

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΔΙΚΤΥΩΝ

## ΔΙΑΛΕΞΗ 9

ΔΙΔΑΣΚΩΝ: ΑΝΑΡΓΥΡΟΣ ΣΙΔΕΡΗΣ

`<sideris@epp.teiher.gr>`

`<https://eclass2.teicrete.gr/courses/TP182/>`

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ

ΤΕΙ ΚΡΗΤΗΣ



# TCP Congestion Algorithms



# Γενικά (1)

- Ο ρυθμός μετάδοσης δεδομένων σε μιά TCP κίνηση εξαρτάται από το διαφημιζόμενο παράθυρο (το στέλνει ο παραλήπτης) και το παράθυρο εκπομπής του αποστολέα.
  - Σε ιδανικές συνθήκες (χωρίς απώλειες) το παράθυρο του αποστολέα μετά από κάποιο χρονικό διάστημα θα γίνει ίσο (σχεδόν) με αυτό του παραλήπτη.
    - Γιατί όχι κατευθείαν;
- Για να υπολογίσουμε το θεωρητικό παράθυρο μιας TCP σύνδεσης χρησιμοποιούμε το παρακάτω τύπο:
  - $Win=B*RTT$ :
    - Όπου B η ταχύτητα της σύνδεσης μας, RTT ο χρόνος μεταξύ αποστολής ενός TCP πακέτου και η λήψη της επιβεβαίωσης του και Win το μέγεθος του παραθύρου (συνήθως σε bytes).



# Γενικά (2)

- Ερώτηση:
  - Η ταχύτητα σύνδεσης μας είναι 100 Mbps, και το RTT 100 ms. Πόσο είναι το παράθυρο που θα πρέπει να διαφημίσει ο πελάτης στον αποστολέα;
    -
  - Σε περιπτώσεις συμφόρησης το παράθυρο εκπομπής μειώνεται και είναι γνωστό σαν παράθυρο συμφόρησης:
    - Γιατί μειώνεται;
  - Μετά τη μείωση τι γίνεται;
  - Υπάρχουν διάφοροι αλγόριθμοι ελέγχου συμφόρησης στο TCP:
    - Διαφέρουν στο τρόπο μείωσης και αύξησης του παραθύρου συμφόρησης.

# Γενικά (3)

- Κατηγορίες TCP αλγόριθμων ελέγχου συμφόρησης:
  - Loss Based: BIC, HTCP, CUBIC, HSTCP, STCP
  - Delay Based: TCP-Vegas, Fast-TCP
  - Loss and delay Based (Mixed): TCP Africa
  - ECN (Explicit Congestion Notification) Based: XCP
- Η κάθε κατηγορία ελέγχει το παράθυρο συμφόρησης βάση των απωλειών, καθυστέρησης (RTT) και/ή του ECN.
  - Τι είναι το ECN;

# TCP Reno (1)

- Το reno σε περίπτωση συμφόρησης ρίχνει το μέγεθος του παραθύρου συμφόρησης στο μισό.
  - $W=W/2$
- Μετά το αυξάνει κατά 1 segment:
  - $W=W+(1/W)$
- Το  $W$  σχετίζεται με το λόγο απωλειών με τον εξής τύπο:
  - $W=1.22/(p^{0.5})$
- Δεν είναι ιδανικό για υψηλής ταχύτητας δίκτυα



# TCP Reno (2)

- Στην πράξη:
  - $RTT=100ms$
  - $TCP\ segment=1500\ bytes.$
  - $Bandwidth=1Gbps$
- Τι  $window$  χρειαζόμαστε για να πιάσω όλο το band;
  - $W=RTT*B=((1 \times 10^9)/8\ Bps) * 100 * (10^{-3})\ sec=12500000\ byte=12500000/1500= 8334\ segments$
- Και ποιό είναι το  $packet\ loss$  για να πιάσω το band;
  - $W=1.22/(p^{0.5}) \Rightarrow p=(1.22/W)^2=(1.22/8334)^2=2.18 \times 10^{-8}$



# TCP Reno (3)

- Και άμα γίνει συμφόρηση πόσο χρόνο θα χρειαστεί για να ξαναχρησιμοποιήσω όλο το band;
  - Το νέο  $W=8334/2=4167$ . Άρα χρειαζόμαστε άλλα τόσα για να φτάσουμε στο προ συμφόρησης παράθυρο.
  - Άρα  $T=4167*0.1=416.7$  sec.



# STCP (1)

- Το scalable TCP σε περίπτωση συμφόρησης ρίχνει το μέγεθος του παραθύρου συμφόρησης.
  - $W=W-(W*0.125)$
- Μετά το αυξάνει κατά:
  - $W=W+(0.01*W)$
- Το  $W$  σχετίζεται με το λόγο απωλειών με τον εξής τύπο:
  - $W=0.0745/p$
- Είναι ιδανικό για υψηλής ταχύτητας δίκτυα



# STCP (2)

- Στην πράξη:
  - $RTT=100ms$
  - $TCP\ segment=1500\ bytes.$
  - $Bandwidth=1Gbps$
- Τι  $window$  χρειαζόμαστε για να πιάσω όλο το band;
  - $W=RTT*B=((1 \times 10^9)/8\ Bps) * 100 * (10^{-3})\ sec=12500000\ byte=12500000/1500= 8334\ segments$
- Και ποιό είναι το  $packet\ loss$  για να πιάσω το band;
  - $W=0.0745/p \Rightarrow p=(0.0745/W)=(0.0745/8334)=8.9 \times 10^{-6}$



# STCP (3)

- Και άμα γίνει συμφόρηση πόσο χρόνο θα χρειαστεί για να ξαναχρησιμοποιήσω όλο το band;
- Το νέο  $W=8334-(0.125*8334)=7293$ . Άρα χρειαζόμαστε άλλα 1041 για να φτάσουμε στο προ συμφόρησης παράθυρο.
- Άρα  $T=((8334-7293)/(7293)*0.01)-1)*0.1=1.4$  sec.



# TCP Reno vs STCP

- Δεδομένα:
  - $RTT=200ms$
  - $TCP\ segment=1500\ bytes.$
  - $Bandwidth=10Gbps$
- Σε περίπτωση συμφόρησης πόσο γρήγορα φτάνουμε το αρχικό παράθυρο εκπομπής.
  - $W=B*RTT=( (10Gbps*0.2sec) / 8) / 1500=166667\ segments$
  - Reno:  $W=W/2=83333, time=83333*0.2=16666\ sec$
  - STCP=  $W=W-(0.125*166667)=145832,$   
 $time=(( (166667-145832) / (145832) * 0.01) - 1) * 0.2=2.7\ sec.$



# ΒΙΒΛΙΟΓΡΑΦΙΑ

- "Δίκτυα Υπολογιστών", A.S. Tanenbaum, Εκδόσεις Παπασωτηρίου, 3η έκδοση.
- "The TCP/IP Guide", Charles M. Kozierok, [http://www.tcpiptide.com/free/t\\_IPSecurityIPSecProtocols.htm](http://www.tcpiptide.com/free/t_IPSecurityIPSecProtocols.htm), τελευταία πρόσβαση 20/05/2010.
- "802.1X Port-Based Authentication HOWTO", Lars Strand, [http://tldp.org/HOWTO/html\\_single/8021X-HOWTO/](http://tldp.org/HOWTO/html_single/8021X-HOWTO/), τελευταία πρόσβαση 20/05/2010.