



# Luento 7: Verkkokerros tehtävät, reititin ja IP-protokolla

**Ma 18.11.2013**

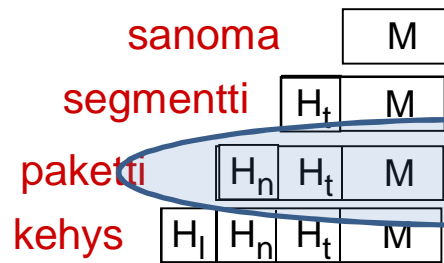
**Tiina Niklander**

Kurose&Ross  
Ch4.1-4.5

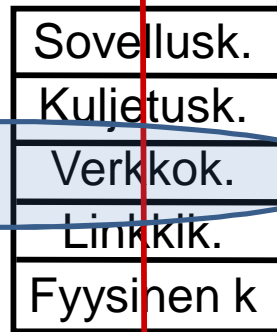
Pääasiallisesti kuvien  
© J.F Kurose and K.W. Ross,  
All Rights Reserved

# Luennon sisältöä

Lähettäjä (sender)



message,  
segment  
datagram  
frame

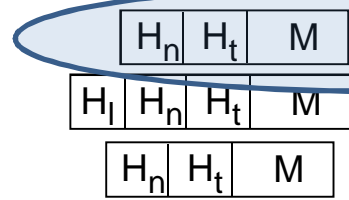
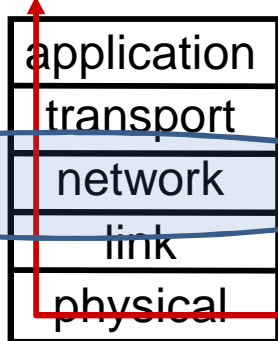
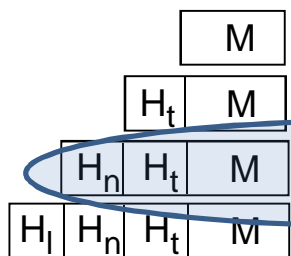


kytkin



Fig 1.24 [KR12]

Vastaanottaja (recipient)



reititin



# Sisältöä

**Verkkokerros**

**Reititin**

**IP-protokolla**

IP-osoitteet, DHCP, NAT

Reititysalgoritmit



Oppimistavoitteet:

- Osaa selittää, kuinka IP-paketteja välitetään verkossa
- Tietää, mitä tietoja sisältyy IP-pakettiin (ja miksi)
- Osaa selittää reitittimen rakenteen ja toiminnan
- Osaa kuvailla, kuinka reitittimet kokoavat reititystietonsa  
= linkkitila- ja etäisyysvektorialgoritmien toimintaideat



# Verkkokerros

## Verkkokerroksen tehtävät



# Verkkokerros: Toimittaa kuljetuskerroksen segmentit vastaanottajalle

## Lähetys

- Luo kuljetuskerroksen segmenteistä verkkokerroksen IP-paketteja
- Lisää otsaketietoja: mm. IP-osoitteet

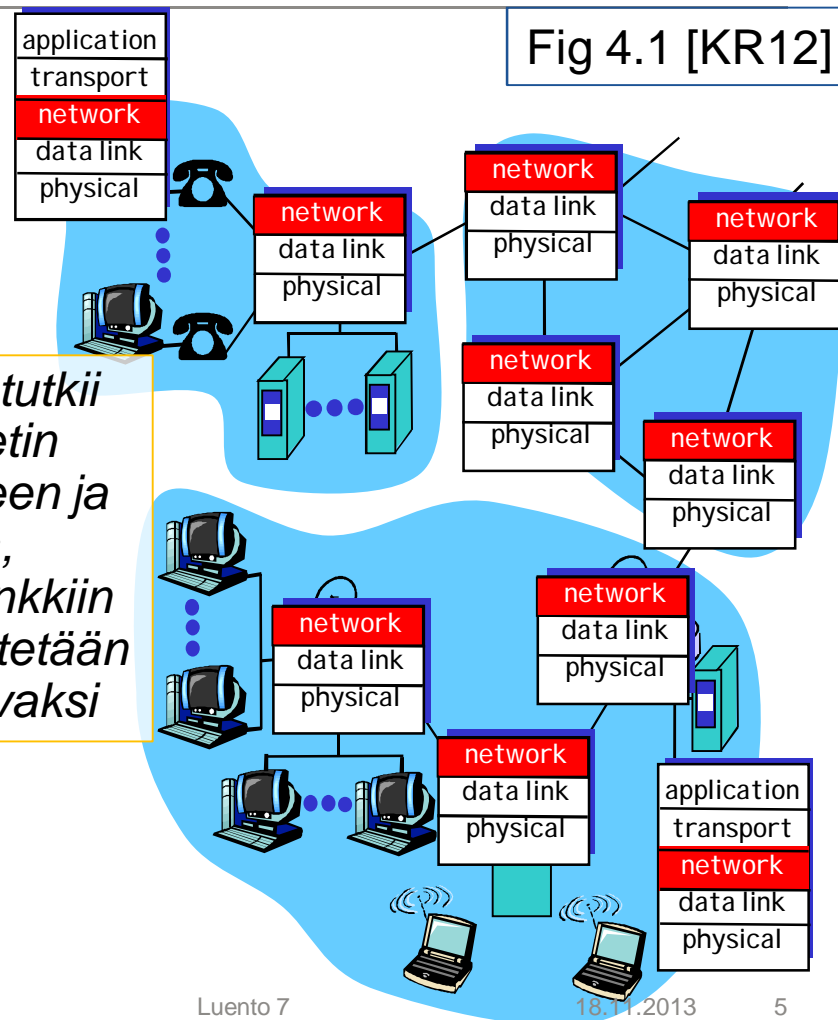
## Pakettien kulku verkossa

- Isäntä (lähde) – reititin - ...- reititin – isäntä (kohde)

## Vastaanotto

- Poista otsake
- Anna segmentti kuljetuskerrokselle

*Reititin tutkii IP-paketin otsakkeen ja päättää, mihin linkkiin se lähetetään seuraavaksi*

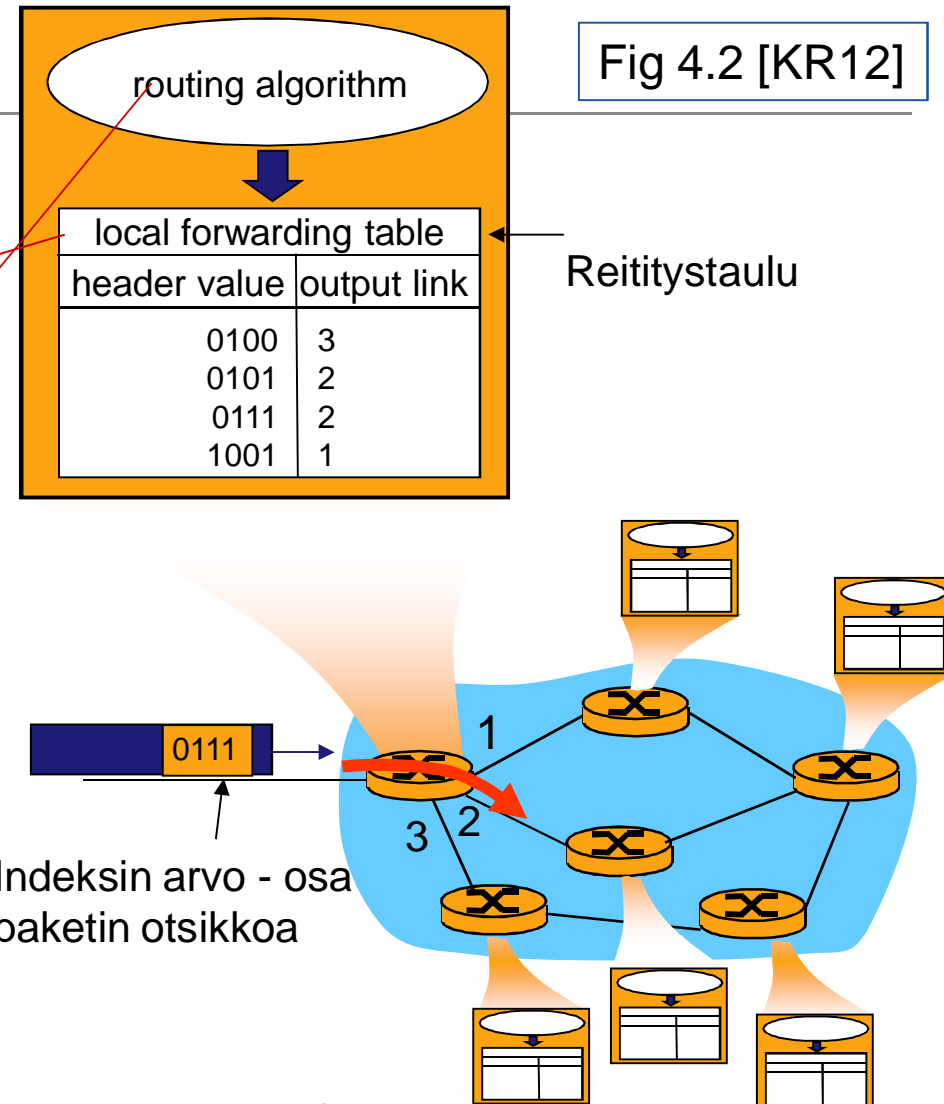




# Paketin välitys ja reititys

Reitittimen kaksi tehtävää:

- Paketin välitys (forwarding)
  - Paikallinen päätös
  - mihin ulosmenolinkkiin paketti lähetetään
  - katsoo linkin reititystaulusta
- Reititys (routing)
  - Globaali päätös
  - mitä reittiä paketti kulkee lähettäjältä vastaanottajalle
  - reititystaulun ylläpito





# Miksi verkkokerros?

---

Internet koostuu hyvin heterogeenisista verkoista

- Verkot toteutettu eri teknologioilla
- Verkoilla kehyksen (frame) maksimikoko erilainen
- Palvelu: yhteydellinen / yhteydetön,
- Erilaisia osoittamistapoja: yksitasoinen/hierarkkinen
- Monilähetys / yleislähetys
- Toiminnot: virheenkäsittely, vuonvalvonta, ruuhkanvalvonta, yhteyden laatu / laatutakuu (QoS), turvaus, laskutus, ..



# Verkkoprotokolla: IP

”Everything over IP, IP over everything”

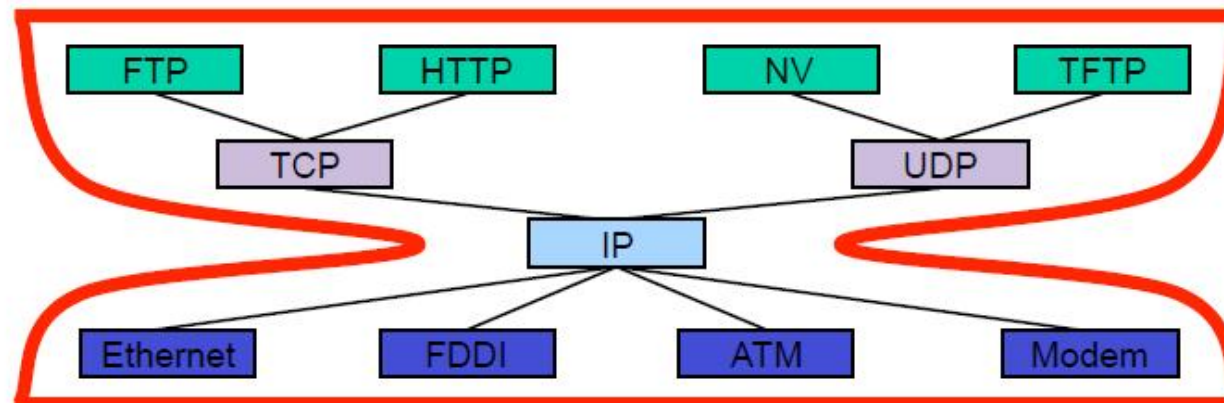
Verkkoprotokolla IP (Internet Protocol) on verkkokerroksen yhteinen kieli

Internetin isäntäkoneiden ja reitittimien kommunikointitapa

- “Kullakin omat murteensa ja tapansa, mutta kaikki osaavat IP:tä.”

Yhteinen osoitustapa: IP-osoite

Yksikäsitteiset osoitteet



Prof. B. Godfrey,  
4.3.2011,  
luentokalvo,  
Univ. Illinois





# Verkon palvelun laatu

Table 4.2 [KR12]

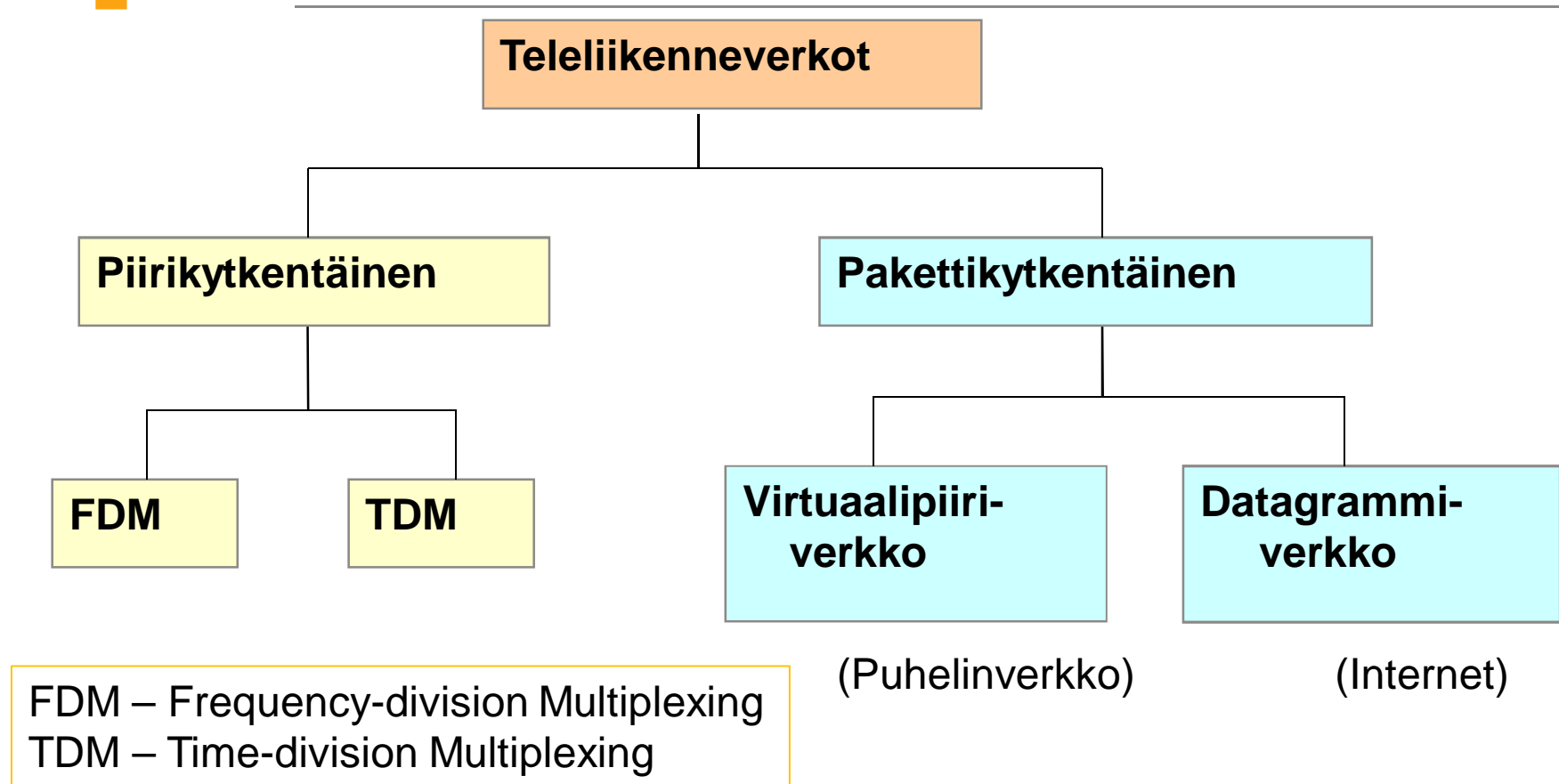
Network Architecture	Service Model	Guarantees ?			Congestion feedback
		Bandwidth	Loss	Order Timing	
Internet	best effort	none	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	yes

ATM – Asynchronous Transfer Mode  
 CBR – Constant Bit Rate  
 ABR – Available Bit Rate

(ATM-verkko on puhelinverkon ytimessä)



# Verkkojen taksonomia





# Pakettikytkentäinen verkko

---

Joko datagrammiverkkona tai virtuaalipiiriverkkona

Sanoman jokainen paketti reititetään erikseen

- kohteen IP-osoitteen perusteella

*“Tyhmä verkkokerros”*: vain pakettien välitys koneelta koneelle

*“Fiksut isäntäkoneet”*: virheenvälitys, vuonvalvonta, järjestys

Kaikki sanoman paketit samaa reittiä

- linkkiin liitetyn virtuaalipiirinumeron perusteella

Signaalointiprotokolla: yhteydenmuodostus, ylläpito, purku

- yhteyden tietoja reitittimessä (virtuaalipiirin muunnostaulukko)
- mahd. myös kaistanvarausta

*Fiksu verkkokerros*: vuonvalvonta, virhevalvonta, järjestys

*Tyhmät isäntäkoneet*: vrt. Puhelin

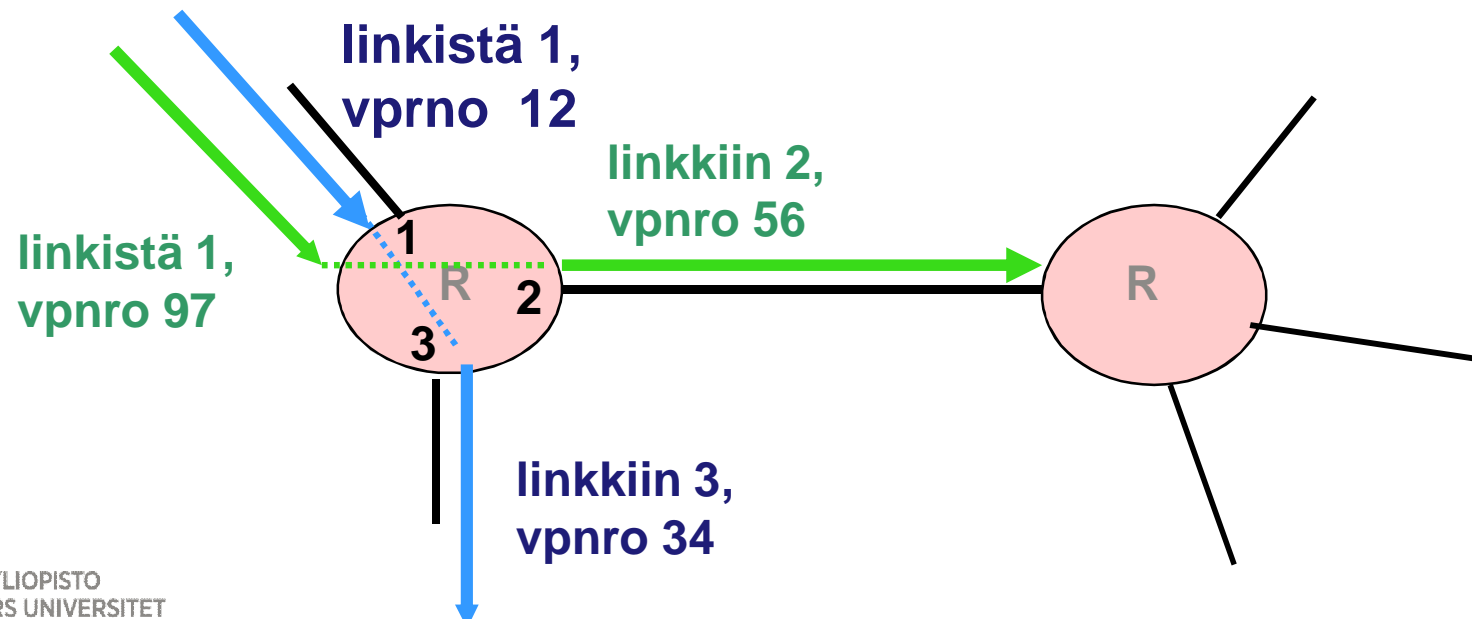
ATM-verkot (Asynchronous Transfer Mode), X.25-verkot



# Reititys: Virtuaalipiiriverkko

1. paketti muodostaa reitin, muut paketit kulkevat samaa reittiä  
otsakkeessa kohdeosoitteen lisäksi virtuaalipiirinumero (vpnro)  
reititin ylläpitää tietoa piirinumeroista ('hajujälki')

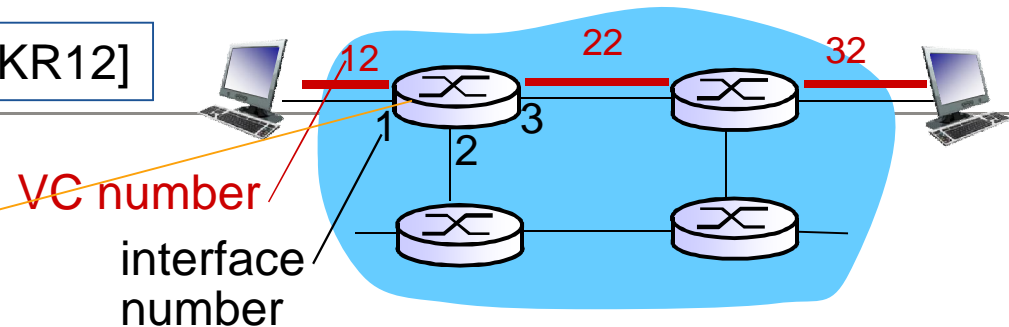
Reititys = selvittää vpnro:a vastaava linkki, välittää paketti linkille





# Virtuaalipiirin muunnostaulukko

Fig 4.3 [KR12]



*Reitittimen muunnostaulukko:*

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

**Taulukkoa päivitettävä aina kun uusi yhteys on muodostettu tai vanha purettu!  
Pakettivälitystä: Ylläpitää kyllä tilatietoja yhteydestä (=vpnro),  
mutta ei varaa resursseja etukäteen!**

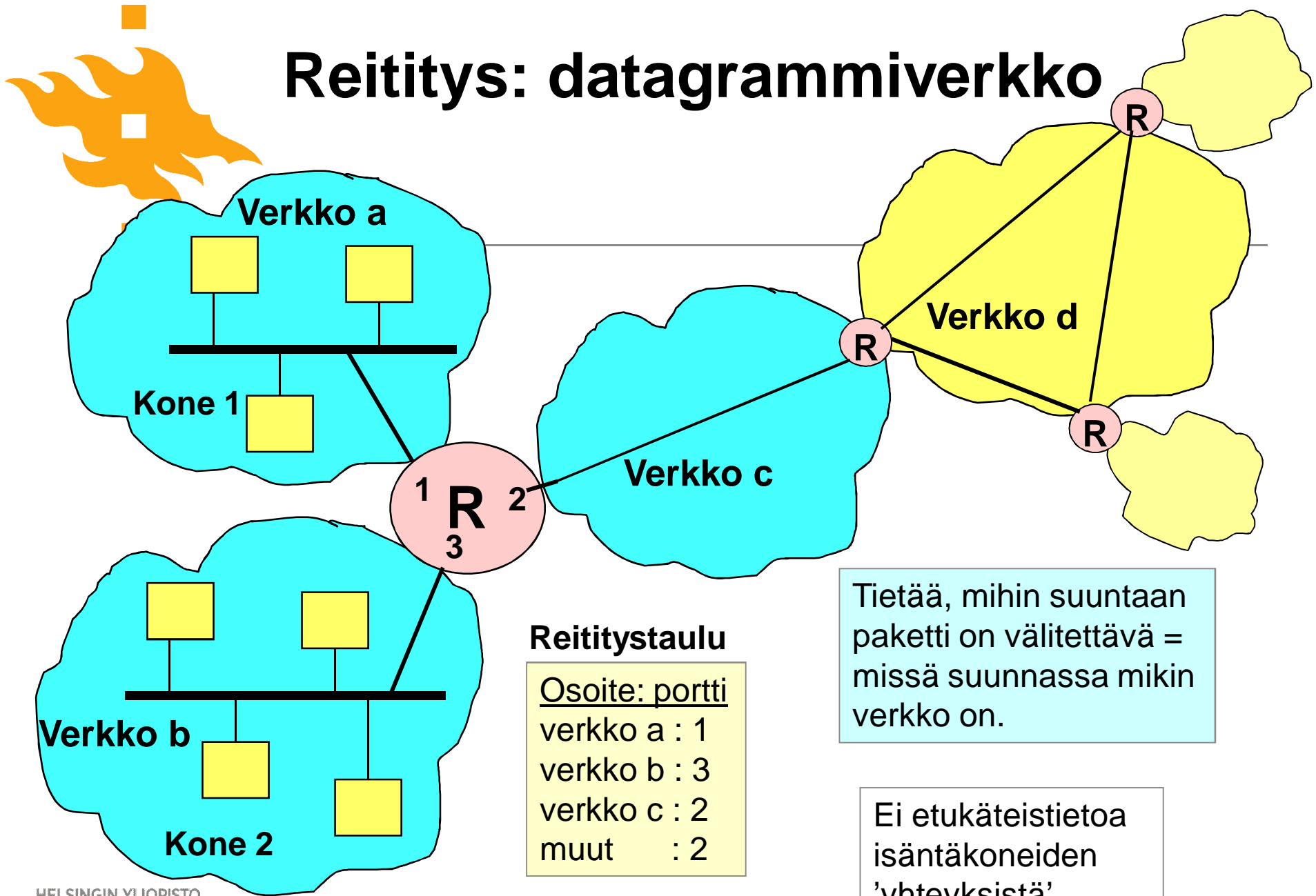


# Virtuaalipiirin muunnostaulukko

---

- Virtuaalisyhteyden joka linkillä omat VP-numerot
  - reititin antaa VP-numerot
- *Miksi ei käytetä koko yhteydellä samaa VP-numeroa?*
  - Tarvittaisiin paljon enemmän numeroita!
  - Nyt riittää pienempi numeroavaruus =>
  - tarvitaan pienempi kenttä numeroa varten
    - 0-255 => riittää 8 bittiä
    - 0-4095 => tarvitaan 12 bittiä
  - Yhteisestä, koko verkon läpikäyvästä numeroinnista sopiminen on isossa verkossa lähes mahdoton tehtävä!

# Reititys: datagrammiverkko



## Reititystaulu

Osoite: portti	
verkko a :	1
verkko b :	3
verkko c :	2
muut :	2

Tietää, mihin suuntaan paketti on välitettävä = missä suunnassa mikin verkko on.

Ei etukäteistietoa isäntäkoneiden 'yhteyksistä'



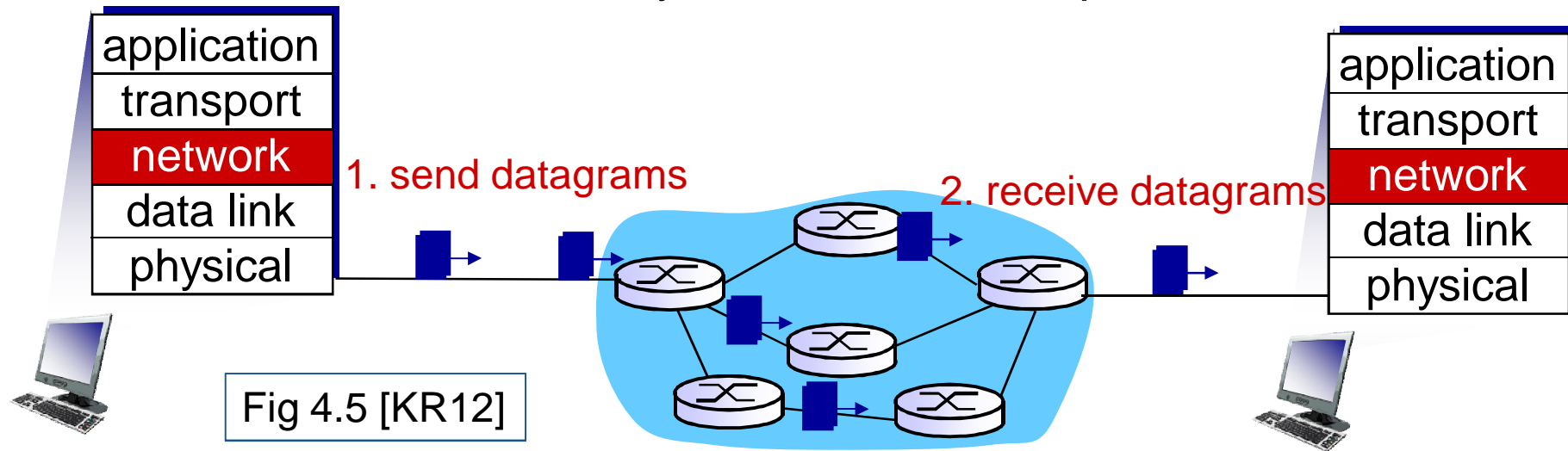
# Datagrammiverkko

Ei yhteyden muodostusta erikseen

Reitittimet: ei tietoa päästä-päähän yhteyksistä

Ei käsitettä 'yhteys' (connection) verkkokerroksella

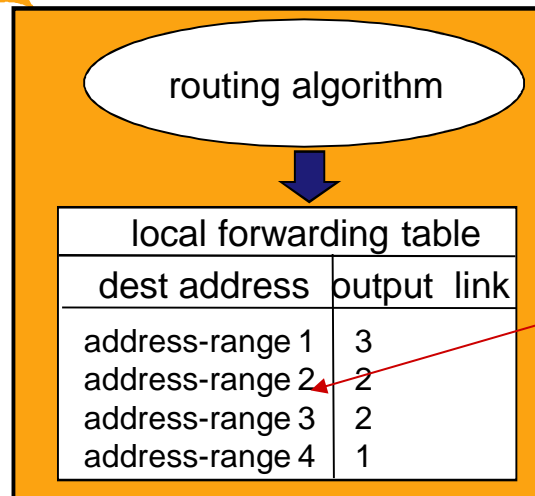
Pakettien välitys kohdeosoitteen perusteella







# Datagrammiverkon muunnostaulukko



4 miljardia IPv4-osoitetta, yksittäisten osoitteiden sijasta osoitevälejä (aggregate table entries)

Ipv6: n.  $3 \cdot 10^{38}$  osoitetta

IP destination address in arriving packet's header

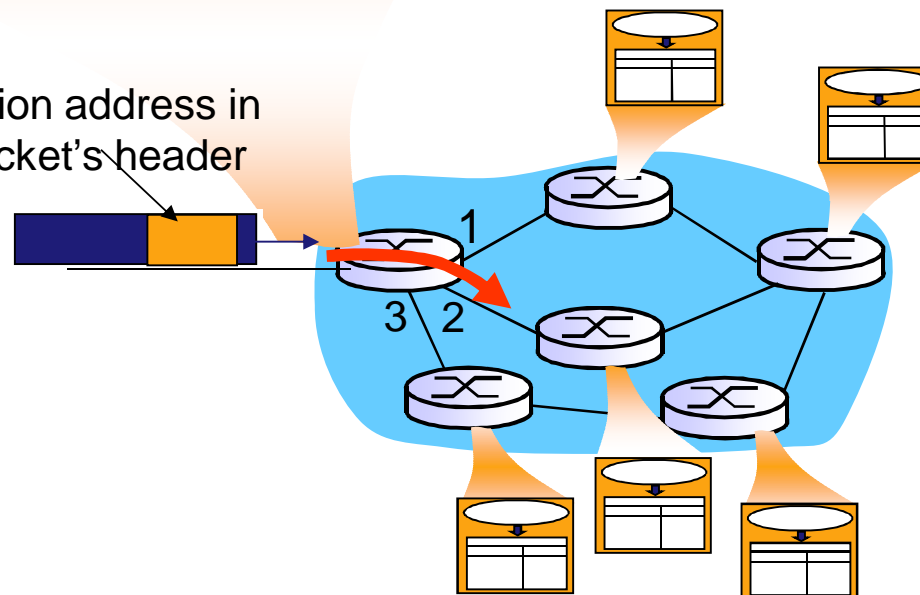


Fig 4.2 [KR12]



# Datagrammiverkon muunnostaulukko

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3



# Pisin alkuosa (longest prefix matching)

Muunnostaulukosta etsitään osumaa. Jos useita, niin käytetään sitä, jolla on **pisin** yhteinen osoitteen **alkuosa**.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

esimerkki:

DA: 11001000 00010111 00010**110** 10100001

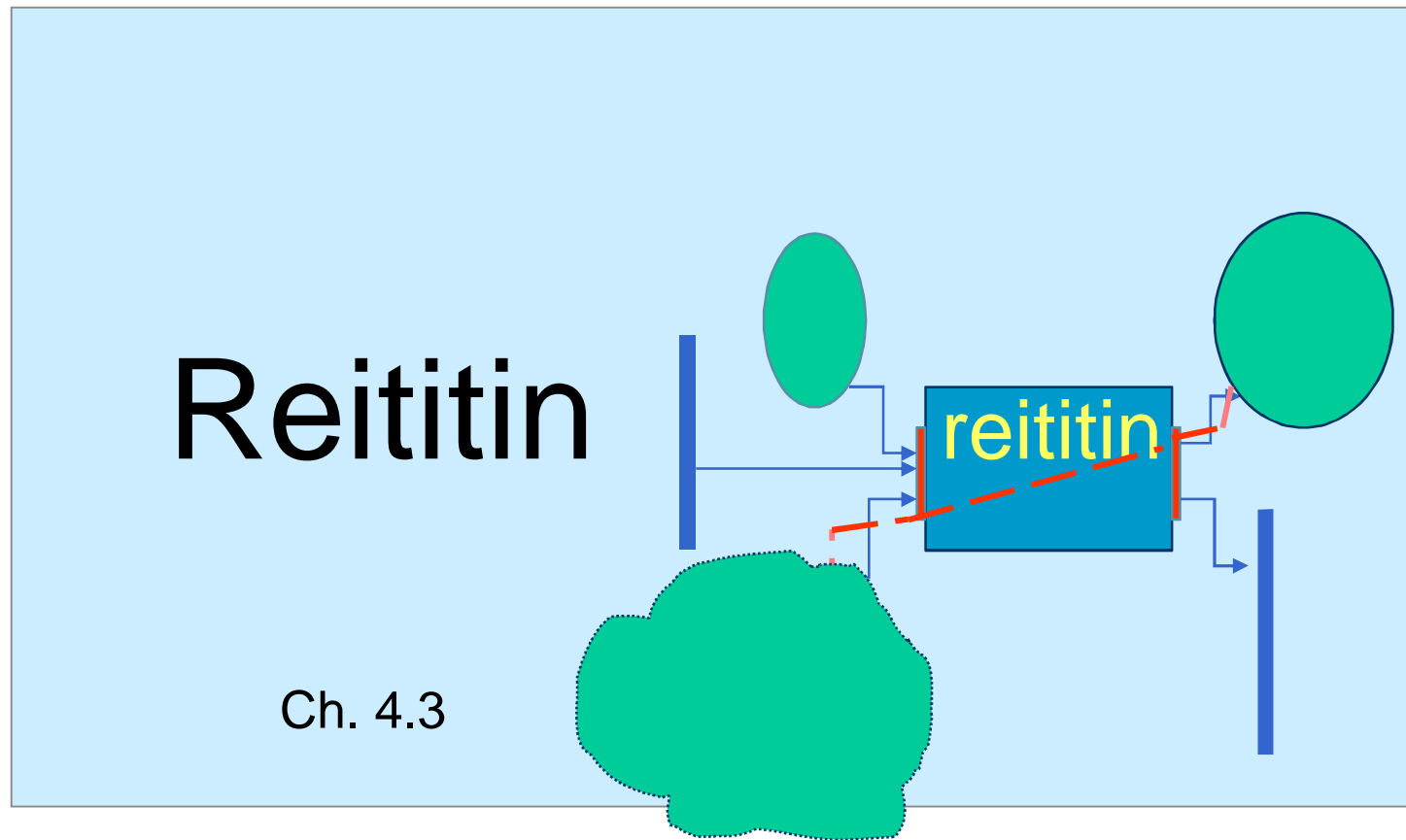
Mihin linkkiin?

DA: 11001000 00010111 00011000 **10101010**

Mihin linkkiin?



# Verkkokerros





# Verkkokerros ~ reitittäminen

## Reititin (router)

Osoite muunnokset siihen  
kytkettyjen  
teknologioiden välillä

Sisääntulolinkki ja  
ulosmenolinkki voivat  
olla *eri teknologiaa*

Välittää **verkkokerroksen**  
otsakkeen perusteella  
(IP-osoite)

Laitteistotoimintona tai osin  
ohjelmallisesti

## Kytkin (switch)

Sekä sisääntulolinkki että  
ulosmenolinkki ovat  
*samaa teknologiaa*

Lähiverkon sisällä välitys  
**linkkikerroksen**  
otsakkeen perusteella

Poikkeuksetta aina  
laitetason toimintona

## Keskitin (hub)

Perustuu toistimiin  
**Fyysisellä kerrokselle**



# Reitittimen arkkitehtuuri

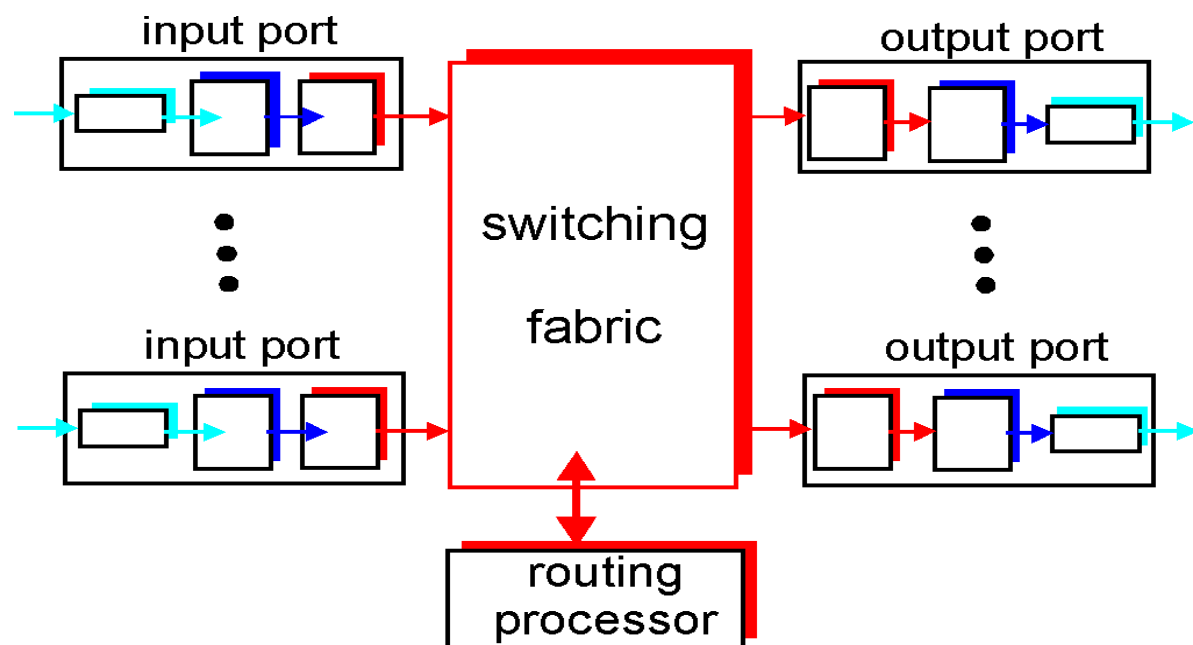
Fig 4.6 [KR12]

## Kaksi tehtävää

- Välitä paketteja tulolinkeistä ulosmenolinkkeihin
- Suorita reititys algoritmia / -protokollaa

## Portti ~ verkkokortti

Useita portteja  
niputettu yhteen  
linjakortiksi (line  
card)

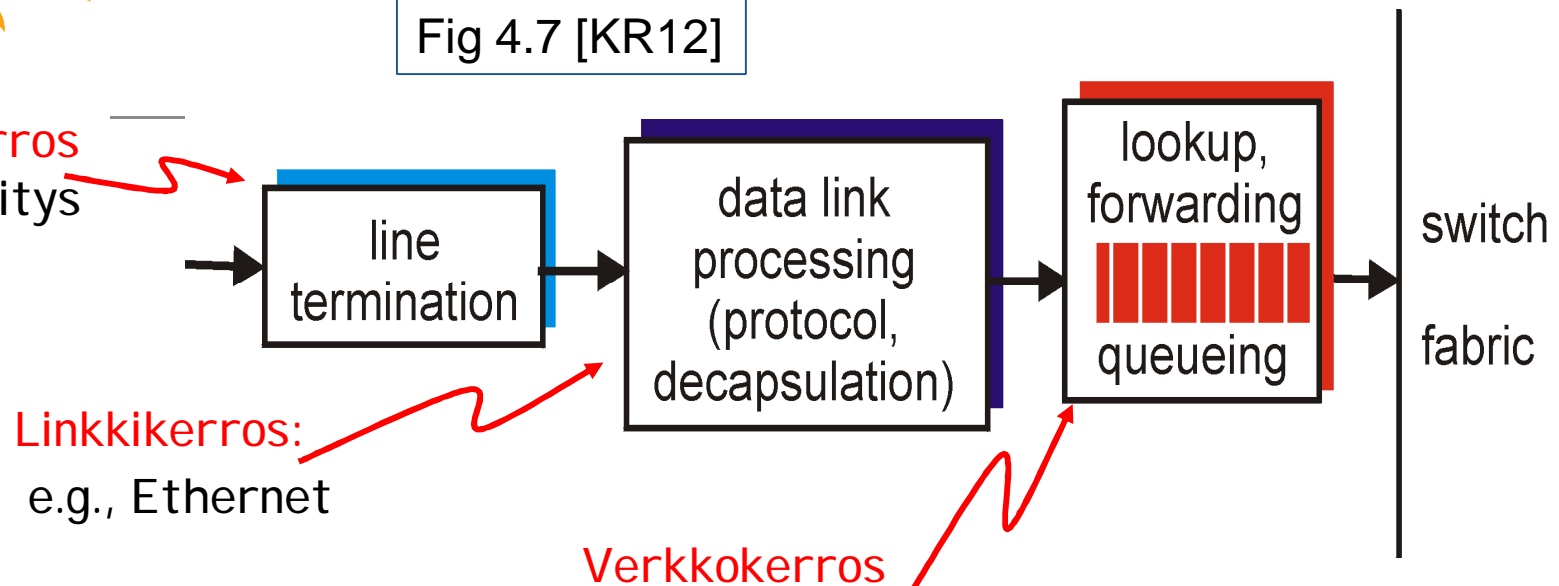




# Sisääntuloportti (input port)

Fig 4.7 [KR12]

Fyysinen kerros  
bittitason esitys



Tavoite: paketti ulos sisääntulon nopeudella

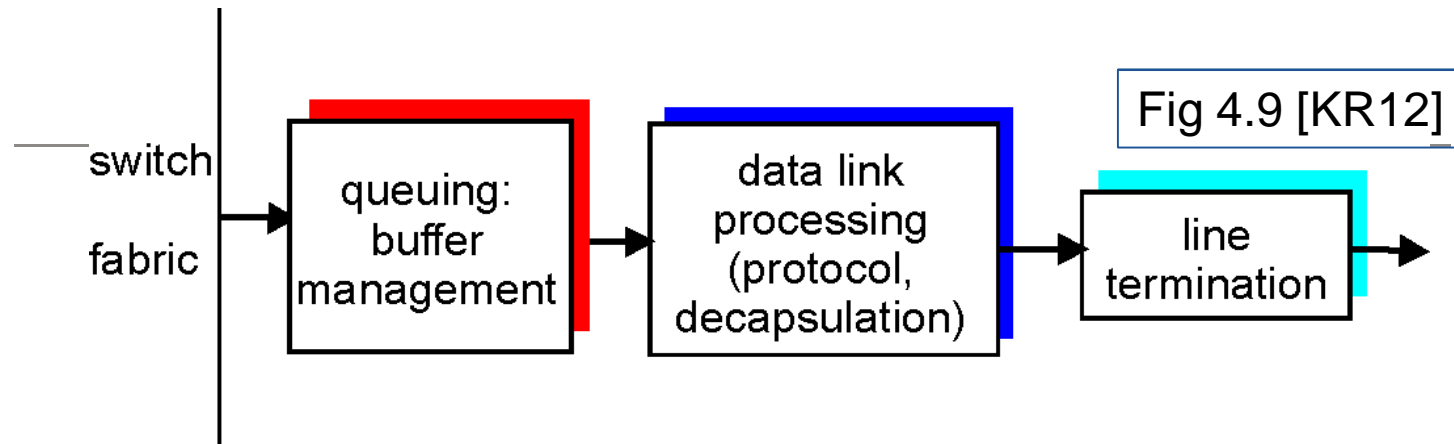
Jonotus: jos ulosmeno hitaampi kuin sisääntulo tai joku muu siirtää samaan ulostuloon

myös HOL (head-of-line blocking) – sisääntuleva jää jonon tulpaksi, kun se ulosmenojono on käytössä

Jos linjanopeus 2.5 Gbps ja paketin koko 256 tavua => 1.2 miljoonaa pakettia sekunnissa!



# Ulosmenoportti (output port)



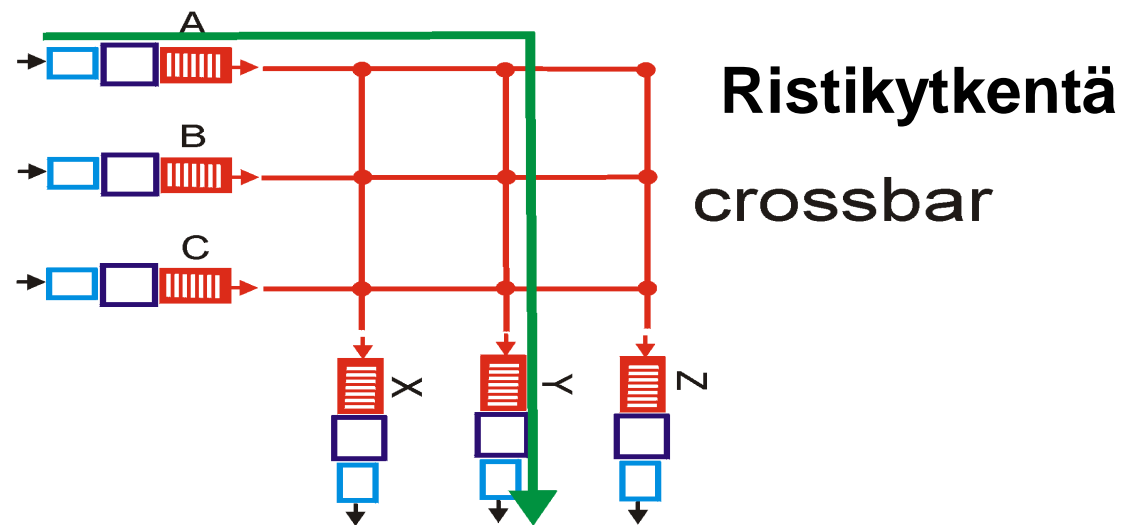
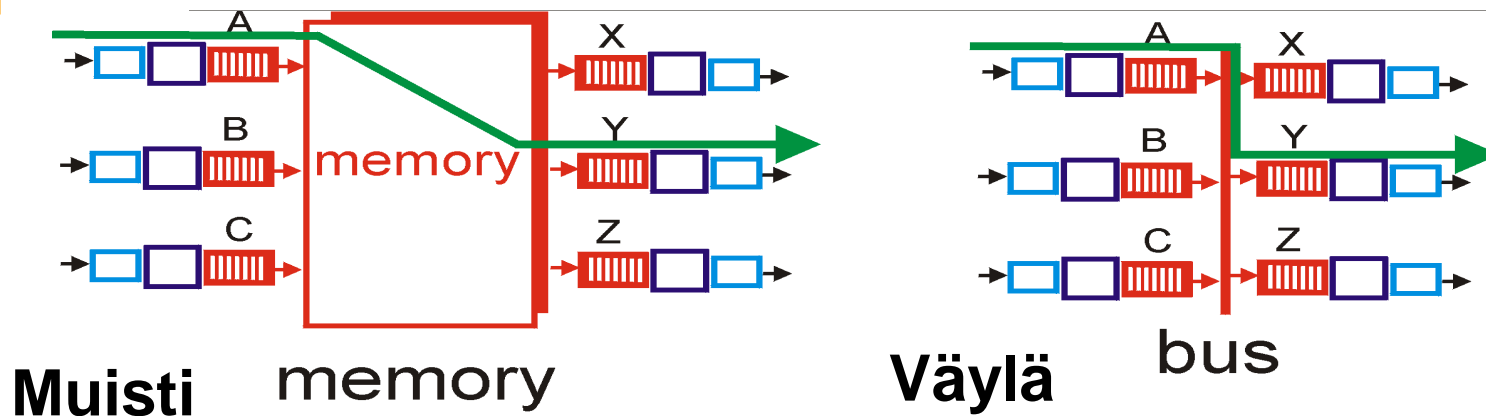
- **Puskuroi**, jos paketteja tulee nopeammin kuin ulosmenon siirtonopeus sallii
  - Sisääntulo nopeampi tai monesta samaan kohteeseen
- Voi käyttää priorisointia (packet scheduling)
  - FCFS (First Come First Serviced), WFQ (Weighted Fair Queueing),...
  - QoS (Quality of Service) (ei käsitellä tällä kurssilla!)
- Suorittaa linkki- ja fyysisen kerroksen operaatiot





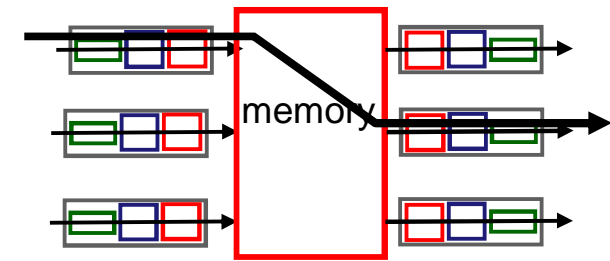
# Kolme erilaista kytkentätapaa:

Fig 4.8 [KR12]

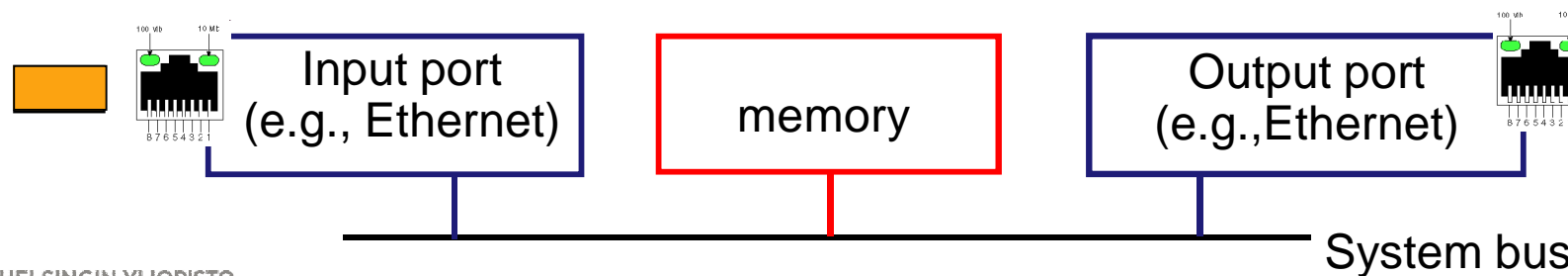




# Kytkeä muistin kautta

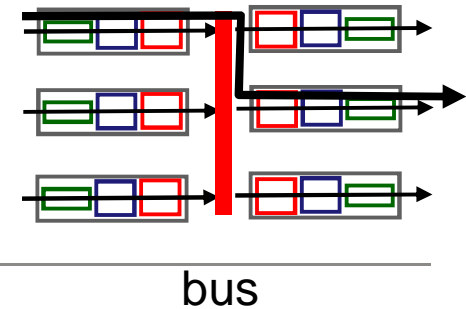


- “Tavallinen” tietokone reitittimenä
  - Sisääntulo (Input): keskeytys, CPU kopioi paketin muistiin, tutkii minne on menossa
  - Ulosmeno (Output) : CPU kopioi paketin muistista
  - **Väylä** (Bus) on pullonkaula: 2 kopiointia per paketti
- Linkkikerros ja fyysinen kerros laite-toimintoja
- Jonot keskusmuistissa





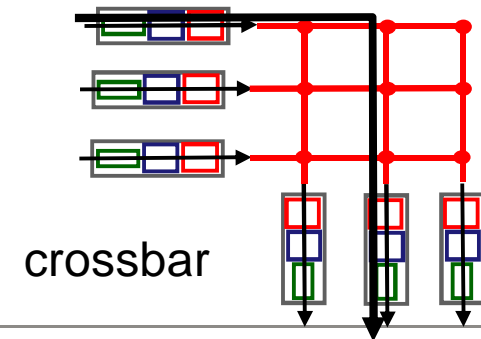
# KytKentä väylän kautta



- Sisääntulo siirtää paketin väylän kautta suoraan ulosmenoporttiin
- Vain yksi kytkentä aktiivinen kerrallaan
  - Väylä edelleen pullonkaula
- Väylänopeus rajoittaa kytkentänopeutta
  - Gbps nopeudet, riittävä LAN- ja yritysverkoilla



# KytKentä risti- kytkennän kautta



- Yhden väylän sijaan ristikytkentä (crossbar switch)
  - $2 \cdot N$  väylää yhdistää  $N$  sisääntuloa ja  $N$  ulosmenoa
  - Valitse vaaka- ja pystylinja
- Jos sama ulosmeno/sisääntulo, odotus sisääntuloportissa (HOL)
  - Sisääntulo voi pilkkoa paketin pienemmiksi soluiksi (cell) ja välittää yksi kerrallaan
  - Ulostulo kokoaa solut taas paketeiksi
- Suuri siirtonopeus
  - Esim. Cisco 12000: 64 Gbps



# Reititys ja reititysprosessori

---

- Prosessori suorittaa reititysprotokollaa
  - Reititysinformaation välitystä reitittimeltä toiselle
  - RIP, OSPF, BGP,...
  - Esim. 5 minuutin välein
- Sisääntulot toimittavat reititysprotokollien paketit prosessorille
- Ylläpitää porttien reititystauluja
  - Kun muuttuu, uusi kopio kullekin portille
- Hallinta- ja ylläpitotoimintoja
  - Reitittimelläkin voi olla suoritettavana sovelluksia



# Pakettien hylkäys

---

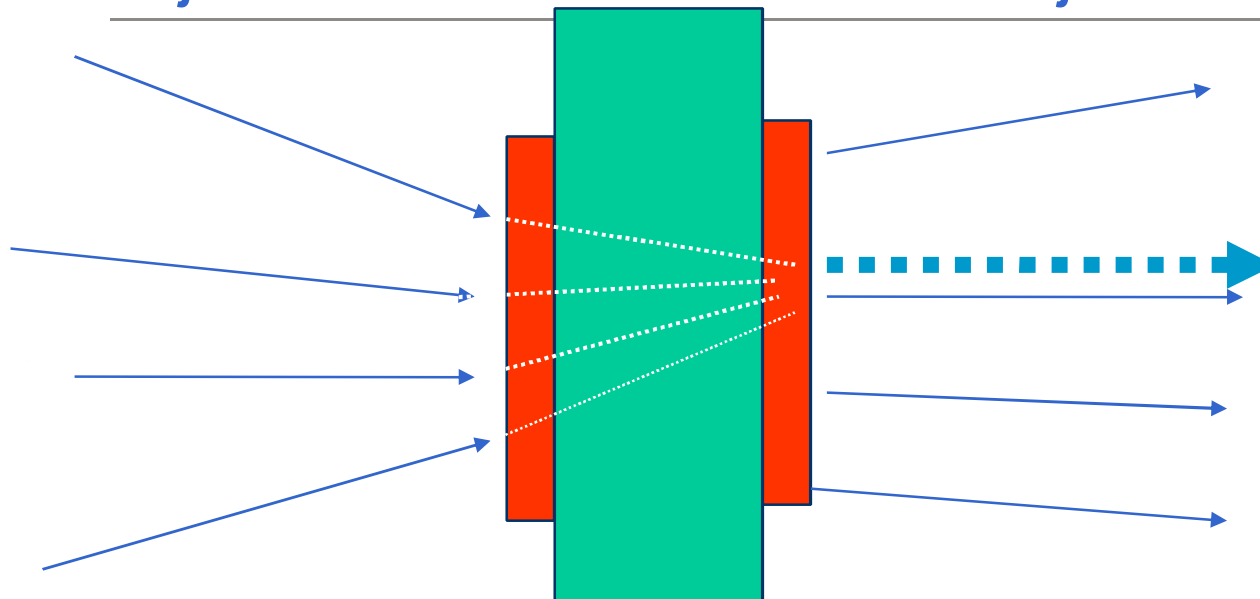
1. Kun puskuritila ei riitä
  - Hylkää saapuva paketti (drop-tail) tai joku muu ..
  - Se kummassa jonossa paketit hylätään, riippuu kytkennän ja linjan nopeuden suhteista
  - RED (Random Early Detection): hylkää jo ennenkuin puskuri täyttyy
2. Siirtovirhe
  - Linkkikerros saa hylätä virheellisen
  - Verkkokerros saa hylätä virheellisen (ICMP-protokolla )
3. Paketin elinaika (Time-to-live, TTL)



# Paketin hylkäys ulosmenossa

N linjaa sisään

N linjaa ulos

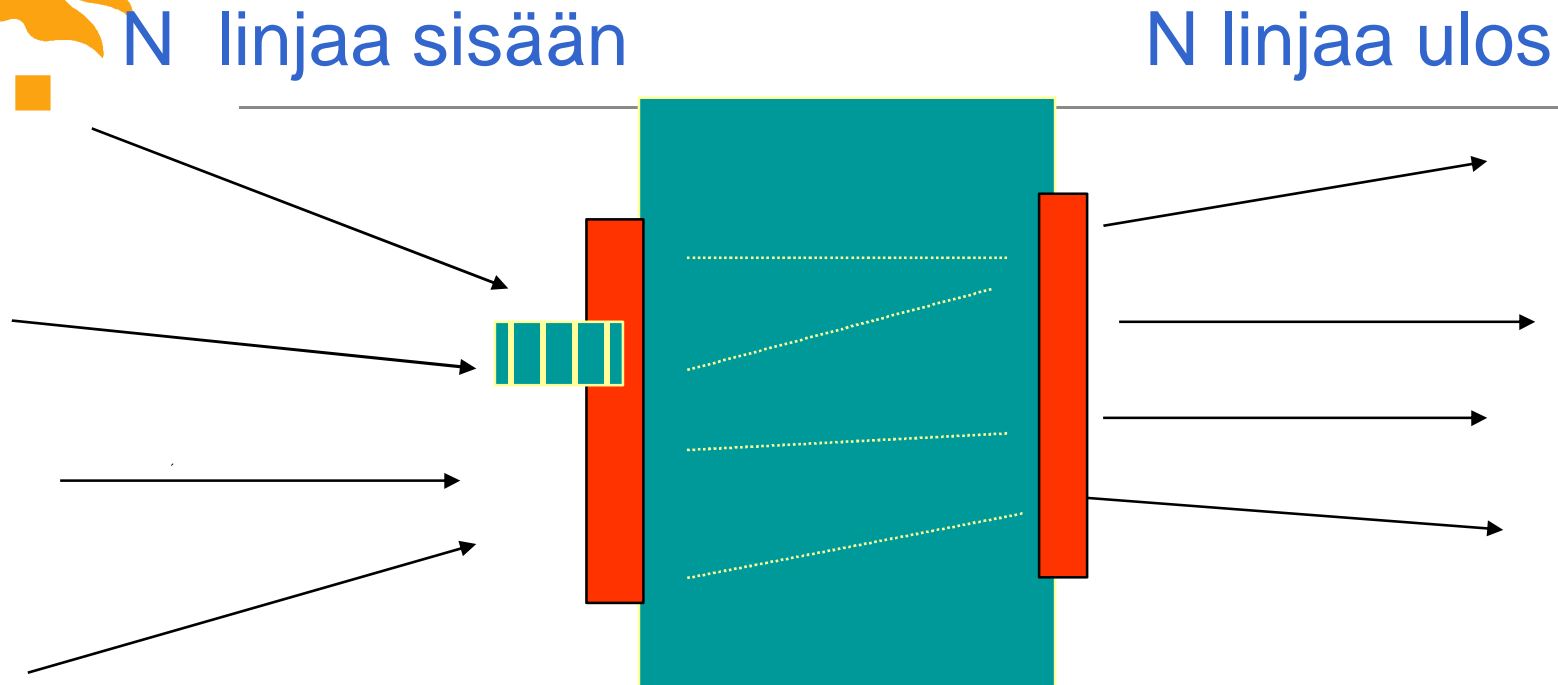


Kytкин toimii riittävällä nopeudella, joten sisääntulossa ei tarvitse jonottaa.

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



# Paketin hylkäys sisääntulossa



Jos kytkin ei toimi tarpeeksi nopeasti, sisääntuloportteihin syntyy jonoja.

Esim. Ristikytkenässä 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.

(HOL head-of-the-line-blocking)



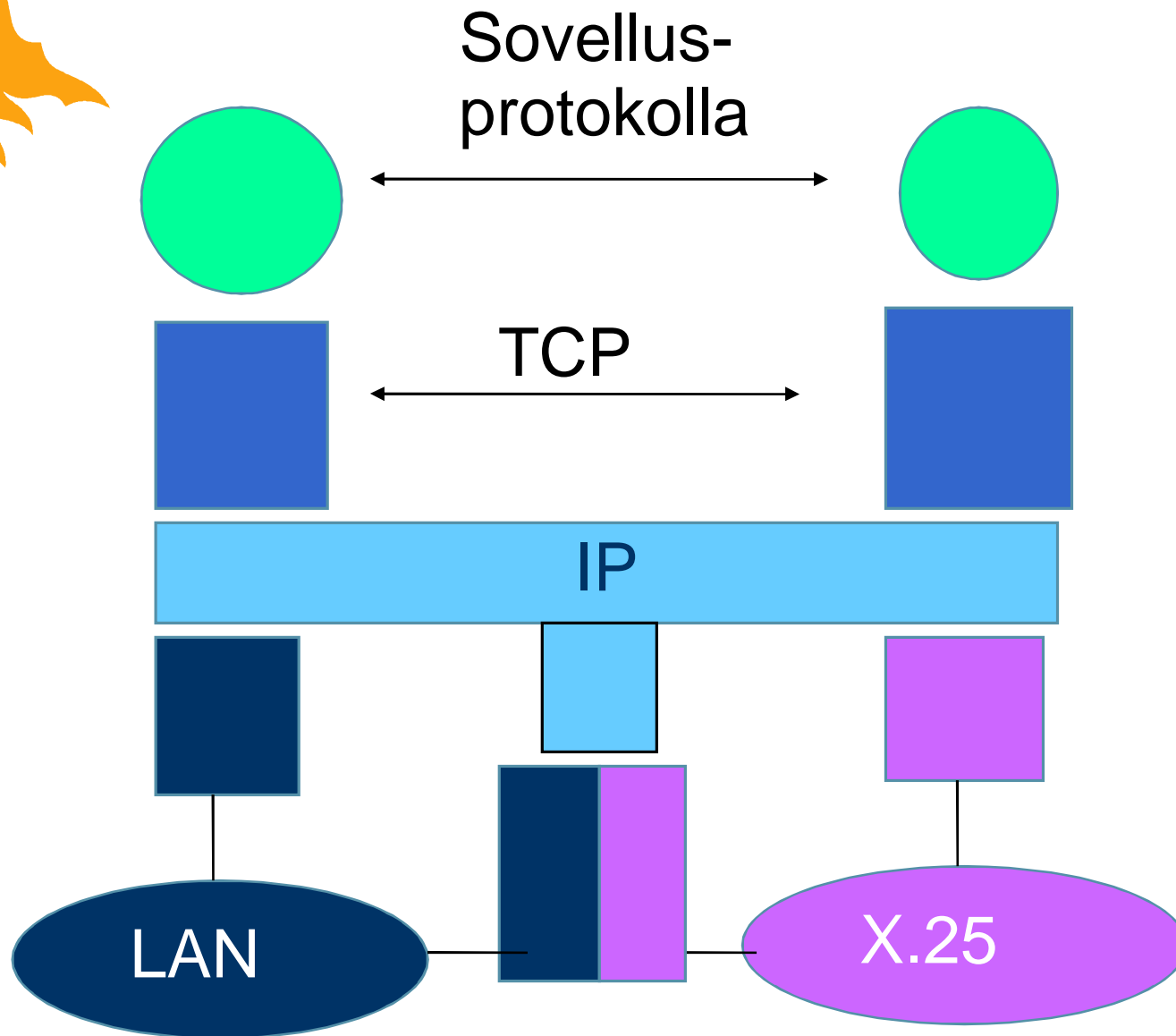


# Verkkokerros

## IP-protokolla

### Ch 4.4

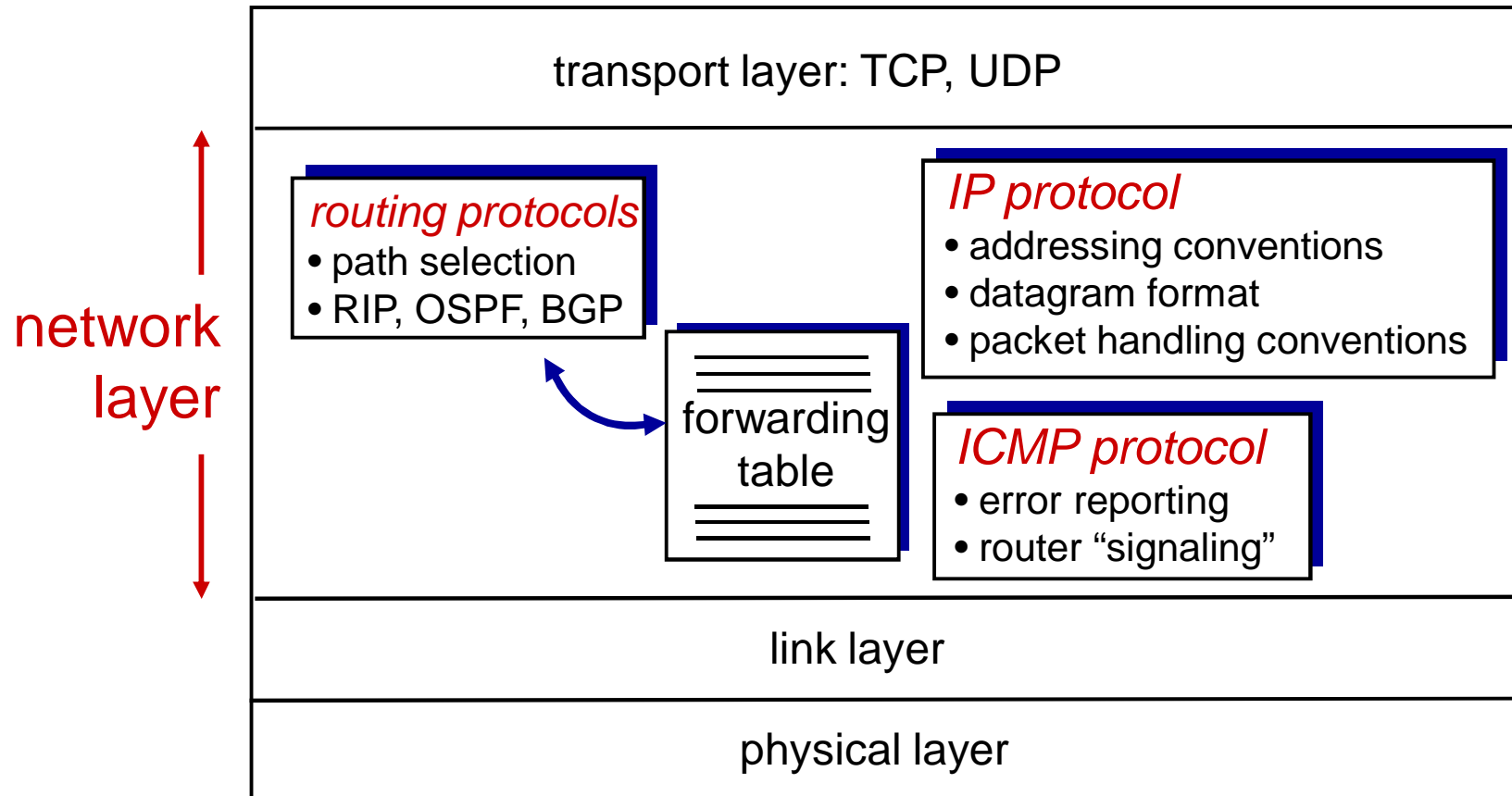
RFC 791





# Internetin verkkokerros

Fig 4.12 [KR12]





# Internetin verkkokerros

**Tällä kurssilla:** IPv4 ja reitityksen periaatteet  
Etäisyysvektoreireititys ja Linkkitilareititys (luento 8)

## **Internet-protokollat kurssilla:**

IPv6, ICMP (Internet Control Message Protocol)

Reititysprotokollat

Reititystaulujen (forwarding table) ylläpitämistä varten

Erillään tavallisten pakettien lähetyksestä

RIP (Routing Information Protocol): etäisyysvektorialgoritmi

OSPF (Open Shortest Path First): linkkitila-algoritmi

BGP (Border Gateway Protocol): hierarkkinen,  
autonomisten alueiden välinen algoritmi



# Internetin verkkokerros

---

## ICMP (Internet Control Message Protocol)

Protokolla, jolla isännät ja reitittimet vaihtavat verkkokerroksen kuulumisia

Tavallaan verkkokerroksen päällä: IP-paketissa kuljetuskerroksen tietojen sijasta ICMP-dataa

Virheraportointi: unreachable host/network/port/protocol

Reititin ei tiedä, minne toimittaisi ...

Kaiutus: echo request / reply

tätä ping ja traceroute käyttävät RTT:n mittaamisessa

## IPv6

Uudistettu versio IP-protokollasta, 128 bitin IP-osoite  
mm. kiinteänkokoinen otsake, ei tarkistussummaa,  
pakettien paloittelu jo lähettäjän koneessa



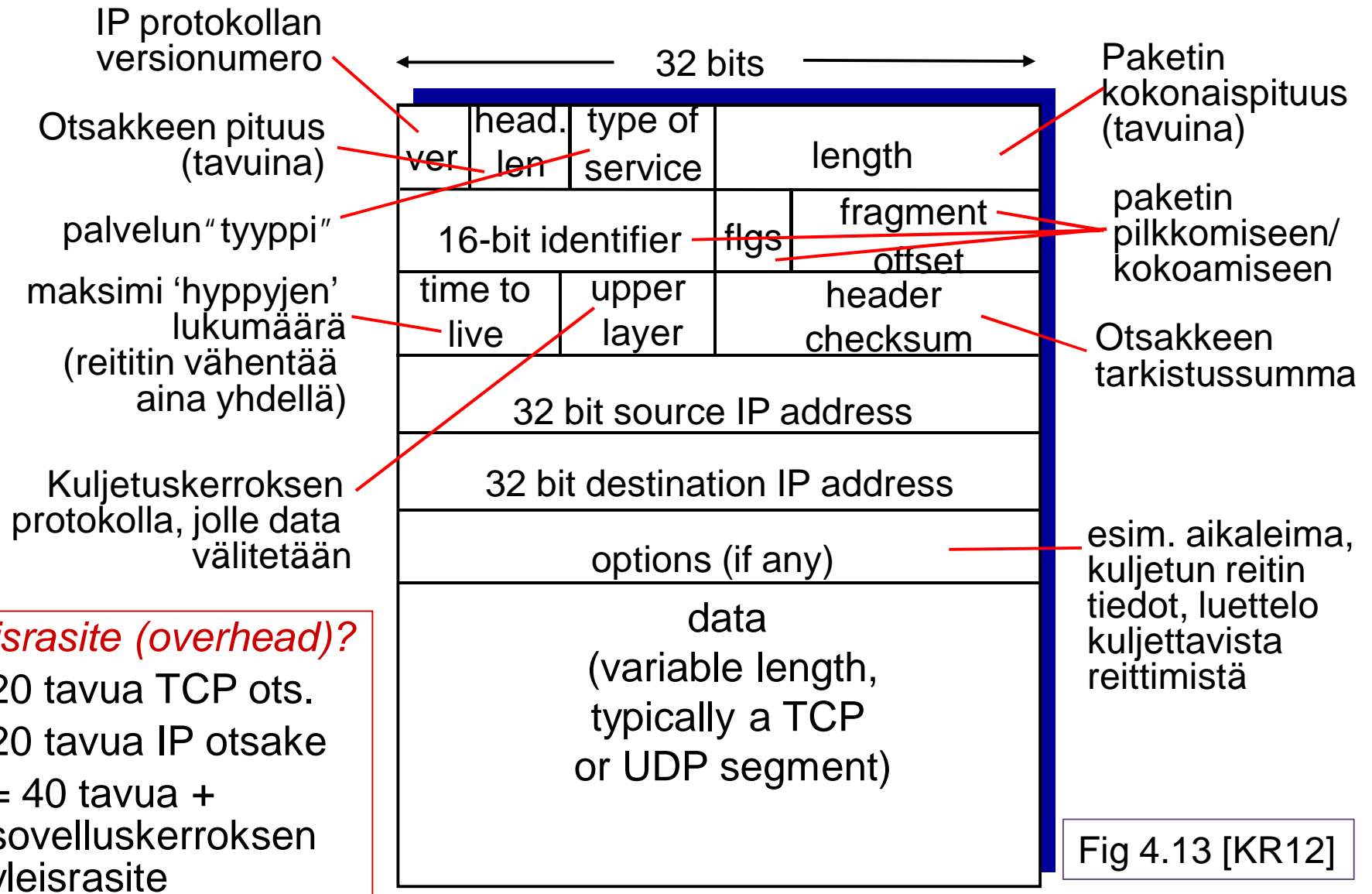
# Verkkokerroksen IP-protokolla siirtää kuljetuskerroksen segmentit lähdekoneelta kohdekoneelle

---

- Tarvitaan
  - Osoitteet (lähettäjä, vastaanottaja)
  - Tieto ylemmän kerroksen protokollasta (UDP, TCP tai joku muu), jotta osaa antaa oikealle rutiinille
  - Liian ison IP-paketin paloittelu tarvittaessa pienemmiksi IP-paketeiksi
  - 'Harhautuneiden' pakettien hävittäminen (time-to-live)
  - Tarkistukset (checksum)
- Hyviä ominaisuuksia (?)
  - Siirtopalvelun eriyttäminen erityyppisille sovelluksille
  - Lähdereititys (source routing): lähettäjä määrää reitin, paketissa tieto siirtopolusta

# IPv4 paketin rakenne

otsake 20+ tavua



## Yleisrasite (overhead)?

- ❖ 20 tavua TCP ots.
- ❖ 20 tavua IP otsake
- ❖ = 40 tavua + sovelluskerroksen yleisrasite

Fig 4.13 [KR12]



# IP-otsake

---

- Versionumero
  - IPv4 vai IPv6, kummallakin erilainen otsake
- Otsakkeen pituus (header length)
  - Vaihtelevan pituinen optiokenttä, minimi on 20 B
- TOS-kenttä (Type of Service)
  - Varattu halutun palvelun kertomiseen:
    - Nopeus, luotettavuus, kapasiteetti; ääni vs. tiedosto
  - Yleensä ei ole käytössä (osa käytössä uusissa reitittimissä)
- IP-paketin pituus (Datagram length)
  - Koko IP-paketin pituus tavuina, maksimi 65535 B
  - Tavallisimmin 576-1500 B
- Paketin tunniste (16-bit identifier), lippuja (flags), palan paikka (fragmentation offset)
  - Paketin pilkkomiseen pienemmiksi ja kokoamiseen takaisin isoksi





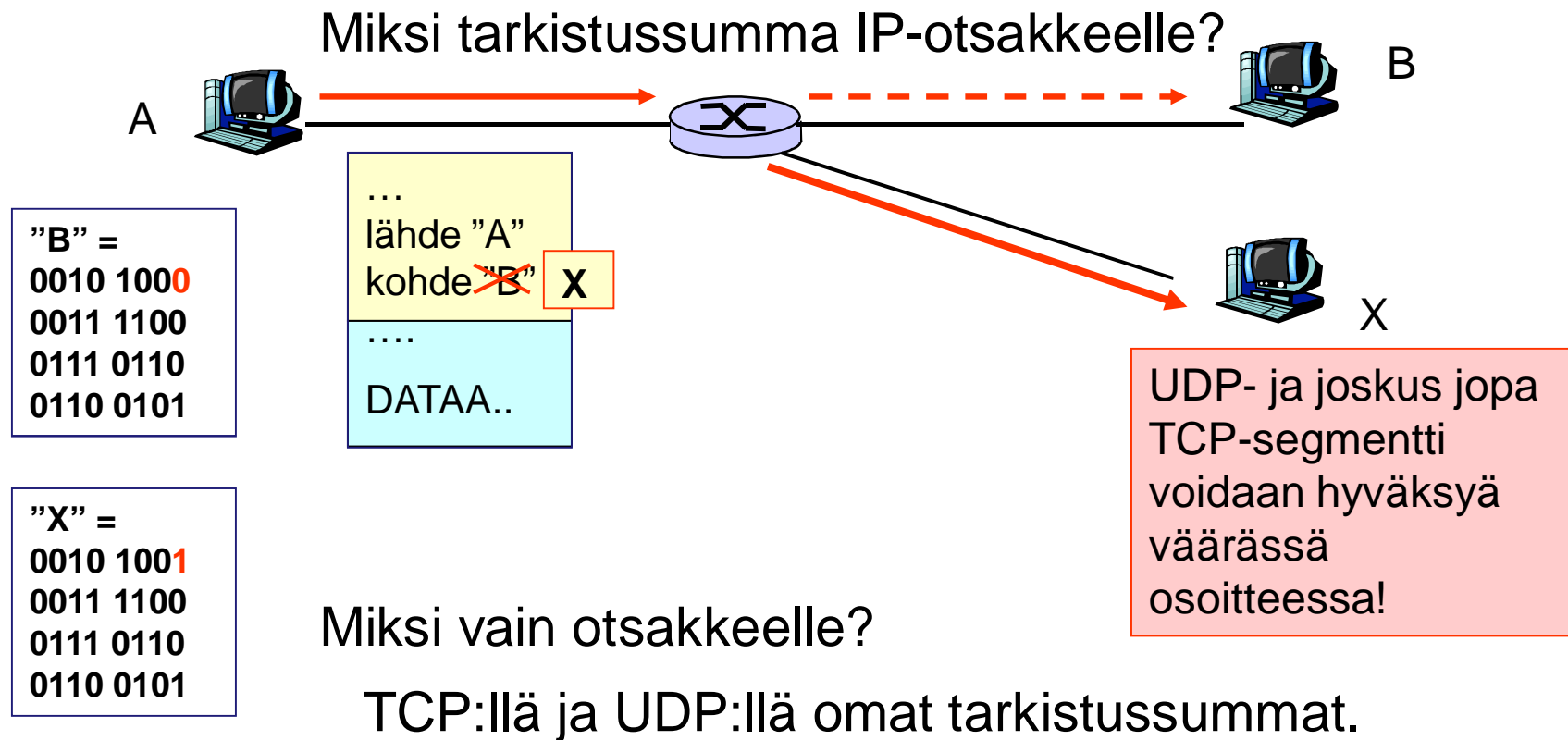
# IP-otsake (jatkuu)

---

- Elinaika (time-to-live, TTL)
  - Rajoittaa paketin elinaikaa, maksimi 255
  - Vähenee joka hypyllä reitittimestä toiseen, kun TTL=0, hylätään
- Kuljetusprotokolla (Upper-layer protocol)
  - Kumpi kuljetuskerroksen protokolla (TCP=6, UDP=17) vai kenties verkkokerroksen sisäistä dataa (ICMP, reititysprotokolla)
- Otsakkeen tarkistussumma (Header checksum)
  - Vain otsakkeelle (Internet checksum)
  - *Tarkista ja laske uusi joka reitittimessä (TTL, Options)*
  - Hylkää virheellinen paketti
- Osoitteet (Source IP Address, Destination IP Address)
  - Lähteen ja kohteen IP-osoitteet
- Optiot (Options)
  - Laajennuksia: mm. lähdereititys, harvoin käytetty



# IP-otsakkeen tarkistussumma

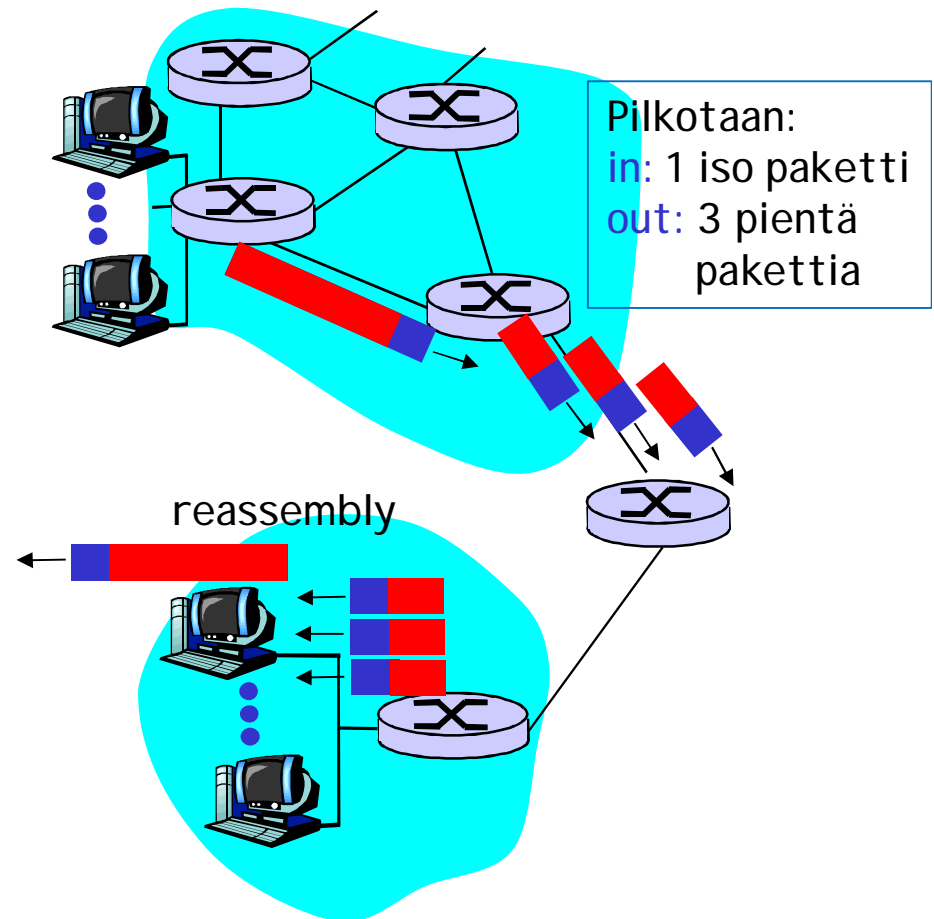




# IP-pakettien paloittelu (fragmentointi)

Fig 4.14 [KR12]

- Maximum transfer Unit (MTU)
  - suurin mahdollinen IP-paketti eri linkeillä eri kokoinen, esim. Ethernet 1500 B
- *Reititin pilkkoo tarvittaessa* liian ison paketin pienemmiksi paketeiksi (fragmenteiksi)
  - paketit voivat kukin kulkea eri reittiä
- *Kohdekone kokoaa yhdeksi*
  - IP-otsakkeessa kentät yhteenkuuluvien fragmenttien tunnistamiseksi
- IPv6 ei käytä fragmentointia





# Esimerkki: IP-paketin paloittelu (ei enää IPv6:ssa)

length =4000	ID =x	fragflag =0	offset =0
-----------------	----------	----------------	--------------

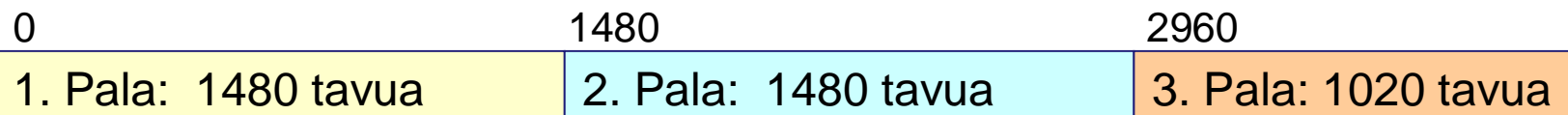
4000 tavun IP-paketti:  
dataa 3980 B  
MTU 1500 B

Yhdestä IP-paketista tulee  
3 pienempää IP-pakettia

1480 B dataa  
20 B IP-otsaketta

offset = 1480/8  
(8 tavun monikertoja)

length =1500	ID =x	fragflag =1	offset =0
length =1500	ID =x	fragflag =1	offset =185
length =1040	ID =x	fragflag =0	offset =370

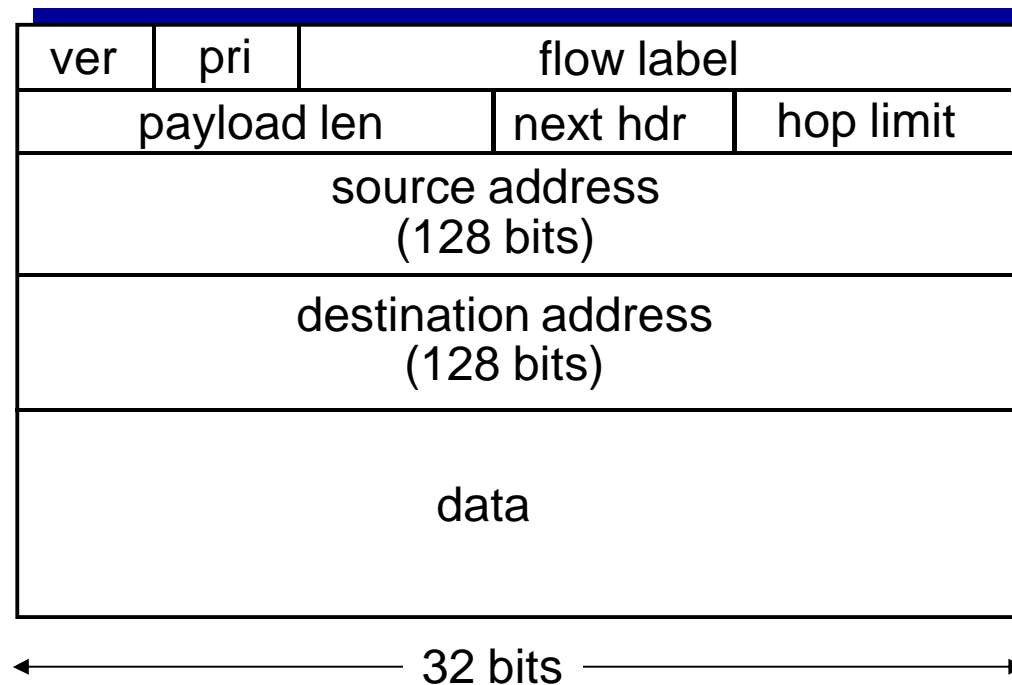




# IPv6 paketti (engl. Datagram)

Otsake: 40 tavua

- 128-bittiset osoitteet
- yksinkertainen otsake -> nopea käsittely reitittimissä
- Prioriteetti - luokittelu
- Vuo (flow)
- Ei paloittelua matkalla
  - 'Packet too big' ICMP virheviesti
- Ei tarkistussummaa

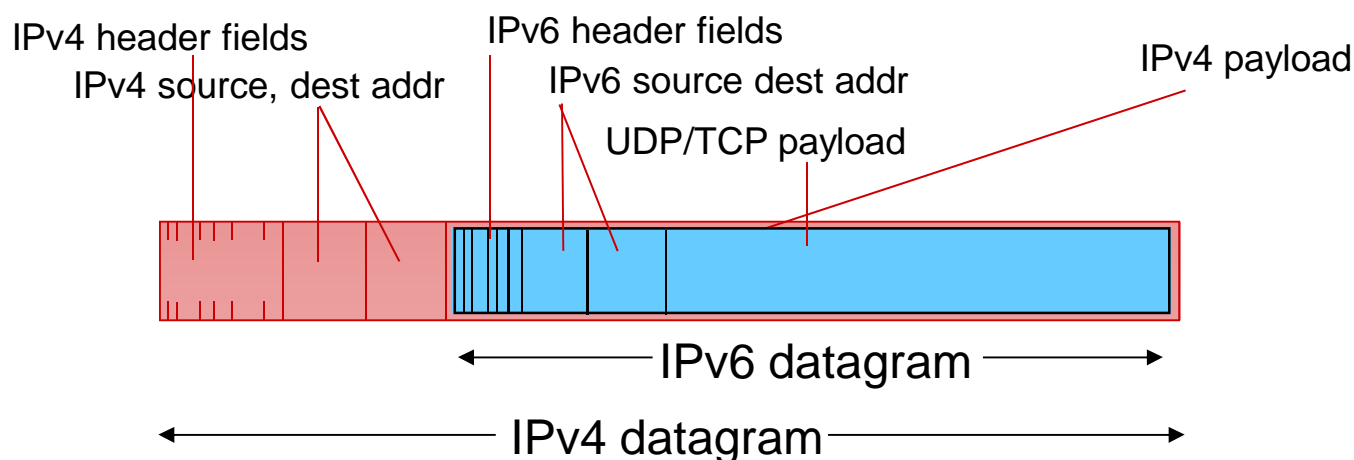




# Siirtymä IPv4:stä IPv6:een

- Kaikkea ei voi päivittää kerralla
  - Käytetään sekaisin uusia ja vanhoja reitittimiä

**tunnelointi:** IPv6-paketti IPv4-paketin datana, kun siirretään IPv4-reitittimien välillä (tai päinvastoin)





# Tunnelointi (tunneling)

logical view:



physical view:

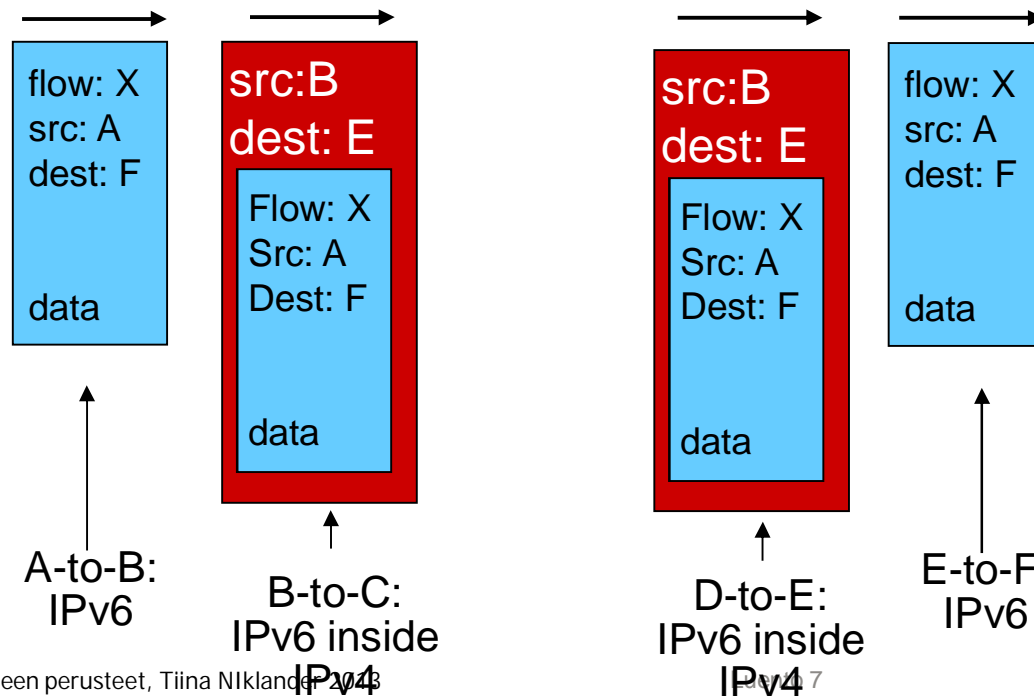
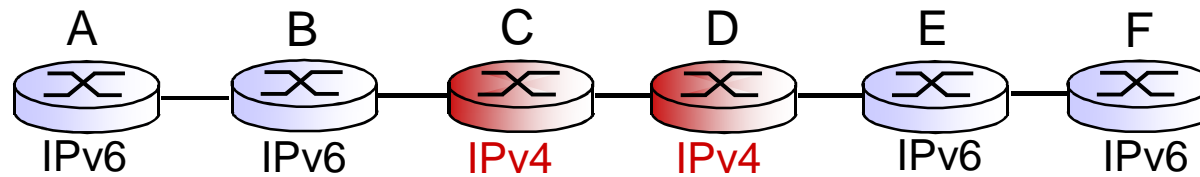


Fig 4.26 [KR12]



# Kerrokset ja otsakkeet tähän mennessä





# Paketin sisältö

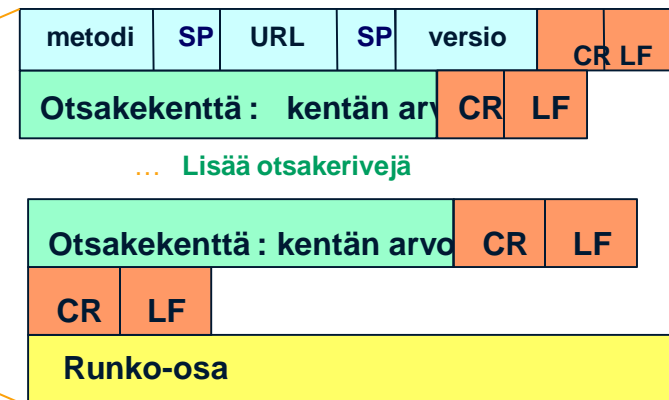
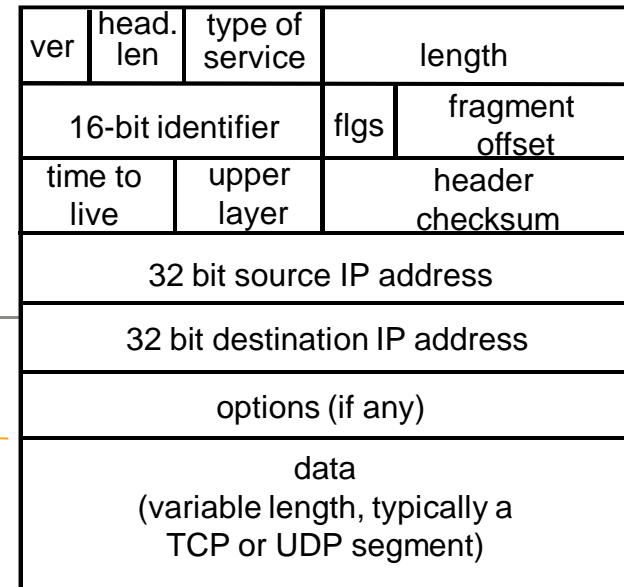
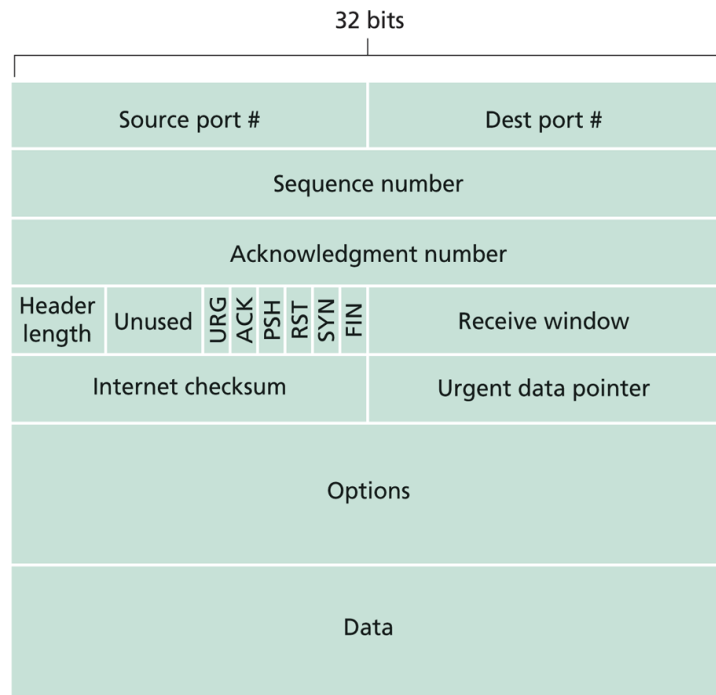


Figure 3.29 ♦ TCP segment structure