



# Networks, protocols and data formats

How computers exchange and  
store data and information



# Summary

This lecture is a summary of the following:

- Kernigan, Part III Communications, Networking, The Internet, The World Wide Web (pages 117-186)
- Juneday Wiki:
  - [http://wiki.juneday.se/mediawiki/index.php/ITIC:Networks\\_and\\_protocols](http://wiki.juneday.se/mediawiki/index.php/ITIC:Networks_and_protocols)

# Questions?

- Any questions on Kernigan?
- Any questions on the Wiki?

# Network basics

- Computers are connected via cables or radio
- Each physical device has a (preferably) unique numeric address
  - Mac address or “Ethernet address”
  - Looks like 9c:b6:d0:f2:e1:f9
- Protocols like TCP/IP allows for more advanced communication
  - IP numbers, domain names
- The actual data transferred uses some higher protocol, like SMTP, POP3, HTTP
  - The data for higher protocols depends on TCP/IP to put the sent packages together

# Names and numbers

- Computers are assigned an IP number (referring to the IP protocol)
  - Allows for communication over the local or international networks
  - Private IP numbers vs. public IP numbers
- Fully qualified domain names allows for easier-to-remember node names
- DNS is a service for looking up names and numbers
- TCP allows for the concept of “Ports” - one node can offer more than one network service
  - Some typical port numbers include HTTP on port 80, SSH on port 22 etc

# Names and numbers

- Look up your
  - Network card physical address
  - Network card IP number (is it a public or private IP number?)
  - Computers DNS settings
  - Favorite web site's IP number (using DNS)

# Data and information

- Of course, networks are used to communicate information
- Information is coded as data (ones and zeros)
- A computer (application) needs to know how to interpret the data
- Plain text is commonly used
  - Why? It can be understood by both humans and applications
- But how do we encode data as plain text?
  - What is the plain text for information about the next bus at some bus stop?

# Plain text data formats

- Standards have evolved
- We need to be able to both verify that the data has the correct format (isn't gibberish or corrupted) - the application can discard it otherwise, and is valid (makes sense according to defined schemas)
- There are some commonly used plain text formats for data transfer
  - XML
  - Json
  - CSV



# Data and metadata

Data sent or stored on a computer often has two parts:

- Metadata (data about the data)
- Raw data (facts, numbers etc)

XML:

```
<stop_name>Lindholmen</stop_name>  
  ^           ^  
 metadata    data
```

Json:

```
"stop_name": "Lindholmen"  
  ^           ^  
 metadata    data
```

# XML

- Elements
  - start-tag
  - content
  - end-tag
    - can be empty - short form: `<element-name/>`
- Tag
  - name
  - attributes (name="value")
  - start-tag: `<bus_stop>`
  - end-tag: `</bus_stop>`
- Elements can be nested

# XML example

```
<?xml version="1.0" encoding="UTF-8"?>
<next_bus>
  <bus_stop>Lindholmen</bus_stop>
  <bus_number accessible="true">16</bus_number>
  <direction>Högsbohöjd</direction>
  <arrival>2019-07-02T10:31:53+02:00</arrival>
</next_bus>
```

# Json

- Objects (between { and } )
  - key-value pairs
  - key: a double-quoted name
  - value: String, number, boolean, null
- Arrays
  - Comma-separated list of values or objects

# Json example

```
{  
  "bus_stop": "Lindholmen",  
  "bus_number": "16",  
  "accessible": true  
  "direction": "Högsbohöjd",  
  "arrival": "2019-07-02T10:31:53+02:00"  
}
```

# Protocols and data

- Computer communicate using protocols
- Data sent has a format
- Examples:
  - Using HTTP to send json
  - Using HTTP to send XML
  - Using SMTP to send an image
  - Using FTP to send a sound file
  - Using SCP to send a PDF
  - Using IRC to send plain text messages
- At a lower level, Internet uses TCP/IP so that the data arrives to the right place, and to manage connections (open/close, ports etc)