

# Δίκτυα Υπολογιστών I

Δίκτυα Μεταγωγής και Διαδίκτυα: Μέρος Γ'



Ευάγγελος Παπαπέτρου

Τμ. Μηχ. Η/Υ & Πληροφορικής, Παν. Ιωαννίνων

## Διάρθρωση

- 1 Ο ρόλος της Δρομολόγησης
- 2 Δρομολόγηση συντομότερης διαδρομής  
Μοντελοποίηση και βασικά ζητήματα υλοποίησης  
Δυναμική δρομολόγηση
- 3 Αλγόριθμοι συντομότερης διαδρομής  
Αλγόριθμοι διανυσμάτων απόστασης  
Αλγόριθμοι κατάστασης συνδέσμων
- 4 Ιεραρχική δρομολόγηση



## Διάρθρωση

- 1 Ο ρόλος της Δρομολόγησης
- 2 Δρομολόγηση συντομότερης διαδρομής  
Μοντελοποίηση και βασικά ζητήματα υλοποίησης  
Δυναμική δρομολόγηση
- 3 Αλγόριθμοι συντομότερης διαδρομής  
Αλγόριθμοι διανυσμάτων απόστασης  
Αλγόριθμοι κατάστασης συνδέσμων
- 4 Ιεραρχική δρομολόγηση



## Σκοπός της Δρομολόγησης

Ζήτημα: Ποια είναι η **βέλτιστη** διαδρομή για την παράδοση δεδομένων από έναν **υπολογιστή αποστολέα (source node)** σε έναν **υπολογιστή παραλήπτη (destination node)**;

### Δρομολόγηση

Ο μηχανισμός με τον οποίο επιλέγεται το **βέλτιστο μονοπάτι** σε ένα δίκτυο μεταγωγής (ή διαδίκτυο) για την επικοινωνία δύο υπολογιστών

### Δρομολόγηση (από τη σκοπιά του μεταγωγέα)

Επιλογή της **βέλτιστης θύρας εξόδου (επόμενου άλματος)** με βάση τη διεύθυνση προορισμού

⇒ Οι λειτουργίες της δρομολόγησης και της μεταγωγής εκτελούνται συνήθως καταναμημένα στους υπολογιστές που ονομάζονται **δρομολογητές**



## Δρομολόγηση vs Προώθηση (1/2)

Διάκριση μεταξύ προώθησης και δρομολόγησης στα δίκτυα μεταγωγής πακέτου

**Δρομολόγηση:** η διαδικασία σύνταξης των πινάκων προώθησης υλοποιείται με σύνθετους καταναμημένους αλγόριθμους

**Προώθηση:** η αποστολή του πακέτου προς μια κατεύθυνση με βάση τη διεύθυνση προορισμού και τον πίνακα προώθησης

απλή διαδικασία που εκτελείται τοπικά στο μεταγωγέα

Η δρομολόγηση παράγει τον **πίνακα δρομολόγησης** για κάθε μεταγωγέα

- ▶ η σύνταξη του πίνακα προώθησης γίνεται με βάση τον πίνακα δρομολόγησης

⇒ Αν και διαφορετικοί, σε πολλές περιπτώσεις οι όροι **πίνακας δρομολόγησης** και ο **πίνακας προώθησης** χρησιμοποιούνται ισοδύναμα

## Δρομολόγηση vs Προώθηση (2/2)

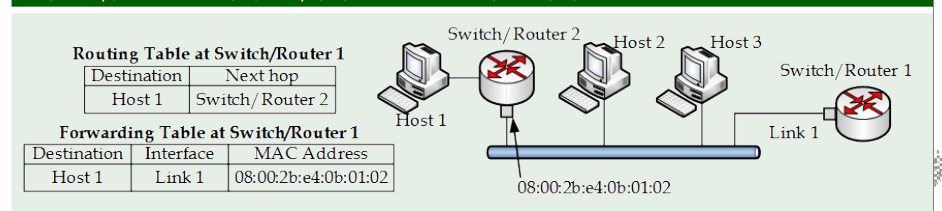
Ένας πίνακας δρομολόγησης περιέχει **αντιστοιχίες** μεταξύ **υπολογιστή προορισμού** και **εξερχόμενης θύρας** (ή επόμενου άλματος)

- ▶ επίσης μπορεί να περιέχει πληροφορίες που σχετίζονται με την υλοποίηση του αλγόριθμου δρομολόγησης

Ο πίνακας προώθησης στηρίζεται στον πίνακα δρομολόγησης αλλά περιέχει **επιπλέον πληροφορίες:**

- ▶ τον προσδιορισμό της θύρας εξόδου (ή του κλάδου εξόδου)
- ▶ λεπτομέρειες για τη μετάδοση των δεδομένων στο σύνδεσμο (π.χ. η διεύθυνση MAC του επόμενου άλματος)

Παράδειγμα πίνακα δρομολόγησης και πίνακα προώθησης



## Πολυδιανομή και ευρεία εκπομπή

Η παραδοσιακή δρομολόγηση επιλύει το πρόβλημα εύρεσης διαδρομής για την **επικοινωνία δύο υπολογιστών υπηρεσίας (unicast επικοινωνία)**

Η γενίκευση του προβλήματος της δρομολόγησης περιλαμβάνει την εύρεση της **βέλτιστης ομάδας διαδρομών** για την επικοινωνία στα παρακάτω σενάρια:

- ▶ ένας προς πολλούς (one to many) ή **πολυδιανομή (multicast)**
  - ▶ ένας προς όλους (one to all) ή **ευρεία εκπομπή (broadcast)**
- ένας απλός αλγόριθμος: **πλημμύρα (flooding)**

## Διάρθρωση

- 1 Ο ρόλος της Δρομολόγησης
- 2 Δρομολόγηση συντομότερης διαδρομής
  - Μοντελοποίηση και βασικά ζητήματα υλοποίησης
  - Δυναμική δρομολόγηση
- 3 Αλγόριθμοι συντομότερης διαδρομής
  - Αλγόριθμοι διανυσμάτων απόστασης
  - Αλγόριθμοι κατάστασης συνδέσμων
- 4 Ιεραρχική δρομολόγηση

## Διάρθρωση

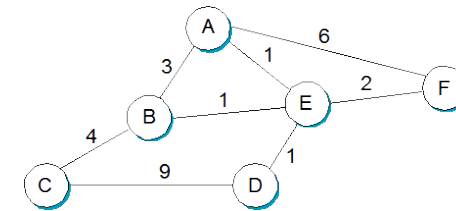
- 1 Ο ρόλος της Δρομολόγησης
- 2 Δρομολόγηση συντομότερης διαδρομής
  - Μοντελοποίηση και βασικά ζητήματα υλοποίησης
  - Δυναμική δρομολόγηση
- 3 Αλγόριθμοι συντομότερης διαδρομής
  - Αλγόριθμοι διανυσμάτων απόστασης
  - Αλγόριθμοι κατάστασης συνδέσμων
- 4 Ιεραρχική δρομολόγηση



## Μοντελοποίηση (1/2)

Ένα δίκτυο μπορεί να αναπαρασταθεί με ένα **γράφημα**:

- ▶ οι κόμβοι αναπαριστούν τους **υπολογιστές υπηρεσίας**, τους **δρομολογητές** ή ακόμα και **ολόκληρα δίκτυα**
- ▶ οι ακμές αναπαριστούν τους **συνδέσμους** του δικτύου
- ▶ κάθε ακμή περιγράφεται με ένα **κόστος**



⇒ Η πιο ακριβής αναπαράσταση ενός δικτύου απαιτεί τη χρήση δύο ακμών για κάθε σύνδεσμο (μία για κάθε κατεύθυνση) και ενός κόστους για κάθε ακμή

## Μοντελοποίηση (2/2)

Διαδρομή μεταξύ δύο κόμβων: ένα σύνολο από διαδοχικές ακμές που "ενώνουν" τους κόμβους αυτούς

Κόστος διαδρομής: το **άθροισμα από τα κόστη των ακμών** που αποτελούν τη διαδρομή

Πρόβλημα δρομολόγησης: εντοπισμός της διαδρομής με το **μικρότερο κόστος**

- ▶ η δρομολόγηση αυτής της μορφής είναι γνωστή με τον όρο **δρομολόγηση συντομότερης διαδρομής (shortest path routing)**



## Συναρτήσεις κόστους (1/2)

Η επιλογή της συνάρτησης κόστους καθορίζει τα **χαρακτηριστικά της βέλτιστης διαδρομής** και επομένως την **παρεχόμενη υπηρεσία**

- ▶ παράδειγμα: κόστος=καθυστέρηση: η βέλτιστη διαδρομή είναι αυτή με τη **μικρότερη καθυστέρηση** και συνήθως είναι χρήσιμη για **μεταφορά δεδομένων σε πραγματικό χρόνο (real-time)**

Τυπικές συναρτήσεις κόστους που χρησιμοποιούνται:

- ▶ συνήθως σχετίζονται με την **καθυστέρησης διάδοσης**, το **εύρος ζώνης**, το **τρέχον φόρτο** ή/και την **καθυστέρηση αναμονής**
- ▶ σπανιότερα σχετίζονται με την **αξιοπιστία**
- ▶ σε αρκετές περιπτώσεις για απλότητα της υλοποίησης γίνεται η παραδοχή ότι **το κόστος κάθε ακμής είναι ένα**



## Συναρτήσεις κόστους (2/2)

Η επιλογή της συνάρτησης κόστους μπορεί να επηρεαστεί από επιμέρους ζητήματα όπως:

- ▶ η ευκολία υλοποίησης
  - π.χ. σε πολλές περιπτώσεις χρησιμοποιούνται **μετρικές που δε μεταβάλλονται με το χρόνο**
- ▶ η γενικότερη εύρυθμη λειτουργία του δικτύου
  - π.χ. εμφάνιση του φαινομένου των **μετατοπίσεων φορτίου (load oscillations)**
- ▶ η εξοικονόμηση πόρων
  - π.χ. η επιλογή μιας συνάρτησης κόστους μπορεί να οδηγήσει σε διαδρομές με πολλά άλματα και επομένως σε σπατάλη των πόρων του δικτύου



## Σηματοδοσία δρομολόγησης και κόστος

Ένας αλγόριθμος δρομολόγησης απαιτεί την συλλογή **πληροφοριών δρομολόγησης** που περιγράφουν την **κατάσταση του δικτύου**

Οι πληροφορίες δρομολόγησης μεταφέρονται σε **μηνύματα δρομολόγησης (routing messages)** που ανταλλάσσουν οι δρομολογητές

- ▶ τα μηνύματα είναι επίσης γνωστά με τον όρο **σηματοδοσία (routing signaling)** ή **φορτίο δρομολόγησης (routing overhead)**

Το φορτίο δρομολόγησης αποτελεί το σημαντικότερο κόστος ενός αλγόριθμου δρομολόγησης γιατί **καταναλώνει το εύρος ζώνης**

Άλλα κόστη ενός αλγόριθμου δρομολόγησης:

- ▶ η απαιτούμενη **επεξεργαστική ισχύς**
- ▶ ο απαιτούμενος **χώρος αποθήκευσης**, κλπ



## Κατανεμημένη vs Κεντροποιημένη υλοποίηση (1/2)

Σημαντική απόφαση: **κατανεμημένη (decentralized)** ή **κεντροποιημένη (centralized)** υλοποίηση της δρομολόγησης

Κεντροποιημένη προσέγγιση:

- ▶ ένας κόμβος **συγκεντρώνει τις πληροφορίες δρομολόγησης**, υπολογίζει τις **συντομότερες διαδρομές** και τις **διανέμει** στους υπόλοιπους κόμβους

Ζητήματα στην κεντροποιημένη προσέγγιση:

- ▶ δεν προσφέρει μια **κλιμακώσιμη** λύση
- ▶ η ορθή λειτουργία του δικτύου εξαρτάται από την ορθή λειτουργία ενός κόμβου



## Κατανεμημένη vs Κεντροποιημένη υλοποίηση (2/2)

Κατανεμημένη προσέγγιση:

- ▶ κάθε κόμβος ανταλλάσσει μηνύματα δρομολόγησης με τους άλλους κόμβους του δικτύου για να συλλέξει πληροφορίες δρομολόγησης
- ▶ κάθε κόμβος υπολογίζει **τοπικά** τις βέλτιστες διαδρομές προς κάθε άλλο κόμβο

Ζήτημα στην κατανεμημένη υλοποίηση:

- ▶ δύο κόμβοι ενδέχεται να έχουν **διαφορετική εικόνα** για την κατάσταση του δικτύου **σε μια δεδομένη χρονική στιγμή**
  - αιτία: η καθυστέρηση μέχρι ένας κόμβος να λάβει τις πληροφορίες δρομολόγησης για ένα τμήμα του δικτύου
  - πρόβλημα: ο **σχηματισμός βρόχων** επειδή δύο κόμβοι ενδέχεται να υπολογίζουν διαφορετικά μονοπάτια προς έναν προορισμό



## Διάρθρωση

- 1 Ο ρόλος της Δρομολόγησης
- 2 Δρομολόγηση συντομότερης διαδρομής
  - Μοντελοποίηση και βασικά ζητήματα υλοποίησης
  - Δυναμική δρομολόγηση
- 3 Αλγόριθμοι συντομότερης διαδρομής
  - Αλγόριθμοι διανυσμάτων απόστασης
  - Αλγόριθμοι κατάστασης συνδέσμων
- 4 Ιεραρχική δρομολόγηση



## Προσαρμογή στις μεταβολές του δικτύου

Ζήτημα: η κατάσταση ενός δικτύου συνήθως **μεταβάλλεται** με το χρόνο

- ▶ **αστοχίες** κόμβων και συνδέσμων, **προσθήκη** νέων κόμβων στο δίκτυο
- ▶ **μεταβολή** του κόστους των συνδέσμων
- ▶ **κινητικότητα** κόμβων

Κάθε δρομολογητής πρέπει **δυναμικά** να ανανεώνει την εικόνα που έχει για το δίκτυο

Ακριβής γνώση του δικτύου  $\Rightarrow$  Επιτυχημένη δρομολόγηση

Κάθε δρομολογητής πρέπει:

- ▶ να **συλλέγει πληροφορίες δρομολόγησης** ώστε η εικόνα για το δίκτυο να είναι έγκυρη (up to date)
- ▶ να **υπολογίζει εκ νέου** τις συντομότερες διαδρομές κάθε φορά που η εικόνα του δικτύου μεταβάλλεται

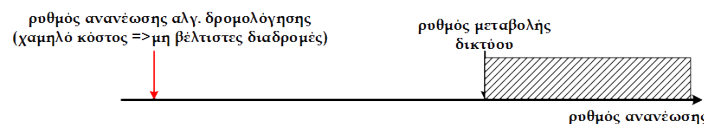


## Τεχνικές ανανέωσης (1/2)

Ζήτημα: Πότε πρέπει ένας δρομολογητής να ανανεώνει την εικόνα του για το δίκτυο;

- ▶ Μια πρώτη απάντηση: κάθε φορά που το δίκτυο αλλάζει
- ▶ Προβληματισμοί: η συλλογή πληροφοριών δρομολόγησης συνεπάγεται ένα **φορτίο δρομολόγησης** (κατανάλωση εύρους ζώνης)

Η επιλογή του ρυθμού ανανέωσης καθορίζει το ισοζύγιο **φορτίο δρομολόγησης vs ποιότητα υπολογιζόμενων διαδρομών**



### Εγκυρότητα πληροφοριών δρομολόγησης

$\Rightarrow$  Σε κάθε περίπτωση η εγκυρότητα των πληροφοριών δρομολόγησης είναι περιορισμένη εξαιτίας του απαιτούμενου χρόνου για τη συλλογή τους

## Τεχνικές ανανέωσης (2/2)

Οι σημαντικότερες προσεγγίσεις για τη δυναμική ανανέωση των συντομότερων διαδρομών είναι:

- ▶ **περιοδική (periodic)**: ο υπολογισμός των διαδρομών γίνεται περιοδικά ανεξάρτητα από το αν έχει μεταβληθεί ή όχι η κατάσταση του δικτύου
- ▶ **πυροδοτούμενη (triggered) ή εξαναγκασμένη**: ο υπολογισμός των διαδρομών γίνεται μόνο αν έχει προκύψει κάποια αλλαγή στην κατάσταση του δικτύου
- ▶ **αιτούμενη (on-demand)**: ο υπολογισμός μιας διαδρομής γίνεται μόνο αν η διαδρομή είναι απαραίτητη



## Διάρθρωση

- 1 Ο ρόλος της Δρομολόγησης
- 2 Δρομολόγηση συντομότερης διαδρομής  
Μοντελοποίηση και βασικά ζητήματα υλοποίησης  
Δυναμική δρομολόγηση
- 3 Αλγόριθμοι συντομότερης διαδρομής  
Αλγόριθμοι διανυσμάτων απόστασης  
Αλγόριθμοι κατάστασης συνδέσμων
- 4 Ιεραρχική δρομολόγηση



## Κατηγορίες αλγόριθμων δρομολόγησης

- Οι αλγόριθμοι δρομολόγησης συντομότερης διαδρομής μπορούν να κατηγοριοποιηθούν με διαφορετικούς τρόπους
- Οι συνηθέστερες κατηγοριοποιήσεις αφορούν:
- ▶ την τεχνική ανανέωσης που χρησιμοποιείται  
περιοδικής, αιτούμενης ή εξαναγκασμένης ανανέωσης
  - ▶ το είδος και την έκταση των πληροφοριών κατάστασης του δικτύου, τις οποίες συλλέγουν  
αλγόριθμοι διανυσμάτων απόστασης (distance vector)  
αλγόριθμοι κατάστασης συνδέσμων (link state)



## Διάρθρωση

- 1 Ο ρόλος της Δρομολόγησης
- 2 Δρομολόγηση συντομότερης διαδρομής  
Μοντελοποίηση και βασικά ζητήματα υλοποίησης  
Δυναμική δρομολόγηση
- 3 Αλγόριθμοι συντομότερης διαδρομής  
Αλγόριθμοι διανυσμάτων απόστασης  
Αλγόριθμοι κατάστασης συνδέσμων
- 4 Ιεραρχική δρομολόγηση



## Εισαγωγή

Η χρήση διανυσμάτων απόστασης (distance vector) στη δρομολόγηση είναι μια ιδέα που έχει εφαρμοστεί σε πολλούς αλγόριθμους δρομολόγησης

Κεντρική ιδέα: κάθε κόμβος μπορεί να μάθει το δίκτυο λαμβάνοντας επαναλαμβανόμενες ενημερώσεις από τους γειτονικούς του κόμβους

- ▶ κάθε ενημέρωση περιέχει πληροφορίες για τις διαδρομές που ο αποστολέας της ενημέρωσης γνωρίζει

Οι αλγόριθμοι διανυσμάτων απόστασης υλοποιούνται **κατανεμημένα**

⇒ Ο βασικός αλγόριθμος διανυσμάτων απόστασης ονομάζεται και αλγόριθμος Bellman-Ford προς τιμήν αυτών που τον πρότειναν



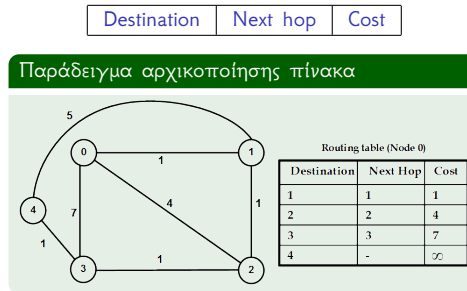
## Πίνακας δρομολόγησης και αρχικοποίηση

Κάθε κόμβος διατηρεί ένα πίνακα δρομολόγησης με μία **εγγραφή (διάνυσμα)** για κάθε πιθανό προορισμό που περιέχει:

- ▶ το επόμενο άλμα (κόμβο) ώστε το πακέτο να φτάσει στον προορισμό
- ▶ το **συνολικό κόστος (απόσταση)** μέχρι τον προορισμό

Αρχικοποίηση πίνακα:

- ▶ το συνολικό κόστος για τους γείτονες αρχικοποιείται με το κόστος του συνδέσμου
- ▶ για όλους τους άλλους κόμβους το συνολικό κόστος αρχικά θεωρείται άπειρο



## Αποστολή και λήψη ενημερώσεων (1/2)

Κάθε κόμβος αποστέλλει **μηνύματα δρομολόγησης (ή ενημερώσεις)** σε **όλους τους γείτονες**

- ▶ οι ενημερώσεις αποστέλλονται κάθε φορά που ο κόμβος παρατηρεί μια **μεταβολή στον πίνακα δρομολόγησης**

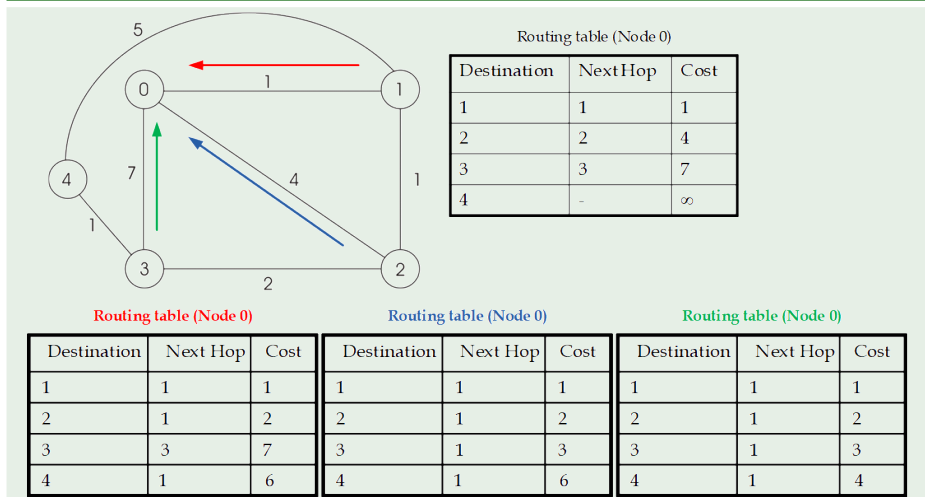
Ένα μήνυμα δρομολόγησης περιέχει **τις αποστάσεις** που ο κόμβος γνωρίζει για τους άλλους κόμβους του δικτύου

Αντίστοιχα, κάθε κόμβος **λαμβάνει ενημερώσεις** από τους γείτονές του

- ▶ χρησιμοποιεί τις πληροφορίες που λαμβάνει για να ανακαλύψει **συντομότερες διαδρομές**
- ▶ αν ένας κόμβος  $v$  λάβει μια ενημέρωση από τον  $z$  όπου  $cost(z, w) + cost(v, z) < cost(v, w)$  τότε η διαδρομή μέσω του  $z$  είναι η **συντομότερη διαδρομή** του  $v$  προς τον  $w$

## Αποστολή και λήψη ενημερώσεων (2/2)

Παράδειγμα ενημέρωσης πίνακα δρομολόγησης: σειρά λήψης ενημερώσεων: 1,2,4,3



## Σύγκλιση αλγόριθμου

Η αποστολή ενημερώσεων από έναν κόμβο σταματά όταν δεν προκύπτουν αλλαγές στον πίνακα δρομολόγησής του

### Σύγκλιση (Convergence)

Η κατάσταση κατά την οποία οι κόμβοι του δικτύου έχουν υπολογίσει τις συντομότερες διαδρομές και δεν ανταλλάσσουν ενημερώσεις

Πριν τη σύγκλιση του αλγόριθμου κάποιες διαδρομές δεν έχουν υπολογιστεί ενώ κάποιες άλλες δεν είναι οι συντομότερες

Ο χρόνος σύγκλισης ενός αλγόριθμου δρομολόγησης εξαρτάται από πολλούς παράγοντες όπως:

- ▶ το μέγεθος του δικτύου
- ▶ η διάμετρος του δικτύου
- ▶ η μεταβλητότητα του δικτύου
- ▶ οι καθυστερήσεις διάδοσης και μετάδοσης στους συνδέσμους

## Πρόβλημα μέτρησης μέχρι το άπειρο

Η αστοχία ενός κόμβου ή συνδέσμου μπορεί να οδηγήσει στην αδυναμία σύγκλισης του αλγόριθμου

- ▶ η κατάσταση είναι γνωστή ως **μέτρηση ως το άπειρο (count-to-infinity)**

Οι γνωστότερες τεχνικές αντιμετώπισης του φαινομένου είναι:

- ▶ χρήση ενός μικρού αριθμού για την αναπαράσταση του άπειρου κόστους
- ▶ χωριστός ορίζοντας (split horizon)
- ▶ χωριστός ορίζοντας με αντιστροφή δηλητηρίου (split horizon with poison reverse)

A	B	C	D	E		A	B	C	D	E	
•	•	•	•	•	Initially	•	1	2	3	4	Initially
1	•	•	•	•	After 1 exchange	•	3	2	3	4	After 1 exchange
1	2	•	•	•	After 2 exchanges	•	3	4	3	4	After 2 exchanges
1	2	3	•	•	After 3 exchanges	•	5	4	5	4	After 3 exchanges
1	2	3	4	•	After 4 exchanges	•	5	6	5	6	After 4 exchanges
						•	7	6	7	6	After 5 exchanges
						•	7	8	7	8	After 6 exchanges
						•	•	•	•	•	

**fails** (circled in red) points to node A in the 6th row.

**B marks route to A as broken** (blue arrow) points to node B in the 6th row.



## Πυροδοτούμενη και Περιοδική ανανέωση

Η αποστολή ενημερώσεων σε κάθε μεταβολή του πίνακα δρομολόγησης είναι μια **πυροδοτούμενη (triggered)** τεχνική ενημέρωσης

**Ζήτημα:** ανίχνευση της **κατάρρευσης κόμβων και συνδέσμων**

**Λύση:** ένας κόμβος αποστέλλει και **περιοδικές (periodic)** ενημερώσεις

- ▶ η μέθοδος έχει σκοπό να ενημερώσει τους γειτονικούς κόμβους:
  - ότι ο αποστολέας της ενημέρωσης εξακολουθεί να λειτουργεί για εναλλακτικές διαδρομές προς κόμβους των οποίων η προηγούμενη διαδρομή έχει καταστεί ανενεργή
- ▶ η περίοδος αποστολής ενημερώσεων εξαρτάται από τη μεταβλητότητα του δικτύου



## Κριτική αλγόριθμου

**Πλεονεκτήματα:**

- ▶ κατανεμημένος αλγόριθμος με χαμηλή πολυπλοκότητα υλοποίησης
- ▶ απαιτεί μικρό αποθηκευτικό χώρο σε κάθε κόμβο
- ▶ το φορτίο δρομολόγησης που παράγεται υπό φυσιολογικές συνθήκες λειτουργίας είναι σχετικά χαμηλό
  - μόνο οι γειτονικοί κόμβοι ανταλλάσσουν μηνύματα δρομολόγησης

**Μειονεκτήματα:**

- ▶ το πρόβλημα μέτρησης μέχρι το άπειρο
- ▶ η σύγκλιση του αλγόριθμου μπορεί να είναι αργή σε μεγάλα και ταχέως μεταβαλλόμενα δίκτυα
- ▶ σε εξαιρετικές περιπτώσεις υψηλής μεταβλητότητας του δικτύου η σύγκλιση μπορεί να μην είναι εφικτή



## Διάρθρωση

1 Ο ρόλος της Δρομολόγησης

2 Δρομολόγηση συντομότερης διαδρομής  
Μοντελοποίηση και βασικά ζητήματα υλοποίησης  
Δυναμική δρομολόγηση

3 Αλγόριθμοι συντομότερης διαδρομής  
Αλγόριθμοι διανυσμάτων απόστασης  
Αλγόριθμοι κατάστασης συνδέσμων

4 Ιεραρχική δρομολόγηση





## Εισαγωγή (1/2)

Η ιδέα της χρήσης πληροφοριών **κατάστασης συνδέσμων (link state)** χρησιμοποιείται από πολλούς αλγόριθμους δρομολόγησης

**Κεντρική ιδέα:** κάθε κόμβος αποστέλλει σε όλους τους κόμβους του δικτύου **πληροφορίες κατάστασης** σχετικά με τους γειτονικούς του κόμβους

- ▶ με τον τρόπο αυτό όλοι οι κόμβοι μαθαίνουν την κατάσταση ολόκληρου του δικτύου

### Distance Vector vs Link State

- Distance vector: κάθε κόμβος λαμβάνει ενημερώσεις μόνο από τους γείτονες και μαθαίνει τις πληροφορίες που αυτοί έχουν για όλους τους κόμβους του δικτύου
- Link State: κάθε κόμβος λαμβάνει ενημερώσεις από όλους τους κόμβους και μαθαίνει πληροφορίες σχετικά με τους γείτονές τους



## Εισαγωγή (2/2)

Ο βασικός αλγόριθμος αποτελείται από δύο επιμέρους μηχανισμούς:

- ▶ την διάδοση των πληροφοριών κατάστασης που υλοποιείται με έναν μηχανισμό **αξιόπιστης πλημμύρας (reliable flooding)**
- ▶ τον **υπολογισμό των διαδρομών**
  - ▶ **προσοχή:** είναι δυνατός ο υπολογισμός της **πλήρους διαδρομής** και όχι μόνο του επόμενου άλματος

Η τεχνική μπορεί να εφαρμοστεί με **κατανεμημένο** ή **κεντρικοποιημένο** τρόπο

- ▶ **κατανεμημένη υλοποίηση:** οι κόμβοι διαδίδουν πληροφορίες και ο καθένας υπολογίζει τις καλύτερες διαδρομές
- ▶ **κεντρικοποιημένη υλοποίηση:** όλοι οι κόμβοι διαδίδουν πληροφορίες προς έναν κόμβο αλλά μόνο ο κόμβος αυτός υπολογίζει τις διαδρομές και τις διανέμει στους άλλους κόμβους



## Μηνύματα LSP και πίνακας δρομολόγησης (1/2)

Οι πληροφορίες που αποστέλλει κάθε κόμβος οργανώνονται σε πακέτα που ονομάζονται **πακέτα κατάστασης συνδέσμων (Link State Packets, LSPs)**

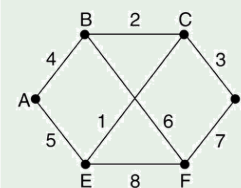
Ένα πακέτο LSP περιέχει:

- ▶ το αναγνωριστικό (ID) του κόμβου που δημιούργησε το πακέτο
- ▶ έναν αριθμό ακολουθίας (sequence number)
- ▶ ένα χρόνο ζωής (time to live, TTL) για το πακέτο
- ▶ ένα κατάλογο από εγγραφές του τύπου **<γείτονας, κόστος συνδέσμου>**



## Μηνύματα LSP και πίνακας δρομολόγησης (2/2)

### Example



		Link		State		Packets	
A	B	C	D	E	F		
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6		
E 5	C 2	D 3	F 7	C 1	D 7		
	F 6	E 1		F 8	E 8		

Κάθε κόμβος διατηρεί ένα **πίνακα δρομολόγησης** με εγγραφές της μορφής **<προορισμός, επόμενο άλμα, κόστος>**

⇒ Στην πραγματικότητα η δομή του πίνακα δρομολόγησης εξαρτάται από την τεχνική μεταγωγής. Π.χ., στη δρομολόγηση προέλευσης ο πίνακας δρομολόγησης μπορεί να περιέχει **ολόκληρη τη διαδρομή**.

## Ανίχνευση μεταβολών και ανταλλαγή μηνυμάτων LSP

Ένας δρομολογητής ανιχνεύει τις αλλαγές τοπολογίας **περιοδικά**:

- ▶ συνθέστερη μέθοδος: κάθε κόμβος μεταδίδει περιοδικά **μηνύματα "χαιρετισμού"** (hello messages)
  - η μέθοδος μπορεί να ανιχνεύσει και την **κατάρρευση ενός γειτονικού κόμβου ή του αντίστοιχου συνδέσμου**

Κάθε κόμβος που αντιλαμβάνεται μια αλλαγή της κατάστασης των συνδέσμων του εκκινεί τη μετάδοση ενός μηνύματος LSP



## Διανομή μηνυμάτων LSP (1/2)

**Στόχος**: η αξιόπιστη παράδοση ενός μηνύματος LSP από έναν κόμβο σε **όλους τους κόμβους του δικτύου**

Ένας κόμβος  $v$  που θέλει να στείλει ένα μήνυμα LSP μεταδίδει το μήνυμα σε όλους τους γείτονές του

- ▶ στο μήνυμα εισάγει τη διεύθυνσή του ( $src$ ), έναν ακολουθιακό αριθμό ( $seq^{src}$ ) και το χρόνο ζωής του πακέτου
- ▶ ακολουθιακός αριθμός: ένας ακέραιος αριθμός που αυξάνεται κάθε φορά που ένας κόμβος στέλνει ένα πακέτο LSP

Ο ακολουθιακός αριθμός έχει διπλή χρησιμότητα:

- ▶ καθορίζει την "ηλικία" του πακέτου
- ▶ το ζεύγος τιμών  $\langle src, seq^{src} \rangle$  **προσδιορίζει μοναδικά** ένα πακέτο LSP μέσα στο δίκτυο



## Διανομή μηνυμάτων LSP (2/2)

Κάθε κόμβος  $w$  που λαμβάνει ένα μήνυμα LSP από έναν κόμβο  $v$  εξετάζει το ζεύγος τιμών  $\langle src, seq^{src} \rangle$ :

- ▶ προωθεί το πακέτο σε **όλους τους γείτονες εκτός από τον  $v$**  και αποθηκεύει τις τιμές  $\langle src, seq^{src} \rangle$  ( $seq_w(src) \leftarrow seq^{src}$ ) αν:
  - δεν έχει δει άλλο πακέτο από τον κόμβο  $src$
  - $seq^{src} > seq_w(src)$
- ▶ απορρίπτει το πακέτο αν  $seq_w^{src} \geq seq^{src}$ 
  - το πακέτο LSP είναι παλαιότερο από κάποιο άλλο που έλαβε προηγούμενα ο  $w$  από τον  $src$

Ο χρόνος ζωής (time to live) καθορίζει το μέγιστο χρόνο παραμονής ενός πακέτου LSP στο δίκτυο

- ▶ επιτρέπει τη διαγραφή παλιών πακέτων LSP από κόμβους που έχουν καταρρεύσει και την αποδοχή νέων όταν οι κόμβοι επανακάμψουν
- ▶ ένας κόμβος μπορεί να μεταδώσει ξανά ένα μήνυμα LSP με χρόνο ζωής μηδέν για να εξαναγκάσει τους άλλους κόμβους να διαγράψουν το συγκεκριμένο πακέτο



## Υπολογισμός διαδρομών (1/2)

Μετά την ανταλλαγή μηνυμάτων LSP όλοι οι κόμβοι έχουν λάβει πληροφορίες που **περιγράφουν ολόκληρο το δίκτυο**

Ο υπολογισμός των συντομότερων διαδρομών μπορεί να γίνει με τον **αλγόριθμο του Dijkstra**

- $s$ : ο κόμβος προέλευσης
- $N$ : σύνολο που περιέχει τους κόμβους του δικτύου
- $M$ : το σύνολο των κόμβων που έχουν εξεταστεί από τον αλγόριθμο
- $l(i, j)$ : το κόστος του συνδέσμου που συνδέει τους κόμβους  $i$  και  $j$
- $C(n)$ : το κόστος της διαδρομής από τον  $s$  στον  $n$

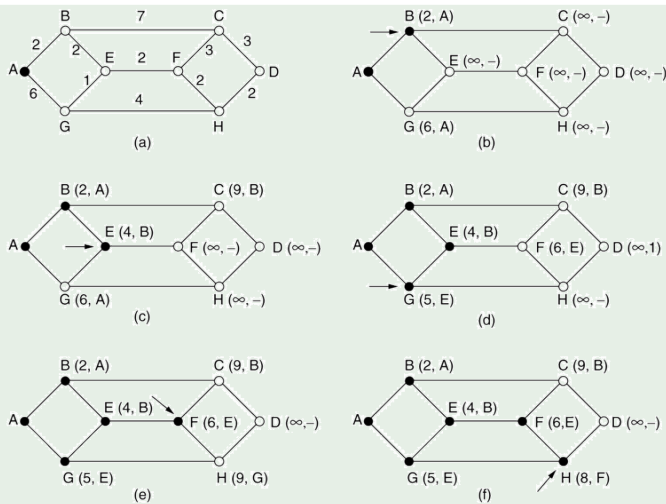
```

1: procedure DIKSTRA()
2:    $M = \{s\}$ 
3:   for each  $n \in N - \{s\}$  do
4:      $C(n) = l(s, n)$ 
5:   end for
6:   while  $\neq N$  do
7:      $M = M \cup \{w\} : C(w)$  is minimum,  $\forall w \in N - M$ 
8:     for each  $n \in N - M$  do
9:        $C(n) = \min\{C(n), C(w) + l(w, n)\}$ 
10:    end for
11:  end while
12: end procedure
  
```



## Υπολογισμός διαδρομών (2/2)

## Example



## Κριτική αλγόριθμου

## Πλεονεκτήματα:

- ▶ καταλήγει γρήγορα στον υπολογισμό των συντομότερων διαδρομών
- ▶ ανταποκρίνεται γρήγορα στις μεταβολές της τοπολογίας
- ▶ μπορούν να υπολογιστούν παραπάνω από μια διαδρομές για κάθε ζεύγος κόμβων
- ▶ ο σχηματισμός βρόχων είναι λιγότερο πιθανός
- ▶ μπορεί να υλοποιηθεί τόσο με **κατανεμημένο** όσο και με **κεντροποιημένο** τρόπο

## Μειονεκτήματα:

- ▶ το φορτίο δρομολόγησης που παράγεται από τη διάδοση των μηνυμάτων LSP είναι **σημαντικά υψηλό**
- ▶ σε κάθε κόμβο πρέπει να αποθηκεύεται **σημαντική ποσότητα πληροφοριών**



## Διάρθρωση

- 1 Ο ρόλος της Δρομολόγησης
- 2 Δρομολόγηση συντομότερης διαδρομής  
Μοντελοποίηση και βασικά ζητήματα υλοποίησης  
Δυναμική δρομολόγηση
- 3 Αλγόριθμοι συντομότερης διαδρομής  
Αλγόριθμοι διανυσμάτων απόστασης  
Αλγόριθμοι κατάστασης συνδέσμων
- 4 Ιεραρχική δρομολόγηση



## Αναγκαιότητα

- Καμία από τις τεχνικές δρομολόγησης που περιγράφηκαν **δεν παρουσιάζει καλές ιδιότητες κλιμάκωσης**
- ▶ το φορτίο δρομολόγησης αλλά και ο απαιτούμενος αποθηκευτικός χώρος αυξάνονται σημαντικά με το μέγεθος του δικτύου

Λύση: δρομολόγηση με τη χρήση ιεραρχίας (**ιεραρχική δρομολόγηση, hierarchical routing**)

Αντάλλαγμα: οι υπολογιζόμενες διαδρομές **ενδέχεται** να μην είναι οι **συντομότερες**



## Βασικές αρχές (1/3)

Το δίκτυο χωρίζεται σε μικρότερα τμήματα που καλούνται **περιοχές δρομολόγησης (routing domain)**

Οι περιοχές δρομολόγησης συνδέονται μεταξύ τους με τους **κόμβους πύλες (gateway nodes)**

- ▶ μια περιοχή δρομολόγησης μπορεί να περιέχει παραπάνω από έναν κόμβους πύλες

### Περιοχή δρομολόγησης

Μια περιοχή του δικτύου της οποίας οι δρομολογητές εφαρμόζουν ένα κοινό αλγόριθμο δρομολόγησης

⇒ Ο σχηματισμός των περιοχών δρομολόγησης εξαρτάται από τους διαχειριστές των δρομολογητών του δικτύου

## Βασικές αρχές (2/3)

Σε κάθε περιοχή δρομολόγησης μπορεί να χρησιμοποιηθεί οποιοσδήποτε από τους γνωστούς αλγόριθμους δρομολόγησης

- ▶ τα πρωτόκολλα δρομολόγησης στην περίπτωση αυτή καλούνται **ενδοπεριοχικά πρωτόκολλα (intradomain protocols)**

Οι **κόμβοι πύλες (gateways)** υλοποιούν επιπλέον ένα ξεχωριστό πρωτόκολλο δρομολόγησης που καλείται **διαπεριοχικής δρομολόγησης (inter-domain routing protocol)**

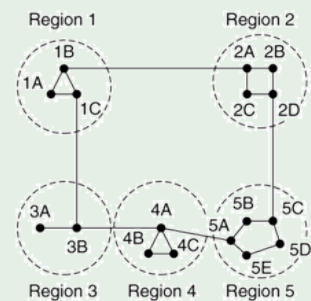
- ▶ **αφαίρεση:** οι κόμβοι πύλες σχηματίζουν ένα νέο δίκτυο στο οποίο εφαρμόζεται το πρωτόκολλο διαπεριοχικής δρομολόγησης

Τα δεδομένα προωθούνται:

- ▶ εντός των περιοχών δρομολόγησης με τη βοήθεια των ενδοπεριοχικών πρωτοκόλλων
- ▶ μεταξύ των περιοχών (κόμβων πύλης) με τη βοήθεια του διαπεριοχικού πρωτοκόλλου

## Βασικές αρχές (3/3)

### Ιεραρχική vs Παραδοσιακή δρομολόγηση



Full table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4