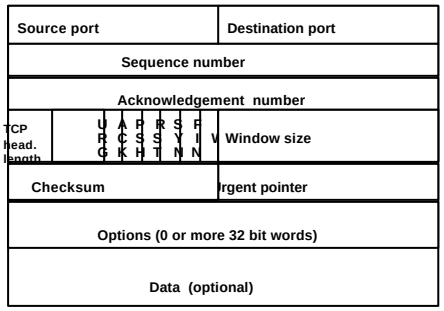
TCP

- TCP:n peruspiirteiden toiminta tarkemmin
  - osin vain harjoitustehtävissä
- TCP:n uusia piirteitä
  - SACK
  - Window scaling
  - time stamping
  - RED (Random Early Detection)
  - ECN (Explicit Congestion Notification)

TCP-otsakkeen kentät



TCP-optiot

- Optio-kenttä
  - erilaisia valinnaisia piirteitä varten
  - Option pituus on 40 tavua
    - » data offset -kenttä = 4 bittiä kertoo otsakkeen pituuden 32 bitin sanoina => 15*4 tavua = 60 tavua
    - » 60 tavua -20 tavua vakio-otsaketta => enintään 40 tavua optioita varten
  - käytetään ilmoittamaan maksimi segmentin koko
  - muita uusia piirteitä
    - » aikaleimaus (timestamp)
    - » ikkunan skaalaus (window scaling factor)

Erilaisia suorituskykyongelmia

- TCP-protokolla käytössä hyvin erilaisissa ympäristöissä
  - » pitkän viipeen satelliittiyhteyksillä
  - » erittäin nopeilla yhteyksillä
  - » langattomilla yhteyksillä
- => suorituskykyongelmia
  - otsakkeen kentät liian pieniä
    - » ikkunankoko 16 bittiä
      - rajoittaa lähetysnopeutta satelliittiyhteyksillä
    - » järjestysnumero 32 bittiä
      - rajoittaa lähetysnopeutta erittäin nopeilla yhteyksillä

Ikkunan skaalaus (Window scale factor)

- ikkunakoko = 16 bittiä => 65536 tavua
  - » vuonvalvonnassa
  - » kertoo vastaanottajan ikkunan = kuinka monta tavua voi lähettää ennenkuin täytyy jäädä odottamaan kuittausta
- jos käytössä ikkunan skaalaus -optio, ikkunakentän arvo kerrotaan 2**F, jossa F on skaalausoption arvo.
  - » Suurin F:n arvo on 14
- käytetään vain yhteyden aloituspyynnössä

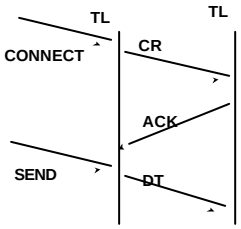
Aikaleima (timestamp)

- Kaksi eri optiota
  - Timestamp Value
    - » lähtevissä segmenteissä,
  - Timestamp Echo Reply
    - » kuittauksessa
    - » sama kuin kuitatun segmentin Timestamp arvo
- Voidaan käyttää missä tahansa datasegmentissä
- => kiertoviiveiden jatkuva tarkkailu helpompaa

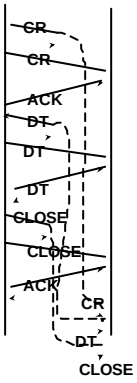
Yhteyden muodostus

- perusmalli hyvin yksinkertainen
- ongelmana viivästetyneet kaksoiskappaleet
  - » esim. yhteys pankkiin laskun maksamiseksi
  - » lasku maksetaan useaan kertaan
- =>yksikäsitteisen yhteyden muodostaminen vaikeaa

Yhteydenmuodostus: perusmalli



Hyvin yksinkertaista!



Yhteyden muodostus ruuhkaisessa verkossa

Jokainen paketti lähetetään kahteen kertaan

Kun yhteys on purettu, viivästyneet kaksoiskappaleet saapuvat

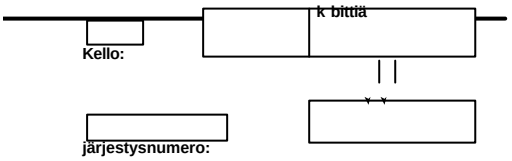
Ne tulkitaan uudeksi yhteydeksi, ja data otetaan vastaan kahteen kertaan!

Ongelman ratkaisuehdotuksia:

- kertakäyttöiset kuljetusosoitteet
  - » nimipalvelu?
- yhteystunnus jokaiselle yhteydelle
  - – yhteyden purkamisen jälkeen sen TPDU:t epäkelpoja
    - » lista epäkelvoista yhteystunnuksista
  - – kuinka kauan historiatietoja säilytettävä?
  - – entä jos kone kaatuu ja unohtaa tietonsa?
- **rajallinen elinikä paketeille**
  - » elinaikalaskuri, hyppylaskuri

Tomlinsonin menetelmä

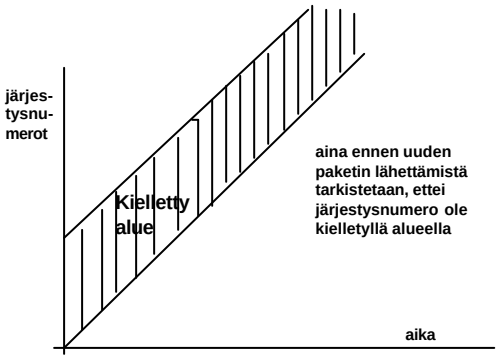
- koneessa vikasietoinen kello
  - » etenevä laskuri
  - » vaikka kone kaatuu, laskuri toimii
- yhteyttä muodostettaessa
  - » kellon bitit ilmoittavat numeroinnin aloituskohdan
  - » bittejä riittävästi (32 bittiä)
    - jotta uudelleen käytöön vasta riittävän pitkän ajan päästä
    - vanhoilla numeroilla varustetut segmentit ehtivät hävitä
- yhteydellä ei koskaan kahta saman numeroista segmenttiä



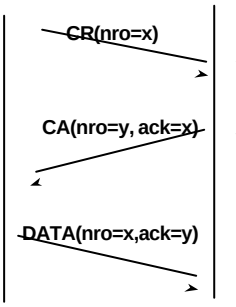
Eri yhteyksillä numerointi aloitetaan eri kohdasta

◆ Kun kone kaatuu
» se kadottaa tiedon viimeksi käytetystä järjestysnumerosta
» uusi numero ei saa olla sama kuin jonkun vielä elossa olevan TPDU:n

◆ Miten selvitään tilanteesta?
– odotetaan T aikayksikköä
» kaikki aikaisemmat TPDU:t varmasti kadonneet ja voidaan aloittaa, mistä numerosta tahansa
» entä, jos T on pitkä
– 'kielletyn alueen' (forbidden region) käyttö
» ei oteta käyttöön numeroita, joiden duplikaatit voivat olla vielä elossa



Kolminkertainen kättely



yhteyspyynnössä pyytäjän nro x

vahvistuksessa sekä pyytäjän että suostujan numero

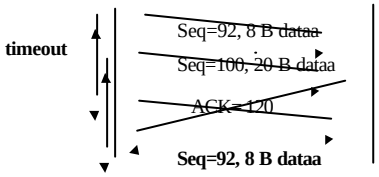
ensimmäisessä datalähetyksessä molemmat numerot

Kuittaukset TCP:ssä

- TCP käyttää kumulatiivista kuittausta
  - kuittaus varmistaa lähettäjälle, että kaikki segmentit kuitattuun segmenttiin saakka ovat saapuneet kunnolla perille
  - väärässä järjestyksessä saapuneita segmenttejä ei kuitata
  - ei käytetä NAK-kuittausta
  - "duplicate ACK" = virhetilanteissa lähetetään uudelleen kuittaus samasta jo kuitatusta segmentistä

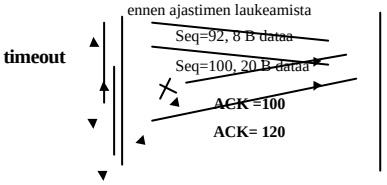
Uudelleenlähettäminen

- TCP lähettää segmentin uudestaan, kun ajastin laukeaa
  - TCP ei automaattisesti lähetä kaikkia puuttuvan segmentin jälkeisiä segmenttejä uudelleen



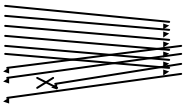
Vain osin "Go-Back -N -tyyppinen"

- Oletetaan, että segmentit 1-N tulevat oikein perille ja kuittaus esim. segmenttiin 1 katoaa. Jos muut kuittaukset tulevat perille, enintään yksi segmentti 1 uudelleenlähetetään.
  - Eikä sitäkään tarvitse lähettää, jos seuraava kuittaus tulee ennen ajastimen laukeamista



Kuittaukset voivat kadota

- Erilliset kuittaukset eli pelkät ACK:it eivät sisällä yhtään tavua dataa, joten niitä ei numeroida eikä kuitata.
  - Kuittauksia voi helposti hävitä



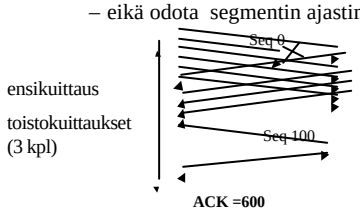
Kumulatiivisissa kuittauksissa seuraava kuittaus paikkaa hävinneen informaation

"Duplicate Ack"

- ensikuittaus (first-time ACK)
  - segmentin ensimmäinen kuittaus
  - tähän saakka kaikki on kunnossa
- toistokuittaus (duplicate ACK)
  - vastaanottaja kuittaa viimeksi saatua hyväksyttyä segmenttiä aina kun saa virheellisen tai väärässä järjestyksessä tulevan segmentin
  - NAKin korvike, jolla ilmoitetaan ongelmista lähettäjälle

Nopea uudelleenlähetys (Fast retransmit)

- Kun lähettäjä vastaanottaa 3 toistokuittausta samalle segmentille, se lähettää heti puuttuvan segmentin uudestaan
  - eikä odota segmentin ajastimen laukeamista

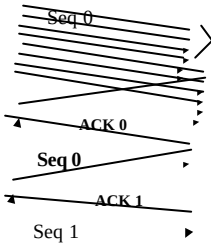


Viivästetty ACK (Delayed ACK)

- Ei tarvitse välttämättä kuitata jokaista segmenttiä
  - kuitenkin kuitattava ainakin joka toinen ja viive saa olla korkeintaan 500 ms,
    - usein noin 200ms
- Hyöty: kuittaus kulkee datan mukana
  - samalla kertaa ikkunan muotos, kuittaus ja kaiutus
- Haitta: kiertoviiveen laskeminen, pakettien kellotus

Suorituskykyongelmia!

- Kun useita segmenttejä katoaa 'samasta ikkunasta'



Segmentit hävivät

Joka kierroksella pystytään uudelleenlähettämään vain yksi kadonnut segmentti!!

Entä jos suuri kiertoviive

SACK (Selective Acknowledgement)

- RFC 2018
  TCP Selective Acknowledgement Options.
  M. Mathis, J. Mahdavi, S.Floyd, A. Romanow. October 1996.
  (Status: **PROPOSED STANDARD)**

INTERNET DRAFT
Mark Allman, Ethan Blanton. "A Conservative SACK-based Loss Recovery Algorithm for TCP".
(draft-allman-tcp-sack-02.txt), January, 2001

- Kumulatiivinen kuittaus paljastaa aina vain yhden puuttuvan kerrallaan
- SACK paljastaa kaikki puuttuvat
  - » ilmoittamalla, mitkä segmenttivälit on jo vastaanotettu
- Esim. Segmentin koko 1000 tavua
  - 1. segmentti katoaa ja muut tulevat perille
    - segmentin 2 kuittaus: ACK 0 , 1000: 2000
    - segmentin 10 kuittaus:ACK 0, 1000: 10000
  - 1. ja 3. segmentti katoavat
    - segmentin 10 kuittaus:
    - ACK 0, 1000:2000 3000:10000

SACK-optiot

- SACK- permitted yhteyden muodostuksessa eli vain SYN-segmentissä ilmoittamaan, että yhteydellä voidaan käyttää SACK-kuittauksia
  - (type = 4, length = 2)
- SACK-optio
  - kuljettaa lisäinformaatiota saapuneista segmenteistä eli kertoo, mitkä 'tavupätkät' ovat jo valmiina vastaanottajan puskurissa
  - kuljetetaan TCP-segmentin optio-osassa

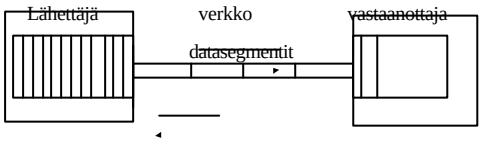
TCP:n SACK-optio

4 lohkoa mahtuu yhteen TCP-segmenttiin, jossa optiolle on varattu 40 tavua, jos ei käytetä muita optioita kuten aikaleimaa (timestamp).

Vain neuvoa-antava!

- Ohjeellista tietoa lähettäjälle
  - vastaanottaja voi tarvittaessa poistaa SACK-optiossa ilmoittamiaan tavuja puskureistaan
- Jos vastaanottaja käyttää SACK-optiota, niin sitä on käytettävä aina kun vastaanottajalla on puskureissaan epäjärjestyksessä olevaa dataa
  - tällöin kaikissa ACK:ssa on oltava ajantasalla oleva tieto siitä, mitkä tavut on jo puskureissa

TCP ruuhkanvalvonta



ACK

toimii lähetyksen tahdistajana

putkestä poistunut dataa, joten voidaan lähettää sama määrä lisää

TCP self-clocking

- TCP tahdistaa itse oman lähetyksensä ACK:ien avulla
  - nopeutta voi rajoittaa
    - » verkko
      - ruuhkan takia syytä vielä pienentää lähetysnopeutta
    - » vastaanottaja
      - lähetysnopeus ok
  - lähettäjä ei voi tietää kumpi

Vuonvalvonta on hankalaa!

- Sitä varten on koko ajan kehitetty yhä parempia menetelmiä
  - uudelleenlähetysajastimen arvo
    - » RTT:n varianssin arviointi
    - » Karnin algoritmi
    - » exponential retransmission timer backoff
  - lähetysikkunan hallinta
    - » slow start
    - » dynamic window size on cogestion
    - » fast retransmit
    - » fast recovery

Limited Transmit

3042 Enhancing TCP's Loss Recovery Using Limited Transmit. M. Allman,
H. Balakrishnan, S. Floyd. January 2001. (Format: TXT=19885 bytes)
(Status: PROPOSED STANDARD)

Uudelleenlähetysajastin

2988 Computing TCP's Retransmission Timer. V. Paxson, M. Allman. November 2000. (Format: TXT=15280 bytes) (Status: PROPOSED STANDARD)

ECN (Explicit Congestion Notification)

INTERNET DRAFT:
K. K. Ramakrishnan, Sally Floyd, D. Black.
"The Addition of Explicit Congestion Notification (ECN) to IP". (draft-ietf-tsvwg-ecn-01.txt), January, 2001.

S. Floyd. "TCP and Explicit Congestion Notification." ACM Computer Communications Review, 24, October 1994

RED

NewReno

2581 TCP Congestion Control. M. Allman, V. Paxson, W. Stevens. Apr 1999. (Format: TXT=31351 bytes) (Obsoletes RFC2001) (Status: PROPOSED
STANDARD)

2582 The NewReno Modification to TCP's Fast Recovery Algorithm. S. Floyd, T. Henderson. April 1999. (Format: TXT=29393 bytes) (Status EXPERIMENTAL)

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 monilähetyspuissa

- Protokolla kaistanleveyden varaamiseen
  - ei varausten totettamiseen 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
  - 'signallointiprotokolla'
    - isäntäkoneet voivat varata siirtokapsiteettia tietovuolle

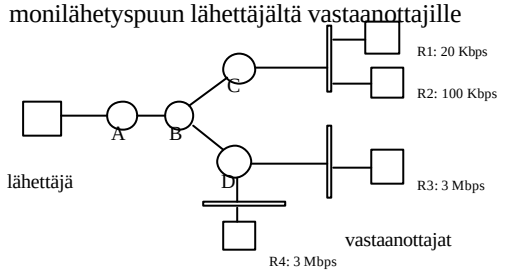
Heterogeenisyys

- 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

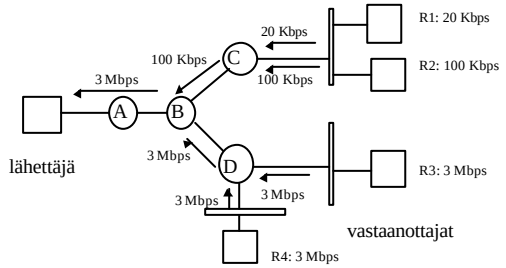
- 'sessio'
  - useita monilähetysdatavoita
  - useita lähettäjiä
  - joka vuolla sama monilähetysosoite
  - reitittimet tunnistavat paketeista, mihin sessioon ja mihin vuohon ne kuuluvat
    - esim. Monilähetysosoite => sessio
    - IPv6:n vuonimiö => vuo
  - lähettäjä lähettää usealle vastaanottajalle videokuvaa kilpailusta
    - joka paketissa monilähetysosoite => vastaanottajat

- Monilähetysprotokolla on muodostanut monilähetyspuun lähettäjältä vastaanottajille



- Jokainen vastaaanottaja lähettää varaussanoman
  - käyttäen 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

Varaussanomat



Tehdyt varaukset