

## Networks and Interprocess Communication

Johan Montelius

KTH

HT15

- Performance
- Scalability
- Reliability
- Security
- Mobility
- Quality of Service
- Multicasting

1 / 28

2 / 28

## Types of networks

## latency

- WAN - Wide Area Networks
- MAN - Metropolitan Area Networks
- LAN - Local Area Networks
- PAN - Personal Area Networks

Transfer rate:

What is the rate at which we can send data?

- Latency - how long time does it take to send an empty message?
- Transfer rate - what is the rate at which we can send data?

Why does it take time to send a message?

- distance - speed of signal (light)
- access - granting of resource
- routing - processing in nodes

5 / 28

6 / 28

fast as ..

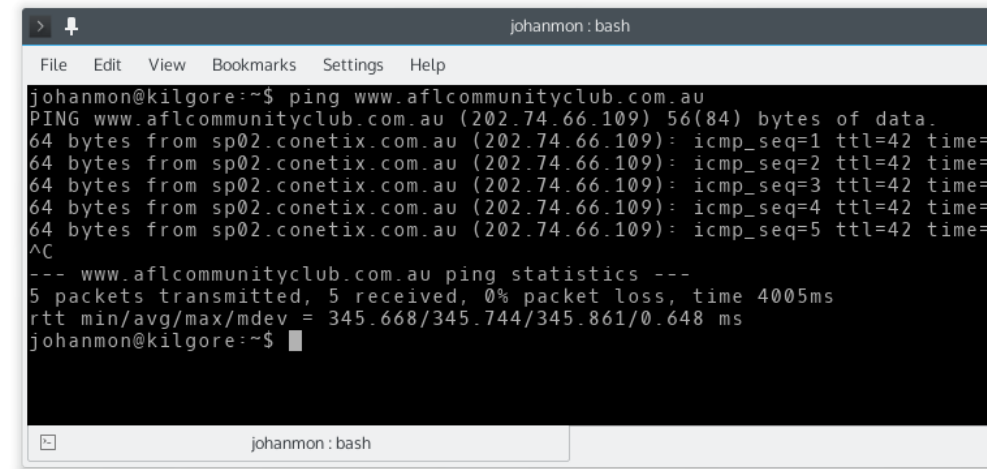
ping

What is the speed of light?

Distance in ms:

- Stockholm - Hamburg
- Stockholm - NYC
- Stockholm - Melbourne

*Routers, switches and fiber optics adds to this so Melbourne is aprx 300 ms away.*



```
johanmon : bash
File Edit View Bookmarks Settings Help
johanmon@kilgore:~$ ping www.aflcommunityclub.com.au
PING www.aflcommunityclub.com.au (202.74.66.109) 56(84) bytes of data:
64 bytes from sp02.conetix.com.au (202.74.66.109): icmp_seq=1 ttl=42 time=
64 bytes from sp02.conetix.com.au (202.74.66.109): icmp_seq=2 ttl=42 time=
64 bytes from sp02.conetix.com.au (202.74.66.109): icmp_seq=3 ttl=42 time=
64 bytes from sp02.conetix.com.au (202.74.66.109): icmp_seq=4 ttl=42 time=
64 bytes from sp02.conetix.com.au (202.74.66.109): icmp_seq=5 ttl=42 time=
^C
--- www.aflcommunityclub.com.au ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 345.668/345.744/345.861/0.648 ms
johanmon@kilgore:~$
```

*Using ICMP packages might give a better value, UDP might be slower.*

7 / 28

8 / 28

- LAN/WLAN - local area networks (Ethernet/WiFi)
- WAN - wide area networks (IP routed)
- Mobile networks
- Satellite (geo-stationary)

How does latency vary with the size of the messages?

9 / 28

10 / 28

The rate at which we can send data (does not mean that it has arrived).

What is the transfer rate of:

- ADSL
- Ethernet
- 802.11
- 3G/4G

Is this shared with others?

- medium access: 802.11 - RTS, CTS
- error handling: detection, forward error correction, ARQ
- header: MAC header, IP header, TCP ...
- flow control: TCP window

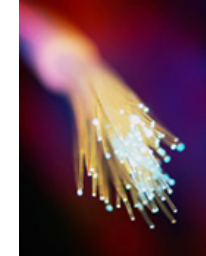
Stockholm to Gothenburg - 400 km, best possible data communication layer?

The application layer transfer rate is much lower than the physical layer bit rate.

How does the application layer latency differ from the network layer latency?



100  $m^3$  or five million BlueRay 50Gbyte disks, delivered in 6 h, two trucks every day



10 Gbit/s

13 / 28

14 / 28

## Communication layers

- application: the end product
- presentation: encoding of information, serialization, marshaling
- session: security, authentication, initialization
- transport: messages, streams, reliability, flow control
- network: addressing of nodes in a network, routing, switching
- data link: point to point deliver of frames, medium access, link control
- physical layer: bits to analog signals, electrical, optical, radio ...

## Internet stack

- HTTP, FTP, SMTP
- TCP, UDP, SCTP, ICMP
- IP, ARP
- Ethernet, WiFi, ..

15 / 28

16 / 28

What would the world look like ...

.. if we only had Ethernet?

Two approaches:

- Distance vector: send routing table to neighbors, RIP, BGP
- Link state: tell everyone about your direct links, OSPF

Pros and cons?

17 / 28

## IP addresses

What is the structure of an IP address?

How would you allocate IP addresses to make routing easier?

What is actually happening?

## UDP and TCP

18 / 28



One word that describes the difference between UDP and TCP.

19 / 28

20 / 28

Introduces two communication abstractions:

- UDP: datagram
- TCP: stream
- Gives us port numbers to address processes on a node.
- About hundred other protocols defined using IP. (ICMP, IGMP, RSVP, SCTP...)
- More protocols defined on top of UDP and TCP.
- A datagram abstraction, independent messages, limited in size.
- Low cost, no set up or tear down phase.
- No acknowledgment.

21 / 28

22 / 28

- A duplex stream abstraction.
- Reliability, lost or erroneous packets are retransmitted.
- Flow control, to prevent the sender from flooding the receiver.
- Congestion friendly, slows down if a router is choked.
- UDP: small size messages, build your own streams
- TCP: large size messages, flow control of a stream of messages

*Can you trust TCP delivery?*

Sockets is the programmers abstraction of the network layer:

- datagram sockets for messages (UDP)
- stream sockets for duplex byte streams (TCP)

- server
  - Create a *listen socket* attached to a port (could be in several steps: create, bind, listen)
  - Accept incoming request and create a *communication socket*, this is the socket used for reading/writing.
- client
  - Create a *communication socket* and connect to a server given a specified port.
  - Read and write from socket.

25 / 28

## A server in Erlang

```
init(Port) ->
  case gen_tcp:listen(Port, [..]) of
    {ok, Listen} ->
      handler(Listen),
      gen_tcp:close(Listen);
    {error, Error} ->
      error
  end.

handler(Listen) ->
  case gen_tcp:accept(Listen) of
    {ok, Client} ->
      request(Client),
      handler(Listen);
    {error, Error} ->
      error
  end.
```

27 / 28

26 / 28

## A server in Erlang

```
request(Client) ->
  case gen_tcp:recv(Client, 0) of
    {ok, Request} ->
      Response = reply(Request), a byte sequence
      gen_tcp:send(Client, Response);
    {error, Error} ->
      error
  end,
  gen_tcp:close(Client).
```

28 / 28

- server
  - Create *message socket* and bind to port.
  - Read an incoming message (message contains source address and port).
- client
  - Create *message socket* with source port.
  - Create message and give destination address and port.
  - Send message.

How do we transform internal datstructure into sequencing of bytes?

- Language dependent: Java serialization, Erlang external term format
- Independent: XML, Google Protocol Buffer, ASN.1
  - message format defined by specification: XML Schema, .proto, ...
  - specification is used by compiler to generate encoder and decoder

29 / 28

30 / 28

## example

ANS.1 specification	C data structures
FooProtocol DEFINITIONS ::= BEGIN	int foo_question {
FooQuestion ::= SEQUENCE {	int tracking_number;
trackingNumber INTEGER,	char question[128];
question IA5String}	}
FooAnswer ::= SEQUENCE {	int questionNumber = {5, "Anybody there?"};
questionNumber INTEGER,	BOOLEAN}
answer	
END	

30 13 02 01 05 16 0e 41 6e 79 62 6f 64 79 20 74 68 65 7

## summary

The application layer should in a perfect world be independent of underlying layers.

The world is not perfect.

Understanding underlying network characteristics is essential when developing distributed applications.