

-
-
-

5. Verkkokerros

- **sovelluskerros**
 - ‘asiakas’
- **kuljetuskerros**
 - ‘end-to-end’
- **verkkokerros**
 - ‘deliver packets given to it by its customers’
- **siirtoyhteyskerros**
- **peruskerros**

-
-
-

Verkkokerroksen palvelut

- **tavoitteet**

- palvelut riippumattomia aliverkkojen tekniikasta
- kuljetuskerros eristettävä aliverkkojen ominaisuuksista
 - lukumäärä
 - tyypit
 - topologia
- kuljetuskerroksen käyttämät **verkko-osoitteet globaaleja**

-
-
-

connection-oriented ~ connectionless

- **yhteydetön** (Internet, 30 vuoden kokemus)
 - aliverkot ovat luonnostaan epäluotettavia
 - tehtävä: bittien kuljetus
 - operaatiot: send packet, receive packet
 - virheen tarkistus, vuonvalvonta isäntäkoneille
- **yhteydellinen** (puhelin 100 vuoden kokemus)
 - muodostetaan yhteys, neuvotellaan parametrit (palvelunlaatu (QOS), kustannus)
 - kaksisuuntainen kuljetus, paketit järjestyksessä

-
-
-

Verkkokerroksen tärkein tehtävä: reititys

- **(hajautettu) päätöksenteko reitistä**
 - yhteydellinen: alussa
 - yhteydetön: jatkuvasti
- **jatkuvaa muutosta verkossa**
 - rikkoutuvat komponentit, muuttuva topologia
- **ristiriitaisia vaatimuksia reititykselle**
 - optimaalisuus /reiluus (fairness)
- **reitityksen suorituskyky**
 - mean packet delay, network throughput

-
-
-

Reititysalgoritmi

- **Päättää, mikä reitti valitaan**
 - mihin paketti ohjataan seuraavaksi
- **dynaaminen verkkoympäristö => dynaaminen reititys**
 - jatkuvaan verkon tarkkailuun perustuva
 - Internetin reititys
 - muuttumaton ympäristö => käytetään kerran laskettuja reittejä tai sovittua lähetystapaa
 - tulvitus (flooding)
 - Dijkstran algoritmilla lasketut lyhyimmät reitit

-
-
-

Tulvitus

- jokainen saapunut paketti lähetetään kaikille muille ulosmenoille
 - => verkko täyttyy pian paketeista
- eri tapoja tulvituksen lopettamiseen
 - käsitellään harjoituksissa
- käyttö
 - tietyissä erityistilanteissa tilanteissa hyödyllinen
 - käsitellään harjoituksissa

-
-
-

Dijkstran algoritmi

- **‘lyhyin’ reitti yhdestä solmusta muihin**
 - $A \rightarrow \{\text{muut solmut}\}$
- **kaariin liittyy kustannus**
 - kapasiteetti (bps)
 - viive: hyppyjä, aikaa
 - raha
 - virhetodennäköisyys

-
-
-

• Algoritmi

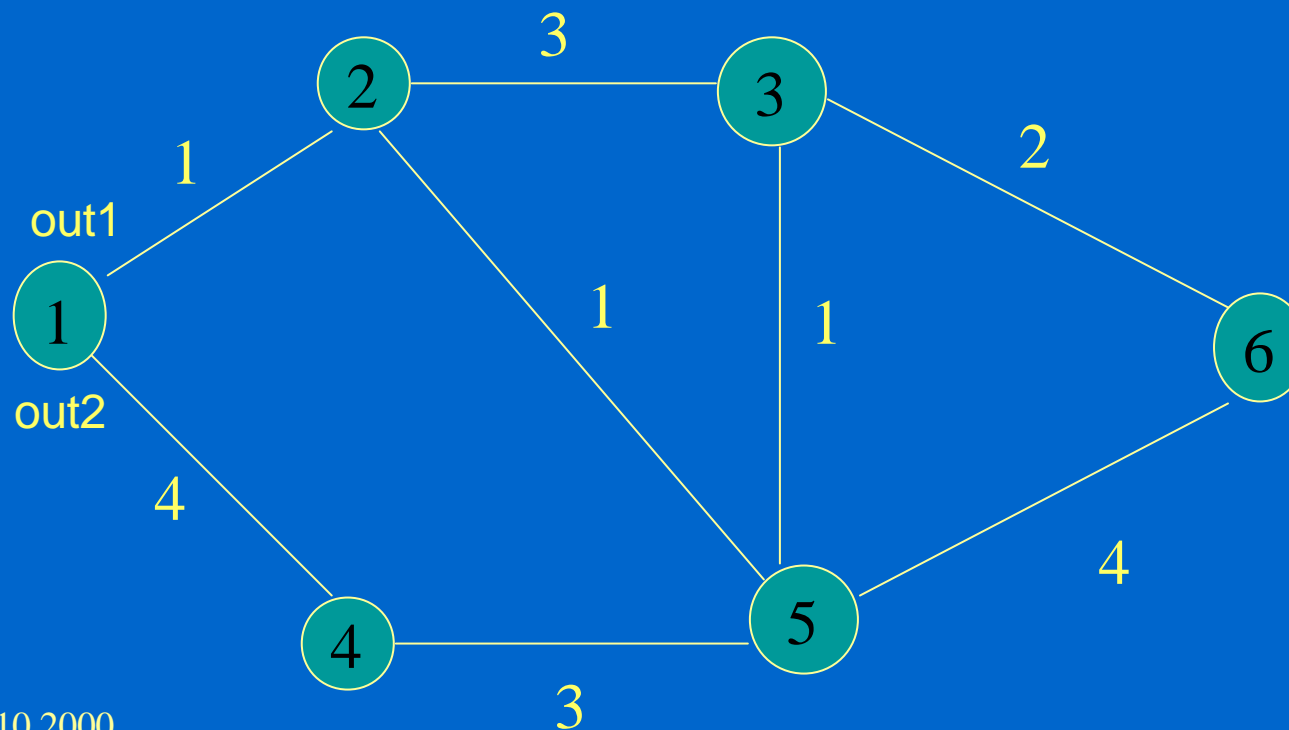
- merkitään D_i on solmun i tähän asti tutkituista reiteistä solmuun A halvin kustannus eli lyhyin pituus
- verkko $G = (V, E)$, V on solmujen joukko, E kaarten joukko
- olkoon d_{ij} on kaaren (i,j) kustannus (> 0). Jos kaarta ei ole, d_{ij} on ääretön
 - algoritmossa oletetaan, että kaikki kustannukset ovat ei-negatiivisiä

1. $P := \{1\}; D_1 := 0; D_j := d_{j1} \ (j \neq 1);$
2. while $P \neq V$ do
3. etsi solmu i , joka ei vielä ole
joukossa P ja jolle $D_i = \min D_j$ P :hen
kuulumattomista solmuista
4. $P := P \cup \{i\}$
5. kaikille muille P :hen kuulumattomille
solmuille j $D_j := \min\{D_j, d_{ij} + D_i\}$
6. end while
7. end

•
•
•

Esimerkki

- Tarkastellaan esimerkkinä verkkoa



1. $P = \{1\}$; $D1 := 0$; $D2 := 1$; $D3 := \text{ääretön}$,
 $D4 := 4$; $D5 := \text{ääretön}$, $D6 := \text{ääretön}$

3. pienin D_i on solmulla 2 (=1)

4. $P = \{1, 2\}$

5. $D3 := 1 + 3 = 4$, $D4 = 4$, $D5 := 1 + 1 = 2$,
 $D6 = \text{ääretön}$

3. pienin D_i nyt solmulla 5 (=2)

4. $P=\{1,2,5\}$

5. $D3:= 1+2=3, D4:=4, D6:=4+2=6$

3. pienin Di solmulla 3 (=3)

4. $P=\{1,2,3,5\}$

5. $D4:=4, D6:=2+3=5;$

3. Pienin Di solmulla 4 (=4)

4. $P=\{1,2,3,4,5\}$

5. $D6=5$

4. $P=\{1,2,3,4,5,6\}$

•
•
•

Löydetyt reitit ja kustannukset

- 1-> 2 :1
- 1-> 2->5->3: 3
- 1-> 4: 4
- 1->2->5: 2
- 1->2->5->3->6: 5

Solmu	linkki	kustann.
2	2	1
3	2	3
4	1	4
5	2	2
6	2	5

Solmulle 1

Reititystaulu

- **Kukin reititin pitää kirjaa reittitiedoista**
 - minne paketti seuraavaksi lähetetään

Kohde	minne lähetetään
Abc	reititin D, ulosmeno 2
...
Xyz	reititin T, ulosmeno 3

- **reitittimien tietojen hankinta ja ylläpito?**
 - erityisen nopeasti muuttuvassa hyvin isossa verkossa

-
-
-

Reititystietojen keruu

- kukin reititin kerää ‘kustannustietoja’ omasta ympäristöstään
 - esim. viiveet naapurireitittimiin
- ja vaihtaa tietoja muiden reitittimien kanssa
 - tai lähettää tiedot reitittimelle, joka keskitetysti laskee parhaat reitit
- kukin laskee esim. Dijkstran algoritmilla parhaat reitit koko verkosta
 - tai saa tarvitsemansa reititystiedot ne laskeneelta

•
•
•

Etäisyysvektoreireitys (distance vector)

- **Arpanetin alkuperäinen reitys**

- vieläkin jonkin verran käytössä Intenetissä

- **kullakin reitittimellä reititystaulu**

- kullekin verkon reitittimelle

- ulosmenolinja
- aika/etäisyys kohteeseen

- hyppyjen lkm

- arvioitu viive

- jononpituus

- jokin mitattavissa oleva

-
-
-

reititystaulun ylläpito

- **tietojen vaihto naapurireitittimien kanssa**
 - tietyin aikaväleihin
 - tilan vaihtuessa
- **lasketaan uudet reittitaulut**
 - ‘kustannus’ naapuriin +
 - naapurin ilmoittama ‘kustannus’ kohteeseen
 - kullekin solmulle valitaan pienimmän ‘kustannuksen’ reitti

-
-
-

Ongelma: tietojen muuttumisnopeus

- **tietojen muuttamiseen kuluu aikaa**
- **reagoi nopeasti hyviin uutisiin**
 - uusi nopea reitti löytynyt/linkki jälleen pystyssä
 - tieto etenee joka vaihdossa yhden hypyn
- **reagoi hitaasti huonoihin uutisiin**
 - linkki nurin => etäisyys ääretön
 - joka vaihdossa ‘paras arvio’ huononee yhdellä
 - **count - to - infinity** -ongelma

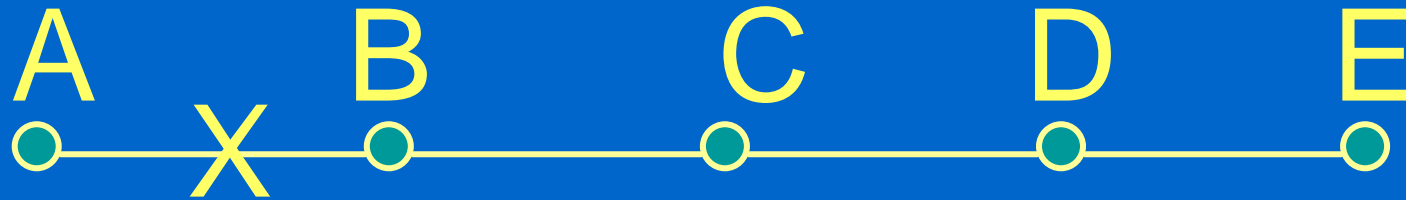
Hyvät uutiset etenevät nopeasti:



Aluksi yhteys A:han on poikki ja sitten linkki AB toimii taas:

	B	C	D	E
	ääretön	ääretön	ääretön	ääretön
	1	ääretön	ääretön	ääretön
	1	2	ääretön	ääretön
	1	2	3	ääretön
	1	2	3	4

Huonot uutiset etenevät hitaasti:



Toimiva linkki katkeaa välillä AB:

	B	C	D	E
1	1	2	3	4
3	3	2	3	4
3	3	4	3	4
5	5	4	5	4
5	5	6	5	6
7	7	6	7	6
7	7	8	7	8

-
-
-

Linkintilareititys (Link State Routing)

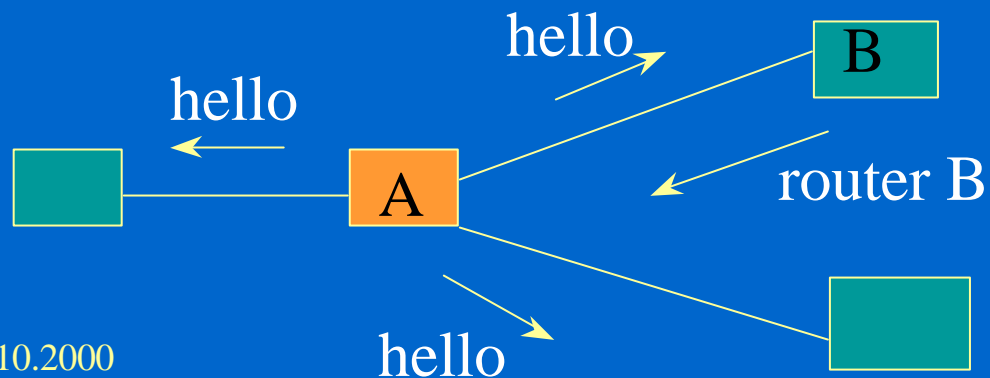
- **reitittimen tehtävät**

- selvitettävä naapurit ja niiden osoitteet
- mitattava etäisyys / kustannus naapureihin
- koottava tietopaketti ko. tiedoista
- lähetttävä tietopaketti kaikille reitittimille
- laskettava lyhin reitti kaikkiin muihin reitittimiin esim. Dijkstran algoritmilla

•
•
•

Naapurien löytäminen

- reititin lähettää jokaiseen kaksipisteyhteyteen **HELLO**-paketin
- linjan toisessa päässä oleva reititin vastaa ja lähettää nimensä
 - router ID
 - nimien oltava yksikäsitteisiä koko verkossa



-
-
-

Etäisyyden mittaaminen

- **kaikille naapureille ECHO-paketti**
 - vastaanottajan palautettava paketti välittömästi
- **=> kiertoviive (round-trip-time)**
 - dynaaminen etäisyysmitta
- **pitäisikö ottaa kuormitus huomioon?**
 - kello käynnistetään , kun paketti viedään jonoon
 - kello käynnistetään, kun paketti lähtee
 - kuormitus mukana kuvaa todellista tilannetta
 - jos kuormitus mukana => reititys muuttaa kuormitusta
=> reititys suosii huonoa reittiä

-
-
-

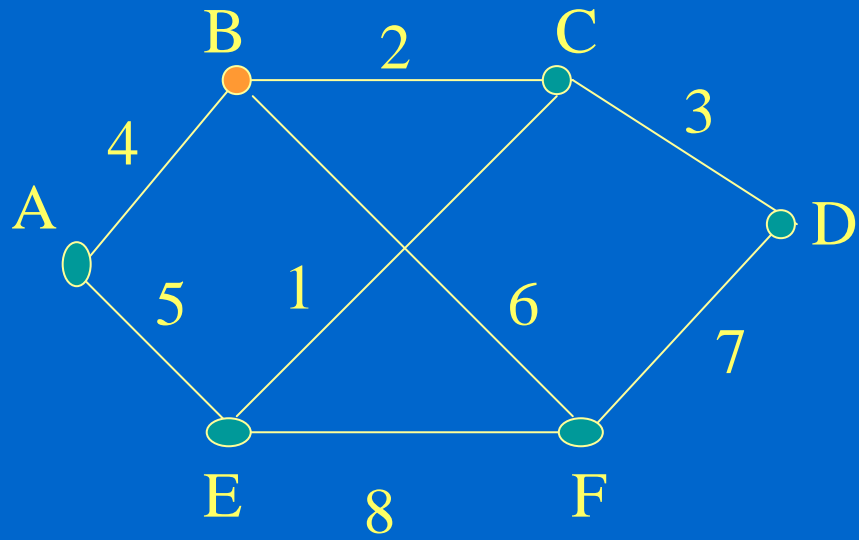
Tietopakettien kokoaminen

- **muodostus**

- tietyin aikavälein
- kun muutoksia havaittu

- **sisältö**

- reitittimen tunnus
- paketin järjestysnumero
- paketin ikä
- etäisyydet kuhunkin reitittimen naapuriin



B	
seq	
age	
A	4
C	2
F	6

-
-
-

Tietopakettien jakelu

- **käytetään tulvitusta (n. 10 minuutin välein)**
 - **pidetään kirjaa jo nähdystä paketeista**
 - reititin A, paketti 145
 - => paketti lähetetään korkeintaan kerran
 - paketissa elinaikalaskuri (age, time-to-live)
 - väärät ja vanhentuneet tiedot katoavat aikanaan, vaikka reititin itse olisikin vikaantunut
- **tietopaketit kuitataan**
 - linjavirheiden takia

-
-
-

Miksi elinaikalaskuri on tarpeen?

- **virheellinen järjestysnumero**

- kaatunut reititin aloittaa väärästä numerosta

- edennyt jo pakettiin 204 ja aloittaa uudestaan paketista 0 => kaikki seuraavat paketit hylätään duplikaatteina pakettiin 205 saakka

- virhe tietopaketin seq-kentässä

- 4 muuttuu virheellisesti 65540:ksi => seuraavat paketit hylätään pakettiin 65541 saakka

-
-
-

elinaikalaskuri (TTL-laskuri)

- **laskuri vähenee ajan kuluessa**
 - vähenee yhdellä sekunnin välein
- **paketti tuhotaan, kun laskuri = 0**
 - vanhentunut (virheellinen) tieto poistetaan
 - pitkäikäinen elinaika \gg päivitysten väli
 - tuhoetaan vain jos reititin kaatunut
 - usea (6) paketti on jäänyt saapumatta reitittimeltä
- **käytössä myös tulvituksessa**
 - kukin reititin vähentää yhdellä

-
-
-

Lisäparannuksia

- **paketteja ei lähetetä välittömästi eteenpäin**
 - ne jätetään odottamaan
 - jos samalta reitittimeltä tulee muita paketteja, niistä valitaan vain yksi, tuorein edelleenlähetettäväksi

•
•
•

Reittitaulun laskeminen

- **kukin reititin laskee omat reittitaulunsa**
- **kaikki tarvittava tieto on saatu tietopakettien avulla**
 - kukin linkki molempiin suuntiin
- **laskeminen Dijkstran algoritmilla**
 - lyhyin reitti kuhunkin muuhun reitittimeen
 - isoissa verkoissa voi olla muisti- ja laskenta-aikaongelmia

-
-
-

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

-
-
-

Käyttö

- **paljon käytetty nykyisissä verkoissa**
 - Internetin OSPF-protokolla
 - ISO:n IS-IS -protokolla

-
-
-

Hierarkkinen reititys

- **reitityksen skaalautuvuus**

- isossa verkossa runsaasti reitittämiä (Internet: miljoonia)

- reititystaulut suuria
- reittien laskeminen raskasta
- tietopaketit kuluttavat linjakapasiteettia

- **hallinta-autonomia => autonominen järjestelmä AS**

- organisaatio päättää omista asioistaan

- myös reitityksestä
 - oma sisäinen reititystapa

•
•
•

Reitityshierarkia

- **Ylimmällä tasolla AS**
 - sama reititys AS:n sisällä
 - tehokkuus tärkeää
 - reititys AS:ien välillä
 - ‘poliittinen asia’
- **AS:n sisällä alueita**
 - jaetaan reitittimet ryhmiin (alueet, regions)
 - kukin reititin tuntee kaikki alueensa sisällä
 - tietää mikä reititin hoitaa liikenteen muihin alueisiin

-
-
-

Hierarkkisen reitityksen ongelmat

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

•
•
•

5.4 Internetworking

- **verkot erilaisia**
 - nyt ja aina
- **verkkoja yhdistävät**
 - **toistin**: bittien kopiointia
 - **silta**: kehys, store-and-forward
 - **reititin**: paketti, store-and-forward, erilliset verkot
 - kulj.kerr. yhdyskäyt.: tavuvirta kuljetuskerroksessa
 - sovelluskerr. yhdysk.: sovelluksen tietoyksiköitä

-
-
-

- **silta**

- tunnettava kehysotsikot
- ei tarpeen tietää hyötykuorman takana olevaa protokollaa

- **reititin**

- tunnettava verkkoprotokolla
- alakerran toimintatavoista ei väliä

- **käytännössä termejä käytetään vapaammin!**

-
-
-

Verkkojen erot

- **palvelu**
 - yhteydellinen / yhteydetön
- **protokolla**
- **osoittaminen**
 - yksitasoinen / hierarkkinen
- **monilähetys/yleislähetys**
 - on / ei
- **paketin koko**

•
•
•

lisää eroja:

- **palvelulaatu**
- **virheiden käsittely**
- **vuon valvonta**
- **ruuhkan valvonta**
- **turvaus**
- **parametrit**
- **laskutus**

-
-
-

- **ongelmana on erilaisten toiminnallisuuksien yhteensopivuus**

- luotettavuus
- ruuhkan valvonta
- kuittaukset
- toimitusaikatakuut

-
-
-

Yhteydettömien verkkojen yhdistäminen

- verkkokerroksen protokollien oltava (lähes) samoja
- osoittaminen
 - IP: 32-bittinen osoite
 - OSI: puhelinnumeron kaltainen osoite
 - osoitteiden yhteensovittaminen?
 - globaaliosoiteavaruus? standardi?

-
-
-

Yhdistetyn verkon reititys

- **kahden tason reititysalgoritmi**
 - kunkin verkon sisällä (intranet routing)
 - interior gateway protocol
 - verkkojen välillä (internet routing)
 - exterior gateway protocol
 - gateway tässä vanhempi termi reitittimelle!
- **eroja**
 - EGP eri maiden välillä
 - EGP: erilaiset verkkokustannukset, erilainen QoS

-
-
-

Pakettien pirstominen (fragmentation)

- **kaikissa verkoissa paketilla jokin maksimikoko**
 - laitteisto (TDM-viipaleen pituus)
 - käyttöjärjestelmä (käytetty puskurinkoko)
 - protokolla (pituuskentän bittien lukumäärä)
 - standardinmukaisuus
 - virheistä johtuvan uudelleenlähetyksen vähentäminen
 - tasapuolisuuden tavoite
- **48 tavua (atm) => 65515 tavua (IP)**

-
-
-

Liian iso paketti verkkoon

- **liian iso paketti paloitellaan yhdyskäytävässä**
- **missä paketti kootaan?**
 - samassa verkossa, missä paloiteltiin
 - kaikki paketit ohjattava samaan yhdyskäytävään
 - jatkuvaa pilkkomista ja kokoamista!
 - vasta määränpäässä
 - pieni pakettikoko => lisää yleisrasitetta
 - kaikkien solmujen kyettävä kokoamaan paketteja

Pakettien kokoaminen

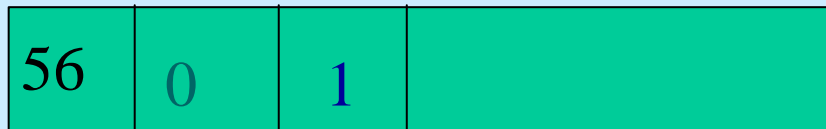
- **edellyttää palojen ‘numerointia’**
 - on tiedettävä, mikä paketin mikä osa on kyseessä
- **kaikissa paloissa alkuperäisen paketin tunniste + sijainti paketissa**
 - sijainti: pakettiin kuuluvan ensimmäisen tavun sijainti alkuperäisessä paketissa
- **lisäksi tieto, onko pala paketin viimeinen**

6.10.2000

- tai tiedettävä paketin pituus



alkuperäinen paketti



paketin alkuosa



paketin loppuosa

paketin
tunnus

sijainti-
kohta
eli

osan
numero

viimeinen
paketin
osa?

paketin data

•
•
•

5.5 Internetin verkkokerros

- **Internet**

- on kokoelma ‘itsenäisiä’ aliverkkoja eli autonomisia järjestelmiä (AS, Autonomous Subsystem)
- joita yhdistää runkolinjat

- **IP-protokolla**

- verkkotason protokolla, joka pitää Internetin koossa
- tavoite: kuljettaa paketti (datagram) lähteestä kohteeseen yli kaikkien tarpeellisten verkkojen

-
-
-

IP-protokolla

- **IP-datasähke**

- otsake
- dataosa

- **otsake**

- pituus 20 tavun kiinteä osa
- vaihtelevan mittainen valinnainen osuus
- big endian: vasemmalta oikealle, korkein bitti ensin

-
-
-

IP-otsakkeen kentät

- **versio**

- **IHL**

- otsakkeen pituus vähintään viisi 32 bitin sanaa (20-60 tavua)

- **type of service**

- kertoo halutun palvelun

- nopeus, luotettavuus, kapasiteetti
- ääni <-> tiedostonsiirto

- yleensä ei käytössä

6.10.2000

49

-
-
-

Type of service -bitit:

- **presedence-kenttä** (3 bittiä)
 - sanoman **prioriteetti** 0-7
 - 0 normaali
 - 7 verkon valvontapaketti
- **D-bitti, T-bitti, R-bitti**
 - mikä on tärkeää yhteydessä
 - D: viive (Delay),
 - T: läpimeno (Throughput)
 - R: luotettavuus (Reliability)
- lisäksi vielä 2 käyttämätöntä bittiä

-
-
-

IP-otsakkeen kentät jatkuvat

- **Total length**

- koko datäsähkeen pituus
- maksimi 65535 tavua
 - maksimipituus vielä riittävä, mutta tulevaisuuden nopeille verkoille jo ongelma

- **Identification**

- tietosähkeen numero
- kaikissa saman tietosähkeen osissa sama tunnus

-
-
-

IP-otsakkeen kentät jatkuvat

- **DF- bitti (Don't fragment)**
 - kieltää paloittelun
 - esim. jos vastaanottaja ei kykene kokoamaan datasähkettä
- **MF-bitti (More fragments)**
 - ilmoittaa, onko datasähkeen viimeinen osio vai tuleeko vielä lisää

-
-
-

IP-otsakkeen kentät jatkuvat

- **Fragment offset**

- osion paikka datasähkeessä
- osioiden oltava 8 tavun monikertoja (paitsi viimeisen)
- 13 bittiä => korkeintaan 8192 osiota yhdessä datasähkeessä

- **lisäksi 1 käyttämätön bitti**

-
-
-

IP-otsakkeen kentät jatkuvat

- **Time to live**

- rajoittaa paketin elinaikaa
- maksimi 255 sekuntia
- vähenee
 - joka hypyllä reitittimestä toiseen
 - myös odottaessaan reitittimessä (ei yleensä)
 - paketti hävitetään, kun laskuri menee nolille

- **Protocol**

- mille kuljetuskerrokselle kuuluu
 - esim. TCP- tai UDP-siirtoon kuuluva

-
-
-

IP-otsakkeen kentät jatkuvat

- **Header checksum**
 - tarkistussumma lasketaan vain otsakkeelle
 - 16-bitin sanat lasketaan yhteen yhden komplementin aritmetiikalla
 - laskettava uudestaan joka reitittimessä
- **Source address, Destination address**
 - kohteen ja lähettäjän osoitteet muodossa
 - verkon numero ja isäntäkoneen numero

-
-
-

IP-otsakkeen kentät jatkuvat

- **Options**

- vaihtelevan mittaisia

- 1. tavu kertoo option koodin
 - voi seurata pituuskenttä
 - datakenttiä
 - täytettä jotta 4 tavun monikertoja

- käytössä 5 optiota

- mutta reitittimet eivät välttämättä ymmärrä

-
-
-

Optiot

- **Security**

- datasähkeen luottamuksellisuus ja salassapidettävyys

- **Strict source routing**

- datasähkeen kuljettava tarkalleen annettua reittiä

- **Loose source routing**

- kuljettava ainakin annettujen reitittimien kautta

- **Record route**

- reitin varrella olevat reitittimet liittävät tunnuksensa

- **Timestamp**

- tunnuksen lisäksi liitettävä myös aikaleima