# Deployment of mission-critical surveillance applications on wireless sensor networks

Université de Franche-Comté
IUT de Belfort
LIFC
Friday, March 12th, 2010

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

# Déploiement d'applications critiques de surveillance sur réseaux de capteurs sans-fils

Université de Franche-Comté
IUT de Belfort
LIFC
Vendredi 12 mars 2010

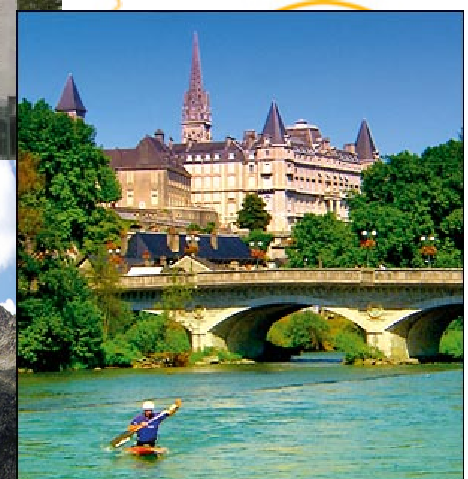
Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France

# SITY OF PAU
## ES

Diaporama des
Campus de l'UPPA

THE 3 GE
SITES OF

ayonn
nglet




# LIUPPA

**COMPUTER SCIENCE LAB**

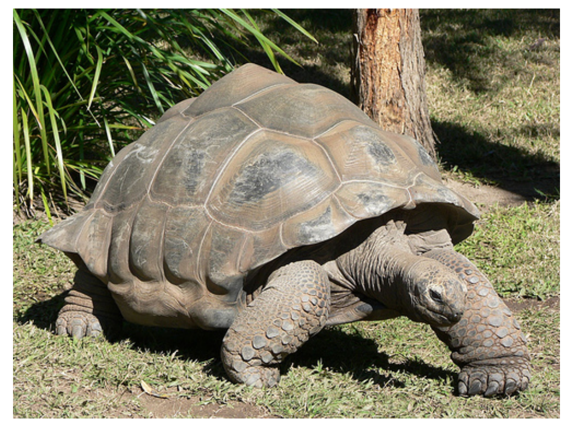
- **32 FACULTY MEMBERS**
- **25 PHD STUDENTS**

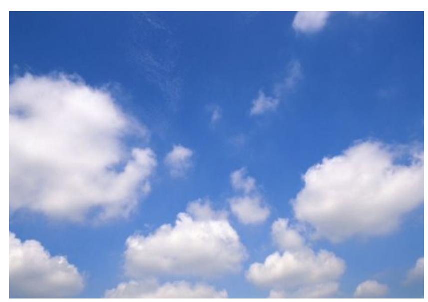
**2 RESEARCH TEAMS**

- **MODELING, VISUALIZATION, EXECUTION & SIMULATION**
- **INFORMATION PROCESSING, INTERACTIONS AND ADAPTATION**

# Deployment of mission-critical surveillance applications on wireless sensor networks






# Deployment of mission-critical surveillance applications on wireless sensor networks





# Surveillance applications (1)

- Lesson 1: don't miss important events

What is captured

Whole understanding of the scene is wrong!!!

# Surveillance applications (2)

- Lesson 2: high-quality not necessarily good

333x358 16M colors, no light

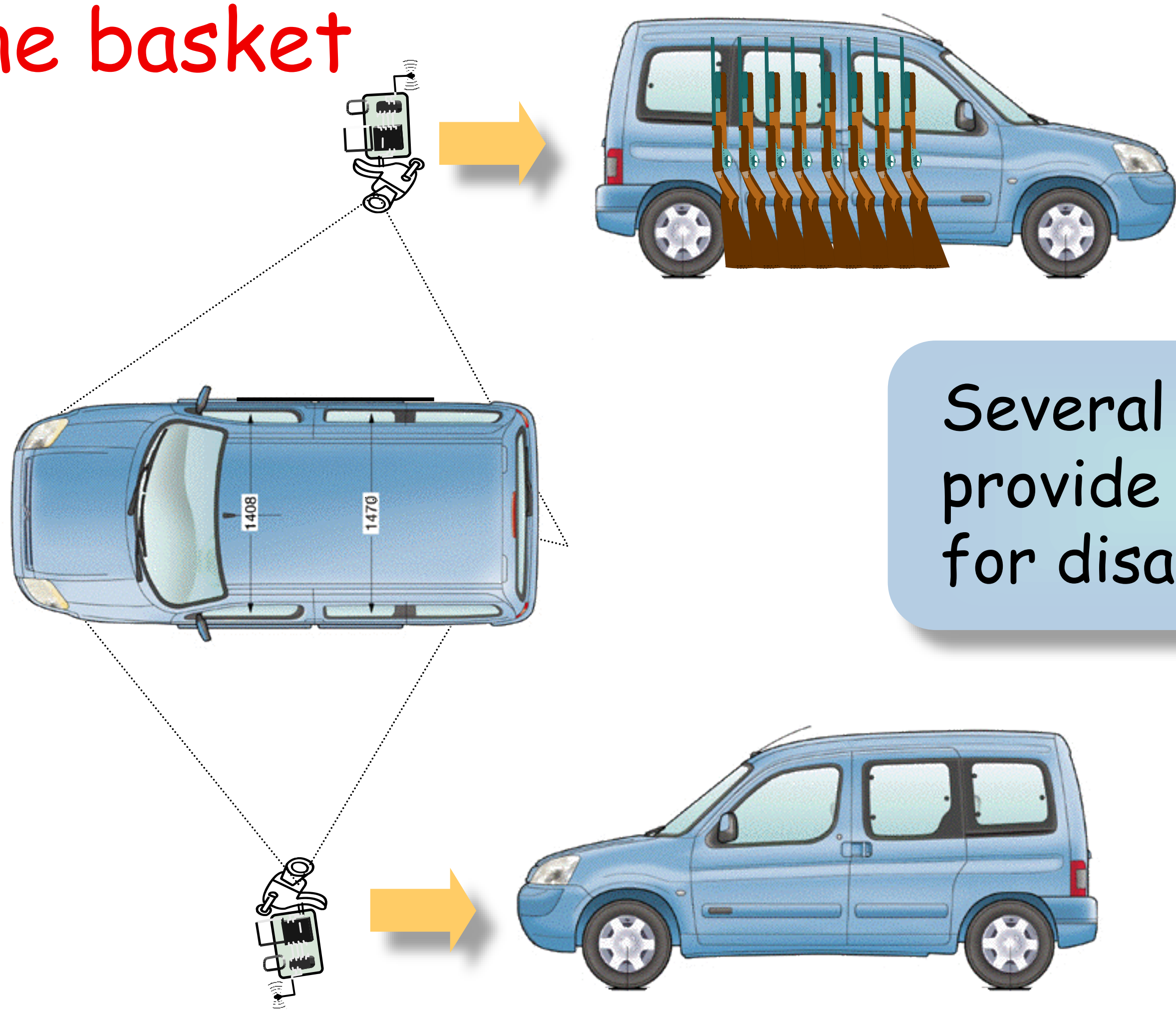
167x180 16 colors, light

167x180 BW (2 colors), light

Keep in mind the goal of the application!

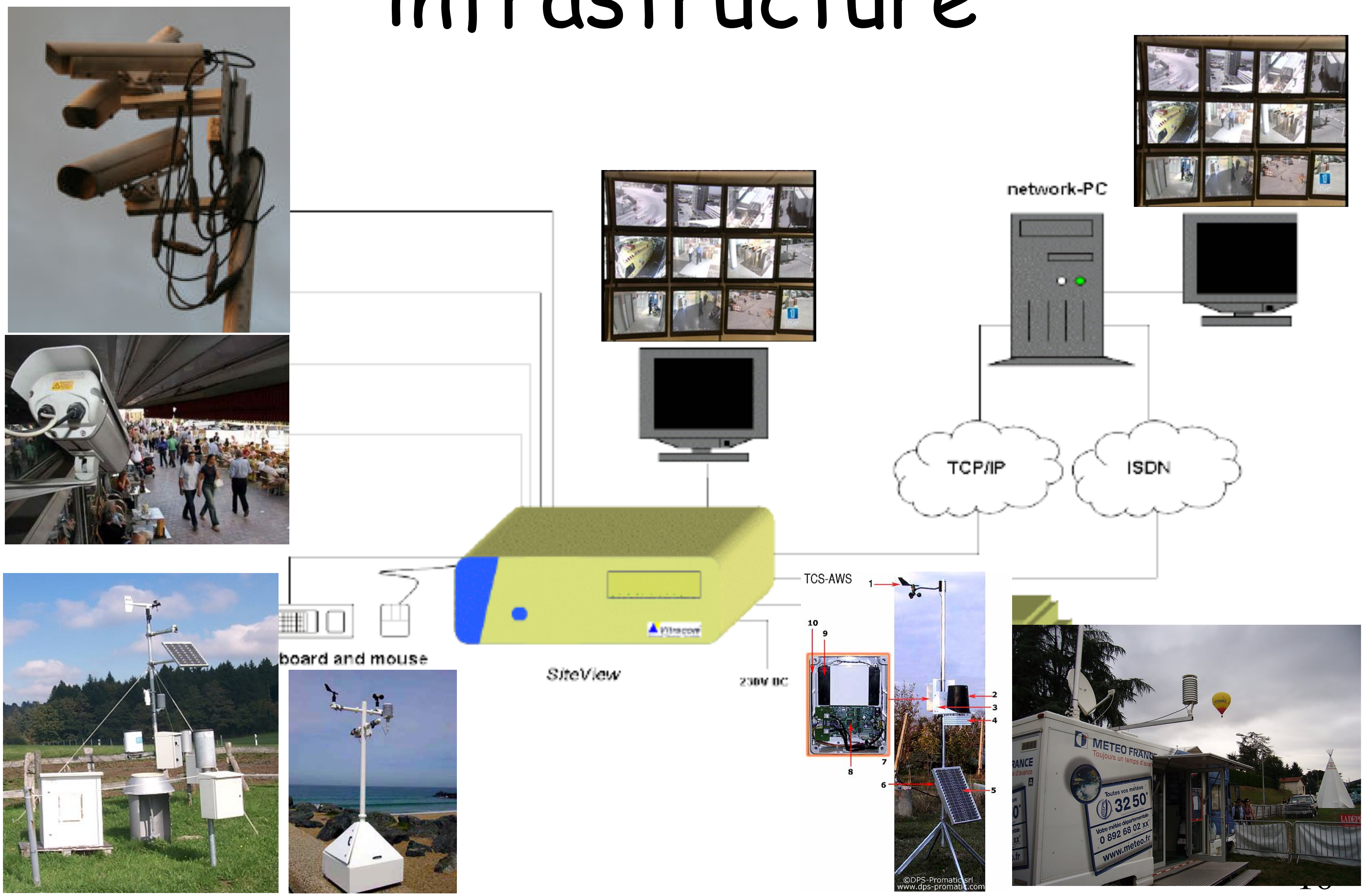
# Surveillance applications (3)

- Lesson 3: don't put all your eggs in one basket



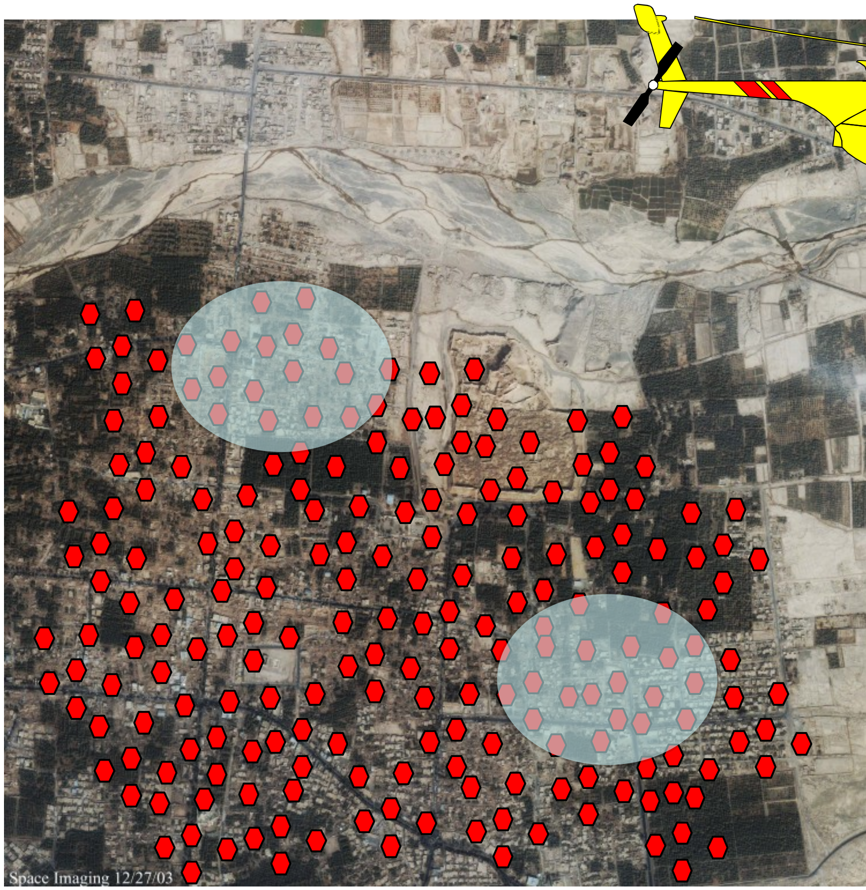

Several sources provide multi-view for disambiguation

# Traditionnal surveillance infrastructure



# Small, Automous Sensors

## disaster relief - security

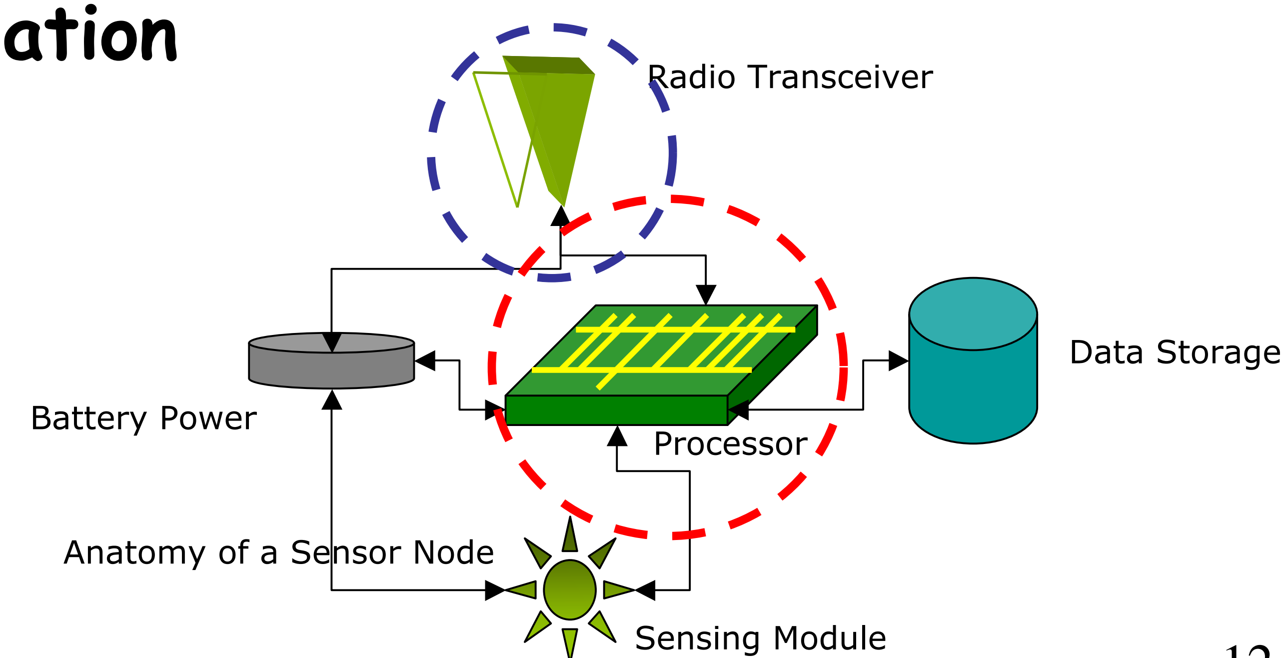



Organization of rescue in large scale disasters relief operations

Rapid deployment of fire detection systems in high-risk places

# Wireless autonomous sensor

- In general: low cost, low power (the battery may not be replaceable), small size, prone to failure, possibly disposable
- Role: sensing, **data processing**, **communication**



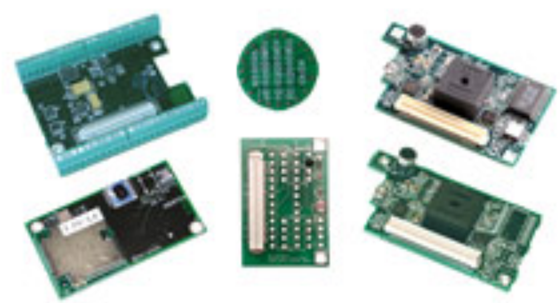
# Berkeley Motes (contd.)

- Each Mote has two separate boards
  - A main CPU board with radio communication circuitry
  - A secondary board with sensing circuitry
- Decouples sensing hardware from communication hardware
- Allows for customization since application specific sensor hardware can be plugged-on to the main board

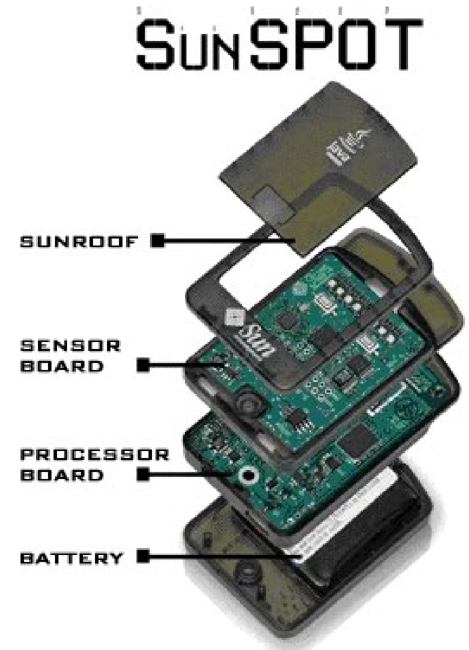

MICA2


Imote2

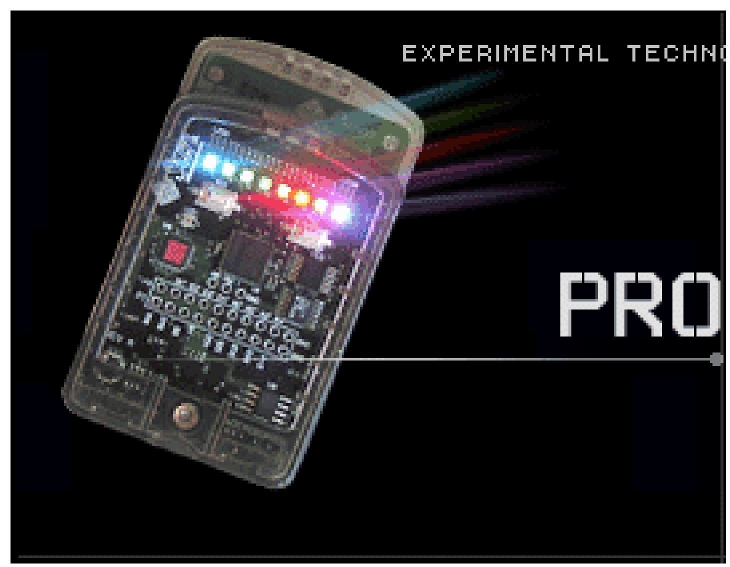

MICAz


Sensing boards

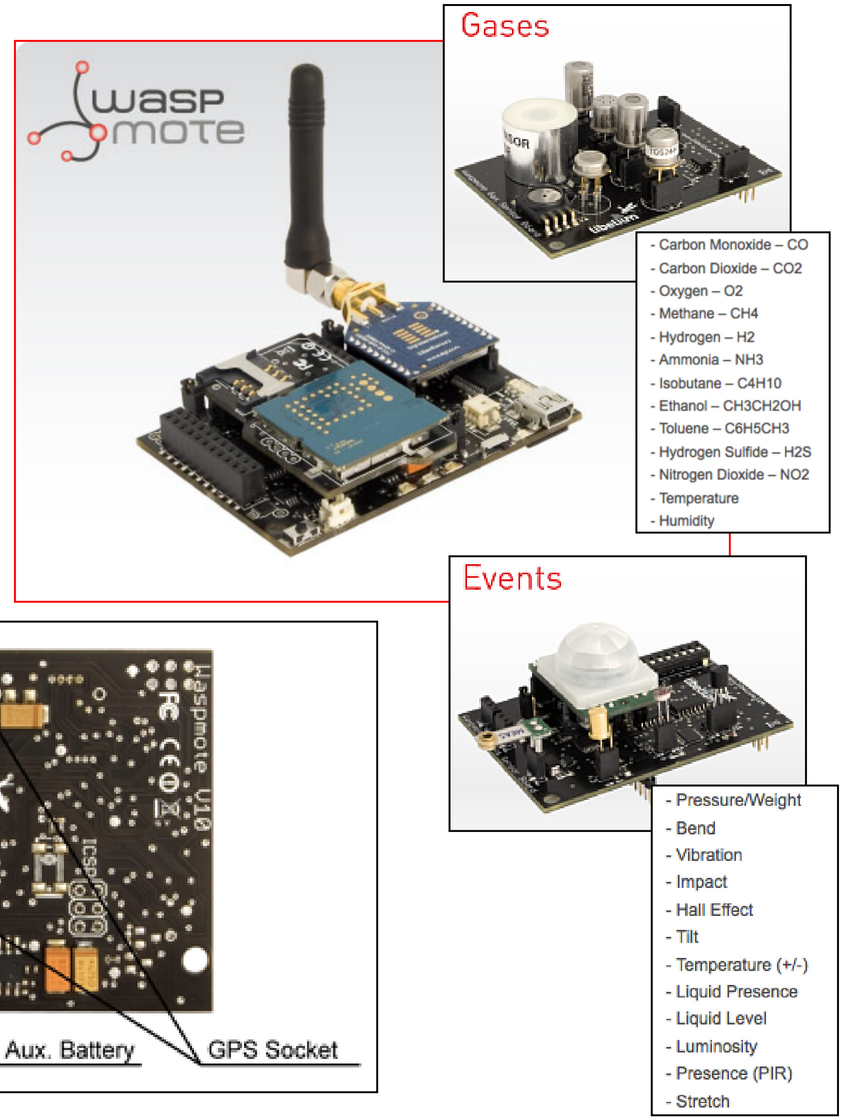
# SUN SPOT

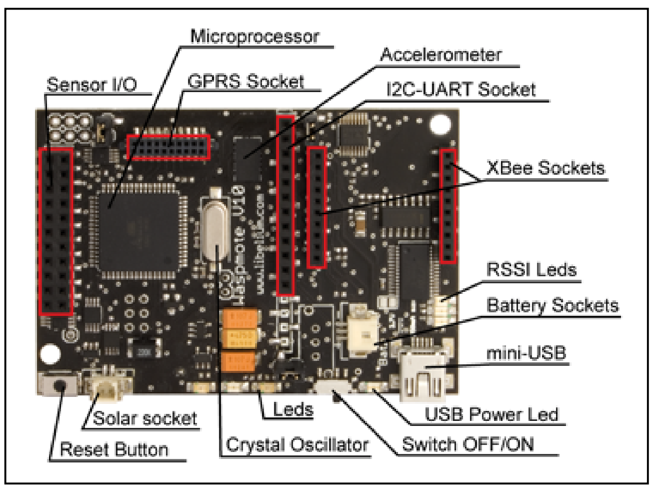


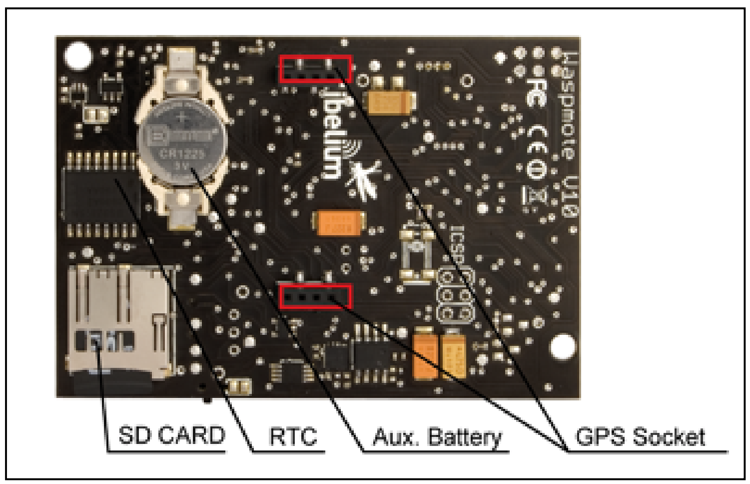

- Processor : ARM920T 180MHz 32-bit
- 512K RAM & 4M Flash.
- Communication : 2.4GHz, radio chipset: TI CC2420 (ChipCon) – IEEE 802.15.4 compatible
- Java Virtual Machine (Squawk)
- LIUPPA is official partner



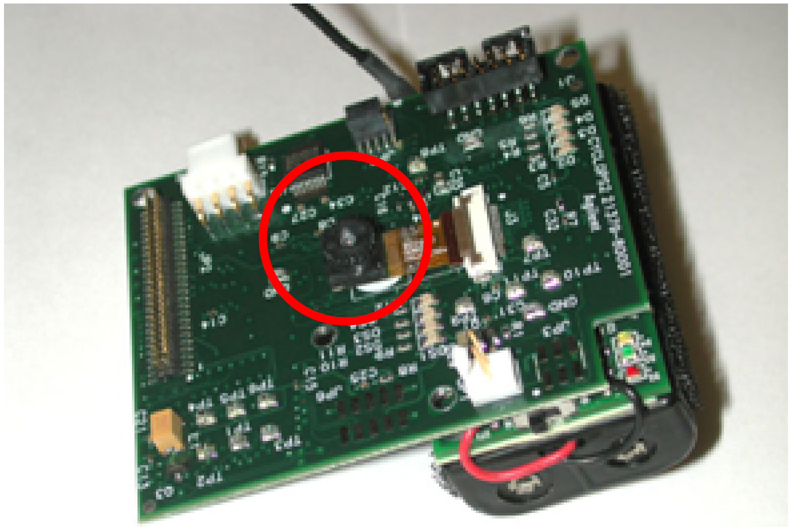
- ATmega1281 microcontroller
- 8K RAM & 1G SD card.
- 2.4GHz IEEE 802.15.4 compatible. RF and GSM/GPRS



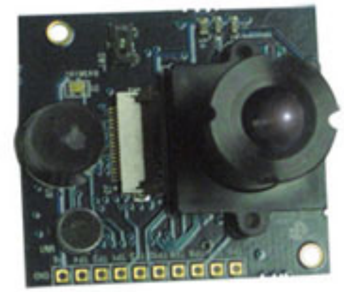



# Wireless Video Sensors

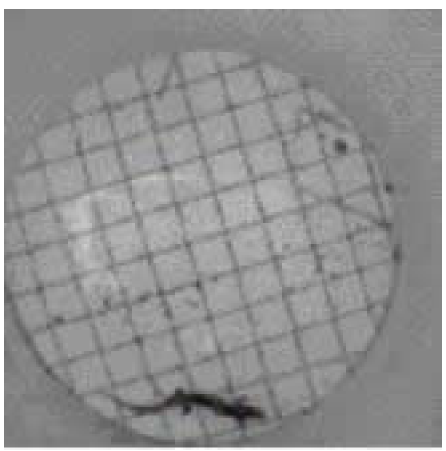


Cyclops video board on Mica motes

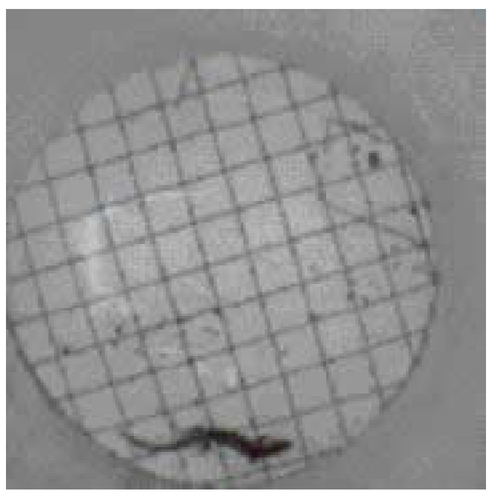
Imote2



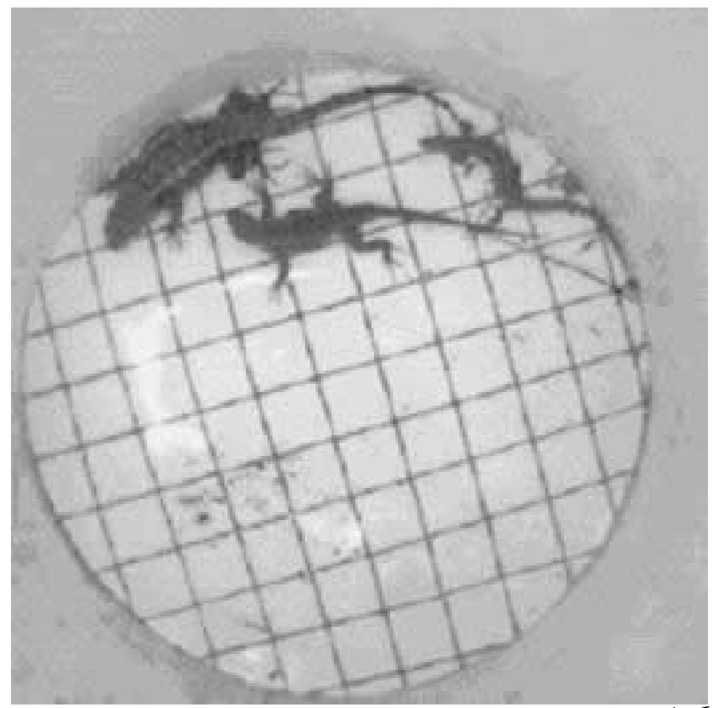
Multimedia board



128x128



140x140



240x240

# Sensor as common object
## toward very large scale deployment

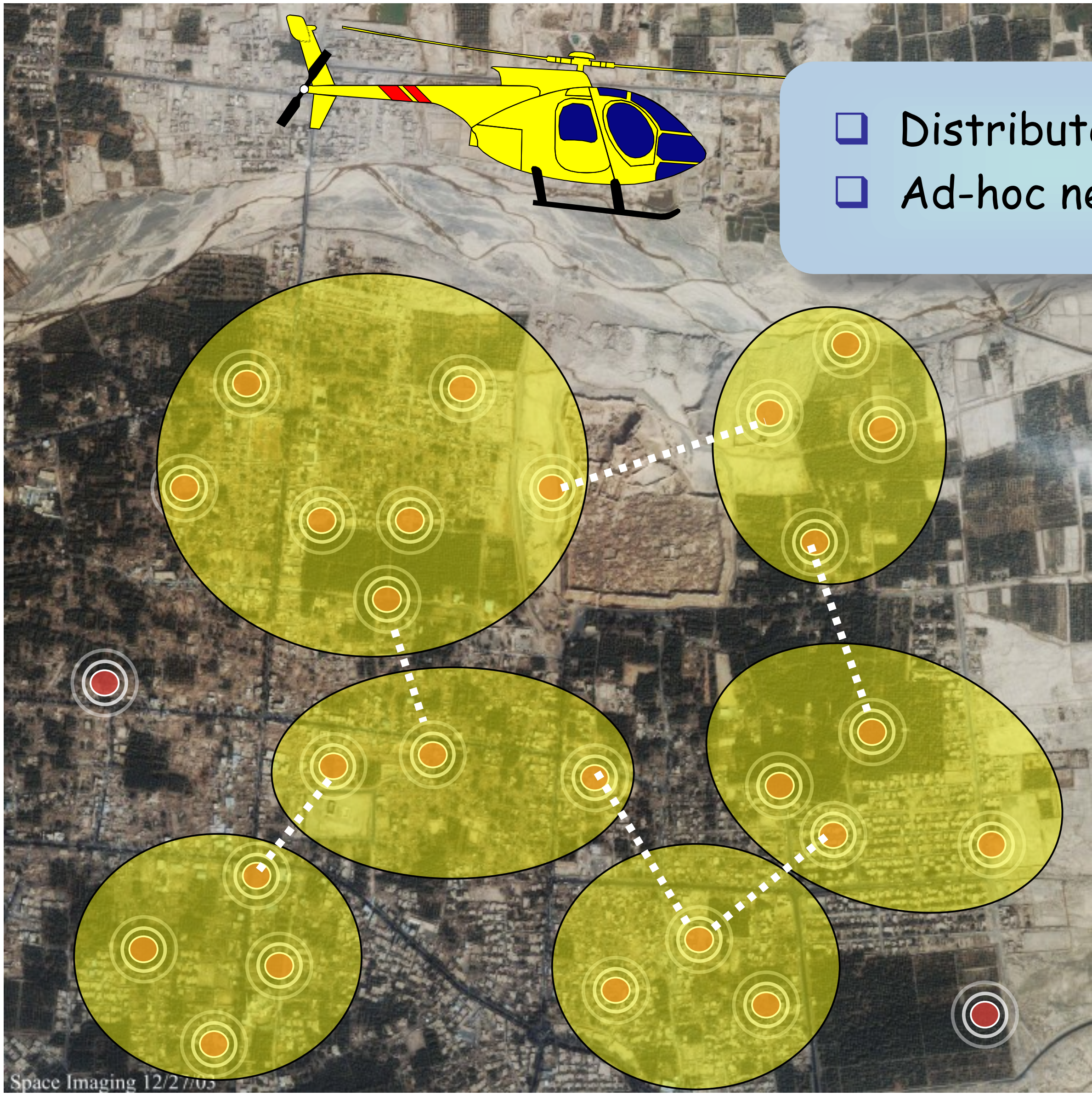
Environmental monitoring
- air
- water

Cell-phones with embedded CO sensor

- most ubiquitous device (millions)
- not deployment cost
- high replacement rate
- no energy constraints

# Wireless Sensor Network (1)

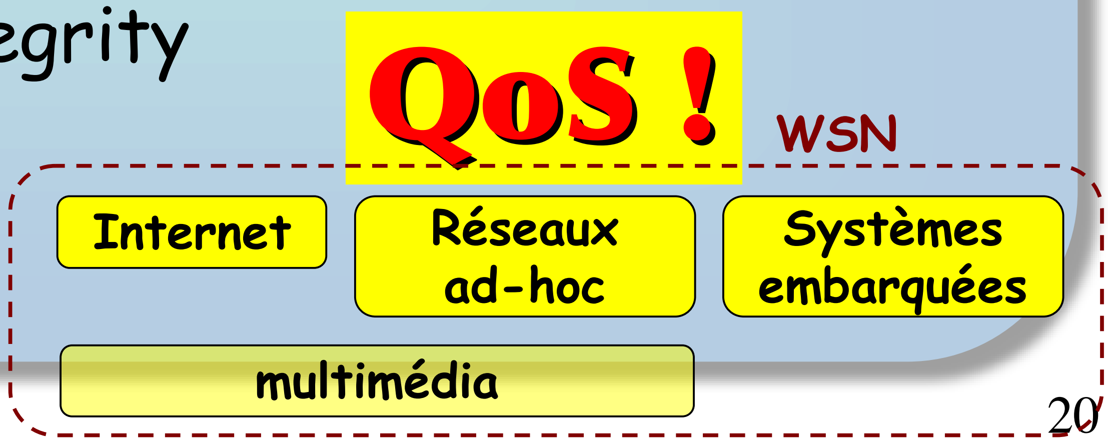
# Wireless Sensor Network (2)




- Distributed system
- Ad-hoc network

# Mission-critical surveillance applications

- Availability: 24/24 surveillance
  - Hardware : failures, energy depletion
  - Data
- Quality
  - Enhance/validate/disambiguate information with several sources of information
  - Adaptation to local conditions
- Reliability/integrity
  - Hardware
  - Network
  - Data



# Surveillance as a Service

ACCOUNTABILITY

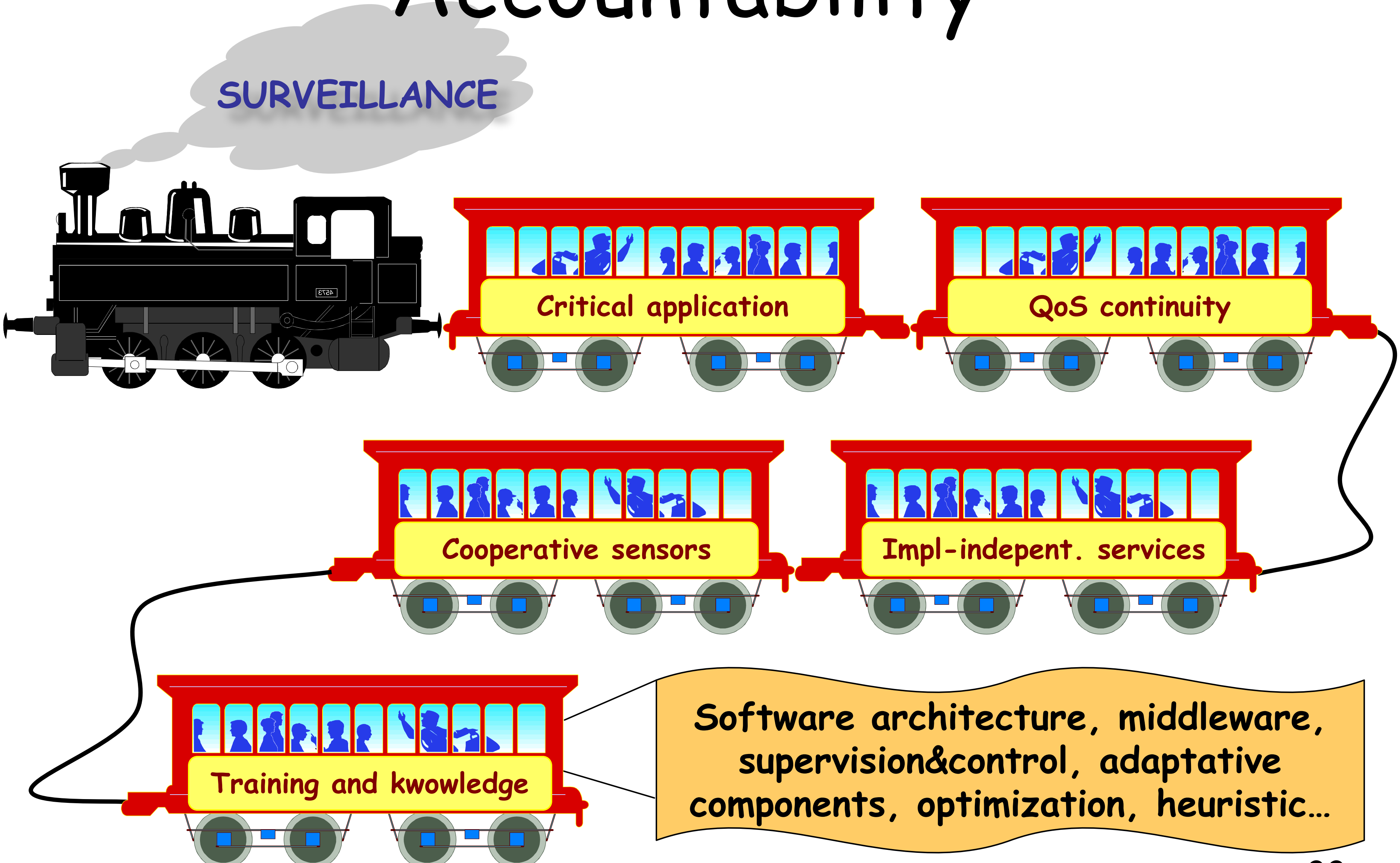
- Similar to Service Level Agreement

➔ SURVEILLANCE AT ANY PRICE ⇐

no discontinuity of service
against node's failures

data availability and reliability

# Surveillance as a service: Accountability

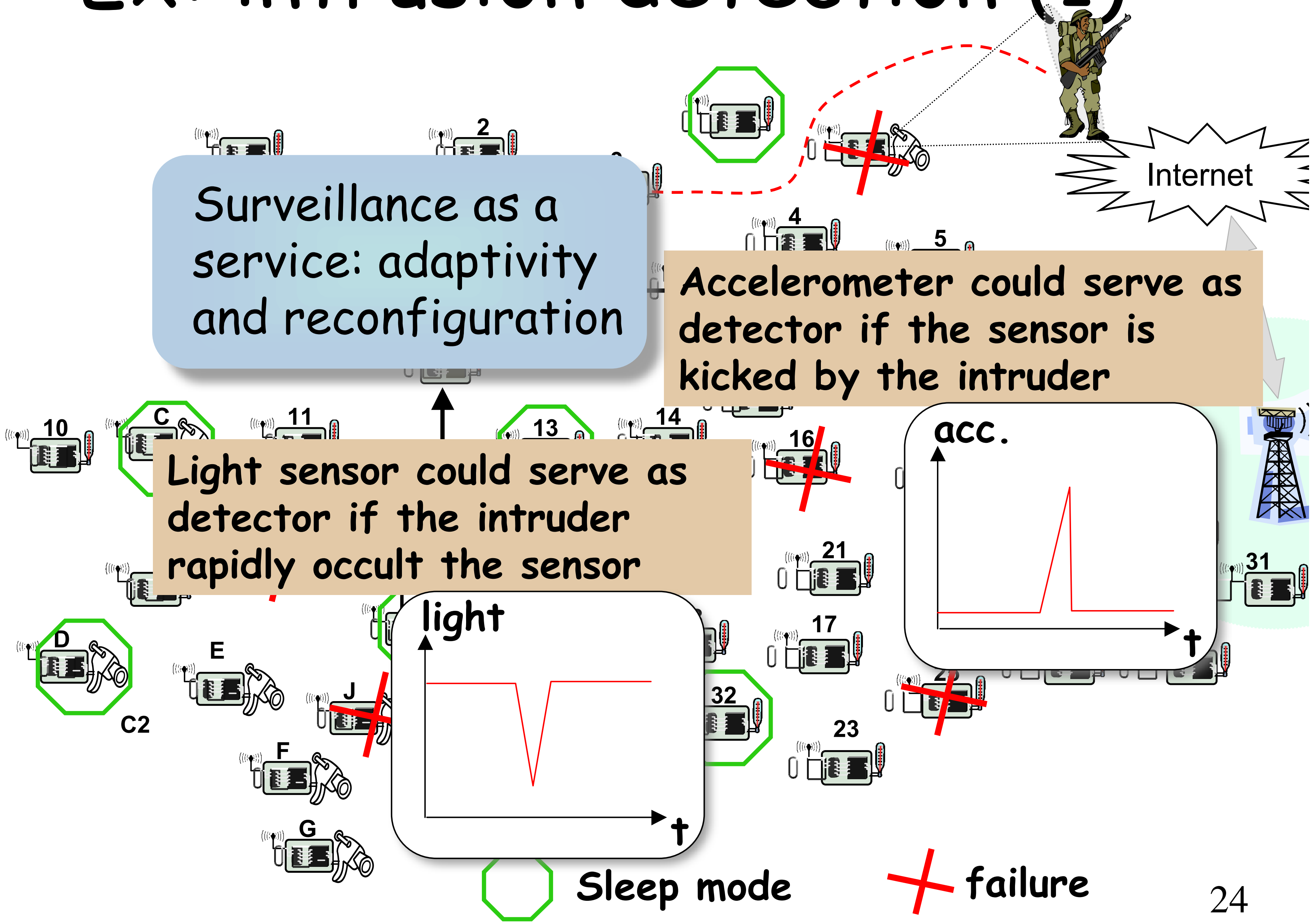


# Ex: intrusion detection (1)

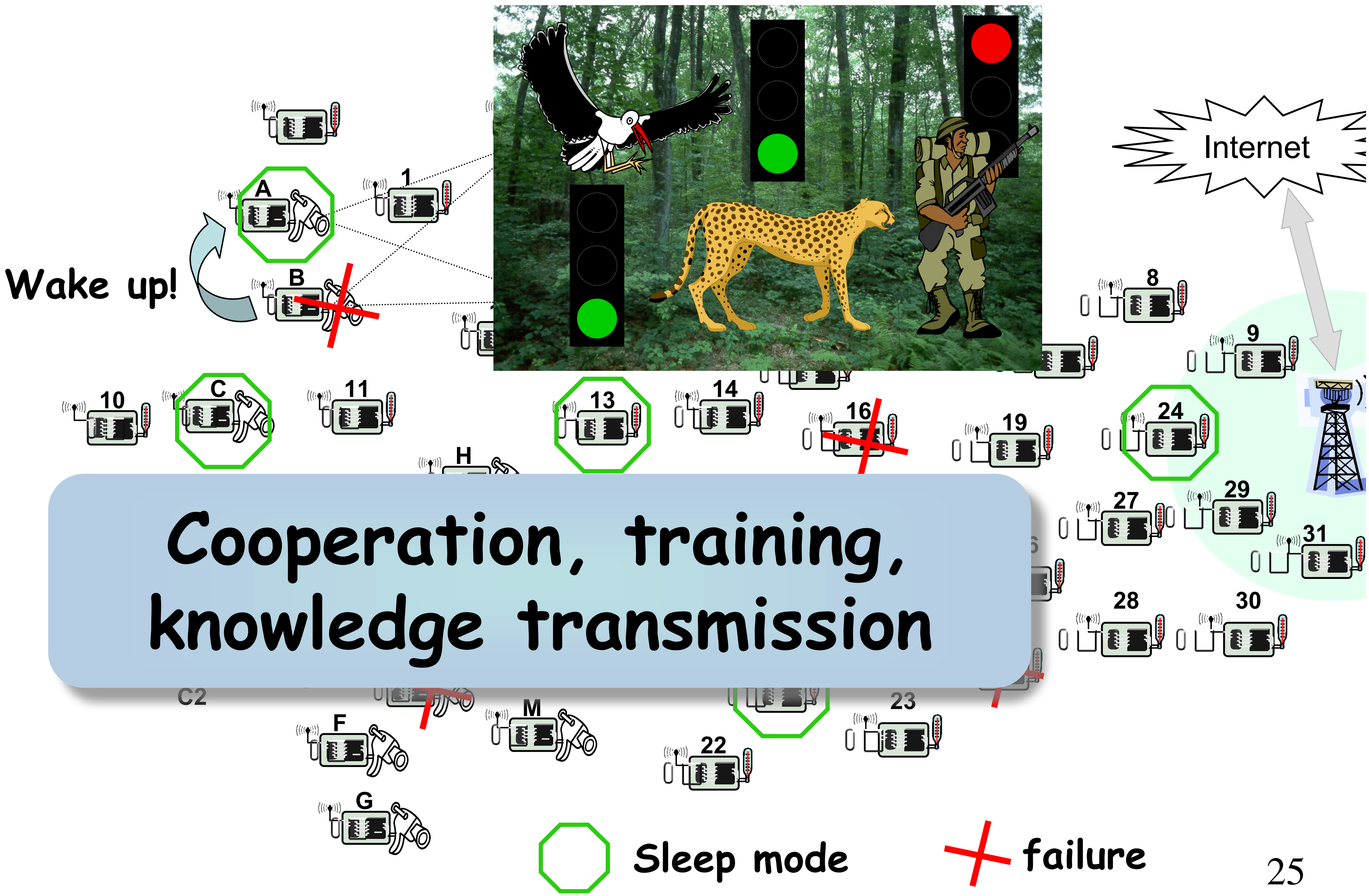
Surveillance as a service: adaptivity and reconfiguration

Accelerometer could serve as detector if the sensor is kicked by the intruder

Light sensor could serve as detector if the intruder rapidly occult the sensor



# Ex: intrusion detection (2)




**Cooperation, training, knowledge transmission**

# Towards smart sensor grids

- The ultimate goal is to define a customizable sensor grid that could be dynamically programmed according to the application's objectives and needs
- Similar to the so-called active networking concept for the Internet…
- …but much easier to achieve!!

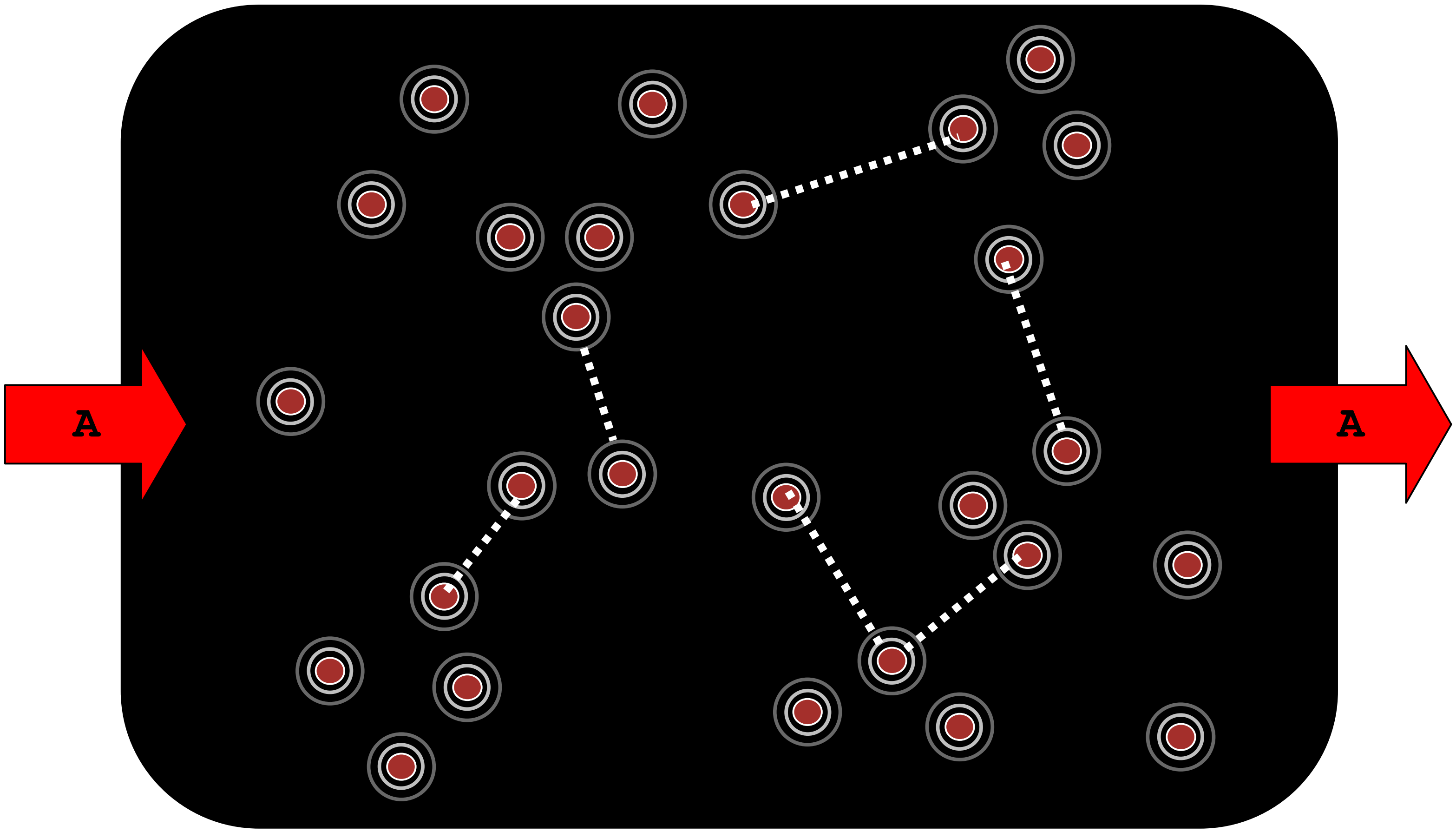
# The Internet we wanted, the one we have...

| | |
|---|---|
| End-to-end Quality of Service | No Quality of Service |
| Distributed processing | Very centralized architecture |
| Internet-wide, uniform control and policies | Heterogeneous and domain/ISP specific policies |
| Fast integration of new needs, new applications, new technologies | Upgrades and incremental deployments are slow |

# Net Neutrality or Not?

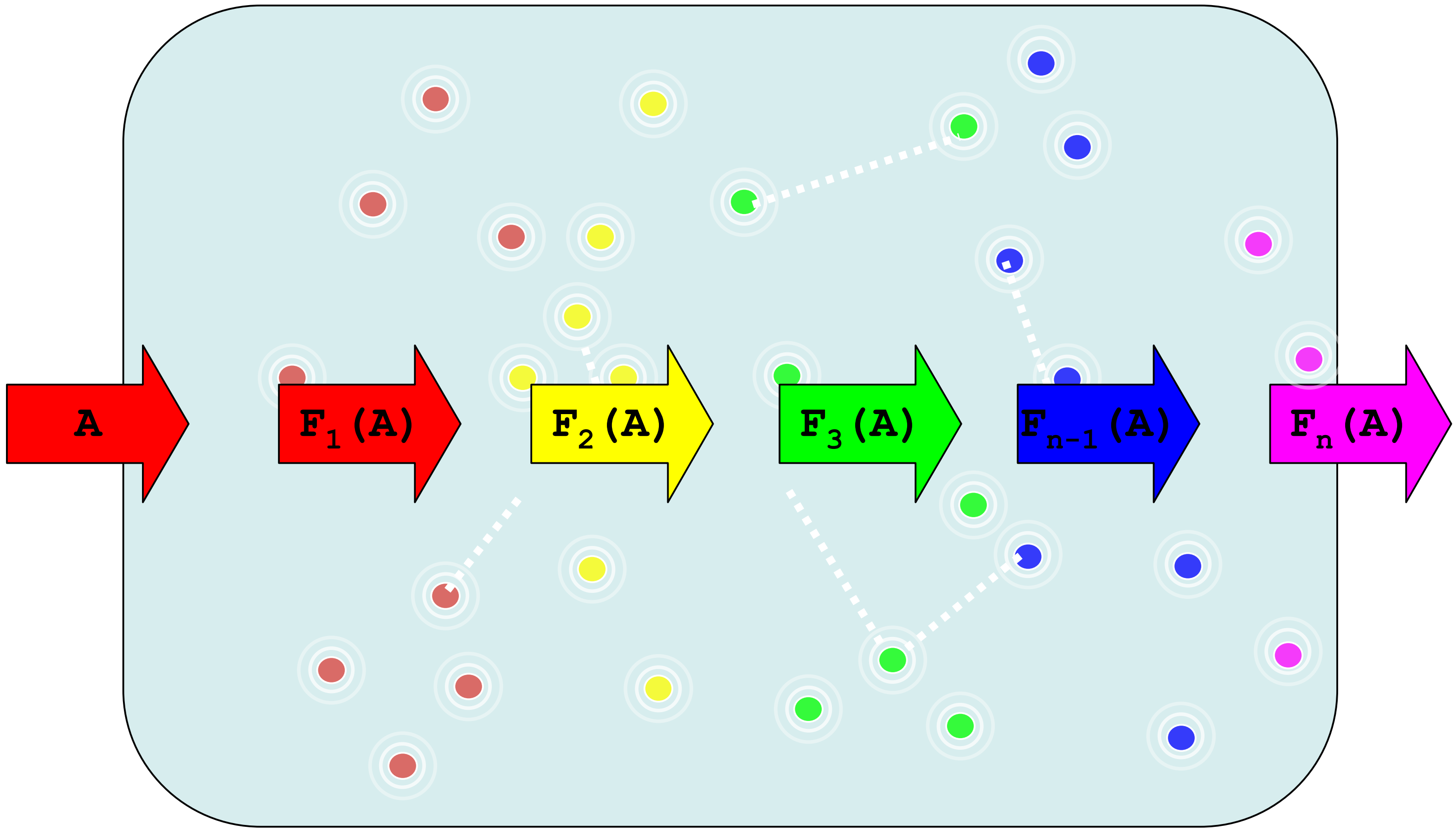
- The Internet's success is in a large part debtful to what's called Net Neutrality (IP neutrality)
- But, Net Neutrality is the main brake for achieving large scale QoS : IP routers only forward packets!
- Some services can be best supported or enhanced using information that is only available inside the network!
- Fortunately, in a sensor network, each node has de-facto specific processing capabilities

# One vision for enabling QoS in Sensor Nets (1)



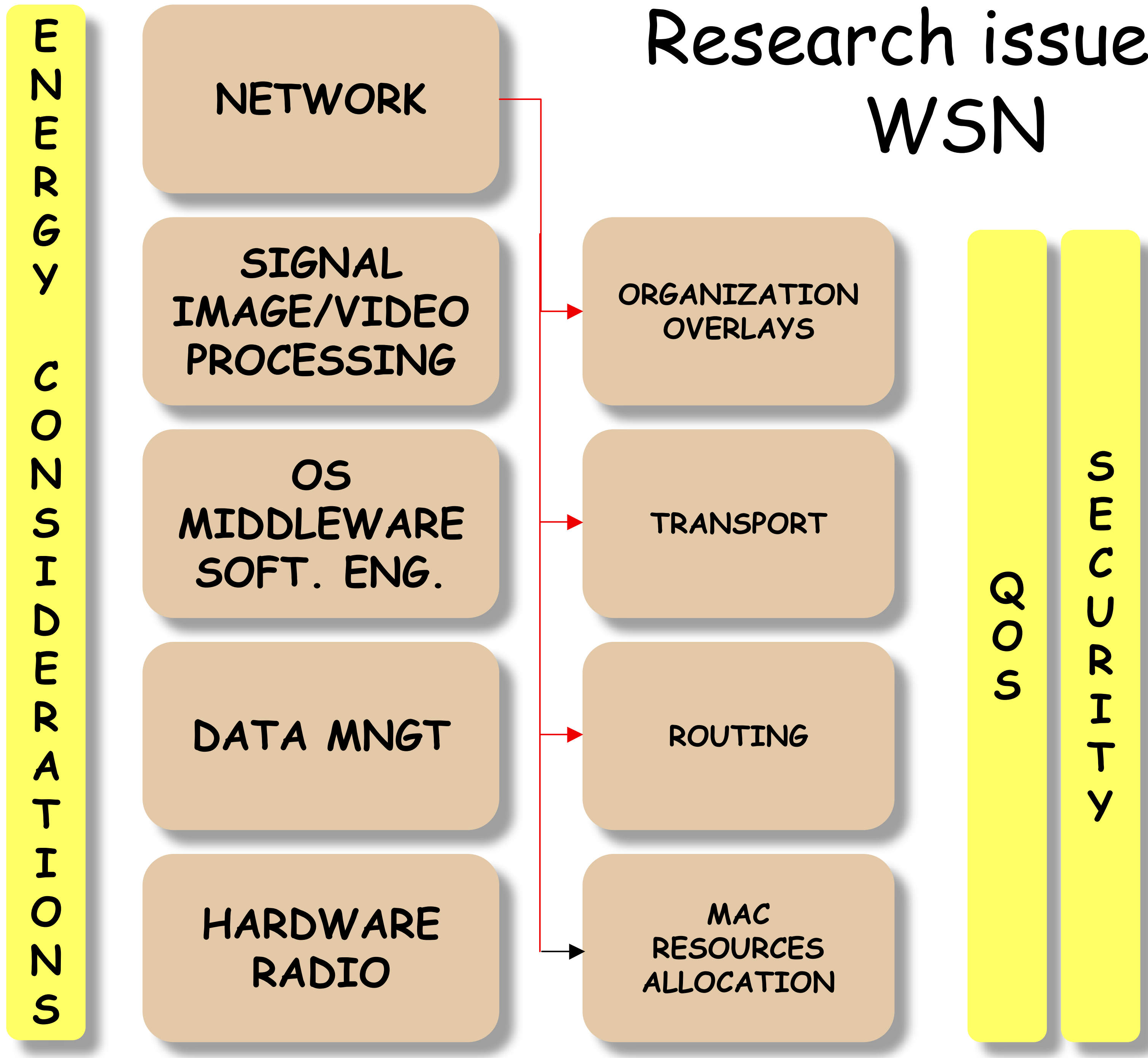

AVOIDS THE BLACK-BOX VISION

# One vision for enabling QoS in Sensor Nets (2)

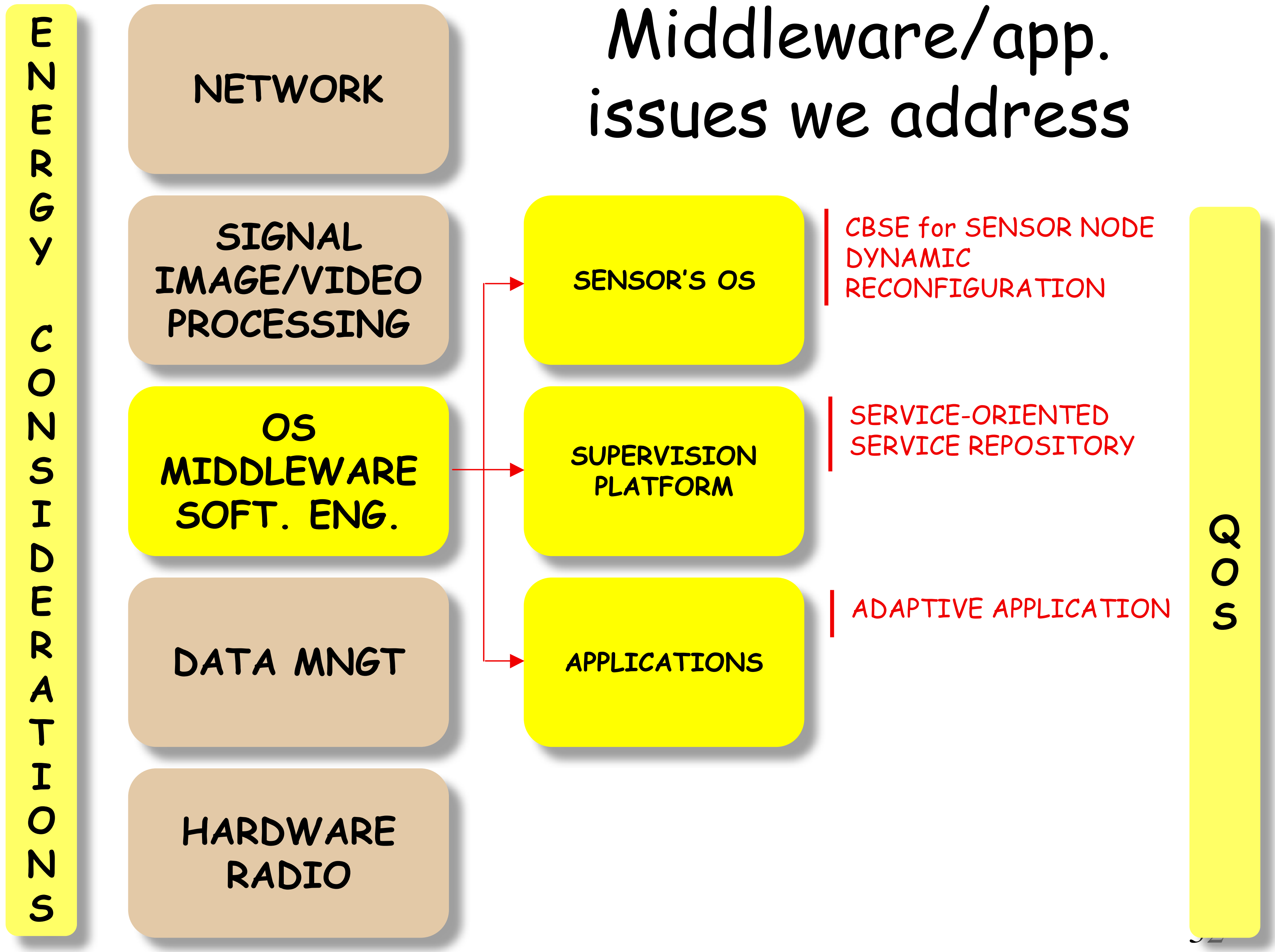



**AVOIDS THE BLACK-BOX VISION**

# Research issues in WSN



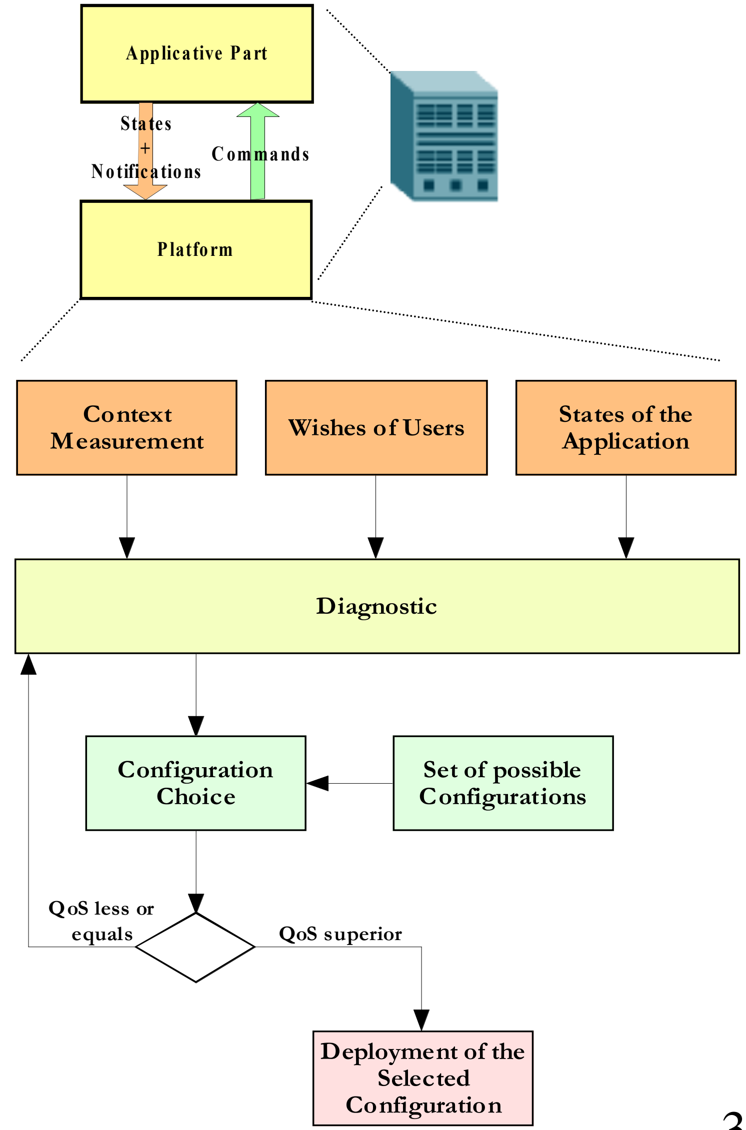
# Middleware/app. issues we address



# Supervision platform

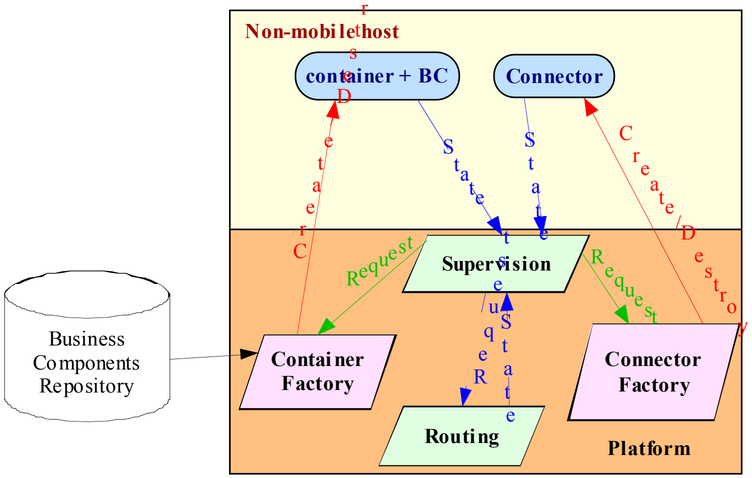
M. Dalmau & P. Roose

- Take care of user's QoS and QoS continuity
- Allows for a service-oriented surveillance system
- Discovery and publish mechanisms
- In charge of determining which configuration is better

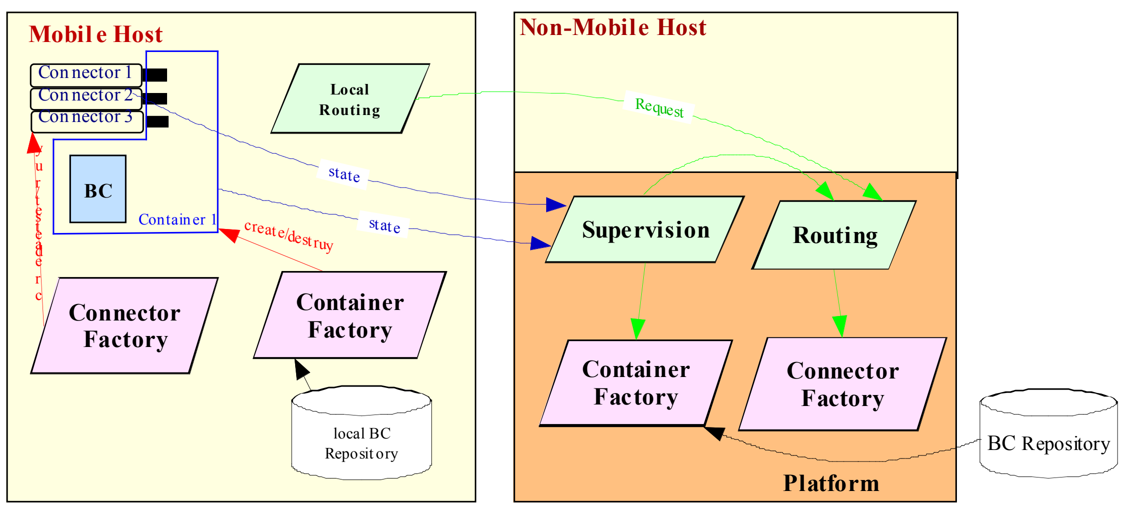


# A bit of the internal design

## Fixed-node/base station




## Mobile/lightweight-node



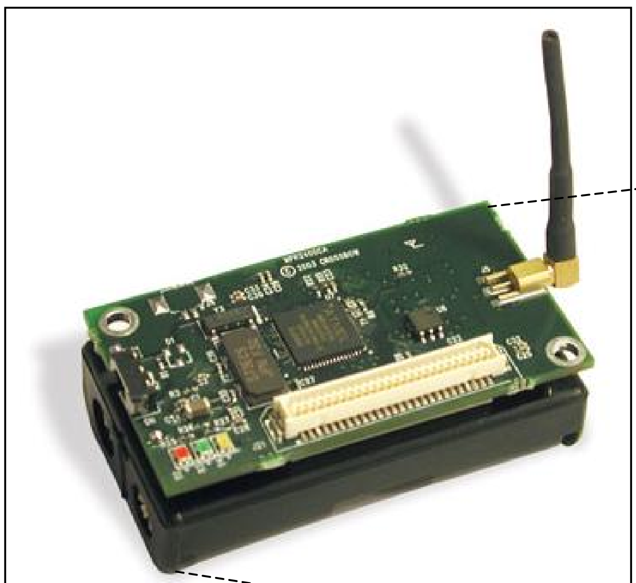
# Dynamic reconfiguration (1)

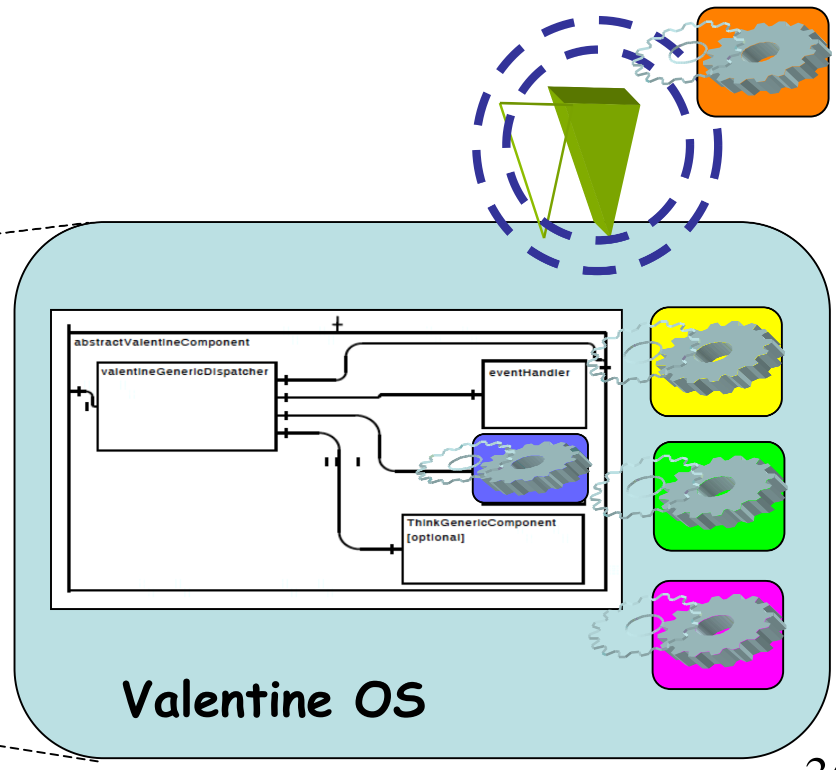
N. Hoang (PhD student)

- Avoids monolithic OS (à la TinyOS)
- We use the Think framework, which is an implementation of the Fractal component model to generate dynamic and reconfigurable OS and services
- First step towards the « active and programmable networking » concept applied to Wireless Sensor Networks

# Dynamic reconfiguration (2)

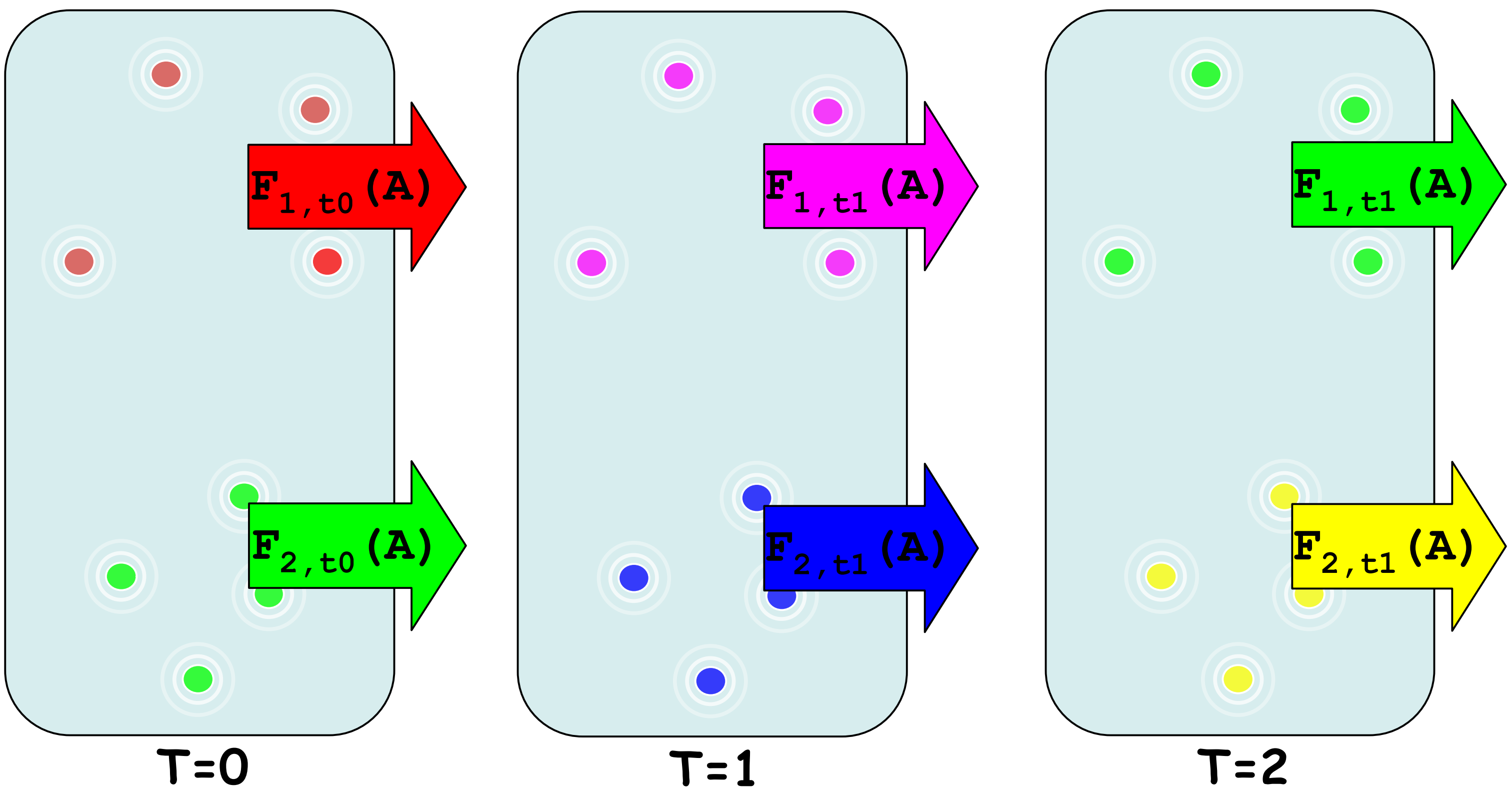
- Target platform: MicaZ
- Extension of the Think generic components
  ➔Valentine OS



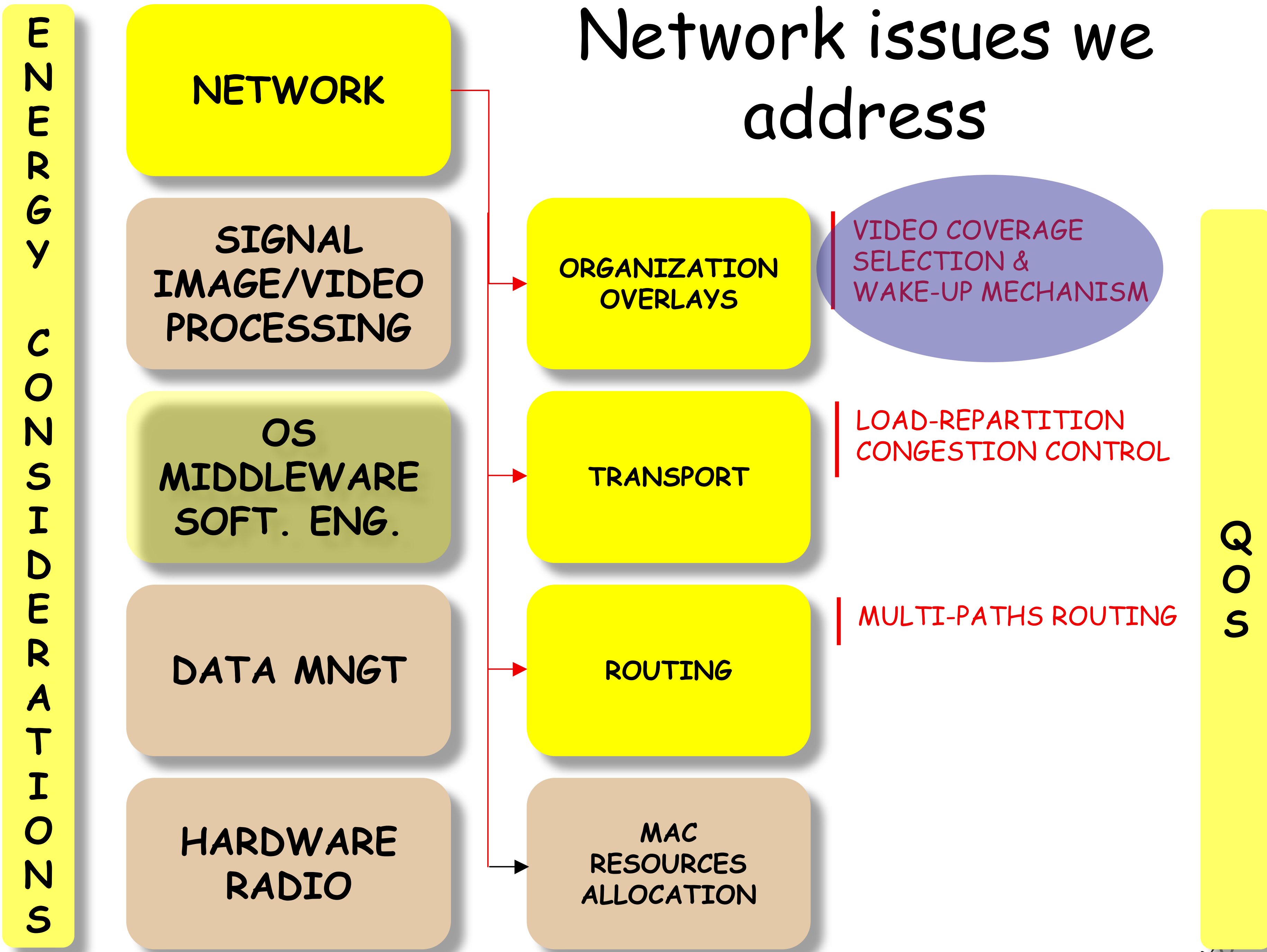


# Towards Service Oriented Architecture

- Fast reconfiguration enables dynamic and on-the-fly new services deployment

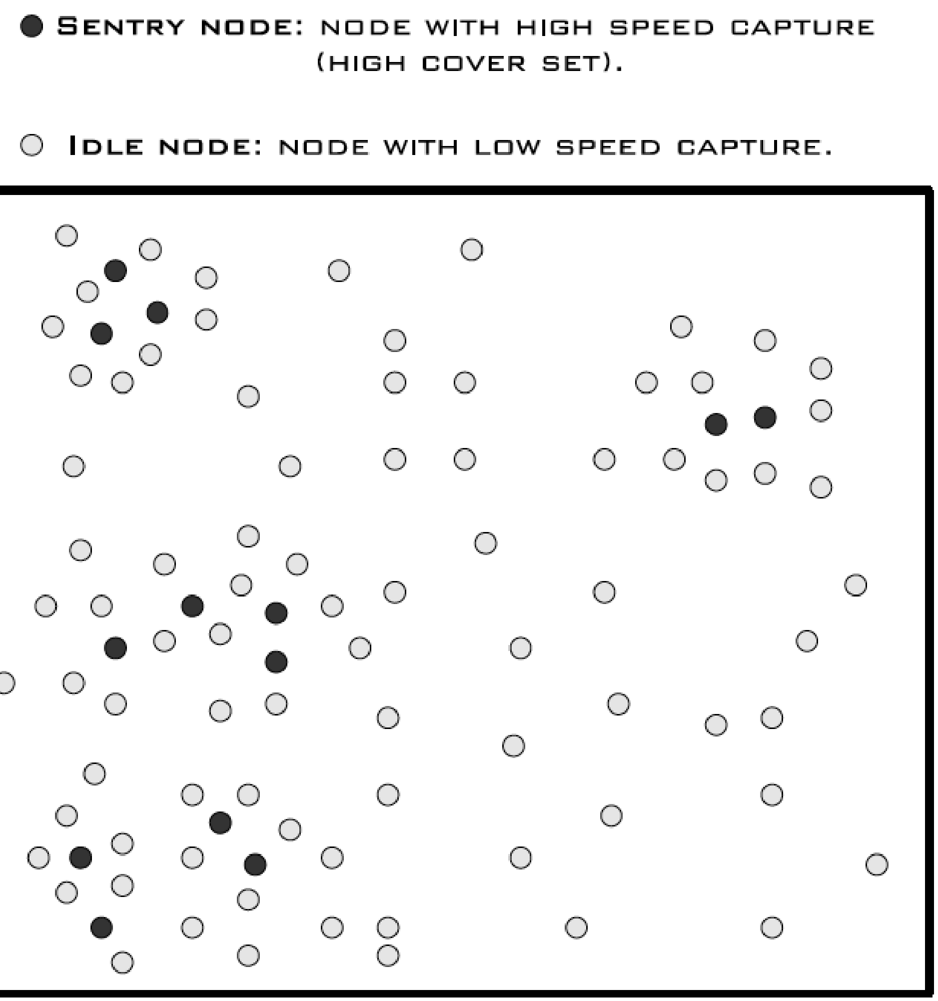


# Network issues we address



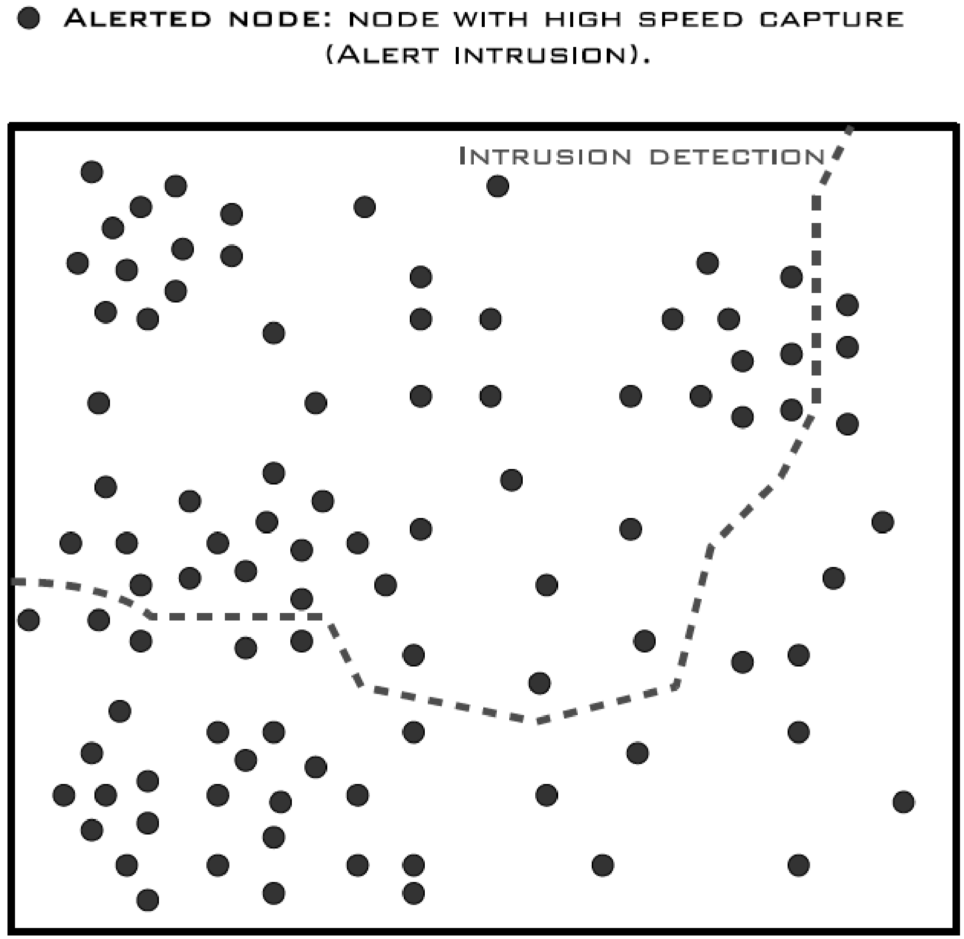
# Surveillance scenario (1)

- Randomly deployed video sensors
- Not only barrier coverage but general intrusion detection
- Most of the time, network in so-called *hibernate mode*
- Most of active sensor nodes in *idle mode* with low capture speed
- Sentry nodes with higher capture speed to quickly detect intrusions



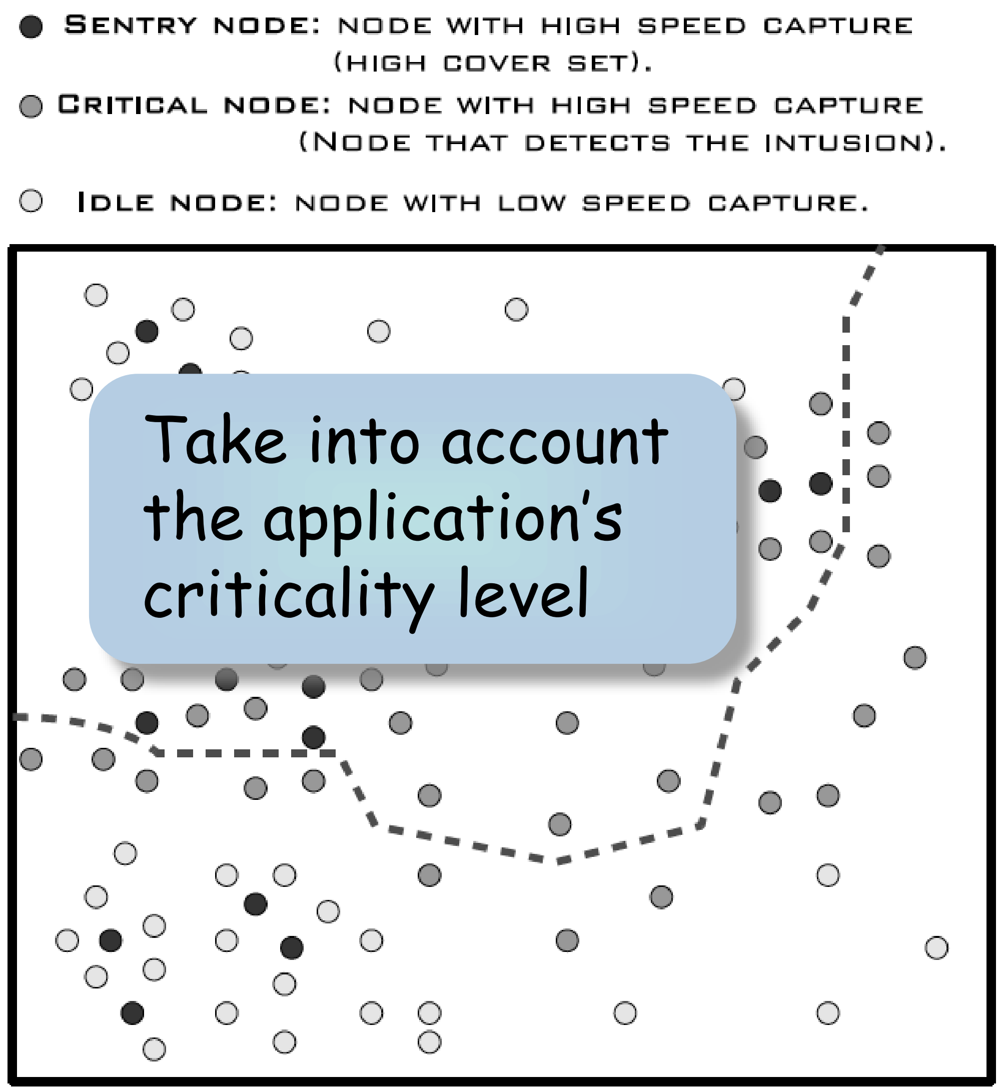
# Surveillance scenario (2)

- Nodes detecting intrusion must alert the rest of the network
- 1-hop to k-hop alert
- Network in so-called *alerted mode*
- Capture speed must be increased
- Ressources should be focused on making tracking of intruders easier



# Surveillance scenario (3)

- Network should go back to *hibernate mode*
- Nodes on the intrusion path must keep a high capture speed
- Sentry nodes with higher capture speed to quickly detect intrusions



# Application's criticality

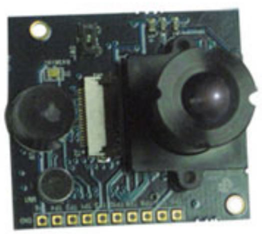
- All surveillance applications may not have the same criticality level, $r^0 \in [0,1]$
  - Environmental, security, healthcare,...
- Capture speed should decrease when $r^0$ decreases
- Sensor nodes could be initialized with a given $r^0$ prior to deployment

# How to meet app's criticality

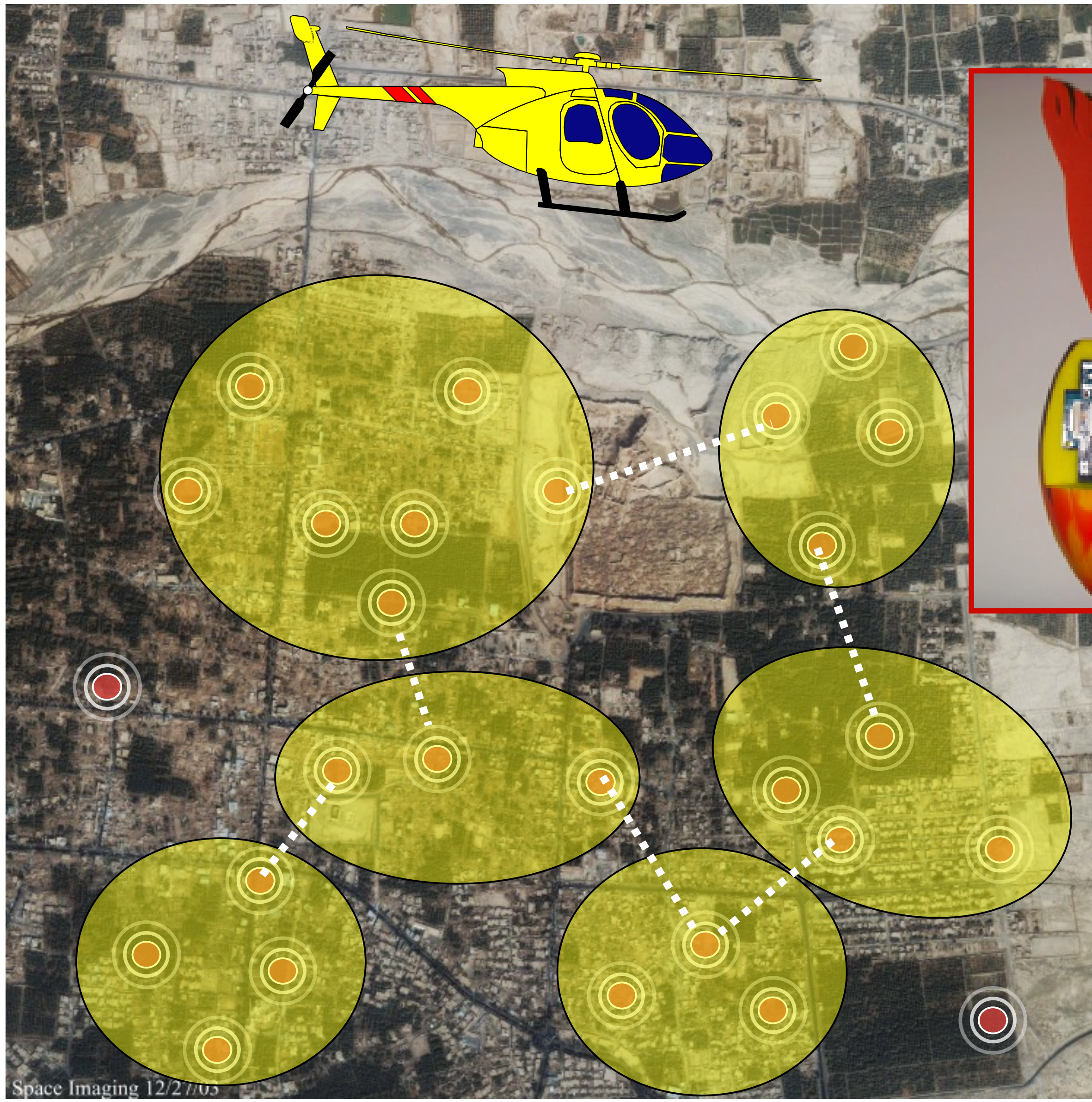
- Capture speed can be a « quality » parameter
- Capture speed for node v should depend on the app's criticality and on the level of redundancy for node v
- V's capture speed can increase when as V has more nodes covering its own FoV - cover set

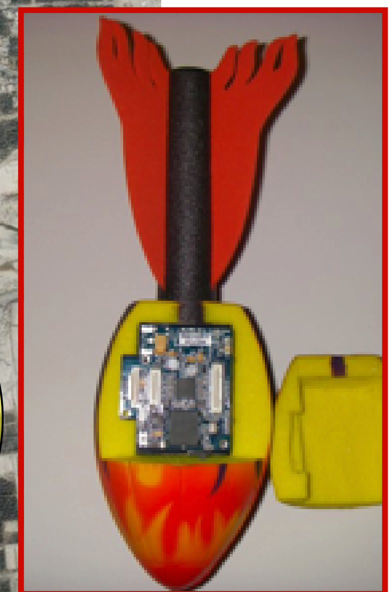
# Video Sensor Nodes

Imote2



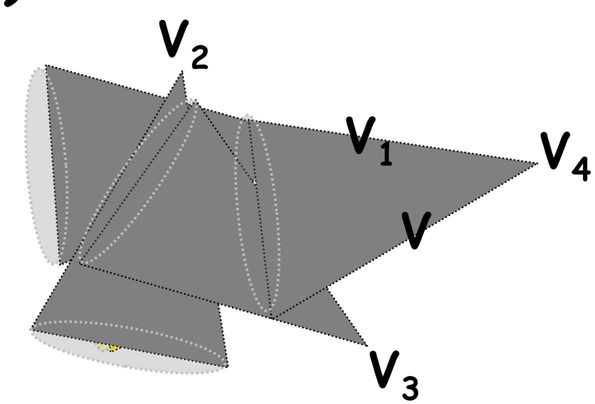
Multimedia board



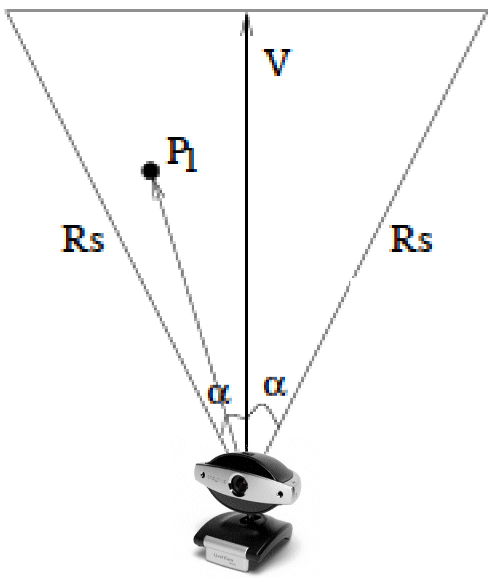
# Node's cover set

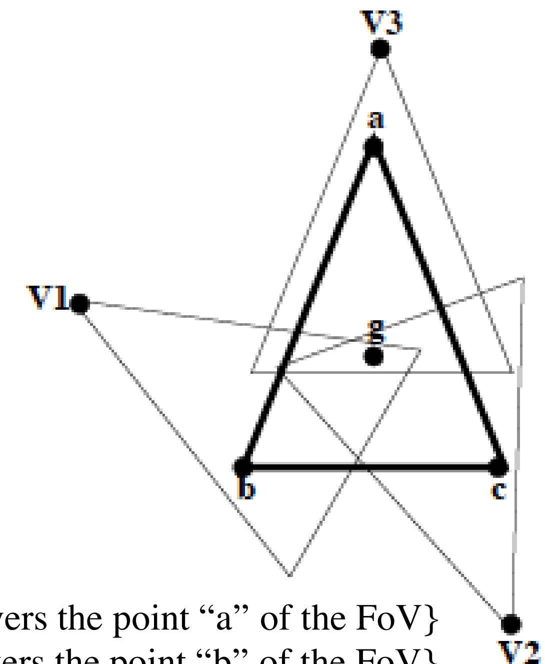
- Each node v has a Field of View, $FoV_v$
- $Co_i(v)$ = set of nodes v' such as $\bigcup_{v' \in Coi(v)} FoV_{v'}$ covers $FoV_v$
- $Co(v)$= set of $Co_i(v)$




$Co(v)=\{V_1,V_2,V_3,V_4\}$

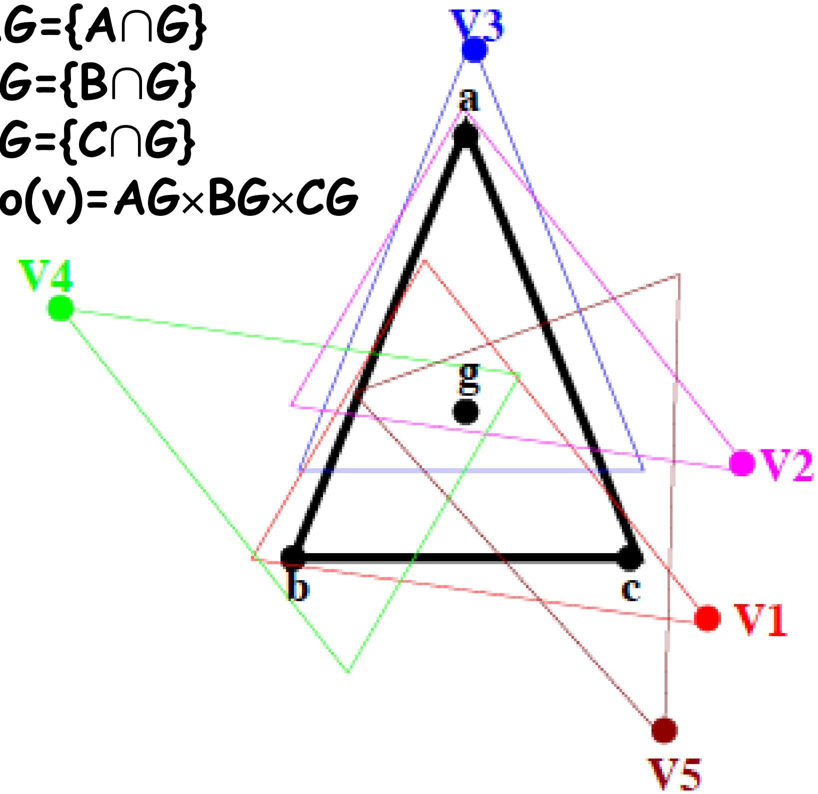
# Finding v's cover set




A = {v ∈ N(V ) : v covers the point "a" 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}

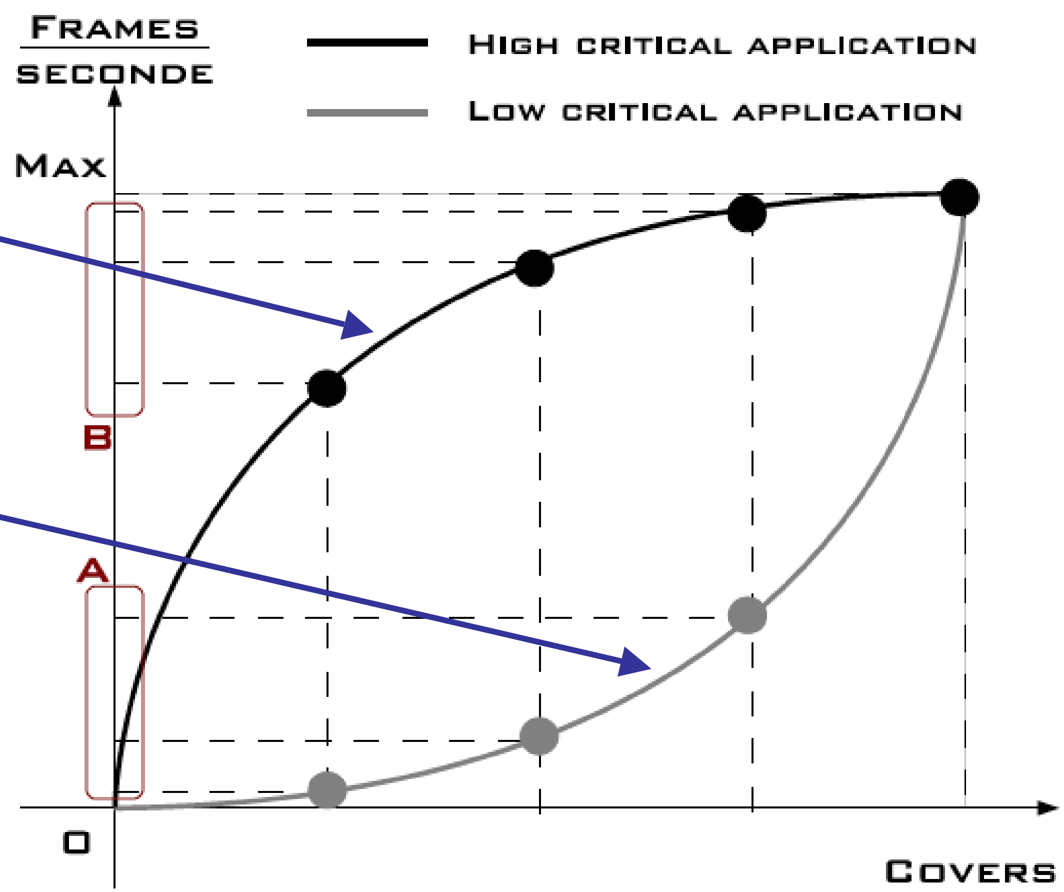
AG={A∩G}
BG={B∩G}
CG={C∩G}
Co(v)=AG×BG×CG




Co(V)= {
{V },
{V 2, V 1},
{V 3, V 1},
{V 2, V 4, V 5},
{V 3, V 4, V 5}}

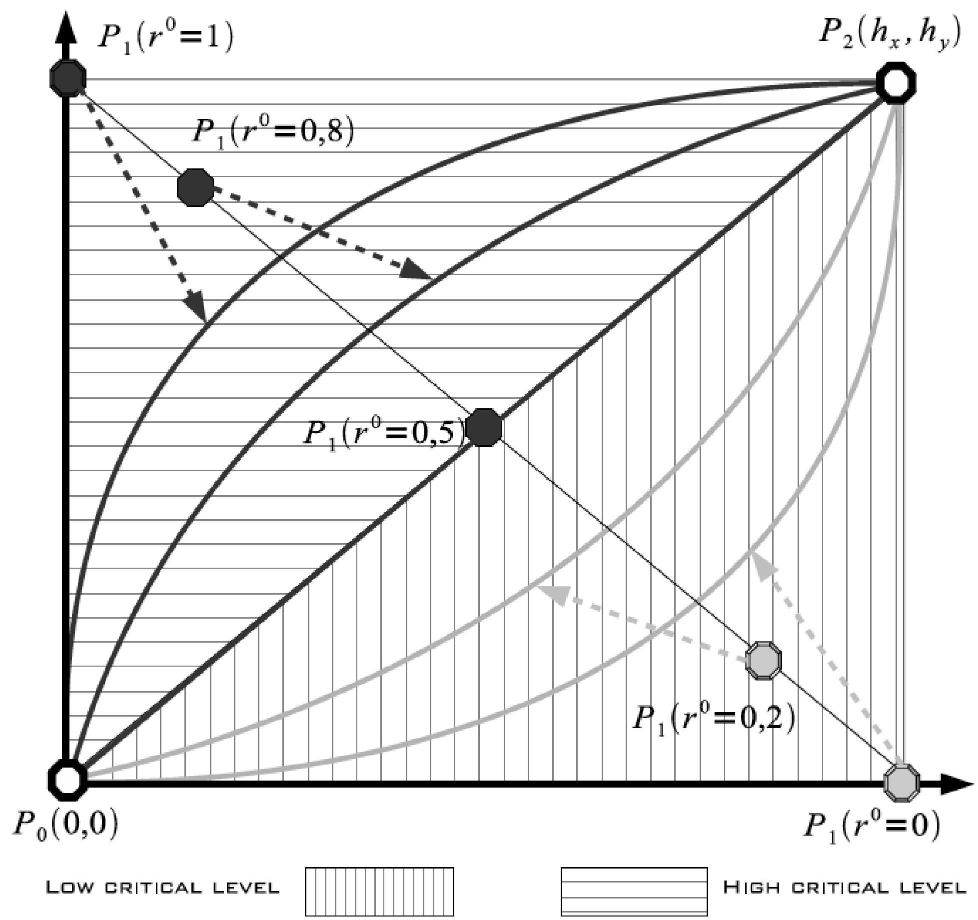
# Criticality model (1)

- 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



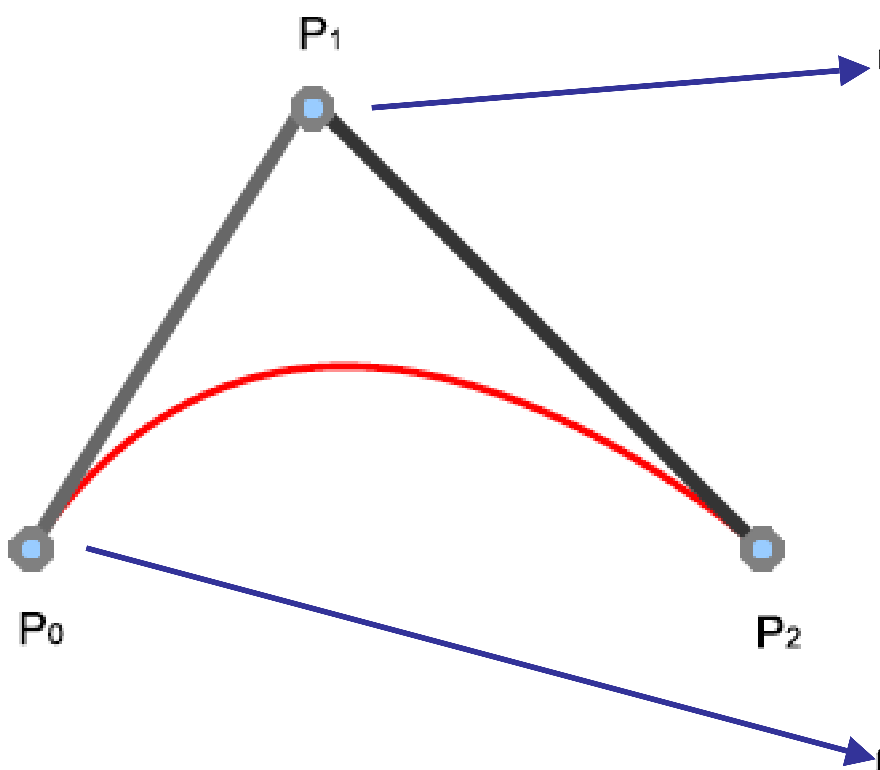
# Criticality model (2)

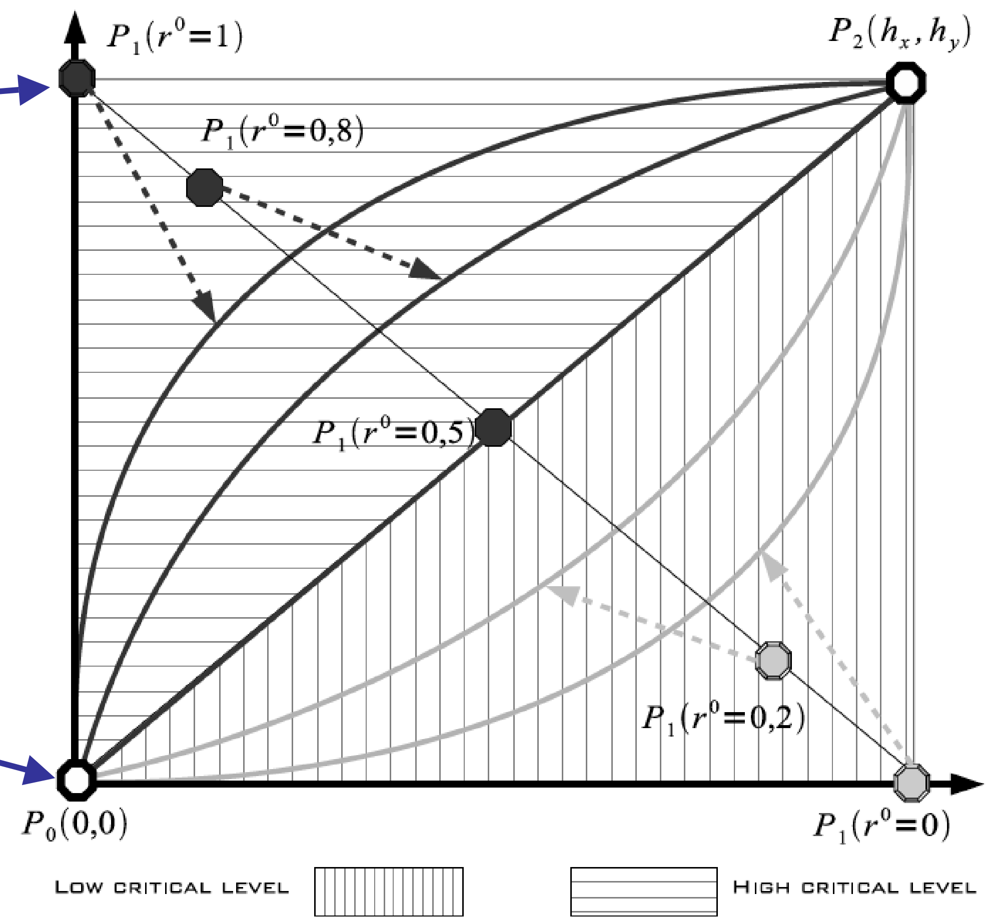
- $r^0$ can vary in [0,1]
- BehaVior functions (BV) defines the capture speed according to $r^0$
- $r^0 < 0.5$
  - Concave shape BV
- $r^0 > 0.5$
  - Convex shape BV
- We propose to use Bézier curves to model BV functions



# BehaVior function

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



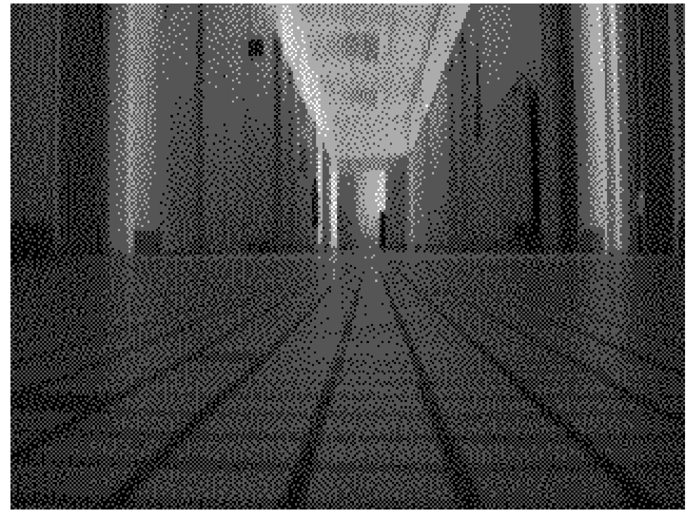
# Some typical capture speed

- Maximum capture speed is 6fps
- Nodes with size of cover set greater than 6 capture at the maximum speed

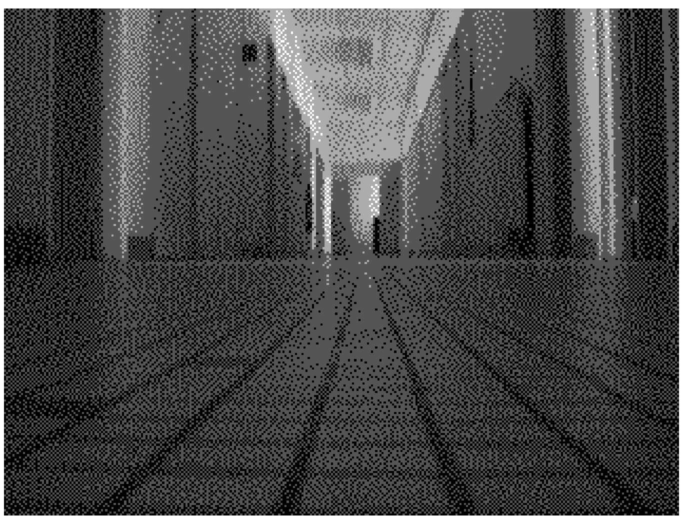
| $r^0$ \ $\lvert Co(v) \rvert$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 0.0 | 0.05 | 0.20 | 0.51 | 1.07 | 2.10 | 6.00 |
| 0.2 | 0.30 | 0.73 | 1.34 | 2.20 | 3.52 | 6.00 |
| 0.5 | 1.00 | 2.00 | 3.00 | 4.00 | 5.00 | 6.00 |
| 0.8 | 2.48 | 3.80 | 4.66 | 5.27 | 5.70 | 6.00 |
| 1.0 | 3.90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

320x200, 30 fps, 256 gray scale
15Mbps raw

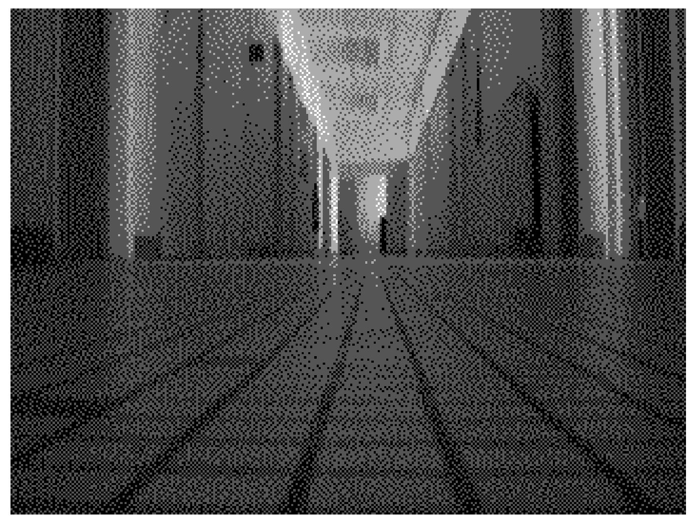
5 fps, 4 gray scale, 640kbps raw



2 fps, 4 gray scale, 256kbps raw



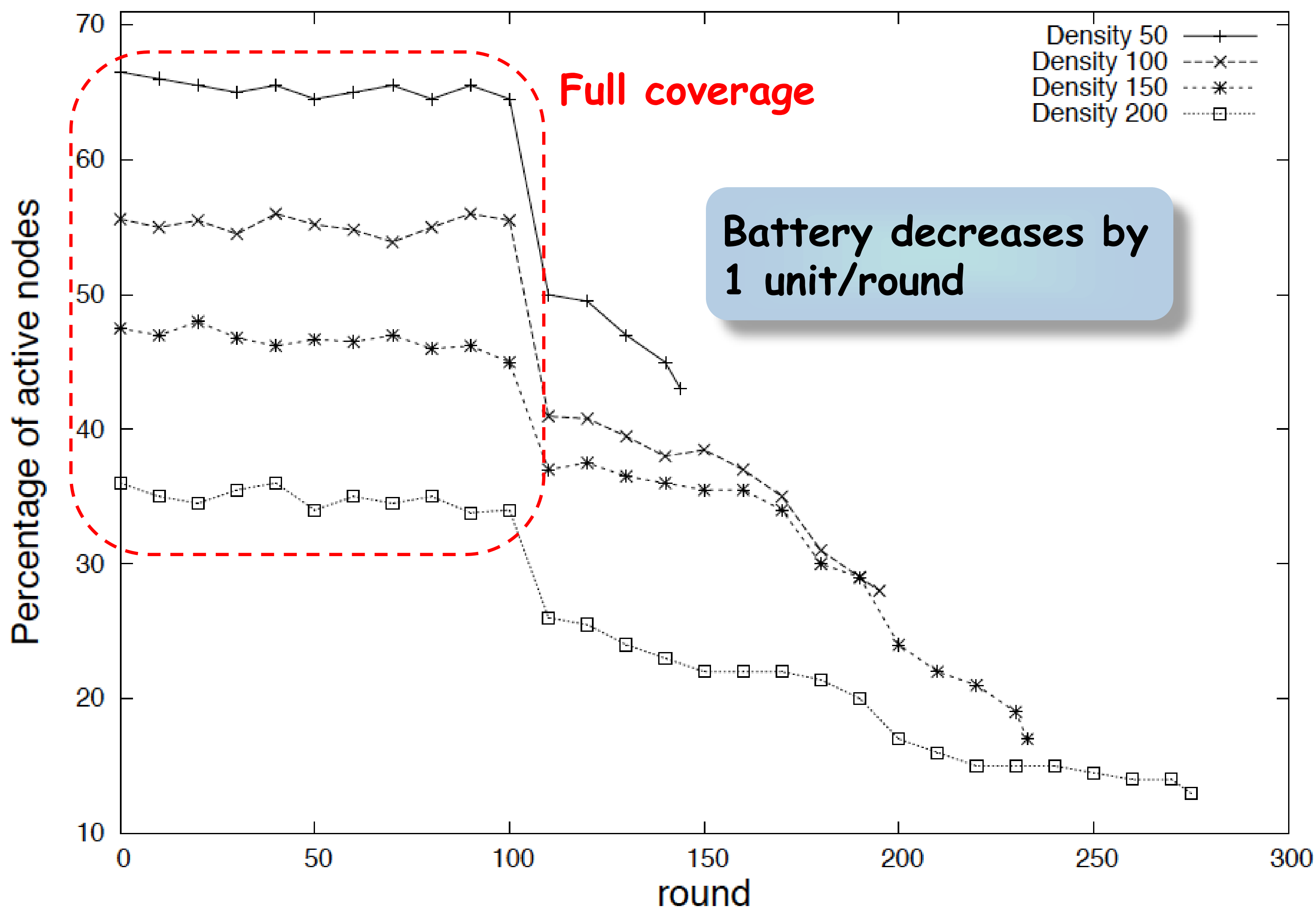
1 fps, 4 gray scale, 128kbps raw



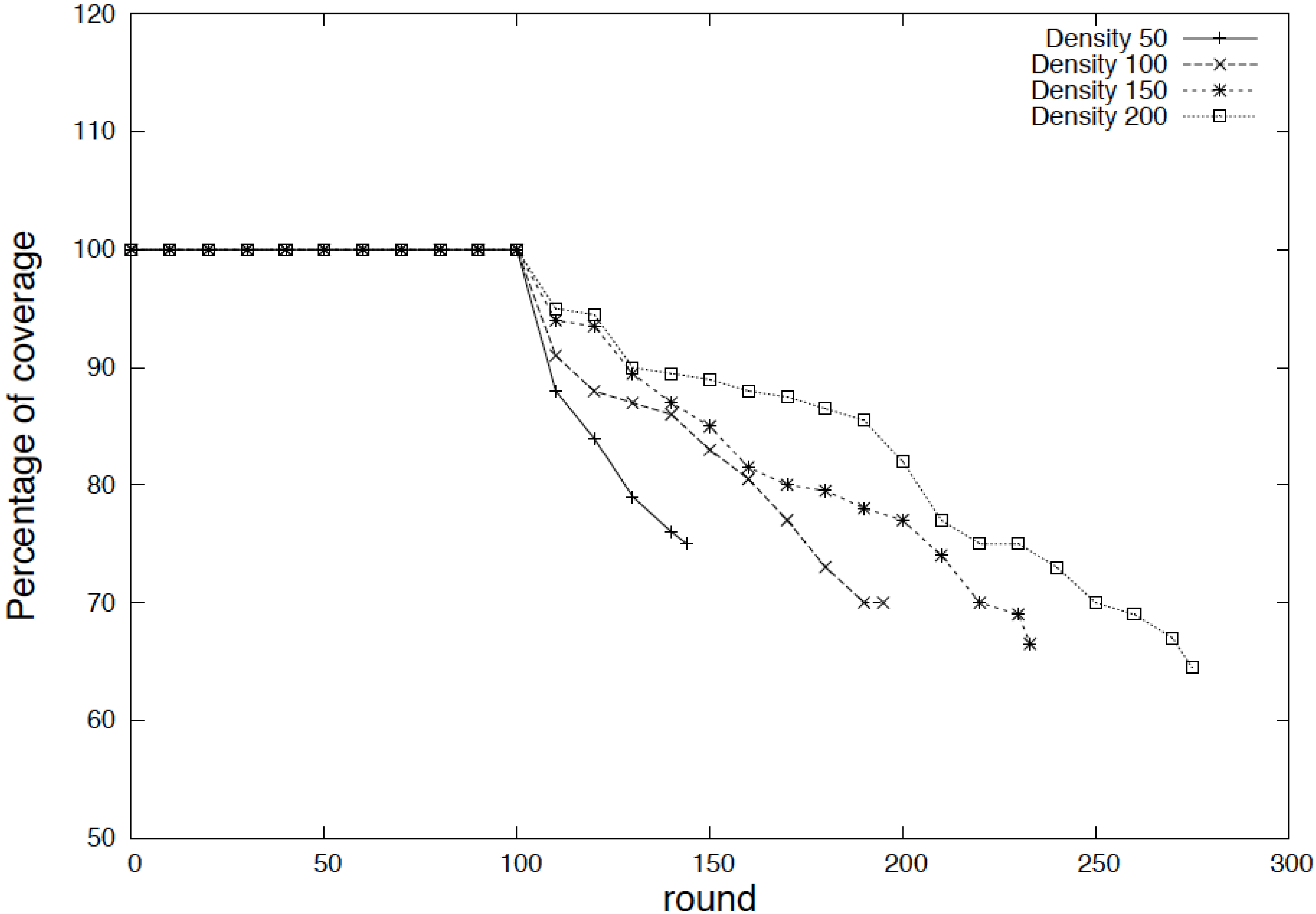
# Simulation settings

- OMNET++ simulation model
- Video nodes have communication range of 30m and video sensing range of 25m, FoV is a sector of 60°
- Battery has 100 units
- Full coverage is defined as the region initially covered when all nodes are active

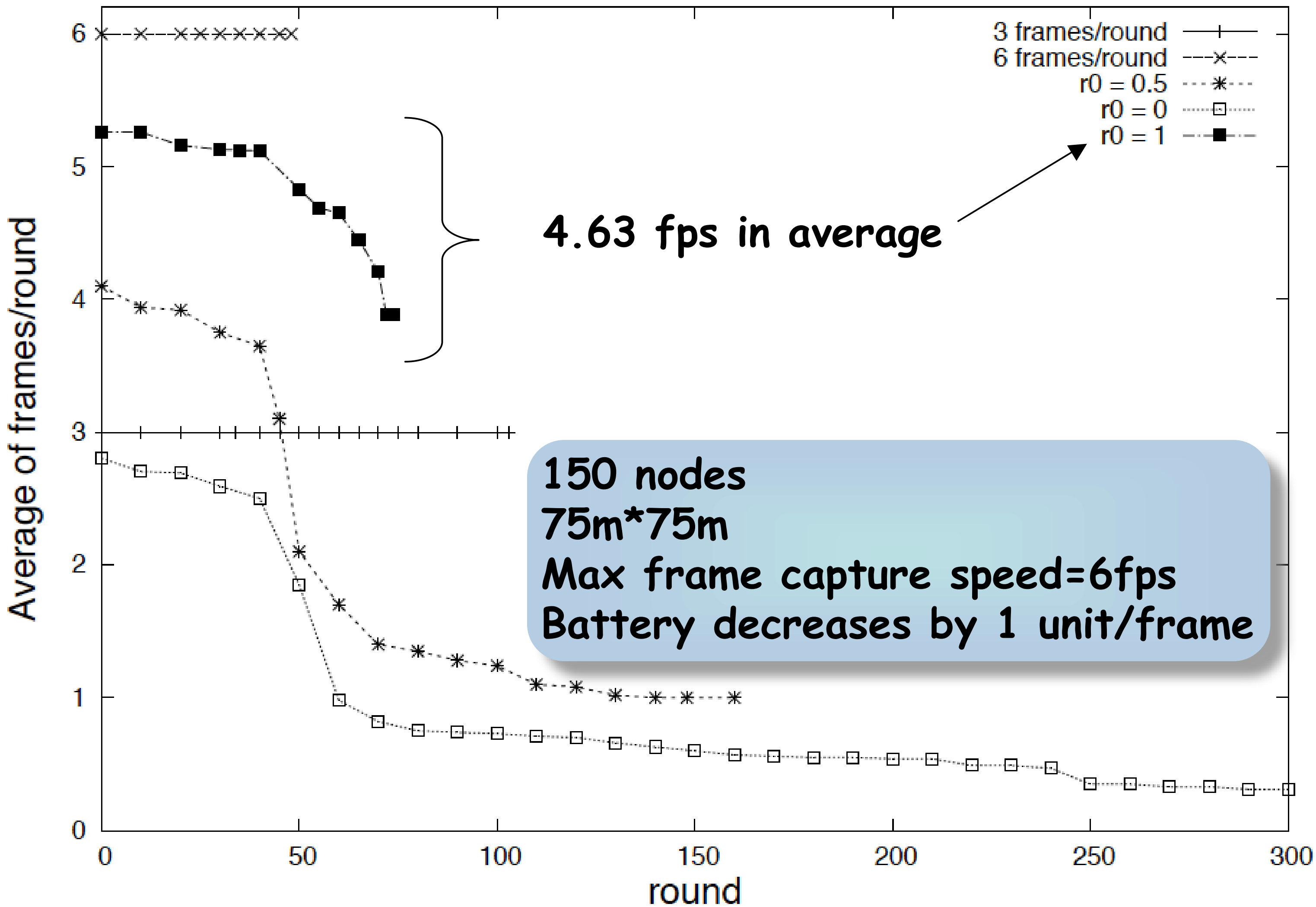
# Percentage of active nodes



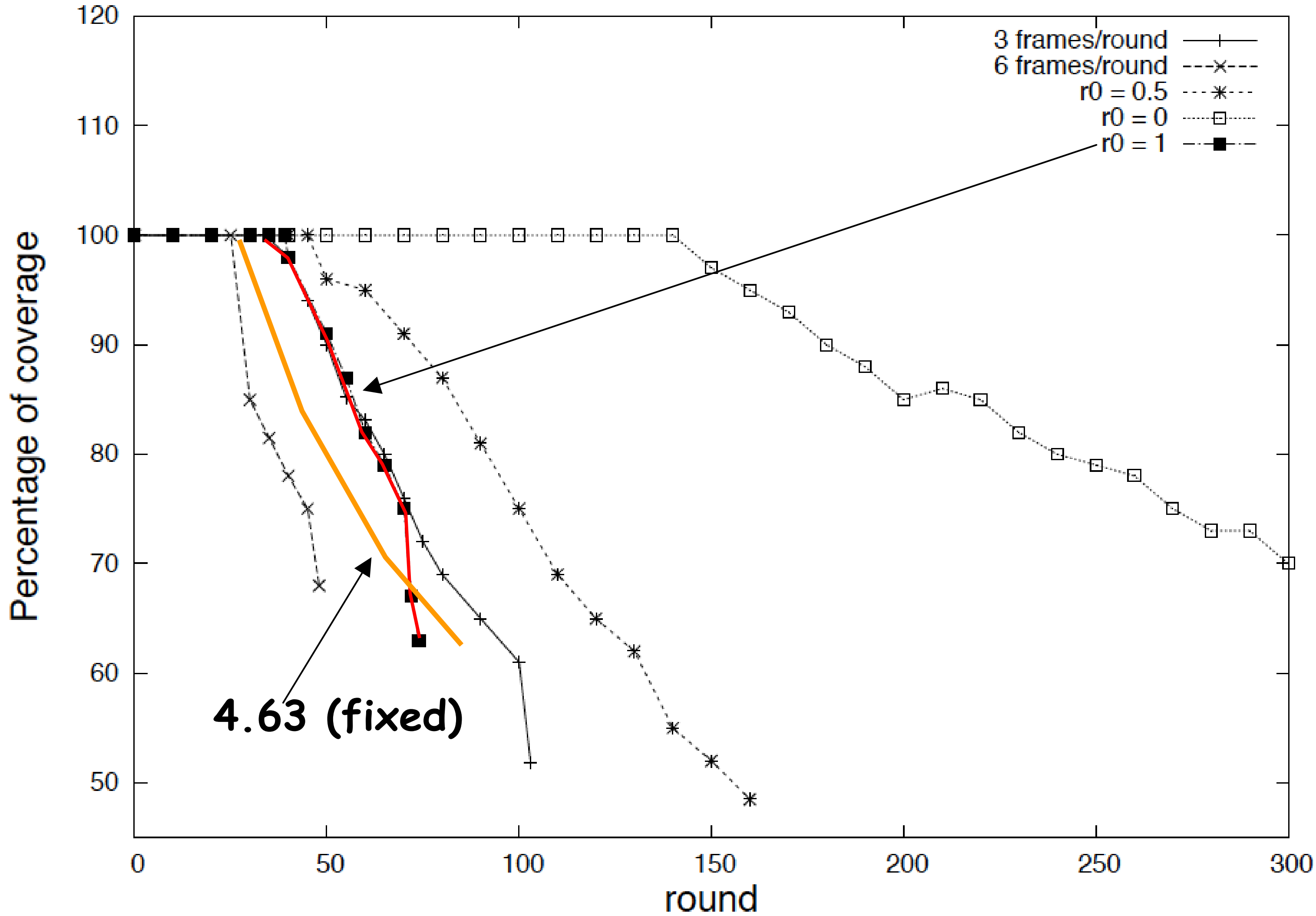
# Percentage of coverage



# Average capture speed



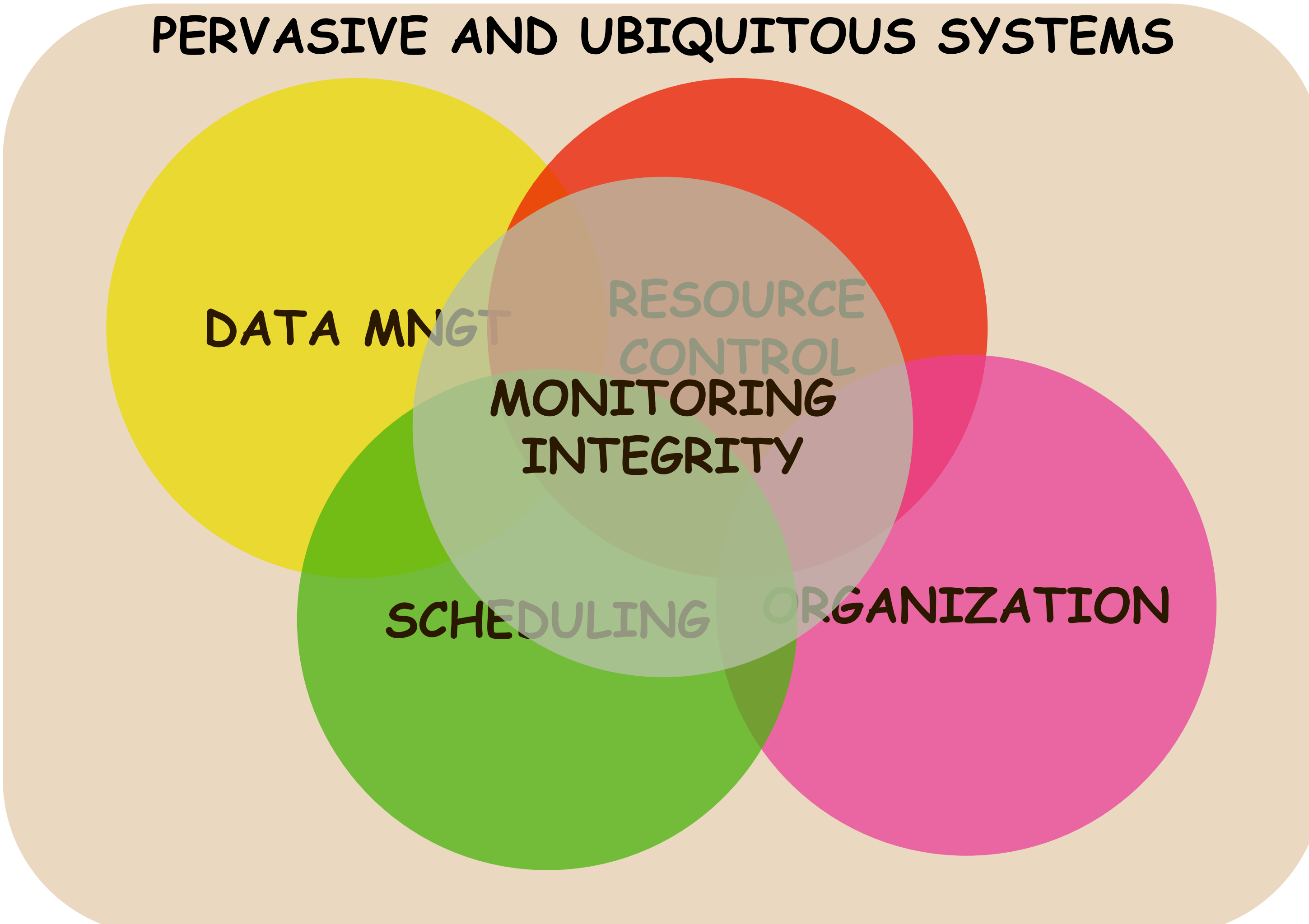
# Fixed vs adaptive



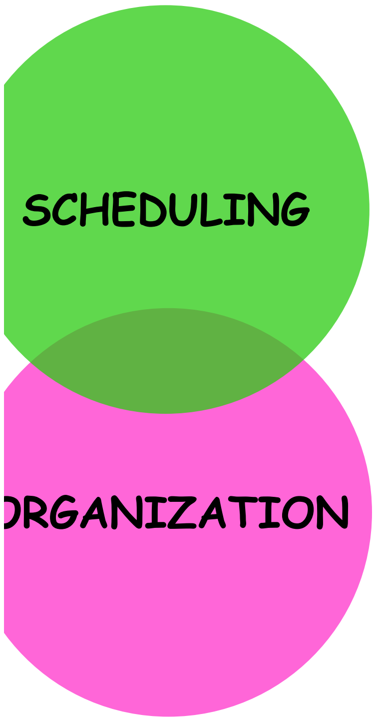
# Conclusions

- Surveillance applications have a high level of criticity which make accountability important
- Criticality model with adaptive scheduling of nodes
- Optimize the resource usage by dynamically adjusting the provided service level
- Extension for risk-based scheduling in intrusion detection systems

# Research directions

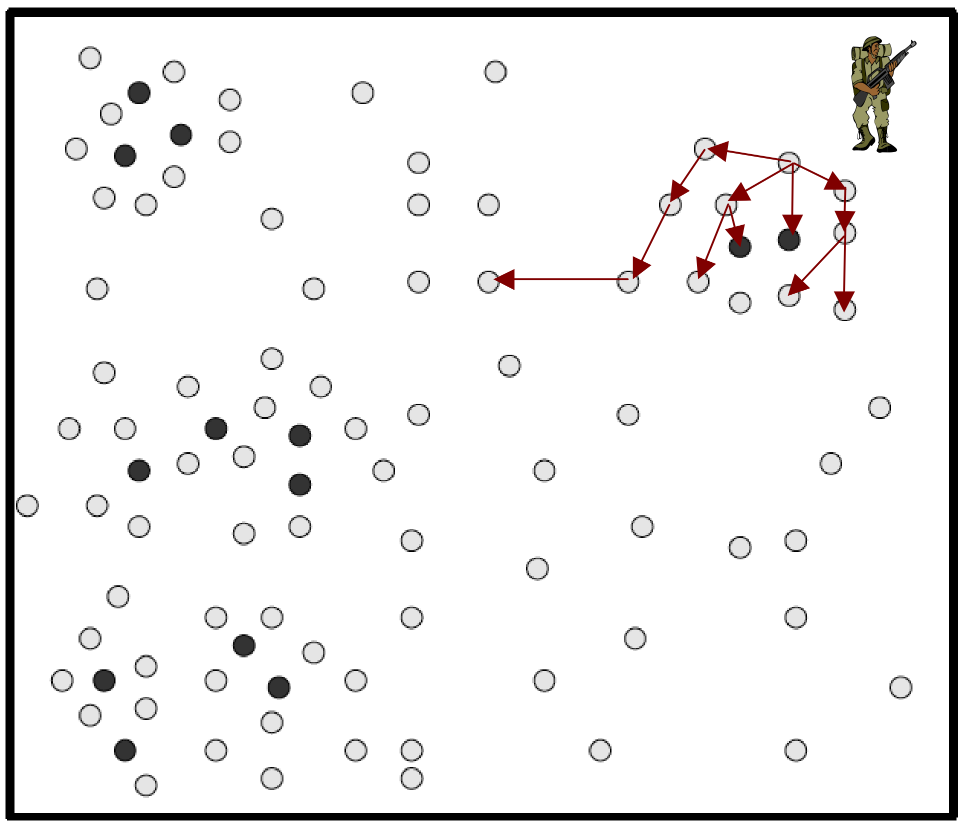


# Controlled propagation (1)

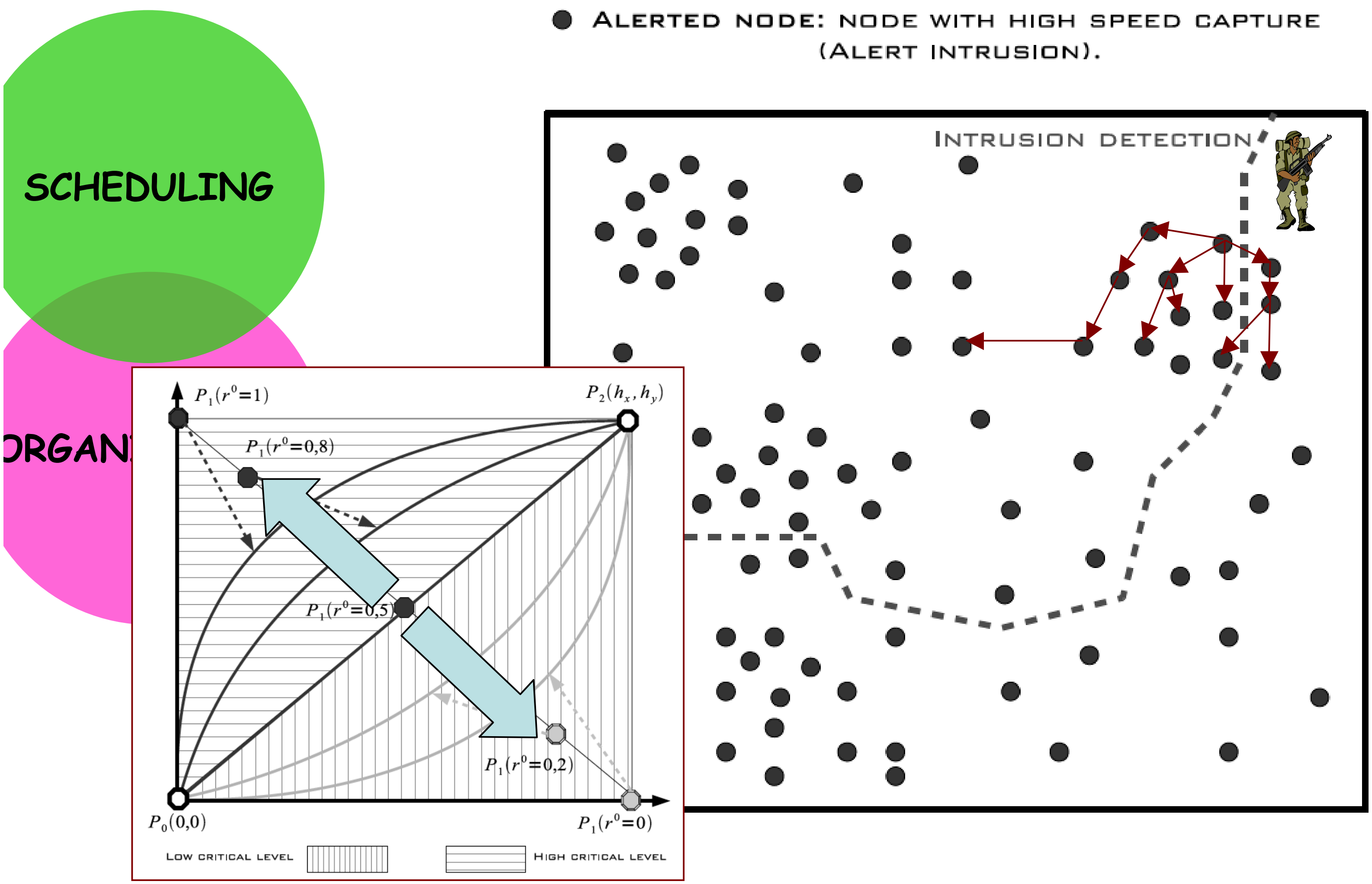



● SENTRY NODE: NODE WITH HIGH SPEED CAPTURE (HIGH COVER SET).

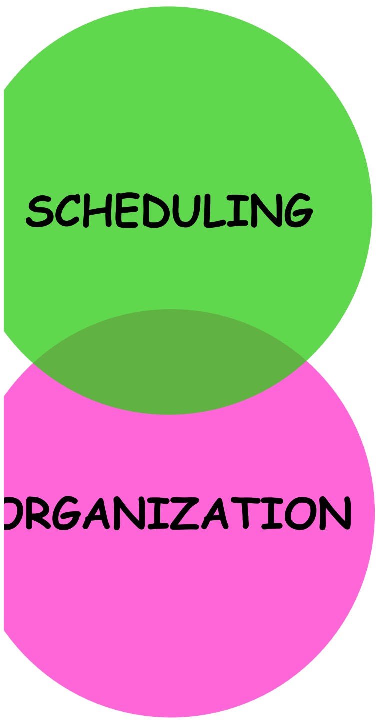
○ IDLE NODE: NODE WITH LOW SPEED CAPTURE.



# Controlled propagation (2)



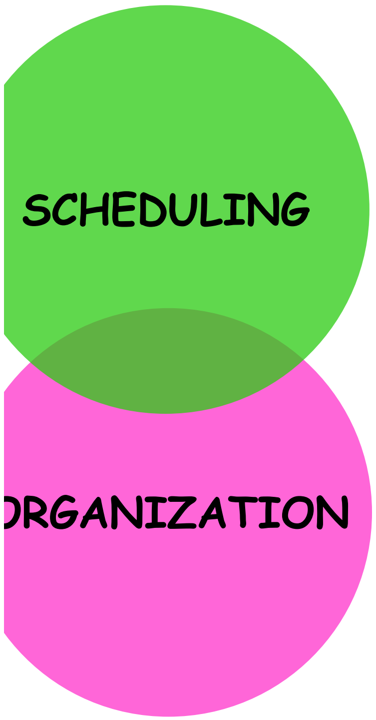
# Controlled propagation (3)

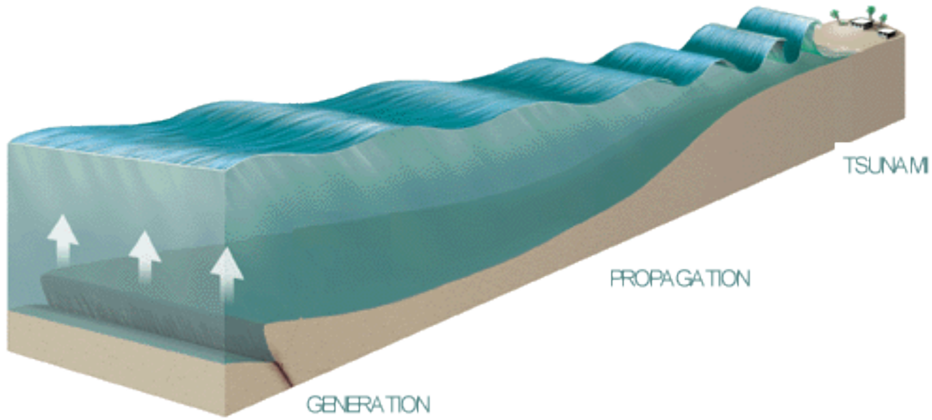


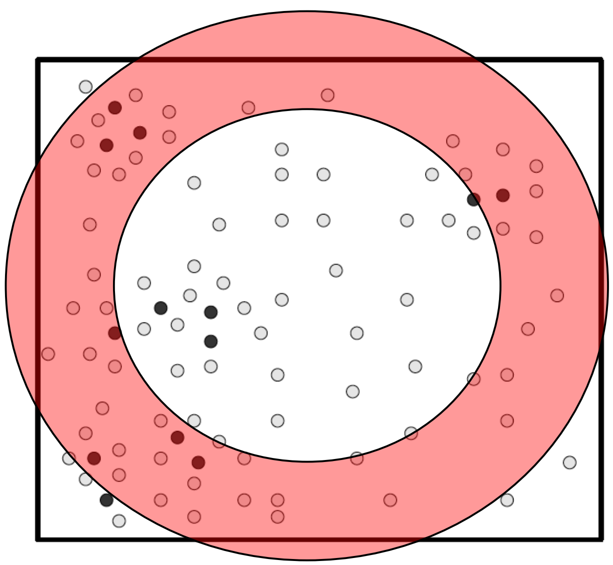

- Not a simple propagation or broadcast algorithm
  - Not all nodes need to be at the maximum (same) alert level
  - Which nodes should be more than others?
- Borrow propagation model from other disciplines
  - Epidemic propagation, percolation, wave propagation,…
  - According to the model, map the parameter of a surveillance system to the model's parameters

# Controlled propagation (4)

ex: tsunami generation



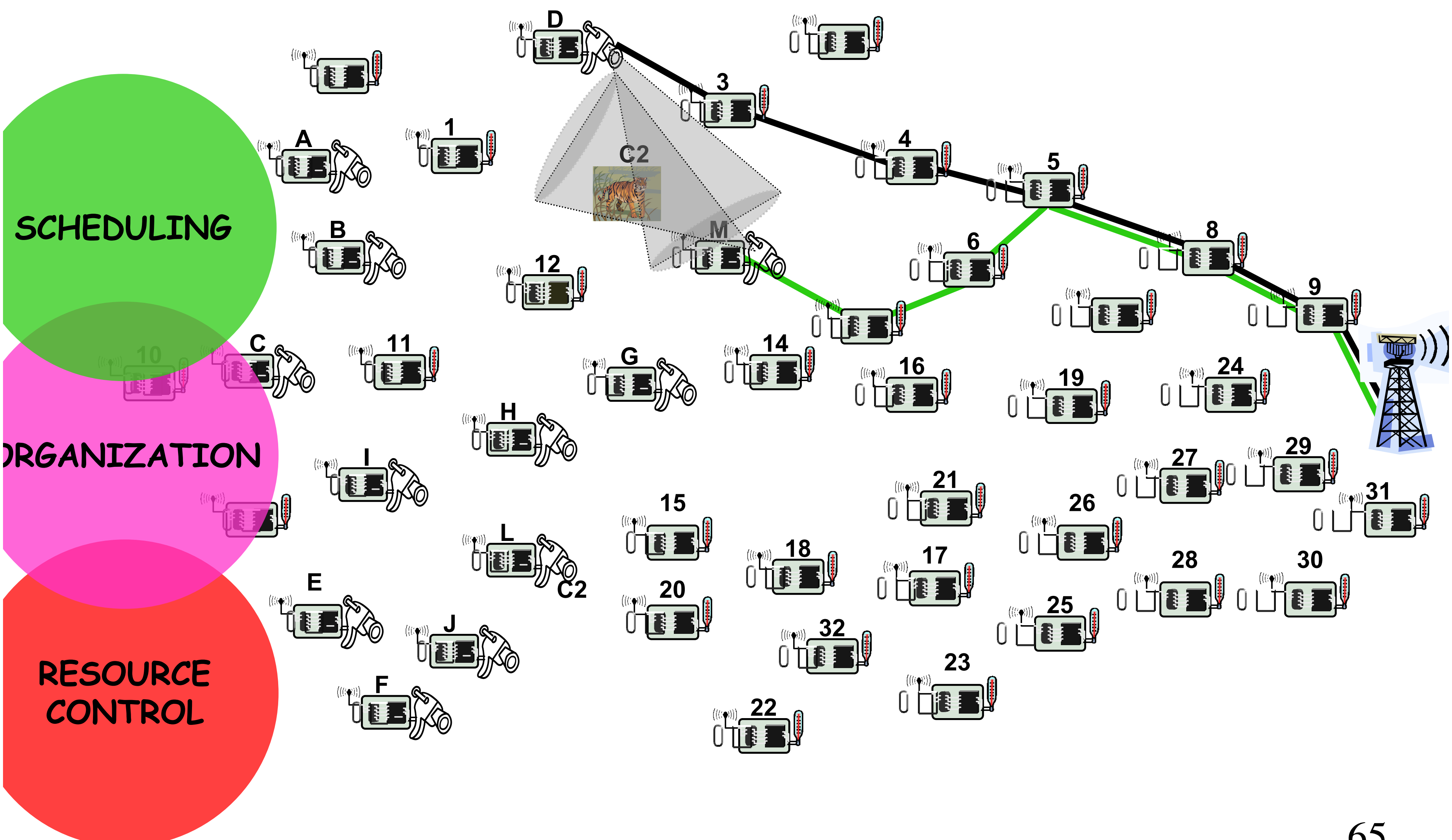



sensor nodes near the border may need to be « alerted » than others, they could have an amplification factor greater than those near the centre

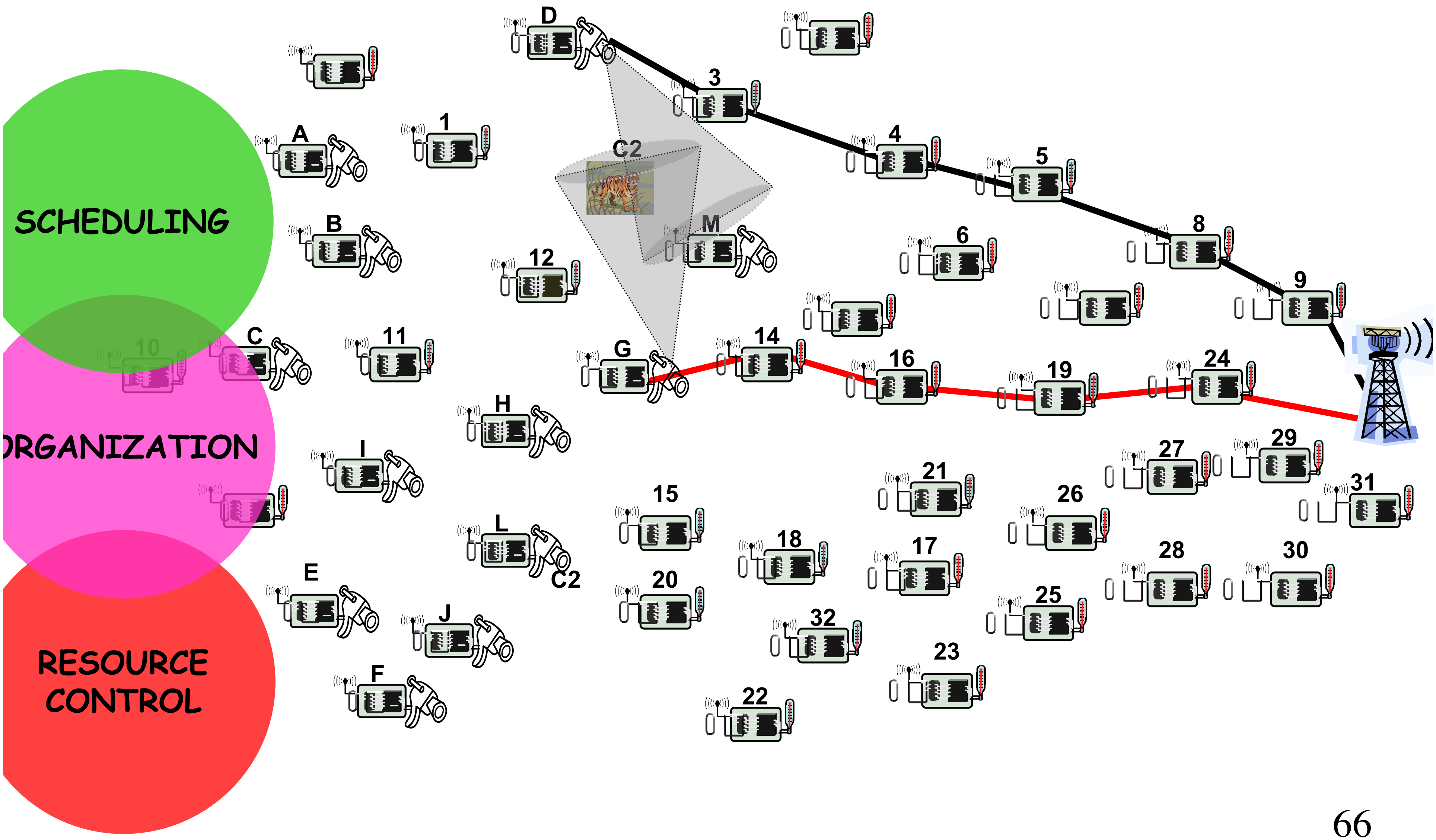
# Congestion control (1)


- Lot's of sensor nodes=lot's of trafic
- High probability of bottleneck, lot's of packet drop, no useful data back to user!
- Scheduling is tighly linked to resource control to be efficient
- Scheduling is then not only find these nodes that « see » the event, but also how to select a subset of those nodes that minimizes congestion

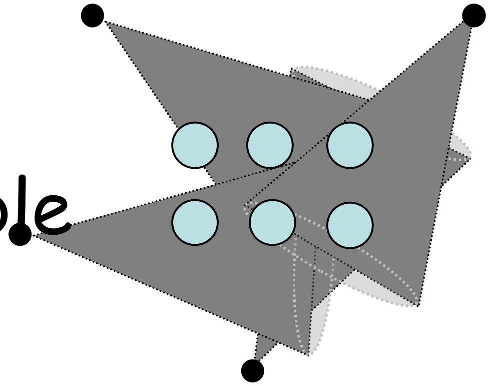
# Congestion control (2)



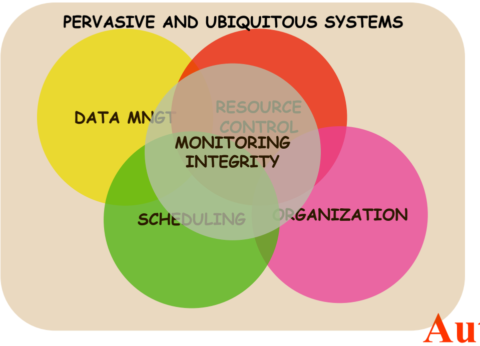
# Congestion control (3)



# Scheduling cover-set




- On intrusion, it is desirable to use more camera
  - To circumvent occlusions
  - To help for disambiguation
- It is not necessary that all activated camera capture at a same speed (probably high speed)
  - How define different the target frame capture speed for each node of the same cover set?




# Towards wide-area situation awareness

Combination of randomly and manually deployed sensors