

# Internetwork og Web-programmering

## Introduktion til netværk

Forelæsning 9  
Brian Nielsen

Distributed, Embedded, Intelligent Systems



# Agenda

1. Struktur af Internettet
2. Packet switching princip
3. Forsinkelse, Gennemløbsrate (Throughput), og Flaskehalse
4. Internet Protokol stakken

# Struktur af Internettet

Hvilke komponenter består nettet af?

Hvordan er det opbygget i netværk af netværk?

# Hvad består Internettet af?



- Milliarder af forbundne “computere” :
  - **hosts = end systems**
  - Afvikler netværks-applikationer
- **Kommunikations-forbindelser (links)**
  - fiber, kobber, radio, satellite,...
  - Transporterer data med en vis transmissions rate:
  - (Mega) bits-per-sekund (Mbps)
- **packet switches:** enhed, der videresender data-pakker i nettet
  - **routers** og Lag 2 -**switches**
- **Netværk:** samling af hosts, routere, links, som administreres af en organisation

# Hvad består Internettet af? "Things"

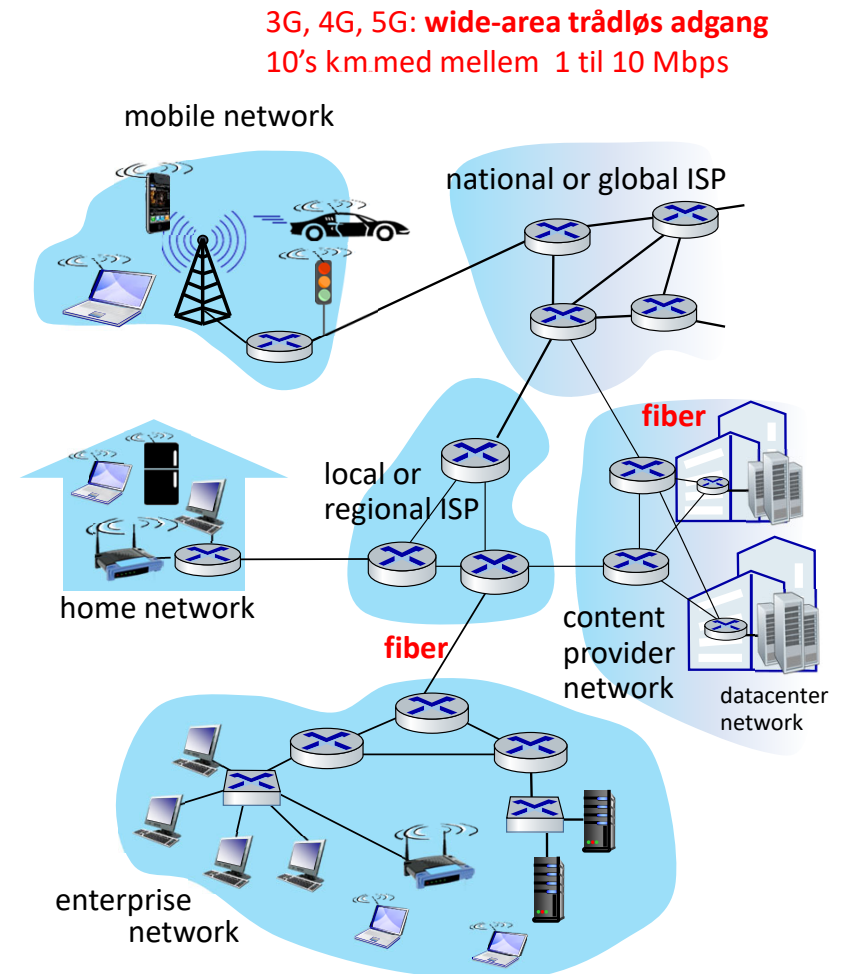


- Nu også end-systemer som sensorer, robotter, maskiner, TV, radiator termostater,...
- Internet of things
- Cyber-physical systems

<http://cityprobe.ciss.dk/>

# Simple model for structure of Internet

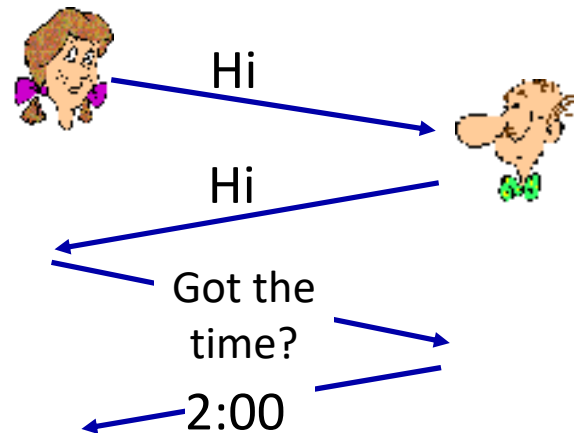
- **Netværks-udkant (edge):**
  - **hosts = end systems**
  - Klienter og servere
  - Servers, typisk placeret i datacentre
- **Adgangs netværk (access)**
  - Det yderste led, forbinder abonnenter til deres ISP
  - Lokal net + forbindelse til ISP
  - Trådede (wired),
  - Trådløse (wireless) forbindelser
- **Netværks kernen (core):**
  - Sammenkoblede ISP routere
  - Hierarkier
  - **Netværk af netværk**



ISP=Internet Service Provider

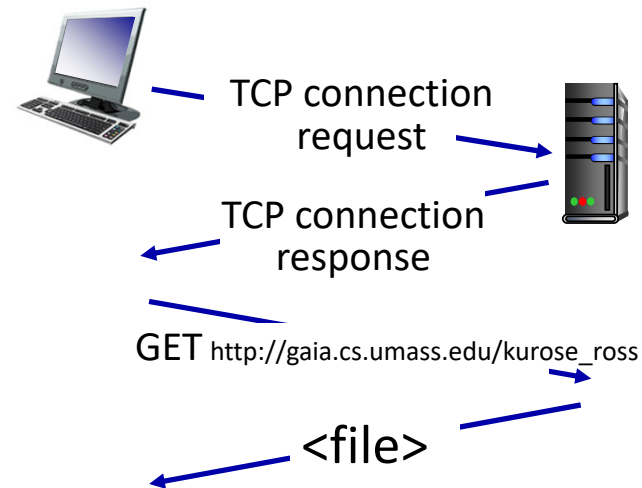
# Hvad består Internettet af?

## Mellem mennesker



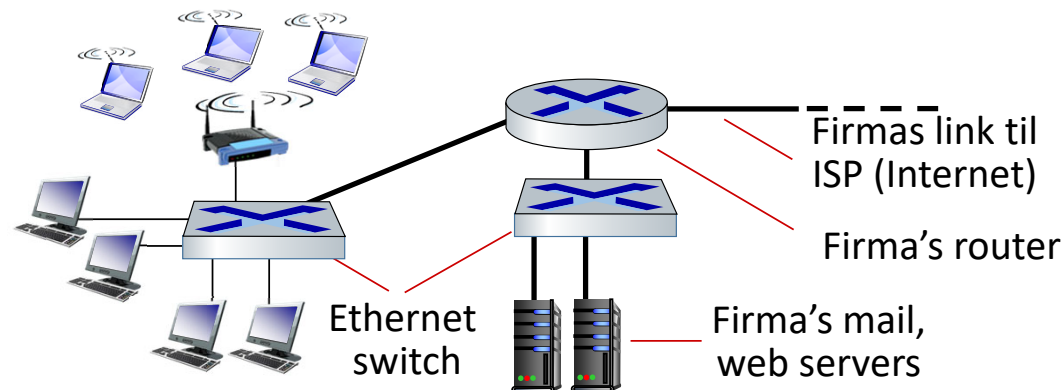
tid

## Mellem computer

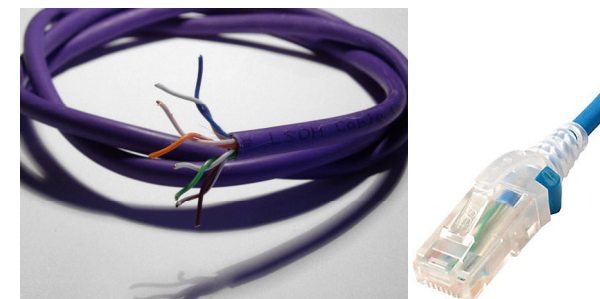


- **Protokoller:** Regler, der styrer afsendelse og modtagelsen af meddelelser
  - Dataindhold og format af meddelelser
  - Hvilke handlinger skal der ske ved modtagelse eller afsendelse af meddelelser?
  - e.g., TCP, IP, HTTP, Skype, 802.11
- **Internet standarder**
  - Beskrevet i "RFCs": Request for comments (<http://www.rfc-editor.org/standards>)
  - af IETF: Internet Engineering Task Force (<https://www.ietf.org/about/>)
- **Protokollerne er implementeret i en masse software!**

# Access Netværk i Firmaer (Ethernet)



- Flere lokale netværk typisk brugt i firmaer, universiteter, etc.
- Nu til dags er end-systemer typisk forbundet til en Ethernet (Lag 2) switch
  - Den mest brugte netværksteknologi i adgangs netværket
  - Faste forbindelse med 10 M bps, 100 M bps, 1G bps, 10G bps transmissions rater
- WiFi: Trådløs, access-points 11, 54, 450 Mbps

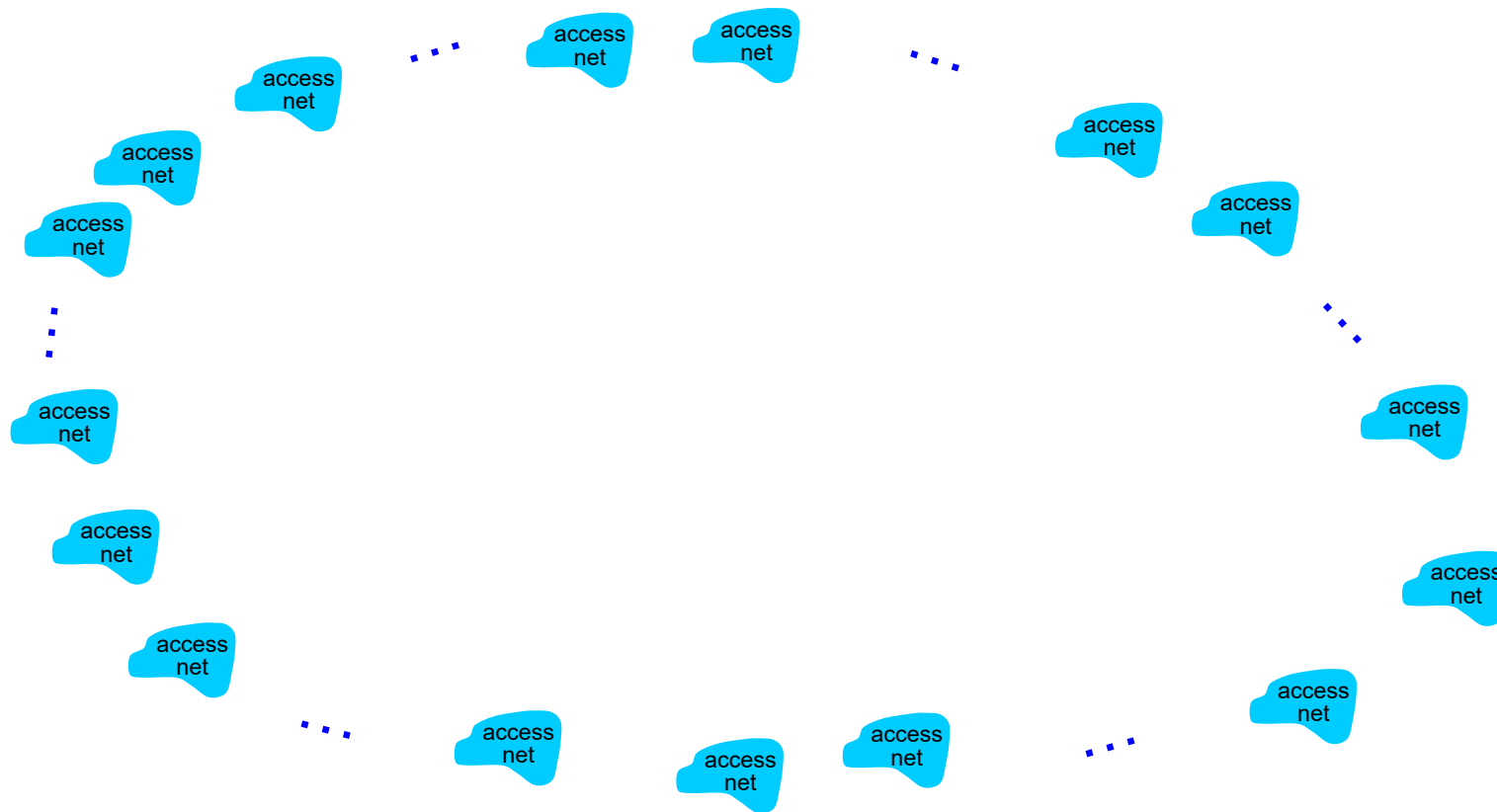


Ethernet fysisk medium:  
Isolerede par af snoede kobber ledninger



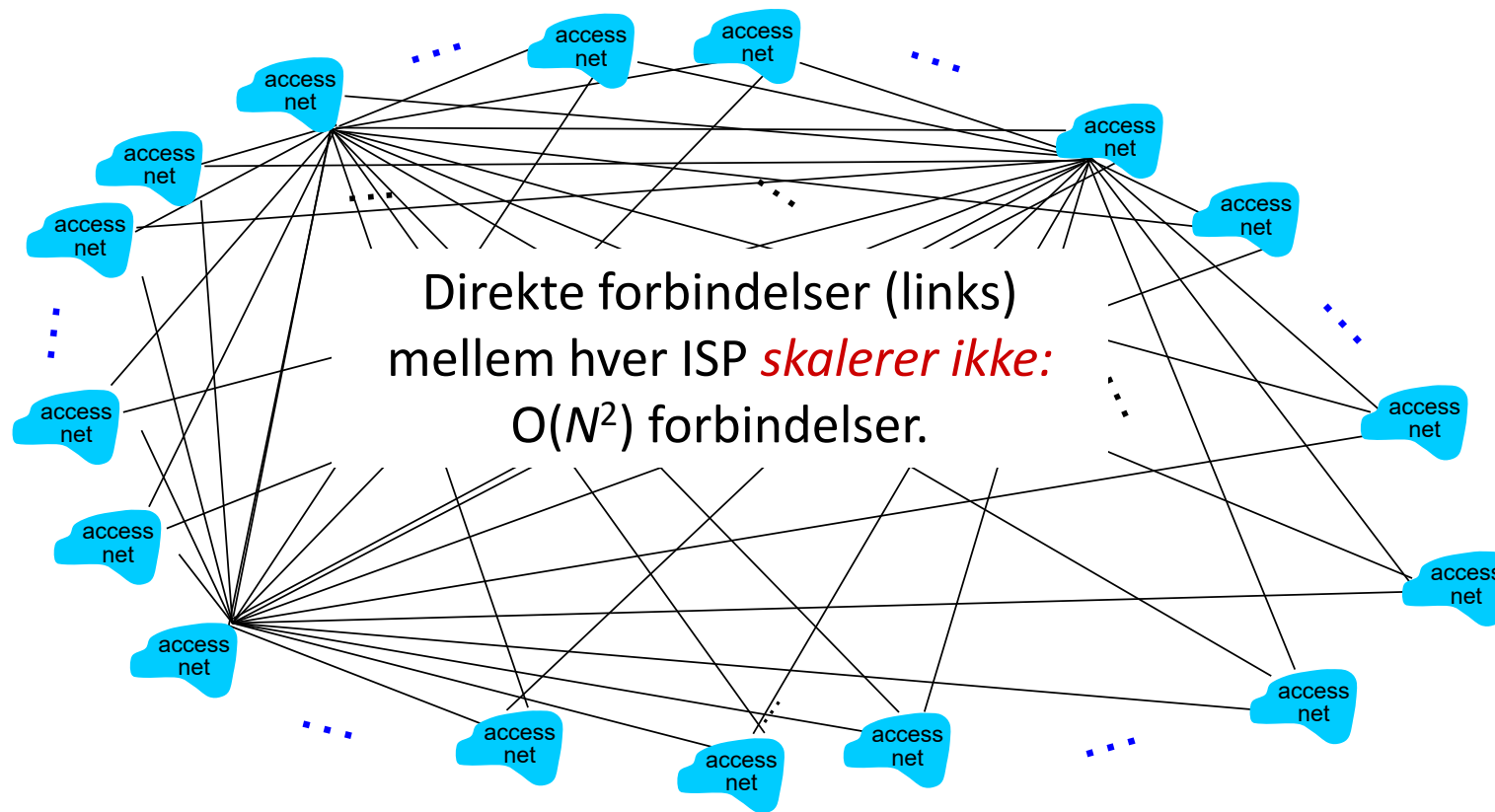
# Internet Struktur: Netværk af Netværk

*Spørgsmål:* Hvordan forbindes *millioner* af ISP adgangs-netværk?



# Internet Struktur: Netværk af Netværk

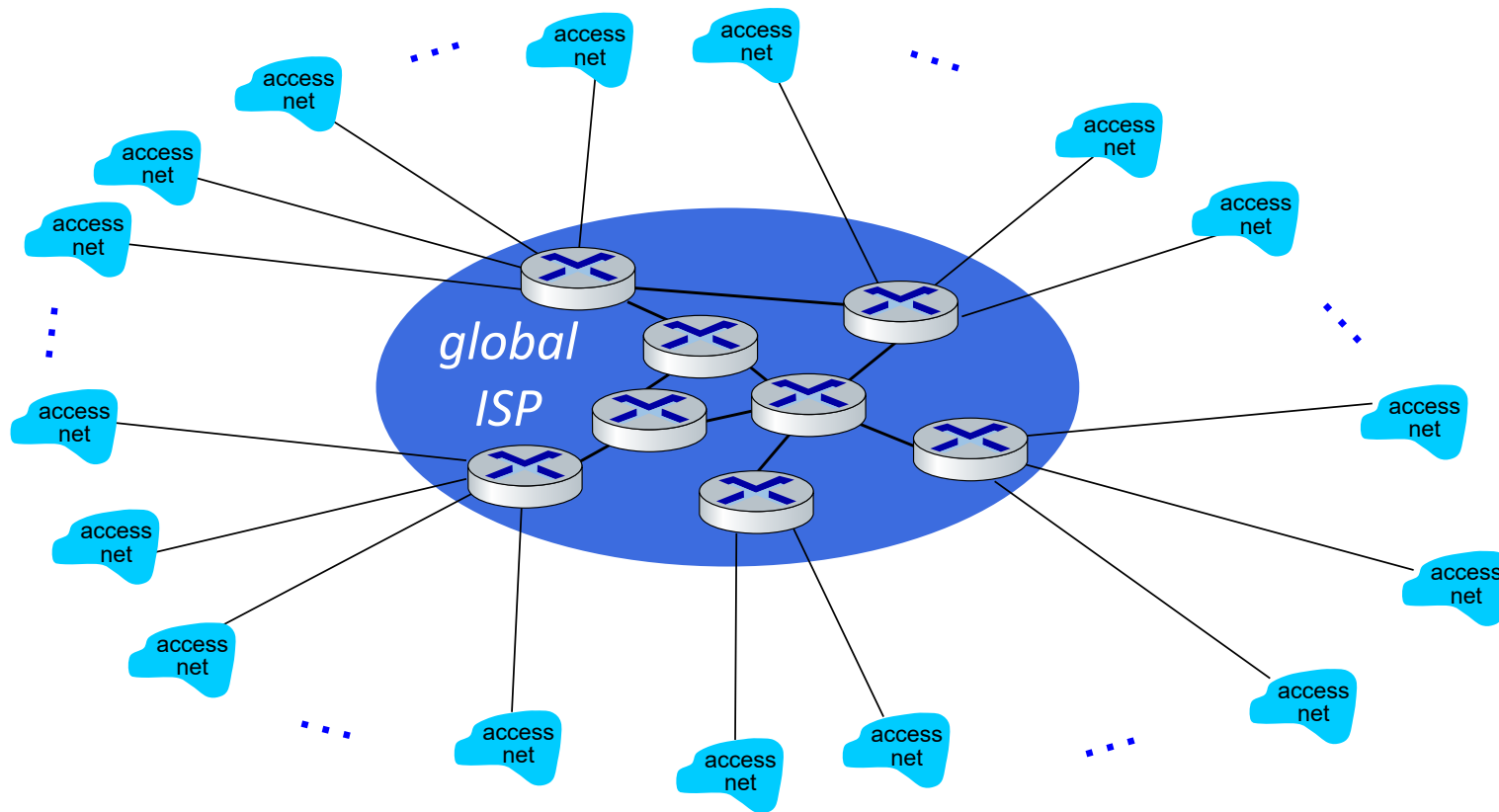
*Spørgsmål:* Hvordan forbindes *millioner* af ISP adgangs-netværk?



# Internet Struktur: Netværk af Netværk

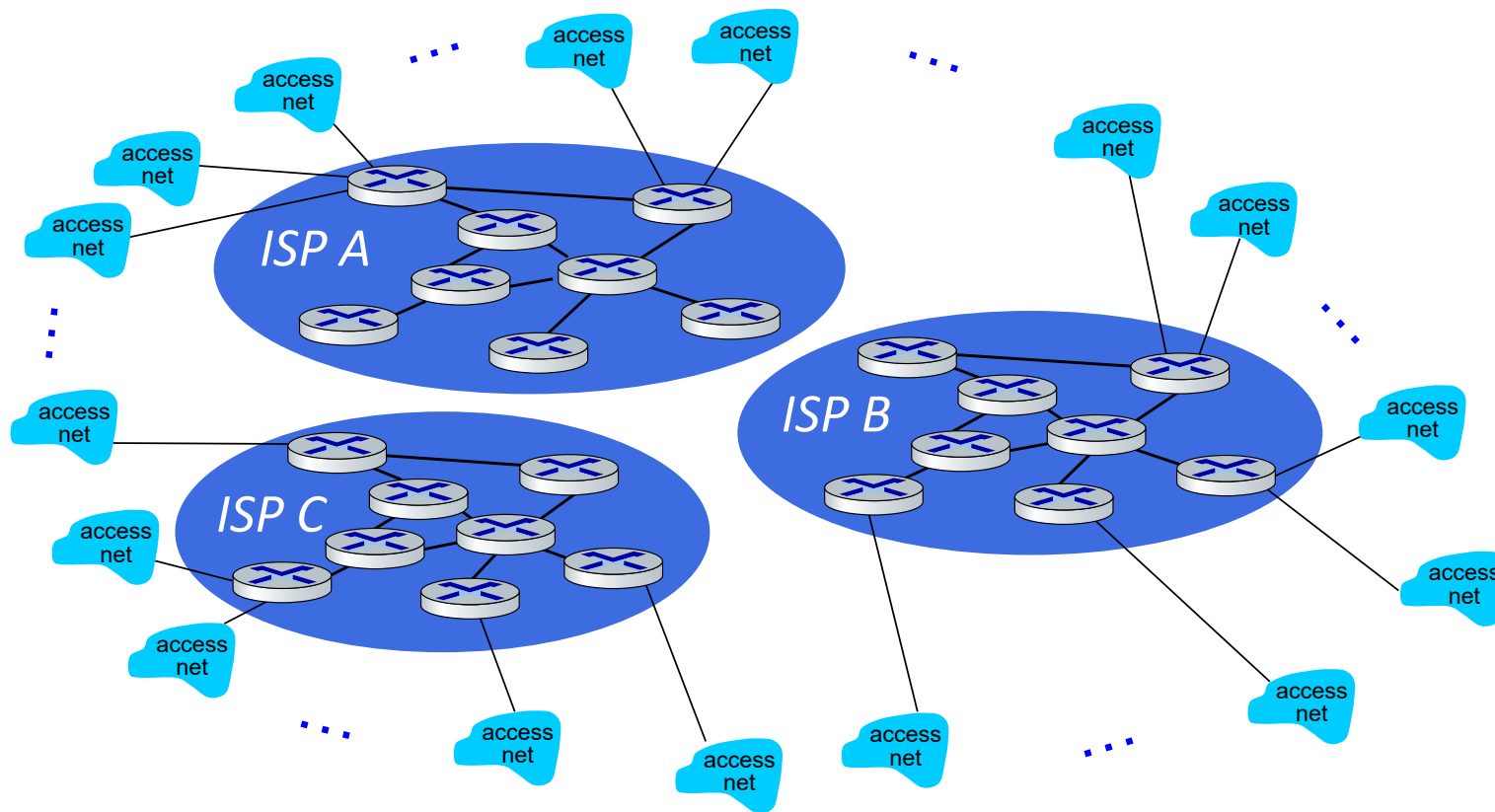
*En mulighed: forbind hver adgangs ISP til en global transit ISP?*

*Forretningsaftale imellem kunde- og leverandør ISPs*



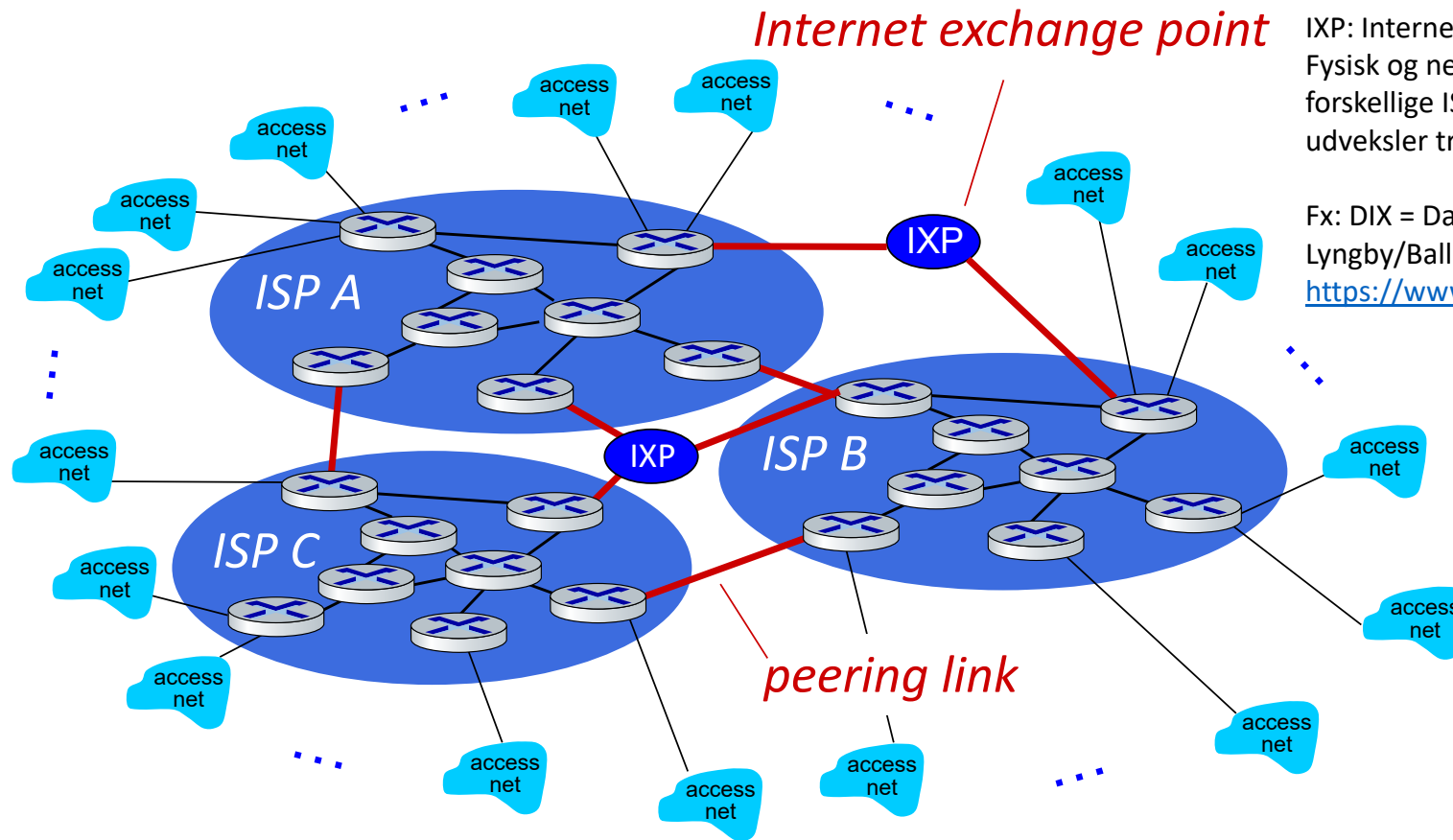
# Internet Struktur: Netværk af Netværk”

Bedre med flere transit ISP'ere: Konkurrence, decentralisering, skalering, ...



# Internet Struktur: Netværk af Netværk"

Nogle Transit ISPere ønsker at forbinde med hinanden

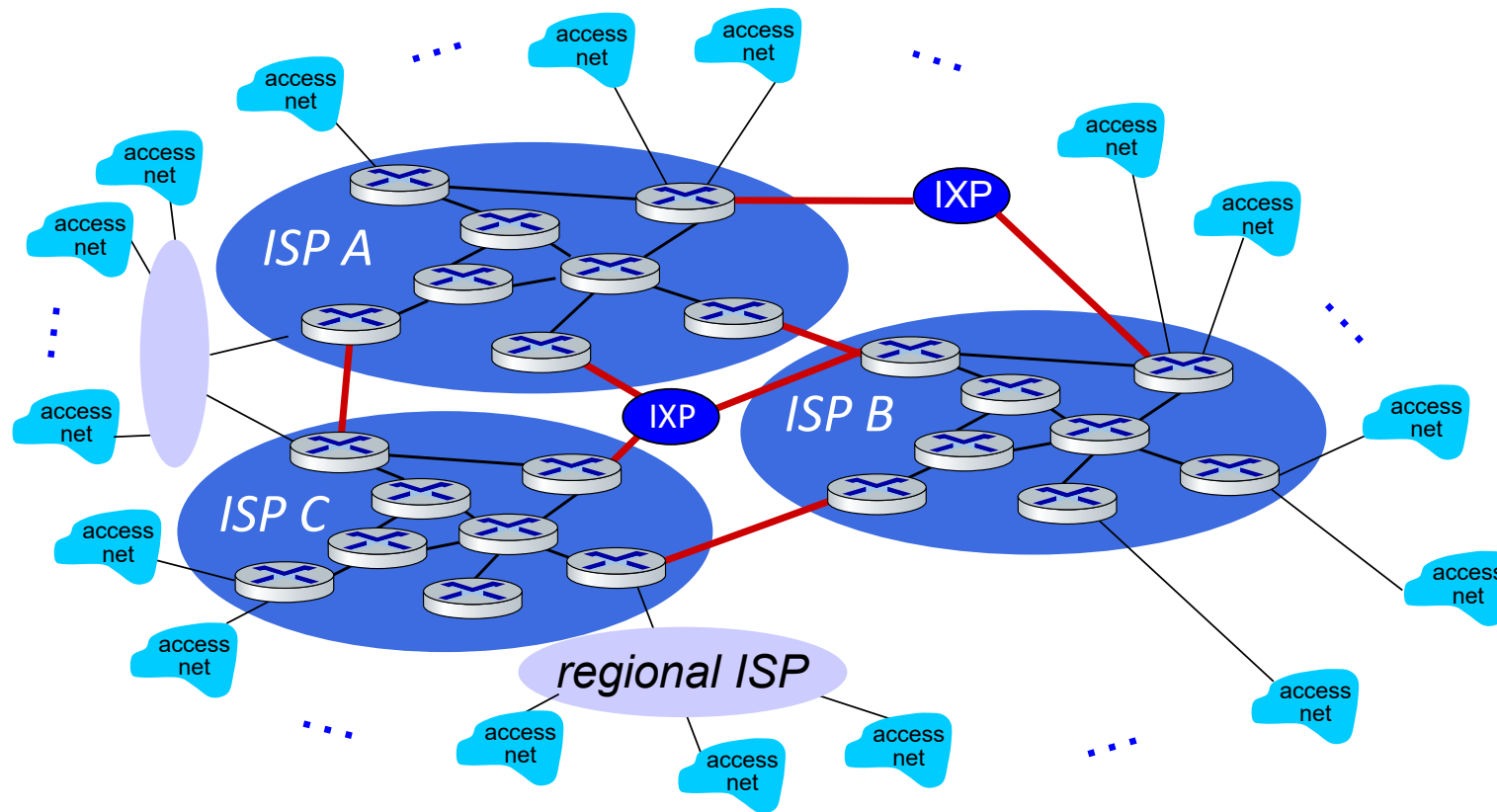


IXP: Internet Exchange Point (IXP) |  
Fysisk og neutral lokation, hvor  
forskellige ISP netværk mødes og  
udveksler trafik.

Fx: DIX = Danish Internet Exchange Point  
Lyngby/Ballerup/Skanderborg  
<https://www.dix.dk/>

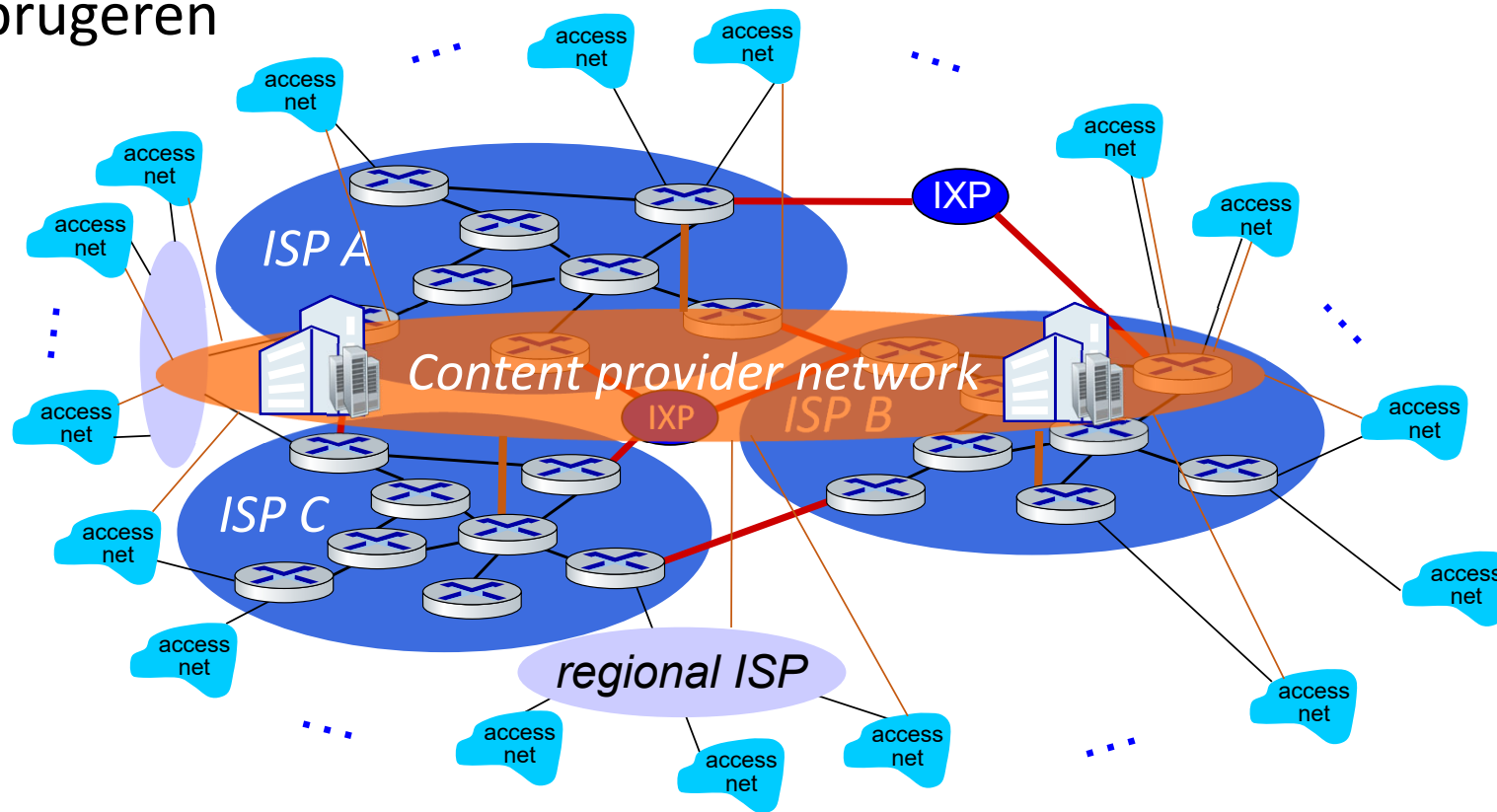
# Internet Struktur: Netværk af Netværk

... der kan opstå regionale netværk, der ønskes forbundet til det globale netværk

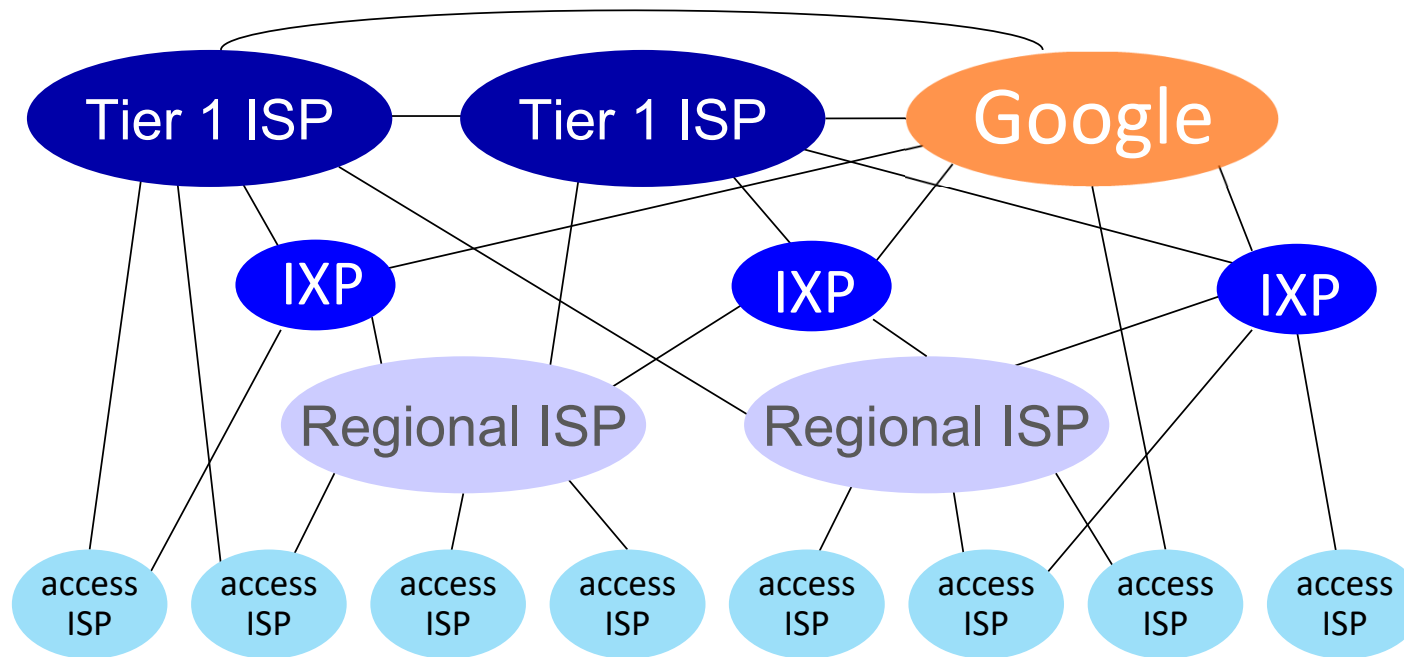


# Internet Struktur: Netværk af Netværk

... og større indholds-leverandører (e.g., Google, Microsoft, Akamai) driver deres egne netværk for at levere services og indhold tættere på slut-brugeren



# Internet Struktur: Netværk af Netværk



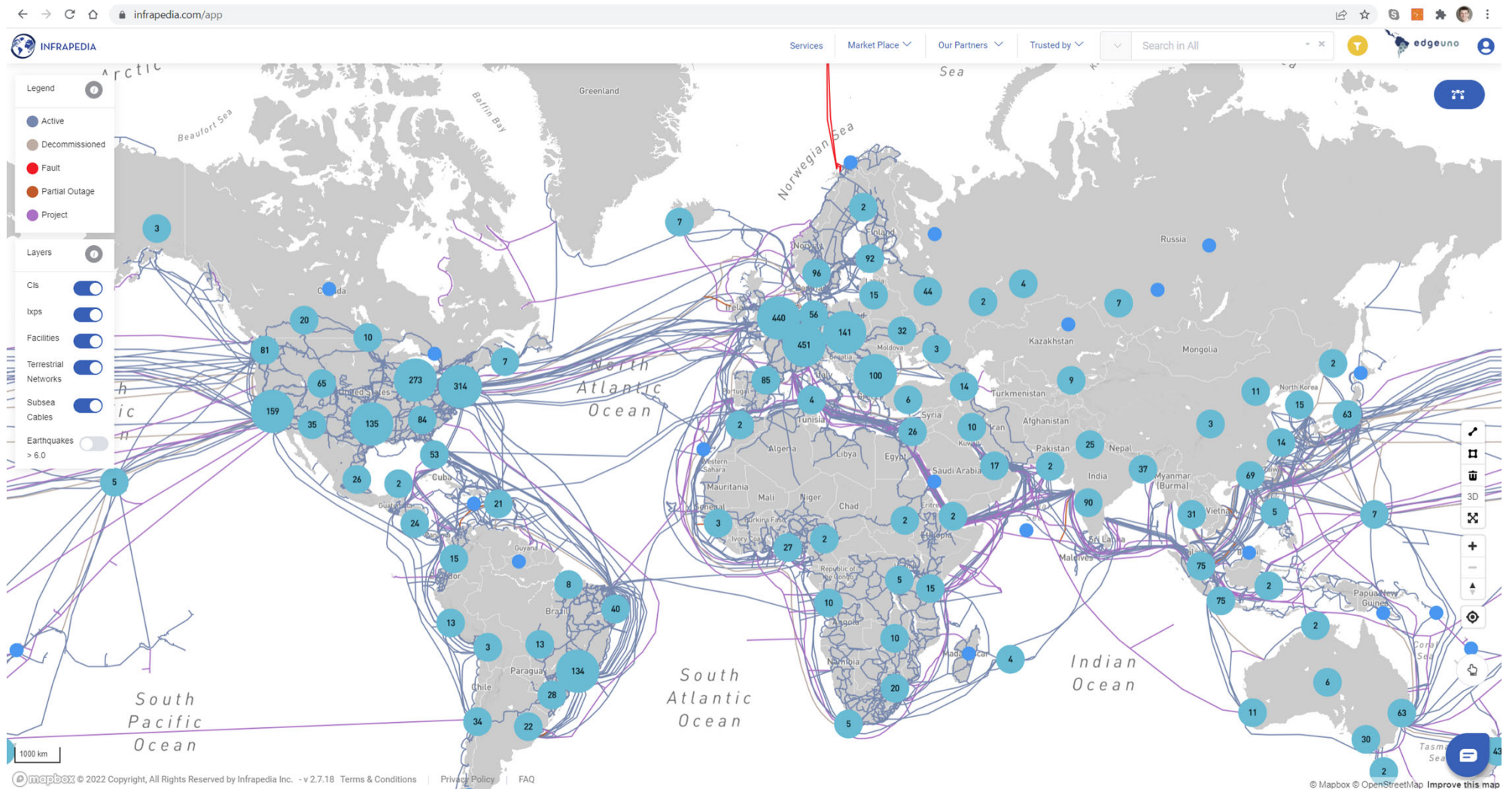
En kompleks struktur af *inter-forbundne netværk* blandt

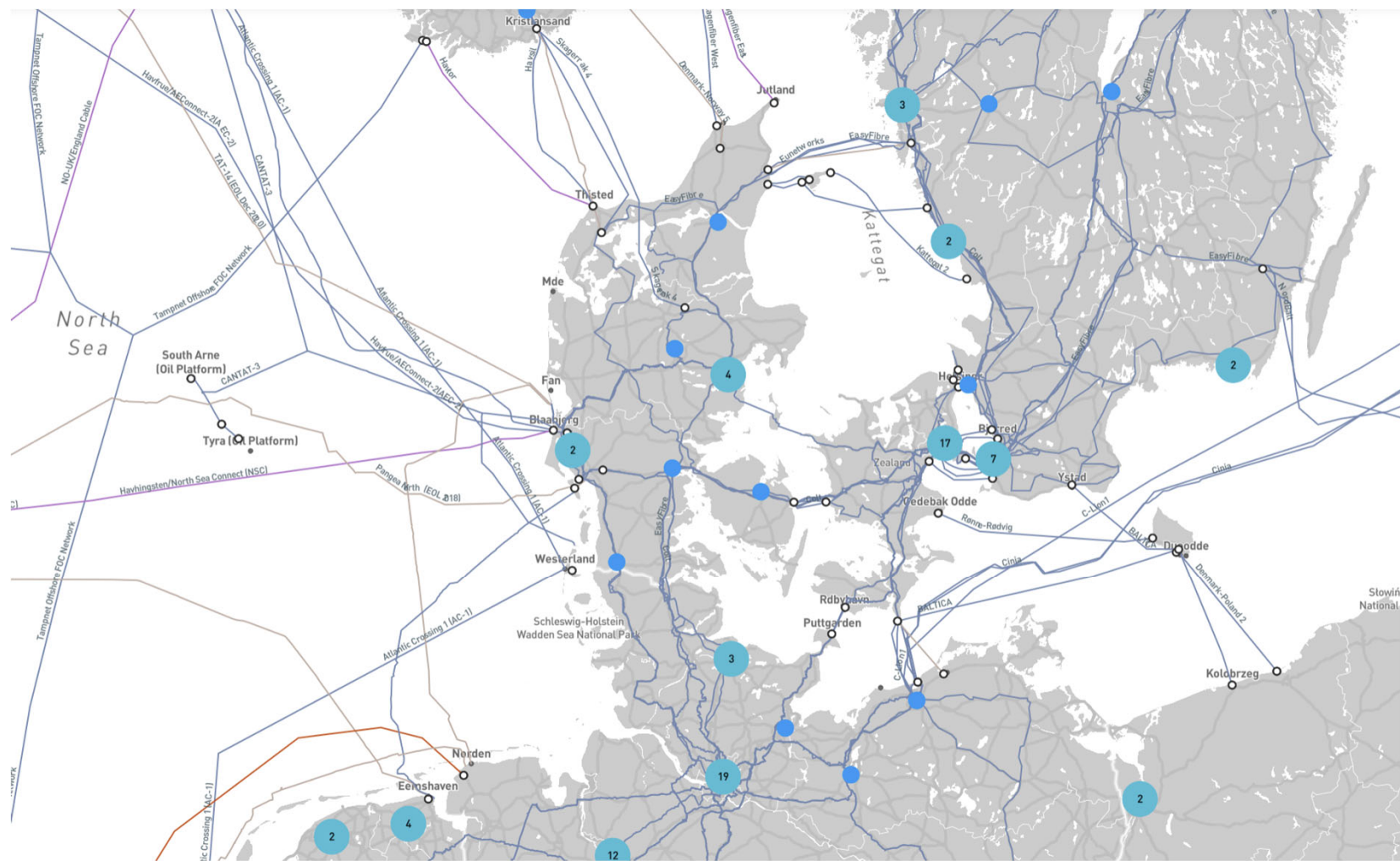
- ISPs, Telekom udbydere, Indholdsudbyderes netværk (fx., Google, Microsoft, Akamai)
- I kernen: en mindre antal velforbundne store netværk, drevet af Større tele- og data-kommunikationsselskaber
- Data centre koncentrerer mange servers ("the cloud"), og indholdsleverandører koncentrerer meget trafik, ofte i egne netværk udenom lag-1 or regionale ISPs



# Infrapedia

Crowd-sourced information om netværks-forbindelser og infrastruktur (ufuldstændigt)





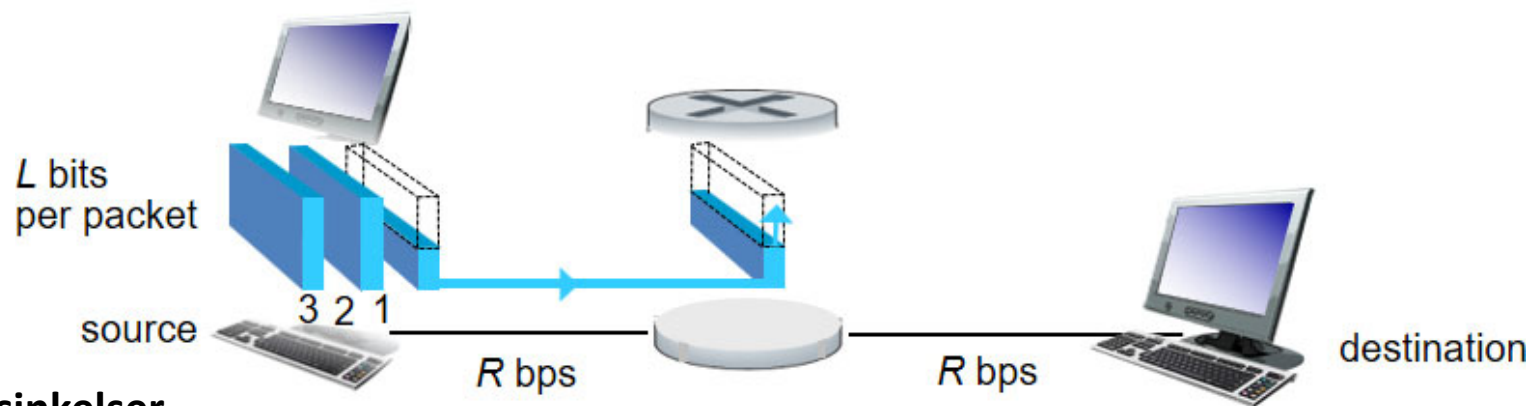
# Packet-switching

Hvad er packet switching?

Hvorfor og hvornår er det smart?

# Packet Switching

- **Pakke-kobling (packet-switching):** sender host deler sine data op i mindre “data-pakker” \*)
  - Videre-send pakker fra en router til den næste på stien af links fra source til destination
  - Hver pakke sendes med linkets (fulde) transmissionsrate
- **Gem-og-videre-send (store and forward):** hele pakken skal modtages af en router og gemmes i dennes hukommelse før den kan videre-sendes på den næste link

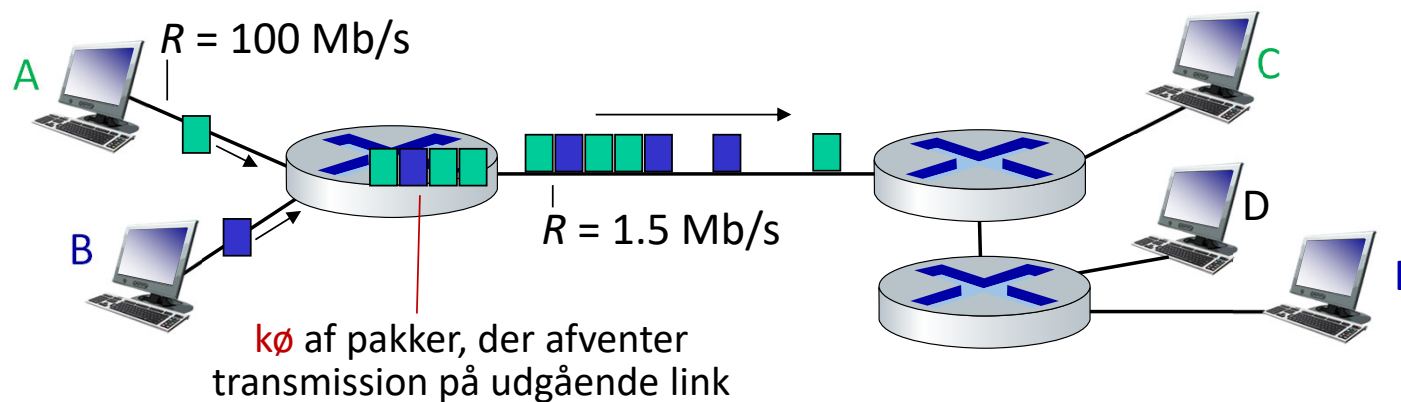


- **Forsinkelser**
  - En pakke der er  $L$ -bits lang, sendt med raten  $R$  bps, tager  $L/R$  sekunder at sende
  - Ex. Sende-forsinkelse:  $1000 \text{ bits} / 1 \text{ Mbps} = 1 \text{ ms}$
  - Ex. 2 hop til destination: Samlet forsinkelse:  $2L/R$  (plus noget mere...)

\*) Fx, På Ethernet er maksimal pakkestørrelse  $L = 1522 \text{ bytes} = 12176 \text{ bits}$

# Packet Switching: Pakke-køer og pakke-tab

- Hvis ankomst raten til et link overstiger dets transmission rate over en kort periode:
  - Pakker bliver sat i kø, og afventer transmission på link



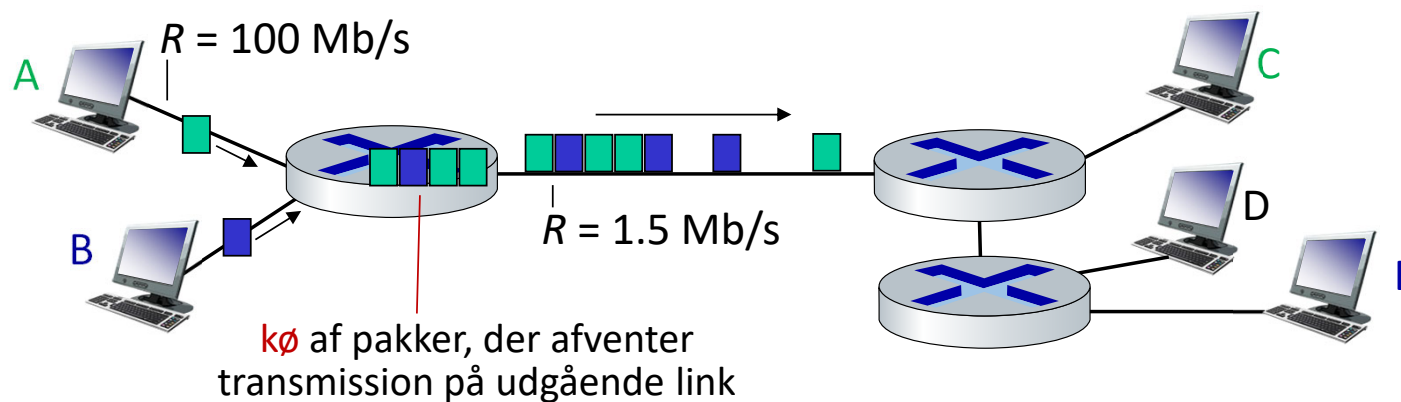
Coronatest kø?!

Der opstår køer når arbejde ankommer hurtigere end det kan behandles



# Packet Switching: Pakke-køer og pakke-tab

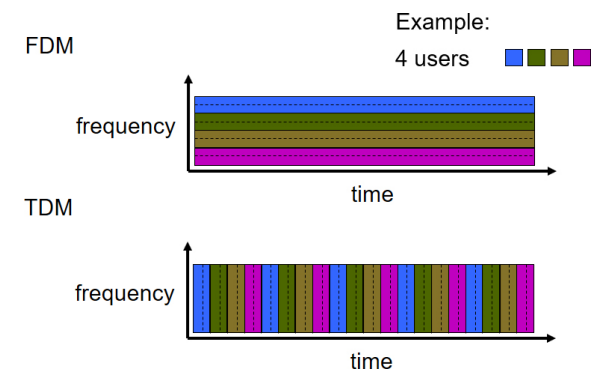
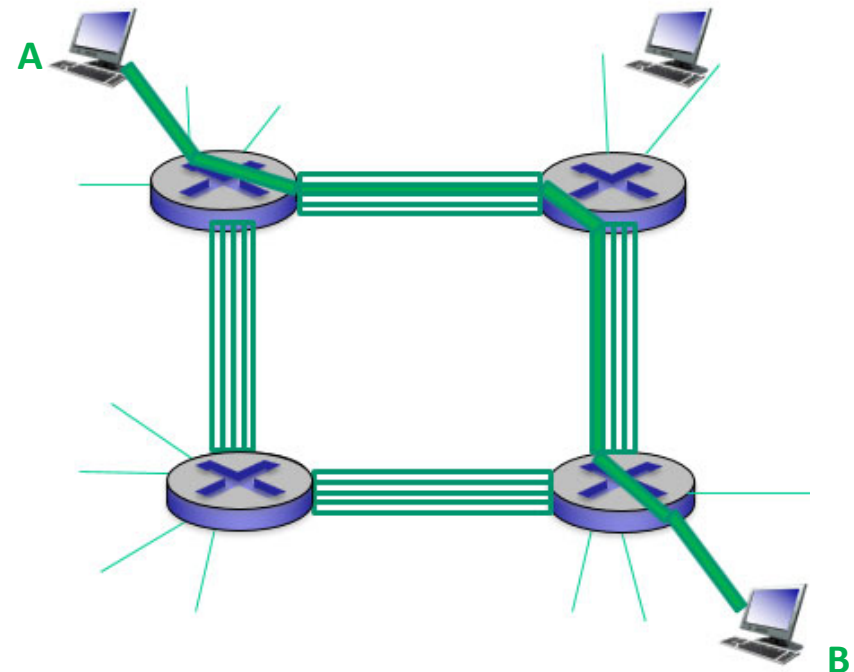
- Hvis ankomst raten til et link overstiger dets transmission rate over en kort periode:
  - Pakker bliver sat i kø, og afventer transmission på link
  - Pakker smides ud (droppes/tapes) hvis bufferen løber tør for plads



- **Godt til data-trafik som kommer i salver (bursts):** resource deling ("statistical multi-plexing")
  - Hvis A og B sender samtidigt må de deles om output linkets kapacitet (fx 1.5 Mbps)
  - Det er mindre sandsynligt at A og B begge sender med  $R$  på samme tid langvarigt
    - Hvis A sender "nu" og B ikke gør så bliver A's trafik videresendt med fuld 1.5 Mbps hastighed
- Uden kontrol: uhæmmet grad af "forstoppelse" (congestion)
- Alternativt princip : circuit switching

# Kredsløbskobling (circuit switching)

- Der oprettes en forbindelse mellem sender og modtager
- Hver forbindelse får forud reserveret en fast transmissionsrate, fx 100 kbps
  - Tilrådighed hele tiden uanset større eller mindre behov.
- Teknikker til opdeling af links kapacitet
  - Frequency-Division Multiplexing
  - Time-Division Multiplexing



# Pakke kobling vs. kredsløbskobling

## eksempel:

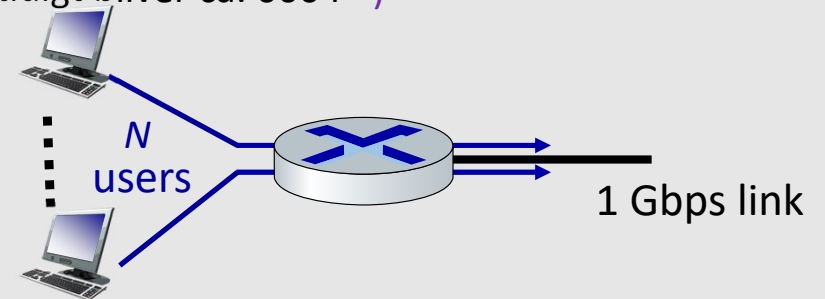
1 Gbps link, hvor hver bruger anvender 100 Mbps, men er kun “aktiv” 10% af tiden

### circuit-switching

- tillader ialt: 10 brugere
- Dårlige resource udnyttelse
- Godt til kritisk data
- Godt til tids-følsomt data

### packet switching:

- $P(\text{en given bruger sender})=0.1$
- Ud af fx 35 (uafhængige) brugere, bliver sandsynligheden for at mere end 10 er aktivt samtidigt bliver ca. 0004 \*)



- God ressource udnyttelse ved “bursty” trafik (kommer i “stød/salver”).
- Simple, ingen opsætning af kredsløbet
- Forstoppelse (congestion) er muligt, giver pakketab
  - => behov for mere avancerede protokoller til pålidelig data-overførsel og forhindring af forstoppele

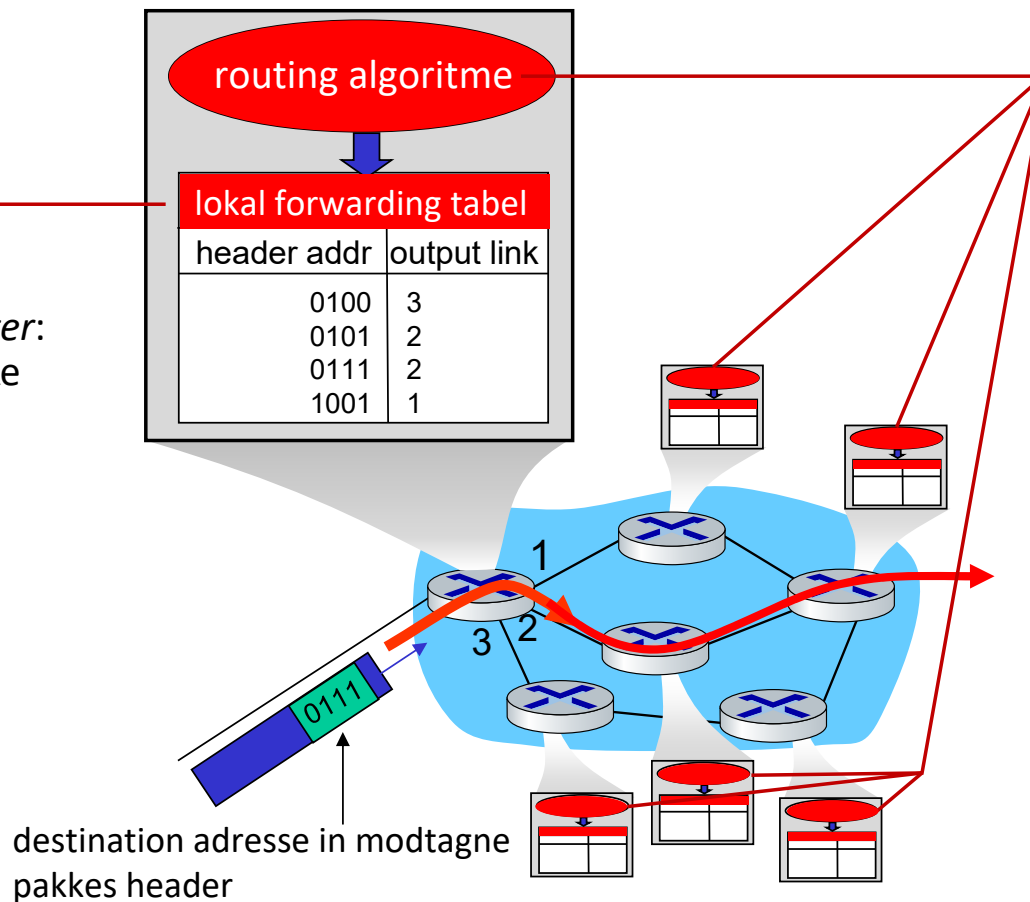
\*) Statistik med binomial fordelingen! SLIAL?



# To nøgle funktioner i en router

## Forwarding:

- aka “switching”
- *lokal* handling i en router: flyt ankommende pakke fra input link til rette output link



## Routing:

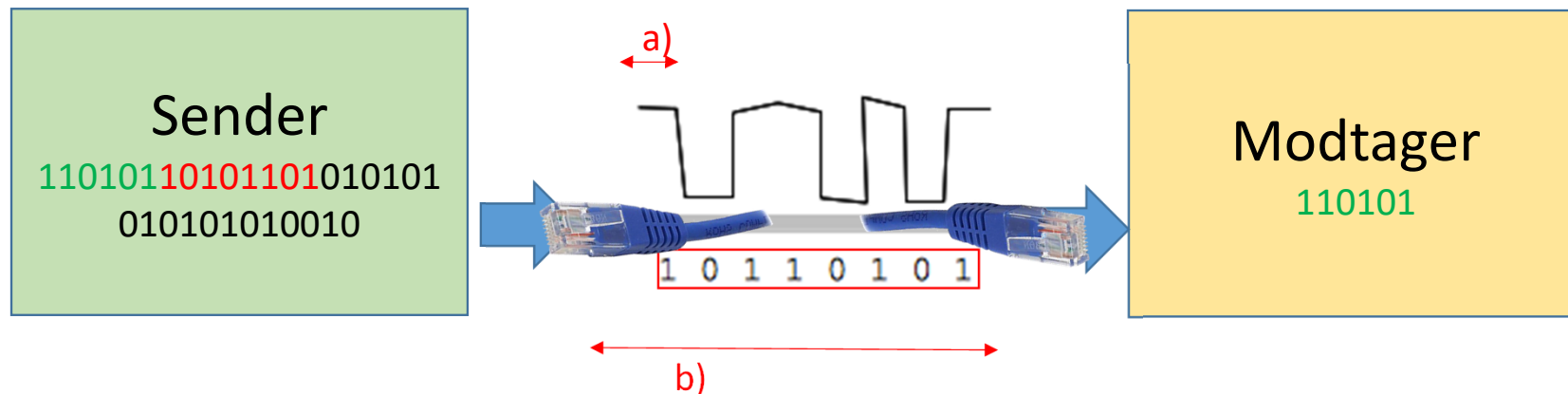
- *global* handling: beregn source-destination sti, som pakker skal følge
- routing algoritmer

# Forsinkelse, Throughput, Tab

Hvordan bestemmer man forsinkelse gennem netværket?

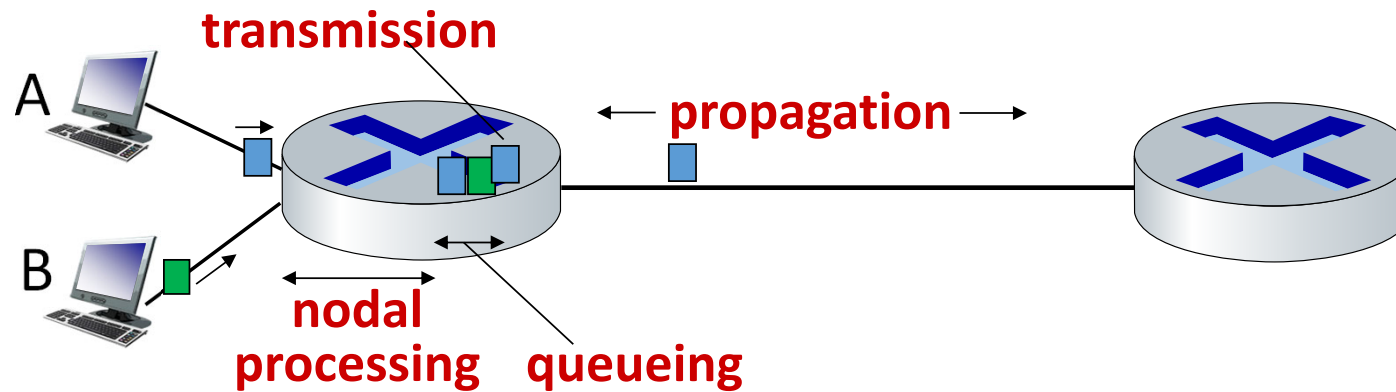
Hvordan bestemmes den mulige overførselshastighed?

# Udbredelse af et digital signal på et medie



- Det tager noget tid at sende (transmittere) en bit på mediet **(a)**
  - (for-)simplet eksempel
    - Et logisk 1 kodes som 3 volt i 1  $\mu$ s
    - Et logisk 0 kodes om 0 volt i 1  $\mu$ s
    - Giver transmissions hastighed på 1 mega bits per sekund
- Der er en signal-udbredelsestid mellem sender og modtager **(b)**
  - Afhænger af mediet og afstanden mellem sender og modtager
  - Knap lysets hastighed (ca.  $2 \cdot 10^8$  m/s)
  - Elektrisk signal i kobber: ca. 70% af lysets hastighed

# Kilder til forsinkelse i en router

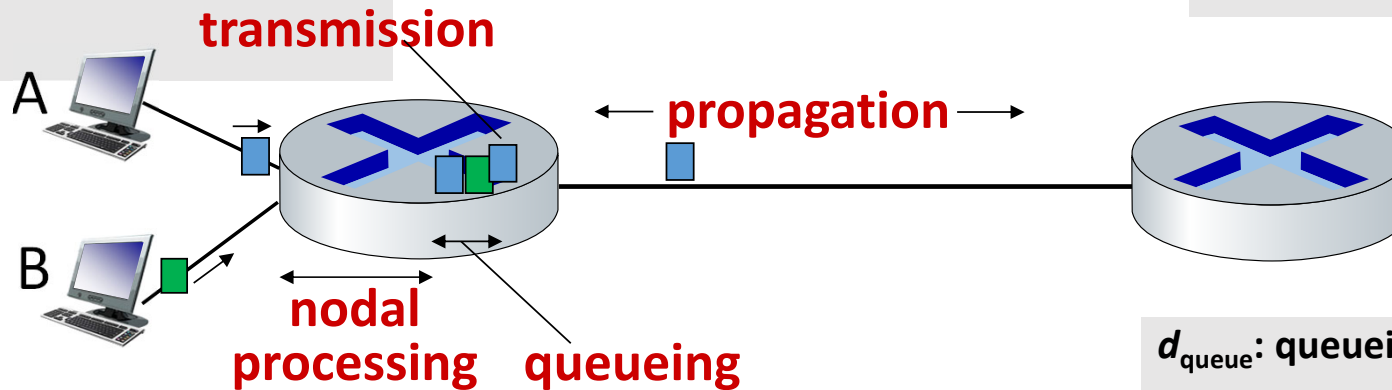


$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

# Kilder til forsinkelse i en router

## transmissions delay:

- Det tager tid at "kode en bit op" på mediet
- $L$ : længden af pakken (bits)
- $R$ : link **transmissions rate (bps)**
- $d_{trans} = \frac{L}{R}$



## Propagerings (udbredelse) delay:

- $d$ : distance (længde af fysiske link)
- $s$ : udbredelses hastighed (knap lysets hastighed (ca  $2 \cdot 10^8$  m/s))
- $d_{prop} = \frac{d}{s}$ , (kun betydende når  $d$  stor)

## $d_{proc}$ : nodal processering

- Knudepunktets data-behandlingstid
- check bit fejl
- bestemme output link
- Indsæt i kø
- typisk  $< \text{msec}$

## $d_{queue}$ : queueing delay

- Ventetid i kø, før pakken kan sendes på udgående link
- Afhænger af hvor antallet af ophobede pakker (congestion level)

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

# Trafik Intensitet

*Forholdet imellem indkommende og udgående trafik rate*

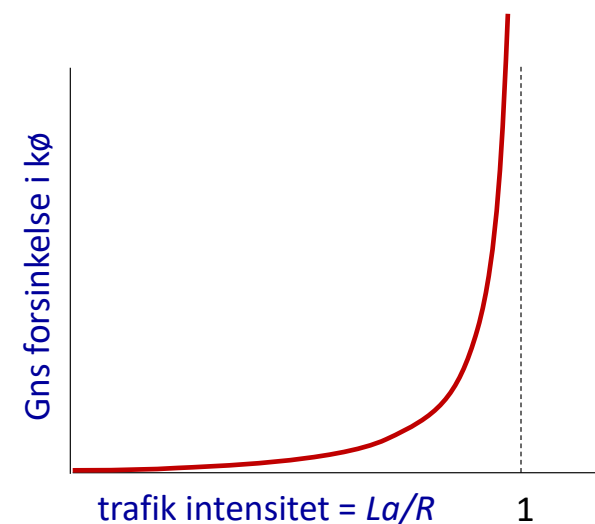
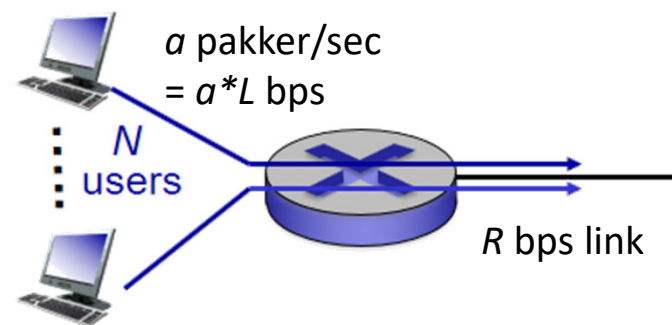
- $L$ : pakke længde (bits)
- $a$ : gennemsnitlig pakke ankomst rate (pakker er sec.)
- $L \times a$  = ankomst rate (bps)
- $R$ : link transmissions rate (bps)
- Trafik intensitet:  $\frac{L \times a}{R}$



- Hvis  $\frac{La}{R}$  overstiger 1: fortsat akkumulering af ventende pakker: Konsekvens=Pakketab!
- Hvis  $\frac{La}{R}$  er lille  $\sim 0$ : klar bane
- Hvis  $\frac{La}{R}$  nærmer sig 1: lang ventetid

(eksakt forsinkelse afhænger af klumpers fordeling i trafikken)

- Systemet skal designes så intensiteten bliver moderat



# Rigtige forsinkelser og router på Internettet

**traceroute:** gaia.cs.umass.edu to [www.eurecom.fr](http://www.eurecom.fr)

Udsender kontrol pakker for at måle en pakkes rute og forsinkelse igennem nettet:

```
1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms
6 abilene-vbns.abilene.ucaid.edu (198.32.11.9) 22 ms 18 ms 22 ms
7 nycm-wash.abilene.ucaid.edu (198.32.8.46) 22 ms 22 ms 22 ms
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms
9 de2-1.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms
10 de.fr1.fr.geant.net (62.40.96.50) 113 ms 121 ms 114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms
12 nio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms
13 nice.cssi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms
14 r3t2-nice.cssi.renater.fr (195.220.98.110) 126 ms 126 ms 124 ms
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54) 135 ms 128 ms 133 ms
16 194.214.211.25 (194.214.211.25) 126 ms 128 ms 126 ms
17 * * *
18 * * *
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms
```

3 målinger af forsinkelse fra  
gaia.cs.umass.edu til cs-gw.umass.edu

3 målinger af forsinkelse til  
border1-rt-fa5-1-0.gw.umass.edu

trans-atlantisk link

looks like delays  
decrease! Why?

\* Angiver "intet svar" (probe-pakke tabt, router svarer ikke)

\* Forsøg at lave traceroute til nogle eksotiske lande på [www.traceroute.org](http://www.traceroute.org)

# Demo ("tracert" program)

```
Kommandoprompt

-w timeout      Wait timeout milliseconds for each reply.
-R             Trace round-trip path (IPv6-only).
-S srcaddr      Source address to use (IPv6-only).
-4             Force using IPv4.
-6             Force using IPv6.

C:\Users\bniel>tracert www.cs.aau.dk

Tracing route to www.cs.aau.dk [130.225.63.3]
over a maximum of 30 hops:

  1    2 ms    2 ms    2 ms  192.168.0.1
  2    3 ms    4 ms    3 ms  85.203.152.129
  3    4 ms    3 ms    3 ms  stovr01ds01-ae24-0.eniig-net.dk [85.191.209.76]
  4    4 ms    4 ms    5 ms  aarsx01cr01_ae11.em-net.dk [85.191.209.38]
  5    5 ms    5 ms    5 ms  10.10.1.53
  6    7 ms    6 ms    5 ms  10.10.0.1
  7    6 ms    6 ms    6 ms  87.116.38.121
  8   21 ms    8 ms    8 ms  93.176.93.8
  9   10 ms   11 ms   21 ms  dk-uni.nordu.net [192.38.7.50]
 10   12 ms   10 ms   10 ms  lgb.core.fsknet.dk [109.105.102.159]
 11   10 ms   11 ms   10 ms  100g-lgb.ore.core.fsknet.dk [130.225.245.154]
 12   18 ms   18 ms   27 ms  edge1.aau.dk [130.226.249.146]
 13   17 ms   17 ms   17 ms  Eth1-20.aau-core1.aau.dk [192.38.59.27]
 14   18 ms   18 ms   17 ms  Eth5-15.dc2-gw02.aau.dk [192.38.59.203]
 15   28 ms   27 ms   18 ms  vm-ig-www2.portal.aau.dk [130.225.63.3]

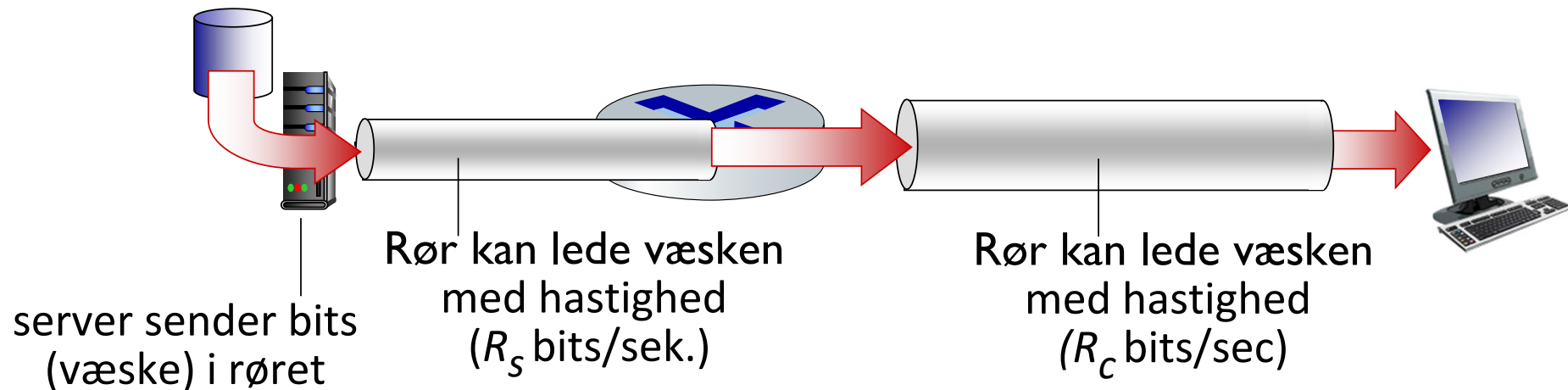
Trace complete.

C:\Users\bniel>
```



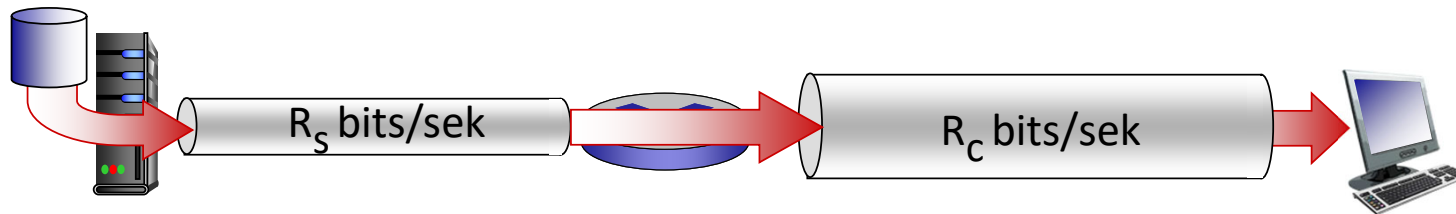
# Throughput

- **throughput:** (gennemløbsrate) hastighed (i bits/sekund) som sendes fra sender til modtager
  - **Øjeblikkelig:** hastighed på et givet tidspunkt
  - **Gennemsnitlig:** hastighed over en længere tidsperiode
- Rørlednings analogi

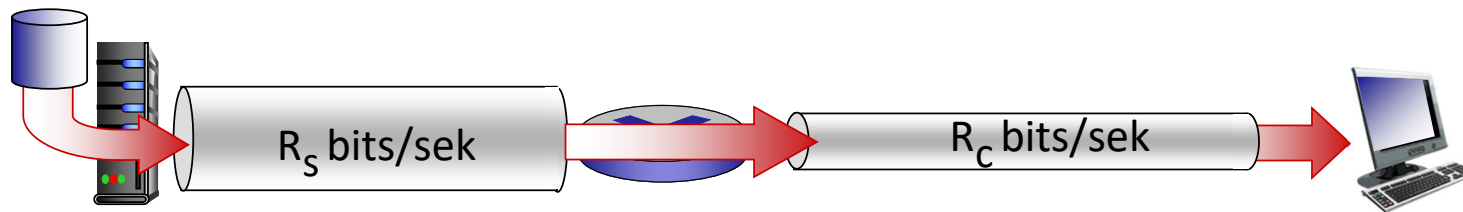


# Throughput

$R_s < R_c$  Hvad er de gennemsnitlige throughput mellem start og mål?



$R_s > R_c$  Hvad er de gennemsnitlige throughput mellem start og mål?

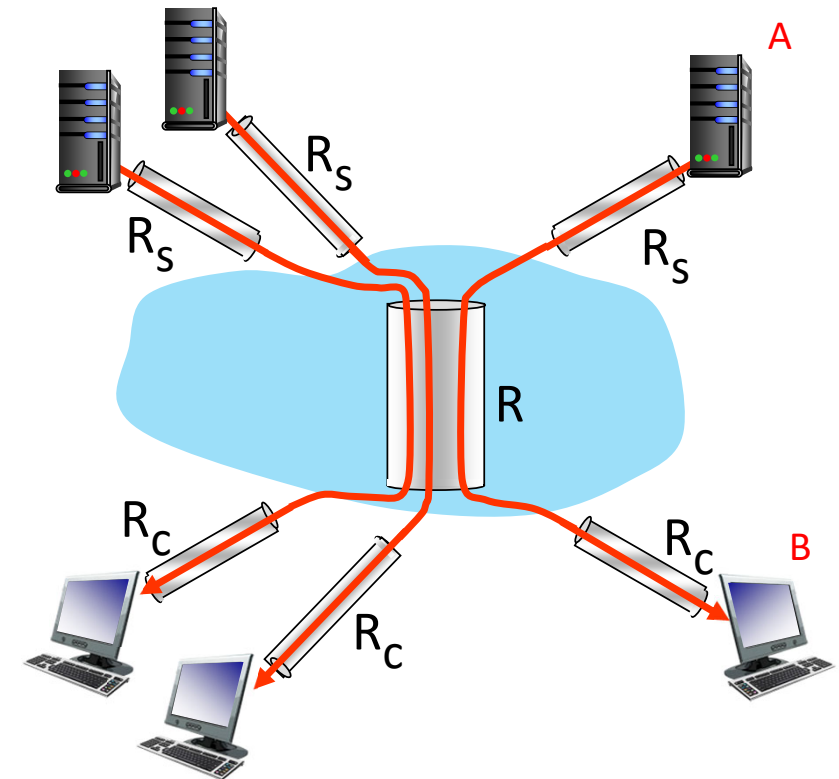


*flaskehals link*

Det link på "end-to-end" stien som begrænser "end-to-end" throughput

# Throughput: Internet Scenarie

- Bestemmes af link med mindst rate:
- Throughput  $\sim \min \left( R_c, R_s, \frac{R}{n} \right)$
- EX,
  - $R_c = 2$  Mbps
  - $R_s = 5$  Mbps
  - $R = 100$  Mbps
  - $N = 10$  client server forbindelser
  - $\Rightarrow$  Throughput  $\sim 2$  Mbps
- I praksis er  $R_c$  eller  $R_s$  ofte flaske-halsen



$N=10$  forbindelser mellem 10 klient/server par, deler  $R$  bits/s fair

# TCP/IP protokol stakken

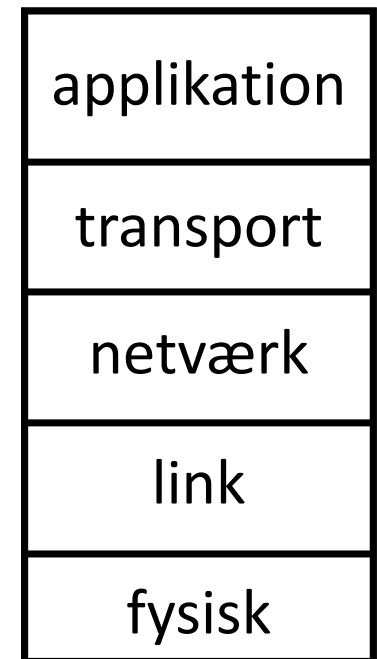
Hvad er en protokol stak?

Hvordan ser en model for lagdeling af internet protokollerne ud?

Hvordan behandles pakker i IP stakken?

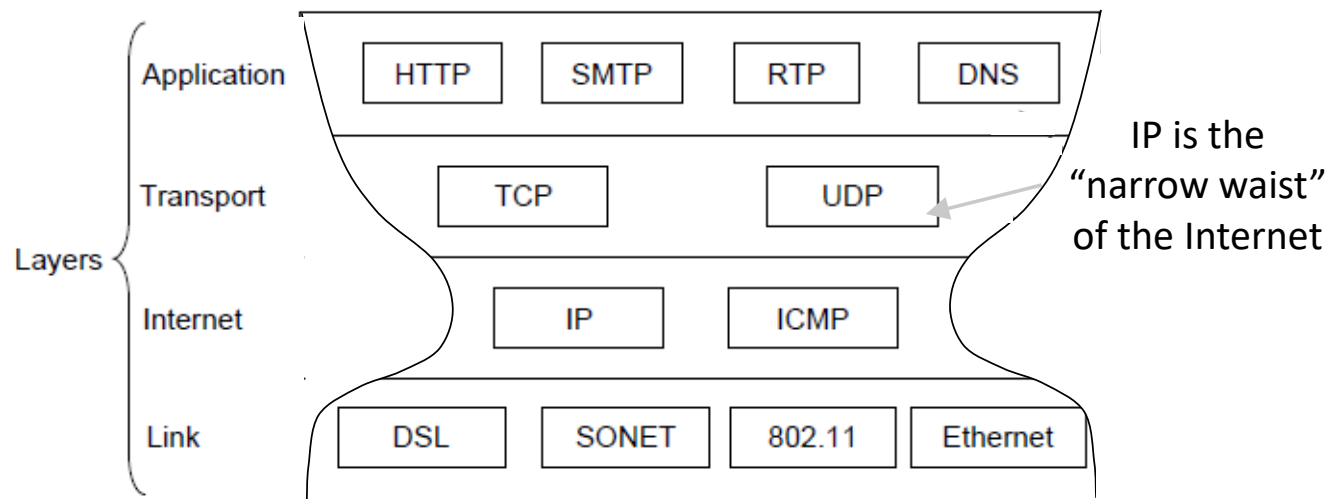
# TCP/IP Reference Model

- **“Protokol stak”** samling af relaterede protokoller, der er organiseret i lag
- **TCP/IP Reference Model:** En 5 lags model for Internet protokollerne , som er uddraget baseret på observationer af et konkret netværk
- **Applikations-lag:** Protokoller som understøtter afvikling af netværks-applikationer (mail, browser, fil-transport, ...)
  - FTP, SMTP, HTTP
- **Transport-lag:** Overfører data fra et kørende program (process) på end-system A til modpart på end-system B
  - TCP, UDP
- **Netværks-lag:** routing og videresendelse af pakker fra source til destination
  - IP
- **Link-lag:** data overførsel mellem direkte naboer i netværket, dvs. koblet sammen med et enkelt link.
  - Ethernet, IEEE-802.11 (WiFi), PPP, SONET
- **Fysiske-lag:** bits “på ledningen”



# TCP/IP Reference Model

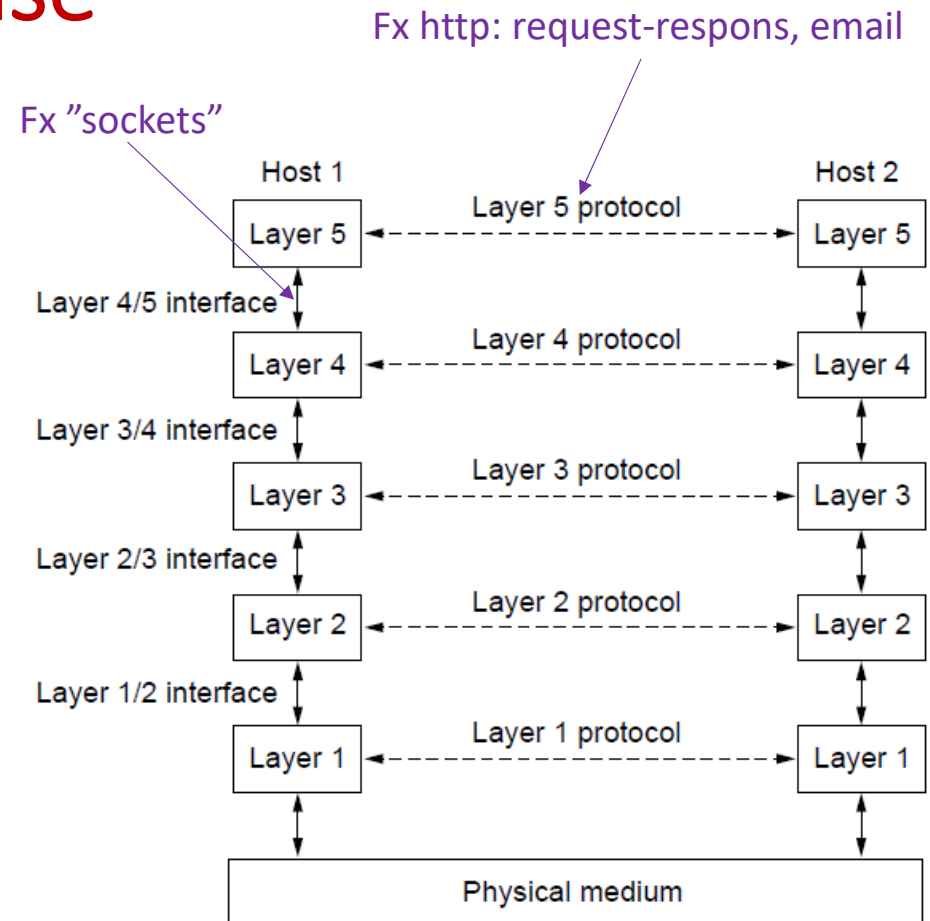
- Snæver-talje/timeglas-model: Hvis vi kan transportere IP på mediet X, virker alle transport og applikations-protokoller også.
- “Mindste fællesnævner”



Udvalgte Internet protokoller er vist på deres respektive lag

# Abstrakt service beskrivelse

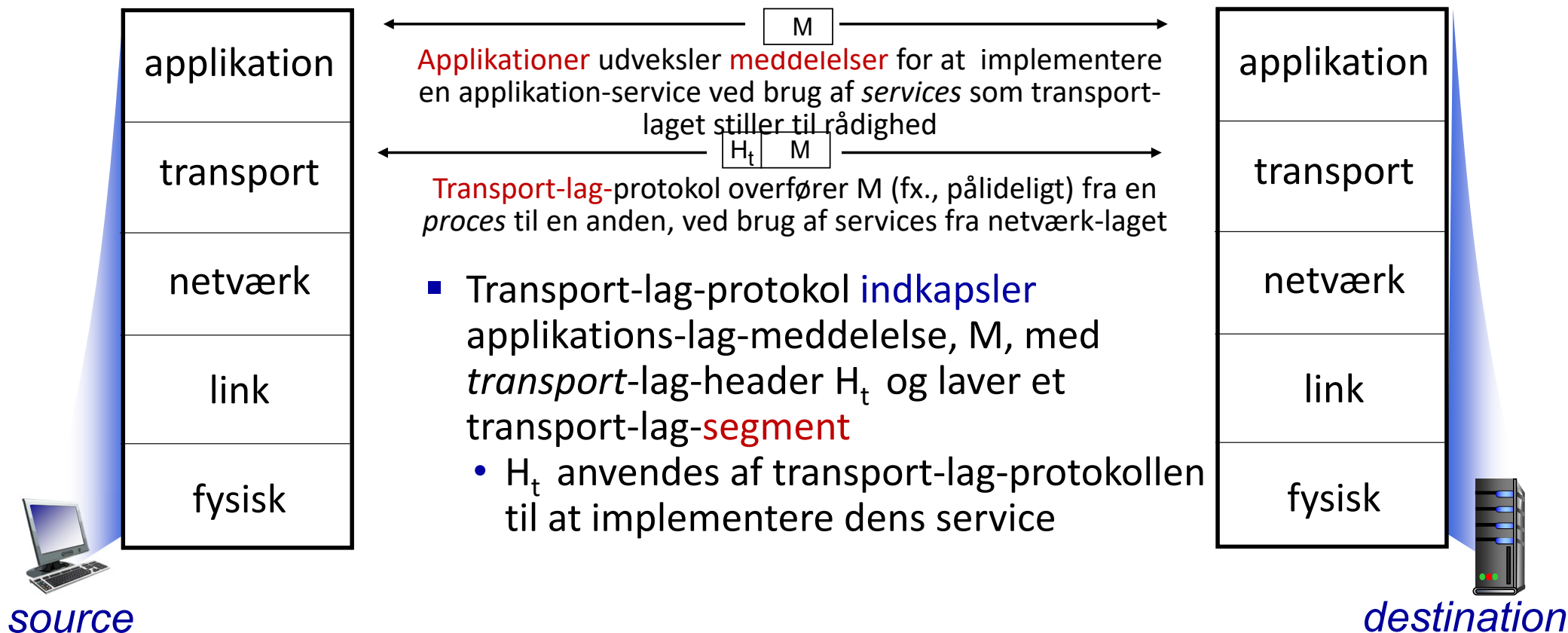
- Hver protokol instans “snakker” virtuelt direkte med dens modpart (ligemand=peer)
- Hvert lag kommunikerer **kun** ved brug af laget under
- Services udbudt af et lavere lag kan tilgås af overliggende lag via et interface
- I bunden overføres meddelelser på et fysisk medie



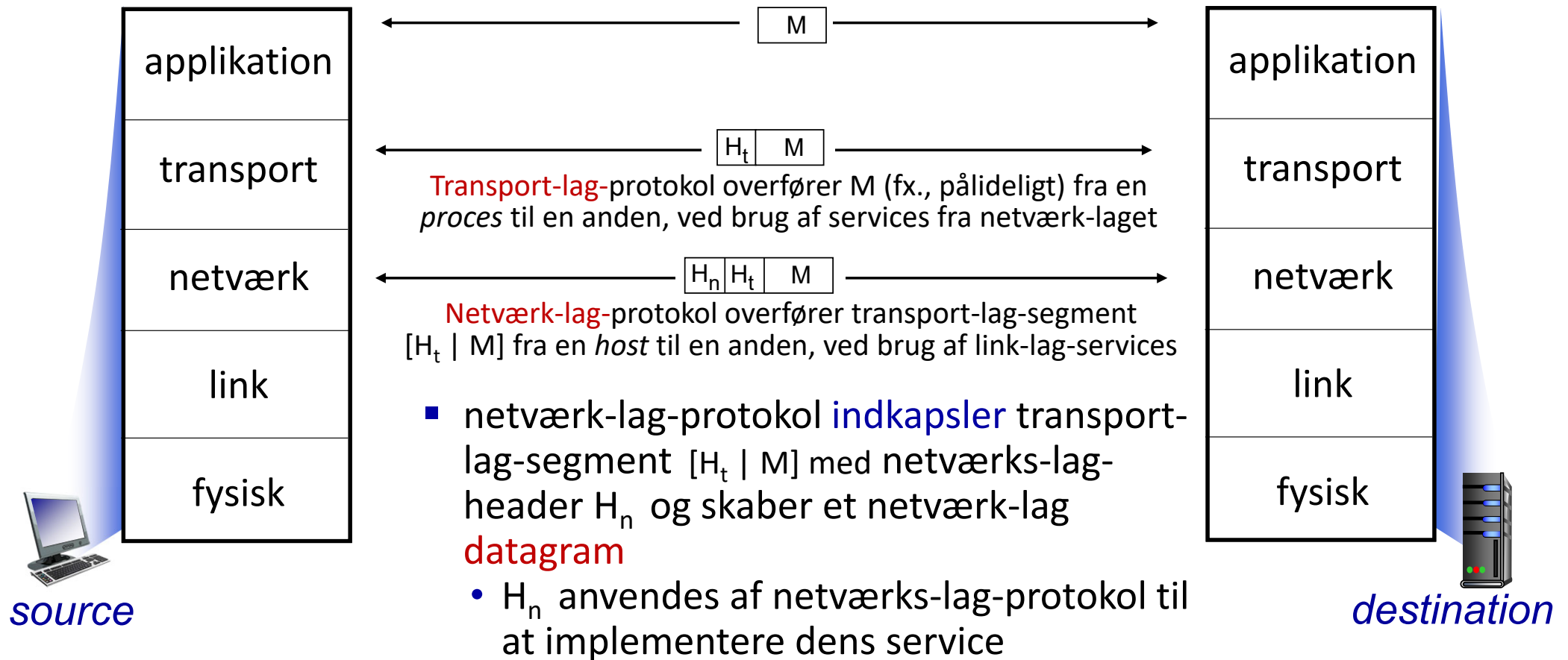
Services, Lag-opdeling, og  
indkapsling



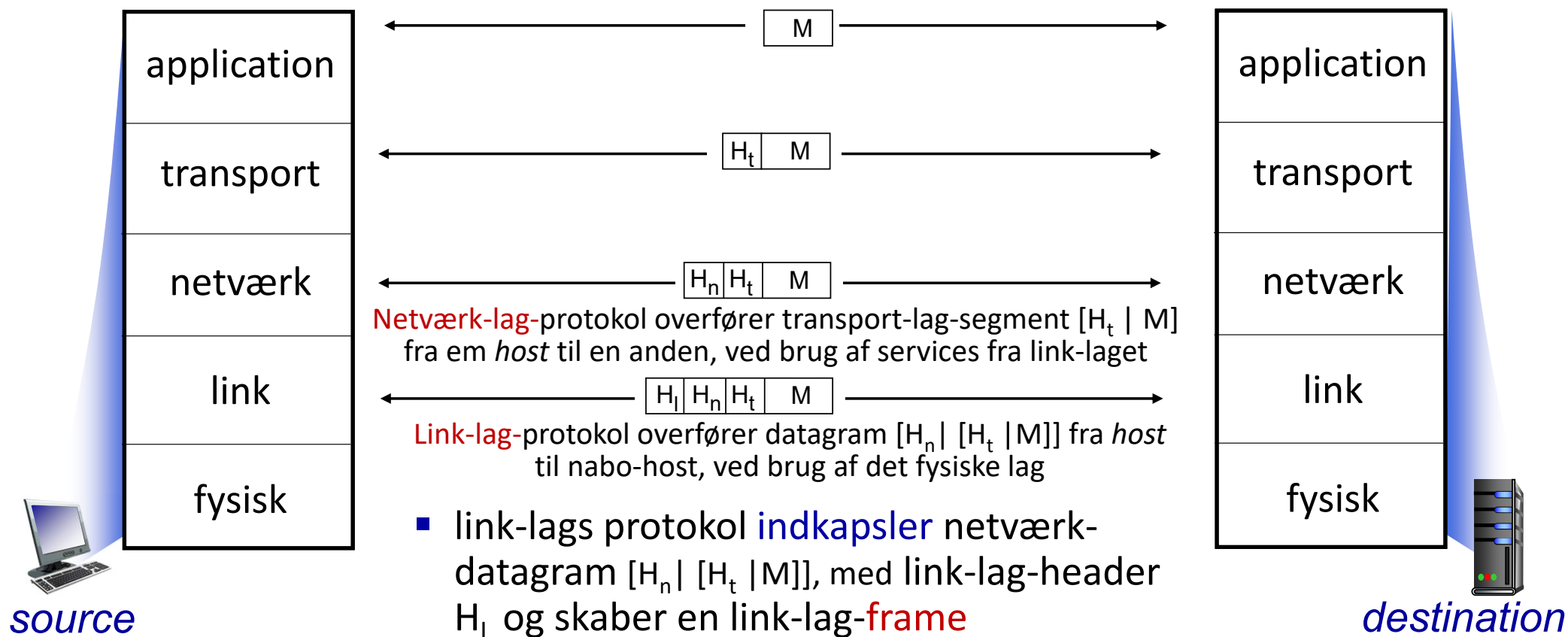
# Services, Lag-opdeling, og indkapsling



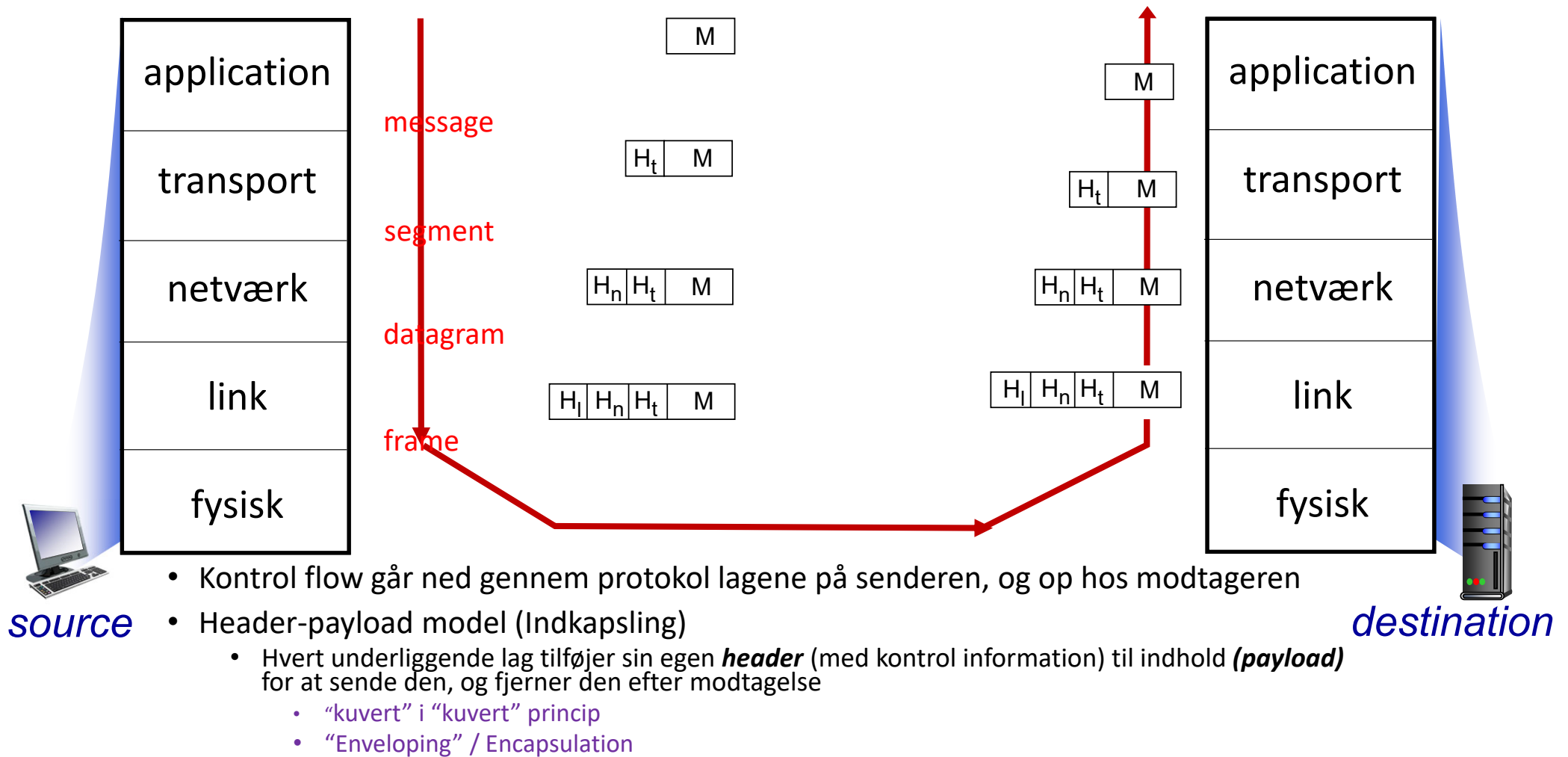
# Services, Lag-opdeling, og indkapsling



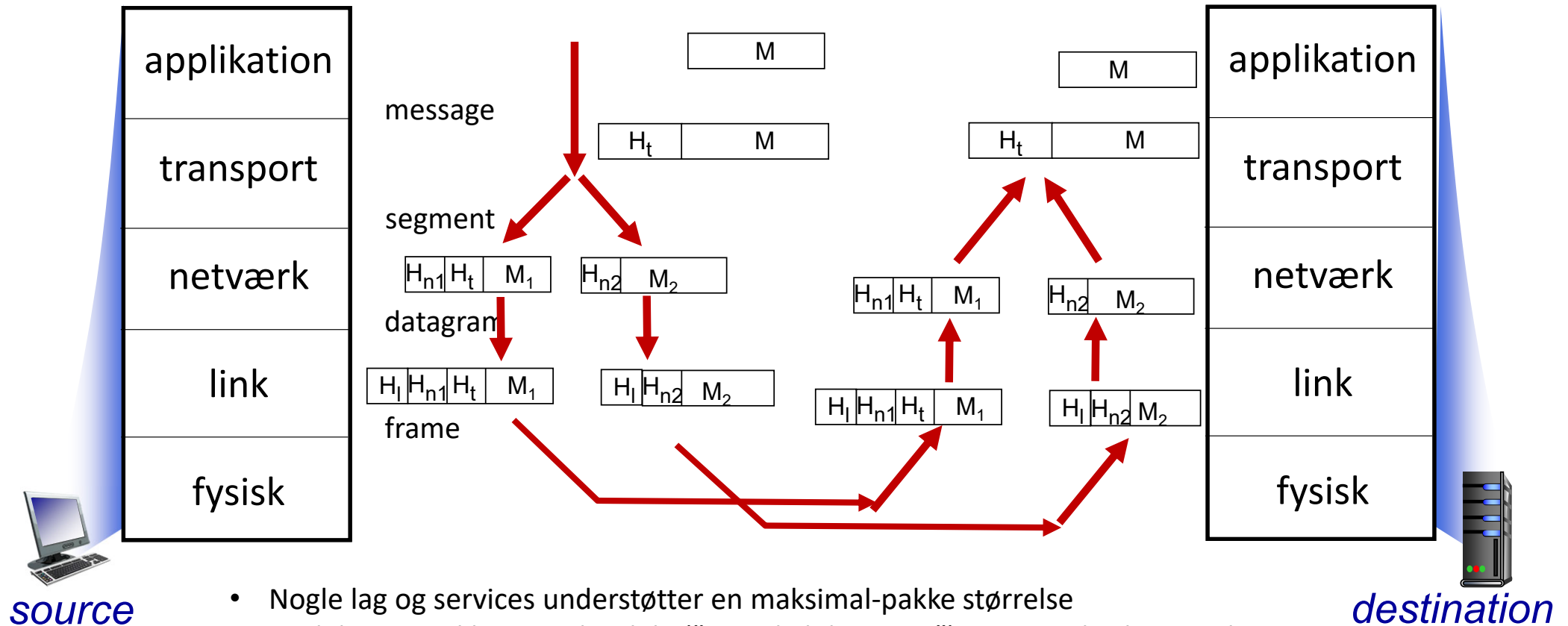
# Services, Lag-opdeling, og indkapsling



# Services, Lag-opdeling, og indkapsling



# Segmentering og gen-samling



# Hvorfor lagdeling?

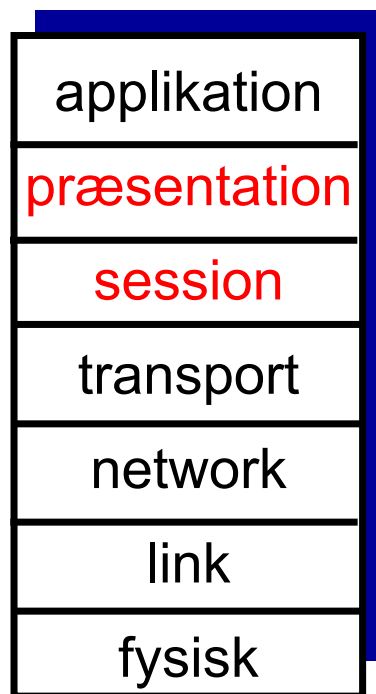
- **Simplificerer opbygning af komplekse systemer:**
  - Eksplicit struktur tillader at enkelt dele og deres sammenhæng kan identificeres og forstås i mindre bidder
  - Opbygning i simple moduler letter vedligehold og opdatering af et system
    - Ændring af implementationen af en service kan foretages uden at ændre resten af systemet
- **Ulemper ved lagdeling ?**
  - Nogle funktioner kan blive besværlige eller performance mæssigt dyre at implementere
    - Statefull firewalls
    - NAT

# Hvorfor lagdeling?

- **Simplificerer opbygning af komplekse systemer:**
  - Eksplicit struktur tillader at enkelt dele og deres sammenhæng kan identificeres og forstås i mindre bidder
  - Opbygning i simple moduler letter vedligehold og opdatering af et system
    - Ændring af implementationen af en service kan foretages uden at ændre resten af systemet
- **Ulemper ved lagdeling ?**
  - Nogle funktioner kan blive besværlige eller performance mæssigt dyre at implementere
    - Statefull firewalls
    - NAT
- **VIGTIGT GENERELT DATALOGISK PRINCIP:**
  - Opdeling af kompleks funktionalitet i abstraktions-/virtualiserings-lag hver med vel-defineret funktionalitet, som kan være mere eller mindre strikt indkapslet

# OSI Reference Model

- En mere “principiel” model, standardiseret 7-lags model
  - Open Systems Interconnection Reference Model
  - International Organization for Standardization ("ISO 7498-2")



- Giver funktioner som brugere har behov for
- Konvertering mellem forskellige data-repræsentationer: gør at applikationer kan fortolke data ud fra deres betydning, f.x. kryptering, komprimering, maskine specifikke konventioner
- Styrer en dialog: synkronisering, checkpointing, gen-oprettelse af data udveksling
- Giver end-to-end levering
- Sender datagrammer over multiple links
- Sender frames med data
- Sender bits kodet som "signaler"



# Kritik af OSI & TCP/IP

## OSI:

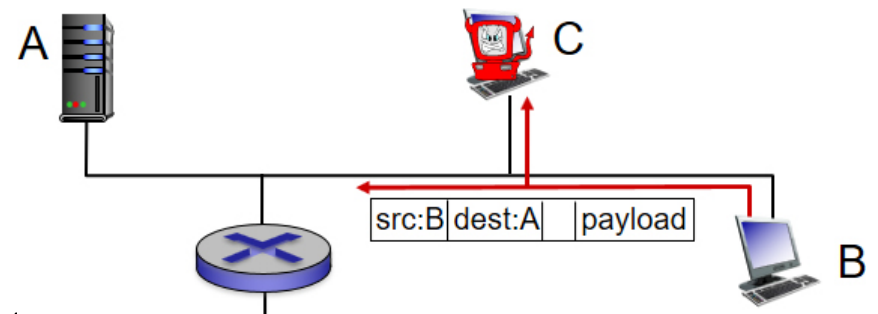
- + Meget indflydelsesrig model med klare begreber
- Modellen, protokollerne og adoption hæmmet af politik og kompleksitet

## TCP/IP:

- + Meget succesfulde protokoller, som virker godt og er meget udbredte
- Svag model, som er ***udledt efterfølgende*** ud fra de aktuelle protokoller

# Sikkerhed på nettet???

- Nettet blev oprindeligt lavet for en mindre lukket brugerskare med tillid til hinanden
  - "sikkerhed" er ikke indbygget, men tilføjet ved nye protokoller og opsætning af begrænsninger på hvad videresendes
- Pakker kan "sniffes" (wireshark)
  - Broadcast medier a la WiFi
  - Installeret på router
- Pakker kan "spoofes" (opfind din egen header)
- Denial of service angreb, DOS,...
- ...



# Packet sniffing med Wireshark

The image shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons for packet capture and analysis. The main window is divided into three panes:

- Packets Pane:** Displays a list of captured packets. The selected packet is 17, a UDP packet from 192.168.0.102 to 192.168.0.111, length 334 bytes, containing an HTTP 200 OK response.
- Packet Details Pane:** Shows the hierarchical structure of the selected packet. It includes Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Simple Service Discovery Protocol (SSDP).
- Packet Bytes Pane:** Displays the raw packet data in hexadecimal and ASCII format.

The detailed view of the selected packet (17) is as follows:

- Ethernet II:** Src: PhilipsL\_70:a2:32 (00:17:88:70:a2:32), Dst: IntelCor\_97:47:4e (f4:4e:e3:97:47:4e)
- Internet Protocol Version 4:** Src: 192.168.0.102, Dst: 192.168.0.111
  - Version: 4
  - Header Length: 20 bytes (5)
  - Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  - Total Length: 320
  - Identification: 0x2a69 (10857)
  - Flags: 0x40, Don't fragment
  - Fragment Offset: 0
  - Time to Live: 64
  - Protocol: UDP (17)
  - Header Checksum: 0x8d1e [validation disabled]
  - [Header checksum status: Unverified]
  - Source Address: 192.168.0.102
  - Destination Address: 192.168.0.111
- User Datagram Protocol:** Src Port: 1900, Dst Port: 52625
  - Source Port: 1900
  - Destination Port: 52625
  - Length: 300
  - Checksum: 0x6887 [unverified]
  - [Checksum Status: Unverified]
  - [Stream index: 0]
  - [Timestamps]
  - UDP payload (292 bytes)
- Simple Service Discovery Protocol:**
  - HTTP/1.1 200 OK\r\n
  - HOST: 239.255.255.250:1900\r\n
  - EXT:\r\n
  - CACHE-CONTROL: max-age=100\r\n
  - LOCATION: http://192.168.0.102:80/description.xml\r\n
  - SERVER: Hue/1.0 UPnP/1.0 IpBridge/1.50.0\r\n
  - hue-bridgeid: 001788FFFE70A232\r\n
  - ST: upnp:rootdevice\r\n
  - USN: uuid:2f402f80-da50-11e1-9b23-00178870a232::upnp:rootdevice\r\n
  - \r\n

The raw packet data in the bottom pane shows the hexadecimal and ASCII representation of the packet bytes, including the SSDP message structure.

# Opgaverne idag

- Review: Har man forstået grundlæggende begreber
  - Packet switching, forsinkelser, flaskehalse, protokol stak
- Øvelser: Kan man anvende dem i nye eksempler?
  - Konkrete delay beregning
  - Flaskehalse
- Praktiske: Kan anvende netværkssværktøjerne
  - Traceroute, start på Wireshark.

SLUT