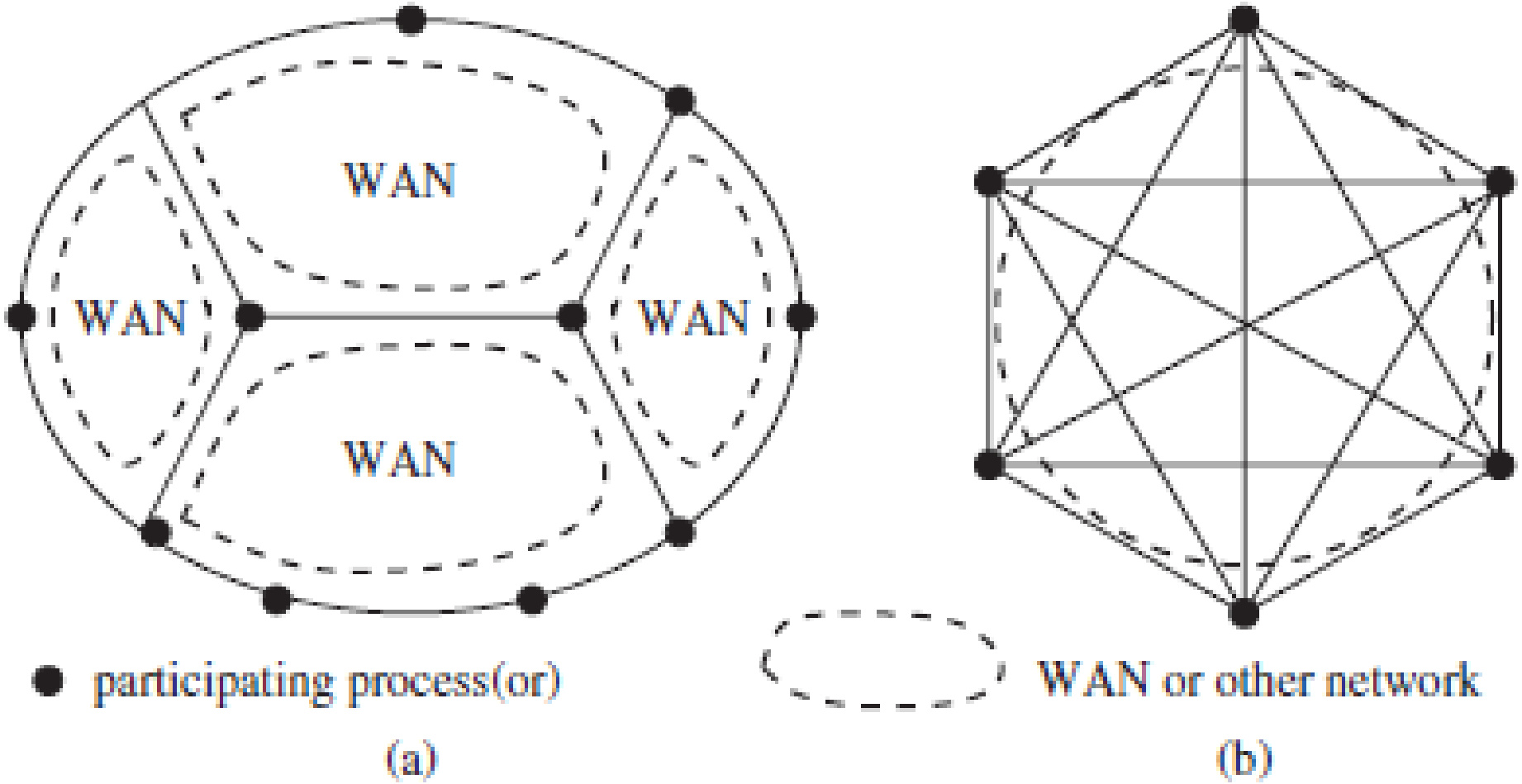
BITS Pilani
Hyderabad Campus

Distributed Computing (CS 3 contd..)

Distributed Computing Terminology and Basic Algorithms

Prof. Geetha
Associate Professor, Dept. of Computer Sc. & Information Systems
BITS Pilani Hyderabad Campus
geetha@hyderabad.bits-pilani.ac.in

Topological abstraction in distributed computing



Centralized and distributed algorithms

- **centralized algorithm –**
  - predominant amount of work is performed by one (or possibly a few) processors
  - other processors play a relatively smaller role in accomplishing the joint task
  - other processors usually request or supply information

- **distributed algorithm -** each processor plays an equal role in sharing the message overhead, time overhead, and space overhead

Symmetric and asymmetric algorithms

- **symmetric algorithm -** algorithm in which all the processors execute the same logical functions

- **asymmetric algorithm** - algorithm in which different processors execute logically different functions

Synchronous and asynchronous systems

- **synchronous system -** satisfies the following properties:
  - a known upper bound on the message communication delay
  - a known bounded drift rate for the local clock of each processor with respect to real-time
  - a known upper bound on the time taken by a process to execute a logical step in the execution

- **asynchronous system** - a system in which none of the above 3 properties of synchronous systems are satisfied

Failure models

- **Failure model -** specifies the manner in which the component(s) of the system may fail
- **Process failure models:**

➢ **Fail-stop** - a properly functioning process may fail by stopping execution from some instant, other processes can learn that the process has failed

➢ **Crash** - a properly functioning process may fail by stopping to function from any instance, other processes do not learn of this crash

➢ **Byzantine or malicious failure, with authentication** - a process may exhibit any arbitrary behavior, authentication can be used to detect forgery

➢ **Byzantine or malicious failure** - a process may exhibit any arbitrary behavior, no authentication techniques are applicable

Notation & Definitions

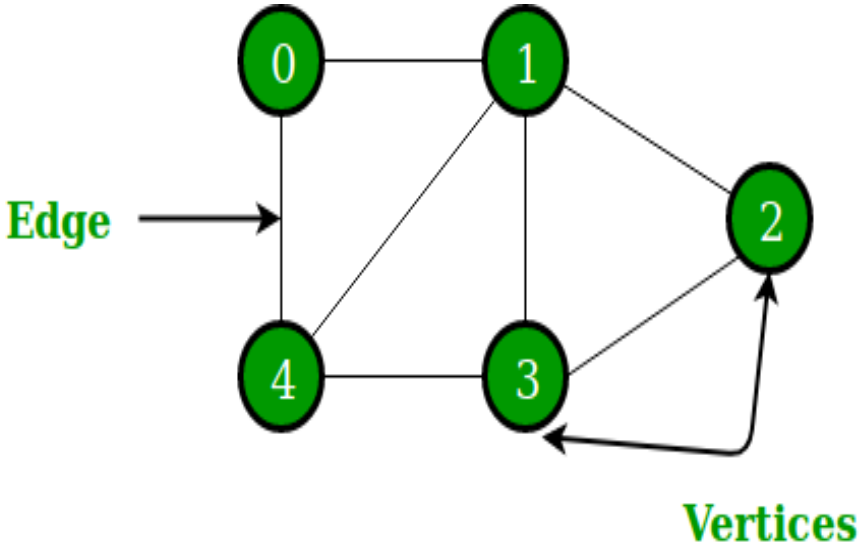
- undirected unweighted graph G = (N, L) represents topology
- $n = |N|$
- $l = |L|$
- diameter of a graph –
  - minimum number of edges that need to be traversed to go from any node to any other node
  - diameter = $max_{i,\, j \in N}$ {length of the shortest path between i and j}
  - For a tree embedded in the graph, its depth is denoted as h

Model

- Node Information:
  - Adjacent links
  - Neighboring nodes.
  - Unique identity.
- Graph model: undirected graph.
  - Nodes V
  - Edges E
  - Diameter D

Graph data structure

- A Graph is a non-linear data structure consisting of nodes and edges
- The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph
- The formal definition is : *A Graph consists of a finite set of vertices(or nodes) and set of Edges which connect a pair of nodes*
- In the above Graph, the set of vertices V = {0,1,2,3,4} and the set of edges E = {01, 12, 23, 34, 04, 14, 13}.
- Graphs are used to solve many real-life problems. Graphs are used to represent networks
- In distributed computing, the network of processes/processors can be modeled as a graph

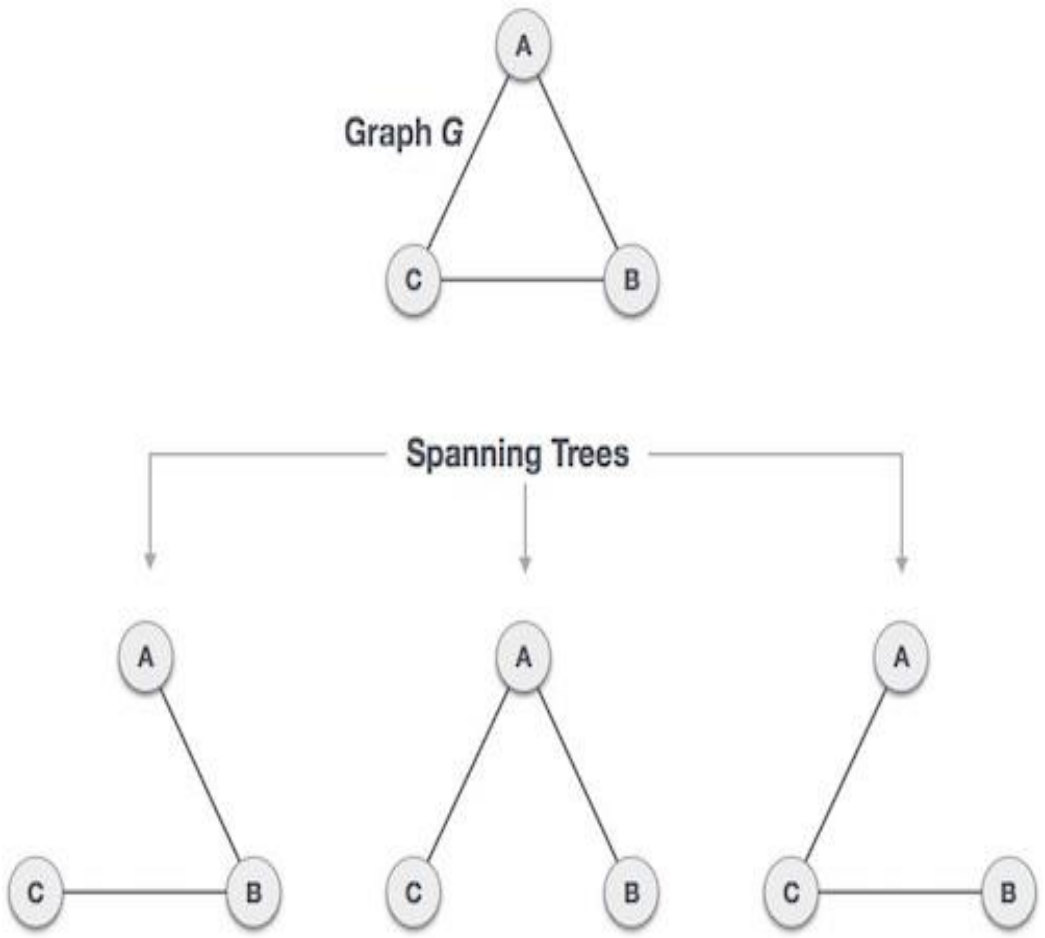


Constructing Minimal Spanning Trees (MST) from Graphs

- A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. Hence, a spanning tree does not have cycles and it cannot be disconnected..

- Thus every connected and undirected Graph G has at least one spanning tree. A disconnected graph does not have any spanning tree, as it cannot be spanned to all its vertices

- The distributed minimum spanning tree (MST) problem involves the construction of a minimum spanning tree by a distributed algorithm, in a network where nodes communicate by message passing



Model Definition

- Synchronous Model:
  - There is a global clock.
  - Packet can be sent only at clock ticks.
  - Packet sent at time *t* is received by *t+1.*
- Asynchronous model:
  - Packet sent is eventually received.

Complexity Measures

- **Message complexity:**
  - **Number of messages sent (worst case).**
  - **Usually assume "small messages".**
- **Time complexity**
  - **Synchronous Model: number of clock ticks**
  - **Asynchronous Model:**
    - **Assign delay to each message in (0,1].**
    - **Worse case over possible delays**

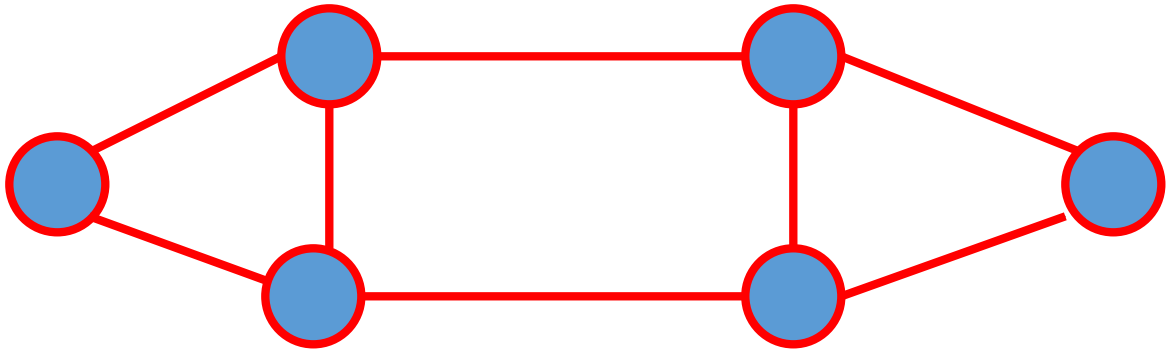
Problems in Distributed Computing

- **Broadcast - Single initiator:**
  - **The initiator has an information.**
  - **At the end of the protocol all nodes receive the information**
  - **Example : Topology update.**
- **Spanning Tree:**
  - **Single/Many initiators.**
  - **On termination there is a spanning tree.**
- **BFS tree:**
  - **A minimum distance tree with respect to the originator.**

Basic Flooding: Algorithm

- **Initiator:**
  - **Initially send a packet p(info) to all neighbors.**
- **Every node:**
  - **When receiving a packet p(info):**
    - **Sends a packet p(info) to all neighbors.**

Basic Flooding: Example



Basic Flooding - complexity

- Time Complexity
  - Synchronous (and asynchronous):
    - Node at distance *d* receives the info by time *d.*
- **Message complexity:**
  - **There is no termination!**
  - **Message complexity is unbounded!**
  - **How can we correct this?**

Synchronous Single-Initiator Spanning Tree Algorithm using Flooding

- algorithm proceeds in rounds (hence, synchronous)
- assumes a designated root node
- root initiates
  - the algorithm
  - a flooding of QUERY messages to identify tree edges
- processes must produce a spanning tree rooted at the root node
- each process P_i ($P_i \neq$ root) should output its own parent for the spanning tree
- spanning tree is a BFS spanning tree

Synchronous Single-Initiator Spanning Tree Algorithm using Flooding

Local Variables maintained at each P_i

- **int** *visited*
- **int** *depth*
- **int** *parent*
- **set of int** *Neighbors*

The root initiates a flooding of QUERY messages in the graph to identify tree edges. The parent of a node is that node from which a QUERY is first received; if multiple QUERYs are received in the same round, one of the senders is randomly chosen as the parent

Synchronous Single-Initiator Spanning Tree Algorithm using Flooding – contd..

Initially at each P_i

- *visited* = 0
- *depth* = 0
- *parent* = NULL
- *Neighbors* = set of neighbors

Synchronous Single-Initiator Spanning Tree Algorithm using Flooding contd..

Algorithm for P_i

Round r = 1

if P_i = root **then**

visited = 1

depth = 0

send QUERY to *Neighbors*

if P_i receives a QUERY message **then**

visited = 1

depth = r

parent = root

plan to send QUERYs to *Neighbors* at next round

Synchronous Single-Initiator Spanning Tree Algorithm using Flooding

Algorithm for P_i

Round r > 1 and r <= *diameter*

if P_i planned to send in previous round **then**

  P_i sends QUERY to *Neighbors*

if *visited* = 0 **then**

  if P_i receives QUERY messages **then**

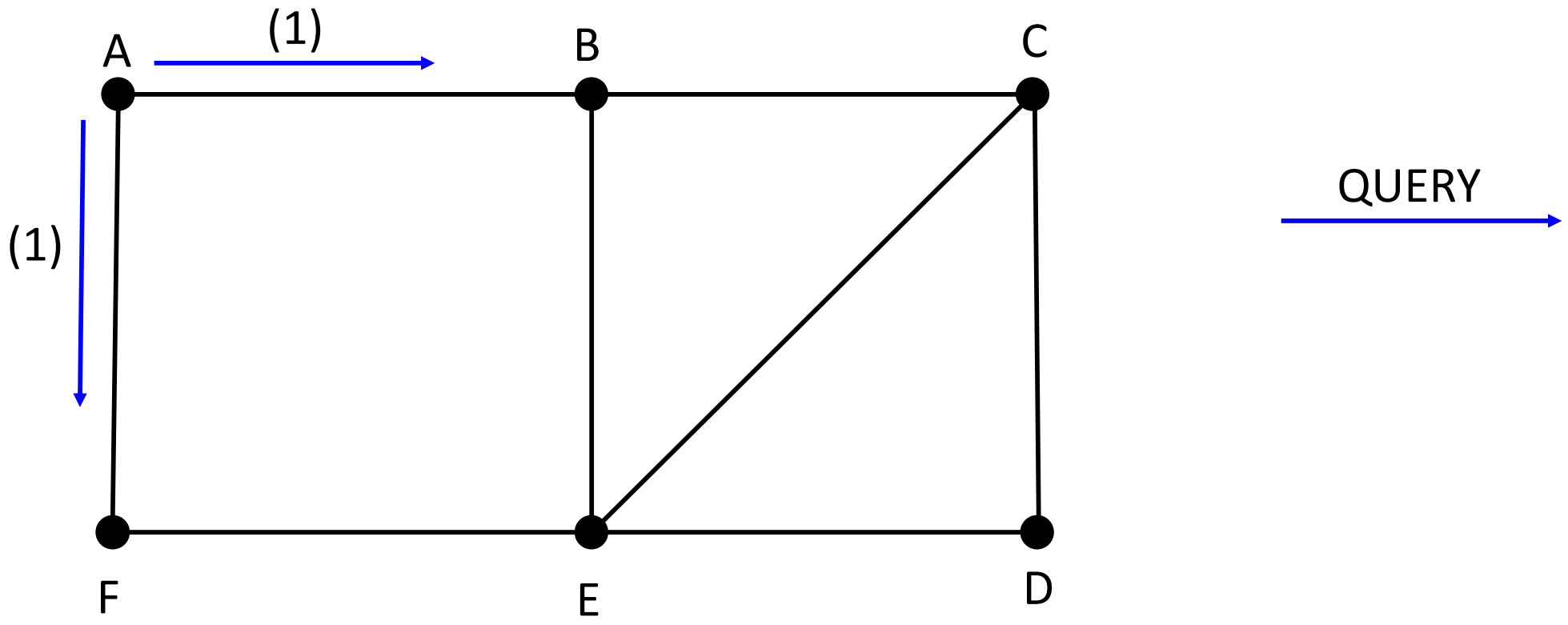
    visited = 1

    depth = r

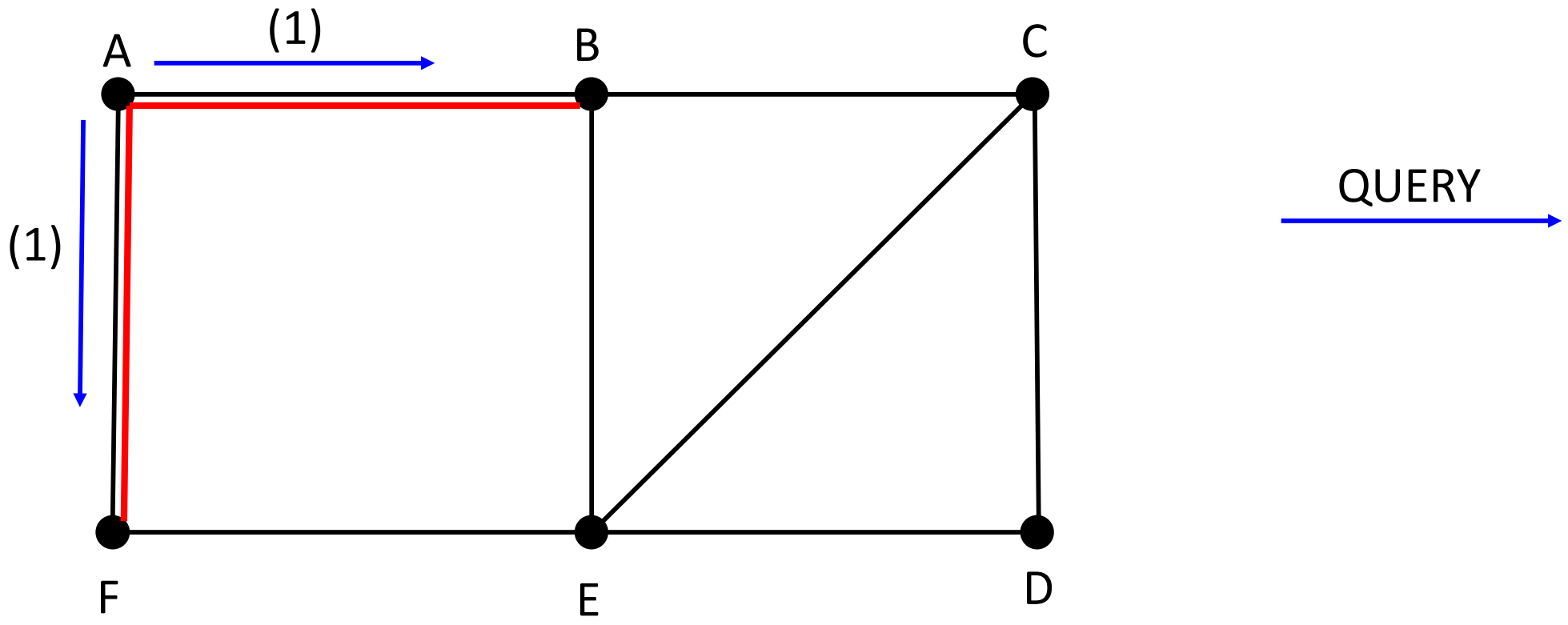
    parent = any randomly selected node from which QUERY was received

    plan to send QUERY to *Neighbors* \ {senders of QUERYs received in r}

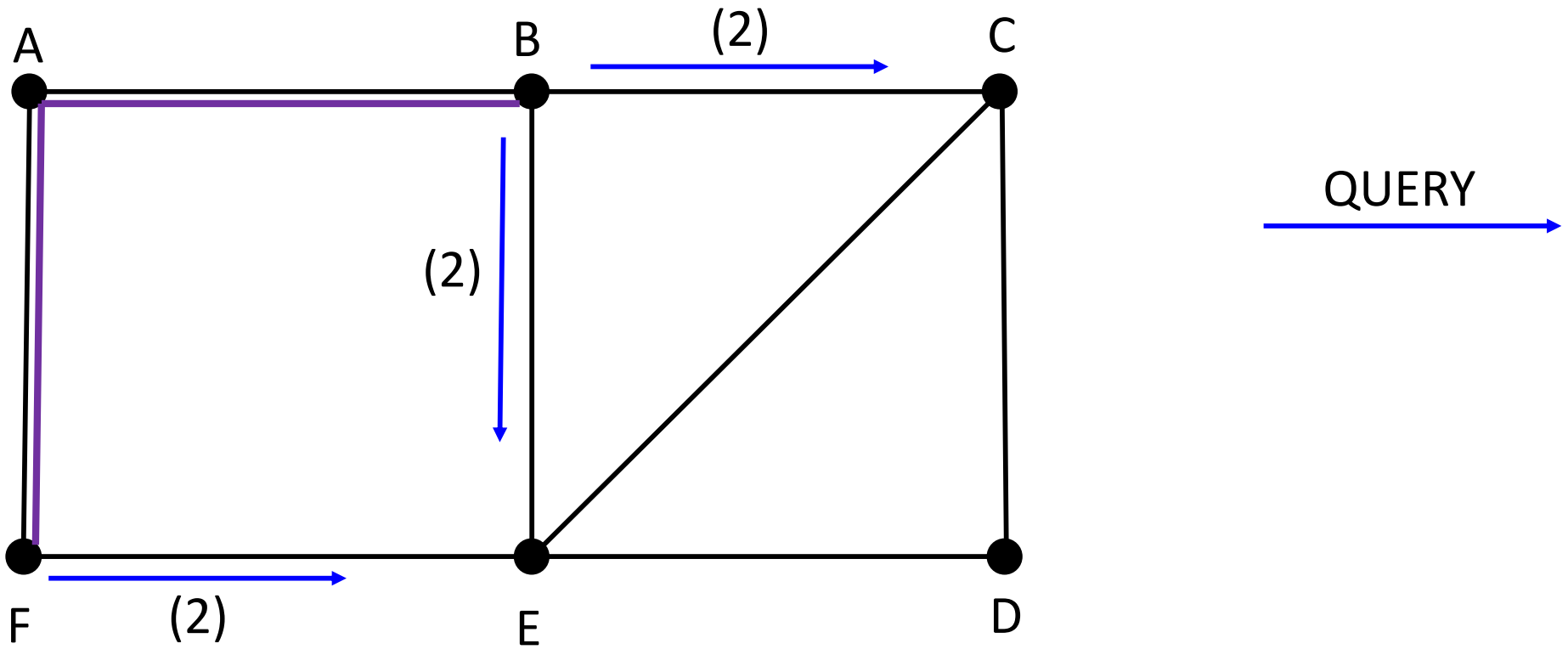
Synchronous Single-Initiator Spanning Tree Algorithm using Flooding



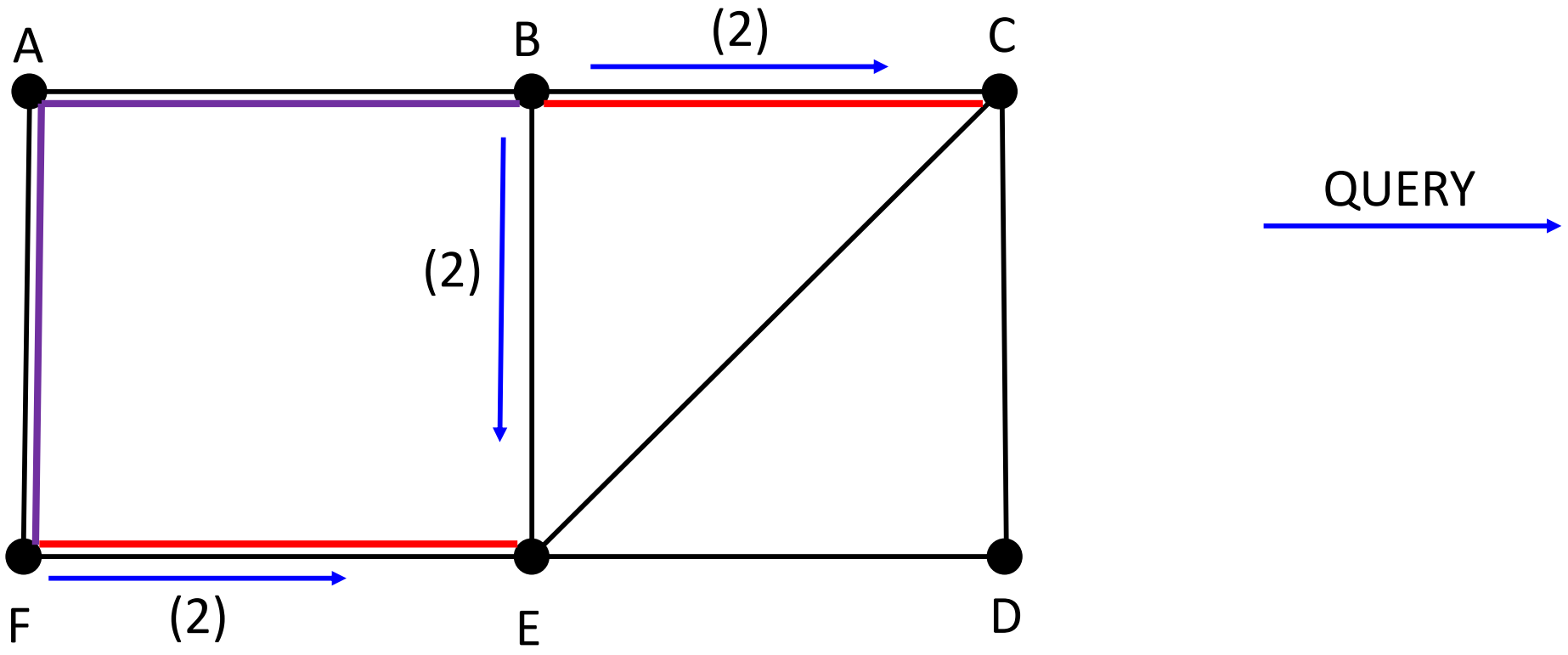
Synchronous Single-Initiator Spanning Tree Algorithm using Flooding contd..



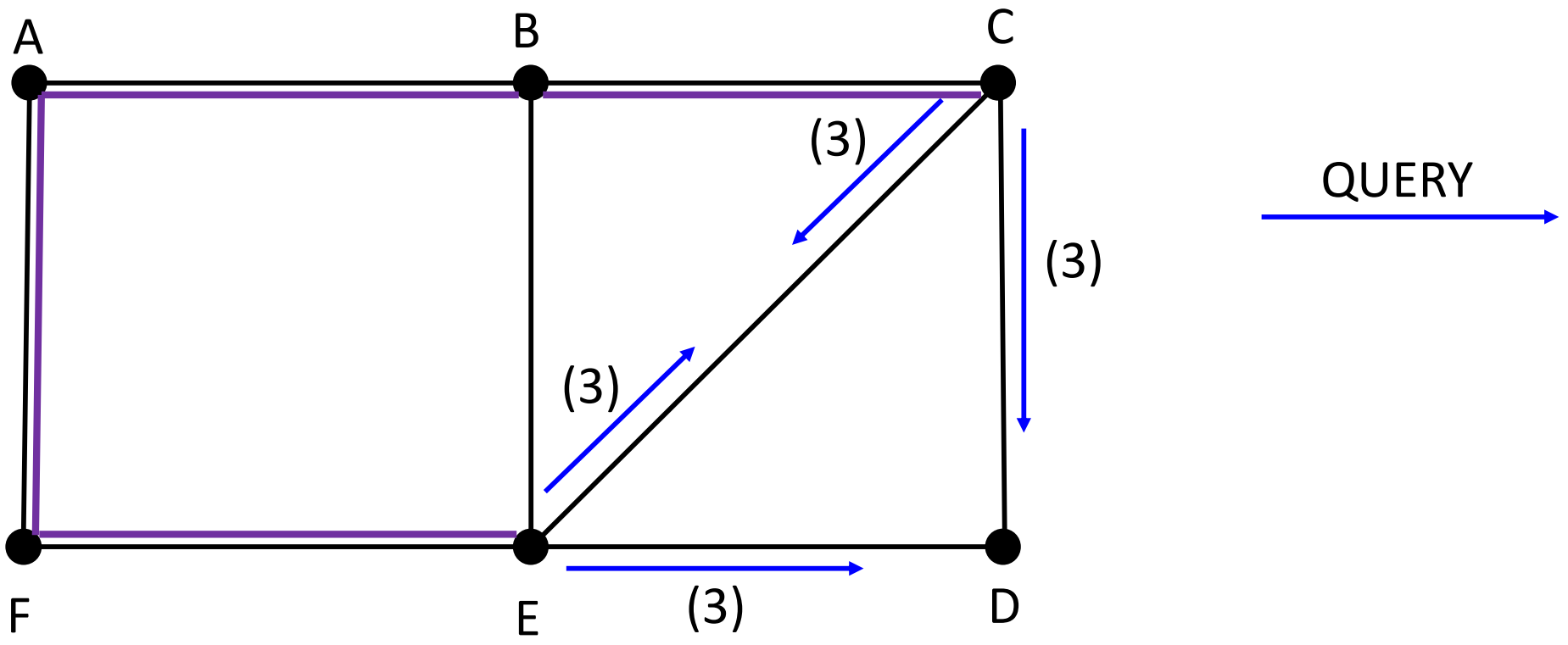
Synchronous Single-Initiator Spanning Tree Algorithm using Flooding contd..



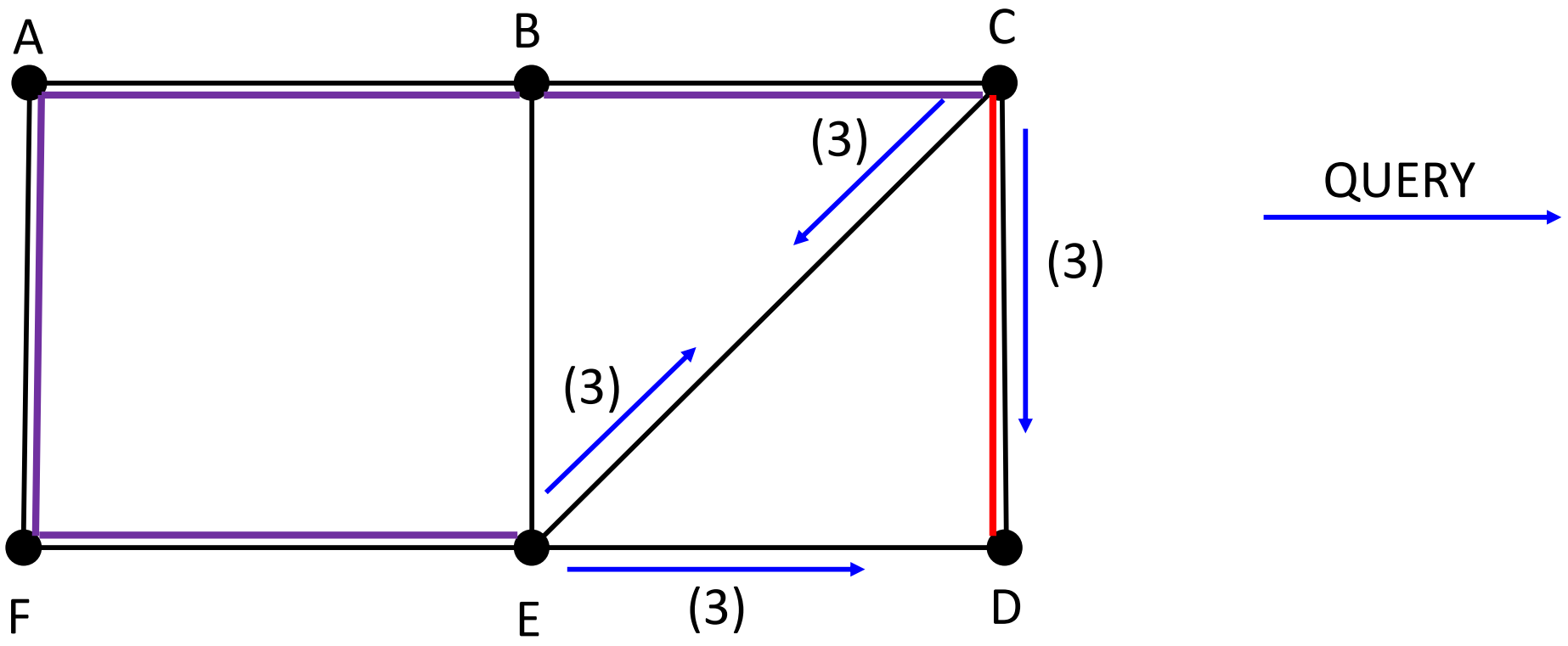
Synchronous Single-Initiator Spanning Tree Algorithm using Flooding contd..



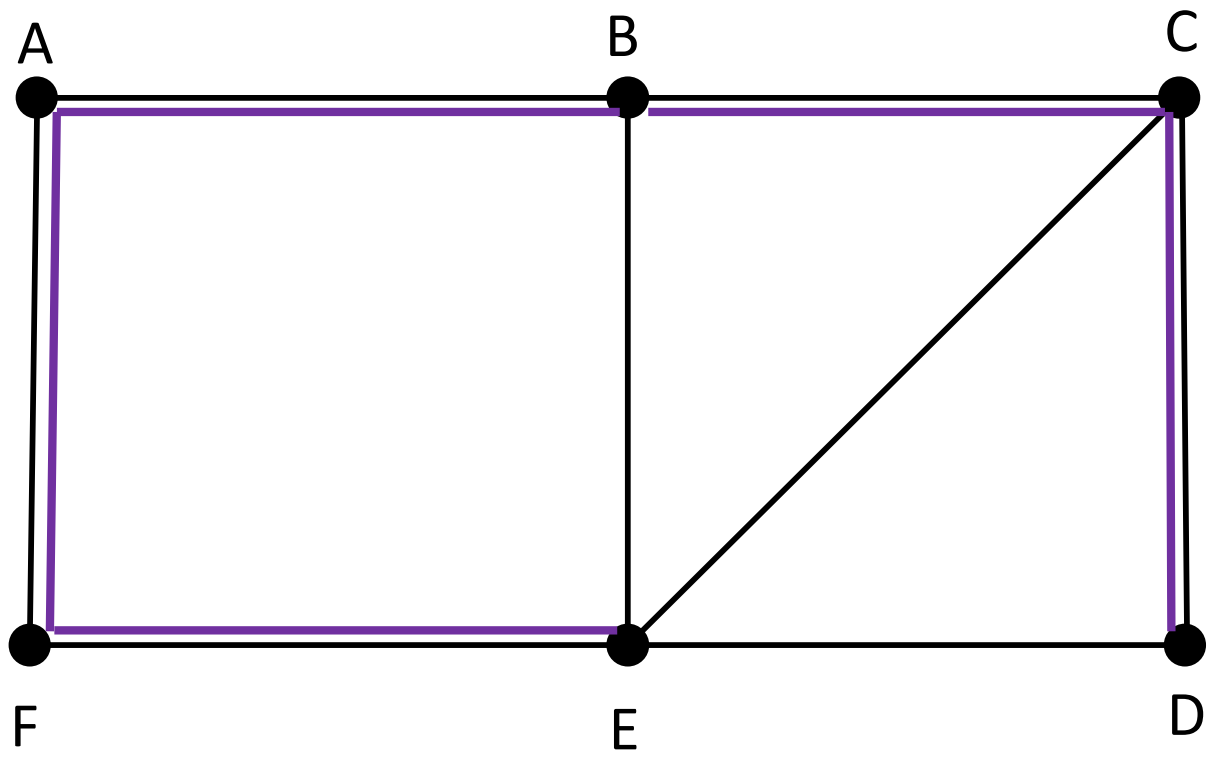
Synchronous Single-Initiator Spanning Tree Algorithm using Flooding contd..



Synchronous Single-Initiator Spanning Tree Algorithm using Flooding contd..

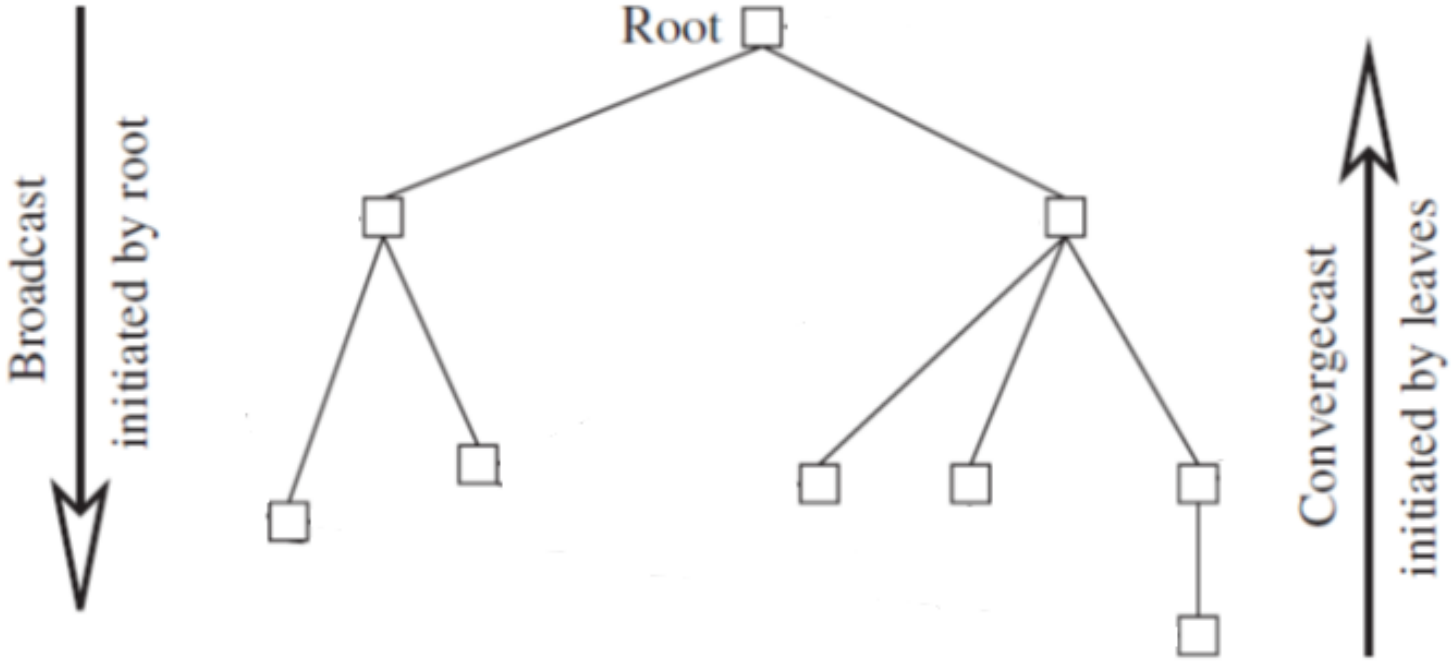


Synchronous Single-Initiator Spanning Tree Algorithm using Flooding contd..



Broadcast and Convergecast on a Tree

A spanning tree is useful for distributing (via a broadcast) and collecting (via a convergecast) information to/from all the nodes



Broadcast and Convergecast on a Tree

A broadcast algorithm on a spanning tree can be specified by 2 rules:

- ***BC1:***
  - The root sends the information to be broadcast to all its children.
  - Terminate.
- ***BC2:***
  - When a (non-root) node receives information from its parent, it copies it and forwards it to its children.
  - Terminate.

Broadcast and Convergecast on a Tree

- A convergecast algorithm collects information from all the nodes at the root node in order to compute some global function

- It is initiated by the leaf nodes of the tree, usually in response to receiving a request sent by the root using a broadcast

Broadcast and Convergecast on a Tree

The convergecast algorithm is specified as follows:

- ***CVC1:***
  - Leaf node sends its report to its parent.
  - Terminate.
- ***CVC2:***
  - At a non-leaf node that is not the root: When a report is received from all the child nodes, the collective report is sent to the parent.
  - Terminate.
- ***CVC3:***
  - At the root: When a report is received from all the child nodes, the global function is evaluated using the reports.
  - Terminate.

Broadcast and Convergecast on a Tree

Complexity

- each broadcast and each convergecast requires $n - 1$ messages
- each broadcast and each convergecast requires time equal to the maximum height h of the tree, which is $O(n)$

Applications:

- computation of minimum of integer variables associated with the nodes of an application
- leader election

Synchronizers

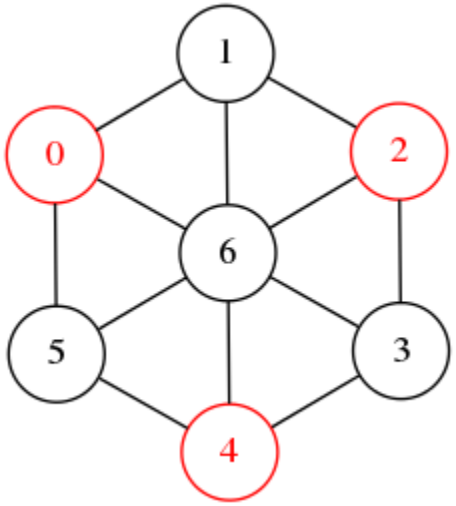
- It is difficult to design algorithms for asynchronous systems
- **solution** – simulate synchronous behavior on an asynchronous system
- **synchronizers** - transformation algorithms to run synchronous algorithms on asynchronous systems
- synchronizer signals to each process when it is sure that all messages to be received in the current round have arrived and it is safe to proceed to the next round

Synchronizers

- **Simple synchronizer:**
  - requires each process to send every neighbor only one message in each round
  - if no message is to be sent in the synchronous algorithm, an empty dummy message is sent in the asynchronous algorithm
  - if more than one messages are sent in the synchronous algorithm, they are combined into one message in the asynchronous algorithm
  - in any round, when a process receives a message from each neighbor, it moves to the next round
- **α, β, and γ synchronizers:** (rounds, rooted spanning tree; n/w of clusters)
  - use the notion of process safety
  - a process i is safe in round r if all messages sent by i in round r have been received

Maximal Independent Set

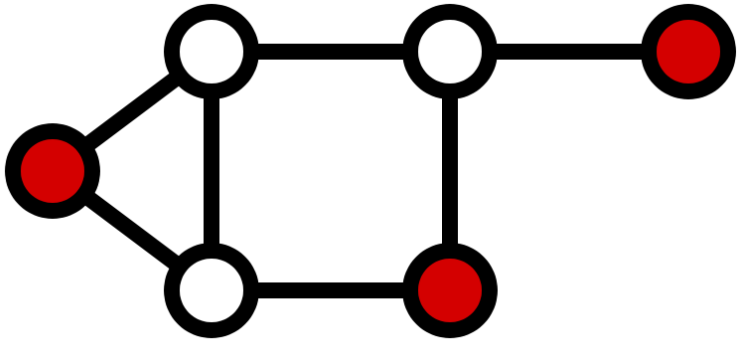
- For a graph (N, L), an independent set of nodes N', where N'⊂ N, is such that for each i and j in N', (i, j)∉ L
- An independent set N' is a maximal independent set if no strict superset of N' is an independent set
- A graph may have multiple maximal independent sets
- Adjacent nodes must not be chosen



source - http://www.martinbroadhurst.com/greedy-max-independent-set-in-c.html

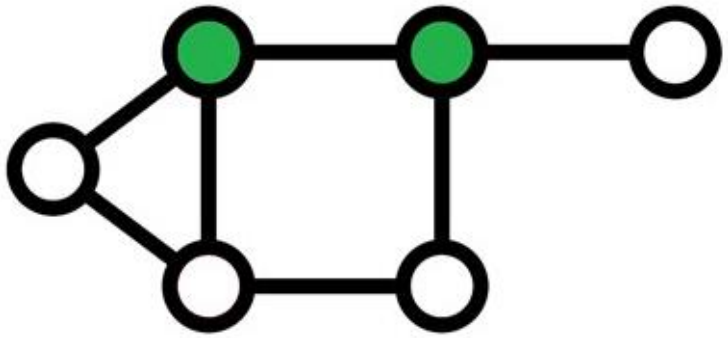
Connected dominating set

- A dominating set of graph (N, L) is a set N’ ⊆ N such that each node in N \N’ has an edge to some node in N’



source - https://en.wikipedia.org/wiki/Dominating_set

- A connected dominating set (CDS) of (N, L) is a dominating set N’ such that the subgraph induced by the nodes in N’ is connected



source - https://www.google.com/search?q=connected+dominating+set&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi48raDoPPcAhVENY8KHd_1AkMQ_AUICigB&biw=1920&bih=966#imgrc=B3d-Z8XurdIQLM:

Reference

- Ajay D. Kshemkalyani, and Mukesh Singhal, Chapter 5, “Distributed Computing: Principles, Algorithms, and Systems”, Cambridge University Press, 2008 (Reprint 2013).

Recap Quiz

1. In a graph, the minimum number of edges that need to be traversed to go from any node to any other node is called
(a) Diameter (b) radius (c) perimeter (d) circumference

2. The Single-Initiator Spanning Tree Algorithm using Flooding is synchronous because
(a) There is a global clock (b) algo works in rounds (c) there are local clocks (d) spanning tree is used

3. A Convergecast algorithm is initiated by ___ node of the tree
(a) Root (b) intermediate (c) leaf (d) none

4. Let h be the maximum height of the tree. Then the time required for broadcast and convergecast algorithms will be
(a) h (b) logh (c) 2h (d) h^2

5. The number of maximal independent sets a graph can have in graph algorithms is
(a) 0 (b) 1 (c) any number (d) nil

Recap Quiz - key

Q1	Q2	Q3	Q4	Q5
a	b	c	a	c