SHARC'16

Detection of Failed Boundary Nodes in Wireless Sensor Networks

Farid Lalem, Ahcène Bounceur, Rahim Kacimi, and Reinhardt Euler

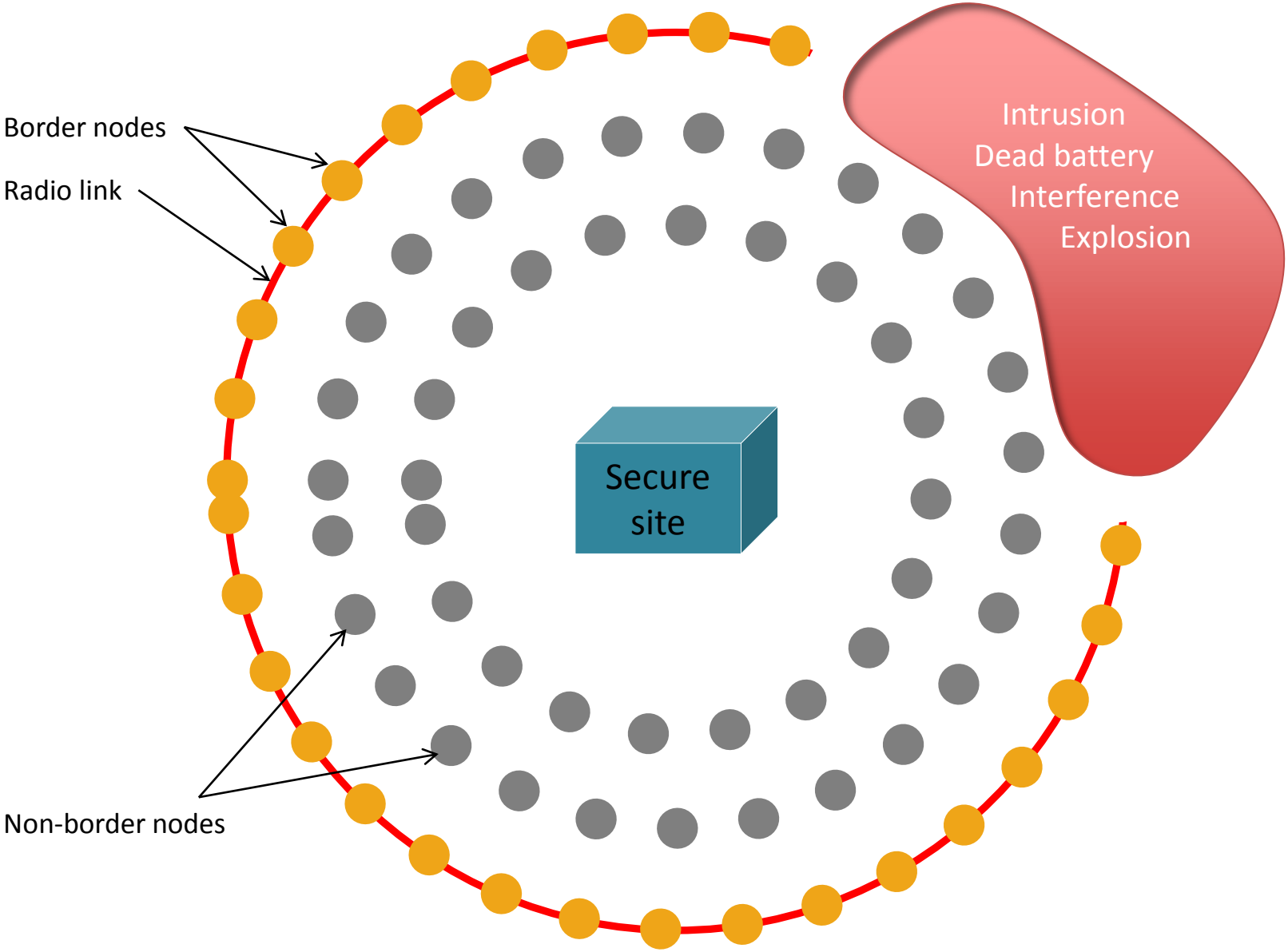
juin, 2016

cnrs Lab-STICC
UNIVERSITÉ TOULOUSE III PAUL SABATIER
IRIT Institut de Recherche en Informatique de Toulouse

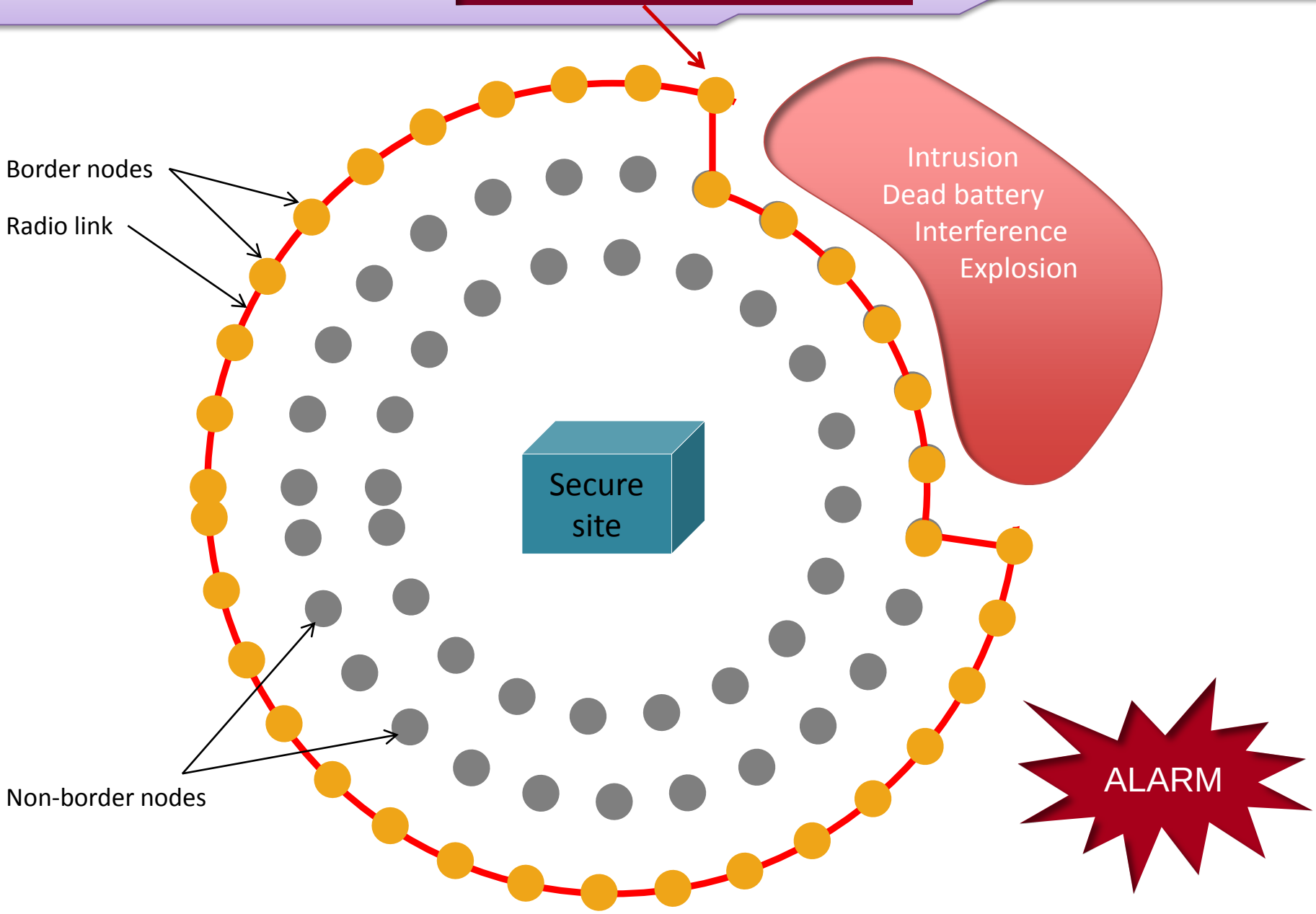
Outline

1. Introduction
2. Border Node determination algorithm
3. Border Node Failure Detection
4. Simulation results
5. Conclusion & Future Work

Introduction

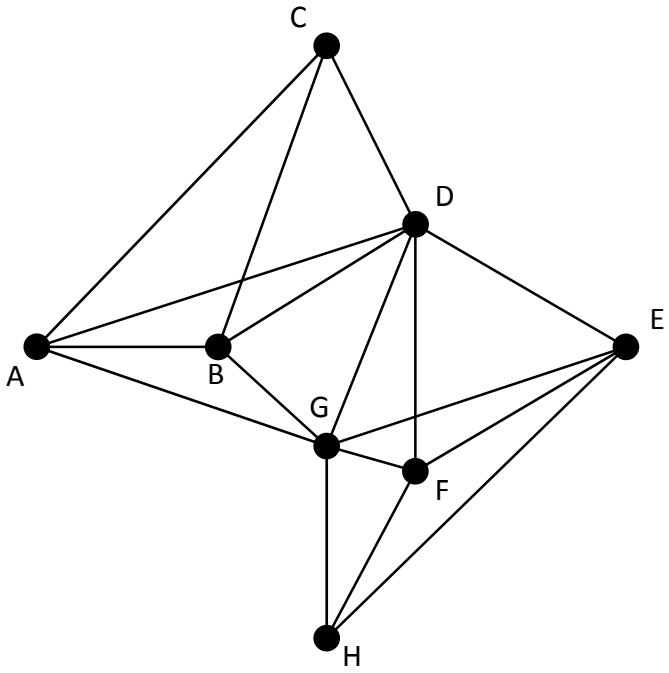


Introduction



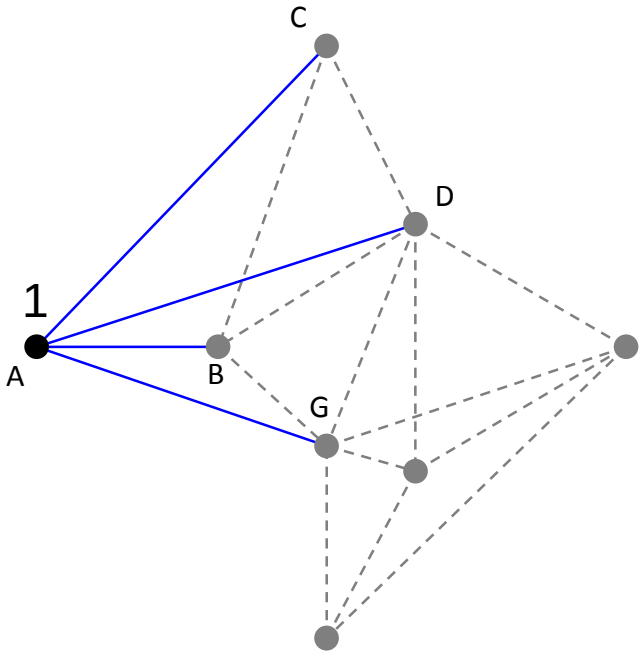
Border Node determination

2



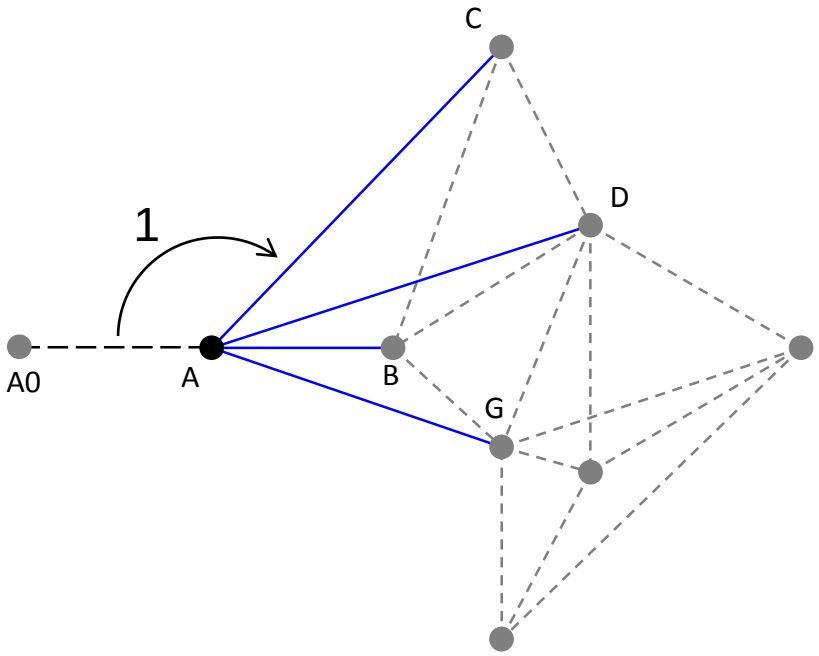
Farid.Lalem@univ-brest.fr

Border Node determination



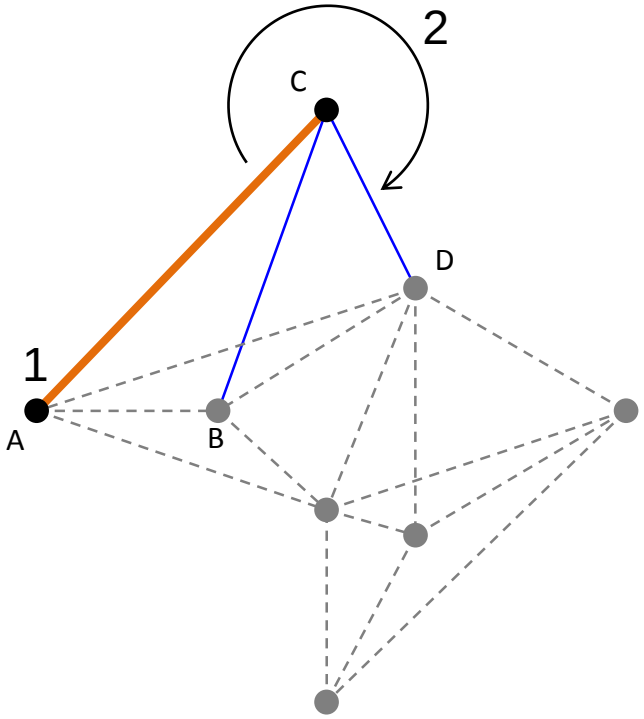
Farid.Lalem@univ-brest.fr

Border Node determination



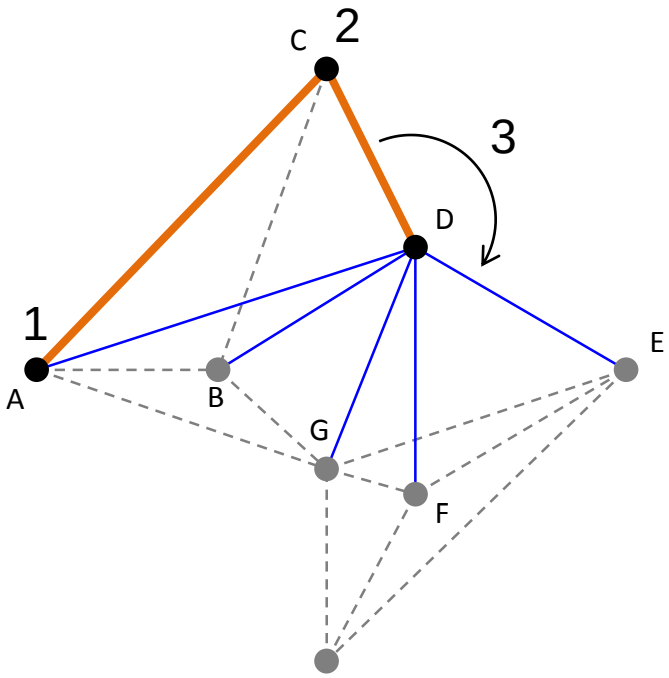
Farid.Lalem@univ-brest.fr

Border Node determination



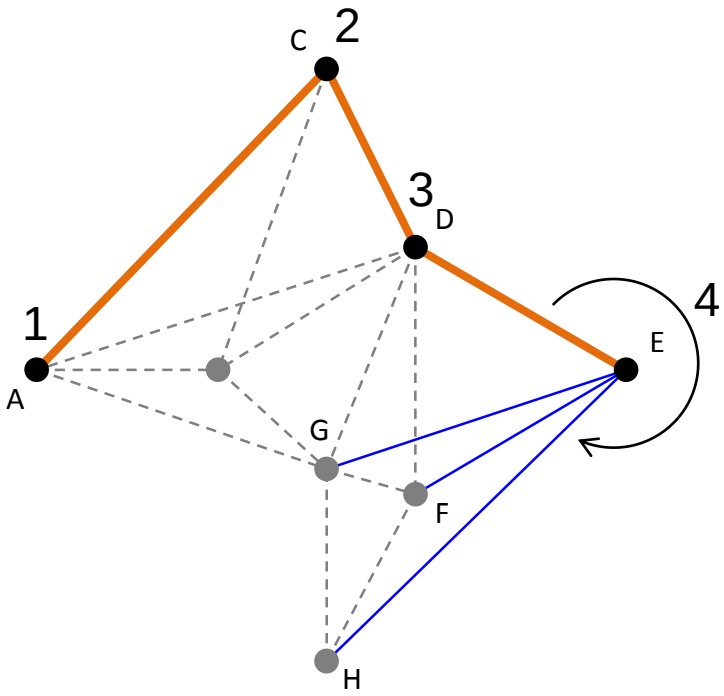
Farid.Lalem@univ-brest.fr

Border Node determination



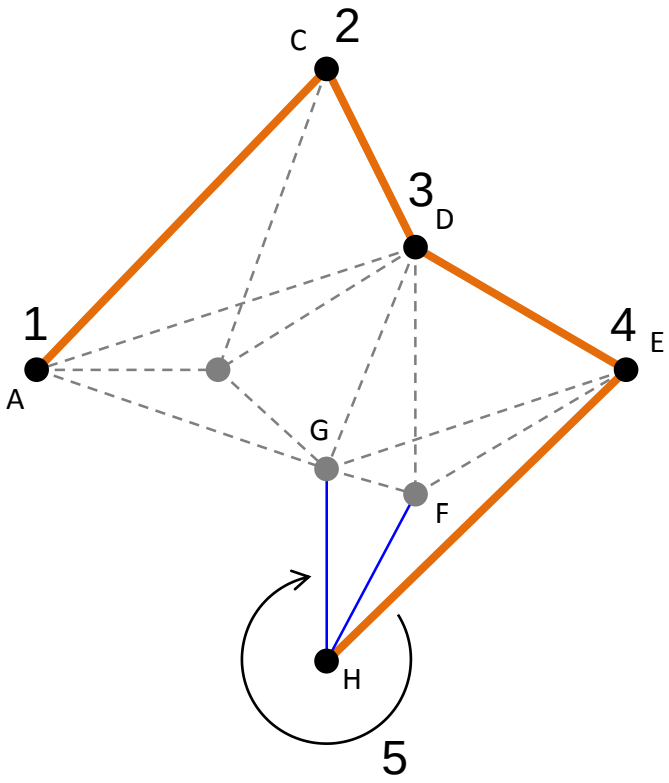
Farid.Lalem@univ-brest.fr

Border Node determination

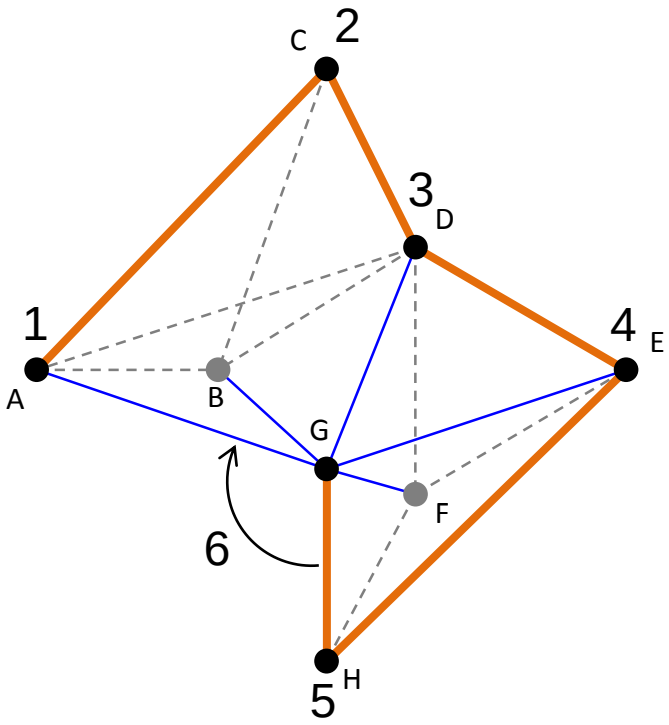


Farid.Lalem@univ-brest.fr

Border Node determination

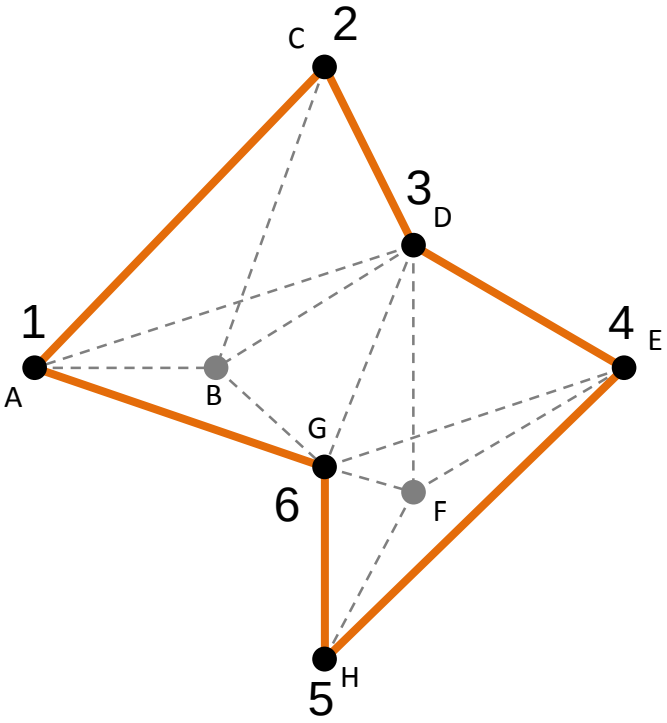


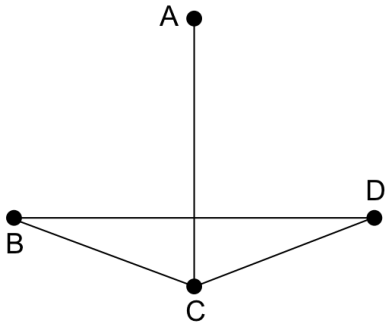
Border Node determination



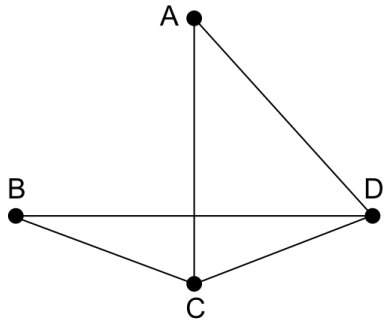
Farid.Lalem@univ-brest.fr

Border Node determination





Anchor graph



Boat graph

Farid.Lalem@univ-brest.fr

Border Node determination

2: $P_c \leftarrow$ point having the minimum x-coordinate
3: $\mathbb{B}_V \leftarrow [P_c]$
4: $P_f \leftarrow P_c$
5: $P_p \leftarrow$ fictive point situated in the left of P_f
6: **repeat**
7: $\mathbb{A} = \emptyset$
8: $P_v = \underset{P_j \in N(P_c) \ \& \ P_j \notin \mathbb{A}}{\text{argmin}} \{pa(P_p, P_c, P_j)\}$
9: **if** intersection detected **then**
10: $\mathbb{A} = \mathbb{A} \cup \{P_v\}$
11: Go to 8
12: **end if**
13: $\mathbb{B}_V \leftarrow [\mathbb{B}_V, P_v]$
14: $P_p \leftarrow P_c$
15: $P_c \leftarrow P_v$
16: **until** $P_v = P_f$
17: **return** $\mathbb{B}_V$
18: **end procedure**

Farid.Lalem@univ-brest.fr

Border Node determination

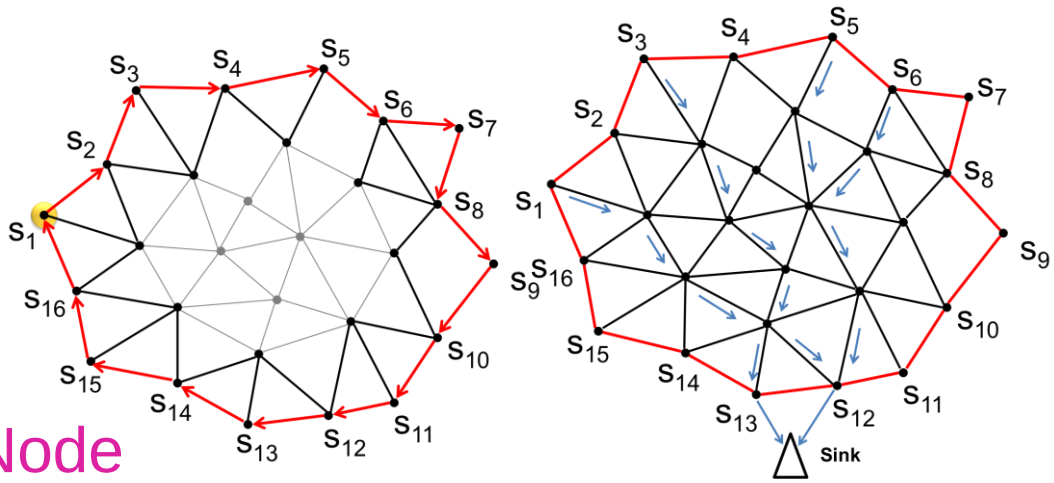
- Version 1
  - Complexity: **O(kh)**
  - k: Maximum degree of the graph
  - h: Number of the border nodes
- Version 2 [1]
  - Complexity: **$O(kh^2)$**
  - k: Maximum degree of the graph
  - h(h+1)/2: Number of the border nodes by considering the intersection detection

[1] Ahcène Bounceur, Reinhardt Euler, Ali Benzerbadj, Farid Lalem, Massinissa Saoudi, Tahar Kechadi, Marc Sevaux, Finding a Polygon Hull in Wireless Sensor Networks, Invited talk, EURO 2015, Glasgow.

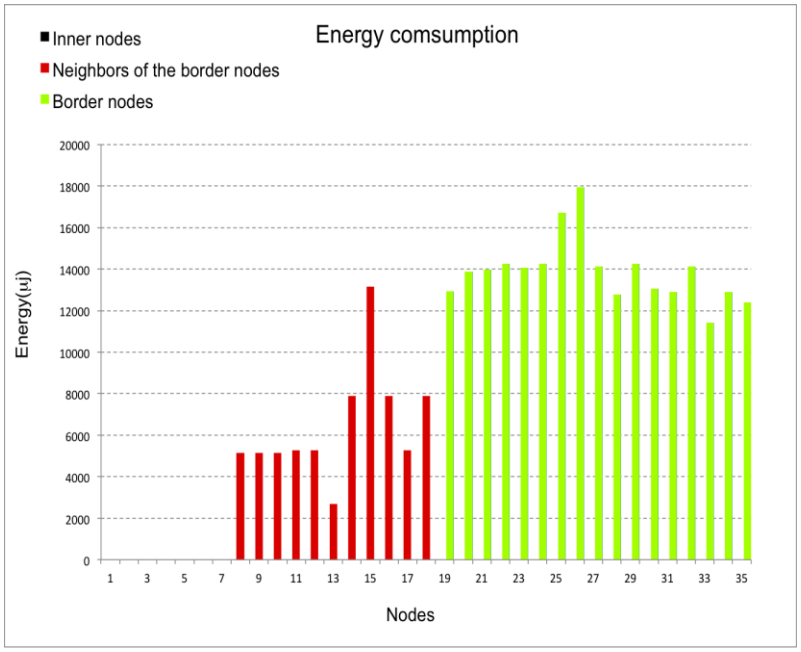
Farid.Lalem@univ-brest.fr

The distributed version

- 3 types of messages:
  - AC : Ask for Coordinates
  - CS : Coordinate Sending
  - SN : Select a Boundary Node



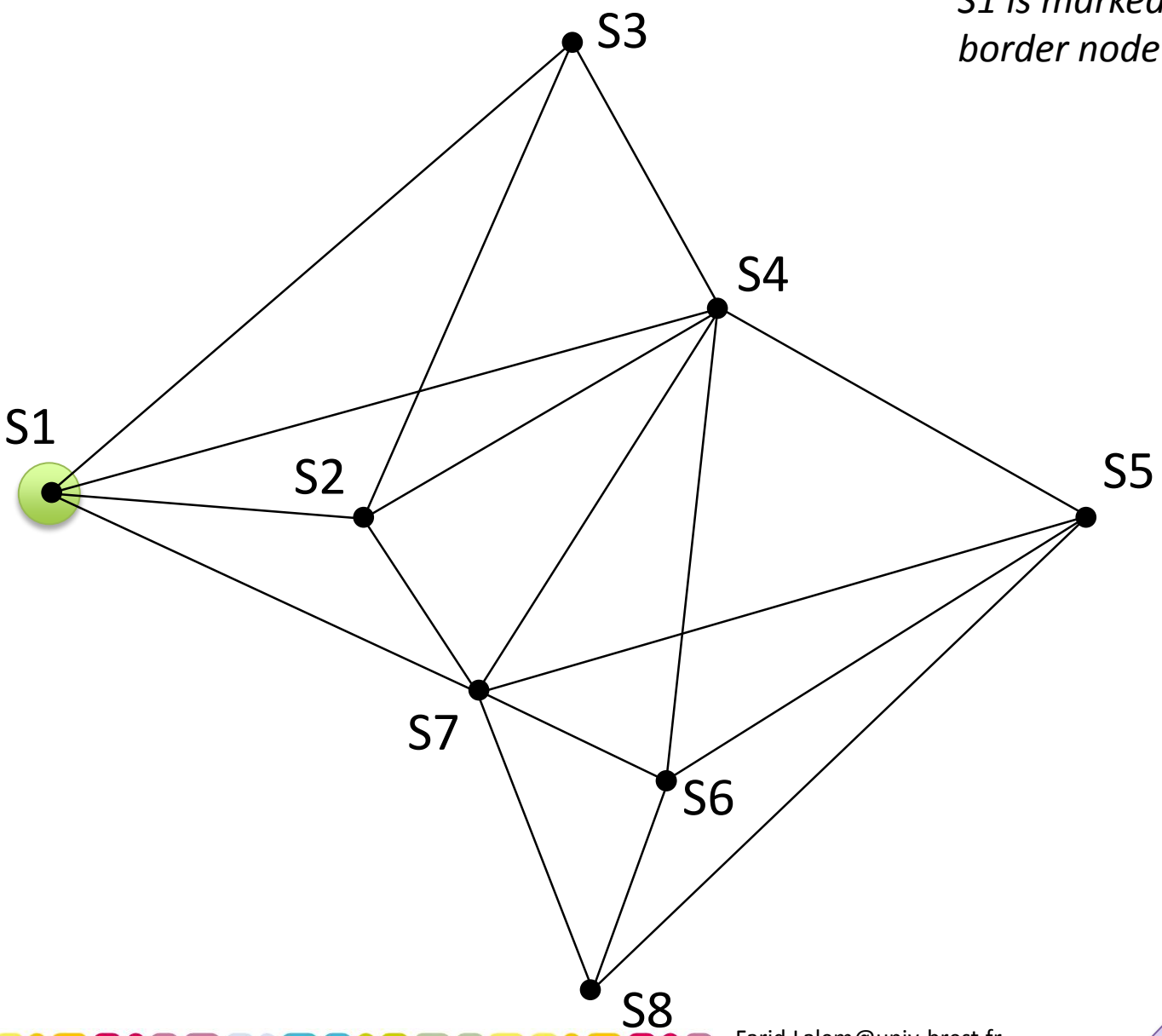
- Format:
  - Message AC : **id | AC**
  - Message CS : **id | CS | x,y**
  - Message SN : **id | SN | x,y**



Farid.Lalem@univ-brest.fr

The distributed version

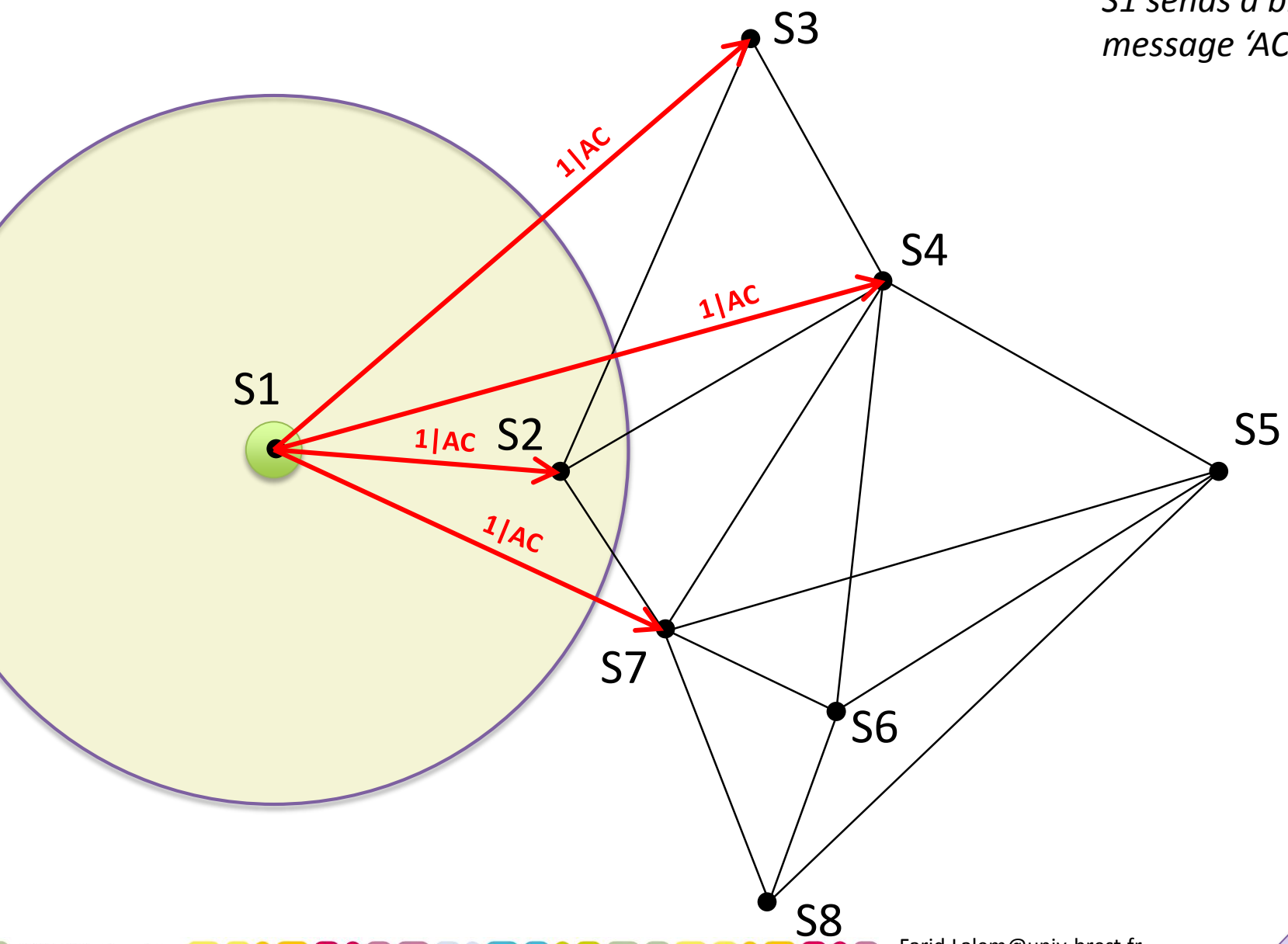
S1 is marked as a border node



Farid.Lalem@univ-brest.fr

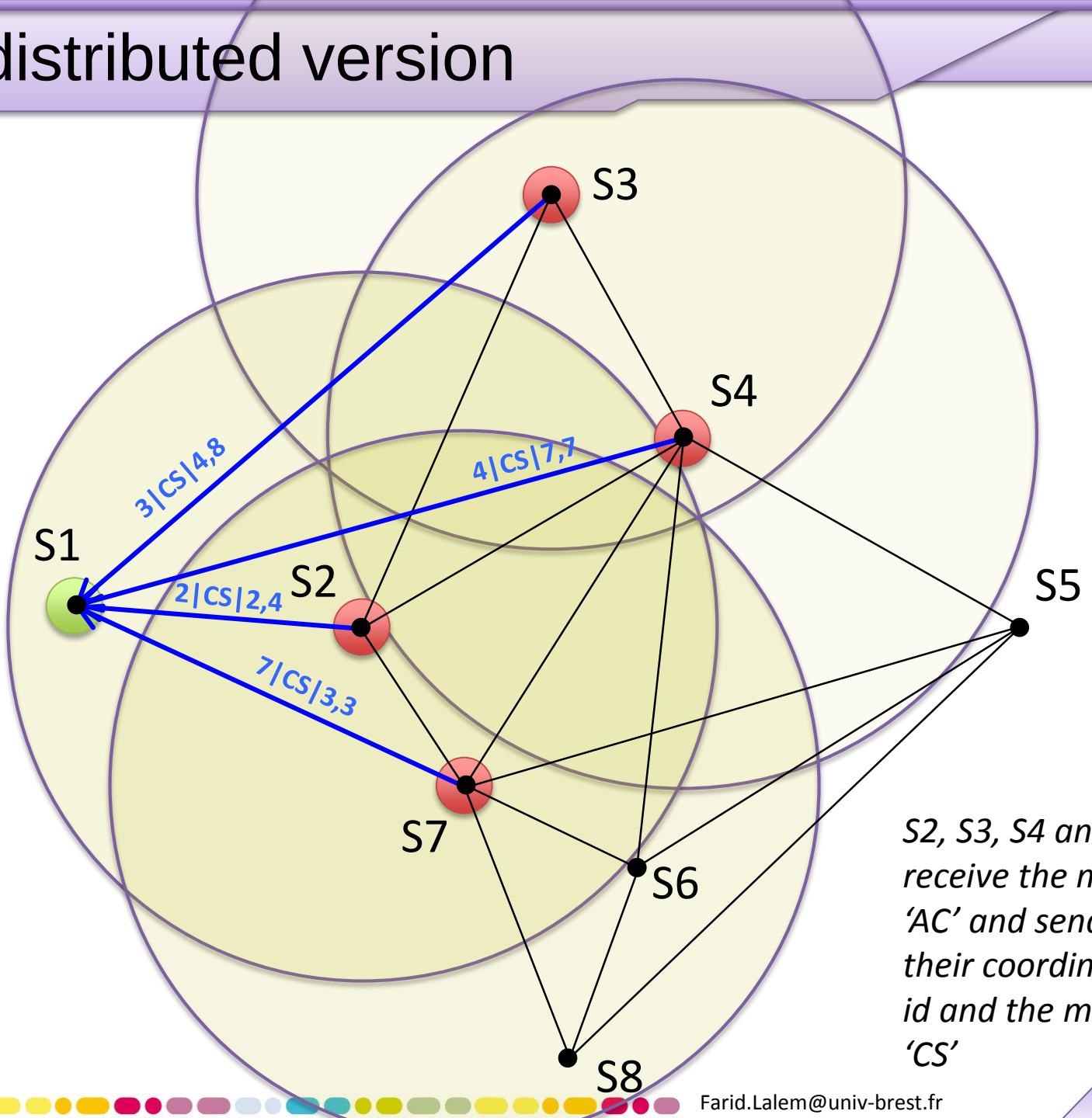
The distributed version

S1 sends a broadcast message 'AC' and its id



Farid.Lalem@univ-brest.fr

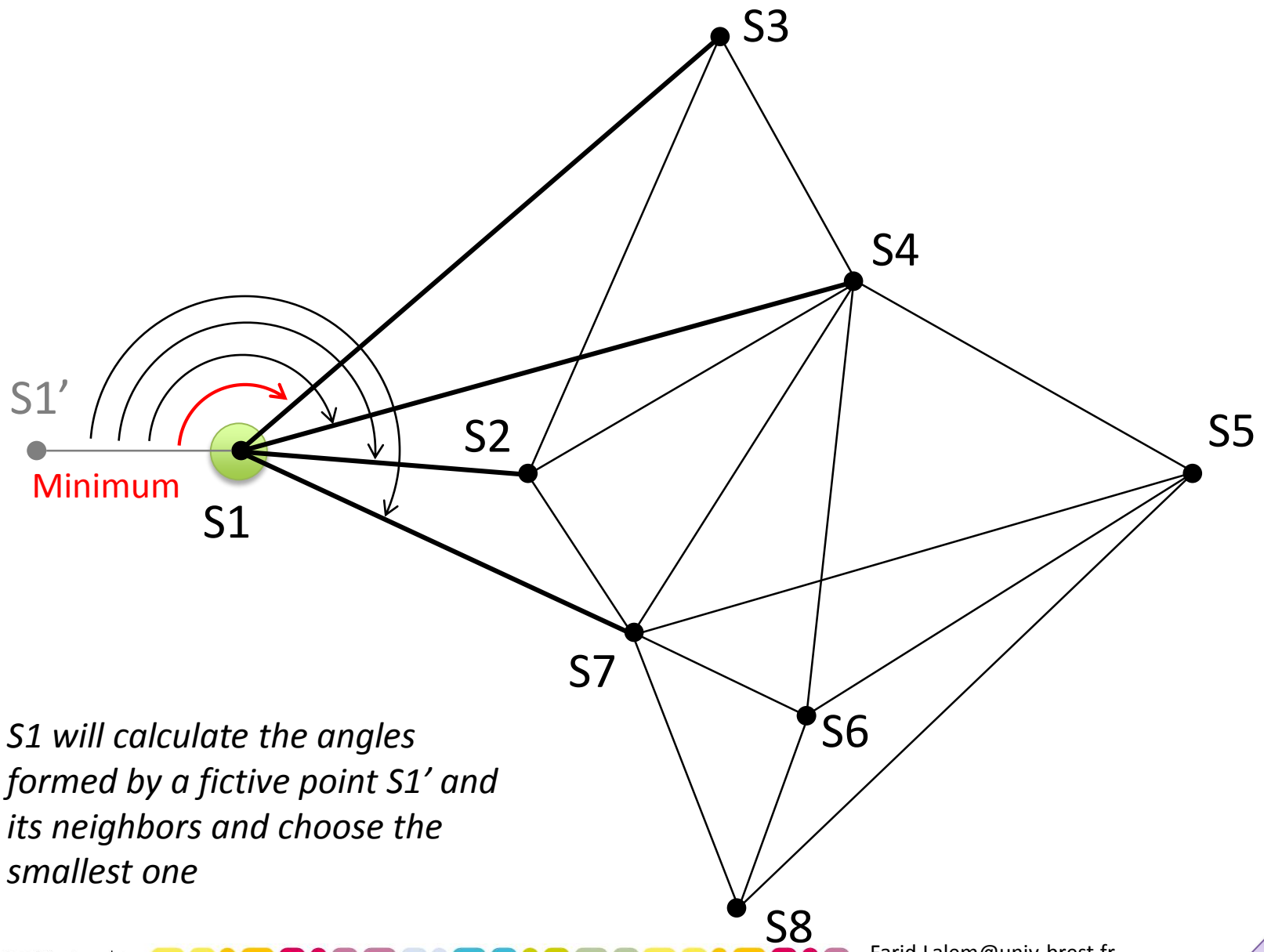
The distributed version



S2, S3, S4 and S7 receive the message 'AC' and send to S1 their coordinates, their id and the message 'CS'

Farid.Lalem@univ-brest.fr

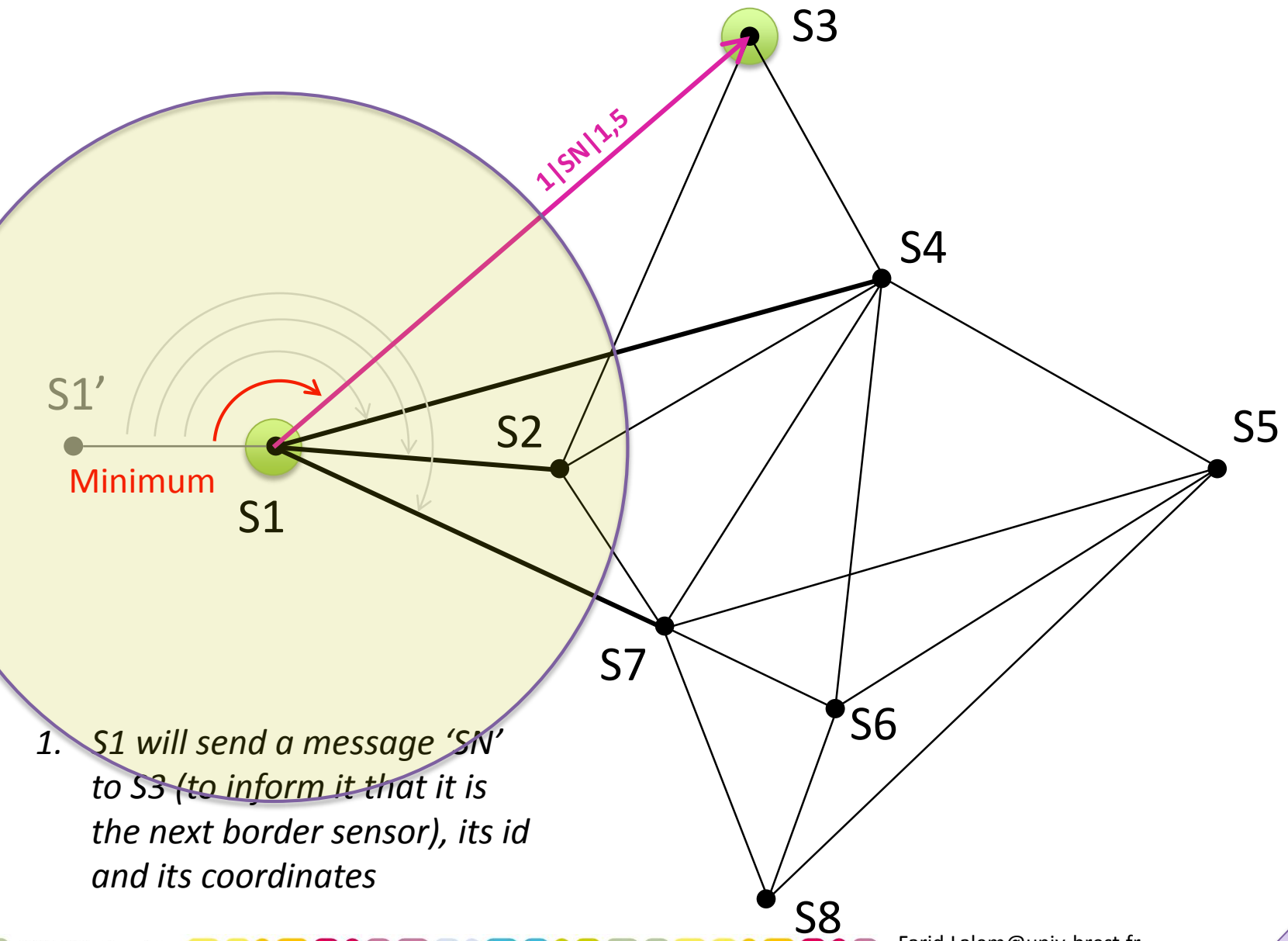
The distributed version



S1 will calculate the angles formed by a fictive point S1’ and its neighbors and choose the smallest one

Farid.Lalem@univ-brest.fr

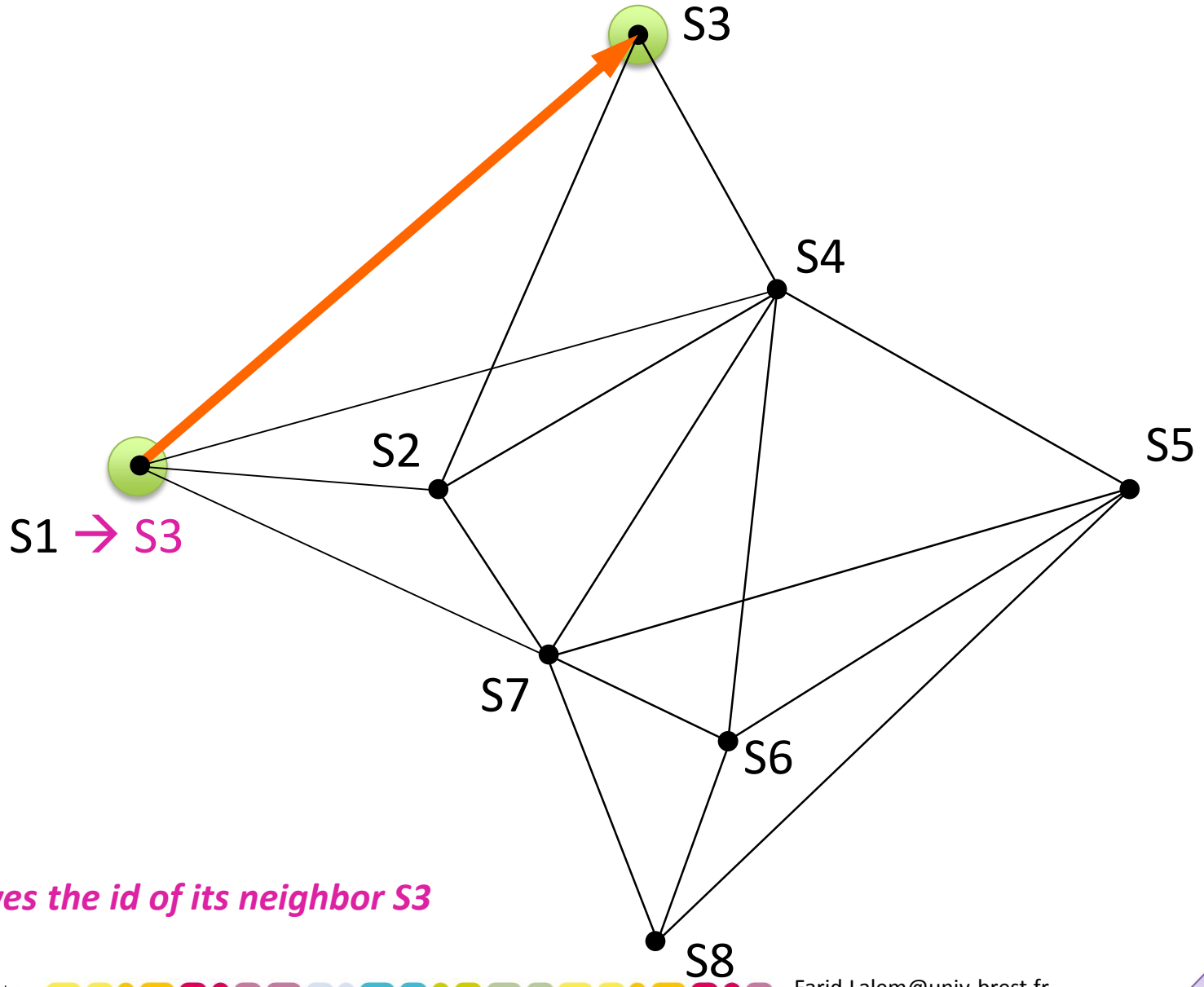
The distributed version



1. *S1 will send a message 'SN' to S3 (to inform it that it is the next border sensor), its id and its coordinates*

Farid.Lalem@univ-brest.fr

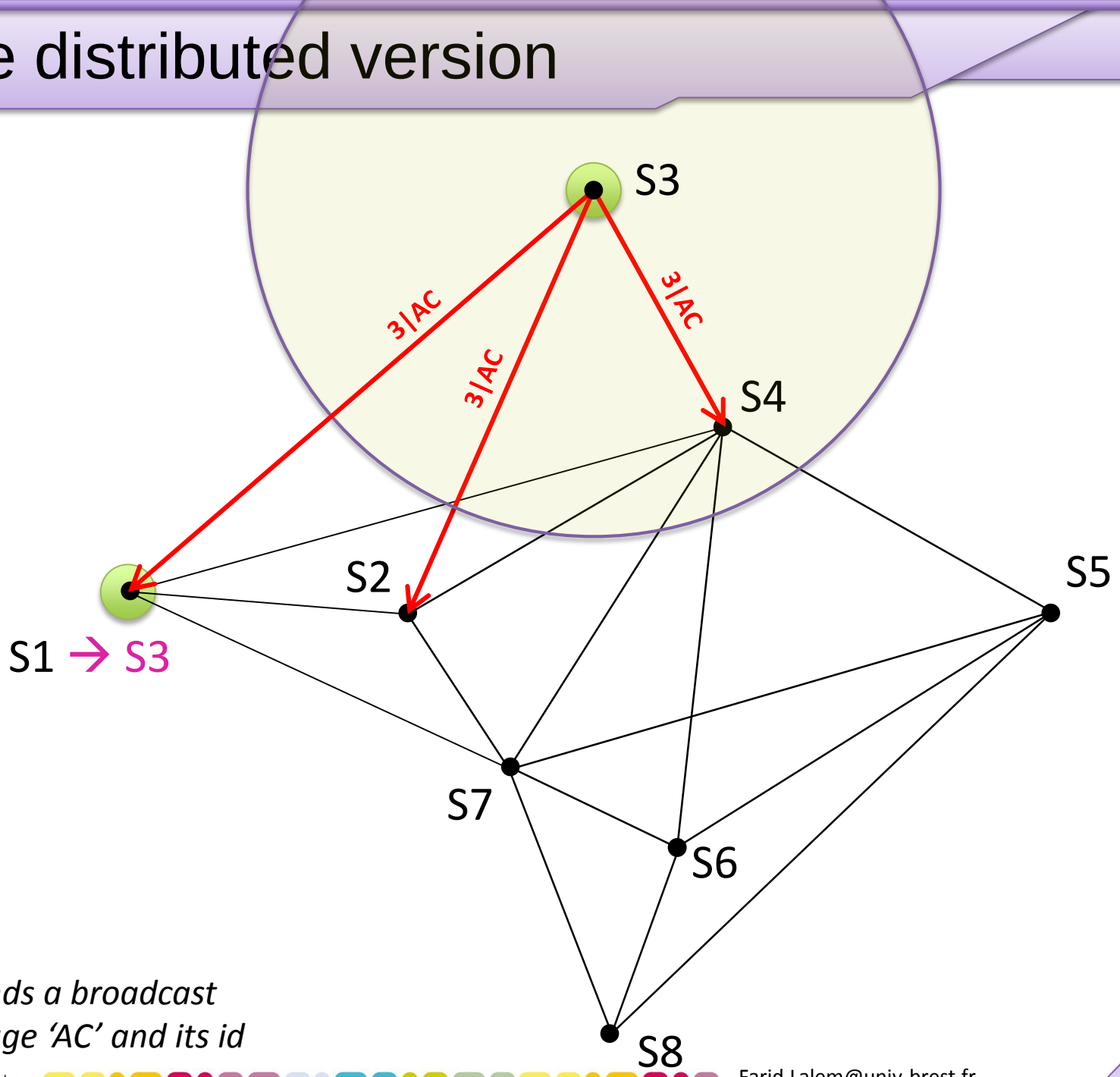
The distributed version



S1 saves the id of its neighbor S3

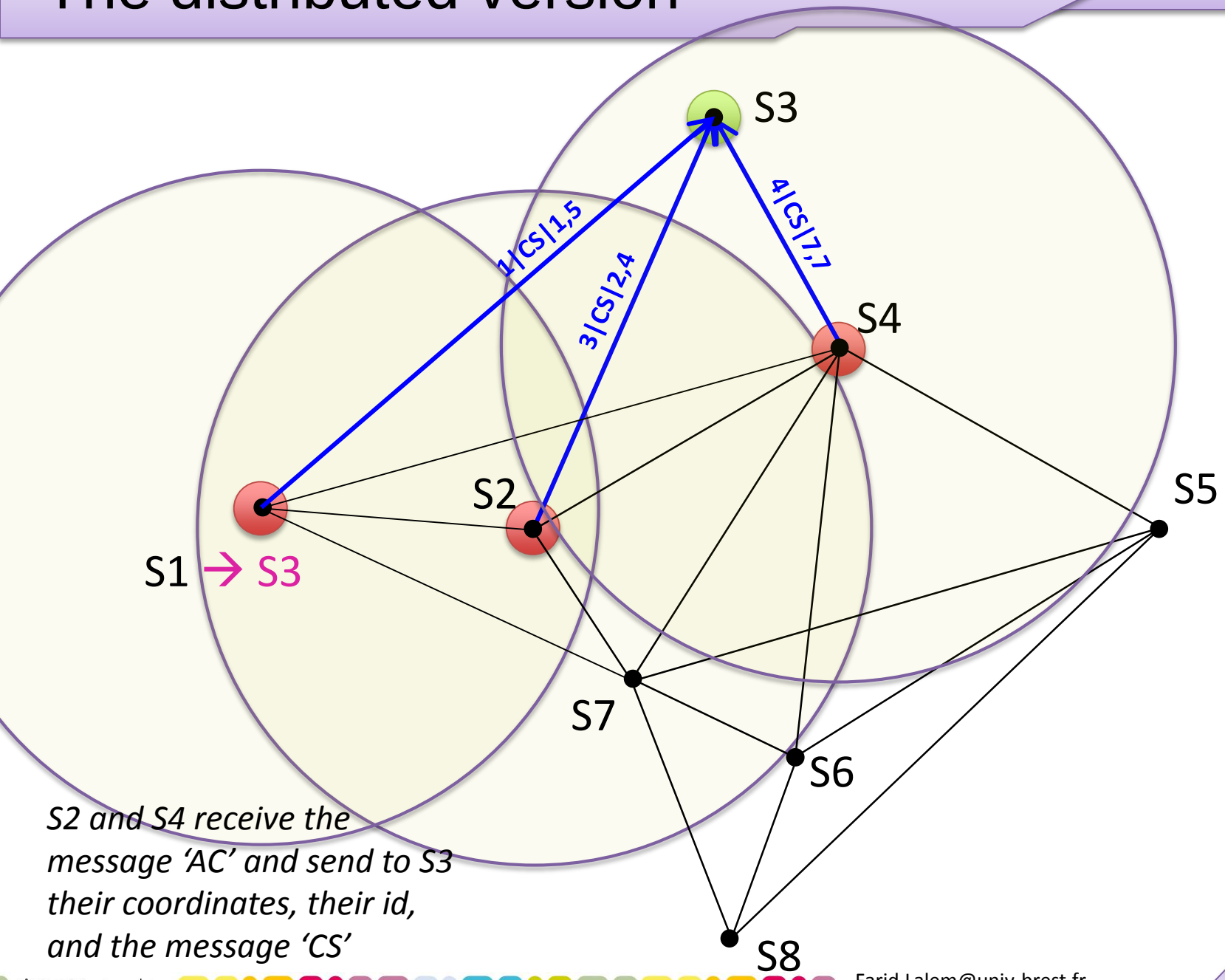
Farid.Lalem@univ-brest.fr

The distributed version



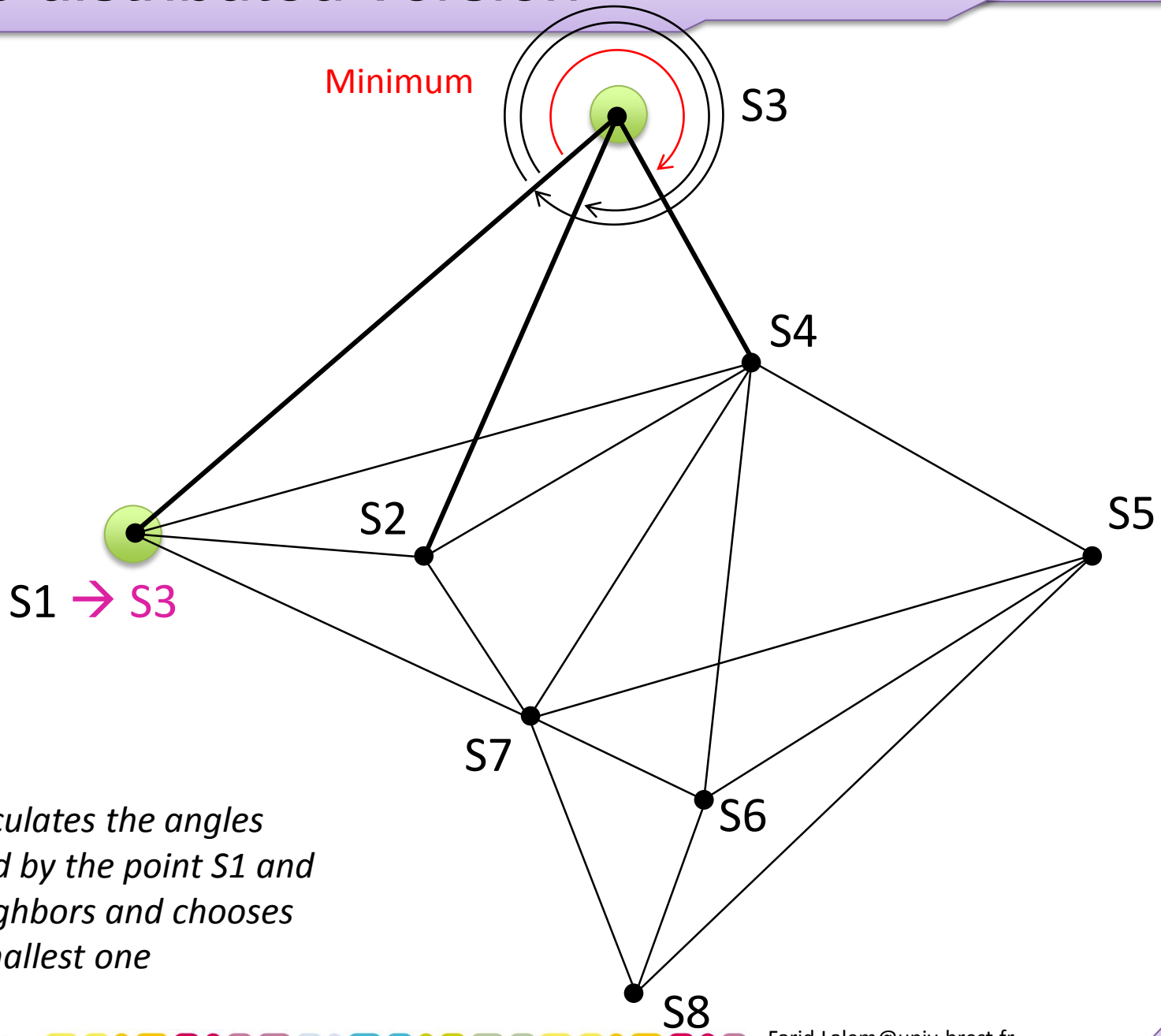
S3 sends a broadcast message 'AC' and its id

The distributed version



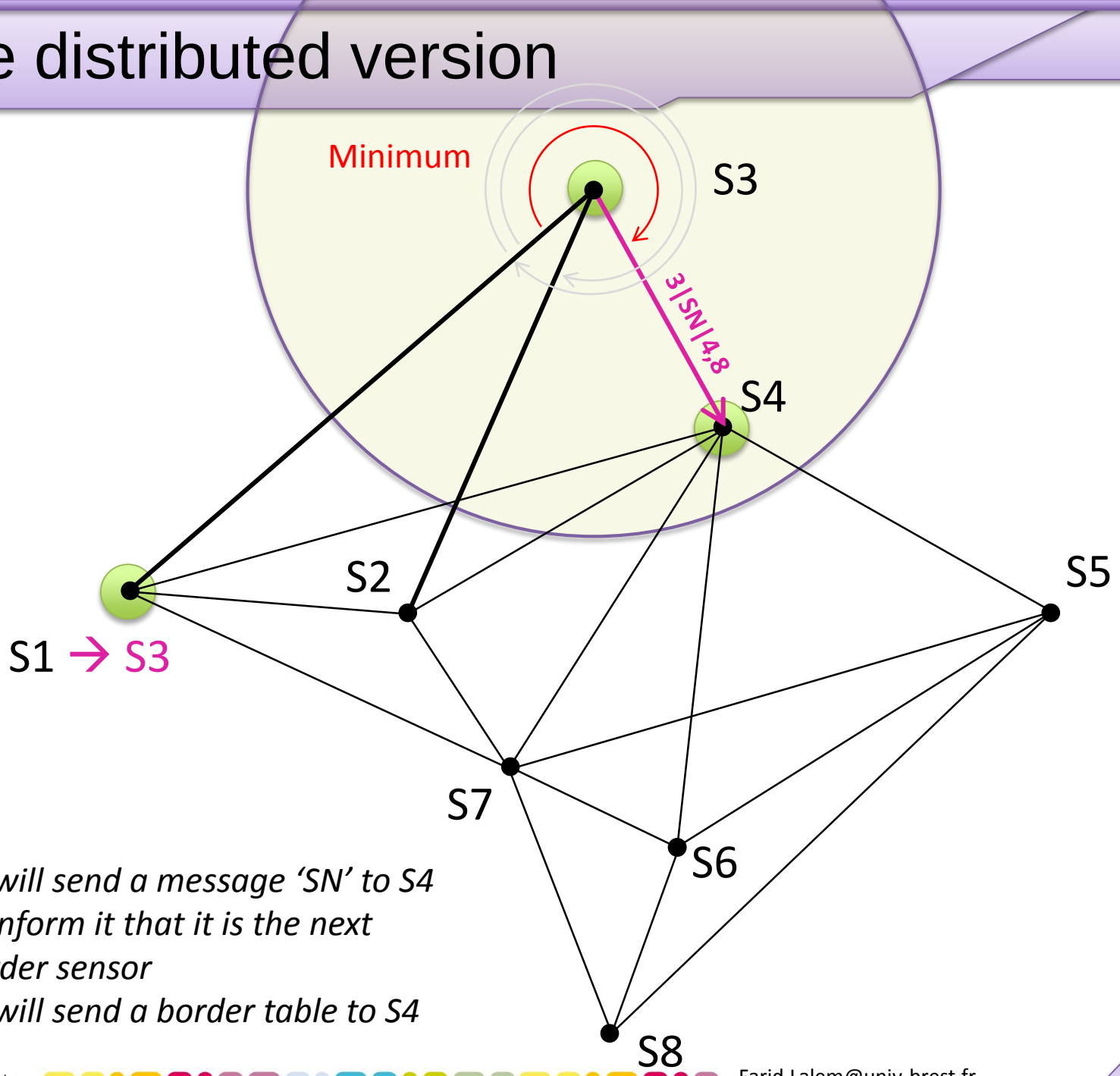
S2 and S4 receive the message 'AC' and send to S3 their coordinates, their id, and the message 'CS'

The distributed version



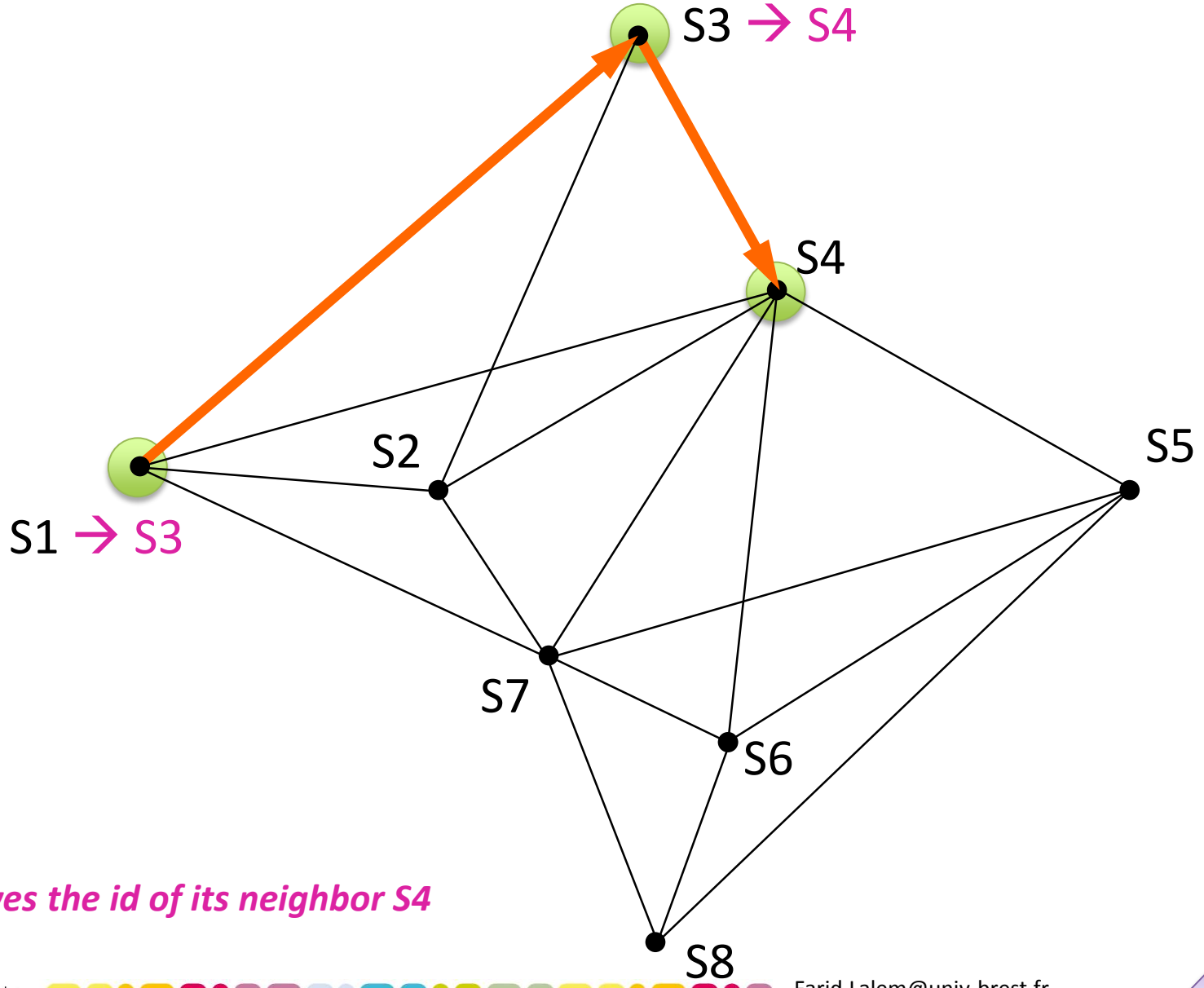
S3 calculates the angles formed by the point S1 and its neighbors and chooses the smallest one

The distributed version



1. *S3 will send a message 'SN' to S4 to inform it that it is the next border sensor*
2. *S3 will send a border table to S4*

The distributed version



S3 saves the id of its neighbor S4

The distributed version

Real WSN with
Arduino/XBee platform

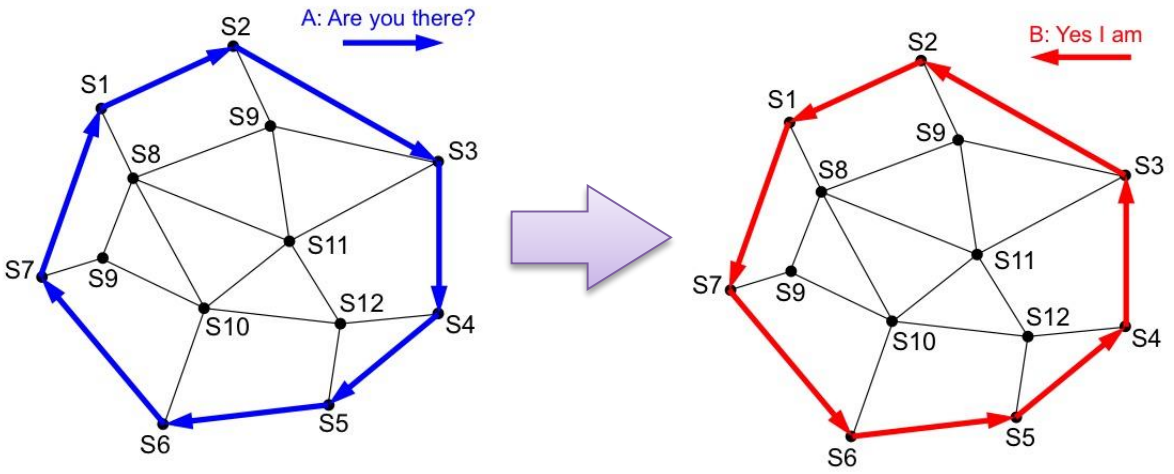
Farid.Lalem@univ-brest.fr

Boundary Node Detection

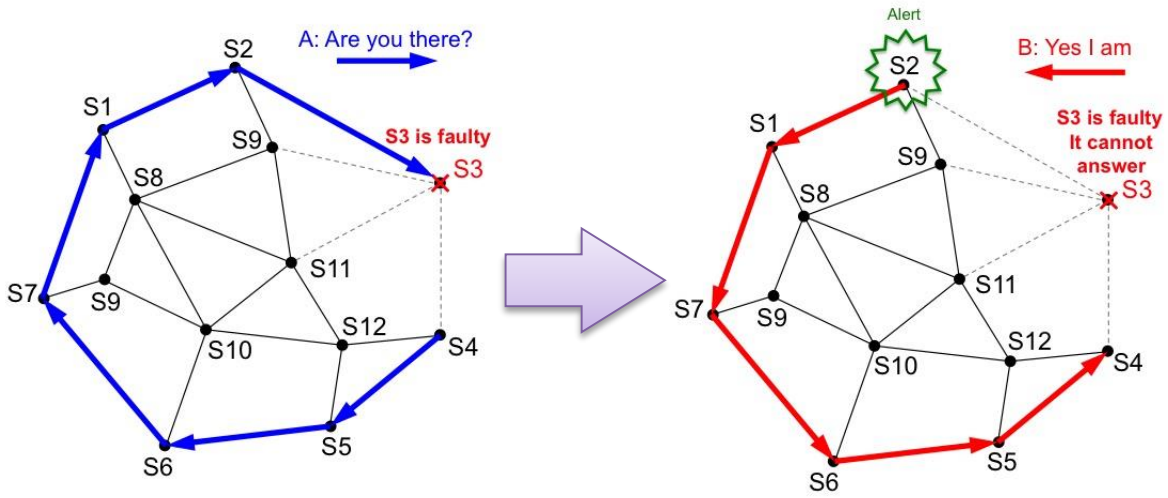
3

Test the presence

No faulty boundary nodes



1 faulty boundary node



Results with Castalia simulator

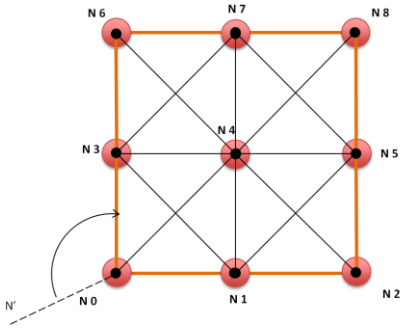
Simulation

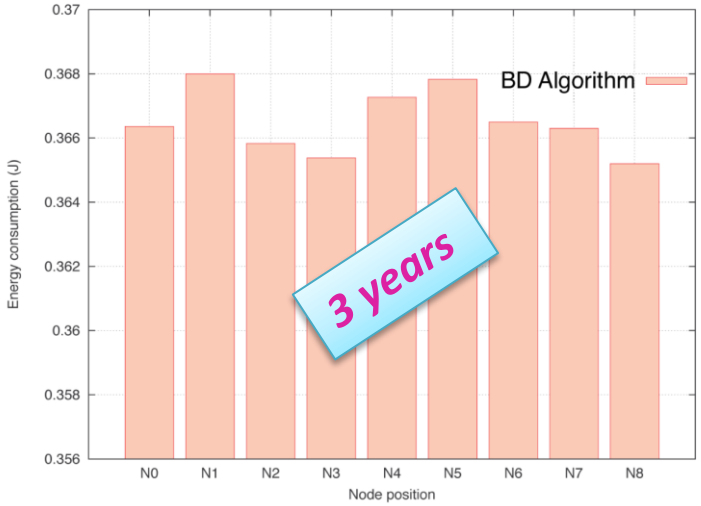
Castalia-3.3 framework

OMNeT++ 4.6 simulator

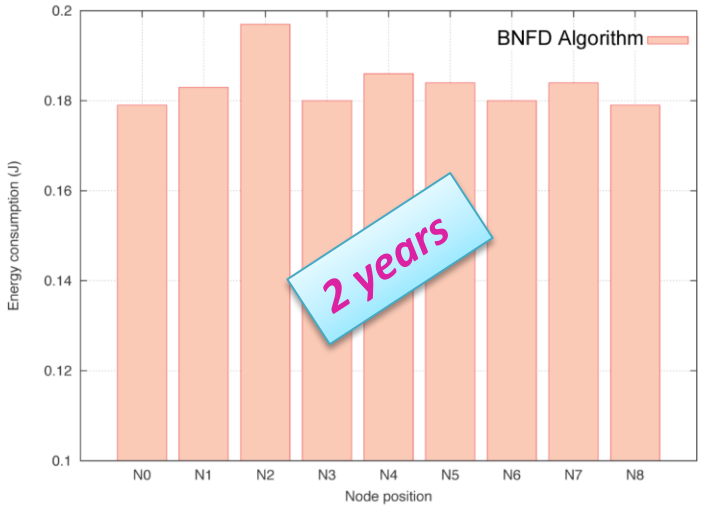
Initial battery capacity:

18720 Joules

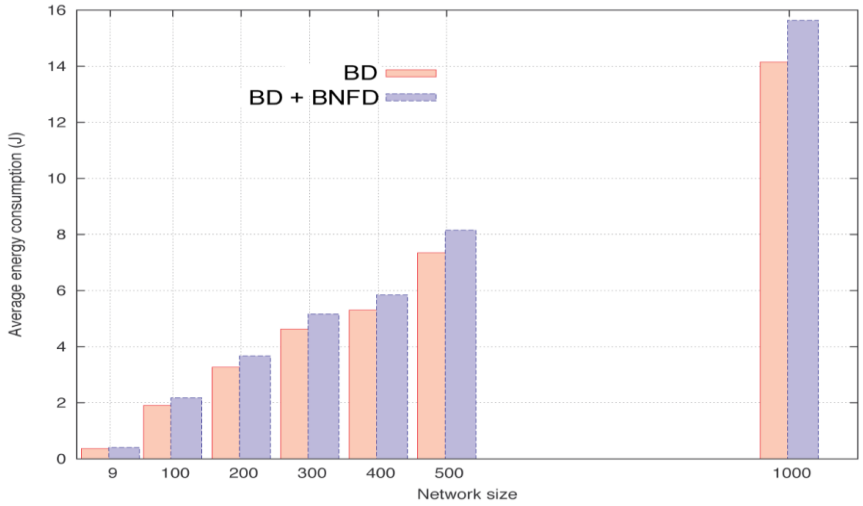




Energy Consumption per Node (BD Algorithm).



Energy Consumption per Node (BNFD Algorithm).



Energy consumption (BD and BD+BNFD Algorithms).

Farid.Lalem@univ-brest.fr

Conclusion

- A distributed approach to detect failure nodes in Wireless Sensor Networks (WSN) has been presented
- The proposed approach finds the WSN boundary nodes to monitor a sensitive area
- The evaluation of the performances shows that it is energy efficient

Farid.Lalem@univ-brest.fr

Thank You for your attention

Questions?

Farid Lalem, Rahim Kacimi, Ahcène Bounceur, and Reinhardt Euler

Farid. Lalem@univ-brest.fr