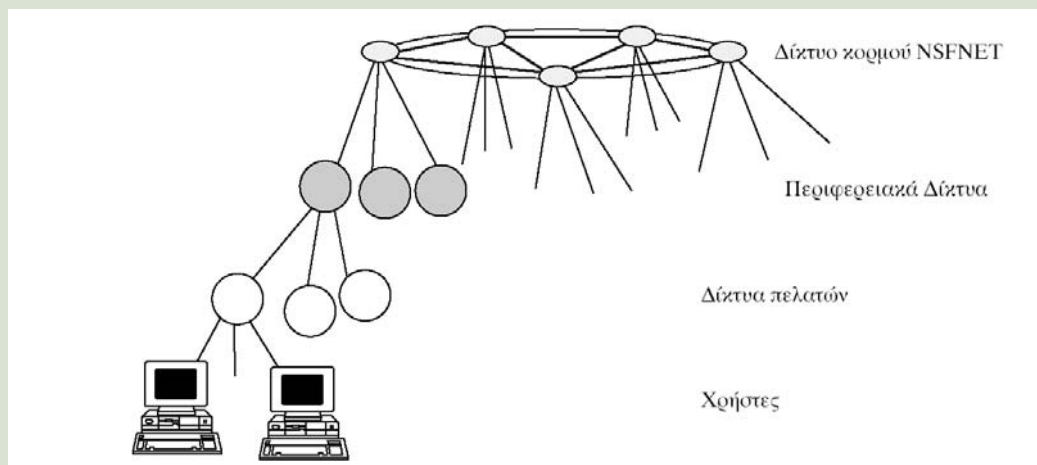
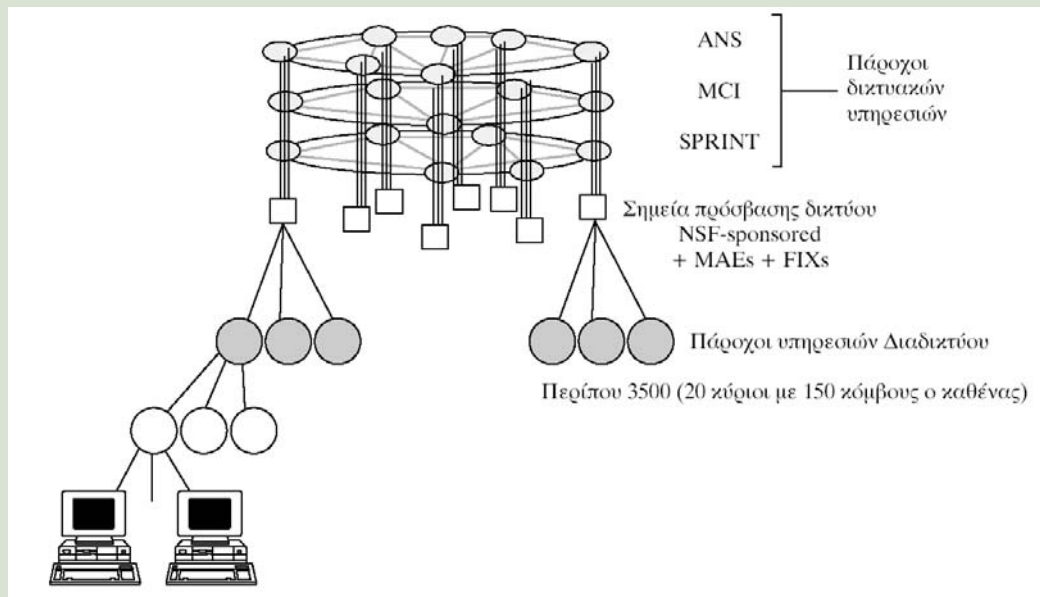


Σχεδιάγραμμα του Διαδικτύου γύρω στο 1990.



- 1962: προτείνεται η μεταγωγή πακέτων (Paul Baran – Rand Corporation)
- 1969: συνδέονται οι τέσσερις πρώτοι κόμβοι του ARPANET
- 1974: δημοσιεύονται οι βασικοί μηχανισμοί του TCP (Vint Cerf και Bob Kahn)
- 1982: ορίζεται το σύνολο πρωτοκόλλων TCP/IP για το ARPANET
- 1984: εισάγεται το σύστημα ονομασίας πεδίων (Domain Name System - DNS)
- 1986: δημιουργείται το NFSNET (στα 56 kbps)
- 1992: δημοσιοποιείται ο Παγκόσμιος Ιστός (World Wide Web – WWW)  
(Tim Berners-Lee - CERN)
- 2005: GENI (Global Environment for Network Innovations) ξεκινά μεγάλης κλίμακας προσπάθεια «εξέλιξης» του Internet με ανοικτή την προοπτική ανάπτυξης ενός νέου διαδικτύου

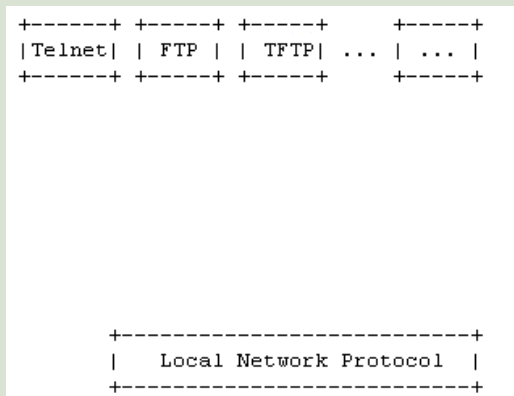
Σχεδιάγραμμα του Internet το 1997.



## Αρχιτεκτονική των δικτύων TCP/IP

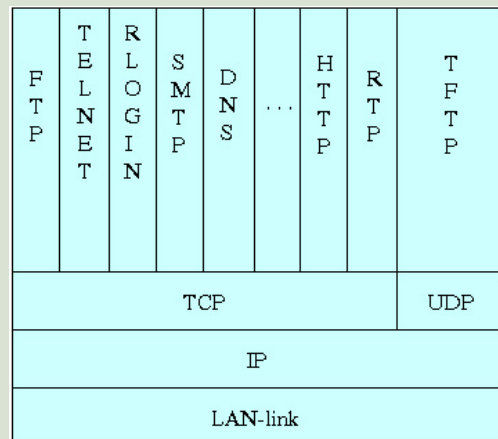
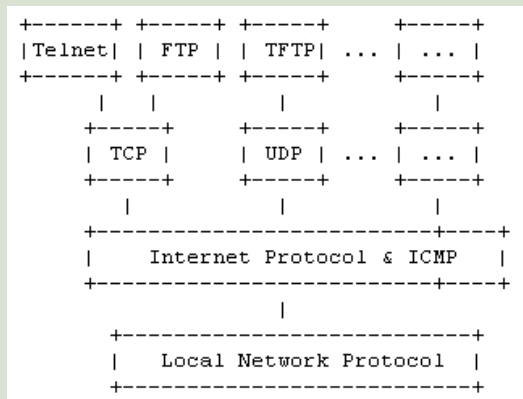
### Κίνητρο

- Διασύνδεση «ετερογενών» δικτύων που το καθένα εξυπηρετεί πολλές εφαρμογές
  - Διασύνδεση δικτύων
  - Διευθυνσιοδότηση





## Αρχιτεκτονική των δικτύων TCP/IP



- Επίπεδο Εφαρμογής
- Επίπεδο Μεταφοράς
- Επίπεδο Δικτύου
- Επίπεδο Τοπικού Δικτύου-Ζεύξης

## Επίπεδο Τοπικού Δικτύου-Ζεύξης

Ζεύξεις / LANs και οτιδήποτε είναι μεταξύ δρομολογητών

- Μια ζεύξη ενδέχεται να περιλαμβάνει περισσότερες από μία φυσικές ζεύξεις (π.χ. δρομολογητές IP που διασυνδέονται με μόνιμα εικονικά κυκλώματα ATM)
- Κύρια χαρακτηριστικά ζεύξης:
  - Μέγιστη μονάδα μεταφοράς (Maximum Transfer Unit - MTU) (π.χ. 1500 bytes για το Ethernet)
  - ρυθμός μετάδοσης (π.χ. 1Gbps για το gigabit Ethernet)
  - ρυθμός σφαλμάτων πακέτων (Packet Error Rate - PER) (για «τυπικές» ενσύρματες ζεύξεις << 1%)

### Διευθυνσιοδότηση κόμβων - Δρομολόγηση πακέτων

- πρωτόκολλο Internet (MTU=64Kbytes) → διατεμαστική παράδοση πακέτων
- Internet Control Message Protocol - ICMP → επίβλεψη / έλεγχος
  
- 1. Δρομολόγηση αυτοδύναμων πακέτων – διεύθυνση πηγής / προορισμού στο πακέτο
- 2. Στο IP οι δρομολογητές δεν κρατούν καταστάσεις σχετικά με τα πακέτα / συνδέσεις (σε αντίθεση με το ATM και τα εικονικά κυκλώματα) [⇒ Stateless]
- 3. Εύρωστο, λόγω 1, 2, αφού μετά από απώλεια ενός δρομολογητή τα πακέτα μπορούν να βρουν εναλλακτικούς δρόμους
- 4. Έλεγχος συμφόρησης δεν γίνεται από τους δρομολογητές (IP επίπεδο)
- 5. Κλιμάκωση ευνοείται λόγω των 1,2,4

### επιβλέπει τη διατεμαστική παράδοση των πακέτων

- Το επίπεδο μεταφοράς υλοποιείται από τα δύο τερματικά συστήματα  
⇒ το Internet (δρομολογητές) διατηρείται όσο γίνεται απλούστερο
- Δύο πρωτόκολλα επιπέδου μεταφοράς για το διαδίκτυο:
  - TCP (Transmission Control Protocol)
  - UDP (User Datagram Protocol)
- UDP απορρίπτει εσφαλμένα πακέτα, όχι επαναμετάδοση, όχι έλεγχο ρυθμού
- TCP έχει μηχανισμούς επαναμετάδοσης πακέτων και ελέγχου ροής / συμφόρησης

## Επίπεδο Εφαρμογής

υλοποιεί υπηρεσίες παράδοσης πληροφορίας που χρειάζονται εφαρμογές στο χρήστη

### FTP (File Transfer Protocol) [πρωτόκολλο μεταφοράς αρχείων]

- Μεταφορά αρχείων μεταξύ υπολογιστών μέσω δικτύου
- Συνήθως αμφίδρομη, δυνατότητα για δημιουργία, αλλαγή και ανάγνωση αρχείων από μακριά
- Τρόποι μεταφοράς : με συνεχή ροή (stream), με συμπίεση, με τμήματα (block) όταν υπάρχουν πολλά σφάλματα
- 2 TCP συνδέσεις – μία για τις εντολές και μία για τα δεδομένα
- Μία TCP διεργασία τρέχει συνεχώς στον κόμβο (host) περιμένοντας εντολές
- Οι εντολές φθάνουν από συγκεκριμένη θύρα (21)
- Έλεγχος και επαλήθευση της ταυτότητας του χρήστη γίνεται με κωδικό

### SMTP (Simple Mail Transfer Protocol) [απλό πρωτόκολλο μεταφοράς ταχυδρομείου]

- Μια διεργασία εξυπηρετητή ταχυδρομείου «τρέχει» συνέχεια και δέχεται μηνύματα ταχυδρομείου και διευθύνσεις προώθησης

## Επίπεδο Εφαρμογής

### TELNET [Πρωτόκολλο εικονικού τερματικού του διαδικτύου]

- Προσομοίωση άμεσης σύνδεσης ενός τερματικού με έναν υπολογιστή μέσω δικτύου. Απλό, ώστε να είναι υλοποιήσιμο από κάθε είδους τερματικό

### rlogin (remote log in) [εντολή της απομακρυσμένης σύνδεσης]

- σύνδεση τοπικού τερματικού με έναν απομακρυσμένο κόμβο υποδοχής

### HTTP (Hypertext Transfer Protocol) [πρωτόκολλο μεταφοράς υπερκειμένου]

- Ο διαφυλλιστής του Ιστού (web browser) είναι ο πελάτης και ο υπολογιστής όπου βρίσκεται το αιτούμενο υπερκείμενο είναι ο εξυπηρετητής
- Ανοίγεται μία TCP σύνδεση από την πελάτη, εκτελείται η μεταφορά του εγγράφου (η πολλών) και η σύνδεση κλείνει

### RTP (Real-time Transfer Protocol) [πρωτόκολλο μεταφοράς πραγματικού χρόνου]

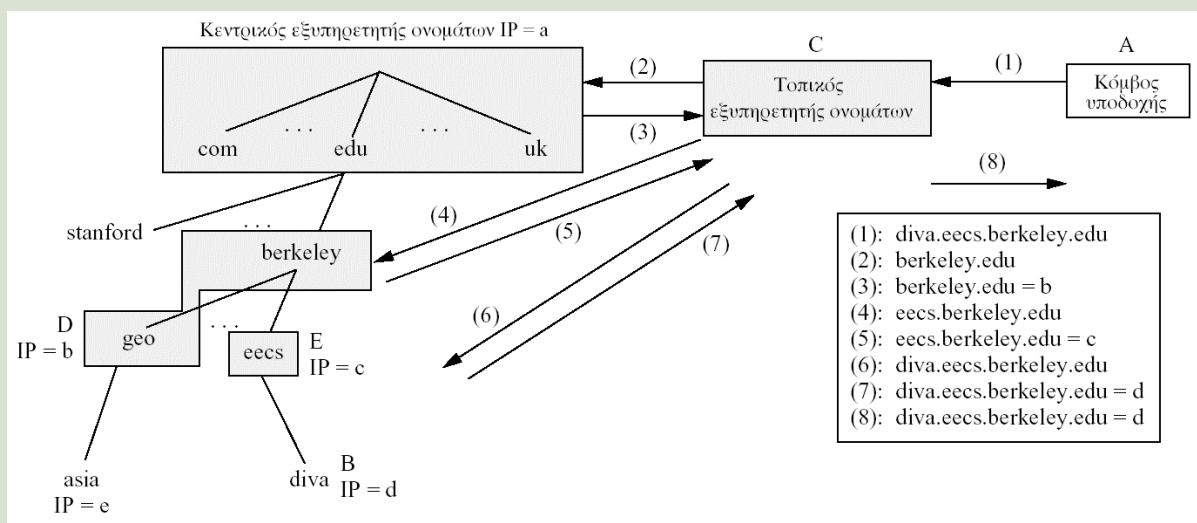
- Για τη μεταφορά ήχου και κινούμενης εικόνας μέσω του Διαδικτύου
- Προσθέτει επικεφαλίδες που ενημερώνουν για τον τρόπο συμπίεσης, το χρόνο παραγωγής του πακέτου, κ.λ.π.

## Όνόματα και Διευθύνσεις

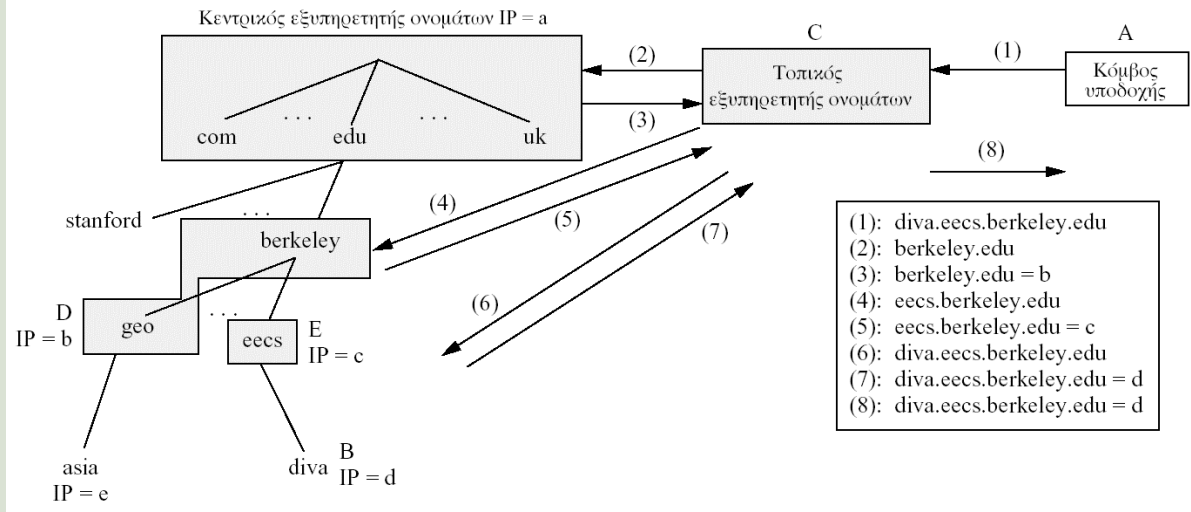
- Χρήστες (& εφαρμογές): χρήση ονομάτων
- Επίπεδο μεταφοράς και δικτύου: χρήση διευθύνσεων δικτύου
- ⇒ Βασικό πρόβλημα: όνομα υπολογιστή → διεύθυνση δικτύου (αντίστοιχο παράδειγμα: τηλεφωνικός κατάλογος)
- Ονόματα στο Internet: ιεραρχική δομή (όχι γεωγραφική, μεταβλητού «βάθους»)
  - πεδία (domains): .com, .edu, .gov, .int, .mil, ..., .gr, .fr, .be, ...
  - υπόπεδία (Sub-domains): berkeley.edu, uoa.gr, forth.gr, ...
  - υπό-υπόπεδία (sub-sub-domains): di.uoa.gr, thetis.ics.forth.gr, ...
- κάθε (υπο) πεδίο έχει έναν υπεύθυνο εξυπηρετητή ονομάτων (Domain Name Server - DNS)
- ένας DNS γνωρίζει τις διευθύνσεις των DNS των υποπεδίων του

## Όνόματα και Διευθύνσεις

Δομή ονομάτων στο IP και ανταλλαγή μηνυμάτων για την εύρεση από τον A της διεύθυνσης IP του B (diva.eecs.berkeley.edu)



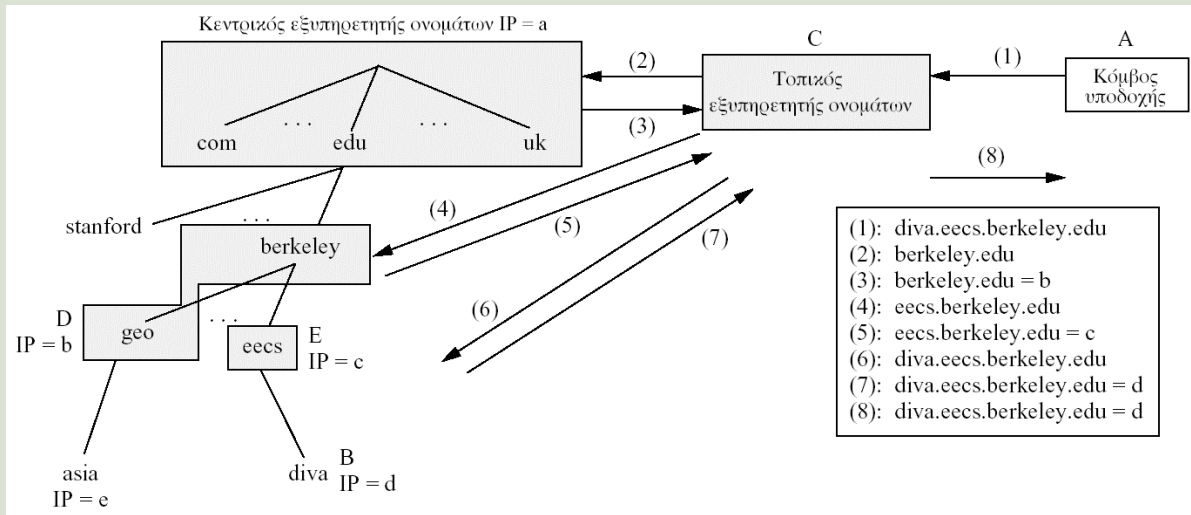
## Ονόματα και Διευθύνσεις



Ο Α αναζητά το όνομα του Β στη μνήμη cache (έστω ότι δεν το βρίσκει)

- (1) Ο Α ζητά τη διεύθυνση από τον τοπικό εξυπηρετητή ονομάτων, έστω C
- (2), (3) Ο C ζητά από τον κεντρικό εξυπηρετητή ονομάτων τη διεύθυνση του εξυπηρετητή ονομάτων του υποπεδίου berkeley.edu, έστω D

## Ονόματα και Διευθύνσεις



- (4), (5) Ο C ζητά από τον εξυπηρετητή ονομάτων D τη διεύθυνση του εξυπηρετητή ονομάτων του eecs.berkeley.edu, έστω E
- (6), (7) Ο C ζητά από τον E τη διεύθυνση του B
- (8) Ο C δίνει τη διεύθυνση του B στον A



## Διευθυνσιοδότηση

- Διευθύνσεις μήκους 32 bits (IPv4)  
⇒ Ασύμφορη η διατήρηση και ενημέρωση της πληροφορίας για κάθε κόμβο σε κάθε δρομολογητή

- Ιεραρχική δομή διευθύνσεων

Στόχοι:

- Μείωση της απαιτούμενης πληροφορίας / δρομολογητή
- Αποδοτική διαχείριση διευθύνσεων

Υλοποιήσεις:

- Διευθυνσιοδότηση βασιζόμενη σε κλάσεις
  - Υποδικτύωση (Subnetting)
  - CIDR (Classless Interdomain Routing)
- Δεκαδική γραφή διευθύνσεων ανά byte  
π.χ. 10000000 00100000 10011000 00011010 → 128.32.152.26

10000000	00100000	10011000	00011010
128	32	152	26

## Διευθυνσιοδότηση βασιζόμενη σε κλάσεις

Μορφή διευθύνσεων: "δίκτυο.κόμβος\_υποδοχής" ("network.host")

Διαισθητικά: Το Διαδίκτυο αποτελείται από δίκτυα που αποτελούνται από κόμβους

- Το πρόθεμα καθορίζει την κλάση (A → 0 B → 10 C → 110)  
(κλάση, # bits → δίκτυο, # bits → κόμβους) → (A, 8, 24), (B, 16, 16), (C, 24, 8)

Μορφή των διευθύνσεων και πλήθος των δικτύων και των κόμβων υποδοχής σε διαφορετικές κλάσεις.

	7	24				
A	0	Δίκτυο	Κόμβος υποδοχής	128 / 16 εκατομμύρια		
	14	16				
B	1	0	Δίκτυο	Κόμβος υποδοχής	16 χιλιάδες / 64 χιλιάδες	
	21	8				
C	1	1	0	Δίκτυο	Κόμβος υποδοχής	2 εκατομμύρια / 256

- Κλάση διευθύνσεων D δεσμευμένη για ομάδες πολυεκπομπής

## Διευθυνσιοδότηση βασιζόμενη σε κλάσεις

π.χ.

10000000 00100000 10011000 00011010 → 128.32.152.26

⇒ διεύθυνση κλάσης B

10	000000 00100000	10011000 00011010
----	-----------------	-------------------

⇒ δίκτυο 128.32

Τα μεγέθη των δικτύων δεν αντιστοιχούν καλά στις ανάγκες

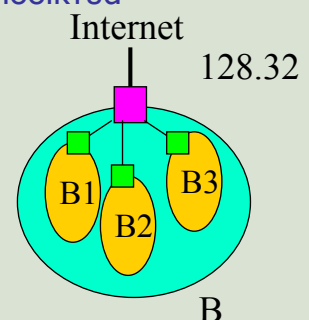
- Class-based διευθύνσεις μπορεί να μην είναι αρκετές για τους hosts ενός π.χ. Class C δικτύου ή να είναι πολλές για ένα π.χ. class A δίκτυο
- ⇒ Αναγκαίο να υπάρξει ευελιξία στα όρια μεταξύ Network και host διεύθυνσης
- ⇒ subnetting και classless addressing

## Υποδικτύωση (Subnetting)

Μηχανισμός διαχωρισμού ενός (μεγάλου) δικτύου σε υποδίκτυα

Προσθήκη ενός ακόμα επιπέδου στην ιεραρχία

- Ένα μεγάλο (π.χ. κλάσης B) δίκτυο χωρίζεται σε πολλά μικρότερα που:
  - όλα έχουν την ίδια (κλάσης B) διεύθυνση δικτύου, αλλά
  - διαφορετικές διευθύνσεις υποδικτύου (subnet)
- «Εξωτερικά» είναι γνωστό μόνο το δίκτυο
  - Οι «εξωτερικοί» δρομολογητές γνωρίζουν μόνο τη διεύθυνση του δικτύου και όχι του υποδικτύου(⇒ δεν επιβαρύνονται με περισσότερες διευθύνσεις στη δρομολόγηση)
- Εντός του δικτύου γίνεται η διαχείριση πολλαπλών υποδικτύων
  - «Οι εσωτερικοί» δρομολογητές δρομολογούν την κίνηση βάση του υποδικτύου

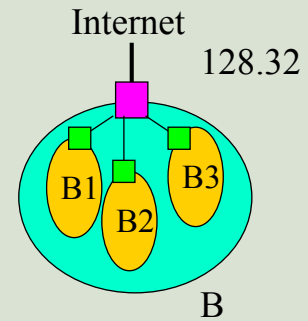


## Υποδικτύωση (Subnetting)

- Σε κάθε κόμβο δίνεται μία διεύθυνση δικτύου και μία μάσκα υποδικτύου



16K/64K



Subnet mask: 255.255.255.0

Network address: 128.32.134.26

- Το υποδίκτυο του κόμβου καθορίζεται από bit-by-bit "AND" της IP διεύθυνσης και της μάσκας. Π.χ.:

10000000 00100000 10011000 00011010 (128.32.152.26) διεύθυνση δικτύου κόμβου

⊗ 11111111 11111111 11111111 00000000 (255.255.255.0) μάσκα υποδικτύου

= 10000000 00100000 10011000 00000000 (128.32.152.0)

⇒ υποδίκτυο 128.32.152.0

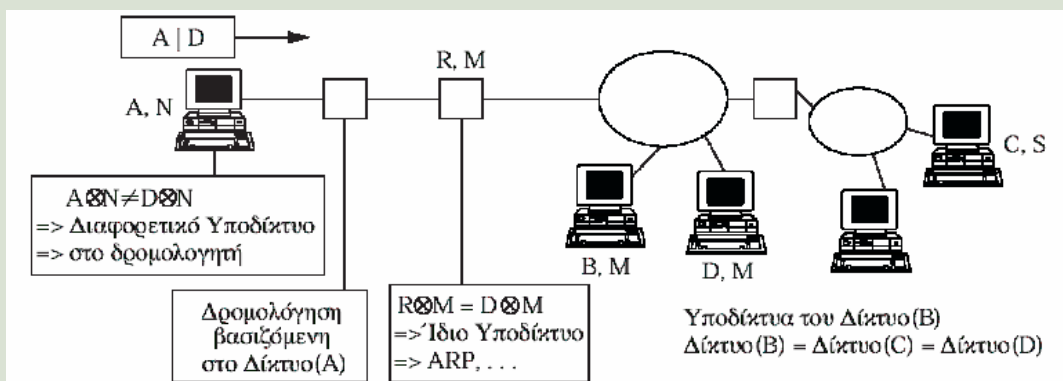
## Υποδικτύωση (Subnetting)

Υπολογιστής A με διεύθυνση A = 128.32.152.26 και μάσκα N = 255.255.255.0 θέλει να στείλει πακέτο στον υπολογιστή B με διεύθυνση B = 128.32.134.56

$A \otimes N = 128.32.152.0 \neq B \otimes N = 128.32.134.0$

⇒ Ο B δε βρίσκεται στο ίδιο υποδίκτυο

Ο A στέλνει το πακέτο στο δρομολογητή του (υποδικτύου του)



## CIDR (Classless Interdomain Routing) - Υπερδικτύωση

- Αγνοείται ο ορισμός των κλάσεων
- Επιτρέπονται διευθύνσεις δικτύων με αυθαίρετο μήκος (δηλ. ισοδύναμα, μάσκες αυθαίρετου μήκους και χωρίς κλάσεις)
- Το δίκτυο καθορίζεται από (διεύθυνση δικτύου, μάσκα)
- Οι δρομολογητές για κάθε δίκτυο αποθηκεύουν (διεύθυνση δικτύου, μάσκα)
- Η δρομολόγηση βασίζεται στην εύρεση του μέγιστου προθέματος που ταιριάζει με τη διεύθυνση προορισμού του πακέτου
- Βασίζεται στη απόδοση «συνεχόμενων» διευθύνσεων
- Επιτρέπει την «ομαδοποίηση» δικτύων («απαιτείται» ιεραρχική δομή δικτύου, ιεραρχική απόδοση διευθύνσεων)

## CIDR (Classless Interdomain Routing)

- Το δίκτυο καθορίζεται από (διεύθυνση δικτύου, μάσκα)

195.134.64.0/18 (μάσκα: 18 πρώτα bits ορίζουν το δίκτυο)

195.134.64.0, 255.255.192.0

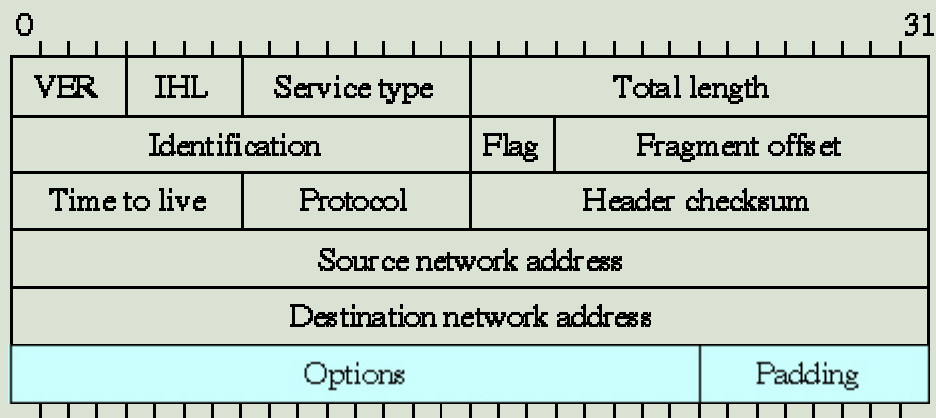
195.134.64.0 - 195.134.127.255

11000011	10000110	01000000	00000000	195.134.64.0
----------	----------	----------	----------	--------------

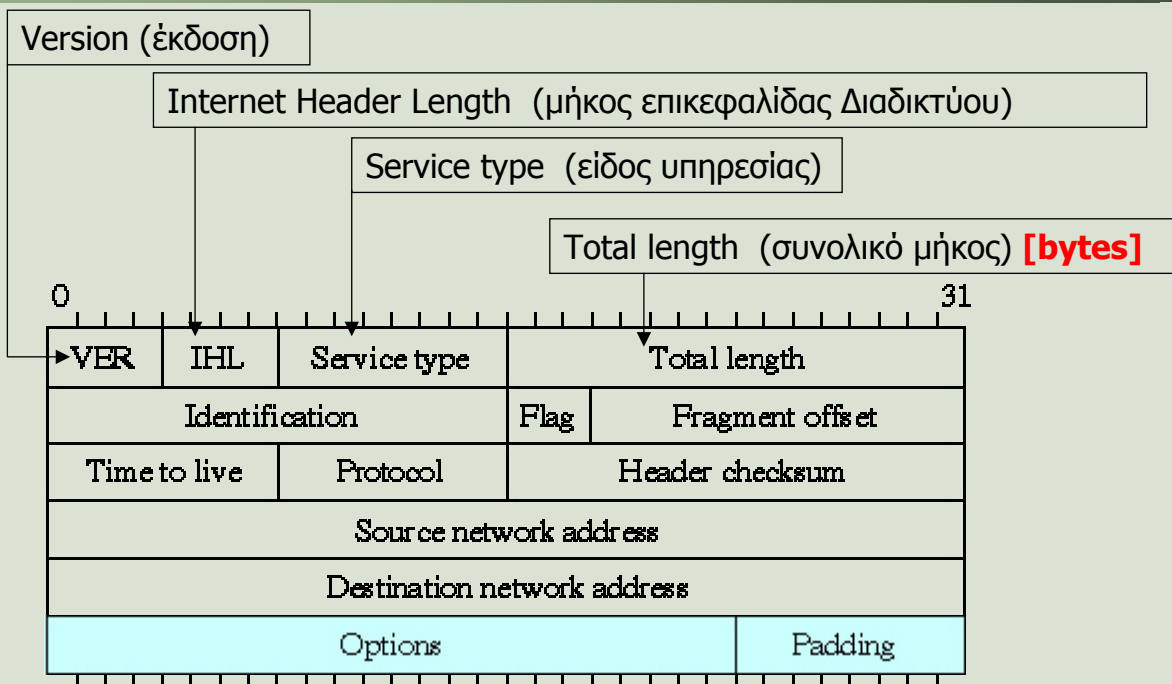
11111111	11111111	11000000	00000000	255.255.192.0
----------	----------	----------	----------	---------------

11000011	10000110	01xxxxxx	xxxxxxx	
----------	----------	----------	---------	--

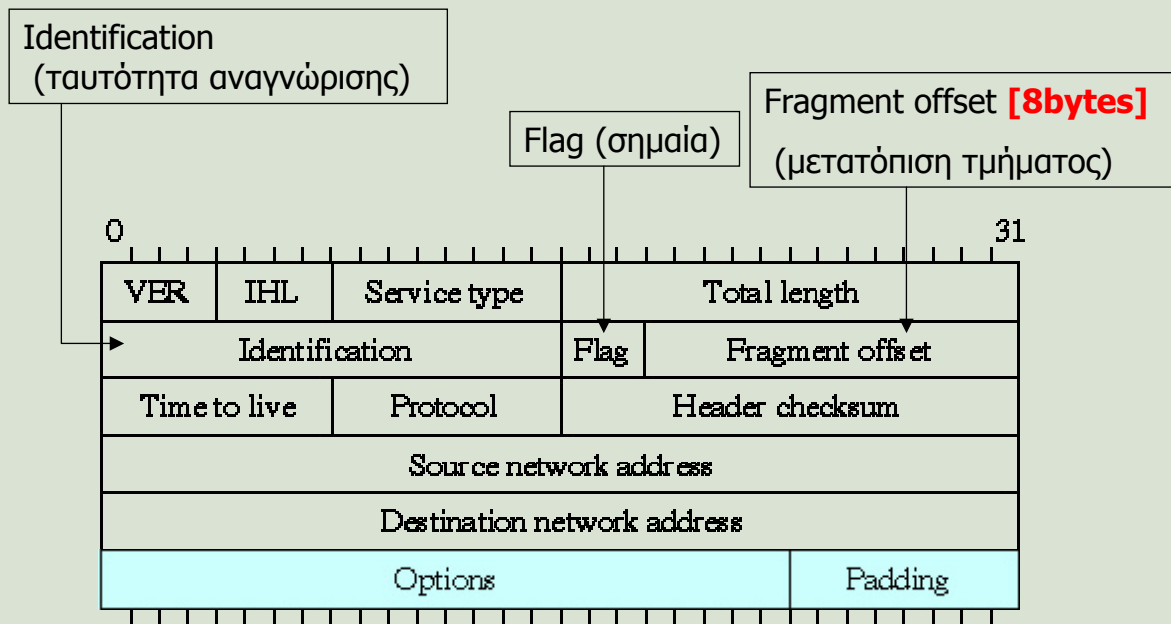
## Επικεφαλίδα IP



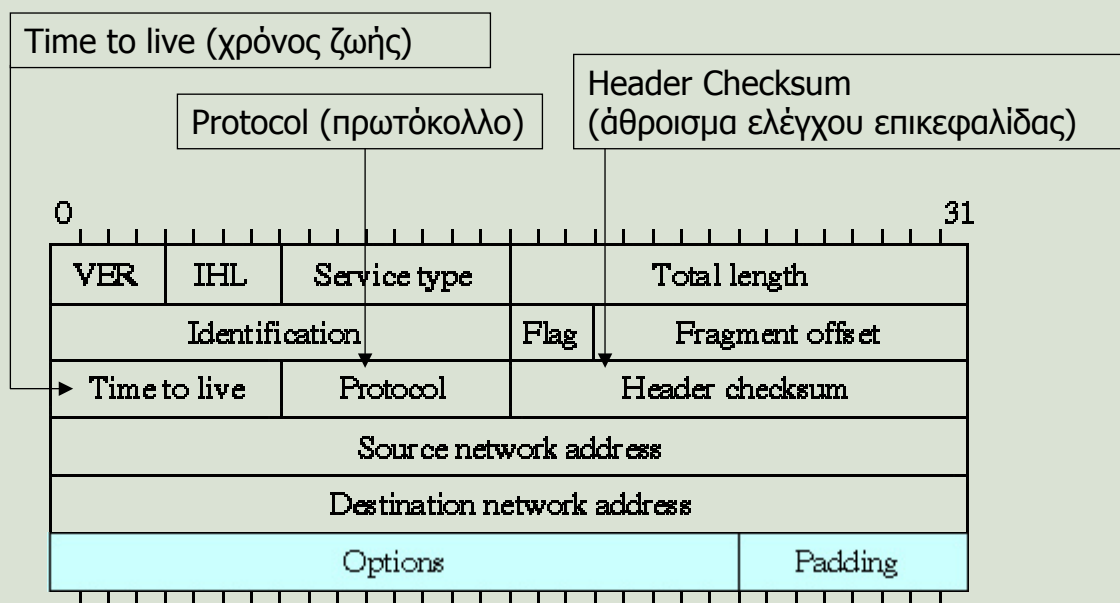
## Επικεφαλίδα IP



## Επικεφαλίδα IP



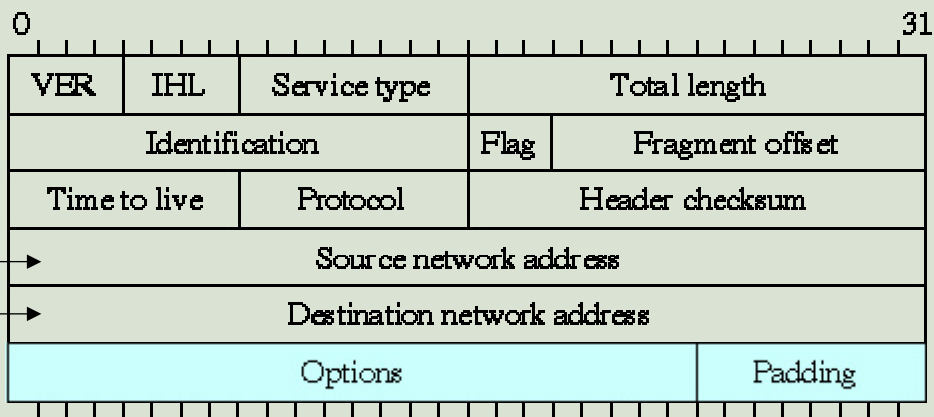
## Επικεφαλίδα IP



## Επικεφαλίδα IP

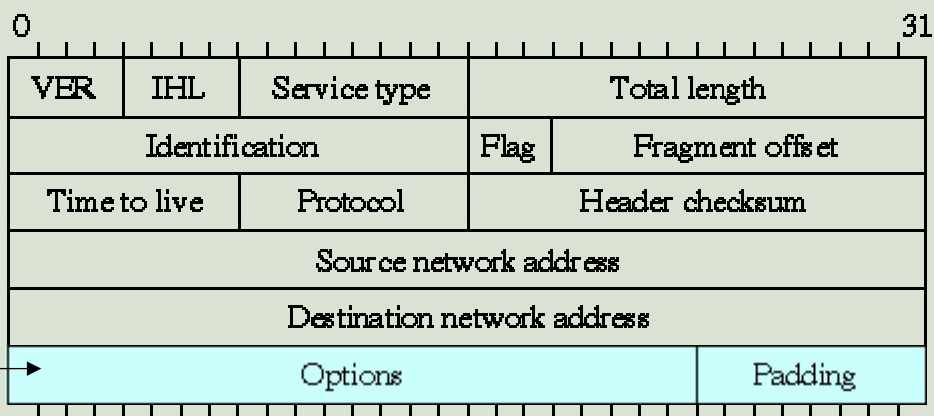
Source network address  
(διεύθυνση δικτύου πηγής)

Destination network address  
(διεύθυνση δικτύου προορισμού)



## Επικεφαλίδα IP

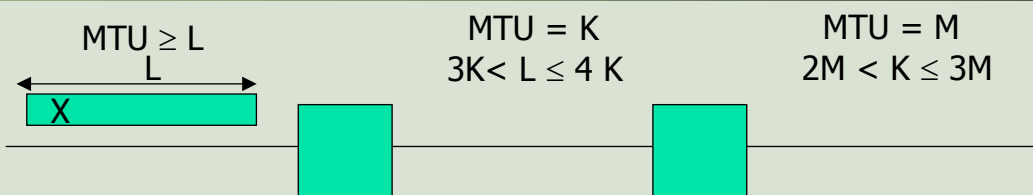
Options (επιλογές)



## Κατάτμιση

- Οι ζεύξεις χαρακτηρίζονται από ένα μέγιστο μέγεθος πακέτου που μεταδίδουν  
→ μέγιστη μονάδα μεταφοράς (Maximum Transfer Unit - MTU)
- Αν μέγεθος πακέτου > MTU της εξερχόμενης ζεύξης  
ο δρομολογητής τεμαχίζει το πακέτο σε τμήματα (fragments)
- το πακέτο επανασυντίθεται στον προορισμό
- Τα τμήματα σημειώνονται με:
  - τον αριθμό ταυτότητας αναγνώρισης (identification number) του αρχικού πακέτου
  - τη μετατόπιση (offset) του τμήματος σε σχέση με το πρώτο byte του πακέτου

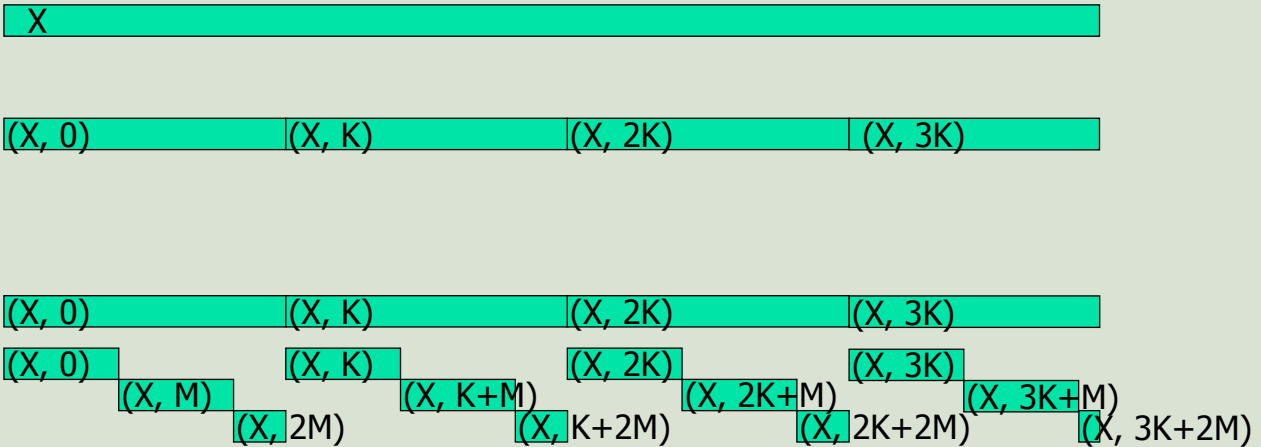
## Κατάτμιση



- Έστω πακέτο με αριθμό ταυτότητας αναγνώρισης  $X$  και μήκος  $L$
- Ο 1ος δρομολογητής τεμαχίζει το πακέτο σε τέσσερα κομμάτια με μήκη  $K$   $K$   $K$   $K'$ ,  $K' = L - 3K \leq K$  που τα σημειώνει ως  $(X, 0)$   $(X, K)$   $(X, 2K)$   $(X, 3K)$
- Ο 2ος δρομολογητής τεμαχίζει το κάθε τμήμα σε τρία τμήματα (υπολογίζει τις μετατοπίσεις των επιμέρους τμημάτων με βάση την αρχική μετατόπιση του κάθε τμήματος και τη νέα παράμετρο MTU)
- Π.χ. το τρίτο τμήμα  $(X, 2K)$  τεμαχίζεται σε τρία τμήματα με μήκη  $M$   $M$   $M'$  που σημειώνονται ως  $(X, 2K)$   $(X, 2K + M)$   $(X, 2K + 2M)$



# Κατάτμιση



- Το συγκεκριμένο παράδειγμα δεν είναι ακριβές (επικεφαλίδες, τμήμα= $n \cdot 8$ bytes)

# Κατάτμιση

Παράδειγμα από RFC 791

Data = 452 bytes

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 472 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=0| Fragment Offset = 0 |
| Time = 123 | Protocol = 6 | header checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|

```

MTU = 280 bytes

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 276 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=1| Fragment Offset = 0 |
| Time = 119 | Protocol = 6 | Header Checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|

```

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 216 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=0| Fragment Offset = 32 |
| Time = 119 | Protocol = 6 | Header Checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|

```

# Κατάτμηση

Παράδειγμα από RFC 791

IPv4

Data = 452 bytes

452+20

Μήκος επικεφαλίδας 5\* 4bytes

```

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 472 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=0| Fragment Offset = 0 |
| Time = 123 | Protocol = 6 | header checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
    
```

3ο bit=0 ⇒ Δεν ακολουθούν άλλα τμήματα

Θα πρέπει 2ο bit=0 ⇒ επιτρέπεται η κατάτμηση

MTU = 280 bytes

```

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 276 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=1| Fragment Offset = 0 |
| Time = 119 | Protocol = 6 | Header Checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
    
```

```

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 216 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=0| Fragment Offset = 32 |
| Time = 119 | Protocol = 6 | Header Checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
    
```

# Κατάτμηση

Παράδειγμα από RFC 791

Data = 452 bytes

Το τελευταίο τμήμα διατηρεί του αρχικού

```

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 472 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=0| Fragment Offset = 0 |
| Time = 123 | Protocol = 6 | header checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
    
```

$8 \text{ bytes} * k + 20 \text{ bytes} \leq 280 \text{ bytes}$   
 $\Rightarrow k = 32 \Rightarrow 1\text{o τμήμα} \rightarrow 256 \text{ bytes data}$   
 $2\text{o τμήμα} 196 \text{ bytes data}$

MTU = 280 bytes

```

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 276 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=1| Fragment Offset = 0 |
| Time = 119 | Protocol = 6 | Header Checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
    
```

```

0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 216 |
|-----|-----|-----|-----|
| Identification = 111 |Flg=0| Fragment Offset = 32 |
| Time = 119 | Protocol = 6 | Header Checksum |
|-----|-----|-----|-----|
| source address |
|-----|-----|-----|-----|
| destination address |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
| data |
|-----|-----|-----|-----|
    
```

Παράδειγμα από RFC 791

Data = 452 bytes

Υπολογίζονται ξανά

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 472 |
+-----+-----+-----+-----+
| Identification = 111 |Flg=0| Fragment Offset = 0 |
+-----+-----+-----+-----+
| Time = 123 | Protocol = 6 | header checksum |
+-----+-----+-----+-----+
| source address |
+-----+-----+-----+-----+
| destination address |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+

```

MTU = 280 bytes

```

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 276 |
+-----+-----+-----+-----+
| Identification = 111 |Flg=1| Fragment Offset = 0 |
+-----+-----+-----+-----+
| Time = 119 | Protocol = 6 | Header Checksum |
+-----+-----+-----+-----+
| source address |
+-----+-----+-----+-----+
| destination address |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+

0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|Ver= 4 |IHL= 5 |Type of Service| Total Length = 216 |
+-----+-----+-----+-----+
| Identification = 111 |Flg=0| Fragment Offset = 32 |
+-----+-----+-----+-----+
| Time = 119 | Protocol = 6 | Header Checksum |
+-----+-----+-----+-----+
| source address |
+-----+-----+-----+-----+
| destination address |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+
| data |
+-----+-----+-----+-----+

```

8 bytes \* k + 20 bytes ≤ 280 bytes  
 ⇒ k = 32 ⇒ 1ο τμήμα → 256 bytes data  
 2ο τμήμα 196 bytes data

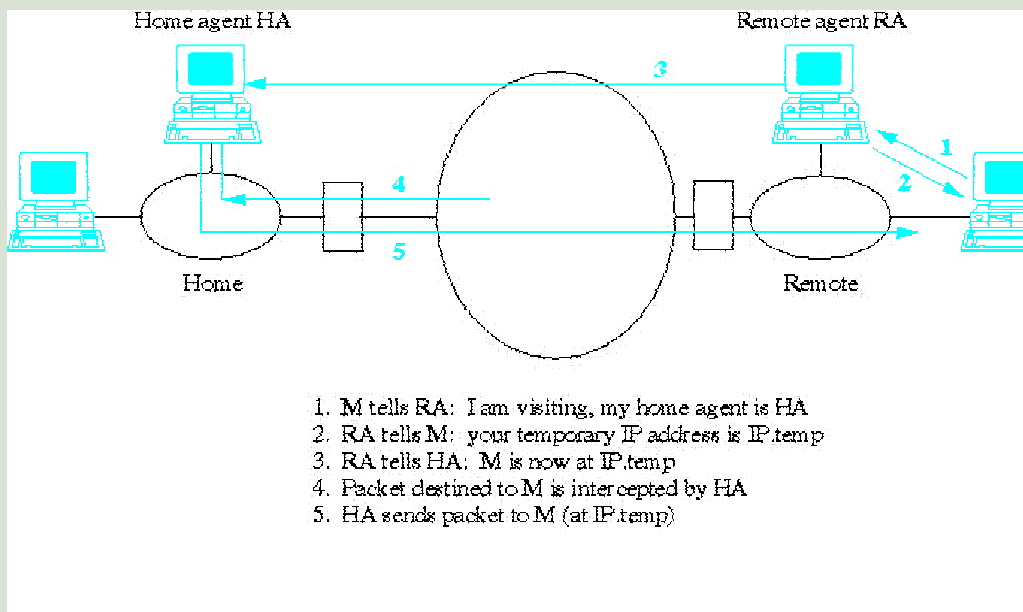
# Dynamic Host Configuration Protocol (DHCP)

(πρωτόκολλο δυναμικής διαμόρφωσης κόμβου υποδοχής)

- δυναμική εκχώρηση διευθύνσεων
  - πιο αποδοτική χρήση (δέσμευση για όσο χρειάζεται)
  - κινητός υπολογιστής που αλλάζει δίκτυα
- εκχώρηση διεύθυνσης με χρόνο ζωής

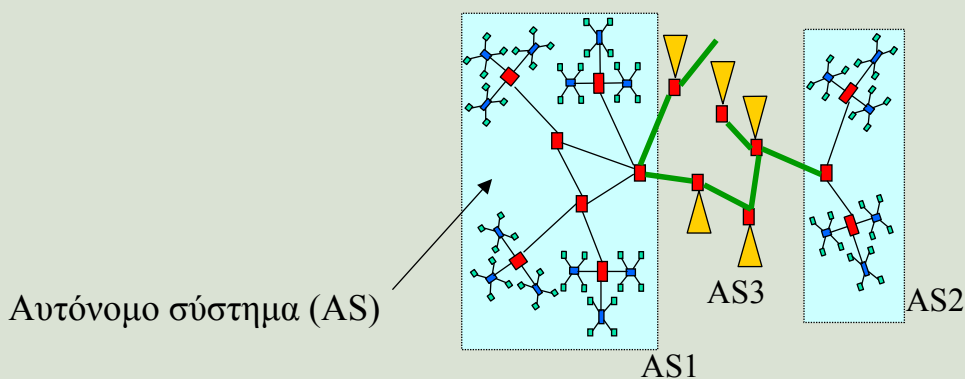
Επιτρέπει την ανεύρεση ενός υπολογιστή που βρίσκεται (προσωρινά) σε ξένο δίκτυο

- Το πρωτόκολλο χρησιμοποιεί: DHCP + πράκτορες (agents)



## Ιεραρχική δρομολόγηση

- Κατατάσσει κόμβους σε αυτόνομα συστήματα (Autonomous Systems - ASs) (AS: υποδίκτυο που βρίσκεται υπό τον έλεγχο ενός μόνο οργανισμού)
- Αυτόνομα συστήματα συνδέονται με συννοριακές πύλες (border gateways)
- Εσωτερικά στο AS: OSFP (Open Shortest Path First) (αλγόριθμο Dijkstra) [«αντικατέστησε» το Routing Information Protocol - RIP (Bellman-Ford)]
- Μεταξύ ASs: BGP (Border Gateway Protocol)



## Open Shortest Path First (OSPF)

Κάθε δρομολογητής κατασκευάζει ένα «χάρτη» του δικτύου βασιζόμενος σε μηνύματα που λαμβάνει και υπολογίζει τα ελάχιστου κόστους μονοπάτια με χρήση του αλγόριθμου του Dijkstra

- ο δρομολογητής υπολογίζει μια (ή περισσότερες) μετρική για κάθε εξερχόμενη ζεύξη
  - η μετρική βασίζεται π.χ. στο ρυθμό μετάδοσης, τη μέση καθυστέρηση
- Ο δρομολογητής  $i$  προετοιμάζει ένα μήνυμα της μορφής

$$M = [ i | s | k_1, d_1 | k_2, d_2 | \dots | k_m, d_m ]$$

όπου:

$s$  : αριθμός ακολουθίας μηνύματος

$k_1, k_2, \dots, k_m$  : γείτονες

$d_1, d_2, \dots, d_m$  : αντίστοιχες μετρικές

- Ο δρομολογητής  $i$  στέλνει το μήνυμα  $M$  σε κάθε εξερχόμενη ζεύξη

## Open Shortest Path First (OSPF)

Όταν ο δρομολογητής  $j$  λάβει ένα τέτοιο μήνυμα :

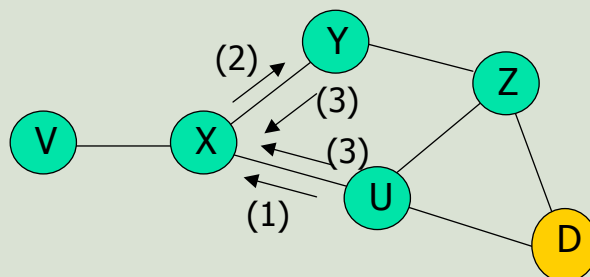
- Ελέγχει αν  $i = j$ 
  - Αν  $i = j$  το πακέτο απορρίπτεται
- Ελέγχει αν  $s >$  μέγιστος αριθμός ακολουθίας μηνύματος που έχει λάβει από τον  $i$
- Αν  $s$  μεγαλύτερο  $\Rightarrow$   $M$  νέο μήνυμα
  - Ανανέωση μέγιστου αριθμού ακολουθίας μηνύματος που έχει λάβει από τον  $i$
  - Αποθήκευση μηνύματος
  - Αποστολή σε όλες τις ζεύξεις (εκτός από εκεί που ήρθε)  
[πλημμύρα (flooding)]

## Border Gateway protocol (BGP)

- Χρησιμοποιείται για την δρομολόγηση ανάμεσα σε Αυτόνομα Συστήματα
- Κατανεμημένο πρωτόκολλο (διαφορετικοί δρομολογητές έχουν, ενδεχομένως, διαφορετική πληροφορία για το δίκτυο)
- Κάθε AS ενημερώνει για προτιμητέες διαδρομές
- Μετάδοση πληροφορίας για ολόκληρο το μονοπάτι
  - για την αποφυγή βρόχων
  - επιβολή πολιτικών ως προς άλλα αυτόνομα συστήματα
- οι «εσωτερικές» μετρικές για την ενημέρωση του BGP δεν είναι απαραίτητο να είναι ίδιες, αρκεί να είναι συνεπείς

## Border Gateway protocol (BGP)

- Ασυνέπειες λόγω παλαιώσης πληροφορίας



(1) το U γνωστοποιεί μία μετρική = 12 για τον προορισμό D

(2) το X διαδίδει αυτή την εκτίμηση στο Y

(3)

- το Y υπολογίζει μια μετρική = **17 για τον D (μέσω του X)**, τη στέλνει στο X
  - το U έχει μία νέα εκτίμηση της μετρική προς τον **D (=18)**, τη στέλνει στο X
- Αν δε στέλνονταν τα μονοπάτια δεν αποκλείεται το X να επιλέξει το Y ως ενδιάμεσο προς το D (αν  $L(X,Y) \leq L(X,U)$ )  $\Rightarrow$  βρόχος

## Αλγόριθμος Dijkstra

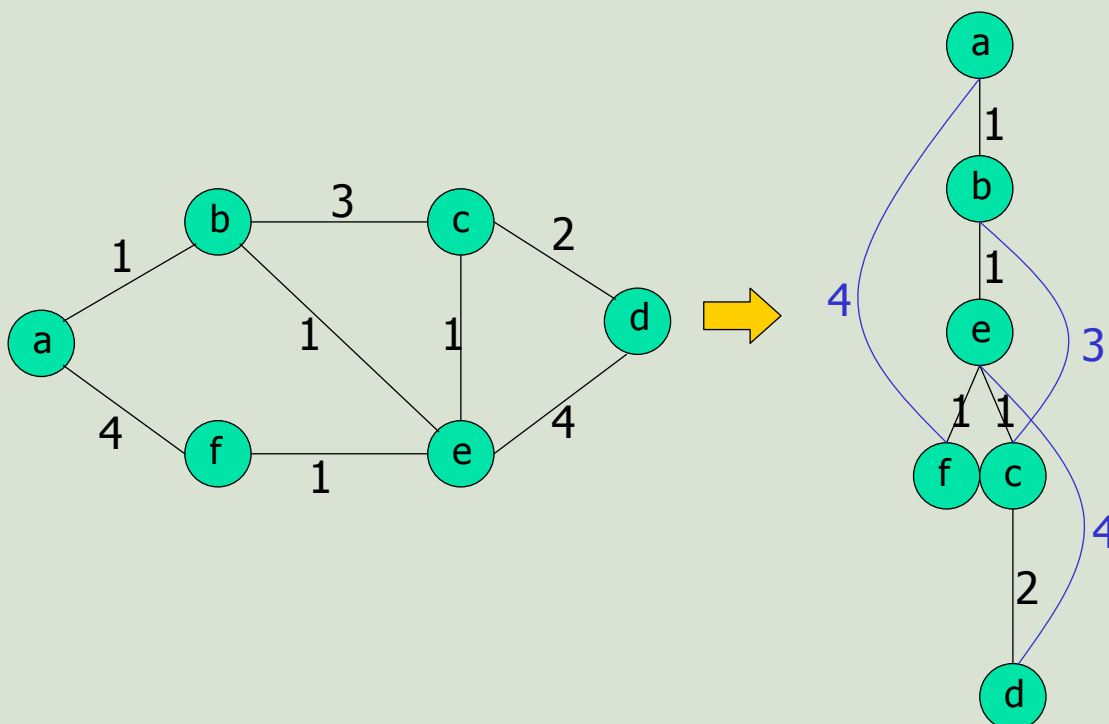
- Τοπολογία δικτύου και κόστη συνδέσεων θεωρούνται γνωστά
  - Κόστος: καθυστέρηση, επίπεδο συμφόρησης κτλ.
- Σκοπός ο υπολογισμός μονοπατιών ελάχιστου κόστους από την πηγή στον προορισμό (έτσι προκύπτει ο πίνακας δρομολόγησης)
- Το δίκτυο μοντελοποιείται σαν ένας γράφος όπου οι δρομολογητές παρίστανται σαν κόμβοι και οι ζεύξεις σαν ακμές

Έστω:

- **S** σύνολο από N κόμβους
- $\{L(i,j), i, j \in \mathbf{S}\}$  μήκη ζεύξεων μεταξύ των κόμβων ( $L(i,j)=\infty$  αν δεν υπάρχει ζεύξη)

Εύρεση συντομότερου μονοπατιού από τον κόμβο  $a \in \mathbf{S}$  προς κάθε άλλο κόμβο

## Αλγόριθμος Dijkstra - Φυσικό Μοντέλο



## Αλγόριθμος Dijkstra

- **U(n)**: το σύνολο των κόμβων για τους οποίους το ελαχίστου μήκους μονοπάτι έχει καθοριστεί στο βήμα n  
(σύνολο από μπάλες οι οποίες έχουν σηκωθεί από το πάτωμα στο βήμα n)
- **F(n)**: το σύνολο των κόμβων για τους οποίους το ελαχίστου μήκους μονοπάτι δεν έχει καθοριστεί στο βήμα n  
(σύνολο από μπάλες που εξακολουθούν να βρίσκονται στο πάτωμα στο βήμα n)  
(  $U(n) \cup F(n) = S$  )
- **b<sub>n</sub>**: κόμβος για τον οποίο το ελαχίστου μήκους μονοπάτι καθορίζεται στο βήμα n  
(μπάλα σηκώνεται από το πάτωμα στο βήμα n)
- **N(n)**: σύνολο των γειτονικών ως προς το **b<sub>n</sub>** κόμβων για τους οποίους το ελαχίστου μήκους μονοπάτι δεν έχει καθοριστεί στο βήμα n  
(σύνολο από γειτονικές ως προς τη **b<sub>n</sub>** μπάλες που βρίσκονται ακόμα στο πάτωμα)
- **dn(i)**: μήκος του μονοπατιού ελαχίστου μήκους, από τον κόμβο a στον i στο βήμα n
- (P<sub>n</sub>(i) , i): τελευταία ζεύξη του μονοπατιού μήκους dn(i). Ο κόμβος **Pn(i)** είναι ο κόμβος πρόσβασης στον i του μονοπατιού ελάχιστου μήκους dn(i).

## Αλγόριθμος Dijkstra

$n=1$

$b_1 = a, d_1(a) = 0,$

$U(1) = \{a\}, F(1) = S - U(1),$

Αρχικοποίηση αλγορίθμου

$d_1(i) = \infty, P_1(i) = \emptyset \quad \forall i \in F(1)$

$N(n) = \{j \in F(n) \mid L(b_n, j) < \infty\}$  (γειτονικοί κόμβοι εκτός U)

$d_{n+1}(i) = \min\{d_n(i), d_n(b_n) + L(b_n, i)\} \quad \forall i \in N(n)$

Ανανέωση κόστους η διατήρηση του παλαιού

$n < N$   
 $P_{n+1}(i) = \begin{cases} P_n(i) & \text{εάν } d_n(i) \leq d_n(b_n) + L(b_n, i), \\ \{b_n\} & \text{εάν } d_n(i) > d_n(b_n) + L(b_n, i). \end{cases}$

Ανανέωση (η μη) κόμβου πρόσβασης του μονοπατιού ελάχ. Κόστους

$b_{n+1} = \arg \min\{d_{n+1}(i), i \in F(n)\}$  (επιλογή κόμβου με μονοπάτι ελαχ κόστους από ΟΛΟΥΣ εκτός U)

$U(n+1) = U(n) \cup \{b_{n+1}\}$  και  $F(n+1) = S - U(n+1)$  (τον βάζουμε μέσα στον U)

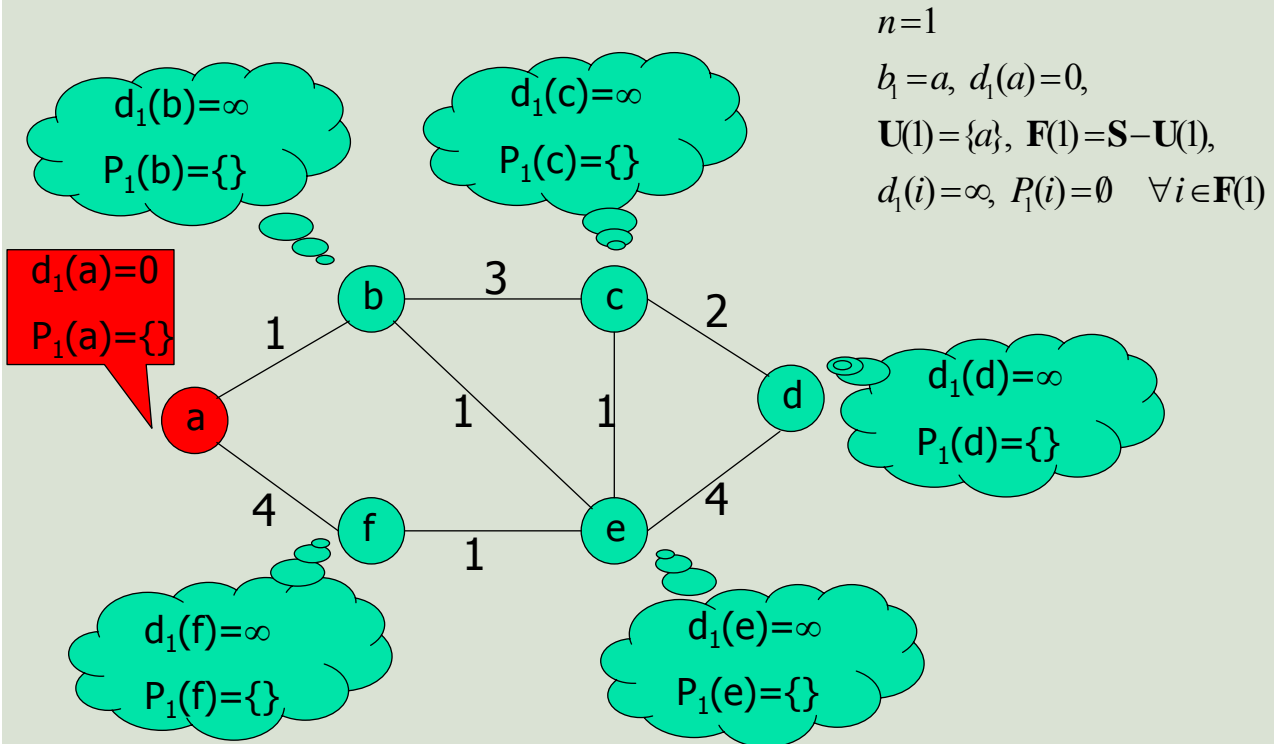
$n = n + 1$

$d_{n+1}(i) = d_n(i),$   
 $P_{n+1}(i) = P_n(i) \quad \forall i \in S - N(n)$

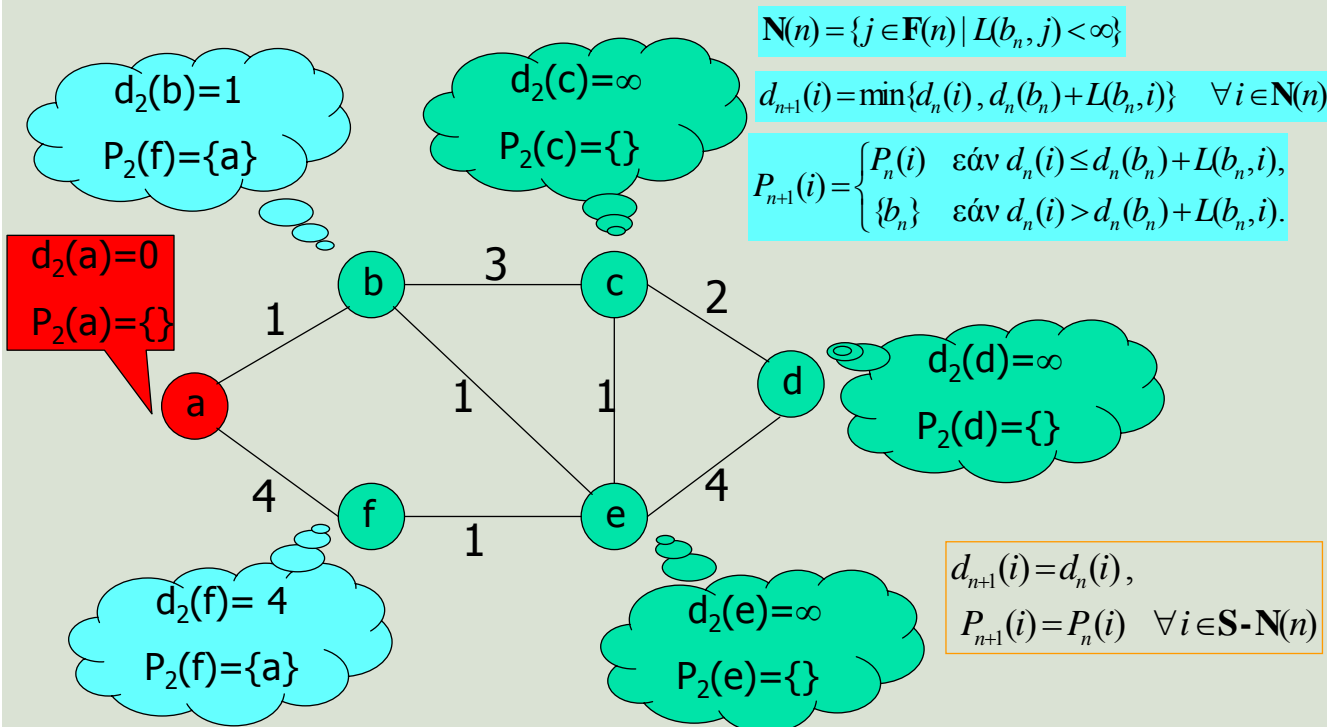
Ανανέωση εκτιμήσεων των κόμβων εκτός U και μη γειτόνων



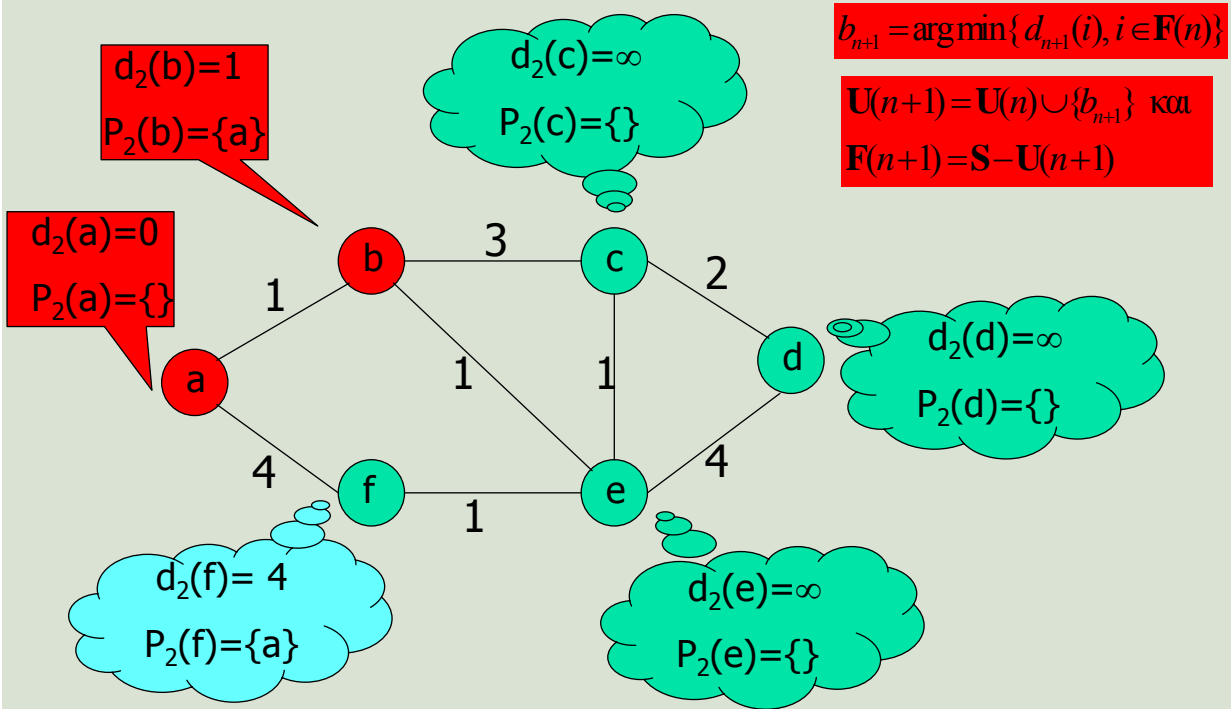
# Αρχικοποίηση (n=1)



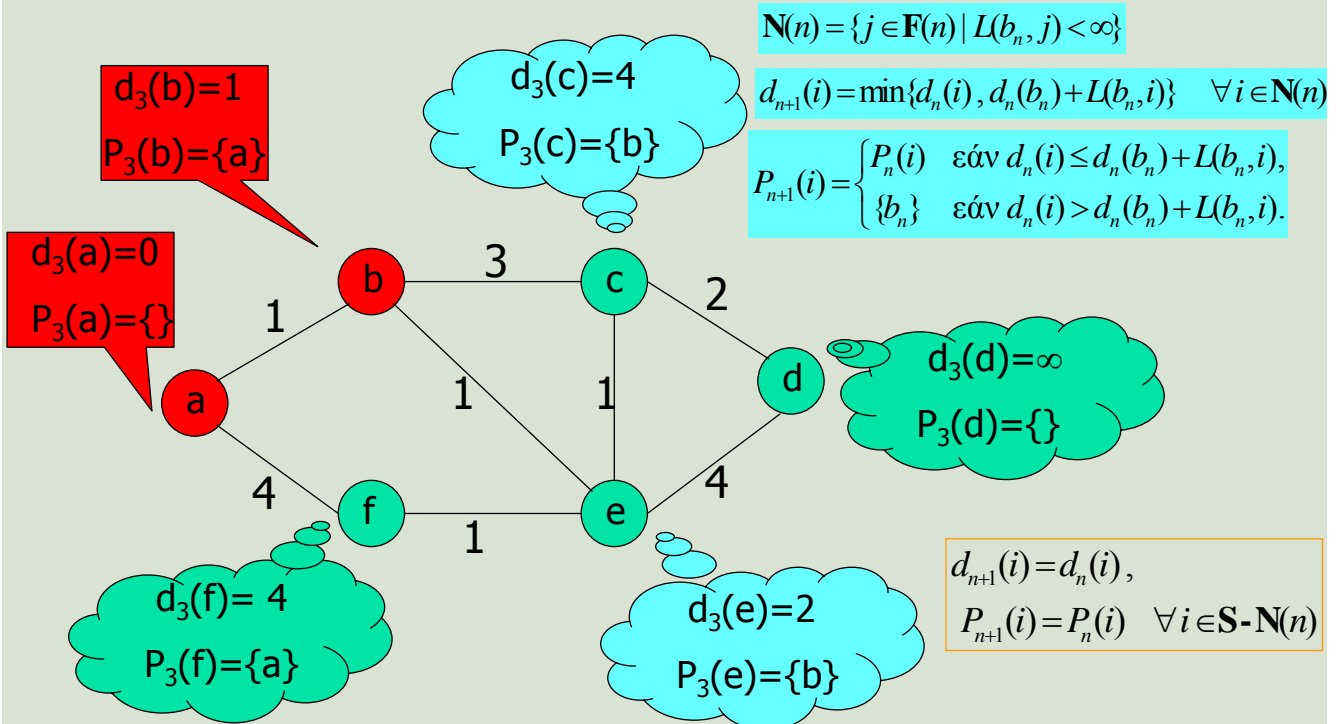
# (n=1)



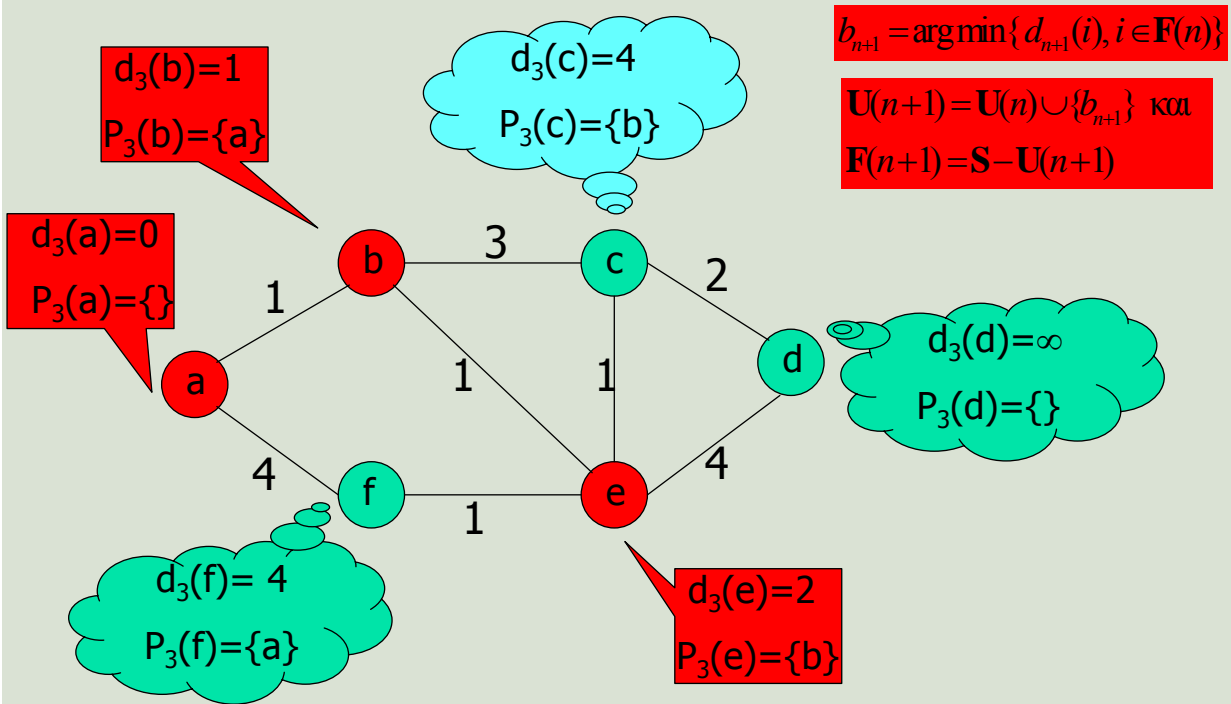
(n=1)



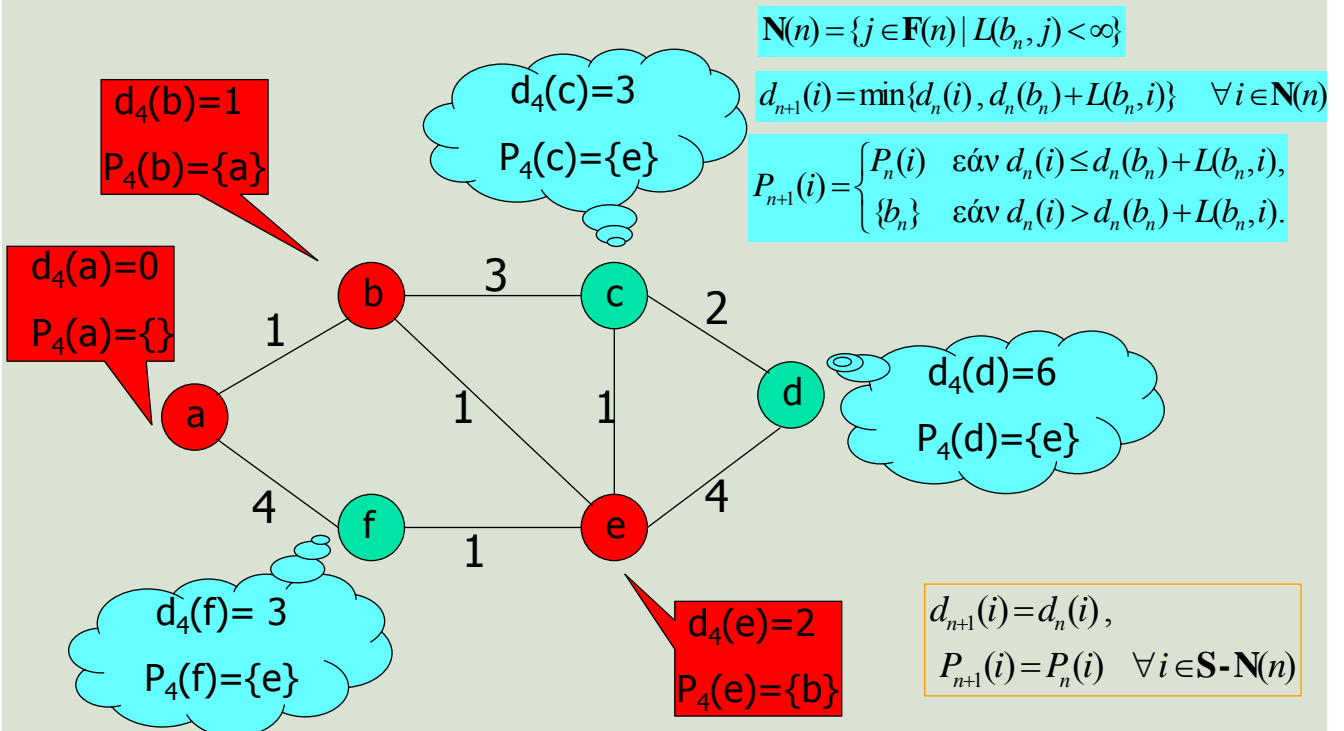
(n=2)



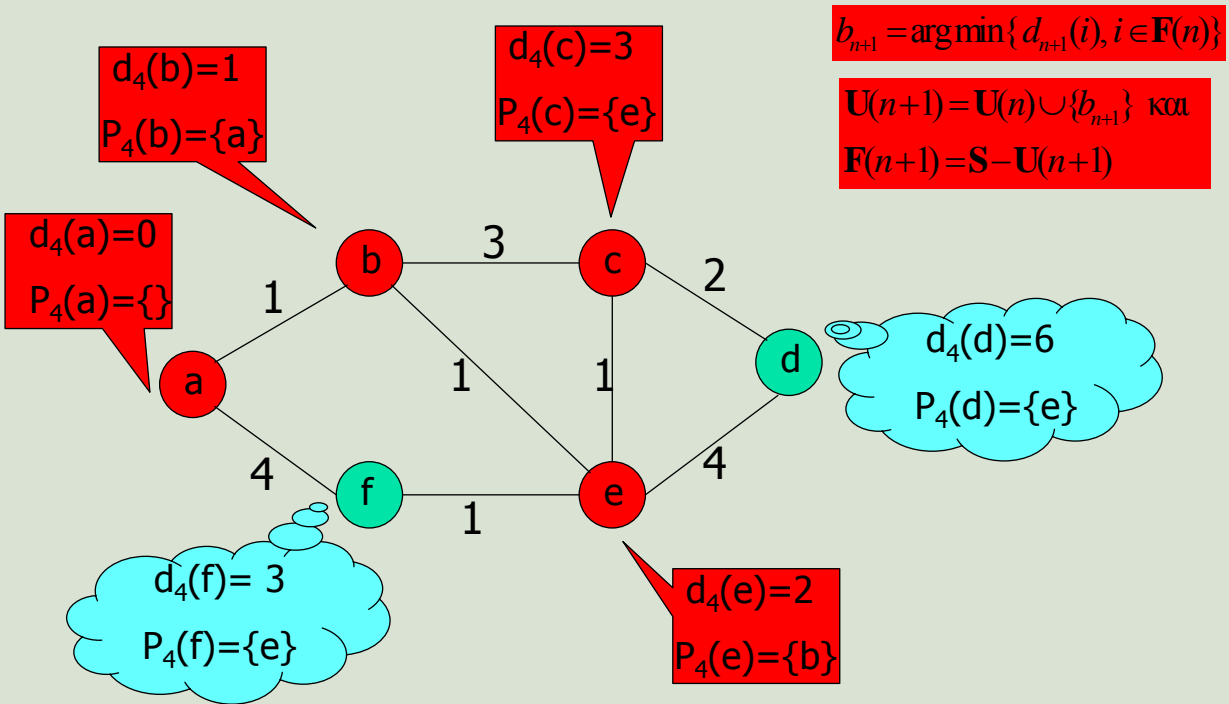
(n=2)



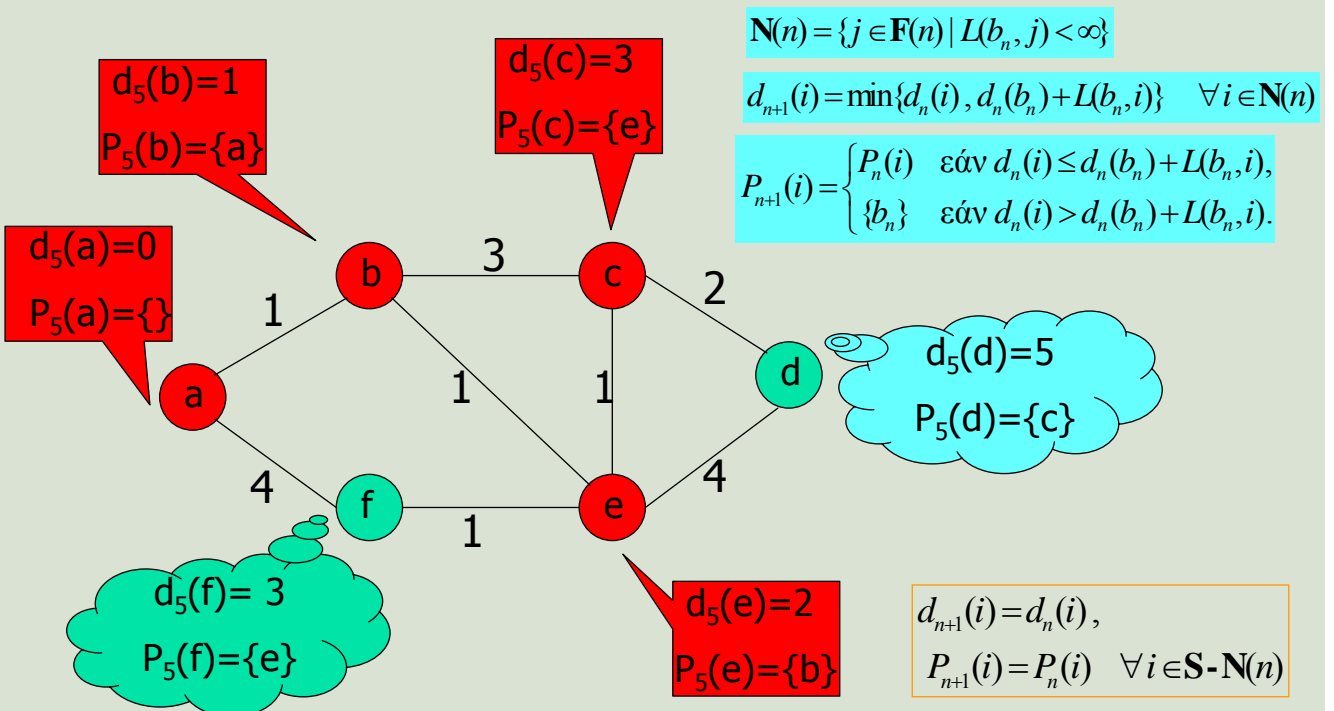
(n=3)



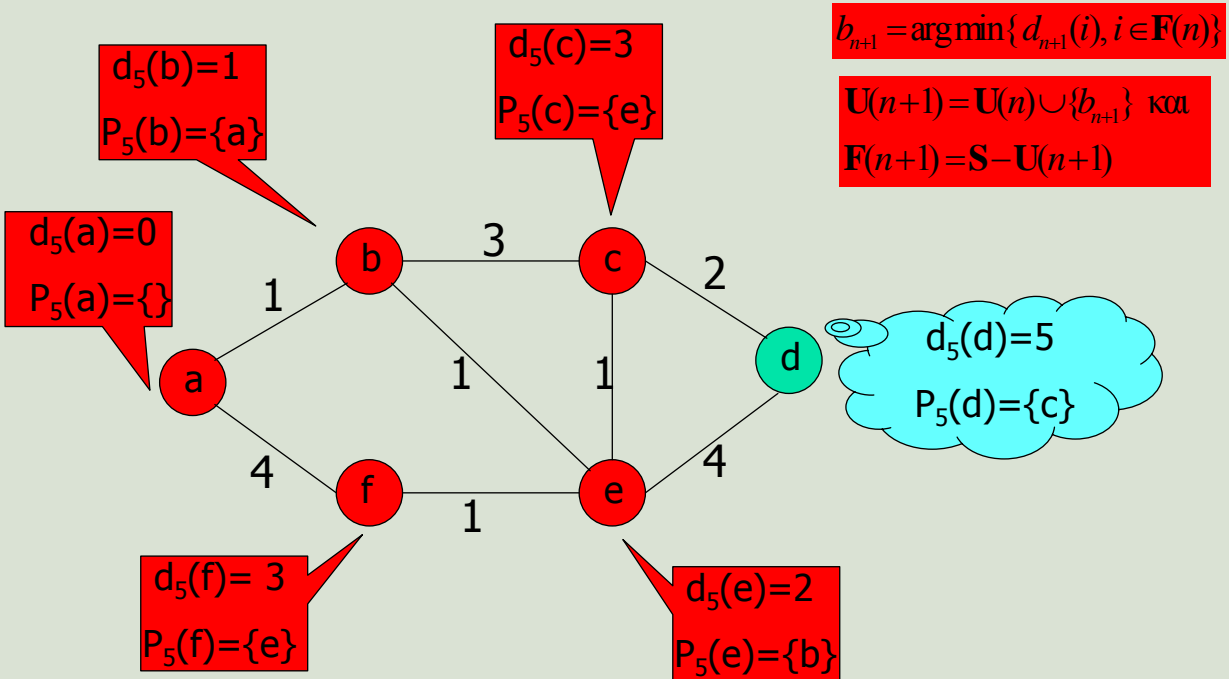
(n=3)



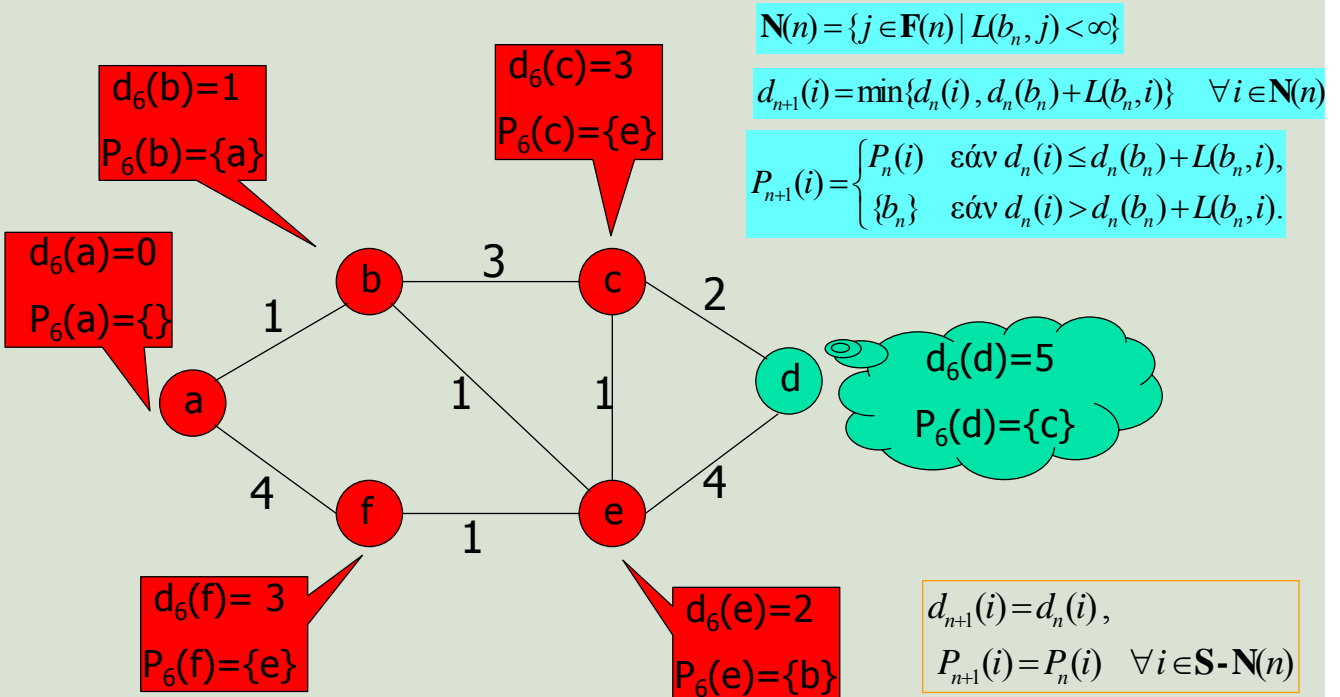
(n=4)



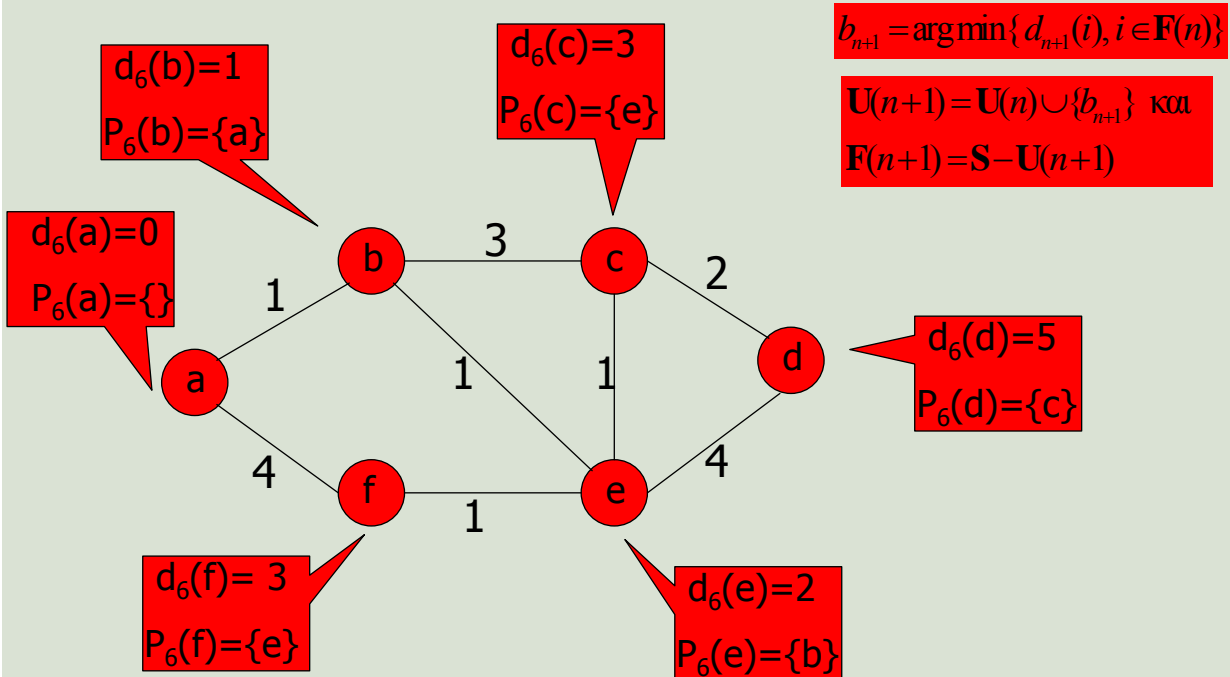
(n=4)



(n=5)



(n=5)



## Αλγόριθμος Dijkstra

### ο αλγόριθμος παράγει τα συντομότερα μονοπάτια:

- στη n-οστή επανάληψη έχει βρει τα συντομότερα μονοπάτια από τον a στους κόμβους του  $U(n)$  και
- τα μονοπάτια αυτά έχουν μήκη  $d_n(i)$  όπου  $i \in U(n)$

Απόδειξη με επαγωγή:

- πρόταση αληθής για  $n = 1$  (συντομότερο μονοπάτι από τον a στον a με μήκος 0)
- Έστω ότι η πρόταση είναι αληθής για την n-οστή επανάληψη
- θ.δ.ο. είναι αληθής και για την επανάληψη  $n + 1$ 
  - απαγωγή σε άτοπο

## Αλγόριθμος Dijkstra

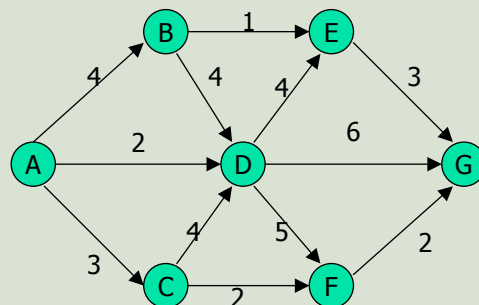
- Έστω ότι υπάρχει ένα μονοπάτι από τον  $a$  στον  $f$  ( $:= b_{n+1}$ ) με μήκος αυστηρά μικρότερο από  $d_{n+1}(f)$
- Έστω  $j$  ο τελευταίος κόμβος στο  $U(n)$  κατά μήκος του μονοπατιού αυτού, όπου  $j$  μπήκε στο  $U$  σε κάποιο προηγούμενο βήμα  $m \leq n$  (δηλ.  $b_m = j$ )
- Έστω  $k$  ο επόμενος (του  $j$ ) κόμβος κατά μήκος του μονοπατιού αυτού από τον  $a$  στον  $b_{n+1}$  (αφού  $j$  ο τελευταίος στο  $U(n)$ )  $\rightarrow$   $k$  είναι εκτός  $U(n)$ )
- Ισχύει  $d_m(j) + L(j, k) = d_n(j) + L(j, k) = d_{n+1}(k) \geq d_{n+1}(f)$

[αν δεν ίσχυε η ανισότητα τότε ο  $k$  θα είχε επιλεγεί πριν από τον  $f$  για να περιληφθεί στο  $U(n)$ ]

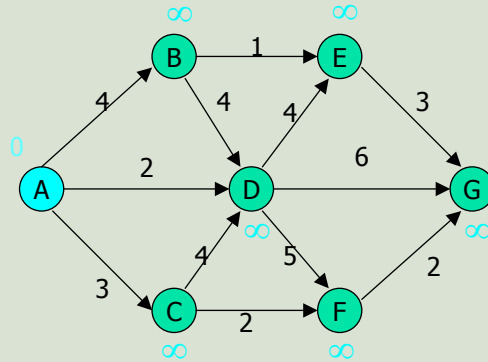
άτοπο

## Παράδειγμα

Εύρεση του συντομότερου μονοπατιού από τον κόμβο  $A$  προς κάθε άλλο κόμβο

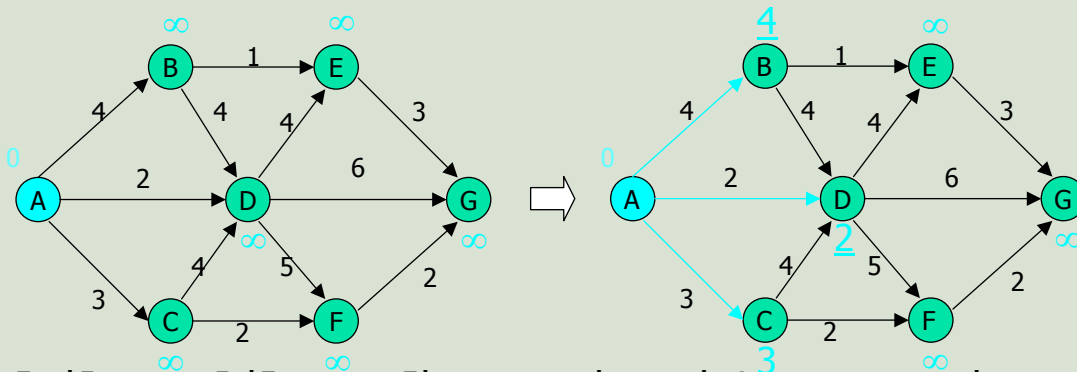


## Παράδειγμα



- Ο αλγόριθμος ξεκινά από τον κόμβο A (πηγή)
  - δίπλα σε κάθε κόμβο σημειώνεται μία εκτιμώμενη απόσταση από τον A προς τον κόμβο ( $= \infty$ , εκτός από αυτήν του κόμβου A ( $= 0$ ))
  - Ο κόμβος A σκιάζεται  $\Leftrightarrow$  έχει την ελάχιστη εκτιμώμενη απόσταση από την πηγή (κόμβος A)  $\Leftrightarrow$  το ελαχίστου μήκους μονοπάτι προς αυτόν έχει καθορισθεί ( $\Leftrightarrow$  έχει σηκωθεί από το πάτωμα)

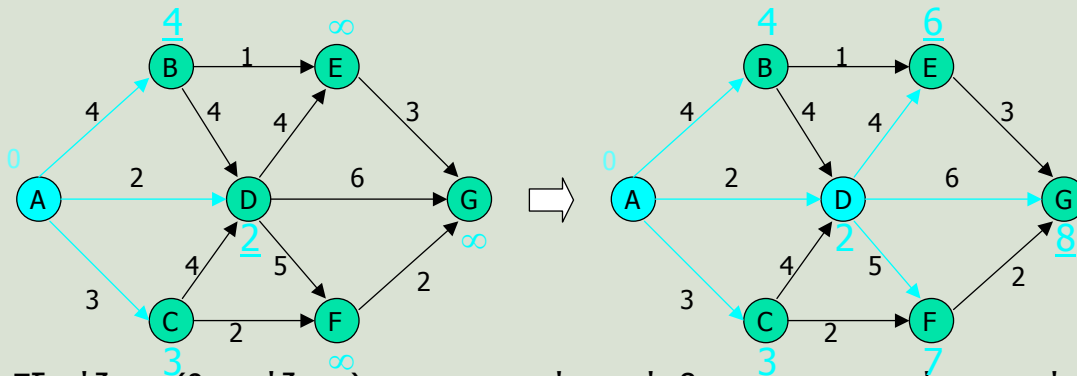
## Παράδειγμα



- εξετάζονται οι ζεύξεις που εξέρχονται από τον κόμβο A και ενημερώνονται οι εκτιμήσεις των κόμβων που μετέχουν στις ζεύξεις (νέες εκτιμήσεις  $\rightarrow$     )
- σημειώνονται οι ζεύξεις που οδηγούν σε μείωση των εκτιμήσεων ως παρούσες υποψήφιες για το συντομότερο μονοπάτι προς τους αντίστοιχους κόμβους (ενδεχομένως «ακυρώνονται» κάποιες προηγούμενα υποψήφιες ζεύξεις)

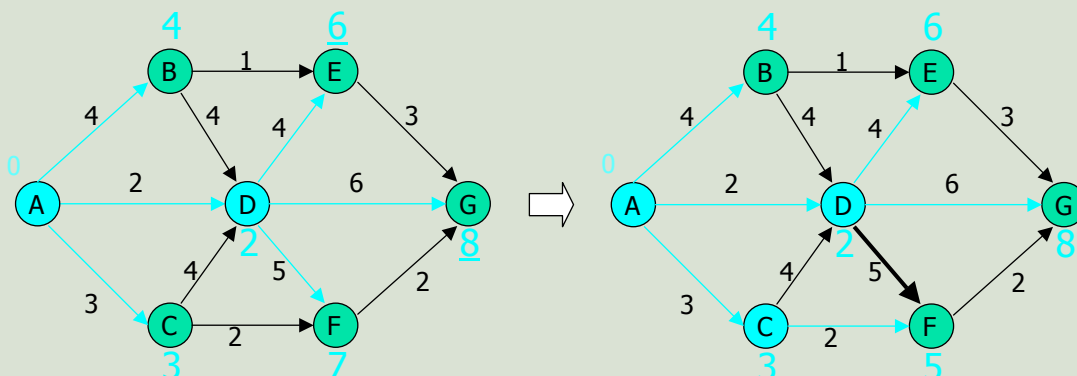


## Παράδειγμα



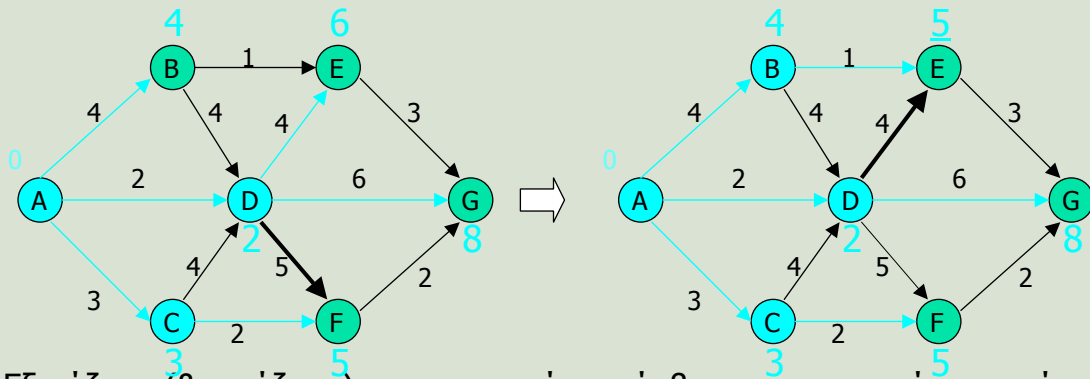
- Εξετάζεται (& σκιάζεται) ο μη-σκιασμένος κόμβος με την παρούσα μικρότερη εκτίμηση (κόμβος D)
  - εξετάζονται οι ζεύξεις που εξέρχονται από τον κόμβο αυτό και ενημερώνονται οι εκτιμήσεις των κόμβων που μετέχουν στις ζεύξεις (νέες εκτιμήσεις →\_\_)
  - σημειώνονται οι ζεύξεις που οδηγούν σε μείωση των εκτιμήσεων ως παρούσες υποψήφιες για το συντομότερο μονοπάτι προς τους αντ. κόμβους (ενδεχομένως «ακυρώνονται» κάποιες προηγούμενα υποψήφιες ζεύξεις)

## Παράδειγμα



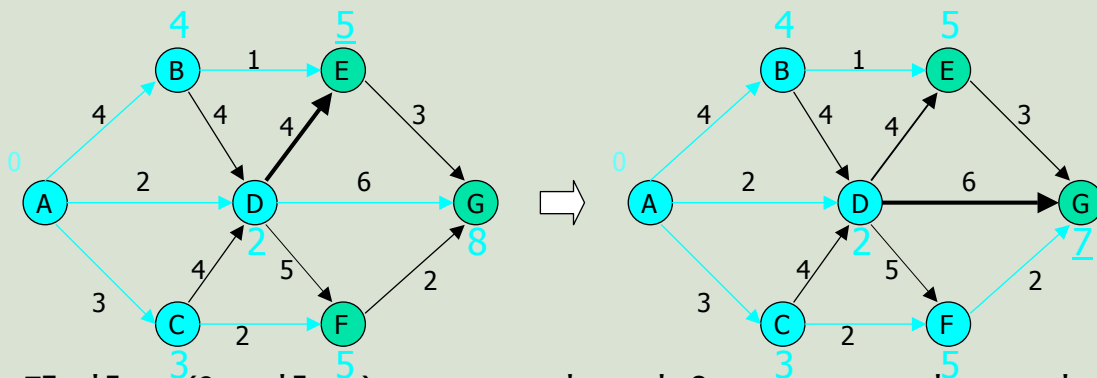
- Εξετάζεται (& σκιάζεται) ο μη-σκιασμένος κόμβος με την παρούσα μικρότερη εκτίμηση (κόμβος C)
  - εξετάζονται οι ζεύξεις που εξέρχονται από τον κόμβο αυτό και ενημερώνονται οι εκτιμήσεις των κόμβων που μετέχουν στις ζεύξεις (νέες εκτιμήσεις →\_\_)
  - σημειώνονται οι ζεύξεις που οδηγούν σε μείωση των εκτιμήσεων ως παρούσες υποψήφιες για το συντομότερο μονοπάτι προς τους αντ. κόμβους («ακυρώνεται» η ζεύξη DF, η CF οδηγεί σε μείωση της εκτίμησης)

## Παράδειγμα



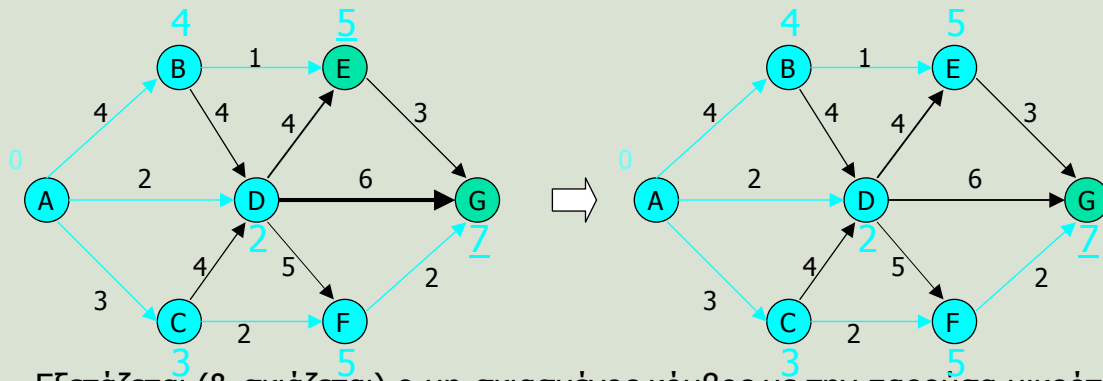
- Εξετάζεται (& σκιάζεται) ο μη-σκιασμένος κόμβος με την παρούσα μικρότερη εκτίμηση (κόμβος B)
  - εξετάζονται οι ζεύξεις που εξέρχονται από τον κόμβο αυτό και ενημερώνονται οι εκτιμήσεις των κόμβων που μετέχουν στις ζεύξεις (νέες εκτιμήσεις →\_\_)
  - σημειώνονται οι ζεύξεις που οδηγούν σε μείωση των εκτιμήσεων ως παρούσες υποψήφιες για το συντομότερο μονοπάτι προς τους αντ. κόμβους («ακυρώνεται» η ζεύξη DE, η BE οδηγεί σε μείωση της εκτίμησης)

## Παράδειγμα



- Εξετάζεται (& σκιάζεται) ο μη-σκιασμένος κόμβος με την παρούσα μικρότερη εκτίμηση (κόμβος E ή κόμβος F, έστω ο F)
  - εξετάζονται οι ζεύξεις που εξέρχονται από τον κόμβο αυτό και ενημερώνονται οι εκτιμήσεις των κόμβων που μετέχουν στις ζεύξεις (νέες εκτιμήσεις →\_\_)
  - σημειώνονται οι ζεύξεις που οδηγούν σε μείωση των εκτιμήσεων ως παρούσες υποψήφιες για το συντομότερο μονοπάτι προς τους αντ. κόμβους («ακυρώνεται» η ζεύξη DG, η FG οδηγεί σε μείωση της εκτίμησης)

## Παράδειγμα



- Εξετάζεται (& σκιάζεται) ο μη-σκιασμένος κόμβος με την παρούσα μικρότερη εκτίμηση (κόμβος E)
  - εξετάζονται οι ζεύξεις που εξέρχονται από τον κόμβο αυτό και ενημερώνονται οι εκτιμήσεις των κόμβων που μετέχουν στις ζεύξεις (νέες εκτιμήσεις  $\rightarrow$ \_\_)
  - σημειώνονται οι ζεύξεις που οδηγούν σε μείωση των εκτιμήσεων ως παρούσες υποψήφιες για το συντομότερο μονοπάτι προς τους αντ. κόμβους (ενδεχομένως «ακυρώνονται» κάποιες προηγούμενα υποψήφιες ζεύξεις)

## Αλγόριθμος Bellman-Ford

- κατανημημένος αλγόριθμος: οι δρομολογητές έχουν διαφορετικές και ελλιπείς πληροφορίες για το γράφημα
- Σε κάθε βήμα ένας κόμβος
  - υπολογίζει μία εκτίμηση της ελάχιστης απόστασής του από τον προορισμό
  - γνωστοποιεί την εκτίμηση αυτή στους γείτονές του

### Περιγραφή

- **S**: σύνολο από N κόμβους
- $\{L(i, j), i, j \in S\}$  ( $L(i, j) = \infty$  εάν δεν υπάρχει ζεύξη από τον i στον j)

### Εύρεση συντομότερων μονοπατιών προς έναν προορισμό $D \in S$

- Υποθέσεις:
    - όλα τα βήματα είναι συγχρονισμένα
    - οι ζεύξεις είναι αμφίδρομες (αν  $L(i, j) < \infty$ , τότε  $L(j, i) < \infty, \forall i, j$ )
- [ο αλγόριθμος εκτελείται σωστά και χωρίς τις υποθέσεις αυτές]

## Αλγόριθμος Bellman-Ford

- Έστω  $x_n(i)$ : εκτίμηση της απόστασης του  $i$  από τον  $D$  στο βήμα  $n$
- Στο βήμα 0
  - ο κόμβος  $i$  έχει τις εκτιμήσεις  $x_0(i)=\infty \forall i \in \mathbf{S}, i \neq D$  και  $x_0(D)=0$
  - οι κόμβοι γνωστοποιούν τις εκτιμήσεις αυτές στους γείτονές τους
- Στο βήμα  $n+1$  ο κόμβος  $k \in \mathbf{S}, k \neq D$  εξετάζει τα μηνύματα τα οποία έλαβε από τους γείτονές του στο προηγούμενο βήμα  $n$  και
  - ενημερώνει την εκτίμησή του για την απόστασή του από τον  $D$  με

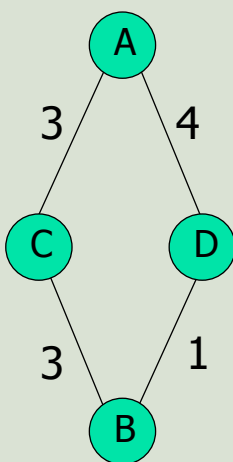
$$x_{n+1}(k) = \min \{L(i, j) + x_n(j), j \in \mathbf{S}\} \quad k \in \mathbf{S}, k \neq D$$

$N-1$  εξισώσεις που υπολογίζονται παράλληλα από  $N-1$  κόμβους

και παράγεται το επόμενο σύνολο εκτιμήσεων  $\{x_{n+1}(k), k \in \mathbf{S}, k \neq D\}$

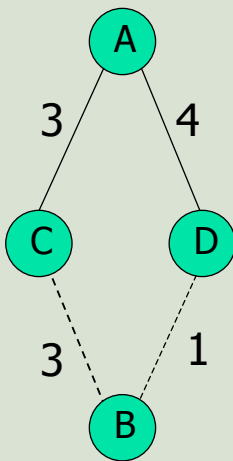
- γνωστοποιεί τις νέες εκτιμήσεις στους γείτονές του

## Παράδειγμα



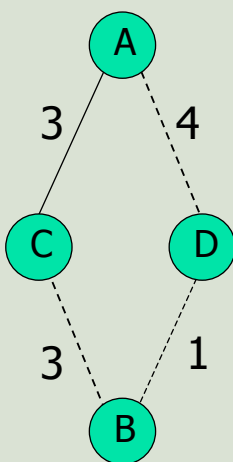
- Εύρεση συντομότερων μονοπατιών από όλους τους κόμβους προς τον κόμβο  $B$
- $x_0(A) = x_0(C) = x_0(D) = \infty$   $x_0(B) = 0$
- ο  $B$  πληροφορεί τους γείτονές του ( $C, D$ ) για την ελάχιστη απόστασή του από τον προορισμό ( $B$ )

## Παράδειγμα



- Εύρεση συντομότερων μονοπατιών από όλους τους κόμβους προς τον κόμβο B
- $x_0(A) = x_0(C) = x_0(D) = \infty$ ,  $x_0(B) = 0$
- ο B πληροφορεί τους γείτονές του (C, D) για την ελάχιστη απόστασή του από τον προορισμό (B)
- οι C και D ενημερώνουν τις εκτιμήσεις τους για την ελάχιστη απόστασή τους από τον B  
 $x_1(C) = 3$ ,  $x_1(D) = 1$
- οι C και D πληροφορούν τους γείτονές τους (A) για την ελάχιστη απόστασή τους από τον B

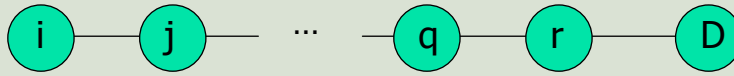
## Παράδειγμα



- Εύρεση συντομότερων μονοπατιών από όλους τους κόμβους προς τον κόμβο B
- $x_0(A) = x_0(C) = x_0(D) = \infty$ ,  $x_0(B) = 0$
- ο B πληροφορεί τους γείτονές του (C, D) για την ελάχιστη απόστασή του από τον προορισμό (B)
- οι C και D ενημερώνουν τις εκτιμήσεις τους για την ελάχιστη απόστασή τους από τον B  
 $x_1(C) = 3$ ,  $x_1(D) = 1$
- οι C και D πληροφορούν τους γείτονές τους (A) για την ελάχιστη απόστασή τους από τον B
- ο A ενημερώνει την εκτίμησή του για την ελάχιστη απόστασή του από τον B
- $x_2(A) = \min \{L(A,C) + x_1(C), L(A,D) + x_1(D)\}$   
 $= \min \{3+3, 4+1\} = 5$

## Αλγόριθμος Bellman-Ford

- ο αλγόριθμος συγκλίνει στις σωστές εκτιμήσεις σε πεπερασμένο χρόνο  
 $\exists m < \infty: \forall n \geq m, x_n(i)$  είναι η ελάχιστη απόσταση από τον  $i$  στον  $D, \forall i \in S$



- έστω κόμβος  $i \neq D$  και έστω  $n$  μονοπάτι από τον  $i$  στον  $D$  με  $n \geq 1$  ζεύξεις
- στο βήμα 1: ο  $r$  λαμβάνει μήνυμα από τον  $D$  και  $x_1(r) = L(r,D)$
- Στο βήμα 2: ο  $q$  λαμβάνει μήνυμα (τουλάχιστον) από τον  $r \Rightarrow x_2(q) \leq L(r,D) + L(q,r)$
- ...
- Στο βήμα  $n$ :  $x_n(i) =$  ελάχιστο των μηκών των μονοπατιών με έως  $n$  ζεύξεις από τον  $i$  στον  $D$
- το συντομότερο μονοπάτι από τον  $i$  στον  $D$  είναι ελεύθερο βρόχων
- $\Rightarrow$  η εκτίμηση του  $i$  παίρνει την τιμή της ελάχιστης απόστασης από τον  $i$  στον  $D$  το πολύ μετά από  $m(i)$  βήματα
- $m(i)$ : μέγιστο μήκος ενός μονοπατιού ελεύθερου βρόχων από τον  $i$  στον  $D$

$\Rightarrow$  ο αλγόριθμος συγκλίνει το πολύ σε  $m$  βήματα,  $m = \max\{m(i), i \in S\}$

## Αλγόριθμος Bellman-Ford

- ο αλγόριθμος συγκλίνει για αυθαίρετες μη αρνητικές αρχικές εκτιμήσεις  $x_0(i), i \in S$  με  $x_0(D) = 0$
- Έστω ότι ο αλγόριθμος αρχίζει με αρχικές εκτιμήσεις  $x_0(i) = 0$  για κάθε  $i \in S$
- έστω  $\{y_n(i), i \in S\}$  τις εκτιμήσεις που προκύπτουν από τις αρχικές αυτές τιμές

$$y_n(i) \leq y_{n+1}(i), i \in S, n \geq 0$$

- με επαγωγή στο  $n$  επαληθεύεται:

$$y_n(i) \leq x_n(i), i \in S, n \geq 0.$$

- $\forall i \in S$

- $x_n(i)$  συγκλίνει στην ελάχιστη απόσταση  $L(i)$  από τον  $i$  στον  $D$  σε πεπερασμένο αριθμό βημάτων  $\Rightarrow y_n(i)$  συγκλίνει
- έστω  $V(i)$  το όριο του  $y_n(i), n \rightarrow \infty$ .

$$y_{n+1}(k) = \min\{L(i, j) + y_n(j), j \in S\} \Rightarrow V(i) = \min\{L(i, j) + V(j), j \in S\}$$

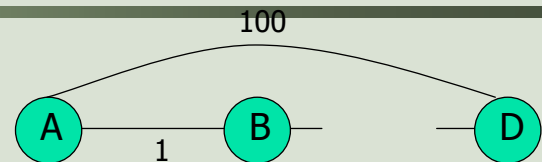
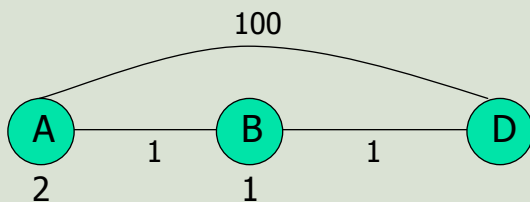
- $V(D) = 0 \Rightarrow V(i) = L(i)$

## Αλγόριθμος Bellman-Ford

- για αυθαίρετες (μη αρνητικές) εκτιμήσεις  $x_0(i)$  με  $x_0(D) = 0$   
έστω  $\{z_n(i), i \in S\}$  οι εκτιμήσεις που προκύπτουν από τις αρχικές αυτές τιμές
  - με επαγωγή στο  $n$  επαληθεύεται:  $y_n(i) \leq z_n(i) \leq x_n(i), i \in S, n \geq 0$ .
  - $x_n(i)$  και  $y_n(i)$  συγκλίνουν στο  $L(i) \Rightarrow z_n(i)$  συγκλίνει στο  $L(i)$
- $\Rightarrow$  Προσαρμογή:
- έστω ότι οι εκτιμήσεις έχουν λάβει κάποιες τιμές
  - η κατάσταση στο δίκτυο αλλάζει (διαφορετικά μήκη ζεύξεων)
  - οι εκτιμήσεις θα συγκλίνουν στις νέες ελάχιστες αποστάσεις

## Αλγόριθμος Bellman-Ford - Αργή Σύγκλιση

- η σύγκλιση μπορεί να είναι πολύ αργή
- προορισμός D
- $L(A, B) = L(B, D) = 1, L(A, D) = 100$
- η ζεύξη (B, D) τίθεται εκτός λειτουργίας

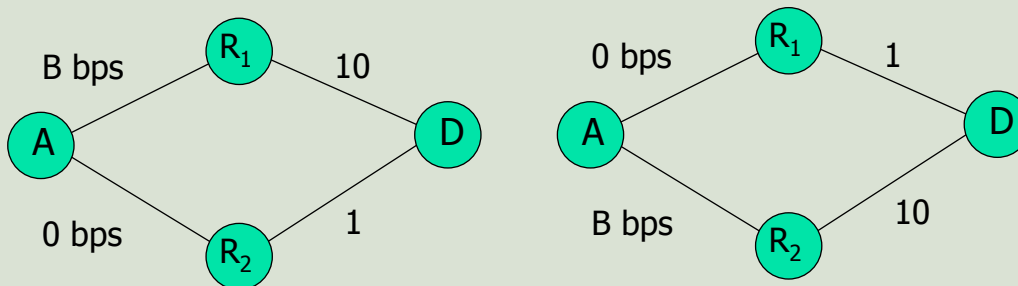


2	$2+1=3, \rightarrow A$
$3+1=4, \rightarrow B$	3
4	$4+1=5, \rightarrow A$
$5+1=6, \rightarrow B$	5
6	$6+1=7, \rightarrow A$
...	...
$97+1=98, \rightarrow B$	97
98	$98+1=99, \rightarrow A$
$99+1=100, \rightarrow B$	99
100	$100+1=101, \rightarrow A$
100	101

## Αλγόριθμος Bellman-Ford - Ταλαντώσεις

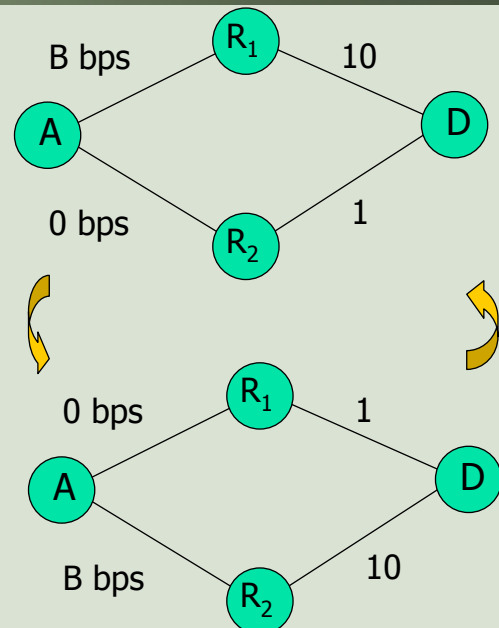
Ο αλγόριθμος Bellman-Ford ενδέχεται να οδηγήσει σε ταλαντώσεις  
Παράδειγμα:

- Ο κόμβος A έχει να στείλει στον D B bps
  - έστω ότι η καθυστέρηση στις ζεύξεις (R1, D) και (R2, D) είναι ίση με:
    - 1 όταν σχεδόν καθόλου κίνηση δε στέλνεται (χρόνος διάδοσης=1\*)
    - 10 όταν η ζεύξη μεταφέρει B bps
- [θεωρούνται αμελητέες οι καθυστερήσεις στις ζεύξεις (A, R1) και (A, R2)]



## Αλγόριθμος Bellman-Ford - Ταλαντώσεις

- έστω ότι αρχικά ο A στέλνει όλη την κίνηση μέσω του R1
- ⇒ καθυστέρηση στη (R1, D) = 10, καθυστέρηση στη (R2, D) = 1
- R<sub>1</sub>, R<sub>2</sub> γνωστοποιούν τις καθυστ. στον A
- ⇒ προτιμώμενο μονοπάτι μέσω του R2
- Ο A στέλνει όλη την κίνησή στον R2
- ⇒ καθυστέρηση στη (R1, D) = 1, καθυστέρηση στη (R2, D) = 10
- R<sub>1</sub>, R<sub>2</sub> γνωστοποιούν τις καθυστ. στον A
- .....
- ⇒ ταλάντωση μεταξύ δύο καταστάσεων
- Εναλλακτικά: ο A στέλνει τη μισή κίνησή στον R1 και την άλλη μισή στον R2
  - προσεγγίζεται αν ο A μετατοπίζει μικρό ποσοστό της κίνησής σε κάθε επανάληψη
  - όμως ο αλγόριθμος καθίσταται βραδύτερος στην περίπτωση που κάποιες ζεύξεις τεθούν εκτός λειτουργίας





## Δένδρο Επικάλυψης - Αλγόριθμος του Prim

- **S**: σύνολο από  $N$  κόμβους
- $\{L(i, j), i, j \in S\}$  ( $L(i, j) = \infty$  εάν δεν υπάρχει ζεύξη από τον  $i$  στον  $j$ )
- πλήρως αμφίδρομες ζεύξεις ( $L(i, j) = L(j, i)$ )

Σε ένα γράφημα  $\{S, L\}$ , ένα δένδρο επικάλυψης είναι ένα υπογράφημα το οποίο διασχίζει όλους τους κόμβους στο  $S$  και δεν περιέχει βρόχους

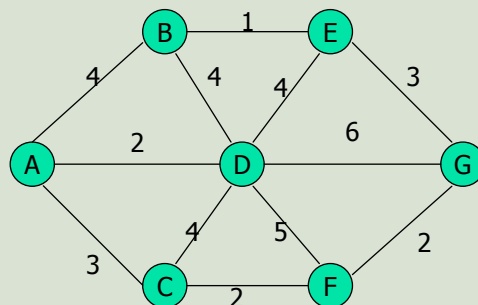
Ένα ελάχιστο δένδρο επικάλυψης είναι ένα δένδρο επικάλυψης με το ελάχιστο άθροισμα μηκών ζεύξεων (γενικά, όχι μοναδικό)

Αλγόριθμος ελαχίστου δένδρου επικάλυψης του Prim

- Ο αλγόριθμος ξεκινά με την επιλογή ενός αυθαίρετου κόμβου  $i_0$
- Σε κάθε βήμα προσαρτά τον πλησιέστερο κόμβο στο δένδρο το οποίο έχει κατασκευάσει έως τότε

## Αλγόριθμος του Prim

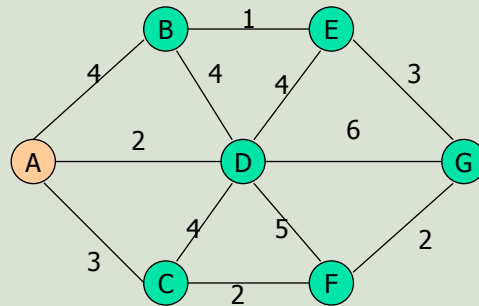
Παράδειγμα



## Αλγόριθμος του Prim

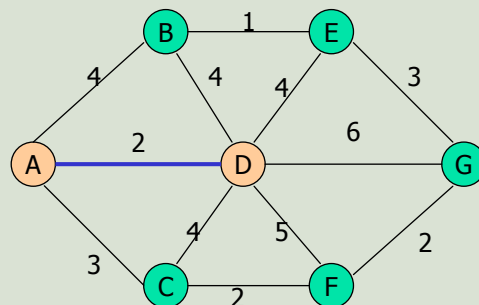
Παράδειγμα

Έστω  $i_0 \rightarrow A$



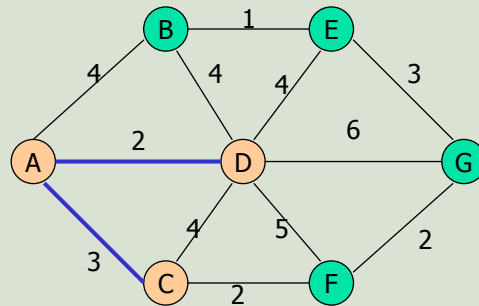
## Αλγόριθμος του Prim

Παράδειγμα



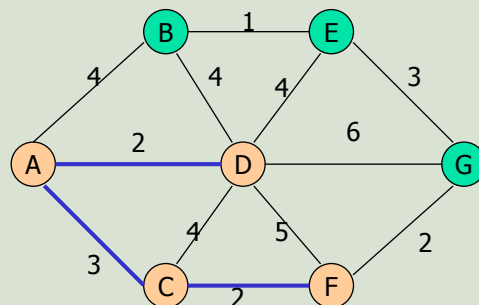
## Αλγόριθμος του Prim

Παράδειγμα



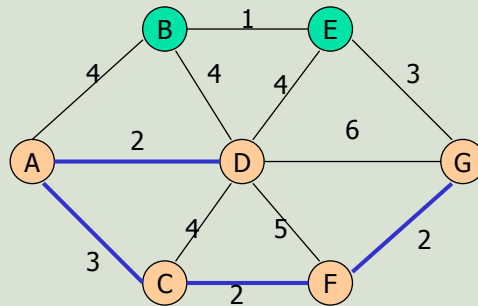
## Αλγόριθμος του Prim

Παράδειγμα



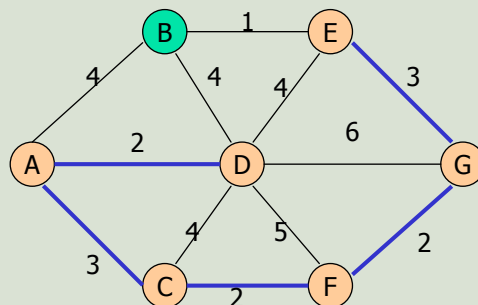
## Αλγόριθμος του Prim

Παράδειγμα



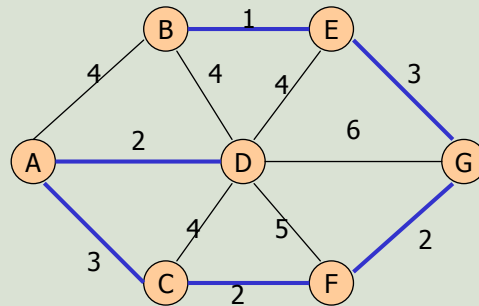
## Αλγόριθμος του Prim

Παράδειγμα



## Αλγόριθμος του Prim

Παράδειγμα



## Αλγόριθμος του Prim - Ανάλυση

Ένα δένδρο επικάλυψης **T** είναι ελάχιστο  $\Leftrightarrow$  περιέχει τη μικρότερη ζεύξη κάθε συνόλου τομής

η συνθήκη είναι αναγκαία

(Ένα δένδρο επικάλυψης **T** είναι ελάχιστο  $\Rightarrow$  περιέχει τη μικρότερη ζεύξη κάθε συνόλου τομής)

- το **T** είναι δένδρο επικάλυψης  $\Rightarrow$  περιέχει τουλάχιστον μία ζεύξη σε κάθε σύνολο τομής
- Εάν δεν περιέχει τη μικρότερη ζεύξη σε κάποιο σύνολο τομής, τότε το δένδρο μπορεί να γίνει μικρότερο αντικαθιστώντας μία ζεύξη του **T** σε εκείνο το σύνολο τομής με τη μικρότερη

## Αλγόριθμος του Prim - Ανάλυση

### η συνθήκη είναι ικανή

(το **T** περιέχει τη μικρότερη ζεύξη κάθε συνόλου τομής  $\Rightarrow$  είναι ελάχιστο)

- Έστω ότι κάποιο άλλο δένδρο **T'** είναι το ελάχιστο δένδρο επικάλυψης
- $\Rightarrow$  Στο **T'** ανήκει τουλάχιστον μια ζεύξη (έστω **L**) που δεν ανήκει στο **T**
- Απομακρύνοντας την **L** από το **T'** το σύνολο των κόμβων χωρίζεται σε δύο υποσύνολα **A** και **B** (όλοι οι κόμβοι στα **A** και **B** παραμένουν συνδεδεμένοι από το **T'**)
  
- οι ζεύξεις που ενώνουν 1 κόμβο του **A** και 1 του **B** συγκροτούν ένα σύνολο τομής
- Εξ υποθέσεως, το **T** περιέχει τη μικρότερη ζεύξη **L'** σε αυτό το σύνολο τομής
- αν η **L'** είναι μικρότερη από την **L**, τότε με αντικατάσταση της **L** από την **L'** στο **T'** σχηματίζεται ένα μικρότερο δένδρο (άτοπο)

## Διατερματική Μετάδοση

- το επίπεδο δικτύου υλοποιεί τη διατερματική παράδοση των πακέτων
- το επίπεδο μεταφοράς επιβλέπει δύο στοιχεία αυτής της παράδοσης
  - έλεγχο σφαλμάτων
  - έλεγχο συμφόρησης / ροής
  
- Έλεγχος ροής: διαδικασία που χρησιμοποιεί η πηγή για να ρυθμίζει το ρυθμό μετάδοσης της ώστε να μην υπερφορτώνει το δέκτη  
[Στο TCP, ο δέκτης γνωστοποιεί στην πηγή τον αριθμό των πακέτων που είναι σε θέση να δεχθεί]
  
- Έλεγχος συμφόρησης : μηχανισμός που χρησιμοποιούν οι πηγές για να περιορίσουν τη συμφόρηση στους κόμβους του δικτύου  
[Στο TCP, οι πηγές χρησιμοποιούν τις καθυστερήσεις των επιβεβαιώσεων ως ενδείξεις συμφόρησης και προσαρμόζουν ανάλογα το μέγεθος του παραθύρου του πρωτοκόλλου επαναμετάδοσης]
  
- Τα σφάλματα κατά την παράδοση των πακέτων οφείλονται
  - σφάλματα μετάδοσης που αλλοιώνουν τα πακέτα
  - απόρριψη πακέτων από το δρομολογητή όταν ο καταχωρητής του είναι γεμάτος

- επαναμεταδίδει μόνο τα πακέτα που δε φτάνουν σωστά στον προορισμό
- υλοποιείται από τους υπολογιστές στην πηγή και τον προορισμό
- ο προορισμός στέλνει μία επιβεβαίωση για κάθε σωστό πακέτο που λαμβάνει
- όταν η πηγή στέλνει ένα πακέτο, ξεκινά ένα χρονομετρητή
- η πηγή υποθέτει ότι το πακέτο δεν έφτασε σωστά αν η επιβεβαίωση δεν επιστρέψει μέσα σε συγκεκριμένο χρόνο [*προθεσμία (timeout)*] και επαναμεταδίδει το πακέτο

### η πηγή

- χρησιμοποιεί ένα μέγεθος παραθύρου  $W$
- στέλνει τα πακέτα 1, 2, ... ,  $W$  και περιμένει να επιστρέψει η επιβεβαίωση του πακέτου 1 πριν να στείλει το πακέτο  $W+1$
- αν όλα πάνε καλά λαμβάνει τις επιβεβαιώσεις πριν από τη λήξη της προθεσμίας και συνεχίζει να μεταδίδει
- αν μια επιβεβαίωση δε φτάσει πριν από τη λήξη της προθεσμίας, η πηγή επαναμεταδίδει αυτό το πακέτο

### ο προορισμός

- στέλνει επιβεβαίωση με αριθμό ακολουθίας  $K+1$  εάν έχει λάβει τα πακέτα 1, 2, ... ,  $K$  αλλά δεν έχει λάβει το  $K+1$
- παραδίδει τα πακέτα 1, 2, ... ,  $K$  που έλαβε στη σωστή σειρά και
- αποθηκεύει τα πακέτα  $K+2$ ,  $K+3$ ,  $K+4$  για να τα παραδώσει αφού έρθει το πακέτο  $K+1$

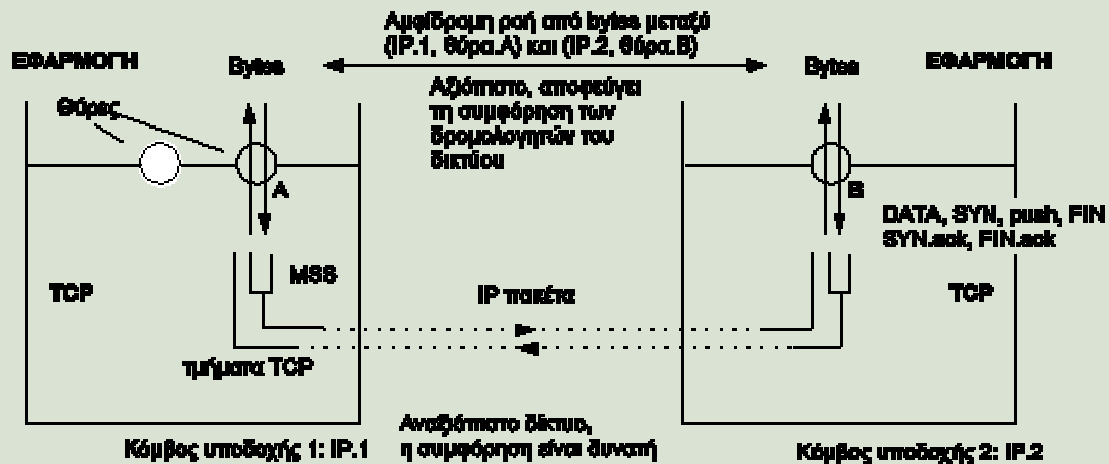
μηχανισμός γρήγορης επαναμετάδοσης (fast retransmit)

- Αν ο προορισμός λάβει τα πακέτα 1, 2, ... , K, K+2, K+3, K+4
- στέλνει τις επιβεβαιώσεις 2, 3, ... , K+1, K+1, K+1, K+1  
[επαναλαμβανόμενες επιβεβαιώσεις (duplicate acknowledgments)]
- Αν η η πηγή δει τρεις όμοιες επιβεβαιώσεις υποθέτει ότι το πακέτο K+1 έχει χαθεί και το επαναμεταδίδει χωρίς να περιμένει να εκπνεύσει η προθεσμία

- υλοποιεί μία (εικονική)  
**πλήρως αμφίδρομη ζεύξη μεταφοράς bytes**  
ανάμεσα σε δύο τερματικούς υπολογιστές
- η ροή των bytes είναι **ελεύθερη σφαλμάτων** (χρήση SRP)
- οι συνδέσεις TCP διαφοροποιούνται με βάση τον αριθμό θύρας (port number)  
(π.χ. η διεργασία του ηλεκτρονικού ταχυδρομείου παρακολουθεί τη θύρα 25)
- αντιμετώπιση του TCP σαν να εγκαθιδρύει δύο συνδέσεις:  
πελάτης → εξυπηρετητής      εξυπηρετητής → πελάτης
- τυπικά βήματα σε μία σύνδεση TCP:
  1. Ο πελάτης ξεκινά μία αμφίδρομη σύνδεση με τον εξυπηρ και ζητά ένα έγγραφο
  2. Ο εξυπηρετητής στέλνει το έγγραφο στον πελάτη
  3. Ο εξυπηρετητής κλείνει τη σύνδεση προς τον πελάτη
  4. Ο πελάτης κλείνει τη σύνδεση προς τον εξυπηρετητή

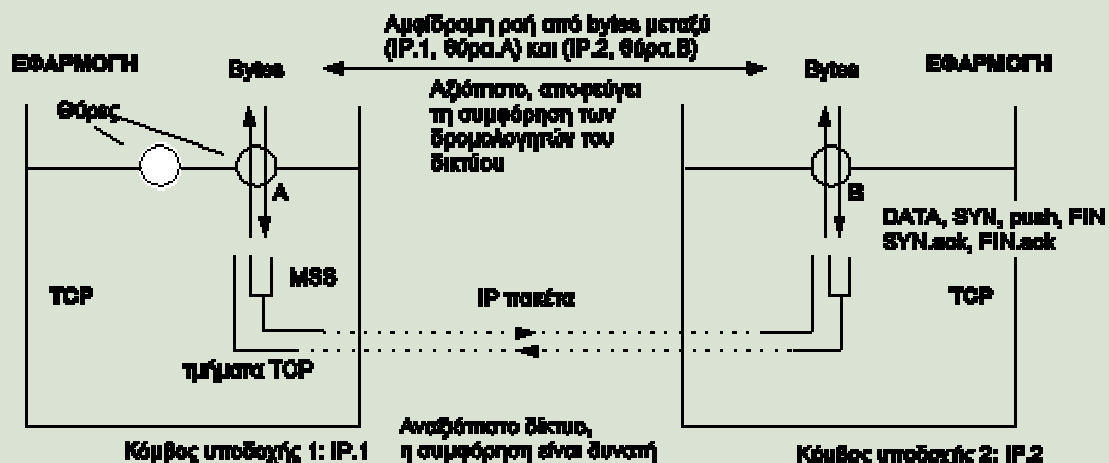


## Transmission Control Protocol – TCP (πρωτόκολλο ελέγχου μετάδοσης)



- Όταν ο εξυπηρετητής στέλνει δεδομένα στον πελάτη
  - στέλνει μία ροή από bytes στο TCP
  - τα bytes εισέρχονται σε έναν καταχωρητή
  - όταν γεμίσει ο καταχωρητής
    - το TCP τοποθετεί μία ομάδα από αυτά τα bytes σε ένα τμήμα (segment)
    - το παραδίδει στο IP

## Transmission Control Protocol – TCP (πρωτόκολλο ελέγχου μετάδοσης)



- Ο καταχωρητής έχει μέγεθος MSS Maximum Segment Size
- καθορίζεται κατά την έναρξη της σύνδεσης ή έχει προεπιλεγμένη τιμή (π.χ. 1460 bytes για ένα δίκτυο Ethernet)
  - Η πηγή μπορεί να εκδώσει μία εντολή προώθησης (PUSH) στο TCP (ώστε να αποφύγει να περιμένει μέχρι να γεμίσει ο καταχωρητής) ⇒  
όσα bytes υπάρχουν στον καταχωρητή τοποθετούνται σε ένα τμήμα που παραδίδεται στο IP

## TCP – Έναρξη σύνδεσης

βασίζεται σε μία χειραψία τριών βημάτων (three-way handshake)\*

- Ο πελάτης ζητά μία σύνδεση με τον εξυπηρετητή, καθορίζοντας
  - τη διεύθυνση IP και τον αριθμό θύρας του
  - τη διεύθυνση IP και τον αριθμό θύρας του εξυπηρετητή
  - τον αρχικό αριθμό ακολουθίας της σύνδεσης
  - (με βάση έναν ορολογιοδηγούμενο καταχωρητή (clock-register) που διαθέτει)
- Ο εξυπηρετητής στέλνει πίσω στον πελάτη μία επιβεβαίωση με
  - τις ίδιες διευθύνσεις
  - τους ίδιους αριθμούς θύρας
  - Τον ίδιο αριθμό ακολουθίας
- Ο πελάτης στέλνει το 1ο πακέτο δεδομένων με τον αρχικό αριθμό ακολουθίας

\*αποτρέπει τη δημιουργία σύγχυσης από την ενδεχόμενη άφιξη άσχετων, καθυστερημένων πακέτων (τέτοια πακέτα δεν θα είχαν τους ίδιους αριθμούς ακολουθίας)

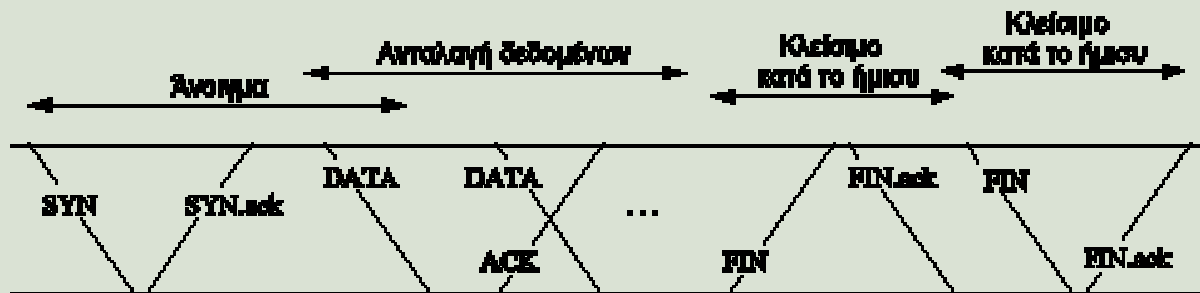
## TCP – Έναρξη σύνδεσης

- Εάν η έναρξη της σύνδεσης δεν επιβεβαιωθεί εγκαίρως
  - ο πελάτης ξαναπροσπαθεί μετά από 6 δευτερόλεπτα
  - αν αποτύχει, προσπαθεί τρεις φορές ακόμη (κάθε 24 δευτερόλεπτα)
  - μετά παραιτείται από την προσπάθεια
- Το πρωτόκολλο επιλεκτικής επανάληψης επιβλέπει τη μεταφορά των δεδομένων

### Κλείσιμο σύνδεσης

- Μία χειραψία δύο βημάτων (two-way handshake) κλείνει κάθε ήμισυ της σύνδεσης:
  - “Κλείνω τη σύνδεσή μου προς εσένα”
  - ακολουθεί μία επιβεβαίωση
- Αφού ο εξυπηρετητής στείλει την επιβεβαίωση της λήξης, παραμένει ανοικτός για κάποιο διάστημα  
(«βεβαιώνεται» ότι ο πελάτης έλαβε την επιβεβαίωση και δεν έστειλε ξανά μήνυμα λήξης)

## Τυπική ακολουθία μεταφορών στο TCP



## Επικεφαλίδα TCP

[ *S.port* | *D.port* | *Seq* | *Ack* | *FLAG* | *Window* | *CKS* | *URG* ]

**S.port , D.port** : αριθμοί θύρας της πηγής και του προορισμού

**Seq** : αριθμός ακολουθίας του πρώτου byte στο πακέτο

- η πηγή αριθμεί διαδοχικά τα bytes τα οποία στέλνει (μέσα στη ροή από bytes)
- σκοπός είναι να μην εκληφθεί εσφαλμένα ως πακέτο της παρούσας ανοικτής σύνδεσης ένα καθυστερημένο πακέτο προηγούμενης σύνδεσης
- ο αρχικός αριθμός ακολουθίας καθορίζεται από έναν ορολογιοδηγούμενο αριθμητή
- η αρίθμηση των πακέτων ξαναρχίζει από τον ίδιο αρχικό αριθμό ακολουθίας μετά από χρονικό διάστημα > μέγιστο χρόνο ζωής ενός πακέτου στο δίκτυο

## Επικεφαλίδα TCP

[ *S.port* | *D.port* | *Seq* | *Ack* | *FLAG* | *Window* | *CKS* | *URG* ]

**Ack** : αριθμός ακολουθίας του επόμενου byte που περιμένει ο προορισμός

- έχει λάβει όλα τα bytes με αριθμούς ακολουθίας έως  $Ack-1$
- δεν έχει λάβει το επόμενο byte Ack

(ενδέχεται να έχει λάβει bytes με αριθμούς ακολουθίας μεγαλύτερους από  $Ack+1$ )

**Window** : μέγεθος παραθύρου που είναι διατεθειμένος να αποδεχθεί ο προορισμός

- παράθυρο που χρησιμοποιεί η πηγή για το πρωτόκολλο επιλεκτικής επανάληψης  
=  $\min \{Window, W \text{ αλγόριθμου αργής εκκίνησης και αποφυγής συμφόρησης}\}$

**CKS** : άθροισμα ελέγχου

- υπολογίζεται από την πηγή σε ολόκληρο το TCP πακέτο

**URG** : δείκτης που υποδεικνύει το τελευταίο επείγον byte στη ροή

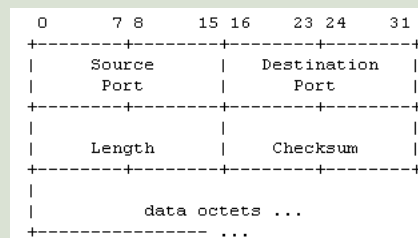
## Επικεφαλίδα TCP

[ *S.port* | *D.port* | *Seq* | *Ack* | *FLAG* | *Window* | *CKS* | *URG* ]

τιμή σημαίας	⇒ το πακέτο είναι
<i>FLAG=SYN</i>	η έναρξη μίας σύνδεσης
<i>FLAG=SYN.ack</i>	η επιβεβαίωση της έναρξης μίας σύνδεσης
<i>FLAG=Urgent</i>	ένα επείγον πακέτο
<i>FLAG=PUSH</i>	μία εντολή προώθησης
<i>FLAG=FIN</i>	η λήξη μίας σύνδεσης
<i>FLAG=FIN.ack</i>	η επιβεβαίωση της λήξης μίας σύνδεσης
<i>FLAG=reset</i>	μία επαναρύθμιση (reset) (εξαναγκάζει τον τερματισμό μίας σύνδεσης)

- προσθέτει στις δυνατότητες του IP:
  - πολυπλεξία  
(χρήση αριθμού θύρας (16 bits) για τη διάκριση των διεργασιών της πηγής και του προορισμού)
  - ανίχνευση σφαλμάτων
- Σε αντίθεση με το TCP (ΔΕΝ)
  - δεν χρησιμοποιεί επιβεβαιώσεις
  - δεν επαναμεταδίδει τα εσφαλμένα πακέτα
  - δεν ελέγχει τη ροή των δεδομένων

- επικεφαλίδα UDP



## Πρωτόκολλα Ζεύξης

- Serial Line IP – SLIP (πρωτόκολλο Internet σειριακής γραμμής)
- Point-to-Point Protocol – PPP (πρωτόκολλο σημείου προς σημείο)  
επιτρέπουν τη μεταφορά πακέτων IP μέσω μίας ζεύξης σημείου προς σημείο (τυπικά, τηλεφωνική γραμμή & modems)

### Serial Line IP - SLIP (πρωτόκολλο Internet σειριακής γραμμής)

- τα πλαίσια του SLIP
  - δεν έχουν πεδίο που να προσδιορίζει το είδος του πλαισίου (η ζεύξη μπορεί να χρησιμοποιηθεί μόνο για το συγκεκριμένο πρωτόκολλο)
  - δεν έχουν πεδίο ανίχνευσης σφαλμάτων (τα σφάλματα ελέγχονται από τα ανώτερα επίπεδα ή με πλαισίωση γραμμής)
- συμπιεσμένο SLIP (Compressed SLIP - CSLIP)
  - αποφεύγει την επανάληψη των επικεφαλίδων
  - διατηρεί πληροφορίες για την κατάσταση έως και 16 συνδέσεων TCP
  - μεταδίδει μόνο τις αλλαγές στα πεδία των επικεφαλίδων που αλλάζουν στα επακόλουθα πακέτα μίας δεδομένης σύνδεσης TCP

### ο αποστολέας

- αρχίζει και τελειώνει κάθε IP πακέτο με έναν ειδικό χαρακτήρα ASCII 0xc0 (END)
- αν ένα byte του πακέτου IP = END το αντικαθιστά με ESC END (ESC = 0xdb)
- αν ένα byte του πακέτου IP = ESC το αντικαθιστά με 0xdb και 0xdd

### ο παραλήπτης

- όταν δει το πρώτο END ανιχνεύει την αρχή του πακέτου IP
- όταν δει το ESC παραλείπει το byte αυτό, ελέγχει το επόμενο byte
  - αν το επόμενο byte = END ⇒ αυτό το END είναι ένα byte του πακέτου IP
  - αν το επόμενο byte = 0xdd ⇒ μέσα στο IP πακέτο υπήρχε ένα byte 0xdb

(αντικαθιστά το SLIP σε πολλές ζεύξεις σημείου προς σημείο)

- Η ζεύξη μπορεί να είναι
  - ασύγχρονη (μεταδίδει ένα χαρακτήρα ASCII κάθε φορά)
  - σύγχρονη (μεταδίδει μεγάλα πακέτα)
- επιτρέπει στις δύο συσκευές να διαπραγματεύονται επιλογές όπως
  - να αποφασίζουν ποιο πρωτόκολλο θα χρησιμοποιηθεί στη ζεύξη
  - να συμφωνούν εάν θα χρησιμοποιηθεί συμπίεση των επικεφαλίδων
- τα πακέτα του PPP περιέχουν πεδίο ανίχνευσης σφαλμάτων (CRC των 2 bytes)



Σχεδιάστηκε για να διορθώσει ορισμένες αδυναμίες του IPv4:

- Οι διευθύνσεις των 32 bits κινδυνεύουν να εξαντληθούν (το IPv6 χρησιμοποιεί διευθύνσεις των 128 bits)
- Το IPv4 έχει ανεπαρκή έλεγχο της ποιότητας υπηρεσίας που παρέχει
- Το IPv4 δεν έχει ενσωματωμένους μηχανισμούς ασφάλειας
  
- Η επικεφαλίδα των πακέτων IPv6 αποτελείται από
  - μία βασική επικεφαλίδα (40 bytes)
  - (ενδεχόμενες) επικεφαλίδες επέκτασης που ακολουθούν