S-38.118 Teletekniikan perusteet
Perusasioita tiedonsiirrosta

Markus Peuhkuri

21. syyskuuta 1999

Luennon aiheet

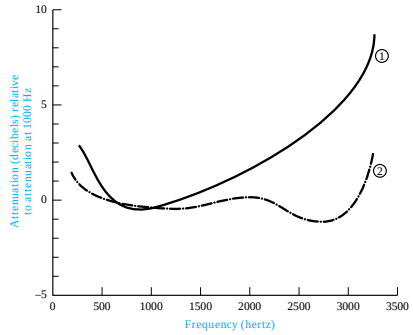
- Miksi ~~vihreitä~~ virheitä syntyy
- Miten virheet havaitaan
- Miten virheet korjataan

Kommunikointi

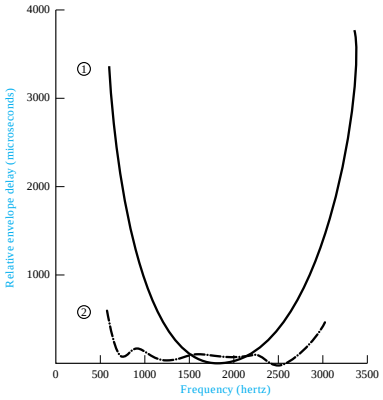
1. Lähettäjä lähettää sähkömagneettisen symbolin
   ⇒ lähettimen epäideaalisuuden vuoksi vääristyy
2. Symboli kulkee siirtotiellä
   ⇒ hukkuu kohinaan, vääristyy
3. Vastaanottaja vastaanottaa symbolin
   ⇒ tunnistaa sen vääräksi symboliksi

Siirtotiellä syntyvät häiriöt

- Vaimentuma ja vaimentumahäviöt
- Viiveen hajonta
- Kohina



(a) Attenuation



(b) Delay distortion

Vaimentuma

- Kasvaa etäisyyden kasvaessa

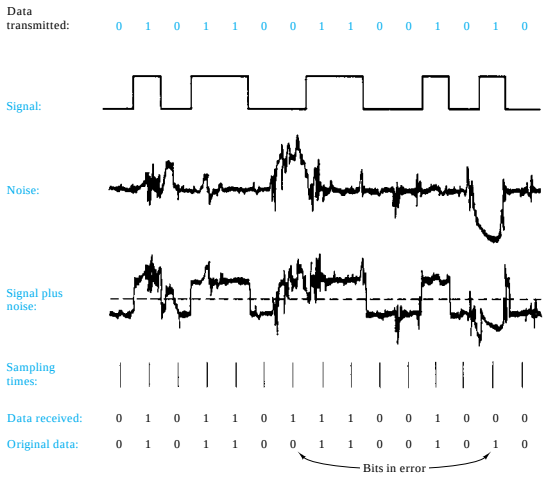
  ohjattu siirto yleensä logaritminen: db/km

  ohjaamaton siirto monimutkaisempi
- Kasvaa taajuuden kasvaessa
- Signaalin kulkunopeus ohjatussa siirrossa riippuu taajuudesta

Kohina

- Lämpokohina ($N = kTW$)
  - valkoinen kohina, taajuusriippumatonta
- Keskeismodulaatio
  - f_1, f_2: häiriöitä $\pm nf_1 \pm mf_2$
- Ylikuuluminen

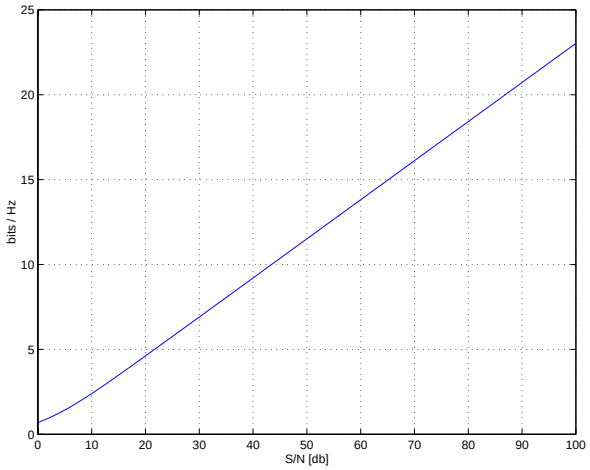
- Impulssikohina
  - salamat, kytkennät: lyhyitä ja voimakkaita
    $\Rightarrow$ tuhoisa digitaliselle, vähäinen analogiselle



Siirtokanavan kapasiteetti

- Siirtonopeus [bps]
- Kaistanleveys [Hz]
- Kohina [W, S/N]
- Virhesuhde [P(virhe)]
  todennäköisyys, että vastaanotettu symboli tulkitaan väärin
- Shannon: kohina rajoittaa kapasiteettia

$$C = B \log \left(1 + \frac{S}{N}\right)$$



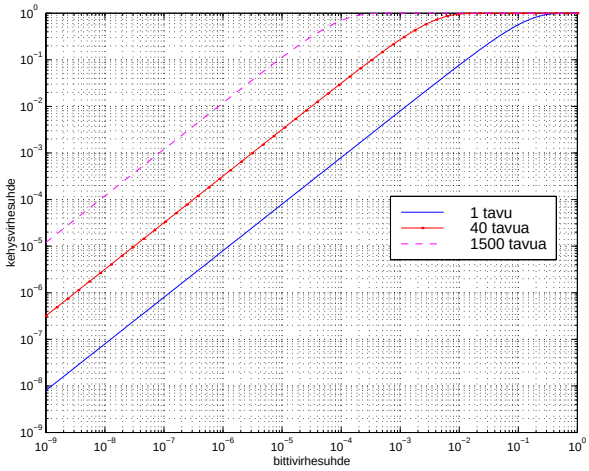
Mitäs me pienistä virheistä

- Virheitä syntyy sitä enemmän mitä lähempänä kapasiteetin maksimia liikutaan
- Virheitä myös päätelaitteissa ja tallennuksessa

- Bittivirheen merkitys riippuu datan tyypistä ja virheen sijainnista

Datan tyyppi	min	max
ääni	ei havaittava	napsaus
kuva (ei pakattu)	ei havaittava	täplä
kuva (pakattu)	ei havaittava	käyttökelvoton
teksti	kirjain	muotoilu
luku (ASCII)	± 1	$\pm 8 * 10^N$
luku (32 bit)	± 1	$\pm 2^{31}$
luku (liukuluku)	$\pm \epsilon$	$\pm 10^{308}$

- Mikä on hyväksyttävä virhemäärä?



Miten havaitaan?

Pariteetti jokaista merkkiä kohti lisätään yksi bitti, joka tekee ykkösten kokonaismäärän parilliseksi (tai parittomaksi)
$\Rightarrow$ havaitsee parittoman (1, 3, ...) määrän bittivirheitä

`0101 0110` $\rightarrow$ `0101 0110 0`

Pitkittäinen pariteetti pariteettimerkin bitti C_i on pariteettibitti jokaisen merkin bittistä i
$\Rightarrow$ havaitsemattomien virheiden määrä pienenee 2–4 suuruusluokkaa

1	0	1	1	0	1	1	1
1	1	0	1	0	1	1	1
0	0	1	1	1	0	1	0
1	1	1	1	0	0	0	0
1	0	0	0	1	0	1	1
0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0

Tarkistussumma lasketaan summaamalla peräkkäiset tavut (tai sanat) yhteen
$\Rightarrow$ kaksi bittivirhettä voi kumota toisensa

- nopea laskea kopionnin yhteydessä
  $\Rightarrow$ käytössä esimerkiksi IP:n otsikon tarkistussummassa

  > The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

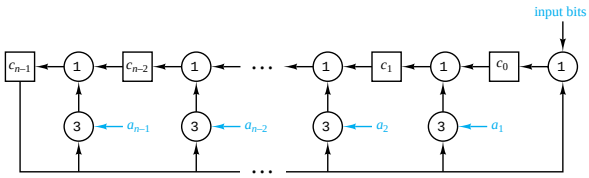
- riittää, mikäli alempana on luotettava tarkiste

Syklinen tarkistussumma (CRC)

- Lähetetään viesti M ja tarkiste F (pituus n bittiä)
- F määritetään sellaiseksi, että $T = 2^n M + F$:n jako P:llä menee tasan (modulo-2 aritmetiikka)
- Mikäli vastaanottajalla ei mene tasan, viestissä on virhe
- Havaitaan
  1. kaikki yhden bitin virheet
  2. kaikki kahden bitin virheet jos polynomissa on vähintään kolme termiä
  3. kaikki parittomat virhemäärät jos termi $X + 1$
  4. kaikki virhepurskeet, joiden pituus $< n$
  5. useimmat pidemmät purkeet
- Esitetään usein polynomina

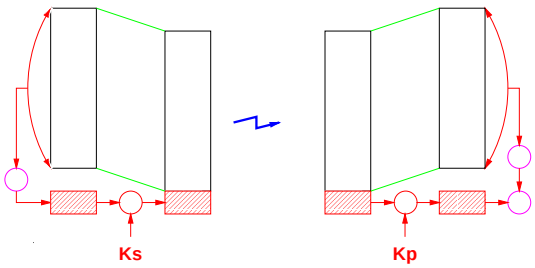
$$\begin{aligned} CRC-12 &= X^{12}+X^{11}+X^3+X+1 \\ CRC-16 &= X^{16}+X^{15}+X^2+1 \\ CRC-CCITT &= X^{16}+X^{12}+X^5+1 \\ CRC-32 &= X^{32}+X^{26}+X^{23}+X^{22} \\ &\quad +X^{16}+X^{12}+X^{11}+X^{10} \\ &\quad +X^8+X^7+X^5+X^4 \\ &\quad +X^2+X+1 \end{aligned}$$

- Voidaan toteuttaa laitteistotasolla siirtorekisterien ja ehdoton-TAI porttien avulla



Kryptografisesti vahvat tarkisteet

- Tarkisteet tarkoitettu suojamaan "*luonnollisilta*" virheiltä
  $\Rightarrow$ eivät suojaa tarkoituksellisilta vääristelyiltä
- Vahvoja tarkisteita tarvitaan sähköisessä asioinnissa
  - allekirjoitukset
  - eheyden tarkistus



$$h = H(M)$$

- h:n laskeminen helppoa, jos M tunnetaan
- M:n laskeminen vaikeaa h:n perusteella
- M':n löytäminen vaikeaa vaikka M tunnetaan, $H(M) = H(M')$

MD5 Message Digest 5 (Ron Rivest) tuottaa 128-bittisen tiivisteen

SHA Secure Hash Algorithm: 160 bittiä

Virhe havaittu: apuva!

Onko mahdollista saada data uudestaan?

on pyydetään data uudestaan

ei pitää varautua ennakolta
$\Rightarrow$ käytetään virheenkorjaavaa koodausta

Cross Interleave Reed-Solomon Code (CIRC)
data hajautetaan ja lasketaan useampi pariteetti (4/12 databittiä)

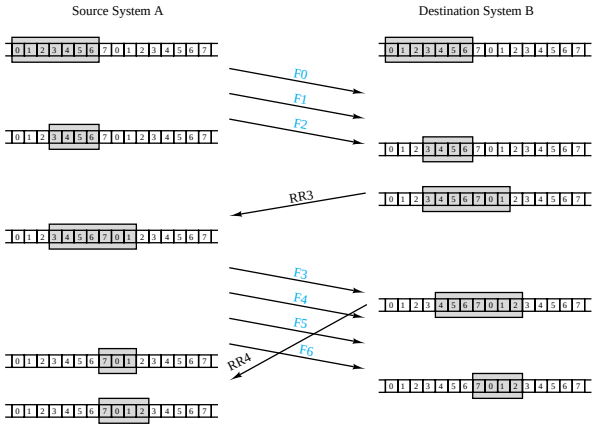
Uudelleenlähetykset

Positiiviset kuittaukset

- viestit numeroitu
- vastaanottaja kuittaa kun vastaanottaa virheittä
  - pysähdy ja odota
  - liukuva ikkuna
- aikavalvonta lähettäjällä

Negatiiviset kuittaukset

- vastaanottaja kuittaa kun vastaanottaa roskaa



Yhteenveto

- Siirrossa syntyy virheitä
- Virheet voidaan havaita
  (tietyllä todennäköisyydellä)
- Virheen korjaus onnistuu