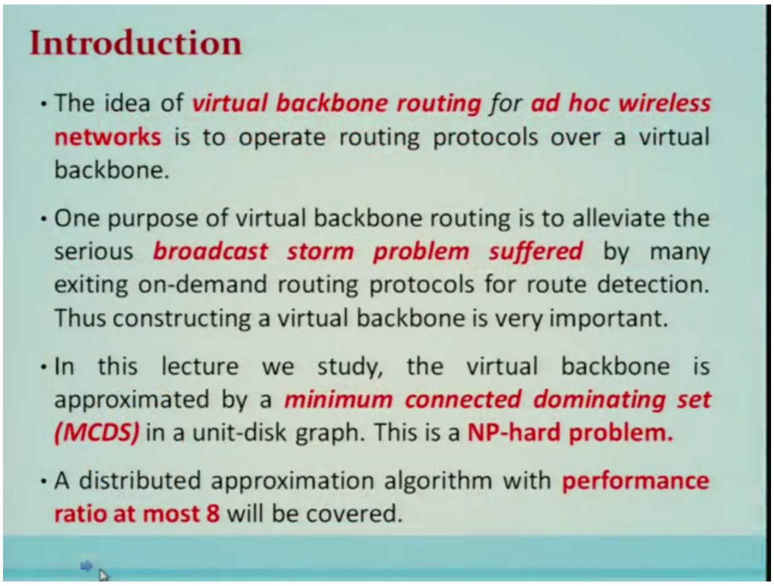
Distributed Systems
Dr. Rajiv Misra
Department of Computer Science and Engineering
Indian Institute of Technology, Patna

Lecture - 24
Distributed Algorithms for Sensor Networks

Distributed Algorithms for Sensor Network Introduction the idea of virtual backbone routing for ad hoc wireless network is to operate routing protocol over the virtual backbone.

(Refer Slide Time: 00:18)



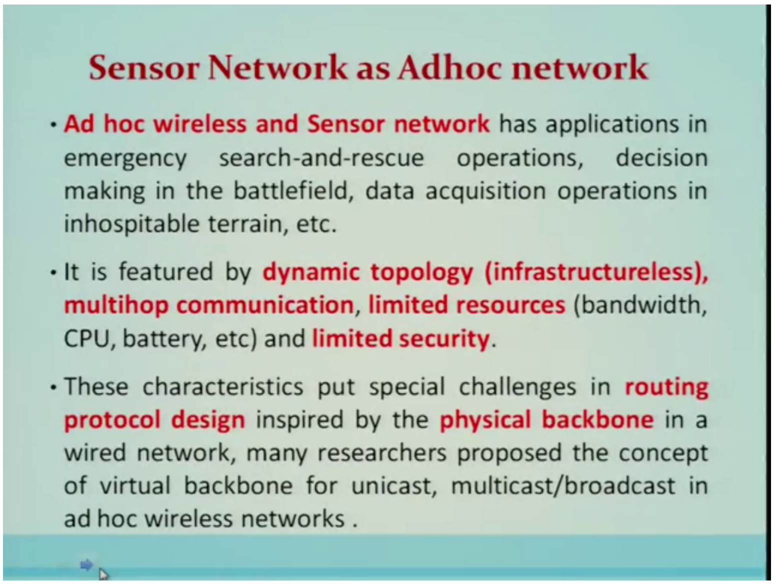
Introduction

- 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.

The purpose of virtual backbone routing is to alleviate the serous broadcast system problem suffered by many existing on-demand routing protocols for route detection and thus constructing a virtual backbone is very important. In this lecture we study the virtual backbone construction by an approximation algorithm which is approximated by a minimum connected dominating set MCDS in a unit-disk graph this is an NP-hard problem.

The distributed algorithm for approximation this particular problem has the performance ratio of 8 that will be discussed.

(Refer Slide Time: 01:15)



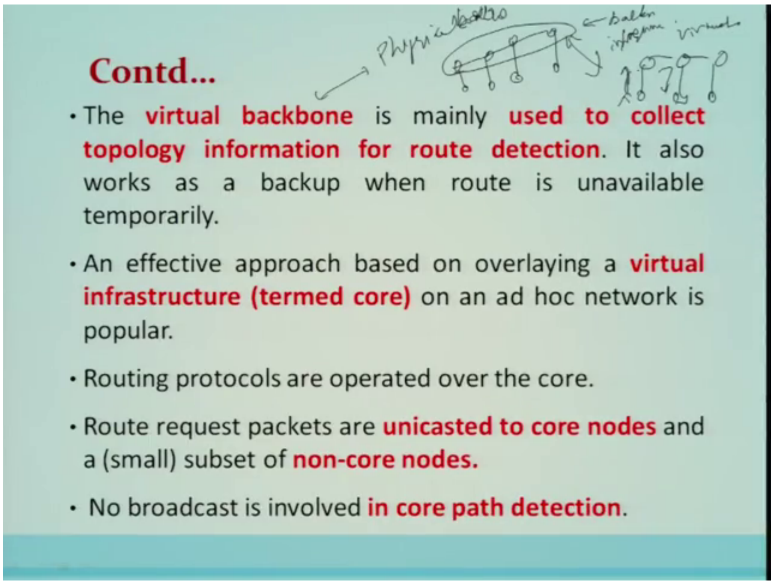
Sensor Network as Adhoc network

- **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 .

So, before going ahead let us define some of the basic terminologies. So, sensor networks is an Ad hoc network- ad hoc wireless sensor network has an application in emergency search-and-rescue operations, decision making in battlefield, data acquisition operation in inhospitable terrain and so on and so forth. So, such network can be spontaneously formed to basically do the surveillance and the data acquisition such kind of activities.

So, it is featured by the dynamic topology and that dynamic topology should basically be infrastructure less for spontaneous construction. Multihop communication, limited resources and limited security are some of the characteristics of such networks. These characteristics put special challenges in routing protocol design which are inspired by the physical backbone in a wired network. Many researchers propose the concept of a virtual backbone for unicast, multicast, broadcast in ad hoc networks.

(Refer Slide Time: 02:32)



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**.

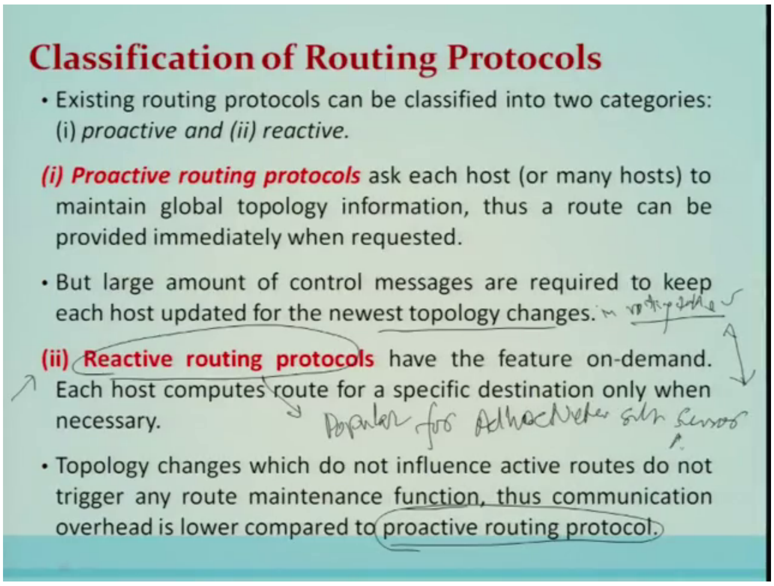
So, virtual backbone is mainly used to collect the topology information for route detection. It also works as a backup when the route is unavailable temporarily. An effective approach based on overlaying a virtual infrastructure which is termed as core on an ad hoc network is very popular scheme.

Routing protocols are operated over this particular core, which is formed using virtual backbone. Router request packets are unicasted and a small subset of non-core nodes, no broadcast is involved in the core path detection.

Let us before we go ahead let us take this particular example of the virtual backbone. So, virtual backbone is basically in contrast with the physical backbone in the wired network. So, in the physical backbone system you might see a backbone to which all other nodes are connected. This backbone is a high speed backbone in the physical network maybe an optical fiber networks and these are the other non backbone nodes they can communicate through this particular backbone not for the data communication. Inspired from this infrastructure less networks like sensor networks and ad hoc networks they also inspired by this physical backbone and they have come out to construct a similar kind of backbone structure which is called a virtual backbone.

So, if the virtual backbone is in place. So, the nodes which are not in the backbone they can communicate through the backbone for example, x want to communicate to y. So, x will send the message to the backbone and the backbone which is closest to y will basically transmitted. So, this particular backbone is very important to basically facilitate the routing activity also this particular backbone will solve the broadcast storm problem that we are going to see in the next slide.

(Refer Slide Time: 05:03)



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. in routing table

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

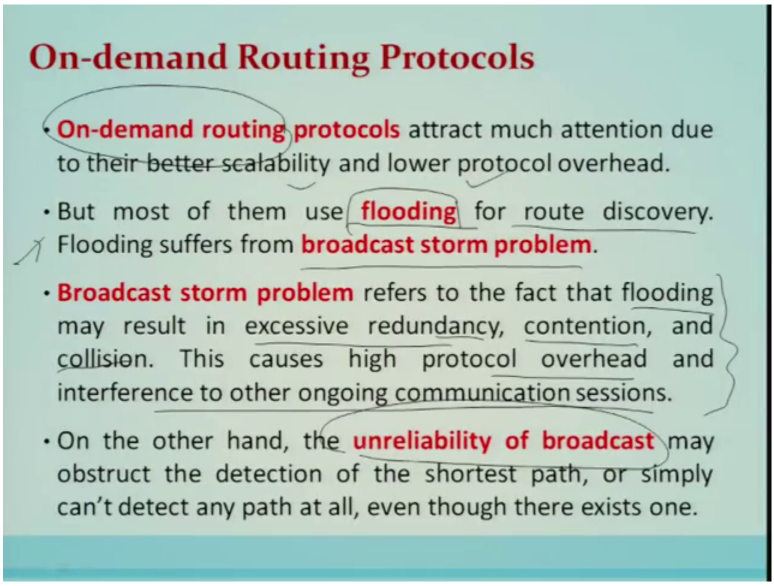
- Topology changes which do not influence active routes do not trigger any route maintenance function, thus communication overhead is lower compared to proactive routing protocol.

So, the routing protocols in the wireless networks like sensor networks and ad hoc network can be classified in 2 different types. The first one is called proactive the other one is called reactive proactive routing protocols they ask each node or many node which is also called as a host to maintain a global topology information and thus route can be provided immediately whenever it is requested. But a large amount of control messages are required to keep the each host updated for the newest topology changes in it is routing table. So, routing table has to be updated whenever there is a slight change in the topologies are encountered and this is called proactive routing protocol whether it is used or not the routing table has to be updated at all point of time.

In contrast to the proactive routing protocol there is an alternative solution which is called reactive protocols; reactive routing protocol have a feature of on demand construction of the routing path. So, each node computes the route for a specific destination only whenever it is necessary so; obviously, in contrast to the proactive the reactive routing protocols are affordable are basically can be used in such a network which are resource constrained network. So, topology changes which do not influence that to route do not trigger any route maintenance function and thus the communication over it is lower.

So, basically proactive routing protocols are way. So, in compared to proactive routing protocol reactive routing protocols is very popular for ad hoc networks such as sensor network.

(Refer Slide Time: 07:08)



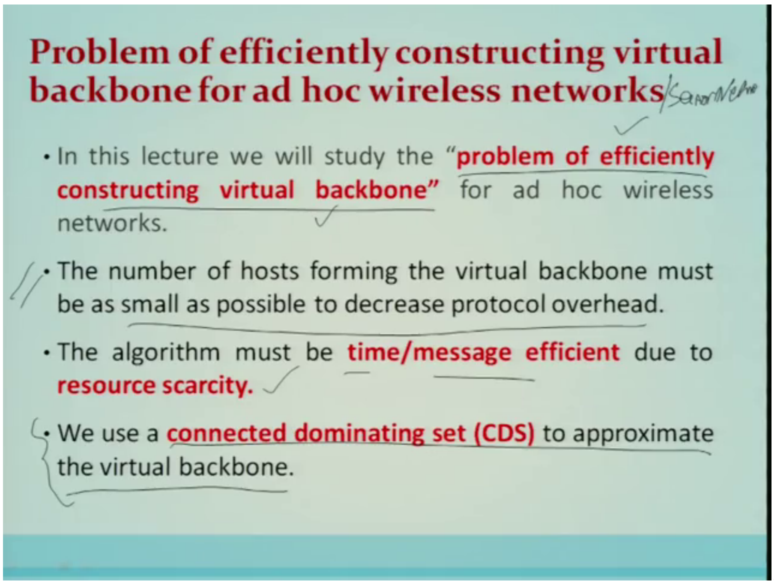
On-demand Routing Protocols

- 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.

Now On-demand routing protocol attract much attention due to their better scalability and lower protocol overhead that I have just talked about, but let us see the more detailed more problematic in this kind of setting, but most of them most of them means most of these on demand routing protocol which are on demand construction of a routing path uses the flooding to discover the routes, but this flooding suffers from a broadcast storm problem broadcast storm refers to excessive flooding may result into an excessive redundancy excessive contention and excessive collision.

So, this particular problem of excessive redundancy contention and collision is called a broadcast storm problem this causes high protocol overheads and interference to the other ongoing communication channel. So, due to the broadcast system problem this particular flooding based on demand routing protocol requires some more innovative intuitive or innovative solutions. So, on the other hand unreliability of the broadcast obstructs the detection of the shortest path or simply cannot detect any path at all even there exists one.

(Refer Slide Time: 08:29)



Problem of efficiently constructing virtual backbone for ad hoc wireless networks/SensorNetw

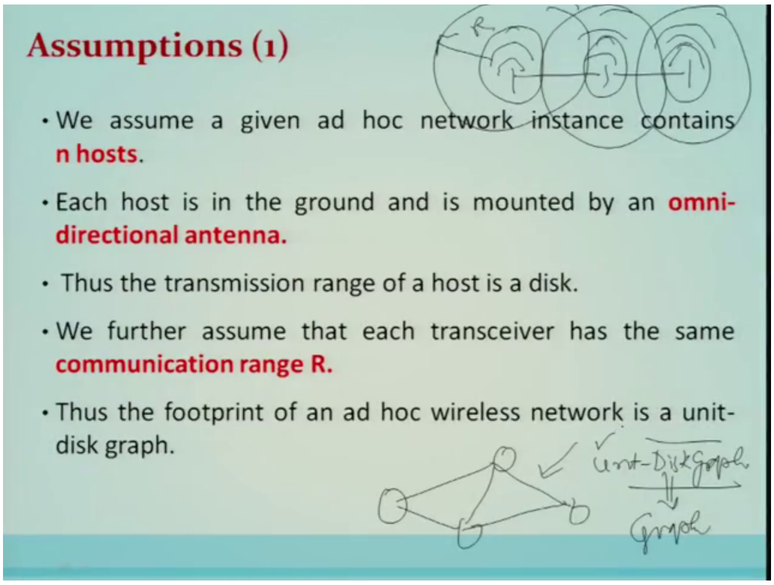
- In this lecture we will study the "**problem of efficiently constructing virtual backbone**" for ad hoc wireless networks.
- The number of hosts forming the virtual backbone must be as small as possible to decrease protocol overhead.
- The algorithm must be **time/message efficient** due to **resource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

So, Problem of efficiently constructing the virtual backbone for ad hoc wireless networks such as the sensor networks.

So, the alternative solution is of the broadcast problem is basically to maintain a virtual backbone which is quite similar to the physical backbone, but in a infrastructure less network like sensor network. So, how efficiently maintain this virtual backbone which can be used by the routing purposes on this particular network. So, constructing a virtual backbone efficiently becomes an important task, which we are going to see in this part of the lecture. So, number of host forming the virtual backbone must be as small as possible to decrease the protocol overhead this is the most important issue in designing the virtual backbone for such networks.

So, the algorithm must be time and message efficient due to the resource scarcity. So, we use the connected dominating set to approximate the virtual backbone. So, we used to we will construct we will see that how this connected dominating set will approximate the virtual backbone and then if the virtual backbone is in place, then the routing will become more efficient and will not have the problem like broadcast storm problem.

(Refer Slide Time: 10:02)



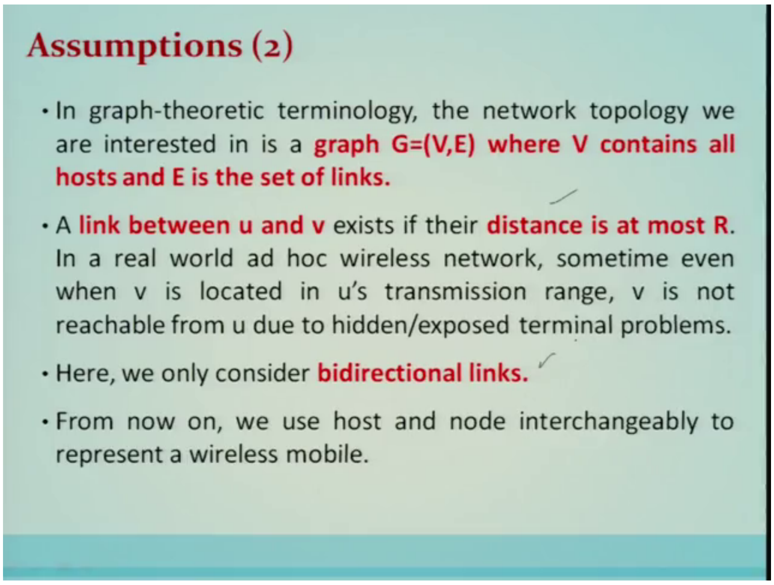
Assumptions (1)

- 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.

So, for that let us see the assumptions. So, we assume a given ad hoc network and instance which contains n different hosts, each host is in the ground and is mounted by an omni-directional antenna this will facilitate the transmission network or a communication network among these nodes. This transmission range of a host can be constructed in a form of a disk of a particular radius R because it is omni-directional it can communicate around a circle of a radius R.

Thus the footprint of an ad hoc wireless network is a unit disk graph. So, the graph which have this kind of overlapping disks they can be considered as an edge of a graph and these nodes are the vertices. So, they will form a graph in this particular manner this particular graph is called a unit disk graph. So, in such networks which are called ad hoc networks or the sensor network the graph which is constructed is of unit disk graph. So, in brief we can call it as a graph, but before that let us see what the unit disk graph structure is.

(Refer Slide Time: 11:41)



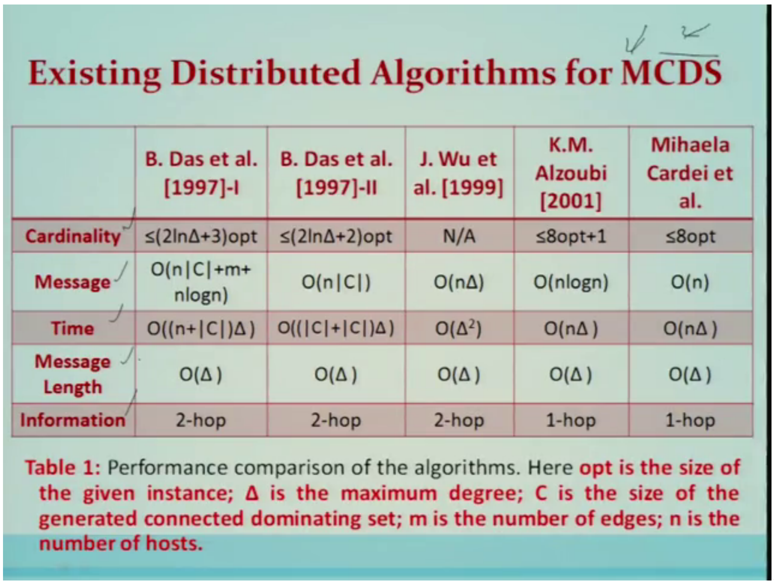
Assumptions (2)

- 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.

So, in a graph theoretic terminology the network topology is in the graph where we contains all the hosts and E is the set of links if they are overlapping in their unit disk and the 2 nodes are in their communication range within each other.

Now, we consider we also assume that these links are bidirectional that is both nodes both sides of the edge they can communicate with each other wirelessly, we assume that kind of bidirectional links are established using the omni directional antenna in the wireless sensor network. From now on we will use the host and node inter changeably to represents a wireless mobile node.

(Refer Slide Time: 12:34)



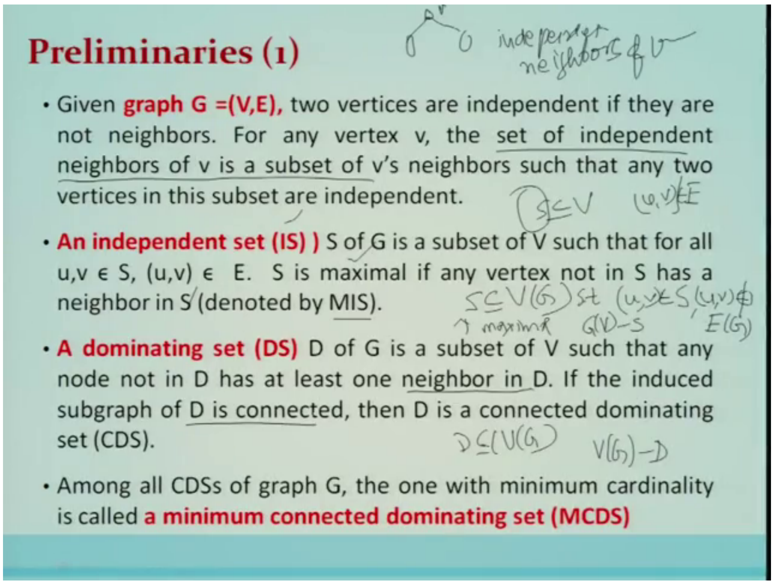
Existing Distributed Algorithms for MCDS

	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.
Cardinality	$\leq(2\ln\Delta+3)$opt	$\leq(2\ln\Delta+2)$opt	N/A	≤ 8opt+1	≤ 8opt
Message	$O(n\lvert C\rvert+m+n\log n)$	$O(n\lvert C\rvert)$	$O(n\Delta)$	$O(n\log n)$	$O(n)$
Time	$O((n+\lvert C\rvert)\Delta)$	$O((\lvert C\rvert+\lvert C\rvert)\Delta)$	$O(\Delta^2)$	$O(n\Delta)$	$O(n\Delta)$
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.

The existing distributed algorithm for MCDS means minimum connected dominating set problem. So, there are various metrics like message time message length information and cardinality and based on that different algorithms, they have given their optimal results that we will discuss a bit later.

(Refer Slide Time: 13:04)



Preliminaries (1)

independent neighbors of v

- 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.

$S \subseteq V$ $(u,v) \notin E$

- **An independent set (IS))** S of G is a subset of V such that for all u,v ∈ S, (u,v) ∈ E. S is maximal if any vertex not in S has a neighbor in S (denoted by MIS).

$S \subseteq V(G)$ st $(u,v) \in S, (u,v) \notin E(G)$ maximal $G(V)-S$

- **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).

$D \subseteq (V(G))$ $V(G)-D$

- Among all CDSs of graph G, the one with minimum cardinality is called **a minimum connected dominating set (MCDS)**

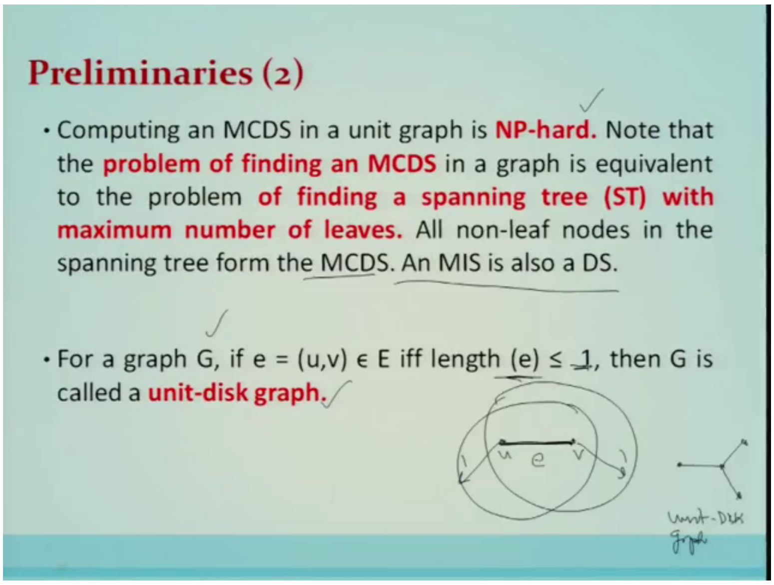
Now, let us go ahead with a preliminaries of the graph which is formed out of the unit disk and that is called unit disk graph. Some of the preliminaries we are going to see from a graph theoretic point of view why because the footprint of ad hoc network or the sensor network becomes a graph.

So, let us see that what this graph has what are the properties which it has let us take the preliminaries, let graph G which is denoted as V, these 2 vertices which are independent they are neighbor. Hence this particular set of independent neighbor of v is called basically the independent neighbors of v, independent set S of G is a subset of V such that for all vertices u, v such as all edges which are in u, v.

So, for any particular for any 2 pair of vertices they do not have an edge. So, such set of subset of vertices is an independent set. So, again I am repeating an independent set S of G is a subset of v s is a subset of v, the set of vertices of G such that for any 2 vertices or any pair of vertices in S, let us say u v is a pair of vertices in S, then u v pair is not having an edge in that particular graph or they are independent a set of a subset of independent vertices is called independent set. Now this independent set is maximal if the vertices which are not in S that is G of v minus S these vertices has a neighbor in S then it is called maximal independent set.

So, a dominating set D of G is a subset of vertices such that any node which is not in D that is not in D has at least one neighbor in D. So, if the induced sub graph of D is connected then D is a connected dominating set otherwise D is a dominating set. Now among all connected dominating sets in the graph the one with a minimal cardinality; minimum cardinality is called minimum connected dominating set or MCDS. So, this is the definition of a minimum connected dominating set.

(Refer Slide Time: 16:35)



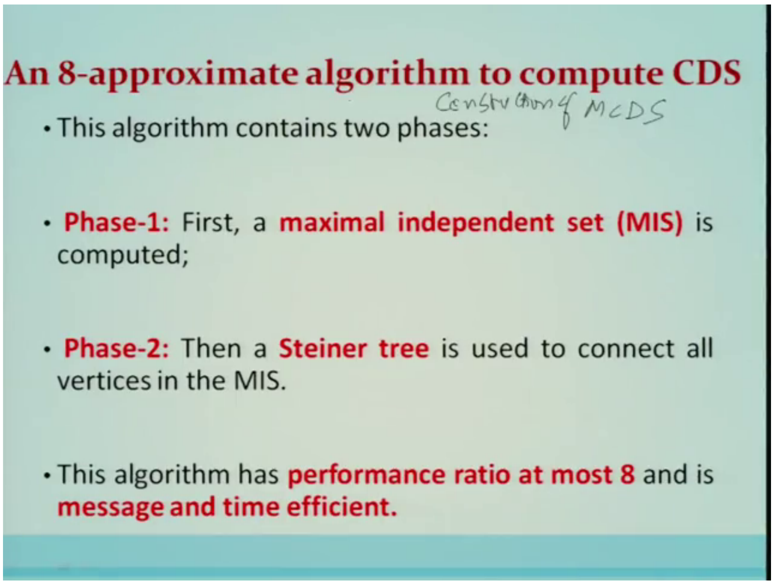
Preliminaries (2)

- 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) ≤ 1, then G is called a **unit-disk graph.**

Computing and MCDS in a unit disk graph is an NP-hard problem that is the construction of MCDS in a unit disk 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 with a maximum number of leaves. We are all non-leaf nodes in the spanning tree will form the MCDS, there this particular problem MCDS or a non leaf nodes of a maximum number of leaf is also NP-hard problem.

Now, an MIS is also or this or dominating set. Now for a graph G if e is an edge where the edge of is of length less than 1 for all edges, which are having length less than 1 then this particular graph is called unit disk graph. Take this particular example if u and v is having an edge. So, if we draw a disk of radius 1 and here also if we draw a disk of radius 1, then you see this particular length is having the length less than 1. So, such edges if it is formed in the graph then this graph is called unit disk graph, where the lengths of these edges are less than 1.

(Refer Slide Time: 18:14)



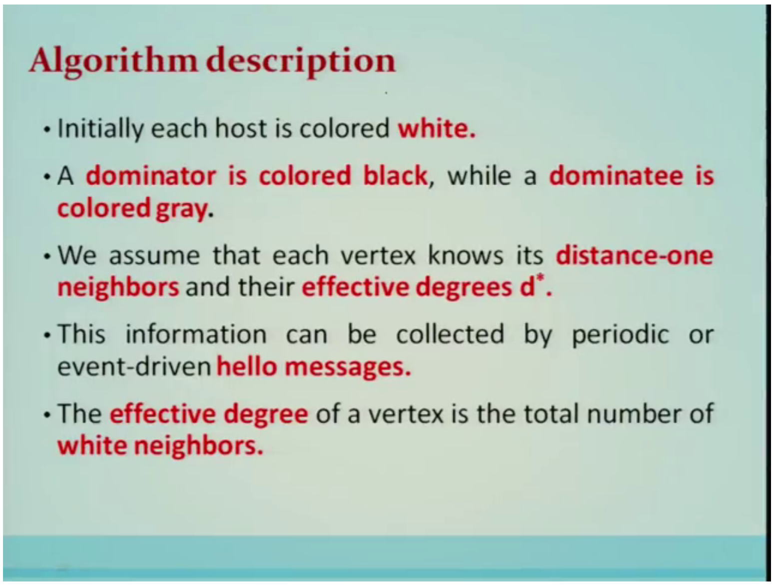
An 8-approximate algorithm to compute CDS

Construction of MCDS

- 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.**

Now, we are going to discuss the algorithm to construct a connected minimum connected dominating set.

(Refer Slide Time: 18:36)

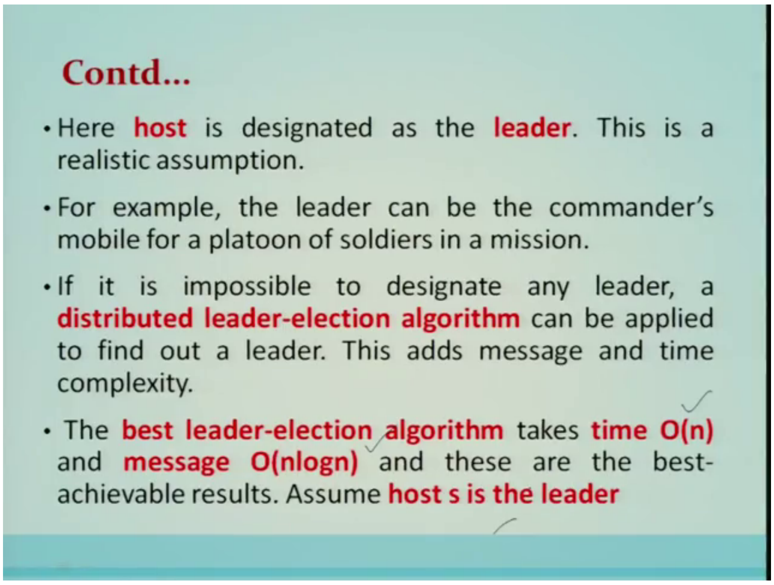


Algorithm description

- 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.**

So, the algorithm description that initially each node is colored white, the dominator is colored black and the dominate is colored gray. We assume that the each vertex knows it is distance 1 neighbor and their affective degrees. This information can be collected by periodic event driven hello messages; the effective degree of a node is the total number of white neighbors.

(Refer Slide Time: 19:00)

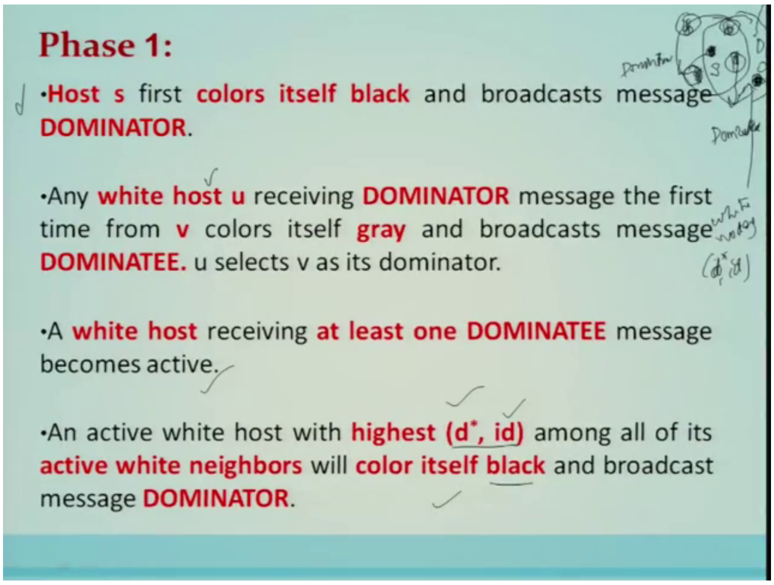


Contd...

- 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**

Here the host is designated as the leader, this is a realistic assumptions. For example, if the leader is not there then it is possible to elect a leader using a distributed leader election algorithm and that will at a additional complexity of in the time order of n and using the best possible algorithm uses the message complexity of order n log n. Hence our assumption of existing or assuming a leader is the realistic assumption and let s is a leader which is given for this particular algorithm.

(Refer Slide Time: 19:46)

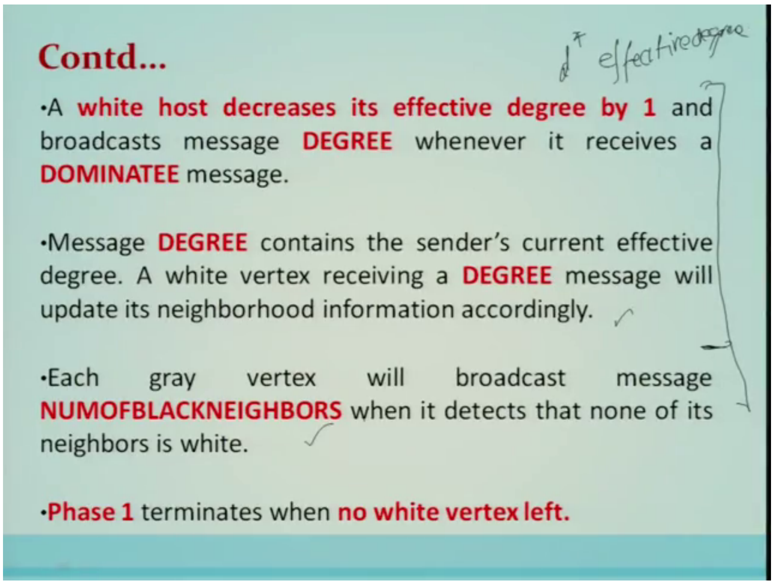


Phase 1:

- **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**.

Let us see the algorithm this algorithm works in 2 phases; the phase 1 goes like this host s color itself a black let us color it as a black and broadcast a message dominator. So, it will broadcast in an a disk of radius 1 and this particular message is called dominator will be broadcast.

(Refer Slide Time: 20:07)



Contd...

d^* effective degree

- •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.**

Now any host u which basically is there inside this particular radius or is a neighbor of node s will receive this particular message dominator. So, this is they are called white hosts on receiving dominator message first time from v they will color itself as a gray.

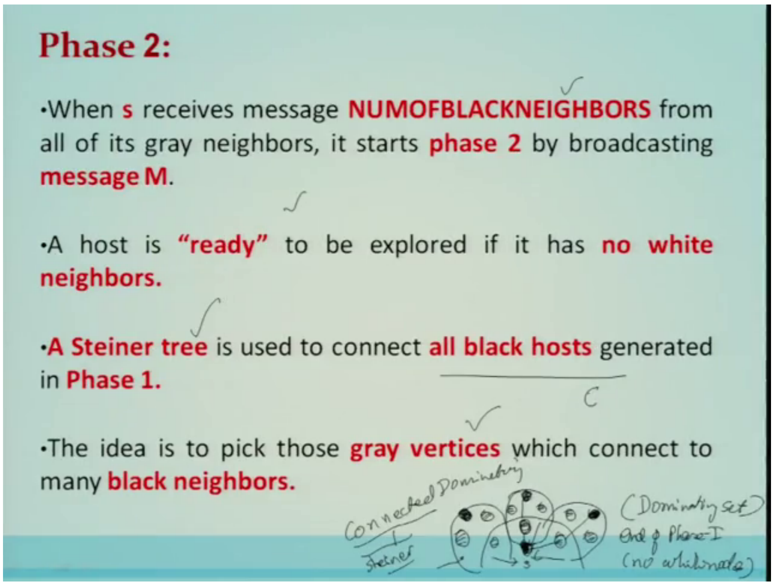
So, let us color it as a gray and broadcast the message called Dominatee let us consider that this particular node broadcast a message dominatee. So, if it broadcast a message dominatee then this message will go towards outside this particular range of this node s, the white node receiving at least 1 dominatee message becomes active. So, this is the range where the white nodes are there. So, when they will receive this is the dominating message they will become active, an active white host with a highest d star d star is effective degree and the id which will form the total order among all of it is active white neighbors will color itself the black.

So; that means, among those white neighbors the one with a highest number of d star and id will color itself as a black let us say this will color a black and broadcast a dominator. A white host decreases it is effective degree by 1 and broadcast a Degree whenever it receives a Dominatee message.

So, from here up to this point it is going to modify the effective degree of the node. So, the message the degree contains the senders current effective degree. So, that is broadcasted, all the white node will in turn will modify. So, that the effective degree value correct value or updated value is known in the neighborhood of a particular white neighbor.

Now, each gray vertex will broadcast the message number of black neighbors when it detects that none of it is neighbors are white and the phase one terminates when no white vertex left.

(Refer Slide Time: 23:03)



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.**

Phase 2 when s receives a message that the number of black neighbors from all of it is gray neighbors it is start phase 2 by broadcasting the message M. A host is ready to be explode if it has no white neighbors a Steiner tree is used to connect all the black hosts generated in phase 1, the idea is to pick those gray vertices which connect to many black neighbors let us understand these steps of phase 2.

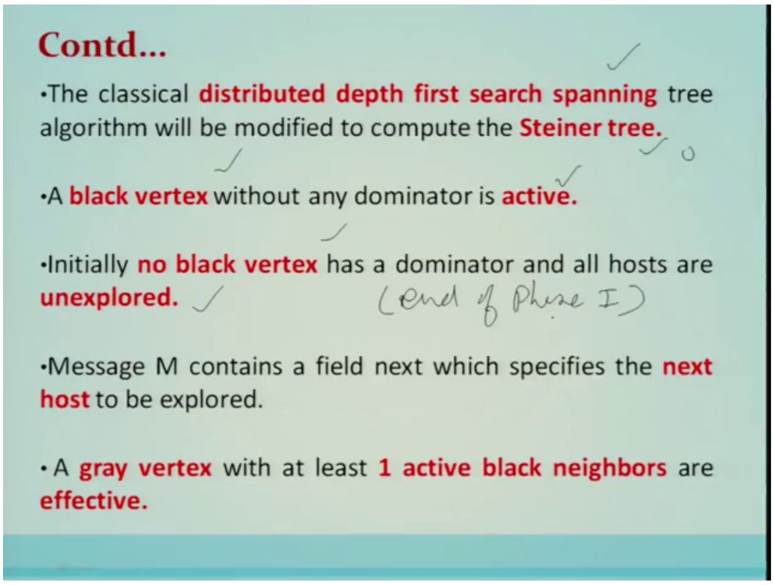
Let us say that in phase 1 it starts with the leader s and in it is neighborhood there are some gray nodes formed and in these gray neighborhoods these nodes are the white nodes, among these white nodes the 1 which is having the highest affective degree they will become the black nodes. So, let us see this is becomes a black node and when they will send the dominator message. So, they will become the gray node also. So, at the phase when the phase 1 ends at the end of phase 1 there is no white node left then only the phase 1 will finish and phase 2 will begin.

Now, when phase 2 begins then these gray neighbors these gray nodes knows that how many blacks are in it is neighborhood for example, this particular gray has 1 black and 1 black 2 black neighborhoods. Similarly this also will have these many number of black nodes and so on. So, when these s message so; that means, these gray nodes will inform to s about how many number of black neighbors it has and this information s is now having and s will now send a message m in phase 2. So, the host to be explored if it has no white neighbors so here you just see that in this when the phase one ends there is no white nodes.

Now, a Steiner tree will be constructed to connect all the black nodes using those grey nodes which can connect 2 or more black nodes. So, now, some of this in phase 2 some of these gray nodes will become the black nodes, those gray nodes which basically have the maximum number of black neighbors they will be blackened up and they will be the Steiner tree construction which will connect all the black nodes. So, the phase 1 will form a dominating set and phase 2 is going to construct out of the gray nodes the connected dominating set.

So; that means, the connection among the dominators or dominating set is established using a tree which is called a Steiner tree, which will connect all the black nodes. Now you know that if there are c different black nodes, then to have a connected tree that is how many nodes are required that is c minus 1. Because c minus 1 is basically if there are n number of nodes then basically how many nodes will be required to form a how many edges are required to form a tree is c minus 1.

(Refer Slide Time: 27:58)



Contd...

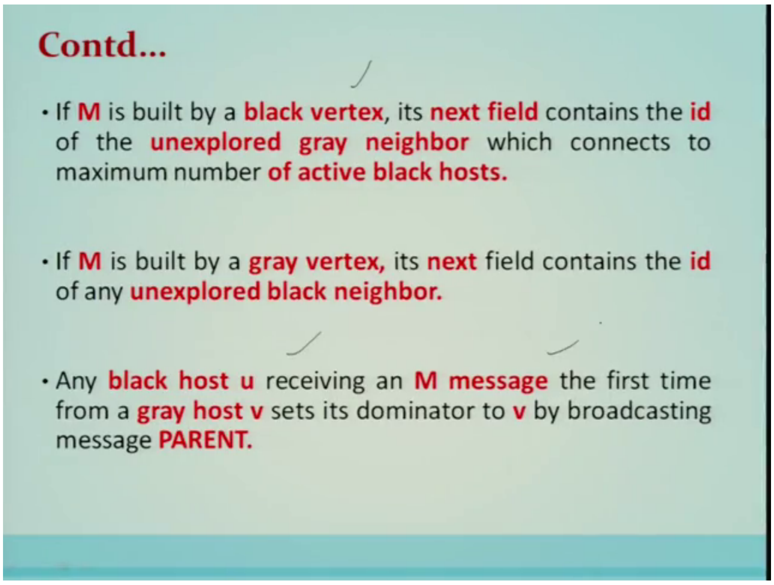
- The classical **distributed depth first search spanning** tree algorithm will be modified to compute the **Steiner tree.**
- A **black vertex** without any dominator is **active.**
- Initially **no black vertex** has a dominator and all hosts are **unexplored.** (end of Phase I)
- Message M contains a field next which specifies the **next host** to be explored.
- A **gray vertex** with at least **1 active black neighbors** are **effective.**

So, the Steiner tree will construct minimum of c minus or maximum of c minus 1 node formation and that is done through a modified depth first search distributed depth first search spanning tree construction and that is the phase 2 approach.

So, here let us go ahead quickly about the phase 2 a black vertex without any dominator is active meaning to say that that if you see. So, the black node without any dominator why because they are independent they are not connected if they connect through another node then it is called a dominator. So, these 2 blacks will have a dominator now at this point of time of at the end of phase 1 these black nodes do not have the dominators if they do not have the dominators then they are active. So, all the black nodes who do not have the dominator they become active initially no black vertex has the dominator and all the hosts are unexplored that after the end of phase 1.

Now phase 2 will try to connect those black nodes with the grey node becoming or gray node are becoming blackened and they are called as dominators. So, those black nodes will now get dominated. So, message m contains a field next which specifies the next host to be explored in dfs formation. So, the grey vertex with at least 1 active black neighbors are effective.

(Refer Slide Time: 29:47)



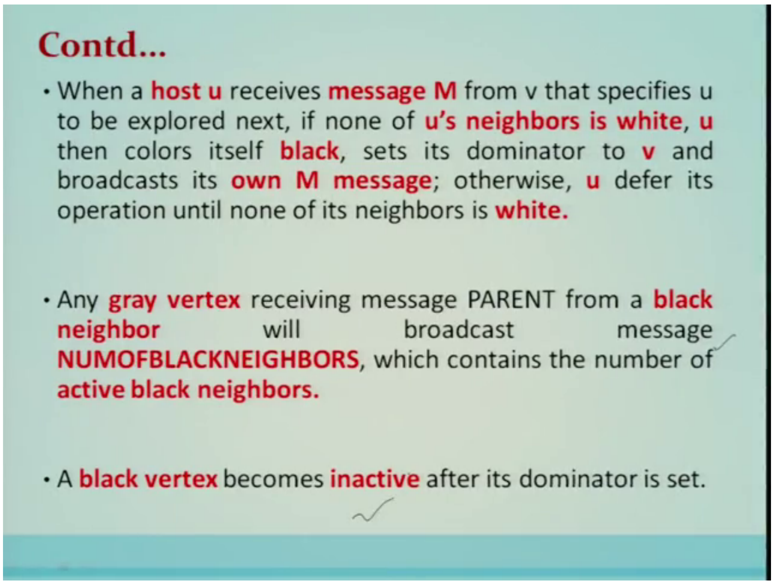
Contd...

- 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.**

Effective means they are going to be a potential for becoming blackening and will contribute to the dominators to the other black nodes. So, if m is built by a black vertex it is next vertex contains an id of unexplored gray neighbors, which connects to the maximum number of active black hosts.

If M is built by the gray vertex it is next field contains an id of an unexplored black neighbors. So, either the black or the gray both basically are able to find out the dominators for the black nodes. So, any black host u receiving M message the first time from the gray vertex v sets it is dominator to v by broadcasting the message parent to it.

(Refer Slide Time: 30:33)



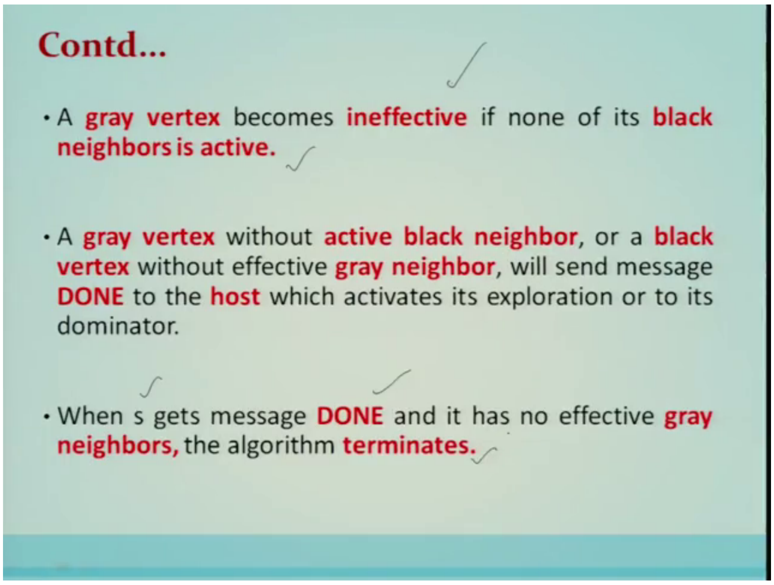
Contd...

- 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.

So, when a host u receives a message M from v that specifies u to be explored next, if none of u's neighbors is white, u then color itself a black, sets it is dominator to v and broadcast it is own message M; otherwise, u defer it is operation until none of it is neighbors is white. Any gray vertex receiving the message parent from the black will become will broadcast the message number of black nodes neighbors which contains the number of active black nodes.

A black node become inactive after it is dominator flag is set a gray becomes ineffective none of it is black neighbors is active.

(Refer Slide Time: 31:13)

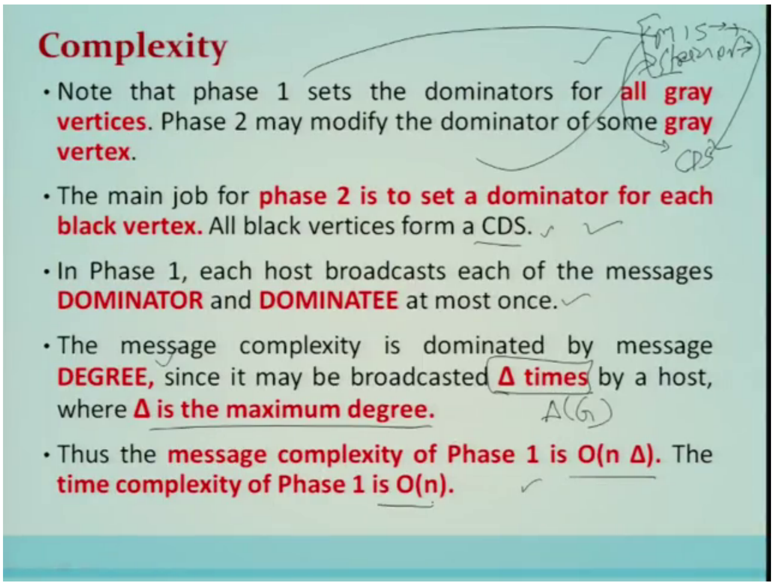


Contd...

- 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.**

So, a gray node without active black neighbor or a black node without active gray never will send the message done to the host which activates it is acceleration or it is or to it is dominator. So, when s that is the leader or the initial node gets done it has no effective a gray neighbors then the algorithm terminates by the construction of a connected dominating set.

(Refer Slide Time: 31:47)



Complexity

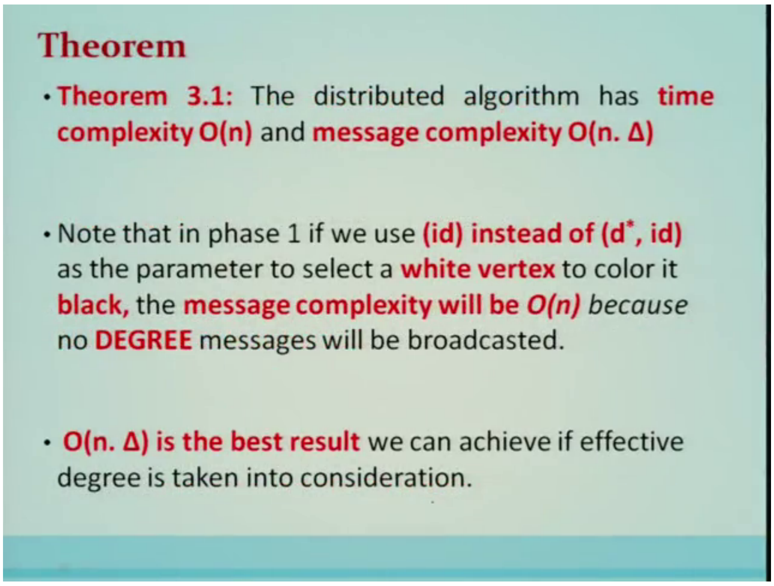
- Note that phase 1 sets the dominators for **all gray vertices**. Phase 2 may modify the dominator of some **gray vertex**.
- The main job for **phase 2 is to set a dominator for each black vertex.** All black vertices form a CDS.
- In Phase 1, each host broadcasts each of the messages **DOMINATOR** 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).**

Now the complexity in phase 1 sets the dominators for all gray vertices phase 2 may modify the dominator of some gray vertex.

So, I told you that in phase 1 it constructs MIS and in phase 2 it will add the additional Steiner tree node and together basically will form the total nodes of connected dominating set. So, we are going to count how many nodes are there. So, the main job of the phase 2 is to set the dominator for each black vertex, all the black vertices whether it is through the Steiner or whether through the MIS total count is the CDS that is what is represented here in phase one each host broadcast each of the message dominator and dominating at most once that we have seen.

So, the message complexity is dominated by the degree because degree means they are sending to their neighbors. Since it has broadcasted that big delta times by the host big delta is the maximum degree of a graph this is the property of a graph. So, the message hence does the message complexity of a phase 1 is of the order n times big delta and the time complexity of the phase 1 is of the order n.

(Refer Slide Time: 33:30)

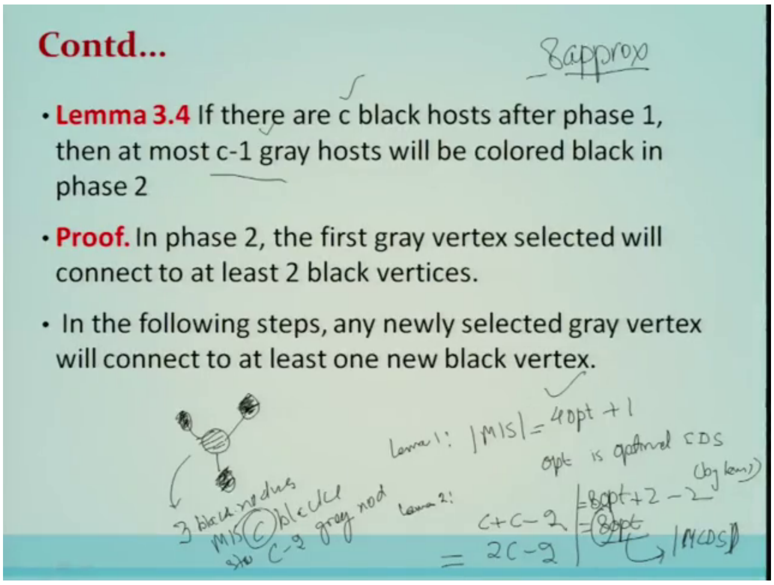


Theorem

- **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.

So, this particular computation of time and message becomes the Theorem.

(Refer Slide Time: 33:53)



Contd...

8approx

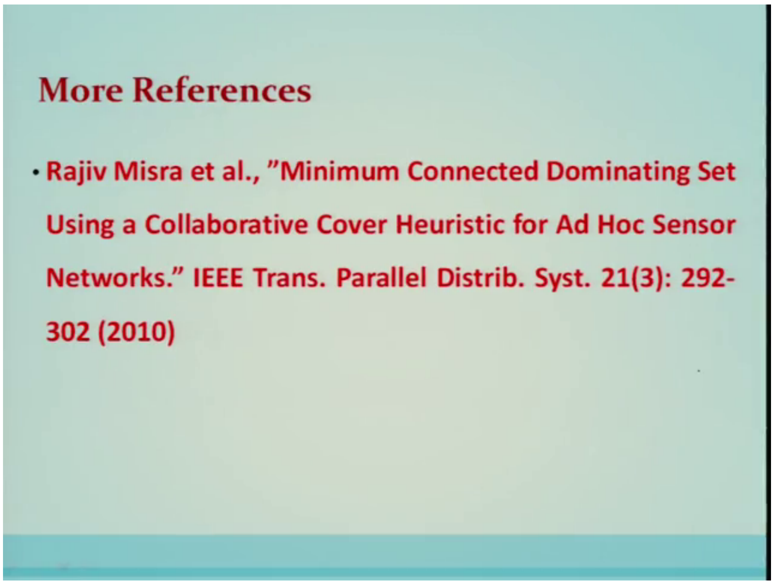
- **Lemma 3.4** If there are c black hosts after phase 1, then at most c-1 gray hosts will be colored black in phase 2
- **Proof.** In phase 2, the first gray vertex selected will connect to at least 2 black vertices.
- In the following steps, any newly selected gray vertex will connect to at least one new black vertex.

Now let us see that there is a lemma which we have seen earlier that the size of MIS is equal to 4 opt plus 1 where opt is the optimal size of connected dominating set this is lemma 1 and the lemma which we have seen earlier.

So, using this particular result another lemma 2 says that if there are c black node after phase 1, then at most c minus 1 gray host will be blackened in phase 2; that means, the Steiner tree will be having c minus t. Now another construction is that if this is the gray this is these are the black nodes and this is the gray node, now if this particular gray node is going to connect 3 different black nodes; that means, 1 gray is going to connect 3 black nodes, then if there are c number of black hosts then these particular how many gray nodes are required is c minus 2 gray nodes are required to connect all the black nodes; that means, c plus c minus 2 that is 2 c minus 2 is basically total size this is MIS and this is the Steiner tree this is the total size.

Now, you know that the size of the MIS from lemma 1 if we basically put it. So, it will become 8 opt plus 2 minus 2 by lemma 1, this will become 8 opt where opt is the size of minimum connected dominating set so; that means, the approximation of this algorithm is 8 approximate algorithm it is that I have proved here in this part of the discussion and this is what is mentioned whatever already proved.

(Refer Slide Time: 36:45)

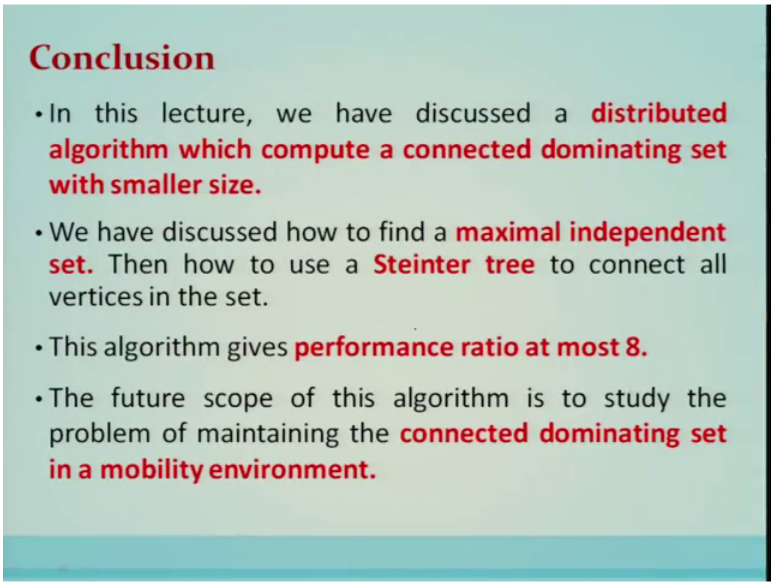


More References

- 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)

Now further References more details of the minimum connected dominating set using other heuristics called collaborative heuristics for ad hoc sensor networks you can refer this particular reference by Dr Rajiv Misra that is I triple E transaction parallel and distributed computing system that is published in 2010 2010.

(Refer Slide Time: 37:10)



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 **Steinter 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.**

Conclusion in this lecture we have discussed a distributed algorithm which compute a connected dominating set of a smaller size we have discussed how to find maximal independent set, then we have used about the use of Steiner tree to connect all the vertices in the set this algorithm gives the performance ratio of 8 the future scope of this algorithm is to study the problem of maintaining connected dominating set in environment.

Thank you.