

**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**



**ΔΙΚΤΥΑ ΕΠΙΚΟΙΝΩΝΙΩΝ**

(2018-2019)

*1<sup>η</sup> ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΝΑΦΟΡΑ*

Ονοματεπώνυμο : Χρήστος Τσούφης

A.M. : 0311716

## 1. Εισαγωγή στο Network Simulator v2 (NS2)

## 2. Το πρώτο Tcl script – Γνωριμία με το NAM

Στην άσκηση αυτή γράφθηκε βήμα προς βήμα ένα Tcl script για NS, το οποίο προσομοιώνει μια απλή τοπολογία δικτύου. Επίσης έγινε γνωστός ο τρόπος που εγκαθίστανται κόμβοι και ζεύξεις, πώς στέλνονται δεδομένα από τον ένα κόμβο στον άλλον, πώς να παρακολουθείται μια ουρά αναμονής και πώς να χρησιμοποιείται το NAM (Network Animator) για να αναπαρασταθεί γραφικά η προσομοίωση.

### ΕΡΩΤΗΣΕΙΣ

- Ποιος είναι ο ρυθμός μετάδοσης σε bit/sec;  
Σύμφωνα με τα δεδομένα και λαμβάνοντας υπόψη ότι 1 byte = 8 bits, ο ρυθμός μετάδοσης θα είναι:

$$u = \frac{1 \text{ πακέτο}}{0,005 \text{ sec}} = \frac{1000 \text{ bytes}}{0,005 \text{ sec}} = 200.000 \frac{\text{bytes}}{\text{sec}} = 200.000 * 8 \frac{\text{bits}}{\text{sec}} = 1.600.000 \frac{\text{bits}}{\text{sec}}$$

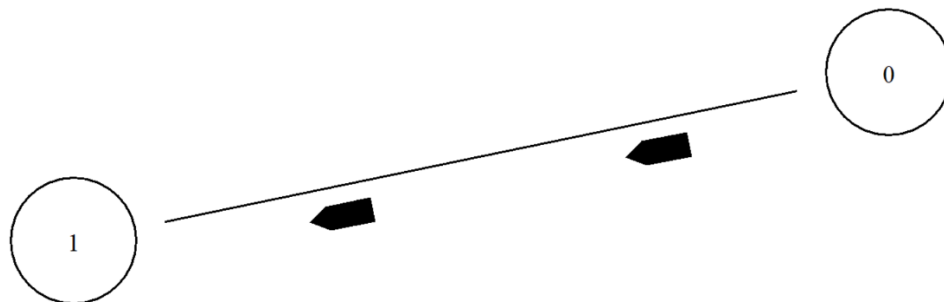
- Ποιος είναι ο συνολικός αριθμός bytes και bits που μεταφέρθηκαν από την αρχή ως το τέλος της προσομοίωσης;

Ο χρόνος μετάδοσης ισούται με 8,5 sec – 1,5 sec = 7,0 sec , άρα συνολικά:

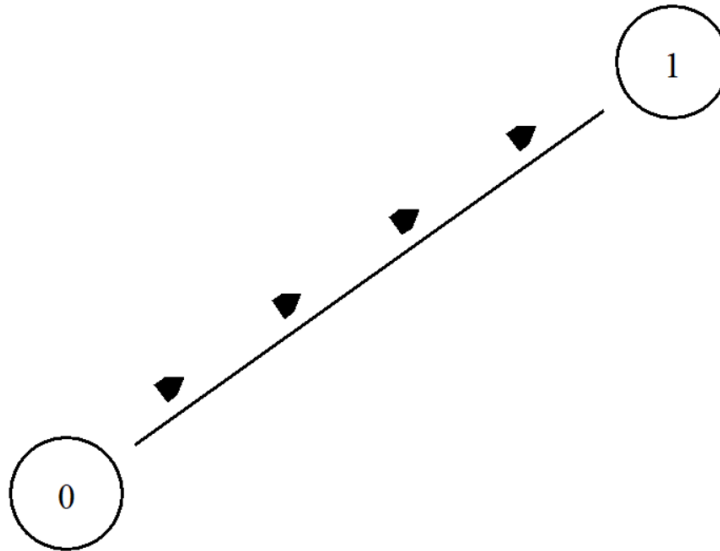
$$7,0 \text{ sec} * 1.600.000 \frac{\text{bits}}{\text{sec}} = 11.200.000 \text{ bits ή } 1.400.000 \text{ bytes}$$

- Πόσα bytes υπάρχουν πάνω στη γραμμή ζεύξης κάθε στιγμή; Επιβεβαιώστε την απάντησή σας από το animation.

Γνωρίζοντας το ρυθμό μετάδοσης και την καθυστέρηση της ζεύξης υπολογίζεται ότι θα είναι συνολικά  $200.000 \frac{\text{bytes}}{\text{sec}} * 0,01 \text{ sec} = 2000 \text{ bytes}$  πάνω στη γραμμή ζεύξης κάθε χρονική στιγμή. Εφόσον το κάθε πακέτο απαρτίζεται από 1000 bytes θα είναι ακριβώς δύο πακέτα κάθε χρονική στιγμή στη γραμμή και αυτό επιβεβαιώνεται από το animator όπως φαίνεται στο παρακάτω σχήμα. Προφανώς, τα παραπάνω ισχύουν για το χρονικό διάστημα 1,5 – 8,5 sec, στο οποίο μεταδίδεται.



- Ποια είναι η απάντησή σας στο προηγούμενο ερώτημα, αν διπλασιαστεί η καθυστέρηση της ζεύξης; Επιβεβαιώστε την απάντησή σας από το animation.  
Αν διπλασιαστεί η καθυστέρηση της ζεύξης (10 msec → 20 msec) εύκολα γίνεται κατανοητό ότι θα είναι τα διπλάσια δηλ. 4000 bytes, δηλαδή πάνω στη γραμμή ζεύξης κάθε χρονική στιγμή και επομένως τέσσερα πακέτα όπως φαίνεται και στο παρακάτω σχήμα από το animation.



- Εάν υποθέσουμε ότι σε κάθε πακέτο οι επικεφαλίδες του IP και του UDP μαζί έχουν μήκος 40 byte, ποιος είναι ο καθαρός ρυθμός μετάδοσης των δεδομένων σε bit/sec;  
Σε αυτή την περίπτωση θα είναι 1000 bytes – 40 bytes = 960 bytes για μετάδοση δεδομένων.  
Επομένως, ο καθαρός ρυθμός μετάδοσης θα είναι:  
$$\frac{960 \text{ bytes}}{0,005 \text{ sec}} = 192.000 \frac{\text{bytes}}{\text{sec}} = 1.536.000 \frac{\text{bits}}{\text{sec}}$$

- Ποιες παράμετροι μπορεί να αλλαχθούν για να μεταβληθεί ο ρυθμός μετάδοσης και με ποιες εντολές επιτυγχάνονται αυτές οι αλλαγές;  
Είτε αυξάνοντας το μέγεθος των πακέτων στην εντολή  
**\$traffic0 set packetSize\_ 1000**  
Είτε μειώνοντας το χρόνο ανάμεσα σε δύο διαδοχικά πακέτα στην εντολή  
**\$traffic0 set interval\_ 0.005**

- Αν επιθυμούμε να έχουμε καθαρό ρυθμό μετάδοσης δεδομένων ίσο με 1,1 Mbit/sec, μεταβάλλοντας κάθε φορά μία από τις ανωτέρω παραμέτρους, ποιες τιμές προτείνετε για κάθε μία; Ελέγξτε κάθε φορά αν οι απαντήσεις σας δίνουν ρυθμό μετάδοσης μικρότερο από τη χωρητικότητα της ζεύξης.

Ισχύει ότι: ρυθμός μετάδοσης =  $\frac{\text{μέγεθος πακέτου}}{\text{interval}}$ , όπου interval είναι το χρονικό διάστημα αποστολής δύο διαδοχικών πακέτων.

1. Μεταβάλλοντας το μέγεθος του πακέτου προκύπτει:

$$1,1 \frac{\text{Mbit}}{\text{sec}} = \frac{N}{0,005 \text{ sec}} \Rightarrow N = 0,0055 \text{ Mbit} \Rightarrow N = 687,5 \text{ bytes}$$

Τότε ο συνολικός ρυθμός μετάδοσης θα είναι:

$$\frac{(687,5+40) \text{ bytes}}{0,005 \text{ sec}} = \frac{727,5 \text{ bytes}}{0,005 \text{ sec}} = \frac{5.820 \text{ bits}}{0,005 \text{ sec}} = 1.164.000 \frac{\text{bits}}{\text{sec}}$$

2. Μεταβάλλοντας το interval προκύπτει:

$$1,1 \frac{\text{Mbit}}{\text{sec}} = \frac{(1000-40) \text{ bytes}}{t} \Rightarrow t = 0,006981 \text{ sec}$$

Τότε ο συνολικός ρυθμός μετάδοσης θα είναι:

$$\frac{(1000+40) \text{ bytes}}{0,006981 \text{ sec}} = \frac{1040 \text{ bytes}}{0,006981 \text{ sec}} = \frac{8.320 \text{ bits}}{0,006981 \text{ sec}} \cong 1.191.806 \frac{\text{bits}}{\text{sec}}$$

Παρατηρείται ότι οι παραπάνω δυο ρυθμοί μετάδοσης είναι σχεδόν ίδιοι καθώς επίσης δεν ξεπερνούν τη χωρητικότητα της ζεύξης η οποία είναι 8 Mbps.

- Για ποιες τιμές των ανωτέρω παραμέτρων θα αρχίσει να παρατηρείται οριακά η απώλεια πακέτων; Επιβεβαιώστε την απάντησή σας τρέχοντας το *tcl script* και το *animation*.

Η απώλεια πακέτων και άρα δεδομένων θα αρχίσει να παρατηρείται όταν θα γίνει προσπάθεια να μεταφερθούν περισσότερα δεδομένα από ότι «μπορεί» να μεταφέρει η ζεύξη μας, δηλαδή οριακά όταν ο ρυθμός μετάδοσης θα ισούται με την χωρητικότητα της ζεύξης, δηλαδή με 8 Mbps.

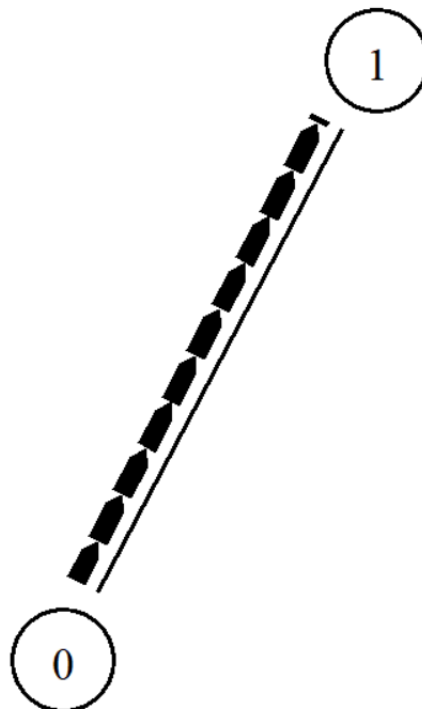
1. Για σταθερό μέγεθος πακέτου και ίσο με 1.000 bytes το παραπάνω παρατηρείται για interval ίσο με :

$$\frac{1000 \text{ bytes}}{8 \frac{\text{Mbits}}{\text{sec}}} = \frac{8000 \text{ bits}}{8.000.000 \text{ bits}} \text{ sec} = 0,001 \text{ sec}$$

2. Για σταθερό interval και ίσο με 0,005 sec το παραπάνω παρατηρείται για μέγεθος πακέτου ίσο με :

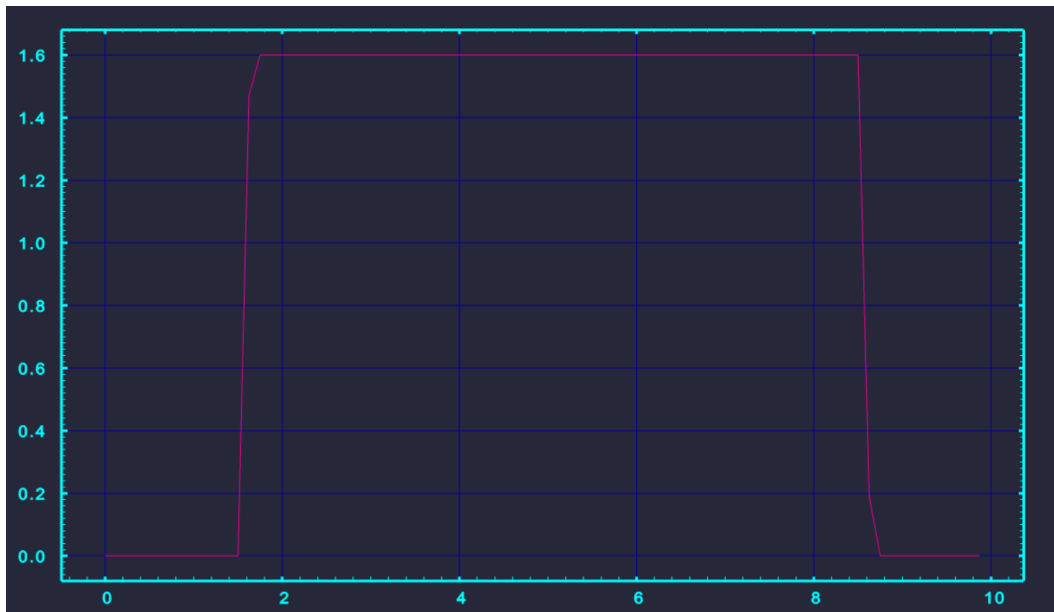
$$8 \frac{\text{Mbits}}{\text{sec}} * 0,005 \text{ sec} = 40.000 \text{ bits} = 5.000 \text{ bytes}$$

Τα παραπάνω επιβεβαιώνονται από το animation όπως φαίνεται στο παρακάτω σχήμα.



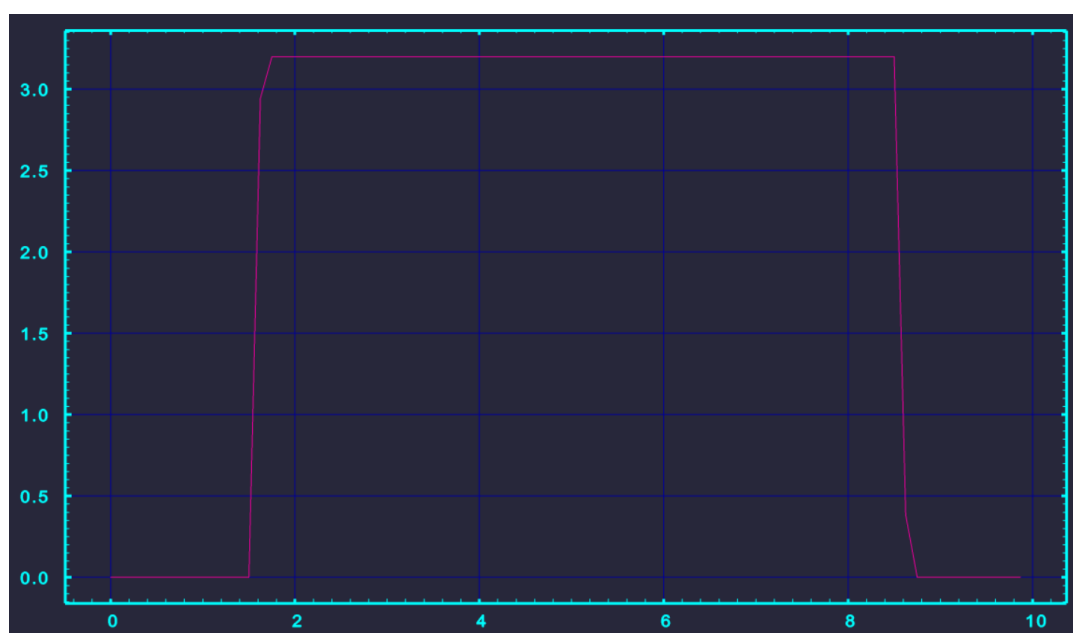
### 3. Μελέτη απλής τοπολογίας με το Xgraph

Σε αυτή την άσκηση τροποποιήθηκε το αρχικό Tcl script ώστε να αξιοποιηθεί το πρόγραμμα Xgraph και να δημιουργηθούν γραφικές παραστάσεις για την κίνηση στο δίκτυο μας. Παρακάτω παρουσιάζονται η αρχική γραφική παράσταση της κίνησης καθώς και οι απαντήσεις των ερωτήσεων της ενότητας.

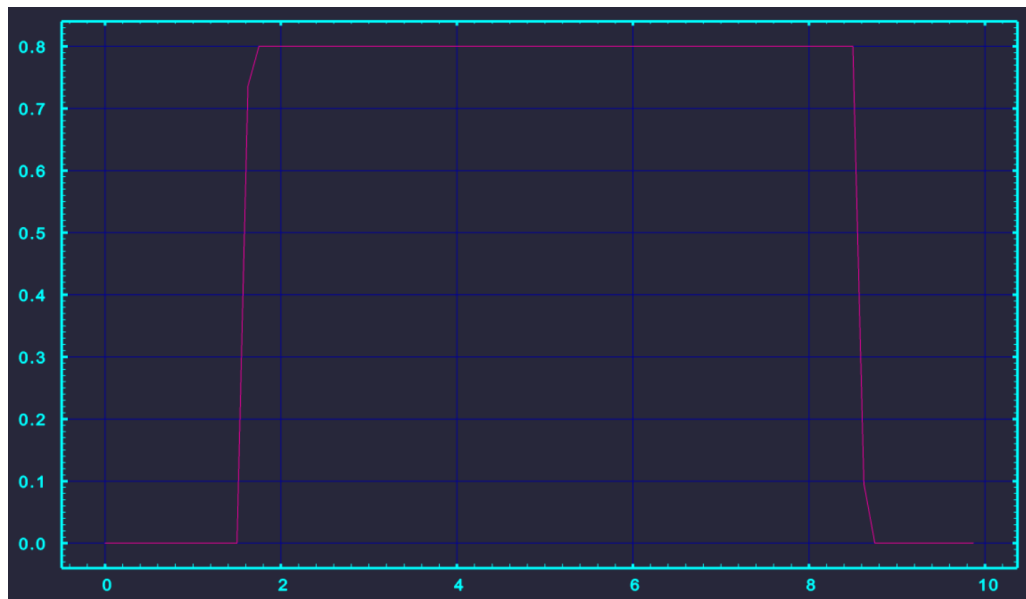


#### ΕΡΩΤΗΣΕΙΣ

- Μεταβάλλοντας την τιμή του μήκους του πακέτου διαπιστώστε και σχολιάστε πώς μεταβάλλεται η γραφική παράσταση της μεταφερόμενης κίνησης.  
Αυξάνοντας το μήκος του πακέτου π.χ. στα 2.000 bytes, προκύπτει η παρακάτω γραφική παράσταση από την οποία είναι φανερό ότι αυξήθηκε και ο ρυθμός μετάδοσης στα  $\approx 3,2$  Mbps.

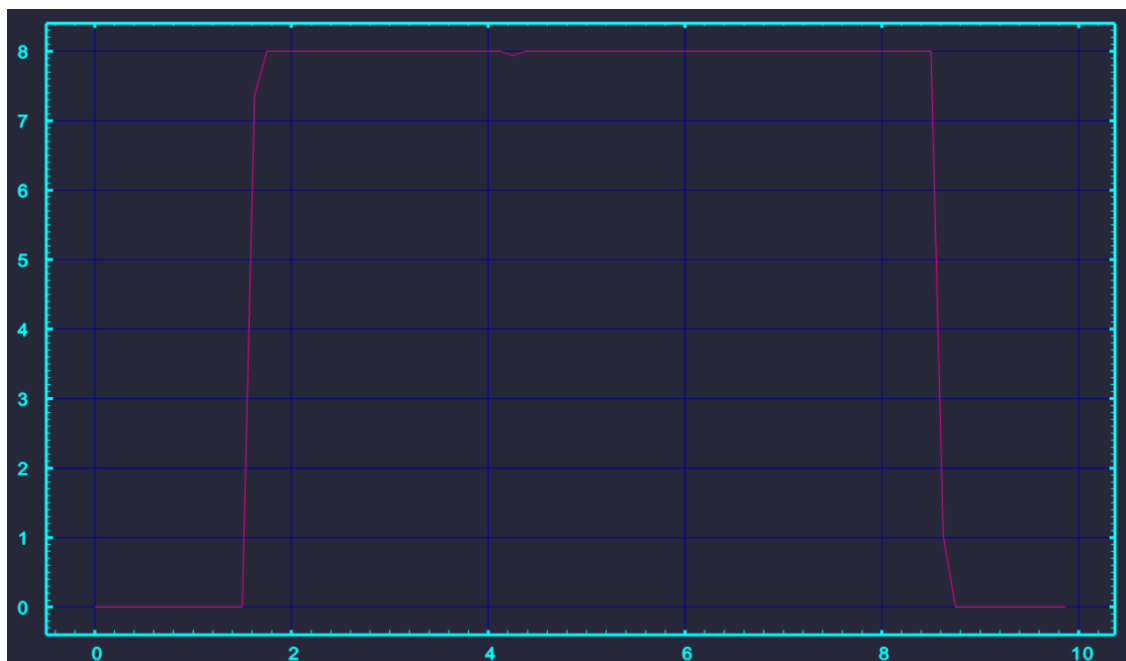


Ενώ, μειώνοντάς το π.χ. στα 500 bytes, λαμβάνεται η ακόλουθη γραφική παράσταση από την οποία παρατηρεί κανείς ότι μειώθηκε και ο ρυθμός μετάδοσης στα 0,8 Mbps.

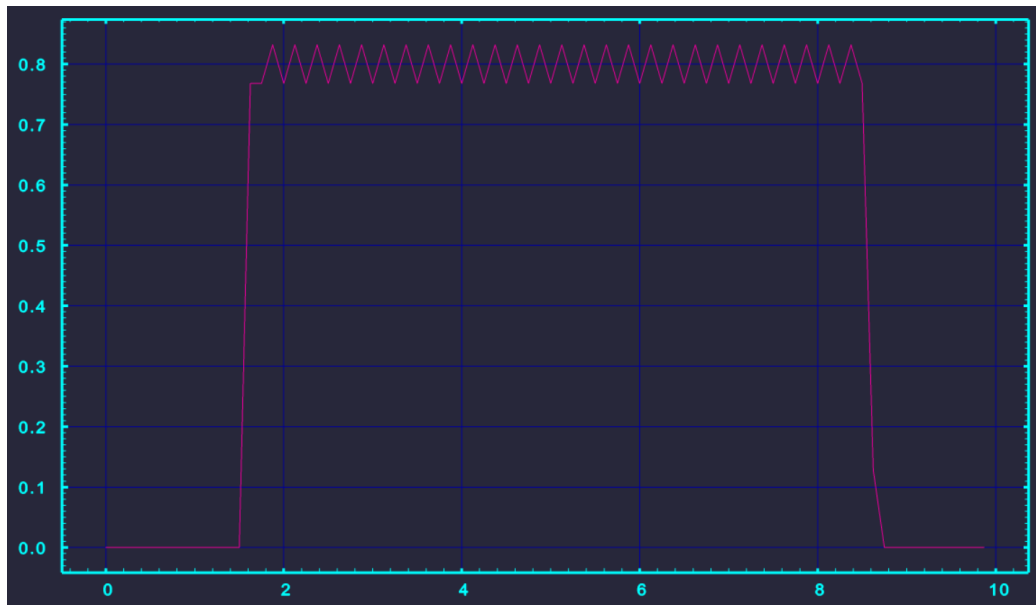


Τα παραπάνω ήταν αναμενόμενα αφού από τον τύπο υπολογισμού του ρυθμού μετάδοσης φαίνεται ότι αυτός είναι ανάλογος του μήκους του πακέτου, άρα αυξάνοντας το μήκος πακέτου θα αυξηθεί και ο ρυθμός μετάδοσης ενώ αντίθετα μειώνοντας το μήκος του πακέτου θα μειωθεί και ο ρυθμός μετάδοσης των δεδομένων.

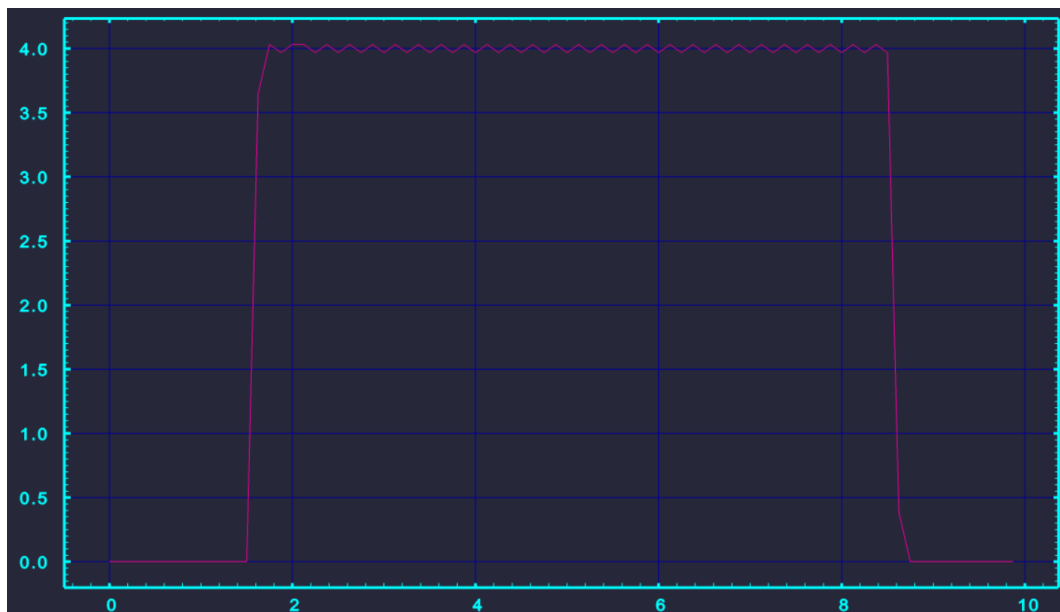
- Ποιο είναι το μέγιστο μήκος πακέτου που μπορεί να αποσταλεί χωρίς να ξεπερνάται η χωρητικότητα της γραμμής;  
Έχει υπολογιστεί παραπάνω ότι με σταθερό interval 0,005 sec, το μέγιστο μήκος πακέτου είναι 5.000 bytes, όπου και έχουμε οριακά απώλεια πακέτων δεδομένων.



- Διατηρώντας σταθερό το μήκος πακέτου, μεταβάλλετε τον ρυθμό μετάδοσης. Τι παρατηρείτε στη γραφική παράσταση της μεταφερόμενης κίνησης και πώς το ερμηνεύετε;  
Για να μεταβληθεί ο ρυθμός μετάδοσης κρατώντας σταθερό το μήκος πακέτου θα πρέπει να μεταβληθεί το interval.  
Αυξάνοντας το interval π.χ. στα 0,01 sec, γίνεται προφανές ότι ο ρυθμός μετάδοσης μειώθηκε στα 0,8 Mbps.



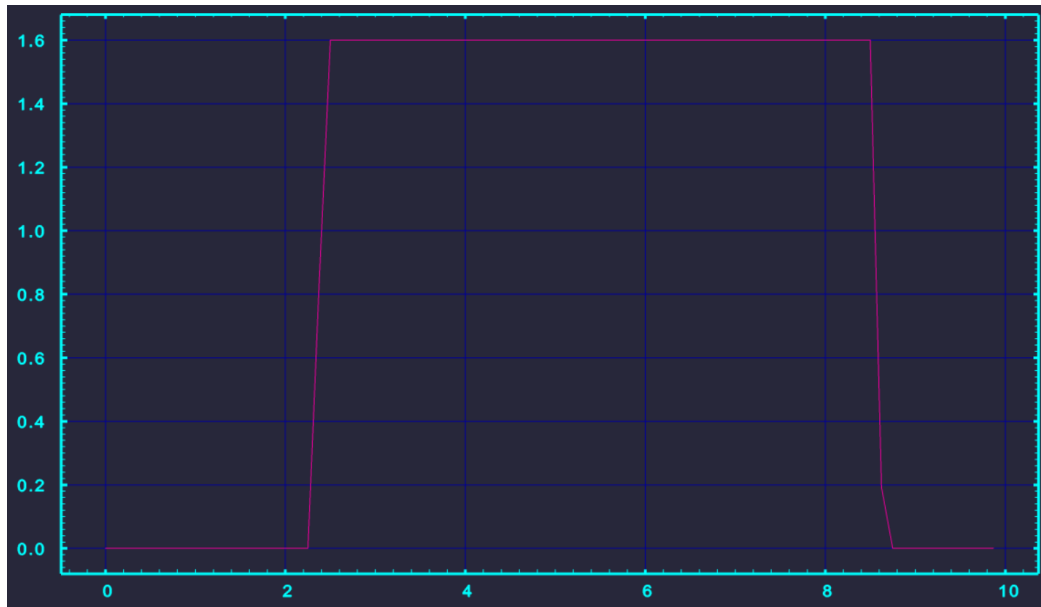
Αντίστοιχα, μειώνοντας το interval π.χ. στα 0,002 sec, παρατηρείται ότι ο ρυθμός μετάδοσης αυξήθηκε στα 4 Mbps.



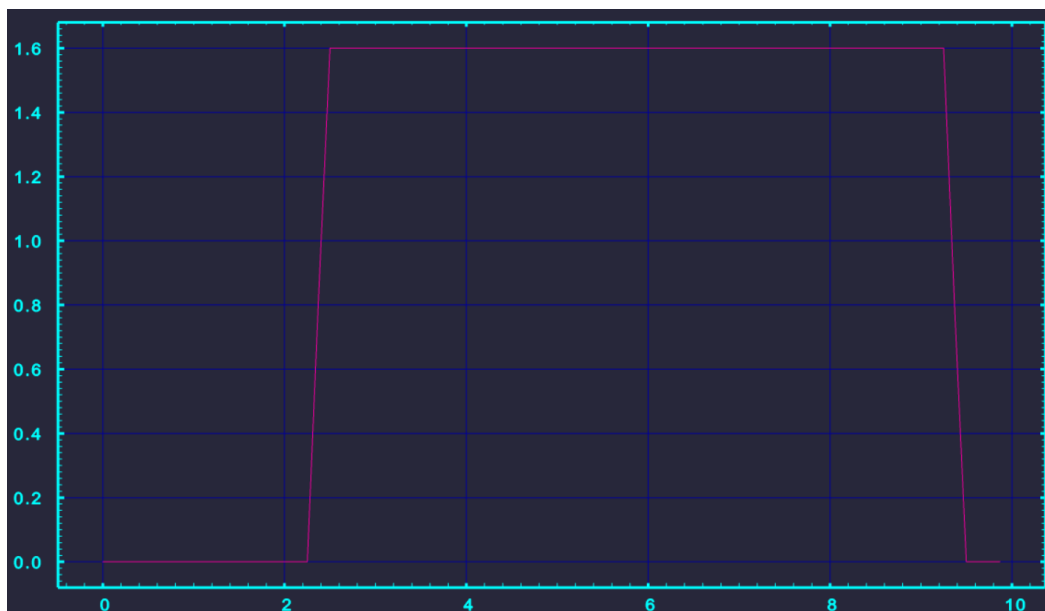
Τα παραπάνω, όπως και στην προηγούμενη ερώτηση, ήταν αναμενόμενα αφού εύκολα προκύπτουν από τον τύπο υπολογισμού του ρυθμού μετάδοσης ότι ο ίδιος είναι αντιστρόφως ανάλογος του interval.

- Αν  $a$  ( $a=6$ ) είναι το τελευταίο ψηφίο του αριθμού μητρώου σας, αυξήστε την καθυστέρηση της γραμμής σύνδεσης των δυο κόμβων σε  $(0.2 + (a \bmod 8)/10)$  (δηλ. 0,8) δευτερόλεπτα. Τι παρατηρείτε στην γραφική παράσταση της μεταφερόμενης κίνησης; Επαναφέρετε την καθυστέρηση στην αρχική τιμή.

Αυξάνοντας την καθυστέρηση της γραμμής από 0,005 sec σε 0,8 sec παρατηρείται στην γραφική παράσταση της κίνησης μια ολική μετατόπιση προς τα δεξιά αφού πλέον η μετάδοση αρχίζει στα 2,3 sec και όχι στα 1,505 sec. (αντίστοιχα και για τη λήξη της)



Μόνο στην είσοδο



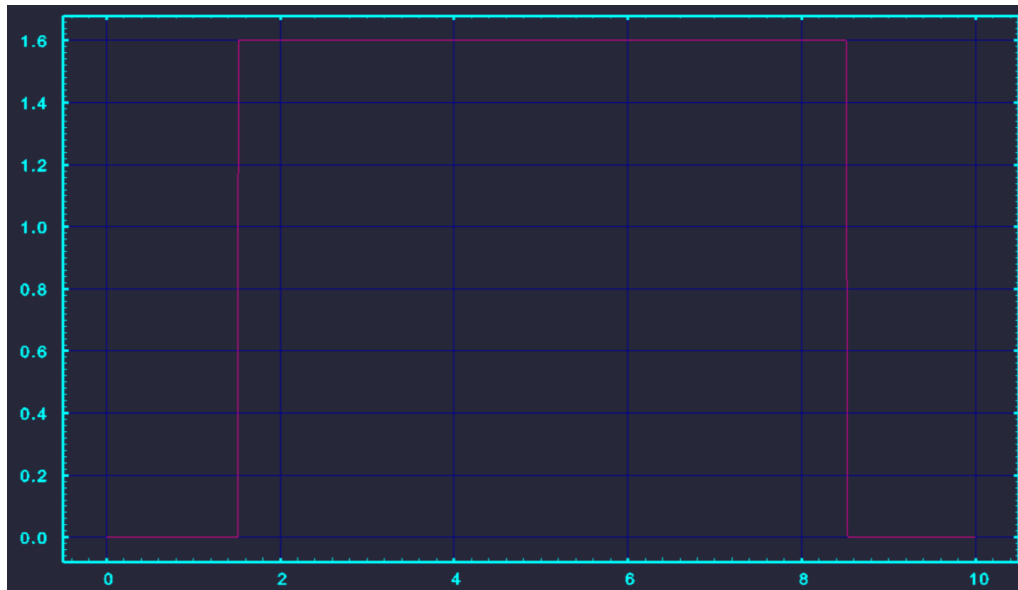
Και είσοδος και έξοδος



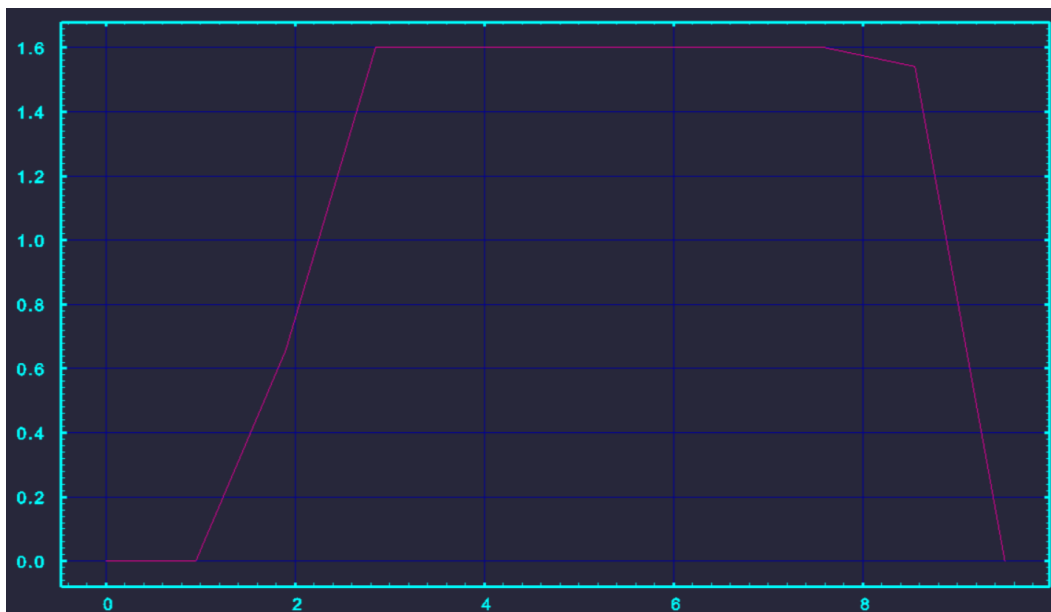
- Πώς επηρεάζει τη γραφική παράσταση ο χρόνος που επαναλαμβάνεται η διαδικασία “record”. Προτείνετε έναν κατάλληλο χρόνο για να επιτύχετε μια γραφική παράσταση στιγμιαίας κίνησης και μια μέσης.

Ο συγκεκριμένος χρόνος εκφράζει τον ρυθμό δειγματοληψίας της κίνησης, δηλαδή την περίοδο με την οποία το πρόγραμμα ελέγχει την κίνηση για μεταβολές στο ρυθμό μετάδοσης. Λογικό είναι ότι όσο μικρότερος είναι αυτός ο χρόνος τόσο πιο απότομες θα είναι και οι μεταβολές στη γραφική παράσταση και το αντίστροφο.

Για χρόνο π.χ. ίσο με 0,01 sec θα προκύψει μια γραφική παράσταση στιγμιαίας κίνησης όπως φαίνεται παρακάτω:



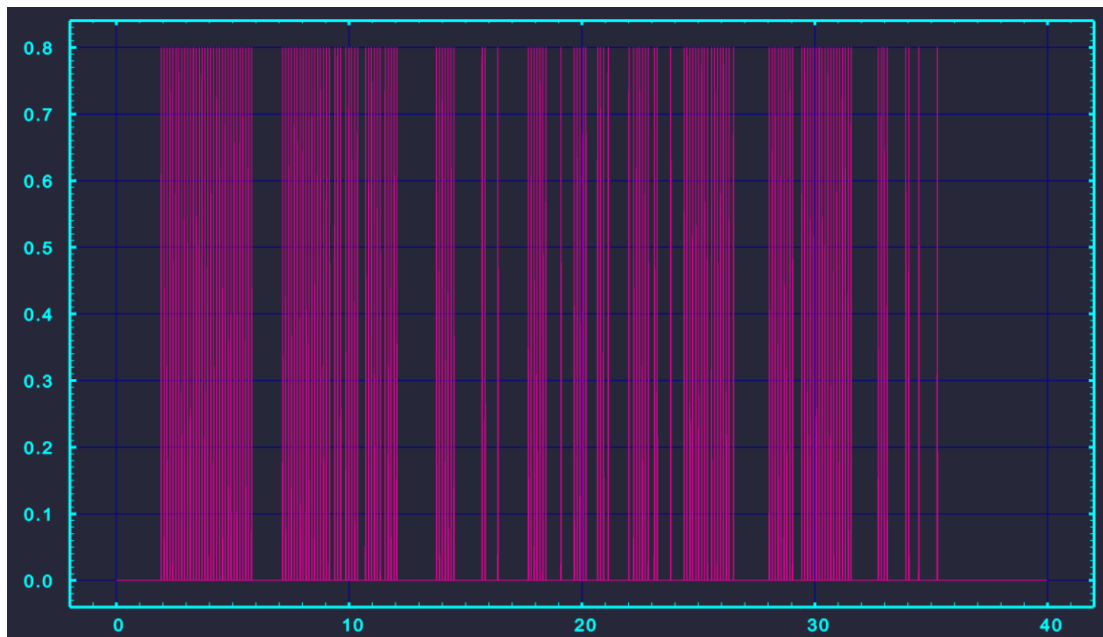
Αντίθετα, για χρόνο π.χ. ίσο με 0,95 sec θα προκύψει η αντίστοιχη γραφική παράσταση μέσης κίνησης κι αυτό φαίνεται στο παρακάτω σχήμα.



- Η κίνηση μεταξύ των δυο κόμβων είναι σταθερής ροής (CBR). Αλλάζτε την κίνηση σε εκθετική θέτοντας “Exponential” όπου υπάρχει “CBR”. Εξηγήστε τη γραφική παράσταση της κίνησης. Για να βγάλετε σωστά συμπεράσματα, θα πρέπει να αυξήσετε τον χρόνο αποστολής της κίνησης, και φυσικά της προσομοίωσης, τουλάχιστον σε 25 δευτερόλεπτα και να ξανατρέξετε το script.

Αλλάζοντας :

- Την γεννήτρια κίνησης από σταθερού ρυθμού bit (CBR) σε εκθετικού ρυθμού bit (Exponential).
- Την χρονική διάρκεια από διάδοσης δεδομένων σε 30 sec (1,5 – 35,5 sec)
- Τον χρόνο αποστολής κίνησης σε 0.1 sec
- Την περίοδο δειγματοληψίας σε 0.08 sec



Παρατηρεί κανείς πολύ πιο απότομες μεταβολές σε σύγκριση με τα προηγούμενα κι αυτό οφείλεται στη δομή των Exponential objects τα οποία δημιουργούν On/Off κίνηση. Στις περιόδους “on” παράγονται πακέτα συνεχούς ροής ενώ στις περιόδους “off” δεν παράγεται κίνηση. Οι χρόνοι για τους οποίους η κίνηση είναι “on” και οι αντίστοιχοι για τους οποίους η κίνηση είναι “off” ακολουθούν εκθετικές κατανομές, γι’ αυτό υπάρχει και κάποια σχετική τυχαιότητα στην παραπάνω γραφική παράσταση.

## Πηγαίος Κώδικας

```
#Δημιουργία object προσομοίωσης
set ns [new Simulator]
#Δημιουργία αρχείου "lab1.nam" για την καταχώρηση
των δεδομένων της προσομοίωσης σχετικά με το
πρόγραμμα "NAM"
set nf [open lab1.nam w]
$ns namtrace-all $nf
#Δημιουργία αρχείου "lab1.tr" για την καταχώρηση
των δεδομένων της προσομοίωσης σχετικά με το
πρόγραμμα "Xgraph"
set xf [open lab1.tr w]

#Διαδικασία καταγραφής των δεδομένων κίνησης
proc record {} {
    global sink xf
    set ns [Simulator instance]
    #Ορισμός της ώρας που η διαδικασία θα ξανακληθεί
    set time 0.125
    #Καταγραφή των bytes
    set bw [$sink set bytes_]
    #Λήψη της τρέχουσας ώρας
    set now [$ns now]
    #Υπολογισμός του bandwidth και καταγραφή αυτού
    στο αρχείο
    puts $xf "$now [expr
    (((bw/$time)*8)/1000000)]"
    #Κάνει την τιμή bytes_ 0
    $sink set bytes_ 0
    #Επαναπρογραμματισμός της διαδικασίας
    $ns at [expr $now+$time] "record"
}

#Διαδικασία η οποία κλείνει το αρχείο δεδομένων
proc finish {} {
    global ns nf xf
    $ns flush-trace
    close $nf
    close $xf
    exit 0
}

##Δύο κόμβοι και μια ζεύξη
#Ορισμός κόμβου «n0»
set n0 [$ns node]
```

```
#Ορισμός κόμβου «n1»
set n1 [$ns node]

#Ορισμός αμφίδρομης ζεύξης κόμβων «n0» και «n1»
#Εύρος ζώνης: 8Mbps, Καθυστέρηση: 10ms, Ουρά
αναμονής: Droptail
$ns duplex-link $n0 $n1 8Mb 10ms DropTail

##Αποστολή δεδομένων
#Δημιουργία του UDP agent «agent0»
set agent0 [new Agent/UDP]
$agent0 set packetSize_ 1000
#Προσάρτηση του agent «agent0» στον κόμβο «n0»
$ns attach-agent $n0 $agent0

#Δημιουργία της πηγής κίνησης CBR (σταθερού
ρυθμού bit) «traffic0»
set traffic0 [new Application/Traffic/CBR]
#Ορισμός μεγέθους πακέτου: "packetSize" = 1000
bytes
$traffic0 set packetSize_ 1000
#Αποστολή ενός πακέτου κάθε 0,005s
$traffic0 set interval_ 0.005
#Τοποθέτηση γεννήτριας στον agent «agent0»
$traffic0 attach-agent $agent0

#Δημιουργία του Sink agent «sink»
set sink [new Agent/LossMonitor]
#Προσάρτησή του agent «sink» στον κόμβο «n1»
$ns attach-agent $n1 $sink
#Σύνδεση των agents «agent0» και «sink»
$ns connect $agent0 $sink

#Ορισμός χρόνου αποστολής δεδομένων από τον
agent «agent0»
#Εκτέλεση της διαδικασίας «record» στην αρχή
$ns at 0.0 "record"
#Εναρξη στο 1,5sec
$ns at 1.5 "$traffic0 start"
#Λήξη στα 8,5sec
$ns at 8.5 "$traffic0 stop"
#Εκτέλεση της διαδικασίας «finish» μετά από 10,0 sec
$ns at 10.0 "finish"

#Εναρξη προσομοίωσης
$ns run
```