Δίκτυα Υπολογιστών II

Ακολουθεί επεξήγηση και παράδειγμα του κώδικα της άσκησης 3.5.

Το πρόγραμμα ακολουθεί την παρακάτω γενική ροή:

1. Διαβάζει τα δεδομένα από τον χρήστη.
2. Υπολογίζει ένα αρμοστό `timeout`.
3. Υπολογίζει τον ελάχιστο απαραίτητο χρόνο για να φτάσει ένα πακέτο `D_suc`.
4. Υπολογίζει τον μέσο επιπλέον χρόνο που απαιτείται λόγω λαθών `D_err`.
5. Εμφανίζει τις παραπάνω πληροφορίες.

1. Εισαγωγή δεδομένων

Το πρόγραμμα ζητάει τις πληροφορίες τους συνδέσμου και κάνει έλεγχο ορθότητας, τα ορθά πεδία είναι τα ακόλουθα:

$$0 < P$$
$$0 < C$$
$$0 \leq PROP$$
$$0 \leq BEP \leq 1$$
$$0 \leq max\_err \leq 2$$

2. Υπολογισμός `timeout`

Το `timeout` πρέπει να είναι ελαχίστως μεγαλύτερο από τον χρόνο που απαιτείται για να σταλεί το πακέτο, και ο αποστολέας να λάβει απάντηση, δηλαδή:

$$\frac{P}{C} + prop + \frac{ACK}{C} + prop$$

Επιλέχθηκε στην τύχη το πακέτο ACK από τον δέκτη να έχει μέγεθος 10% σε σχέση με του αποστολέα.

3. Υπολογισμός `D_suc`

Ο υπολογισμός γίνεται με τον από τύπο της καθυστέρησης:

$$\frac{P}{C} + prop$$

4. Υπολογισμός `D_err`

Ο επιπλέον χρόνος που απαιτείται για την επαναποστολή του πακέτου ισούται με τον χρόνο ανα resend (`timeout`) επί τον μέσο όρο των επαναποστολών μείων 1 (την επιτυχής).

Η τιμή του μέσου όρου μπορεί να υπολογιστεί με την χρήση της μέσης τιμής της γεωμετρικής κατανομής, όπου p η πιθανότητα να φτάσει σωστό πακέτο. Ο ορισμός του "σωστού πακέτου" αλλάζει βάσει τα μέγιστα επιτρεπτά λάθη.

Η πιθανότητα να φτάσει σωστό ένα πακέτο ισούται με την πιθανότητα να κάνει 0 έως `max_err` λάθη σε ένα πακέτο. Δηλαδή αν X ακολουθεί διωνυμική κατανομή, οπού n = P, p = BEP. Δηλαδή:

$$\begin{aligned} P\_suc &= \sum_{i=0}^{max\_err} P(X = i) \\ &= \sum_{i=0}^{max\_err} \binom{P}{i} BEP^i (1 - BEP)^{P-i} \end{aligned}$$

Σημειώνεται λόγω τεχνικών περιορισμών, ο υπολογισμός των συνδυασμών γίνεται με κάποιες προ εφαρμοσμένες απλοποιήσεις, συγκεκριμένα:

$$\begin{aligned} \binom{n}{k} &= \frac{n!}{k!(n-k)!} \\ &= \frac{n(n-1)\cdots 1}{k!(n-k)(n-k-1)\cdots 1} \\ &= \frac{n(n-1)\cdots(n-k+1)}{k!} \end{aligned}$$

5. Εμφάνιση αποτελεσμάτων

Το πρόγραμμα εμφανίζει τους προηγούμενους υπολογισμούς καθώς και το άθροισμα των καθυστερήσεων.

Παράδειγμα

Έστω δίνουμε στο πρόγραμμα τις ακόλουθες εισόδους:

```
P          = 32
C          = 8
PROP       = -1
BEP        = 0.05
max_err    = 1
```

Το πρόγραμμα βρίσκει το λάθος στην είσοδο του PROP, δίνουμε ως νέα τιμή το 1.

Βάσει τα στοιχεία πρέπει το πρόγραμμα να εμφανίσει τα ακόλουθα αποτελέσματα:

$$timeout = \frac{32}{8} + 1 + \frac{0.1 * 32}{8} + 1$$
$$= 4 + 1 + \frac{3}{8} + 1$$
$$= 6.375$$

$$D\_err = 6.375 * (\frac{1}{0.5199} - 1)$$
$$= 6.375 * 0.9234$$
$$= 5.886$$

$$X \sim B(32, 0.05)$$
$$P\_suc = P(X = 0) + P(X = 1)$$
$$= 0.1937 + 0.3262$$
$$= 0.5199$$

$$D\_suc = \frac{32}{8} + 1 = 5$$

$$D\_total = 5 + 5.886$$
$$= 10.886$$

Όντως το πρόγραμμα υπολόγισε σωστά τις καθυστερήσεις και timeout.

```
Delay_total:    10.886
Delay_err:      5.886
Delay_suc:      5.000
Timeout:        6.375
Press any key to continue . . .
```

Κουλουράς Ιωάννης
E20075