

VIDEOSENSE: A SIMULATION MODEL OF IMAGE SENSORS UNDER OMNET++/CASTALIA

RESSACS 2012
JUNE, 7 TH, 2012
IUT BAYONNE, ANGLET



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



VIDEOSENSE: UN MODÈLE DE SIMULATION DE CAPTEURS VIDÉO SOUS OMNET++/CASTALIA

JOURNÉE RESSACS 2012
7 JUIN, 2012
IUT BAYONNE, ANGLET



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



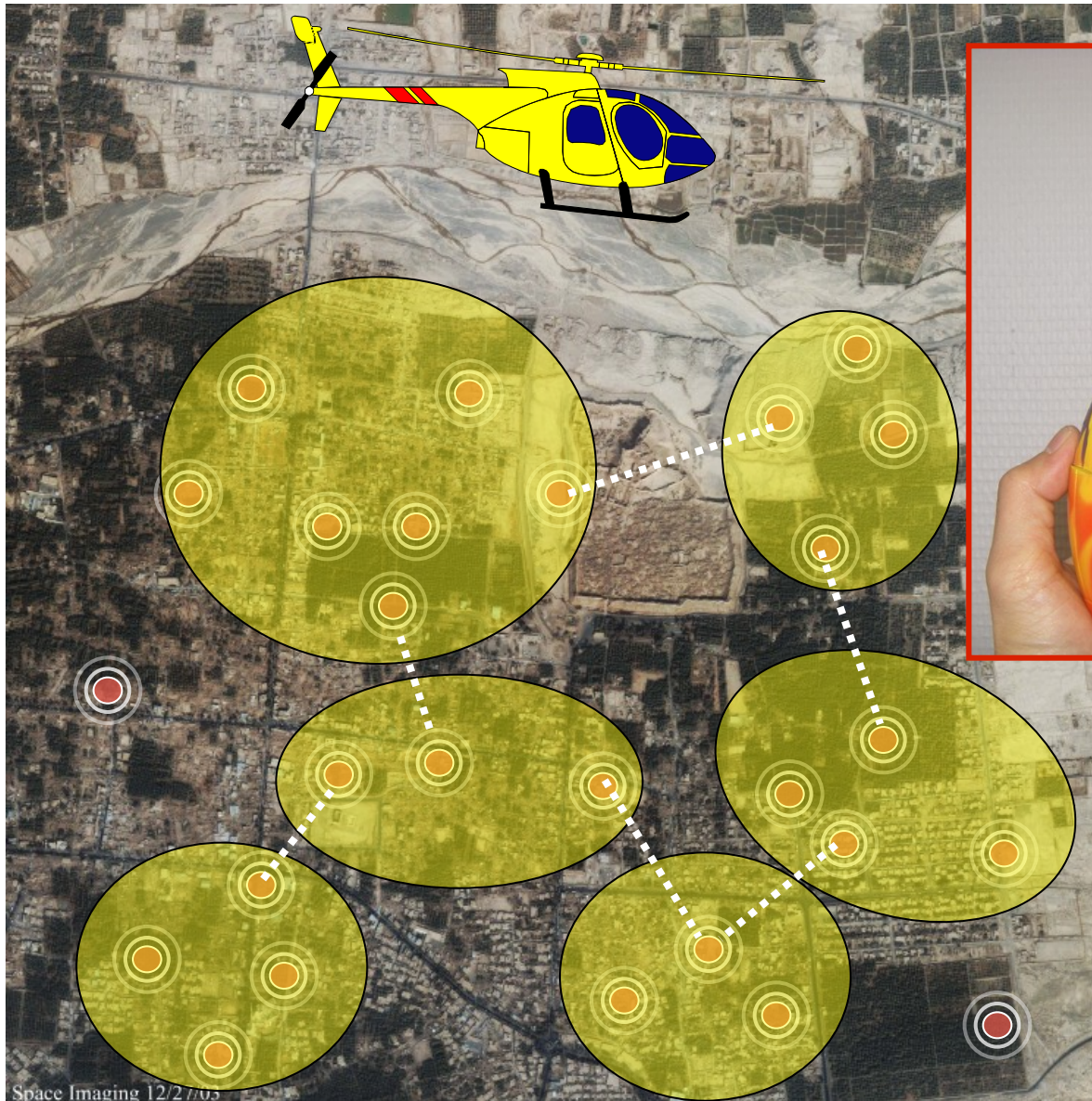
SEARCH & RESCUE



Imote2



Multimedia board

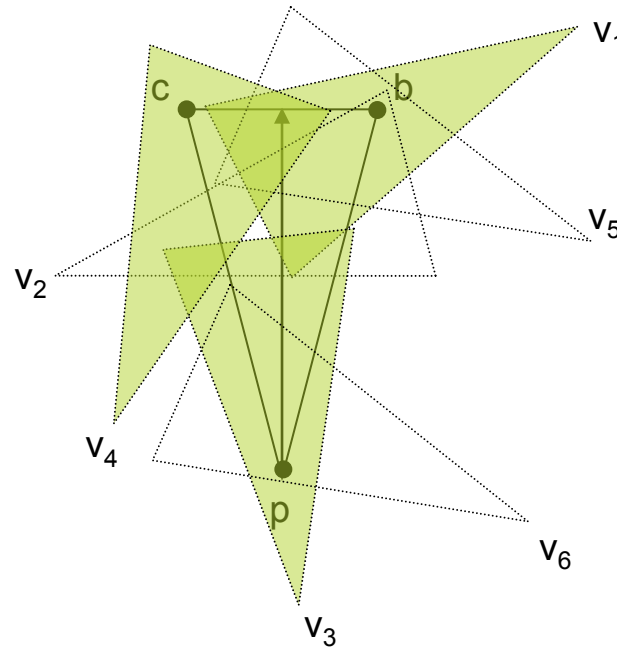


NODE'S COVER SET

$\text{Co}(V) = \{$
 $\{V\},$
 $\{V_1, V_3, V_4\},$
 $\{V_2, V_3, V_4\},$
 $\{V_3, V_4, V_5\},$
 $\{V_1, V_4, V_6\},$
 $\{V_2, V_4, V_6\},$
 $\{V_4, V_5, V_6\}$
 $\}$



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

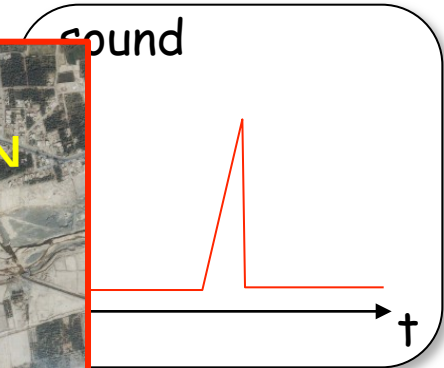
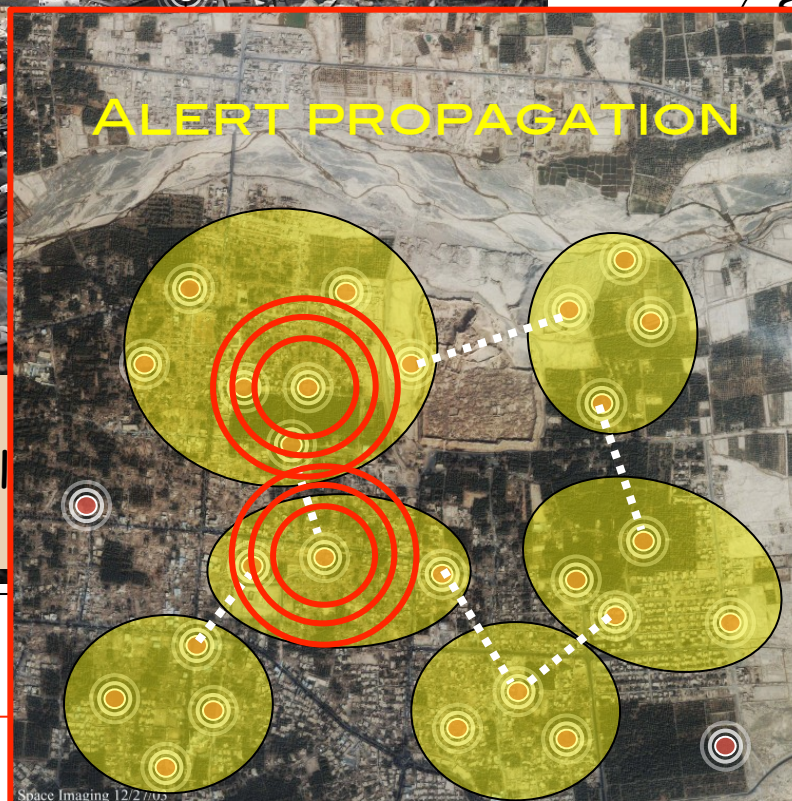


ACTIVITY DETECTION

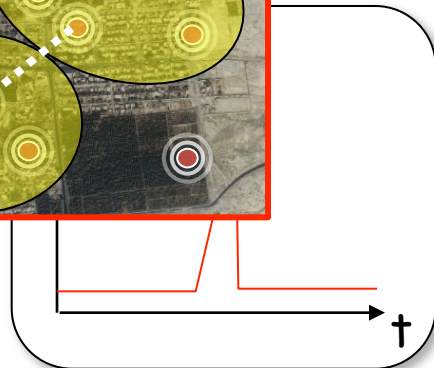
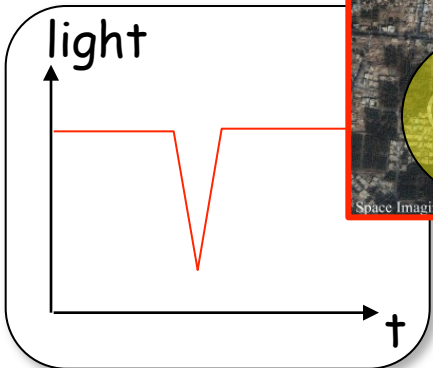


Sound could serve as detector if the person manifest itself

Light sensor could serve as detector if the person rapidly occults the sensor



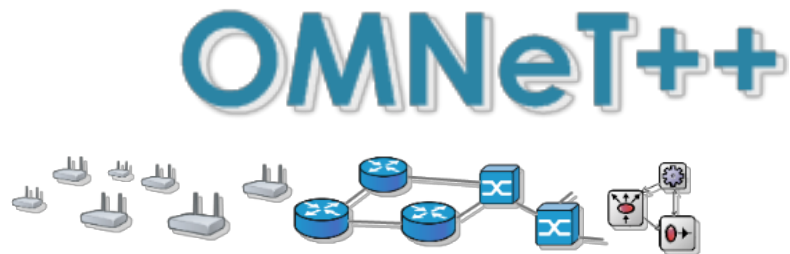
... as shaken



GET IMAGES FROM DEPLOYED SENSORS

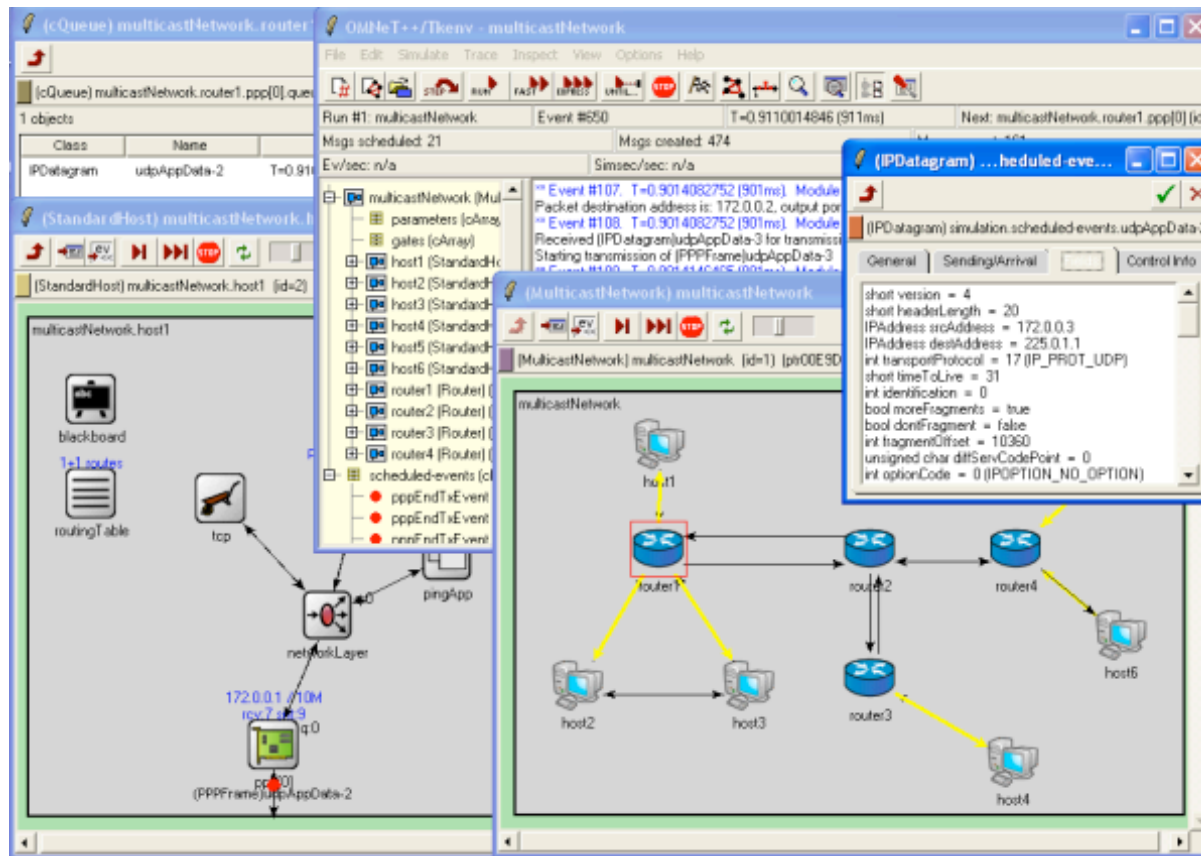


SIMULATION ENVIRONMENT



OMNET++

□ UNDERLYING SIMULATION FRAMEWORK, USE VERSION 4.1



Many packages for
simulating IP,
mobile, wireless,
adhoc,...

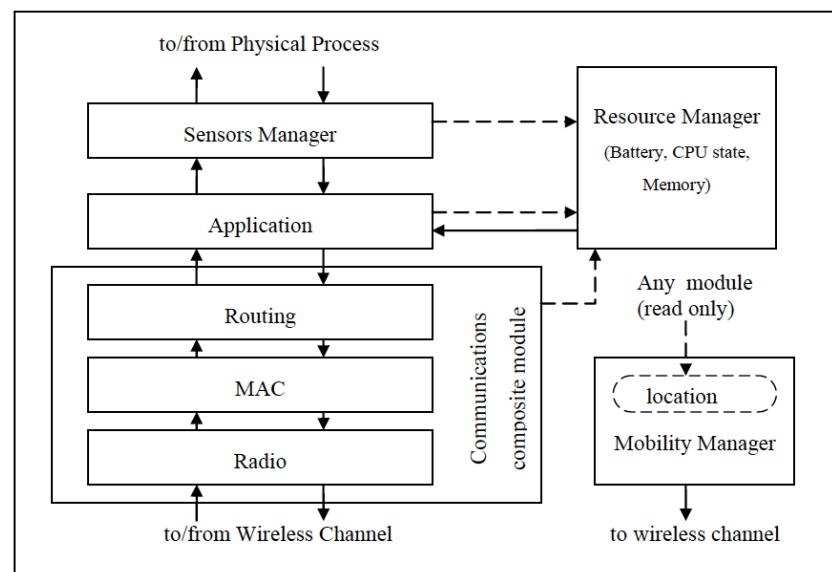
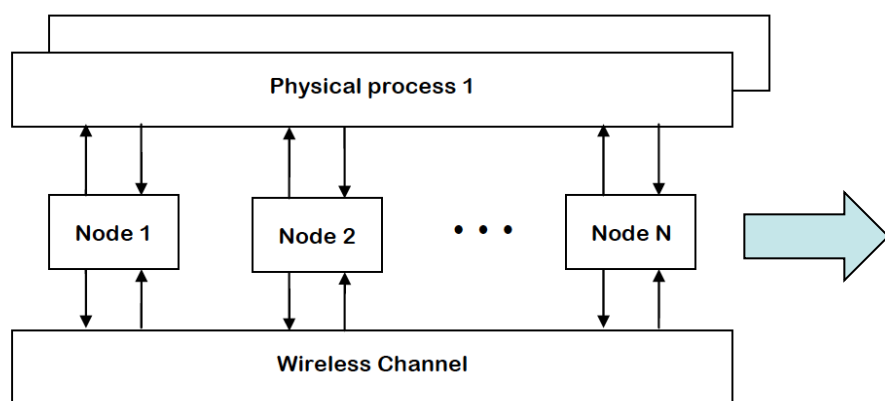
C++ based

Graphic interface



CASTALIA

- ❑ BAN AND WSN MODELS ON TOP OF OMNET++
- ❑ DEFINES WIRELESS ENVIRONMENT & SENSOR NODE ARCHITECTURE

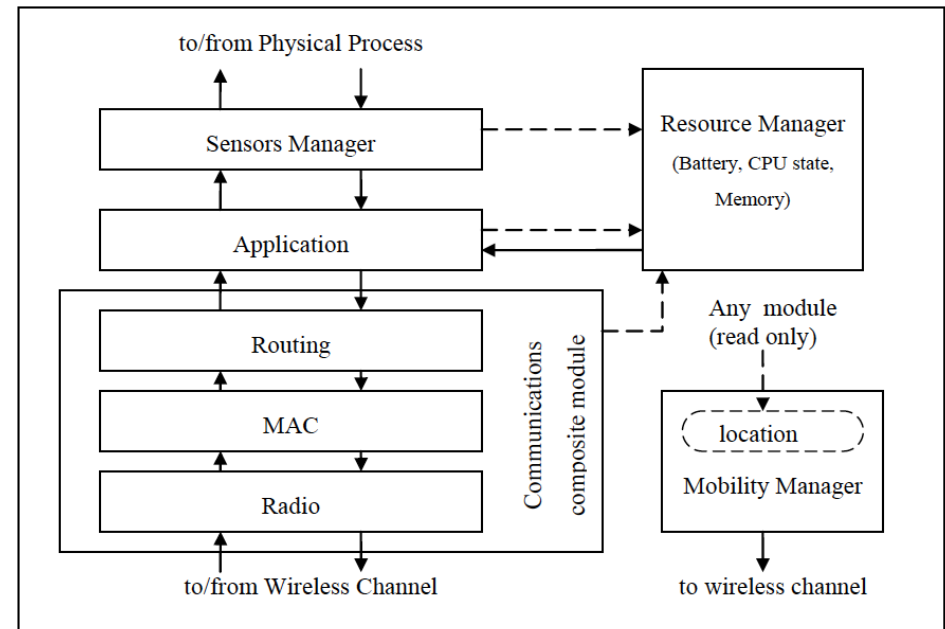


VIDEOSENSOR MODEL (1)

Coverset computation
Node scheduling
Intrusion detection
Alert propagation
Image transmission

Contrib:
- AODV
- LEACH
LIUPPA:
- GPSR

Castalia:
- CSMA (TunableMAC)
- SMAC, TMAC
- 802.15.4 MAC



VIDEOSENSOR MODEL (2)

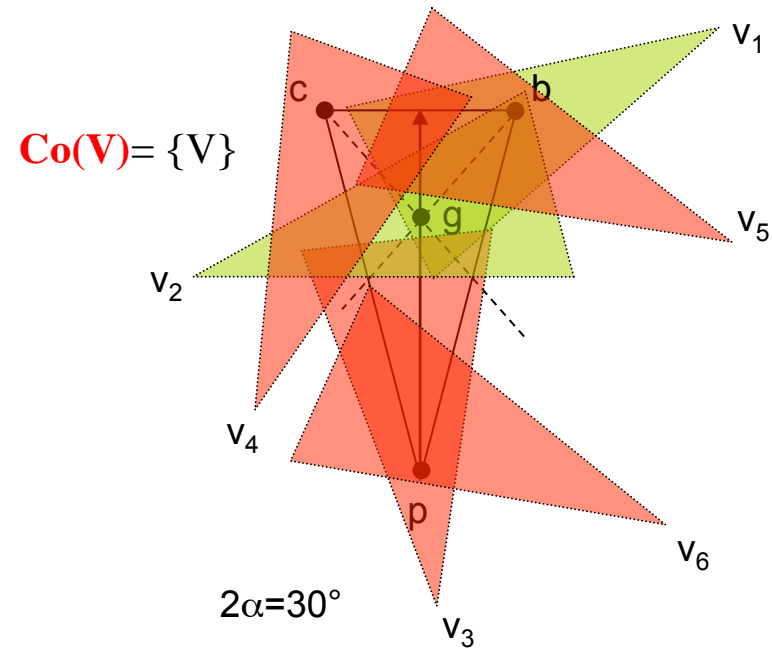
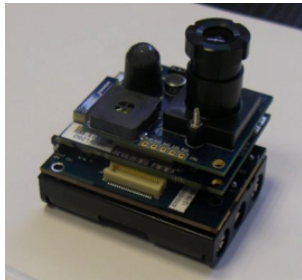
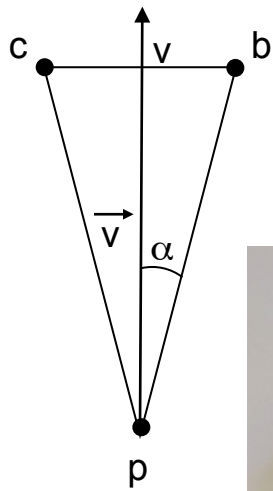
- ❑ MANY PARAMETERS CAN BE CHANGED WITHOUT RECOMPILING (OMNETPP.INI FILE)
- ❑ COMPILATION FLAGS ARE USED FOR SELECTING SOME GLOBAL BEHAVIORS

The image displays the OMNeT++ simulation environment for the VideoSensor model. It consists of three main windows:

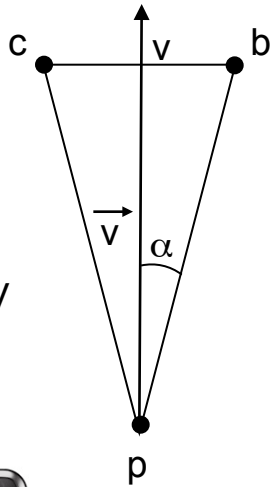
- (Node) SN.node[0]:** Shows the internal architecture of a sensor node. It includes a **MobilityManager** block, a **ResourceManager** block, an **Application** block, and a **SensorManager** block. A **CommunicationModule** window is also open, showing the internal components of the communication module: **Radio**, **MAC**, and **Routing**.
- OMNeT++/Tkenv - SN:** The main simulation window. The console shows the initialization of multiple nodes (SN.node54 to SN.node59) with various parameters like ID, position (px, py), and criticality level. It also shows the initialization of the **SN** module for each node.
- (SN) SN:** A network topology window showing a grid of nodes (node0 to node30) connected in a mesh-like structure. A **Coordinator** node is also visible.

An inset image in the bottom-left corner shows a physical hardware device, likely a video sensor or camera module, which is the real-world implementation of the model.

COVERSET COMPUTATION

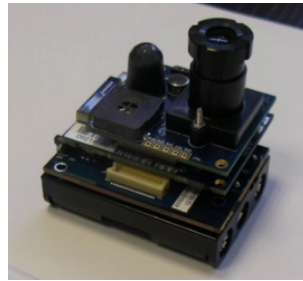


FINDING V'S COVER SET

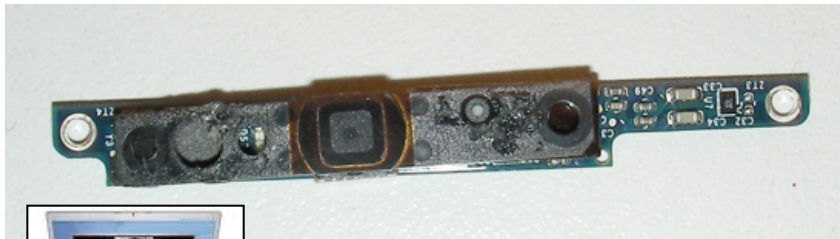


$$2\alpha = \text{AoV}$$

$$\text{AoV} = 20^\circ$$



$$\text{AoV} = 38^\circ$$



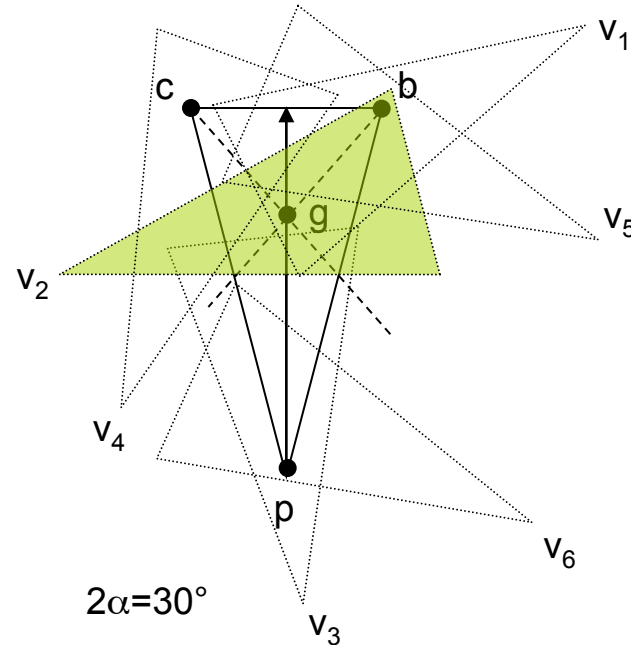
$$\text{AoV} = 31^\circ$$

$P = \{v \in N(V) : v \text{ covers the point "p" of the FoV}\}$

$B = \{v \in N(V) : v \text{ covers the point "b" of the FoV}\}$

$C = \{v \in N(V) : v \text{ covers the point "c" of the FoV}\}$

$G = \{v \in N(V) : v \text{ covers the point "g" of the FoV}\}$



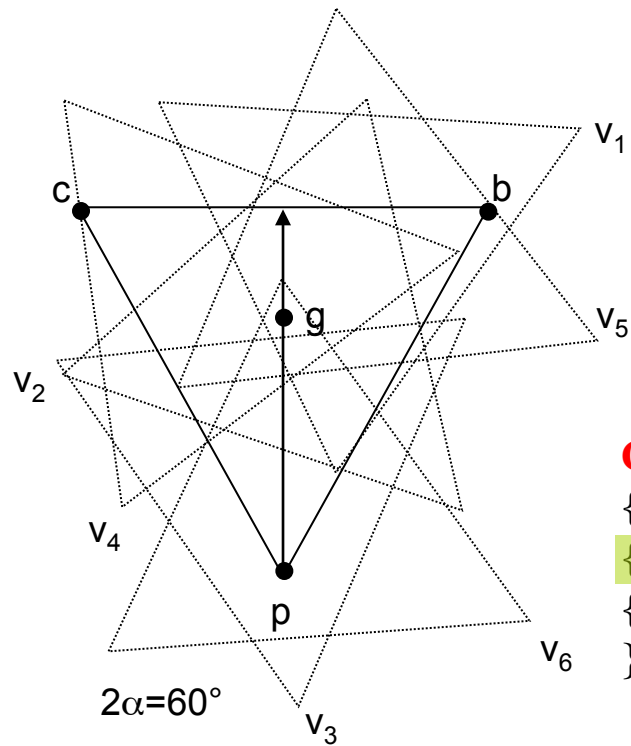
$$PG = \{P \cap G\}$$

$$BG = \{B \cap G\}$$

$$CG = \{C \cap G\}$$

$$Co(v) = PG \times BG \times CG$$

LARGE ANGLE OF VIEW



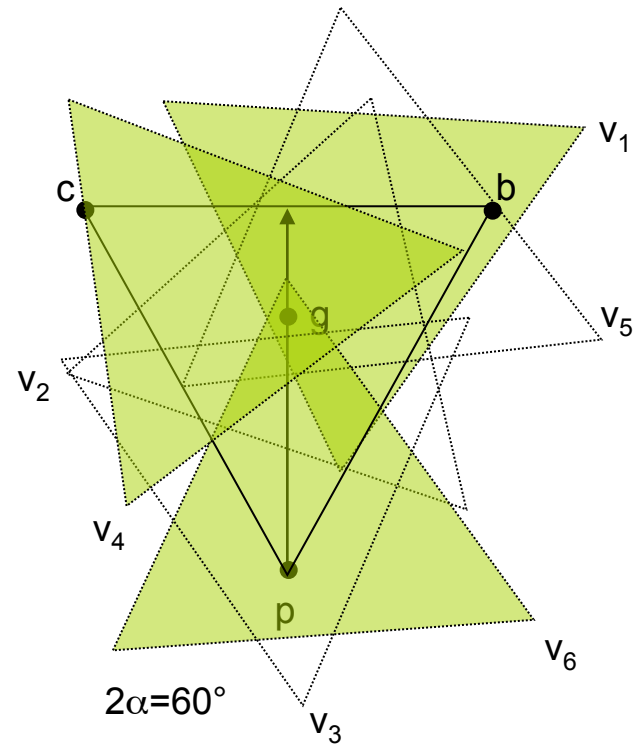
$$\text{Co}(\mathbf{V}) = \{$$

$$\{\mathbf{V}\},$$

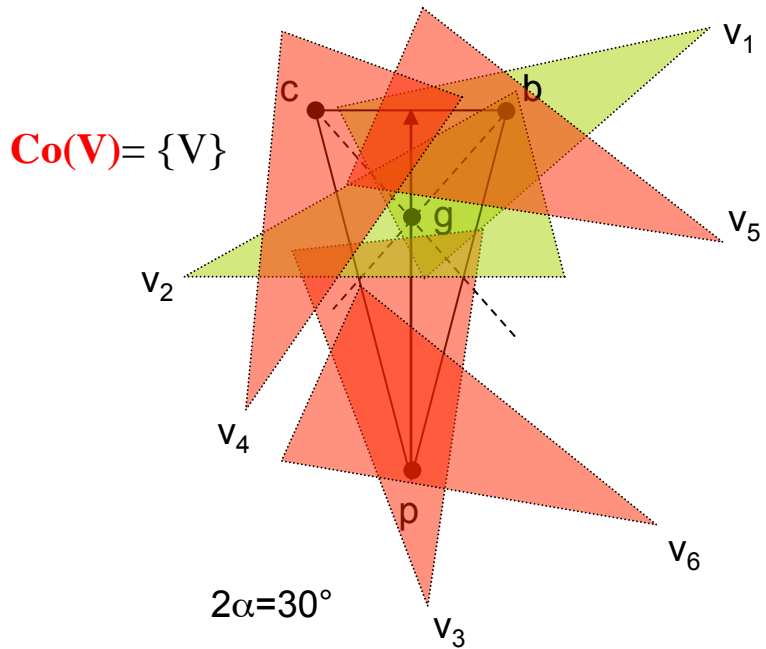
$$\{\mathbf{V}_1, \mathbf{V}_4, \mathbf{V}_6\},$$

$$\{\mathbf{V}_4, \mathbf{V}_5, \mathbf{V}_6\}$$

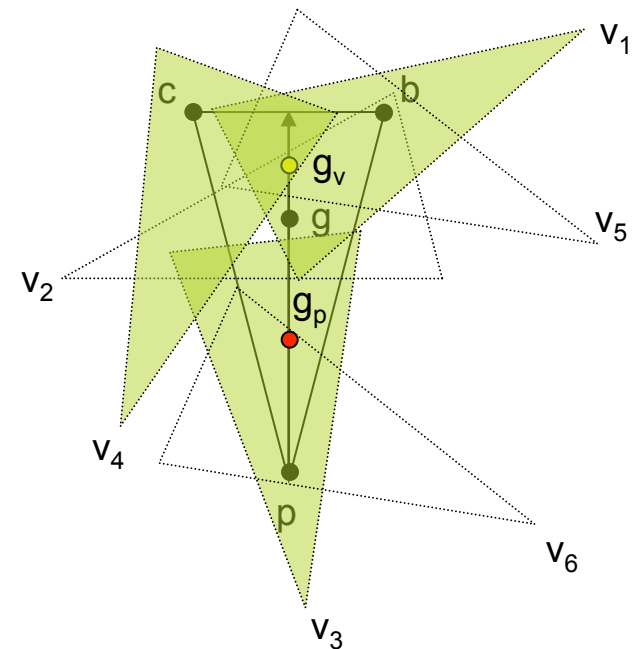
$$\}$$



SMALL ANGLE OF VIEW

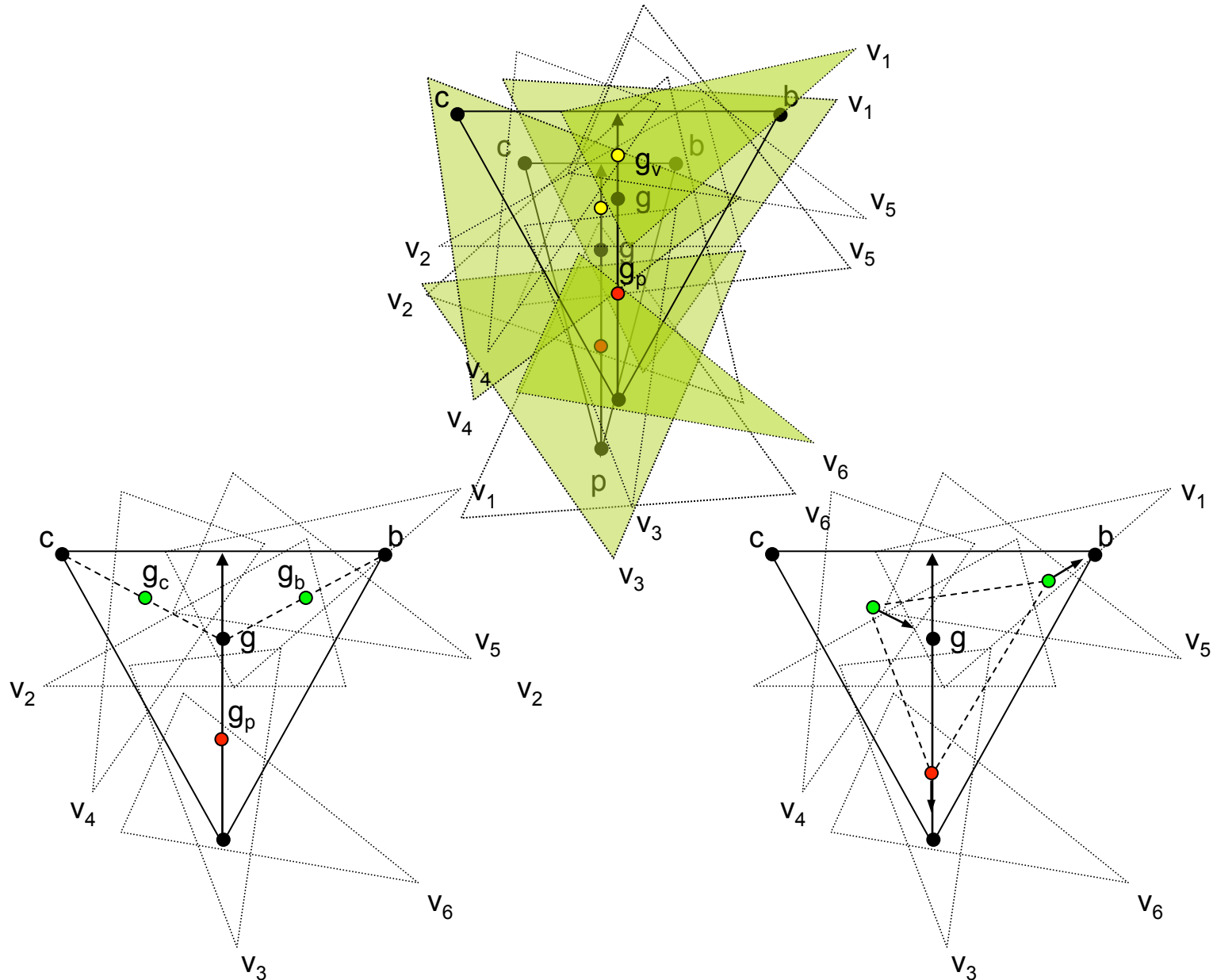


$\text{Co}(\mathbf{V}) = \{$
 $\{\mathbf{V}\},$
 $\{V_1, V_3, V_4\},$
 $\{V_2, V_3, V_4\},$
 $\{V_3, V_4, V_5\},$
 $\{V_1, V_4, V_6\},$
 $\{V_2, V_4, V_6\},$
 $\{V_4, V_5, V_6\}$
 $\}$

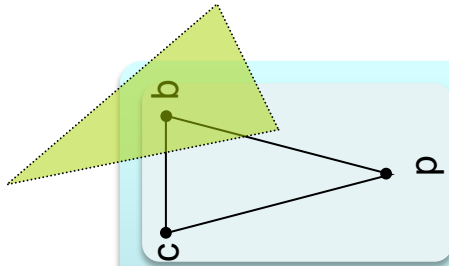


$PG = \{P \cap g_p\}$
 $BG = \{B \cap g_v\}$
 $CG = \{C \cap g_v\}$
 $\text{Co}(v) = PG \times BG \times CG$

HETEROGENEOUS AOV

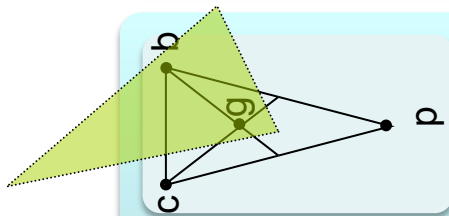


COMPARISON OF COVERSET STRATEGIES



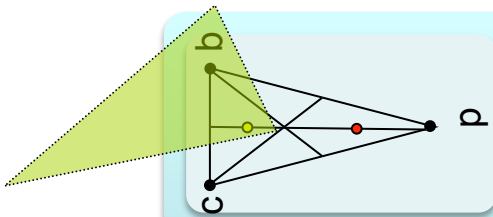
COV_{woG}

- Only triangle's points, without point G



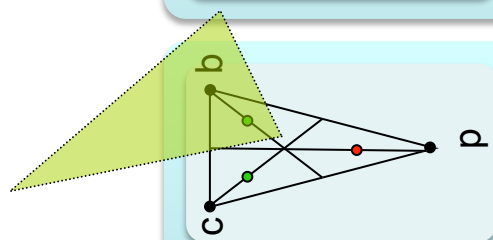
COV_{wG}

- Triangle points, with point G



COV_{waGpv}

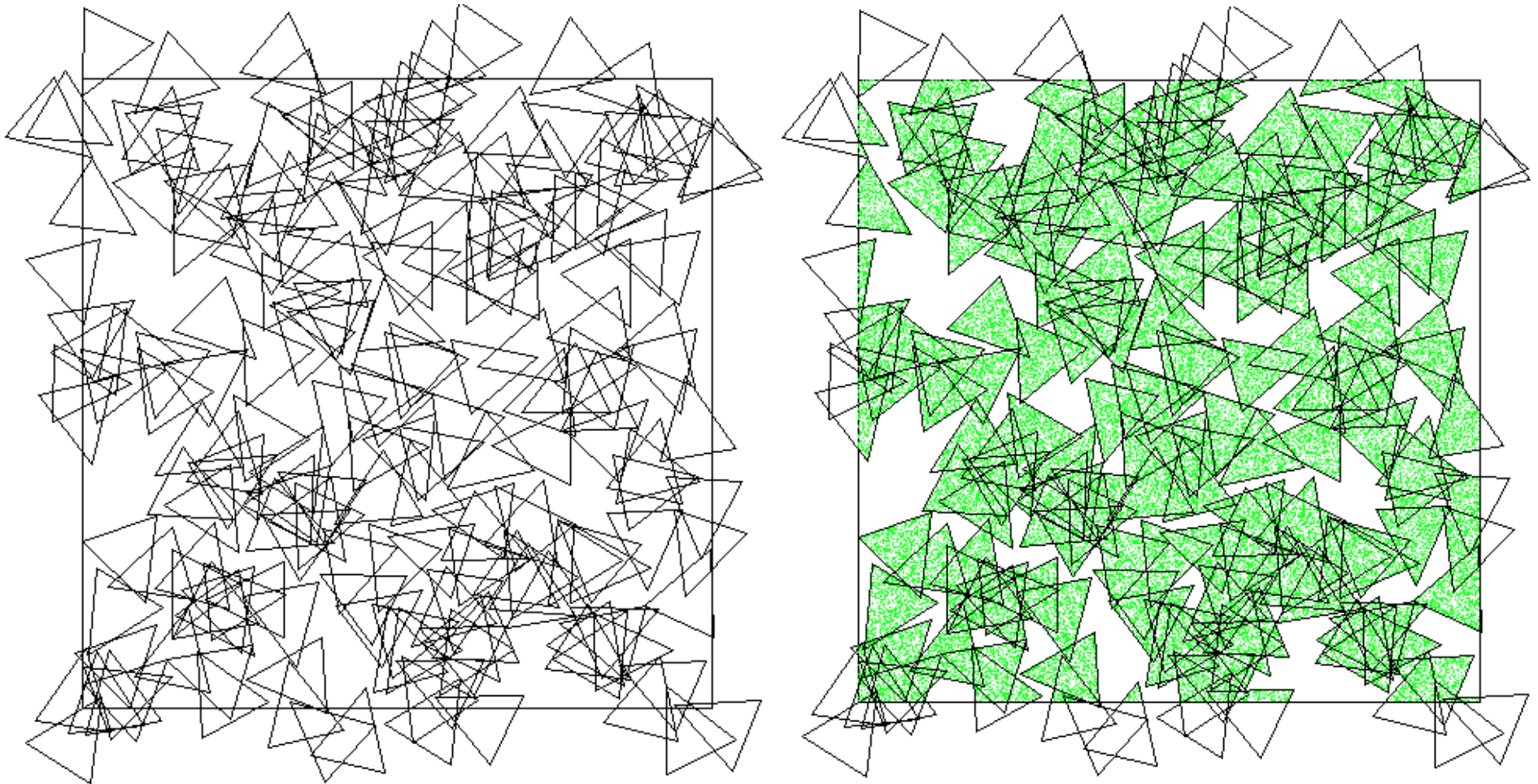
- Triangle points, with alternate gp & gv



COV_{waGbc}

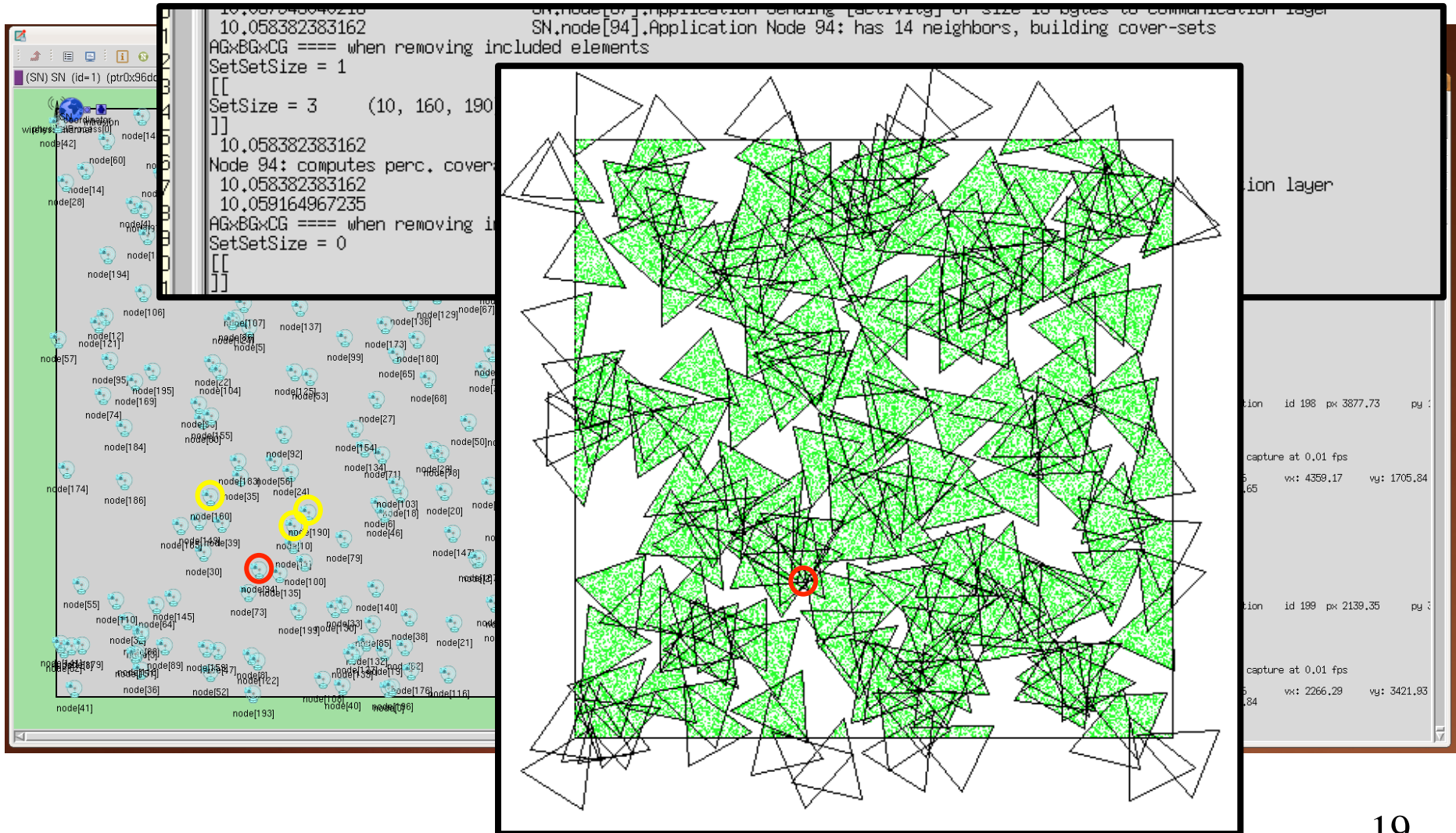
- Triangle points, with alternate gp , gb & gc

NODE FOV VISUALIZATION

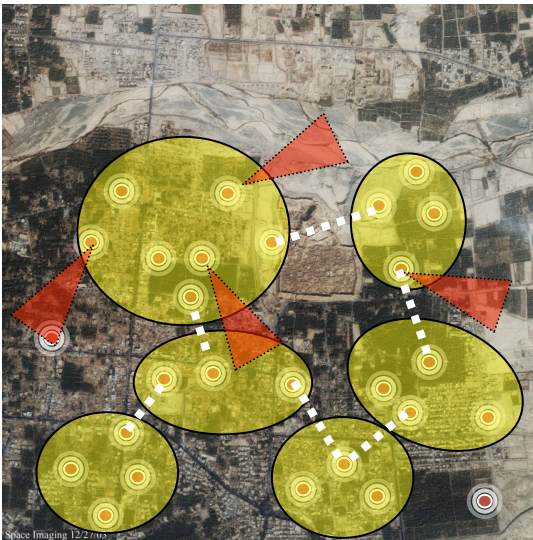


200 sensor nodes in an 400mx400m area

START SIMULATION



SCHEDULING IMAGE SENSORS



FIRST OF ALL: DON'T MISS IMPORTANT EVENTS!



WHOLE
UNDERSTANDING
OF THE SCENE IS
WRONG!!!

WHAT IS CAPTURED

HOW TO MEET SURVEILLANCE APP'S CRITICALITY

- ❑ CAPTURE SPEED CAN BE A « QUALITY » PARAMETER
- ❑ CAPTURE SPEED FOR NODE V SHOULD DEPEND ON THE APP'S CRITICALITY AND ON THE LEVEL OF REDUNDANCY FOR NODE V
- ❑ V 'S CAPTURE SPEED CAN INCREASE WHEN AS V HAS MORE NODES COVERING ITS OWN FOV - COVER SET

CRITICALITY MODEL (1)

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

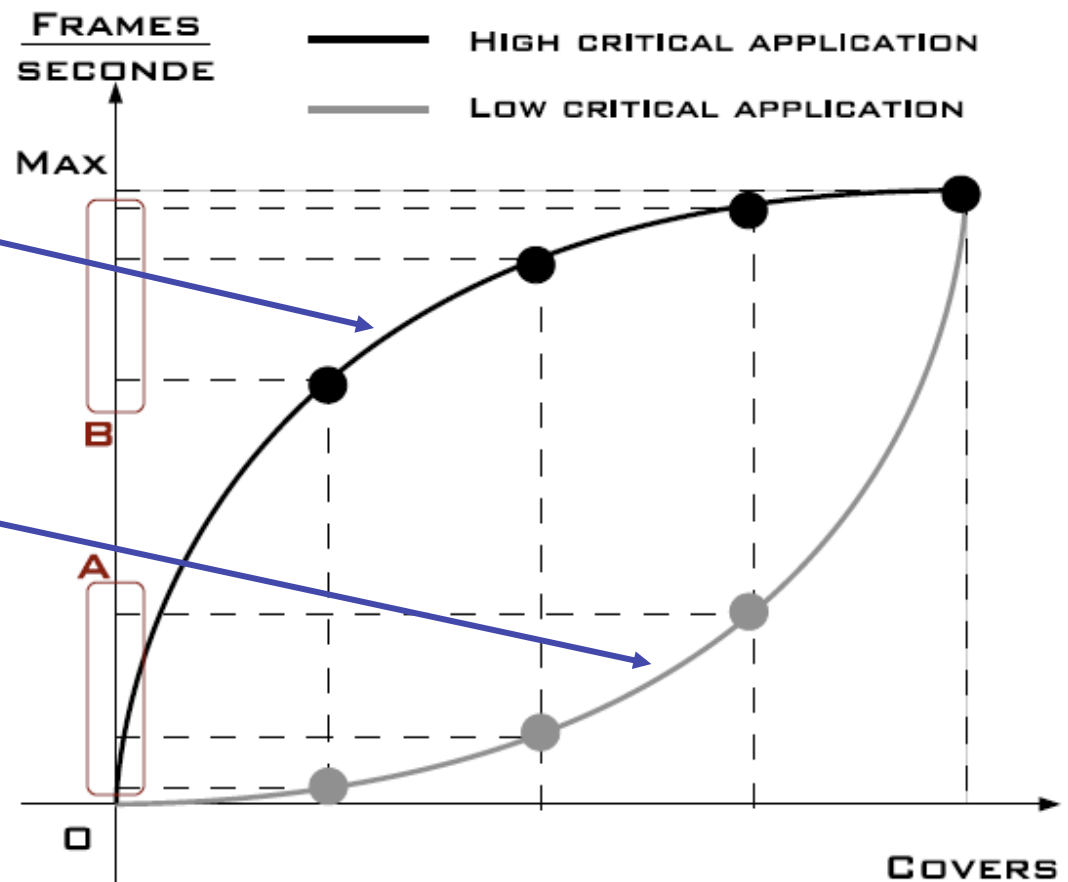
- HIGH CRITICALITY

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

- LOW CRITICALITY

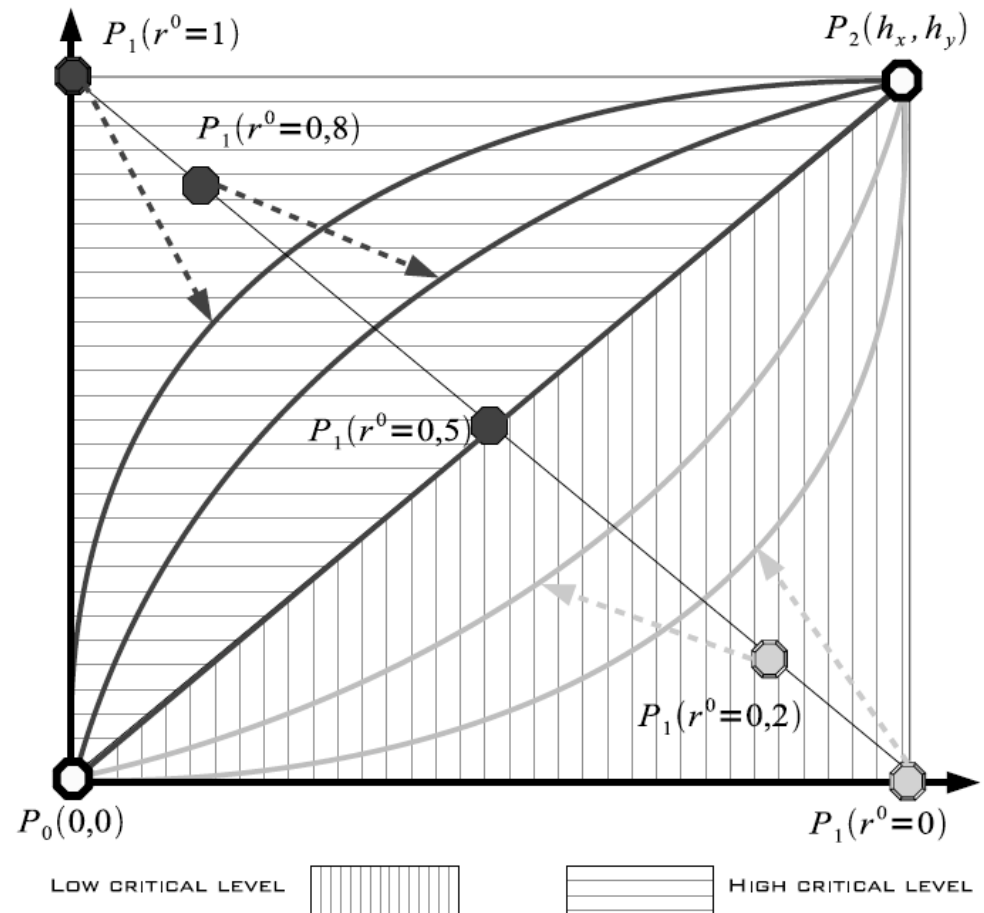
- CONCAVE SHAPE
- MOST PROJECTIONS OF X ARE CLOSE TO THE MIN CAPTURE SPEED

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



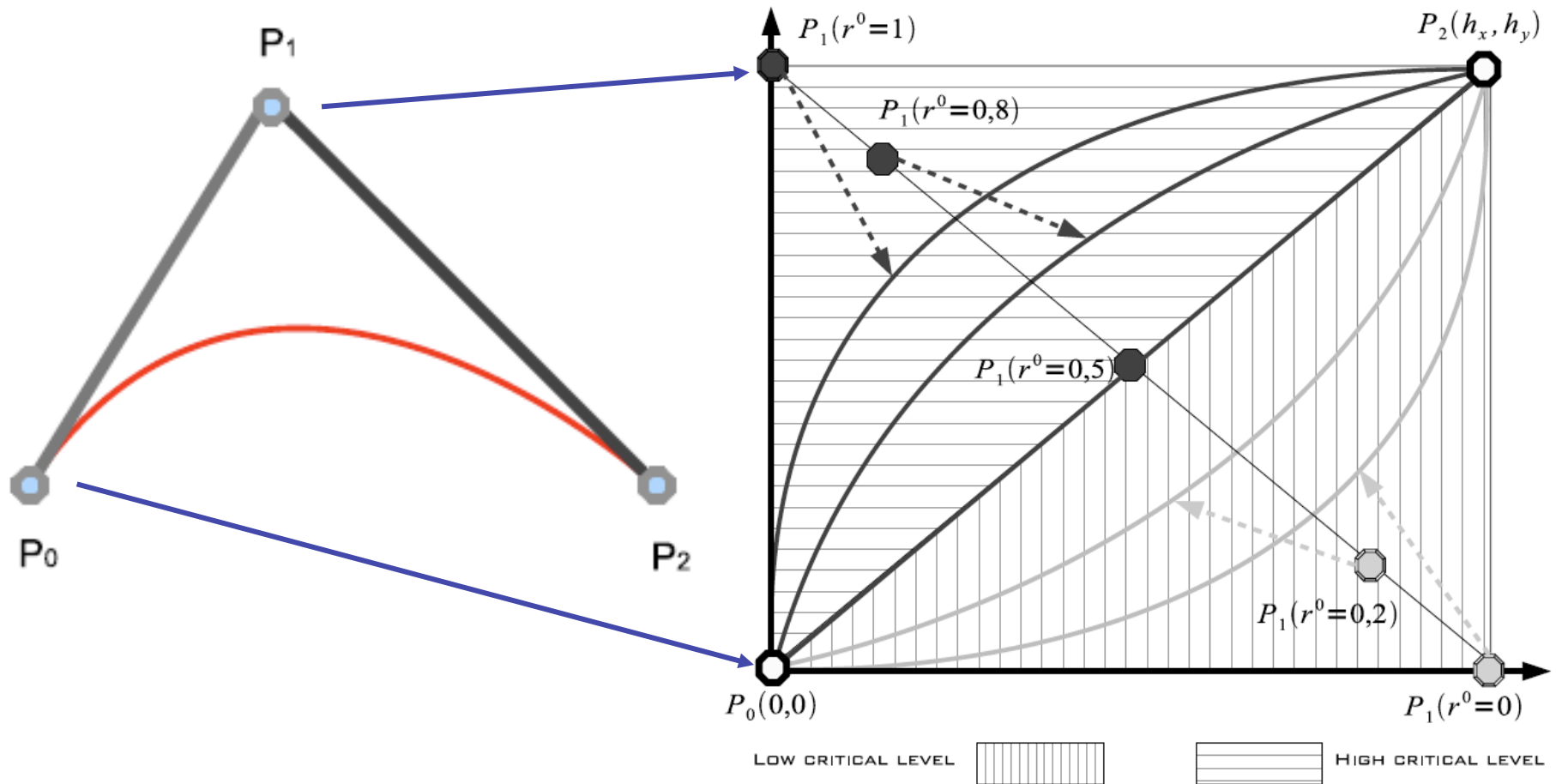
CRITICALITY MODEL (2)

- ❑ 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



BEHAVIOR FUNCTION

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



SOME TYPICAL CAPTURE SPEED

- MAXIMUM CAPTURE SPEED IS 6FPS OR 12FPS
- NODES WITH SIZE OF COVER SET GREATER THAN N CAPTURE AT THE MAXIMUM SPEED

N=6
P₂(6,6)

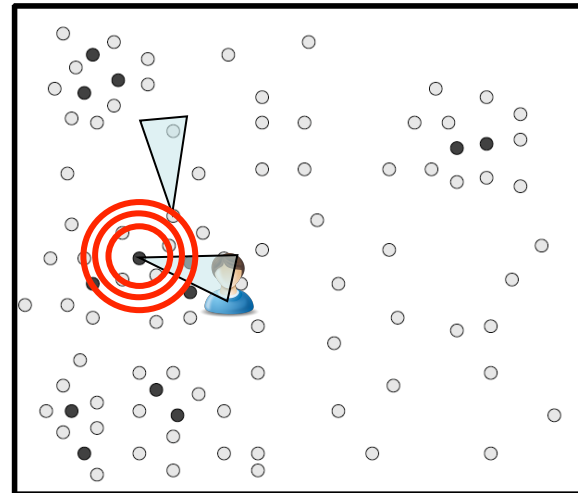
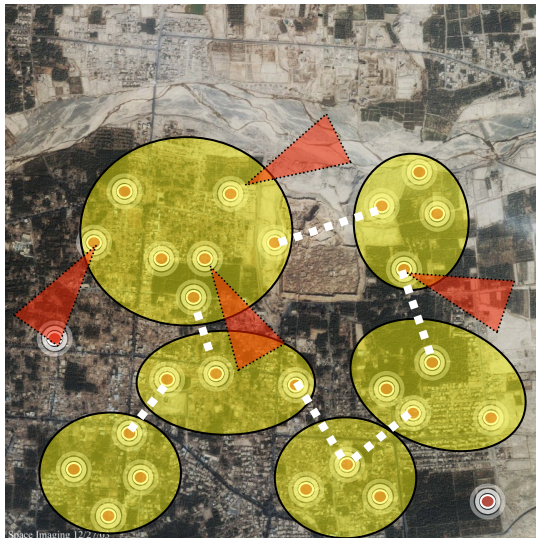
$r^0 \backslash Co(v) $	1	2	3	4	5	6
0.0	0.05	0.20	0.51	1.07	2.10	6.00
0.2	0.30	0.73	1.34	2.20	3.52	6.00
0.5	1.00	2.00	3.00	4.00	5.00	6.00
0.8	2.48	3.80	4.66	5.27	5.70	6.00
1.0	3.90	4.93	5.49	5.80	5.95	6.00

N=12
P₂(12,3)

r^0	1	2	3	4	5	6	7	8	9	10	11	12
0	.01	.02	.05	0.1	.17	.26	.38	.54	.75	1.1	1.5	3
.2	.07	.15	.25	.37	.51	.67	.86	1.1	1.4	1.7	2.2	3
.4	.17	.35	.55	.75	.97	1.2	1.4	1.7	2.0	2.3	2.6	3
.6	.36	.69	1.0	1.3	1.5	1.8	2.0	2.2	2.4	2.6	2.8	3
.8	.75	1.2	1.6	1.9	2.1	2.3	2.5	2.6	2.7	2.8	2.9	3
1	1.5	1.9	2.2	2.4	2.6	2.7	2.8	2.9	2.9	2.9	2	3



INTRUSION DETECTION



RANDOM INTRUSION

The screenshot displays the OMNeT++/TKenv simulation environment for a network named 'SN'. The main window shows a network topology with numerous nodes (represented by green circles) and a central Coordinator node (blue circle). A red circle highlights a specific node in the network. The console window at the bottom shows the following log entries:

```

Run #0: SN                               Event #664832                               T=51.005914493073                               Next: SN.node[33].Communication.F
Msgs scheduled: 621                               Msgs created: 659912                               Msgs present: 2125
Ev/sec: n/a                                       Simsec/sec: n/a                                       Ev/simsec: n/a
WC-BEGIN... TunableMac data packet... continueTXPeriodic energy calculation,Periodic energy calculation,... displaystatscapture,...
WC-BEGIN... TunableMac data packet... WC-ENDPeriodic energy calculation,Periodic energy calculation... capture capture,...

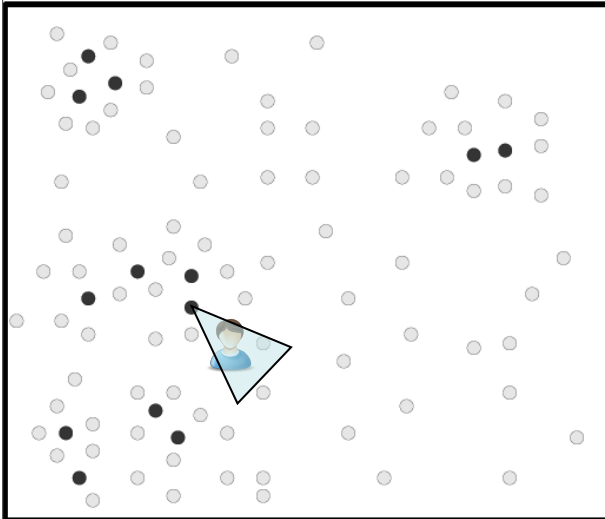
node[63] (Node) (id=69) 40.540381585414 SN.node[2].Application Sending [Image] of size 86 bytes to communication layer
node[64] (Node) (id=70) 40.540381585414 SN.node[2].Application Sending [Image] of size 51 bytes to communication layer
node[65] (Node) (id=71) 40.540381585414 SN.node[2].Application Node 2 -> REAL IMAGE(0) to node 54. #packet: 206, stored #packets: 20
node[66] (Node) (id=72) 40.5929374939315 SN.node[54].Application Node 54: WRITES IMAGE FILE(0) from node 147. Size=1082,R54(40.6)-147(0)
node[67] (Node) (id=73) 40.5929374939315 SN.node[54].Application Node 54: DISPLAY REAL IMAGE(0) from node 147. 20/206. First packet rec
node[68] (Node) (id=74) 41.169515847965 SN.node[172].Application Node 172: INTRUSION SEEN
node[69] (Node) (id=75) 41.169515847965 SN.node[172].Application Sending [alert] of size 30 bytes to communication layer
node[70] (Node) (id=76) Node 172: has active neighbors:
node[71] (Node) (id=77) SetSize = 11 (43, 48, 58, 77, 83, 146, 157, 168, 181, 182, 197)
node[72] (Node) (id=78) Node 172: updates its active coverset list
node[73] (Node) (id=79) 41.169515847965 SN.node[172].Application Node 172: CANNOT ACTIVATE COVERSET, NO ACTIVE COVERSET
node[74] (Node) (id=80) 41.169515847965 SN.node[172].Application Node 172: SEND INTRUSION IMAGE
node[75] (Node) (id=81) 41.169515847965 SN.node[172].Application Node 172: 0 INTRUSION IMAGE(S) REMAINING
node[76] (Node) (id=82) 41.169515847965 SN.node[172].Application Sending [Image] of size 94 bytes to communication layer
node[77] (Node) (id=83) 41.169515847965 SN.node[172].Application Sending [Image] of size 91 bytes to communication layer
node[78] (Node) (id=84) 41.169515847965 SN.node[172].Application Sending [Image] of size 89 bytes to communication layer
node[79] (Node) (id=85) 41.169515847965 SN.node[172].Application Sending [Image] of size 77 bytes to communication layer
node[80] (Node) (id=86) 41.169515847965 SN.node[172].Application Sending [Image] of size 83 bytes to communication layer
node[81] (Node) (id=87) 41.169515847965 SN.node[172].Application Sending [Image] of size 88 bytes to communication layer
node[82] (Node) (id=88) 41.169515847965 SN.node[172].Application Sending [Image] of size 81 bytes to communication layer

```

INTRUSION DETECTION SCENARIO

(A)

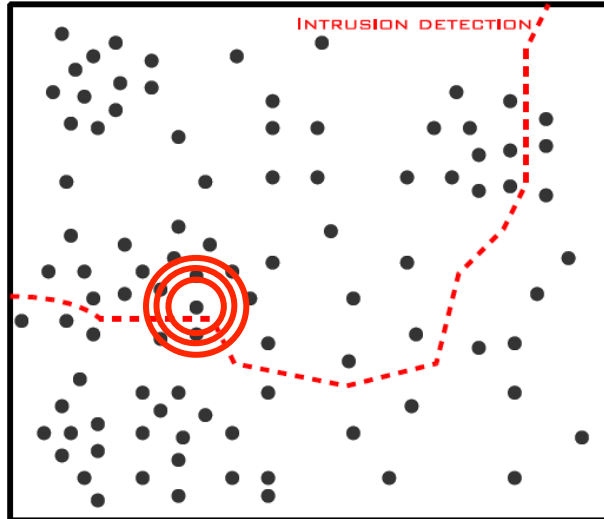
- SENTRY NODE: NODE WITH HIGH SPEED CAPTURE (HIGH COVER SET).
- SLEEP NODE: NODE WITH LOW SPEED CAPTURE.



HIBERNATE MODE
 $r^o = 0$

(B)

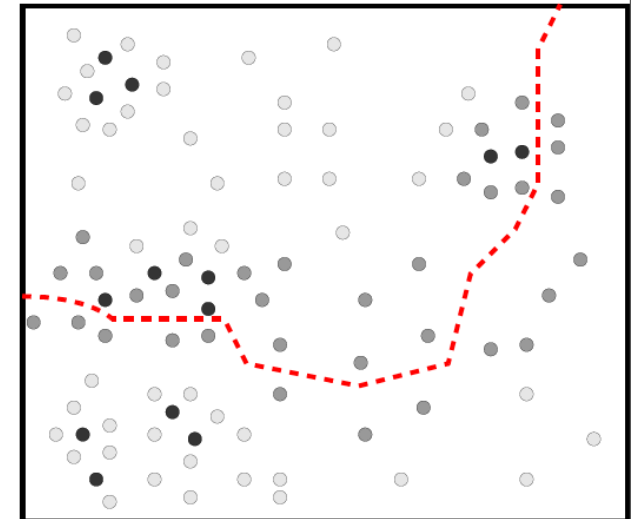
- ALERTED NODE: NODE WITH HIGH SPEED CAPTURE (ALERT INTRUSION).



ALERT MODE
 $r^o = \text{MAX}$

(C)

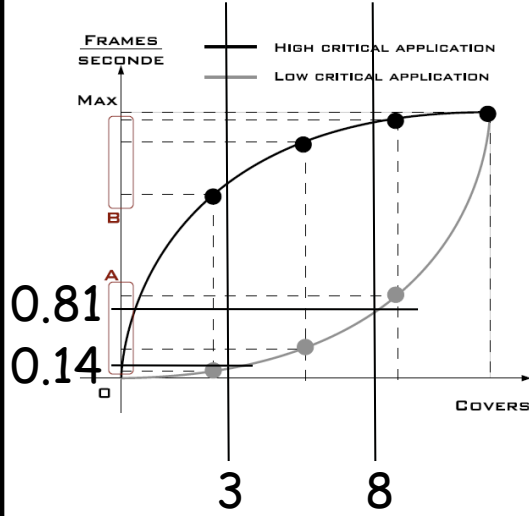
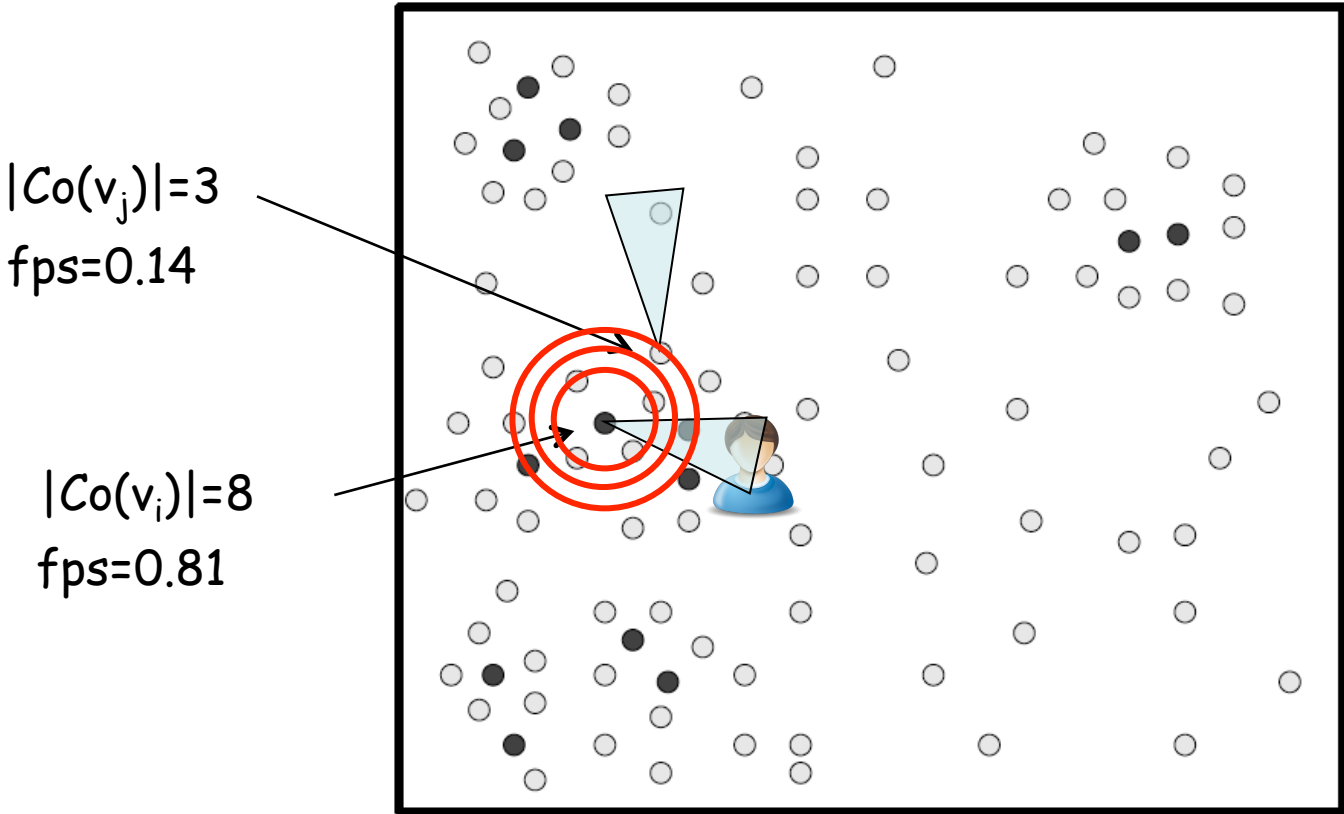
- SENTRY NODE: NODE WITH HIGH SPEED CAPTURE (HIGH COVER SET).
- CRITICAL NODE: NODE WITH HIGH SPEED CAPTURE (NODE THAT DETECTS THE INTRUSION).
- SLEEP NODE: NODE WITH LOW SPEED CAPTURE.



HIBERNATE MODE (AFTER INTRUSION)
 $r^o = 0$

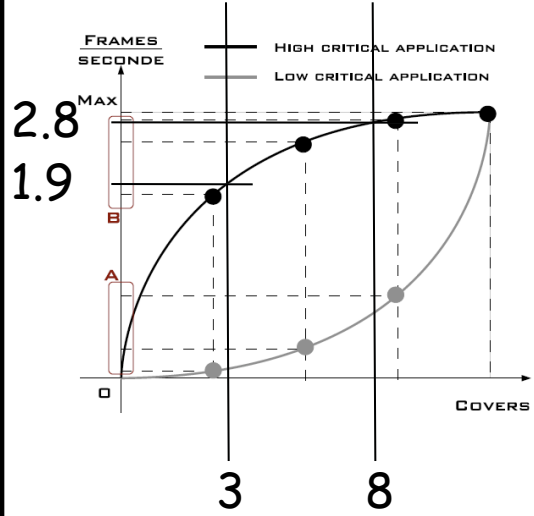
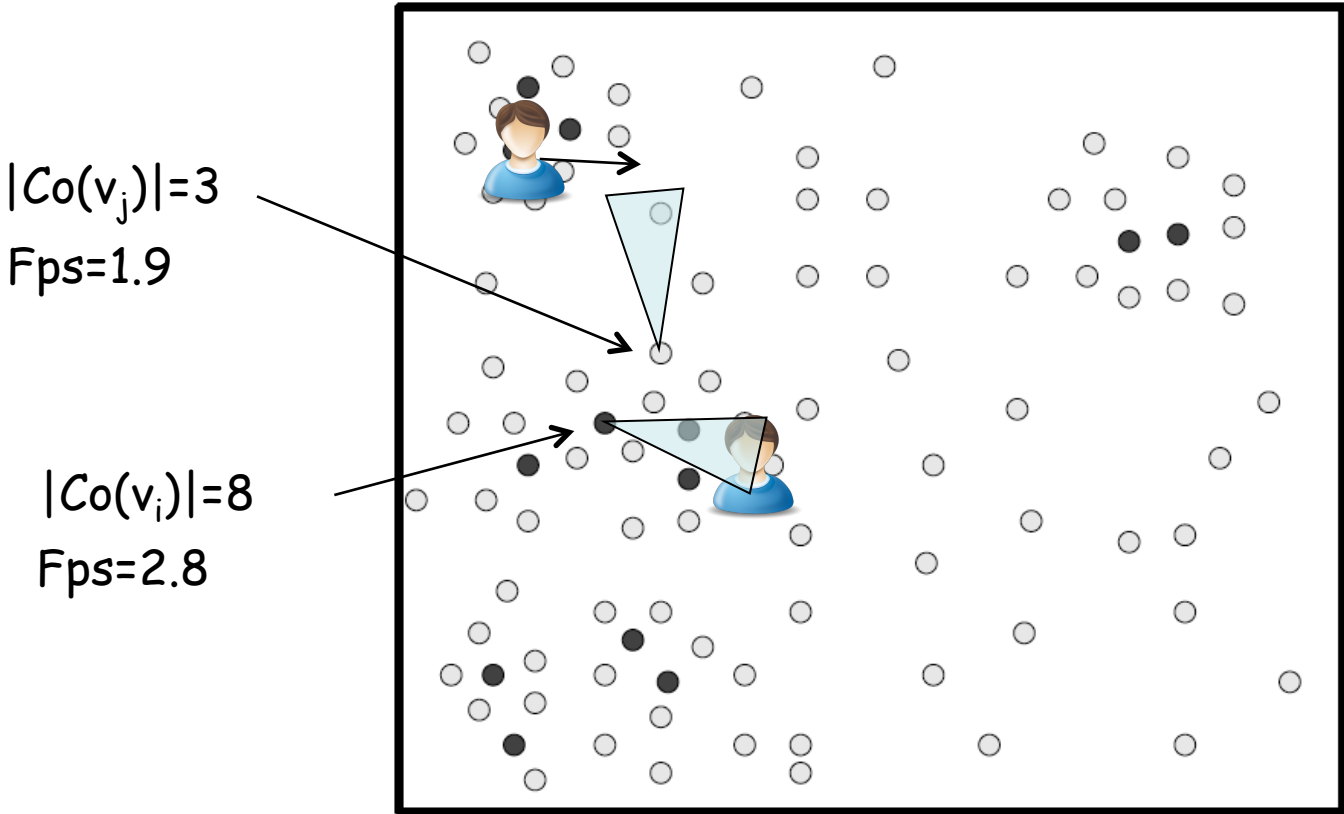
RISK-BASED SCHEDULING IN IMAGES (1)

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



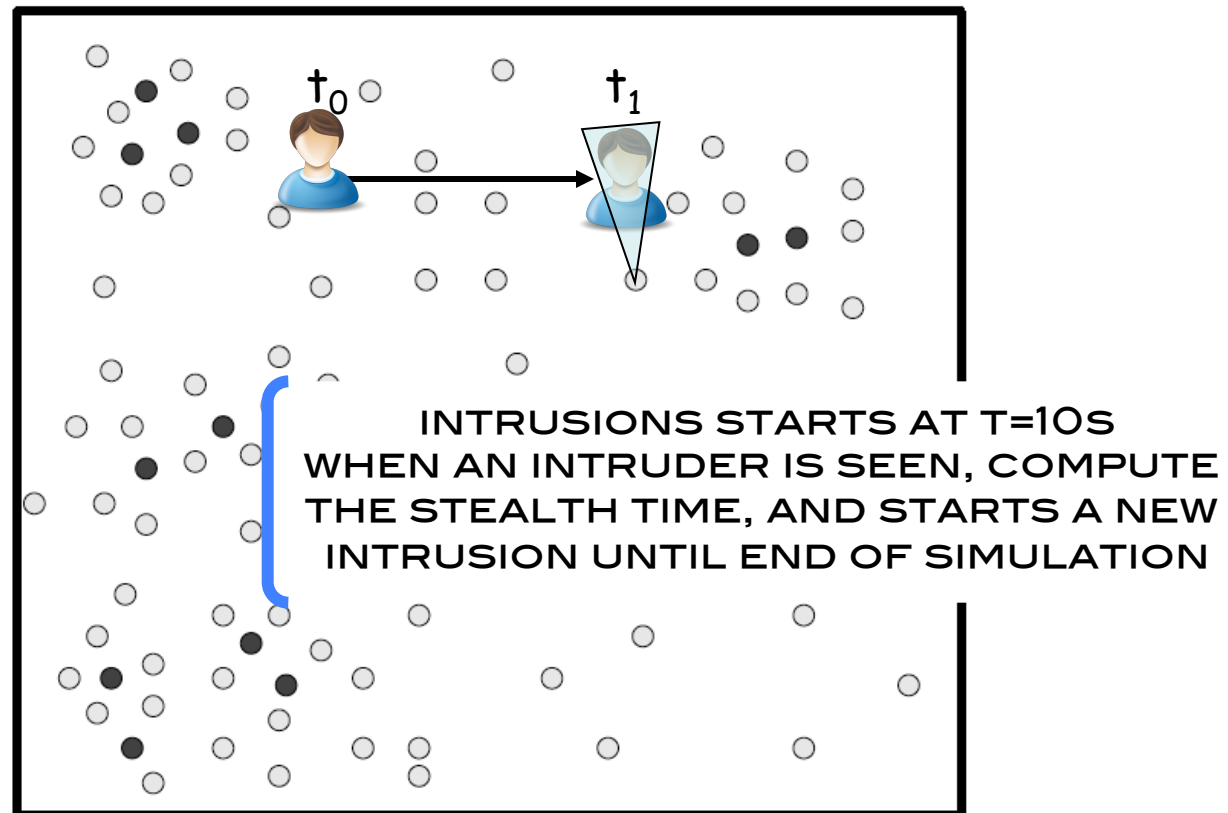
RISK-BASED SCHEDULING IN IMAGES (2)

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

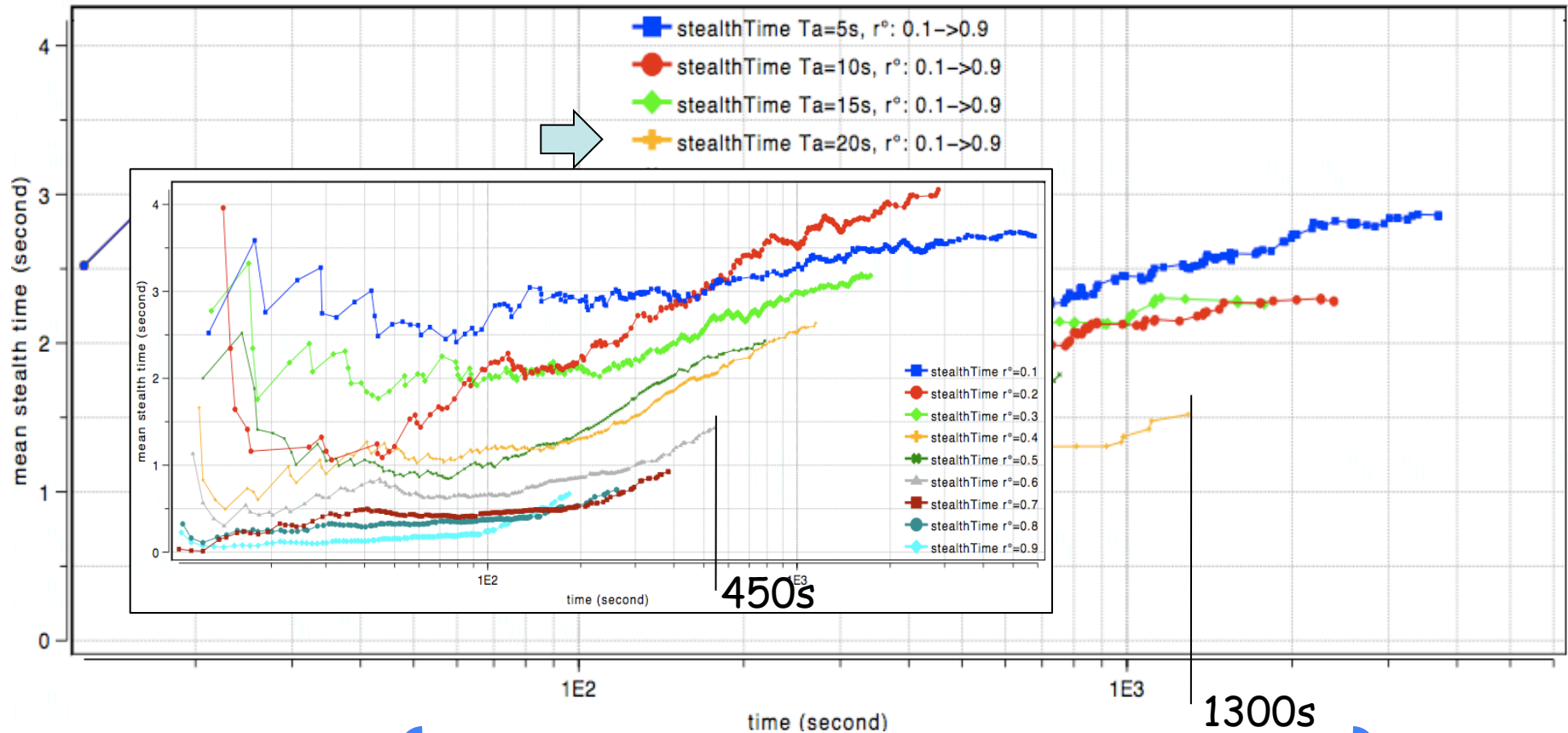


MEAN STEALTH TIME

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

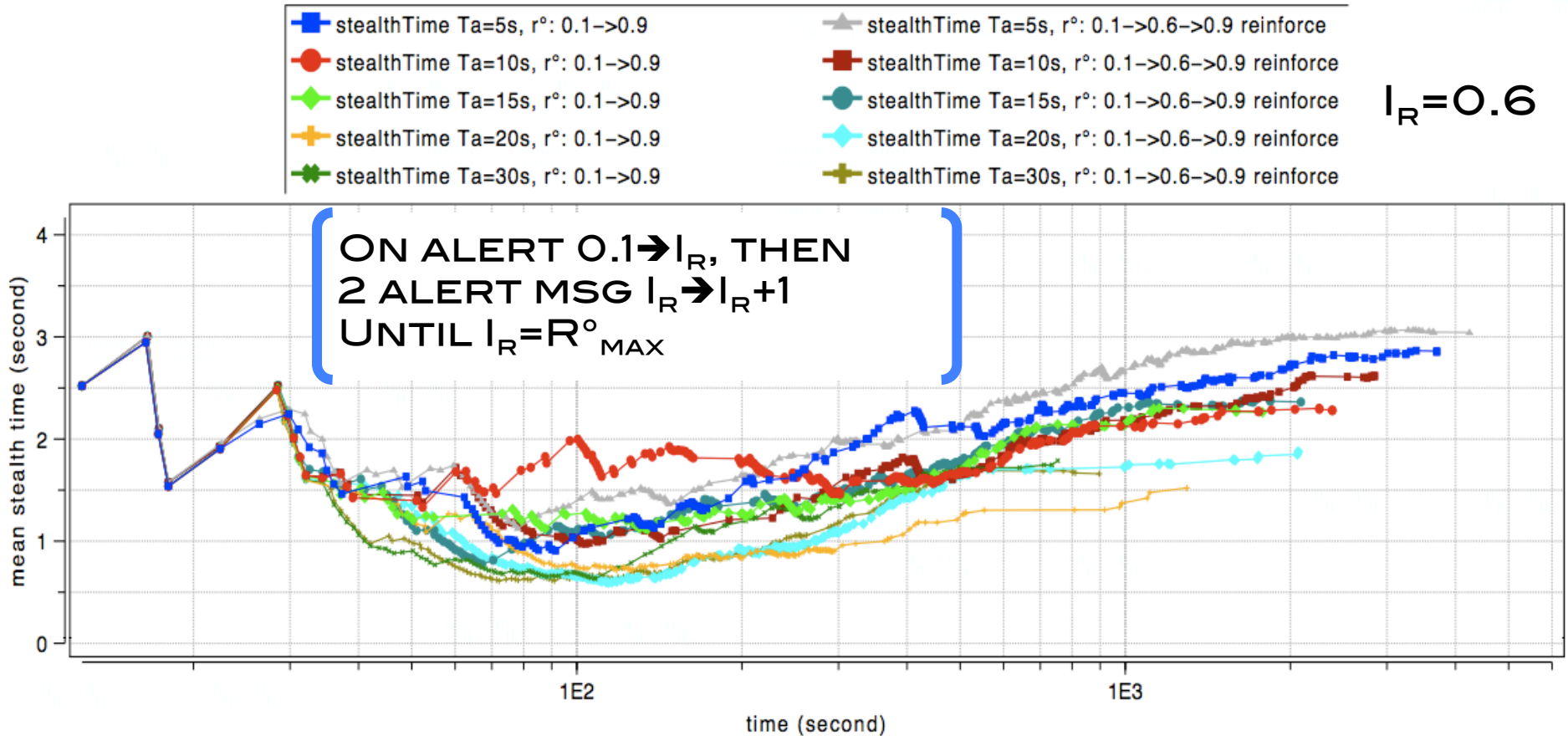


MEAN STEALTH TIME RISK-BASED SCHEDULING



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

MEAN STEALTH TIME W/WO REINFORCEMENT

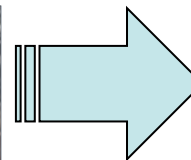
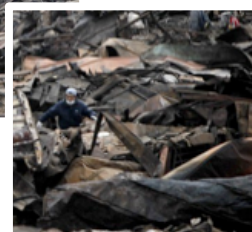
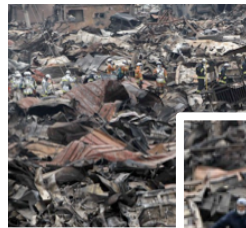
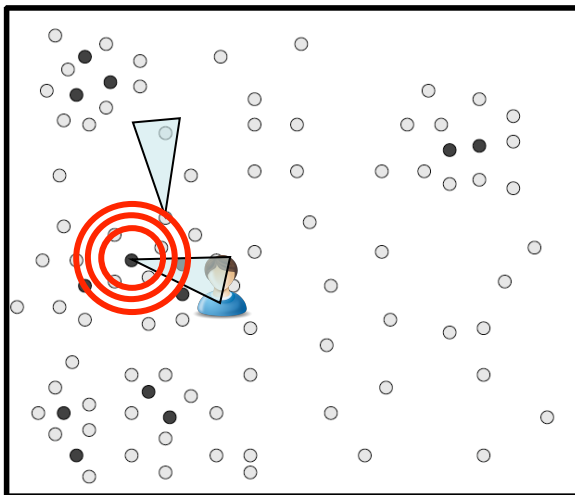


REINFORCEMENT ALWAYS INCREASES THE NETWORK LIFETIME
MEAN STEALTH TIME IS CLOSE TO THE NO-REINFORCEMENT
CASE, ESPECIALLY WHEN $T_A > 20S$

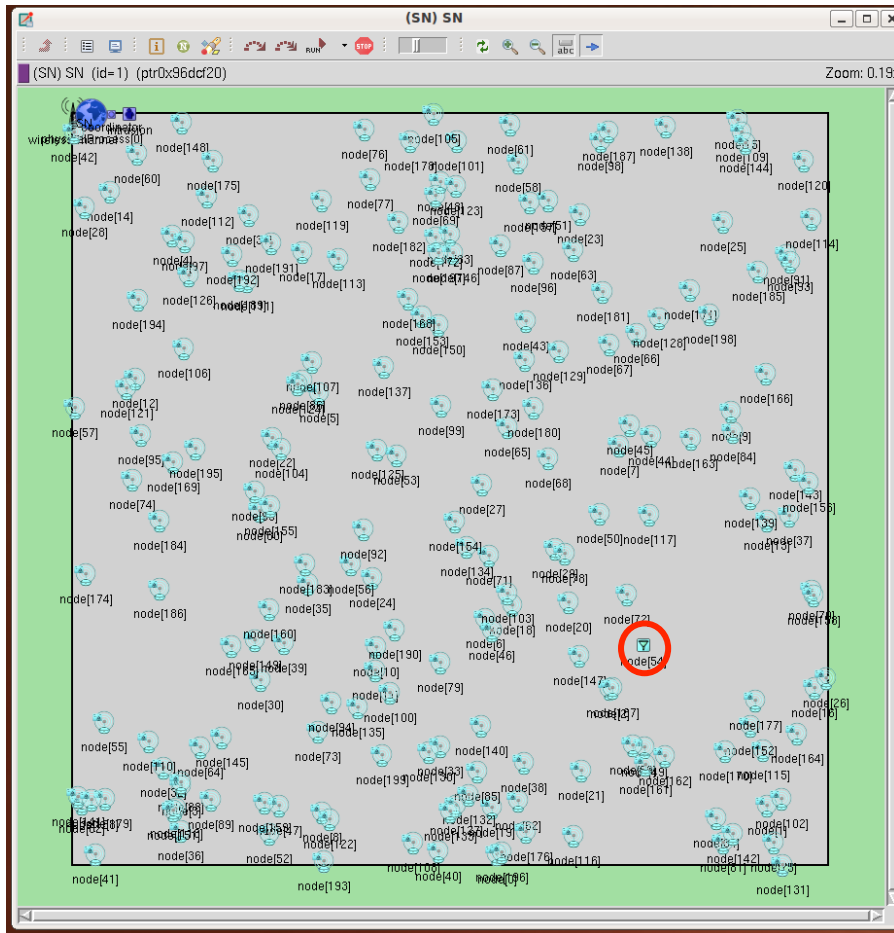
MODEL PARAMETERS SCHEDULING

- ❑ MANY FEATURES CAN BE CONTROLLED
 - ❑ INITIAL CRITICALITY LEVEL
 - ❑ MAXIMUM CRITICALITY LEVEL
 - ❑ MAXIMUM NUMBER OF COVERSET
 - ❑ MAXIMUM CAPTURE SPEED
 - ❑ INTRUDER'S VELOCITY
 - ❑ RE-INFORCEMENT BEHAVIOR
 - ❑ ACTIVATE COVERSET ON INTRUSION
 - ❑ CAMERA ROTATION CAPABILITY
 - ❑ FORCED INTRUSIONS FOR MULTIPLE-INTRUSION SCENARIOS
 - ❑ OCCLUSION DUE TO OBSTACLES

IMAGE TRANSMISSION, COMMUNICATION ISSUES



SEND IMAGES TO THE SINK

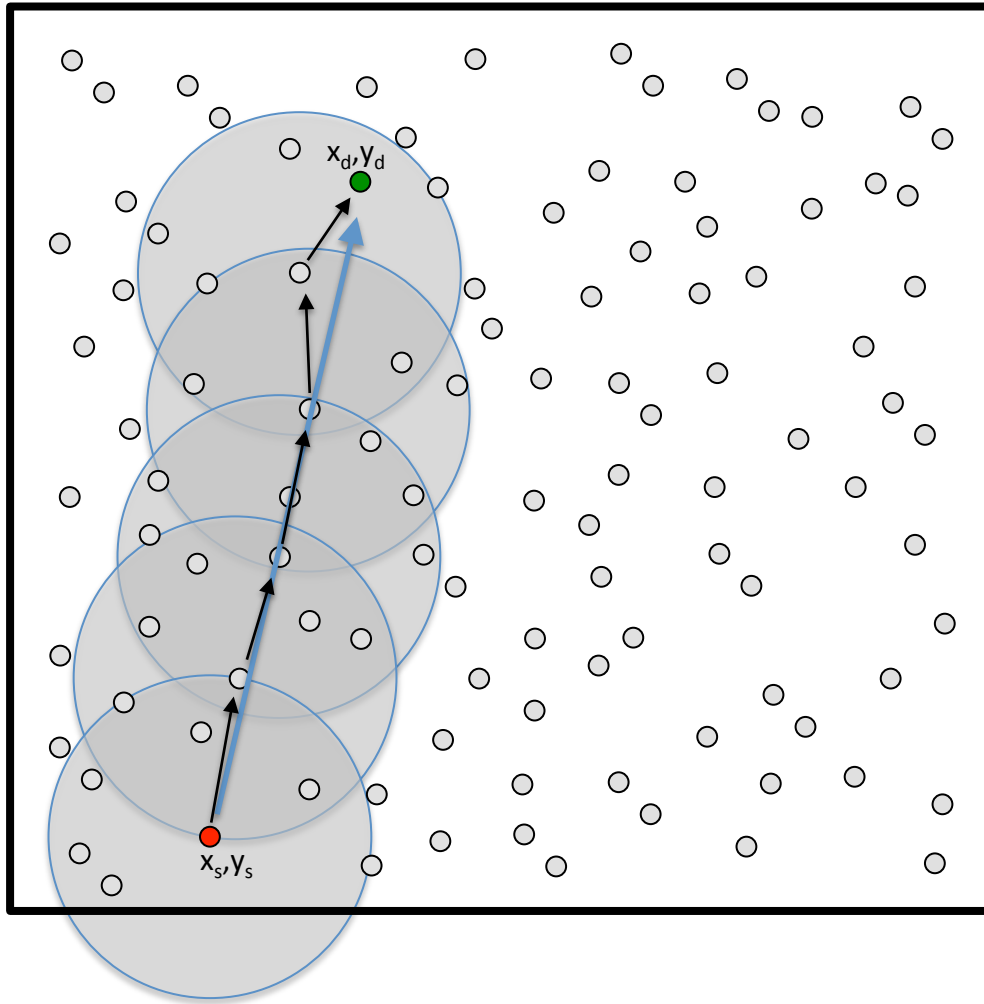


The log window displays the simulation's internal state and events. The top section shows simulation statistics: Run #0: SN, Event #1, T=0, Next: SN.node[0].ResourceManager (id=20), Msgs scheduled: 1002, Msgs created: 2202, Msgs present: 2202, Ew/sec: n/a, Simsec/sec: n/a, Ew/simsec: n/a. The log shows the initialization of various modules for nodes 197 and 198, including Sensor Dev Mgr, Application, and Communication modules. It also indicates that these nodes will use DRN encoding for real images and will not try to change TX power level. The log entries are as follows:

```
will NOT try to change TX power level
Application TX nb output levels=1
node 197: initialized with criticality level 0.1 MAX Coverset 12 capture at 0.01 fps
--> node 197: will use DRN encoding for real images
Application id: 197 px: 2436.62 py: 1246.41 alpha: 26 vx: 2049.42 vy: 930.058
          bx: 1895.12 by: 1118.91 cx: 2203.71 cy: 741.21
          Dpv: 500 Dpb: 556.301 Dpc: 556.301
          Initializing module SN,node[197], stage 0
          Initializing module SN,node[197].MobilityManager, stage 0
          Initializing module SN,node[197].ResourceManager, stage 0
          Initializing module SN,node[197].SensorManager, stage 0
          Initializing module SN,node[197].Communication, stage 0
          Initializing module SN,node[197].Communication.Radio, stage 0
          Initializing module SN,node[197].Communication.MAC, stage 0
          Initializing module SN,node[197].Communication.Routing, stage 0
          Initializing module SN,node[197].Application, stage 0
          0
          SN,node[197].Application Application id 198 px 3877.73 py :
Application TX output power=0dBm
has NO Castalia modified radio
will NOT try to change TX power level
Application TX nb output levels=1
node 198: initialized with criticality level 0.1 MAX Coverset 12 capture at 0.01 fps
--> node 198: will use DRN encoding for real images
Application id: 198 px: 3877.73 py: 1570.55
          bx: 4425 by: 1471.03 cx: 4293.33 cy: 1940.65
          Dpv: 500 Dpb: 556.301 Dpc: 556.301
          Initializing module SN,node[198], stage 0
          Initializing module SN,node[198].MobilityManager, stage 0
          Initializing module SN,node[198].ResourceManager, stage 0
          Initializing module SN,node[198].SensorManager, stage 0
          Initializing module SN,node[198].Communication, stage 0
          Initializing module SN,node[198].Communication.Radio, stage 0
          Initializing module SN,node[198].Communication.MAC, stage 0
          Initializing module SN,node[198].Communication.Routing, stage 0
          Initializing module SN,node[198].Application, stage 0
          0
          SN,node[198].Application Application id 199 px 2139.35 py :
Application TX output power=0dBm
has NO Castalia modified radio
will NOT try to change TX power level
Application TX nb output levels=1
**initial_coverage** t=0 pert.coverage 73.433
node 199: initialized with criticality level 0.1 MAX Coverset 12 capture at 0.01 fps
--> node 199: will use DRN encoding for real images
Application id: 199 px: 2139.35 py: 3905.54
          bx: 2030.42 by: 3350.01 cx: 2502.17 cy: 3483.84
          Dpv: 500 Dpb: 556.301 Dpc: 556.301
          Initializing module SN,wirelessChannel, stage 1
```

GEOGRAPHIC ROUTING

GPSR



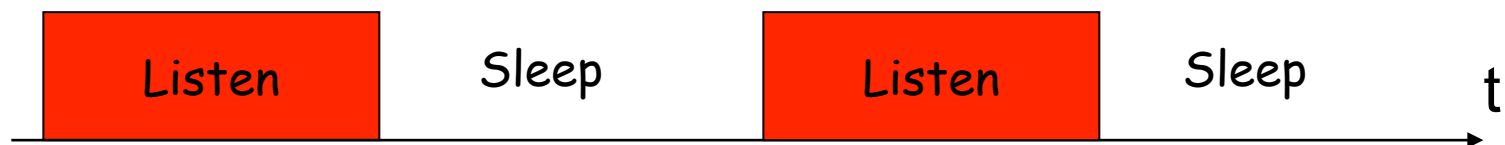
Avoids keeping routing information

Relies on geographic (GPS) coordinates to find next-hop node

Reduces route maintenance overhead

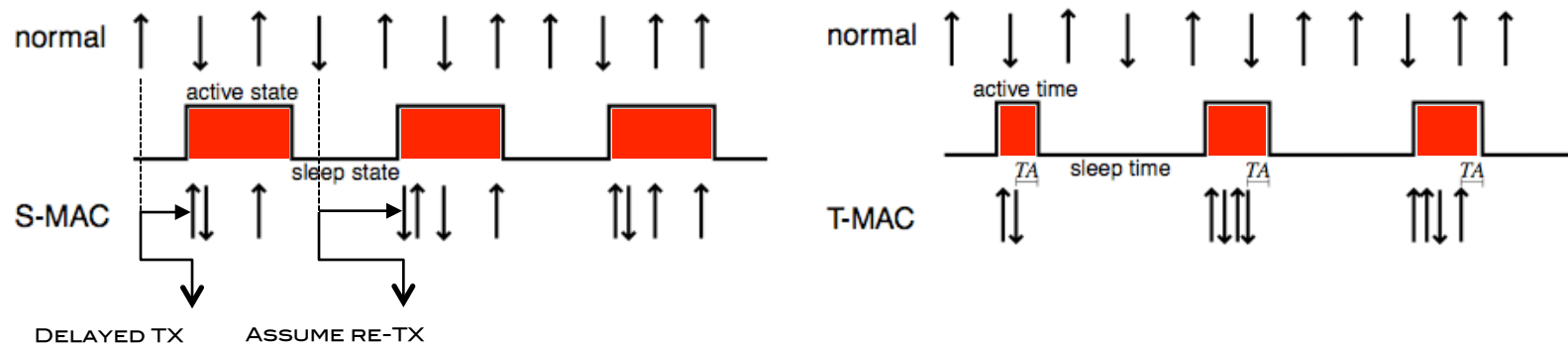
S-MAC - SENSOR MAC

- **NODES PERIODICALLY SLEEP**
- **TRADES ENERGY EFFICIENCY FOR LOWER THROUGHPUT AND HIGHER LATENCY**
- **SLEEP DURING OTHER NODES TRANSMISSIONS**
- **NEEDS COMPLEX SYNCHRONIZATION MECHANISMS**



T-MAC - TIMEOUT MAC

- TRANSMIT ALL MESSAGES IN BURSTS OF VARIABLE LENGTH AND SLEEP BETWEEN BURSTS
- RTS / CTS / ACK SCHEME
- SYNCHRONIZATION SIMILAR TO S-MAC

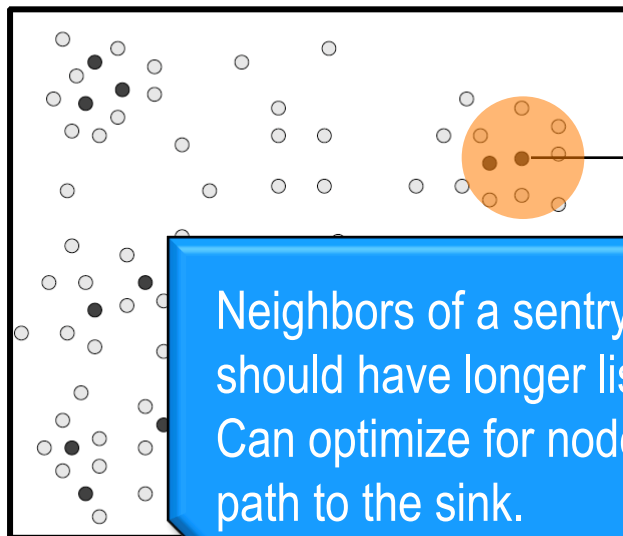


OUR CURRENT RESEARCH ON MAC LAYER

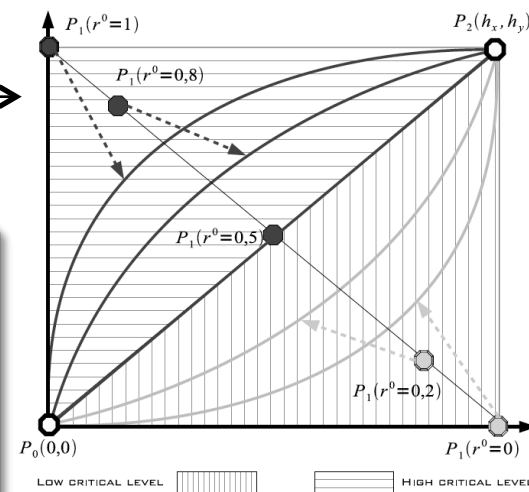
□ DUTY-CYCLED MAC (E.G. SMAC)



□ LINK THE LISTENING TIME TO THE CRITICALITY MODEL



Neighbors of a sentry nodes should have longer listening time. Can optimize for nodes on the path to the sink.



DISPLAY IMAGES

The screenshot displays the OMNeT++/Tkenv simulation environment for a network named 'SN'. The interface includes a network diagram on the left, a log window on the right, and a sink image in the top-left corner. The log window shows a series of messages from node 172 to node 54, including image data packets and their reception. A blue callout box provides information on the data being displayed.

Provide information on:

- sink id
- sender id
- latency
- number of received packets

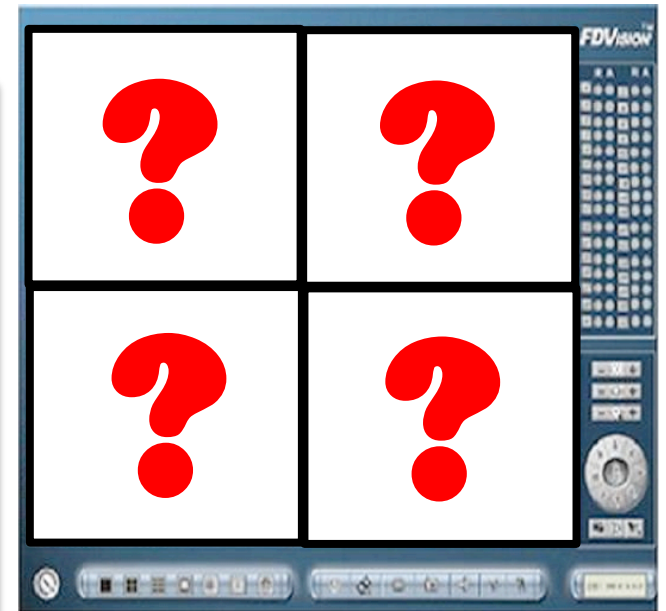
Log Window Content:

```
WC_BEGIN... TunableMac data packet... Timer message...
WC_END... Periodic energy calculation... NormalCriticality level... Timer message...
SN.node[172],Application Sending [Image] of size 83 bytes to communication layer
SN.node[172],Application Sending [Image] of size 90 bytes to communication layer
SN.node[172],Application Sending [Image] of size 95 bytes to communication layer
SN.node[172],Application Sending [Image] of size 91 bytes to communication layer
SN.node[172],Application Sending [Image] of size 85 bytes to communication layer
SN.node[172],Application Sending [Image] of size 89 bytes to communication layer
SN.node[172],Application Sending [Image] of size 86 bytes to communication layer
SN.node[172],Application Sending [Image] of size 87 bytes to communication layer
SN.node[172],Application Sending [Image] of size 79 bytes to communication layer
SN.node[172],Application Sending [Image] of size 83 bytes to communication layer
SN.node[172],Application Sending [Image] of size 89 bytes to communication layer
SN.node[172],Application Sending [Image] of size 79 bytes to communication layer
SN.node[172],Application Sending [Image] of size 91 bytes to communication layer
SN.node[172],Application Sending [Image] of size 90 bytes to communication layer
SN.node[172],Application Sending [Image] of size 80 bytes to communication layer
SN.node[172],Application Sending [Image] of size 89 bytes to communication layer
SN.node[172],Application Sending [Image] of size 77 bytes to communication layer
SN.node[172],Application Sending [Image] of size 95 bytes to communication layer
SN.node[172],Application Sending [Image] of size 88 bytes to communication layer
SN.node[172],Application Sending [Image] of size 89 bytes to communication layer
SN.node[172],Application Sending [Image] of size 82 bytes to communication layer
SN.node[172],Application Sending [Image] of size 85 bytes to communication layer
SN.node[172],Application Sending [Image] of size 95 bytes to communication layer
SN.node[172],Application Sending [Image] of size 89 bytes to communication layer
SN.node[172],Application Sending [Image] of size 83 bytes to communication layer
SN.node[172],Application Sending [Image] of size 81 bytes to communication layer
SN.node[172],Application Sending [Image] of size 74 bytes to communication layer
SN.node[172],Application Sending [Image] of size 81 bytes to communication layer
SN.node[172],Application Sending [Image] of size 83 bytes to communication layer
SN.node[172],Application Sending [Image] of size 90 bytes to communication layer
SN.node[172],Application Sending [Image] of size 84 bytes to communication layer
SN.node[172],Application Sending [Image] of size 87 bytes to communication layer
SN.node[172],Application Sending [Image] of size 94 bytes to communication layer
SN.node[172],Application Sending [Image] of size 94 bytes to communication layer
SN.node[172],Application Sending [Image] of size 87 bytes to communication layer
SN.node[172],Application Sending [Image] of size 86 bytes to communication layer
SN.node[172],Application Sending [Image] of size 51 bytes to communication layer
SN.node[172],Application Node 172 -> REAL_IMAGE(0) to node 54, #packet: 206, stored #packets: 206
SN.node[54],Application Node 54: WRITES IMAGE FILE(0) from node 2, Size=11377_R54(41,4)-2(0)-206of206-L=0,76
SN.node[54],Application Node 54: DISPLAY REAL_IMAGE(0) from node 2, 206/206, First packet received at 40.6225
SN.node[54],Application Node 54: WRITES IMAGE FILE(0) from node 44, Size=5054_R54(42,4)-44(0)-92of206-L=0,34
SN.node[54],Application Node 54: DISPLAY REAL_IMAGE(0) from node 44, 92/206, First packet received at 42.1051
```

END-TO-END PERFORMANCES?



Holes in deployment
Limited buffers
Multi-hop overhead
Congestion
Channel contention
Duty-cycling MAC
Physical interference
Small PDU
Nodes availability
...



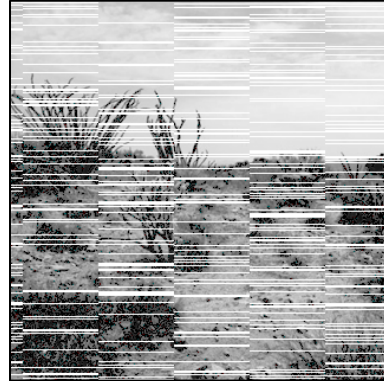
END-TO-END LOSS RATE IS EXPECTED TO BE HIGH!

IMAGE QUALITY? UNCOMPRESSED BMP

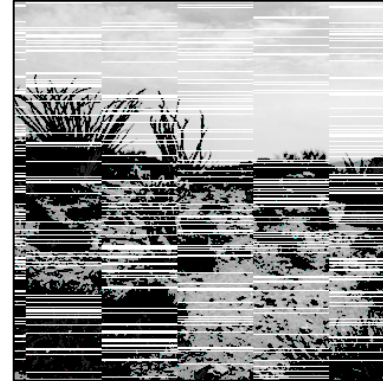
1617 PACKETS, 64 BYTES PAYLOAD, ONE HOP
LOSS RATE: 20%, NO LOSS BURSTS (RADIO), NO DUTY-CYCLING



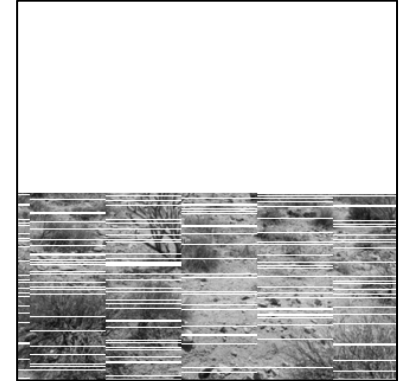
ORIGINAL 320X320
256 GRAY LEVELS,
BMP 102400 BYTES



1340 OUT OF 1617
PACKETS RECEIVED



1303 OUT OF 1617
PACKETS RECEIVED



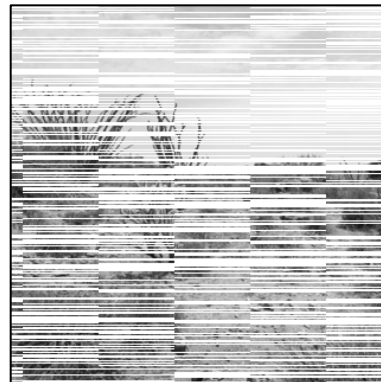
674 OUT OF 1617
PACKETS RECEIVED

MAX TX RATE = 250 KPS
(IEEE 802.15.4)

MINIMUM LATENCY = 6.46S

Cannot really use the
compressed version of
BMP using RLE.

WITH LOSS BURSTS (RADIO)



921 OUT OF 1617
PACKETS RECEIVED



689 OUT OF 1617
PACKETS RECEIVED



913 OUT OF 1617
PACKETS RECEIVED

IMAGE QUALITY? STANDARD JPG

427 PACKETS, 64 BYTES PAYLOAD, ONE HOP
LOSS RATE: 20%, NO LOSS BURSTS (RADIO), NO DUTY-CYCLING



ORIGINAL 320X320
256 GRAY LEVELS,
JPG 27303 BYTES

MAX TX RATE = 250 KPS
(IEEE 802.15.4)

MINIMUM LATENCY = 1.61S



348 OUT OF 427
PACKETS RECEIVED



351 OUT OF 427
PACKETS RECEIVED

9 OUT OF 12 IMAGES
COULD NOT BE DECODED



349 OUT OF 1617
PACKETS RECEIVED

WITH LOSS BURSTS (RADIO)



258 OUT OF 427
PACKETS RECEIVED



270 OUT OF 427
PACKETS RECEIVED

8 OUT OF 12 IMAGES
COULD NOT BE DECODED



269 OUT OF 427
PACKETS RECEIVED

Encoding cost of
JPEG2000 is too high for
these devices.

IMPROVING IMAGE ROBUSTNESS

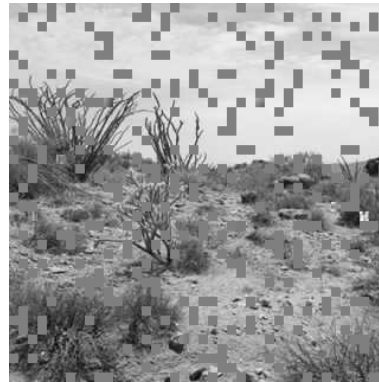
302 PACKETS, 64 BYTES PAYLOAD, ONE HOP
LOSS RATE: 20%, NO LOSS BURSTS (RADIO), NO DUTY-CYCLING



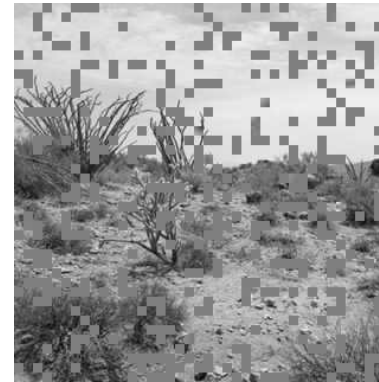
ORIGINAL 320X320
256 GRAY LEVELS,
WSN SPECIFIC 17199 BYTES

MAX TX RATE = 250 KPS
(IEEE 802.15.4)

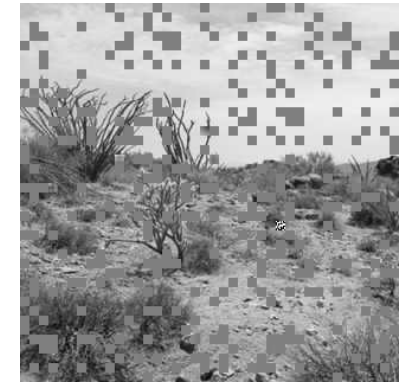
MINIMUM LATENCY = 1.14S



248 OUT OF 302
PACKETS RECEIVED

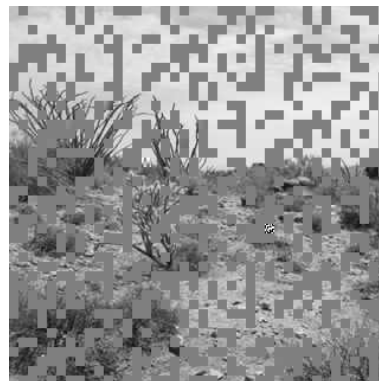


236 OUT OF 302
PACKETS RECEIVED

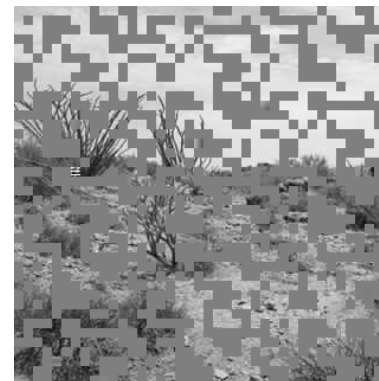


243 OUT OF 302
PACKETS RECEIVED

WITH LOSS BURSTS (RADIO)



188 OUT OF 302
PACKETS RECEIVED



167 OUT OF 302
PACKETS RECEIVED



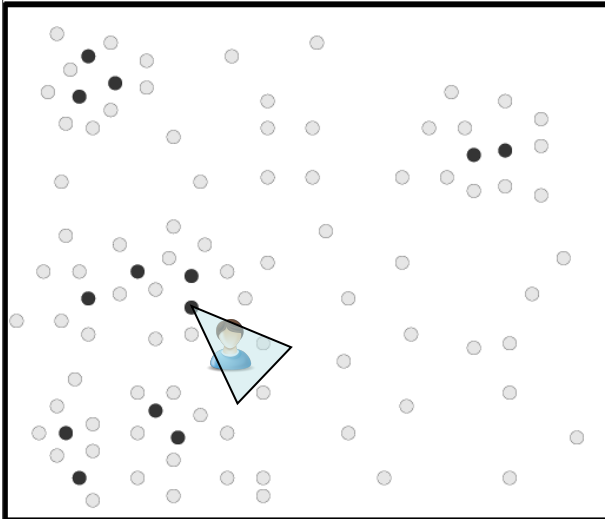
158 OUT OF 302
PACKETS RECEIVED

Collaboration with CRAN
laboratory, Nancy, France,
for robust image encoding
techniques for WSN.

INTRUSION DETECTION SCENARIO

(A)

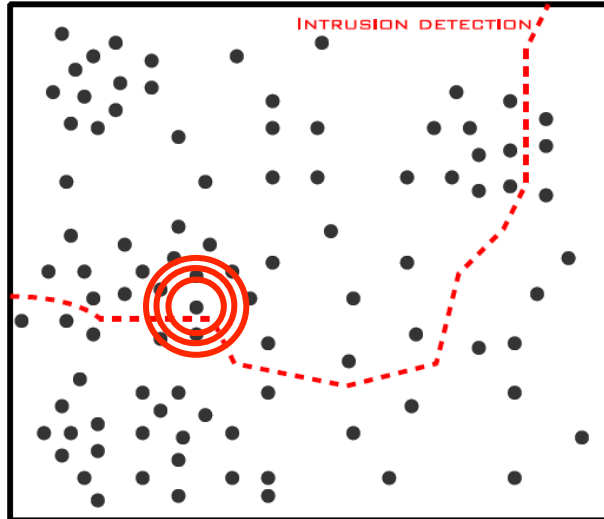
- SENTRY NODE: NODE WITH HIGH SPEED CAPTURE (HIGH COVER SET).
- SLEEP NODE: NODE WITH LOW SPEED CAPTURE.



HIBERNATE MODE
 $r^o = 0$

(B)

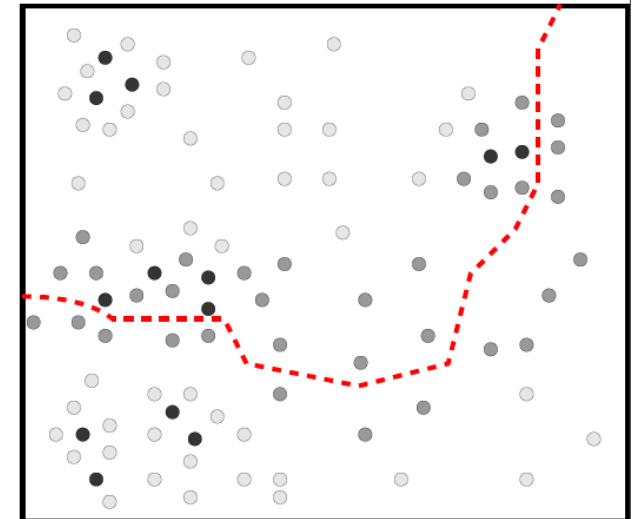
- ALERTED NODE: NODE WITH HIGH SPEED CAPTURE (ALERT INTRUSION).



ALERT MODE
 $r^o = \text{MAX}$

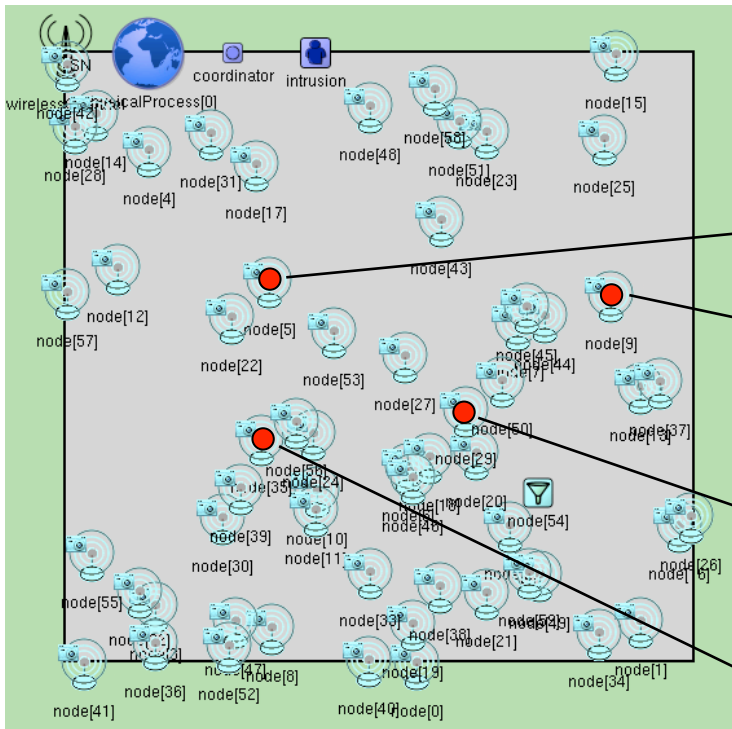
(C)

- SENTRY NODE: NODE WITH HIGH SPEED CAPTURE (HIGH COVER SET).
- CRITICAL NODE: NODE WITH HIGH SPEED CAPTURE (NODE THAT DETECTS THE INTRUSION).
- SLEEP NODE: NODE WITH LOW SPEED CAPTURE.



HIBERNATE MODE (AFTER INTRUSION)
 $r^o = 0$

SOME IMAGES DISPLAYED BY THE SINK



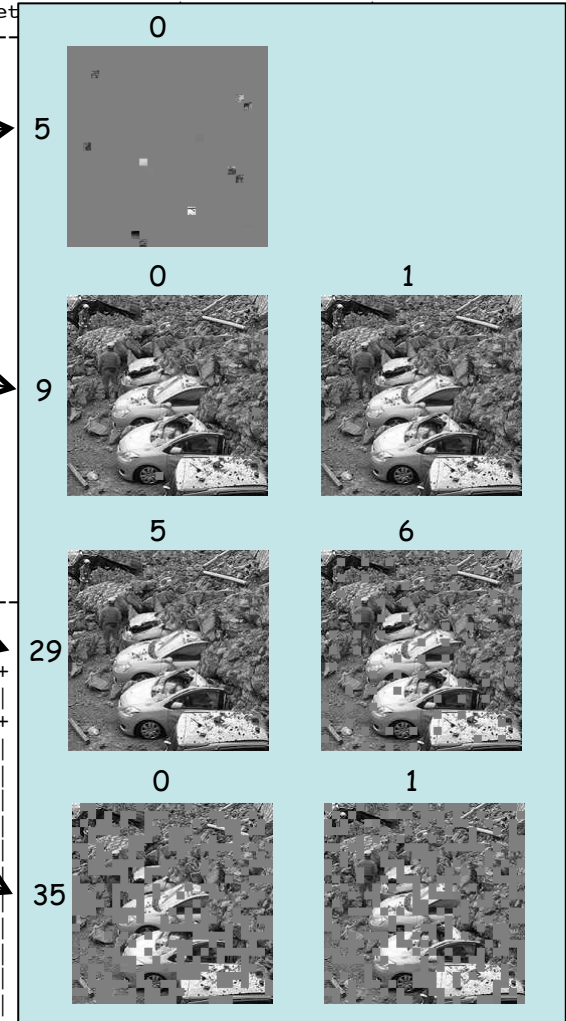
60 IMAGE SENSOR NODES
75MX75M
1 SINK (NODE 54)

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



MODEL PARAMETERS

IMAGE TRANSMISSION

- ❑ MANY FEATURES CAN BE CONTROLLED
 - ❑ NAME OF ORIGINAL IMAGE FILE TO SEND
 - ❑ ENERGY PER IMAGE (CAPTURE, COMPRESSION, TRANSMISSION)
 - ❑ NUMBER OF IMAGES ON INTRUSION
 - ❑ FORCED IMAGE SENDING, TIME INTERVAL FOR FORCED IMAGE SENDING
 - ❑ NUMBER OF IMAGES ON COVERSET ACTIVATION
 - ❑ AUTOMATIC DISPLAY OF IMAGE BY SINK
 - ❑ KEEP RECEIVED IMAGES ON DISK
 - ❑ RANDOM PACKET DROP AND CORRUPTED BYTES AT APPLICATION LEVEL

VIDEOSENSOR NODE'S PARAMETER

The videoSensor node's parameters are defined in the .ned file following the OMNET++ syntax.

```
VideoSensorNode.ned (-:/Castalia-3.2/src/nodes/application/videoSensorNode) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
VideoSensorNode.ned
package node.application.videoSensorNode;

// The sensor node module. Connects to the wireless channel in order to communicate
// with other nodes. Connects to physical processes so it can sample them.

simple VideoSensorNode like node.application.iApplication {
parameters:
    string applicationID = default ("videoSensorNode");
    bool collectTraceInfo = default (true);
    int priority = default (1);
    int packetHeaderOverhead = default (0); // in bytes
    int constantDataPayload = default (0); // in bytes

    // specific to our model
    bool isSink = default (false);

    double minCaptureRate = default (0.01); // in fps
    double maxCaptureRate = default (3.0); // in fps
    double staticCaptureRate = default (0.2); // in fps

    double aov = default (36.0); // in degree, alpha = aov/2 = 18°
    double dov = default (25.0); // in meters
    double lineOfSight = default (-1.0); // in degree, negative line of sight means random line of sight between 0..360° (0..2PI)

    double maxBatteryLevel = default (100.0); // in units
    double energyPerCapture = default (1.0); // in battery units
    double measuredEnergyPerImageCapture = default (0.0); // in mJ
    double measuredEnergyPerImageCompression = default (0.0); // for instance, prior to transmission, in mJ
    double measuredEnergyPerImageProcessing = default (0.0); // for instance, for intrusion detection, in mJ

    bool alwaysActive = default (false);

    double criticalityLevel = default (0.9); // must be between 0 and 1
    double minCriticalityLevel = default (0.1); // must be between 0 and 1
    double maxCriticalityLevelPeriod = default (5.0); // in seconds
    double maxDefinedCoverSetNumber = default (12.0);
    // -1 means automatically detect the number of coversets to get the capture rate
    // n would mean that the computation of the capture rate uses n coversets regardless of the real number of coversets
    // which is usefull for test scenario with a small number of nodes
    double forceNumberOfCoversets = default (-1);

    bool isMobile = default (false);
    double isMobileProb = default (1.0); // must be between 0 and 1
    bool isCamRotatable = default (false);
    double isCamRotatableProb = default (1.0); // must be between 0 and 1
    double maxMobilitySpeed = default (0.5); // in m/s
    double maxCamRotationTime = default (5.0); // in seconds for a complete rotation
    double maxCamRotationCount = default (2.0); // in number of rotations
    double load = default (0.5); // not used for the moment

    bool idReportToSink = default (false); // can be used to send test packets to the sink
    // specify the sink for source-initiated on-demand routing such as AODV
    // if "NO" then the routing protocol is assumed to be a pro-active protocol such as MPRings (Castalia) or OLSR
    string nextRecipient = default ("NO");

    bool sendImageOnIntrusion = default (false);
    double forcedIntrusionProb = default (0.0);
    bool forceSendImage = default (false);
    double forceSendImageInterval = default (10.0); // in seconds
    bool activateCoverSetOnIntrusion = default (false);
    bool propagateCoverSetActivation = default (false);
    double forcedCoverSetActivationAt = default (-1.0);
    int imageCountOnIntrusion = default (1); // how many images a node send when it detects an intrusion
    int imageCountOnCoverSetActivation = default (1); // how many images a node send when it is activated by a neighbor node
    int imageByteSize = default (32000); // in bytes -> 320x200, 16 colors/pixel
    int imageChunkSize = default (256); // in bytes
    string imageFilename = default ("NO");
    bool cranEncoding = default (false);
    string imageBMPOriginalFilename = default ("NO");
    bool displayReceivedImage = default (false);
    bool waitForKeyPressWhenDisplayImage = default (false);
    double displayReceivedImageTimer = default(10); // in s
```

LAUNCHING SIMULATIONS

- ❑ USE THE POWERFUL *CASTALIA* SCRIPT TO LAUNCH SPECIFIC SIMULATION SCENARIO
- ❑ DEFINE CONFIGURATIONS IN OMNETPP.INI FILE
 - ❑ [Config Video200Config]
 - ❑ [Config CSMA], [Config SMAC], [Config TMAC]
 - ❑ [Config GPSRRoutingForcedImageTest60_300]
 - ❑ [Config BMP128x128ConsumptionModel]
 - ❑ [Config CRAN200x200ConsumptionModel]
- ❑ SELECT CONFIGURATION COMBINATION
 - ❑ Castalia -c
CSMA,GPSRRoutingForcedImageTest60_300,Video200Config,CRAN200x200ConsumptionModel

ENABLING LARGE-SCALE, OPERATIONAL SEARCH & RESCUE APPLICATIONS



SENSOR & ROBOTS

❑ WIRELESS SENSOR NETWORKS

- ❑ LARGE SCALE SENSING
- ❑ NATURAL COLLABORATION THROUGH DATA AGGREGATION, REPORTING, ...
- ❑ MOBILITY IS NOT A PRIORITY

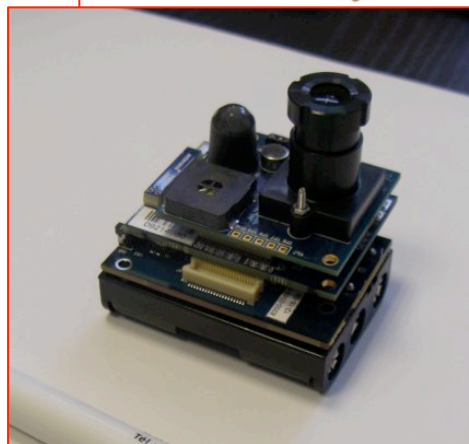
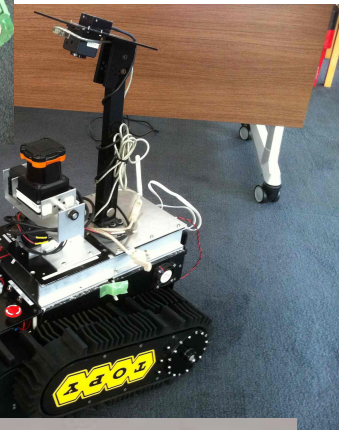
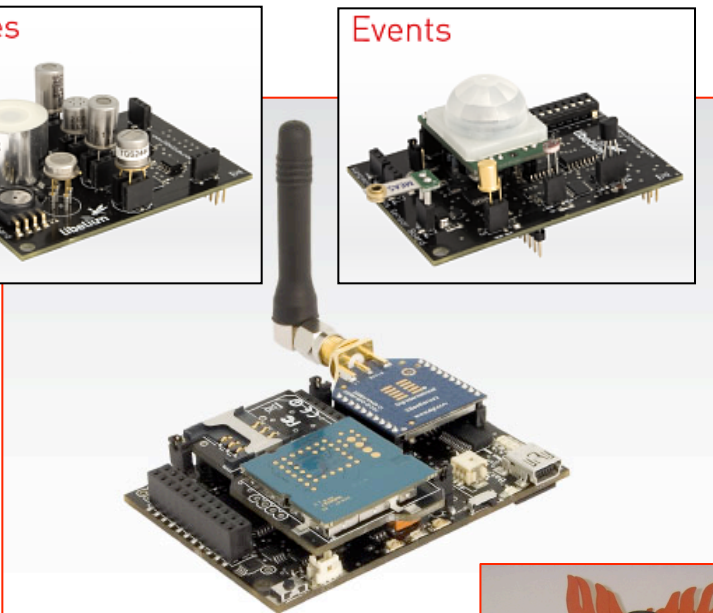
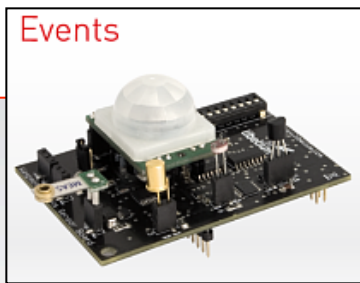
❑ ROBOTS

- ❑ MOBILITY IS A FUNDAMENTAL FEATURE
- ❑ EXPLORATION, RESCUE

❑ SENSOR & ROBOTS

- ❑ WSN PROVIDE SENSING DATA TO ROBOTS
- ❑ ROBOTS MAINTAIN CONNECTIVITY
- ❑ SENSORS COULD HELP FOR LOCALIZATION WHEN GPS DATA ARE DOWN

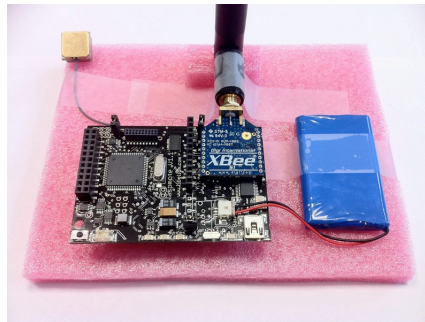
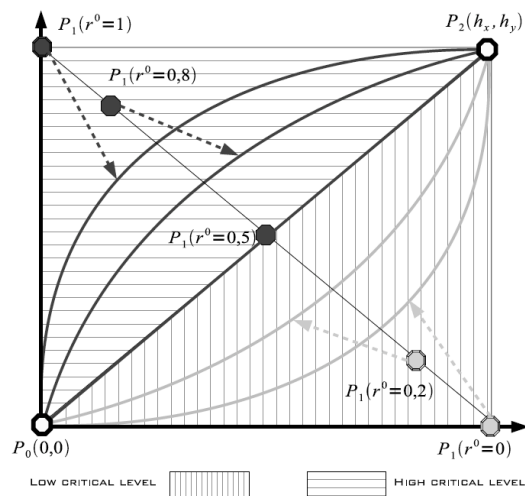
CHALLENGING COOPERATION IMPLIES DIFFERENCES!



SENSORS & ROBOTS

PROPOSE NEW INTERACTION SCHEMES

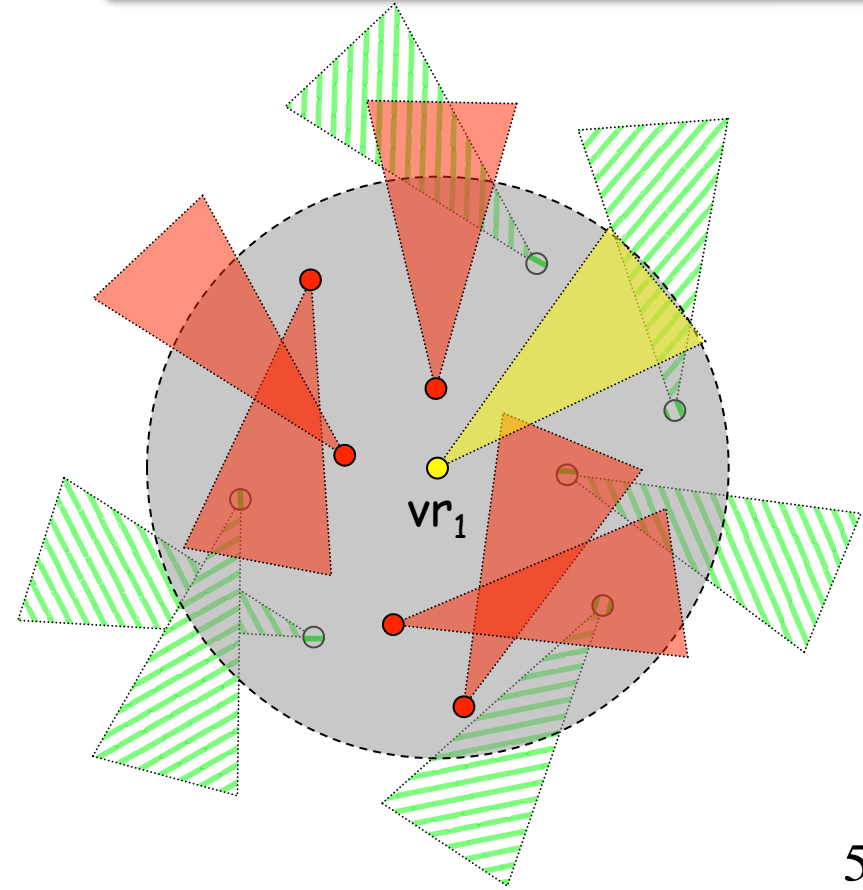
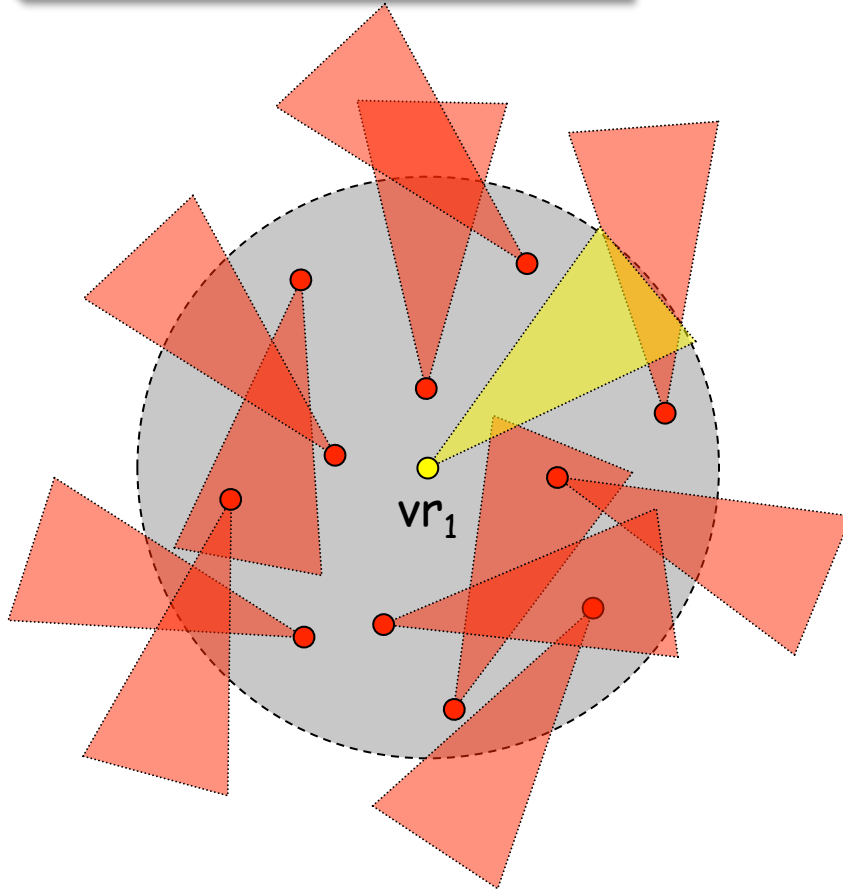
- USE THE CRITICALITY MODEL TO CONTROL BOTH SENSORS AND ROBOTS
- PROTOTYPING ON REAL HARDWARE



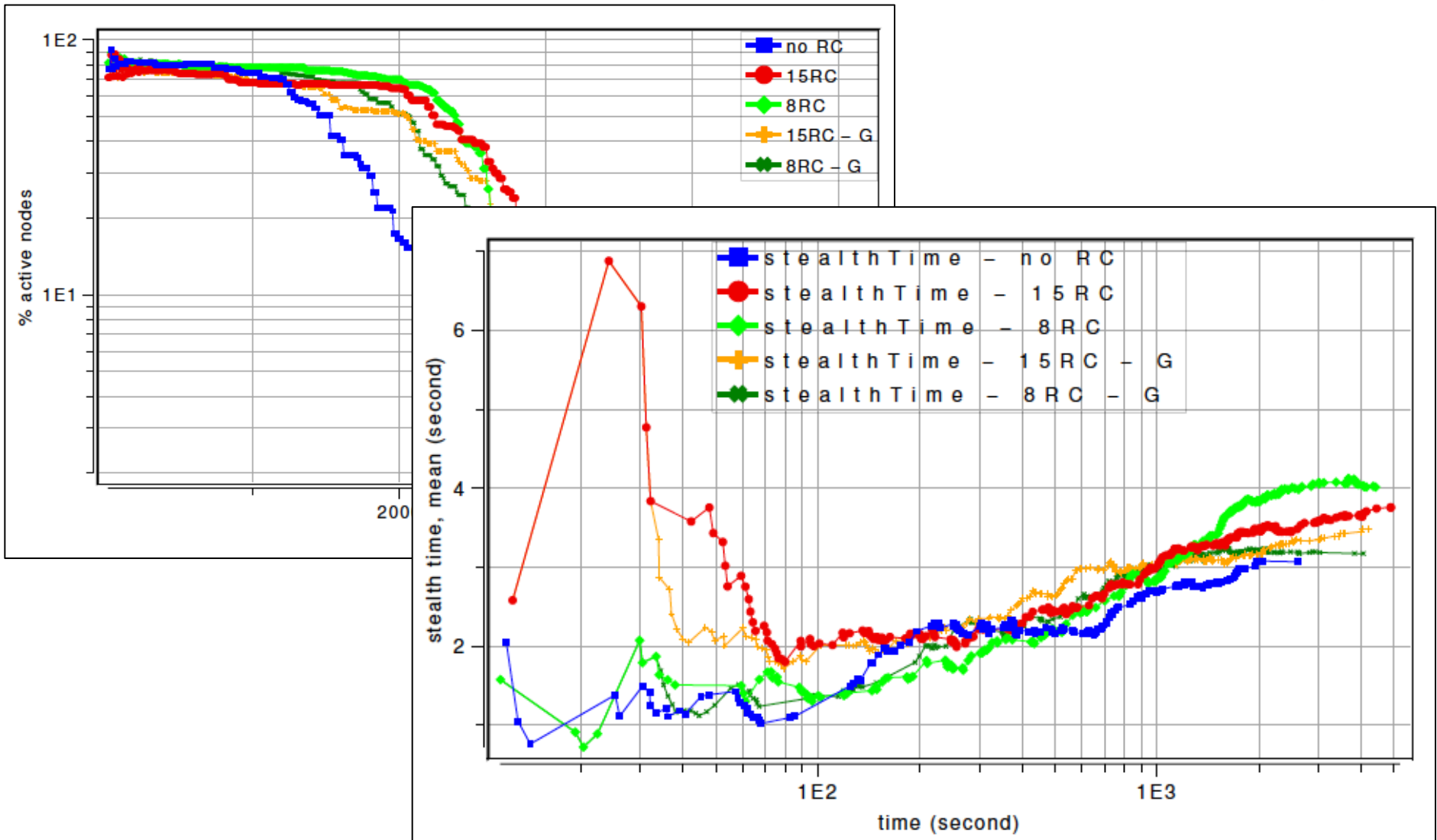
COOPERATION WITH CAMERAS ON MOBILE ROBOTS

Fixed image sensors near a mobile camera can decrease their criticality level

ONLY fixed image sensors whose FoV's center is covered by a mobile camera **CAN** decrease their criticality level

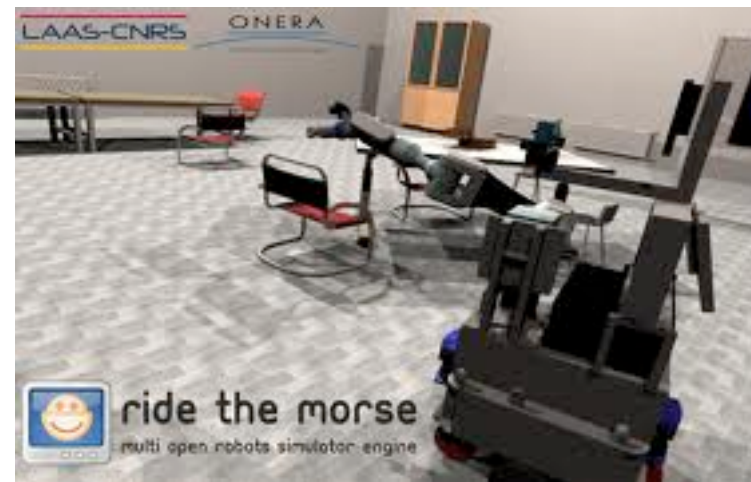
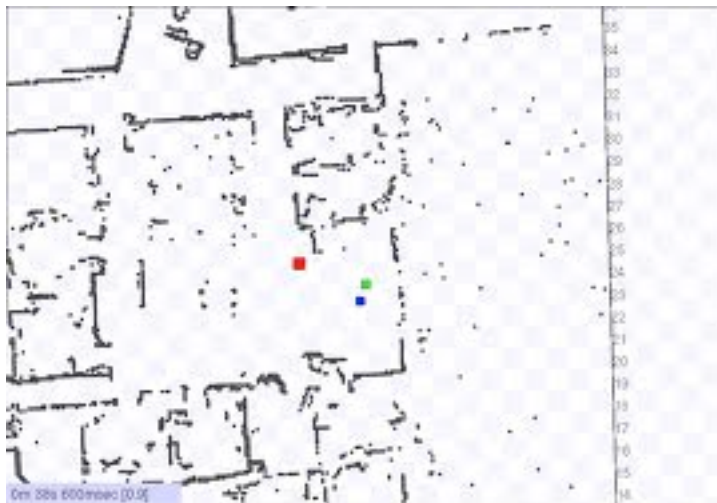


IMPACT ON LIFETIME & STEALTH TIME



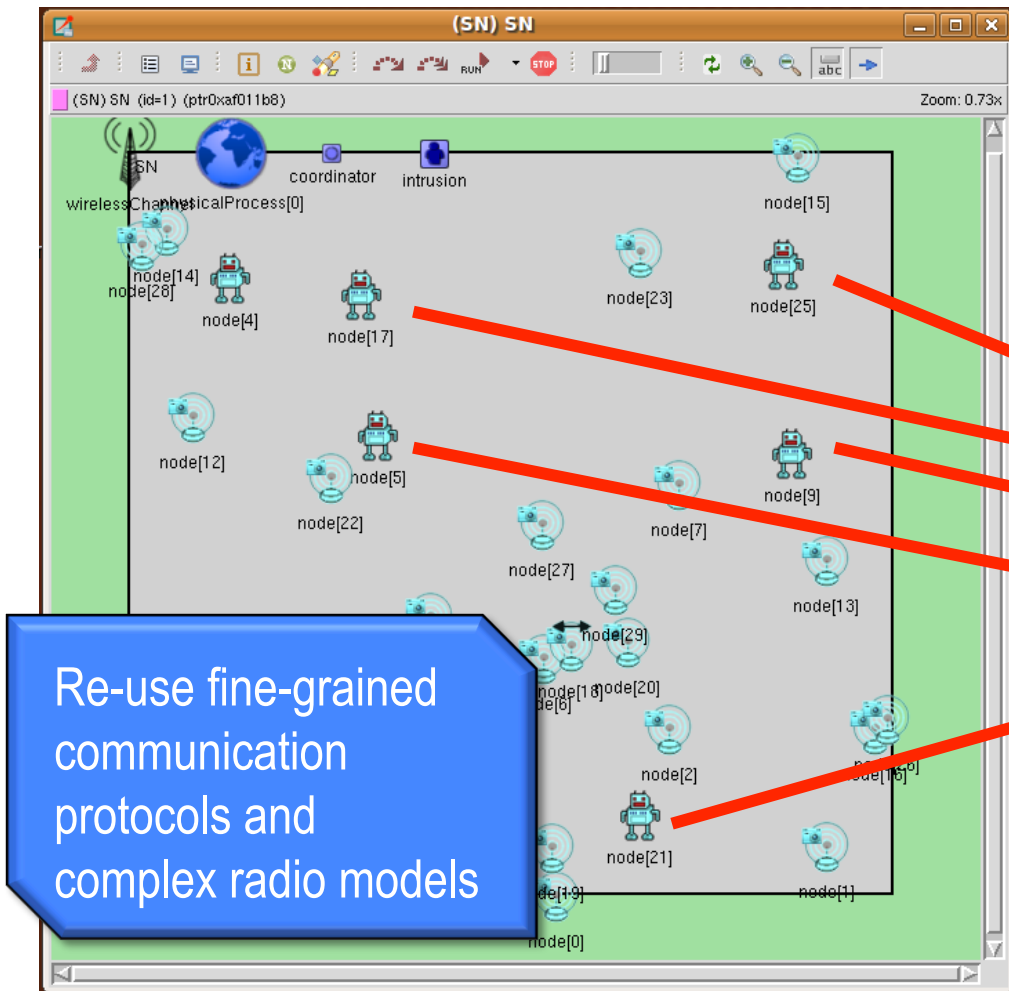
ROBOT SIMULATORS

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

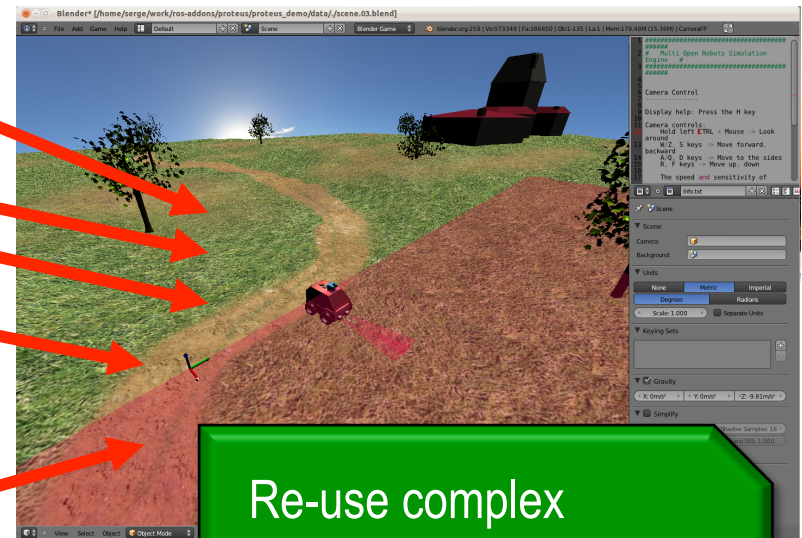


SENSORS & ROBOTS ENABLE REALISTIC INTERACTION STUDIES

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,...)

CONCLUSIONS

- ❑ WSN'S NATURAL APPLICATION IS SURVEILLANCE BUT...
- ❑ ... USING WSN TECHNOLOGY FOR MISSION-CRITICAL APPLICATIONS IS FAR FROM BEING MATURE!
- ❑ NEED TO TAKE THE APPLICATION'S CRITICALITY INTO ACCOUNT WHEN DESIGNING CONTROL MECHANISMS AND PROTOCOLS
- ❑ BUILDING EFFICIENT, RELIABLE LOW LAYERS IS CHALLENGING!
- ❑ SENSORS & ROBOTS ARE COMPLEMENTARY TECHNOLOGIES FOR MISSION-CRITICAL APPLICATIONS BUT...
- ❑ ...NEED SUITABLE TOOLS!

WEB LINKS

[HTTP://WEB.UNIV-PAU.FR/~CPHAM/IWEB/WSN/HOME.HTML](http://web.univ-pau.fr/~cpham/iweb/wsn/home.html)

PUBLICLY AVAILABLE MODEL

ARCHIVE

[HTTP://WEB.UNIV-PAU.FR/~CPHAM/WSN-MODEL/DISTRIB/WVSNMODEL-V4.TGZ](http://web.univ-pau.fr/~cpham/wsn-model/distrib/wvsnmodel-v4.tgz)

DOCUMENTATION

[HTTP://WEB.UNIV-PAU.FR/~CPHAM/WSN-MODEL/WVSN-CASTALIA.HTML](http://web.univ-pau.fr/~cpham/wsn-model/wvsn-castalia.html)

RESTRICTED ACCESS

ARCHIVE

[HTTP://WEB.UNIV-PAU.FR/~CPHAM/WSN-MODEL/DISTRIB-PRIVATE/](http://web.univ-pau.fr/~cpham/wsn-model/distrib-private/)

DOCUMENTATION

[HTTP://WEB.UNIV-PAU.FR/~CPHAM/WSN-MODEL/WVSN-CASTALIA-PRIVATE.HTML](http://web.univ-pau.fr/~cpham/wsn-model/wvsn-castalia-private.html)