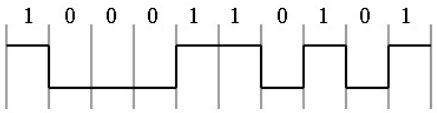
Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Tietoverkkolaboratorio
S-38.108 Tietoliikenneverkkojen arkkitehtuurit
4. laskuharjoitus 7.11.2001

DEMOTEHTÄVIEN RATKAISUT

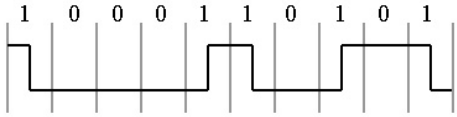
1. Mitä on bittijono 1000110101

   a) NRZ-koodattuna



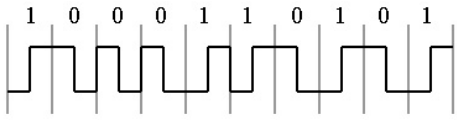
   NRZ:n ongelmana ovat pitkät nolla-ja ykkössarjat, joiden aikana tahdistus helposti katoaa. Lisäksi vastaanottajan on vaikea erottaa pitkää nollasarjaa ja kuollutta linkkiä toisistaan. Pitkät ykkössarjat taas nostavat koko signaalin keskiarvoa, jonka perusteella vastaanottaja tekee eron ykkös- ja nollasignaalien välille.

   b) NRZI-koodattuna



   NRZI ratkaisee NRZ:n ykkössarjaongelman, koska jännite vaihtuu jokaisen ykkösen kohdalla. Sen sijaan nollasarjaongelma jää.

   c) Manchester-koodattuna

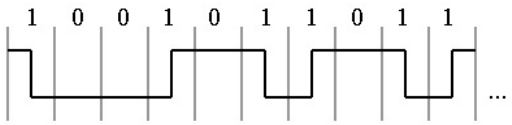


   Manchester-koodauksessa jännite vaihtuu jokaisen bitin kohdalla, joten pitkä sarjoja ei pääse syntymään. Signaalin muutosnopeus on kuitenkin kaksinkertainen NRZ- ja NRZI-koodauksiin verrattuna. Käytännössä tämä tarkoittaa, että samassa ajassa, kun NRZ(I)-koodattuja bittejä siirretään x kappaletta, saadaan Manchester-koodattuja bittejä liikkeelle vain 0,5x.

   d) 4B/5B-koodattuna

   4B/5B-koodauksessa neljän bitin joukko koodataan viideksi bitiksi, jotka lähetetään matkaan NRZI-koodaatuna. Neljää bittiä vastaavat viisi bittiä saadaan selville taulukosta, jossa jokaiselle neljän bitin yhdistelmälle on oma viisibittinen koodinsa. Tarkoituksena on poistaa NRZI:n nollasarjaongelma, joten viisibittiset koodit on valittu siten, ettei niissä tai niiden yhdistelmissä ole enempää kuin kolme nollaa peräkkäin. Joka viides johdossa kulkeva bitti on siis tavallaan turha, mutta koska NRZI käyttää jokaisen signaalinmuutoksen bitin esittämiseen, käyttöaste on peräti 80% (0,8x).

Tehtävän bittijonosta 1000110101 voidaan 4B/5B-koodata kaksi neljän bitin sarjaa: 1000 → 10010 ja 1101 → 11011. Kaksi viimeistä bittiä jää osaksi seuraavaa sarjaa. NRZI-koodattuna tulos näyttäisi tältä:



4B/5B-taulukko

Koodattavat bitit	Viisibittinen koodi
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

2. Kehykset lähetetään vasemmalta alkaen. Generaattoripolynomi on 11100.

   a) Vastaanotettu kehys on 10011100010. Onko se virheetön?

   Jaetaan kehys generaattoripolynomilla:

```
          1100101
11100|10011100010
      11100
       11111
       11100
        00110
        00000
         01100
         00000
          11000
          11100
           01001
           00000
            10010
            11100
             1110
```

   Jakojäännös ≠ 0, joten kehys on virheellinen.

b) Lähetettävä viesti on 1101001. Mikä on sen CRC-tarkistussumma, ja minkälainen kehys lopulta lähtee vastaanottajalle?

Viesti V = 1101001
Generaattoripolynomi G = 11100 (n+1 bittiä pitkä)
Tarkistussumma T = ? (n bittiä pitkä)

Generaattoripolynomi on neljännen asteen polynomi ($x^4+x^3+x^2$), joten viestin loppuun lisätään neljä nollaa ennen jakolaskua.

```
          1010111
    ┌──────────────
11100 11010010000
      11100
       01100
       00000
        11001
        11100
         01010
         00000
          10100
          11100
           10000
           11100
            11000
            11100
             0100
```

Viestin CRC-tarkistussumma on 0100. Lähetettävä kehys saadaan laskemalla yhteen jakolaskussa käytetty viesti ja tarkistussumma: 11010010000 + 0100 = 11010010100.

c) Voiko bitit laittaa matkaan kummasta päästä tahansa alkaen?

Kehyksen bittijonot eivät ole palindromeja, joten lähettäjän ja vastaanottajan on sovittava, kummasta päästä bittien lähetys aloitetaan. Jos eniten merkitsevä bitti (MSB) lähtee liikkeelle ensimmäisenä, voi vastaanottopää alkaa tarkistaa CRC:tä jo ennen kuin koko kehys on perillä.

3. Tietoturvakäsitteitä

a) symmetrinen ja asymmetrinen salaus
Symmetrisessä salauksessa samaa avainta käytetään sekä viestin salaamiseen että purkamiseen. Purkuavain saatetaan myös laskea salausavaimesta. Tunnettu symmetristä salausta käyttävä protokolla on SSL (Secure Sockets Layer).
Asymmetrisessä salauksessa käytetään kahta avainta, toista viestin salaamiseen ja toista purkamiseen. Avainparin avaimia ei pysty laskemaan toistensa perusteella, joten salaamiseen käytetyn ns. julkisen avaimen voi antaa kenelle tahansa. Asymmetristä salausta kutsutaan myöt julkisen avaimen salausmenetelmäksi, ja yksi tunnetuimpia sen sovelluksia on PGP.

b) todennus (authentication)
Todennus vastaa kysymykseen: ”Kuka sinä olet?”. Todennus voidaan tehdä esimerkiksi salasanan tai vaikkapa sormenjälkien avulla.

c) valtuutus (authorization)
Valtuutus vastaa kysymykseen: ”Mitä sinä saat tehdä?”. Valtuutuksessa voidaan käyttää suunnilleen samoja tunnistuskeinoja kuin todennuksessakin, mutta silti valtuutus ei edellytä todennusta.

d) DoS-hyökkäys (Denial of Service)
Hyökkääjä yrittää estää käyttäjiltä pääsyn kohonkin palveluun esimerkiksi kaatamalla palvelinkoneen. Hyökkäys voi kohdistua tiettyyn käyttäjään tai koko järjestelmään.

e) Man-in-the-Middle –hyökkäys
A ja B haluavat kryptata toisilleen lähettämänsä viestit käyttäen julkisen avaimen salausmenetelmää. A lähettää julkisen avaimensa B:lle, mutta verkossa A:n ja B:n välissä onkin ilkeämielinen C, joka kaappaa avaimen ja lähettää B:lle A:n avaimen sijasta oman julkisen avaimensa. B vastaa A:lle lähettämällä oman julkisen avaimensa (kuvitellulla) A:n avaimella kryptattuna, mutta kryptaus ei auta, koska B:n käyttämä avain onkin C:n avain. C kaappaa viestin, ottaa haltuunsa B:n julkisen avaimen ja lähettää A:lle oman julkisen avaimensa oikealla A:n avaimella kryptattuna. A ja B luulevat vaihtaneensa julkisia avaimia keskenään, mutta tosiasiassa heillä molemmilla on C:n julkinen avain, joten C, ”man-in-the-middle”, pystyy lukemaan kaikki heidän viestinsä.