



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σύγχρονα και Ασύγχρονα Πρωτόκολλα
Οπτικών Δικτύων Τεχνολογίας WDM**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δέσποινα Α. Παυλίδη

Επιβλέπων : Ιωάννης Πουντουράκης
Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2008



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σύγχρονα και Ασύγχρονα Πρωτόκολλα Οπτικών Δικτύων Τεχνολογίας WDM

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δέσποινα Α. Παυλίδη

Επιβλέπων : Ιωάννης Πουντουράκης
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21^η Οκτωβρίου 2008.

.....
Ι. Πουντουράκης
Καθηγητής Ε.Μ.Π

.....
Μ. Αναγνώστου
Καθηγητής Ε.Μ.Π

.....
Ε. Συκάς
Καθηγητής Ε.Μ.Π

Αθήνα, Οκτώβριος 2008

.....
Δέσποινα Α. Παυλίδη
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Δέσποινα Α. Παυλίδη, 2008
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Στην παρούσα διπλωματική εργασία μελετώνται σύγχρονα και ασύγχρονα πρωτόκολλα οπτικών δικτύων WDMA τεχνολογίας σε τοπολογία παθητικού αστερά. Η WDMA τεχνολογία επιτρέπει στα δίκτυα να προσπεράσουν το πρόβλημα της ηλεκτρονικής συμφόρησης που προκαλείται από τη χαμηλή ταχύτητα διάδοσης σε σύγκριση με την πολύ υψηλή ταχύτητα μετάδοσης των δεδομένων στην οπτική ίνα. Αναπτύσσονται αλγόριθμοι που εκμεταλλεύονται την υψηλή καθυστέρηση διάδοσης, για αποφυγή των συγκρούσεων στα κανάλια δεδομένων. Επίσης μελετάται το φαινόμενο των συγκρούσεων στο δέκτη και πώς αυτό μπορεί να αντιμετωπιστεί, για τη βελτίωση σημαντικών παραμέτρων ενός δικτύου όπως είναι η διέλευση (throughput) και η καθυστέρηση.

Λέξεις Κλειδιά

WDMA τεχνολογία, σύγχρονα και ασύγχρονα πρωτόκολλα, παθητικός αστεράς, καθυστέρηση διάδοσης, συγκρούσεις στο δέκτη, διέλευση, καθυστέρηση

Abstract

In the present thesis, we study and develop synchronous and asynchronous protocols for WDMA technology optical networks. The WDMA technique provides a way for optical networks to overcome the electronic bottleneck caused by the high propagation delay latency in comparison with the high transmission rates in optical networks. We develop efficient algorithms that use the propagation delay in favor of the network, in order to avoid data channels collisions. In addition, the presented algorithms deal with the receiver collisions phenomenon, in order to improve the crucial networks parameters, throughput and delay.

Key Words

WDMA, synchronous and asynchronous protocols, passive star topology, propagation delay latency, receiver collisions phenomenon, throughput, delay

Ευχαριστίες

*Ο ασήμαντος δάσκαλος λέει.
Ο καλός δάσκαλος εξηγεί.
Ο ανώτερος δάσκαλος αποδεικνύει.
Ο σπουδαίος δάσκαλος εμπνέει.*

William Arthur Ward

Ευχαριστώ τον κ. Πουντουράκη που καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας υπήρξε ένας σπουδαίος δάσκαλος!

Ένα μεγάλο ευχαριστώ στην κα. Π. Μπαζιάνα για τη συνεργασία, την εμπύχωση και την πολύτιμη βοήθειά της.

Ευχαριστώ στους φίλους για τη συμπαράσταση, για την ενθάρρυνση και τα χαμόγελα. Χαρούλα, Νάντια, Γιώργο, Μαίρη, Νίκη, Τόμα, Δέσποινα σας ευχαριστώ.

Ευχαριστώ τους αδερφούς μου, Παύλο και Μανώλη για τη συμπόρευση κατά τη διάρκεια των φοιτητικών μου χρόνων και τη μικρή Αλεξάνδρα που τα γέλια της οδήγησαν σε ξεκούραστα ταξίδια.

Ευχαριστώ τους γονείς μου για άλλη μία φορά, που σε αυτήν, τη δύσκολη, μα παράλληλα πολύ δημιουργική περίοδο της ζωής μου, ήταν πάντα εκεί, ετοιμοπόλεμοι και ψύχραιμοι, συμβουλευτικοί κι ελπιδοφόροι, με αυτή τη μεγάλη και ζεστή αγκαλιά τους πάντα ανοιχτή.

Στη γιαγιά τη Δέσποινα

Περιεχόμενα

Περίληψη.....	5
Abstract	7
Ευχαριστίες.....	9
Περιεχόμενα.....	13
Κεφάλαιο 1.....	15
Εισαγωγή	15
Οπτικές Ίνες – Οπτικές Επικοινωνίες – Οπτικά Δίκτυα	16
Τεχνολογία και Δίκτυα WDM – Πολυπλεξία με Διαίρεση Μήκους Κύματος	17
Κεφάλαιο 2.....	19
Τοπολογία WDM Δικτύων Παθητικού Αστέρα	19
Πρωτόκολλα και Πολιτική Πρόσβασης στο μέσο μετάδοσης (Media Access Control Protocols)	20
Η επίδραση της καθυστέρησης διάδοσης (propagation delay).....	20
Το φαινόμενο των συγκρούσεων στο δέκτη (Receiver Collisions Phenomenon).....	21
Κεφάλαιο 3.....	23
Σύγχρονο WDMA Multi Control/Multi Data Σύστημα με Poisson Αφίξεις	23
Αρχιτεκτονική του δικτύου και παραδοχές.....	23
Πολιτική Μετάδοσης (Transmission Policy).....	24
Πολιτική Αποδοχής Πακέτου (Receiving Policy).....	25
Ανάλυση – Αριθμητικά Αποτελέσματα – Γραφικές Παραστάσεις	25
Ανάλυση χωρίς την επίδραση του Receiver Collisions Phenomenon.....	25
Ανάλυση της καθυστέρησης	30
Ανάλυση με την επίδραση του Receiver Collisions Phenomenon	37
Μέσης Πιθανότητας Απόρριψης.....	41
Ανάλυση της καθυστέρησης	45
Κεφάλαιο 4.....	53
Ασύγχρονο WDMA Single Control/Multi Data Σύστημα με Poisson Αφίξεις.....	53
Αρχιτεκτονική του δικτύου και παραδοχές.....	53
Πολιτική Μετάδοσης (Transmission Policy).....	54
Πολιτική Αποδοχής Πακέτου (Receiving Policy).....	55
Ανάλυση – Αριθμητικά Αποτελέσματα – Γραφικές παραστάσεις	55
Ανάλυση χωρίς την επίδραση του Receiver Collisions Phenomenon.....	55
Ανάλυση της καθυστέρησης	57
Ανάλυση με την επίδραση του Receiver Collisions Phenomenon	61
Μέσης Πιθανότητας Απόρριψης.....	62

Ανάλυση της καθυστέρησης	65
Κεφάλαιο 5	71
Μελλοντική έρευνα.....	71
Διαδικασίες Self – Similar.....	71
Κατανομή Pareto για τη γέννηση Self – Similar κίνησης.....	73
Συμπεράσματα	77
Παραρτήματα.....	79
Κώδικας κεφαλαίου 3	79
Κώδικας κεφαλαίου 4	92
Παραπομπές – Βιβλιογραφία.....	105

Κεφάλαιο 1

Εισαγωγή

Το 1880, τέσσερα χρόνια μετά την εφεύρεση του τηλεφώνου, ο Alexander Graham Bell χρησιμοποίησε το φως για να μεταδώσει το λόγο. Ονόμασε τη συσκευή του «Φωτόφωνο».

Πέρασαν σχεδόν 80 χρόνια έως ότου ξεκινήσει η ανάπτυξη των οπτικών ινών και συσκευών κατάλληλων για οπτικές επικοινωνίες (1966). Αρχικά, τα καινούρια αυτά μέσα μετάδοσης παρουσίαζαν πολύ υψηλές απώλειες, αλλά τη δεκαετία του '70 οι ερευνητές και οι μηχανικοί κατάφεραν να κατασκευάσουν οπτικές ίνες με αποδεκτή εξασθένιση.

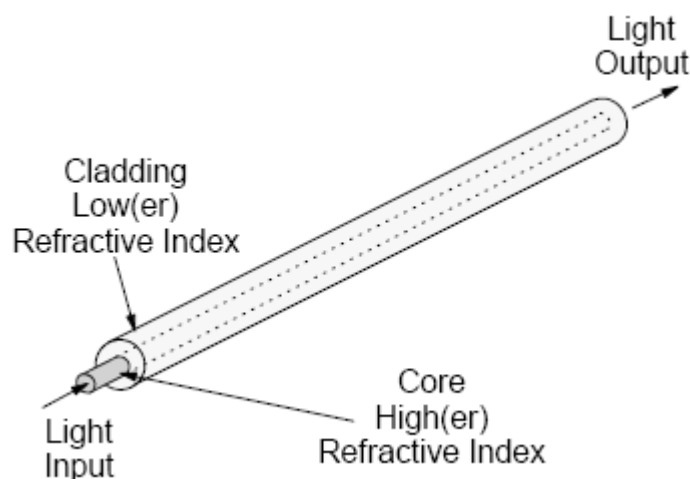


Εικ. 1 Alexander Graham Bell. Το 1880 εφήυρε το «Φωτόφωνο – Photophone», ένα οπτικό τηλεφωνικό σύστημα, το οποίο όμως δεν είχε ανταπόκριση.

Η πραγματική αλλαγή και εμπορική ανάπτυξη των οπτικών δικτύων ήρθε τη δεκαετία του '80, με την ανάπτυξη της τεχνολογίας στον τομέα της φωτονικής και των υλικών. Γρήγορα αποδείχθη ότι οι οπτικές ίνες μπορούσαν να διαθέσουν πολύ περισσότερο εύρος ζώνης απ' ό,τι οποιοδήποτε άλλο μέσο μετάδοσης, είτε αυτό ήταν καλώδιο σε συστρεφόμενο ζεύγος (twisted – pair wire), είτε καλώδιο χαλκού, είτε ασύρματη μετάδοση. Από τότε τα οπτικά δίκτυα αναπτύχθηκαν προς την κατεύθυνση ακόμα πιο συναρπαστικών τεχνολογιών περιλαμβάνοντας οπτικά φίλτρα, οπτικούς πολυπλέκτες και αποπολυπλέκτες και οπτικούς ενισχυτές έτσι ώστε μία μόνο ίνα να μπορεί να μεταφέρει δεδομένα σε ρυθμό που υπερβαίνει την τάξη των Tbps.

Οπτικές Ίνες – Οπτικές Επικοινωνίες – Οπτικά Δίκτυα

Η οπτική ίνα είναι ένα πολύ λεπτό νήμα διοξειδίου του πυριτίου χωρίς προσμίξεις, με τη μορφή γυάλινου κυλίνδρου που καθορίζεται από δύο εξίσου σημαντικούς παράγοντες: (α) η απώλεια φωτός κατά μήκος της ίνας είναι παρά πολύ μικρή, (β) το φως μένει και οδηγείται από την ίνα παρά τις καμπυλώσεις που μπορεί αυτή να παρουσιάσει.



Εικ. 2 Οπτική ίνα. Ο πυρήνας (core) έχει υψηλότερο δείκτη διάθλασης από το μανδύα (cladding).

Οι οπτικές ίνες χρησιμοποιούν υπέρυθρο φως με τα μήκη κύματος κατανεμημένα σε τρία παράθυρα, το παράθυρο των 1550 nm, αυτό των 1300 nm και τέλος το παράθυρο των 850 nm. Τα δύο πρώτα παράθυρα, (1300 nm και 1550 nm) έχουν πολύ μικρές απώλειες και χρησιμοποιούνται για μεταδόσεις σε μεγάλες αποστάσεις, ενώ το παράθυρο των 850 nm παρουσιάζει μεγαλύτερες απώλειες και συναντάται σε LAN, για αποστάσεις περίπου έως 10 km.

Μία άλλη κατηγοριοποίηση των οπτικών ινών είναι η διάκριση σε μονότροπες (single mode fiber) και πολύτροπες ίνες (multi mode fiber). Οι μονότροπες ίνες έχουν μικρή διάμετρο πυρήνα (περίπου 10μm) ενώ στις πολύτροπες ίνες η διάμετρος είναι 50 – 100 μm. Για μεταδόσεις σε μεγάλες αποστάσεις χρησιμοποιούνται μονότροπες ίνες, ενώ αντίστοιχα για μικρών αποστάσεων μεταδόσεις οι πολύτροπες ίνες.

Η διάδοση της χρήσης των οπτικών ινών οδήγησε στην ανάπτυξη συστημάτων επικοινωνίας δεύτερης γενιάς (η πρώτη γενιά περιλαμβάνει δίκτυα που οι ζεύξεις είναι ζεύξεις χαλκού). Τα οπτικά αυτά δίκτυα επιτυγχάνουν καλύτερο εύρος ζώνης και καλύτερο ρυθμό σφαλμάτων και ένα χαρακτηριστικό παράδειγμα είναι το FDDI (Fiber Digital Distributed Interface). Τα συστήματα δεύτερης γενιάς αν και ταχύτερα από αυτά της πρώτης γενιάς παρουσιάζουν αυτό που ονομάζεται «ηλεκτρονική συμφόρηση» (electronic bottleneck) και είναι απόρροια των αναπόφευκτων ηλεκτροοπτικών και οπτικοηλεκτρικών μετατροπών. Εκτός της καθυστέρησης που προκύπτει κατά τη μετατροπή των σημάτων από ηλεκτρικά σε οπτικά και το αντίστροφο, το εύρος ζώνης συχνοτήτων των ηλεκτρονικών είναι κατά πολύ μικρότερο συγκριτικά με εκείνο της οπτικής ίνας.

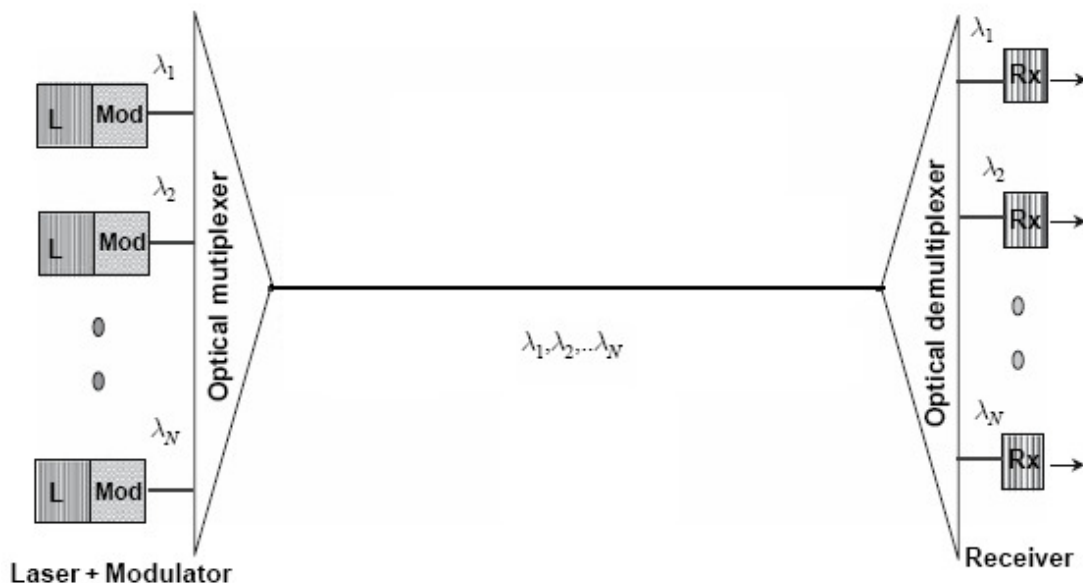
Η τρίτη γενιά δικτύων αποσκοπεί να επιτύχει τη διαθεσιμότητα του εύρους ζώνης, του χαμηλού ρυθμού σφαλμάτων και της ταχύτητας διάδοσης της ίνας κατά μήκος όλης της διαδρομής. Τα δίκτυα αυτά που αξιοποιούν όλα τα χαρακτηριστικά της οπτικής ίνας και όπου οι συνδέσεις δεν διακόπτονται από οπτικοηλεκτρονικές μετατροπές ονομάζονται «αμιγώς – οπτικά» δίκτυα (All – optical Networks).

Τεχνολογία και Δίκτυα WDM – Πολυπλεξία με Διαίρεση Μήκους Κύματος

Η τεχνολογία WDM (πολυπλεξία με διαίρεση μήκους κύματος) είναι η βασική τεχνολογία των οπτικών δικτύων. Η τεχνική της έγκειται στη χρήση μίας ίνας (ή οπτικής συσκευής) για τη δημιουργία πολλών, ξεχωριστών και ανεξάρτητων οπτικών καναλιών.

Ένας τρόπος να φανταστεί κάποιος το WDM είναι να θεωρήσει ότι κάθε κανάλι αποτελείται από φως διαφορετικού χρώματος. Παρόλο που οι συχνότητες που χρησιμοποιούνται ανήκουν στο αόρατο φάσμα, θα μπορούσε κανείς να πει ότι ένα WDM σύστημα μεταδίδει ένα «ουράνιο τόξο».

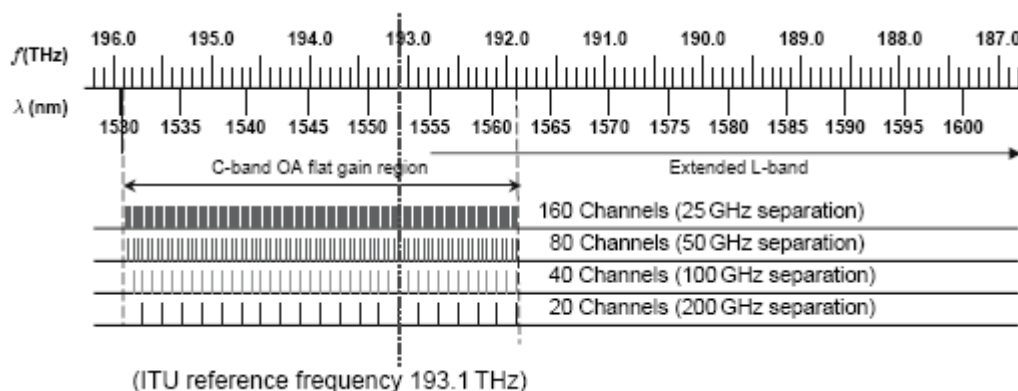
Η τεχνική αυτή επιτρέπει την ταυτόχρονη μετάδοση μέχρι και 1000 διαφορετικών σημάτων και τη μετάδοση προς δύο κατευθύνσεις με μία μόνο οπτική ίνα. Αυτό επιτυγχάνεται μεταδίδοντας σε διαφορετικό μήκος κύματος κάθε ξεχωριστό σήμα και μπορεί να εφαρμοστεί είτε με μονότροπες είτε με πολύτροπες ίνες χρησιμοποιώντας laser ή LEDs.



Εικ. 3 Τα πλεονεκτήματα των WDM δικτύων. N κανάλια (N διαφορετικά μήκη κύματος) με τη χρήση μίας μόνο ίνας.

Οι συχνότητες που απαντώνται στα WDM συστήματα επικοινωνίας ανήκουν στο φασματικό εύρος από 0.8 έως 1.6 μm. Το μήκος κύματος κάθε καναλιού καθορίζεται από τις προδιαγραφές που έχει θέσει η ITU – T (Telecommunication Standardization Sector of the International Telecommunications Union). Για το DWDM (Dense

WDM) το πλέγμα συχνοτήτων καθορίζεται με απόσταση 25, 50, 100 και 200 GHz μεταξύ των καναλιών. Συχνότητα αναφοράς είναι τα 196.10 THz (ή 1528.77 nm). Ο μέγιστος αριθμός καναλιών είναι τα 320 οπτικά κανάλια αλλά θεωρητικά αν χρησιμοποιηθεί το πλήρες φάσμα μπορούμε να έχουμε ως και 1000 κανάλια. Η τεχνική DWDM χρησιμοποιείται σε εφαρμογές μεγαλύτερων αποστάσεων όπου απαιτείται υψηλή απόδοση και υψηλός ρυθμός δεδομένων. Για το CWDM (Coarse WDM) το αντίστοιχο πλέγμα δίνει 18 συχνότητες, από 1280 έως 1640 nm με απόσταση 20 nm μεταξύ των καναλιών. Η τεχνική CWDM χρησιμοποιείται σε εφαρμογές μικρών αποστάσεων όπου σημαντικό ρόλο παίζει το χαμηλό κόστος, όπως για παράδειγμα τα οπτικά LANs [3].



Εικ. 4 Το DWDM πλέγμα συχνοτήτων.

Τα στοιχεία που απαρτίζουν ένα WDMA (wavelength division multiple access) σύστημα είναι [1], [4]:

- laser – εκπομποί (tunable laser transmitters) με την ικανότητα να συντονίζονται στο επιθυμητό μήκος κύματος (οπτικό κανάλι) ή laser – εκπομποί σταθερά συντονισμένοι σε κάποιο μήκος κύματος (fixed laser transmitters)
- οπτικοί πολυπλέκτες και αποπολυπλέκτες
- οπτικοί ενισχυτές
- φίλτρα – δέκτες (tunable filter receivers) με ικανότητα συντονισμού σε διαφορετικούς διαύλους ή φίλτρα – δέκτες σταθερά συντονισμένα σε κάποιο μήκος κύματος (fixed filter receivers) .

Οι παράγοντες που καθορίζουν την απόδοση πρωτοκόλλων σε δίκτυα WDM είναι [1]:

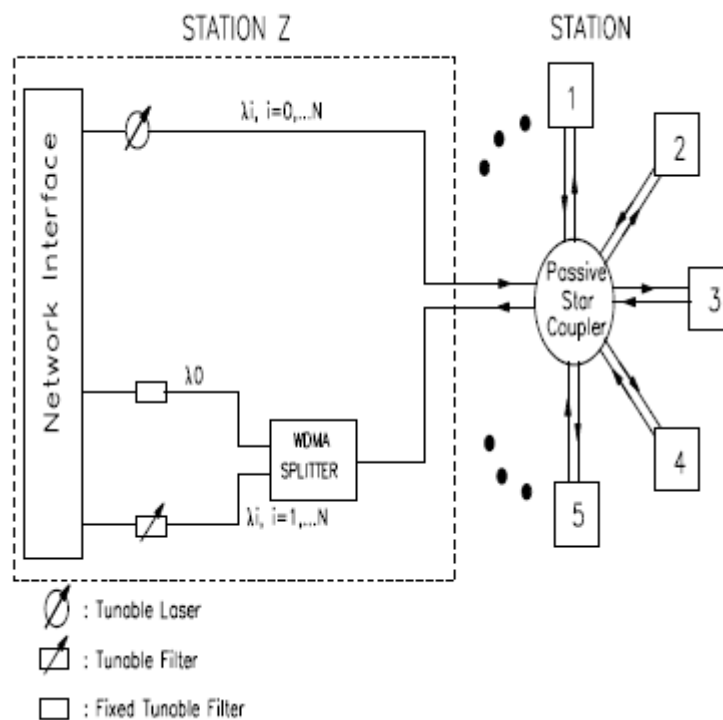
- ο χρόνος συντονισμού (tuning time) των πομποδεκτών
- το εύρος συντονισμού (tuning range) των πομποδεκτών
- οι απαιτήσεις επεξεργασίας
- η καθυστέρηση διάδοσης (propagation delay) σε σχέση με το χρόνο μετάδοσης ενός πακέτου (packet transmission time)
- ο χρόνος αναμονής (waiting time) πριν τη μετάδοση ενός πακέτου
- χρόνος που ενδεχομένως απαιτείται για το συντονισμό του δικτύου
- οι συγκρούσεις στα κανάλια (channel collisions)
- οι συγκρούσεις στον προορισμό (receiver collisions)

Κεφάλαιο 2

Τοπολογία WDM Δικτύων Παθητικού Αστέρα

Η τοπολογία ενός δικτύου ορίζεται ως ο καθορισμός και η σχεδίαση όλων των δυνατών μονοπατιών από οποιονδήποτε εκπομπό προς οποιονδήποτε δέκτη του δικτύου τόσο σε φυσικό όσο και σε λογικό επίπεδο.

Η τοπολογία ΠΑΘΗΤΙΚΟΥ ΑΣΤΕΡΑ περιλαμβάνει ένα οπτικό παθητικό συζεύκτη με τον οποίο συνδέονται όλοι οι σταθμοί που περιλαμβάνει το δίκτυο με διαφορετική οπτική ίνα ο κάθε ένας. Όλα τα εισερχόμενα σήματα συνδυάζονται σε μία ίνα οδηγώντας σε ένα σήμα και εν συνεχεία αυτό το σήμα αποστέλλεται σε όλους τους σταθμούς (broadcast medium). Εν συνεχεία κάθε σταθμός, με τη χρήση ενός διαχωριστή (WDM splitter) λαμβάνει τα διαφορετικά εξερχόμενα από τον passive star coupler σήματα, ενώ με ένα φίλτρο (tunable/fixed filter) δεσμεύει το επιθυμητό γι' αυτόν σήμα. Γι' αυτό και κάθε σταθμός λαμβάνει την ηχώ και των δικών του μεταδόσεων ύστερα από μία χρονική καθυστέρηση, η τιμή της οποίας καθορίζεται από το χρόνο μετάδοσης και το χρόνο διάδοσης της πληροφορίας στο δίκτυο.



Εικ. 5 Η τοπολογία παθητικού αστέρα.

Στα WDM δίκτυα που μελετώνται το σύστημα χρησιμοποιεί ένα ή περισσότερα κανάλια ελέγχου, κοινά για όλους τους σταθμούς με σκοπό τον έλεγχο των μεταδόσεων των πακέτων πληροφορίας (Εικ. 5, τοπολογία με ένα κανάλι ελέγχου) και N κανάλια μετάδοσης πληροφορίας (data channels).

Πρωτόκολλα και Πολιτική Πρόσβασης στο μέσο μετάδοσης (Media Access Control Protocols)

Τα πρωτόκολλα που θα μελετήσουμε διακρίνονται σε δύο βασικές κατηγορίες, 1) τα **σύγχρονα πρωτόκολλα** όπου οι σταθμοί συγχρονίζονται με ένα κοινό ρολόι (common clock) και μπορούν να μεταδώσουν μόνο σε συγκεκριμένα χρονικά πλαίσια, στην αρχή των χρονοσχησμών (time slots) που το κοινό ρολόι ορίζει και 2) τα **ασύγχρονα πρωτόκολλα**, όπου οι πομποί δε συγχρονίζονται με κάποιο ρολόι για την εκπομπή των δεδομένων τους, οπότε είναι ελεύθεροι να προβούν οποιαδήποτε στιγμή σε μετάδοση [1].

Η πολιτική που ακολουθείται είναι η πολιτική “tell and wait”, σύμφωνα με την οποία όταν ένας σταθμός θέλει να στείλει δεδομένα στο δίκτυο, αρχικά ενημερώνει όλους τους υπόλοιπους σταθμούς (“tell”), αποστέλλοντας ένα control packet σε κάποιον control δίαυλο – εάν το σύστημα διαθέτει παραπάνω από ένα control διαύλους – ή στον μοναδικό δίαυλο ελέγχου – στην περίπτωση που το δίκτυο έχει έναν control δίαυλο – και εν συνεχεία περιμένει (“wait”). Αν το control πακέτο του επιτυγχάνει να μεταδοθεί, ο σταθμός προχωράει στη μετάδοση των δεδομένων του (data packet), βάσει του πρωτοκόλλου που ακολουθείται. Για την πρόσβαση στο κανάλι (κανάλια) ελέγχου χρησιμοποιείται random access protocol (Slotted Aloha) [5].

Η επίδραση της καθυστέρησης διάδοσης (propagation delay)

Παρόλο που η ταχύτητα μετάδοσης δεδομένων έχει αυξηθεί κατά πολλές τάξεις μεγέθους, η ταχύτητα του φωτός στην ίνα και η ταχύτητα διάδοσης ηλεκτρικών σημάτων σε ένα καλώδιο δεν έχει αλλάξει. Η ταχύτητα διάδοσης παραμένει η ίδια όπως στα παραδοσιακά LAN και είναι πολύ πιο χαμηλή από την ταχύτητα μετάδοσης των δεδομένων σε ένα οπτικό τηλεπικοινωνιακό σύστημα, με αποτέλεσμα η καθυστέρηση διάδοσης να είναι κατά πολύ υψηλότερη του χρόνου μετάδοσης της πληροφορίας.

Αρχικά μελετάται η επίδραση της υψηλής καθυστέρησης διάδοσης σε σύγχρονο WDMA δίκτυο με V κανάλια ελέγχου και N δεδομένων. Το propagation delay latency αποδεικνύεται καταλυτικός παράγοντας στην παράμετρο της καθυστέρησης μετάδοσης των πακέτων πληροφορίας.

Εν συνεχεία, στη μελέτη Ασύγχρονου WDMA δικτύου με έναν control δίαυλο και N κανάλια δεδομένων, προτείνεται πρωτόκολλο που εκμεταλλεύεται την υψηλή καθυστέρηση διάδοσης με σκοπό την ανάπτυξη αλγορίθμων που δίνουν τη δυνατότητα στους σταθμούς να γνωρίζουν την κατάσταση των καναλιών δεδομένων (αν το κανάλι επιλογής κάποιου σταθμού θα είναι ελεύθερο ή κατειλημμένο) μετά το πέρας του propagation delay και αφού έχουν επιτυχώς μεταδώσει το αντίστοιχο control packet τους στο δίαυλο ελέγχου. Η γνώση αυτή εξασφαλίζει ότι σε κάθε κανάλι θα μεταδώσει πακέτο δεδομένων μονάχα ένας σταθμός, αποφεύγοντας έτσι τις συγκρούσεις στα κανάλια δεδομένων και βελτιώνοντας της απόδοση (throughput) του συστήματος [6].

Το φαινόμενο των συγκρούσεων στο δέκτη (Receiver Collisions Phenomenon)

Όπως έχει αναφερθεί και στο κεφάλαιο 1, (βλ. σελ. 18), οι σταθμοί που είναι συνδεδεμένοι στο δίκτυο διαθέτουν φίλτρα δέκτες με την ικανότητα να συντονίζονται στα κανάλια δεδομένων έτσι ώστε οι σταθμοί να λαμβάνουν πακέτα δεδομένων που προορίζονται γι' αυτούς. Εάν κατά τη διάρκεια παραλαβής ενός πακέτου δεδομένων από το δέκτη ενός σταθμού, φθάσει στο σταθμό ένα άλλο πακέτο δεδομένων, το πακέτο αυτό δε θα μπορέσει να παραληφθεί από το σταθμό – δέκτη, παρόλο που έχει καταφέρει να μεταδοθεί επιτυχώς και θα πρέπει να επανεκπεμφθεί. Το φαινόμενο αυτό της απόρριψης ενός data πακέτου στο δέκτη αποτελεί το receiver collisions phenomenon.

Στα κεφάλαιο 3 και 4 μελετάται η επίδραση του φαινομένου των συγκρούσεων στο δέκτη στις κρίσιμες παραμέτρους του δικτύου, throughput και delay. Οι αλγόριθμοι που προτείνονται αντιμετωπίζουν το receiver collisions phenomenon. Συγκεκριμένα, οι σταθμοί γνωρίζουν όχι μόνο την κατάσταση των καναλιών δεδομένων αλλά και την κατάσταση του δέκτη του σταθμού στον οποίο θέλουν να στείλουν δεδομένα τη στιγμή που το data πακέτο τους θα φθάσει στο σταθμό – προορισμό.

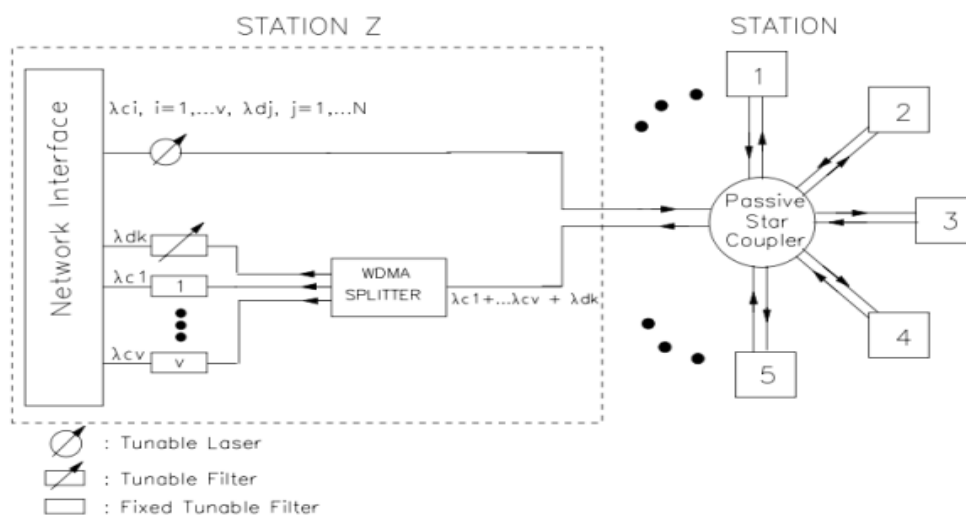
Κεφάλαιο 3

Σύγχρονο WDMA Multi Control/Multi Data Σύστημα με Poisson Αφίξεις

Αρχιτεκτονική του δικτύου και παραδοχές

Το σύστημα που μελετάται είναι ένα WDMA σύστημα σε τοπολογία παθητικού αστέρα. Αρχικά θεωρούμε ότι ο αστέρας εξυπηρετεί άπειρο αριθμό σταθμών (Μελέτη αγνοώντας της επίδραση του Receiver Collisions Phenomenon) και εν συνεχεία μελετάμε την περίπτωση όπου η προσφερόμενη κίνηση προέρχεται από πεπερασμένο αριθμό σταθμών, M (βλ. και Μελέτη της επίδρασης του Receiver Collisions Phenomenon). Η προσφερόμενη κίνηση ακολουθεί την κατανομή Poisson. Οι σταθμοί που είναι συνδεδεμένοι στο σύστημα διαθέτουν buffers μοναδιαίας χωρητικότητας. Αν ένα πακέτο αποτύχει να εκπεμφθεί αποθηκεύεται στον buffer και επανεκπέμπεται ανάλογα με την πολιτική επανεκπομπής. Η τοπολογία και οι παραδοχές του συστήματος ακολουθούν το [8].

Χρησιμοποιούνται συνολικά $(V+N)$ μήκη κύματος εκ των οποίων τα V μήκη κύματος διατίθενται για μετάδοση πληροφορίας ελέγχου (control channels) – σημειογραφία CC^V – και τα N μήκη κύματος διατίθενται για μετάδοση δεδομένων (data channels). Κάθε κόμβος διαθέτει έναν συντονιζόμενο laser εκπομπό για να συντονίζεται σε κάποιο κανάλι ελέγχου για μετάδοση control πληροφορίας ή σε κάποιο κανάλι δεδομένων για μετάδοση data πακέτου – σημειογραφία TT . Επίσης κάθε κόμβος διαθέτει V σταθερά συντονισμένους δέκτες έτσι ώστε να λαμβάνει όλη την control πληροφορία από τα V κανάλια ελέγχου – FR^V – και ένα φίλτρο δέκτη με ικανότητα συντονισμού στο κανάλι επιλογής του σταθμού – TR – για να λαμβάνει δεδομένα που στέλλονται από τους άλλους σταθμούς και έχουν αυτόν ως τελικό αποδέκτη. Έτσι το σύστημα περιγράφεται ως ένα $CC^V - TT - FR^V - TR$ σύστημα [8].



Εικ. 6 Η τοπολογία παθητικού αστέρα. Τα κανάλια λ_{ci} $i=1, \dots, V$ είναι τα κανάλια ελέγχου και τα κανάλια λ_{dj} $j=1, \dots, N$ είναι τα κανάλια δεδομένων.

Το μέγεθος των control πακέτων θεωρείται σταθερό και ο χρόνος μετάδοσης ενός control πακέτου λαμβάνεται ως χρονική μονάδα αναφοράς (Transmission time of a control packet= 1 time unit). Το μέγεθος των data πακέτων θεωρείται επίσης σταθερό και ο χρόνος μετάδοσης ενός πακέτου δεδομένων σε κανονικοποιημένες μονάδες χρόνου θεωρείται ίσος με L time units. Ο χρόνος διάδοσης από οποιονδήποτε σταθμό στο συζεύκτη του παθητικού αστέρα και εν συνεχεία σε οποιονδήποτε σταθμό θεωρείται σταθερός και ίδιος για όλους τους σταθμούς του συστήματος, ίσος με $R \times L$ time units. Ορίζουμε ως cycle το χρονικό διάστημα που περιλαμβάνει μία χρονική μονάδα για τη μετάδοση του control πακέτου, $R \times L$ χρονικές μονάδες για τη διάδοση του control πακέτου και L χρονικές μονάδες για τη μετάδοση ενός data πακέτου, δηλαδή, $1 \text{ cycle} = 1 + (R+1) \times L$ time units. Ο άξονας του χρόνου είναι διαιρεμένος σε συνεχή cycles, που αποτελούν κοινή αναφορά για τα control και data κανάλια.

Επίσης θεωρούμε ότι οι χρόνοι συντονισμού των tunable transmitters και receivers καθώς και ο χρόνος που απαιτείται για την επεξεργασία της control πληροφορίας από τους σταθμούς είναι αμελητέοι.

Οι πληροφορίες που περιέχονται σε κάθε control πακέτο είναι η διεύθυνση του αποστολέα, η διεύθυνση του παραλήπτη καθώς και το κανάλι δεδομένων που έχει επιλέξει ο σταθμός για αποστολή των δεδομένων του [8].

CONTROL PACKET

T	R	λ_k
---	---	-------------

Εικ. 7 Το control πακέτο.

T → διεύθυνση αποστολέα

R → διεύθυνση παραλήπτη

λ_k → το μήκος κύματος του καναλιού
δεδομένων

Πολιτική Μετάδοσης (Transmission Policy)

Όταν ένας σταθμός θέλει να μεταδώσει ένα πακέτο δεδομένων προς έναν άλλο σταθμό, αρχικά επιλέγει τυχαία ένα κανάλι δεδομένων λ_{dj} , $j=1, \dots, N$ στο οποίο θα μεταδώσει το πακέτο του. Στη συνέχεια επιλέγει τυχαία ένα κανάλι ελέγχου, λ_{ci} , $i=1, \dots, V$ για να στείλει το πακέτο ελέγχου που ενημερώνει όλους τους υπόλοιπους σταθμούς για την επικείμενη μετάδοση data πακέτου. Ο σταθμός περιμένει την έναρξη του αμέσως επόμενου cycle και μεταδίδει το control packet του στο πρώτο time unit αυτού του cycle, που συναγωνίζεται με τα control packets των υπολοίπων σταθμών βάσει των κανόνων του πρωτοκόλλου Slotted Aloha για να κερδίσει πρόσβαση σε κάποιο control κανάλι.

Ο σταθμός ενημερώνεται για την έκβαση του control πακέτου του, καθώς για την έκβαση των πακέτων ελέγχου άλλων σταθμών, μετά το πέρας $R \times L$ χρονικών μονάδων, (περίοδος επιβεβαίωσης, η διάρκεια του χρόνου διάδοσης) ακούγοντας ταυτόχρονα τα V control κανάλια με τους V fixed tuned receivers που διαθέτει. Έτσι με το πέρας αυτής της περιόδου είναι σε θέση να γνωρίζει τόσο αν το κανάλι δεδομένων που έχει επιλέξει θα είναι ελεύθερο για μετάδοση, όσο και το αν ο δέκτης του σταθμού στον οποίο προορίζει τα δεδομένα του θα είναι ανενεργός μετά το πέρας των επόμενων $(R+1) \times L$ time units (L time units για τη μετάδοση και $R \times L$ time units για τη διάδοση του data πακέτου). Σε περίπτωση που το επιλεγμένο κανάλι

δεδομένων είναι επιλογή και άλλων σταθμών, επιλέγεται τυχαία ένας σταθμός για να μεταδώσει με την προϋπόθεση ότι αυτός ο σταθμός θα βρει ανενεργό τον δέκτη του σταθμού – προορισμού. Κατ' αυτόν τον τρόπο δεν έχουμε συγκρούσεις ούτε στα κανάλια δεδομένων αλλά ούτε και στον προορισμό.

Οι σταθμοί που είτε το control πακέτο τους απέτυχε να μεταδοθεί, είτε δεν κέρδισαν πρόσβαση σε κάποιο κανάλι δεδομένων είτε τελικά δε μεταδίδουν επειδή το πακέτο τους θα απορριφθεί από το δέκτη, θεωρούνται backlogged σταθμοί [8] και θα προσπαθήσουν ξανά να μεταδώσουν το πακέτο τους.

Πολιτική Αποδοχής Πακέτου (Receiving Policy)

Όλοι οι σταθμοί παρακολουθούν συνεχώς τι συμβαίνει στα control κανάλια. Ένας σταθμός που αναγνωρίζει τη διεύθυνσή του σε ένα επιτυχώς μεταδιδόμενο control πακέτο (πεδίο R του control πακέτου, βλ. και Εικ. 7), αρχικά εξετάσει αν ο σταθμός που έστειλε το πακέτο αυτό θα καταφέρει να μεταδώσει τα δεδομένα του σε κάποιο data κανάλι. Στη συνέχεια και εφόσον ξέρει ότι θα δεχθεί πακέτο δεδομένων, συντονίζει το φίλτρο – δέκτη του στο μήκος κύματος που αναφέρεται στο control πακέτο μετά το πέρας $(R+1) \times L$ time units από τη στιγμή που «άκουσε» την πληροφορία και λαμβάνει το data πακέτο.

Ανάλυση – Αριθμητικά Αποτελέσματα – Γραφικές Παραστάσεις

Ανάλυση χωρίς την επίδραση του Receiver Collisions Phenomenon

Ως χρονική μονάδα αναφοράς χρησιμοποιείται ο χρόνος μετάδοσης του control πακέτου:

$T_C = 1$ time unit → χρόνος μετάδοσης του control πακέτου

$T_D = L$ time units → χρόνος μετάδοσης του πακέτου δεδομένων
(L time units = 1 data slot)

$T_r = R \times L$ time units → χρόνος διάδοσης

$C = 1 + L + T_r = 1 + (R+1) \times L$ time units → η διάρκεια ενός κύκλου του συστήματος.

V = το πλήθος των καναλιών ελέγχου

N = το πλήθος των καναλιών δεδομένων

G = το μέσο φορτίο ανά κύκλο, (control packets/cycle).

Η μετάδοση των πακέτων ελέγχου γίνεται πάντα στο πρώτο time unit κάθε κύκλου. Όταν ένας σταθμός θέλει να εκπέμψει σε κάποιο control δίαυλο, επιλέγει τυχαία και ισοπίθανα ένα κανάλι ανάμεσα στους V διαύλους ελέγχου με σταθερή πιθανότητα, $\frac{1}{V}$.

Άρα το μέσο φορτίο που ένα κανάλι ελέγχου δέχεται ανά κύκλο είναι:

$$G_C = \frac{G}{V} \quad (\text{control packets/cycle})$$

Η μετάδοση σε κάθε κανάλι ελέγχου ακολουθεί το Slotted Aloha Protocol [2], οπότε το throughput κάθε καναλιού ελέγχου είναι:

$$S_c = \frac{G}{V} e^{-\frac{G}{V}} \quad (3.1)$$

και το συνολικό throughput απ' όλα τα κανάλια ελέγχου είναι:

$$S_{Ctotal} = V \frac{G}{V} e^{-\frac{G}{V}} \rightarrow S_{Ctotal} = G e^{-\frac{G}{V}} \quad (3.2)$$

Άρα κατά μέσο όρο, σε κάθε cycle, $G e^{-\frac{G}{V}}$ πακέτα ελέγχου καταφέρνουν να μεταδοθούν επιτυχώς και συνεπώς $G e^{-\frac{G}{V}}$ σταθμοί ανταγωνίζονται για να κερδίσουν πρόσβαση στα κανάλια δεδομένων. Βάσει του πρωτοκόλλου [8], αν ένα κανάλι δεδομένων επιλεγεί από 1 ή περισσότερους σταθμούς, τότε επιλέγεται τυχαία το πακέτο κάποιου από αυτούς τους σταθμούς και μεταδίδεται. Αρκεί ένα κανάλι να επιλεγεί για να σημειωθεί μία επιτυχής μετάδοση. Άρα το throughput του συστήματος καθορίζεται από το πλήθος των καναλιών δεδομένων που θα επιλεγθούν για μετάδοση.

Έστω k το πλήθος των σταθμών που επιχειρούν μετάδοση στα N κανάλια δεδομένων σε έναν κύκλο. Οι k αυτοί σταθμοί επιλέγουν κανάλι προς μετάδοση, δίνοντας N^k διαφορετικές διατάξεις. Αν ένα κανάλι δεν επιλεγεί από κανένα σταθμό, τότε τα k πακέτα διανέμονται στα $N-1$ εναπομείναντα κανάλια, δίνοντας $(N-1)^k$ διαφορετικές διατάξεις. Συνεπώς η πιθανότητα ένα κανάλι να μείνει κενό είναι [9]:

$$P_0(k) = \frac{(N-1)^k}{N^k} = \left(1 - \frac{1}{N}\right)^k \quad (3.3)$$

και η συμπληρωματική της, δηλαδή η πιθανότητα να επιλεγεί από έναν ή παραπάνω σταθμούς είναι:

$$P_{Dsuc}(k) = 1 - \left(1 - \frac{1}{N}\right)^k \quad (3.4)$$

Οι σταθμοί που επιχειρούν να μεταδώσουν στα κανάλια δεδομένων είναι όσοι έχουν επιτύχει σε κάποιο control διάυλο, άρα στη μόνιμη κατάσταση $k=S_{Ctotal} = G e^{-\frac{G}{V}}$.

Συνεπώς η πιθανότητα επιλογής ενός data channel από έναν ή παραπάνω σταθμούς στη μόνιμη κατάσταση:

$$(3.4) \xrightarrow{k=S_{Ctotal}} P_{Dsuc}(S_{Ctotal}) = 1 - \left(1 - \frac{1}{N}\right)^{S_{Ctotal}} \quad (3.5)$$

Από (3.5), η πιθανότητα να επιλεγθούν n κανάλια από N κανάλια δεδομένων είναι:

$$P_{Dn} = \binom{N}{n} P_{Dsuc}^n (1 - P_{Dsuc})^{N-n} \quad (3.6)$$

Η επιλογή ενός καναλιού δεδομένων (ανεξάρτητα από πόσους σταθμούς θα επιλεγεί) συνεπάγεται και επιτυχή μετάδοση ενός πακέτου δεδομένων. Αν η μεταβλητή $D_{channels}$ εκφράζει το πλήθος των καναλιών δεδομένων που επιλέγονται για μετάδοση με:

$$P(D_{channels} = n) = \binom{N}{n} P_{Dsuc}^n (1 - P_{Dsuc})^{N-n} \quad (3.7)$$

τότε η μέση τιμή της δίνει το throughput του συστήματος (μέση τιμή των επιτυχώς μεταδιδόμενων πακέτων δεδομένων σε ένα cycle).

Η μέση τιμή της μεταβλητής $D_{channels}$ είναι [9], [11]:

$$E[P(D_{channels} = n)] = \sum_{n=1}^N n \binom{N}{n} P_{Dsuc}^n (1 - P_{Dsuc})^{N-n} = NP_{Dsuc}$$

Οπότε:

$$S_{total} = \frac{L}{C} E[D_{channels}] \xrightarrow{[9]} S_{total} = \frac{L}{C} NP_{Dsuc} \quad (3.8)$$

Ο μέσος όρος των πακέτων που μεταδίδονται επιτυχώς ανά cycle ανά κανάλι δεδομένων είναι:

$$\frac{S_{total}}{N} = S_D = \frac{L NP_{Dsuc}(S_{Ctotal})}{C N} \rightarrow S_D = \frac{L}{C} P_{Dsuc}(S_{Ctotal})$$

$$\xrightarrow{(3.5)} S_D = \frac{L}{C} \left(1 - \left(1 - \frac{1}{N} \right)^{S_{Ctotal}} \right) \xrightarrow{(3.2)}$$

$$S_D = \frac{L}{C} \left(1 - \left(1 - \frac{1}{N} \right)^{Ge^{-\frac{G}{V}}} \right) \quad (3.9)$$

Στα γραφήματα που ακολουθούν αναπαρίσταται το throughput συναρτήσει του προσφερόμενου φορτίου, για V σταθερό και κυμαινόμενο N και για κυμαινόμενο V και σταθερό N .

➤ Κυμαινόμενο N σταθερό V

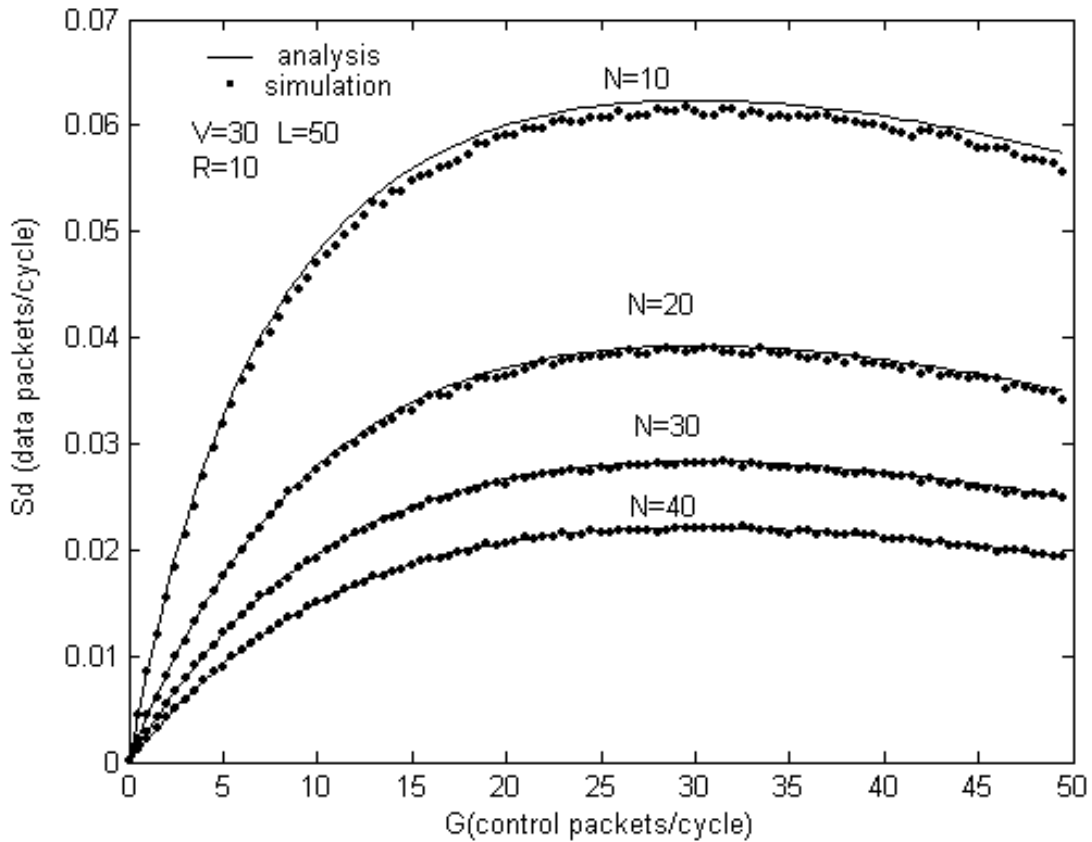


Chart 3. 1: The throughput per cycle per channel, S_D (data packets/cycle) versus the offered load, G (control packets/cycle) for $N=\{10,20,30,40\}$ data channels, $V=30$ control channels and $C=551$ time units ($L=50$ time units and $T_r=500$ time units).

Στην περίπτωση που το πλήθος των control channels είναι σταθερό (Chart 3.1, $V=30$ control channels), αύξηση των data channels οδηγεί σε μείωση του throughput ανά κανάλι δεδομένων.

Το πλήθος των καναλιών ελέγχου είναι σταθερό, οπότε η μέση τιμή των σταθμών που επιτυγχάνουν και κερδίζουν πρόσβαση στα κανάλια δεδομένων για κάποια συγκεκριμένη τιμή του φορτίου είναι επίσης σταθερή (βλ. και σχέση (3.2) $S_{Ctotal} = Ge^{-\frac{G}{V}}$). Οπότε όσο περισσότερα είναι τα κανάλια δεδομένων (με δεδομένο φορτίο) τόσο μειώνεται η πιθανότητα να επιλεγεί κάποιο κανάλι για μετάδοση (πιθανότητα επιλογής ενός καναλιού δεδομένων $1/N$), οπότε μειώνεται και το throughput/data

channel (βλ. και σχέση (3.9) $\rightarrow S_D = \frac{L}{C} \left(1 - \left(1 - \frac{1}{N} \right) Ge^{-\frac{G}{V}} \right)$.

Το συνολικό όμως throughput του συστήματος

$$S_{total} = \frac{L}{C} NP_{Dsuc} \xrightarrow{(3.2) \text{ και } (3.5)} S_{total} = \frac{L}{C} N \left(1 - \left(1 - \frac{1}{N} \right)^{Ge^{-\frac{G}{V}}} \right) \quad (3.10)$$

αυξάνεται με την αύξηση του πλήθους των καναλιών δεδομένων:

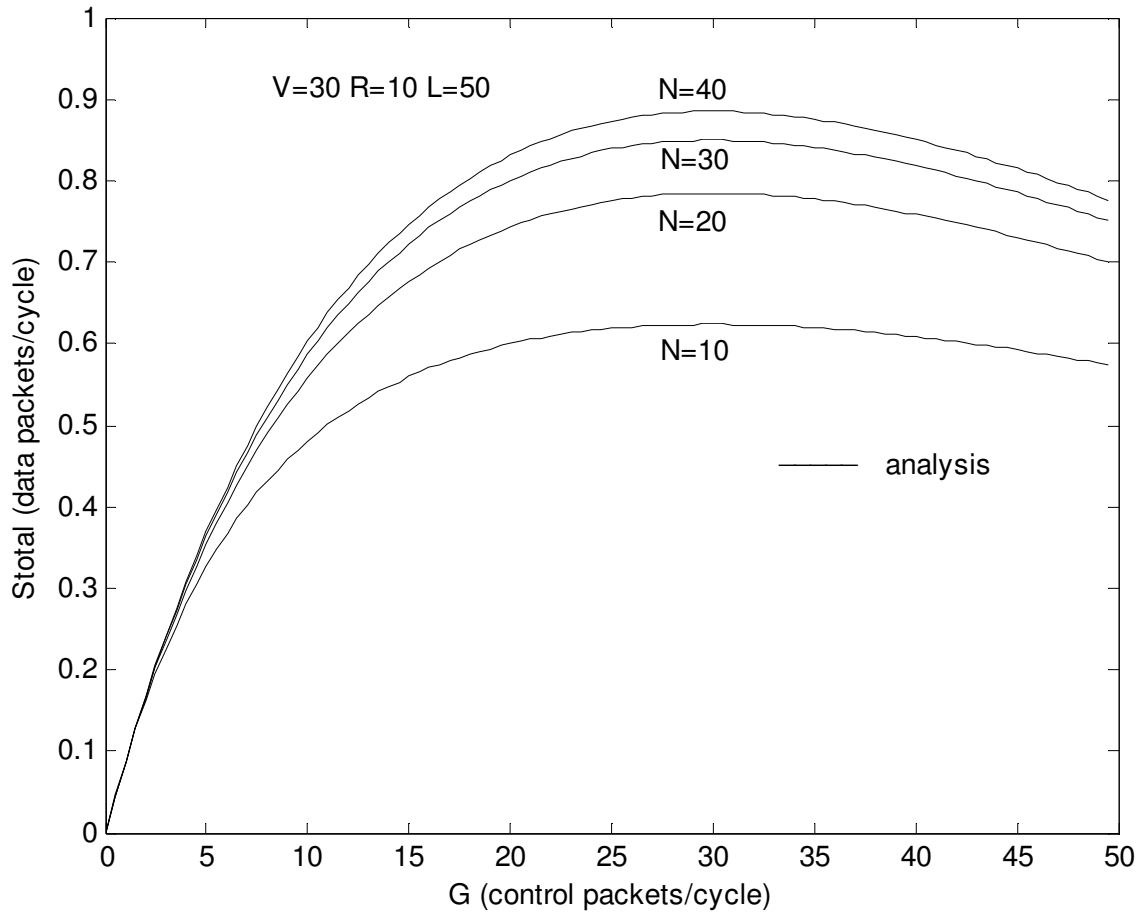


Chart 3.2: The total throughput, S_{total} (data packets/cycle) versus the offered load, G (control packets/cycle) for $N = \{10, 20, 30, 40\}$ data channels, $V=30$ control channels, $C=551$ time units ($L=50$ time units and $R=10$ data slots).

Οι σταθμοί που επιτυγχάνουν στους control διαύλους επιλέγουν τυχαία ένα κανάλι δεδομένων για να εκπέμψουν το data πακέτο τους. Με περισσότερα κανάλια αυξάνεται το εύρος επιλογής των σταθμών, αυξάνεται το πλήθος των καναλιών που επιλέγονται για εκπομπή ((3.8) $\rightarrow E[D_{channels}] = NP_{Dsuc}$, αύξουσα συνάρτηση του N) και συνεπώς αυξάνονται και τα data πακέτα που μεταδίδονται.

➤ V κυμαινόμενο, N σταθερό

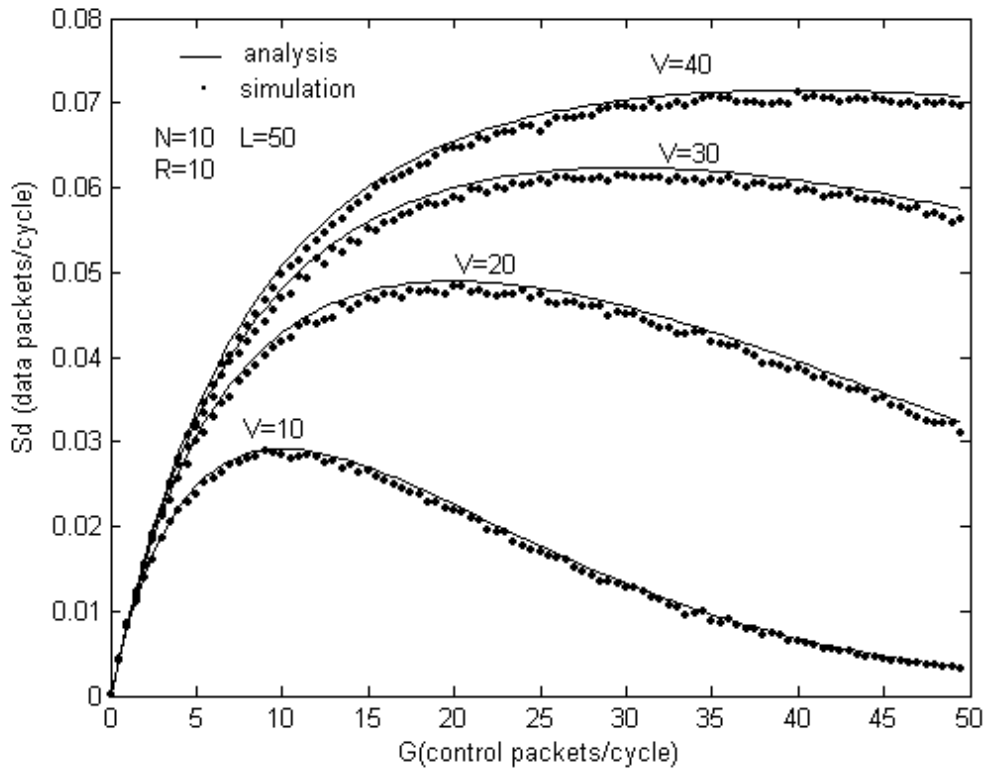


Chart 3.3: The throughput per cycle per channel, S_D (data packets/cycle) versus the offered load, G (control packets/cycle) for $V=\{10,20,30,40\}$ control channels, $N=10$ data channels and $C=551$ time units ($L=50$ time units and $R=10$ data slots).

Με σταθερό το πλήθος των καναλιών δεδομένων, το throughput αυξάνεται όσο αυξάνεται το πλήθος των καναλιών ελέγχου. Όσο περισσότερα κανάλια ελέγχου διατίθενται τόσο περισσότεροι σταθμοί καταφέρνουν να κερδίσουν πρόσβαση και τελικά να μεταδώσουν στα data channels (chart 3.3).

Ανάλυση της καθυστέρησης

Η καθυστέρηση ορίζεται ως το χρονικό διάστημα μεταξύ της γέννησης ενός πακέτου και της επιτυχούς μετάδοσής του στο δίαυλο δεδομένων, [1] και [5].

$$D = D_w + D_r + C + R \times L \quad (3.11)$$

$C + R \times L = 1 + L + 2 \times T_r = 1 + (2R + 1) \times L$ time units, ο χρόνος που απαιτείται για τη μετάδοση και διάδοση του control packet, συν το χρόνο μετάδοσης και διάδοσης του πακέτου δεδομένων (3.12)

D_w = η καθυστέρηση από τη στιγμή της γέννησης ενός πακέτου έως την αρχή του επόμενου cycle, με μέση τιμή:

$$E[D_w] = \frac{1}{c} \int_0^c t dt \rightarrow E[D_w] = \frac{c}{2} \text{ time units} \quad (3.13)$$

D_r = η καθυστέρηση από τη χρονική στιγμή που μεταδίδεται το control packet για πρώτη φορά έως την έναρξη του cycle που ο σταθμός καταφέρνει να αποστείλει επιτυχώς το πακέτο δεδομένων του.

Έστω ότι ένας σταθμός καταβάλλει n προσπάθειες για να επιτύχει να μεταδώσει ένα πακέτο. Υποθέτοντας ότι η πιθανότητα επιτυχούς μετάδοσης του πακέτου είναι ίδια για κάθε προσπάθεια, η πιθανότητα επιτυχίας μετά από n προσπάθειες είναι:

$$Q(n) = P_{suc}(1 - P_{suc})^{n-1} \quad (3.14)$$

με μέση τιμή προσπαθειών:

$$A = E[n] = \sum_{n=1}^{\infty} nQ(n) = P_{suc}^{-1} \quad (3.15)$$

όπου P_{suc} είναι η πιθανότητα επιτυχούς μετάδοσης ενός πακέτου δεδομένων.

$$\begin{aligned} \text{Ισχύει: } S_{total} &= \frac{L}{c} G P_{suc} \rightarrow P_{suc} = \frac{c S_{total}}{LG} \xrightarrow{(3.8)} P_{suc} = \frac{c^L N P_{Dsuc}}{LG} \rightarrow P_{suc} = \frac{N P_{Dsuc}}{G} \\ \xrightarrow{(3.5),(3.2)} P_{suc} &= \frac{N \left(1 - \left(1 - \frac{1}{N} \right) G e^{-\frac{G}{V}} \right)}{G} \quad (3.16) \text{ και } P_{suc}^{-1} = \frac{G}{N \left(1 - \left(1 - \frac{1}{N} \right) G e^{-\frac{G}{V}} \right)} \quad (3.17) \end{aligned}$$

$$(3.15) \xrightarrow{(3.17)} A = \frac{G}{N \left(1 - \left(1 - \frac{1}{N} \right) G e^{-\frac{G}{V}} \right)} \quad (3.18)$$

Στις A προσπάθειες που καταβάλλει ο σταθμός για να επιτύχει μετάδοση, οι πρώτες $A-1$ είναι προφανώς αποτυχημένες. Σε κάθε αποτυχημένη προσπάθεια καταναλίσκονται $1+T_r$ time units (1 time unit για μετάδοση του control packet T_r time units για τη διάδοση του control packet).

Οπότε η μέση τιμή της καθυστέρησης D_r προκύπτει:

$$\begin{aligned} E[D_r] &= (A - 1)[1 + T_r] \\ \xrightarrow{(3.18)} E[D_r] &= \left(\frac{G}{N \left(1 - \left(1 - \frac{1}{N} \right) G e^{-\frac{G}{V}} \right)} - 1 \right) (1 + T_r) \quad (3.19) \end{aligned}$$

Από τις σχέσεις (3.11), (3.12), (3.13) και (3.19) προκύπτει η μέση τιμή της καθυστέρησης:

$$E[D] = E[D_w + D_r + C + R \times L] = E[D_w] + E[D_r] + C + R \times L \rightarrow$$

$$E[D] = \frac{C}{2} + \left(\frac{G}{N \left(1 - \left(1 - \frac{1}{N} \right)^{G e^{-\frac{G}{V}}} \right)} - 1 \right) (1 + T_r) + 1 + (2R + 1) \times L \quad (3.20)$$

Στα γραφήματα που ακολουθούν αναπαρίσταται η μέση καθυστέρηση συναρτήσει του προσφερόμενου φορτίου, για V σταθερό και κυμαινόμενο N και για κυμαινόμενο V και σταθερό N .

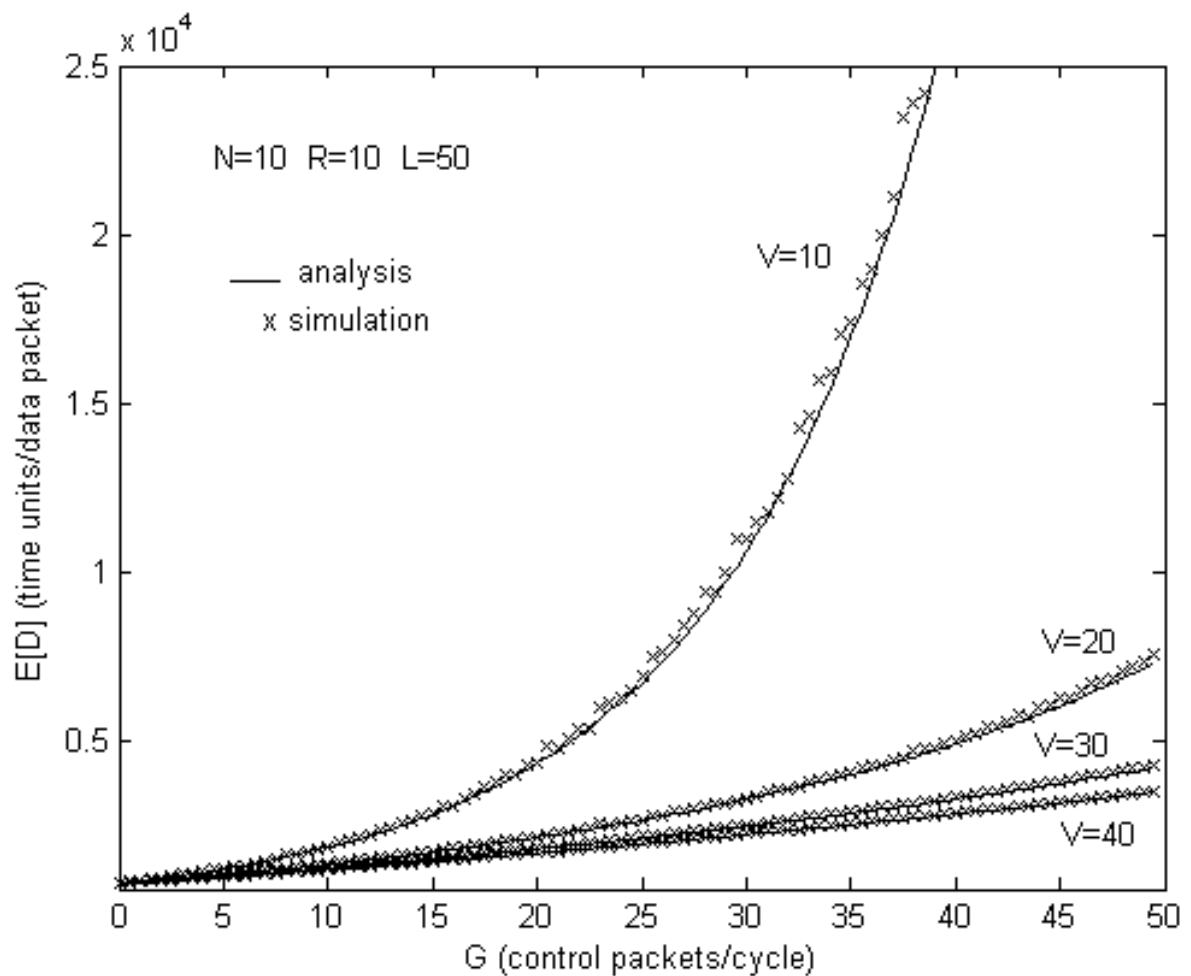


Chart 3.4: The Average Delay per packet, $E[D]$ (time units/packet) versus the offered load, G (control packets/cycle) for $V=\{10,20,30,40\}$ control channels, $N=10$ data channels and $C=551$ time units ($L=50$ time units and $R=10$ data slots).

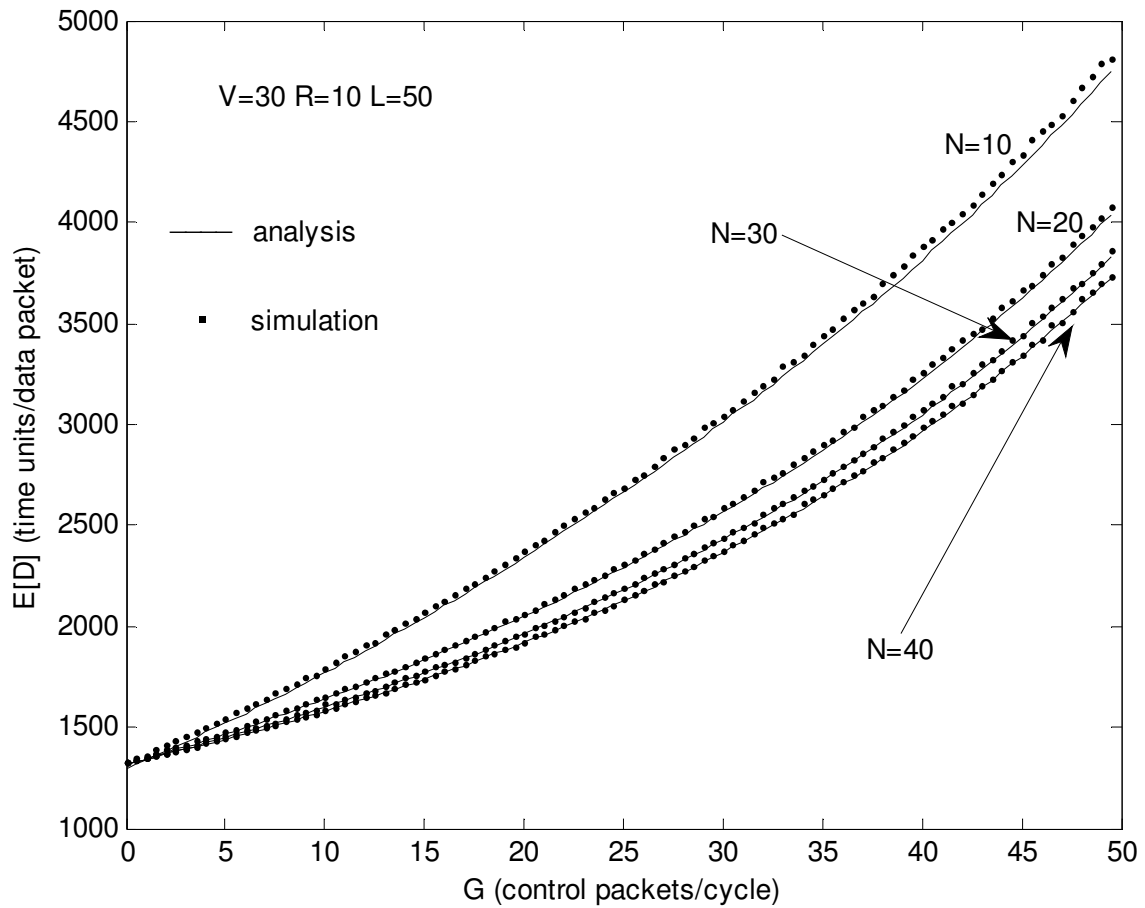


Chart 3.5: The Average Delay per packet, $E[D]$ (time units/packet) versus the offered load, G (control packets/cycle) for $N=\{10,20,30,40\}$ data channels, $V=3$ control channels and $C=551$ time units ($L=50$ time units and $R=10$ data slots).

Η μέση καθυστέρηση μειώνεται καθώς αυξάνεται το πλήθος των καναλιών (ελέγχου και δεδομένων). Όσο περισσότερα κανάλια ελέγχου διατίθενται τόσο μειώνονται οι συγκρούσεις και συνεπώς και η μέση καθυστέρηση. Επίσης, όσο αυξάνεται το πλήθος των καναλιών δεδομένων τόσο μειώνεται η πιθανότητα απόρριψης ενός πακέτου από το κανάλι επιλογής του αντίστοιχου σταθμού, εξαιτίας της κατάληξης του καναλιού αυτού από κάποιο άλλο πακέτο. Άρα οδηγούμαστε σε μείωση της καθυστέρησης (γραφήματα 3.4 και 3.5).

Στα γραφήματα που ακολουθούν αναπαρίσταται η μέση καθυστέρηση ανά πακέτο δεδομένων συναρτήσει του throughput ανά κανάλι δεδομένων, για V σταθερό και κυμαινόμενο N και για κυμαινόμενο V και σταθερό N .

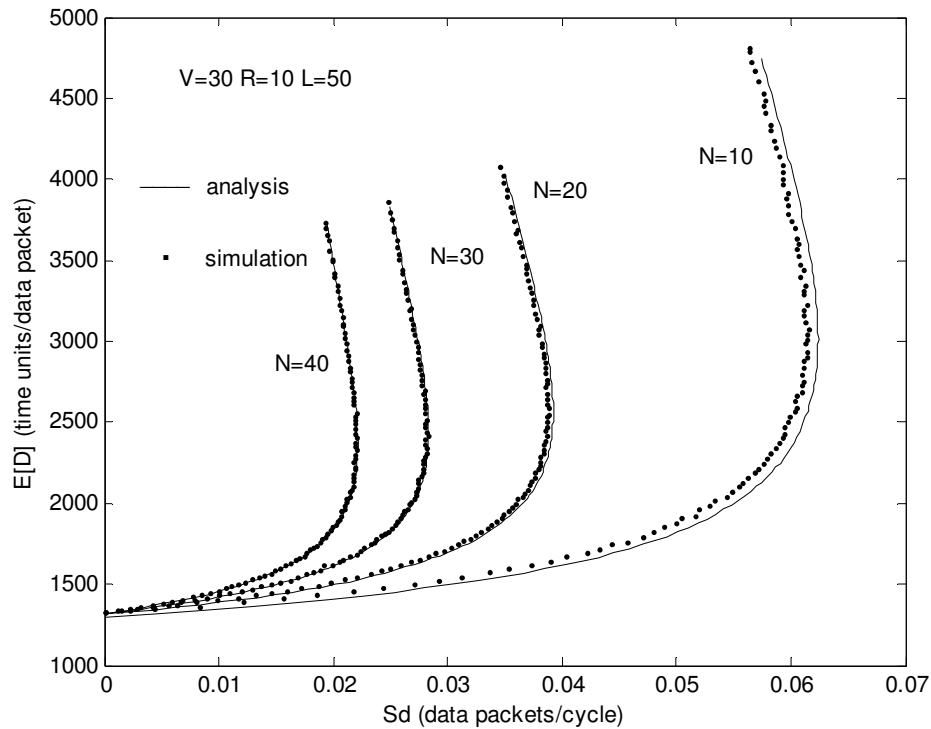


Chart 3.6: The Average Delay per packet, $E[D]$ (time units/ data packet) versus the throughput per data channel (data packets/cycle) for $N=\{10,20,30,40\}$ data channels, $V=30$ control channels and $C=551$ time units ($L=50$ time units and $R=10$ data slots).

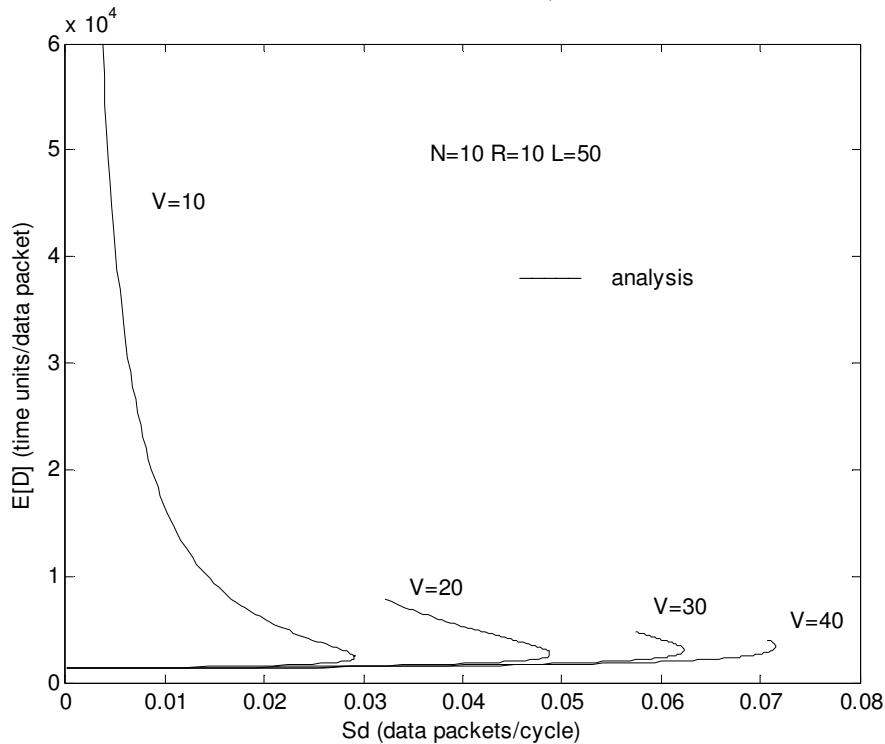


Chart 3.7: The Average Delay per packet, $E[D]$ (time units/ data packet) versus the throughput per cycle per data channel (data packets/cycle) for $N=10$ data channels, $V=\{10, 20, 30, 40\}$ control channels and $C=551$ time units ($L=50$ time units and $R=10$ data slots).

Ακολουθούν διαγράμματα της καθυστέρησης συναρτήσει του συνολικά προσφερόμενου φορτίου και εν συνεχεία του throughput ανά κανάλι δεδομένων για διάφορες τιμές του propagation delay.

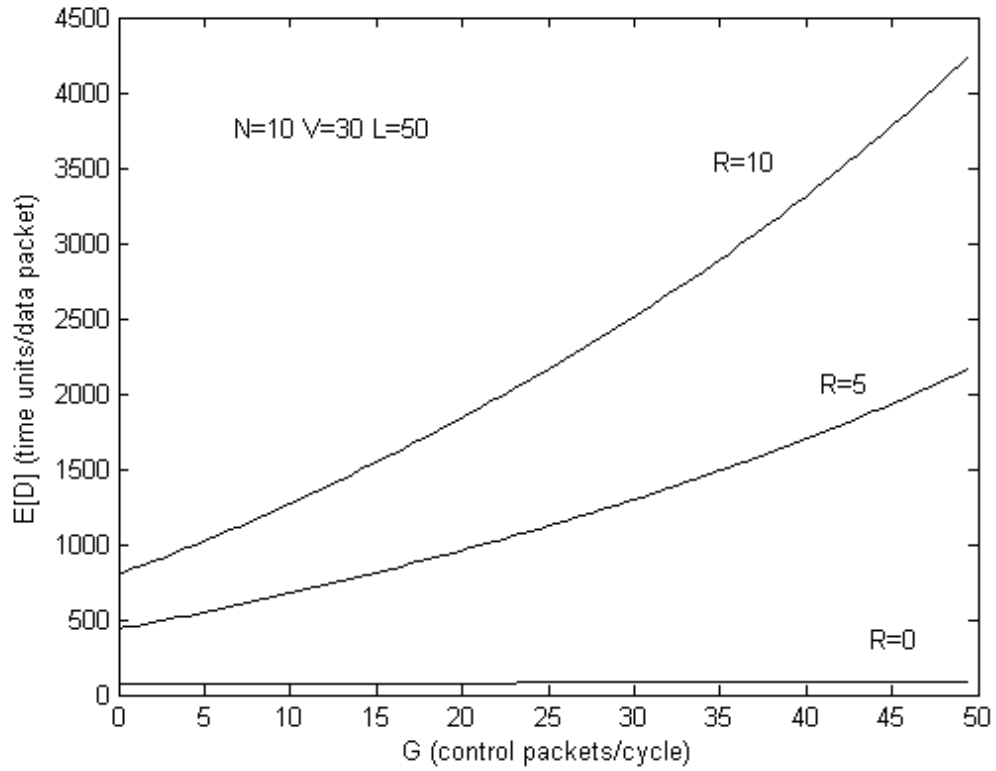


Chart 3.8: Analysis Results The Average Delay per packet, $E[D]$ (time units/data packet) versus the offered load, G (control packets/cycle) for $R=\{0,5,10\}$ data slots, $N=10$ data channels, $V=30$ control channels and $L=50$ time units.

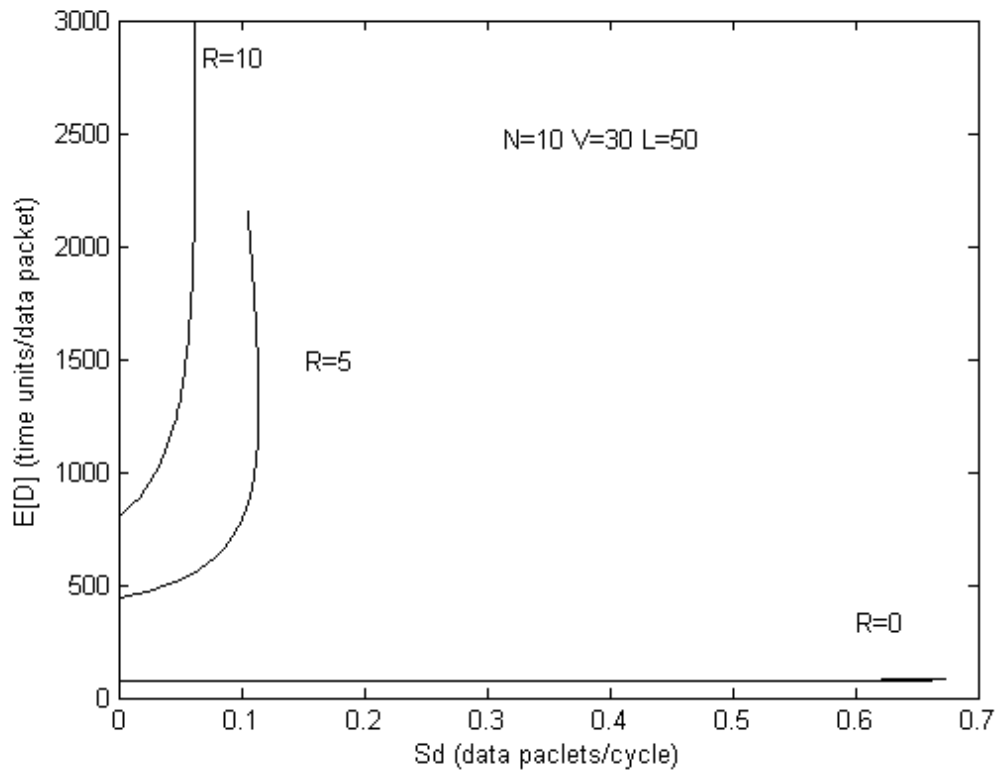


Chart 3.9: Analysis Results The Average Delay per packet, $E[D]$ (time units/data packet) versus the throughput per cycle per channel, S_D (data packets/cycle) for $R=\{0,5,10\}$ data slots, $N=10$ data channels, $V=30$ control channels and $L=50$ time units.

Και στα δύο άνω γραφήματα κοινό συμπέρασμα είναι ότι η αύξηση της τιμής της παραμέτρου R οδηγεί σε σημαντική αύξηση της μέσης τιμής της καθυστέρησης ανά πακέτο, καθιστώντας την καθυστέρηση διάδοσης πολύ σημαντική παράμετρο για τα WDMA συστήματα. Από τη σχέση (3.20), η μέση τιμή της καθυστέρησης είναι αύξουσα συνάρτηση του propagation delay.

Ανάλυση με την επίδραση του Receiver Collisions Phenomenon

Προχωρώντας την ανάλυση και λαμβάνοντας υπόψη το φαινόμενο των receiver collisions, θεωρούμε ότι το συνολικό φορτίο G (control packets/cycle) προέρχεται ομοιόμορφα από πεπερασμένο αριθμό σταθμών, M .

Η σχέση (3.8) δίνει τη μέση τιμή των πακέτων που έχουν επιλεγθεί για να μεταδοθούν στα N κανάλια δεδομένων σε ένα cycle. Αν κάποιος σταθμός – δέκτης είναι προορισμός παραπάνω από ενός σταθμών που έχουν το δικαίωμα να μεταδώσουν, τότε ο σταθμός – προορισμός δέχεται μονάχα ένα πακέτο. Οι υπόλοιποι σταθμοί θα πρέπει να επανεκπέμψουν το πακέτο τους.

Η πολιτική επιτυχίας ενός πακέτου στους δέκτες είναι η ίδια με την πολιτική επιτυχίας στα κανάλια δεδομένων. Άρα κατ' αντιστοιχία με την ανάλυση στα κανάλια δεδομένων:

Έστω k το πλήθος των πακέτων που κέρδισαν πρόσβαση στα N κανάλια δεδομένων και προορίζονται στους M σταθμούς. Οι k σταθμοί που τα πακέτα τους μεταδόθηκαν επιτυχώς επιλέγουν έναν σταθμό – προορισμό με πιθανότητα $1/M$, δίνοντας M^k διαφορετικές διατάξεις. Αν ένας σταθμός δεν αποτελεί προορισμό κανενός εκ των k επιτυχών πακέτων, τότε τα k πακέτα διανέμονται στους $M-1$ εναπομείναντες σταθμούς, δίνοντας $(M-1)^k$ διαφορετικές διατάξεις. Συνεπώς η πιθανότητα ένας σταθμός να μη δεχθεί κανένα πακέτο είναι:

$$P_0(k) = \frac{(M-1)^k}{M^k} = \left(1 - \frac{1}{M}\right)^k \quad (3.21)$$

και η συμπληρωματική της, δηλαδή η πιθανότητα ένας σταθμός να επιλεγεί από ένα ή παραπάνω σταθμούς είναι:

$$P_M(k) = 1 - \left(1 - \frac{1}{M}\right)^k \quad (3.22)$$

Στη μόνιμη κατάσταση, το πλήθος των k πακέτων σε ένα cycle ισούται με το $S_{total} = \frac{L}{c} NP_{Dsuc}$ από τη σχέση (3.8).

Συνεπώς η πιθανότητα επιλογής ενός σταθμού από 1 ή παραπάνω πακέτα είναι στη μόνιμη κατάσταση:

$$(3.22) \xrightarrow{k=S_{total}} P_M(S_{total}) = 1 - \left(1 - \frac{1}{M}\right)^{S_{total}} \quad (3.23)$$

Η πιθανότητα να επιλεγθούν m σταθμοί στο σύνολο των M ως τελικοί προορισμοί είναι:

$$P_{Mm} = \binom{M}{m} P_M^m (1 - P_M)^{M-m} \quad (3.24)$$

Αν η μεταβλητή $R_{stations}$ εκφράζει πόσοι σταθμοί επιλέγονται ως τελικοί προορισμοί, με:

$$P(R_{stations} = m) = \binom{M}{m} P_M^m (1 - P_M)^{M-m} \quad (3.25)$$

η μέση τιμή αυτής της μεταβλητής δίνει το throughput του συστήματος (μέση τιμή των επιτυχώς λαμβανομένων πακέτων δεδομένων σε ένα cycle) [9].

Η μέση τιμή της μεταβλητής $R_{stations}$ είναι [9], [11]:

$$E[P(R_{stations} = m)] = \sum_{m=1}^M m \binom{M}{m} P_M^m (1 - P_M)^{M-m} = MP_M$$

Οπότε:

$$S_{RCtotal} = \frac{L}{C} E[R_{stations}] \xrightarrow{[9]} S_{RCtotal} = \frac{L}{C} MP_M \quad (3.26)$$

Ο μέσος όρος των πακέτων που μεταδίδονται και λαμβάνονται επιτυχώς ανά cycle ανά κανάλι δεδομένων είναι:

$$\frac{S_{RCtotal}}{N} = S_{RC} = \frac{L MP_M (S_{total})}{C N} \xrightarrow{(3.23)}$$

$$S_{RC} = \frac{L M}{C N} \left(1 - \left(1 - \frac{1}{M} \right)^{S_{total}} \right) \xrightarrow{(3.8)}$$

$$S_{RC} = \frac{L M}{C N} \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L NP_{Dsuc}}{C}} \right) \xrightarrow{(3.5)}$$

$$S_{RC} = \frac{L M}{C N} \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L N}{C} \left(1 - \left(1 - \frac{1}{N} \right)^{S_{Ctotal}} \right)} \right) \xrightarrow{(3.2)}$$

$$S_{RC} = \frac{L M}{C N} \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L N}{C} \left(1 - \left(1 - \frac{1}{N} \right)^{Ge^{-\frac{G}{V}}} \right)} \right) \quad (3.27)$$

Η σχέση (3.27) εκφράζει το throughput ανά κανάλι δεδομένων ανά cycle.

Ακολουθούν γραφήματα που απεικονίζουν το throughput συναρτήσει του προσφερόμενου φορτίου για διάφορες τιμές του πλήθους των καναλιών ελέγχου και δεδομένων.

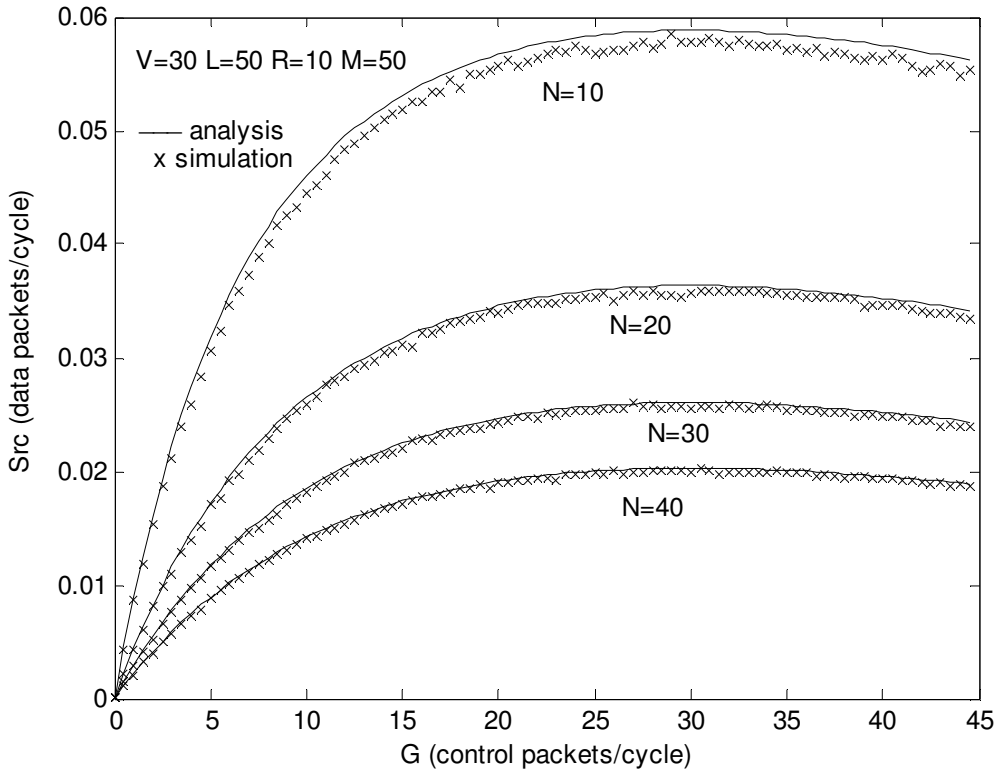


Chart 3.10: The throughput per cycle per channel, S_{RC} (data packets/cycle) versus the offered load, G (control packets/cycle) for $N=\{10,20,30,40\}$ data channels, $V=30$ control channels and $C=551$ time units ($L=50$ time units and $T_r=500$ time units).

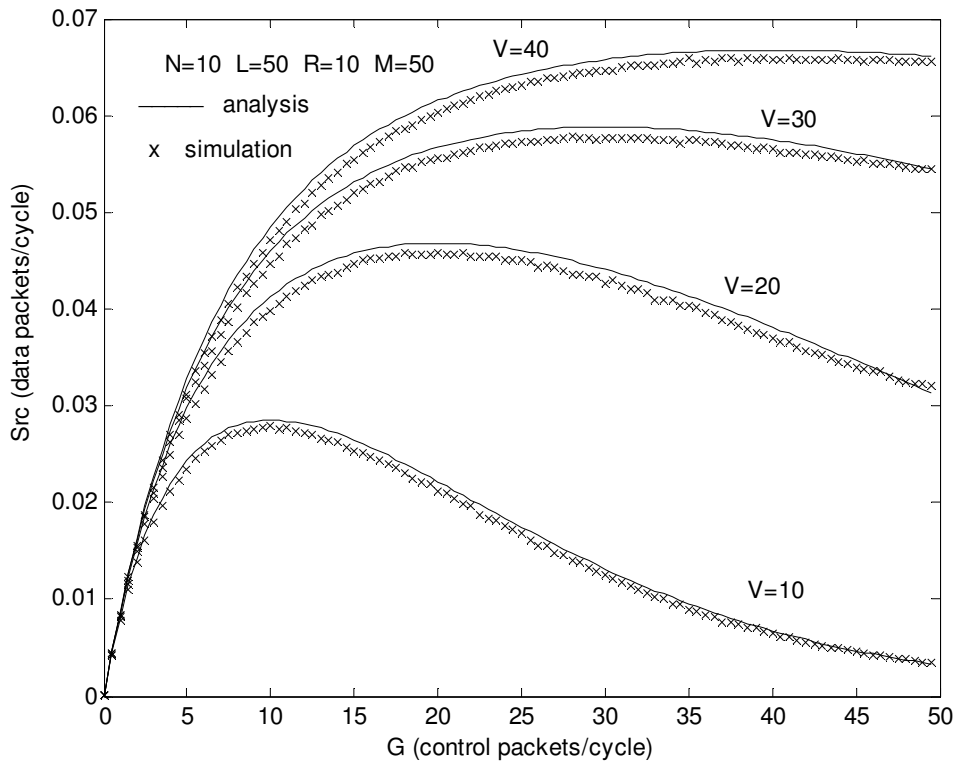


Chart 3.11: The throughput per cycle per data channel, S_{RC} (data packets/cycle) versus the offered load, G (control packets/cycle) for $V=\{10,20,30,40\}$ control channels, $N=10$ data channels and $C=551$ time units ($L=50$ time units and $R=10$ data slots).

Στα γραφήματα που ακολουθούν αναπαρίσταται το throughput συναρτήσει του προσφερόμενου φορτίου με και χωρίς Receiver Collision Phenomenon Analysis. Παρατηρεί κανείς τη διαφορά στο throughput όταν λαμβάνονται υπόψη οι συγκρούσεις στο δέκτη και όταν οι συγκρούσεις αυτές αγνοούνται.

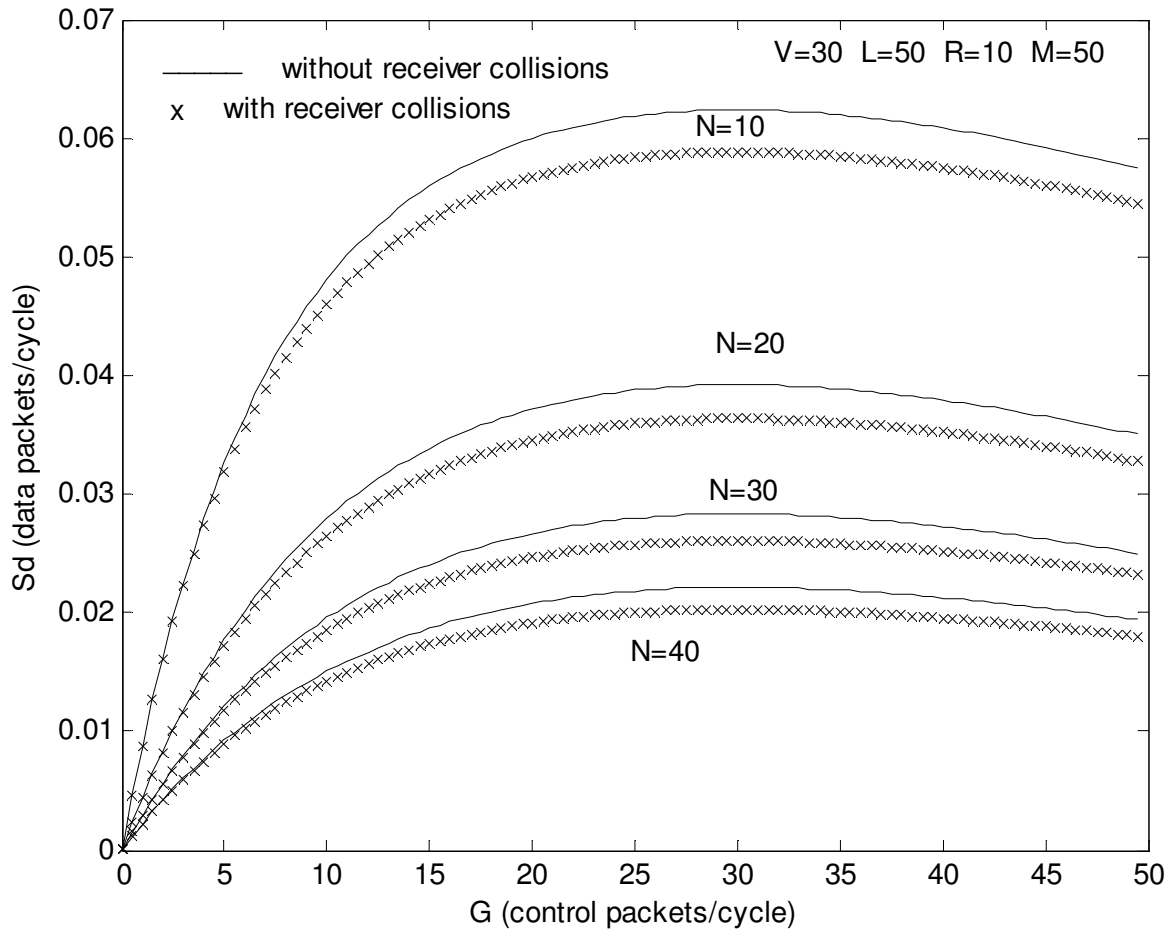


Chart 3.12: Analysis Results: The throughput per cycle per channel, $\{S_D, S_{RCD}\}$ (data packets/cycle) versus the offered load, G (control packets/cycle) for $N=\{10,20,30,40\}$ data channels, $V=30$ control channels and $C=551$ time units ($L=50$ time units and $T_r=500$ time units).

Η απόκλιση στις 2 καμπύλες αυξάνεται με την αύξηση του αριθμού των καναλιών ελέγχου.

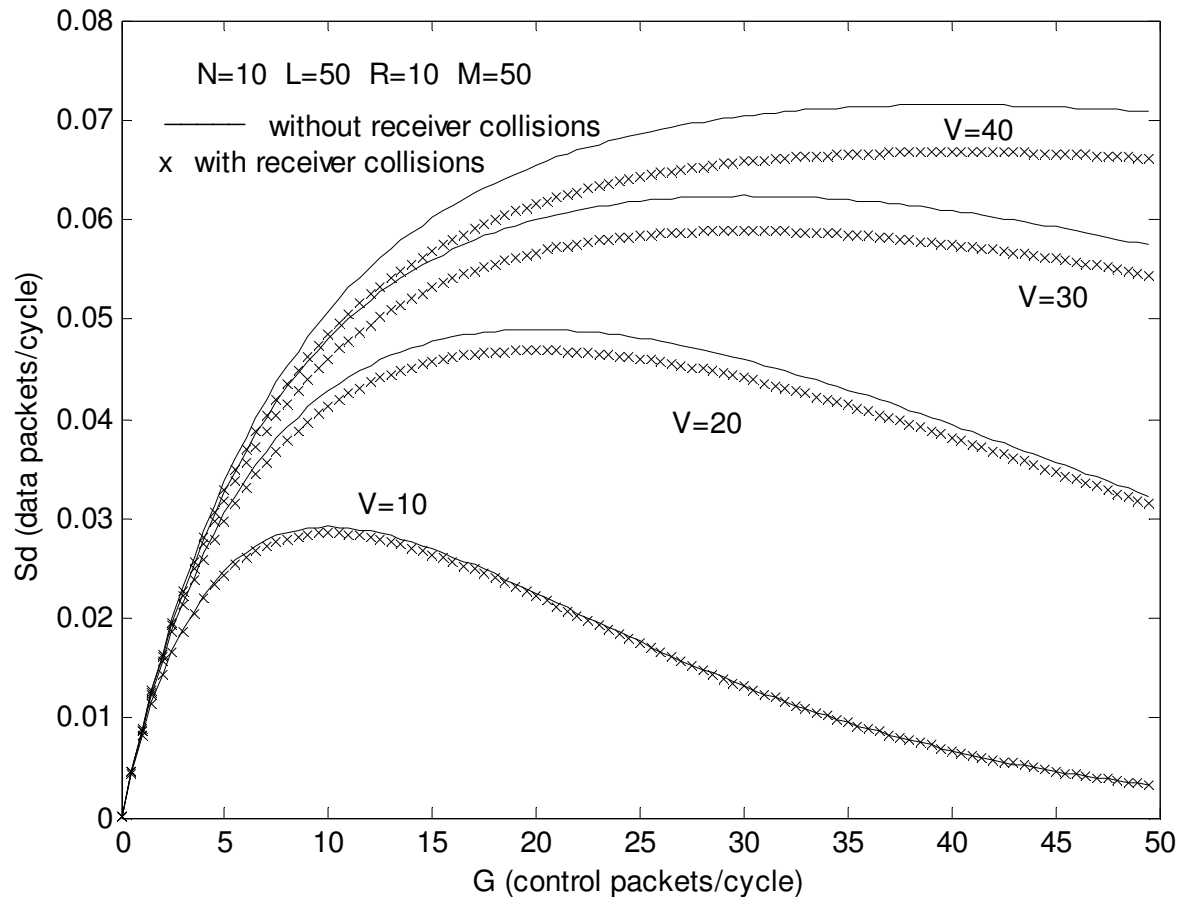


Chart 3.13: Analysis Results: The throughput per cycle per channel, $\{S_D, S_{RCD}\}$ (data packets/cycle) versus the offered load, G (control packets/cycle) for $N=\{10,20,30,40\}$ data channels, $V=30$ control channels and $C=551$ time units ($L=50$ time units and $R=10$ data slots).

Μέσης Πιθανότητας Απόρριψης

Η Μέση Πιθανότητα Απόρριψης στη μόνιμη κατάσταση, ορίζεται ως ο λόγος της μέσης τιμής των πακέτων δεδομένων που απορρίπτονται στον προορισμό από ενεργούς δέκτες προς τη μέση τιμή των επιτυχώς μεταδιδόμενων πακέτων δεδομένων[10]:

$$P_{rej} = \frac{S_{total} - S_{RCtotal}}{S_{total}} \quad (3.28)$$

Στις ακόλουθες γραφικές παραστάσεις της Μέσης Πιθανότητας Απόρριψης συναρτήσεως του προσφερόμενου φορτίου, αύξηση είτε του αριθμού των control καναλιών είτε του πλήθους των data καναλιών οδηγεί σε αύξηση του μεγέθους της Average Rejection Probability.

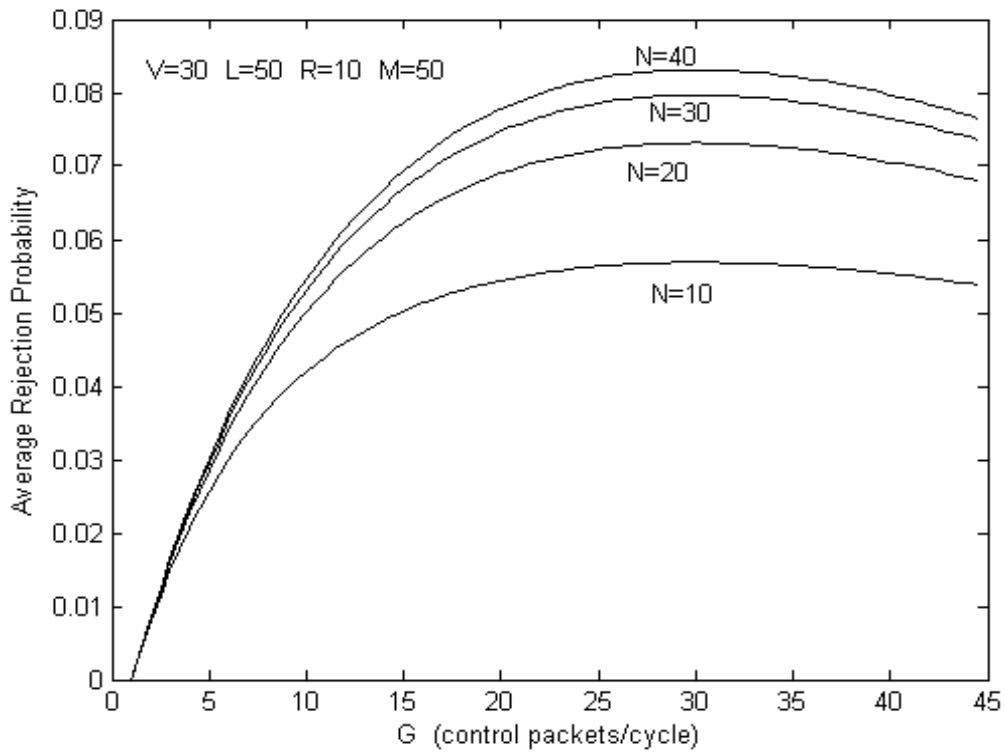


Chart 3.14: Average Rejection Probability versus offered load, G (control packets/time unit), for $N=\{10,20,30,40\}$ data channels, $V=30$ control channels $L=50$ time units $R=10$ data slots, $M=50$ stations.

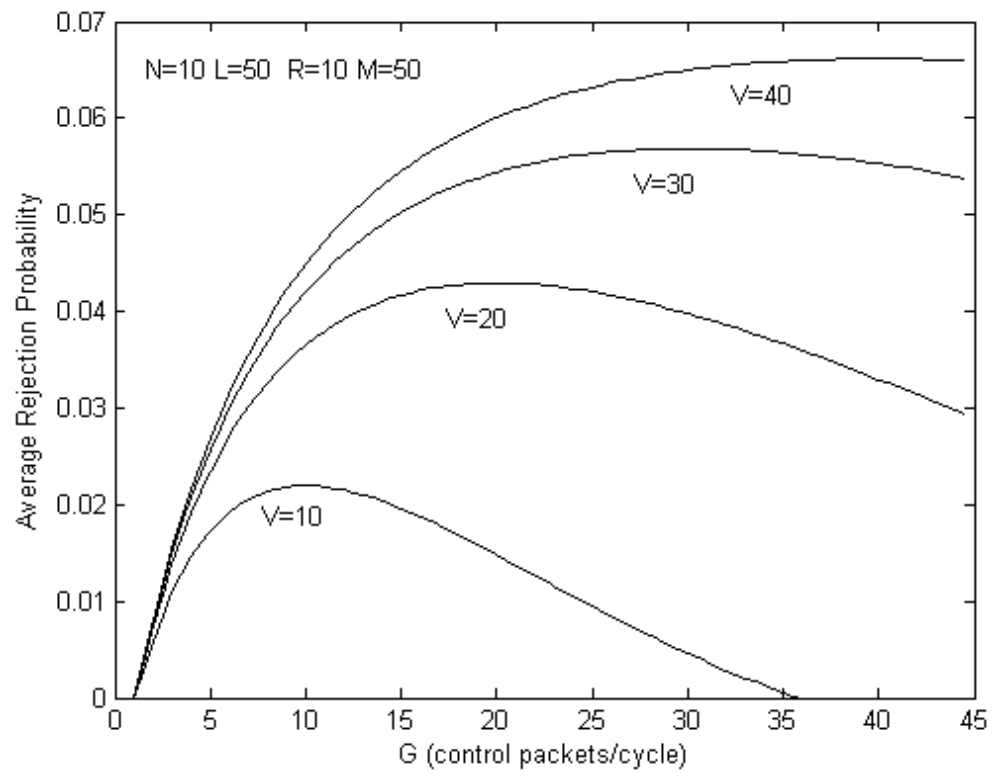


Chart 3.15: Average Rejection Probability versus offered load, G (control packets/time unit), for $V=\{10,20,30,40\}$ control channels, $N=10$ data channels $L=50$ time units $R=10$ data slots, $M=50$ stations.

Στα παρακάτω γραφήματα αναπαρίσταται η Μέση Πιθανότητα Απόρριψης για διαφορετικά πλήθη σταθμών, $M=\{50, 100, 150\}$. Όπως ήταν αναμενόμενο η P_{rej} μειώνεται καθώς ο αριθμός των σταθμών αυξάνεται –η πιθανότητα ένας σταθμός να είναι προορισμός είναι αντιστρόφως ανάλογη του πλήθους των σταθμών, M .

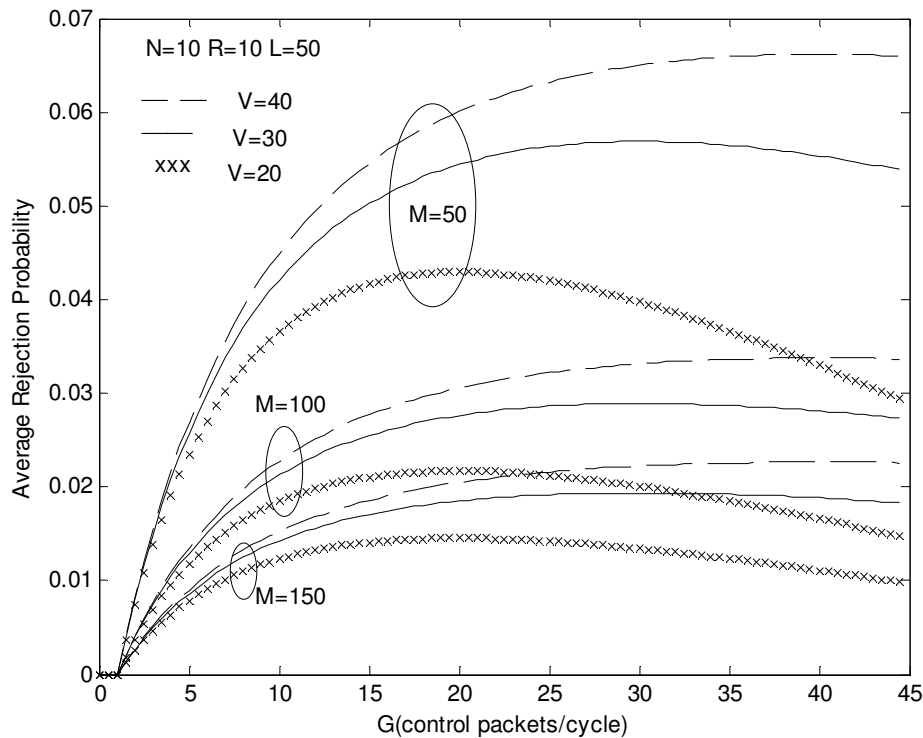


Chart 3.16: Average Rejection Probability versus offered load, G (control packets/time unit), for $V=\{20,30,40\}$ control channels, $N=10$ data channels $L=50$ time units $R=10$ data slots, $M=\{50, 100, 150\}$ stations.

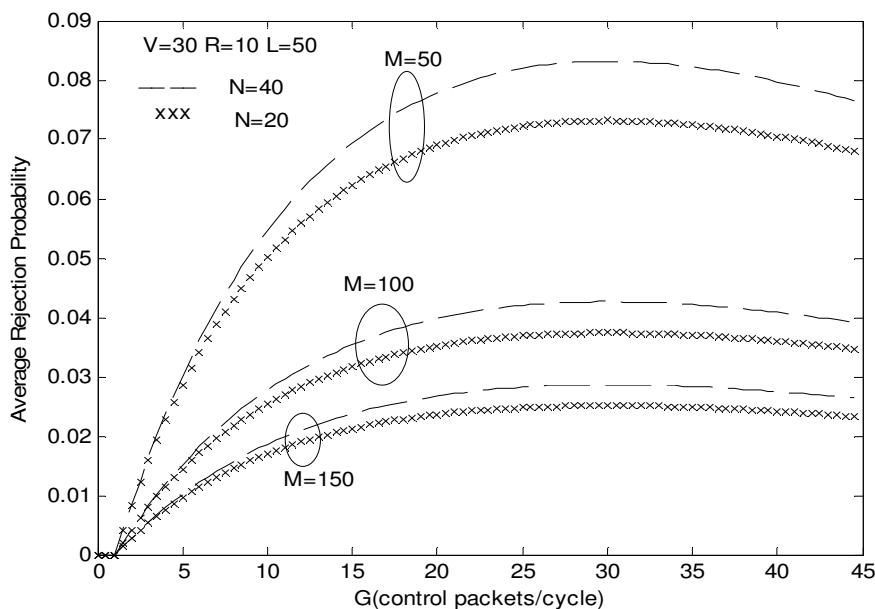


Chart 3.17: Average Rejection Probability versus offered load, G (control packets/time unit), for $V=30$ control channels, $N=\{20, 40\}$ data channels $L=50$ time units $R=10$ data slots, $M=\{50, 100, 150\}$ stations.

Η επίδραση του πλήθους των σταθμών αλλά και του αριθμού καναλιών δεδομένων και ελέγχου στην τιμή της P_{rej} γίνεται περισσότερο αντιληπτή μέσα από τα ραβδογράμματα που ακολουθούν.

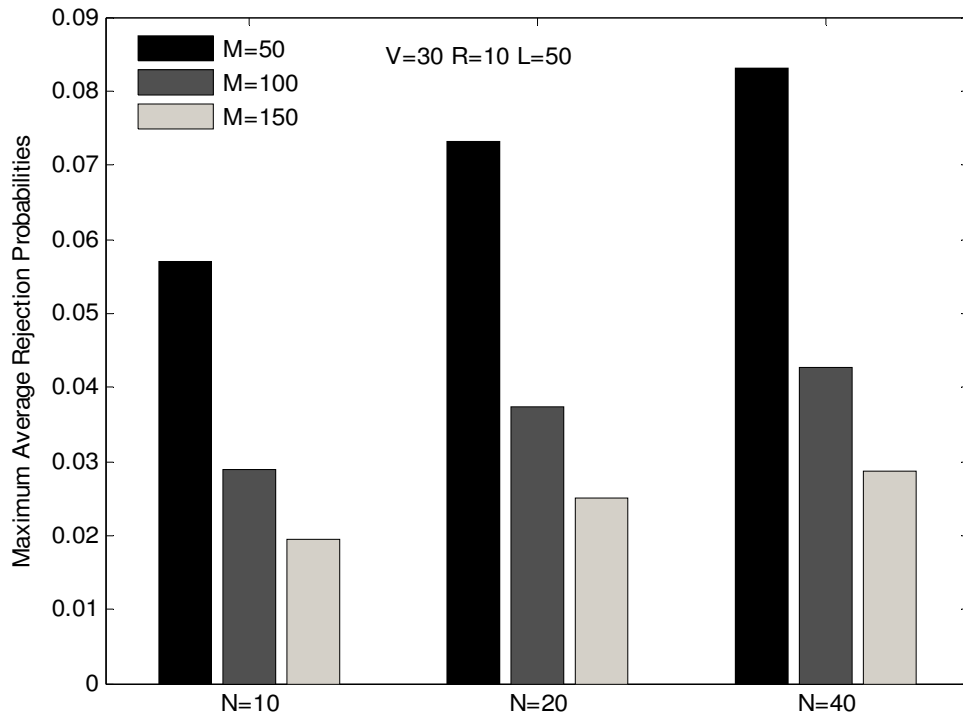


Chart 3.18: Bar Diagram: Maximum Average Rejection Probability $V=30$ control channels, $N=\{10, 20, 40\}$ data channels $L=50$ time units $R=10$ data slots, $M=\{50, 100, 150\}$ stations.

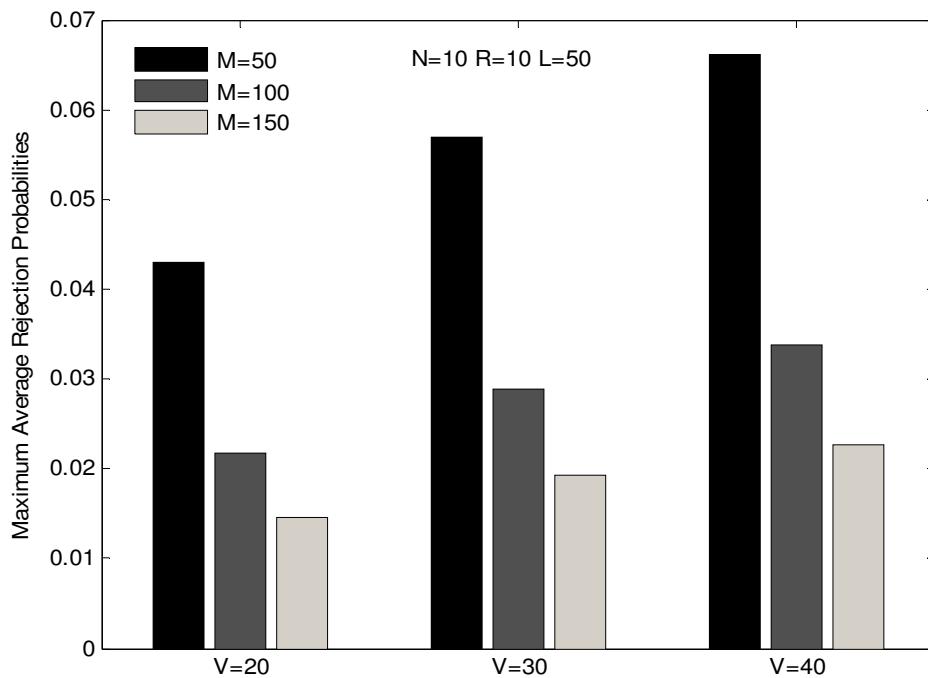


Chart 3.19: Bar Diagram: Maximum Average Rejection Probability $V=\{20, 30, 40\}$ control channels, $N=10$ data channels $L=50$ time units $R=10$ data slots, $M=\{50, 100, 150\}$ stations.

Ανάλυση της καθυστέρησης

Η ανάλυση της καθυστέρησης λαμβάνοντας υπόψη το φαινόμενο των συγκρούσεων στους δέκτες, ακολουθεί τη λογική της ανάλυσης που προηγήθηκε για την περίπτωση που το φαινόμενο δε λαμβάνεται υπόψη. Στην παρούσα ανάλυση, με τον όρο «επιτυχή μετάδοση πακέτου», εννοούμε πάντα «επιτυχή μετάδοση και αποδοχή ενός πακέτου από το δέκτη του σταθμού – προορισμού».

Η καθυστέρηση ορίζεται ως το χρονικό διάστημα μεταξύ της γέννησης ενός πακέτου και της επιτυχούς παραλαβής του από το δέκτη του σταθμού προορισμού.

$$D = D_w + D_r + C + R \times L \quad (3.29)$$

Όπου:

$C + R \times L = 1 + L + 2 \times T_r = 1 + (2R + 1) \times L$ time units, ο χρόνος που απαιτείται για τη μετάδοση και διάδοση του control packet, συν το χρόνο μετάδοσης και διάδοσης του πακέτου δεδομένων (3.30)

D_w = η καθυστέρηση από τη στιγμή της γέννησης ενός πακέτου έως την αρχή του επόμενου cycle, με μέση τιμή:

$$E[D_w] = \frac{1}{c} \int_0^c t dt \rightarrow E[D_w] = \frac{c}{2} \text{ time units} \quad (3.31)$$

D_r = η καθυστέρηση από τη χρονική στιγμή που μεταδίδεται το control packet για πρώτη φορά έως την έναρξη του cycle που ο σταθμός καταφέρνει να αποστείλει επιτυχώς το πακέτο δεδομένων του.

Έστω ότι ένας σταθμός καταβάλλει n προσπάθειες για να επιτύχει να μεταδώσει ένα πακέτο. Υποθέτοντας ότι η πιθανότητα επιτυχούς μετάδοσης του πακέτου είναι ίδια για κάθε προσπάθεια, η πιθανότητα επιτυχίας μετά από n προσπάθειες είναι:

$$Q(n) = P_{suc}(1 - P_{suc})^{n-1} \quad (3.32)$$

με μέση τιμή προσπαθειών:

$$A = E[n] = \sum_{n=1}^{\infty} nQ(n) = P_{suc}^{-1} \quad (3.33)$$

όπου P_{suc} είναι η πιθανότητα επιτυχούς μετάδοσης ενός πακέτου .

Ισχύει:

$$S_{RCtotal} = \frac{L}{c} G P_{suc} \rightarrow P_{suc} = \frac{c S_{RCtotal}}{LG}$$

$$\xrightarrow{(3.26)} P_{suc} = \frac{c \frac{L}{c} M P_M}{LG} \rightarrow P_{suc} = \frac{M P_M}{G}$$

$$\xrightarrow{(3.23)} P_{suc} = \frac{M \left(1 - \left(1 - \frac{1}{M} \right)^{S_{total}} \right)}{G}$$

$$\xrightarrow{(3.8)} P_{suc} = \frac{M \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L}{C} N P_{Dsuc}} \right)}{G}$$

$$\xrightarrow{(3.5)} P_{suc} = \frac{M \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L}{C} N (1 - (1 - \frac{1}{N})^{S_{Ctotal}})} \right)}{G}$$

$$\xrightarrow{(3.2)} P_{suc} = \frac{M \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L}{C} N \left(1 - \left(1 - \frac{1}{N} \right)^{G e^{-\frac{G}{V}}} \right)} \right)}{G} \quad (3.34)$$

και

$$P_{suc}^{-1} = \frac{G}{M \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L}{C} N \left(1 - \left(1 - \frac{1}{N} \right)^{G e^{-\frac{G}{V}}} \right)} \right)} \quad (3.35)$$

$$(3.33) \xrightarrow{(3.35)} A = \frac{G}{M \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L}{C} N \left(1 - \left(1 - \frac{1}{N} \right)^{G e^{-\frac{G}{V}}} \right)} \right)} \quad (3.36)$$

Στις A προσπάθειες που καταβάλλει ο σταθμός για να επιτύχει μετάδοση, οι πρώτες A-1 είναι προφανώς αποτυχημένες. Σε κάθε αποτυχημένη προσπάθεια καταναλίσκονται $1+T_r$ time units (1 time unit για μετάδοση του control packet T_r time units για τη διάδοση του control packet).

Οπότε η μέση τιμή της καθυστέρησης D_r προκύπτει:

$$E[D_r] = (A - 1)[1 + T_r] \xrightarrow{(3.36)} \left(\frac{G}{M \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L}{C} N \left(1 - \left(1 - \frac{1}{N} \right)^{G e^{-\frac{G}{V}}} \right)} \right)} - 1 \right) (1 + T_r) \quad (3.37)$$

Από τις σχέσεις (3.29), (3.30), (3.31) και (3.37) προκύπτει η μέση τιμή της καθυστέρησης:

$$E[D] = E[D_w + D_r + C + R \times L] = E[D_w] + E[D_r] + C + R \times L \rightarrow$$

$$E[D] = \frac{C}{2} + \left(\frac{G}{M \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L}{C} N \left(1 - \left(1 - \frac{1}{N} \right)^{G e^{-\frac{G}{V}}} \right)} \right)} - 1 \right) (1 + T_r) + C + R \times L \rightarrow$$

$$E[D] = \frac{1 + (R + 1) \times L}{2} + \left(\frac{G}{M \left(1 - \left(1 - \frac{1}{M} \right)^{\frac{L}{C} N \left(1 - \left(1 - \frac{1}{N} \right)^{G e^{-\frac{G}{V}}} \right)} \right)} - 1 \right) \times (1 + R \times L) +$$

+1 + (2R + 1) \times L \quad (3.38)

Στα γραφήματα που ακολουθούν αναπαρίσταται η μέση καθυστέρηση συναρτήσει του προσφερόμενου φορτίου, για V σταθερό και κυμαινόμενο N και για κυμαινόμενο V και σταθερό N.

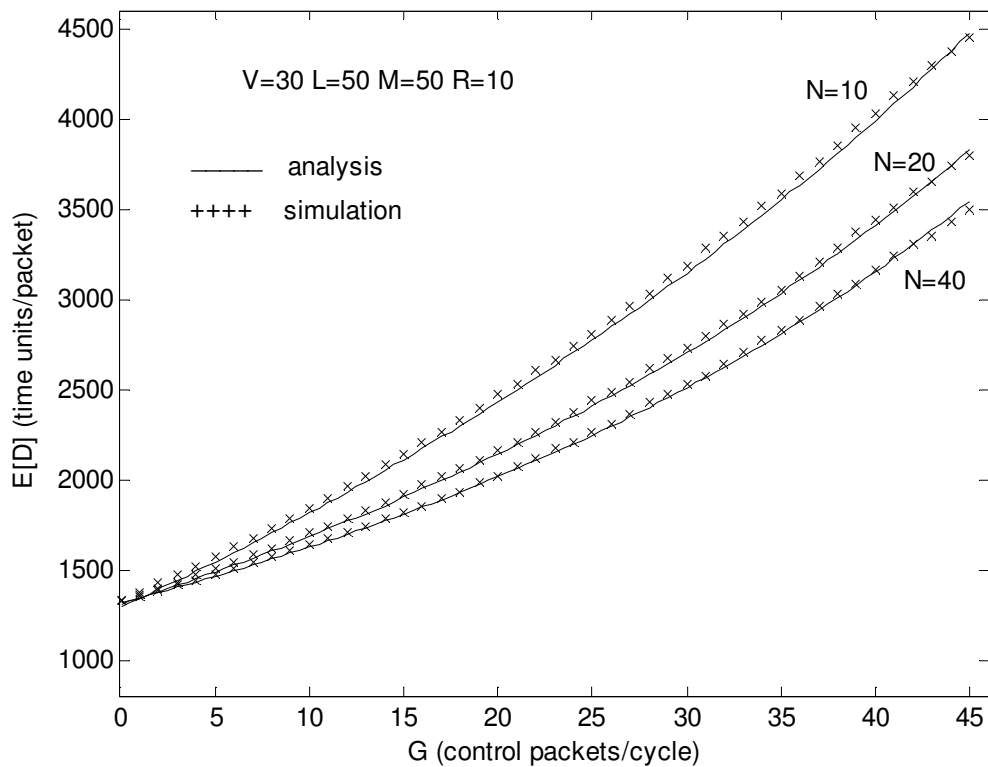


Chart 3.20: Average Delay per data packet, $E[D]$ (time units/data packet) versus the offered load, G (control packets/cycle) for $N = \{10, 20, 40\}$ data channels, $V=30$ control channels, $M=50$ stations and $C=551$ time units.

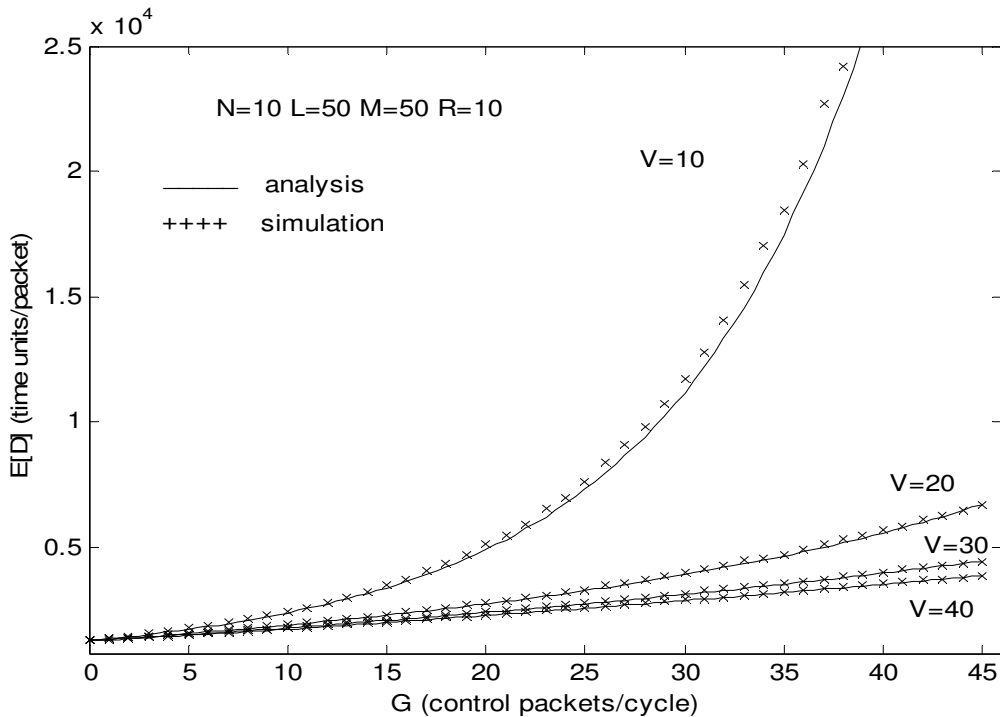


Chart 3.21: Average Delay per data packet, $E[D]$ (time units/data packet) versus the offered load, G (control packets/cycle) for $V= \{10, 20, 30, 40\}$ control channels, $N=10$ control channels, $M=50$ stations and $C=551$ time units.

Η αύξηση του αριθμού των καναλιών δεδομένων και των καναλιών ελέγχου αυξάνει την πιθανότητα επιτυχούς μετάδοσης των πακέτων δεδομένων και ελέγχου αντίστοιχα οπότε οδηγεί σε μείωση της καθυστέρησης.

Στα γραφήματα που ακολουθούν αναπαρίσταται η μέση καθυστέρηση συναρτήσει του throughput, για V σταθερό και κυμαινόμενο N και για κυμαινόμενο V και σταθερό N .

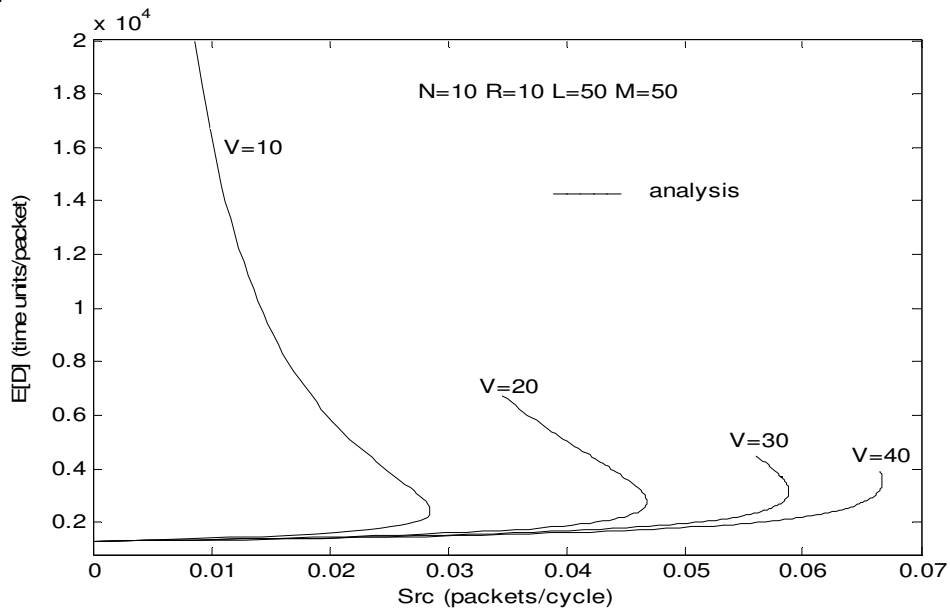


Chart 3.22: Average Delay per data packet, $E[D]$ (time units/data packet) versus the throughput per data channel, SRC (data packets/cycle) for $V= \{10, 20, 30, 40\}$ control channels, $N=10$ control channels, $M=50$ stations and $C=551$ time units.

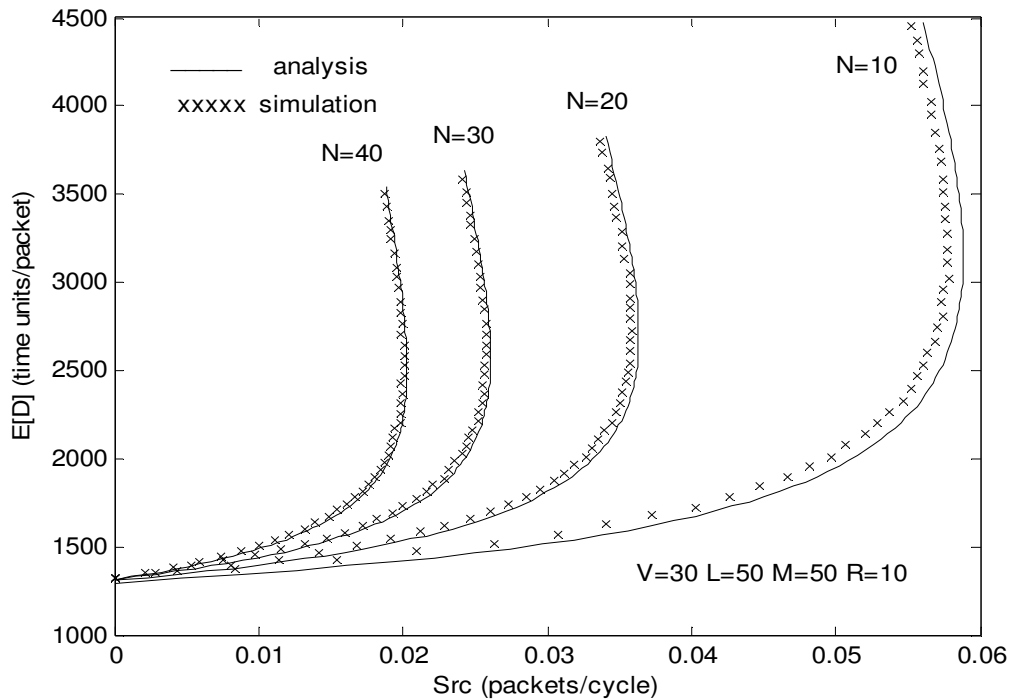


Chart 3.23: Average Delay per data packet, $E[D]$ (time units/data packet) versus the throughput per data channel, SRC (data packets/cycle) for $N= \{10, 20, 30, 40\}$ data channels, $V=30$ control channels, $M=50$ stations and $C=551$ time units.

Ακολουθούν κοινές γραφικές παραστάσεις της καθυστέρησης συναρτήσει του throughput και εν συνεχεία του προσφερόμενου φορτίου χωρίς και με Receiver Collision Phenomenon.

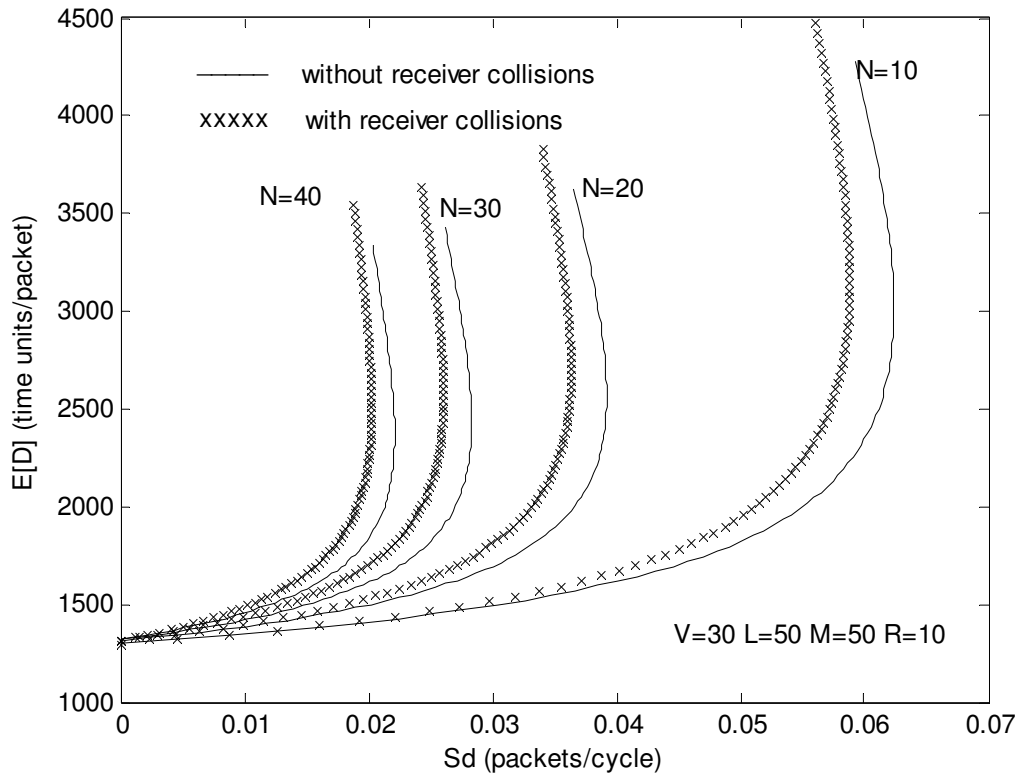


Chart 3.24: Average Delay per data packet, $E[D]$ (time units/data packet) versus the throughput per data channel, $\{SRC, SD\}$ (data packets/cycle) for $N= \{10, 20, 30, 40\}$ data channels, $V=30$ control channels, $M=50$ stations and $C=551$ time units.

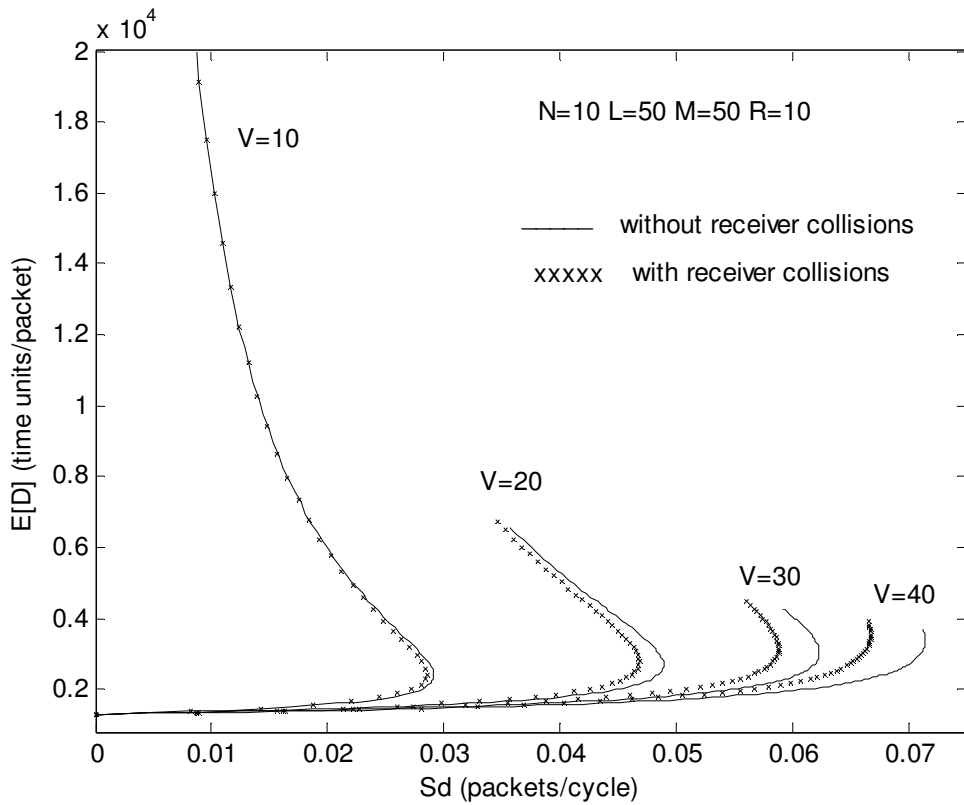


Chart 3.25: Average Delay per data packet, $E[D]$ (time units/data packet) versus the throughput per data channel, $\{SRC, SD\}$ (data packets/cycle) for $V = \{10, 20, 30, 40\}$ control channels, $N=10$ data channels, $M=50$ stations and $C=551$ time units.

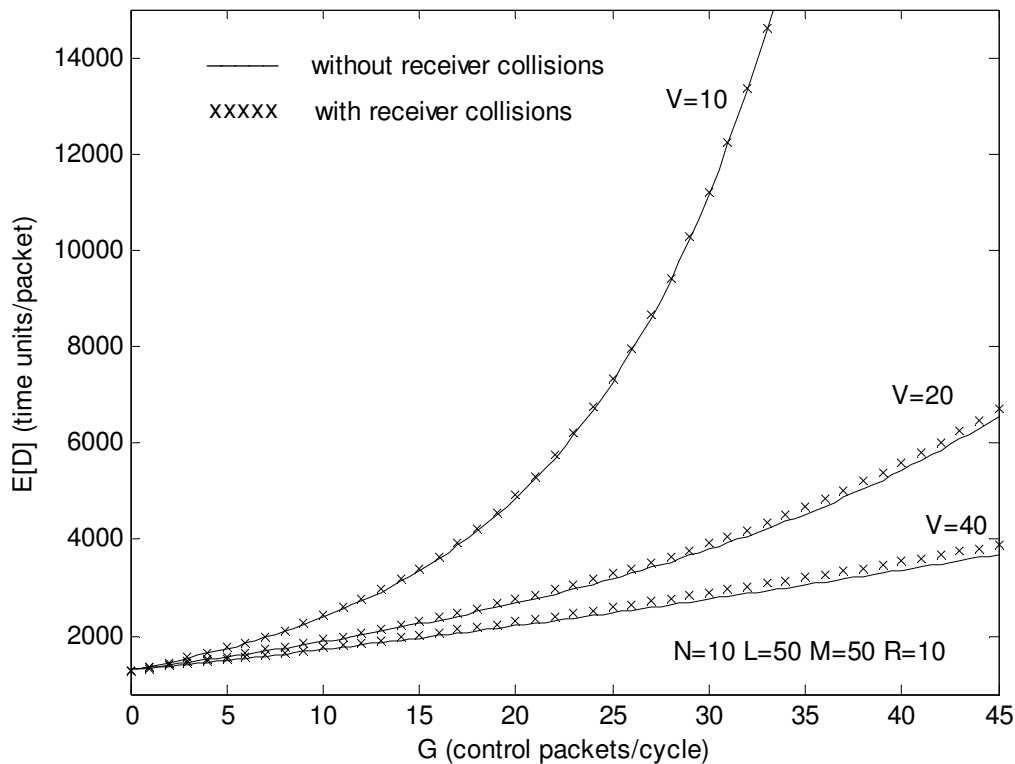


Chart 3.26: Average Delay per data packet, $E[D]$ (time units/data packet) versus the offered load, G (control packets/cycle) for $V = \{10, 20, 40\}$ control channels, $N=10$ data channels, $M=50$ stations and $C=551$ time units.

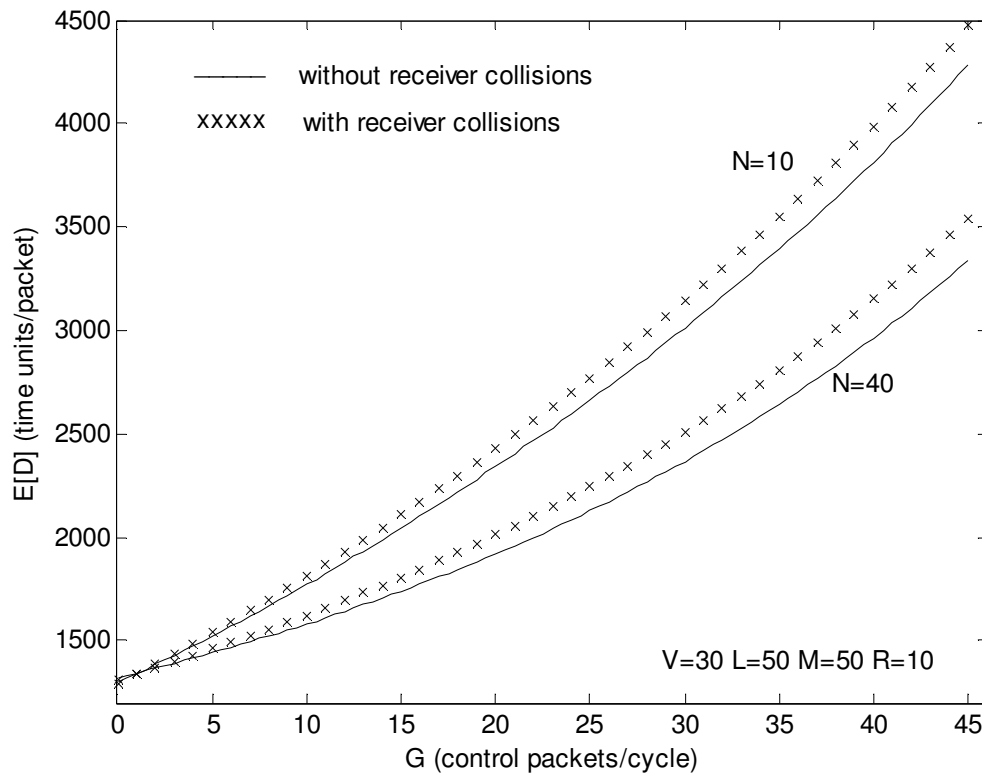


Chart 3.27: Average Delay per data packet, $E[D]$ (time units/data packet) versus the offered load, G (control packets/cycle) for $N= \{10, 40\}$ data channels, $V=30$ control channels, $M=50$ stations and $C=551$ time units.

Η επίδραση του φαινομένου των Receiver Collisions στο μέγεθος της καθυστέρησης είναι εμφανής από τις ανωτέρω γραφικές παραστάσεις.

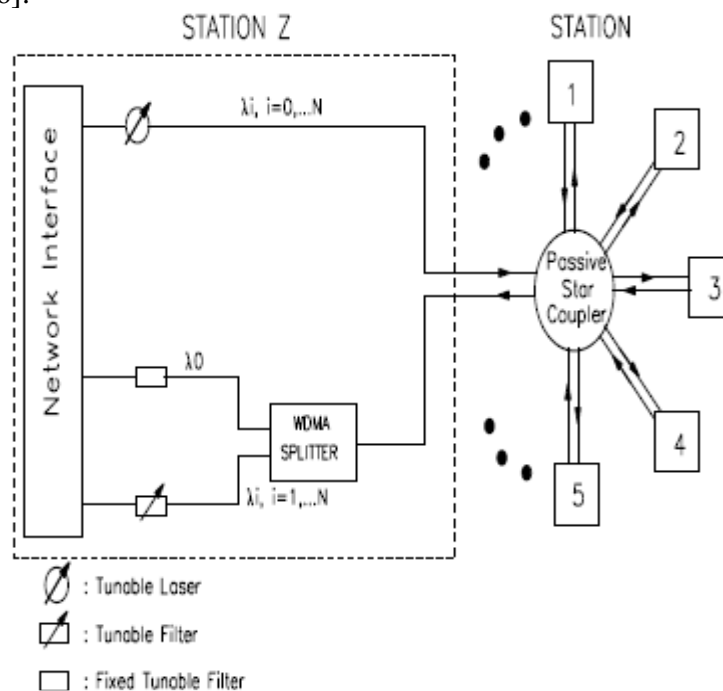
Κεφάλαιο 4

Ασύγχρονο WDMA Single Control/Multi Data Σύστημα με Poisson Αφίξεις

Αρχιτεκτονική του δικτύου και παραδοχές

Και σε αυτήν την περίπτωση μελετάται ένα WDMA σύστημα σε τοπολογία παθητικού αστέρα. Αρχικά θεωρούμε ότι ο αστέρας εξυπηρετεί άπειρο αριθμό σταθμών (Without Receiver Collisions Phenomenon Analysis) και εν συνεχεία μελετάμε την περίπτωση όπου η προσφερόμενη κίνηση προέρχεται από πεπερασμένο αριθμό σταθμών, M (Analysis with the Receiver Collisions Phenomenon Effect). Η προσφερόμενη κίνηση δίνεται από την κατανομή Poisson. Οι σταθμοί που είναι συνδεδεμένοι στο σύστημα διαθέτουν buffers μοναδιαίας χωρητικότητας. Αν ένα πακέτο αποτύχει να εκπεμφθεί αποθηκεύεται στον buffer και επανεκπέμπεται ανάλογα με την πολιτική επανεκπομπής. Η τοπολογία και οι παραδοχές του συστήματος ακολουθούν το [6].

Χρησιμοποιούνται συνολικά $(1+N)$ μήκη κύματος εκ των οποίων ένα μήκος κύματος διατίθεται για μετάδοση πληροφορίας ελέγχου (control channel) – σημειογραφία CC – και τα N μήκη κύματος διατίθενται για μετάδοση δεδομένων (data channels). Κάθε κόμβος διαθέτει έναν συντονιζόμενο laser εκπομπό για να συντονίζεται σε κάποιο κανάλι δεδομένων για μετάδοση data πακέτου – σημειογραφία TT. Επίσης κάθε κόμβος διαθέτει έναν σταθερά συντονισμένο δέκτη - φίλτρο έτσι ώστε να λαμβάνει την control πληροφορία από το κανάλι ελέγχου – FR – και ένα φίλτρο δέκτη με ικανότητα συντονισμού στο κανάλι επιλογής του σταθμού – TR – για να λαμβάνει δεδομένα που στέλνονται από τους άλλους σταθμούς και έχουν αυτόν ως τελικό αποδέκτη. Έτσι το σύστημα περιγράφεται ως ένα CC – TT – FR – TR σύστημα [6].



Εικ. 8 Η τοπολογία παθητικού αστέρα.

Το μέγεθος των control πακέτων θεωρείται σταθερό και ο χρόνος μετάδοσης ενός control πακέτου λαμβάνεται ως χρόνος αναφοράς (Transmission time of a control packet= 1 time unit). Το ένα time unit που απαιτείται για τη μετάδοση του control πακέτου αποτελεί ένα control slot. Το κανάλι ελέγχου είναι slotted. Ο άξονας του χρόνου είναι διαιρεμένος σε συνεχή control slots και για να μεταδώσει κάποιος σταθμός ένα control πακέτο πρέπει να περιμένει την έναρξη του αμέσως επόμενου control slot. Η ενέργεια αυτή συντονίζεται διανέμοντας από ένα ρολόι σε όλους τους συνδεδεμένους στον παθητικό αστέρα σταθμούς (common clock).

Το μέγεθος των data πακέτων θεωρείται επίσης σταθερό και ο χρόνος μετάδοσης ενός πακέτου δεδομένων σε κανονικοποιημένες μονάδες χρόνου θεωρείται ίσος με L time units. Οι L χρονικές μονάδες αποτελούν ένα data slot. Ο χρόνος διάδοσης από οποιονδήποτε σταθμό στο συζεύκτη του παθητικού αστέρα και εν συνεχεία σε οποιονδήποτε σταθμό (roundtrip propagation delay) θεωρείται σταθερός και ίδιος για όλους τους σταθμούς του συστήματος, ίσος με $R \times L$ time units. Η πρόσβαση στα κανάλια δεδομένων είναι ασύγχρονη.

Επίσης θεωρούμε ότι οι χρόνοι συντονισμού των tunable transmitters και receivers καθώς και ο χρόνος που απαιτείται για την επεξεργασία της control πληροφορίας από τους σταθμούς είναι αμελητέοι.

Οι πληροφορίες που περιέχονται σε κάθε control πακέτο είναι η διεύθυνση του αποστολέα, η διεύθυνση του παραλήπτη, το κανάλι δεδομένων που έχει επιλέξει ο σταθμός για αποστολή των δεδομένων του καθώς και το χρονικό στιγμιότυπο που ο σταθμός έστειλε το πακέτο ελέγχου στο διάλυο ελέγχου [6].

CONTROL PACKET

T	R	λ_k	t_{th}
---	---	-------------	----------

Εικ. 9 Το control πακέτο.

T → διεύθυνση αποστολέα

R → διεύθυνση παραλήπτη

λ_k → το μήκος κύματος του καναλιού δεδομένων

t_{th} → η χρονική στιγμή αποστολής του control πακέτου

Πολιτική Μετάδοσης (Transmission Policy)

Όταν ένας σταθμός θέλει να μεταδώσει ένα πακέτο δεδομένων προς έναν άλλο σταθμό, αρχικά επιλέγει τυχαία ένα κανάλι δεδομένων λ_{kj} $j=1, \dots, N$ στο οποίο θα μεταδώσει το πακέτο του. Στη συνέχεια στέλνει ένα πακέτο ελέγχου που ενημερώνει όλους τους υπόλοιπους σταθμούς για την επικείμενη μετάδοση data πακέτου. Ο σταθμός περιμένει την έναρξη του αμέσως επόμενου control slot και μεταδίδει το control packet του που συναγωνίζεται με τα control packets των υπολοίπων σταθμών βάσει των κανόνων του πρωτοκόλλου Slotted Aloha για να κερδίσει πρόσβαση στο control κανάλι.

Ο σταθμός παρακολουθεί συνεχώς το control κανάλι ακούγοντας όλες τις μεταδόσεις πακέτων ελέγχου με τον fixed tuned receiver που διαθέτει. Δίνει ιδιαίτερη όμως έμφαση στις L τελευταίες χρονικές μονάδες της περιόδου επιβεβαίωσης (περίοδος επιβεβαίωσης, η διάρκεια του χρόνου διάδοσης, $R \times L$). Ο σταθμός ενημερώνεται για την έκβαση του control πακέτου του μετά το πέρας $R \times L$ χρονικών μονάδων, ακούγοντας το κανάλι ελέγχου. Έτσι με το πέρας αυτής της περιόδου είναι

σε θέση να γνωρίζει τόσο αν το κανάλι δεδομένων που έχει επιλέξει θα είναι ελεύθερο για μετάδοση, όσο και το αν ο δέκτης του σταθμού στον οποίο προορίζει τα δεδομένα του θα είναι ανενεργός μετά το πέρας των επόμενων $(R + 1) \times L$ time units (L time units για τη μετάδοση και $R \times L$ time units για τη διάδοση του data πακέτου). Αν το κανάλι δεδομένων επιλογής του σταθμού είναι ελεύθερο, ο σταθμός προβαίνει σε αποστολή των δεδομένων του το αμέσως επόμενο χρονικό slot από το τέλος της περιόδου επιβεβαίωσης. Σε περίπτωση που το επιλεγμένο κανάλι δεδομένων θα είναι κατειλημμένο ή ο δέκτης του σταθμού – προορισμού θα είναι ενεργός, ο σταθμός αναβάλλει την αποστολή για κάποια μεταγενέστερη χρονική στιγμή βάσει των κανόνων της πολιτικής επαναμετάδοσης. Κατ' αυτόν τον τρόπο δεν έχουμε συγκρούσεις ούτε στα κανάλια δεδομένων αλλά ούτε και στον προορισμό.

Η πολιτική επαναμετάδοσης (Retransmission Policy) ορίζει ότι σταθμοί που απέτυχαν να μεταδώσουν στην πρώτη απόπειρα, επιλέγουν τυχαία και ισοπίθανα μία μεταγενέστερη χρονική στιγμή με εύρος επιλογής K time units και επαναμεταδίδουν τα δεδομένα τους (και control και data πακέτο) βάσει της πολιτικής μετάδοσης.

Πολιτική Αποδοχής Πακέτου (Receiving Policy)

Όλοι οι σταθμοί παρακολουθούν συνεχώς τι συμβαίνει στο control κανάλι. Οπότε είναι σε θέση να γνωρίζουν την κατάσταση των καναλιών δεδομένων κάθε χρονική στιγμή Ένας σταθμός που αναγνωρίζει τη διεύθυνσή του σε ένα επιτυχώς μεταδιδόμενο control πακέτο (πεδίο R του control πακέτου, βλ. και Εικ. 9), αρχικά διαβάσει τη χρονική στιγμή που εστάλη το συγκεκριμένο control πακέτο (πεδίο t_{th} του control πακέτου) και εξετάσει αν ο σταθμός που έστειλε το πακέτο αυτό θα καταφέρει να μεταδώσει τα δεδομένα του σε κάποιο data κανάλι το προσεχές time unit από το πέρας της περιόδου επιβεβαίωσης. Στη συνέχεια, και εφόσον ξέρει ότι ο δέκτης του θα είναι ανενεργός και ότι θα δεχθεί πακέτο δεδομένων, συντονίζει το φίλτρο – δέκτη του στο μήκος κύματος που αναφέρεται στο control πακέτο μετά το πέρας $(R+1) \times L$ time units από τη στιγμή που «άκουσε» την πληροφορία και λαμβάνει το data πακέτο.

Ανάλυση – Αριθμητικά Αποτελέσματα – Γραφικές παραστάσεις

Ανάλυση χωρίς την επίδραση του Receiver Collisions Phenomenon

Από τη θεωρητική ανάλυση στο [10], έχουμε:

$$S_T = LGP_C P_D \rightarrow S_D = L \frac{G}{N} P_C P_D \quad (4.1)$$

Όπου:

L time units \rightarrow ο χρόνος μετάδοσης ενός πακέτου δεδομένων σε κανονικοποιημένες χρονικές μονάδες (data slot)

G (control packets/time unit) \rightarrow το μέση φορτίο πακέτων ελέγχου που επιχειρούν μετάδοση στο κανάλι ελέγχου σε μία χρονική μονάδα (control slot)

N \rightarrow το πλήθος των καναλιών δεδομένων

$S_T \rightarrow$ το ολικό throughput του συστήματος ανά data slot (η μέση τιμή των πακέτων που μεταδίδονται επιτυχώς στα κανάλια δεδομένων κατά τη διάρκεια L χρονικών μονάδων)

$S_D \rightarrow$ το throughput σε ένα κανάλι δεδομένων ανά data slot

$P_C = e^{-G}$ είναι η πιθανότητα επιτυχίας στο διάυλο ελέγχου (4.2)

$P_D = [1 - \frac{G}{N} e^{-G}]^{L-1}$ [6] είναι η πιθανότητα επιτυχούς μετάδοσης σε κάποιο από τα κανάλια δεδομένων. (4.3)

Άρα, το throughput, δηλαδή η μέση τιμή του πλήθους των επιτυχώς μεταδιδόμενων πακέτων δεδομένων κατά τη διάρκειας ενός data slot, σε κάποιο κανάλι δεδομένων, από τις σχέσεις (4.1), (4.2) και (4.3) προκύπτει:

$$S_D = L \frac{G}{N} e^{-G} \left[1 - \frac{G}{N} e^{-G} \right]^{L-1} \quad (4.4)$$

Στο γράφημα 4.1, που ακολουθεί, αναπαρίσταται το throughput, S_D (packets/data slot) ανά κανάλι δεδομένων, συναρτήσει του προσφερόμενου φορτίου, G (control packets/unit).

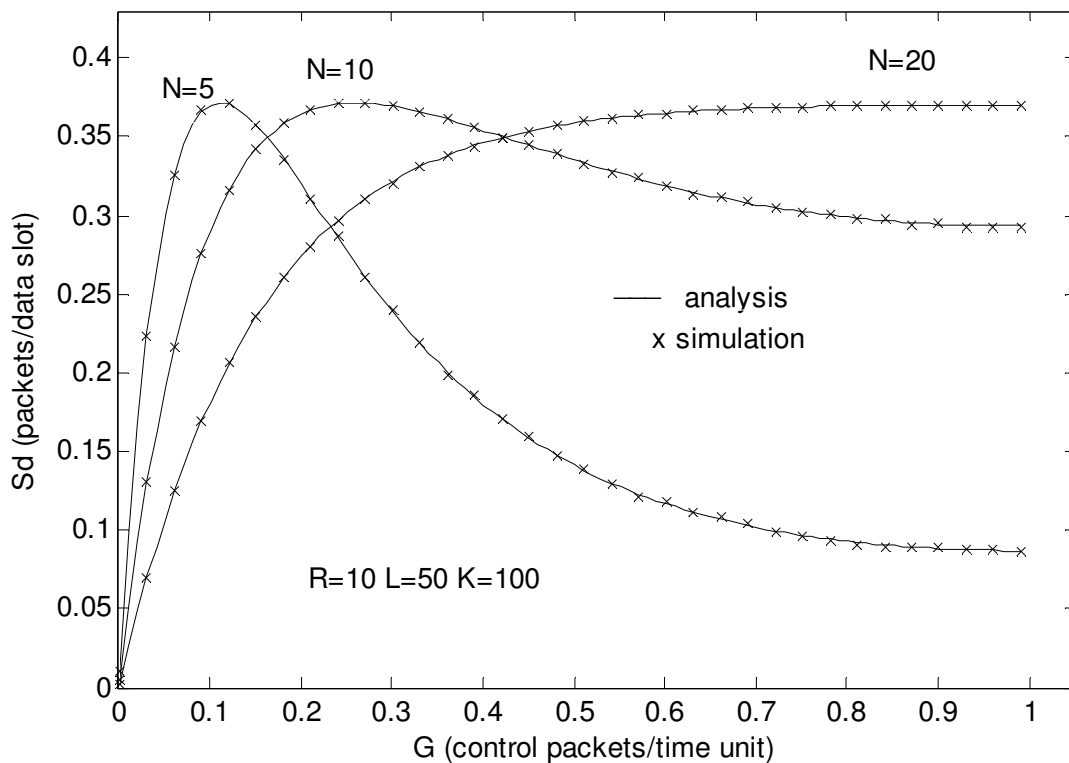


Chart 4.1: The throughput, S_D (packets/data slot) versus the offered load, G (control packets/time unit) for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units.

Ανάλυση της καθυστέρησης

Η καθυστέρηση ορίζεται ως το χρονικό διάστημα μεταξύ της γέννησης ενός πακέτου και της επιτυχούς μετάδοσής του μέσα από το δίαυλο δεδομένων, [1] και [5].

$$D = D_W + D_r + (2R + 1) \times L + 1 \quad (4.5)$$

$(2R+1) \times L + 1 = 1 + R \times L + L + R \times L \rightarrow 1 + R \times L$ απαιτούμενος χρόνος για τη μετάδοση και διάδοση του πακέτου ελέγχου
 $\rightarrow L + R \times L$ ο απαιτούμενος χρόνος για τη μετάδοση και διάδοση του πακέτου δεδομένων

D_W = το διάστημα από τη στιγμή γέννησης ενός πακέτου έως την έναρξη του επόμενου control slot, με μέση τιμή:

$$E[D_W] = \frac{1}{T} \int_0^T t dt \xrightarrow{T=1 \text{ time unit}} E[D_W] = 1/2 \text{ time units} \quad (4.6)$$

D_r = η καθυστέρηση από τη χρονική στιγμή που θα μεταδοθεί το πακέτο ελέγχου για πρώτη φορά μέχρι την έναρξη της επιτυχούς μετάδοσης του πακέτου δεδομένων.

Η χρονική καθυστέρηση D_r εξαρτάται από το πόσες φορές θα εκπέμψει ένας σταθμός ένα πακέτο έως ότου καταφέρει την επιτυχή μετάδοσή του.

Έστω ότι ένας σταθμός καταβάλλει n προσπάθειες για να επιτύχει να μεταδώσει ένα πακέτο. Υποθέτοντας ότι η πιθανότητα επιτυχούς μετάδοσης του πακέτου είναι ίδια για κάθε προσπάθεια, η πιθανότητα επιτυχίας μετά από n προσπάθειες είναι:

$$Q(n) = P_{\text{suc}}(1 - P_{\text{suc}})^{n-1} \quad (4.7)$$

όπου P_{suc} είναι η πιθανότητα επιτυχούς μετάδοσης ενός πακέτου και για το συγκεκριμένο σύστημα

$$(4.2, 4.3) \rightarrow P_{\text{suc}} = P_C P_D = e^{-G} \left[1 - \frac{G}{N} e^{-G} \right]^{L-1} \quad (4.8)$$

όπου P_C είναι η πιθανότητα επιτυχούς μετάδοσης στο κανάλι ελέγχου και P_D είναι η πιθανότητα επιτυχούς μετάδοσης σε κάποιο κανάλι δεδομένων

Ο μέσος όρος των προσπαθειών n είναι:

$$A = E[n] = \sum_{n=1}^{\infty} n Q(n) = P_{\text{suc}}^{-1} \xrightarrow{(4.8)} A = e^G \left[1 - \frac{G}{N} e^{-G} \right]^{1-L} \quad (4.9)$$

Άρα, αν A είναι η μέση τιμή των προσπαθειών έως ότου σημειωθεί επιτυχής μετάδοση, $A-1$ είναι το πλήθος των αποτυχημένων εκπομπών κατά μέσο όρο (η τελευταία προσπάθεια είναι επιτυχής).

Σε κάθε αποτυχημένη προσπάθεια καταναλίσκονται $D_B + R \times L + 1$ time units (4.10)

Όπου:

D_B είναι η χρονική καθυστέρηση μεταξύ δύο διαδοχικών επαναμεταδόσεων πακέτων ελέγχου, δηλαδή ο χρόνος που ο σταθμός περιμένει έως ότου ξαναπροσπαθήσει να εκπέμψει το πακέτο του. Μετά από κάθε αποτυχημένη

εκπομπή κάθε σταθμός επιλέγει τυχαία μετά από πόσο χρόνο θα επανεκπέμψει, με ελάχιστο χρόνο αναμονής 1 time unit και μέγιστο K time units. Η μέση τιμή της καθυστέρησης D_B είναι:

$$E[D_B] = \frac{1}{K} \sum_{i=1}^K i = \frac{1}{K} \left[\frac{1}{2} K(K+1) \right] = \frac{(K+1)}{2} \quad (4.11)$$

$R \times L$ time units είναι η καθυστέρηση λόγω διάδοσης του κάθε πακέτου ελέγχου

1 (time unit) είναι η καθυστέρηση λόγω μετάδοσης του κάθε πακέτου ελέγχου

Από τις σχέσεις 4.9, 4.10 και 4.11 προκύπτει:

$$E[D_r] = (A-1)[E[D_B] + R \times L + 1] \quad (4.12)$$

$$\xrightarrow{(4.9)} E[D_r] = (e^G [1 - \frac{G}{N} e^{-G}]^{1-L} - 1) (E[D_B] + R \times L + 1) \xrightarrow{(4.11)}$$

$$E[D_r] = (e^G [1 - \frac{G}{N} e^{-G}]^{1-L} - 1) \left(\frac{(K+1)}{2} + R \times L + 1 \right) \quad (4.13)$$

Επομένως για τη μέση τιμή της καθυστέρησης ανά πακέτο από τις σχέσεις (4.5), (4.6), (4.12) και (4.13) :

$$E[D] = E[D_w + D_r + (2R+1) \times L + 1] \rightarrow$$

$$E[D] = E[D_w] + E[D_r] + (2R+1) \times L + 1 \xrightarrow{(4.6), (4.12)}$$

$$E[D] = \frac{1}{2} + (A-1)[E[D_B] + R \times L + 1] + (2R+1) \times L + 1 \xrightarrow{(4.9), (4.11)}$$

$$E[D] = \frac{1}{2} + (P_{suc}^{-1} - 1) \left[\frac{K+1}{2} + R \times L + 1 \right] + (2R+1) \times L + 1 \xrightarrow{(4.9)}$$

$$E[D] = \frac{1}{2} + \left(e^G \left[1 - \frac{G}{N} e^{-G} \right]^{1-L} - 1 \right) \left[\frac{K+1}{2} + R \times L + 1 \right] + (2R+1) \times L + 1 \quad (4.14)$$

Στο γράφημα 4.2 που ακολουθεί έχει παρασταθεί η μέση καθυστέρηση ανά πακέτο, $E[D]$ (time units/packet) συναρτήσε του throughput, (S_D), και στο γράφημα 4,3, συναρτήσε της κυκλοφορίας, (G), για $N=\{5, 10, 20\}$ data channels. Και στις δύο γραφικές παραστάσεις, όσο αυξάνεται ο αριθμός των προσφερόμενων καναλιών δεδομένων, η καθυστέρηση μειώνεται, κάτι που εύκολα διαπιστώνεται παρατηρώντας και την ανωτέρω σχέση, (4.14).

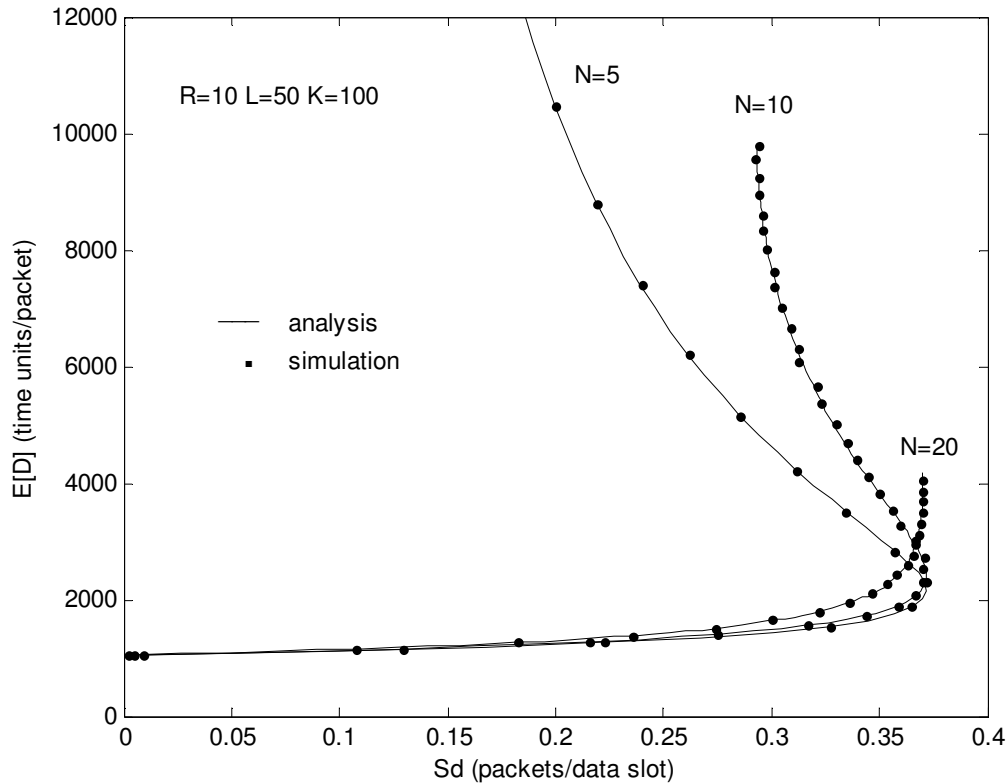


Chart 4.2: The Average Delay, $E[D]$ (time units/packet) versus the throughput S_d (packets/data slot) for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units.

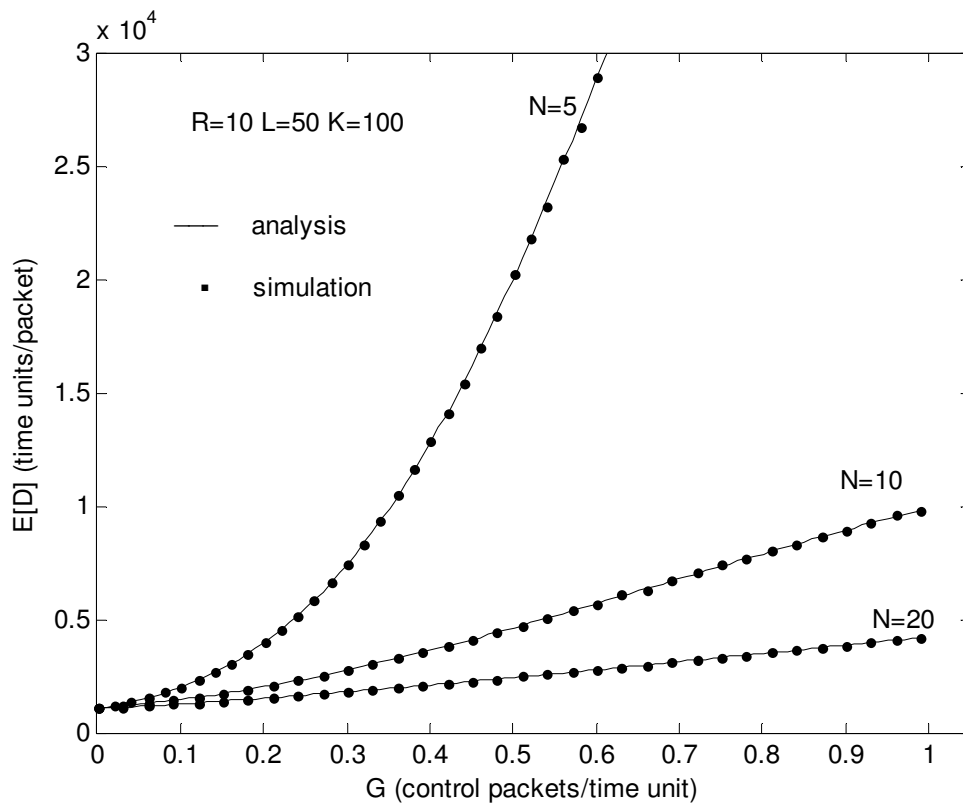


Chart 4.3: The Average Delay, $E[D]$ (time units/packet) versus the offered load G (control packets/time unit) for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units.

Μία πολύ σημαντική παράμετρος που επηρεάζει το μέγεθος της καθυστέρησης είναι το propagation delay, $R \times L$. Ακολουθούν γραφικές παραστάσεις για διάφορες τιμές της καθυστέρησης διάδοσης.

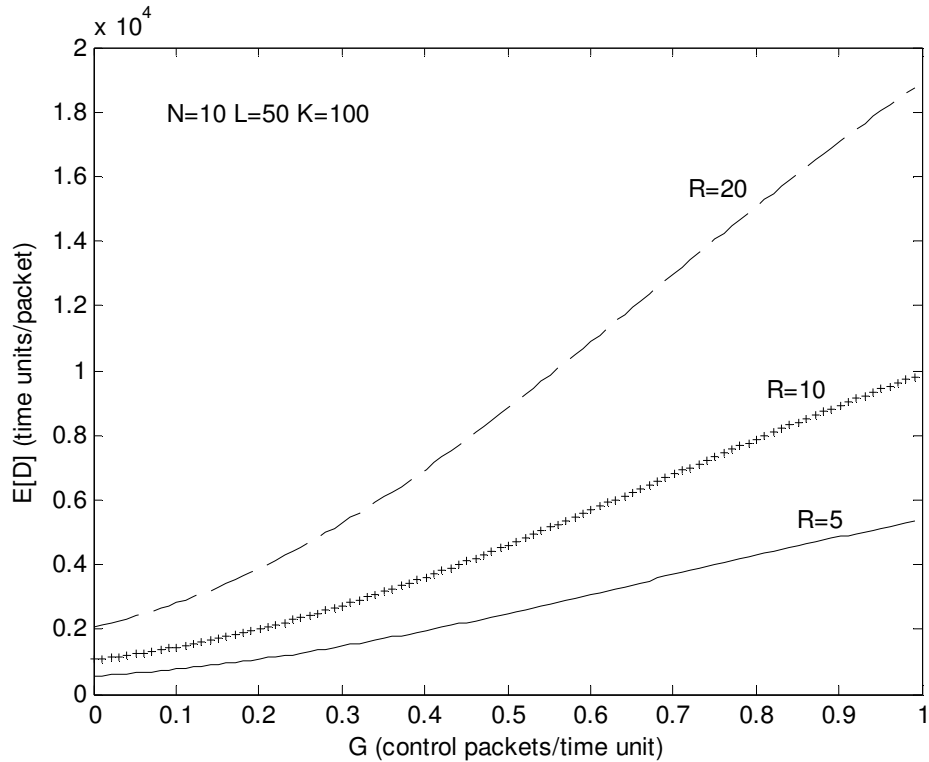


Chart 4.4: The Average Delay, $E[D]$ (time units/packet) versus the offered load G (control packets/time unit) for $R=\{5, 10, 20\}$ data slots, $N=10$ data channels, $L=50$ time units, $K=100$ time units.

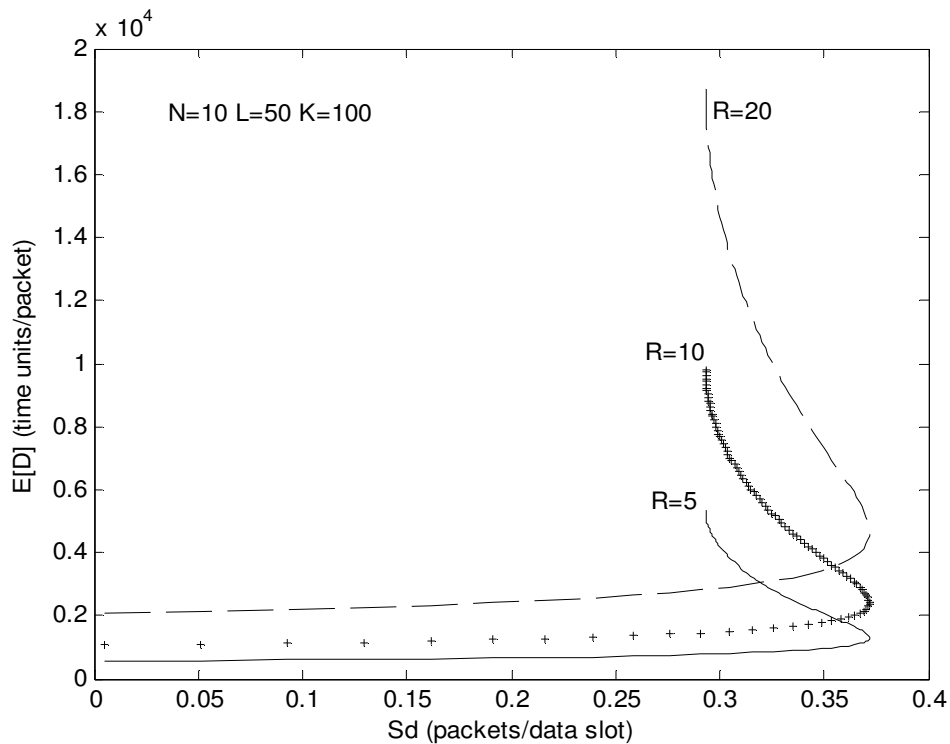


Chart 4.5: The Average Delay, $E[D]$ (time units/packet) versus the throughput S_D (packets/data slot) for $R=\{5, 10, 20\}$ data slots, $N=10$ data channels, $L=50$ time units, $K=100$ time units.

Ανάλυση με την επίδραση του Receiver Collisions Phenomenon

Προχωρώντας την ανάλυση και λαμβάνοντας υπόψη το φαινόμενο των receiver collisions, θεωρούμε ότι το συνολικό φορτίο G (control packets/cycle) προέρχεται ομοιόμορφα από πεπερασμένο αριθμό σταθμών, M .

Από [10] :

$S_T = GLP_c P_D$ είναι η μέση τιμή των πακέτων που μεταδίδονται επιτυχώς στα κανάλια δεδομένων κατά τη διάρκεια ενός data slot

$S_{RCtotal} = M \frac{S_T}{M} \left[1 - \frac{S_T}{ML} \right]^{L-1}$ (4.15) το ολικό throughput, λαμβάνοντας υπόψη και την επίδραση των receiver collisions, δηλαδή η μέση τιμή των πακέτων που έχουν παραλειφθεί επιτυχώς από τους σταθμούς – δέκτες, δεδομένου ότι έχουν μεταδοθεί επιτυχώς στα κανάλια δεδομένων, όπου M είναι το πλήθος των σταθμών στο σύστημα.

Άρα το throughput ανά κανάλι δεδομένων προκύπτει:

$$S_{RC} = \frac{S_T}{N} \left[1 - \frac{S_T}{ML} \right]^{L-1} \quad (4.16)$$

όπου από (4.1), (4.2) και (4.3) $S_T = LGe^{-G} \left[1 - \frac{G}{N} e^{-G} \right]^{L-1}$.

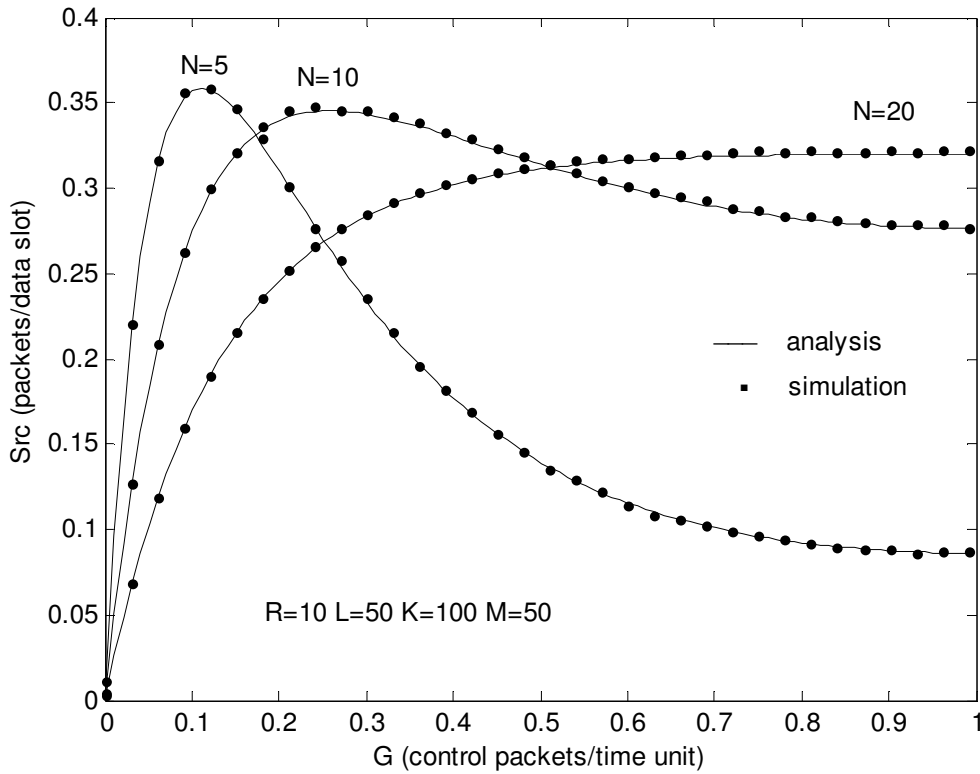


Chart 4.6: The throughput, S_{RC} (packets/data slot) versus the offered load, G (control packets/time unit) for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units.

Για να εστιάσουμε στη διαφοροποίηση των αποτελεσμάτων στη μελέτη με και χωρίς Receiver Collisions Phenomenon προχωρούμε με το παρακάτω συγκριτικό διάγραμμα που απεικονίζεται ξανά το throughput συναρτήσει της κυκλοφορίας.

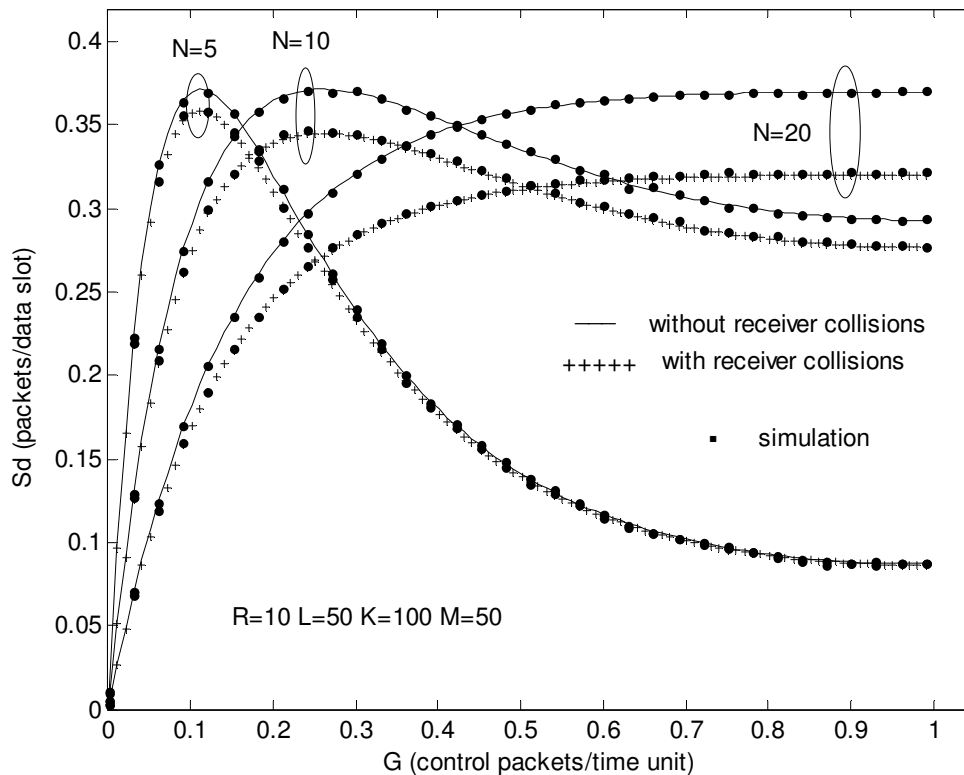


Chart 4.7: The throughput, S_D (packets/data slot) versus the offered load, G (control packets/time unit) with and without receiver collisions analysis and simulation for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units.

Παρατηρούμε ότι όσο αυξάνεται το πλήθος των καναλιών τόσο αυξάνει και η απόκλιση στις αντίστοιχες καμπύλες. Όσο τα διαθέσιμα για μετάδοση κανάλια αυξάνουν, τόσο αυξάνουν και τα επιτυχώς μεταδιδόμενα πακέτα και συνεκδοχικά αυξάνει και η πιθανότητα απόρριψης αυτών από τους σταθμούς – δέκτες.

Μέσης Πιθανότητας Απόρριψης

Τα ως άνω γίνονται ευκρινέστερα μέσω της γραφικής αναπαράστασης της Μέσης Πιθανότητας Απόρριψης (Average Rejection Probability, P_{rej}) συναρτήσει της κυκλοφορίας, όπου από το [10]:

$$P_{rej} = \frac{S_{total} - S_{RCtotal}}{S_{total}} \quad (4.17)$$

Τα S_{total} και $S_{RCtotal}$ δίνονται από τις σχέσεις (4.1) και (4.15)

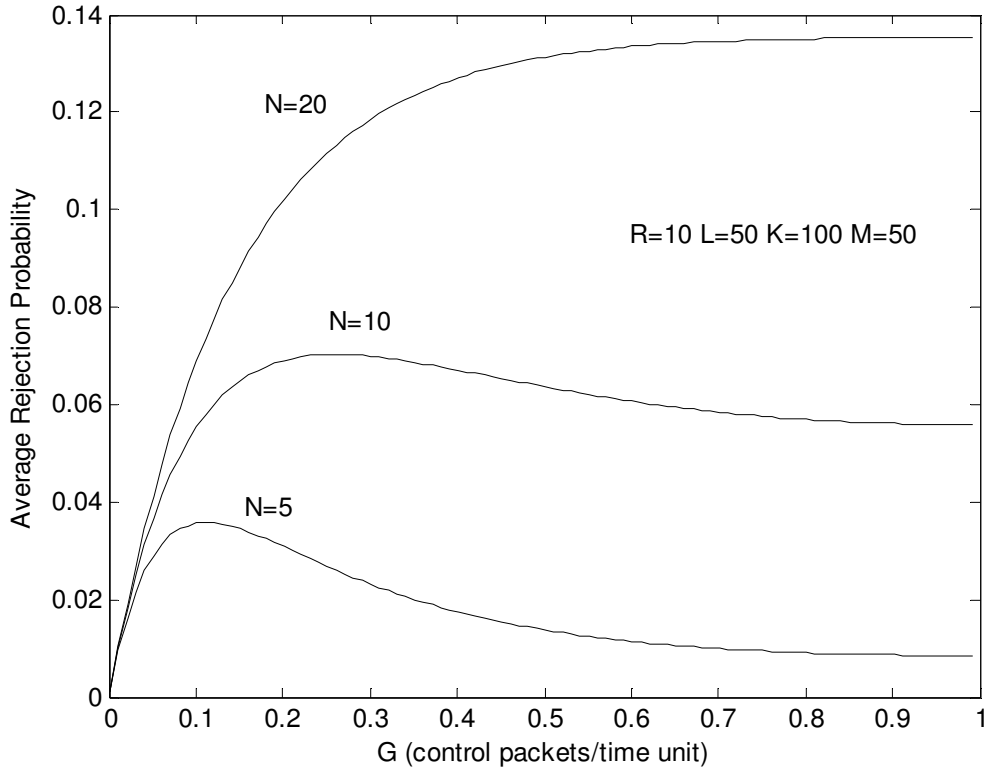


Chart 4.8: Average Rejection Probability versus the offered load, G (control packets/time unit) for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units and $M=50$ stations.

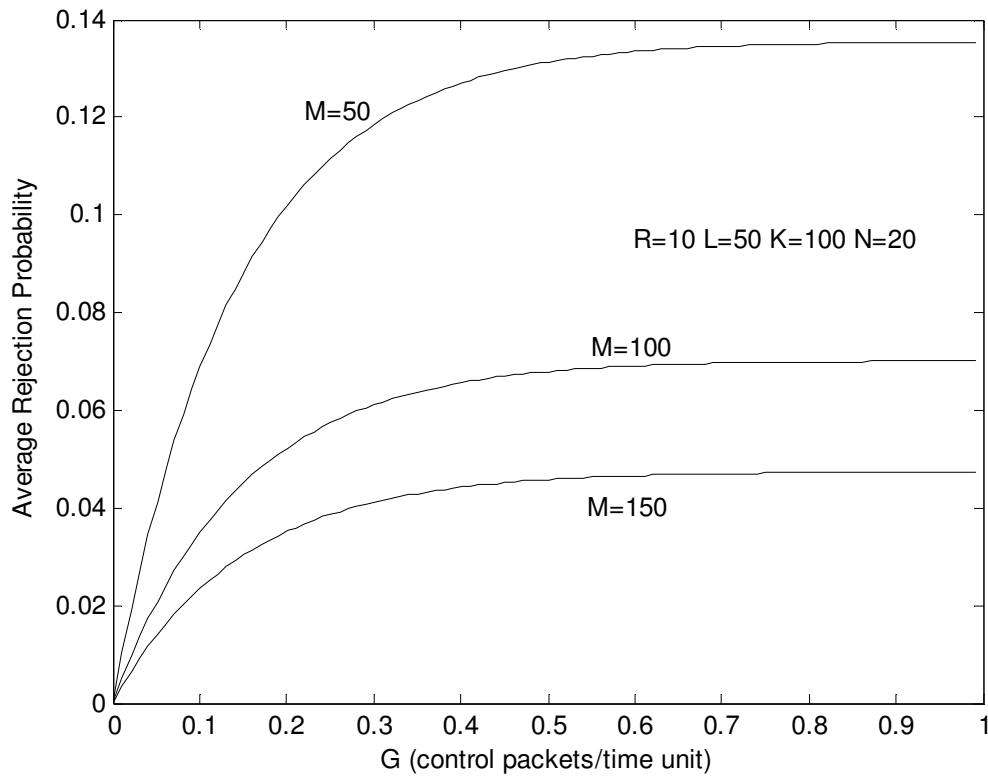


Chart 4.9: Average Rejection Probability versus the offered load, G (control packets/time unit) for $M=\{50, 100, 150\}$ stations, $N=20$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units.

Ενώ αυξάνει το πλήθος των καναλιών δεδομένων, αυξάνει και η πιθανότητα απόρριψης, άμεση συνέπεια της αύξησης της διέλευσης από τα κανάλια δεδομένων. Έχοντας υπόψη τη μεταβολή της διέλευσης συναρτήσε του προσφερόμενου φορτίου, και σε σχέση με το πλήθος N , παρατηρεί κανείς ότι οι καμπύλες της μεταβολής της P_{rej} έχουν ανάλογη μορφή με αυτήν της μεταβολής της διέλευσης, άμεση συνέπεια της σχέσης (4.17) που δίνει την P_{rej} .

Με την αύξηση όμως του πλήθους των σταθμών (βλ. και Chart 4.9) η Μέση Πιθανότητα Απόρριψης μειώνεται, όπως ήταν αναμενόμενο. Η πιθανότητα να επιλεγεί κάποιος σταθμός ως τελικός προορισμός είναι αντιστρόφως ανάλογη του πλήθους των σταθμών, $1/M$. Όσο περισσότεροι είναι οι σταθμοί, τόσο μικρότερη είναι η πιθανότητα να βρεθεί κατελιμμένος ο σταθμός που έχει επιλέξει ένα επιτυχές στα data channels πακέτο.

Στο διάγραμμα που ακολουθεί αναπαρίσταται η Μέση Πιθανότητα Απόρριψης τόσο για διαφορετικό πλήθος σταθμών, όσο και για διαφορετικό πλήθος καναλιών.

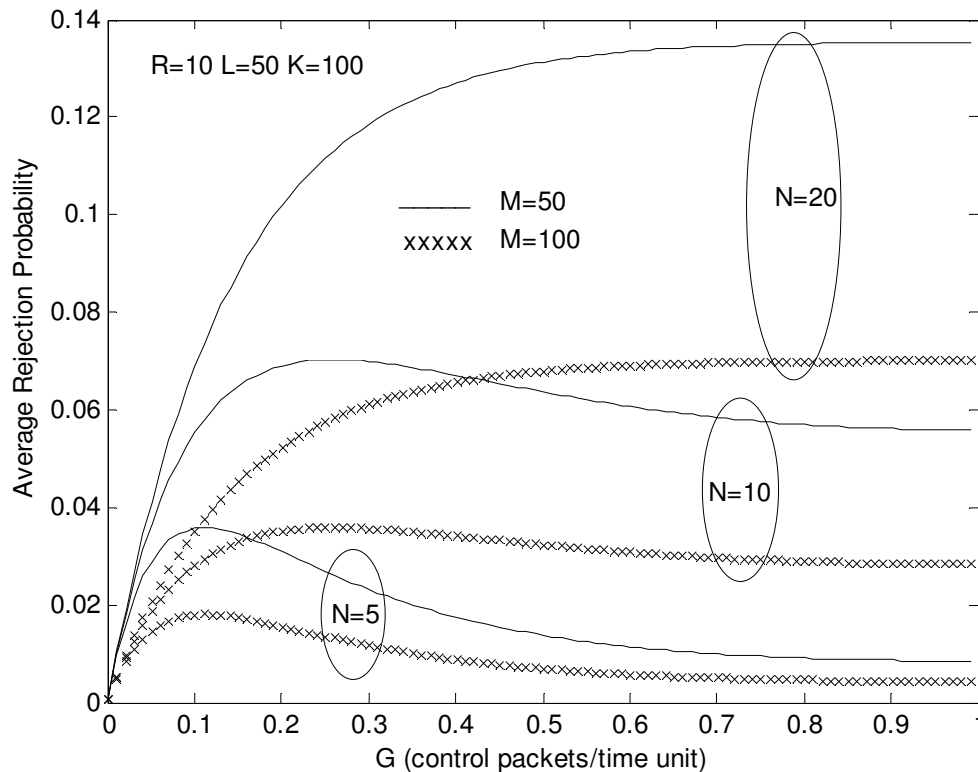


Chart 4.10: Average Rejection Probability versus the offered load, G (control packets/time unit) for $M=\{50, 100\}$ stations, $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units.

Στο παρακάτω ραβδόγραμμα παριστάνονται οι μέγιστες τιμές της Μέσης Πιθανότητας Απόρριψης για διάφορα N και διάφορα M .

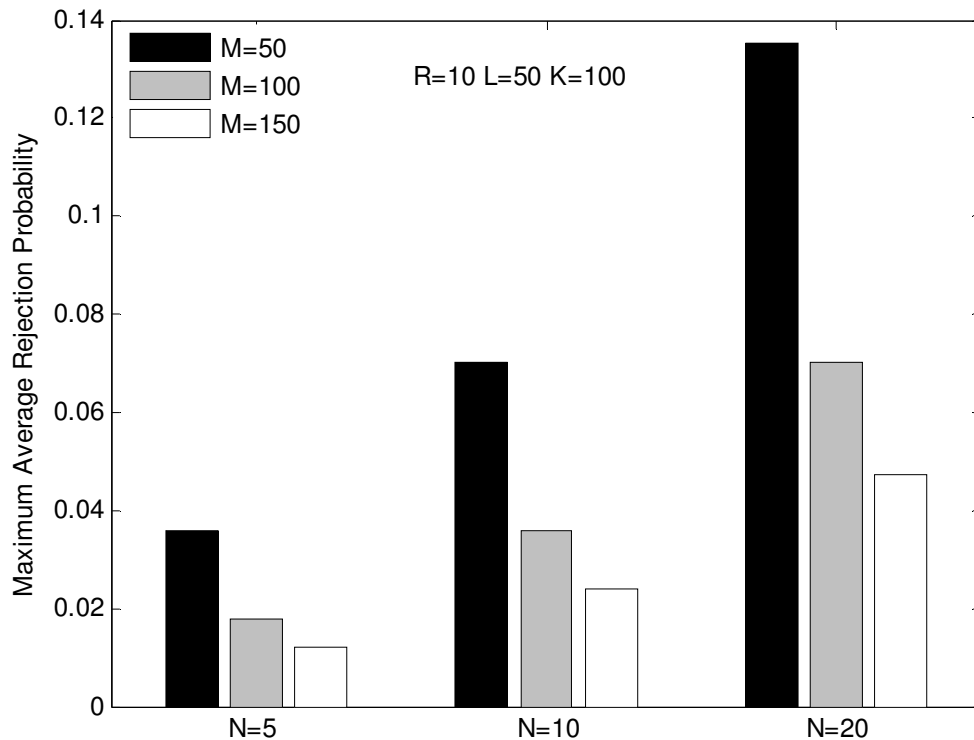


Chart 4.11: Maximum Average Rejection Probability for $M=\{50, 100, 150\}$ stations, $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units.

Ανάλυση της καθυστέρησης

Η ανάλυση της καθυστέρησης λαμβάνοντας υπόψη το φαινόμενο των συγκρούσεων στους δέκτες, ακολουθεί τη λογική της ανάλυσης που προηγήθηκε για την περίπτωση που το φαινόμενο δε λαμβάνεται υπόψη. Στην παρούσα ανάλυση, με τον όρο «επιτυχή μετάδοση πακέτου», εννοούμε πάντα «επιτυχή μετάδοση και αποδοχή ενός πακέτου από το δέκτη του σταθμού – προορισμού».

Η καθυστέρηση ορίζεται ως το χρονικό διάστημα μεταξύ της γέννησης ενός πακέτου και της επιτυχούς παραλαβής του από το δέκτη του σταθμού προορισμού.

$$D = D_w + D_r + (2R + 1) \times L + 1 \quad (4.18)$$

$$(2R+1) \times L + 1 = 1 + R \times L + L + R \times L \rightarrow 1 + R \times L \text{ απαιτούμενος χρόνος για τη μετάδοση και διάδοση του πακέτου ελέγχου}$$

$$\rightarrow L + R \times L \text{ ο απαιτούμενος χρόνος για τη μετάδοση και διάδοση του πακέτου δεδομένων}$$

D_w = το διάστημα από τη στιγμή γέννησης ενός πακέτου έως την έναρξη του επόμενου control slot, με μέση τιμή:

$$E[D_w] = \frac{1}{T} \int_0^T t dt \xrightarrow{T=1 \text{ time unit}} E[D_w] = 1/2 \text{ time units} \quad (4.19)$$

D_r = η καθυστέρηση από τη χρονική στιγμή που θα μεταδοθεί το πακέτο ελέγχου για πρώτη φορά μέχρι την έναρξη της επιτυχούς μετάδοσης του πακέτου δεδομένων.

Η χρονική καθυστέρηση D_r εξαρτάται από το πόσες φορές θα εκπέμψει ένας σταθμός ένα πακέτο έως ότου καταφέρει την επιτυχή μετάδοσή του.

Έστω ότι ένας σταθμός καταβάλλει n προσπάθειες για να επιτύχει να μεταδώσει ένα πακέτο. Υποθέτοντας ότι η πιθανότητα επιτυχούς μετάδοσης του πακέτου είναι ίδια για κάθε προσπάθεια, η πιθανότητα επιτυχίας μετά από n προσπάθειες είναι:

$$Q(n) = P_{suc}(1 - P_{suc})^{n-1} \quad (4.20)$$

όπου P_{suc} είναι η πιθανότητα επιτυχούς μετάδοσης ενός πακέτου και για το συγκεκριμένο σύστημα

$$P_{suc} = \frac{S_{RCtotal}}{LG} \quad (4.15)$$

$$P_{suc} = \frac{M \frac{S_T}{M} \left[1 - \frac{S_T}{ML}\right]^{L-1}}{LG} \quad (4.1)$$

$$P_{suc} = e^{-G} \left[1 - \frac{G}{N} e^{-G}\right]^{L-1} \left[1 - \frac{Ge^{-G} \left[1 - \frac{G}{N} e^{-G}\right]^{L-1}}{M}\right]^{L-1} \quad (4.21)$$

Ο μέσος όρος των προσπαθειών n είναι:

$$A = E[n] = \sum_{n=1}^{\infty} nQ(n) = P_{suc}^{-1} \quad (4.21)$$

$$A = e^G \left[1 - \frac{G}{N} e^{-G}\right]^{1-L} \left[1 - \frac{Ge^{-G} \left[1 - \frac{G}{N} e^{-G}\right]^{L-1}}{M}\right]^{1-L} \quad (4.22)$$

Άρα, αν A είναι η μέση τιμή των προσπαθειών έως ότου σημειωθεί επιτυχής μετάδοση και αποδοχή, $A-1$ είναι το πλήθος των αποτυχημένων εκπομπών κατά μέσο όρο (η τελευταία προσπάθεια είναι επιτυχής).

Σε κάθε αποτυχημένη προσπάθεια καταναλίσκονται $D_B + R \times L + 1$ time units (4.23)

Όπου:

D_B είναι η χρονική καθυστέρηση μεταξύ δύο διαδοχικών επαναμεταδόσεων πακέτων ελέγχου, δηλαδή ο χρόνος που ο σταθμός περιμένει έως ότου ξαναπροσπαθήσει να εκπέμψει το πακέτο του. Μετά από κάθε αποτυχημένη εκπομπή κάθε σταθμός επιλέγει τυχαία μετά από πόσο χρόνο θα επανεκπέμψει, με ελάχιστο χρόνο αναμονής 1 time unit και μέγιστο K time units. Η μέση τιμή της καθυστέρησης D_B είναι:

$$E[D_B] = \frac{1}{K} \sum_{i=1}^K i = \frac{1}{K} \left[\frac{1}{2} K(K+1) \right] = \frac{(K+1)}{2} \quad (4.24)$$

$R \times L$ time units είναι η καθυστέρηση λόγω διάδοσης του κάθε πακέτου ελέγχου

1 (time unit) είναι η καθυστέρηση λόγω μετάδοσης του κάθε πακέτου ελέγχου

Από τις σχέσεις 4.22, 4.23 και 4.24 προκύπτει:

$$E[D_r] = (A-1)[E[D_B] + R \times L + 1] \quad (4.25) \quad \xrightarrow{(4.22)}$$

$$E[D_r] = \left(e^G \left[1 - \frac{G}{N} e^{-G} \right]^{1-L} \left[1 - \frac{Ge^{-G} \left[1 - \frac{G}{N} e^{-G} \right]^{L-1}}{M} \right]^{1-L} - 1 \right) (E[D_B] + R \times L + 1) \quad (4.24)$$

$$E[D_r] = \left(e^G \left[1 - \frac{G}{N} e^{-G} \right]^{1-L} \left[1 - \frac{Ge^{-G} \left[1 - \frac{G}{N} e^{-G} \right]^{L-1}}{M} \right]^{1-L} - 1 \right) \times \left(\frac{(K+1)}{2} + R \times L + 1 \right) \quad (4.26)$$

Επομένως για τη μέση τιμή της καθυστέρησης ανά πακέτο από τις σχέσεις (4.18), (4.19), (4.25) και (4.26) :

$$E[D] = E[D_w + D_r + (2R+1) \times L + 1] \rightarrow$$

$$E[D] = E[D_w] + E[D_r] + (2R+1) \times L + 1 \quad \xrightarrow{(4.19), (4.25)}$$

$$E[D] = \frac{1}{2} + (A-1)[E[D_B] + R \times L + 1] + (2R+1) \times L + 1 \quad \xrightarrow{(4.22), (4.24)}$$

$$E[D] = \frac{1}{2} + (P_{\text{suc}}^{-1} - 1) \left[\frac{K+1}{2} + R \times L + 1 \right] + (2R+1) \times L + 1 \quad \xrightarrow{(4.22), (4.26)}$$

$$E[D] = \frac{1}{2} + \left(e^G \left[1 - \frac{G}{N} e^{-G} \right]^{1-L} \left[1 - \frac{Ge^{-G} \left[1 - \frac{G}{N} e^{-G} \right]^{L-1}}{M} \right]^{1-L} - 1 \right) \left[\frac{K+1}{2} + R \times L + 1 \right] + (2R+1) \times L + 1 \quad (4.27)$$

Στα γραφήματα που ακολουθούν αναπαρίσταται η μέση καθυστέρηση συναρτήσει του προσφερόμενου φορτίου για κυμαινόμενο N .

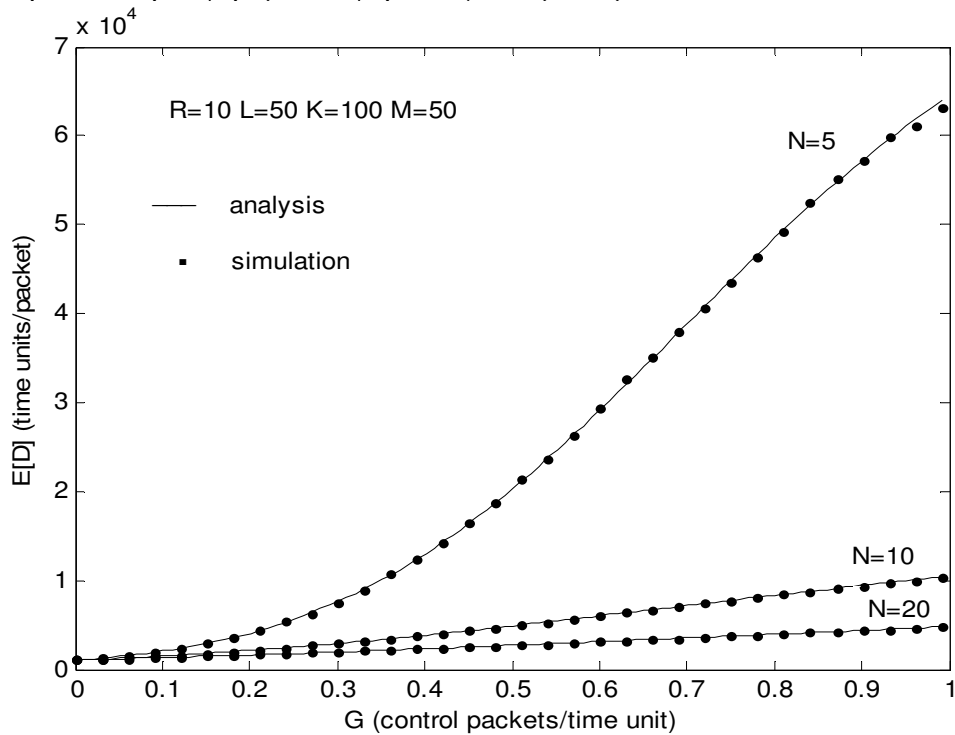


Chart 4.12: Average Delay $E[D]$ (time units/packet) versus the offered load, G (control packets/time unit) for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units, $M=50$ stations.

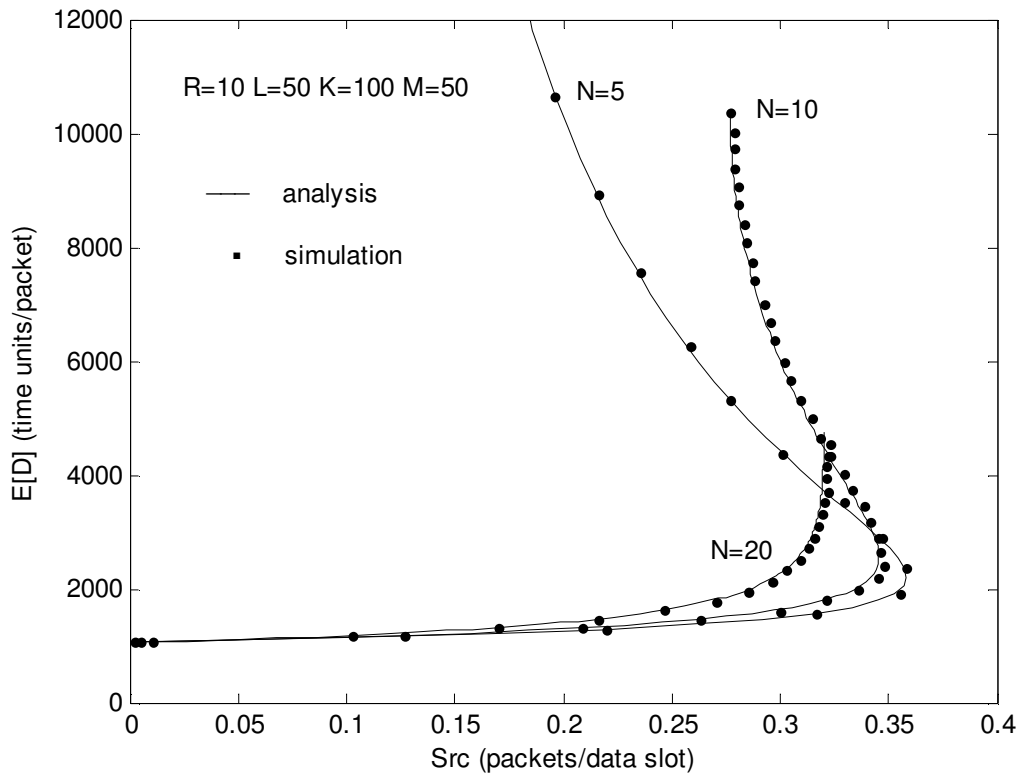


Chart 4.13: Average Delay $E[D]$ (time units/packet) versus the throughput, S_{RC} (packets/data slot) for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units, $M=50$ stations.

Για να εστιάσουμε στη διαφοροποίηση των αποτελεσμάτων στη μελέτη με και χωρίς Receiver Collisions Phenomenon προχωρούμε σε συγκριτικά διαγράμματα της μέσης καθυστέρησης.

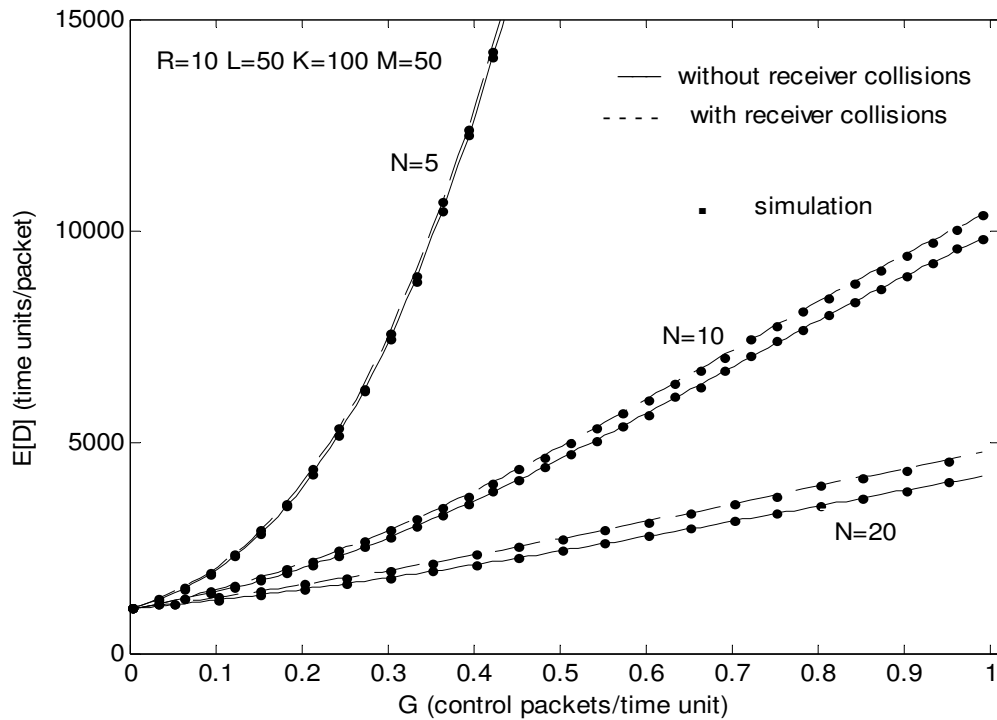


Chart 4.14: Average Delay $E[D]$ (time units/packet) versus the average load, G (control packets/time unit) with and without Receiver Collisions analysis and simulation, for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units, $M=50$ stations.

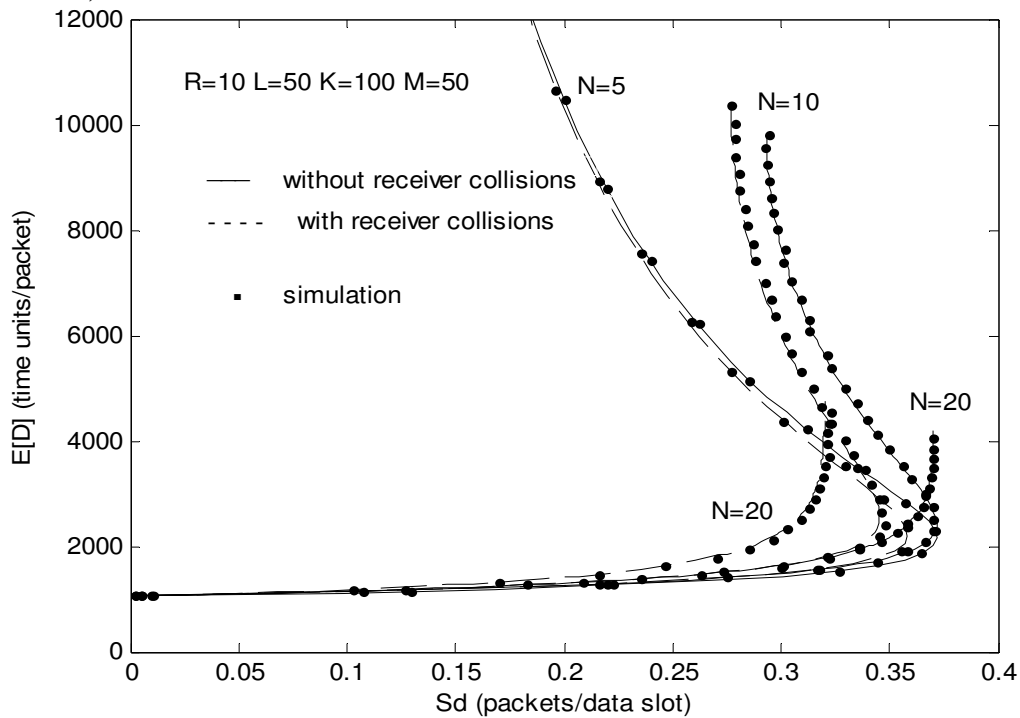


Chart 4.15: Average Delay $E[D]$ (time units/packet) versus the throughput, S_D (packets/data slot) with and without Receiver Collisions analysis and simulation, for $N=\{5, 10, 20\}$ data channels, $L=50$ time units, $R=10$ data slots, $K=100$ time units, $M=50$ stations.

Κεφάλαιο 5

Μελλοντική έρευνα

Το 1989 οι Will Leland και Daniel V. Wilson [12], ερευνητές στα εργαστήρια του Bellcore (Bell Communication Research), είχαν στη διάθεσή τους ένα μόνιτορ υψηλής ανάλυσης τόσο σε επίπεδο λογισμικού όσο και σε επίπεδο υλικού. Έτσι μπορούσαν να καταγράφουν και να συλλέγουν ίχνη πακέτων από Ethernet LANs χωρίς απώλειες. Στην ομάδα αυτή εντάχθηκε και ο W. Willinger και παρατηρώντας τα δεδομένα που είχαν πλέον από την καταγραφή των δικτύων του Bellcore, κατέληξαν στο συμπέρασμα ότι η μετρούμενη κίνηση παρουσίαζε χαρακτηριστικά self – similar διαδικασιών, δηλαδή ο ρυθμός των πακέτων που μεταδίδονταν στο δίκτυο ήταν περίπου ο ίδιος είτε η χρονική μονάδα αναφοράς ήταν της τάξης των milliseconds, είτε δευτερολέπτων, είτε λεπτών είτε και ακόμη μεγαλύτερη.

Διαδικασίες Self – Similar

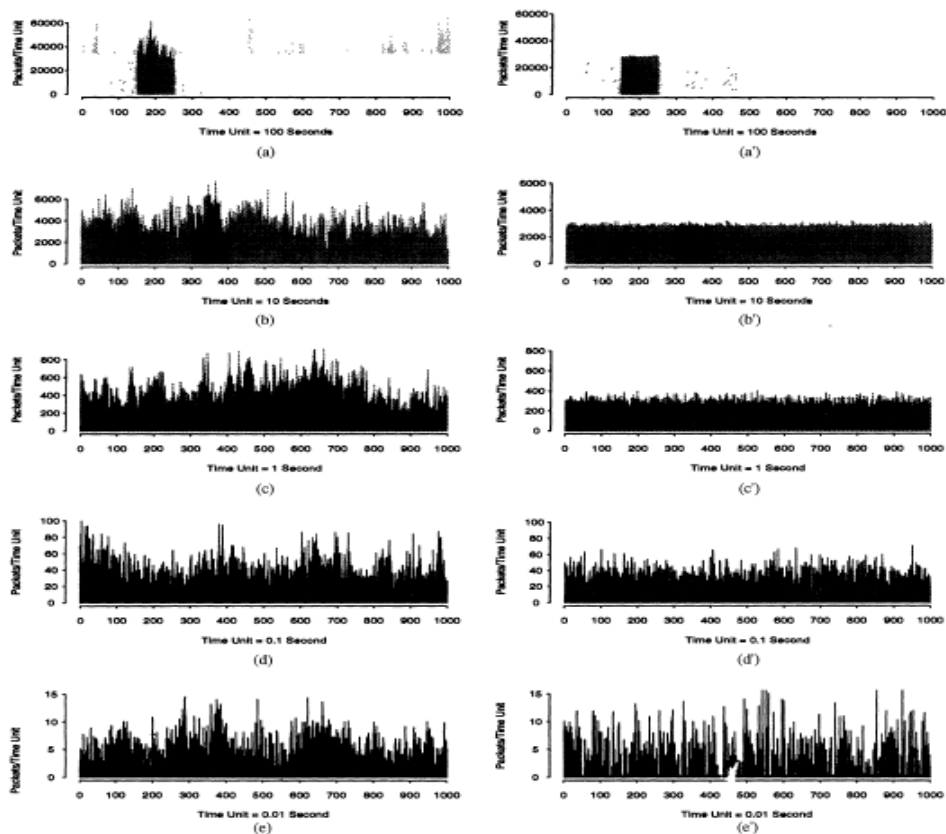
Έστω $X=(X_t: t=0,1,2,\dots)$ στοχαστική διαδικασία με μέση τιμή μ , απόκλιση σ^2 , και συνάρτηση αυτοσυσχέτισης $r(k) \sim k^{-\beta} L_1(k)$ as $k \rightarrow \infty$ [13]
Για $m=1,2,3,\dots$, ορίζουμε μία νέα σειρά, την

$$X_k^{(m)} = \frac{1}{m} (X_{km-m+1} + \dots + X_{km}), k = 1,2,3 \dots$$

που προκύπτει από τους μέσους όρους μη επικαλυπτόμενων μπλοκ μεγέθους m της αρχικής σειράς.

Η διαδικασία X ονομάζεται self – similar με παράμετρο $H=1-\beta/2$ αν για όλα τα $m=1,2,3,\dots$ η προκύπτουσα διαδικασία $X_k^{(m)}$ έχει την ίδια απόκλιση και την ίδια αυτοσυσχέτιση, όπως η X .

Στην εικόνα που ακολουθεί, αναπαρίσταται Ethernet κίνηση (packets per time units) σε πέντε διαφορετικές χρονικές κλίμακες και συνθετική Poisson κίνηση στις ίδιες χρονικές κλίμακες.



Indication of self-similarity: Ethernet traffic (packets per time unit) on five different time scales (a)–(e); for comparison, synthetic traffic from an appropriately chosen compound Poisson model on the same five different time scales (a')–(e'). (Different gray levels are used to identify the same segments of traffic on the different time scales.)

Εικ. 10 Self – similar traffic and Poisson Traffic:
A comparison Figure [13]

Παρατηρεί κανείς ότι ενώ η κίνηση είναι στην πραγματικότητα εκρηκτική και στις 5 διαφορετικές κλίμακες, η συνθετική κίνηση που προκύπτει από το μοντέλο Poisson δε δίνει τα ίδια αποτελέσματα.

Το 1995, οι Vern Paxson και Sally Floyd στο άρθρο τους με τον χαρακτηριστικό τίτλο: “Wide - Area Traffic: The Failure of Poisson Modeling”, ερευνούν ένα μεγάλο αριθμό διαδικασιών αφίξεων πακέτων, με σκοπό να αναγνωρίσουν το λάθος που προκύπτει με τη χρήση του μοντέλου Poisson. Σημειώνουν ότι παρόλο που η κίνηση θεωρείται συνήθως ότι ακολουθεί την κατανομή Poisson, στα τοπικά δίκτυα αλλά και σε δίκτυα ευρείας περιοχής η κίνηση αυτή μοντελοποιείται πολύ καλύτερα με τη χρήση self – similar διαδικασιών[14].

Υιοθετώντας το μοντέλο των On – Off πηγών είναι δυνατόν να γεννήσουμε κίνηση που να διαθέτει self – similar χαρακτηριστικά. Θεωρούμε ότι οι σταθμοί για κάποια χρονικά διαστήματα, οι ON περίοδοι, εκπέμπουν ανελλιπώς πακέτα χωρίς χρονικά κενά μεταξύ των εκπομπών, ενώ σιωπούν στο ενδιάμεσο των ON περιόδων – OFF περίοδοι. Οι ON και OFF περίοδοι εναλλάσσονται αυστηρά, μια ON περίοδος ακολουθείται πάντα από μία OFF περίοδο και το αντίστροφο.

Κατανομή Pareto για τη γέννηση Self – Similar κίνησης

Μία μαθηματική κατανομή που δύναται να χρησιμοποιηθεί για τη γέννηση της διάρκειας των ON και OFF περιόδων των σταθμών που είναι συνδεδεμένοι σε κάποιο σύστημα είναι η κατανομή Pareto. Όταν οι παράμετροι αυτής της κατανομής κυμαίνονται σε συγκεκριμένο εύρος τιμών, τότε η κίνηση που προκύπτει έχει self – similar χαρακτηριστικά.

Η γέννηση τυχαίων τιμών της τυχαίας μεταβλητής της Pareto γίνεται με εφαρμογή του θεωρήματος που ακολουθεί [15].

Θεώρημα

Αν η F είναι συνεχής συνάρτηση κατανομής πιθανότητας (cumulative distribution function, CDF), η οποία είναι γνησίως αύξουσα και U είναι τυχαία μεταβλητή που ακολουθεί την Ομοιόμορφη Κατανομή (Uniform Distribution), στο διάστημα $(0,1)$ τότε:

$$X=F^{-1}(U)$$

είναι μία τυχαία μεταβλητή που έχει συνάρτηση κατανομής πιθανότητας την F .

Η CDF της κατανομής Pareto είναι:

$$F_X(x) = 1 - \left(\frac{k}{x}\right)^a \quad x \in [k, +\infty)$$

όπου k είναι παράμετρος που δίνει την ελάχιστη τιμή της X και ονομάζεται scale parameter, $k \in (0, +\infty)$

a είναι παράμετρος που καθορίζει την κλίση της PDF και CDF και ονομάζεται shape parameter, $a \in (0, +\infty)$

Η CDF_{PARETO} είναι γνησίως αύξουσα συνάρτηση:

Έστω x_1, x_2 δύο τυχαίες τιμές της τυχαίας Pareto μεταβλητής X , με $0 < x_1 < x_2$, τότε:

$$\begin{aligned} x_1 < x_2 &\rightarrow \frac{1}{x_1} > \frac{1}{x_2} \Rightarrow \frac{k}{x_1} > \frac{k}{x_2} \Rightarrow \left(\frac{k}{x_1}\right)^a > \left(\frac{k}{x_2}\right)^a \rightarrow -\left(\frac{k}{x_1}\right)^a < -\left(\frac{k}{x_2}\right)^a \\ &\rightarrow 1 - \left(\frac{k}{x_1}\right)^a < 1 - \left(\frac{k}{x_2}\right)^a \rightarrow F_X(x_1) < F_X(x_2) \end{aligned}$$

Άρα με εφαρμογή του θεωρήματος και αντιστροφή της συνάρτησης κατανομής πιθανότητας της Pareto κατανομής:

$$X_{PARETO} = F_{PARETO}^{-1}(U) = \left[1 - \left(\frac{k}{x}\right)^a\right]^{-1/a} = k(1 - U)^{-1/a} \rightarrow$$

$$X_{PARETO} = k(1 - U)^{-1/a} U \in (0,1)$$

Μοντελοποίηση ON περιόδων:

Αν είναι γνωστά τα μεγέθη

$1 < a_{ON} < 2$ η shape παράμετρος της κατανομής Pareto για τις ON περιόδους.

k_{ON} Η ελάχιστη διάρκεια που μπορεί να έχει μία On περίοδος

τότε η διάρκεια των ON περιόδων προκύπτει:

$$X_{ON_PARETO} = k_{ON}(1 - U)^{-1/\alpha_{ON}}$$

Μοντελοποίηση OFF περιόδων:

Αν είναι γνωστά τα μεγέθη

$1 < \alpha_{OFF} < 2$ η share παράμετρος της κατανομής Pareto για τις OFF περιόδους

k_{OFF} Η ελάχιστη διάρκεια που μπορεί να έχει μία Off περίοδος

τότε η διάρκεια των OFF περιόδων είναι:

$$X_{OFF_PARETO} = k_{OFF}(1 - U)^{-1/\alpha_{OFF}}$$

Από τα [13], [14], [15], [16] ο μελετητής μπορεί να βρει τιμές για τις παραμέτρους α_{ON} , k_{ON} , α_{OFF} και k_{OFF} .

Ο C. Jelger στο [16] μελετάει δίκτυο με $\alpha_{ON}=1.4$, $k_{ON}=12$ (σε msec) και $\alpha_{OFF}=1.2$. Για την εκτίμηση της παραμέτρου k_{OFF} με γνωστά τα υπόλοιπα μεγέθη και με δεδομένο φορτίο ακολουθείται η παρακάτω μεθοδολογία.

Έστω δίκτυο με M σταθμούς οι οποίοι προσφέρουν ίδια μέση κίνηση, G_i (packets/time unit). Η συνολική κίνηση είναι επομένως:

$$G = M \times G_i$$

Ορίζεται ως frame το χρονικό διάστημα από την έναρξη μίας on περιόδου έως τη λήξη της αμέσως επόμενης off περιόδου με μέση τιμή, $\mu_{frame} = \mu_{ON} + \mu_{OFF}$ time units, όπου μ_{ON} είναι η μέση διάρκεια των ON περιόδων και μ_{OFF} είναι η μέση διάρκεια των OFF περιόδων. Η μέση τιμή της On περιόδου κάθε σταθμού είναι (μέση τιμή της κατανομής Pareto):

$$E[X_{ON}] = \mu_{ON} = \alpha_{ON} k_{ON} (\alpha_{ON} - 1)^{-1} \text{time units}$$

Άρα κάθε σταθμός εκπέμπει κατά μέσο όρο πακέτα για μ_{ON} time units σε κάθε frame, δηλαδή σε $\mu_{frame} = \mu_{ON} + \mu_{OFF}$ time units, άρα:

$$G_i = \frac{\mu_{ON}}{\mu_{ON} + \mu_{OFF}} \frac{\text{packets}}{\text{time unit}} \rightarrow \mu_{OFF} = \frac{(1 - G_i)\mu_{ON}}{G_i} \text{time units}$$

$$\mu_{OFF} = \alpha_{OFF} k_{OFF} (\alpha_{OFF} - 1)^{-1} \text{time units} \rightarrow k_{OFF} = \mu_{OFF} \frac{\alpha_{OFF} - 1}{\alpha_{OFF}} \frac{\mu_{OFF} = \frac{(1 - G_i)\mu_{ON}}{G_i}}{\mu_{OFF}}$$

$$k_{OFF} = \frac{(1 - G_i)\mu_{ON}}{G_i} \frac{\alpha_{OFF} - 1}{\alpha_{OFF}} \frac{G_i = \frac{G}{M}, \mu_{ON} = \alpha_{ON} k_{ON} (\alpha_{ON} - 1)^{-1}}{\mu_{OFF}}$$

$$k_{OFF} = \frac{\left(1 - \frac{G}{M}\right)}{\frac{G}{M}} \frac{\alpha_{ON} k_{ON}}{(\alpha_{ON} - 1)} \frac{\alpha_{OFF} - 1}{\alpha_{OFF}} \rightarrow$$

$$k_{OFF} = \frac{(M - G)}{G} \frac{\alpha_{ON} k_{ON}}{(\alpha_{ON} - 1)} \frac{\alpha_{OFF} - 1}{\alpha_{OFF}} \text{time units}$$

Βάσει της παραπάνω ανάλυσης μπορεί κανείς να γεννήσει κίνηση κατά Pareto και μέσω της variance – time plot (ή άλλες τεχνικές plotting που αναφέρονται στη βιβλιογραφία) [17] να διαπιστώσει αν η παραγόμενη κίνηση έχει self - similar χαρακτηριστικά.

Συμπεράσματα

Στη διπλωματική αυτή εργασία μελετήσαμε τα χαρακτηριστικά των WDM δικτύων αναπτύσσοντας και μελετώντας ένα σύγχρονο και ένα ασύγχρονο πρωτόκολλο μετάδοσης. Εστίασαμε κυρίως στις κρίσιμες παραμέτρους των δικτύων, το throughput και την καθυστέρηση και πως αυτές επηρεάζονται από την υψηλή καθυστέρηση διάδοσης και το φαινόμενο των συγκρούσεων στους δέκτες.

Στο σύγχρονο πρωτόκολλο, το σύστημα πολλαπλών καναλιών ελέγχου μειώνει τις ανάγκες επεξεργασίας της πληροφορίας ελέγχου, εξαιτίας των σταθερά συντονισμένων φίλτρων – δεκτών των σταθμών. Ο αλγόριθμος που αναπτύχθηκε δίνει βελτιωμένο throughput εξαιτίας της αποφυγής των συγκρούσεων στα κανάλια δεδομένων. Επίσης εκτιμάται η επίδραση του φαινομένου των συγκρούσεων στους δέκτες στο throughput και την καθυστέρηση. Εξετάστηκε η επίδραση του propagation delay στην απόδοση του δικτύου και πώς η επίδραση αυτή μεταβάλλεται με τη μεταβολή στο μέγεθος της καθυστέρησης διάδοσης.

Στο ασύγχρονο πρωτόκολλο ο αλγόριθμος που αναπτύχθηκε δίνει τη δυνατότητα στους σταθμούς να μεταδώσουν τα δεδομένα τους στο κανάλι επιλογής τους γνωρίζοντας a priori την κατάσταση των καναλιών δεδομένων αλλά και του δέκτη – προορισμού. Η δυνατότητα αυτή παρέχεται εκμεταλλευόμενοι οι σταθμοί την υψηλή καθυστέρηση διάδοσης οδηγώντας σε βελτιωμένο throughput για το σύστημα που μελετάται. Εξετάζεται επίσης η επίδραση στην καθυστέρηση του propagation delay και του φαινομένου των receiver collisions.

Παραρτήματα

Κώδικας κεφαλαίου 3

```
/******
```

```
D. A. Pavlidi
```

```
Synchronous Multi Control Multi Data System  
Without Receiver Collisions Analysis
```

```
*****/
```

```
package practice;
```

```
import static java.lang.Math.exp;  
import static java.lang.Math.pow;
```

```
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.math.BigDecimal;
```

```
public class SPMCMDWithoutRC {
```

```
    public static void main(String[] args) {
```

```
        try{
```

```
            Synchronous1 syn1=new Synchronous1();  
            syn1.estimate1();
```

```
        }catch(IOException e){
```

```
            System.out.println("Found an exception:"+e.getMessage());
```

```
        }
```

```
    }
```

```
}
```

```
class Synchronous1{
```

```
    public Synchronous1(){  
    }
```

```
    BigDecimal S;//throughput
```

```
    BigDecimal g;//load
```

```
    BigDecimal D;//delay
```

```
    double N=10;//data channels
```

```
    double V=30;//control channels
```

```
    double L=50;//normalised transmission time
```

```
    double R=10;
```

```
    double S1;
```

```

double Sc;
double Sd;
double D1=0;
double g1;
double Pc;//control success probability
double Pd;//data success probability

public void estimate1() throws IOException{
    FileOutputStream
res= new FileOutputStream("SPMCMDWithoutRC_N10V30_R10.txt");
    res.write(("N=10 V=30\nR=10 L=50\n" + g + "S\tD\t\n").toString().getBytes());
    double cycle=1+R*L+L;
    g1=0.001;

    while(g1<=50){

        Sc=g1*exp(-g1*pow(V,-1));
        Sd=L*N*(1-pow(1-pow(N,-1),Sc))*pow(N*cycle,-1);
        D1=cycle*pow(2,-1)+(g1*pow(N*(1-pow(1-pow(N,-1), Sc)),-1)-
1)*(1+R*L)+cycle+R*L;

        g=new BigDecimal(g1);
        g = g.setScale(3,BigDecimal.ROUND_HALF_UP);
        S=new BigDecimal(Sd);
        S = S.setScale(6,BigDecimal.ROUND_HALF_UP);
        D=new BigDecimal(D1);
        D = D.setScale(6,BigDecimal.ROUND_HALF_UP);

        res.write((g+"\t"+S+"\t"+D+"\n").toString().getBytes());

        g1=g1+0.5;
    }
    res.close();
}
}

```

/******

D. A. Pavlidi

Synchronous Multi Control Multi Data System
Without Receiver Collisions Simulation

*****/

package practice;

import static java.lang.Math.*;


```

import java.io.FileOutputStream;
import java.io.IOException;
import java.math.*;

public class SimulationSPMCMDWithoutRC {

    public static void main(String[] args) {
        try{
            SimulateSPMCMD1 f1=new SimulateSPMCMD1();
            f1.simulate1();
        }catch(IOException e){
            System.out.println("Found an exception:"+e.getMessage());
        }
    }
}

class SimulateSPMCMD1{

    public SimulateSPMCMD1(){

        BigDecimal S;//throughput
        BigDecimal G;//load
        BigDecimal D;//delay
        double simulationTime=5*pow(10,6);//ο χρόνος εξομοίωσης

        int N=10;//data channels
        int V=40;//control channels
        int L=50;//normalised transmission time
        int R=10;
        int ControlTable[]=new int [V];
        int DataTable[]=new int [N];
        double S1;
        double D1;
        double g1;

        public void simulate1() throws IOException{

            FileOutputStream
            res= new FileOutputStream("Sim_SPMCMDWithoutRC_N10V40_R10.txt");
            res.write(("V=40 N=10 \n R=10 L=50 \nG\tSexp\tD\n").toString().getBytes());

            VariablePoisson1 var=new VariablePoisson1();

            g1=0.001;
            while(g1<50){
                int n=0;
                int vchoose=0;//controlchannel choise

```

```

int dchoose=0;//datachannel choice
double succeed=0;
double delay=0;
S1=0;
D1=0;
double cycle=0;

for(int t=0; t<=simulationTime;t=t+(1+R*L+L)){
    cycle++;
    n=var.choose(g1);//how many send to the control channels
    for(int i=0; i<V; i++){
        ControlTable[i]=0;
    }
    for(int i=0; i<N; i++){
        DataTable[i]=0;
    }

    //CONTROL CHANNELS
    if (n!=0){
        for(int i=1; i<=n;i++){
            vchoose=(int)(random()*V);//choose a specific control
channel
            ControlTable[vchoose]=ControlTable[vchoose]+1;//register the choice of each
one
        }
        for(int i=0; i<V;i++){
            if (ControlTable[i]==1){//if one has succeeded in a
control channel
                dchoose=(int)(random()*N);//chooses for a data
channel
                DataTable[dchoose]=DataTable[dchoose]+1;//registe the data channel choice
            }
            else if (ControlTable[i]>1){
                delay=delay+ControlTable[i];
            }
        }
        //DATA CHANNELS
        for(int i=0; i<N; i++){
            if (DataTable[i]>=1) succeed++;
            if (DataTable[i]>1)delay=delay+DataTable[i]-1;
        }
    }
}

S1=succeed*L*pow((((double)N*cycle*(1+R*L+L)), -1);

```

```

D1=delay*(1+R*L)*pow(succeed,-
1)+(1+2*R*L+L)+(1+R*L+L)*pow(2,-1);

```

```

G=new BigDecimal(g1);
S= new BigDecimal(S1);
D= new BigDecimal(D1);
S=S.setScale(6,BigDecimal.ROUND_HALF_UP);
D=D.setScale(3,BigDecimal.ROUND_HALF_UP);
G=G.setScale(3, BigDecimal.ROUND_HALF_UP);
res.write((G+"\t"+S+"\t"+D+"\n").toString().getBytes());

```

```

g1=g1+0.5;

```

```

}

```

```

res.close();

```

```

}

```

```

}

```

```

class VariablePoisson1 extends SimulateSPMCMD1{

```

```

    public VariablePoisson1(){
    }

```

```

// ypologismos paragontikou

```

```

    public double paragontiko(int n){
        double n_par=1;
        if (n==0 ){
            n_par=1;
        }
        else{
            for(int i=1; i<=n; i++){
                n_par = n_par*i;
            }
        }
        return n_par;
    }

```

```

}

```

```

    public int choose(double g2){
        double p=0;
        double u=random();
        int send=0;
        while(p<u){
            p=p+exp(-g2)*pow(g2,send)*pow(paragontiko(send),-1);
            if (p<u) send++;
        }
        return send;
    }

```

```
}  
}
```

```
/******  
D. A. Pavlidi
```

```
Synchronous Multi Control Multi Data System  
Without Receiver Collisions Analysis  
Total Throughput Estimation  
*****
```

```
package practice;  
import static java.lang.Math.exp;  
import static java.lang.Math.pow;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.math.BigDecimal;  
  
public class SPMCMDWithoutRC_Total_Throughput {  
    public static void main(String[] args) {  
        try{  
            Synchronous5 syn5=new Synchronous5();  
            syn5.estimate5();  
        }catch(IOException e){  
            System.out.println("Found an  
exception:"+e.getMessage());  
        }  
    }  
  
    class Synchronous5{  
        public Synchronous5(){  
            BigDecimal Stotal;//throughput  
            BigDecimal g;//load  
            BigDecimal D;//delay  
            double N=40;//data channels  
            double V=30;//control channels  
            double L=50;//normalised transmission time  
            double R=10;  
            double S1;  
            double Sc;  
            double Sd;  
            double D1=0;  
            double g1;  
            int M=50;  
            double Pc;//control success probability  
            double Pd;//data success probability
```

```

        public void estimate5() throws IOException{
            FileOutputStream res= new
FileOutputStream("SPMCMDWithoutRC_TotalS_N40V30_1.txt");
            res.write("N=40          V=30\nR=10
L=50\nng\tS\tD\t\n").toString().getBytes());
            double cycle=1+R*L+L;
            g1=0.001;
            while(g1<50){

                Sc=g1*exp(-g1*pow(V,-1));
                Sd=L*N*(1-pow(1-pow(N,-1),Sc))*pow(cycle,-1);

                g=new BigDecimal(g1);
                g = g.setScale(3,BigDecimal.ROUND_HALF_UP);
                Stotal=new BigDecimal(Sd);
                Stotal = Stotal.setScale(6,BigDecimal.ROUND_HALF_UP);

                res.write((g+"\t"+Stotal+"\t\n").toString().getBytes());

                g1=g1+0.5;
            }
            res.close();
        }
    }
}

```

/******

D. A. Pavlidi

Synchronous Multi Control Multi Data System With Receiver Collisions Analysis

*****/

```

package practice;

import static java.lang.Math.*;

import java.io.*;
import java.math.BigDecimal;

public class SPMCMD {

    public static void main(String[] args) {

        try{
            Synchronous syn=new Synchronous();
            syn.estimate();
        }
    }
}

```

```

        }catch(IOException e){
            System.out.println("Found an exception:"+e.getMessage());
        }
    }
}

class Synchronous{

    public Synchronous(){

        BigDecimal S;//throughput
        BigDecimal g;//load
        BigDecimal D;//delay
        int N=10;//data channels
        int V=40;//control channels
        int L=50;//normalised transmission time
        int R=10;
        double S1;
        double Sc;
        double Sd;
        double D1;
        double g1;
        int M=50;
        double Pc;//control success probability
        double Pd;//data success probability

        public void estimate() throws IOException{
            FileOutputStream
            res= new FileOutputStream("SPMCMD_RC_N10V40_R10.txt");
            res.write(("N=10 V=40 R=10\n").getBytes());

            g1=0.001;

            while(g1<50){

                Sc=g1*exp(-g1*pow(V,-1));
                Sd=N*(1-pow((1-pow(N,-1)),Sc));
                S1=L*M*(1-pow((1-pow(M,-1)),Sd))*pow((N*(1+R*L+L)), -1);
                D1=(L+R*L+1)*pow(2,-1)+(g1*pow(M*(1-pow((1-pow(M,-1)),Sd)), -
1)-1)*(R*L+1)+L+2*R*L+1;
                g=new BigDecimal(g1);
                g = g.setScale(3,BigDecimal.ROUND_HALF_UP);
                S=new BigDecimal(S1);
                D=new BigDecimal(D1);
                S = S.setScale(6,BigDecimal.ROUND_HALF_UP);
                D = D.setScale(6,BigDecimal.ROUND_HALF_UP);
                res.write((g+"t"+S+"t"+D+"\n").getBytes());
            }
        }
    }
}

```

```

        g1=g1+0.5;
    }
    res.close();
}
}

```

```

/*****

```

D. A. Pavlidi

Synchronous Multi Control Multi Data System With Receiver Collisions Simulation

```

*****/

```

```

package practice;

import static java.lang.Math.*;

import java.io.*;
import java.math.BigDecimal;

public class SimulationSPMCMD {

    public static void main(String[] args) {
        try{
            SimulateSPMCMD f=new SimulateSPMCMD();
            f.simulate();
        }catch(IOException e){
            System.out.println("Found an exception:"+e.getMessage());
        }
    }
}

class SimulateSPMCMD{

    public SimulateSPMCMD(){

        BigDecimal S;//throughput
        BigDecimal G;//load
        BigDecimal D;//delay
        double simulationTime=5*pow(10,6);//ο χρόνος εξομίωσης

        int N=10;//data channels
        int V=40;//control channels
        int L=50;//normalised transmission time
        int R=10;
        //int K=5;
    }
}

```

```

int M=50;
int ControlTable[]=new int [V];
int DataTable[]=new int [N];
int DestinTable[]=new int[M];
double S1;
double D1;
double g1;

public void simulate() throws IOException{

    FileOutputStream
    res= new FileOutputStream("Sim_SPMCMD_RC_N10V40_R10.txt");
    res.write(("N=40          V=30          R=10          L=50
M=50\nG\tSrc\tD\n").toString().getBytes());

    VariablePoisson var=new VariablePoisson();

    g1=0.001;
    while(g1<50){
        int n=0;
        int vchoose=0;//controlchannel choice
        int dchoose=0;//datachannel choice
        int Mchoose=0;
        double succeed=0;
        double delay=0;
        //double smalldelay=0;
        S1=0;
        double cycle=0;

        for(int t=0; t<=simulationTime;t=t+(1+R*L+L)){
            cycle++;
            n=var.choose(g1);//how many send to the control channels
            //System.out.println(n);
            for(int i=0; i<V; i++){
                ControlTable[i]=0;
            }
            for(int i=0; i<N; i++){
                DataTable[i]=0;
            }
            for(int i=0; i<M; i++){
                DestinTable[i]=0;
            }
            //CONTROL CHANNELS
            if (n!=0){
                for(int i=1; i<=n;i++){
                    vchoose=0;
                    vchoose=(int)(random()*V);//choose a specific control
channel

```



```

ControlTable[vchoose]=ControlTable[vchoose]+1;//register the choice of each
one
    }
    for(int i=0; i<V;i++){
        if (ControlTable[i]==1){//if one has succeeded in a
control channel
            dchoose=0;
            dchoose=(int)(random()*N);//chooses for a data
channel

            DataTable[dchoose]=DataTable[dchoose]+1;//register the data channel choice
            }
            else if (ControlTable[i]>1){
                delay=delay+ControlTable[i];
            }
        }
        //DATA CHANNELS
        for(int i=0; i<N; i++){
            if (DataTable[i]>=1){
                Mchoose=0;
                Mchoose=(int)(random()*M);

                DestinTable[Mchoose]=DestinTable[Mchoose]+1;
                if (DataTable[i]>1)delay=delay+DataTable[i]-1;
            }
        }

        for(int i=0;i<M;i++){
            if(DestinTable[i]>=1) {

                if(DestinTable[i]>1)delay=delay+DestinTable[i]-1;
                ++succeed;
            }
        }
    }
}

S1=succeed*L*pow((N*(cycle*(1+R*L+L))),-1);
D1=(delay*(double)(1+R*L))*pow(succeed,-
1)+(double)(1+R*L+L)*pow(2,-1)+(double)(1+2*R*L+L);
G=new BigDecimal(g1);
S=new BigDecimal(S1);
S=S.setScale(6,BigDecimal.ROUND_HALF_UP);
D=new BigDecimal(D1);
D=D.setScale(6,BigDecimal.ROUND_HALF_UP);
G=G.setScale(3, BigDecimal.ROUND_HALF_UP);
res.write((G+"\t"+S+"\t"+D+"\n").toString().getBytes());

g1=g1+0.5;

```

```

    }

    res.close();

}
}

class VariablePoisson extends SimulateSPMCMD{

    public VariablePoisson(){
    }

// ypologismos paragontikou
    public double paragontiko(int n){
        double n_par=1;
        if (n==0 ){
            n_par=1;
        }
        else{
            for(int i=1; i<=n; i++){
                n_par = n_par*i;
            }
        }
        return n_par;
    }

    public int choose(double g2){
        double p=0;
        double u=random();
        //System.out.println("h u einai:"+u+"\n");
        int send=0;
        while(p<u){
            p=p+exp(-g2)*pow(g2,send)*pow(paragontiko(send),-1);
            if (p<u) send++;
            if (send>M){
                send--;
                return send;
            }
        }
        return send;
    }
}
}

```

```
/******
```

D. A. Pavlidi

Synchronous Multi Control Multi Data System

Rejection Probability

```
*****/
```

```
package practice;

import static java.lang.Math.*;

import java.io.FileOutputStream;
import java.io.IOException;
import java.math.BigDecimal;

public class Synchronous_RejectionProbabilty {
    public static void main(String[] args) {
        try{
            Reject_sych rejsych = new Reject_sych();
            rejsych.calc_rejprob_sych();
        }catch(IOException e){
            System.out.println("FountanException:"+e.getMessage());
        }
    }
}

class Reject_sych{
    BigDecimal Sd;
    BigDecimal G;
    BigDecimal Src;
    BigDecimal Prej;
    double Sd1;
    double Sdhelp;
    double Prej1;
    double Src1;
    double Sc;
    double g;
    double L=50;
    double V=40;
    double N=10;
    double M=100;
    double R=10;

    public void calc_rejprob_sych()throws IOException{
        FileOutputStream res= new
        FileOutputStream("SPMCMD_RejProb_N10V40M100.txt");
```

```

g=0.001;
double cycle=1+R*L+L;
res.write("N=10          V=40          M=100\nL=50
R=10\nG\tPrej\tSd\tSrc\n").getBytes());
while(g<=45){

    Sc=g*exp(-g*pow(V,-1));
    Sd1=N*L*(1-pow(1-pow(N,-1),Sc))*pow(cycle,-1);
    Sd=new BigDecimal(Sd1);
    Sd=Sd.setScale(6, BigDecimal.ROUND_HALF_UP);

    //Sdhelp=N*(1-pow((1-pow(N,-1)),Sc));
    Src1=L*M*(1-pow((1-pow(M,-1)),N*(1-pow(1-pow(N,-
1),Sc))))*pow((cycle),-1);
    Src=new BigDecimal(Src1);
    Src=Src.setScale(6, BigDecimal.ROUND_HALF_UP);
    Prej1=(Sd1-Src1)*pow(Sd1,-1);
    if (Prej1<0) Prej1=0;
    Prej=new BigDecimal(Prej1);
    Prej=Prej.setScale(6, BigDecimal.ROUND_HALF_UP);
    G=new BigDecimal(g);
    G=G.setScale(3, BigDecimal.ROUND_HALF_UP);
    res.write((G+"\t"+Prej+"\t"+
                Sd+"\t"+Src+"\n").getBytes());
    g=g+0.5;
}
}
}

```

Κώδικας κεφαλαίου 4

```

/*****

```

D. A. Pavlidi

Asynchronous Single Control Multi Data System
Without Receiver Collisions Analysis

```

*****/

```

```

package practice;
import static java.lang.Math.*;
import java.io.*;
import java.math.BigDecimal;

```

```

public class TheoreticalPoissonSingleControlMultyData {

    public static void main(String[] args) {
        try{
            Analysis an = new Analysis();
            an.calc_throughput();
        }catch(IOException e){
            System.out.println("Found an Exception:"+e.getMessage());
        }
    }
}

class Analysis{
    BigDecimal Sd;
    BigDecimal G;
    BigDecimal D;
    double S1;
    double D1;
    double g;
    double L=50;
    double R=10;
    double N=20;
    int K=100;

    public void calc_throughput()throws IOException{
        FileOutputStream
res= new FileOutputStream("APSCMD_analysis_N20.txt");
        g=0.001;
        res.write(("R=10                                K=100
L=50\nG\t\t\tS\t\t\t\tD\n").toString().getBytes());
        while(g<=1){

            S1=L*g*exp(-g)*pow((1-(g*exp(-g)/N)),(L-1))*pow(N,-1);
            Sd=new BigDecimal(S1);
            Sd=Sd.setScale(6, BigDecimal.ROUND_HALF_UP);
            D1=pow(2,-1)+(exp(g)*pow((1-g*exp(-g)*pow(N,-1)),1-L)-
1)*(R*L+1+(1+K)/2)+(2*R+1)*L+1;
            D=new BigDecimal(D1);
            D=D.setScale(3, BigDecimal.ROUND_HALF_UP);
            G=new BigDecimal(g);
            G=G.setScale(3, BigDecimal.ROUND_HALF_UP);
            res.write((G+"\t"+Sd+"\t"+D+"\n").toString().getBytes());
            g=g+0.01;
        }
    }
}

```

```
/******
```

D. A. Pavlidi

Asynchronous Single Control Multi Data System Without Receiver Collisions Simulation

```
*****/
```

```
package practice;

import static java.lang.Math.*;
import java.io.*;
import java.math.*;

public class PoissonSingleControlMultyDataAlohaAloha {
    public static void main(String[] args) {
        try{
            S2_Aloha r1 = new S2_Aloha();
            r1.sim2();
        }catch(IOException e){
            System.out.println("Found an exception:"+e.getMessage());
        }
    }
}

class S2_Aloha{
    BigDecimal G;
    BigDecimal Sd;
    BigDecimal D;
    double g;//load
    int n=0;//packets to be transmitted
    double simulation_time=pow(10,6);
    int size=(int)simulation_time;
    int R=10;
    int N=20;//data channels
    int L=50;//packet length
    double K=100;
    int ControlTable[]=new int[L];//vulnerable period check
    double succeed;//succesfully transmitted
    double S;//Throughput
    double D1;
    double delay;
```

```

public S2_Aloha(){
}
public void sim2() throws IOException{

    FileOutputStream
    res= new FileOutputStream("APSCMD_simulation_N20.txt");
    res.write(("K=100
L=50\nG\t\t\tSd\t\t\t\tD\n").toString().getBytes());
    R=10

    Variable2 var=new Variable2();
    int flag;
    int c;

    g = 0.001;
    while(g<=1){
        succeed=0;
        S=0;
        D1=0;
        delay=0;

        for(int i=0;i<L;i++){
            ControlTable[i]=0;
        }
        for(int t=0;t<size;t++){
            c=0;
            flag=0;

            n=var.choose(g);

            for(int j=0;j<L-1;j++){//right shift
                ControlTable[j]=ControlTable[j+1];
            }

            ControlTable[L-1]=0;

            if (n==1){
                c=(int)ceil(random()*N);
                ControlTable[L-1]=c;
                for(int j=0;j<L-1;j++){
                    if (ControlTable[j]==c)
flag=flag+1;//data channel occupied
                }
                if (flag==0) ++succeed;
            }
        }
    }
}

```

```

        else delay++;
    }
    else delay=delay+n;
}

D1=pow(2,-1)+(delay)*((1+K)*pow(2,-
1)+1+R*L)*pow(succeed,-1)+(1+(2*R+1)*L);
S=succeed*L/(N*(simulation_time));
D=new BigDecimal(D1);
G=new BigDecimal(g);
Sd=new BigDecimal(S);
D=D.setScale(3, BigDecimal.ROUND_HALF_UP);
G=G.setScale(3, BigDecimal.ROUND_HALF_UP);
Sd=Sd.setScale(6, BigDecimal.ROUND_HALF_UP);
res.write((G+"\t"+Sd+"\t"+D+"\n").getBytes());
g=g+0.01;
    }
}
}

```

```
class Variable2 extends S2_Aloha{
```

```

    public Variable2(){
    }

```

```
// factorial
```

```

    public double paragontiko(int n){
        double n_par=1;
        if (n==0 ){
            n_par=1;
        }
        else{
            for(int i=1; i<=n; i++){
                n_par = n_par*i;
            }
        }
        return n_par;
    }

```

```

}
    public int choose(double g1){
        double p=0;
        double u=random();

        int send=0;
        while(p<u){
            p=p+exp(-g1)*pow(g1,send)/paragontiko(send);
        }
    }
}

```



```

        if (p<u) send++;
    }
    return send;
}
}

```

```

/*****

```

D. A. Pavlidi

Asynchronous Single Control Multi Data System With Receiver Collisions Analysis

```

*****/

```

```

package practice;

```

```

import static java.lang.Math.*;
import java.io.*;
import java.math.BigDecimal;

```

```

public class Theoretical_APSCMD_RECO {

```

```

    public static void main(String[] args) {
        try{
            Simulate10 sim10 = new Simulate10();
            sim10.calc_throughput10();
        }catch(IOException e){
            System.out.println("FoutanException:"+e.getMessage());
        }
    }
}

```

```

}

```

```

class Simulate10{
    BigDecimal S;
    BigDecimal G;
    BigDecimal D;
    double S1=0;
    double D1=0;
    double Psuc=0;
    double g=0;

    double L=50;
    double R=10;
}

```

```

double N=20;
double M=50;
double K=100;

public void calc_throughput10()throws IOException{
    FileOutputStream
res= new FileOutputStream("APSCMD_RECO_analysis_N20.txt");
    g=0.001;
    res.write(("R=10                K=100                L=50
M=50\nG\tS\tD\n").toString().getBytes());
    while(g<=1){

        Psuc=exp(-g)*pow(1-g*pow(N,-1)*exp(-g),L-1)*pow(1-
g*exp(-g)*pow(1-g*pow(N,-1)*exp(-g),L-1)*pow(M,-1),L-1);
        //System.out.println(Psuc);
        S1=L*g*Psuc*pow(N,-1);
        S=new BigDecimal(S1);
        S=S.setScale(6, BigDecimal.ROUND_HALF_UP);
        D1=pow(2,-1)+(pow(Psuc,-1)-1)*(R*L+1+(1+K)*pow(2,-
1)))+(2*R+1)*L+1;
        D=new BigDecimal(D1);
        D=D.setScale(3, BigDecimal.ROUND_HALF_UP);
        G=new BigDecimal(g);
        G=G.setScale(4, BigDecimal.ROUND_HALF_UP);
        res.write((G+"\t"+S+"\t"+D+"\t"+"n").toString().getBytes());
        g=g+0.01;
    }
}
}
}

```

```

/*****

```

D. A. Pavlidi

Asynchronous Single Control Multi Data System
With Receiver Collisions Simulation

```

*****/

```

```

package practice;

import static java.lang.Math.*;

import java.io.*;
import java.math.BigDecimal;

```

```

public class MultiPoissonRecCollision {
    public static void main(String[] args) {
        try{
            Sim_RecCo w1 = new Sim_RecCo();
            w1.execution2();
        }catch(IOException e){
            System.out.println("Found an exception:"+e.getMessage());
        }
    }
}
class Sim_RecCo{

    BigDecimal Sd;
    BigDecimal Sth;
    BigDecimal g;
    BigDecimal Dth;
    BigDecimal D;
    double simulation_time=pow(10,6);
    int M=50;//Stations
    int N=5;//Data channels
    int L=50;//Normalized Packet Length
    int ControlDestination[]=new int[L];//Declare destination station
    int ControlTable[]=new int[L];//check vulnerable period
    int R=10;//propagation delay in data slots
    int K=100;
    double S1;
    double D1;
    double g1; //offered load

    double delay;

    public void simulate2()throws IOException{
        FileOutputStream
        res= new FileOutputStream("APSCMD_RECO_simulation_N5.txt");
        res.write(("M=50 L=50 R=10 K=100\n" +
            "G\tS\tD\n").toString().getBytes());

        Variables3 var = new Variables3();

        g1=0.001;
        while(g1<=1){
            double success=0;//succesful transmissions
            S1=0;

            delay=0;
            D1=0;

```

```

    for (int i=0;i<L;i++){//initialize tables
        ControlDestination[i]=0;
        ControlTable[i]=0;
    }

for(double t=0; t<simulation_time;t++){
    int c=0;
    int d=0;
    int flag=0;
    int flag1=0;
    int n =var.choose(g1); //packets to be transmitted

    for(int j=0;j<L-1;j++){
        ControlTable[j]=ControlTable[j+1];
        ControlDestination[j]=ControlDestination[j+1];
    }

    ControlTable[L-1]=0;
    ControlDestination[L-1]=0;

    if (n==1){
        c=(int)ceil(random()*N);

        ControlTable[L-1]=c;
        d=(int)ceil(random()*M);
        ControlDestination[L-1]=d;
        for(int j=0;j<L-1;j++){
            if (ControlTable[j]==c) flag=flag+1;//data channel
occupied
        }

        if (flag==0) {
            for(int j=0;j<L-1;j++){
                if (ControlDestination[j]==d)
flag1=flag1+1;//destination station occupied
            }
            if (flag1==0) {
                ++success;
            } else delay++;
        } else {
            delay++;
            ControlDestination[L-1]=0;
        }
    } else delay=delay+n;
}
}

```

```

        S1=pow((N*simulation_time),-1)*success*L;
        D1=pow(2,-1)+(delay*(1+R*L+(K+1)*pow(2,-1)))*pow(success,-
1)+(1+(2*R+1)*L);
        g=new BigDecimal(g1);
        g = g.setScale(3,BigDecimal.ROUND_HALF_UP);
        Sd=new BigDecimal(S1);
        D=new BigDecimal(D1);
        Sd = Sd.setScale(6,BigDecimal.ROUND_HALF_UP);
        D = D.setScale(3,BigDecimal.ROUND_HALF_UP);

        /*Through th=new Through();
        double thr=th.throughput(g1);
        double dtr=th.delay(g1);
        Dth=new BigDecimal(dtr);
        Dth = Dth.setScale(6,BigDecimal.ROUND_HALF_UP);
        Sth=new BigDecimal(thr);
        Sth = Sth.setScale(6,BigDecimal.ROUND_HALF_UP);*/
        res.write((g+"\t"+Sd+"\t"+D+"\n").toString().getBytes());

        g1=g1+0.01;

    }
    res.close();
}

public void execution2() throws IOException{
    simulate2();
}
}

```

```

class Variables3 extends Sim_RecCo{

    public Variables3(){
    }

    // factorial
    public double paragontiko(int n){
        double n_par=1;
        if (n==0 ){
            n_par=1;
        }
        else{
            for(int i=1; i<=n; i++){
                n_par = n_par*i;
            }
        }
    }
}

```

```

        }
        }
        return n_par;
    }

//POISSON
public int choose(double f1){
    double p=0;
    double u=random();
    int send=0;
    while((p<u)&&(send<=M)){
        p=p+exp(-f1)*pow(f1,send)/paragontiko(send);
        if (p<u) send++;
    }
    return send;
}
}

```

/******

D. A. Pavlidi

Asynchronous Single Control Multi Data System Rejection Probability

*****/

```

package practice;

import static java.lang.Math.E;
import static java.lang.Math.exp;
import static java.lang.Math.pow;

import java.io.FileOutputStream;
import java.io.IOException;
import java.math.BigDecimal;

public class RejectionPropability {

    public static void main(String[] args) {
        try{
            Reject rej = new Reject();
            rej.calc_rejprob();

```

```

        }catch(IOException e){
            System.out.println("FountainException:"+e.getMessage());
        }
    }
}

class Reject{
    BigDecimal Sd;
    BigDecimal G;
    BigDecimal Src;
    BigDecimal Prej;
    double Sd1;
    double Prej1;
    double Src1;
    double St;
    double g;
    double L=50;
    double N=5;
    double M=100;

    public void calc_rejprob()throws IOException{
        FileOutputStream res= new FileOutputStream("Prej_N5M100.txt");
        g=0.001;
        res.write(("N=5 M=100\nG\tPrej\tSd\tSrc\n").toString().getBytes());
        while(g<=1){

            Sd1=L*g*pow(E,-g)*pow((1-(g*exp(-g)/N)),(L-1))*pow(N,-
1);

            Sd=new BigDecimal(Sd1);
            Sd=Sd.setScale(6, BigDecimal.ROUND_HALF_UP);
            St=L*g*exp(-g)*pow((1-(g*exp(-g)*pow(N,-1))),L-1);
            Src1=St*pow(1-St*pow((M*L),-1),L-1)*pow(N,-1);
            Src=new BigDecimal(Src1);
            Src=Src.setScale(6, BigDecimal.ROUND_HALF_UP);
            Prej1=(Sd1-Src1)*pow(Sd1,-1);
            Prej=new BigDecimal(Prej1);
            Prej=Prej.setScale(6, BigDecimal.ROUND_HALF_UP);
            G=new BigDecimal(g);
            G=G.setScale(3, BigDecimal.ROUND_HALF_UP);
            res.write((G+"\t"+Prej+"\t"+
                Sd+"\t"+Src+"\n").toString().getBytes());
            g=g+0.01;
        }
    }
}

```


Παραπομπές – Βιβλιογραφία

- [1] I. E. Πουντουράκης, καθ. ΕΜΠ, Σημειώσεις μαθήματος «Συστήματα Μετάδοσης και Δίκτυα Οπτικών Ινών», 8ο εξάμηνο σπουδών της Σχολής Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο.
- [2] John Crisp, “Introduction to Fiber Optics”, Second Edition, Newnes, 2001.
- [3] S. V. Kartalopoulos, “Next Generation Intelligent Optical Networks”, Springer, 2008
- [4] Harry J. R. Dutton, “Understanding Optical Communications”, Prentice Hall PTR., 1998
- [5] I. E. Πουντουράκης, καθ. ΕΜΠ, Σημειώσεις μαθήματος «Εξομοίωση Συστημάτων Επικοινωνιών», 8ο εξάμηνο σπουδών της Σχολής Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο.
- [6] I.E. Pountourakis, Propagation delay latency and a data channel collision avoidance protocol, J. Inform. Sci. 157 (2003) 111–119.
- [7] I. E. Pountourakis, D. A. Pavlidi, “Unavoidable Constraints and Collision Avoidance Techniques in Performance Evaluation of Asynchronous Transmission WDMA Protocols”, 12th WSEAS CSCC Multiconference, Heraklion, Crete Island, Greece, July 2008
- [8] I.Pountourakis, P.Baziana, G.Panagiotopoulos: 'Propagation Delay and Receiver Collision Analysis in WDMA Protocols', in: Proc. of 5th International Symposium Communication Systems Networks and Digital Signal Processing (CSNDSP2006), pp. 120-124, July 19-21, 2006 Patra, Greece
- [9] I.E.Pountourakis, “A Multiwavelength Control Architecture for Electronic Processing Bottleneck Reduction in WDMA Lightwave Networks”, Computer Communications, vol.22, No 15, pp. 1468-1480, Sept. 1999.
- [10] I.E. Pountourakis, “Performance evaluation with receiver collisions analysis in very high-speed optical fiber local area networks using passive star topology”, IEEE J. Lightwave Technol. 16 (1998) 2303–2310.
- [11] Feller, W. “An introduction to probability theory and its applications”, Third Edition, 1968, Wiley&Sons
- [12] W. Willinger, “The Discovery of Self – Similar Traffic”, AT&T Labs – Research
- [13] W. Willinger, M. Taqqu, W. Leland, D. Wilson, “Self – Similarity in High – Speed Packet Traffic: Analysis and Modeling of Ethernet Traffic Measurements
- [14] V. Paxson, S. Floyd, “Wide – Area Traffic: The Failure of Poisson Modeling
- [15] Edward R. Dougherty, “Probability and Statistics for the Engineering, Computing and Physical Sciences”, Prentice Hall International Inc, 1990
- [16] Jelger, C. “Characterization of a Wavelength Division Multiplexing Multi-Ring Network”, Thesis submitted to the University of Wales in Candidature for the Degree of Master of Philosophy , September 2001
- [17] Oleg I. Sheluhin, Sergey M. Smolskiy, Andrey V. Osin, “Self – Similar processes in Telecommunications”, John Wiley & Sons Ltd, 2007