Distributed Systems

ID2201

Networks and network protocols

Johan Montelius

Requirements

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

Performance

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

What is latency?

- Why does it take time?
  - physical distance
  - routing delay

Fiber all the way :-)

- What is the speed of light?
  - 300 000 km/s
  - or 300 km/ms
- Distance in ms:
  - Stockholm - Hamburg
    - aprx 800 km or 3 ms
  - Stockholm - NYC
    - aprx 6600 km or 23 ms
  - Stockholm - Melbourne
    - aprx 15600km or 52 ms

How to measure latency?

```
johanmon@ktrout:~$ ping www.google.com
PING www.google.com (173.194.69.105) 56(84) bytes of data.
64 bytes from bk-in-f105.1e100.net (173.194.69.105): icmp_req=1 ttl=46 time=24.3 ms
64 bytes from bk-in-f105.1e100.net (173.194.69.105): icmp_req=2 ttl=46 time=24.3 ms
64 bytes from bk-in-f105.1e100.net (173.194.69.105): icmp_req=3 ttl=46 time=24.3 ms
^C
--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 24.301/24.320/24.356/0.025 ms
johanmon@ktrout:~$
```

Uses ICMP over IP, not the same as UDP over IP!

Typical networks

- LAN - local area networks (Ethernet)
  - 1 - 10 ms
- WAN - wide area networks (IP routed)
  - 20 - 400 ms
- WLAN -wireless LAN (WiFi)
  - 5 - 10 ms
- Mobile networks
  - 40 - 800 ms
- Satellite
  - geo-stationary, > 250 ms

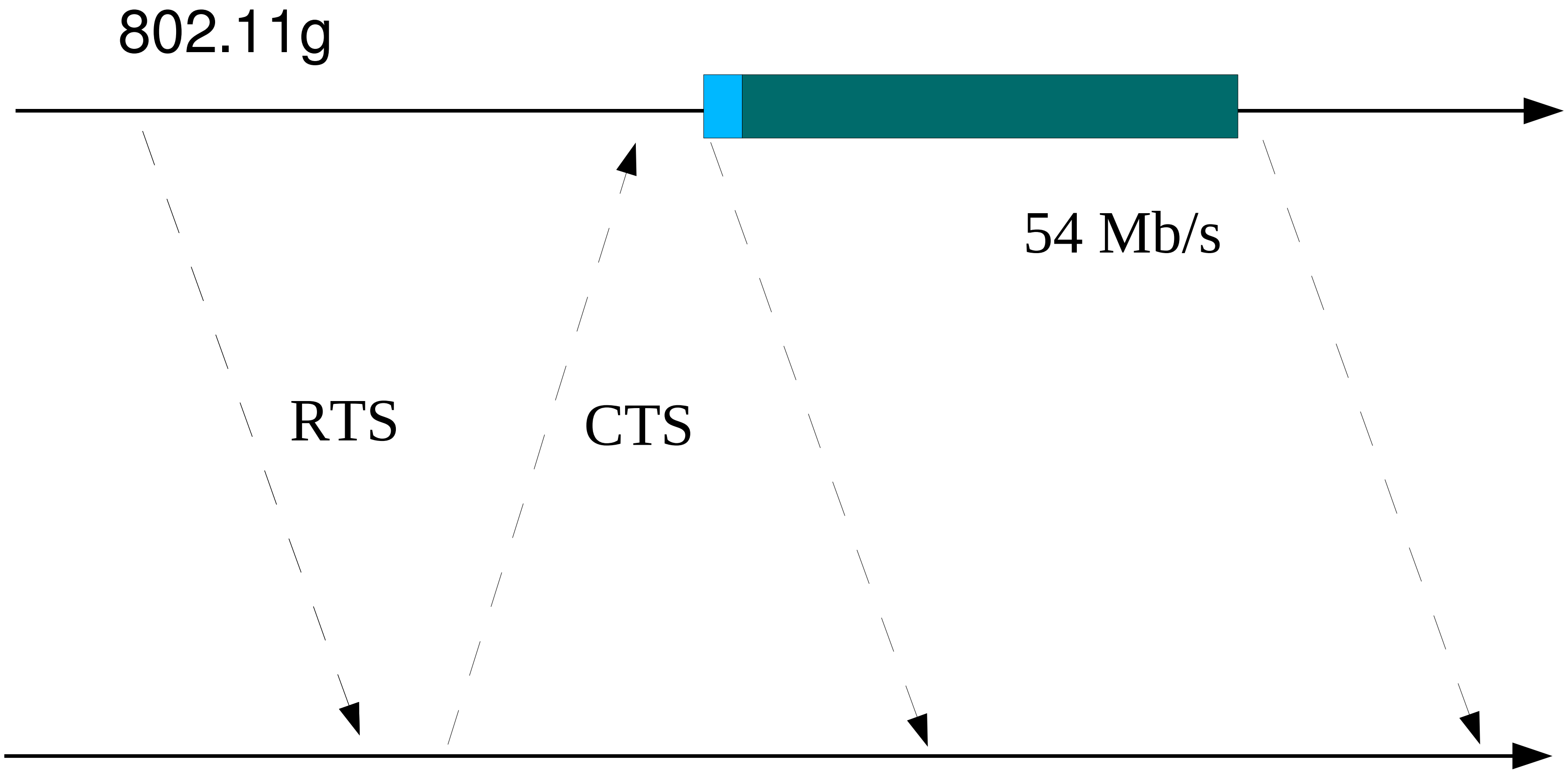
Latency

- How does latency vary with the size of messages?
  - it does not
  - the larger the message the longer the latency

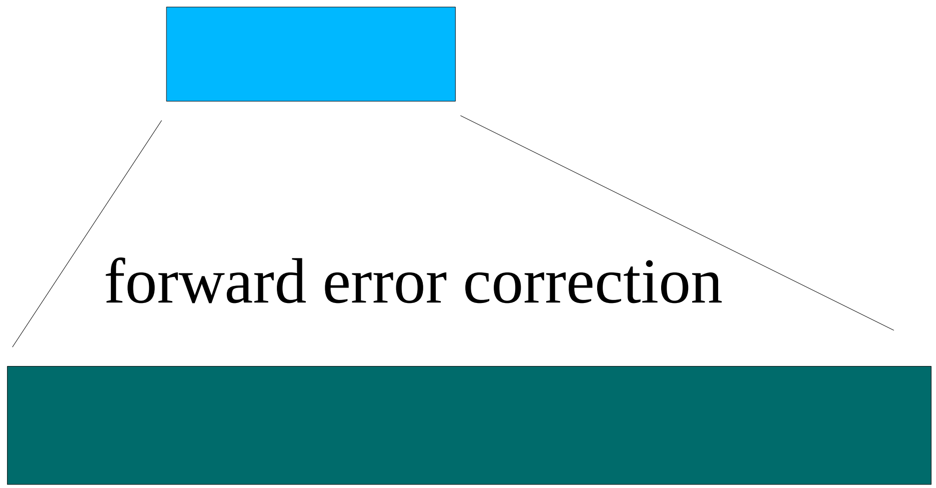
Transfer rate

- The rate at which we can send data (does not mean that it has arrived).
- What is the transfer rate of:
  - ADSL
    - 1 - 20 Mb/s
  - Ethernet
    - 100 Mb/s - 1 Gb/s
  - 802.11
    - 11 Mb/s, 54 Mb/s, 72 Mb/s ...
  - 3G/4G
    - 1 Mb/s, 2 Mb/s, ... 100 Mb/s

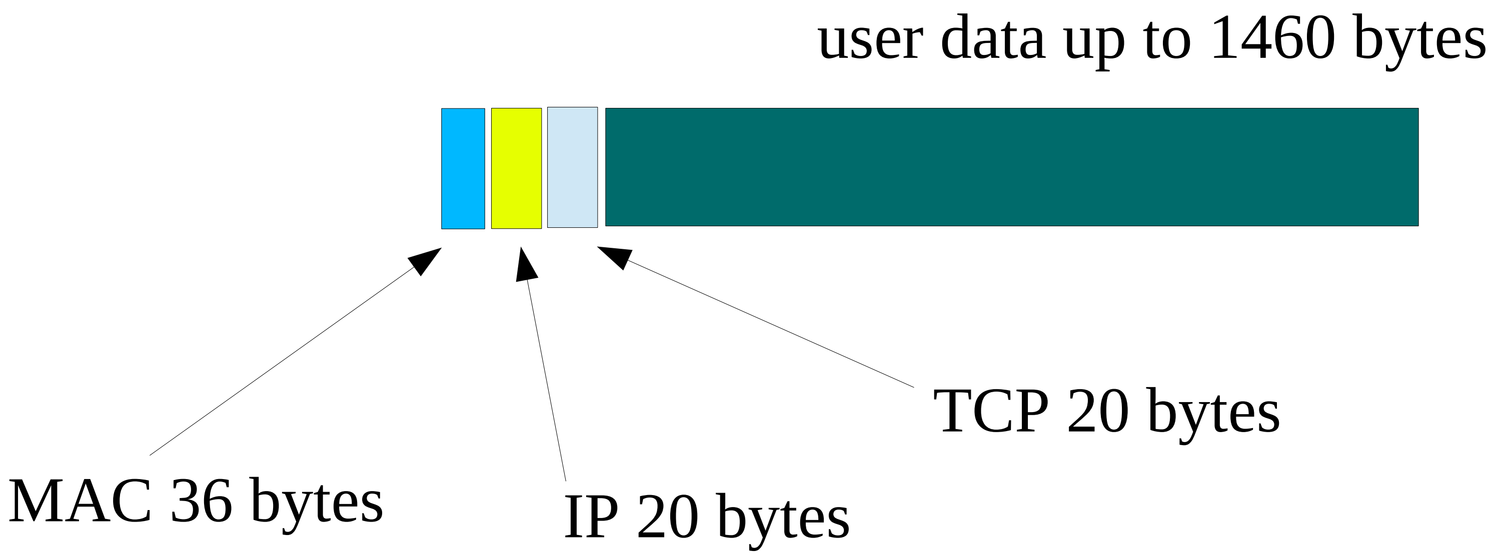
medium access overhead



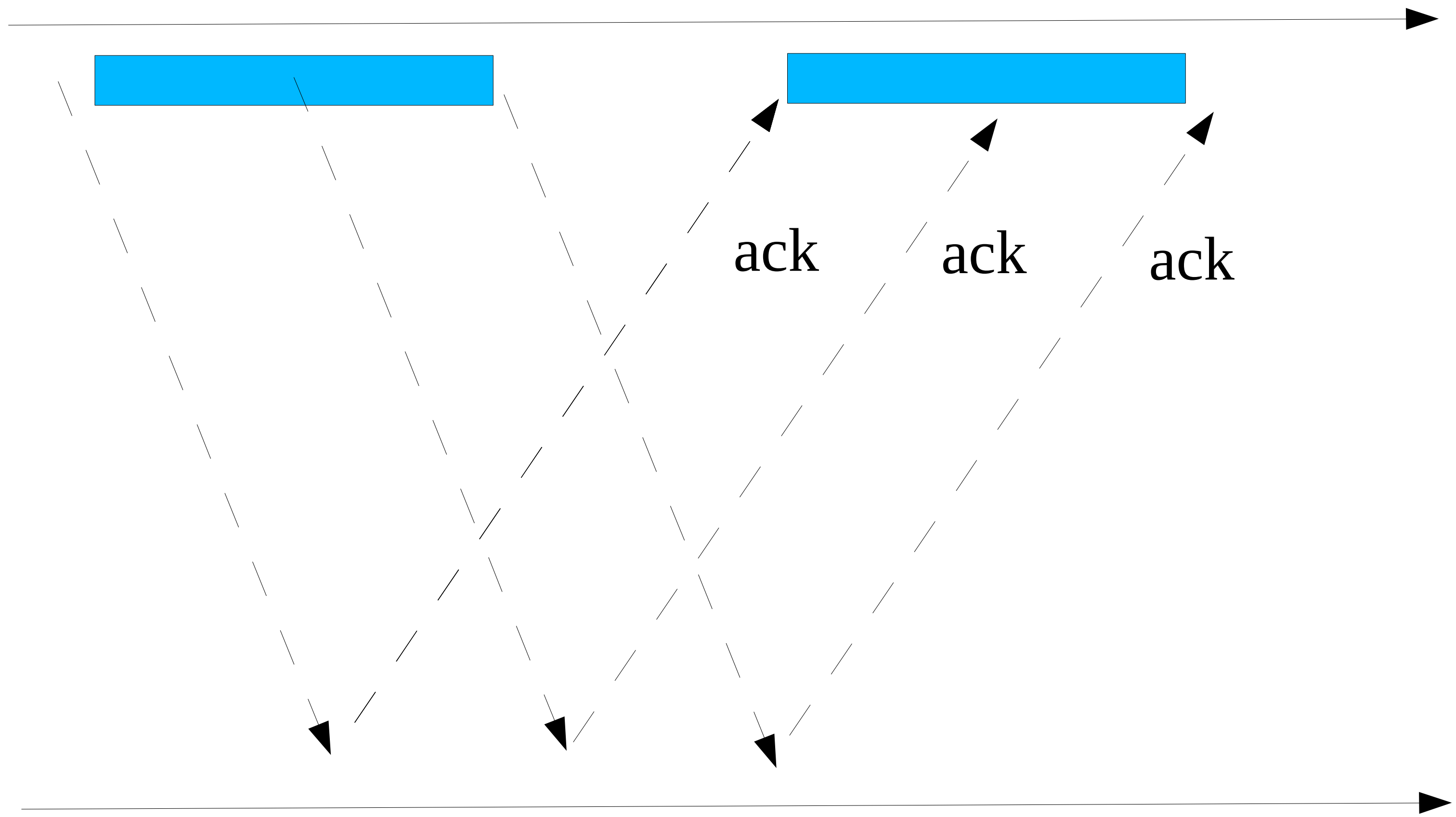
error coding overhead



header overhead



flow control



transfere rate

- Taken overhead into account:
  - the maximum transfer rate is much lower than the maximum signaling rate

This is easy...

KTH
to
Chalmers

one trip per day, 120 m^3

speed of light, 10 Gb/s

Communication layers

Application: ...

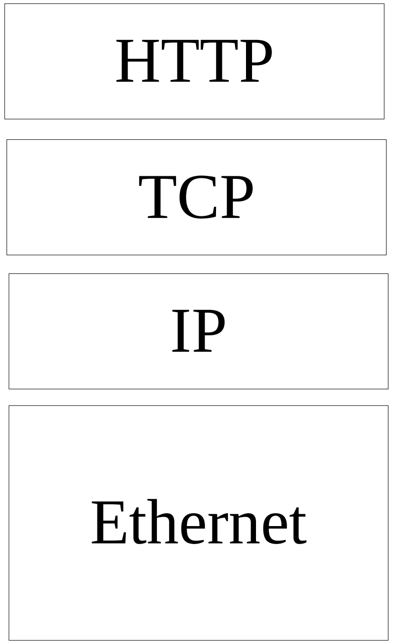
Transport: messages, streams, host-to-host, reliability, flow control,...

Network: addressing, frames, switching, routing, ...

Data link: medium access, frames acknowledgement, error correction...

Physical layer: how are bits turned into signals: electrical, optical, ...

Which layers



Hubs and switches

- What is the difference between a hub and a switch?

Packet vs circuit switching

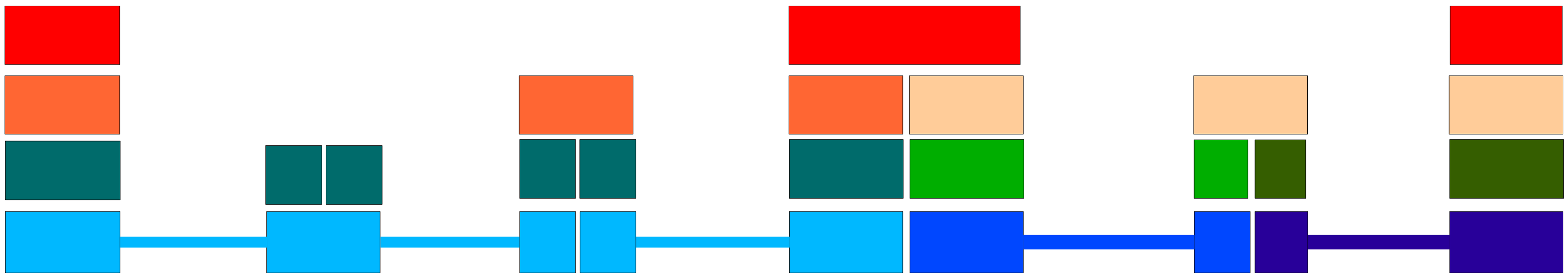
- What are the pros and cons of packet vs circuit switching?
  - Which scheme will take advantage of increased computing power?
  - Which scheme will take advantage of increased link capacity?

What would the world look like...

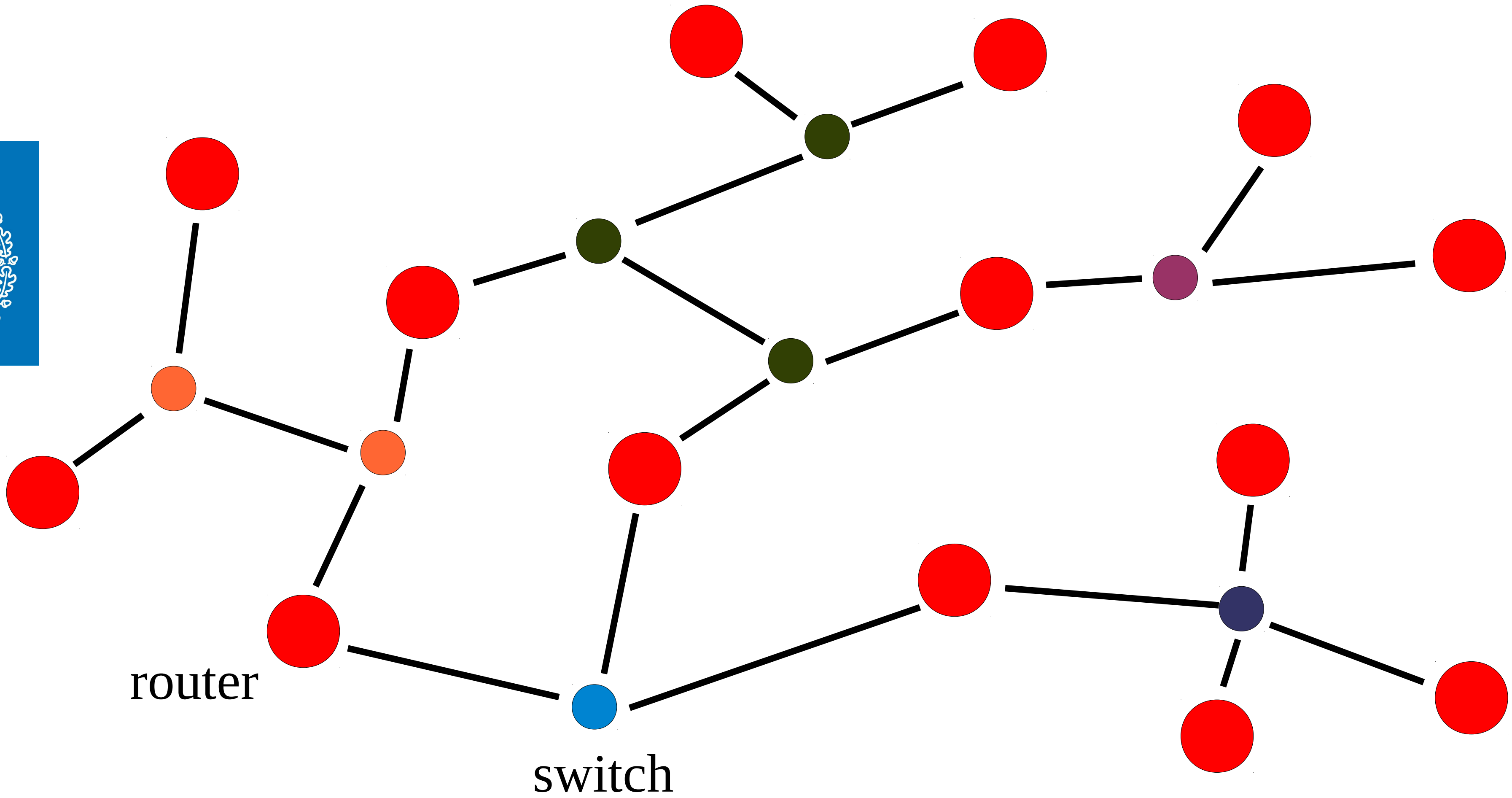
- .. if we only had Ethernet

Thank god for IP...

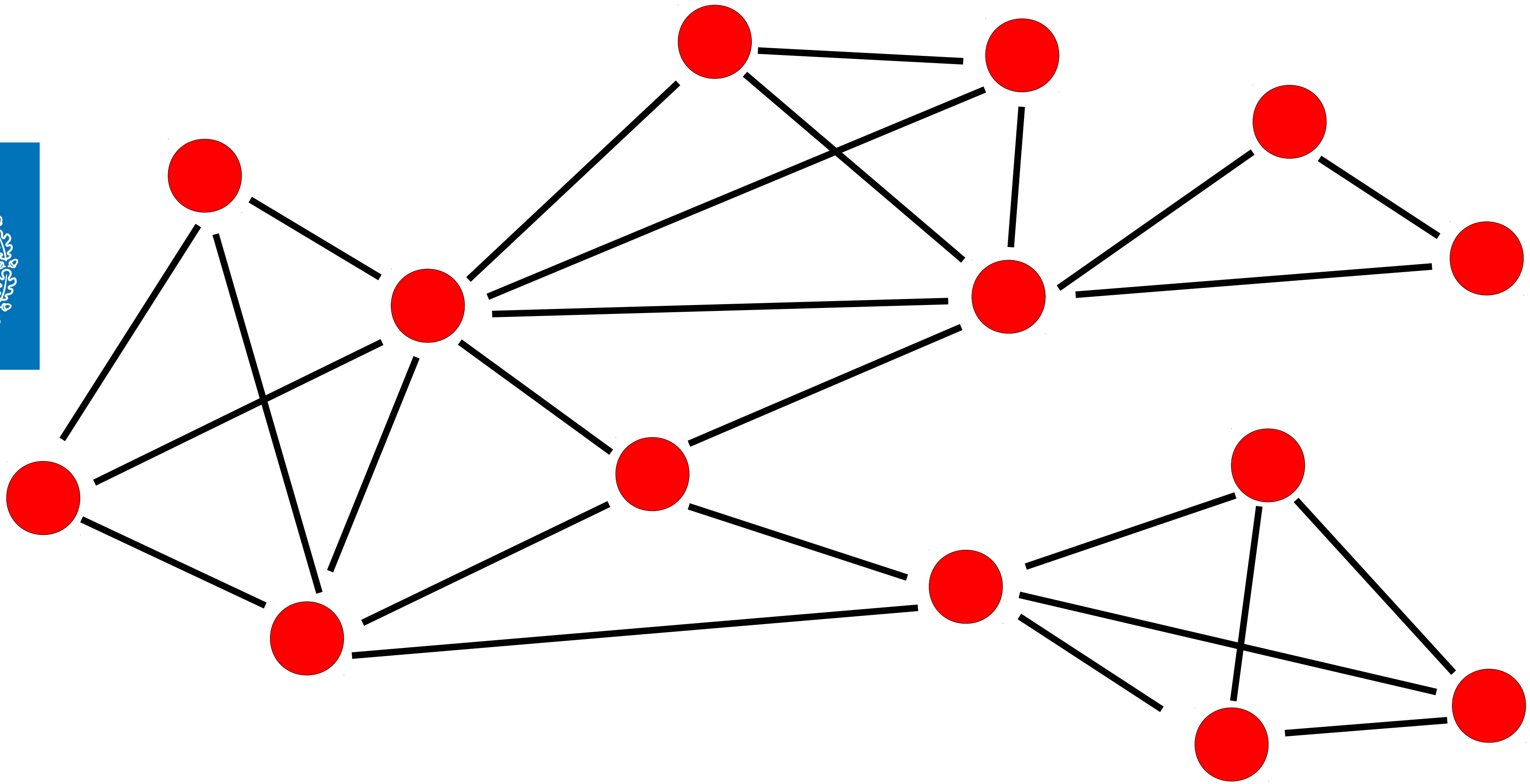
- .. but what does it give us?
- What is the job of a router and how is it different from a switch?



Physical connection



Logical connection



Routing

- Two approaches:
  - Distance vector: send routing table to neighbors, RIP, BGP
  - Link state: tell everyone about your direct links, OSPF
- Pros and cons?

IP addresses

- What is the structure of an IP address?
- How would you allocate IP addresses to make routing easier?
- What is actually happening?

http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml

Mobility

- What do we do when nodes move?

IP short comings

- IP routing is unpredictable
  - why?
- IP datagrams can be read by anyone and can originate from anyone
  - we would like to have some privacy and authentication
- IPv4 address space is too small
  - IPv6 is a solution

IP is not enough

- IP datagrams have a max size
  - we would like to send arbitrary large messages
  - large datagrams are fragmented
- IP datagrams are one-way
  - we would like to have a duplex communication
- IP addresses identify network interfaces
  - not a process

UDP and TCP

- 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.

UDP

- A <u>datagram abstraction</u>
  - independent messages
  - limited in size (what is the limit?)
- Low cost
  - No set up or tear down phase.
- No acknowledgment
  - How do we know it was received?

TCP

- A duplex stream abstraction.
  - The stream is divided into a ordered sequence of packets.
- 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 or TCP?

- UDP
  - small size messages
  - build your own streams
- TCP
  - large size messages
  - retransmission can be allowed
  - confirmed delivery?

UDP and TCP

One word that that describes
the difference between UDP and TCP.

TCP – a reliable protocol?

- If the network is down TCP will of course not be able to send anything
- If a network goes down the sender does not know if a segment has arrived or not.
- An *ack message* means that a byte sequence has been received and is now in the receivers buffer. It does not mean that a message has been handled by the receiving process.

Sockets

- Sockets is the programmers abstraction of the network layer:
  - datagram sockets for messages (UDP)
  - stream sockets for duplex byte streams (TCP)

Stream Socket

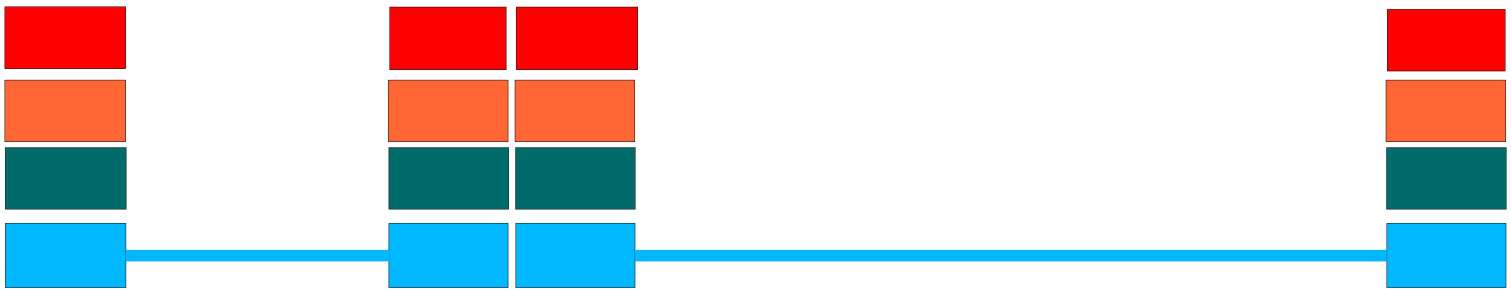
- Server:
  - Create a *listen socket* attached to a port
    - could be in several steps: create, bind, listen
  - Accept incoming request and create a *communication socket*
    - this is the socket used for reading/writing
- Client
  - Connect to a server given a specified port.
    - this is the socket used for reading/writing

Datagram Socket

- 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

NAT/NAPT

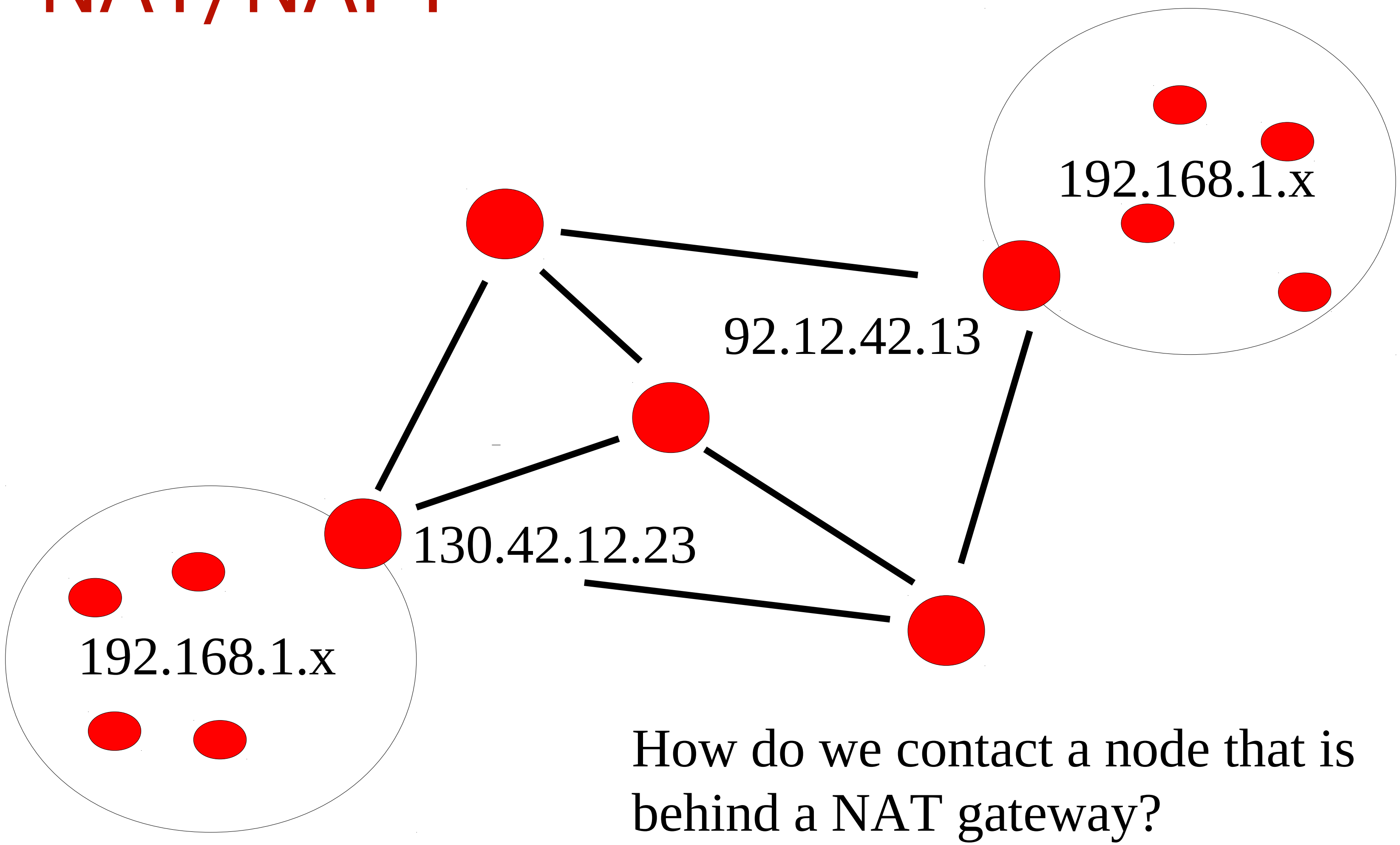
Gateway translates all local IP addresses to the IP address of the gateway.



Node has local IP address that is only useful inside own network.

Remote node thinks it's communicating with gateway.

NAT/NAPT



How do we contact a node that is behind a NAT gateway?