

## 4. Reititys (Routing)

- **Verkkokerroksen tehtävänä on toimittaa data (paketit) lähettäjän koneelta vastaanottajan koneelle**
  - Välissä voi olla hyvin monimutkainen monista erilaisista aliverkoista koostuva verkko.
    - Internet, jossa miljoonia reitittämiä ja yli sata miljoonaa konetta, eri yritysten omistuksessa
      - 2.11.2000: 100. miljoonas 'host'
- **Miten tämä saadaan aikaiseksi?**

# Reititys Verkkokerroksen tärkein tehtävä

- **(hajautettu) päätöksenteko**
  - yhteydellinen: alussa
  - yhteydetön: jatkuvasti
- **jatkuvaa muutosta, vikoja ja virheitä**
  - rikkoutuvat komponentit, muuttuva topologia
  - tästä huolimatta reitityksen toimittava
- **ristiriitaisia vaatimuksia**
  - optimaalisuus /reiluus (fairness)
- **suorituskyky**
  - mean packet delay, network throughput

- **Koneilla ja reitittimillä yksikäsitteiset verkko-osoitteet**
  - IP-osoite: verkko ja koneen osoite verkossa
  - NAT (Network Address Translation)
- **Muuttuvassa verkossa pystyttävä selvittämään kulloinkin parhaat reitit**
  - reititysalgoritmi, reititysprotokolla
- **Reititin ohjaa reitillä eteenpäin**
  - reititystaulu (routing table)

# Reititysalgoritmit

- **mukautuva / mukautumaton** (adaptive / nonadaptive)
  - dynaaminen /staattinen
  - mittaukseen perustuva vai ‘kirjanpitolohjain’
  - suuri ero aikaskaalassa (sekunteja, minutteja  $\Leftrightarrow$  päiviä, ihminen muuttaa)
- **optimaaliset reitit**
  - kaikista lähteistä annettuun kohteeseen
  - $\Rightarrow$  puu, jonka juurena kohde
- **Ongelmia:**
  - reitittimien tietojen hankinta
  - verkko elää $\Rightarrow$  reitittimien tietojen ylläpito?

# Reititysalgoritmeja

- **Dijkstran reititysalgoritmi** (Shortest Path Routing)
  - yleisesti käytetty
- **tulvitus (flooding)**
- **satunnainen ('kuuma peruna', hot potato)**
- **vuopohjainen reititys** (Flow-Based Routing)
  - network flow problem
  - ratkaistaan matemaattisesti

# Vuopohjainen reititys (Flow-Based Routing)

- **viive = jonotusaika + siirtoaika**
  - etsitään pienin mahdollinen viive koko verkolle
- **tunnettava**
  - verkon topologia
  - kapasiteettimatriisi
    - eri linkkien kapasiteetti
  - liikennematriisi
    - eri solmujen välinen liikenne
  - alustava reititys

- **lasketaan**

- **kunkin linjan kuormitus  $\lambda_i$**
- **keskim. pakettien määrä kullakin linjalla  $\mu C_i$** 
  - keskim. pakettikoko =  $1/\mu$  . (esim. 800 bittiä)
- **keskim viive kullekin linjalle**

$$T = 1/(\mu C - \lambda) \text{ (jonoteoriasta)}$$

$1/\mu$  = keskim. paketin koko bitteinä

$C$  = kapasiteetti bps

$\lambda$  = keskim. pakettivirta (kuormitus)

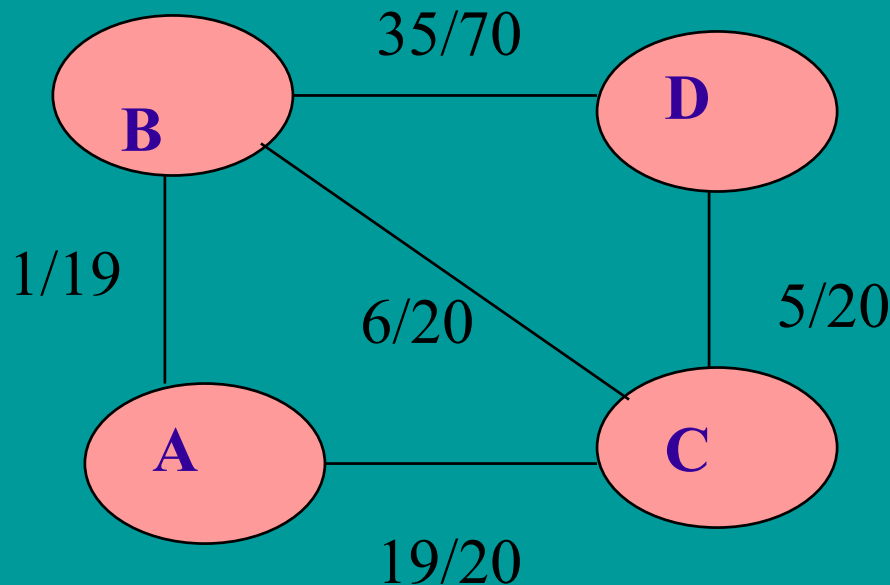
paketteina sekunnissa

- **koko verkon viive**
  - painotettu keskiarvo eri linkkien viipeistä
    - painotuksena linkin osuus koko liikenteestä
- **eri reititys algoritmien vertailu**
  - lasketaan erikseen kaikille reititysvaihtoehdoille
    - mahdollinen, vaikka raskas
  - valitaan ‘paras’
- **edellyttää kuormituksen pysyvän melko samanlaisena**
  - ei oikein sovellu koko ajan muuttuvaan verkkoon



# Piirikytkentäisten verkkojen reititys

- lyhyin polku (Shortest path first) (ABD tai ACD)
- vähiten kuormitettu polku (Least loaded path) (ABCD)
- eniten vapaita piirejä omaava polku (Maximum free circuit) (ABD)



A/B

B = linkin kapasiteetti

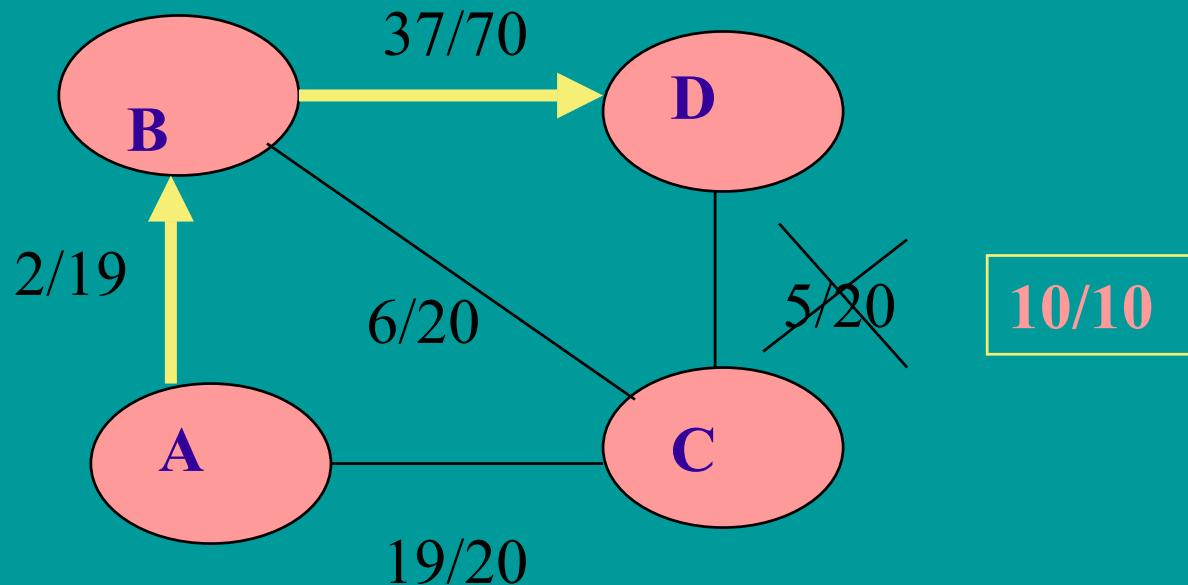
A = siitä käytössä

oleva osuus,

esim. piirien määränä

# • Piirikytkentäisissä verkoissa

- kaikki reitittimet tietävät kaikkien linkkien tilan
  - linkkitilatyypinen reititys
- tietojen oikeellisuus ja ajantasaisuus tärkeää



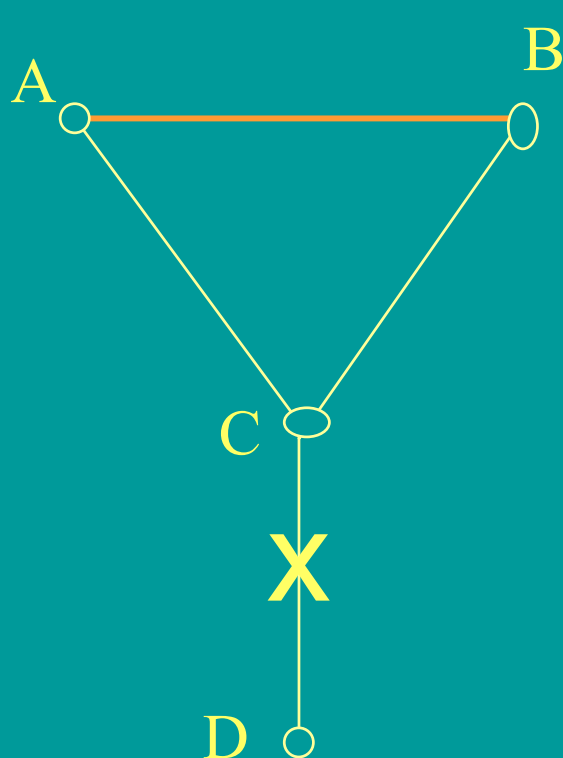
# Yleisesti käytetyt reititysalgoritmit

- **Etäisyysvektoreititys** (Distance Vector Routing)
  - ARPA-verkon alkuperäinen reititysalgoritmi
  - Internetin RIP-algoritmi
  - Ciscon IGRP ja EIGRP (mm. useita eri kustannusmittoja)
- **linkkitilareititys** (Link State Routing)
  - ARPA-verkon reititysalgoritmi vuodesta 1979
  - Internetin **OSPF**-algoritmi
  - ISO:n IS-IS

# Etäisyysvektoreititys

- **Solmut vaihtavat informaatiota vain naapuriensa kanssa**
- **Eri solmuilla eri näkemys verkosta**
  - hyvät uutiset etenevät nopeasti, huonot hitaasti
    - count- to-infinity,
    - simple split horizon : ei ilmoita naapurille sen kautta meneviä parhaita reittejä
    - Split horizon with poisoned reverse" ilmoittaa, mutta merkitsee ne äärettömiksi.

- ratkaisu ei toimi aina



Linkki CD katkeaa,  
A ja B ilmoittavat C:lle  
ettei D:hen pääse

C pääättelee, että D:tä ei  
voi saavuttaa

Kuitenkin A kuulee B:ltä,  
että sillä on etäisyys 2  
D:hen => oma etäisyys 3

# Reititysprotokollia

- **RIP**

- etäisyysvektoreireititys

- autonomisen alueen sisäinen protokolla

- naapurit vaihtavat reititystietoja keskenään

- Counting to Infinity

- Split Horizon

- Triggered Updates

- RIPv1

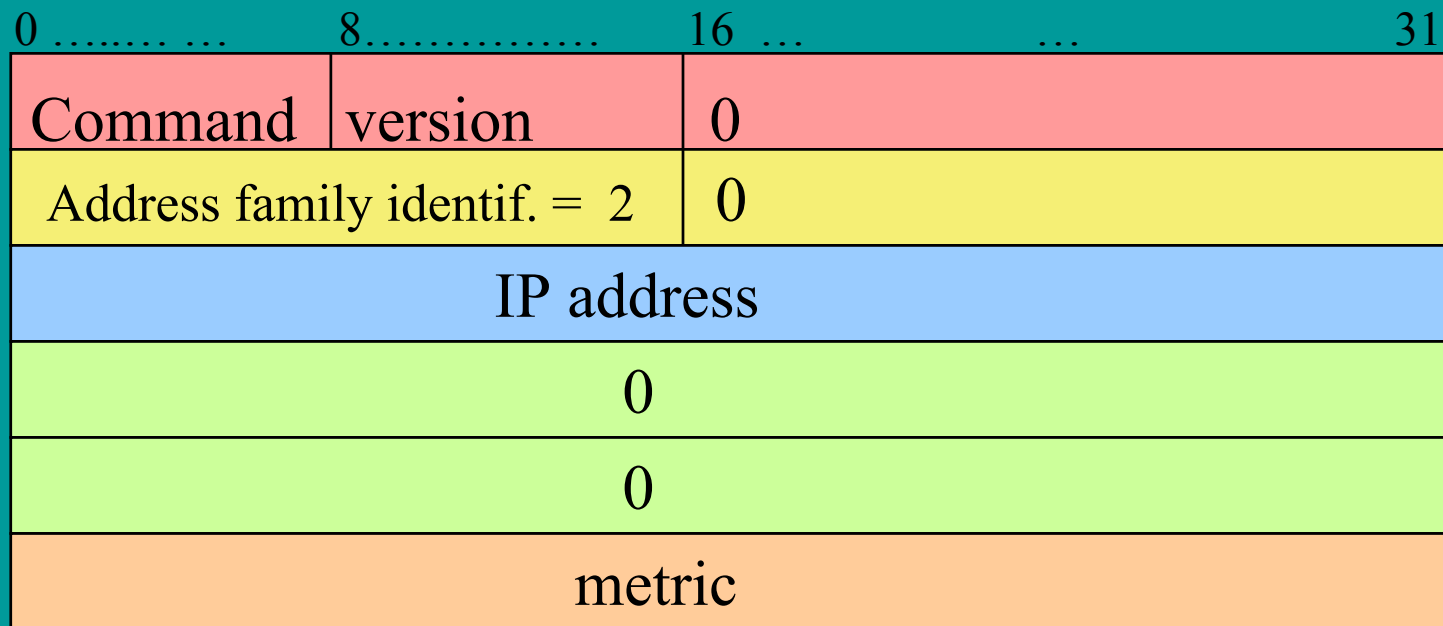
- RIPv2

- RIPng

# RIP (RFC 1058)

- **joka linkillä kustannus 1**
  - hyppyjä: 1-15 hyppyä
    - maksimi 15 => korkeintaan 15 hypyn matka mahdollinen
- **reititystietojen vaihto naapureiden kanssa**
  - RIP response message (advertisement)
    - yleislähetystenä (broadcast), jos mahdollista
  - n. 30 s välein. Jos naapuri ei lähettele 180 s sisällä, linkin oletetaan olevan poikki.
  - UDP-protokollaa käyttäen
    - RIP on toteutettu sovelluskerroksen prosessina ja siis sovelluskerroksen protokolla, joka käyttää UDP-porttia 520 sanomien lähettämiseen ja vastaanottoon

# RIP-sanoman otsake



**Command** = sanoman tyyppi: 1= pyyntö (request), 2 = vastaus (response)

**Address family identifier** = peruja UNIX-BSD:ssä käytetystä osoitustavasta; ajatuksena toteuttaa RIP muihin osoitusmuotoihin (esim. X.25, XNS)

**metric** = kustannus hyppinä ; max. = 16 eli ääretön



# RIP:n toiminta

- **Normaalisti lähetetään vastauksia**
  - 30 sekunnin välein
  - kun omassa taulussa muutoksia
    - ei heti, kun oma taulu muuttuu; vasta 1-5 sekunnin kuluttua
- **Reititin käsittelee saamansa vastaukset yhden kerrallaan**

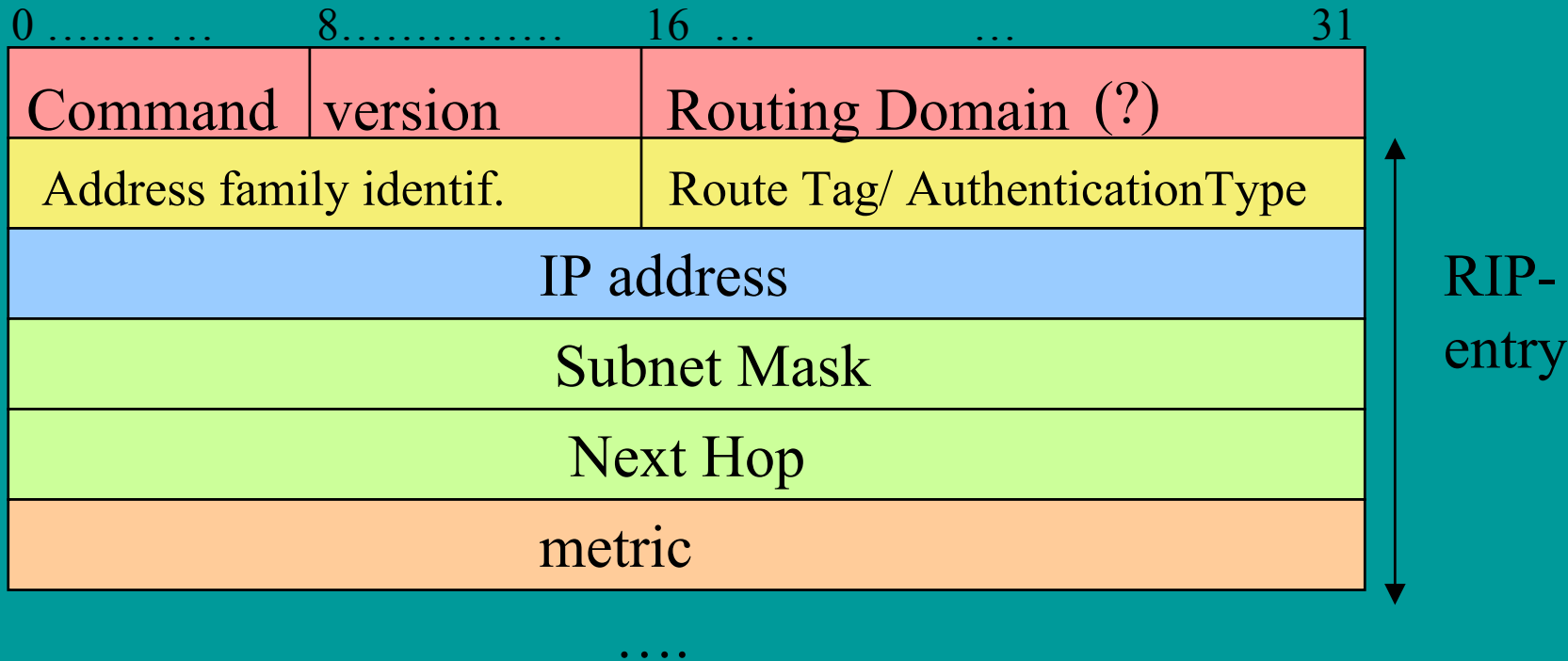
Kohteen osoite	etäisyysmitta	seuraava reititin	äskettäin päivitetty	useita ajastimia
192.55.2.5	10	193.46.4.8	U(ppdated)	26
.....	.....	.....	.....	.....

- **Yhdessä sanomassa korkeintaan 25 alkion tiedot**
  - tarvittaessa useita peräkkäisiä sanomia
- **Reititystietopyyntö, kun reititin aloittaa toimintansa**
  - koko reititystaulun sisältö
    - 0.0.0.0 osoitteena (default osoite) ja kustannuksena ‘ääretön’
    - normaali operaatio
  - tietyt reitit
    - kyselyssä ilmoitettuihin osoitteisiin
    - lähinnä vikojen selvittämisessä

# RIP-2 (RFC 2453)

- **tehokkaampi koodaus**
  - ei turhien nollakenttien lähettämistä
- **aliverkkoreititys**
  - RIP-1: aliverkot eivät näy ulospäin
  - RIP-2: aliverkkomaski osoitteen mukana => CIDR
- **autentikointi**
  - RIP-1 luotti porttiin 520, jota sai käyttää vain etuoikeutettu käyttäjä
  - RIP-2: ensimmäinen alkio voi olla autentikointisegementti
- **Next Hop, monilähetys**
  - RIP-1: yleislähetys

# RIP-2-sanoman otsake



# RIPng (RFC 2080) ja muita parannuksia

- **RIP-protokollan käyttö IPv6:n kanssa**
  - parannetut turvapiirteet
    - IPv6 turvapiirteet
  - pitemmät IP-osoitteet
- **päivitysten kuittaukset**
- **useiden eri kustannusmittojen käyttö**
- **“count-to-infinity”-ongelma**
  - ‘source-tracing’-algoritmi, joka etsii silmukat

# Linkkitilareititys

- **Globaali reititysalgoritmi**

- Kullakin reitittimellä käytössään koko verkon informaatio
- tästä lasketaan hajautetusti tai keskitetysti parhaat reitit
- monimutkainen algoritmi
  - => paljon laajempi standardi

# Linkkitilareititys (Link State Routing)

- **reitittimen tehtävät**
  - selvitettävä naapurit ja niiden osoitteet
  - mitattava etäisyys / kustannus naapureihin
  - koottava tietopaketti ko. tiedoista
  - lähetettävä tietopaketti kaikille reitittimille
  - laskettava lyhin reitti kaikkiin muihin reitittimiin
- **kyseessä maailman laajuinen verkko**
  - kaikki häiriöt sattuvat
    - joskus ja jossain
  - **vikasietoisuus**

# ongelmia

- **väärin toimiva reititin**
  - kertoo väärää tietoa
  - ei välitä tietopaketteja
  - väärentää tietopaketteja
  - laskee reitit väärin
- **isossa verkossa aina joku toimii väärin**
  - tavoitteena rajata ongelmat pienelle alueelle



# OSPF (Open Shortest Path First)

- **linkkitilaprotokolla**

- tavoitteet:

- avoin (eli julkinen)
- erilaisia eäisyysmittoja
- dynaaminen algoritmi
- myös palvelutyypin perustava reititys
- kyettävä kuorman tasoittamiseen ja usean reitin käyttämiseen
- hierarkkinen reititys
- suojauspiirteitä
- myös tunneloinnilla yhdistetyt reitittimet

- **eri reitit voivat olla ‘yhtä pitkiä’**
  - => liikenne voidaan reitittää usean reitin yli
  - => kuormituksen tasapainoitus
- **eikä välttämättä kaikkia paketteja lähetetä samaa reittiä**
  - osa parasta reittiä
  - osa toiseksi parasta

**lopputulos voi olla parempi**

# OSPF:n käyttöalueet:

- **kahden reitittimen välinen kaksipisteyhteys**
- **monen reitittimen yleislähetysverkot**
  - esim. useimmat lähiverkot (LAN)
- **monen reitittimen verkot, joissa ei ole yleislähetystä**
  - useimmat laajaverkot (WAN)

- 
- 

## • **Verkosta tehdään malli (suunnattu verkko)**

- reitittimet ja verkot solmuina, niiden väliset linjat kaarina
- kaarilla kustannuksina etäisyys, kustannus, luotettavuus
- multiaccess-verkkoa vastaa oma solmu, josta kustannus reitittimeen on nolla

## • **mallilla lasketaan lyhyin reitti kaikkien reititinparien välille**

- eri etäisyysmitoille omat reitit

# OSPF:n toiminta

- **reititystietojen vaihto**

- linkkitilaviestejä säännöllisin väliajoin ja topologian muuttuessa

- viestit tulvitetaan, viestit numeroidaan, viestit kuitataan

- viestit ohjataan valitulle (designed) välittäjäreitittimelle

- kommunikoi LAN:n tai alueen muiden reitittimien kanssa; kerää tiedot ja välittää ne eteenpäin

- jokainen reititin ei lähetä jokaiselle, vaan omalle välittäjäreitittimelleen

- vähentää viestien määrää:  $n(n-1)/2 \implies 2(n-1)$ , jos  $n = 20$ , niin  $20*19/2 = 190$  ja  $2*19 = 38!$

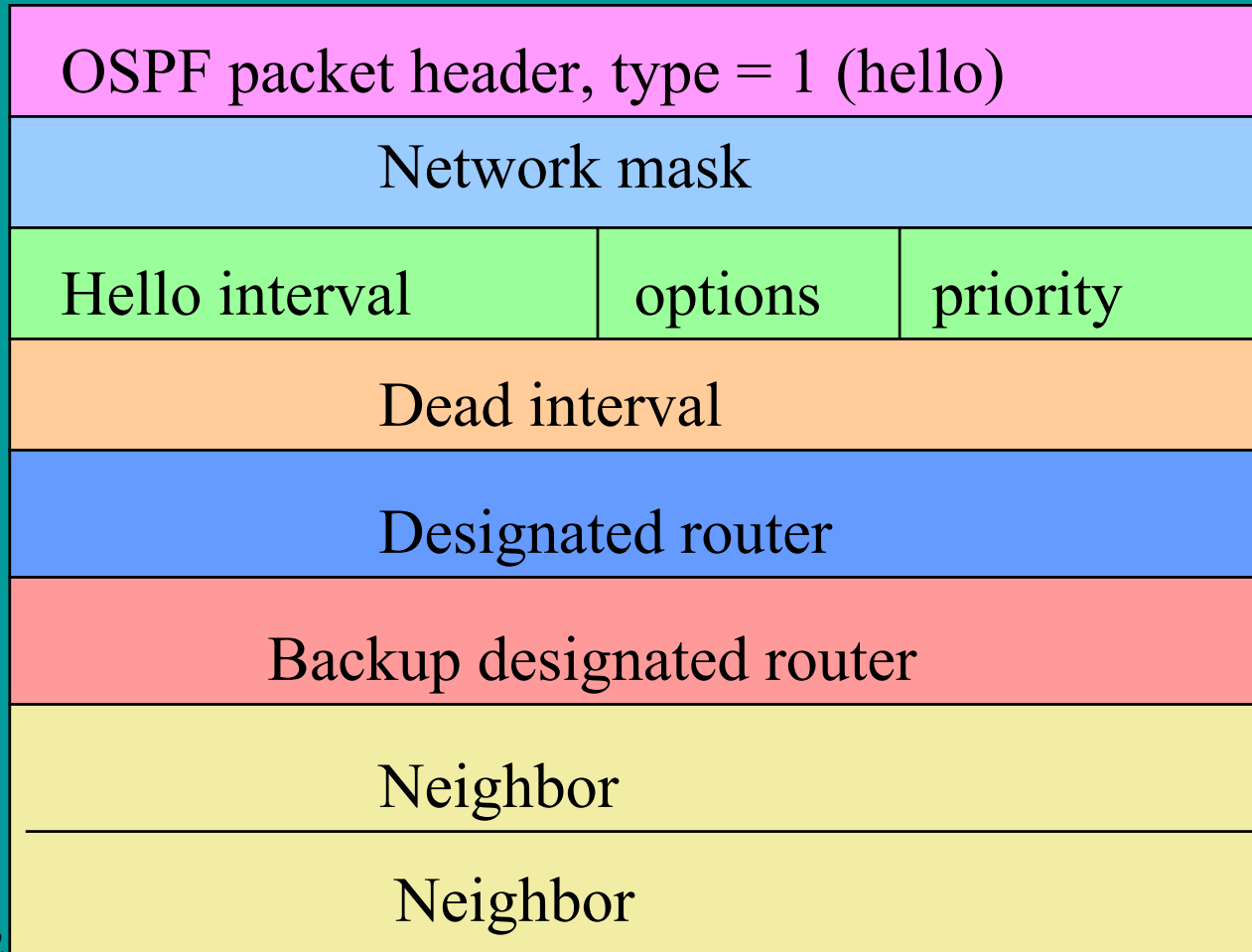
# Välittäjäreititin

- **Välittäjä valitaan Hello-protokollalla**
- **välittäjäreititin vähentää tulvituspaketteja**
  - riittää ensin lähettää monilähetyksenä välittäjäreitittimille
    - osoite 224.0.0.6=> kaikille välittäjäreitittimille
  - tarvittaessa välittäjäreititin monilähettää kaikille OSPF-reitittimille (224.0.0.5)
  - Entä, kun välittäjäreititin kaatuu?
    - valitaan myös varavälittäjä, joka vastaanottaa monilähetykspaketteja, mutta ei vastaa mihinkään
    - välittäjän kaatuminen havaitaan Hello-protokollalla

# OSPF-sanomat

- **hello**
  - naapurien selvillesaaminen
- **link state update**
  - omien linkkikustannusten lähettäminen
- **link state ack**
  - vastaanotettujen linkkikustannusten kuittaus
- **database description**
  - tietokannan ajantasaisuuden selvittäminen
- **link state request**
  - toisen linkkikustannusten kysyminen

# Hello-paketti





# Hello-paketin kentät

- **Network mask = liitännäkortin aliverkkomaski**
- **Hello interval = hello-sanomien lähetysväli**
- **Options:**
  - T-bitti => TOS-reitityskykyinen
  - E-bitti = ulkoisten reittien vastaanotto ja lähetys
- **Priority: reitittimen prioriteetti 0-255**
  - välittäjäksi korkeimman prioriteetin reititin;
  - jos sama arvo usealla, niin suurin ID-numero valitaan
- **Dead interval**
  - jos tässä ajassa ei tule hello-sanomaa, merkitään 'kuolleiden' listaan

# Hello-paketin kentät jatkuvat

- **Designated router**
- **Backup designated router**
  - reititin ilmoittaa haluavansa toimia välittäjäreitittimenä tai varavälittäjäreitittimenä
  - valintaa suoritetaan jatkuvasti ja joka hello-sanomassa
  - reititin muistaa, ketkä ilmoittautuneet välittäjiksi

# Hierarkkinen reititys

- **reitityksen skaalautuvuus**

- isossa verkossa runsaasti reitittämiä
  - reititystaulut suuria
  - reittien laskeminen raskasta
  - tietopaketit kuluttavat linjakapasiteettia

- **hierarkiaa**

- jaetaan verkko ja sen reitittimet autonomisiin osiin
  - AS (autonomous system)
    - yritysten ja organisaatioiden omat verkot
    - “A set of routers and networks under the same administration.”
    - Kullakin AS:llä on oma 16-bittinen AS-numero.

# Hierarkkisen reitityksen ongelmat

- **reitin pituus kasvaa**
  - aina ei voida käyttää optimaalista reittiä
  - yleensä siedettävä
- **hierarkiatasojen määrä**
  - suorituskyky
  - hallinto