

# **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΜΕΛΕΤΗ ΑΠΟΔΟΣΗΣ  
ΠΡΩΤΟΚΟΛΛΟΥ ΤΥΧΑΙΑΣ ΠΡΟΣΠΕΛΑΣΗΣ ΓΙΑ  
ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΥΠΗΡΕΣΙΩΝ  
ΦΩΝΗΣ-ΔΕΔΟΜΕΝΩΝ**

**Ιωάννης Κ. Ρομπογιαννάκης**

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΧΑΝΙΑ, ΣΕΠΤΕΜΒΡΙΟΣ 1996**

## Διπλωματική Εργασία

*Σχεδιασμός και Μελέτη Απόδοσης Πρωτοκόλλου Τυχαίας  
Προσπέλασης για Ασύρματα Δίκτυα Ολοκληρωμένων Υπηρεσιών  
Φωνής-Δεδομένων.*

Ιωάννης Κ. Ρομπογιαννάκης

Τμήμα Ηλεκτρονικής και Μηχανικών Υπολογιστών  
Πολυτεχνείο Κρήτης

Επιτροπή Παρακολούθησης :

1. Πατεράκης Μιχάλης, Αναπληρωτής Καθηγητής (Εισηγητής)
2. Διγαλάκης Βασίλης, Επίκουρος Καθηγητής
3. Μαράς Ανδρέας, Αναπληρωτής Καθηγητής

Χανιά, Σεπτέμβριος 1996.

# ΠΕΡΙΕΧΟΜΕΝΑ

## Περίληψη

### Κεφάλαιο 1

Ολοκληρωμένη Τυχαία Προσπέλαση Πακέτων Φωνής και Δεδομένων.....	1
---	---

1.1 Αλγόριθμοι πολλαπλής προσπέλασης .....	2
1.2 Αλγόριθμοι τυχαίας προσπέλασης .....	3
1.3 Αλγόριθμοι επίλυσης συγκρούσεων .....	6
1.3.1 Δυαδική έναντι τριαδικής ανάδραση .....	7
1.3.2 Κανόνες πρώτης μετάδοσης .....	8
1.4 Ολοκλήρωση φωνής και δεδομένων .....	10

### Κεφάλαιο 2

Μοντελοποίηση Φωνής-Δεδομένων και Μοντέλο Συστήματος Ολοκληρωμένης Προσπέλασης .....	16
--	----

2.1 Αλγόριθμοι πολλαπλής προσπέλασης .....	16
2.1.1 Αλγόριθμοι τυχαίας προσπέλασης για τη φωνή .....	19
2.1.2 Αλγόριθμοι τυχαίας προσπέλασης για τα δεδομένα .....	20
2.2 Μοντελοποίηση της φωνής και ενέργειες του σταθμού βάσης .....	20
2.3 Η αποτυχία της κατανομής <b>Poisson</b> .....	24
2.4 Η κατανομή <b>Pareto</b> .....	26
2.4.1 Ιδιότητες της κατανομής <b>Pareto</b> .....	29

### Κεφάλαιο 3

Ο Αλγόριθμος Two-Cell Stack .....	33
-----------------------------------	----

3.1 Ο αλγόριθμος Two-Cell Stack .....	33
---------------------------------------	----

3.2 Επιλογή του παραθύρου βελτίστου μήκους $\Delta$ .....	36
3.2.1 Αποτελέσματα της προσομοίωσης του αλγορίθμου Two-Cell Stack .....	36

## Κεφάλαιο 4

Μετρικές απόδοσης του συστήματος Ολοκληρωμένης Προσπέλασης για Poisson αφίξεις μηνυμάτων δεδομένων .....	39
--	----

4.1 Καθορισμός των παραμέτρων του συστήματος ολοκληρωμένης τυχαίας προσπέλασης πακέτων φωνής και δεδομένων .....	39
4.2 Αποτελέσματα προσομοίωσης .....	40
4.2.1 Κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων.....	44

## Κεφάλαιο 5

Μετρικές απόδοσης του συστήματος Ολοκληρωμένης Προσπέλασης όταν οι χρόνοι μεταξύ αφίξεων διαδοχικών μηνυμάτων δεδομένων χαρακτηρίζονται από την κατανομή Pareto.....	52
--	----

5.1 Αποτελέσματα προσομοίωσης.....	52
5.1.1 Μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων.....	53
5.1.2 Κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων.....	56
5.2 Σύγκριση των καθυστερήσεων προσπέλασης των πακέτων δεδομένων για χρόνους αφίξεων κατανεμημένους σύμφωνα με τις κατανομές <b>Pareto</b> και Εκθετική ( <b>Poisson</b> αφίξεις).....	63

## Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	70
--	----

## ΠΑΡΑΡΤΗΜΑ 1

Ο κώδικας Προσομοίωσης του Αλγορίθμου Two-Cell Stack.....	73
---	----

## **ΠΑΡΑΡΤΗΜΑ 2**

<b>Ο κώδικας Προσομοίωσης του Συστήματος Ολοκληρωμένης Τυχαίας Προσπέλασης Πακέτων Φωνής και Δεδομένων.....</b>	<b>79</b>
<b>Βιβλιογραφία.....</b>	<b>118</b>

## ΠΕΡΙΛΗΨΗ

Σε αυτή την εργασία θεωρούμε το πρόβλημα της **Ολοκληρωμένης Τυχαίας Προσπέλασης Πακέτων Φωνής και Δεδομένων** ενός χρονισμένου ασύρματου επικοινωνιακού καναλιού. Αφίξεις πακέτων ή μηνυμάτων (δηλαδή ενός συνόλου πακέτων) πληροφορίας καθώς και συνδέσεων για μεταφορά πληροφορίας σε τέτοια κανάλια συχνά μοντελοποιούνται σαν στοχαστικές διαδικασίες **Poisson**, διότι οι τελευταίες έχουν ενδιαφέρουσες θεωρητικές ιδιότητες και εξαιτίας του γεγονότος ότι έχουν μελετηθεί εκτεταμένα και έχουν κατανοηθεί σε μεγάλο βαθμό. Εν τούτοις από μελέτες της κίνησης πακέτων (μηνυμάτων) σε δίκτυα επικοινωνιών, έχει φανεί ότι οι χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων (μηνυμάτων) δεν είναι πάντα εκθετικά κατανομημένοι. Για παράδειγμα, πρόσφατες μελέτες κίνησης πακέτων (μηνυμάτων) σε επικοινωνιακό κανάλι τοπικού δικτύου έχουν δείξει ότι η συνολική κίνηση πακέτων μοντελοποιείται καλύτερα χρησιμοποιώντας στοχαστικά αυτο-όμοιες διαδικασίες, οι οποίες έχουν πολύ διαφορετικές ιδιότητες από την **Poisson**.

Εξαιτίας του σοβαρού ρόλου που αναμένεται να παίζουν οι **Ολοκληρωμένες Υπηρεσίες Φωνής - Δεδομένων** στα μελλοντικά (τρίτης γενιάς) ψηφιακά ασύρματα δίκτυα επικοινωνίας κινητών χρηστών, σε αυτή την εργασία μελετούμε το πρόβλημα της αποδοτικής ολοκλήρωσης των διαφορετικών απαιτήσεων φωνής και δεδομένων και της ελαχιστοποίησης του ανταγωνισμού (για προσπέλαση του καναλιού) μεταξύ των χρηστών των δύο παραπάνω τύπων πληροφορίας. Τα χαρακτηριστικά και οι διαφορετικές απαιτήσεις φωνής και δεδομένων προσδίδουν μεγάλο ενδιαφέρον για τους τρόπους με τους οποίους μπορεί να αντιμετωπιστεί το πρόβλημα. Επιπλέον, η υψηλή εκρηκτικότητα που χαρακτηρίζει την κίνηση πακέτων (μηνυμάτων) που δημιουργείται από εφαρμογές δεδομένων αλληλεπίδρασης (interactive data applications, οι οποίες αποτελούν σημαντική συνιστώσα των μελλοντικών ασυρμάτων επικοινωνιών, αλλά δεν έχουν προσελκύσει την ανάλογη προσοχή μέχρι τώρα), θα μπορέσει να μας δώσει ενδείξεις για το κατά πόσο οι αλγόριθμοι που έχουν επινοηθεί, σχεδιαστεί, και μελετηθεί για **Poisson** αφίξεις πακέτων (μηνυμάτων), διατηρούν τα επίπεδα απόδοσης τους όταν η κίνηση των πακέτων (μηνυμάτων) χαρακτηρίζεται από κατανομή πολύ εκρηκτικότερη της **Poisson**.

Η εργασία αρχίζει με το Κεφάλαιο 1 το οποίο αποτελεί μία συνοπτική εισαγωγή, στο θέμα της **Ολοκλήρωσης Φωνής και Δεδομένων** καθώς και στις βασικές αρχές των **Αλγορίθμων Επίλυσης Συγκρούσεων Πακέτων**. Στη συνέχεια, στο Κεφάλαιο 2, παραθέτουμε την προσέγγιση που χρησιμοποιούμε για να επιτύχουμε την **Ολοκληρωμένη Τυχαία Προσπέλαση Πακέτων Φωνής και Δεδομένων** καθώς και το αντίστοιχο μοντέλο. Επίσης, το Κεφάλαιο αυτό αναφέρεται στην κατανομή Pareto καθώς και στους λόγους για τους οποίους αυτή η κατανομή χρησιμοποιείται σε αυτή την εργασία για να χαρακτηρίσει τους χρόνους μεταξύ αφίξεων διαδοχικών πακέτων στο σύστημα μας. Το Κεφάλαιο 3 αναφέρεται στον αλγόριθμο **Two-Cell Stack** ο οποίος χρησιμοποιείται για την μετάδοση των πακέτων δεδομένων στο σύστημα της ολοκληρωμένης προσπέλασης. Τέλος, στα Κεφάλαια 4 και 5 παραθέτουμε τα αποτελέσματα της προσομοίωσης του συστήματος ολοκληρωμένης προσπέλασης για τις μετρικές απόδοσης των πακέτων δεδομένων.

## **ΚΕΦΑΛΑΙΟ 1**

## **Ολοκληρωμένη Τυχαία Προσπέλαση Πακέτων Φωνής και Δεδομένων**

Το κεφάλαιο αυτό επιχειρεί μια συνοπτική (και συνεπώς όχι πλήρη) εισαγωγή στο θέμα της **Ολοκλήρωσης Φωνής και Δεδομένων**. Ασχολείται, επίσης με τις βασικές αρχές και ιδιότητες των **Αλγορίθμων Επίλυσης Συγκρούσεων Πακέτων (Collision Resolution Algorithms)**.

### **1.1 Αλγόριθμοι Πολλαπλής Προσπέλασης**

Θεωρούμε την περίπτωση όπου γεωγραφικά κατανομημένοι, ανεξάρτητοι χρήστες, επιθυμούν να επικοινωνήσουν μεταξύ τους ή με έναν κεντρικό σταθμό, 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. Ένας τέτοιος αλγόριθμος, επίσης, θα πρέπει να πετυχαίνει τη μεγαλύτερη δυνατή χρησιμοποίηση του καναλιού εκμεταλευόμενος μεταξύ των άλλων και ορισμένα ειδικά χαρακτηριστικά της πληροφορίας που μεταφέρεται πάνω στο κανάλι.

Το πρόβλημα αυτό είναι γνωστό σαν **Πρόβλημα Πολλαπλής Προσπέλασης (Multiple Access Problem)**, και οι αλγόριθμοι που χρησιμοποιούνται **Αλγόριθμοι Πολλαπλής Προσπέλασης (Multiple Access Algorithms)** [4]. Αυτοί οι αλγόριθμοι, όπως θα δούμε στη συνέχεια, λύνουν το πρόβλημα της πολλαπλής προσπέλασης αλλά σε ορισμένες περιπτώσεις δεν πετυχαίνουν βέλτιστη χρησιμοποίηση του καναλιού.

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 (collision)** 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 στην σύγκρουση πρέπει να αναμεταδωθεί σε κάποια επόμενη χρονική στιγμή, με πιθανές επαναλήψεις της αναμετάδοσης μέχρι να μεταδοθεί επιτυχώς.

Ένας αλγόριθμος πολλαπλής προσπέλασης μπορεί να είναι συγκεντρωτικός (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 αλγόριθμοι χωρίς συγκρούσεις (collision free algorithms).

Στον αλγόριθμο FDMA, το διαθέσιμο εύρος του καναλιού (channel bandwidth) χωρίζεται σε ζώνες και κάθε ζώνη παραχωρείται σε ένα χρήστη. Στον αλγόριθμο TDMA, ο χρόνος διαιρείται σε **πλαίσια (frames)**, κάθε πλαίσιο διαιρείται σε **χρονικά τμήματα (slots)**. Κάθε χρονικό τμήμα του πλαισίου παραχωρείται σε ένα χρήστη και αυτός μπορεί να μεταδώσει μόνο στο αντίστοιχο slot σε κάθε frame. Όταν όλοι οι χρήστες έχουν πακέτα έτσι ώστε να χρησιμοποιούν συνεχώς το τμήμα του καναλιού που τους έχει παραχωρηθεί (ζώνη συχνότητας στον FDMA ή slot σε κάθε frame στον TDMA), οι δύο παραπάνω αλγόριθμοι επιτρέπουν 100% χρήση του καναλιού.

Στην αντίθετη περίπτωση όμως που κάποιοι χρήστες γεννούν πακέτα κατά εκρηκτικό τρόπο (δηλαδή για ένα χρονικό διάστημα έχουν πολλά πακέτα για μετάδοση, ενώ στη συνέχεια για κάποιο άλλο διάστημα δεν έχουν πακέτα για μετάδοση), και ο αριθμός των χρηστών που χρησιμοποιούν το κανάλι είναι μεγάλος και πιθανώς μεταβάλλεται με το χρόνο, και οι δύο παραπάνω αλγόριθμοι εισάγουν μεγάλες καθυστερήσεις, ενώ παράλληλα δεν επιτρέπουν αποδοτική χρήση του καναλιού. Αυτό συμβαίνει διότι, οι εκρηκτικοί χρήστες κατά το χρονικό διάστημα που δεν έχουν πακέτα προς μετάδοση συνεχίζουν να κρατούν δεσμευμένο το τμήμα του καναλιού που τούς αναλογεί, το οποίο από τη μια δεν χρειάζονται και από τη άλλη δεν μπορεί να χρησιμοποιηθεί από κάποιον άλλο χρήστη που έχει πακέτα προς μετάδοση. Συνεπώς, στην περίπτωση αυτή απαιτούνται διαφορετικές τεχνικές

διαμερισμού του καναλιού από αυτές που έχουν περιγραφεί παραπάνω. Με αυτές τις τεχνικές θα ασχοληθούμε στην επόμενη παράγραφο αυτού του κεφαλαίου.

## **1.2 Αλγόριθμοι Τυχαίας Προσπέλασης**

Η βασική διαφορά ανάμεσα σε ένα **Αλγόριθμο Τυχαίας Προσπέλασης** (Random Access Algorithm) και ένα 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.

Ένας αλγόριθμος τυχαίας προσπέλασης αποτελείται από το συνδυασμό του **Κανόνα Πρώτης Μετάδοσης** (First Transmission Rule) και ενός **Αλγορίθμου Επίλυσης Συγκρούσεων** (Collision Resolution Algorithm) [4]. Ο αλγόριθμος επίλυσης συγκρούσεων είναι υπεύθυνος για την επίλυση των συγκρούσεων που δημιουργούνται λόγω των ταυτόχρονων μεταδόσεων, ενώ ο κανόνας πρώτης μετάδοσης ρυθμίζει πότε οι χρήστες με “φρέσκα” πακέτα θα μεταδώσουν για πρώτη φορά στο κανάλι.

Στην συνέχεια, επικεντρώνουμε την παρουσίαση μας στην οικογένεια ευσταθών (ο ορισμός της ευστάθειας δίδεται παρακάτω) και υλοποιήσιμων αλγορίθμων τυχαίας προσπέλασης (δες σχετική συζήτηση στις σελ.289 - 293 του [4], και στο [1]). Η παραπέρα συζήτηση που ακολουθεί δεν περιλαμβάνει αλγορίθμους τυχαίας προσπέλασης τύπου ALOHA [1], [4], παρόλο που οι παραπάνω αλγόριθμοι είναι αρκετά δημοφιλείς στην πράξη, διότι είναι από τη φύση τους ασταθείς (ακόμα και όταν υπάρχουν, οι ευσταθείς παραλλαγές των αλγορίθμων τύπου ALOHA είναι δύσκολα υλοποιήσιμοι στην πράξη, [1], [4]). Ο πιο απλός ευσταθής αλγόριθμος τυχαίας προσπέλασης είναι ο **Αλγόριθμος Ελεύθερης Προσπέλασης** (Free-Access Algorithm, FAA), ο οποίος είναι γνωστός και ως αλγόριθμος στοίβας (stack), και στον οποίο ένας χρήστης μεταδίδει ένα “φρέσκο” πακέτο στο πρώτο χρονικό τμήμα που ακολουθεί την άφιξη του ακόμα και κατά τη διάρκεια εν εξελίξει επιλύσεων συγκρούσεων άλλων πακέτων. Οι υπόλοιποι ευσταθείς αλγόριθμοι (στους οποίους θα δώσουμε έμφαση στην εργασία αυτή), ονομάζονται **Αλγόριθμοι Δεσμευμένης Προσπέλασης** (Blocked-Access Algorithms, BBA), οι οποίοι είναι γνωστοί και ως

αλγόριθμοι έρευνας δέντρου (tree search), και στους οποίους ένας χρήστης μεταδίδει ένα “φρέσκο” πακέτο για πρώτη φορά ακολουθώντας έναν *κανόνα πρώτης μετάδοσης*. Σε αυτή την περίπτωση δηλαδή, κανένα “φρέσκο” πακέτο δεν επιτρέπεται να μεταδοθεί στο κανάλι κατά το χρονικό διάστημα που διαρκεί μια επίλυση σύγκρουσης.

Στη συνέχεια, το θέμα που θα διερευνηθεί είναι ο σχεδιασμός αλγορίθμων επίλυσης συγκρούσεων. Αυτοί που επιλύουν τις συγκρούσεις πακέτων όταν προκύπτουν, προσπαθώντας έτσι να διατηρούν ευσταθή τη λειτουργία του συστήματος με χαμηλές καθυστερήσεις πακέτων. Πριν όμως προχωρήσουμε, θεωρούμε χρήσιμη μια σύντομη περιγραφή και ανάλυση των παραμέτρων σχεδιασμού και υποθέσεων που χρησιμοποιούν οι αλγόριθμοι που θα αναφερθούν [1,4] :

(1) Σύγχρονα συστήματα, με τον άξονα του χρόνου διαιρεμένο σε ίσα χρονικά τμήματα (slots). Τα μηνύματα πληροφορίας των χρηστών 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 σωστή λήψη, ή επιτυχημένη μετάδοση, έχουμε αν ένας μόνο χρήστης μεταδώσει πακέτο σε ένα δεδομένο χρονικό τμήμα. Αν δύο οι 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 όλων των συγκρουόμενων πακέτων.

(3) Άμεση πληροφόρηση ανάδρασης. Στο τέλος κάθε χρονικού τμήματος κάθε χρήστης ή ίσως μόνο ένα υποσύνολο των χρηστών, πληροφορούνται ταυτόχρονα για το αν το συγκεκριμένο χρονικό τμήμα περιείχε σύγκρουση ή όχι (κενό ή επιτυχία).

(4) Άπειρος αριθμός χρηστών. Το σύστημα έχει ένα πολύ μεγάλο (θεωρητικά άπειρο) αριθμό χρηστών και κάθε “φρέσκο” πακέτο πηγαίνει σε καινούργιο χρήστη. Αυτή η υπόθεση οδηγεί στην μοντελοποίηση της χειρότερης περίπτωσης, μια και κάθε πακέτο που γεννιέται στο σύστημα είναι αναγκασμένο να χρησιμοποιήσει τυχαία προσπάθεια για να μεταδοθεί επιτυχώς στο κανάλι. Ο σχεδιασμός των αλγορίθμων κάτω από αυτή την υπόθεση παρουσιάζει ενδιαφέρον, διότι η απόδοση τους δε θα

έξαρτάται από τον αριθμό των χρηστών στο σύστημα, αλλά από τον συνολικό ρυθμό γέννησης πληροφορίας από τους χρήστες.

(5) Ευστάθεια αλγορίθμου και throughput. Ένας αλγόριθμος τυχαίας προσπέλασης καναλιού είναι ευσταθής, αν και μόνο αν, υπάρχει ένας θετικός αριθμός  $\Lambda$  τέτοιος ώστε για όλους τους ρυθμούς άφιξης νέων πακέτων  $\lambda$  με  $\lambda \leq \Lambda$ , ο ρυθμός άφιξης  $\lambda$  και ο ρυθμός αναχώρησης  $S$  (ρυθμός επιτυχών μεταδόσεων στο κανάλι) είναι ίσοι. Ο θετικός αριθμός  $\Lambda$  ονομάζεται throughput του αλγορίθμου τυχαίας προσπέλασης, ενώ η περιοχή των ρυθμών άφιξης μεταξύ 0 και  $\Lambda$  ορίζει την περιοχή ευστάθειας (stability region) του αλγορίθμου.

(6) Καθυστέρηση πακέτου, ορίζεται ο χρόνος που μεσολαβεί από τη στιγμή που γεννιέται ένα πακέτο μέχρι το τέλος του χρονικού τμήματος που περιέχει την επιτυχή μετάδοση του.

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]. Οι υποθέσεις των συγκρούσεων και της σωστής λήψης, αγνοούν την πιθανότητα λάθους που οφείλεται σε θόρυβο και την δυνατότητα χρησιμοποίησης τεχνικών σύλληψης (capture), αντίστοιχα. Στην περίπτωση της σύλληψης, ο δέκτης μπορεί να συλλέξει (να λάβει δηλαδή σωστά) μια μετάδοση πακέτου σε ένα χρονικό τμήμα που περιέχει πολλαπλές ταυτόχρονες μεταδόσεις πακέτων. Η υπόθεση της άμεσης πληροφορίας ανάδρασης μπορεί να μην είναι ρεαλιστική, ειδικά στην περίπτωση δορυφορικών καναλιών, [4]. Είναι σημαντικό εδώ να τονιστεί ότι η καθυστέρηση πληροφορίας ανάδρασης αντιμετωπίζεται με απλές αλλαγές στην λειτουργία των αλγορίθμων (που έχουν σχεδιαστεί κάτω από την υπόθεση άμεσης πληροφορίας ανάδρασης), και δεν επηρεάζει την ευστάθεια και το throughput των αλγορίθμων.

### 1.3 Αλγόριθμοι Επίλυσης Συγκρούσεων

Ο πρώτος ευσταθής **αλγόριθμος επίλυσης συγκρούσεων** προτάθηκε από τον Καπετανάκη το 1977 [3]. Ο Καπετανάκης πρότεινε ένα αλγόριθμο επίλυσης συγκρούσεων πακέτων που καταλήγει σε ευσταθές σύστημα τυχαίας προσπέλασης καναλιού, όταν ο συνολικός ρυθμός άφιξης δεν είναι υψηλός.

Ένας αλγόριθμος επίλυσης συγκρούσεων είναι ένας αλγόριθμος (με κατανεμημένη εκτέλεση 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 όσοι σχετίζονται με την επίλυση μιας σύγκρουσης) να ενημερώνονται για αυτό το γεγονός. Ένας τέτοιος αλγόριθμος επίλυσης συγκρούσεων ονομάζεται **Q-αδικός** (όπου Q ακέραιος μεγαλύτερος ή ίσος του 2), αν ένας χρήστης αφού συμμετάσχει σε μια σύγκρουση πρέπει να επιλέξει πιθανοτικά ανάμεσα σε Q εναλλακτικές δράσεις [2].

Ο βασικός μηχανισμός επίλυσης συγκρούσεων πακέτων αλγορίθμων είναι κοινός για τους αλγορίθμους τυχαίας προσπέλασης που θεωρούμε σε αυτή την εργασία. Μετά από κάθε σύγκρουση κάθε χρήστης που συμμετείχε στη σύγκρουση, ρίχνει ένα νόμισμα με Q πλευρές με τιμές 1,2,...,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). Με αυτόν τον τρόπο οι χρήστες που συμμετείχαν αρχικά στη σύγκρουση χωρίζονται σε Q υποσύνολα. Σε κάθε ένα από αυτά τα 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, ανήκουν στο **i-ιοστό** υποσύνολο με  $i = 0,1,...,Q-1$ , και 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 του δείκτη εξαιτίας συμμετοχής τους σε προηγούμενη σύγκρουση που δεν έχει ακόμη επιλυθεί, αυξάνουν τον δείκτη τους κατά  $Q - 1$  (ώστε να δημιουργήσουν χώρο στη στοίβα για τους χρήστες της τελευταίας 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 ελεύθερο χρονικό τμήμα. Εάν στο χρονικό τμήμα αυτό δεν υπάρχει σύγκρουση όλοι οι χρήστες με δείκτη μεγαλύτερο ή ίσο της μονάδας μειώνουν κατά ένα τον δείκτη

αυτό. Συνεπώς, εάν μετά από μια αρχική σύγκρουση δύο ή περισσότερων πακέτων καθένα από τα  $Q$  υποσύνολα που προκύπτουν περιέχει το πολύ ένα πακέτο, τότε η σύγκρουση θα επιλυθεί ακριβώς σε  $Q$  χρονικά τμήματα υπό την προϋπόθεση ότι δεν παρεμβάλλονται αφίξεις φρέσκων πακέτων. Το αντίστοιχο διάστημα επίλυσης σύγκρουσης θα έχει διάρκεια  $Q + 1$  χρονικά τμήματα αφού στο διάστημα αυτό προστίθεται και το αρχικό χρονικό τμήμα στο οποίο συνέβει η σύγκρουση. Το αντίστοιχο διάστημα όταν δεν έχουμε σύγκρουση (μεταδίδονται ένα ή κανένα πακέτο) είναι ίσο με ένα χρονικό τμήμα. Το τέλος μίας επίλυσης σύγκρουσης μπορεί να ανιχνευθεί από κάθε χρήστη παρατηρώντας την ανάδραση του καναλιού [14].

Όλοι οι χρήστες που συμμετέχουν στην διαδικασία επίλυσης σύγκρουσης, καθορίζουν τη δράση τους σε ένα χρονικό τμήμα βασιζόμενοι στην πληροφορία ανάδρασης που λαμβάνουν από το βασικό σταθμό (ή δέκτη) για το προηγούμενο χρονικό τμήμα. Παρατηρούμε ότι, αν και όλοι οι παραπάνω χρήστες είναι εν γένει ενεργοί (έχουν πακέτα για μετάδοση) σε παραπάνω από ένα χρονικό τμήμα της διάρκειας επίλυσης της συγκεκριμένης σύγκρουσης, δεν μεταδίδουν όλοι μαζί σε κάθε χρονικό τμήμα. Αντίθετα, παρακολουθώντας την ανάδραση του καναλιού και εφαρμόζοντας τον αλγόριθμο επίλυσης σύγκρουσης, κάνουν μια συλογική προσπάθεια, η οποία εγγυάται ένα ευσταθές σύστημα.

Οι αλγόριθμοι επίλυσης συγκρούσεων διαχωρίζονται περαιτέρω με βάση τον τύπο της ανάδρασης του καναλιού που χρησιμοποιούν.

### ***1.3.1 Δυαδική έναντι Τριαδικής Ανάδρασης***

Με τον όρο τριαδική ανάδραση εννοούμε ότι, κάθε χρήστης πληροφορείται αν στο συγκεκριμένο χρονικό τμήμα υπήρξε σύγκρουση, επιτυχημένη μετάδοση ή αν το συγκεκριμένο χρονικό τμήμα ήταν κενό (στη δυαδική ανάδραση η πληροφορία της επιτυχημένης μετάδοσης και του κενού χρονικού τμήματος συμπυκνώνεται στην πληροφορία ότι στο συγκεκριμένο χρονικό τμήμα δεν υπήρξε σύγκρουση). Η χρήση της τριαδικής ανάδρασης μπορεί να επιταχύνει μια διαδικασία επίλυσης σύγκρουσης και κατά συνέπεια μπορεί να οδηγήσει σε μεγαλύτερα throughputs. Η εφαρμογή της όμως είναι λιγότερο αξιόπιστη στην περίπτωση ύπαρξης λαθών στην ανάδραση που παρέχει το κανάλι, με αποτέλεσμα στην περίπτωση ύπαρξης λάθους ο αλγόριθμος του

Καπετανάκη να μπαίνει σε αδιέξοδο (deadlock) και να μην τερματίζει ποτέ την επίλυση σύγκρουσης, [1].

### **1.3.2 Κανόνες Πρώτης Μετάδοσης**

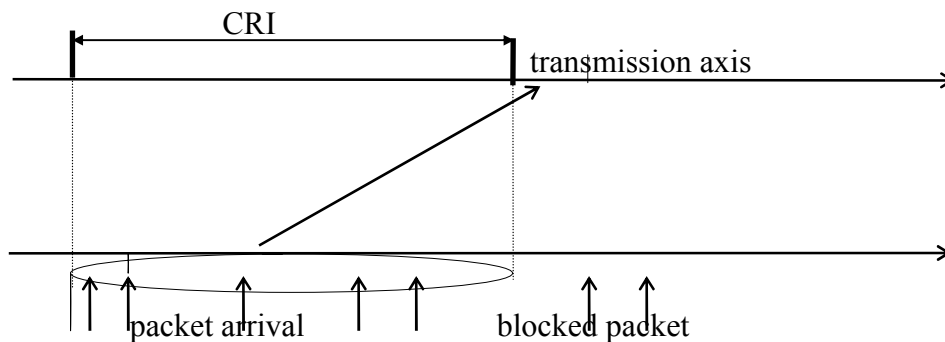
Όπως αναφέραμε παραπάνω οι αλγόριθμοι τυχαίας δεσμευμένης προσπέλασης απαιτούν ένα κανόνα πρώτης μετάδοσης, ο οποίος καθορίζει με πιο τρόπο θα μεταδίδονται για πρώτη φορά τα “φρέσκα” πακέτα. Υπενθυμίζουμε εδώ, ότι οι αλγόριθμοι ελεύθερης τυχαίας προσπέλασης επιτρέπουν στους χρήστες με “φρέσκα” πακέτα να μεταδώσουν αυτά τα πακέτα στο αμέσως επόμενο χρονικό τμήμα που ακολουθεί την άφιξή τους.

Σύμφωνα με τον προφανή κανόνα πρώτης μετάδοσης, όλοι οι χρήστες που γέννησαν “φρέσκα” πακέτα κατά τη διάρκεια ενός διαστήματος επίλυσης σύγκρουσης, μπορούν να μεταδώσουν τα πακέτα αυτά για πρώτη φορά αμέσως μόλις τελειώσει το διάστημα αυτό (σχ. 1.α). Σ ' ένα πλήρως κατανομημένο 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 πακέτο προς μετάδοση). 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].

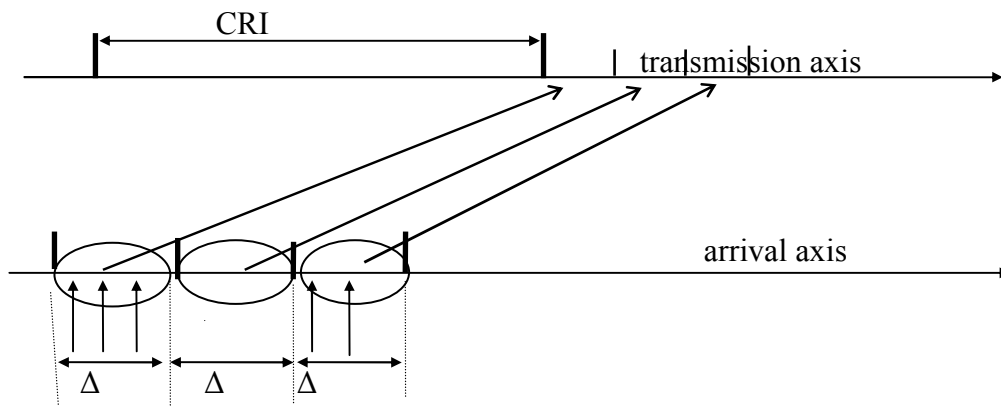
Με τον προφανή κανόνα πρώτης μετάδοσης, ο αριθμός των πακέτων που μεταδίδονται στο πρώτο χρονικό διάστημα ενός διαστήματος επίλυσης σύγκρουσης (καθώς και η διάρκεια αυτού του διαστήματος) σχετίζονται άμεσα με τη διάρκεια του προηγούμενου διαστήματος επίλυσης σύγκρουσης. Έτσι, μεγάλα διαστήματα επίλυσης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

Αυτό κάνει ο μη προφανής κανόνας πρώτης μετάδοσης, σύμφωνα με τον οποίο μόνο τα πακέτα που γεννιούνται κατά τη διάρκεια ενός χρονικού διαστήματος μεγίστου μήκους  $\Delta$ , επιτρέπεται να μεταδοθούν στο πρώτο χρονικό τμήμα που ακολουθεί ένα

διάστημα επίλυσης σύγκρουσης (σχ. 1.β). 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 μεγαλύτερο δυνατό throughput.



**Σχήμα 1.α :** Προφανής κανόνας πρώτης μετάδοσης



**Σχήμα 1.β :** Μη προφανής κανόνας πρώτης μετάδοσης

#### 1.4 Ολοκλήρωση Φωνής και Δεδομένων

Το πεδίο των ασύρματων επικοινωνιών (wireless communications) παρουσιάζει μια απότομη ανάπτυξη τα τελευταία χρόνια, ως αποτέλεσμα της μεγάλης προόδου των



ψηφιακών επικοινωνιών, των κυψελικών δικτύων (cellular networks), των φορητών υπολογιστών (portable computers) και των προσωπικών συστημάτων επικοινωνίας (personal communication systems). Η παροχή υπηρεσιών για δεδομένα (data services) αποτελεί μια πολύ σημαντική συνιστώσα των μελλοντικών ασύρματων επικοινωνιών, αλλά δεν είχε προσελκύσει μέχρι πρόσφατα την ανάλογη προσοχή [5].

Το εύρος ζώνης ενός ραδιοφωνικού καναλιού στα σημερινά κυψελικά συστήματα είναι περιορισμένο, οι απαιτήσεις για πρόσβαση συνεχίζουν να αυξάνονται με γοργούς ρυθμούς, και καμία ουσιαστική αύξηση του εύρους ζώνης δεν προβλέπεται στο άμεσο μέλλον. Η χωρητικότητα του συστήματος συνεπώς μπορεί να αυξηθεί μόνο με δύο τρόπους : χρησιμοποιώντας μικροκυψέλες (microcells), έτσι ώστε να αυξηθεί η επανάληψη χρήσης των διαθέσιμων συχνοτήτων, και/ή χρησιμοποιώντας αποδοτικά πρωτόκολλα πολλαπλής προσπέλασης τα οποία θα εκμεταλεύονται τις διάφορες ιδιότητες και θα εξυπηρετούν τις διαφορετικές απαιτήσεις για παροχή υπηρεσιών των πηγών πληροφορίας, [6], [35].

Στις πυκνοκατοικημένες περιοχές, οι μικροκυψέλες αναμένεται να φέρουν το τέλος του διαχωρισμού ανάμεσα στα ασύρματα και τα κινητά τηλέφωνα και να παρέχουν πρόσβαση στα ευρέως φάσματος (broadband) ISDN τοπικά δίκτυα για μεγάλο αριθμό κινητών χρηστών φωνής και δεδομένων. Η διάμετρος της μικροκυψέλης αναμένεται είναι της τάξης των 100 μέτρων [6]. Μέσα σε μία μικροκυψέλη, χωρικά διασκορπισμένοι χρήστες μοιράζονται ένα κανάλι πολλαπλής προσπέλασης το οποίο τους συνδέει με ένα ακίνητο σταθμό βάσης (base station). Ο σταθμός βάσης διανέμει την πληροφορία ανάδρασης προς τους χρήστες και παρέχει τη διασύνδεση με το ενσύρματο δίκτυο επικοινωνίας.

Τα πρωτόκολλα πολλαπλής προσπέλασης πρέπει να ολοκληρώνουν αποδοτικά τις διαφορετικές απαιτήσεις της φωνής και των δεδομένων. Τα πακέτα φωνής θα πρέπει να παραδίδονται στον προορισμό τους χωρίς να υπερβούν μια προκαθορισμένη μέγιστη καθυστέρηση (της τάξης των 150 - 200 ms), αλλιώς η καθυστέρηση γίνεται αντιληπτή από τον άνθρωπο. Πακέτα με καθυστέρηση μεγαλύτερη από την προκαθορισμένη μέγιστη τιμή απορρίπτονται από τον αλγόριθμο. Η ομιλία μπορεί να αναπαραχθεί χωρίς υποβάθμιση της ποιότητας της που να γίνεται αντιληπτή από τον άνθρωπο, ακόμα και με απώλεια της τάξεως του 1 - 2% των συνολικών πακέτων φωνής, εξ' αιτίας της πλεονάζουσας πληροφορίας που περιέχεται στην

κωδικοποιημένη ψηφιακή φωνή. Τα δεδομένα, από την άλλη πλευρά, είναι περισσότερο ανεκτικά στις καθυστερήσεις (καθυστερήσεις μεγαλύτερες από 200ms είναι αποδεκτές), αλλά συνήθως απαιτούν 100% παράδοση σωστών πακέτων (π.χ στην μεταφορά ενός αρχείου) [5,6].

Οι αλγόριθμοι τυχαίας προσπέλασης με κρατήσεις στο χρόνο (Reservation Random Access Algorithms) έχουν προταθεί για να αυξήσουν το throughput σε ένα δυναμικό κανάλι πολλαπλής προσπέλασης. Οι παραπάνω αλγόριθμοι συνδυάζουν ένα αλγόριθμο τυχαίας προσπέλασης που βασίζεται στον ανταγωνισμό των χρηστών για προσπέλαση στο κανάλι (contention - based algorithm), με ένα αλγόριθμο διαίρεσης στο χρόνο (TDMA). Με βάση τις ακριβείς απαιτήσεις των χρηστών φωνής, όσον αφορά την καθυστέρηση των πακέτων φωνής, μέχρι σήμερα έχουν προταθεί δύο κατηγορίες τέτοιων αλγορίθμων για τα μελλοντικά ασύρματα δίκτυα [5], [35].

Οι δύο αυτές κατηγορίες έχουν τις εξής ομοιότητες. Το κανάλι διαιρείται σε πλαίσια (frames), κάθε πλαίσιο υποδιαιρείται σε χρονικά τμήματα (slots), όπου κάθε χρονικό τμήμα μπορεί να είναι **διαθέσιμο** (available) ή **κρατημένο** (reserved). Οι χρήστες με πακέτα φωνής αλλά χωρίς κράτηση πρέπει να ανταγωνιστούν για τα διαθέσιμα χρονικά τμήματα. Οι χρήστες φωνής με κράτηση μεταδίδουν τα πακέτα τους με ένα ελεγχόμενο τρόπο στο αντίστοιχο χρονικό τμήμα χωρίς παρεμβολές.

Οι δύο παραπάνω κατηγορίες διαφέρουν στην υλοποίηση του μηχανισμού με βάση τον οποίον οι χρήστες ανταγωνίζονται για διαθέσιμα χρονικά τμήματα. Στη πρώτη περίπτωση, οι χρήστες φωνής με πακέτα αλλά χωρίς κράτηση χρησιμοποιούν τα ίδια τα πακέτα φωνής για να προσπελάσουν άμεσα τα διαθέσιμα χρονικά τμήματα [6], [35]. Στη δεύτερη περίπτωση, ένα τμήμα του χρονικού πλαισίου χωρίζεται σε μικροτμήματα (minislots), και οι χρήστες ανταγωνίζονται για τα διαθέσιμα χρονικά τμήματα προσπαθώντας να μεταδώσουν επιτυχώς ειδικά πακέτα κρατήσεων μικρής διάρκειας (reservation minipackets) σε αυτά τα μικροτμήματα, [6], [10].

Στους αλγορίθμους τυχαίας προσπέλασης, στους οποίους οι χρήστες χρησιμοποιούν τα πακέτα τους για να προσπελάσουν άμεσα τα διαθέσιμα χρονικά τμήματα, δύο μέθοδοι είναι πιθανές για την ολοκληρωμένη προσπέλαση πακέτων φωνής και δεδομένων. Σύμφωνα με τη πρώτη μέθοδο, οι χρήστες δεδομένων έχουν τη δυνατότητα να ανταγωνιστούν άμεσα με τους χρήστες φωνής, όχι όμως με τους ίδιους όρους, μια και η φωνή λόγω του ότι όπως εξηγήθηκε παραπάνω είναι

ευαίσθητη στις καθυστερήσεις, πρέπει να έχει προτεραιότητα έναντι των δεδομένων. Η δεύτερη μέθοδος χωρίζει τα διαθέσιμα χρονικά τμήματα σε διαθέσιμα τμήματα φωνής και διαθέσιμα τμήματα δεδομένων, για να ελαχιστοποιήσει τον ανταγωνισμό μεταξύ χρηστών δεδομένων και φωνής [6], [35].

Ο αλγόριθμος PRMA (Packet Reservation Multiple Access) [7], είναι ένα παράδειγμα της πρώτης περίπτωσης. Σύμφωνα με τον αλγόριθμο αυτό, οι χρήστες στην προσπάθειά τους να κάνουν κάποια κράτηση χρησιμοποιούν τον αλγόριθμο τυχαίας προσπέλασης ALOHA, [4]. Αν κάποιος χρήστης φωνής (δεδομένων) δεν καταφέρει να μεταδώσει επιτυχώς το πακέτο του στο πρώτο διαθέσιμο χρονικό τμήμα, μεταδίδει ξανά αυτό το πακέτο με πιθανότητα  $p(q)$  στο αμέσως επόμενο διαθέσιμο χρονικό τμήμα. Αυτή η διαδικασία συνεχίζεται έως ότου ο χρήστης να μεταδώσει επιτυχώς το πακέτο του. Οι παράμετροι  $p(q)$  (permission probabilities) αποτελούν σχεδιαστικές παραμέτρους του συστήματος. Για να υποστηριχτεί η προτεραιότητα των πακέτων φωνής έναντι των πακέτων δεδομένων, τίθεται  $p > q$ . Στην δεύτερη περίπτωση [6-9] και [35], οι συγκρούσεις μεταξύ πακέτων φωνής και δεδομένων εξαλείφονται με την επιλογή ενός κατάλληλου αλγορίθμου τυχαίας προσπέλασης με κρατήσεις στο χρόνο για τη φωνή που να επιτρέπει στα τερματικά δεδομένων να αναγνωρίζουν το τέλος της περιόδου ανταγωνισμού των χρηστών φωνής (Voice Contention Period, VCP) παρακολουθώντας απλά την ανάδραση του καναλιού. Κατ' αυτόν τον τρόπο, οι χρήστες δεδομένων προσπαθούν να μεταδώσουν τα πακέτα τους στα διαθέσιμα χρονικά τμήματα του πλαισίου, μόνο εφόσον ο ανταγωνισμός μεταξύ χρηστών φωνής με πακέτο αλλά χωρίς κράτηση έχει τελειώσει σε κάποιο χρονικό τμήμα νωρίτερα από το τέλος του πλαισίου. Συνεπώς, σε αυτή την περίπτωση οι χρήστες φωνής έχουν απόλυτη προτεραιότητα έναντι των χρηστών δεδομένων, και οι τελευταίοι δεν παρεμβάλλονται στις μεταδόσεις φωνής. Και στις δύο περιπτώσεις, όμως ο σκοπός του αλγορίθμου είναι να παρέξει ένα ικανοποιητικό throughput για τα δεδομένα, ενώ παράλληλα θα υποστηρίζει την προτεραιότητα της φωνής.

Οι αλγόριθμοι τυχαίας προσπέλασης οι οποίοι έχουν μηχανισμό ο οποίος εξαλείφει τον ανταγωνισμό ανάμεσα στους χρήστες φωνής και δεδομένων για τα διαθέσιμα χρονικά τμήματα αποτελούν μία πολλά υποσχόμενη εναλλακτική λύση αντί του αλγορίθμου PRMA [5, 6]. Αυτοί οι αλγόριθμοι με το να επιτρέπουν στους χρήστες

δεδομένων να αναγνωρίζουν το τέλος της περιόδου ανταγωνισμού για τη φωνή, παρέχουν σε όλους τους χρήστες μέσα σε μια μικροκυψέλη τη δυνατότητα διαχωρισμού ανάμεσα σε διαθέσιμα χρονικά τμήματα φωνής και δεδομένων (μια δυνατότητα που δεν την παρέχει ο PRMA). Ξεχωρίζοντας τους δύο διαφορετικούς τύπους μετάδοσης και επιλύοντας πρώτα τα ανταγωνιζόμενα πακέτα φωνής επιβάλλεται η προτεραιότητα της φωνής. Επιπλέον, οι αλγόριθμοι για τη μετάδοση πακέτων φωνής μπορούν να συνδυαστούν με αποδοτικούς αλγορίθμους επίλυσης συγκρούσεων πακέτων δεδομένων για να αυξηθεί το throughput των δεδομένων, χωρίς να επηρεάζεται η απόδοση της φωνής.

Όσον αφορά τις λεπτομέρειες του τρόπου με τον οποίο ελαχιστοποιείται ο ανταγωνισμός μεταξύ χρηστών φωνής και δεδομένων για τα διαθέσιμα χρονικά τμήματα, η αρχική πρόταση ήταν κάθε χρονικό πλαίσιο να αρχίζει με μια περίοδο ανταγωνισμού για τους χρήστες φωνής (αν κάποιος απ' αυτούς είχε πακέτα να μεταδώσει) [5, 6, 9, 35]. Στη διάρκεια της περιόδου αυτής μόνο πακέτα φωνής επιτρέπεται να μεταδωθούν, ενώ μόνο πακέτα δεδομένων επιτρέπεται να μεταδωθούν στα διαθέσιμα χρονικά τμήματα που ακολουθούν την περίοδο ανταγωνισμού της φωνής. Έτσι, για το πλαίσιο που φαίνεται στο **Σχήμα 2.α**, τα τμήματα 2, 4, 8 και 9 είναι διαθέσιμα (Av) για φωνή, τα τμήματα 11 και 13 είναι διαθέσιμα (Ad) για δεδομένα, ενώ τα υπόλοιπα τμήματα είναι κρατημένα (R) είτε για φωνή είτε για δεδομένα.

Με τη μέθοδο αυτή όμως, εάν ένας χρήστης φωνής γεννήσει ένα πακέτο σε ένα χρονικό τμήμα μετά το τέλος της περιόδου ανταγωνισμού για τη φωνή, θα πρέπει να περιμένει ως την αρχή του επόμενου πλαισίου για να προσπαθήσει να προσπελάσει ένα διαθέσιμο χρονικό τμήμα. Αυτό το γεγονός προσθέτει μία επιπλέον σημαντική καθυστέρηση σε αυτά τα πακέτα φωνής. Το πρόβλημα που προκύπτει μπορεί να αντιμετωπισθεί με τη δεύτερη μέθοδο, η οποία προτείνεται στο [8]. Σύμφωνα με τη

Frame i

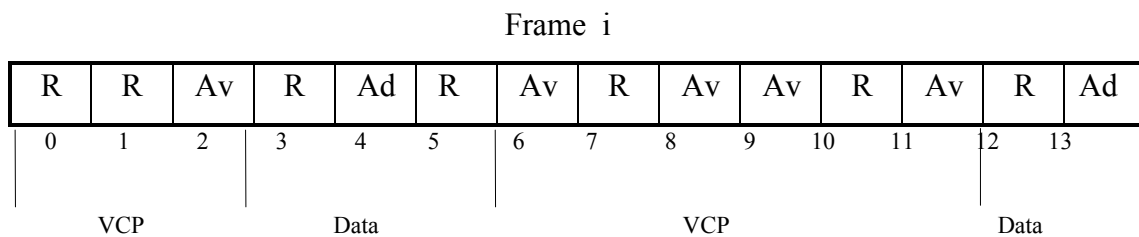
R	R	Av	R	Av	R	R	R	Av	Av	R	Ad	R	Ad
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Voice Contention Period (VCP)

Data

**Σχήμα 2.α :** Υπάρχει ακριβώς μία περίοδος ανταγωνισμού για τη φωνή στην αρχή κάθε πλαισίου.

---



**Σχημα 2.β :** Οι περίοδοι ανταγωνισμού για τη φωνή συμβαίνουν σε όλη τη διάρκεια του πλαισίου.

---

μέθοδο αυτή, περίοδοι ανταγωνισμού για τη φωνή μπορούν να συμβούν σε όλη τη διάρκεια του πλαισίου (σε αντίθεση με την προηγούμενη περίπτωση που υπήρχε μόνο μία τέτοια περίοδος στην αρχή του κάθε πλαισίου). Η ολοκληρωμένη προσπέλαση φωνής και δεδομένων επιτυγχάνεται επιτρέποντας μόνο σε χρήστες δεδομένων να μεταδίδουν στο πρώτο διαθέσιμο χρονικό τμήμα που ακολουθεί το τέλος κάθε περιόδου ανταγωνισμού για τη φωνή. Έτσι για το πλαίσιο που φαίνεται στο **Σχήμα 2.β** τα χρονικά τμήματα 2, 6, 8, 9 και 11 είναι διαθέσιμα για φωνή, ενώ τα 4 και 13 είναι διαθέσιμα για δεδομένα. Η μέθοδος αυτή, από τη μία δεν προσθέτει επιπλέον πολυπλοκότητα στο σύστημα, από τη άλλη δίνει μικρότερες καθυστερήσεις προσπέλασης (access delay) για τη φωνή, σημαντικά μικρότερη πιθανότητα να απορριφθεί από τον αλγόριθμο ένα πακέτο φωνής (εξαιτίας της υπέρβασης της μέγιστης καθυστέρησης), ενώ διατηρεί το throughput των πακέτων δεδομένων στα ίδια επίπεδα με την πρώτη μέθοδο.

## **ΚΕΦΑΛΑΙΟ 2**

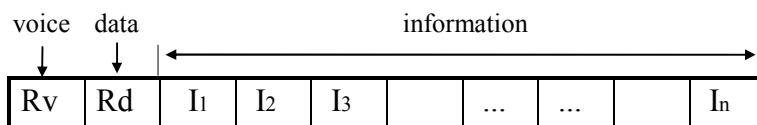
**Μοντελοποίηση Φωνής - Δεδομένων και Μοντέλο  
Συστήματος Ολοκληρωμένης Προσπέλασης**

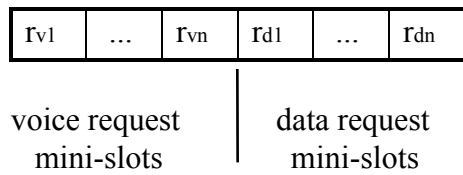
Στο κεφάλαιο αυτό παραθέτουμε την προσέγγιση που θα χρησιμοποιήσουμε για να επιτύχουμε την **Ολοκληρωμένη Τυχαία Προσπέλαση Πακέτων Φωνής και Δεδομένων** καθώς και το αντίστοιχο μοντέλο. Κάνουμε, επίσης, μια συνοπτική εισαγωγή στην εκρηκτική κατανομή **Pareto**, η οποία χρησιμοποιείται για να περιγράψει τους χρόνους μεταξύ διαδοχικών αφίξεων μηνυμάτων δεδομένων στο παραπάνω σύστημα.

## 2.1 Αλγόριθμοι Τυχαίας Προσπέλασης

Το εύρος ζώνης του καναλιού (channel bandwidth) είναι οργανωμένο με τη μορφή περιοδικών χρονικών πλαισίων (frames) σταθερής διάρκειας. Η διάρκεια κάθε frame επιλέγεται έτσι ώστε ένας χρήστης φωνής σε περίοδο ομιλίας (talkspurt) να “γεννάει” ακριβώς ένα πακέτο ανά frame. Όπως μπορούμε να δούμε στο **Σχήμα 2.1**, κάθε frame αποτελείται από τρία χρονικά διαστήματα. Ένα διάστημα αιτήσεων για τη φωνή (voice request interval), ένα διάστημα αιτήσεων για τα δεδομένα (data request interval), και το διάστημα πληροφορίας (information interval). Κατά το διάστημα πληροφορίας, κάθε slot εξυπηρετεί ακριβώς ένα, σταθερού μήκους, πακέτο το οποίο περιέχει πληροφορία φωνής (ή δεδομένων) και την επικεφαλίδα (header).

Κάθε ένα από τα διαστήματα αιτήσεων αποτελείται από ένα slot, και υποδιαιρείται σε mini-slots. Σε κάθε mini-slot μπορεί να μεταδοθεί ακριβώς ένα, σταθερού μήκους, πακέτο αίτησης (request packet). Τόσο για την κίνηση φωνής, όσο και για την κίνηση δεδομένων, κάθε πακέτο αίτησης πρέπει να συμπεριλαμβάνει την ταυτότητα της πηγής του πακέτου (source identifier). Για την κίνηση δεδομένων το πακέτο αίτησης ίσως να συμπεριλαμβάνει και επιπλέον πληροφορία, όπως το μήκος του μηνύματος που πρόκειται να μεταδοθεί, την προτεραιότητα του κ.λ.π. Για να απλοποιήσουμε την παρουσίαση μας, θεωρούμε ότι και τα δύο διαστήματα αιτήσεων αποτελούνται από τον ίδιο αριθμό mini-slots. Θεωρούμε, επίσης, ότι οι χρήστες δεδομένων χρειάζονται ένα slot πληροφορίας σε κάθε frame.





**Σχήμα 2.1:** Απεικόνιση των διαστημάτων αιτήσεων για τη φωνή και τα δεδομένα, καθώς και του διαστήματος πληροφορίας για ένα frame.

---

Οι χρήστες φωνής (δεδομένων) με πακέτα προς μετάδοση και χωρίς κράτηση slot πληροφορίας, ανταγωνίζονται για να αποκτήσουν δυνατότητα μετάδοσης (να κάνουν κάποια κράτηση ελεύθερου slot πληροφορίας), χρησιμοποιώντας ένα αλγόριθμο τυχαίας προσπέλασης για να μεταδώσουν τα πακέτα αιτήσεων τους μόνο κατά τη διάρκεια του διαστήματος αίτησης φωνής (δεδομένων). Ο σταθμός βάσης εκπέμπει ένα σύντομο δυαδικό σήμα ανάδρασης στο τέλος του κάθε mini-slot, το οποίο περιέχει μόνο την πληροφορία ύπαρξης ή όχι σύγκρουσης στο συγκεκριμένο mini-slot. Επειδή το πακέτο ανάδρασης είναι μικρό (μερικά bits) η καθυστέρηση μετάδοσης του στη μικροκυψέλη θεωρείται μηδενική. Αυτό μας δίνει τη δυνατότητα να θεωρήσουμε ότι η πληροφορία ανάδρασης είναι διαθέσιμη αμέσως, πριν το επόμενο mini-slot, στους χρήστες. Όταν ένας χρήστης μεταδώσει επιτυχώς ένα πακέτο αίτησης σε ένα mini-slot, περιμένει έως το τέλος του frame για να μάθει για ποιο slot έχει κάνει κράτηση. Στην αντίθετη περίπτωση, που ο χρήστης δεν καταφέρει να μεταδώσει επιτυχώς ένα πακέτο αίτησης σε κανένα από τα mini-slot του διαστήματος αιτήσεων, προσπαθεί ξανά στο διάστημα αιτήσεων του επόμενου frame (στην περίπτωση τέτοιου χρήστη φωνής υπάρχει η περίπτωση να απορριφθούν από τον αλγόριθμο κάποια πακέτα φωνής που έχουν υπερβεί τη μέγιστη επιτρεπτή καθυστέρηση). Ένας χρήστης με κράτηση slot πληροφορίας μεταδίδει ελεύθερα (χωρίς παρεμβολές) κατά τη διάρκεια του slot για το οποίο έχει κάνει κράτηση στα επόμενα frames.

Ο σταθμός βάσης για να μπορεί να γνωρίζει τον τρόπο με τον οποίο είναι καταναμεμένοι οι πόροι του καναλιού (channel resources) κρατάει σε ένα δυναμικό πίνακα πληροφορίες για τους ενεργούς χρήστες μέσα στη μικροκυψέλη. Έτσι για παράδειγμα, ο πίνακας αυτός μπορεί να περιέχει πληροφορίες για την ταυτότητα των



χρηστών της μικροκυψέλης, τη “ροή πληροφορίας” που χρησιμοποιεί κάθε χρήστης (virtual circuit), τους πόρους του καναλιού που έχουν δεσμευθεί γι’ αυτόν και πληροφορίες για την ποιότητα των παρεχόμενων υπηρεσιών (quality of service, QOS). Όταν ένα πακέτο αίτησης μεταδοθεί επιτυχώς, ο σταθμός βάσης το αναγνωρίζει και βάζει την αντίστοιχη αίτηση σε μια ουρά εξυπηρέτησης αιτήσεων (request queue). Ο σταθμός βάσης κατανέμει τους πόρους του καναλιού που απαιτούνται για την εξυπηρέτηση των αιτήσεων που έχει λάβει (οι οποίες βρίσκονται στην ουρά εξυπηρέτησης αιτήσεων) στο τέλος του frame, αν είναι διαθέσιμοι. Στην περίπτωση που δεν υπάρχουν αυτοί οι πόροι η αίτηση παραμένει στην ουρά. Οι χρήστες φωνής με αίτηση στην ουρά και οι χρήστες δεδομένων με πακέτα πρέπει συνεχώς να παρακολουθούν το κανάλι στην κατεύθυνση από το σταθμό βάσης προς τους κινητούς χρήστες. Όταν ολοκληρωθεί μια μετάδοση, ή όταν ένας ενεργός χρήστης βρεθεί έξω από τα όρια της μικροκυψέλης, ο σταθμός βάσης, μετά από ένα προκαθορισμένο χρονικό διάστημα, θα διαγράψει τις πληροφορίες του αντίστοιχου χρήστη από τον πίνακα του και θα απελευθερώσει τους πόρους του καναλιού που είχε δεσμεύσει για τον παραπάνω χρήστη.

Στην εργασία αυτή, θα εστιάσουμε την προσοχή μας στην περίπτωση που πρόσβαση στο κανάλι μπορεί να έχουν ένας καθορισμένος αριθμός από κινητούς χρήστες. Επιπλέον, θα κάνουμε τις παρακάτω υποθέσεις όσον αφορά τις πολιτικές εξυπηρέτησης και κατανομής των πόρων του καναλιού. Κατά την εξυπηρέτηση μίας αίτησης, ο σταθμός βάσης δίνει πάντα σ’ αυτή την αίτηση το κοντινότερο (χρονικά) ελεύθερο slot πληροφορίας. Στη συνέχεια, θεωρούμε ότι η φωνή έχει μεγαλύτερη προτεραιότητα από τα δεδομένα. Έτσι, ο σταθμός βάσης θα εξυπηρετεί όλες τις αιτήσεις φωνής πριν εξυπηρετήσει κάποια αίτηση δεδομένων. Για κάθε κατηγορία κίνησης με την ίδια προτεραιότητα η εξυπηρέτηση γίνεται σύμφωνα με το μοντέλο FCFS (First Come First Served). Τέλος, αν κάποια νέα αίτηση φωνής έρθει και βρεί το frame κατειλημένο (χωρίς ελεύθερο slot πληροφορίας), ο σταθμός βάσης προσπαθεί να εξυπηρετήσει την αίτηση φωνής ακυρώνοντας μία κράτηση για δεδομένα (αν υπάρχει). Όταν μια κράτηση δεδομένων ακυρώνεται, ο σταθμός βάσης πληροφορεί τον αντίστοιχο χρήστη δεδομένων, και τοποθετεί μία αίτηση γι’ αυτόν τον χρήστη δεδομένων στην κορυφή της ουράς εξυπηρέτησης των αιτήσεων για τα δεδομένα (data request queue).

### ***2.1.1 Αλγόριθμοι Τυχαίας Προσπέλασης για τη Φωνή***

Για τη μετάδοση των πακέτων αίτησης οι χρήστες φωνής θεωρούμε ότι χρησιμοποιούν τον **ιδανικό** αλγόριθμο. Σύμφωνα με τον αλγόριθμο αυτό, όλα τα πακέτα αίτησης που υπάρχουν στην αρχή ενός frame μεταδίδονται επιτυχώς μέσα στη διάρκεια του διαστήματος αιτήσεων φωνής του frame. Χρησιμοποιούμε τον αλγόριθμο αυτό για δύο λόγους. Πρώτον, επειδή παρέχει ένα άνω όριο στην χωρητικότητα φωνής του συστήματος (στον αριθμό των χρηστών φωνής που μπορεί να εξυπηρετήσει το σύστημα, υπό τον περιορισμό ότι η πιθανότητα να απορριφθεί ένα πακέτο φωνής είναι μικρότερη από 1%). Δεύτερον, διότι παρέχει ένα κάτω όριο στην καθυστέρηση πρόσβασης για τους χρήστες φωνής (voice access delay). Με τον όρο καθυστέρηση πρόσβασης εννοούμε το χρονικό διάστημα από τη στιγμή που ένας χρήστης γίνεται ενεργός γεννώντας το πρώτο του πακέτο έως το τέλος της πρώτης επιτυχημένης μετάδοσης πακέτου φωνής σε ένα δεσμευμένο (reserved) slot.

### ***2.1.2 Αλγόριθμοι Τυχαίας Προσπέλασης για τα Δεδομένα***

Για την μετάδοση των πακέτων αίτησης οι χρήστες δεδομένων χρησιμοποιούν τον αλγόριθμο τυχαίας προσπέλασης **Two - Cell Stack** [12]. Ο αλγόριθμος αυτός παρουσιάζεται αναλυτικά στο Κεφάλαιο 3 αυτής της εργασίας. Επίσης, χρησιμοποιείται ο *μη προφανής κανόνας πρώτης μετάδοσης* (δες παράγραφο 1.3.2) νεογεννηθέντων μηνυμάτων (messages) δεδομένων. Ο όρος μηνύματα δεδομένων χρησιμοποιείται για να γίνει διαχωρισμός μεταξύ των πακέτων αίτησης και πληροφορίας, και υπονοεί ότι το γεγονός της άφιξης πληροφορίας προς μετάδοση σε ένα χρήστη δεδομένων μπορεί να αποτελείται από παραπάνω του ενός πακέτα πληροφορίας.

## ***2.2 Μοντελοποίηση της Φωνής και Ενέργειες του Σταθμού Βάσης***

Πρίν προχωρήσουμε στην παρουσίαση του μοντέλου για την πηγή φωνής θα ήταν χρήσιμο να αναφέρουμε ορισμένες βασικές υποθέσεις πάνω στις οποίες στηρίζεται το μοντέλο αυτό [10].

(1) Μία πηγή φωνής εναλλάσσεται μεταξύ περιόδων ομιλίας (talkspurt) και περιόδων ησυχίας (silence). Όλες οι μεταβάσεις της πηγής φωνής μεταξύ των δύο παραπάνω καταστάσεων (αλλαγές κατάστασης) υποτίθεται ότι συμβαίνουν στην αρχή ενός frame. Έτσι, ένας ανταγωνιζόμενος χρήστης φωνής ο οποίος αποτυγχάνει να μεταδώσει επιτυχώς την αίτηση του κατά τη διάρκεια του διαστήματος αιτήσεων απορρίπτει στο τέλος του frame το πακέτο φωνής και προσπαθεί να μεταδώσει την αίτηση του σε ένα από τα διαστήματα αιτήσεων των επόμενων frames, έως είτε να έχει μία επιτυχημένη μετάδοση αίτησης, είτε να μεταβεί σε κατάσταση ησυχίας.

(2) Το κανάλι είναι χωρίς λάθη (error free) και δεν λαμβάνει χώρα το φαινόμενο της σύλληψης (δες παράγραφο 1.2). Επιπλέον, ο σταθμός βάσης εκπέμπει σωστά την ανάθεση των πόρων του δικτύου πριν την αρχή του επόμενου frame. Ως αποτέλεσμα, λάθη στο σύστημα μπορούν να συμβούν μόνο όταν έχουμε την ταυτόχρονη μετάδοση δύο ή περισσότερων πακέτων αίτησης σε ένα mini-slot.

(3) Τα δεσμευμένα slots απελευθερώνονται αμέσως (immediate deallocation scheme). Αυτό υπονοεί ότι ένας χρήστης φωνής ή δεδομένων που τελειώνει τη μετάδοση του ελευθερώνει αμέσως το slot πληροφορίας το οποίο είχε δεσμεύσει και έτσι αυτό μπορεί να χρησιμοποιηθεί από κάποιον άλλο χρήστη στο αμέσως επόμενο frame. Σε διαφορετικά σχήματα (delayed deallocation schemes), ο σταθμός βάσης θεωρεί ότι ένας ενεργός χρήστης (φωνής ή δεδομένων) τελείωσε τη μετάδοση του βλέποντας ότι σε ένα δεσμευμένο (από τον αντίστοιχο χρήστη) slot πληροφορίας δεν υπήρξε μετάδοση. Σ' αυτή την περίπτωση χάνεται ένα slot πληροφορίας κάθε φορά που ένας χρήστης τελειώνει την μετάδοση του.

(4) Ο αριθμός των ενεργών χρηστών φωνής στο σύστημα είναι σταθερός για το συγκεκριμένο χρονικό διάστημα που μας ενδιαφέρει. Η υπόθεση είναι λογική γιατί οι αλλαγές στον αριθμό των ενεργών χρηστών συμβαίνουν ανά χρονικά διαστήματα της τάξης των δεκάδων δευτερολέπτων, ενώ η διάρκεια του frame είναι της τάξης των milliseconds.

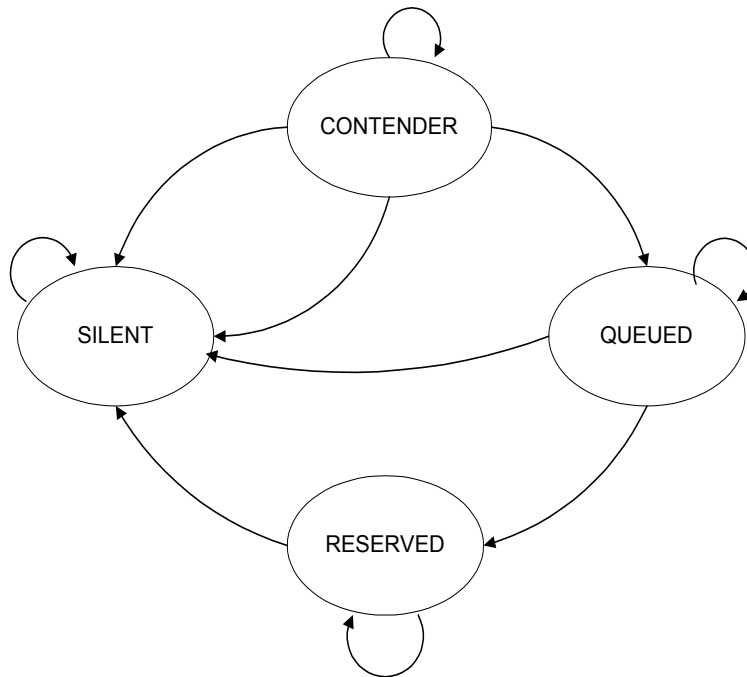
(5) Κάθε χρήστης φωνής είναι εφοδιασμένος με ένα *ανιχνευτή δραστηριότητας φωνής* (voice activity detector) ο οποίος ανιχνεύει τα διαστήματα που ο χρήστης αυτός είναι ενεργός (διαστήματα ομιλίας με διάρκεια τουλάχιστον 20ms) και τα κενά διαστήματα ή διαστήματα παύσης (διάρκειας τουλάχιστον 200ms). Έτσι, η φωνή μπορεί να μοντελοποιηθεί με μία αλυσίδα Markov δύο καταστάσεων διακριτή στο χρόνο (two-

state discrete Markov chain). Οι μεταβάσεις από ομιλία σε ησυχία συμβαίνουν με πιθανότητα  $p_{TS}$  και οι μεταβάσεις από ησυχία σε ομιλία συμβαίνουν με πιθανότητα  $p_{ST}$ . Τα διαστήματα ομιλίας και ησυχίας ακολουθούν τη γεωμετρική κατανομή με μέση τιμή  $1 / p_{TS}$  και  $1 / p_{ST}$  frames, αντίστοιχα. Στη μόνιμη κατάσταση, η πιθανότητα ένας χρήστης να βρίσκεται σε κατάσταση ομιλίας,  $p_T$ , και σε κατάσταση ησυχίας,  $p_S$ , δίδονται από τις ακόλουθες σχέσεις :

$$p_T = \frac{p_{ST}}{p_{ST} + p_{TS}} \text{ και } p_S = 1 - p_T$$

Το μοντέλο για την κίνηση φωνής που θα παρουσιάσουμε παρακάτω είναι ένα γενικό μοντέλο. Ο ιδανικός αλγόριθμος, τον οποίο υποθέτουμε ότι χρησιμοποιούν οι χρήστες φωνής για την τυχαία προσπέλαση στο κανάλι, θεωρεί μία απλοποιημένη μορφή του μοντέλου αυτού. Όλοι οι χρήστες φωνής που ανταγωνίζονται για τα διαθέσιμα slots πληροφορίας, μεταδίδουν επιτυχώς τα πακέτα των αιτήσεων τους στο διάστημα αιτήσεων φωνής του frame.

Όπως φαίνεται στο **Σχήμα 2.2**, ένας ενεργός χρήστης φωνής μπορεί να βρίσκεται σε μία από τις παρακάτω τέσσερις καταστάσεις : *ησυχίας* (silent), *ανταγωνισμού* (contender), *εξυπηρέτησης αίτησης* (queued) και *κράτησης* (reserved). Ένας χρήστης φωνής σε κατάσταση ησυχίας δεν έχει πακέτα να μεταδώσει και κατά συνέπεια δεν προσπαθεί να προσπελάσει το κανάλι. Μόλις ο χρήστης γίνει ενεργός, μεταβεί δηλαδή σε περίοδο ομιλίας, εισέρχεται στην κατάσταση ανταγωνισμού και παραμένει εκεί μέχρι να μεταδώσει επιτυχώς ένα πακέτο αίτησης. Η χρονική αυτή διάρκεια είναι εν γένει ίση με ακέραιο αριθμό frames, ας υποθέσουμε  $j$  frames,  $j \geq 1$ , στην διάρκεια της οποίας ο χρήστης φωνής απορρίπτει  $j-1$  πακέτα του talkspurt του.



**Σχήμα 2.2 :** Το διάγραμμα καταστάσεων για ένα ενεργό χρήστη φωνής

Στην περίπτωση του ιδανικού αλγορίθμου που υποθέτουμε,  $j=1$  και δεν έχουμε απόρριψη πακέτου φωνής σε αυτό το στάδιο. Όταν το πακέτο αίτησης του χρήστη μεταδοθεί επιτυχώς, η αντίστοιχη αίτηση μπαίνει στην ουρά εξυπηρέτησης αιτήσεων στο σταθμό βάσης και παραμένει εκεί έως ο σταθμός βάσης να δώσει στον χρήστη φωνής που έχει κάνει αυτή την αίτηση μία κράτηση για ένα άδειο slot πληροφορίας (αυτό, στην καλύτερη, περίπτωση, μπορεί να συμβεί στο τέλος του ίδιου frame που έγινε η αίτηση). Τέλος όταν ο χρήστης κάνει μία κράτηση μπαίνει στην κατάσταση κράτησης και μεταδίδει ένα πακέτο φωνής σε κάθε frame, στο αντίστοιχο slot για το οποίο έχει κάνει κράτηση έως ότου φύγει από την περίοδο ομιλίας (και βρεθεί ξανά σε περίοδο ησυχίας).

### ***2.3 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 των 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, [16].

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 την συμπεριφορά της. Σύμφωνα με το [16], η κίνηση σε **Τοπικό Δίκτυο** (Local Access Network, LAN) μοντελοποιείται καλύτερα χρησιμοποιώντας στατιστικά αυτό-όμοιες (self-similar) διαδικασίες [16], [22], οι οποίες έχουν πολύ διαφορετικές ιδιότητες από την **Poisson** (αυτό-όμοιες ονομάζονται οι διαδικασίες των οποίων οι συναρτήσεις αυτοσυσχέτισης φθίνουν κατά υπερβολικό τρόπο (hyperbolically decaying)). 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 μία 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 συνένωση πακέτων μεμονωμένων συνδέσεων σε

δίκτυα ευρείας γεωγραφικής περιοχής. Αυτές οι διαδικασίες αφίξεων φαίνεται να μοντελοποιούνται ικανοποιητικά με τη χρήση στατιστικά αυτό-όμοιων διαδικασιών. Για παράδειγμα, στο [16] έχει επίσης βρεθεί ότι η χρήση του μοντέλου **Poisson** με σταθερό ρυθμό αφίξεων, υποεκτιμά 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 μία σύνδεση TELNET υποεκτιμά την εκρηκτικότητα της κίνησης της σύνδεσης, [16]. 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 μίας 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 το ηλεκτρονικό ταχυδρομείο (e-mail, SMTP) και τα νέα δικτύου (NNTP), οι αφίξεις πακέτων δεν μοντελοποιούνται 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 χρονιστές. Η περιοδικότητα της κίνησης IP (Internet Packets) [23] 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 [24]).

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 χρησιμοποιώντας 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 μία 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**, [16].

Το κυρίως σώμα της παρατηρούμενης κατανομής των χρόνων μεταξύ αφίξεων πακέτων σε όλες σχεδόν τις προαναφερθείσες εφαρμογές μοντελοποιείται ικανοποιητικά από την κατανομή **Pareto**, [11]. Επιπρόσθετη 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, [16].

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 μία 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**.

## **2.4 Η κατανομή Pareto**

Η απόδοση των ψηφιακών κινητών δικτύων ολοκληρωμένων υπηρεσιών τρίτης γενιάς, επηρεάζεται προφανώς από τα πρωτόκολλα πολλαπλής προσπέλασης που χρησιμοποιούνται στα υποσυστήματα ασύρματης πρόσβασης, [11]. Αν και υπάρχουν ικανοποιητικά μοντέλα για την διαδικασία άφιξης των πακέτων μιας πηγής φωνής, που χρησιμοποιεί ανιχνευτή περιόδων της δραστηριότητας της [15], για μια πληθώρα εφαρμογών δεδομένων (όπως μεταφορά αρχείων, είσοδο 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, [16], [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 της δυνατότητες των **Συστημάτων Ασύρματης Προσπέλασης - Ολοκληρωμένων Υπηρεσιών** (Service Integrated Radio Access, **SIR**) να διαχειριστούν συνδιασμό 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 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 μια έντονα εκρηκτική διαδικασία αφίξεων πακέτων ισοδυναμεί με σχεδίαση και μελέτη απόδοσης υπό αυστηρές συνθήκες.

Το αναλυτικό μοντέλο άφιξης πακέτων που χρησιμοποιείται στη μελέτη μας [16], 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, περιέχει χαρακτηριστικά της συμπεριφοράς των χρηστών που σχετίζονται με τις διαλογικές εφαρμογές δεδομένων, οι οποίες αναμένεται να βρούν ευρεία εφαρμογή στα μελλοντικά ασύρματα συστήματα κινητής επικοινωνίας, [11].

Ένα πλήρες μοντέλο κίνησης TELNET μπορεί να περιλαμβάνει κατανομές των αφίξεων μηνυμάτων μέσα σε μια χρονική περίοδο και του μεγέθους των μηνυμάτων (σε πακέτα). Η χρονική κλίμακα πάνω στην οποία φυσιολογικά παρατηρούνται οι περίοδοι αφίξεων (της τάξεως των δεκάδων δευτερολέπτων) είναι πολύ μακριά από τον 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 μίας χρονικής περιόδου και εξετάζουμε το πρωτόκολλο με καθορισμένο αριθμό από εγκατεστημένες μονάδες δεδομένων, [11]. Ακολουθώντας το [16], οι χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων υποτίθεται ότι είναι ανεξάρτητες τυχαίες μεταβλητές κατανεμημένες σύμφωνα με την κατανομή Pareto με παραμέτρους  $a$  και  $k$  :

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

Οι παράμετροι σ' αυτή τη κατανομή είναι η παράμετρος τοποθεσίας (location parameter)  $k$ , που συμβολίζει τον ελάχιστο χρόνο μεταξύ αφίξεων δύο διαδοχικών πακέτων και η παράμετρος σχήματος (shape parameter)  $a$ , όπου αν  $a \leq 2$  η κατανομή έχει *άπειρη διασπορά* ενώ για  $a \leq 1$  η κατανομή έχει *άπειρη διασπορά* και *άπειρη μέση τιμή* (δες παράγραφο 2.4.2 αυτής της εργασίας). Η παραπάνω μαθηματική ιδιότητα ευθύνεται για την έντονη εκρηκτικότητα της **Pareto** διαδικασίας αφίξεων πακέτων.

Στον Πίνακα 2.1 φαίνεται ένα ιστόγραμμα των χρόνων μεταξύ αφίξεων δύο διαδοχικών πακέτων όταν είναι κατανεμημένοι σύμφωνα με κατανομή Pareto με παραμέτρους  $a = 1.5$  και  $k = 0.95$ . Ο πίνακας προέκυψε από την προσομοίωση δέκα χιλιάδων δειγμάτων. Όπως φαίνεται από τον πίνακα το 7.55% των δειγμάτων είναι ανάμεσα στο 0.95 και 1 χρονικό τμήμα (slot), ενώ ο κύριος όγκος των δειγμάτων, 59.35%, βρίσκεται από 1 μέχρι 2 slots και το ποσοστό των δειγμάτων όπου οι χρόνοι μεταξύ αφίξεων είναι μεγαλύτεροι από 20 slots είναι της τάξης του 1%. Από το πίνακα αυτό φαίνεται ότι σε μεγάλο ποσοστό εμφανίζεται το πολύ μία άφιξη πακέτου σε κάθε slot. Επίσης, η παρατηρούμενη συγκέντρωση μεγάλου ποσοστού των δειγμάτων στην περιοχή από 0.95 μέχρι 5 slot και η ύπαρξη κάποιου ποσοστού δειγμάτων στα επόμενα διαστήματα δείχνει ότι η κατανομή **Pareto** είναι έντονα εκρηκτική, επειδή εμφανίζει κάποια slots χωρίς πακέτα (δηλαδή, χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων μεγαλύτεροι της μονάδας), και στη συνέχεια εισάγει συνεχόμενα πακέτα στο σύστημα (δηλαδή, εμφανίζει διαδοχικά μικρούς χρόνους μεταξύ αφίξεων). Υπενθυμίζουμε εδώ ότι η χρονική μονάδα, slot, έχει οριστεί σαν τον χρόνο που χρειάζεται για την μετάδοση ενός πακέτου.

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 το σταθμό βάσης (δέκτη)).

Interarrival Times (slots)	Probability (%)
0,95 - 1	7.55
1 - 2	59.35
2 - 5	24.63
5 - 20	7.43
20 - 60	0.78
60 - 500	0.26

Πίνακας 2.1: Η κατανομή Pareto για  $a = 1.5$  και  $k = 0.95$  slots

### 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  $T$  κατανεμημένη σύμφωνα με την κατανομή **Pareto** με παράμετρο σχήματος  $a$  και παράμετρο τοποθεσίας  $k$  έχει την ακόλουθη Αθροιστική Συνάρτηση Πιθανότητας :

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

με αντίστοιχη Συνάρτηση Πυκνότητας Πιθανότητας :

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

Η Δεύτερη Ροπή της τυχαίας μεταβλητής  $T$  δίδεται από τις σχέσεις :

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

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

ενώ η Μέση τιμή της τυχαίας μεταβλητής  $T$  δίδεται από τις σχέσεις :

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

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

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

Η Διασπορά της τυχαίας μεταβλητής  $T$  είναι ίση με :

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

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) [26] και [27]. Μία κατανομή **Pareto** με  $1.05 < a < 1.25$  έχει βρεθεί ότι μοντελοποιεί ικανοποιητικά τον χρόνο CPU που καταναλώνεται για την εκτέλεση μίας τυχαίας διαδικασίας, [29].

Ακολουθώντας το [22], ο ορισμός μίας κατανομής ως *μακριάς ουράς* είναι :

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

Από τον τύπο αυτό φαίνεται ότι για κάποιες σταθερές  $a$  και  $c$ , ο λόγος  $P[T \geq t] / (ct^{-a})$  τείνει στο 1 καθώς  $t \rightarrow \infty$ . Σύμφωνα με τον παραπάνω ορισμό μπορεί να αποδειχθεί ότι η κατανομή **Pareto** είναι κατανομή *μακριάς ουράς*, [30].

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

$$CMEx = E[T - t \mid T \geq t],$$

είναι μία αύξουσα συνάρτηση του  $t$ , [31].

Χρησιμοποιώντας τον δεύτερο ορισμό της κατανομής *μακριάς ουράς*, έστω ότι η τυχαία μεταβλητή  $T$  παριστάνει το χρόνο αναμονής ενός χρήστη σε ένα σύστημα εξυπηρέτησης. Για χρόνους αναμονής κατανεμημένους σύμφωνα με κατανομές

*μικρής 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 *μακριάς 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 συνθήκη 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 [25]. Δηλαδή, για την τυχαία μεταβλητή  $T$  με κατανομή **Pareto** και για  $y \geq t_0$  ισχύει :

$$P[T > y \mid T > t_0] = P[(t_0 / k)T > y] = (t_0 / y).$$

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

## ΚΕΦΑΛΑΙΟ 3

### Ο Αλγόριθμος Two-Cell Stack

Στο κεφάλαιο αυτό θα αναφερθούμε στον αλγόριθμο **Two-Cell Stack** όπως αυτός παρουσιάζεται στα [12] και [34]. Θα δούμε, επίσης, ποιά είναι το βέλτιστο μήκος  $\Delta$  του παραθύρου ενεργοποίησης “ φρέσκων” πακέτων στον χρονικό άξονα

των αφίξεων που χρησιμοποιεί ο αλγόριθμος, όταν οι χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων κατανέμονται σύμφωνα με την κατανομή **Pareto**.

### 3.1 Ο Αλγόριθμος *Two-Cell Stack*

Ο αλγόριθμος **Two-Cell Stack**, που ανήκει στην κατηγορία των αλγορίθμων τυχαίας προσπέλασης, προτάθηκε το 1989 από τους Μιχάλη Πατεράκη, Λεωνίδα Γεωργιάδη και Παπαντώνη-Καζάκου. Ο αλγόριθμος αυτός, για **Poisson** αφίξεις πακέτων παρουσιάζει throughput ίδιο με αυτό του αλγορίθμου που έχει προταθεί από τον Καπετανάκη [3], αλλά με μικρότερες μέσες καθυστερήσεις πακέτων για ρυθμούς αφίξεων μικρότερους από 0.30 [12].

Θεωρούμε ένα σύγχρονο κανάλι (ο άξονας των χρόνων είναι διαιρεμένος σε ίσα χρονικά τμήματα slots), δυαδική ανάδραση καναλιού (C-NC feedback) στο τέλος του κάθε slot, μηδενική καθυστέρηση διάδοσης και απουσία λαθών στο κανάλι. Υποθέτουμε επίσης ότι για τα συγκρουόμενα πακέτα απαιτείται αναμετάδοση. Ο χρόνος μετράται σε slots όπου το slot  $t$  αντιστοιχεί στο χρονικό διάστημα  $[t, t+1)$ , και με  $X(t)$  συμβολίζουμε την ανάδραση του καναλιού στο τέλος του slot  $t$ . Αν  $X(t) = C$  ( $X(t) = NC$ ) στο slot  $t$  είχαμε “σύγκρουση” (“όχι σύγκρουση”), αντίστοιχα. Για το παραπάνω σύστημα υποθέτουμε επίσης, ότι οι όλοι οι χρήστες παρατηρούν την ανάδραση του καναλιού συνεχώς. Αυτό σημαίνει ότι κάθε χρήστης παρατηρεί την ανάδραση του καναλιού ακόμα και όταν δεν έχει πακέτο να μεταδώσει. Οι αλγόριθμοι αυτοί ονομάζονται **αλγόριθμοι συνεχούς αίσθησης της ανάδρασης** (continuous feedback sensing). Οι αλγόριθμοι που απαιτούν από τους χρήστες να παρακολουθούν την ανάδραση του καναλιού μόνο όταν αυτοί είναι ενεργοί (ένας χρήστης είναι ενεργός μόνο όταν έχει πακέτο προς μετάδοση), ονομάζονται **αλγόριθμοι περιορισμένης αίσθησης της ανάδρασης** (limited feedback sensing).

Η λειτουργία του αλγορίθμου **Two-Cell Stack** βελτιστοποιείται με την επιλογή του μήκους  $\Delta$  ενός διαστήματος (παραθύρου). Έστω  $t$  η χρονική στιγμή που αντιστοιχεί στην αρχή ενός slot τέτοιου ώστε για κάποιο  $t_1 < t$  όλα τα πακέτα που είχαν αφιχθεί στο διάστημα  $(0, t_1]$  έχουν μεταδοθεί επιτυχώς από τον αλγόριθμο μέχρι τη χρονική στιγμή  $t$ , και καμιά πληροφορία όσον αφορά τον αριθμό των πακέτων που έχουν αφιχθεί στο διάστημα  $(t_1, t]$  δεν είναι διαθέσιμη στον αλγόριθμο. Η χρονική

στιγμή  $t$  ονομάζεται **Σημείο Επίλυσης Σύγκρουσης** (Collision Resolution Point, CRP), το διάστημα  $(0, t_1]$  ονομάζεται **“επιλυμένο διάστημα”** (“resolved interval”) και το διάστημα  $(t_1, t]$  ονομάζεται **“ανεξέταστο διάστημα”** (“unresolved interval” or “lag at  $t$ ”). Στο slot  $t$  ο αλγόριθμος επιτρέπει να μεταδώσουν εκείνα μόνο τα πακέτα που έχουν αφιχθεί στο διάστημα  $(t_1, t_2]$ , όπου το  $t_2 = \min(t_1 + \Delta, t)$ , και το διάστημα αφίξεων  $(t_1, t_2]$  ονομάζεται **“εξεταζόμενο διάστημα”** (“examined interval”). Αν το διάστημα  $(t_1, t_2]$  περιέχει το πολύ ένα πακέτο ( $X = NC$ ), τότε αυτό επιλύεται στο slot  $t$ . Αν το  $(t_1, t_2]$  περιέχει τουλάχιστον δύο πακέτα ( $X = C$ ), μία σύγκρουση συμβαίνει στο slot  $t$  και η επίλυση της αρχίζει στο slot  $t+1$ . Έως ότου η σύγκρουση που συνέβη στο slot  $t$  επιλυθεί, κανένα πακέτο που έχει αφιχθεί μετά τη χρονική στιγμή  $t_2$  δεν επιτρέπεται να μεταδοθεί (*μη προφανής κανόνας πρώτης μετάδοσης*, δες παράγραφο 1.3.2). Το χρονικό διάστημα από το slot που συνέβη η σύγκρουση έως το slot που όλα τα πακέτα που συμμετείχαν σε αυτή τη σύγκρουση μεταδοθούν επιτυχώς, ονομάζεται **Διάστημα Επίλυσης Σύγκρουσης** (Collision Resolution Interval, CRI). Αν το “εξεταζόμενο διάστημα” περιέχει ένα πακέτο τότε το διάστημα επίλυσης σύγκρουσης ολοκληρώνεται μέσα σε ένα slot.

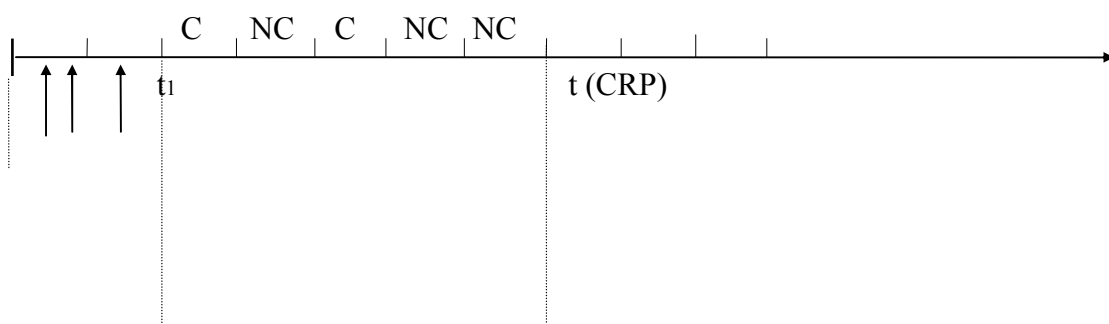
Κατά την διάρκεια των διαστημάτων επίλυσης σύγκρουσης που ξεκινούν με ένα slot στο οποίο υπάρχει σύγκρουση, κάθε χρήστης δρα ανεξάρτητα χρησιμοποιώντας ένα μετρητή (counter) του οποίου η τιμή τη χρονική στιγμή  $t$  συμβολίζεται με  $r(t)$ . Όταν ένας χρήστης μεταδίδει για πρώτη φορά θέτει  $r(t) = 1$ . Ο μετρητής μπορεί να πάρει τις τιμές 1 ή 2, και χρησιμοποιείται, ενημερώνεται σύμφωνα με τους παρακάτω κανόνες :

- 1) Ένας χρήστης μεταδίδει στο slot  $t$  αν και μόνο αν  $r(t) = 1$ . Ένα πακέτο έχει μεταδοθεί επιτυχώς στο slot  $t$  αν και μόνο αν  $r(t) = 1$  και  $X(t) = NC$
- 2) Οι τιμές των μετρητών μεταβάλλονται με το χρόνο ως εξής :

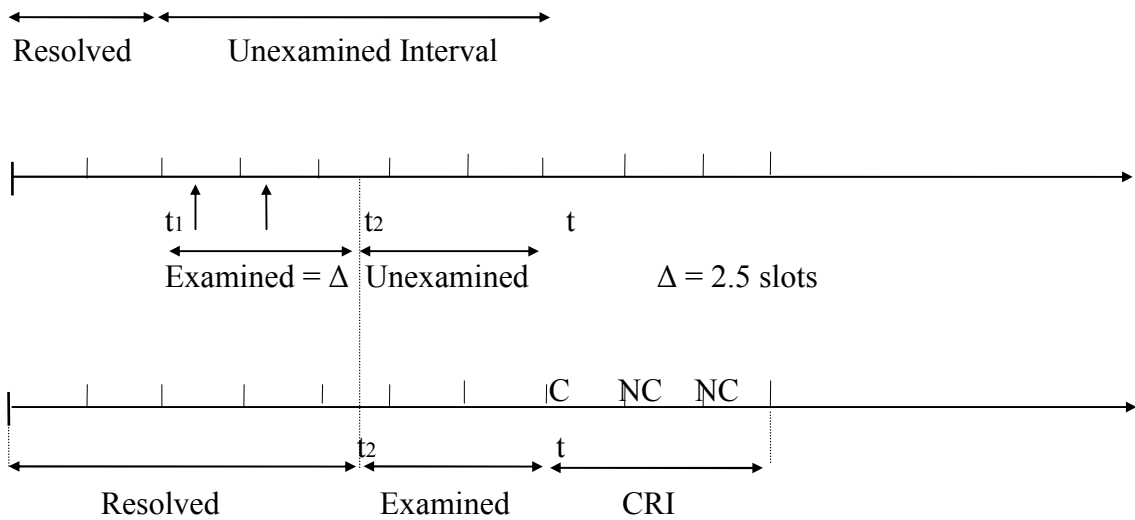
α) αν  $X(t-1) = NC$  και  $r(t-1) = 2$ , τότε  $r(t) = 1$

β) αν  $X(t-1) = C$  και  $r(t-2) = 2$ , τότε  $r(t) = 2$

γ) αν  $X(t-1) = C$  και  $r(t-1) = 1$ , τότε  $r(t) = 1$  ή 2 με πιθανότητα 0.5



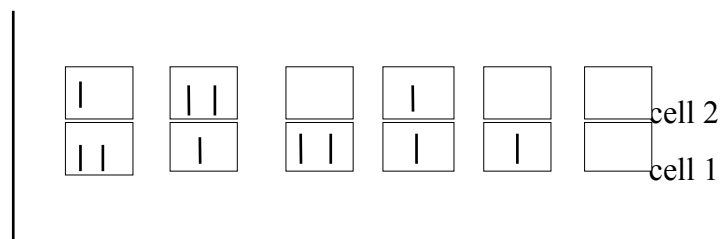


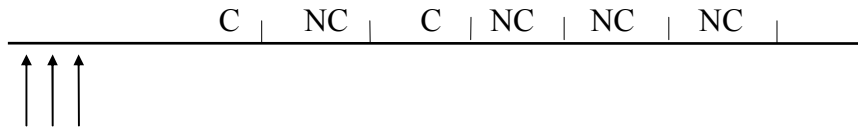


**Σχήμα 3.1 :** Η λειτουργία του αλγορίθμου Two-Cell Stack.

Από τους παραπάνω κανόνες γίνεται προφανές ότι ένα διάστημα επίλυσης σύγκρουσης που ξεκινά με ένα slot στο οποίο υπάρχει σύγκρουση τελειώνει με δύο διαδοχικά slots στα οποία δεν υπάρχει σύγκρουση. Επιπλέον δύο διαδοχικά slots χωρίς σύγκρουση δεν μπορούν να συμβούν πριν το τέλος ενός διαστήματος επίλυσης σύγκρουσης. Έτσι η ύπαρξη δύο διαδοχικών slots χωρίς σύγκρουση κάνει γνωστό σε όλους τους χρήστες που παρατηρούν την ανάδραση του καναλιού ότι το διάστημα επίλυσης σύγκρουσης τελείωσε.

Όπως φαίνεται στο **Σχήμα 3.2**, η λειτουργία του αλγορίθμου μπορεί να περιγραφεί από μια στοίβα με δύο κελιά ( cells), όπου κάθε χρονική στιγμή  $t$ , το κελί 1 περιέχει τους χρήστες που έχουν δικαίωμα να μεταδώσουν (αυτούς με  $r(t) = 1$ ), και το κελί 2 περιέχει τους χρήστες που προσωρινά έχουν εκχωρήσει το δικαίωμα τους να μεταδώσουν (αυτούς με  $r(t) = 2$ ). Ο αλγόριθμος συσσωρεύει όλους τους χρήστες που εκχωρούν το δικαίωμα τους να μεταδώσουν σε ένα κελί (κελί 2).





**Σχήμα 3.2 :** Απεικόνιση της λειτουργίας του αλγορίθμου Two-Cell Stack ως μία στοίβα δύο κελιών.

### 3.2 Επιλογή του παραθύρου βελτίστου μήκους $\Delta$

Όπως προκύπτει από την ανάλυση στο [12] (μερικά από τα αποτελέσματα της ανάλυσης αυτής φαίνονται στον πίνακα 3.1) , όταν οι αφίξεις “φρέσκων” πακέτων χαρακτηρίζονται από την κατανομή **Poisson** η βέλτιστη τιμή του  $\Delta$  η οποία μεγιστοποιεί το throughput του αλγορίθμου είναι 2.33 slots και το αντίστοιχο throughput 0.429.

Η επιλογή του βελτίστου  $\Delta$  για την περίπτωση που οι χρόνοι μεταξύ αφίξεων χαρακτηρίζονται από την έντονα εκρηκτική κατανομή **Pareto** δεν έχει εξεταστεί μέχρι τώρα. Η βέλτιστη επιλογή του  $\Delta$  για **Pareto** κατανεμημένους χρόνους μεταξύ αφίξεων διαδοχικών πακέτων επετεύχθει σε αυτή την εργασία με τη μέθοδο της προσομοίωσης, ο αντίστοιχος κώδικας παρουσιάζεται στο **Παράρτημα 1**. Η προσομοίωση αυτή εκτελέστηκε σε UNIX σταθμό εργασίας SUN SPARC station 5. Σε κάθε εκτέλεση της προσομοίωσης (run) προσομοιώθηκε η μετάδοση ενός εκατομμυρίου πακέτων. Ο αριθμός αυτός επελέγει ώστε η συμπεριφορά του αλγορίθμου να έχει φτάσει σε “μόνιμη κατάσταση” (steady-state). Σε πολλές περιπτώσεις προσομοιώσαμε λιγότερα και περισσότερα πακέτα (500,000 και 2,000,000) και παρατηρήσαμε ότι σε όλες τις περιπτώσεις αυτές η προσομοίωση ενός εκατομμυρίου πακέτων εκτιμούσε κατά ικανοποιητικό τρόπο τις μετρικές απόδοσης (μέση καθυστέρηση πακέτων και μήκος ανεξέταστου διαστήματος (lag)) του αλγορίθμου σε “μόνιμη κατάσταση”.

#### 3.2.1 Αποτελέσματα της προσομοίωσης του αλγορίθμου *Two-Cell Stack*

Αρχικά, για να βεβαιωθούμε ότι ο προσομοιωτής μας δουλεύει σωστά, τρέξαμε την προσομοίωση με τις αφίξεις των πακέτων να χαρακτηρίζονται από την κατανομή

**Poisson.** Τα αποτελέσματα της προσομοίωσης αυτής, καθώς και τα αποτελέσματα από την ακριβή ανάλυση του ίδιου συστήματος η οποία παρουσιάζεται στο [12], φαίνονται στον **Πίνακα 3.1**. Από τα αποτελέσματα αυτά παρατηρούμε ότι για ρυθμούς αφίξεων πακέτων έως και 0.36 η προσομοίωση μας δίνει καθυστερήσεις που βρίσκονται μέσα στα όρια που έχουν προκύψει από την ανάλυση. Για ρυθμούς μεγαλύτερους από 0.36 παρατηρούμε μία μικρή απόκλιση. Το γεγονός αυτό όμως είναι αναμενόμενο αφού όσο ο ρυθμός αφίξεων πακέτων πλησιάζει το throughput του αλγορίθμου, είναι αρκετά δύσκολο να εκτιμήσουμε με ακρίβεια μετρικές απόδοσης (όπως η μέση καθυστέρηση πακέτου) μέσω προσομοίωσης.

Στη συνέχεια θα παρουσιάσουμε τα αποτελέσματα της προσομοίωσης, με βάση τα οποία θα γίνει η επιλογή του κατάλληλου  $\Delta$  για τον αλγόριθμο Two-Cell Stack όταν οι χρόνοι μεταξύ αφίξεων διαδοχικών πακέτων χαρακτηρίζονται από την κατανομή **Pareto**. Ζητάμε την κατάλληλη τιμή του  $\Delta$  η οποία να βελτιστοποιεί την λειτουργία του αλγορίθμου για ένα ευρύ σύνολο τιμών της παραμέτρου  $k$  της κατανομής **Pareto**.

Στον **Πίνακα 3.2** παραθέτουμε τα αποτελέσματα της προσομοίωσης όταν η παράμετρος  $k$  πέρνει την τιμή 0.7. Τα αποτελέσματα αυτά συμπεριλαμβάνουν την μέση καθυστέρηση πακέτου  $E(D)$  και το μήκος του ανεξέταστου διαστήματος (lag) σε slots μετά το τέλος της προσομοίωσης.

Packet Arrival Rates	Mean Delay Analytical Results		Simoulation Results Mean Delay
	Lower Bound	Upper Bound	
0.02	1.562	1.563	1.562
0.06	1.708	1.716	1.712
0.10	1.888	1.917	1.893
0.16	2.257	2.363	2.280
0.24	3.103	3.467	3.225
0.30	4.412	5.197	4.868
0.32	5.162	6.170	5.870
0.36	7.941	9.665	9.630
0.38	11.008	13.398	13.960

0.40	18.262	22.024	24.430
------	--------	--------	--------

**Πίνακας 3.1 :** Η μέση καθυστέρηση πακέτου για διάφορους ρυθμούς αφίξεων (Poisson), όπως προκύπτει από την προσομοίωση, καθώς και τα άνω και κάτω όρια της ίδιας μέσης καθυστέρησης όπως προκύπτουν από την ανάλυση του συστήματος, [12].

		$\Delta = 1.8$		$\Delta = 1.6$		$\Delta = 1.4$	
a	$\lambda$	E(D)	lag	E(D)	lag	E(D)	lag
1.2	0.238	12.12	1	10.51	1	10.88	16
1.3	0.329	40.71	43	31.47	1	32.9	22
1.34	0.362	101	76	69.68	5	71.77	39
1.36	0.378	187.2	281	131	25	155	94
1.37	0.385	316.2	420	172	53	269	209
1.38	0.393	570.6	706	419	319	367	222
1.39	0.40	1176	6364	534	331	2039	450
1.4	0.408	5796	12933	2122	1180	6463	7835

**Πίνακας 3.2 :** Η μέση καθυστέρηση πακέτου (σε slots) και το μήκος του ανεξέταστου διαστήματος (σε slots), για διάφορες τιμές του ρυθμού αφίξεων πακέτων  $\lambda$  (packets/slot) όταν  $k = 0.7$  (παράμετρος  $k$  της κατανομής Pareto).

Από τον **Πίνακα 3.2** παρατηρούμε ότι τόσο η μέση καθυστέρηση πακέτου όσο και το μήκος του ανεξέταστου διαστήματος ελαχιστοποιούνται όταν το  $\Delta$  παίρνει την τιμή 1.6 slots. Όπως θα δούμε και από τα άλλα αποτελέσματα της προσομοίωσης η τιμή αυτή δεν είναι απόλυτη αλλά αποτελεί το κεντρικό σημείο μίας στενής περιοχής τιμών του  $\Delta$  οι οποίες βελτιστοποιούν την λειτουργία του αλγορίθμου Two-Cell Stack για όλες τις τιμές της παραμέτρου  $k$  της κατανομής **Pareto** που μας ενδιαφέρουν. Έτσι, όπως φαίνεται από τον **Πίνακα 3.3** όταν η παράμετρος  $k$  πάρει την τιμή 0.6 η βέλτιστη τιμή του  $\Delta$  παραμένει στο 1.6 slots αλλά με τις τιμές της μέσης καθυστέρησης πακέτου να μην διαφέρουν σημαντικά από τις αντίστοιχες τιμές για  $\Delta = 1.8$  slots. Το γεγονός αυτό θα μπορούσε να μας επιτρέψει να θεωρήσουμε σαν βέλτιστη τιμή του  $\Delta$  τα 1.7 ή 1.8 slots, αν αυτό που ζητούσαμε ήταν η τιμή του  $\Delta$  που

θα βελτιστοποιούσε τον αλγόριθμο Two-Cell Stack όταν η παράμετρος  $k$  της κατανομής **Pareto** παίρνει την τιμή 0.6.

		$\Delta = 1.8$		$\Delta = 1.6$		$\Delta = 1.4$	
a	$\lambda$	E(D)	lag	E(D)	lag	E(D)	lag
1.05	0.079	16.97	14	16.43	9	18.89	30
1.1	0.151	30.48	100	28.8	37	35.58	55
1.15	0.217	63	129	61.52	150	93.15	106.6
1.17	0.242	99.89	222	94.80	218	166.83	130
1.2	0.277	225.13	404.5	259.33	624	636.74	1833
1.22	0.30	591.52	238.5	559.5	325	1942	4424
1.25	0.333	2804	4456	2402	5696	35304	67636

**Πίνακας 3.3 :** Η μέση καθυστέρηση πακέτου (σε slots) και το μήκος του ανεξέταστου διαστήματος (σε slots), για διάφορες τιμές του ρυθμού αφίξεων πακέτων  $\lambda$  (packets/slot) όταν  $k = 0.6$ .

Παρόμοια συμπεράσματα μπορούμε να βγάλουμε από τα αποτελέσματα της προσομοίωσης όταν  $k = 0.8$  τα οποία φαίνονται στον **Πίνακα 3.4**. Σε αυτή την περίπτωση παρατηρούμε ότι η βέλτιστη τιμή του  $\Delta$  είναι 1.4 slots. Όμως οι τιμές της μέσης καθυστέρησης πακέτου για  $\Delta = 1.6$  και 1.4 slots δεν διαφέρουν σημαντικά όταν ο ρυθμός άφιξης πακέτων δεν ξεπερνά τα 0.4 πακέτα ανά slot.

Από την παραπάνω συζήτηση (και όπως θα δούμε και από τα αποτελέσματα που παρουσιάζονται στο Κεφάλαιο 5), γίνεται φανερό ότι η βέλτιστη τιμή του  $\Delta$  κάτω από τις προϋποθέσεις που θέσαμε (ζητάμε εκείνη την τιμή του  $\Delta$  που φέρνει την λειτουργία του αλγορίθμου όσο το δυνατό πλησιέστερα στη βέλτιστη για όλες τις τιμές της παραμέτρου  $k$  της κατανομής **Pareto** του χρόνου μεταξύ αφίξεων διαδοχικών πακέτων που μας ενδιαφέρουν), είναι τα 1.6 slots.

		$\Delta = 1.8$		$\Delta = 1.6$		$\Delta = 1.4$	
a	$\lambda$	E(D)	lag	E(D)	lag	E(D)	lag

1.4	0.357	7.76	1	5.38	1	5.24	1
1.5	0.416	23	15.54	10.4	1	9.4	2.5
1.55	0.433	76.6	96.8	19.65	8.5	17.1	10
1.6	0.468	2639	966	45.44	25.5	74.73	12
1.64	0.487	30067	57350	301	141.5	106.9	169
1.645	0.490	-	-	386	200	132.67	190
1.65	0.492	-	-	749	249	336	453
1.66	0.496	-	-	4196	5678	1597	679

**Πίνακας 3.4 :** Η μέση καθυστέρηση πακέτου (σε slots) και το μήκος του ανεξέταστου διαστήματος (σε slots), για διάφορες τιμές του ρυθμού αφίξεων πακέτων  $\lambda$  (packets/slot) όταν  $k = 0.8$ .

## ΚΕΦΑΛΑΙΟ 4

### Μετρικές απόδοσης του συστήματος Ολοκληρωμένης Προσπέλασης για Poisson αφίξεις μηνυμάτων δεδομένων

Στο κεφάλαιο αυτό παραθέτουμε τα αποτελέσματα της προσομοίωσης του συστήματος για την **Ολοκληρωμένη Τυχαία Προσπέλαση Πακέτων Φωνής και Δεδομένων** το οποίο συζητήθηκε αναλυτικά στο Κεφάλαιο 2. Στο τμήμα αυτό της εργασίας οι αφίξεις “φρέσκων” μηνυμάτων δεδομένων χαρακτηρίζονται από την κατανομή **Poisson**, ενώ για τον αριθμό των πακέτων ανά μήνυμα θα εξετάσουμε δύο περιπτώσεις. Στην πρώτη περίπτωση, ο αριθμός των πακέτων δεδομένων ανά μήνυμα

είναι γεωμετρικά κατανομημένη τυχαία μεταβλητή, ενώ στη δεύτερη περίπτωση, ο αριθμός αυτός είναι σταθερός.

#### **4.1 Καθορισμός των παραμέτρων του συστήματος Ολοκληρωμένης Τυχαίας Προσπέλασης Πακέτων Φωνής και Δεδομένων**

Στην προσομοίωση του συστήματος Ολοκληρωμένης Τυχαίας Προσπέλασης Πακέτων Φωνής και Δεδομένων θα χρησιμοποιήσουμε τις παραμέτρους που περιέχονται στον **Πίνακα 4.1**. Ο ρυθμός μετάδοσης του καναλιού είναι από τα [32,33]. Ο ρυθμός ομιλίας προϋποθέτει διαμόρφωση **ADPCM** (Adaptive Differential Pulse Code Modulation), [28]. Το μέγεθος του πακέτου (53 bytes) επελέγει για συμβατότητα με το **ATM** (Asynchronous Transfer Mode) πακέτο (cell). Οι τιμές για την μέση διάρκεια της περιόδου ομιλίας/ησυχίας δίνουν την πληροφορία ότι ένας χρήστης φωνής είναι ενεργός (βρίσκεται σε περίοδο ομιλίας) το 44% του χρόνου, [32]. Η διάρκεια του frame είναι 12ms και χωρίζεται σε 51 slots. Η μέγιστη επιτρεπτή καθυστέρηση για ένα πακέτο φωνής είναι 24ms, δηλαδή 2 frames. Λαμβάνοντας υπόψη θέματα συγχρονισμού του συστήματος, θεωρούμε ότι σε κάθε frame είναι διαθέσιμα 50 slots. Το πρώτο και το δεύτερο slot από τα 50 διαθέσιμα σε κάθε frame αποτελούν τα διαστήματα αιτήσεων για τη φωνή και τα δεδομένα, αντίστοιχα, και το κάθε ένα από αυτά υποδιαιρείται σε mini-slots. Τα υπόλοιπα 48 slots αποτελούν το διάστημα πληροφορίας. Επιλέγουμε 6 mini-slots ανά διάστημα αίτησης (ώστε σε κάθε ένα από αυτά να μπορούν να μεταδοθούν περίπου 70 bits) .

Speech rate	32 Kb/s
Channel rate	1.2 Mb/s
Packet size (bits)	424
Speech/Data (bits)	384
Header (bits)	40
Frame duration (ms)	12
Slots per frame	50
Voice delay limit (ms)	24
Mean talkspurt duration (s)	1.41
Mean silence duration (s)	1.78

**Πίνακας 4.1 :** Οι παράμετροι του συστήματος Ολοκληρωμένης Τυχαίας Προσπέλασης Πακέτων φωνής και δεδομένων.

---

#### 4.2 Αποτελέσματα Προσομοίωσης

Η προσομοίωση του συστήματος εκτελέστηκε σε UNIX σταθμό εργασίας **SUN SPARCstation 5**. Τα αποτελέσματα βασίζονται σε 10 ανεξάρτητες εκτελέσεις της προσομοίωσης των 305,000 frames και σε πολλές περιπτώσεις των 605,000 frames για κάθε εκτέλεση. Άλλες φορές πάλι η προσομοίωση εκτελέστηκε μία φορά αλλά για πολύ μεγάλο χρόνο (2,000,000 ή 3,000,000 frames). Σε όλες αυτές τις περιπτώσεις η προσπάθεια μας ήταν να έχει φτάσει το σύστημα σε μόνιμη κατάσταση (steady-state). Έτσι, για να ελαχιστοποιήσουμε το φαινόμενο της έναρξης (start-up), τα πρώτα 5000 frames θεωρούνται ως περίοδος προθέρμανσης της προσομοίωσης. Σε κάθε προσομοίωση, ο αριθμός των χρηστών φωνής στο σύστημα παραμένει σταθερός, οι χρήστες φωνής αρχικά βρίσκονται σε κατάσταση ησυχίας και στην περίπτωση των 10 ανεξάρτητων εκτελέσεων τα τελικά αποτελέσματα αποτελούν την μέση τιμή των αποτελεσμάτων των εκτελέσεων αυτών.

Υποθέτουμε ότι τα μηνύματα πληροφορίας γεννιούνται από ένα μεγάλο άγνωστο αριθμό χρηστών δεδομένων (θεωρητικά άπειρο) το μέγιστο του οποίου καθορίσαμε στους 500 χρήστες και ο συνολικός αριθμός αφίξεων πακέτων δεδομένων περιγράφεται από την κατανομή **Poisson** με μέση τιμή  $\lambda$  (messages per frame). Επιπλέον, υποθέτουμε ότι ο αριθμός των πακέτων ανά μήνυμα είναι είτε γεωμετρικά κατανομημένος με παράμετρο  $q$  και μέση τιμή  $B = 1/q$  όπου  $B = 8$ , είτε σταθερός και ίσος με 8 πακέτα ανά μήνυμα ( $M = 8$ ).

Πρίν προχωρήσουμε στην παρουσίαση των αποτελεσμάτων, θεωρούμε χρήσιμη μια σύντομη περιγραφή των μετρικών απόδοσης που θα χρησιμοποιηθούν. Έτσι, με τον όρο καθυστέρηση προσπέλασης (wait or access delay) ορίζεται το χρονικό διάστημα ανάμεσα στην άφιξη του μηνύματος και το τέλος της επιτυχημένης μετάδοσης του πρώτου πακέτου του μηνύματος αυτού σε ένα δεσμευμένο slot. Η καθυστέρηση μηνύματος (message delay) ορίζεται ως το χρονικό διάστημα ανάμεσα στην άφιξη του μηνύματος και το τέλος της επιτυχημένης μετάδοσης του τελευταίου πακέτου του μηνύματος σε ένα δεσμευμένο slot. Τέλος, το throughput ορίζεται ως το ποσοστό των slots που επιτυχώς μεταφέρουν πακέτα πληροφορίας δεδομένων.



Στους **Πίνακες 4.2-4.3** παραθέτουμε τα αποτελέσματα της προσομοίωσης για τα δεδομένα όταν στο σύστημα υπάρχουν 86, 90 και 95 ενεργοί χρήστες φωνής. Οι συγκεκριμένες αυτές τιμές δεν είναι τυχαίες αλλά αποτελούν μία κρίσιμη περιοχή τιμών για τους χρήστες φωνής. Έτσι, όπως προκύπτει από το **[10]** όταν στο σύστημα υπάρχουν μέχρι 86 ενεργοί χρήστες φωνής ( $N = 86$ ) από τη μία το ποσοστό των πακέτων φωνής που απορρίπτονται (λόγω της υπέρβασης της μέγιστης επιτρεπτής καθυστέρησης τους) είναι σχεδόν μηδενικό και από την άλλη η καθυστέρηση προσπέλασης για τα πακέτα φωνής διατηρείται σε χαμηλά επίπεδα (περίπου 15ms). Ενώ όταν ο αριθμός των ενεργών χρηστών φωνής στο σύστημα υπερβεί τους 95 ( $N > 95$ ) τόσο το ποσοστό των πακέτων που απορρίπτονται όσο και η καθυστέρηση πρόσβασης για τα πακέτα φωνής αυξάνονται απότομα.

Όπως προκύπτει από τα αποτελέσματα του **Πίνακα 4.2**, όταν ο συνολικός ρυθμός αφίξεων για τα δεδομένα δεν υπερβαίνει τα 0.50 μηνύματα / frame η καθυστέρηση προσπέλασης για τα πακέτα δεδομένων είναι αρκετά μικρή, κάτω από το επίπεδο των 100ms, ενώ όταν ο ρυθμός άφιξης μηνυμάτων δεδομένων δεν υπερβαίνει το 0.75 μηνύματα / frame η αντίστοιχη καθυστέρηση βρίσκεται κάτω από το επίπεδο των 200ms. Όταν όμως ο ρυθμός αφίξεων δεδομένων υπερβεί το 0.75 μηνύματα / frame η καθυστέρηση προσπέλασης για τα πακέτα δεδομένων αυξάνεται απότομα με αποτέλεσμα ρυθμοί αφίξεων μεγαλύτεροι από 1.00 μηνύματα / frame να είναι απαγορευτικοί για το σύστημα. Επίσης, μπορούμε να παρατηρήσουμε την πολύ μικρή διαφορά στην καθυστέρηση προσπέλασης όταν ο αριθμός πακέτων ανά μήνυμα είναι γεωμετρικά κατανομημένος ( $B = 1/q$ ,  $B = 8$ ) και όταν ο αντίστοιχος αριθμός είναι σταθερός ( $M = 8$ ).

Τα αποτελέσματα για την καθυστέρηση μηνύματος και το throughput είναι τα αναμενόμενα θεωρητικά. Η καθυστέρηση μηνύματος μπορεί να προκύψει αν στην καθυστέρηση προσπέλασης προσθέσουμε το χρόνο που απαιτείται για να μεταδωθούν τα πακέτα του μηνύματος, εκτός το πρώτο του οποίου η καθυστέρηση συμπεριλαμβάνεται στην καθυστέρηση προσπέλασης. Έτσι, αν υποθέσουμε ότι από τη στιγμή που ένας χρήστης με  $m$  πακέτα δεδομένων κάνει μία κράτηση μεταδίδει ένα πακέτο σε κάθε ένα από τα επόμενα frame (στο αντίστοιχο δεσμευμένο γι' αυτόν slot), τότε η καθυστέρηση μηνύματος θα είναι :

$$\text{Message Delay} = \text{Wait Delay} + (m-1) * \text{Frame Duration}$$

Για παράδειγμα, για ρυθμό άφιξης δεδομένων 0.50,  $m=8$  και  $\text{Frame Duration}=12\text{ms}$ , η καθυστέρηση προσπέλασης είναι 81.83 ms τότε η καθυστέρηση μηνύματος θα είναι περίπου  $81.83\text{ms} + 7 * 12\text{ms} = 165.68 \text{ ms}$ . Όπως φαίνεται από την δεύτερη γραμμή του **Πίνακα 4.2** η τιμή αυτή είναι πολύ κοντά στην αντίστοιχη τιμή που μας δίνει η προσομοίωση.

Το θεωρητικά αναμενόμενο throughput δίνεται από τη σχέση :

$$\text{Throughput} = \lambda * (\text{number of packets per message})$$

Έτσι για ένα ρυθμό άφιξης δεδομένων 0.50 μηνύματα ανά frame και σταθερό αριθμό πακέτων ανά μήνυμα ίσο με 8, το throughput είναι 4.00 πακέτα ανά frame. Όπως φαίνεται και από τη δεύτερη γραμμή του **Πίνακα 4.2** το ίδιο ακριβώς αποτέλεσμα μας δίνει και η προσομοίωση.

Στους **Πίνακες 4.3, 4.4** παραθέτουμε τα αποτελέσματα της προσομοίωσης όταν ο αριθμός των ενεργών χρηστών φωνής στο σύστημα αυξηθεί σε 90 και 95, αντίστοιχα ( $N = 95, 90$ ). Έτσι, παρατηρούμε ότι όσο ο αριθμός αυτός αυξάνεται τόσο οι καθυστερήσεις για τα δεδομένα αυξάνονται, πράγμα αναμενόμενο αφού μειώνεται ο αριθμός των διαθέσιμων για τα δεδομένα slots πληροφορίας σε κάθε frame (να υπενθυμίσουμε εδώ την απόλυτη προτεραιότητα της φωνής έναντι των δεδομένων). Ειδικά για την περίπτωση που στο σύστημα υπάρχουν 95 ενεργοί χρήστες φωνής η καθυστέρηση προσπέλασης (άρα και η καθυστέρηση μηνύματος) παίρνει πολύ μεγάλες τιμές ακόμα και για πολύ μικρές του ρυθμού άφιξης μηνυμάτων δεδομένων. Έτσι, για την περίπτωση που ο ρυθμός άφιξης των δεδομένων είναι 0.50 μηνύματα / frame και ο αριθμός των ενεργών χρηστών φωνής στο σύστημα αυξηθεί από τους 86 στους 90, η καθυστέρηση προσπέλασης για τα πακέτα δεδομένων υπερδιπλασιάζεται, ενώ με παραπέρα αύξηση στους 95 χρήστες η αντίστοιχη καθυστέρηση γίνεται εννέα φορές μεγαλύτερη από αυτήν για την περίπτωση των 86 ενεργών χρηστών φωνής.

Arrivals (msg/fr)	Wait Mean Delay (ms)	Message Mean Delay (ms)	Throughput (packets/frame)
----------------------	-------------------------	----------------------------	-------------------------------

	B = 8 (B = 1/q)	M = 8 (constant)	B = 8 (B = 1/q)	M = 8 (constant)	B = 8 (B = 1/q)	M = 8 (constant)
0.25	42.45	42.96	128.83	129.55	2.00	2.00
0.50	81.83	78.65	168.11	165.16	3.99	4.00
0.75	180.46	178.07	266.82	264.69	5.99	6.00
1.00	593.27	571.75	679.601	658.67	8.00	8.00

**Πίνακας 4.2 :** Οι μετρικές απόδοσης του αλγορίθμου Two-Cell Stack για τα δεδομένα, όταν στο σύστημα υπάρχουν 86 ενεργοί χρήστες φωνής.

Arrivals (msg/fr)	Wait Mean Delay (ms)		Message Mean Delay (ms)		Throughput (packets/frame)	
	B = 8 (B = 1/q)	M = 8 (constant)	B = 8 (B = 1/q)	M = 8 (constant)	B = 8 (B = 1/q)	M = 8 (constant)
0.25	79.75	76.02	169.62	166.08	2.00	2.00
0.50	190.51	190.08	279.62	279.44	4.00	4.00
0.75	582.53	524.64	671.14	613.74	5.99	6.00
1.00	2474.12	2443.71	2562.21	2532.78	7.83	8.00

**Πίνακας 4.3 :** Οι μετρικές απόδοσης του αλγορίθμου Two-Cell Stack για τα δεδομένα, όταν στο σύστημα υπάρχουν 90 ενεργοί χρήστες φωνής.

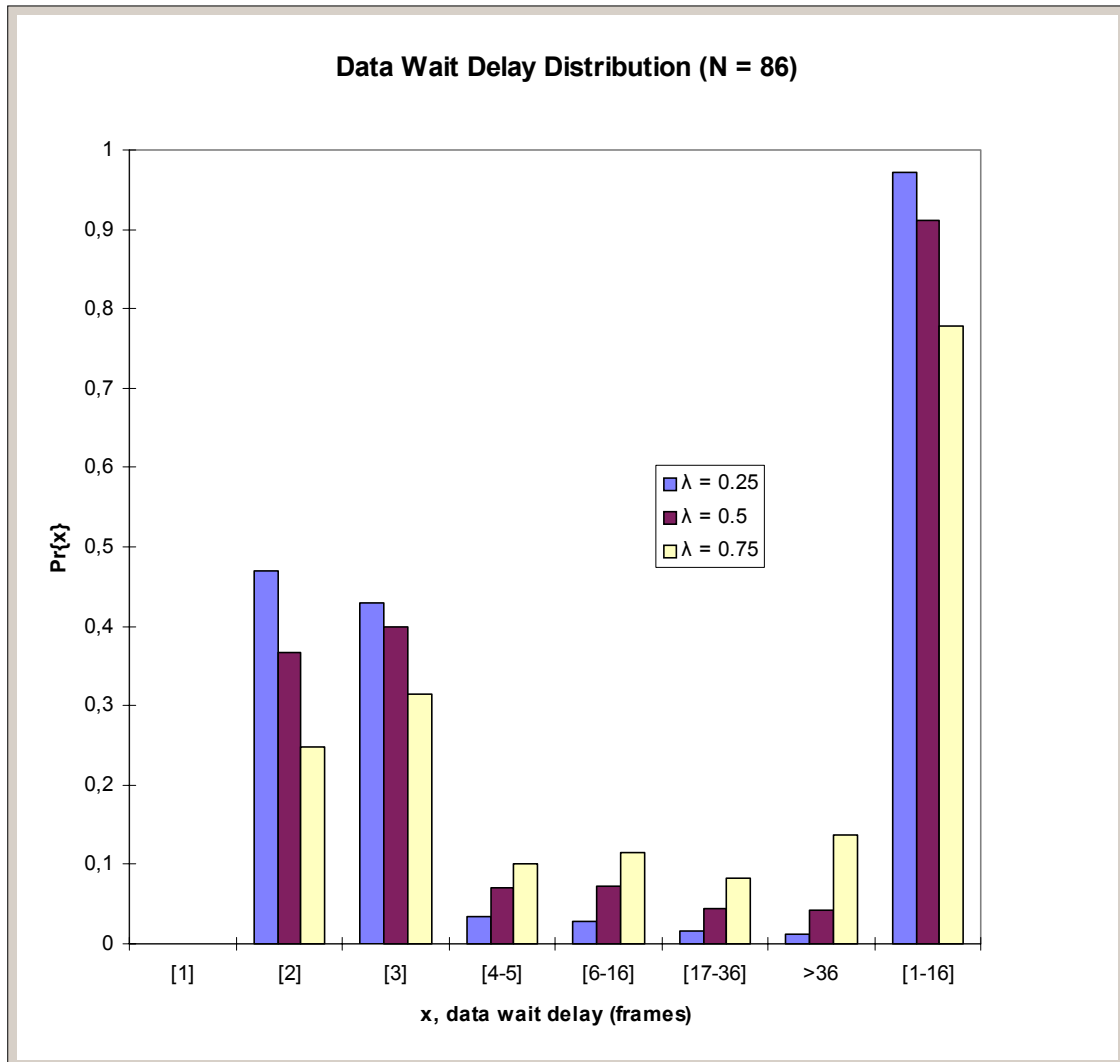
Arrivals (msg/fr)	Wait Mean Delay (ms)		Message Mean Delay (ms)		Throughput (packets/frame)	
	B = 8 (B = 1/q)	M = 8 (constant)	B = 8 (B = 1/q)	M = 8 (constant)	B = 8 (B = 1/q)	M = 8 (constant)
0.10	105.57	108.19	206.49	210.15	0.80	0.80
0.20	174.70	171.50	274.11	271.05	1.60	1.60
0.25	208.48	225.82	306.62	325.00	1.99	2.00
0.50	706.51	701.76	802.05	797.63	4.01	4.00

**Πίνακας 4.4 :** Οι μετρικές απόδοσης του αλγορίθμου Two-Cell Stack για τα δεδομένα, όταν στο σύστημα υπάρχουν 95 ενεργοί χρήστες φωνής.

#### 4.2.1 Κατανομή της Καθυστέρησης Προσπέλασης των Πακέτων Δεδομένων

Επειδή η μέση καθυστέρηση δεν αποτελεί το μοναδικό κριτήριο απόδοσης για ένα σύστημα (ή για ένα αλγόριθμο) αυτό που κάναμε στη συνέχεια ήταν να εκτιμήσουμε μέσω της προσομοίωσης την κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων, καθώς και την αντίστοιχη τυπική απόκλιση. Είναι γνωστό άλλωστε, ότι στους αλγορίθμους Τυχαίας Προσπέλασης γνώση μόνο της μέσης τιμής της καθυστέρησης πακέτων δεν επιτρέπει ασφαλείς προβλέψεις για την καθυστέρηση κάποιου τυχαία επιλεγμένου πακέτου, διότι η κατανομή της καθυστέρησης χαρακτηρίζεται συνήθως από μεγάλη διασπορά, μακριά ουρά και τιμή μέσου (median) αρκετά μεγαλύτερη της μέσης τιμής (mean), [1], [36], [37].

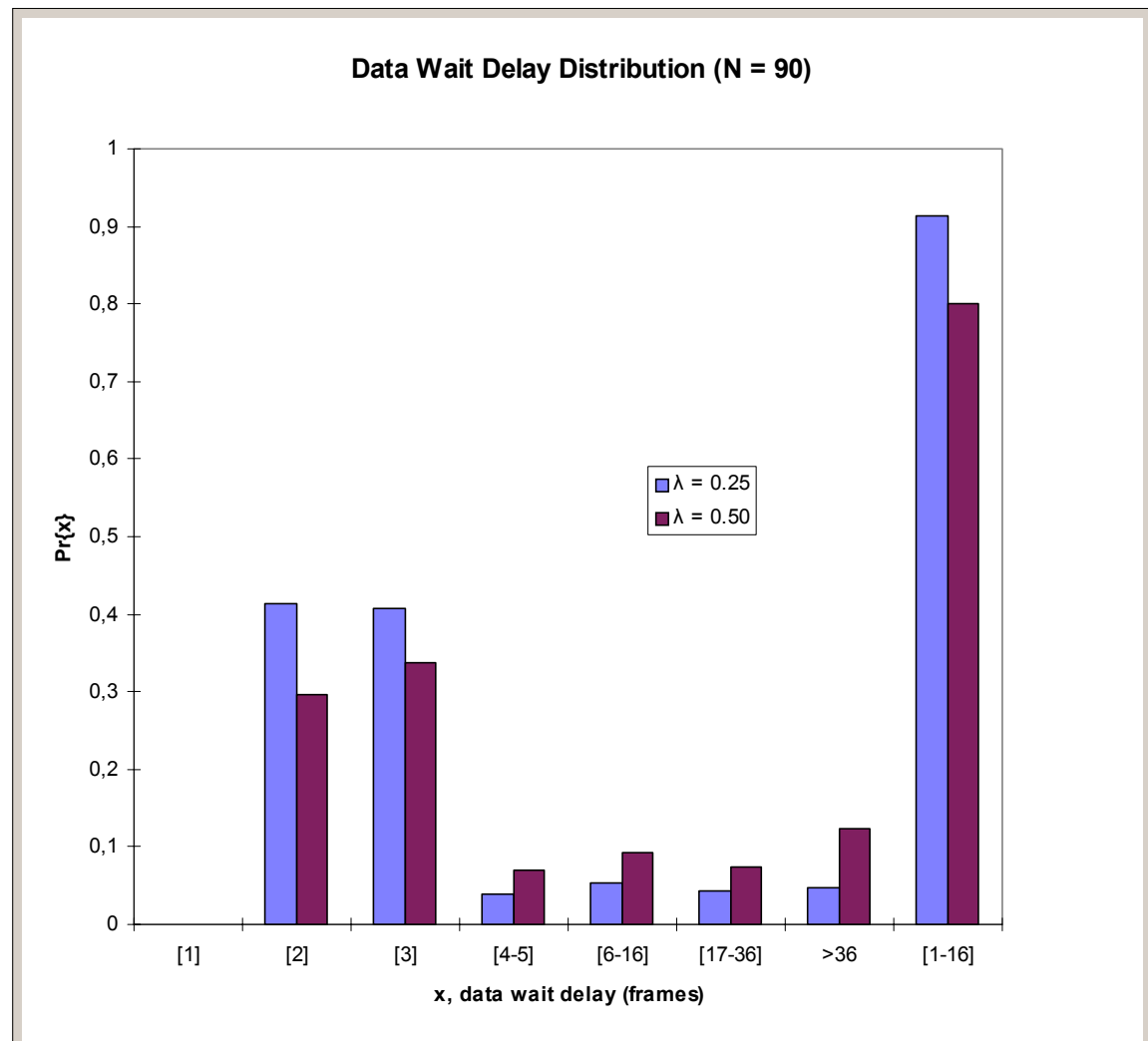
Στο **Σχήμα 4.1** φαίνεται η κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για διάφορους ρυθμούς άφιξης των δεδομένων, όταν στο σύστημα υπάρχουν 86 ενεργοί χρήστες φωνής ( $N = 86$ ). Ο συμβολισμός  $[x-y]$  στον οριζόντιο άξονα, δηλώνει χρονικό διάστημα καθυστέρησης προσπέλασης πακέτων που εκτείνεται από  $x$  έως και  $y$  frames. Όπως μπορούμε να δούμε καθαρά από το σχήμα αυτό, όταν ο ρυθμός άφιξης δεδομένων είναι 0.25 μηνύματα / frame το μεγαλύτερο ποσοστό των πακέτων δεδομένων, το 91% περίπου, έχουν καθυστερήσεις προσπέλασης που δεν ξεπερνούν τα 3 frames (36ms), ενώ το 97% των πακέτων έχει καθυστερήσεις προσπέλασης που δεν υπερβαίνουν τα 16 frames (192ms). Όσο ο ρυθμός άφιξης δεδομένων αυξάνεται παρατηρούμε μία μετατόπιση της μάζας των καθυστερήσεων των πακέτων προς μεγαλύτερες τιμές. Έτσι όταν ο ρυθμός αφίξεων γίνει 0.75 μηνύματα / frame, το ποσοστό των πακέτων για τα οποία η καθυστέρηση προσπέλασης δεν ξεπερνά τα 3 frames μειώνεται στο 56% και το ποσοστό των πακέτων που έχουν καθυστερήσεις προσπέλασης μέχρι 16 frames (192ms) μειώνεται στο 77.8%. Πρέπει να πούμε εδώ ότι τα 16 frames δεν αποτελούν ένα απόλυτο όριο για την καθυστέρηση προσπέλασης των πακέτων δεδομένων, αλλά μία τιμή που εμείς επιλέξαμε στην προσπάθεια μας να έχουμε κάποιο μέτρο σύγκρισης.



**Σχήμα 4.1 :** Κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $N = 86$ .

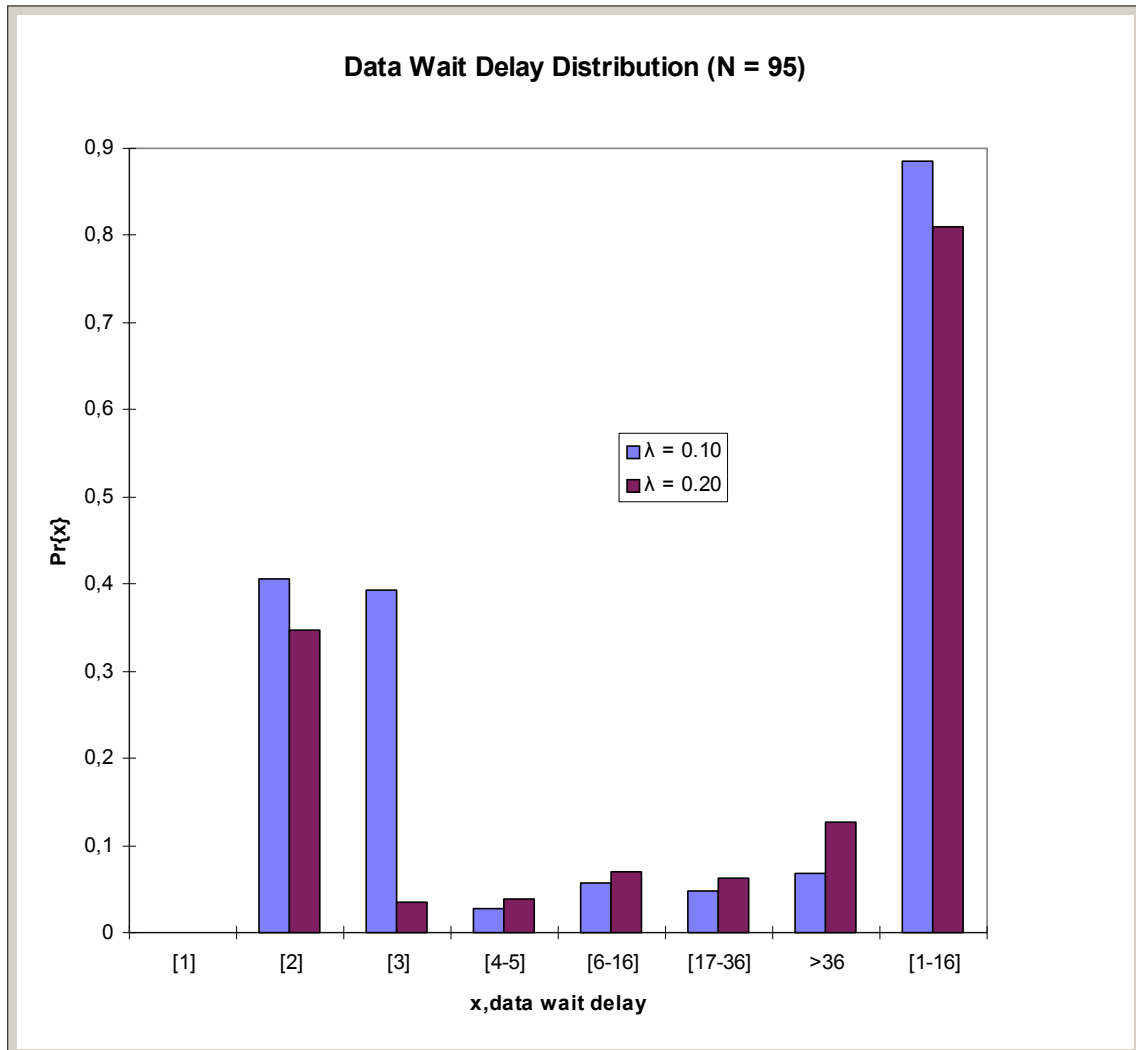
Στα **Σχήματα 4.2, 4.3** παραθέτουμε τα αποτελέσματα της προσομοίωσης για την κατανομή της καθυστέρησης προσπέλασης πακέτων δεδομένων όταν οι ενεργοί χρήστες φωνής στο σύστημα αυξηθούν σε 90 και 95, αντίστοιχα. Όπως είχαμε παρατηρήσει και στην περίπτωση της μέσης καθυστέρησης προσπέλασης των πακέτων δεδομένων όταν “φορτώνουμε” το σύστημα με περισσότερους ενεργούς χρήστες φωνής η κατάσταση, για τα δεδομένα γίνεται όλο και πιο δυσμενής. Έτσι και στην περίπτωση της κατανομής της καθυστέρησης προσπέλασης των πακέτων δεδομένων όταν ο αριθμός των ενεργών χρηστών φωνής αυξηθεί στους 90 χρήστες, και για ρυθμό άφιξης δεδομένων 0.50 μηνύματα / frame το ποσοστό των πακέτων για τα οποία η καθυστέρηση προσπέλασης δεν ξεπερνά τα 3 frames (36ms) είναι 63.45%,

και το ποσοστό πακέτων με καθυστερήσεις προσπέλασης που δεν υπερβαίνουν τα 16 frames (192ms) είναι περίπου 80%, όταν στην περίπτωση των 86 χρηστών φωνής τα ποσοστά αυτά είναι περίπου 78% και 91%, αντίστοιχα.



**Σχήμα 4.2 :** Κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $N = 90$ .

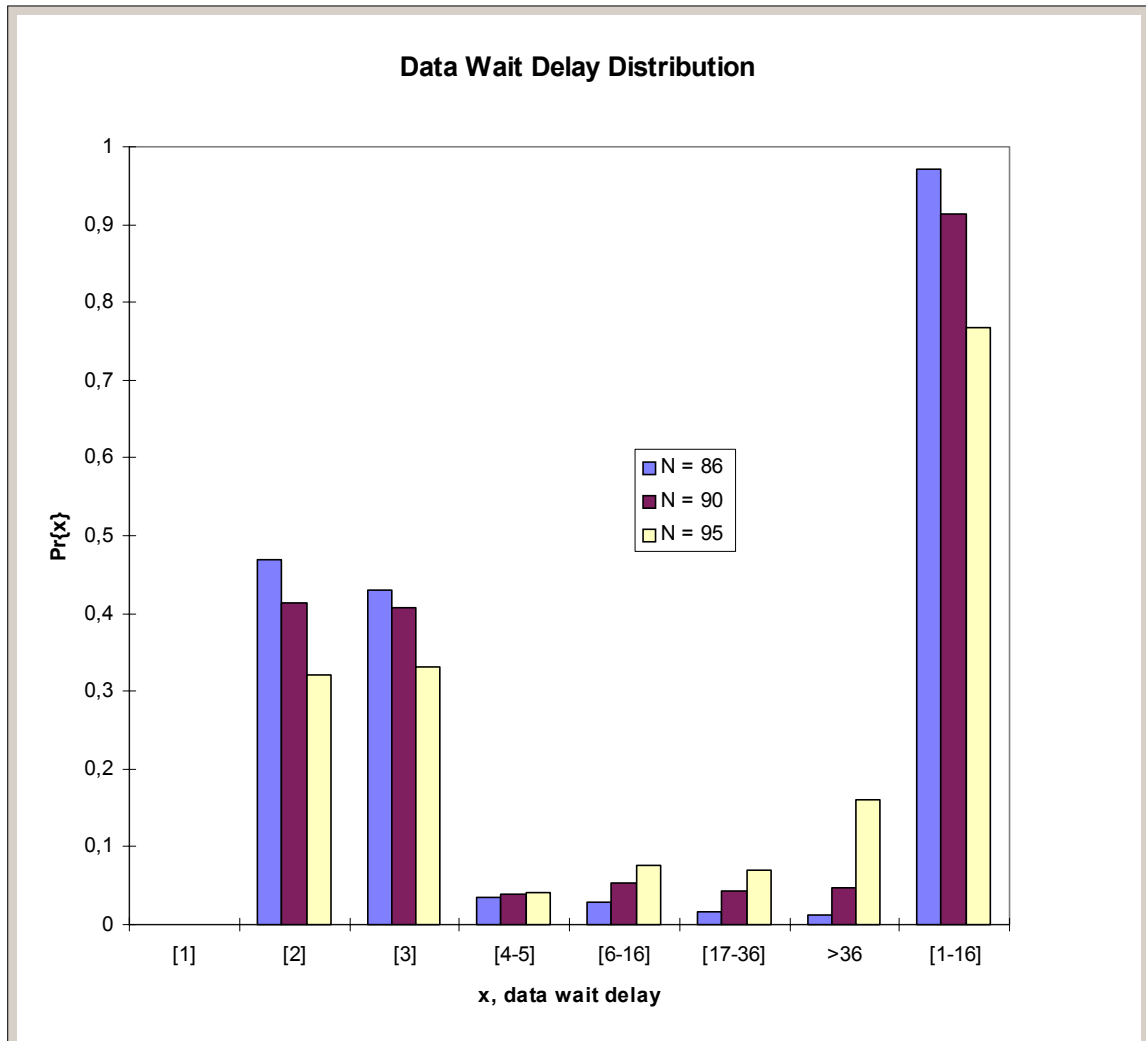
Όταν τώρα “φορτώσουμε” το σύστημα ακόμα παραπάνω, δηλαδή όταν ο αριθμός των ενεργών χρηστών φωνής γίνει ίσος με 95, και για ρυθμό άφιξης δεδομένων μόλις 0.20 μηνύματα / frame, το ποσοστό των πακέτων δεδομένων που η καθυστέρηση προσπέλασης τους δεν ξεπερνά τα 3 frames είναι περίπου 70%, ενώ το ποσοστό των πακέτων δεδομένων που βρίσκεται κάτω από το όριο των 16 frames είναι 81%.



**Σχήμα 4.3 :** Κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $N = 95$ .

Επιπλέον, στο **Σχήμα 4.4** παραθέτουμε την συγκριτική κατανομή για αριθμό ενεργών χρηστών φωνής 86, 90 και 95, αντίστοιχα, και για ρυθμό άφιξης δεδομένων 0.25 μηνύματα / frame. Όπως παρατηρούμε από το σχήμα αυτό όσο ο αριθμός των ενεργών χρηστών φωνής στο σύστημα αυξάνεται τόσο η μάζα των καθυστερήσεων πακέτων δεδομένων μετατοπίζεται προς μεγαλύτερες τιμές.

Τέλος, στο **Σχήμα 4.5** απεικονίζεται η συγκριτική κατανομή για την περίπτωση που ο αριθμός των πακέτων ανά μήνυμα είναι γεωμετρικά κατανομημένος με παράμετρο  $q = 1/B$  (μέση τιμή  $B = 8$ ), και για την περίπτωση που ο αντίστοιχος αριθμός είναι σταθερός και ίσος με 8 πακέτα ανά μήνυμα ( $M = 8$ ).



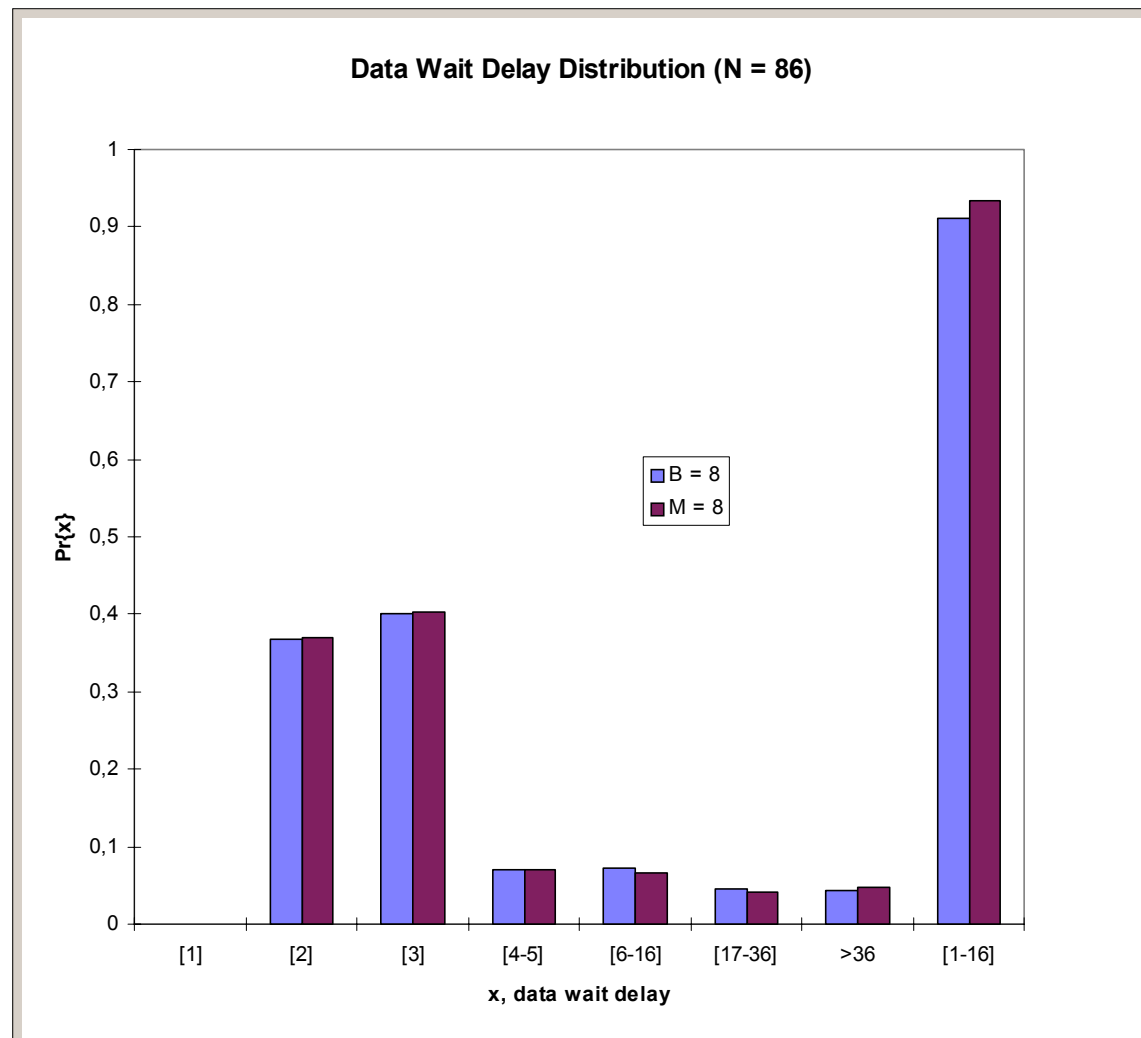
**Σχήμα 4.4 :** Συγκριτική κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $N = 86$ ,  $90$  και  $95$ , αντίστοιχα.

Και στις δύο περιπτώσεις, έχουμε  $86$  ενεργούς χρήστες φωνής στο σύστημα και ο ρυθμός άφιξης δεδομένων είναι  $0.50$  μηνύματα / frame. Όπως είχαμε δει και στην περίπτωση της μέσης καθυστέρησης προσπέλασης των πακέτων δεδομένων, έτσι και στην κατανομή της αντίστοιχης καθυστέρησης οι διαφορές μεταξύ των δύο παραπάνω περιπτώσεων είναι πολύ μικρές.

Στη συνέχεια παραθέτουμε τα αποτελέσματα της τυπικής απόκλισης,  $\sigma(D)$ , της καθυστέρησης προσπέλασης των πακέτων δεδομένων. Στο **Σχήμα 4.6** απεικονίζεται το πηλίκο της τυπικής απόκλισης,  $\sigma(D)$ , διά της μέσης τιμής,  $E(D)$ , της καθυστέρησης προσπέλασης των πακέτων δεδομένων για τις περιπτώσεις  $N = 86$ ,  $90$  και  $95$  για



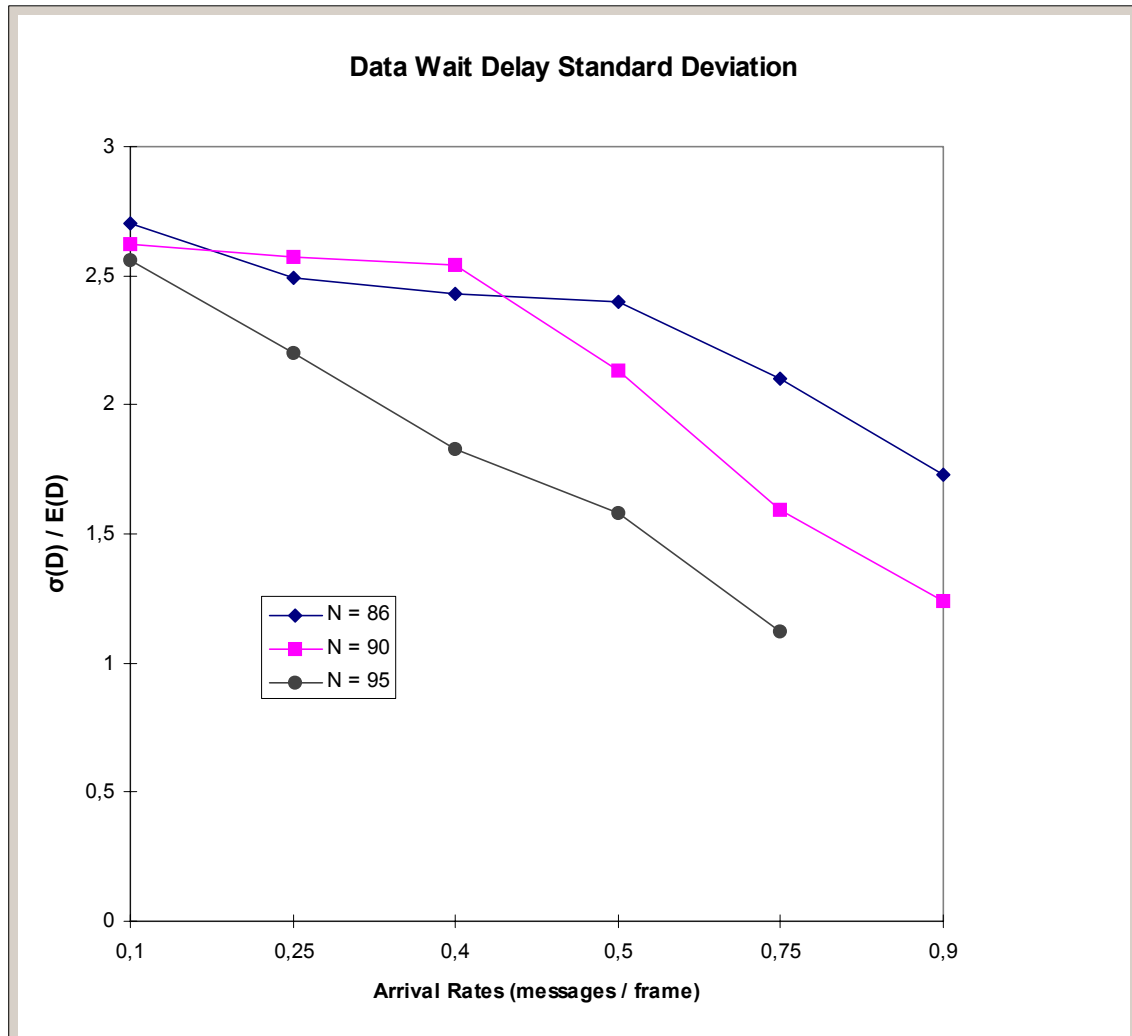
διάφορους ρυθμούς άφιξης μηνυμάτων δεδομένων στο σύστημα. Όπως φαίνεται από το σχήμα αυτό το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής



**Σχήμα 4.5 :** Συγκριτική κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για σταθερό ( $M = 8$ ) και γεωμετρικά κατανεμημένο αριθμό πακέτων ανά μήνυμα ( $B = 8$ ), με  $N = 86$  και ρυθμό άφιξης δεδομένων  $0.50$  μηνύματα / frame.

της καθυστέρησης προσπέλασης των πακέτων δεδομένων, για ρυθμούς άφιξης δεδομένων μέχρι  $0.50$  μηνύματα / frame για τις περιπτώσεις των 86 και 90 ενεργών χρηστών φωνής, και μέχρι  $0.25$  μηνύματα / frame για την περίπτωση των 95 ενεργών χρηστών φωνής, διατηρείται μεταξύ 2.5 και 2 ακολουθώντας μία φθίνουσα πορεία όσο ο ρυθμός άφιξης αυξάνεται. Για μεγαλύτερους ρυθμούς άφιξης μηνυμάτων δεδομένων το κλάσμα της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης

προσπέλασης των πακέτων δεδομένων μειώνεται και φτάνει μέχρι την τιμή 1.12 ( για  $N = 95$ , και  $\lambda = 0.75$  μηνύματα/frame).



**Σχήμα 4.6 :** Το πηλίκο της τυπικής απόκλισης διά της μέσης τιμής της καθυστέρησης προσπέλασης των πακέτων δεδομένων, σαν συνάρτηση του ρυθμού άφιξης μηνυμάτων στο σύστημα για  $N = 86, 90$  και  $95$ .

Από τα αποτελέσματα αυτά μπορούμε να συμπεράνουμε ότι η προβλεψιμότητα της καθυστέρησης των πακέτων δεδομένων με βάση τη γνώση της μέσης τιμής, αυξάνεται καθώς αυξάνεται είτε ο ρυθμός άφιξης μηνυμάτων (για σταθερό αριθμό ενεργών χρηστών φωνής στο σύστημα), είτε ο αριθμός ενεργών χρηστών φωνής στο σύστημα (για σταθερό ρυθμό άφιξης μηνυμάτων), κάτι που συμβαδίζει με την εμπειρική γνώση.





## ΚΕΦΑΛΑΙΟ 5

### Μετρικές απόδοσης του συστήματος Ολοκληρωμένης Προσπέλασης όταν οι χρόνοι μεταξύ αφίξεων διαδοχικών μηνυμάτων δεδομένων χαρακτηρίζονται από την κατανομή Pareto

Στο κεφάλαιο αυτό παραθέτουμε τα αποτελέσματα της προσομοίωσης για το σύστημα **Ολοκληρωμένης Τυχαίας Προσπέλασης Φωνής και Δεδομένων** και το οποίο παρουσιάσαμε αναλυτικά στο Κεφάλαιο 2, στην περίπτωση που οι χρόνοι μεταξύ αφίξεων διαδοχικών μηνυμάτων δεδομένων χαρακτηρίζονται από την εκρηκτική κατανομή **Pareto**. Ο κώδικας της αντίστοιχης προσομοίωσης παρουσιάζεται στο **Παράρτημα 2**.

#### 5.1 Αποτελέσματα Προσομοίωσης

Στο Κεφάλαιο 3 είδαμε ότι η βέλτιστη τιμή του μήκους  $\Delta$  του παραθύρου κάτω από τις προϋποθέσεις που θέσαμε (υπενθυμίζουμε ότι ζητούσαμε εκείνη την τιμή του  $\Delta$  που οδηγεί την λειτουργία του αλγορίθμου όσο το δυνατό πλησιέστερα στην βέλτιστη, για όλες τις τιμές της παραμέτρου  $k$  της κατανομής **Pareto** η οποία συμβολίζει τον ελάχιστο χρόνο μεταξύ διαδοχικών αφίξεων πακέτων), είναι τα 1.6 slots. Όπως αναφέραμε και στο Κεφάλαιο 3 αυτή η επιλογή στηρίχτηκε και στα αποτελέσματα της προσομοίωσης του συστήματος Ολοκληρωμένης Τυχαίας Προσπέλασης Πακέτων Φωνής και Δεδομένων. Στους **Πίνακες 5.1-5.2** φαίνεται όπως προέκυψε από τα αποτελέσματα της προσομοίωσης (η μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων) για τις τιμές της παραμέτρου  $k$  που είχαμε συζητήσει και στο Κεφάλαιο 3 (και για τις οποίες η επιλογή του  $\Delta$  στα 1.6 slots δεν ήταν απόλυτα δικαιολογημένη με τα αποτελέσματα που είχαμε παρουσιάσει εκεί), όταν το  $\Delta$  παίρνει τις τιμές 1.8, 1.6 και 1.4 αντίστοιχα και όταν στο σύστημα υπάρχουν 86 ενεργοί χρήστες φωνής.

	$\Delta = 1.8$	$\Delta = 1.6$	$\Delta = 1.4$
Arrival Rates (msg/frame)	Wait Mean Delay (ms)	Wait Mean Delay (ms)	Wait Mean Delay (ms)
0.25	59.57	59.16	59.8
0.10	89.55	86.96	89.51
0.50	398.45	182.77	185.38
1.00	560.81	542.31	545.94

**Πίνακας 5.1 :** Μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων για διάφορες τιμές του ρυθμού άφιξης μηνυμάτων δεδομένων, όταν  $k = 0.8$ .

	$\Delta = 1.8$	$\Delta = 1.6$	$\Delta = 1.4$
ArrivaRates (msg/frame)	Wait Mean Delay (ms)	Wait Mean Delay (ms)	Wait Mean Delay (ms)
0.25	71.68	71.65	74.87
0.50	108.19	108.42	112.69
0.75	199.72	200.18	209.28
1.00	516.82	500.89	568.9

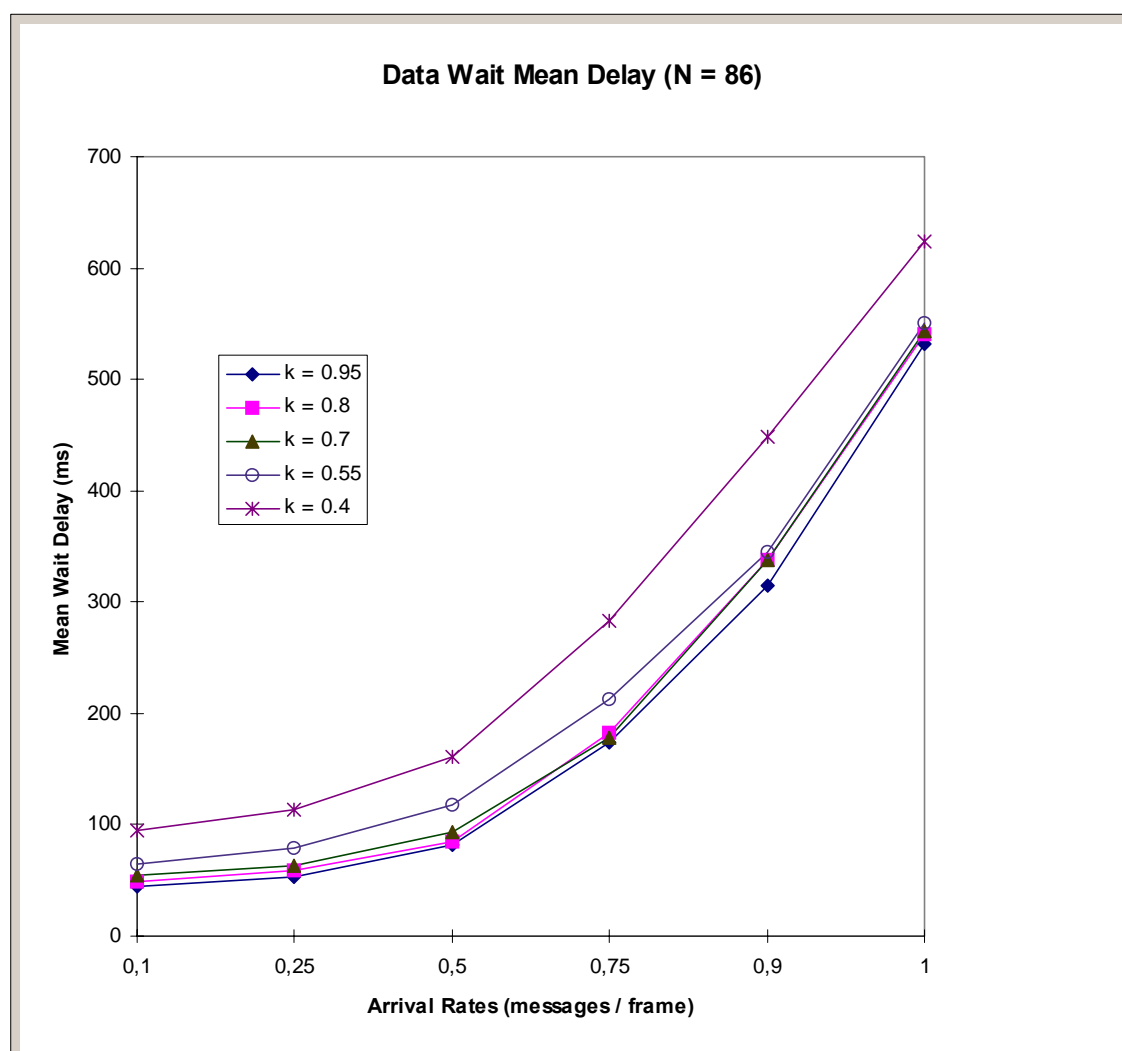
**Πίνακας 5.2 :** Μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων για διάφορες τιμές του ρυθμού άφιξης μηνυμάτων δεδομένων, όταν  $k = 0.6$ .

Όπως παρατηρούμε από τα αποτελέσματα των παραπάνω πινάκων όταν το  $\Delta = 1.6$  slots τα πακέτα δεδομένα επιτυγχάνουν τις μικρότερες καθυστερήσεις προσπέλασης. Το γεγονός αυτό μας πιστοποιεί ότι η επιλογή του  $\Delta = 1.6$  slots είναι η καλύτερη δυνατή κάτω από τις προϋποθέσεις που θέσαμε.

### 5.1.1 Μέση Καθυστέρηση Προσπέλασης των Πακέτων Δεδομένων

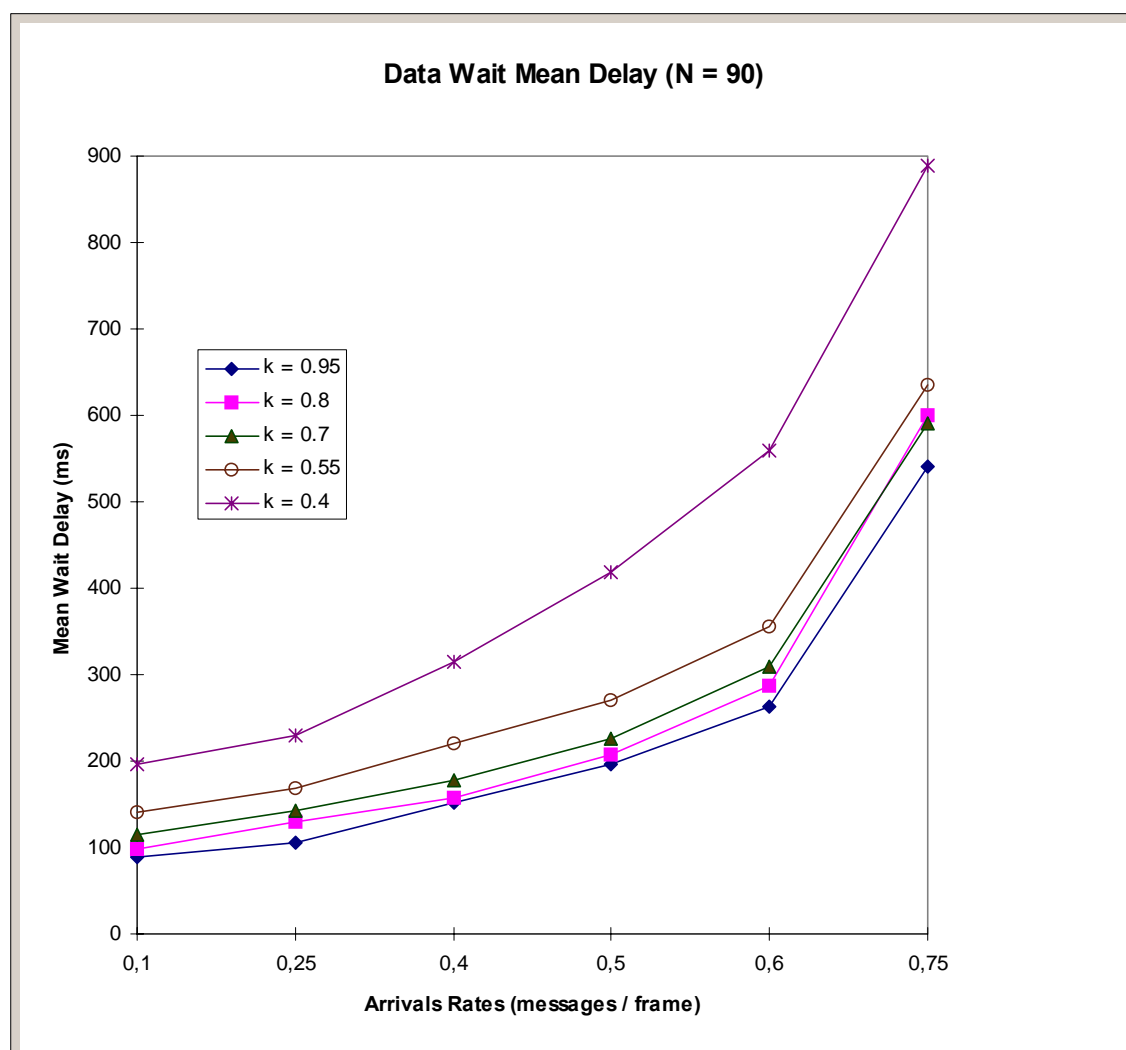
Στο **Σχήμα 5.1** φαίνονται τα αποτελέσματα της προσομοίωσης για τη μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων όταν η παράμετρος  $k$  της κατανομής **Pareto** παίρνει τις τιμές  $k = 0.95, 0.8, 0.7, 0.55$  και  $0.4$  (να υπενθυμίσουμε εδώ ότι η παράμετρος  $k$  συμβολίζει τον ελάχιστο χρόνο μεταξύ αφίξεων δύο διαδοχικών μηνυμάτων ενώ η παράμετρος  $a$  καθορίζεται από τις τιμές των  $\lambda$  και  $k$

σύμφωνα με τη σχέση της μέσης τιμής τυχαίας μεταβλητής κατανομημένης σύμφωνα με την κατανομή Pareto), και στο σύστημα υπάρχουν 86 ενεργοί χρήστες φωνής ( $N = 86$ ). Όπως βλέπουμε από το σχήμα, για ένα σταθερό ρυθμό άφιξης μηνυμάτων όσο η παράμετρος  $k$  μειώνεται τόσο η μέση καθυστέρηση προσπέλασης για τα πακέτα δεδομένων αυξάνεται. Έτσι, για ρυθμό άφιξης 0.50 μηνύματα / frame όταν  $k = 0.80$  η μέση καθυστέρηση είναι 86.96ms, ενώ όταν  $k = 0.4$  παίρνει την τιμή 160.44ms. Η μεγάλη αύξηση της μέσης καθυστέρησης προσπέλασης των πακέτων δεδομένων είναι αναμενόμενη, αφού η τιμή  $k = 0.4$  σημαίνει ότι είναι αρκετά πιθανές μέχρι και δύο αφίξεις μηνυμάτων δεδομένων στη διάρκεια ενός frame.



**Σχήμα 5.1 :** Η μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων (σε ms) για  $k = 0.95, 0.8, 0.7, 0.55, 0.4$  όταν  $N = 86$ .

Στο **Σχήμα 5.2** ο αριθμός των ενεργών χρηστών φωνής στο σύστημα αυξάνεται στους 90 ( $N = 90$ ). Σε αυτή την περίπτωση παρατηρούμε πάλι μία σταδιακή αύξηση της μέσης καθυστέρησης προσπέλασης όσο μειώνεται η παράμετρος  $k$  αλλά με μεγαλύτερους ρυθμούς από ότι στην περίπτωση με  $N = 86$ . Έτσι για την περίπτωση που ο ρυθμός άφιξης είναι 0.50 μηνύματα / frame η αύξηση της μέσης καθυστέρησης προσπέλασης των πακέτων δεδομένων όταν η παράμετρος  $k$  μειώνεται από 0.8 σε 0.7 είναι 14ms ενώ για  $N = 86$  η αντίστοιχη αύξηση ήταν μόνο 4ms. Το ίδιο φαινόμενο παρατηρείται και για όλους τους υπόλοιπους ρυθμούς άφιξης μηνυμάτων.



**Σχήμα 5.2 :** Η μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων (σε ms) για  $k = 0.95, 0.8, 0.7, 0.55, 0.4$  όταν  $N = 90$ .



Στην περίπτωση που στο σύστημα υπάρχουν 95 ενεργοί χρήστες φωνής οι καθυστερήσεις προσπέλασης των πακέτων δεδομένων είναι πολύ μεγάλες ακόμα και για πολύ μικρούς ρυθμούς άφιξης μηνυμάτων. Στον **Πίνακα 5.1** φαίνονται τα αποτελέσματα της προσομοίωσης για την περίπτωση αυτή για  $k = 0.95$  και  $0.80$ .

	$k = 0.95$	$k = 0.8$
Arrival Rates (msg/frame)	Wait Mean Delay (ms)	Wait Mean Delay (ms)
0.10	214.27	251.25
0.250	324.00	355.28
0.50	719.51	779.47
0.750	3230.9	3407

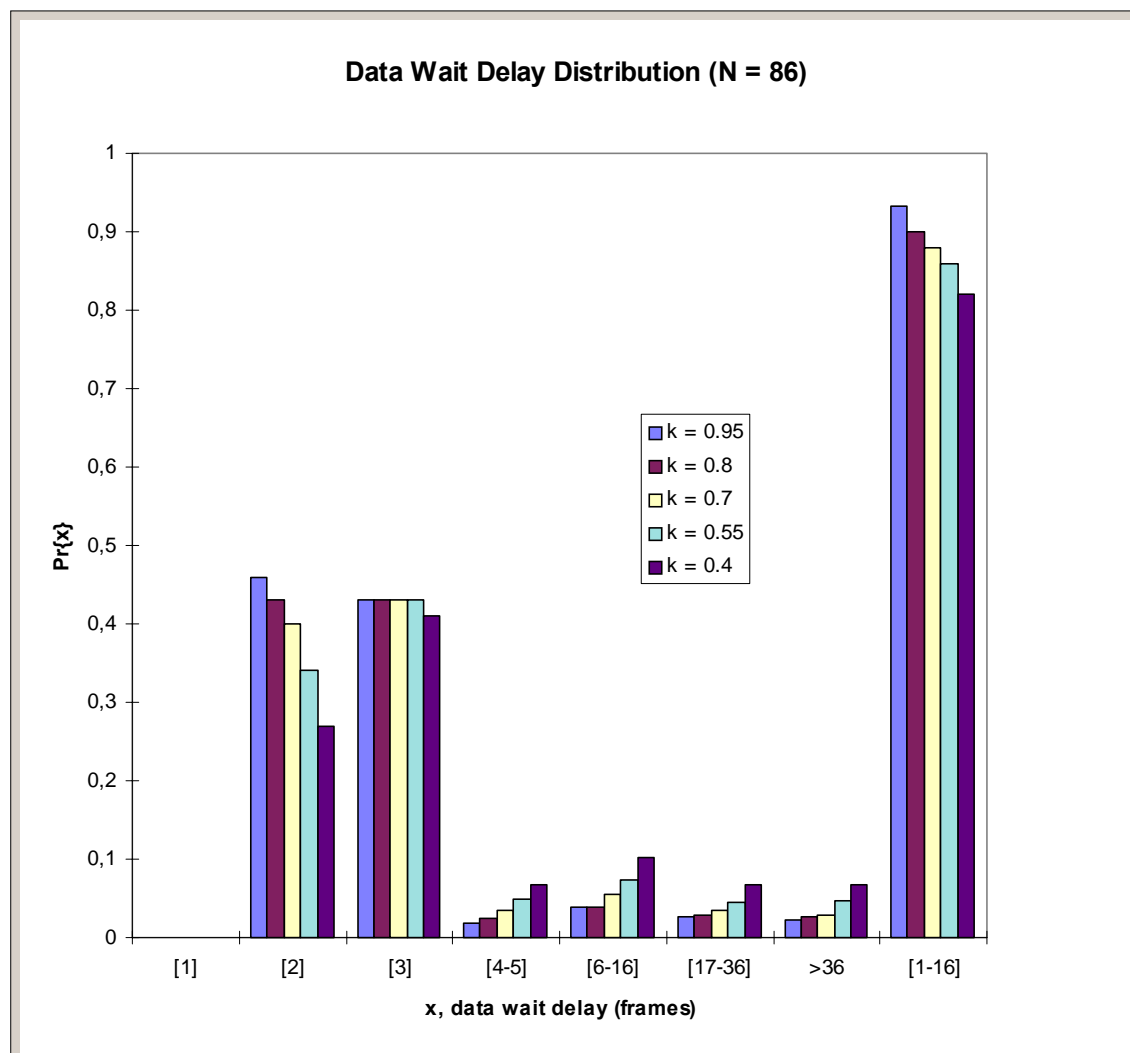
**Πίνακας 5.1 :** Η μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων για  $N = 95$  και  $k = 0.95, 0.8$ .

Όπως μπορούμε να δούμε από τα αποτελέσματα αυτά αν και η παράμετρος  $k$  παίρνει μεγάλες τιμές, ακόμα και για ρυθμό άφιξης 0.10 μηνύματα / frame η μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων είναι πολύ μεγάλη (πάνω από το όριο των 200ms), ίση με 214.27ms στην περίπτωση  $k = 0.95$  και ίση με 251.25ms στην περίπτωση  $k = 0.80$ .

### 5.1.2 Κατανομή της Καθυστέρησης Προσπέλασης των Πακέτων Δεδομένων

Για τους λόγους που αναφέραμε στην παράγραφο 4.2.1, στην συνέχεια παραθέτουμε τα αποτελέσματα της προσομοίωσης για την κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων. Σε κάθε ένα από τα ιστογράμματα απεικονίζεται η κατανομή της καθυστέρησης προσπέλασης για ένα σταθερό ρυθμό άφιξης μηνυμάτων όταν η παράμετρος  $k$  παίρνει τις τιμές 0.95, 0.8, 0.7, 0.55 και 0.4 και για 86 ή 90 ενεργούς χρήστες φωνής. Όπως παρατηρούμε από το **Σχήμα 5.3**, όταν ο ρυθμός άφιξης είναι 0.25 μηνύματα / frame και στο σύστημα υπάρχουν 86 ενεργοί χρήστες φωνής, το μεγαλύτερο ποσοστό πακέτων έχουν καθυστερήσεις που δεν ξεπερνούν τα 3 frames (36 ms). Η μεγάλη μάζα πιθανότητας

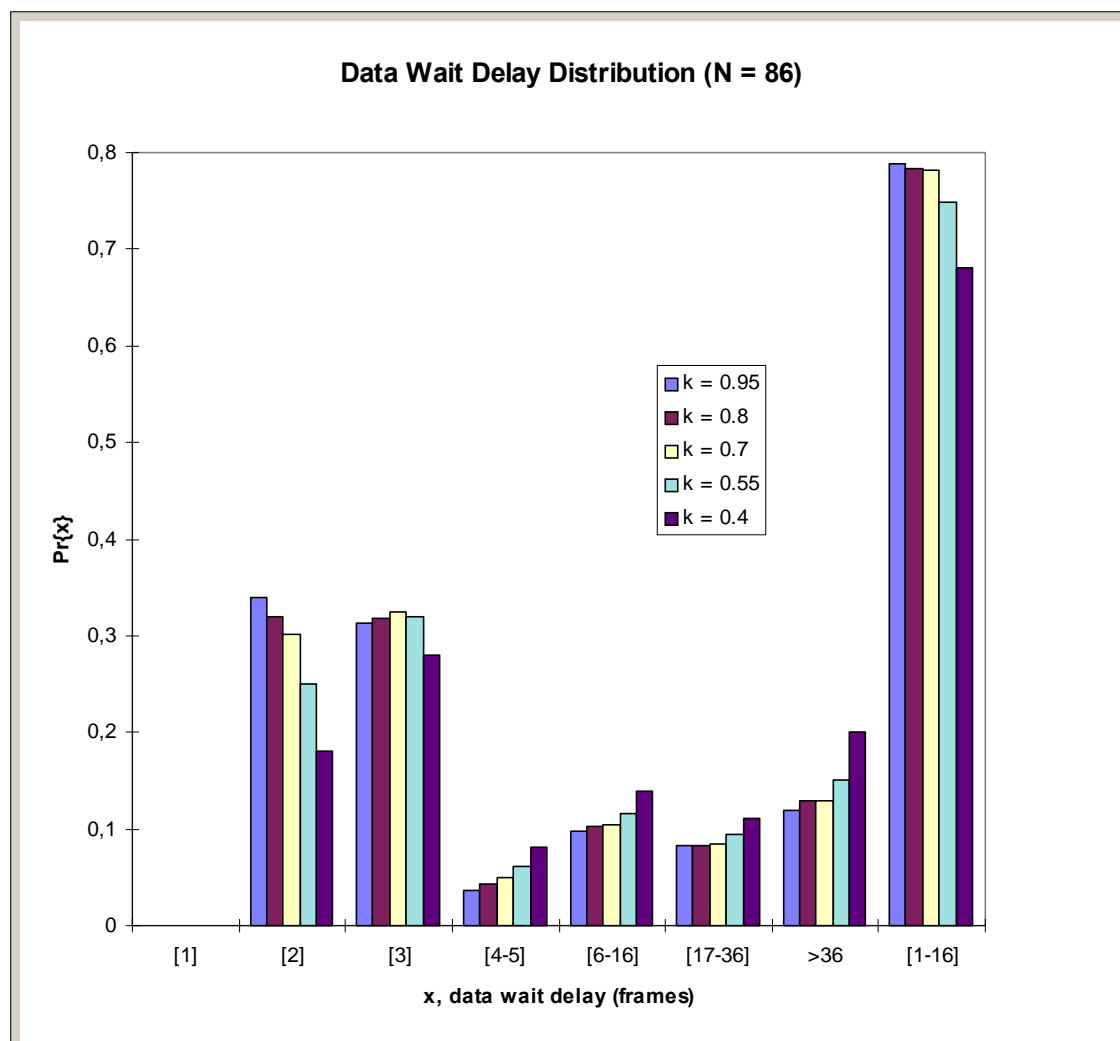
των καθυστερήσεων των πακέτων δεδομένων, όπως θα δούμε παρακάτω, μετακινείται προς μεγαλύτερες τιμές καθώς αυξάνεται ο ρυθμός άφιξης μηνυμάτων



**Σχήμα 5.3 :** Η κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $\lambda = 0.25$  μηνύματα / frame και  $N = 86$ .

ή καθώς μειώνεται η παράμετρος  $k$  της κατανομής **Pareto**. Έτσι, όταν  $k = 0.95$  και  $\lambda = 0.25$  μηνύματα / frame το 89% των πακέτων δεδομένων έχουν καθυστερήσεις προσπέλασης που δεν ξεπερνούν τα 3 frames, και το 93.3% των πακέτων καθυστερήσεις που δεν ξεπερνούν τα 16 frames (192 ms). Όταν όμως  $k = 0.4$ , για τον ίδιο όπως πριν ρυθμό άφιξης μηνυμάτων, τα παραπάνω ποσοστά μειώνονται σε 68% και 82%, αντίστοιχα. Στο **Σχήμα 5.4** απεικονίζεται η κατανομή της καθυστέρησης προσπέλασης όταν ο ρυθμός άφιξης είναι 0.75 μηνύματα / frame και στο σύστημα υπάρχουν 90 ενεργοί χρήστες φωνής (συνολικό φορτίο συστήματος 83%). Όπως

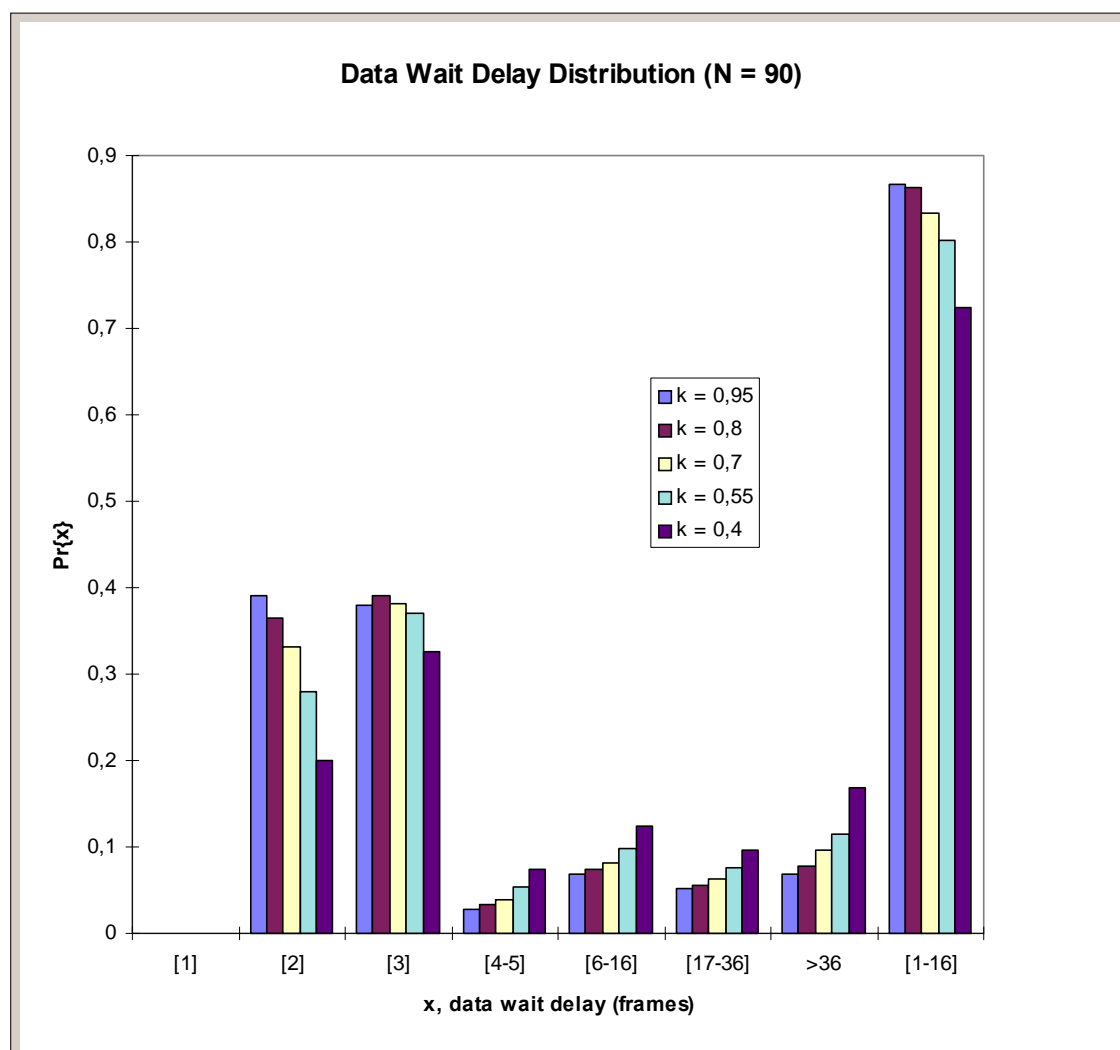
φαίνεται από το σχήμα αυτό όταν  $k = 0.7$  το ποσοστό των πακέτων που οι καθυστερήσεις προσπέλασης τους δεν υπερβαίνουν τα 3 frames (36 ms) είναι ίσο με 63% ενώ το ποσοστό των πακέτων που έχουν καθυστερήσεις μικρότερες από 16 frames (192 ms) είναι 78%. Τα ποσοστά αυτά όταν ο ρυθμός άφιξης είναι 0.25 μηνύματα / frame (δες Σχήμα 5.3) είναι 83% και 88%, αντίστοιχα.



**Σχήμα 5.4 :** Η κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $\lambda = 0.75$  μηνύματα / frame και  $N = 86$  (συνολικό φορτίο συστήματος 83%).

Στα **Σχήματα 5.5-5.6** απεικονίζεται η κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων όταν στο σύστημα υπάρχουν 90 ενεργοί χρήστες φωνής ( $N=90$ ) και ο ρυθμός άφιξης είναι 0.25 και 0.50 μηνύματα / frame, αντίστοιχα. Παρατηρούμε σ' αυτή την περίπτωση, ότι και στην περίπτωση του συστήματος με 86

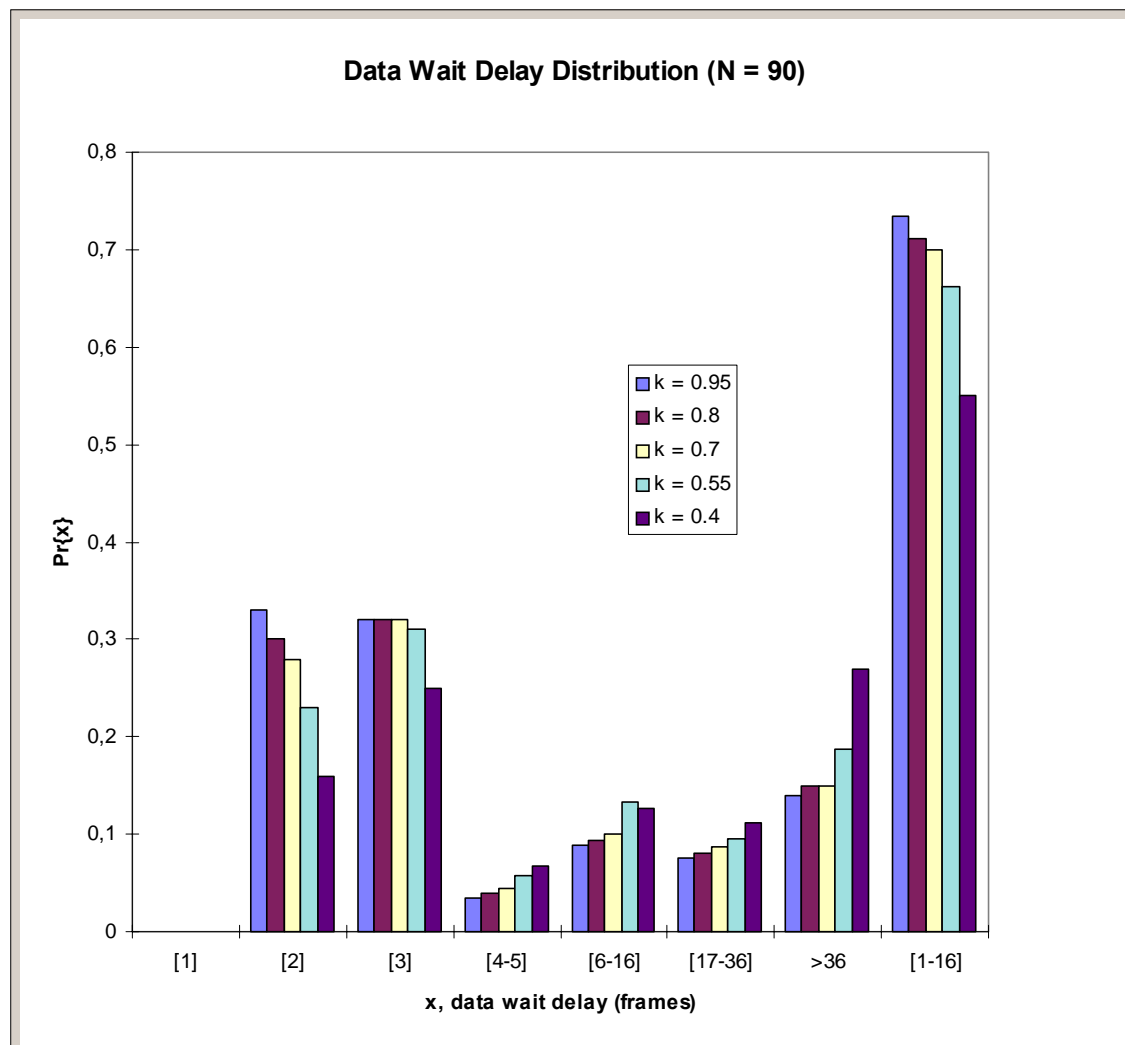
ενεργούς χρήστες φωνής. Δηλαδή, όσο αυξάνει ο ρυθμός άφιξης μηνυμάτων στο σύστημα ή όσο μειώνεται η τιμή της παραμέτρου  $k$  η μάζα πιθανότητας των καθυστερήσεων προσπέλασης των πακέτων δεδομένων μετατοπίζεται προς μεγαλύτερες τιμές.



**Σχήμα 5.5 :** Η κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $\lambda = 0.25$  μηνύματα / frame και  $N = 90$  (συνολικό φορτίο συστήματος 86.6%).

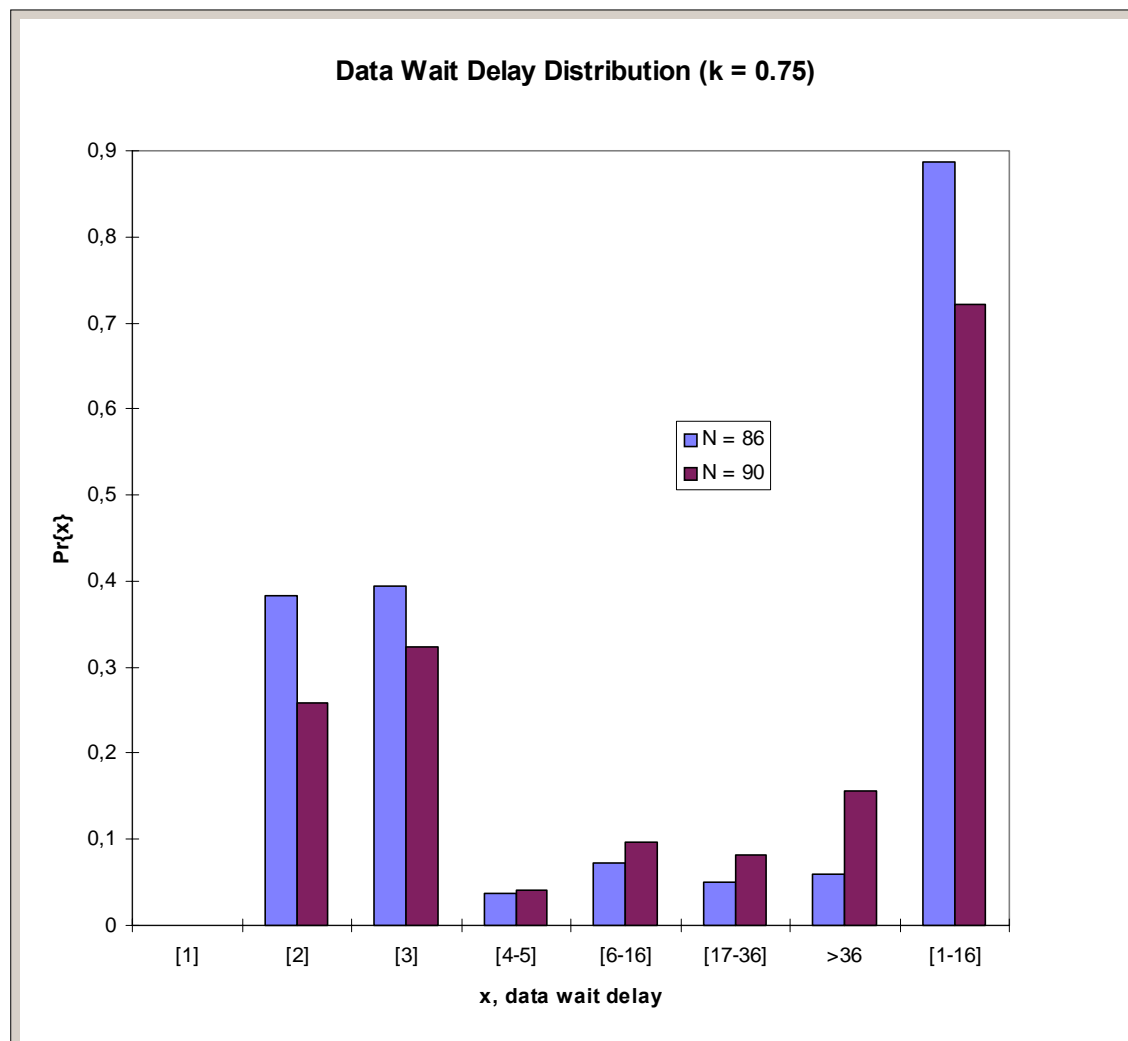
Τέλος, στο **Σχήμα 5.7** παραθέτουμε την συγκριτική κατανομή για τις περιπτώσεις των 86 και 90 ενεργών χρηστών φωνής, για μια τιμή της παραμέτρου  $k$  ( $k = 0.75$ ), και για ρυθμό άφιξης δεδομένων 0.50 μηνύματα / frame. Όπως παρατηρούμε από το σχήμα αυτό το ποσοστό των πακέτων που παρουσιάζουν μεγάλες καθυστερήσεις

προσπέλασης είναι μεγαλύτερο στην περίπτωση του συστήματος με 90 ενεργούς χρήστες φωνής.



**Σχήμα 5.6 :** Η κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $\lambda = 0.50$  μηνύματα / frame και  $N = 90$ .

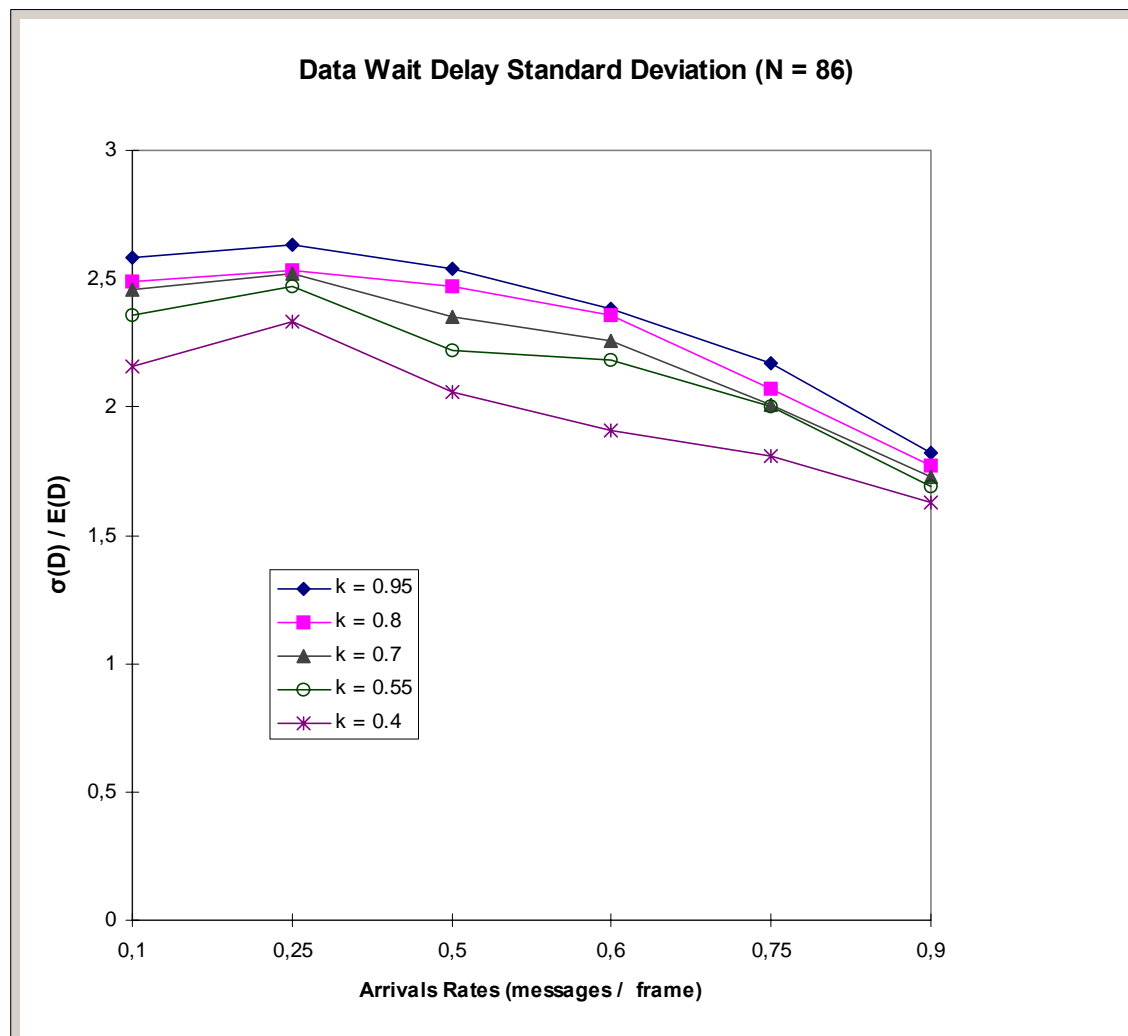
Έτσι όταν  $N = 86$  το ποσοστό των πακέτων που έχουν καθυστερήσεις προσπέλασης μικρότερες ή ίσες από 3 frames (36 ms) είναι 77.7%, ενώ το ποσοστό των πακέτων για τα οποία η καθυστέρηση προσπέλασης δεν ξεπερνά τα 16 frames (192 ms) είναι 88.7%. Τα ποσοστά αυτά στην περίπτωση  $N = 90$  είναι 58.3% και 72.1%, αντίστοιχα.



**Σχήμα 5.7 :** Η κατανομή της καθυστέρησης προσπέλασης των πακέτων δεδομένων για  $\lambda = 0.50$  μηνύματα / frame,  $N = 86, 90$  και  $k = 0.75$ .

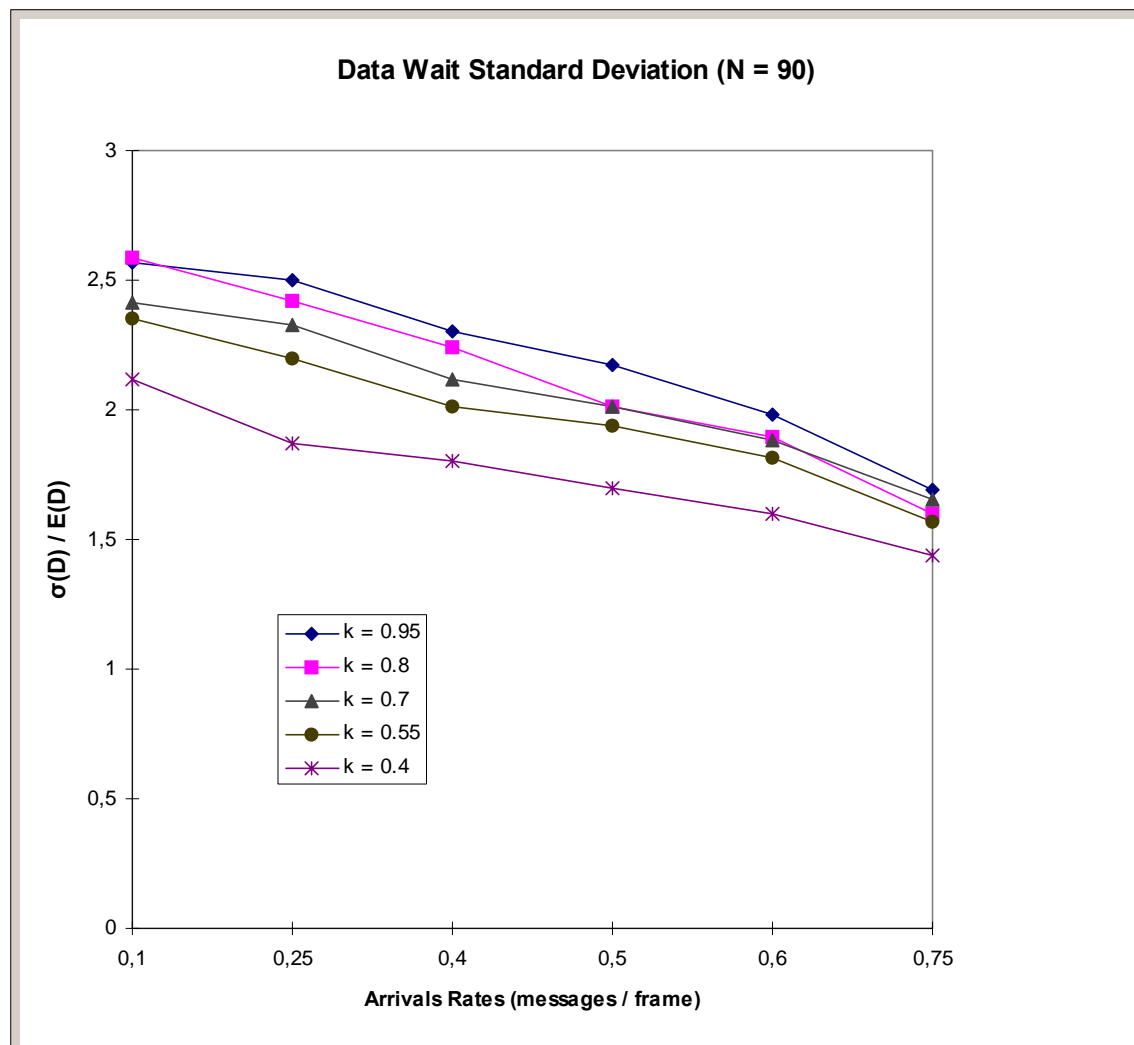
Στη συνέχεια παραθέτουμε τα αποτελέσματα της προσομοίωσης για την εκτίμηση της τυπικής απόκλισης της καθυστέρησης προσπέλασης των πακέτων δεδομένων. Στο **Σχήμα 5.8** απεικονίζεται το πηλίκο της τυπικής απόκλισης,  $\sigma(\mathbf{D})$ , διά της μέσης τιμής,  $E(\mathbf{D})$ , της καθυστέρησης προσπέλασης των πακέτων δεδομένων, για διάφορες τιμές της παραμέτρου  $k$ , όταν στο σύστημα υπάρχουν 86 ενεργοί χρήστες φωνής ( $N = 86$ ). Όπως φαίνεται από το σχήμα αυτό, το παραπάνω κλάσμα κυμαίνεται μεταξύ 1.62 και 2.58 πράγμα που μας οδηγεί στο συμπέρασμα ότι η προβλεψιμότητα της καθυστέρησης προσπέλασης των πακέτων δεδομένων με βάση γνώση της μέσης τιμής της είναι πολύ καλή. Επίσης, παρατηρούμε ότι η προβλεψιμότητα αυτή αυξάνεται

καθώς αυξάνεται ο ρυθμός άφιξης μηνυμάτων δεδομένων ή / και καθώς μειώνεται η παράμετρος  $k$  της κατανομής **Pareto**.



**Σχήμα 5.8 :** Το πηλίκο της τυπικής απόκλισης,  $\sigma(D)$ , διά τη μέση τιμή,  $E(D)$ , της καθυστέρησης προσπέλασης των πακέτων δεδομένων για διάφορες τιμές της παραμέτρου  $k$ , όταν  $N = 86$ .

Στο **Σχήμα 5.9** απεικονίζεται το πηλίκο της τυπικής απόκλισης,  $\sigma(D)$ , διά τη μέση τιμή,  $E(D)$ , της καθυστέρησης προσπέλασης των πακέτων δεδομένων για διάφορες τιμές της παραμέτρου  $k$  και όταν στο σύστημα υπάρχουν 90 ενεργοί χρήστες φωνής. Τα συμπεράσματα που βγάζουμε από αυτό το σχήμα για την προβλεψιμότητα της καθυστέρησης προσπέλασης των πακέτων δεδομένων με βάση τη γνώση της μέσης τιμής της είναι ακριβώς τα ίδια με αυτά για την περίπτωση που στο σύστημα υπάρχουν 86 ενεργοί χρήστες φωνής.



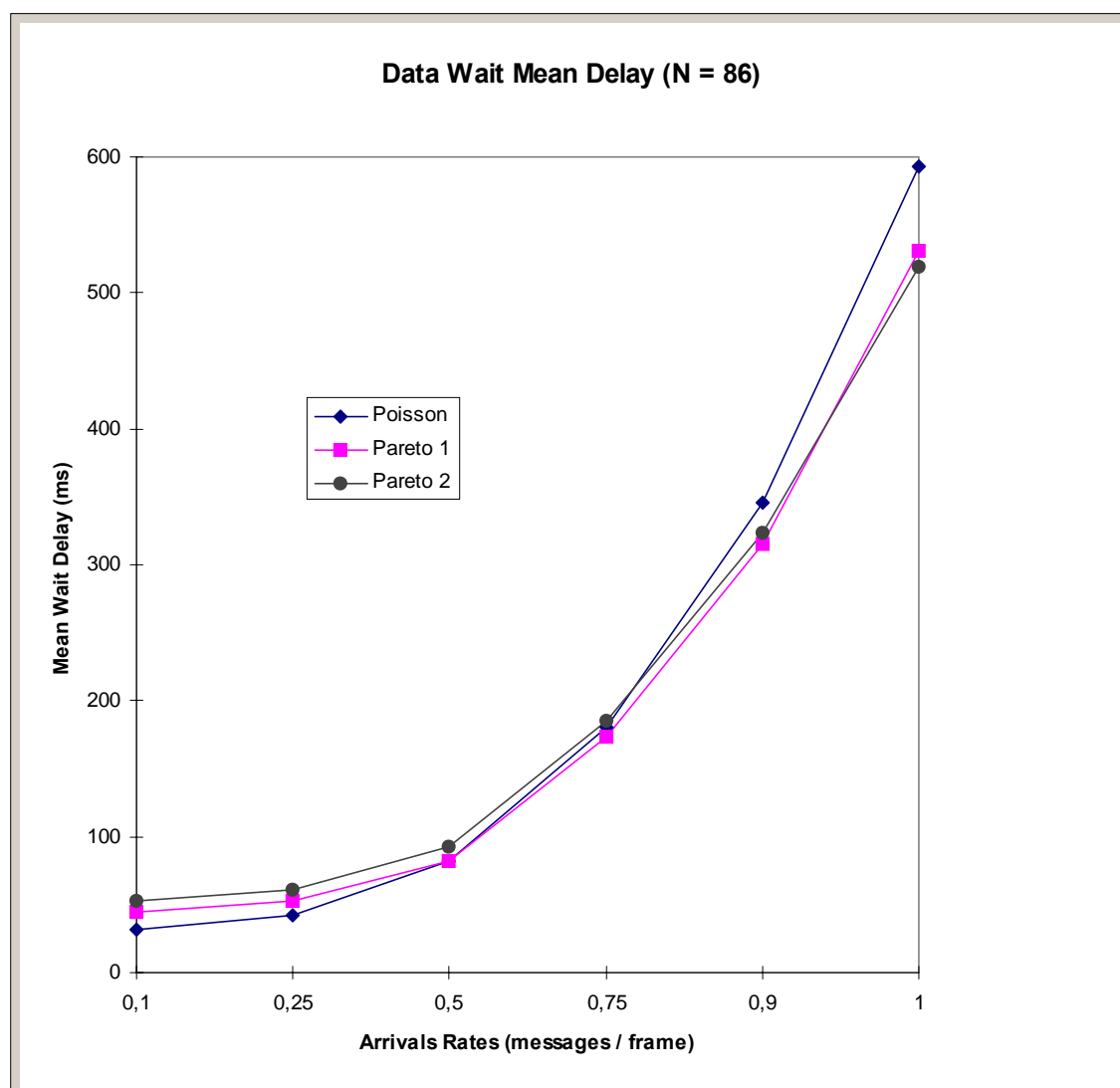
**Σχήμα 5.9 :** Το πηλίκο της τυπικής απόκλισης  $\sigma(D)$  διά τη μέση τιμή  $E(D)$  της καθυστέρησης προσπέλασης των πακέτων δεδομένων για διάφορες τιμές της παραμέτρου  $k$ , όταν  $N = 90$ .

### 5.2 Σύγκριση των καθυστερήσεων προσπέλασης των πακέτων δεδομένων για χρόνους αφίξεων κατανομημένους σύμφωνα με τις κατανομές *Pareto* και *Εκθετική (Poisson αφίξεις)*.

Στο **Σχήμα 5.10** απεικονίζεται η μέση τιμή της καθυστέρησης προσπέλασης των πακέτων δεδομένων όταν στο σύστημα υπάρχουν 86 ενεργοί χρήστες φωνής για τις κατανομές **Pareto 1**, **Pareto 2** και **Poisson**. Ως **Pareto 1** ορίζεται η κατανομή **Pareto** με παράμετρο τοποθεσίας  $k = 0.95$  slots, ενώ ως **Pareto 2** αυτή με παράμετρο τοποθεσίας  $k = 0.75$  slots. Όπως παρατηρούμε από το σχήμα αυτό όταν οι αφίξεις μηνυμάτων δεδομένων χαρακτηρίζονται από την διαδικασία **Poisson** και για μικρούς

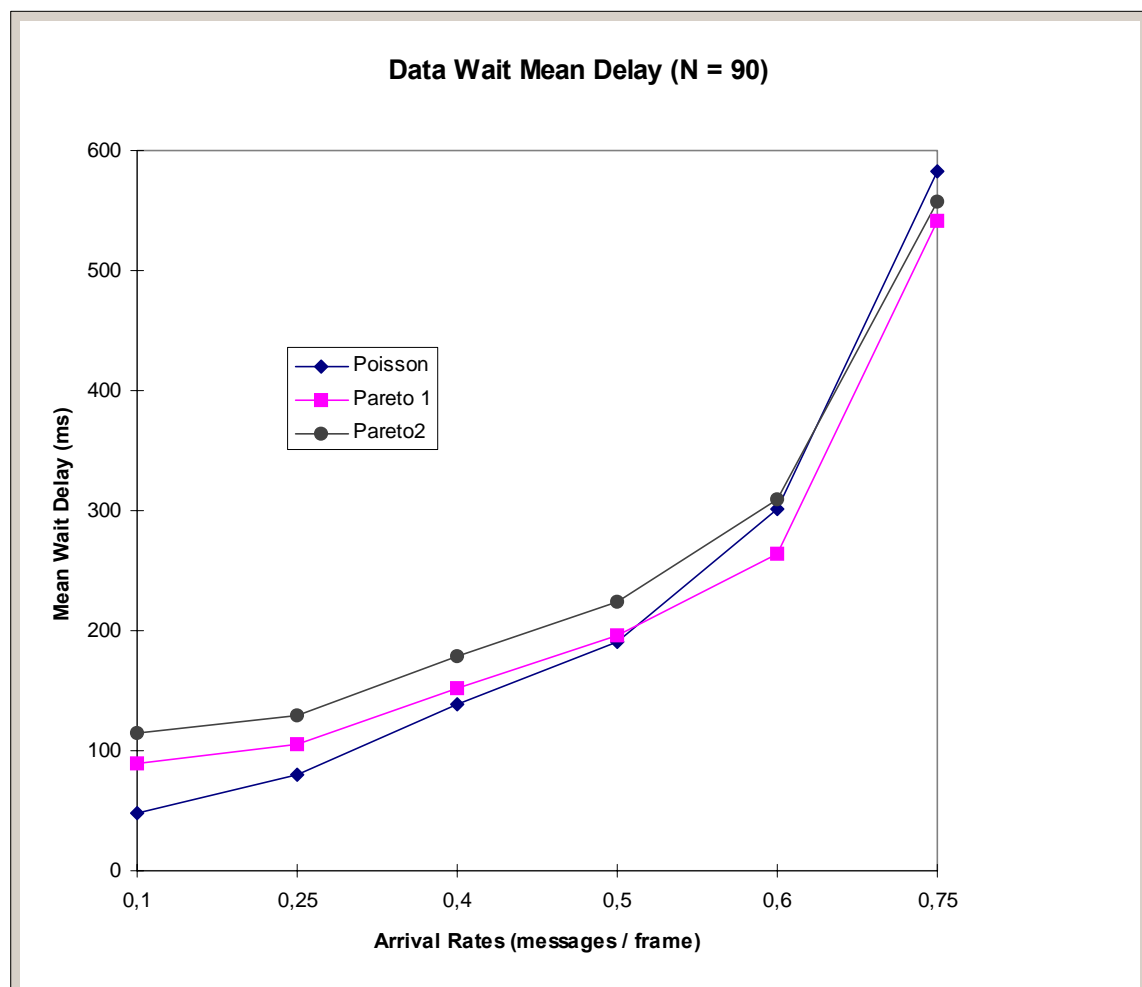


ρυθμούς αφίξεων (μέχρι 0.25 μηνύματα / frame) έχουμε μικρότερες μέσες καθυστερήσεις προσπέλασης για τα πακέτα των δεδομένων από ότι όταν οι αφίξεις μηνυμάτων χαρακτηρίζονται από **Pareto** κατανομημένους χρόνους μεταξύ αφίξεων. Για μεγάλους ρυθμούς άφιξης όμως (μεγαλύτερους από 0.50 μηνύματα / frame για την Pareto 1 και μεγαλύτερους από 0.60 για την Pareto 2), τόσο η **Pareto 1** όσο και η **Pareto 2** εμφανίζουν μικρότερες καθυστερήσεις προσπέλασης των πακέτων δεδομένων από ότι με την κατανομή **Poisson**.



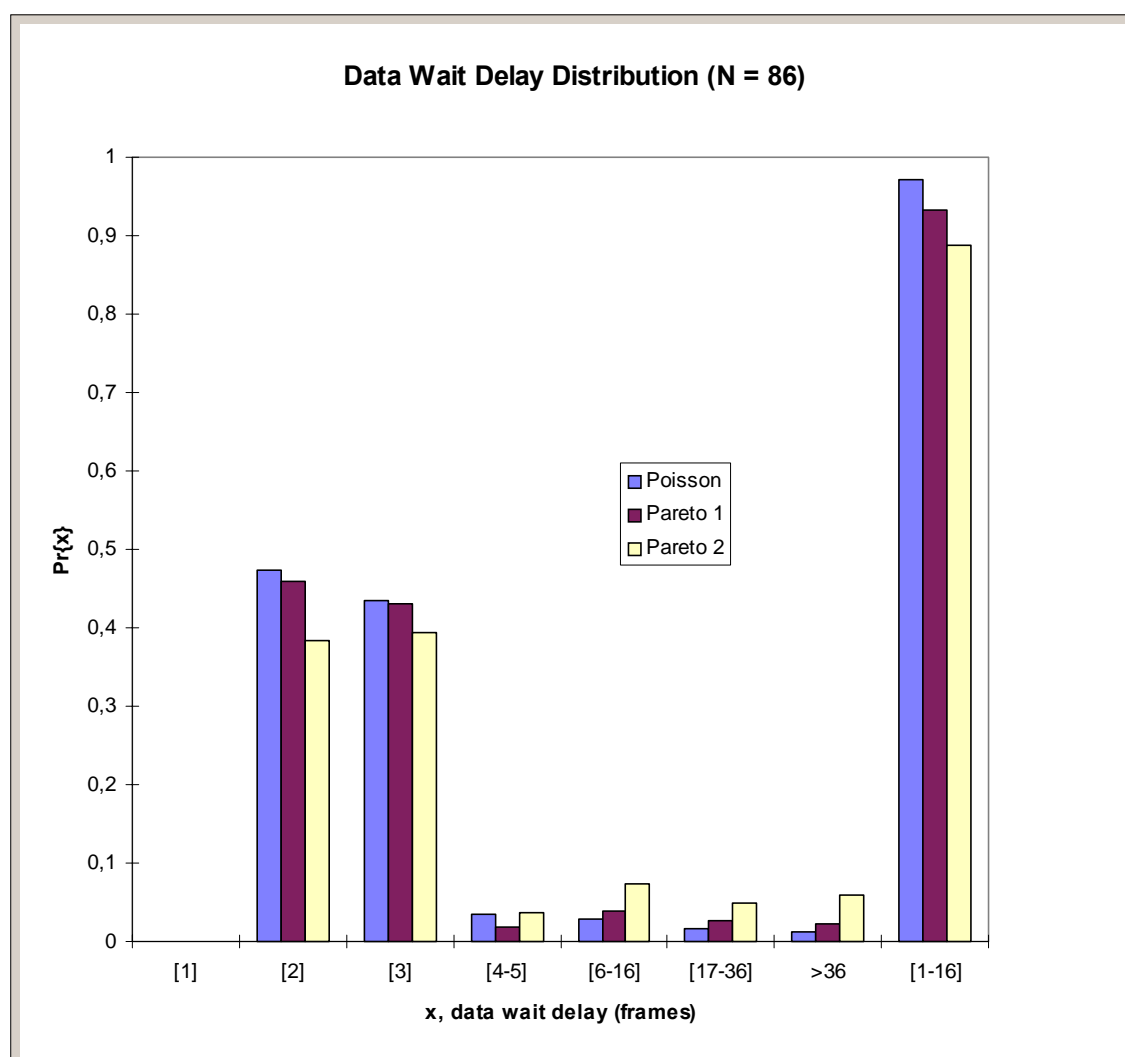
**Σχήμα 5.10 :** Μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων για τις κατανομές **Pareto 1**, **Pareto 2** και **Poisson** όταν  $N = 86$  για διάφορους ρυθμούς άφιξης μηνυμάτων δεδομένων.

Στο **Σχήμα 5.11** φαίνεται η μέση τιμή της καθυστέρησης προσπέλασης των πακέτων δεδομένων όταν στο σύστημα υπάρχουν 90 ενεργοί χρήστες φωνής για τις κατανομές **Pareto 1**, **Pareto 2** και **Poisson**. Τα συμπεράσματα της σύγκρισης είναι παρόμοια με αυτά που καταλήξαμε από το **Σχήμα 5.10**. Όταν οι αφίξεις μηνυμάτων χαρακτηρίζονται από την κατανομή **Poisson** και για ρυθμούς άφιξης μηνυμάτων μέχρι 0.60 μηνύματα / frame έχουμε μικρότερες μέσες καθυστερήσεις προσπέλασης των πακέτων δεδομένων από αυτές όταν οι αφίξεις μηνυμάτων δεδομένων χαρακτηρίζονται από **Pareto** κατανεμημένους χρόνους μεταξύ αφίξεων. Για ρυθμούς άφιξης όμως μεγαλύτερους από 0.60 μηνύματα / frame τόσο η **Pareto 1** όσο και η **Pareto 2** δίνουν μικρότερες μέσες καθυστερήσεις προσπέλασης για τα πακέτα δεδομένων.



**Σχήμα 5.11** : Μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων για τις κατανομές **Pareto 1**, **Pareto 2** και **Poisson** όταν  $N = 90$ , για διάφορους ρυθμούς άφιξης μηνυμάτων δεδομένων.

Στο **Σχήμα 5.12** απεικονίζεται η κατανομή της μέσης καθυστέρησης προσπέλασης των πακέτων δεδομένων για τις τρεις προαναφερθείσες περιπτώσεις όταν ο ρυθμός άφιξης είναι 0.50 μηνύματα / frame. Όπως φαίνεται από το σχήμα αυτό οι κατανομές **Poisson** και **Pareto 1** παρουσιάζουν περίπου την ίδια κατανομή καθυστέρησης προσπέλασης των πακέτων δεδομένων.

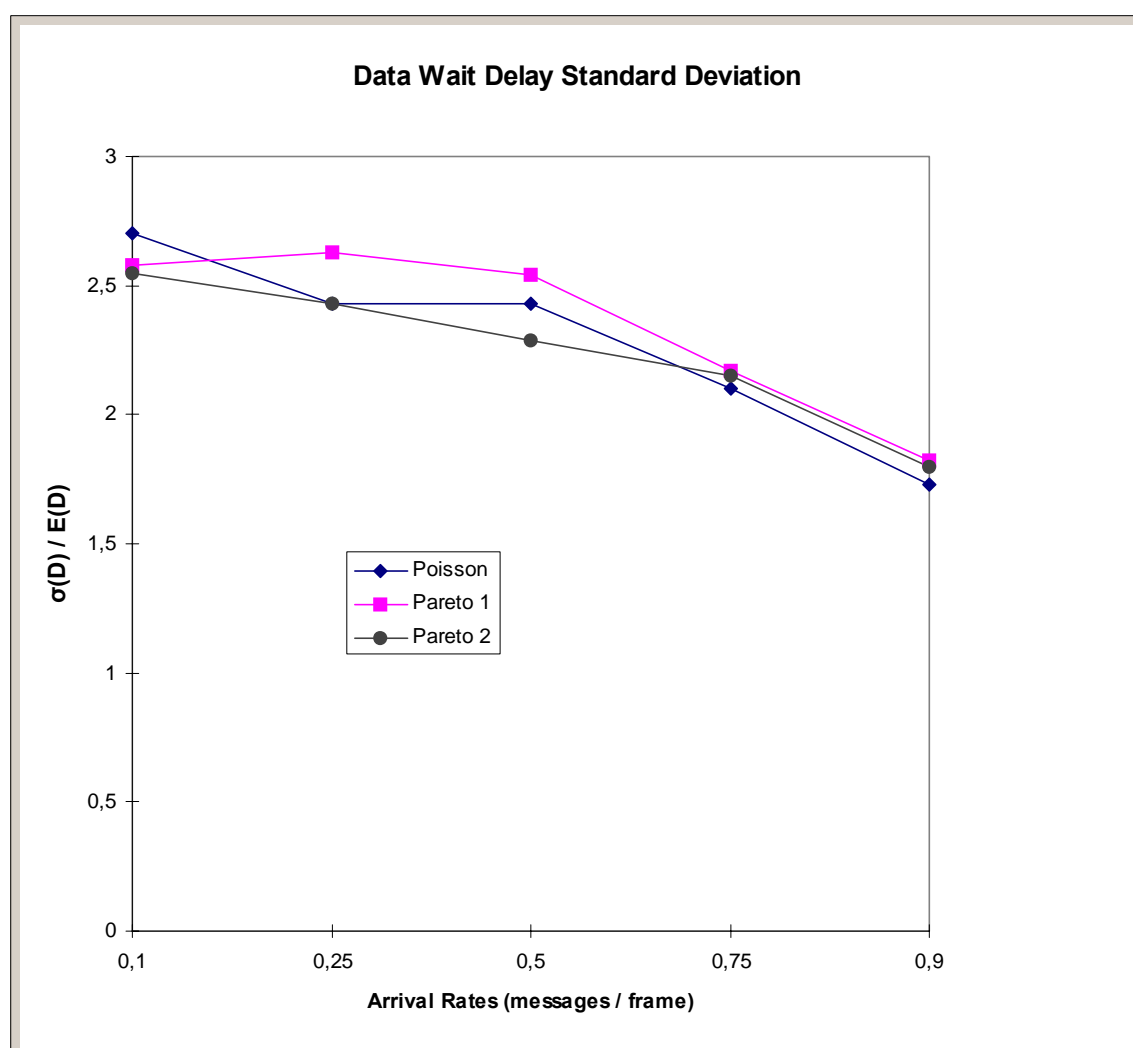


**Σχήμα 5.12 :** Η κατανομή της μέσης καθυστέρησης προσπέλασης δεδομένων για τις κατανομές **Poisson**, **Pareto 1** και **Pareto 2** όταν  $N = 86$  και  $\lambda = 0.50$  μηνύματα / frame.

Έτσι για την περίπτωση που οι αφίξεις μηνυμάτων χαρακτηρίζονται από την κατανομή **Poisson** το ποσοστό των πακέτων δεδομένων με καθυστερήσεις

προσπέλασης που δεν ξεπερνούν τα 3 frames (36ms) είναι 90% και το ποσοστό των πακέτων δεδομένων για τα οποία οι καθυστερήσεις προσπέλασης δεν υπερβαίνουν τα 16 frames (192ms) είναι 97%. Τα αντίστοιχα ποσοστά στην περίπτωση που οι χρόνοι μεταξύ διαδοχικών αφίξεων μηνυμάτων δεδομένων χαρακτηρίζονται από την κατανομή **Pareto 1** είναι 89% και 93.3%, αντίστοιχα.

Στο **Σχήμα 5.13** παρουσιάζεται το πηλίκο της τυπικής απόκλισης,  $\sigma(D)$ , διά της μέσης τιμής,  $E(D)$ , της καθυστέρησης προσπέλασης των πακέτων δεδομένων για τις τρεις προαναφερθείσες περιπτώσεις.



**Σχήμα 5.13 :** Το πηλίκο της τυπικής απόκλισης,  $\sigma(D)$ , διά τη μέση τιμή,  $E(D)$ , της καθυστέρησης προσπέλασης των πακέτων δεδομένων για τις κατανομές **Pareto 1**, **Pareto 2** και **Poisson** για διάφορους ρυθμούς άφιξης μηνυμάτων και  $N = 86$

Όπως παρατηρούμε από το σχήμα αυτό το παραπάνω κλάσμα είναι περίπου το ίδιο και στις τρεις παραπάνω περιπτώσεις. Έτσι μπορούμε να πούμε ότι η προβλεψιμότητα της καθυστέρησης προσπέλασης των πακέτων δεδομένων με βάση τη γνώση της μέσης τιμής δεν διαφέρει σημαντικά για τις περιπτώσεις που οι χρόνοι μεταξύ αφίξεων μηνυμάτων ακολουθούν την κατανομή **Pareto** ή είναι εκθετικά κατανομημένοι (**Poisson** αφίξεις).

Τέλος στον **Πίνακα 5.4** φαίνεται η εκτίμηση του throughput με βάση τα αποτελέσματα της προσομοίωσης, όταν οι αφίξεις μηνυμάτων χαρακτηρίζονται από την κατανομή **Poisson** και όταν χαρακτηρίζονται από την κατανομή **Pareto** (για διάφορες τιμές της παραμέτρου **k**). Όπως παρατηρούμε από τον πίνακα αυτό για την **Poisson** κατανομή το throughput είναι το αναμενόμενο θεωρητικά (δες παράγραφο 4.2) σε όλες τις περιπτώσεις. Στην περίπτωση όμως που οι χρόνοι μεταξύ αφίξεων διαδοχικών μηνυμάτων δεδομένων χαρακτηρίζονται από κατανομή **Pareto** σε μερικές περιπτώσεις το throughput όπως εκτιμάται από την προσομοίωση είναι μεγαλύτερο από το αναμενόμενο (δηλαδή από τον ρυθμό αφίξεων πακέτων δεδομένων στο σύστημα). Το γεγονός αυτό είναι αποτέλεσμα της έντονης εκρηκτικότητας που χαρακτηρίζει την κατανομή **Pareto**.

	k = 0.95	k = 0.75	k = 0.55	Poisson
Arrival Rates (msg/frame)	Throughput (pack/frame)	Throughput (pack/frame)	Throughput (pack/frame)	Throughput (pack/frame)
0.250	2.08	2.11	2.40	2.00
0.50	4.01	4.04	4.08	3.99
0.75	6.00	6.01	6.01	5.99
1.00	8.00	7.99	8.00	8.00

**Πίνακας 5.4 :** Το throughput του αλγορίθμου Two-Cell Stack όταν οι αφίξεις των πακέτων χαρακτηρίζονται από την κατανομή Poisson και από Pareto κατανομημένους χρόνους μεταξύ αφίξεων (για διάφορες τιμές της παραμέτρου **k** της κατανομής Pareto).

Έτσι στην περίπτωση που ο ρυθμός άφιξης είναι 0.25 μηνύματα / frame και για **k** = 0.55, το throughput που εκτιμάται μέσω της προσομοίωσης είναι 2.40 πακέτα / frame αντί 2.00 πακέτα (0.25 μηνύματα / frame \* 8 πακέτα / μήνυμα). Αυτό συμβαίνει λόγω

της έντονα εκρηκτικής φύσης της κατανομής **Pareto**, σε ένα frame φέρνει κατά μέσο όρο στη διάρκεια της προσομοίωσης 0.30 μηνύματα αντί 0.25 που είναι ο δοσμένος ρυθμός άφιξης μηνυμάτων δεδομένων. Παρατηρούμε όμως, ότι το παραπάνω φαινόμενο εξαλείφεται σε μεγαλύτερους ρυθμούς άφιξης μηνυμάτων δεδομένων.

## ΚΕΦΑΛΑΙΟ 6

### Συμπεράσματα και Μελλοντικές Επεκτάσεις

Σε αυτή την εργασία εξετάσαμε το πρόβλημα της **Ολοκληρωμένης Τυχαίας Προσπέλασης Πακέτων φωνής και Δεδομένων**, δίνοντας περισσότερη προσοχή στην μελέτη απόδοσης των δεδομένων, όταν οι χρόνοι μεταξύ αφίξεων διαδοχικών μηνυμάτων δεδομένων χαρακτηρίζονται από την εκρηκτική κατανομή **Pareto**. Η κατανομή αυτή 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 μηνυμάτων διαφόρων εφαρμογών δεδομένων.

Η κατανομή **Pareto** 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 μηνύματος, τόσο μεγαλύτερος είναι ο αναμενόμενος επιπλέον χρόνος μέχρι την επόμενη άφιξη μηνύματος).

Εξετάσαμε τον αλγόριθμο **Two-Cell Stack** (ο οποίος ανήκει στην κατηγορία των αλγορίθμων τυχαίας δεσμευμένης προσπέλασης) και επιλέξαμε τη βέλτιστη τιμή του μήκους  $\Delta$  του παραθύρου ενεργοποίησης "φρέσκων πακέτων", η οποία βελτιστοποιεί την λειτουργία του αλγορίθμου. Η επιλογή αυτή έγινε για ένα σύνολο τιμών της παραμέτρου  $k$  της κατανομής **Pareto** που μας ενδιαφέρει, και για οποιοδήποτε ρυθμό άφιξης μηνυμάτων δεδομένων. Η βέλτιστη τιμή καθορίστηκε στα  $\Delta = 1.6$  slots.

Όσον αφορά τα αποτελέσματα της μελέτης της απόδοσης των δεδομένων στο σύστημα **Ολοκληρωμένης Προσπέλασης**, είδαμε ότι η μέση καθυστέρηση προσπέλασης των πακέτων δεδομένων είναι αρκετά ικανοποιητική για τις περιπτώσεις των 86 και 90 ενεργών χρηστών φωνής (όχι όμως και για 95 ενεργούς χρήστες φωνής), και αυξάνεται με την μείωση της παραμέτρου  $k$  (να υπενθυμίσουμε εδώ ότι η παράμετρος  $k$  συμβολίζει τον ελάχιστο χρόνο μεταξύ αφίξεων δύο διαδοχικών μηνυμάτων) ή με την αύξηση του ρυθμού άφιξης μηνυμάτων δεδομένων.

Για την κατανομή της καθυστέρησης προσπέλασης είδαμε ότι, για τις περιπτώσεις των 86 και 90 ενεργών χρηστών φωνής, ένα πολύ μεγάλο ποσοστό των πακέτων δεδομένων έχουν καθυστερήσεις μικρότερες από 3 frames (36ms), με σταδιακή μείωση του ποσοστού αυτού καθώς αυξάνεται ο ρυθμός άφιξης μηνυμάτων ή καθώς μειώνεται η παράμετρος  $k$  της κατανομής **Pareto**. Επίσης, υπολογίσαμε (για τις περιπτώσεις των 86 και 90 ενεργών χρηστών φωνής) το πηλίκο της τυπικής απόκλισης,  $\sigma(D)$ , διά τη μέση τιμή,  $E(D)$ , της καθυστέρησης προσπέλασης των πακέτων δεδομένων και είδαμε ότι η προβλεψιμότητα της παραπάνω καθυστέρησης με βάση γνώση της μέσης τιμής της είναι πολύ καλή, και αυξάνεται καθώς αυξάνεται ο ρυθμός άφιξης μηνυμάτων δεδομένων ή / και καθώς μειώνεται η παράμετρος  $k$  της κατανομής **Pareto**.

Εξετάσαμε, επίσης, την απόδοση των δεδομένων για **Poisson** αφίξεις μηνυμάτων δεδομένων για τις περιπτώσεις που ο αριθμός των πακέτων δεδομένων ανά μήνυμα είναι γεωμετρικά κατανεμημένος ή σταθερός, και παρατηρήσαμε ότι οι διαφορές για τις μετρικές απόδοσης των πακέτων δεδομένων (μέση τιμή και κατανομή της καθυστέρησης προσπέλασης) είναι πολύ μικρές.

Συγκρίναμε, επίσης, την απόδοση των δεδομένων για χρόνους αφίξεων κατανεμημένους σύμφωνα με τις κατανομές **Pareto** (για  $k = 0.95, 0.75$ ) και **Εκθετική** (**Poisson** αφίξεις) και παρατηρήσαμε ότι όσον αφορά τη μέση καθυστέρηση προσπέλασης πακέτων δεδομένων η **Pareto** (και για τις δύο παραπάνω τιμές της παραμέτρου  $k$ ) εμφανίζει μικρότερες τιμές όταν ο ρυθμός άφιξης μηνυμάτων δεδομένων υπερβεί μία συγκεκριμένη τιμή (μεγαλύτερος από 0.50 μηνύματα / frame για  $k = 0.95$ , και μεγαλύτερος από 0.60 μηνύματα / frame για  $k = 0.75$ ). Όσον αφορά την κατανομή και το κλάσμα της τυπικής απόκλισης διά την μέση τιμή της καθυστέρησης προσπέλασης των πακέτων δεδομένων, οι δύο παραπάνω περιπτώσεις εμφανίζουν παρόμοια συμπεριφορά.

Στην προσέγγιση που χρησιμοποιήσαμε για το μοντέλο **Ολοκληρωμένης Τυχαίας Προσπέλασης Πακέτων Φωνής και Δεδομένων** θεωρήσαμε ότι οι χρήστες φωνής χρησιμοποιούν τον ιδανικό αλγόριθμο. Σύμφωνα με τον αλγόριθμο αυτό, όλα τα πακέτα αίτησης που υπάρχουν στην αρχή ενός frame μεταδίδονται επιτυχώς μέσα στη διάρκεια του διαστήματος αιτήσεων φωνής του frame. Αυτό που αξίζει να μελετηθεί μελλοντικά είναι η συμπεριφορά των μετρικών απόδοσης των πακέτων δεδομένων



κάτω από την υπόθεση ότι οι χρήστες φωνής θα χρησιμοποιούν κάποιον υλοποιήσιμο αλγόριθμο τυχαίας προσπέλασης (π.χ. ALOHA, τον Two-Cell Stack, ή τον 0.487 του Gallager [4]) για την μετάδοση των πακέτων αίτησης τους στο διάστημα αιτήσεων φωνής του frame. Ένας τέτοιος αλγόριθμος θα υποστηρίζει μικρότερο αριθμό από ενεργούς χρήστες φωνής από τον ιδανικό λόγω των συγκρούσεων που θα συμβαίνουν σε mini-slots του διαστήματος αιτήσεων φωνής. Συνεπώς, ο αλγόριθμος προσπέλασης των δεδομένων θα έχει την δυνατότητα να υποστηρίζει μεγαλύτερους ρυθμούς άφιξης μηνυμάτων δεδομένων, και να επιτύχει χαμηλότερες καθυστερήσεις πακέτων δεδομένων από τους αντίστοιχους με τον ιδανικό αλγόριθμο. Τέλος, ενδιαφέρον παρουσιάζει η μελέτη του συστήματος κάτω από διαφορετικές κατανομές για τις αφίξεις μηνυμάτων δεδομένων αλλά και για τον αριθμό πακέτων δεδομένων ανά μήνυμα.

## ΠΑΡΑΡΤΗΜΑ 1

Στο παράρτημα αυτό παραθέτουμε τον κώδικα της προσομοίωσης του αλγορίθμου **Two-Cell Stack** καθώς και τον κώδικα για την γεννήτρια τυχαίων αριθμών κατανεμημένων σύμφωνα με την εκρηκτική κατανομή **Pareto**. Η προσομοίωση αυτή χρησιμοποιήθηκε για την επιλογή του βελτίστου μήκους  $\Delta$  του παραθύρου ενεργοποίησης.

File name : Two-Cell\_Stack.c  
Two-Cell Stack Simulator

```
#include <stdio.h>
#include <math.h>
#define a 1.68
#define kk 0.8
#include "pareto.h"

#define npr 1000000 /*number of packets required*/
#define D 1.5 /* window size*/
#define p 0.5 /*retransmission probability*/

long num_new_packets,slot_success,slot_collision,num_packets_trans;
float delay[20],y,count,t,count_delta,delta,slot_empty,current_slot,
      mean_delay,persent_slot_suc,persent_slot_emp,persent_slot_col,
      persent_slot_emp_resol,lag,lag2,lag1,b,x,rate,ak,ck;
double cumdelay;
int counter[20],m,c,i,j,k,pack=0,hit,l,n,r,z,v,f,h,flag;

void initialize()
{
    lag=1; /*unexamined interval*/
    lag1=0; /*unexamined interval produced in a collision*/
    lag2=0; /*unexanmined interval after a transmission*/
    b=0.0; /*lag - D*/
    num_new_packets=0;
    count_delta=0.0;
    current_slot=1;
    flag=0;
    cumdelay=0.0; /*total delay of packets succesfully transmitted*/
    mean_delay=0.0; /* mean packet delay*/
    slot_success=0;
    slot_collision=0;
    slot_empty=0;
    persent_slot_suc=0.0;
    persent_slot_col=0.0;
    persent_slot_emp=0.0;
```

```

num_packets_trans=0;

for (i=0;i<20;++i)
{
    delay[i]=0.0; /*delay of the i-th packet*/
    counter[i]=0; /*counter value of the i-th packet*/
}
}

void main()
{
    initialize();
    while (num_packets_trans < npr) /*the simulation stops after the transmission of npr packets*/
    {
        if (lag>D)
            delta=D;
        else if(lag>0)
            delta=lag;
        else if(lag<=0)
            delta=1;
        if ((count_delta<delta)&&(current_slot!=1))
        {
            delay[0]=delta-count_delta;
            i=1;
        }
        else
            i=0;
        while(count_delta < delta)
        {
            t=pareto(); /*time between packet arrivals*/
            count_delta+=t;
            if (count_delta<delta)
            {
                delay[i]=delta-count_delta;
                ++i;
            }
            ++num_new_packets;
        }
        count_delta-=delta;

        if ((num_new_packets>0)&&(current_slot==1))
            --num_new_packets;

        if (num_new_packets==0)          /* No Collision */
        {
            ++current_slot;
            flag=1;
        }
        else if (num_new_packets==1)    /*No Collision*/
        {
            flag=1;
            lag2=lag-delta;
            ++num_packets_trans;
            ++slot_success;

            if (lag2<=0)
                cumdelay+=delay[0]+1;
            else

```

```

        cumdelay+=delay[0]+lag2+1;
        ++current_slot;
    }

else if (num_new_packets>=2)    /*Collision*/
{
    flag=0;
    ++slot_collision;
    ++current_slot;
    ++lag1;
    n=num_new_packets;
    c=n;
    cumdelay+=n;

    /******
    Two-Cell Stack Algorithm
    *****/

    for (k=0;k<n;++k)    /*Initialize counter values */
    {
        if ((y=uniran())> p)
            counter[k]=2;
        else
            counter[k]=1;
    }
    while (c!=0) /*there are packets which have not transmitted yet*/
    {

        if ((c<=n)&&(c>0));
        {
            hit=0; /*packets which have permission to transmit*/

            for (i=0;i<n;++i)
            {
                if (counter[i]==1)
                {
                    ++hit;
                    pack=i; /*packet which has permission to transmit*/
                }
            }

            if (hit==1) /*No Collision*/
            {
                lag2=lag-delta;
                ++num_packets_trans; /*successful transmission*/
                ++slot_success;
                ++lag1;
                ++current_slot;

                if(lag2<=0)
                    cumdelay+=delay[pack]+1;
                else
                    cumdelay+=delay[pack]+lag2+1;

                delay[pack]=0;
                counter[pack]=0;
                --c;
            }
        }
    }
}

```

```

        for(m=0;m<n;++m)
        {
            if(counter[m]!=0)
                delay[m]+=1;
        }
        for(z=0;z<n;++z)
        {
            if(counter[z]==2)
                counter[z]=1;
        }
    }

    else if (hit>=2) /*Collision*/

    {
        for(v=0;v<n;++v)
        {
            if(counter[v]==1) /*initialize counter values*/
            {
                if((y=uniran())>p)
                    counter[v]=2;
                else
                    counter[v]=1;
            }
        }
        ++slot_collision;
        ++lag1;
        ++current_slot;

        for(h=0;h<n;++h)
        {
            if(counter[h]!=0) /* the packet has not transmitted yet*/
                ++delay[h];
        }

    }

    else if (hit==0) /*No Collision*/

    {
        for(h=0;h<n;++h)
        {
            if(counter[h]!=0)
                ++delay[h];
        }
        ++slot_empty;
        ++lag1;
        ++current_slot;

        for(f=0;f<n;++f)
        {
            if (counter[f]==2)
                counter[f]=1;
        }
    }
}
}

```

```

b=lag-D; /*the difference between unexamined interval and window size ( $\Delta$ ) after
                                                the end of the collision*/

if((b>0)&&(flag==1))
    lag=b+lag1+1;
else if((b>0)&&(flag==0))
    lag=b+lag1;
else if((b<=0)&&(flag==1))
    lag=lag1+1;
else if((b<=0)&&(flag==0))
    lag=lag1;

lag1=0;
b=0;
c=0;
n=0;
l=0;
num_new_packets=0;
lag2=0;

}

mean_delay=cumdelay/num_packets_trans;
persent_slot_suc=(slot_success/current_slot);
persent_slot_col=slot_collision/current_slot;
persent_slot_emp_resol=slot_empty/current_slot;
persent_slot_emp=1-(persent_slot_suc + persent_slot_col);

printf("Number of num_packet_trans : %ld \n",num_packets_trans);
printf("simulation slots   : %f\n",current_slot);
printf("Mean value of delay : %f\n",mean_delay);
printf("Percent of collision slots   : %f\n",persent_slot_col);
printf("Percent of empty slots      : %f\n",persent_slot_emp);
printf("percent of empty slots in resolution :%f\n",persent_slot_emp_resol);*/
printf("Unexamined interval       : %f\n", lag);
}

```

**File name : pareto.h**  
**Pareto Number Generator**

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define q 2147483647

float pareto()
{
    int r;
    float t,u,pw;
    r=0; t=0.0; u=0.0; pw=0.0;

```

```
r=rand();
u=(float)r/q;
pw=pow(u,1/a);

if (pw!=0.0)
    t=(float)kk/pw;
else
    {
        while (pw==0)
        {
            r=rand();
            u=(float)r/q;
            pw=pow(u,1/a);
            if (pw!=0.0)
                t=(float)kk/pw;
        }
    }
return t;
}
```

The random number generator in (0,1] :

```
float uniran()
{
    int r=0;
    float u=0.0;

    r=rand();
    u=(float)r/q;
    return u;
}
```