

# ΚΕΦΑΛΑΙΟ 1

## Αλγόριθμοι επίλυσης συγκρούσεων

Η παρούσα εργασία ασχολείται με τη σχεδίαση και τη μελέτη απόδοσης αλγορίθμων πολλαπλής προσπέλασης - επίλυσης συγκρούσεων πακέτων για την εκρηκτική κατανομή χρόνου μεταξύ αφίξεων διαδοχικών πακέτων *Pareto* [12]- [13]. Αυτό το κεφάλαιο επιχειρεί μια σύντομη (και συνεπώς όχι πλήρη) εισαγωγή στους **Αλγόριθμους επίλυσης συγκρούσεων πακέτων (Collision Resolution Algorithms)**, βλέπε επίσης [4] για μια πληρέστερη ανάλυση.

### 1.1 Εισαγωγή στους Αλγόριθμους επίλυσης συγκρούσεων

Θεωρούμε την περίπτωση όπου γεωγραφικά κατανεμημένοι, ανεξάρτητοι χρήστες, επιθυμούν να επικοινωνήσουν μεταξύ τους ή με έναν κεντρικό σταθμό, μεταδίδοντας μηνύματα πάνω σε ένα κοινό κανάλι. Το πρόβλημα είναι ο σχεδιασμός ενός αλγορίθμου, δηλαδή ενός συνόλου από κανόνες, που καθορίζουν τις χρονικές στιγμές που μπορεί να μεταδώσει ο κάθε χρήστης.

Το πρόβλημα αυτό, αναφέρεται ως **Πρόβλημα Πολλαπλής Προσπέλασης (Multiple Access Problem)**, και οι αλγόριθμοι που χρησιμοποιούνται ονομάζονται **Αλγόριθμοι Πολλαπλής Προσπέλασης (Multiple Access Algorithms)** [4]. Τυπικά παραδείγματα συστημάτων τηλεπικοινωνιών πολλαπλής προσπέλασης είναι τα ακόλουθα :

(α) *Γεωσύγχρονα δορυφορικά συστήματα*, όπου πολλοί σταθμοί εδάφους, μεταδίδουν σ' ένα κοινό δορυφορικό δέκτη και τα λαμβανόμενα μηνύματα αναμεταδίδονται σε σταθμούς εδάφους.

(β) *Ασύρματα δίκτυα με πακέτα* (ως πακέτο ορίζεται η μονάδα πληροφορίας που διακινείται στο δίκτυο), όπου τα πακέτα που μεταδίδονται μ' έναν ράδιο - μεταδότη μπορεί να ληφθούν - πάνω σε μία μεγάλη περιοχή - από ένα αριθμό από χρήστες. Η δυνατότητα αυτή αναφέρεται ως δυνατότητα ευρείας μεταδόσεως. Έτσι το επίγειο ασύρματο κανάλι που μεταδίδει πακέτα παρέχει μία πλήρη τοπολογία συνδεδεμένου δικτύου για ένα μεγάλο αριθμό από χρήστες, οι οποίοι βρίσκονται μέσα στα όρια μεταδόσεως ο ένας του άλλου.

Υποθέτουμε ότι μόνο ένα μήνυμα μπορεί να μεταδοθεί επιτυχώς σε κάποια χρονική στιγμή, επάνω σε ένα κοινό τηλεπικοινωνιακό κανάλι. Αν μεταδοθούν, στο κοινό κανάλι, πάνω από ένα μηνύματα ταυτόχρονα, θεωρούμε ότι προκαλείται *σύγκρουση* και το λαμβανόμενο σήμα είναι το άθροισμα των σημάτων που μεταδόθηκαν, παραμορφωμένα πιθανά από θόρυβο. Η πληροφορία που περιέχεται στα αρχικά μηνύματα θεωρείται ότι χάθηκε. Έτσι κάθε μήνυμα που συμμετείχε στη σύγκρουση πρέπει να αναμεταδοθεί σε κάποια επόμενη χρονική στιγμή, με πιθανές επαναλήψεις της αναμετάδοσης μέχρι να μεταδοθεί επιτυχώς.

Ένας αλγόριθμος πολλαπλής προσπέλασης μπορεί να είναι *συγκεντρω-τικός* (centralized), όπου ένας κεντρικός σταθμός συντονίζει τη δράση των χρηστών, ή *κατανεμημένος* (distributed), όπου κάθε χρήστης ενεργεί σύμφωνα με προκαθορισμένους κανόνες, χωρίς τη μεσολάβηση κεντρικού συντονιστή, ή ακόμη και ένας συνδυασμός των δύο. Η απόδοση ενός αλγορίθμου πολλαπλής προσπέλασης εξαρτάται από τα στατιστικά του χαρακτηριστικά και από τα στατιστικά χαρακτηριστικά της διαδικασίας γέννησης μηνυμάτων στο σύστημα. Οι *αλγόριθμοι διαίρεσης στο χρόνο* (Time Division Multiple Access, TDMA) και *διαίρεσης στη συχνότητα* (Frequency Division Multiple Access, FDMA), ήταν οι πρώτοι που σχεδιάστηκαν για να επιλύσουν το πρόβλημα της Πολλαπλής Προσπέλασης. Είναι και οι δύο *αλγόριθμοι χωρίς συγκρούσεις* (collision free algorithms).

Στον αλγόριθμο FDMA, το διαθέσιμο εύρος του καναλιού (channel bandwidth) χωρίζεται σε ζώνες και κάθε ζώνη παραχωρείται σ'έναν χρήστη. Στον αλγόριθμο TDMA, ο χρόνος διαιρείται σε *πλαίσια (frames)*, κάθε πλαίσιο υποδιαιρείται σε *χρονικά τμήματα (slots)*, και επιτρέπεται σε κάθε χρήστη να μεταδίδει μόνο μια φορά μέσα σ' ένα πλαίσιο και μόνο μέσα στο χρονικό τμήμα που του αντιστοιχεί. Όταν όλοι οι χρήστες στέλνουν συνεχώς πακέτα, ο FDMA και ο TDMA είναι πολύ αποδοτικοί, επιτρέπουν 100% χρήση του

καναλιού. Εάν όμως ο αριθμός των χρηστών που χρησιμοποιούν το κανάλι είναι μεγάλος και πιθανόν μεταβάλλεται με το χρόνο, και αν ο κάθε χρήστης μεταδίδει πακέτα κατά εκρηκτικό τρόπο (δηλαδή για ένα χρονικό διάστημα έχει πολλά πακέτα για μετάδοση, ενώ στη συνέχεια για κάποιο άλλο διάστημα δεν έχει πακέτο για μετάδοση), και οι δύο αλγόριθμοι εισάγουν μεγάλες καθυστερήσεις (της τάξης του αριθμού των χρηστών στο δίκτυο). Έστω, για παράδειγμα, ένας TDMA αλγόριθμος. Υποθέτουμε ότι υπάρχουν  $M$  ( $M \gg 1$ ), χρήστες στο δίκτυο. Ένα πακέτο που γεννάει ένας χρήστης πρέπει να περιμένει κατά μέσο όρο  $M/2$  χρονικά τμήματα για να μεταδοθεί, ακόμη και αν κανένας άλλος χρήστης δεν έχει πακέτα για μετάδοση. Μετρήσεις που έχουν γίνει σε συστήματα διαμερισμού χρόνου επικοινωνιακού καναλιού (time-sharing communication systems), δείχνουν ότι τα δεδομένα υπολογιστών, συμπεριφέρονται κατά έντονα εκρηκτικό (bursty) τρόπο. Ανάλογα με την ταχύτητα του καναλιού ο λόγος του υψηλότερου προς το μέσο ρυθμό γέννησης πακέτων δεδομένων μπορεί να είναι μέχρι και 2000 προς 1 (!) [1]. Από τις παραπάνω αναφορές γίνεται εμφανές ότι, εάν ο αριθμός των χρηστών είναι αρκετά μεγάλος και οι χρήστες γεννούν σποραδικά “εκρήξεις” με πακέτα (packet bursts), απαιτούνται διαφορετικές τεχνικές διαμερισμού του καναλιού από αυτές που έχουν περιγραφεί παραπάνω.

Οι **Αλγόριθμοι Τυχαίας Προσπέλασης** (Random Access Algorithms RAA), σχεδιάστηκαν ακριβώς γι’ αυτό το λόγο. Η βασική διαφορά ανάμεσα σε έναν αλγόριθμο τυχαίας προσπέλασης και έναν αλγόριθμο πολλαπλής προσπέλασης διαίρεσης χρόνου ή συχνότητας (TDMA ή FDMA) είναι ότι ο πρώτος, επιτρέπει ταυτόχρονες προσπάθειες μετάδοσης από διαφορετικούς χρήστες στην ίδια ζώνη συχνότητας. Έτσι είναι πιθανές οι *συγκρούσεις*.

Στη συνέχεια το θέμα που θα διερευνηθεί είναι ο σχεδιασμός αλγορίθμων επίλυσης συγκρούσεων, που επιλύουν τις συγκρούσεις πακέτων όταν προκύπτουν, και διατηρούν ευσταθή τη λειτουργία του συστήματος με χαμηλές καθυστερήσεις πακέτων. Ως *καθυστέρηση πακέτου* ορίζεται, ο χρόνος που μεσολαβεί από τη στιγμή που γεννιέται ένα πακέτο, μέχρι το τέλος της χρονικής στιγμής που περιέχει την επιτυχή μετάδοσή του. Εξαιτίας του γεγονότος ότι ένα τμήμα του χρόνου του καναλιού δαπανάται για την επίλυση συγκρούσεων, η χρησιμοποίηση του καναλιού είναι γενικά μικρότερη από 100%. Πάντως, οι καθυστερήσεις των πακέτων στην περίπτωση χαμηλών ρυθμών γέννησης πακέτων πληροφορίας, είναι χαμηλές και η χρησιμοποίηση του καναλιού εξαρτάται από το συνολικό ρυθμό γέννησης πακέτων πληροφορίας (φορτίο του καναλιού), και όχι από τον αριθμό των χρηστών.

Στη συνέχεια θα εξεταστούν μόνο σύγχρονα, ως προς το χρόνο (με τον άξονα του χρόνου διαιρεμένο σε ίσα χρονικά τμήματα), **Συστήματα Επικοινωνιών Τυχαίας Προσπέλασης** (Random Access Communication Systems). Ακολουθεί, σύντομη περιγραφή και ανάλυση των παραμέτρων σχεδιασμού, που υποθέτουν οι αλγόριθμοι που θα αναφερθούν [4] :

(1) *Σύγχρονα συστήματα*, με τον άξονα του χρόνου διαιρεμένο σε ίσα χρονικά τμήματα (slots). Τα μηνύματα χωρίζονται σε σταθερού μήκους πακέτα και κάθε πακέτο απαιτεί ένα χρονικό τμήμα για να μεταδοθεί. Όλοι οι χρήστες είναι συγχρονισμένοι στην αρχή κάθε χρονικού τμήματος και η μετάδοση πακέτων μπορεί να αρχίσει μόνο στην αρχή κάποιου χρονικού τμήματος.

(2) *Συγκρούσεις και σωστή λήψη*. Υποθέτουμε ότι αν δύο ή περισσότεροι χρήστες προσπαθήσουν να μεταδώσουν τα πακέτα τους σε ένα δεδομένο χρονικό τμήμα, προκύπτει σύγκρουση και όλες οι πληροφορίες που περιέχονται στα μεταδιδόμενα πακέτα καταστρέφονται με αποτέλεσμα να είναι απαραίτητη η επαναμετάδοση. Αν ένας μόνο χρήστης στείλει πακέτο σε ένα δεδομένο χρονικό τμήμα, τότε το πακέτο μεταδίδεται επιτυχώς. Στο τέλος κάθε χρονικού τμήματος, όλοι οι πομποί πληροφορούνται ταυτόχρονα αν το συγκεκριμένο τμήμα χρόνου περιείχε σύγκρουση ή όχι (κενό ή επιτυχία).

(3) *Άμεση πληροφόρηση ανάδρασης*. Στο τέλος κάθε χρονικού τμήματος, κάθε χρήστης ή ίσως μόνο ένα υποσύνολο των χρηστών, παίρνουν πληροφορίες ανάδρασης για την δραστηριότητα μεταδόσεων του συγκεκριμένου χρονικού τμήματος.

(4) *Άπειρος αριθμός χρηστών*. Το σύστημα έχει έναν πολύ μεγάλο (θεωρητικά άπειρο) αριθμό χρηστών και κάθε “φρέσκο” πακέτο (πακέτο που μόλις έφτασε) πηγαίνει σε καινούριο χρήστη. Αυτή η υπόθεση οδηγεί στην μοντελοποίηση της χειρότερης περίπτωσης, μια και κάθε πακέτο που γεννιέται στο σύστημα είναι αναγκασμένο να χρησιμοποιήσει τυχαία προσπέλαση για να μεταδοθεί επιτυχώς στο κανάλι. Ο σχεδιασμός των αλγορίθμων κάτω από αυτή την υπόθεση παρουσιάζει ενδιαφέρον, διότι η απόδοση τους δε θα εξαρτάται από τον αριθμό των χρηστών στο σύστημα, αλλά από τον συνολικό ρυθμό γέννησης πληροφορίας από τους χρήστες.

Η υπόθεση ότι το σύστημα είναι χωρισμένο σε χρονικά τμήματα, οδηγεί σε σύστημα διακριτό στο χρόνο και έτσι απλοποιείται η αναλυτική προσέγγιση. Ο χρονικός συντονισμός των χρηστών δεν είναι απλός, αλλά μπορεί να επιτευχθεί με σχετικά σταθερά ρολόγια και κάποιο χρονικό περιθώριο, ανάμεσα στο τέλος της μετάδοσης ενός πακέτου και στην αρχή του επόμενου χρονικού τμήματος [4]. Η υπόθεση των συγκρούσεων και της σωστής λήψης

αγνοεί την πιθανότητα λαθών, που οφείλονται στο θόρυβο, καθώς επίσης αγνοεί την πιθανότητα τεχνικών σύλληψης (capture), με τις οποίες ο δέκτης μπορεί να *συλλάβει* (να λάβει δηλαδή σωστά) μια μετάδοση σε ένα χρονικό τμήμα που περιέχει πολλαπλές ταυτόχρονες μεταδόσεις. Η υπόθεση της άμεσης πληροφορίας ανάδρασης μπορεί να είναι μη ρεαλιστική, ειδικά στην περίπτωση δορυφορικών καναλιών. Ευτυχώς όμως, η καθυστερημένη ανάδραση, αν και περιπλέκει κάπως την περιγραφή των αλγορίθμων τυχαίας προσπέλασης, δεν προκαλεί επιπλέον βασικά προβλήματα [4].

### 1.2 Ο Αλγόριθμος *ALOHA*

Ο πιο παλιός χρονολογικά αλγόριθμος τυχαίας προσπέλασης είναι ο αλγόριθμος *ALOHA* [5]. Αναπτύχθηκε γύρω στο 1970, για να παρέχει ασύρματη επικοινωνία μεταξύ ενός κεντρικού υπολογιστή και πολλών κατανεμημένων τερματικών στο Πανεπιστήμιο της Hawaii. Ο αλγόριθμος *ALOHA* ανήκει στους αλγορίθμους που η λειτουργία τους βασίζεται στον ανταγωνισμό των πακέτων που ανεβαίνουν στο κοινό κανάλι. Κάθε χρήστης με “φρέσκο” πακέτο (πακέτο που μόλις έχει γεννηθεί στο σύστημα) το μεταδίδει στην αρχή του χρονικού τμήματος που ακολουθεί τη χρονική στιγμή της άφιξης του. Αν δύο ή περισσότεροι χρήστες μεταδώσουν ταυτόχρονα στο κοινό κανάλι, συγκρούονται και κανένα πακέτο δεδομένων δεν λαμβάνεται σωστά. Η πληροφορία ανάδρασης δίνεται με μήνυμα λήψης (acknowledgment type). Έτσι, μόνο οι χρήστες που συμμετέχουν σε μια σύγκρουση ενημερώνονται γι’ αυτήν. Για να επιλύσουν τη σύγκρουση οι παραπάνω χρήστες επιλέγουν τυχαίες χρονικές στιγμές, για την αναμετάδοση των πακέτων τους σύμφωνα με ένα κοινό για όλους πιθανοτικό μηχανισμό. Σαν αποτέλεσμα, το σύστημα περιέχει δύο τύπους χρηστών σε κάθε χρονική στιγμή: αυτούς που έχουν να μεταδώσουν “φρέσκα” πακέτα, και αυτούς των οποίων τα πακέτα είχαν συγκρουστεί σε προηγούμενα χρονικά τμήματα και προσπαθούν να αναμεταδώσουν (μπλοκαρισμένοι χρήστες).

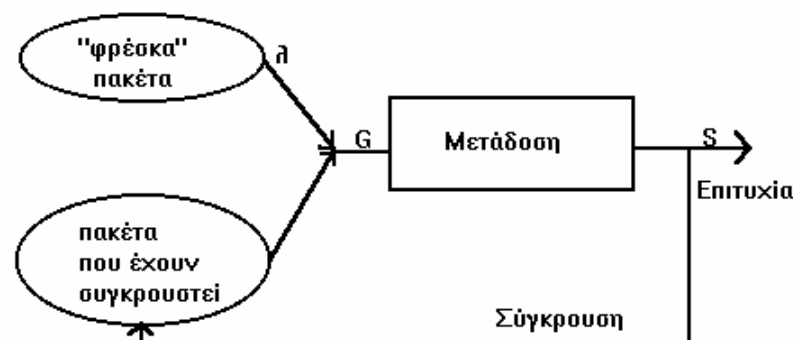
Ο αλγόριθμος *ALOHA* αναλύθηκε για πρώτη φορά από τον Abramson [5]. Ο αριθμός των χρηστών θεωρήθηκε άπειρος και ο αριθμός των “φρέσκων” πακέτων που γεννιούνται σε ένα χρονικό τμήμα υποτέθηκε ότι ακολουθεί κατανομή *Poisson*, με ρυθμό άφιξης πακέτων λ ανά χρονικό τμήμα. Για την απλοποίηση της ανάλυσης έγιναν κάποιες επιπρόσθετες υποθέσεις για την κατανομή του αριθμού των μπλοκαρισμένων χρηστών, και επίσης για την ύπαρξη συνθήκης σταθερής κατάστασης (ότι το σύστημα δηλαδή πάντα φθάνει σε κατάσταση στατιστικής ισορροπίας). Δυστυχώς, μια προσεκτική εξέταση του αλγορίθμου [6] έδειξε ότι

το σύστημα δεν είναι ευσταθές για οποιοδήποτε  $\lambda > 0$ , με την έννοια ότι ο ρυθμός επιτυχών μεταδόσεων πακέτων σαν συνάρτηση του χρόνου σταδιακά μειώνεται στο μηδέν.

Για την κατανόηση της συνέχειας θεωρούνται απαραίτητες οι γνώσεις της έννοιας της **ευστάθειας** και της έννοιας του **throughput** (μέγιστος ρυθμός αφίξεων πακέτων που διατηρεί το σύστημα ευσταθές). Γι' αυτό το λόγο θα ανοίξουμε μια μικρή παρένθεση.

Ορισμός της ευστάθειας ενός δικτύου, που λειτουργεί με έναν συγκεκριμένο αλγόριθμο τυχαίας προσπέλασης. Έστω ότι με  $\lambda$  και  $S$  συμβολίζουμε τους ρυθμούς άφιξης και αναχώρησης, αντίστοιχα, των πακέτων προς και από το δίκτυο (δες **Σχ.1**). Ένα δίκτυο που λειτουργεί με έναν αλγόριθμο τυχαίας προσπέλασης, ονομάζεται ευσταθές αν και μόνο αν υπάρχει ένας θετικός αριθμός  $\Lambda$ , τέτοιος ώστε, για όλους τους ρυθμούς άφιξης  $\lambda$  φρέσκων πακέτων, που γεννιούνται στο δίκτυο, τέτοιους ώστε  $\lambda \leq \Lambda$ , ο ρυθμός άφιξης  $\lambda$  και ο ρυθμός αναχώρησης  $S$  είναι ίσοι. Με άλλα λόγια, σ' ένα ευσταθές σύστημα  $S = \lambda$ , δηλαδή οι ρυθμοί διατηρούνται ίσοι, για όλα τα  $\lambda \leq \Lambda$  όπου  $\Lambda > 0$ . Ο θετικός αριθμός  $\Lambda$ , ονομάζεται **throughput** του αλγορίθμου τυχαίας προσπέλασης, και η κλίμακα των ρυθμών αφίξεων πακέτων ανάμεσα στο 0 και το  $\Lambda$  καθορίζουν την περιοχή ευστάθειας του συγκεκριμένου αλγορίθμου τυχαίας προσπέλασης.

Ο αλγόριθμος **ALOHA** είναι ασταθής. Έχει αποδειχτεί αναλυτικά και έχει επαληθευτεί μέσω προσομοιώσεων, ότι για μεγάλο αριθμό χρηστών δεν μπορεί να δώσει ευσταθή συστήματα για κανένα θετικό ρυθμό άφιξης πακέτων. Με άλλα λόγια, ανεξάρτητα από τον ρυθμό γέννησης φρέσκων πακέτων (μπορεί να είναι όσο κοντά στο μηδέν επιθυμούμε), εάν το σύστημα λειτουργεί για πολύ χρόνο, στο τέλος υποφέρει από τον μεγάλο αριθμό των συγκρούσεων και δεν



**Σχ.1** : Ο ρυθμός άφιξης  $\lambda$ , το προσφερόμενο φορτίο  $G$ , και ο ρυθμός αναχώρησης  $S$  σε ένα **ALOHA** δίκτυο.

είναι δυνατόν να μεταδώσει επιτυχώς οτιδήποτε. Συνεπώς, όταν το κανάλι φορτώνεται με ένα απροσδιόριστο μεγάλο (θεωρητικά άπειρο) αριθμό χρηστών, ο αλγόριθμος **ALOHA** είναι ασταθής, δηλαδή  $\Lambda=0$ . Ακόμη και συστήματα με πεπερασμένο αριθμό χρηστών μπορεί να υποφέρουν από αστάθεια [παρ.4.2.2 του 4].

Η οικογένεια των αλγορίθμων **ALOHA** - εξαιτίας των απλών απαιτήσεων υλοποίησης τους - έγινε δημοφιλής και είχε ευρεία ανάπτυξη στην πράξη, παρά τα προβλήματα ασταθούς λειτουργίας που αναφέρθηκαν και συζητήθηκαν παραπάνω. Η φιλοσοφία της ατομικότητας (individualism) των αλγορίθμων **ALOHA** υπαγορεύει στους χρήστες δικτύων να ενδιαφερθούν για τα αποτελέσματα μόνο των δικών τους μεταδόσεων (δηλαδή, την ανάδραση του καναλιού μόνο κατά τα χρονικά τμήματα που μεταδίδουν). Έτσι, ένας χρήστης δεν χρειάζεται να παρατηρεί την δραστηριότητα του καναλιού συνεχώς, αλλά μόνο αμέσως μετά από την μετάδοσή του. Εάν η μετάδοση είναι μη επιτυχής, δηλαδή ο χρήστης δεν έλαβε κάποιο επιτυχές σήμα λήψης, επανασχεδιάζει μία μελλοντική μετάδοση του πακέτου του και απέχει από την αίσθηση της ανάδρασης του καναλιού μέχρι να έρθει η στιγμή αυτής της επαναμετάδοσης.

Υπάρχουν δύο εναλλακτικές λύσεις για την επιλογή του χρόνου αναμετάδοσης :

- ομοιόμορφη επιλογή του χρονικού διαστήματος μέσα στα επόμενα  $\omega$  χρονικά τμήματα,
- μετάδοση στο επόμενο χρονικό τμήμα με πιθανότητα  $p$ .

Οι παράμετροι  $\omega$  και  $p$  μπορούν να είναι είτε στατικά επιλεγμένες, είτε να αλλάζουν δυναμικά με βάση τον αριθμό των συγκρούσεων τις οποίες έχει υποστεί ένας χρήστης στην προσπάθεια μετάδοσης του πακέτου του. Η επιλογή των παραμέτρων αυτών, έχει ως αποτέλεσμα την αύξηση ή τη μείωση των συγκρούσεων. Η λογική αυτή, της μικρότερης πιθανότητας αναμετάδοσης των πακέτων που έχουν συγκρουστεί, σε σχέση με τα "φρέσκα" πακέτα, δίνει στα "φρέσκα" πακέτα τη δυνατότητα να μεταδοθούν επιτυχώς, με μεγαλύτερη πιθανότητα από τα "μπλοκαρισμένα" πακέτα.

Ευσταθές ALOHA. Η πιθανότητα επαναμετάδοσης μπορεί να αλλάξει δυναμικά με τον αριθμό των μπλοκαρισμένων χρηστών (αυτοί που έχουν πακέτα, τα οποία έχουν υποστεί σύγκρουση και προσπαθούν να αναμεταδώσουν). Ιδανικά, θα θέλαμε να επιλέξουμε την πιθανότητα  $p = 1/n$ , όπου το  $n$  συμβολίζει τον αριθμό των μπλοκαρισμένων χρηστών.

Πρακτικά όμως, δεν μπορούμε να γνωρίζουμε το  $n$ , και χρειάζεται να χρησιμοποιηθούν επιτηδευμένες τεχνικές για να το εκτιμήσουμε. Στη συνέχεια, παρουσιάζεται ένας αναδρομικός αλγόριθμος στον χρόνο, για τον προσεγγιστικό υπολογισμό του  $n$  [4] :

$$\tilde{n}_{\omega+1} = \begin{cases} \max\{\lambda, \tilde{n}_{\omega} + \lambda - 1\} & , \text{για τα χρονικά τμήματα που δεν περιέχουν σύγκρουση} \\ \tilde{n}_{\omega} + \lambda + (e-2)^{-1} & , \text{για τα χρονικά τμήματα που περιέχουν σύγκρουση} \end{cases}$$

όπου  $\tilde{n}_{\omega}$  η εκτίμηση του αριθμού μπλοκαρισμένων χρηστών στην αρχή της χρονικής μονάδας  $\omega$ .

Αυτός ο αλγόριθμος, είναι ευσταθής (επιτυγχάνει *throughput* ίσο με  $1/e \approx 0.368$ ) και απαιτεί από όλους τους χρήστες, να κάνουν χρήση της ίδιας πιθανότητας αναμετάδοσης σε κάθε χρονικό τμήμα ( $p_{\omega} = 1/\tilde{n}_{\omega}$ , για το χρονικό τμήμα  $\omega$ ). Μια που ο συνολικός ρυθμός άφιξης "φρέσκων" πακέτων  $\lambda$  δεν είναι γνωστός στους χρήστες, χρησιμοποιείται όμως στον αναδρομικό αλγόριθμο, πρέπει και αυτός να εκτιμηθεί. Δεδομένου όμως ότι το σύστημα είναι ευσταθές, ο συνολικός ρυθμός άφιξης πακέτων  $\lambda$  είναι ίσος με το ρυθμό επιτυχημένων μεταδόσεων πακέτων. Συνεπώς η εκτίμηση του  $\lambda$  από κάθε χρήστη είναι εύκολη, παρατηρώντας τις επιτυχημένες μεταδόσεις πακέτων στο κανάλι.

Αυτός ο αλγόριθμος και οι άλλοι σταθεροποιημένοι αλγόριθμοι τύπου *ALOHA*, απαιτούν οι χρήστες να παρακολουθούν την ανάδραση του καναλιού σε κάθε χρονικό τμήμα (συνεχής αίσθηση της ανάδρασης), και να προσαρμόζουν τις εκτιμήσεις τους βασισμένοι στην πιο πρόσφατη πληροφορία ανάδρασης που έχουν λάβει από το κανάλι. Η ανάδραση μπορεί να περιέχει δυαδική πληροφορία ("σύγκρουση", ή "μη σύγκρουση"), ή τριαδική πληροφορία ("σύγκρουση", "μη σύγκρουση", ή "κενό"), ή κάποια άλλη πληροφορία που δίνεται στους χρήστες από τον κεντρικό σταθμό (δέκτη). Βασισμένοι σ' αυτή την ανάδραση, όλοι οι χρήστες σε κάθε χρονικό τμήμα καθορίζουν την παράμετρο  $p$  του αλγορίθμου στην ίδια τιμή. Είναι φανερό ότι αυτού του είδους η λειτουργία μπορεί να θεωρηθεί ως αντιβαίνουσα στη φιλοσοφία της ατομικότητας του χρήστη στον αλγόριθμο *ALOHA*.

Μια που ο αλγόριθμος *ALOHA*, στην απλή μορφή λειτουργίας που είχε αρχικά προταθεί, είναι ασταθής, το εύλογο ερώτημα είναι αν είναι δυνατός ο σχεδιασμός ευσταθούς αλγορίθμου τυχαίας προσπέλασης, ο οποίος να πετυχαίνει *throughput* μεγαλύτερο από αυτό του ευσταθούς *ALOHA* που συζητήθηκε προηγουμένα. Η πρώτη αναφορά ευσταθούς αλγορίθμου τυχαίας προσπέλασης για τη τυχαία γεννήτρια "φρέσκων" πακέτων κατανεμημένη σύμφωνα με την κατανομή *Poisson*, παρουσιάστηκε ανεξάρτητα στις [7] και



[8]. Σ' αυτές τις δημοσιεύσεις το κανάλι υποτίθεται χρονικά σύγχρονο και η ανάδραση πληροφορο-ρεί όλους τους χρήστες του συστήματος εάν το χρονικό τμήμα περιείχε μηδέν, μία, ή περισσότερες από μία μεταδόσεις πακέτων (τριαδική ανάδραση, "σύγκρουση" / "επιτυχία" / "κενό"). Η δράση των χρηστών εξαρτάται μόνο από την λαμβανόμενη ανάδραση. Επίσης, κάθε χρήστης πρέπει να παρατηρεί το κανάλι ανάδρασης συνεχώς ανεξάρτητα εάν έχει πακέτο να μεταδώσει, από την αρχή της λειτουργίας του συστήματος (συνεχής αίσθηση της ανάδρασης). Ο προτεινόμενος αλγόριθμος μπορεί να λειτουργεί ακόμη και αν η ανάδραση είναι δυαδική (πληροφορώντας αν το χρονικό τμήμα περιέχει περισσότερες από μία μεταδόσεις πακέτων ή όχι, ή με άλλα λόγια ανάδραση τύπου "σύγκρουσης" / "μη σύγκρουσης" ) και μπορεί να αποδειχτεί ότι είναι ευσταθής εάν  $\lambda < 0.345$  [7] και [8] (ο αλγόριθμος αυτός χρησιμοποιεί τον *προφανή κανόνα πρώτης μετάδοσης*, οι λεπτομέρειες του οποίου θα εξηγηθούν παρακάτω). Μια τροποποίηση αυτού του αλγόριθμου αυξάνει το μέγιστο *throughput* στο 0.42 [11]. Υπό τριαδική ανάδραση, έχουν σχεδιαστεί αλγόριθμοι με μέγιστο *throughput* 0.487 [9] και [10] (οι αλγόριθμοι αυτοί χρησιμοποιούν τον *μη προφανή κανόνα πρώτης μετάδοσης*, οι λεπτομέρειες του οποίου θα εξηγηθούν επίσης παρακάτω). Λεπτομέρειες για τους παραπάνω αλγορίθμους παρουσιάζονται παρακάτω σε αυτό το Κεφάλαιο. Έχει όμως αποδειχτεί ότι παρουσία σφαλμάτων στην ανάδραση (εξαιτίας, για παράδειγμα, θορύβου στο κανάλι), οι αλγόριθμοι που χρησιμοποιούν τριαδική ανάδραση γίνονται ασταθείς [11], ενώ αυτοί που βασίζονται σε δυαδική ανάδραση συνεχίζουν να είναι ευσταθείς, επιτυγχάνονται όμως μειωμένα *throughputs*, εξαιτίας των σφαλμάτων στην ανάδραση [4] και [11].

Η απαίτηση της συνεχούς παρατήρησης της ανάδρασης είναι συνήθως μη ρεαλιστική. Στην πράξη, ένας χρήστης είναι ενεργός μόνο όταν έχει πακέτο να μεταδώσει. Είναι συνεπώς επιθυμητό, να σχεδιάζονται αλγόριθμοι οι οποίοι απαιτούν ότι κάθε χρήστης πρέπει να παρατηρεί την ανάδραση μόνο όταν είναι ενεργός. Οι αλγόριθμοι αυτοί ονομάζονται **αλγόριθμοι περιορισμένης αίσθησης της ανάδρασης** (Limited Feedback Sensing RAAs). Οι αλγόριθμοι τυχαίας προσπέλασης, με συνεχή ή περιορισμένη αίσθηση της ανάδρασης, χρησιμοποιούν περισσότερη πληροφορία ανάδρασης από αυτήν που χρησιμοποιείται με τον αλγόριθμο *ALOHA*. Υπενθυμίζεται ότι ο αλγόριθμος *ALOHA* απαιτεί κάθε χρήστη να παρατηρεί μόνο την ανάδραση που αντιστοιχεί στις δικές του μεταδόσεις.

### 1.3 Αλγόριθμοι επίλυσης συγκρούσεων

Η πρώτη μεγάλη καινοτομία στο πεδίο των αλγορίθμων πολλαπλής τυχαίας προσπέλασης μετά την σχεδίαση του αλγορίθμου *ALOHA*, ήρθε το 1977 με την εισαγωγή από τον Καπετανάκη της ιδέας των **αλγορίθμων επίλυσης συγκρούσεων** [7]. Ο Καπετανάκης πρότεινε την χρήση αλγορίθμου επίλυσης συγκρούσεων, που καταλήγει σε ευσταθές σύστημα τυχαίας προσπέλασης, όταν ο συνολικός ρυθμός άφιξης των πακέτων δεν είναι πολύ υψηλός. Το ενδιαφέρον σ' αυτή την περίπτωση είναι η ύπαρξη απλών αλγορίθμων επίλυσης συγκρούσεων πακέτων, που έχουν υψηλότερο *throughput* από το αντίστοιχο του ευσταθούς *ALOHA*.

Ένας αλγόριθμος επίλυσης συγκρούσεων μπορεί να οριστεί ως ένας αλγόριθμος (καταναμημένος στο χώρο και στο χρόνο) που οργανώνει τις αναμεταδόσεις των πακέτων που έχουν υποστεί σύγκρουση, με τέτοιο τρόπο ώστε τελικά κάθε πακέτο να μεταδίδεται επιτυχώς με πεπερασμένη καθυστέρηση και επιπλέον όλοι οι χρήστες (ή τουλάχιστον όλοι όσοι σχετίζονται με την επίλυση μιας σύγκρουσης) να ενημερώνονται γι' αυτό το γεγονός. Το χρονικό διάστημα, από το τμήμα χρόνου όπου συνέβη η αρχική σύγκρουση ως το τμήμα χρόνου στο οποίο όλοι οι χρήστες (ή τουλάχιστον αυτοί που έχουν σχέση με την επίλυση σύγκρουσης) αναγνωρίζουν ότι όλα τα πακέτα που συμμετείχαν στην παραπάνω σύγκρουση ελήφθησαν σωστά (μεταδόθηκαν επιτυχώς), ονομάζεται **Διάστημα Επίλυσης Σύγκρουσης** (Collision Resolution Interval, CRI). Λέμε ότι ένας αλγόριθμος επίλυσης συγκρούσεων είναι Q-αδικός (όπου Q είναι ακέραιος μεγαλύτερος ή ίσος του 2) αν ο χρήστης, αφού συμμετάσχει σε μία σύγκρουση, πρέπει να επιλέξει πιθανοτικά ανάμεσα σε Q εναλλακτικές δράσεις [2].

Ένας αλγόριθμος τυχαίας προσπέλασης, αποτελείται από τον συνδυασμό ενός αλγορίθμου επίλυσης συγκρούσεων και του κανόνα πρώτης μετάδοσης που ελέγχει πότε οι χρήστες με "φρέσκα" πακέτα θα μεταδώσουν για πρώτη φορά στο κανάλι [4]. Είναι σημαντικό να σημειωθεί ότι, όπως και στο πρωτόκολλο *ALOHA*, η δράση του κάθε χρήστη δε βασίζεται στη γνώση της κατάστασης των άλλων χρηστών.

Ο πιο απλός αλγόριθμος τυχαίας προσπέλασης είναι ο **Αλγόριθμος Ελεύθερης Προσπέλασης** (Free-Access Algorithm, FAA), στον οποίο ένας χρήστης μεταδίδει ένα "φρέσκο" πακέτο στο πρώτο χρονικό τμήμα που ακολουθεί την άφιξη του. Οι υπόλοιποι αλγόριθμοι ονομάζονται **Αλγόριθμοι Δεσμευμένης Προσπέλασης** (Blocked-Access Algorithms, BAA). Ο προφανής αλγόριθμος δεσμευμένης προσπέλασης είναι αυτός στον οποίο ο χρήστης μετα-δίδει ένα "φρέσκο" πακέτο στο πρώτο χρονικό τμήμα που ακολουθεί

την επίλυση όλων των συγκρούσεων που συνέβησαν πριν την άφιξη του συγκεκριμένου πακέτου. Οι υπόλοιποι αλγόριθμοι χαρακτηρίζονται ως μη προφανείς.

Στον αλγόριθμο ελεύθερης προσπέλασης, ο οποίος είναι γνωστός και ως *αλγόριθμος στοίβας* (stack) [2]-[4] και [8], τα “φρέσκα” πακέτα μεταδίδονται στην αρχή του πρώτου χρονικού τμήματος που ακολουθεί το χρονικό τμήμα της άφιξης τους, δηλαδή ακόμη και κατά τη διάρκεια εν εξελίξει επιλύσεων συγκρούσεων.

Στον αλγόριθμο δεσμευμένης προσπέλασης, που είναι επίσης γνωστός ως αλγόριθμος έρευνας δέντρου (tree search) [7], κανένα “φρέσκο” πακέτο δεν επιτρέπεται να εισέλθει στο σύστημα κατά το χρονικό διάστημα που διαρκεί μια επίλυση σύγκρουσης.

Ο βασικός μηχανισμός επίλυσης συγκρούσεων των δύο αλγορίθμων είναι κοινός. Μετά από κάθε σύγκρουση, κάθε χρήστης που συμμετείχε ρίχνει ένα νόμισμα με  $Q$  πλευρές με τιμές  $1, 2, \dots, Q$  (τα  $Q$ -αδικά αυτά νομίσματα δεν απαιτείται να είναι δίκαια, υποθέτουμε όμως ότι τα νομίσματα όλων των χρηστών πολώνονται με τον ίδιο τρόπο, όποτε χρησιμοποιείται πόλωση). Μ' αυτό τον τρόπο χωρίζονται οι αρχικοί χρήστες που συμμετείχαν στη σύγκρουση σε  $Q$  υποσύνολα. Σε καθένα από αυτά τα υποσύνολα, δίνεται η τιμή που έφερε το νόμισμα σαν δείκτης (κάποια από αυτά τα υποσύνολα μπορεί να είναι κενά). Οι χρήστες που φέρνουν  $i$  στην ρίψη του νομίσματος τους (δηλαδή ανήκουν στο  $i$ -οστο υποσύνολο),  $i = 0, 1, \dots, Q-1$ , μεταδίδουν αφού καθοριστεί ότι όλοι οι χρήστες στα υποσύνολα  $j < i$  έχουν μεταδώσει επιτυχώς. Οι χρήστες που δεν συμμετείχαν στην παρούσα σύγκρουση, αλλά οι οποίοι έχουν ήδη θετική τιμή δείκτη εξαιτίας συμμετοχής τους σε προηγούμενη σύγκρουση που δεν έχει ακόμη επιλυθεί, αυξάνουν τον δείκτη τους κατά  $Q-1$  (ώστε να δημιουργήσουν χώρο στην στοίβα, για τους χρήστες της παρούσας σύγκρουσης). Οι χρήστες που έχουν δείκτη μηδέν, μαζί με τους χρήστες που έχουν “φρέσκα” πακέτα - εφόσον ο αλγόριθμος είναι ελεύθερης προσπέλασης - μεταδίδουν στο αμέσως επόμενο χρονικό τμήμα. Εάν η ανάδραση αυτού του χρονικού τμήματος είναι “όχι σύγκρουση”, τότε όλοι οι χρήστες στους οποίους έχει ανατεθεί δείκτης μεγαλύτερος ή ίσος της μονάδας, μειώνουν αυτόν τον δείκτη κατά ένα. Συνεπώς, εάν μετά από μία αρχική σύγκρουση (χωρίς προηγούμενη ανάθεση δεικτών) δύο, ή περισσότερων πακέτων, καθένα από τα  $Q$  υποσύνολα που προκύπτουν περιέχει το πολύ έναν χρήστη, τότε η σύγκρουση θα επιλυθεί ακριβώς μετά από  $Q$  χρονικά τμήματα, που ακολουθούν την αρχική σύγκρουση (υποθέτοντας ότι δεν παρεμβάλλονται “φρέσκα” πακέτα). Δηλαδή, το διάστημα επίλυσης σύγκρουσης θα έχει διάρκεια ίση με  $Q+1$  χρονικά τμήματα. Η διάρκεια του χρονικού διαστήματος επίλυσης σύγκρουσης όταν μηδέν ή ένα πακέτα

μεταδίδονται είναι ίση με ένα χρονικό τμήμα. Ο βασικός αλγόριθμος επίλυσης συγκρούσεων χρησιμοποιεί δυαδική ανάδραση καναλιού.

Στην τυχαία προσπέλαση με αλγόριθμους επίλυσης συγκρούσεων πακέτων βασισμένων στην έρευνα δέντρου ή στην στοιβάδα, κάθε χρήστης μπορεί να ανιχνεύσει τον τερματισμό των διαστημάτων επίλυσης συγκρούσεων παρατηρώντας την ανάδραση του καναλιού. Τη στιγμή τερματισμού του διαστήματος επίλυσης σύγκρουσης, όλα τα πακέτα που είχαν συγκρουστεί στην αρχή και κατά τη διάρκεια του διαστήματος αυτού (εφόσον χρησιμοποιούμε αλγόριθμο ελεύθερης προσπέλασης) μεταδόθηκαν επιτυχώς. Αντίθετα, όταν χρησιμοποιείται το πρωτόκολλο *ALOHA*, η ανίχνευση από τους χρήστες του τερματισμού των διαστημάτων επίλυσης σύγκρουσης παρατηρώντας την ανάδραση του καναλιού είναι αδύνατη.

Σύμφωνα με τους αλγορίθμους διαχωρισμού, όλοι οι ενεργοί χρήστες (χρήστες με πακέτα προς μετάδοση), που συμμετέχουν σε μια διαδικασία επίλυσης σύγκρουσης, καθορίζουν την δράση τους σε ένα χρονικό τμήμα βασιζό-μενοι στην πληροφορία ανάδρασης που λαμβάνουν για το προηγούμενο χρονικό τμήμα. Παρατηρούμε ότι αν και όλοι οι παραπάνω χρήστες είναι εν γένει ενεργοί σε παραπάνω από ένα χρονικό τμήμα της διάρκειας επίλυσης της συγκεκρι-μένης σύγκρουσης, δεν μεταδίδουν όλοι μαζί σε κάθε χρονικό τμήμα. Αντίθετα, παρακολουθώντας την ανάδραση του καναλιού και εφαρμόζοντας τον αλγόριθμο επίλυσης σύγκρουσης κάνουν μια συλλογική προσπάθεια, η οποία εγγυάται ένα ευσταθές σύστημα. Οι αλγόριθμοι διαχωρίζονται παραπέρα με βάση τον κανόνα πρώτης μετάδοσης “φρέσκων” πακέτων, και τον τύπο της ανάδρασης καναλιού που χρησιμοποιούν.

Κανόνες Πρώτης Μετάδοσης. Οι αλγόριθμοι δεσμευμένης προσπέλασης έρευνας δέντρου κατηγοριοποιούνται σύμφωνα με τον κανόνα πρώτης μετάδοσης. Ο *προφανής κανόνας πρώτης μετάδοσης* υποχρεώνει όλους τους χρήστες που γέννησαν “φρέσκα” πακέτα κατά τη διάρκεια ενός διαστήματος επίλυσης σύγκρουσης, να μεταδώσουν αυτά τα πακέτα για πρώτη φορά αμέσως μόλις τελειώσει το διάστημα αυτό. Σ’ ένα πλήρως καταναμημένο σύστημα, οι χρήστες μπορούν να καταλάβουν το τέλος των διαστημάτων επίλυσης σύγκρουσης, μόνο εάν παρατηρούν την ανάδραση του καναλιού από την αρχή της λειτουργίας του συστήματος (ακόμη και αν δεν έχουν να μεταδώσουν πακέτο!). Αυτή η απαίτηση μπορεί να παραληφθεί αν υπάρχει κεντρικός σταθμός - δέκτης. Ο σταθμός αυτός, μπορεί να παρατηρεί συνεχώς την ανάδραση του καναλιού και να δίνει σήμα στους χρήστες μετά το τέλος κάθε διαστήματος επίλυσης σύγκρουσης [1].

Με τον προφανή κανόνα πρώτης μετάδοσης, ο αριθμός των πακέτων που μεταδίδονται στο πρώτο χρονικό τμήμα ενός διαστήματος επίλυσης σύγκρουσης (καθώς και η διάρκεια αυτού του διαστήματος) σχετίζονται άμεσα με τη διάρκεια του προηγούμενου διαστήματος επίλυσης σύγκρουσης. Έτσι, μεγάλα διαστήματα επίλυσης σύγκρουσης ακολουθούνται, με μεγάλη πιθανότητα, από επίσης μεγάλα διαστήματα επίλυσης σύγκρουσης, φαινόμενο το οποίο εάν ισχύει για πολύ χρόνο θα έχει ως αποτέλεσμα την σημαντική πτώση της απόδοσης του συστήματος. Η απόδοση αναμένεται ότι μπορεί να αυξηθεί - και όντως αυξάνεται - εάν ο χρονικός άξονας αφίξεων αποξευχθεί από τον χρονικό άξονα μετάδοσης.

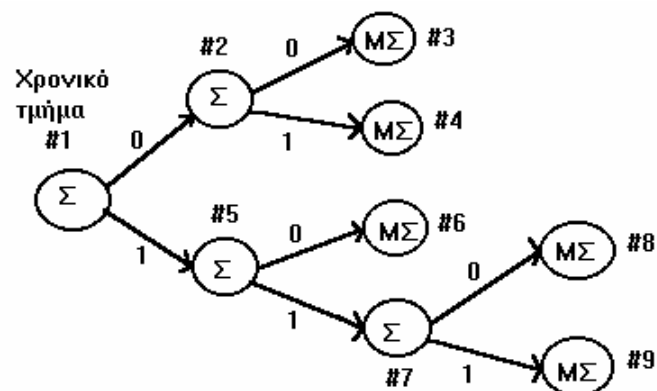
Αυτό κάνει ο μη προφανής κανόνας πρώτης μετάδοσης, σύμφωνα με τον οποίο μόνο τα πακέτα που φτάνουν κατά τη διάρκεια ενός χρονικού διαστήματος μεγίστου μήκους  $\Delta$  επιτρέπεται να μεταδοθούν στο πρώτο χρονικό τμήμα που ακολουθεί ένα διάστημα επίλυσης σύγκρουσης. Έτσι, εάν το τελευταίο διάστημα επίλυσης σύγκρουσης έχει μήκος μεγαλύτερο από  $\Delta$ , μόνο οι χρήστες που έχουν γεννήσει πακέτα στο αριστερό τμήμα μήκους  $\Delta$  του παραπάνω διαστήματος θα μεταδώσουν στο πρώτο χρονικό τμήμα που ακολουθεί το διάστημα επίλυσης σύγκρουσης. Η παράμετρος  $\Delta$  επιλέγεται ώστε ο αλγόριθμος να επιτυγχάνει το μέγιστο δυνατό *throughput*.

Δυαδική έναντι τριαδικής ανάδρασης. Ένα παράδειγμα ενός αλγορίθμου έρευνας δυαδικού δέντρου, με δυαδική ανάδραση, φαίνεται στο **Σχ.2**. Ακολουθώντας καθεμία σύγκρουση, τα πακέτα που έχουν συγκρουστεί διαχωρίζονται τυχαία ανάμεσα στους κόμβους του δέντρου, που έχει σαν ρίζα το χρονικό τμήμα της παρούσας μετάδοσης. Στο **Σχ.3** φαίνεται ένας αλγόριθμος ελεύθερης προσπέλασης στοίβας (με παράμετρο διαχωρισμού  $Q=2$ ) με δυαδική ανάδραση. Ακολουθώντας καθεμιά σύγκρουση, τα πακέτα που έχουν συγκρου-στεί διαμερίζονται τυχαία ανάμεσα στα πρώτα  $Q$  κελιά της στοίβας, σπρώχνοντας τα πακέτα που υπάρχουν ήδη στη στοίβα προς τα πάνω. Το δέντρο στο **Σχ.2**, καθώς και η στοίβα στο **Σχ.3** είναι απλά χρήσιμοι μηχανισμοί για την κατανόηση από εμάς της λειτουργίας των αλγορίθμων και δεν χρειάζεται να συνειδητοποιηθούν από τους χρήστες του δικτύου. Οι χρήστες, για παράδειγμα, χρειάζεται να έχουν στη διάθεση τους, μόνο ένα ακέραιο μετρητή για να εφαρμόσουν τον αλγόριθμο στοίβας [11].

Η αναζήτηση των πακέτων που έχουν συγκρουστεί σε έναν αλγόριθμο έρευνας δέντρου, γίνεται κατά μήκος των κλάδων του δέντρου και σε καθεμιά σύγκρουση το δέντρο διαιρείται για μια ακόμη φορά. Σ' έναν αλγόριθμο στοίβας, η έρευνα γίνεται παρατηρώντας τα περιεχόμενα των κελιών της στοίβας. Το πρώτο κελί, κελί αριθμός μηδέν, είναι το κελί μετάδοσης. Μέσω του ίδιου κελιού,

τα "φρέσκα" πακέτα εισάγονται, για πρώτη φορά, στο σύστημα. Και στις δύο περιπτώσεις, μετά από κάθε διαχωρισμό που ακολουθεί μια σύγκρουση, κάθε υποσύνολο περιέχει τη σειρά του να μεταδώσει, αφού το προηγούμενο υποσύνολο τελειώσει τις μεταδόσεις του.

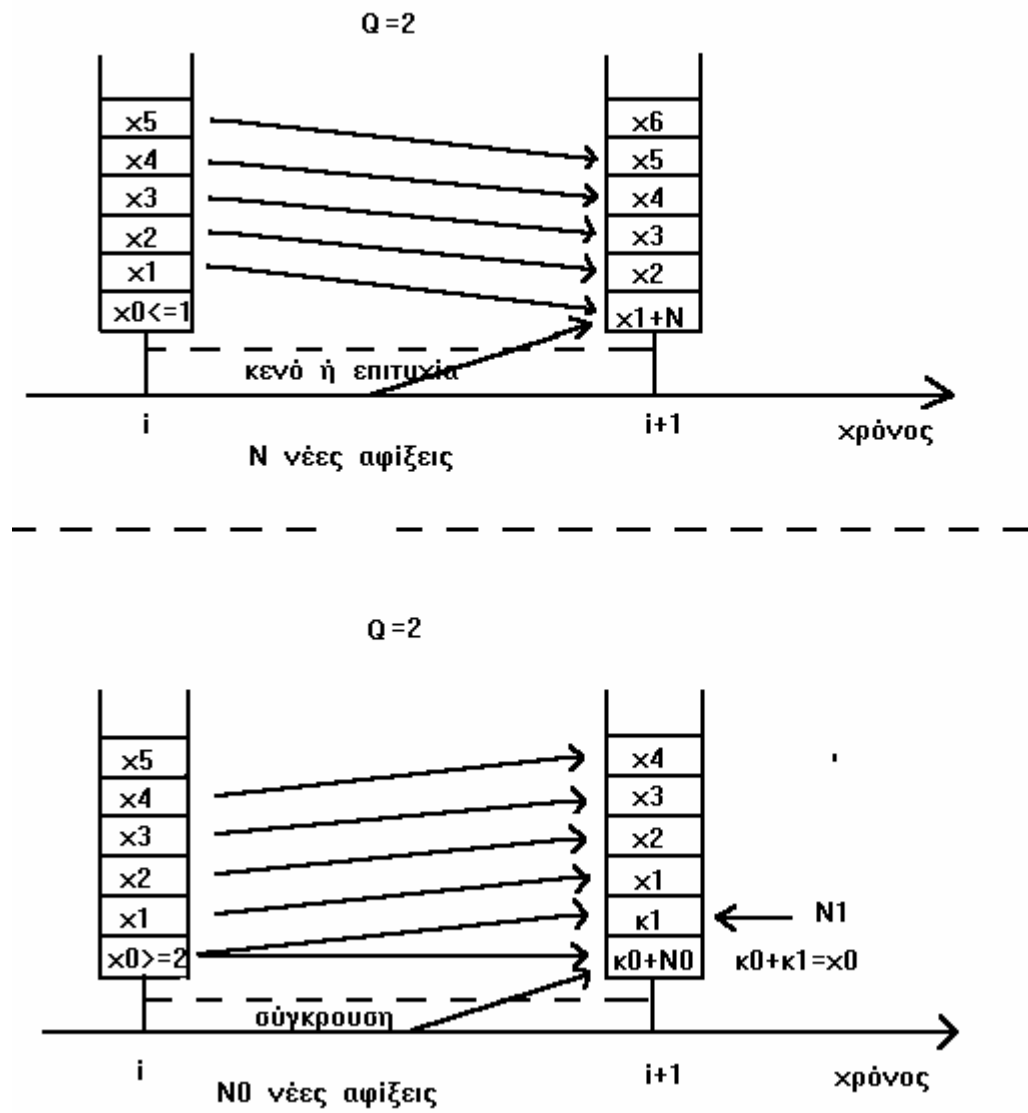
Εάν η επίλυση σύγκρουσης άρχισε με σύγκρουση και τα επόμενα (Q-1) χρονικά τμήματα είναι κενά (υποθέτουμε ότι η ανάδραση του καναλιού είναι τριαδική, ώστε η ανάδραση να μπορεί να πληροφορήσει τους χρήστες εάν ένα



**Σχ.2** : Ένα δυαδικό δέντρο έρευνας του αλγόριθμου επίλυσης συγκρούσεων με πληροφορία ανάδρασης δυαδική  $\Sigma/\text{Μ}\Sigma$  (Σύγκρουση / Μη Σύγκρουση), με διάστημα επίλυσης της σύγκρουσης ίσο με 9 χρονικά τμήματα.

χρονικό τμήμα είναι κενό), είναι σίγουρο ότι το ακόλουθο χρονικό τμήμα (Q-οστό) θα περιέχει σύγκρουση (διότι όλα τα αρχικά συγκρουόμενα πακέτα έφεραν το ίδιο αποτέλεσμα, Q-1, στην ρίψη των νομισμάτων τους). Για παράδειγμα, στο **Σχ.2**, αν το χρονικό τμήμα #6, δοκιμαστεί και βρεθεί κενό, τότε είναι βέβαιο ότι το χρονικό τμήμα #7 θα περιέχει σύγκρουση (οι χρήστες το συνάγουν αυτό, πριν ακόμη γίνει η μετάδοση στο χρονικό τμήμα #7). Αυτή η σίγουρη σύγκρουση, μπορεί να αποφευχθεί εφόσον ο αλγόριθμος ζητήσει από τους συγκρουόμενους στη χρονική μονάδα #5 χρήστες, να χρησιμοποιήσουν τον τυχαίο χρονικό δια-χωρισμό, για άλλη μια φορά πριν την αρχή του χρονικού τμήματος #7. Έτσι, τα χρονικά τμήματα που εξακριβωμένα θα περιέχουν σύγκρουση μπορούν να αποφευχθούν. Με παρόμοιο τρόπο, μπορούν να κατασκευαστούν και αλγόριθμοι στοίβας βασισμένοι σε τριαδική ανάδραση.

Αν και η χρήση τριαδικής ανάδρασης έχει σαφή πλεονηκτήματα, μια και οδηγεί σε μεγαλύτερα *throughputs*, η εφαρμογή της είναι λιγότερο αξιόπιστη στην περίπτωση παρουσίας λαθών στην ανάδραση που παρέχει το κανάλι. Για παράδειγμα, αν λόγω της παρουσίας λάθους στην ανάδραση σε έναν αλγόριθμο έρευνας δέντρου, ένα κενό χρονικό τμήμα μέσα σε ένα διάστημα επίλυσης σύγκρουσης εκληφθεί ως σύγκρουση, τότε ο αλγόριθμος μπαίνει σε αδιέξοδο (δηλαδή δεν τερματίζει ποτέ) [4].



Σχ.3 : Μία στοίβα του αλγόριθμου επίλυσης συγκρούσεων με πληροφορία ανάδρασης δυαδική και παράμετρο διαχωρισμού  $Q=2$ .

Ελεύθερη έναντι Δεσμευμένης Προσπέλασης. Στον αλγόριθμο ελεύθερης προσπέλασης, η ελεύθερη είσοδος των “φρέσκων” πακέτων στο κανάλι μπορεί να προκαλέσει την αύξηση των ανταγωνιζόμενων χρηστών κατά τη διάρκεια επίλυσης μίας σύγκρουσης, κάτι που δεν είναι δυνατόν να συμβεί στην περίπτωση του αλγορίθμου δεσμευμένης προσπέλασης. Από πρακτικής πλευράς η μόνη διαφορά μεταξύ των παραπάνω αλγορίθμων είναι ότι στον δεύτερο κάθε χρήστης πρέπει να διατηρεί τη γενική μεταβλητή βάθους (της στοίβας ή του δέντρου) καθώς και να παρατηρεί την ανάδραση του καναλιού χωρίς διακοπή, ενώ στον πρώτο, απαιτείται η παρατήρηση της ανάδρασης του καναλιού μόνο όσο ο χρήστης έχει πακέτο το οποίο δεν έχει μεταδοθεί ακόμη επιτυχώς. Επίσης, οι αλγόριθμοι ελεύθερης προσπέλασης δεν απαιτούν από τους χρήστες να ανιχνεύουν το τέλος των χρονικών διαστημάτων επίλυσης συγκρούσεων, η υλοποίησή τους είναι απλή και απαιτείται μόνο έναν μετρητή για κάθε χρήστη ο οποίος ενημερώνεται σύμφωνα με την ανάδραση του καναλιού η οποία λαμβάνεται σε κάθε χρονικό τμήμα. Οι παραπάνω αλγόριθμοι λειτουργούν αποδοτικά χρησιμοποιώντας την σθεναρή (robust) δυαδική ανάδραση, “σύγκρουση” / “μη σύγκρουση”, επιτυγχάνουν ευσταθή λειτουργία, με *throughput* το οποίο είναι υψηλότερο από αυτό οποιουδήποτε άλλου αλγορίθμου τυχαίας προσπέλασης, με εξαίρεση αυτό των αλγορίθμων μπλοκαρισμένης προσπέλασης έρευνας δέντρου. Οι τελευταίοι όμως αλγόριθμοι χρησιμοποιούν τον δύσκολα υλοποιήσιμο μη προφανή κανόνα πρώτης μετάδοσης.

#### **1.4 Περιγραφή Διπλωματικής**

Το Κεφάλαιο 2 αποτελεί μια εισαγωγή στην εκρηκτική κατανομή *Pareto* όπου αναφέρονται οι λόγοι για τους οποίους αυτή η κατανομή χρησιμοποιείται σε αυτή την εργασία για να χαρακτηρίσει τους χρόνους μεταξύ αφίξεων διαδοχικών πακέτων στο σύστημα μας. Στο Κεφάλαιο 3 τροποποιούμε ορισμένα χαρακτηριστικά του ιδανικού ευσταθούς αλγορίθμου *ALOHA*, ώστε να μπορεί να χειριστεί κατά τον καλύτερο δυνατό τρόπο αφίξεις πακέτων που χαρακτηρίζονται από *Pareto* κατανεμημένους χρόνους μεταξύ διαδοχικών αφίξεων, μελετούμε το μέγιστο *throughput* και χαρακτηριστικά της κατανομής της καθυστέρησης πακέτων που εισάγει ο αλγόριθμος. Το Κεφάλαιο 4 μελετά το μέγιστο *throughput* και τα χαρακτηριστικά καθυστέρησης πακέτων δύο αλγορίθμων ελεύθερης προσπέλασης στοίβας για *Pareto* κατανεμημένους χρόνους μεταξύ αφίξεων. Στο Κεφάλαιο 5 γίνεται μία σύγκριση της καθυστέρησης των αλγορίθμων που αναλύονται στα προηγούμενα Κεφάλαια για χρόνους μεταξύ αφίξεων πακέτων κατανεμημένους σύμφωνα με τις κατανομές



*Pareto* και Εκθετική (*Poisson* αφίξεις). Τα συμπεράσματα αυτής της εργασίας περιλαμβάνονται στο Κεφάλαιο 6.

## ΚΕΦΑΛΑΙΟ 2

### Η Εκρηκτική κατανομή Pareto

Στο Κεφάλαιο αυτό παραθέτουμε εισαγωγή στην κατανομή Pareto και στις κυριότερες ιδιότητες της. Το υλικό του Κεφαλαίου δανείζεται αρκετά στοιχεία από την εκτεταμένη σχετική συζήτηση στο [13].

#### **2.1 Η αποτυχία της κατανομής *Poisson* να μοντελοποιήσει ικανοποιητικά ορισμένες εφαρμογές μετάδοσης πακέτων δεδομένων.**

Οι αφίξεις πακέτων ή μηνυμάτων πληροφορίας στα δίκτυα, συχνά μοντελοποιούνται με τη διαδικασία *Poisson* για αναλυτική ευκολία, ακόμη και όταν η μελέτη της κίνησης (δηλαδή, των αφίξεων πακέτων και/ή μηνυμάτων πληροφορίας) έχει δείξει ότι οι χρόνοι μεταξύ διαδοχικών αφίξεων δεν είναι εκθετικά κατανομημένοι. Οι αφίξεις πακέτων σε εφαρμογές που υλοποιούνται πάνω σε δίκτυα ευρείας γεωγραφικής περιοχής (όπως για παράδειγμα, της διαδικασίας αφίξεων **TCP** που περιλαμβάνει αφίξεις περιόδου και συνδέσεων, αφίξεις δεδομένων **FTP** μέσα σε περιόδους FTP και αφίξεις πακέτων της

εφαρμογής **TELNET**), η από απόσταση εισαγωγή σε δίκτυο (remote login) και η μεταφορά αρχείων (file transfer)), μοντελοποιούνται ικανοποιητικά με τη διαδικασία **Poisson** μόνο όταν ο ρυθμός αφίξεων πακέτων της κατανομής θεωρηθεί μεταβαλλόμενος με το χρόνο και καθορίζεται κατάλληλα ανά ώρα. Για παράδειγμα, έχει βρεθεί ότι η μοντελοποίηση των χρόνων μεταξύ αφίξεων διαδοχικών πακέτων στην εφαρμογή TELNET με την εκθετική κατανομή σταθερού ρυθμού (δηλαδή, **Poisson** αφίξεις σταθερού ρυθμού) , υποεκτιμά την εκρηκτικότητα που παρατηρείται σε αυτήν την διαδικασία αφίξεων πακέτων [13].

Όταν μοντελοποιούμε την κίνηση ενός δικτύου, συχνά υποθέτουμε ότι οι αφίξεις των πακέτων είναι **Poisson**, για τον επιπρόσθετο λόγο ότι αυτή η διαδικασία έχει μελετηθεί εκτενώς και γνωρίζουμε καλά τη συμπεριφορά της. Σύμφωνα με το [13], η κίνηση σε **Τοπικό Δίκτυο** (Local Access Network, LAN) μοντελοποιείται καλύτερα χρησιμοποιώντας στατιστικά *αυτό-όμοιες (self-similar)* διαδικασίες [13], [27], οι οποίες έχουν πολύ διαφορετικές ιδιότητες από την **Poisson** (ο ορισμός των αυτό-όμοιων στοχαστικών διαδικασιών δίδεται στην Παράγραφο 2.2.2 του Κεφαλαίου). Για μία κίνηση δικτύου που χαρακτηρίζεται από μία στατιστικά *αυτό-όμοια* διαδικασία, δεν υπάρχει καθορισμένο μήκος για μία “έκρηξη” (όπου ως “έκρηξη” ορίζεται μία συνέχεια αφίξεων πακέτων, που ακολουθείται από μία ανενεργή περίοδο - όπου κανένα πακέτο δεν γεννιέται στο σύστημα), αλλά το μήκος της “έκρηξης” μπορεί να μεταβάλλεται σε μια ευρεία χρονική κλίμακα (οι αυτό-όμοιες διαδικασίες χαρακτηρίζονται συνεπώς, από υψηλότερη εκρηκτικότητα πάνω σε μία ευρεία χρονική κλίμακα).

Η διαδικασία **Poisson** έχει βρεθεί ότι είναι ικανοποιητική μόνο για τον χαρακτηρισμό των αφίξεων στον κάθε μεμονωμένο χρήστη (π.χ., μεμονωμένες συνδέσεις TELNET ή συνδέσεις “ελέγχου” FTP) αλλά αποτυγχάνει να μοντελοποιήσει ικανοποιητικά άλλες διαδικασίες αφίξεων πακέτων, οι οποίες προκύπτουν από πολυπλεξία ή συννένωση πακέτων μεμονομένων συνδέσεων σε δίκτυα ευρείας γεωγραφικής περιοχής. Αυτές οι διαδικασίες αφίξεων φαίνεται να μοντελοποιούνται ικανοποιητικά με τη χρήση *στατιστικά αυτό-όμοιων* διαδικασιών. Για παράδειγμα, στο [13] έχει επίσης βρεθεί ότι η χρήση του μοντέλου **Poisson** με σταθερό ρυθμό αφίξεων (μέση τιμή), υποεκτιμάει σημαντικά την εκρηκτικότητα της κίνησης TCP όταν αυτή παρατηρείται πάνω σε μεγάλα χρονικά διαστήματα. Επίσης, οι εκθετικά κατανομημένοι χρόνοι μεταξύ αφίξεων που χρησιμοποιούνται για να μοντελοποιήσουν αφίξεις πακέτων που γεννιούνται από το χρήστη σε μια σύνδεση TELNET υποεκτιμάει την εκρηκτικότητα της κίνησης της σύνδεσης [13]. Ο υψηλός βαθμός πολυπλεξίας που χρησιμοποιεί η σύνδεση TELNET, οδηγεί σε κατανομή των χρόνων μεταξύ

αφίξεων διαδοχικών πακέτων πολύ διαφορετική από την εκθετική. Έχει επίσης βρεθεί ότι κατά τη διάρκεια μιας σύνδεσης TELNET (όχι όμως στη φάση της εγκαθίδρυσης της), οι χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων έχουν κατανομή *μακριάς ουράς (long tailed)*, έτσι ώστε η συνάρτηση ουράς της κατανομής να μειώνεται πιο αργά από οποιαδήποτε εκθετική συνάρτηση.

Για μεταδόσεις μικρής διάρκειας, όπως είναι το *ηλεκτρονικό ταχυδρομείο (e-mail, SMTP)* και *τα νέα του δικτύου (NNTP)*, οι αφίξεις πακέτων δεν μοντελο-ποιούνται ικανοποιητικά με **Poisson** κάτι που είναι αναμενόμενο αφού και οι δύο τύποι συνδέσεων είναι πολύ εκρηκτικοί, αρχικοποιούνται και ελέγχονται από το δίκτυο και οδηγούνται από χρονοστές. Η περιοδικότητα της κίνησης IP (Internet Packets) [31] που γεννιέται από το δίκτυο, όπως οι *περιοδικές ενημερώσεις των πινάκων δρομολόγησης (routing updates)*, μπορούν να έχουν ως αποτέλεσμα τις ταυτόχρονες αφίξεις / αναχωρήσεις πακέτων στο / από το δίκτυο, κάτι που δεν μπορεί να προβλεφθεί από το μοντέλο **Poisson** (η διαδικασία **Poisson** δεν επιτρέπει ταυτόχρονες αφίξεις πακέτων [30]).

Συνεπώς, η κίνηση δικτύου παρατηρούμενη σε μεγάλα χρονικά διαστήματα είναι πολύ πιο εκρηκτική από ότι προβλέπει το μοντέλο **Poisson**. Για το λόγο αυτό, το μοντέλο δεν προβλέπει σωστά περιπτώσεις συμφόρησης του δικτύου καθώς και την απόδοση λειτουργίας του. Συμπερασματικά, δεν μπορούμε να ελπίζουμε σε ικανοποιητική μοντελοποίηση των αφίξεων συνδέσεων χρησιμο-ποιώντας μια απλή ομογενή διαδικασία **Poisson**.

Μόνο οι αφίξεις πακέτων που εγκαθιδρύουν τις συνδέσεις FTP και TELNET είναι, υπό συνθήκες, στατιστικά συμβατές με αφίξεις **Poisson**, ενώ οι αφίξεις των πακέτων που εγκαθιδρύουν συνδέσεις FTPDATA (δηλαδή, οι συνδέσεις δεδομένων FTP μέσα σε μια περίοδο FTP, οι οποίες αρχικοποιούνται όποτε ο χρήστης καταγράφει έναν κατάλογο (directory), είτε κάνει μεταφορά ενός αρχείου), καθώς και τα πακέτα της εφαρμογής FTPDATA που ακολουθούν, δεν μοντελοποιούνται ικανοποιητικά από την κατανομή **Poisson**. Τέλος, οι αφίξεις των συνδέσεων NNTP και WWW (World Wide Web) δεν είναι συμβατές με την κατανομή **Poisson** [13].

Το κυρίως σώμα της παρατηρούμενης κατανομής των χρόνων μεταξύ αφίξεων πακέτων σε όλες σχεδόν τις προαναφερθείσες εφαρμογές μοντελοποιείται ικανοποιητικά από την κατανομή **Pareto** [12]. Επιπλέον παρατήρηση της ουράς της κατανομής του όγκου δεδομένων (σε bytes) ανά FTPDATA “έκρηξη”, οδηγεί στο συμπέρασμα ότι η κατανομή **Pareto** μοντελοποιεί ικανοποιητικά και την συμπεριφορά του επάνω 3-5% της ουράς της κατανομής των δεδομένων της FTPDATA σύνδεσης [13].

Οι ανεξάρτητες συνδέσεις TELNET μοντελοποιούνται ικανοποιητικά από την κατανομή *Pareto*, η οποία είναι μία μέθοδος γεννήσεων ψεύδο-αυτό-όμοιων αφίξεων πακέτων που μοιάζουν με αυτό-όμοιες μόνο πάνω σε μια περιοχή της κλίμακας του χρόνου. Συνεπώς, στις μικρότερες κλίμακες του χρόνου, στατιστική αυτό-ομοιότητα μπορεί να επέλθει από το γεγονός ότι οι αφίξεις πακέτων μέσα σε ανεξάρτητες συνδέσεις TELNET μοντελοποιούνται καλύτερα με την αμετάβλητη, ανεξάρτητη κατανομή *Pareto* παρά με την *Poisson*.

## 2.2 Η κατανομή *Pareto*

Η απόδοση των ψηφιακών κινητών δικτύων ολοκληρωμένων υπηρεσιών τρίτης γενιάς, επηρεάζεται προφανώς από τα πρωτόκολλα πολλαπλής προσπέ-λασης που χρησιμοποιούνται στα υποσυστήματα ασύρματης πρόσβασης [12]. Εάν και υπάρχουν ικανοποιητικά μοντέλα για την διαδικασία άφιξης των πακέτων μιας πηγής φωνής, που χρησιμοποιεί ανιχνευτή των χρονικών περιόδων της δραστηριότητας της - Voice Activity Detector [16], για μια πληθώρα εφαρμο-γών δεδομένων (όπως μεταφορά αρχείων, είσοδος σε απομακρυσμένο σύστημα και υποβολή ερωτήσεων σε βάση δεδομένων) δεν είναι εύκολα διαθέσιμο ένα ικανοποιητικό μοντέλο. Από εκτεταμένες έρευνες που έχουν γίνει πάνω στα τοπικά και στα ευρύτερα δίκτυα φάνηκε ότι η κατανομή του χρόνου μεταξύ αφίξεων πακέτων της συνολικής κίνησης διαφέρει εμφανώς από την εκθετική κατανομή [13], [17] και [18]. Επίσης, ο προσδιορισμός και η σχεδίαση ενός αλγορίθμου πολλαπλής προσπέλασης, για την υποστήριξη της δραστηριότητας των σημερινών εξελιγμένων κινητών συστημάτων επικοινωνίας [19] και [20] είναι απαραίτητο να βασίζεται σε πραγματικά χαρακτηριστικά της διαδικασίας κίνησης.

Για να μπορέσουμε να υπολογίσουμε τις δυνατότητες των **Συστημάτων Ασύρματης Προσπέλασης - Ολοκληρωμένων Υπηρεσιών** (Service Integrated Radio Access, **SIR**) να διαχειριστούν συνδυασμό κίνησης φωνής και δεδομένων, επιλέγουμε να “φορτώσουμε” το σύστημα με κίνηση δεδομένων που γεννιέται από *διαλογικές* εφαρμογές (π.χ., πρόσβαση σε βάση δεδομένων από απομακρυσμένο τερματικό). Η απόφαση αυτή υποκινήθηκε από **α)την αναμενόμενη σημασία αυτών των εφαρμογών στα μελλοντικά κινητά συστήματα ασύρματης προσπέλασης, και β)την εκρηκτικότητα των διαδικασιών αφίξεων πακέτων που χαρακτηρίζουν αυτές τις εφαρμογές.** Η σχεδίαση του αλγορίθμου προσπέλασης και η μελέτη της απόδοσης του κάτω από μία έντονα εκρηκτική διαδικασία αφίξεων πακέτων ισοδυναμεί με σχεδίαση και μελέτη απόδοσης υπό αυστηρές συνθήκες.

Το αναλυτικό μοντέλο άφιξης πακέτων που χρησιμοποιείται στην μελέτη μας [13], βασίζεται στην παρατήρηση της “ζωντανής” κίνησης TCP με έμφαση στη διαλογική

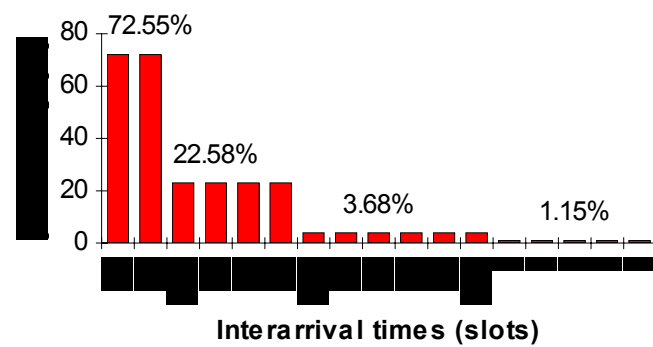
εφαρμογή TELNET πάνω σε ευρεία κλίμακα χρόνου. Αν και το περιβάλλον λειτουργίας για το παραπάνω σύστημα είναι διαφορετικό από το αντίστοιχο ενός ασύρματου κινητού συστήματος, συλλαμβάνει χαρακτηριστικά της συμπεριφοράς των χρηστών που σχετίζονται με τις διαλογικές εφαρμογές δεδομένων, οι οποίες αναμένεται να βρουν ευρεία εφαρμογή στα μελλοντικά ασύρματα συστήματα κινητής επικοινωνίας.

Ένα πλήρες μοντέλο κίνησης TELNET μπορεί να περιλαμβάνει κατανομές των αφίξεων μηνυμάτων μέσα σε μία χρονική περίοδο και του μεγέθους των μηνυμάτων (σε πακέτα). Η χρονική κλίμακα, πάνω στην οποία φυσιολογικά παρα-τηρούνται οι περίοδοι αφίξεων (της τάξεως των δέκα δευτερολέπτων) είναι πολύ μακριά από το χρονικό ορίζοντα που μας απασχολεί για τη μελέτη απόδοσης του πρωτοκόλλου Ασύρματης Προσπέλασης (της τάξεως των εκατό millisecond). Έτσι δεν απασχολούμαστε με τις πιθανές μεταβολές της διαδικασίας αφίξεων πακέτων κατά τη διάρκεια μιας χρονικής περιόδου και εξετάζουμε το πρωτόκολλο με καθορισμένο αριθμό από εγκατεστημένες μονάδες δεδομένων [12]. Ακολουθώντας την [13], οι χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων υποτίθεται ότι είναι ανεξάρτητες τυχαίες μεταβλητές κατανομημένες σύμφωνα με την κατανομή *Pareto* με παραμέτρους  $a$  και  $k$  :

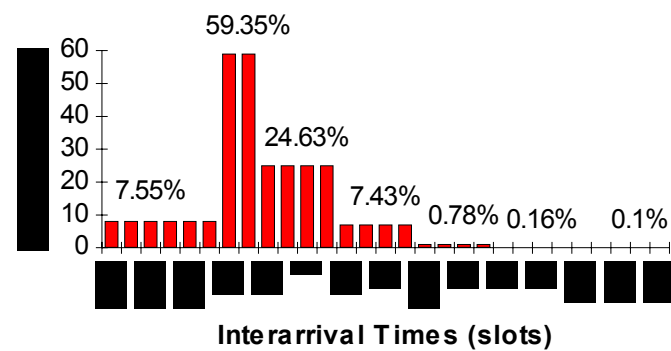
$$P\{T \leq t\} = 1 - [k/t]^a \quad k, a \geq 0, t \geq k.$$

Οι παράμετροι σ' αυτή την κατανομή είναι η *παράμετρος τοποθεσίας* (location parameter)  $k$ , που συμβολίζει τον ελάχιστο χρόνο μεταξύ αφίξεων δύο διαδοχικών πακέτων (στην παρούσα εργασία θεωρούμε  $0 < k < 1$ ) και η *παράμετρος σχήματος* (shape parameter)  $a$ , όπου αν  $a \leq 2$  η κατανομή έχει άπειρη διασπορά, ενώ για  $a \leq 1$  η μέση τιμή είναι επίσης άπειρη (δες Παράγραφο 2.2.1 αυτής της εργασίας). Έτσι η κατανομή έχει βαριά ουρά (heavy tailed) με άπειρη διασπορά και άπειρη μέση τιμή. Η παραπάνω μαθηματική ιδιότητα ευθύνεται για την έντονη εκρηκτικότητα της *Pareto* διαδικασίας αφίξεων πακέτων.

Τα σχήματα Σχ.4 και Σχ.5 παριστούν το ιστόγραμμα των χρόνων μεταξύ αφίξεων δύο διαδοχικών πακέτων για την κατανομή *Pareto*. Τα σχήματα αυτά προέκυψαν από προσομοίωση δέκα χιλιάδων δειγμάτων. Παρατηρούμε στο Σχ.4, όπου ο ελάχιστος χρόνος μεταξύ αφίξεων είναι 0.1 χρονικές μονάδες (slots), ότι



Σχ.4 : Η κατανομή της Pareto για  $a=1.95$ ,  $k=0.1$  .



Σχ.5 : Η κατανομή της Pareto για  $a=1.5$ ,  $k=0.95$  .

το 72.55% των δειγμάτων βρέθηκαν από την προσομοίωση στο διάστημα από 0.1 μέχρι 0.2 slot, το 22.58% των δειγμάτων στο διάστημα από 0.21 μέχρι 0.5 slot. Το μεγάλο ποσοστό των δειγμάτων που είναι συγκεντρωμένα στα δύο αυτά διαστήματα δείχνει ότι ένα σύστημα που χρησιμοποιεί την κατανομή *Pareto* με τις συγκεκριμένες παραμέτρους είναι αδύνατο να επιτύχει ευστάθεια αφού βομβαρδίζεται από συνεχείς αφίξεις πακέτων μέσα σε ένα slot. Να θυμηθούμε εδώ ότι η χρονική μονάδα slot έχει οριστεί σαν το χρόνο που χρειάζεται για την μετάδοση ενός πακέτου. Ενώ στο Σχ.5, όπου ο ελάχιστος χρόνος μεταξύ αφίξεων είναι 0.95 slots, παρατηρούμε ότι ένα ποσοστό 7.55% των δειγμάτων είναι ανάμεσα στο 0.95 και στο 1 slot, ενώ ο κύριος όγκος των δειγμάτων, 59.35%, βρίσκεται από 1 μέχρι 2 slots και το ποσοστό των δειγμάτων που οι χρόνοι μεταξύ αφίξεων είναι μεγαλύτεροι από 20 slots είναι μικρότερο του 1%. Από το σχήμα αυτό φαίνεται ότι σε μεγάλο ποσοστό εμφανίζεται το πολύ μία άφιξη πακέτου στο κάθε slot. Επίσης, η παρατηρούμενη συγκέντρωση μεγάλου ποσοστού των δειγμάτων στην περιοχή από 0.95 μέχρι 5 slots και η ύπαρξη κάποιου ποσοστού

δειγμάτων στα επόμενα διαστήματα δείχνει ότι η κατανομή **Pareto** είναι έντονα εκρηκτική, επειδή εμφανίζει κάποια χρονικά διαστήματα (slots) χωρίς πακέτα (δηλαδή, χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων μεγαλύτεροι της μονάδας) και στη συνέχεια εισάγει συνεχόμενα πακέτα στο σύστημα (δηλαδή, μικροί χρόνοι μεταξύ αφίξεων).

Αν και οι εφαρμογές που αναφέρονται στο Κεφάλαιο αυτό αφορούν ανταλλαγή πληροφοριών και προς τις δύο κατευθύνσεις, εμείς θα εξετάσουμε *κίνηση* μόνο προς τη μία κατεύθυνση (από τον κινητό χρήστη προς τη βάση (δέκτη)), και θα θεωρήσουμε ότι ο κινητός χρήστης ασχολείται μόνο με την μετάδοση δεδομένων. Το θέμα της μετάδοσης φωνής έχει εξεταστεί εκτενώς στις [14] και [15] και στις παραπομπές που βρίσκονται σε αυτές τις δημοσιεύσεις.

### 2.2.1 Ιδιότητες της κατανομής Pareto

Σε αυτό το τμήμα του Κεφαλαίου παραθέτουμε και σχολιάζουμε τις πιο σημαντικές ιδιότητες της κατανομής **Pareto**.

Η τυχαία μεταβλητή  $T$  κατανομημένη σύμφωνα με την κατανομή **Pareto** με παράμετρο σχήματος  $a$  και παράμετρο τοποθεσίας  $k$  έχει την ακόλουθη *Αθροιστική Συνάρτηση Πιθανότητας* [24] :

$$F(x) = P [T \leq x] = 1 - (k/x)^a, \quad k, a \geq 0, x \geq k,$$

με αντίστοιχη *Συνάρτηση Πυκνότητας Πιθανότητας* :

$$f(x) = a k^a x^{-(a+1)}.$$

Η *Διασπορά* της τυχαίας μεταβλητής  $T$  βρίσκεται από τις σχέσεις :

$$E(T^2) = a k^2 / (a-2), \quad a > 2,$$

$$E(T^2) = \infty, \quad a \leq 2,$$

ενώ η *Μέση Τιμή* της τυχαίας μεταβλητής  $T$  δίνεται από τις σχέσεις:

$$E(T) = a k / (a-1), \quad a > 1,$$

$$E(T) = \infty, \quad a \leq 1.$$

Δηλαδή αν  $a \leq 2$  η κατανομή έχει άπειρη διασπορά, και αν  $a \leq 1$  έχει και άπειρη μέση τιμή.

Η *Διακύμανση* της τυχαίας μεταβλητής  $T$  είναι ίση με :

$$V(T) = E(T)^2 - [E(T)]^2 \Rightarrow V(T) = a k^2 / [(a-2)(a-1)^2],$$

ενώ η *Τυπική Απόκλιση* της τυχαίας μεταβλητής  $T$  δίνεται από :

$$\sigma = (V(T))^{1/2} = [k/(a-1)] [a/(a-2)]^{1/2}.$$

Η κατανομή **Pareto**, η οποία είναι γνωστή και ως *κατανομή ισχύος-νόμου* (power-law), ως *διπλή εκθετική κατανομή* (double-exponential) και ως *υπερβολική κατανομή* (hyperbolic), έχει χρησιμοποιηθεί για να μοντελοποιήσει κατανομές εισοδήματος που υπερβαίνουν μία ελάχιστη τιμή, τα μεγέθη αστεροειδών, νησιών, πόλεων και γεγονότα διαγραφής - θανάτου (extinction events) [25] και [28]. Μία κατανομή **Pareto** με  $1.05 < a < 1.25$  έχει βρεθεί ότι μοντελοποιεί ικανοποιητικά το ποσόστο του χρόνου CPU που καταναλώνεται για την εκτέλεση μίας τυχαίας διαδικασίας [26].

Ακολουθώντας την [27], ο ορισμός μιας κατανομής ως *βαριάς ουράς* είναι :

$$P [T \geq t] \sim c x^{-a}, \text{ καθώς το } t \rightarrow \infty, a \geq 0.$$

Από τον τύπο αυτό φαίνεται ότι για κάποιες σταθερές  $a$  και  $c$ , ο λόγος  $P [T \geq t] / (c x^{-a})$  τείνει στο 1 καθώς το  $t \rightarrow \infty$ . Σύμφωνα με τον παραπάνω ορισμό η κατανομή **Pareto** είναι κατανομή *βαριάς ουράς* [22].

Ένας πιο γενικός ορισμός των κατανομών *βαριάς ουράς*, ορίζει μία κατανομή ως *βαριάς ουράς* εάν η **υπό συνθήκη υπέρβαση της μέσης τιμής** (Conditional Mean Exceedance,  $CME_x$ ) της αντίστοιχης τυχαίας μεταβλητής  $T$ ,

$$CME_x = E [T - t | T \geq t],$$

είναι μια αύξουσα συνάρτηση του  $t$  [24].

Χρησιμοποιώντας τον δεύτερο ορισμό της κατανομής *βαριάς ουράς*, έστω ότι η τυχαία μεταβλητή  $T$  παριστάνει τον χρόνο αναμονής ενός χρήστη σε ένα σύστημα εξυπηρέτησης. Για χρόνους αναμονής κατανεμημένους σύμφωνα με κατανομές *ελαφριάς ουράς*, όπως είναι για παράδειγμα η ομοιόμορφη κατανομή, η υπέρβαση της υπό συνθήκη μέσης τιμής είναι μία φθίνουσα συνάρτηση του  $t$ . Για μία τέτοια τυχαία μεταβλητή, όσο περισσότερο έχει μείνει ο χρήστης στην αναμονή, τόσο πιο γρήγορα πρόκειται να εξυπηρετηθεί. Για τυχαίους χρόνους αναμονής κατανεμημένους σύμφωνα με κάποια κατανομή *μέσης ουράς*, όπως για παράδειγμα η (χωρίς μνήμη) εκθετική κατανομή, ο αναμενόμενος πρόσθετος χρόνος αναμονής του χρήστη στο σύστημα εξυπηρέτησης είναι ανεξάρτητος από τον χρόνο της μέχρι τη συγκεκριμένη στιγμή αναμονής του. Σε αντίθεση, για τυχαίους χρόνους αναμονής κατανεμημένους σύμφωνα με μία κατανομή *βαριάς ουράς*, όσο περισσότερο έχει μείνει ο χρήστης στην αναμονή, τόσο μεγαλύτερος είναι ο αναμενόμενος πρόσθετος χρόνος αναμονής



του στο σύστημα. Για την κατανομή **Pareto** με  $a > 1$  (δηλαδή με πεπερασμένη μέση τιμή) η υπέρβαση της υπό συνθήκης μέσης τιμής είναι μία γραμμική συνάρτηση του  $t$  [21, p. 70] :

$$CME_x = t / (a-1).$$

Η κατανομή **Pareto** είναι αμετάβλητη με την κλίμακα του χρόνου, δηλαδή η πιθανότητα ο τυχαίος χρόνος αναμονής να είναι τουλάχιστον  $2t$  λεπτά διά της πιθανότητας ότι ο τυχαίος χρόνος αναμονής είναι τουλάχιστον  $t$  λεπτά, είναι μία σταθερά ανεξάρτητη του  $t$  για οποιοδήποτε  $t \geq k$ .

Ένα σχετικό αποτέλεσμα δείχνει ότι η κατανομή **Pareto** είναι η μόνη κατανομή η οποία είναι “αμετάβλητη υπό περικοπή από κάτω” (invariant under truncation from below) [21] και [29]. Δηλαδή, για την τυχαία μεταβλητή  $T$  με κατανομή **Pareto** και για  $y \geq t_0$  ισχύει :

$$P [ T > y | T > t_0 ] = P [ (t_0/k)T > y ] = (t_0/y)^a.$$

Έτσι η υπό συνθήκη κατανομή είναι επίσης **Pareto** με την ίδια παράμετρο σχηματισμού  $a$  και μία νέα παράμετρο τοποθεσίας  $k' = t_0$ .

### 2.2.2 Χρόνοι μεταξύ αφίξεων πακέτων σύμφωνα με την κατανομή **Pareto**

Σ' αυτό το τμήμα του Κεφαλαίου θα συζητήσουμε και θα προσπαθήσουμε να εξηγήσουμε διαισθητικά τις εξαρτήσεις ευρείας χρονικής κλίμακας που παρατηρούνται στους χρόνους μεταξύ αφίξεων πακέτων της εφαρμογής TELNET. Εδώ είναι χρήσιμο να υπενθυμίσουμε, ότι έχει παρατηρηθεί ότι η κυρίως μάζα πιθανότητας της κατανομής των χρόνων μεταξύ αφίξεων διαδοχικών πακέτων της εφαρμογής TELNET χαρακτηρίζεται ικανοποιητικά από κατανομή **Pareto** με παράμετρο σχήματος  $a=0.9$ , καθώς επίσης το επάνω 3% της μάζας πιθανότητας της ίδιας κατανομής (ουρά), χαρακτηρίζεται ικανοποιητικά από κατανομή **Pareto** με παράμετρο σχήματος  $a=0.95$  [13]. Θεωρούμε πακέτα που γεννιούνται από μία απλή σύνδεση υποθέτοντας χρόνους μεταξύ αφίξεων πακέτων ανεξάρτητους μεταξύ τους και κατανεμημένους σύμφωνα με την κατανομή **Pareto**, με παράμετρο σχήματος  $a$  και παράμετρο τοποθεσίας  $k$ .

Στη συνέχεια επικεντρώνουμε την προσοχή μας στην διαδικασία μέτρησης αφίξεων πακέτων  $X = \{ X_i \}_{i=0,1,2,\dots}$ , όπου  $X_i$  είναι ο αριθμός των πακέτων που έχουν αφιχθεί κατά τη διάρκεια του  $i$ -οστού χρονικού τμήματος (bin), κάθε χρονικό τμήμα υποτίθεται σταθερού μήκους  $b$ . Δίνουμε μία εξήγηση της παρατηρούμενης εξάρτησης ευρείας χρονικής κλίμακας της διαδικασίας μέτρησης, εξετάζοντας τις ιδιότητες της διαδικασίας αφίξεων των πακέτων με ιδιαίτερη έμφαση στους χρόνους μεταξύ αφίξεων. Αν και η διαδικασία μέτρησης αφίξεων

πακέτων δεν παρουσιάζει εξάρτηση ευρείας χρονικής κλίμακας, όταν παρατηρείται σε πεπερασμένη κλίμακα χρόνου εμφανίζει ιδιότητες που παρατηρούνται στις αυτό-όμοιες διαδικασίες. Συγκεκριμένα, αποδεικνύεται ότι αυξάνοντας το  $b$  (το μήκος του χρονικού τμήματος) τα κυρίαρχα χαρακτηριστικά της διαδικασίας μέτρησης αφίξεων πακέτων δεν αλλάζουν.

Μία στάσιμη (stationary) στοχαστική διαδικασία λέγεται ότι έχει εξάρτηση ευρείας χρονικής κλίμακας (long-range dependent) εάν η συνάρτηση αυτοσυσχέτισης της  $r(k)$  είναι μη αθροίσιμη (δηλαδή,  $\sum_k r(k) = \infty$ ). Ο ορισμός αυτός έχει εφαρμογή μόνο σε στοχαστικές διαδικασίες απείρου ορίζοντα (infinite time series). Τα απλούστερα μοντέλα με εξάρτηση ευρείας χρονικής κλίμακας είναι οι **αυτό-όμοιες διαδικασίες** (self-similar), των οποίων οι συναρτήσεις αυτοσυσχέτισης φθίνουν κατά υπερβολικό τρόπο (hyperbolically decaying). Στοχαστικά μοντέλα που βασίζονται σε αυτό-όμοιες στοχαστικές διαδικασίες είναι ιδιαίτερα χρήσιμα διότι η εξάρτηση ευρείας χρονικής κλίμακας χαρακτηρίζεται από μία μόνο παράμετρο, η οποία μπορεί να εκτιμηθεί από παρατηρήσεις της στοχαστικής διαδικασίας που προσπαθούμε να μοντελοποιήσουμε. Ακόμη και όταν δείγμα της διαδικασίας αφίξεων πακέτων που προκύπτει από παρατήρηση πεπερασμένης χρονικής διάρκειας δεν μας οδηγεί στο συμπέρασμα ότι η διαδικασία αφίξεων είναι αυτό-όμοια, εάν περιέχει σημαντικές συσχετίσεις ευρείας χρονικής κλίμακας είναι καλύτερο να προσεγγιστεί από μία αυτό-όμοια διαδικασία παρά από μία διαδικασία **Poisson**. Συνεπώς, η μοντελοποίηση με αυτό-όμοιες στοχαστικές διαδικασίες εμφανίζεται σαν μία πολλά υποσχόμενη εναλλακτική λύση της **Poisson** εκεί όπου η τελευταία αποτυγχάνει να παράξει ικανοποιητικά μοντέλα [13].

Έστω ότι το  $\{X_i^{(b)}\}$  συμβολίζει τη διαδικασία που σχετίζεται με τις μετρήσεις των αφίξεων πακέτων όταν χρησιμοποιούμε χρονικά τμήματα παρατήρησης μήκους  $b$ . Ενδιαφερόμαστε για τη συμπεριφορά του  $\{X_i^{(b)}\}$  για διαφορετικές τιμές  $b$ .

Αντί να αναλύσουμε τις σχέσεις ανάμεσα στις ακριβείς τιμές της διαδικασίας στα διάφορα χρονικά τμήματα, απλοποιούμε το πρόβλημα και απλά κοιτάμε εάν για ένα δεδομένο  $i$ ,  $X_i^{(b)}=0$  ή  $X_i^{(b)}>0$ . Στην πρώτη περίπτωση αναφερόμαστε σε ένα κενό χρονικό τμήμα, ενώ στη δεύτερη σε ένα απασχολημένο χρονικό τμήμα. Επιπλέον, για  $j \geq i$ , καλούμε την  $X_{i,\dots,j}^{(b)}$  μία “έκρηξη” από απασχολημένα χρονικά τμήματα αν για όλα τα  $m$ ,  $i \leq m \leq j$ , το χρονικό τμήμα  $m$  είναι απασχολημένο. Παρομοίως,  $X_{i,\dots,j}^{(b)}$  ονομάζεται ένα ανενεργό διάστημα εάν για όλα τα  $m$ ,  $i \leq m \leq j$ , τα αντίστοιχα χρονικά τμήματα είναι κενά.

Ενδιαφερόμαστε για την σχετική υπερίσχυση των “εκρήξεων” έναντι των ανενεργών διαστημάτων καθώς αλλάζουμε το μήκος του χρονικού τμήματος παρατήρησης  $b$  και την

παράμετρο σχήματος  $a$  της κατανομής *Pareto* που χαρακτηρίζει τους χρόνους μεταξύ αφίξεων διαδοχικών πακέτων.

Ας υποθέσουμε ότι το τμήμα  $i$  είναι απασχολημένο και το τμήμα  $i - 1$  είναι κενό. Τότε στο τμήμα  $i$  αρχίζει μία “έκρηξη”. Με κάθε τμήμα σχετίζεται ένα σύνολο *Pareto* κατανεμημένων χρόνων μεταξύ αφίξεων πακέτων, που αρχίζει με το  $I_n$  το οποίο αντιστοιχεί στην πρώτη άφιξη σ’ αυτό το χρονικό τμήμα. Γνωρίζουμε ότι  $I_n > b$  επειδή το προηγούμενο χρονικό τμήμα είναι κενό. Θεωρούμε τώρα τους ακόλουθους χρόνους μεταξύ αφίξεων πακέτων  $I_{n+1} \dots I_{n+l}$  που συνεισφέρουν στην “έκρηξη” των συνεχόμενων απασχολημένων χρονικών τμημάτων. Καθεμία από αυτές τις τυχαίες μεταβλητές πρέπει να είναι μικρότερη από  $2b$ , διαφορετικά κάποιο τμήμα θα έμενε κενό και έτσι η “έκρηξη” θα τερματιζόταν νωρίτερα. Επιπλέον, κάθε χρόνος μεταξύ αφίξεων διαδοχικών πακέτων στο διάστημα  $(b, 2b)$  μπορεί να καταστήσει το επόμενο χρονικό τμήμα κενό (αυτό βέβαια εξαρτάται από τη χρονική στιγμή στην οποία συνέβη η τελευταία άφιξη πακέτου μέσα στο τελευταίο χρονικό τμήμα). Συνεπώς, κάθε χρόνος  $I$  μεταξύ αφίξεων διαδοχικών πακέτων, τέτοιος ώστε  $I > 2b$  σίγουρα θα τερματίσει την “έκρηξη” ενώ εάν απλά  $I > b$  η “έκρηξη” ίσως να τερματιστεί.

Μια που οι χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων θεωρούνται ανεξάρτητοι, έχουμε μία κατάσταση παρόμοια μ’ αυτή της γεωμετρικής τυχαίας μεταβλητής : για κάθε δεδομένο χρόνο μεταξύ αφίξεων η “έκρηξη” θα τερματιστεί με πιθανότητα  $p_i$  και θα συνεχιστεί με πιθανότητα  $1-p_i$ . Εδώ το  $p_i$  είναι συνάρτηση της ακριβούς χρονικής στιγμής της τελευταίας άφιξης πακέτου στο υπό εξέταση χρονικό τμήμα, αλλά περιορίζεται από τα ακόλουθα όρια:

$$P[ I > 2b ] \leq p_i \leq P[ I > b ]$$

$$\text{όπου } P[ I > 2b ] = (k/2b)^a \text{ και } P[ I > b ] = (k/b)^a ,$$

όπου  $k$  και  $a$  είναι οι παράμετροι τοποθεσίας και σχήματος της κατανομής *Pareto* και  $b$  είναι το μήκος του χρονικού τμήματος που εξετάζουμε [13]. Η εξήγηση της παραπάνω σχέσης είναι ότι η πιθανότητα του τερματισμού μιας “έκρηξης”,  $p_i$ , είναι τουλάχιστον ίση με την πιθανότητα  $P[I > 2b]$  (μια και εάν  $I > 2b$  η “έκρηξη” τερματίζεται σίγουρα), και μικρότερη ή ίση από την πιθανότητα  $P[I > b]$  (μια και εάν  $I > b$  η “έκρηξη” ίσως να τερματιστεί).

Μπορούμε συνεπώς να υπολογίσουμε τα άνω και κάτω όρια στο αναμενόμενο (μέσο) μήκος μίας έκρηξης χρησιμοποιώντας την μέση τιμή των γεωμετρικών τυχαίων μεταβλητών με πιθανότητα “επιτυχίας” που αντιστοιχεί στα άνω και κάτω όρια της προηγούμενης σχέσης.

Έστω ότι  $B$  είναι ο μέσος αριθμός χρονικών τμημάτων μίας “έκρηξης”. Μπορεί να αποδειχτεί ότι :

$$B \approx \begin{cases} b/k & , \text{εάν } a=2, b \gg k \\ \log(b/k) & , \text{εάν } a=1, b \gg k \text{ και} \\ \in [1, 2^{1/2}] & , \text{εάν } a=1/2, \end{cases}$$

όπου  $b \gg k$  ισχύει εάν  $b-k \approx b$ .

Για  $a=2$ , καθώς ευρύνουμε το χρονικό πεδίο παρατήρησης μας επιλέγο-ντας όλο και μεγαλύτερο  $b$ , θα παρατηρούμε κατά μέσο όρο όλο και μεγαλύτερης διάρκειας “εκρήξεις”. Για  $a=1$  η μέση διάρκεια των “εκρήξεων” γίνεται μεγαλύτερη καθώς αυξάνουμε το μήκος του χρονικού τμήματος παρατήρησης αλλά η αύξηση αυτή συμβαίνει πολύ αργά, και τέλος για  $a=1/2$ , η μέση διάρκεια των “εκρήξεων” έχει σταθερό μήκος ανεξάρτητο από το μέγεθος  $b$  των χρονικών τμημάτων (!).

Στη συνέχεια θα εξετάσουμε το μήκος του ανενεργού διαστήματος που χωρίζει δύο συνεχόμενες “εκρήξεις”. Έστω ότι  $L$  είναι τυχαία μεταβλητή που συμβολίζει το μήκος ενός ανενεργού διαστήματος και  $L_b$  τυχαία μεταβλητή που συμβολίζει τον αριθμό των χρονικών τμημάτων (μήκους  $b$ ) που περιέχονται στο ανενεργό διάστημα μήκους  $L$ . Κάθε ανενεργό διάστημα οφείλεται σε ένα μοναδικό χρόνο μεταξύ αφίξεων διαδοχικών πακέτων που είναι σίγουρα μεγαλύτερος από  $b$  και πιθανόν μεγαλύτερος από  $2b$ . Εξαιτίας της ιδιότητας της κατανομής **Pareto** να είναι αμετάβλητη υπό περικοπή από κάτω, η κατανομή του  $L$  θα είναι στοχαστικά περιορισμένη ανάμεσα στις κατανομές  $P(b,a)$  και  $P(2b,a)$ , όπου το  $P(k,a)$  συμβολίζει την κατανομή **Pareto** με παραμέτρους  $k$  και  $a$ . Συνεπώς ισχύει :

$$P[L \leq x \mid L > 2b] \leq P[L \leq x \mid \text{ανενεργό}] \leq P[L \leq x \mid L > b]$$

οπότε

$$1 - (2b/x)^a \leq P[L \leq x \mid \text{ανενεργό}] \leq 1 - (b/x)^a,$$

και για  $x=mb$  προκύπτει :

$$1 - (2/m)^a \leq P[L_b \leq m] \leq 1 - (1/m)^a.$$

Έτσι η κατανομή της τυχαίας μεταβλητής  $L_b$  περιορίζεται στοχαστικά από δύο κατανομές ανεξάρτητες από το  $b$ . Δηλαδή, ανεξάρτητα από την κλίμακα του χρόνου πάνω στην οποία παρατηρούμε τη διαδικασία μέτρησης αφίξεων, τα ανε-νεργά διαστήματα ανάμεσα στις “εκρήξεις” εκφρασμένα σε αριθμό χρονικών τμημάτων, θα φαίνονται τα ίδια (χαρακτηρίζονται από την ίδια κατανομή).

Συμπεράσματα για τη συμπεριφορά της διαδικασίας μέτρησης αφίξεων πακέτων καθώς μεταβάλλεται το  $a$ .

- ♦ Για  $a=2$ , ο μέσος αριθμός των χρονικών τμημάτων που περιέχονται σε μία “έκρηξη” αυξάνεται γραμμικά με το  $b$ , ενώ τα χρονικά τμήματα που περιέχονται στα ανενεργά διαστήματα παραμένουν στοχαστικά αμετάβλητα καθώς αλλάζουμε τα  $b$ .
- ♦ Για  $a=1/2$ , το μέσο μήκος μιας “έκρηξης” (εκφρασμένο σε αριθμό χρονικών τμημάτων) είναι σταθερό για όλες τις κλίμακες χρόνου, όπως και τα μήκη των ανενεργών διαστημάτων (εκφρασμένα σε αριθμό χρονικών τμημάτων), και η διαδικασία εμφανίζεται ως αυτό-όμοια πάνω σε όλες τις κλίμακες χρόνου.
- ♦ Για  $a=1$ , το μήκος της “έκρηξης” (εκφρασμένο σε χρονικά τμήματα) αυξάνεται πολύ αργά (λογαριθμικά). Αυτό σημαίνει ότι πάνω σε ένα μεγάλο χρονικό διάστημα παρατήρησης, η υπερίσχυση των “εκρήξεων” έναντι των ανενεργών διαστημάτων διατηρείται ουσιαστικά αμετάβλητη, και η διαδικασία εμφανίζεται ως αυτό-όμοια πάνω σε πολλές κλίμακες χρόνου.

Η διαδικασία με  $a=1$  είναι παρόμοια με αυτή της κίνησης μιας απλής σύνδεσης TELNET, όπου χρησιμοποιούμε την κατανομή **Pareto** με  $a=0.95$  για να χαρακτηρίσουμε τους χρόνους μεταξύ αφίξεων διαδοχικών πακέτων για το επάνω μέρος της ουράς της κατανομής. Το μοντέλο αυτό εξηγεί εν μέρει γιατί η κίνηση μιας απλής σύνδεσης TELNET παρατηρούμενη στην πράξη εμφανίζει στατιστικά χαρακτηριστικά που συναντούμε στις αυτό-όμοιες στοχαστικές διαδικασίες.

Τέλος, δίνουμε μια εξήγηση του λόγου για τον οποίο οι διαδικασίες μέτρησης αφίξεων πακέτων με  $a=1$  και  $a=1/2$  δεν είναι στην πραγματικότητα αυτό-όμοιες διαδικασίες παρόλο που η παρατηρούμενη ισορροπία ανάμεσα στις “εκρήξεις” και τα ανενεργά διαστήματα υποδεικνύει ότι θα μπορούσαν να είναι. Έχουμε δείξει ότι το μήκος  $L$  των ανενεργών διαστημάτων είναι στοχαστικά περιορισμένο από δύο κατανομές **Pareto** με την ίδια παράμετρο σχήματος  $a$ . Αλλά για  $a \leq 1$ , η μέση τιμή της τυχαίας μεταβλητής μιας κατανομής **Pareto** είναι άπειρη. Από την άλλη πλευρά, το αναμενόμενο μήκος της “έκρηξης” για  $a \leq 1$  είναι πεπερασμένο (δες τον τύπο που δίδει το Β, σε αυτή την παράγραφο). Χρησιμοποιώντας αυτά τα γεγονότα και βλέποντας τις διάρκειες των “εκρήξεων” και των ανενεργών διαστημάτων της διαδικασίας μέτρησης αφίξεων σαν μια εναλλασσό-μενη ανανεωτική διαδικασία (alternating renewal process) [30] συνεπάγεται ότι για  $a \leq 1$  και εφόσον η

διαδικασία έχει φτάσει στην μόνιμη κατάσταση της, κάθε χρονικό τμήμα είναι κενό με πιθανότητα 1 (ανεξάρτητα από την τιμή  $b$  του μεγέθους του χρονικού τμήματος). Η συνάρτηση αυτοσυσχέτισης της διαδικασίας μέτρησης αφίξεων είναι συνεπώς μηδέν παντού, και συνεπώς ολοκληρώσιμη, και έτσι η διαδικασία δεν έχει εξάρτηση ευρείας χρονικής κλίμακας (ως εκ' τούτου δεν μπορεί να είναι *αυτό-όμοια*) [13].

Αν και οι διαδικασίες μέτρησης αφίξεων πακέτων δεν είναι αυστηρά *αυτό-όμοιες*, είναι σημαντικό να τονιστεί ότι όταν παρατηρηθούν πάνω σε μία πεπερασμένη κλίμακα χρόνου (με άλλα λόγια πριν αυτές φτάσουν στη μόνιμη κατάσταση τους), έχει βρεθεί για παράδειγμα ότι η διαδικασία μέτρησης αφίξεων πακέτων με *Pareto* κατανομημένους χρόνους μεταξύ αφίξεων (με  $a \leq 1$ ), εμφανίζεται με πολλούς τρόπους παρόμοια με *αυτό-όμοια* διαδικασία [13]. Υποθέτοντας ότι αυτή η παρατηρούμενη ομοιότητα ισχύει και όταν η διαδικασία μέτρησης αφίξεων πακέτων είναι *πολυπλεγμένη*, μπορούμε να κατανοήσουμε το λόγο για τον οποίο η παρατηρούμενη στην πράξη κίνηση πακέτων της εφαρμογής TELNET εμφανίζεται ως *αυτό-όμοια*.

Το γεγονός ότι η διαδικασία μέτρησης αφίξεων πακέτων δεν έχει *εξάρτηση ευρείας χρονικής κλίμακας* δεν συνεπάγεται ότι η κίνηση πακέτων της εφαρμογής TELNET δεν είναι πραγματικά *αυτό-όμοια*. Ίσως η κίνηση πακέτων της εφαρμογής TELNET είναι πράγματι *αυτό-όμοια*, αλλά οι απλοποιητικές υποθέσεις στους συλλογισμούς μας (π.χ., ανεξάρτητοι και ομοιόμορφα κατανομημένοι χρόνοι μεταξύ αφίξεων χωρίς χρήση πολυπλεξίας) αποτυγχάνουν να μοντελοποιήσουν πιστά τις ιδιότητες της κίνησης πακέτων που είναι απαραίτητες για αληθινή *αυτό-ομοιότητα* [13].

Ο παραπάνω συλλογισμός δείχνει, επίσης, ότι είναι πιθανό μια διαδικασία η οποία δεν έχει *εξάρτηση ευρείας χρονικής κλίμακας*, να εμφανίζεται ότι έχει πάνω σε πολλές πεπερασμένες χρονικές κλίμακες. Αυτό δείχνει μερικούς από τους κινδύνους που εμφανίζονται όταν κάποιος προσπαθεί να ισχυριστεί (ή να αποδείξει) πραγματική *αυτό-ομοιότητα* (ή πιο γενικά *εξάρτηση ευρείας χρονικής κλίμακας*) βασιζόμενος κατ' ανάγκη μόνο στις διαθέσιμες χρονικά πεπερασμένες μετρήσεις, χωρίς να διαθέτει ένα μοντέλο με βάση το οποίο να μπορεί να αποδείξει την *αυτό-ομοιότητα* αναλυτικά [13].

## **ΚΕΦΑΛΑΙΟ 3**

### **Ο ευσταθής Αλγόριθμος ALOHA**

Σκοπός της διπλωματικής εργασίας είναι η μελέτη της απόδοσης (*throughput*, και συμπεριφορά της κατανομής καθυστερήσεων πακέτων) αλγορίθμων τυχαίας προσπέλασης οι οποίοι έχουν επινοηθεί και σχεδιαστεί για *Poisson* αφίξεις πακέτων, όταν οι χρόνοι μεταξύ αφίξεων πακέτων χαρακτηρίζονται από την κατανομή *Pareto*. Για τον ορισμό του *throughput* ενός αλγορίθμου τυχαίας προσπέλασης παραπέμπουμε στην σχετική συζήτηση στο Κεφάλαιο 1 αυτής της εργασίας. Η καθυστέρηση πακέτου ορίζεται σαν το χρονικό διάστημα από την χρονική στιγμή άφιξης του πακέτου μέχρι το τέλος της χρονικής μονάδας (slot) που περιέχει την επιτυχημένη μετάδοση του ίδιου πακέτου.

Οι αλγόριθμοι τυχαίας προσπέλασης που επελέγησαν να μελετηθούν είναι : (α) ο ευσταθής *ALOHA* με δυναμικά μεταβαλλόμενη πιθανότητα επαναμετάδοσης αντίστροφα ανάλογη του στιγμιαίου αριθμού πακέτων στο backlog, και (β) ο αλγόριθμος στοίβας επίλυσης συγκρούσεων πακέτων (για τον αναγνώστη που επιθυμεί να φρεσκάρει την μνήμη του όσον αφορά τα χαρακτηριστικά λειτουργίας των παραπάνω αλγορίθμων τυχαίας προσπέλασης παραπέμπουμε στο Κεφάλαιο 1 αυτής της εργασίας).

Ανάμεσα στην πληθώρα των αλγορίθμων τυχαίας προσπέλασης που υπάρχουν στην βιβλιογραφία, επιλέξαμε τους παραπάνω δύο για τον λόγο ότι και οι δύο είναι ελεύθερης προσπέλασης, ώστε η πρόσβαση “φρέσκων” πακέτων στο σύστημα να γίνεται εύκολα και με απλά υλοποιήσιμο τρόπο (οι χρήστες με “φρέσκα” πακέτα δεν χρειάζεται να γνωρίζουν την κατάσταση του συστήματος προτού μεταδώσουν τα “φρέσκα” πακέτα τους για πρώτη φορά). Επιπλέον η οικογένεια των αλγορίθμων τύπου *ALOHA* είναι αρκετά δημοφιλής, ο κλασικός αλγόριθμος *ALOHA* έχει χρησιμοποιηθεί ευρέως στην πράξη (π.χ., στο Ethernet LAN), παρά τα προβλήματα αστάθειας που παρουσιάζει. Στο Κεφάλαιο 3 μελετάμε την απόδοση του ευσταθούς *ALOHA*, ενώ ο αλγόριθμος στοίβας εξετάζεται στο επόμενο Κεφάλαιο.

### 3.1 Εισαγωγικά για την εφαρμογή του αλγορίθμου *ALOHA*

Ο αλγόριθμος που εξετάζεται είναι ο ευσταθής ιδανικός *ALOHA* με πιθανότητα αναμετάδοσης πακέτου από το backlog ίση με  $(d-\lambda)/n$ , όπου  $d$  είναι μία σταθερά  $\lambda < d \leq 1$ , η οποία επιλέγεται έτσι ώστε να επιτευχθεί η μέγιστη βελτιστοποίηση της μέσης καθυστέρησης πακέτου (δηλαδή, η μικρότερη μέση καθυστέρηση),  $\lambda$  είναι ο συνολικός ρυθμός άφιξης πακέτων στο σύστημα, και  $n$  είναι ο αριθμός των πακέτων που ανήκουν στο *backlog* (σύνολο των πακέτων που έχουν συγκρουστεί τουλάχιστον μία φορά και προσπαθούν να αναμεταδώσουν). Παρόλο που η υλοποίηση αυτού του αλγορίθμου παρουσιάζει πρακτικές δυσκολίες μια και οι χρήστες δεν είναι δυνατόν να γνωρίζουν ανά



πάσα στιγμή τις ακριβείς τιμές των  $n$  και  $\lambda$ , στην πράξη αυτές οι ποσότητες πρέπει να εκτιμηθούν από τους χρήστες με βάση την ανάδραση που παρέχει το κανάλι (δες και τη σχετική εκτενή συζήτηση στο Κεφάλαιο 1 της παρούσας εργασίας), επιλέξαμε να τον μελετήσουμε διότι η απόδοση του θα είναι καλύτερη από οποιονδήποτε πρακτικά υλοποιήσιμο αλγόριθμο τύπου ευσταθούς *ALOHA*. Πιο συγκεκριμένα, ο ιδανικός αλγόριθμος *ALOHA* που εξετάζουμε επιτυγχάνει μικρότερη μέση καθυστέρηση πακέτου για κάθε τιμή του ρυθμού άφιξης  $\lambda$ , από αυτήν οποιουδήποτε πρακτικά υλοποιήσιμου αλγόριθμου τύπου *ALOHA*, διότι ο τελευταίος δεν έχει ακριβή γνώση των ποσοτήτων  $n$  και  $\lambda$  κάθε χρονική στιγμή της λειτουργίας του. Συνεπώς η συμπεριφορά της καθυστέρησης του μελετούμε-νου ιδανικού ευσταθούς *ALOHA* παρέχει κάτω όρια στις καθυστερήσεις που μπορούν να επιτευχθούν με ευσταθείς αλγορίθμους που ακολουθούν την φιλοσοφία μεταδόσεων και επαναμεταδόσεων του πρωτόκολλου *ALOHA*.

Κατά την εφαρμογή του αλγορίθμου *ALOHA*, θεωρούμε *άπειρο αριθμό χρηστών* και υποθέτουμε ότι *κάθε “φρέσκο” πακέτο φτάνει σε διαφορετικό χρήστη στο σύστημα* [4]. Τα “φρέσκα” πακέτα που γεννιούνται στο σύστημα στην διάρκεια ενός χρονικού τμήματος (slot), μεταδίδονται στο κανάλι στην διάρκεια του επόμενου χρονικού τμήματος (ελεύθερη προσπέλαση “φρέσκων” πακέτων). Τα “φρέσκα” πακέτα που συμμετέχουν σε μια σύγκρουση εισάγονται στο *backlog* και μένουν εκεί μέχρι να μεταδοθούν επιτυχώς. Τα πακέτα που έχουν εισαχθεί στο backlog ρίχνουν ένα πολωμένο νόμισμα και ανάλογα με το αποτέλεσμα μεταδίδουν ή όχι στο αμέσως επόμενο χρονικό τμήμα. Ο κώδικας που υλοποιεί τον παραπάνω αλγόριθμο παρατίθεται στο Παράρτημα 1.

Η προσομοίωση του αλγορίθμου εκτελέστηκε σε UNIX σταθμό εργασίας *SUN SPARCstation 5*. Σε κάθε εκτέλεση της προσομοίωσης (*run*) προσομοιώθη-κε η μετάδοση ενός εκατομμυρίου πακέτων και έγιναν οι απαραίτητες προσο-μοιώσεις (*runs*) ώστε να βρεθεί η μικρότερη μέση καθυστέρηση των πακέτων στο σύστημα για κάθε ρυθμό άφιξης  $\lambda$  με διάφορες παραμέτρους τοποθεσίας  $k$ . Ο αριθμός του ενός εκατομμυρίου μεταδιδόμενων πακέτων, επελέγει ώστε η συμπεριφορά του αλγορίθμου να έχει φτάσει σε “μόνιμη κατάσταση” (steady-state). Σε αρκετές περιπτώσεις προσομοιώσαμε λιγότερα και περισσότερα μεταδιδόμενα πακέτα (500,000 και 2,000,000 αντίστοιχα), και παρατηρήσαμε ότι σε όλες τις περιπτώσεις αυτές η προσομοίωση ενός εκατομμυρίου μεταδιδόμενων πακέτων εκτιμούσε κατά ικανοποιητικό τρόπο τις μετρικές απόδοσης του αλγορίθμου σε “μόνιμη κατάσταση”.

Ο προσομοιωτής υπολογίζει και εκτιμά διάφορες μετρικές όπως για παράδειγμα, το ρυθμό άφιξης των πακέτων στο σύστημα δεδομένων των παραμέτρων τοποθεσίας  $k$  και σχήματος  $a$  της κατανομής *Pareto*, τον αριθμό των πακέτων που μένουν πίσω στο backlog μετά το πέρας της προσομοίωσης του ενός εκατομμυρίου πακέτων, τον αριθμό των χρονικών τμημάτων (slots) που προσομοιώνονται, τη συνολική καθυστέρηση των πακέτων στο σύστημα, τη μέση τιμή της καθυστέρησης τους, τη διασπορά της καθυστέρησης, κ.λ.π.

### 3.2 Αποτελέσματα της προσομοίωσης του αλγορίθμου του ιδανικού ALOHA

Το σύστημα μετά από εξαντλητική προσομοίωση (εξαιτίας του γεγονότος ότι η συμπεριφορά του αλγορίθμου για την κατανομή αφίξεων πακέτων *Pareto* ήταν άγνωστη, μια και απ' όσο γνωρίζουμε δεν έχει εξεταστεί μέχρι τώρα) βρέθηκε ότι είναι ευσταθές για τιμές της παραμέτρου τοποθεσίας  $k \geq 0.7$  χρονικά τμήματα, ενώ οι τιμές της παραμέτρου σχηματισμού  $a$  που οδηγούν σε ευσταθή σύστημα ανήκουν στο διάστημα  $[1, 1.83]$ . Το μέγιστο *throughput* αυτού του αλγορίθμου χρησιμοποιώντας την κατανομή *Pareto* για την μοντελοποίηση των χρόνων μεταξύ αφίξεων των πακέτων που εισάγονται στο σύστημα είναι **0.4774(!)**, το οποίο είναι αρκετά μεγαλύτερο (περίπου κατά 30%) από το μέγιστο *throughput* του ίδιου αλγορίθμου για χρόνους μεταξύ αφίξεων πακέτων κατανεμημένους σύμφωνα με την εκθετική κατανομή (*Poisson* αφίξεις) που είναι  $1/e=0.3678$ .

Η συμπεριφορά της μέσης καθυστέρησης πακέτου σαν συνάρτηση του συνολικού ρυθμού αφίξεων πακέτων στο σύστημα, εξετάζεται στη συνέχεια. Η συμπεριφορά αυτή μεταβάλλεται καθώς μεταβάλλονται οι παράμετροι της κατανομής *Pareto*. Στο Κεφάλαιο 2, εξετάστηκε η εκρηκτικότητα της συγκεκριμένης κατανομής με την μεταβολή της τιμής της παραμέτρου σχηματισμού  $a$ . Στα σχήματα Σχ.6 και Σχ.7 φαίνεται η συμπεριφορά της καθυστέρησης πακέτων με την αλλαγή της παραμέτρου τοποθεσίας  $k$ , για διάφορους ρυθμούς αφίξεων πακέτων στο σύστημα  $\lambda$ , που επιτυγχάνονται μεταβάλλοντας την τιμή της παραμέτρου  $a$ . Κάθε σημείο στα αναφερόμενα σχήματα προήλθε από προσομοίωση ενός εκατομμυρίου πακέτων. Στον κάθετο άξονα του Σχ.6, παρουσιάζεται η βέλτιστη μέση καθυστέρηση πακέτων στο σύστημα, η οποία αντιστοιχεί στην κατάλληλη πιθανότητα αναμετάδοσης από το backlog (οι αντίστοιχες βέλτιστες πιθανότητες περιλαμβάνονται στον Πίνακα 1. Ο τρόπος με τον οποίο οι πιθανότητες αυτές βρέθηκαν, για κάθε  $\lambda$ , μέσω προσομοίωσης θα εξηγηθεί σύντομα).

Από το Σχ.6 παρατηρούμε ότι η αύξηση της βέλτιστης μέσης καθυστέρησης ξεκινάει νωρίτερα (για μικρότερες τιμές του  $\lambda$ ) όσο μειώνεται η παράμετρος τοποθεσίας  $k$ , της

κατανομής **Pareto**. Στην μέγιστη τιμή του  $k$  που εξετάζουμε ( $k=0.95$ ) παρατηρούμε στη γραφική παράσταση ότι η βέλτιστη μέση καθυστέρηση διατηρείται αισθητά μικρότερη από 20 slots για ρυθμούς αφίξεων πακέτων μικρότερους από 0.45, και πρακτικά αμελητέα για ρυθμούς αφίξεων πακέτων μικρότερους από 0.40, ενώ για την μικρότερη τιμή του  $k$  που εξετάζουμε ( $k=0.75$ , υπενθυμίζεται ότι η ελάχιστη τιμή του  $k$  που δίδει ευσταθή λειτουργία του αλγορίθμου είναι 0.7), η τιμή της βέλτιστης μέσης καθυστέρησης είναι πάνω από 20 slots για ρυθμούς αφίξεων πακέτων πάνω από 0.3. Επιπλέον παρατηρούμε ότι για δεδομένη τιμή του ρυθμού αφίξεων πακέτων  $\lambda$ , η βέλτιστη μέση καθυστέρηση μειώνεται καθώς το  $k$  αυξάνει. Το τελευταίο αποτέλεσμα επιβεβαιώνει την διαίσθηση μας, διότι καθώς το  $k$  αυξάνεται προς τη μονάδα η πιθανότητα συγκρούσεων πακέτων μειώνεται προς το μηδέν δεδομένου ότι η παράμετρος τοποθεσίας  $k$ , της κατανομής **Pareto** συμβολίζει τον ελάχιστο χρόνο μεταξύ αφίξεων διαδοχικών πακέτων. Στη γραφική παράσταση δεν φαίνεται η τιμή της βέλτιστης μέσης καθυστέρησης για την παράμετρο τοποθεσίας  $k=0.85$  και για ρυθμό άφιξης πακέτων μεγαλύτερο από 0.4, επειδή η αντίστοιχη βέλτιστη μέση καθυστέρηση είναι πολύ υψηλή και ξεπερνάει τα όρια της γραφικής παράστασης.

Είναι γνωστό ότι στους αλγόριθμους τυχαίας προσπέλασης γνώση μόνο της μέσης καθυστέρησης πακέτου δεν επιτρέπει ασφαλείς προβλέψεις για την καθυστέρηση κάποιου τυχαία επιλεγμένου πακέτου, και αυτό διότι η κατανομή της καθυστέρησης συνήθως χαρακτηρίζεται από μεγάλη διασπορά, και μεγάλες τιμές υψηλότερων ροπών [3], [32] και [33]. Γι' αυτό το λόγο, εκτιμούμε μέσω της προσομοίωσης το κλάσμα της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης σαν συνάρτηση του ρυθμού αφίξεων πακέτων  $\lambda$  και για διάφορες τιμές της παραμέτρου  $k$  της κατανομής **Pareto**. Τα αποτελέσματα εμφανίζονται στο Σχ.7. Οι χρησιμοποιούμενες εκτιμήτριες της διασποράς και της μέσης τιμής της καθυστέρησης είναι οι συνήθεις και για πληρότητα δίδονται παρακάτω .

Η χρησιμοποιούμενη *εκτιμήτρια της διακύμανσης* της καθυστέρησης πακέτου είναι μη πολωμένη και δίδεται από :

$$Var(D_i) = \left( \sum_{i=1}^N [D_i - \underline{D}(N)]^2 \right) / (N-1) = \left( \sum_{i=1}^N D_i^2 \right) / (N-1) - (N [\underline{D}(N)]^2) / (N-1)$$

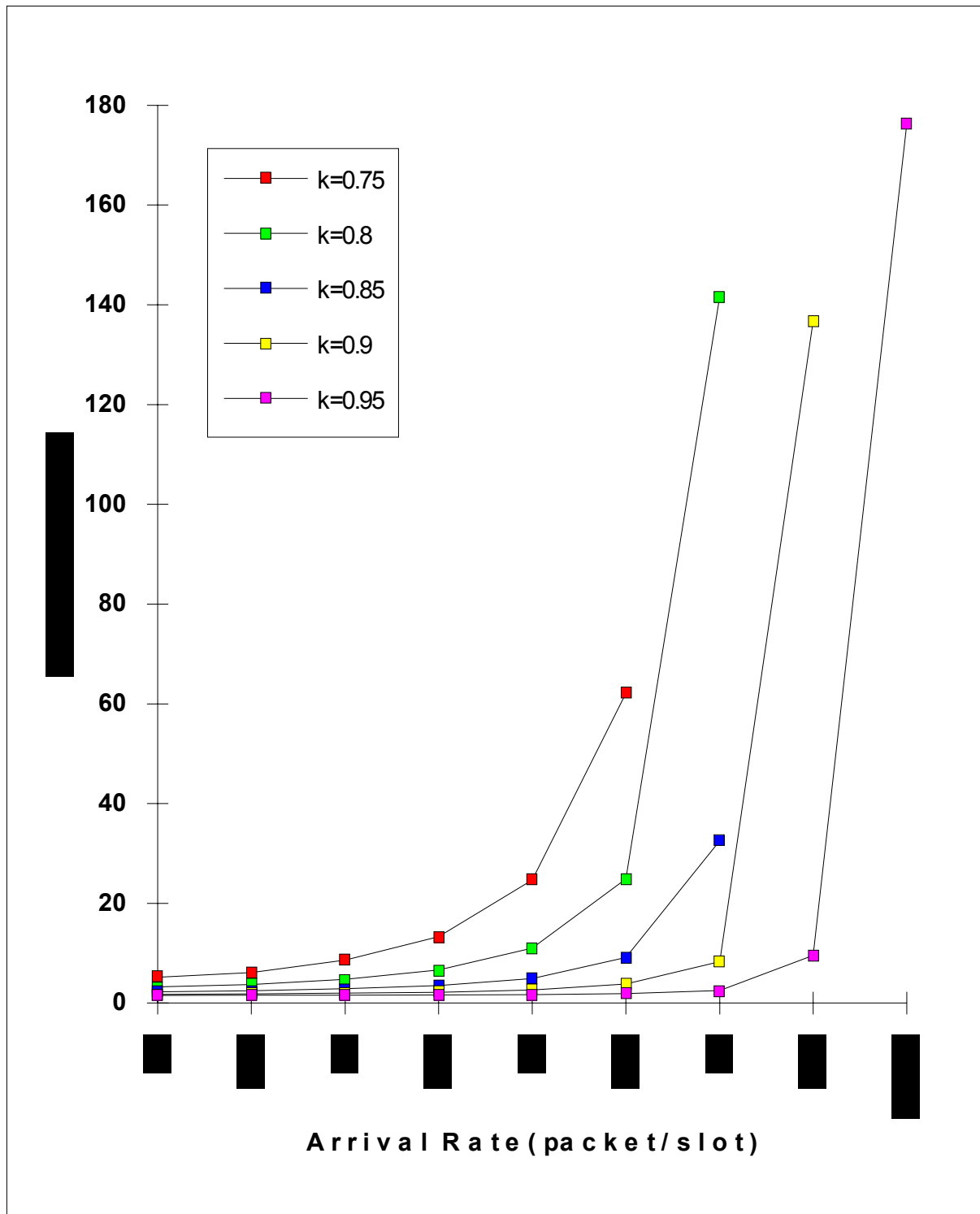
Κατόπιν, η *τυπική απόκλιση* εκτιμάται ως :

$$\sigma(D_i) = (Var(D_i))^{1/2},$$

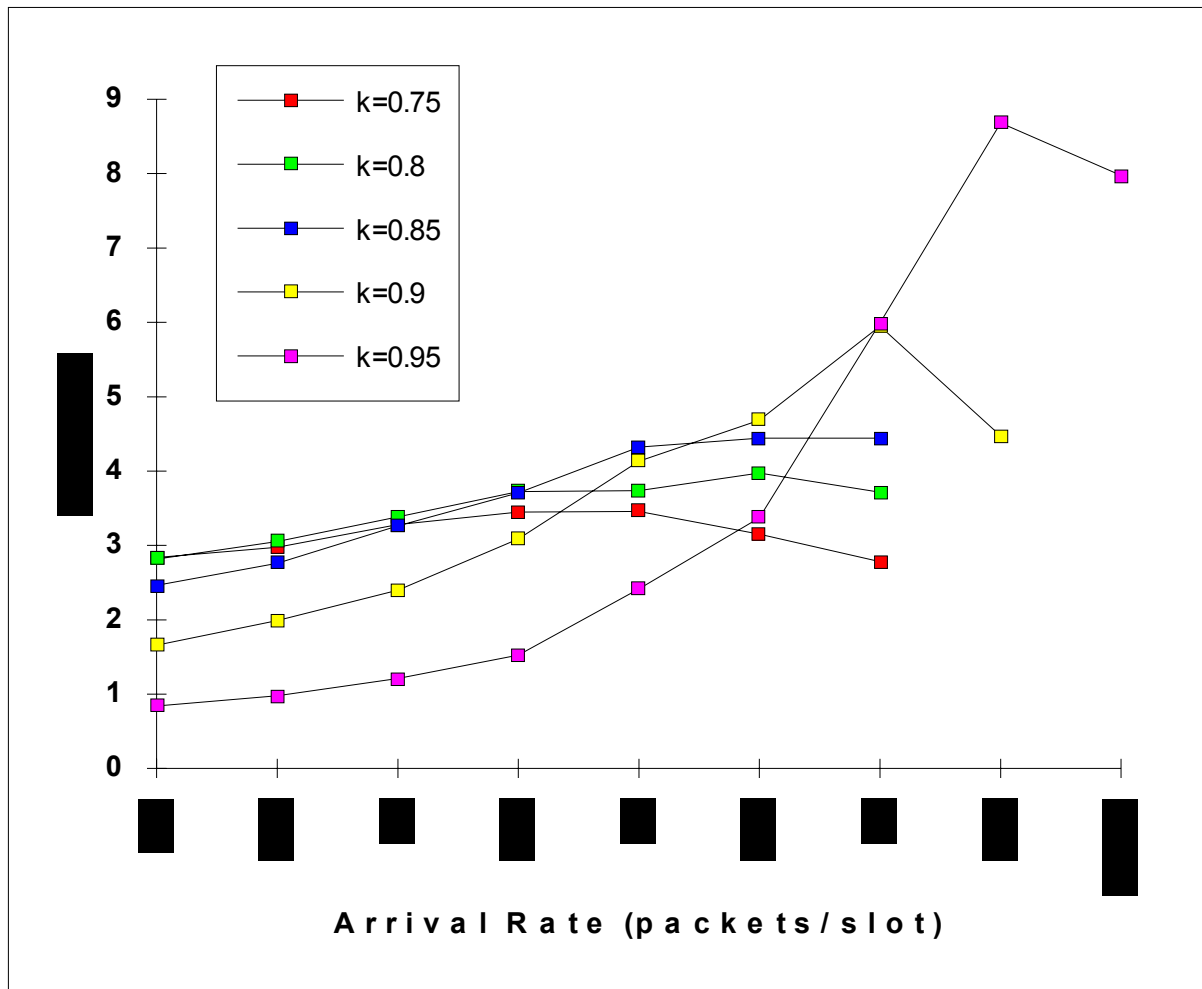
όπου  $D_i$  η παρατηρούμενη (μέσω προσομοίωσης) καθυστέρηση του  $i$ -οστού μετα-διδόμενου πακέτου,  $N$  ο αριθμός των μεταδιδόμενων πακέτων κατά την διάρκεια της προσομοίωσης, και

$\underline{D}(N)$  η εκτιμώμενη μέση τιμή της καθυστέρησης πακέτου με βάση τις επί μέρους καθυστερήσεις των  $N$  μεταδιδόμενων πακέτων που προσομοιώνονται. Η εκτίμηση της μέσης καθυστέρησης γίνεται κατά τον γνωστό τρόπο, δηλαδή  $\underline{D}(N) = (1/N) \sum_{i=1}^N D_i$ .

Από το Σχ.7 φαίνεται ότι καθώς αυξάνεται η παράμετρος τοποθεσίας  $k$  της κατανομής *Pareto*, η προβλεψιμότητα της καθυστέρησης ενός τυχαία επιλεγμένου πακέτου με βάση τη γνώση της μέσης τιμής της καθυστέρησης μειώνεται. Ενώ για  $k=0.75$  το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης διατηρείται κοντά στο 3, για  $k=0.95$  κυμαίνεται από 1 μέχρι 8 (ανάλογα με την τιμή του ρυθμού αφίξεων πακέτων  $\lambda$ ), γεγονός που επιτρέπει στην καθυστέρηση ενός τυχαία επιλεγμένου πακέτου να μεταβάλλεται πολύ



Σχ.6 : Η γραφική παράσταση της βέλτιστης μέσης καθυστέρησης πακέτων στο σύστημα σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$  για διάφορες τιμές της παραμέτρου τοποθεσίας  $k$  της κατανομής *Pareto*.



Σχ.7 : Η γραφική παράσταση του πηλίκου της τυπικής απόκλισης  $\sigma(D)$  διά της μέσης τιμής  $E(D)$  της καθυστέρησης σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$ , για διάφορες τιμές της παραμέτρου τοποθεσίας  $k$  της κατανομής *Pareto*.

γύρω από τη μέση τιμή της. Αυτό οδηγεί στο συμπέρασμα ότι γνωρίζοντας μόνο τις μέσες τιμές της καθυστέρησης πακέτου, δεν μπορούμε να έχουμε σημαντικές ελπίδες πρόβλεψης της καθυστέρησης ενός τυχαία επιλεγμένου πακέτου, μια και στη χειρότερη περίπτωση η τελευταία μπορεί να είναι και 8 φορές μεγαλύτερη από τη μέση τιμή.

Οι βέλτιστες πιθανότητες αναμετάδοσης ενός πακέτου που ανήκει στο backlog που φαίνονται στον ακόλουθο πίνακα (Πίνακας 1), προήλθαν από εξαντλητικό ψάξιμο μέσω προσομοίωσης. Η μεταβλητή σε αυτό το ψάξιμο ήταν η σταθερά  $d$  στον αριθμητή της έκφρασης  $(d-\lambda)/n$  που δίδει την πιθανότητα επαναμετάδοσης, όπου  $\lambda \leq d \leq 1$ .

p	$\kappa=0.75$	$\kappa=0.8$	$\kappa=0.85$	$\kappa=0.9$	$\kappa=0.95$
$\lambda= 0.1$	$(0.54-\lambda)/n$	$(0.55-\lambda)/n$	$(0.49-\lambda)/n$	$(0.5-\lambda)/n$	$(0.73-\lambda)/n$
$\lambda= 0.15$	$(0.58-\lambda)/n$	$(0.53-\lambda)/n$	$(0.48-\lambda)/n$	$(0.52-\lambda)/n$	$(0.6-\lambda)/n$
$\lambda= 0.2$	$(0.55-\lambda)/n$	$(0.55-\lambda)/n$	$(0.49-\lambda)/n$	$(0.54-\lambda)/n$	$(0.6-\lambda)/n$
$\lambda= 0.25$	$(0.55-\lambda)/n$	$(0.54-\lambda)/n$	$(0.52-\lambda)/n$	$(0.54-\lambda)/n$	$(0.59-\lambda)/n$
$\lambda= 0.3$	$(0.59-\lambda)/n$	$(0.61-\lambda)/n$	$(0.55-\lambda)/n$	$(0.52-\lambda)/n$	$(0.54-\lambda)/n$
$\lambda= 0.35$	$(0.67-\lambda)/n$	$(0.59-\lambda)/n$	$(0.6-\lambda)/n$	$(0.61-\lambda)/n$	$(0.61-\lambda)/n$
$\lambda= 0.4$	$(0.79-\lambda)/n$	$(0.67-\lambda)/n$	$(0.65-\lambda)/n$	$(0.56-\lambda)/n$	$(0.53-\lambda)/n$
$\lambda= 0.45$	---	---	---	$(0.62-\lambda)/n$	$(0.57-\lambda)/n$
$\lambda=0.4774$	---	---	---	---	$(0.54-\lambda)/n$

**Πίνακας 1 :** Η βέλτιστη πιθανότητα αναμετάδοσης από το backlog που χρησιμοποιήθηκε στα σχήματα Σχ.6 και Σχ.7.

p	backlog	E(D)	$\sigma(D)/E(D)$
$(.51-\lambda)/n$	1127	1050,89	8.407
$(.52-\lambda)/n$	117	175.66	7.699
$(.53-\lambda)/n$	295	337.21	7.282
$(.54-\lambda)/n$	167	212.77	7.254

**Πίνακας 2 :** Η μεταβολή της συμπεριφοράς καθυστέρησης του αλγορίθμου σε συνάρτηση με την πιθανότητα αναμετάδοσης, για ρυθμό άφιξης  $\lambda=0.4742$  (με  $k=0.95$ ,  $a=1.82$ ).

p	backlog	E(D)	$\sigma(D)/E(D)$
$(0.6-\lambda)/n$	0	31.490	4.431
$(0.61-\lambda)/n$	0	34.473	4.333
$(0.62-\lambda)/n$	0	31.115	3.910
$(0.63-\lambda)/n$	0	29.360	3.998
$(0.64-\lambda)/n$	0	32.952	4.003
$(0.65-\lambda)/n$	0	28.892	3.729
$(0.84-\lambda)/n$	68	48.287	3.299

**Πίνακας 3 :** Η μεταβολή της συμπεριφοράς καθυστέρησης του αλγορίθμου σε συνάρτηση με την πιθανότητα αναμετάδοσης, για ρυθμό άφιξης  $\lambda=0.357$  (με  $k=0.8$ ,  $a=1.4$ ).

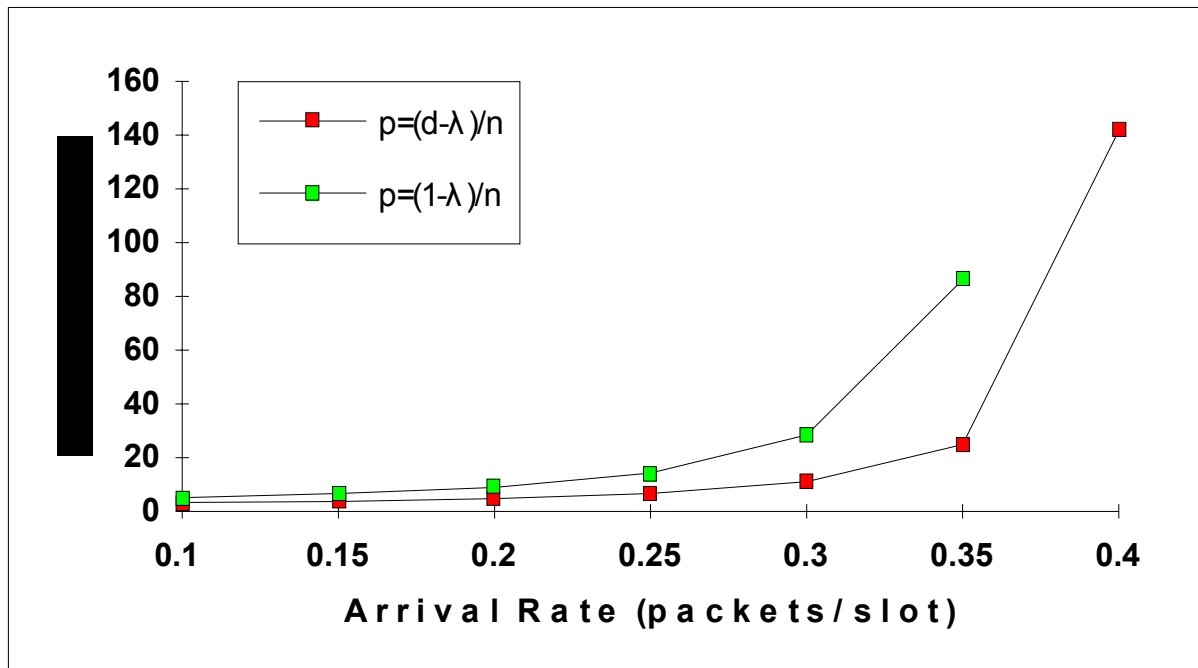
Οι περιπτώσεις με --- είναι ασταθείς, δηλαδή δεν υπάρχει  $d$  τέτοιο ώστε  $\lambda \leq d \leq 1$ , που να οδηγεί σε ευσταθή λειτουργία του αλγορίθμου.

Η υψηλή ευαισθησία της συμπεριφοράς της καθυστέρησης του αλγορίθμου για μικρές μεταβολές της πιθανότητας αναμετάδοσης πακέτου από το backlog, φαίνεται στους Πίνακες 2 και 3, όπου με  $E(D)$  συμβολίζουμε την μέση τιμή και με  $\sigma(D)$  την τυπική απόκλιση της καθυστέρησης πακέτου στο σύστημα. Η δεύτερη στήλη στους Πίνακες 2 και 3, περιέχει τον αριθμό πακέτων στο backlog την χρονική στιγμή που τελειώνει η προσομοίωση. Ο ρυθμός άφιξης πακέτων που θεωρήθηκε στην περίπτωση που παρουσιάζεται στον Πίν. 2 ( $\lambda=0.4742$ ) βρίσκεται πολύ κοντά στο μέγιστο *throughput* του αλγορίθμου, και αντιστοιχεί σε παραμέτρους της κατανομής αφίξεων *Pareto*  $a=1.82$  και  $k=0.95$ .

Στα σχήματα Σχ.8 και Σχ.9 απεικονίζεται η αλλαγή της συμπεριφοράς καθυστέρησης πακέτων του ιδανικού αλγορίθμου *ALOHA* όταν, αντί της βέλτιστης πιθανότητας αναμετάδοσης από το backlog, χρησιμοποιείται μία διαφορετική και απλούστερη, για διάφορους ρυθμούς αφίξεων πακέτων στο σύστημα και για τιμές της παραμέτρου τοποθεσίας  $k$  ίσες με 0.8 και 0.95, αντίστοιχα. Οι δύο καμπύλες που εμφανίζονται στα σχήματα έχουν προκύψει μετά από προσομοίωση του αλγορίθμου με πιθανότητα αναμετάδοσης των πακέτων από το backlog ίση με  $(1-\lambda)/n$  και  $(d-\lambda)/n$  αντίστοιχα, όπου στην δεύτερη περίπτωση η τιμή του  $d$  επιλέγεται κατάλληλα για κάθε  $\lambda$  ώστε η προκύπτουσα μέση καθυστέρηση πακέτου να ελαχιστοποιείται. Και στα δύο σχήματα είναι εμφανές ότι η καμπύλη που αντιστοιχεί στη βέλτιστη λειτουργία του αλγορίθμου επιτυγχάνει τις χαμηλότερες μέσες καθυστερήσεις από την καμπύλη που αντιστοιχεί σε πιθανότητα επαναμετάδοσης από το backlog ίση με  $(1-\lambda)/n$ . Από τα σχήματα Σχ.8 και Σχ.9 επίσης παρατηρούμε ότι το *throughput* του αλγορίθμου είναι μικρότερο στη δεύτερη (υποβέλτιστη) περίπτωση. Επίσης στα σχήματα αυτά παρατηρούμε ότι η αλλαγή στην μέση καθυστέρηση από τις βέλτιστες τιμές γίνεται νωρίτερα αισθητή για μικρότερες τιμές του  $k$ . Όταν το  $k=0.8$  ( $k=0.95$ ), η διαφορά στην μέση καθυστέρηση γίνεται σημαντική για  $\lambda>0.25$  ( $\lambda>0.35$ ).

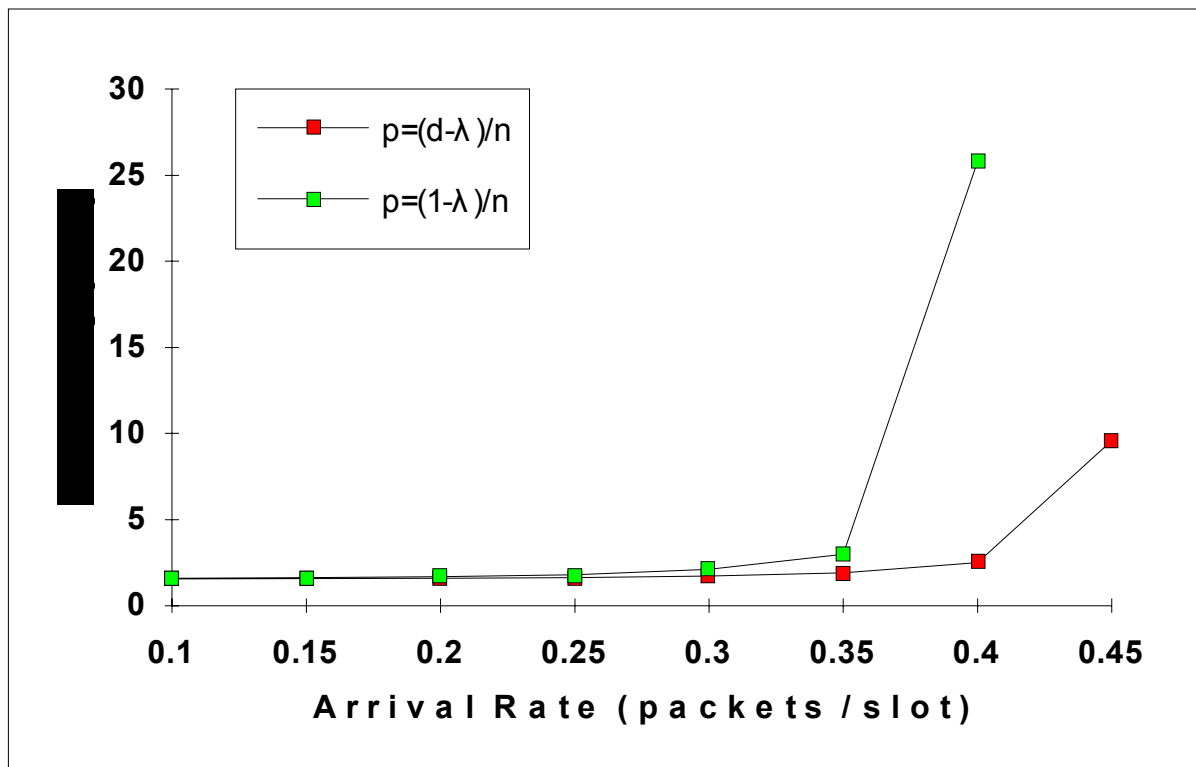
Από τα αποτελέσματα της προσομοίωσης του ιδανικού αλγορίθμου *ALOHA* κύριο συμπέρασμα που εξάγεται είναι ότι, καθώς αυξάνεται η παράμετρος τοποθεσίας  $k$  της κατανομής *Pareto*, η προβλεψιμότητα της καθυστέρησης ενός τυχαία επιλεγμένου πακέτου με βάση τη γνώση της μέσης τιμής της καθυστέρησης πακέτου μειώνεται, λόγω της υψηλής διασποράς που χαρακτηρίζει την κατανομή καθυστέρησης. Η κατανομή της καθυστέρησης πακέ-





**Σχ.8 :** Η γραφική παράσταση της βέλτιστης μέσης καθυστέρησης πακέτων στο σύστημα που επιτεύχθηκε με την επιλογή της κατάλληλης πιθανότητας επαναμετάδοσης  $(d-l)/n$  για κάθε  $\lambda$  και της μέσης καθυστέρησης που επιτυγχάνεται όταν η πιθανότητα αναμετάδοσης δίδεται από  $(1-l)/n$ , σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$  για την τιμή  $k=0.8$  της κατανομής *Pareto*.

του χαρακτηρίζεται εν γένει από μικρή μέση τιμή (εκτός από την περίπτωση που ο ρυθμός αφίξεων πακέτων πλησιάζει το μέγιστο *throughput* του αλγορίθμου), και τυπική απόκλιση πολλαπλάσια της μέσης τιμής.



Σχ.9 : Η γραφική παράσταση της βέλτιστης μέσης καθυστέρησης πακέτων στο σύστημα που επιτεύχθηκε με την επιλογή της κατάλληλης πιθανότητας επαναμετάδοσης  $(d-l)/n$  για κάθε  $\lambda$  και της μέσης καθυστέρησης που πετυχαίνεται με πιθανότητα αναμετάδοσης ίση με  $(1-l)/n$ , σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$  για την τιμή  $k=0.95$  της κατανομής *Pareto*.

## ΚΕΦΑΛΑΙΟ 4

### Οι Αλγόριθμοι Στοιβάς

Στο Κεφάλαιο αυτό θα μελετηθούν δύο αλγόριθμοι ελεύθερης προσπέλασης που επιλύουν τις συγκρούσεις των πακέτων χρησιμοποιώντας μία *στοίβα* (stack) με απεριόριστο αριθμό κελιών [2], [8], [34] και [35]. Ο πρώτος αλγόριθμος χωρίζει στατιστικά τα πακέτα που μετέχουν σε μία σύγκρουση σε δύο υποσύνολα, τα οποία καταλαμβάνουν ανά ένα τα δύο πρώτα κελιά της στοίβας, ενώ ο δεύτερος χωρίζει στατιστικά τα πακέτα που μετέχουν σε μία σύγκρουση σε τρία υποσύνολα, τα οποία καταλαμβάνουν ανά ένα τα τρία πρώτα κελιά της στοίβας. Στην εξέταση του δεύτερου αλγορίθμου οδηγηθήκαμε από την παρατήρηση ότι ένα αρκετά μεγάλο ποσοστό (της τάξης του 30%) των συγκρούσεων πακέτων είναι πολλαπλότητας ίσης με τρία.

#### *4.1 Εισαγωγικά για την προσομοίωση των αλγορίθμων στοίβας*

Οι αλγόριθμοι στοίβας που εξετάζονται είναι οι αλγόριθμοι ελεύθερης πρόσβασης σε στοίβα με άπειρο αριθμό κελιών. Τα “φρέσκα” πακέτα που γεννιούνται στο σύστημα στην διάρκεια ενός χρονικού τμήματος, εισάγονται στην αρχή του επόμενου χρονικού τμήματος στο πρώτο κελί της στοίβας, μαζί με τα πακέτα που καταλήγουν σε αυτό το κελί μετά από τον στατιστικό χωρισμό που ακολουθεί την σύγκρουση που έλαβε χώρα στο δεδομένο χρονικό τμήμα (εφόσον υπήρχε τέτοια σύγκρουση). Τα παραπάνω πακέτα μεταδίδονται κατά τη διάρκεια του επόμενου χρονικού τμήματος. Τα πακέτα που μετέχουν σε μία σύγκρουση χρησιμοποιούν μία γεννήτρια τυχαίων αριθμών και ανάλογα με το αποτέλεσμα πηγαίνουν στο

πρώτο ή στο δεύτερο (στοίβα δύο κελιών) ή στο τρίτο κελί (στοίβα τριών κελιών) της στοίβας. Η υλοποίηση αυτών των αλγορίθμων στην πράξη είναι σχετικά εύκολη, αφού οι χρήστες δεν χρειάζεται να έχουν γνώση της κατάστασης του καναλιού πριν μεταδώσουν για πρώτη φορά. Από τη στιγμή όμως που θα εισαχθούν στη στοίβα (γεννήσουν “φρέσκο” πακέτο) πρέπει να παρακολουθούν ανελλιπώς την ανάδραση του καναλιού για να καθορίζουν την συμπεριφορά τους στα ακόλουθα χρονικά τμήματα, μέχρις ότου μεταδώσουν επιτυχώς το πακέτο τους (*περιορισμένη αίσθηση της ανάδρασης του καναλιού, limited feedback sensing*).

Κατά την εφαρμογή των αλγορίθμων στοίβας, θεωρούμε *άπειρο αριθμό χρηστών* και υποθέτουμε ότι *κάθε “φρέσκο” πακέτο φτάνει σε διαφορετικό χρήστη στο σύστημα*.

Ο κώδικας που υλοποιεί τους δύο παραπάνω αλγορίθμους στοίβας παρατίθεται στο Παράρτημα 2.

Η προσομοίωση των αλγορίθμων εκτελέστηκε σε **UNIX** σταθμό εργασίας **SUN SPARCstation 5**. Σε κάθε εκτέλεση της προσομοίωσης (run) προσομοιώθη-κε η μετάδοση ενός εκατομμυρίου πακέτων και έγιναν οι απαραίτητες προσομοιώσεις (runs) ώστε να βρεθεί η μικρότερη μέση καθυστέρηση των πακέτων στο σύστημα για κάθε ρυθμό άφιξης  $\lambda$  και για διάφορες τιμές της παραμέτρου τοποθεσίας  $k$  της εξεταζόμενης κατανομής **Pareto** των χρόνων μεταξύ αφίξεων διαδοχικών πακέτων. Ο προσομοιωτής υπολογίζει και εκτιμάει διάφορες μετρικές όπως για παράδειγμα τον αριθμό των χρονικών τμημάτων (slots) που προσομοιώ-θηκαν για να μεταδοθεί επιτυχημένα ο απαιτούμενος αριθμός του ενός εκατομμυρίου πακέτων, τη συνολική καθυστέρηση των πακέτων στο σύστημα, τη μέση τιμή της καθυστέρησης, τη διασπορά της καθυστέρησης, κ.λ.π.

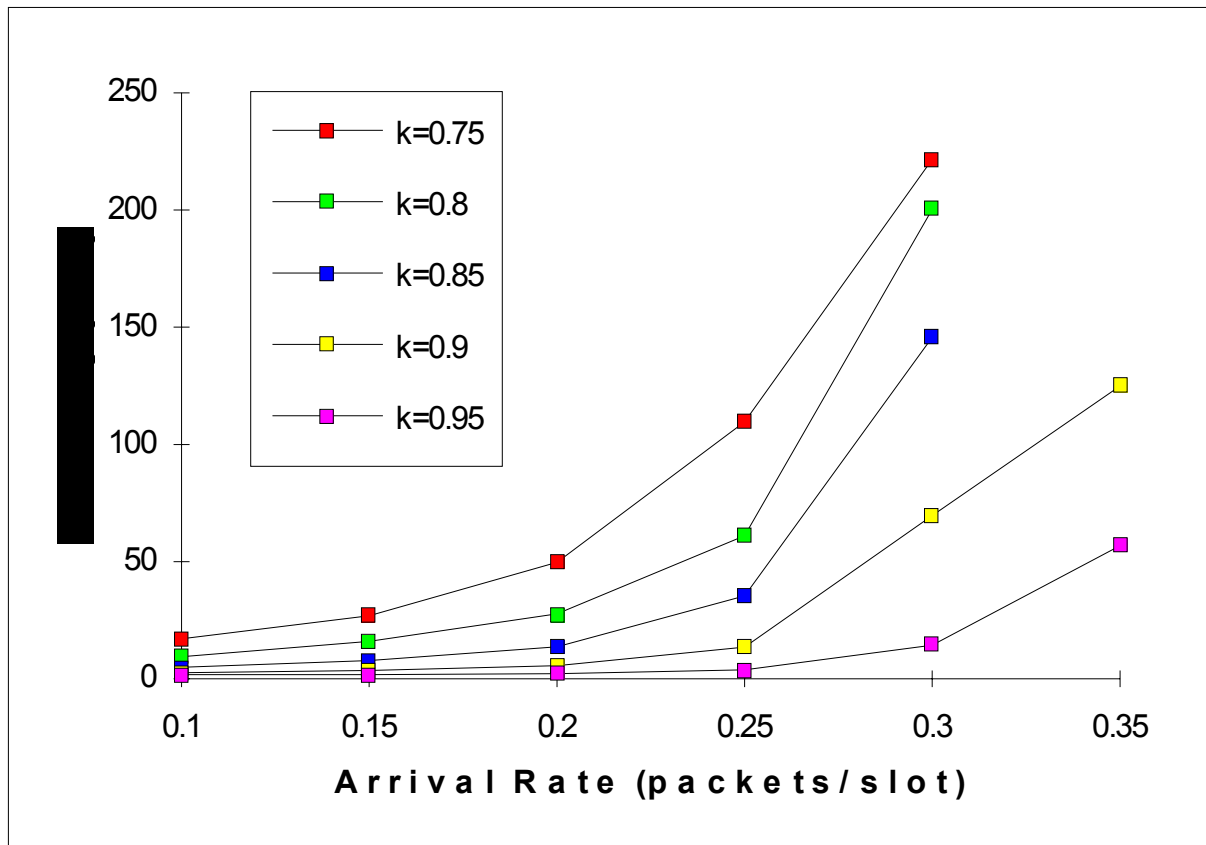
#### **4.2 Αποτελέσματα της προσομοίωσης του αλγορίθμου στοίβας με δυαδικό χωρισμό των συγκρουόμενων πακέτων**

Η συμπεριφορά της μέσης καθυστέρησης πακέτου σαν συνάρτηση του συνολικού ρυθμού αφίξεων πακέτων στο σύστημα, εξετάζεται στη συνέχεια. Η συμπεριφορά αυτή μεταβάλλεται με την μεταβολή των παραμέτρων της κατανομής **Pareto**. Στα σχήματα **Σχ.10** και **Σχ.11** φαίνεται η συμπεριφορά της καθυστέρησης πακέτων με την αλλαγή της παραμέτρου τοποθεσίας  $k$ , για διάφορους ρυθμούς άφιξης πακέτων στο σύστημα  $\lambda$ . Κάθε σημείο στα αναφερόμενα σχήματα προήλθε από προσομοίωση ενός εκατομμυρίου πακέτων. Στον κάθετο άξονα του **Σχ.10** παρουσιάζεται η βέλτιστη μέση καθυστέρηση πακέτων στο σύστημα, η οποία αντιστοιχεί στην κατάλληλη πιθανότητα διαχωρισμού των συγκρουόμενων πακέτων ανάμεσα στα δύο πρώτα κελιά της στοίβας (οι αντίστοιχες βέλτιστες πιθανότητες

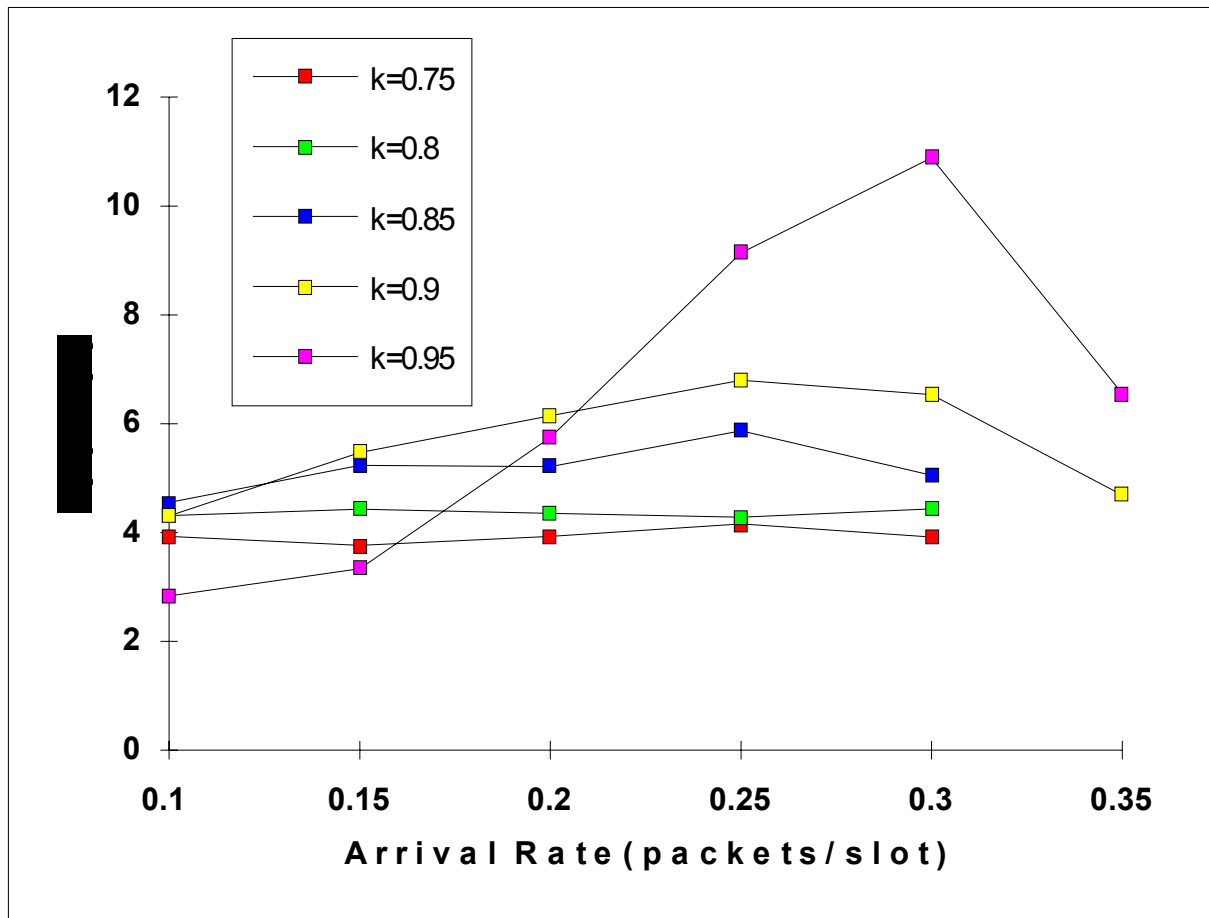
περιλαμβάνονται στον **Πίνακα 4**, ο τρόπος με τον οποίο οι πιθανότητες αυτές βρέθηκαν για κάθε  $\lambda$ , μέσω προσομοίωσης, θα εξηγηθεί στη συνέχεια).

Από το **Σχ.10** παρατηρούμε ότι η αύξηση της βέλτιστης μέσης καθυστέρησης ξεκινάει νωρίτερα (από μικρότερες τιμές του  $\lambda$ ) όσο μειώνεται η παράμετρος τοποθεσίας  $k$  της κατανομής **Pareto**. Στη μέγιστη τιμή του  $k$  που εξετάσαμε ( $k=0.95$ ) παρατηρούμε από τη γραφική παράσταση ότι η βέλτιστη μέση καθυστέρηση διατηρείται μικρότερη από 15 slots για ρυθμούς αφίξεων πακέτων μικρότερους από 0.3, ενώ για την μικρότερη εξεταζόμενη τιμή του  $k$  ( $k=0.75$ ) η τιμή της βέλτιστης μέσης καθυστέρησης αυξάνεται συνεχώς αρχίζοντας από 17 slots για ρυθμό άφιξης πακέτων 0.1 και φτάνοντας μέχρι 200 slots όταν ο ρυθμός άφιξης πακέτων είναι ίσος με 0.3. Επιπλέον παρατηρούμε ότι για δεδομένη τιμή του ρυθμού άφιξης πακέτων  $\lambda$ , η βέλτιστη μέση καθυστέρηση μειώνεται αισθητά καθώς αυξάνεται το  $k$ . Για παράδειγμα για  $\lambda=0.25$ , η βέλτιστη μέση καθυστέρηση μειώνεται από 109.719 slots για  $k=0.75$  σε 3.812 slots για  $k=0.95$ . Συγκρίνοντας με το αντίστοιχο σχήμα για τον ιδανικό ευσταθή αλγόριθμο **ALOHA** (**Σχ. 6**), παρατηρούμε ότι οι βέλτιστες μέσες καθυστερήσεις του αλγο-ρίθμου στοίβας είναι μεγαλύτερες και τα **throughputs** μικρότερα από τα αντίστοιχα του ιδανικού **ALOHA**.

Στο **Σχ.11** φαίνεται ότι καθώς αυξάνεται η παράμετρος τοποθεσίας  $k$  της κατανομής **Pareto** η προβλεψιμότητα της καθυστέρησης ενός τυχαία επιλεγμένου πακέτου με βάση τη γνώση της μέσης καθυστέρησης μειώνεται. Ενώ για  $k=0.75$  το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης διατηρείται κοντά στο 4, για  $k=0.95$  κυμαίνεται από 3 μέχρι και **10(!)**, γεγονός που επιτρέπει στην καθυστέρηση ενός τυχαία επιλεγμένου πακέτου να μεταβάλλεται πολύ γύρω από τη μέση τιμή της. Αυτό οδηγεί στο συμπέρασμα ότι γνωρίζοντας μόνο τις μέσες τιμές της καθυστέρησης πακέτου δεν μπορούμε να έχουμε σημαντικές ελπίδες πρόβλεψης της καθυστέρησης ενός τυχαία επιλεγμένου πακέτου, μια και στην χειρότερη περίπτωση η τελευταία μπορεί να είναι και 10 φορές μεγαλύτερη από την μέση τιμή.



**Σχ.10 :** Η γραφική παράσταση της βέλτιστης μέσης καθυστέρησης πακέτων στο σύστημα σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$ , για διάφορες τιμές της παραμέτρου τοποθεσίας  $k$  της κατανομής *Pareto*.



**Σχ.11** : Η γραφική παράσταση του πηλίκου της τυπικής απόκλισης της  $\sigma(D)$  διά της μέσης τιμής  $E(D)$  της καθυστέρησης σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$ , για διάφορες τιμές της παραμέτρου τοποθεσίας  $k$  της κατανομής *Pareto*.

Οι βέλτιστες πιθανότητες διαχωρισμού των πακέτων που συμμετέχουν σε μία σύγκρουση, έτσι ώστε να επιτευχθεί η βέλτιστη μέση καθυστέρηση των πακέτων στο σύστημα, που φαίνονται στον ακόλουθο πίνακα (**Πίνακας 4**) προήλθαν από εξαντλητικό ψάξιμο μέσω προσομοίωσης. Σύμφωνα με τις πιθανότητες που φαίνονται στον πίνακα, κάθε πακέτο που μετέχει σε μία σύγκρουση αποφασίζει να παραμείνει στο πρώτο κελί της στοίβας και συνεπώς να επαναμεταδώσει στο επόμενο χρονικό τμήμα. Από τον **Πίνακα 4** παρατηρούμε ότι η βέλτιστη πιθανότητα διαχωρισμού διατηρείται γύρω στο  $1/2$ , με εξαίρεση τις περιπτώσεις μικρού  $k$  ή μεγάλων  $k$  και  $\lambda$ . Οι περιπτώσεις με --- είναι ασταθείς (δηλαδή, δεν βρέθηκε τιμή της πιθανότητας διαχωρισμού  $0 < p < 1$ , τέτοια ώστε να προκύπτει ευσταθές σύστημα).

$p$	$k=0.75$	$k=0.8$	$k=0.85$	$k=0.9$	$k=0.95$

$\lambda=0.1$	0.47	0.48	0.48	0.48	0.46
$\lambda=0.15$	0.47	0.5	0.47	0.46	0.48
$\lambda=0.2$	0.47	0.47	0.46	0.48	0.49
$\lambda=0.25$	0.46	0.46	0.46	0.49	0.52
$\lambda=0.3$	0.53	0.5	0.52	0.48	0.5
$\lambda=0.35$	----	----	----	0.55	0.55

**Πίνακας 4 :** Η βέλτιστη πιθανότητα διαχωρισμού των συγκρουόμενων πακέτων που χρησιμοποιήθηκε στα σχήματα **Σχ.10** και **Σχ.11**.

#### **4.3 Αποτελέσματα της προσομοίωσης του αλγορίθμου στοίβας με τριαδικό χωρισμό των συγκρουόμενων πακέτων**

Η συμπεριφορά της καθυστέρησης πακέτου σαν συνάρτηση του συνολικού ρυθμού αφίξεων πακέτων στο σύστημα, εξετάζεται στη συνέχεια. Η συμπεριφορά αυτή μεταβάλλεται με την μεταβολή των παραμέτρων της κατανομής *Pareto*. Το σύστημα βρέθηκε ότι είναι ευσταθές για τιμές της παραμέτρου τοποθεσίας  $k \geq 0.7$  χρονικά τμήματα, ενώ οι τιμές της παραμέτρου σχηματισμού  $a$  οδηγούν σε ευσταθή συστήματα ανήκουν στο διάστημα [1, 1.646]. Το μέγιστο *throughput* αυτού του αλγορίθμου χρησιμοποιώντας την κατανομή *Pareto* για την μοντελοποίηση των χρόνων μεταξύ αφίξεων των πακέτων που εισάγονται στο σύστημα είναι **0.41315** το οποίο είναι περίπου ίδιο με το μέγιστο *throughput* του ίδιου αλγορίθμου για χρόνους μεταξύ αφίξεων πακέτων κατανομημένους σύμφωνα με την Εκθετική κατανομή (*Poisson* αφίξεις) που είναι **0.4** [2] και [4].

Στα σχήματα **Σχ.12** και **Σχ.13** φαίνεται η συμπεριφορά της καθυστέρησης πακέτων με την αλλαγή της παραμέτρου τοποθεσίας  $k$ , για διάφορους ρυθμούς άφιξης πακέτων στο σύστημα  $\lambda$ . Κάθε σημείο στα αναφερόμενα σχήματα προήλθε από προσομοίωση ενός εκατομμυρίου πακέτων. Στον κάθετο άξονα του **Σχ.12** παρουσιάζεται η βέλτιστη μέση καθυστέρηση πακέτων στο σύστημα, η οποία αντιστοιχεί στις κατάλληλες πιθανότητες διαχωρισμού των συγκρουόμενων πακέτων ανάμεσα στα τρία πρώτα κελιά της στοίβας (οι αντίστοιχες πιθανότητες παρουσιάζονται στον **Πίνακα 5**. Ο τρόπος με τον οποίο οι πιθανότητες αυτές βρέθηκαν για κάθε  $\lambda$ , μέσω προσομοίωσης, θα εξηγηθεί στη συνέχεια).

Από το **Σχ.12** παρατηρούμε ότι η αύξηση της βέλτιστης μέσης καθυστέρησης ξεκινάει νωρίτερα (από μικρότερες τιμές του  $\lambda$ ), όσο μειώνεται η παράμετρος τοποθεσίας  $k$  της κατανομής *Pareto*. Στη μέγιστη τιμή του  $k$  που εξετάσαμε ( $k=0.95$ ) παρατηρούμε από τη

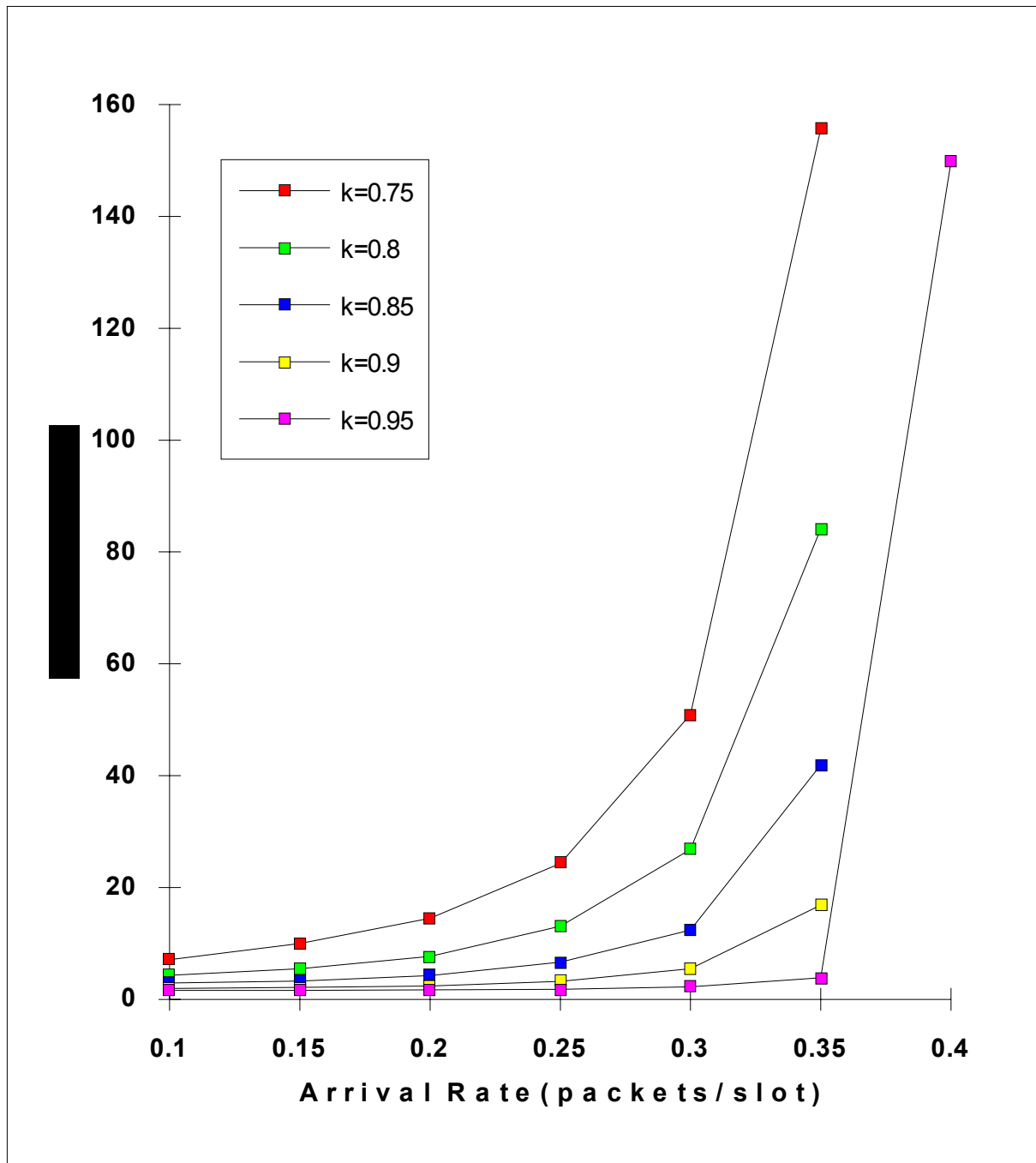


γραφική παράσταση ότι η βέλτιστη μέση καθυστέρηση διατηρείται μικρότερη από 4 slots για ρυθμούς αφίξεων πακέτων μικρότε-ρους από 0.35, ενώ για την μικρότερη εξεταζόμενη τιμή του  $k$  ( $k=0.75$ ) η τιμή της βέλτιστης μέσης καθυστέρησης αρχίζει από 7 slots για ρυθμό άφιξης πακέτων 0.1 και φτάνει μέχρι και 155 slots για ρυθμό άφιξης 0.35. Επιπλέον παρατηρούμε ότι για δεδομένη τιμή του ρυθμού άφιξης πακέτων  $\lambda$ , η βέλτιστη μέση καθυστέρηση μειώνεται αισθητά καθώς αυξάνεται το  $k$ .

Στο **Σχ.13** φαίνεται ότι καθώς αυξάνεται η παράμετρος τοποθεσίας  $k$  της κατανομής *Pareto* η προβλεψιμότητα της καθυστέρησης ενός τυχαία επιλεγμέ-νου πακέτου με βάση τη γνώση της μέσης καθυστέρησης μειώνεται. Για  $k=0.75$  το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης κυμαίνεται από 4 μέχρι 7, ενώ για  $k=0.95$  κυμαίνεται από 1.5 μέχρι και **13(!)**, γεγονός που επιτρέπει στην καθυστέρηση ενός τυχαία επιλεγμένου πακέτου να μεταβάλλεται πολύ γύρω από τη μέση τιμή της. Αυτό οδηγεί στο συμπέρασμα ότι γνωρίζοντας μόνο τις μέσες τιμές της καθυστέρησης πακέτου δεν μπορούμε να έχουμε σημαντικές ελπίδες πρόβλεψης της καθυστέρησης ενός τυχαία επιλεγμέ-νου πακέτου, μια και στην χειρότερη περίπτωση η τελευταία μπορεί να είναι και 13 φορές μεγαλύτερη από την μέση τιμή.

Οι βέλτιστες πιθανότητες διαχωρισμού των συγκρουόμενων πακέτων στα τρία πρώτα κελιά της στοίβας που φαίνονται στον ακόλουθο πίνακα (**Πίνακα 5**), προήλθαν από εξαντλητικό ψάξιμο μέσω προσομοίωσης. Οι παράμετροι  $p_1$  και  $p_2$  ορίζουν τα διαστήματα τιμών σύμφωνα με τα οποία διαχωρίζονται τα συγκρουό-μενα πακέτα ανάμεσα στα τρία πρώτα κελιά της στοίβας. Δηλαδή, το κάθε συγκρουόμενο πακέτο αποφασίζει να πάει στο πρώτο, στο δεύτερο ή στο τρίτο κελί της στοίβας, αν ο τυχαίος αριθμός που “γεννάει” ο αντίστοιχος χρήστης βρίσκεται στο διάστημα  $(0, p_1)$ ,  $[p_1, p_2]$  και  $(p_2, 1)$ , αντίστοιχα.

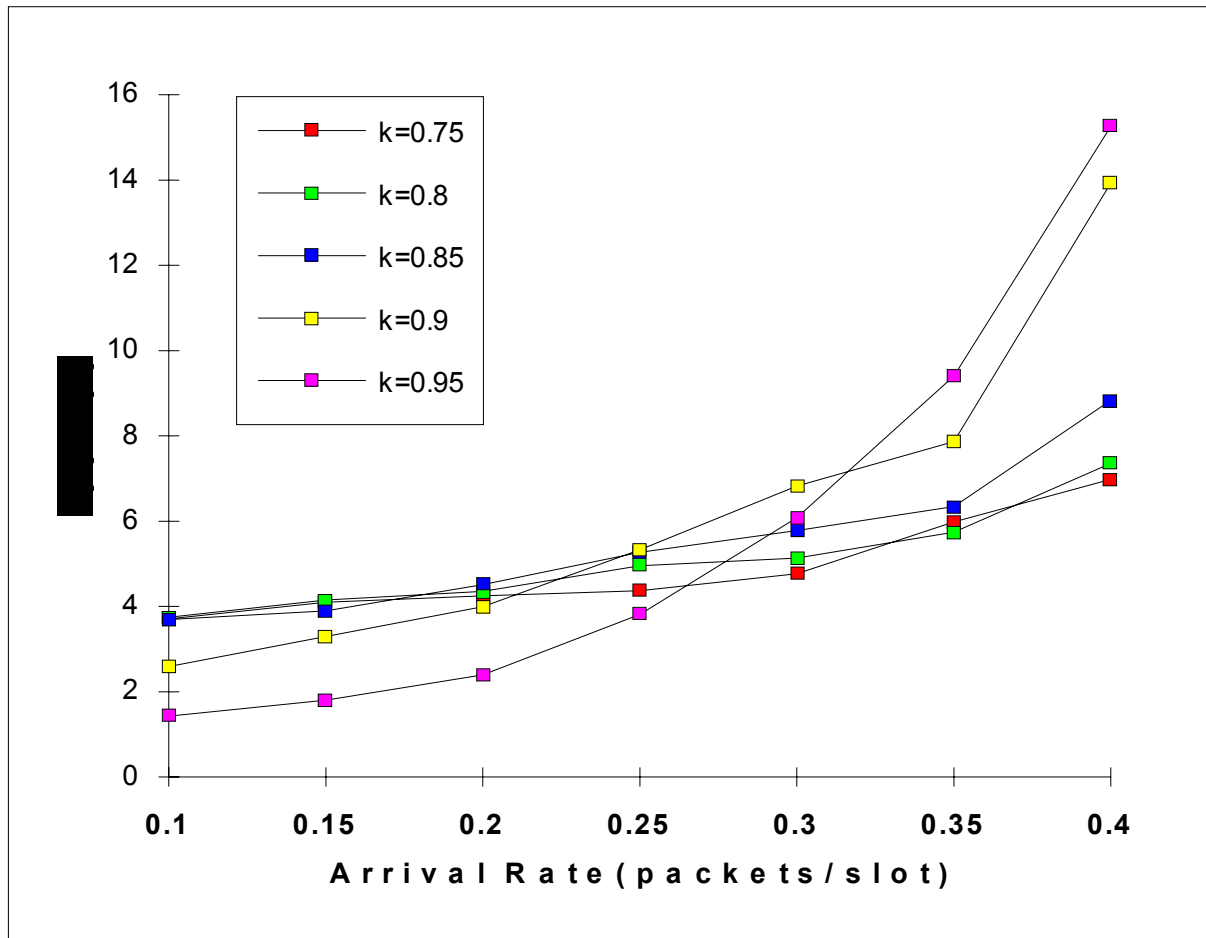
Η ευαισθησία της κατανομής της καθυστέρησης στην επιλογή των παραμέτρων  $p_1$  και  $p_2$  φαίνεται στον **Πίνακα 6**, όπου  $k=0.9$ ,  $\lambda=0.4$ ,  $p_1=0.33$  και  $p_2$  μεταβάλλεται μεταξύ 0.60 και 0.66. Παρατηρούμε την σημαντική αλλαγή της μέσης τιμής και της τυπικής απόκλισης της καθυστέρησης πακέτου, καθώς και του αριθμού των πακέτων που μένουν στην *στοίβα* (stack) μετά το τέλος της προσομοίωσης. Επισημαίνεται ότι η επιλογή του  $p_1=0.33$  είναι αυθαίρετη, η



**Σχ.12 :** Η γραφική παράσταση της βέλτιστης μέσης καθυστέρησης πακέτων στο σύστημα σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$ , για διάφορες τιμές της παραμέτρου τοποθεσίας  $k$  της κατανομής *Pareto*.

αντίστοιχη βέλτιστη επιλογή των παραμέτρων  $p_1$  και  $p_2$  ( $p_1=0.28$ ,  $p_2=0.60$ ), φαίνεται στον **Πίνακα 5**. Η τιμή της παραμέτρου  $p_1=0.33$  επιλέχτηκε ώστε να γίνει εμφανής η υψηλή

ευαισθησία της μέσης καθυστέρησης πακέτου στις μεταβολές των τιμών των παραμέτρων  $p_1$  και  $p_2$ .



**Σχ.13 :** Η γραφική παράσταση του πηλίκου της τυπικής απόκλισης  $\sigma(D)$  διά της μέσης τιμής  $E(D)$  της καθυστέρησης σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$ , για διάφορες τιμές της παραμέτρου τοποθεσίας  $k$  της κατανομής *Pareto*.

$p_1 / p_2$	$\kappa=0.75$	$\kappa=0.8$	$\kappa=0.85$	$\kappa=0.9$	$\kappa=0.95$
$\lambda= 0.1$	0.31/0.67	0.31/0.62	0.3/0.63	0.33/0.6	0.31/0.66
$\lambda= 0.15$	0.32/0.65	0.29/0.61	0.3/0.63	0.3/0.59	0.3/0.64
$\lambda= 0.2$	0.32/0.64	0.3/0.59	0.3/0.63	0.28/0.63	0.28/0.66
$\lambda= 0.25$	0.29/0.65	0.28/0.64	0.3/0.6	0.32/0.65	0.33/0.63
$\lambda= 0.3$	0.31/0.63	0.28/0.6	0.28/0.63	0.29/0.64	0.33/0.63
$\lambda= 0.35$	0.28/0.64	0.29/0.59	0.28/0.59	0.28/0.62	0.3/0.59
$\lambda= 0.4$	0.32/0.62	0.3/0.65	0.31/0.64	0.28/0.60	0.28/0.60

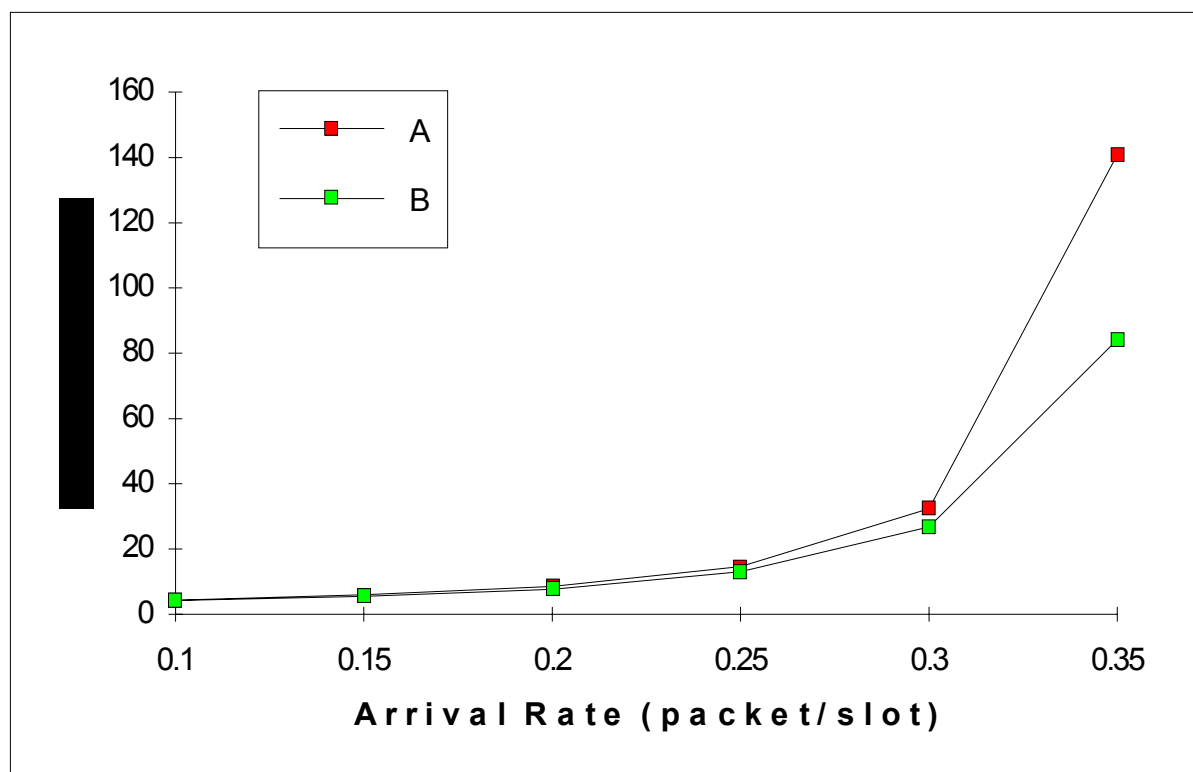
**Πίνακας 5 :** Οι τιμές των παραμέτρων  $p_1$  και  $p_2$  που ορίζουν τις βέλτιστες πιθανότητες διαχωρισμού των συγκρουόμενων πακέτων που χρησιμοποιήθηκαν στα σχήματα **Σχ.12** και **Σχ.13**.

$p_1 / p_2$	$E(D)$	$\sigma(D)$	stack
0.33 / 0.66	507.642	8.94	984
0.33 / 0.65	667.719	11.133	612
0.33 / 0.64	713.614	9.019	28
0.33 / 0.63	650.316	8.937	43
0.33 / 0.62	1029.402	10.963	805
0.33 / 0.61	834.04	10.096	1091
0.33 / 0.6	542.17	9.163	508

**Πίνακας 6 :** Η μεταβολή των χαρακτηριστικών της καθυστέρησης του αλγορίθμου σε σχέση με τις παραμέτρους που καθορίζουν τον τρόπο που μοιράζονται τα πακέτα ανάμεσα στα τρία κάτω κελιά της στοίβας όταν  $\lambda=0.4$  και  $k=0.9$ . (Η τέταρτη στήλη απεικονίζει το περιεχόμενο της στοίβας μετά το τέλος της προσομοίωσης).

Στα σχήματα **Σχ.14** και **Σχ.15** απεικονίζεται η αλλαγή της συμπεριφοράς καθυστέρησης του αλγορίθμου στοίβας με τριαδικό χωρισμό των συγκρουόμενων πακέτων όταν αντί των βέλτιστων παραμέτρων  $p_1$  και  $p_2$  χρησιμοποιούνται οι απλούστερες και προφανείς (που απλώς χωρίζουν το διάστημα της μονάδας σε τρία ίσα μέρη), για διάφορους ρυθμούς αφίξεων πακέτων στο σύστημα και για τιμές της παραμέτρου τοποθεσίας  $k$  ίσες με 0.8 και 0.95, αντίστοιχα. Και στα δύο σχήματα είναι εμφανές ότι η καμπύλη που αντιστοιχεί στη βέλτιστη λειτουργία του αλγορίθμου (B) επιτυγχάνει τις χαμηλότερες μέσες καθυστερήσεις από την καμπύλη που αντιστοιχεί στον ισοπίθανο στατιστικό διαχωρισμό των συγκρουόμενων πακέτων ανάμεσα στα τρία πρώτα κελιά της στοίβας. Επίσης από τα αποτελέσματα της προσομοίωσης παρατηρούμε ότι η μέση καθυστέρηση αυξάνεται πολύ νωρίτερα στην υποβέλτιστη περίπτωση. Από τα σχήματα **Σχ.14** και **Σχ.15** παρατηρούμε επίσης ότι η αλλαγή στην μέση καθυστέρηση από τις βέλτιστες τιμές γίνεται νωρίτερα αισθητή για μικρότερες τιμές του  $k$ . Όταν το  $k=0.8$  ( $k=0.95$ ) η διαφορά της μέσης καθυστέρησης μεταξύ βέλτιστης και υποβέλτιστης περίπτωσης γίνεται σημαντική για  $\lambda>0.3$  ( $\lambda>0.35$ ).

Συγκρίνοντας τα αποτελέσματα καθυστέρησης πακέτου των αλγορίθμων στοίβας που μελετήσαμε, παρατηρούμε ότι ο αλγόριθμος που χρησιμοποιεί τριαδικό χωρισμό των συγκρουόμενων πακέτων πετυχαίνει πολύ μικρότερες

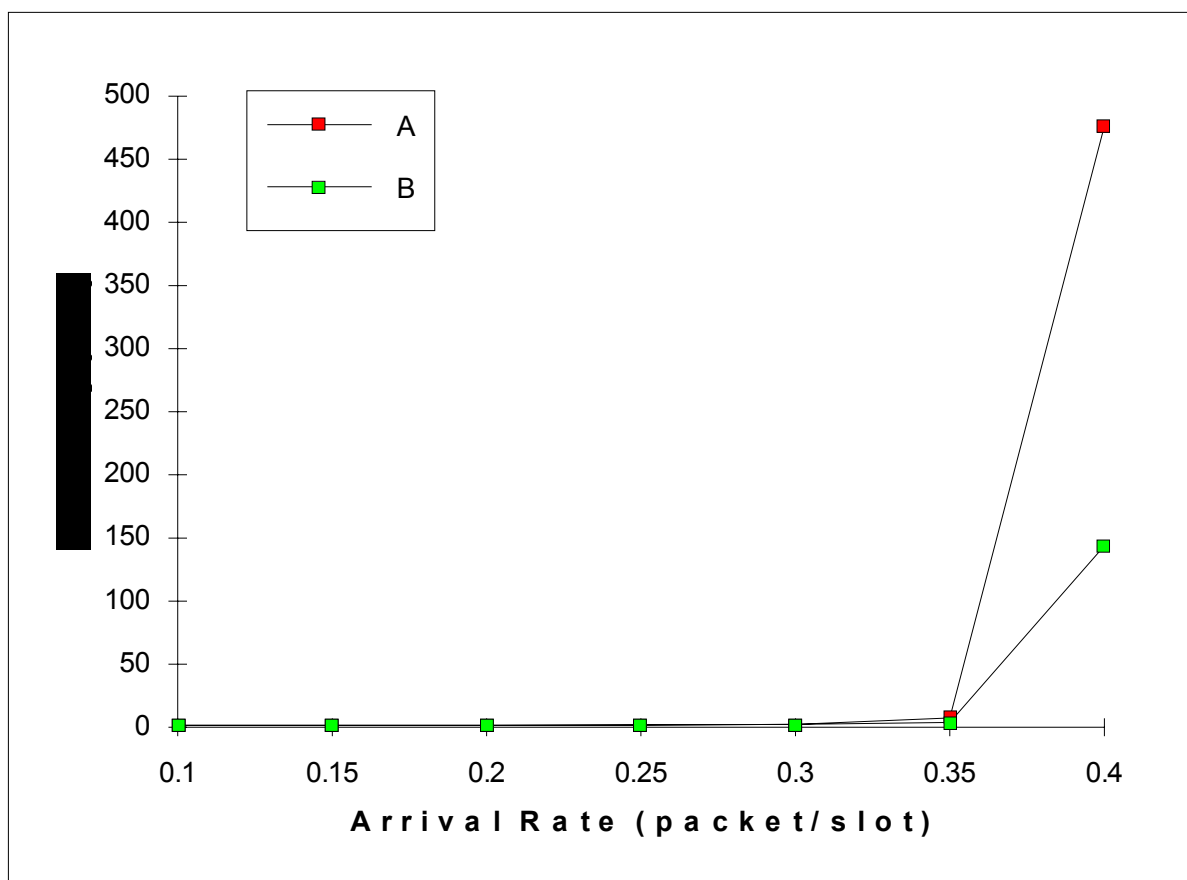


**Σχ.14 :** Η γραφική παράσταση της βέλτιστης μέσης καθυστέρησης πακέτων στο σύστημα που επιτεύχθηκε με την επιλογή των κατάλληλων παραμέτρων ( $p_1$  και  $p_2$ ) που καθορίζουν τον στατιστικό διαχωρισμό των συγκρουόμενων πακέτων ανάμεσα στα τρία πρώτα κελιά της στοίβας (B), και της μέσης καθυστέρησης που επιτυγχάνεται όταν οι παράμετροι αυτοί είναι ίσες με  $p_1=1/3$  και  $p_2=2/3$  (A), σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$  για  $k=0.8$ .

μέσες καθυστερήσεις, με ταυτόχρονη αύξηση του κλάσματος της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης. Το παραπάνω φαινόμενο *αντιστάθμισης* (trade off) έχει παρατηρηθεί και σε άλλους αλγορίθμους επίλυσης σύγκρουσης με *Poisson* αφίξεις πακέτων [3].

Συγκρίνοντας τα αποτελέσματα καθυστέρησης πακέτων των αλγορίθμων στοίβας με τριαδικό διαχωρισμό και του ιδανικού ευσταθούς αλγορίθμου *ALOHA*, παρατηρούμε ότι ο τελευταίος επιτυγχάνει χαμηλότερες βέλτιστες μέσες καθυστερήσεις, υψηλότερα *throughputs*, και χαμηλότερες τιμές του κλάσματος τυπικής απόκλισης διά της μέσης τιμής

της καθυστέρησης πακέτων. Οι διαφορές όμως δεν είναι πολύ μεγάλες. Υπενθυμίζουμε εδώ ότι ο αλγόριθμος *ALOHA* που εξετάσαμε στο Κεφάλαιο 3 είναι ιδανικός, δύσκολος στην υλοποίηση μια και



**Σχ.15 :** Η γραφική παράσταση της βέλτιστης μέσης καθυστέρησης πακέτων στο σύστημα που επιτεύχθηκε με την επιλογή των κατάλληλων παραμέτρων που καθορίζουν τον στατιστικό διαχωρισμό των συγκρουόμενων πακέτων ανάμεσα στα τρία πρώτα κελιά της στοίβας (B), και της μέσης καθυστέρησης που επιτυγχάνεται όταν οι παράμετροι αυτοί είναι ίσες με  $p_1=1/3$  και  $p_2=2/3$  (A), σαν συνάρτηση του ρυθμού άφιξης πακέτων  $\lambda$  για  $k=0.95$ .

υποθέτει την χρήση ιδανικών εκτιμητριών του ρυθμού άφιξης πακέτων  $\lambda$  και του αριθμού των backlogged πακέτων  $n$ . Σε αντίθεση, οι αλγόριθμοι στοίβας που εξετάζονται σε αυτό το Κεφάλαιο δεν χρησιμοποιούν εκτιμήτριες ακρίβειας, είναι από τη φύση τους ευσταθείς και εύκολα υλοποιήσιμοι στην πράξη.

## ΚΕΦΑΛΑΙΟ 5

### Σύγκριση Απόδοσης των Αλγορίθμων

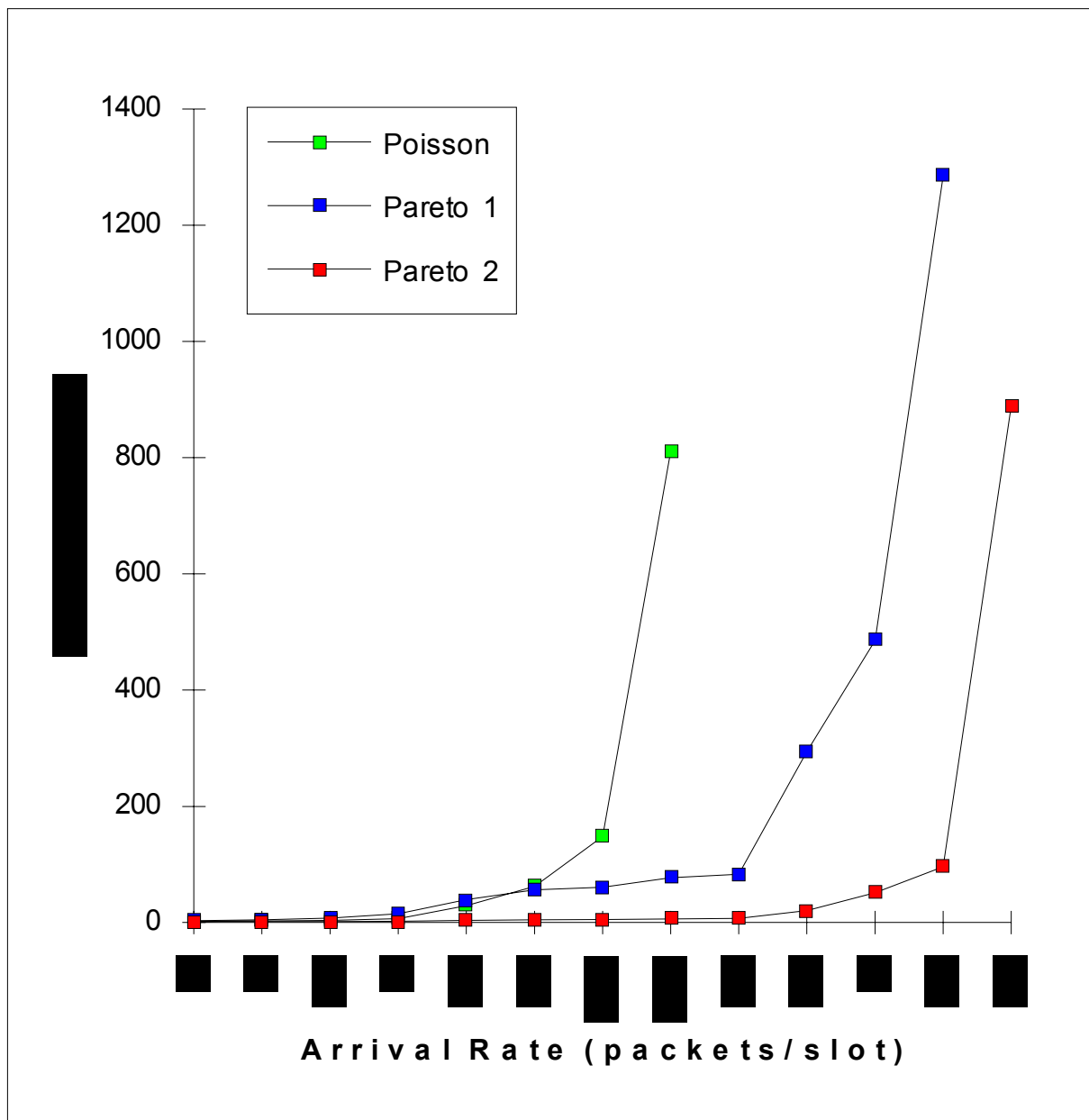
Στο Κεφάλαιο αυτό συγκρίνουμε τα αποτελέσματα της μέσης τιμής και της τυπικής απόκλισης της καθυστέρησης πακέτου για τις κατανομές αφίξεων *Poisson* και *Pareto* με ίδιες πιθανότητες αναμετάδοσης από το backlog (στην περίπτωση αλγορίθμου τύπου *ALOHA*) ή στατιστικού διαχωρισμού των συγκρουό-μενων πακέτων (στην περίπτωση αλγορίθμου στοίβας με τριαδικό διαχωρισμό).

Οι υποθέσεις που γίνονται για την προσομοίωση και ο τρόπος υλοποίησης της είναι οι ίδιες με αυτές που αναφέρθηκαν στα δύο προηγούμενα Κεφάλαια, και δε θεωρείται αναγκαία η επανάληψη τους σε αυτό το σημείο.

**5.1 Σύγκριση της καθυστέρησης του ιδανικού αλγορίθμου *ALOHA* για χρόνους μεταξύ αφίξεων πακέτων κατανεμημένους σύμφωνα με τις κατανομές *Pareto* και *Εκθετική (Poisson αφίξεις)*.**

Στο **Σχ.16** απεικονίζεται η μέση τιμή της καθυστέρησης των πακέτων για τις κατανομές **Pareto 1**, **Pareto 2** και **Poisson** για διάφορους ρυθμούς αφίξεων πακέτων. Στην

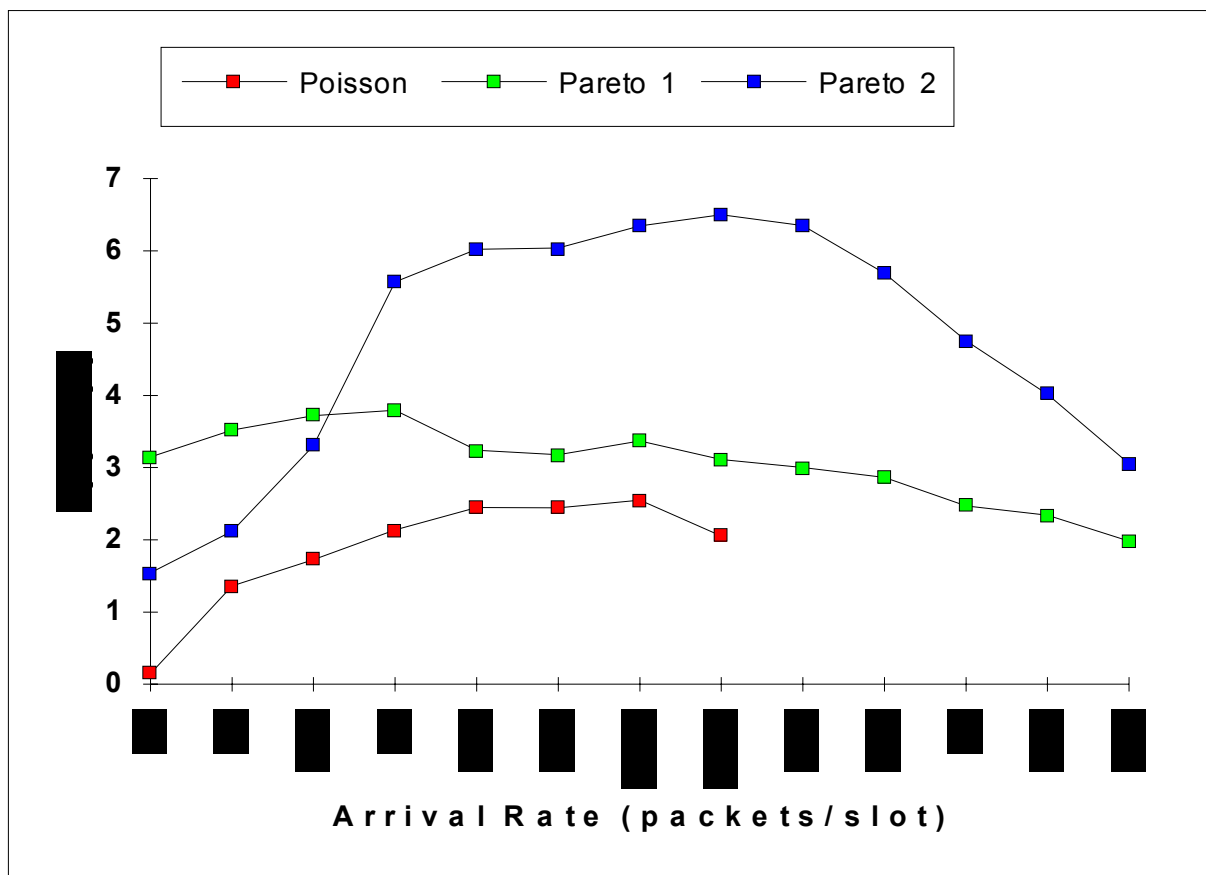
προσομοίωση και των τριών αυτών περιπτώσεων χρησιμοποιήθηκε η ίδια πιθανότητα αναμετάδοσης από το backlog  $p = (1-\lambda)/(n-\lambda)$ , όπου  $\lambda$  είναι ο ρυθμός άφιξης πακέτων και  $n$  ο αριθμός των πακέτων στο backlog. Η παραπάνω πιθανότητα αναμετάδοσης χρησιμοποιήθηκε διότι όπως



**Σχ.16 :** Η γραφική παράσταση απεικονίζει την μέση τιμή της καθυστέρησης των πακέτων σαν συνάρτηση του ρυθμού άφιξης των πακέτων για τις κατανομές χρόνου μεταξύ αφίξεων διαδοχικών πακέτων **Pareto 1**, **Pareto 2** και Εκθετική (**Poisson** αφίξεις).



αποδεικνύεται μεγιστοποιεί το *throughput* του ιδανικού αλγορίθμου *ALOHA* για *Poisson* αφίξεις πακέτων. Ως **Pareto 1** ορίζεται η κατανομή *Pareto* με παράμετρο τοποθεσίας  $k=0.85$  slots, ενώ ως **Pareto 2** η κατανομή *Pareto* με παράμετρο τοποθεσίας  $k=0.95$  slots. Απο το **Σχ.16** παρατηρούμε ότι ο ιδανικός αλγόριθμος *ALOHA* που εξετάζουμε πετυχαίνει καλύτερο *throughput* για αφίξεις



**Σχ.17** : Η γραφική παράσταση απεικονίζει το πηλίκο της τυπικής απόκλισης διά την μέση τιμή της καθυστέρησης πακέτων σαν συνάρτηση του ρυθμού άφιξης πακέτων στο σύστημα για τις κατανομές χρόνου μεταξύ αφίξεων διαδοχικών πακέτων **Pareto 1**, **Pareto 2** και Εκθετική (**Poisson** αφίξεις).

πακέτων χαρακτηριζόμενες από *Pareto* χρόνους μεταξύ αφίξεων από αυτό με εκθετικούς χρόνους μεταξύ αφίξεων (*Poisson* αφίξεις).

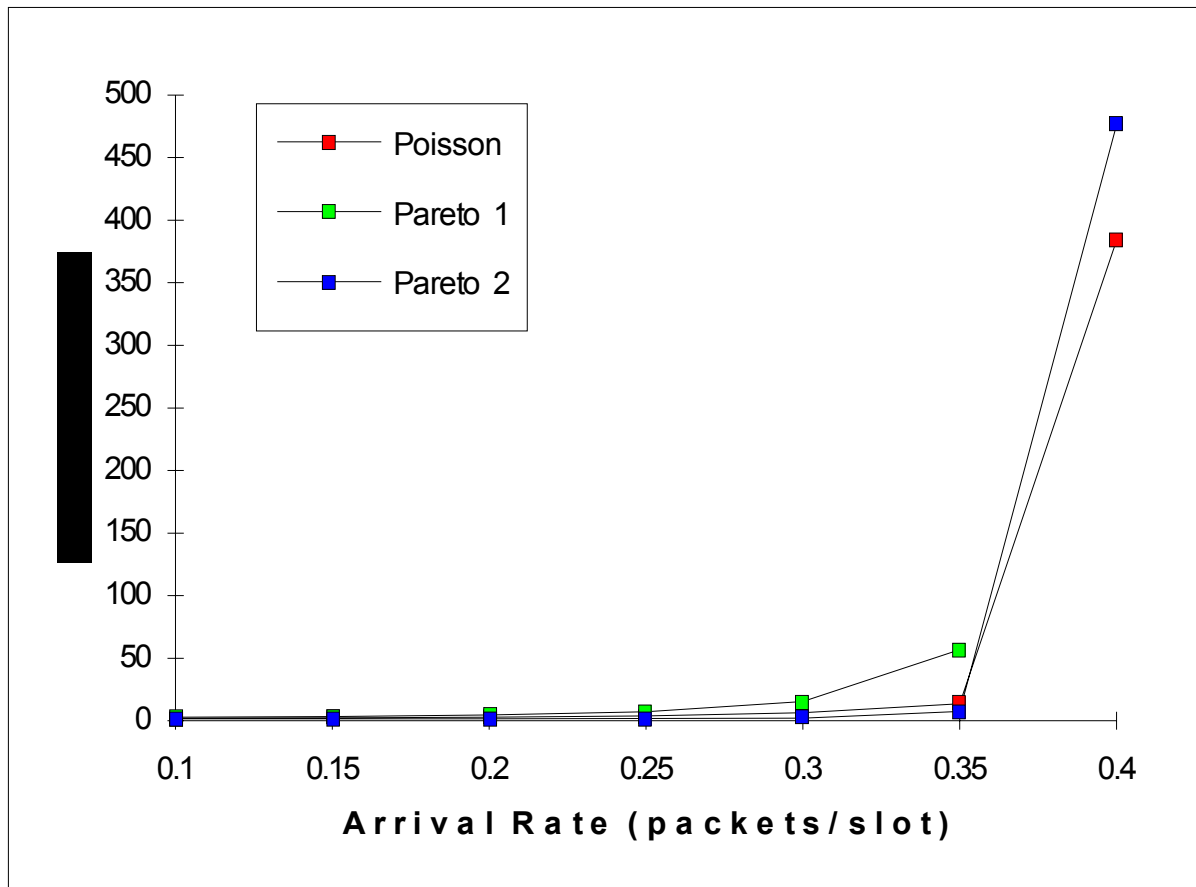
Στο **Σχ.17** παρουσιάζεται το πηλίκο της τυπικής απόκλιση  $\sigma(D)$  διά την μέση τιμή  $E(D)$  της καθυστέρησης των πακέτων για τις τρεις προαναφερθείσες περιπτώσεις, για διάφορους ρυθμούς αφίξεων πακέτων στο σύστημα. Από το **Σχ.17** φαίνεται ότι, για **Poisson** αφίξεις πακέτων, το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης

διατηρείται κοντά στο 2, ενώ για την **Pareto 1** το πηλίκο αυτό κυμαίνεται από 2 έως 4 και για την **Pareto 2** μεταβάλλεται έντονα φτάνοντας μέχρι και το 6. Από τα αποτελέσματα αυτά γίνεται εμφανές ότι στην περίπτωση των **Pareto** κατανεμημένων χρόνων μεταξύ αφίξεων πακέτων έχουμε λίγες ελπίδες πρόβλεψης της καθυστέρησης ενός τυχαία επιλεγμένου πακέτου, με βάση τη γνώση της μέσης τιμής της καθυστέρησης, μια και στη χειρότερη περίπτωση η καθυστέρηση του τυχαία επιλεγμένου πακέτου μπορεί να είναι και 6 φορές μεγαλύτερη από τη μέση τιμή. Σε αντίθεση με αυτήν την περίπτωση, τα αποτελέσματα για **Poisson** αφίξεις είναι καλύτερα αφού στην χειρότερη περίπτωση η καθυστέρηση ενός τυχαία επιλεγμένου πακέτου μπορεί να είναι μέχρι δύο φορές μεγαλύτερη από την μέση τιμή της καθυστέρησης (όμως εφιστάται η προσοχή του αναγνώστη στο ότι η μέση τιμή της καθυστέρησης για **Poisson** αφίξεις πακέτων είναι εν γένει πολύ μεγαλύτερη από την αντίστοιχη μέση τιμή για **Pareto** αφίξεις πακέτων,δες **Σχ.16**, άρα έχουμε ξανά μία εμφάνιση του φαινομένου της αντιστάθμισης στο οποίο πρωτα-ναφερθήκαμε στην παράγραφο 4.2).

### **5.2 Σύγκριση της καθυστέρησης του αλγορίθμου στοίβας με τριαδικό διαχωρισμό των συγκρουόμενων πακέτων για χρόνους μεταξύ αφίξεων πακέτων κατανεμημένους σύμφωνα με τις κατανομές Pareto και Εκθετική (Poisson αφίξεις).**

Στο **Σχ.18** απεικονίζεται η μέση τιμή της καθυστέρησης πακέτων για τις κατανομές του χρόνου μεταξύ αφίξεων **Pareto 1**, **Pareto 2** και Εκθετική (**Poisson** αφίξεις), για διάφορους ρυθμούς αφίξεων πακέτων. Στην προσομοίωση και των τριών αυτών περιπτώσεων χρησιμοποιήθηκαν οι παράμετροι  $p_1=1/3$  και  $p_2=2/3$ , έτσι ώστε να διαχωρίζονται τα συγκρουόμενα πακέτα με την ίδια πιθανότητα ανάμεσα στα τρία πρώτα κελιά της στοίβας. Ως **Pareto 1** ορίζεται και πάλι εκείνη η κατανομή **Pareto** με παράμετρο τοποθεσίας  $k=0.85$  slots, ενώ ως **Pareto 2** εκείνη η κατανομή **Pareto** με παράμετρο τοποθεσίας  $k=0.95$  slots. Οι γεινίαση των καμπύλων μέσης καθυστέρησης για τις περιπτώσεις **Pareto 2** και **Poisson** που παρατηρούμε στο **Σχ.18**, είναι αναμενόμενη με δεδομένο ότι όπως αναφέρθηκε στην παράγραφο 4.2 τα **throughputs** που επιτυγχάνει ο αλγόριθμος στοίβας σε αυτές τις περιπτώσεις είναι παραπλήσια. Η μέση καθυστέρηση πακέτου για την περίπτωση της **Pareto 1** και για ρυθμό άφιξης  $\lambda=0.4$ , αν και το σύστημα παραμένει ευσταθές, ξεπερνάει τα όρια που απεικονίζονται στο **Σχ.18** γι' αυτό δεν θεωρείται απαραίτητη η αναπαράσταση της στο αντίστοιχο σχήμα.

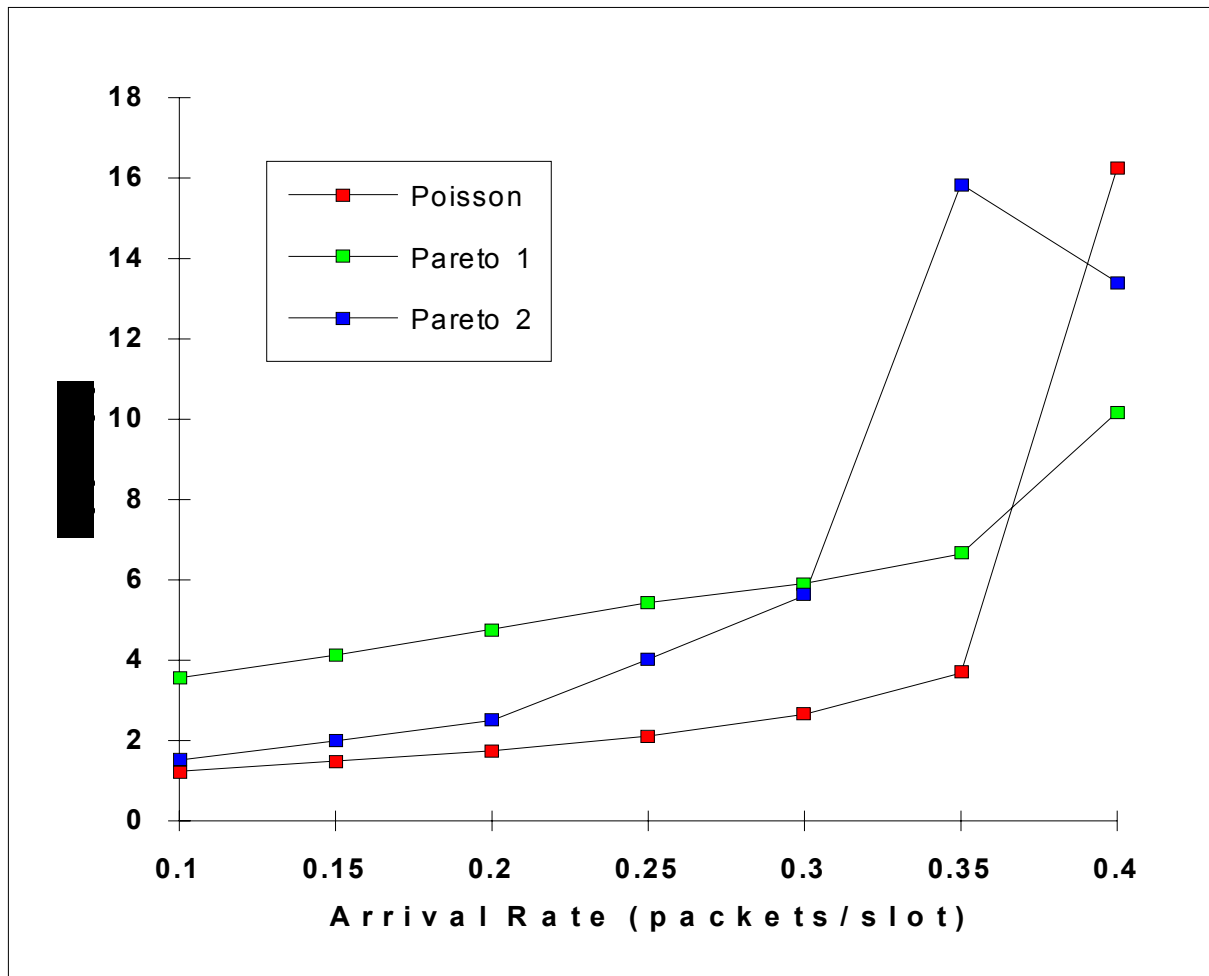
Στο **Σχ.19** παρουσιάζεται το πηλίκο της τυπικής απόκλιση  $\sigma(D)$  διά της μέσης τιμής  $E(D)$  της καθυστέρησης πακέτων για τις τρεις προαναφερθείσες



**Σχ.18 :** Η γραφική παράσταση απεικονίζει την μέση τιμή της καθυστέρησης πακέτων σαν συνάρτηση του ρυθμού αφίξεων για τις κατανομές του χρόνου μεταξύ αφίξεων διαδοχικών πακέτων **Pareto 1**, **Pareto 2** και **Εκθετική (Poisson** αφίξεις).

περιπτώσεις, για διάφορους ρυθμούς αφίξεων πακέτων στο σύστημα. Από το **Σχ.19** φαίνεται ότι, για **Poisson** αφίξεις, το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης διατηρείται μεταξύ 2 και 3 εκτός από την περίπτωση που ο ρυθμός άφιξης γίνεται μεγαλύτερος του 0.35 (για  $\lambda=0.4$  το πηλίκο φτάνει στο 16). Για χρόνους μεταξύ αφίξεων κατανομημένους σύμφωνα με την **Pareto 1** το πηλίκο αυτό κυμαίνεται από 4 μέχρι 10 και για χρόνους μεταξύ αφίξεων κατανομημένους σύμφωνα με την **Pareto 2** το πηλίκο κυμαίνε-ται από 2 μέχρι περίπου 16.

Από τα αποτελέσματα αυτά μπορούμε να συμπεράνουμε ότι η προβλεψι-μότητα της καθυστέρησης πακέτων με βάση τη γνώση της μέσης τιμής για κατανομή χρόνων μεταξύ αφίξεων **Pareto** δεν διαφέρει σημαντικά από την

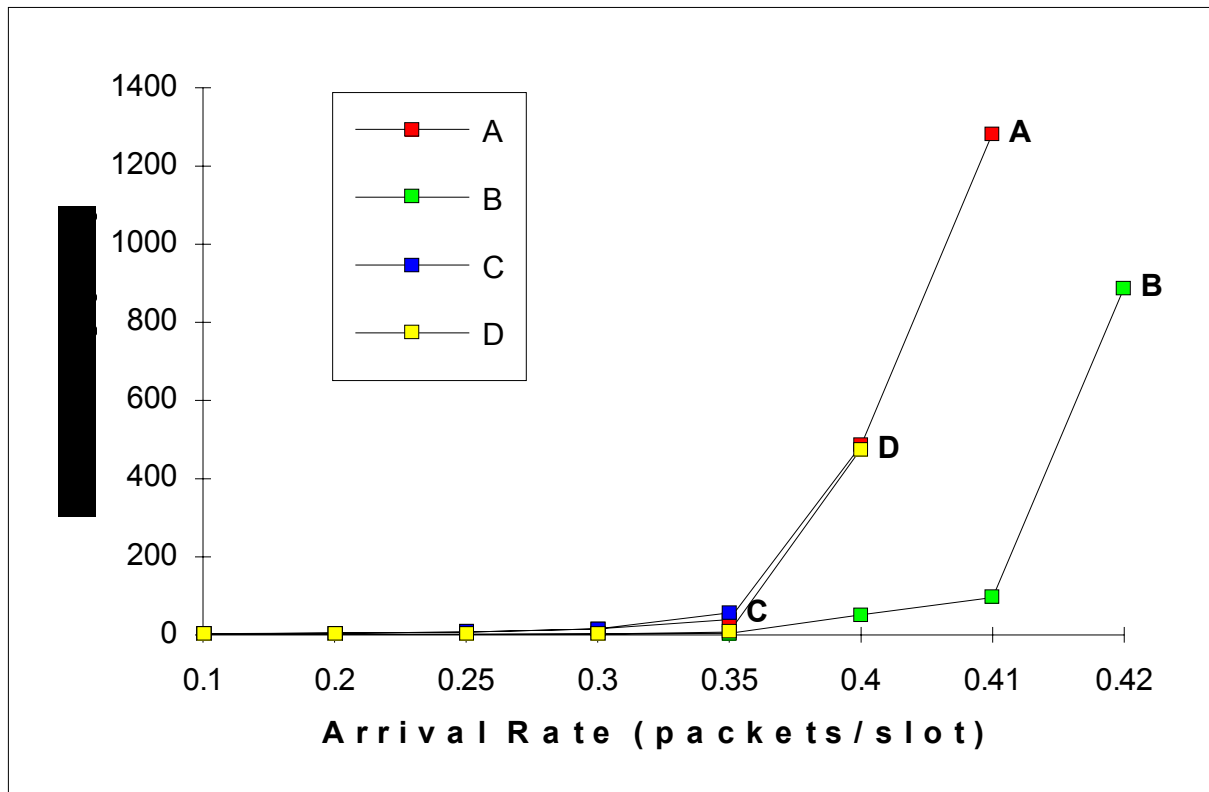


**Σχ.19** : Η γραφική παράσταση απεικονίζει το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης πακέτων σαν συνάρτηση του ρυθμού αφίξεων για τις κατανομές του χρόνου μεταξύ αφίξεων διαδοχικών πακέτων **Pareto 1**, **Pareto 2** και **Εκθετική (Poisson** αφίξεις).

αντίστοιχη προβλεψιμότητα όταν οι χρόνοι μεταξύ αφίξεων είναι εκθετικά κατα-νεμημένοι (**Poisson** αφίξεις).

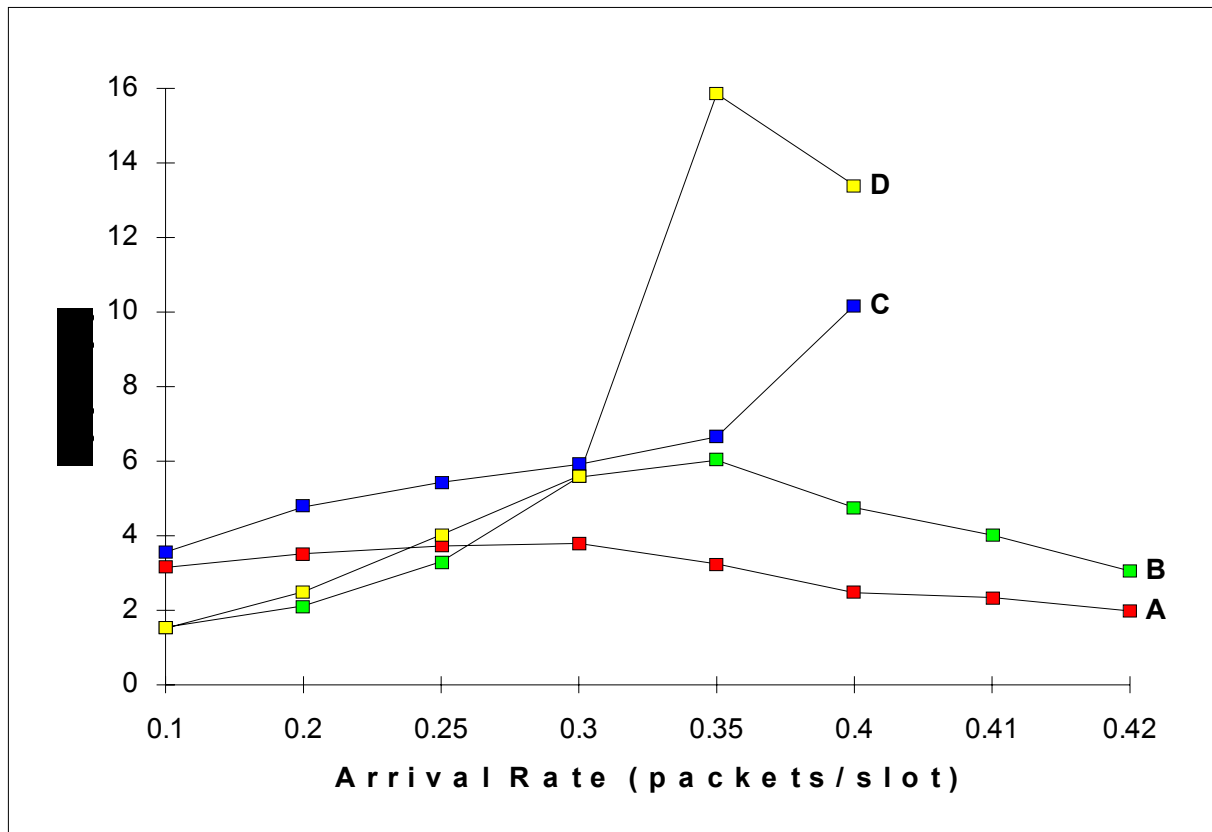
**5.3 Σύγκριση της καθυστέρησης των αλγορίθμων ιδανικού ALOHA και στοίβας με τριαδικό διαχωρισμό των συγκρουόμενων πακέτων για χρόνους μεταξύ αφίξεων πακέτων κατανεμημένους σύμφωνα με την κατανομή Pareto.**

Στο **Σχ.20** απεικονίζεται η μέση τιμή της καθυστέρησης πακέτων για τις κατανομές χρόνου μεταξύ αφίξεων **Pareto 1** και **Pareto 2** για τους αλγορίθμους ιδανικού **ALOHA** και στοίβας με τριαδικό διαχωρισμό των συγκρουόμενων πακέ-



**Σχ.20 :** Η γραφική παράσταση απεικονίζει την μέση τιμή της καθυστέρησης πακέτων σαν συνάρτηση του ρυθμού αφίξεων για τον αλγόριθμο του ιδανικού *ALOHA* και τον αλγόριθμο στοίβας με τριαδικό διαχωρισμό των συγκρουόμενων πακέτων, για τις κατανομές χρόνου μεταξύ αφίξεων διαδοχικών πακέτων **Pareto 1** και **Pareto 2**, όπου οι καμπύλες A και B απεικονίζουν την μέση τιμή της καθυστέρησης για τον ιδανικό αλγόριθμο *ALOHA* και για χρόνους μεταξύ αφίξεων κατανεμημένους σύμφωνα με τις **Pareto 1** και **Pareto 2** αντίστοιχα και οι καμπύλες C και D απεικονίζουν την μέση τιμή της καθυστέρησης για τον αλγόριθμο στοίβας με τριαδικό διαχωρισμό και για χρόνους μεταξύ αφίξεων κατανεμημένους σύμφωνα με τις **Pareto 1** και **Pareto 2** αντίστοιχα.

των, για διάφορους ρυθμούς αφίξεων πακέτων. Ως **Pareto 1** ορίζεται ξανά εκείνη η κατανομή **Pareto** με παράμετρο τοποθεσίας  $k=0.85$  slots, ενώ ως **Pareto 2** εκείνη η κατανομή **Pareto** με παράμετρο τοποθεσίας  $k=0.95$  slots. Από το **Σχ.20** παρατηρούμε ότι οι δύο αλγόριθμοι παρουσιάζουν παραπλήσιες μέσες καθυστερήσεις για ρυθμό άφιξης μικρότερο του 0.35 και για τις δύο τιμές της παραμέτρου τοποθεσίας της κατανομής **Pareto** που εξετάστηκαν. Για ρυθμούς αφίξεων μεγαλύτερους του 0.35 ο αλγόριθμος στοίβας παρουσιάζει μεγαλύτερες



**Σχ.21** : Η γραφική παράσταση απεικονίζει το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης πακέτων σαν συνάρτηση του ρυθμού αφίξεων για τον αλγόριθμο του ιδανικού *ALOHA* και τον αλγόριθμο στοίβας με τριαδικό διαχωρισμό των συγκρουόμενων πακέτων για τις κατανομές χρόνου μεταξύ αφίξεων διαδοχικών πακέτων **Pareto 1** και **Pareto 2**, όπου οι καμπύλες **A** και **B** απεικονίζουν το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης πακέτων για τον ιδανικό αλγόριθμο *ALOHA* και για χρόνους μεταξύ αφίξεων κατανεμημένους σύμφωνα με τις **Pareto 1** και **Pareto 2** αντίστοιχα και οι καμπύλες **C** και **D** απεικονίζουν το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης πακέτων για τον αλγόριθμο στοίβας με τριαδικό διαχωρισμό και για χρόνους μεταξύ αφίξεων κατανεμημένους σύμφωνα με τις **Pareto 1** και **Pareto 2** αντίστοιχα.

μέσες καθυστερήσεις απ' αυτές του ιδανικού αλγορίθμου *ALOHA*, μια και επιτυγχάνει ελαφρά μικρότερα *throughputs*. Επισημαίνεται ότι το τελευταίο σημείο της καμπύλης μέσης καθυστέρησης **C** (**D**) που εμφανίζεται στο **Σχ.20** αντιστοιχεί σε ρυθμό άφιξης 0.35 (0.4) μια και το αντίστοιχο μέγιστο *throughput* είναι 0.389 (0.41315).

Στο **Σχ.21** παρουσιάζεται το πηλίκο της τυπικής απόκλισης  $\sigma(D)$  διά της μέσης τιμής  $E(D)$  της καθυστέρησης των πακέτων για τις δύο προαναφερθείσες περιπτώσεις, για

διάφορους ρυθμούς αφίξεων πακέτων στο σύστημα. Από το **Σχ.21** φαίνεται ότι, για **Pareto** κατανομημένους χρόνους μεταξύ αφίξεων διαδο-χικών πακέτων, το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης για την περίπτωση του ιδανικού αλγορίθμου **ALOHA** μεταβάλλε-ται ελαφρά με τον ρυθμό άφιξης πακέτων (από 2 έως 3 για την **Pareto 1** και από 2 έως 5 για την **Pareto 2**). Αντίθετα η μεταβολή του παραπάνω πηλίκου είναι πιο έντονη όταν εξετάζουμε τον αλγόριθμο στοίβας με τριαδικό διαχωρισμό (από 4 έως 10 για την **Pareto 1** και από 2 έως 16 για την **Pareto 2**). Επιπλέον, για ρυθμούς άφιξης πακέτων μεγαλύτερους του 0.25 ο αλγόριθμος στοίβας παρου-σιάζει πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης μεγαλύτερο του αντίστοιχου πηλίκου του ιδανικού αλγορίθμου **ALOHA**.

Συνεπώς, η προβλεψιμότητα της καθυστέρησης πακέτων με βάση τη γνώση της μέσης τιμής της είναι καλύτερη στον ιδανικό αλγόριθμο **ALOHA** από την αντίστοιχη στον αλγόριθμο στοίβας με τριαδικό διαχωρισμό για ρυθμούς άφιξης πακέτων μεγαλύτερους του 0.25, ενώ για μικρότερους ρυθμούς άφιξης πακέτων οι δύο αλγόριθμοι εμφανίζονται περίπου ισοδύναμοι όσον αφορά την παραπάνω προβλεψιμότητα και τις απόλυτες μέσες τιμές της καθυστέρησης πακέτου (**Σχ.20**).

## ΚΕΦΑΛΑΙΟ 6

### Συμπεράσματα και Μελλοντικές Επεκτάσεις

Σε αυτή την εργασία εξετάσαμε το πρόβλημα της πολλαπλής τυχαίας προσπέλασης ενός επικοινωνιακού καναλιού, το οποίο χρησιμοποιείται από ένα μεγάλο αριθμό χρηστών για την μετάδοση των πακέτων τους, κάτω από την υπόθεση ότι η συνολική ροή γέννησης πακέτων στο κανάλι χαρακτηρίζεται από τυχαίους *Pareto* κατανεμημένους χρόνους μεταξύ αφίξεων διαδοχικών πακέτων. Η κατανομή *Pareto* επελέγη για το λόγο ότι από πειραματικές μετρήσεις κίνησης πακέτων σε δίκτυα ευρείας γεωγραφικής περιοχής έχει βρεθεί να μοντελοποιεί ικανοποιητικά τους χρόνους μεταξύ αφίξεων διαδοχικών πακέτων διαφόρων εφαρμογών δεδομένων.

Η παραπάνω κατανομή χαρακτηρίζεται από έντονη εκρηκτικότητα (για ορισμένες τιμές της παραμέτρου σχήματος  $a$ , η κατανομή έχει άπειρη διασπορά και πεπερασμένη μέση τιμή, ενώ για άλλες τιμές της ίδιας παραμέτρου έχει άπειρη διασπορά και άπειρη μέση τιμή), και από μνήμη (όσο πιο μεγάλος χρόνος έχει περάσει από την τελευταία άφιξη πακέτου, τόσο μεγαλύτερος είναι ο αναμενόμενος επιπλέον χρόνος μέχρι την επόμενη άφιξη πακέτου).

Εξετάσαμε δύο αλγορίθμους πολλαπλής τυχαίας προσπέλασης, ο καθένας από τους οποίους είναι αντιπροσωπευτικός μιας κλάσης τέτοιων αλγορίθμων από τις τρεις γενικές κλάσεις που έχουν επινοηθεί, σχεδιαστεί και μελετηθεί για *Poisson* αφίξεις πακέτων (για περισσότερες λεπτομέρειες σχετικά με τις τρεις κλάσεις αλγορίθμων τυχαίας πολλαπλής προσπέλασης ο αναγνώστης παραπέμπεται στο Κεφάλαιο 1).

Οι αλγόριθμοι που εξετάσαμε παρέχουν ελεύθερη και άμεση πρόσβαση στο κανάλι, στους χρήστες με “φρέσκα” πακέτα, και χρησιμοποιούν μηχανισμούς επαναμετάδοσης, οι οποίοι βασίζονται σε παραμέτρους που αν επιλεγούν κατάλληλα οδηγούν σε βέλτιστη απόδοση λειτουργίας των αλγορίθμων.



Η μελέτη αυτών των αλγορίθμων, μέσω μιας εκτεταμένης προσομοίωσης της λειτουργίας τους, έδειξε ότι είναι *ευσταθείς* και μάλιστα ανάλογα με την παράμετρο τοποθεσίας  $k$ , της κατανομής *Pareto* των χρόνων μεταξύ αφίξεων διαδοχικών πακέτων, πετυχαίνουν μεγαλύτερα *throughputs* από αυτά που επιτυγχάνονται με *Poisson* αφίξεις πακέτων. Επιπλέον, η συμπεριφορά της κατανομής καθυστέρησης πακέτου των παραπάνω αλγορίθμων είναι ικανοποιητική και έχει παρόμοια χαρακτηριστικά με την αντίστοιχη κατανομή για *Poisson* αφίξεις πακέτων. Εν γένει, η κατανομή καθυστέρησης πακέτων χαρακτηρίζεται από μικρές μέσες τιμές και από τυπικές αποκλίσεις πολλαπλάσιες των αντίστοιχων μέσων τιμών (πράγμα που ισχύει και για τις αντίστοιχες μετρικές καθυστέρησης κάτω από την υπόθεση *Poisson* αφίξεων πακέτων). Μόνο όταν ο συνολικός ρυθμός άφιξης “φρέσκων” πακέτων στο σύστημα (φορτίο καναλιού) πλησιάζει το μέγιστο *throughput* που επιτυγχάνει ο συγκεκριμένος αλγόριθμος πολλαπλής προσπέλασης, έχουμε μεγάλη αύξηση της μέσης καθυστέρησης και υψηλό κλάσμα τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης πακέτων.

Από τους δύο αλγορίθμους που μελετήσαμε ο πρώτος είναι ένας ιδανικός, ευσταθής αλγόριθμος τύπου *ALOHA*. Η υλοποίηση του στην πράξη είναι αδύνατη (στην ιδανική του μορφή), μια και για να επιτύχει ευσταθή λειτουργία υποθέτει την χρήση δύο εκτιμητριών (του συνολικού ρυθμού αφίξεων πακέτων στο κανάλι και του αριθμού των πακέτων που έχουν υποστεί τουλάχιστον μία σύγκρουση και βρίσκονται ακόμη στο σύστημα προσπαθώντας να μεταδώσουν επιτυχώς), οι οποίες πρέπει να είναι ακριβείς σε όλες τις χρονικές στιγμές (slots). Ο λόγος που εξετάσαμε τον παραπάνω αλγόριθμο είναι ότι εξαιτίας της υπόθεσης ιδανικών εκτιμητριών παρέχει το μεγαλύτερο *throughput* και τις μικρό-τερες καθυστερήσεις μεταξύ όλων των αλγορίθμων τύπου *ALOHA*.

Ο δεύτερος αλγόριθμος που εξετάσαμε είναι ο από τη φύση του ευσταθής αλγόριθμος στοίβας, ο οποίος δεν κάνει χρήση ιδανικών εκτιμητριών και συνεπώς είναι εύκολα υλοποιήσιμος στην πράξη.

Οι προσομοιώσεις μας έδειξαν ότι και οι δύο αλγόριθμοι έχουν ικανοποιητική απόδοση με τον ιδανικό, ευσταθή αλγόριθμο *ALOHA* να υπερέρχει ελαφρά του αλγόριθμου στοίβας με τριαδικό χωρισμό των συγκρουόμενων πακέτων. Το γεγονός ότι ο πρώτος από τους παραπάνω αλγορίθμους είναι μη υλοποιήσιμος στην πράξη στην ιδανική μορφή που τον μελετήσαμε και ο τελευταίος είναι εύκολα υλοποιήσιμος στην πράξη, μας οδηγεί στο συμπέρασμα ότι ο αλγόριθμος στοίβας αποτελεί μια καλή επιλογή για τον έλεγχο των

μεταδόσεων / επαναμετα-δόσεων πακέτων που χαρακτηρίζονται από **Pareto** κατανομημένους χρόνους μεταξύ αφίξεων σε κοινό κανάλι πολλαπλής προσπέλασης.

Στο μέλλον αξίζει να μελετηθεί η συμπεριφορά των αλγορίθμων πολλα-πλής τυχαίας προσπέλασης που ανήκουν στην κλάση της δεσμευμένης και όχι της άμεσης προσπέλασης των “φρέσκων” πακέτων στο κανάλι. Η παραπάνω κλάση αλγορίθμων περιλαμβάνει ευσταθείς από τη φύση τους αλγορίθμους που επιτυγχάνουν το καλύτερο **throughput** και τις μικρότερες καθυστερήσεις για **Poisson** αφίξεις πακέτων [4]. Οι παραπάνω αλγόριθμοι είναι υλοποιήσιμοι στην πράξη, αλλά σαφώς πιο πολύπλοκοι (για τις λεπτομέρειες υλοποίησης ο αναγνώστης παραπέμπεται στην σχετική συζήτηση στο Κεφάλαιο 1) από αυτούς που ανήκουν στην κλάση των αλγορίθμων πολλαπλής τυχαίας προσπέλασης στοίβας.

# ΠΑΡΑΡΤΗΜΑ 1

## Ο Κώδικας προσομοίωσης του Αλγορίθμου ALOHA

Εδώ παραθέτουμε τον κώδικα που χρησιμοποιήθηκε για την προσομοίωση του αλγορίθμου ALOHA (οι συναρτήσεις *pareto()* και *generator()* που χρησιμοποιούνται παρατίθενται στο Παράρτημα 3, καθώς και η συνάρτηση *poisson()* η οποία χρησιμοποιήθηκε κατά την διάρκεια της προσομοίωσης στην θέση της κατανομής *Pareto* για την παραγωγή των αποτελεσμάτων για αφίξεις πακέτων σύμφωνα με την Εκθετική κατανομή).

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include "gen2.h"

#define y 1000000
#define arr 5000

void main(){

int i,q,backlog_size,h,j,numb_new_packet,e,empty_flag,aker_counter;
float t,counter,v,l,p,n[arr],pareto(),generator(),d[arr],delay[11],v2,sample_var,
    temp_count,temp_delay,total_delay,delay2,s;
long slot_number,num_trans_packet,total_backlog,empty;
double tt;

slot_number=1; /* Το χρονικό τμήμα που εξετάζεται */
backlog_size=0; /* Ο αριθμός των πακέτων που υπάρχουν κάθε στιγμή στο backlog */
h=0;
num_trans_packet=0; /* Ο αριθμός των πακέτων που έχουν μεταδοθεί επιτυχώς */
num_new_packet=0; /* Ο αριθμός των "φρέσκων" πακέτων που εισάγονται στο σύστημα */
p=0;
l=0;
e=0;
counter=0; /* Ο μετρητής που ελέγχει τα πακέτα που εισάγονται στη διάρκεια ενός slot */
t=0.0;
total_delay=0.0; /* Η συνολική καθυστέρηση όλων των πακέτων που μεταδόθηκαν επιτυχώς */
v=0.0;
empty=0; /* Τα αδρανή slots */
total_backlog=0; /* Ο συνολικός αριθμός των πακέτων που περνούν από το backlog */
empty_flag=0;
aker_counter=0;
delay2=0.0;
v2=0.0;
sample_var=0.0;
s=0.0;
tt=2.0;
```

```

for (i=0; i<arr; ++i)
{
    d[i]=0.0;      /* Η καθυστέρηση κάθε πακέτου που εισάγεται στο σύστημα */
    n[i]=1.0;     /* Ο τυχαίος αριθμός που καθορίζει την αναμετάδοση ή όχι ενός πακέτου
                  από το backlog */
}
for (i=0; i<11; I++)
{
    delay[i]=0.0; /* Η καθυστέρηση του πρώτου πακέτου που εισάγεται σε ένα slot)
}

l=(a-1) / (a*k); /* Ο ρυθμός άφιξης πακέτων */

while (num_trans_packet<y) /*Έλεγχος για τον αριθμό των πακέτων που μεταδόθηκαν επιτυχώς */
{
    if ((counter<=1)&&(slot_number!=1))
    {
        delay[1]=1-counter;
        i=2;
    }
    else
        i=1;
    while (counter<=1)
    {
        t=pareto(); /* Ο χρόνος μεταξύ αφίξεων πακέτων */
        counter+=t;
        if (counter<=1)
        {
            delay[i]=1-counter;
            ++i;
        }
        ++numb_new_packet;
    }
    --counter;

    if ((numb_new_packet>0)&&(slot_number==1))
        --numb_new_packet;

/* Εξέταση της περίπτωσης που δεν εισάγεται στο σύστημα “φρέσκο” πακέτο και στο backlog υπάρχουν πακέτα
που περιμένουν να επαναμεταδοθούν */

    if ((numb_new_packet==0)&&(backlog_size!=0))
    {
        for (i=1; i<=backlog_size; ++i)
            if (n[i]<=p) /* Εξετάζουμε πόσα πακέτα από το backlog έχουν τη δυνατότητα να
                        επαναμεταδοθούν */
            {
                ++e;
                h=i;
            }
        if (e==1) /* Αν ένα μόνο από τα πακέτα έχει δικαίωμα να μεταδώσει, μεταδίδει
                  επιτυχώς */
        {
            ++num_trans_packet;
            total_delay+=d[h]+1;
            delay2=pow((d[h]+1),tt);
            v2+=delay2;
            d[h]=d[backlog_size];
            n[h]=n[backlog_size];
            d[backlog_size]=0;

```

```

n[backlog_size]=1;
--backlog_size;
empty_flag=1;
}
else if (e>1) /* Αν περισσότερα πακέτα έχουν δικαίωμα να μεταδώσουν, συγκρούονται
και δεν μεταδίδει κανένα επιτυχώς */
empty_flag=1;
else if (e==0) /* Αν κανένα πακέτο δεν έχει δικαίωμα να μεταδώσει */
empty_flag=0; /* Το ακόλουθο slot είναι αδρανές */
}

/* Εξέταση της περίπτωσης, όπου εισάγεται στο σύστημα ένα “φρέσκο” πακέτο και στο backlog δεν υπάρχει
κανένα πακέτο */

else if ((numb_new_packet==1)&&(backlog_size==0))
{
++num_trans_packet; /* Το πακέτο μεταδίδεται επιτυχώς */
total_delay+=delay[1]+1;
delay2=pow((delay[1]+1),tt);
v2+=delay2;
empty_flag=1;
}

/* Εξέταση της περίπτωσης, όπου εισάγεται στο σύστημα ένα “φρέσκο” πακέτο και στο backlog υπάρχουν
πακέτα για επαναμετάδοση */

else if ((numb_new_packet==1)&&(backlog_size!=0))
{
for (i=1; i<=backlog_size; ++i)
{
if (n[i]<=p) /* Εξετάζουμε πόσα πακέτα από το backlog έχουν τη δυνατότητα
να επαναμεταδοθούν */
++e;
}
if (e==0) /* Εάν κανένα πακέτο από το backlog δεν επαναμεταδίδει */
{
++num_trans_packet; /* Το “φρέσκο” πακέτο μεταδίδεται επιτυχώς */
total_delay+=delay[1]+1;
delay2=pow((delay[1]+1),tt);
v2+=delay2;
empty_flag=1;
}
else /* Διαφορετικά προκαλείται σύγκρουση */
{
++backlog_size;
++total_backlog;
empty_flag=1;
}
}

/* Εξέταση της περίπτωσης, όπου εισάγονται στο σύστημα περισσότερα από ένα “φρέσκα” πακέτα */

else if (numb_new_packet>1)
{
backlog_size+=numb_new_packet; /* Προκαλείται σύγκρουση και εισάγονται στο
backlog */
total_backlog+=numb_new_packet;
empty_flag=1;
}

```

```

if (backlog!=0)
{
p=(1-l)/backlog_size; /* Καθορισμός της πιθανότητας με την οποία θα επαναμεταδοθούν
τα πακέτα από το backlog */
q=1;
for (i=backlog_size- numb_new_packet+1; i<=backlog_size; ++i)
{
d[i]=delay[q];
++q;
}

for (i=1; i<=backlog_size; ++i)
{
n[i]=generator();
+d[i];
}
}

/* Εξέταση τις περίπτωσης όπου πολλά συνεχόμενα slots είναι κενά */

if (numb_new_packet==0)
{
if ((counter>=2)&&(backlog_size==0))
{
aker_counter=counter;
slot_number+=aker_counter;
counter-=aker_counter;
empty+=aker_counter;
}
else if (empty_flag==0)
{
if (counter>1)
++empty;
if (slot_number==1)
++empty;
}
}
++slot_number;
delay2=0.0;
empty_flag=0;
numb_new_packet=0;
e=0;
}

v=total_delay/num_trans_packet; /* Υπολογισμός της μέσης καθυστέρησης πακέτου */
sample_var=(v2-y*pow(v,tt))/(y-1);
s=sqrt((double)sample_var); /* Υπολογισμός της τυπικής απόκλισης της καθυστέρησης */

printf("Number of transmited packets : %ld \n",num_trans_packet);
printf("Number of the simulated slots : %ld \n",slot_number);
printf("Number of empty slots : %ld \n",empty);
printf("Number of the packets in the backlog : %d \n",backlog_size);
printf("Total number of packets that went through the backlog : %d \n",total_backlog);
printf("Arrival time : %f \n",l);
printf("E(t)= %f \n",1/l);
printf("The mean value of delay : %f \n",v);
printf("The deviation is : %f \n",s);
printf("The ratio  $\sigma(D)/E(D)$  : %f \n",s/v);
printf("END \n");

```

}

## ΠΑΡΑΡΤΗΜΑ 2

### Οι Κώδικες προσομοίωσης των Αλγορίθμων Στοιβάς

Εδώ παραθέτουμε τους κώδικες που χρησιμοποιήθηκαν για την προσομοίωση των αλγορίθμων στοιβάς (οι συναρτήσεις *pareto()* και *generator()* που χρησιμοποιούνται παρατίθενται στο Παράρτημα 3, καθώς και η συνάρτηση *poisson()* η οποία χρησιμοποιήθηκε κατά την διάρκεια της προσομοίωσης στην θέση της κατανομής **Pareto** για την παραγωγή των αποτελεσμάτων για αφίξεις πακέτων σύμφωνα με την **Εκθετική** κατανομή).

#### 2.1 Ο κώδικας του αλγορίθμου στοιβάς με δυαδικό στατιστικό διαχωρισμό των συγκρουόμενων πακέτων.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include "gen2.h"

#define y 1000000
#define arr 5000
#define Max_Len 500
#define p 0.5

typedef struct stack { /* Η στοιβά στην οποία εισάγονται τα πακέτα */
    int s[Max_Len];
    float top;
} stack;

int i,q,u,j,g,numb_new_packet,e,empty_flag,aker_counter,s0,s1,sum,
    c {Max_Len},top;
float t,counter,v,l,n[Max_Len],pareto(),generator(),delay[11],d[Max_Len],
    delay2,s,m_out;
double total_delay,v2,sample_var,tt;
long slot_number,num_trans_packet,empty;
stack stk;
```



```

void Initialization() {
slot_number=1; /* Αριθμός των χρονικών τμημάτων (slots) */
num_trans_packet=0; /* Αριθμός των πακέτων που μεταδόθηκαν επιτυχώς */
numb_new_packet=0; /* Αριθμός των “φρέσκων” πακέτων που εισάγονται σε ένα slot */
l=0; /* Ρυθμός άφιξης πακέτων */
counter=0.0; /* Το άθροισμα του χρόνου μεταξύ των πακέτων */
t=0.0; /* Χρόνοι άφιξης από την χρησιμοποιούμενη κατανομή */
total_delay=0.0; /* Ο συνολικός χρόνος καθυστέρησης των πακέτων που μεταδόθηκαν */
v=0.0; /* Η μέση τιμή της καθυστέρησης */
empty=0; /* Αριθμός των slots που είναι ανενεργά */
empty_flag=0; /* Έλεγχος για την κατάσταση του slot */
aker_count=0; /* Το ακέραιο μέρος του counter */
delay2=0.0; /* Το τετράγωνο της καθυστέρησης */
stk.top=-1.0; /* Ο δείκτης που δείχνει την κορυφή της στοίβας */
s0=0;
s1=0;
s=0.0;
v2=0.0;
sample_var=0.0;
tt=2.0;
g=0;

for (i=0; i<Max_Len; ++i)
{
d[i]=0.0;
n[i]=1.0;
}
for (i=0; i<11; ++i)
delay[i]=0.0;
for (i=0; i<=Max_Len; ++i)
stk.s[i]=0;
for (i=0; i<=Max_Len; ++i)
c[i]=Max_Len+1;

sum=0;
l=(a-1) / (a*k);
}

void() {
Initialization();
while (num_trans_packet <y)
{
if ((counter<=1)&&(slot_number!=1))
{
delay[1]=1-counter;
i=2;
}
else
i=1;
while (counter<=1)
{
t=pareto();
counter+=t;
if (counter<=1)
{
delay[i]=1-counter;
++i;
}
}
++numb_new_packet;
}
}

```

```

    }
    --counter;
    if ((numb_new_packet>0)&&(slot_number==1))
        --numb_new_packet;

/* Εξέταση της περίπτωσης όπου το πρώτο κελί της στοίβας περιέχει περισσότερα από ένα πακέτα */

if (stk.s[0]>1)
{
    stk.top++; /* Αυξάνεται το βάθος της στοίβας */
    if (stk.top>1)
    {
        for (i=stk.top; i>1; i--)
            stk.s[i]=stk.s[i-1];
        for (i=1; i<=sum;++i)
            if (c[i]!=0)
                c[i]++;
    }
    e=0;
    for (i=1; i<=stk.s[0];++i)
    {
        u=0;
        for (j=1+e;j<=sum;++j)
        {
            if (c[j]==0)&&(u==0))
            {
                e=j;
                u++;
            }
        }
        n[i]=generator();
        if (n[i]<=p) /* Εξέταση της πιθανότητας του πακέτου */
            s0++; /* να μείνει στο πρώτο κελί */
        else
        {
            s1++; /* ή να πάει στο δεύτερο κελί της στοίβας */
            c[e]++;
        }
    }
    stk.s[0]=s0;
    stk.s[1]=s1;
}

/* Εξέταση της περίπτωσης όπου μόνο ένα πακέτο βρίσκεται στο πρώτο κελί της στοίβας */

else if (stk.s[0]==1)
{
    ++num_trans_packet; /* Μεταδίδεται επιτυχώς */
    for (i=1; i<=sum; ++i)
    {
        if (c[i]==0)
            g=i;
    }
    total_delay+=d[g];
    delay2=pow(d[g],tt);
    v2+=delay2;
    if (g!=sum)
    {
        d[g]=d[sum];
        c[g]=c[sum];
    }
}

```

```

    d[sum]=0;
    c[sum]=Max_Len;
}
else
{
    d[g]=0;
    c[g]=Max_Len;
}
empty_flag=1;
if (stk.top>=1)
{
    for (I=1; i<=stk.top;++i)
        stk.s[i-1]=stk.s[i];
    top=stk.top;
    stk.s[top]=0;
    for (i=1; i<=sum; ++i)
        if (c[i]!=0)
            c[i]--;
}
else
    stk.s[0]=0;
if (stk.top>-1)
    stk.top--;
}

```

/\* Εξέταση της περίπτωσης όπου κανένα πακέτο δεν βρίσκεται στο πρώτο κελί της στοίβας \*/

```

else if (stk.s[0]==0)
{
    if (stk.top>=1)
    {
        for (i=1; i<=stk.top; ++i)
            stk.s[i-1]=stk.s[i];
        for (i=1; i<=sum; ++i)
            if (c[i]!=0)
                c[i]--;
        top=stk.top;
        stk.s[top]=0;
    }
    if (stk.top>-1)
        stk.top--;
}

```

```

if ((stk.top==-1)&&(numb_new_packet!=0))
    stk.top++;
if ((d[sum]=0)&&(sum!=0))
    sum--;
stk.s[0]+=numb_new_packet;
for (i=sum+1; i<=sum+numb_new_packet; ++i)
    c[i]=0;
sum=0;

```

```

if (stk.top>=0)
{
    for (i=0; i<=stk.top; i++)
        sum+=stk.s[i];
    q=1;
    for (i=sum-numb_new_packet+1; i<=sum; ++i)
    {
        d[i]=delay[q];
    }
}

```

```

    ++q;
    }
    for (i=1; i<=sum; ++i)
        ++d[i];
    }

/* Εξέταση τις περίπτωσης όπου πολλά συνεχόμενα slots είναι κενά */

if (numb_new_packet==0)
{
    if ((counter>=2)&&(stk.top==-1))
    {
        aker_counter=counter;
        slot_number+=aker_counter;
        counter-=aker_counter;
        empty+=aker_counter;
    }
    else if (empty_flag==0)
    {
        if (counter>1)
            ++empty;
        if (slot_number==1)
            ++empty;
    }
    }
    ++slot_number;
    delay2=0.0;
    empty_flag=0;
    numb_new_packet;
    s0=0;
    s1=0;
}

v=total_delay/num_trans_packet;
sample_var=(v2-num_trans_packet*pow(v,tt))/(num_trans_packet-1);
s=sqrt(sample_var);
m_out=num_trans_packet/(slot_number-1);

printf("\n");
printf("Number of packet transmited : %ld \n",num_trans_packet);
printf("Number of the simulated slots : %ld \n",slot_number-1);
printf("Number of empty slots : %ld \n",empty);
printf("Output rate of the packets : %f \n",m_out);
printf("Number of packets left in the stack after the end of simulation : %d \n",sum);
printf("The top of the stack : %f \n",stk.top);
printf("Arrival time : %f \n",l);
printf("E(t)= %f \n",1/l);
printf("The mean value of delay : %f \n",v);
printf("The deviation is : %f \n",s);
printf("The ratio  $\sigma(D)/E(D)$  : %f \n",s/v);
printf("END \n");

```

## 2.2 Ο κώδικας του αλγορίθμου στοίβας με τριαδικό στατιστικό διαχωρισμό των συγκρουόμενων πακέτων.

```
#include <stdio.h>
```

```

#include <stdlib.h>
#include <time.h>
#include <math.h>
#include "gen.h"

#define y 1000000
#define arr 10000
#define Max_Len 10000

#define p1 0.33
#define p2 0.66

typedef struct stack {
    int s[Max_Len];
    float top;
} stack;

int i,q,u,j,g,numb_new_packet,e,empty_flag,aker_counter,s0,s1,s2,sum,c{Max_Len},top;
float t,counter,v,l,n[Max_Len],pareto(),generator(),delay[11],d[Max_Len],delay2,s,m_out;
double total_delay,v2,sample_var,tt;
long slot_number,num_trans_packet,empty;
stack stk;

void Initialization() {
    slot_number=1; /* Αριθμός των χρονικών τμημάτων (slots) */
    num_trans_packet=0; /* Αριθμός των μεταδιδόμενων πακέτων */
    numb_new_packet=0; /* Αριθμός των πακέτων που φτάνουν σε ένα slot */
    l=0; /* Ρυθμός άφιξης πακέτων */
    counter=0.0; /* Το άθροισμα του χρόνου μεταξύ των πακέτων */
    t=0.0; /* Χρόνοι άφιξης από την χρησιμοποιούμενη κατανομή */
    total_delay=0.0; /* Ο συνολικός χρόνος καθυστέρησης των πακέτων που μεταδόθηκαν */
    v=0.0; /* Η μέση τιμή της καθυστέρησης */
    empty=0; /* Αριθμός των slots που είναι ανενεργά */
    empty_flag=0; /* Έλεγχος για την κατάσταση του slot */
    aker_count=0; /* Το ακέραιο μέρος του counter */
    delay2=0.0; /* Το τετράγωνο της καθυστέρησης */
    stk.top=-1.0; /* Ο δείκτης που δείχνει την κορυφή της στοίβας */
    s0=0;
    s1=0;
    s3=0;
    s=0.0;
    v2=0.0;
    sample_var=0.0;
    tt=2.0;
    g=0;

    for (i=0; i<Max_Len; ++i)
    {
        d[i]=0.0;
        n[i]=1.0;
    }
    for (i=0; i<11; ++i)
        delay[i]=0.0;
    for (i=0; i<=Max_Len; ++i)
        stk.s[i]=0;
    for (i=0; i<=Max_Len; ++i)
        c[i]=Max_Len+1;
    sum=0;
    l=(a-1) / (a*k);
}

```

```

void main() {
Initialization();

while (num_trans_packet<y)
{
if ((counter<=1)&&(slot_number!=1))
{
delay[1]=1-counter;
i=2;
}
else
i=1;
while (counter<=1)
{
t=pareto();
counter+=t;
if (counter<=1)
{
delay[i]=1-counter;
++i;
}
++numb_new_packet;
}
--counter;

if ((numb_new_packet>0)&&(slot_number==1))
--numb_new_packet;

/* Εξέταση της περίπτωσης όπου το πρώτο κελί της στοίβας περιέχει περισσότερα από ένα πακέτα */

if (stk.s[0]>1)
{
stk.top+=2;          /* Αυξάνεται το βάθος της στοίβας κατά δύο */
if (stk.top>2)
{
for (i=stk.top; i>1; i--)
stk.s[i]=stk.s[i-2];
for (i=1; i<=sum; ++i)
if (c[i]!=0)
c[i]+=2;
}
e=0;
for (i=1; i<=stk.s[0]; ++i)
{
u=0;
for (j=1+e; j<=sum; ++j)
{
if ((c[j]==0)&&(u==0))
{
e=j;
u++;
}
}
}
n[i]=generator();
if (n[i]<=p1)          /* Εξέτασης της πιθανότητας */
s0++;                /* να μείνει το πακέτο στο πρώτο κελί */
else if ((n[i]>p1)&&(n[i]<=p2))
{

```

```

        s1++;          /* να πάει στο δεύτερο κελί */
        c[e]++;
    }
    else if (n[i]>p2)
    {
        s2++;          /* ή στο τρίτο κελί της στοίβας */
        c[e]+=2;
    }
}
stk.s[0]=s0;
stk.s[1]=s1;
stk.s[2]=s2;
}

```

/\* Εξέταση της περίπτωσης όπου μόνο ένα πακέτο βρίσκεται στο πρώτο κελί της στοίβας \*/

```

else if (stk.s[0]==1)
{
    ++num_trans_packet;
    for (i=1; i<=sum; ++i)
        if (c[i]==0)
            g=i;
    total_delay+=d[g];
    delay2=pow(d[g],tt);
    v2+=delay2;
    if (g!=sum)
    {
        d[g]=d[sum];
        c[g]=c[sum];
        d[sum]=0;
        c[sum]=Max_Len;
    }
    else
    {
        d[g]=0;
        c[g]=Max_Len;
    }
    empty_flag=1;
    if (stk.top>=1)
    {
        for (i=1; i<=stk.top; ++i)
            stk.s[i-1]=stk.s[i];
        top=stk.top;
        stk.s[top]=0;
        for (i=1; i<=sum; ++i)
            if (c[i]!=0)
                c[i]-;
    }
    else
        stk.s[0]=0;
    if (stk.top>-1)
        stk.top--;
}

```

/\* Εξέταση της περίπτωσης όπου κανένα πακέτο δεν βρίσκεται στο πρώτο κελί της στοίβας \*/

```

else if (stk.s[0]==0)
{
    if (stk.top>=1)
    {

```

```

        for (i=1; i<=stk.top; ++i)
            stk.s[i-1]=stk.s[i];
        for (i=1; i<=sum; ++i)
            if (c[i]!=0)
                c[i]--;
        top=stk.top;
        stk.s[top]=0;
    }
    if (stk.top>-1)
        stk.top--;
}

if ((stk.top==-1)&&(numb_new_packet!=0))
    stk.top++;
if ((d[sum]==0)&&(sum!=0))
    sum--;
stk.s[0]+=numb_new_packet;
for (i=sum+1; i<=sum+numb_new_packet; ++i)
    c[i]=0;
sum=0;
if (stk.top>=0)
{
    for (i=0; i<=stk.top; ++i)
        sum+=stk.s[i];
    q=1;
    for (i=sum-numb_new_packet+1; i<=sum; ++i)
    {
        d[i]=delay[q];
        ++q;
    }
    for (i=1; i<=sum; ++i)
        ++d[i];
}

/* Εξέταση τις περίπτωσης όπου πολλά συνεχόμενα slots είναι κενά */

if (numb_new_packet==0)
{
    if ((counter>=2)&&(stk.top==-1))
    {
        aker_counter=counter;
        slot_number+=aker_counter;
        counter-=aker_counter;
        empty+=aker_counter;
    }
    else if (empty_flag==0)
    {
        if (counter>1)
            ++empty;
        if (slot_number==1)
            ++empty;
    }
}
++slot_number;
delay2=0.0;
empty_flag=0;
numb_new_packet;
s0=0;
s1=0;
s2=0;

```



```

}

v=total_delay/num_trans_packet;
sample_var=(v2-num_trans_packet*pow(v,tt))/num_trans_packet;
s=sqrt(sample_var);
m_out=num_trans_packet/(slot_number-1);

printf("\n");
printf("Number of packet transmited : %ld \n",num_trans_packet);
printf("Number of the simulated slots : %ld \n",slot_number-1);
printf("Number of empty slots : %ld \n",empty);
printf("Output rate of the packets : %f \n",m_out);
printf("Number of packets left in the stack after the end of simulation : %d \n",sum);
printf("The top of the stack : %f \n",stk.top);
printf("Arrival time : %f \n",l);
printf("E(t)= %f \n",1/l);
printf("The mean value of delay : %f \n",v);
printf("The deviation is : %f \n",s);
printf("The ratio  $\sigma(D)/E(D)$  : %f \n",s/v);
printf("END \n");

```

## ΠΑΡΑΡΤΗΜΑ 3

### Οι Κώδικες των Γεννητριών Τυχαίων Αριθμών

*Στην συνέχεια ακολουθούν οι συναρτήσεις που χρησιμοποιήθηκαν στα προγράμματα των Παραρτημάτων 1 και 2.*

Η γεννήτρια τυχαίων αριθμών κατανεμημένων σύμφωνα με την κατανομή *Pareto* :

```

#define m 2147483647
#define a 1.6
#define k 0.95

float pareto()
{
    int g=0,i=0;
    float t=0.0,p=0.0,dyn=0.0;

    g=rand();
    p=(float)g/m;
    dyn=pow(p,1/a);
    if (dyn!=0.0)
        t=(float)k/dyn;
    else
    {
        while (dyn==0)
        {
            g=rand();
            p=(float)g/m;
            dyn=pow(p,1/a);
            if (dyn!=0)
                t=(float)k/dyn;
        }
    }
    return t;
}

```

Η γεννήτρια τυχαίων αριθμών κατανομημένων σύμφωνα με την κατανομή *Poisson* :

```

#define l 0.3

float poisson()
{
    int g=0;
    float p=0.0,t=0.0;

    g=rand();
    p=(float)g/m;
    while (p==0)
    {
        g=rand();
        p=(float)g/m;
    }
    t=-(1/l)*log(p);
    return t;
}

```

Η γεννήτρια τυχαίων αριθμών που ανήκουν στο διάστημα (0,1] :

```

float generator()
{
    int g=0;
    float f=0.0;

    g=rand();

```

```
f=(float)g/m;  
return f;  
}
```

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1]** Michael Paterakis, " Random Access Algorithms for Multiuser Computer Communication Networks " , Ph.D. Dissertation, University of Virginia, 1988.
- [2]** P. Mathys and P.Flajolet, " Q-ary Collision Resolution Algorithms in Random-Access Systems with Free or Blocked Channel Access " , IEEE Transactions on Information Theory, Vol. IT-31, No. 2, March 1985, pp. 217-243.
- [3]** C.Bisdikian, " A Review of Random Access Algorithms " , IBM Research Report, RC 20348, T.J Watson Research Center Yorktown Heights, New York, 1995.
- [4]** D. Bertsekas and R. Gallager, Data Networks. Prentice Hall, 1992, 2nd Edition.
- [5]** N.Abramson, " The ALOHA System - Another Alternative for Computer Communications " , 1970 Fall Joint Comp. Conf., AFIPS Conf. Proc., Vol. 37, pp. 281-285.

- [6] G.Fayolle and E.Gelenbe, " Stability and Optimal Control of the Packet Switching Broadcast Channel ", Journal of the Association for Computing Machinery, Vol. 24, No. 3, July 1977, pp. 375-386.
- [7] J.I Capetanakis, " Generalized TDMA: The Multiple Accessing Tree Protocol ", IEEE Trans. on Communications, Oct.1979, COM-27, pp. 1476-1484.
- [8] B.S. Tsybakov and V.A Mikhailov, " Free Synchronous Packet Access in a Broadcast Channel With Feedback ", Problemy Peredachi Informatsii, Oct.-Dec. 1978, Vol. 14, No. 4, pp. 34-59.
- [9] R.G. Gallager, " A Perspective of Multiaccess Channels ", IEEE Trans. on Inform. Theory, Special Issue on Random Access Communications, March 1985, Vol. IT-31, No. 2, pp. 280-294.
- [10] B.S Tsybakov and V.A. Mikhailov, " Random Multiple Access of Packets, Part-and-Try Algorithms ", Problemy Peredachi Informatsii, Oct.-Dec. 1980, Vol. 16, No. 3, pp. 65-79.
- [11] J.L Massey, " Collision Resolution Algorithms and Random Access Communications, in Multi-User Communication Systems ", Ed. G. Longo, CISM Courses and Lectures, No. 265, pp. 73-137, Springer-Verlag, New York, 1981.
- [12] G. Anastasi, D. Grillo, L. Lenzini, and E. Mingozzi, " A Bandwidth Reservation Protocol for Speech/Data Integration in TDMA-Based Advanced Mobile Systems ", Proc. of the 1996 IEEE INFOCOM Conf., March 1996, San Fransisco, CA, pp. 722-729.
- [13] V.Paxson, and S.Floyd, " Wide Area Traffic : The failure of Poisson Modelling ", IEEE/ACM Trans. on Networking, June 1995, Vol. 3, No. 3, pp. 226-244.
- [14] M. Paterakis and A. Cleary, " On the Voice-Data Integration in Third Generation Wireless Access Communication Networks ", European Trans. on Telecommunications and Related Technologies, Vol. 5, No. 1, Jan-Feb. 1994, pp. 11-18.
- [15] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, " Packet Reservation Multiple Access for Local Wireless Communications ", IEEE Trans. on Communications, Vol. 37, No. 8, August 1989, pp. 885- 890.
- [16] P.T. Brady, " A Model for Generating ON-OFF Speech Patterns in Two-way Conversation ", Bell. Syst. Tech. J., Sept. 1969, pp. 2445-2472.
- [17] R.Gusella, " Measurment Study of Diskless Workstation Traffic on an Ethernet ", IEEE Trans. on Communications, Vol. COM-38, Sept. 1990, pp. 1557-1568.

- [18] V.Paxson, "Empirically Derived Analytic Models of Wide Area TCP Connections", IEEE/ACM Transactions on Networking, Vol. 2, No. 4, August 1994, pp. 316-336.
- [19] ITU-R Draft New Recommendation FPLMTS.REVAL, " Procedure for Evaluation of Radio Transmission Technologies for FPLMTS ", 8th TG 8/1 meeting, San Diego, February 7 - 17, 1995.
- [20] ETSI Draft Recommendation 050402, " Radio Selection Procedures ", Version 0.8.1, 12th SMG5 meeting, Rome, February 27 - March 3, 1995.
- [21] B.Arnold, Pareto Distributions, Baltimore, MD; International Cooperative Publishing House, 1983.
- [22] D.Duffy, A.McIntosh, M.Rosenstein, and W.Willinger, " Statistical Analysis of CCSN/SS7 traffic data from working CCS subnetworks ", IEEE J. Select. Areas Commun., Vol. 12, April 1994, pp. 544-551.
- [23] M.Carrett and W.Willinger, " Analysis, modeling and generation of self-similar VBR video traffic ", in Proc. SIGCOMM '94, Sept. 1994, pp. 269-280.
- [24] T.Hettmansperger, and M.Keenan, " Tailweight, statistical inference, and families of distributions - A brief survey ", in Statistical Distributions in Scientific Work, Vol. 1, G.P. Patil et al., Eds. Boston, MA: Kluwer, 1980, pp. 161-172.
- [25] S.Kauffman, The Origins of Order : Self-Organization and Selection in Evolution. London, England: Oxford University Press, 1993.
- [26] W.Leland and T.Ott, " Load-balancing heuristics and process behavior ", in PERFORM, '86 and ACM SIGM. 1986 Joint Conf. Comput. Perform., Model., Meas., Eval., North Carolina State Univ., May 1986, pp. 54-69.
- [27] W.Leland, M.Taqqu, W.Willinger, and D.Wilson, " On the self-similar nature of Ethernet traffic (extended version) ", IEEE/ACM Trans. Networking, Vol. 2, Feb. 1994, pp. 1-15.
- [28] B.Mandelbrot, " New methods in statistical economics ", J.Political Economy, Vol. 71, No. 5, Oct. 1963, pp. 421-440.
- [29] \_\_\_\_\_, The Fractal Geometry of Nature. New York: Freeman, 1983.
- [30] E.Cinlar, Introduction to Stochastic Processes. Prentice Hall, 1975.
- [31] T.Saadawi, M.Ammar, and A.Hakeem, Fundamentals of Telecommunication Networks. New York, J.Wiley Interscience, 1994.

**[32]** L.Georgiadis and M.Paterakis, " Bounds on the Delay Distribution of Window Random Access Algorithms ", IEEE Trans. on Communications, Vol. COM -41, No.5, May 1993, pp. 683-693.

**[33]** G.Polyzos and M.Molle, " A Queueing Theoretic Approach to the Delay Analysis for the FCFS 0.487 Conflict Resolution Algoritm ", IEEE Trans. on Information Theory, Vol. 39, No. 6, Nov. 1993, pp. 1887-1906.

**[34]** A.C.Cleary and M. Paterakis, " Voice-Data Integrated Channel Access for Microcellular Networks using Splitting RRA Algorithms ", Book chapter (No. 15) in High-Speed Networking for Multimedia Applications. W. Effelsberg, O. Spaniol, A. Danthine, D. Ferrari (editors), Kluwer Academic Publishers, Boston/ Dordrecht/ London, March 1996.

**[35]** A.C.Cleary and M.Paterakis, " An Investigation of Reservation Random Access Algorithms for Voice-Data Integration in Microcellular Wireless Enviroments ", International Journal of Wireless Information Networks, Vol. 2, No. 1, pp. 1-16, 1995.