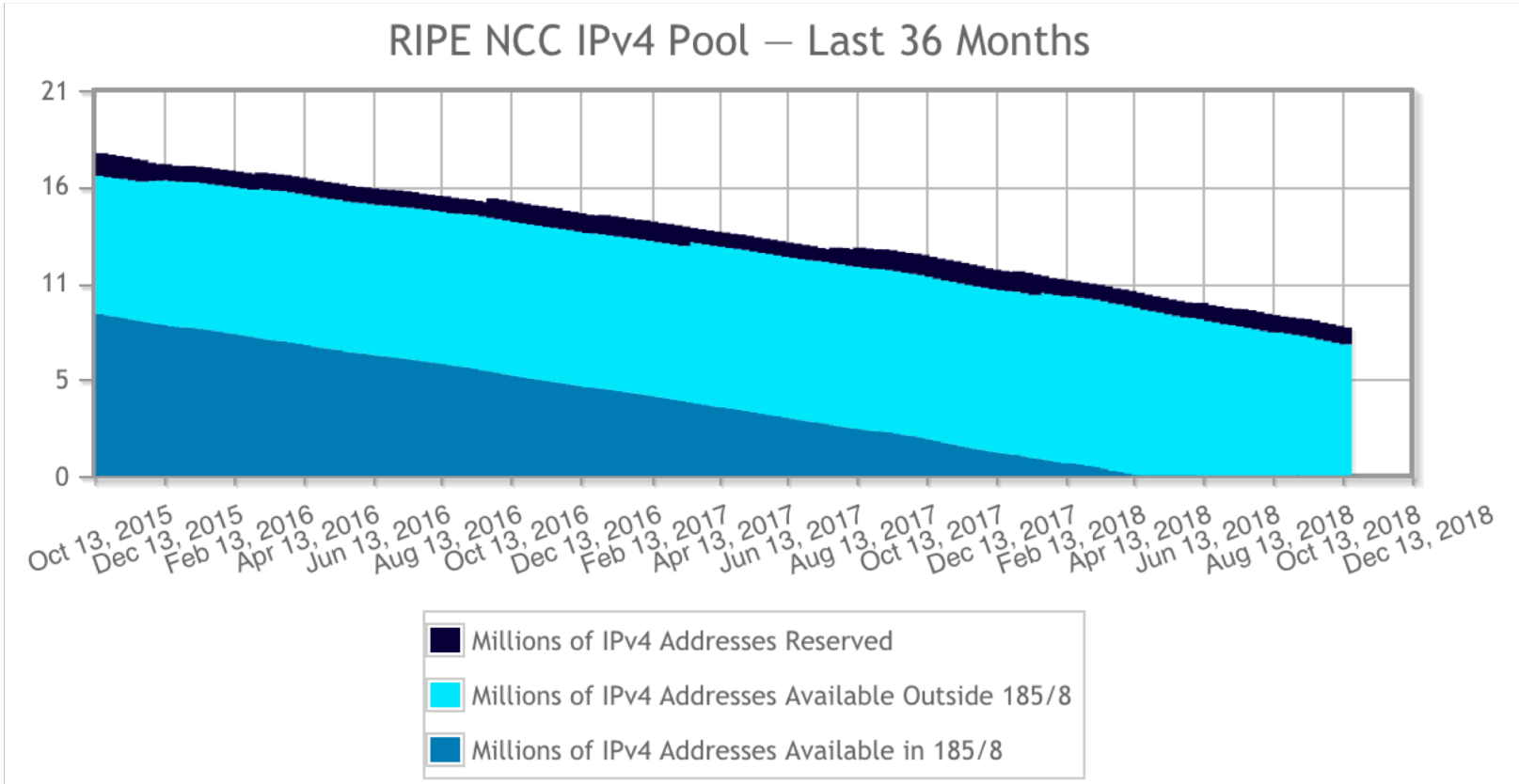
IPv4 -> IPv6

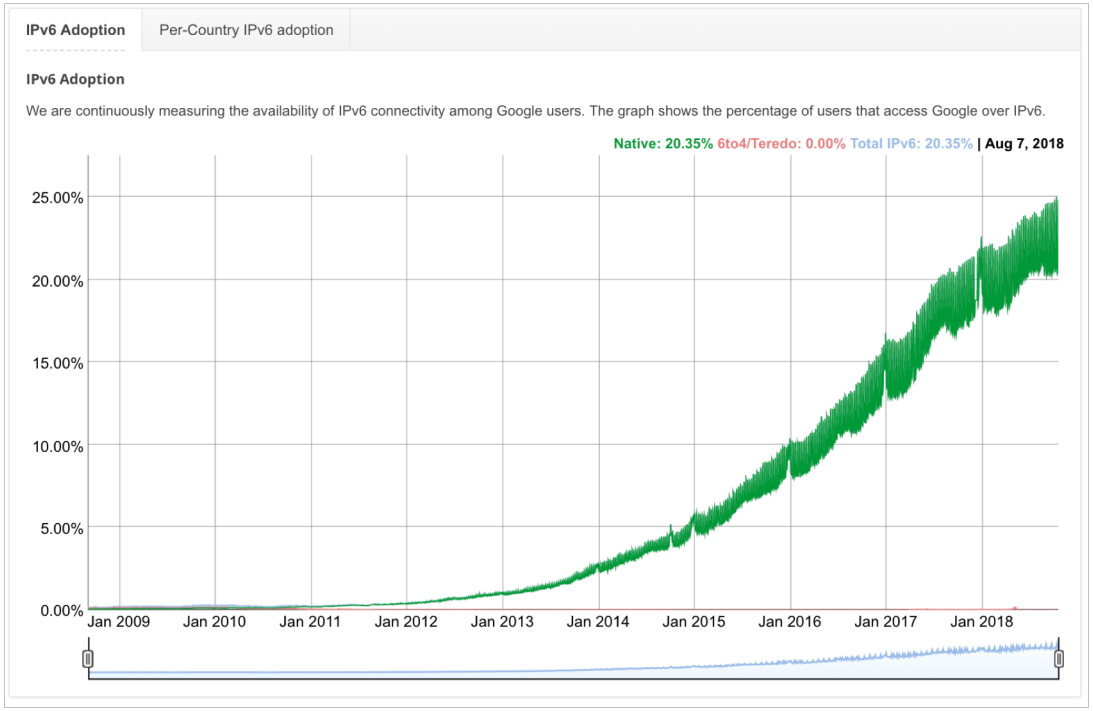


Selv med NAT i bruk på svært mange nettverk, er antall tilgjengelige IPv4-adresser i ferd med å bli kritisk lavt.

Kilde – ripe.net: https://www.ripe.net/publications/ipv6-info-centre/about-ipv6/ipv4-exhaustion/ipv4-available-pool

IPv6

- Lang prosess med å få i drift siden det må støttes i alle noder fra ende-til-ende
- 128 bits IP-adresse (mot 32 i IPv4)
- 2^{128} (eller $3.4 \times 10^{38)}$ mulige adresser

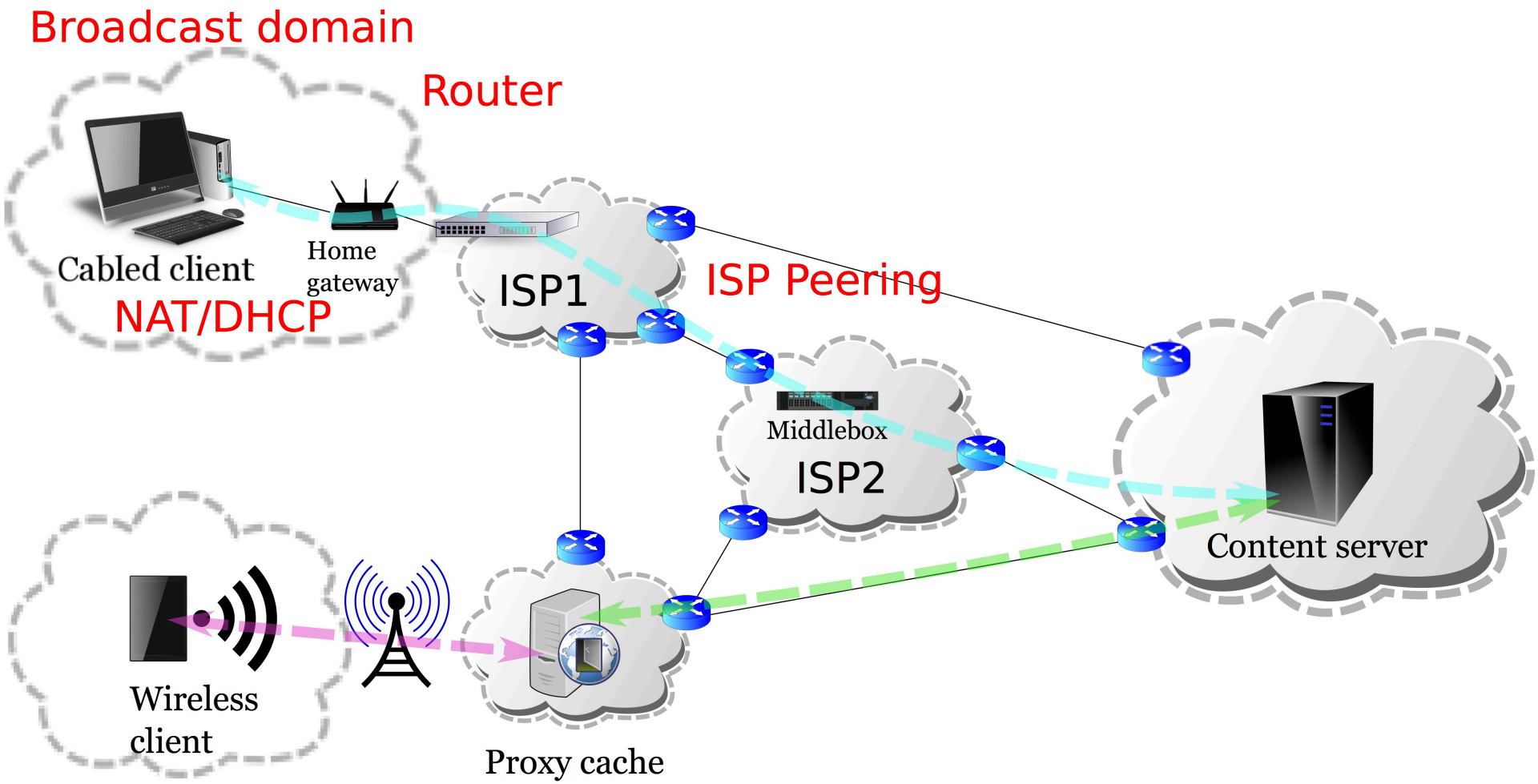


Kilde - google: https://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoption&tab=ipv6-adoption

Ruting i Internett (svært kort)

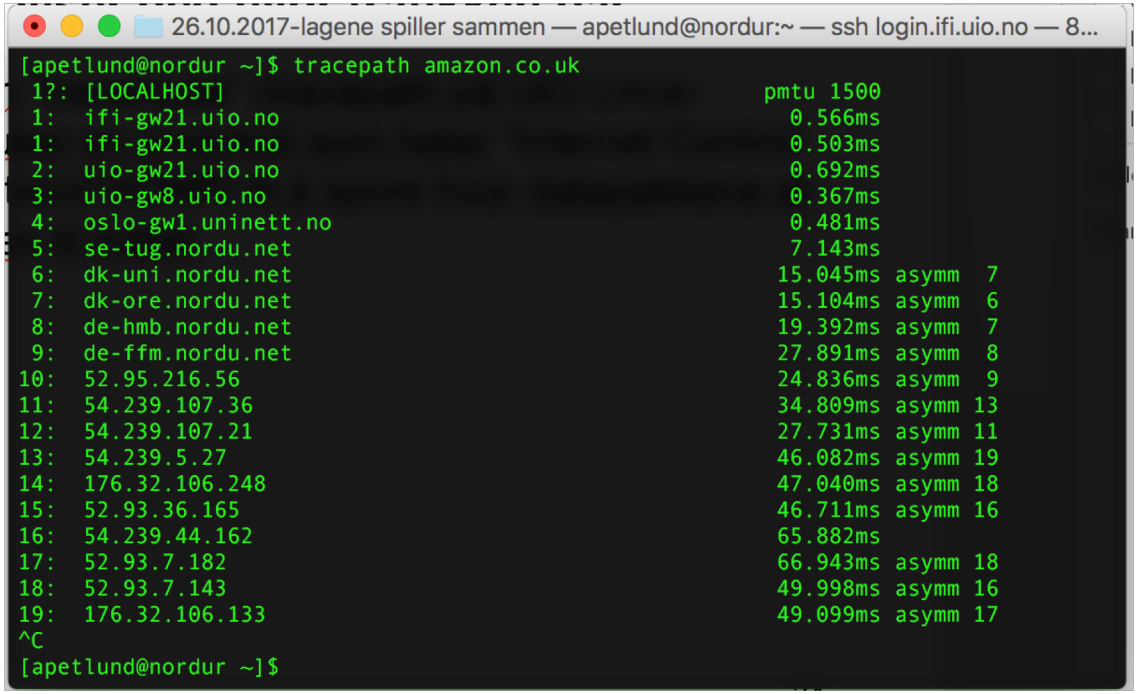
- Målet for ruting: å videresende en datapakke slik at den til slutt når måladressen sin.
- En Internet Service Provider (ISP) driver et nettverk og leverer datatransport til andre ISPer og sluttbrukere.
- ISP Peering er når ISPer inngår avtaler om å videresende hverandres trafikk. Økonomiske prinsipper er da med på å bestemme hvor trafikken flyter.
- Border Gateway Protocol (BGP)– Rutingprotokoll som brukes mellom ISPer. Svært store selskaper kan også bruke BGP.
- OSPF – eksempel på rutingalgoritme for mindre nettverk. Bygger et kart over mulige ruter til måladressene. Skalerer ikke for store nettverk.

Ruting i Internett (svært kort)



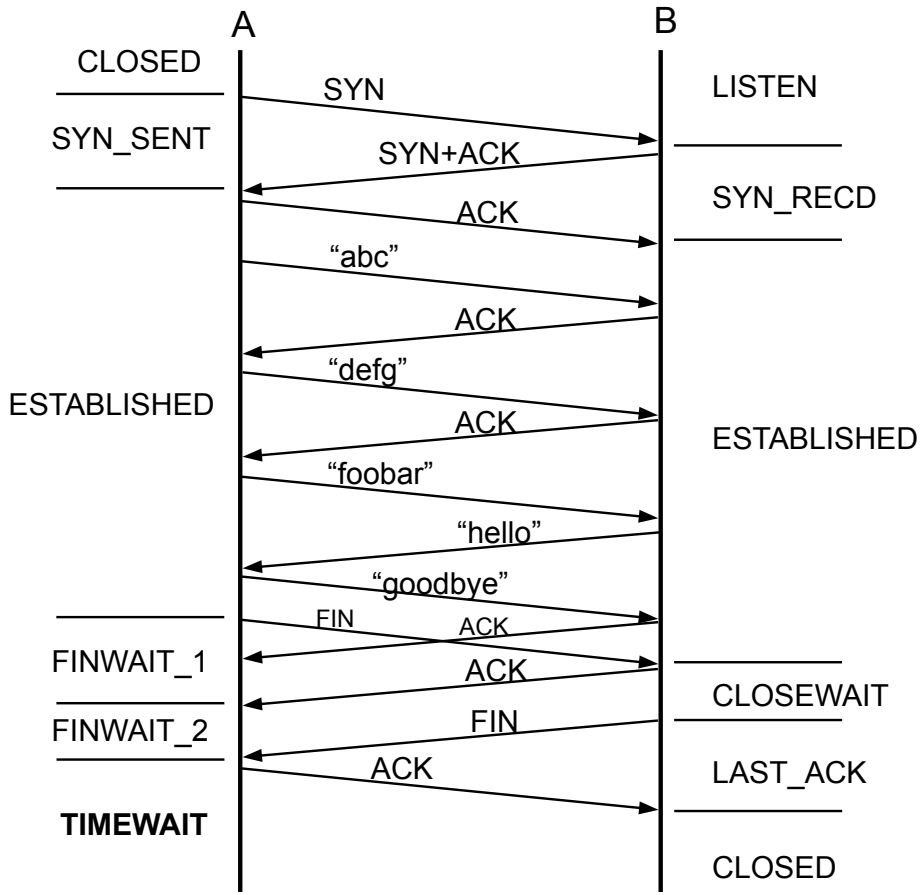
Traceroute – viser deg hvor trafikken går

- Kommandoen ”traceroute” (tracepath på UiO Linux-maskiner) bruker en protokoll som heter ”Internet Control Message Protocol”(ICMP) til å spore hvor datapakkene er innom på veien til målet.



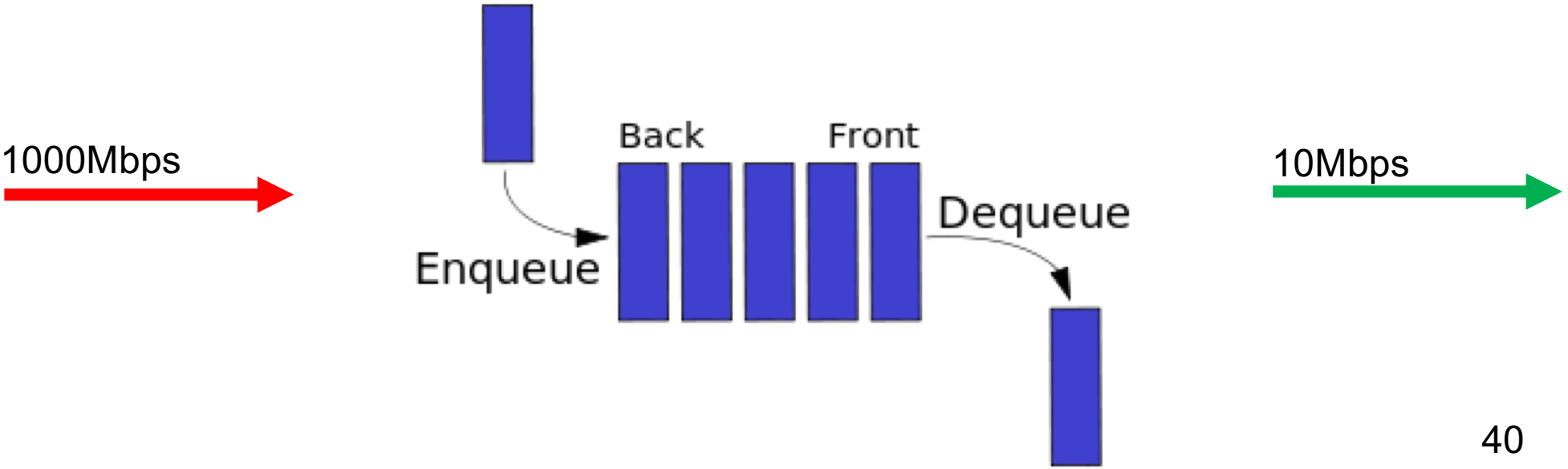
Transmission Control Protocol (TCP)

- Forbindelsesorientert
  - Settes opp ved et 3-veis-håndtrykk
    - SYN-SYN+ACK-ACK (se figur)
- Flytkontroll
  - Ikke sende fortere enn mottageren kan ta imot
- Metningskontroll
- Byte-strøm og levering i rekkefølge
- Pålitelighet
  - Implementert ved at bekreftelser på hver pakke sendes tilbake fra mottakeren
- Feilsjekking av nyttelasten (sjekksum)

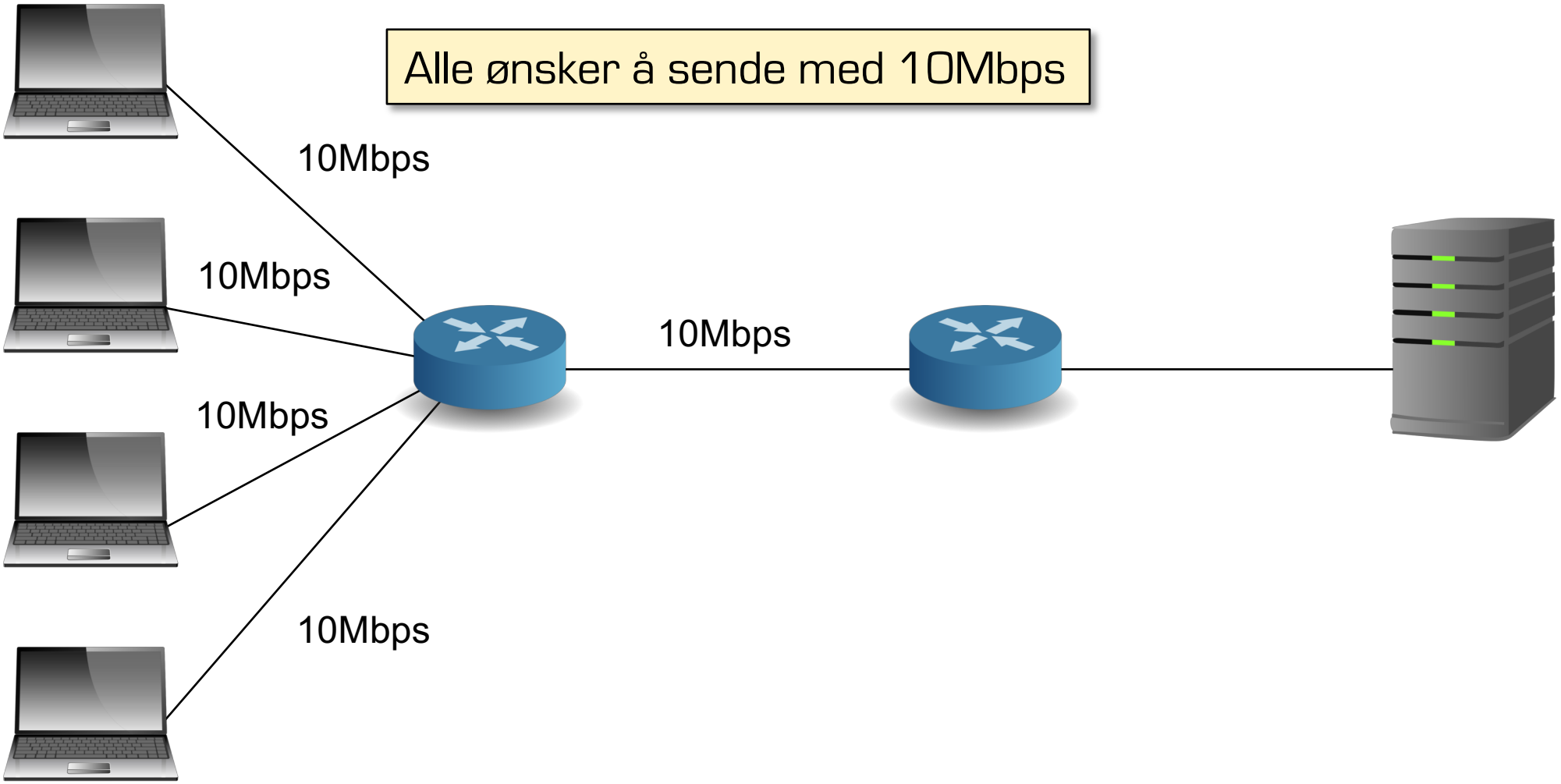


Metningskontroll

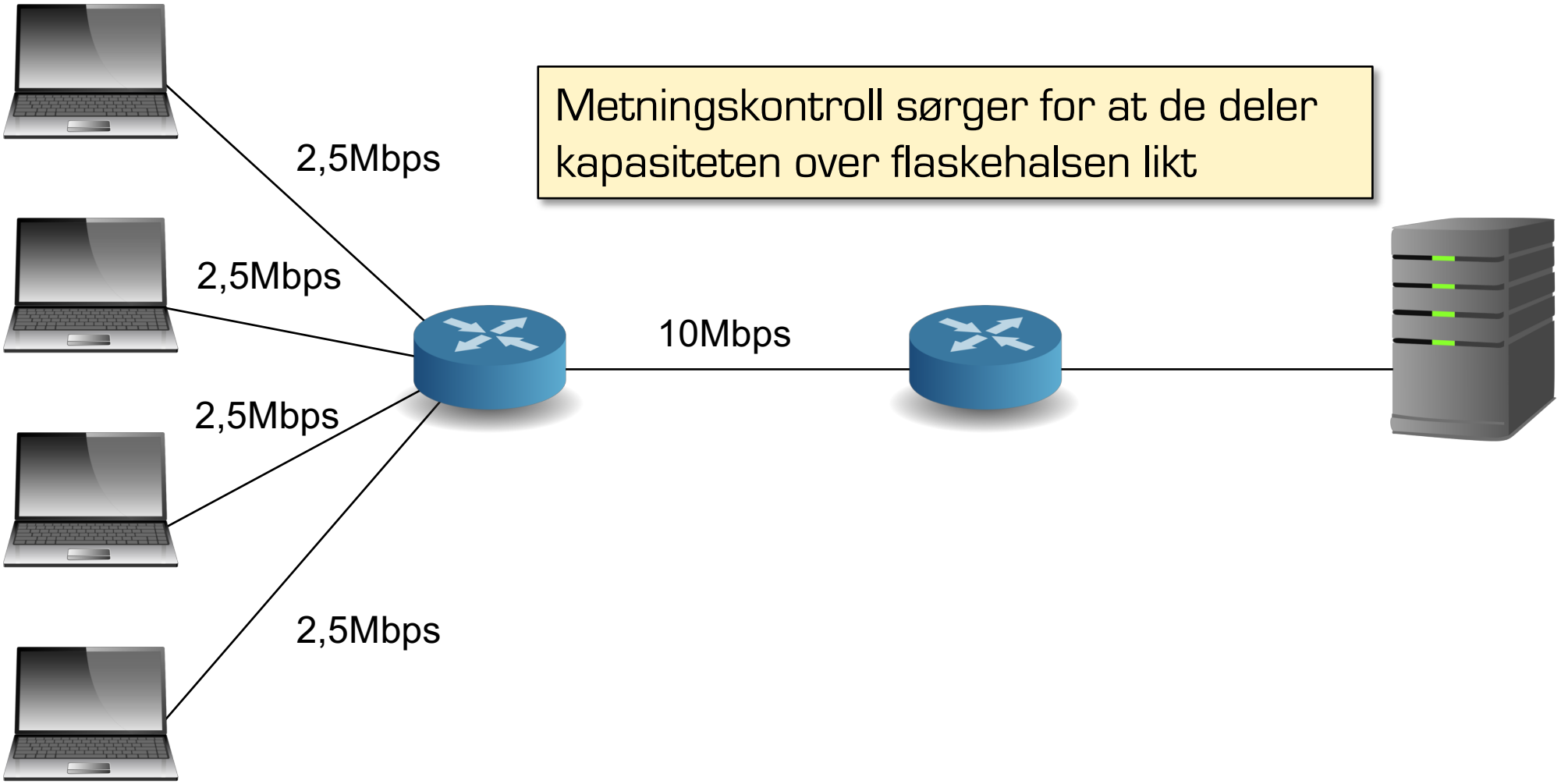
- En ruter er i utgangspunktet bare en FIFO (first-in-first-out) kø.
- Om det blir for mye trafikk over en gitt kø, vil det føre til stopp i trafikken. Dette kalles ”congestion”. Om det er så mye traffikk at det blir full stopp for alle, kalles det “congestion collapse”.
- For å unngå dette bygde man inn mekanismer i TCP for å tilpasse senderaten til nettverksforholdene.



fortsatt metningskontroll

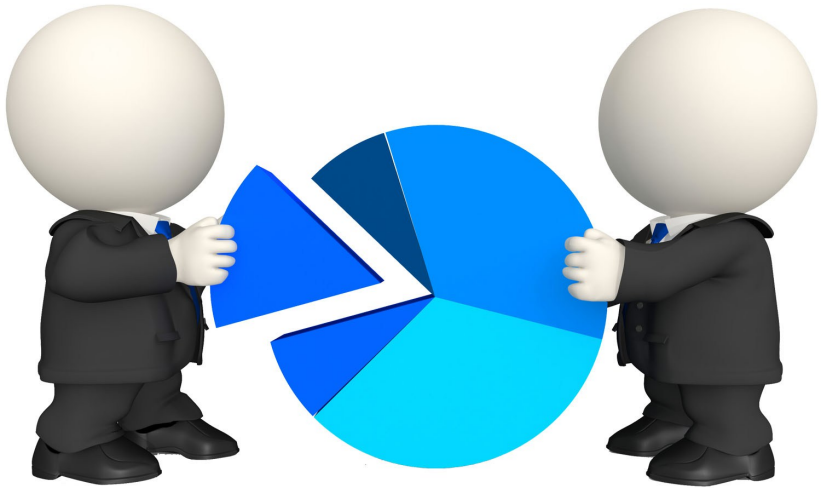


ennå ikke ferdig med metningskontroll

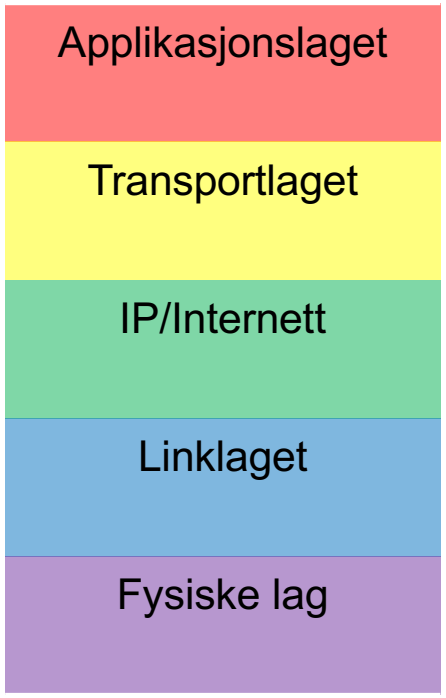


stadig metningskontroll

- Ressursene deles teoretisk sett likt mellom **forbindelser**
- For å kapre mer av kapasiteten: åpne flere forbindelser I paralell.
- Ingen god metode for å fordele ressurser i Internett rettferdig pr. maskin eller pr. bruker i dag.



Kryptering / sikkerhet i lagene



Secure Sockets Layer – kryptering for ende-til-ende Applikasjoner – f.eks nettbank eller butikker.

F.eks. tcpcrypt– har som mål at alle TCP-forbindelser som settes opp skal være kryptert

VPN (IPSEC etc.) – kobler to subnett sammen så det fungerer som ett LAN selv om de er fysisk adskilt

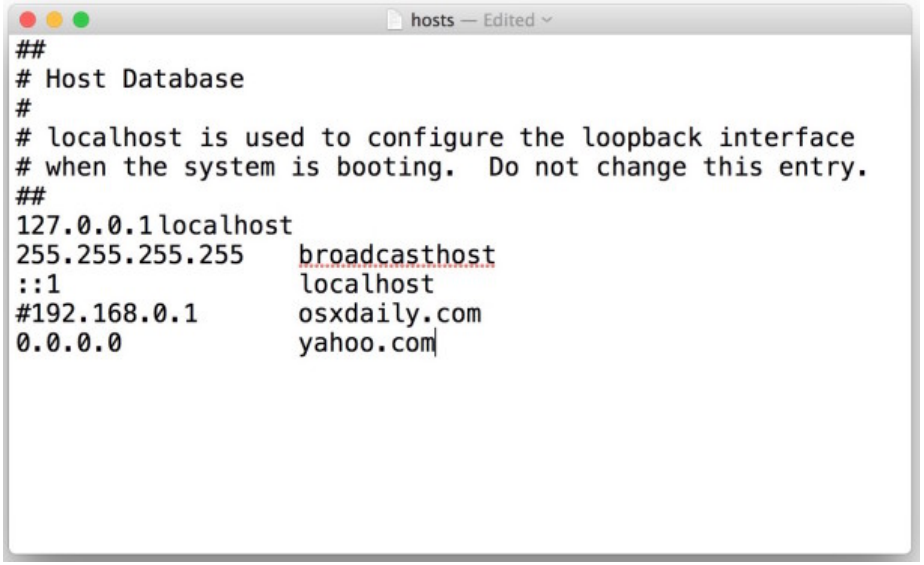
Kryptering på flere lag gjør det vanskeligere for uvedkomne å lytte til kommunikasjonen.
Adressen (avsender / mottakeren) er vanskelig å kryptere, da routere må vite hvor pakken skal leveres.

Hvordan koble til en annen maskin?

Med IP-adresse?

- `telnet 127.0.0.1 23`
  snakker med min egen maskin
  - Brukes ofte da det er en veldefinert adresse

- `wget http://173.194.39.31:80/`
  snakker med en av Google sine maskiner
  mulig å huske, men ikke praktisk.



- `ssh 9.228.93.3`
  forsøke å kontakte en maskin du visste hadde denne adressen i 2001
  umulig å huske om du ikke bruker den daglig

- det er mulig å lage alias for en IP-adresse i filen /etc/hosts (Unix variants)
- Opprinnelig administrert av Stanford Research Institute. Endringer ble distribuert via epost ☺

Hvordan koble til en annen maskin?

Løsning: bruk "fornuftige" navn

- som f.eks.

```
ssh login.ifi.uio.no
wget www.google.com
```

- Ikke bare lettere å huske
- Har også en hierarkisk struktur (gjenspeiler organisasjonen)

Møt **Domain Name System (DNS)**

DNS - overblikk

Domain Name System

Hierarkisk navnetilordning

I motsetning til den originale flate strukturen i /etc/hosts

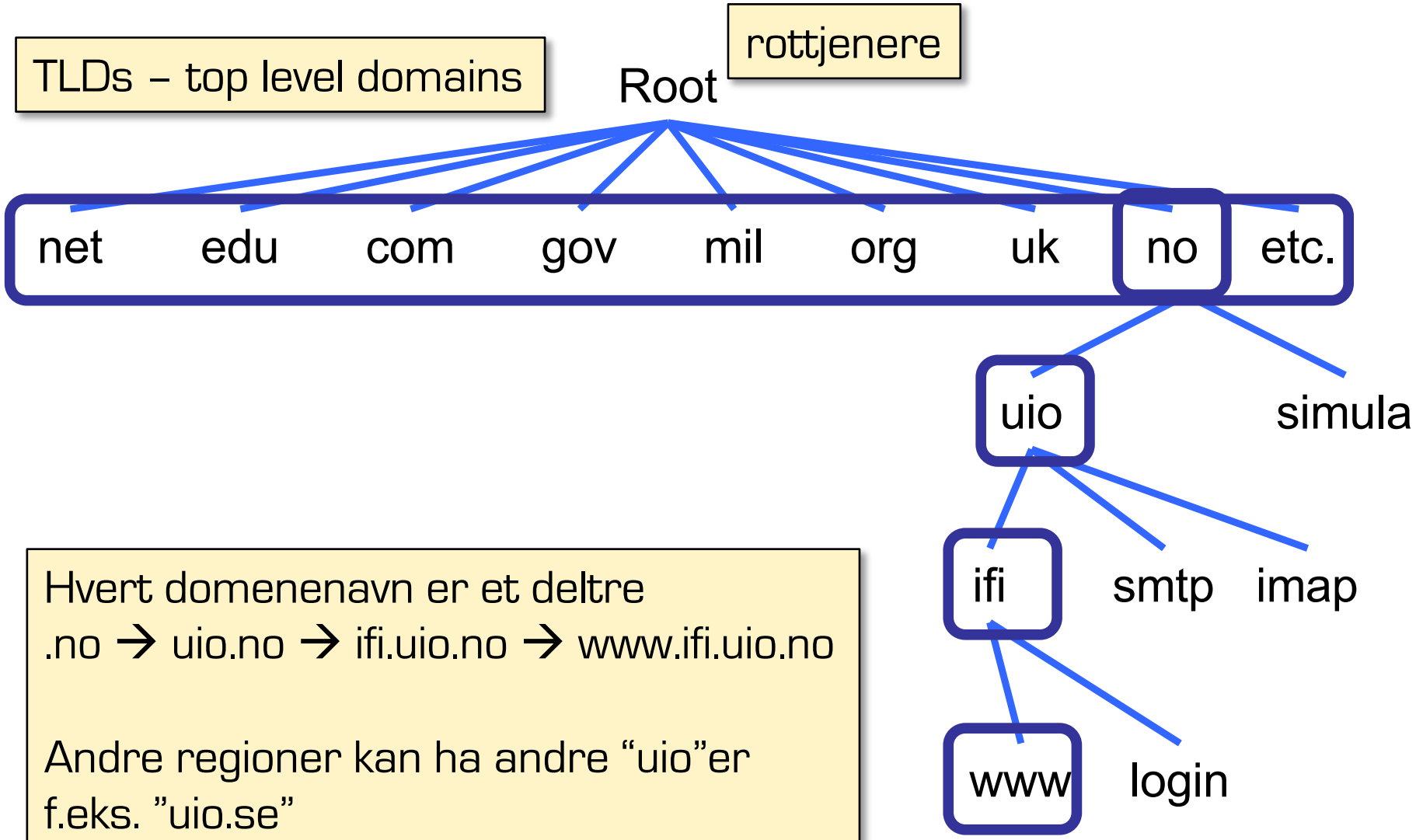
f.eks.: .com → google.com → mail.google.com

Distribuert database

Enkel klient/tjener arkitektur

- UDP eller TCP port 53
- tjenere må bruke TCP seg i mellom (fra nylig)
- klienter som bruker TCP avvises ofte
  - reduserer lasten på DNS-tjeneren

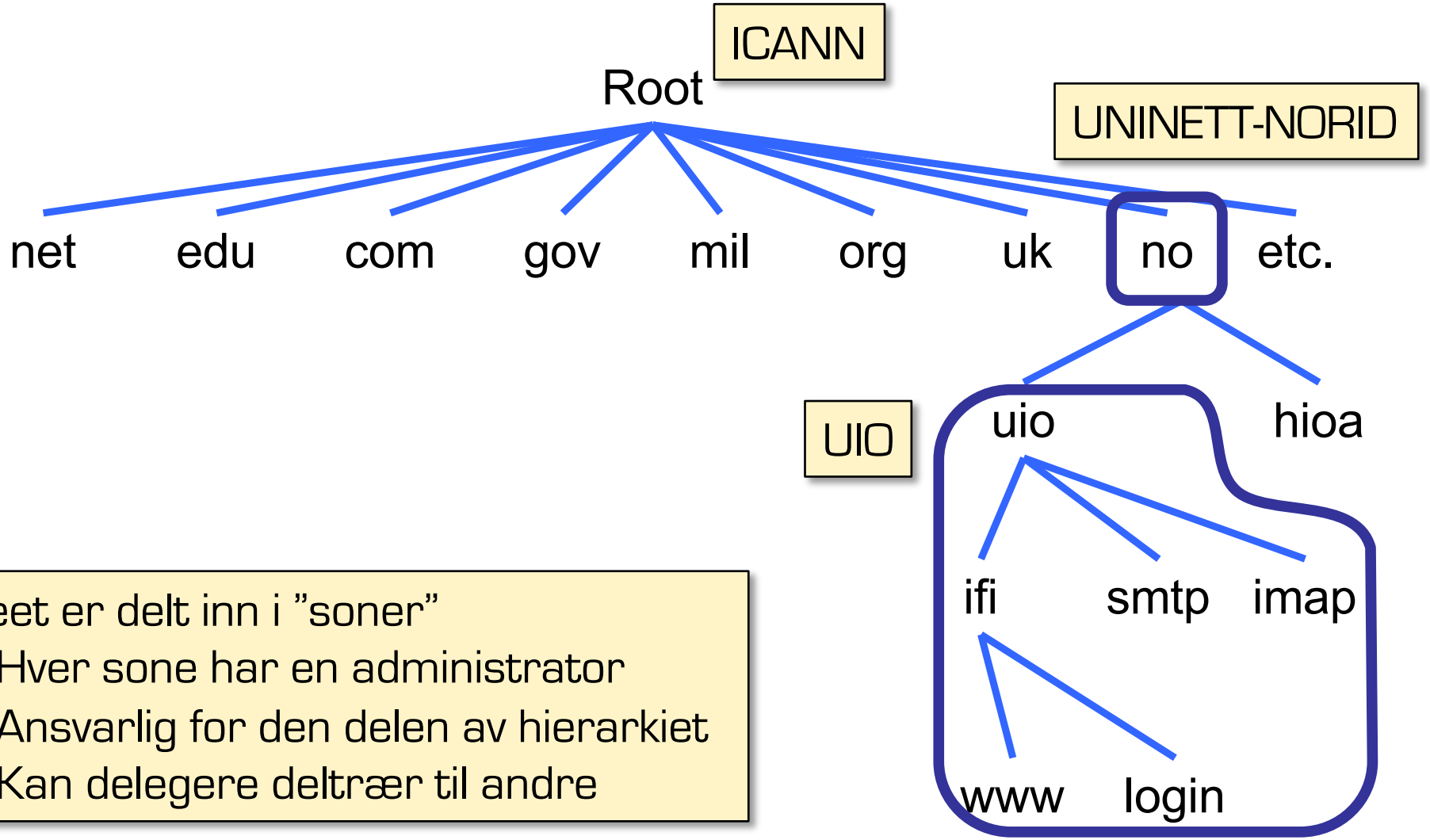
Navnehierarki



Hvert domenenavn er et deltre
.no → uio.no → ifi.uio.no → www.ifi.uio.no

Andre regioner kan ha andre "uio"er
f.eks. "uio.se"

Administrasjon av navnehierarkiet



Treet er delt inn i ”soner”

- Hver sone har en administrator
- Ansvarlig for den delen av hierarkiet
- Kan delegere deltrær til andre

Tjenerhierarki

Funksjonene til hver DNS-tjener

- Autoritet over en del av hierarkiet
  - Ikke behov for å lagre alle DNS-navn
- Lagre alle oppføringene for maskiner/domener i sin sone
  - **Må** replikeres for å sikre oppetid (minst 2 tjenere)
- Må kjenne adressene til rottjenerne
  - Slå opp forespørsler på navn som den ikke lagrer selv

Rottjenerne kjenner til alle TLDene (Top Level Domains)

Rottjenere

Ansvarlige for ”Root Zone File”

- Liste over TLDer og hvem som kontrollerer dem

```
com.            172800   IN      NS      a.gtld-servers.net.
com.            172800   IN      NS      b.gtld-servers.net.
com.            172800   IN      NS      c.gtld-servers.net.
```

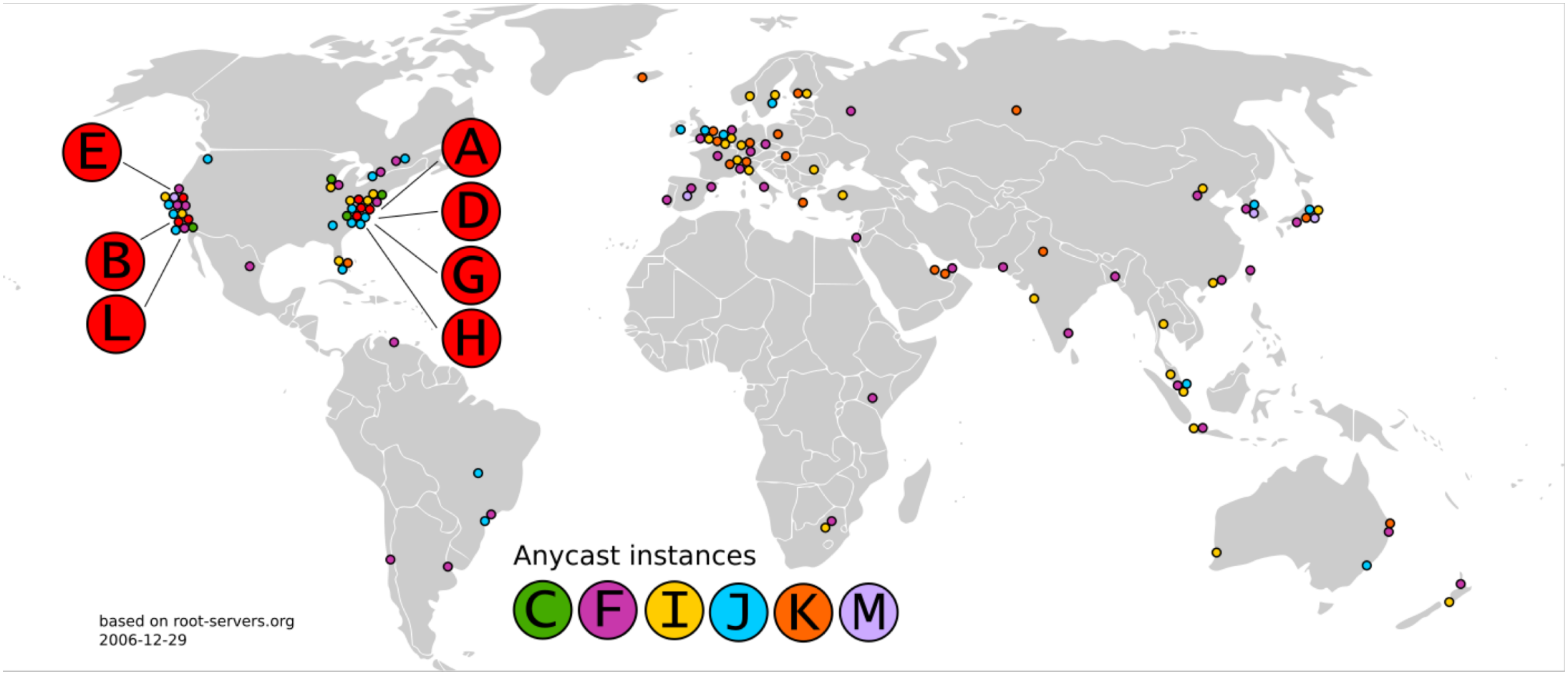
Administrert av “Public Technical Identifiers" (PTI)

- 13 rottjenere servers, labeled A→M
- 6 er replikert globalt med teknikken ”anycast”

Kontaktes når man mislykkes med navneoppslag

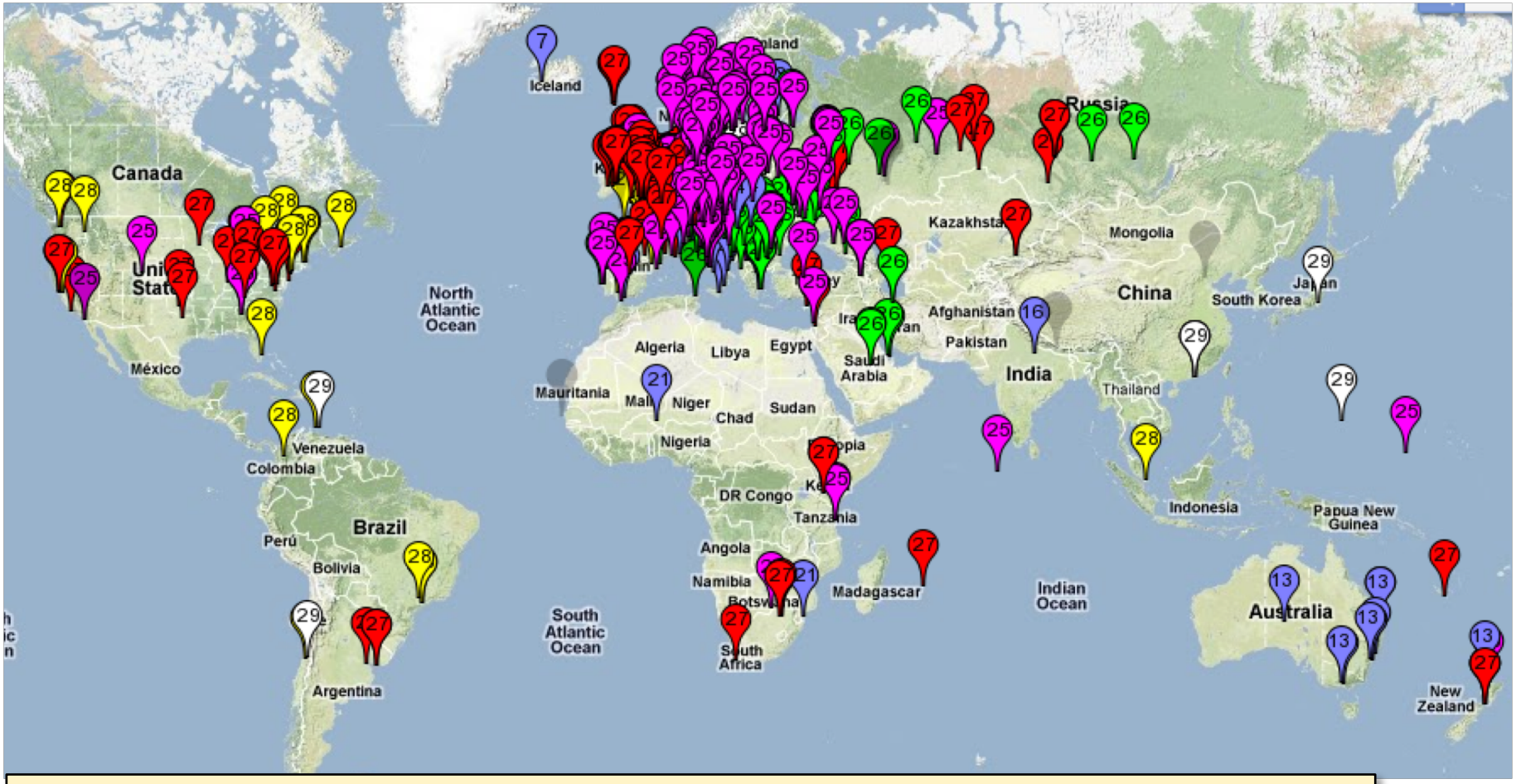
- I praksis blir denne informasjonen cachet på de fleste systemer
- DDoS-angrep kan svekke tjenesten
- Designfeil i infrastrukturen kan skape problemer

ICANN - rottjenere



from: https://upload.wikimedia.org/wikipedia/commons/e/ee/Root-current.svg

Applikasjonslaget



k-root (Europe) er en ”anycast” rotnode
Kart over enheter og hvilken instans av k-root den ser, basert på plassering

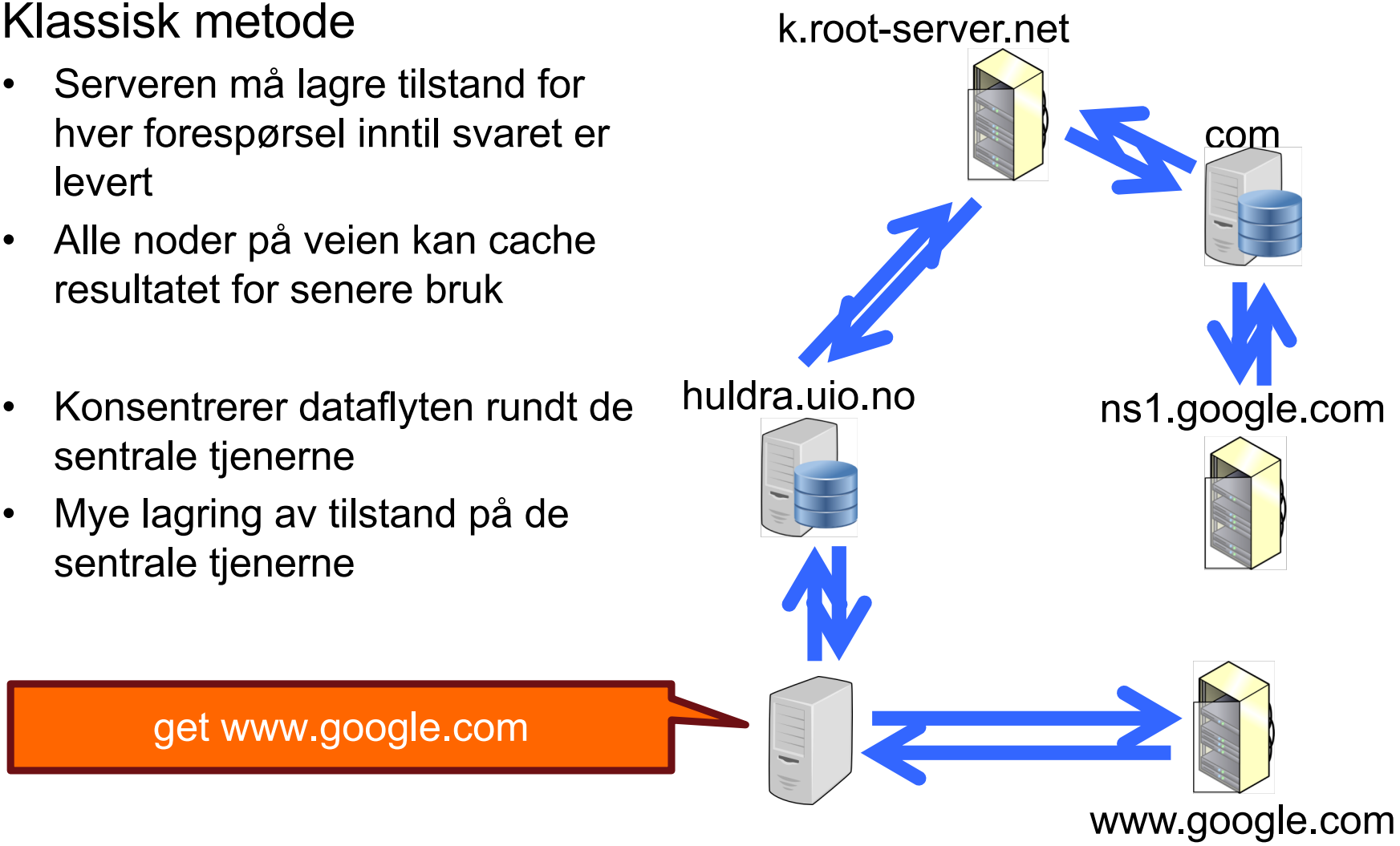
frahttps://labs.ripe.net/Members/kistel/dns-measurements-with-ripe-atlas-data

Rekursivt oppslag i DNS

Klassisk metode

- Serveren må lagre tilstand for hver forespørsel inntil svaret er levert
- Alle noder på veien kan cache resultatet for senere bruk

- Konsentrerer dataflyten rundt de sentrale tjenerne
- Mye lagring av tilstand på de sentrale tjenerne

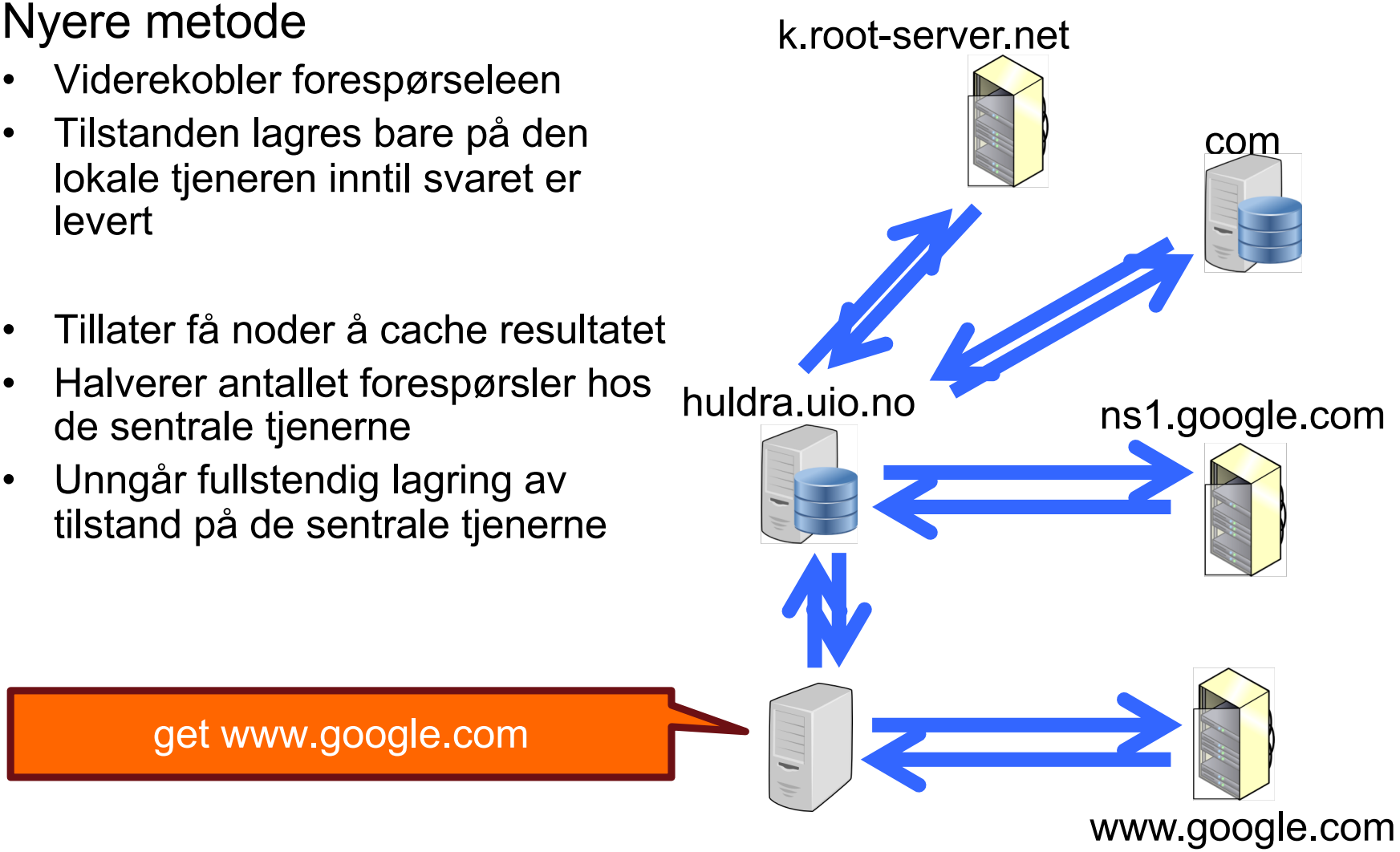


Iterert oppslag i DNS

Nyere metode

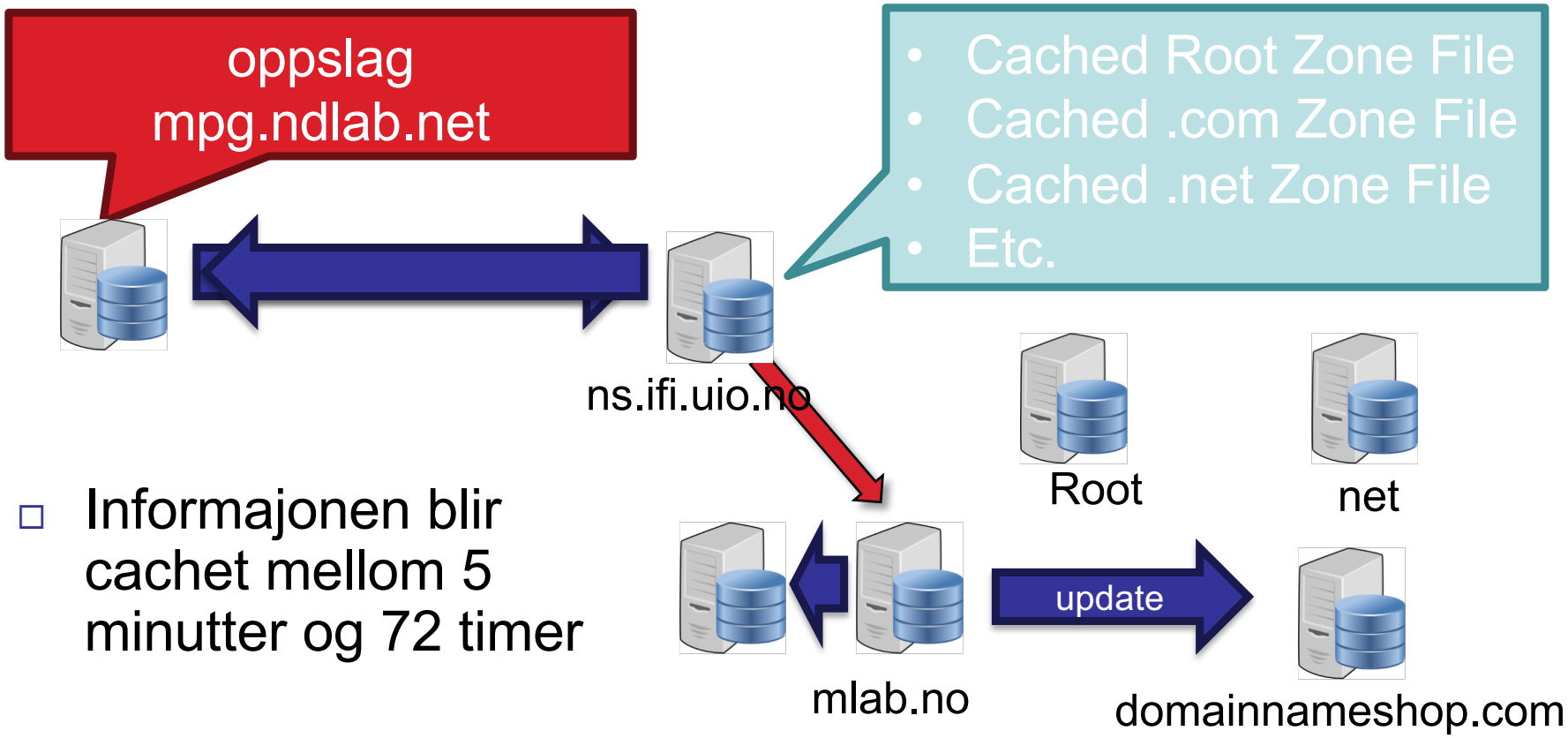
- Viderekobler forespørseleen
- Tilstanden lagres bare på den lokale tjeneren inntil svaret er levert

- Tillater få noder å cache resultatet
- Halverer antallet forespørsler hos de sentrale tjenerne
- Unngår fullstendig lagring av tilstand på de sentrale tjenerne



Caching vs. oppdaterte data

- Caching reduserer forsinkelsen for et DNS-oppslag
- Caching reduserer lasten på DNS-tjenerne
- Caching forsinker oppdateringer

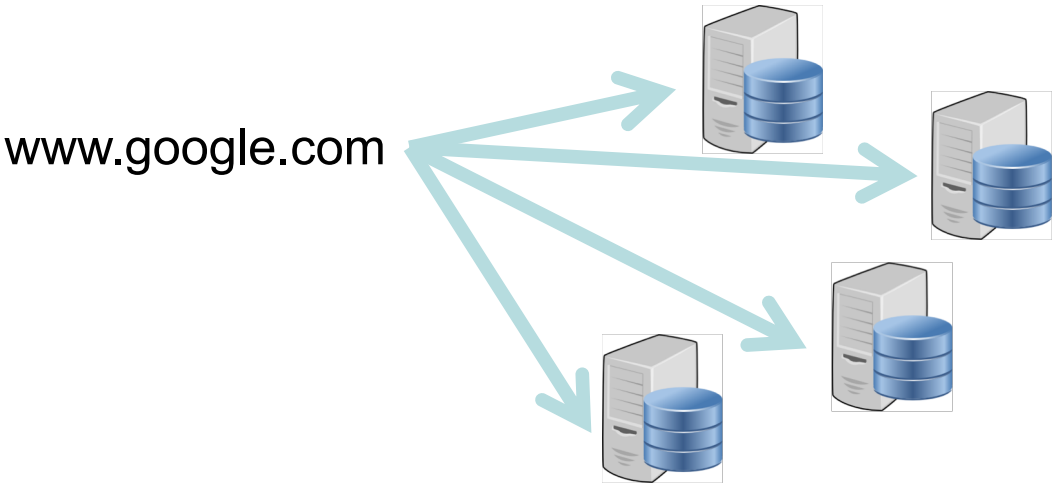


- Informajonen blir cachet mellom 5 minutter og 72 timer

Alias og lastbalansering

Én maskin kan ha mange alias

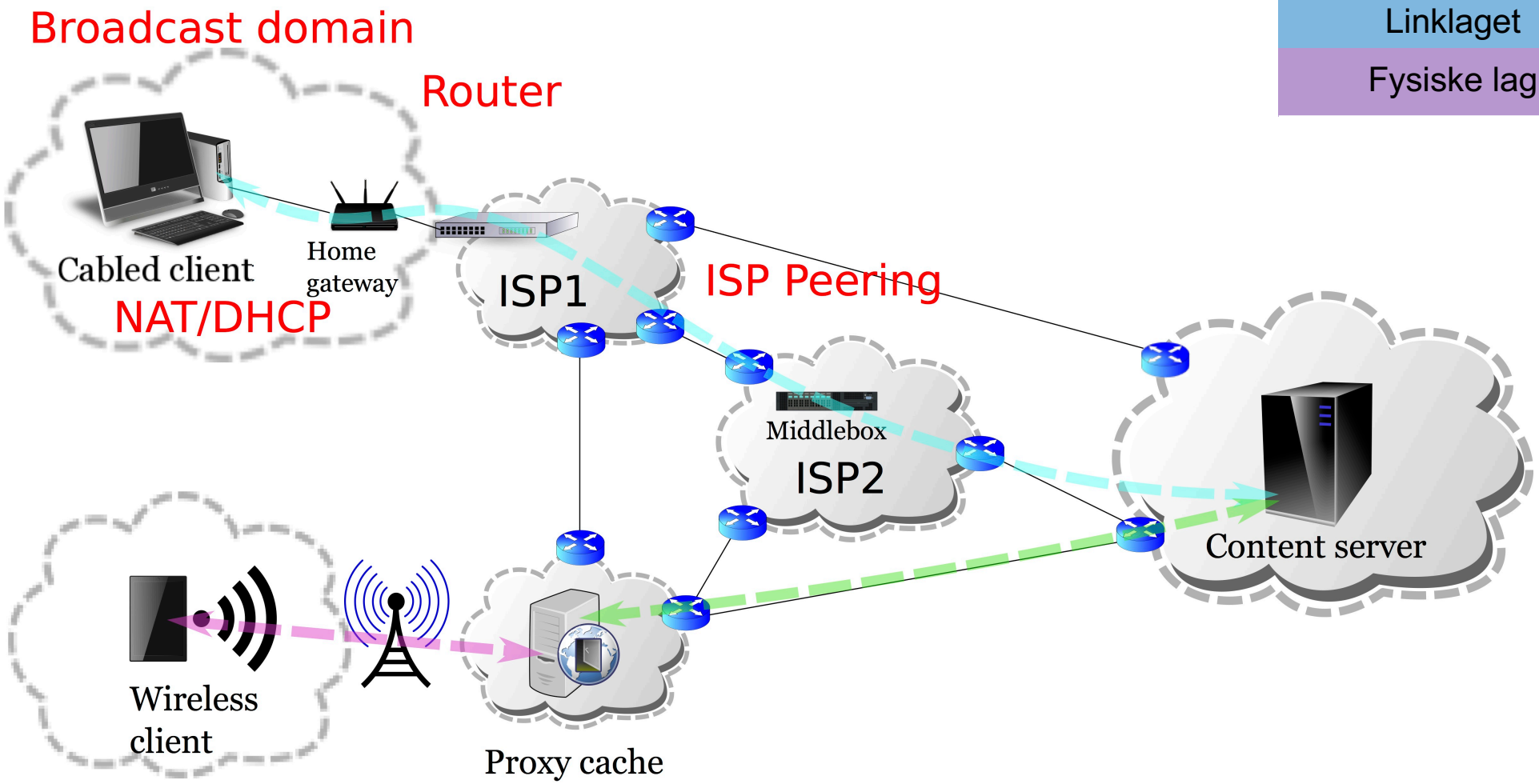
Ett domene kan kobles til mange maskiner



Eksempler:
k.root-server.net
og
login.ifi.uio.no

Applikasjonslaget
Transportlaget
IP/Internett
Linklaget
Fysiske lag

En datapakkes vei gjennom nettet



Når du åpner facebook…

Applikasjonslaget

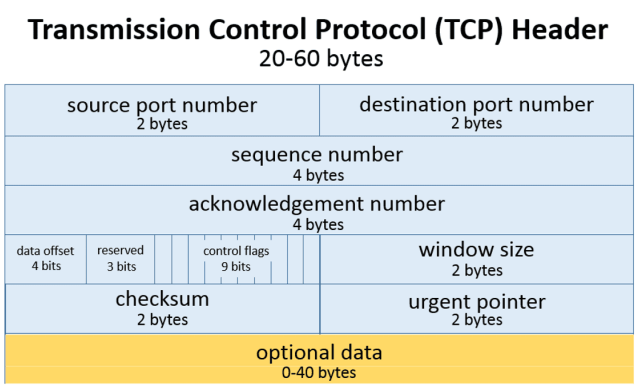
Transportlaget

IP/Internett

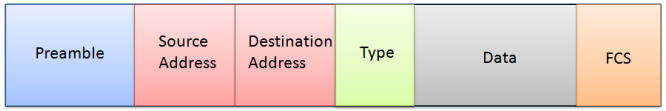
Linklaget

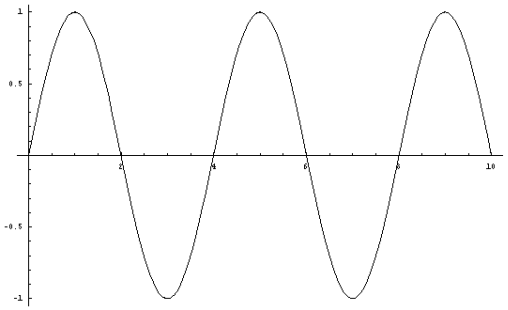
Fysiske lag

DNS



ARP





Nyttige nettverksverktøy

Verktøy	
ping	Måler forsinkelsen mellom din maskin og en annen på internett
traceroute	Gir informasjon om hvert hopp en datapakke er innom på vei til målet.
tracepath	Samme som traceroute, men trenger ikke superusertilgang
netstat	Gir informasjon om nettverksforbindelser på din maskin
dig	Gjøre DNS-oppslag og se hva som returneres.
whois	Slå opp hvem som eier/disponerer en IP-adresse
tcpdump / wireshark	Logge pakker som går inn og ut av maskinen, samt utføre analyse

Oblig 3

- 3 oppgaver på forskjellige temaer
- Oppgave 1: forstå metadata fra de forskjellige lagene
  - Dere får utlevert metadata fra 5 forskjellige pakker
  - Dere må si noe om hver pakke og konteksten rundt dem
- Oppgave 2: Subnetting og tjenester som kreves i et LAN
  - Dere må «designe» et kundenettverk for et bakeri.
- Oppgave 3: Kapasitet og forsinkelse
  - Litt målinger på Internett og beskrivelse av effektene dere observerer

Forsinkelse (latency) vs. kapasitet

Quiz (Yeah!)

https://play.kahoot.it/#/k/97bf1699-52a1-41ae-afb3-58db115713de

Ekstramateriale:

- Wikipedia (gode artikler på disse temaene):
  - Internet Protocol Suite – med link til de andre komponentene: https://en.wikipedia.org/wiki/Internet_protocol_suite
- Bøker og artikler:
  - Tanenbaum, Andrew S. "Computer Networks". Prentice Hall PTR
  - James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach