

## **6 ΕΠΙΠΕΔΟ ΜΕΤΑΦΟΡΑΣ**

### **6.1 ΕΙΣΑΓΩΓΗ**

Το επίπεδο μεταφοράς εποπτεύει την μεταφορά των δεδομένων από το υπολογιστή του αποστολέα στο υπολογιστή του παραλήπτη, δηλαδή την μεταφορά μεταξύ των δύο τελικών υπολογιστικών συστημάτων (end-to-end communication). Προσφέρει δύο ειδών υπηρεσίες στο ανώτερο επίπεδο: α. Υπηρεσίες συνδεσμιικού χαρακτήρα και β. Υπηρεσίες ασυνδεσμιικού χαρακτήρα.

### **6.2 ΥΠΗΡΕΣΙΕΣ ΣΥΝΔΕΣΙΜΙΚΟΥ ΧΑΡΑΚΤΗΡΑ (CONNECTION – ORIENTED SERVICES)**

Σε αυτού του είδους τις υπηρεσίες ο αποστολέας ειδοποιεί το παραλήπτη ότι ετοιμάζεται να στείλει δεδομένα, ώστε ο δεύτερος να είναι έτοιμος για την παραλαβή τους. Η διαδικασία αυτή είναι γνωστή ως διαδικασία χαιρετισμού (handshaking) ανάμεσα στα δύο τελικά συστήματα επικοινωνίας (End Systems ή Hosts). Όταν η διαδικασία χαιρετισμού ολοκληρωθεί, λέμε ότι μια ‘σύνδεση’ αποκαταστάθηκε ανάμεσα στους δύο κεντρικούς υπολογιστές που επικοινωνούν. Ο όρος ‘συνδεσμιικού χαρακτήρα (connection-oriented)’ οφείλεται στο γεγονός ότι, στην πραγματικότητα δεν υπάρχει κάποια στενή σύνδεση μεταξύ των υπολογιστών, αλλά γίνεται μια συμφωνία για τον τρόπο μετάδοσης ανάμεσά τους.

Οι υπηρεσίες συνδεσμιικού χαρακτήρα έχουν ως αντικειμενικό σκοπό την παροχή της αποκαλούμενης αξιόπιστης μεταφοράς δεδομένων (Reliable Data Transfer), τον έλεγχο της ροής των δεδομένων και τον έλεγχο συμφόρησης.

Με τον όρο **αξιόπιστη μεταφορά δεδομένων** περιγράφεται η δυνατότητα να παραδίδονται στο παραλήπτη τα δεδομένα χωρίς να έχουν υποστεί σφάλματα (errors) και στην σωστή σειρά.

**Έλεγχος ροής (Flow Control)** σημαίνει ότι ο αποστολέας θα πρέπει να στέλνει τα δεδομένα με τέτοιο ρυθμό ώστε να μην κατακλύζεται από αυτά ένας ενδεχομένως αργός στην επεξεργασία τους παραλήπτης.

**Έλεγχος συμφόρησης (Congestion Control)** σημαίνει ότι τα δύο τελικά συστήματα που επικοινωνούν πρέπει να έχουν την δυνατότητα να ενημερώνονται για τυχόν συμφόρηση πακέτων δεδομένων στο Διαδίκτυο, ώστε να ελαττώνουν το ρυθμό αποστολής δεδομένων για να αποφεύγονται κρούσματα απώλειας κάποιων εξ αυτών.

#### **6.2.1 ΤΟ ΠΡΩΤΟΚΟΛΛΟ ΕΛΕΓΧΟΥ ΜΕΤΑΔΟΣΗΣ (TRANSMISSION CONTROL PROTOCOL - TCP)**

Στη σχεδίαση του πρωτότυπου ARPANET το υποδίκτυο θεωρούνταν ότι παρείχε υπηρεσία νοητού κυκλώματος (δηλαδή, τέλειας αξιοπιστίας). Το πρώτο επίπεδο μεταφοράς, το NCP (Network Control Protocol – Πρωτόκολλο Ελέγχου Δικτύου), σχεδιάστηκε με υπόθεση την ύπαρξη ενός τέλειου υποδικτύου. Απλώς μετέφερε TPDUs

(που ονομάζονταν μηνύματα) στο επίπεδο δικτύου και υπέθετε ότι θα παραδίδονταν όλα με τη σειρά στον προορισμό. Η εμπειρία έδειξε ότι το ARPANET ήταν πράγματι αρκετά αξιόπιστο ώστε το πρωτόκολλο να είναι πλήρως επιτυχές για κυκλοφορία μέσα στο ARPANET.

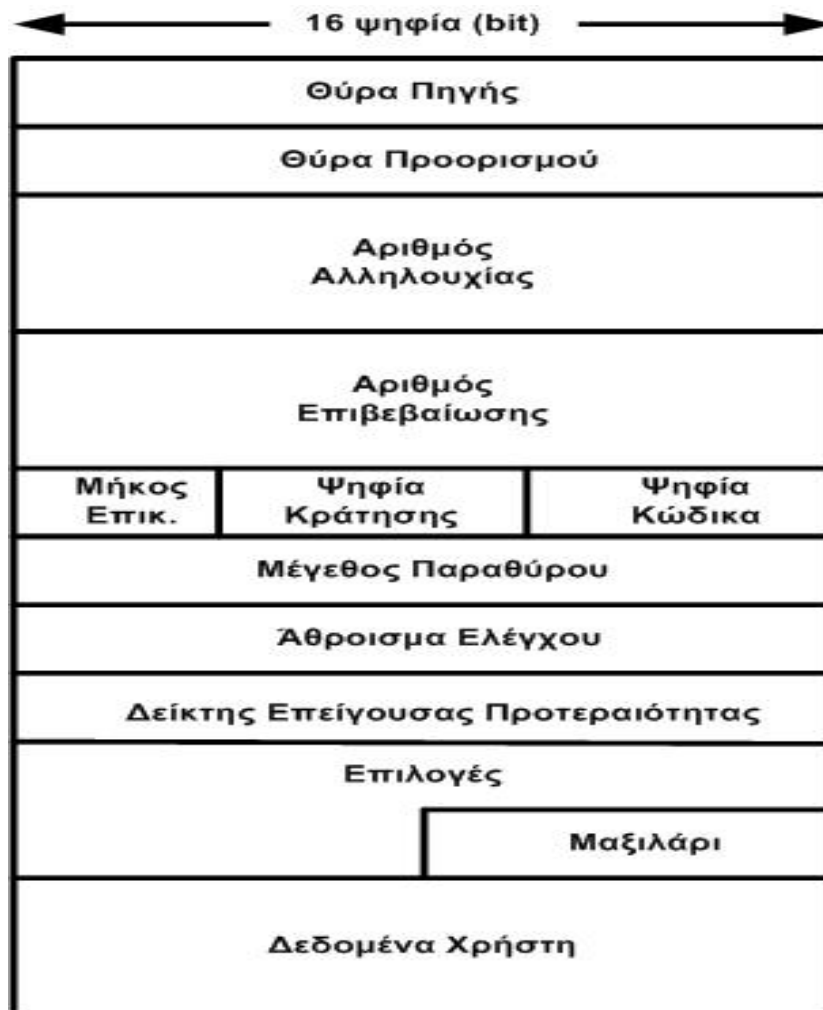
Ωστόσο, με το πέρασμα του χρόνου, καθώς το ARPANET μετεξελίχθηκε σε ARPA Internet που περιείχε πολλά δίκτυα LAN, ένα υποδίκτυο ραδιομετάδοσης πακέτων και πολλά δορυφορικά κανάλια, η από άκρο σε άκρο (end-to-end) αξιοπιστία του δικτύου μειώθηκε. Η ανάπτυξη αυτή οδήγησε σε μια βασική αλλαγή στο επίπεδο μεταφοράς και στη σταδιακή εισαγωγή ενός καινούργιου πρωτοκόλλου μεταφοράς, του TCP (Transmission Control Protocol – Πρωτόκολλο Ελέγχου Μετάδοσης), που ήταν ακριβώς σχεδιασμένο για την ανοχή ενός αναξιόπιστου υποδικτύου (τύπου C στην ορολογία του OSI). Σχετικό με το TCP ήταν ένα καινούργιο πρωτόκολλο επιπέδου δικτύου, το IP. Σήμερα, το TCP/IP δεν χρησιμοποιείται μόνο από το ARPANET και στο Internet ARPA, αλλά επίσης και σε πολλά εμπορικά συστήματα.

Η σύγχρονη έκδοση του πρωτοκόλλου TCP είναι επισήμως ορισμένη στο RFC 793. Ένας αριθμός βελτιώσεων και προδιαγραφών υλοποίησης έχει προστεθεί σε δεύτερη φάση. Εκείνα που απαιτήθηκαν για μια καλύτερη υλοποίηση του TCP κατά το 1989 βρίσκονται καταχωρημένα στο RFC 1122. Από τότε, ένας αριθμός επιπρόσθετων αλλαγών έχει καταγραφεί. Αυτές οι αλλαγές θα αναφερθούν όποτε χρειαστεί.

#### **6.2.1.1 Το πακέτο TCP**

Το πακέτο TCP (TCP Segment) αποτελείται από την επικεφαλίδα και από τα δεδομένα.

---



**Εικόνα 6.1:** Το πακέτο TCP (Επικεφαλίδα και δεδομένα)

Αναλυτικά στο πακέτο TCP περιλαμβάνονται τα ακόλουθα πεδία:

- Θύρα Πηγής (Sequence Port – 16 bit): Προσδιορίζει τη διεύθυνση της εφαρμογής του αποστολέα.
- Θύρα Προορισμού (Source Number – 16 bit): Προσδιορίζει τη διεύθυνση της εφαρμογής του παραλήπτη.
- Αριθμός Ακολουθίας (Sequence Number – 32 bit): Καθορίζει τη θέση μέσα του πρώτου byte δεδομένων του εκάστοτε πακέτου, κατά τη διάρκεια της μεταφοράς του ως ακολουθία από bytes. Αν, π.χ. ο αριθμός αλληλουχίας ενός πακέτου είναι 100 και στο κάθε πακέτο περιέχονται 5 bytes δεδομένων, τότε ο αριθμός αλληλουχίας του επόμενου πακέτου θα είναι 105.
- Αριθμός Επιβεβαίωσης (Acknowledgement Number – 32 bit): Εκφράζει του επόμενου byte δεδομένων, που περιμένει να λάβει ο δέκτης. Επίσης, δηλώνει ότι, ο παραλήπτης έχει λάβει όλα τα προηγούμενα bytes δεδομένων.

- Μήκος Επικεφαλίδας (Header Length – 4 bit): Δηλώνει το μέγεθος της επικεφαλίδας, ώστε να το γνωρίζει ο παραλήπτης, δεδομένου ότι το πεδίο ‘Επιλογές’ είναι μεταβλητού μεγέθους.
- Ψηφία Κράτησης (Reserved – 6 bit): Αυτά τα bit είναι δεσμευμένα για μελλοντική χρήση.
- Ψηφία Κώδικα (Code bit – 6 bit): Το κάθε ένα από τα ψηφία αυτά έχει λειτουργική σημασία για το πακέτο. Συγκεκριμένα:
  1. URG: Δηλώνει αν είναι έγκυρος ο δείκτης επείγουσας προτεραιότητας.
  2. ACK: Δηλώνει αν είναι έγκυρος ο αριθμός επιβεβαίωσης.
  3. PSH: Δηλώνει στην οντότητα TCP αν πρέπει να περάσει τα δεδομένα απευθείας στο επίπεδο εφαρμογής, ή αν μπορεί να τα αποθηκεύσει σε κάποιον ενταμιευτή, περιμένοντας να συσσωρευτούν πολλά.
  4. RST: Δηλώνει στην οντότητα TCP του παραλήπτη ότι πρέπει να τερματιστεί η σύνδεση, λόγω ανωμαλιών λειτουργίας.
  5. SYN: Ζητάει τη δημιουργία σύνδεσης.
  6. FIN: Ειδοποιεί τον παραλήπτη ότι ο αποστολέας δεν έχει άλλα στοιχεία να αποστείλει.
- Μέγεθος Παραθύρου (Window Size – 16 bit): Δηλώνει τον αριθμό των πακέτων που επιθυμεί να λάβει ο αποστολέας. Αυτό το πεδίο βοηθάει στη ρύθμιση του κυκλοφοριακού και εμποδίζει τη δημιουργία συμφόρησης.
- Άθροισμα Ελέγχου (Checksum – 16 bit): Ανιχνεύει τα σφάλματα στο πακέτο TCP.
- Δείκτης Επείγουσας Προτεραιότητας (Urgent Pointer – 16 bit): Αυτός ο δείκτης δηλώνει αν κάποια δεδομένα πρέπει να παραδοθούν άμεσα.
- Μαξιλάρι (Padding – 8 bit): Διαχωρίζει την επικεφαλίδα από τα δεδομένα.
- Επιλογές (Options – 24 bit): Καλύπτει κάποιες επιπλέον λειτουργίες. Η πιο σημαντική επιλογή είναι το Μέγιστο Μέγεθος Πακέτου (MSS – Maximum Segment Size) μπορεί να λάβει ο αποστολέας. Μια άλλη επιλογή είναι ο χρόνος επιβίωσης του κάθε πακέτου (TTL) που δηλώνει πόσο μπορεί το πακέτο να παραμείνει στο Διαδίκτυο, χωρίς να φθάσει στον προορισμό του.
- Δεδομένα (Data): Πρόκειται για το πακέτο του παραπάνω επιπέδου (της εφαρμογής).

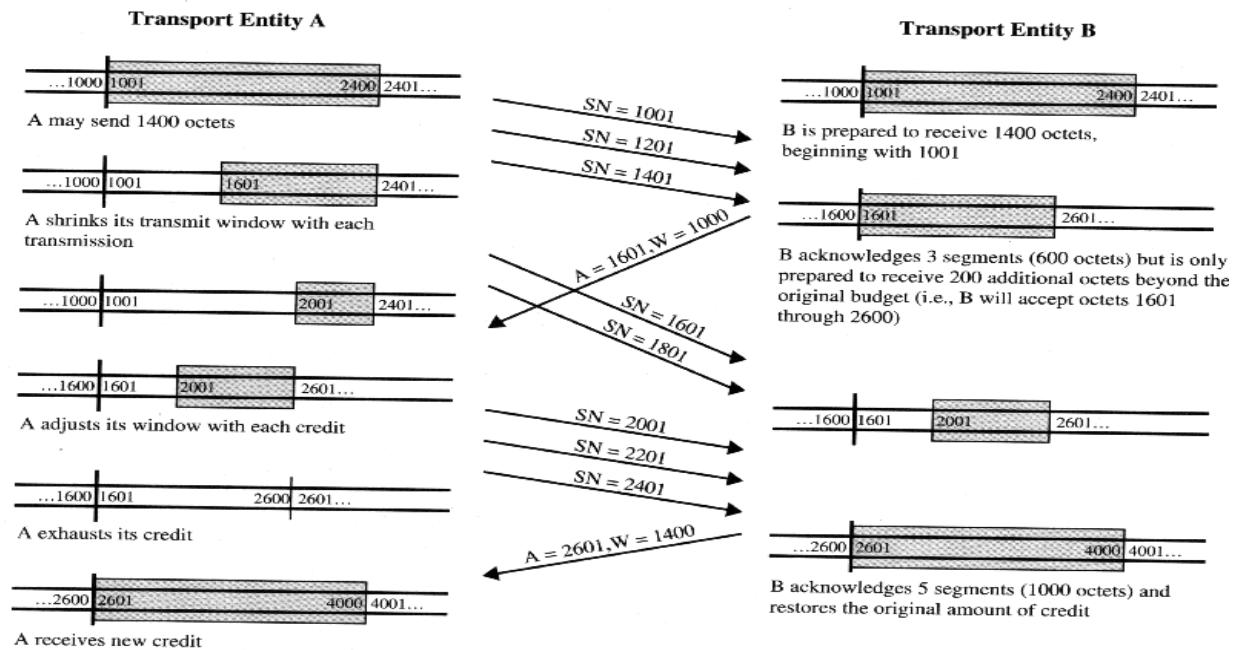
### 6.2.1.2 Έλεγχος Ροής στο TCP

Όπως συμβαίνει με τα περισσότερα πρωτόκολλα που παρέχουν έλεγχο ροής, το TCP χρησιμοποιεί μια μορφή μηχανισμού ολισθαίνοντος παραθύρου. Αυτός διαφέρει από το μηχανισμό που χρησιμοποιείται στα άλλα πρωτόκολλα, όπως είναι το LLC, το HDLC και το X.25, στο ότι αυτός αντιγράφει (decouples) αναγνώριση των λαμβανόμενων μονάδων δεδομένων από την παραχώρηση αδείας να αποστείλει επιπρόσθετες μονάδες δεδομένων.

Ο μηχανισμός ελέγχου ροής που χρησιμοποιείται από το TCP είναι γνωστός ως σχέδιο κατανομής πίστωσης (credit allocation scheme). Για το σχέδιο αυτό, κάθε ανεξάρτητο byte δεδομένων που μεταδίδεται, θεωρείται ότι έχει έναν αριθμό διάταξης (Sequence Number). Όταν μια οντότητα TCP στέλνει ένα πακέτο, περιλαμβάνει σε αυτό τον αριθμό διάταξης του πρώτου byte στο πεδίο δεδομένων του πακέτου. Μια οντότητα

TCP αναγνωρίζει ένα εισερχόμενο πακέτο με ένα μήνυμα της μορφής ( $A = i, W = j$ ), που έχει την παρακάτω μετάφραση:

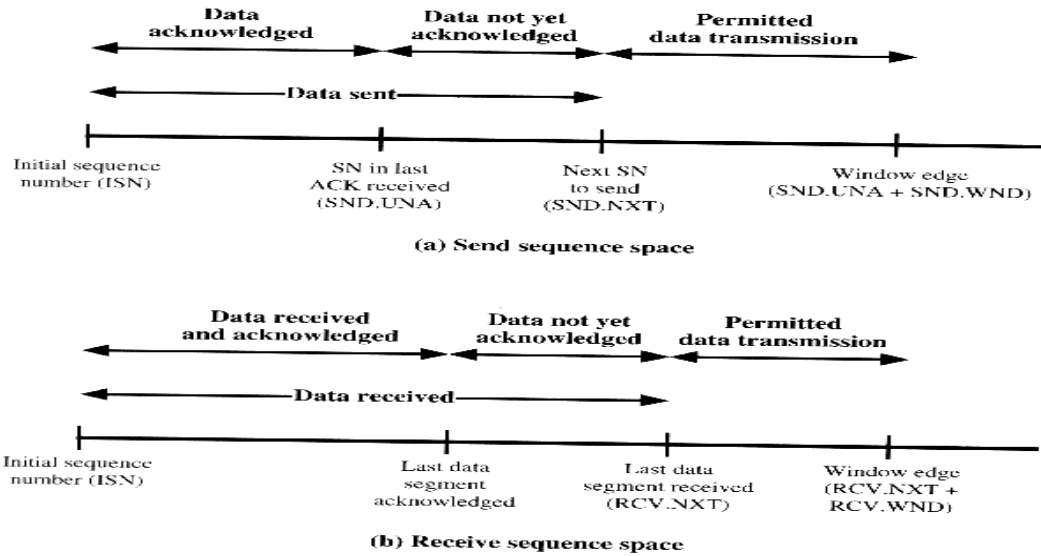
- Όλα τα bytes μέχρι τον αριθμό διάταξης  $i - 1$  αναγνωρίζονται, ενώ το επόμενο προς αναμονή byte έχει αριθμό διάταξης  $i$ .
- Παραχωρείται η άδεια για την αποστολή ενός επιπρόσθετου παραθύρου ( $W$ ) με  $j$  bytes δεδομένων. Αυτό σημαίνει, τα  $j$  bytes που αντιστοιχούν μεταξύ των αριθμών διάταξης  $i$  και  $i + j - 1$ .



**Εικόνα 6.2: Παράδειγμα του μηχανισμού ελέγχου ροής πακέτων στο πρωτόκολλο TCP**

Η εικόνα 6.2 υλοποιεί το μηχανισμό. Για απλότητα, η ροή των δεδομένων παρουσιάζεται μόνο προς τη μια κατεύθυνση και υποτίθεται πως 200 bytes δεδομένων αποστέλλονται σε κάθε πακέτο. Αρχικά, μέσω της διαδικασίας δημιουργίας της σύνδεσης, οι αποστελλόμενοι και λαμβανόμενοι αριθμοί διάταξης συγχρονίζονται και στην οντότητα A παραχωρείται μια αρχική κατανομή πίστωσης 1400 bytes, ξεκινώντας με τον αριθμό byte 1001. Αφού στείλει 600 bytes σε τρία πακέτα, η οντότητα A έχει συρρικνώσει το παράθυρό της στο μέγεθος των 800 bytes (αριθμοί διάταξης 1601 μέχρι 2400). Ακολουθεί η λήψη των πακέτων αυτών, η οντότητα B επιβεβαιώνει τη λήψη όλων των bytes μέχρι του 1601, και εκπέμπει πίστωση για 1000 bytes. Αυτό σημαίνει ότι η A μπορεί να στείλει τα bytes από 1601 έως 2600 (πέντε πακέτα). Ωστόσο, ως τη χρονική στιγμή που φθάνει το σήμα της B στην A, η A έχει ήδη στείλει δύο πακέτα, που περιλαμβάνουν τα bytes από 1601 ως 2000 (τα οποία είναι επιτρεπτά εξαιτίας της αρχικής κατανομής πίστωσης). Επομένως, η υπόλοιπη πίστωση για την A στο σημείο αυτό είναι μόνο 600 bytes (3 πακέτα). Καθώς προχωρεί η ανταλλαγή, η οντότητα A

εκμεταλλεύεται την πίσω άκρη του παραθύρου της κάθε φορά που μεταδίδει και εκμεταλλεύεται την μπροστινή άκρη μόνο όταν της παραχωρείται πίστωση.



**Εικόνα 6.3:** Η άποψη του μηχανισμού ελέγχου ροής από την πλευρά του αποστολέα (a) και του παραλήπτη (b)

Η εικόνα 6.3 παρουσιάζει την άποψη του μηχανισμού και από τα δύο άκρα, της αποστολής και της λήψης. Τυπικά, και τα δύο άκρα παρατηρούνται διότι τα δεδομένα μπορούν να ανταλλάσσονται και προς τις δύο κατευθύνσεις (full duplex).

Ο μηχανισμός κατανομής πίστωσης είναι πολύ ευέλικτος. Για παράδειγμα, ας υποθεθεί ότι το τελευταίο μήνυμα που εξέπεμψε η οντότητα B ήταν ( $A = i, W = j$ ) και ότι το τελευταίο byte δεδομένων που έλαβε η B ήταν το byte  $i - 1$ . Τότε

Για να αυξήσει την πίστωση σε ένα ποσό  $k$  ( $k > j$ ), όταν δεν έχουν φθάσει επιπρόσθετα δεδομένα, η B εκπέμπει το σήμα ( $A = i, W = k$ ).

Για να αναγνωρίσει ένα εισερχόμενο πακέτο που περιέχει  $m$  bytes δεδομένων ( $m < j$ ), χωρίς παραχώρηση επιπρόσθετης πίστωσης, η οντότητα B εκπέμπει ( $A = i + m, W = j - m$ ).

Σημειώνεται ότι ο παραλήπτης δεν απαιτείται να επιβεβαιώσει αμέσως εισερχόμενα πακέτα, αλλά μπορεί να περιμένει και να εκπέμψει ένα συσσωρευτικό μήνυμα αναγνώρισης για έναν αριθμό πακέτων.

Ο παραλήπτης χρειάζεται να υιοθετήσει κάποια πολιτική που αφορά το ποσό των δεδομένων που αυτός επιτρέπει στον αποστολέα να μεταδώσει. Η συντηρητική προσέγγιση είναι να επιτρέψει μόνο καινούργια πακέτα, ως το όριο του διαθέσιμου διαστήματος του ενταμιευτή. Εάν αυτή η πολιτική ήταν σε εφαρμογή στην εικόνα 6.2, το πρώτο μήνυμα πίστωσης υπαγορεύει ότι η B έχει 1000 διαθέσιμα bytes στον ενταμιευτή της, και το δεύτερο μήνυμα ότι η B έχει 1400 διαθέσιμα bytes στον ενταμιευτή της.

Ένα σχέδιο συντηρητικού ελέγχου ροής μπορεί να περιορίσει το ρυθμό εξυπηρέτησης της σύνδεσης μεταφοράς σε περιπτώσεις μεγάλης καθυστέρησης. Ο παραλήπτης θα μπορούσε, ενδεχομένως, να αυξήσει το ρυθμό εξυπηρέτησης, αν αισιόδοξα παραχωρούσε πίστωση για κάποιο διάστημα στον ενταμιευτή το οποίο δε διαθέτει. Για παράδειγμα, αν ένας ενταμιευτής του παραλήπτη είναι γεμάτος, αλλά αυτός προσδοκεί ότι θα ελευθερώσει χώρο για 1000 bytes εντός ενός εκτιμούμενου χρόνου κυκλικού-ταξιδιού, μπορεί αμέσως να στείλει μια πίστωση για 1000 bytes. Αν ο παραλήπτης μπορέσει να συγχρονιστεί με τον αποστολέα, αυτό το σχέδιο μπορεί να αυξήσει το ρυθμό εξυπηρέτησης και δε μπορεί να προκαλέσει καμία ζημιά. Ωστόσο, αν ο αποστολέας είναι πιο γρήγορος από τον παραλήπτη, κάποια πακέτα ίσως να απορριφθούν, καθιστώντας αναγκαία μια αναμετάδοση. Επειδή με μια αξιόπιστη υπηρεσία δικτύου (κατά την απουσία συμφόρησης διαδικτύου), οι αναμεταδόσεις δεν είναι διαφορετικά αναγκαίες, ένα βέλτιστο σχέδιο ελέγχου ροής θα περιπλέξει το πρωτόκολλο.

### 6.2.1.3 Η Επίδραση του Μεγέθους του Παραθύρου στην Επίδοση

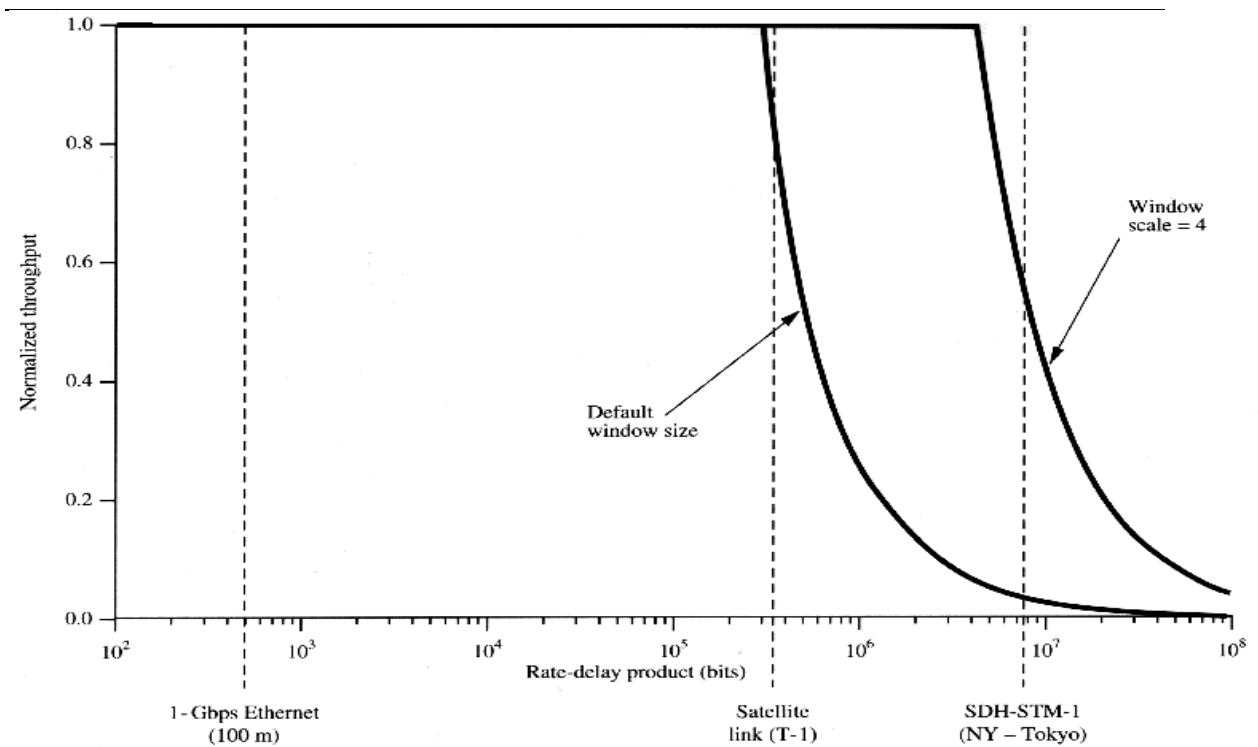
Σε μια σύνδεση TCP είναι εφικτό να προσδιοριστεί ο μέγιστος δυνατός ρυθμός εξυπηρέτησης (Throughput). Ο ρυθμός εξυπηρέτησης εξαρτάται από το μέγεθος του παραθύρου, την καθυστέρηση διάδοσης και το ρυθμό των δεδομένων. Στην περίπτωση του TCP, το μέγεθος του παραθύρου και ο αριθμός διάταξης αναφέρονται σε ανεξάρτητα bytes. Στην περίπτωση αυτή χρησιμοποιούνται οι παρακάτω συμβολισμοί:

- W = Το μέγεθος παραθύρου του TCP (σε bytes).
- R = Ο ρυθμός δεδομένων (data rate – σε bps) στην πηγή TCP που είναι διαθέσιμη σε μια δεδομένη σύνδεση TCP.
- D = Η καθυστέρηση διάδοσης (propagation delay – σε seconds) ανάμεσα στην πηγή TCP και στον προορισμό πάνω σε μια δεδομένη σύνδεση TCP.

Για λόγους απλούστευσης, ας αγνοηθούν τα αρχικά bits σε ένα πακέτο TCP. Υποτίθεται ότι η οντότητα-πηγή TCP αρχίζει να μεταδίδει μια ακολουθία bytes σε μια σύνδεση προς έναν προορισμό. Θα χρειαστούν D δευτερόλεπτα για το πρώτο byte να φθάσει στον προορισμό και ένας επιπρόσθετος χρόνος D για την επιβεβαίωση να επιστρέψει. Κατά τη διάρκεια αυτού του χρονικού διαστήματος, η πηγή, αν δεν είναι περιορισμένη, θα μπορούσε να μεταδώσει ένα σύνολο 2RD bits, ή RD/4 bytes. Στην πραγματικότητα, η πηγή περιορίζεται σε μέγεθος παραθύρου W bytes έως ότου παραληφθεί μια επιβεβαίωση. Επομένως, εάν  $W > RD/4$ , τότε ο μέγιστος δυνατός ρυθμός εξυπηρέτησης μπορεί να επιτευχθεί σε αυτή τη σύνδεση. Εάν  $W < RD/4$ , τότε ο μέγιστος κατορθωτός κανονικοποιημένος ρυθμός εξυπηρέτησης είναι ακριβώς ο λόγος του D προς RD/4. Έτσι, ο κανονικοποιημένος ρυθμός εξυπηρέτησης S μπορεί να εκφραστεί ως:

$$S = \begin{cases} 1 & W > RD/4 \\ \frac{4W}{RD} & W < RD/4 \end{cases} \quad (6.1)$$

Η εικόνα 1.4 παρουσιάζει το μέγιστο κατορθωτό ρυθμό εξυπηρέτησης σαν μια συνάρτηση του γινομένου RD. Το μέγιστο μέγεθος παραθύρου είναι  $2^{16} - 1 = 65.535$  bytes. Αυτό επαρκεί για τις περισσότερες εφαρμογές. Για παράδειγμα, η εικόνα δείχνει ότι το γινόμενο RD για ένα 1-Gbps Ethernet με ένα μήκος 100 m είναι λιγότερο από  $10^3$  bits. Ακόμα και στην περίπτωση μιας T-1 (1544 Mbps) δορυφορικής σύνδεσης, το μέγιστο μέγεθος παραθύρου παρέχει καλή επίδοση. Ωστόσο, κάποιος μπορεί να φανταστεί συνδέσεις δικτύων για τις οποίες το default μέγεθος παραθύρου είναι ανεπαρκές. Ένα παράδειγμα, που εμφανίζεται στην εικόνα, είναι μια οπτική σύνδεση SDH (Synchronous Digital Hierarchy) που λειτουργεί στα 155 Mbps ανάμεσα σε δύο απόμακρα σημεία. Για τέτοιες περιπτώσεις, ο παράγοντας κλιμάκωσης παραθύρου μπορεί να χρησιμοποιηθεί για τη βελτίωση του δυναμικού ρυθμού εξυπηρέτησης. Η εικόνα δείχνει μια χρήση του παράγοντα κλιμάκωσης παραθύρου με τιμή 4, που αυξάνει το μέγεθος παραθύρου στα  $2^{20} - 1 \approx 10^6$  bytes.



**Εικόνα 6.4: Η επίδραση του παράγοντα κλιμάκωσης παραθύρου**

Η εικόνα 6.4 δίνει μια αίσθηση της δυναμικής επίδοσης μιας σύνδεσης TCP. Ωστόσο, υπάρχουν πολλοί περίπλοκοι παράγοντες που πρέπει να ληφθούν υπόψη, συμπεριλαμβανομένων των παρακάτω:



1. Στις περισσότερες περιπτώσεις, ένας αριθμός συνδέσεων TCP πολυπλέκονται πάνω στην ίδια διασύνδεση δικτύου, έτσι σε κάθε σύνδεση απλώς κατανέμεται ένα κλάσμα της διαθέσιμης χωρητικότητας. Αυτό μειώνει το R και για αυτό ελαττώνει την αναποτελεσματικότητα.
2. Από την άλλη πλευρά, πολλές συνδέσεις TCP σχετίζονται με άλματα κατά μήκος πολλαπλών δικτύων. Σε αυτήν την περίπτωση, το D είναι το άθροισμα των καθυστερήσεων κατά μήκος του κάθε δικτύου και των καθυστερήσεων σε κάθε δρομολογητή (router) κατά μήκος του μονοπατιού (path). Οι καθυστερήσεις στους δρομολογητές συχνά είναι οι μεγαλύτεροι συντελεστές για το D.
3. Η τιμή του R, που αναφέρθηκε στη συνάρτηση 6.1, αναφέρεται στο ρυθμό των δεδομένων που είναι διαθέσιμος στη σύνδεση, στην οντότητα-πηγή TCP. Εάν αυτός ο ρυθμός δεδομένων είναι μεγαλύτερος από το ρυθμό δεδομένων που αντιμετωπίζεται κατά τα άλματα από την πηγή στον προορισμό, τότε μια προσπάθεια για μετάδοση σε υψηλότερο ρυθμό δεδομένων θα προκαλέσει μπουτλιάρισμα σε κάποιο δρομολογητή, αυξάνοντας το D.
4. Αν οποιαδήποτε πακέτα χαθούν και πρέπει να αναμεταδοθούν, ο ρυθμός εξυπηρέτησης ελαττώνεται. Το μέγεθος της επιρροής των χαμένων πακέτων εξαρτάται από την πολιτική αναμετάδοσης, η οποία θα μελετηθεί σε επόμενη παράγραφο.

Στα σύγχρονα διαδίκτυα, ελάχιστα πακέτα χάνονται εξαιτίας σφαλμάτων μετάδοσης στη γραμμή. Μάλλον οι περισσότερες απώλειες πακέτων οφείλονται σε απόρριψη πακέτων από συμφόρηση στους δρομολογητές ή από συμφόρηση στους μεταγωγείς των δικτύων. Ο έλεγχος ροής είναι ένα ζήτημα που αφορά το δίκτυο ή το διαδίκτυο σαν σύνολο. Ωστόσο, υπάρχουν μέτρα που μπορεί να ληφθούν από ανεξάρτητες οντότητες TCP για την αποφυγή ή την απαλλαγή από τη συμφόρηση, τα οποία περιγράφονται σε παρακάτω παραγράφους.

#### 6.2.1.4 Στρατηγική Αναμετάδοσης

Όπως και με τα πρωτόκολλα ελέγχου σύνδεσης (Link Control Protocols), το TCP περιλαμβάνει έναν έλεγχο λαθών παράλληλα με ένα σχέδιο ελέγχου ροής. Στην περίπτωση του TCP, δεν υπάρχει καθόλου σαφής αρνητική αναγνώριση, όπως είναι η REJ και SREJ που απαντώνται στα πρωτόκολλα ελέγχου σύνδεσης. Μάλλον, το TCP βασίζεται αποκλειστικά στη θετική αναγνώριση και αναμετάδοση όταν μια αναγνώριση δε φθάσει εντός μιας δεδομένης χρονικής περιόδου (Timeout Period).

Δύο γεγονότα καθιστούν αναγκαία την αναμετάδοση ενός πακέτου. Πρώτα, το πακέτο μπορεί να έχει καταστραφεί κατά τη μεταφορά αλλά εντούτοις να έχει φθάσει στον προορισμό. Το άθροισμα ελέγχου που περιλήφθηκε με το πακέτο καθιστά ικανή την οντότητα λήψης να ανιχνεύσει το λάθος και να απορρίψει το πακέτο. Το πιο συνηθισμένο ενδεχόμενο είναι ότι ένα πακέτο αποτυγχάνει να φθάσει. Σε οποιαδήποτε περίπτωση, η οντότητα αποστολέας δε γνωρίζει ότι η μετάδοση του πακέτου ήταν ανεπιτυχής.

Αν ένα πακέτο δε φθάσει με επιτυχία, κανένα σήμα επιβεβαίωσης ACK δε θα αποσταλεί και μια αναμετάδοση είναι επακόλουθη. Για να αντιμετωπιστεί αυτή η κατάσταση, πρέπει να υπάρχει ένα χρονικό όριο που να σχετίζεται με το κάθε πακέτο

καθώς αυτό αποστέλλεται. Αν περάσει αυτό το χρονικό όριο προτού αναγνωρισθεί το πακέτο, ο αποστολέας θα πρέπει να το αναμεταδώσει.

Ένα σημαντικό ζήτημα κατά το σχεδιασμό του TCP είναι η τιμή του χρόνου αναμετάδοσης. Αν η τιμή είναι πολύ μικρή, θα υπάρξουν πάρα πολλές χωρίς λόγο αναμεταδόσεις, σπαταλώντας χωρητικότητα του δικτύου. Αν η τιμή είναι πάρα πολύ μεγάλη, το πρωτόκολλο θα είναι νωθρό στην αντιμετώπιση ενός χαμένου πακέτου. Το χρονικό όριο θα πρέπει να είναι λίγο μεγαλύτερο από την καθυστέρηση κυκλικού-ταξιδιού (αποστολή πακέτου, λήψη ACK). Φυσικά, αυτή η καθυστέρηση είναι μεταβλητή ακόμη και κάτω από σταθερού ρυθμού φόρτωση του δικτύου. Χειρότερα, οι στατιστικές της καθυστέρησης θα διακυμαίνονται αλλάζοντας τις συνθήκες του διαδικτύου.

Υπάρχουν δύο στρατηγικές. Μια σταθερή τιμή χρονικού ορίου μπορεί να χρησιμοποιηθεί, βασισμένη στην κατανόηση της τυπικής συμπεριφοράς του διαδικτύου. Αυτή η στρατηγική υποφέρει από μια ανικανότητα στην ανταπόκριση κατά τις αλλαγές συνθηκών του διαδικτύου. Αν η τιμή τεθεί πολύ ψηλά, η υπηρεσία θα είναι πάντοτε νωθρή. Αν αυτή τεθεί πολύ χαμηλά, μια θετική κατάσταση ανάδρασης μπορεί να αναπτυχθεί, κατά την οποία η συμφόρηση του διαδικτύου οδηγεί σε περισσότερες αναμεταδόσεις, αυξάνοντας συνεχώς τη συμφόρηση.

Ένα σχέδιο προσαρμογής παρουσιάζει τα δικά του προβλήματα. Υποτίθεται ότι η οντότητα TCP κρατάει ίχνη του χρόνου που διαρκεί η αναγνώριση πακέτων δεδομένων και θέτει το χρονικό όριο αναμετάδοσης με βάση τη μέση τιμή των παρατηρούμενων καθυστερήσεων. Αυτή η τιμή δεν μπορεί να είναι αξιόπιστη για τρεις λόγους:

- Η αντίστοιχη οντότητα TCP του άλλου άκρου μπορεί να μην επιβεβαιώσει αμέσως ένα πακέτο. Υπενθυμίζεται το γεγονός ότι αυτή έχει το προνόμιο συσσωρευμένων επιβεβαιώσεων – αναγνωρίσεων πακέτων.
- Αν ένα πακέτο έχει αναμεταδοθεί, ο αποστολέας δε μπορεί να ξέρει αν ένα σήμα ACK αφορά μια απάντηση της αρχικής μετάδοσης ή της αναμετάδοσης του πακέτου.
- Οι συνθήκες σε ένα διαδίκτυο μπορεί να αλλάζουν απότομα.

Το πρόβλημα δεν έχει κάποια ολοκληρωμένη λύση. Πάντα θα υπάρχει μια αβεβαιότητα που αφορά την καλύτερη τιμή για το χρονικό όριο αναμετάδοσης. Στην επόμενη παράγραφο παρουσιάζεται ο υπολογισμός του χρονικού ορίου αναμετάδοσης όπως πραγματοποιείται από το RFC 793. Παρακάτω εξετάζονται περισσότερες στρατηγικές επεξεργασίας για την επιλογή του χρονικού ορίου.

### 6.2.1.5 Χρόνος Προσαρμογής Αναμετάδοσης

Καθώς οι συνθήκες του δικτύου ή του διαδικτύου αλλάζουν, ένα στατικό χρονικό όριο αναμετάδοσης (Retransmission Timer) είναι πιθανό να είναι είτε πολύ μεγάλο είτε πολύ μικρό. Επομένως, εικονικά όλες οι υλοποιήσεις TCP προσπαθούν να εκτιμήσουν την πιο πρόσφατη καθυστέρηση κυκλικού-ταξιδιού (round-trip delay) παρατηρώντας τα δείγματα των καθυστερήσεων για τα πρόσφατα πακέτα, και μετά θέτουν το χρονικό όριο σε μια τιμή λίγο μεγαλύτερη από την εκτιμημένη καθυστέρηση κυκλικού-ταξιδιού.

Μια προσέγγιση θα ήταν να χρησιμοποιηθεί η μέση τιμή των χρόνων που παρατηρούνται για έναν αριθμό πακέτων. Αν η μέση τιμή απεικονίζει με ακρίβεια τη μελλοντική καθυστέρηση κυκλικού-ταξιδιού, τότε το χρονικό όριο αναμετάδοσης που θα προκύψει θα αποφέρει καλή επίδοση. Η απλή μέθοδος της μέσης τιμής εκφράζεται από την παρακάτω εξίσωση:

$$ARTT(K+1) = \frac{1}{K+1} \sum_{i=1}^{K+1} RTT(i), \quad (6.2)$$

όπου  $RTT(i)$  είναι ο χρόνος κυκλικού-ταξιδιού που παρατηρείται για το  $i$  – στο μεταδιδόμενο πακέτο και  $ARTT(i)$  είναι ο μέσος χρόνος κυκλικού-ταξιδιού για τα πρώτα  $K$  πακέτα.

Αυτή η έκφραση μπορεί να αναγραφεί ως ακολούθως:

$$ARTT(K+1) = \frac{K}{K+1} ARTT(K) + \frac{1}{K+1} RTT(K+1). \quad (6.3)$$

Με αυτή τη διαμόρφωση, δεν είναι απαραίτητο να υπολογιστεί ξανά ολόκληρο το άθροισμα κάθε φορά.

Παρατηρείται ότι ο κάθε όρος του αθροίσματος έχει το ίδιο βάρος. Αυτό σημαίνει ότι, ο κάθε όρος πολλαπλασιάζεται με την ίδια σταθερά  $1/(K+1)$ . Τυπικά, θα ήταν συνετό να δοθεί μεγαλύτερη βαρύτητα στα περισσότερα πρόσφατα δείγματα γιατί αυτά είναι πιο πιθανό να αντιπροσωπεύουν τη μελλοντική συμπεριφορά. Μια απλή τεχνική για την πρόβλεψη της επόμενης τιμής με βάση την χρονική σειρά των προηγούμενων τιμών, η οποία προσδιορίζεται στο RFC 793, είναι ο εκθετικός μέσος όρος.

$$SRTT(K+1) = \alpha \times SRTT(K) + (1-\alpha) \times RTT(K+1), \quad (6.4)$$

όπου  $SRTT(K)$  καλείται εκτίμηση εξομάλυνσης του χρόνου κυκλικού-ταξιδιού. Χρησιμοποιώντας τη σταθερή τιμή  $\alpha$  ( $0 < \alpha < 1$ ), ανεξάρτητα από τον αριθμό των προηγούμενων παρατηρήσεων, υπάρχει μια περίπτωση όπου όλες οι περασμένες τιμές λαμβάνονται υπόψη, αλλά οι πιο απόμακρες έχουν μικρότερη βαρύτητα. Για να γίνει αυτό περισσότερο ξεκάθαρο, δίδεται η επόμενη επέκταση της εξίσωσης 6.4:

$$SRTT(K+1) = (1-\alpha) \times RTT(K+1) + \alpha(1-\alpha)RTT(K) + \alpha^2(1-\alpha)RTT(K-1) + \dots + \alpha^K(1-\alpha)RTT(1) + \alpha^{K+1}RTT(0).$$

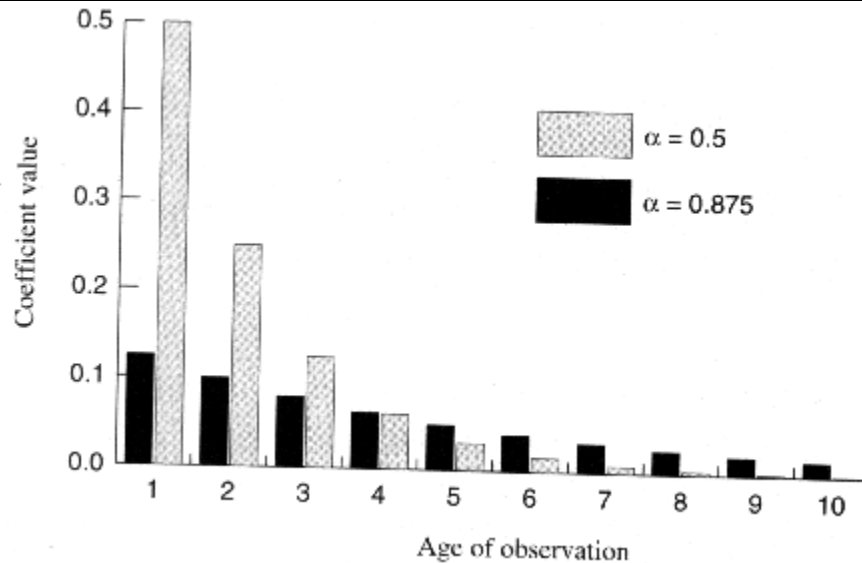
Επειδή τόσο το  $\alpha$  όσο και το  $1-\alpha$  είναι μικρότερα από τη μονάδα, ο κάθε διαδοχικός όρος στον προηγούμενο τύπο είναι μικρότερος. Για παράδειγμα, για  $\alpha = 0.8$  η επέκταση δίνει

$$SRTT(K+1) = 0.2RTT(K+1) + 0.16RTT(K) + 0.128RTT(K-1) + \dots$$

Όσο παλιότερη είναι η παρατήρηση, τόσο πιο μικρή υπολογίζεται κατά το μέσο όρο.

Το μέγεθος του συντελεστή ως συνάρτηση της θέσεώς του στην επεκταμένη εξίσωση παρουσιάζεται στην εικόνα 6.5. Όσο μικρότερη είναι η τιμή του  $\alpha$ , τόσο μεγαλύτερο είναι το βάρος που δίδεται στις πιο πρόσφατες παρατηρήσεις. Για  $\alpha = 0.5$ , ουσιαστικά όλο το βάρος δίνεται στις τέσσερις ή πέντε πιο πρόσφατες παρατηρήσεις, ενώ για  $\alpha = 0.875$ , ο μέσος όρος είναι αποτελεσματικά επεκταμένος για τις δέκα περίπου πιο πρόσφατες παρατηρήσεις. Το πλεονέκτημα της χρήσης μικρής τιμής για το  $\alpha$  είναι

ότι ο μέσος όρος αμέσως θα αντικατοπτρίζει μια απότομη αλλαγή στην παρατηρούμενη ποσότητα. Το μειονέκτημα είναι ότι αν υπάρξει μια απότομη αύξηση στην τιμή της παρατηρούμενης ποσότητας και μετά αυτή οδηγήσει πίσω σε κάποια μέση τιμή, η χρήση μιας μικρής τιμής για το  $\alpha$  θα οδηγήσει λανθασμένες αλλαγές του μέσου όρου.



**Εικόνα 6.5:** Εκθετικοί συντελεστές εξομάλυνσης

Η εικόνα 6.6 συγκρίνει τη μέθοδο του απλού μέσου με αυτή του εκθετικού μέσου (για δύο διαφορετικές τιμές του  $\alpha$ ). Στην εικόνα 6.6a, η παρατηρούμενη τιμή ξεκινάει από το 1, αυξάνεται σταδιακά στην τιμή του 10, και μετά παραμένει εκεί. Στην εικόνα 6.6b, η παρατηρούμενη τιμή ξεκινάει από το 20, μειώνεται σταδιακά στο 10, και έπειτα διατηρείται εκεί. Και στις δύο περιπτώσεις, η διαδικασία ξεκινάει με μια εκτίμηση  $SRTT(0) = 0$ . Σημειώνεται ότι ο εκθετικός μέσος όρος ανιχνεύει τις αλλαγές στη συμπεριφορά της διαδικασίας γρηγορότερα από τον απλό μέσο όρο και ότι η μικρότερη τιμή του  $\alpha$  συντελεί σε μια πιο γρήγορη αντίδραση στην αλλαγή της παρατηρούμενης τιμής.

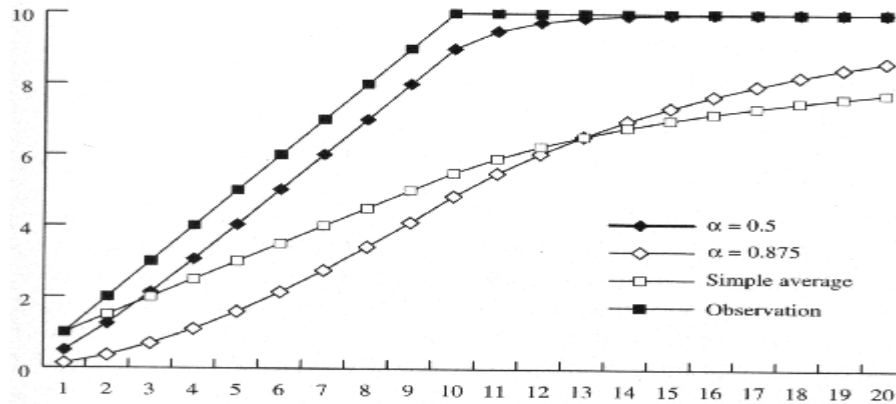
Η εξίσωση 6.4 χρησιμοποιείται στο RFC 793 για την εκτίμηση του πρόσφατου χρόνου κυκλικού-ταξιδιού. Όπως αναφέρθηκε, το χρονικό όριο αναμετάδοσης θα πρέπει να τεθεί σε μια τιμή λίγο μεγαλύτερη από τον εκτιμημένο χρόνο κυκλικού-ταξιδιού. Μια δυνατότητα είναι να χρησιμοποιηθεί η σταθερή τιμή:

$$RTO(K+1) = SRTT(K+1) + \Delta$$

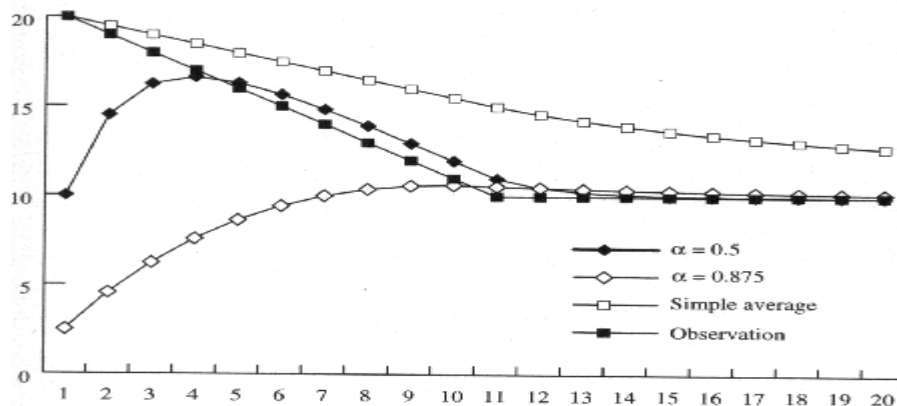
όπου RTO είναι το χρονικό όριο αναμετάδοσης και  $\Delta$  είναι μια σταθερά. Το μειονέκτημα σε αυτό είναι ότι το  $\Delta$  δεν είναι ανάλογο του SRTT. Για μεγάλες τιμές του SRTT, το  $\Delta$  είναι σχετικά μικρό και οι διακυμάνσεις στο πραγματικό RTT θα οδηγήσουν σε μη απαραίτητες αναμεταδόσεις. Για μικρές τιμές του SRTT, το  $\Delta$  είναι σχετικά μεγάλο και προκαλεί μη απαραίτητες καθυστερήσεις σε αναμεταδιδόμενα χαμένα πακέτα. Σύμφωνα με τα παραπάνω, το RFC 793 προσδιορίζει τη χρήση ενός χρονικού ορίου που η τιμή του είναι ανάλογη με το SRTT, εντός κάποιων ορίων:

$$RTO(K+1) = \text{MIN}(\text{UBOUND}, \text{MAX}(\text{LBOUND}, \beta \times \text{SRTT}(K+1))) \quad (6.5)$$

όπου UBOUND και LBOUND είναι προ-διαλεγμένα, σταθερά άνω και κάτω όρια της τιμής του χρονικού ορίου και η  $\beta$  είναι μια σταθερά. Το RFC 793 δεν προτείνει συγκεκριμένες τιμές αλλά δίνει ως παραδείγματα τιμών τα ακόλουθα: το  $\alpha$  ανάμεσα σε 0.8 και 0.9 και το  $\beta$  ανάμεσα σε 1.3 και 2.0.



(a) Increasing function



(b) Decreasing function

Εικόνα 6.6: Η χρήση του εκθετικού μέσου

### 6.2.1.6 Επιλογές Τακτικών Υλοποίησης του TCP

Το πρότυπο του TCP παρέχει έναν ακριβή προσδιορισμό του πρωτοκόλλου που χρησιμοποιείται ανάμεσα στις οντότητες TCP. Ωστόσο, συγκεκριμένες πτυχές του πρωτοκόλλου δέχονται ορισμένες επιλογές υλοποίησης. Αν και δύο υλοποιήσεις που διαλέγουν εναλλακτικές επιλογές θα είναι πρακτικές, ίσως να υπάρξουν λειτουργικές επιπλοκές. Οι περιοχές σχεδιασμού για τις οποίες οι επιλογές προσδιορίζονται είναι οι επόμενες:

- Πολιτική Αποστολής (Send Policy)
- Πολιτική Παράδοσης (Deliver Policy)

- Πολιτική Αποδοχής (Accept Policy)
- Πολιτική Αναμετάδοσης (Retransmit Policy)
- Πολιτική Επιβεβαίωσης (Acknowledgement Policy)

#### **6.2.1.6.1 Πολιτική Αποστολής**

Κατά την απουσία δεδομένων για προώθηση (PUSHed data) και ενός κλεισμένου παραθύρου μετάδοσης, μια οντότητα-αποστολέας TCP είναι ελεύθερη να μεταδώσει δεδομένα σύμφωνα με τη δική της άνεση. Καθώς τα δεδομένα εκπέμπονται από τον χρήστη, αυτά αποθηκεύονται στον ενταμιευτή μετάδοσης. Το TCP μπορεί να κατασκευάσει ένα πακέτο για κάθε δέσμη δεδομένων που παρέχεται από τον χρήστη του ή μπορεί να περιμένει ώσπου ένα συγκεκριμένο ποσό δεδομένων να συσσωρευτεί προτού να κατασκευάσει και να στείλει το πακέτο. Η πραγματική τακτική, που θα ακολουθηθεί, θα εξαρτηθεί από τις θεωρήσεις της λειτουργίας. Αν οι μεταδόσεις είναι σπάνιες και μεγάλες, υπάρχει χαμηλό κόστος όσον αφορά την παραγωγή και την επεξεργασία των πακέτων. Από την άλλη πλευρά, αν οι μεταδόσεις είναι συχνές και μικρές, τότε το σύστημα παρέχει γρήγορη ανταπόκριση.

#### **6.2.1.6.2 Πολιτική Παράδοσης**

Κατά την απουσία εντολής PUSH, μια οντότητα-παραλήπτης TCP είναι ελεύθερη να παραδώσει δεδομένα στον χρήστη σύμφωνα με τη δική της άνεση. Μπορεί να παραδώσει δεδομένα καθώς το κάθε ενδο-διατεταγμένο πακέτο λαμβάνεται, ή μπορεί να αποθηκεύσει δεδομένα από έναν αριθμό πακέτων στον ενταμιευτή λήψης προτού να τα παραδώσει. Η πραγματική τακτική θα εξαρτηθεί από τις θεωρήσεις της λειτουργίας. Αν οι παραδόσεις είναι σπάνιες και μεγάλες, ο χρήστης δε δέχεται τα δεδομένα τόσο γρήγορα όσο θα ήθελε. Από την άλλη πλευρά, αν οι παραδόσεις είναι συχνές και μικρές, ίσως να υπάρξει μη απαραίτητη επεξεργασία τόσο από το TCP όσο και από το λογισμικό του χρήστη, καθώς και ένας αριθμός από μη απαραίτητες διακοπές του λειτουργικού συστήματος.

#### **6.2.1.6.3 Πολιτική Αποδοχής**

Όταν όλα τα πακέτα TCP φθάσουν σε σειρά δια μέσου μιας σύνδεσης TCP, το TCP τοποθετεί τα δεδομένα στον ενταμιευτή λήψης για να τα παραδώσει στον χρήστη. Ωστόσο, είναι δυνατό για τα πακέτα, να φθάσουν χωρίς καμία σειρά. Σε αυτήν την περίπτωση, η οντότητα-παραλήπτης TCP έχει δύο επιλογές:

- Σε σειρά (In Order): Δέχεται μόνο εκείνα τα πακέτα που φθάνουν σε σειρά. Κάθε πακέτο που φθάνει εκτός σειράς απορρίπτεται.
- Σε Παράθυρο (In Window): Δέχεται τα πακέτα που βρίσκονται εντός του παραθύρου λήψης.

Η τακτική In-order χρησιμοποιείται για μια απλή υλοποίηση αλλά δημιουργεί μια επιβάρυνση στην ευκολία του δικτύου, καθώς ο αποστολέας TCP πρέπει να μην

περιμένει και να αναμεταδώσει τα πακέτα που παραλήφθηκαν επιτυχώς αλλά απορρίφθηκαν λόγω λάθος διάταξης. Επιπλέον, αν ένα απλό πακέτο χαθεί στη μετάδοση, τότε όλα τα πακέτα που ακολουθούν θα πρέπει να αναμεταδοθούν καθώς το TCP διακόψει (time out) το χαμένο πακέτο.

Η τακτική In-window ίσως να ελαττώσει τις μεταδόσεις αλλά απαιτεί περισσότερο πολύπλοκη δοκιμή αποδοχής και ένα πιο έξυπνο σχέδιο αποθήκευσης δεδομένων στον ενταμιευτή και θα πρέπει να κρατάει ίχνη για τα δεδομένα που γίνονται δεκτά εκτός σειράς.

#### 6.2.1.6.4 Πολιτική Αναμετάδοσης

Το TCP διατηρεί μια ουρά με τα πακέτα που έχουν σταλεί αλλά δεν έχουν επιβεβαιωθεί ακόμα. Ο προσδιορισμός του TCP υπαγορεύει ότι το TCP θα αναμεταδώσει ένα πακέτο, αν αυτό αποτύχει να λάβει μια επιβεβαίωση εντός ενός δεδομένου χρονικού διαστήματος. Μια υλοποίηση TCP μπορεί να χρησιμοποιήσει μια από τις παρακάτω τρεις στρατηγικές αναμετάδοσης:

- First only: Διατηρεί ένα χρονικό διάστημα αναμετάδοσης για ολόκληρη την ουρά. Αν ληφθεί μια επιβεβαίωση, γίνεται διαγραφή του κατάλληλου πακέτου ή πακέτων από την ουρά και το χρονικό όριο ανανεώνεται. Αν το χρονικό όριο περάσει, γίνεται αναμετάδοση του πακέτου στο μπροστινό μέρος της ουράς και το χρονικό όριο ανανεώνεται.
- Batch: Διατηρεί ένα χρονικό όριο αναμετάδοσης για ολόκληρη την ουρά. Αν ληφθεί μια επιβεβαίωση, γίνεται διαγραφή από την ουρά του κατάλληλου ή των κατάλληλων πακέτων και το χρονικό όριο ανανεώνεται. Αν το χρονικό όριο περάσει, γίνεται αναμετάδοση όλων των πακέτων στην ουρά και το χρονικό όριο ανανεώνεται.
- Individual: Διατηρεί ένα χρονικό όριο για το κάθε πακέτο της ουράς. Αν ληφθεί μια επιβεβαίωση, γίνεται διαγραφή από την ουρά του κατάλληλου ή των κατάλληλων πακέτων και καταστρέφεται το αντίστοιχο χρονικό όριο ή τα αντίστοιχα χρονικά όρια. Αν κάποιο χρονικό όριο περάσει, γίνεται αναμετάδοση του αντίστοιχου πακέτου ανεξάρτητα από τα άλλα, και το χρονικό όριο ανανεώνεται.

Η τακτική First-only είναι αποτελεσματική όσον αφορά την παραγόμενη κυκλοφορία, γιατί μόνο τα χαμένα πακέτα (ή τα πακέτα των οποίων χάθηκε το σήμα ACK) αναμεταδίδονται. Επειδή το χρονικό όριο για το δεύτερο πακέτο στην ουρά δεν τίθεται ώσπου να αναγνωριστεί το πρώτο πακέτο, ωστόσο, μπορούν να υπάρξουν αξιόλογες καθυστερήσεις. Η τακτική Individual επιλύει το πρόβλημα αυτό με κόστος την περισσότερο περίπλοκη υλοποίηση. Η πολιτική Batch επίσης μειώνει την πιθανότητα μεγάλων καθυστερήσεων αλλά μπορεί να αποφέρει μη απαραίτητες αναμεταδόσεις.

Η πραγματική αποτελεσματικότητα της πολιτικής αναμετάδοσης εξαρτάται μερικώς και από την πολιτική αποδοχής του παραλήπτη. Αν ο παραλήπτης χρησιμοποιεί την τακτική αποδοχής In-order, τότε θα απορριφθούν τα πακέτα που λαμβάνονται μετά από ένα χαμένο πακέτο. Αυτό το γεγονός ταιριάζει καλύτερα με την αναμετάδοση τύπου Batch. Αν ο παραλήπτης χρησιμοποιεί τακτική αποδοχής In-window, τότε η καλύτερη

τακτική αναμετάδοσης είναι η First-only ή η Individual. Φυσικά, σε ένα μεικτό δίκτυο υπολογιστών, μπορούν να χρησιμοποιούνται και οι δύο τακτικές αποδοχής.

#### 6.2.1.6.5 Πολιτική Επιβεβαίωσης

Όταν ένα πακέτο δεδομένων φθάσει σε σειρά, ο παραλήπτης TCP έχει δύο επιλογές σχετικά με το χρονικό σημείο στο οποίο θα γίνει η επιβεβαίωση:

- Άμεση (Immediate): Όταν τα δεδομένα γίνονται δεκτά, αμέσως ο παραλήπτης στέλνει ένα άδειο πακέτο που περιέχει τον κατάλληλο αριθμό επιβεβαίωσης.
- Συσσωρευτική (Cumulative): Μόλις τα δεδομένα γίνονται δεκτά, καταγράφεται η ανάγκη για επιβεβαίωση, αλλά ο παραλήπτης περιμένει για ένα εξερχόμενο πακέτο δεδομένων πάνω στο οποίο να στείλει το μήνυμα επιβεβαίωσης. Για την αποφυγή μεγάλων καθυστερήσεων, τίθεται ένα χρονικό όριο παραθύρου. Αν το χρονικό όριο περάσει προτού να σταλεί η επιβεβαίωση, ο παραλήπτης στέλνει ένα άδειο πακέτο που περιέχει τον αριθμό επιβεβαίωσης.

Η τακτική Immediate είναι απλή και κρατάει την οντότητα-αποστολέα TCP συνεχώς ενημερωμένη, πράγμα που περιορίζει τις μη απαραίτητες αναμεταδόσεις. Ωστόσο, η τακτική αυτή αποφέρει επιπλέον μεταδόσεις πακέτων (δηλαδή, των άδειων πακέτων που περιέχουν το μήνυμα ACK). Επιπλέον, η τακτική αυτή μπορεί να αυξήσει το φόρτο του δικτύου. Ας θεωρηθεί ότι μια οντότητα TCP λαμβάνει ένα πακέτο και αμέσως στέλνει το μήνυμα ACK. Έπειτα τα δεδομένα του πακέτου ελευθερώνονται στην εφαρμογή, πράγμα που διευρύνει το παράθυρο λήψης, πυροδοτώντας άλλο ένα πακέτο TCP για να παρέχει επιπλέον πίστωση στην οντότητα-αποστολέα TCP.

Εξαιτίας του δυναμικού κόστους της τακτικής Immediate, η τακτική Cumulative είναι τυπικά εφαρμόσιμη. Αναγνωρίζεται, ωστόσο, ότι η χρήση της τακτικής αυτής απαιτεί περισσότερη επεξεργασία στο άκρο παραλαβής και περιπλέκει το έργο του υπολογισμού του χρόνου κυκλικού-ταξιδιού (round-trip time) από την οντότητα-αποστολέα TCP.

#### 6.2.1.7 Έλεγχος Συμφόρησης

Η συμφόρηση σε ένα δίκτυο ή διαδίκτυο δημιουργεί φανερά προβλήματα στα τερματικά συστήματα (hosts): μειώνει τη διαθεσιμότητα και το ρυθμό εξυπηρέτησης και επιμηκώνει τους χρόνους απόκρισης. Μέσα σε ένα διακλαδισμένο δίκτυο, όπως είναι το δίκτυο μεταγωγής πακέτων (Packet Switching Network) και το δίκτυο αναμετάδοσης πλαισίων σε σταθμούς (Frame Relay Network), μπορεί να χρησιμοποιηθεί δυναμική δρομολόγηση για να βοηθήσει στην ανακούφιση του δικτύου από τη συμφόρηση, απλώνοντας το φορτίο των δεδομένων εξίσου ανάμεσα σε όλους τους μεταγωγείς και τους κόμβους. Ομοίως, αλγόριθμοι δρομολόγησης σε διαδίκτυα μπορούν να απλώσουν το φορτίο των πληροφοριών ανάμεσα στους δρομολογητές (routers) και τα δίκτυα, για να απαλλαγθεί το σύστημα από το φαινόμενο της συμφόρησης. Ωστόσο, αυτά τα μέτρα είναι αποτελεσματικά, όταν αντιμετωπίζουν άνισα φορτία πληροφοριών στο δίκτυο και σχετικά σύντομες αυξήσεις στην κυκλοφορία. Σε τελευταία ανάλυση, η συμφόρηση μπορεί μόνο να ελεγχθεί περιορίζοντας το ολικό ποσό των δεδομένων, που εισάγεται στο



δίκτυο, στο ποσό που το ίδιο το δίκτυο μπορεί να μεταφέρει. Αυτός είναι και ο αντικειμενικός σκοπός όλων των μηχανισμών ελέγχου της συμφόρησης.

Ο έλεγχος της συμφόρησης σε ένα διαδίκτυο βασισμένο στο πρωτόκολλο TCP/IP, είναι ένα πολύπλοκο και δύσκολο εγχείρημα, του οποίου η μελέτη έχει αποφέρει ένα μεγάλο αριθμό ερευνητικών προσπαθειών, πειραμάτων υλοποίησης και σχετικών εγγράφων, μέσα σε μια περίοδο πολλών δεκαετιών. Το θέμα της συμφόρησης είναι πολύ δύσκολο στην αντιμετώπιση του, για τους παρακάτω λόγους:

- Το πρωτόκολλο IP από τη φύση του δεν παρέχει καθόλου τα εφόδια για την ανίχνευση και τον έλεγχο του φαινομένου της συμφόρησης.
- Το πρωτόκολλο TCP παρέχει μόνο έλεγχο ροής από τη μια άκρη στην άλλη (δηλαδή από πομπό σε δέκτη) και μπορεί να συμπεράνει την ύπαρξη συμφόρησης στο διαδίκτυο, μεταξύ των δύο άκρων, με έμμεσο τρόπο. Επιπλέον, η γνώση των συνθηκών του διαδικτύου ή του δικτύου, την οποία παρέχουν οι οντότητες TCP, είναι αναξιόπιστη, επειδή οι καθυστερήσεις είναι ποικίλες και μπορεί να είναι παρατεταμένες.
- Δεν υπάρχει ένας συνεργάσιμος, κατανοητός αλγόριθμος που να δένει μεταξύ τους τις οντότητες TCP. Για αυτό το λόγο, οι οντότητες TCP δε μπορούν να συνεργαστούν ώστε να διατηρήσουν το συνολικό επίπεδο ροής και, πραγματικά, μοιάζουν να ανταγωνίζονται η μια την άλλη για τους διάφορους διαθέσιμους πόρους.

Το μόνο εργαλείο του πρωτοκόλλου TCP που σχετίζεται με τον έλεγχο συμφόρησης στο δίκτυο, είναι η ροή του ολισθαίνοντος παραθύρου και ο μηχανισμός ελέγχου λαθών. Αυτός ο μηχανισμός είναι σχεδιασμένος για τη διαχείριση της κυκλοφορίας, από άκρη σε άκρη, στο δίκτυο. Ωστόσο, ένας αριθμός έξυπνων τεχνικών έχει αναπτυχθεί, ο οποίος καθιστά ικανή την χρήση του μηχανισμού για την ανίχνευση συμφόρησης, την αποφυγή αυτού και την ανάκαμψη του δικτύου. Στο σημείο αυτό, θα γίνει μια εκτεταμένη έρευνα των τεχνικών αυτών.

#### **6.2.1.7.1 Διαχείριση Χρονομέτρου Αναμετάδοσης**

Οι πρώτες τρεις τεχνικές που εξετάζονται, ασχολούνται με τον υπολογισμό του χρόνου αναμετάδοσης (Retransmission Time Out - RTO). Η τιμή αυτού του χρόνου μπορεί να έχει κρίσιμη επίδραση στην αντίδραση των TCP στο φαινόμενο της συμφόρησης. Οι τρεις τεχνικές είναι:

- Η εκτίμηση της διακύμανσης του Χρόνου Κυκλικού Ταξιδιού (RTT – Round Trip Time).
- Εκθετική οπισθοδρόμηση του Χρόνου Αναμετάδοσης (RTO)
- Ο αλγόριθμος του Karn.

#### **A. Εκτίμηση Διακύμανσης του RTT (Αλγόριθμος Jacobson)**

Αυτή η τεχνική αναπτύχθηκε στο Standard Πρωτόκολλο TCP, και περιγράφεται από τις παρακάτω εξισώσεις:

$$SRTT(K+1) = \alpha \times SRTT(K) + (1-\alpha) \times RTT(K+1), \quad (6.6)$$

$$RTO(K+1) = \min(UBOUND, \max(LBOUND, \beta \times SRTT(K+1))). \quad (6.7)$$

Η εξίσωση 6.6 δίνει τον εκτιμώμενο χρόνο κυκλικού ταξιδιού για το μελλοντικό πακέτο που θα σταλεί στο δίκτυο, σε συνάρτηση με τις προηγούμενες τιμές που υπολογίστηκαν από τον πομπό. Η συνάρτηση 6.7 υπολογίζει το χρόνο αναμετάδοσης του μελλοντικού πακέτου από τον πομπό, σε περίπτωση που το πακέτο χαθεί. Δηλαδή υπολογίζει το χρονικό διάστημα που θα περιμένει ο πομπός για την επιβεβαίωση λήψης του πακέτου αυτού. Όπου UBOUND και LBOUND είναι προ-διαλεγμένα όρια για το χρόνο αναμετάδοσης, ενώ το  $\beta$  είναι μια σταθερά.

Οι παραπάνω τύποι δε λειτουργούν ορθά, όταν ο χρόνος κυκλικού ταξιδιού RTT του πακέτου παρουσιάζει μια σχετικά υψηλή διακύμανση. Υπάρχουν κυρίως τρεις λόγοι για μια υψηλή διακύμανση του RTT:

1. Εάν ο ρυθμός των δεδομένων σε μια σύνδεση TCP είναι σχετικά χαμηλός, τότε η καθυστέρηση της μετάδοσης θα είναι σχετικά υψηλή σε σύγκριση με το χρόνο διάδοσης και η διακύμανση στην καθυστέρηση, που οφείλεται στη διακύμανση του μεγέθους των IP πακέτων, θα είναι σημαντική. Για το λόγο αυτό, ο εκτιμητής SRTT επηρεάζεται πάρα πολύ από εκείνα τα χαρακτηριστικά που αποτελούν ιδιότητες των δεδομένων και όχι ιδιότητες του δικτύου.
2. Η φόρτωση κυκλοφορίας και οι συνθήκες στο διαδίκτυο μπορούν να αλλάξουν απότομα εξαιτίας της κυκλοφορίας από άλλες πηγές, προκαλώντας έτσι απότομες αλλαγές στο χρόνο RTT.
3. Η οντότητα – δέκτης TCP μπορεί να μην επιβεβαιώσει κάθε πακέτο αμέσως, εξαιτίας των λειτουργικών της καθυστερήσεων και επειδή ασκεί το προνόμιο που έχει να χρησιμοποιεί συσσωρευτικές επιβεβαιώσεις πακέτων.

Ο αρχικός προσδιορισμός TCP προσπαθεί να ερμηνεύσει αυτή τη μεταβλητότητα πολλαπλασιάζοντας τον εκτιμητή του RTT με κάποιο σταθερό παράγοντα:

$$RTO(K+1) = \beta \times SRTT(K+1),$$

όπου τυπικά χρησιμοποιείται η τιμή  $\beta=2$ . Σε ένα σταθερό περιβάλλον, με μικρή διακύμανση του RTT, αυτή η διατύπωση οδηγεί σε μια χωρίς λόγο υψηλή τιμή του RTO, ενώ σε ένα ασταθές περιβάλλον η τιμή 2 ίσως να είναι ανεπαρκής για την προστασία ενάντια στις χωρίς λόγο αναμεταδόσεις.

Μια πιο αποτελεσματική προσέγγιση είναι να εκτιμηθεί η μεταβλητότητα στις τιμές του RTT και να χρησιμοποιηθεί αυτή ως είσοδος για τον υπολογισμό ενός RTO. Μια δυνατότητα θα ήταν να υπολογιστεί η τυπική απόκλιση του δείγματος. Ωστόσο, αυτό συνεπάγεται τον υπολογισμό ενός τετραγώνου και μιας τετραγωνικής ρίζας. Ένας υπολογισμός της διακύμανσης, που είναι ευκολότερο να εκτιμηθεί, είναι η μέση απόκλιση, που ορίζεται ως:

$$MDEV(X) = E[|X - E[X]|]$$

Όπως και με την εκτίμηση του RTT, μια απλή μέση τιμή θα μπορούσε να χρησιμοποιηθεί για την εκτίμηση του MDEV:

$$AERR(K+1) = RTT(K+1) - ARTT(K)$$

$$\begin{aligned}
ADEV(K+1) &= \frac{1}{K+1} \sum_{i=1}^{K+1} |AERR(i)| \\
&= \frac{K}{K+1} ADEV(K) + \frac{1}{K+1} |AERR(K+1)|
\end{aligned}$$

όπου  $ARTT(K)$  είναι μια απλή μέση τιμή που ορίζεται από την εξίσωση:

$$ARTT(K+1) = \frac{1}{K+1} \sum_{i=1}^{K+1} RTT(i),$$

ενώ  $AERR(K)$  είναι η μέση απόκλιση του δείγματος που μετριέται σε χρόνο  $K$ .

Όπως με τον ορισμό του  $ARRT$ , σε κάθε όρο του αθροίσματος του  $ADEV$  δίνεται ίση βαρύτητα. Αυτό σημαίνει ότι, ο κάθε όρος πολλαπλασιάζεται με την ίδια σταθερά  $1/(K+1)$ . Θα ήταν επιθυμητό να δοθεί μεγαλύτερη βαρύτητα στα πιο πρόσφατα περιστατικά γιατί αυτά είναι πιο πιθανά να αντικατοπτρίζουν τη μελλοντική συμπεριφορά. Ο Jacobson που πρότείνει την χρήση μιας δυναμικής εκτίμησης για τη μεταβλητότητα κατά τον υπολογισμό του  $RTT$  [JACO88], προτείνει την χρήση της ίδιας εκθετικής τεχνικής εξομάλυνσης που χρησιμοποιήθηκε και για τον υπολογισμό του  $SRTT$ . Ολόκληρος ο αλγόριθμος που προτάθηκε από τον Jacobson μπορεί να εκφραστεί ως εξής:

$$\begin{aligned}
SRTT(K+1) &= (1-g) \times SRTT(K) + g \times RTT(K+1) \\
SERR(K+1) &= RTT(K) - SRTT(K) \\
SDEV(K+1) &= (1-h) \times SDEV(K) + h \times |SERR(K+1)| \quad (6.8) \\
RTO(K+1) &= SRTT(K+1) + f \times SDEV(K+1).
\end{aligned}$$

Όπως και στην εξίσωση 1.6, το  $SRTT$  είναι η εκθετική εκτίμηση εξομάλυνσης του  $RTT$ , με  $(1-g)$  ισοδύναμο του  $\alpha$ . Τώρα, ωστόσο, αντί να πολλαπλασιαστεί η εκτίμηση  $SRTT$  με μια σταθερά (εξίσωση 1.7), ένα πολλαπλάσιο της εκτιμούμενης μέσης απόκλισης προστίθεται στο  $SRTT$  για το σχηματισμό του χρόνου αναμετάδοσης. Βασισμένος στα πειράματα χρόνου του, ο Jacobson πρότείνει τις ακόλουθες τιμές για τις σταθερές των τύπων, στην αρχική του εργασία:

$$\begin{aligned}
g &= 1/8 = 0.125 \\
h &= 1/4 = 0.25 \\
f &= 2.
\end{aligned}$$

Μετά από περισσότερες έρευνες [JACO90a], πρότείνει την αλλαγή της τιμής του  $f$  από 2 σε 4, και αυτή είναι η σταθερή τιμή που χρησιμοποιείται στις σύγχρονες υλοποιήσεις.

Η εμπειρία έχει δείξει ότι ο αλγόριθμος του Jacobson μπορεί να βελτιώσει σημαντικά την επίδοση του TCP. Ωστόσο, δε μπορεί να λειτουργήσει αποδοτικά από μόνος. Δύο άλλοι παράγοντες πρέπει να συμπεριληφθούν:

1. Ποια τιμή  $RTO$  πρέπει να χρησιμοποιηθεί για τα αναμεταδιδόμενα πακέτα; Ο αλγόριθμος εκθετικής οπισθοδρόμησης χρησιμοποιείται για το σκοπό αυτό.
2. Ποια δείγματα χρόνου  $RTT$  πρέπει να χρησιμοποιηθούν ως είσοδοι στον αλγόριθμο Jacobson; Ο αλγόριθμος Karn προσδιορίζει ποια δείγματα θα χρησιμοποιηθούν.

## B. Εκθετική Οπισθοδρόμηση του RTO

Όταν για έναν αποστολέα TCP περάσει ο χρόνος επιβεβαίωσης λήψης ενός πακέτου, αυτός πρέπει να στείλει το πακέτο πάλι. Το πρωτόκολλο RFC 793 (πρόγονος του RFC 1122) υποθέτει ότι η ίδια τιμή RTO θα χρησιμοποιηθεί για το αναμεταδιδόμενο πακέτο. Ωστόσο, το γεγονός ότι, η λήξη του χρόνου επιβεβαίωσης οφείλεται προφανώς σε συμφόρηση του δικτύου, υποδηλώνει είτε την απώλεια πακέτου, είτε μεγάλη καθυστέρηση στο χρόνο κυκλικού ταξιδιού (RTT). Με βάση το γεγονός αυτό, η διατήρηση της ίδιας τιμής για το RTO δε θα ήταν καθόλου καλή ιδέα.

Ας θεωρηθεί το παρακάτω σενάριο. Υπάρχει ένας αριθμός ενεργών συνδέσεων TCP από διάφορες πηγές που στέλνουν κυκλοφορία μέσα στο διαδίκτυο. Μια περιοχή συμφόρησης αναπτύσσεται κατά τέτοιο τρόπο ώστε ένας μεγάλος αριθμός πακέτων σε πολλές από αυτές τις συνδέσεις, να χάνεται ή να καθυστερεί, δεδομένου ότι πέρασε το χρονικό περιθώριο RTO των συνδέσεων. Ως αποτέλεσμα, σχεδόν μέσα στον ίδιο χρόνο, πολλά πακέτα θα αναμεταδοθούν στο διαδίκτυο, διατηρώντας ή ακόμα αυξάνοντας τη συμφόρηση. Όλες οι πηγές, τότε, περιμένουν έναν τοπικό (σε κάθε σύνδεση) χρόνο RTO και προβαίνουν πάλι σε αναμετάδοση πακέτων. Αυτή η συμπεριφορά μπορεί να προκαλέσει μια συνεχή κατάσταση συμφόρησης.

Μια πιο λογική πολιτική αναφέρει ότι, η κάθε πηγή TCP αυξάνει το χρόνο RTO κάθε φορά που ένα πακέτο αναμεταδίδεται. Αυτή η διαδικασία είναι γνωστή ως διαδικασία οπισθοδρόμησης (backoff process). Στο σενάριο της προηγούμενης παραγράφου, αφού γίνει η πρώτη αναμετάδοση ενός πακέτου σε κάθε επηρεασμένη σύνδεση, οι πηγές TCP θα περιμένουν μεγαλύτερο χρόνο προτού εκτελέσουν μια δεύτερη αναμετάδοση. Αυτό μπορεί να δώσει στο διαδίκτυο τον απαιτούμενο χρόνο ώστε να σταματήσει το φαινόμενο συμφόρησης. Αν μια δεύτερη αναμετάδοση κριθεί αναγκαία, η κάθε πηγή TCP θα περιμένει για ένα ακόμα μεγαλύτερο χρονικό διάστημα προτού ξεκινήσει μια ενδεχόμενη τρίτη αναμετάδοση, δίνοντας έτσι στο διαδίκτυο ένα ακόμα μεγαλύτερο χρονικό διάστημα να συνέλθει.

Μια απλή τεχνική για την υλοποίηση της διαδικασίας οπισθοδρόμησης του RTO, είναι να πολλαπλασιαστεί το RTO ενός πακέτου με μια σταθερή τιμή για την κάθε αναμετάδοση. Δηλαδή:

$$RTO = q \times RTO. \quad (6.9)$$

Η εξίσωση 1.9 προκαλεί στον χρόνο RTO εκθετική αύξηση σε κάθε αναμετάδοση. Η τιμή που χρησιμοποιείται κυρίως για τη σταθερά  $q$  είναι το 2. Με αυτή τη τιμή, η όλη τεχνική αναφέρεται στη βιβλιογραφία ως δυαδική εκθετική οπισθοδρόμηση. Αυτή η τεχνική χρησιμοποιείται και στο πρωτόκολλο Ethernet CSMA/CD.

## Γ. Ο Αλγόριθμος του Karn

Αν κανένα πακέτο δεν αναμεταδίδεται, η διαδικασία δειγματοληψίας του αλγορίθμου Jacobson είναι ακριβής. Ο RTT για κάθε πακέτο μπορεί να συμπεριληφθεί στον υπολογισμό. Ωστόσο, ας υποθεθεί ότι για ένα πακέτο έχει λήξει ο χρόνος επιβεβαίωσης και πρέπει να αναμεταδοθεί. Αν μια επιβεβαίωση ληφθεί εκ των υστέρων, υπάρχουν δύο πιθανότητες:

- Πρόκειται για την επιβεβαίωση (ACK) της πρώτης μετάδοσης του πακέτου. Σε αυτή την περίπτωση, ο RTT είναι απλά μεγαλύτερος από το αναμενόμενο, αλλά είναι μια σωστή αντανάκλαση των συνθηκών του δικτύου.
- Πρόκειται για την επιβεβαίωση στη δεύτερη μετάδοση.

Η πηγή TCP δε μπορεί να διακρίνει ανάμεσα σε αυτές τις δύο περιπτώσεις. Αν η δεύτερη περίπτωση είναι η σωστή και η οντότητα TCP μετράει τον RTT από την πρώτη μετάδοση ως την λήψη του σήματος ACK, ο μετρημένος χρόνος θα είναι υπερβολικά μεγάλος. Ο μετρημένος χρόνος RTT θα είναι του μεγέθους του πραγματικού RTT συν το RTO. Τροφοδοτώντας τον αλγόριθμο του Jacobson με αυτό το λανθασμένο RTT, θα παραχθεί μια υπερβολικά μεγάλη τιμή για το SRTT και επομένως για το RTO. Επιπλέον, αυτό το αποτέλεσμα δημιουργεί έναν αριθμό επαναλήψεων, αφού η τιμή SRTT της μιας επανάληψης είναι τιμή εισόδου στην επόμενη επανάληψη.

Μια ακόμα χειρότερη προσέγγιση θα ήταν ο υπολογισμός του RTT από τη δεύτερη μετάδοση ως τη λήψη του σήματος ACK. Αν το σήμα αφορά την πρώτη μετάδοση του πακέτου, τότε ο μετρημένος χρόνος RTT θα είναι πάρα πολύ μικρός, παράγοντας έτσι μια πολύ μικρή τιμή για το SRTT και συνεπώς για το RTO. Αυτό οδηγεί πιθανώς σε ένα αποτέλεσμα θετικής ανάδρασης του δικτύου, προκαλώντας επιπρόσθετες αναμεταδόσεις και επιπρόσθετες λανθασμένες μετρήσεις.

Ο αλγόριθμος του Karn επιλύει αυτό το πρόβλημα με τους ακόλουθους κανόνες:

1. Δεν χρησιμοποιείται ο μετρημένος χρόνος RTT για ένα αναμεταδιδόμενο πακέτο, ώστε να ενημερωθεί το SRTT και το SDEV (εξίσωση 6.8).
2. Υπολογίζεται το RTO της οπισθοδρόμησης με χρήση της 6.9, όταν πραγματοποιείται αναμετάδοση.
3. Χρησιμοποιείται η τιμή του RTO της οπισθοδρόμησης για ένα επιτυχώς ληφθέν πακέτο ώστε να φθάσει μια επιβεβαίωση για ένα πακέτο που δεν έχει αναμεταδοθεί.

Όταν ληφθεί μια επιβεβαίωση για ένα μη αναμεταδιδόμενο πακέτο, ο αλγόριθμος Jacobson ενεργοποιείται πάλι για τον υπολογισμό των μελλοντικών τιμών του RTO.

### **6.2.1.7.2 Διαχείριση Παραθύρου**

Μαζί με τις τεχνικές για τη βελτίωση της αποτελεσματικότητας του χρόνου αναμετάδοσης, ένας αριθμός προσεγγίσεων για τη διαχείριση του παραθύρου αποστολής έχει μελετηθεί. Το μέγεθος του παραθύρου αποστολής του TCP μπορεί να έχει κρίσιμη επίδραση στο αν το TCP μπορεί να χρησιμοποιηθεί αποτελεσματικά χωρίς να προκληθεί φαινόμενο συμφόρησης. Θα αναπτυχθούν τέσσερις τεχνικές, οι οποίες βρίσκουν εφαρμογή σε όλες τις μοντέρνες υλοποιήσεις του πρωτοκόλλου TCP:

- Η αργή εκκίνηση (Slow Start).
- Η δυναμική αλλαγή μεγέθους του παραθύρου κατά τη συμφόρηση (Dynamic Window Sizing on Congestion).
- Η γρήγορη αναμετάδοση (Fast Retransmit).

- Η γρήγορη ανάκαμψη (Fast Recovery).

### A. Αργή εκκίνηση

Όσο μεγαλύτερο είναι το παράθυρο αποστολής, τόσο περισσότερα πακέτα μπορούν να σταλούν από μια πηγή TCP προτού αυτή να πρέπει να περιμένει για μια επιβεβαίωση λήψης. Στη συνηθισμένη πορεία των γεγονότων, η ίδια η φύση του TCP, του επιτρέπει να χρονομετρεί τον εαυτό του και επομένως, το TCP μπορεί από μόνο του να ρυθμίζει κατάλληλα το ρυθμό της κυκλοφορίας. Ωστόσο, όταν μια σύνδεση έχει μόλις ενεργοποιηθεί, δεν έχει κάποιο αρχικό ρυθμό για να την κατευθύνει.

Μια στρατηγική που μπορεί να ακολουθηθεί αναφέρει ότι, ο αποστολέας TCP μπορεί να αρχίσει να στέλνει από κάποιο σχετικά μεγάλο παράθυρο, ελπίζοντας να προσεγγίσει το μέγεθος παραθύρου που παρέχεται τελικά και από την ίδια τη σύνδεση. Αυτό είναι πολύ ριψοκίνδυνο επειδή ενδέχεται ο αποστολέας να πλημμυρίσει το διαδίκτυο με πάρα πολλά πακέτα, προτού αυτός συνειδητοποιήσει από τις λήξεις των χρόνων RTO, ότι η ροή είναι υπερβολική.

Ο Jacobson προτείνει μια διαδικασία που είναι γνωστή ως αργή εκκίνηση. Το TCP κάνει χρήση ενός παραθύρου συμφόρησης, του οποίου το μέγεθος μετριέται σε πακέτα και όχι σε οκτάδες (bytes). Σε οποιαδήποτε στιγμή, η μετάδοση του TCP περιορίζεται από την ακόλουθη σχέση:

$$awnd = \text{MIN}[credit, cwnd], \quad (6.10)$$

όπου:

*awnd* = το επιτρεπόμενο παράθυρο, σε πακέτα. Αυτός είναι ο αριθμός των πακέτων που το TCP μπορεί να στείλει χωρίς να περιμένει καμία επιβεβαίωση.

*cwnd* = το παράθυρο συμφόρησης, σε πακέτα. Ένα παράθυρο που χρησιμοποιείται από το TCP κατά τη διάρκεια της εκκίνησης και για να ελαττώνει τη ροή σε περιόδους συμφόρησης.

*credit* = το ποσό, σε πακέτα, της αχρησιμοποίητης πίστωσης που παραχωρείται κατά την πιο πρόσφατη επιβεβαίωση. Όταν μια επιβεβαίωση λαμβάνεται, αυτή η τιμή υπολογίζεται ως *window/segment size*, όπου *window* είναι ένα πεδίο στο εισερχόμενο πακέτο TCP (το ποσό των δεδομένων, που η αντίστοιχη TCP οντότητα παραλήπτης είναι πρόθυμη να δεχθεί).

Όταν ανοίγει μια καινούργια σύνδεση, η οντότητα TCP δίνει αρχική τιμή *cwnd*=1. Αυτό σημαίνει ότι, το TCP επιτρέπεται να στείλει μόνο ένα πακέτο και μετά πρέπει να περιμένει για μια επιβεβαίωση, προτού στείλει ένα δεύτερο πακέτο. Κάθε φορά που λαμβάνεται μια επιβεβαίωση, η τιμή του *cwnd* αυξάνεται κατά ένα, έως ότου φθάσει μια μέγιστη τιμή.

Πρακτικά, ο μηχανισμός της αργής εκκίνησης εξετάζει το διαδίκτυο, για να βεβαιωθεί ότι δε στέλνει πάρα πολλά πακέτα σε ένα ήδη συνωστισμένο περιβάλλον. Καθώς φθάνουν οι επιβεβαιώσεις, το TCP μπορεί να ανοίξει το παράθυρο αποστολής του ώσπου η ροή να ελεγχθεί από τις εισερχόμενες επιβεβαιώσεις παρά από το *cwnd*.

Ο όρος ‘αργή εκκίνηση’ είναι λίγο λανθασμένη ονομασία, γιατί το *cwnd* στην πράξη αυξάνεται εκθετικά. Όταν η πρώτη επιβεβαίωση φθάσει, το TCP αυξάνει το *cwnd* σε δύο και μπορεί να στείλει δύο πακέτα. Όταν αυτά τα δύο πακέτα επιβεβαιωθούν, το TCP μπορεί να αυξήσει το παράθυρο αποστολής του κατά ένα πακέτο για το κάθε εισερχόμενο μήνυμα ACK και μπορεί να αυξήσει το *cwnd* κατά 1 για το κάθε εισερχόμενο ACK. Για το λόγο αυτό, σε αυτό το σημείο το TCP μπορεί να στείλει τέσσερα πακέτα. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς ως κάποια μέγιστη τιμή του *cwnd*.

### Β.Η δυναμική αλλαγή μεγέθους του παραθύρου κατά τη συμφόρηση

Ο αλγόριθμος αργής εκκίνησης δημιουργήθηκε για την αποτελεσματική λειτουργία κατά την αρχικοποίηση μιας σύνδεσης. Καθιστά ικανό τον αποστολέα TCP να προσδιορίσει γρήγορα το λογικό μέγεθος παραθύρου αποστολής για τη σύνδεση. Ίσως να μην είναι χρήσιμη, όμως, η ίδια τεχνική όταν υπάρχει ένας χείμαρρος συμφόρησης. Συγκεκριμένα, ας υποθεθεί ότι μια οντότητα TCP αρχικοποιεί μια σύνδεση και ξεκινάει μια διαδικασία αργής εκκίνησης. Σε κάποιο σημείο, είτε πριν, είτε μετά από όταν το *cwnd* φθάσει το μέγεθος της πίστωσης που προσδιορίστηκε από την άλλη πλευρά, ένα πακέτο χάνεται (επειδή πέρασε ο χρόνος RTO). Αυτό είναι ένα σήμα ότι πραγματοποιείται συμφόρηση στο διαδίκτυο. Δεν είναι ξεκάθαρο το πόσο σοβαρή είναι η συμφόρηση. Για αυτό, μια συντητική διαδικασία θα ήταν να γίνει η τιμή  $cwnd=1$  και να ξεκινήσει μια αργή εκκίνηση παντού.

Αυτή μοιάζει σα μια λογική, συντηρητική διαδικασία, αλλά στην πραγματικότητα δεν είναι και τόσο συντηρητική. Ο Jacobson υποδεικνύει ότι ‘είναι εύκολο να οδηγηθεί ένα δίκτυο σε κορεσμό, αλλά είναι δύσκολο για το δίκτυο να ανακάμψει’. Με άλλα λόγια, όταν συμβεί συμφόρηση, θα χρειαστεί πολύ χρόνο για να ξεπεραστεί η συμφόρηση. Έτσι, η εκθετική αύξηση του *cwnd* κάτω από τη διαδικασία της αργής εκκίνησης μπορεί να είναι πολύ επιθετική και ίσως χειροτερεύσει τη συμφόρηση. Αντίθετα, ο Jacobson πρότεινε τη χρήση της αργής εκκίνησης, αλλά με μια γραμμική αύξηση του *cwnd*. Οι κανόνες είναι οι ακόλουθοι. Όταν πραγματοποιείται μια λήξη του χρόνου RTO, τότε:

- 1 Τίθεται ένα όριο αργής εκκίνησης ίσο με το μισό του τρέχοντος παραθύρου συμφόρησης. Αυτό σημαίνει ότι, τίθεται  $ssthresh = cwnd/2$ .
- 2 Τίθεται  $cwnd=1$  και εκτελείται μια διαδικασία αργής εκκίνησης ώσπου  $cwnd = ssthresh$ . Σε αυτή τη φάση, το *cwnd* αυξάνεται κατά 1 για κάθε λαμβανόμενη επιβεβαίωση.
- 3 Για  $cwnd \geq ssthresh$ , αυξάνεται το *cwnd* κατά 1 για κάθε χρόνο κυκλικού ταξιδιού (RTT – Round Trip Time).

### Γ. Η γρήγορη αναμετάδοση

Ο χρόνος αναμετάδοσης, ο οποίος χρησιμοποιείται από έναν αποστολέα TCP για να προσδιορίσει πότε θα αναμεταδώσει το πακέτο, θα είναι γενικά αξιοσημείωτα μεγαλύτερος από τον πραγματικό χρόνο κυκλικού ταξιδιού (RTT) που θα πάρει στο σήμα ACK το πακέτου ώστε να φθάσει στον αποστολέα. Τόσο ο αρχικός αλγόριθμος του

RFC 793, όσο και ο αλγόριθμος του Jacobson, θέτουν τη τιμή του RTO λίγο μεγαλύτερη από τον εκτιμώμενο χρόνο κυκλικού ταξιδιού SRTT. Υπάρχουν διάφοροι παράγοντες που κάνουν επιθυμητό αυτό το περιθώριο:

- 1 Το RTO υπολογίζεται με βάση μια πρόβλεψη του επόμενου RTT, που εκτιμάται από προηγούμενες τιμές του RTT. Αν οι καθυστερήσεις στο δίκτυο διακυμαίνονται, τότε ο εκτιμηθείς χρόνος RTT μπορεί να είναι μικρότερος από τον πραγματικό RTT.
- 2 Παρομοίως, αν οι καθυστερήσεις στον προορισμό διακυμαίνονται, ο εκτιμηθείς χρόνος RTT γίνεται αναξιόπιστος.
- 3 Το σύστημα του προορισμού μπορεί να μην επιβεβαιώνει το κάθε πακέτο, αλλά μπορεί να επιβεβαιώνει συσσωρευτικά πολλαπλά πακέτα, ενώ ταυτοχρόνως, στέλνει επιβεβαιώσεις όταν υπάρχουν δεδομένα προς αποστολή. Αυτή η συμπεριφορά συνεισφέρει στη διακύμανση του RTT.

Μια συνέπεια αυτών των παραγόντων είναι ότι αν ένα πακέτο χαθεί, το TCP μπορεί να είναι αργό στην αναμετάδοση. Αν ο παραλήπτης TCP χρησιμοποιεί μια πολιτική ενδοδιάταξης κατά τη λήψη, τότε πολλά πακέτα ίσως να χαθούν. Ακόμα και στην περισσότερο πιθανή περίπτωση όπου ο παραλήπτης TCP χρησιμοποιεί μια πολιτική λήψης εσωτερικού παραθύρου (in-window), μια αργή αναμετάδοση μπορεί να προκαλέσει προβλήματα. Για να γίνει αυτό αντιληπτό, ας υποθεθεί ότι το TCP A μεταδίδει μια ακολουθία πακέτων, το πρώτο από τα οποία χάνεται. Όσο το παράθυρο αποστολής του είναι γεμάτο και ο χρόνος RTO δεν έχει περάσει, ο A μπορεί να συνεχίσει να μεταδίδει χωρίς να λάβει κάποια επιβεβαίωση. Ο παραλήπτης B δέχεται όλα τα πακέτα εκτός του πρώτου. Αλλά, ο B πρέπει να ενταμιεύσει όλα αυτά τα εισερχόμενα πακέτα ώσπου αυτό που λείπει να αναμεταδοθεί. Έτσι, δεν μπορεί να καθαρίσει τον ενταμιευτή του στέλλοντας τα δεδομένα του σε μια εφαρμογή, ώσπου να λάβει το πακέτο που λείπει. Αν η αναμετάδοση του πακέτου που λείπει καθυστερήσει πάρα πολύ, ο B θα πρέπει να απορρίπτει εισερχόμενα πακέτα.

Ο Jacobson πρότεινε δύο τεχνικές, τη γρήγορη αναμετάδοση και τη γρήγορη ανάκαμψη, που κάτω από ορισμένες περιστάσεις βελτιώνουν τη λειτουργία που παρέχεται από το RTO. Η γρήγορη αναμετάδοση επωφελείται από τον ακόλουθο κανόνα του πρωτοκόλλου TCP. Αν μια οντότητα TCP λάβει ένα πακέτο εκτός σειράς, πρέπει αμέσως να εκπέμψει μια επιβεβαίωση για το τελευταίο ενδοδιατεταγμένο πακέτο που λήφθηκε. Το TCP θα συνεχίσει να εκπέμψει αυτή την επιβεβαίωση ώσπου το χαμένο πακέτο να φθάσει και να καλύψει την τρύπα που δημιουργήθηκε στον ενταμιευτή του παραλήπτη. Όταν αυτή η τρύπα καλυφθεί, το TCP στέλνει ένα συσσωρευτικό σήμα ACK για όλα τα ενδοδιατεταγμένα πακέτα που έλαβε ως το συγκεκριμένο χρονικό σημείο.

Όταν η πηγή TCP λάβει ένα αντίγραφο σήμα ACK, αυτό σημαίνει ότι, είτε το πακέτο που ακολουθεί το επιβεβαιωμένο πακέτο καθυστέρησε έτσι ώστε τελικά έφθασε εκτός σειράς, είτε το πακέτο έχει χαθεί. Στην πρώτη περίπτωση, το πακέτο τελικά φθάνει και για αυτό το TCP δεν πρέπει να αναμεταδώσει. Αλλά, στη δεύτερη περίπτωση, η άφιξη μιας διπλής επιβεβαίωσης μπορεί να λειτουργήσει ως η πρώτη προειδοποίηση που θα πει στην πηγή TCP ότι ένα πακέτο έχει χαθεί και πρέπει να αναμεταδοθεί. Για να σιγουρευτεί ότι πρόκειται για την περίπτωση δύο και όχι για την πρώτη, ο Jacobson πρότεινε ότι ο αποστολέας TCP πρέπει να περιμένει ώσπου να λάβει 3 αντίγραφα



σήματα ACK για το ίδιο πακέτο (δηλαδή, στο σύνολο τα σήματα ACK για το ίδιο πακέτο είναι 4). Κάτω από αυτές τις συνθήκες, είναι υψηλή η πιθανότητα το πακέτο να έχει χαθεί, οπότε πρέπει αμέσως να αναμεταδοθεί, χωρίς ο αποστολέας να περιμένει το πέρας του χρόνου RTO.

#### Δ. Η γρήγορη Ανάκαμψη

Όταν μια οντότητα TCP αναμεταδίδει ένα πακέτο χρησιμοποιώντας τη τεχνική της γρήγορης αναμετάδοσης, γνωρίζει (ή καλύτερα υποθέτει) ότι ένα πακέτο χάθηκε, αν και δεν έχει περάσει ο χρόνος RTO για το πακέτο αυτό. Άρα πρέπει να λάβει μέτρα αποφυγής συμφόρησης. Μια φανερή στρατηγική είναι η τεχνική της αργής εκκίνησης / διαδικασία αποφυγής συμφόρησης που χρησιμοποιείται με συμβάντα λήξης του χρόνου RTO. Αυτό σημαίνει ότι, η οντότητα μπορεί να θέσει *ssthresh* ίσο με *cwnd/2*, να θέσει *cwnd* = 1 και να ξεκινήσει με μια διαδικασία εκθετικής αργής εκκίνησης ώσπου *cwnd* = *ssthresh*, και έπειτα να αυξήσει γραμμικά το *cwnd*. Ο Jacobson διαφώνησε στο ότι αυτή η προσέγγιση είναι μη απαραίτητα συντηρητική. Όπως υποδείχθηκε, το γεγονός ότι πολλαπλά σήματα ACK έχουν γυρίσει, δηλώνει ότι τα πακέτα δεδομένων περνάνε ομαλά στην άλλη πλευρά του δικτύου. Έτσι, ο Jacobson πρότεινε τη τεχνική της γρήγορης ανάκαμψης: Να γίνει αναμετάδοση του χαμένου πακέτου, να κοπεί στο μισό το *cwnd*, και έπειτα να πραγματοποιηθεί γραμμική αύξηση του *cwnd*. Αυτή η τεχνική αποφεύγει την αρχική διαδικασία της εκθετικής αργής εκκίνησης.

Με περισσότερη ακρίβεια, η τεχνική της γρήγορης ανάκαμψης έχει ως εξής:

- 1 Όταν ένα τρίτο αντίγραφο ACK φθάσει,
  - a. Τίθεται  $ssthresh = cwnd/2$ .
  - b. Αναμεταδίδεται το χαμένο πακέτο.
  - c. Τίθεται  $cwnd = ssthresh + 3$ .Ο λόγος της πρόσθεσης του 3 στο *ssthresh* είναι ότι αυτό υπολογίζει τον αριθμό των πακέτων που έχουν αφήσει το δίκτυο και που η άλλη άκρη έχει κρύψει.
- 2 Κάθε φορά που ένα επιπλέον αντίγραφο ACK (για το ίδιο πακέτο) φθάνει, αυξάνεται το *cwnd* κατά 1 και μεταδίδεται ένα πακέτο αν αυτό είναι δυνατό. Αυτό λογαριάζεται για το επιπρόσθετο πακέτο που έχει αφήσει το δίκτυο και πυροδοτεί το αντίγραφο ACK.
- 3 Όταν το επόμενο σήμα ACK φθάνει, το οποίο επιβεβαιώνει καινούργια δεδομένα (δηλαδή, μια συσσωρευτική επιβεβαίωση του χαμένου πακέτου συν άλλα επόμενα πακέτα), τίθεται  $cwnd = ssthresh$ .

### **6.3 ΥΠΗΡΕΣΙΕΣ ΜΗ ΣΥΝΔΕΣΙΜΙΚΟΥ ΧΑΡΑΚΤΗΡΑ (CONNECTIONLESS SERVICES)**

Στις υπηρεσίες ασυνδεδεσμένου χαρακτήρα, ο αποστολέας ξεκινάει την αποστολή πακέτων χωρίς να έχει πραγματοποιηθεί κάποιου είδους χαιρετισμός από πριν. Αυτό συμβαίνει σε περιπτώσεις όπου απαιτείται η ελάχιστη δυνατή καθυστέρηση αποστολής, όπως είναι οι αποστολή πολυμέσων συνεχούς συρμού (Streaming Multimedia). Επιπλέον, δεν μπορεί να εξασφαλιστεί καμία ποιότητα υπηρεσίας (Quality of Service), ούτε έλεγχος ροής και συμφόρησης.

### **6.3.1 ΤΟ ΠΡΩΤΟΚΟΛΛΟ ΑΥΤΟΔΥΝΑΜΩΝ ΠΑΚΕΤΩΝ ΧΡΗΣΤΩΝ (USER DATAGRAM PROTOCOL)**

Το UDP είναι το πρωτόκολλο που προσφέρει ασυνδεδεσμένες υπηρεσίες στο επίπεδο μεταφοράς, στο Διαδίκτυο. Συγκριτικά με το TCP, είναι ένα πολύ απλό στη λειτουργία του και τελείως αναξιόπιστο πρωτόκολλο. Χρησιμοποιώντας το UDP, δεν χρειάζεται να δημιουργηθεί σύνδεση μεταξύ δύο υπολογιστών πριν την ανταλλαγή δεδομένων και δεν υπάρχει μηχανισμός για την επιβεβαίωση ότι τα δεδομένα λήφθηκαν. Δηλαδή, δεν υπάρχει τρόπος ανάκαμψης από περιπτώσεις απώλειας πακέτων.

Η μονάδα δεδομένων που χρησιμοποιεί το UDP καλείται αυτοδύναμο πακέτο UDP (UDP datagram). Η επικεφαλίδα του αυτοδύναμου πακέτου αποτελείται από 4 πεδία. Αυτά τα πεδία είναι: ένα πεδίο ολικού μήκους (Total Length), ένα πεδίο αθροίσματος ελέγχου (Checksum) για ανίχνευση λαθών, και οι αριθμοί θυρών για την πηγή και τον προορισμό (Source / Destination Port).

Το πρωτόκολλο UDP βρίσκει χρησιμοποιείται σε περιπτώσεις πολυμέσων, αλλά και για να υποστηρίξει το πρωτόκολλο εφαρμογής DNS, στην αντιστοίχιση των διευθύνσεων IP σε ονόματα κεντρικών υπολογιστών. Γενικά, χρησιμοποιείται οποτεδήποτε οι καθυστερήσεις πρέπει να είναι οι ελάχιστες δυνατές. Αυτό συμβαίνει κυρίως στις επονομαζόμενες **εφαρμογές πραγματικού χρόνου (real time applications)**, όπως είναι η εφαρμογή DNS και οι περισσότερες εφαρμογές πολυμέσων.

---



**Εικόνα 6.7: Αυτοδύναμο Πακέτο UDP**

---