Connected Dominating Set and Distributed Algorithm

”Connected Domination in Multihop Ad Hoc Wireless Networks”

Dr. Rajiv Misra
Associate Professor
Dept. of Computer Science & Engg.
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Preface

Recap of Previous Lecture:

In previous lecture, we have discussed the Hamiltonian Graph Traveling Salesman Problem and NP-Completeness.

Content of this Lecture:

In this lecture, we will discuss the Connected Dominating Set and Distributed Algorithm.

Preliminaries

- Given **graph G =(V,E),** two vertices are independent if they are not neighbors. For any vertex v, the set of independent neighbors of v is a subset of v's neighbors such that any two vertices in this subset are independent.
- **An independent set (IS))** S of G is a subset of V such that for all u,v $\in$ S, (u,v) $\notin$ E.
- **MIS:** S is maximal if any vertex not in S has a neighbor in S (denoted by MIS).
- **A dominating set (DS)** D of G is a subset of V such that any node not in D has at least one neighbor in D. If the induced subgraph of D is connected, then D is a connected dominating set (CDS).
- Among all CDSs of graph G, the one with minimum cardinality is called **a minimum connected dominating set (MCDS)**

Introduction

- The idea of ***virtual backbone routing*** *for* ***ad hoc wireless networks*** is to operate routing protocols over a virtual backbone.
- One purpose of virtual backbone routing is to alleviate the serious ***broadcast storm problem suffered*** by many exiting on-demand routing protocols for route detection. Thus constructing a virtual backbone is very important.
- In this lecture we study, the virtual backbone is approximated by a ***minimum connected dominating set (MCDS)*** in a unit-disk graph. This is a **NP-hard problem.**
- A distributed approximation algorithm with **performance ratio at most 8** will be covered.

Sensor Network as Adhoc network

- **Ad hoc wireless and Sensor network** has applications in emergency search-and-rescue operations, decision making in the battlefield, data acquisition operations in inhospitable terrain, etc.
- It is featured by **dynamic topology (infrastructureless), multihop communication**, **limited resources** (bandwidth, CPU, battery, etc) and **limited security**.
- These characteristics put special challenges in **routing protocol design** inspired by the **physical backbone** in a wired network, many researchers proposed the concept of virtual backbone for unicast, multicast/broadcast in ad hoc wireless networks .

Contd...

- The **virtual backbone** is mainly **used to collect topology information for route detection**. It also works as a backup when route is unavailable temporarily.
- An effective approach based on overlaying a **virtual infrastructure (termed core)** on an ad hoc network is popular.
- Routing protocols are operated over the core.
- Route request packets are **unicasted to core nodes** and a (small) subset of **non-core nodes.**
- No broadcast is involved **in core path detection**.

Classification of Routing Protocols

- Existing routing protocols can be classified into two categories: (i) *Proactive and (ii) Reactive.*

(i) Proactive routing protocols ask each host (or many hosts) to maintain global topology information, thus a route can be provided immediately when requested.

- But large amount of control messages are required to keep each host updated for the newest topology changes.

(ii) Reactive routing protocols have the feature on-demand. Each host computes route for a specific destination only when necessary.

- Topology changes which do not influence active routes do not trigger any route maintenance function, thus communication overhead is lower compared to proactive routing protocol.

On-demand Routing Protocols

- **On-demand routing protocols** attract much attention due to their better scalability and lower protocol overhead.
- But most of them use **flooding** for route discovery. Flooding suffers from **broadcast storm problem**.
- **Broadcast storm problem** refers to the fact that flooding may result in excessive redundancy, contention, and collision. This causes high protocol overhead and interference to other ongoing communication sessions.
- On the other hand, the **unreliability of broadcast** may obstruct the detection of the shortest path, or simply can't detect any path at all, even though there exists one.

Problem of efficiently constructing virtual backbone for ad hoc wireless networks

- In this lecture we study the "**problem of efficiently constructing virtual backbone"** for ad hoc wireless networks.
- The number of hosts forming the virtual backbone must be as small as possible to decrease protocol overhead.
- The algorithm must be **time/message efficient** due to **resource scarcity.**
- We use a **connected dominating set (CDS)** to approximate the virtual backbone.

Assumptions (1)

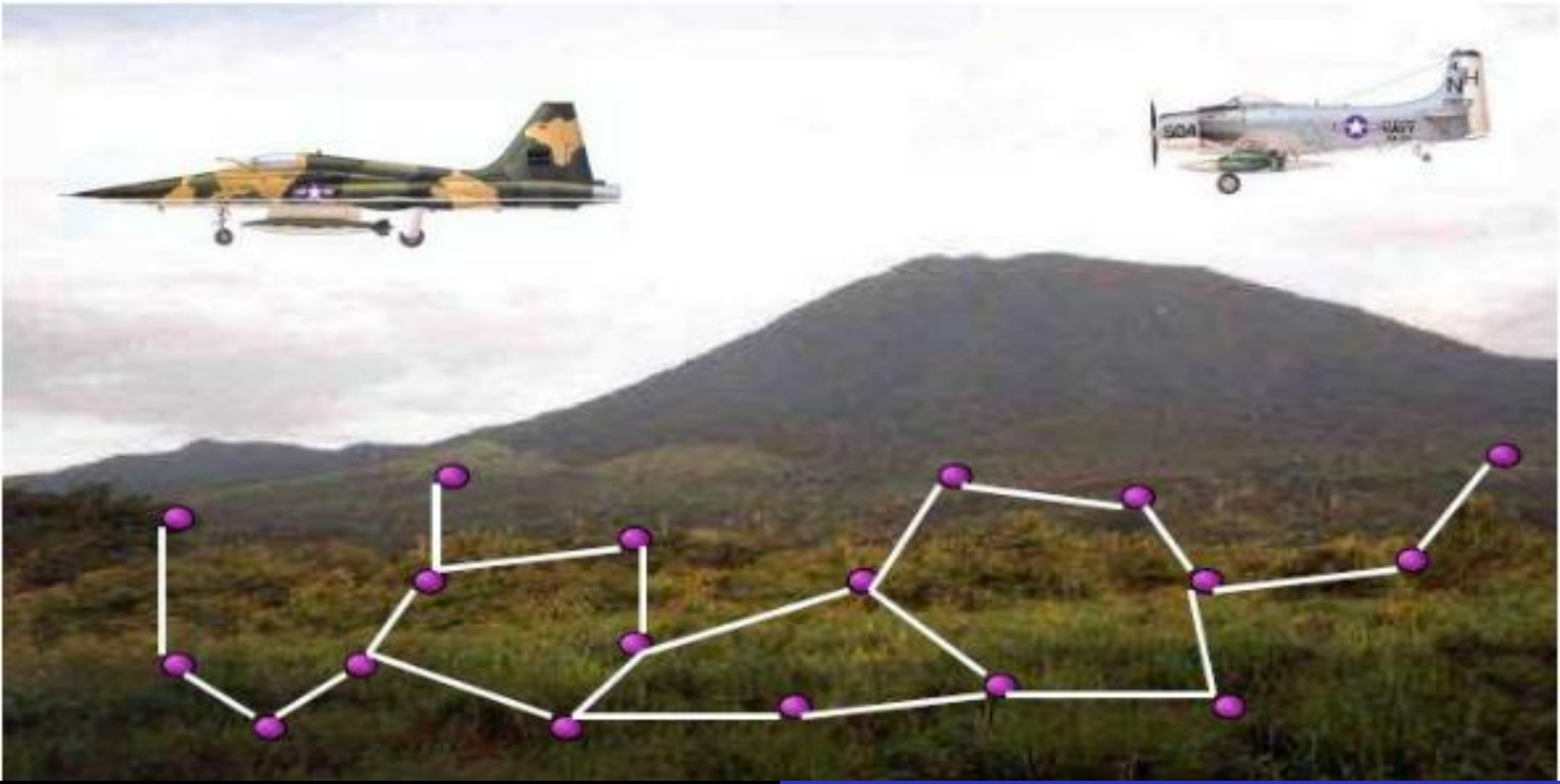
- We assume a given ad hoc network instance contains **n hosts**.
- Each host is in the ground and is mounted by an **omni-directional antenna.**
- Thus the transmission range of a host is a disk.
- We further assume that each transceiver has the same **communication range R.**
- Thus the footprint of an ad hoc wireless network is a unit-disk graph.

Assumptions (2)

- In graph-theoretic terminology, the network topology we are interested in is a **graph G=(V,E) where V contains all hosts and E is the set of links.**
- A **link between u and v** exists if their **distance is at most R**. In a real world ad hoc wireless network, sometime even when v is located in u's transmission range, v is not reachable from u due to hidden/exposed terminal problems.
- Here, we only consider **bidirectional links.**
- From now on, we use host and node interchangeably to represent a wireless mobile.

Sensor Self Deployment

Sensor Dropping for Aircrafts



Existing Distributed Algorithms for MCDS

	B. Das et al. [1997]-I	B. Das et al. [1997]-II	J. Wu et al. [1999]	K.M. Alzoubi [2001]	Mihaela Cardei et al. [2002]
Cardinality	$\le(2\ln\Delta+3)opt$	$\le(2\ln\Delta+2)opt$	N/A	$\le 8opt+1$	$\le 8opt$
Message	$O(n\|C\|+m+n\log n)$	$O(n\|C\|)$	$O(n\Delta)$	$O(n\log n)$	$O(n)$
Time	$O((n+\|C\|)\Delta)$	$O((\|C\|+\|C\|)\Delta)$	$O(\Delta^2)$	$O(n\Delta)$	$O(n\Delta)$
Message Length	$O(\Delta)$	$O(\Delta)$	$O(\Delta)$	$O(\Delta)$	$O(\Delta)$
Information	2-hop	2-hop	2-hop	1-hop	1-hop

Table 1: Performance comparison of the algorithms. Here **opt is the size of the given instance; Δ is the maximum degree; C is the size of the generated connected dominating set; m is the number of edges; n is the number of hosts.**

Preliminaries (1)

- Given **graph G =(V,E),** two vertices are independent if they are not neighbors. For any vertex v, the set of independent neighbors of v is a subset of v's neighbors such that any two vertices in this subset are independent.
- **An independent set (IS))** S of G is a subset of V such that for all $u,v \in S$, $(u,v) \notin E$.
- **MIS:** S is maximal if any vertex not in S has a neighbor in S (denoted by MIS).
- **A dominating set (DS)** D of G is a subset of V such that any node not in D has at least one neighbor in D. If the induced subgraph of D is connected, then D is a connected dominating set (CDS).
- Among all CDSs of graph G, the one with minimum cardinality is called **a minimum connected dominating set (MCDS)**

Preliminaries (2)

- Computing an MCDS in a unit graph is **NP-hard.** Note that the **problem of finding an MCDS** in a graph is equivalent to the problem **of finding a spanning tree (ST) with maximum number of leaves.** All non-leaf nodes in the spanning tree form the MCDS. An MIS is also a DS.

- For a graph G, if e = (u,v) ϵ E iff length (e) $\leq$ 1, then G is called a **unit-disk graph.**

Unit disk graph

- The topology of a wireless ad hoc network can be modeled as a ***unit-disk graph*** *G = (V, E), a geometric graph* in which there is an edge between two nodes if and only if their distance is at most one.

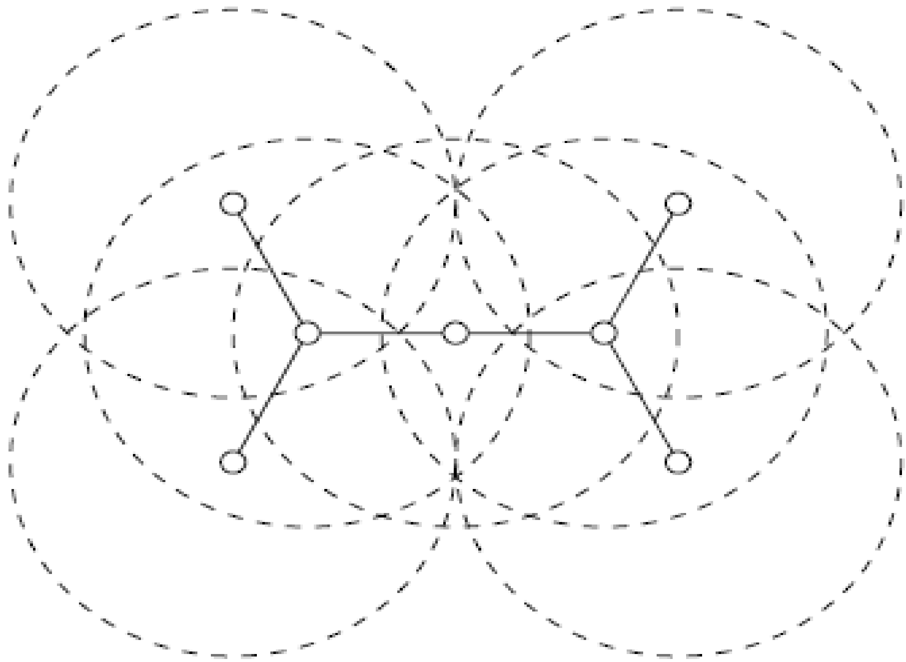


Figure 1: Modeling the topology of wireless ad hoc networks by unit-disk graphs.

Preliminaries (3)

Lemma 1. The size of any independent set in a unit-disk graph G=(V,E) is at most 4opt+1.

Proof

- Let U be any independent set of V, and let T′ be any spanning tree of *OPT*. Consider an arbitrary preorder traversal of T′ given by $v_1,v_2,\cdots,v_{opt}$. Let U_1 be the set of nodes in U that are adjacent to v_1. For any $2 \le i \le opt$, let U_i be the set of nodes in U that are adjacent to v_i but none of $v_1,v_2,\cdots,v_{i-1}$. Then $U_1, U_2,\cdots,U_{opt}$ form a partition of U. As v_1 can be adjacent to at most five independent nodes, $|U_1| \le 5$. For any $2 \le i \le opt$, at least one node in $v_1,v_2,\cdots,v_{i-1}$ is adjacent to v_i. Thus U_i lie in a sector of at most 240 degree within the coverage range of node v_i . This implies that $|U_i| \le 4$. Therefore,

$$|U| = \sum_{i=1}^{opt} |U_i| \le 5 + 4(opt - 1) = 4opt + 1.$$

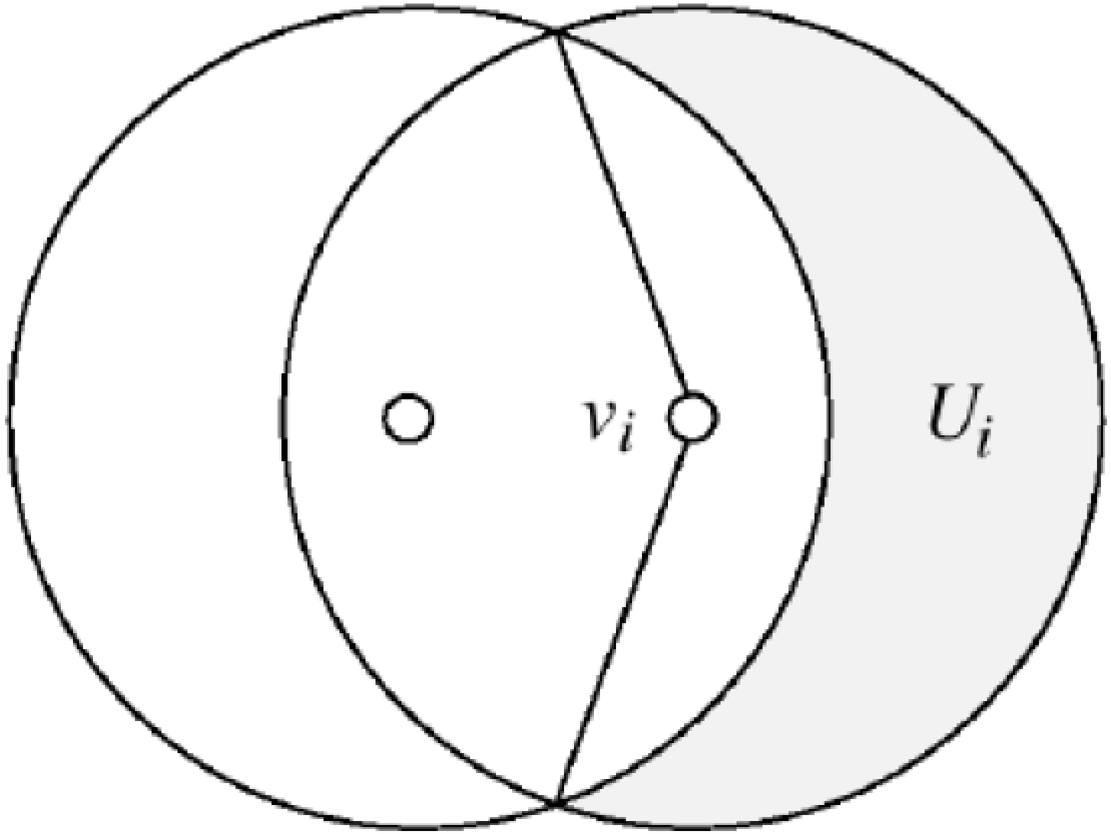


Figure 1: Ui lie in a sector of at most 240 degree within the coverage range of node vi

Preliminaries (4)

- Lemma 1 relates the size of any MIS of unit-disk graph G to the size of its optimal CDS

  Lemma 2.1 From lemma 1, the size of any MIS of G is **at most 4 x opt +1** where opt is the size of any optimal CDS of G.

- For a minimization problem P, the performance ratio of an approximation algorithm A is defined to be

$$\rho_A = sup_{i \in I} \frac{A_i}{opt_i}$$

- where **I is the set of instances** of P, **A_i is the output** from A for instance i and **opt_i is the optimal solution** for instance i. In other words, **ρ is the supreme of A/opt among all instances of P.**

An 8-approximate algorithm to compute CDS

- This algorithm contains two phases:
- **Phase-1:** First, a **maximal independent set (MIS)** is computed;
- **Phase-2:** Then a **Steiner tree** is used to connect all vertices in the MIS.
- This algorithm has **performance ratio at most 8** and is **message and time efficient.**

Algorithm description

- Initially each host is colored **white.**
- A **dominator is colored black**, while a **dominatee is colored gray**.
- We assume that each vertex knows its **distance-one neighbors** and their **effective degrees d*.**
- This information can be collected by periodic or event-driven **hello messages.**
- The **effective degree** of a vertex is the total number of **white neighbors.**

Contd...

- Here **host** is designated as the **leader**. This is a realistic assumption.
- For example, the leader can be the commander's mobile for a platoon of soldiers in a mission.
- If it is impossible to designate any leader, a **distributed leader-election algorithm** can be applied to find out a leader. This adds message and time complexity.
- The **best leader-election algorithm** takes **time O(n)** and **message O(nlogn)** and these are the best-achievable results. Assume **host s is the leader**

Phase 1:

- **Host s** first **colors itself black** and broadcasts message **DOMINATOR**.
- Any **white host u** receiving **DOMINATOR** message the first time from **v** colors itself **gray** and broadcasts message **DOMINATEE.** u selects v as its dominator.
- A **white host** receiving **at least one DOMINATEE** message becomes active.
- An active white host with **highest (d^*, id)** among all of its **active white neighbors** will **color itself black** and broadcast message **DOMINATOR**.

Contd...

- A **white host decreases its effective degree by 1** and broadcasts message **DEGREE** whenever it receives a **DOMINATEE** message.

- Message **DEGREE** contains the sender's current effective degree. A white vertex receiving a **DEGREE** message will update its neighborhood information accordingly.

- Each gray vertex will broadcast message **NUMOFBLACKNEIGHBORS** when it detects that none of its neighbors is white.

- **Phase 1** terminates when **no white vertex left.**

Phase 2:

- When **s** receives message **NUMOFBLACKNEIGHBORS** from all of its gray neighbors, it starts **phase 2** by broadcasting **message M**.

- A host is **"ready"** to be explored if it has **no white neighbors.**

- **A Steiner tree** is used to connect **all black hosts** generated in **Phase 1.**

- The idea is to pick those **gray vertices** which connect to many **black neighbors.**

Contd...

- The classical **distributed depth first search spanning** tree algorithm will be modified to compute the **Steiner tree.**

- A **black vertex** without any dominator is **active.**

- Initially **no black vertex** has a dominator and all hosts are **unexplored.**

- Message M contains a field next which specifies the **next host** to be explored.

- A **gray vertex** with at least **1 active black neighbors** are **effective.**

Contd...

- If **M** is built by a **black vertex**, its **next field** contains the **id** of the **unexplored gray neighbor** which connects to maximum number **of active black hosts.**

- If **M** is built by a **gray vertex,** its **next** field contains the **id** of any **unexplored black neighbor.**

- Any **black host u** receiving an **M message** the first time from a **gray host v** sets its dominator to **v** by broadcasting message **PARENT.**

Contd...

- When a **host u** receives **message M** from v that specifies u to be explored next, if none of **u's neighbors is white**, **u** then colors itself **black**, sets its dominator to **v** and broadcasts its **own M message**; otherwise, **u** defer its operation until none of its neighbors is **white.**

- Any **gray vertex** receiving message PARENT from a **black neighbor** will broadcast message **NUMOFBLACKNEIGHBORS**, which contains the number of **active black neighbors.**

- A **black vertex** becomes **inactive** after its dominator is set.

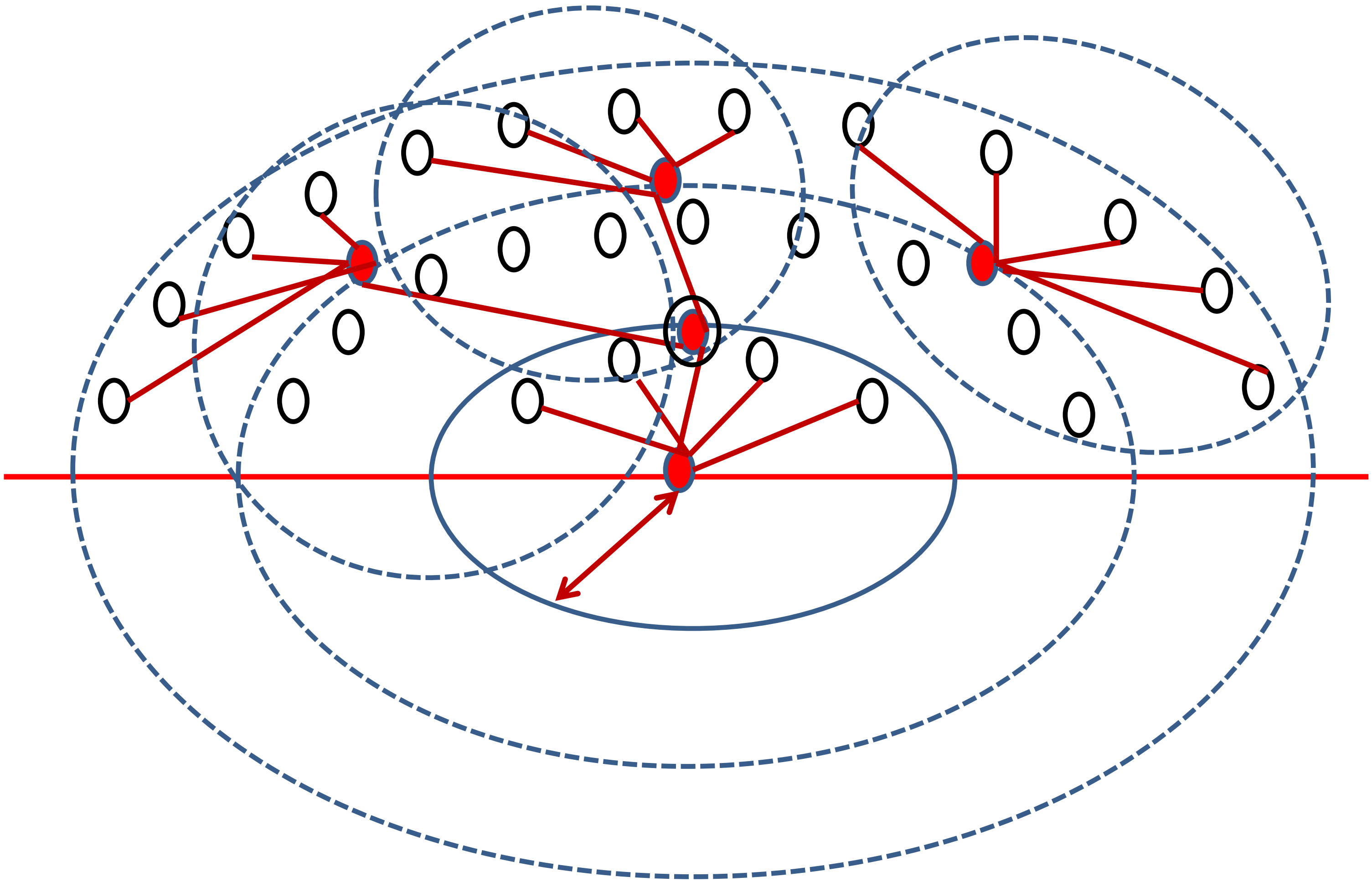
Contd...

- A **gray vertex** becomes **ineffective** if none of its **black neighbors is active.**

- A **gray vertex** without **active black neighbor**, or a **black vertex** without effective **gray neighbor**, will send message **DONE** to the **host** which activates its exploration or to its dominator.

- When s gets message **DONE** and it has no effective **gray neighbors,** the algorithm **terminates.**

Example:



Complexity

- Note that phase 1 sets the dominators for **all gray vertices**. Phase 2 may modify the dominator of some **gray vertex**.
- The main job for **phase 2 is to set a dominator for each black vertex.** All black vertices form a CDS.
- In Phase 1, each host broadcasts each of the messages **DOMINATOR** and **DOMINATEE** at most once.
- The message complexity is dominated by message **DEGREE,** since it may be broadcasted **Δ times** by a host, where **Δ is the maximum degree.**
- Thus the **message complexity of Phase 1 is O(n Δ).** The **time complexity of Phase 1 is O(n).**

Contd...

- In phase 2, **vertices are explored one by one.**

- The **total number of vertices** explored is the **size of the output CDS.** Thus the time complexity is **at most O(n).**

- The message complexity is dominated by message **NUMOFBLACKNEIGHBORS,** which is broadcasted **at most 5 times** by each **gray vertex** because a gray vertex has **at most 5 black neighbors** in a **unit-disk graph.**

- Thus the **message complexity is also O(n).**

Theorem

- **Theorem 3.1:** The distributed algorithm has **time complexity O(n)** and **message complexity O(n. Δ)**

- Note that in phase 1 if we use **(id) instead of (d^*, id)** as the parameter to select a **white vertex** to color it **black,** the **message complexity will be *O(n)*** *because* no **DEGREE** messages will be broadcasted.

- **O(n. Δ) is the best result** we can achieve if effective degree is taken into consideration.

Performance Analysis

- **Lemma 3.2** Phase 1 computes an MIS which contains all black nodes.

- **Proof.** A node is colored black only from white. No two white neighbors can be colored black at the same time since they must have different (d^*, id) .
- When a node is colored black, all of its neighbors are colored gray.
- Once a node is colored gray, it remains in color gray during Phase 1.
- From the proof of **Lemma 3.2**, it is clear that if (id) instead of (d^*, id) is used, we still get an MIS. Intuitively, this result will have a larger size.

Contd...

- **Lemma 3.3** In phase 2, at least one gray vertex which connects to maximum number of black vertices will be selected.

- **Proof.** Let u be a gray vertex with maximum number of black neighbors.
- At some step in phase 2, one of u's black neighbor v will be explored.
- In the following step, u will be explored. This exploration is triggered by v.

Contd...

- **Lemma 3.4** If there are c black hosts after phase 1, then at most c-1 gray hosts will be colored black in phase 2

- **Proof.** In phase 2, the first gray vertex selected will connect to at least 2 black vertices.
- In the following steps, any newly selected gray vertex will connect to at least one new black vertex.

Contd...

- **Lemma 3.5** If there exists a gray vertex which connects to at least 3 black vertices, then the number of gray vertices which are colored black in phase 2 will be at most c-2, where c is the number of black vertices after phase 1.

- **Proof.** From **Lemma 3.3**, at least one gray vertex with maximum black neighbors will be colored black in phase 2. Denote this vertex by u. If u is colored black, then all of its black neighbors will choose u as its dominator. Thus, the selection of u causes more than 1 black hosts to be connected.

Contd...

- **Theorem 3.6** This algorithm has performance ratio at most 8.

- **Proof.** From Lemma 3.2, phase 1 computes a MIS. We will consider two cases here.
- If there exists a gray vertex which has at least 3 black neighbors after phase 1, from Lemma 2.1, the size of the MIS is **at most 4.opt+1.**
- From lemma 3.5, we know the total number of black vertices after phase 2 is **at most 4.opt+1+((4.opt+1)-2=8.opt.**

Contd...

- If the maximum number of black neighbors a gray vertex has is at most 2, then the size of the MIS computed in phase 1 is **at most 2.opt** since any vertex in opt connects to at most 2 vertices in the MIS.
- Thus from **Lemma 3.4,** total number of black hosts will be **2.opt + 2.opt – 1 < 4.opt**
- **Note** that from the proof of **Theorem 3.6,** if **(id) instead of (d^*, id)** is used in phase 1, this algorithm still has **performance ratio at most 8**

More References

- Rajiv Misra et al., ”Minimum Connected Dominating Set Using a Collaborative Cover Heuristic for Ad Hoc Sensor Networks.” IEEE Trans. Parallel Distrib. Syst. 21(3): 292-302 (2010)

Conclusion

- In this lecture, we have discussed a **distributed algorithm which compute a connected dominating set with smaller size.**
- We have discussed how to find a **maximal independent set.** Then how to use a **Steiner tree** to connect all vertices in the set.
- This algorithm gives **performance ratio at most 8.**
- The future scope of this algorithm is to study the problem of maintaining the **connected dominating set in a mobility environment.**