

# 6. Kanalikiht

Side IRT3930

Ivo Mürsepp

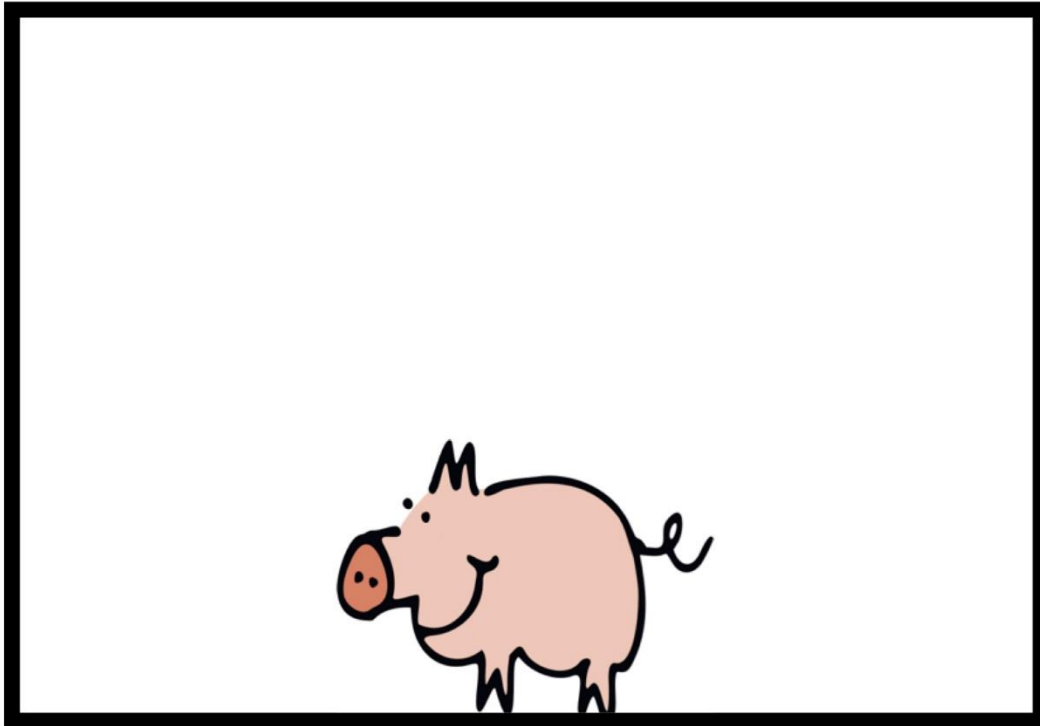
# Kanalikiht

- Füüsiline kiht tegeleb ülekandemeediumiga ja signaalidega.
- Kanalikiht tegeleb juba andmete edastamisega.
- Kanalikihi ülesanded:
  - Voo juhtimine (*flow control*).
  - Veakontroll. Kas vigade tuvastamine või vigade parandamine.
  - Adresseerimine. (MAC aadress)
  - Meediapöördus.

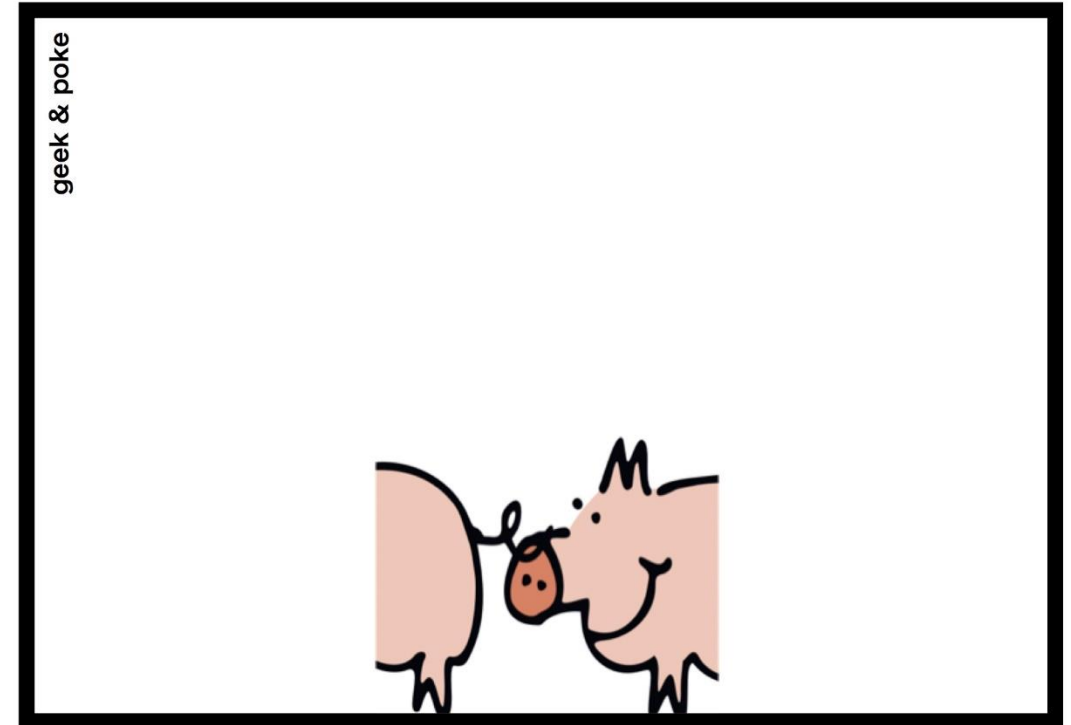
# Meediapöörduskiht *MAC*

- Saatmisel andmete jaotamine kaadriteks, füüsilisel aadressi (MAC aadress), kaadri pikkuse ja veatuvastusväljade lisamine.
- Vastuvõtul andmete eraldamine kaadrist, aadressi ja vigade tuvastamine.
- Kaadri sünkroniseerimine.
- Füüsilisele meediumile juurdepääsu haldamine (*CSMA/CD*).

# SIMPLY EXPLAINED



BIG-ENDIAN

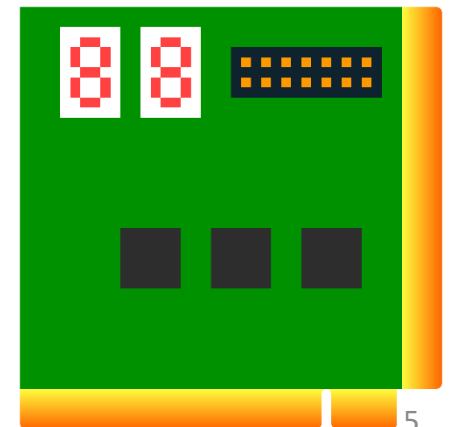


LITTLE-ENDIAN

# MAC -aadress

- 48 bitine (kuus oktetti)
- Esitlusviisid:
  - 01-23-45-67-89-AB
  - 01:23:45:67:89:AB
  - 0123.4567.89AB
- Esimese okteti esimene bitt:
  - 0 – globaalselt unikaalne aadress
  - 1 – lokaalselt muudetud

- Multisaade (*multicast*):
  - Esimese okteti nullis bitt:
    - 0 - unicast
    - 1 - multicast
- Levisaade (*broadcast*):
  - FF-FF-FF-FF-FF-FF



# Veatuuvastus

- Kontrollsumma
  - Lihtne aga ebatõhus
  - Paarsuskontroll
  - Mooduliga liitmine
- CRC (kontrollkood)
  - *Cyclic redundancy check*
  - Põhineb jagamisel
  - Käsitleb andmeid polünoomina
- CRC spetsifikatsioon:
  - Määratud genereeriva polünoomiga

CRC-4:  $x^4 + x + 1$ ; 0x3 (G.704)

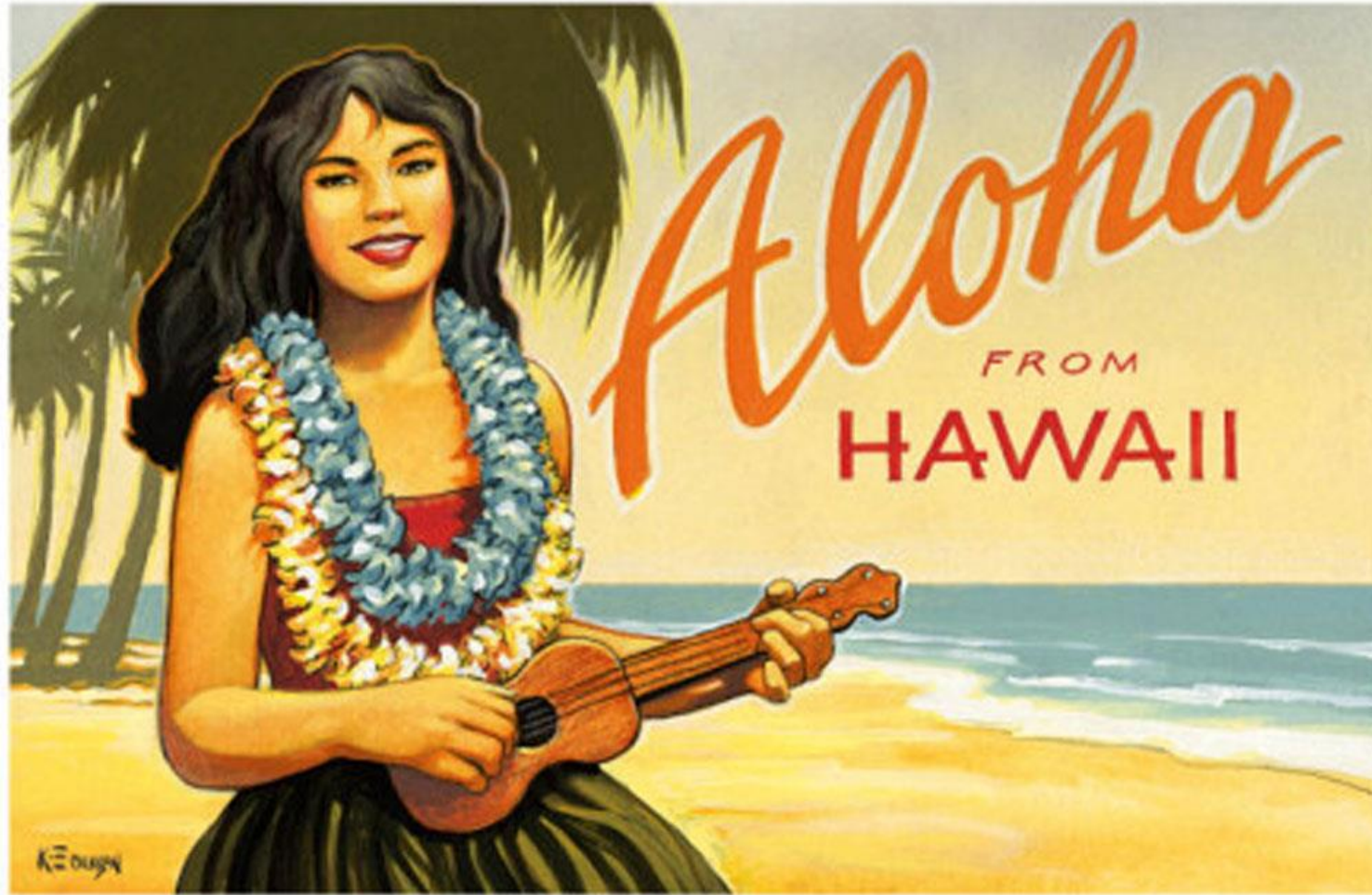
CRC-16-CCITT: 0x1021

  - Algväärtusega
  - Tulemi edastamise järjekorraga
  - Tulemile liidetava konstandiga

# Biti- ja baiditäitmine(*Bit- and byte stuffing*)

- Kaadri algust ja lõppu tähistatakse spetsiifilise väljaga (*flag*): 0x7E
- Juhul kui kaadri sees edastatavates andmetes leidub sama bitijärjestus, siis loeb vastuvõtja selle ekslikult kaadri lõpuks.
- Lahenduseks on nn *bit stuffing*.
  - Iga viie järjestikuse „1“ järele lisatakse „0“ (farssbitt).
- Juhul, kui andmeid edastatakse baidi kaupa on mõistlikum kasutada *byte stuffing*'u nimelist tehnikat (*Control Octet Transparency*).
  - HDLC protokollis kasutatakse spetsiaalselt sümbolit 0x7D (*Control escape octet*), mis asetatakse iga kaadri sees oleva 0x7E või 0x7D oktetit ette. Lisaks inverteeritakse vastava oktetit viies bitt.

# Meediumi jagamine





# Puhas ALOHA

- Edasta kaader millal tahad.
- Peale kaadri edastust oota kinnitust *ACK* aja  $t_{max}$  jooksul

$$t_{max} > \frac{2 \cdot l_{max}}{v}$$

- Kui kinnitust *ACK* ei saabu, siis edasta kaader uuesti.
- Kui  $N$  katse järel kinnitust ei saabu, siis loobu kaadri edastusest.
- Maksimaalne kanali kasutus **18%**
- Pesastatud ALOHA korral **37%**

# CSMA

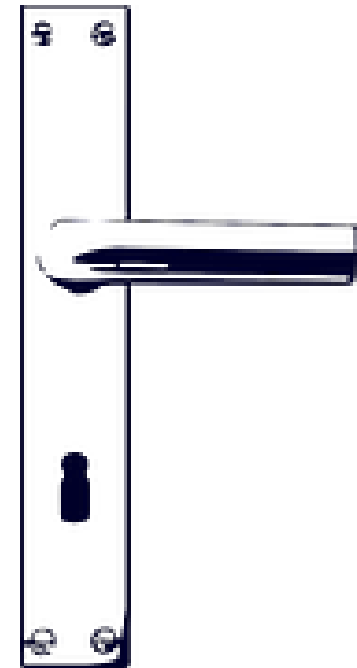
- Kuula, kas keegi teine edastab (meedium vaba ?).
- Kui meedium on vaba, siis edasta kaader.
- Kui meedium ei ole vaba, siis oota juhuslikult valitud aja jooksul ja seejärel alusta uuesti esimesest punktist.
- Alternatiiv (järjekindel CSMA): Kui meedium ei ole vaba, siis kuula edasi. Alusta kaadri edastamist niipea kui meedium vabaneb.

# CSMA/CD

- Kuula, kas keegi teine edastab (meedium vaba ?).
- Kui meedium on vaba, siis edasta kaader.
- Kui meedium ei ole vaba, siis kuula edasi. Alusta kaadri edastamist niipea kui meedium vabaneb. Kuulamist jätkatakse ka edastamise ajal.
- Kui tuvastatakse kokkupõrge (*collision*) siis edasta lühikene teavitussignaali (*jamming*) ja lõpeta seejärel edastus.
- Oota juhuslikult valitud aja (*backoff*) jooksul ja seejärel alusta uuesti esimesest punktist.

# Loogilise ühenduse kiht *LLC*

- Liides kõrgema kihi protokollide jaoks. Andmete multipleksimine (*LSAP*).
- Voo juhtimine (*Stop-and-Wait, Sliding-Window*).
- Vigade tuvastus ja parandamine (*ARQ, FEC*).



# Harjutusülesanded

- Arvuta neljabitine CRC sõnumist 101100111000, kui genereeriv polünoom on 0xA.
- Kaadri sisu on järgmine: 0x7F 0x7E 0x18 0x7D. Kirjuta see välja bittidena enne ja pärast *bit stuffing*'u teostamist.
- Kirjuta eelmises ülesandes antud kaadri sisu välja peale HDLC protokollireeglite järgi *Byte stuffing*'u teostamist.

# Lisaks lugeda

- *William Stallings. Data and Computer Communications* 8<sup>th</sup> edition. Peatükk 7 Data Link Control Protocols. lk 207 – 228.
- **Functions of LLC and MAC sub-layers of Data Link Layer.**  
<http://computernetworkingsimplified.com/data-link-layer/components-data-link-layer-llc-mac/> , 8.10.2017
- Erkki Laaneoks. **Sissejuhatus võrgutehnoloogiasse.** 6 ptk. **OSI kanalikiht.**