

1. *Apskaičiuokite maksimalų kanalo pralaidumą (bitais per sekundę), kai QAM-4 moduliacijos būdu koduotas signalas siunčiamas kanalu kurio juostos plotis yra 40 KHz ir signalo triukšmo lygis 30 dB.*

Sprendimas:

Naikvisto teorema:

$$V = 2H \log_2 M$$

V – kanalo pralaidumas (b/s)

H – kanalo juostos plotis (Hz)

M – vienu metu perduodamos informacijos kiekis (b)

Šenono teorema:

$$C = H \log_2 \left(1 + \frac{S}{N}\right)$$

C – kanalo pralaidumas (b/s)

H – kanalo juostos plotis (Hz)

S/N – signalo triukšmo santykis (b)

H = 40 000Hz;

M = 4 (čia koks prie moduliacijos šiuo atveju QAM-4, tai 4, jei būtų QAM-16, tai būtų 16);

S/N = 1 000 (šį skaičių gauname iš 30dB atlikę tam tikrą konvertaciją, per egzą sakė, kad duos iškarto S/N tipo skaičių, t.y. nereikės verstis (pvz. 20dB = 100, o 10dB = 10)).

Įsistatom į lygtis, apsiskaičiuojam V bei C ir iš jų išvedam minimumą, Ats. = min(V,C).

2. Koks kanalo efektyvumas esant kodo pralaidumui 1 Mbps ir kurio užlaikymas 20ms, kadro ilgis 1500 bitai, naudojant stop-and-wait algoritmą?

Sprendimas:

Kanalo efektyvumas:

$$\text{efektyvumas} = \frac{l}{l + bR}$$

l – kadro dydis bitais

b – kanalo pralaidumas b/s

R – bendras perdavimo laikas (išsiuntimas + gavimas)

$l = 1\,500b$;

$b = 1\,048\,576\text{ b/s}$;

$R = 20 \cdot 2\text{ms}$ (nes mes laiką sąlygoje turim tik į vieną pusę)

Viską įsistatom ir gaunam efektyvumą.

3. Kaip atrodys kadras 0xaf09 pridėjus (prikonkatenavus) vieno baido kontrolinę sumą, kurios generatorius $G(x) = x^8 + x^6 + x^3 + 1$?

Sprendimas:

Tai pirma susigeneruojam G , kas bus lygu 101001001.

Toliau kadimą (0xaf09) pasiverčia į dvejetainį skaičių, kas bus:

1010 1111 0000 1001

Tuomet reikia vykdyti XOR, kaip tiksliai nežinau...

Atsakymą reiktų iš gauto binario skaičiaus konvertuoti į šešioliktinį.

4. Įsivaizduokite, kad Jums reikia sukurti 250 Mbps tinklą su 8 km kabeliu pagal (802.3 veikimo schemą). Signalo greitis kabelyje yra 200 000 km/s. Koks tokiu atveju yra minimalus leistinas kadro ilgis?

Sprendimas:

$$\tau = l/v$$

l - ilgis (kelias)

v - greitis

t(teta) – laikas

$$\text{galutinė formulė} = 2\tau * v$$

l = 8km;

v = 200 000 km/s;

Viską įsistatom ir gaunam atsakymą.

5. Duomenų srautą 0x7ef029ee7fa7027e reikia išskaidyti po keturis baitus, prie kiekvienos dalies pridedant: a. pradžios bei pabaigos markerius – 0x7e, escape simbolis yra 0x02; b. pradžios ir pabaigos bitų seką 0x7e.
- a) Tai pasiimam duomenų srautą 7e f0 29 ee 7f a7 02 7e ir bandom pridėti markerius, būtų **7e7e...**, taip yra negerai, nes yra markeris ir po jo eina kitas kaip atrodytų markeris, tačiau tai jau duomuo, tad reikia dėti escape simbolį, taigi būtų taip 7e **02** 7e f0 29 ee 7f a7 ir ateinam prie 02, kas yra duomens dalis, tačiau jį palaikys kaip escape simbolį, todėl reikia pridėti escape simbolį, gaunasi 7e 02 7e f0 29 ee 7f a7 **02** 02, tuomet sutinkam 7e, kas irgi yra duomens dalis, tai kad ja tokia ir laikytų reikia padėti 02, taiga 7e 02 7e f0 29 ee 7f a7 02 02 **02** 7e ir pačioje pabaigoje uždedam pabaigos markerį 7e. galutinis duomenų srautas su pabaigos ir pradžios markeriais atrodo taip: 7e 02 7e f0 29 ee 7f a7 02 02 02 7e **7e**.
- b) Čia kaip dėstytojas rodė reikia tiesiog viską į dvejetainį skaičių pasiversti: 0111 1110 1111 0000 0010 1001 1110 1110 0111 1111 1010 0111 0000 0010 0111 1110. Tada paprasta taisyklė, iš eilės 5 vienetai negali eiti, tai jei taip yra, po 4 vienetų dedam 0. Gaunam: 0111 1011 0111 1000 0001 0100 1111 0111 0011 1101 1110 1001 1100 0000 1001 1110 **110** (nežinau ką daryti su paskutiniu skaičiumi ar tai yra 0110 ar 1100, t.y. kur nulį reikia prisirašyti). Ir konvertuojam atgal į šešiolyktainį. Bus: 7b 78 14 f7 3d e9 c0 9e c arba **6**.

6. Bitų seką 0xfa91 užrašykite Mančesterio, diferencialinio Mančesterio, Non-Return-To-Zero, Non-Return-To-Zero Inverse, Alternative mark inversion kodais.

Pirmas žingsnis pasiverčiam į dvejetainį fa91 (1111 1010 1001 0001), tuomet visa info yra 4 temos skaidrėje nuo 7 slaido.

7. Tinkle, kuriame yra dešimt maršrutizatorių: young, ying, yang, proton, neutron, electron, sun, moon, star ir earth prijungiamas naujas maršrutizatorius: defender. Tinkle naudojamas distance vector maršrutizavimo protokolas. Maršrutizatorius defender yra prijungiamas prie maršrutizatorių ying, proton, sun, moon iš kurių gaunami tokie atstumų vektoriai:

ying

Router Next hop Dist.

young	young	1
ying	ying	0
yang	yang	1
proton	-	inf
neutron	-	inf
electron	-	inf
sun	-	inf
moon	-	inf
earth	-	inf
star	-	inf

sun

Router Next hop Dist.

young	-	inf
ying	-	inf
yang	-	inf
proton	-	inf
neutron	-	inf
electron	-	inf
sun	sun	0
moon	moon	1
earth	-	inf
star	moon	2

proton

Router Next hop Dist.

young	-	inf
ying	-	inf
yang	-	inf
proton	proton	0
neutron	neutron	1
electron	electron	1
sun	-	inf
moon	-	inf
earth	-	inf
star	-	inf

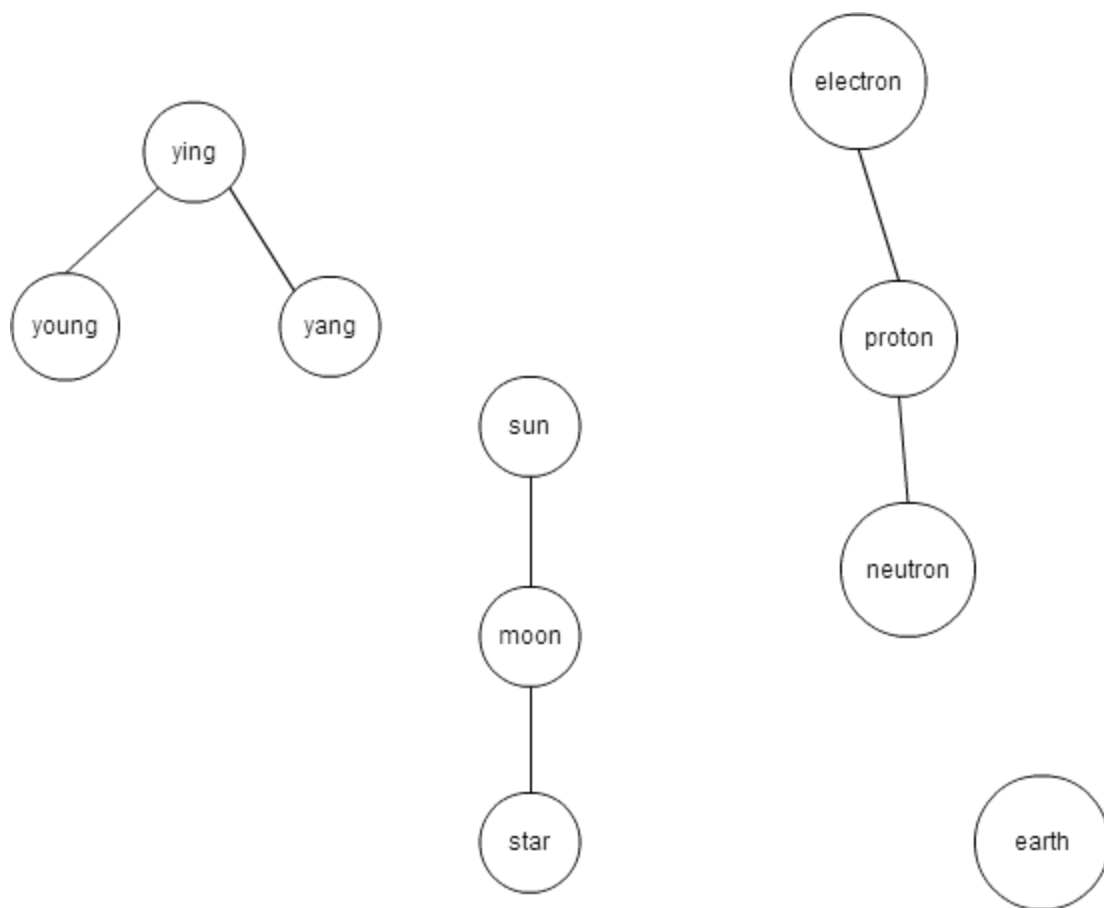
moon

Router Next hop Dist.

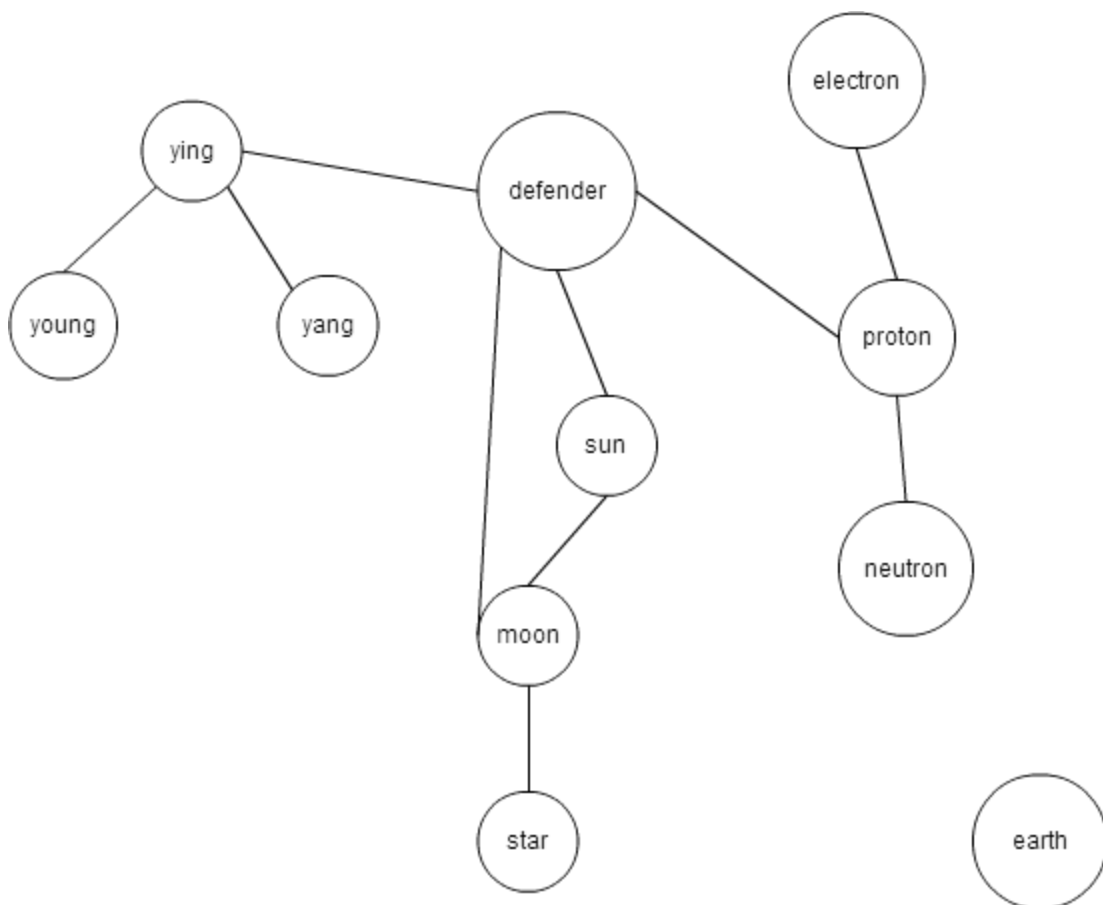
young	-	inf
ying	-	inf
yang	-	inf
proton	-	inf
neutron	-	inf
electron	-	inf
sun	sun	1
moon	moon	0
earth	-	inf
star	moon	1

Kaip atrodys maršrutizatoriaus neutron vektorių lentelė, kai tinklas pilnai konverguos? Laikykite atstumą tarp maršrutizatoriaus defender ir jo kaimynų lygų 1, o maršrutizatoriaus paketo apdorojimo laiką – 0.

Tai pirmą nusipaišom kaip vizualiai tai atrodys:



Čia kaip atrodo tinklas su 10 maršrutizatorių (be defender). Pvz. paišom kaip susijęs Ying su kitais maršrutizatoriais. Žiūrim į Ying lentelę. Router grafa rodo su kuriais routeriais jungiasi, Next hop per kurį routerį einama, o Dist. atstumas. Tai imam young, next hop - young ir dist - 1, tai reiškia, kad tiesiogiai susijungę ir tiesiog nupiešiam liniją. Ying, next hop – ying, dist – 0, tai tiesiog nieko nepaišom, nes pats į save rodo. Yang, next hop – Yang, dist – 1, tai pat kaip su young. Ir taip pat padarom su visais kitais. Dabar pridedam defender maršrutizatorių ir jį prijungiam prie tų maršrutizatorių, kurie paminėti sąlygoje:



Dabar parašom kaip atrodys neutron maršrutizatoriaus vektorių lentelę.

Router	Next Hop	Dist
Yang	Proton	4
Young	Proton	4
Ying	Proton	3
Sun	Proton	3
Moon	Proton	3
Star	Proton	4
Earth	-	Inf
Defender	Proton	2
Electron	Proton	2
Proton	Proton	1

Tai pildymas vyksta taip: parašom į kurį routerį keliaujam, tada per kurį routerį einam, šiuo atveju bus visi išėjimai per Proton routerį, tai Next Hop visur bus Proton, ir tuomet ieškom trumpiausio kelio iki norimo routerio, į Dist įrašom per kiek briaunų pereinam iki tikslo.

8. *Išrašykite visus leistinus IP adresus, žinant kad tinklo adresas yra 20.4.1.0 ir tinklo kaukė (netmask) 255.255.255.252. Koks šio potinklio broadcast adresas?*

Vat šito dunno.. visi adresai paskaičiuoti pagal kažką yra 20.4.1.0 20.4.1.1 20.4.1.2 20.4.1.3. Broadcast'as turbūt bus 20.4.1.3 arba 20.4.1.0 ☺ tai plačiau čiuju reikia googlint ☺

9. *Turimas DNS serverio konfigūracijos fragmentas, aprašantis domeną snieg.as:*

www IN A 81.0.5.10
snieg.as. IN A 81.0.5.10
kur IN A 81.0.5.11
baltas IN A 81.0.5.12
ns1 IN A 81.0.5.10
ns2 IN A 81.0.5.11

Jmonė turi du pašto serverius kur.snieg.as ir baltas.snieg.as. Papildykite DNS serverio konfigūraciją pridėdant šiuos du serverius taip, kad jie abu aptarnautų domeno snieg.as naudotojus. Serveris baltas.snieg.as būtų pirminis pašto serveris, o jam atsijungus, laiškai būtų apdorojami serverio kur.snieg.as.

Čia yra kalbama apie MX įrašus, apie juos plačiau įtariu googlei. O atsakymas:

snieg.as. MX 5 kur

snieg.as. MX 10 baltas

ką čia padarėm, tai tiesiog skyrėm prioritetus, didesnis skaičius aukštesnis prioritetas, tai kol baltas nenulūš, jis bus pirminiu pašto serveriu, kai nulūš, tai kur virs pirminiu serveriu.

10. *Iš HTTP serverio norime parsisiųsti failą esantį /public/videoz/titanic.avi . Kokia užklausa turime nusiųsti serveriui, kad šį failą gautume?*

Tiesiog parašom eilutę:

GET public/videoz/titanic.avi HTTP/1.0