Distributed Systems

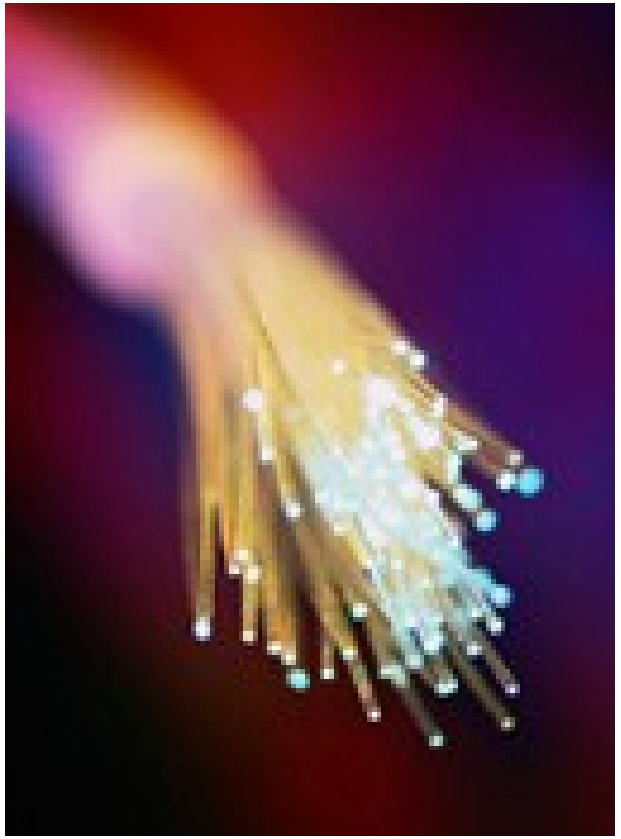
ID2201

Networks and network protocols

Johan Montelius

This is easy...

KTH
to
Chalmers



one trip per day, 120 m^3

speed of light, 10 Gb/s

What is latency?

- How do you define latency?
- How does latency vary with the size of the message?
- What are the typical latencies in:
  - Ethernet LAN or WAN
  - routed network, in Sweden or across the Atlantic
  - mobile networks
  - satellite links

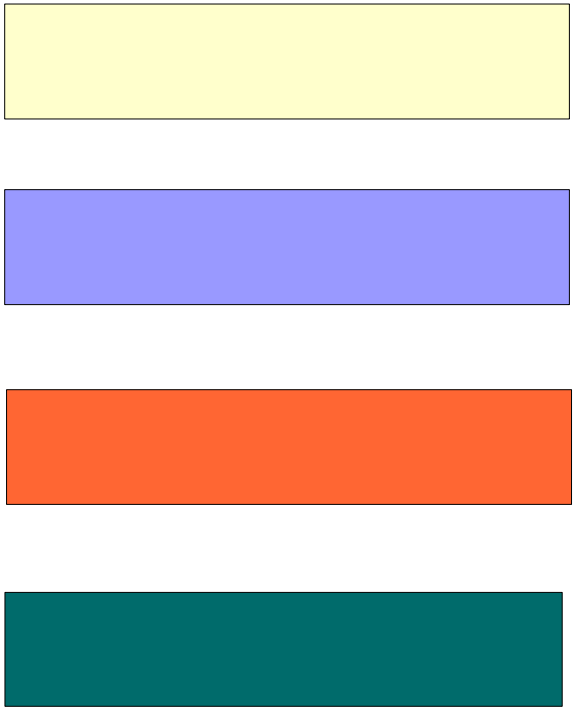
What is capacity?

- How do you measure capacity?
- How do you measure the horsepower of a car?
- What is the capacity of:
  - Ethernet
  - 802.11b
  - 3G
  - satellite-TV

What is bandwidth?

- Bandwidth is the size of resources allocated in a *frequency divided medium* measured in Hz.
- More bandwidth means potentially more capacity.
- What more defines the maximum capacity?

Communication layers



Application: ...

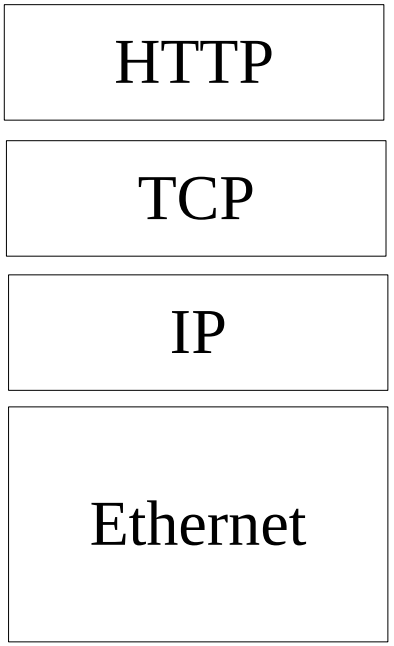
Transport: messages, streams, host-to-host, reliability, flow control,...

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

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

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

More layers



Hubs and switches

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

Packet vs circuit switching

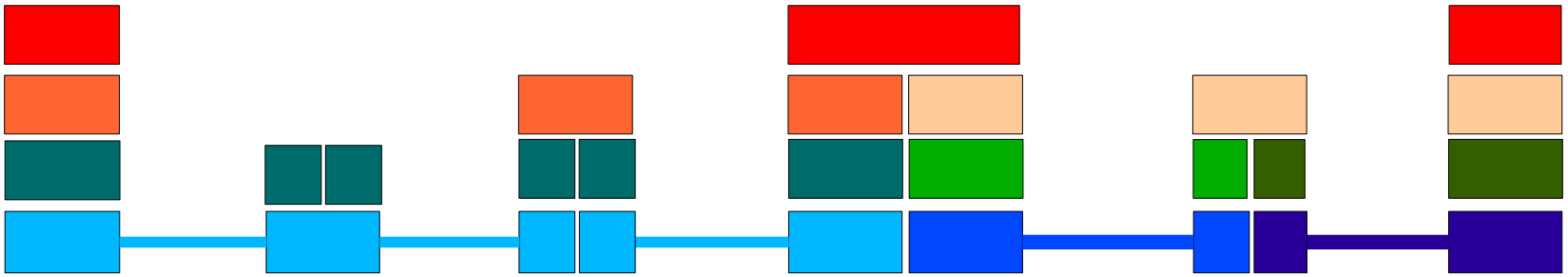
- What are the pros and cons of packet vs circuit switching?
  - Which scheme will take advantage of increased computing power?
  - Which scheme will take advantage of increased link capacity?

What would the world look like...

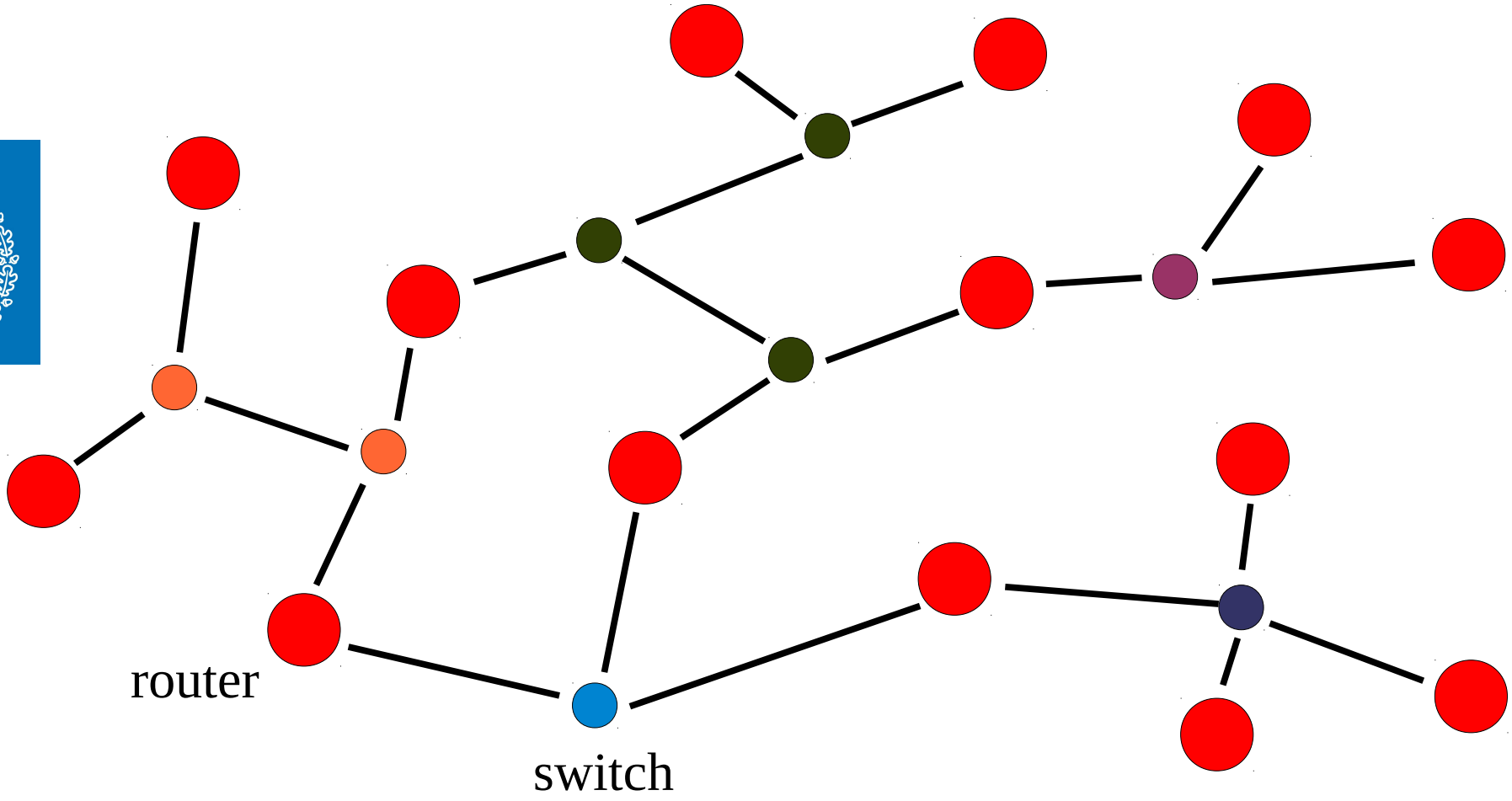
- .. if we only had Ethernet

Thank god for IP...

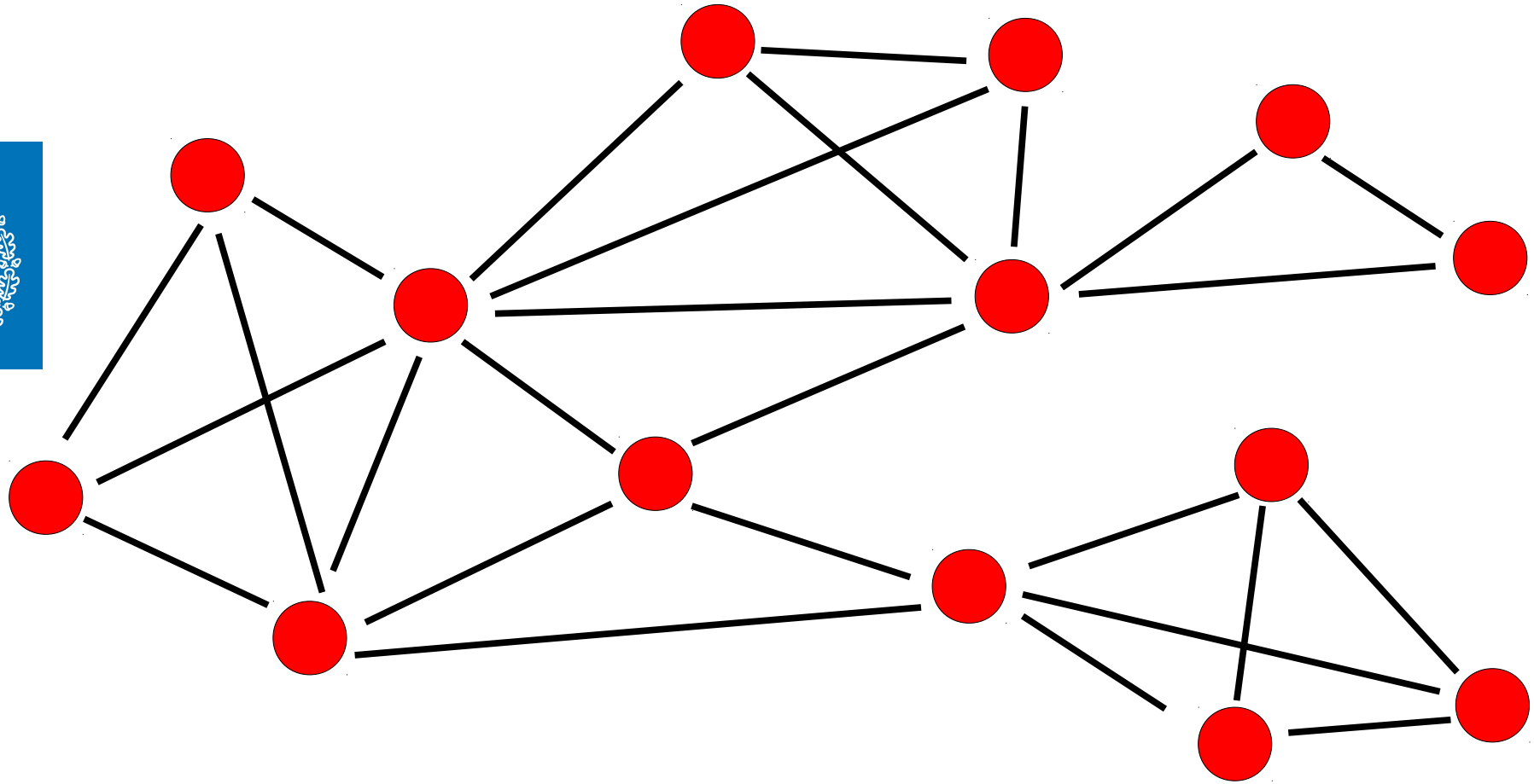
- .. but what does it give us?
- What is the job of a router and how is it different from a switch?



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, ...)
- More protocols defined on top of UDP and TCP.

UDP and TCP

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

UDP

- A datagram abstraction
  - 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
  - streams where retransmission can be allowed (non-realtime)
  - confirmed delivery?

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

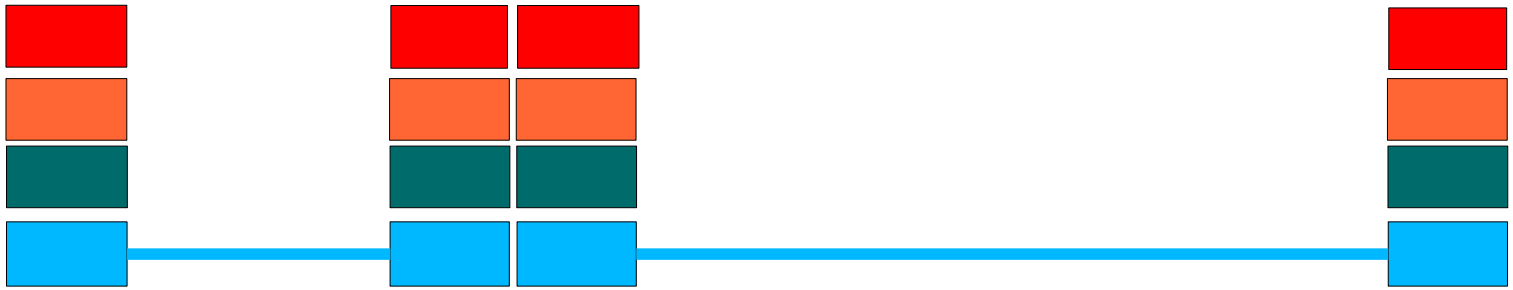
- Server:
  - Create a *listen socket* attached to a port
    - could be in several steps: create, bind, listen
  - Accept incoming request and create a *communication socket*
    - this is the socket used for reading/writing
- Client
  - Connect to a server given a specified port.
    - this is the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

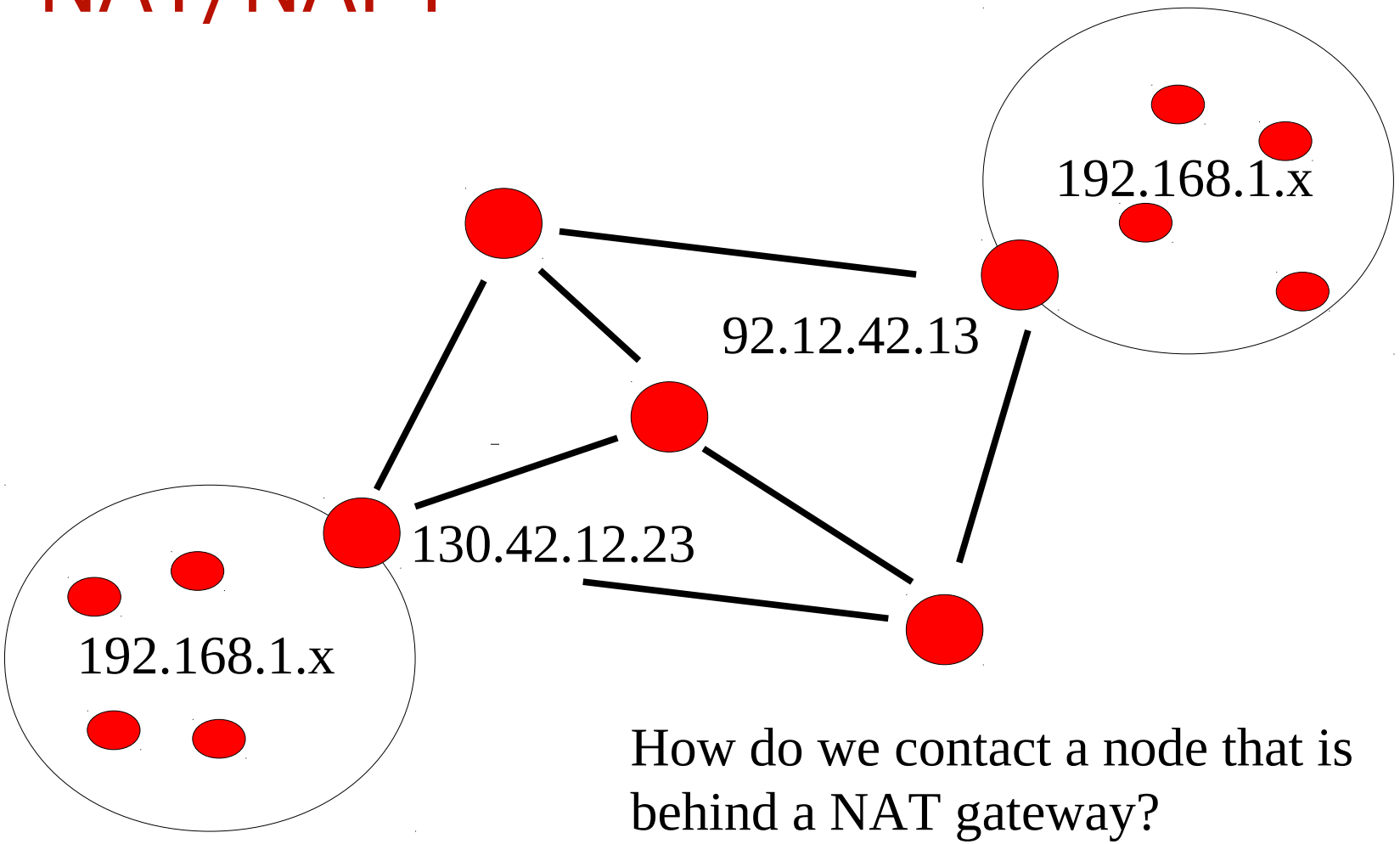
Gateway translates all local IP addresses to the IP address of the gateway.



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?

Domain Name System

delegation of authority

hierarchical
naming scheme

www.kth.se

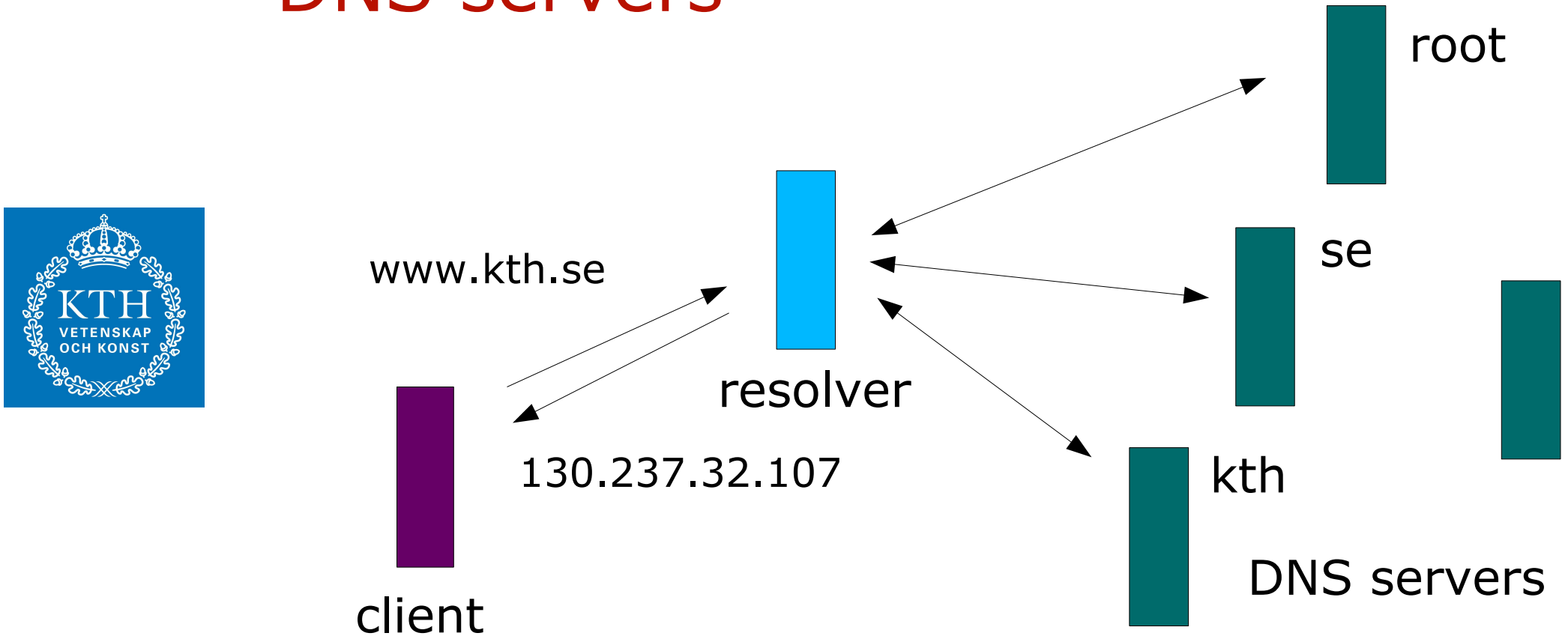
A record: IP address

reverse lookup

MX record: mail server

NS record: DNS server

DNS servers


root
se
kth
DNS servers
www.kth.se
resolver
130.237.32.107
client

Summary

Never under-estimate a girl with a truck.