**ΕΠΙΧΕΙΡΗΜΑΤΙΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ ΔΕΔΟΜΕΝΩΝ-ΣΗΜΕΙΩΣΕΙΣ**

Ένας κατάλογος δεν γίνεται να διαγραφεί αν έχει περιεχόμενο. Πρέπει πρώτα να αδειάσουμε το περιεχόμενο και μετά να τον διαγράψουμε

Το tab συμπληρώνει την εντολή

**Linux:** traceroute διεύθυνση

**Windows:** tracert διεύθυνση

**gw** = gateway(**πύλη**)

**tfr** = transmission frame rate (**ρυθμός μετάδοσης πλαισίου**)

**Timer** (στο **sending node**) = εξυπηρετεί αν χαθεί το πακέτο ή το **acknowledgement**

**Throughput:** (κλάσμα) πόσα πακέτα προσπάθησα να περάσω προς πόσα πακέτα τελικά πέρασαν

**Back off time:** υποχώρηση ενός πακέτου ώστε να μην συμπέσει με άλλο πακέτο

**Propagation time:** χρόνος μετάδοσης του πακέτου

**CSMA/CA -> WIFI**

**CSMA/CD -> ETHERNET**

Το **G** είναι από 0 μέχρι 1

**CSMA :** ακούω πριν μιλήσω (δες διαφάνειες)

Να μελετήσω τα πρωτόκολλα μαζί με τις ασκήσεις

Μεγαλύτερος ρυθμός μεγαλύτερες συνέπειες

Οι ασκήσεις είναι sos για τις εξετάσεις (ειδικά το crc)

**Κ** και **n bits** είναι μια σειρά ακολουθίας από 0 και 1

**Απόσταση Hamming :** κάνω **xor** ανάμεσα στις δύο λέξεις και **modulo 2**  και η απόσταση είναι ίση με το πλήθος των 1(άσσων)

**ΑΣΚΗΣΗ ΕΓΓΥΗΜΕΝΟΥ ΕΝΤΟΠΙΣΜΟΥ ΛΑΘΩΝ**

Τα σχήματα κωδικοποίησης ορίζονται από 3 παραμέτρους n,k,dmin

C(n=codeword, k=dataword),dmin

C(3,2)dmin=2 άρα s=1

101

1 θέση bit: 100 οδηγεί σε μη έγκυρο codeword και απορρίπτεται στην πλευρά του παραλήπτη

2 θέσεις bit: 110 εσφαλμένα γίνεται αποδεκτό ως έγκυρο

C(5,2)dmin=3 άρα s=2

01011

Αλλαγή σε 1 ή 2 θέσεις: 11011 ή 11111 οδηγεί σε μη έγκυρα codewords τα οποία και απορρίπτονται

Αλλαγή σε 3 θέσεις: 00000 εσφαλμένα γίνεται αποδεκτό ως έγκυρο

Ο κώδικας είναι γραμμικός όταν κάνω **xor** ανάμεσα σε δύο **codeword** και το αποτέλεσμα θα είναι ένα άλλο έγκυρο **codeword**

d=0100 c=? : d+r r2r1r0

r0=a2+a1+a0 (mod2)

r1=a3+a2+a1 (mod2)

r2=a1+a0+a3 (mod2)

r0=1+0+0=1(mod2)

r1=0+1+0=1(mod2)

r2=0+0+0=0(mod2)

d+r = 0100011 ->checked ->s2s1s0

s0=b2+b1+b0+q0

s1=b3+b2+b1+q1

s2=b1+b0+b3+q2

s0=1+0+0+1=0(mod2)

s1=0+1+0+1=0(mod2)

s2=0+0+0+0=0(mod2)

s=000(Το s δείχνει σε ποια θέση βρίσκεται το σφάλμα στο codeword (d+r) που παραλείπεται)

Να λύσω μόνος μου και τις άλλες δύο περιπτώσεις

d=11 g=2

11mod2 =1 -> r

c= d+r= 11+ 1 =12

12mod2=0 -> Δεν έχω σφάλμα

13mod2=0 -> Έχω σφάλμα

14mod2=0

111 ---> x2 +x+1

**Διαίρεση**

Dataword: 1001 ---> x3+1: d(x)

Divisor: 1011 ---> x3+x+1: g(x)

Αποστολέας:

1001000

1011

------ Πράξη xor (κάθε φορά κατεβάζω έναν αριθμό από το dataword)

0100

0000

--------Πράξη xor

1000

1011

---------

0110

0000

-----------

110 --->remainder (υπόλοιπο) : x2+x

Codeword: 1001110 ---> x6+x3+x2+x : c(x)

**Παραλήπτης**

**Περίπτωση:** δεν έχω σφάλμα

1001110

1011

------- Πράξη xor

0101

0000

------- Πράξη xor

1011

1011

--------- Πράξη xor

0000

0000

-------- Πράξη xor

000

**Περίπτωση:** έχω σφάλμα

1000110

1011

-------- Πράξη xor

0111

0000

--------- Πράξη xor

1111

1011

-------- Πράξη xor

1000

1011

--------- Πράξη xor

011 ≠ 000 discard

**SOS:** Στις εξετάσεις αν πέσει τέτοιο θέμα τότε καλό είναι να αλλάξω ένα οποιοδήποτε άλλο bit σε σχέση με αυτό του παραπάνω παραδείγματος.

Mod g(x) => g(x)=0 => x3+x+1=0 => x3=-x-1 => x3=x+1

x6+x3  x6=x3\*x3=(x+1)\*(x+1)=x2+2x+1= x+1 **\*το 2x είναι 0 γιατί οι συντελεστές x ανήκουν στο πεδίο Ζ2:{0,1}**

**ΕΡΓΑΣΤΗΡΙΟ**

255.255.255.0 => 3 φορές βλέπω το 255 άρα 3\*8=24 το οποίο θα μπει μετά την κάθετο

Το traceroute δείχνει τα άλματα(**hops**) που χρειάστηκαν για να φτάσω κάπου

**Φόρμα google επανάληψης**

1. 3
2. Το τρίτο
3. Σωστό
4. 1->DNS, 2->αποστολέας, 3->παραλήπτης
5. 1->ls, 2->cd dir, 3->cd, 4->pwd, 5->mkdir dir, 6-> cat>file.txt, 7->rm file, 8->chmod +x file, 9->uname -a
6. τέταρτη και πέμπτη
7. πέμπτη
8. πρώτη και τρίτη
9. δεύτερη
10. Σωστό
11. 1->δεύτερη, 2->τελευταία, 3->πέμπτη, 4->πρώτη
12. 1->τέταρτη, 2->δεύτερη, 3->πρώτη, 4->τρίτη
13. 1 -> 5, 2 -> 3, 3 -> 2, 4 -> 4, 5 -> 1
14. Δεύτερη
15. πρώτη, τέταρτη, πέμπτη, έκτη, έβδομη
16. πρώτη
17. 1 -> 4, 2 -> 4, 3 -> 3, 4 -> 2, 5 -> 1
18. τρίτη
19. τρίτη
20. τρίτη
21. πρώτη**(επειδή είναι 16 κρατάω τα πρώτα 16 bit)**
22. τρίτη
23. πρώτη
24. τρίτη
25. 1 -> 3, 2 -> 2, 3 -> 1
26. πρώτη
27. τρίτη
28. πρώτη, τρίτη, όγδοη
29. τέταρτη
30. τρίτη, τέταρτη