

# Improving the performance of TCP in the case of packet reordering

Στρατάκη Μαρία

# Γενικές Πληροφορίες για το TCP/IP

- TCP (Transmission Control Protocol)
- IP (Internet Protocol)
- Χωρίζουν τα δεδομένα σε τμήματα (τα πακέτα)
- Παραδίδουν τα πακέτα αυτά στον κατάλληλο προορισμό
- Μετά την παράδοση ξαναενώνουν τα πακέτα στην αρχική τους μορφή ώστε να μπορούν να χρησιμοποιηθούν από τον παραλήπτη
- Το TCP έχει την αρμοδιότητα να διαχωρίζει τα δεδομένα σε πακέτα και να τα επανενώνει
- Το IP είναι υπεύθυνο ώστε τα πακέτα να σταλούν στο σωστό προορισμό

## Packet reordering σε TCP δίκτυα

- Το φαινόμενο της επαναδρομολόγησης πακέτων είναι πολύ συχνό σε δίκτυα υψηλών ταχυτήτων όπως προκύπτει από πολλές μελέτες που έχουν γίνει μέχρι σήμερα.
- Συμβαίνει κυρίως λόγω αλλαγών στη διαδρομή που ακολουθεί το πακέτο και στην περίπτωση που η νέα διαδρομή προσφέρει μικρότερη καθυστέρηση σε σχέση με την παλαιότερη διαδρομή
- Οι επαναλαμβανόμενες καταστάσεις επαναδρομολόγησης πακέτων προκαλούν σημαντική μείωση στην απόδοση του δικτύου

## Μέθοδοι εύρεσης απώλειας πακέτου

### I. Χρονομετρητής Επαναμετάδοσης (Retransmission Timer)

Αν η επιβεβαίωση (acknowledgement) δεν φθάσει στον αποστολέα μέσα σε συγκεκριμένο χρονικό διάστημα, το τμήμα των δεδομένων αποστέλλεται και πάλι.

### II. Γρήγορη επαναμετάδοση (Fast retransmit)

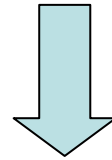
Όταν ο αποστολέας TCP λάβει 3 διπλές επιβεβαιώσεις (duplicates acks) για ένα τμήμα δεδομένων X, υποθέτει ότι το τμήμα δεδομένων Y που ακολουθεί το X έχει χαθεί.

## Μέθοδοι εύρεσης απώλειας πακέτου

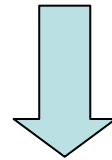
- Επαναστέλει το Υ χωρίς να περιμένει να λήξει ο Χρόνομετρητής Επαναμετάδοσης (Retransmission Timer)
- Η Γρήγορη επαναμετάδοση (Fast Retransmit) χρησιμοποιεί την παράμετρο dupthresh
- Έχει οριστεί ότι στα 3 dupacks συμπαιράνουμε ότι το δίκτυο έχει απορρίψει ένα πακέτο

## Απόδοση TCP – Επαναδρομολόγηση πακέτων

- I. Όταν σε ένα δίκτυο συμβαίνει επαναδρομολόγηση, ο παραλήπτης μπορεί να στείλει πάνω από 3 διπλές επιβεβαιώσεις (dupacks).



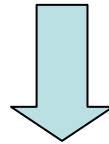
Γρήγορη Επαναμετάδοση (Fast Retransmit)



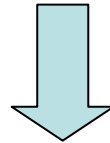
Απώλεια bandwidth λόγω άχρηστης επαναμετάδοσης

## Απόδοση TCP – Επαναδρομολόγηση πακέτων

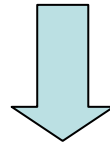
- II. Το πρωτόκολλο μεταφοράς του TCP υποθέτει συμφόρηση (congestion) όταν ένα πακέτο χάνεται



Όταν το TCP λάβει 3 διαδοχικές επιβεβαιώσεις (dupacks) υποθέτει απώλεια πακέτου και άρα συμφόρηση (congestion)



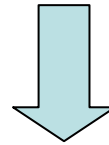
Μείωση του παραθύρου στο μισό του αρχικού



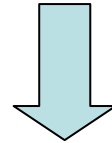
Σημαντική μείωση στο ρυθμό της μετάδοσης

## Απόδοση TCP – Επαναδρομολόγηση πακέτων

- III. Ο παραλήπτης πρέπει να κρατά (buffer out) τα δεδομένα που του έρχονται εκτός σειράς



Σημαντικό βάρος για τον παραλήπτη



Μειώνεται η συνολική αποτελεσματικότητα του συστήματος



## Βασική ιδέα επέκτασης του TCP

- Να επιτρέπεται η διάκριση της απώλειας ή επαναμετάδοσης του πακέτου όταν ο αποστολέας λάβει μια dupack.
- EPDN (Explicit Packet Drop Notification)  
Καταγραφή των πακέτων που χάνονται και ενημέρωση του παραλήπτη για αυτά
- RD-TCP (Reorder Detecting TCP)  
Σύμφωνα με τις πληροφορίες του παραλήπτη ο αποστολέας ενημερώνεται αν έχει το πακέτο έχει επαναδρομολογηθεί ή χαθεί

Πρόβλημα

Το πακέτο

έχει χαθεί

ή

επαναδρομολογηθεί

## Προτεινόμενη λύση – EPDN - RD-TCP

- Διατηρείται ένας πίνακας διασποράς (hashtable) σε κάθε ροή
- Κρατώ
  - α. Μέγιστο Αριθμό της Ακολουθίας (Maximum Sequence Number)
  - β. Ελάχιστο Αριθμό της Ακολουθίας (Minimum Sequence Number)

που χάνονται κάθε φορά στη δίοδο.

## Προτεινόμενη λύση – EPDN- RD-TCP

- Κάθε φορά που έρχεται νέο πακέτο από τη ροή  $i$ , η δίοδος εισάγει το Maximum Sequence Number και το Minimum Sequence Number των πακέτων που έχουν χαθεί στο επόμενο πακέτο των δεδομένων
- Η εγγραφή διαγράφεται από τη δομή
- Ακολουθεί ο παρακάτω έλεγχος για τη μέγιστη και ελάχιστη εισαγωγή που υπάρχει στον πίνακα από τον παραλήπτη του TCP

## Προτεινόμενη λύση –EPDN- RD-TCP

Αν συμβαίνει κάποιο από τα παρακάτω

Υπάρχει Κενό ανάμεσα :

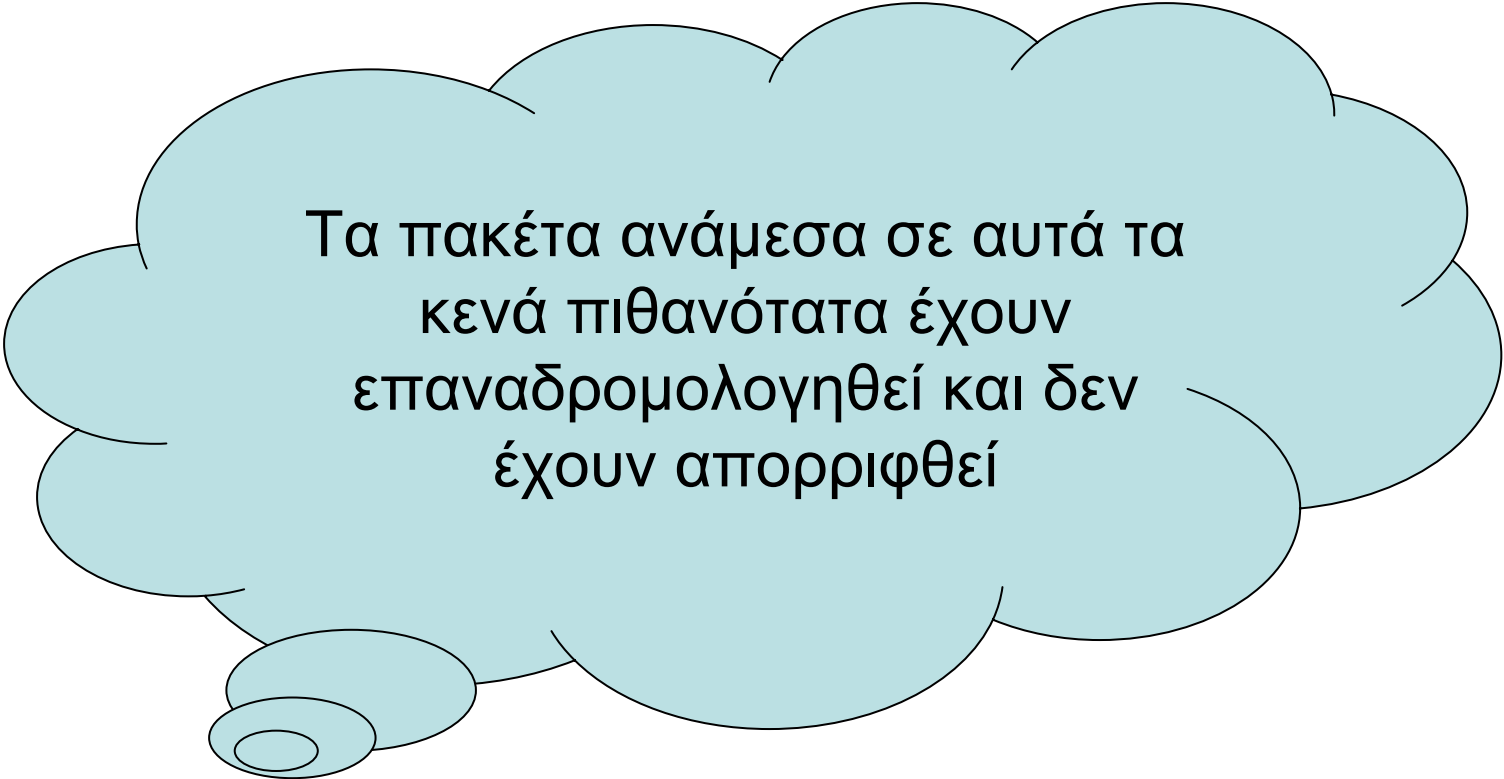
Minimum Sequence Number  
Τελευταίο πακέτο που έχει ληφθεί

ή

Μέγιστη απορριφθείσα εισαγωγή  
(Maximum dropped entry)

Τρέχων πακέτο που λαμβάνεται

## Προτεινόμενη λύση – EPDN- RD-TCP



Τα πακέτα ανάμεσα σε αυτά τα  
κενά πιθανότατα έχουν  
επαναδρομολογηθεί και δεν  
έχουν απορριφθεί

## Προτεινόμενη λύση - EPDN RD-TCP

- Αν δεν υπάρχουν κενά τότε τα πακέτα που βρίσκονται ανάμεσα στα

Τελευταίο πακέτο που έχει ληφθεί

Τωρινό πακέτο που λαμβάνεται

- Το πιο πιθανό που συμβαίνει είναι ότι τα πακέτα αυτά έχουν χαθεί στη δίοδο

## Προτεινόμενη λύση – EPDN - RD-TCP

- Αν όταν παραλήπτης λάβει ένα πακέτο και το ελάχιστο και μέγιστο στοιχείο του πίνακα είναι άδεια, τότε πιθανότατα ισχύει ότι

Τα πακέτα ανάμεσα στο τελευταίο  
ληφθέν πακέτο και το τρέχων  
πακέτο έχουν πιθανότατα  
επαναδρομολογηθεί



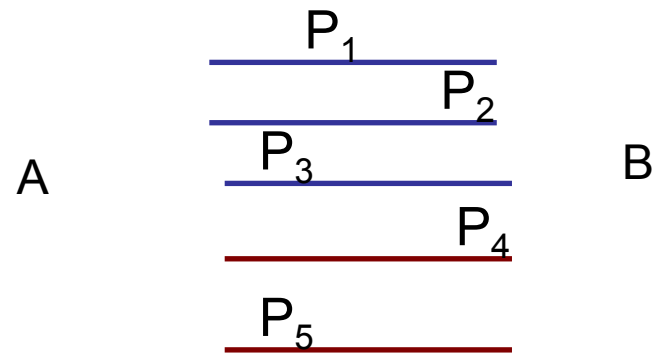
## Προτεινόμενη λύση – EPDN RD-TCP

- Επομένως ο παραλήπτης του TCP μπορεί να αποφασίσει αν τα κενά που υπάρχουν ανάμεσα σε πακέτα που βρίσκονται εκτός σειράς έχουν προκληθεί από επαναδρομολόγηση ή όχι.
- Ο TCP αποστολέας ενημερώνεται για την αιτία
- Αν τα πακέτα έχουν απορριφθεί στο δίκτυο, ο TCP αποστολέας περιμένει  $3+k$  dupacks μέχρι να στείλει ξανά τα πακέτα

# Παράδειγμα

- Έστω το παρακάτω δίκτυο :

Κόμβο πηγής A  
Κόμβο προορισμού B  
Ενδιάμεσοι δίοδοι  $R_1 R_2$



( Δίοδος  $R_1$  ————— Δίοδος  $R_2$  ————— )

## Παράδειγμα

- Έστω ότι η  $R_1$  απορρίπτει τα πακέτα  $P_2, P_3$  λόγω συμφόρησης στο δίκτυο.

- Τότε

$P_2$  : Minimum Sequence Number

$P_3$  : Maximum Sequence Number

- Όταν το πακέτο  $P_4$  περνάει μέσα από τη δίοδο τότε αποκτά για Minimum entry την τιμή  $P_2$  και για το Maximum entry την τιμή  $P_3$

- Ο κόμβος B όταν λάβει το πακέτο  $P_4$ , ελέγχει αν αυτό είναι εκτός σειράς

## Παράδειγμα

- Διαπιστώνει ότι αυτό συμβαίνει, επομένως στη συνέχεια ελέγχει τις τιμές των Maximum, Minimum Sequence Numbers.
- Αφού δεν υπάρχουν κενά ανάμεσα στο P2 (Minimum Sequence Number) και στο P1 (τελευταίο πακέτο σε σειρά), ο λήπτης γνωρίζει ότι τα πακέτα P2, P3 έχουν απορριφθεί.
- Επομένως στις dupacks που αποστέλλει δεν υπάρχει κάποιο ενδεικτικό bit που να δηλώνει επαναδρομολόγηση.
- Ο TCP αποστολέας γνωρίζει ότι το πακέτο έχει απορριφθεί και επαναστέλλει μετά από 3 dupacks

## Παράδειγμα 2

- Έστω η παρακάτω περίπτωση
  - Πακέτο  $P_2$  έχει επαναδρομολογηθεί
  - Πακέτο  $P_3$  έχει απορριφθεί στη δίοδο  $R_1$
- Στον πίνακα διασποράς

Εισαγωγή για το  
μέγιστο :  $P_3$

Εισαγωγή για το  
ελάχιστο :  $P_3$

• Όταν το  $P_4$  περάσει από τη δίοδο παίρνει την τιμή του  $P_3$  για μέγιστη και ελάχιστη εισαγωγή, και η εισαγωγή του πίνακα διασποράς διαγράφεται

## Παράδειγμα 2

- Όταν το πακέτο  $P_4$  φθάσει στον κόμβο B, ελέγχεται αν υπάρχουν κενά
- Διαπιστώνεται ότι υπάρχουν κενά ανάμεσα στο τελευταίο ληφθέν πακέτο (δηλαδή του  $P_1$ ) και του ελάχιστου αριθμού για το πακέτο  $P_4$  (που είναι το  $P_3$ )
- Με βάση τα παραπάνω ο παραλήπτης γνωρίζει ότι το πακέτο  $P_2$  έχει πιθανότητα επαναδρομολογηθεί
- Επιπλέον ο παραλήπτης κρατά μία λίστα με όλα τα πακέτα που έχουν επαναδρομολογηθεί
- Το πακέτο  $P_2$  θα μπει σε αυτή τη λίστα

## Παράδειγμα 2

- Ο παραλήπτης του TCP θέτει το bit της επαναδρομολόγησης για τις dupacks που θα αποστείλει
- Άρα ο αποστολέας του TCP γνωρίζει με βάση το bit αυτό ότι το πακέτο έχει επαναδρομολογηθεί
- Πριν αποστείλει και πάλι το πακέτο ο αποστολέας περιμένει  $3+k$  dupacks.
- Υποθέτει όμως ότι ο παραλήπτης θα έχει λάβει ωστόσο το επαναδρομολογημένο πακέτο. Όταν συμβεί αυτό, το πακέτο διαγράφεται από τη λίστα και στέλνεται μια ack στον αποστολέα

## Παράδειγμα 3

- Υπάρχει περίπτωση **πακέτα που έχουν απορριφθεί** στη δίοδο ο παραλήπτης **λανθασμένα** να θεωρήσει ότι έχουν **επαναδρομολογηθεί** και να θέσει στο αντίστοιχο bit ότι έχουν επαναδρομολογηθεί
- Έστω ότι τα πακέτα
  - $P_1, P_2, P_3$  στέλνονται από ένα διάδρομο
  - και τα  $P_4, P_5, P_6$  στέλνονται από ένα άλλο εναλλακτικό διάδρομο
  - Τα  $P_2, P_3$  απορρίπτονται
  - Δεν υπάρχει όμως πακέτο που να ακολουθεί το  $P_3$



## Παράδειγμα 3

- Όταν φθάσει το  $P_4$  ο παραλήπτης ψάχνει τις max – min εισαγωγές και τις βρίσκει άδειες
- Λανθασμένα ο παραλήπτης υποθέτει ότι τα πακέτα  $P_2$  και  $P_3$  έχουν επαναδρομολογηθεί
- Ο παραλήπτης θέτει τα επαναδρομολογημένα bit για τις dupacks ενώ ο αποστολέας περιμένει για  $3+k$  dupacks
- Αν τα πακέτα έχουν απορριφθεί και ο αποστολέας περιμένει για πολύ χρόνο (αν το  $k$  είναι μεγάλο), ο χρονομετρητής λήγει και τότε το πακέτο  $P_2$  επαναστέλλεται
- Ο αποστολέας υποθέτει ότι όλα τα πακέτα ανάμεσα στο  $P_2$  και μέχρι το  $P_4$  έχουν απορριφθεί.
- Αφού λάβει 3 dupacks επαναστέλλει τα πακέτα αυτά.

## Παράμετροι επιλογής για το κ

1-η Περίπτωση : Μικρή τιμή για το κ

- Το TCP θα συνεχίσει να κάνει μη απαραίτητες επαναληπτικές αποστολές

2-η Περίπτωση : Μεγάλη τιμή για το κ

Υπάρχει ο κίνδυνος να μην ενεργοποιηθεί η επαναπροστολή και να λήξει το χρονικό της περιθώριο.

(retransmission timeout)

## Ποια η καταλληλότερη τιμή για το κ

- Εξαρτάται
  - Από το μέγεθος της επαναδρομολόγησης
  - Από τις τρέχουσες συνθήκες του δικτύου
- Αν ο TCP αποστολέας δεν εντοπίσει spurious επαναπροστολές μπορεί να αυξήσει την τιμή του κ και έτσι περιορίζεται οι περιττές επαναδρομολογήσεις

1. Ο παραλήπτης ανιχνεύει το ποσοστό της επαναδρομολόγησης
2. Ενημερώνει τον αποστολέα
3. Ο αποστολέας καθορίζει την τιμή του κ με βάση αυτή την πληροφορία

## Υπολογιστικό Κόστος

- Το TCP option field είναι 40 bytes
- Χρησιμοποιούνται 4 bytes για την ελάχιστη και μέγιστη εισαγωγή
- Με βάση αυτή την υλοποίηση διατηρούνται πληροφορίες μόνο για τις ροές των οποίων πακέτα έχουν απορριφθεί.
- Οι πληροφορίες για τα πακέτα που έχουν απορριφθεί εισάγονται στο αντίστοιχο πακέτο που καταφέρνει να περάσει με επιτυχία, και η αντίστοιχη εισαγωγή διαγράφεται
- Επομένως στον πίνακα διασποράς διατηρούνται περιορισμένου βαθμού πληροφορίες

## Αποθήκευση - Υπολογιστικό Κόστος

- Έστω ότι υπάρχουν 200.000 ταυτόχρονες ροές και 10% από αυτές έχουν πληροφορία για ένα ή περισσότερα πακέτα που περνάνε μέσα από μία πύλη
  - Ο πίνακας διασποράς θα έχει περίπου 20.000 εισαγωγές (flow id entry) με 2 εισαγωγές να αντιστοιχούν στη maximum και minimum εισαγωγή αντίστοιχα.
  - Απαιτούνται
    - 4 bytes για κάθε flow id entry
    - 4 bytes για κάθε σειριακό αριθμό πακέτου
    - 4 bytes για κάθε δείκτη
- Άρα συνολική μνήμη που θα χρειαστεί είναι **320KB**

## Αποθήκευση - Υπολογιστικό Κόστος

- Επομένως με μία 500 KB SRAM, μπορεί να υλοποιηθεί η λύση
- Τα υπολογιστικά κόστη αφορούν κυρίως σταθερό χρόνο
- Αν μία ροή έχει απορρίψει κάποια πακέτα, τότε η καταχώρηση της απαιτούμενης πληροφορίας απαιτεί σταθερό χρόνο.
- Σταθερό χρόνο επίσης απαιτεί και η διαγραφή της καταχώρησης

## Αποθήκευση - Υπολογιστικό Κόστος

- Αναλυτικότερα :
- Το κόστος διατήρησης της λίστας με τα επαναδρομολογημένα πακέτα εξαρτάται από τον αριθμό των πακέτων που ο παραλήπτης υποθέτει ότι έχουν επαναδρομοληθεί (μετά την μέτρηση των κενών)
- Άρα κατά την εισαγωγή το υπολογιστικό κόστος είναι  $O(n)$ , όπου  $n$  είναι ο αριθμός των πακέτων που απουσιάζουν μέσα στο κενό
- Για διαγραφή και σύγκριση απαιτείται  $O(m)$ , όπου  $m$  είναι το μήκος της λίστας με τα επαναδρομολογημένα πακέτα



## Συμπεράσματα

- Παρουσιάστηκε σήμερα ένας τρόπος αποφυγής άσκοπων επαναμεταδώσεων που προκύπτουν εξαιτίας της επαναδρομολόγησης που συμβαίνει στο δίκτυο.
- Αυτό έγινε δυνατό με την καταγραφή των πακέτων που έχουν απορριφθεί και την ταυτόχρονη ενημέρωση του αποστολέα για το αν έχουμε απόρριψη ή επαναδρομολόγηση.
- Παραπάνω έρευνα απαιτείται ώστε να είναι γνωστό το μέγεθος της επαναδρομολόγησης ώστε να ενημερώνεται ο αποστολέας του TCP για αυτό το μέγεθος