

Κινητός και Διάχυτος Υπολογισμός (Mobile & Pervasive Computing)

Δημήτριος Κατσάρος, Ph.D.

Χειμώνας 2005

Διάλεξη 9η

Ιστοσελίδα του μαθήματος

- http://skyblue.csd.auth.gr/~dimitris/courses/mpc_fall05.htm
- http://skyblue.csd.auth.gr/~dimitris/courses/mpc_fall05/
 - [books/](#)
 - [lectures/](#)
 - [papers/](#)
 - [proj_papers/](#)
 - [present_papers/](#)
- Τοποθετούνται οι διαφάνειες του επόμενου μαθήματος
- Τοποθετούνται τα research papers που αντιστοιχούν σε κάθε διάλεξη. Τα σημαντικά με πρόθεμα **MUST_BE_READ**

Περιεχόμενα

- Κινητά Ad Hoc δίκτυα
 - Το πρόβλημα των Broadcast Storms
 - **Μέθοδοι για 100% κάλυψη (reliable ή deterministic)**
 - **Connected dominating set (CDS)**
 - Clustering σε ad hoc δίκτυα

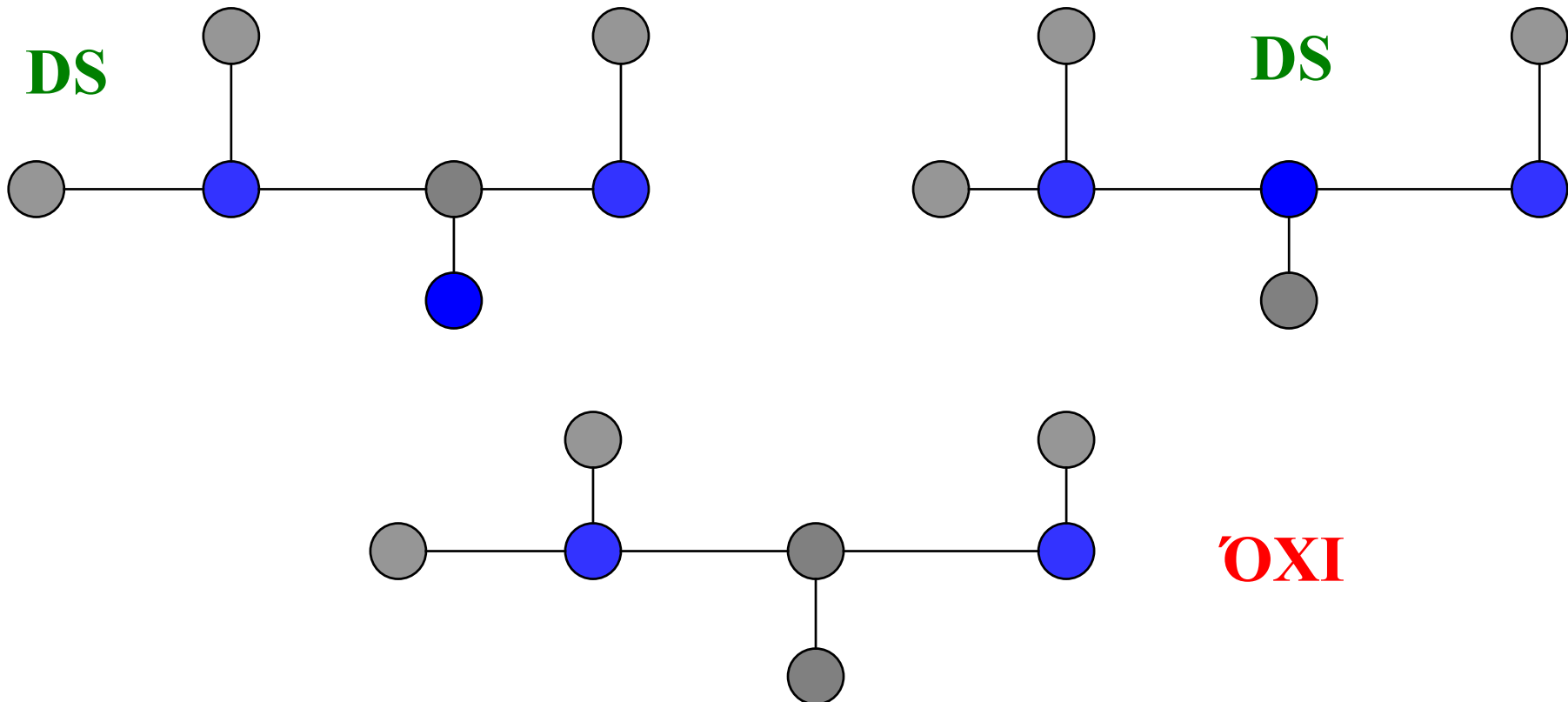
Εισαγωγικά

- Δεν υπάρχει σταθερή υποδομή
- Περιορισμένη ενέργεια
- Η δρομολόγηση (routing) απαιτεί γρήγορη σύγκλιση και μικρή επιβάρυνση σε επικοινωνία
- Γρήγορη προσαρμογή στις αλλαγές τις τοπολογίας

Κυρίαρχο σύνολο (Dominating Set-DS)

Ορισμός:

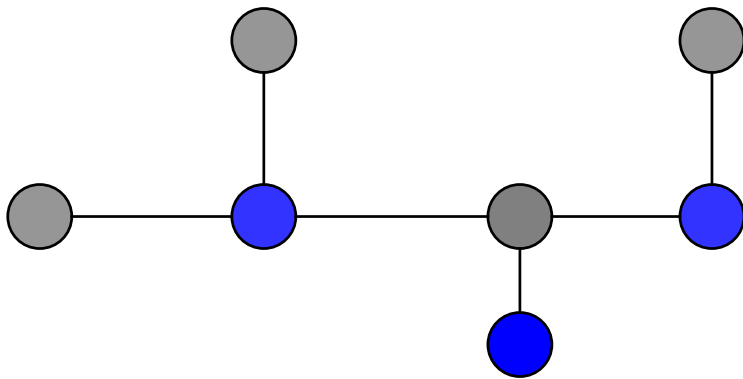
Είναι ένα υποσύνολο των κόμβων τέτοιο ώστε κάθε κόμβος είτε ανήκει στο υποσύνολο αυτό είτε είναι προσκείμενος (adjacent) σε τουλάχιστον έναν από τους κόμβους αυτού του υποσυνόλου.



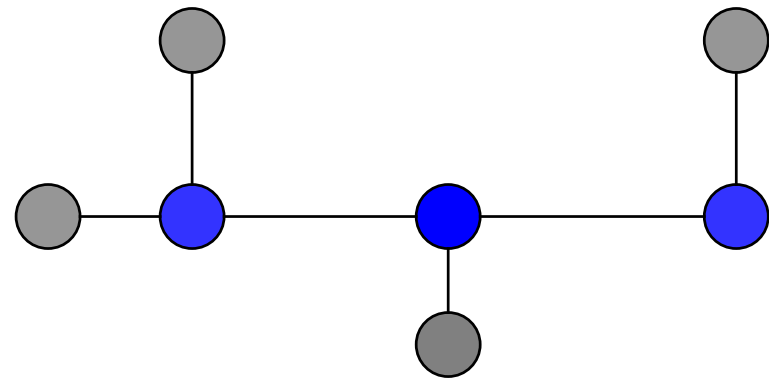
Συνδεδεμένο Κυρίαρχο σύνολο (Connected Dominating Set - CDS)

Ορισμός:

Είναι ένα DS του γραφήματος, τέτοιο ώστε το υπογράφημα που αντιστοιχεί στο DS να είναι συνδεδεμένο.



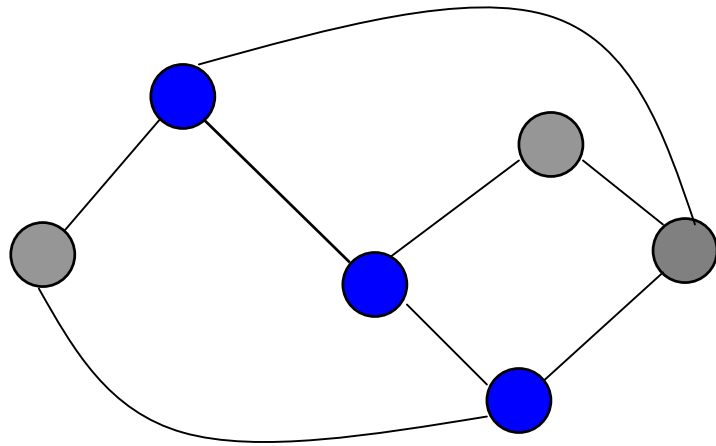
ΌΧΙ



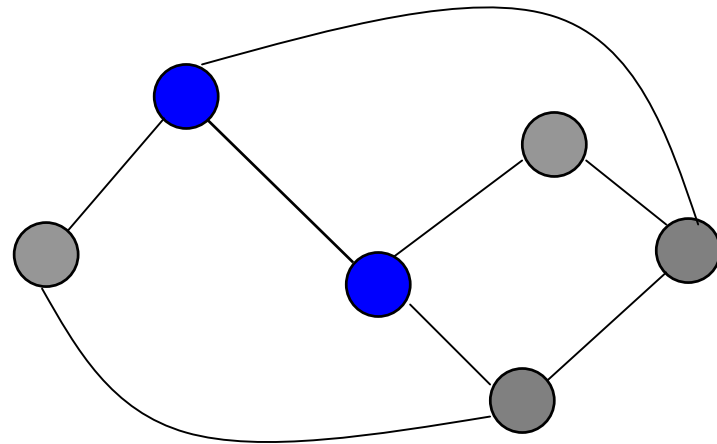
CDS

Ελάχιστο κυρίαρχο σύνολο (Min Connected Dominating Set - MCDS)

- Το MCDS είναι ένα CDS, τέτοιο ώστε να έχει το ελάχιστο μέγεθος (δηλ., το μικρότερο αριθμό κόμβων)



ΌΧΙ

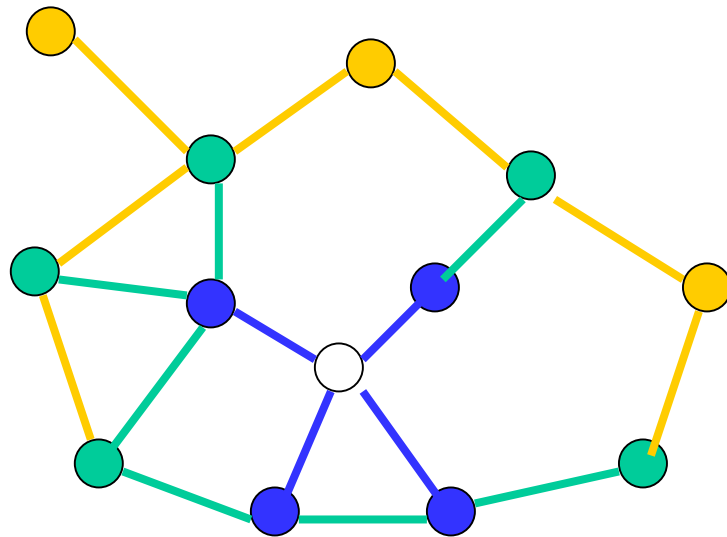


MCDS

Πολυπλοκότητες

- Η εύρεση του MCDS ανήκει στην κλάση προβλημάτων NP-complete, όταν έχουμε πλήρη (global) γνώση όλης της τοπολογίας του δικτύου
- Δεν μπορούμε να υποθέσουμε ότι κάθε κόμβος γνωρίζει όλο το δίκτυο. Για να συμβεί αυτό θα πρέπει να ανταλλάγουν $O(n^2)$ μηνύματα!
Απαγορευτικό κόστος επικοινωνίας
- Χρειαζόμαστε **τοπικούς (localized) αλγόριθμους**
- Δηλ., κατανεμημένες (**distributed**) προσεγγίσεις

Localized αλγόριθμοι



1-hop πληροφορία

2-hop πληροφορία

3-hop πληροφορία

- k-hop πληροφορία
 - Ανακαλύπτεται μετά από k γύρους από ανταλλαγές Hello μηνυμάτων
 - Τοπολογία και άλλες πληροφορίες
 - Συνήθως $k=1, 2$, ή 3
- Συσσώρευση πληροφορίας vs. Διάχυση πληροφορίας

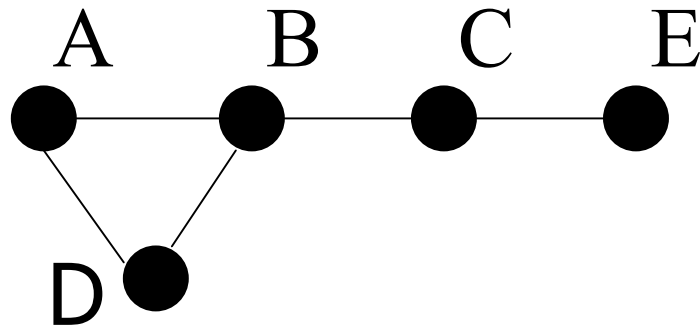
Αλγόριθμος των Wu & Li

- Διαδικασία **Μαρκαρίσματος** (marking)
 - Για την εύρεση του CDS
- Διώχνουμε (prune) τους πλεονάζοντες (redundant) κόμβους από το CDS
 - Για να ελαττώσουμε το μέγεθός του

Διαδικασία Μαρκαρίσματος

- Ορίζουμε ένα δίκτυο ως ένα γράφημα $G = (V, E)$
- Για κόμβο v : $N(v)$ είναι το “ανοιχτό” σύνολο των γειτόνων του δηλ., όλοι οι 1-hop γείτονες του v
- **BHMA 1:** Αρχικά, όλοι οι κόμβοι είναι **unmarked**
- **BHMA 2:** Κάθε κόμβος v **ανταλλάσσει το σύνολο των γειτόνων του $N(v)$** με όλους τους γείτονές του
- **BHMA 3:** **Mark v** εάν υπάρχουν τουλάχιστον 2 μη-συνδεδεμένοι (μεταξύ τους) γείτονες

Παράδειγμα



Μετά το βήμα 2:

A: N(B), N(D)

B: N(A), N(C), N(D)

C: N(B), N(E)

D: N(A), N(B)

E: N(C)

Το “ανοιχτό σύνολο γειτόνων” όλων των κόμβων:

$N(A) = \{B, D\}$

$N(B) = \{A, C, D\}$

$N(C) = \{B, E\}$

$N(D) = \{A, B\}$

$N(E) = \{C\}$

Ανάλυση διαδικασίας Μαρκαρίσματος

Θεώρημα:

Δεδομένου ενός $G = (V, E)$ το οποίο είναι συνδεδεμένο, αλλά όχι πλήρως συνδεδεμένο, το υποσύνολο των κορυφών V' , που παράγεται από την διαδικασία marking, σχηματίζει ένα dominating set του G .

Απόδειξη:

Στη διάλεξη.



Ανάλυση διαδικασίας Μαρκαρίσματος

Θεώρημα:

Το ελαττωμένο γράφημα $G' = G - V'$ είναι συνδεδεμένο.

Απόδειξη:

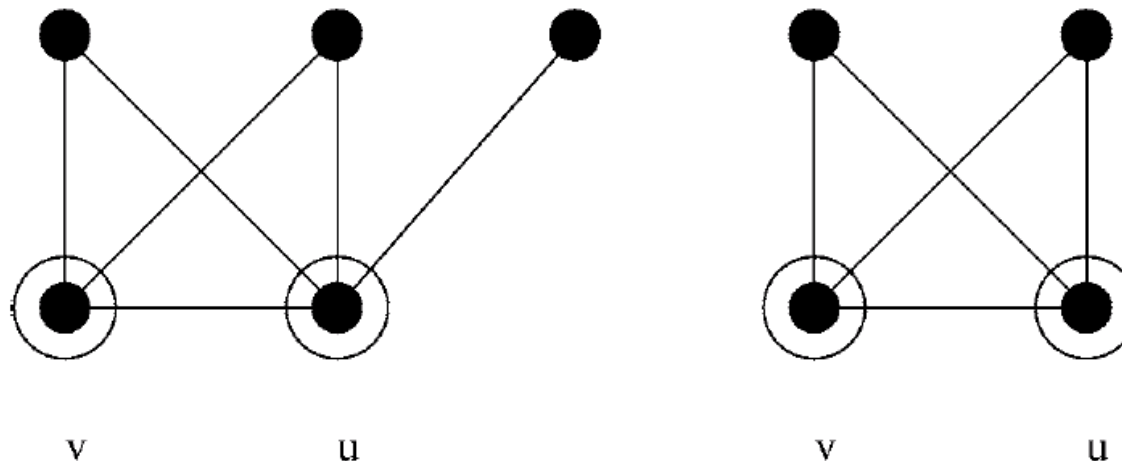
Στη διάλεξη.



Με βάση τα δυο αυτά θεωρήματα, η διαδικασία making παράγει ένα CDS

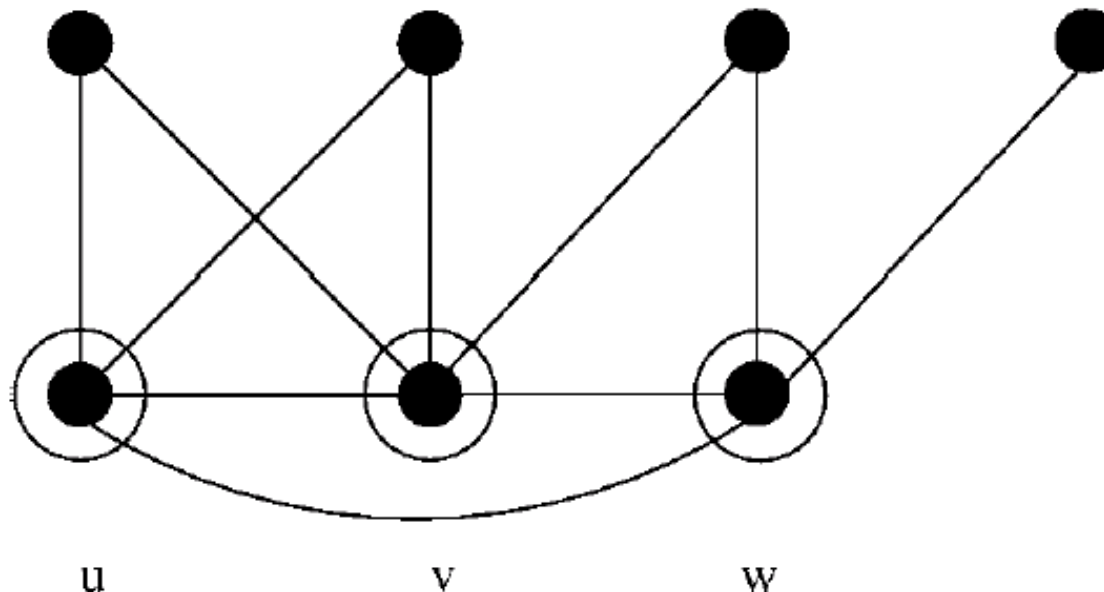
Αποπομπή πλεοναζόντων κόμβων από το CDS

- Αναθέτουμε ένα διακριτό id , $id(v)$ σε κάθε κόμβο $v \in G$
- Ορίζουμε το $N[v]$ ως το “κλειστό” σύνολο των γειτόνων (closed neighbor set) του v , δηλ., συμπεριλαμβάνεται και ο v .
- **Rule 1**: Εξετάζει δυο κόμβους v και u στο G .
Εάν $N[v] \subseteq N[u]$, and $id(v) < id(u)$, unmark v .



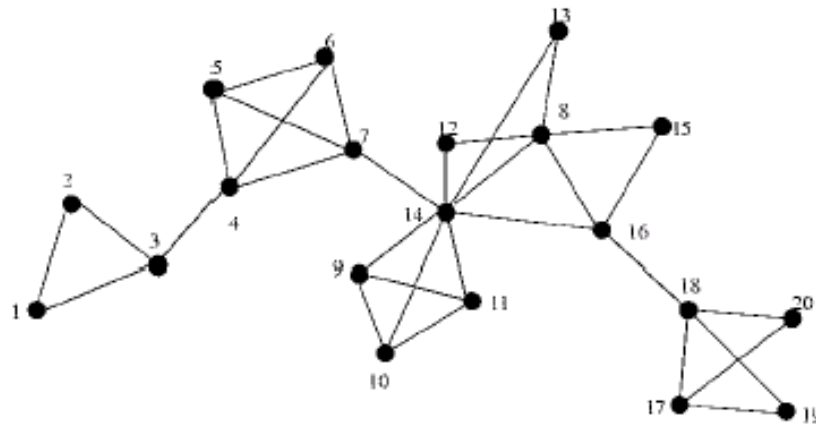
Αποπομπή πλεοναζόντων κόμβων από το CDS

- **Rule 2:** Υποθέστε ότι u και w είναι δυο marked γείτονες του marked κόμβου v στο G' . Εάν $N(v) \subseteq N(u) \cup N(w)$ στο G και ισχύει ότι $\text{id}(v) = \min\{\text{id}(v), \text{id}(u), \text{id}(w)\}$, τότε unmark τον κόμβο v .

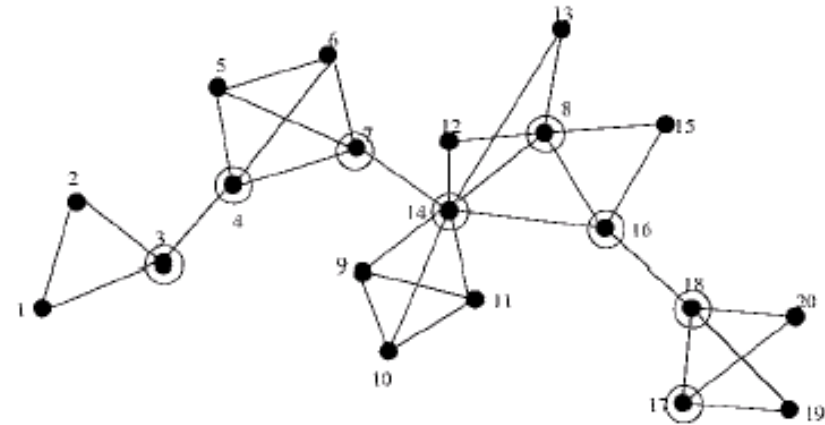


Παράδειγμα αλγορίθμου Wu & Li

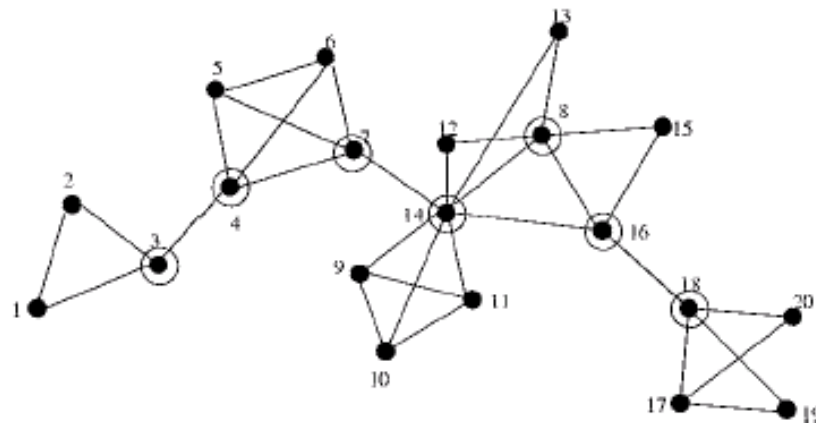
(a) Example graph



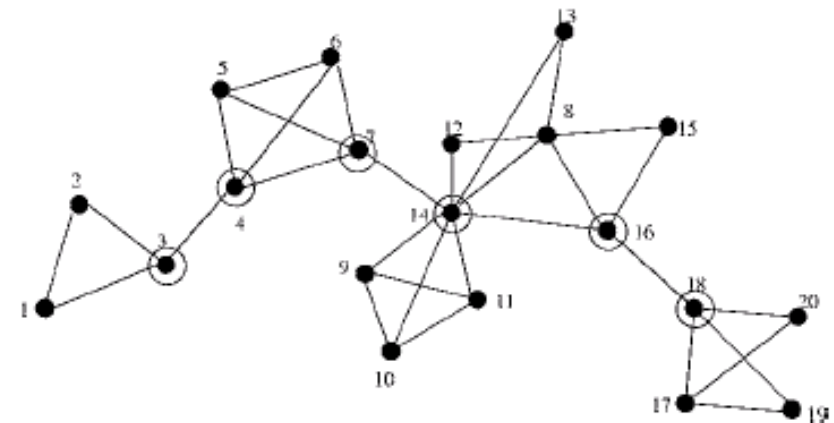
(b) Marked gateways without applying rules



(c) Marked gateways by applying Rule 1



(d) Marked gateways by applying Rule 2



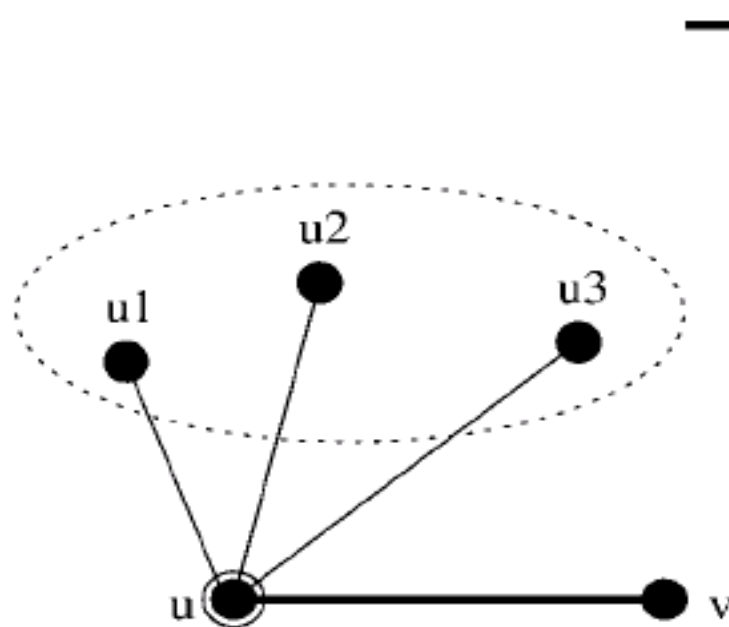
Ενημέρωση/Επανυπολογισμός CDS

Τοπολογικές αλλαγές του ασύρματου ad hoc δικτύου εξαιτίας:

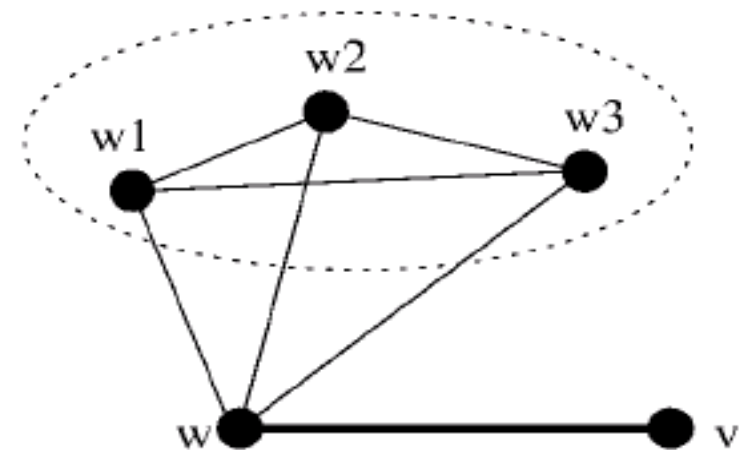
- Ενεργοποίηση (switch on) του mobile host
- Απενεργοποίηση (switch off) του mobile host
- Κίνηση του mobile host

Mobile host's switch on

- Όταν ο κόμβος v switches on, μόνο οι non-gateway γείτονές του πρέπει να ενημερώσουν την κατάστασή τους



(a) gateway neighbor u



(b) non-gateway neighbor w

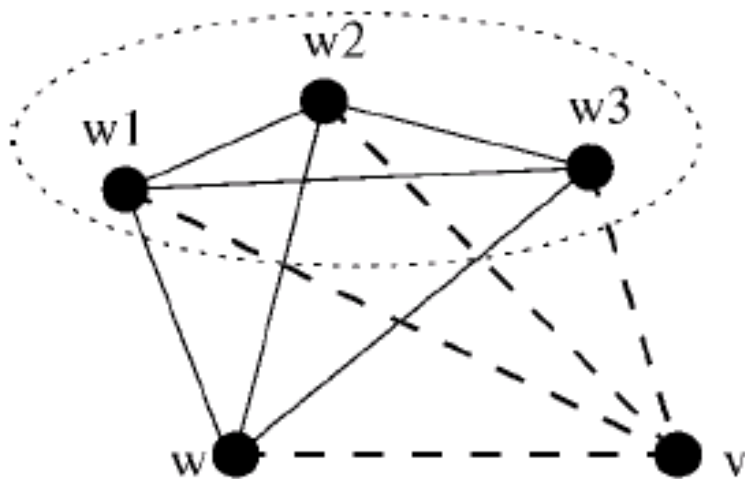
Αντίστοιχη διαδικασία marking

1. Ο κόμβος v broadcasts στους γείτονές του ότι ο ίδιος είναι switch on
2. Κάθε host $w \in v \cup N(v)$ ανταλλάσει το δικό του open neighbor set $N(w)$ με τους γείτονές του
3. Mark τον κόμβο v εάν υπάρχουν 2 μη-συνδεδεμένοι γείτονες
4. Mark κάθε non-gateway $w \in N(v)$ εάν έχει 2 μη-συνδεδεμένους γείτονες
5. Εφαρμόζουμε τον Rule 1 και Rule 2

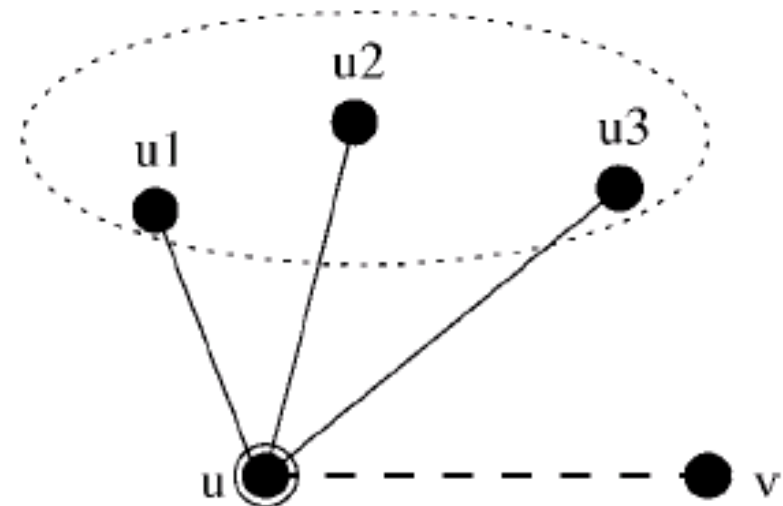
Mobile host's switch off

- Μόνο οι gateway γείτονες του switched off host χρειάζεται να αναθεωρήσουν την κατάστασή τους

--- broken link



(a) non-gateway neighbor w



(b) gateway neighbor u

Αντίστοιχη διαδικασία marking

1. Ο κόμβος v broadcasts στους γείτονές του ότι ο ίδιος θα γίνει switch off
2. Κάθε gateway γείτονας που ανήκει στο $N(v)$ ανταλλάσει το δικό του open neighbors set με τους γείτονές του
3. Unmark gateway εάν όλοι οι γείτονες είναι συνδεδεμένοι ανά δυο (pairwise connected)

Κίνηση του mobile host

Για να συγχρονιστεί η κίνηση του mobile host με τις ενημερώσεις των gateway:

- Πριν ο host v αρχίσει να κινείται, στέλνει ένα ειδικό σήμα **$\{id(v), Start\}$**
- Κατά τη διάρκεια της κίνησής του, στέλνει **$\{id(v), Heart_Beat\}$** ανά τακτά χρονικά διαστήματα
- Όταν σταματήσει, στέλνει **$\{id(v), Stop\}$**

Κίνηση του mobile host

- Όταν κάποιος host u λάβει και τους τρεις τύπους των σημάτων από τον κόμβο v , δεν κάνει τίποτε
- Εάν ο host u λάβει Start σήμα, αλλά δεν λάβει κάποιο Heart_Beat ή Stop σήμα, ο σύνδεσμος από τον u στον v είναι σπασμένος
- Εάν ο u λάβει Heart_Beat, και σήμα Stop χωρίς Start, ο u έχει έναν σύνδεσμο προς τον v

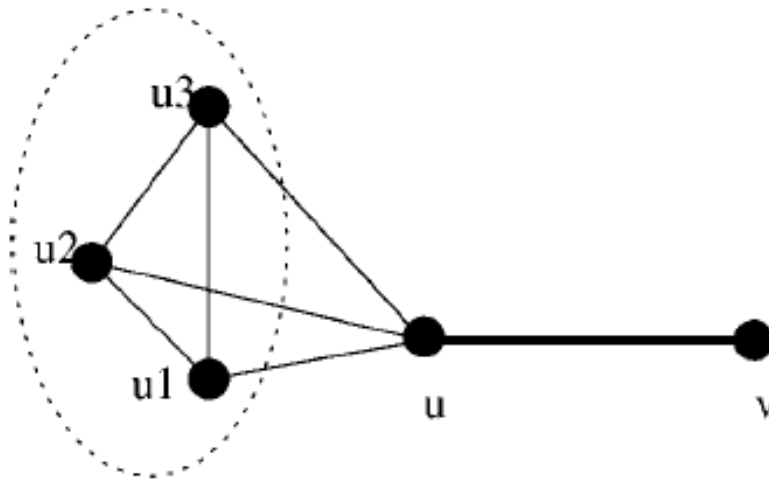
Αναγνώριση ενός νέου Link

Δυο τύποι κόμβων χρειάζεται να υπολογίσουν ξανά την gateway κατάστασή τους:

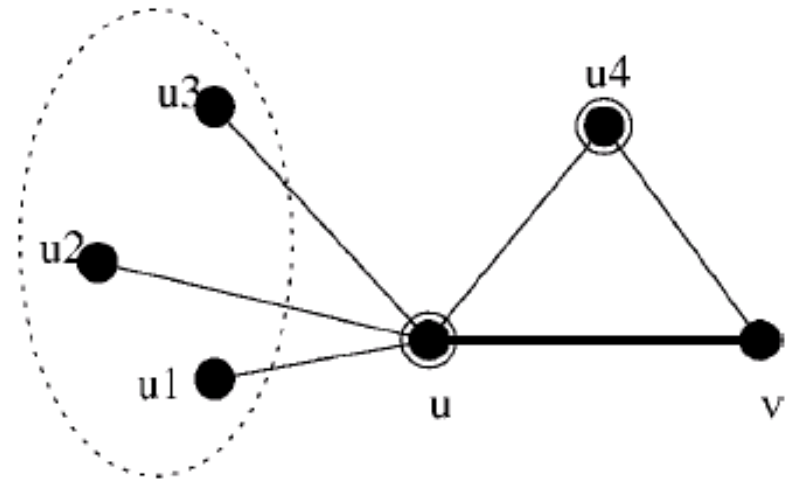
- Ο u , εάν αρχικά ήταν ένας non-gateway
- Οι κοινοί γείτονες του u και του v

Αναγνώριση ενός νέου Link

— new link



(a) non-gateway host u



(b) gateway host u

Figure 8. Mobile host u recognizes new link $\{u, v\}$.

- Εάν ο u θα κάνει mark τον εαυτό του ως gateway, θα εξαρτηθεί από τις συνδέσεις μεταξύ του v και των γειτόνων του u
- Ο $u4$ unmarks τον εαυτό του όταν το φτιάχνεται το uv link

Αντίστοιχη διαδικασία marking

1. Ο u ανιχνεύει ένα νέο σύνδεσμο v , ανταλλάσσει το open neighbor set με τους γείτονές του
2. Όταν λάβει το $N(u)$, ο gateway w κάνει unmark τον εαυτό του, εάν είναι κοινός γείτονας των u , v
3. Εάν ο u είναι gateway, δεν κάνει τίποτε
4. Εάν ο u είναι non-gateway και έχει 2 μη-συνδεδεμένους γείτονες, ο u κάνει mark τον εαυτό του
5. Εφαρμόζουμε τον Rule 1 και Rule 2

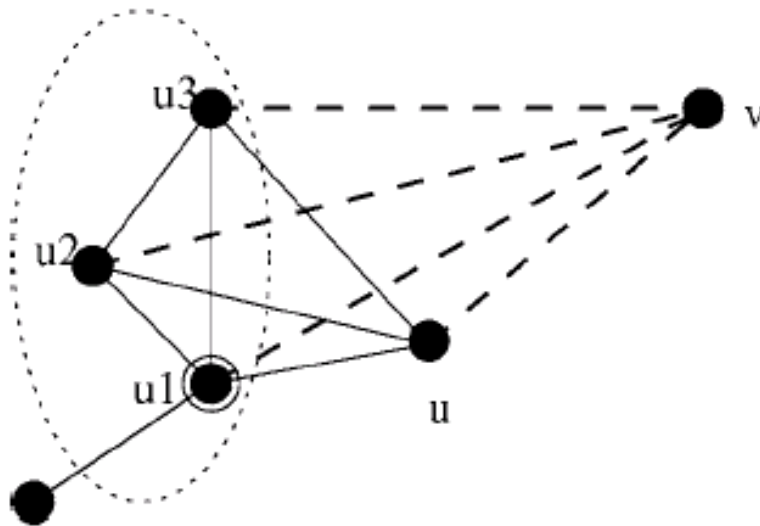
Αναγνώριση ενός broken Link

Δυο τύποι κόμβων χρειάζεται να υπολογίσουν ξανά την gateway κατάστασή τους:

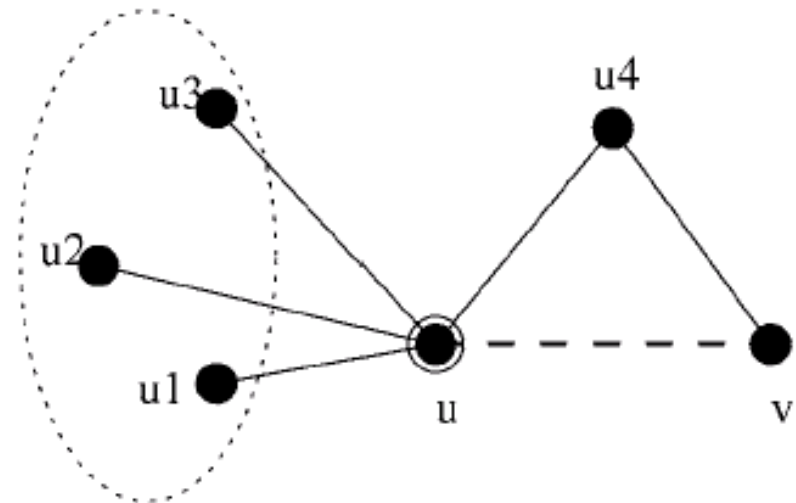
- Ο u , εάν αρχικά ήταν gateway
- Οι κοινοί γείτονες του u και του v

Αναγνώριση ενός broken Link

--- broken link



(a) non-gateway host u



(b) gateway host u

Figure 9. Mobile host u recognizes broken link $\{u, v\}$.

- Οι γείτονες του u είναι όλοι συνδεδεμένοι ανά δυο, έτσι ο u παραμένει
- Ο u_4 marks τον εαυτό του ως gateway

Αντίστοιχη διαδικασία marking

1. Ο u ανιχνεύει το broken link προς τον v , και ανταλλάσσει το $N(u)$ με τους γείτονές του
2. Εάν ο u είναι non-gateway, δεν κάνει τίποτε. Διαφορετικά, ο u θα γίνει unmark εάν οι γείτονές του είναι όλοι συνδεδεμένοι ανά δυο (pairwise connected)
3. Όταν λάβει το $N(u)$, ο non-gateway γείτονας w ξαναυπολογίζει τη δική του gateway status εάν είναι κοινός γείτονας των u και v
4. Εφαρμόζουμε τον Rule 1 και Rule 2

Εκτίμηση επίδοσης

- Μπορεί να φτιάξει trivial CDS
- Πολυπλοκότητα χρόνου: $O(\Delta^2)$
- Πολυπλοκότητα μηνυμάτων: $O(\Delta n)$
- Όχι εγγύηση ότι θα βρει το MCDS

Επανεκτίμηση της επίδοσης

- Πολυπλοκότητα χρόνου με χρήση των δυο κανόνων: $O(\Delta^3)$
- Πολυπλοκότητα μηνυμάτων : $\Theta(m)$
όπου $m = |E(G)|$
- Approximation factor: $n/2$
όπου $n = |V(G)|$

Απόδειξη του approximation factor

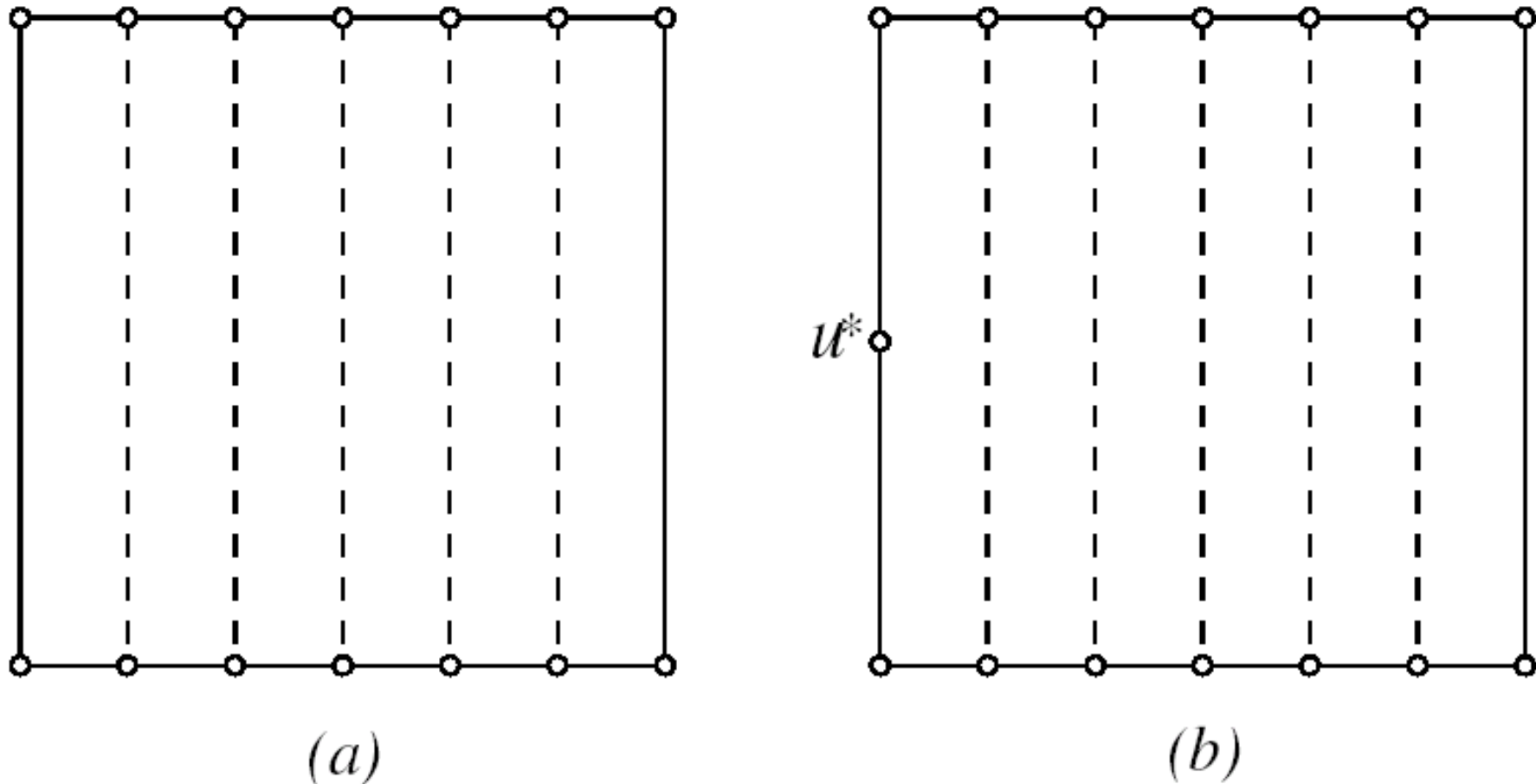


Fig. 3. Instance for which the CDS computed by Wu and Li's algorithm consists of all nodes but the MCDS consists of only two nodes.

Ο αλγόριθμος των P.-J. Wan et al.

Κάτω όριο

Θεώρημα:

Σε asynchronous ασύρματα ad hoc δίκτυα των οποίων ο unit-disk-graph είναι δακτύλιος, ο αριθμός των μηνυμάτων που θα στείλει οποιοσδήποτε κατανεμημένος αλγόριθμος για nontrivial CDS, είναι τουλάχιστον $\Omega(n \log n)$

Αλγόριθμος

Αποτελείται από δυο φάσεις:

1. Κατασκευή του Maximal Independent Set (MIS)
2. Dominating Tree

Κατασκευή του MIS

- Δεδομένο ένα rooted spanning tree T
- Rank ενός κόμβου είναι το ταξινομημένο ζεύγος του επιπέδου του και του ID του
- Ranking διαδικασία
- Color Marking

Διαδικασία Ranking

Δομή:

1. Κάθε κόμβος διατηρεί δυο τοπικές μεταβλητές x_1 , x_2
 - x_1 = # των γειτόνων των οποίων τα επίπεδα δεν έχουν αναγνωριστεί ακόμα (αρχικά είναι ο # των γειτόνων)
 - x_2 = # των παιδιών που δεν έχουν αναφέρει ακόμα την περάτωση (αρχικά είναι ο # των παιδιών)
2. Κάθε κόμβος διατηρεί μια levelList
 - Αποθηκεύει τα επίπεδα των γειτόνων του (αρχικά είναι κενή)
3. Κάθε κόμβος διατηρεί την τοπική μεταβλητή y
 - Αποθηκεύει τον αριθμό των lower-ranked γειτόνων

Διαδικασία Ranking

- Ρίζα ανακοινώνει το επίπεδό της 0 με broadcasting ενός μηνύματος LEVEL
- Όταν ληφθεί ένα μήνυμα LEVEL:
 - levelList =+ (sender's level, sender's rank)
 - x_1 - -
- Εάν sender = its parent στο T τότε
 - its level = sender's level + 1
 - Broadcast its LEVEL message
- Εάν $x_1 = 0$
 - $y = \#$ lower ranked γείτονες

Διαδικασία Ranking

- Εάν $x_2=0$ και its level is set
 - Μετάδοση ενός μηνύματος LEVEL-COMPLETE στον πατέρα
 - Εάν είναι non-leaf:
 - $x_2 = \#$ of children
 - Εάν είναι root:
 - $x_2 = \#$ of children
 - Τερματισμός
- Όταν ληφθεί μήνυμα LEVEL-COMPLETE:
 - $x_2 - -$

Διαδικασία Color Marking

- Αρχικά marked με white χρώμα
- Κάθε κόμβος διατηρεί blackList (ids of its black neighbors)
- Root marks black, broadcasts BLACK message
- Όταν ληφθεί μήνυμα ένα μήνυμα BLACK:
 - blackList += sender's ID
 - Εάν node = white
 - Mark itself gray
 - Broadcast GRAY message
- Όταν ληφθεί μήνυμα ένα μήνυμα GRAY:
 - Εάν sender's rank < its rank
 - A white node decrements y by 1
 - If y = 1 after update, marks itself black., broadcast BLACK message

Διαδικασία Color Marking

- Όταν ένα φύλλο γίνεται marked, στέλνει μήνυμα MARK-COMPLETE στον πατέρα του
- Όταν ληφθεί ένα μήνυμα MARK-COMPLETE:
 - x_2 - -
 - Εάν $x_2 = 0$ και not root:
 - Μεταδίδει μήνυμα MARK-COMPLETE στον πατέρα του
 - Εάν $x_2 = 0$ και root:
 - $x_1 = \#$ of its neighbors
 - Τερματισμός

Κατασκευή του Dominating Tree T^*

Δομή κάθε κόμβου:

- Διατηρεί μια τοπική boolean z
 - Αρχικά είναι 0, τίθεται στο 1 εάν join T^*
- Local variable parent
 - Αποθηκεύει ID of its parent in T^* , αρχικά κενή
- childrenList
 - ID of its children in T^* , αρχικά κενή

Ειδικά, η ρίζα στο T διατηρεί τη μεταβλητή $root$, και $degree$, αρχικοποιημένο στο 0

Κατασκευή του Dominating Tree T^*

- Επιλογή μια ρίζας για το T^* από το T
- Προσκάλεσε κόμβους για να ενωθούν στο T^*
- Οι internal κόμβοι του T^* σχηματίζουν ένα CDS

Επιλογή ρίζας για το T^*

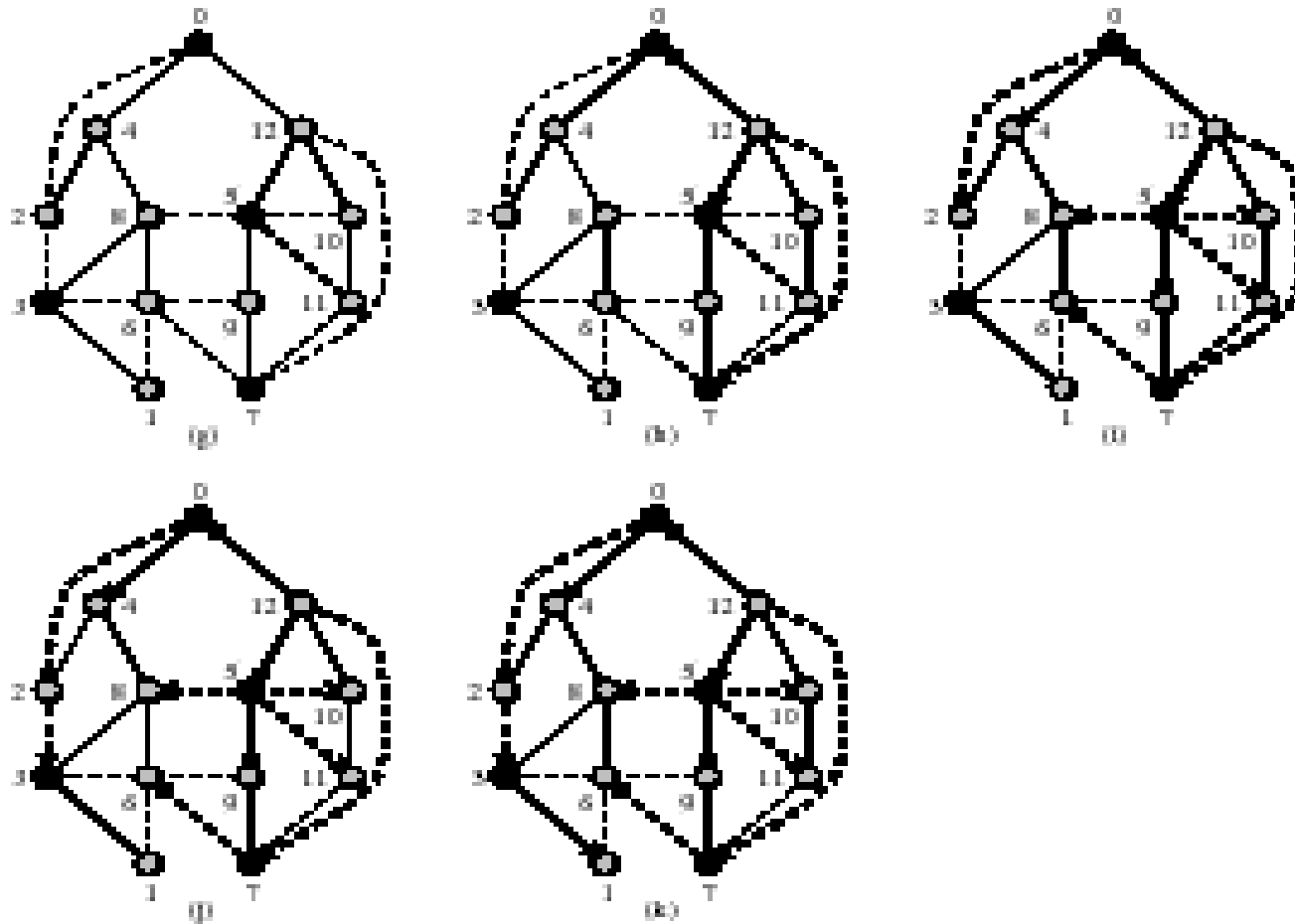
Βασική ιδέα: μια ρίζα του T^* είναι γείτονας της ρίζας του T που έχει τον μεγαλύτερο αριθμό από black γείτονες

- Ρίζα του T κάνει broadcast ένα μήνυμα QUERY
- Όταν ληφθεί το μήνυμα QUERY:
 - Ο κόμβος μεταδίδει στον αποστολέα ένα μήνυμα REPORT που περιέχει τον # των black γειτόνων
 - Ρίζα ελαττώνει x_1 κατά 1; Θέτει degree ίσο με #black γειτόνων εάν είναι μεγαλύτερη; Θέτει root variable ίσο με αυτό το id.
- Εάν $x_1 = 0$, στέλνει μήνυμα ROOT στον κόμβο που το id του είναι στην root variable

Πρόσκληση κόμβων

- Η ρίζα του T^* broadcasts μήνυμα INVITE2
- Όταν ληφθεί το μήνυμα INVITE2
 - Εάν color=black, $z=0$ τότε
 - Θέσε Set $z = 1$
 - Parent = sender's ID
 - Μετάδοση JOIN μηνύματος στον αποστολέα (sender)
 - Broadcast μήνυμα INVITE1
- Όταν ληφθεί μήνυμα INVITE1
 - Εάν color=gray, $z=0$ τότε
 - Θέσε $z = 1$
 - Parent = sender's ID
 - Μετάδοση JOIN μηνύματος στον αποστολέα
 - Broadcast μήνυμα INVITE2
- Όταν ληφθεί μήνυμα JOIN
 - Προσθήκη του sender's ID στην childrenList

Κατασκευή του Dominating Tree T^*



$$CDS = \{12, 0, 5, 7, 2, 3\}$$

Ανάλυση επίδοσης

- Approximation factor ≤ 8
- Πολυπλοκότητα χρόνου: $O(n)$
- Πολυπλοκότητα μηνυμάτων: $O(n \cdot \log n)$
- Nontrivial CDS

Περιεχόμενα

- Κινητά Ad Hoc δίκτυα
 - Το πρόβλημα των Broadcast Storms
 - Μέθοδοι για 100% κάλυψη (reliable ή deterministic)
 - **Clustering σε ad hoc δίκτυα**

Clustering ad hoc δίκτυα

- Επιθυμητό να δημιουργήσουμε μια αφηρημένη δομή πάνω από το δίκτυο, έτσι ώστε τοπικές αλλαγές να μην χρειάζεται να γίνουν γνωστές σε όλο το δίκτυο
- Με χρήση υποδομών που λέγονται **clusters**
- **Clustering**: η διαδικασία ορισμού αυτών των υποδομών μέσα σε όλη την τοπολογία του δικτύου
- Οι κόμβοι διακρίνονται σε:
 - **Clusterheads**
 - **Gateways**
 - **Ordinal nodes**

Απλοί αλγόριθμοι clustering

- Οι παλιότεροι αλγόριθμοι επέλεξαν τα clusterhead με βάση:
 - ID
 - το node degree
- Γενίκευση αυτών με ανάθεση σε κάθε κόμβο ενός βάρους (weight)

Distributed Clustering Algorithm - DCA (1/2)

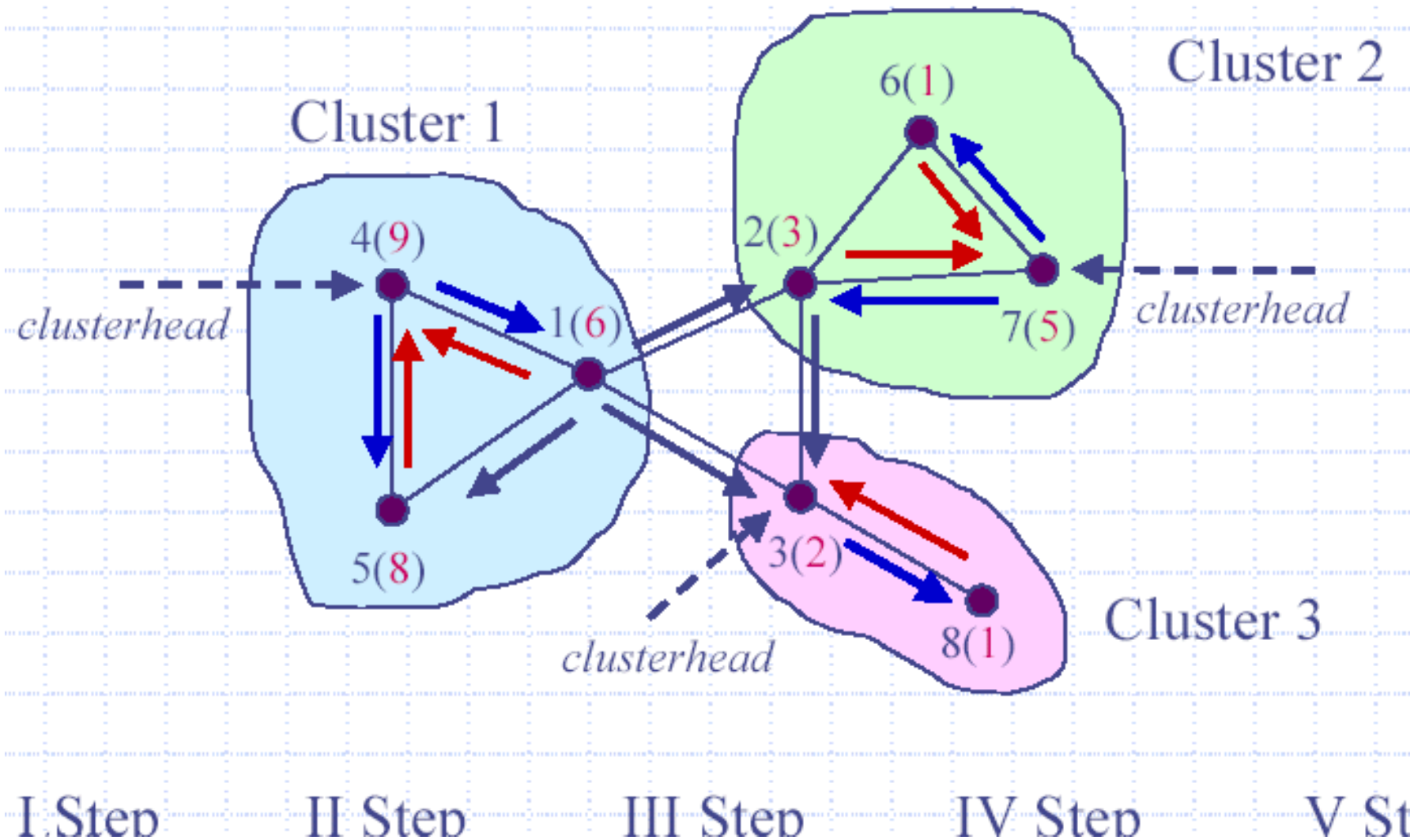
52

- Κάθε κόμβος έχει ένα ID και ένα $\text{weight} \geq 0$
- Δεν υπάρχουν δυο όμοια βάρη στο δίκτυο
- Στόχοι clustering
 - Κάθε ordinal κόμβος έχει ως γείτονα τουλάχιστον ένα clusterhead
 - Κάθε ordinal κόμβος συσχετίζεται με τον γειτονικό clusterhead που έχει το μεγαλύτερο weight
 - Δυο clusterhead δεν μπορεί να γειτνιάζουν
- Η τοπολογία δεν αλλάζει όσο εκτελείται ο αλγόριθμος

DCA (2/2)

- (Μόνο) Δυο μηνύματα:
 - **CH(v)**: Αποστέλλεται από ένα clusterhead v
 - **JOIN(u,t)**: Αποστέλλεται από ένα ordinary κόμβο u όταν “μπαίνει” στο cluster του clusterhead t
- Τρεις (απλές) διαδικασίες:
 - **Init (start up)**
 - **OnReceivingCH(v), OnReceivingJOIN(u,v)**

Παράδειγμα DCA



Πολυπλοκότητα του DCA

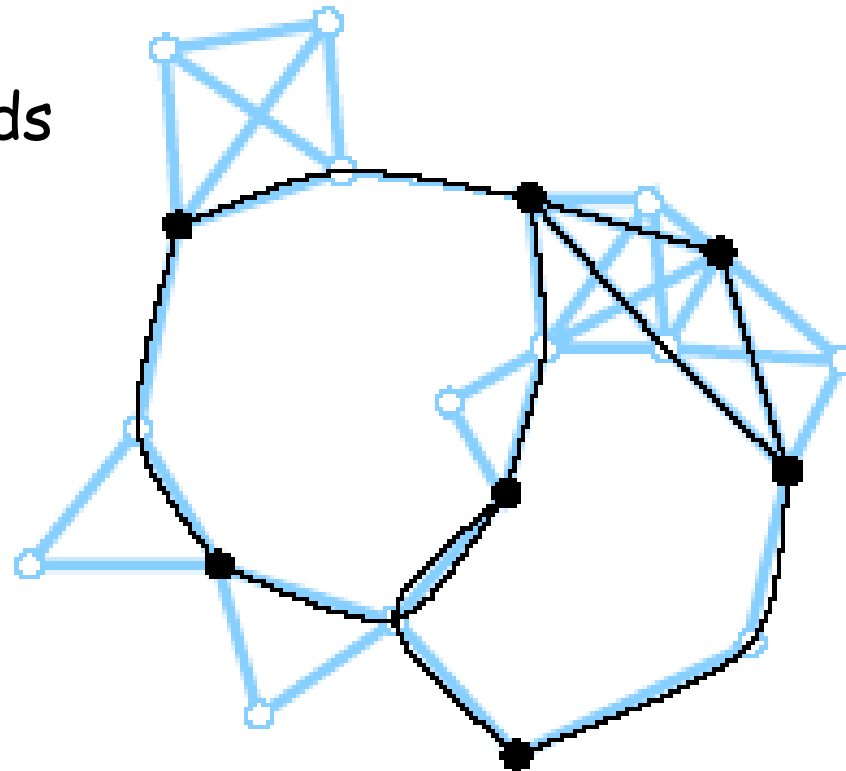
Θεωρήστε τα:

- $\tau : V \rightarrow \{1, 2, 3, \dots, 2k\}$
- $V =$ σύνολο κόμβων, $k =$ αριθμός των clusters
- **Πρόταση:** Κάθε κόμβος v στο V στέλνει ακριβώς ένα μήνυμα σε $\tau(v)$ βήματα
- **Πόρισμα 1:** η πολυπλοκότητα μηνυμάτων του DCA είναι $\mathbf{n} = |V|$
- **Πόρισμα 2:** ο DCA τερματίζεται ορθά το πολύ σε $2k$ βήματα ($\leq V$)

Παράδειγμα ad hoc δικτύου

Μαύροι κόμβοι: clusterheads

Μαύρες ακμές: συνδέσεις
μεταξύ των clusters

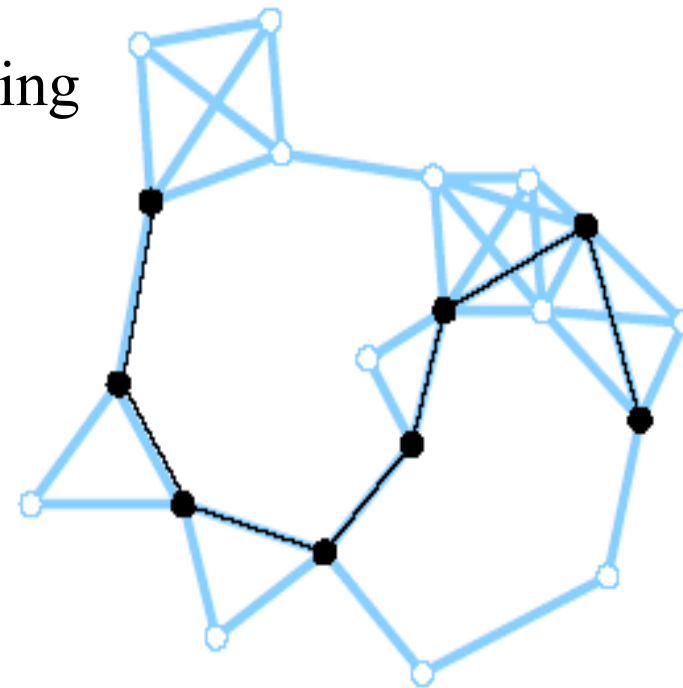


Μικρό dominating set

- Χρήση των κόμβων ενός dominating set ως **clusterheads**
- Αναθέτουμε κάθε κόμβο σ' εκείνο το cluster που αντιστοιχεί στην κορυφή που κυριαρχεί πάνω της
- Μικρός αριθμός από clusterheads για να απλοποιήσουμε τη δομή του δικτύου
- Ως γνωστό, η εύρεση του ελάχιστου dominating set είναι NP-complete

Connected dominating set

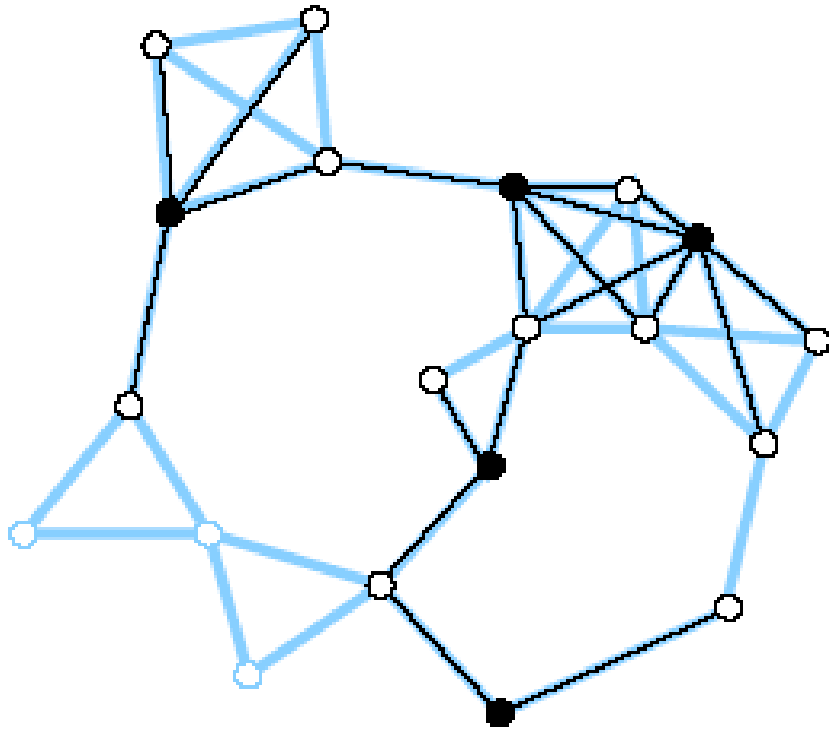
- Ένα connected dominating set (CDS) ενός γραφήματος G είναι ένα dominating set του οποίου το **induced γράφημα** είναι συνδεδεμένο
- Χρησιμοποιείται εύκολα για routing μηνυμάτων μεταξύ των clusters
- Η εύρεση ενός ελάχιστου CDS is NP-complete
- Ο αριθμός των clusters μεγάλος



Weakly induced subgraph

- Ελάττωση του αριθμού των clusters με την “χαλάρωση” της απαίτησης για συνδεσμικότητα
- Το υπογράφημα που είναι **weakly induced** από το $S (S \subseteq V)$ είναι το γράφημα $\langle S \rangle_w = (N[S], E \cap (N[S] \times S))$.
- $\langle S \rangle_w$ περιλαμβάνει τους κόμβους του S και όλους τους κόμβους που είναι γείτονες αυτών ως σύνολο κόμβων του $\langle S \rangle_w$
- Οι ακμές του $\langle S \rangle_w$ είναι όλες οι ακμές του G , που έχουν τουλάχιστον το ένα άκρο τους στο S

Παράδ. weakly induced subgraph

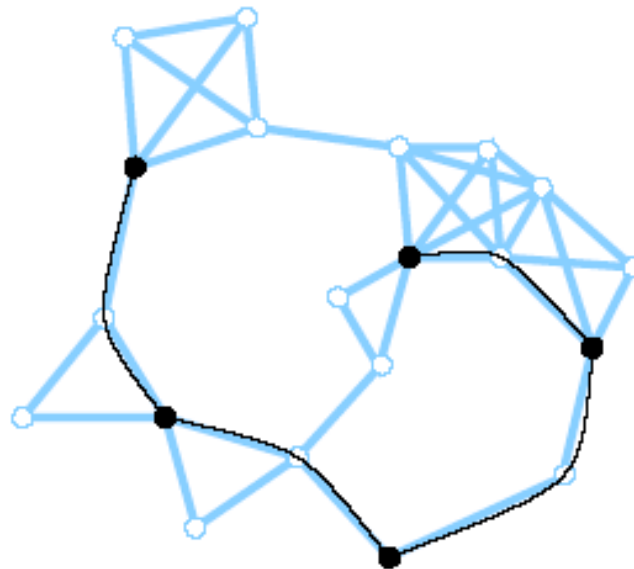


Σύνολο κόμβων: μαύροι κόμβοι

Σύνολο ακμών: μαύρες γραμμές

Weakly-connected dominating set

- Ένα υποσύνολο κόμβων S είναι ένα **weakly-connected dominating set (WCDS)**, εάν το S είναι dominating set και το $\langle S \rangle_w$ είναι συνδεδεμένο



Αλγόριθμ. για εύρεση μικρού WCDS

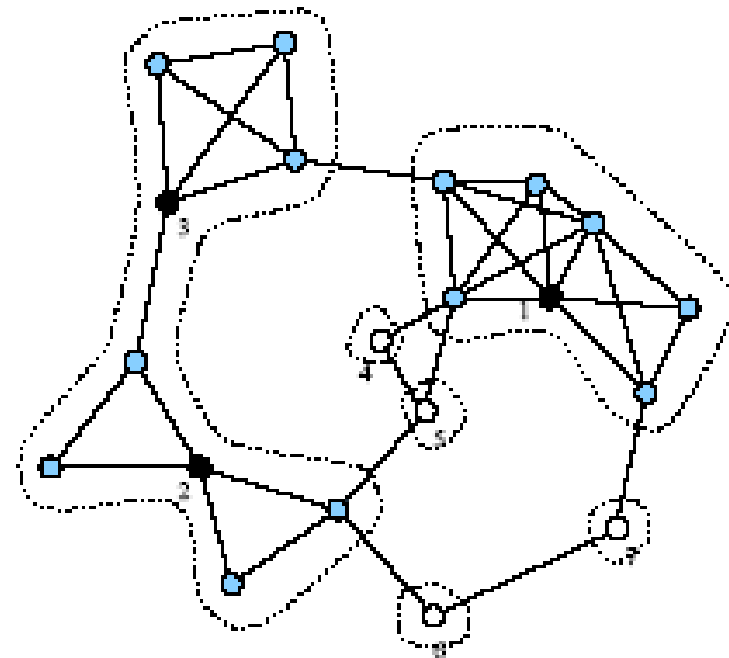
- **Αλγόριθμοι I και II**: Δυο centralized αλγόριθμοι
- **Αλγόριθμοι III και IV**: Distributed υλοποιήσεις των αλγορίθμων I και II
- **Αλγόριθμος V**: Distributed Asynchronous προσέγγιση

Αλγόριθμος I (περίληψη)

- Δεδομένου ενός γραφήματος $G=(V,E)$, κάθε κόμβος συσχετίζεται με ένα χρώμα (*white*, *gray*, ή *black*)
- Όλοι οι κόμβοι αρχικά είναι *white*
- Σε κάθε επανάληψη, ο αλγόριθμος χρωματίζει *black* έναν *white* ή *gray* κόμβο καθώς επίσης χρωματίζει *gray* όλους τους γειτονικούς του κόμβους
- Στο τέλος, οι *black* κόμβοι συνιστούν ένα *weakly-connected dominating set*

Ορολογία: η έννοια “piece”

- Piece αναφέρεται σε μια επιμέρους υποδομή του γραφήματος
- Ένα white piece είναι απλά ένας white κόμβος
- Ένα black piece περιέχει ένα maximal σύνολο από black κόμβους των οποίων το weakly induced υπογράφημα είναι συνδεδεμένο συν όποιοι προσκείμενοι gray κόμβοι



Τα pieces σηματοδοούνται
με dotted περιοχές

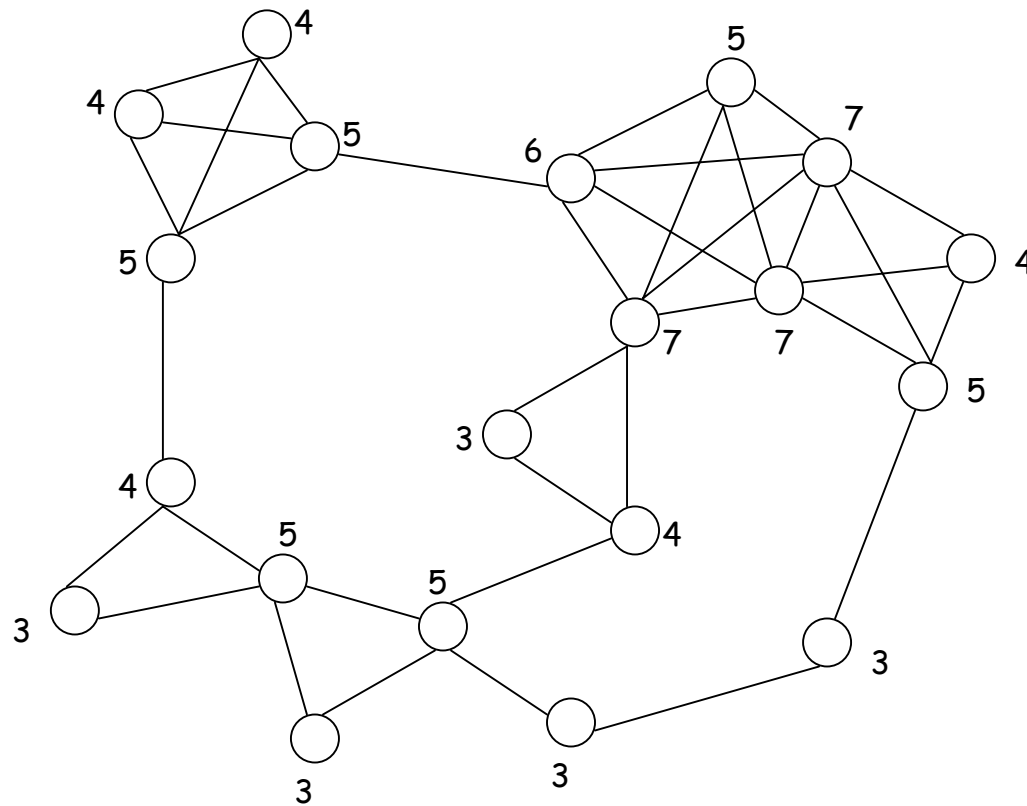
Ορολογία: η έννοια “improvement”

- Το improvement ενός (non-black) κόμβου u είναι ο αριθμός των pieces που θα συνενώνονταν σε ένα μόνο black piece εάν ο u χρωματιζόταν black
- Στο τελευταίο παράδειγμα, χρωματίζοντας τον κόμβο 5 black θα συνένωνε 4 pieces, ενώ χρωματίζοντας black τον κόμβο 4 θα συνένωνε 3 pieces

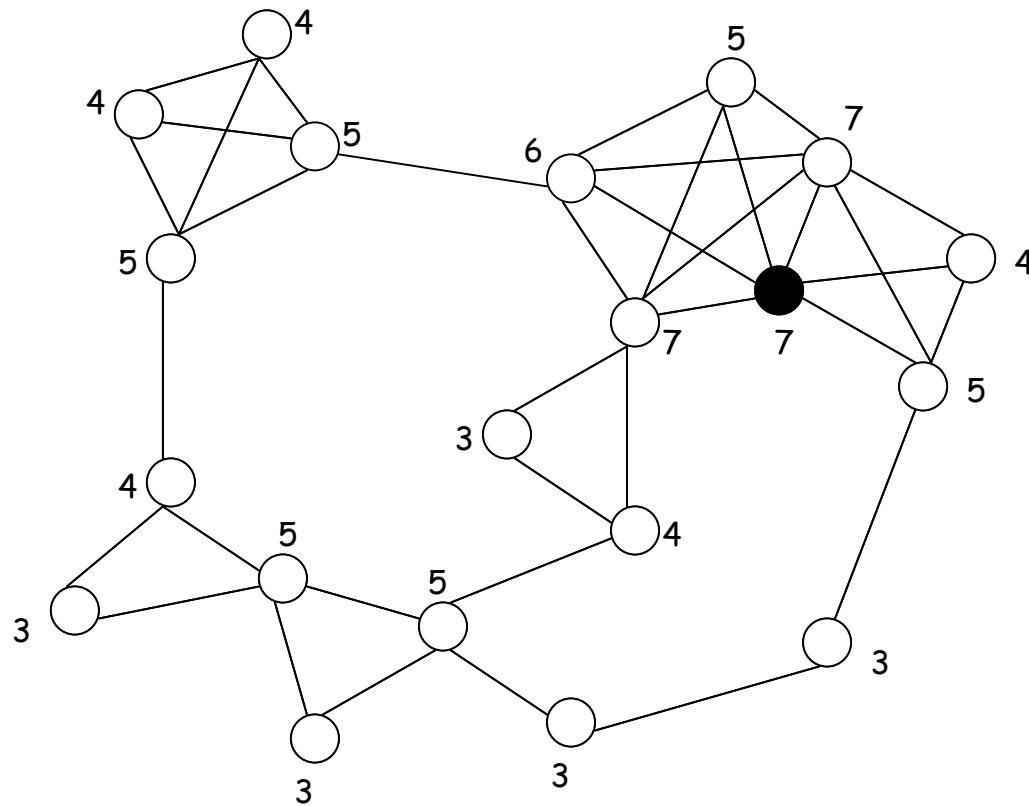
Αλγόριθμος I (λεπτομέρειες)

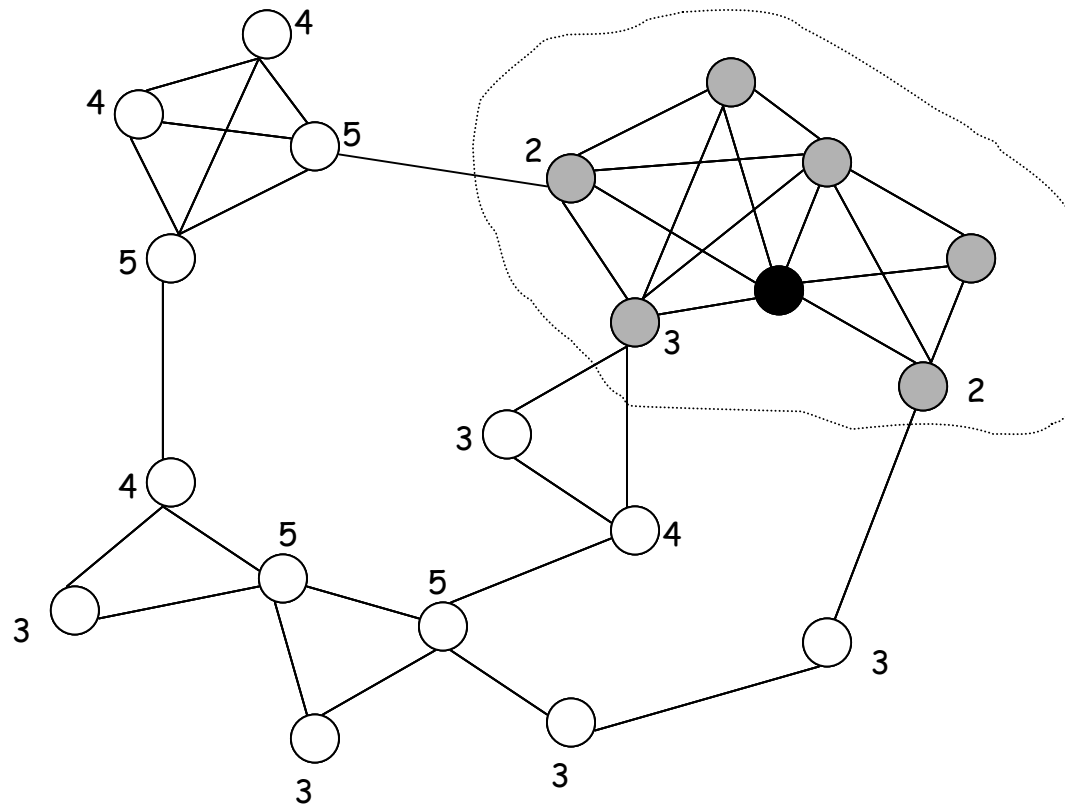
- Σε κάθε επανάληψη, ο αλγόριθμος επιλέγει ένα μόνο white ή gray κόμβο για να χρωματίσει black
- Η επιλογή γίνεται με άπληστο τρόπο: a επιλέγεται ο κόμβος με το μεγαλύτερο improvement
- Μέχρι να απομείνει μόνο ένα piece

Αρχικά όλοι οι κόμβοι είναι white

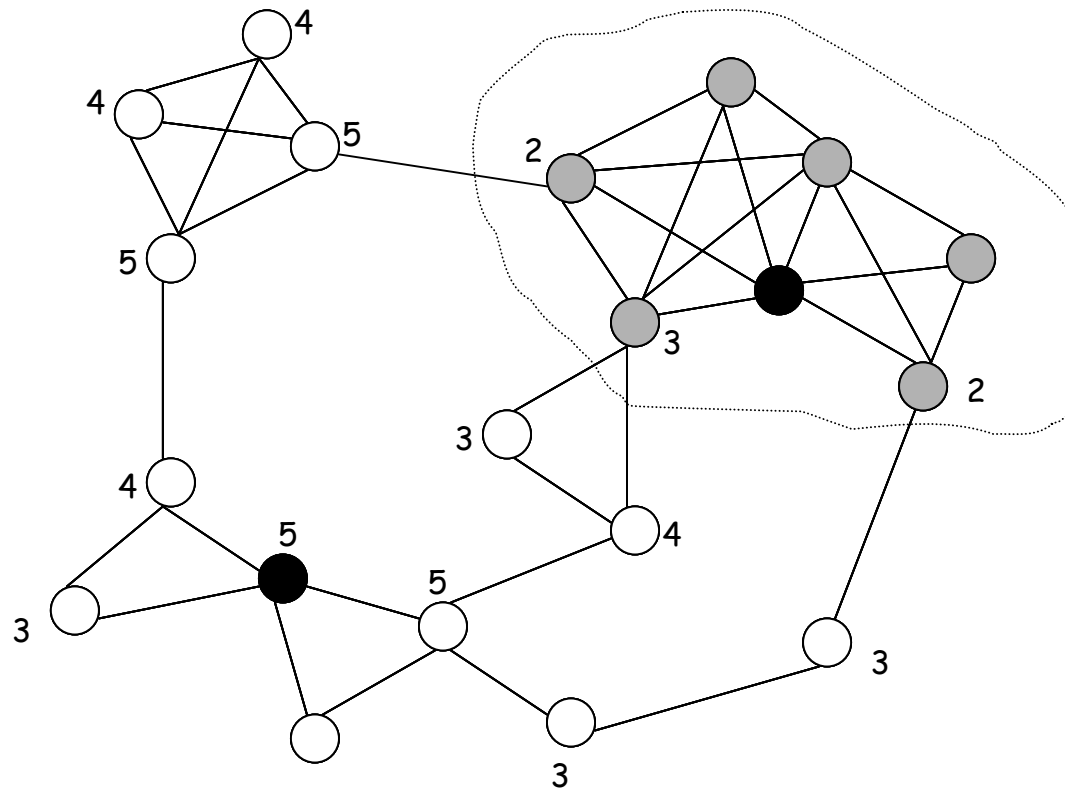


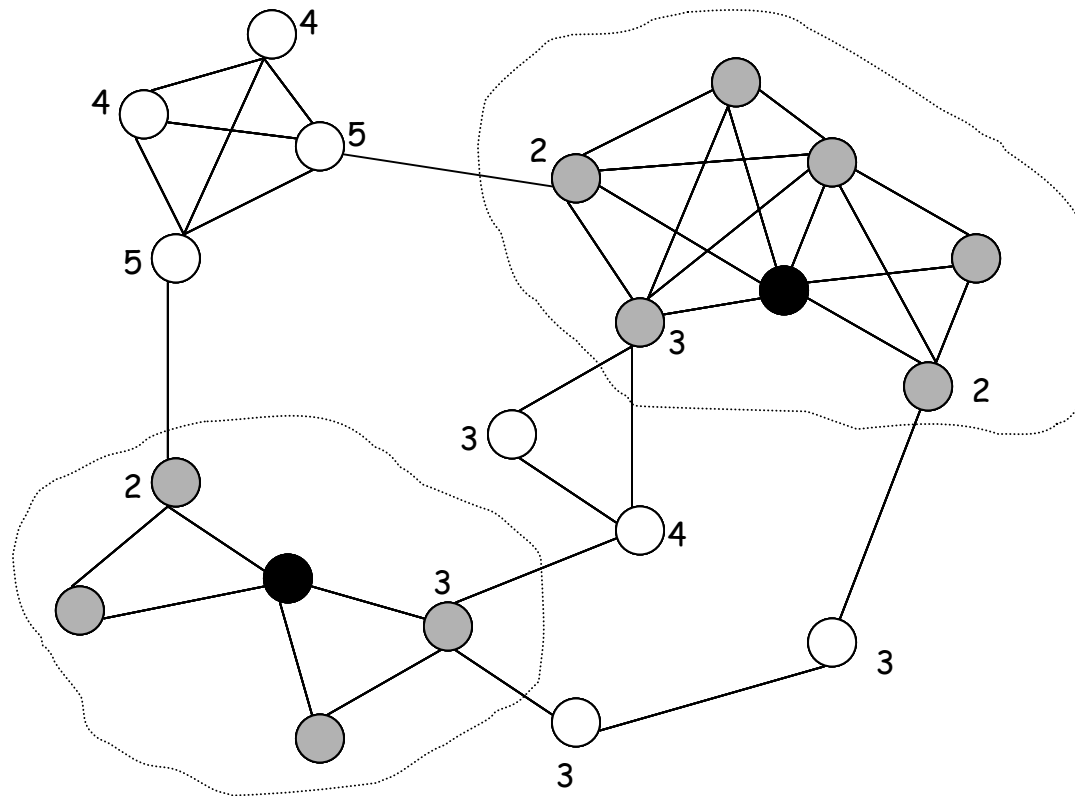
Πρώτη επανάληψη



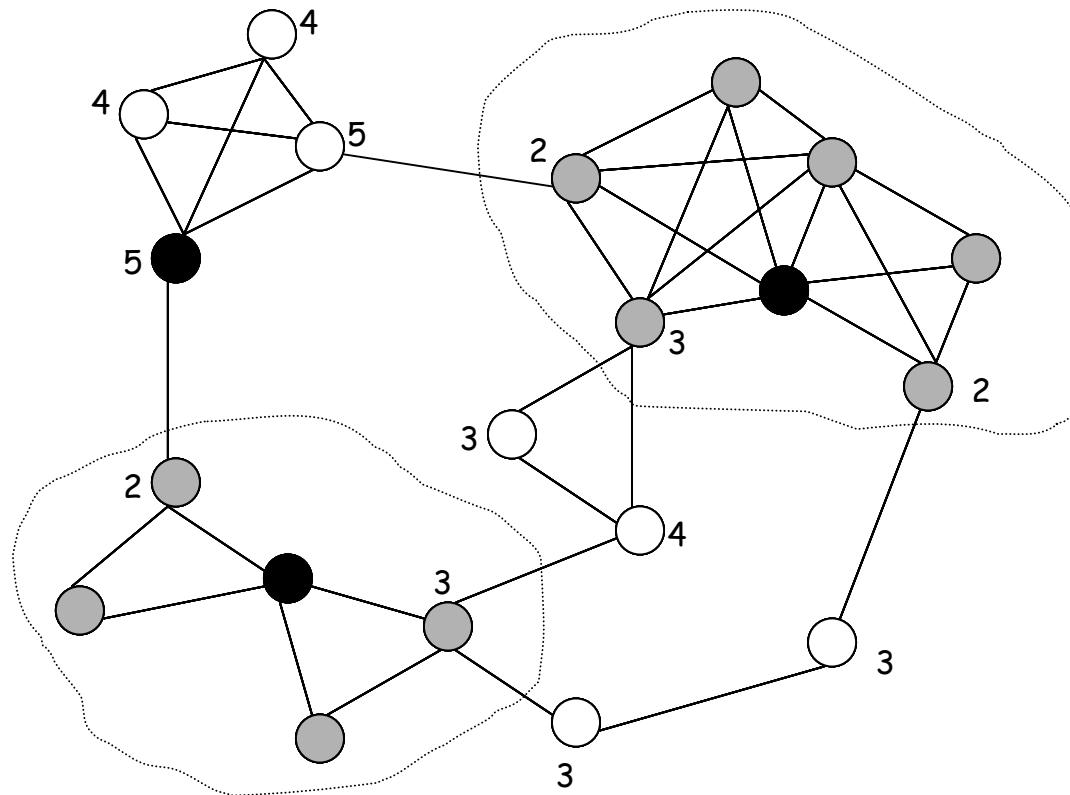


Δεύτερη επανάληψη

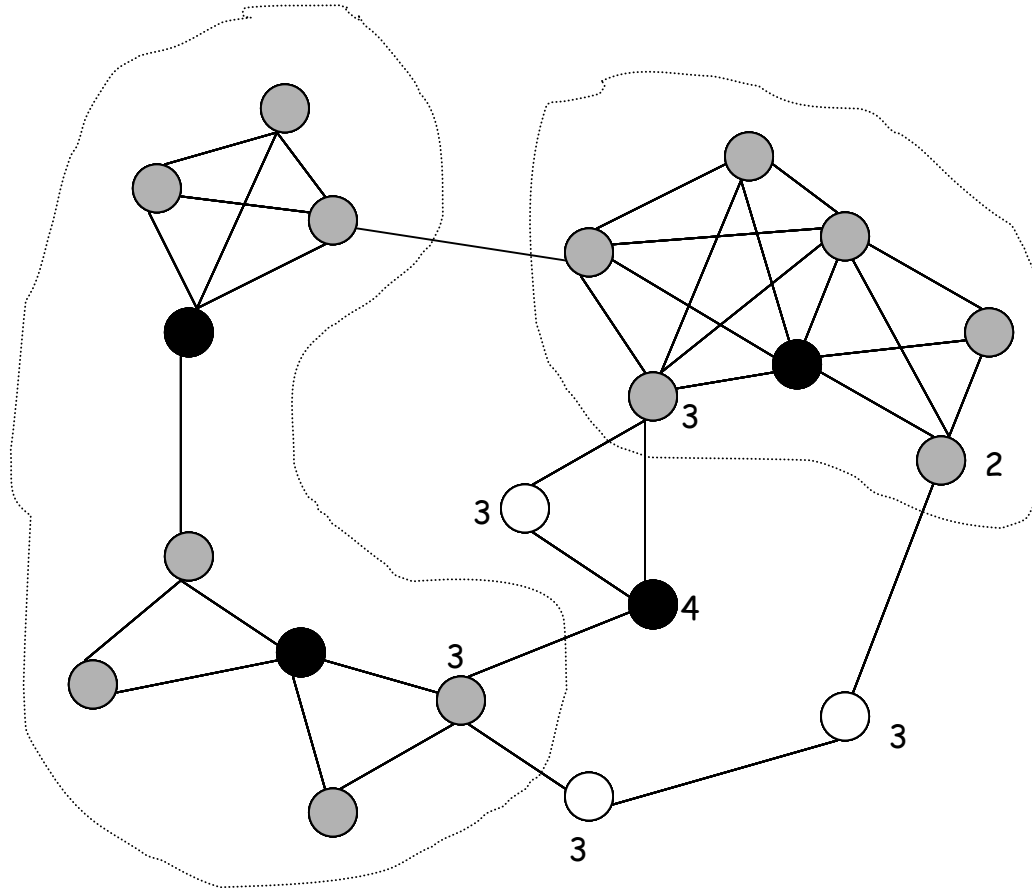


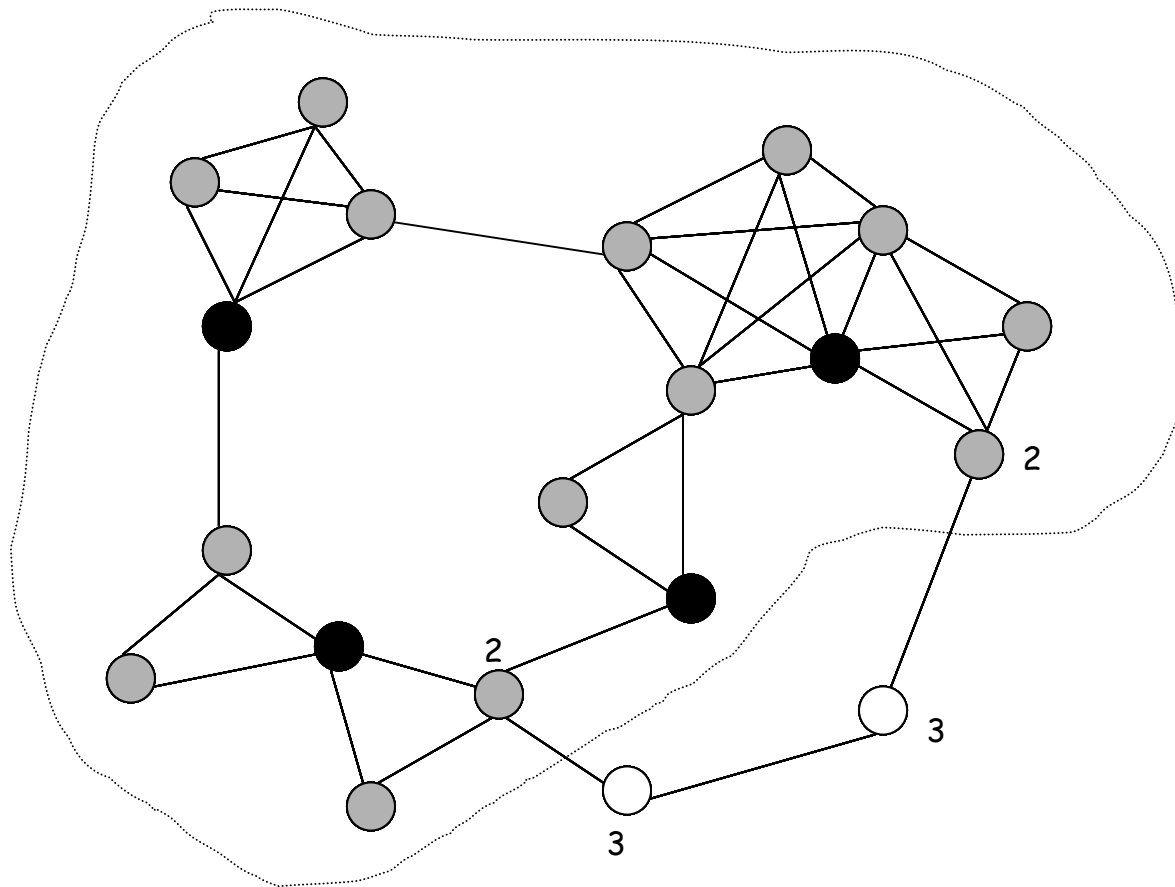


Τρίτη επανάληψη

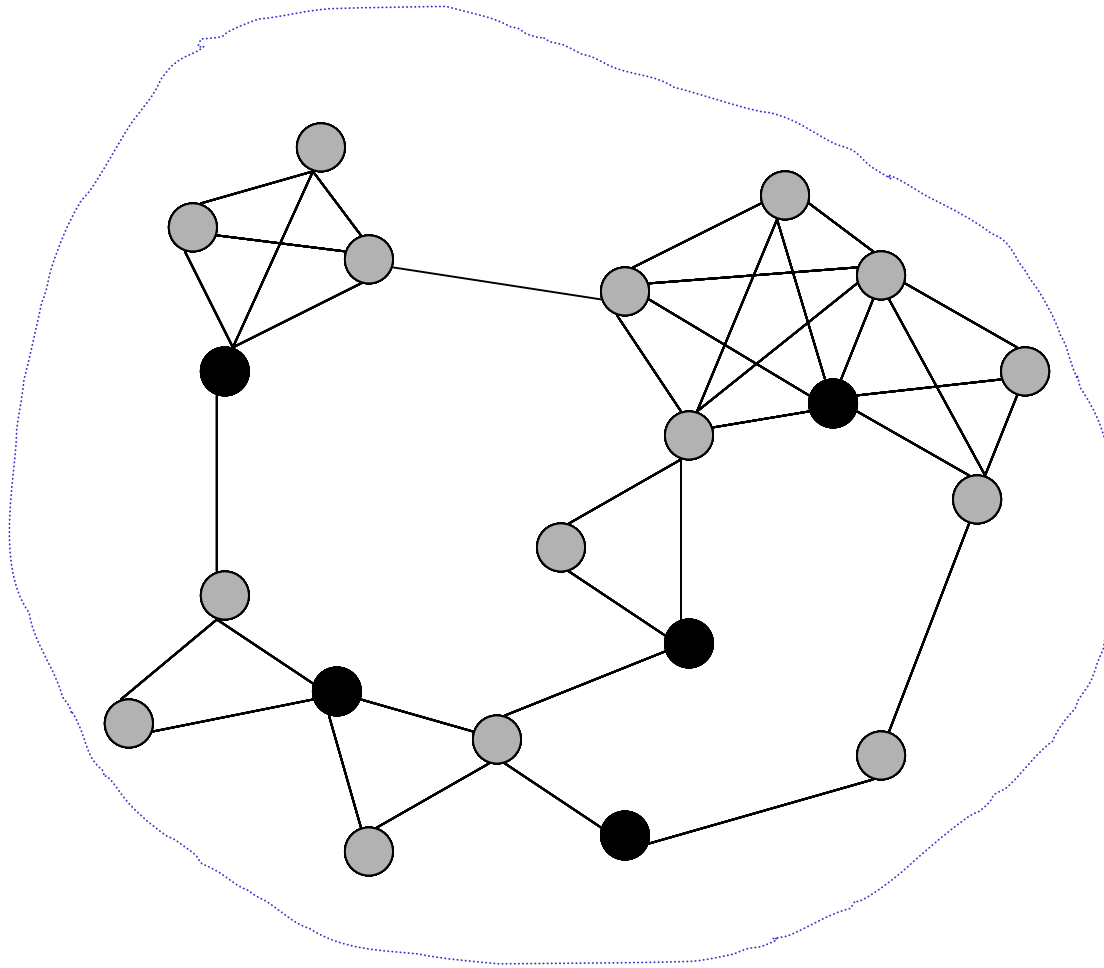


Τέταρτη επανάληψη





Τελευταία επανάληψη



Επίδοση του WCDS

- Το μέγεθος του weakly-connected dominating set που σχηματίζεται από τον Αλγόριθμο I είναι το πολύ $(\ln \Delta + 1) |\text{OPT}|$
- OPT συμβολίζει το ελάχιστο μέγεθος του weakly-connected dominating set για τον G
- Δ συμβολίζει το μέγιστο βαθμό του G

Αλγόριθμος II

- Ο αλγόριθμος ξεκινά με την επιλογή ενός τυχαίου κόμβου του G για να τον χρωματίσει black
- Στις επόμενες επαναλήψεις, επιλέγεται να χρωματιστεί black ο υποψήφιος κόμβος με τον μεγαλύτερο αριθμό white κόμβων στην γειτονιά του
- Το μέγεθος του weakly-connected dominating set είναι το πολύ $(\lg \Delta + 2) |OPT|$

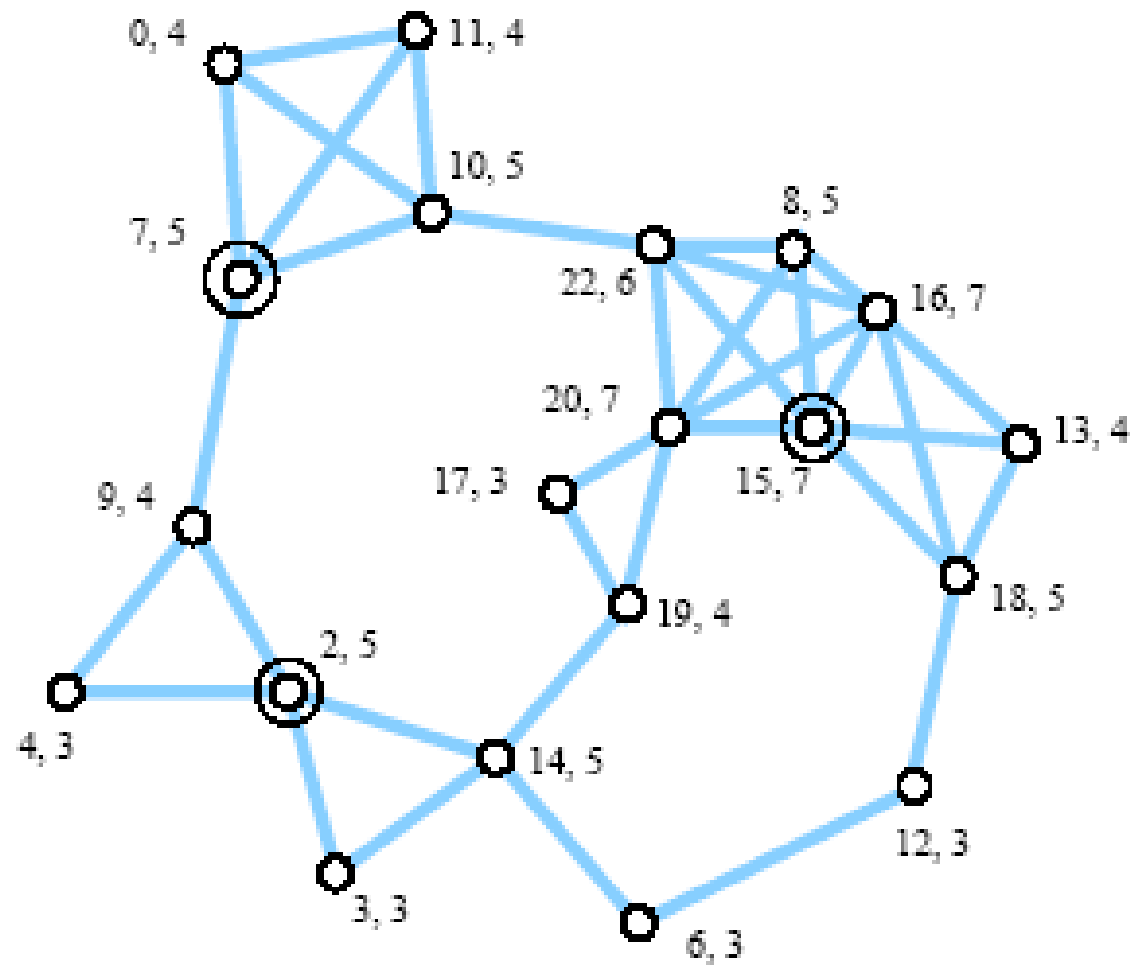
Αλγόριθμοι III και IV

- Σε ad hoc δίκτυα, ο mobile host δεν γνωρίζει τη δομή του δικτύου πέρα από τη γειτονιά της
- Distributed εκδόσεις των Αλγορίθμων I και II
- Εξακολουθούν να είναι κληρονομικά ακολουθιακοί, αφού μόνο ένας κόμβος μπορεί να χρωματιστεί black σε κάθε επανάληψη

Αλγόριθμος V

- Πλήρως distributed προσέγγιση
- Επέκταση πολλαπλών black pieces παράλληλα
- Σε κάθε επανάληψη, κάθε piece υπολογίζει τους δικούς της υποψήφιους
- Ένας υποψήφιος κόμβος ήταν είτε gray κόμβος ή white κόμβος προσκείμενος σε κάποιο gray κόμβο
- Κάθε piece επιλέγει από τους δικούς της υποψήφιους κόμβους τον υποψήφιο με τη μεγαλύτερη improvement και τον χρωματίζει black καθώς και τους γείτονές τους gray

Πρώτη επανάληψη



Δεύτερη επανάληψη

