

4. Verkkokerros

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

30.3.2001

1

Verkkokerroksen palvelut

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

30.3.2001

2

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
 - vuonvalvonta, virhevalvonta

30.3.2001

3

Virtuaaliipiiri (virtual circuit)

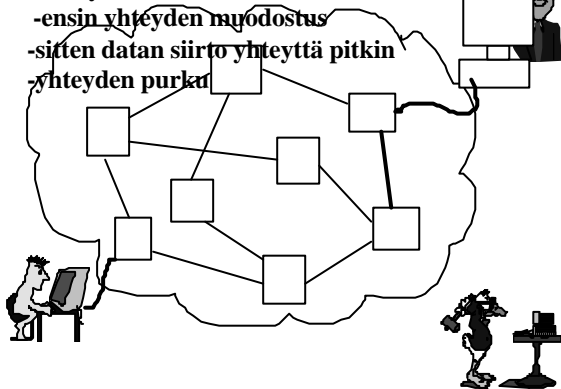
- **Pakettikytkentäinen verkko voidaan toteuttaa kahdella tavalla**
 - datasähkeverkko
 - jokainen paketti käsitellään ja reititetään erikseen
 - pakettien järjestys voi muuttua
 - virtuaaliipiiriverkko
 - ~ piirikytkentäinen verkko
 - ensin yhteyden (virtuaaliipiirin) muodostus
 - sitten pakettien lähettäminen yhteyttä pitkin
 - ATM, X.25

30.3.2001

4

Piirikytkentäinen verkko

- ensin yhteyden muodostus
- sitten datan siirto yhteyttä pitkin
- yhteyden purku



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

30.3.2001

6

Reititysalgoritmi

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

30.3.2001

7

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

30.3.2001

8

Dijkstran algoritmi

- **'lyhyin' reitti yhdestä solmusta muihin**
 - A -> {muut solmut}
- **kaariin liittyy kustannus**
 - kapasiteetti (bps)
 - viive: hyppyjä, aikaa
 - raha
 - virhetodennäköisyys

30.3.2001

9

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-negatiivisia

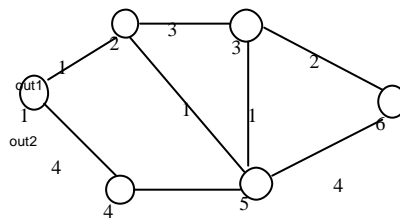
30.3.2001

10

1. $P := \{1\}; D_1 := 0; D_j := d_{j1} (j <> 1);$
2. while $P <> V$ 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



30.3.2001

12

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 D_i solmulla 3 (=3)

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

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

3. Pienin D_i 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

30.3.2001

15

Reititystaulu

- **Kukin reititin pitää kirjaa reititiedoista**

- 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

30.3.2001

16

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

30.3.2001

17

Etäisyysvektoreititys (distance vector)

- **Arpanetin alkuperäinen reititys**

- vieläkin RIP jonkin verran käytössä Intenetiissä

- **kullakin reitittimellä reititystaulu**

- kullekin verkon reitittimelle

- ulosmenolinja
- aika/etäisyys kohteeseen
 - hyppyjen lkm
 - arvioitu viive
 - jononpituus
 - jokin mitattavissa oleva

30.3.2001

18

reititustaulun ylläpito

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

30.3.2001

19

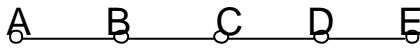
Ongelma: tietojen muuttumisnopeus

- **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

30.3.2001

20

Hyvät uutiset etenevät nopeasti:

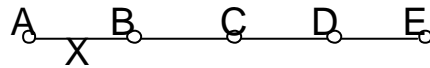


Aluksi yhteys A:han on poikki ja sitten linkki AB toimii taas:

	B	C	D	E
ääretön	ääretön	ääretön	ääretön	ääretön
1	ääretön	ääretön	ääretön	ääretön
1	2	ääretön	ääretön	ääretön
1	2	3	ääretön	ääretön
1	2	3	4	21

30.3.2001

Huonot uutiset etenevät hitaasti:



Toimiva linkki katkeaa välillä AB:

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

30.3.2001

22

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 esim. Dijkstran algoritmilla

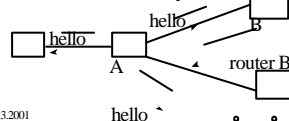
30.3.2001

23

Naapurien löytäminen

- **reititin lähettää jokaiseen kaksipisteyhteyteen HELLO-paketin**
- **linjan toisessa päässä oleva reititin vastaa ja lähettää nimensä**

- router ID
- nimien oltava yksikäsitteisiä koko verkossa



30.3.2001

24

Etäisyyden mittaaminen

- **kaikille naapureille ECHO-paketti**
 - vastaanottajan palautettava paketti välittömästi
- **=> kiertoviive (round-trip-time)**
 - dynaaminen etäisyyssmitta
- **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

30.3.2001

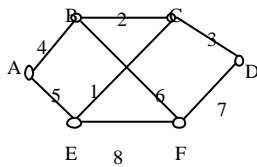
25

Tietopaketin kokoaminen

- **muodostus**
 - tietyin aikavälein
 - kun muutoksia havaittu
- **sisältö**
 - reitittimen tunnus
 - paketin järjestysnumero
 - paketin ikä
 - etäisyydet kuhunkin reitittimen naapuriin

30.3.2001

26



B
seq
age
A 4
C 2
F 6

Tietopaketin 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

30.3.2001

28

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

30.3.2001

29

elinaikalaskuri (TTL-laskuri)

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

30.3.2001

30

Lisäparannuksia

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

30.3.2001

31

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

30.3.2001

32

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

30.3.2001

33

Käyttö

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

30.3.2001

34

Hierarkkinen reititys

- **reitityksen skaalautuvuus**
 - isossa verkossa runsaasti reitittimiä (Internet: miljoonia)
 - reititystaulut suuria
 - reittien laskeminen raskasta
 - tietopaketit kuluttavat linjakapasiteettia
- **hallinta-autonomia => autonominen järjestelmä AS**
 - organisaatio päättää omista asioistaan
 - myös reitityksestä
 - oma sisäinen reititystapa

30.3.2001

35

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

30.3.2001

36

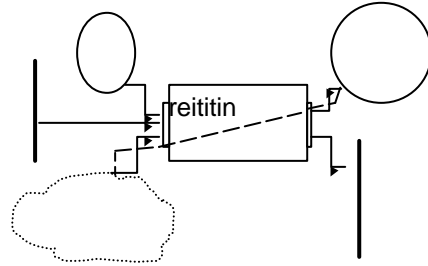
Hierarkkisen reitityksen ongelmat

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

30.3.2001

37

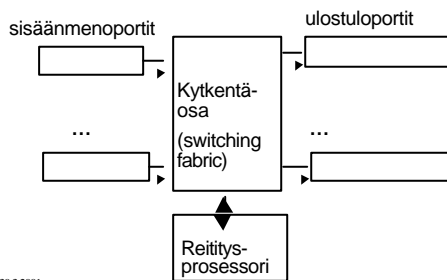
Reititin (Router)



30.3.2001

38

Reitittimen rakenne

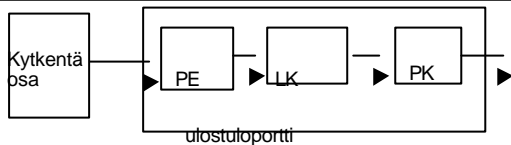
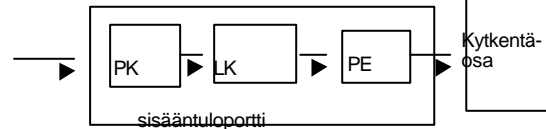


30.3.2001

39

• Portit

- **peruserroksen toiminnot (PK)**
 - fyysisen siirtoyhteyden pää
- **linkkikerroksen toiminnot (LK)**
 - virhetarkistukset, vuonvalvonta,
 - MAC-kerroksen toiminnot
- **pakettien edelleenohjaaminen (PE)**
 - datapaketit kytentäverkon kautta oikeaan ulostuloporttiin
 - valvontapakettit (RIP, OSPF, BGP) reititysprosessorille



Vastaavasti kukin ulostuloportti tallettaa sen kautta eteenpäin lähtevät paketit ja suorittaa niille linkkikerroksen ja peruserroksen vaatimat toimenpiteet.

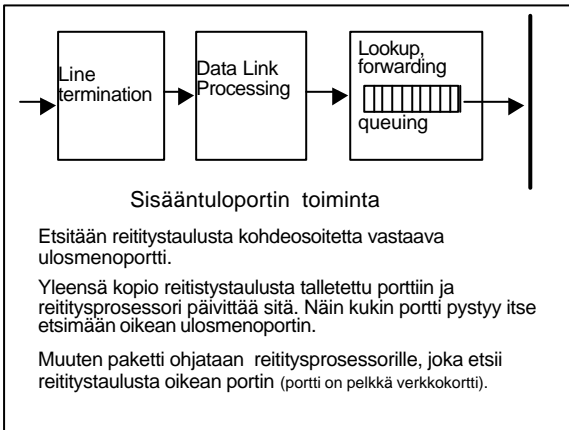
Käytännössä useita portteja on yhdistetty yhdeksi linjakortiksi (line card) reitittimen sisällä.

• **Reititysprosessori**

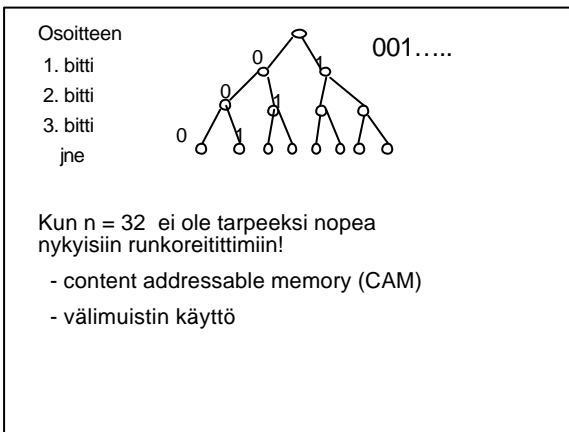
- suorittaa reititysprotokollaa
- RIP, OSPF, BGP, ..
- päivittää reititystauluja
- hallinta- ja ylläpitotoimintoja

• **Kytentäosa**

- yhdistää paketin sisääntuloportit ulostuloportteihin
- paketti siirtyy oikeaan verkkoon
- täysin reitittimen sisällä

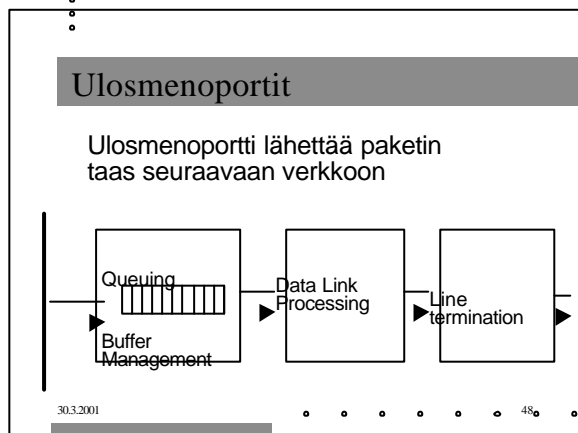


- **Runkolinjareitittimiltä vaaditaan hyvin suuria nopeuksia**
 - miljoonia hakuja sekunnissa
 - pitäisi pystyä toimimaan linjan nopeudella
 - OC48-linkki => 2.5 Gbps
 - jos paketin koko 256 tavua => noin miljoona hakua sekunnissa
- **erilaisia tekniikoita**
 - talletetaan reititaulun alkioit puurakenteina



- ### Kytkeäosa
- **Kytkeä muistin kautta**
 - portit tavallisia käyttöjärjestelmän I/O-laitteita
 - keskeytys ilmoittaa paketin saapumisesta
 - CPU kopioi paketin sisääntuloportista muistiin
 - CPU tutkii osoitteen ja reititystaulusta etsii vastaavan ulosmenoportin
 - CPU kopioi paketin muistista tähän ulosmenoporttiin
 - muistin saant nopeus rajoittaa toimintaa
 - **nykyiset reitittimet**
 - käyttävät linjakortin omia prosessoreita
- 30.3.2001 • Memory shared multiprocessors 46

- **Kytkeä väylän kautta**
 - sisääntuloportit siirtävät paketin väylän kautta suoraan oikeaan ulosmenoporttiin
 - vain yksi paketti kerrallaan voi kulkea väylässä
 - jos väylä on varattu, paketti joutuu odottamaan
 - väylän nopeus rajoittaa kytkentänopeutta
 - Gbps nopeudet riittävät LANeille ja yritysverkoilla
- **Kytkeä kytkentäverkon kautta**
 - ristikkäinkytkin (crossbar switch)
 - $2N$ väylää, jotka yhdistävät N sisääntuloporttia N :ään ulosmenoporttiin
 - voivat tukkeutua => odotusta sisäänmenoportissa
 - Cisco 12000: 64 Gbps

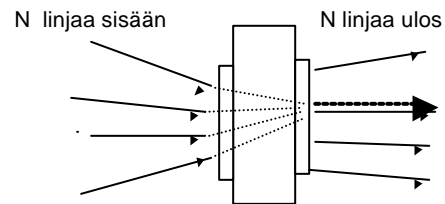


Jonotus reitittimessä

- **Sekä sisäänmeno- että ulostuloporttiin voi syntyä jonoa**
 - näissä jonoissa reititin voi kadottaa paketteja, kun puskuritila ei enää riitä
 - se kummassa jonossa paketit katoavat, riippuu kytkimen ja linjan nopeuden suhteista
 - jonoa voi syntyä myös, koska useasta lähteestä pyritään samaan kohteeseen

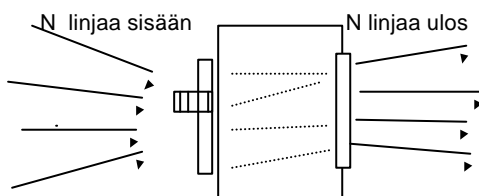
30.3.2001

49



Kytkin toimii riittävällä nopeudella, joten sisääntulossa ei tarvitse jonottaa.

Yhdelle linjalle liian paljon liikennettä => ulosmenoportin puskuritila täyttyy ja paketteja katoaa!



Jos kytkin ei toimi tarpeeksi nopeasti, sisääntuloportteihin syntyy jonoja.

Esim. Ristikkäinkytkimessä paketti joutuu odottamaan, jos samaan kohteeseen on menossa useita paketteja. Jonottava paketti voi tukkia tien myös muilta saman portin paketeilta, jotka muuten voisivat edetä kytkimessä.

(head-of-the-line-blocking)

5.4 Internetworking

- **verkot erilaisia: nyt ja aina**
 - palvelu: yhteydellinen / yhteydetön
 - osoittaminen: yksitasoinen / hierarkkinen
 - monilähetys/yleislähetys
 - paketin koko
 - toiminnot :
 - palvelulaatu (qos) , virheiden käsittely, vuonvalvonta, ruuhkanvalvonta, turvaus ja laskutus
 - protokolla

30.3.2001

52

ongelmana on erilaisten toiminnallisuuden yhteensopivuus

- luotettavuus
- ruuhkan valvonta
- kuittaukset
- toimitusaikatakuut

30.3.2001

53

Yhteydettömien verkkojen yhdistäminen

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

30.3.2001

54

Pakettien paloittelu (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)**

30.3.2001

55

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 kyttävä kokoamaan paketteja

30.3.2001

56

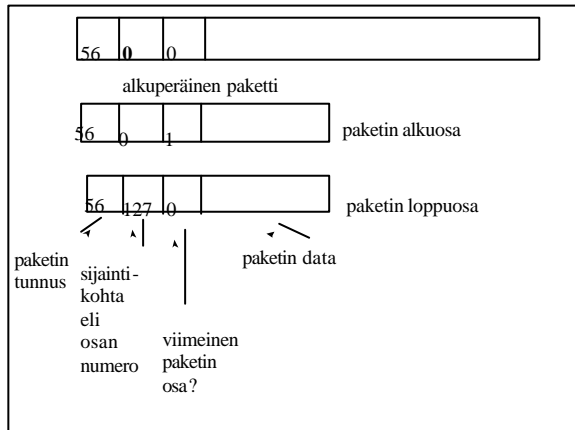
Pakettien kokoaminen

- **edellyttää palojen 'numerointia'**
 - on tiedettävä, mikä paketti 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**

30.3.2001

• tai tiedettävä paketin pituus

57

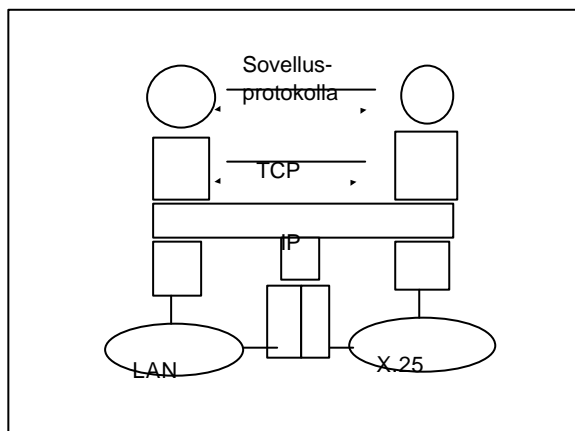


5.5 Internetin verkkokerros

- **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 (datasähke, datagram) lähteestä kohteeseen yli kaikkien välissä olevien erilaisten verkkojen

30.3.2001

59



5.5.1 IP-protokolla

• IP-datasähke

- otsake
- dataosa

• otsake

- 20 tavun kiinteä osa
 - tunnistiedot, pituustiedot, tarkistusbitit (-summa)
 - osoitteet, minkä kuljetusprotokollan sanoma
 - liian pitkän paketin paloittelu ja kokoaminen
 - erilaisen palvelun tarjoaminen eri sovelluksille
- vaihtelevan mittainen valinnainen osuus
 - lisäoptioita

30.3.2001

61

Versio	IHL	TOS	Datasähkeen pituus (tavuja)	
Tunniste		Flag	Siirtymä	
Elinaika	Protokolla	otsakkeen tarkistussumma		
Lähtäjän IP-osoite				
Vastaanottajan IP-osoite				
Optiot (jos on käytössä)				
data				

IPv4 - datasähke

IP-otsakkeen kentät

• Versio IPv4 (IPv6)

• 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 (käytössä uusissa Cisco-reitittimissä)

30.3.2001

63

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

30.3.2001

64

IP-otsakkeen kentät jatkuvat

• Datagram length

- koko datasähkeen pituus
- maksimi 65535 tavua
 - maksimipituus vielä riittävä, mutta tulevaisuuden nopeille verkoille jo ongelma
- yleensä koko 576 -1500 taavua

• Identification

- datasähkeen numero
- kaikissa saman datasähkeen osissa sama tunnus

30.3.2001

65

IP-otsakkeen kentät jatkuvat: liput

• 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

30.3.2001

66

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**

30.3.2001

67

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

30.3.2001

68

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-osoite

30.3.2001

69

IP-otsakkeen kentät jatkuvat

- **Options**

- vaihtelevan mittaisia
 - 1. tavu kertoo option koodin
 - voi seurata pituuskenttä
 - datakenttiä
 - täytetty jotta 4 tavun monikertoja
- käytössä 5 optiota
 - mutta reitittimet eivät välttämättä ymmärrä

30.3.2001

70

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

30.3.2001

4.4. IP-osoitteet

- **jokaisella verkon isäntäkoneella ja reitittimellä on oma yksikäsitteinen osoite muotoa**

- verkon numero
- isäntäkoneen (liitännäkortin) numero

- **osoite on 32-bittinen**

- osoitteen luokasta riippuen bitit jaetaan verkon numeroon ja isäntäkoneen numeroon eri tavoin

- **osoitteet palvelun tarjoajille jakaa ICANN**
(The Internet Corporation for Assigned Names and Numbers)

- nämä puolestaan jakavat muille

30.3.2001

72

osoitteet merkitään yleensä desimaalimuodossa

- kukin osoitteen neljästä tavusta kirjoitetaan desimaalilukuna (0-255)
 - luvut erotetaan pisteellä
- esim.
- heksadesimaaliosoite C0 29 06 14 on 192.41.6.20 eli C0 => 192, 29 => 41, 06 => 6, 14 => 20
- pienin osoite on 0.0.0.0 ja suurin 255.255.255.255

30.3.2001

73



IP-osoitteiden muodot

(alkuperäinen luokallinen osoitus)

IP-osoitteiden luokat

- **A-luokka hyvin isoille verkoille**
 - 7 bittiä verkko-osoitteeseen, 24 bittiä isäntäkoneille
 - **126 verkkoa, 16 miljoonaa konetta/verkko**
- **B-luokka keskikokoisille verkoille**
 - 14 bittiä verkoille, 16 bittiä koneille
 - **16382 verkkoa, 65528 konetta/verkko**C-luokka
- **C-luokka pienille verkoille**
 - 21 bittiä verkoille, 8 bittiä verkon koneille
 - **noin 2 miljoonaa verkkoa, 254 konetta/verkko**

30.3.2001

75

Osoiteluokkien ongelmia

- **verkon kasvu => ongelmia**
 - C-luokan verkossa max 256 osoitetta
 - liian vähän useimmille yrityksille => tarvitsevat B - luokan osoitteen tai monta C-luokan verkko-osoitetta
 - B-luokan verkkoja liian vähän (max 16382) ja niissä liian paljon osoitteita (max 65536)
 - 100000 verkkoa jo 1996!
 - useassa B-verkossa alle 50 konetta
- **=> B-luokan osoitteita tuhlaantuu ja osoitteista pulaa**

30.3.2001

76

CIDR (Classless InterDomain Routing)

- verkko-osa voi olla minkä tahansa kokoinen (ei vain 8,16,24 bittiä)
- a.b.c.d/x, jossa x ilmoittaa verkko-osan bittien lukumäärän
 - esim. yritykselle, jolla 2000 konetta varataan 2048 = 2^{11} koneosoitetta, jolloin verkko-osaa varten jää 21 bittiä
 - C-luokan verkkoja
 - yritys voi itse vielä jakaa koneosoitteen 11 bittiä aliverkko-osoitteeksi ja koneosoitteeksi

30.3.2001

77

CIDR-idea jatkuu

- **jaetaan osoitteet neljään osaan, kukin osa varataan yhdelle maanosalle (Eurooppa, Pohjois-Amerikka, Etelä-Amerikka, Aasia+Pasific)**
 - kullekin noin 32 miljoonaa osoitetta
 - 320 miljoonaa jää vielä varastoon
- **reititetään myös maanosien mukaan**
 - osoitteet 194.0.0.0 - 195.255.255.255 Eurooppaan
- **=> pienemmät reititustaulut**

30.3.2001

78

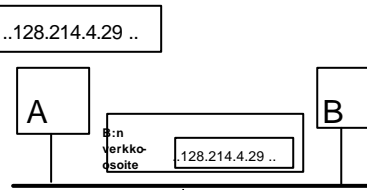
Muita Internet-protokollia

- **ICMP (Internet Control Message Protocol)**
 - verkon koneiden (reitittimien ja isäntäkoneiden) kommunikointiin esim. virhetilanteissa
- **ARP (Address Resolution Protocol)**
 - protokolla lähiverkon koneen verkko-osoitteen selvittämiseksi
- **OSPF (Open Shortest Path First)**
 - linkkitilareititykseen perustuva reititysprotokolla
- **BGP (Border Gateway Protocol)**
 - eri alueiden välinen reititysprotokolla
- **IPv6**
 - uudempi versio IP-protokollasta
- **Näitä käsitellään Tietoliikenne II -kurssilla**

30.3.2001

79

IP-paketissa on vain vastaanottajan IP-osoite



Pitää saada selville IP-osoitetta vastaava verkko-osoite.

Yleislähetyksenä kysely: "Kenen IP-osoite?"

Jokaisella koneella oma ethernet-osoite (48 bittä), jota käytetään MAC-kehyksessä