Large-Scale Intrusion Detection with Low-Cost multi-camera Wireless Image Sensors

C. Pham (Univ. Pau, France)

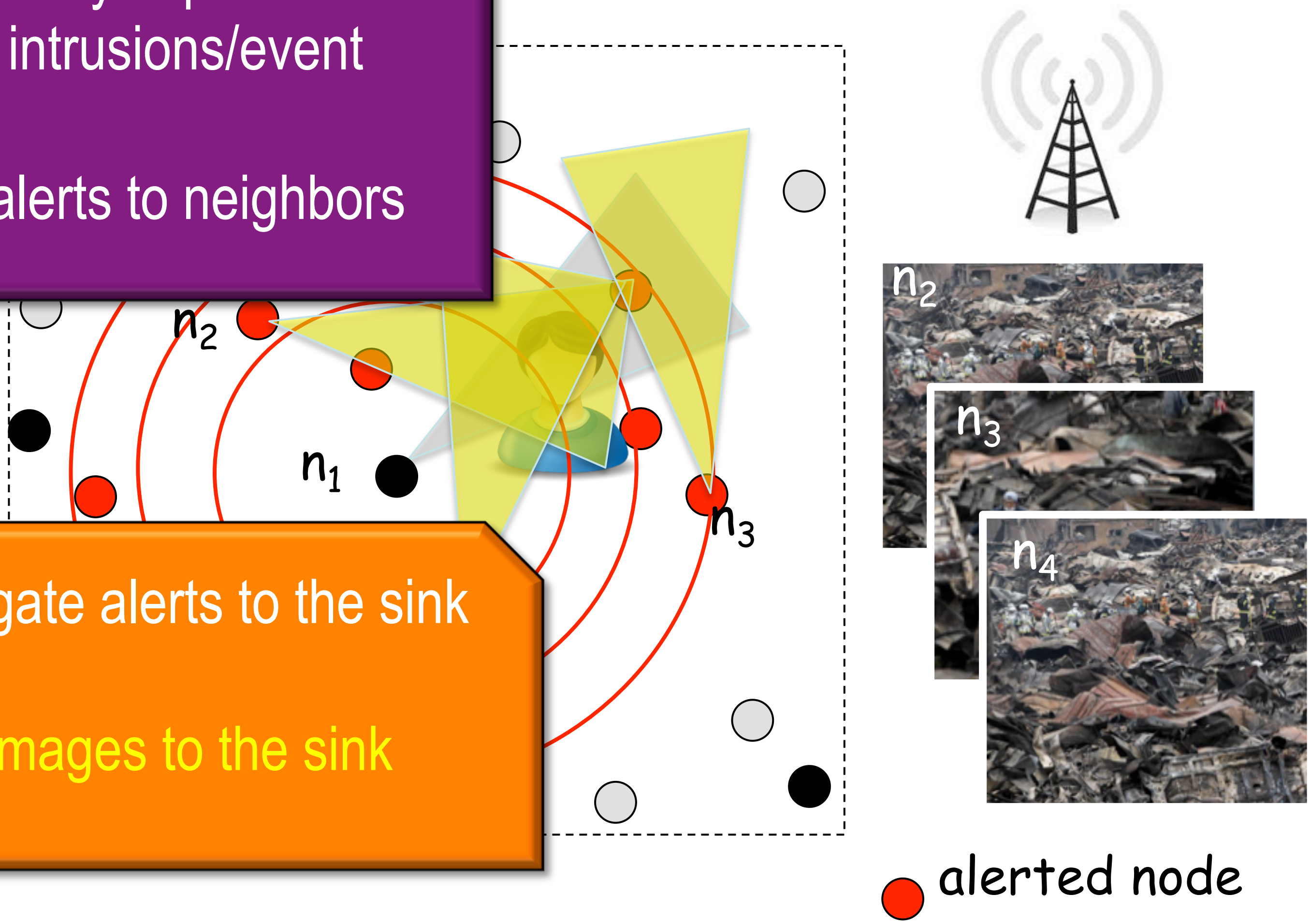
IEEE WiMob 2015
Abou Dhabi, UAE
October 20th, 2015

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

IMAGE SENSORS FOR SURVEILLANCE

Periodically capture to detect intrusions/event

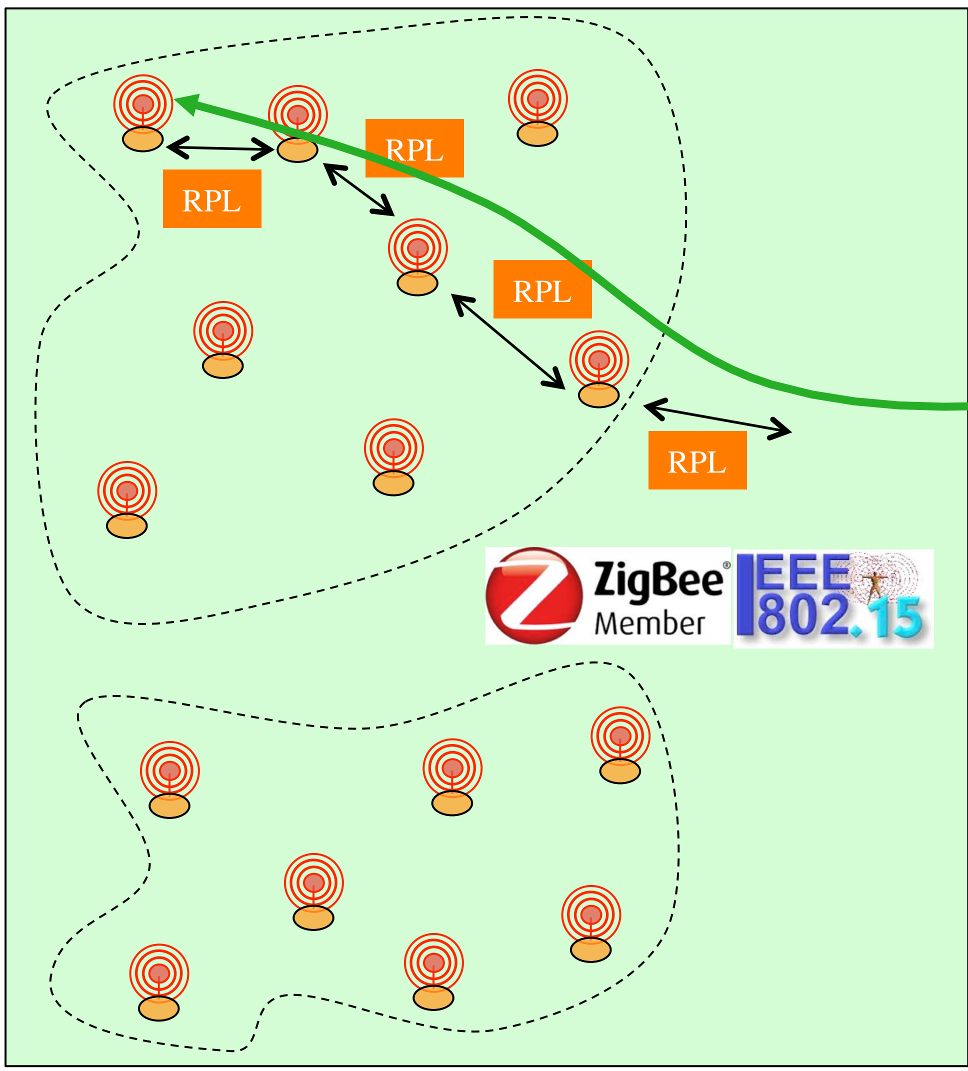
Send alerts to neighbors

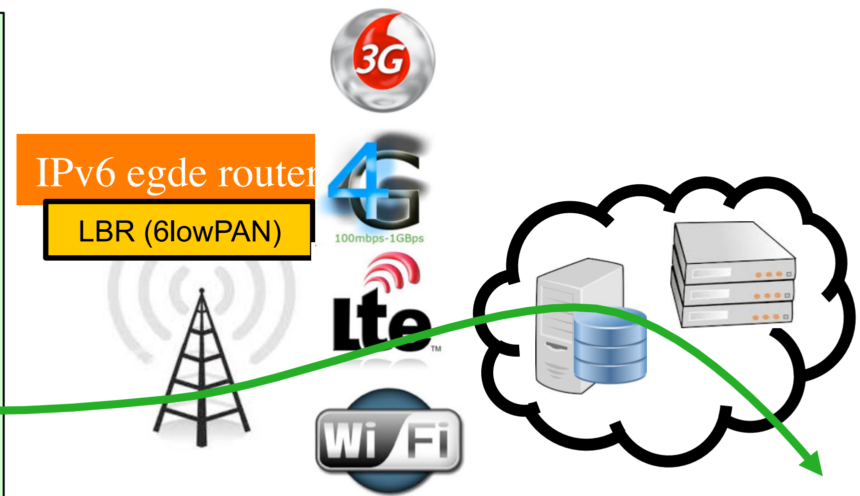


Propagate alerts to the sink

Send images to the sink

Multi-hop to Sink

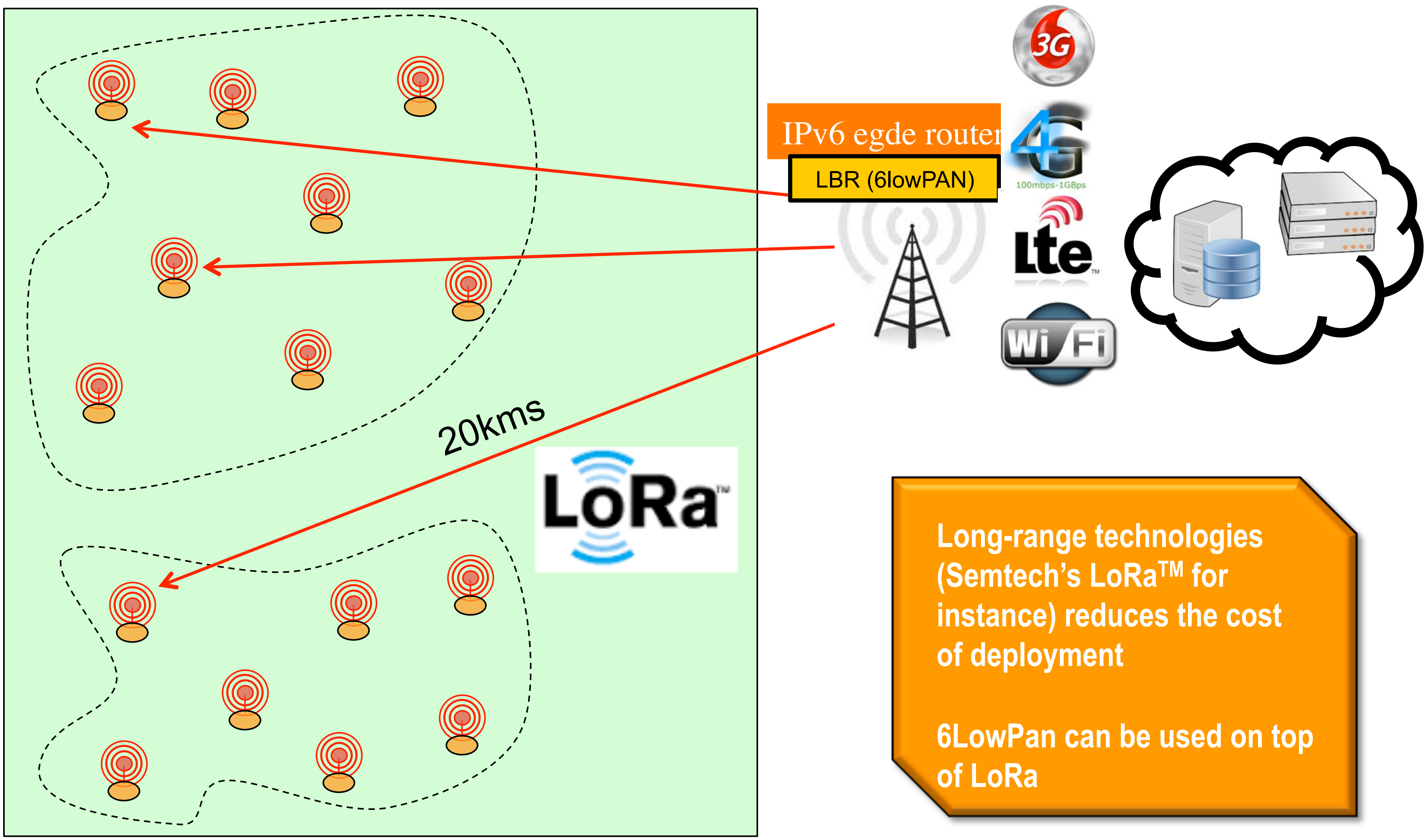




RPL is the 4th protocol standardized by IETF (RIP, OSPF, BGP)

Provides end-to-end IPv6 connectivity

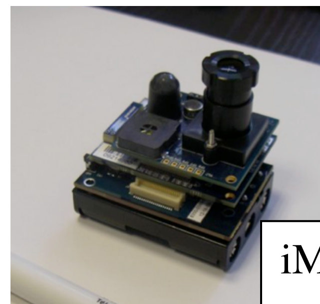
1-hop to Sink with long-range technology



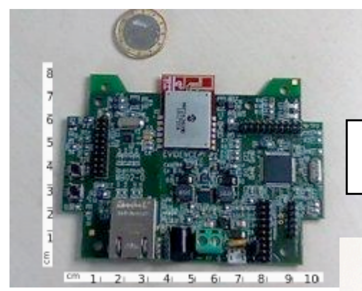
Long-range technologies (Semtech's LoRa™ for instance) reduces the cost of deployment

6LowPan can be used on top of LoRa

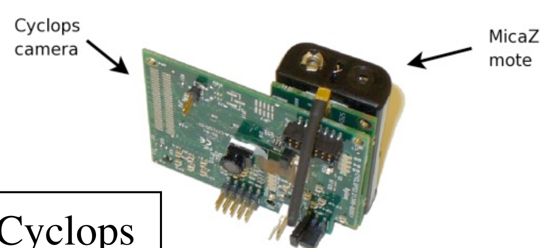
Wireless Image Sensors



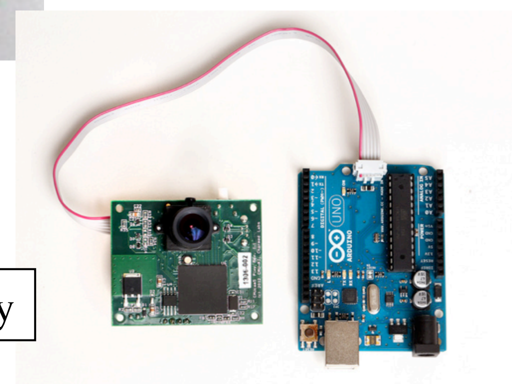
iMote2 with IMB400 multimedia board



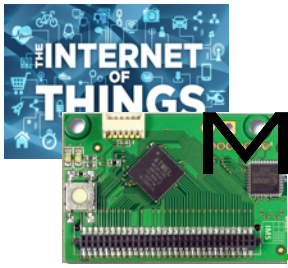
Seedeye



Cyclops



Pixy

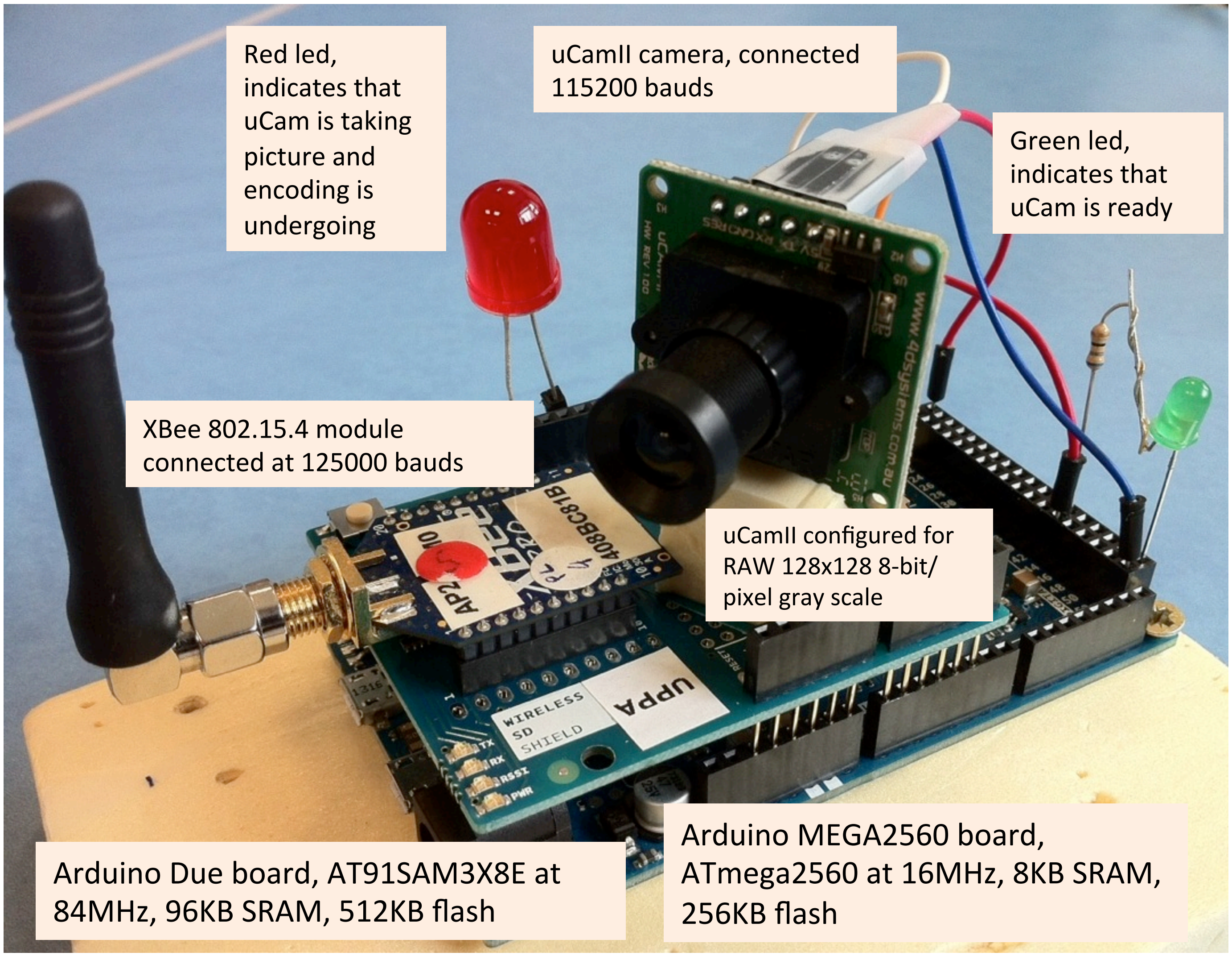


Motivations & objectives

- Off-the-shelf solution for maximum reproducibility
  - Arduino-based solution for maximum flexibility and simplicity in programming and design;
  - Simple, affordable external camera to get raw image data, no soldering
- Fast and efficient compression scheme with the host µC (no additional nor dedicated µC)
  - Small size image
  - packet loss-tolerant bit stream
- Out of the box-surveillance
  - Run on battery
  - Image change detection

C. Pham, Deploying a Pool of Long-Range Wireless Image Sensor with Shared Activity Time. WiMob 2015.

Our low-cost image sensor



Very low-memory platforms

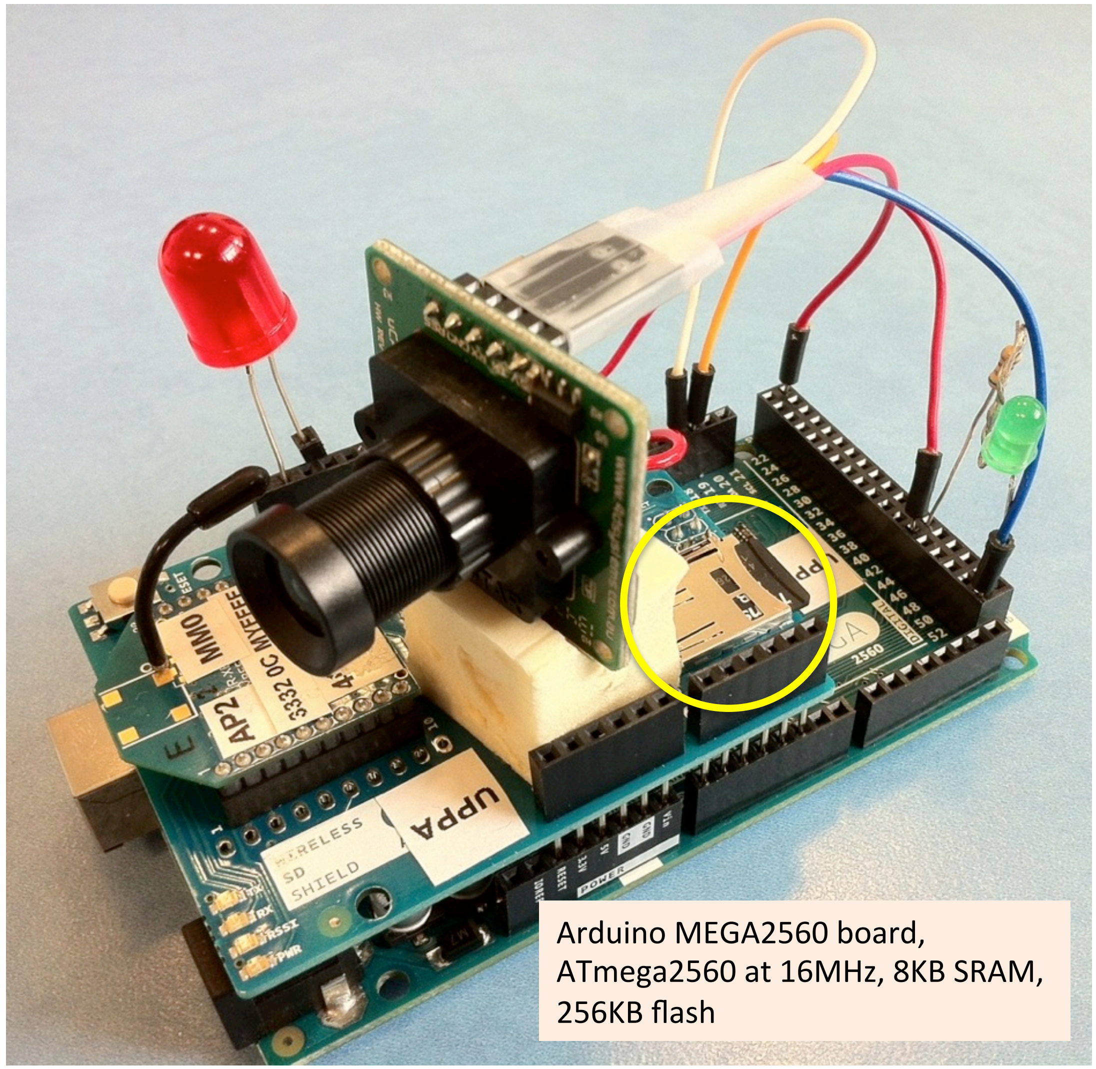
Arduino MEGA2560 at 16MHz, 8KB SRAM

Only 2KB SRAM available at runtime

Modified encoding algorithm to avoid having all the raw image in SRAM: encoding, packetization and transmission in a row per image packet

Reference image and current raw stored in an SD card

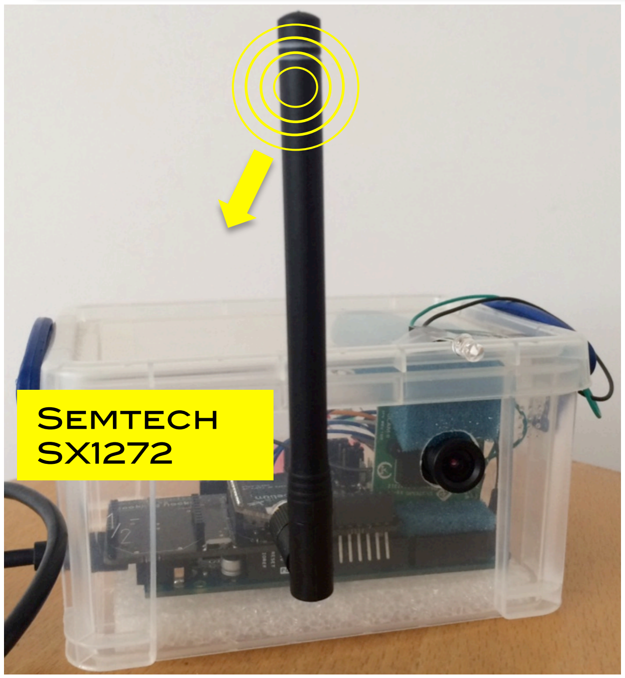
Encoding and packetization will read image blocks from SD card

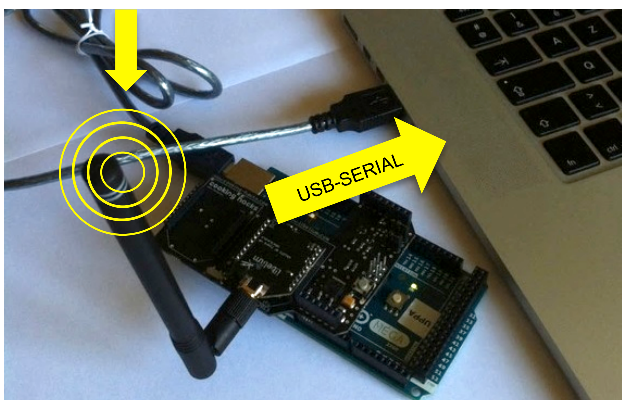


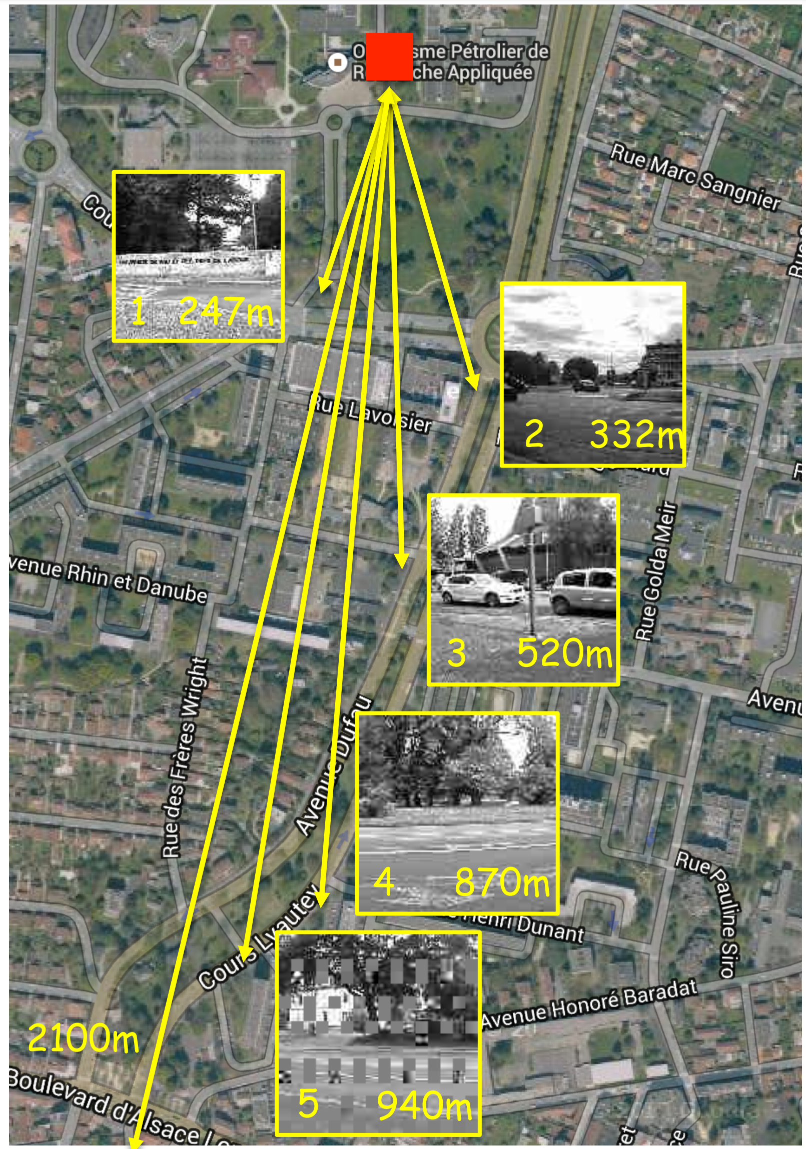
Arduino MEGA2560 board, ATmega2560 at 16MHz, 8KB SRAM, 256KB flash

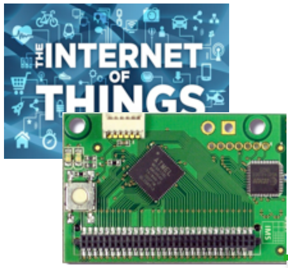
Long-range version

LoRa





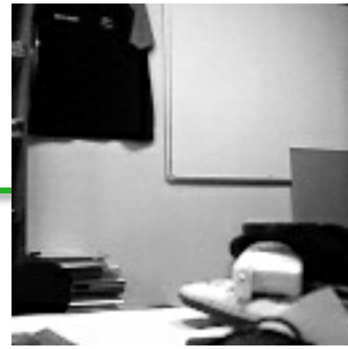




ADJUSTABLE IMAGE QUALITY FACTOR Q

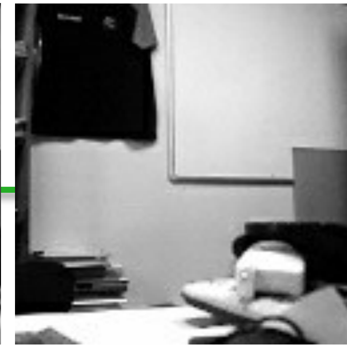
raw 16384b

Q=100; 9768b (1.67)



PSNR=51.344

Q=90; 5125b (3.2)



PSNR=29.414

Q=80; 3729b (4.4)

PSNR=28.866

Q=70; 2957b (5.5)

PSNR=28.477

Q=60; 2552b (6.4)

PSNR=28.024

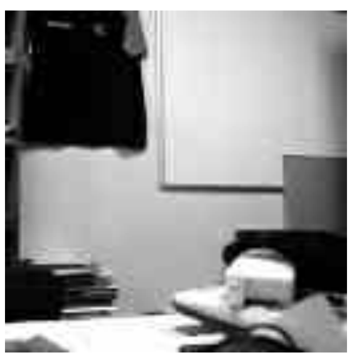
Q=50; 2265b (7.2)

PSNR=27.912

Q=40; 2024b (8.1)

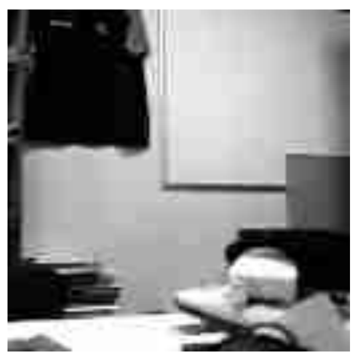
PSNR=27.423

Q=30; 1735b (9.5)



PSNR=26.933

Q=20; 1366b (12)



PSNR=26.038

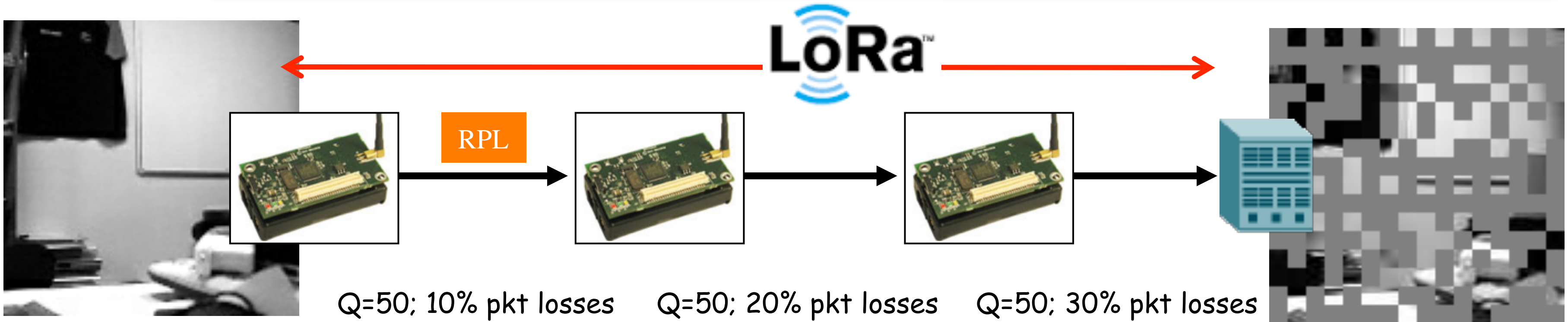
Q=10; 911b (18)

PSNR=25.283

Q=5; 576b (28.44)

PSNR=23.507

PACKET LOSS-TOLERANT BIT STREAM, ANY RECEPTION ORDER



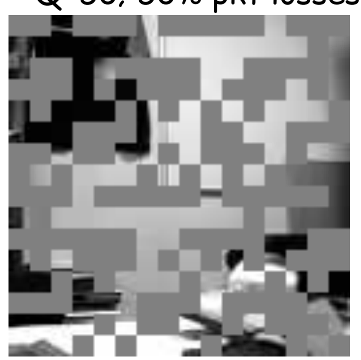
Q=50; 10% pkt losses

Q=50; 20% pkt losses

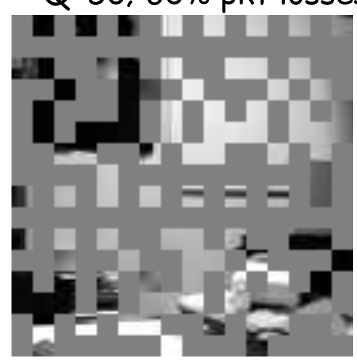
Q=50; 30% pkt losses

Q=50; 40% pkt losses

Q=50; 50% pkt losses

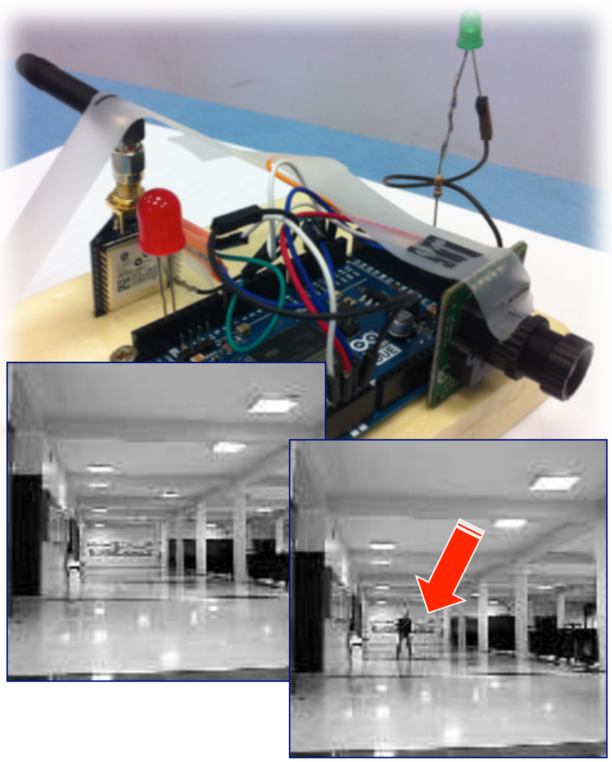


Q=50; 60% pkt losses

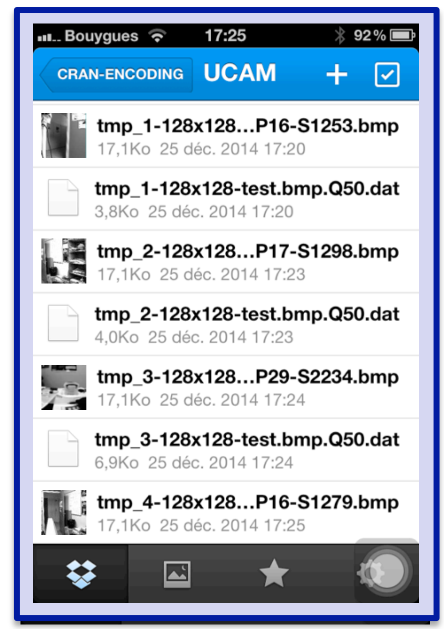


Scientific cooperation with V. Lecuire from CRAN laboratory for the optimized image encoding algorithm

Image change detection for intrusion detection

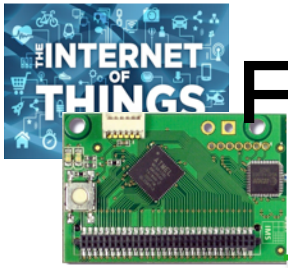


Sends image to gateway on intrusion detection



Very lightweight « simple-differencing » method, takes into account modification in image luminosity

Real-time synchronization with your smartphone through cloud applications, e.g. DropBox



Performance measures Arduino Due

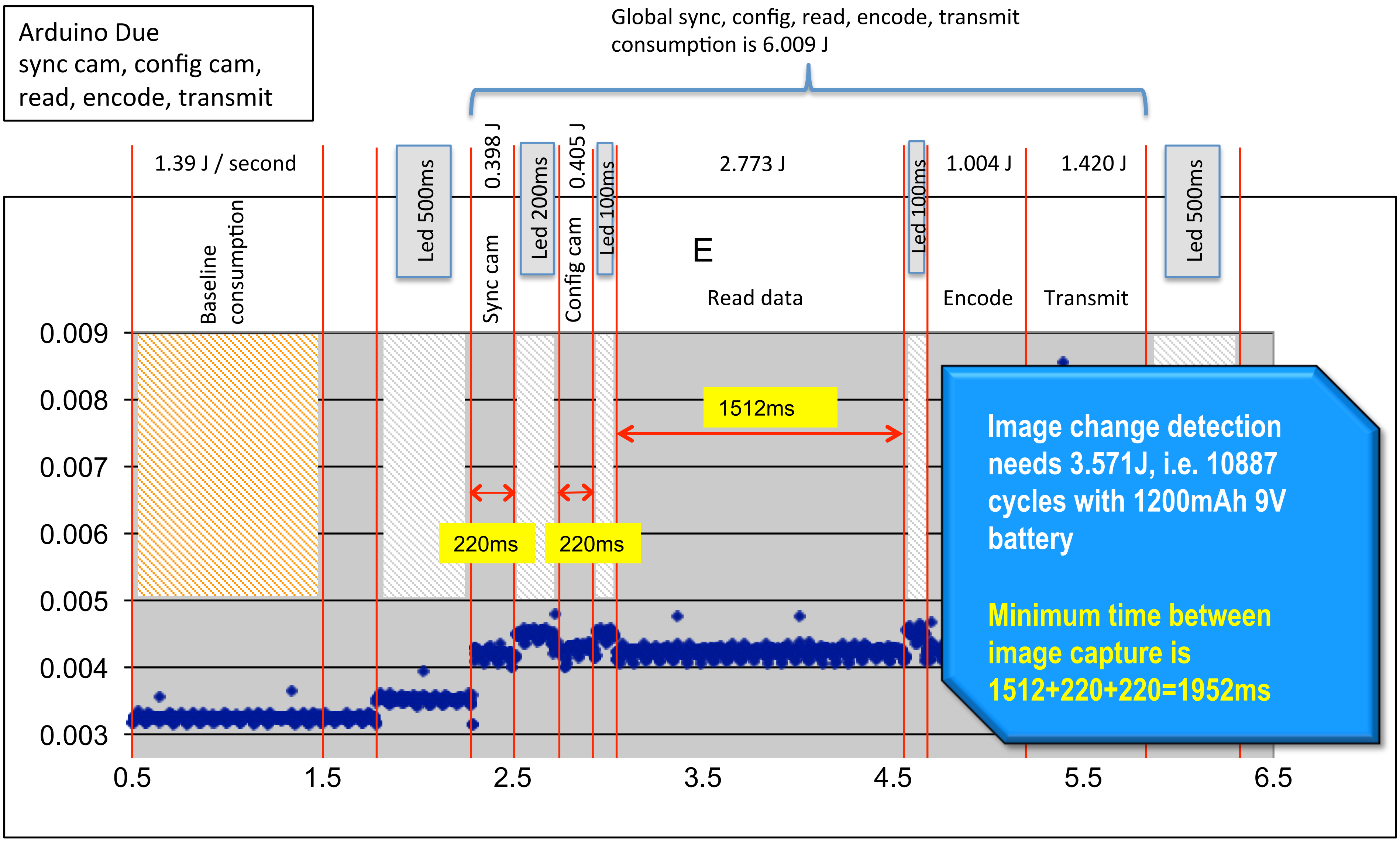
Arduino Due 84MHz

		N	R	A	B	C = D - B	D	E=R+D	F
Quality Factor Q	size in bytes (compression ratio)	number of packets (with MSS=90)	reading time from ucam	encode time	encode + pkt time	transmis-sion time (deduced)	encode + pkt + transmis-sion time	cycle time, with transmis-sion	rcv time at the sink
90	5125 (3.2)	70	1512	512	782	539	1321	2833	799
80	3729 (4.4)	48	1512	511	704	384	1088	2600	599
70	2957 (5.5)	37	1512	519	686	304	990	2502	447
60	2552 (6.4)	32	1512	509	662	263	925	2437	390
50	2265 (7.2)	28	1512	500	646	233	879	2391	349
40	2024 (8.1)	25	1512	516	657	207	864	2376	317
30	1735 (9.5)	21	1512	516	649	177	826	2338	278
20	1366 (12)	17	1512	518	638	140	778	2290	231
10	911 (18)	11	1512	516	628	93	721	2233	177

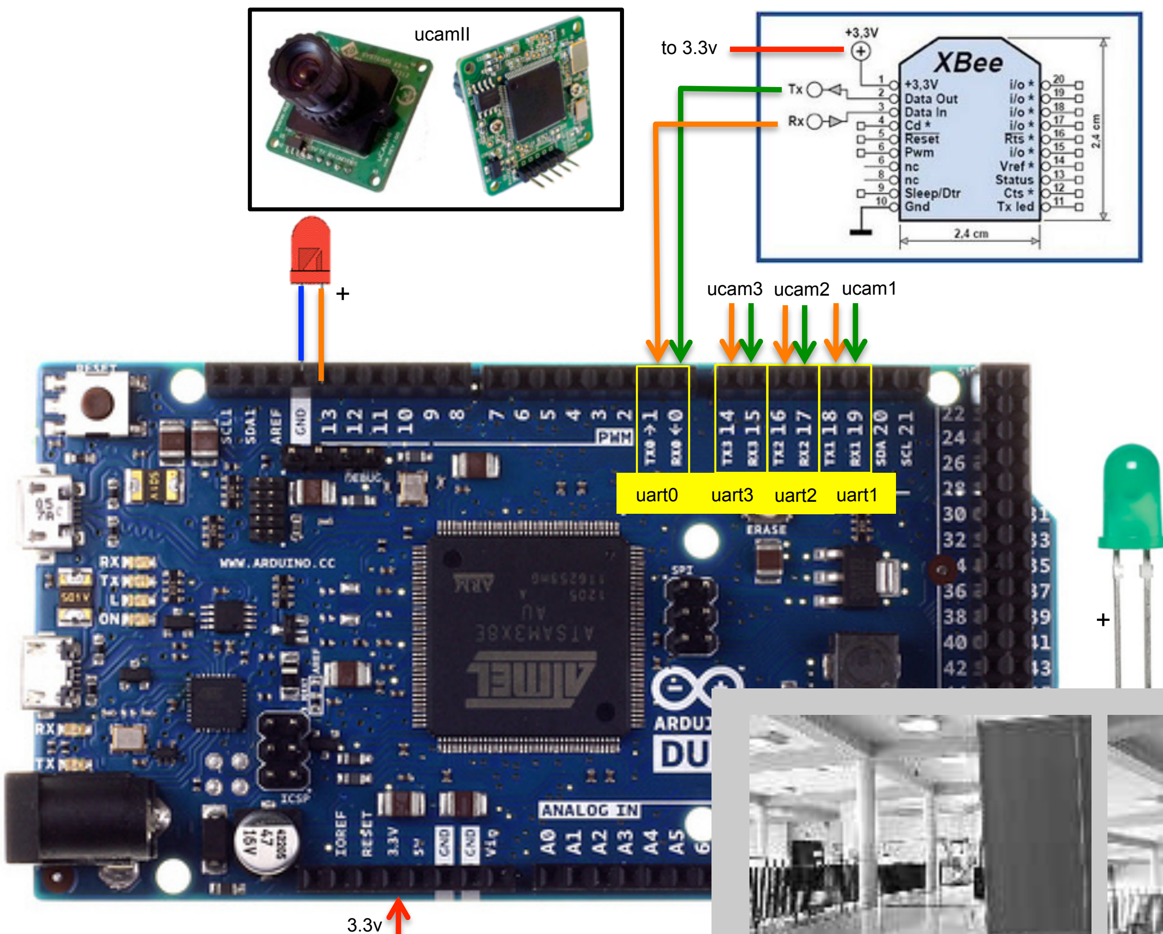
Includes image change detection

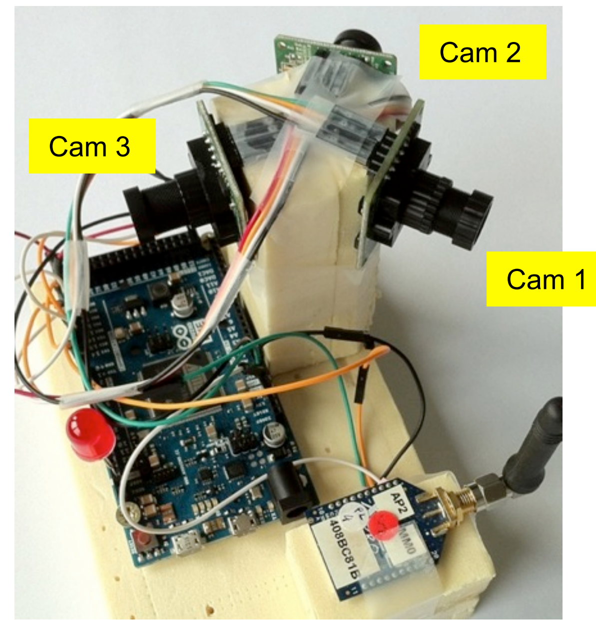
IEEE 802.15

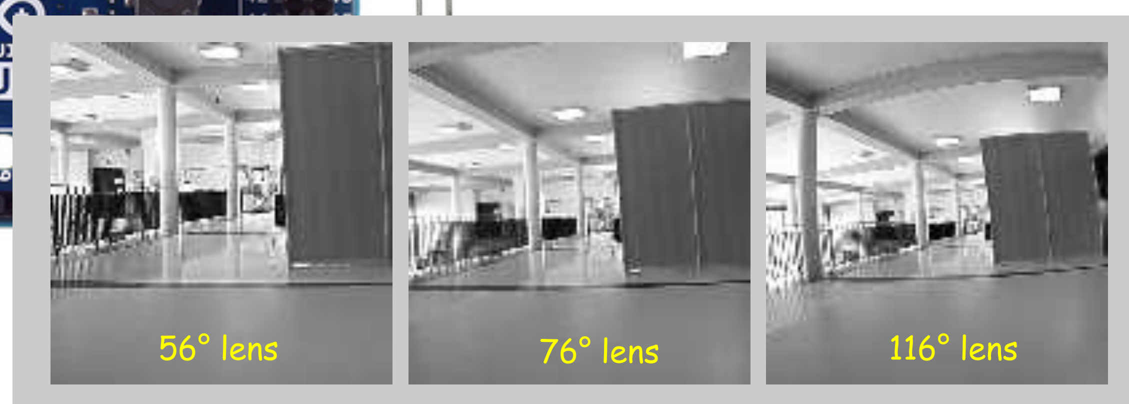
Energy consumption & timing



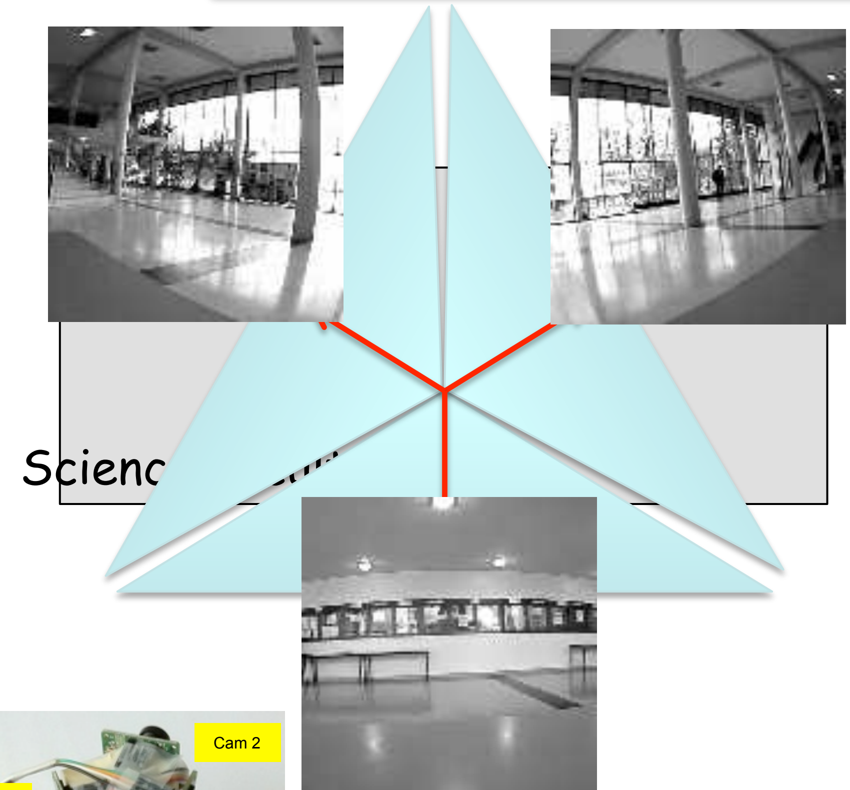
Multi-camera system

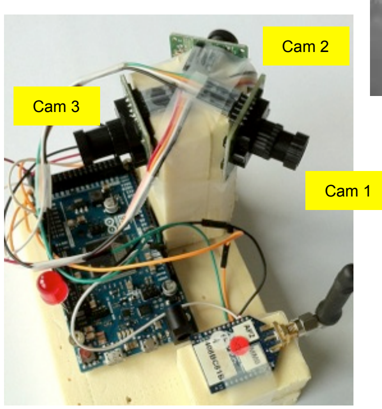






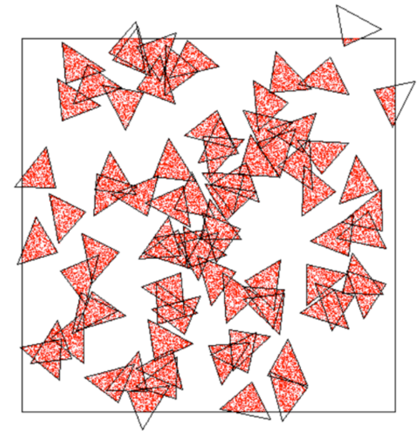
Low-cost omnidirectional visual sensing

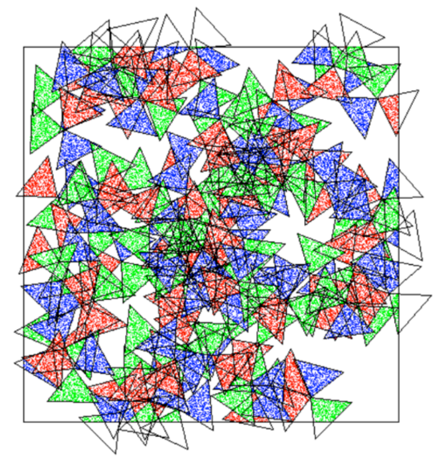




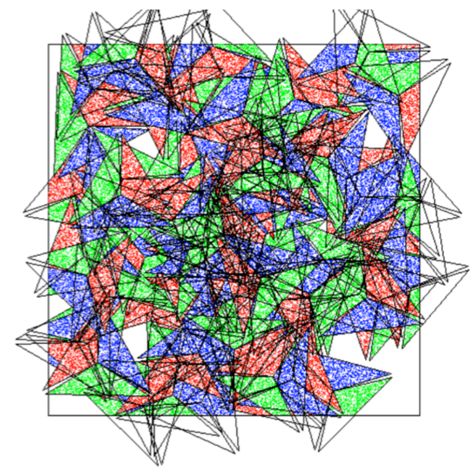
Cameras can be activated in weighted round-robin manner or randomly

80 image sensors,
1 camera/sensor aov=76°



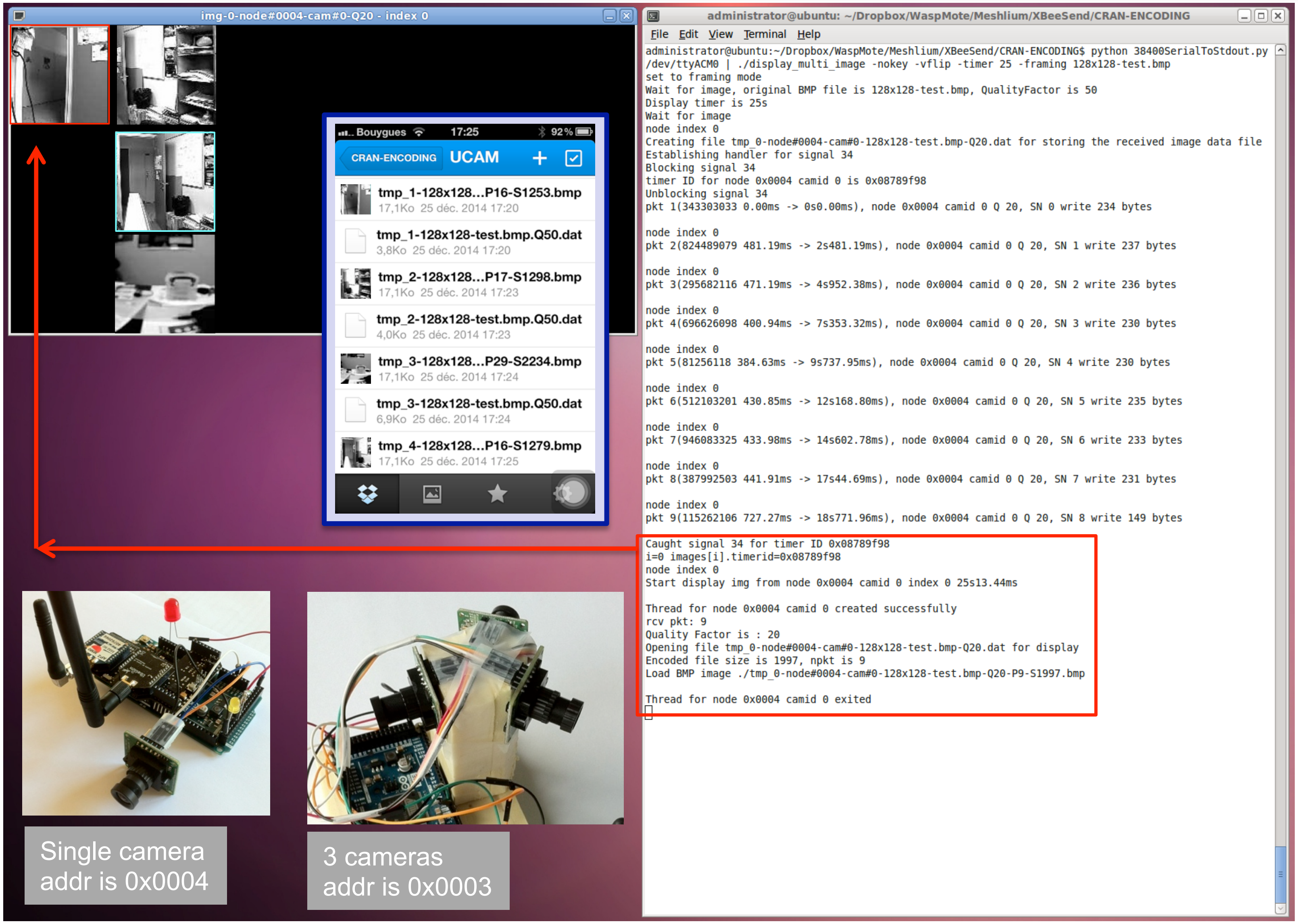


80 image sensors,
3 camera/sensor aov=76°



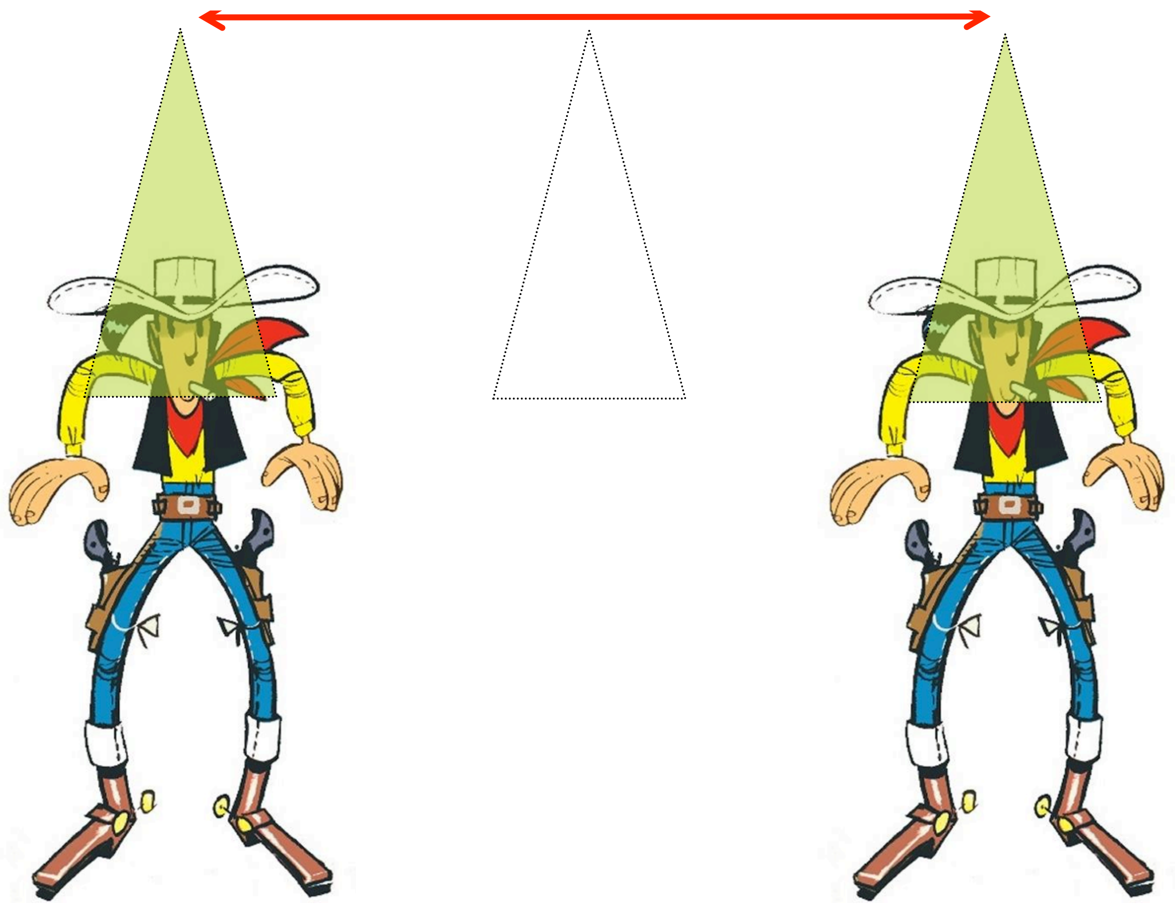
80 image sensors,
3 camera/sensor aov=116°

Out-of-the-box surveillance



Don't miss important events!

Time between 2 image capture



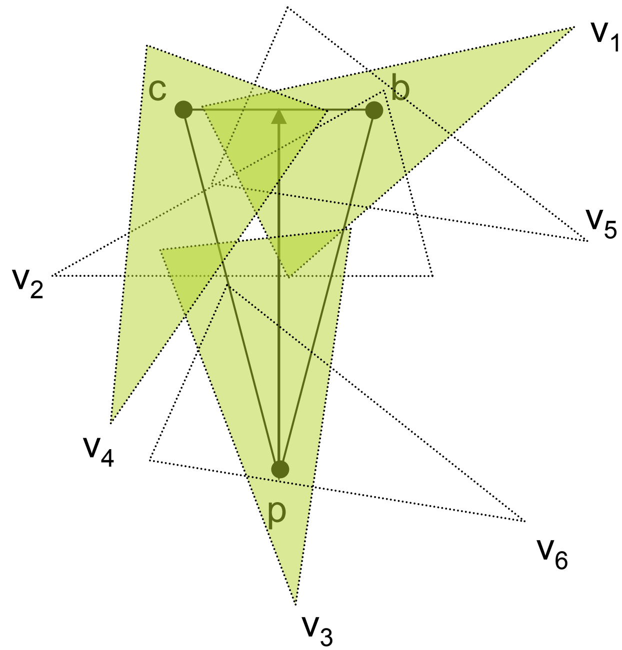
Whole understanding of the scene is wrong!!!

What is captured

Image sensor's cover set

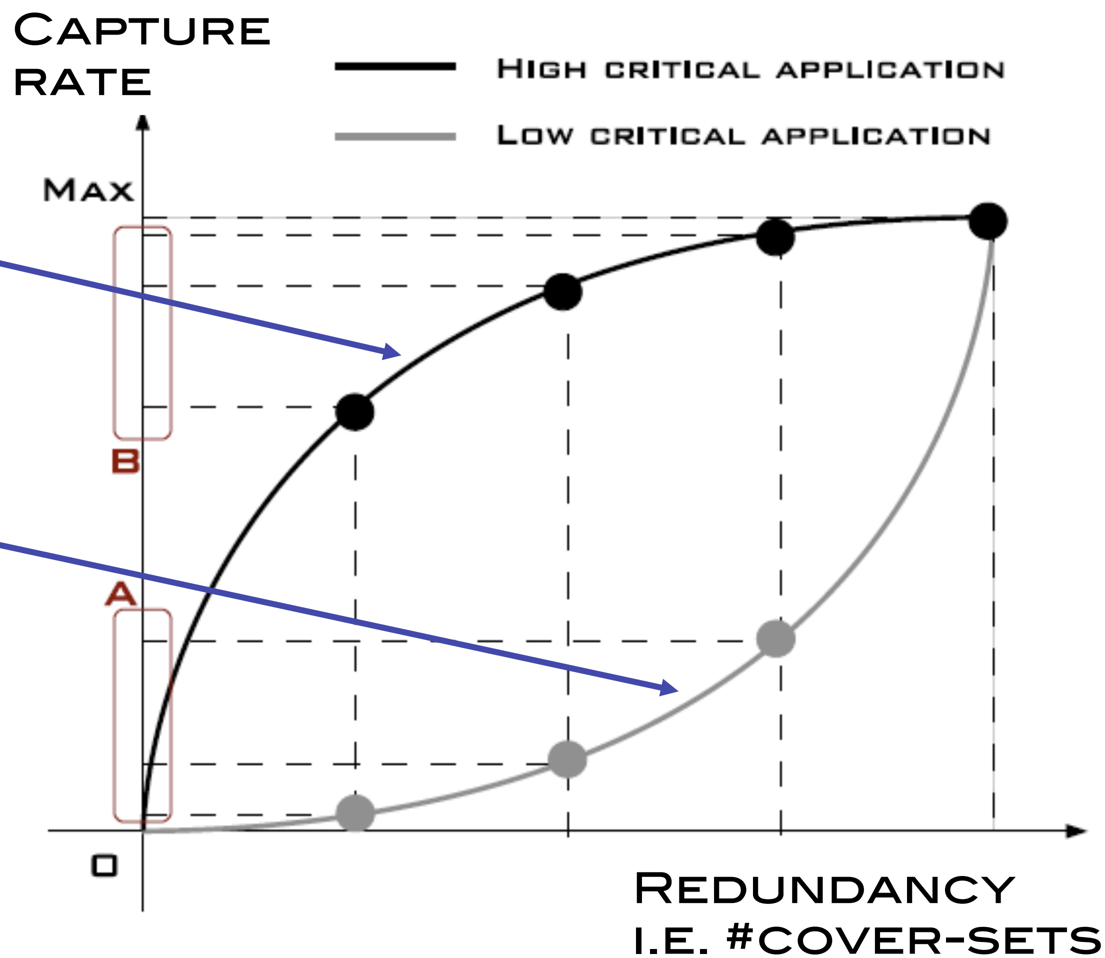
Co(V)= {
{V },
$\{V_1, V_3, V_4\}$,
$\{V_2, V_3, V_4\}$,
$\{V_3, V_4, V_5\}$,
$\{V_1, V_4, V_6\}$,
$\{V_2, V_4, V_6\}$,
$\{V_4, V_5, V_6\}$
}

|Co(V)| = 7



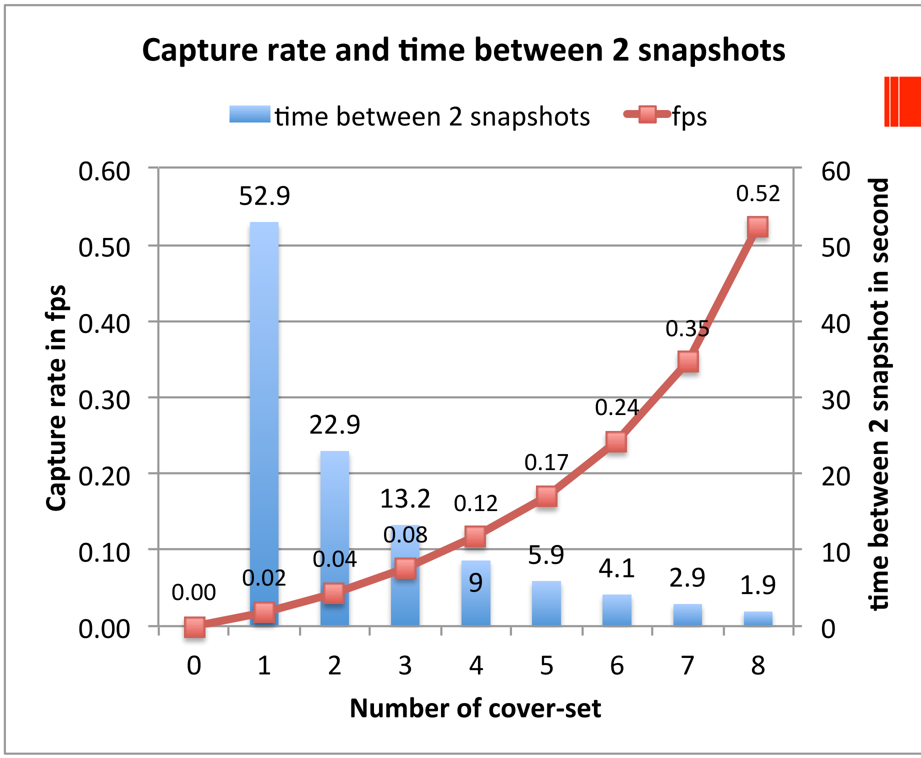
Schedule activity with criticality in mind

- Link the activity to redundancy level
- High criticality
  - Convex shape
  - Most projections of x are close to the max activity
- Low criticality
  - Concave shape
  - Most projections of x are close to the min activity
- Concave and convex shapes automatically define sentry nodes in the network

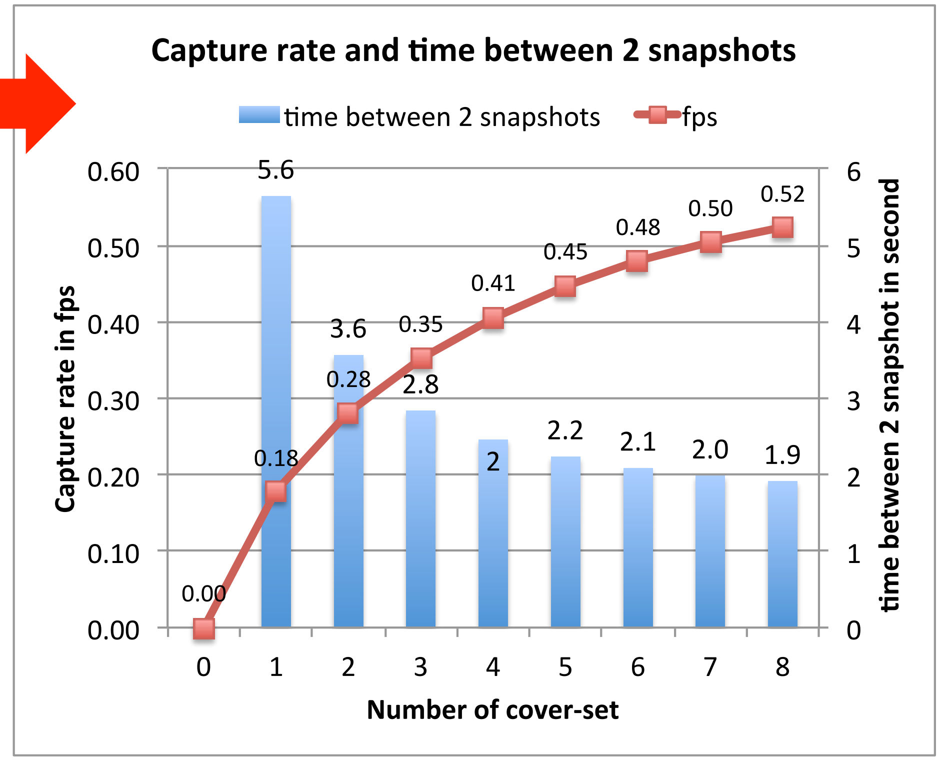


Capture rate of low-cost image sensor

Low criticality



High criticality



Is the maximum capture rate allowed by low-cost image sensor sufficient to provide low latency intrusion detection system?

Simulation parameters

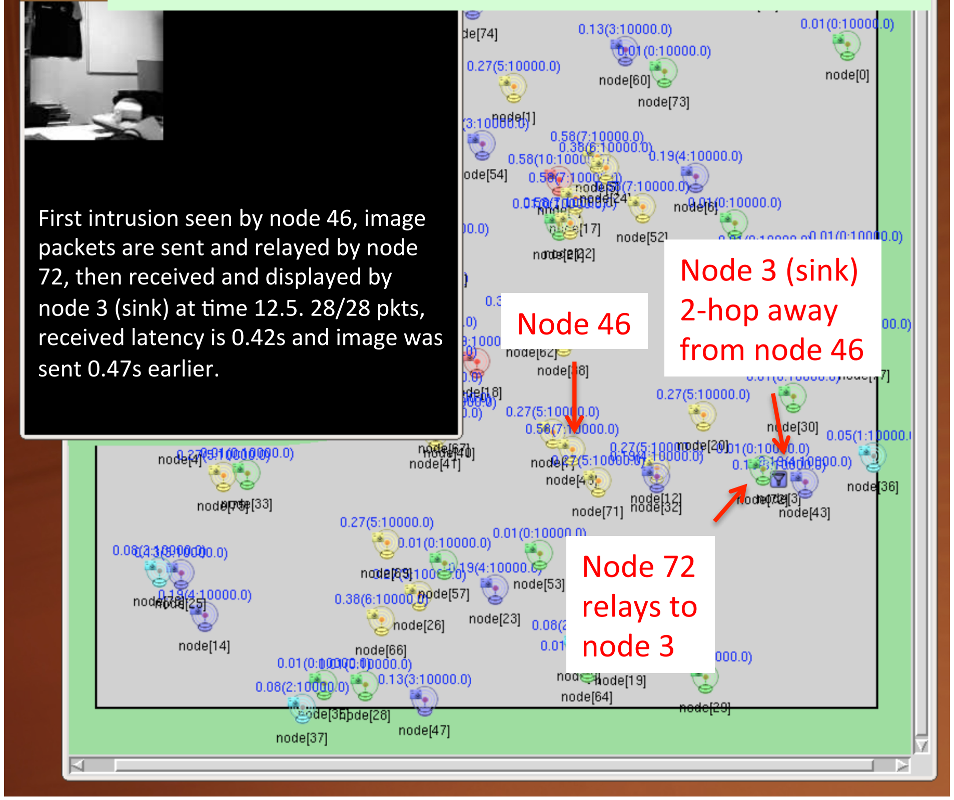
- 128x128 8bpp encoded image
- Quality Factor of 50: encoded image of 2265 bytes in 28 packets
- We need 220ms to configure the camera and 220ms for sync. Time before image data can be processed is 1.512s. Total is 1.952s
- So the maximum frame capture rate & image change detection is 0.52fps
- Camera angle of view could be 56°, 76° or 116°
- Depth of view of 25m
- Packet overhead at the image source is 11ms (8ms for transmission and 3ms for packetization)
- Packet relaying time is 16ms (based on measures of MicaZ relay node platform)

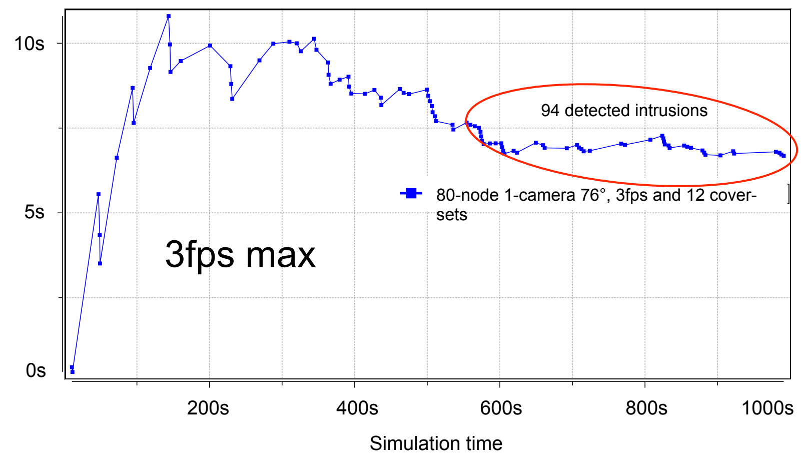
Q=50; 2265b

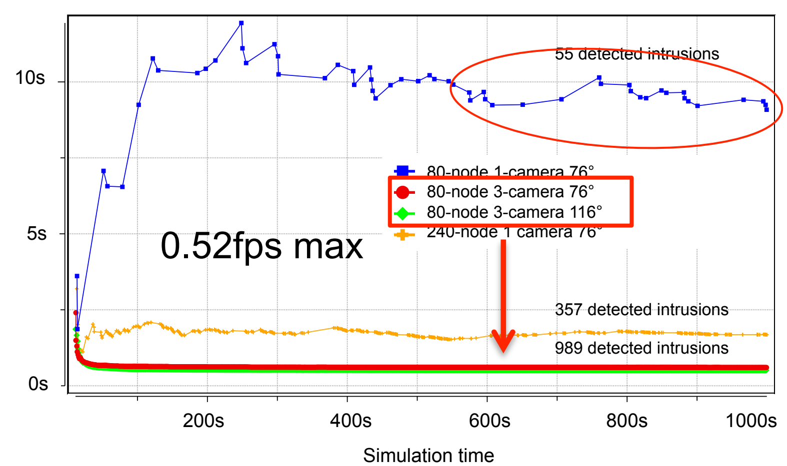
INTRUSION DETECTION LATENCY

The simulation model is used to study performance of large-scale intrusion detection system

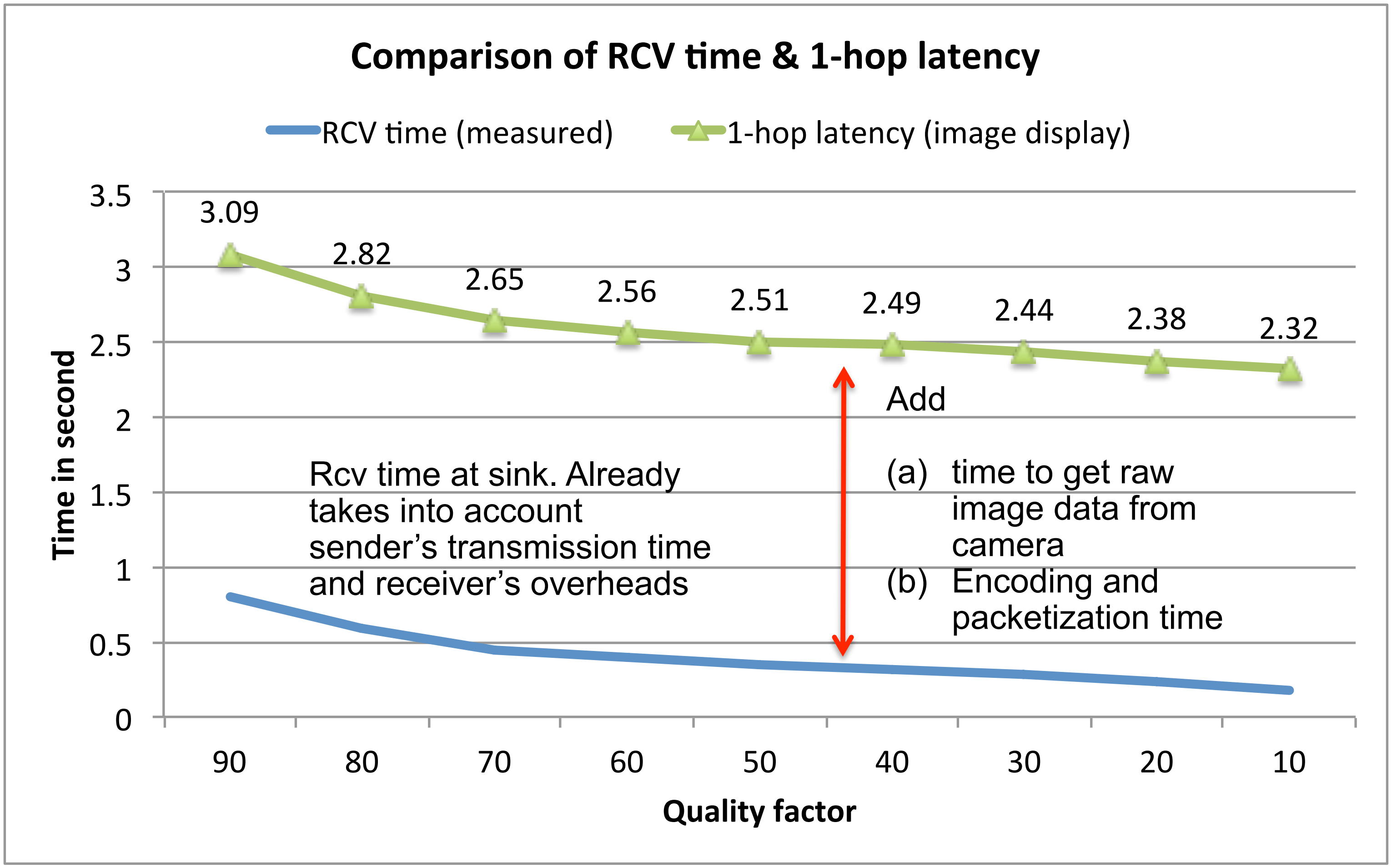
Using real measures for image processing tasks and packet transmission overheads produces very accurate simulation results that are consistent to what have been found in real multi-hop experimentations



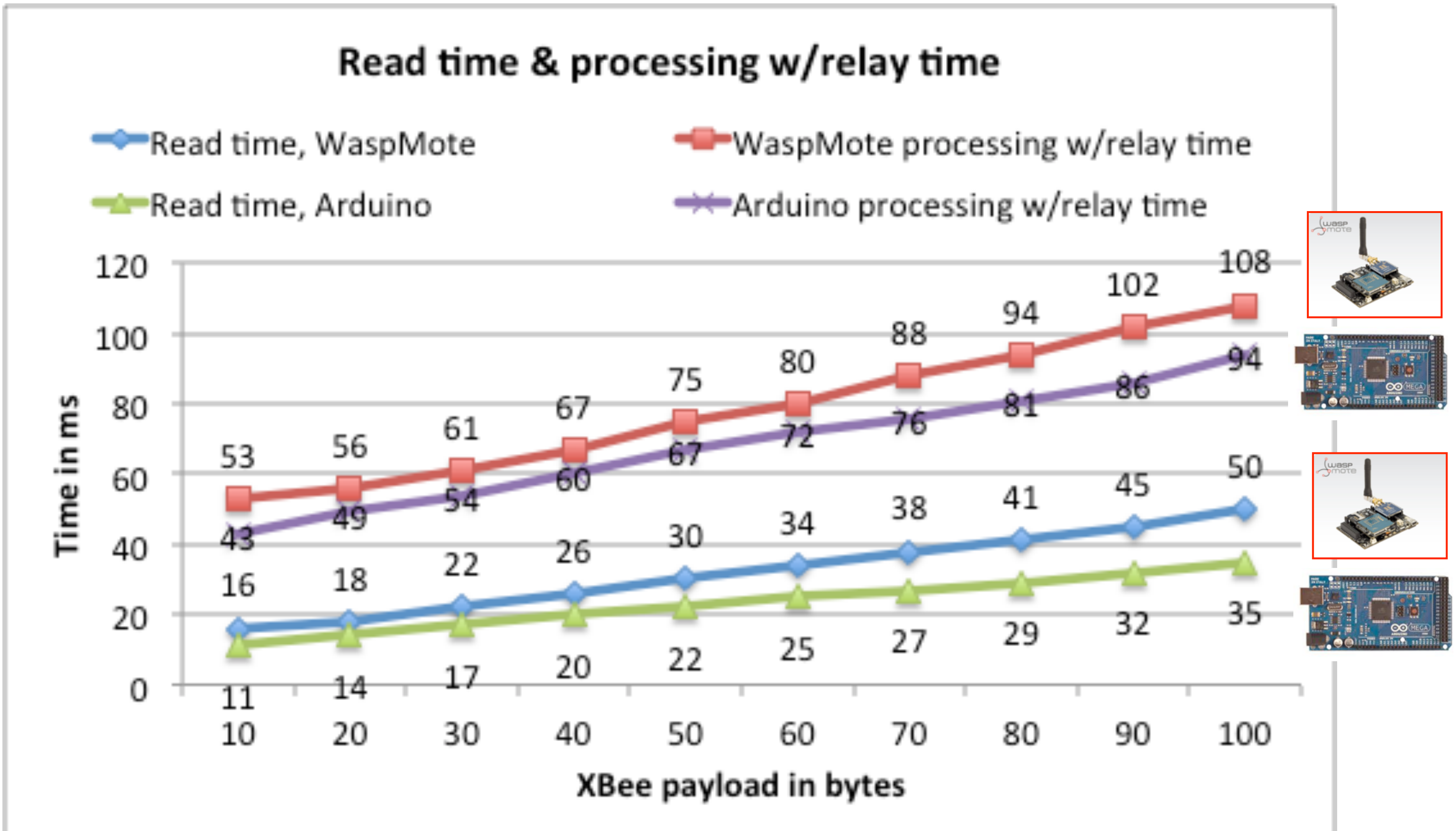




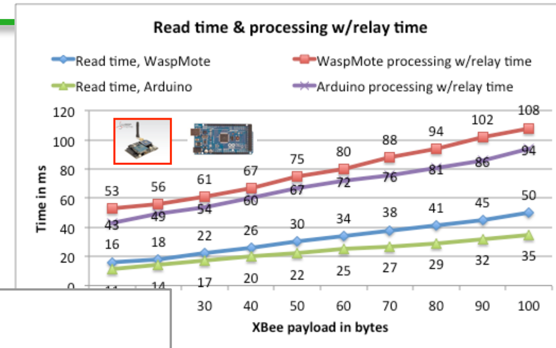
1-hop image latency

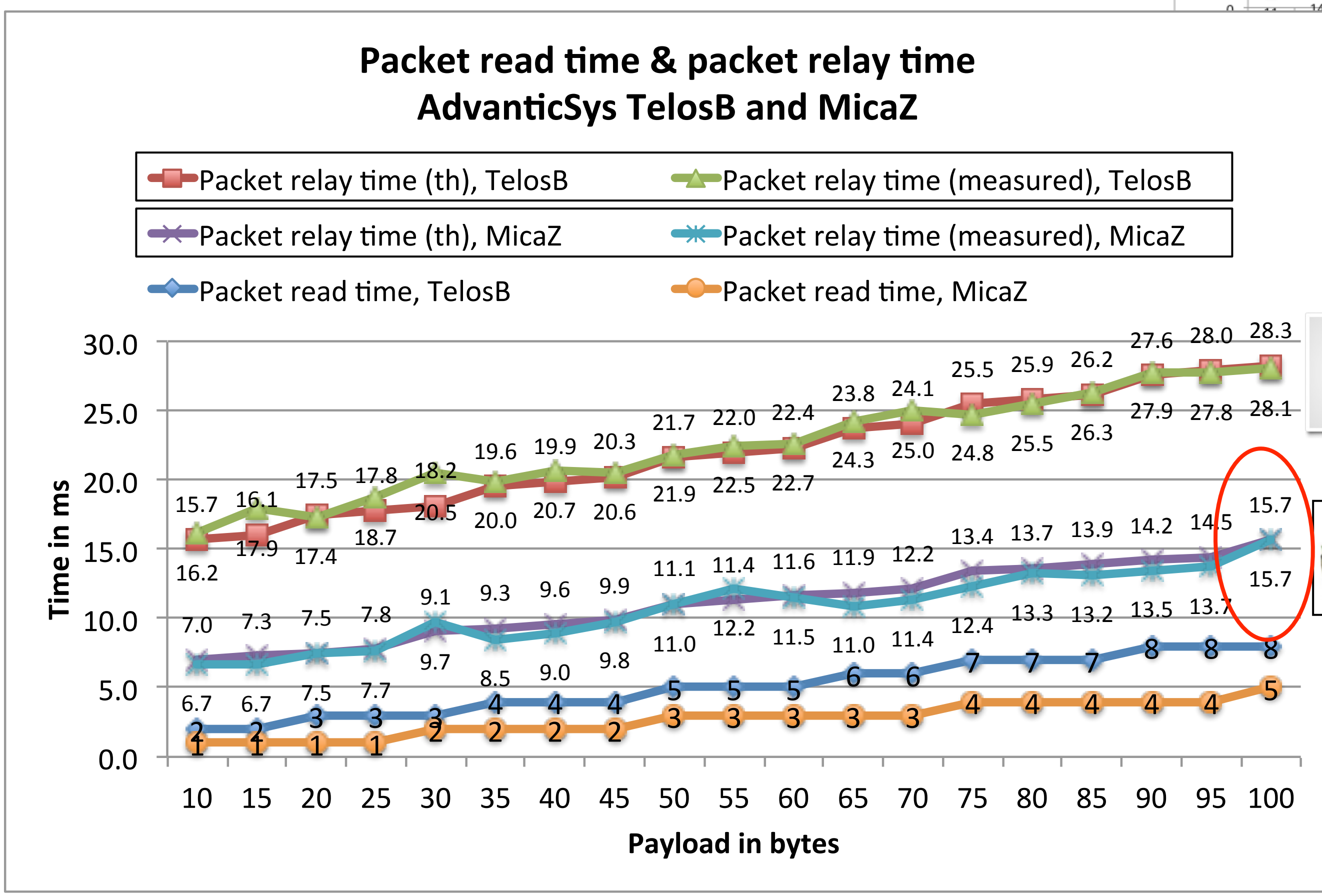


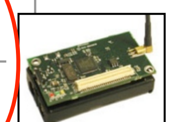
Multi-hop image latency



Multi-hop image latency







Add 16ms per relay node

Conclusions

- Low-cost image sensor from off-the-shelves components with fast and packet loss-tolerant encoding
- Can run out-of-the box to perform surveillance tasks based on image change detection
- Multi-camera systems can be easily built to increase coverage at low-cost
- Accurate large-scale simulations shows the efficiency of low-cost multi-camera system compared to faster single camera configurations

Do it yourself

An image sensor board based ...

cpham.perso.univ-pau.fr/WSN-MODEL/tool-html/imagesensor.html

An image sensor board based on Arduino Due/MEGA and uCamII camera

support for IEEE 802.15.4 and long-range LoRa™ radios

C. Pham, LIUPPA laboratory, University of Pau, France. http://cpham.perso.univ-pau.fr/

last update: May 12th, 2015.

Introduction

There are a number of image sensor boar … Panoptes, CMUcam3&FireFly, CMUcam4, CMUcam5/PIXY, iMote2/IMB400, ArduCam, ... All these platforms and/or products are very good and our motivations in building our own image sensor platform for research on image sensor surveillance applications are:

1. have an off-the-shelf solution so that anybody can reproduce the hardware and software
    1. use an Arduino-based solution for maximum flexibility and simplicity in programming and design
    2. use a simple, affordable external camera to get RAW image data, no JPEG
2. apply a fast and efficient compression scheme with the host microcontroller to produce robust and very small image data suitable for large scale surveillance or search&rescue/situation awareness applications
3. simple enough to demonstrate our criticality-based image sensor scheduling propositions
    1. see our paper : C. Pham, A. Makhoul, R. Saadi, "Risk-based Adaptive Scheduling in Randomly Deployed Video Sensor Networks for Critical Surveillance Applications", *Journal of Network and Computer Applications (JNCA), Elsevier, 34(2), 2011, pp. 783-795*
4. easy integration with our test-bed for studying data-intensive transmission with low-resource mote platforms (audio and image)
    1. see our paper: C. Pham, "Communication performances of IEEE 802.15.4 wireless sensor motes for data-intensive applications: a comparison of WaspMote, Arduino MEGA, TelosB, MicaZ and iMote2 for image surveillance", *Journal of Network and Computer Applications (JNCA), Elsevier, Vol. 46, Nov. 2014*
5. fully similar/compatible with our simulation environment based on OMNET++/Castalia for video/image sensor networks.

Architecture and components

We use both Arduino Due and MEGA2560. The Arduino Due board has enough SRAM memory (96KB) to store an 128x128 8-bit/pixel RAW image (16384 bytes). On the MEGA2560, which has only 8KB of SRAM memory, we store the captured image on an SD card (see right figure below for an exemple) and then perform the encoding process by incrementally reading small portions of the image file.

For the camera, we use the uCamII from 4D systems. You can download the reference manual from 4D system web site. The uCamII can deliver 128x128 raw image data. JPEG compression can be realized by the embedded micro-controller but this feature is not used as JPEG compression is not suitable at all for lossy environments. We instead apply a fast and efficient compression scheme with the host microcontroller to produce robust and very small image data.

http://cpham.perso.univ-pau.fr/WSN-MODEL/tool-html/imagesensor.html

Question?

Congduc.Pham@univ-pau.fr