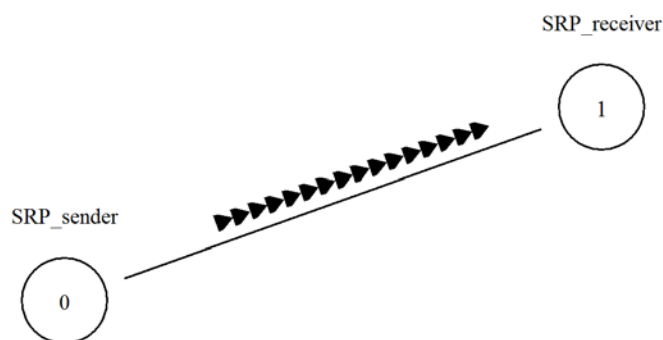


Πρωτόκολλο Selective Repeat

Σε αυτή την εργαστηριακή άσκηση θα μελετήσουμε την επίδοση του πρωτοκόλλου Selective Repeat (πρωτόκολλο επιλεκτικής επανάληψης με αναμετάδοση). Το πρωτόκολλο αυτό χρησιμοποιείται τόσο στο στρώμα ζεύξης δεδομένων όσο και στο στρώμα μεταφοράς, και παρέχει εγγυημένη παράδοση δεδομένων πάνω σε αναξιόπιστο δίκτυο. Απορρίπτεται κάθε κατεστραμμένο πλαίσιο που λαμβάνεται, αλλά τα υπόλοιπα σωστά πλαίσια αποθηκεύονται προσωρινά. Έτσι δεν απορρίπτονται πλαίσια απλώς και μόνο επειδή ένα προηγούμενο πλαίσιο καταστράφηκε ή χάθηκε.

Τόσο ο αποστολέας όσο και ο παραλήπτης διατηρούν ένα παράθυρο με αποδεκτούς αριθμούς ακολουθίας. Το μέγεθος του παραθύρου αποστολής αρχίζει στο 0 και φτάνει μέχρι κάποιο προκαθορισμένο μέγιστο MAX_SEQ. Αντίθετα, το παράθυρο λήψης έχει σταθερό μέγεθος και είναι ίσο με MAX_SEQ. Ο παραλήπτης δεσμεύει ένα χώρο προσωρινής αποθήκευσης για κάθε αριθμό ακολουθίας μέσα στο σταθερό του παράθυρο. Με κάθε περιοχή προσωρινής αποθήκευσης σχετίζεται ένα bit που δείχνει αν η περιοχή είναι γεμάτη ή άδεια. Όταν φτάνει ένα πλαίσιο, ελέγχεται ο αριθμός ακολουθίας του για να δούμε αν βρίσκεται εντός του παραθύρου. Αν συμβαίνει αυτό και αν το πλαίσιο δεν έχει ήδη παραληφθεί, γίνεται αποδεκτό και αποθηκεύεται. Η ενέργεια αυτή γίνεται χωρίς να εξετάζουμε αν περιέχει το επόμενο πακέτο που αναμένεται από το επίπεδο δικτύου ή όχι. Φυσικά, το πλαίσιο θα πρέπει να παραμείνει στο επίπεδο συνδέσμου μετάδοσης δεδομένων και να μη μεταβιβαστεί στο επίπεδο δικτύου μέχρι να παραδοθούν με τη σωστή σειρά στο επίπεδο δικτύου όλα τα πλαίσια με χαμηλότερους αριθμούς ακολουθίας.

Δημιουργούμε μια απλή τοπολογία, με δύο κόμβους $n(0)$ και $n(1)$ που συνδέονται μεταξύ τους με μια πλήρως αμφίδρομη ζεύξη, εύρους ζώνης 10 Mbps με 30ms καθυστέρηση. Δημιουργήσαμε μια TCP σύνδεση ανάμεσα στους δύο κόμβους, αφού το πρωτόκολλο Selective Repeat εμπεριέχεται στο πρωτόκολλο μεταφοράς TCP, και ορίσαμε μέγεθος παραθύρου ίσο με 16. Θεωρήσαμε επίσης ως εφαρμογή που θα δημιουργήσει την κίνηση, την μεταφορά ενός αρχείου απείρου μεγέθους με FTP, έτσι ώστε να υπάρχει πάντα πληροφορία για μετάδοση μεταξύ των χρόνων 0.5 sec και 4.5 sec.



Μελέτη της απόδοσης Selective Repeat

Ερωτήσεις

- α. Πώς πρέπει να τροποποιηθεί ο κώδικας της προσομοίωσης ώστε η ζεύξη μεταξύ των δύο κόμβων της διάταξης να απεικονίζεται σε οριζόντια θέση, όπως φαίνεται στο Σχήμα 1;

Για να έχει σταθερό προσανατολισμό η ζεύξη, προσθέτουμε μετά από τον ορισμό της, την ακόλουθη εντολή:
\$ns duplex-link-op \$n(0) \$n(1) orient right. Έτσι έχουμε:



- β. Να επαληθεύσετε κατά πόσον ισχύει ή όχι η εξίσωση σε περίπτωση απουσίας σφαλμάτων:

$$\eta = \min\left\{\frac{W \cdot TRANSP}{S}, 1\right\}$$

Αντικαθιστώντας με τα κατάλληλα μεγέθη, δηλαδή

$W = 16$ (το μέγεθος του παραθύρου),

$$TRANSP = \frac{\text{μήκος πακέτου}}{\text{ρυθμός μετάδοσης}} = \frac{1500 \cdot 8 \text{ Bits}}{10 \text{ Mbits/s}} = 0.0012 \text{ sec} \text{ (ο χρόνος μετάδοσης ενός πακέτου δεδομένων)}$$

$PROP = 30 \text{ ms}$ (η καθυστέρηση της διάδοσης του πακέτου, που ταυτίζεται με την καθυστέρηση της ζεύξης)

$$TRANSA = \frac{\text{μήκος επιβεβαίωσης}}{\text{ρυθμός μετάδοσης}} = \frac{40 \cdot 8 \text{ bits}}{10 \text{ Mbits/s}} = 0.000032 \text{ sec} \text{ (ο χρόνος μετάδοσης μιας επιβεβαίωσης)}$$

$$S = TRANSP + 2 \cdot PROP + TRANSA = 1.2 + 2 \cdot 30 + 0.032 \text{ ms} = 61.232 \text{ ms}$$

$$\text{Ενώ } W \cdot TRANSP = 16 \cdot 1.2 = 19.2 \text{ ms} \text{ άρα } \eta = \frac{19.2}{61.232} = \mathbf{0.313561}.$$

$$\text{Η πειραματική τιμή υπολογίζεται από τον τύπο } \eta = \frac{\text{ρυθμός μετάδοσης δεδομένων}}{\text{ρυθμός μετάδοσης ζεύξης}} = \frac{3200.8 \text{ Kbits/s}}{10 \text{ Mbits/s}} = \mathbf{0.32008}$$

Η τιμή του ρυθμού μετάδοσης δεδομένων υπολογίστηκε στο ερώτημα δ. Παρατηρούμε ότι η θεωρητική και η πειραματική τιμή παρουσιάζουν πολύ μικρή απόκλιση.

- γ. Ποιος είναι ο αριθμός των πακέτων που παρελήφθησαν; Πόσα δεδομένα παρελήφθησαν από τον παραλήπτη κατά τη διάρκεια της προσομοίωσης;

Εκτελώντας του προγράμματος ανάλυσης awk, παίρνουμε τα ακόλουθα αποτελέσματα:

Total Data received : 1624660 Bytes

Total Packets received : 1055

- δ. Τροποποιήστε κατάλληλα το πρόγραμμα awk, ώστε να προσδιορίζει τη συνολική διάρκεια μετάδοσης των δεδομένων (η οποία περιλαμβάνει και την ολοκλήρωση μετάδοσης όλων των επιβεβαιώσεων). Υπολογίστε το ρυθμό μετάδοσης δεδομένων και τη χρησιμοποίηση του καναλιού.

Προσθέτοντας τις εντολές

```
/^r/ack/ { last_time = $2; }
```

πριν το τμήμα END, και εμφανίζοντας μέσα στην END το αποτέλεσμα με την εντολή

```
printf("Last package delivered at\t: %s sec\n", last_time);
```

παίρνουμε Last package sent at : 4.56064 sec, που σημαίνει ότι ο συνολικός χρόνος μετάδοσης είναι $t_{total} = 4.56064 - 0.5 = 4.06064 \text{ sec}$

Αφού στάλθηκαν συνολικά 1624660 Bytes μέσα σε 4.06064 sec, ο ρυθμός μετάδοσης θα είναι $\frac{1624660}{4.06064} = 400099,491705741 \frac{\text{Bytes}}{\text{sec}} = 3200.8 \text{ Kbits/s}$

Και η χρησιμοποίηση του καναλιού, με εύρος ζώνης ίσο με 10 Mbits/s είναι ίσο με την πειραματική τιμή της απόδοσης του πρωτοκόλλου που υπολογίστηκε στο ερώτημα β, δηλαδή **0.32008** ή **32.008%**.

- ε. Με βάση την εξίσωση που παρατίθεται νωρίτερα, υπολογίστε τη θεωρητική τιμή της χρησιμοποίησης καναλιού, θεωρώντας ότι το μέγεθος των πακέτων αυξάνεται κατά 40 byte λόγω επικεφαλίδων TCP και IP, και ότι οι επαληθεύσεις (ACK) έχουν μέγεθος 40 byte. Ισχύει η εξίσωση; Αν όχι, πού οφείλεται η απόκλιση;

Θεωρούμε πλέον μέγεθος πακέτου $1500+40=1540$ Bytes. Τα ακόλουθα μεγέθη αλλάζουν σε:

$TRANSP = \frac{1540 \cdot 8 \text{ bits}}{10 \text{ Mbits/s}} = 0.001232 \text{ s} = 1.232 \text{ ms}$ και $S = 1.232 + 2 \cdot 30 + 0.032 = 61.264$ και

$\eta = \frac{16 \cdot 1.232}{61.264} = \mathbf{0.321755}$ ή **32.1755%**.

Σύμφωνα με την παραπάνω εξίσωση, λαμβάνουμε μεγαλύτερη θεωρητική τιμή χρησιμοποίησης του καναλιού σε σχέση με την πειραματική. Η απόκλιση αυτή πιθανότατα οφείλεται στο ότι μεσολαβεί κάποια καθυστέρηση στον κόμβο που λαμβάνει τα δεδομένα, μεταξύ της λήψης του πακέτου και της αποστολής της επιβεβαίωσης (ACK) ώστε να προλάβει να γίνει επεξεργασία του πακέτου (για το αν λήφθηκε κατεστραμμένο ή όχι – στην προσομοίωσή μας δεν έχουμε κατεστραμμένα πακέτα, όμως και πάλι θα γίνει η επεξεργασία).

- στ. Διατηρώντας σταθερό το μέγεθος του παραθύρου, αλλάξτε το μήκος των πακέτων, ώστε η θεωρητική απόδοση του πρωτοκόλλου να λάβει τη μέγιστη τιμή της. Για ποιο μήκος πακέτων συμβαίνει αυτό; Υπολογίστε πειραματικά την απόδοση του πρωτοκόλλου (χρησιμοποίηση του καναλιού) για το μήκος πακέτου που προσδιορίσατε εδώ. Υπάρχει απόκλιση μεταξύ πειραματικής και θεωρητικής τιμής;

Θεωρητικά, υπολογίζουμε από τον τύπο του ερωτήματος (α) το μήκος του πακέτου, για να έχουμε μοναδιαία απόδοση, κρατώντας τα υπόλοιπα μεγέθη σταθερά, και υπολογίζουμε μέγεθος πακέτου ίσο με 5005.33 bytes . Πειραματικά, ορίζοντας μέγεθος παραθύρου ίσο με 5006 bytes, παίρνουμε:

Total Data received : 5005592 Bytes

Total Packets received : 992

Last package delivered at : 4.564506 sec

Υπολογίζουμε με βάση τα παραπάνω ρυθμό μετάδοσης $\frac{5005592 \cdot 8}{4.564506 - 0.5} = 9.8523 \text{ Mbits/s}$, που μας δίνει πειραματική τιμή για την απόδοση 98.523 % και όχι 100% όπως υπολογίσαμε θεωρητικά. Εδώ η απόκλιση είναι μεγαλύτερη σε σχέση με το προηγούμενο ερώτημα, και οφείλεται στο γεγονός ότι για μεγαλύτερο μήκος πακέτου, ο κόμβος n(1) θα παίρνει περισσότερη ώρα για να αποφασίσει αν το πακέτο έφτασε κατεστραμμένο ή όχι.

- ζ. Διατηρώντας σταθερό το μήκος πακέτου που υπολογίσατε στο ερώτημα (στ), αυξήστε στο δεκαπλάσιο το ρυθμό μετάδοσης της ζεύξης και ρυθμίστε το μέγεθος του παραθύρου, ώστε και πάλι η απόδοση να λάβει τη μέγιστη τιμή της. Για ποιο μέγεθος παραθύρου συμβαίνει αυτό; Πόσα περισσότερα bits απαιτούνται για την αναπαράσταση των αριθμών ακολουθίας πακέτων του πρωτοκόλλου Selective Repeat στην περίπτωση αυτή;

Ακολουθώντας την ίδια μεθοδολογία όπως και στα προηγούμενα ερωτήματα, υπολογίζουμε για δεκαπλάσιο ρυθμό μετάδοσης ζεύξης ελάχιστο μέγεθος παραθύρου 150,98, και επιλέγουμε 151.

Σύμφωνα με την θεωρία, έχουμε ότι για το μέγεθος παραθύρου ισχύει ο τύπος $W = (\text{Max_Seq} + 1)/2$ όπου Max_Seq είναι ο μέγιστος αριθμός σε μια ακολουθία bits.

Έτσι, για το αρχικό παράθυρο μεγέθους 75, έχουμε $\text{Max_Seq} = 2 \cdot 16 - 1 = 31$, που αναπαρίσταται στο δυαδικό σύστημα ως 11111, δηλαδή απαιτούνται 5 bits για παράθυρο μεγέθους 16.

Για παράθυρο μεγέθους 151, έχουμε $\text{Max_Seq} = 2 \cdot 151 - 1 = 301$ bits, για την αναπαράσταση του οποίου (100101101) απαιτούνται 9 bits. Άρα για αυτή τη περίπτωση χρειαζόμαστε 4 παραπάνω bit για την αναπαράσταση των αριθμών ακολουθίας πακέτων του πρωτοκόλλου Selective Repeat.

- η. Εφαρμόστε τώρα το πρωτόκολλο για την παραμετροποίηση του ερωτήματος (ζ), θεωρώντας όμως ζεύξη με πενταπλάσια καθυστέρηση διάδοσης. Υπολογίστε την απόδοση του πρωτοκόλλου στη νέα αυτή ζεύξη τόσο θεωρητικά, όσο και πειραματικά. Αιτιολογείστε τυχόν αποκλίσεις που παρατηρούνται.

Με μέγεθος παραθύρου 151 που υπολογίσαμε προηγουμένως σε ζεύξη με δεκαπλάσιο ρυθμό μετάδοσης, αλλά και με πενταπλάσια καθυστέρηση διάδοσης αυτή τη φορά, θεωρητικά η απόδοση υπολογίζεται $\eta = 20.130\%$.

Πειραματικά έχουμε τα κάτωθι αποτελέσματα:

Total Data received : 10667204 Bytes

Total Packets received : 2114

Last package delivered at : 4.766245 sec

Επομένως ο ρυθμός μεταφοράς είναι $10667204 \cdot \frac{8}{4.766245 - 0.5} = 20.003 \text{ Mbits/s}$, μέγεθος που δίνει πειραματική απόδοση $\eta = \frac{20.003}{100} = 20.003\%$.

Η μικρή απόκλιση που παρατηρούμε πιθανότατα οφείλεται στο ελάχιστο μεγαλύτερο μέγεθος κάθε πακέτου μέσα στο παράθυρο (που χρειάζεται 4 bits παραπάνω που δεν έχουμε υπολογίσει για την αναπαράσταση των αριθμών ακολουθίας πακέτων) και φυσικά στις στρογγυλοποιήσεις των υπολογισμών μας.

- θ. Εφαρμόστε το πρωτόκολλο Go Back N αντί του Selective Repeat στην τελευταία παραμετροποίηση της προσομοίωσης και μετρήστε την απόδοση του πρωτοκόλλου αυτού πειραματικά. Διαφέρουν οι πειραματικές αποδόσεις των δύο πρωτοκόλλων; Γιατί;

Αλλάζοντας την εντολή `set tcpθ [new Agent/TCP/Sack1]` σε `set tcpθ [new Agent/TCP/Reno]` χρησιμοποιούμε πλέον το πρωτόκολλο Go Back N. Τα αποτελέσματα ωστόσο είναι ακριβώς τα ίδια από άποψη χρόνου παράδοσης πακέτων και ποσό δεδομένων που λήφθηκαν. Αυτό συμβαίνει επειδή η προσομοίωσή μας είναι ιδανική, δηλαδή δεν περιλαμβάνει σφάλματα. Σε περίπτωση ύπαρξης σφαλμάτων, το πρωτόκολλο Selective Repeat θα παρουσίαζε καλύτερη απόδοση, αφού θα έστελνε πάλι μόνο το εσφαλμένα πακέτα, και όχι όλα τα πακέτα που βρίσκονται εντός ενός παραθύρου του οποίου η παράδοση εμφάνισε πρόβλημα.