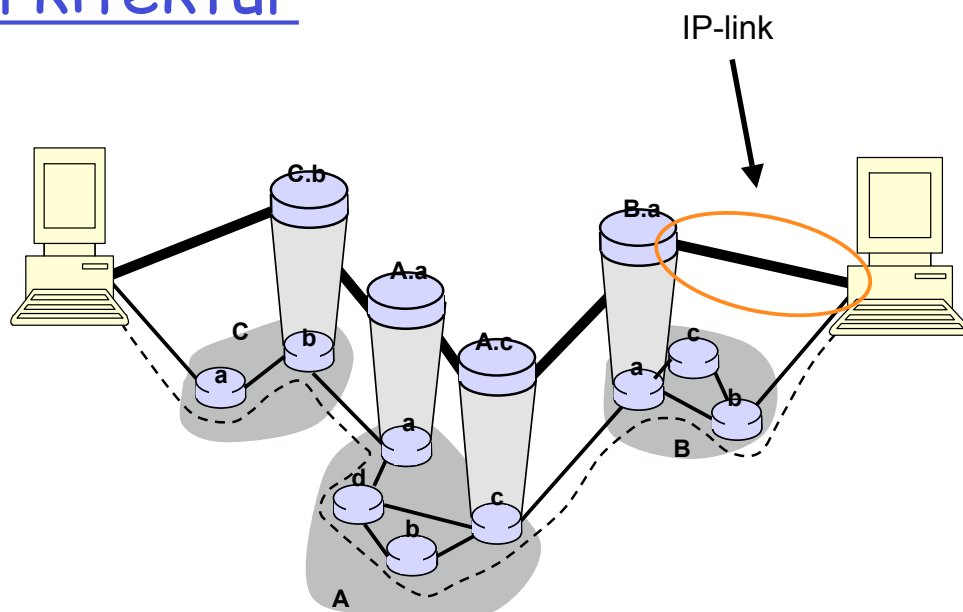


Linklaget

Olav Lysne

(med bidrag fra Stein Gjessing og Frank Eliassen)

Internettets Overlay Arkitektur



Link-typer

Tre typer av linker:

- (a) Punkt-til-punkt (enkel kabel)
- (b) Broadcast link (delt kabel eller annet medium; f. eks eldre Ethernet og trådløs)
- (c) Svitsjet (f.eks svitsjet Ethernet og ATM)

Link Lagets tjenester

□ *Framing og linkaksess:*

- Pakke data inn i en "ramme" (frame), og legge til hode og hale.
- Implementere kanalaksess dersom det er delt medium.
- 'fysiske adresser' blir brukt i rammeheaderene til å identifisere kilde og destinasjon når det er delt medium.

□ *Pålitelig levering:*

- Sjelden brukt i fiber optikk, co-axial kabel og noen varianter av twisted pair pga. lav feil-rate.
- Brukt på trådløslinker hvor målet er å redusere feil, og unngå ende til ende retransmisjoner.

Linklagets tjenester (mer)

□ **Flytkontroll:**

- Hastighetsavpassing mellom sender og mottaker.

□ **Feildeteksjon:**

- Feil kommer av støy og signalreduksjon
- Mottaker oppdager feil i mottatt ramme.
- Den signalerer for retransmisjon, eller den bare kaster rammen.

□ **Feilkorreksjon:**

- Mekanisme hvor mottakeren retter feilen uten å be om retransmisjon.

Feilfinning/feilretting

□ **Oppgaver:**

- 1. Finne feil
- 2. Rette feil
 - To alternativer til å rette feil:
 - A. Ha nok informasjon til å rette opp de mottatte dataene
 - B. Be om at dataene (rammen) blir sendt en gang til
 - (C. Gi blanke, det er ikke så farlig å miste litt data)
- Generelt prinsipp i informatikken:
Oppdag feilen så fort som mulig etter at den har oppstått !

Feil-deteksjon

- Bit-feil i rammer
 - behov for mekanismer som oppdager bit-feil
- Teknikker som ofte benyttes i datanett
 - Paritet - to-dimensjonal paritet
 - BISYNC ved ASCII overføring
 - Sjekksum
 - flere Internett-protokoller
 - Cyclic Redundancy Check (CRC)
 - svært utbredt

Paritet (tversum)

- Ett paritetsbit:
 - F.eks. 7 bit data, sendes som 8 bit
 - Like paritet dvs. et like antall enere i resultatet
 - Odde paritet dvs. et odde antall enere i resultatet
- Like paritet: 0110001 sendes som 01100011
- Odde paritet: 0110001 sendes som 01100010
- Mulig med flere paritetsbit
 - Generelt: Jo mer data til redundanse, jo flere feil oppdages.

Internett sjekksum algoritme

- Se på en melding som en sekvens av 16-biters heltall
 - Senderen adderer disse heltallene sammen ved bruk av 16-biters aritmetikk
 - Dette 16-biters tallet er sjekksummen
 - Mottaker utfører samme beregning og sammenligner resultatet med den mottatte sjekksum
 - Får mottaker feil resultat er det bitfeil enten i dataene eller i sjekksummen
- Benyttes ende til ende i Internett

CRC: Cyclic Redundancy Check

Generalisering av paritet
Punkter i et n-dimensjonalt rom

Kodeord

Data (med hode)	Sjekk/CRC
-----------------	-----------

Like paritet: Kodeordet delt på 2 skal ikke gi rest
CRC: Kodeordet delt på et tall, G , skal ikke gi rest

Dette tallet vi deler på kaller vi *Generatorpolynomet*
Deling foregår med modulo-2 regning, dvs ikke mente eller låning.

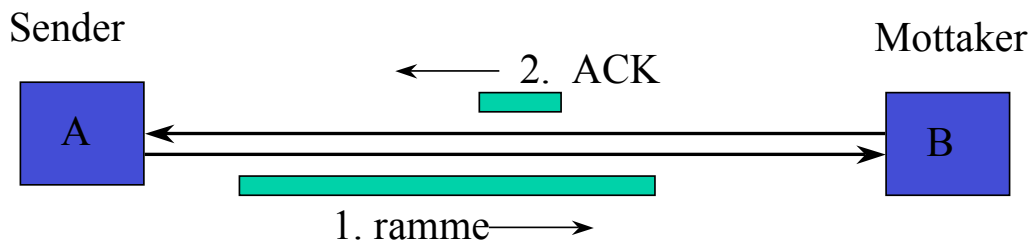
Pålitelig overføring

- ❑ rammer med feil CRC kastes
- ❑ Fint om vi kan rette opp feilen
- ❑ Hvis feilen ikke kan rettes opp, og vi trenger rammen, da
må den sendes en gang til !
- ❑ Også her er det en avveining:
Ende-til ende eller mellom noder ?
(Problemkomplekset med dobbelt/triplet (mm.) funksjonalitet)

Pålitelig overføring

- ❑ Når omsending av rammer er nødvendig:
- ❑ To fundamentale mekanismer
 - kvitteringer (engelsk: acknowledgements, ack)
 - tidsfrister (timeouts) vha. vekkeklokke (timer)
- ❑ Husk at også kvitteringer kan bli borte
- ❑ Ønsker vi at rammene skal komme frem i riktig rekkefølge?

Stop-and-Wait (stopp og vent)



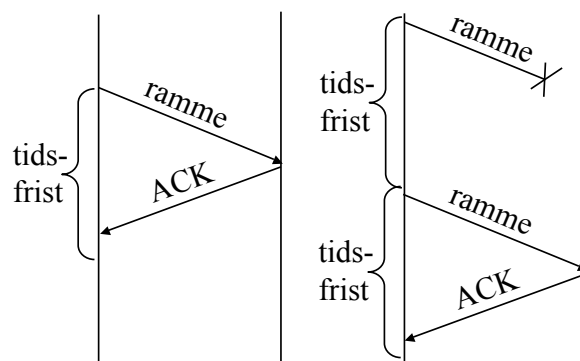
Mottaker sender ack tilbake når en ramme er mottatt, og først når sender mottar ack, sendes ny ramme. På denne måten blir ikke mottaker oversvømmet av rammer, og avsender vet at alle rammer er kommet trygt fram.

Men hva hvis rammer blir borte?

Stop-and-Wait (stopp og vent)

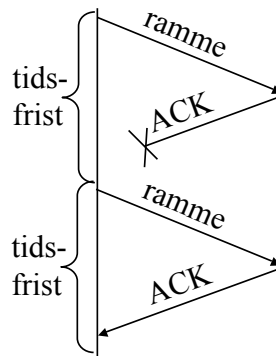
□ Grunnleggende algoritme:

- send én ramme og vent på kvittering (ACK ramme)
- dersom ACK ikke mottatt innen gitt tidsfrist, send rammen på nytt.



Stop-and-Wait

- Problem 1 med grunnleggende algoritme:
 - Men kanskje det var ACK som ble borte
 - Vi må kunne sende den samme rammen på nytt, selv om den allerede er kommet riktig frem

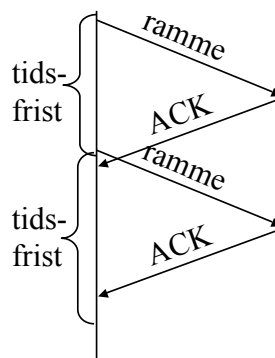


[[simula](#) . research laboratory]

Linklaget 15

Stop-and-Wait

- Problem 2 med grunnleggende algoritme:
 - Kanskje vi sendte rammen omigjen for tidlig
 - Vi må godta at ACK kommer for sent

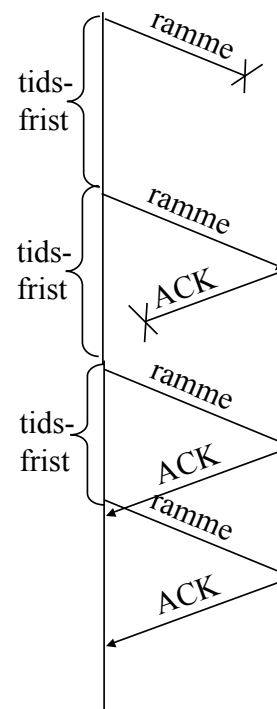


[[simula](#) . research laboratory]

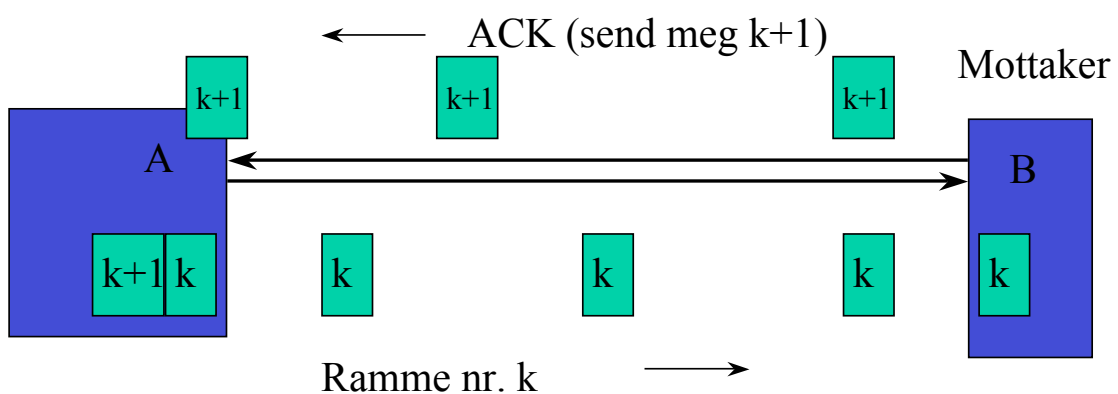
Linklaget 16

Stop-and-Wait

- Må sende rammen på nytt og på nytt helt til ACK kommer tilbake



Løsning: sekvensnummer



Neste bilde:
Det er nok med en en-bit teller (0 og 1)
 $k, k+1$ regnes da ut modulo 2.
(En buffers "Sliding window" protokoll)

Sekvensnummer som 0 og 1

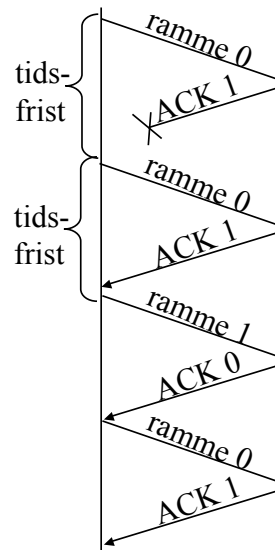
□ 0 og 1 som sekvensnummere

- En bit er nok når vi har én ramme ad gangen

ACK 1: Send ramme med odde sekvensnummer

ACK 0: Send ramme med like sekvensnummer

- Altså:
ramme 0, 1, 2, 3, 4, 5, ... sendes som
ramme 0, 1, 0, 1, 0, 1, ...



0 - 1 sekvensnummer

■ Går dette bra?

■ Ja, fordi:

- like ramme (0) sendes ut
- ignorerer (gamle) ACK 0
- resender like ramme (0)

I det ACK 1 kommer:

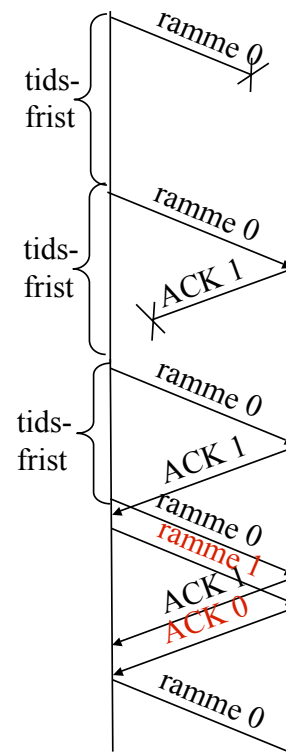
- odde ramme (1) sendes ut
- ignorerer (gamle) ACK 1
- resender odde ramme (1)

I det ACK 0 kommer:

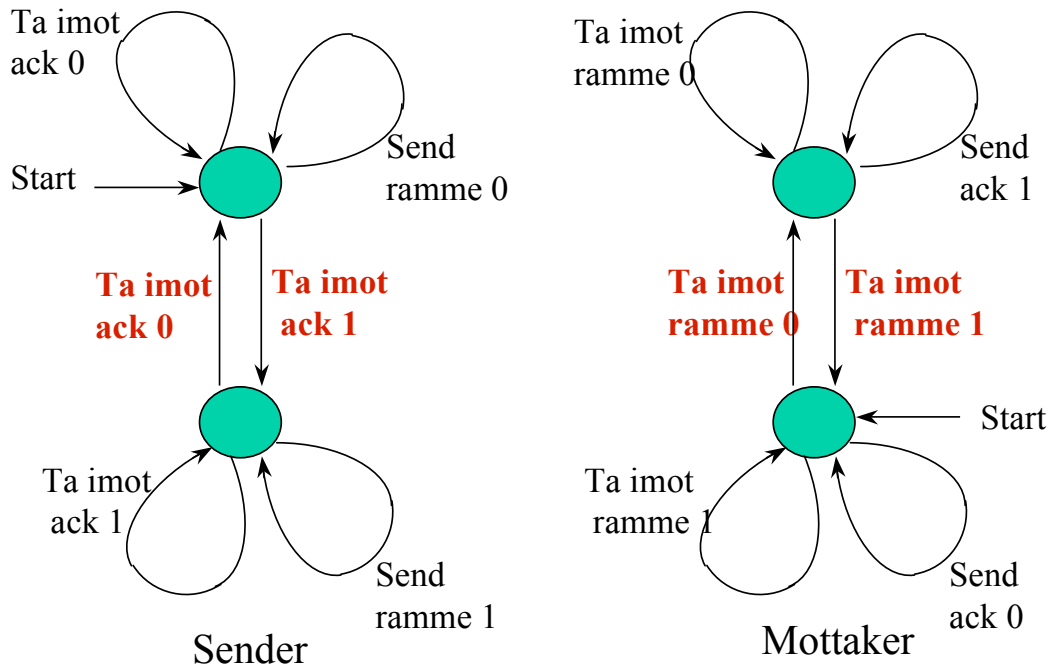
- like ramme (0) sendes ut
- ignorerer (gamle) ACK 0
- resender like ramme (0)

I det ACK 1 kommer:

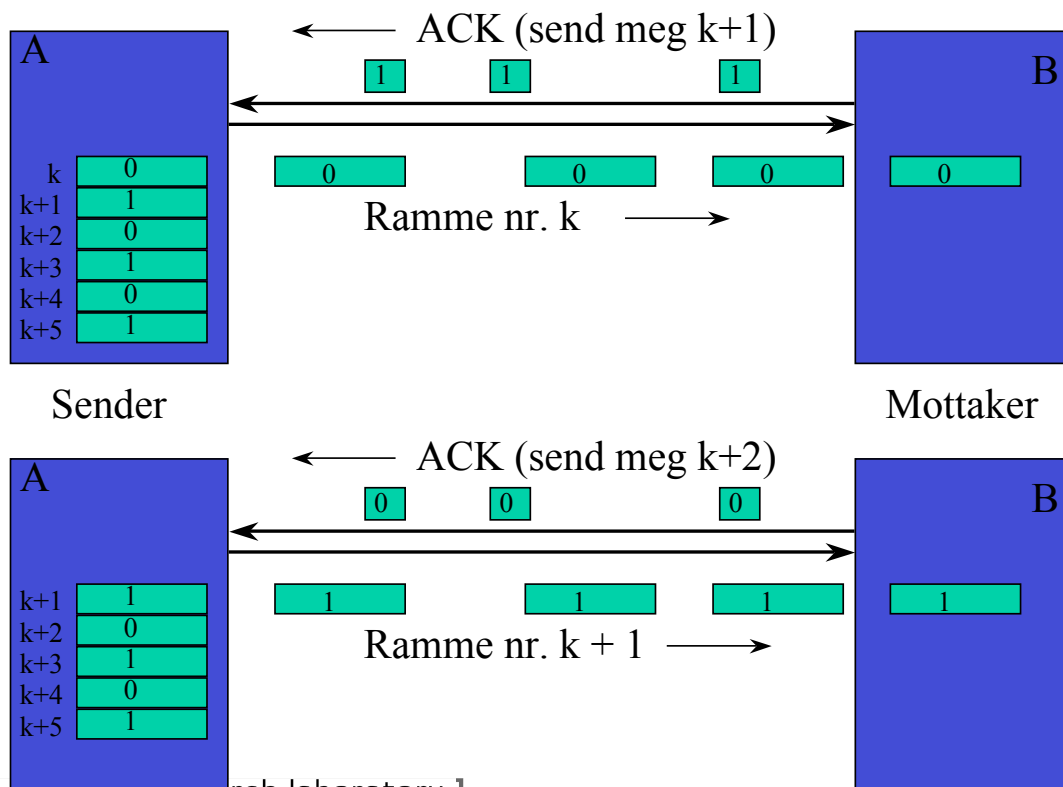
osv.



Tilstandsmaskin for en bit protokoll

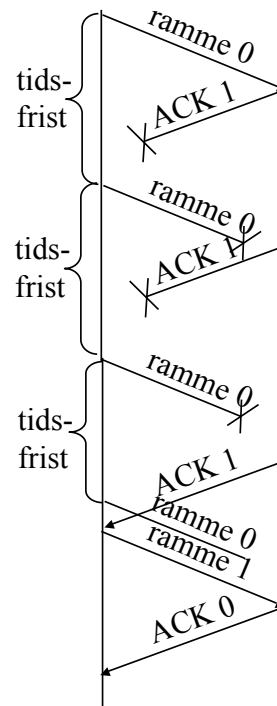


Et bit protokoll



Stop-and-Wait

- Det gjør ikke noe om mottaker gjentar en ACK, men det er ikke vanlig, dvs. at det er bare vanlig å sende ACK når en ramme ankommer. Ved mye rammetap kan det være gunstig å gjenta ACK selv om det ikke kommer en ny ramme



Stop-and-Wait

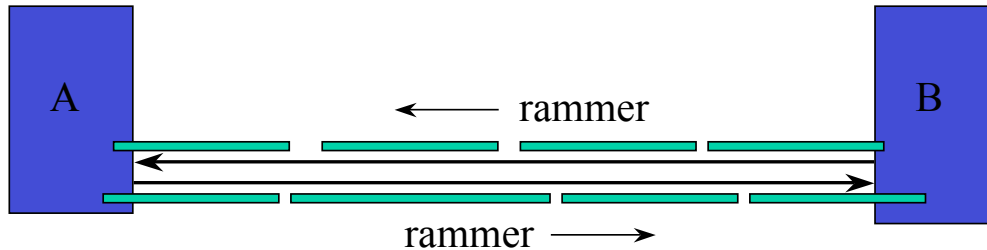
- Grunnleggende svakhet:
 - utnytter linjekapasiteten dårlig
 - senderen kan bare ha én utestående ramme til enhver tid
- Eksempel:
 - 1.5Mbps link x 45ms RTT = 67.5Kb (8KB)
 - dvs. 8KB data kan sendes før første ack kan ventes tilbake
 - anta rammestørrelse 1KB
 - stop-and-wait bruker ca. 1/8 av linjekapasiteten (om ingen feil)
 - Mål: senderen må kunne sende opp til 8 rammer før den må vente på en ACK
 - moralen er: "fyll opp røret"



Fyll opp røret

Utnytte linjen bedre.

Sender flere rammer rett etter hverandre:



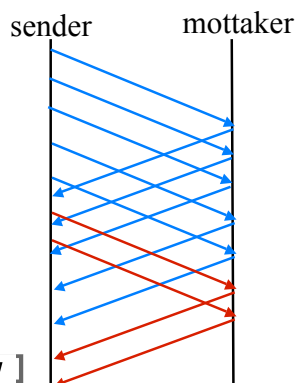
Putte ACK/NAK på ryggen til meldinger som går den andre veien ("piggyback")

Glidende vindu

□ Idé:

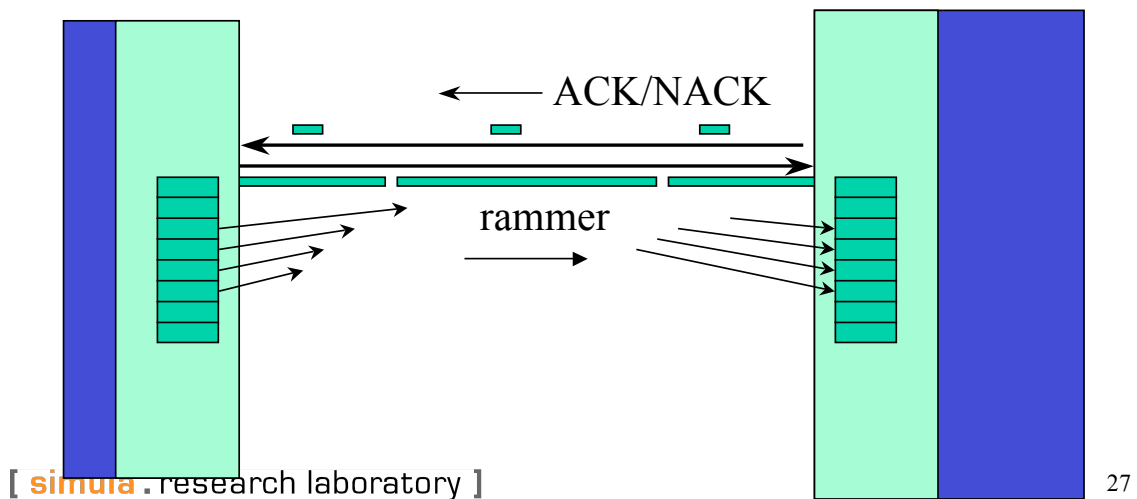
- Tillat senderen å sende flere rammer før den mottar ACK for derved å holde "røret fullt".
- Det må være en øvre grense på antall rammer som kan være utestående (som det ikke er mottat ACK for).

Eksempel:
maks. 5
utestående
rammer



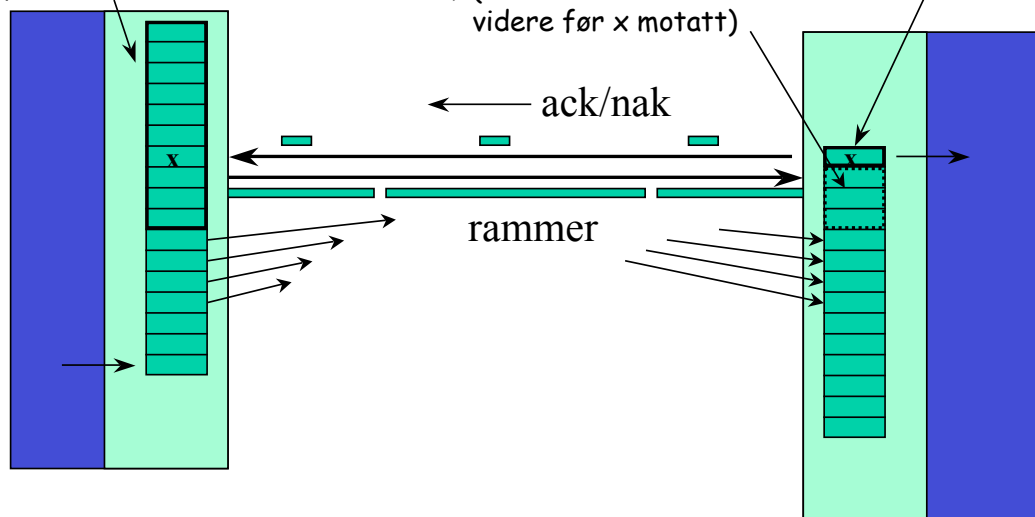
Glidende vindu

Flere bufre hos sender og flere bufre hos mottaker, mange rammer med forskjellige nummer og mange ack/nack med forskjellige nummer under veis hele tiden.



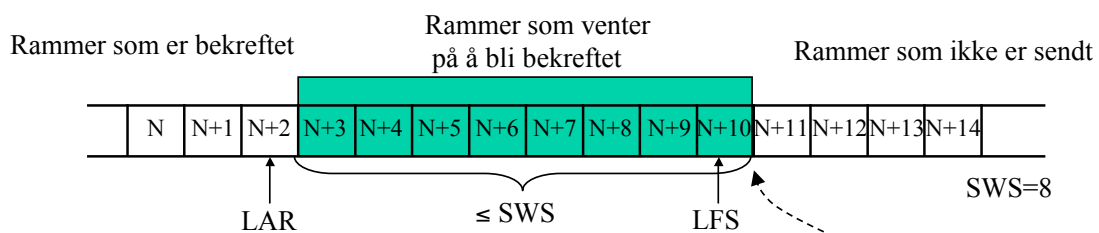
Glidende vindu

Sendt men ikke fått ack, må kanskje sendes på nytt (vil bli resendt om ack aldri mottas) Disse tre er motatt og kvittert (ack sendt) Ikke motatt (men kan ikke sendes videre før x motatt)



Glidende vindu: sender

- Tilordner sekvensnummer til hver ramme (SeqNum)
- Vedlikeholder tre tilstandsvariable
 - send window size (SWS)
 - last acknowledgment received (LAR)
 - last frame sent (LFS)
- Vedlikeholder invariant: $LFS - LAR \leq SWS$



Når ACK mottas, økes LAR, og derved kan ny ramme sendes slik at LFS økes.

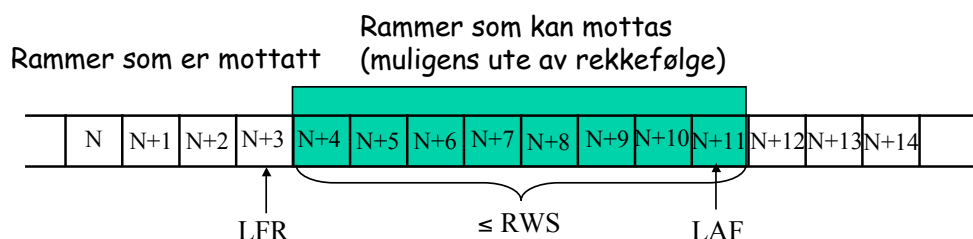
Buffer for opptil SWS rammer, dvs SWS rammer i "røret" samtidig

[[simula](#) . research laboratory]

Linklaget 29

Glidende vindu: mottaker

- Vedlikeholder tre tilstandsvariable
 - receive window size (RWS)
 - largest acceptable frame (LAF)
 - last frame received (LFR) (med alle "mindre" rammer også motatt)
- Vedlikeholder invariant: $LAF - LFR \leq RWS$

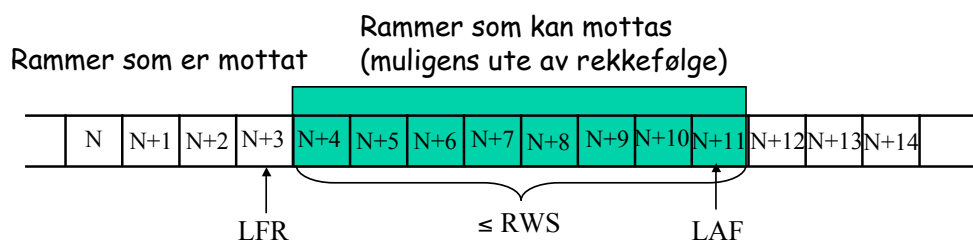


[[simula](#) . research laboratory]

Linklaget 30

Glidende vindu: mottaker

- Kumulativ kvittering ("go back n" protokoll), dvs. vi ack-er ikke nye rammer hvis det er hull i sekvensen av mottatte rammer
 - **if** $LFR < MottatRamme.SeqNum < LAF$ **then**
 - ta-imot-rammen-og-legg-den-på-plass;
 - beregn-nye-grenser ($MottatRamme.SeqNum$); // se neste lysark
 - else**
 - kast rammen;
 - send ACK ($LFR + 1$)



[[simula](#) . research laboratory]

Linklaget 31

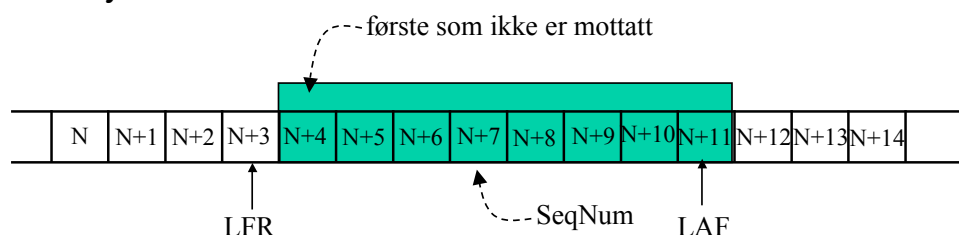
Glidende vindu: mottaker

- Invariant: $LFR + 1$ er første ramme som ikke er mottatt
- beregn-nye-grenser(seqNo)

```

• if seqNo = LFR + 1
  {   beregn ny LFR og LAF:
      for (i= LFR + 1; ramme i er mottatt; i++) { } ;
      LFR = i-1;
      LAF = LFR + RWS;
  }

```



[[simula](#) . research laboratory]

Linklaget 32

Glidende vindu: mottaker

- Varianter til "go back n" protokoll
 - Negativ kvittering (NAK)
 - mottaker sender NAK på rammer som savnes
 - Selektiv kvittering (SAK)
 - mottaker sender ACK på nøyaktig de rammer som mottas
 - disse behøver da ikke re-sendes
 - Både SAK og NAK øker kompleksiteten til implementasjonen, men kan potensielt bedre utnyttelsen av linjen

Glidende vindu: endelige sekvensnummer

- I praksis representeres sekvensnummer med et endelig antall biter.
- n biters sekvensnummer => intervall sekv. nr. = $(0..2^n-1)$
- I HDLC: $n = 3$, dvs. sekvensnummerintervall $(0..7)$
 - => sekvensnummer må gjenbrukes

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--

Problem

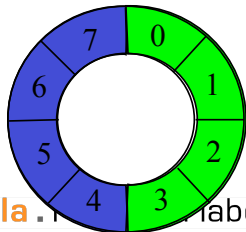
skille mellom ulike inkarnasjoner av samme numre

Minimum krav:

sekvensnummerintervallet må være større enn maks. antall utestående rammer

Glidende vindu: endelige sekvensnummer

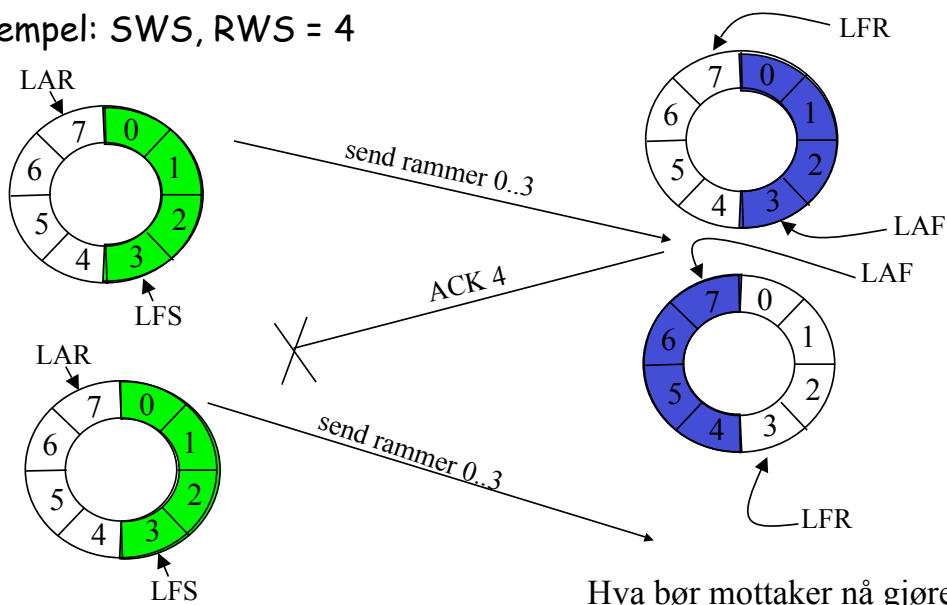
- **SWS \leq MaxSeqNum+1** er ikke tilstrekkelig
 - anta 3 biters SeqNum felt (0..7)
 - SWS=RWS=8
 - senderen transmitterer rammene 0..6
 - mottas uten feil, men ACK går tapt
 - senderens tidsfrist utløper, rammene 0..6 sendes på nytt
 - mottaker forventer 7, 0..5, men mottar andre inkarnasjon av 0..5
- **SWS, RWS \leq (MaxSeqNum+1) / 2** er riktig regel



Hindrer overlapp mellom
nedre kant av sendervinduet og
øvre kant av mottakervinduet.

Glidende vindu: endelige sekvensnummer

- Eksempel: SWS, RWS = 4



Hva bør mottaker nå gjøre?

ETHERNET

CSMA/CD (Ethernet)

□ Historie

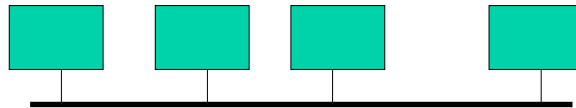
- Utviklet på Xerox PARC (Palo Alto Research Center) ca. 1975
- Standardisert av Xerox, DEC og Intel i 1978
- Videre standardisert av IEEE (802.3)

- Utviklet fra Aloha pakke-radio protokoll (Univ. Hawaii)
 - Aloha protokollen sender en pakke (ut i "eteren") og håper at det ikke blir noen kollisjon (delt "eter")
 - Hvis ACK tilbake: OK
 - Ikke ACK tilbake/NAK:
 - antagelig kollisjon - send på nytt
 - Ble brukt til å sende data mellom øyene på Hawaii

CSMA/CD (Ethernet)

□ CSMA/CD

- Carrier Sense (noder kan skille mellom ledig og opptatt linje)
- Multiple Access (flere noder aksesserer samme medium)
- Collision Detection (en node kan lytte mens den sender)



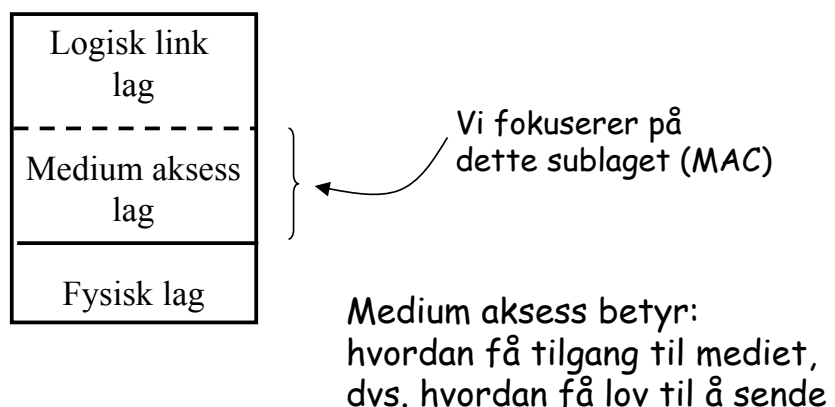
Båndbredde: 10Mbps, 100Mbps, 1Gbps
(Virkemåten til 100Mbps og 1Gbps kommer senere)

Problem:

Distribuert algoritme som gir rettferdig adgang til et felles medium (Ethernet-kabelen)

CSMA/CD (Ethernet)

□ Relasjon til OSI modellen



CSMA/CD (Ethernet)

□ Virkemåte (oversikt)

- Som en bus: bare én node kan sende om gangen
- I motsetning til intern databus:
 - I en databus sørger egen logikk for eksklusiv adgang til bussen
- Ethernet:
 - ingen kontakt (annen enn Ethernetkabelen selv) mellom senderne

1. Lytt på nettet (kabelen)

2. Hvis ledig: send ut rammen

3. Fortsett å lytte på nettet (mens sendig pågår)

Hvis det som høres er det samme som sendes: OK

Hvis det som høres ikke er lik det som sendes:

Kollisjon (noen andre prøvde å sende rett etter meg)

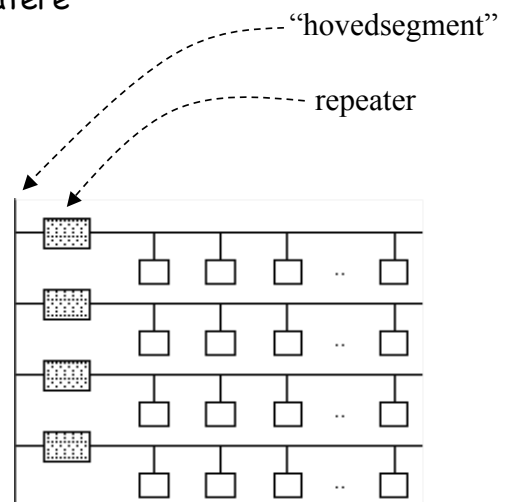
Prøv på nytt senere.

CSMA/CD (Ethernet)

□ Klassisk Ethernet

- Maksimum 500 meter per segment (50ohm coaxial kabel)
- Segmenter kobles sammen med repeater
- Ikke mer enn to repeater mellom hver node (⇒ max 1500 meter)
- Max 1024 noder
- Kringkastingsnett ("alle hører alt")
- Kalles også 10Base5 ("tykketer")

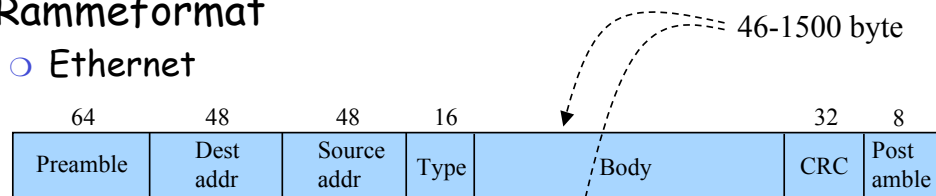
- Begrensning på lengde for å være sikker på at sender hører kollisjonen før han er ferdig med å sende (T/R tid)



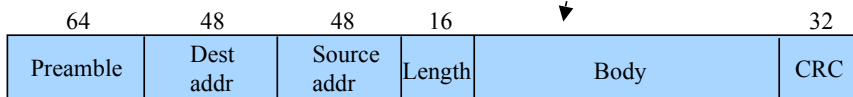
CSMA/CD (Ethernet)

□ Rammeformat

○ Ethernet



IEEE 802.3



Adresser. Kalles MAC-adresser

Entydig 48-biters adresse tilordnes hver adapter ("brent" inn i adapteren)

Eksempel: 8:0:2b:e4:b1:2

Kringkastingsadresse: ff:ff:ff:ff:ff:ff (kun 1-ere)

Multicast: første bit er 1

[[simula](#) . research laboratory]

Linklaget 43

CSMA/CD (Ethernet)

□ Mottaker algoritme

○ En Ethernet adapter mottar alle rammer og aksepterer:

- rammer adressert til dens unicast (én til én) adresse
- rammer adressert til kringkastingsadressen
- rammer adressert til enhver multicast adresse den er blitt programmert til å akseptere
- alle rammer når den kjører i åpen ("promiscuous") modus

[[simula](#) . research laboratory]

Linklaget 44

CSMA/CD (Ethernet)

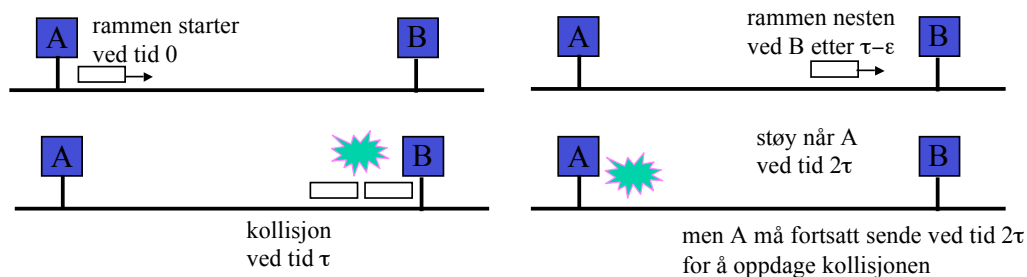
- Sender algoritme:
 - Hvis linjen er ledig
 - send umiddelbart
 - maks meldingsstørrelse (data) er 1500 byte
 - Må vente 51µs mellom etterfølgende rammer
 - gir andre stasjoner en sjanse å sende sine rammer
 - Hvis linjen er opptatt
 - vent til den blir ledig og send umiddelbart
 - kalles 1-persistent
 - sender med sannsynlighet 1 når linjen er ledig
 - spesialtilfelle av p-persistent ($0 < p < 1$)

CSMA/CD (Ethernet)

- Sender algoritme (forts.):
 - Hvis kollisjon oppstår:
 - send ytterlige 512 biter ("jam"), stopp så å sende rammen
 - minimum ramme er 64 byte (header + 46 byte data)
 - Vent en tid og forsøk igjen
 - 1. gang: uniformt fordelt mellom 0 og 51.2µs
 - 2. gang: uniformt fordelt mellom 0 og 102.4µs
 - 3. gang: uniformt fordelt mellom 0 og 204.8µs
 - n-te gang: uniformt fordelt mellom 0 og $2^{(n-1)} * 51.2µs$
 - kalles "exponential backoff"
 - gir opp etter flere forsøk (vanligvis 16)

CSMA/CD (Ethernet)

- Hvorfor minst 64 byte ramme og 51.2 μ s ?
 - T/R-tid kan være opptil 51.2 μ s
 - Nok tid til å sende 512 bit over 10Mbps Ethernet
 - Sikrer at alle adaptere oppdager kollisjonen (inkl. senderen!)
 - Ventetid mellom etterfølgende rammer fra samme adapter på minimum 51.2 μ s (jfr. over)
 - sikrer at kabelen er "tom overalt" før neste forsøk



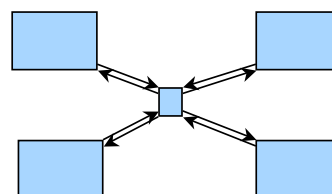
[[simula](#) . research laboratory]

Linklaget 47

CSMA/CD (Ethernet)

- Ethernet i praksis
 - 10 - 200 noder (ikke 1024)
 - Lengden som regel en tiendedel av maks.
 - (RTT (T/R-tid) nærmere 5μ enn 51μ)
 - Bør ikke ha større last enn 30%
 - ellers går for mye tid bort i kollisjoner
 - Best med store rammer
 - Stjernenett brukes mer og mer
 - svitsjede Ethernet med en stasjon per segment

100 Mbit og 1Gbit "Ethernet"



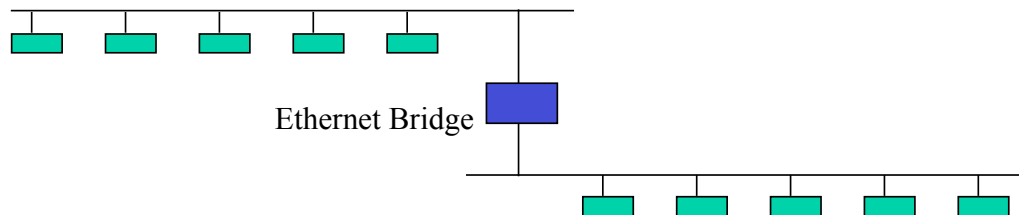
[[simula](#) . research laboratory]

Linklaget 48

3.2 Lokalnett-svitsjing

Bridge (norsk: Bro) brukes generelt om en enhet som binder sammen to nett, gjerne med forskjellige protokoller.

- Hvordan forbinde flere Ethernet
- Kalles lokalnett-svitsjing (LAN-switching)
- En Ethernet svitsj er gjerne en avansert bro



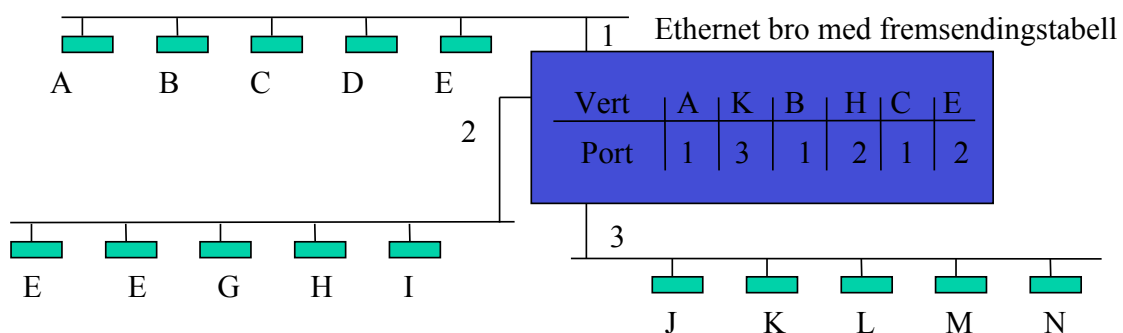
Broen kan ta imot alle rammer og sende dem videre på det andre nettet (promiskuøs modus)
Men er dette lurt?

[[simula](#) . research laboratory]

Linklaget 49

Lærenemme broer

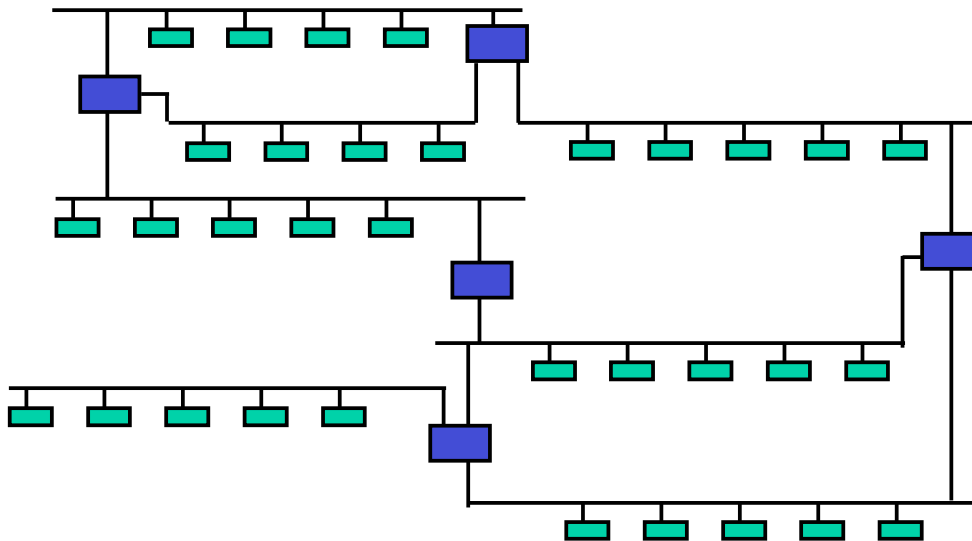
- For alle rammer som kommer inn, noter *avsenderadresse*, *A*, og innkommende link, *L*.
- Når en ramme kommer inn på en annen link med *mottakeradresse A*, send den ut på link *L*.
- Hvis mottakeradressen ikke finnes i tabellen, kringkastes rammen



[[simula](#) . research laboratory]

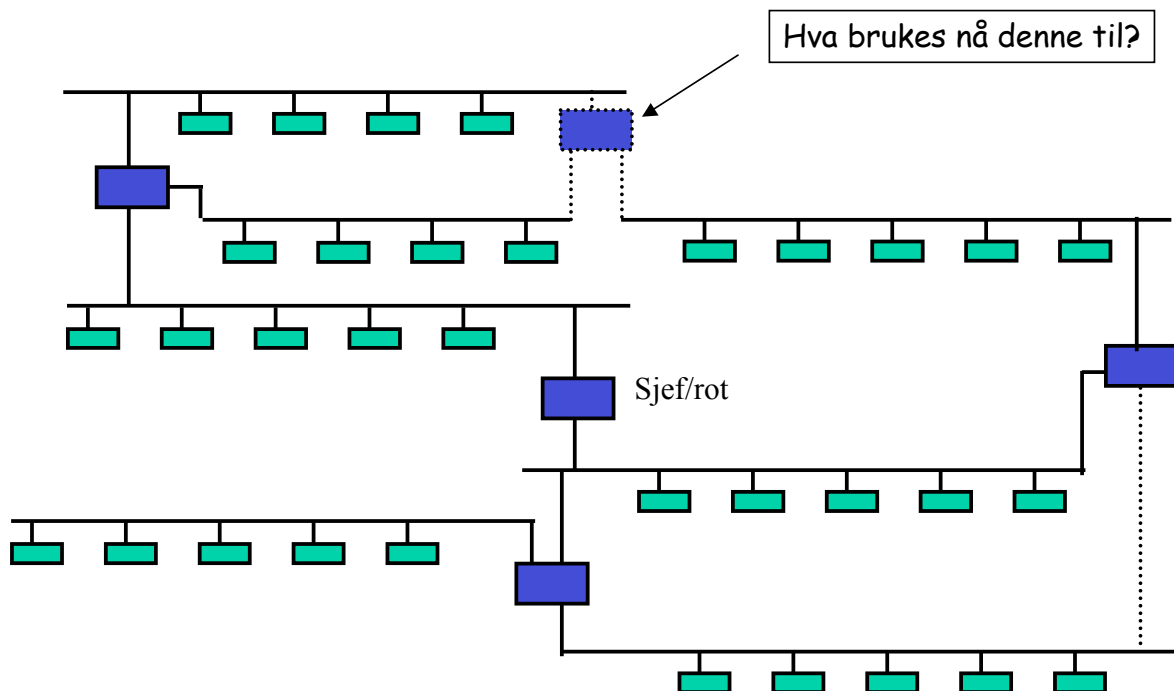
Linklaget 50

Løkker i lokalnett



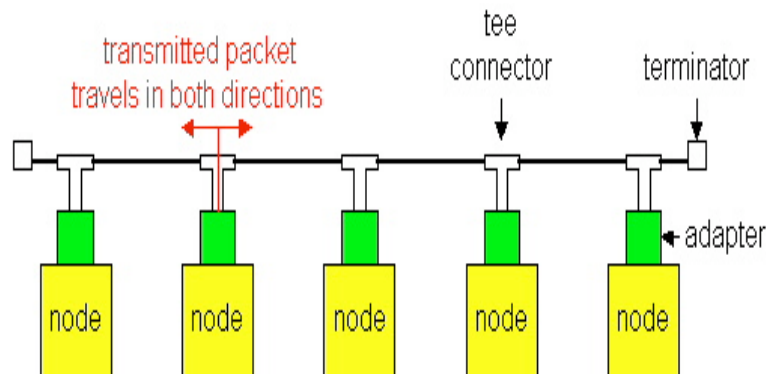
Da blir jo rammer videresendt i det uendelige...

En valgt sjef lager et spenntre



Ethernet Teknologier: 10Base2

- ❑ 10==10Mbps; 2==under 200 meter maks kabellengde i et segment.
- ❑ Bruker tynn koaksialkabel i en bussteknologi.
- ❑ Repeatere blir brukt til å koble sammen multiple segmenter (opp til 5).



[[simula](#) . research laboratory]

Linklaget 53

10BaseT og 100BaseT

- ❑ 10/100 Mbps rate; sistnevnte kalles "fast Ethernet"
- ❑ T står for Twisted Pair
- ❑ Nodene er koblet til en Hub med twisted pair, altså i "stjernetopologi"
- ❑ CSMA/CD implementert i Hub'en.

[[simula](#) . research laboratory]

Linklaget 54

10BaseT og 100BaseT (mer)

- ❑ Max distanse fra node til Hub er 100 meter
- ❑ Hub kan disconnekte en "pratmaker adapter"; 10base2 ville slutte å virke dersom ett adapter "snakket" hele tiden.
- ❑ Hub kan samle informasjon og statistikk til LAN administratoren.

Gbit Ethernet

- ❑ Bruker standard Ethernet rammeformat
- ❑ Tillater punkt-til-punkt linker og delte broadcast-kanaler.
- ❑ CSMA/CD brukes i delt modus; kort avstand mellom noder for effektivitet.
- ❑ Full-Duplex ved 1 Gbps for punkt-til-punkt linker