

# FIXED IMAGE SENSORS AND MOBILE CAMERA ROBOTS INTERACTIONS FOR MISSION- CRITICAL SURVEILLANCE APPLICATIONS

IROS'2012, ROSIN WORKSHOP  
OCTOBER 11TH, 2012  
VILAMOURA, PORTUGAL

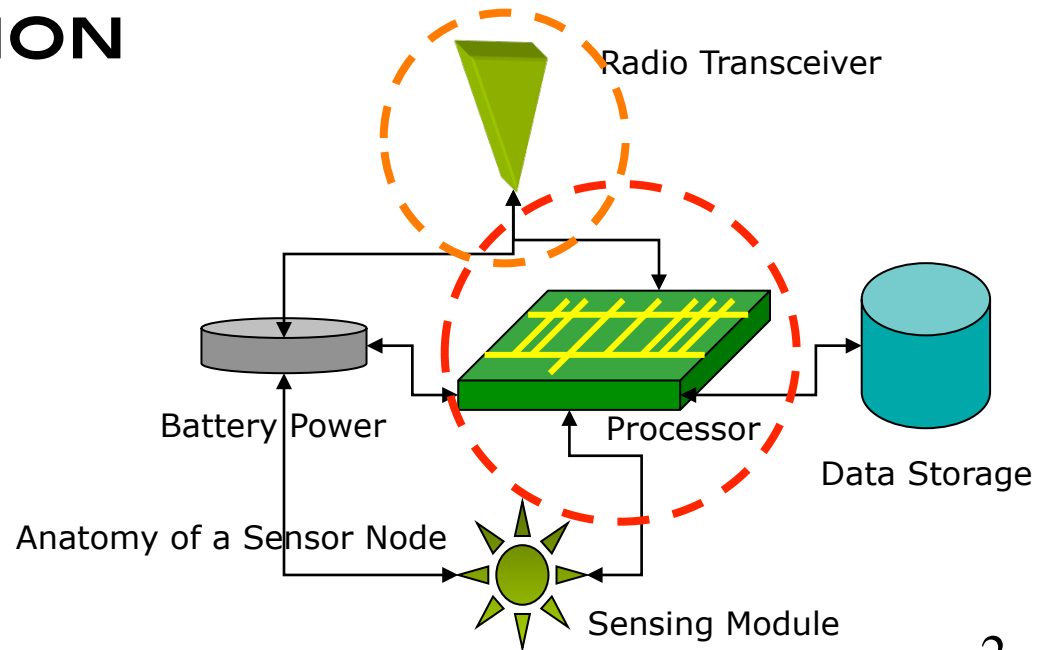
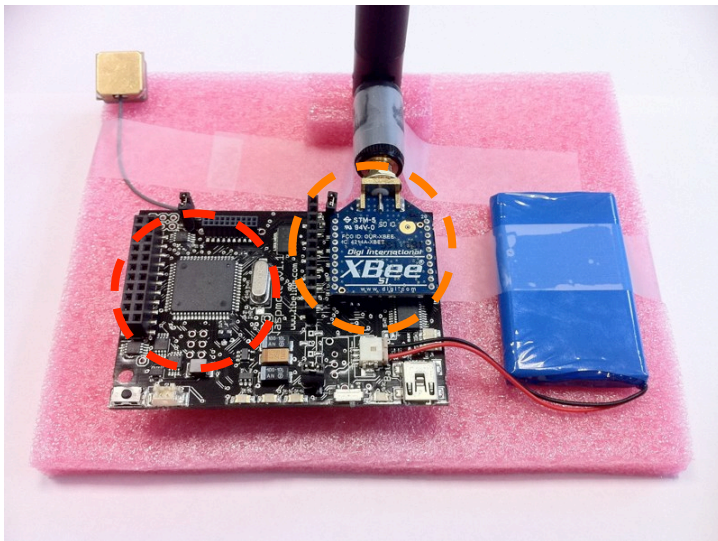


PROF. CONGDUC PHAM  
[HTTP://WWW.UNIV-PAU.FR/~CPHAM](http://www.univ-pau.fr/~cpham)  
UNIVERSITÉ DE PAU, FRANCE

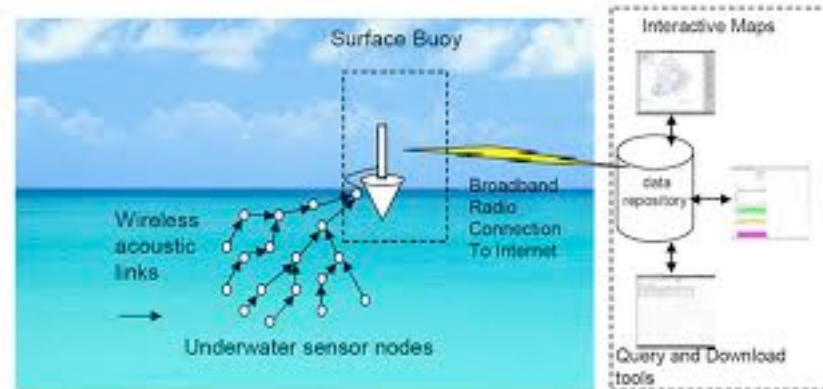


# WIRELESS AUTONOMOUS SENSORS

- ❑ IN GENERAL: LOW COST, LOW POWER (THE BATTERY MAY NOT BE REPLACEABLE), SMALL SIZE, PRONE TO FAILURE, POSSIBLY DISPOSABLE
- ❑ ROLE: SENSING, DATA PROCESSING, COMMUNICATION



# MONITORING/SURVEILLANCE

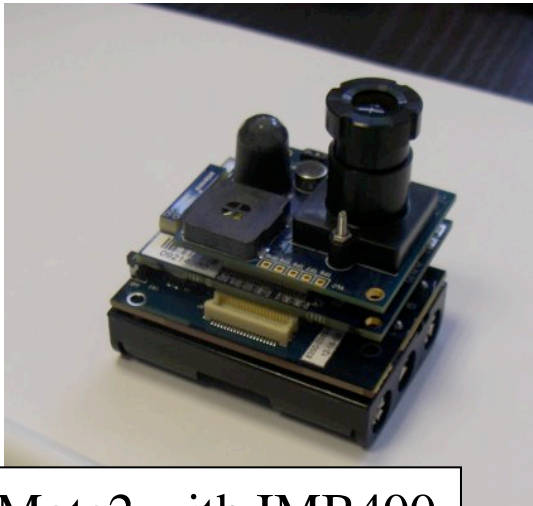
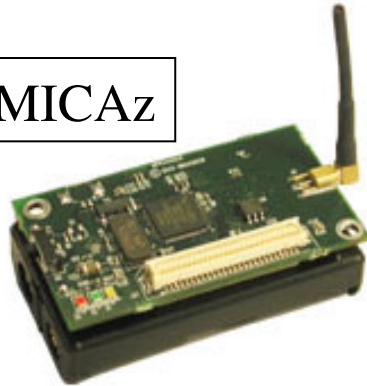


# IMAGE SENSOR MOTES



iMote2

MICAz



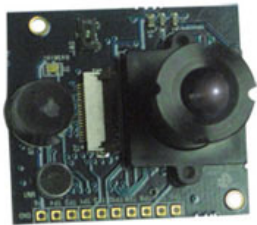
iMote2 with IMB400 multimedia board



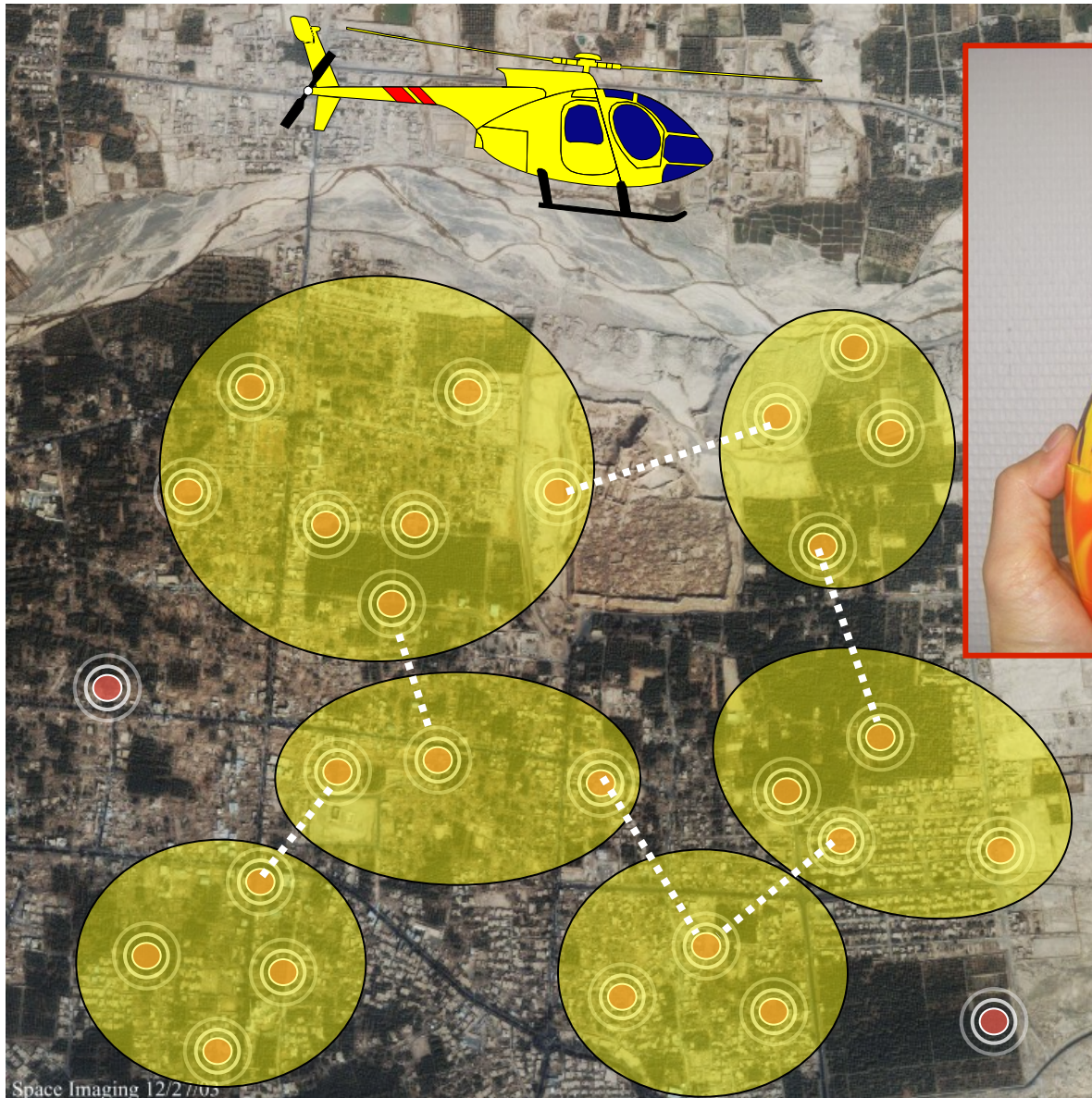
# SEARCH & RESCUE



Imote2



Multimedia board

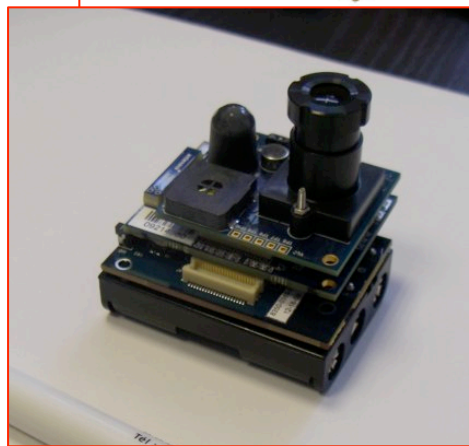
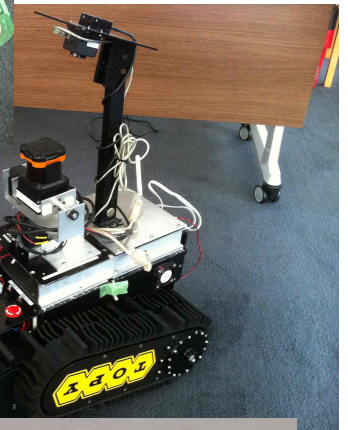
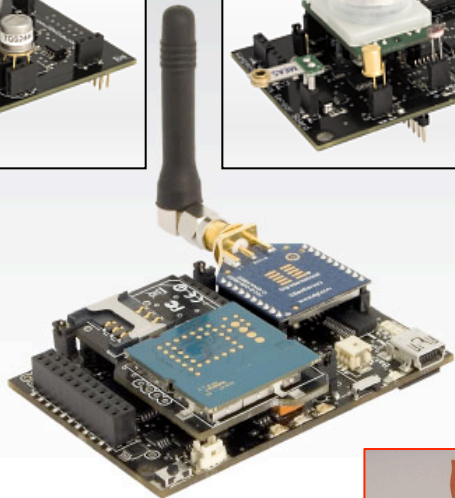
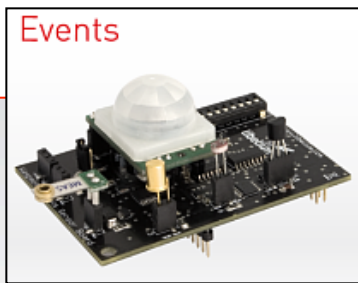


# GET IMAGES FROM DEPLOYED SENSORS



# SENSORS & ROBOTS

TAKING ADVANTAGES OF DIFFERENCES!



# SENSORS & ROBOTS

## TAKING ADVANTAGES OF DIFFERENCES!

Gases

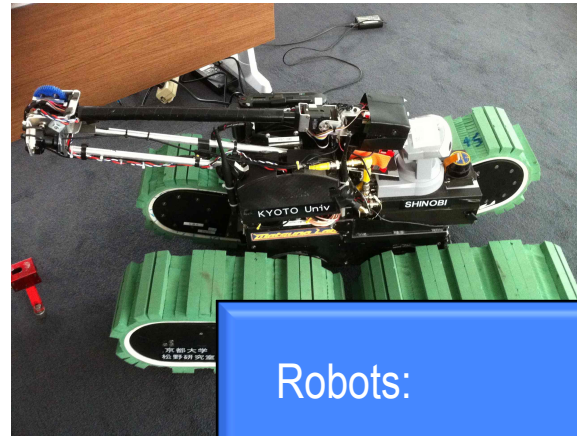
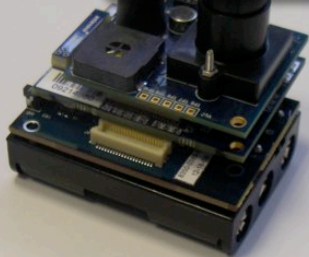


Events



### Wireless Sensor Networks:

- Large scale sensing
- Natural collaboration though data aggregation, reporting, ...
- Mobility is not a priority



### Robots:

- Mobility is a fundamental feature
- Exploration, rescue





# SENSOR & ROBOTS SEARCH & RESCUE

- RESCUE COULD BE OPERATED IN SEVERAL PHASES (1)

Deploy in mass a WSN to get a first snapshot of the area conditions: images, radiation level, targets,...



# SENSOR & ROBOTS SEARCH & RESCUE

## □ RESCUE COULD BE OPERATED IN SEVERAL PHASES (2)

Based on collected data, optimize deployment/selection of autonomous robots, depending on mobility type or embedded hardware.



# SENSOR & ROBOTS SEARCH & RESCUE

## □ RESCUE COULD BE OPERATED IN SEVERAL PHASES (3)

Robots could serve as relay or install communication gateways to maintain WSN connectivity and increase data storage capability



# SENSOR & ROBOTS SEARCH & RESCUE

## □ RESCUE COULD BE OPERATED IN SEVERAL PHASES (4)

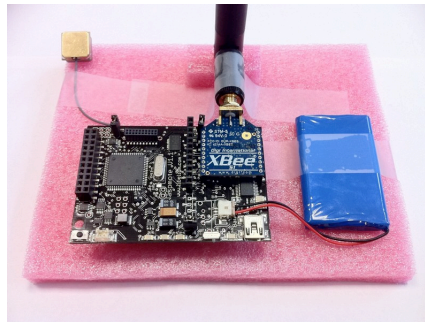
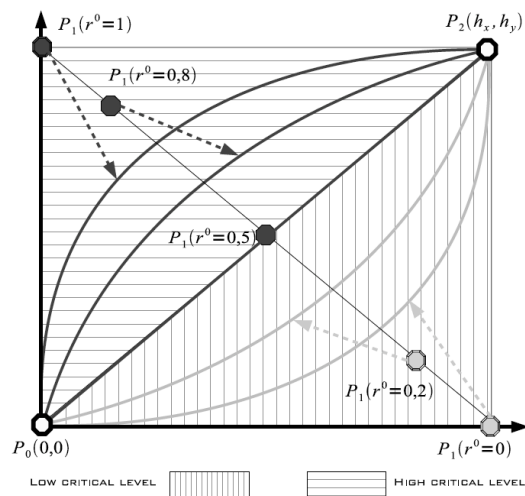
Sensor & Robots will continuously collaborate during the rescue process: localization, path optimization, remote sensing,...



# SENSORS & ROBOTS

## PROPOSE NEW INTERACTION SCHEMES

- USE THE CRITICALITY MODEL TO CONTROL BOTH SENSORS AND ROBOT
- SAVE ENERGY OF SENSOR NODES

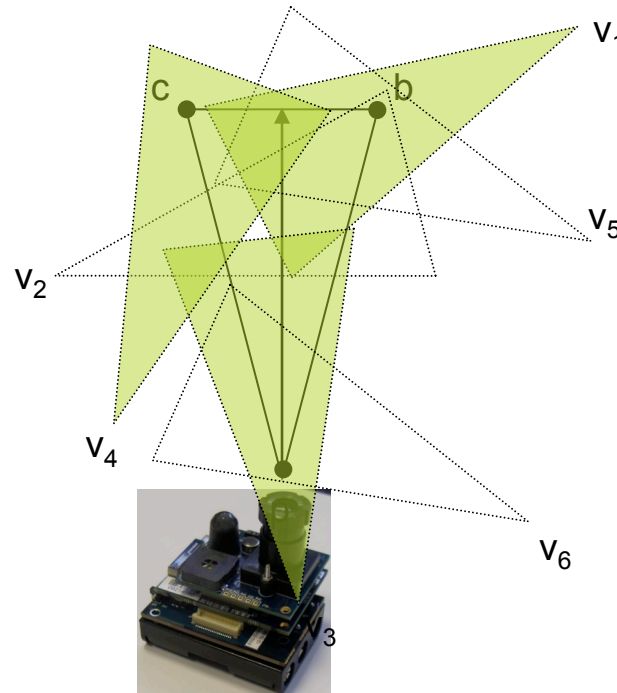


# SENSOR NODE'S COVER SET

$\text{Co}(\mathbf{V}) = \{$   
 $\{\mathbf{V}\},$   
 $\{\mathbf{V}_1, \mathbf{V}_3, \mathbf{V}_4\},$   
 $\{\mathbf{V}_2, \mathbf{V}_3, \mathbf{V}_4\},$   
 $\{\mathbf{V}_3, \mathbf{V}_4, \mathbf{V}_5\},$   
 $\{\mathbf{V}_1, \mathbf{V}_4, \mathbf{V}_6\},$   
 $\{\mathbf{V}_2, \mathbf{V}_4, \mathbf{V}_6\},$   
 $\{\mathbf{V}_4, \mathbf{V}_5, \mathbf{V}_6\}$   
 $\}$



$|\text{Co}(\mathbf{V})| = 7$



# CRITICALITY MODEL (1)

- LINK THE CAPTURE RATE TO THE SIZE OF THE COVER SET

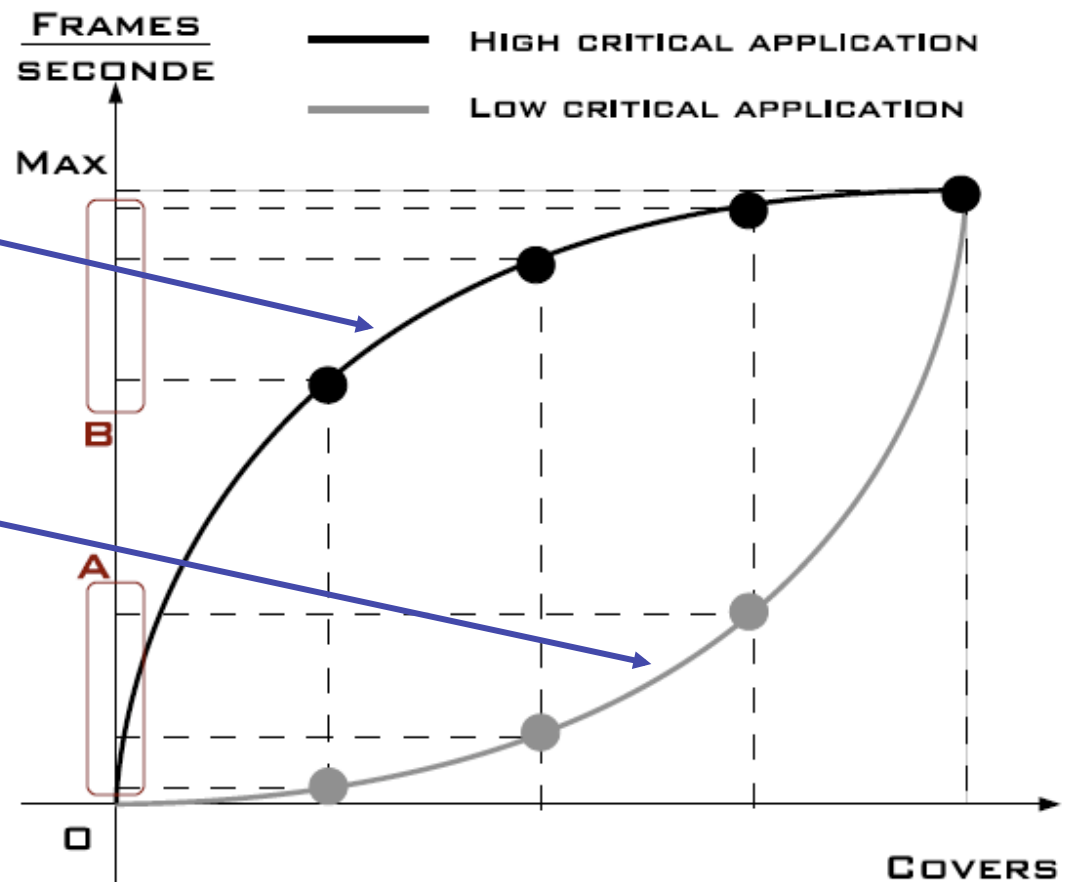
- HIGH CRITICALITY

- CONVEX SHAPE
- MOST PROJECTIONS OF X ARE CLOSE TO THE MAX CAPTURE SPEED

- LOW CRITICALITY

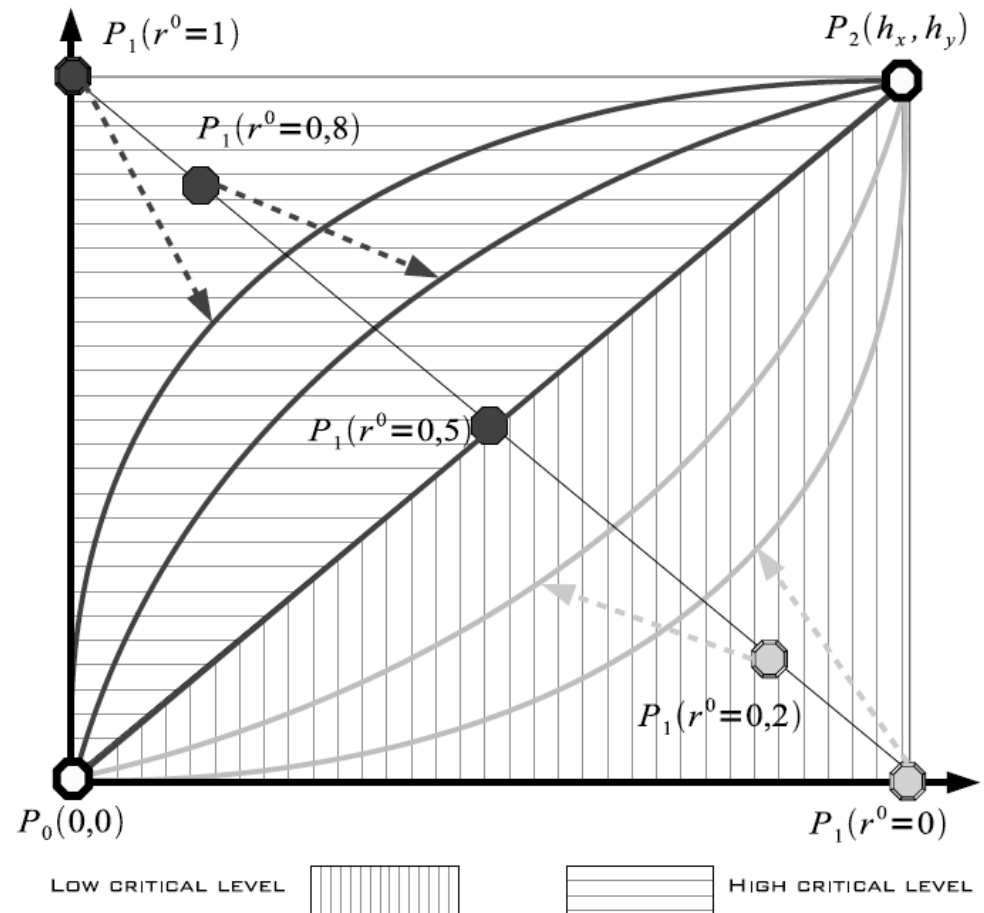
- CONCAVE SHAPE
- MOST PROJECTIONS OF X ARE CLOSE TO THE MIN CAPTURE SPEED

- CONCAVE AND CONVEX SHAPES AUTOMATICALLY DEFINE SENTRY NODES IN THE NETWORK



# CRITICALITY MODEL (2)

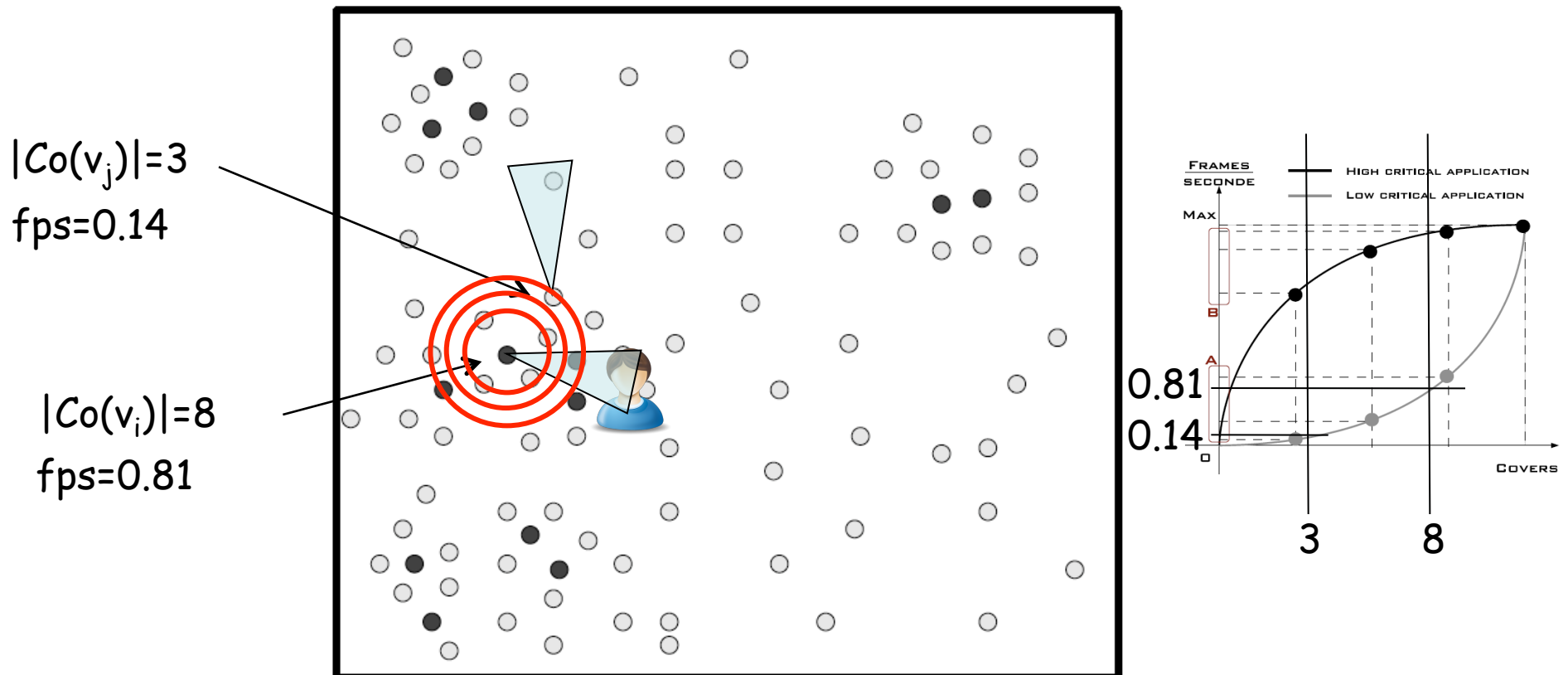
- ❑  $R^0$  CAN VARY IN  $[0,1]$
- ❑ BEHAVIOR FUNCTIONS (BV) DEFINES THE CAPTURE SPEED ACCORDING TO  $R^0$
- ❑  $R^0 < 0.5$ 
  - ❑ CONCAVE SHAPE BV
- ❑  $R^0 > 0.5$ 
  - ❑ CONVEX SHAPE BV
- ❑ WE PROPOSE TO USE BEZIER CURVES TO MODEL BV FUNCTIONS





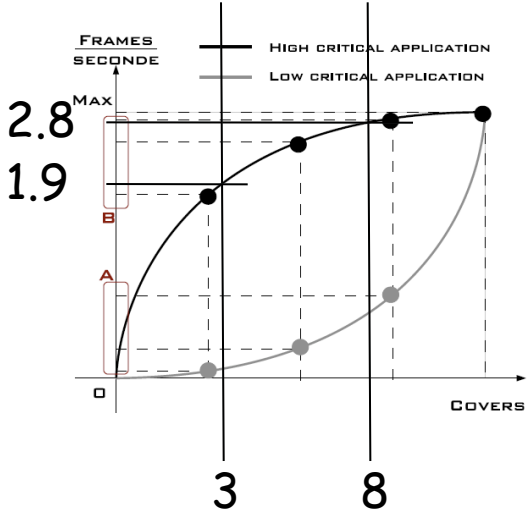
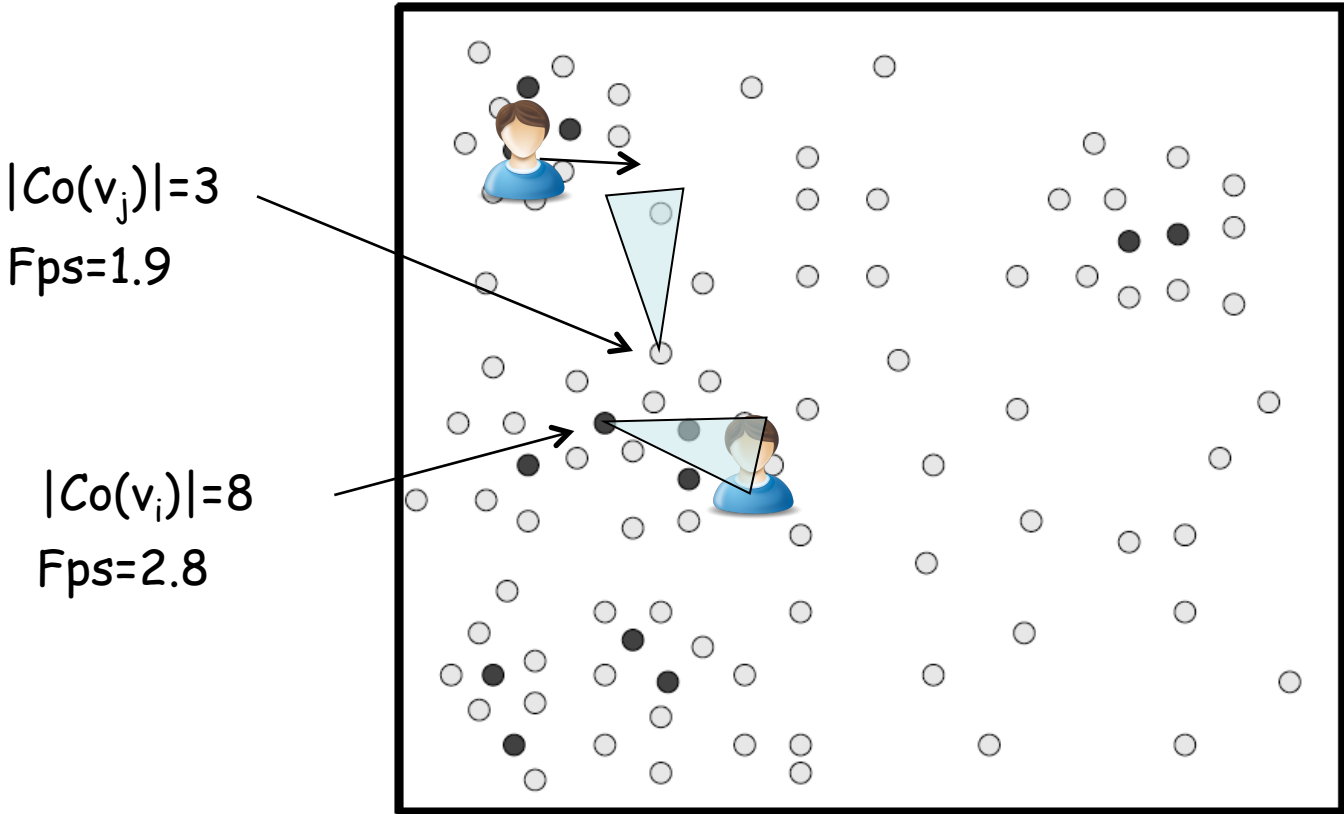
# RISK-BASED SCHEDULING IN IMAGES (1)

□  $R^\circ = R^\circ_{\text{MIN}} = 0.1$ ,  $R^\circ_{\text{MAX}} = 0.9$ , NO ALERT



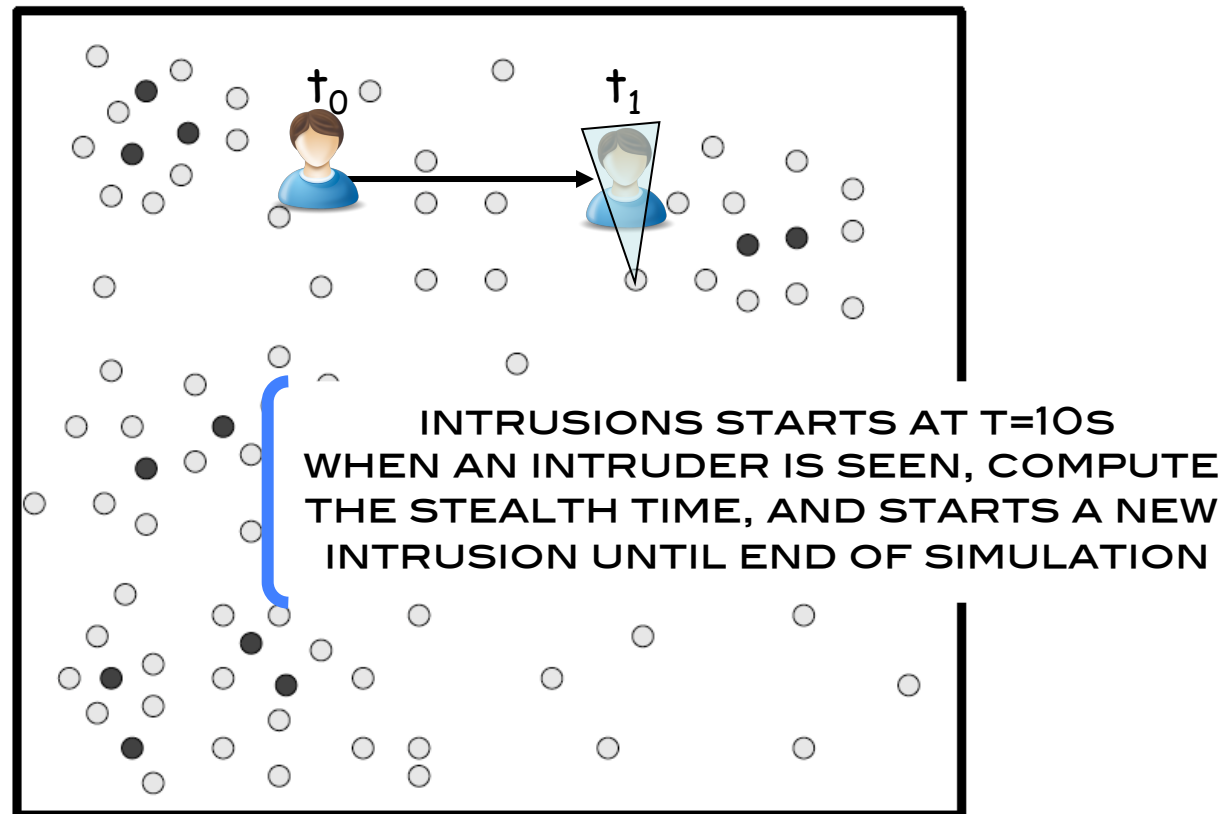
# RISK-BASED SCHEDULING IN IMAGES (2)

$R^o \rightarrow R^o = R^o_{MAX} = 0.9$

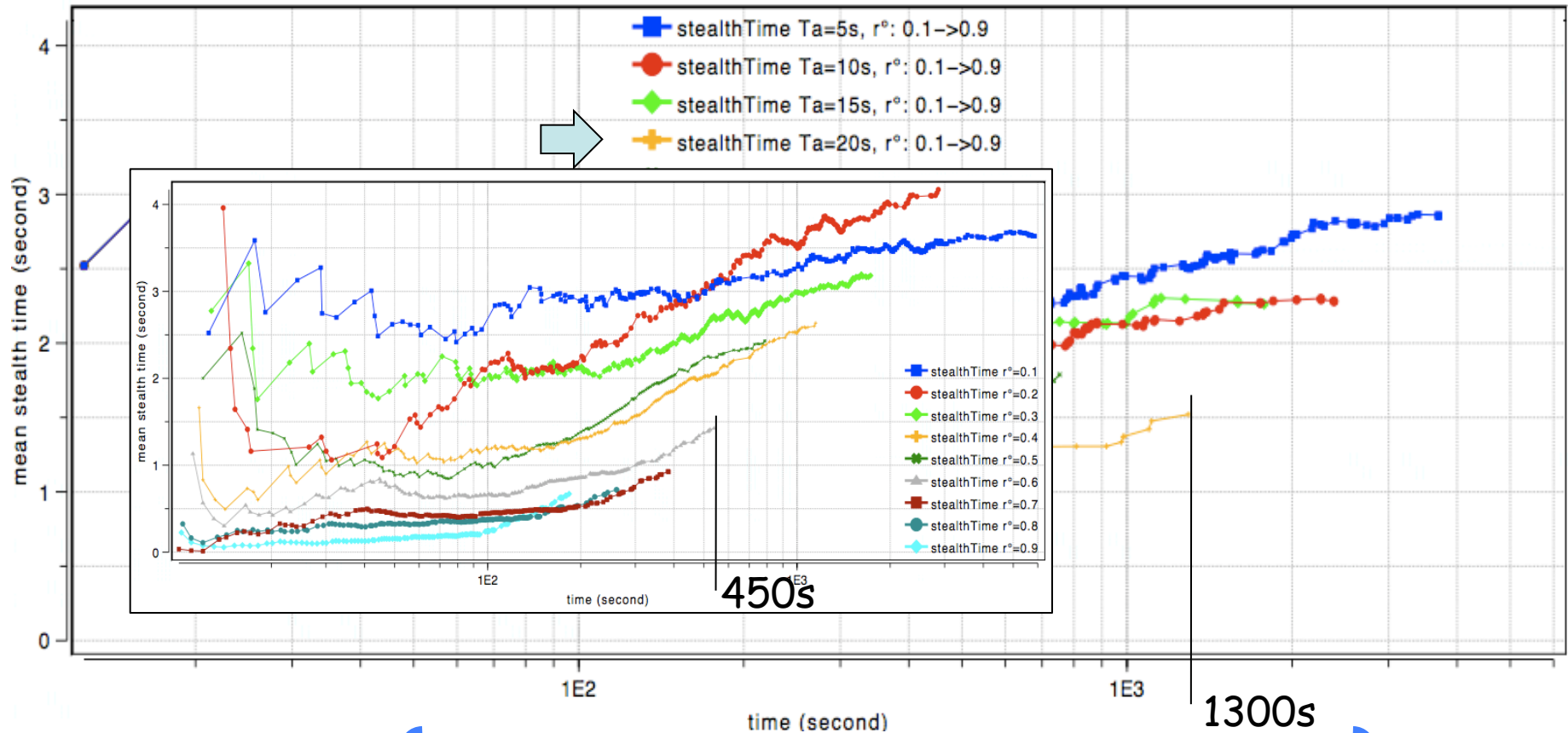


# MEAN STEALTH TIME

$T_1 - T_0$  IS THE INTRUDER'S STEALTH TIME  
VELOCITY IS SET TO 5M/S



# MEAN STEALTH TIME RISK-BASED SCHEDULING

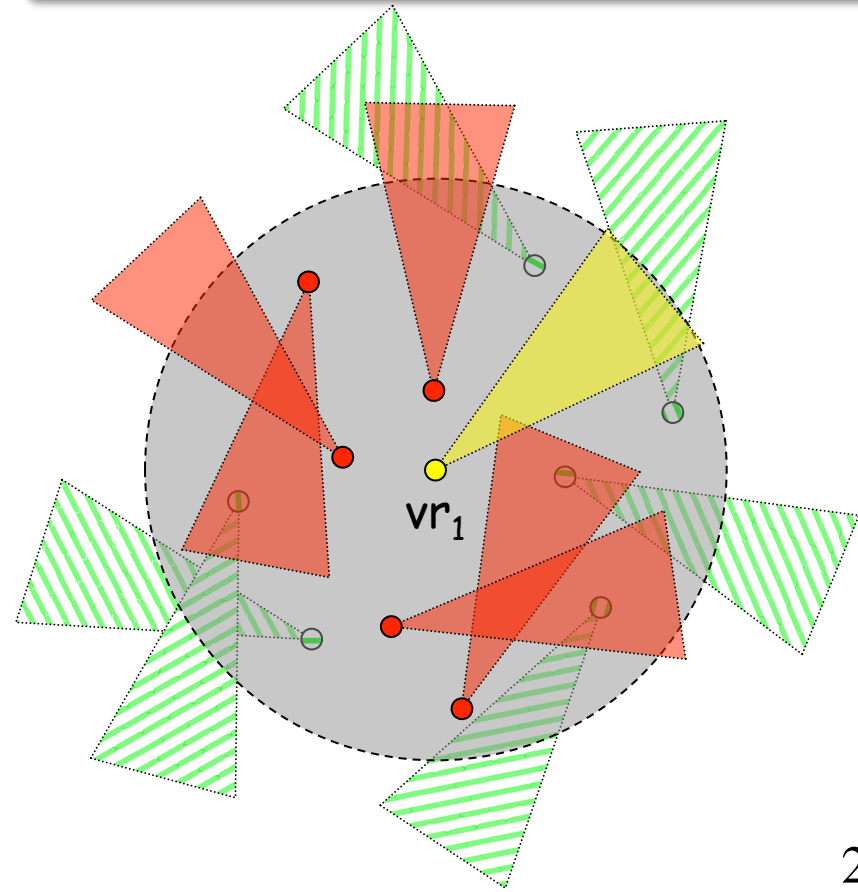
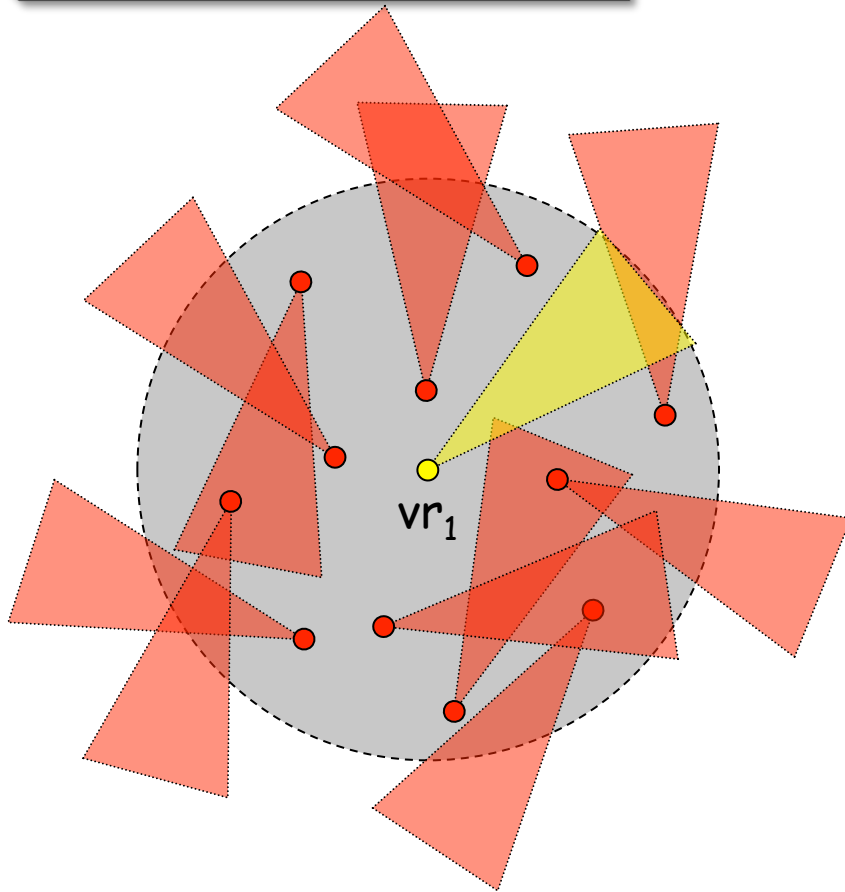


SENSOR NODES START AT 0.1 THEN INCREASE TO 0.9 IF ALERTED (BY INTRUDERS OR NEIGHBORS) AND STAY ALERTED FOR TA SECONDS

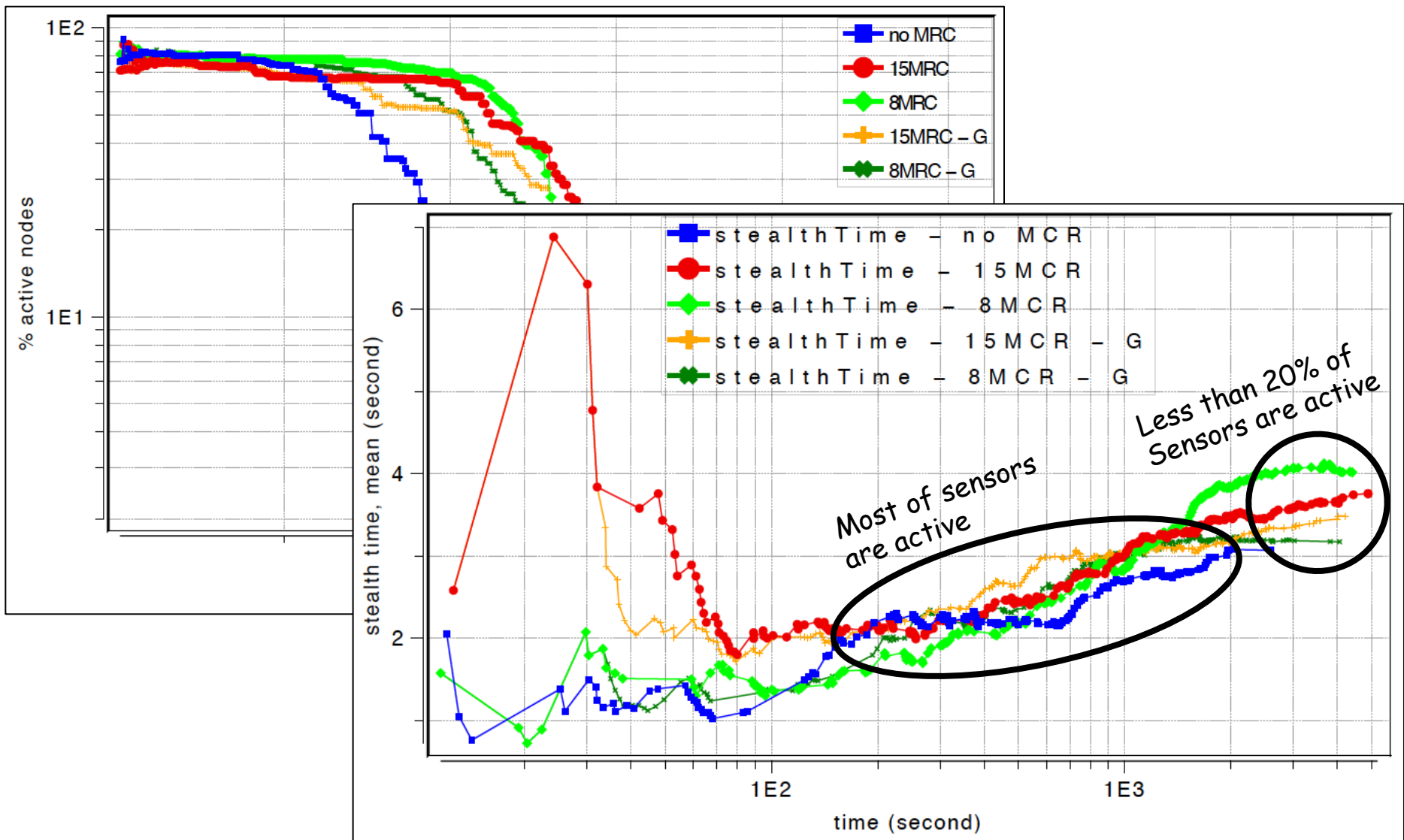
# COOPERATION WITH CAMERAS ON MOBILE ROBOTS

Fixed image sensors near a mobile camera can decrease their criticality level to  $R_{\min}^{\circ}$

**ONLY** fixed image sensors whose FoV's center is covered by a mobile camera **CAN** decrease their criticality level to  $R_{\min}^{\circ}$

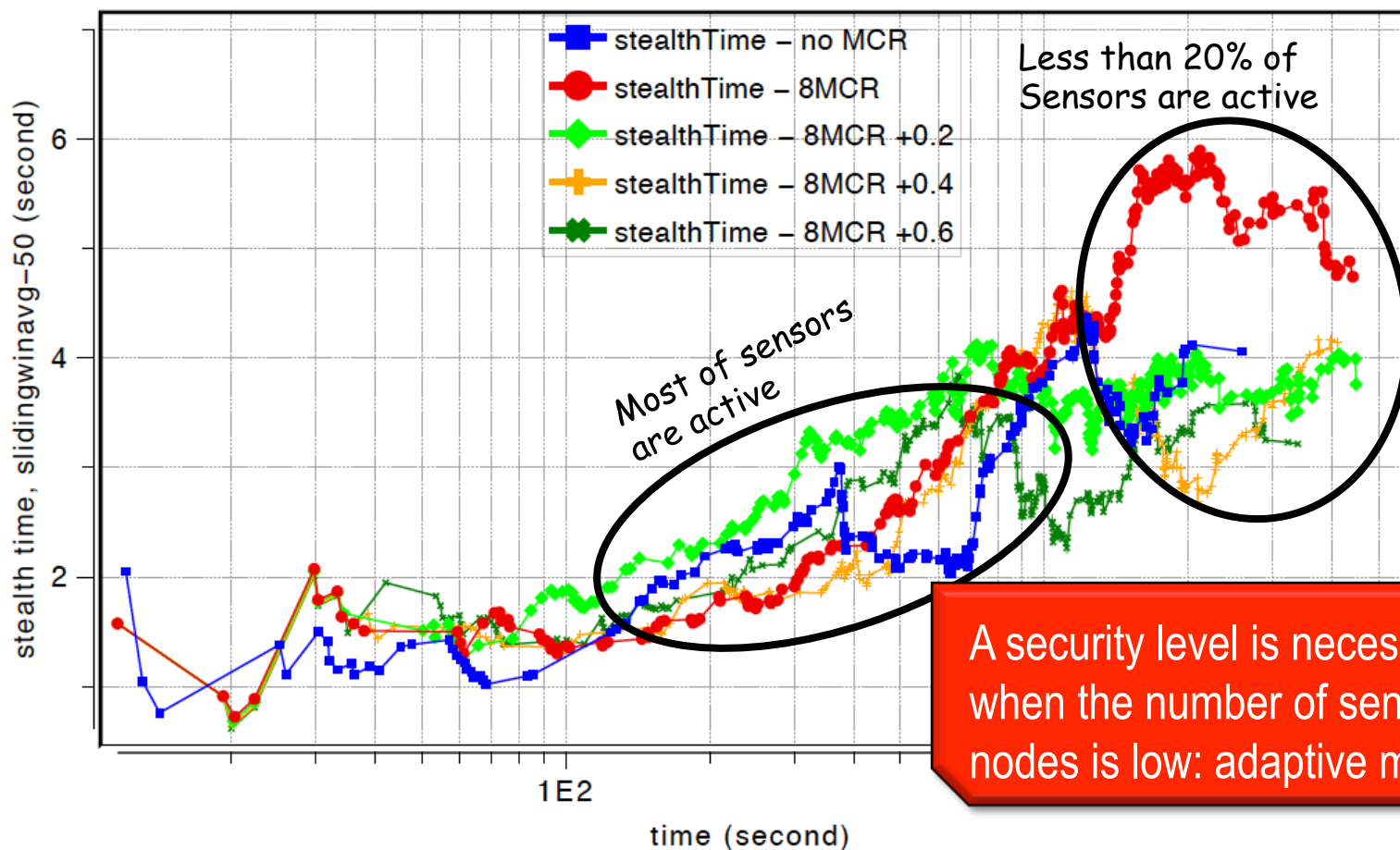


# IMPACT ON LIFETIME & STEALTH TIME



# HOW LOW THE CRITICALITY LEVEL COULD BE REDUCED?

□ WE ADD A SECURITY LEVEL:  $R^{\circ}_{MIN} + S$

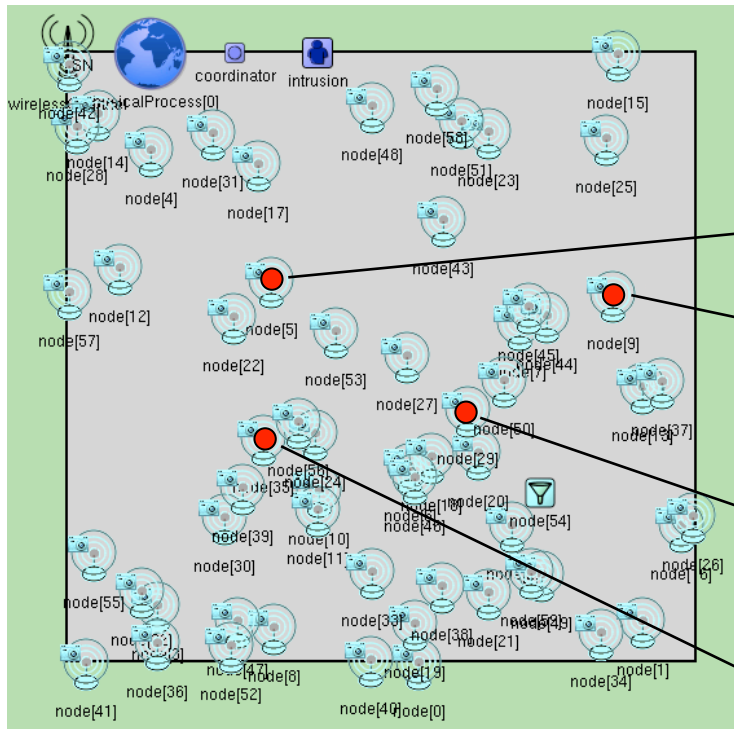


# CONCLUSIONS

- ❑ **SENSORS & ROBOTS ARE COMPLEMENTARY TECHNOLOGIES FOR MISSION-CRITICAL APPLICATIONS**
- ❑ **WE PROPOSE THAT BOTH SENSORS AND MOBILE ROBOTS SHARE THE SAME CRITICALITY MODEL**
- ❑ **THEN MOBILE ROBOTS CAN HELP FIXED SENSOR NODES TO SAVE THEIR ENERGY WITHOUT DEGRADING THE SURVEILLANCE QUALITY**
- ❑ **THERE ARE A LOT OF VARIANTS THAT COULD BE IMPLEMENTED**
- ❑ **ENERGY SAVING IS NOT THE ONLY GAIN**
  - ❑ **FEWER CONGESTIONS, LESS CONTENTION ON RADIO MEDIUM, MORE IMAGE PACKETS RECEIVED AT THE SINK**
  - ❑ **SYNCHRONOUS MAC LAYER COULD USE CAMERA ROTATION TIME TO ADAPT THE LISTENING INTERVAL OR DETERMINE NAV-LIKE RESERVATION VECTORS**
  - ❑ **NODES COULD RATHER FOCUS ON RELAYING PACKETS**



# A SIMULATOR FOR IMAGE SENSORS



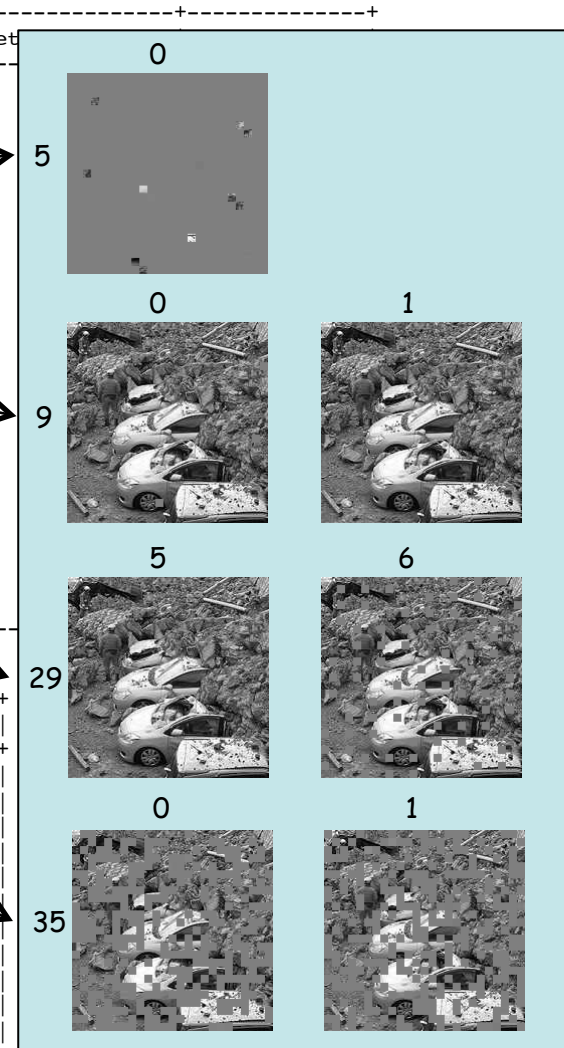
60 IMAGE SENSOR NODES  
75MX75M  
1 SINK (NODE 54)

Application:Image sent

|         | Images | Packets | by coverset |
|---------|--------|---------|-------------|
| node=2  | 1      | 206     | 0           |
| node=5  | 4      | 824     | 0           |
| node=9  | 2      | 412     | 2           |
| node=10 | 6      | 1236    | 6           |
| node=12 | 1      | 206     | 0           |
| node=15 | 2      | 412     | 2           |
| node=17 | 1      | 206     | 0           |
| node=19 | 3      | 618     | 0           |
| node=22 | 4      | 824     | 0           |
| node=23 | 2      | 412     | 0           |
| node=24 | 6      | 1236    | 0           |
| node=26 | 1      | 206     | 1           |
| node=27 | 6      | 1236    | 0           |
| node=29 | 7      | 1442    | 6           |
| node=33 | 6      | 1236    | 6           |
| node=35 | 12     | 2472    | 0           |
| node=37 | 5      | 1030    | 0           |
| node=40 | 8      | 1648    | 3           |
| node=46 | 2      | 412     | 2           |
| node=48 | 2      | 412     | 0           |
| node=50 | 2      | 412     | 2           |

Application:Image displayed

|          | all | complete | truncated |
|----------|-----|----------|-----------|
| index=-1 | 39  | 21       | 18        |
| index=5  | 1   | 0        | 1         |
| index=9  | 2   | 1        | 1         |
| index=10 | 6   | 3        | 3         |
| index=23 | 2   | 0        | 2         |
| index=24 | 3   | 0        | 3         |
| index=27 | 4   | 4        | 0         |
| index=29 | 7   | 6        | 1         |
| index=33 | 3   | 3        | 0         |
| index=35 | 4   | 0        | 4         |
| index=37 | 5   | 3        | 2         |
| index=50 | 2   | 1        | 1         |



# SENSORS & ROBOTS ENABLE REALISTIC INTERACTION STUDIES

Sensor specific simulator for communication stack

Get robot's position from robot simulator

Re-use fine-grained communication protocols and complex radio models

Re-use complex hardware (laser scan, ...) and control software (navigation stacks,...)

The image displays a network diagram on the left, titled "(SN) SN", showing a central "coordinator" node connected to an "intrusion" node, and a "wirelessChannel" node. Below these are numerous "node" icons, each with a robot icon and a label like "node[14]", "node[28]", "node[4]", "node[17]", "node[23]", "node[25]", "node[12]", "node[5]", "node[22]", "node[7]", "node[9]", "node[27]", "node[13]", "node[29]", "node[20]", "node[8]", "node[2]", "node[16]", "node[19]", "node[0]", "node[1]", "node[21]", "node[10]", "node[11]", "node[15]", "node[18]", "node[24]", "node[26]", "node[3]", "node[6]", "node[12]", "node[13]", "node[14]", "node[15]", "node[16]", "node[17]", "node[18]", "node[19]", "node[20]", "node[21]", "node[22]", "node[23]", "node[24]", "node[25]", "node[26]", "node[27]", "node[28]", "node[29]".

The right side shows a 3D robot simulator window titled "Blender" with a scene of a red robot on a grassy hill. The right panel of the simulator shows a "Camera Control" panel with a list of keys and actions: "Display Help: Press the H key", "Camera controls:", "Hold left CTRL - Mouse - Look around", "W, S keys - Move Forward/backward", "A, D keys - Move to the sides", "R, F keys - Move up, down".

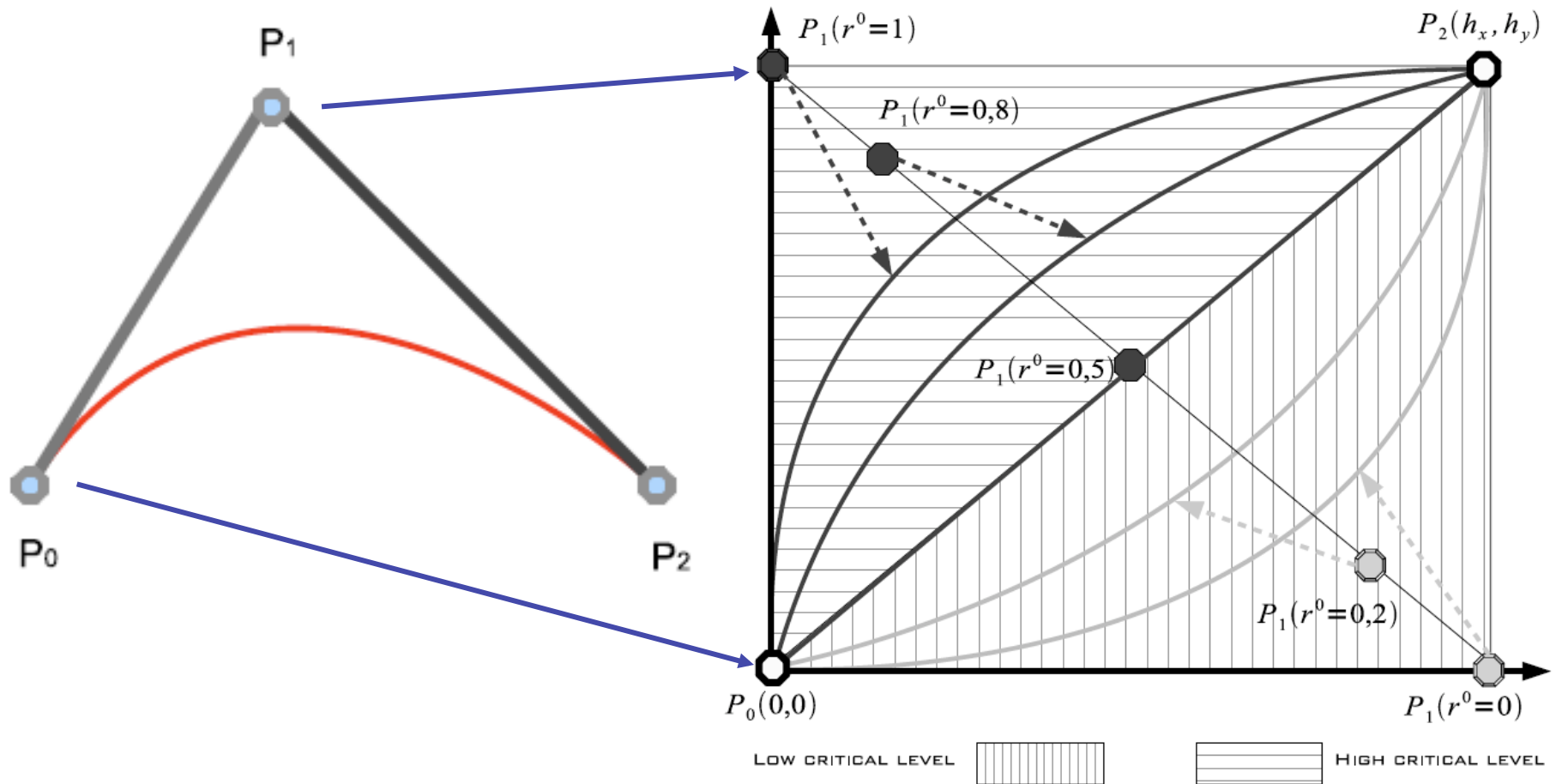
[

]

# ADDITIONAL SLIDES

# BEHAVIOR FUNCTION

$$B(t) = (1 - t)^2 * P_0 + 2t(1 - t) * P_1 + t^2 * P_2$$



# IMAGE SENSOR SIMULATION MODEL UNDER OMNET++

- ❑ COMMUNICATION LAYERS ARE VERY IMPORTANT FOR WSN
- ❑ USE SPECIFIC SIMULATOR

The image displays two screenshots from the OMNeT++ simulation environment. The left screenshot shows the internal structure of a node (SN.node[0]), including components like MobilityManager, ResourceManager, Application, SensorManager, and a CommunicationModule. The CommunicationModule contains Radio, MAC, and Routing layers. A small inset image shows a physical sensor node. The right screenshot shows a network topology with multiple nodes (node0 to node59) and a console window displaying simulation events and statistics. An orange callout box is overlaid on the right screenshot.

Need to know the power consumption for capturing an image, processing/compressing an image & transmitting an image...

# STUDY THE IMPACT OF COMMUNICATION LAYER ON SURVEILLANCE QUALITY

The image displays a simulation environment with three main components:

- Top Left:** A camera view window titled "79(33.8) <-46(1)" showing a desert landscape with a road and some vegetation.
- Top Middle:** A network map window titled "(SN) SN" showing a network of nodes (represented by colored circles) and their connections. The map includes a toolbar with "RUN", "STOP", and "Zoom: 0.79x".
- Right:** A terminal window titled "OMNeT++/Tkenv - SN" showing the simulation's output. The terminal displays various messages, including "Msgs created: 667040", "Msgs present: 1867", "insec/sec: 0.778365", and "Ev/simsec: 15059.8". The main output shows a series of "Application Sending [image] of size 288 bytes to communication layer" messages from various nodes (e.g., SN.node[46], SN.node[148], SN.node[5], SN.node[6], SN.node[124], SN.node[5], SN.node[24], SN.node[24], SN.node[6]).

```
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Sending [image] of size 288 bytes to communication layer
SN.node[46].Application Node 46 -> REAL IMAGE(1) to node 79
SN.node[96].Application Node 96: INTRUSION SEEN

SN.node[148].Application Node 148: INTRUSION SEEN

SN.node[148].Application Sending [alert] of size 30 bytes to communication layer
SN.node[148].Application Node 148: INTRUSION SEEN
pert.coverage 99.8628%
nb active nodes 100%

SN.node[5].Application Node 5: INTRUSION SEEN

SN.node[5].Application Sending [alert] of size 30 bytes to communication layer
SN.node[6].Application Node 6: INTRUSION SEEN
SN.node[6].Application Sending [alert] of size 30 bytes to communication layer
SN.node[5].Application Node 5: INTRUSION SEEN
SN.node[6].Application Node 6: INTRUSION SEEN
SN.node[124].Application Node 124: INTRUSION SEEN
SN.node[124].Application Sending [alert] of size 30 bytes to communication layer
SN.node[5].Application Node 5: INTRUSION SEEN
SN.node[24].Application Node 24: INTRUSION SEEN
SN.node[24].Application Sending [alert] of size 30 bytes to communication layer
SN.node[6].Application Node 6: INTRUSION SEEN

SN.node[79].Application Node 79: WRITES IMAGE FILE(1) from node 10
SN.node[79].Application Node 79: DISPLAY REAL IMAGE(1) from node 10
```

# ROBOT SIMULATORS

- ❑ MOBILITY, EXPLORATION, NAVIGATION, TRACKING, CONTROL AND DESIGN ARE VERY IMPORTANT FOR ROBOTS
- ❑ USE SPECIFIC ROBOT SIMULATORS

