

# Palvelun laatu (QoS) Internetissä

(Kurose-Ross, Computer Networking, ss. 536-556, Tanenbaum, ss. 393-395)

---

- ◆ Sovellus ei saa mitään takuita palvelun laadusta (Best effort)
  - joskus kaikki toimii hyvin, joskus ei
  - sovellus ei voi paljoa siihen vaikuttaa
- ◆ Tällainen palvelu ei sovi kaikille sovelluksille
  - ◆ audio/video
  - ◆ multimedia
  - ◆ IP-puhelu
- ◆ QoS-ajattelu myös Internetiin?
  - viive, viipeen vaihtelu
  - virheettömyys

# Paremmat takeet palvelun laadulle

---

## ◆ **Integrated Services** (IntServ)

- sovelluksilla erilaisia datavoita, joilla erilaiset tarpeet
- varataan etukäteen resurssit, jotta eri datavoiden vaatimukset voidaan täyttää

## ◆ **Differentiated Services** (DiffServ)

- erilaisia paketteja, joilla erilaiset tarpeet
- reititin kohtelee näitä paketteja eri tavoin
  - ◆ esim. omat ulosmenojonot tärkeille paketeille

# IntServ eli Integroidut palvelut (Intergrated Services)

---

- ◆ Integrointi eli palvelun laatuajattelun liittäminen Internetiin
- ◆ Käyttäjä voi valita erilaisia palveluluokkia ja yhteyden laatutasoja
  - » kullekin sovellukselle palveluja sen tarpeiden mukaan
  - » laadusta joutuu yleensä maksamaan
- ◆ yhteysajattelu (liikennevuo)
  - sovitaan ensin yhteydellä käytettävän palvelun laadusta
  - verkko (= reitittimet) huolehtivat siitä, että sovellus saa tarvitsemansa palvelunlaadun

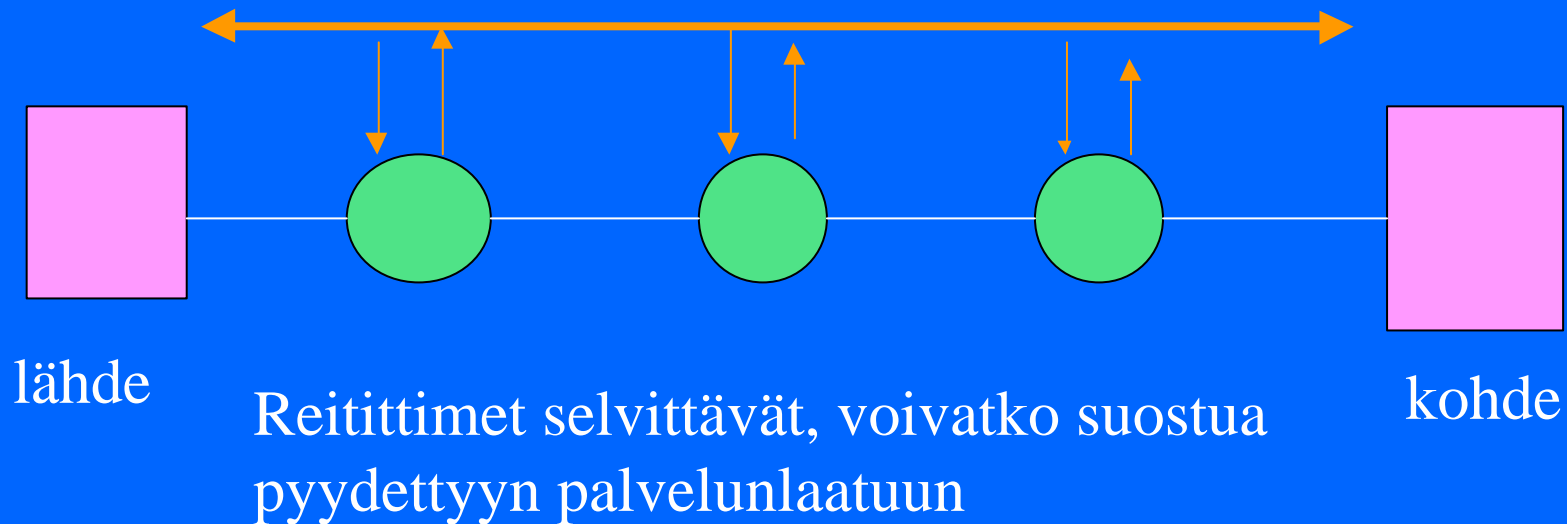
# Takuu perustuu resurssien varaamiseen

---

- ◆ Jokainen reititin yhteyden reitillä päättää, pystyykö se antamaan yhteydelle sen haluaman palvelun
- ◆ yhteyden muodostusvaiheessa (Call Setup) kunkin reitittimen on
  - tunnettava yhteyttä haluavan sovelluksen liikennevuo
  - tiedettävä millaista palvelua sovellus tälle liikennvuolle haluaa
  - tiedettävä oma tilansa eli pystyykö täyttämään vaatimukset
    - ◆ miten paljon resursseja on vielä jäljellä
    - ◆ miten paljon resursseja on jo varattu

# Yhteyden muodostusvaihe

‘Signallointi’ yhteyden muodostamiseksi



# Yhteyden muodostuksessa tarvitaan

---

- ◆ Liikennekuvaus (traffic characterization)
  - Tspec (RFC 2210)
- ◆ Halutun palvelunlaadun määrittely (specification of the desired QoS)
  - Rspec (RFC 2215)
- ◆ Yhteydenmuodostuksessa käytetty protokolla (signaalointiprotokolla)
  - kuljettaa liikennekuvauksen ja palvelumäärittelyn reitin reittimeltä toiselle
  - valittu protokolla **RSVP** (Resource reSerVation Protocol) (RFC 2205)

# Tspec: Token\_Bucket\_Tspec

---



31

0

# Liikenteen tasoitus (traffic shaping)

---

- ◆ liikenne tyypillisesti purskeista
  - » aiheuttaa ruuhkaisuutta
- ◆ tasoitetaan liikennevirtaa puskurilla
  - » puskuri toimii jonona
    - vuotava ämpäri
    - vuoromerkkiämpäri
- ◆ liikennevirran määrittely
  - määrittelee asiakkaan oikeudet ja velvollisuudet



# Vuotava ämpäri (leaky bucket)

---

- ◆ purskeisuutta tasoittaa iso puskuri, josta liikenne valuu tasaisesti
  - » ‘vuotava ämpäri’
  - » yksi tavu / yksi paketti lähtee jossain aikayksikössä, **jos on lähetettävää**
- ◆ jos datapurske mahtuu puskuriin, se aikanaan pääsee matkaan
  - » äärellinen jono
  - » yläraja saapumistiheydelle
  - » jos saapumistiheys liian suuri, niin dataa katoaa

# Esimerkki

---

- ◆ Tietokone tuottaa dataa purskeissa hyvin suurella nopeudella. Oletetaan 50 ms purske, jonka aikana dataa lähetetään 2 MB. Mikä on keskimääräinen datanopeus purskeen aikana? Kuinka suuri puskuri vuotavaan ämpäriin tarvitaan? Kuinka kauan kestää purskeen käsittely, jos ämpärin valuma on tasainen 4 MB sekunnissa?
- ◆ Datanopeus purskeessa on  $2\text{MB}/50 \cdot 10^{-3} = 40\text{MB/s}$
- ◆ Purskeen pitää mahtua puskuriin eli tarvitaan lähes 2MB puskuri.
- ◆ Koska vuotavan ämpärin ulostulo on 4 MB/s eli 2 MB:n purskeen käsittely kestää 500 ms.

# Vuoromerkkiämpäri (Token bucket)

---

- ◆ lähettäminen vaatii vuoromerkin
- ◆ vuoromerkkejä generoituu tasaisella nopeudella
- ◆ jos ei lähetettävää, merkkejä jää säästöön
  - » korkeintaan niin paljon kuin ämpäriin mahtuu
  - » => sallii rajoitetut 'minipurskeet', maksimissaan ämpäriin kokoiset
- ◆ joustavampi kuin vuotava ämpäri
  - » purskeet voivat aiheuttaa ruuhkaa => vuotava ämpäri vuoromerkkiämpäriin perään

# Kahdenlaista palvelua

---

- ◆ **Taattu palvelu** (guaranteed service) (RFC 2212)
    - takaa rajat jonotusviiveille reitittimen jonoissa
      - » kokonaissiirtoviive riippuu käytetystä reitistä ja linkkien nopeuksista
  - ◆ **Valvotun kuorman palvelu** (controlled-load service) (RFC 2211)
    - “vastaava palvelunlaatu, jonka sama vuo suunnilleen saisi kuormittamattomalta reitittimeltä”
- 30.11.2001    » hyvä laatu, ilman takuita

# Taatun palvelun perusidea:

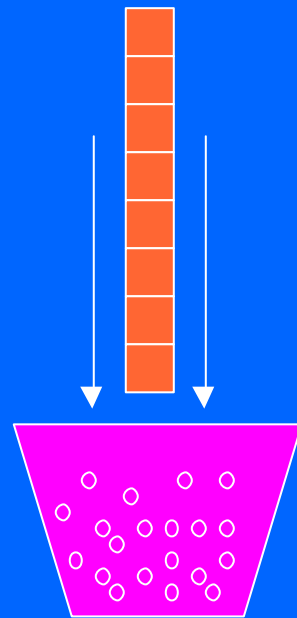
---

- ◆ Liikenne kuvataan vuoromerkkiämpäriin avulla
  - lähetyslupien määrä sekunnissa =  $r$
  - vuoromerkkiämpäriin koko =  $b$  lähetyslupaa
- ◆ palvelu haluttuna siirtonopeutena  $R$  bps
- ◆  $\Rightarrow$  maksimiviive reitittimessä on rajoitettu
  - Jotta puskuri ei vuotaisi yli lähettäjä saa lähettää  $t$ :n mittaisena aikana  $t$  korkeintaan  $r \cdot t + b$  bittiä
  - Jos siirtonopeus jonosta on vähintään  $R$  ( $> r$ ), niin maksimiviive on korkeintaan  $b/R$

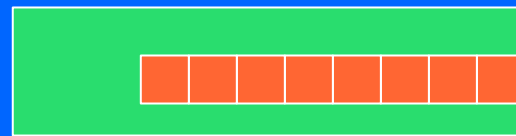
Oletetaan  
alkutilanteeksi  
ämpäri täynnä  
lupia.

$$R' > r$$

Maksimissaan voi  
tulla  $b:n$  paketin  
kokoinen ryöppy



Ryöpyn kaikki  $b$   
pakettia siirtyvät  
reitittimen  
ulosmenojonoon



Esim.  $b = 20$  lupaa  
 $r = 2$  lupaa/s

Yhdellä luvalla saa  
lähettää yhden paketin

Jos paketeille on luvattu  
siirtonopeus  $R'$  pakettia /s,  
niin jonon purkaminen  
kestää  $b/R'$  sekuntia eli  
viimeisenkin paketin viive  
on rajoitettu.

Todellisuus on hieman  
monimutkaisempi!

# Valvotun kuorman palvelu

---

- ◆ Hyvä ‘best-effort’-palvelu:
  - lähes kaikki paketit ehjinä perille
  - jonotusviive reitittimissä on lähes olematon
- ◆ sovellus ilmoittaa Tspec:insä ja kukin reititin varmistuu siitä, että sillä on tarpeeksi resursseja
  - ◆ kaistanleveyttä , puskuritilaa ja käsittelykapasiteettia
  - jos resurssit eivät riitä, niin ei hyväksytä
- ◆ sovellus ei voi esittää mitään erityisiä vaatimuksia virheettömyydelle tai viipeelle

- 
- ◆ Yksinkertainen tapa toteuttaa monien nykyisten sovellusten tarpeet
    - sovellukset toimivat periaatteessa hyvin nykyisessä Internetissä, mutta eivät kestä verkon ruuhkautumista
    - esim. monet tosiaikaiset multimediasovellukset
      - » ‘joustavat’ sovellukset



# RSVP (Resource reSerVation Protocol)

---

- ◆ Sovellukset voivat varata itselleen resursseja Internetistä
  - tietovuot, monilähetykset, multimediasovellukset
    - ◆ esim. videolähetyksellä vastaanottajalle
  - resurssi ~ kaistanleveys, (puskuritila)
- ◆ vastaanottaja huolehtii varauksista
- ◆ resurssit varataan monilähetyksissä

- 
- ◆ Protokolla kaistanleveyden varaamiseen
    - ei varausten toteuttamiseen verkossa
      - » on reitittimien asia huolehtia siitä, että tietovuot todella saavat niille varatun kaistanleveyden
        - ◆ skedulointi
    - ei myöskään määrää, mille linkeille varaukset tehdään
      - » reititysprotokollat huolehtivat reittien valitsemisesta
    - ‘signaalointiprotokolla’
      - ◆ isäntäkoneet voivat varata siirtokapasiteettia tietovuolle

# Heterogeenisyys

---

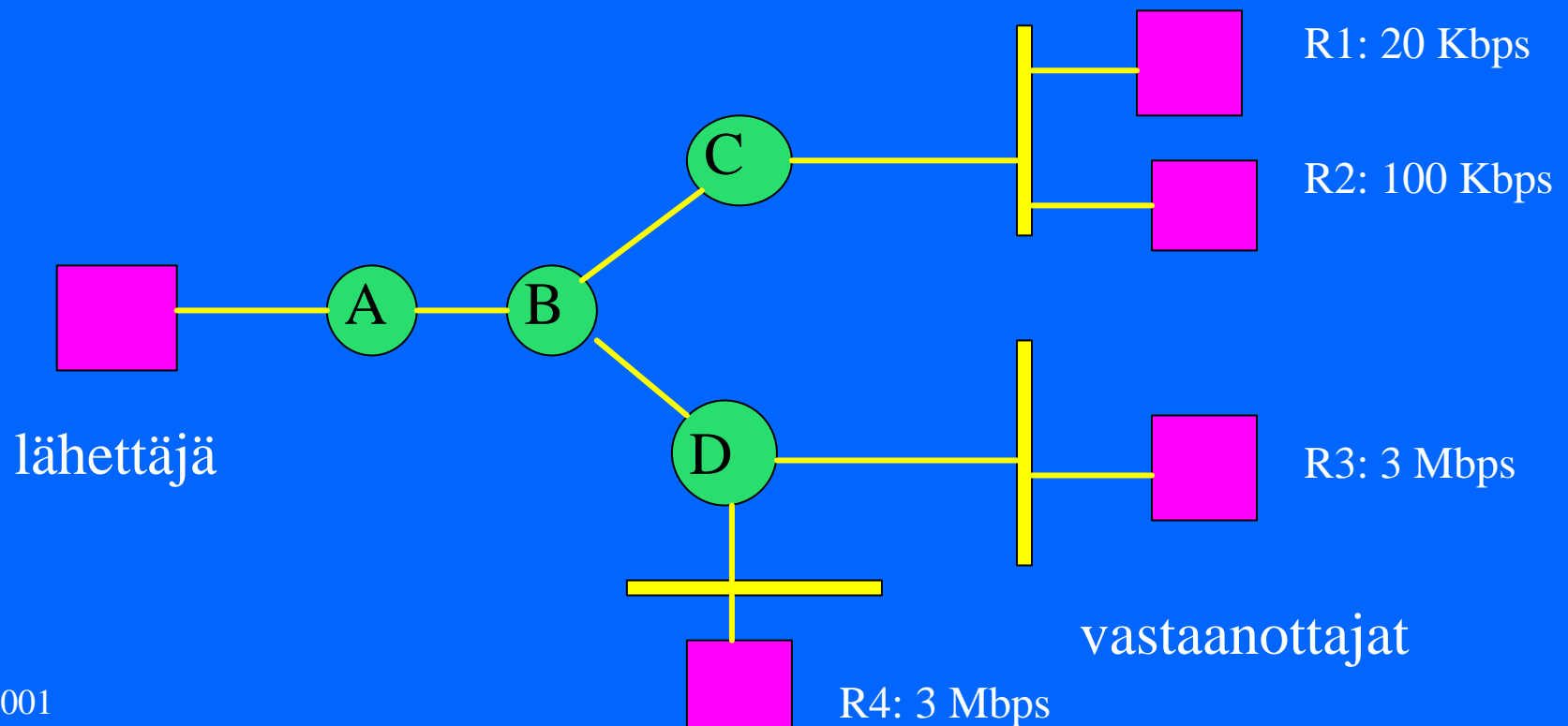
- ◆ Tietovuon vastaanottajat voivat olla hyvin heterogeenisia
  - pystyvät vastaanottamaan eri nopeudella
    - ◆ Videota voidaan vastaanottaa nopeudella 28.8 Kbps, 128 Kbps tai 10 Mbps
    - ◆ koodataan video useana eri kerroksena
  - lähettäjän tarvitsee tietää vain vastaanottajajoukon korkein siirtonopeus

# Esimerkki: videolähetys urheilukilpailusta

---

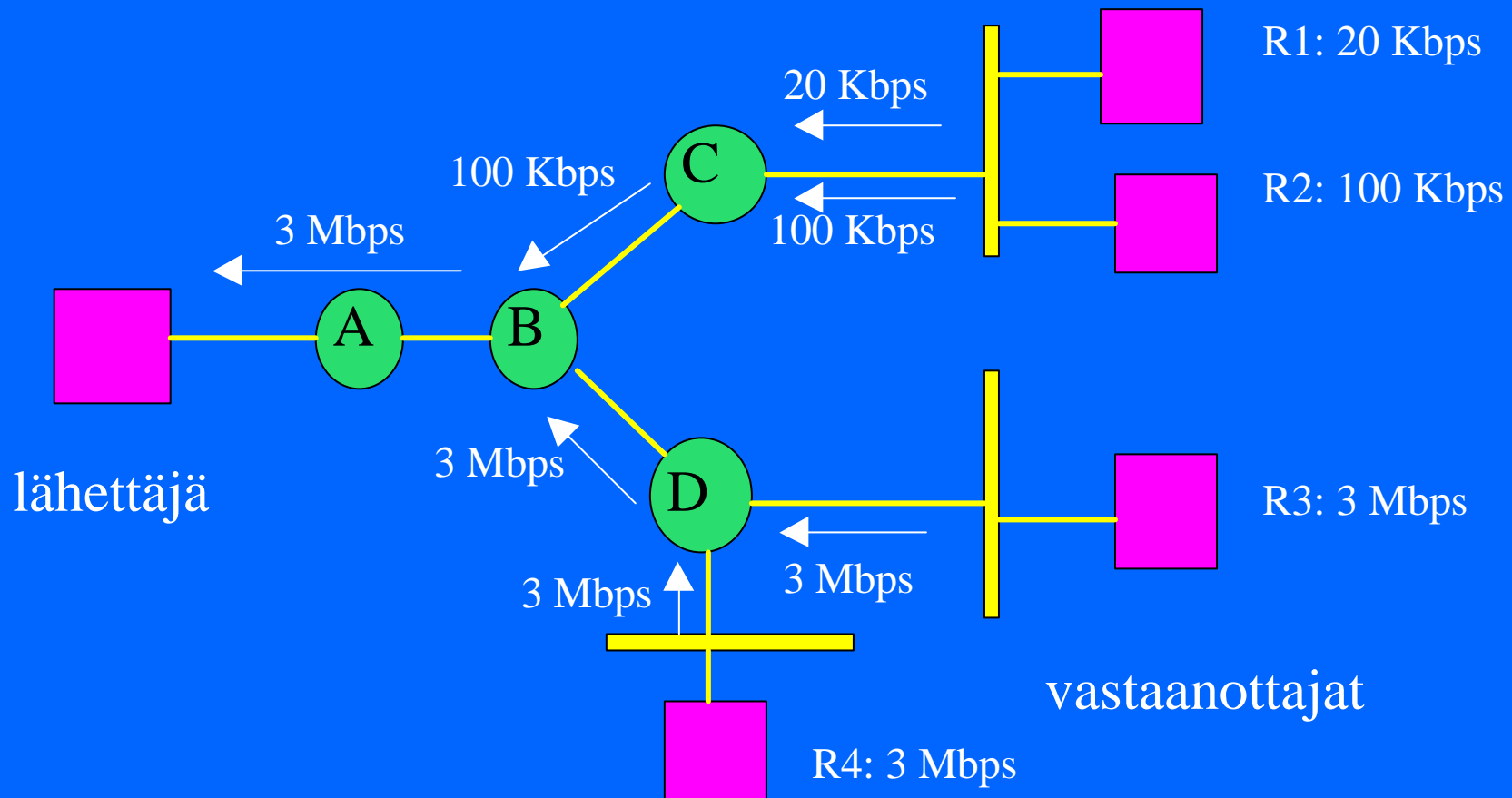
- ◆ ‘sessio’ (istunto, Session)
  - useita monilähetysdatavoita
  - useita lähettäjiä
  - joka vuolla sama monilähetysosoite
  - reitittimet tunnistavat paketeista, mihin sessioon ja mihin vuohon ne kuuluvat
    - ◆ esim. Monilähetysosoite => sessio
    - ◆ IPv6:n suonimiö => vuo
  - lähettäjä lähettää usealle vastaanottajalle videokuva kilpailusta
    - ◆ joka paketissa monilähetysosoite => vastaanottajat

- ◆ Monilähetyksprotokolla on muodostanut monilähetykspuun lähettäjältä vastaanottajille

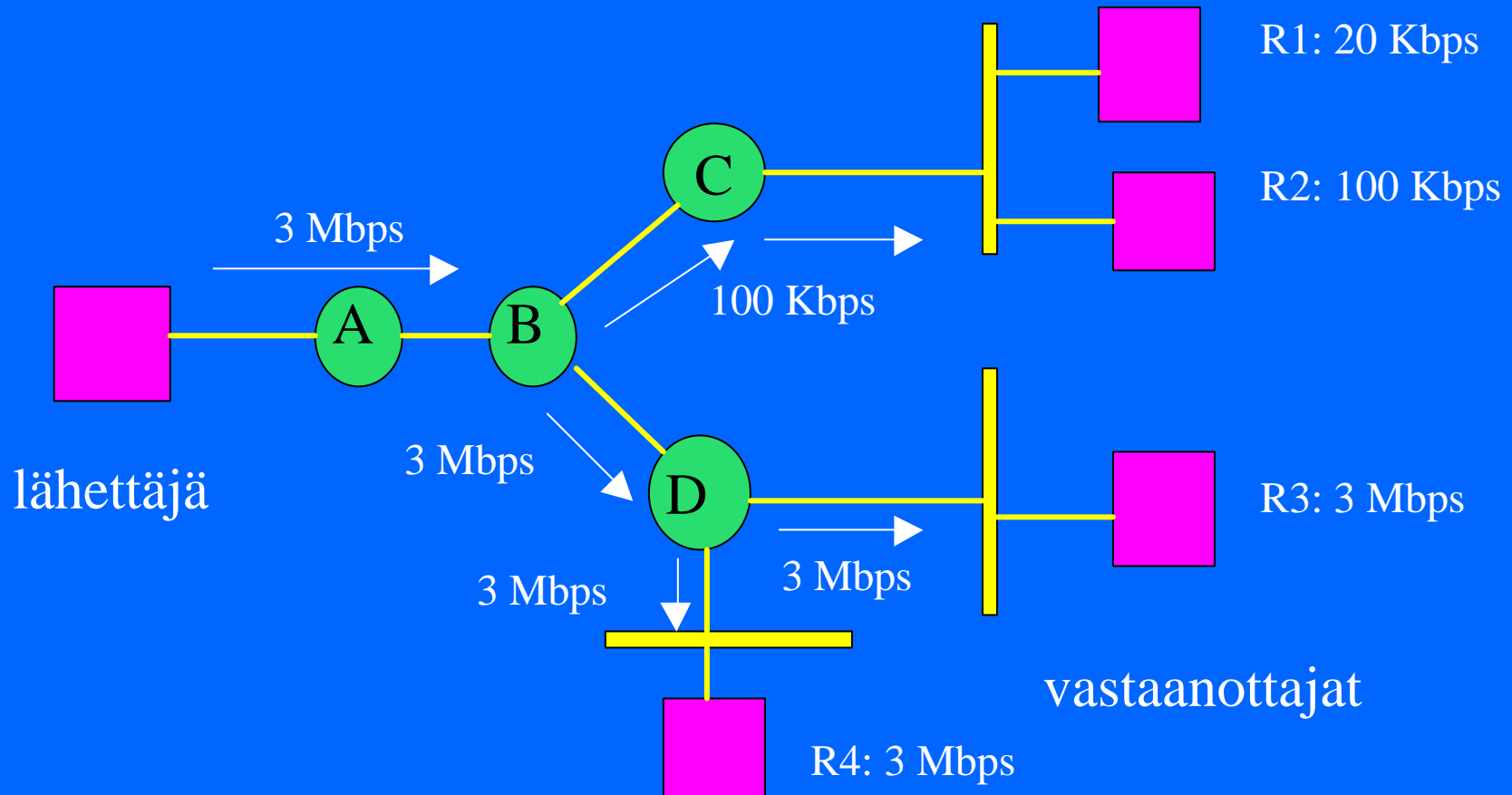


- 
- ◆ Jokainen vastaanottaja lähettää varaussanomman
    - ◆ käyttäen esim. reverse path forwarding algoritmia
    - ◆ kertoo millä nopeudella haluaa vastaanottaa lähettäjältä
  - ◆ sanoman saanut reititin varautuu antamaan pyydetyn kapasiteetin
    - ◆ pakettien skeduloija
  - ◆ reititin lähettää eteenpäin vain suurimman saamistaan varauksista

# Varaussanommat



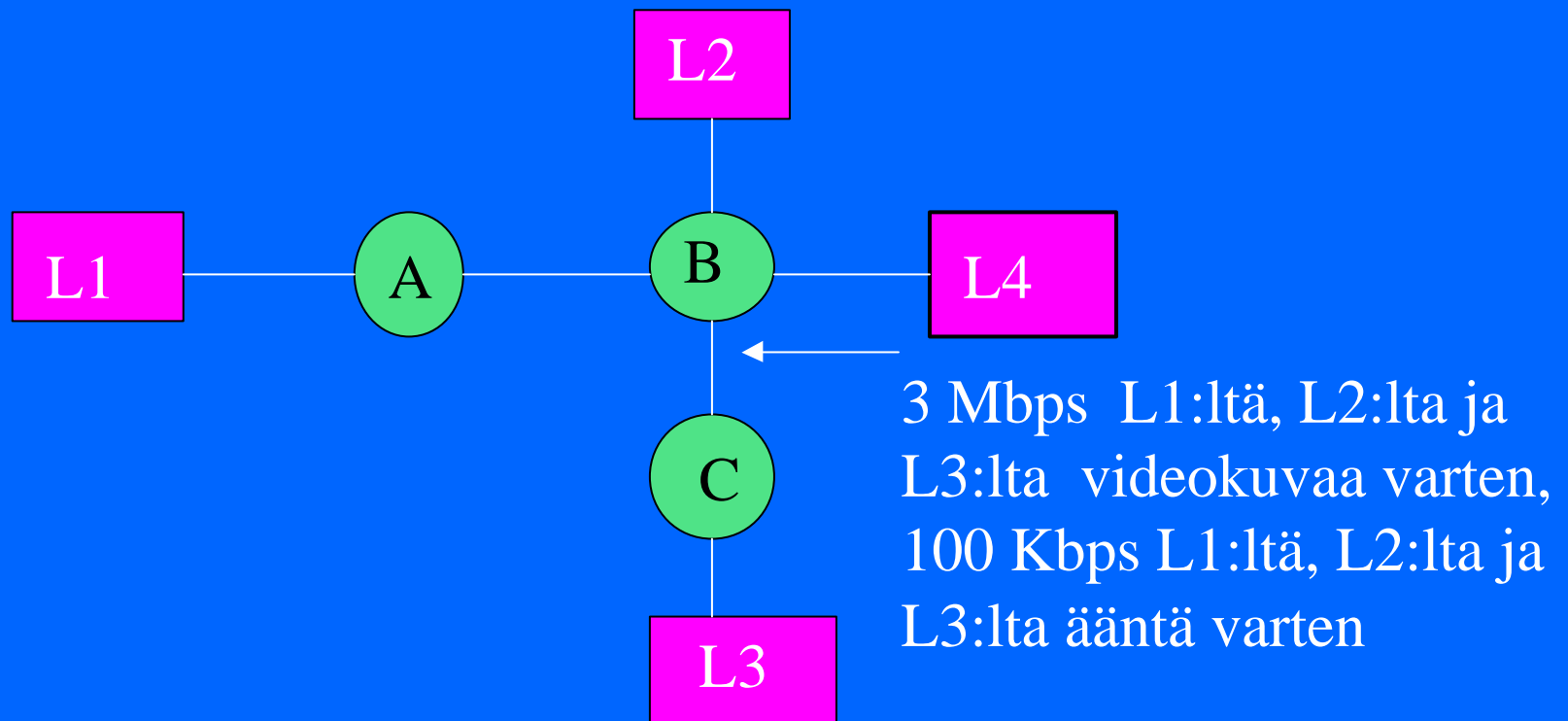
# Tehdyt varaukset





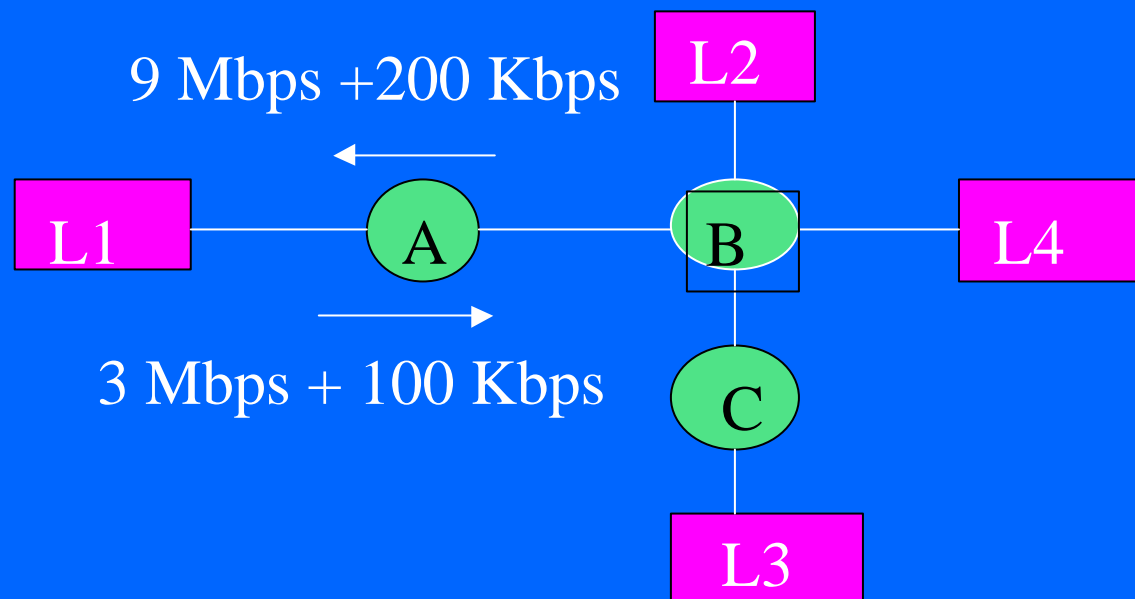
# Videokonferenssi, jossa 4 osallistujaa

- ◆ kullakin videokuva- ja ja audioyhteys muihin
  - videokuva tarvitsee 3 Mbps ja audioyhteys 100 Kbps



## Reitittimet varaavat seuraavasti:

- ◆ videokuvaa varten kullekin tulee  $3 * 3$  Mbps eli 9 Mbps ja kullakin lähtee 3 Mbps
- ◆ audioyhteyksiä varten riittää  $2 * 100$  Kbps (korkeintaan kaksi samanaikaista ääniyhteyttä) tulevaan ja 100 Kbps lähtevään audiovirtaan



- 
- ◆ Pääsytesti (admission test)
    - testaa, voidaanko varaus hyväksyä
    - jos ei => hylkäys
    - RSVP ei määrää millainen testin pitää olla
  - ◆ Polkusanomat (path messages)
    - lähettäjät ilmoittavat, mitä reittiä varaukset tulee lähettää
    - kulkevat monilähetyspuuta
    - reititin A kertoo IP-osoitteen ja lähetyksensä

# Varaustyylejä

---

- ◆ Tyyli ilmoittaa
  - saako varauksia yhdistää
  - keiltä lähettäjiltä halutaan vastaanottaa
- ◆ kolme eri varaustyyliä
  - kaikilta lähettäjiltä ja varattu kaista on kaikkien lähetysten yhteiskäyttöön
  - listan lähettäjiltä, kullekin ilmoitettu oma kaistaleveys
  - listan lähettäjiltä, kaista kaikkien yhteiskäytössä
- ◆ pakettiradio /videokonferenssi
- ◆ vain samalla tyylillä varattuja saa yhdistää

# Intservin ongelmia

---

- ◆ Intservissä QoS on vuokohtainen
  - ◆ resurssit varataan koko vuolle päästä päähän
  - ◆ palvelunlaatu on vuokohtainen
- ◆ resurssivaraukset ja kirjanpito jokaisesta reitittimen kautta kulkevasta vuosta
  - ◆ OC-3-linkillä noin 256 000 yhteyttä yhdessä minuutissa runkoreitittimellä!
- ◆ Joukko ennalta määriteltyjä palveluluokkia, ei näiden keskinäisiä eroja
  - » ensimmäinen luokka <=> turistiluokka
  - » platinakortti > kultakortti > standardiluottokortti

# Diffserv eli eriytyneet palvelut (Differentiated Services)

---

- ◆ Internetiin skaalautuva ja joustava palvelun eriyttäminen
  - » verkossa pystytään käsittelemään eri liikennettä eri tavoin
  - » uusia palveluluokkia voi syntyä ja vanhoja poistua
- ◆ ei määritellä eri palveluita eikä palveluluokkia
  - vaan toiminnalliset komponentit, joilla tällaiset palvelut voidaan toteuttaa

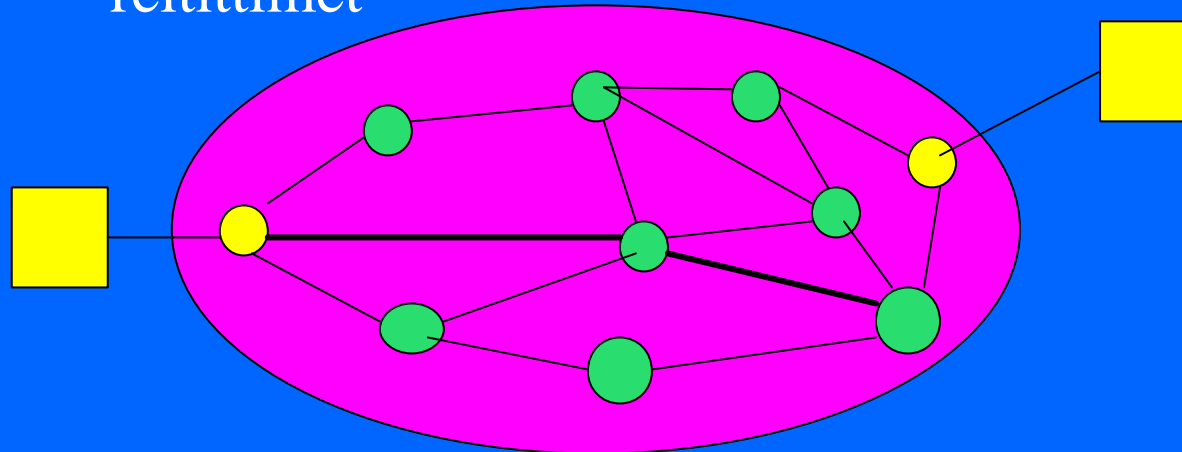
# Diffserv-arkkitehtuurin kulmakivet

---

## ◆ Kahdenlaisia toimintoja

– reunatoiminnot (edge functions)

» isäntäkoneet tai ensimmäiset diffserv-taitoiset reitittimet



– ydintoiminnot (core functions)

» muut reitittimet

# Reunatoiminnot

---

## ◆ Pakettien luokittelu

– merkitsee saapuneet paketit

» DS-kenttä (differentiated service) saa tietyn arvon

» merkintä kertoo, mihin liikenneluokkaan paketti kuuluu

◆ “behavior aggregate”

» eri merkinnöin varustetut paketit saavat eri palvelun verkon reitittimissä

## ◆ Liikenteen valvonta (traffic conditioning)

– merkitty paketti joko lähetetään heti verkkoon, sitä viivästetään tai se jopa hävitetään



# Ydintoiminnot

---

- ◆ Pakettien eteenpäin reitittäminen

- kun merkitty paketti saapuu diffserv-kykyiseen reitittimeen, se ohjataan eteenpäin paketin luokan mukaisesti kohdeltuna (per-hop behavior)

- ◆ miten paketti saa käyttöönsä linjakapasiteettia

- ◆ miten sitä kohdellaan purskureissa

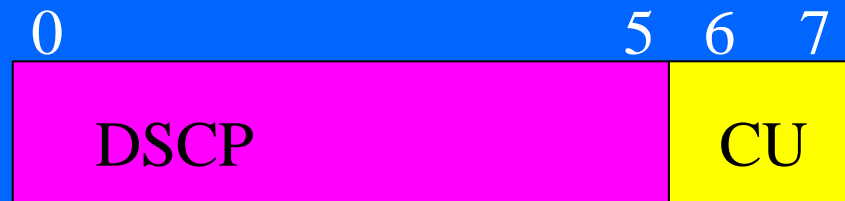
- » paketin kohtelu riippuu **vain** sen merkinnästä, ei sen kohteesta tai lähteestä

- ◆ ei tarvita tilatietoja eri yhteyksistä!

# DS-kenttä

---

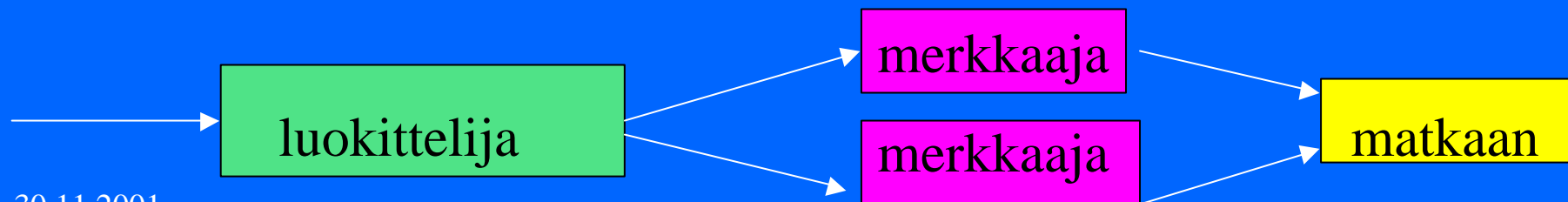
- ◆ IPv4: TOS-kenttä (Type of Service)
- ◆ IPv6: liikenneluokkakenttä (Traffic Class Field)



- DSCP (Differentiated service code point)
  - CU ei toistaiseksi käytössä
- ◆ DS-kenttä määrää paketin kohtelun muissa reitittimissä

# Pakettien luokittelu ja merkkkaus

- ◆ Luokittelija lajittelee paketit jonkin kentän perusteella
  - » lähde- tai kohdekone,
  - » lähde- tai kohdeportti
  - » protokolla, jne
- ◆ ja lähettää ne kyseisen luokan merkkaajalle, joka laittaa DS-kenttään sopivan arvon



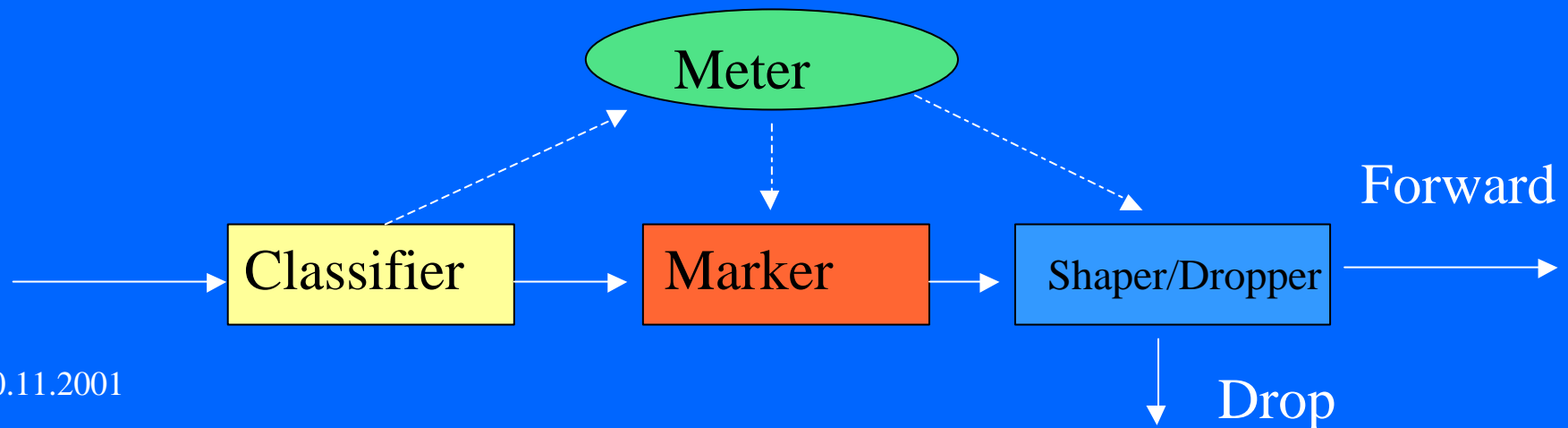
# Liikenneprofiili ja liikenteen valvonta

---

- ◆ Lähettäjä voi myös sopia käytetystä liikenneprofiilista
  - » huippunopeus
  - » purskeisuus
- ◆ jos lähetys poikkeaa sovitusta , niin
  - » ne voidaan meritä eri tavoin
  - » niitä voidaan viivyttää
  - » tai ne voidaan hävittää

# Liikenteen mittaus

- ◆ Liikenteenmittaaja vertaa pakettivuota sovittuun ja päättelee onko se sovituksen mukaista
  - » Diffserv-arkkitehtuuri ei määrittele mitä poikkeavan vuon paketeille tapahtuu



# Ydintoiminnot (Per-Hop Behaviors)

---

- ◆ Ulkoisesti havaittava eri käsittely eri luokan paketeille
  - eri luokan paketeille eri suorituskyky
    - ◆ mitattavissa oleva ominaisuus
- ◆ Voidaan toteuttaa eri menetelmin
  - ◆ etuilua puskurijonoissa
  - ◆ taataan tietty prosentti linkkikapasiteetista
  - nopeutettu edelleenlähetys
    - ◆ aina vähintään tietyllä nopeudella eteenpäin
  - taattu edelleenlähetys
    - ◆ eri luokkia, joista kullekin vähintään tietty määrä puskurikapasiteettia ja kaistanleveyttä
    - ◆ luokkien sisällä kolme eri ‘pudotusluokkaa’



LOPPU  
HYVIN,  
KAIKKI  
HYVIN!

TILI II