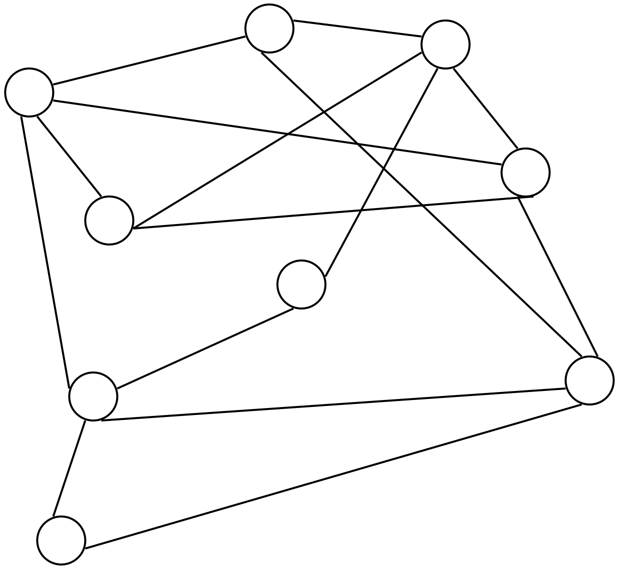
MOVING AND COMPUTING

IN

DISCRETE SPACES

GRASTA/MAC Tutorial 2015

Netscape

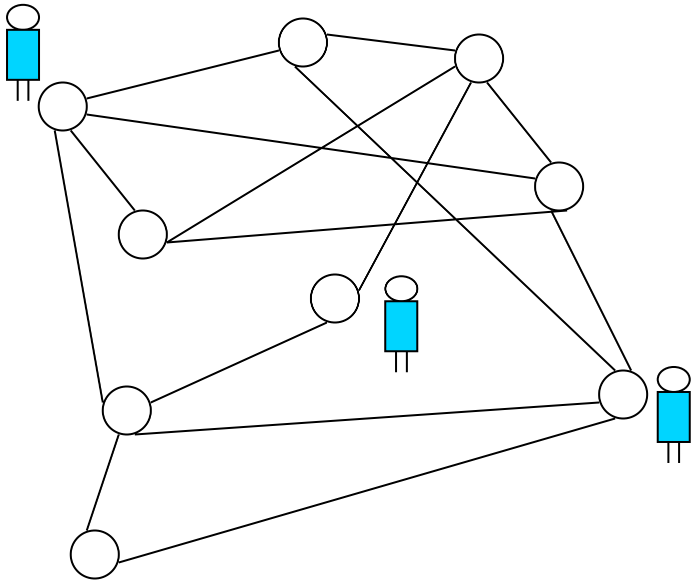


Graph G

node (site , host)

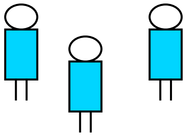
edge (link , channel)

Netscape

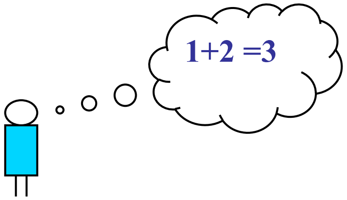


Netscape inhabited by computational mobile entities called agents

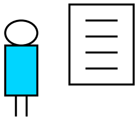
Each Agent



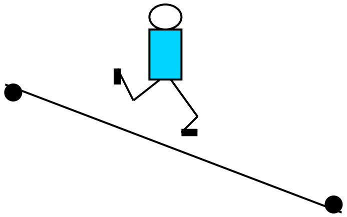
Has computing capabilities



Has limited storage



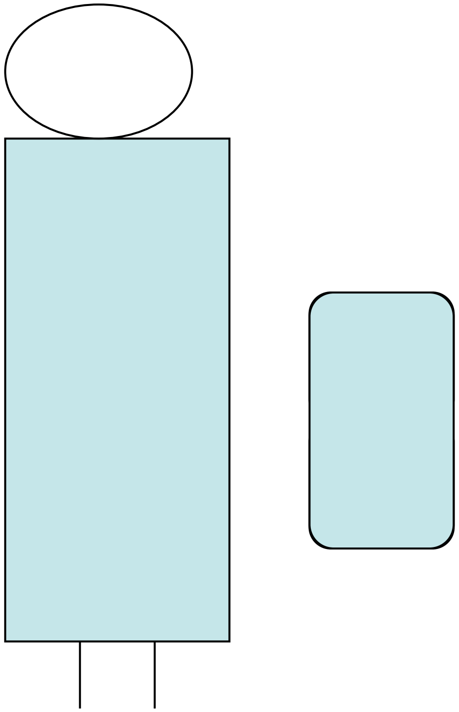
Can move from node to neighboring node



Computational Models : Existing Systems (some)

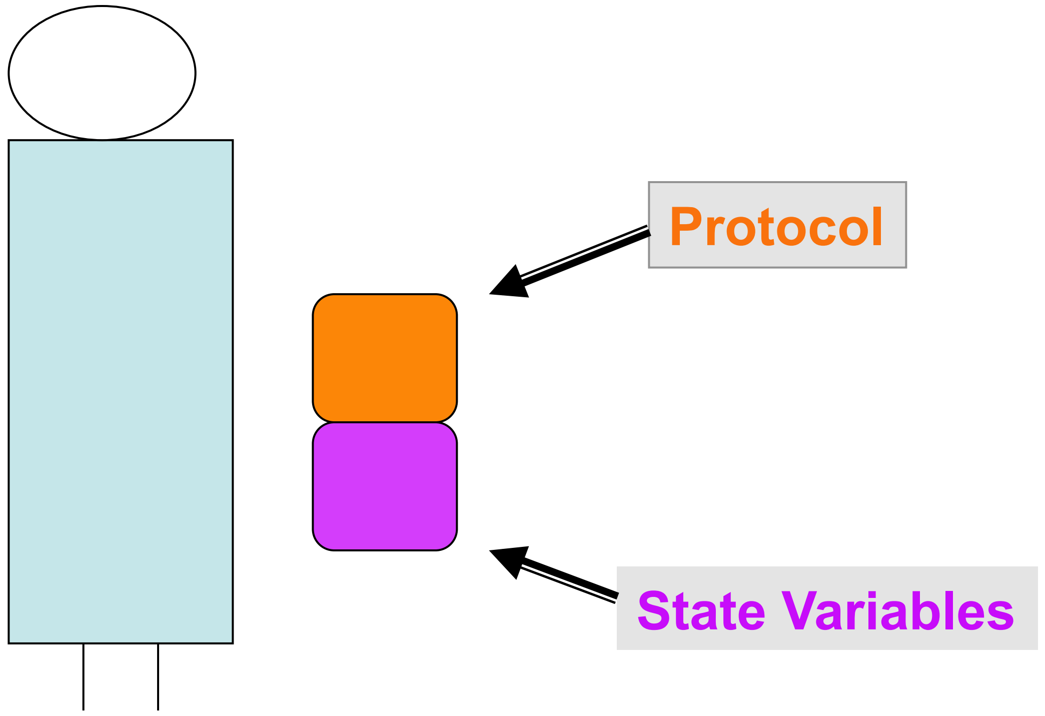
Name	Description	Developer	Language	Application
Concordia	Framework for agent development	Mitsubishi E.I.T.	Java	Mobile computing, Data base
Aglet	Java Class libraries	IBM, Tokyo	Java	Internet
Agent Tcl	Transportable agent system	R. Gray, U Dart.	Tcl Tk	Information management
Odyssey	Set of Java Class libraries	General Magic	Telescript	Electronic commerce
OAA	**O**pen **A**gent **A**rchitecture	SRI International, AI	C, C-Lisp, Java, VB	General purpose
Ara	**A**gent for **R**emote **A**ction	U Kaiserslautern	C/C++, Tcl, Java	Partially connected c. D.D.B.
Tacoma	**T**romso **a**nd **C**ornel **M**oving **A**gent	Norway & Cornell	**C**, **UNIX-**based ,	Client/Server model issues / OS support
Voyager	Platform for distributed applic.	ObjectSpace	Java	Support for agent systems
AgentSpace	Agent building platform	Ichiro Sato, O. U.	Java	General purpose
Mole	First Java-Based MA system	Stuttgart U. Germany	Java, **UNIX**-based	General purpose
MOA	**M**obile **O**bject and **A**gents	OpenGroup, UK	Java	General purpose
Kali Scheme	Distributed impl. of Scheme	NEC Research I.	Scheme	Distributed data mining, load balancing
The Tube	mobile code system	David Halls, UK	Scheme	Remote execution of Scheme
Ajanta	Network mobile object	Minoseta U.	Java	General purpose
Knowbots	Research infrastructure of MA	CNRI	Python	Distributed systems / Internet
AgentSpace	Mobile agent framework	Alberto Sylva	Java	Support for dynamic and dist. Appl.
Plangent	Intelligent Agent system	Toshiba Corporation	Java	Intelligent tasks
JATlite	Java Agent framework dev/KQML	Standford U.	Java	Information retrievial, Interface agent
Kafka	Multiagent libraries for Java	Fujitsu Lab. Japan	Java, UNIX-based	General purpose
Messengers	Autonomous messages	UCI	C (Messenger-C)	General purpose

Computational Models : Existing Systems



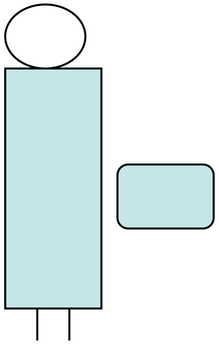
Computational Models : Existing Systems

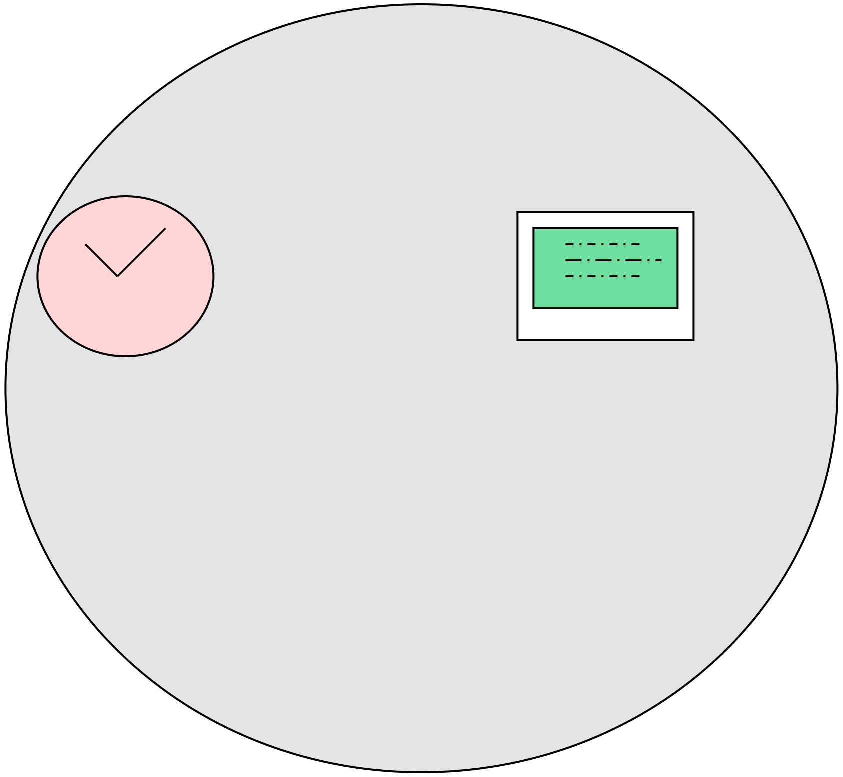
MOBILE CODE



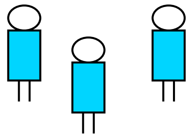
Computational Models : Existing Systems

Execute the code





Agents



Have the same behavior **(execute the same protocol)**

Collectively

they perform some task **(solve a problem)**

Computability

What problems can be solved ?

Under what conditions ?

Computability and Complexity

What problems can be solved ?

Under what conditions ?

At what cost ?

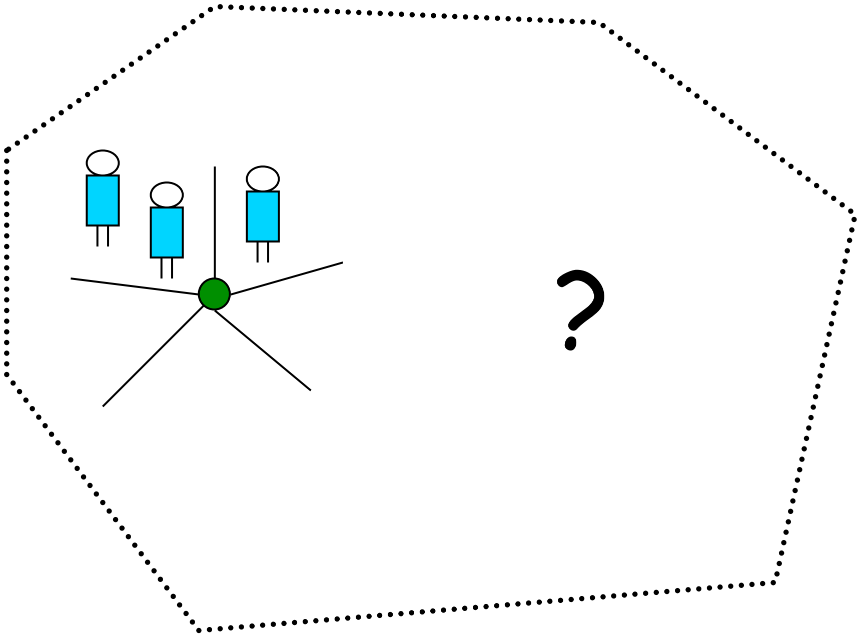
Tasks / Problems

Search

Exploration

Map Construction

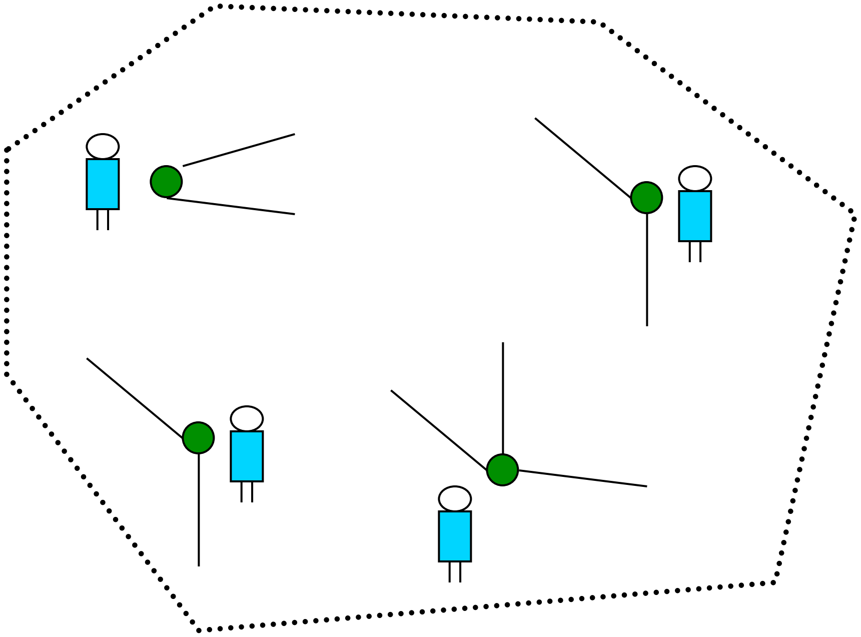
Map Verification



Tasks / Problems

RendezVous

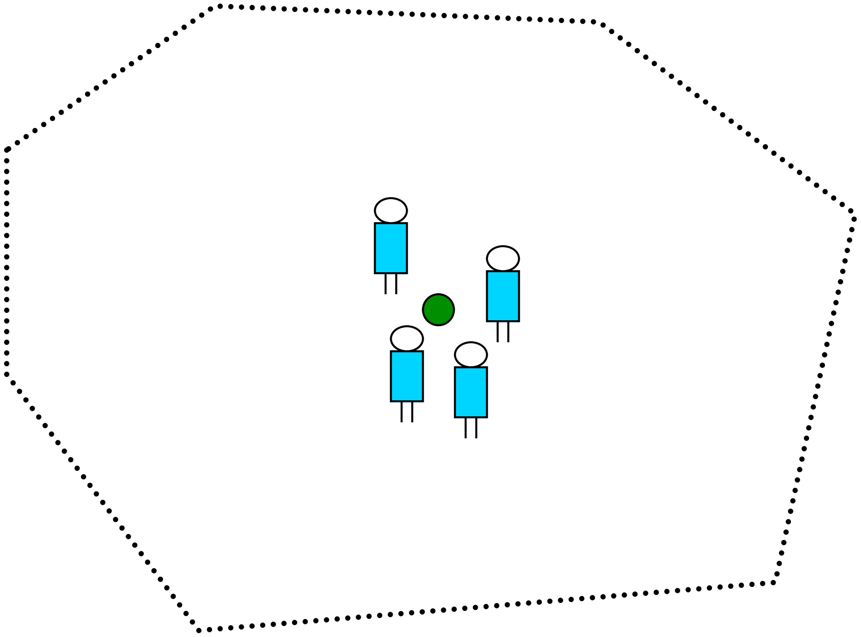
Gathering



Tasks / Problems

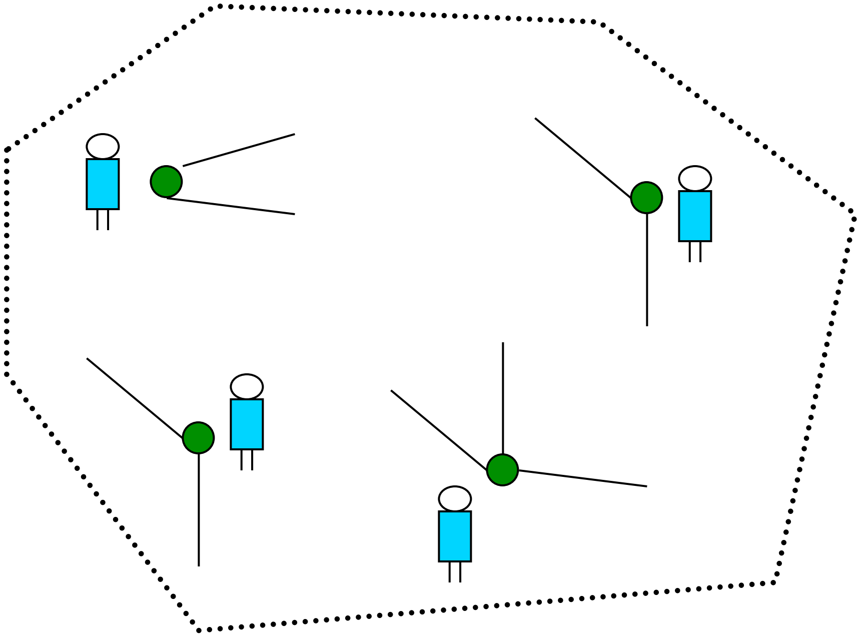
RendezVous

Gathering



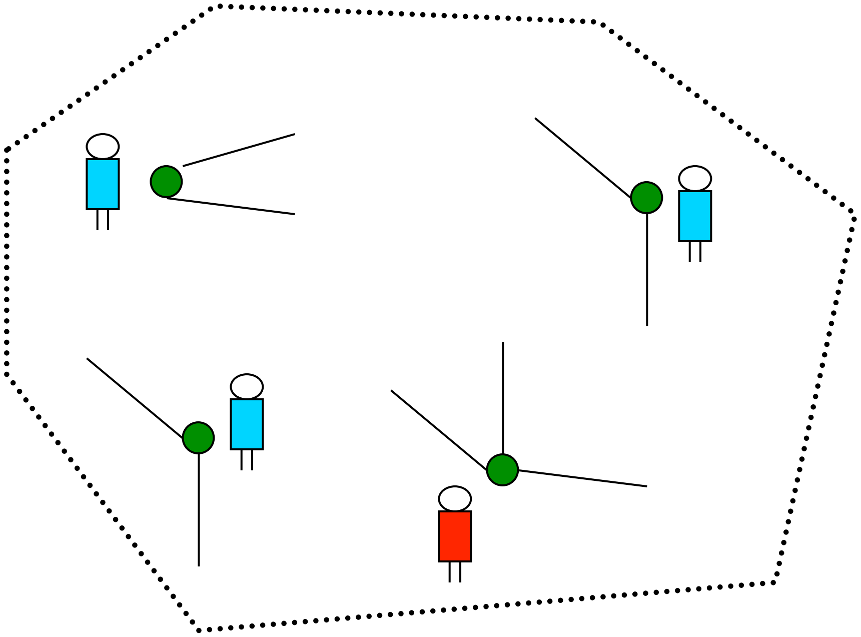
Tasks / Problems

Election



Tasks / Problems

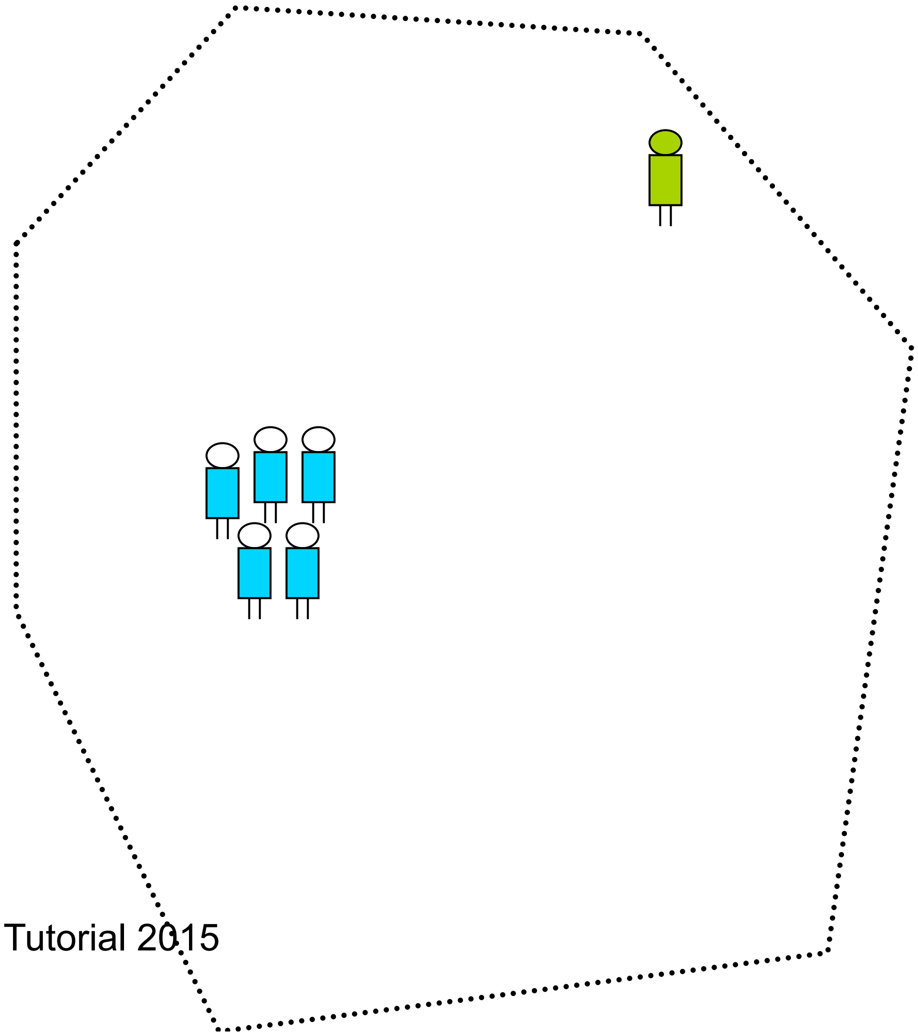
Election



Tasks / Problems

Intruder Capture

Decontamination



Exploration/ Map Construction/ Map Verification

Shannon [JMF’51]

Blum, Kozen [FOCS’78]

Dudek, Jenkin, Milios, Wilkes [*Robotics and Automation* ’91]

Bender, Slonim [FOCS ’94]

Betke, Rivest, Singh [*Machine Learning* ’95]

Bender, Fernandez, Ron, Sahai, Vadhan [STOC ’98]

Deng, Papadimitriou [*J. Graph Theory* ’99]

Panaite, Pelc [*J. Algorithms* ’99]

Awerbuch, Betke, Rivest, Singh [*Information and Comp. ’99*]

Panaite, Pelc [*Networks* ’00]

Albers, Henzinger [*SIAMJC* ’00]

Duncan, Kobourov, Kumar [SODA ’01]

… …. ….

Exploration/ Map Construction/ Map Verification

… … …

Das, Flocchini, Kutten, Nayak, Santoro [*Theo.Comp.Sci.*' 07]

Fraigniaud, Ilcinkas, Pelc [*Inform. Comput ‘08*]

Ilcinkas [*Theo.Comp.Sci. ‘08*]

Chalopin, Flocchini, Mans, Santoro [WG '10]

Sudo,Baba,Nakamura,Ooshita,Kakugawa, Masuzawa [WRAS ‘10]

Flocchini, Ilcinkas, Pelc, Santoro [*Theo.Comp.Sci.* ‘10]

Brass, Cabrera, Gasparri, Xiao [*IEEE Tr. Robotics'11*]

D'Angelo, Di Stefano,Navarra [DISC ‘12]

Dieudonne, Pelc [*Trans.Alg.* '14]

Kralovic [SOFSEM *‘14*]

Pelc, Tiane [*I.J.Found. Comp. Sci.* '14]

AND MANY MANY MORE …

Rendezvous / Election

Alpern [*SIAM J Cont. Optimization 95*]
Anderson, Weber [*J Applied Probability 99*]
Yu, Yung [ICALP 96]
Alpern, Boston, Essegarer [*J Appl. Probability 99*]
Howard et al [*Operation research 99*]
Barrière, Flocchini, Fraigniaud, Santoro [SPAA03]
Dessmark, Fraigniaud, Pelc [ESA 03]
Kranakis, Krizac, Santoro,Sawchuk [ICDCS' 03]
Barrière,Flocchini,Fraigniaud,Santoro [*T.Comp.Sys. 05*]
Dessmark,Fraigniaud,Kowalski,Pelc [*Networks 06*]
Chalopin [*Theo.Comp.Sci.' 08*]
Klasing, Markou, Pelc [*Theo.Comp.Sci.' 08*]
Czyzowicz, Pelc, Labourel [*ACM Trans. Alg. '13*]
D'Angelo, Di Stefano, Klasing, Navarra [*Theo.Comp.Sci. 14*]

AND MANY MORE …

Dangerous Explorations / Dangerous Rendezvous

Dobrev, Flocchini, Prencipe, Santoro [*Algorithmica* '06]
Dobrev, Flocchini, Prencipe, Santoro [*Distributed Comp.* '06]
Czyzowicz, Kowalski,Markou,Pelc [*Found. Infor.* '06]
Dobrev,Flocchini,Kralovic,Prencipe,Ruzicka,Santoro [*Networks* '06]
Dobrev, Flocchini, Santoro [*Theo.Comp.Sci.* '06]
Klasing, Markou, Radzik,Sarracco [*Theo.Comp.Sci.* '07]
Czyzowicz, Kowalski, Markou, Pelc [*Comb.Prob.Comp.* '07]
Chalopin, Das, Santoro [DISC '07]
Klasing,Markou,Radzik,Sarracco [*Networks* 08]
Balamohan, Flocchini, Miri, Santoro [*Discrete Math.* '11]
Chalopin, Das, Labourel, Markou [DISC '11]
Flocchini, Ilcinkas, Santoro [*Algorithmica* '12]
D'Emidio, Frigioni, Navarra [ADHOCNOW '13]
Balamohan, Dobrev, Flocchini, Santoro [*Theo.Comp.Sci.* '14]

AND MORE …

Intruder Capture / Network Decontamination

Breisch [*Southwestern Cavers*, 1967]

Makedon and Sudborough [ICALP ' 83]

Kirousis, Papadimitriou [*J. Theo. Com.Sc.*, 1986]

Megiddo, Hakimi, Garey, Johnson, Papadimitriou [*JACM*, 1988]

Bienstock and Seymour [*J. of Algo.*, 1991]

Lapaugh [*JACM*, 1993]

Ellis, Sudborough, and Turner [*Info & Comp.*, 1994]

Fomin, Golovach [*SIAM J.Disc.Math.*, 2000]

Thilikos [*Disc. Applied Maths*, 2000]

....

Agents can JUMP …

AND MORE …

Intruder Capture / Network Decontamination

Breisch [*Southwestern Cavers*, 1967]
Barriere, Flocchini, Fraigniaud, Santoro [*SPAA*’ 02]
Barriere, Fraigniaud, Santoro, Thilikos [*WG*’ 03]
Blin, Fraigniaud, Nisse, Vial [*SIROCCO*’ 06]
Fraigniaud,Nisse [LATIN’ 06]
Fraigniaud,Nisse [*WG*’ 06]
F.L.Luccio [FUN ‘07]
Flocchini, Huang, F.L.Luccio [*Int.J.Found.Comp.Sci.*’ 07]
Luccio, Pagli, Santoro [*Int.J.Found.Comp.Sci.* ‘07]
Flocchini, Huang, F.L.Luccio [*Networks* 08]
Dereniowski [*SIAM J. Discrete Math.* ‘12]
Barriere,Flocchini,Fomin,Nisse,Santoro,Thilikos [*Inf. Comp.* ‘12]
Borowiecki, Dereniowski, Kuszner [DISC ‘14]
Flocchini, Luccio, Pagli, Santoro [*Discrete Appl. Math.* ‘15]
Nisse, Soares [*Discrete Applied Mathematics*’15]

AND MORE …

Computability and Complexity

COMPUTATIONAL MODELS

COST MEASURES

Computability and Complexity

➡ **Many factors to be considered**

Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent

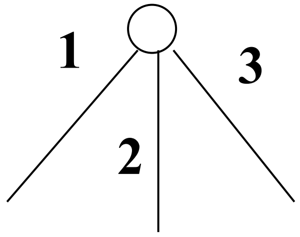
Example

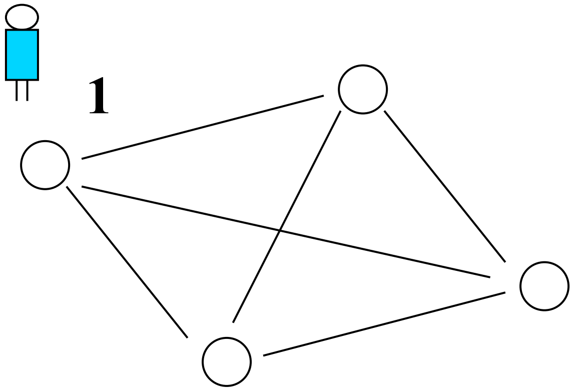
Map Construction

a map of the graph must be constructed by at least one agent

Exploration

every node must be visited by at least one agent

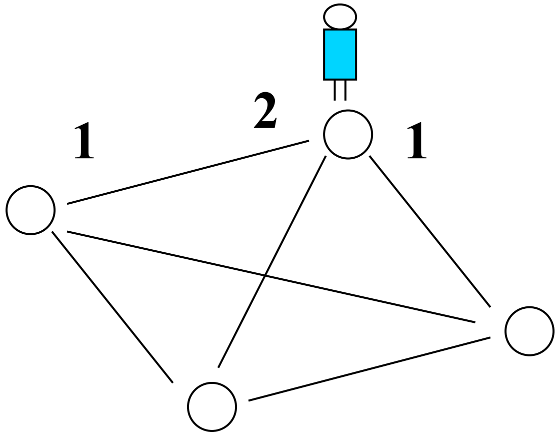




Example

Map Construction a map of the graph must be constructed by at least one agent

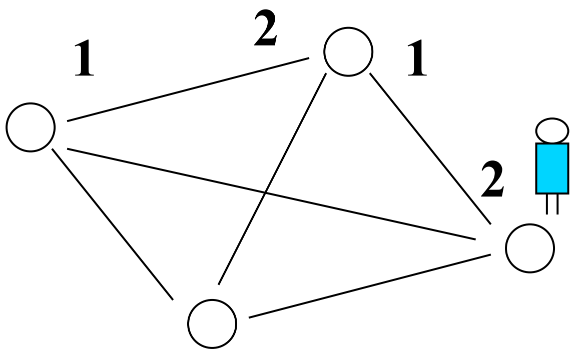
Exploration every node must be visited by at least one agent



Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent

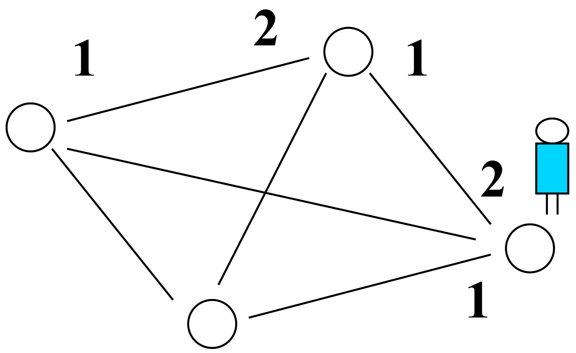


Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent

Exploration completed !

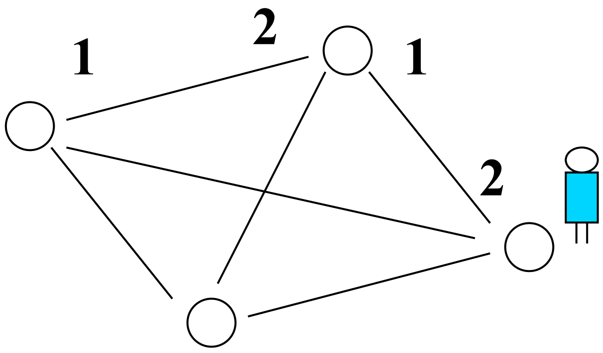


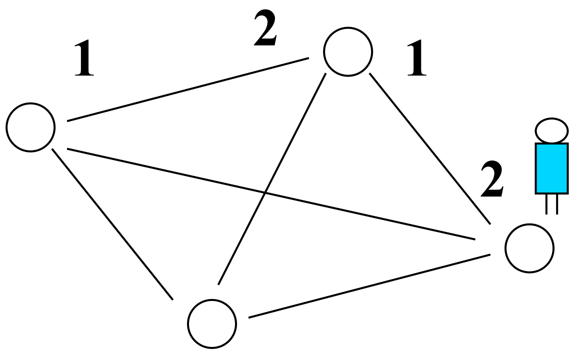
how does it know ?

Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent

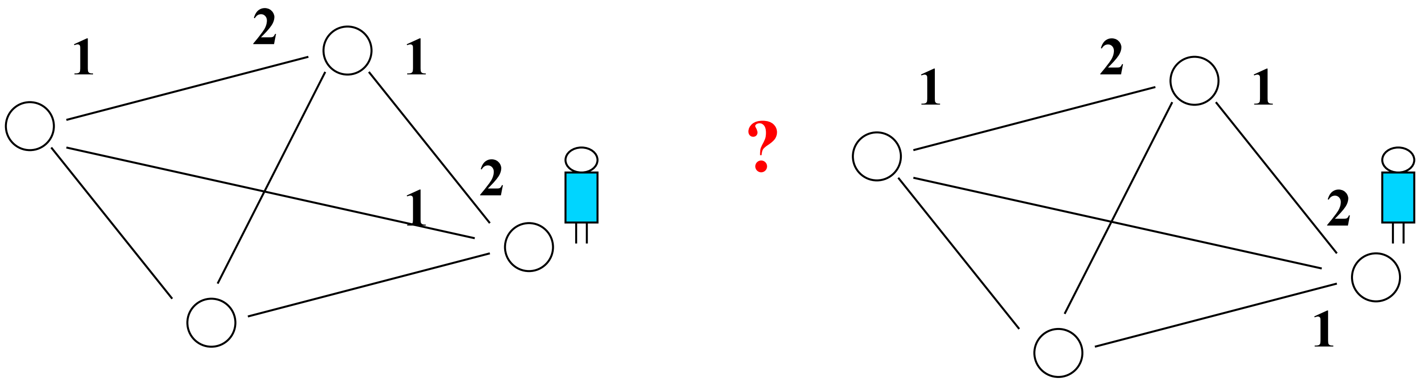




Example

Map Construction a map of the graph must be constructed by at least one agent

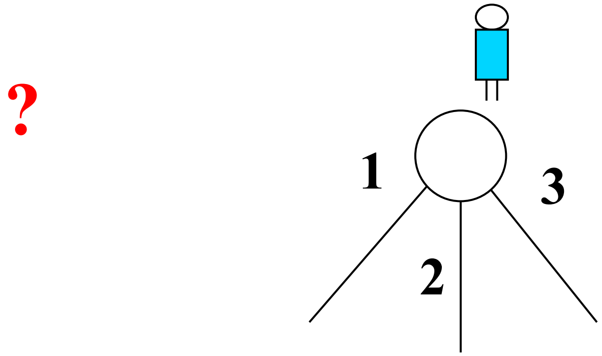
Exploration every node must be visited by at least one agent



Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent

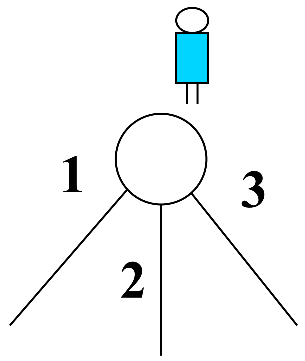


Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent

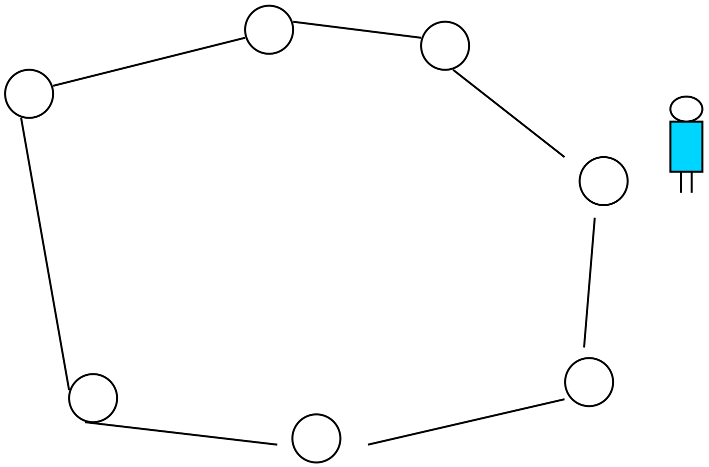
Have I been here before ?



Example

Map Construction a map of the graph must be constructed by at least one agent

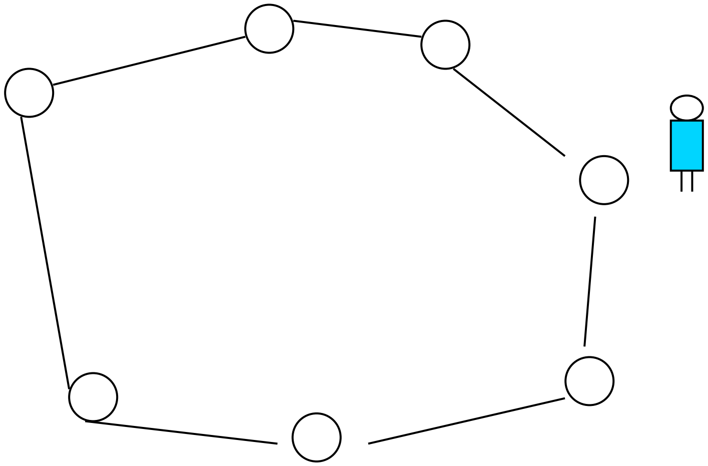
Exploration every node must be visited by at least one agent



Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent

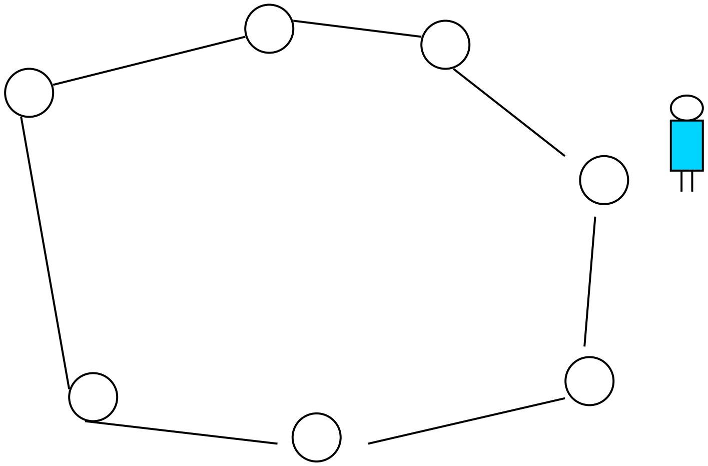


GO AROUND !

Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent



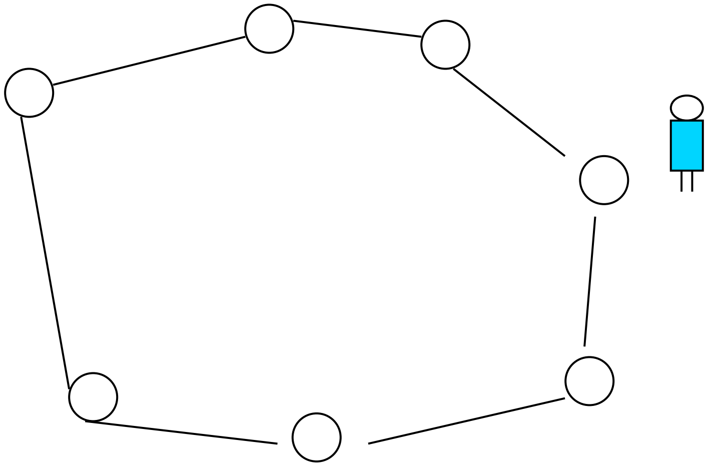
GO AROUND !

When can I stop ?

Example

Map Construction a map of the graph must be constructed by at least one agent

Exploration every node must be visited by at least one agent



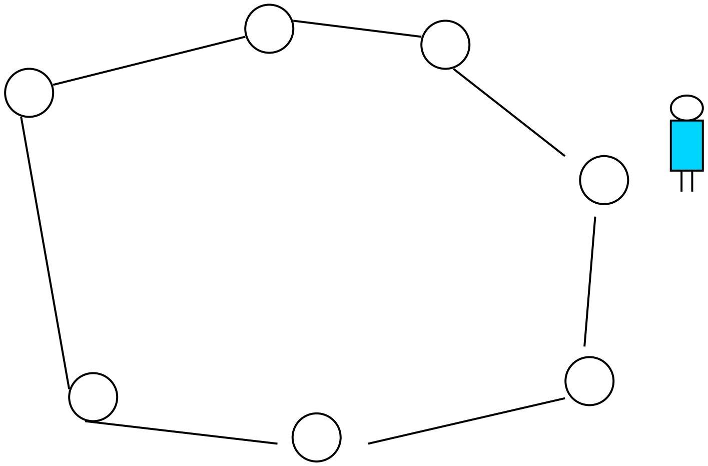
GO AROUND !

When can I stop ?

NEVER !

Example

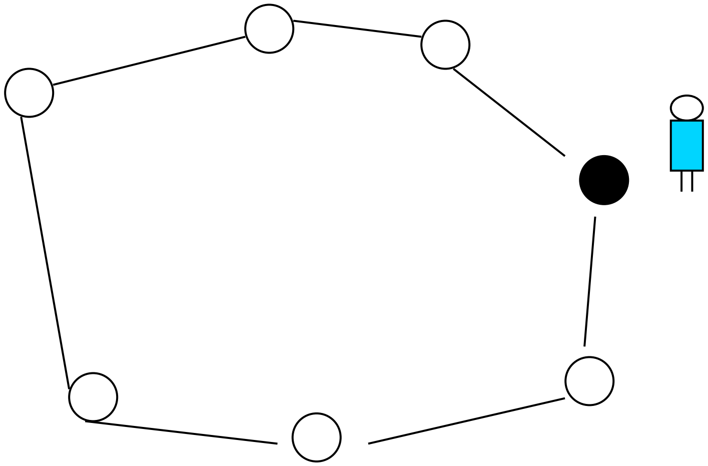
If it can mark the node:



GO AROUND !

Example

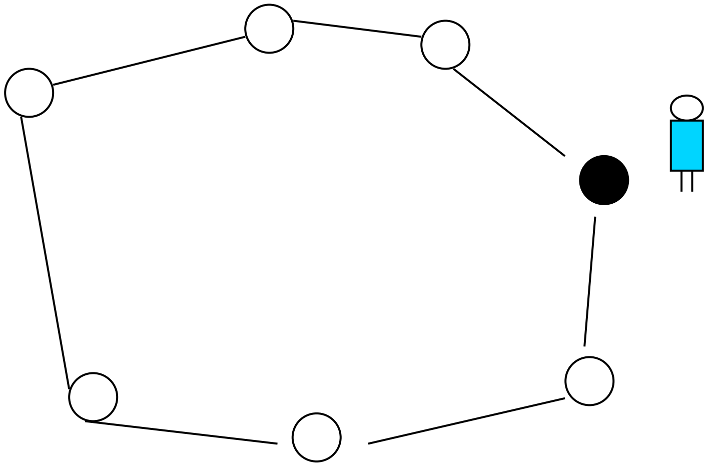
If it can mark the node:



GO AROUND
until find marked node

Example

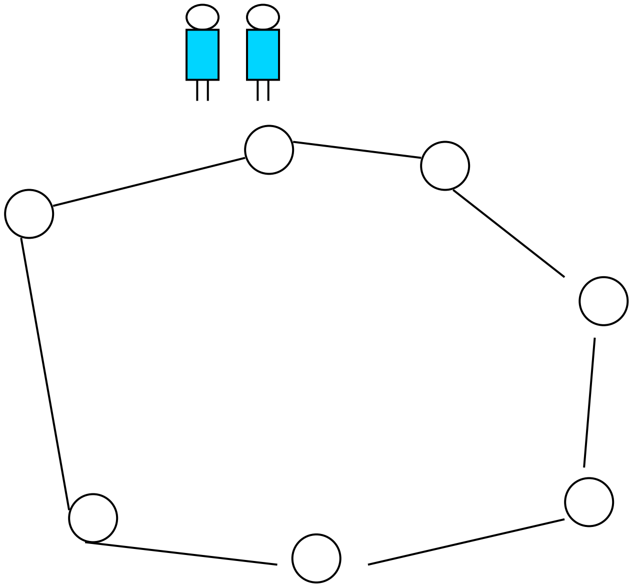
If it can mark the node:



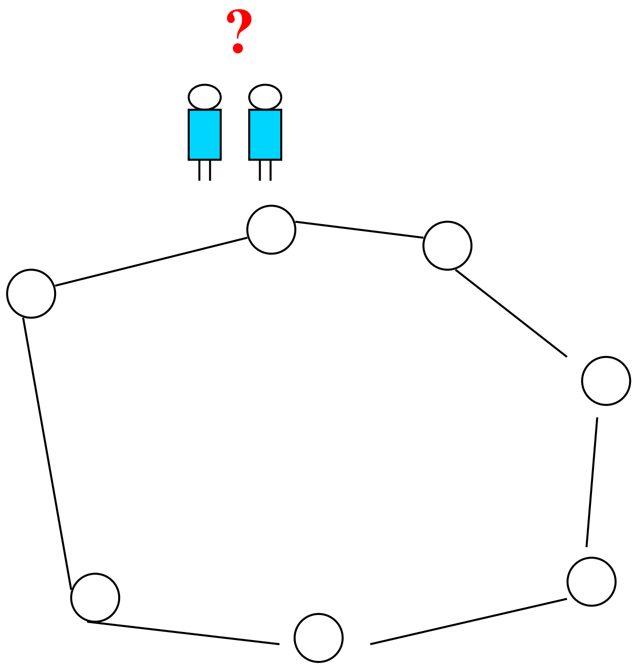
GO AROUND
until find marked node

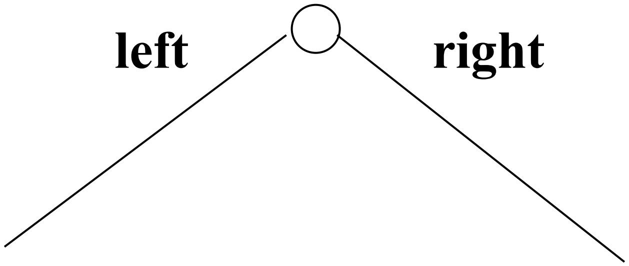
Example

More than one agent:



More than one agent:





GO AROUND in different directions until meet other

who is going left ?

Computability and Complexity

→ **Many factors to be considered**

Computability and Complexity

Asynchronous vs Synchronous

Anonymous vs Distinct Ids

Finite-State vs Turing

Oblivious vs Persistent Memory

Interaction and Communication Mechanisms

Interaction and Communication:

Vision

Face-to-Face

Tokens

Whiteboards

Interaction and Communication:

Vision

Face-to-Face

Tokens

Whiteboards

Vision

Each can see the graph and the location of the other robots within its visibility range

- **LOOK-COMPUTE-MOVE cycles**
- **Oblivious robots**
- **Anonymous nodes**

FSYNC, SSYNC, ASYNC
Limited vs Global Visibility
Labelled vs Unlabelled edges
Sense of Direction vs No Orientation

Vision

GATHERING

Klasing, Markou, Pelc, *TCS* 2008

Klasing, Kosowski, Navarra, *TCS* 2010

Kamei,Lamani,Ooshita,Tixeuil, MFCS 2012

Di Stefano,Navarra, SIROCCO 2013

Izumi,Izumi,Kamei,Ooshita, IEICE Trans. 2013

Di Stefano,Navarra, SSS 2014

D'Angelo,Di Stefano,Navarra, *Distributed Computing* 2014

D'Angelo,Di Stefano,Navarra, *J. Discrete Alg.* 2014

Kamei,Lamani,Ooshita, SRDS 2014

Di Stefano,Montanari,Navarra, IWOCA 2015

Cicerone,Di Stefano, Navarra, ALGOSENSORS 2015

D'Angelo,Di Stefano,Klasing,Navarra, *TCS* 2015

D'Angelo,Di Stefano,Navarra,Nisse,Suchan, *Algorithmica* 2015

Vision

EXPLORATION

Devismes, Petit,Tixeuil, SIROCCO 2009

Chalopin, Flocchini,Mans, Santoro. WG 2010

Lamani, Potop-Butucaru,Tixeuil, SIROCCO 2010

Flocchini, Ilcinkas, Pelc, Santoro, *TCS* 2010

Flocchini, Ilcinkas, Pelc, Santoro, *IPL* 2011

Devismes,Lamani,Petit,Raymond,Tixeuil, SSS 2012

Datta,Lamani,Larmore,Petit, ICDCS 2013

Flocchini, Ilcinkas, Pelc, Santoro, *Algorithmica* 2013

Interaction and Communication:

Vision

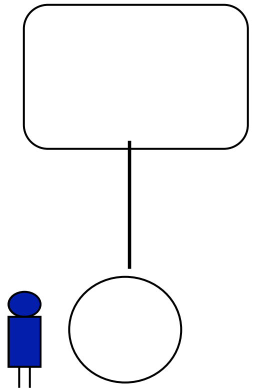
Face-to-Face

Tokens

Whiteboards

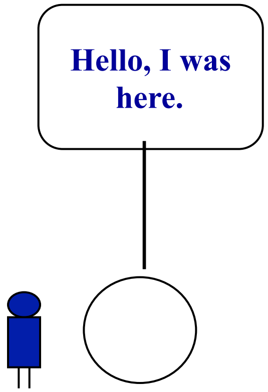
Whiteboards

Each node has a whiteboard



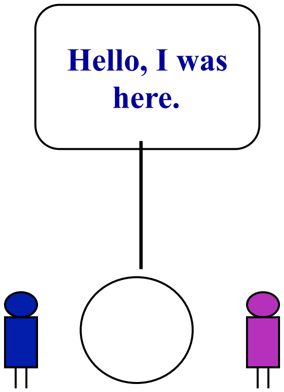
When at a node, an agent can write on the whiteboard

Whiteboards



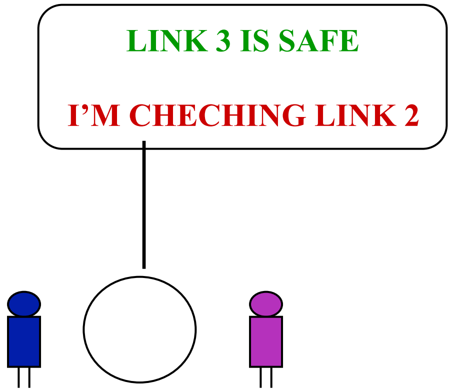
When at a node, an agent can write on the whiteboard

Whiteboards



When at a node, an agent can read what is written on the whiteboard

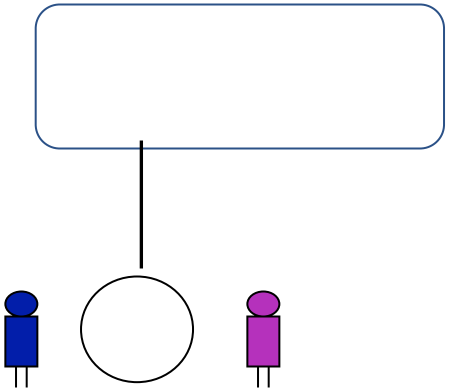
Whiteboards



Agents communicate through whiteboards, accessed in fair mutual exclusion

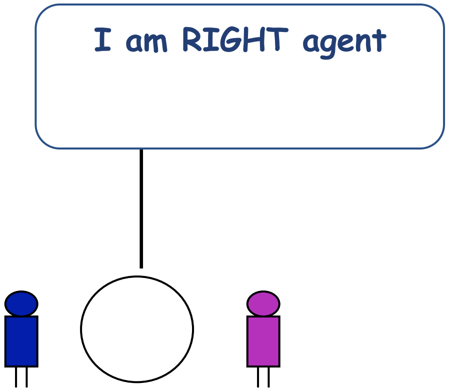
Whiteboards are powerful

- Allow to BREAK SYMMETRY



Whiteboards are powerful

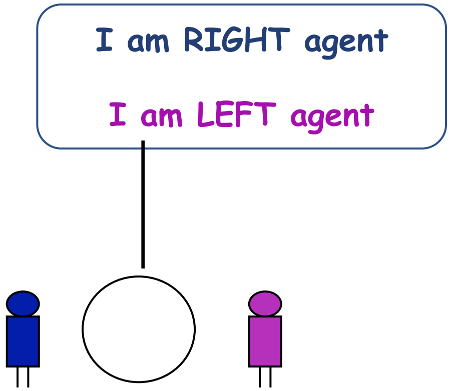
- Allow to **BREAK SYMMETRY**



I am RIGHT agent

Whiteboards are powerful

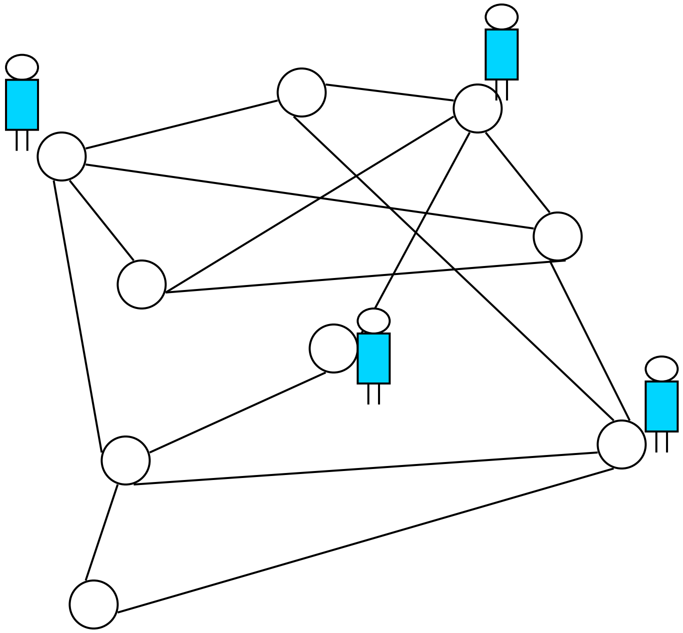
- Allow to **BREAK SYMMETRY**



I am RIGHT agent

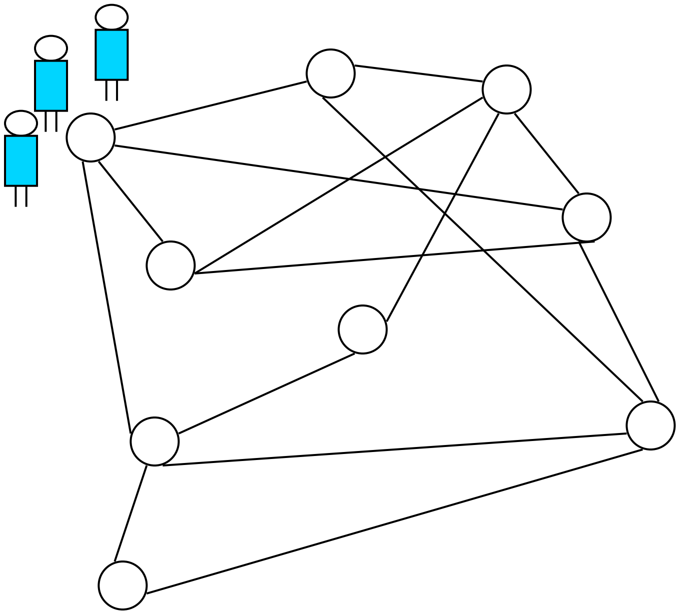
I am LEFT agent

Cost Measures : Team size



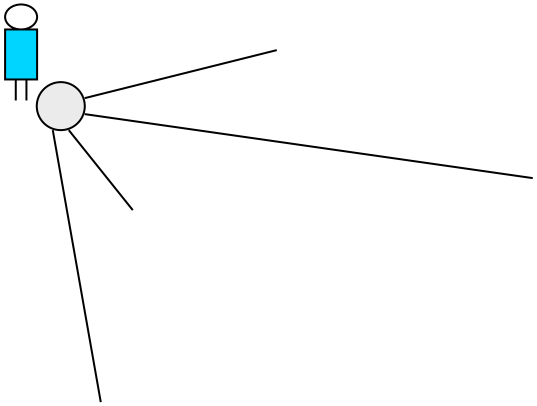
Number of agents used to solve problem / perform task

Cost Measures : Number of Moves



Number of moves made by agents to solve problem

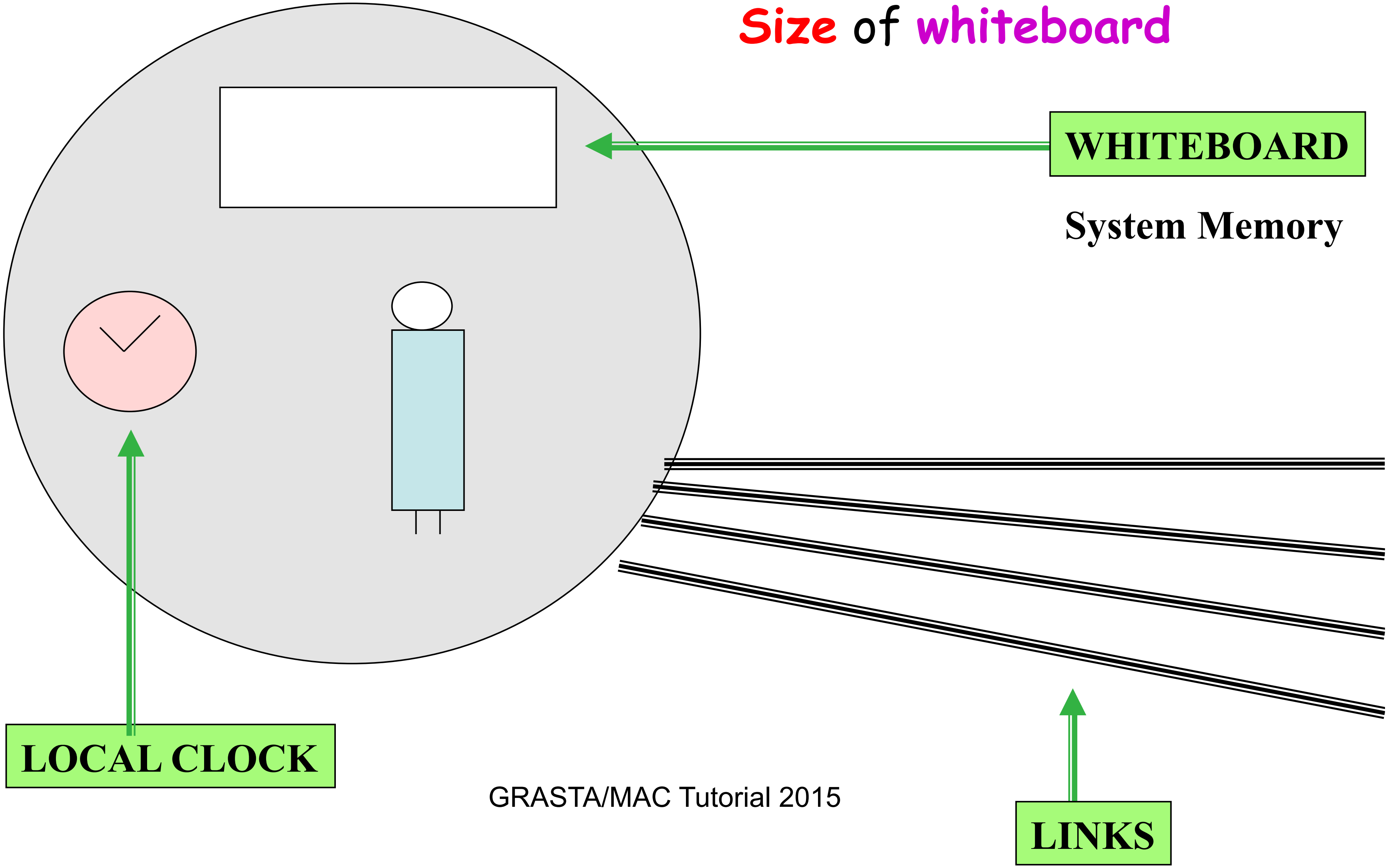
Cost Measures : Memory



Amount of **memory** a node provides to the agents

Cost Measures : Memory

Size of whiteboard



WHITEBOARD

System Memory

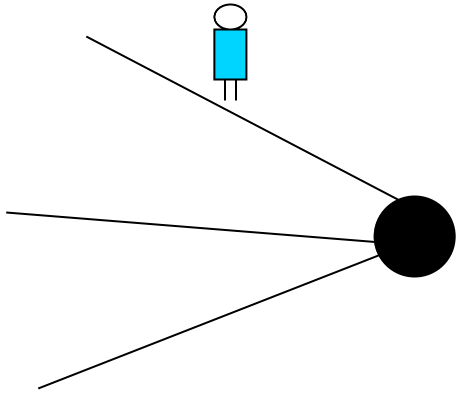
LOCAL CLOCK

LINKS

BLACK HOLE SEARCH

Black Hole Search

Black Hole



Destroys any agent arriving at that node

Does not leave any trace of destruction

Its location is unknown to the agents

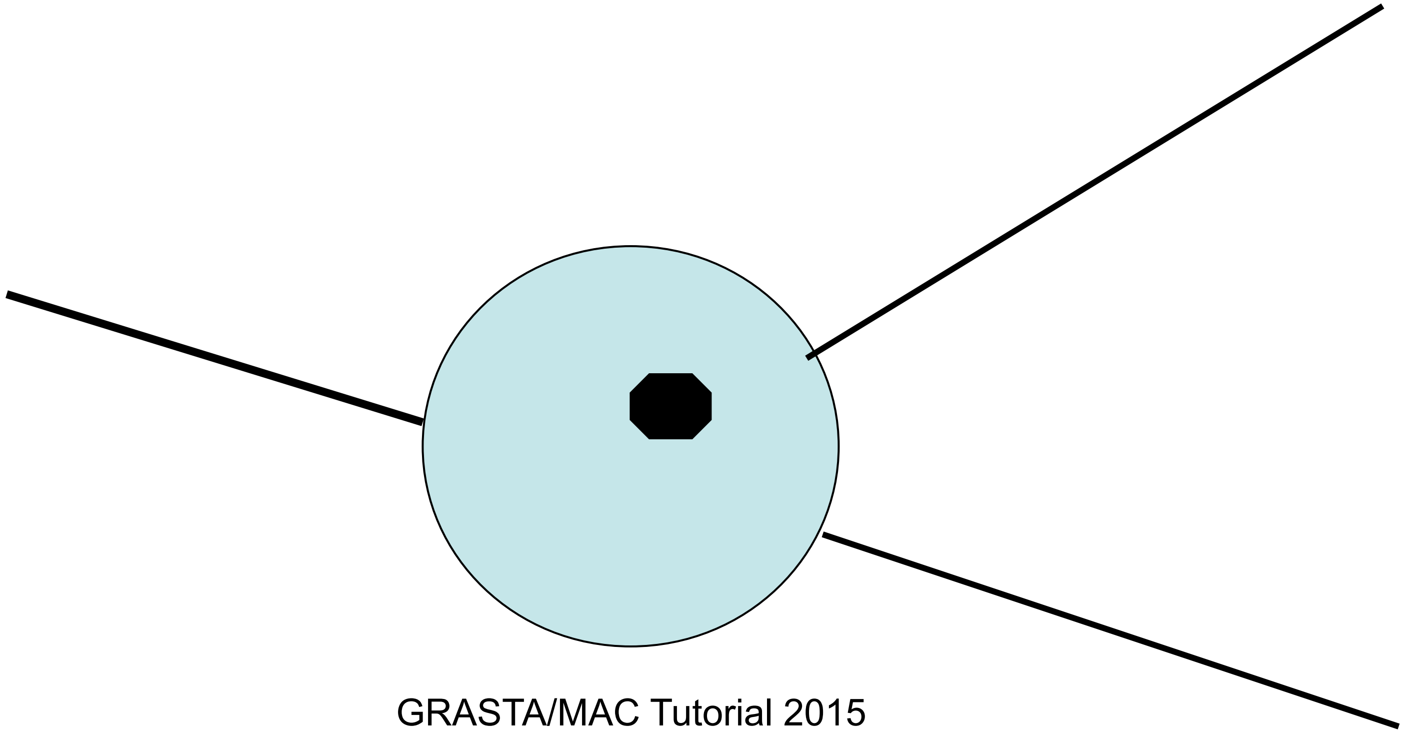
Black Hole Search

Dobrev, Flocchini, Prencipe, Santoro, *Algorithmica* 2006
Dobrev, Flocchini, Prencipe, Santoro, *Distributed Comp.* 2006
Czyzowicz, Kowalski,Markou,Pelc, *Found. Infor.* 2006
Dobrev,Flocchini,Kralovic,Prencipe,Ruzicka,Santoro, *Networks* 2006
Dobrev, Flocchini, Santoro, *TCS 20*'06
Klasing, Markou, Radzik,Sarracco, *TCS 20*'07
Czyzowicz, Kowalski, Markou, Pelc, *Comb.Prob.Comp*. 2007
Chalopin, Das, Santoro, DISC 2007
Klasing,Markou,Radzik,Sarracco, *Networks* 2008
Dobrev,Santoro,Shi, *IJFCS* 2008
Balamohan, Flocchini, Miri, Santoro, *Discrete Math.* 2011
Kosowski,Navarra,Pinotti, *TCS* 2011
Flocchini, Ilcinkas, Santoro, *Algorithmica* 2012
Chalopin, Das, Labourel, Markou, *TCS* 2013
Balamohan, Dobrev, Flocchini, Santoro, *TCS* 2014
Shi,Garcia-Alfaro. Corriveau, *J. Par. Dist. Comp*. 2014
D'Emidio, Frigioni, Navarra, *TCS* 2015

BH

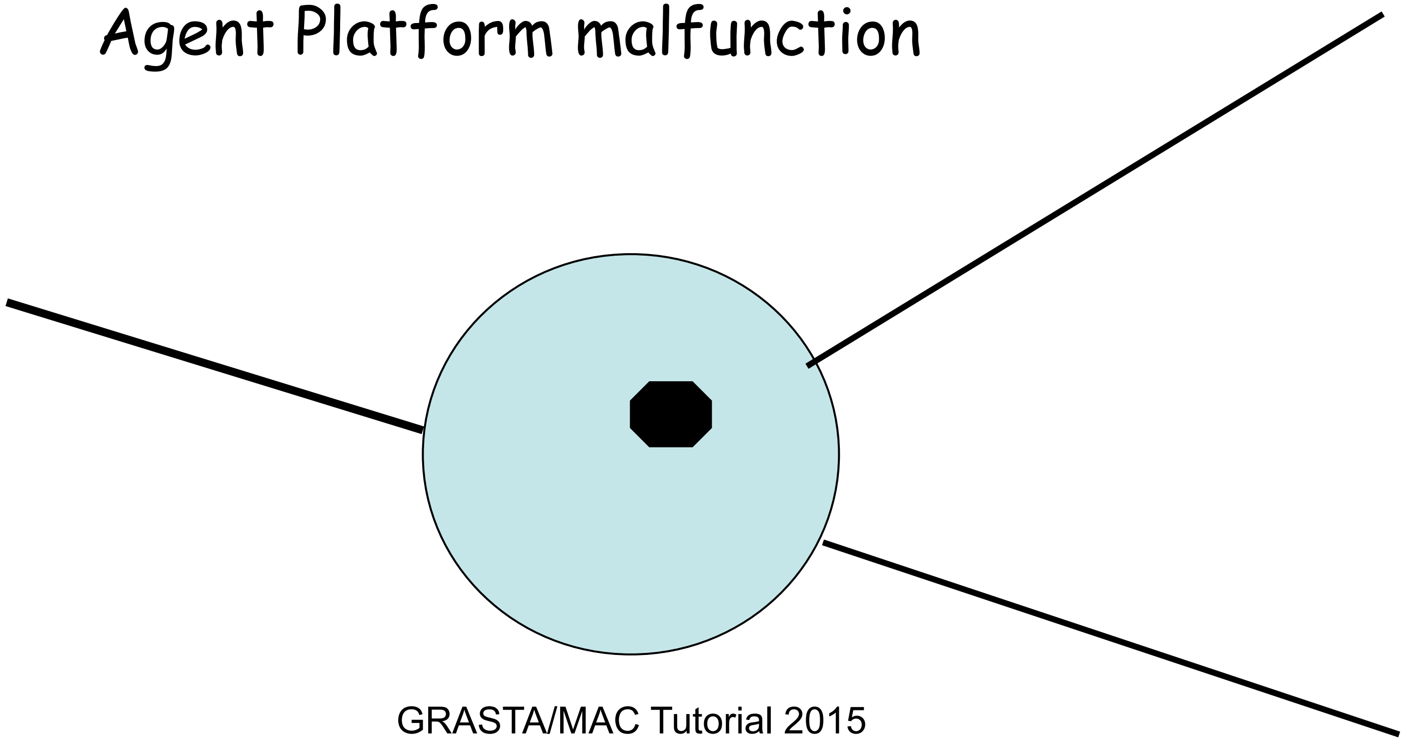
is not uncommon !!

Local Process / Software Failure



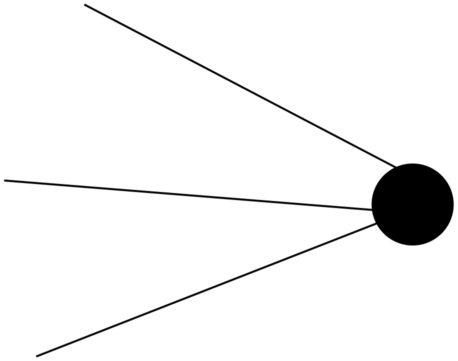
Local Process / Software Failure

- Virus
- Software malfunction
    - Port management malfunction
    - Security malfunction
    - Agent Platform malfunction



Hardware Failure

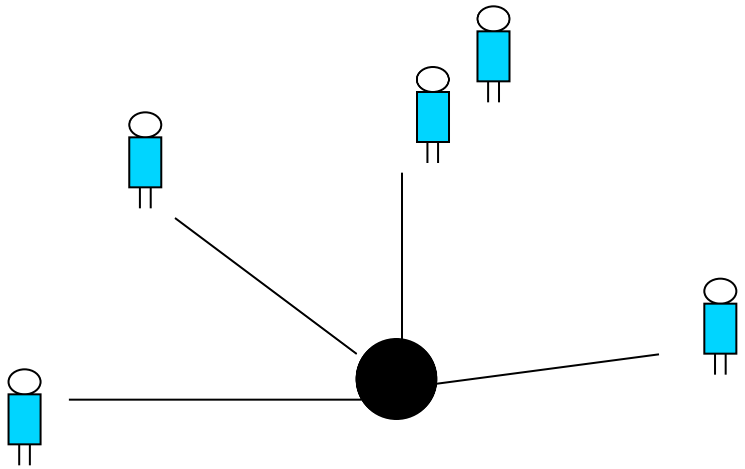
Crash failure of a site in a asynchronous network = **Black Hole**



Black Hole Search

Find the location of the black hole.

At least one agents must survive and know the location of the black hole

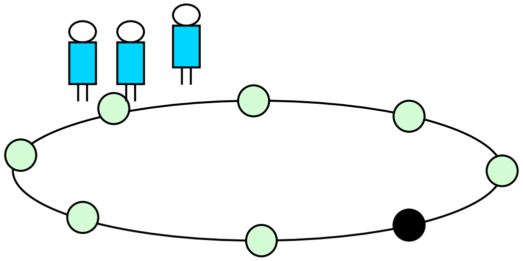


Timing

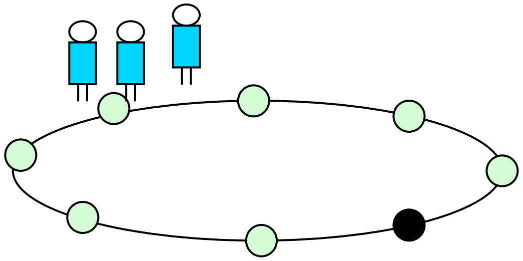
Asynchronous

Agent's actions (computations, movements) take a finite but otherwise unpredictable amount of time

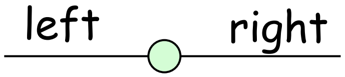
AN EXAMPLE : Black Hole Search in a Ring



AN EXAMPLE : Black Hole Search in a Ring

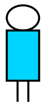


n nodes
k agents
one black hole

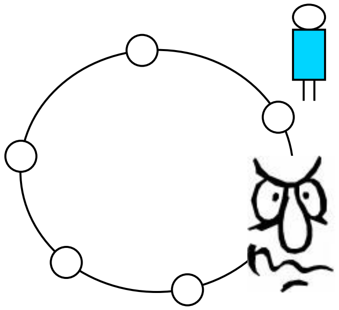


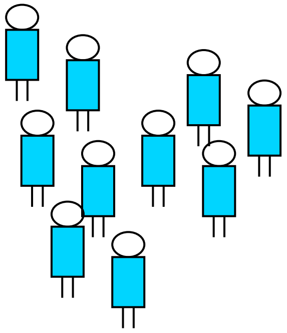
n known Min k ? How ?

Ring: Basic Facts

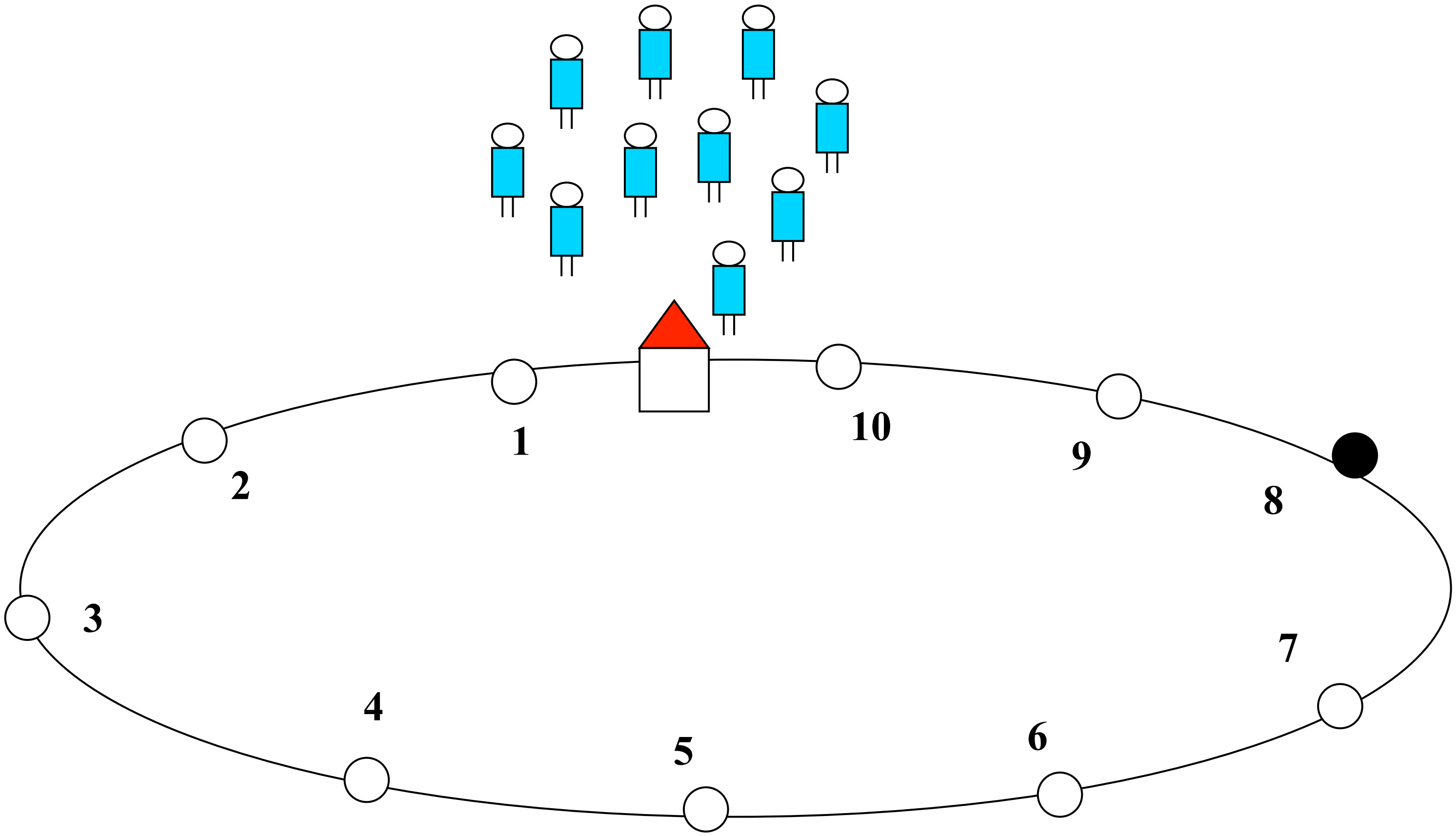


One agent cannot locate the black hole alone

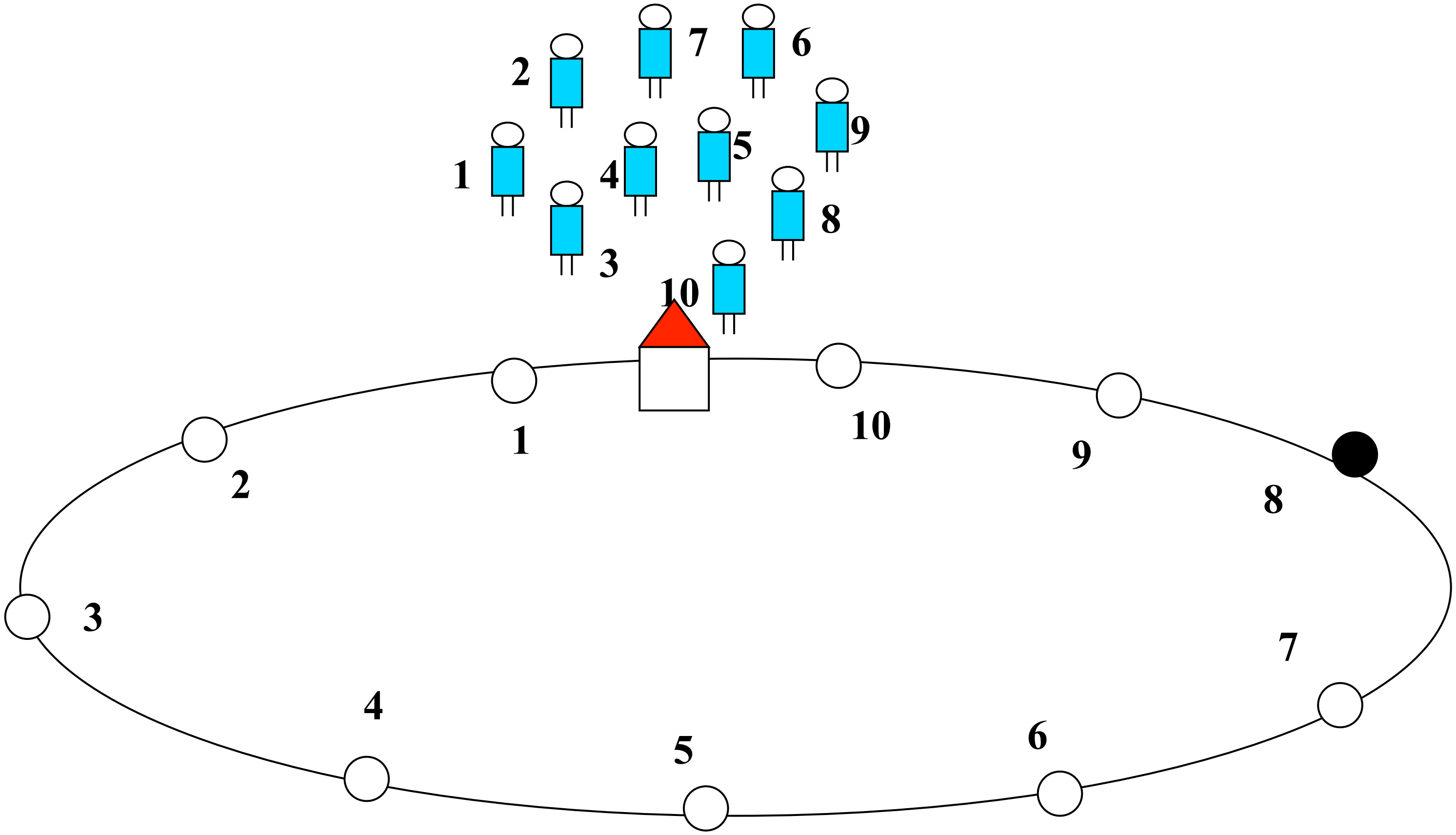




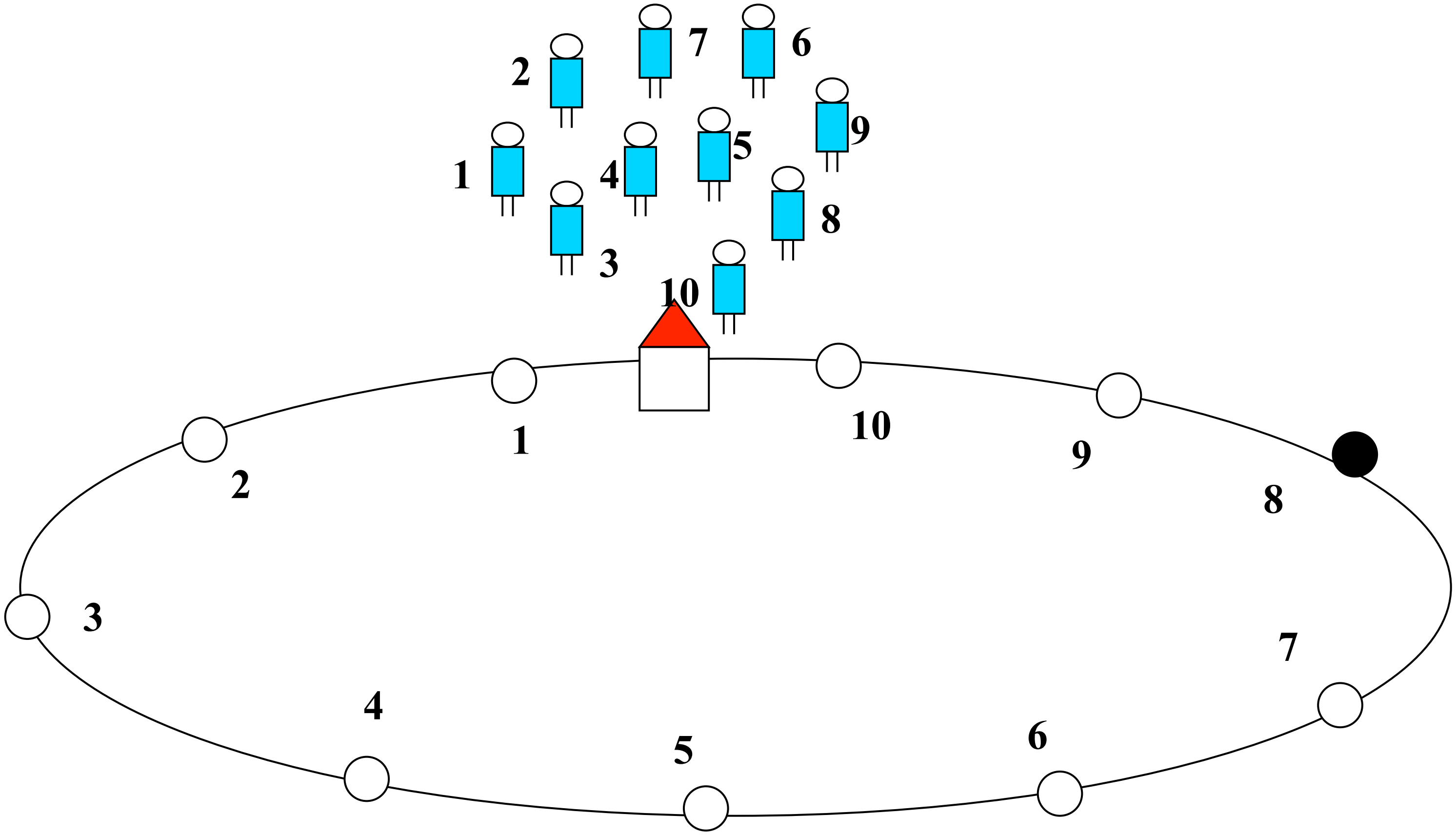
n-1 agents can

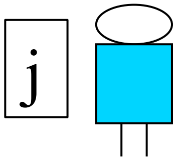


how ?

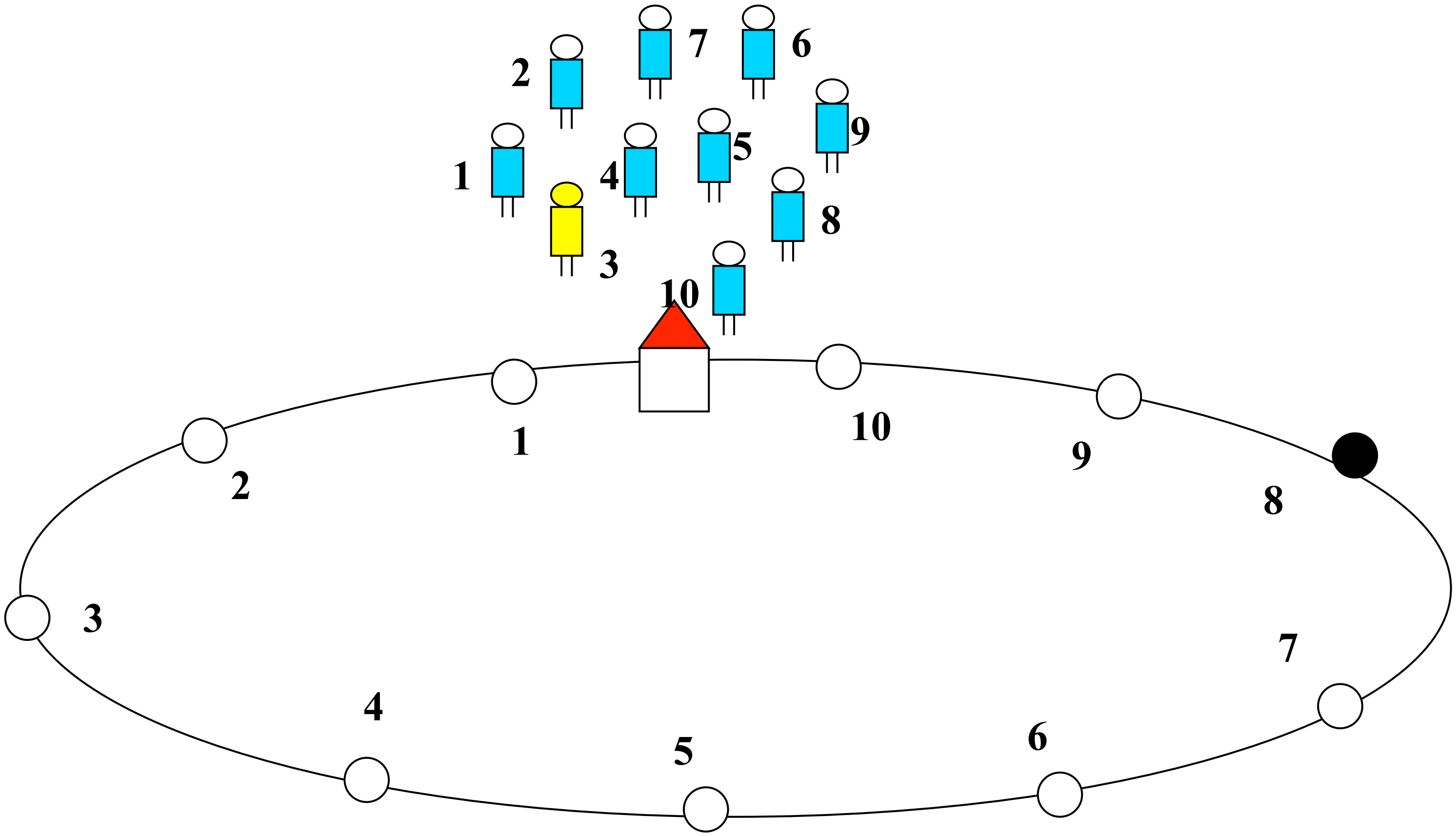


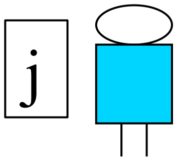
how ?



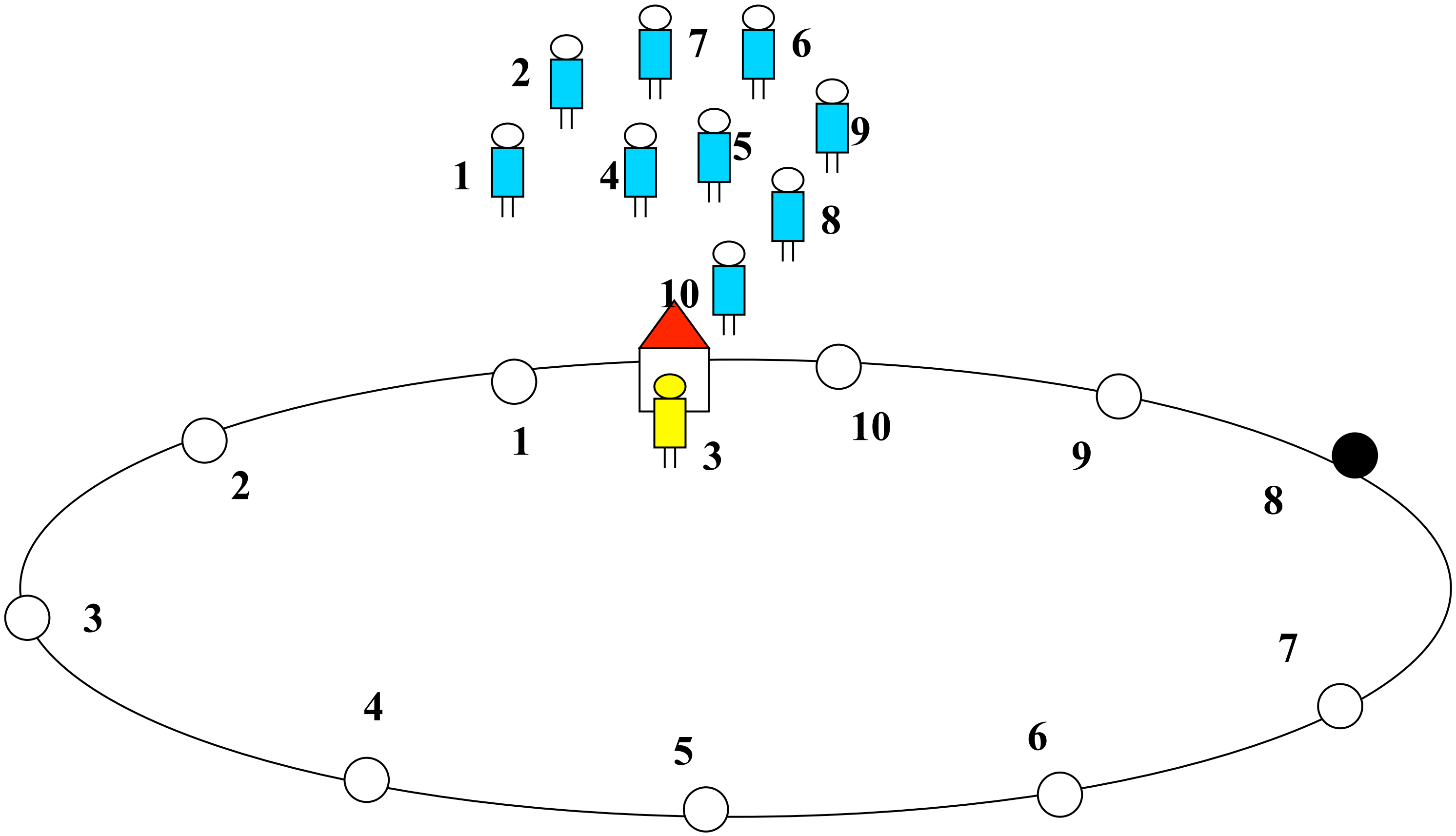


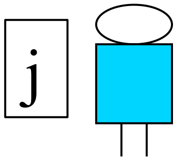
Check if node j is BH



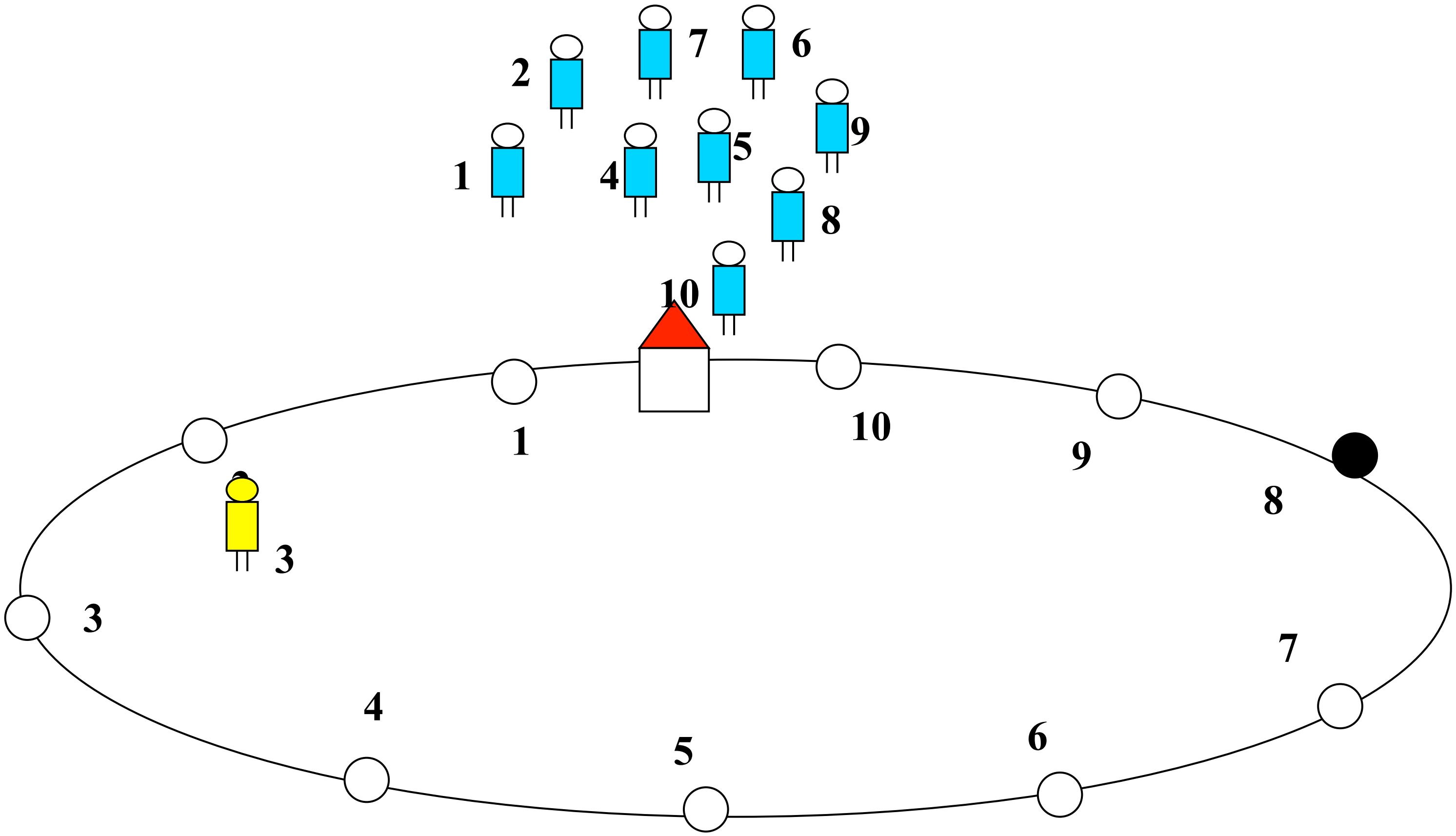


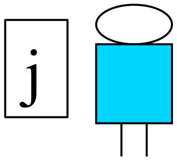
Check if node j is BH :
go to j-1, return; **go to j+1**, return.





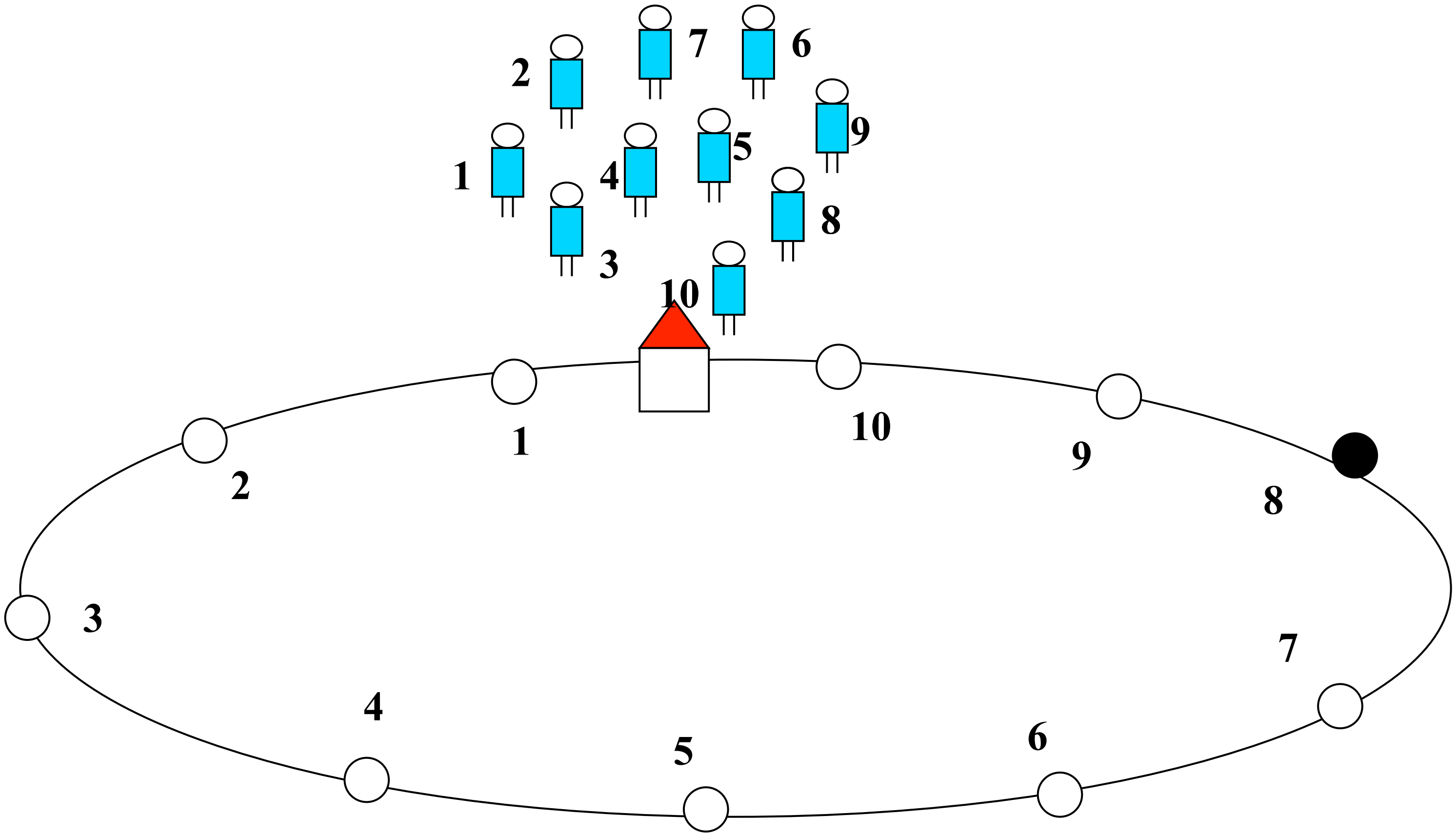
Check if node j is BH :
go to j-1, return; **go to j+1**, return.

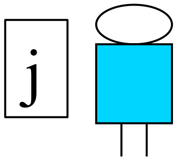




Check if node j is BH :
go to j-1, return; **go to j+1**, return.

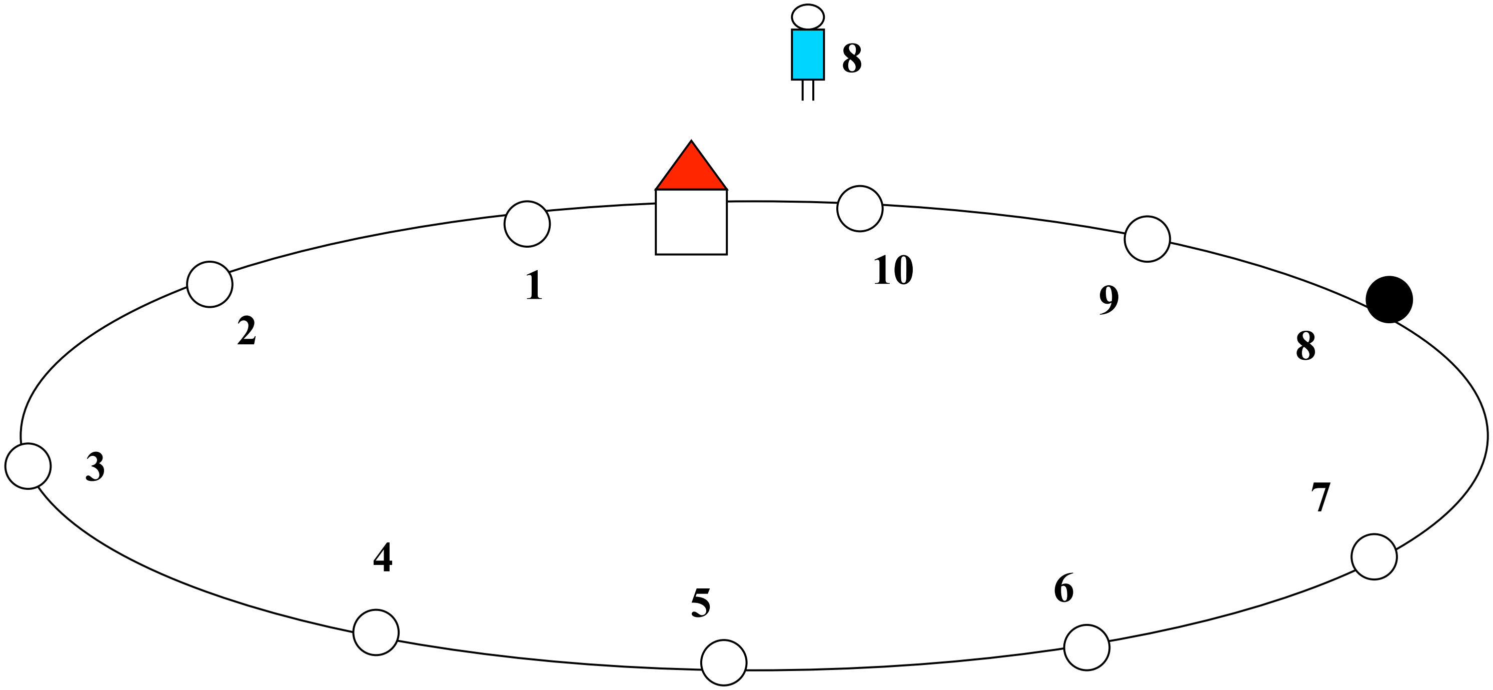
GRASTA/MAC Tutorial 2015

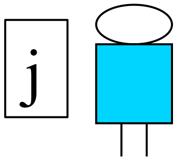




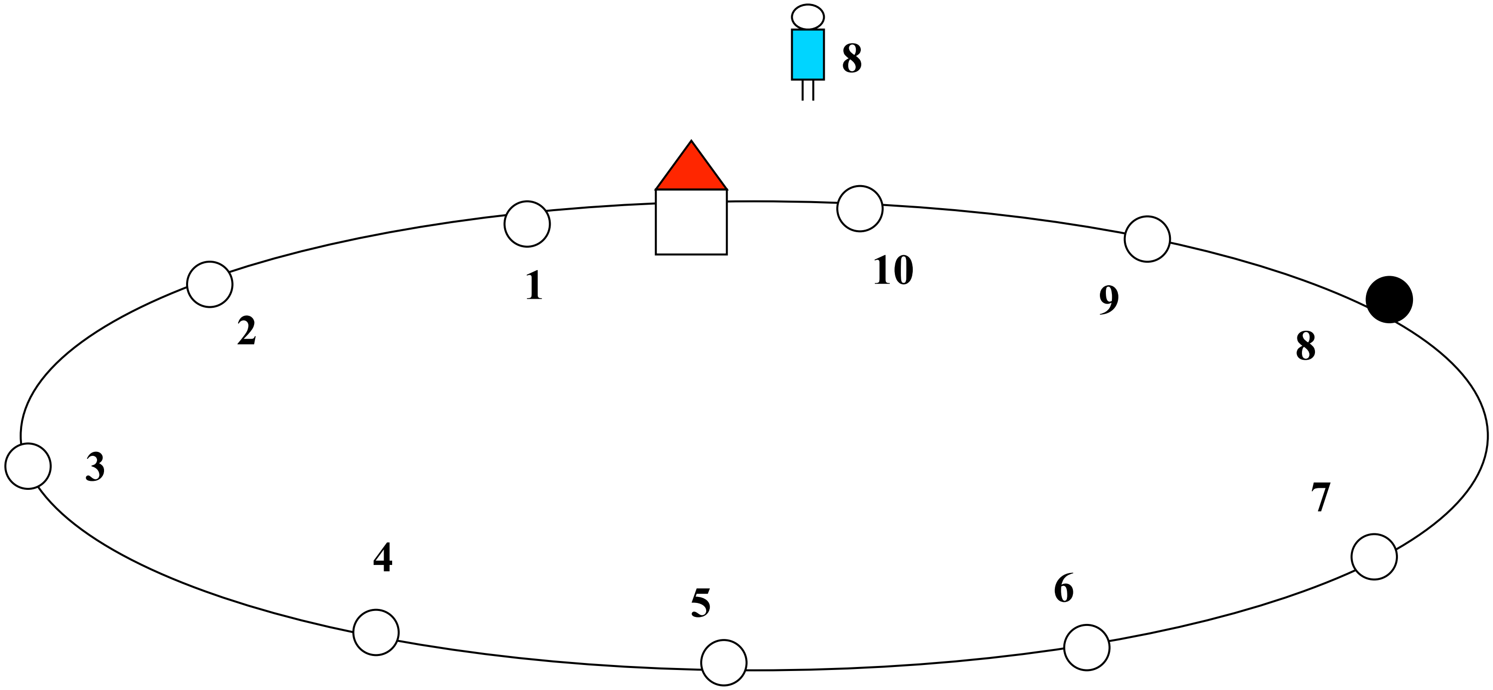
Check if node j is BH :

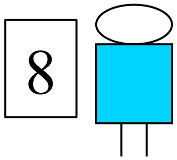
go to j-1, return; go to j+1, return.





Check if node j is BH :
go to j-1, return; **go to j+1**, return.





only one agent survives and knows BH

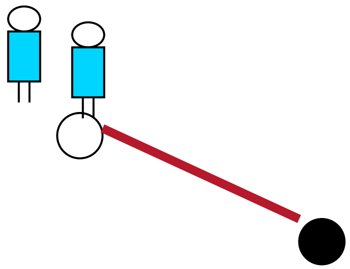
n-2 agents die

O(n^2) moves

Minimize Casualties

Minimize Casualties

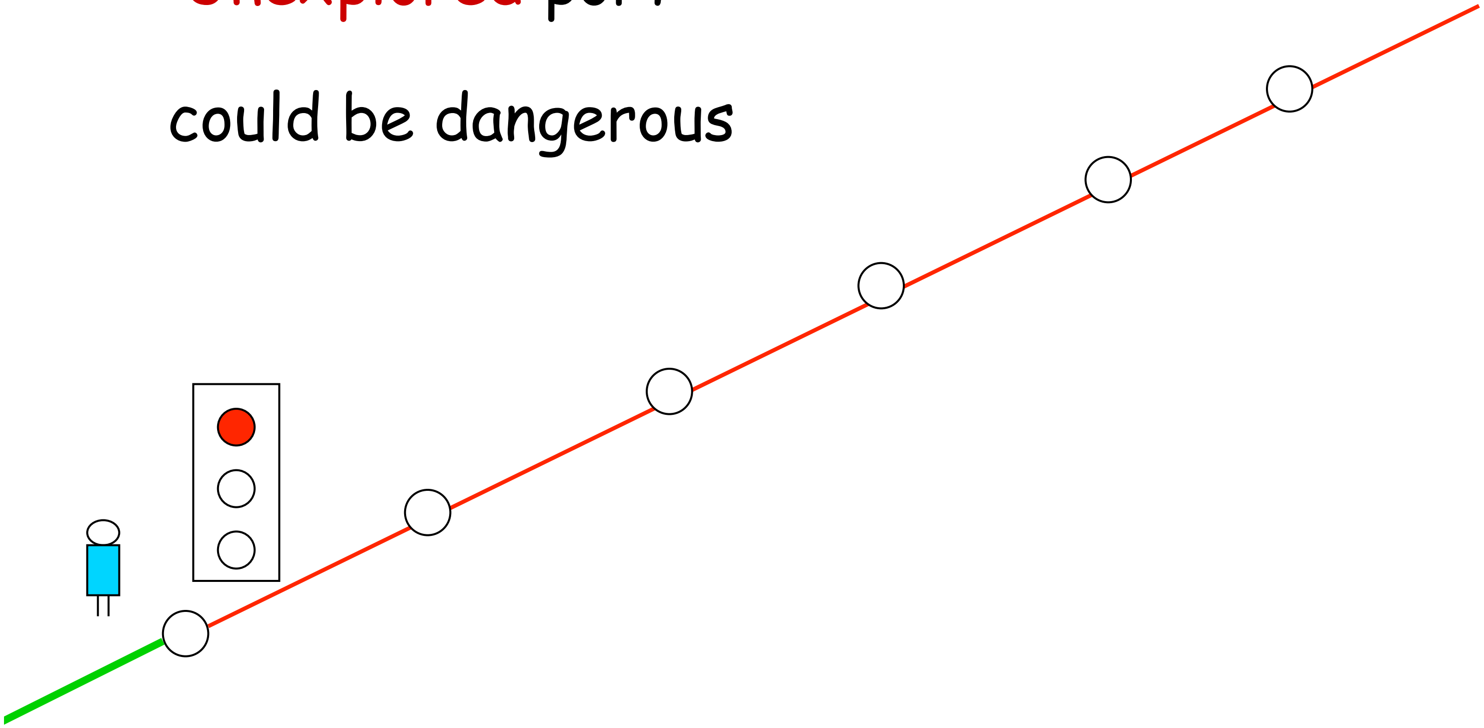
At most one agent disappears on the same link !



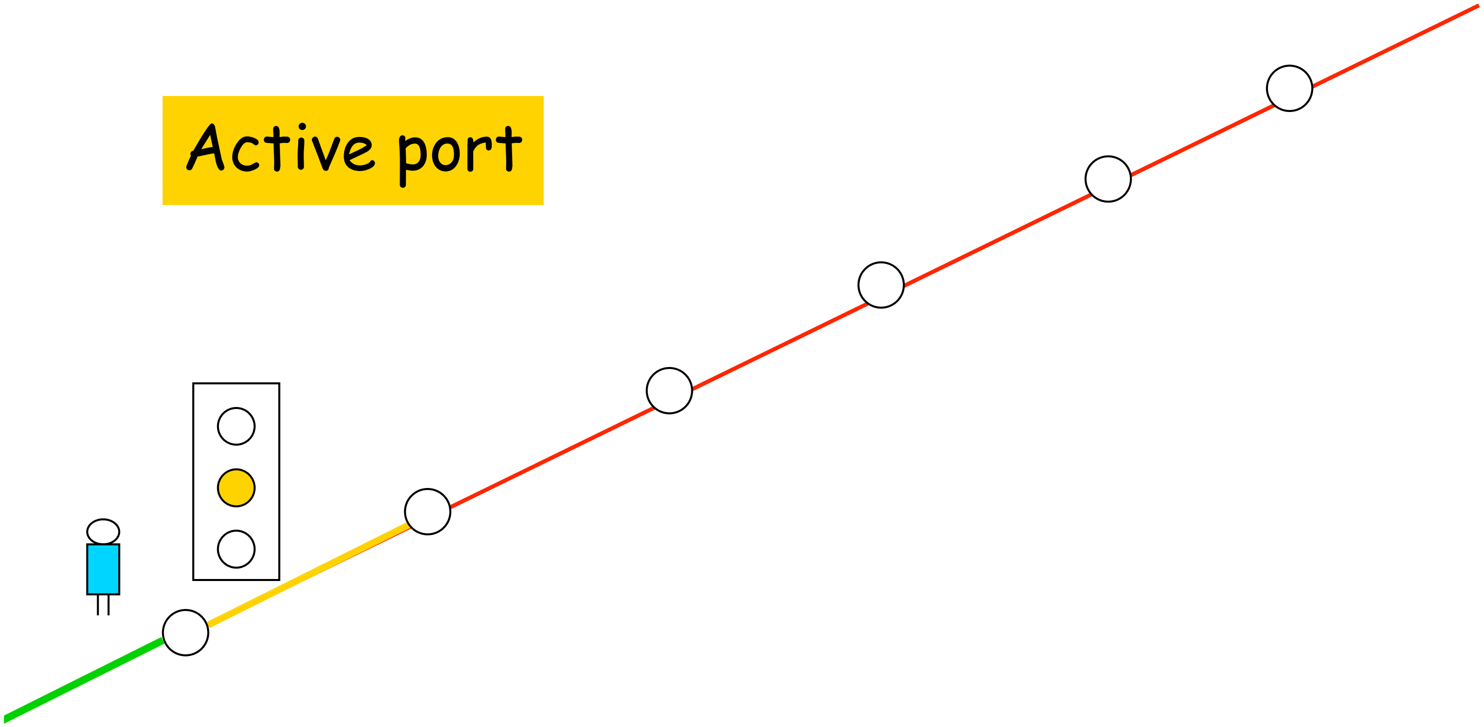
Cautious Walk

Unexplored port

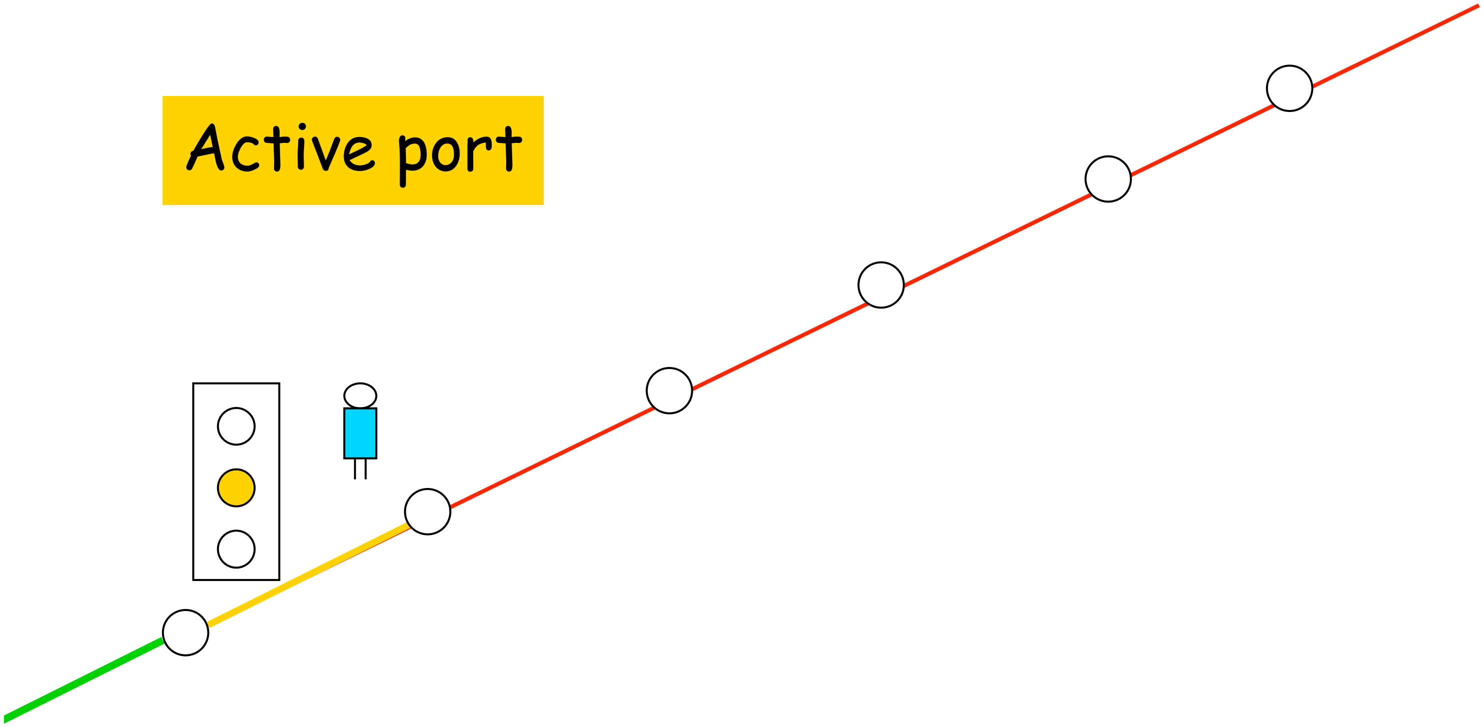
could be dangerous



Cautious Walk



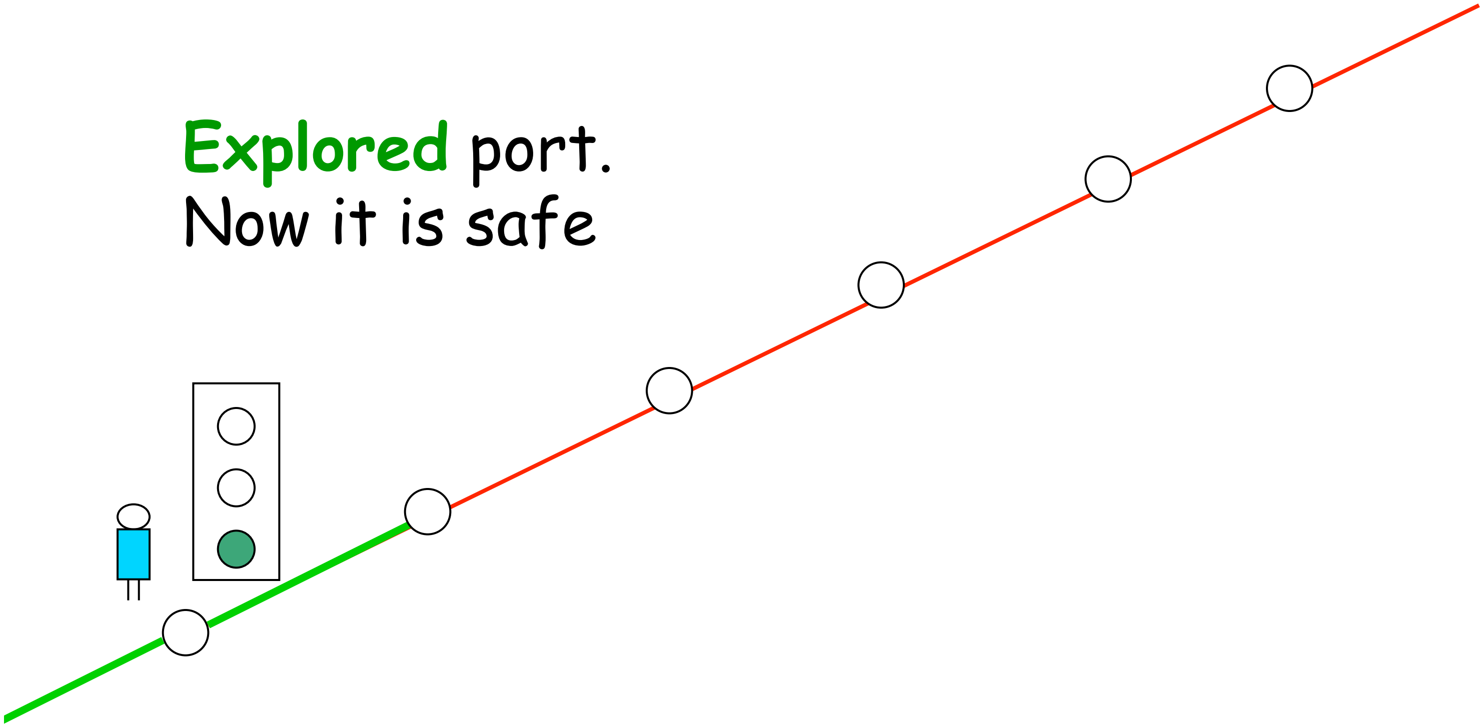
Cautious Walk



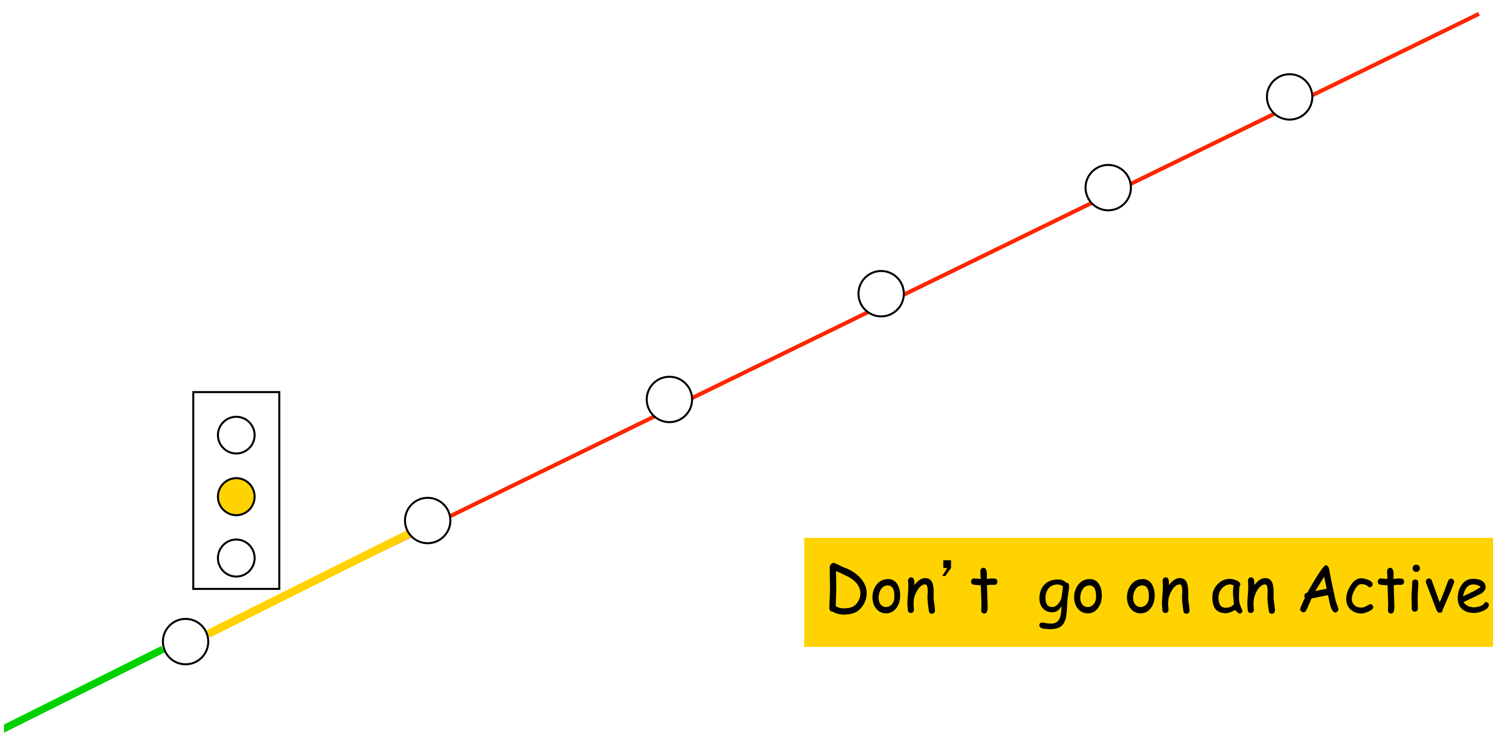
Active port

Cautious Walk

Explored port.
Now it is safe

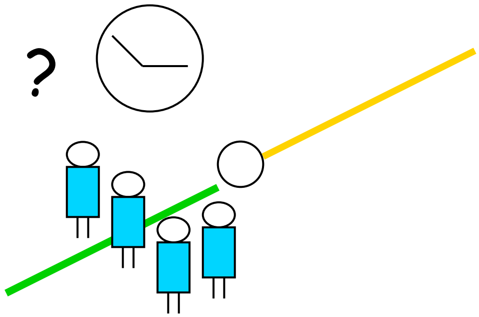


Cautious Walk : RULES

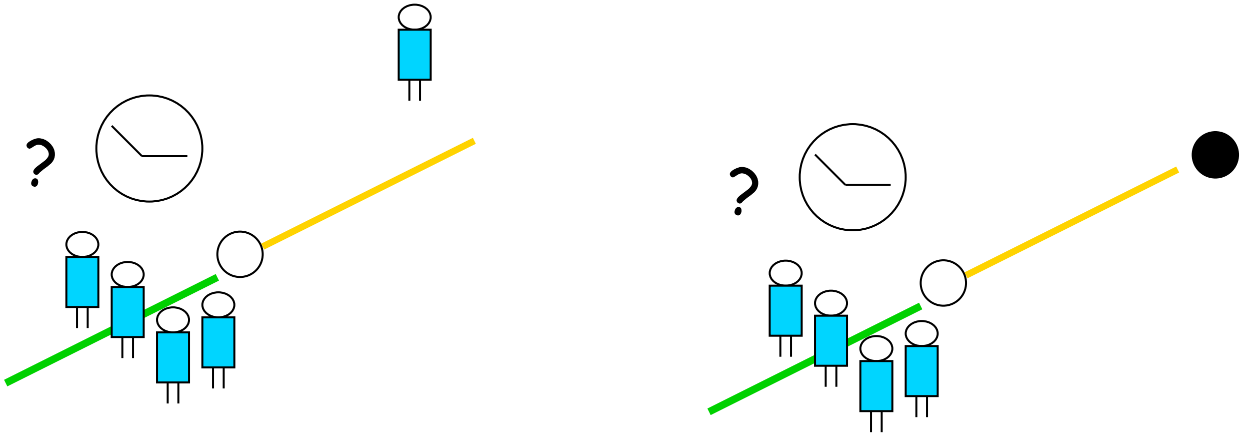


Don’t go on an Active port

ASYNCHRONY makes the problem DIFFICULT



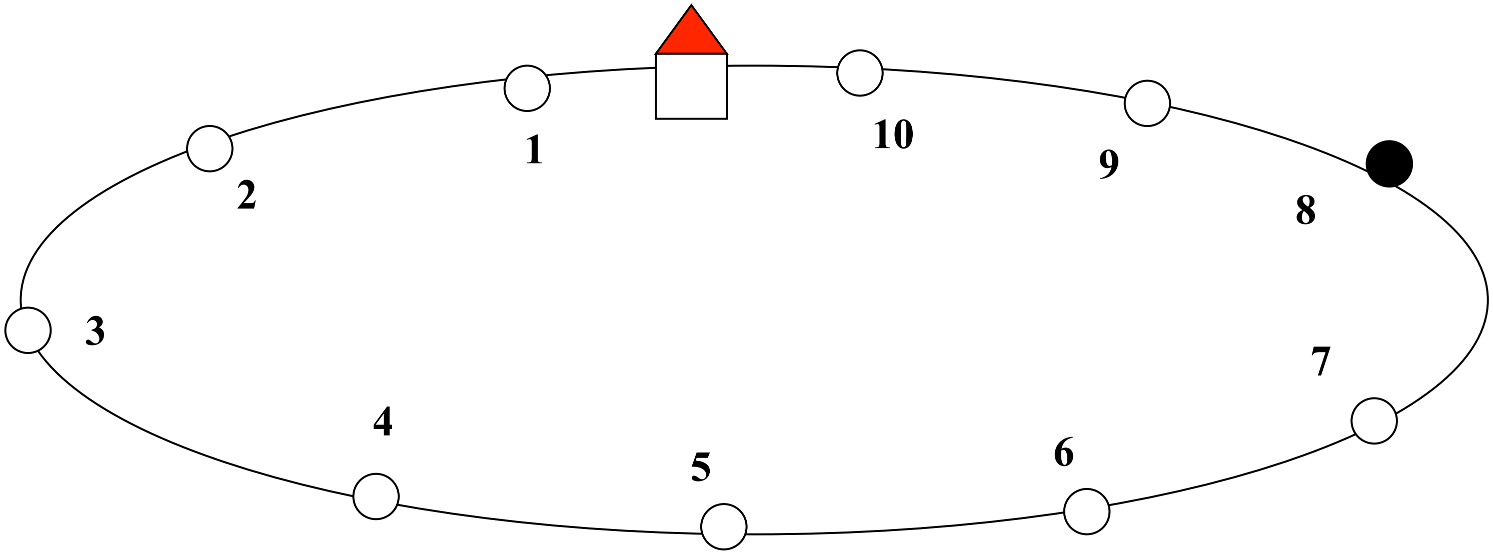
ASYNCHRONY makes the problem DIFFICULT



I cannot use timeouts

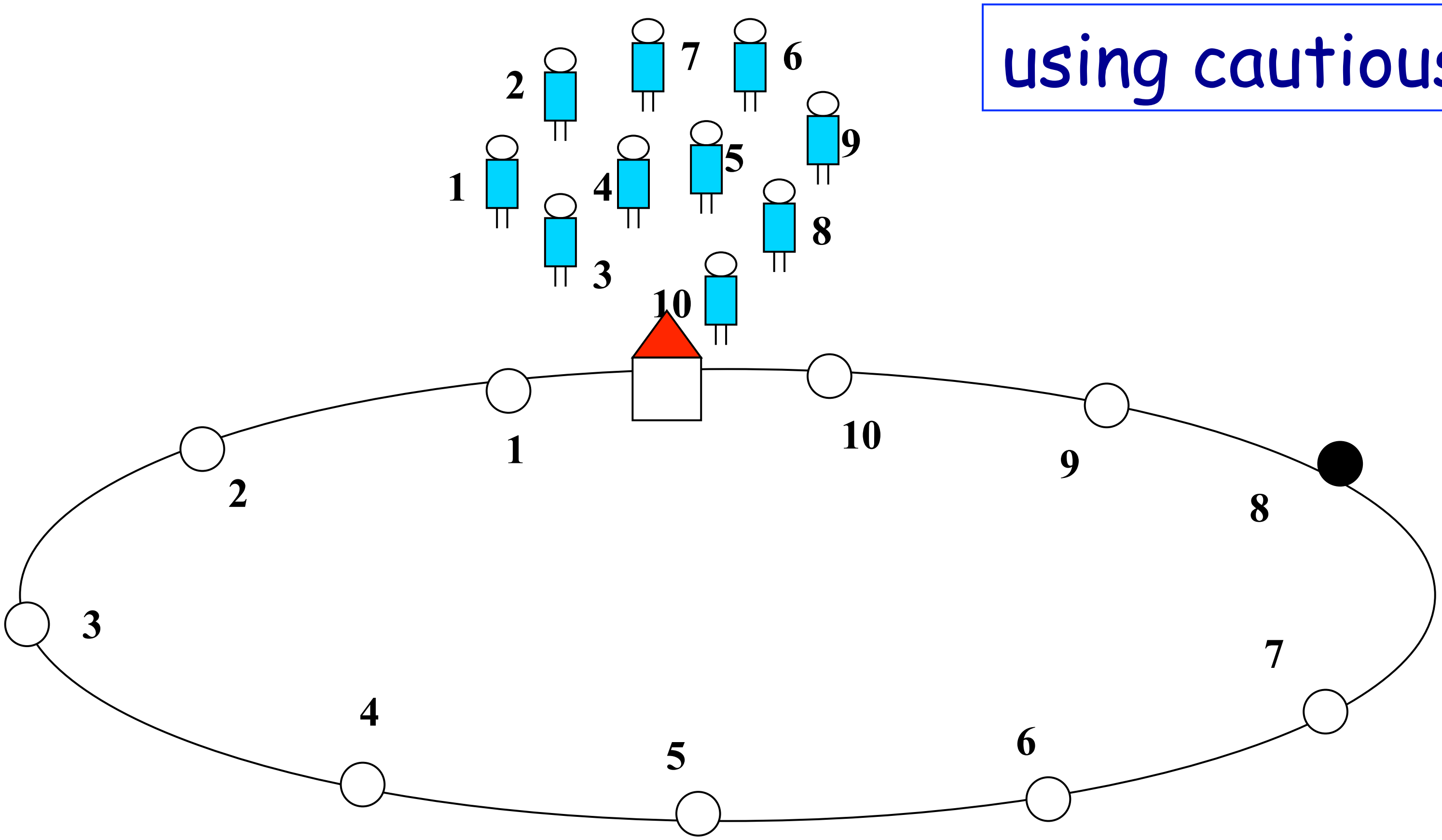
I might wait forever

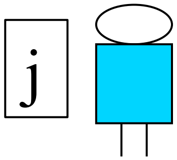
Black Hole Search in a Ring



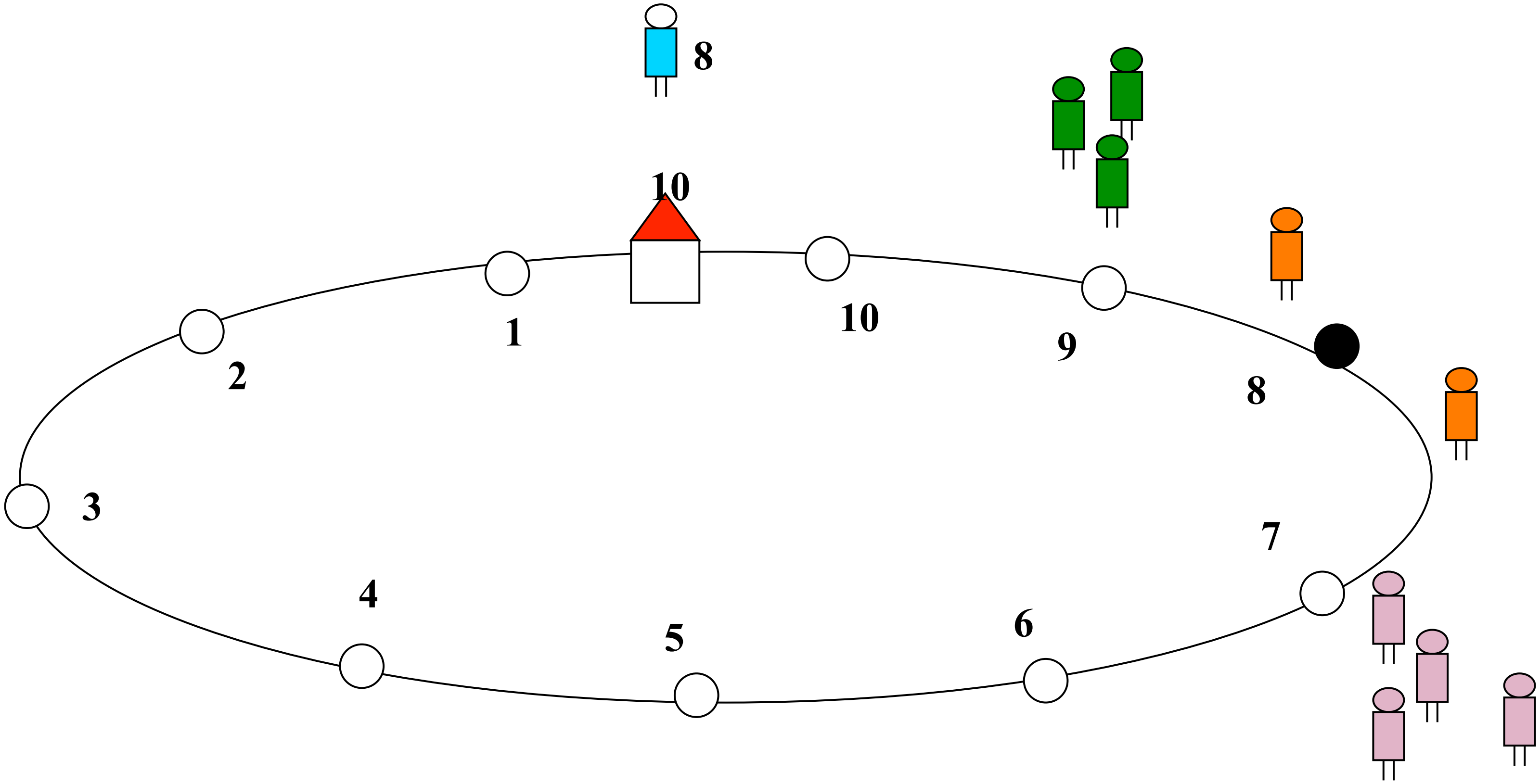
using cautious walk

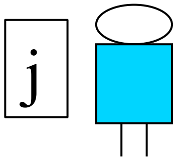
using cautious walk



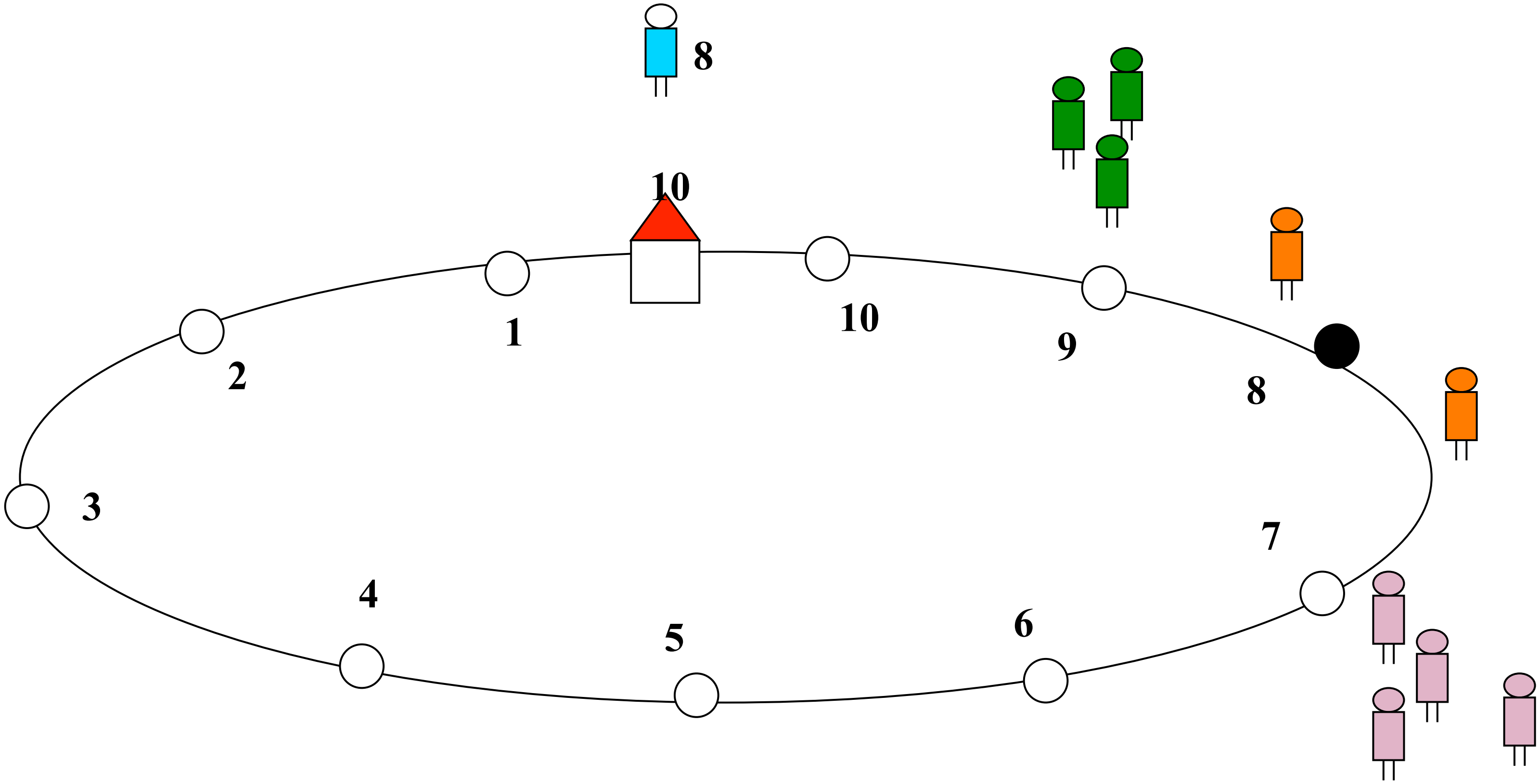


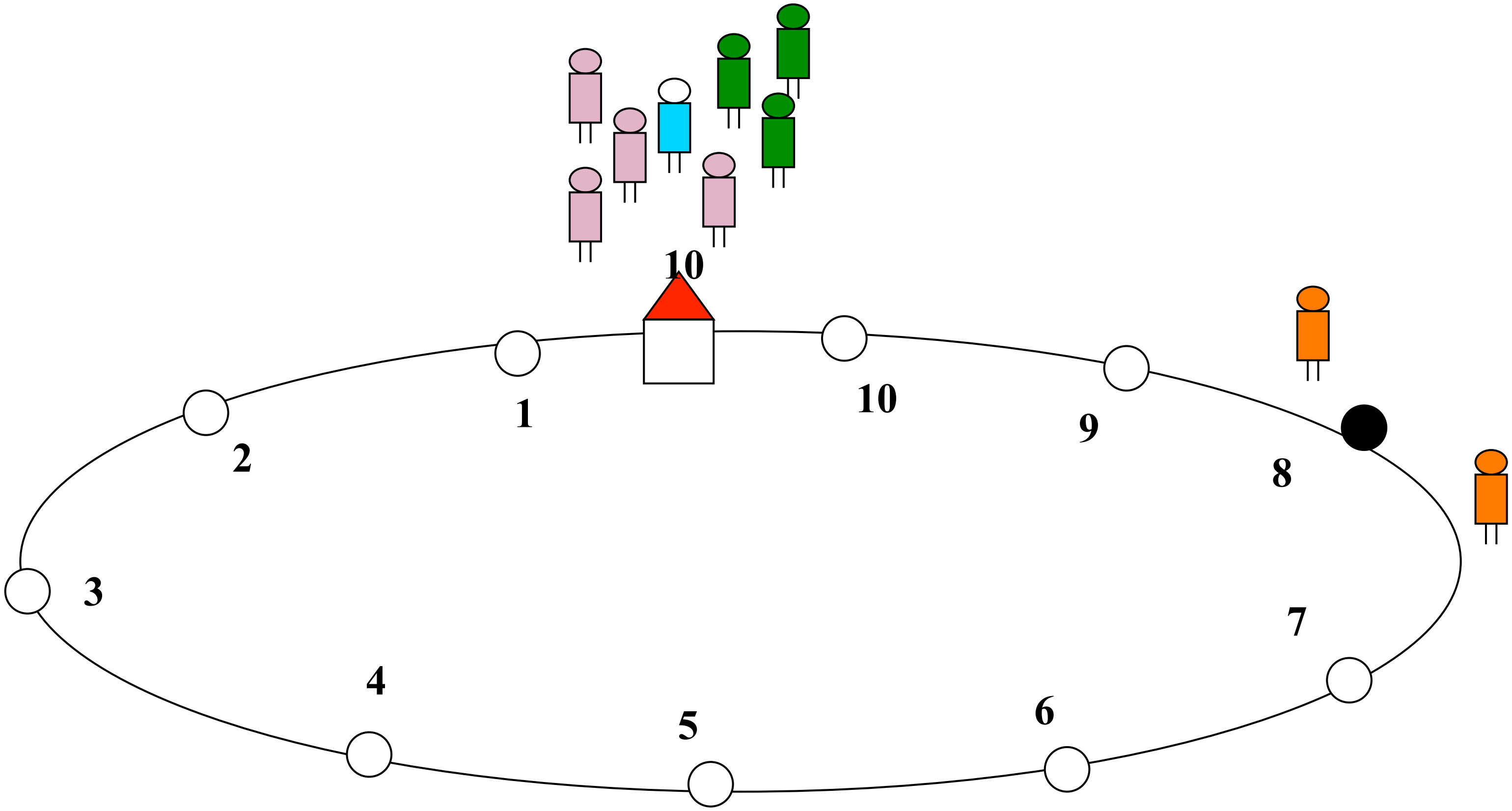
Check if node j is BH :
go to j-1, return; go to j+1, return.





Check if node j is BH :
go to j-1, return; go to j+1, return.


8
10
1
2
3
4
5
6
7
8
9
10



with n-1 agents using cautious walk

at most 2 agents die

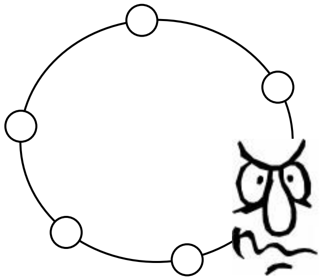
$O(n^2)$ moves

Minimize number of agents

What is the smallest number of agents ?

Black Hole Search in a Ring

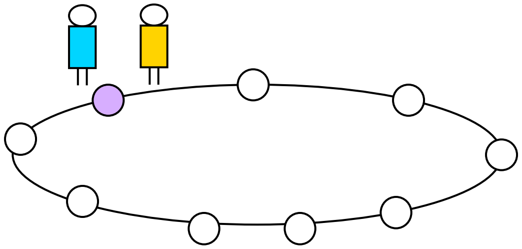
One agent cannot locate the black hole alone



How about **two** agents ? (one dies, one survives) **YES !**

Less than $O(n^2)$ moves ? **YES !**

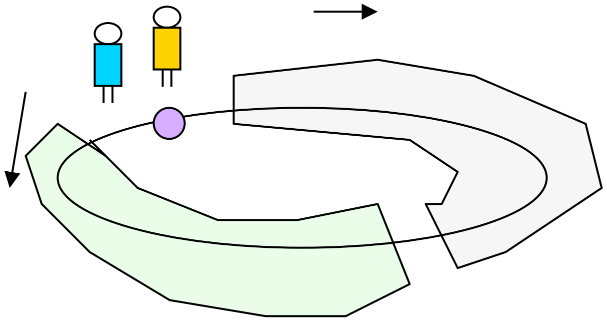
Ring: Two Agents



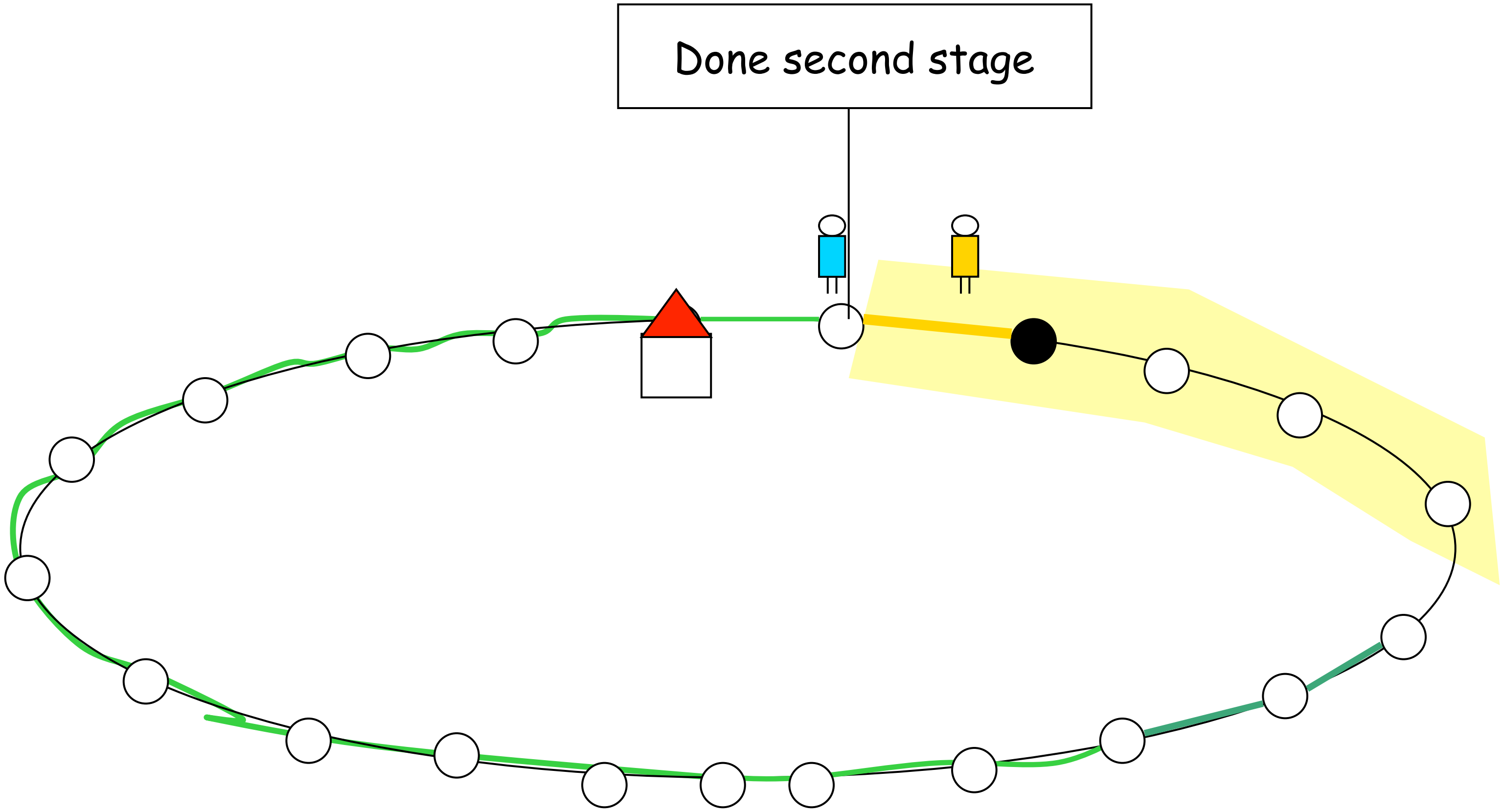
Proceed in phases.

At phase i

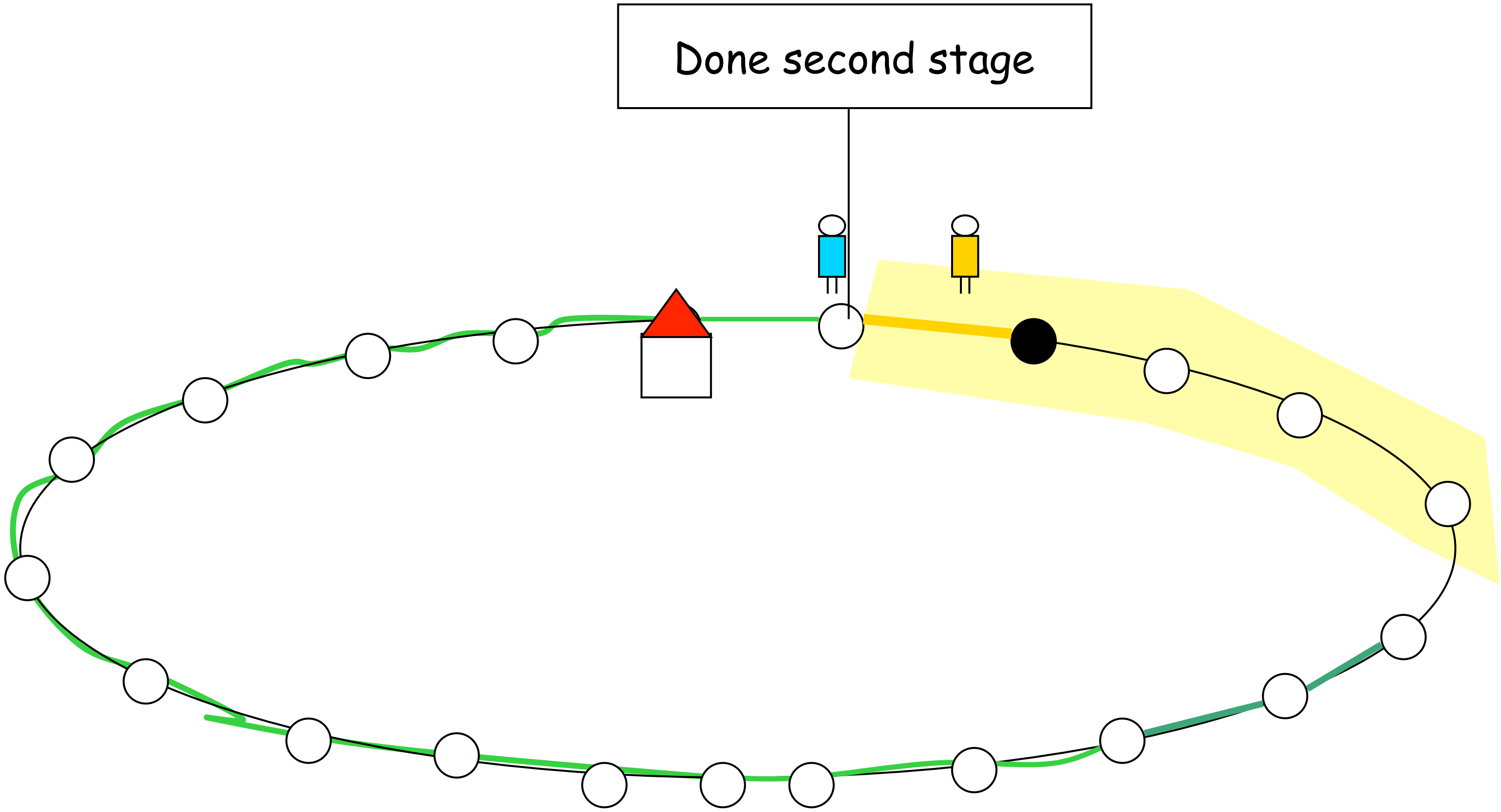
- divide the unexplored area in two contiguous disjoint parts of almost equal size



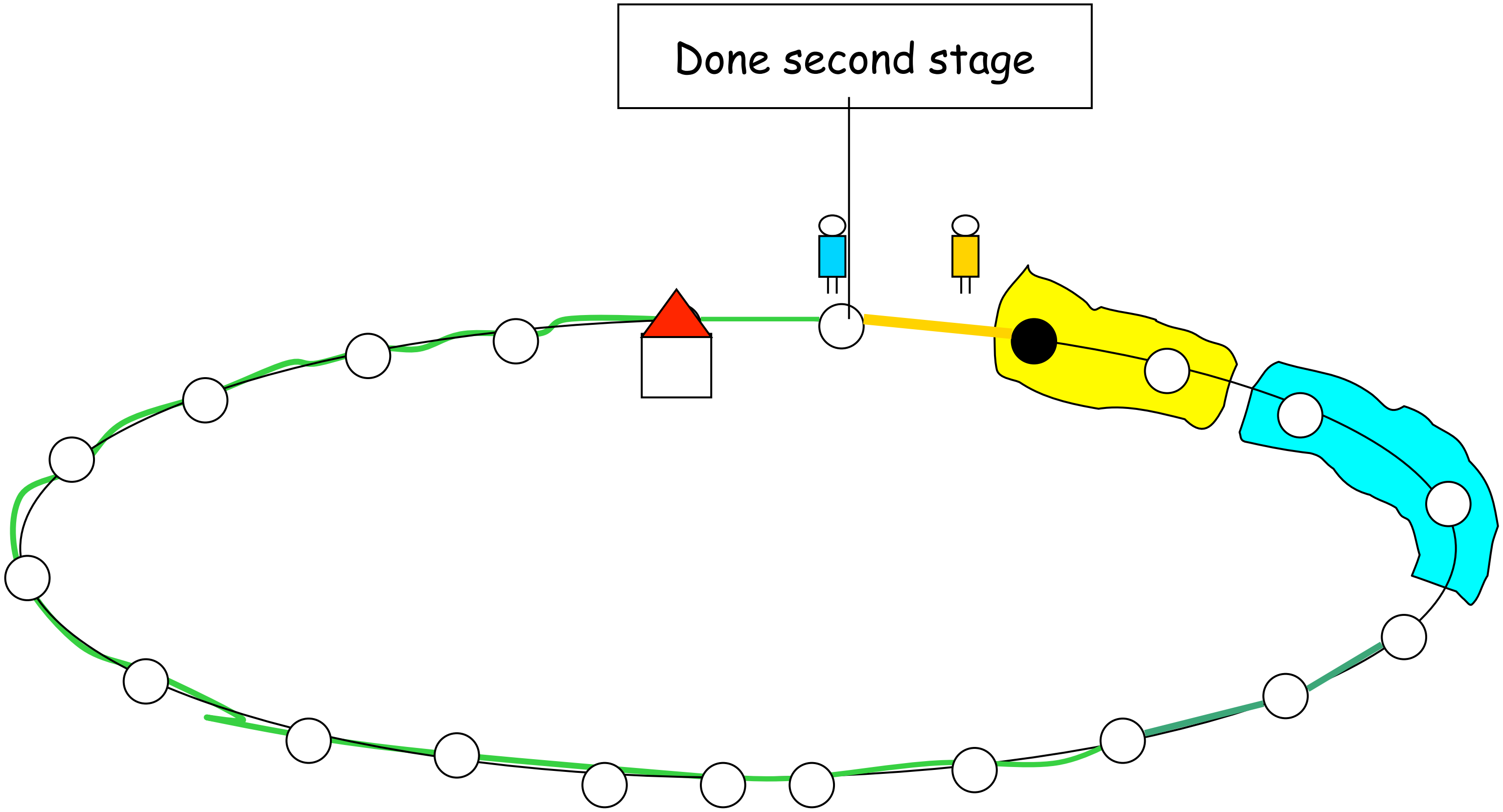
- agents explore (using cautious walk) the different parts



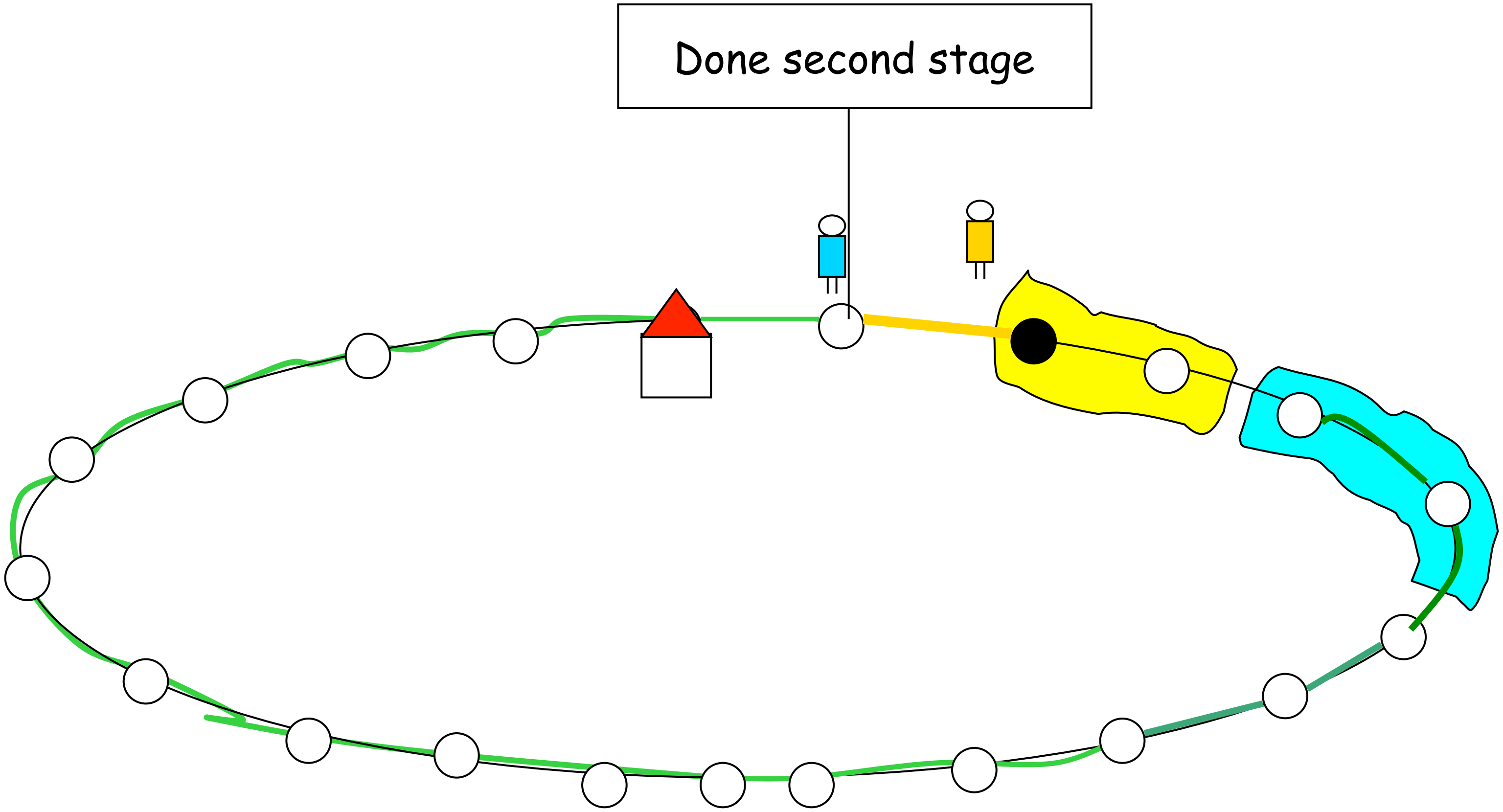
Leave a message for the other agent



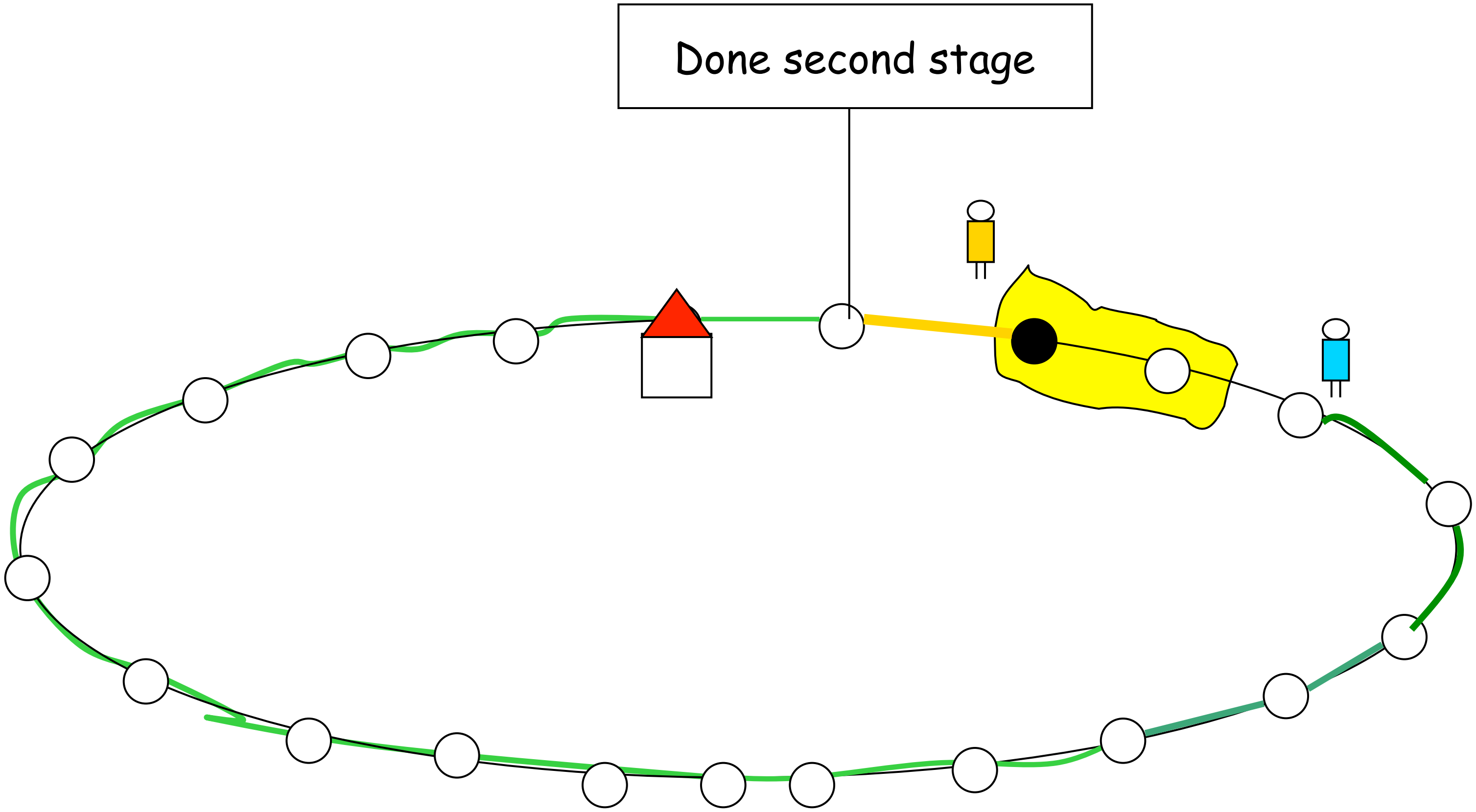
Partition unexplored region
into two areas of almost equal size



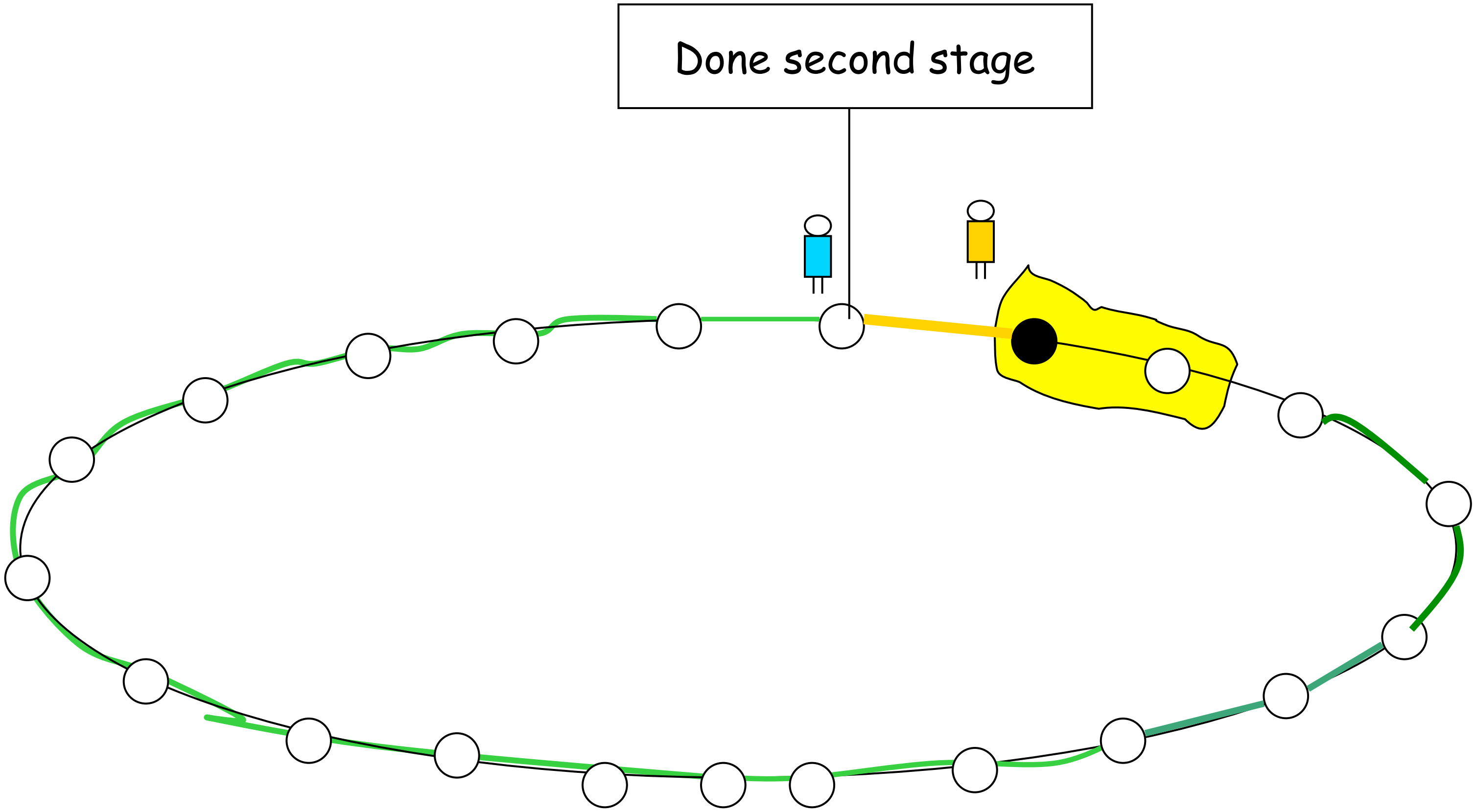
Partition unexplored region
into two areas of almost equal size



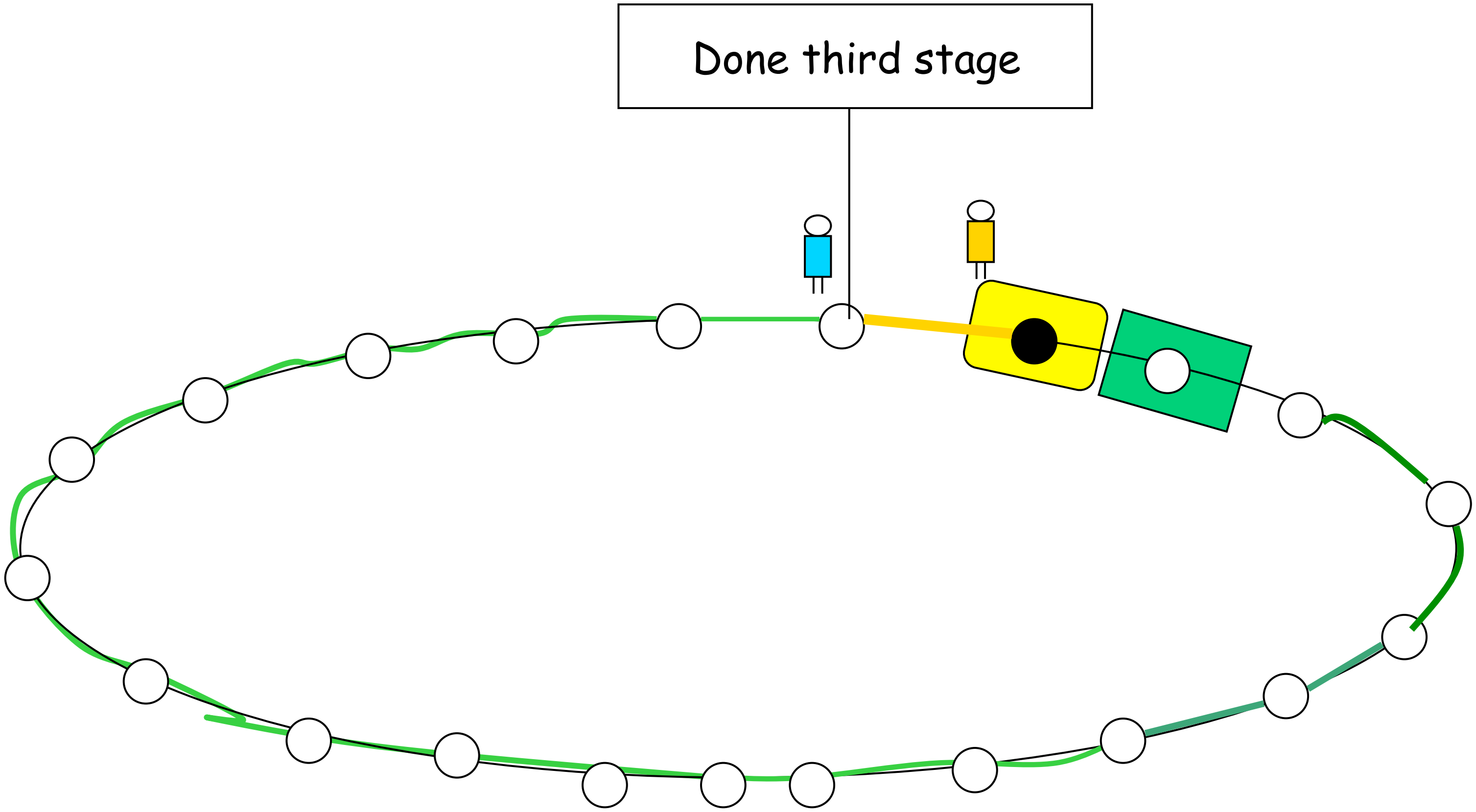
Explore your assigned region.



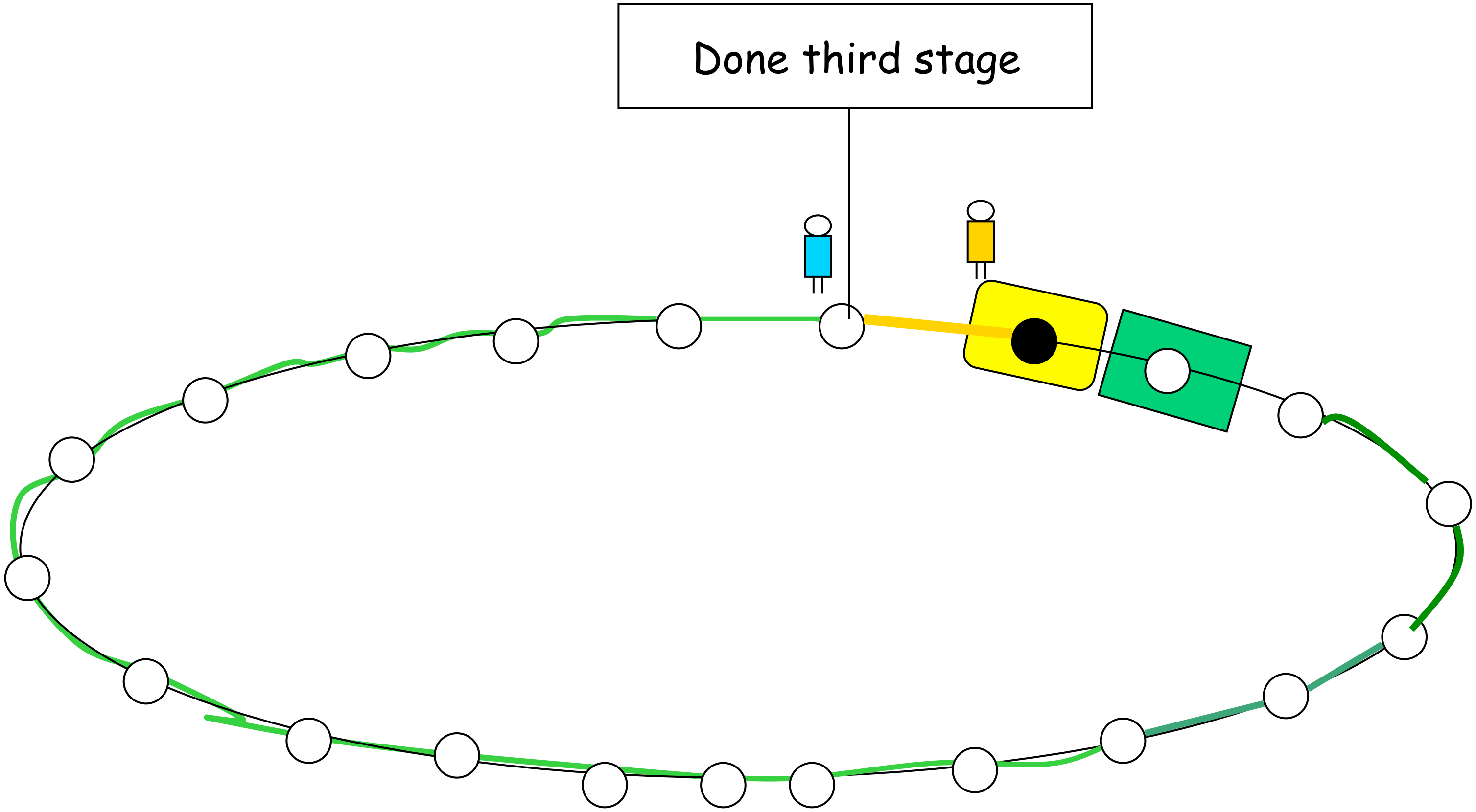
When finished, go to the last safe node of other agent



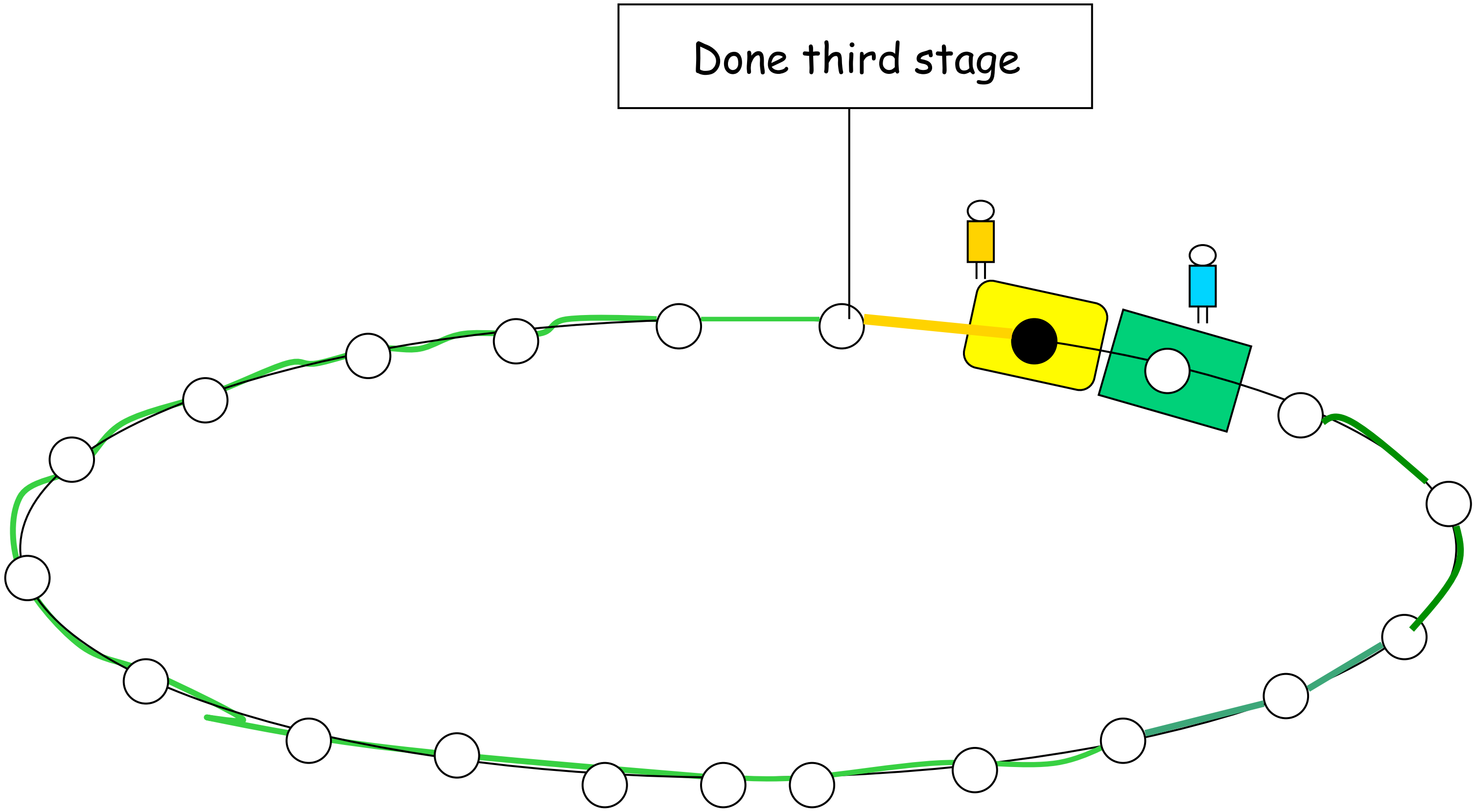
Leave a message for the other agent



Partition unexplored region into two areas of almost equal size

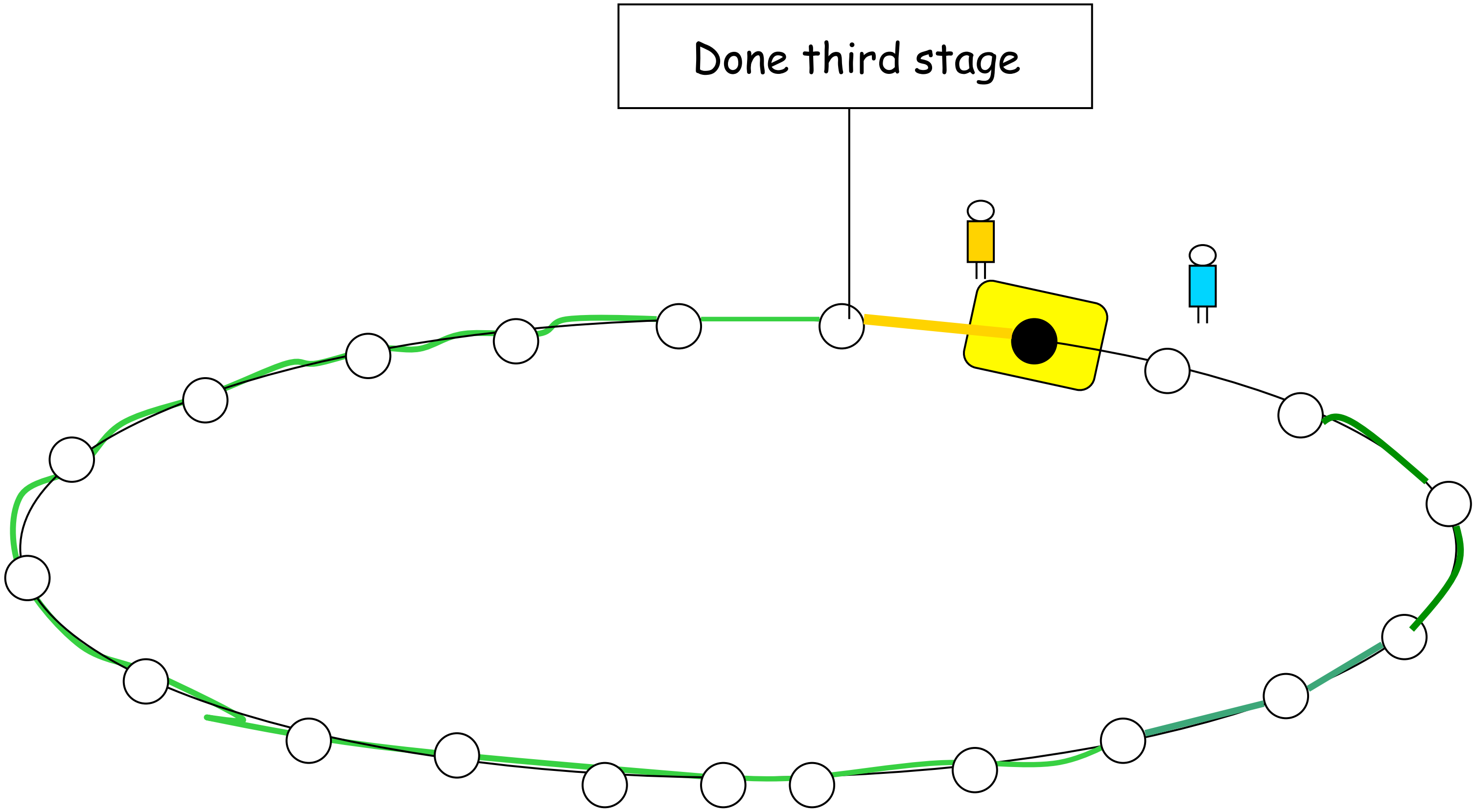


Explore your assigned region.



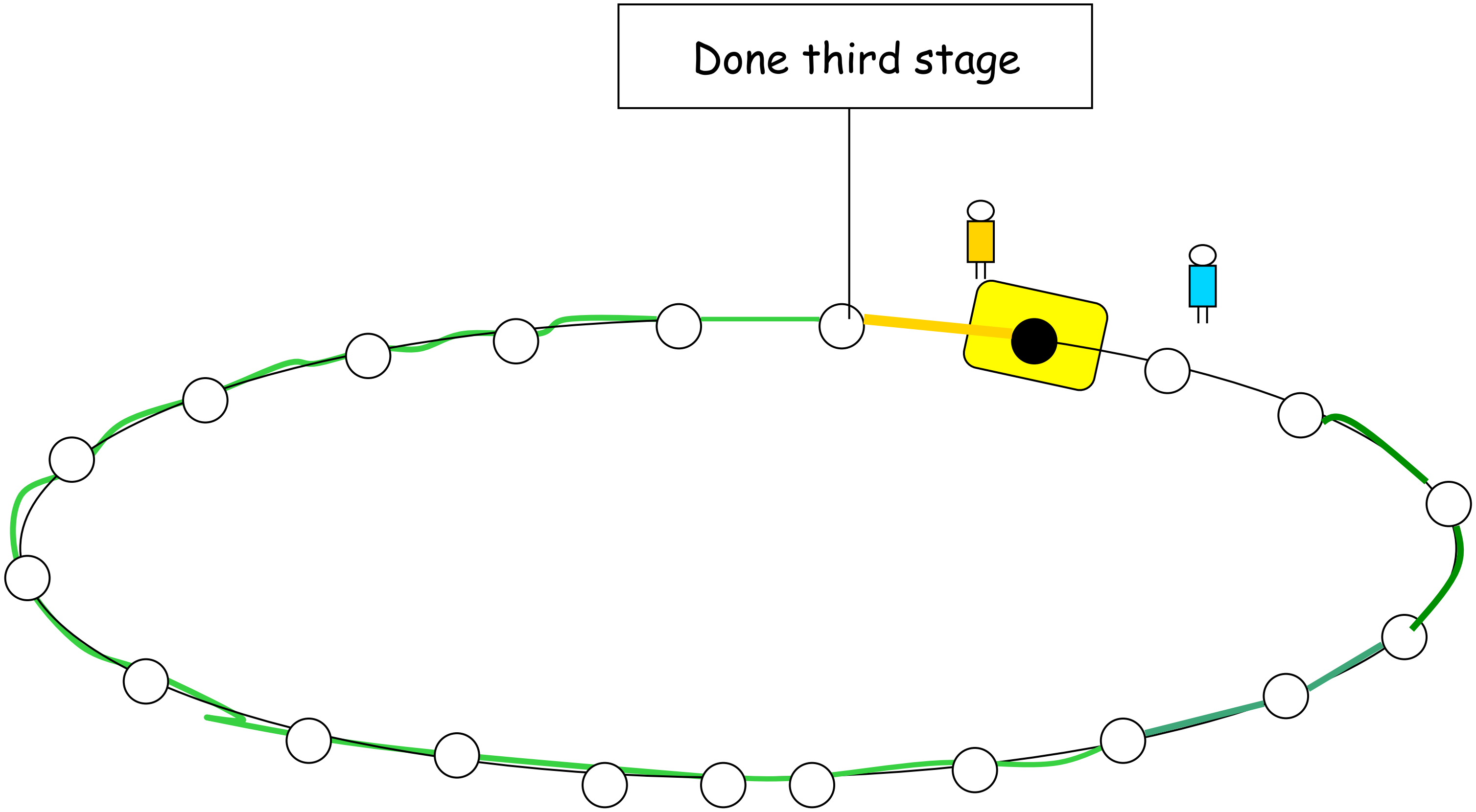
When finished, if only one node is unexplored

THAT IS THE BLACK HOLE !

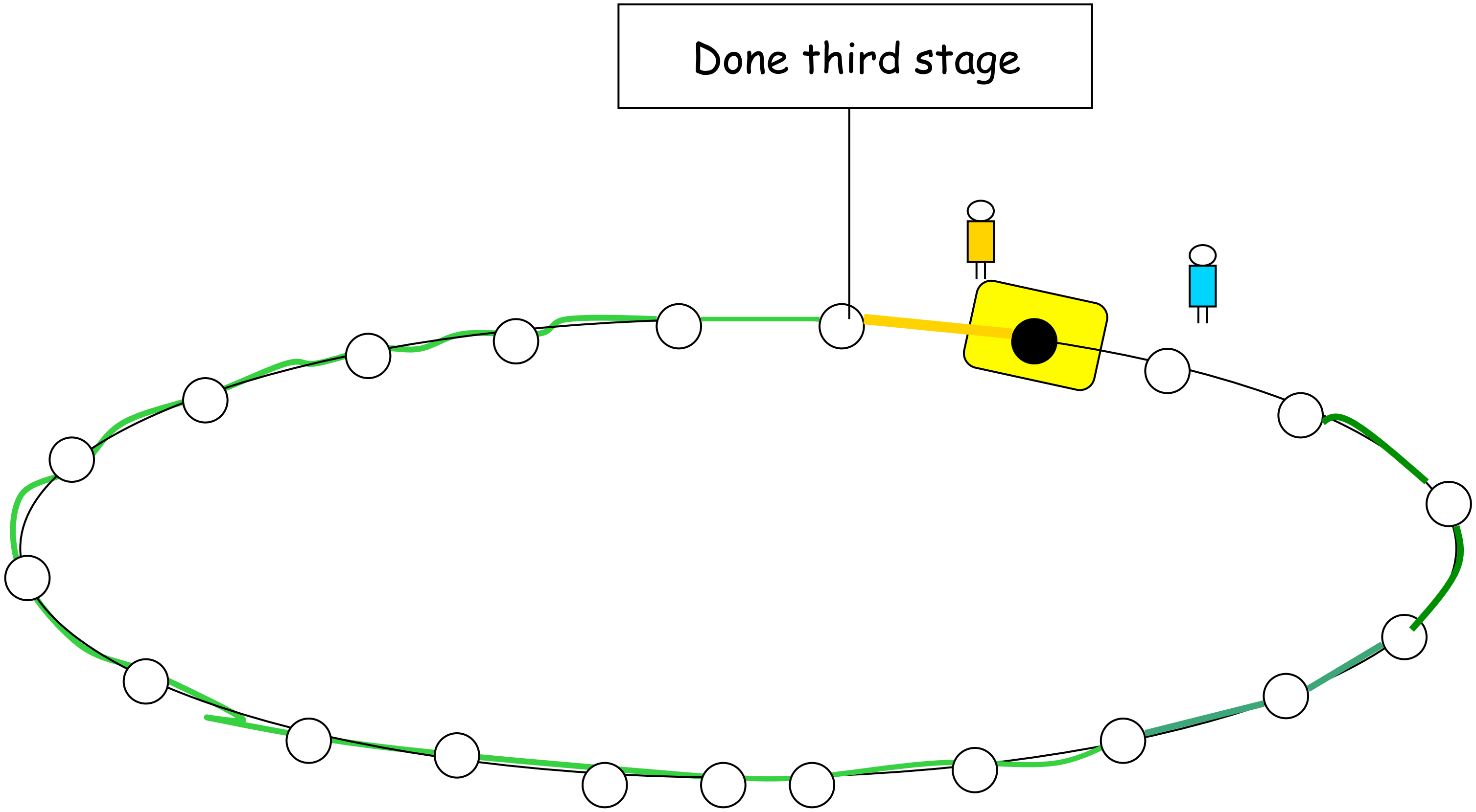


Done third stage

O(log n) stages



O(log n) stages
O(n) moves per stage



O(log n) stages
O(n) moves per stage

O(n log n) moves

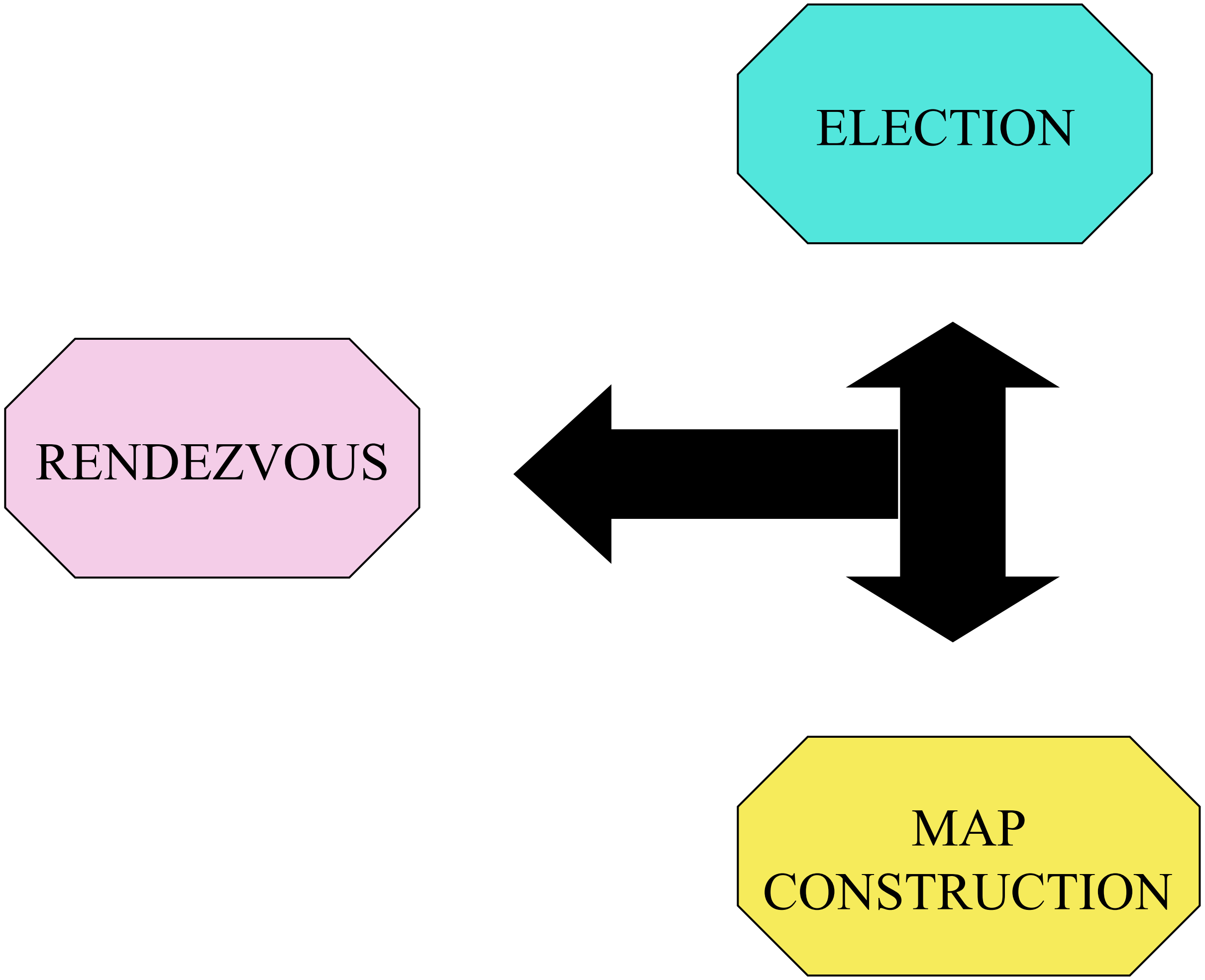
OPTIMAL

Dobrev, Flocchini, Santoro ALGORITHMICA 2007

Whiteboards

This result for asynchronous agents is obtained using whiteboards

Whiteboards



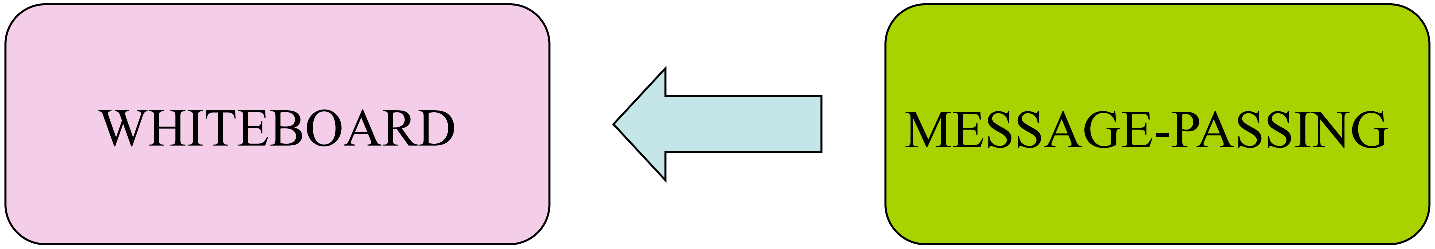
ELECTION

RENDEZVOUS

MAP CONSTRUCTION

Equivalent

Whiteboards



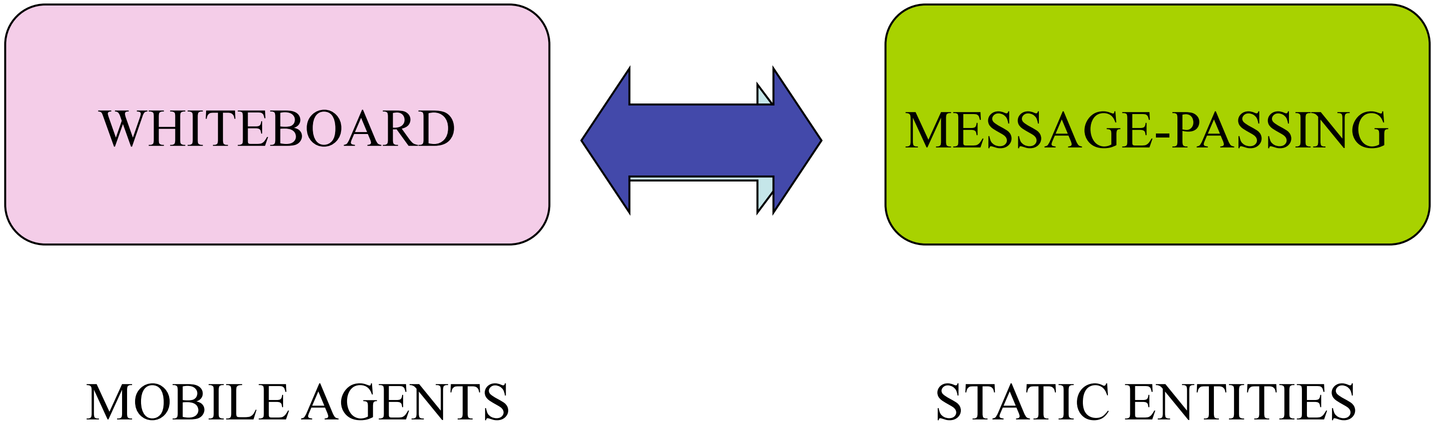
WHITEBOARD

MESSAGE-PASSING

MOBILE AGENTS

STATIC ENTITIES

Whiteboards are powerful



WHITEBOARD

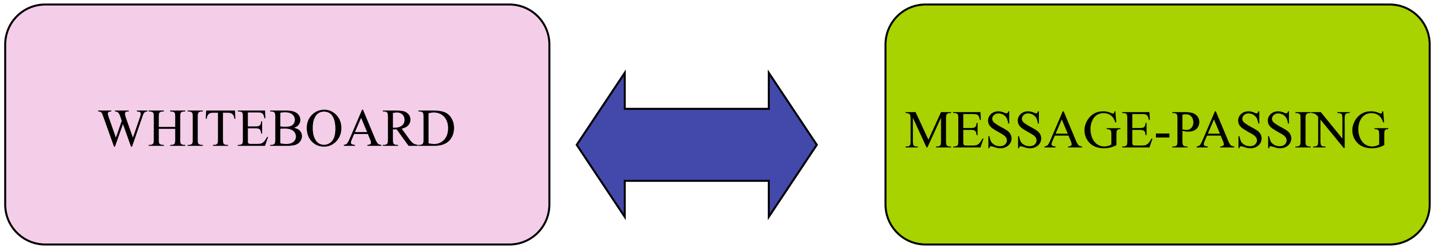
MESSAGE-PASSING

MOBILE AGENTS

STATIC ENTITIES

Chalopin, Godard, Metiver,Ossamy OPODIS' 06

Whiteboards are powerful



WHITEBOARD

MESSAGE-PASSING

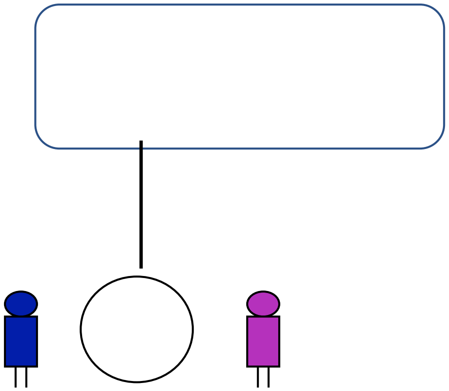
k>0 MOBILE AGENTS

n STATIC ENTITIES

Das, Flocchini, Santoro, Yamashita, SIROCCO' 07

Whiteboards are powerful

- Allow to **BREAK SYMMETRY**

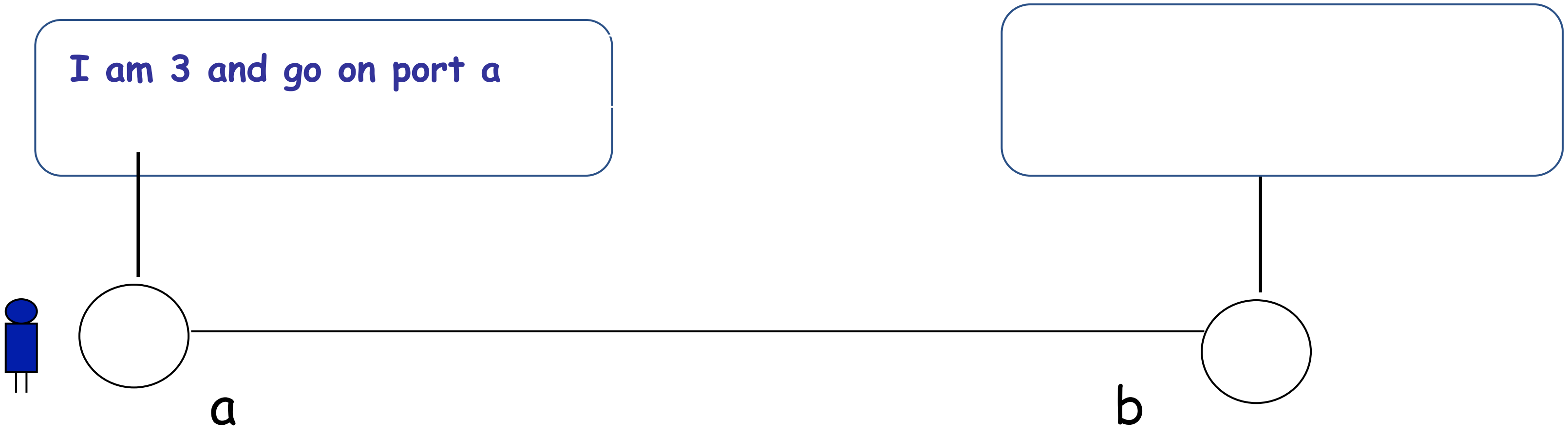


Whiteboards are powerful

- Allow to BREAK SYMMETRY
- Allow to ASSUME FIFO LINKS

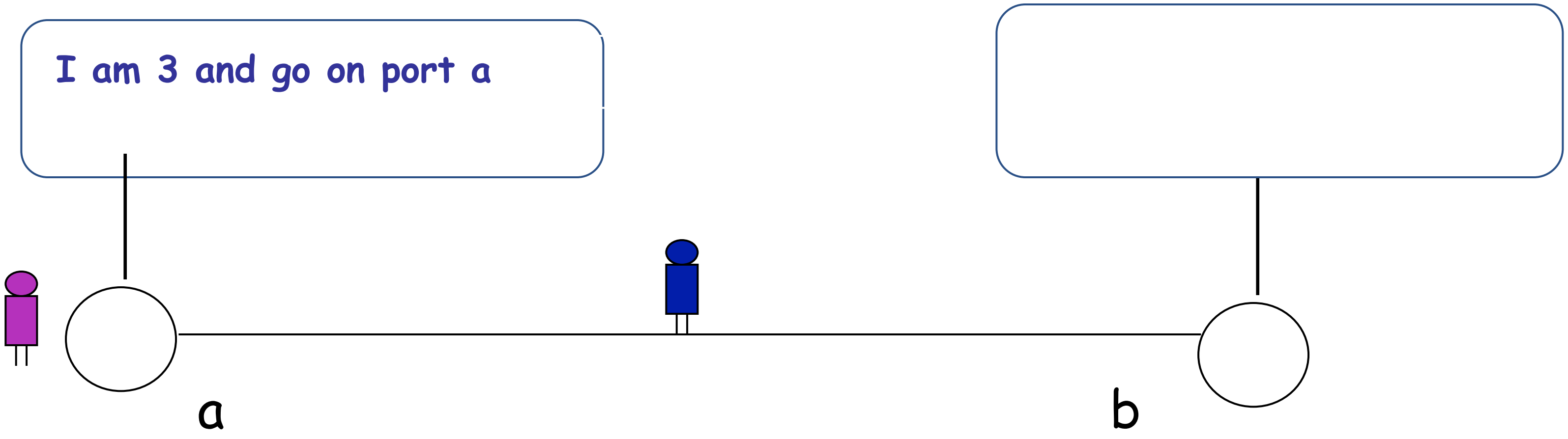
Whiteboards are powerful

- Allow to BREAK SYMMETRY
- Allow to ASSUME FIFO LINKS



Whiteboards are powerful

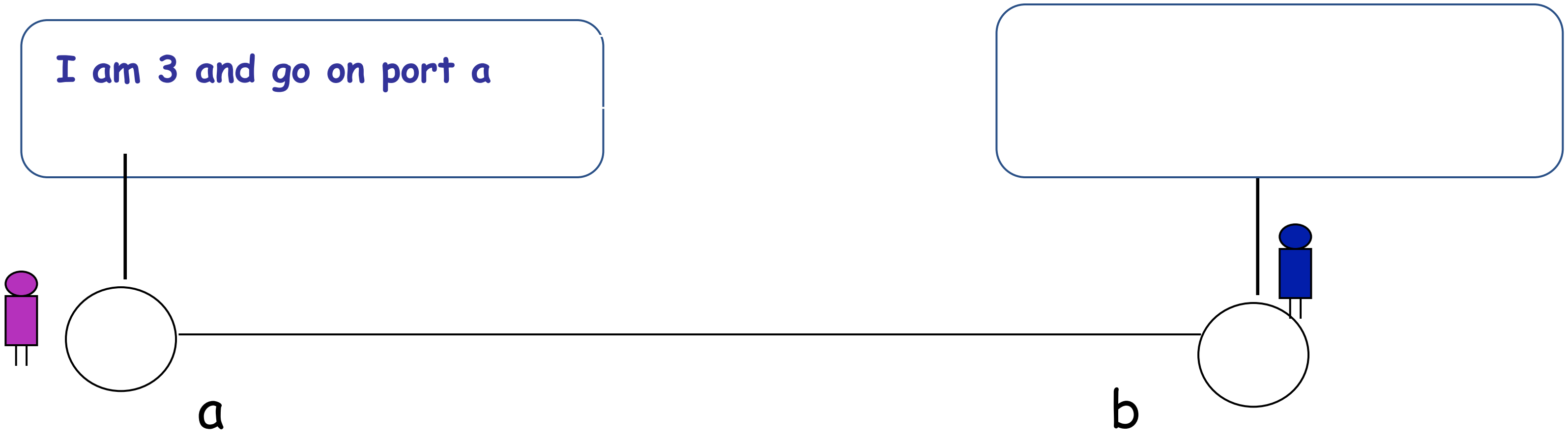
- Allow to BREAK SYMMETRY
- Allow to ASSUME FIFO LINKS



Whiteboards are powerful

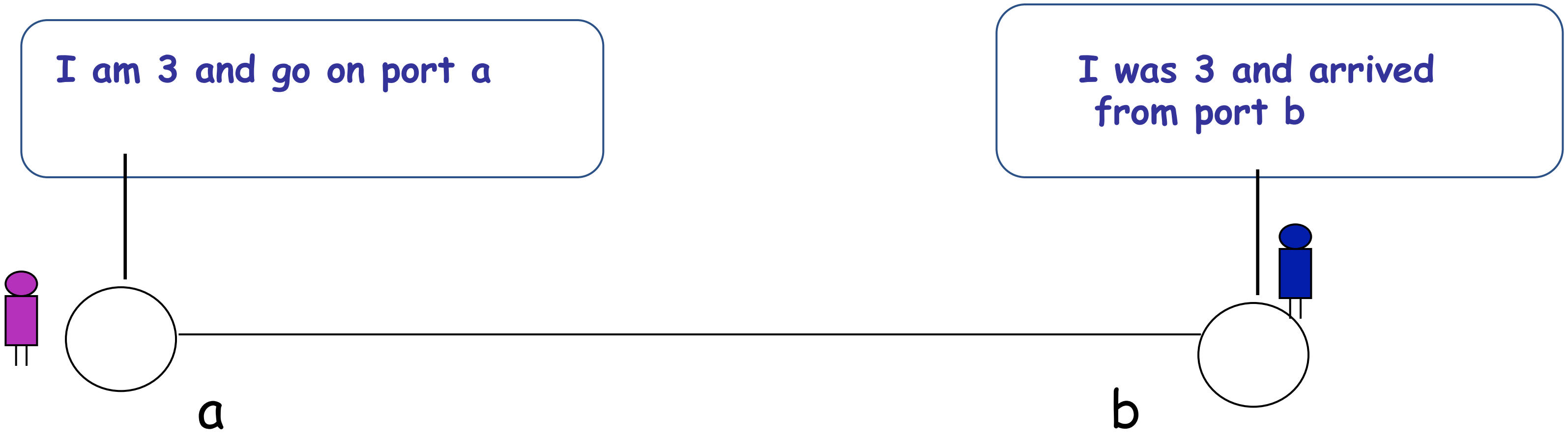
- Allow to BREAK SYMMETRY

- Allow to ASSUME FIFO LINKS



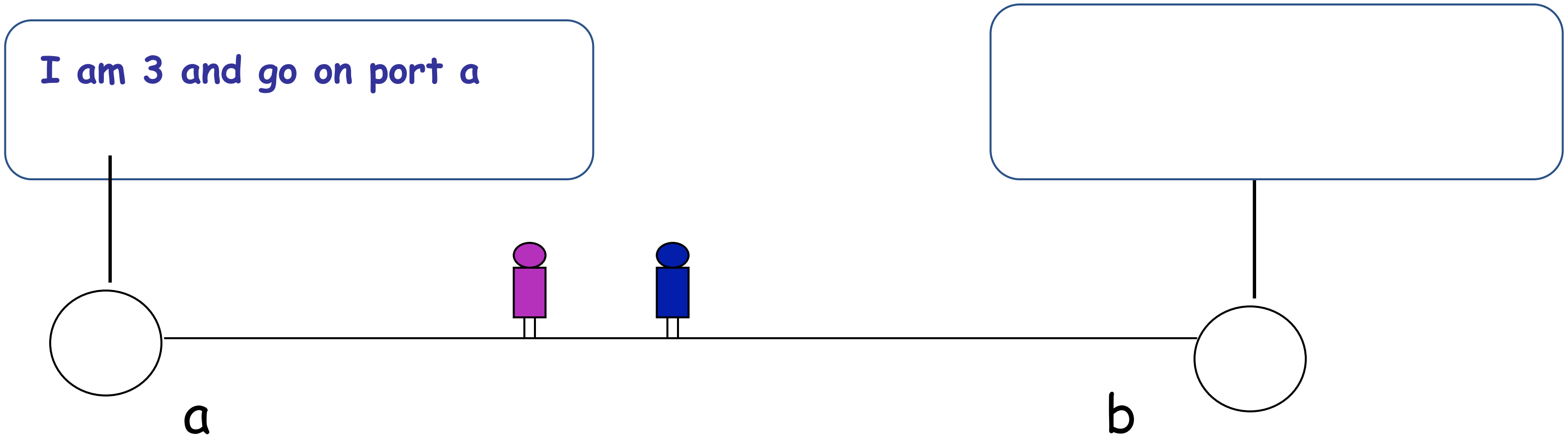
Whiteboards are powerful

- Allow to BREAK SYMMETRY
- Allow to ASSUME FIFO LINKS



Whiteboards are powerful

- Allow to BREAK SYMMETRY
- Allow to ASSUME FIFO LINKS

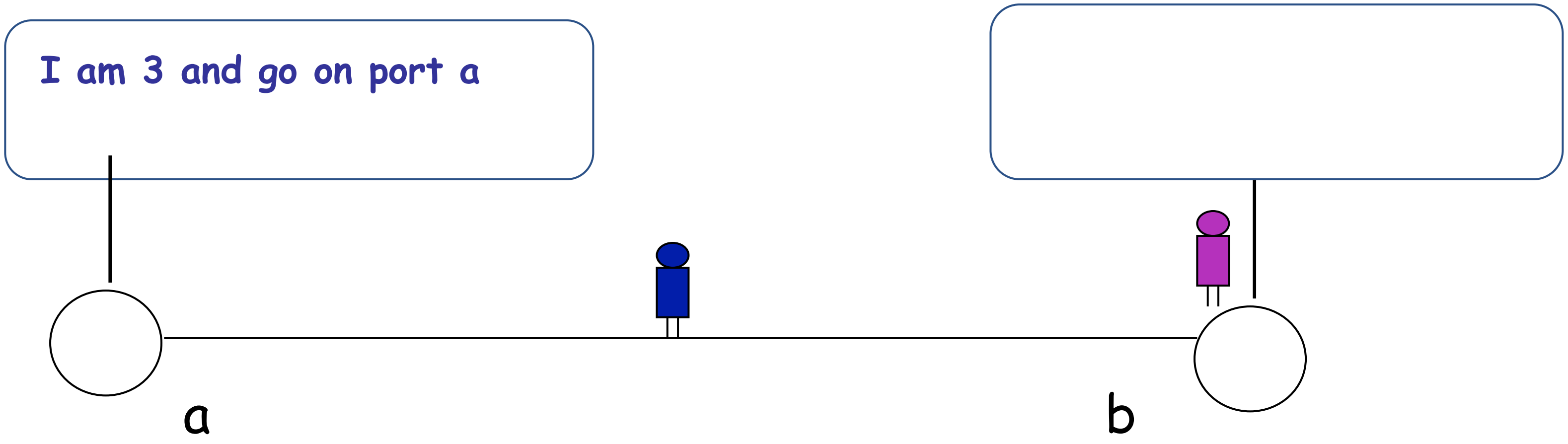


I am 3 and go on port a

Whiteboards are powerful

- Allow to BREAK SYMMETRY

- Allow to ASSUME FIFO LINKS



Whiteboards have drawbacks

- Explicit (no privacy)

Whiteboards have drawbacks

- Explicit (no privacy)
- Expensive
  - persistent dedicated memory at each node
  - fair access in mutual exclusion

Is it possible to solve problems

asynchronously in a model **weaker** than

whiteboards ?

Is it possible to locate the black hole asynchronously in a model weaker than whiteboards ?

Is it possible to **locate the black hole** **asynchronously** in a model **weaker** than whiteboards ?

Interaction and Communication:

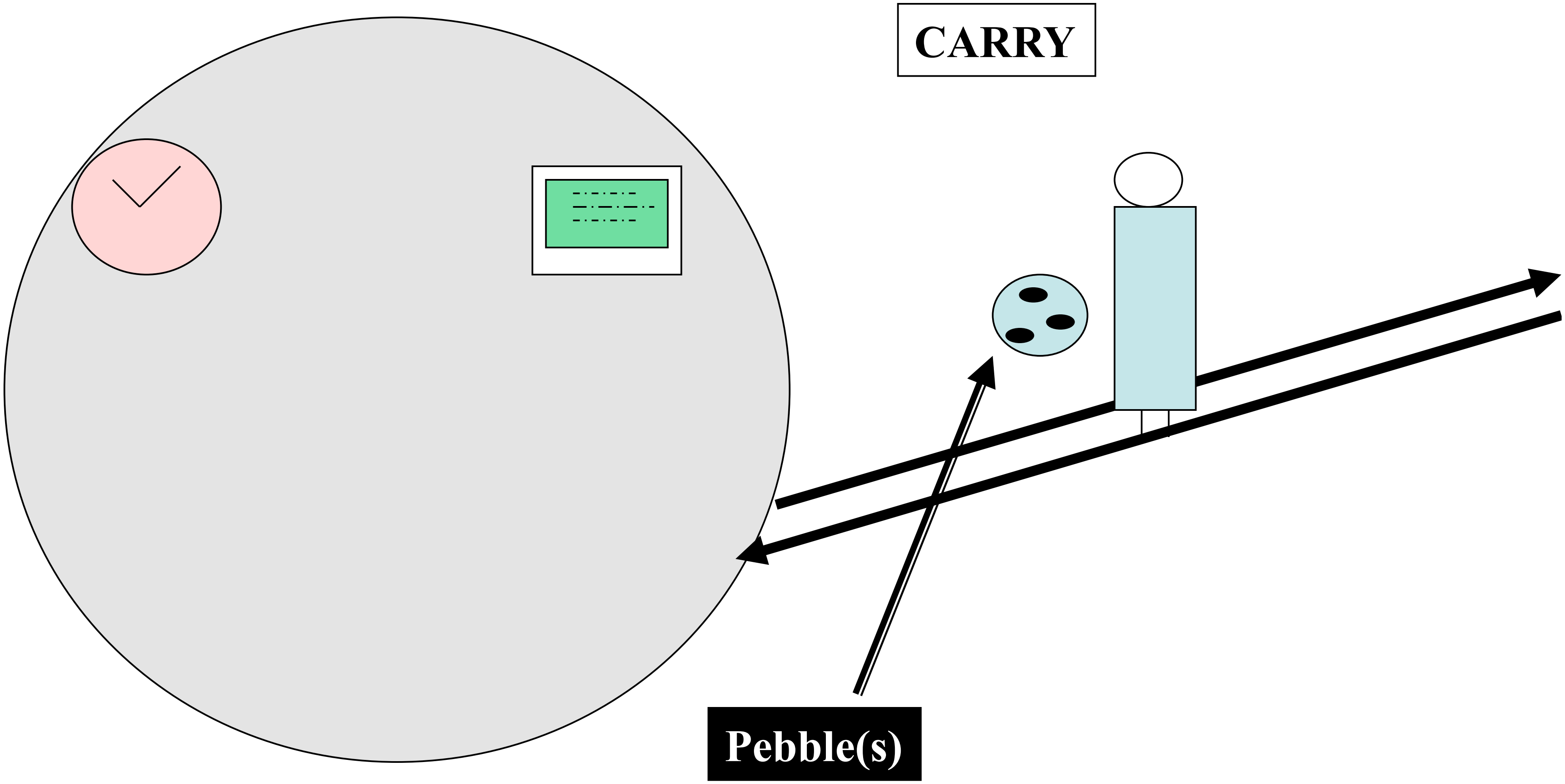
Vision

Face-to-Face

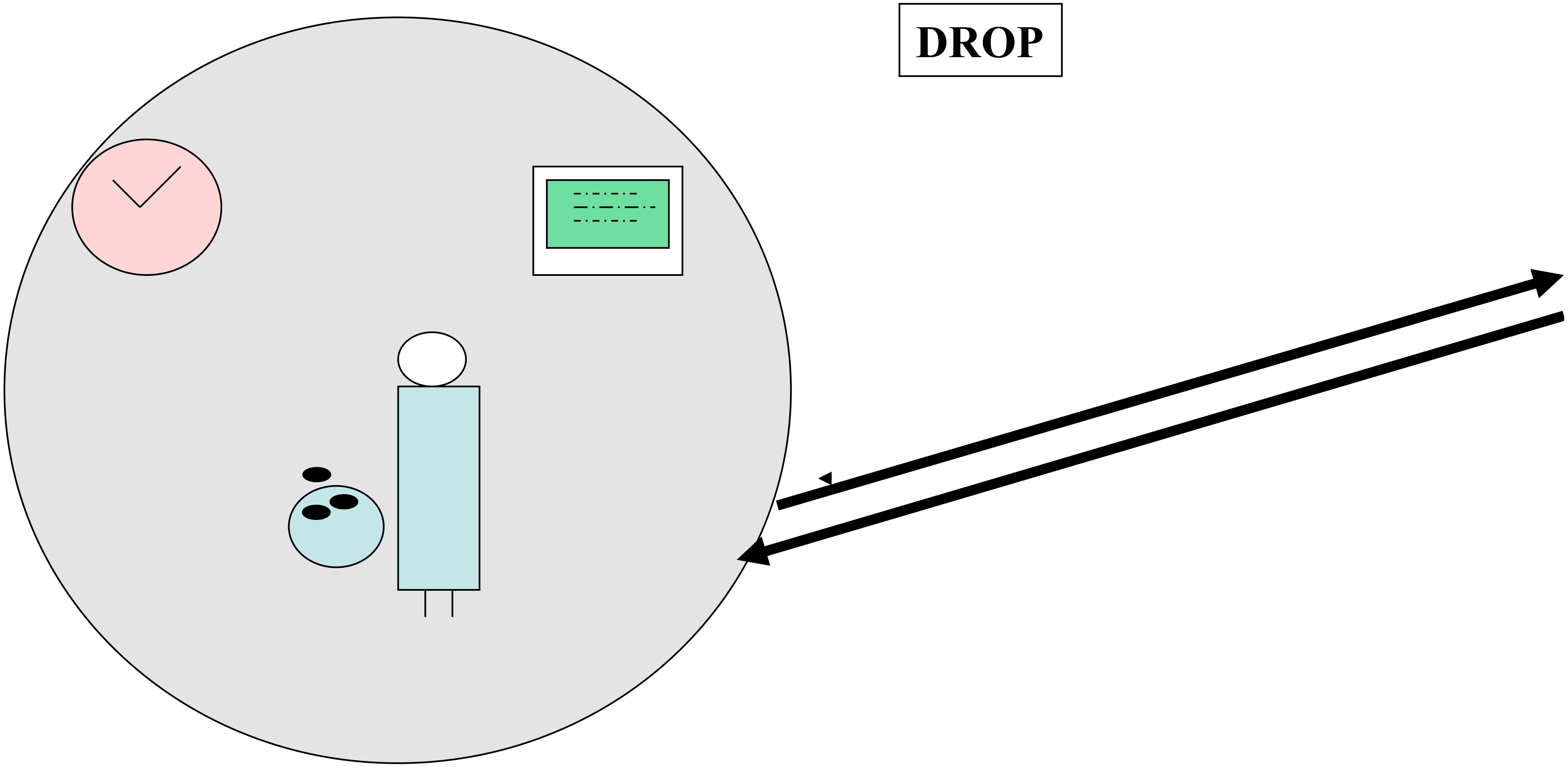
Tokens

Whiteboards

Tokens

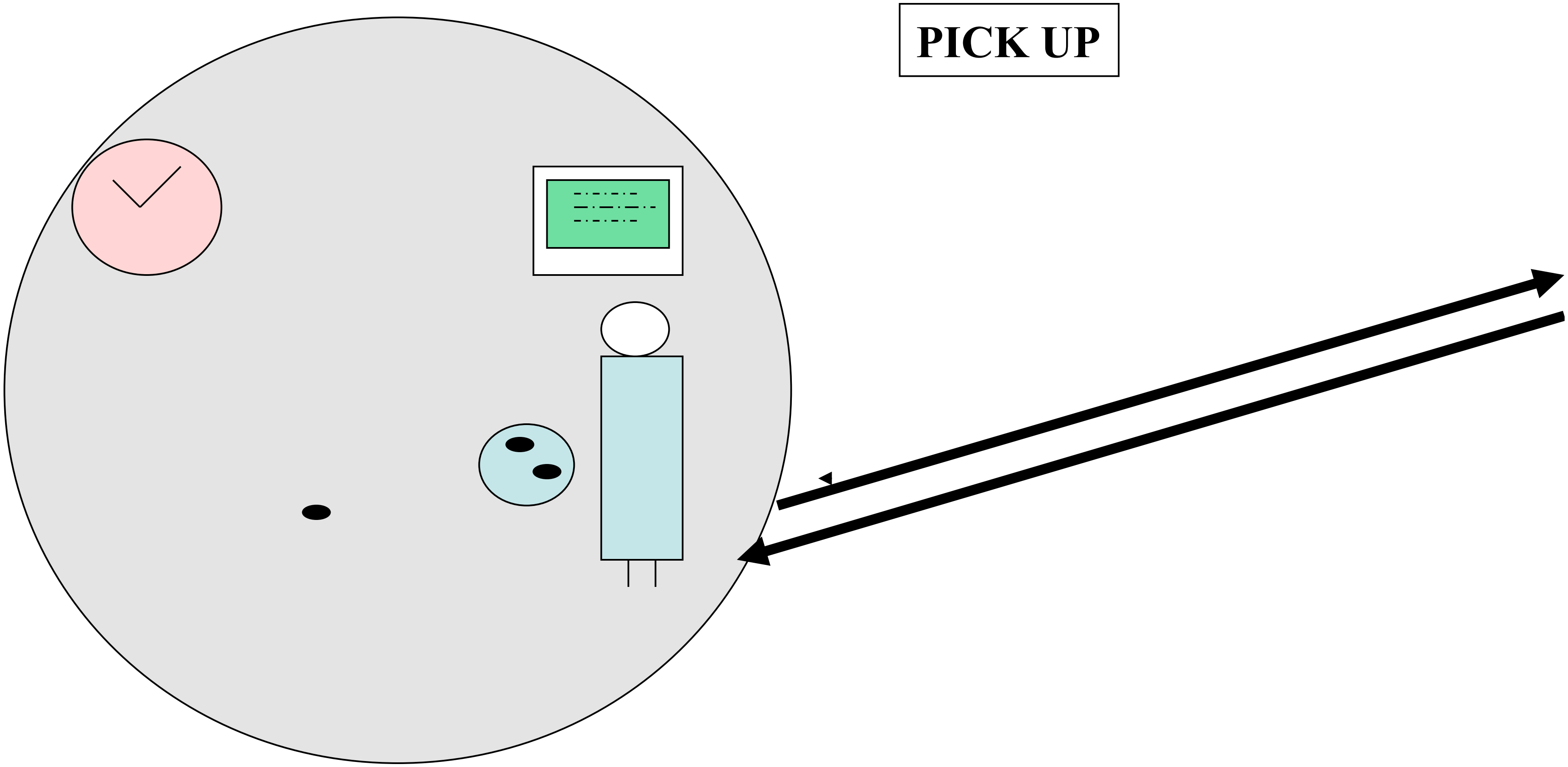


Tokens



DROP

Tokens



TOKEN < WHITEBOARD

Is it possible to locate the black hole

without a map

asynchronously using identical **pebbles** ?

TOKENS < WHITEBOARDS

Is it possible to locate the black hole

without a map

asynchronously using identical **pebbles** ?

YES !

-**with a team of Δ+1 agents** (optimal !)
-**with one** (identical) **pebble per agent**
-**with a polynomial number of moves**

$O(\Delta^2 m^2 n^7)$!!!

TOKENS < WHITEBOARDS

Is it possible to locate the black hole

without a map

asynchronously using identical **tokens** ?

YES !

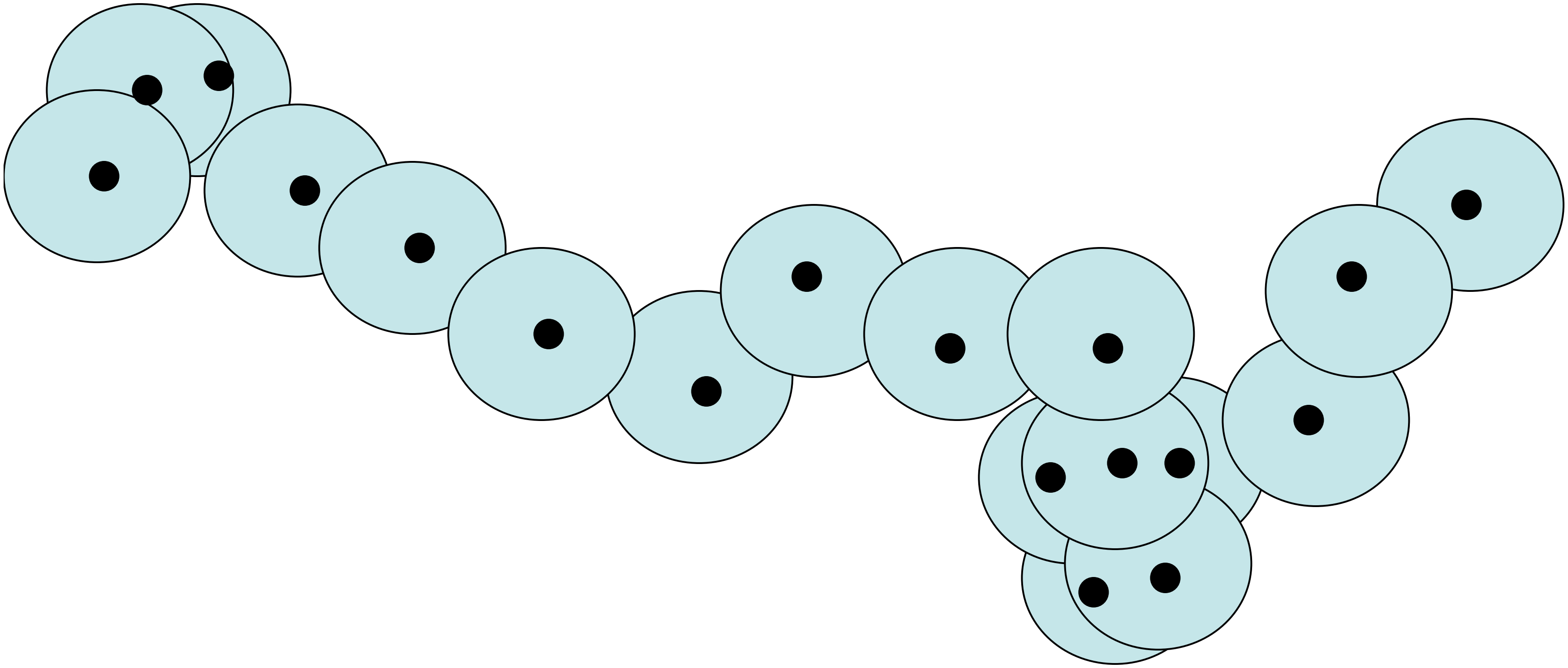
The algorithm is however rather complex

Dobrev, Flocchini, Kralovic, Santoro, *Theoretical Computer Science*, 2013

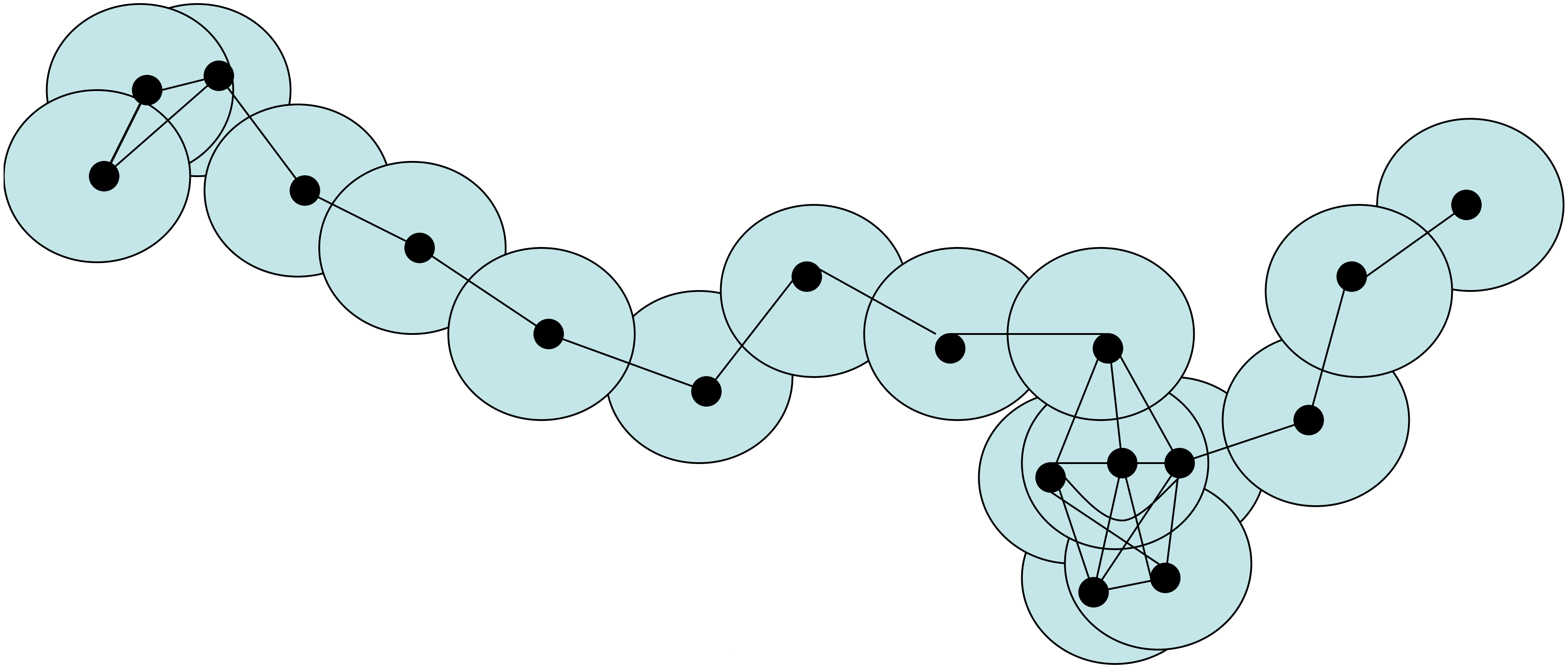
New Directions

AGENTS MOVING IN TIME-VARYING GRAPHS

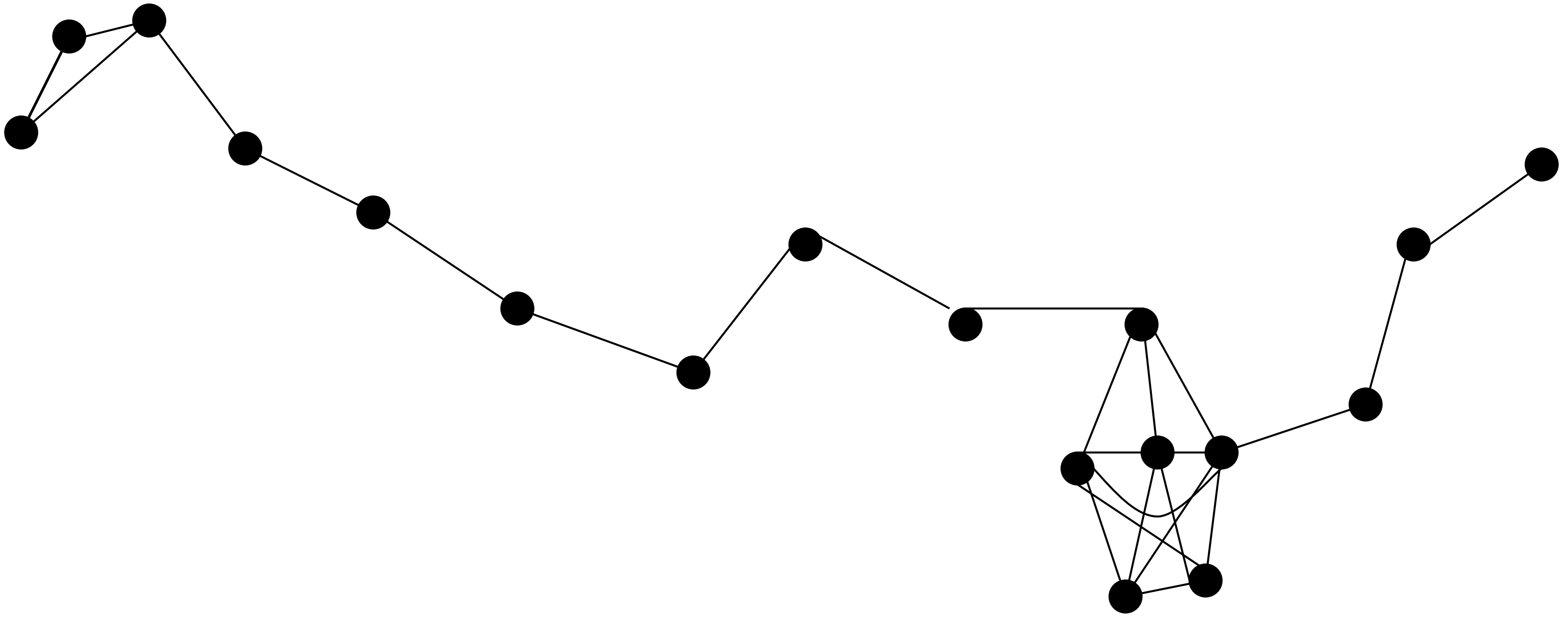
WIRELESS MOBILE ENTITIES



WIRELESS MOBILE ENTITIES

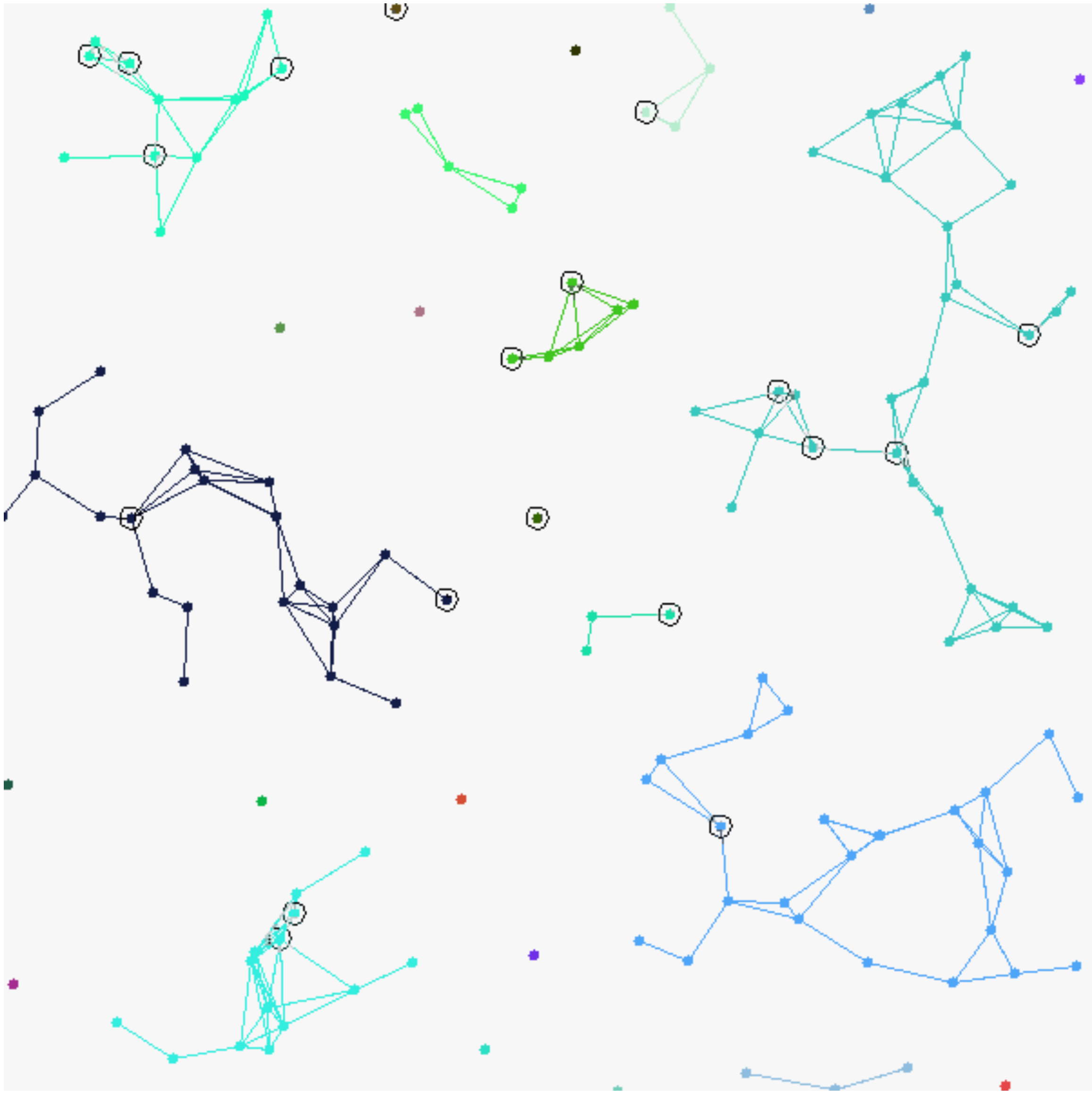


WIRELESS MOBILE ENTITIES



Communication Graph

WIRELESS MOBILE ENTITIES



WIRELESS MOBILE ENTITIES

end-to-end connectivity does not necessarily hold

the network might be always disconnected

still …

communication may be available over **time** and **space**

making **computations** feasible

WIRELESS MOBILE ENTITIES

Delay tolerant networks

Disruption tolerant networks

Challenged networks

Opportunistic networks

Modeled as TIME-VARYING GRAPHS

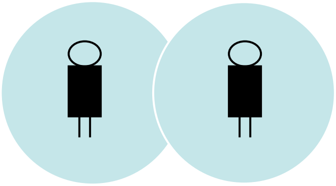
A. Casteigts, P. Flocchini, W. Quattrociocchi, N.Santoro.
Time-varying graphs and dynamic networks. *IJPEDS* 2012

Time-Varying Graph

TVG-entity (node, vehicle, sensor, ...)

Time-Varying Graph

Edge = connection between two TVG-entities
= (possibility of) interaction



Edge = **temporally** defined connection with specific properties

Time-Varying Graph

V TVG-**entities** ==> **nodes**

E **connections** between TVG-entities ==> **edges**

L **properties** of connections ==> **labels**

$$E \subseteq V \times V \times L$$

Time-Varying Graph

Connections

temporally defined

take place over a contiguous time span $\mathbf{T} \subseteq \mathbb{T}$

- **T** **lifetime** of the system.

TIME $\mathbb{T}$	Z	discrete-time systems
	R	continuous-time systems.

Time-Varying Graph (TVG)

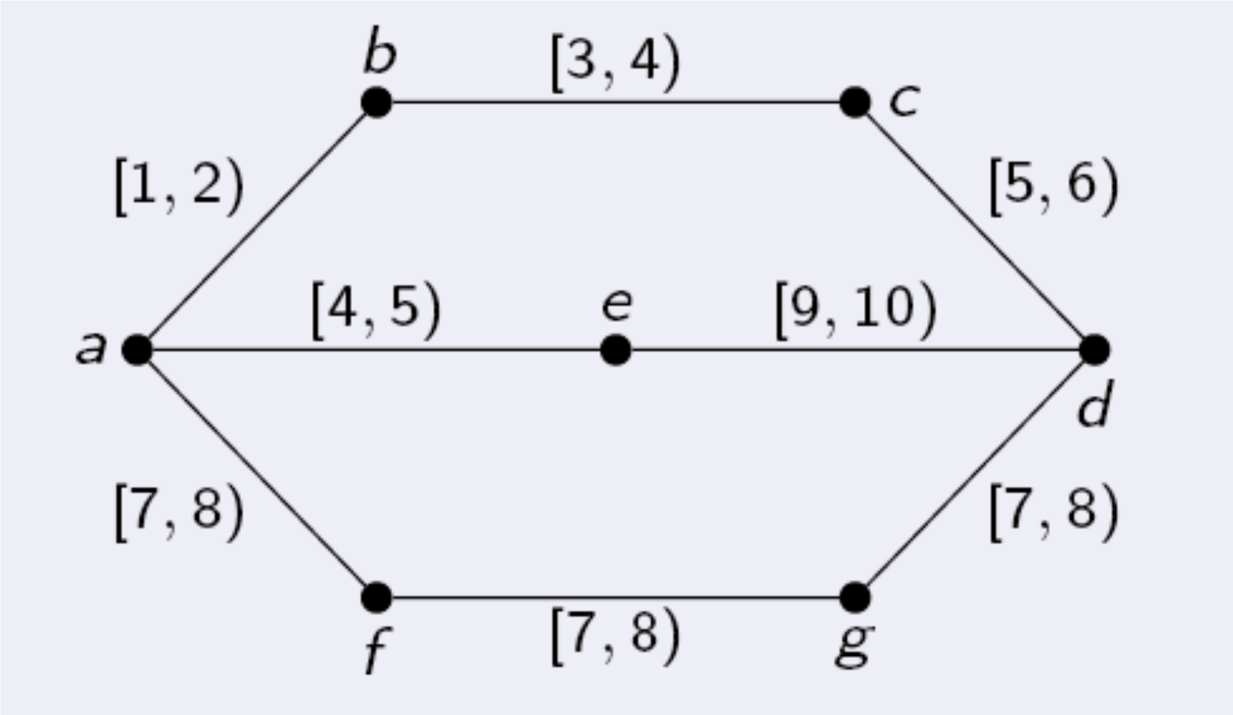
$\mathcal{G} = (V, E, \mathbf{T}, \psi, \rho, \zeta)$

$\psi : V \times \mathbf{T} \rightarrow \{0, 1\}$ **node presence** function
indicates whether a given node is available at a given time

$\rho : E \times \mathbf{T} \rightarrow \{0, 1\}$ **edge presence** function
indicates whether a given edge is available at a given time

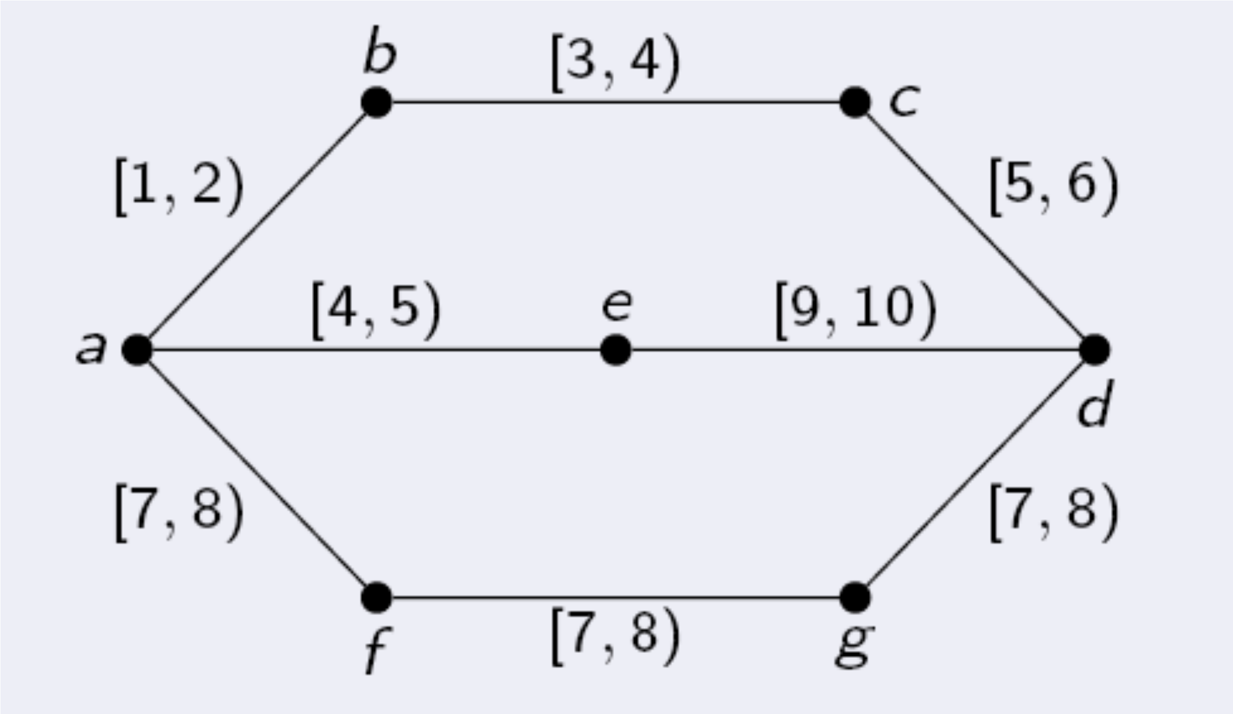
$\zeta : E \times \mathbf{T} \rightarrow \mathbf{T}$ **latency** function
indicates the time it takes for interaction to take place in a connection if starting at a given date

Time-Varying Graph



The interval(s) on an edge represent the period of time when it exists.

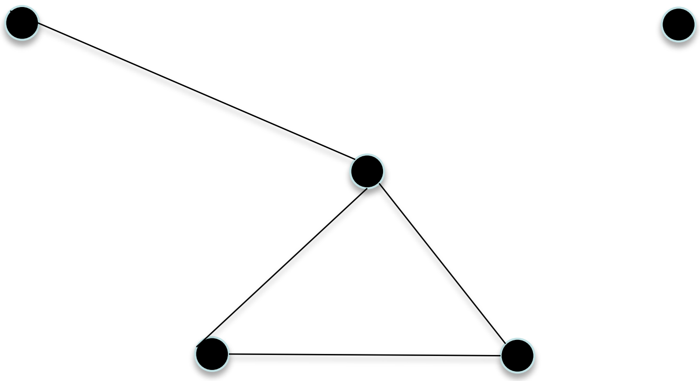
Time-Varying Graph



Compact graph representation of network dynamics

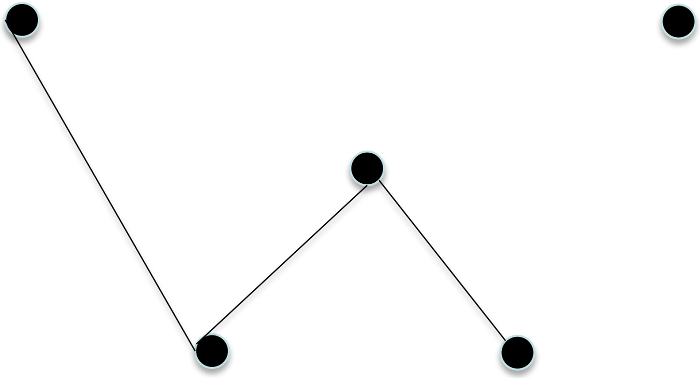
Time-Varying Graphs

t=0



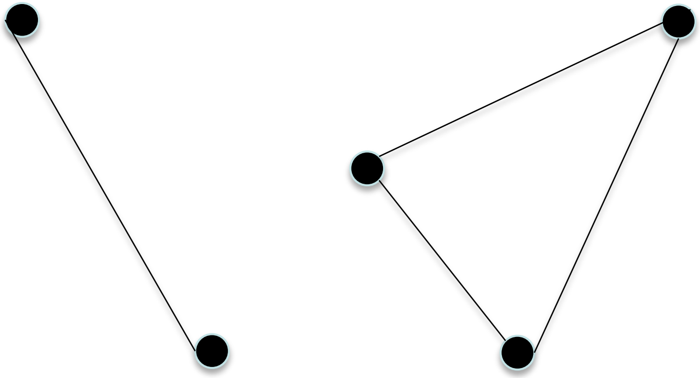
Time-Varying Graphs

t=1



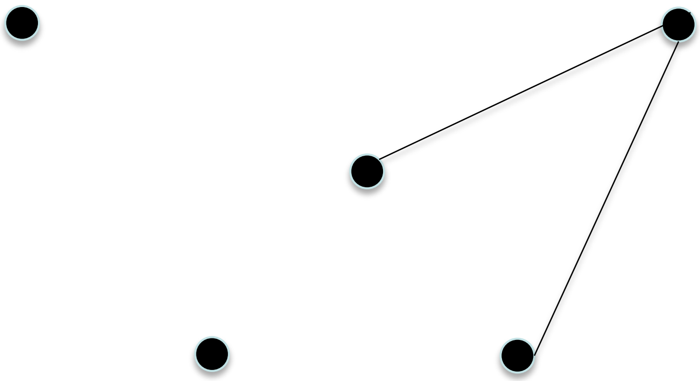
Time-Varying Graphs

t=2

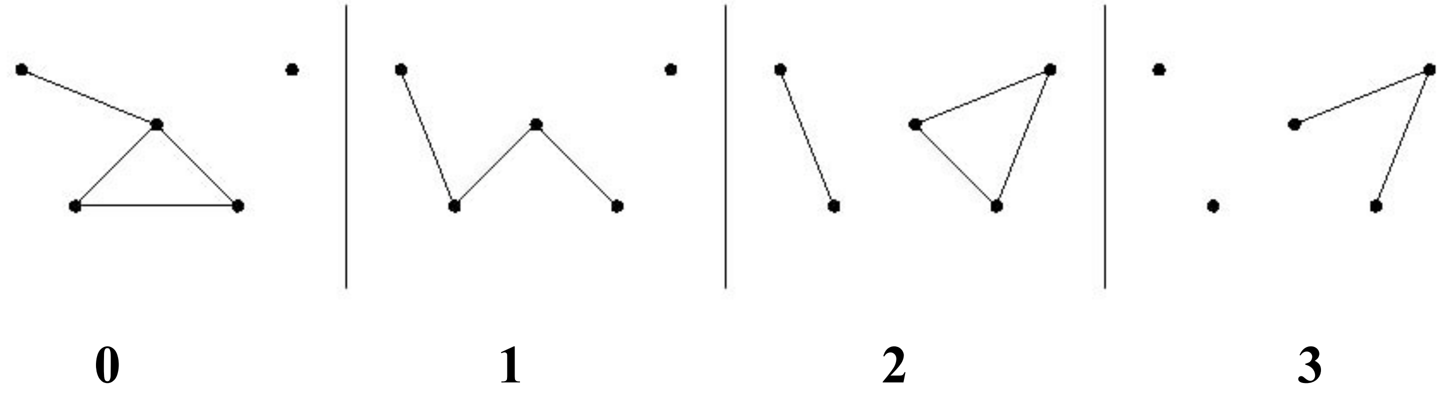


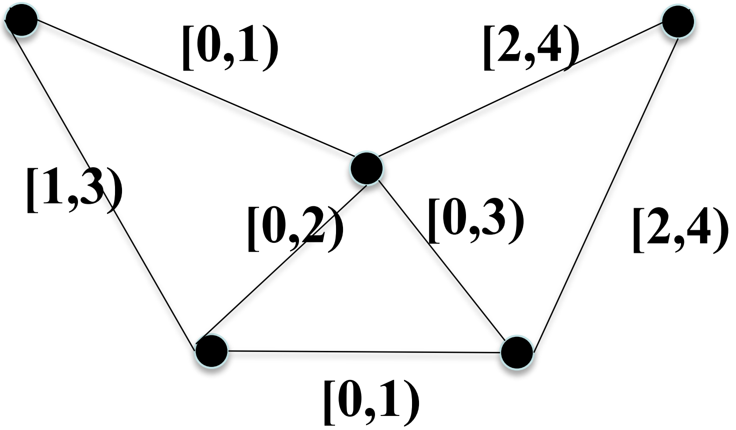
Time-Varying Graphs

t=3

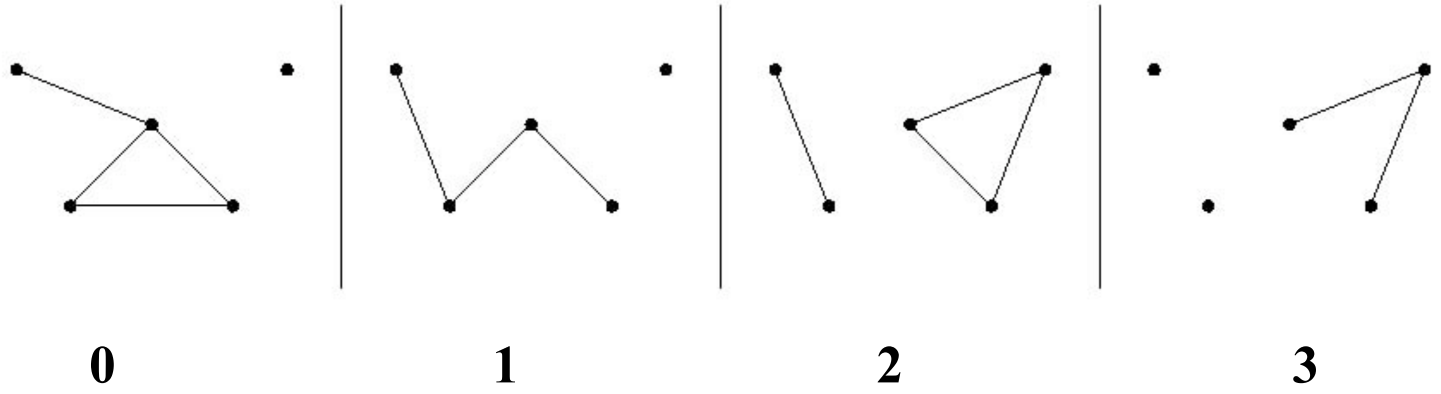


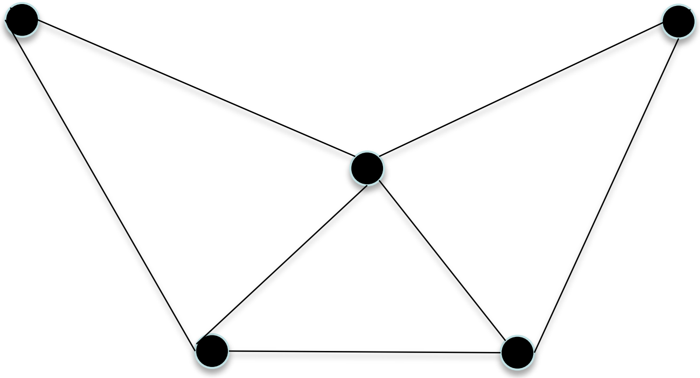
Time-Varying Graphs





Time-Varying Graphs

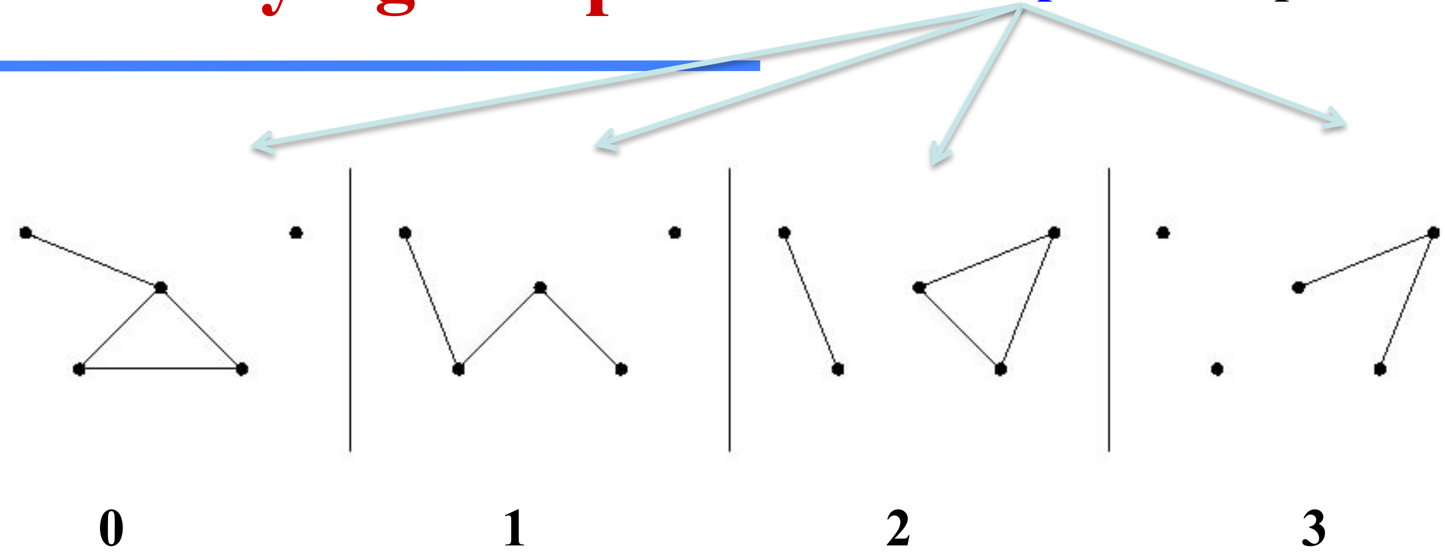


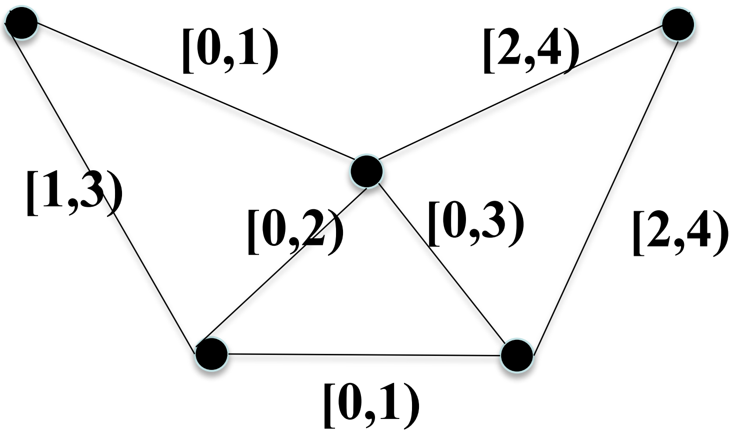


Footprint
of the graph in
an interval of time -
[0,4) in this case

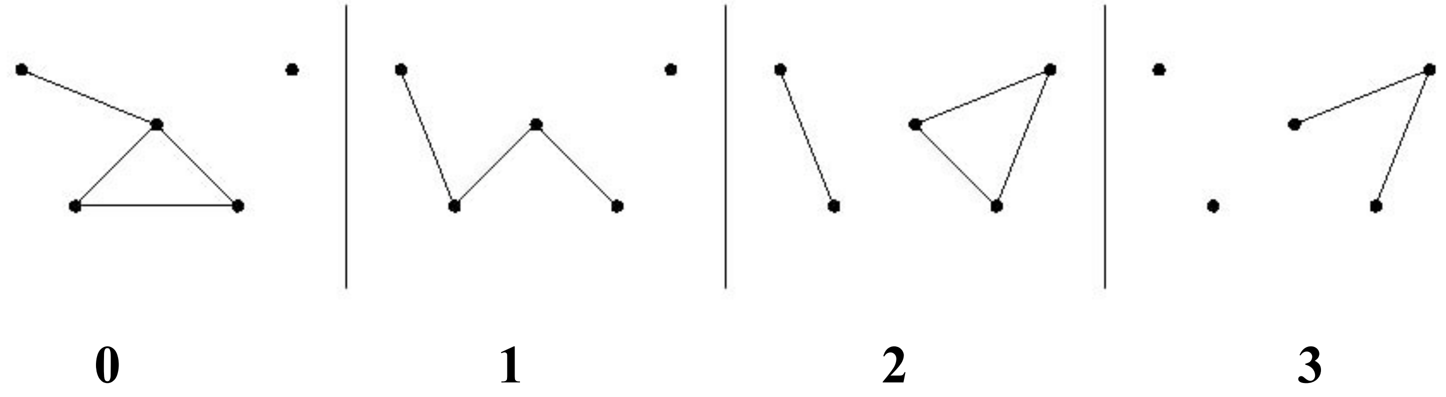
Time-Varying Graphs

Snapshots at specific times





Time-Varying Graphs

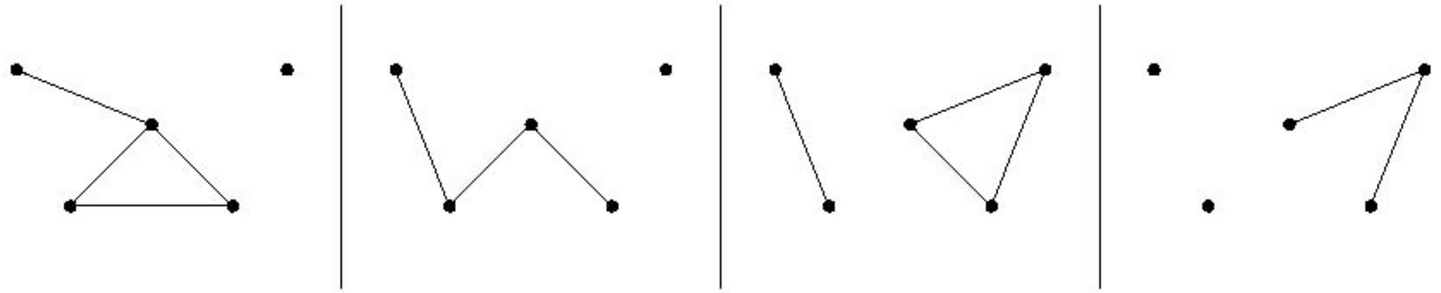


sequence of SNAPSHOTS

$G(t_1)$, $G(t_2)$, $G(t_3)$, $G(t_4)$, …

EVOLVING GRAPH

Time-Varying Graphs

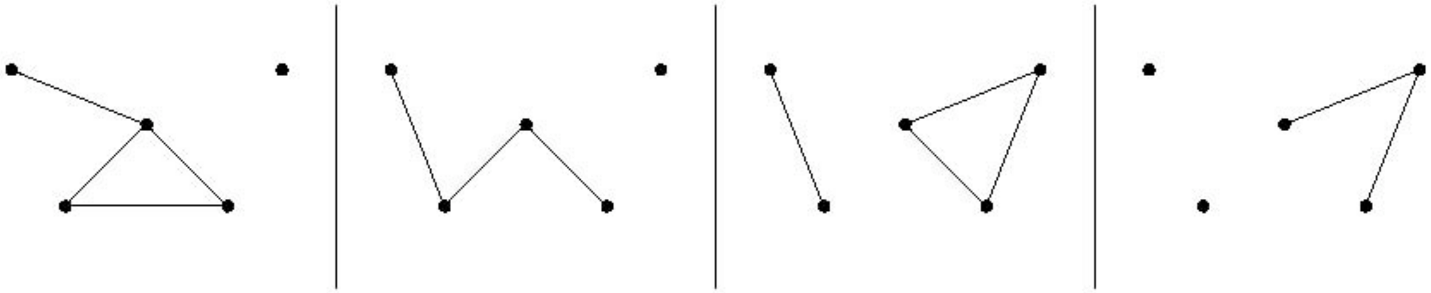


F. Harary and G. Gupta. “Dynamic graph models”. *Mathematical and Computer Modelling*, 1997.

$G(t_1)$, $G(t_2)$, $G(t_3)$, $G(t_4)$, …

EVOLVING GRAPH

Time-Varying Graphs



F. Harary and G. Gupta. “Dynamic graph models”. *Mathematical and Computer Modelling*, 1997.

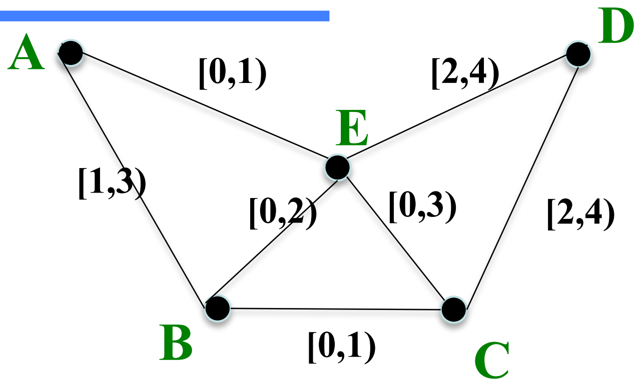
A.Ferreira “Building a reference combinatorial model for MANETs”. *IEEE Network*, 2007.

$G(t_1),\ G(t_2),\ G(t_3),\ G(t_4), \ldots$

EVOLVING GRAPH

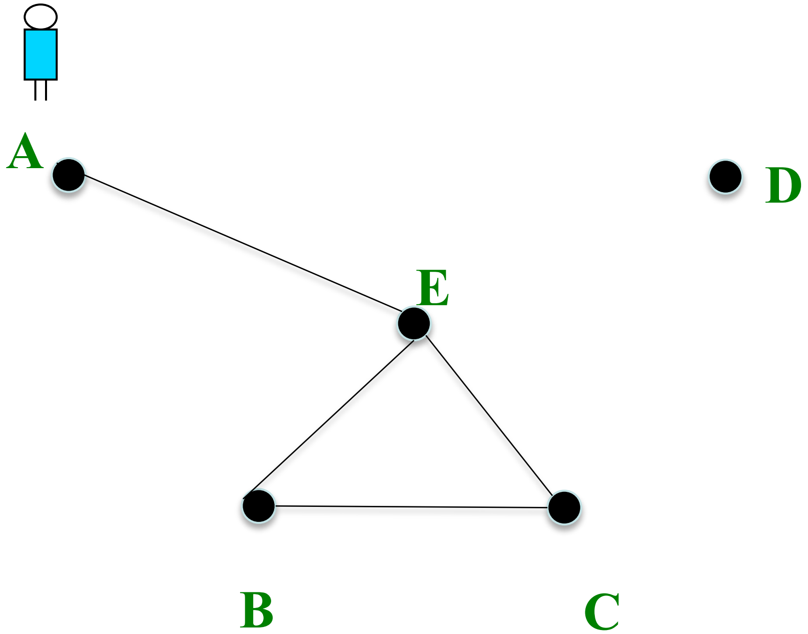
Journey: moving on a TVG

mobile agent

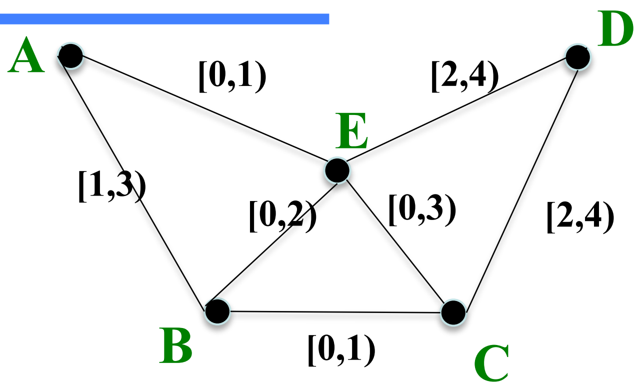


Traverse path [A,E,D]

t=0



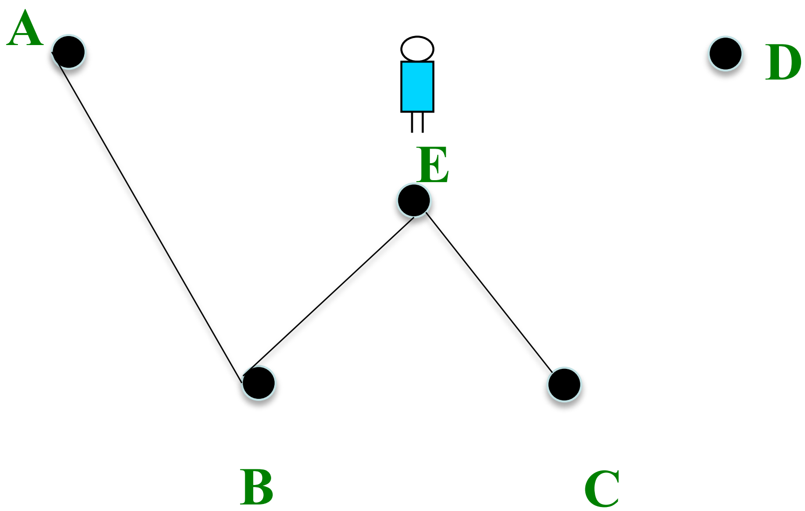
Example



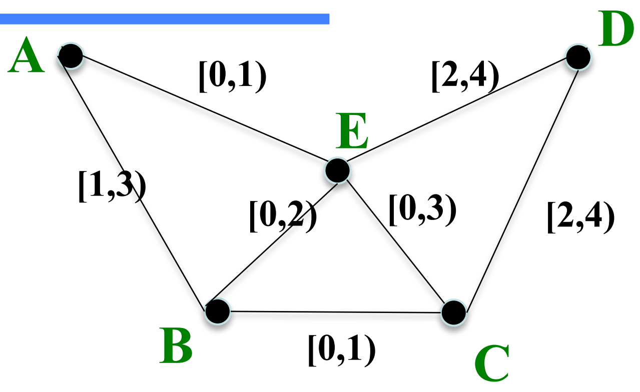
Traverse path [A,E,D]

t=1

WAIT at E



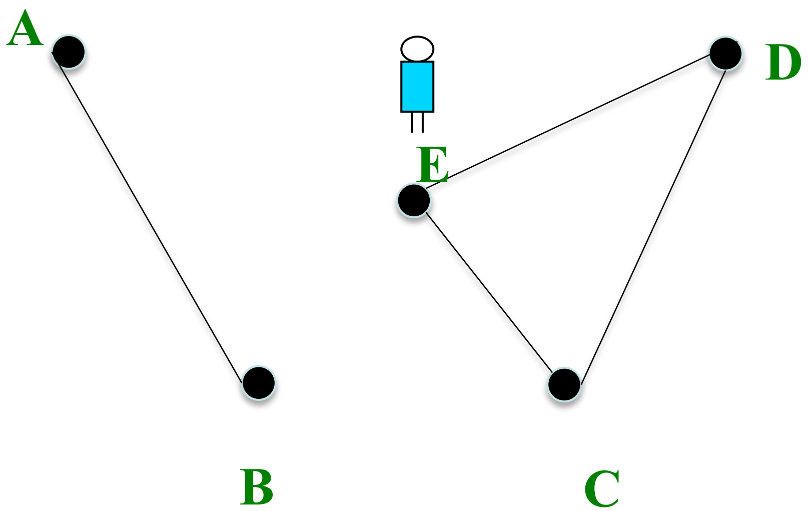
Example



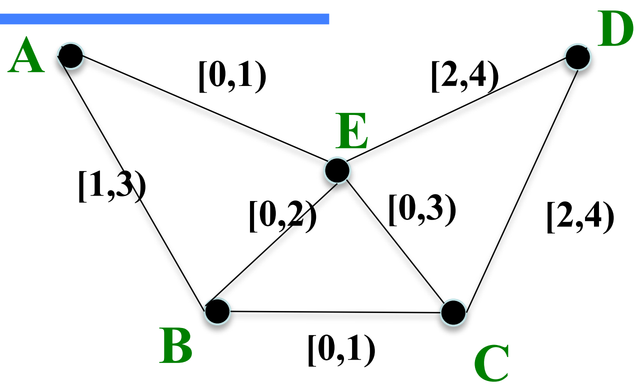
Traverse path [A,E,D]

t=3

t=2



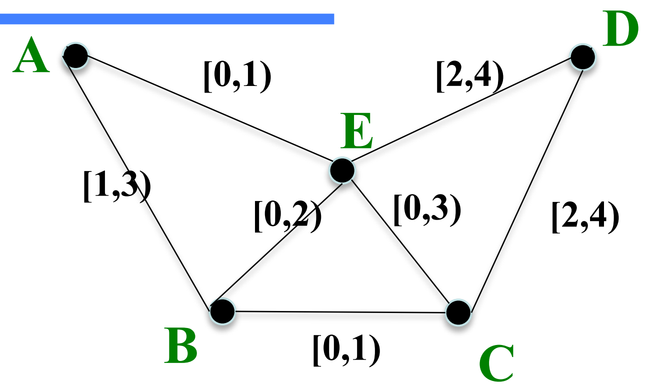
Journey: moving on a TVG



Indirect journey Waiting (buffering)

Direct journey No waiting (no buffering)

Journey: moving on a TVG

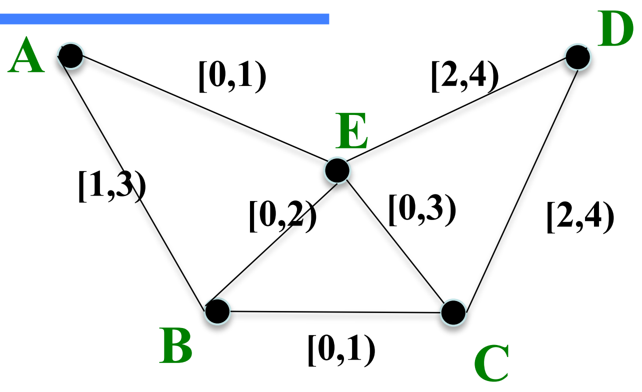


Journey [A,E,D] at time t=0

Indirect: **feasible**

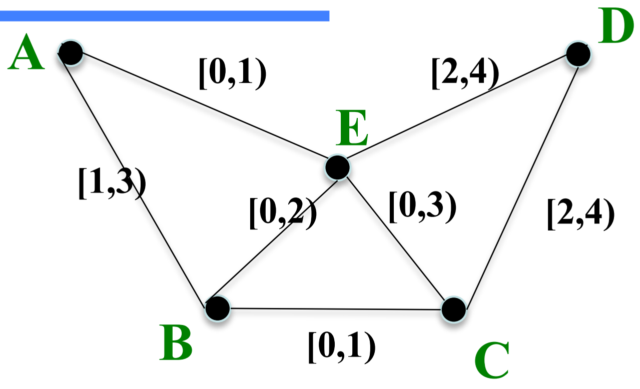
Direct: **unfeasible**

Journey: moving on a TVG



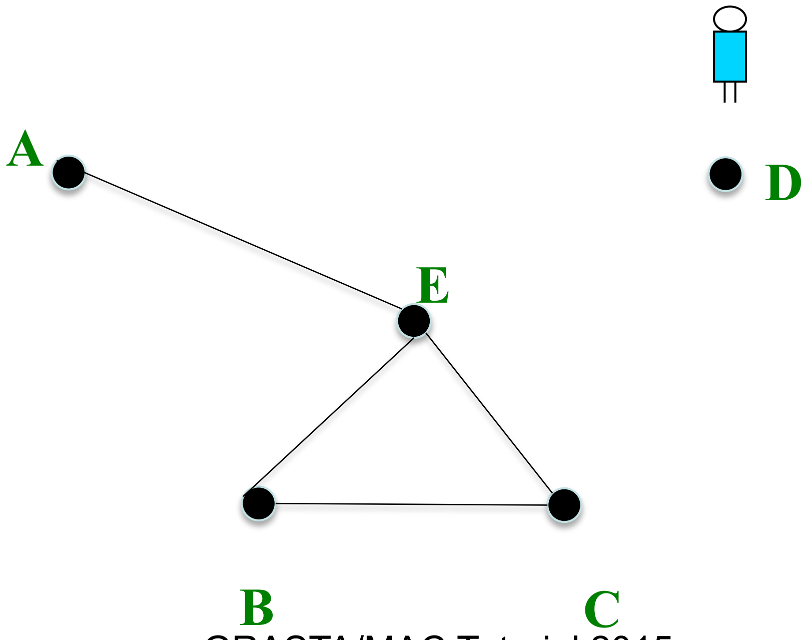
Even with waiting, the journey might be no feasible

Journey: moving on a TVG

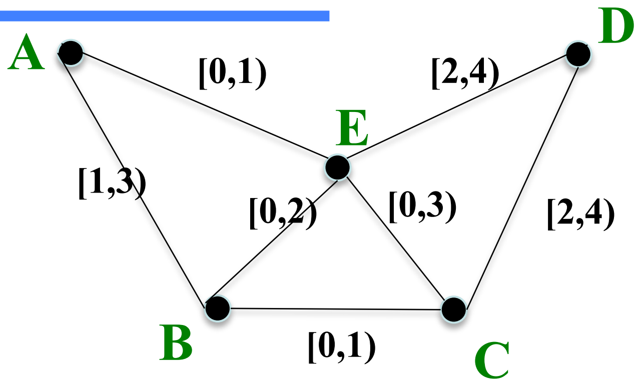


Task: go to B

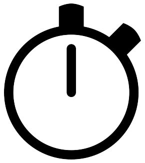
t=0



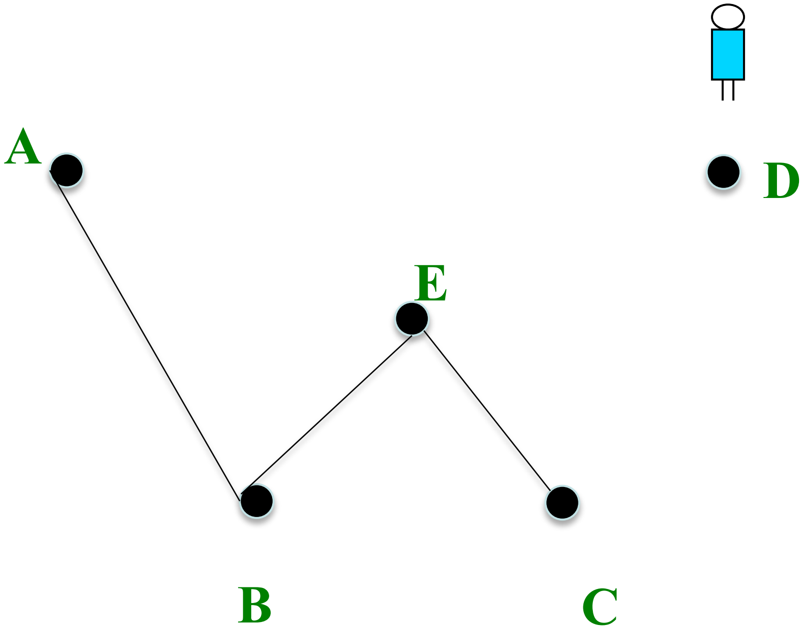
Journey: moving on a TVG



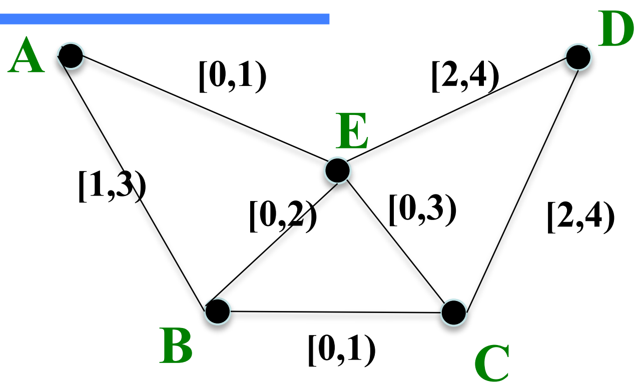
Task: go to B



t=1

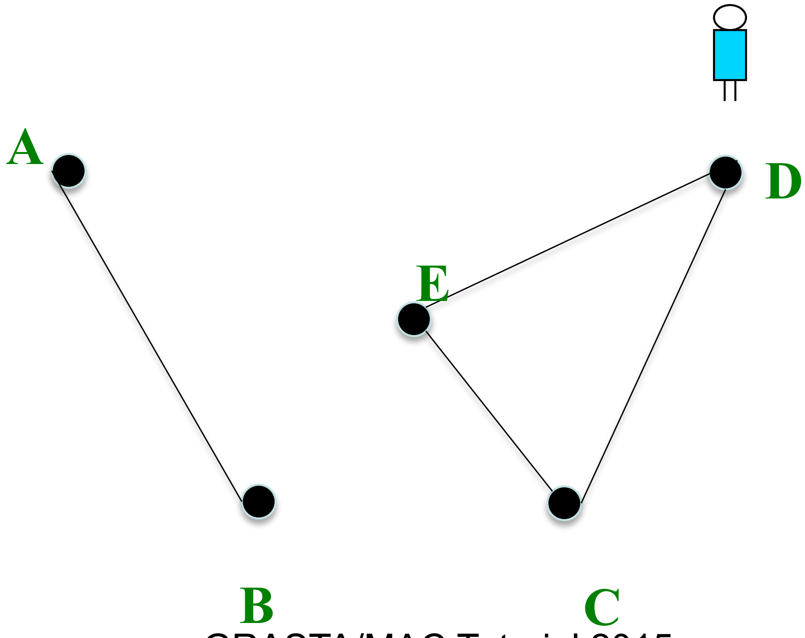


Journey: moving on a TVG

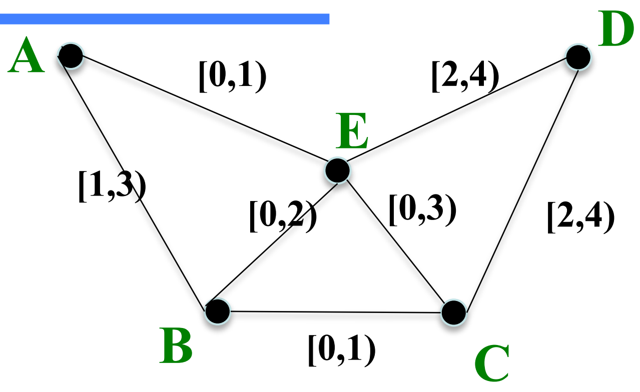


Task: go to B

t=2



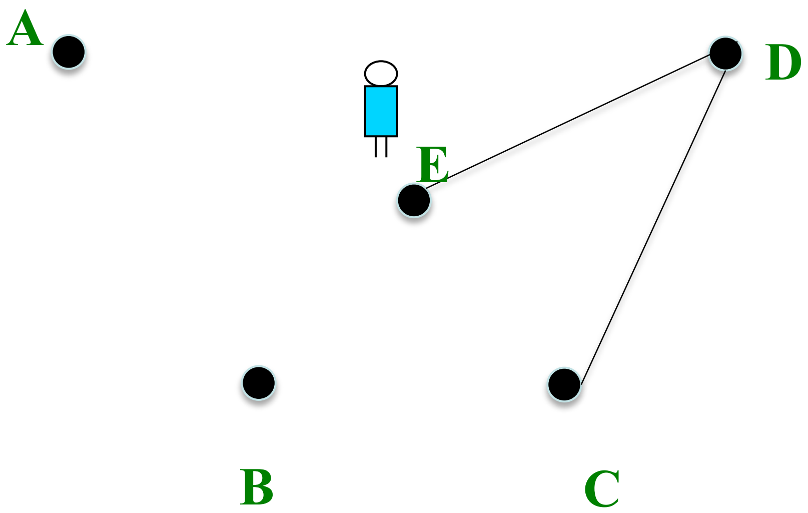
Journey: moving on a TVG



Task: go to B

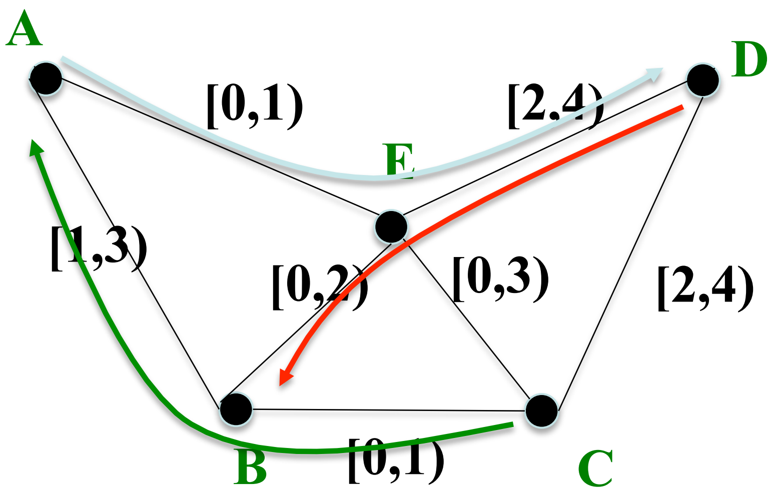
t=3

the agent is stuck and cannot reach B



Journey: temporal path

Starting at time 0, assuming it takes 1 time unit to traverse an edge:



C - - - > B - - - > A   Feasible

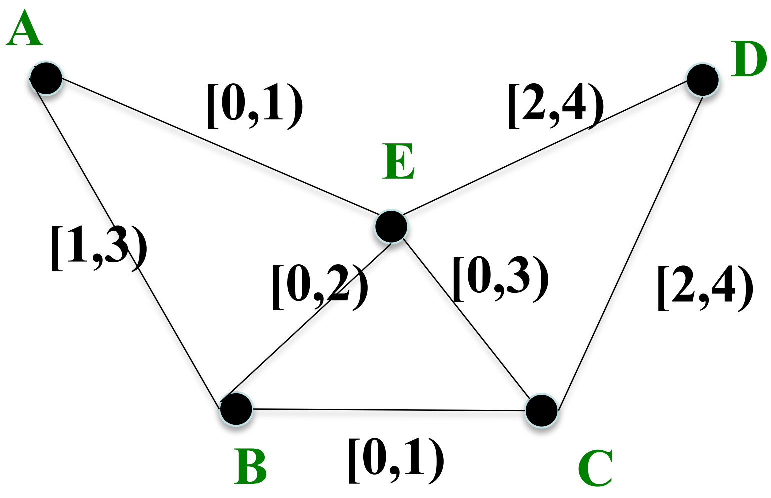
A - - - > E - - - > D   Feasible if waiting

D - - - > E - - - > B   NOT feasible at all at any time

DIRECT JOURNEY

INDIRECT JOURNEY

Journey: expressivity

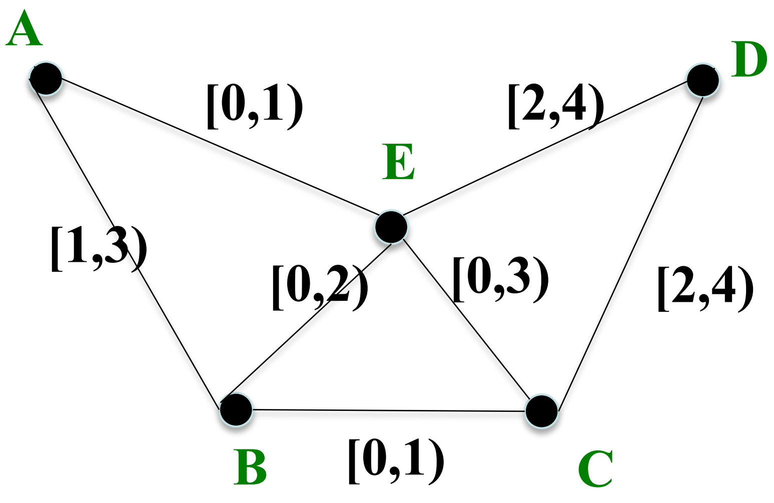


Interpreting a TVG as an automaton and node labels as alphabet symbols

a feasible journey corresponds to a string of symbols

C → B → A	Feasible	CBA
A → E → D	Feasible if waiting	AED

Journey: expressivity



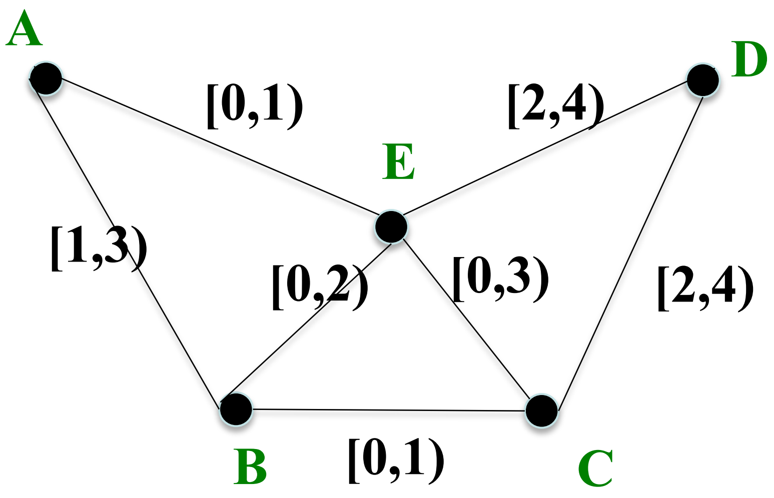
Interpreting a TVG as an automaton and node labels as alphabet symbols

a feasible journey corresponds to a string of symbols

what LANGUAGES of feasible journeys does the TVG-automaton recognize ?

C - - - > B - - - > A	**Feasible**	**CBA**
A - - - > E - - - > D	**Feasible if waiting**	**AED**

Journey: expressivity



Interpreting a TVG as an automaton and node labels as alphabet symbols

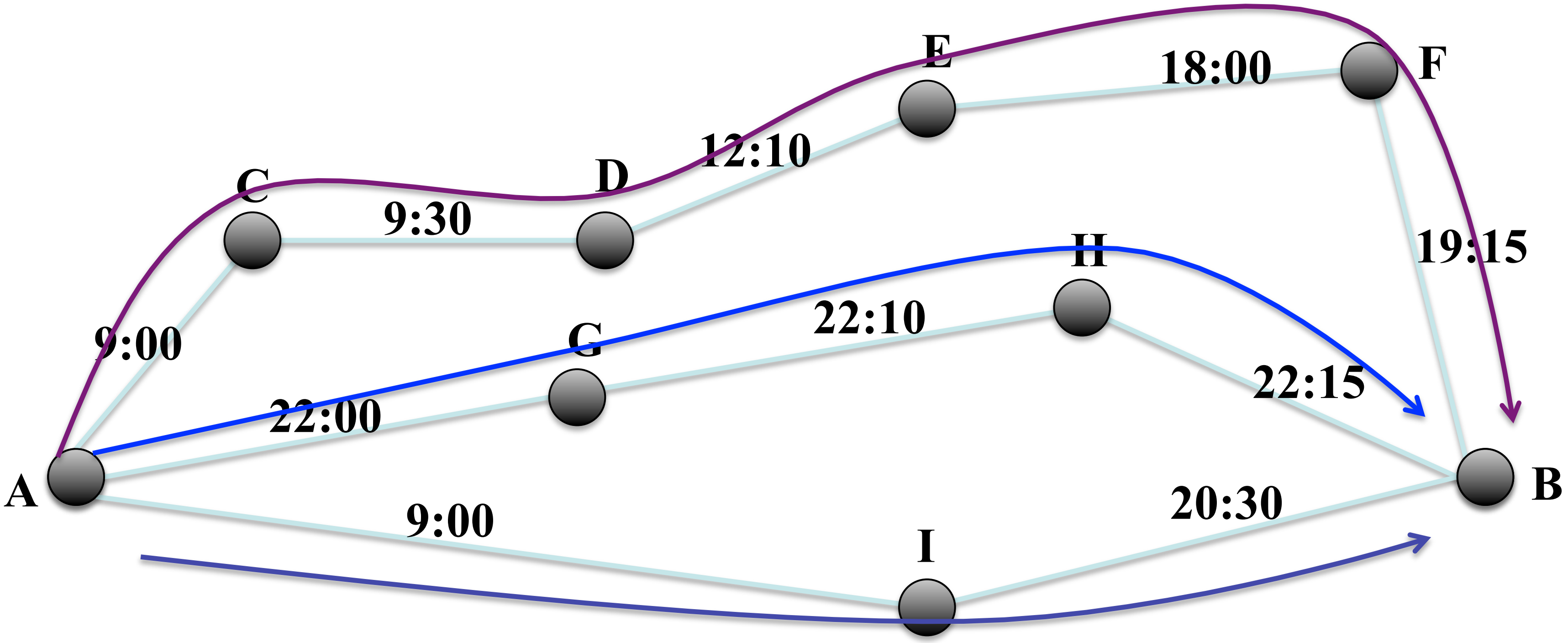
what LANGUAGES of feasible journeys does the TVG-automaton recognize ?

If **waiting IS NOT allowed**
TVG-automata recognize **ALL** languages

If **waiting IS allowed**
TVG-automata recognize only **REGULAR** languages

A. Casteigts, P.Flocchini, E. Godard, N. Santoro, M. Yamashita.
On the Expressivity of Time-Varying Graphs. *Theoretical Computer Science*, 2015

Journey: length



shortest	A,I,B	min # hops	(2 hops)
foremost	A,C,D,E,F	earliest arrival	(19:15)
fastest	A,G,H,B	smallest duration	(15 minutes)

Network metrics

Classical Measures

Distance
Diameter
Eccentricity
Centrality (betweeness, closeness)
Clustering Coefficient
…

Temporal Network metrics

Temporal Measures

Temporal Distance
Temporal Diameter
Temporal Eccentricity
Temporal Centrality (betweeness, closeness)
Temporal Clustering Coefficient
…

Keeping into account the temporal dimension of the network

Mobile Agents in Time-Varying Graphs

EXPLORATION

- Avin, C., Koucký, M., Lotker, Z.: How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). (*ICALP* 2008).

- D. Ilcinkas, A.M.Wade. On the Power of Waiting when Exploring Public Transportation Systems. (*OPODIS* 2011)

- P. Flocchini, M. Kellett, P.C. Mason, N. Santoro. Searching for Black Holes in Subways. *Theory of Computing Systems*, 2012.

- P. Flocchini, B. Mans, N. Santoro. On the exploration of time-varying networks. *Theoretical Computer Science,* 2013.

- D. Ilcinkas, A.M.Wade Exploration of the *T* -Interval-Connected Dynamic Graphs: The Case of the Ring. (*SIROCCO* 2013).

Mobile Agents in Time-Varying Graphs

EXPLORATION

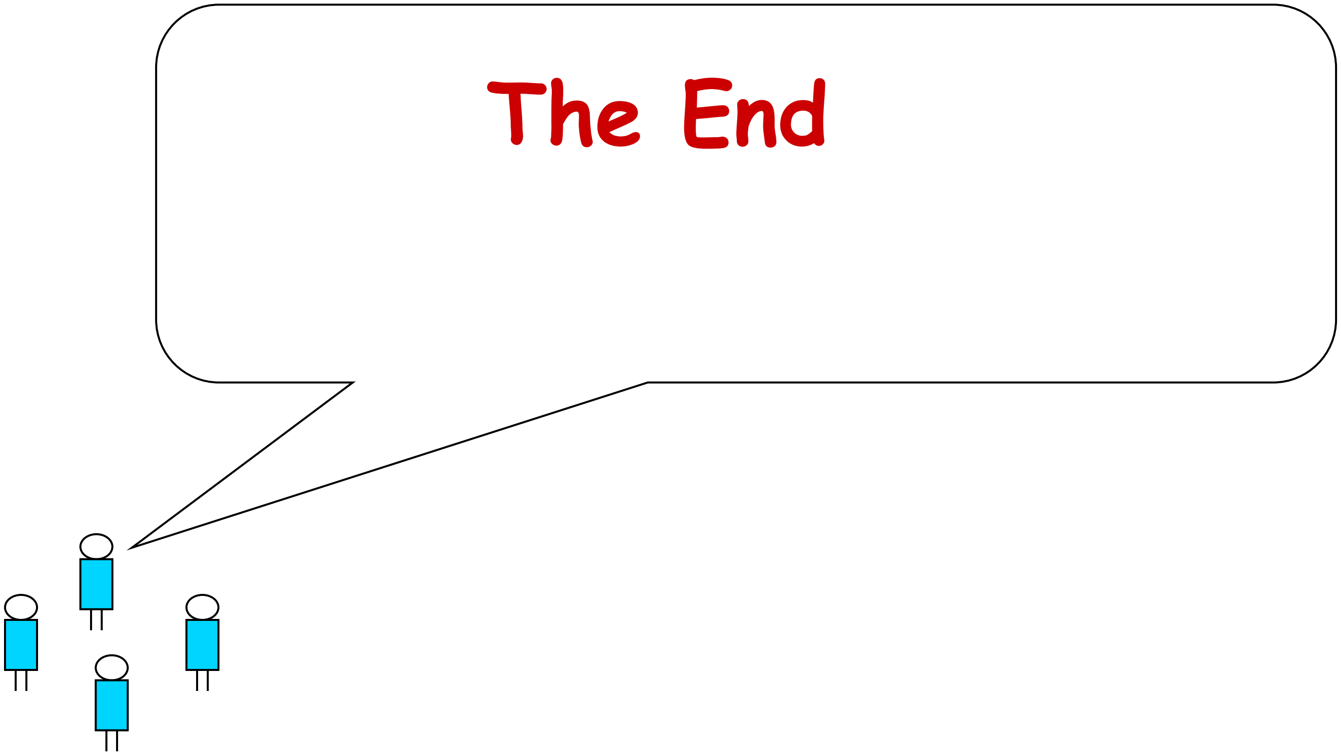
- P. Flocchini, M. Kellett, P.C. Mason, N. Santoro.
  Mapping an unfriendly subway system. (*FUN* 2014)

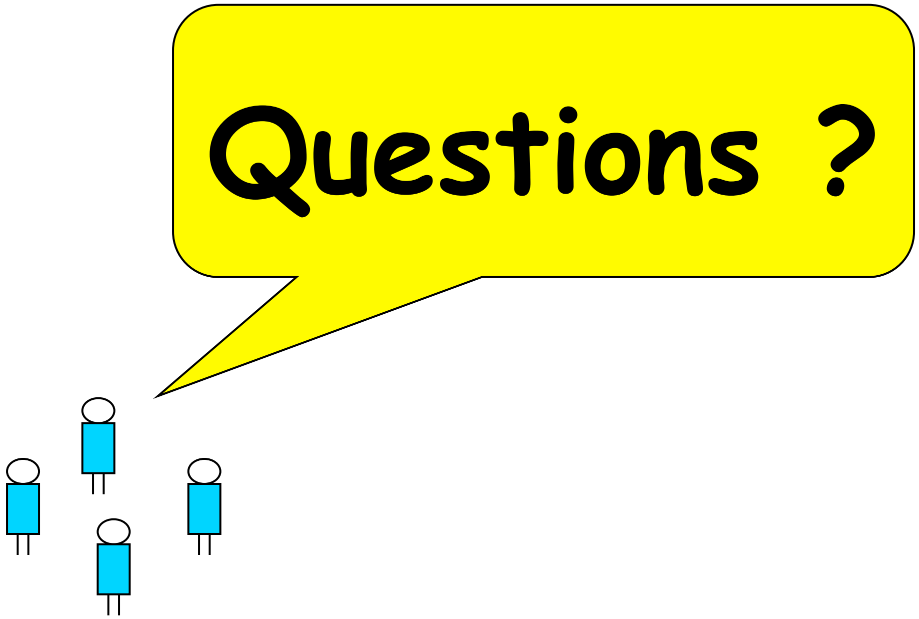
- D. Ilcinkas, R. K;asing, A.M.Wade. Exploration of constantly connected dynamic graphs based on cactuses. (*SIROCCO* 2014).

- E. Aaron, D. Krizanc, E. Meyerson. DMVP: Foremost waypoint coverage of Time-Varying Graphs, (*WG* 2014).

- T. Erlebach, M. Hoffmann, F. Kammer
  On Temporal Graph Exploration. (*arXiv* 2015).

- G.A. Di Luna, S. Dobrev, P. Flocchini, N. Santoro.
  Exploring 1-interval-connected rings. (*arXiv* 2015).


The End



Questions ?

Thank you