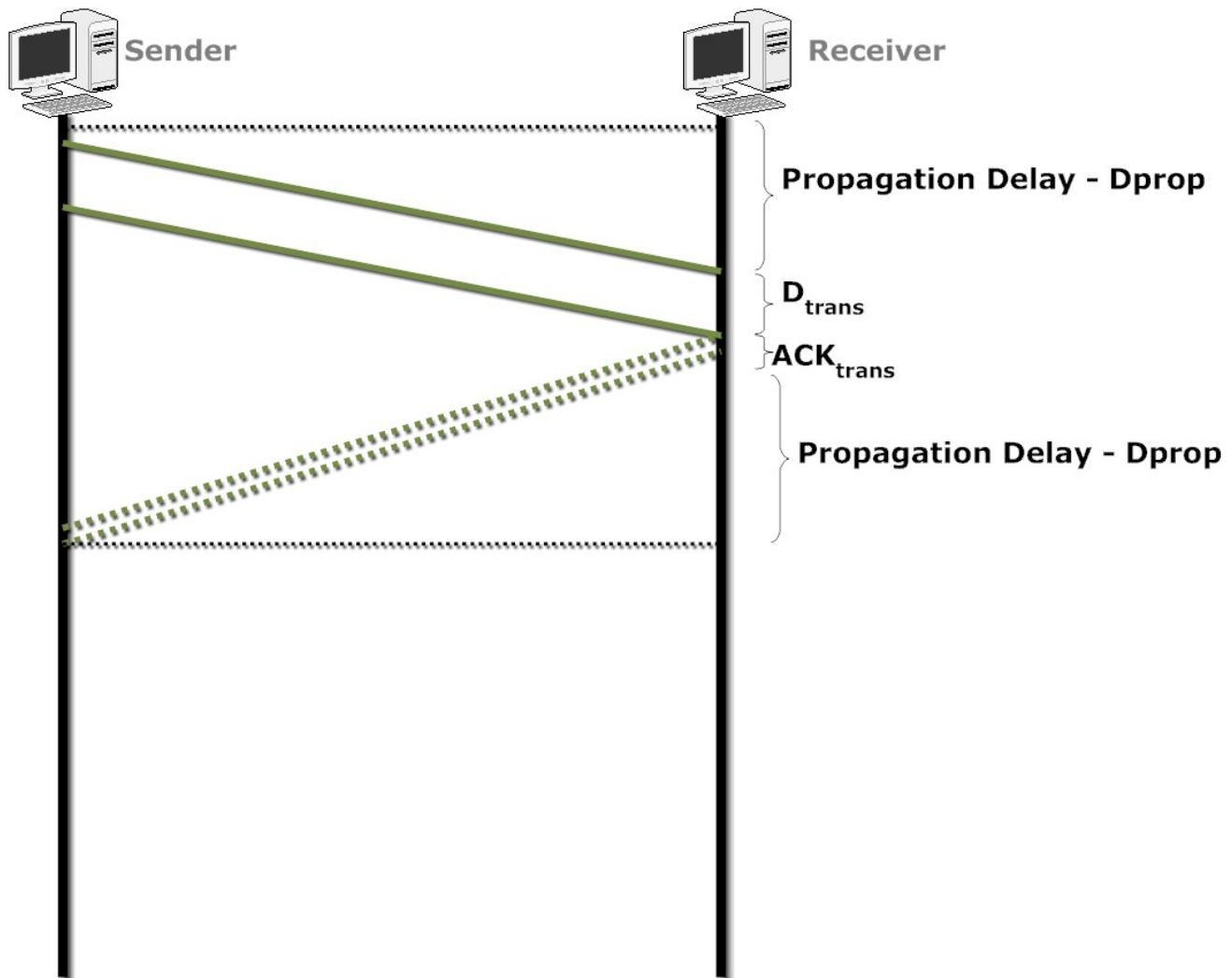


I. Παράδειγμα 1: Απόδοση TCP με παράθυρο αποστολέα = 1

- Ο μηχανισμός όπως έχει περιγραφεί ως τώρα στέλνει μόνο ένα πακέτο και σταματάει να μεταδίδει έως ότου πάρει το ack του πακέτου αυτού (λειτουργία stop-and-wait). Αυτός ο τρόπος όμως δεν είναι αποδοτικός.
- Προτού παρουσιαστεί παράδειγμα παρουσίαση κλασικού διαγράμματος για TCP και εξήγηση propagation delay και transmission delay.



Σχήμα 1: TCP αποστολέας με παράθυρο αποστολέα = 1

Propagation Delay – Καθυστέρηση διάδοσης σήματος στο φυσικό μέσο – χρόνος που απαιτείται για να φθάσει στον παραλήπτη το πρώτο bit της μετάδοσης.

Transmission Delay (D_{trans}) – Χρόνος που χρειάζεται για να προωθηθούν όλα τα bits του πακέτου στη ζεύξη. Εξαρτάται από το ρυθμό μετάδοσης της ζεύξης και από το μέγεθος του πακέτου.

ACK_{trans} – Χρόνος που χρειάζεται για να μεταδοθεί το ACK στο μέσο.

Throughput (ρυθμαπόδοση) – Ρυθμός με τον οποίο ο παραλήπτης μίας ζεύξης λαμβάνει δεδομένα. Μπορεί να υπολογιστεί γενικά ως τον λόγο του συνολικού αριθμού από bytes που λήφθηκαν από ένα κόμβο δια το συνολικό χρόνο που χρειάστηκε για να τα παραλάβει.

Utilization (Χρησιμοποίηση του μέσου μετάδοσης): Κλάσμα του χρόνου που ο αποστολέας μετέδιδε δεδομένα στο κανάλι σε σχέση με το συνολικό χρόνο που χρειάζεται για να ολοκληρωθεί η μετάδοση

- c. Σενάριο: δύο τερματικά κατά μήκος ενός μονοπατιού στο internet. Υπόθεση για μηδενικό queuing και processing delay καθώς και για $ACK_{trans} = 0$. Έστω επίσης μέγεθος πακέτου $L = 1000\text{bytes}$, χωρητικότητα μονοπατιού $R = 1\text{Gbps}$ και καθυστέρηση διάδοσης $D_{prop} = 15\text{msec}$.

Ο χρόνος για να μεταδοθεί το πακέτο στο μέσο είναι: $D_{trans} = \frac{L}{R} = \frac{1000 * 8}{10^9} = 0,008\text{msec}$.

Ο συνολικός χρόνος που χρειάζεται για να ολοκληρωθεί η μετάδοση του πακέτου είναι:

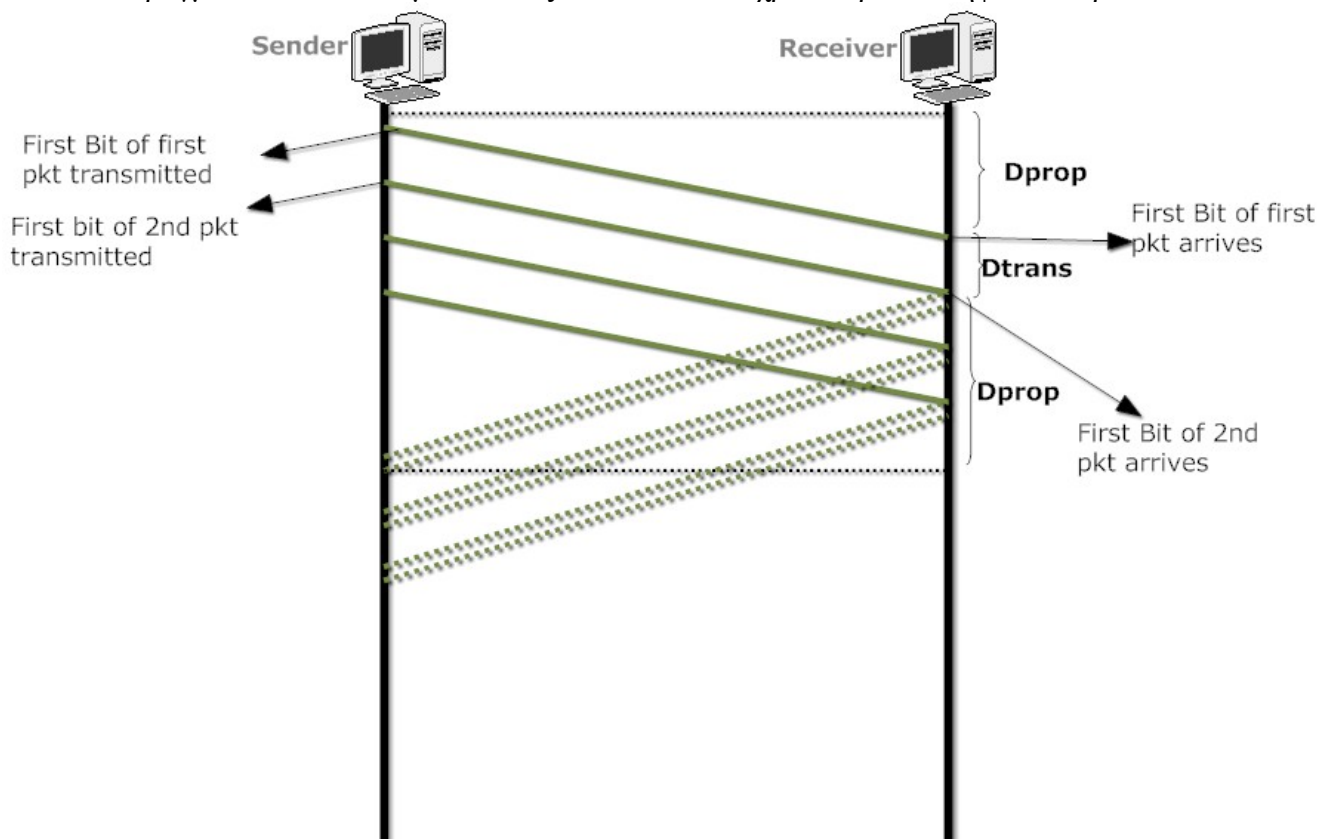
$$D_{trans} + ACK_{trans} + 2 * D_{prop} \stackrel{ACK_{trans}=0}{=} 30,008$$

Συνεπώς, το utilization του καναλιού είναι $U = \frac{0,008}{30,008} = 0,00027$.

Συνολικά το throughput που επιτυγχάνεται περνώντας από ένα μονοπάτι με χωρητικότητα 1Gbps είναι 267Kbps.

II. Παράδειγμα 2: Κίνητρο για σωλήνωση μεταδόσεων – παράδειγμα με περισσότερες μεταδόσεις χωρίς αναμονή για επιβεβαίωση

Έστω ότι πραγματοποιούνται 3 μεταδόσεις πακέτων ταυτόχρονα προτού ληφθεί το πρώτο ACK.



Σχήμα 2: Αποστολέας TCP με παράθυρο μετάδοσης = 3

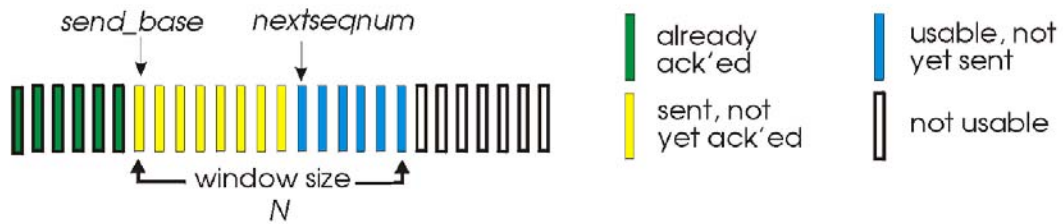
Τώρα σε συνολικό χρόνο $D_{trans} + ACK_{trans} + 2 * D_{prop} \stackrel{ACK_{trans}=0}{=} 30,008$ γίνεται η μετάδοση 3 πακέτων αντί για 1 άρα το utilization κατά παρόμοιο τρόπο θα είναι:

$$U = \frac{3 * 0,008}{30,008} = 3 * 0,00027 \text{ \u03c1\u03b1 \u03b5\u03c7\u03b5\u03b9 \u03c4\u03c1\u03b9\u03c0\u03bb\u03b1\u03c3\u03b9\u03b1\u03c3\u03c4\u03b5\u03b9.}$$

\u039c\u03b1 \u03c4\u03c1\u03b9\u03b1 \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03c4\u03b1 \u03cc\u03c0\u03b9\u03b1 \u03bc\u03b5\u03c4\u03b1\u03b4\u03b9\u03b4\u03bf\u03bd\u03c4\u03b1\u03b9 \u03c7\u03c9\u03c1\u03b9\u03c2 \u03b1\u03bd\u03b1\u03bc\u03bf\u03bd\u03b7 \u03b5\u03c9\u03c2 \u03cc\u03c4\u03bf \u03bb\u03b7\u03c6\u03b8\u03b5\u03b9 \u03c4\u03bf \u03c0\u03c1\u03cc\u03c4\u03bf ACK \u03bb\u03b5\u03bc\u03b5 \u03cc\u03c4\u03b9 \u03bc\u03b5\u03c4\u03b1\u03b4\u03b9\u03b4\u03bf\u03bd\u03c4\u03b1\u03b9 \u03bc\u03b5 \u03c4\u03b7 \u03c7\u03c1\u03b7\u03c3\u03b7 \u03c3\u03c9\u03bb\u03b7\u03bd\u03c9\u03c3\u03b7\u03c2 \u03b7 \u03b1\u03bb\u03bb\u03b9\u03c9\u03c2 \u03cc\u03c4\u03b9 \u03b5\u03c7\u03bf\u03bc\u03b5 pipelined transmissions.

III. \u0395\u03c0\u03b5\u03ba\u03c4\u03b1\u03c3\u03b7 \u03b1\u03be\u03b9\u03cc\u03c0\u03b9\u03c3\u03c4\u03b9\u03b1\u03c2 \u03b3\u03b9\u03b1 TCP \u03bc\u03b5 \u03c0\u03b1\u03c1\u03ac\u03b8\u03c5\u03c1\u03bf \u03b1\u03c0\u03cc\u03c3\u03c4\u03bf\u03bb\u03b5\u03b1 > 1

- \u039c\u03b5 \u03c4\u03b7\u03bd \u03c0\u03c1\u03bf\u03b3\u03bf\u03c5\u03bc\u03b5\u03bd\u03b7 \u03c4\u03b5\u03c7\u03bd\u03b9\u03ba\u03b7 \u03b5\u03b9\u03bd\u03b1 \u03c3\u03b1\u03bd \u03bd\u03b1 \u03b5\u03c7\u03bf\u03bc\u03b5 \u03b1\u03c5\u03be\u03b9\u03c3\u03b5\u03b9 \u03c4\u03bf \u03c0\u03bb\u03ac\u03c4\u03bf\u03c2 \u03c4\u03b7\u03c2 \u03c3\u03c9\u03bb\u03b7\u03bd\u03c9\u03c3\u03b7\u03c2 \u03bc\u03b5\u03c3\u03b1 \u03b1\u03c0\u03cc \u03c4\u03b7\u03bd \u03cc\u03c0\u03b9\u03b1 \u03b4\u03b9\u03b5\u03c1\u03c7\u03bf\u03bd\u03c4\u03b1\u03b9 \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03b3\u03b9\u03b1\u03bd\u03c4\u03bf \u03ba\u03b1\u03b9 \u03b1\u03c5\u03c4\u03b7 \u03b7 \u03c4\u03b5\u03c7\u03bd\u03b9\u03ba\u03b7 \u03cc\u03bd\u03bf\u03bc\u03ac\u03b6\u03b5\u03c4\u03b1\u03b9 \u03ba\u03b1\u03b9 \u03c3\u03c9\u03bb\u03b7\u03bd\u03c9\u03c3\u03b7 \u03bc\u03b5\u03c4\u03b1\u03b4\u03cc\u03c3\u03b5\u03c9\u03bd.
- \u039e\u03b9 pipelined \u03bc\u03b5\u03c4\u03b1\u03b4\u03cc\u03c3\u03b5\u03b9\u03c2 \u03c0\u03b5\u03c1\u03b9\u03c0\u03bb\u03b5\u03ba\u03bf\u03bd \u03c4\u03bf \u03c7\u03b5\u03c1\u03b9\u03c3\u03bc\u03cc \u03c4\u03c9\u03bd \u03b1\u03c0\u03c9\u03bb\u03b5\u03b9\u03c9\u03bd \u03c0\u03b1\u03ba\u03b5\u03c4\u03c9\u03bd \u03ba\u03b1\u03b8\u03c9\u03c2 \u03c4\u03c9\u03c1\u03b1 \u03bc\u03c0\u03bf\u03c1\u03b5\u03b9 \u03bd\u03b1 \u03b5\u03c7\u03b5\u03b9 \u03c7\u03b1\u03b8\u03b5\u03b9 \u03cc\u03c0\u03b9\u03cc\u03b4\u03b7\u03c0\u03bf\u03c4\u03b5 \u03ba\u03b1\u03b9 \u03cc\u03c3\u03b1\u03b4\u03b7\u03c0\u03bf\u03c4\u03b5 \u03b1\u03c0\u03cc \u03cc\u03c3\u03b1 \u03c3\u03c4\u03b5\u03bb\u03bd\u03bf\u03bd\u03c4\u03b1\u03b9 \u03c4\u03b1\u03c5\u03c4\u03cc\u03c7\u03c1\u03bf\u03bd\u03b1.
- \u0391\u03c0\u03b1\u03b9\u03c4\u03bf\u03bd\u03c4\u03b1\u03b9 \u03bb\u03bf\u03b9\u03c0\u03cc\u03bd \u03bc\u03b7\u03c7\u03b1\u03bd\u03b9\u03c3\u03bc\u03cc\u03b9 \u03cc\u03b9 \u03cc\u03c0\u03b9\u03cc\u03b9 \u03b8\u03b1 \u03c5\u03c0\u03cc\u03c3\u03c4\u03b7\u03c1\u03b9\u03b6\u03bf\u03bd \u03b1)\u03c0\u03b1\u03c1\u03ac\u03bb\u03b7\u03bb\u03b7\u03c3 \u03bc\u03b5\u03c4\u03b1\u03b4\u03cc\u03c3\u03b5\u03b9\u03c2 \u03c0\u03b1\u03ba\u03b5\u03c4\u03c9\u03bd \u03b2)\u03b4\u03b9\u03b1\u03c7\u03b5\u03b9\u03c1\u03b9\u03c3\u03b7 \u03c3\u03c6\u03b1\u03bb\u03bc\u03ac\u03c4\u03c9\u03bd \u03c0\u03b1\u03ba\u03b5\u03c4\u03c9\u03bd \u03b7 \u03b5\u03c0\u03b9\u03b2\u03b5\u03b2\u03b1\u03b9\u03c9\u03c3\u03b5\u03c9\u03bd \u03b5\u03bd\u03c4\u03cc\u03c2 \u03b5\u03bd\u03cc\u03c2 \u03c0\u03b1\u03c1\u03b1\u03b8\u03c5\u03c1\u03bf \u03bc\u03b5\u03c4\u03b1\u03b4\u03cc\u03c3\u03b5\u03c9\u03bd \u03b3)\u03c0\u03b1\u03c1\u03ac\u03b4\u03cc\u03c3\u03b7 \u03c0\u03b1\u03ba\u03b5\u03c4\u03c9\u03bd \u03c3\u03c4\u03b7 \u03c3\u03c9\u03c3\u03c4\u03b7 \u03c3\u03b5\u03b9\u03c1\u03ac \u03c3\u03c4\u03bf \u03b5\u03c0\u03b9\u03c0\u03b5\u03b4\u03bf \u03b5\u03c6\u03b1\u03c1\u03bc\u03bf\u03b3\u03b7\u03c2 \u03c3\u03c4\u03b7\u03bd \u03c0\u03bb\u03b5\u03c5\u03c1\u03ac \u03c4\u03bf \u03c0\u03b1\u03c1\u03b1\u03bb\u03b7\u03c0\u03c4\u03b7 (in-order packet delivery) \u03b4)\u03b5\u03c0\u03b9\u03b2\u03b5\u03b2\u03b1\u03b9\u03c9\u03c3\u03b5\u03b9\u03c2 \u03b3\u03b9\u03b1 \u03c4\u03b7 \u03c3\u03b7\u03bc\u03b1\u03bd\u03c3\u03b7 \u03c3\u03c9\u03c3\u03c4\u03b7\u03c2 \u03bb\u03b7\u03c8\u03b7\u03c2 \u03c0\u03b1\u03ba\u03b5\u03c4\u03c9\u03bd.
- \u0391\u03c2 \u03c5\u03c0\u03cc\u03c4\u03b5\u03b8\u03b5\u03b9 \u03cc\u03c4\u03b9 \u03cc \u03b1\u03c0\u03cc\u03c3\u03c4\u03bf\u03bb\u03b5\u03b1\u03c2 \u03c7\u03c1\u03b7\u03c3\u03b9\u03bc\u03c0\u03bf\u03b9\u03b5\u03b9 \u03b5\u03bd\u03b1 \u03c0\u03b1\u03c1\u03ac\u03b8\u03c5\u03c1\u03bf \u03bc\u03b5\u03b3\u03b5\u03b8\u03bf\u03c5\u03c2 N. \u039c\u03bf N \u03bc\u03b5 \u03b1\u03bb\u03bb\u03b1 \u03bb\u03cc\u03b3\u03b9\u03b1 \u03c0\u03b1\u03c1\u03b9\u03c3\u03c4\u03b1\u03bd\u03b5\u03b9 \u03c4\u03bf\u03bd \u03b1\u03c1\u03b9\u03b8\u03bc\u03cc \u03c4\u03c9\u03bd \u03c0\u03b1\u03ba\u03b5\u03c4\u03c9\u03bd (bytes) \u03c4\u03b1 \u03cc\u03c0\u03b9\u03b1 \u03b5\u03c7\u03bf\u03bd \u03bc\u03b5\u03c4\u03b1\u03b4\u03cc\u03b8\u03b5\u03b9 \u03ba\u03b1\u03b9 \u03b4\u03b5\u03bd \u03b5\u03c7\u03bf\u03bd \u03b5\u03c0\u03b9\u03b2\u03b5\u03b2\u03b1\u03b9\u03c9\u03b8\u03b5\u03b9. \u039e\u03b9 \u03b1\u03bb\u03bb\u03b1\u03b3\u03b5\u03c2 \u03c0\u03bf\u03c5 \u03b1\u03c0\u03b1\u03b9\u03c4\u03bf\u03bd\u03c4\u03b1\u03b9 \u03c3\u03c4\u03bf TCP \u03c9\u03c3\u03c4\u03b5 \u03bd\u03b1 \u03b9\u03ba\u03b1\u03bd\u03cc\u03c0\u03bf\u03b9\u03b5\u03b9 \u03c4\u03b9\u03c2 \u03c0\u03b1\u03c1\u03b1\u03c0\u03b1\u03bd\u03c9 \u03b1\u03c0\u03b1\u03b9\u03c4\u03b7\u03c3\u03b5\u03b9\u03c2 \u03b5\u03b9\u03bd\u03b1:
 - o \u0391\u03cd\u03be\u03b9\u03c3\u03b7 \u03c4\u03bf\u03c5 \u03b5\u03c5\u03c1\u03bf\u03c5\u03c2 \u03c4\u03c9\u03bd \u03b1\u03c1\u03b9\u03b8\u03bc\u03cc\u03bd \u03b1\u03ba\u03cc\u03bb\u03bf\u03b8\u03b9\u03b1\u03c2
 - o \u039c\u03c7\u03c1\u03b7\u03c3\u03b7 \u03b5\u03bd\u03c4\u03b1\u03bc\u03b5\u03c5\u03c4\u03c9\u03bd \u03c3\u03c4\u03bf\u03bd \u03b1\u03c0\u03cc\u03c3\u03c4\u03bf\u03bb\u03b5\u03b1 \u03ba\u03b1\u03b9 \u03c3\u03c4\u03bf\u03bd \u03c0\u03b1\u03c1\u03b1\u03bb\u03b7\u03c0\u03c4\u03b7. \u0393\u03b9\u03b1\u03c4\u03b9? \u039c\u03b9 \u03bc\u03b5\u03b3\u03b5\u03b8\u03bf\u03c2 \u03b5\u03c7\u03bf\u03bd? \u0395\u03c6\u03cc\u03c3\u03bf\u03bd \u03cc \u03b1\u03c0\u03cc\u03c3\u03c4\u03bf\u03bb\u03b5\u03b1\u03c2 \u03bc\u03c0\u03bf\u03c1\u03b5\u03b9 \u03bd\u03b1 \u03c3\u03c4\u03b5\u03b9\u03bb\u03b5\u03b9 \u03c0. \u03c7. \u03c4\u03bf \u03c0\u03cc\u03bb\u03c5 N \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03c4\u03b1\u03c5\u03c4\u03cc\u03c7\u03c1\u03bf\u03bd\u03b1 \u03c7\u03c9\u03c1\u03b9\u03c2 \u03bd\u03b1 \u03c0\u03b1\u03c1\u03b5\u03b9 \u03b5\u03c0\u03b9\u03b2\u03b5\u03b2\u03b1\u03b9\u03c9\u03c3\u03b7, \u03b8\u03b1 \u03c0\u03c1\u03b5\u03c0\u03b5\u03b9 \u03bd\u03b1 \u03ba\u03c1\u03b1\u03c4\u03b1\u03b5\u03b9 \u03b1\u03c5\u03c4\u03b1 \u03c4\u03b1 N \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03ba\u03ac\u03c0\u03bf\u03c5 \u03b1\u03c0\u03b8\u03b7\u03ba\u03b5\u03c5\u03bc\u03b5\u03bd\u03b1 \u03b5\u03c9\u03c2 \u03cc\u03c4\u03bf \u03b5\u03c1\u03b8\u03b5\u03b9 \u03b5\u03c0\u03b9\u03b2\u03b5\u03b2\u03b1\u03b9\u03c9\u03c3\u03b7 \u03b3\u03b9\u03b1\u03c4\u03b9 \u03bc\u03c0\u03bf\u03c1\u03b5\u03b9 \u03bd\u03b1 \u03c7\u03c1\u03b5\u03b9\u03b1\u03c3\u03c4\u03b5\u03b9 \u03bd\u03b1 \u03c4\u03b1 \u03be\u03bd\u03b1\u03c3\u03c4\u03b5\u03b9\u03bb\u03b5\u03b9 (sender buffer). \u0391\u03c5\u03c4\u03b1 \u03c4\u03b1 N \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03c3\u03c4\u03bf\u03bd \u03c0\u03b1\u03c1\u03b1\u03bb\u03b7\u03c0\u03c4\u03b7 \u03bc\u03c0\u03bf\u03c1\u03b5\u03b9 \u03bd\u03b1 \u03bc\u03b7 \u03c6\u03c4\u03ac\u03c3\u03bf\u03bd \u03c3\u03c4\u03b7 \u03c3\u03b5\u03b9\u03c1\u03ac. \u039c\u03c0\u03bf\u03c1\u03b5\u03b9 \u03c0\u03c7 \u03bd\u03b1 \u03b5\u03c1\u03b8\u03b5\u03b9 \u03c4\u03bf 4\u03cc \u03c0\u03b1\u03ba\u03b5\u03c4\u03bf, \u03bc\u03b5\u03c4\u03ac \u03c4\u03bf 7\u03cc \u03c4\u03bf 1\u03cc \u03ba.o.k. \u039c\u03bf \u03c0\u03c1\u03c9\u03c4\u03cc\u03ba\u03bf\u03bb\u03bf TCP \u03c0\u03b1\u03c1\u03b5\u03c7\u03b5\u03b9 \u03c5\u03c0\u03b7\u03c1\u03b5\u03c3\u03b9\u03b1 in-order-delivery \u03c3\u03c4\u03bf \u03c5\u03c8\u03b7\u03bb\u03cc\u03c4\u03b5\u03c1\u03bf \u03b5\u03c0\u03b9\u03c0\u03b5\u03b4\u03bf \u03c4\u03c9 \u03b4\u03b9\u03ba\u03c4\u03b9\u03c5. \u0391\u03c5\u03c4\u03bf \u03c3\u03b7\u03bc\u03b1\u03bd\u03b5\u03b9 \u03cc\u03c4\u03b9 \u03c3\u03c4\u03bf \u03b5\u03c0\u03b9\u03c0\u03b5\u03b4\u03bf \u03c0\u03bf\u03c5 \u03b5\u03b9\u03bd\u03b1 \u03b1\u03bc\u03b5\u03c3\u03c9\u03c2 \u03bc\u03b5\u03c4\u03ac \u03b1\u03c0\u03cc \u03b1\u03c5\u03c4\u03bf \u03b8\u03b1 \u03c0\u03b1\u03c1\u03b1\u03b4\u03b9\u03b4\u03b5\u03b9 \u03c4\u03b1 \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03bc\u03b5 \u03c4\u03b7 \u03c3\u03c9\u03c3\u03c4\u03b7 \u03c3\u03b5\u03b9\u03c1\u03ac. \u039c\u03cc\u03c3\u03bf \u03bb\u03bf\u03b9\u03c0\u03cc\u03bd \u03c0\u03b1\u03c1\u03b1\u03bb\u03b1\u03bc\u03b2\u03b1\u03bd\u03b5\u03b9 \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03c7\u03c1\u03b5\u03b9\u03ac\u03b6\u03b5\u03c4\u03b1\u03b9 \u03bd\u03b1 \u03c4\u03b1 \u03b1\u03c0\u03b8\u03b7\u03ba\u03b5\u03c5\u03b5\u03b9 \u03ba\u03ac\u03c0\u03bf\u03c5 \u03b5\u03c9\u03c2 \u03cc\u03c4\u03bf \u03b5\u03c1\u03b8\u03bf\u03bd \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03c0\u03bf\u03c5 \u03ba\u03b1\u03bb\u03c5\u03c0\u03c4\u03bf\u03bd \u03c0\u03b9\u03b8\u03b1\u03bd\u03ac \u03ba\u03b5\u03bd\u03ac \u03b1\u03c0\u03cc \u03b1\u03c1\u03b9\u03b8\u03bc\u03cc\u03c5 \u03b1\u03ba\u03cc\u03bb\u03bf\u03b8\u03b9\u03b1\u03c2 (receiver buffer).
 - o \u039c\u03b5 \u03b2\u03ac\u03c3\u03b7 \u03c4\u03b7\u03bd \u03b1\u03c1\u03c7\u03b9\u03ba\u03b7 \u03c5\u03c0\u03cc\u03b8\u03b5\u03c3\u03b7 \u03cc\u03c4\u03b9 \u03cc \u03b1\u03c0\u03cc\u03c3\u03c4\u03bf\u03bb\u03b5\u03b1\u03c2 \u03c7\u03c1\u03b7\u03c3\u03b9\u03bc\u03c0\u03bf\u03b9\u03b5\u03b9 \u03b5\u03bd\u03b1 \u03c0\u03b1\u03c1\u03ac\u03b8\u03c5\u03c1\u03bf \u03bc\u03b5\u03b3\u03b5\u03b8\u03bf\u03c5\u03c2 N, \u03c3\u03b5 \u03ba\u03ac\u03b8\u03b5 \u03bc\u03b5\u03c4\u03ac\u03b4\u03cc\u03c3\u03b7 \u03bc\u03c0\u03bf\u03c1\u03bf\u03c5\u03bd \u03bd\u03b1 \u03c3\u03c4\u03b1\u03bb\u03bf\u03bd \u03ba\u03ac\u03c0\u03bf\u03b9\u03b1 \u03b1\u03ba\u03cc\u03bc\u03b1 \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03c7\u03c9\u03c1\u03b9\u03c2 \u03bd\u03b1 \u03c0\u03b5\u03c1\u03b9\u03bc\u03b5\u03bd\u03b5\u03b9 \u03b5\u03c0\u03b9\u03b2\u03b5\u03b2\u03b1\u03b9\u03c9\u03c3\u03b7. \u0391\u03c5\u03c4\u03cc\u03c2 \u03cc \u03b5\u03bb\u03b5\u03b3\u03c7\u03cc\u03c2 \u03c5\u03bb\u03cc\u03c0\u03bf\u03b9\u03b5\u03b9\u03c4\u03b1\u03b9 \u03bc\u03b5 \u03c4\u03b7 \u03c7\u03c1\u03b7\u03c3\u03b7 \u03b5\u03bd\u03cc\u03c2 \u03ba\u03c5\u03bb\u03b9\u03cc\u03bc\u03b5\u03bd\u03bf\u03c5 \u03c0\u03b1\u03c1\u03b1\u03b8\u03c5\u03c1\u03bf \u03c0\u03b1\u03bd\u03c9 \u03b1\u03c0\u03cc \u03c4\u03bf\u03c5\u03c2 \u03b4\u03b9\u03b1\u03b8\u03b5\u03c3\u03b9\u03bc\u03cc\u03c5 \u03b1\u03c1\u03b9\u03b8\u03bc\u03cc\u03c5 \u03b1\u03ba\u03cc\u03bb\u03bf\u03b8\u03b9\u03b1\u03c2. \u039c\u03cc\u03c0\u03c9\u03c2 \u03c6\u03b1\u03b9\u03bd\u03b5\u03b9\u03c4\u03b1\u03b9 \u03ba\u03b1\u03b9 \u03c3\u03c4\u03bf \u03c3\u03c7\u03b7\u03bc\u03b1 3, \u03c4\u03b1 \u03c3\u03b7\u03bc\u03b5\u03b9\u03c9\u03bc\u03b5\u03bd\u03bf \u03bc\u03b5 \u03c0\u03c1\u03ac\u03c3\u03b9\u03bd\u03bf \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03b5\u03c7\u03bf\u03bd \u03c3\u03c4\u03b1\u03bb\u03b5\u03b9 \u03ba\u03b1\u03b9 \u03b5\u03c7\u03b5\u03b9 \u03bb\u03b7\u03c6\u03b8\u03b5\u03b9 \u03c4\u03bf ACK \u03c4\u03bf\u03c5\u03c2. \u039c\u03bf \u03b5\u03bd\u03b5\u03c1\u03b3\u03cc \u03c0\u03b1\u03c1\u03ac\u03b8\u03c5\u03c1\u03bf \u03c0\u03b5\u03c1\u03b9\u03bb\u03b1\u03bc\u03b2\u03b1\u03bd\u03b5\u03b9 \u03c4\u03b1 sequence numbers \u03c0\u03bf\u03c5 \u03b5\u03b9\u03bd\u03b1 \u03ba\u03b9\u03c4\u03c1\u03b9\u03bd\u03b1 \u03ba\u03b1\u03b9 \u03b3\u03b1\u03bb\u03ac\u03b6\u03b9\u03b1. \u039c\u03bf sequence numbers \u03c0\u03bf\u03c5 \u03b5\u03b9\u03bd\u03b1 \u03bc\u03b1\u03c1\u03ba\u03c1\u03b9\u03c3\u03bc\u03b5\u03bd\u03bf \u03bc\u03b5 \u03ba\u03b9\u03c4\u03c1\u03b9\u03bd\u03bf \u03b1\u03bd\u03c4\u03b9\u03c3\u03c4\u03bf\u03b9\u03c7\u03bf\u03c5\u03bd \u03c3\u03b5 \u03c0\u03b1\u03ba\u03b5\u03c4\u03b1 \u03c0\u03bf\u03c5 \u03b5\u03c7\u03bf\u03bd \u03c3\u03c4\u03b1\u03bb\u03b5\u03b9 \u03b1\u03bb\u03bb\u03ac \u03b4\u03b5\u03bd \u03b5\u03c7\u03b5\u03b9 \u03c0\u03b1\u03c1\u03b5\u03b9 \u03b1\u03ba\u03cc\u03bc\u03b1 \u03b5\u03c0\u03b9\u03b2\u03b5\u03b2\u03b1\u03b9\u03c9\u03c3\u03b7 \u03cc \u03b1\u03c0\u03cc\u03c3\u03c4\u03bf\u03bb\u03b5\u03b1\u03c2. \u039e\u03b9 \u03c0\u03b1\u03c1\u03ac\u03bc\u03b5\u03c4\u03c1\u03bf\u03b9 \u03c3\u03b5\u03bd_base (sb) \u03ba\u03b9 \u03bd\u03b5\u03be\u03c4\u03c3\u03b5\u03c1\u03b9\u03b1\u03bd\u03b9\u03c5\u03bd (nsn) \u03b5\u03b9\u03bd\u03b1 \u03bc\u03b5\u03c4\u03b1\u03b2\u03b1\u03bb\u03bb\u03cc\u03bc\u03b5\u03bd\u03b5\u03c2 (\u03b1\u03c5\u03be\u03b1\u03bd\u03bf\u03bd\u03b1\u03b9) \u03b4\u03b5\u03bd \u03b5\u03b9\u03bd\u03b1 \u03c3\u03c4\u03b1\u03b8\u03b5\u03c1\u03b5\u03c2. \u039a\u03ac\u03b8\u03b5 \u03c6\u03c9\u03c1\u03ac \u03b1\u03bd\u03ac\u03bb\u03cc\u03b3\u03b1 \u03bc\u03b5 \u03c4\u03bf \u03b3\u03b5\u03b3\u03bf\u03bd\u03cc\u03c2 \u03b7 \u03c4\u03b9\u03bc\u03b7 \u03c4\u03bf\u03c5\u03c2 \u03b1\u03c5\u03be\u03b1\u03bd\u03b5\u03c4\u03b1\u03b9 \u03bc\u03b5 \u03b2\u03ac\u03c3\u03b7 \u03c4\u03bf \u03bc\u03b5\u03b3\u03b5\u03b8\u03bf\u03c2 \u03c4\u03c9 \u03c0\u03b1\u03ba\u03b5\u03c4\u03bf\u03c5 \u03c0\u03bf\u03c5 \u03b1\u03c0\u03cc\u03c3\u03c4\u03b5\u03bb\u03bb\u03b5\u03c4\u03b1\u03b9 \u03b7 \u03c0\u03bf\u03c5 \u03b3\u03b9\u03bd\u03b5\u03c4\u03b1\u03b9 Ack. \u0395\u03c0\u03b9 \u03c4\u03b7\u03c2 \u03cc\u03c5\u03c3\u03b9\u03b1\u03c2 \u03b7 \u03bc\u03b5\u03c4\u03b1\u03b2\u03bb\u03b7\u03c4\u03b7 \u03c3\u03b5\u03bd_base \u03c0\u03b1\u03c1\u03b9\u03c3\u03c4\u03b1\u03bd\u03b5\u03b9 \u03c4\u03bf \u03c0\u03b1\u03bb\u03b1\u03b9\u03cc\u03c4\u03b5\u03c1\u03bf \u03bc\u03b7 \u03b3\u03bd\u03c9\u03c3\u03c4\u03bf\u03c0\u03bf\u03b9\u03b8\u03b5\u03bd \u03b1\u03c1\u03b9\u03b8\u03bc\u03cc \u03b1\u03ba\u03cc\u03bb\u03bf\u03b8\u03b9\u03b1\u03c2 \u03c0\u03b1\u03ba\u03b5\u03c4\u03bf\u03c5 \u03b5\u03bd\u03c9 \u03b7 \u03bd\u03b5\u03be\u03c4\u03c3\u03b5\u03c1\u03b9\u03b1\u03bd\u03b9 \u03c4\u03bf\u03bd \u03b5\u03c0\u03cc\u03bc\u03b5\u03bd\u03bf \u03b4\u03b9\u03b1\u03b8\u03b5\u03c3\u03b9\u03bc\u03cc \u03b1\u03c1\u03b9\u03b8\u03bc\u03cc \u03b1\u03ba\u03cc\u03bb\u03bf\u03b8\u03b9\u03b1\u03c2.



Σχήμα 3: Διαχείριση παραθύρου αποστολέας

- Χρήση χρονομετρητών (timeout counters) στη μεριά του αποστολέα για το διάστημα αναμονής για επιβεβαίωση (ack).

IV. Υλοποίηση TCP αποστολέα και παραλήπτη

- Λειτουργικότητα αποστολέα:
 - Θα πρέπει να διαχειριστεί τρία βασικά γεγονότα: a)Λήψη δεδομένων από το επίπεδο εφαρμογής b)Εκπνοή ενός timeout counter c)λήψη επιβεβαίωσης
 - a) Για κάθε πακέτο που παραλαμβάνει από το επίπεδο εφαρμογής ελέγχει αν $nextseqnum < send_base + N$. Ελέγχει δηλαδή αν το παράθυρο είναι ήδη γεμάτο. Γεμάτο είναι το παράθυρο του αποστολέα όταν εκκρεμεί η επιβεβαίωση αριθμού πακέτων (bytes) όσο και το μέγεθος του παραθύρου του.

Στη συνέχεια, αν ο timeout counter ΔΕΝ είναι ενεργοποιημένος, ενεργοποιείται και αρχίζει να μετράει το διάστημα για το οποίο ο αποστολέας θα περιμένει μια επιβεβαίωση.

Τέλος, υπολογίζει $nextseqnum += \text{Μέγεθος πακέτου}$ και παραδίδει το πακέτο στο IP

- b) Αν παρατηρηθεί εκπνοή του timeout counter τότε επαναμεταδίδει (παίρνοντας το από στον sender buffer) το πακέτο με το μικρότερο sequence number για το οποίο δεν έχει ληφθεί κάποια επιβεβαίωση. Στη συνέχεια επανεκκινεί τον timeout counter.
- c) Αν ληφθεί μια επιβεβαίωση διαβάζει τον αριθμό ακολουθίας που περιλαμβάνει η επιβεβαίωση και τη συγκρίνει με τη παράμετρο $send_base$. Υπενθυμίζεται ότι η μεταβλητή $send_base$ παριστάνει το παλαιότερο μη γνωστοποιηθέν αριθμό ακολουθίας πακέτου. Επειδή το TCP χρησιμοποιεί cumulative acks, μια επιβεβαίωση με αριθμό ακολουθίας y υποδεικνύει στον παραλήπτη ότι έχει παραλάβει σωστά όλα τα bytes πακέτων έως και $y-1$.

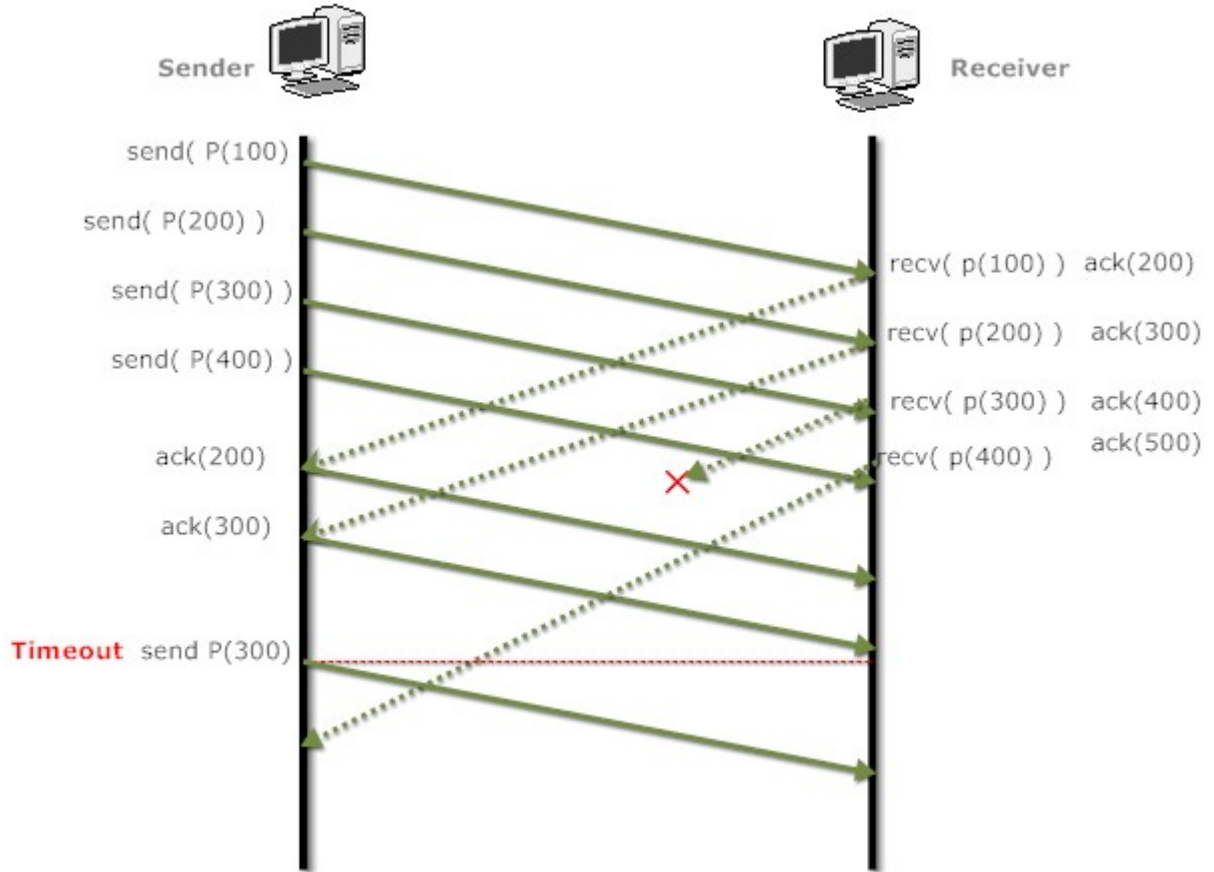
Αν $y > send_base \rightarrow send_base = y$

Αν υπάρχουν μη γνωστοποιηθέντα πακέτα στον αποστολέα, επανεκκινείται ο timeout counter

- Λειτουργικότητα παραλήπτη:
 - Αν ληφθεί πακέτο με αριθμό ακολουθίας n και μέγεθος L και στη σωστή σειρά τότε στέλνεται στον αποστολέα μια επιβεβαίωση με πεδίο $n+L$ καταδεικνύοντας τον επόμενο αναμενόμενο αριθμό ακολουθίας.

Σε οποιαδήποτε άλλη περίπτωση, ξαναστέλνει ένα ACK για το πακέτο που λήφθηκε πιο πρόσφατα και στη σωστή σειρά.

V. Παράδειγμα 3: Απώλεια επιβεβαίωσης από ένα παράθυρο με μέγεθος 400 bytes



Σχήμα 4: Απώλεια πακέτου από ένα παράθυρο με μέγεθος 400bytes

Έστω αρχικά ότι $sb=100$, $nsn=100$. Επίσης εφόσον το παράθυρο του αποστολέα είναι 400bytes αυτό σημαίνει ότι ανά πάσα στιγμή μπορούν να έχουν σταλεί και να παραμένουν μη επιβεβαιωμένα πακέτα των οποίων το συνολικό μέγεθος είναι 400bytes. Έστω επίσης ότι όλα τα πακέτα που μεταδίδει ο αποστολέας έχουν σταθερό μέγεθος $L=100$ bytes.

Το σχήμα 4 παριστάνει τις ενέργειες που γίνονται τόσο στη μεριά του αποστολέα όσο και στη μεριά του παραλήπτη για κάθε γεγονός. Ο παρακάτω πίνακας συνοψίζει τις ενέργειες του αποστολέα και του παραλήπτη για κάθε γεγονός:

Γεγονός	Αποστολέας	Παραλήπτης
	$Nsn < sb + N$ <code>Send(p(100))</code>	
<code>Send(p(100))</code>	1) $Nsn = Nsn + L = 200$ 2) <code>Set Timer</code>	

	3)Nsn < sb+N Send(p(200))	
Send(p(200))	1)Nsn = Nsn + L = 300 2)Nsn < sb+N = 500, Send(p(300))	
Send(p(300))	1)Nsn = Nsn + L = 400 2)Nsn < sb+N = 500, Send(p(400))	
Send(p(400))	1)Nsn = Nsn + L = 500	
Recv(p(100))		Last correct sn = 100 Ack(200)
Recv(p(200))		Last correct sn = 200 Ack(300)
Recv(p(300))		Last correct sn = 300 Ack(400)
Recv(p(400))		Last correct sn = 500 Ack(500)
Ack(200) - for packet p(100)	Sb = 200 Reset Timer p(100) = Acked	
Ack(300)	Sb = 300 Reset Timer p(200) = Acked	
	If(data from APP layer) Nsn < sb + N = 300 + 400 = 700 Send(p(500))	
Send(p(500))	1)Nsn = Nsn + L = 600 2)Nsn < sb+N = 700, Send(p(600))	
Send(p(600))	1)Nsn = Nsn + L = 700	
Timeout	Send(p(300)) Reset Timer	

VI. Παράδειγμα 4: Απώλεια επιβεβαίωσης τυχαίου πακέτου εν μέσω παραθύρου και υπολογισμός ελάχιστων και μέγιστων αναγκαίων επαναμεταδόσεων

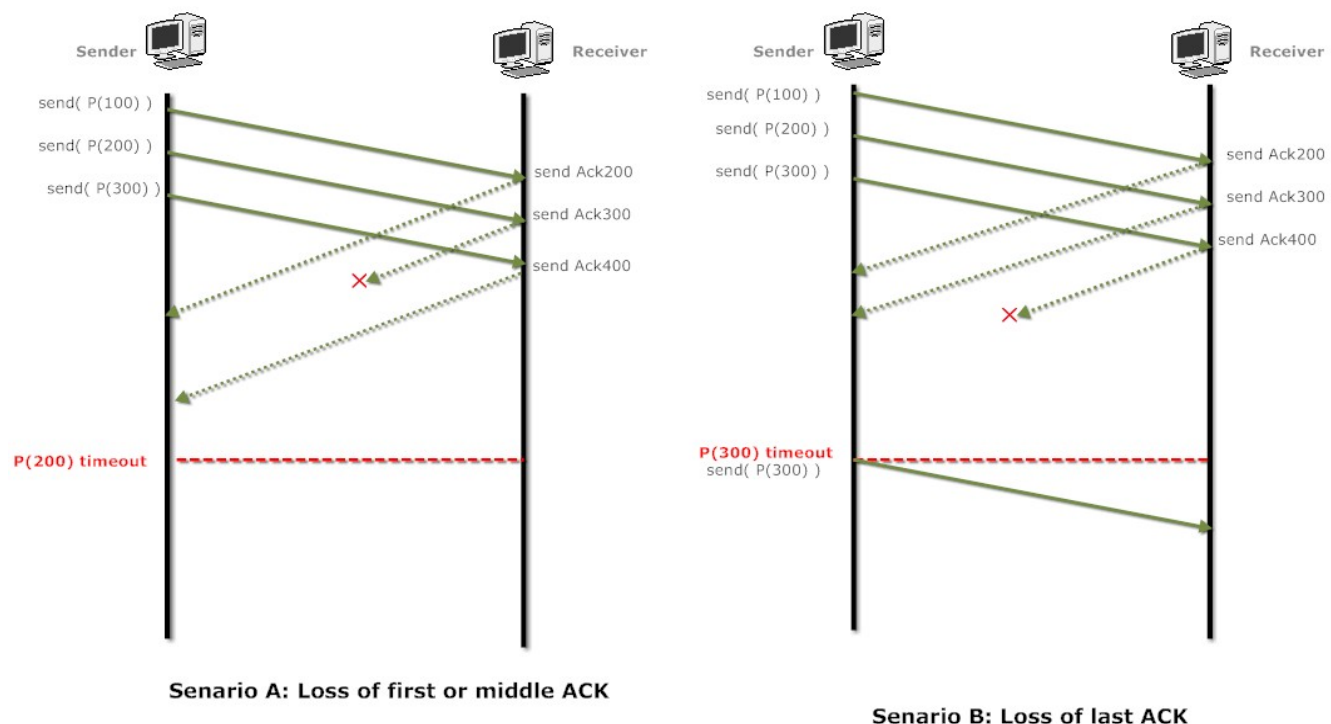
Η συσκευή A στέλνει ένα αρχείο μεγέθους M bits σε μία συσκευή B χρησιμοποιώντας TCP. Το κάθε πακέτο έχει μέγεθος p. Όλα τα πακέτα λαμβάνονται στη σωστή σειρά χωρίς κανένα λάθος από τον B. Ας υποθέσουμε όμως ότι ακριβώς ένα ACK για ένα από τα παραπάνω data packets χάνεται, αλλά όλα τα υπόλοιπα φτάνουν σωστά στον A πριν το δικό τους timeout. Ποιός είναι ο μέγιστος και ποιός ο ελάχιστος αριθμός data packets που θα ξαναστείλει ο A και γιατί?

Έστω ότι το παράθυρο που χρησιμοποιεί η συσκευή A ισοδυναμεί σε 3 πακέτα. Αυτό σημαίνει ότι μπορεί να έχει ανά πάσα στιγμή το πολύ 3 πακέτα που έχουν ήδη μεταδοθεί χωρίς να έχει λάβει επιβεβαίωση για κανένα από αυτά. Αν 100bytes είναι το μέγεθος του πακέτου και αρχικά sb=nsn=100 ο αποστολέας μπορεί να έχει το πολύ 300bytes δεδομένων τα οποία έχει στείλει σε πακέτα στον παραλήπτη και δεν έχουν επιβεβαιωθεί.

Ο ελάχιστος και ο μέγιστος αριθμός επαναμεταδόσεων που θα απαιτηθούν εξαρτάται από το ποια επιβεβαίωση θα χαθεί. Στο Scenario A του σχήματος 5, ο αποστολέας παραλαμβάνει σωστά τα ack(200), ack(400) που αφορούν τα πακέτα με sequence numbers 100 και 300 αντίστοιχα. Το

ack(300) αντιθέτως χάνεται. Επειδή όμως το TCP χρησιμοποιεί cumulative acks, το ack(400) επιβεβαιώνει όλα τα bytes με sequence number < 400 που σημαίνει ότι επιβεβαιώνει στον αποστολέα ότι έχει παραλάβει σωστά και τα 3 πακέτα. Συνεπώς, όταν λήξει ο timeout counter που έγινε set με την λήψη του ack(400), το παράθυρο του αποστολέα θα είναι άδαιο καθώς $sb=nsn=400$ που σημαίνει ότι δεν υπάρχουν δεδομένα on-the-fly στο δίκτυο τα οποία δεν έχουν επιβεβαιωθεί.

Στο δεύτερο σενάριο, (scenario B στο σχήμα 5) χάνεται το τελευταίο ack -> ack(400) ενώ τα ack(200), ack(300) παραλαμβάνονται σωστά. Μόλις εκπνεύσει ο timeout counter το μόνο πακέτο για το οποίο ο παραλήπτης δεν έχει πάρει επιβεβαίωση είναι το πακέτο με sequence number 400 άρα θα γίνει μία επαναμετάδοση. Συνεπώς ο ελάχιστος αριθμός επιβεβαιώσεων που μπορούν να χρειαστούν είναι 0 και ο μέγιστος 1.



Σχήμα 5: Απώλεια τυχαίου πακέτου εν μέσω παραθύρου μετάδοσης

Γενικεύοντας το παραπάνω παράδειγμα: αν το παράθυρο αποστολέα (sender's window) που χρησιμοποιεί η συσκευή A είναι N (πακέτα) τότε διακρίνουμε δύο περιπτώσεις: Αν χαθεί η επιβεβαίωση για οποιοδήποτε από τα N-1 πακέτα, τότε η επιβεβαίωση για το τελευταίο (N-th) πακέτο εφόσον είναι συσσωρευτική θα επιβεβαιώσει τη παραλαβή όλων των πακέτων στη B οπότε δε θα χρειαστεί καμία επαναμετάδοση. Μόνο στην περίπτωση που χαθεί η επιβεβαίωση για το τελευταίο πακέτο του παραθύρου, θα χρειαστεί μία επαναμετάδοση.

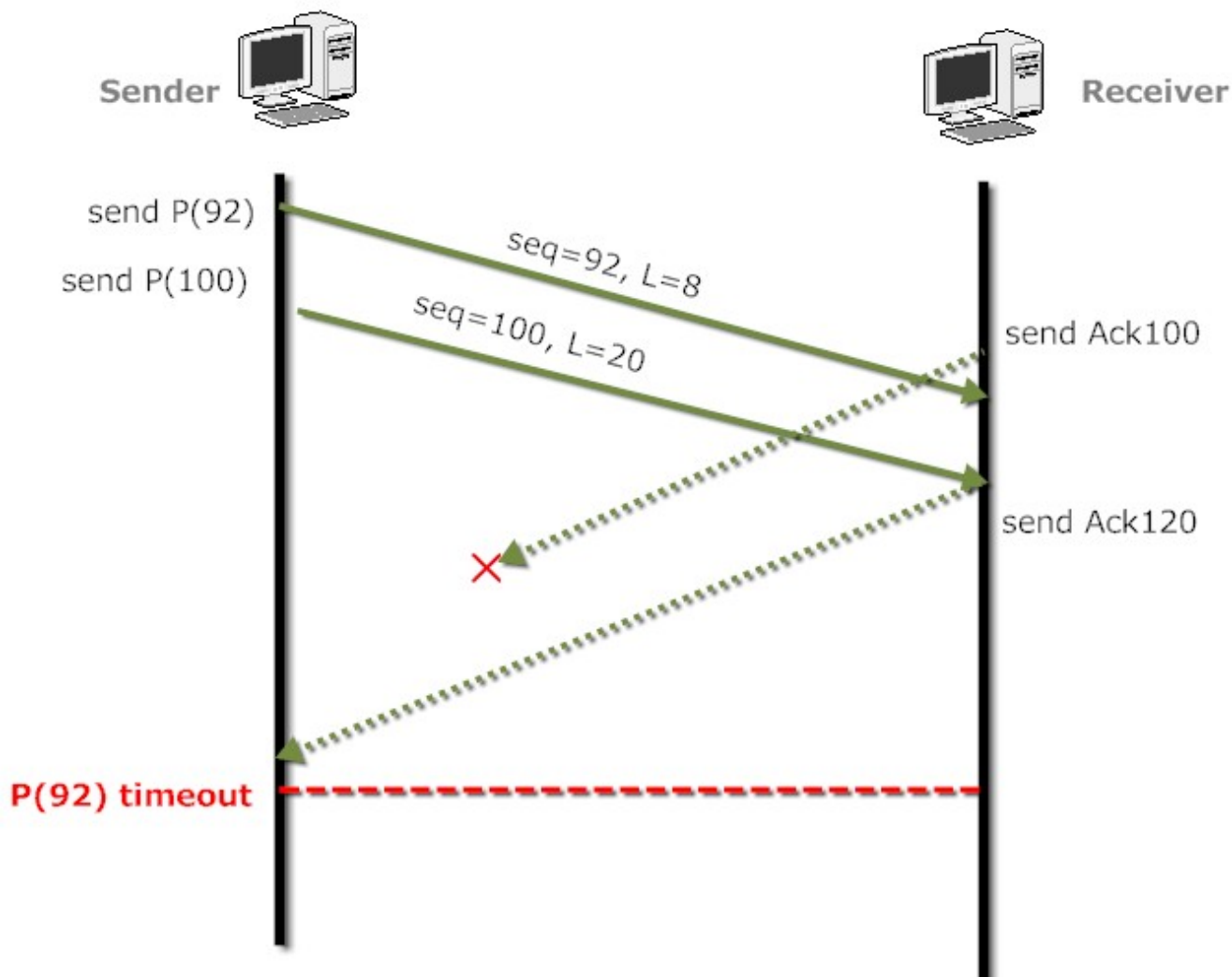
VII. Παράδειγμα 5: Παράδειγμα συσσωρευτικής επιβεβαίωσης

Ας υποθέσουμε ένα σενάριο όπου ο αποστολέας στέλνει δύο πακέτα στον παραλήπτη με αριθμούς ακολουθίας 92 και 100 αντίστοιχα. Έστω επίσης ότι το μέγεθος των πακέτων είναι 8 και 20 αντίστοιχα ενώ το μέγεθος του παραθύρου είναι 28bytes.

Αν αρχικά, $sb=nsn=92$, μόλις ο αποστολέας στείλει τα δύο πακέτα, $nsn = 92+8+20 = 120$.

Ο παραλήπτης δημιουργεί δύο επιβεβαιώσεις με αριθμούς ακολουθίας 100 και 120 αντίστοιχα. Η επιβεβαίωση με αριθμό ακολουθίας 100 χάνεται και παραλαμβάνεται μόνο το ack(120).

Όταν ο αποστολέας παραλάβει το ack(120) θα θέσει sb=120=nsn που σημαίνει ότι ο παραλήπτης έχει πάρει σωστά όλα τα bytes από 92 έως και 119. Όταν θα εκπνεύσει ο timeout counter, δεν θα υπάρχουν ανεπιβεβαίωτα δεδομένα στον αποστολέα και έτσι δεν θα γίνει καμία επαναμετάδοση.



Σχήμα 6: Απώλεια τυχαίας επιβεβαίωσης

VIII. Παράδειγμα 6: Υπολογισμός μεγέθους παραθύρου

Έστω το σενάριο που μελετήθηκε στην Παράδειγμα 1 αυτής της αναφοράς (σχήμα 1). Γίνεται υπόθεση για μηδενικό queuing και processing delay καθώς και για $Acktrans = 0$. Έστω επίσης μέγεθος πακέτου $L = 1000$ bytes, χωρητικότητα μονοπατιού $R = 1$ Gps και καθυστέρηση διάδοσης $D_{prop} = 15$ msec.

Ο χρόνος για να μεταδοθεί το πακέτο στο μέσο είναι: $D_{trans} = \frac{L}{R} = \frac{1000 * 8}{10^9} = 0,008$ m sec

Πόσο θα πρέπει να είναι το μέγεθος του παραθύρου έτσι ώστε να χρησιμοποιείται το 90% της χωρητικότητας του μονοπατιού από τον source στον destination?

Για να χρησιμοποιείται το 90% της χωρητικότητας, τότε θα πρέπει ο αποστολέας το 90% του χρόνου που χρειάζεται για να σταλούν η πακέτα να μεταδίδει κίνηση στο δίκτυο.

Γνωρίζουμε από το προηγούμενο παράδειγμα ότι ο συνολικός χρόνος για να σταλεί ένα πακέτο είναι:

$$D_{trans} + ACK_{trans} + 2 * D_{prop}^{ACK_{trans}=0} = 30,008 \text{ και ότι } L/R = 0,008.$$

Έτσι θέλουμε: $\frac{n * 0,008}{30,008} = 0,9$ όπου n το μέγεθος του παραθύρου. Συνεπώς, ο αριθμός των πακέτων που πρέπει να σταλούν ταυτόχρονα είναι περίπου 3376.