

# ΔΙΚΤΥΑ ΕΠΙΚΟΙΝΩΝΙΩΝ

## Εργαστηριακή Άσκηση 6

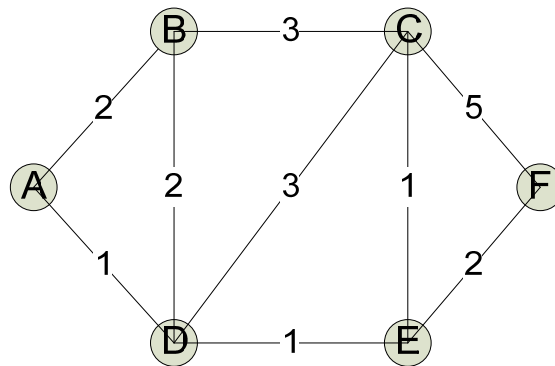
### Δρομολόγηση Διανύσματος Αποστάσεων

Σε αυτή την άσκηση θα μελετηθεί η επίδοση του πρωτοκόλλου δρομολόγησης *διανύσματος αποστάσεων* (*distance vector, DV*). Η λειτουργία του πρωτοκόλλου αυτού περιγράφεται αναλυτικά στο 5<sup>ο</sup> κεφάλαιο του βιβλίου του μαθήματος.

Η άσκηση περιλαμβάνει δύο μέρη. Στο πρώτο μέρος θα δημιουργηθεί ένα δίκτυο με σύνθετη τοπολογία με την οποία θα επαληθευθεί η λειτουργία του πρωτοκόλλου δρομολόγησης DV. Στο δεύτερο μέρος θα δειχθεί ότι για τοπολογίες συγκεκριμένης μορφής το πρωτόκολλο DV αποτυγχάνει.

#### 1. Επαλήθευση πρωτοκόλλου DV.

Αρχικά δημιουργούμε το δίκτυο του Σχ. 1, όπου ο αριθμός πάνω σε κάθε ζεύξη είναι το κόστος της.



Σχήμα 1: Τοπολογία πρώτου μέρους της άσκησης

Ο τρόπος με τον οποίο δημιουργείται αυτή η τοπολογία σας είναι ήδη γνωστός, με βάση τις προηγούμενες ασκήσεις. Ο κόμβος C στέλνει κίνηση CBR πάνω από UDP προς τον κόμβο A και ο κόμβος F στον D.

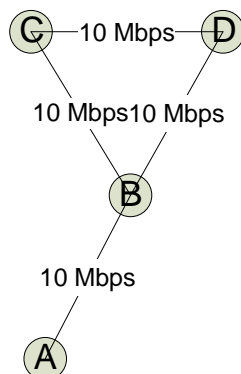
Ο κώδικας για την εκτέλεση αυτού του σεναρίου βρίσκεται στο Παράρτημα Α.

#### 1.1 Ερωτήσεις

- Ποιες διαδρομές ακολουθούν τα πακέτα για τις δύο μεταδόσεις;
- Να εκτελέσετε με το χέρι τον αλγόριθμο διανύσματος αποστάσεων και να επαληθεύσετε την ορθή λειτουργία του πρωτοκόλλου.
- Την χρονική στιγμή 1sec η γραμμή D-E τίθεται εκτός λειτουργίας. Ποιες διαδρομές ακολουθούν τώρα τα πακέτα για τις δύο ανωτέρω περιπτώσεις; Είναι οι συντομότερες διαθέσιμες διαδρομές;
- Στην αρχική τοπολογία, το κόστος της γραμμής D-E γίνεται 3. Ποιες είναι στην περίπτωση αυτή οι διαδρομές για τις δύο μεταδόσεις;
- 
- 
-

## 2. Αποτυχία πρωτοκόλλου *Distance Vector*

Δημιουργήστε την τοπολογία του Σχ. 2.



Σχήμα 2: Τοπολογία 2

Το κόμβος C αποστέλλει κίνηση CBR ρυθμού 960kbps προς τον κόμβο A, ενώ ο κόμβος D αποστέλλει κίνηση CBR ρυθμού 960kbps προς τον κόμβο B.

Τη χρονική στιγμή  $t=1\text{sec}$ , η γραμμή B – A τίθεται εκτός λειτουργίας.

Ο κώδικας για την εκτέλεση αυτού του σεναρίου βρίσκεται στο Παράρτημα Β.

### 2.1 Ερωτήσεις

- Τι συμβαίνει; Ποια διαδρομή ακολουθούν τα πακέτα που στέλνει ο C;
- Εξηγήστε αυτή τη συμπεριφορά.
- Να δώσετε τα σχήματα του  $xgraph$  που δείχνουν τον ρυθμό άφιξης δεδομένων στους προορισμούς τους.
- Επηρεάζεται η ροή πακέτων από τον κόμβο D προς στον B λόγω της αστοχίας της ζεύξης B – A;
- Να προτείνετε κάποια λύση για την αντιμετώπιση αυτής της συμπεριφοράς.

### 3. Παραρτήματα

#### Παράρτημα Α- Κώδικας για την προσομοίωση του πρώτου μέρους

```
### Arxeio prosomopiosis gia meleth epidosis prostokollou
### dromologisis Distance Vector (DV). To senario apoteleitai
### apo 6 komvovs syndedemenous me zeykseis diaforerikou varous,
### stin topologia tou parakato Sxhmatos.
###
###      B--3--C      Gia tin prosomoiysi o komvos A onomazetai
###      2/|      /|\5      node 0, o B->1, o C->2, o D->3, o E->4 kai
###      / |      / | \      o F->5
###      A  2  3  1  F
###      \ | /      | /      O kombos C stelnei CBR kinisi ston A, kai o
###      1\|/      |/2      F stelnei CBR kinisi ston D.
###      D--1--E
###
### Gia tin ektelesi tou arxeio pliktrologoume
###      ns ask7a.tcla
###
### Ta apotelesmata vriskontai sta arxeia out.nam (Animation)
### kai out.tr (Trace File). Epsilon ta arxeia out1.tr kai out2.tr
### perilamvanoun grafikes parastaseis tis kinisis (xgraph).

### FileName: ask7a.tcl
### Author:   D. J. Vergados

# Dimiourgia antikeimenou prosomoioti
set ns [new Simulator]

# Dimiourgia Animation
set nf [open out.nam w]
$ns namtrace-all $nf
# Dimiourgia Trace File
set trf [open out.tr w]
$ns trace-all $trf

# Entoles termatismou
proc finish {} {
    global ns nf f1 f2 trf
    $ns flush-trace
    close $nf
    close $f1
    close $f2
    close $trf
    exit 0
}

# Kathorismos protokollou dromolofisis
$ns rtproto DV

# Dimiourgia Komvon Diktyou
for {set i 0} {$i < 6} {incr i} {
    set n($i) [$ns node]
}

# Dimourgia zeykseon kai kathorismos costous
$ns duplex-link $n(0) $n(1) 1Mb 20ms DropTail
$ns cost $n(0) $n(1) 2
$ns cost $n(1) $n(0) 2
$ns duplex-link $n(0) $n(3) 1Mb 10ms DropTail
$ns cost $n(0) $n(3) 1
$ns cost $n(3) $n(0) 1
```

```

$ns duplex-link $n(1) $n(2) 1Mb 30ms DropTail
$ns cost $n(1) $n(2) 3
$ns cost $n(2) $n(1) 3
$ns duplex-link $n(1) $n(3) 1Mb 20ms DropTail
$ns cost $n(1) $n(3) 2
$ns cost $n(3) $n(1) 2
$ns duplex-link $n(2) $n(3) 1Mb 30ms DropTail
$ns cost $n(2) $n(3) 3
$ns cost $n(3) $n(2) 3
$ns duplex-link $n(2) $n(4) 1Mb 10ms DropTail
$ns cost $n(2) $n(4) 1
$ns cost $n(4) $n(2) 1
$ns duplex-link $n(2) $n(5) 1Mb 50ms DropTail
$ns cost $n(2) $n(5) 5
$ns cost $n(5) $n(2) 5
$ns duplex-link $n(3) $n(4) 1Mb 10ms DropTail
$ns cost $n(3) $n(4) 1
$ns cost $n(4) $n(3) 1
$ns duplex-link $n(4) $n(5) 1Mb 20ms DropTail
$ns cost $n(4) $n(5) 2
$ns cost $n(5) $n(4) 2

```

```

#Στρώμα Μεταφοράς, κόμβος 2: πηγή, κόμβος 03: προορισμός
set udp1 [new Agent/UDP]
$ns attach-agent $n(2) $udp1
$udp1 set fid_ 1
$ns color 1 red
set sink1 [new Agent/LossMonitor]
$ns attach-agent $n(0) $sink1

```

```

#Στρώμα Μεταφοράς, κόμβος 5: πηγή, κόμβος 3: προορισμός
set udp2 [new Agent/UDP]
$ns attach-agent $n(5) $udp2
$udp2 set fid_ 2
$ns color 2 blue
set sink2 [new Agent/LossMonitor]
$ns attach-agent $n(3) $sink2

```

```

#Σύνδεση των πηγών και των προορισμών
$ns connect $udp1 $sink1
$ns connect $udp2 $sink2

```

```

#Στρώμα εφαρμογής
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp2

```

```

# Didikasia Katagrafis kinisis
proc record {} {
    global sink1 sink2 f1 f2
    set ns [Simulator instance]
    #Ορισμός της ώρας που η διαδικασία θα ξανακληθεί
    set time 0.1
    #Καταγραφή των bytes
    set bw1 [$sink1 set bytes_]
    set bw2 [$sink2 set bytes_]

    #Λήψη της τρέχουσας ώρας
    set now [$ns now]

    #Υπολογισμός του bandwidth και καταγραφή αυτού στο αρχείο
    puts $f1 "$now [expr (($bw1/$time)*8)/1000000]"
}

```

```

puts $f2 "$now [expr (($bw2/$time)*8)/1000000]"

#Κάνει την μεταβλητή bytes 0
$sink1 set bytes_ 0
$sink2 set bytes_ 0

#Επαναπρογραμματισμός της διαδικασίας
$ns at [expr $now+$time] "record"
}

# Anoigma arkeion xgraph
set f1 [open out1.tr w]
set f2 [open out2.tr w]

#Ορισμός γεγονότων
$ns at 0.0 "record"
$ns at 0.3 "$cbr1 start"
$ns at 0.5 "$cbr2 start"
$ns at 2.5 "$cbr1 stop"
$ns at 2.5 "$cbr2 stop"
$ns at 3 "finish"

# Ektelesi tis prosomoiosis
$ns run

```

## ***Παράρτημα Β – Κώδικας για την προσομοίωση του δεύτερου μέρους***

```

### Arxeio prosomopiosis gia meleth epidosis prostokollou
### dromologisis Distance Vector (DV). To senario apoteleitai
### apo 4 komvous syndedemenous stin topologia tou parakato
### Sxhmatos.

###      10Mbps
###      C-----D      Gia tin prosomiosi o komvos A onomazetai
###      \      /      node 0, o B->1, o C->2, kai o D->3.
### 10Mbps 10Mbps
###      \ /
###      B
###      /      O kombos C stelnei CBR kinisi ston A, kai o
###      /10Mbps    D stelnei CBR kinisi ston B.
###      A

### Gia tin ektelesi tou arxeio pliktrologoume
### ns ask7b.tcl
###
### Ta apotelesmata vriskontai sta arxeia out.nam (Animation)
### kai out.tr (Trace File). Epsilon ta arxeia out1.tr kai out2.tr
### perilamvanoun grafikes parastaseis tis kinisis (xgraph).

### FileName: ask7b.tcl
### Author: D. J. Vergados

# Dimiourgia antikeimenou prosomoioti
set ns [new Simulator]

# Dimiourgia Animation
set nf [open out.nam w]
$ns namtrace-all $nf

# Dimiourgia Trace File
set trf [open out.tr w]
$ns trace-all $trf

# Entoles termatismou
proc finish {} {

```

```

        global ns nf f1 f2 trf
        $ns flush-trace
        close $nf
        close $f1
        close $f2
        close $trf
        exit 0
    }

# Kathorismos protokollou dromolofisis
$ns rtproto DV

# Dimiourgia Komvon Diktyou
for {set i 0} {$i < 4} {incr i} {
    set n($i) [$ns node]
}

# Dimourgia zeykseon
$ns duplex-link $n(0) $n(1) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(2) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(3) 1Mb 10ms DropTail
$ns duplex-link $n(2) $n(3) 1Mb 10ms DropTail

# Emfanisi ouras anamonnis sto nam
$ns duplex-link-op $n(2) $n(1) queuePos 0.5

#Στρώμα Μεταφοράς, κόμβος 2: πηγή, κόμβος 0: προορισμός
set udp1 [new Agent/UDP]
$ns attach-agent $n(2) $udp1
$udp1 set fid_ 1
$ns color 1 red
set sink1 [new Agent/LossMonitor]
$ns attach-agent $n(0) $sink1

#Στρώμα Μεταφοράς, κόμβος 3: πηγή, κόμβος 1: προορισμός
set udp2 [new Agent/UDP]
$ns attach-agent $n(3) $udp2
$udp2 set fid_ 2
$ns color 2 blue
set sink2 [new Agent/LossMonitor]
$ns attach-agent $n(1) $sink2

#Σύνδεση των πηγών και των προορισμών
$ns connect $udp1 $sink1
$ns connect $udp2 $sink2

#Στρώμα εφαρμογής
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 600
$cbr1 set interval_ 0.005

set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp2
$cbr2 set packetSize_ 600
$cbr2 set interval_ 0.005

# Didikasia Katagrafis kinisis
proc record {} {
    global sink1 sink2 f1 f2
    set ns [Simulator instance]
    #Ορισμός της ώρας που η διαδικασία θα ξανακληθεί
    set time 0.1
    #Καταγραφή των bytes

```

```

set bw1 [$sink1 set bytes_]
set bw2 [$sink2 set bytes_]
#Λήψη της τρέχουσας ώρας
set now [$ns now]
#Υπολογισμός του bandwidth και καταγραφή αυτού στο αρχείο
puts $f1 "$now [expr (($bw1/$time)*8)/1000000]"
puts $f2 "$now [expr (($bw2/$time)*8)/1000000]"
#Κάνει την μεταβλητή bytes 0
$sink1 set bytes_ 0
$sink2 set bytes_ 0
#Επαναπρογραμματισμός της διαδικασίας
$ns at [expr $now+$time] "record"
}

```

```

# Anoigma arkeion xgraph
set f1 [open out1.tr w]
set f2 [open out2.tr w]

```

```

#Ορισμός γεγονότων
$ns at 0.0 "record"
$ns at 0.3 "$cbr1 start"
$ns at 0.3 "$cbr2 start"
$ns at 2.9 "$cbr1 stop"
$ns at 2.9 "$cbr2 stop"
$ns at 3 "finish"

```

```

# Astoxia Zeyksis 0-1
$ns rtmodel-at 1.0 down $n(0) $n(1)

```

```

# Epanafora Zeyksis 0-1
$ns rtmodel-at 2.0 up $n(0) $n(1)

```

```

# Ektelesi tis prosomoiosis
$ns run

```