Αριστοτέλειο Πανεπιστήμιο
Θεσσαλονίκης

Τμήμα
Πληροφορικής

Ατομική εργασία στο μάθημα *“Ψηφιακές Επικοινωνίες”*

Όνομα: Άννα
Επώνυμο: Γκίκα-Ζοσάν
ΑΕΜ: 3349
Ακαδημαϊκό έτος: 2019-2020

Περιεχόμενα

0 Εισαγωγή

Το ακόλουθο φύλλο αναφοράς περιγράφει την λειτουργία που προγράμματος που σχεδιάστηκε για την ανίχνευση σφαλμάτων με τον αλγόριθμο CRC. Ακολουθούν τα σημαντικότερα κομμάτια του κώδικα, με εκτενή σχόλια, σε γλώσσα προγραμματισμού **python**, παραδείγματα που αποδεικνύουν την ορθότητα του προγράμματος καθώς και μερικά ποσοστά σχετικά με την εσφαλμένη αποστολή στον αποδέκτη,την εσφαλμένη ανίχνευση άλλα και την “ορθή” / “επιθυμητή” ανίχνευση.

Σε αυτό το σημείο θα ήταν καλό να αναφερθεί ότι η συγκεκριμένη αναφορά γίνεται στα πλαίσια του μαθήματος “Ψηφιακές επικοινωνίες” του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης στο δεύτερο έτος προπτυχιακών σπουδών του τμήματος πληροφορικής.

1.1 Το πρόγραμμα

Το συγκεκριμένο “κεφάλαιο” του φύλλου αναφοράς εξηγεί τον τρόπο με τον οποίο λειτουργεί το πρόγραμμα και αναλύονται μερικά επιπλέον σημεία τα οποία χρειάζονται περαιτέρω εξήγηση από το σχόλια που υπάρχουν μέσα στον κώδικα.

1.1 Παραδείγματα Λειτουργίας

Για την απόδειξη την ορθής λειτουργίας του προγράμματος θα παρουσιαστούν 4 παραδείγματα 1.000.000 επαναλήψεων για τυχαίες τιμές(bit). Για τα παρακάτω ισχύει:

K=10
P=110101
BER=10^-3

```
Mistakes in BER:  10028
Not Accepted:  9981

Percentage of mistakes at the receiver :  1.0028
Percentage of signals detected as errors from CRC :  0.9981
Percentage of signals that are errors but are not being detected:  0.0047
```

Παράδειγμα 1.1.1

```
Mistakes in BER:  9959
Not Accepted:  9925

Percentage of mistakes at the receiver :  0.9959
Percentage of signals detected as errors from CRC :  0.9925
Percentage of signals that are errors but are not being detected:  0.0034
```

Παράδειγμα 1.1.2

```
Mistakes in BER:  9920
Not Accepted:  9882

Percentage of mistakes at the receiver :  0.992
Percentage of signals detected as errors from CRC :  0.9882
Percentage of signals that are errors but are not being detected:  0.0038
```

Παράδειγμα 1.1.3

```
Mistakes in BER:  10065
Not Accepted:  10005

Percentage of mistakes at the receiver :  1.0065
Percentage of signals detected as errors from CRC :  1.0005
Percentage of signals that are errors but are not being detected:  0.006
```

Παράδειγμα 1.1.4

1.2 Περιγραφή των σημαντικότερων σημείων.

Το πρόγραμμα μπορεί να χωριστεί σε 3 μικρότερα προβλήματα: το FCS,την “δημιουργία” λάθος, την ανίχνευση σφάλματος. Τα τρία αυτά τμήματα περιγράφονται παρακάτω.

Το συγκεκριμένο κομμάτι κώδικα χωρίζεται σε δύο μικρότερα κομμάτια. Το πρώτο δημιουργεί ένα τυχαίο μήνυμα. Το δεύτερο κάνει τη διαίρεση μεταξύ του μηνύματος και του p.

```
data = ''
for i in range(0, 10):
    data = data + str(random.randint(0, 1))
data_ = [int(x) for x in str(data)]
data = data + '00000'
data2 = [int(x) for x in str(data)]   #χρησιμοποιούνται strings για
k = 0                                 #καλύτερη διαχείριση

_____________________________________
while n - k + 1 > len(p):
    if data2[k] != 0:
        for i in range(0, 6):
            data2[k + i] = data2[k + i] ^ p[i]
    k = k + 1
```

Σε αυτό το σημείο γίνεται η δημιουργία του σφάλματος. Έχουμε έναν τυχαίο αριθμό από το 0 μέχρι 1. Αν ο αριθμός είναι μικρότερος του BER τοτέ αλλάζουμε το ψηφίο bit.

```
for k in range(0, len(data_)):
    if random.random() < 0.001:
        go = False
        if data_[k] == 1:
            data_[k] = 0
        else:
            data_[k] = 1
if not go:
    mistakes_ber = mistakes_ber + 1
```

Αν γίνει αλλαγή (το go γίνεται false) τότε αυξάνουμε τον αριθμό των μηνυμάτων που στέλνονται λανθασμένα στο αποδέκτη.

Στο τελευταίο,αλλά εξίσου σημαντικό κομμάτι του κώδικα ελέγχουμε για το πόσα σφάλματα ανιχνεύει ο CRC

```
k = 0
while n - k + 1 > len(p):
    if data3[k] != 0:
        for o in range(0, 6):
            data3[k + o] = data3[k + o] ^ p[o]
    k = k + 1

found = False
for i in range(0, 15):
    if data3[i] != 0:
        found = True
if found:
    mistakes = mistakes + 1
```

Αν βρεί λάθη(δεν είναι όλα τα ψηφία 0) τότε αυξάνει τον αριθμό λανθασμένων κατά 1.

Σημείωση: Έχει μπει σε comment η δυνατότητα να δώσει ο χρήστης το P

2 Απάντηση των ερωτημάτων

Το τελευταίο “κεφάλαιο” αυτής της εργασίας παρουσιάστηκε,*ελλιπώς,* στα παρουσίαση των παραδειγμάτων. Παρατηρήθηκε ότι το το ποσοστό λάθους είναι παίρνει τιμές κυρίως μεταξύ του 0,9 και 1,5%(για 1.000.000 ψηφία), με το ποσοστό των σφαλμάτων που λαμβάνονται από το CRC να είναι ελάχιστα μικρότερος. Ωστόσο το ποσοστό λάθους που **δεν** περνάει από το CRC είναι ~0,004%.

Παρακάτω παρουσιάζεται ένας μέσος όρος που βρέθηκε ύστερα από 10 επαναλήψεις των 1.000.000 σημάτων (με τρία δεκαδικά ψηφία) και απαντάει λίγο πιο *“συμμαζεμένα”* στα δοθείσα ερωτήματα.

Ποσοστό που φτάνουν με σφάλμα στον αποδέκτη : 1,0006%
Ποσοστό που ανιχνεύονται εσφαλμένα από το CRC : 0,9966%
Ποσοστό που φθάνουν με σφάλμα στον αποδέκτη και δεν ανιχνεύονται από το CRC : 0,004%
