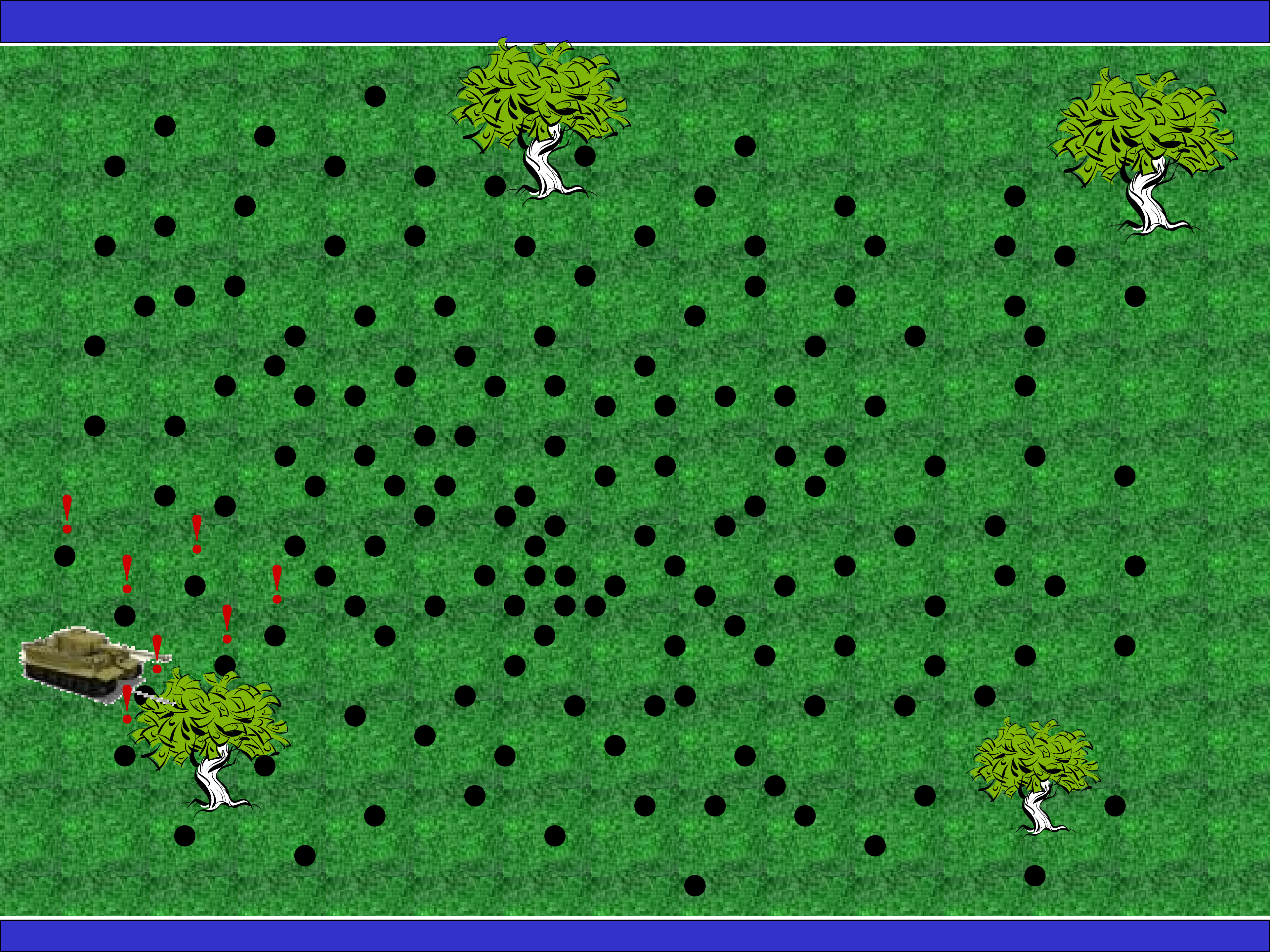
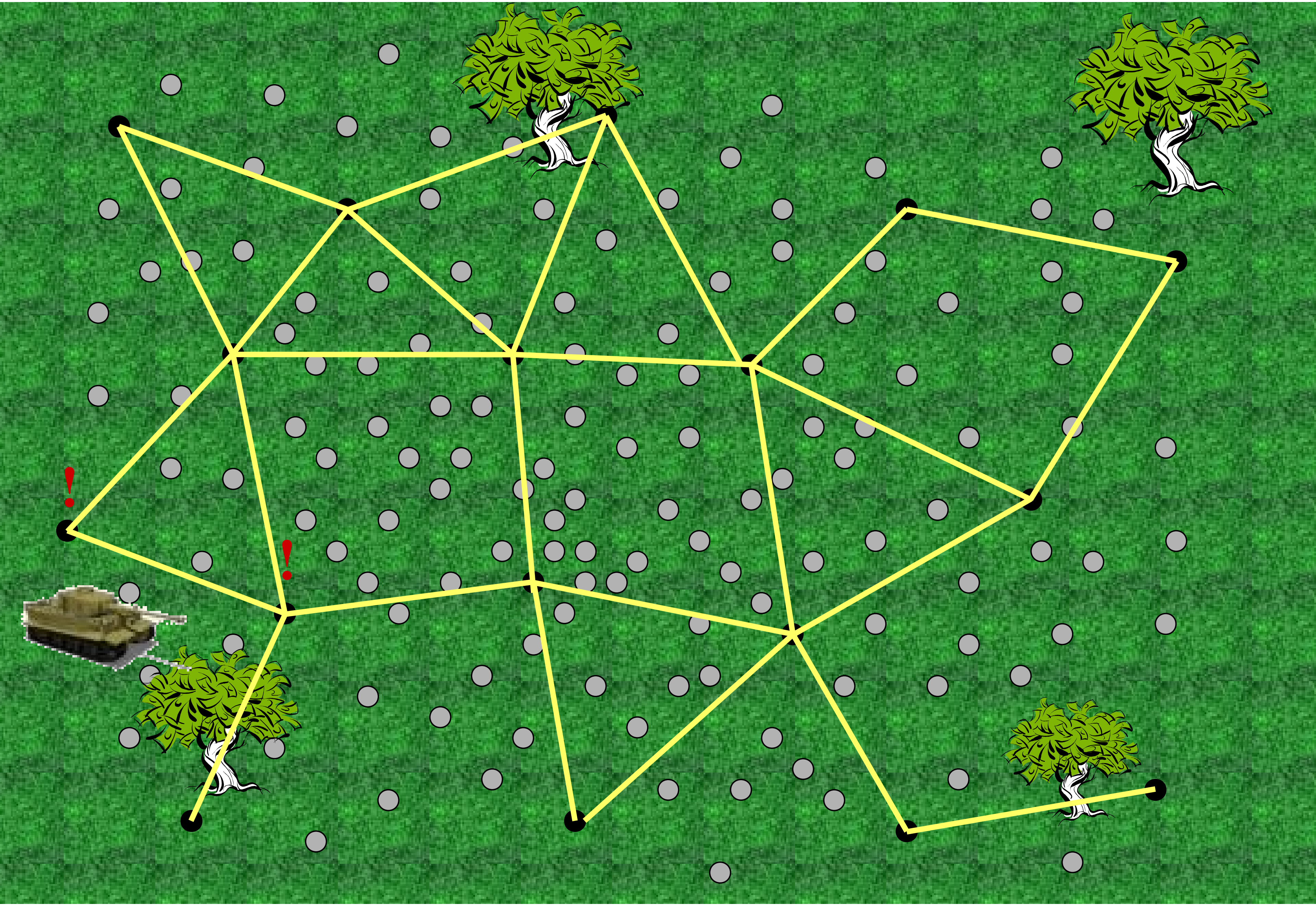
Connected Infrastructures





Self-stabilizing connected infrastructures

- Objective : Choose a set of nodes M such that

  - each node in the system is either in M or neighbor of a node in M (covering)

  - nodes in M can communicate with each other (conectivity)

Quality of service: self-organization and fault-tolerance

Model

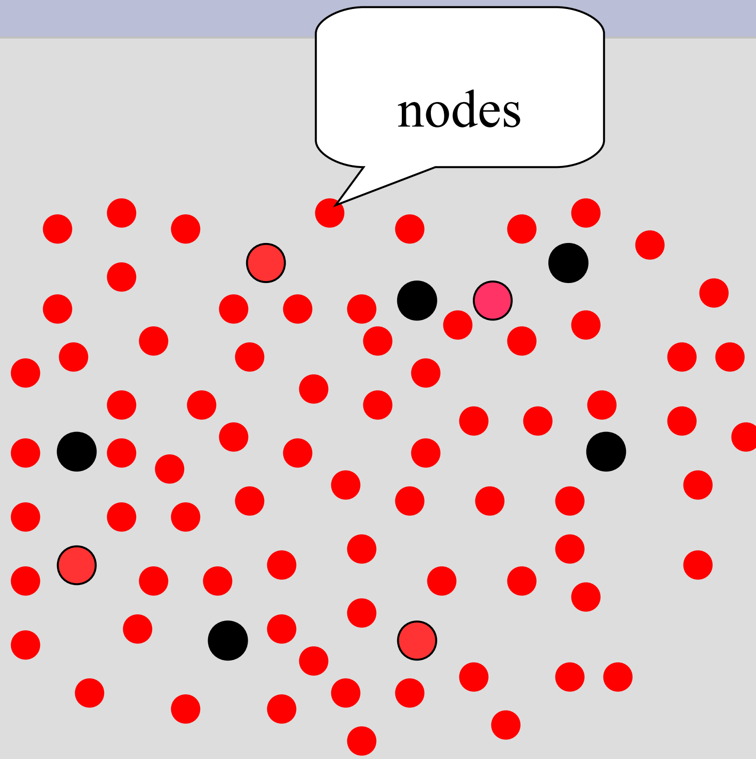
- Id uniques.
- Indicator (eg. bandwidth , energy level, storage level)

- Local communication

First Solution :
Maximal Independent Set

Network

- « Passive »
- « Active »



Node i

Rule 1:

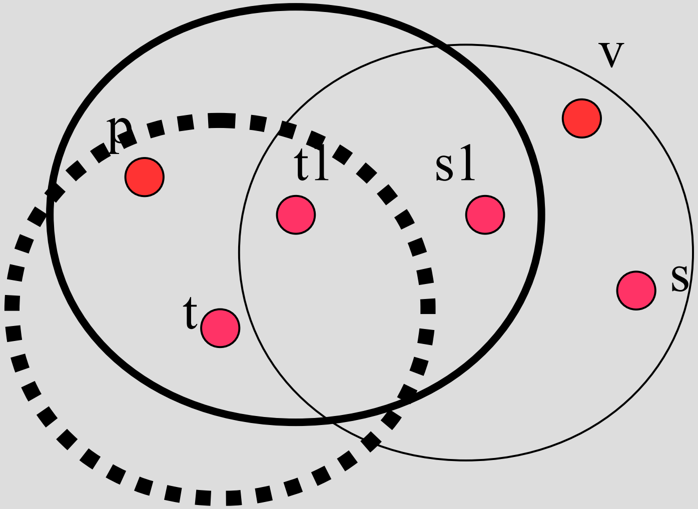
- Passive and Candidate(i) ⟹
  Change to Active

Rule 2:

- Active and (not Candidate(i)) ⟹
  Change to Passive

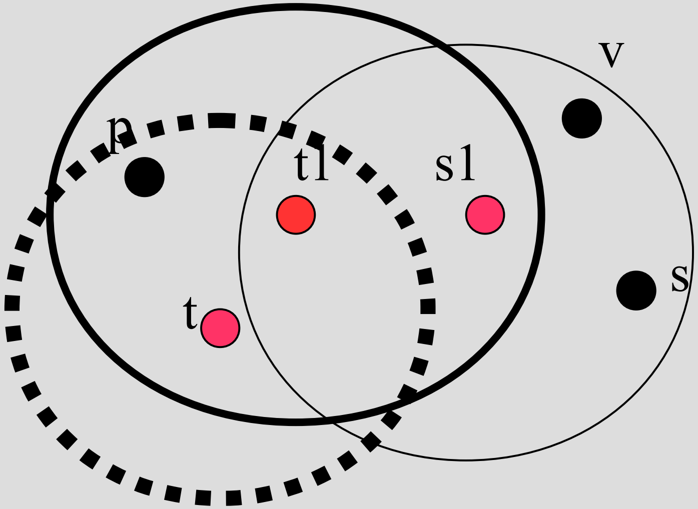
- Candidate(i) iff i has no neighbor j Active or i has the best indicator in its neighborhood

MIS execution



p – execute Rule 1 (candidate not active)

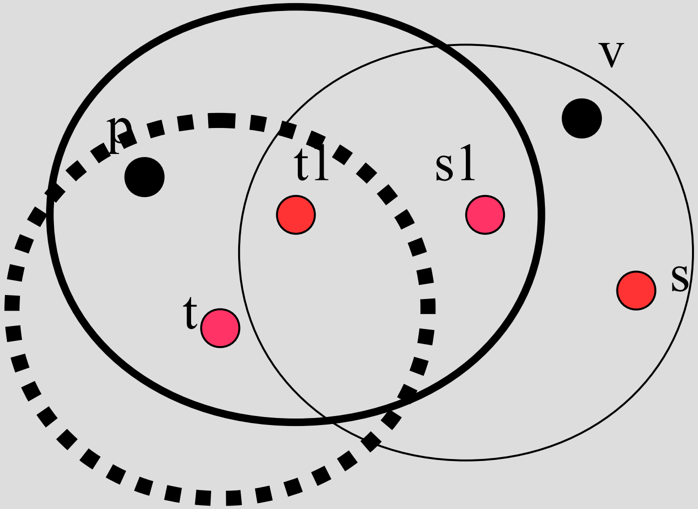
MIS execution



p – executed Rule 1

v et s – execute Rule 1

MIS execution

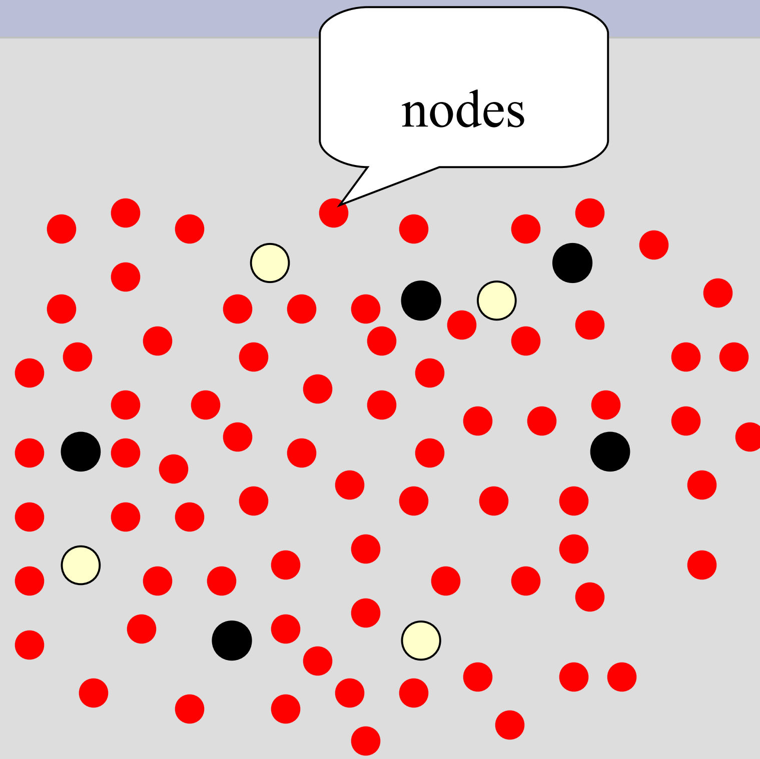


p – executed Rule 1

s – execute Rule 2 (v has a stronger indentifier)

Network

- « Passive»
- « Active»
- « Bridge »



CDS : Node i

Step 1:

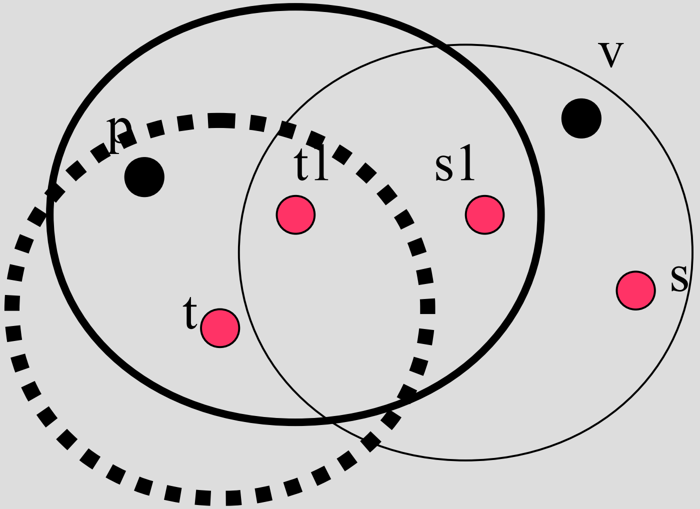
- Passive and CandidateBridge(i) and (not Covered(i)) ⇒ change to Bridge

Step 2:

- Bridge and (not CandidateBridge(i) or Covered(i)) ⇒ change to Passive

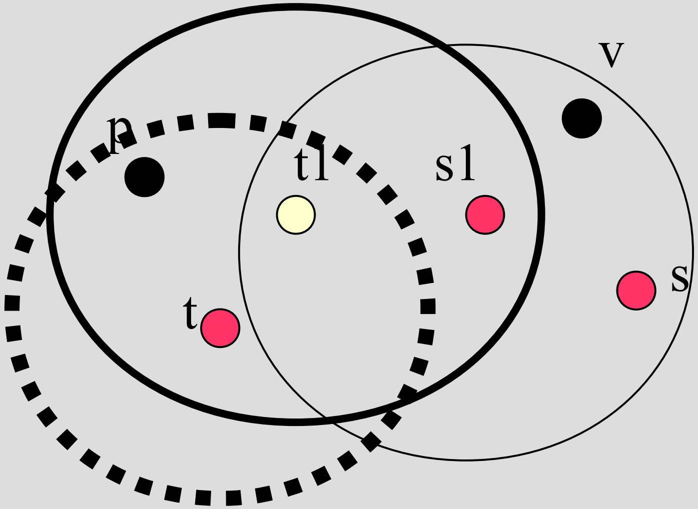
- CandidateBridge(i) iff i has a neighbor j (Active) and the neighborhood of i is not included in those of j
- Covered(i) iff i has a neighbor j such that
  - neighborhood of i is included in the neighborhood of j or
  - i and j have the same neighbors and j has a better indicator

CDS execution



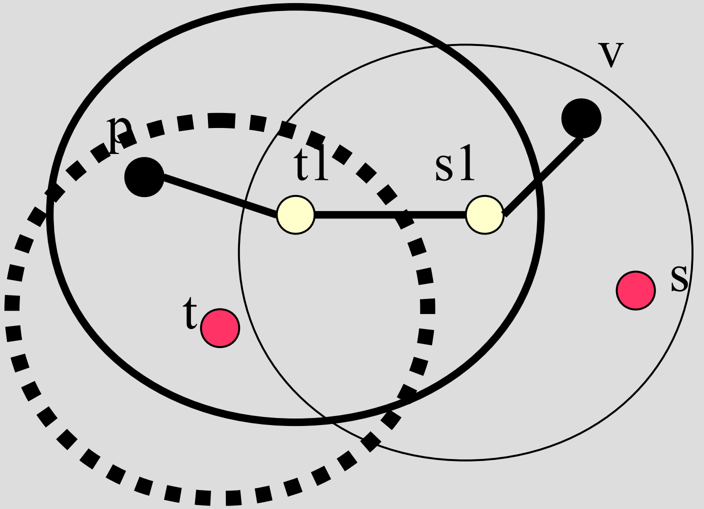
t1 – execute Rule 1 (candidate « bridge» and not covered by an « active » node)

CDS execution



t1 – execute Rule 1
t – stays « passive »
s1 – can execute Rule 1

CDS execution

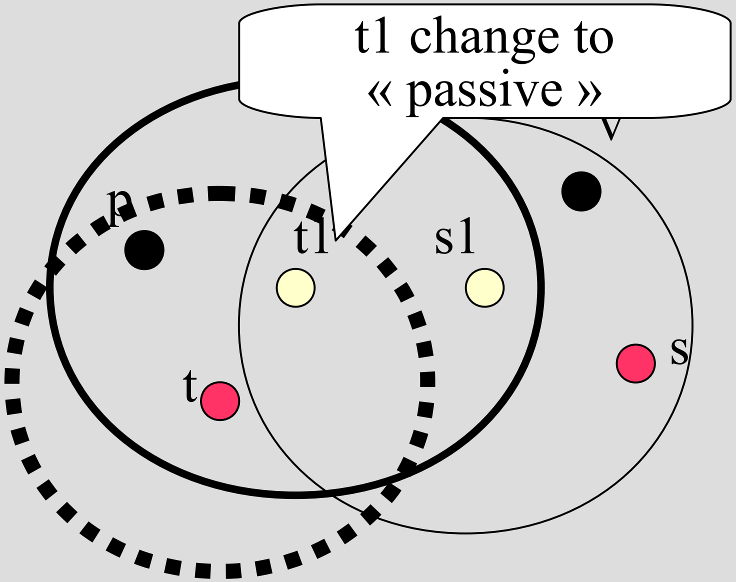


t1 - executed Rule 1
t - stays « passive»
s1 - executed Rule 1
s - stays « passive »

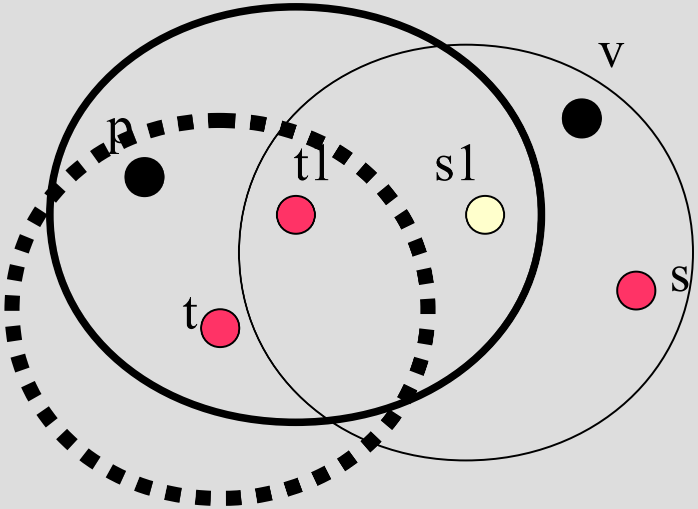
Faults

- Wrong initialisation
- Corruptions of nodes memory
- Faulty nodes and communication links

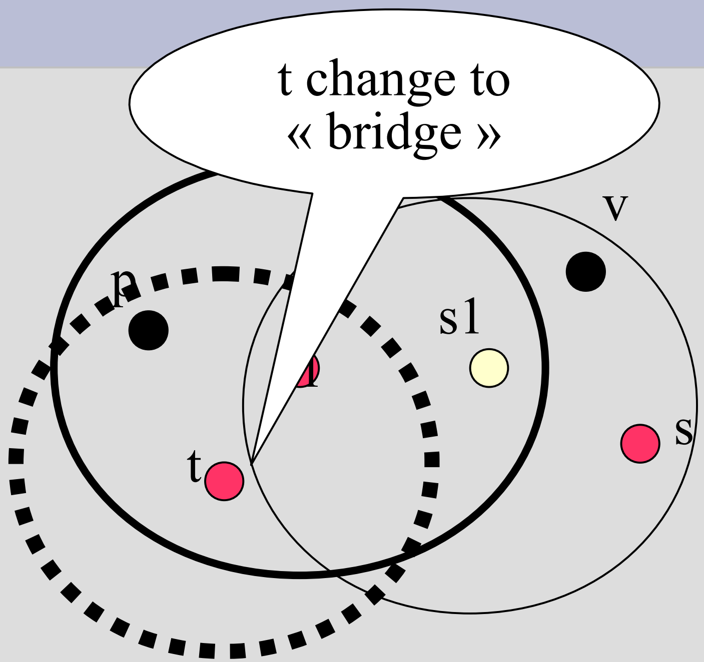
Correction of faults



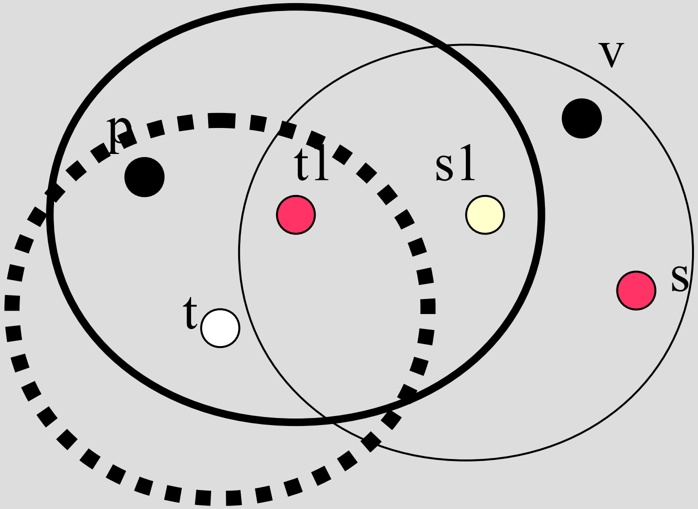
Correction of faults



Correction of faults

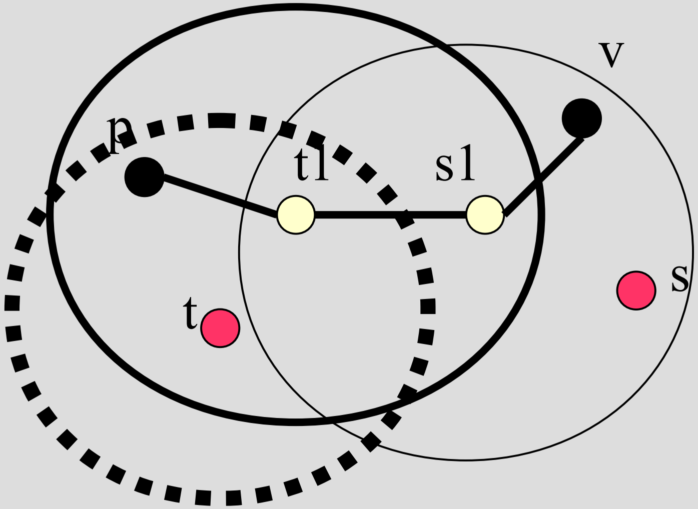


Correction of faults



- t execute Rule 2 because it is covered by t1 and hence corrects its state
- t1 execute Rule 1 because it is a bridge and hence corrects its state

Stable state



Algorithm Complexity

- States : 3 (2 bits)
- Time Complexity: O(f(n)+n) where O(f(n)) is the complexity of a MIS algorithm

Second solution : Dominating Sets

Node i

Step 1:

- Passive and IndependentNeighbors(i) and not Dominated(i) ⇒ Active

Step 2:

- Active and (exists neighbor j, j Active and Dominated(i) per j) ⇒ Passive

Step 3:

- Passive and the same neighborhood as its neighbors and MaxIndicator(i) ⇒ Active

- IndependantNeighbors(i) iff i has two neighbors who are not mutually neighbors
- Dominated(i) per j iff
  - Neigborhood of i is included in the neighborhood of j or
  - i et j have the same neighbors and j has a better indicator
- MaxIndicator(i) iff i has the maximal indicator in his neighborhood

Algorithm complexity

- State Complexity : 2 (1 bit)
- Time Complexity: n steps