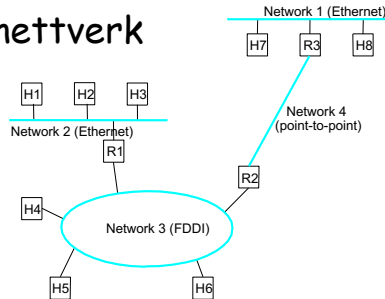
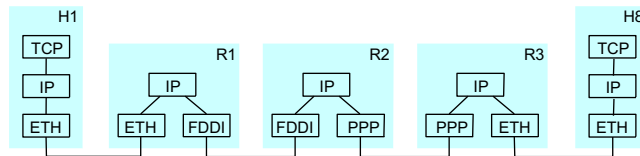


# IP Internet

## □ Sammensetning av nettverk

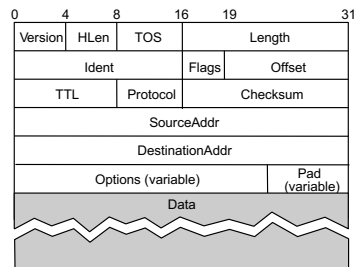


## □ Protokollstack



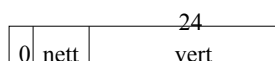
# Tjenestemodell

- Forbindelsesløs (datagram-basert)
- Best-effort levering (upålitelig)
  - pakketap
  - Pakker leveres i feil rekkefølge
  - Duplikate pakker leveres
  - Pakker kan ha lang forsinkelse
- Datagram format



## Globale adresser

- formatet; 32 bit (4 oktetter):
- uten struktur ⇒ problematisk ruting
  - globalt unike adresser
  - hierarkisk: nettverks-id + vert-id



Klasse A



Klasse B



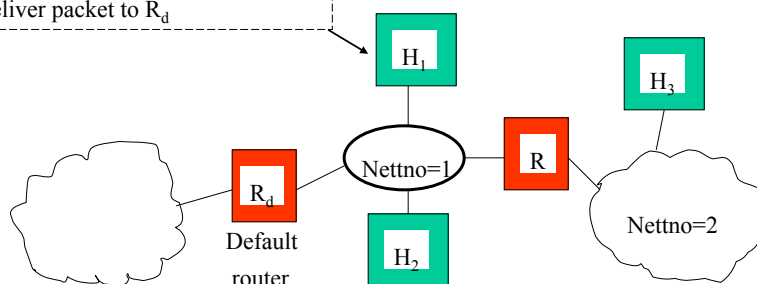
Klasse C

Dot notasjon:

129.240.68.88

## Fremsending

```
if (Destin-nettno=my-nettno) then
  deliver directly
else
  if (Destin-nettno in my route-table) then
    deliver to next-hop router
  else deliver packet to Rd
```



## Fremsending

- hver vertsmaskin inneholder en "ruter-halvpart"
- hvordan fremsende pakker til riktig mottaker?
  - ekstrahere IP-adressen til mottaker (ruter)
  - få tak i den tilsvarende lokale nett-adressen
  - pakke inn IP-pakken i lokal nett-pakke og sende

## IP til lokal nett-adresse mapping

- Address Resolution Protocol (ARP)
  - Request: (IP-addr; ??)
  - Response: (IP-addr; MAC-addr)
- ARP-tabell
  - sammenheng IP-addr og Ethernet-addr
  - permanent (satt manuelt)
  - temporær (med timeout) (satt av ARP protokollen)

## ARP tabellen

- timeout i løpet av ca 10 min (uten oppfrisking)
- oppdater tabellen med avsender hvis du er mottaker
- oppdater tabellen om et innslag i tabellen kan ha skiftet MAC adresse

## Skalerbarhet

IP "skjuler" maskiner i adressehierarkiet, men...

- Ineffektiv bruk av adresserommet
  - klasse C nv. med 2 maskiner ( $2/255 = 0.78\%$  effekt)
  - klasse B nv. med 256 maskiner ( $256/65535 = 0.39\%$  effekt)
- For mange nettverk:
  - dagens Internett har i størrelsesorden 20000 nettverk
  - routingtabellene skalerer ikke
  - propageringsprotokollene skalerer ikke

## Subnetting

Nett 1 : 129.240.0.0 Destin-addr: 129.240.002.35

subnett 1: 129.240.1.0 Nett-maske = 255.255.255.0

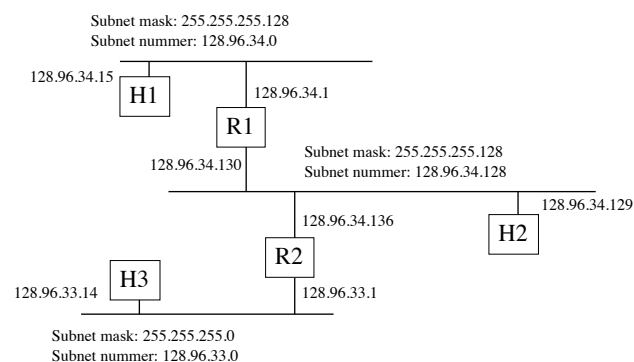
subnett 2: 129.240.2.0

subnett 3: 129.240.3.0

129.240.002.35 & 255.255.255.0 --> 129.240.002.0

- subnetting innfører et nytt adressenivå
- nett-masken identifiserer subnett-adressen
- subnett er bare synlige innenfor lokalområdet
- det er fullt mulig med flere subnett på samme fysiske nett

## Eksempel på subnett



### Forwarding tabell i ruter R1:

Subnet Number	Subnet Mask	Next Hop
128.96.34.0	255.255.255.128	interface 0
128.96.34.128	255.255.255.128	interface 1
128.96.33.0	255.255.255.0	R2

## Forwarding Algoritme

```
D = destination IP address
for each entry < SubnetNum, SubnetMask, NextHop>
  D1 = SubnetMask & D
  if D1 = SubnetNum
    if NextHop is an interface
      deliver datagram directly to destination
    else
      deliver datagram to NextHop (a router)
```

## Poenger

- ❑ Benytt en default ruter dersom intet passer
- ❑ Det er ikke nødvendig at alle enere i subnett-masken kommer etter hverandre.
- ❑ Kan legge flere subnett på samme fysiske nettverk.
- ❑ Subnett er ikke synlige for resten av internettet.

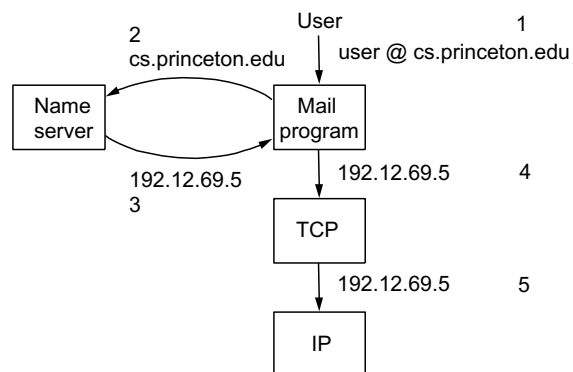
## "Supernet"

- Tilordne blokker av kontinuerlige nettverksnummer til flere nettverk i samme område
- Dette kalles CIDR: Classless Inter-Domain Routing
- Representer blokker med et enkelt par  
`<first_network_address, count>`
- Restriker blokkstørrelser til potenser av 2
- Bruk en bit-maske (CIDR mask) til å identifisere blokk-størrelse.
- Alle rutere må forstå CIDR adresser

## Navnhåndtering (DNS)

## Oversikt

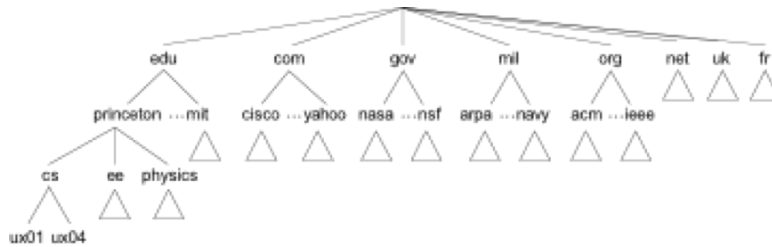
- Navn og adresser
  - navn er laget for at mennesker skal kunne lese dem
  - adressene er laget for å lette rutingen
- Navnerommet
  - definerer settet av mulige navn
  - flat versus hierarkisk
  - består av et sett med navn til adresse *bindinger*





## Domene Hierarki

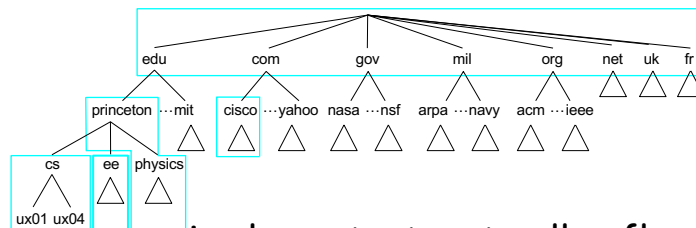
- Eksempel på hierarki



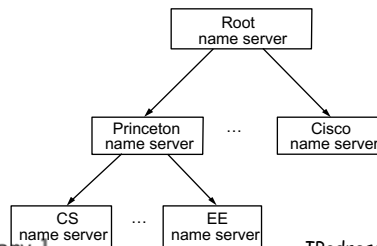
- Eksempel på navn: `ux04.cs.princeton.edu`

## Navnetjenere

- Del opp hierarkiet i *soner*



- Hver sone er implementert av to eller flere *navnetjenere*



## Resource Records

- Hver navnetjener vedlikeholder et sett av *resource records*
  - <Name, Value, Type, Class, TTL>
- Name/Value: Ikke nødvendigvis navn til IP adresser
- Type
  - NS: Value feltet gir domenenavnet til en maskin som er navnetjener som kan gi deg den adressen du er ute etter.
  - A: Value feltet gir IP-adressen.
  - CNAME: Value feltet gir kanonisk name for en gitt maskin. Dette benyttes til å innføre aliaser.
  - MX: Value feltet benyttes til å identifisere mail-server.
- TTL: hvor lenge resource record'en er gyldig

## Eksempel

Root server:

```
<arizona.edu, telcom.arizona.edu, NS, IN>  
<telcom.arizona.edu, 128.196.128.233, A, IN>  
  
<bellcore.com, thumper.bellcore.com, NS, IN>  
<thumper.bellcore.com, 128.96.32.20, A, IN>
```

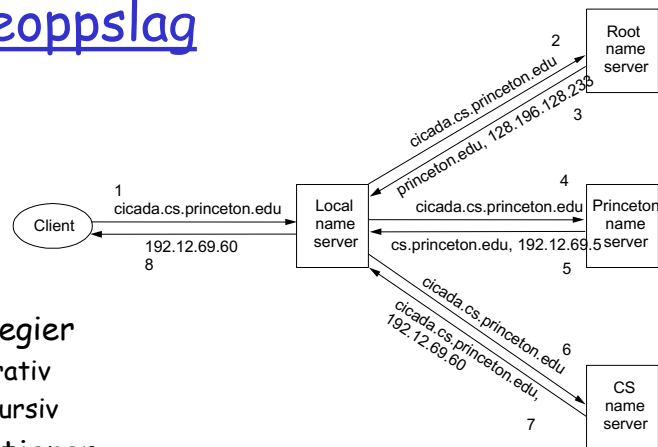
## Arizona server:

```
<cs.arizona.edu, optima.cs.arizona.edu, NS, IN>  
<optima.cs.arizona.edu, 192.12.69.5, A, IN>  
  
<ece.arizona.edu, helios.ece.arizona.edu, NS, IN>  
<helios.ece.arizona.edu, 128.196.28.166, A, IN>  
  
<jupiter.physics.arizona.edu, 128.196.4.1, A, IN>  
<saturn.physics.arizona.edu, 128.196.4.2, A, IN>  
<mars.physics.arizona.edu, 128.196.4.3, A, IN>  
<venus.physics.arizona.edu, 128.196.4.4, A, IN>
```

## CS server:

```
<cs.arizona.edu, optima.cs.arizona.edu, MX, IN>  
  
<cheltenham.cs.arizona.edu, 192.12.69.60, A, IN>  
<che.cs.arizona.edu, cheltenham.cs.arizona.edu,  
  CNAME, IN>  
  
<optima.cs.arizona.edu, 192.12.69.5, A, IN>  
<opt.cs.arizona.edu, optima.cs.arizona.edu,  
  CNAME, IN>  
  
<baskerville.cs.arizona.edu, 192.12.69.35, A, IN>  
<bas.cs.arizona.edu, baskerville.cs.arizona.edu,  
  CNAME, IN>
```

# Navneoppslag



## □ Strategier

- iterativ
- rekursiv

## □ Lokal tjener

- Trenger å kjenne root på bare ett sted (ikke i hver maskin)
- En cache per "nettverk".