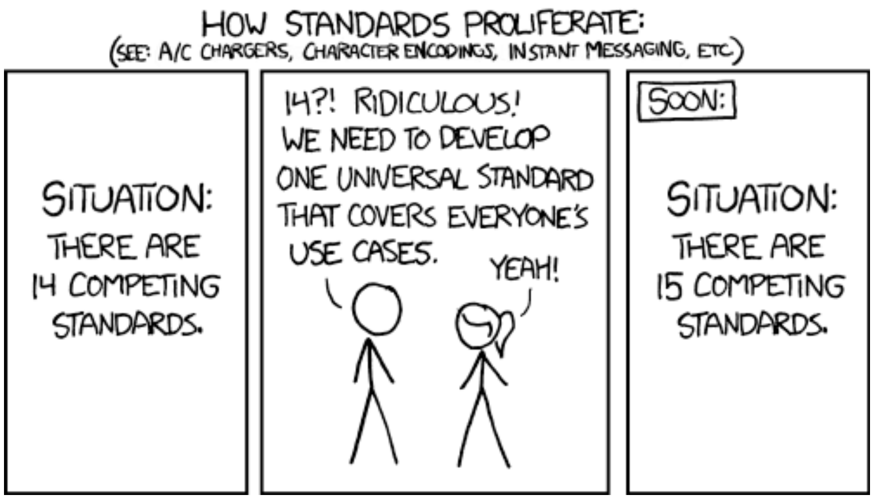
Networked Systems

- All networked systems built using these basic components:
  - Hosts – the source and destination(s)
  - Links – physical realisation of the channel, conveying messages
  - Switches/routers – connect multiple links
- Layered on top are network protocols which give meaning to the messages that are exchanged



Source: http://www.opte.org/maps/

Coming up next...



Source: https://xkcd.com/927/

The nice thing about standards is that you have so many to choose from.

(Andrew S. Tanenbaum)

Image source: http://www.linux-magazine.com/Online/News/Why-Can-t-Computers-Just-Work-All-the-Time