

# 7. Palvelun laatu (QoS) Internetissä

---

- ◆ Sovellus ei saa mitään takuita palvelun laadusta: IP tarjoaa tasapuolisen palvelun (best effort) kaikille)
  - joskus kaikki toimii hyvin, joskus ei
  - sovellus ei voi paljoa siihen vaikuttaa
- ◆ Tällainen palvelu ei sovi monille multimediasovelluksille!
- ◆ QoS-ajattelu myös Internetiin?
  - viive, viipeen vaihtelu
  - virheettömyys

# Tosiaikainen interaktiivinen lähetys

---

- ◆ viiveherkkä (delay-sensitive)
- ◆ huojuntaherkkä (jitter-sensitive): viiveen vaihtelu
- ◆ kohtalaisesti hävikkiä sietävä (loss-tolerant)
  - Internet-puhelin (viive 150-400 ms, hävikki 10-20 %)
  - videokonferenssi (viive muutama sata ms)

# Internet-puhelin

---

Lähetetään vain puheryöppyjä ei taukoja:

- 20 ms välein 160 tavun lohkoja



Vastaanotossa viivettä, viiveen vaihtelua (=> huojuntaa) sekä **virheellisiä** tai **puuttuvia** lohkoja

# Datavirta (stream)

---

- ◆ video (tilausvideo, urheilukilpailun suora lähetys), audio (musiikin lataus, sinfoniaorkesterin konsertin suorälähetys)
- ◆ sietää jonkinverran sanomien katoamisia
  - FEC-korjaus, interpolointi, entisen toistaminen
- ◆ aikarajat, joita noudatettava
  - uudelleenlähetys ei yleensä toimi
- ◆ Tasainen, riittävän nopea lähetys
  - TCP:n ruuhkanhallinta haittaa
  - UDP-segmentit saapuvat epäjärjestyksessä ja voivat juuttua ruuhkaan

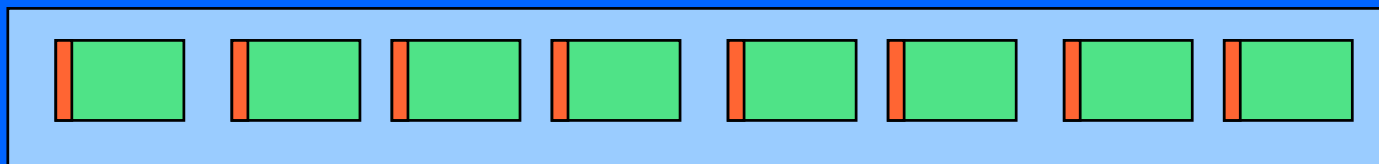
# RTP (Real-time Transport Protocol) (RFC 1889)

---

- ◆ välineet, joilla vastaanottaja huomaa puuttuvat sanomat ja pystyy rekonstruoimaan sanomien lähetysjärjestyksen
  - järjestysnumerot, aikaleimat
  - kuormatiedot (payload)
- ◆ ei takaa sanomien saapumista ajoissa eikä muuta palvelunlaatua (qos)
- ◆ yleensä toimii UDP-protokollan päällä
- ◆ ei täydellinen protokolla; ei omaa kerrosta, vaan osa sovellusta
  - » profiilimäärittely: eri sovelluksille tarpeelliset lisät
  - » kuoman muoto: miten tietty koodaus kuljetetaan RTP:ssä

# RTP: Datakanava ja kontrollikanava

- ◆ datakanavassa kulkee pieniä datapaketteja



» joihin on liitetty RTP-otsake

- ◆ kontrollikanavassa RTCP-sanomia (Real-time Transport Control Protocol)
  - lähettäjien ja vastaanottajien välillä
  - lähetys- ja vastaanottoraportteja

# RTP-datavirrat ja RTP-istunto

---

- ◆ videokonferenssi
  - neljä eri datavirtaa:
    - » kaksi äänelle; yksi kumpaankin suuntaan
    - » kaksi videokuvalle; yksi kumpaankin suuntaan
  - yksi yhdistetty, koodattu datavirta (esim. MPEG1 (Moving Picture Experts Group) ja MPEG2)
- ◆ myös monelta lähettäjältä monelle vastaanottajalle
  - lähettäjillä sama monilähetysryhmä
- ◆ RTP-istunto
  - yhteenkuuluvat datavirrat
    - » esim. videokonferenssin audio- ja videodatavirrat

# RTP-otsakkeen kentät

---

kuorma- tyyppi	järjestys- numero	aikaleima	lähteen tunniste	muita kenttiä
7 bittiä	16 bittiä	32 bittiä	32 bittiä	

Kuorman tyyppi audiolle tai videolle ilmoittaa äänen tai videokuvan koodaustavan (esim. PCM tai MPEG1); lähettäjä voi ilmoittaa koodauksen muuttumisesta myös lähetyksen keskellä!

Jokainen lähetetty paketti kasvattaa järjestysnumeroa yhdellä



- 
- ◆ aikaleima kertoo RTP-paketin 1. tavun otoshetken. Aikaleimakello käy, vaikka lähde ei lähetäkään.
  - ◆ lähteen tunnistekenttä
    - jokaisella datavirralla on satunnainen tunnistenumero; jos jo käytössä valitaan uusi
  - ◆ sekalaisia muita kenttiä

# RTP:n toteuttava sovellus

---

- ◆ voidaan kirjoittaa käsin
  - RTP-kapselointi lähettäjän puolella ja purkaminen vastaanottajalla
- ◆ RTP-kirjastojen (C) ja valmiiden Java-luokkien avulla
  - Java Media Framework (JMF) sisältää täydellisen RTP-toteutukse

# RTCP (Real-time Transport Control Protocol) (RFC 1889)

---

- ◆ kaikki RTP-istuntoon osallistuvat voivat lähettää RTCP-sanomia
  - yleensä käyttäen RTP-istunnon yhteistä monilähetysosoitetta
  - paketteja lähetetään ajoittain
  - niissä kerrotaan lähetyksiin liittyviä tilastoja (montako pakettia lähetetty, paljonko havaittu huojuntaa)
  - standardi ei määrittele, mitä sovelluksen pitää tälle datalle tehdä

# RTCP (Real-time Transport Control Protocol) (RFC 1889)

---

## ◆ viisi erilaista sanomaa

### – **Sender report:**

- » Aktiivisen lähettäjän tiedot tietyn aikavälin lähetyksistä ja vastaanotoista
- » Absoluuttinen aikaleima (sekunteja 1.1.1970 lähtien), jotta pystytään yhdistämään audio- ja videovirrat

### – **Receiver report**

- » Passiivinen vastaanottaja raportoi palvelunlaadusta (”miten suuri osa jäi saapumatta”, ”keskim. huojunta” jne )

### – **Source description message**

- » Lähde kertoo itsestään: lähteen ja sen valvojan kontaktitietoja

### – **Bye message:** lähde ilmoittaa lopettavansa lähetyksen

### – **Application-specific message:** voi määritellä uusia sanomatyyppejä

# RTCP:n skaalausongelma

---

- ◆ RTP-pakettien määrä ei kasva, vaikka vastaanottajien määrä kasvaa, mutta RTCP-pakettien määrä kasvaa
  - » kukin lähettäjä lähettää omat raporttinsa
- ◆ RTCP-liikenne korkeintaan 5% kaistan kapasiteetista
  - 75% vastaanottajille ja 25 % lähettäjille
  - kukin vastaanottaja saa saman osuuden

# UDP kuljettaa sekä RTP- että RTCP-sanomia

---

- ◆ tilapäinen porttinumero
  - RTP: parillinen numero
  - RTCP: seuraava parillinen numero
- ◆ UDP-segmentit kuljetetaan IP-datagrammeissa

# Istunnon kontrolliprotokollat

---

- ◆ video- tai audioistunnon kontrollointi
  - yhteyden muodostus
  - yhteyden ylläpitäminen
  - yhteyden purkaminen
- ◆ Esim. Videokonferenssikokous
  - Sovittava käytetty koodaus ja kuljetusprotokolla
  - Tiedettävä osoitteet ja portit
  - Valvottava kokouksen aikana

# SIP (Session Initiation Protocol) (RFC 3261)

---

- ◆ IP-puheluiden soittamiseen ja vastaanottamiseen
  - vastaanottajan IP-osoitteen selvittäminen
    - » tilapäisiä osoitteita
    - » useita eri laitteita, joilla eri IP-osoite
  - puhelun aloittaminen , koodauksesta sopiminen ja puhelun päättäminen
  - puhelun hallinta
    - » uusien mediavirtojen lisääminen
    - » koodauksen muuttaminen
    - » uusien osallistujien mukaanottaminen
    - » call transfer, call holding
- ◆ myös videokonferenssit ja muut multimediatyhteydet
  - monen pelaajan pelit



- 
- ◆ puhelinnumerot esitetään SIP URL:eina
    - » sip: pekka@cs.helsinki.fi
    - » sip: maija@195.156.43.67
    - » sip: simo@358 0 19144240
  - ◆ protokolla HTTP-tyyppinen tekstipohjainen
  - ◆ INVITE, ACK, BYE, OPTIONS, CANCEL, REGISTER
  - ◆ + paljon muita piirteitä

# SIP-rekisteröijä

---

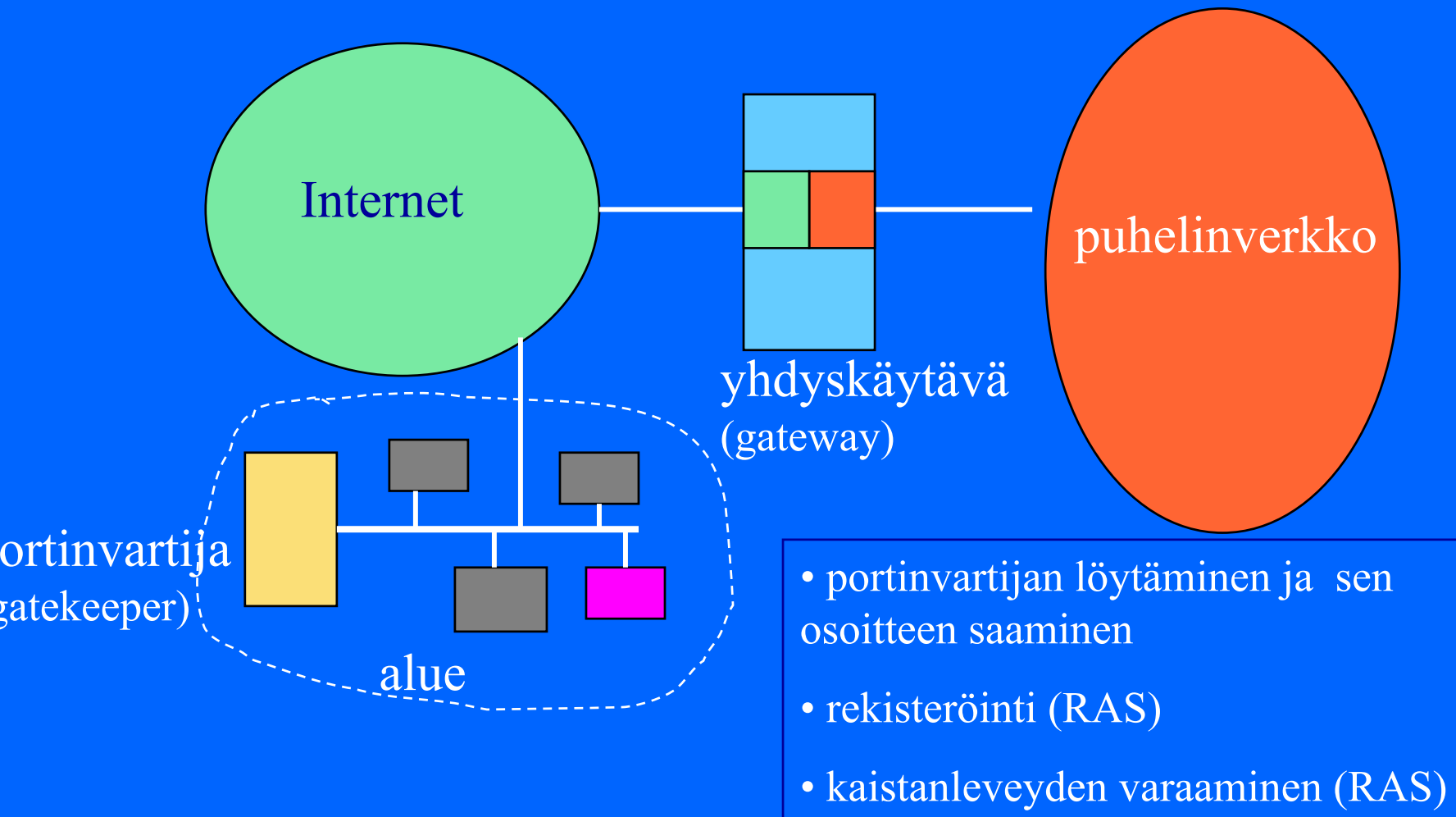
- ◆ on SIP-laitteessa
  - kun käyttäjä käynnistää SIP-sovelluksen, sovellus ilmoittaa oman IP-osoitteensa rekisteröijälle
    - » 'käyttäjä on tavoitettavissa täältä'
  - ilmoitus vahvistetaan tietyin väliajoin ja kun käyttäjä siirtyy uudelle laitteelle, laite ilmoittaa uuden osoitteen
  - uudelleenohjaus toiseen rekisteriin, jos käyttäjä ei ole enää rekisteröitynyt

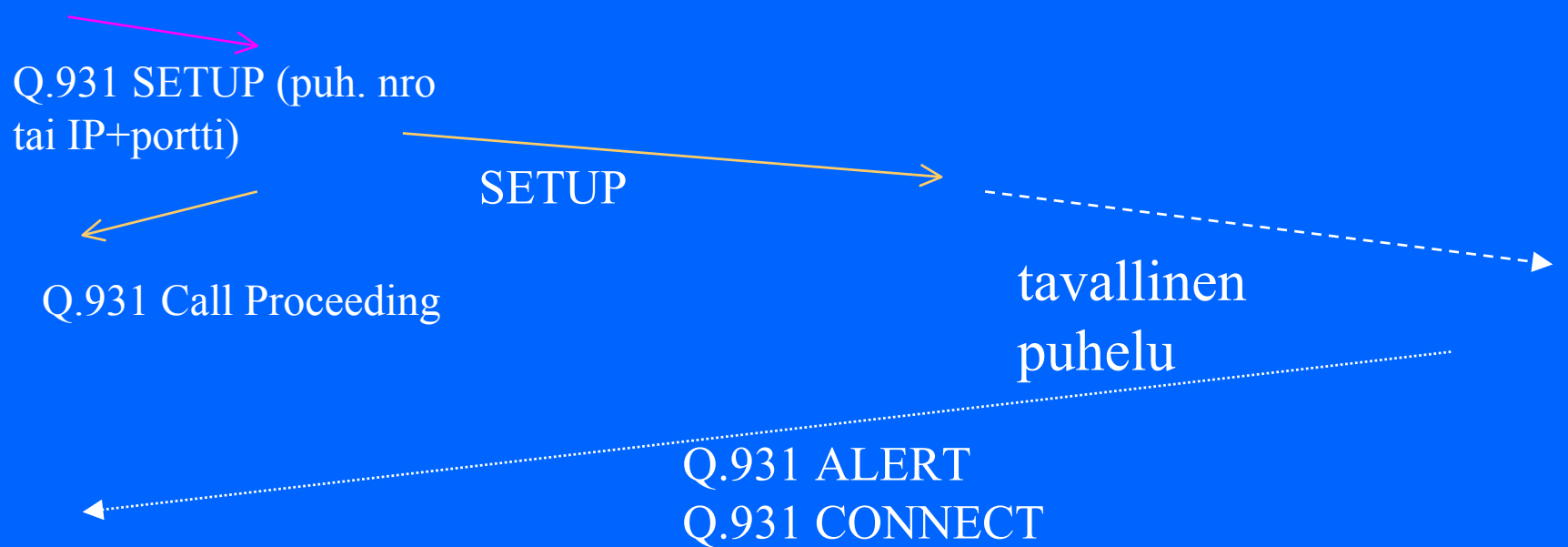
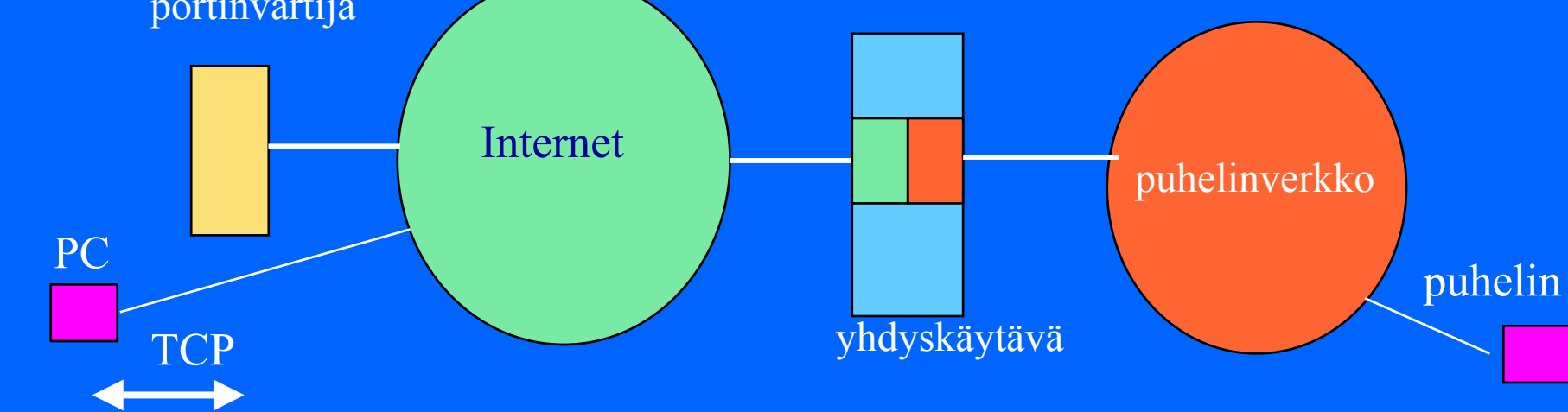
# H.323 (Visual Telephone systems and Equipment for Local Area Networks Which Provide a Non-Quaranteed Quality of Service)

---

- ◆ arkkitehtuuri IP-puheluun ja videokonferenssiin tarvittavista standardeista
  - ITU:n vaihtoehto SIP-protokollalle (*KISS; 250 sivua*)
  - sisältää suuren joukon eri toimintoihin tarvittavia
    - » puheenkoodaus (PCM) ja tiivistys
    - » puhelunvalvonta
    - » puhelun signallointi
    - » rekisteröinti
    - » tiedonsiirto (RTP) ja sen valvonta (RTCP)
  - monimutkainen ja laaja standardi (1400 sivua)
  - Ensimmäisten Internet-puhelinjärjestelmien käyttämä

# H.323:n Internet-puhelimen arkkitehtuuri

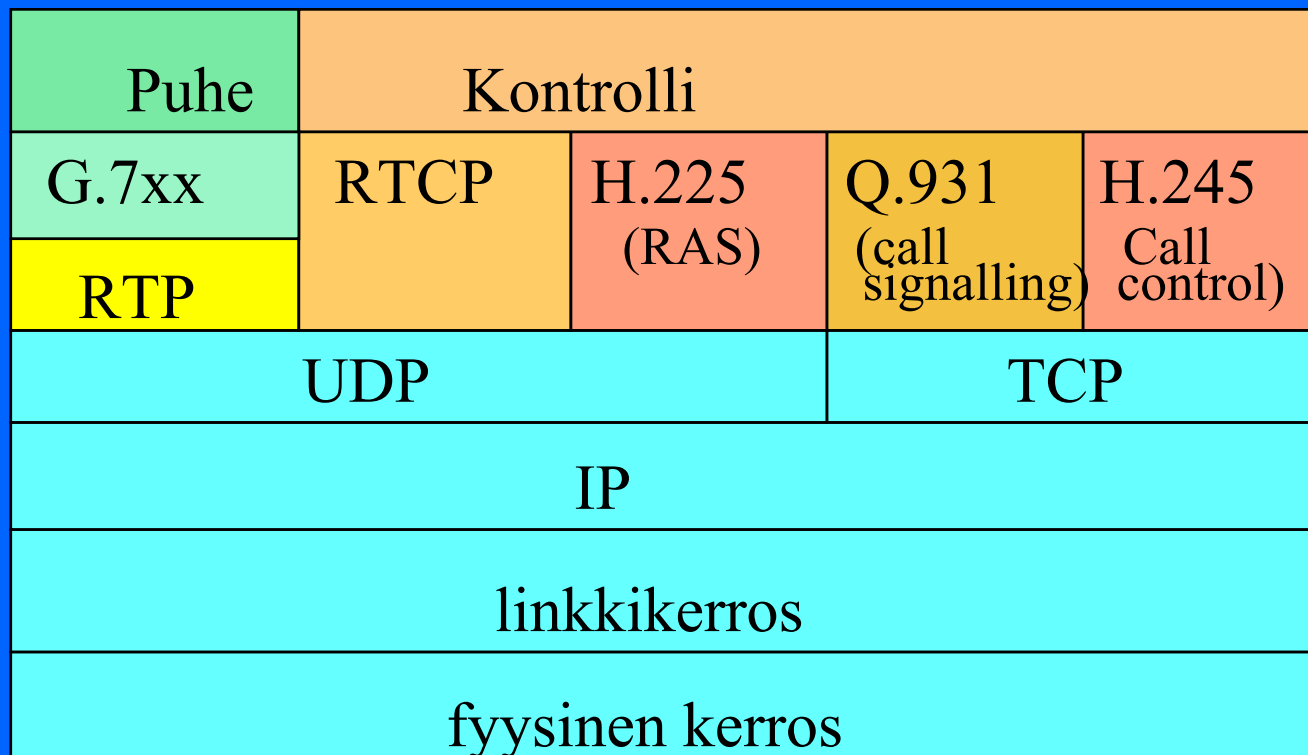




Tuloksena 'bittiputki' puhelimen ja PC:n välillä (fyysinen taso)

- 
- ◆ puhelun parametrit H.245-protokollan avulla (oma kanava)
    - mitä ominaisuuksia pystyy hoitamaan
  - ◆ tämän jälkeen 2 datakanavaa, jossa RTP-protokollan viestit kulkevat
  - ◆ + kontrollia varten RTCP-kanava
  - ◆ puhelu puretaan Q.931-signallointikanavaa käyttäen

# H.323-protokollapino



G.7XX äänen koodaus; G711 = PCM + muita koodausstadardeja

RAS (registration/admission/status): liittyminen, kaistan pyytäminen yms

# (Musiikki)mediapalvelimen tehtävät

---

- ◆ käyttäjäliitännät
  - stereoliitännän kaltainen, usein räätälöityissä
- ◆ siirtovirheet
  - parittomat ja parilliset näytteet eri paketeissa (a' 5 ms)
  - interpolointi
- ◆ musiikin purkaminen
- ◆ huojunnan estäminen
  - puskurin avulla

Huom. MP3 esitys on noin 4 MB, 56 kbps linjalla sen siirtäminen kestää noin 10 minuuttia. => lähetysvirta palvelimelta

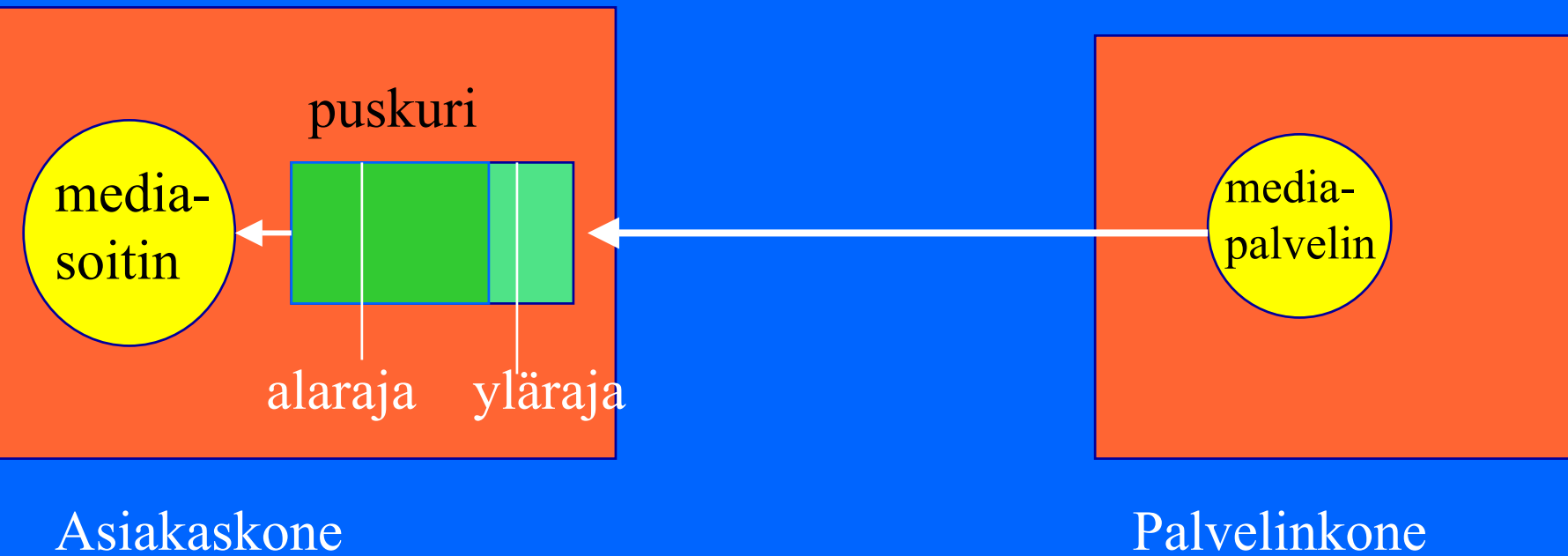


# RTSP (Real-time Streaming Protocol) (RFC 2326)

---

- ◆ lähetysvirran esittämisen kontrollia ('kaukosäädin' soittimen ja mediapalvelimen välissä)
  - esittäminen (play) ja pysäyttäminen (pause)
  - yhteyden muodostaminen palvelimeen (setup) ja sen purkaminen (teardown)
  - tallentaminen (record)
  - mediaparametrien listaus (describe)

# Puskurointi tarpeen (10 –15 ms)



Kun puskurin yläraja ylittyy, niin mediasoitin pyytää mediapalvelinta keskeyttämään lähetyksen, kun alaraja alittuu, niin mediasoitin pyytää mediapalvelinta lähettämään lisää.

## 7.2 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

# Palvelunlaadun varmistaminen

---

- ◆ Mitä vaaditaan, jotta qos mahdollista
  - reitittimen on pystyttävä erottamaan eri prioriteetin paketit ja datavirrat
  - erilainen käsittely reitittimissä eri prioriteetin paketeille
  - 'väärinkäyttäytyvät' datavirrat eivät saa häiritä toisia
  - kapasiteetin varaaminen
  - tehokkuus ei saa kärsiä

# Reitittimen jonot

---

- ◆ FIFO-jono: ei erottele prioriteettejä
- ◆ prioriteettijonot: kullekin prioriteetille omansa
- ◆ vuorotellen (round robin)
- ◆ painotettu vuorotellen: kullakin prioriteetilla oman painotus (eri aikamäärä)

# IntServ eli Integroidut palvelut (Integrated 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

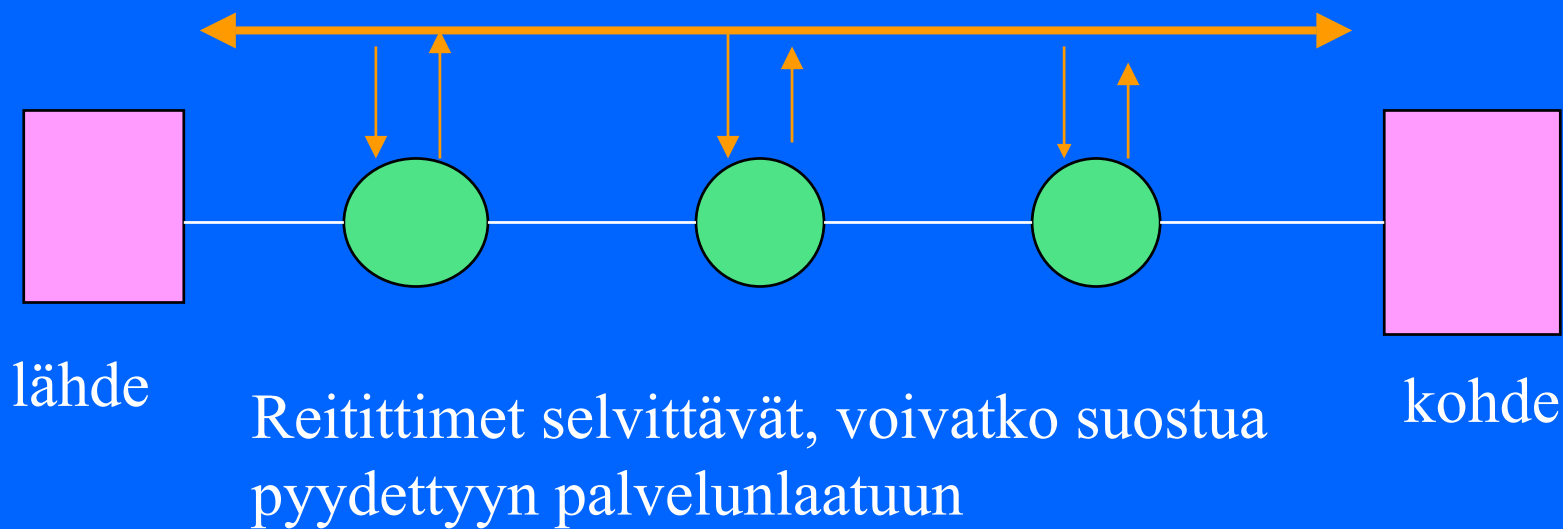
# Takuu perustuu resurssien varaamiseen

---

- ◆ 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

# 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 ämpärin kokoiset
- ◆ joustavampi kuin vuotava ämpäri
  - » purskeet voivat aiheuttaa ruuhkaa => vuotava ämpäri vuoromerkkiämpäriin perään

# Kahdenlaista palvelua

---

## ◆ **Taattu palvelu** (quaranteed 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”

# Taatun palvelun perusidea:

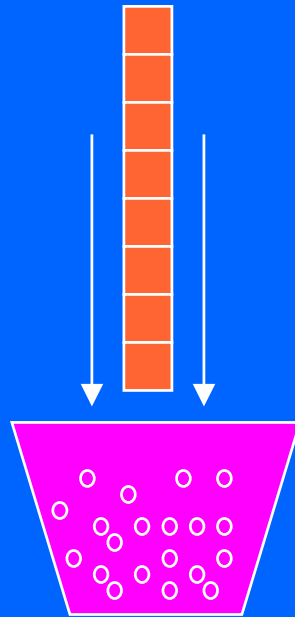
---

- ◆ Liikenne kuvataan vuoromerkkiämpärin avulla
  - lähetyslupien määrä sekunnissa =  $r$
  - vuoromerkkiämpärin 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  $r*t+b$  bittiä
  - Jos siirtonopeus jonosta on vähintään  $R$  ( $>r$ ), niin maksimiviive on korkeintaan  $b/R$

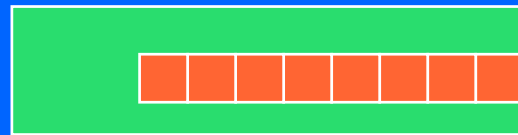
Oletetaan  
alkutilanteeksi  
ämpäri täynnä  
lupia.

$$R' > r$$

Maksimissaan voi  
tulla  $b$ :n paketin  
kokoinen ryöppy



Ryöpyn kaikki  $b$   
pakettia siirtyvät  
reitittimen  
ulosmenojoonoon



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<sub>41</sub>

- 
- ◆ 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. videolähetys usealla 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

---

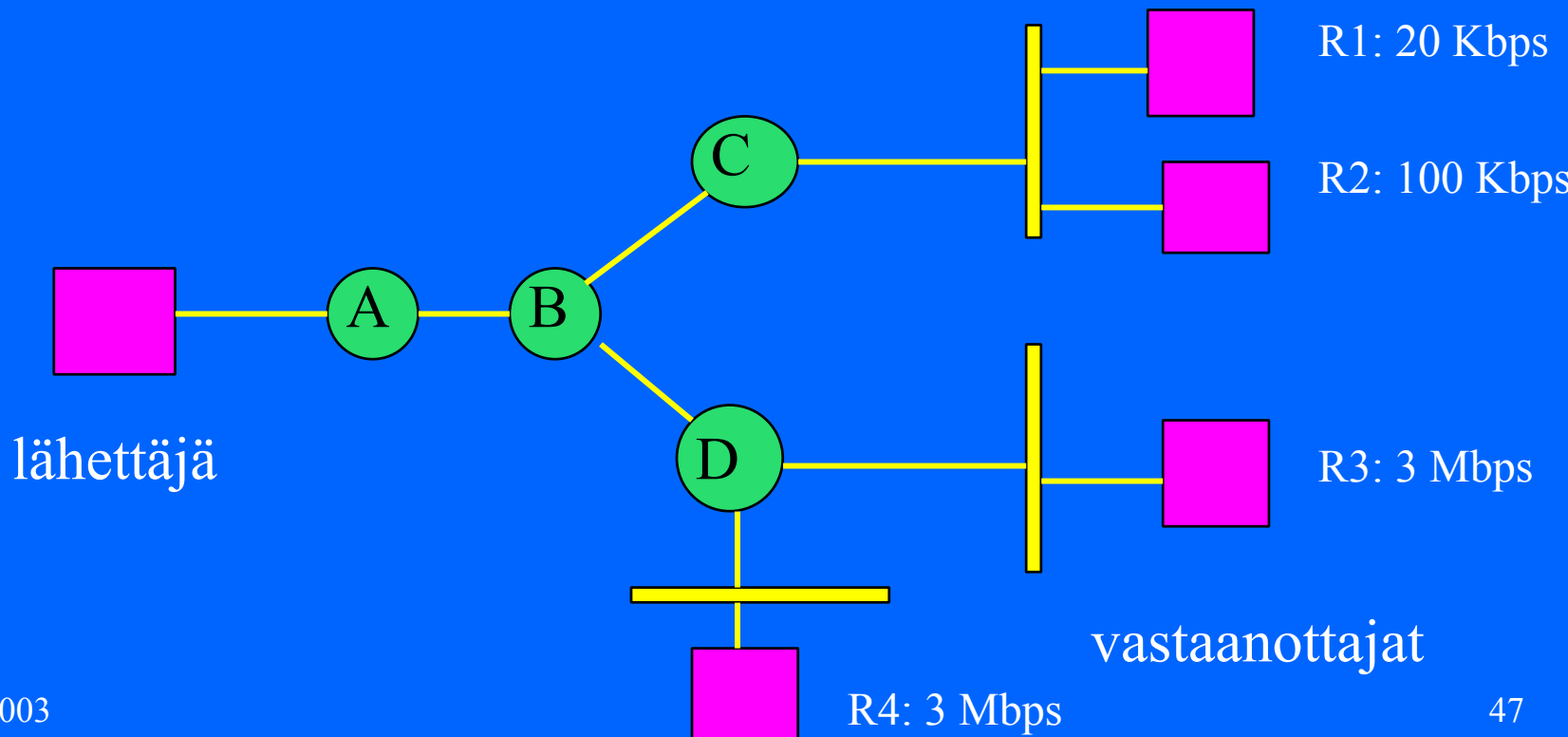
- ◆ Tietovuon vastaanottajat voivat olla hyvin heterogeenisia
  - pystyvät vastaanottamaan eri nopeudella
    - ◆ Videota voidaan vastaanottaa nopeudella 28.8 Kbps, 128 Kbps tai 10 Mbps
    - ◆ koodataan video useana eri kerroksena
  - lähettäjän tarvitsee tietää vain vastaanottajajoukon korkein siirtonopeus

# 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 vuonimiö => vuo
  - lähettäjä lähettää usealle vastaanottajalle videokuva kilpailusta
    - ◆ joka paketissa monilähetysosoite => vastaanottajat

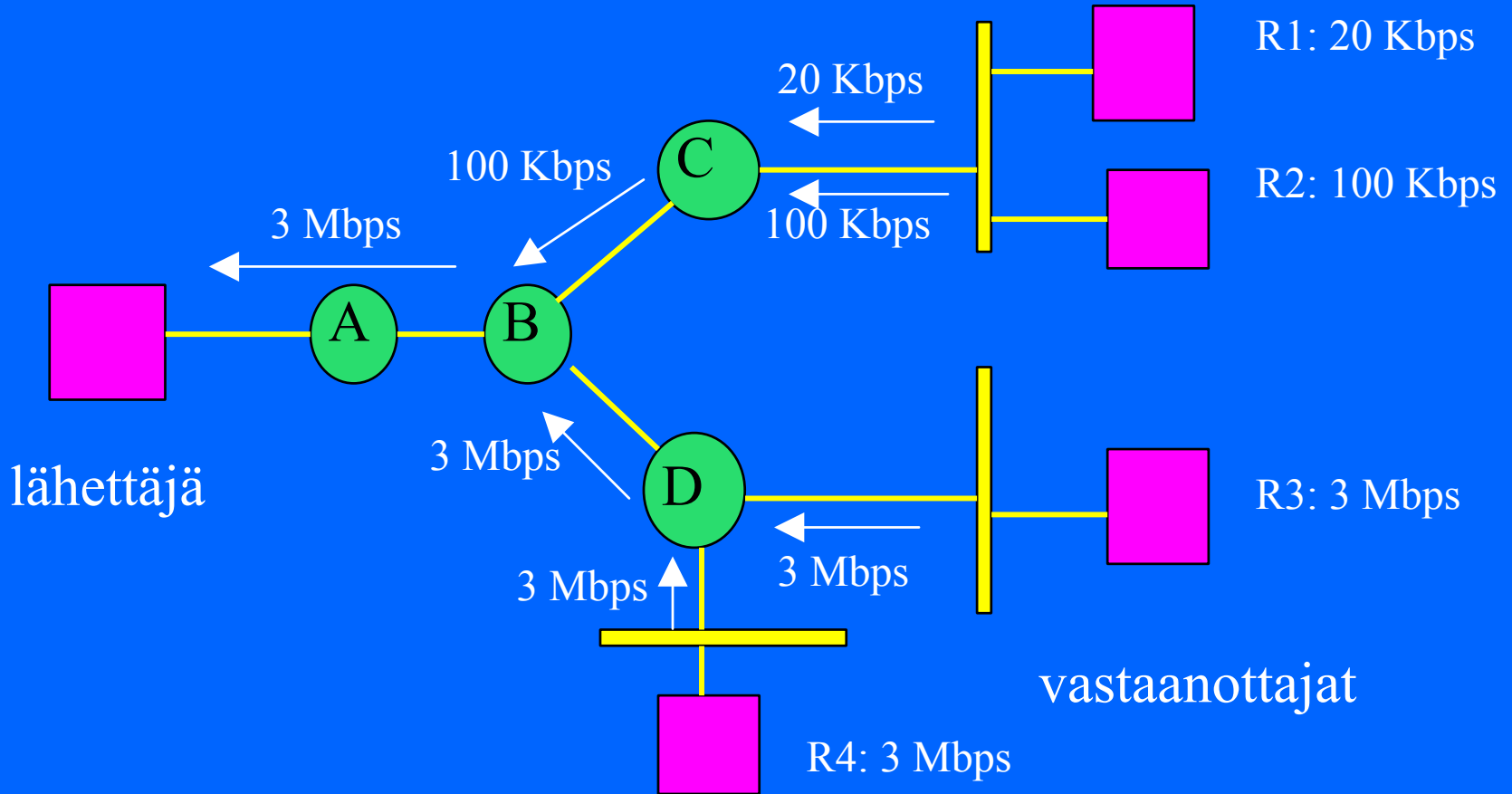
- ◆ Monilähetysprotokolla on muodostanut monilähetyspuun lähettäjältä vastaanottajille



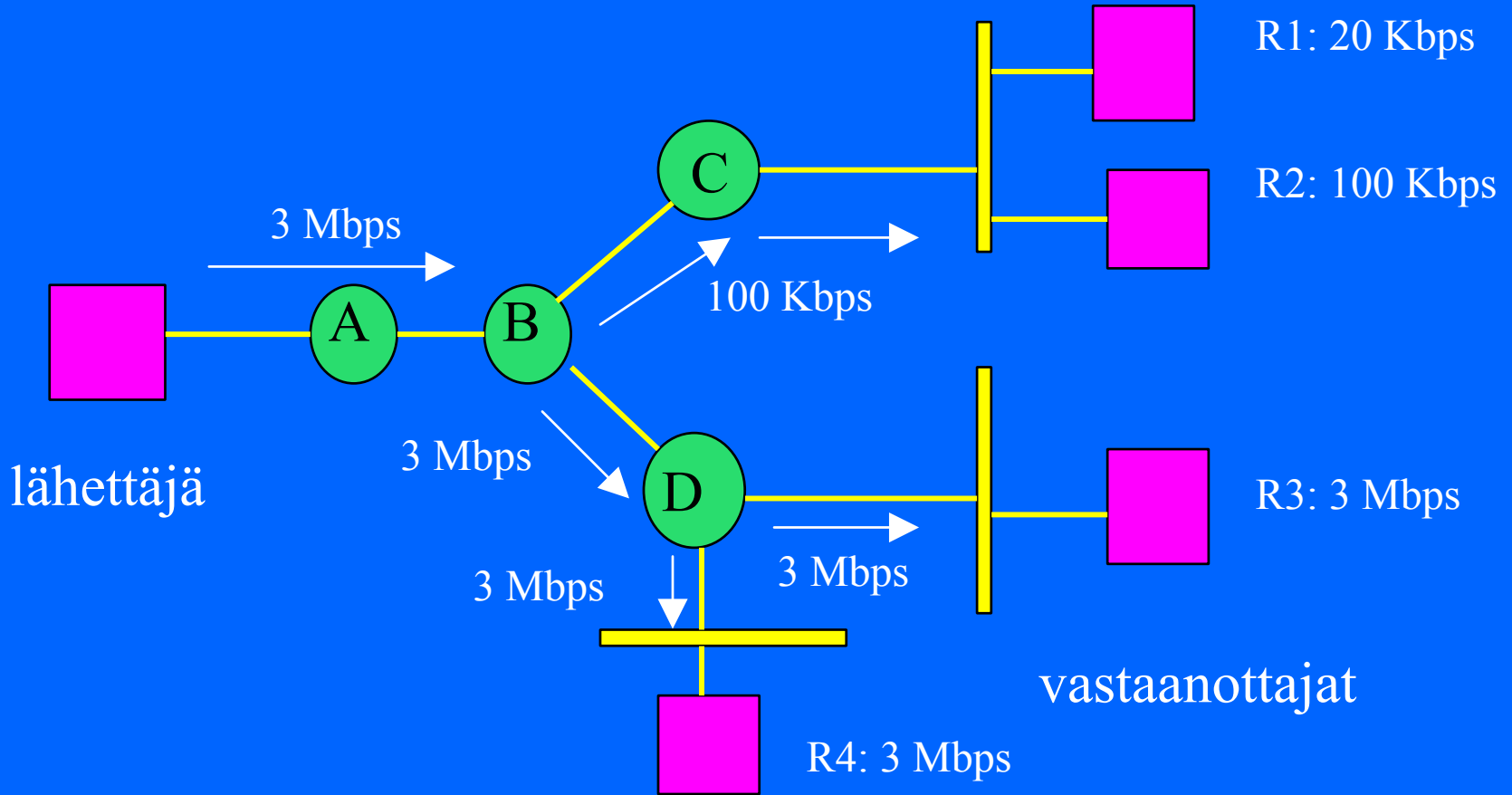
- 
- ◆ Jokainen vastaanottaja lähettää varaussanomaa
    - ◆ käyttäen esim. reverse path forwarding algoritmia
    - ◆ kertoo millä nopeudella haluaa vastaanottaa lähettäjältä
  - ◆ sanoman saanut reititin varautuu antamaan pyydetyn kapasiteetin
    - ◆ pakettien skeduloija
  - ◆ reititin lähettää eteenpäin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



# Varaussanommat

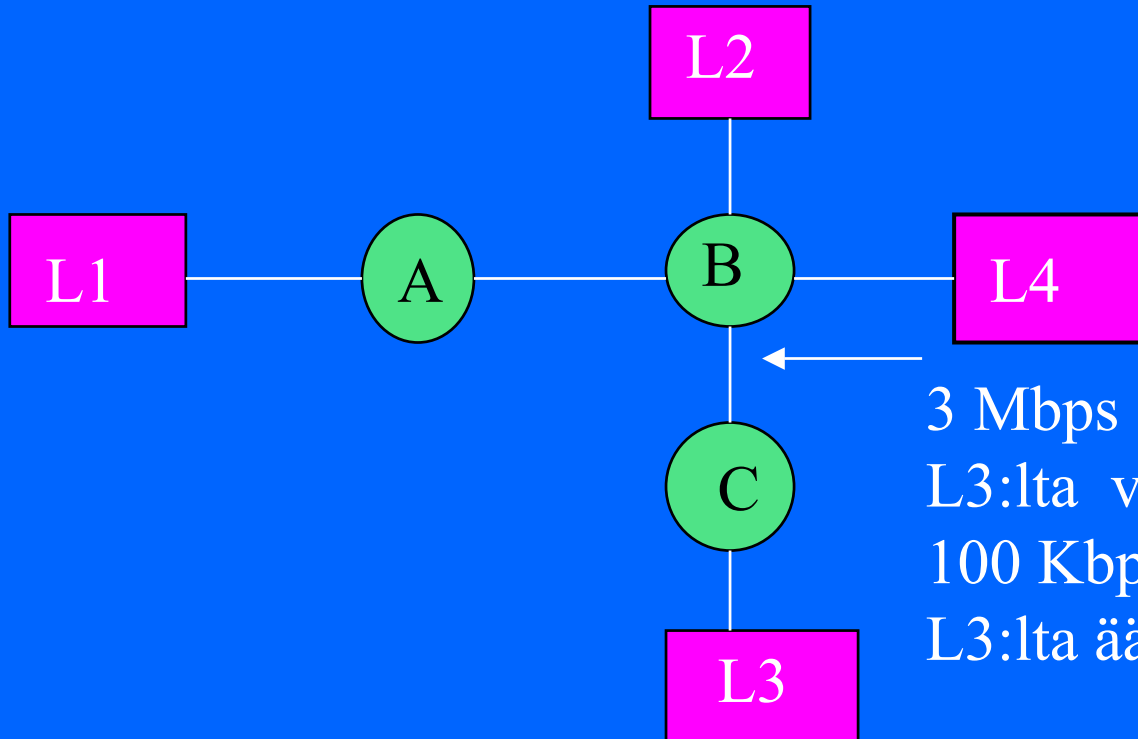


# Tehdyt varaukset



# Videokonferenssi, jossa 4 osallistujaa

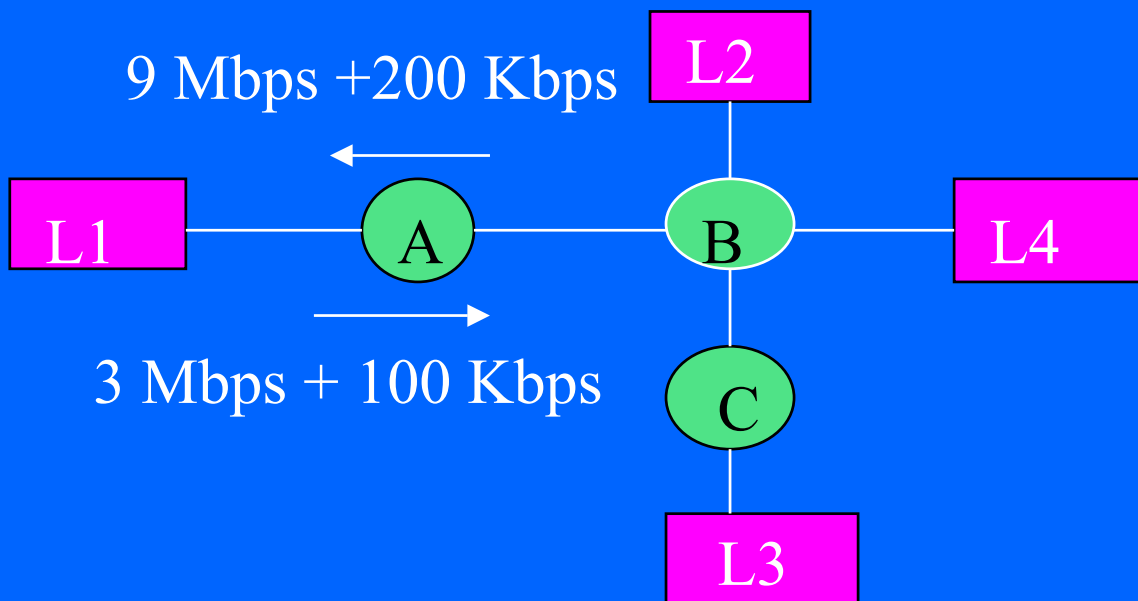
- ◆ kullakin videokuva- ja audioyhteys muihin
  - videokuva tarvitsee 3 Mbps ja audioyhteys 100 Kbps



3 Mbps L1:ltä, L2:lta ja L3:lta videokuva varten  
100 Kbps L1:ltä, L2:lta ja L3:lta ääntä varten

# Reitittimet varaavat seuraavasti:

- ◆ videokuvaa varten kullekin tulee  $3 * 3$  Mbps eli 9 Mbps ja kultakin 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-osoitteensa ja lähetyksensä

# Varaustyyliä

---

- ◆ 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

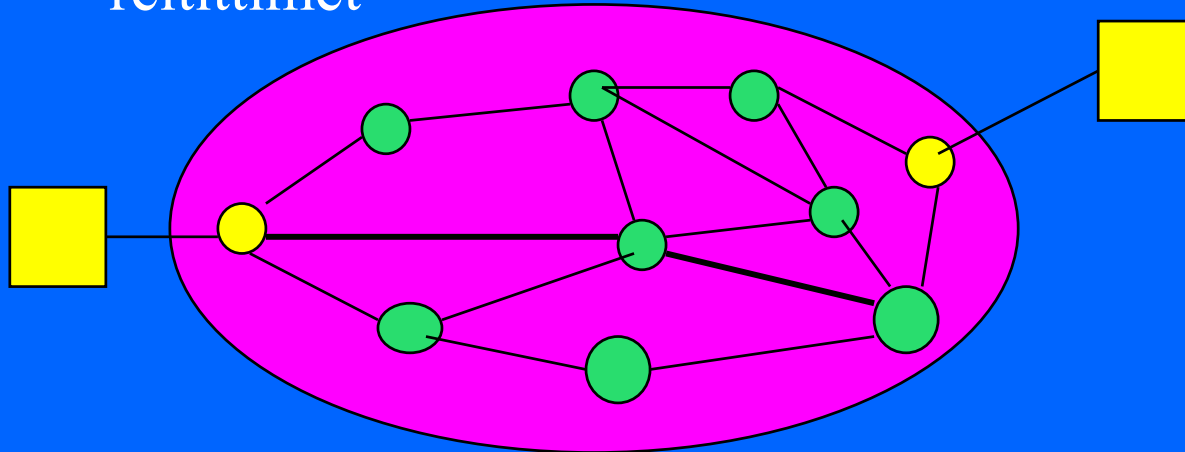


# Diffserv-arkkitehtuurin kulmakivet

## ◆ 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 linjakapasiteettiä
- ◆ miten sitä kohdellaan puskureissa

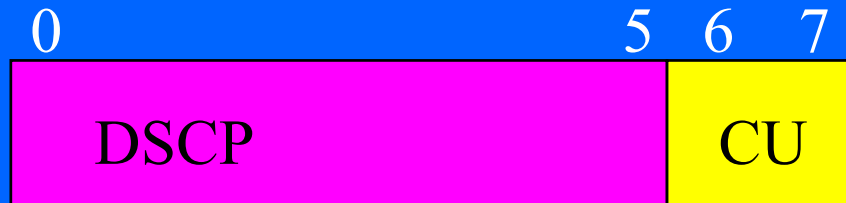
» paketin kohtelu riippuu **vain** sen merkinnästä, ei sen kohteesta tai lähteestä

- ◆ ei tarvita tilatietoja eri yhteyksistä!

# DS-kenttä

---

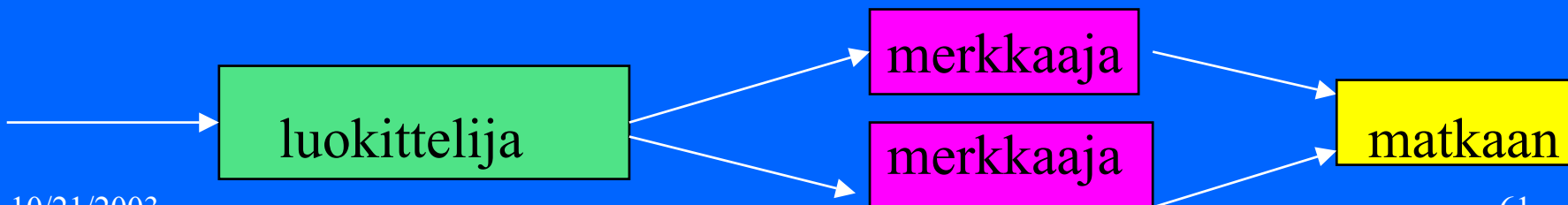
- ◆ IPv4: TOS-kenttä (Type of Service)
- ◆ IPv6: liikenneluokkakenttä (Traffic Class Field)



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

# Pakettien luokittelu ja merkkaus

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



# Liikenneprofiili ja liikenteen valvonta

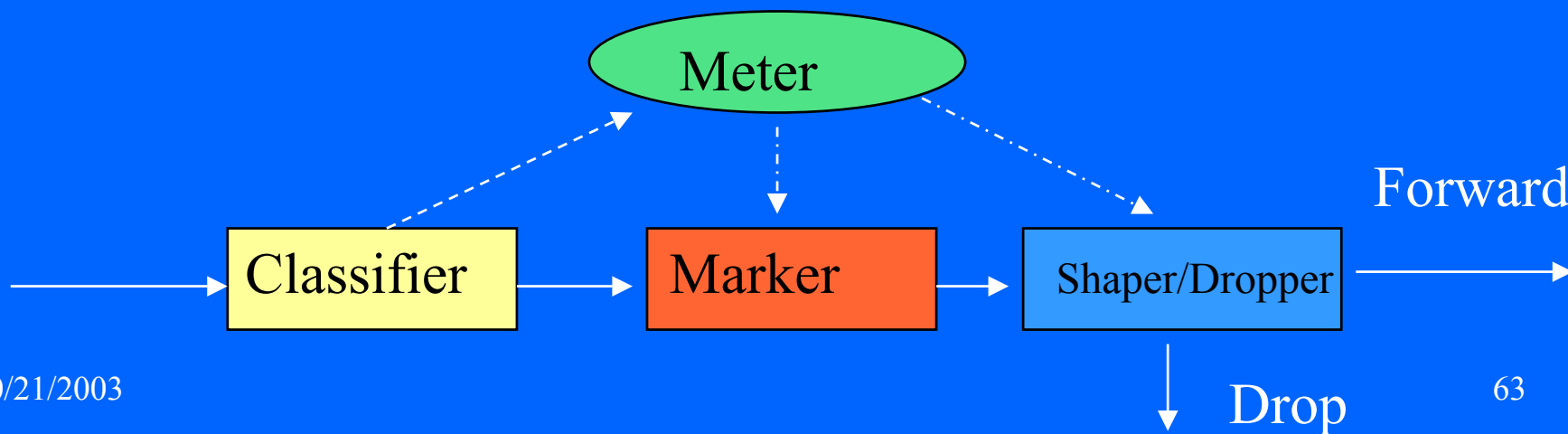
---

- ◆ 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’



# Diffserv-kritiikkiä

---

- ◆ vuosien varrella useita yrityksiä tuoda QoS pakettiverkkoon
  - atm-verkko
  - TCP + RSVP
- ◆ Diffserv usean teleoperaattorin välillä
  - yhteistyö tarpeen
- ◆ Laskutus, tarkistukset, rangaistukset
- ◆ Onko palveluiden välillä oikeasti eroa?