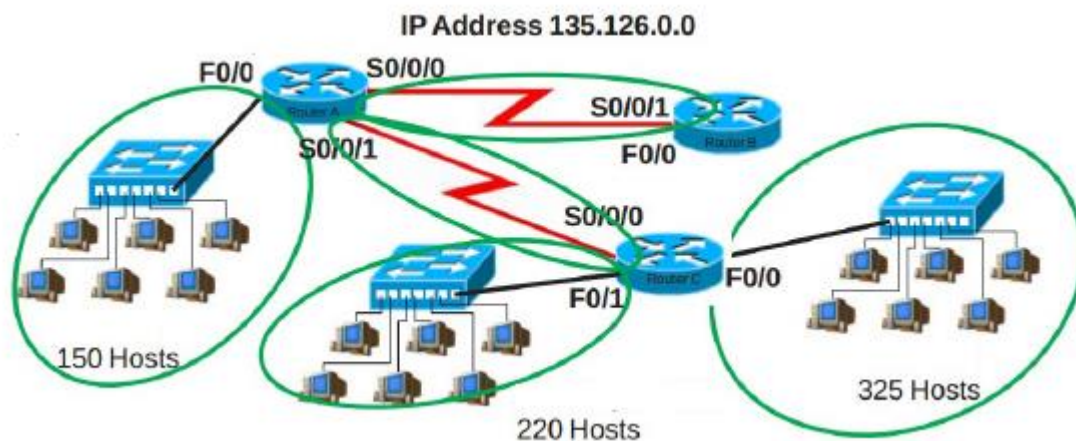


## ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ

### 2<sup>Η</sup> ΕΡΓΑΣΙΑ

Σιγούρου Άλκηστις Αικατερίνη ΑΜ: 1059661

#### Α ΜΕΡΟΣ



1. Για να προσδιορίσουμε την κλάση του δικτύου εξετάζουμε το πρώτο byte. Το πρώτο byte είναι το  $135_{(10)}$ , άρα βρίσκεται στην περιοχή από 128 έως 191. Επομένως ανήκει στην κλάση B. Οι διευθύνσεις τις class B αρχίζουν με  $10xx_{(2)}$ .

2. Custom Subnet Mask

Default Subnet Mask => 255.255.0.0

Το δίκτυο απαιτεί κατ' ελάχιστο 5 υποδίκτυα (όπως φαίνονται και στην εικόνα).

Για το +70% έχουμε :  $5 + 5 \cdot 0,7 = 5 + 3,5 = 8,5 \approx 9$ . Άρα απαιτούνται τουλάχιστον 4 subnet bits.

Άρα η custom subnet mask είναι :  $255.255.11110000_2.0 \rightarrow 255.255.240.0$

3. Ο ελάχιστος αριθμός υποδικτύων είναι 5.
4. Ο ελάχιστος αριθμός υποδικτύων προσαυξημένος με 70% είναι 9.
5. **Χωρίς το 70%** -> Έχουμε 5 υποδίκτυα, άρα χρειαζόμαστε 3 subnet mask bits, άρα παραμένουν ελεύθερα 13 bits για host addresses. Επομένως έχουμε  $2^{13} - 2 = 8192 - 2 = 8190$  host addresses σε κάθε υποδίκτυο.  
Οι δυο διευθύνσεις που αφαιρέσαμε είναι default για το ID του υποδικτύου και για την broadcast διεύθυνση το.

**Με το 70%** -> Έχουμε 9 υποδίκτυα, άρα χρειαζόμαστε 4 subnet mask bits, άρα παραμένουν ελεύθερα 12 bits για host addresses. Επομένως έχουμε  $2^{12} - 2 = 4096 - 2 = 4094$  host addresses σε κάθε υποδίκτυο.

## 6. Χωρίς το 70%

Υποδίκτυο	Host Bits			Εύρος IP διευθύνσεων
Υπ #0	0000 0001	0000 1111	0000 0000 1111 1111	135.126.0.0 135.126.31.255
Υπ #1	0010 0011	0000 1111	0000 0000 1111 1111	135.126.32.0 135.126.63.255
Υπ #2	0100 0101	0000 1111	0000 0000 1111 1111	135.126.64.0 135.126.95.255
Υπ #3	0110 0111	0000 1111	0000 0000 1111 1111	135.126.96.0 135.126.127.255
Υπ #4	1000 1001	0000 1111	0000 0000 1111 1111	135.126.128.0 135.126.159.255

**Άρα οι διευθύνσεις που είναι ελεύθερες για κάθε υποδίκτυο είναι :**

Υπ #0 : 135.126.0.1 - 135.126.31.254

Υπ #1 : 135.126.32.1 - 135.126.63.254

Υπ #2 : 135.126.64.1 - 135.126.95.254

Υπ #3 : 135.126.96.1 - 135.126.127.254

Υπ #4 : 135.126.128.1 - 135.126.159.254

**Με το 70%**

Υποδίκτυο	Host Bits			Εύρος IP διευθύνσεων
Υπ #0	00000 00001	000 111	0000 0000 1111 1111	135.126.0.0 135.126.15.255
Υπ #1	00010 00011	000 111	0000 0000 1111 1111	135.126.16.0 135.126.31.255
Υπ #2	00100 00101	000 111	0000 0000 1111 1111	135.126.32.0 135.126.47.255
Υπ #3	00110 00111	000 111	0000 0000 1111 1111	135.126.48.0 135.126.63.255
Υπ #4	01000 01001	000 111	0000 0000 1111 1111	135.126.64.0 135.126.79.255
Υπ #5	01010 01011	000 111	0000 0000 1111 1111	135.126.80.0 135.126.95.255
Υπ #6	01100 01101	000 111	0000 0000 1111 1111	135.126.96.0 135.126.111.255
Υπ #7	01110 01111	000 111	0000 0000 1111 1111	135.126.112.0 135.126.127.255
Υπ #8	10000 10001	000 111	0000 0000 1111 1111	135.126.128.0 135.126.143.255

**Άρα οι διευθύνσεις που είναι ελεύθερες για κάθε υποδίκτυο είναι :**

Υπ #0 : 135.126.0.1 - 135.126.15.254  
Υπ #1 : 135.126.16.1 - 135.126.31.254  
Υπ #2 : 135.126.32.1 - 135.126.63.254  
Υπ #3 : 135.126.48.1 - 135.126.61.254  
Υπ #4 : 135.126.64.1 - 135.126.79.254  
Υπ #5 : 135.126.80.1 - 135.126.95.254  
Υπ #6 : 135.126.96.1 - 135.126.111.254  
Υπ #7 : 135.126.112.1 - 135.126.127.254  
Υπ #8 : 135.126.128.1 - 135.126.143.254

7. Για να κάνουμε VLSM ξεκινάμε από το υποδίκτυο με τους περισσότερους hosts

Για το Υπ #0 έχουμε 325 Hosts και 2 επιπλέον διευθύνσεις για το subnet id και broadcast address . Επομένως  $\log_2 327 \approx 9$ .

$2^9=512$ . Άρα έχουμε από 135.126.0.0 έως 135.126.1.255

Για το Υπ #1 έχουμε 220 Hosts και 2 επιπλέον διευθύνσεις για το subnet id και broadcast address . Επομένως  $\log_2 222 \approx 8$ .

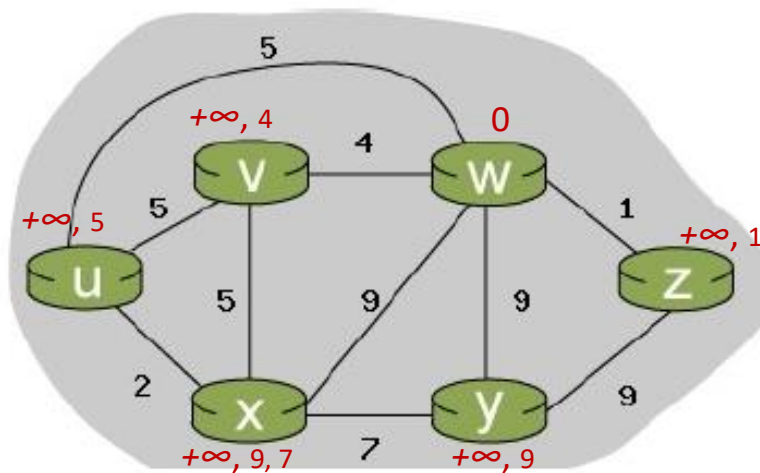
$2^8=256$ . Άρα έχουμε από 135.126.2.0 έως 135.126.2.255

Για το Υπ #2 έχουμε 150 Hosts και 2 επιπλέον διευθύνσεις για το subnet id και broadcast address . Επομένως  $\log_2 152 \approx 8$ .

$2^8=256$ . Άρα έχουμε από 135.126.3.0 έως 135.126.3.255

## **B ΜΕΡΟΣ**

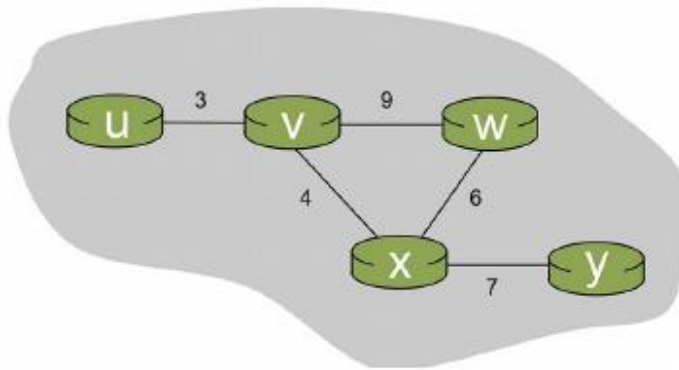
### ***α) Αλγόριθμος Dijkstra***



α/α	Κόμβος	Προσωρινή Ετικέτα	Τελική Τιμή
1	w	0	0
2	z	$+\infty, 1$	1
3	v	$+\infty, 4$	4
4	u	$+\infty, 5$	5
5	x	$+\infty, 9, 7$	7
6	y	$+\infty, 9$	9

## b) Αλγόριθμος Bellman-Ford

### 1. Αρχικά Διανύσματα

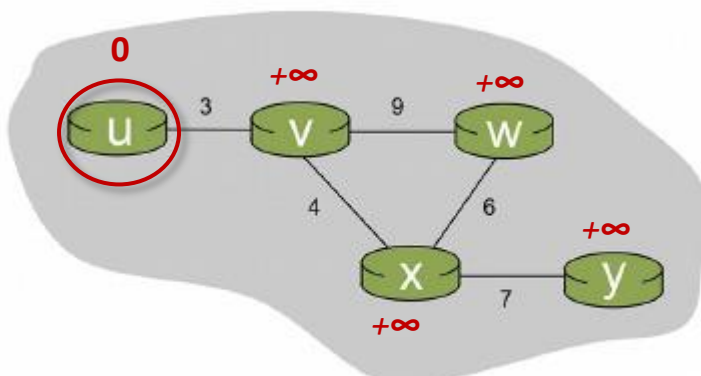


Προς Από	<i>u</i>	<i>v</i>	<i>x</i>	<i>w</i>	<i>y</i>
<i>u</i>	0	3	$+\infty$	$+\infty$	$+\infty$
<i>v</i>	3	0	4	9	$+\infty$
<i>x</i>	$+\infty$	4	0	6	7
<i>w</i>	$+\infty$	9	6	0	$+\infty$
<i>y</i>	$+\infty$	$+\infty$	7	$+\infty$	0

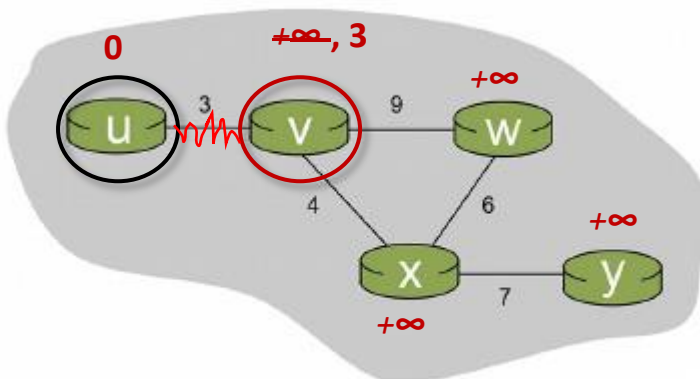
### 2. Αφού συγκλίνει ο αλγόριθμος

Αρχικά

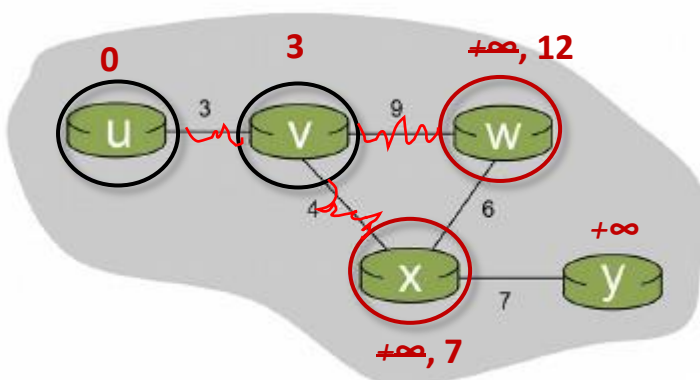
Έστω ότι ξεκινάμε από τον *u* έχουμε :



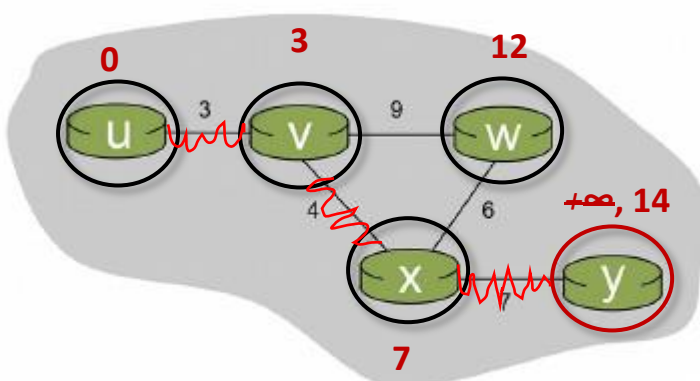
### 1st Iteration



### 2<sup>nd</sup> Iteration



### 3<sup>rd</sup> Iteration



### Τελικά

<i>Κόμβοι</i> <i>Iteration</i>	<i>u</i>	<i>v</i>	<i>x</i>	<i>w</i>	<i>y</i>
<i>0</i>	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$
<i>1</i>	0	3	$+\infty$	$+\infty$	$+\infty$
<i>2</i>	0	3	7	12	$+\infty$
<i>3</i>	0	3	7	12	14

Αναλόγως προκύπτει πως αν ξεκινήσουμε από τον *v* :

<i>Κόμβοι</i> <i>Iteration</i>	<i>u</i>	<i>v</i>	<i>x</i>	<i>w</i>	<i>y</i>
<i>0</i>	$+\infty$	0	$+\infty$	$+\infty$	$+\infty$
<i>1</i>	3	0	4	9	$+\infty$
<i>2</i>	3	0	4	9	22 (μέσω <i>vwxy</i> )
<i>3</i>	3	0	4	9	11 (μέσω <i>vxγ</i> )

Αναλόγως προκύπτει πως αν ξεκινήσουμε από τον *x* :

<i>Κόμβοι</i> <i>Iteration</i>	<i>u</i>	<i>v</i>	<i>x</i>	<i>w</i>	<i>y</i>
<i>0</i>	$+\infty$	$+\infty$	0	$+\infty$	$+\infty$
<i>1</i>	$+\infty$	4	0	6	7
<i>2</i>	18 (μέσω <i>xwvu</i> )	4	0	6	7
<i>3</i>	7	4	0	6	7

Αναλόγως προκύπτει πως αν ξεκινήσουμε από τον  $w$  :

Κόμβοι Iteration	$u$	$v$	$x$	$w$	$y$
0	$+\infty$	$+\infty$	$+\infty$	0	$+\infty$
1	$+\infty$	9	6	0	$+\infty$
2	13 (μέσω $wxvu$ )	9	6	0	20 (μέσω $wvxy$ )
3	12	9	6	0	13

Αναλόγως προκύπτει πως αν ξεκινήσουμε από τον  $y$  :

Κόμβοι Iteration	$u$	$v$	$x$	$w$	$y$
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	0
1	$+\infty$	$+\infty$	7	$+\infty$	0
2	$+\infty$	11	7	13	0
3	14	11	7	13	0

Παίρνοντας όλες τις πιθανές επιλογές βλέπουμε ότι η βέλτιστη λύση προκύπτει αν ξεκινήσουμε από τον κόμβο  $x$ .

## Γ ΜΕΡΟΣ

- Data Size/MTU = 4000/1500

	Length	Id	Fragflag	offset	
	4000	x	0	0	

Από τα 1500 bytes που μπορούμε να έχουμε σε κάθε κομμάτι τα 20 bytes είναι το header του IP, άρα η πληροφορία που μεταφέρεται είναι 1480 bytes . Την μετατόπιση (Offset) την υπολογίζουμε από την πράξη  $1480/8 = 185$  .Επίσης μόνο το τελευταίο πακέτο φέρει fragment flag (MF) ως 0 για να μπορεί να ξεχωρίζει ο δέκτης το τέλος του πακέτου .

	a/a	Length	Id	Fragflag	Offset	
	1°	1500	x	1	0	
	2°	1500	x	1	185	
	3°	1040	x	0	370	



- **Data Size/MTU = 2000/500**

	Length	Id	Fragflag	offset	
	2000	x	0	0	

Από τα 500 bytes που μπορούμε να έχουμε σε κάθε κομμάτι τα 20 bytes είναι το header του IP, άρα η πληροφορία που μεταφέρεται είναι 480 bytes . Την μετατόπιση (Offset) την υπολογίζουμε από την πράξη  $1480/8 = 60$ . Επίσης μόνο το τελευταίο πακέτο φέρει fragment flag (MF) ως 0 για να μπορεί να ξεχωρίζει ο δέκτης το τέλος του πακέτου .

	a/a	Length	Id	Fragflag	Offset	
	1°	500	x	1	0	
	2°	500	x	1	60	
	3°	500	x	1	120	
	4°	500	x	1	180	
	5°	80	x	0	240	

- **Data Size/MTU = 2000/1000**

	Length	Id	Fragflag	offset	
	2000	x	0	0	

Από τα 1000 bytes που μπορούμε να έχουμε σε κάθε κομμάτι τα 20 bytes είναι το header του IP, άρα η πληροφορία που μεταφέρεται είναι 980 bytes . Την μετατόπιση (Offset) την υπολογίζουμε από την πράξη  $980/8 = 122,5$  (στρογγυλοποιούμε προς τα πάνω και παίρνουμε το 123 για βήμα) . Επίσης μόνο το τελευταίο πακέτο φέρει fragment flag (MF) ως 0 για να μπορεί να ξεχωρίζει ο δέκτης το τέλος του πακέτου .

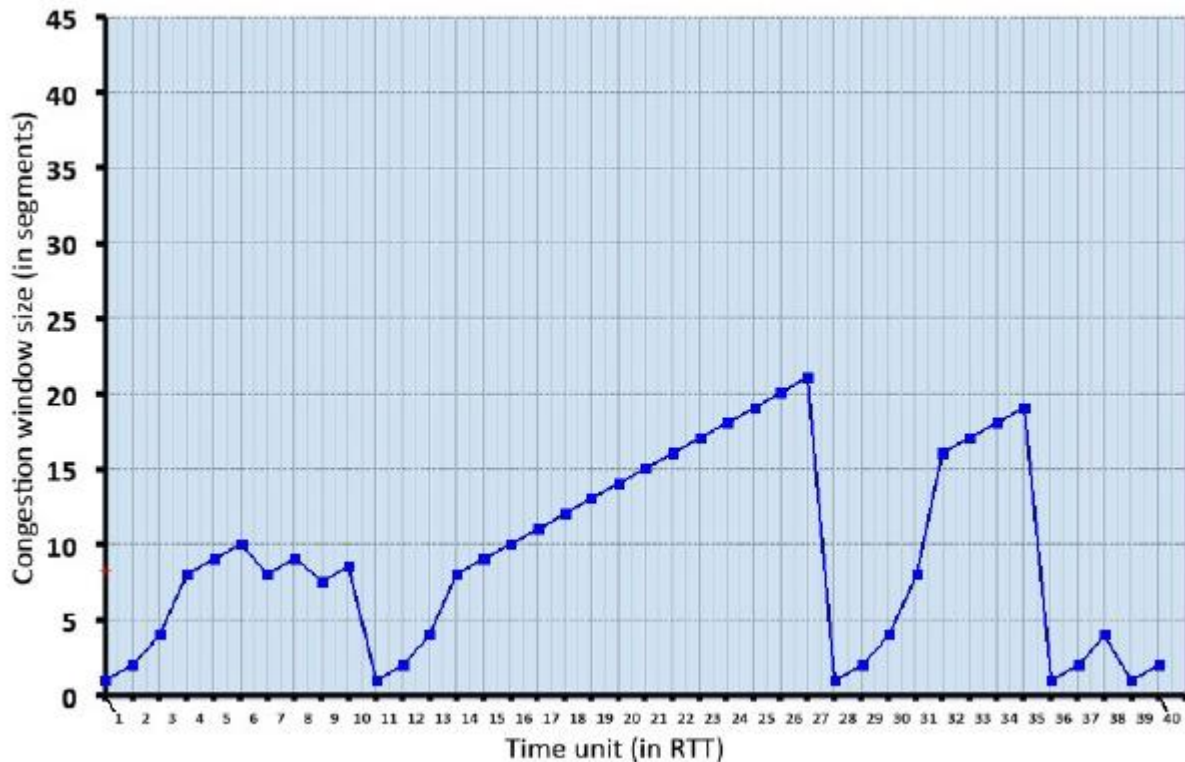
	a/a	Length	Id	Fragflag	Offset	
	1°	1000	x	1	0	
	2°	2000	x	1	123	
	3°	40	x	0	246	

- **Data Size/MTU = 4000/6000**

	Length	Id	Fragflag	offset	
	4000	x	0	0	

Το Data Size < MTU ,οπότε δεν χρειάζεται περαιτέρω σπάσιμο το πακέτο . Είναι εξ' αρχής ένα κομμάτι .

## Δ ΜΕΡΟΣ



- Η έκδοση του πρωτοκόλλου είναι TCP Reno . Αυτό το αντιλαμβανόμαστε από το χρονικό διάστημα  $t_6-t_{10}$  , όπου παρατηρείται το φαινόμενο της αποστολής 3πλότυπων ACK πακέτων ,και το πρωτόκολλο μπαίνει σε κατάσταση fast recovery .
- Το πρωτόκολλο βρίσκεται σε :
  - Slow start** : Τα χρονικά διαστήματα  $t_1-t_4$  ,  $t_{10}-t_{14}$  ,  $t_{28}-t_{32}$  και  $t_{36}-t_{38}$  ( θετική αύξηση του ConWin).
  - Congestion Avoidance** : Τα χρονικά διαστήματα  $t_4-t_6$  ,  $t_{14}-t_{27}$  και  $t_{32}-t_{35}$  (γραμμική αύξηση του ConWin).
  - Fast Recovery** : Τα χρονικά διαστήματα  $t_7-t_8$  και  $t_9-t_{10}$
- Το πρωτόκολλο αντιλαμβάνεται απώλεια πακέτου που οφείλεται σε :
  - τριπλά αντίγραφα ACK** : Τις χρονικές στιγμές  $t_6$  και  $t_8$  , αυτό το αντιλαμβανόμαστε από την διαδικασία fast recovery που ξεκινάει αμέσως μετά ,δηλαδή το ssthresh τίθεται σε τιμή ίση με το  $\frac{1}{2}$  του ConWin πριν την απώλεια προσαυξημένο με 3 .
  - timeout**: Τις χρονικές στιγμές  $t_{10}$  ,  $t_{27}$  ,  $t_{35}$  και  $t_{38}$  , αυτό φαίνεται από ότι το ConWin πέφτει κατευθείαν στην τιμή 1 .
- 1<sup>η</sup> μετάδοση – ConWin = 1 πακέτα No.1
  - 2<sup>η</sup> μετάδοση – ConWin = 2 πακέτα No.2,3
  - 3<sup>η</sup> μετάδοση – ConWin = 4 πακέτα No.4,5,6,7
  - 4<sup>η</sup> μετάδοση – ConWin = 8 πακέτα No.8,9,10,11,12,13,14,15
  - 5<sup>η</sup> μετάδοση – ConWin = 9 πακέτα No.16,17,18,19,20,21,22,23,24
  - 6<sup>η</sup> μετάδοση – ConWin = 10 πακέτα No.25,26,27,28,29,30,31,32,33,34
  - 7<sup>η</sup> μετάδοση – βρίσκεται σε fast recovery δεν μεταφέρονται πακέτα
  - 8<sup>η</sup> μετάδοση –λαμβάνει πάλι διπλότυπο ACK δεν μεταφέρονται πακέτα
  - 9<sup>η</sup> μετάδοση –βρίσκεται σε fast recovery δεν μεταφέρονται πακέτα
  - 10<sup>η</sup> μετάδοση – παθαίνει timeout δεν μεταφέρονται πακέτα

11<sup>η</sup> μετάδοση – ConWin = 1 πακέτα No.35

12<sup>η</sup> μετάδοση – ConWin = 2 πακέτα No.36,37

**Άρα έχουν μεταδοθεί 37 πακέτα την 12<sup>η</sup> στιγμή.**

13<sup>η</sup> μετάδοση – ConWin = 4 πακέτα No.38,39,40,41

14<sup>η</sup> μετάδοση – ConWin = 8 πακέτα No.42,43,44,45,46,47,48,49

15<sup>η</sup> μετάδοση – ConWin = 9 πακέτα No. 50,51,52,53,54,55,56,57,58

16<sup>η</sup> μετάδοση – ConWin = 10 πακέτα No.59,60,61,62,63,64,65,66,67,68

17<sup>η</sup> μετάδοση – ConWin =11 πακέτα No.69,70,71,72,73,74,75,76,77,78,79

18<sup>η</sup> μετάδοση – ConWin = 12 πακέτα No.80,81,82,83,84,85,86,87,88,89,90,91

19<sup>η</sup> μετάδοση – ConWin = 13 πακέτα No.92,93,94,95,96,97,98,99,100,101,102,103,104

20<sup>η</sup> μετάδοση – ConWin = 14 πακέτα No.105-No.118

21<sup>η</sup> μετάδοση – ConWin = 15 πακέτα No.119-No.133

22<sup>η</sup> μετάδοση – ConWin = 16 πακέτα No.134-No.149

23<sup>η</sup> μετάδοση – ConWin = 17 πακέτα No.150-No.166

24<sup>η</sup> μετάδοση – ConWin = 18 πακέτα No.167-No.184

25<sup>η</sup> μετάδοση – ConWin = 19 πακέτα No.185-No.203

26<sup>η</sup> μετάδοση – ConWin = 20 πακέτα No.204-No.223

27<sup>η</sup> μετάδοση – παθαίνει timeout δεν μεταφέρονται πακέτα

28<sup>η</sup> μετάδοση – ConWin = 1 πακέτα No.224

29<sup>η</sup> μετάδοση – ConWin = 2 πακέτα No.225,226

30<sup>η</sup> μετάδοση – ConWin = 4 πακέτα No.227,228,229,230

31<sup>η</sup> μετάδοση – ConWin = 8 πακέτα No.231-No.238

32<sup>η</sup> μετάδοση – ConWin = 16 πακέτα No.239-No.254

33<sup>η</sup> μετάδοση – ConWin = 17 πακέτα No.255-No.271

34<sup>η</sup> μετάδοση – ConWin = 18 πακέτα No.272-No.289

35<sup>η</sup> μετάδοση – παθαίνει timeout δεν μεταφέρονται πακέτα

36<sup>η</sup> μετάδοση – ConWin = 1 πακέτα No.290

37<sup>η</sup> μετάδοση – ConWin = 2 πακέτα No.291,292

38<sup>η</sup> μετάδοση – παθαίνει timeout δεν μεταφέρονται πακέτα

39<sup>η</sup> μετάδοση – ConWin = 1 πακέτα No.293

40<sup>η</sup> μετάδοση – ConWin = 2 πακέτα No.294,295

**Άρα μέχρι την λήξη έχουν μεταδοθεί 365 πακέτα**

5. Το ssthresh αλλάζει τις χρονικές στιγμές :

- t6, λαμβάνουμε διπλότυπο πακέτο ACK άρα το  $ssthresh = (1/2) conwin + 3 \Rightarrow ssthresh = 5 + 3 = 8$
- t8, λαμβάνουμε διπλότυπο πακέτο ACK άρα το  $ssthresh = (1/2) conwin + 3 \Rightarrow ssthresh = 4,5 + 3 = 7,5$  (στρογγυλοποιούμε στο 8) άρα  $ssthresh = 8$ .
- t10, έχουμε timeout άρα το  $ssthresh = (1/2) conwin \Rightarrow ssthresh = 4,5$
- t27, έχουμε timeout άρα το  $ssthresh = (1/2) conwin \Rightarrow ssthresh = 10,5$  (στρογγυλοποιούμε στην επόμενη κοντινή δύναμη του 2) άρα  $ssthresh = 16$
- t35, έχουμε timeout άρα το  $ssthresh = (1/2) conwin \Rightarrow ssthresh = 9,5$  (στρογγυλοποιούμε στην επόμενη κοντινή δύναμη του 2) άρα  $ssthresh = 16$
- t38, έχουμε timeout άρα το  $ssthresh = (1/2) conwin \Rightarrow ssthresh = 2$

6. Άμα δεν υπάρχουν απώλειες ,σημαίνει ότι η πορεία συνεχίζει κανονικά και αφού το conwin έχει φτάσει ήδη στο ssthresh , από εδώ και στο εξής κινείται γραμμικά σε κατάσταση congestion avoidance, άρα :

- t41 ->conwin =3
- t42 ->conwin=4
- t43-> conwin=5
- t44->conwin=6

7. Η τιμή του παραθύρου είναι 1 και του ssthresh είναι 2