Networks, protocols and dataformats

How computers communicate and exchange data

‹video›

Network

- Computers are connected in networks
- Local, local or wide area, international
- Programs on two computers communicate using *protocols*
- Computers on a network have addresses
- Programs on two computers can exchange data
- There are several standard data formats for data exchange

Routers and gateways

- When you connect to a network you get an address from a local router
- The router is responsible for routing traffic in the network and to other networks
- On the Internet, routers make sure traffic gets where it should
- The computer between a local network and e.g. Internet is called a *gateway*
- Gateways are the entry and exit points between networks
- Routers are the traffic control points within networks

Wireless networks

- You should only connect to encrypted and trusted wireless networks
- The router providing access to a wireless network can be called *access point*, *wireless router*, *hotspot* or *wireless gateway* (the latter if it also connects you to e.g. the Internet)
- Without encryption, anyone on the same network can see all traffic to and from your computer (which may or may not be encrypted) and what computers you communicate with

IP address

- In order to communicate, the computers need to have addresses
- When you connect to a network, your computer broadcasts its hardware address of its network card
- A router runs a server program which detects your computer and assigns it a unique *IP Address* and often some other settings necessary for you to use the network
- On the Internet, the IP protocol (Internet Protocol) which is the most basic protocol (contains metadata, sender address and destination address)
- Other protocols handles the actual transmission and communication (e.g. TCP - Transmission Control Protocol)

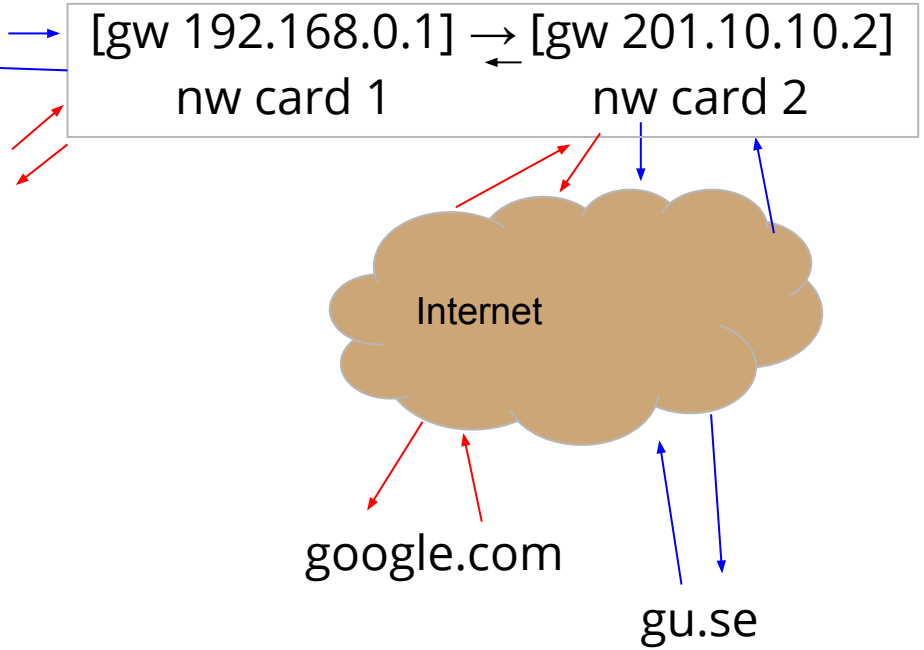
Private networks and public networks

- Locally, you will get an IP address which only makes sense on your local network
- The gateway will have two network cards
  - One with a local IP address for local communication
  - One with a public IP address for communicating with the Internet
- All computers on the local network will share the gateway's public IP address when communicating on the Internet
- The gateway will keep track of which local computer talks to what computer on the Internet

Private networks and public networks

[local computer1 192.168.0.100]
talks to gu.se

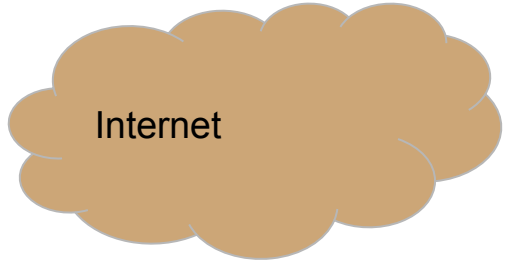
[local computer2 192.168.0.101]
talks to google.com



Private networks and public networks

[local computer1 192.168.0.100]
talks to computer2

[local computer2 192.168.0.101]



google.com

gu.se

IPv4 Addresses

- Four *octets* (four eight bit numbers)
- Eight bit unsigned numbers (0-255)
- Examples 10.0.116.35 (a private IP), 130.241.151.114 (public IP)
- Local/private ranges:
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255
- IPv4 addresses use 32 bits, which is 4 294 967 296 addresses (2^{32})
- Will it suffice?

IPv6 addresses

- Eight 16 bit numbers (expressed in hex to save some typing)
- Use 128 bits, 2^{128} addresses
- How many is that?
- 3.403×10^{38} - more than 3 followed by 38 zeros
- Example: fe80::de1f:acad:148:d9b4 (Rikard's computers network card)
- Zeros are truncated - :: means :0000:

Domain Name System

- Numbers add up to nothing and are hard to remember
- Humans remember names better
- The DNS translates between names and numbers
- A Fully Qualified Name (FQN) like wiki.juneday.se uniquely identifies a computer on the Internet
- To contact it, your computer needs the IP address
- You got a DNS IP address from your router and that will be used

Example - try it!

```
$ nslookup wiki.juneday.se
Server:    127.0.1.1
Address:   127.0.1.1#53

Non-authoritative answer:
Name:  wiki.juneday.se
Address: 130.241.135.117

$ host wiki.juneday.se
wiki.juneday.se has address 130.241.135.117

$ dig +short wiki.juneday.se
130.241.135.117
```

Ports

- A computer can offer more than one service at the same time
- A service, or server, is an application many *clients* can use at the same time
- A computer can both run a web server and a file server
- Since the computer has only one IP address, we need to specify what service we want when connecting to it
- Ports are used - a number specifying a service on the computer
- A bit like an apartment number - same street address can have many apartments

Port numbers are a great help

John Smith
Mulholland Dr. 3

Confused mailman

John Smith

John Smith

Jane Doe

John Doe

Börje Langefors

Mulholland Drive 3 postboxes

Some standard port numbers

- 20: File Transfer Protocol (FTP) (Data Transfer)
- 21: File Transfer Protocol (FTP) (Command Control)
- 22: Secure Shell (SSH)
- 23: Telnet remote login service, unencrypted text messages
- 25: Simple Mail Transfer Protocol (SMTP)
- 53: Domain Name System (DNS)
- 80: Hypertext Transfer Protocol (HTTP) (web)
- 110: Post Office Protocol (POP3)
- 123: Network Time Protocol (NTP)
- 143: Internet Message Access Protocol (IMAP)
- 194: Internet Relay Chat (IRC)
- 443: HTTP Secure (HTTPS) "HTTP over TLS/SSL"

Protocols - DHCP

- Dynamic Host Configuration Protocol
- Your computer joins a network
- It broadcasts its hardware address
- A DHCP server (probably on the router) picks it up and configures an IP address and DNS etc and sends that to your computer

DNS

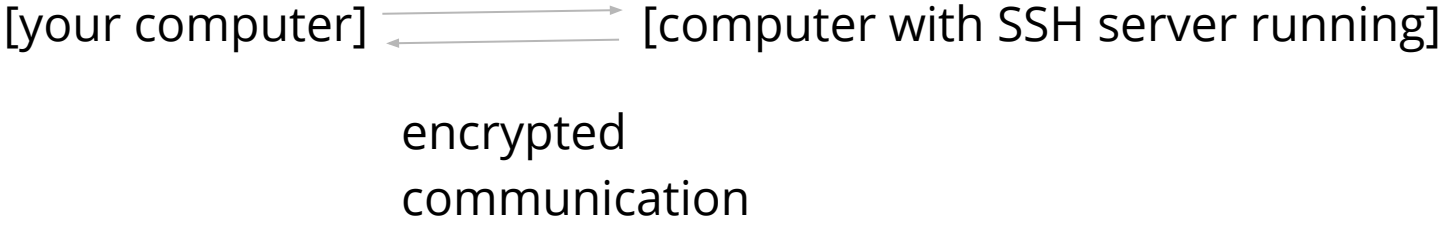
- Domain Name System
- Let’s you look up an IP from a FQN (domain name) and vice versa
- You can try the tools host, dig, nslookup

HTTP(S)

- HyperText Transfer Protocol (Secure)
- The way a client (e.g. a browser) requests a file from a web server, which responds with the file (if it exists)
- Hypertext refers to links between documents
- Basis for the world wide web

SSH

- Secure SHell
- Let’s you login to a remote computer securely (encrypted)
- Also used for copying files and issuing a command on the remote computer
- The *host* runs an SSH server
- The *client* runs an SSH client



SMTP

- Simple Mail Transfer Protocol
- Used for sending and relaying mail
- Your email program has configured an SMTP server for outgoing mail
- The SMTP server will probably relay the mail to a second SMTP server for the domain of the recipient

POP

- Post Office Protocol
- Used for fetching or polling new mail to your computer

Some words on encryption

- There’s data (which can be encrypted)
- Then, there’s metadata (which can’t be encrypted) about who you are communicating with
- If you want to keep who you are contacting private, you should consider a VPN (virtual private network)
- A VPN treats all of its connected computers as the same, so that it looks like the VPN gateway talks to computers, when in fact, many computers may be the source
- Same goes for email - you may encrypt the message, but you can’t hide the fact that you are sending it and to whom

Data

- The reason we use networks is so that we can exchange data
- How should we represent the data?
- There are standard data formats that are commonly used
- Example: A system on the network needs information about customers from another system, also on the network. The customer data needs to be represented so that the first system can make sense of it
- Typically data consists of metadata (data about the data) and the actual data (facts and figures etc)
- Using a standard format for this reduces vendor lock-in

Customer data example

- Metadata and data:
  - Customer number (a six digit whole number)
  - Customer name (first name, last name as plain text)
  - Customer email (plain text email address)
- This can be represented in various formats, such as XML, Json or CSV
- We can use HTTP as the protocol for requesting data and responding with the data

XML

- eXtensible Markup Language - plain text markup
- Uses *tags* for metadata surrounding actual data
- An *element* is a *start tag* and possible data and an *end tag*
  `<firstname>Henrik</firstname>`
- Start tags can have *attributes* (typical for metadata)
  `<email type="work">`
- Elements can be nested

```
<name>
  <firstname>Henrik</firstname>
  <lastname>Sandklef</lastname>
</name>
```

XML example

```
<customers>
  <customer number="000001">
    <name>
      <firstname>Henrik</firstname>
      <lastname>Sandklef</lastname>
    </name>
    <email type="work">henke@gu.se</email>
    <email type="home">henke@henkeshome.se</email>
  </customer>
  <customer number="000002">
    <name>
      <firstname>Rikard</firstname>
      <lastname>Fröberg</lastname>
    </name>
    <email type="work">rille@gu.se</email>
    <email type="home">rille@rilleshome.se</email>
  </customer>
</customers>
```

XML example - xml declaration

- The very first line of an XML document should be an XML declaration which can look like this: `<?xml version="1.0" encoding="UTF-8"?>`
- The first opening tag of the document starts what is called the *root element*

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer number="000001">
    <name>
      <firstname>Henrik</firstname>
      <lastname>Sandklef</lastname>
    </name>
    <email type="work">henke@gu.se</email>
    <email type="home">henke@henkeshome.se</email>
  </customer>
  <customer number="000002">
    <name>
      <firstname>Rikard</firstname>
      <lastname>Fröberg</lastname>
    </name>
    <email type="work">rille@gu.se</email>
    <email type="home">rille@rilleshome.se</email>
  </customer>
</customers>
```

More on XML on the wiki

http://wiki.juneday.se/mediawiki/index.php/Web:Introduction_to_XML

Json

- JavaScript Object Notation
- Also plain text
- A very common data format
- Less verbose than XML
- Looks more like programming code

```
{
  "firstName": "Rikard"
}
```

Objects and lists of objects

- An object consists of name-value pairs, or just values
- Names are always quoted
- Values of type text are quoted, numbers are not
- Some literals: `true false null`
- Lists of objects are called *arrays* and use square brackets

```
"colors": [ "red", "green", "blue"]
```

- Arrays can have objects as elements

Json customer example

```
{ "customers":[
  {
    "customerNumber": "000001"
    "firstName": "Henrik",
    "lastName": "Sandklef",
    "email": "henke@gu.se"
  },
  {
    "customerNumber": "000001"
    "firstName": "Rikard",
    "lastName": "Fröberg",
    "email": "rille@gu.se"
  }]
}
```

Json customer example - anonymous array

```
[
  {
    "customerNumber": "000001"
    "firstName": "Henrik",
    "lastName": "Sandklef",
    "email": "henke@gu.se"
  },
  {
    "customerNumber": "000001"
    "firstName": "Rikard",
    "lastName": "Fröberg",
    "email": "rille@gu.se"
  }
]
```

More on Json on the wiki

http://wiki.juneday.se/mediawiki/index.php/Web:Introduction_to_JSON

CSV

- Comma Separated Values (plain text)
- Metadata in the first line (like headers)
- Data separated by commas, order match the headers
- You need to handle data that contain commas (use quotes)
- Small and simple

```
customer_number,first_name,last_name,email
000001,Henrik,Sandklef,henke@gu.se
000002,Rikard,Fröberg,rille@gu.se
```

Pros and cons of CSV

Pros:

- Small and simple
- Can be opened by database management systems and spreadsheet apps
- Easy for computers to “parse” (make sense of)

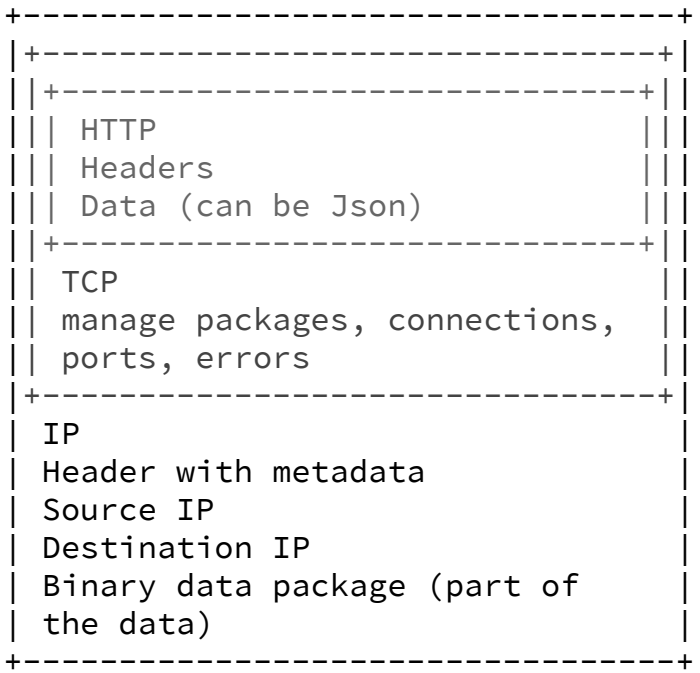
Cons:

- Commas in data need to be handles
- Not clear what type data has - is 123 a string or a number? 0x123? 0.123?

Relation between protocol and data format

- Protocols are how an application communicates with another application
- Data is send *over a protocol* (or *using a protocol*)
- You can use HTTP to request a Json file from a web server
- Plain text data formats can be understood by both humans and computers, which makes HTTP and the web suitable (who doesn't have a browser?)

Simplified model of protocol and data



‹/video›