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

Université de Pau, France

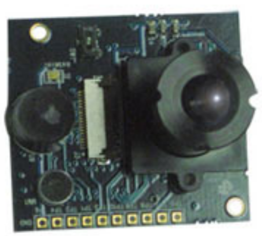
videoSense: un modèle de simulation de capteurs vidéo sous OMNET++/Castalia

Journée RESSACS 2012
7 Juin, 2012
IUT Bayonne, Anglet

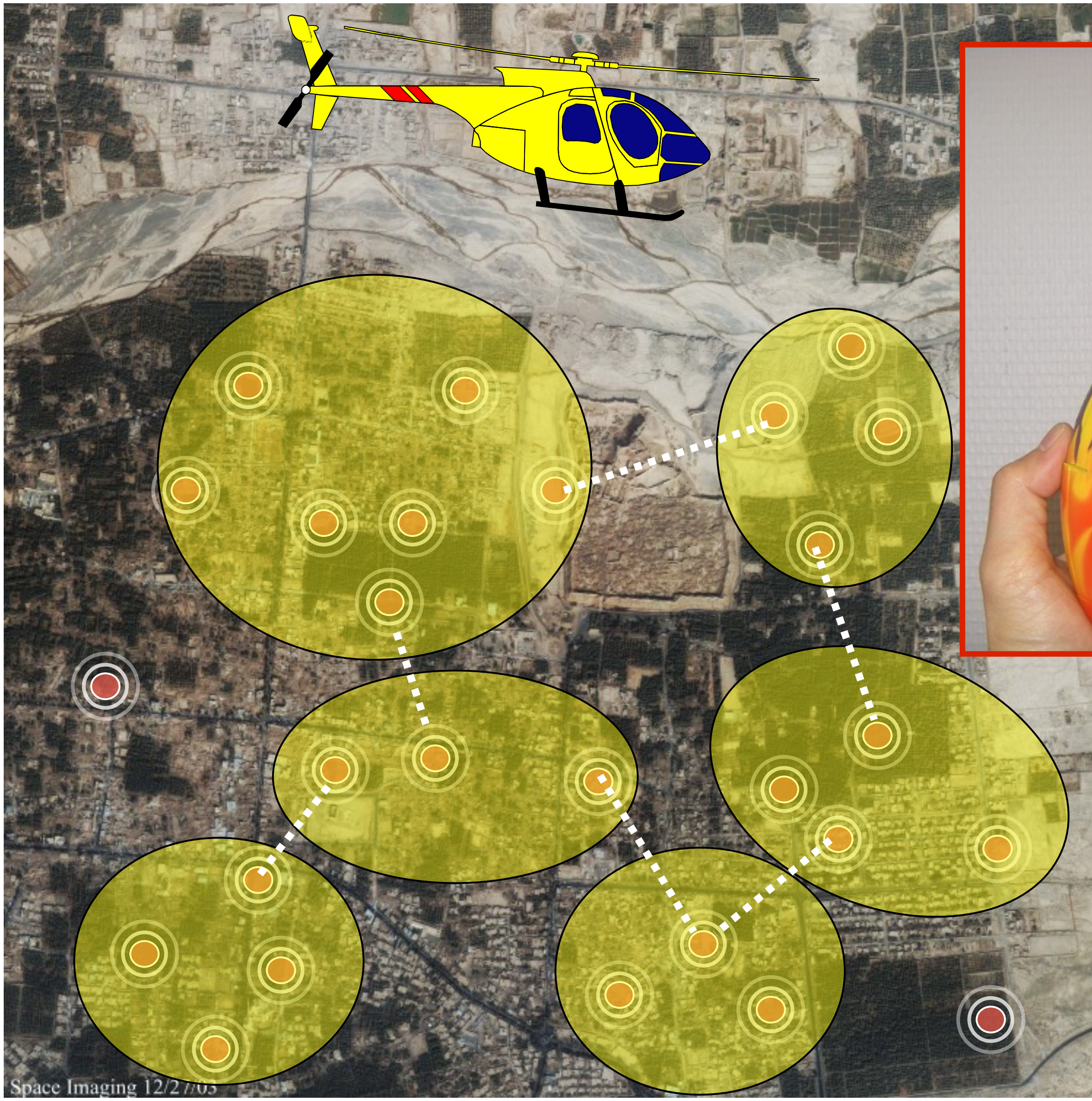
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France

Search & Rescue

Imote2



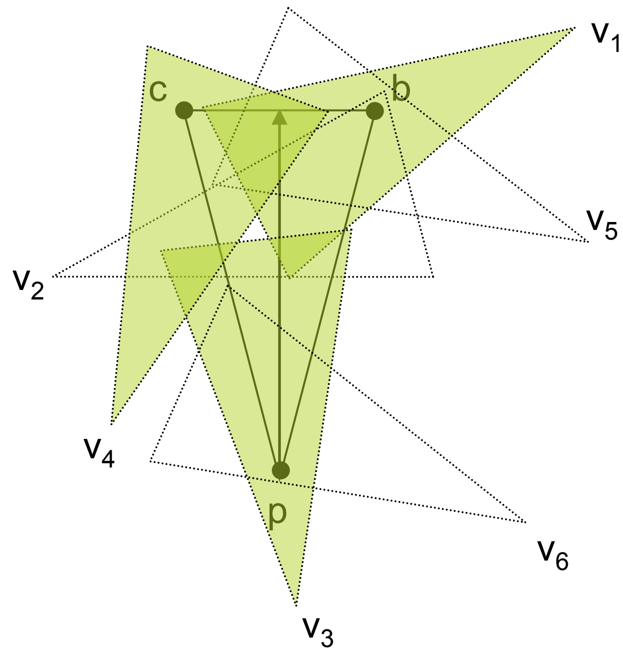
Multimedia board



Node's cover set

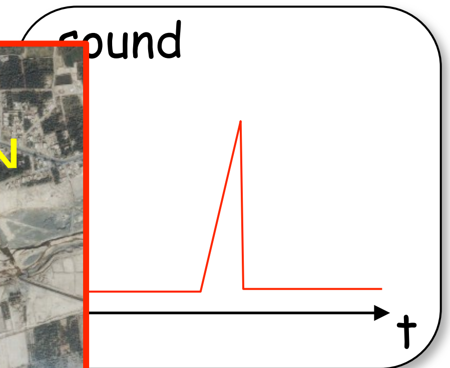
Co(V)= {
{V },
{V_1, V_3, V_4},
{V_2, V_3, V_4},
{V_3, V_4, V_5},
{V_1, V_4, V_6},
{V_2, V_4, V_6},
{V_4, V_5, V_6}
}

|Co(V)| = 7



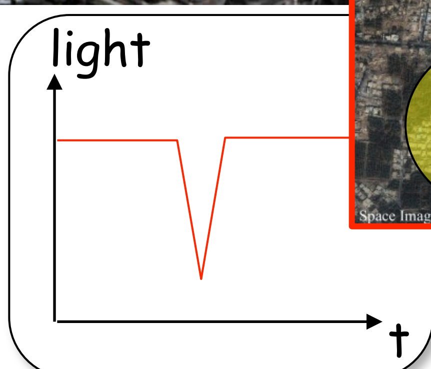
Activity detection

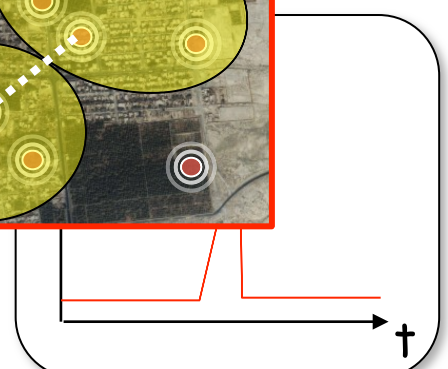
Sound could serve as detector if the person manifest itself

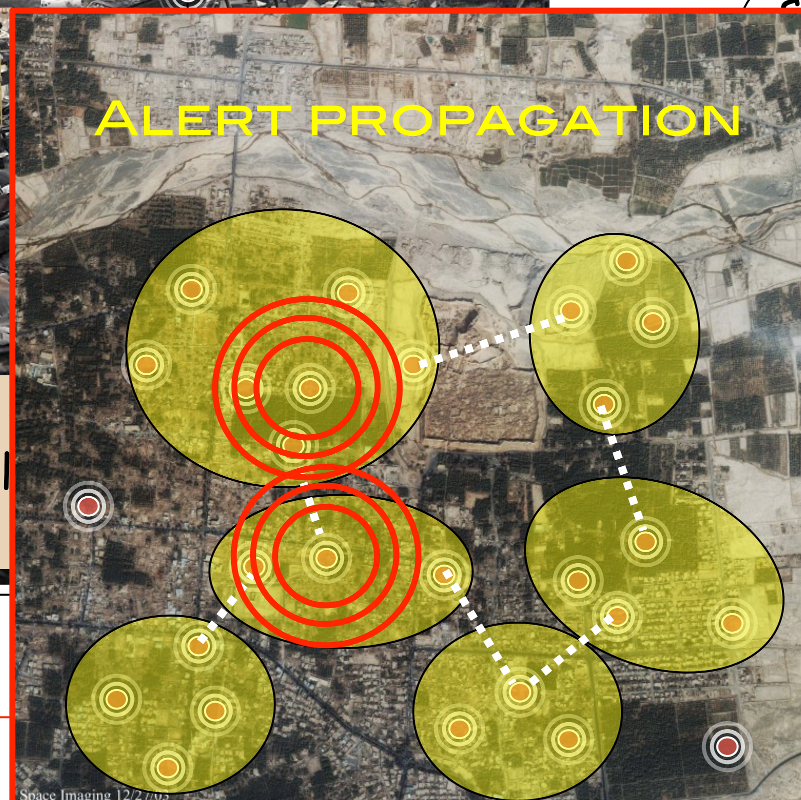


Light sensor could serve as detector if the person rapidl occult the sensor

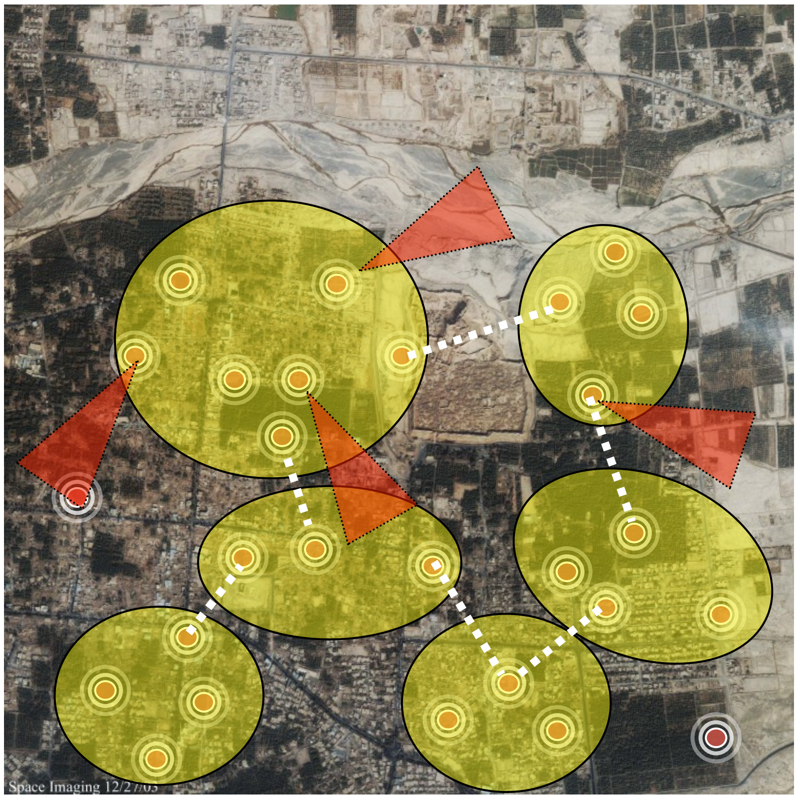
ve as shaked







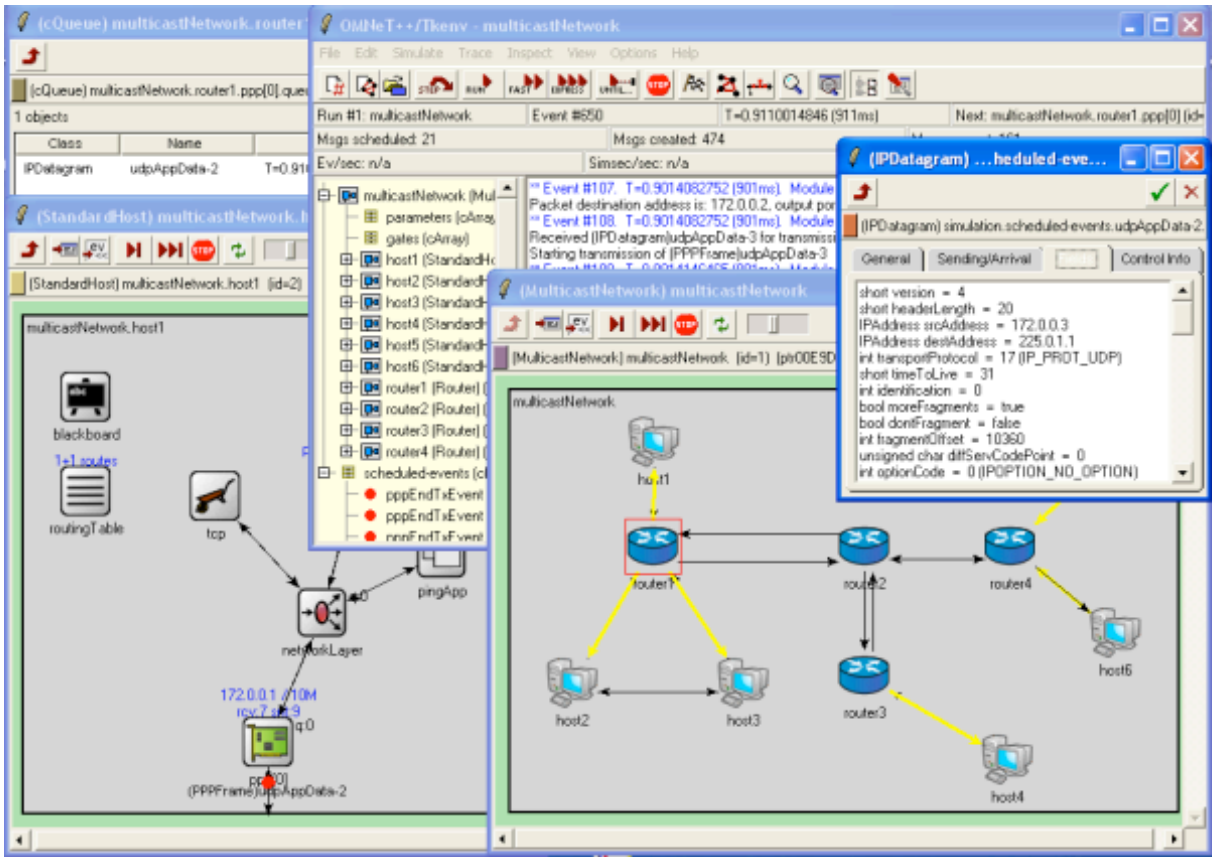
GET IMAGES FROM DEPLOYED SENSORS



Simulation Environment

OMNET++

- Underlying simulation framework, use version 4.1



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

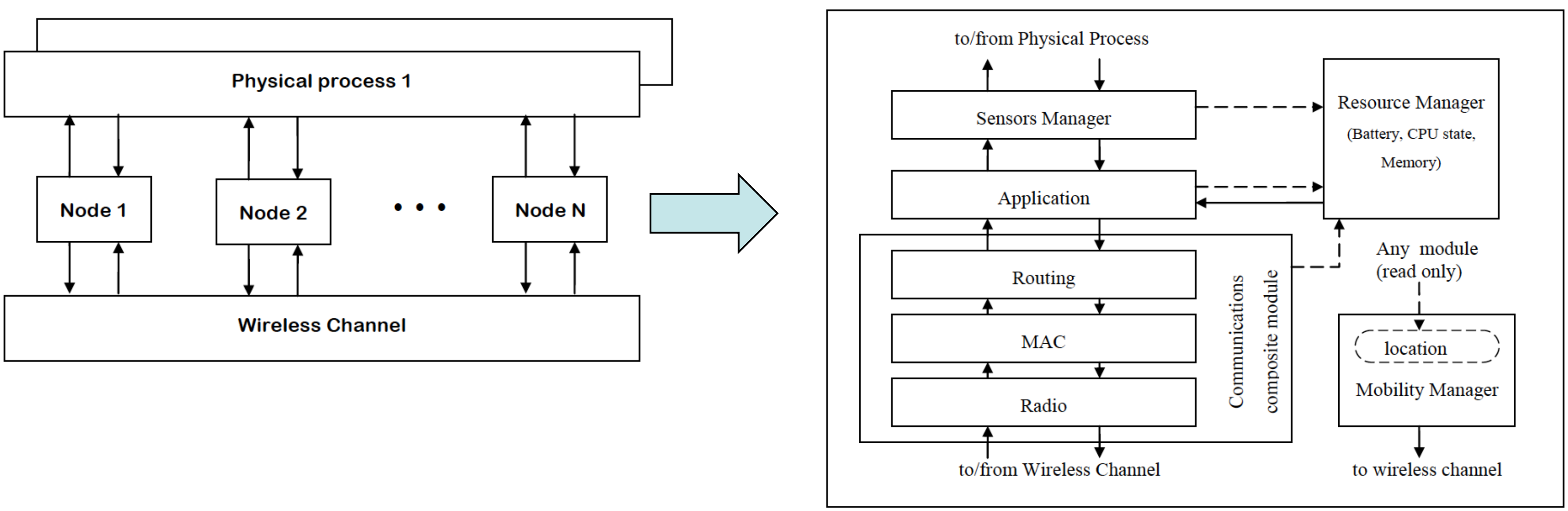
C++ based

Graphic interface

INET FRAMEWORK INETMANET veins ReaSE MiXiM Castalia NICTA

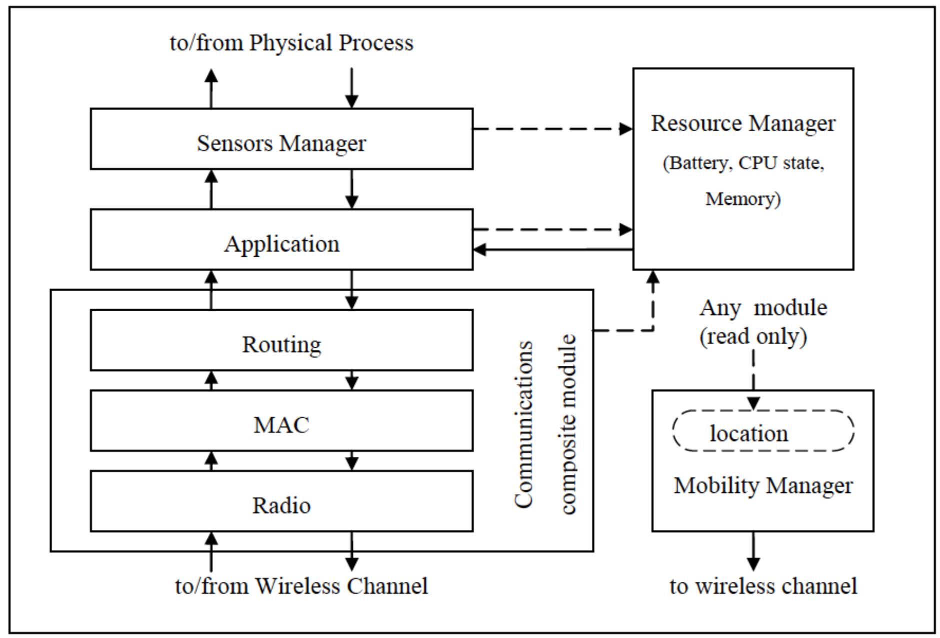
Castalia

- 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



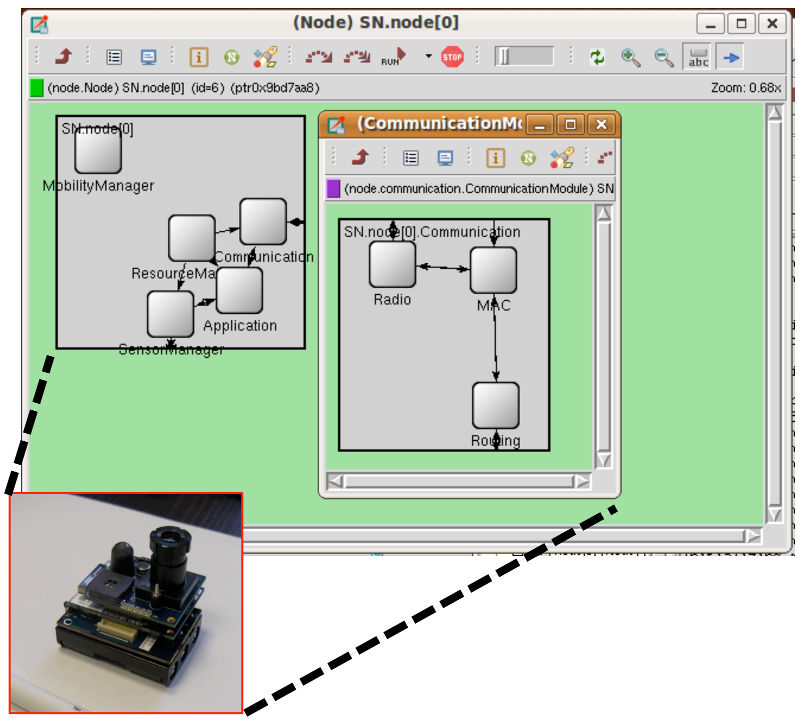
Contrib:
- AODV
- LEACH

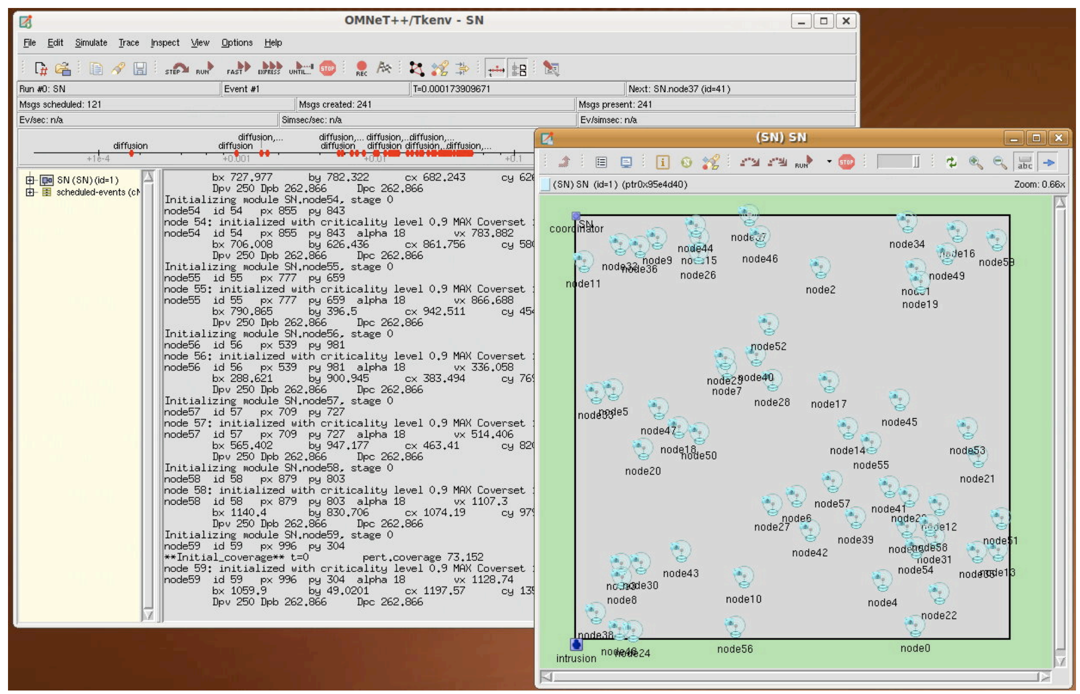
LIUPPA:
- GPSR

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

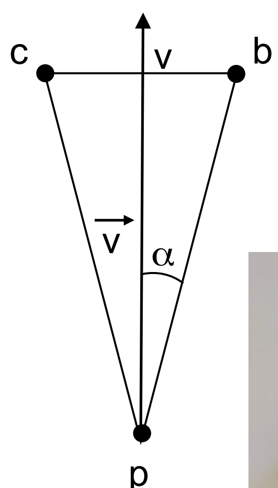
videoSensor model (2)

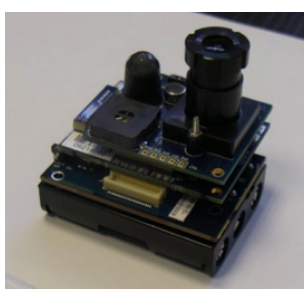
- Many parameters can be changed without recompiling (omnetpp.ini file)
- Compilation flags are used for selecting some global behaviors



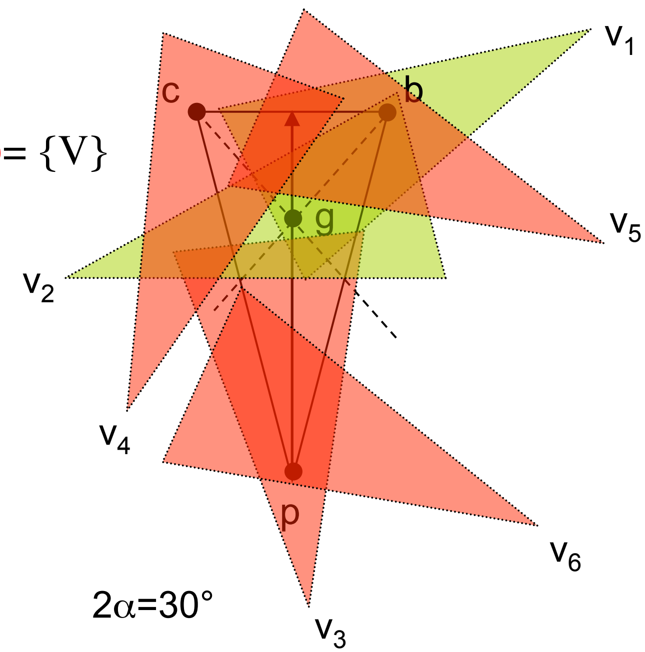


Coverset Computation

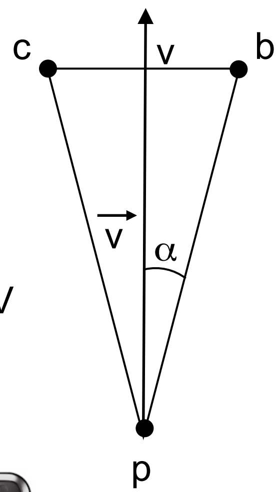




$\textbf{Co(V)} = \{V\}$

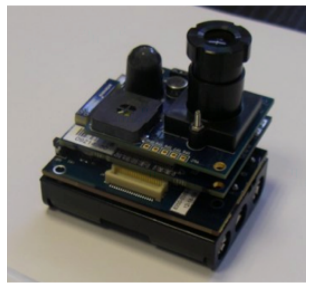


Finding v's cover set

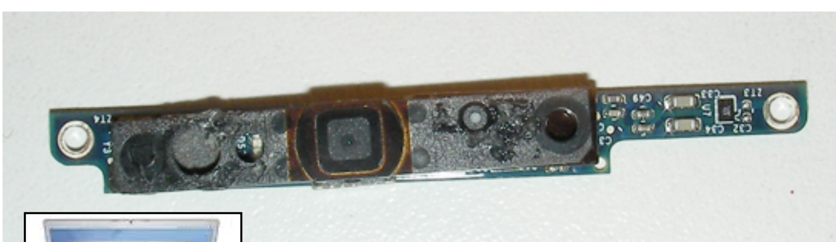


2α=AoV

AoV=20°

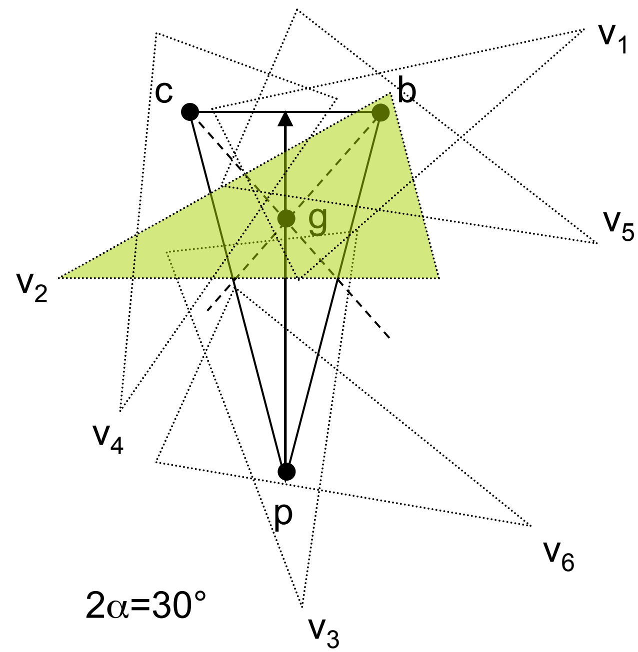


AoV=38°



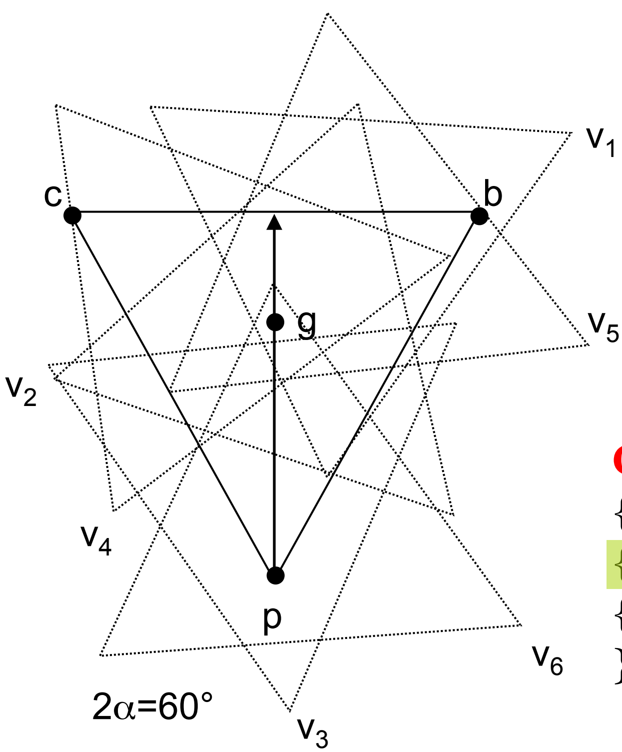
AoV=31°

P = {v ∈ N(V) : v covers the point "p" of the FoV}
B = {v ∈ N(V) : v covers the point "b" of the FoV}
C = {v ∈ N(V) : v covers the point "c" of the FoV}
G = {v ∈ N(V) : v covers the point "g" of the FoV}

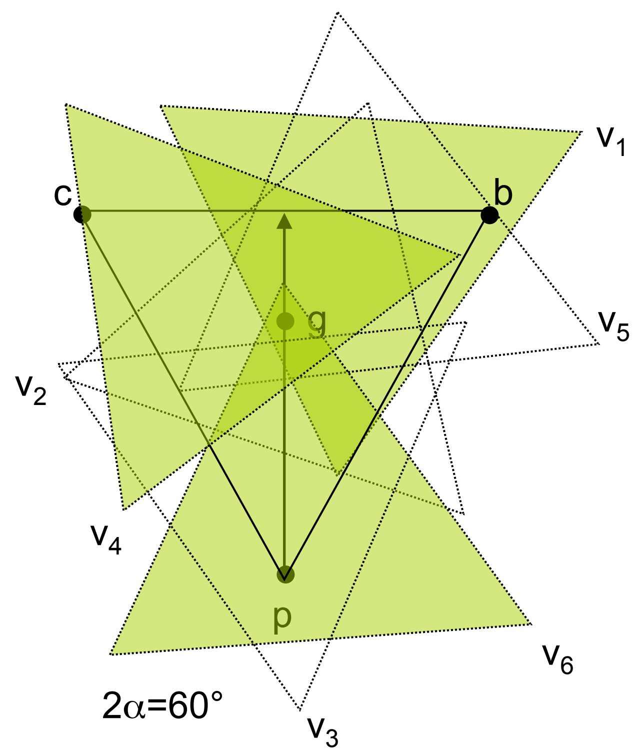


PG={P∩G}
BG={B∩G}
CG={C∩G}
Co(v)=PG×BG×CG

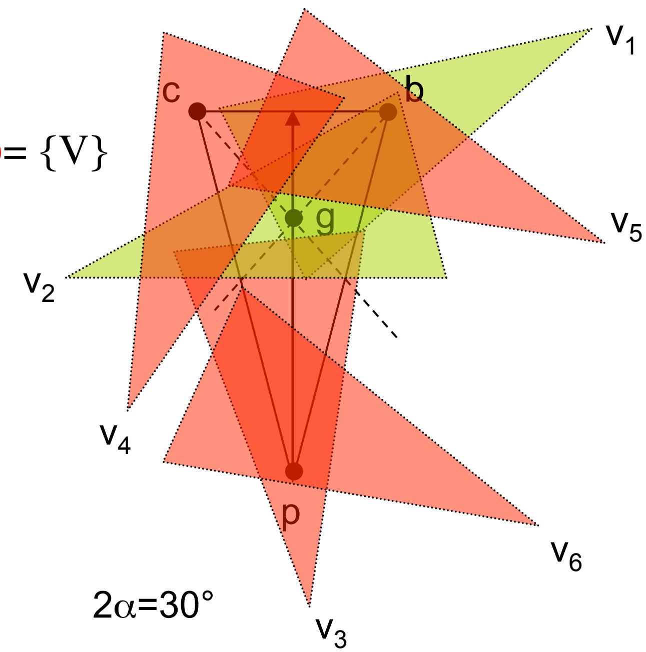
Large Angle of View



Co(V)= {
{V },
{V_1, V_4, V_6},
{V_4, V_5, V_6}
}

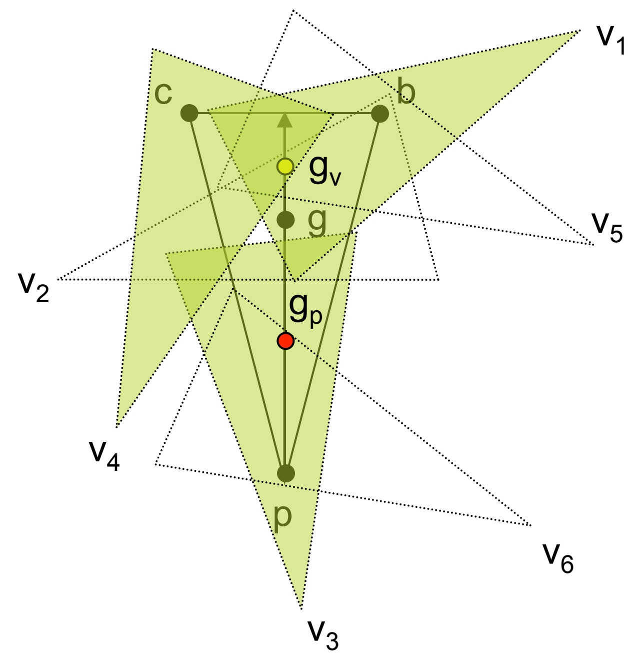


Small Angle of View



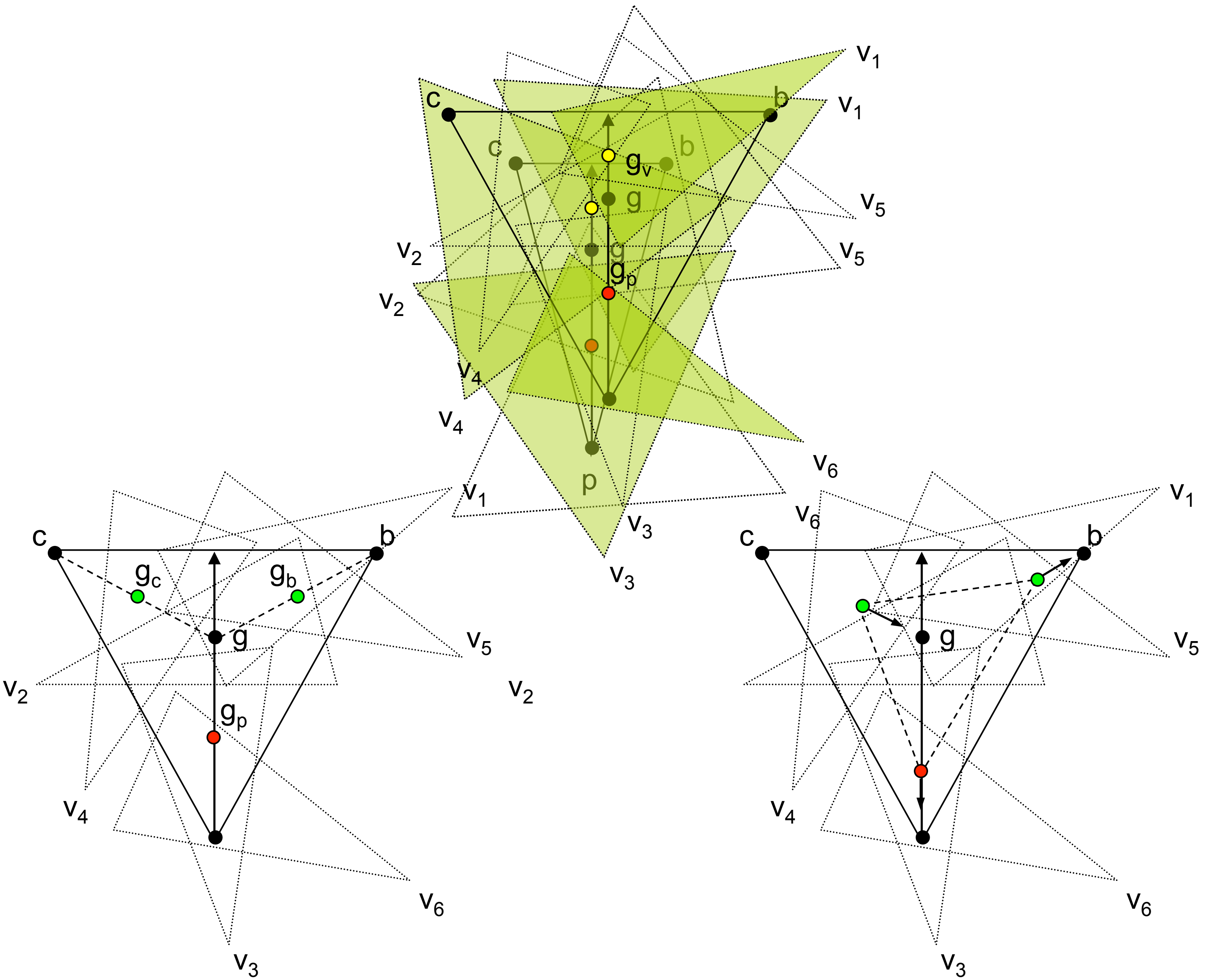
Co(V)= {V}

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

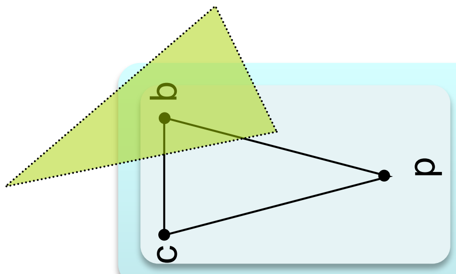


$PG=\{P \cap g_p\}$
$BG=\{B \cap g_v\}$
$CG=\{C \cap g_v\}$
$Co(v)=PG \times BG \times CG$

Heterogeneous AoV

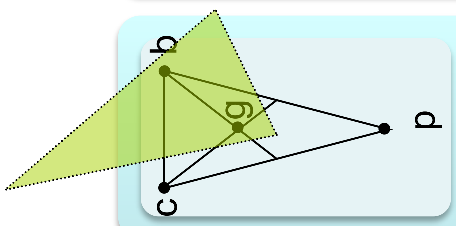


Comparison of coverset strategies



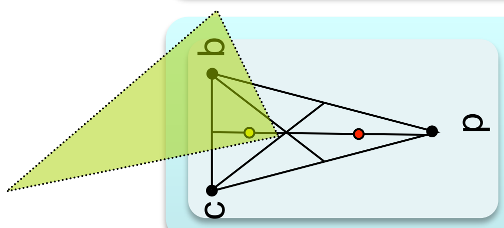
COV_{woG}

- Only triangle's points, without point G



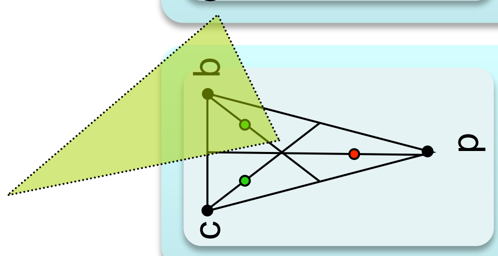
COV_{wG}

- Triangle points, with point G



COV_{waGpv}

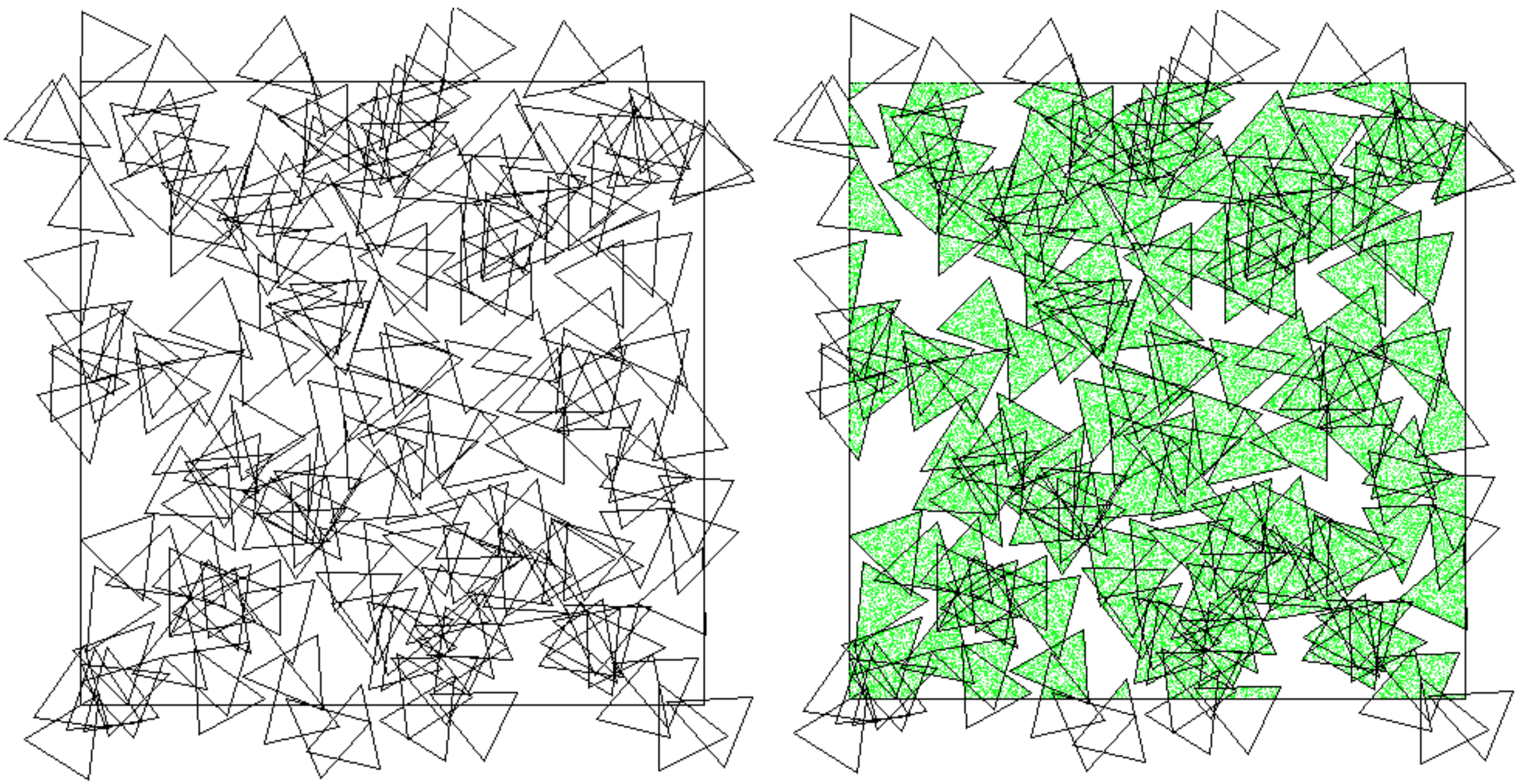
- Triangle points, with alternate gp & gv



COV_{waGbc}

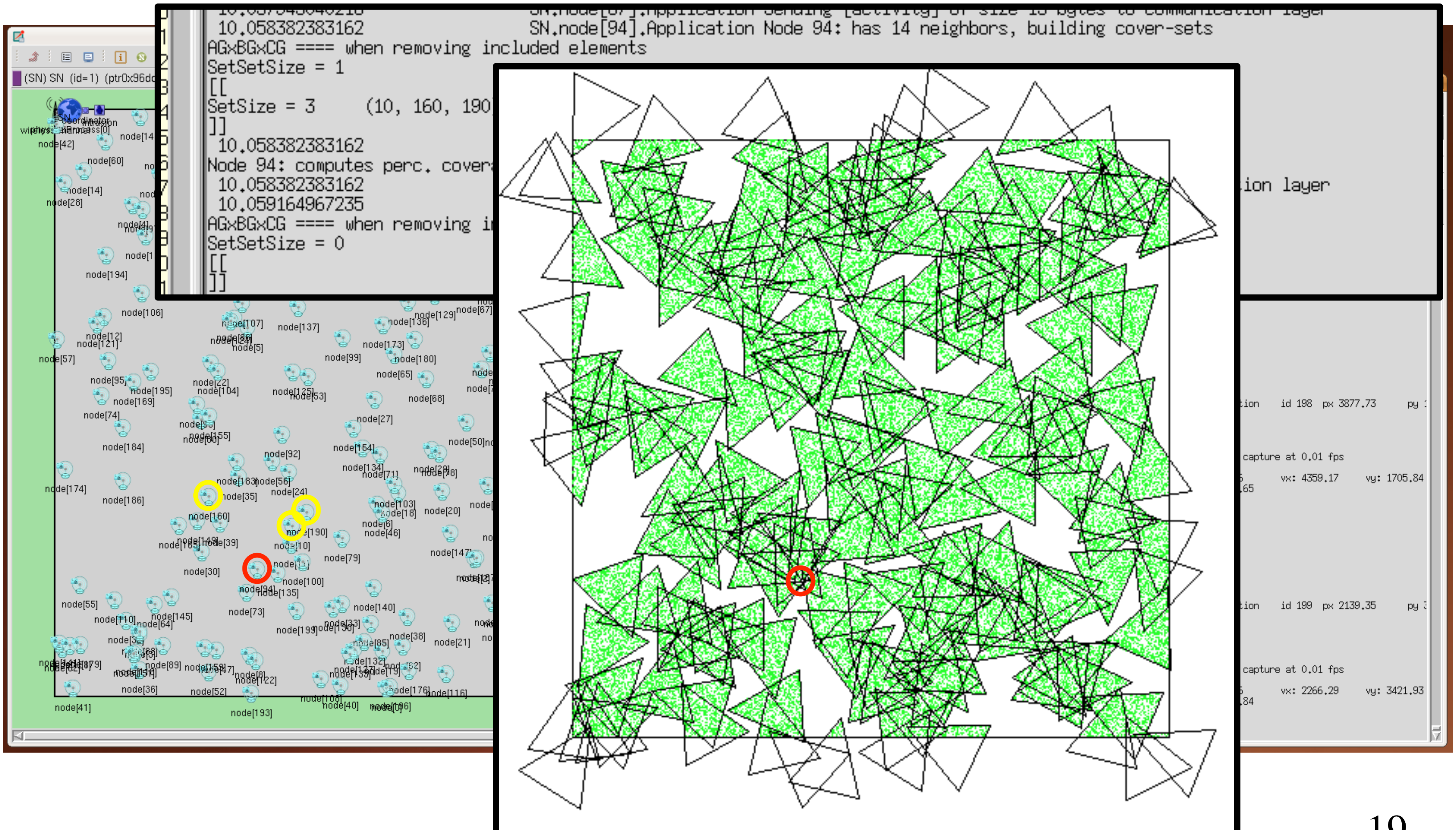
- Triangle points, with alternate gp, gb & gc

Node FoV visualization

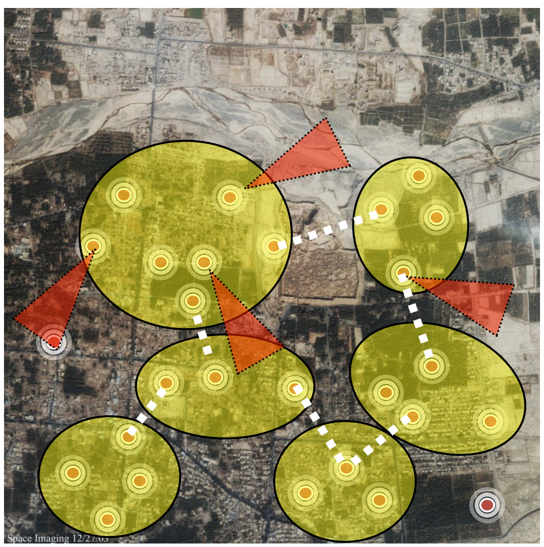


200 sensor nodes in an 400mx400m area

Start simulation



Scheduling image sensors



First of all: Don't miss important events!

Whole understanding of the scene is wrong!!!

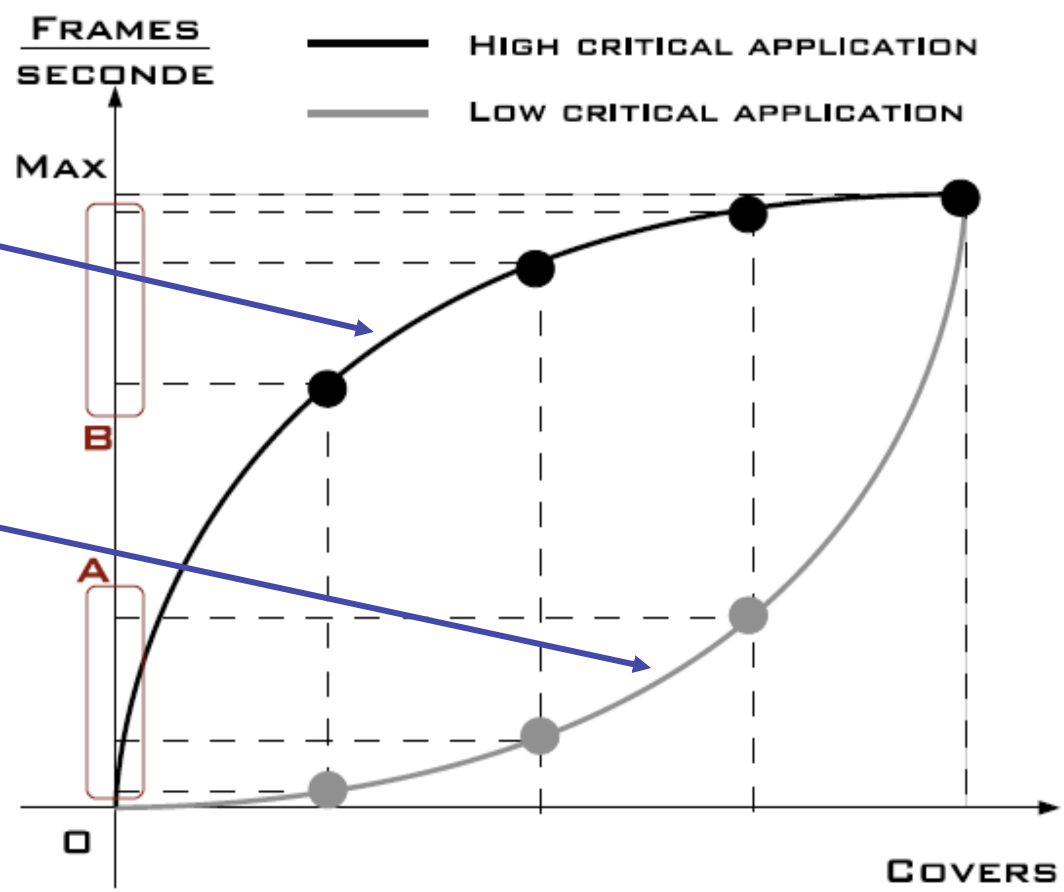
What is captured

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

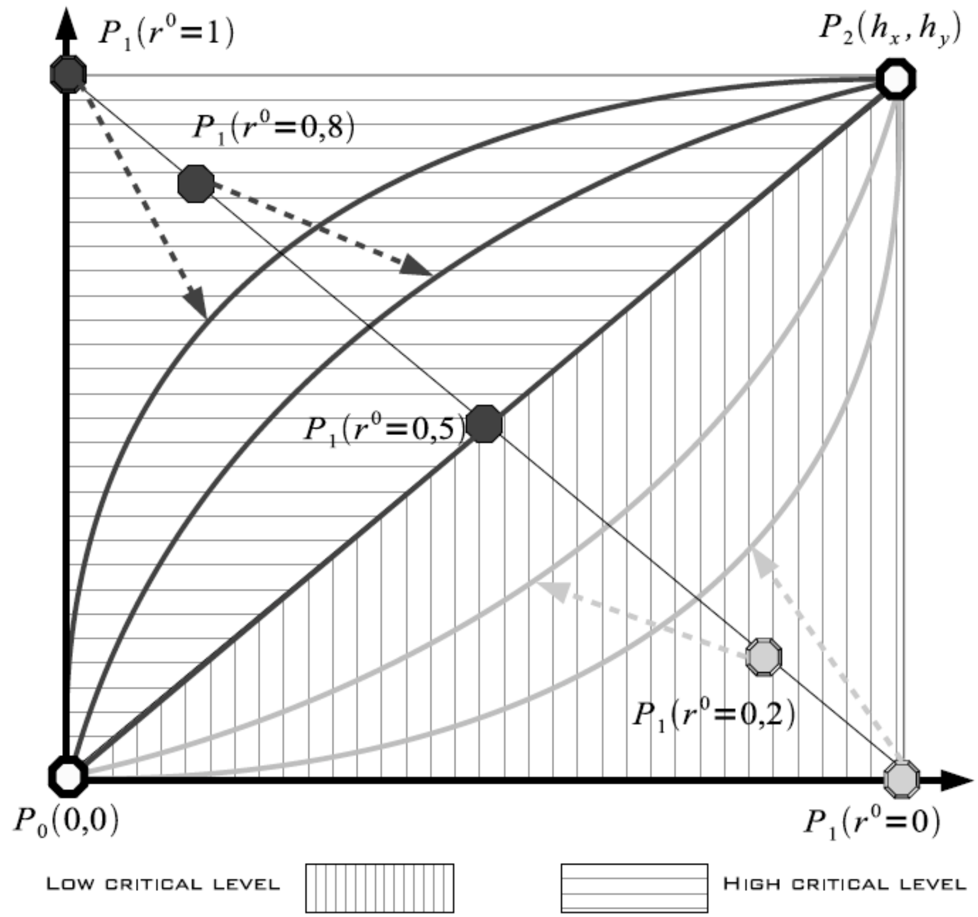
Criticality model (1)

- 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



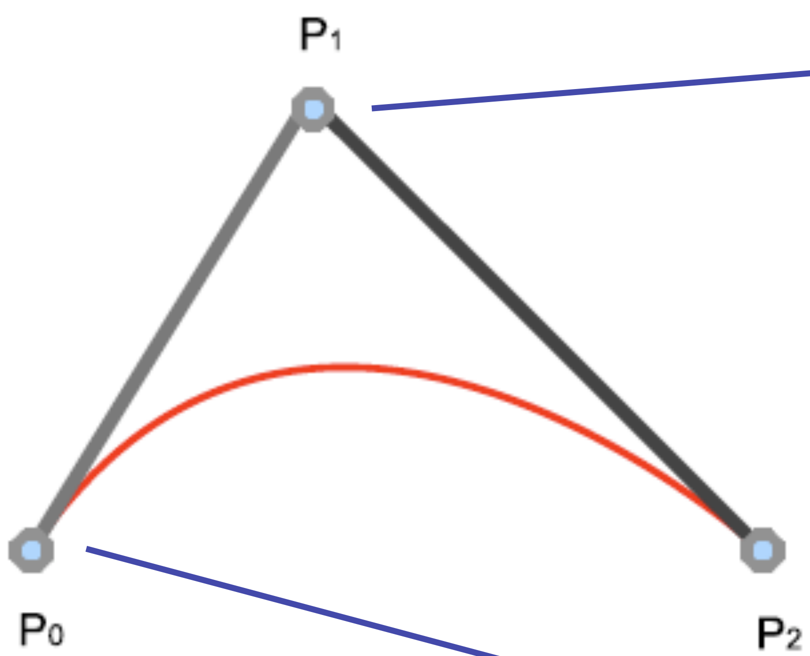
Criticality model (2)

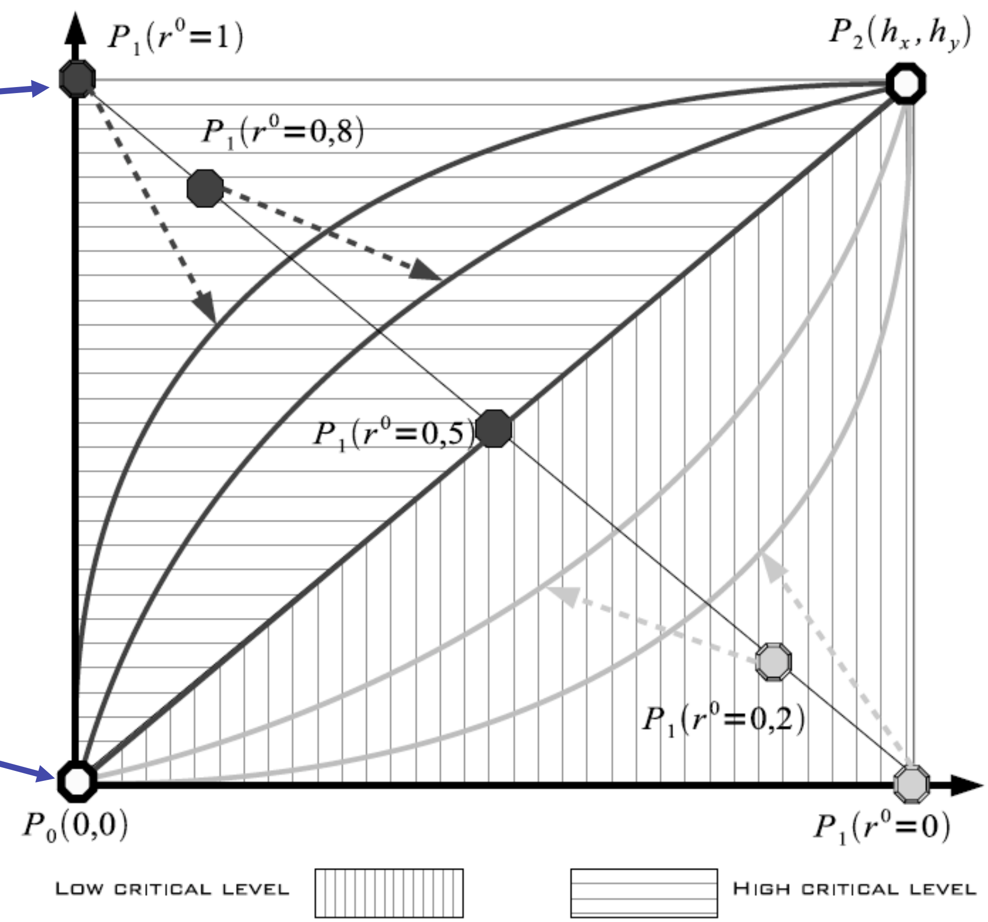
- r^o can vary in [0,1]
- Behavior functions (BV) defines the capture speed according to r^o
- $r^o < 0.5$
  - Concave shape BV
- $r^o > 0.5$
  - Convex shape BV
- We propose to use Bezier curves to model BV functions



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

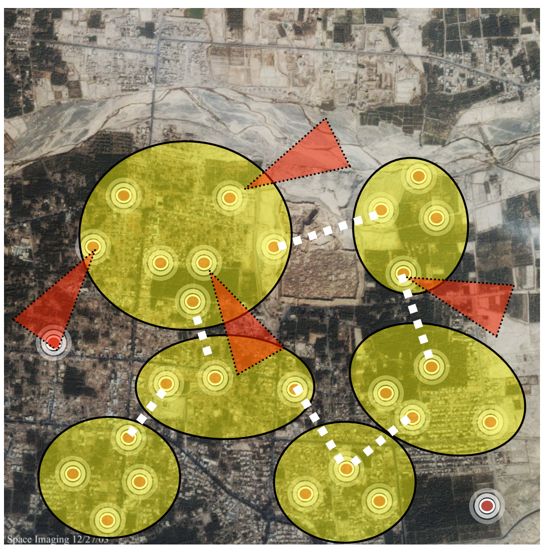
N=6
$P_2(6,6)$

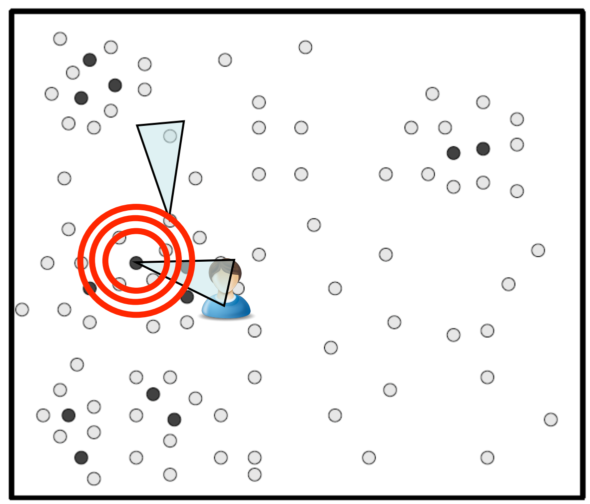
r^0 \ $\|Co(v)\|$	1	2	3	4	5	6
0.0	0.05	0.20	0.51	1.07	2.10	6.00
0.2	0.30	0.73	1.34	2.20	3.52	6.00
0.5	1.00	2.00	3.00	4.00	5.00	6.00
0.8	2.48	3.80	4.66	5.27	5.70	6.00
1.0	3.90	4.93	5.49	5.80	5.95	6.00

N=12
$P_2(12,3)$

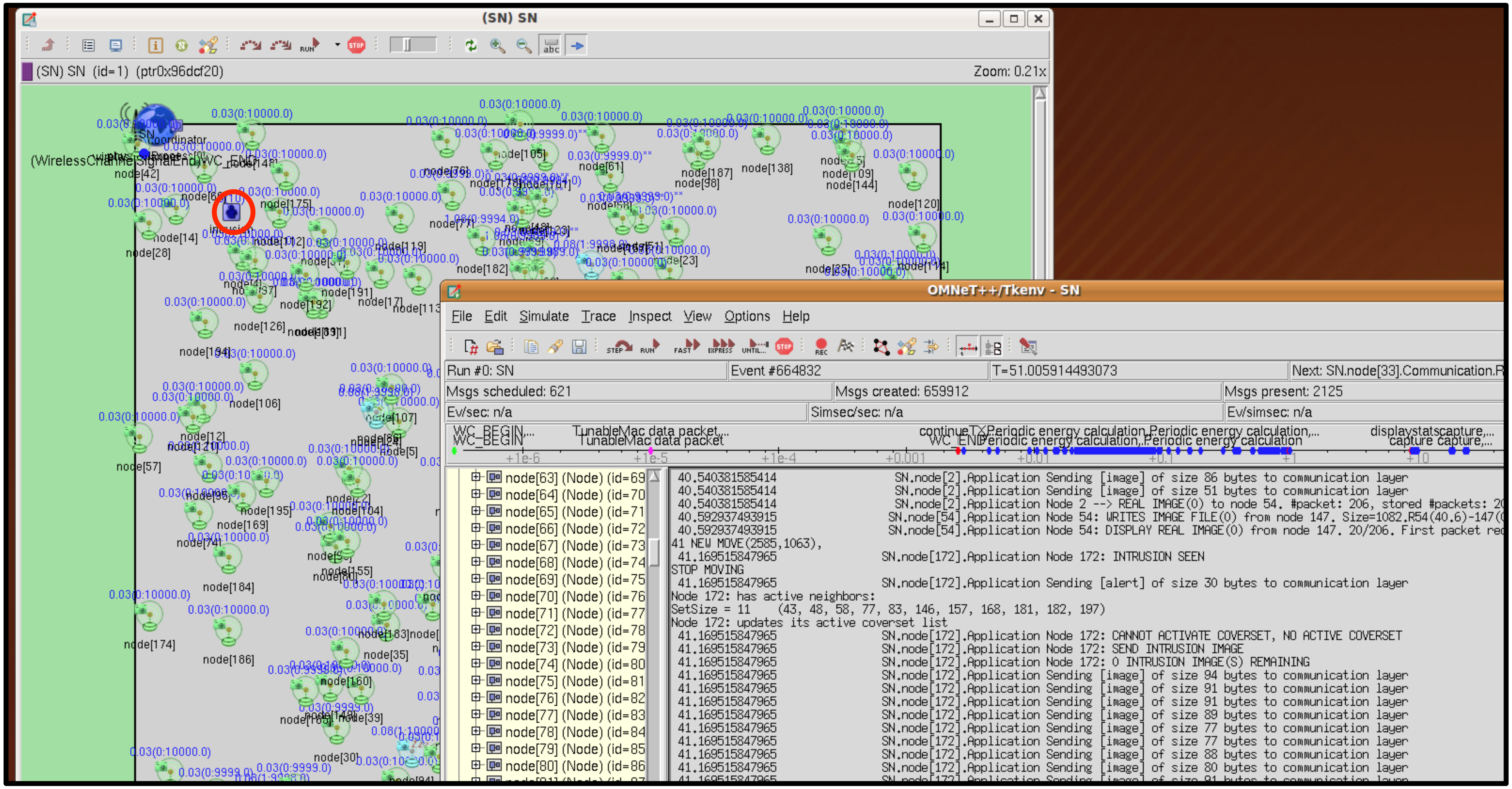
r^0	1	2	3	4	5	6	7	8	9	10	11	12
0	.01	.02	.05	0.1	.17	.26	.38	.54	.75	1.1	1.5	3
.2	.07	.15	.25	.37	.51	.67	.86	1.1	1.4	1.7	2.2	3
.4	.17	.35	.55	.75	.97	1.2	1.4	1.7	2.0	2.3	2.6	3
.6	.36	.69	1.0	1.3	1.5	1.8	2.0	2.2	2.4	2.6	2.8	3
.8	.75	1.2	1.6	1.9	2.1	2.3	2.5	2.6	2.7	2.8	2.9	3
1	1.5	1.9	2.2	2.4	2.6	2.7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

Intrusion detection

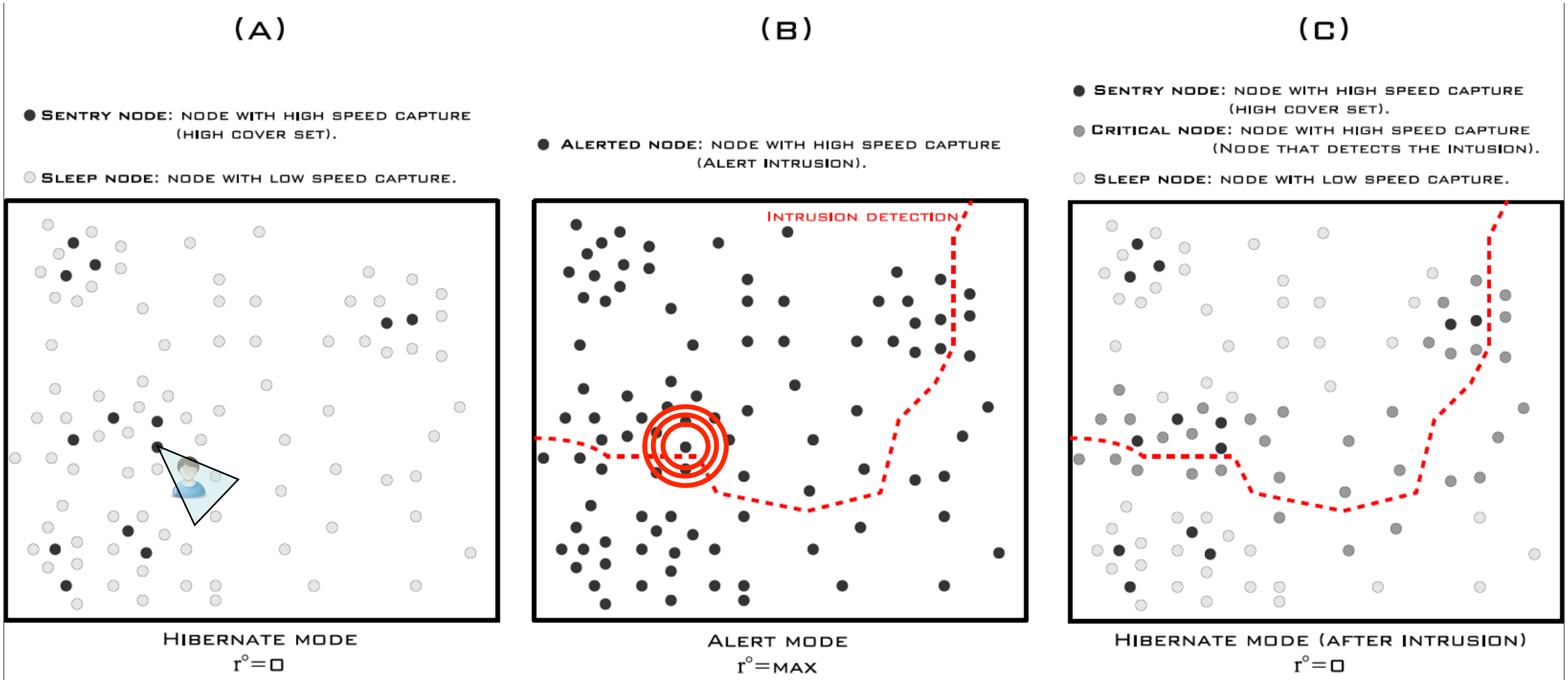




Random Intrusion

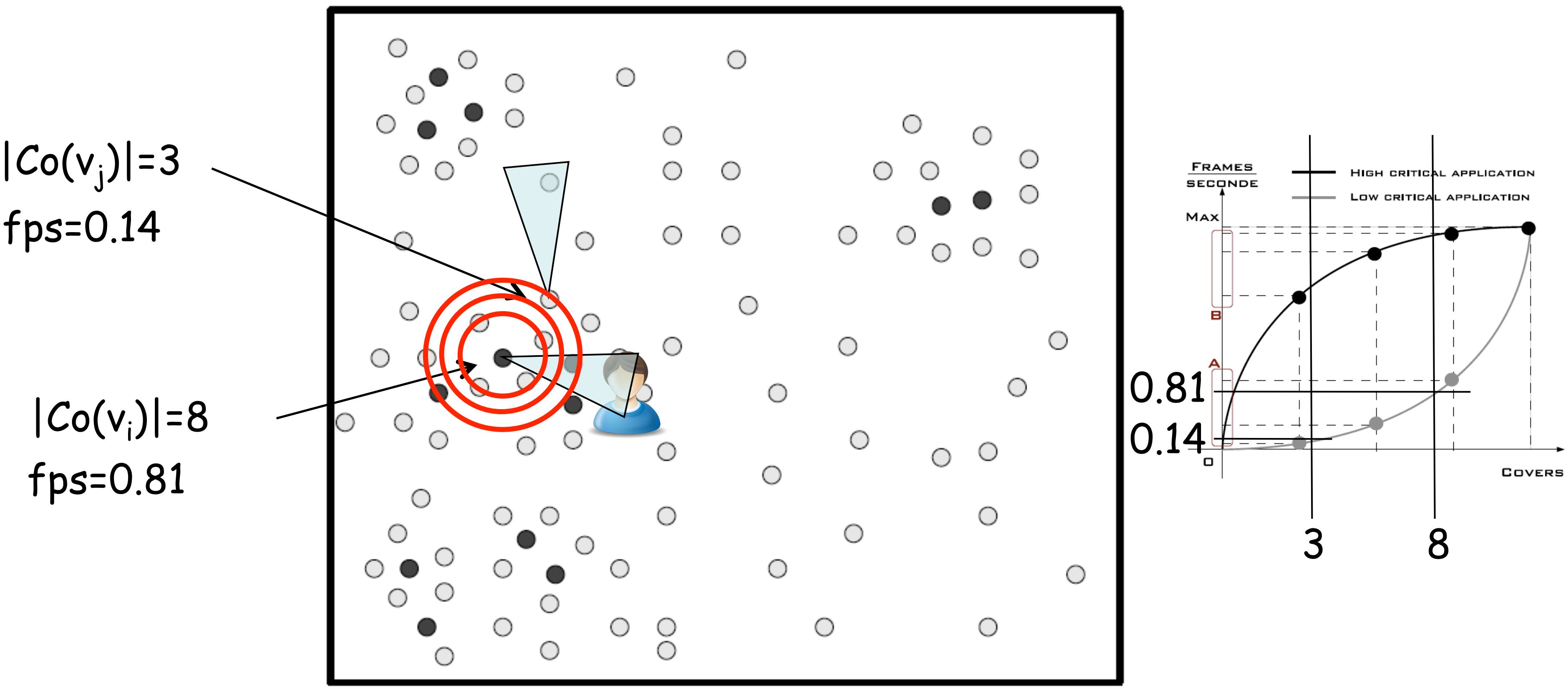


Intrusion detection scenario



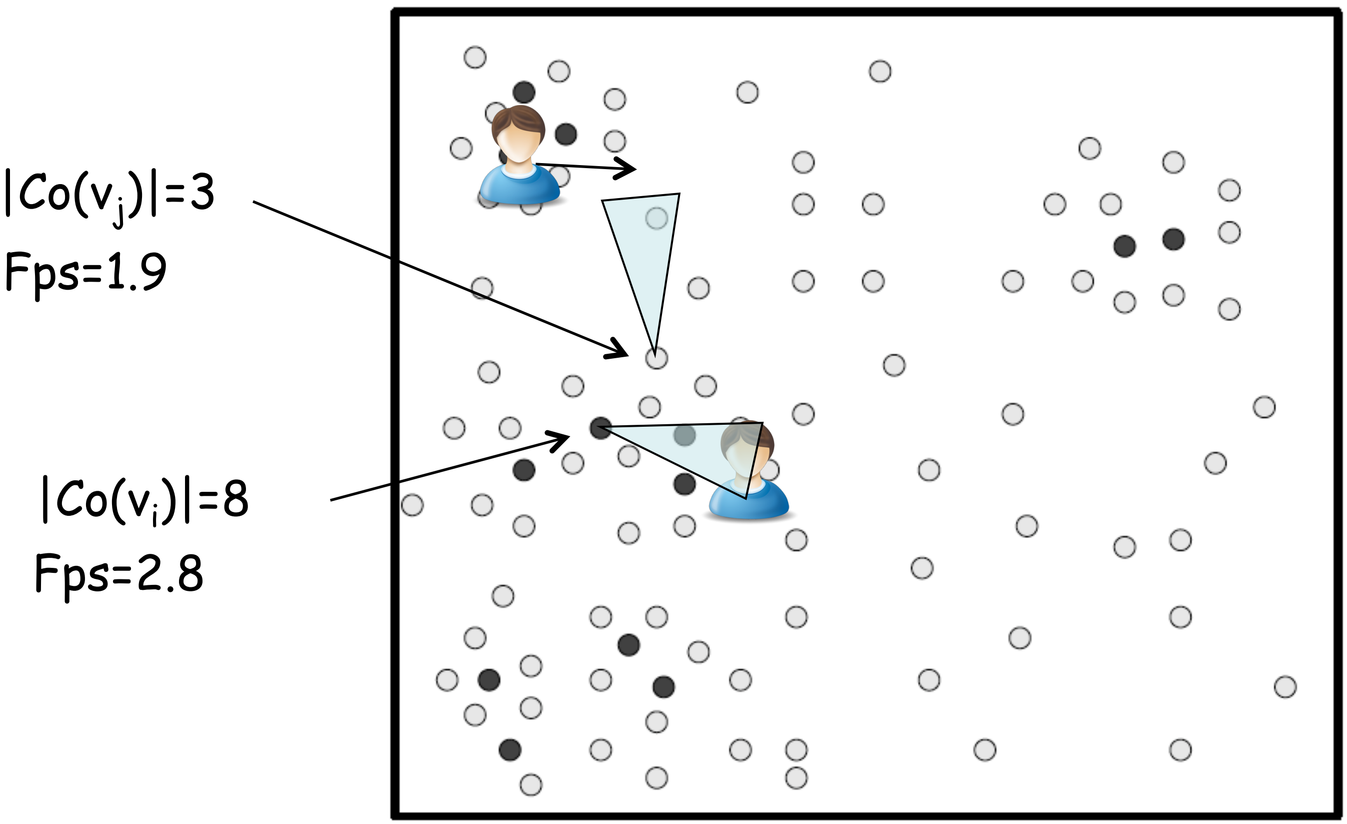
Risk-based scheduling in images (1)

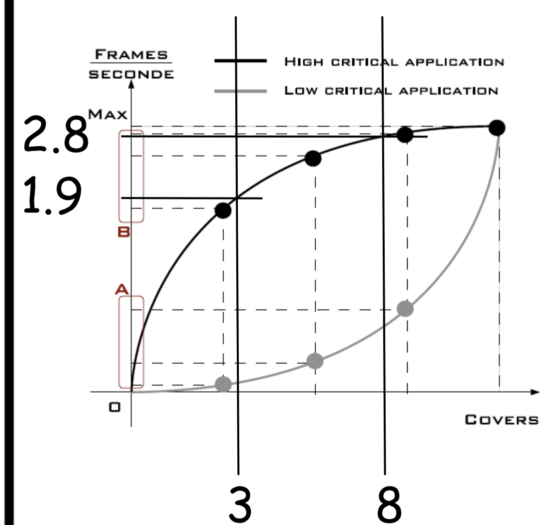
- ❑ $R^o = R^o_{MIN} = 0.1$, $R^o_{MAX} = 0.9$, no alert



Risk-based scheduling in images (2)

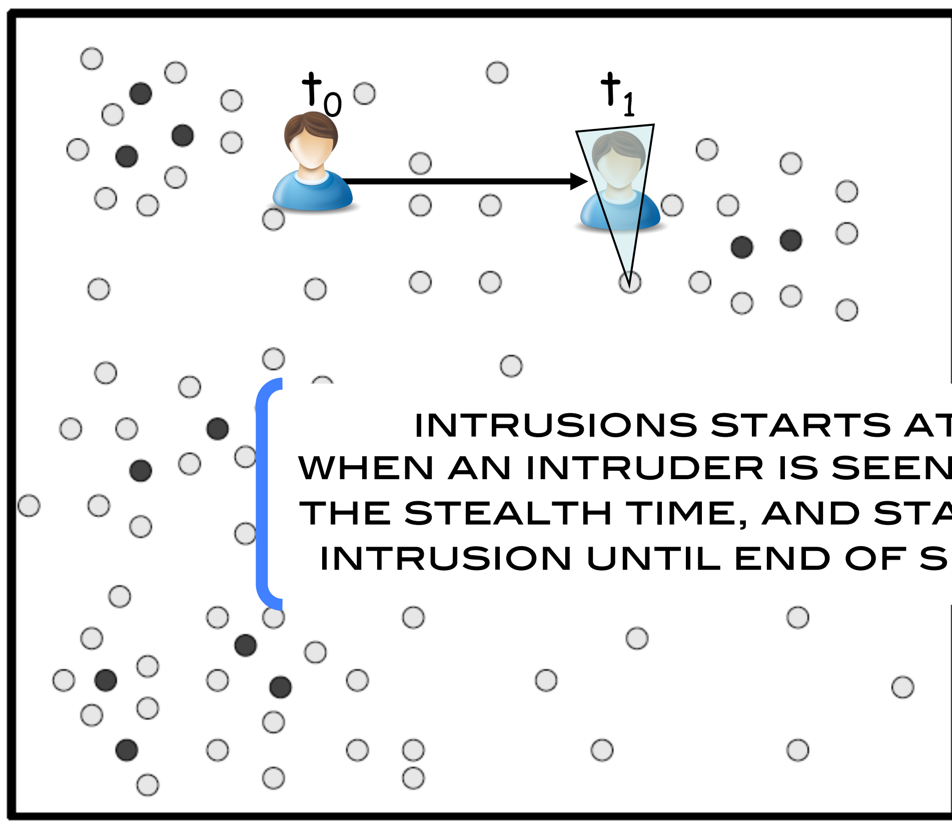
- $R° \rightarrow R° = R°_{MAX} = 0.9$





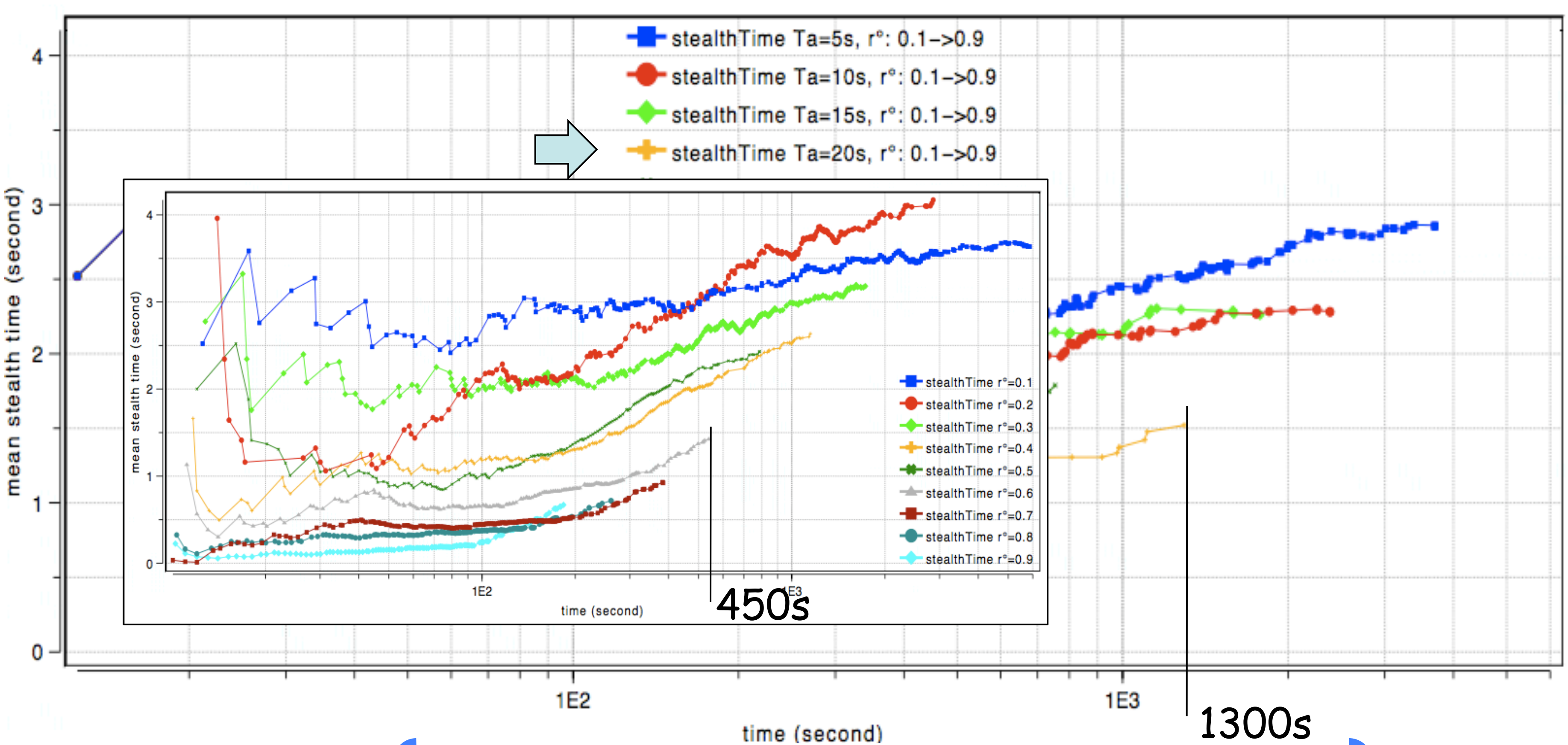
MEAN STEALTH TIME

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



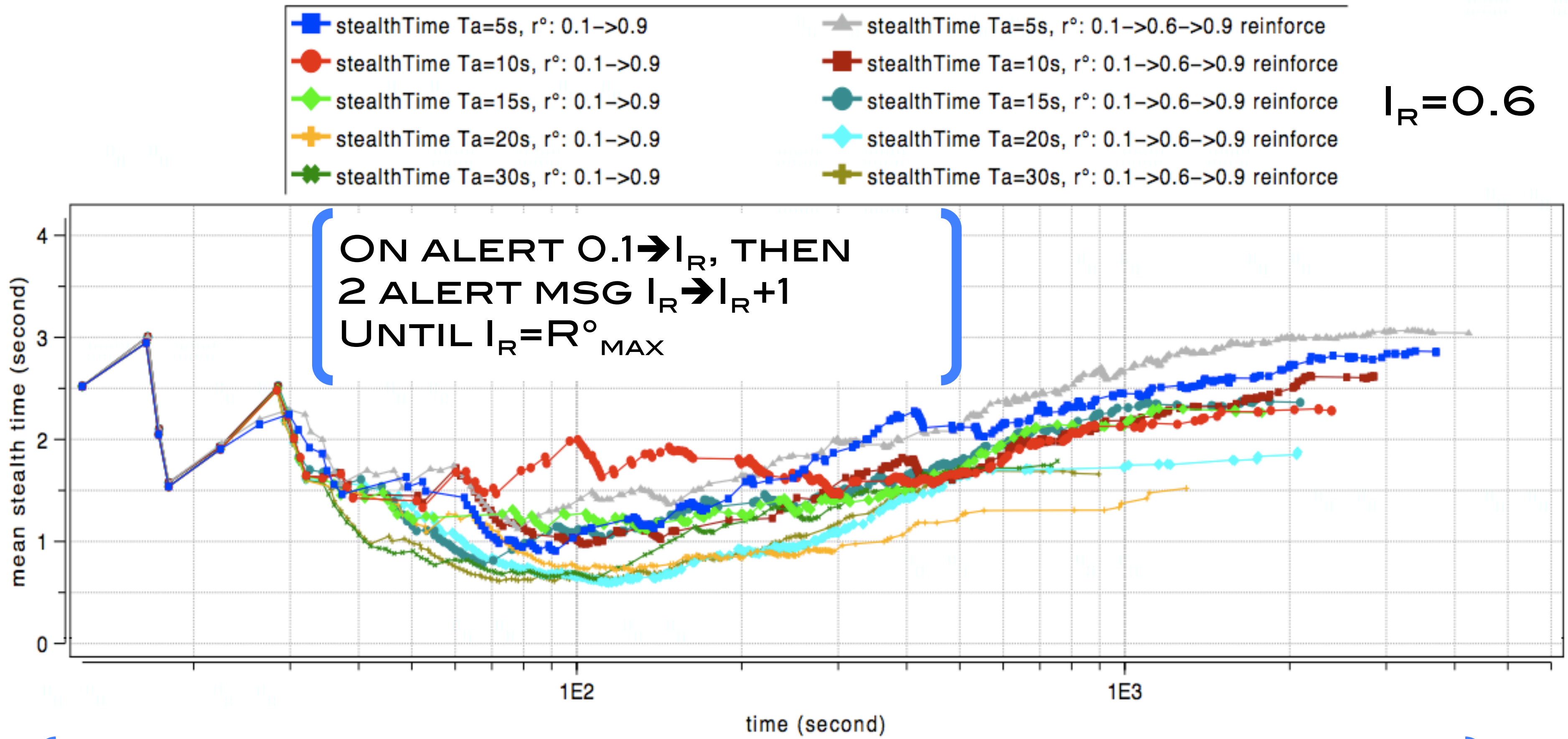
INTRUSIONS STARTS AT T=10S
WHEN AN INTRUDER IS SEEN, COMPUTE THE STEALTH TIME, AND STARTS A NEW INTRUSION UNTIL END OF SIMULATION

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



I_R=0.6

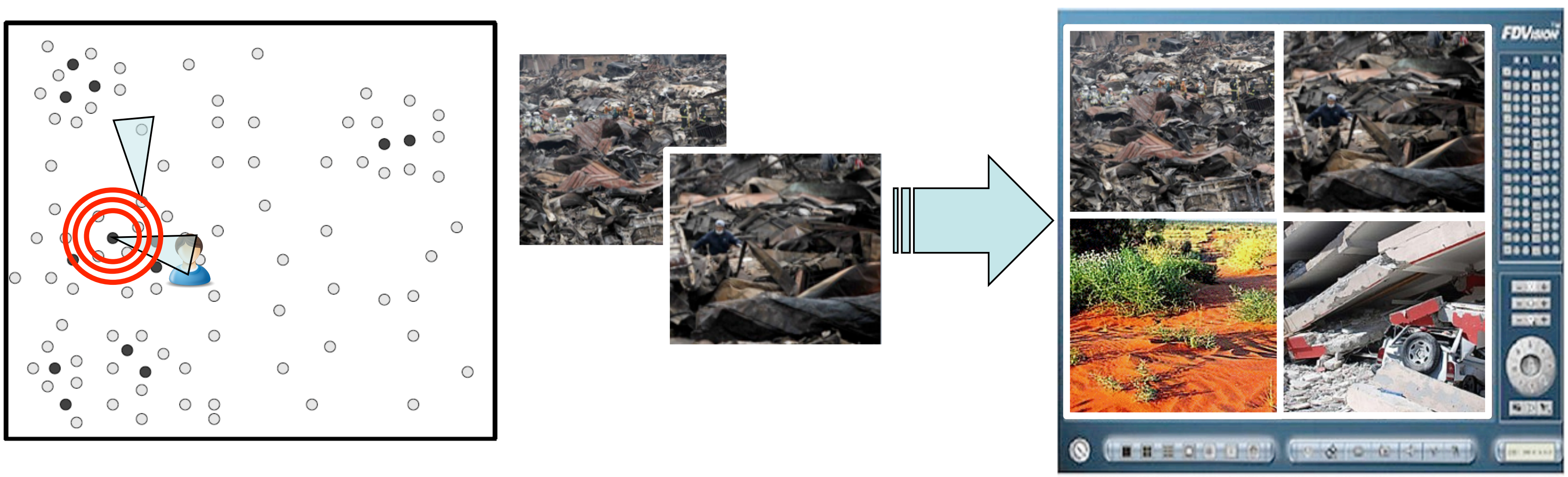
ON ALERT 0.1→I_R, THEN
2 ALERT MSG I_R→I_R+1
UNTIL I_R=$R°_{MAX}$

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

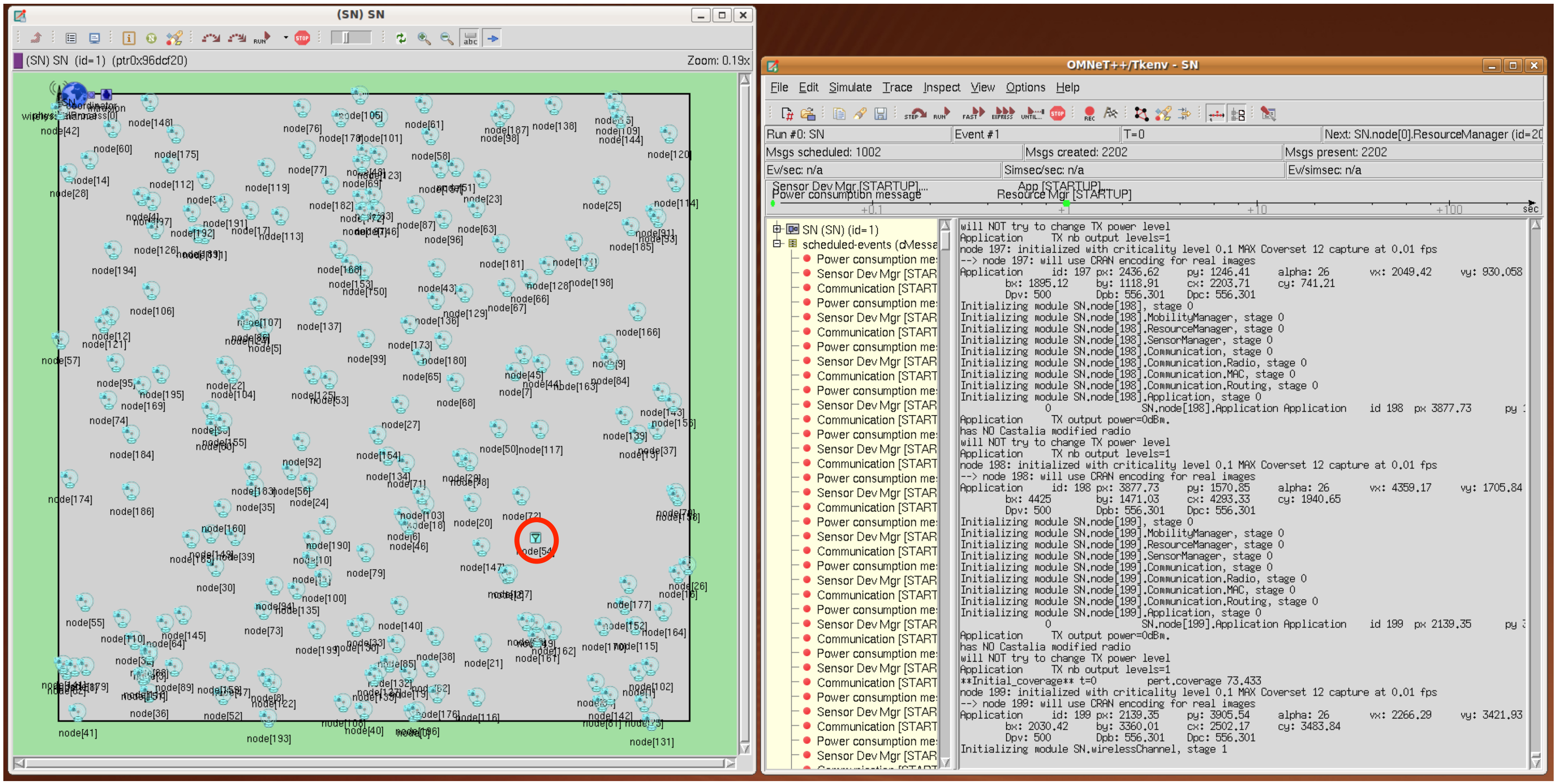
Model parameters scheduling

- 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

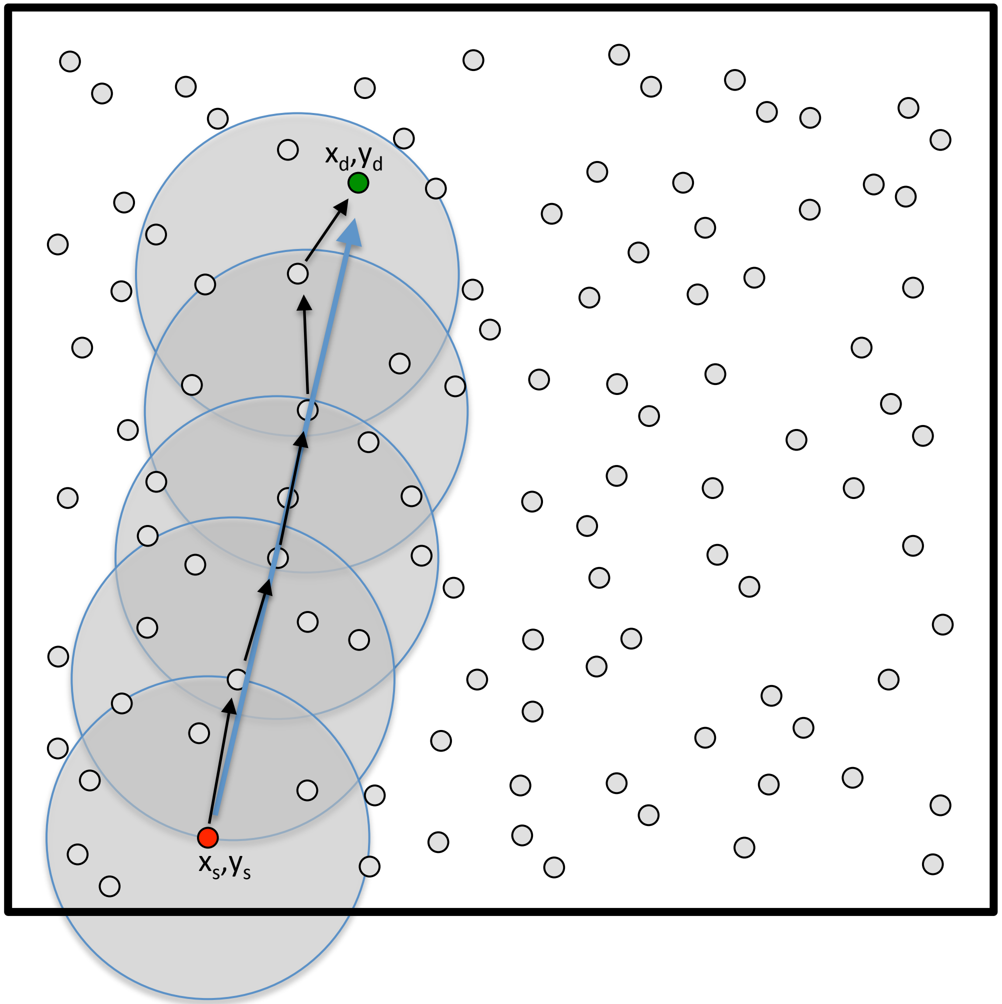
Image transmission, Communication issues



Send images to the sink



Geographic routing
GPSR



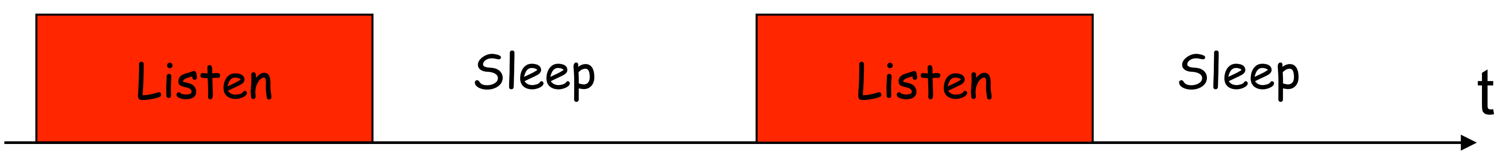
Avoids keeping routing information

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

Reduces route maintenance overhead

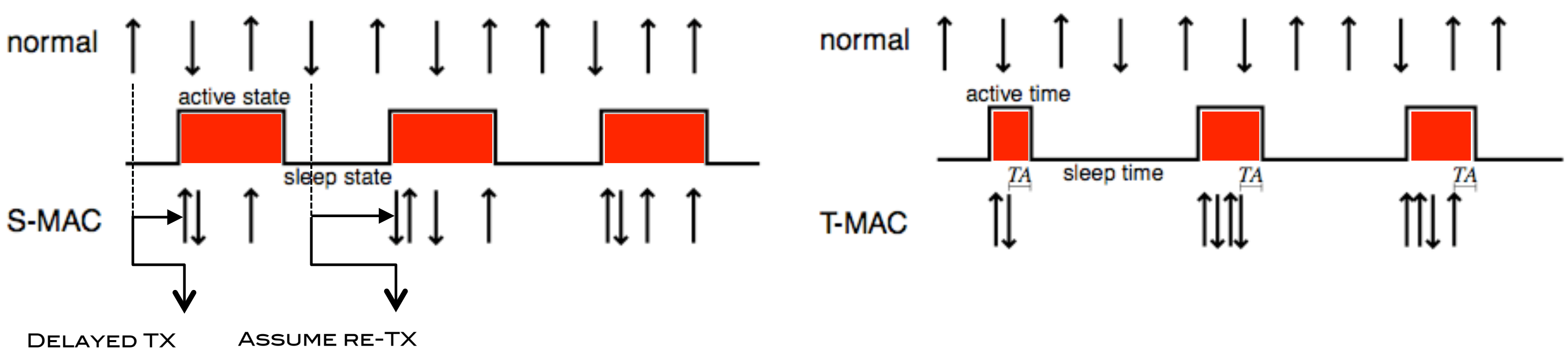
S-MAC - Sensor MAC

- Nodes periodically sleep
- Trades energy efficiency for lower throughput and higher latency
- Sleep during other nodes transmissions
- Needs complex synchronization mechanisms



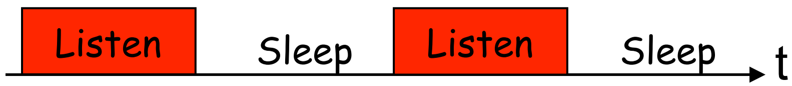
T-MAC - Timeout MAC

- Transmit all messages in bursts of variable length and sleep between bursts
- RTS / CTS / ACK Scheme
- Synchronization similar to S-MAC

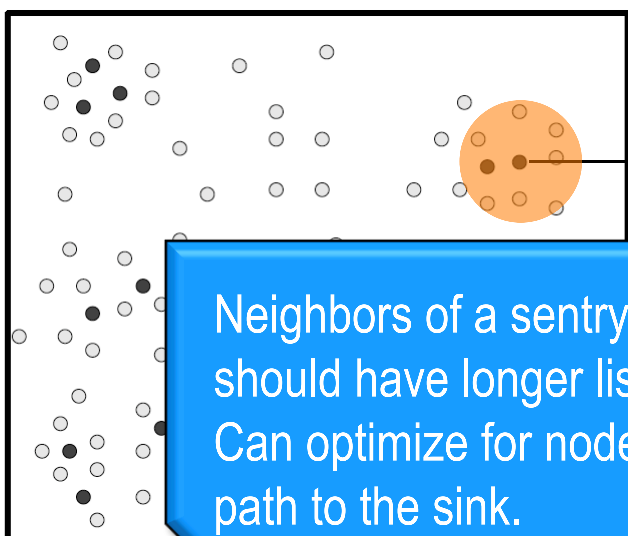


Our current research on MAC layer

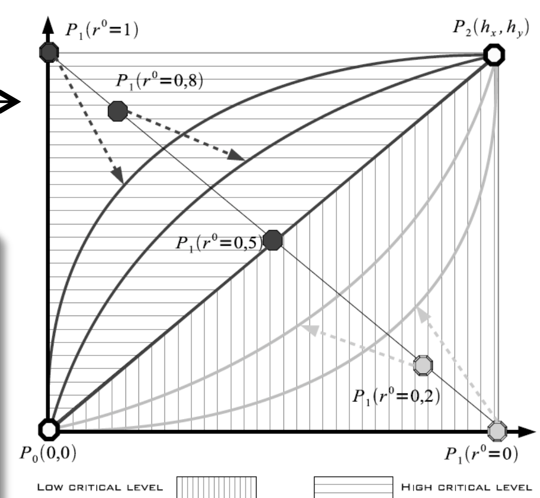
- Duty-cycled MAC (e.g. SMAC)



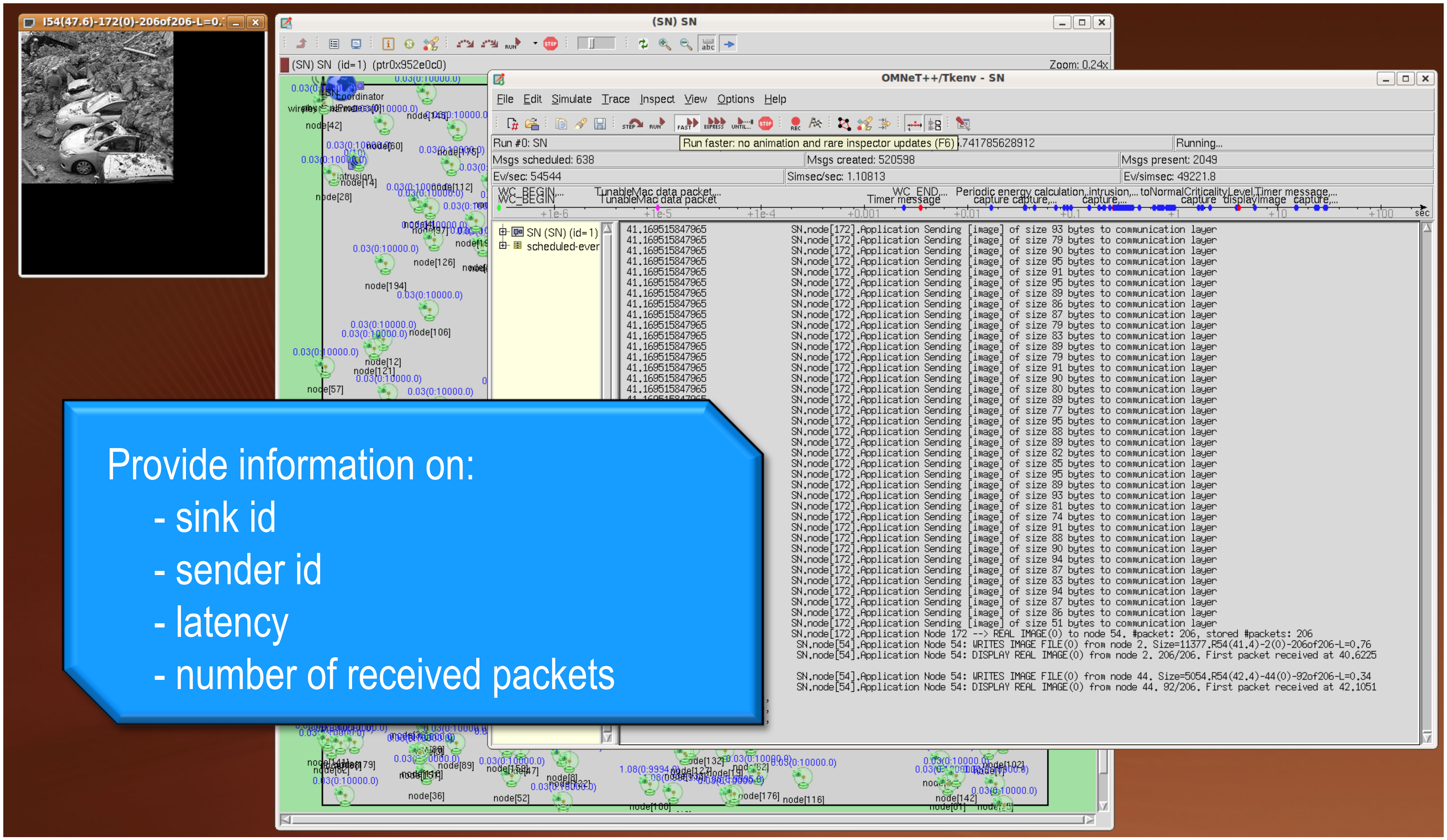
- Link the listening time to the criticality model



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



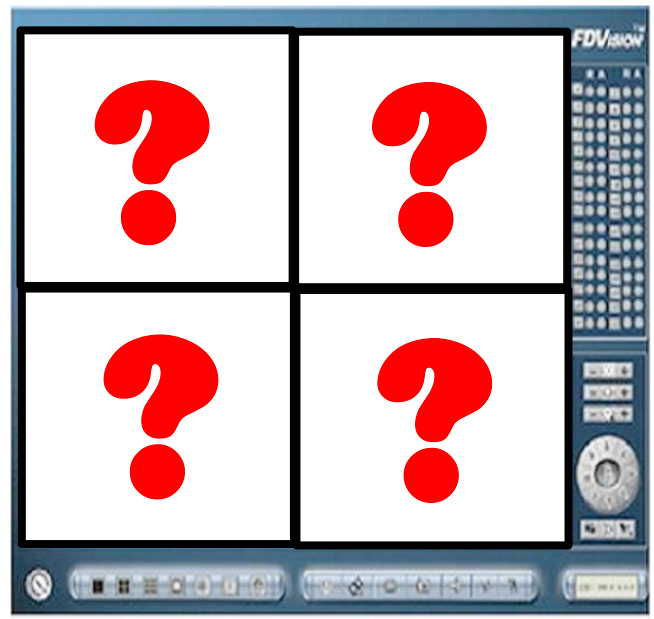
Display images



Provide information on:
- sink id
- sender id
- latency
- number of received packets

End-to-end performances?

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

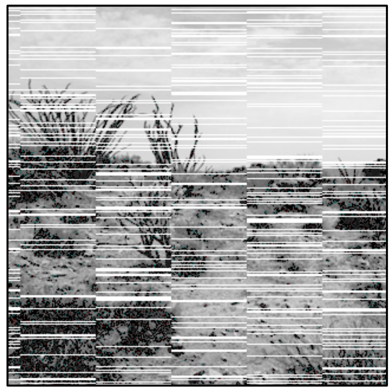


END-TO-END LOSS RATE IS EXPECTED TO BE HIGH!

Image quality? Uncompressed BMP

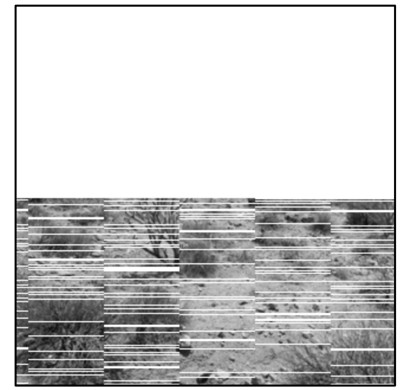
1617 packets, 64 bytes payload, one Hop
Loss rate: 20%, No loss bursts (radio), No duty-cycling

Original 320x320
256 gray levels,
BMP 102400 bytes



1340 out of 1617
packets received

1303 out of 1617
packets received



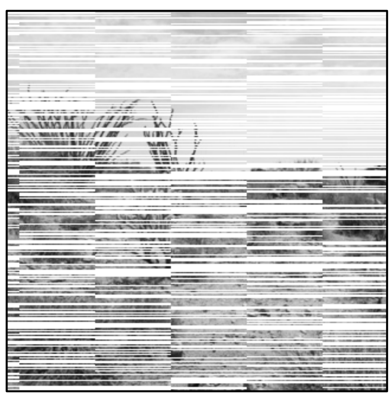
674 out of 1617
packets received

Max TX rate = 250 kps
(IEEE 802.15.4)

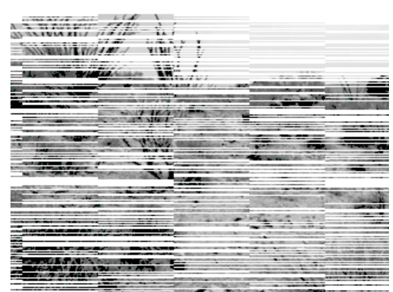
Minimum latency = 6.46s

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

Encoding cost of JPEG2000 is too high for these devices.

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

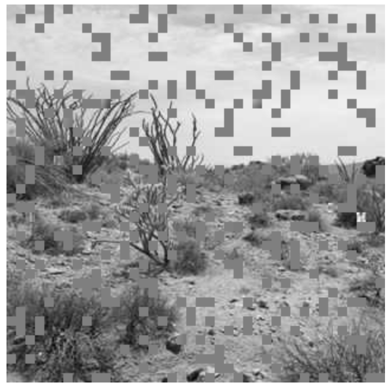
Improving image robustness

302 packets, 64 bytes payload, one Hop
Loss rate: 20%, No loss bursts (radio), No duty-cycling

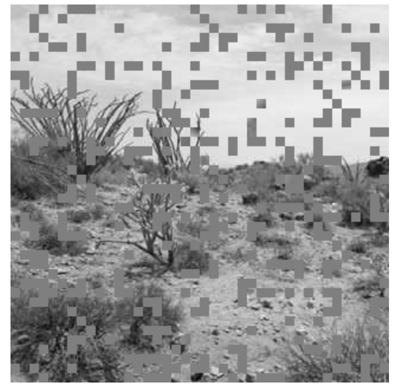
Original 320x320
256 gray levels,
WSN specific 17199 bytes

Max TX rate = 250 kps
(IEEE 802.15.4)

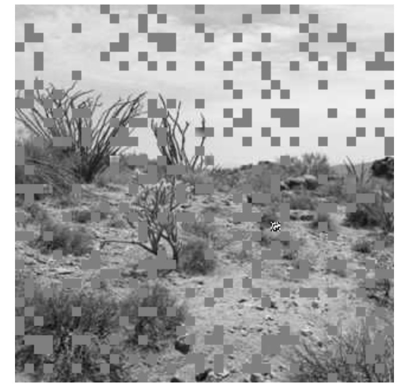
Minimum latency = 1.14s



248 out of 302
packets received

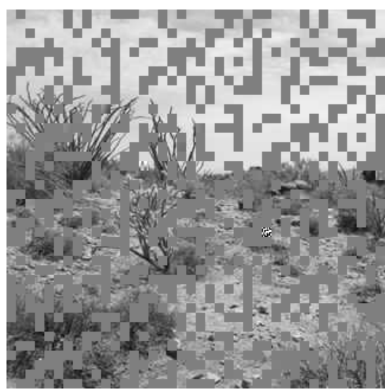


236 out of 302
packets received

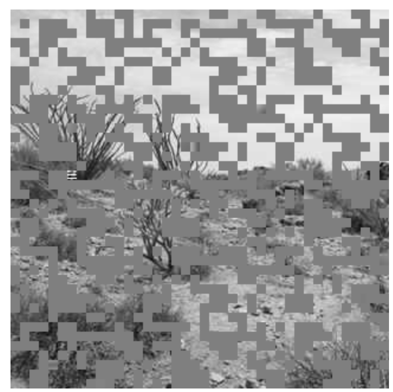


243 out of 302
packets received

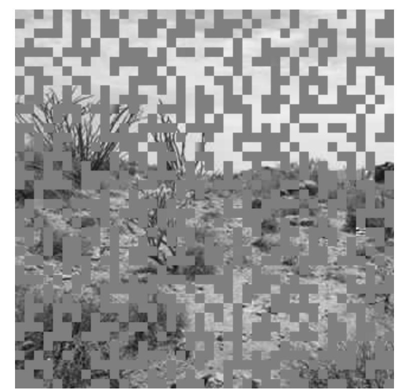
With loss bursts (radio)



188 out of 302
packets received



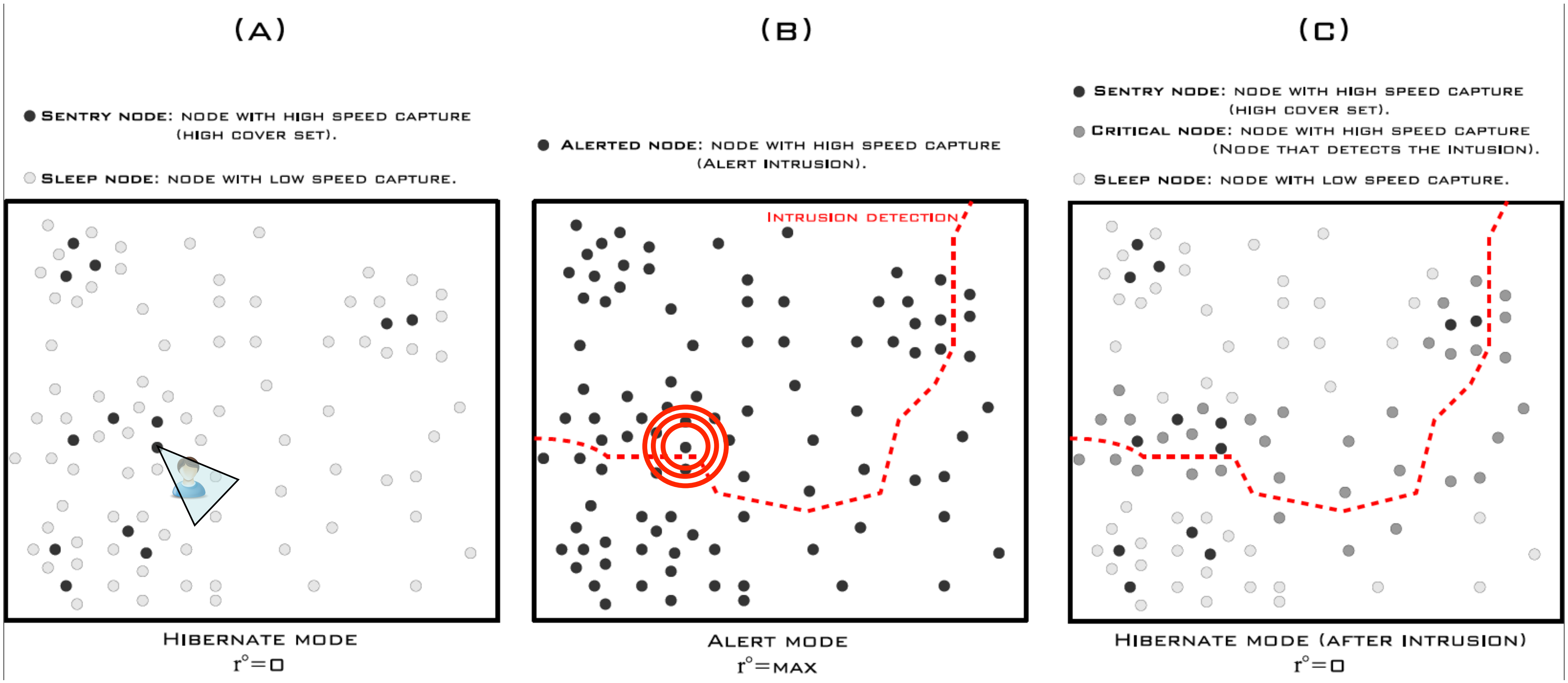
167 out of 302
packets received



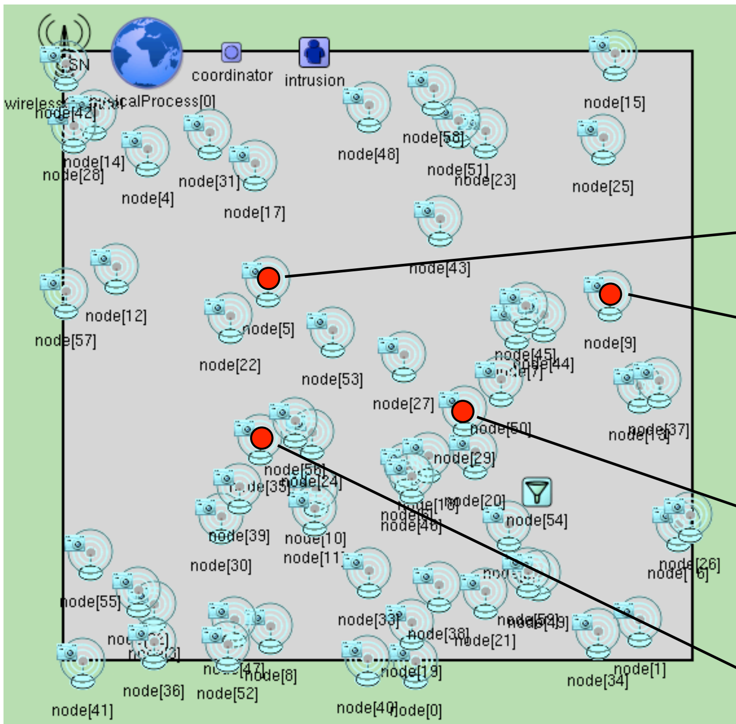
158 out of 302
packets received

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

Intrusion detection scenario



Some images displayed by the sink

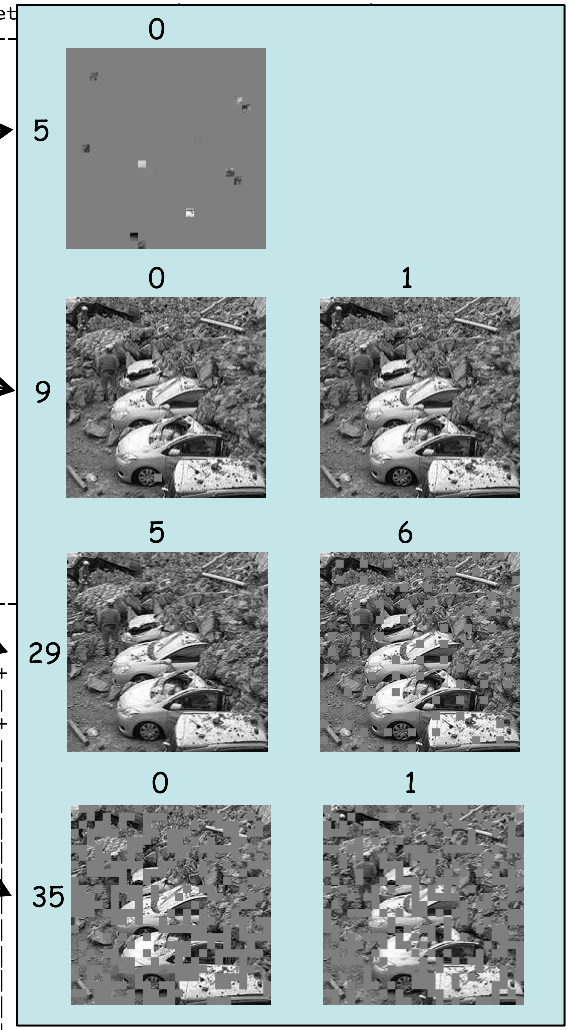


Application:Image sent

	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



60 image sensor nodes
75mx75m
1 sink (node 54)

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/node/application/videoSensorNode) - gedit

```
package node.application.videoSensorNode;

// The sensor node module. Connects to the wireless channel in order to communicate
// with other nodes. Connects to psysical 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 (8); // 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

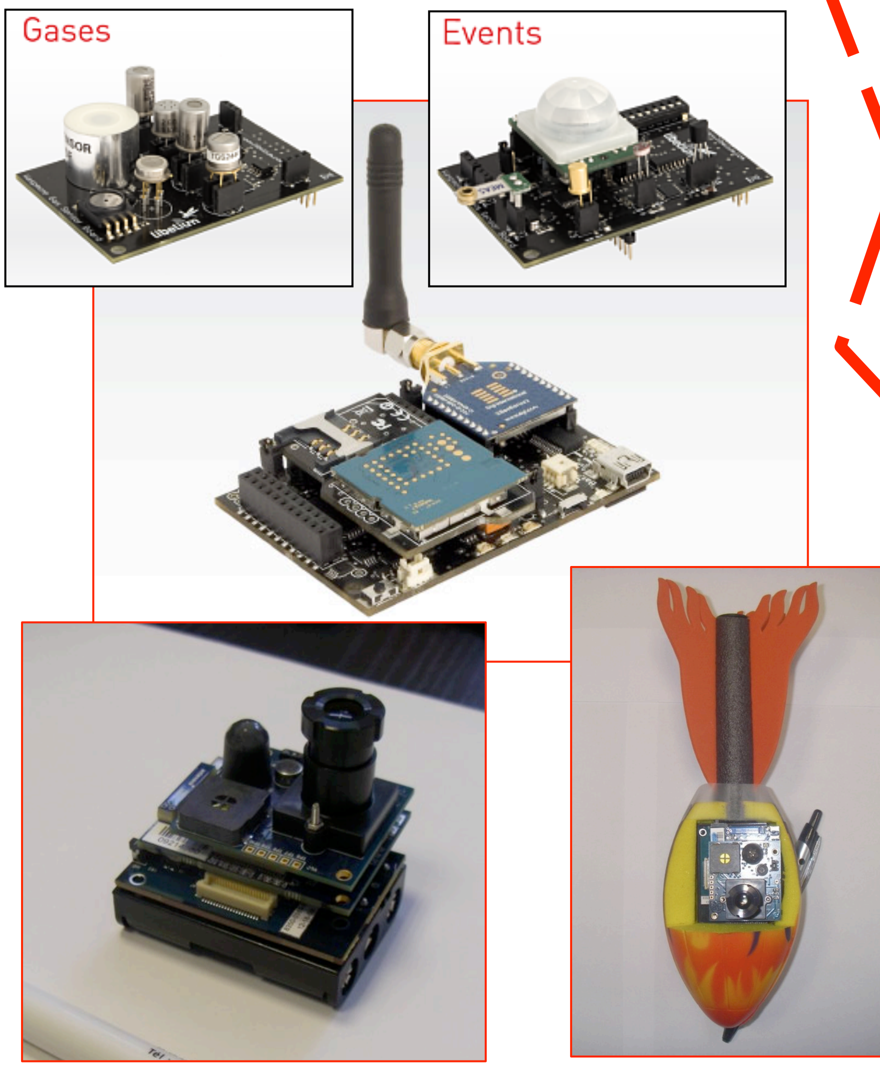
- Use the powerful *Castalia* script to launch specific simulation scenario
- Define configurations in omnetpp.ini file
  - `[Config Video200Config]`
  - `[Config CSMA], [Config SMAC], [Config TMAC]`
  - `[Config GPSRRoutingForcedImageTest60_300]`
  - `[Config BMP128x128ConsumptionModel]`
  - `[Config CRAN200x200ConsumptionModel]`
- Select configuration combination
  - `Castalia -c CSMA,GPSRRoutingForcedImageTest60_300,Video200Config,CRAN200x200ConsumptionModel`

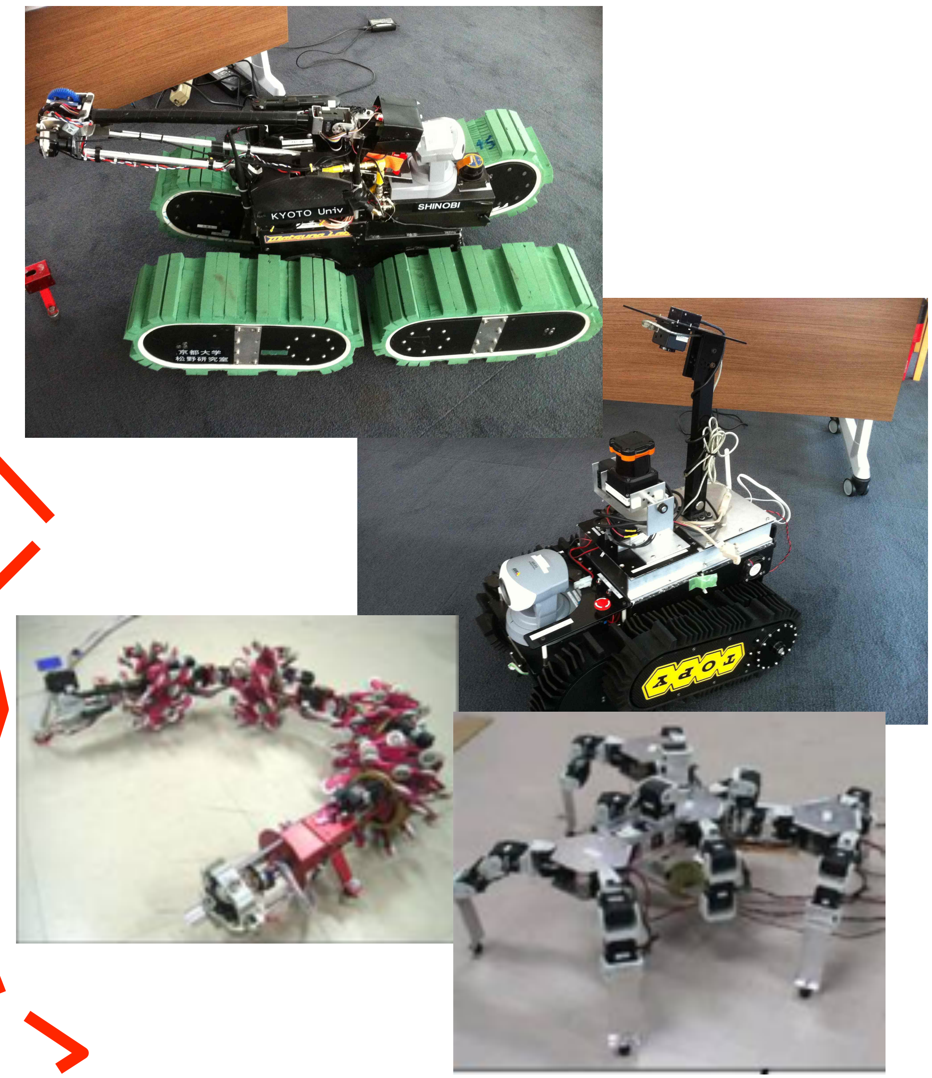
Enabling large-scale, operational search & rescue applications

Sensor & Robots

- Wireless Sensor Networks
  - Large scale sensing
  - Natural collaboration though 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

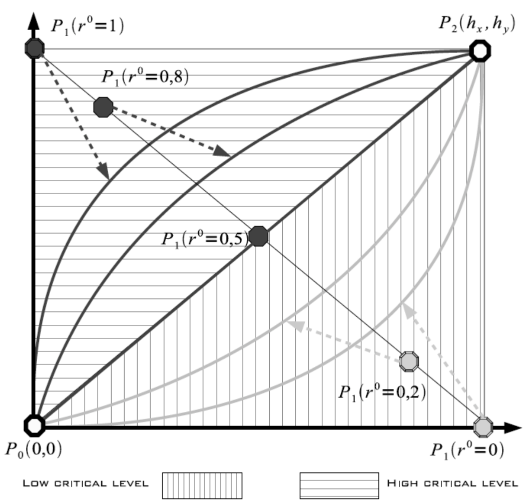
CHALLENGING COOPERATION IMPLIES DIFFERENCES!





Sensors & Robots Propose new interaction schemes

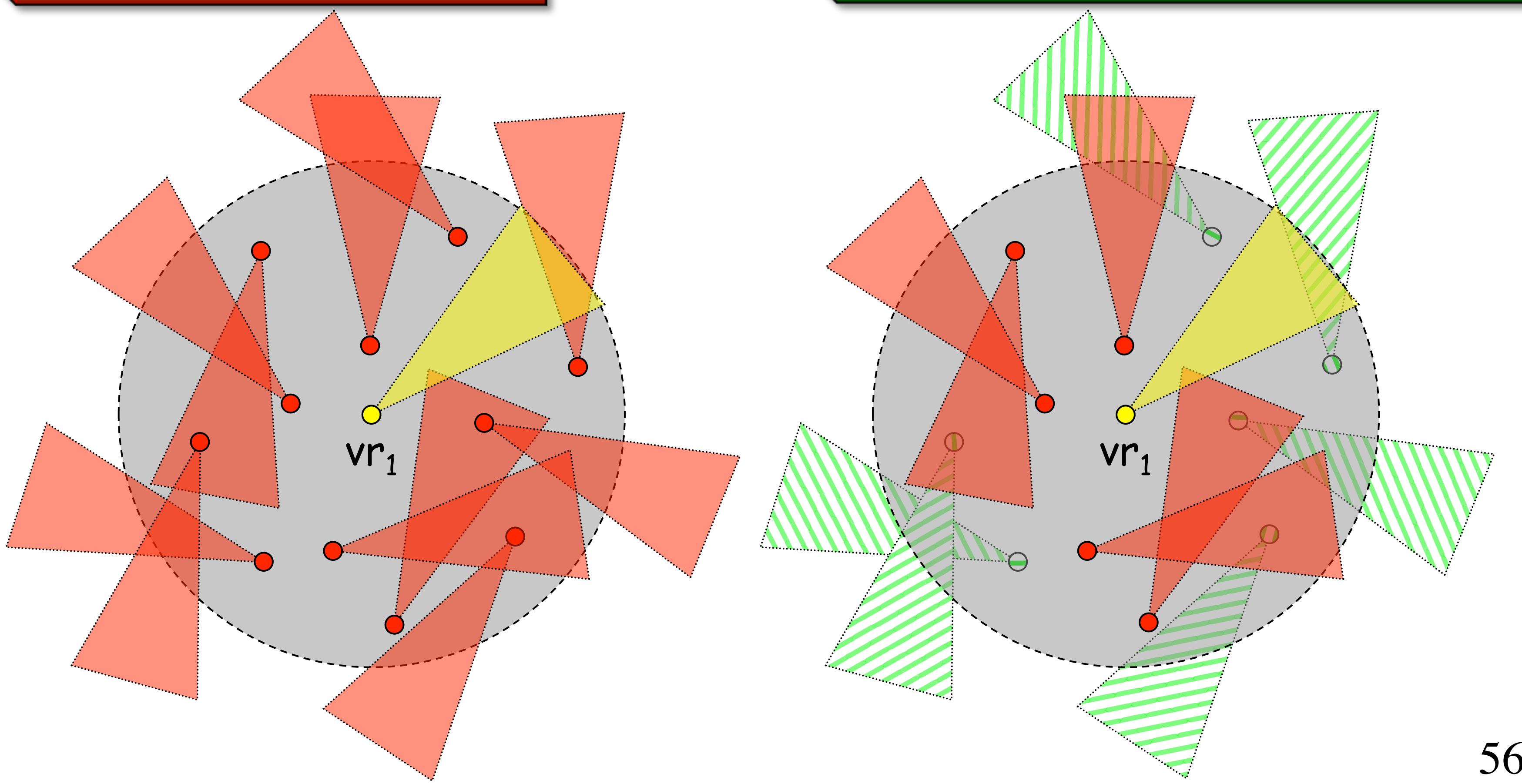
- Use the criticality model to control both sensors and robots
- Prototyping on real hardware



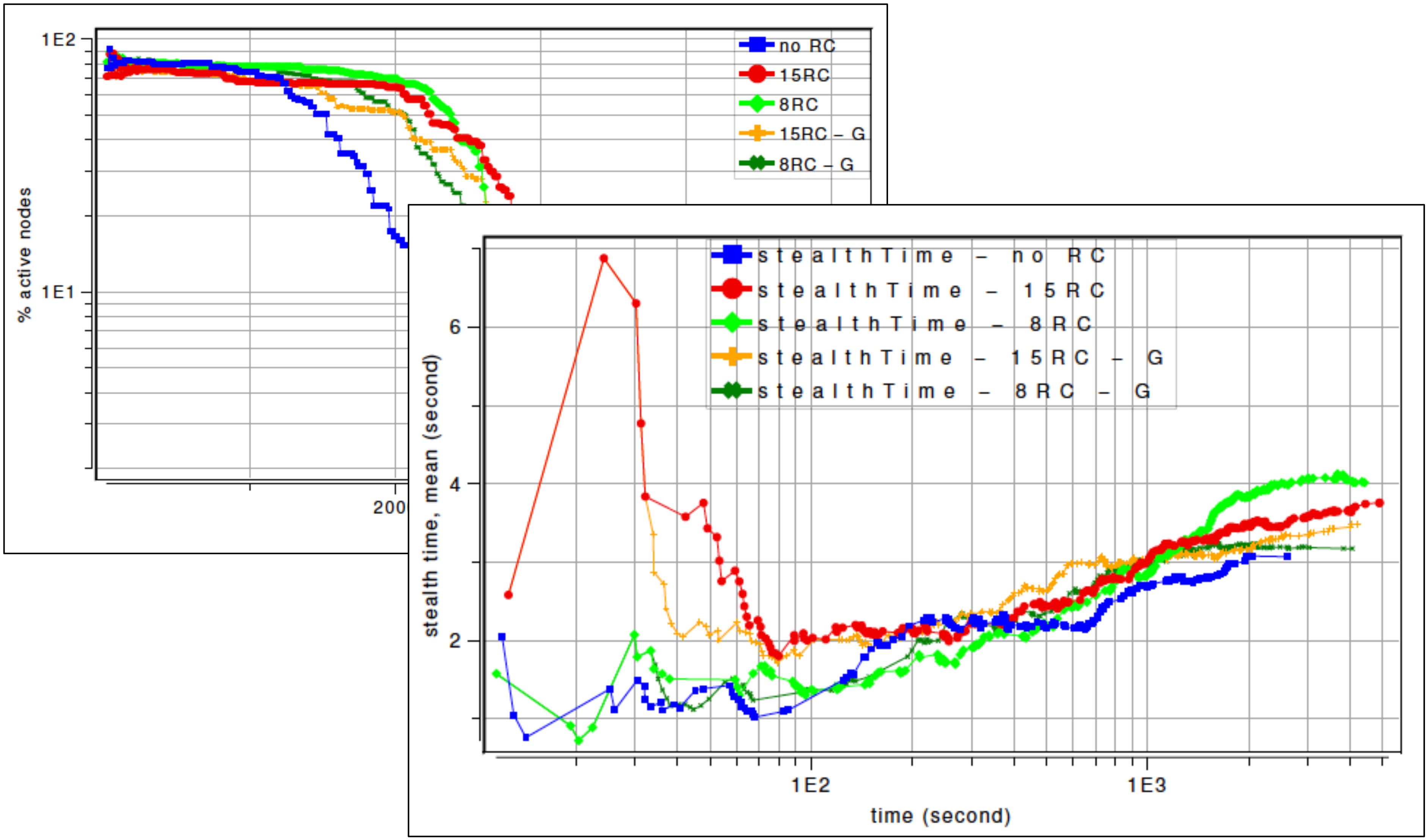
Cooperation with cameras on mobile robots

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

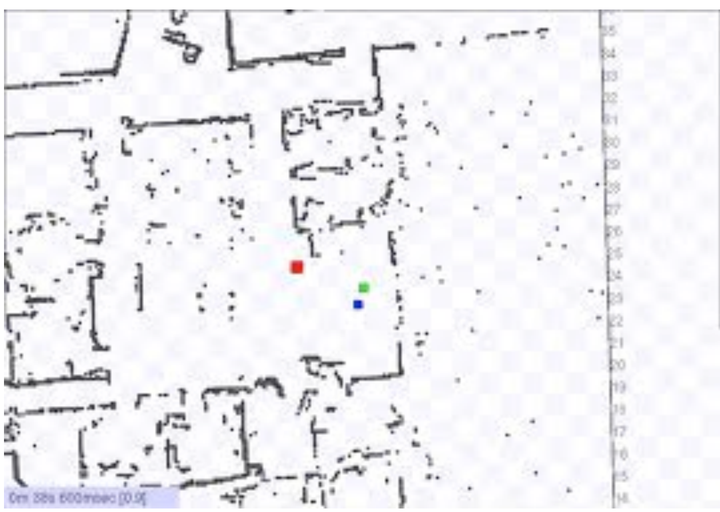


Impact on lifetime & stealth time



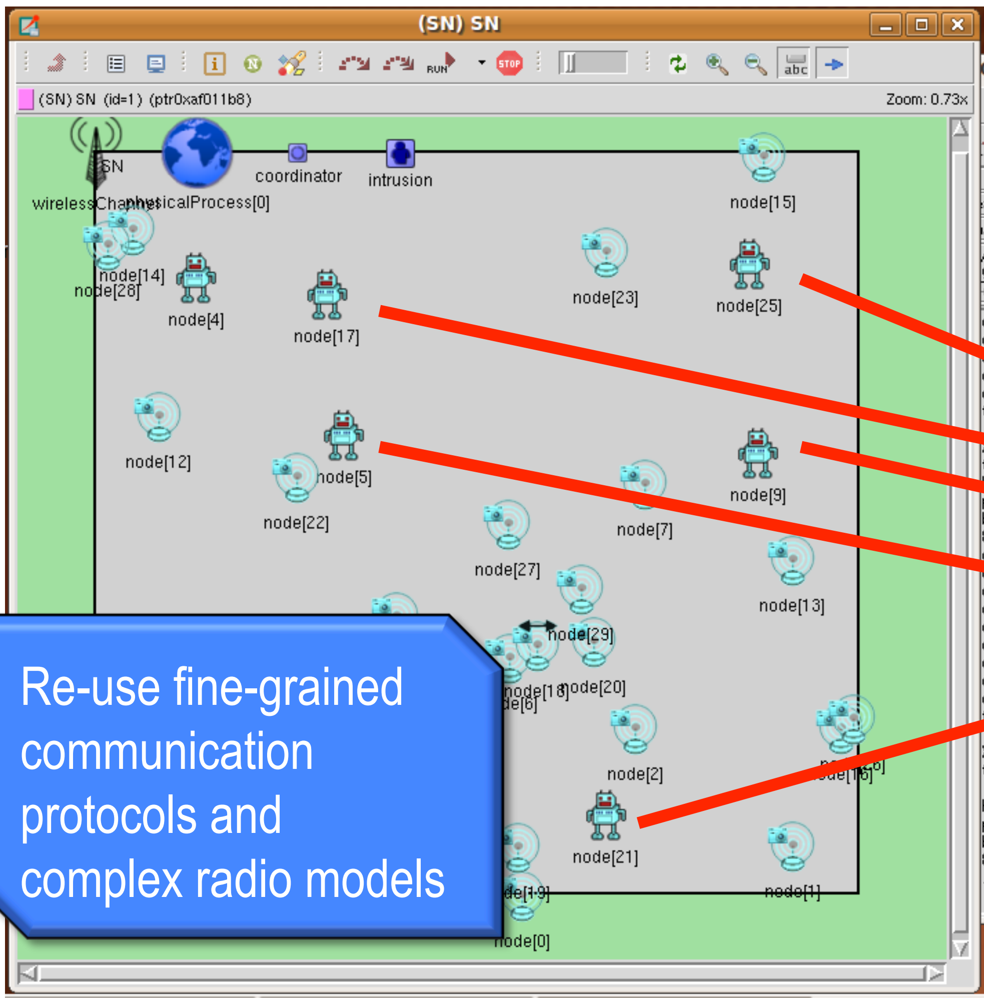
Robot simulators

- Mobility, exploration, navigation, tracking, control and design are very important for robots
- Use specific robot simulators

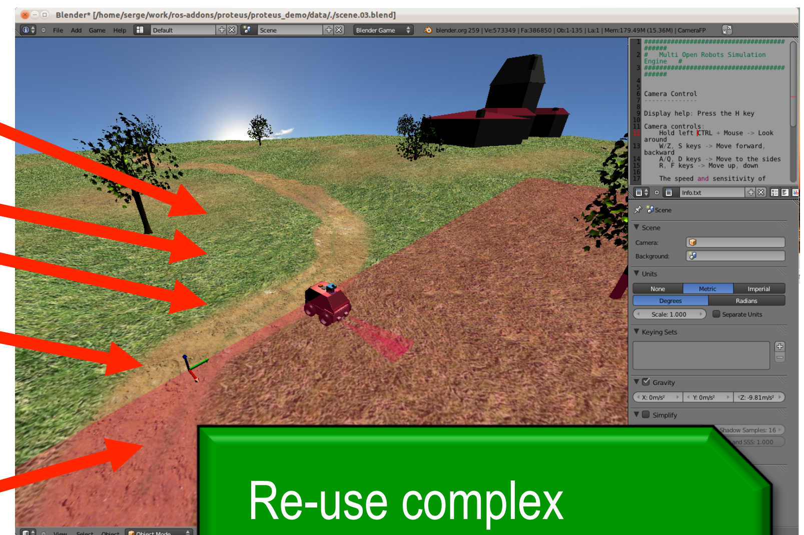


[Sensors & Robots enable realistic interaction studies]

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

Publicly available model

Archive
http://web.univ-pau.fr/~cpham/WSN-MODEL/distrib/wvsnmodel-v4.tgz

Documentation
http://web.univ-pau.fr/~cpham/WSN-MODEL/wvsn-castalia.html

Restricted access

Archive
http://web.univ-pau.fr/~cpham/WSN-MODEL/distrib-private/

Documentation
http://web.univ-pau.fr/~cpham/WSN-MODEL/wvsn-castalia-private.html