

# ΔΙΚΤΥΑ ΕΠΙΚΟΙΝΩΝΙΩΝ

## Εργαστηριακή Άσκηση 8

### Ασύρματα Τοπικά Δίκτυα IEEE 802.11

#### 1. Εισαγωγή

Σε αυτή την άσκηση θα μελετηθεί η συμπεριφορά ασυρμάτων τοπικών δικτύων που ακολουθούν το πρότυπο IEEE 802.11. Η αρχιτεκτονική, η στοίβα και η λειτουργία πρωτοκόλλων, η δομή των πλαισίων και οι υπηρεσίες του προτύπου αυτού περιγράφονται αναλυτικά στο βιβλίο «Δίκτυα Υπολογιστών» (A. Tanenbaum), στο Κεφάλαιο 4.

Αρχικά θα γραφεί σε γλώσσα tcl ένα script για τον NS2, το οποίο θα προσομοιώνει ένα ασύρματο τοπικό δίκτυο IEEE 802.11 με δύο κόμβους, όπου ο ένας στέλνει πακέτα στον άλλον. Από την προσομοίωση θα προκύψει αρχείο ίχνους (trace file), από την ανάλυση του οποίου θα προκύψουν κάποια μέτρα επίδοσης του δικτύου. Συγκεκριμένα θα μετρηθεί η μέση τιμή της καθυστέρησης ανά πακέτο, καθώς και η χρησιμοποίηση του διαύλου. Ακολούθως, θα γίνει προσομοίωση με περισσότερους κόμβους και θα ζητηθεί η μέτρηση αντιστοίχων μεγεθών.

Στην ενότητα 2 παρουσιάζεται το πρότυπο IEEE 802.11 εστιάζοντας στα σημεία που υπεισέρχονται στην παρούσα εργαστηριακή άσκηση. Στην ενότητα 3 παρουσιάζονται οι λεπτομέρειες του σεναρίου προσομοίωσης και στην ενότητα 4 εκτελείται σειρά προσομοιώσεων, όπου μελετάται η συμπεριφορά του IEEE 802.11.

#### 2. Πρότυπο IEEE 802.11

Το πρότυπο IEEE 802.11 υποστηρίζει λειτουργία με δύο τρόπους:

1. Με παρουσία σταθμού βάσης.
2. Με απουσία σταθμού βάσης.

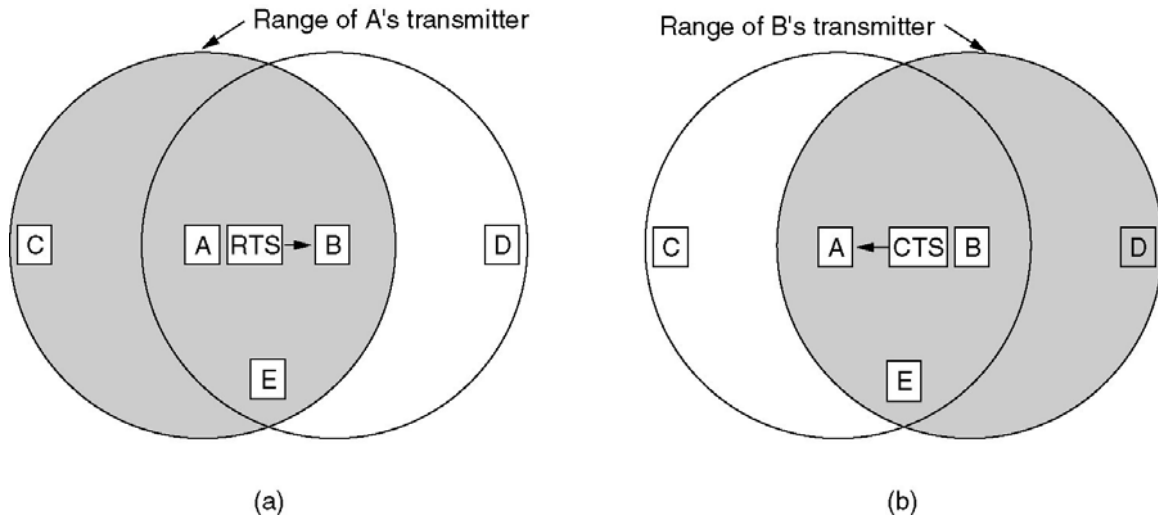
Στην πρώτη περίπτωση όλες οι επικοινωνίες πρέπει να περνούν από το σταθμό βάσης, ο οποίος ονομάζεται **σημείο πρόσβασης** (access point) στην ορολογία του 802.11. Στη δεύτερη περίπτωση οι υπολογιστές απλώς μεταδίδουν απευθείας ο ένας στον άλλον. Αυτός ο τρόπος λειτουργίας ονομάζεται συχνά **δίκτυωση ad hoc** (ad hoc networking).

Η εργαστηριακή αυτή άσκηση μελετά τη δεύτερη περίπτωση λειτουργίας του προτύπου IEEE 802.11 και γίνεται η υπόθεση ότι όλοι οι ραδιοπομποί έχουν την ίδια σταθερή εμβέλεια. Όταν ένας δέκτης βρίσκεται μέσα στην εμβέλεια δύο ενεργών πομπών, το σήμα που προκύπτει είναι γενικά παραμορφωμένο και άχρηστο – με άλλα λόγια, στην ανάλυση που ακολουθεί δεν θα ασχοληθούμε με συστήματα τύπου CDMA. Το γεγονός ότι εν γένει στα ασύρματα LAN, ο κάθε σταθμός δεν βρίσκεται απαραίτητως μέσα στην εμβέλεια όλων των άλλων σταθμών οδηγεί σε ποικιλία προβλημάτων. Επιπλέον, για τα ασύρματα LAN εντός κτιρίων, η παρουσία των τοίχων, ή αντίστοιχων εμποδίων ανάμεσα στους σταθμούς μπορεί να έχει μεγάλο αντίκτυπο στην πραγματική εμβέλεια του κάθε σταθμού.

Μια απλοϊκή προσέγγιση στη χρήση ενός ασύρματου LAN θα μπορούσε να είναι η προσπάθεια χρήσης του CSMA: να ακούμε για άλλες μεταδόσεις και να μεταδίδουμε μόνο αν κανείς άλλος δεν το κάνει. Το πρωτόκολλο αυτό όμως δεν είναι πραγματικά κατάλληλο για την περίπτωση, επειδή αυτό που έχει σημασία είναι οι παρεμβολές στον παραλήπτη, και όχι στον αποστολέα.

Ένα πρώιμο πρωτόκολλο που σχεδιάστηκε για ασύρματα LAN είναι το **πρωτόκολλο Πολλαπλής Πρόσβασης με Αποφυγή Συγκρούσεων** ή **MACA** (Multiple Access with Collision Avoidance, Karn 1990). Η βασική ιδέα πίσω από το πρωτόκολλο αυτό είναι να «διεγείρει» ο αποστολέας τον παραλήπτη

έτσι ώστε ο παραλήπτης να στείλει ένα μικρό πλαίσιο με αποτέλεσμα οι κοντινοί σε αυτόν σταθμοί να εντοπίσουν τη μετάδοση και να αποφύγουν τις μεταδόσεις κατά τη διάρκεια του επερχόμενου (μεγάλου) πλαισίου δεδομένων. Το πρωτόκολλο MACA απεικονίζεται στο Σχήμα 1.



Σχήμα 1 - Το πρωτόκολλο MACA. (a) Ο Α στέλνει ένα πλαίσιο RTS στον Β. (b) Ο Β ανταποκρίνεται στον Α με ένα πλαίσιο CTS.

Ας δούμε τώρα πώς ο Α στέλνει ένα πλαίσιο στον Β. Ο Α ξεκινά στέλνοντας ένα πλαίσιο **Αίτησης Αποστολής** ή **RTS** (Request To Send) στον Β, όπως φαίνεται στο Σχήμα 1 (a). Αυτό το μικρό πλαίσιο (30 byte) περιέχει το μέγεθος του πλαισίου δεδομένων που θα ακολουθήσει αργότερα. Ο Β ανταποκρίνεται στη συνέχεια με ένα πλαίσιο **Έγκρισης Αποστολής** ή **CTS** (Clear to Send), όπως φαίνεται στο Σχήμα 1 (b). Το πλαίσιο CTS περιέχει το μήκος των δεδομένων (αντιγράφεται από το πλαίσιο RTS). Με λήψη του πλαισίου CTS, ο Α αρχίζει τη μετάδοση.

Ας δούμε τώρα πώς αντιδρούν οι σταθμοί που ακούν κάποιο από αυτά τα πλαίσια. Ένας σταθμός που ακούει το RTS βρίσκεται προφανώς κοντά στον Α, και θα πρέπει να παραμείνει σιωπηλός για αρκετό χρόνο έτσι ώστε να επιστραφεί το πλαίσιο CTS στον Α χωρίς σύγκρουση. Κάθε σταθμός που ακούει το CTS βρίσκεται προφανώς κοντά στον Β και θα πρέπει να παραμείνει σιωπηλός κατά την επερχόμενη μετάδοση δεδομένων, το μήκος της οποίας μπορεί να προσδιοριστεί μέσω του πλαισίου CTS.

Στο Σχήμα 1, ο C βρίσκεται εντός της εμβέλειας του Α αλλά εκτός της εμβέλειας του Β. Κατά συνέπεια, ακούει το πλαίσιο RTS από τον Α αλλά όχι το πλαίσιο CTS από τον Β. Με την προϋπόθεση ότι δεν θα παρεμβληθεί με το CTS, είναι ελεύθερος να μεταδώσει ενώ στέλνεται το πλαίσιο δεδομένων. Αντιθέτως, ο D βρίσκεται εντός της εμβέλειας του Β αλλά όχι και του Α. Δεν ακούει το RTS αλλά ακούει το CTS. Η ακρόαση του CTS τον ενημερώνει ότι βρίσκεται κοντά σε ένα σταθμό που πρόκειται να λάβει ένα πλαίσιο, έτσι αποφεύγει να στείλει οτιδήποτε μέχρι τον αναμενόμενο χρόνο ολοκλήρωσης του πλαισίου αυτού. Ο σταθμός Ε ακούει και τα δύο μηνύματα ελέγχου και, όπως και D, θα πρέπει να παραμείνει σιωπηλός μέχρι να ολοκληρωθεί τα πλαίσια δεδομένων.

Παρά τις προφυλάξεις αυτές, μπορούν και πάλι να συμβούν συγκρούσεις. Για παράδειγμα, ο Β και ο C μπορούν να στείλουν ταυτόχρονα και οι δύο πλαίσια RTS στον Α. Τα πλαίσια αυτά θα συγκρουστούν και θα χαθούν. Σε περίπτωση σύγκρουσης, ο πομπός όπου συνέβη η αποτυχία (δηλαδή, ένας πομπός που δεν ακούει το πλαίσιο CTS μέσα στο αναμενόμενο χρονικό διάστημα) θα περιμένει ένα τυχαίο χρονικό διάστημα και θα ξαναδοκιμάσει αργότερα. Ο αλγόριθμος που χρησιμοποιείται είναι η δυαδική εκθετική οπισθοχώρηση, η οποία χρησιμοποιείται και στο Ethernet.

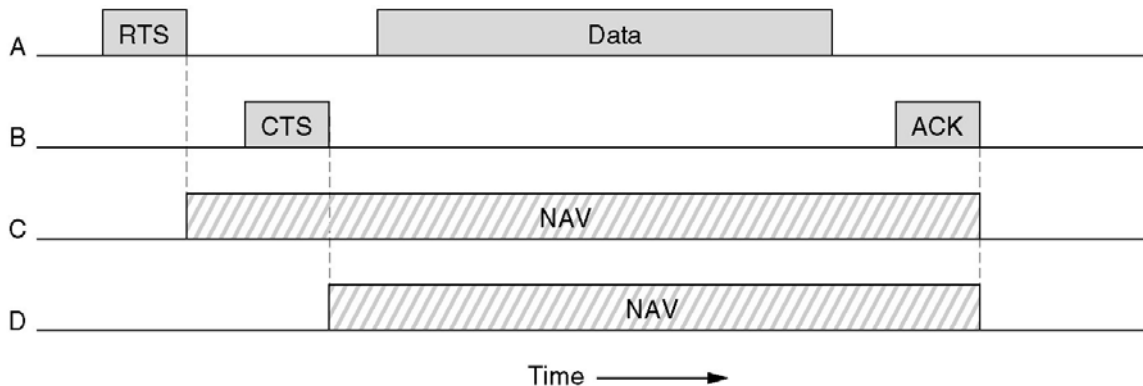
Όπως αναφέρθηκε στην αρχή, το 802.11 υποστηρίζει δύο καταστάσεις λειτουργίας. Η πρώτη δεν χρησιμοποιεί κάποιο είδος κεντρικού ελέγχου (οπότε από αυτή την άποψη, είναι παρόμοια με το Ethernet) και ονομάζεται **Κατανεμημένη Λειτουργία Συντονισμού** ή **DCF** (Distributed Coordination Function). Η άλλη χρησιμοποιεί το σταθμό βάσης για τον έλεγχο όλων των δραστηριοτήτων στην αντίστοιχη κυψέλη του και ονομάζεται **Σημειακή Λειτουργία Συντονισμού** ή **PCF** (Point Coordination Function). Όλες οι υλοποιήσεις πρέπει να υποστηρίζουν την DCF, ενώ η PCF είναι

προαιρετική. Θα εξετάσουμε μόνο την πρώτη από τις δύο καταστάσεις λειτουργίας παρακάτω, καθώς αυτή αποτελεί το αντικείμενο της παρούσας εργαστηριακής άσκησης.

Όταν χρησιμοποιείται η DCF, το 802.11 χρησιμοποιεί ένα πρωτόκολλο που ονομάζεται CSMA με Αποφυγή Συγκρούσεων ή CSMA/CA (CSMA with Collision Avoidance). Στο πρωτόκολλο αυτό χρησιμοποιείται ανίχνευση τόσο του φυσικού όσο και του εικονικού καναλιού. Το CSMA/CA υποστηρίζει δύο μεθόδους λειτουργίας. Στην πρώτη μέθοδο, όταν ένας σταθμός θέλει να μεταδώσει ανιχνεύει το κανάλι. Αν είναι αδρανές αρχίζει να μεταδίδει. Καθώς μεταδίδει δεν ανιχνεύει το κανάλι, αλλά στέλνει ολόκληρο το πλαίσιο του, το οποίο μπορεί βέβαια να καταστραφεί στον παραλήπτη λόγω παρεμβολών εκεί. Αν το κανάλι είναι απασχολημένο, ο αποστολέας αναβάλλει τη μετάδοση μέχρι το κανάλι να γίνει αδρανές, και τότε αρχίζει να μεταδίδει. Αν συμβεί μια σύγκρουση, οι σταθμοί που συγκρούστηκαν αναμένουν ένα τυχαίο χρονικό διάστημα χρησιμοποιώντας τον αλγόριθμο δυαδικής εκθετικής οπισθοχώρησης του Ethernet και ξαναδοκιμάζουν αργότερα.

Ο άλλος τρόπος λειτουργίας του CSMA/CA βασίζεται στον MACAW και χρησιμοποιεί ανίχνευση του εικονικού καναλιού, όπως φαίνεται στο Σχήμα 2. Στο παράδειγμα αυτό, ο A θέλει να στείλει στον B. Ο C εντός της εμβέλειας του A (και πιθανόν εντός της εμβέλειας του B, αυτό όμως δεν έχει σημασία). Ο D είναι ένας σταθμός εντός της εμβέλειας του B αλλά όχι εντός της εμβέλειας του A.

Το πρωτόκολλο αρχίζει όταν ο A αποφασίσει ότι θέλει να στείλει δεδομένα στον B. Ξεκινά στέλνοντας ένα πλαίσιο RTS στον B, ζητώντας άδεια να του στείλει ένα πλαίσιο. Όταν ο B λάβει την αίτηση αυτή, μπορεί να αποφασίσει να παραχωρήσει τη ζητούμενη άδεια, οπότε επιστρέφει ένα πλαίσιο CTS. Με τη λήψη του CTS, ο A στέλνει το πλαίσιο του και ξεκινά ένα χρονόμετρο επιβεβαίωσης. Αφού λάβει ορθά το πλαίσιο δεδομένων, ο B απαντά με ένα πλαίσιο επιβεβαίωσης, ολοκληρώνοντας την ανταλλαγή. Αν το χρονόμετρο επιβεβαίωσης του A λήξει πριν φτάσει σε αυτόν η επιβεβαίωση, ολόκληρο το πρωτόκολλο εκτελείται ξανά.



Σχήμα 2 - Ανίχνευση εικονικού καναλιού με το CSMA/CA.

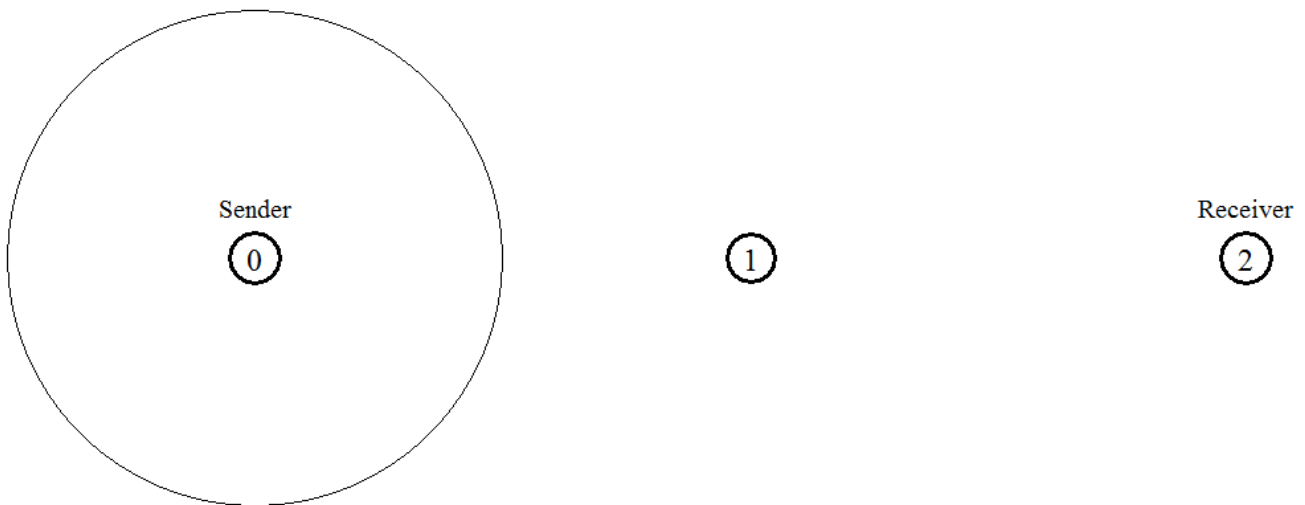
Ας εξετάσουμε τώρα την ανταλλαγή αυτή από την οπτική γωνία των C και D. Ο C βρίσκεται εντός της εμβέλειας του A, έτσι μπορεί να λάβει το πλαίσιο RTS. Αν το λάβει αντιλαμβάνεται ότι σε λίγο κάποιος θα στείλει δεδομένα, έτσι για το καλό όλων αποφεύγει να μεταδώσει οτιδήποτε μέχρι να ολοκληρωθεί η ανταλλαγή. Από τις πληροφορίες που παρέχονται στην αίτηση RTS μπορεί να εκτιμήσει πόσο χρόνο θα πάρει η ανταλλαγή, συμπεριλαμβανομένης της τελικής επιβεβαίωσης, έτσι ενεργοποιεί για τον εαυτό του ένα σήμα ότι το εικονικό κανάλι είναι απασχολημένο, γεγονός που σημειώνεται στο Σχήμα 2 ως Διάνυσμα Εκχώρησης Δικτύου ή NAV (Network Allocation Vector). Ο D δεν ακούει το μήνυμα RTS, ακούει όμως το CTS, έτσι ενεργοποιεί και αυτός από την πλευρά του το σήμα NAV. Σημειώστε ότι τα σήματα NAV δεν μεταδίδονται. Είναι απλώς εσωτερικές υπενθυμίσεις ότι ο σταθμός πρέπει να παραμείνει σιωπηλός για μια συγκεκριμένη χρονική περίοδο.

Όλα τα παραπάνω ισχύουν για την κατάσταση λειτουργίας DCF του 802.11. Σε αυτή την κατάσταση λειτουργίας δεν υπάρχει κεντρικός έλεγχος και οι σταθμοί ανταγωνίζονται για το κανάλι, ακριβώς όπως και στο Ethernet.

### 3. Σενάριο προσομοίωσης

#### 3.1 Παράμετροι Προσομοίωσης, Τοπολογία Δικτύου και Ορισμός Αρχείων Ιχνών

Στο πλαίσιο της προσομοίωσης θα υλοποιηθεί μια ασύρματη δικτυακή τοπολογία όπως φαίνεται για παράδειγμα στο Σχήμα 3.



Σχήμα 3 – Τοπολογία ασυρμάτου τοπικού δικτύου.

Στην τοπολογία αυτή, ορίζουμε επίπεδο πλέγμα διαστάσεων  $opt(gridx) \times opt(gridy)$ , όπου όλοι οι κόμβοι είναι τοποθετημένοι στο μέσο του άξονα  $y$  και σε απόσταση  $opt(distx)$  μεταξύ τους στον άξονα  $x$ , συμμετρικά ως προς το κέντρο του. Στην οριζόντια διάταξη που προκύπτει, ο πρώτος κόμβος από αριστερά παράγει  $opt(fiprh)$  πακέτα FTP με προορισμό τον τελευταίο κόμβο από τα δεξιά. Στο τμήμα του κώδικα που ακολουθεί, καθορίζονται οι παράμετροι της προσομοίωσης. Σημαντικότερες από αυτές τις παραμέτρους είναι το πλήθος των ασύρματων κόμβων  $opt(nn)$ , η οριζόντια απόσταση μεταξύ δύο διαδοχικών κόμβων  $opt(distx)$ , το μέγεθος των πακέτων TCP  $opt(tcpsize)$ , η χρονική στιγμή έναρξης της κίνησης FTP  $opt(fipstart)$ , η διάρκεια της προσομοίωσης  $opt(simstop)$ . Αυτές οι παράμετροι πρέπει να αλλάζουν σε κάθε προσομοίωση, όπως περιγράφεται στη συνέχεια.

```
# =====  
# Ορισμός παραμέτρων προσομοίωσης  
# =====  
set opt(chan)          Channel/WirelessChannel ;# Τύπος καναλιού  
set opt(prop)          Propagation/TwoRayGround ;# Μοντέλο ραδιομετάδοσης  
set opt(ant)           Antenna/OmniAntenna ;# Τύπος κεραίας  
set opt(ll)            LL ;# Τύπος επιπέδου σύνδεσης  
set opt(ifq)           Queue/DropTail/PriQueue ;# Τύπος ουράς  
set opt(ifqlen)        50 ;# Μέγιστος αριθμός πακέτων  
                        ;# στην ουρά  
set opt(netif)         Phy/WirelessPhy ;# Τύπος δικτυακής επαφής  
set opt(mac)           Mac/802_11 ;# Πρωτόκολλο MAC  
set opt(rp)            DSDV ;# Πρωτόκολλο δρομολόγησης  
set opt(nn)            2 ;# Αριθμός κόμβων  
set opt(gridx)         600 ;# Μήκος πλέγματος (m)
```

```

set opt(gridy)          100                ;# Πλάτος πλέγματος (m)
set opt(distx)         100                ;# Οριζόντια απόσταση μεταξύ
                                        ;# διαδοχικών κόμβων (e)m
set opt(tcpsize)       1440               ;# Μέγεθος τεμαχίου TCP
set opt(ftppn)         1000               ;# Αριθμός πακέτων FTP
set opt(ftpstart)      30.0               ;# Χρόνος εκκίνησης FTP
set opt(simstop)       50.0               ;# Χρόνος λήξης προσομοίωσης

```

```

$opt(mac) set basicRate_ 1Mb
$opt(mac) set dataRate_ 11Mb

```

```

# Δημιουργία αντικειμένου προσομοίωσης
set ns [new Simulator]

```

```

# Ορισμός αρχείων ιχνών
set tf [open lab8.tr w]
$ns trace-all $tf

```

```

set nf [open lab8.nam w]
$ns namtrace-all-wireless $nf $opt(gridx) $opt(gridy)

```

```

# Συνάρτηση τερματισμού
proc finish {} {
    global ns tf nf
    $ns flush-trace
    close $tf
    close $nf
    exit 0
}

```

```

# Δημιουργία αντικειμένου τοπογραφίας
set topo [new Topography]
# Επίπεδο πλέγμα ( $opt(gridx) x $opt(gridy) ) m^2
$topo load_flatgrid $opt(gridx) $opt(gridy)

```

```

# Δημιουργία αντικειμένου GOD (General Operations Director) που αποθηκεύει τον
# συνολικό αριθμό κόμβων και έναν πίνακα με τον ελάχιστο αριθμό βημάτων (hops)
# μεταξύ των κόμβων
create-god $opt(nn)

```

```

# Παραμετροποίηση κόμβων
$ns node-config -adhocRouting $opt(rp) \

```

```

-llType $opt(ll) \
-macType $opt(mac) \
-ifqType $opt(ifq) \
-ifqLen $opt(ifqlen) \
-antType $opt(ant) \
-propType $opt(prop) \
-phyType $opt(netif) \
-channel [new $opt(chan)] \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace OFF

#
# Οι κόμβοι τοποθετούνται στον άξονα x συμμετρικά ως προς το μέσο του,
# ενώ η απόσταση μεταξύ δύο διαδοχικών κόμβων είναι ίση με $opt(distx)
#

# Ορισμός θέσης πρώτου κόμβου στον άξονα x
set posx(0) [expr $opt(gridx)/2-(($opt(nn)-1)/2)*$opt(distx)]

# Δημιουργία κόμβων και ορισμός συτταγμένων τους στο επίπεδο πλέγμα
for {set i 0} {$i < $opt(nn)} {incr i} {
    # Δημιουργία κόμβου
    set n($i) [$ns node]
    # Απενεργοποίηση τυχαίας κίνησης κόμβου
    $n($i) random-motion 0
    # Ορισμός θέσης κόμβου στον άξονα x
    $n($i) set X_ [expr $posx(0) + $i*$opt(distx)]
    # Ορισμός θέσης κόμβου στον άξονα y
    $n($i) set Y_ [expr $opt(gridy)/2]
    # Ορισμός θέσης κόμβου στον άξονα z
    $n($i) set Z_ 0.0
}

```

## 2.2 Γεννήτρια Κίνησης

Για τη μετάδοση δεδομένων μεταξύ των ακραίων κόμβων χρησιμοποιούμε το πρωτόκολλο FTP. Η κίνηση αυτή ξεκινά τη χρονική στιγμή *opt(fipstart)*.

```

# Ορισμός μεγέθους τεμαχίου TCP (χωρίς την επικεφαλίδα TCP)
Agent/TCP set packetSize_ $opt(tcpsize)

```

```

set agent [new Agent/TCP]

# Ορισμός πρώτου κόμβου ως αποστολέα κίνησης FTP
set app [new Application/FTP]
$app attach-agent $agent
$ns attach-agent $n(0) $agent

# Ορισμός τελευταίου κόμβου ως παραλήπτη
set sink [new Agent/TCPSink]
$ns attach-agent $n([expr $opt(nn)-1]) $sink

# Σύνδεση αποστολέα - παραλήπτη
$ns connect $agent $sink

```

### 2.3 Εκτέλεση του σεναρίου

Τέλος, ορίζουμε τα γεγονότα της προσομοίωσης:

```

# Αρχικοποίηση κινητών κόμβων
for {set i 0} {$i < $opt(nn) } {incr i} {
    $ns initial_node_pos $n($i) 10
}

# Ορισμός ετικέτας πρώτου κόμβου
$ns at 0.0 "$n(0) label Sender"
# Ορισμός ετικέτας τελευταίου κόμβου
$ns at 0.0 "$n([expr $opt(nn)-1]) label Receiver"

# Καθυστέρηση έναρξης κίνησης FTP $opt(ftpstart) δευτερόλεπτων για να
# ολοκληρωθεί η δημιουργία των πινάκων δρομολόγησης
$ns at $opt(ftpstart) "$app produce $opt(ftppn)"
# Reset όλων των κόμβων
for {set i 0} {$i < $opt(nn)} {incr i} {
    $ns at $opt(simstop) "$n($i) reset";
}
# Τερματισμός
$ns at $opt(simstop) "finish"

$ns run

```

Με βάση τα παραπάνω δημιουργούμε το αρχείο **lab8.tcl**, ενώ για την εκτέλεση της προσομοίωσης χρησιμοποιούμε την εντολή **ns lab8.tcl**. Μετά την εκτέλεση αυτής της εντολής θα πρέπει να έχουν δημιουργηθεί τα αρχεία **lab8.tr** και **lab8.nam**. Με την εντολή **nam lab8.nam** μπορούμε να δούμε την

τοπολογία του δικτύου καθώς και την κίνηση που έχει δημιουργηθεί.

Το αρχείο **lab8.tr** περιέχει πληροφορίες για όλα τα γεγονότα που συνέβησαν κατά την προσομοίωση. Στην επόμενη ενότητα θα εξηγηθεί πώς αναλύονται αυτά τα δεδομένα.

### 3. Ανάλυση αρχείου ίχνους (trace file)

Αφού έχουμε δημιουργήσει το σενάριο προσομοίωσης, το έχουμε εκτελέσει και έχουμε δημιουργήσει τα αρχεία αποτελεσμάτων, τα αρχεία αυτά πρέπει να αναλυθούν ώστε να πάρουμε τις πληροφορίες που θέλουμε.

#### 3.1 Μορφή αρχείου ίχνους (trace file)

Το αρχείο **lab8.tr** που δημιουργήθηκε προηγουμένως περιέχει πληροφορίες της μορφής:

```
s 30.060199152 _1_ AGT --- 48 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [15 0] 0 0
r 30.060199152 _1_ RTR --- 48 ack 40 [0 0 0 0] ----- [1:0 0:0 32 0] [15 0] 0 0
s 30.060199152 _1_ RTR --- 48 ack 60 [0 0 0 0] ----- [1:0 0:0 32 0] [15 0] 0 0
r 30.061516152 _0_ AGT --- 43 ack 60 [13a 0 1 800] ----- [1:0 0:0 32 0] [12 0] 1 0
s 30.061516152 _0_ AGT --- 49 tcp 1480 [0 0 0 0] ----- [0:0 1:0 32 0] [25 0] 0 0
r 30.061516152 _0_ RTR --- 49 tcp 1480 [0 0 0 0] ----- [0:0 1:0 32 0] [25 0] 0 0
s 30.061516152 _0_ RTR --- 49 tcp 1500 [0 0 0 0] ----- [0:0 1:0 32 1] [25 0] 0 0
s 30.061516152 _0_ AGT --- 50 tcp 1480 [0 0 0 0] ----- [0:0 1:0 32 0] [26 0] 0 0
r 30.061516152 _0_ RTR --- 50 tcp 1480 [0 0 0 0] ----- [0:0 1:0 32 0] [26 0] 0 0
s 30.061516152 _0_ RTR --- 50 tcp 1500 [0 0 0 0] ----- [0:0 1:0 32 1] [26 0] 0 0
r 30.063053485 _0_ AGT --- 44 ack 60 [13a 0 1 800] ----- [1:0 0:0 32 0] [13 0] 1 0
```

Κάθε γραμμή του αρχείου αυτού αντιστοιχεί σε ένα γεγονός που συνέβη κατά τη διάρκεια της προσομοίωσης. Ο πρώτος χαρακτήρας κάθε γραμμής υποδηλώνει το είδος του γεγονότος που συνέβη. Ο χαρακτήρας “s” σημαίνει επιτυχημένη αποστολή πακέτου, ενώ ο χαρακτήρας “r” σημαίνει επιτυχημένη λήψη πακέτου, και ο χαρακτήρας “D” σημαίνει απόρριψη πακέτου.

Η δεύτερη λέξη της κάθε γραμμής είναι η χρονική στιγμή κατά την οποία συνέβη το γεγονός που καταγράφεται. Η τρίτη λέξη περιγράφει τον κόμβο στον οποίον αναφέρεται το γεγονός. Η τέταρτη λέξη δηλώνει το επίπεδο όπου λαμβάνει χώρα το γεγονός της γραμμής (π.χ. AGT → agent, RTR → router). Η έκτη λέξη αναπαριστά το μοναδικό αριθμό ακολουθίας που αποδίδει το NS2 στα πακέτα. Η έβδομη λέξη δηλώνει το είδος του πακέτου και η όγδοη λέξη περιγράφει το μέγεθος του πακέτου (συμπεριλαμβάνονται οι επικεφαλίδες TCP και IP). Τα υπόλοιπα πεδία δεν χρησιμοποιούνται στην ανάλυση των αποτελεσμάτων της προσομοίωσης που ακολουθεί.

#### 3.2 Ανάλυση με τη γλώσσα awk

Για να μπορούμε να υπολογίσουμε την χρησιμοποίηση (utilization) του καναλιού, όταν χρησιμοποιείται το πρωτόκολλο IEEE 802.11, πρέπει να μετρήσουμε την ποσότητα των δεδομένων που ελήφθησαν κατά τη διάρκεια της προσομοίωσης. Στη συνέχεια αυτός αριθμός θα διαιρεθεί με τη διάρκεια της μετάδοσης του συνόλου των πακέτων, ώστε να υπολογιστεί ο ρυθμός διέλευσης δεδομένων πάνω από το ασύρματο κανάλι (throughput), και αυτή η τιμή θα διαιρεθεί με τον ονομαστικό ρυθμό μετάδοσης του καναλιού (11 Mbps) ώστε να προκύψει η χρησιμοποίηση (utilization).

Για κάθε πακέτο που λαμβάνει ο αποδέκτης, η ποσότητα των δεδομένων αυξάνεται κατά το μέγεθος



του πακέτου. Συνεπώς πρέπει να εντοπίσουμε τις γραμμές του αρχείου ίχνους που φανερώνουν ορθή λήψη πακέτου *tcp* και επιπέδου *agent*. Αυτές οι γραμμές αρχίζουν με τον χαρακτήρα “Γ” και περιλαμβάνουν τις λέξεις *tcp* και *AGT*. Για κάθε τέτοια γραμμή που εντοπίζεται, η μεταβλητή *packets* (αριθμός πακέτων που ελήφθησαν) αυξάνεται κατά ένα, ενώ η μεταβλητή *data* (πλήθος δεδομένων που ελήφθησαν) αυξάνεται κατά το μέγεθος του πακέτου (το οποίο βρίσκεται στην όγδοη λέξη κάθε γραμμής του αρχείου ίχνους). Έτσι καταγράφεται η ποσότητα των δεδομένων που ελήφθησαν.

Η καθυστέρηση κάθε πακέτου προκύπτει αν αφαιρεθεί ο χρόνος λήψης κάθε πακέτου από τον χρόνο αποστολής του. Ως χρόνο αποστολής ενός πακέτου θεωρούμε τη χρονική στιγμή κατά την οποία το πακέτο εξέρχεται από την ουρά αναμονής στον κόμβο αποστολής. Με το ακόλουθο τμήμα κώδικα:

```
/^s/AGT/tcp/ {
  sendtimes[$6] = $2;
}
```

εντοπίζονται οι γραμμές που αναφέρονται σε αποστολή πακέτου *tcp* σε επίπεδο *agent*. Η χρονική στιγμή αποστολής (\$2) αποθηκεύεται στον πίνακα *sendtimes*. Ο χρόνος αποστολής αφαιρείται από τον χρόνο λήψης και προκύπτει η καθυστέρηση του κάθε πακέτου. Αθροίζοντας όλες τις καθυστερήσεις και διαιρώντας προς τον αριθμό των πακέτων προκύπτει η μέση καθυστέρηση.

Ο κώδικας *awk* για την εκτέλεση αυτής της ανάλυσης είναι ο εξής:

```
BEGIN {
  data = 0;
  packets = 0;
  sumDelay = 0;

  # Do we need to fix the decimal mark?
  if (sprintf(sqrt(2)) ~ /,/) dmfix = 1;
}

{
  # Apply dm fix if needed
  if (dmfix) sub(/\. /, ", ", $0);
}

/^s/AGT/tcp/ {
  sendtimes[$6] = $2;
}

/^r/AGT/tcp/ {
  data += $8;
  packets++;
  sumDelay += $2 - sendtimes[$6];
}
```

```

END {
    printf("Total Data received\t: %d Bytes\n", data);
    printf("Total Packets received\t: %d\n", packets);
    printf("Average Delay\t\t: %f sec\n", (1.0 * sumDelay)/packets);
}

```

### 3.3 Εκτέλεση του προγράμματος ανάλυσης

Για την εκτέλεση προγραμμάτων awk χρησιμοποιείται ο διερμηνέας (interpreter) awk με παραμέτρους το όνομα του αρχείου που περιγράφει τις διαδικασίες της ανάλυσης και το όνομα του αρχείου που περιλαμβάνει τα δεδομένα που θα αναλυθούν. Εάν ο παραπάνω κώδικας έχει αποθηκευτεί στο αρχείο **lab8.awk** και τα δεδομένα βρίσκονται στο αρχείο **lab8.tr**, τότε εκτελούμε την εντολή:

```
awk.exe -f lab8.awk < lab8.tr
```

Αυτή η εντολή θα εκτυπώσει στην οθόνη τον αριθμό των πακέτων και το πλήθος των δεδομένων που ελήφθησαν.

Για να αποθηκεύσουμε τα αποτελέσματα της ανάλυσης στο αρχείο **results.txt** εκτελούμε την εντολή

```
awk.exe -f lab8.awk < lab8.tr > results.txt
```

**Προσοχή:** Σε περιβάλλον MS Windows ο διερμηνέας awk χρησιμοποιεί ως σύμβολο υποδιαστολής (decimal mark), το σύμβολο που είναι καθορισμένο από τα *regional settings* της εγκατάστασης. Σε πολλές περιπτώσεις αυτό το σύμβολο είναι το κόμμα (,) και όχι η τελεία (.), με αποτέλεσμα το awk να μην μπορεί να χειριστεί σωστά πραγματικούς αριθμούς από το αρχείο εισόδου. Για να αντιμετωπιστεί το πρόβλημα αυτό, έχουμε προσθέσει τον ακόλουθο κανόνα ο οποίος αντικαθιστά όπου χρειάζεται τις τελείες με κόμματα.

```

{
    # Apply dm fix if needed
    if (dmfix) sub(/\./, ",", $0);
}

```

Εναλλακτικά, θα μπορούσαμε να ορίσουμε την τελεία ως το σύμβολο για την υποδιαστολή αλλάζοντας τις ρυθμίσεις των Windows (Control Panel -> Regional and Language Options).

## 4. Μελέτη συμπεριφοράς ασύρματου τοπικού δικτύου IEEE 802.11

### 4.1 Επίδραση απόστασης κόμβων

Εκτελέστε σειρά προσομοιώσεων με δύο κόμβους (θέτοντας την παράμετρο *opt(nn)* ίση με 2) μεταβάλλοντας την οριζόντια απόστασή τους (αλλάζοντας την τιμή της παραμέτρου *opt(distx)*). Πρέπει να πραγματοποιηθούν τουλάχιστον δέκα προσομοιώσεις.

(α) Για αυτές τις προσομοιώσεις και με τη βοήθεια του awk script να γίνει γραφική παράσταση της

μέσης τιμής της καθυστέρησης ανά πακέτο συναρτήσει της απόστασης των δύο κόμβων.

- (β) Ποια είναι η ελάχιστη τιμή της παραμέτρου  $opt(distx)$  για την οποία είναι αδύνατη η μεταφορά δεδομένων μεταξύ των δύο κόμβων; Ποια είναι η εμβέλεια του δικτύου για τους δύο κόμβους;
- (γ) Τέλος, να γίνει γραφική παράσταση της τιμής της χρησιμοποίησης του καναλιού (utilization) συναρτήσει της απόστασης των δύο κόμβων. Τι παρατηρείτε;

#### 4.2 Επίδραση αριθμού κόμβων

Στη συνέχεια θα γίνει σειρά προσομοιώσεων με μεταβλητό πλήθος κόμβων και θα μελετηθεί η επίδραση του πλήθους των κόμβων στη χρησιμοποίηση του καναλιού και στη μέση καθυστέρηση. Για το σκοπό αυτό θα γίνει σειρά προσομοιώσεων με σταθερή απόσταση διαδοχικών κόμβων ίση με 100 m και τοποθετώντας διαδοχικά 2, 4 και 6 κόμβους πάνω στον άξονα x.

- (α) Υπολογίστε με τη βοήθεια του awk script τη μέση τιμή της καθυστέρησης ανά πακέτο για κάθε μία από αυτές τις τρεις τοπολογίες. Πώς επηρεάζει την τιμή αυτή το πλήθος των κόμβων και γιατί; Πόσο χρόνο διαρκεί η πρώτη συμπλήρωση των πινάκων δρομολόγησης σε κάθε μία από τις τρεις παραπάνω περιπτώσεις;
- (β) Εκτελέστε άλλη μία προσομοίωση προσθέτοντας δύο ακόμη κόμβους και προσαρμόζοντας ανάλογα την παράμετρο  $opt(gridx)$  που καθορίζει το μήκος του πλέγματος. Τι παρατηρείτε; Πώς δικαιολογείται η συμπεριφορά του δικτύου; Τι τροποποίηση πρέπει να γίνει στην παραμετροποίηση της προσομοίωσης ώστε να ολοκληρωθεί επιτυχώς η μεταφορά των 1000 πακέτων; Πόση είναι τότε η μέση τιμή της καθυστέρησης ανά πακέτο;