NITOS BikesNet: Enabling Mobile Sensing Experiments through the OMF Framework in a city-wide environment

Presenting: Giannis Kazdaridis
e-mail: iokazdarid@gmail.com

Donatos Stavropoulos, Vasilis Maglogiannis,

Thanasis Korakis, Spyros Lalis and Leandros Tassiulas

Date: 15/07/2014
Conference: IEEE Mobile Data Management, Brisbane, Australia

CERTH

NITLAB
NETWORK IMPLEMENTATION TESTBED LABORATORY

UNIVERSITY OF THESSALY

Unprecedented growth of Mobile Devices

- ✓ The latest mobile devices such as:
  - ✓ smart watches,
  - ✓ tablets
  - ✓ smart-phones, etc.

have become increasingly
  - ✓ sophisticated
  - ✓ and miniaturized

due to recent technological advances.

Google's Glasses

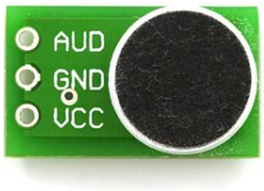
Smart-watch

Smart-phone

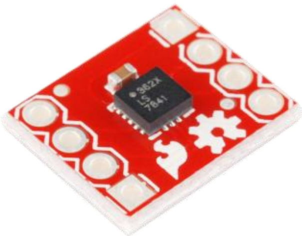
tablet

Embedded sensing modules

- The aforementioned mobile devices feature a vast number of embedded sensing modules:
  - GPS
  - Accelerometer
  - Gyroscope
  - Environmental sensors
  - Camera
  - Microphone



Microphone


Accelerometer

- All these sensors can be used to sense/capture different parameters.

Camera module

GPS sensor

Environmental sensor

Participatory Sensing

- ✓ These devices are carried by people in their every day activities.
- ✓ They can record a wealth of data that can be useful for the society.

- ✓ That is the concept of the so-called **participatory-sensing**:
  - ✓ Individuals gather and publish information without even knowing each other.

Source: http://clout-project.eu/

UNIVERSITY OF THESSALY

Our Work - BikesNet

- In our work we developed a **city-scale mobile sensing infrastructure** that relies **on bicycles** of volunteer users.
- We equip the bicycles with **custom-made, open-source devices** capable of capturing several parameters, such as:
  - available WiFi networks,
  - temperature & humidity,
  - light intensity and the,
  - exact location and time of each measurement.
- Experimenters can **remotely control** the operation of each sensor node as well as to **collect/visualize** measurements through the OMF/OML in a Delay Tolerant fashion.

Requirements

- Experimentation Capability:
  - User should be able to **plan, execute and control an experiment.**

- Remote Configuration:
  - User should be able to configure mobile sensor node **remotely**, without requiring physical access.

- Disconnected Operation:
  - We need to take under consideration that the sensor node mounted on the bicycle is unlikely to have constant or reliable network connectivity.

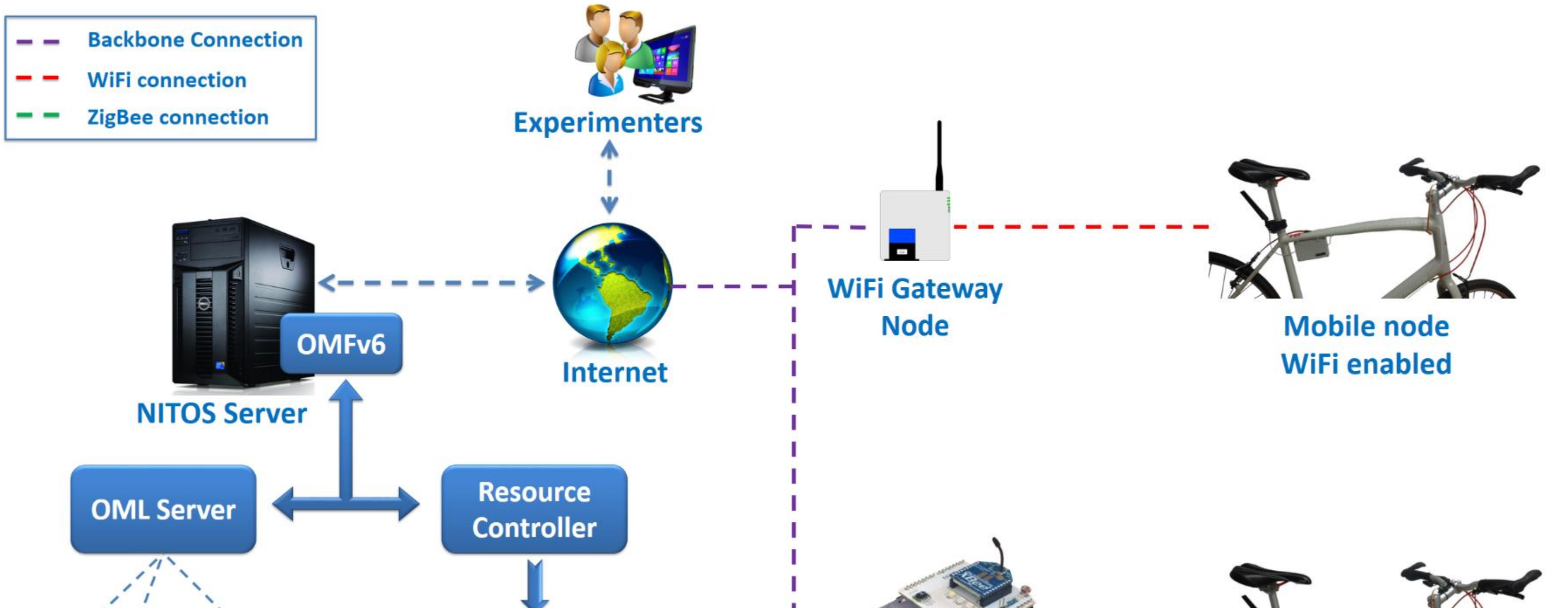
- Support for different networking technologies:
  - Develop a general approach with **flexibility in terms of networking**.

Requirements

- Low-cost:
  - Low-cost sensor enables **large-scale deployments**.
- Low-power:
  - The mobile sensor node should be able to **operate for a long period of time on batteries**, without requiring everyday charging.
- Small-size:
  - The mobile sensor node should be small in order to be easily mounted on the bicycle **without blocking the cyclists' moves**.
- Extensibility:
  - The sensor node should be **modular**, making it possible to **add new sensors** that are not available in the standard package.

NITOS BikesNet Architecture

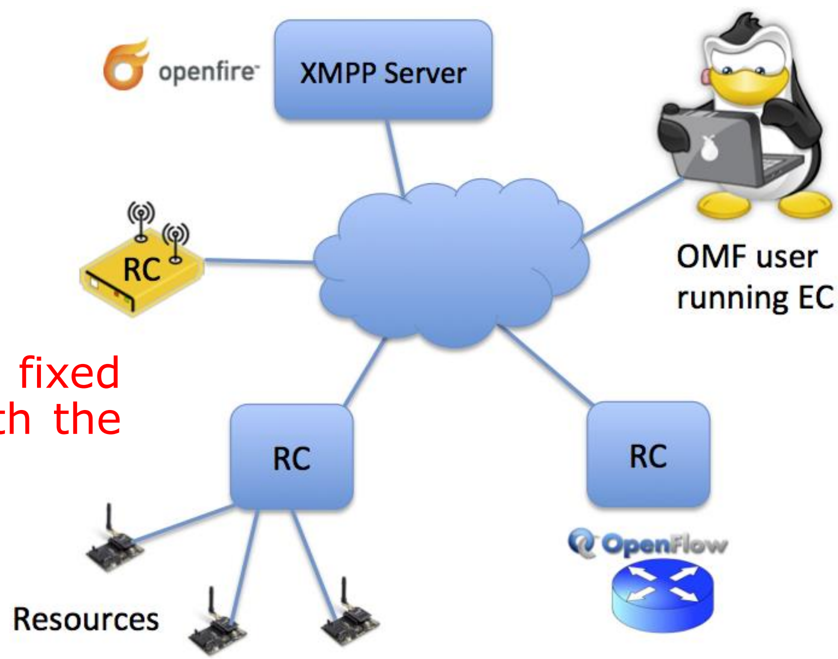
- Three main components are used in NITOS BikesNet:
  - Mobile Sensing Devices.
  - Gateway nodes acting.
  - Resource Controller running on the NITOS Server.



Node id	TCP connection
1	Connected/Disconnected
2	Connected/Disconnected
⋮	
N	Connected/Disconnected

OMF/OML Framework

- ✓ **cOntrol and Management Framework (OMF)** is a control and management open-source tool that supports ease of use and reproducibility of experimentation in testbeds.
- ✓ **OML** is used for the instrumentation and measurements collection.
- ✓ The large-scale testbeds world-wide have adopted the OMF.



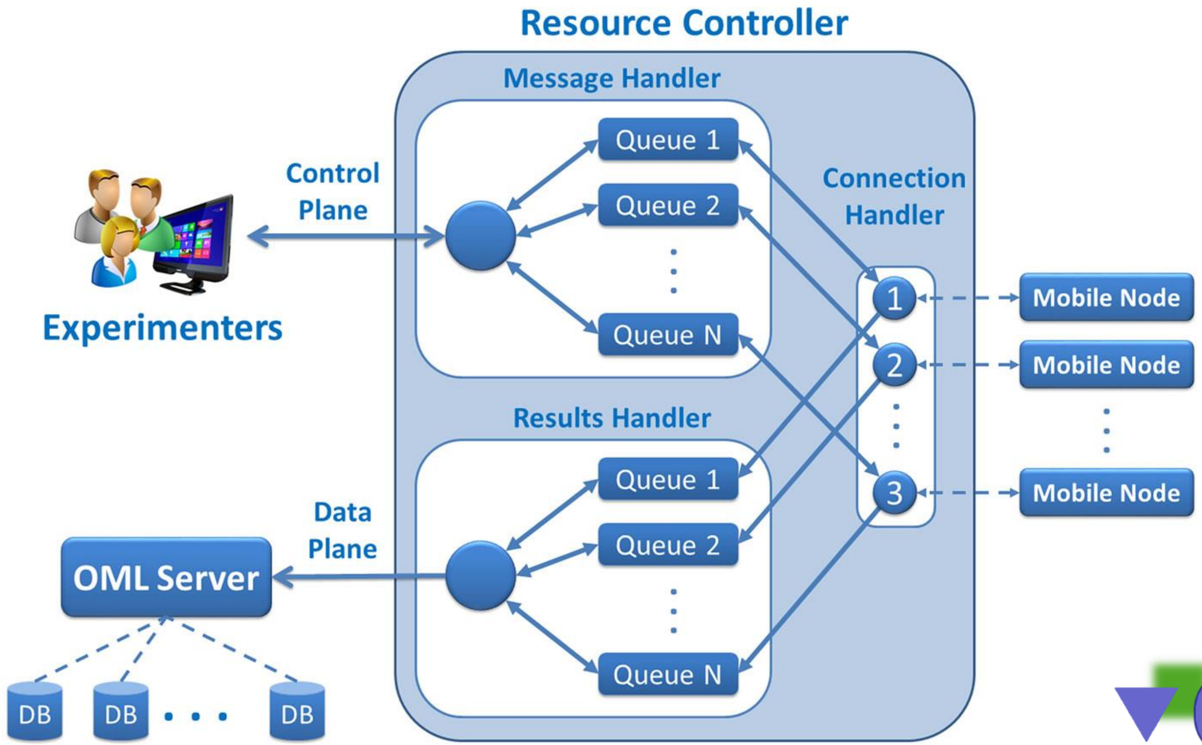
OMF assumes fixed connection with the resources.

NITOS testbed - UTH

ORBIT testbed - Rutgers

Enhancements in the OMF

- ✓ We enhanced OMF to support NITOS BikesNet in a Delay Tolerant fashion by developing a new Resource Controller (RC).
- ✓ RC manages all the messages/results coming from experimenters/bikes.



Resource Controller Architecture

Messaging Protocol

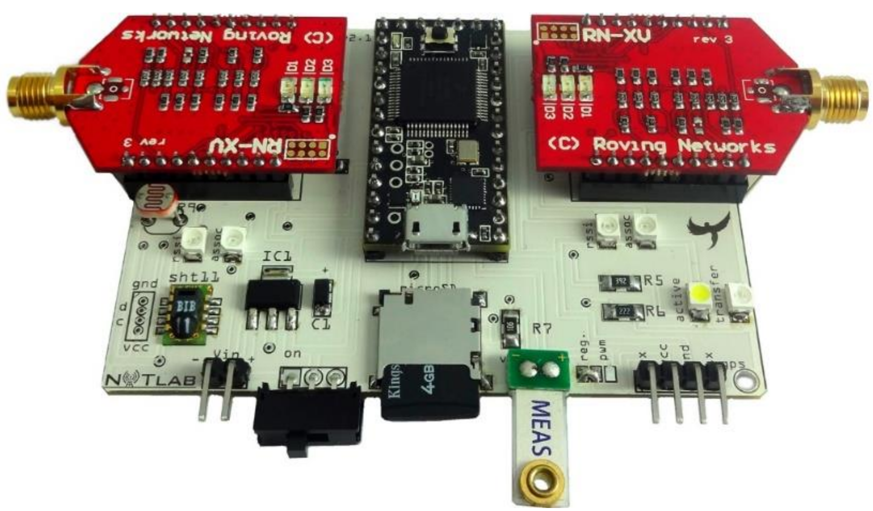
✓ We defined four basic messages to enable communication between the node and the RC.

Direction :: Message	**Comments**
N → RC :: <node id>:HELLO	Node announces its availability. The RC should update the node status internally. Initiate transmission of pending commands from the RC to the node, as well as transmission of pending results from the node to the RC.
RC → N :: <node id>:CMD:<seq. #>:<payload> N → RC :: <node id>:CMDACK:<seq. #>	RC sends the next pending command, and waits for an acknowledgment. Can be repeated several times.
N → RC :: <node id>:RSLT:<seq. #>:<payload> RC → N :: <node id>:RSLTACK:<seq. #>	Node sends next batch of results to the RC, and waits for acknowledgment. Can be repeated several times.
N → RC :: <node id>:BYE	Node informs that it will sign off and become unavailable. The RC should update the node status internally.

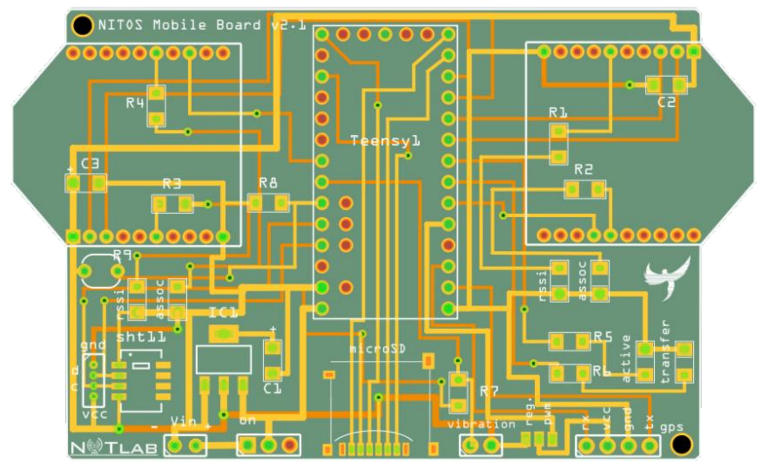
Messaging Protocol for the communication between the RC and the nodes

Mobile Device

- ✓ The mobile device consists of **open-source** and **configurable** modules properly connected through the fabricated PCB.



Mobile Sensing Device

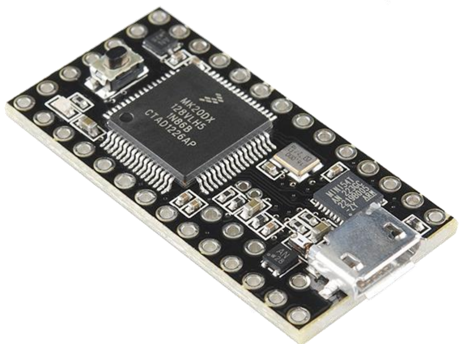


PCB Layout

PCB designed with Fritzing Software

Microprocessor board

- ✓ The core module is the Teensy 3.0 board.



Teensy 3.0 microprocessor board

ARDUINO OPEN-SOURCE COMMUNITY

Arduino IDE

- ✓ 32-bit ARM Cortex-M4 micro-processor.
- ✓ 128KB of Flash memory (program space) and 2KB of EEPROM memory (long-term).
- ✓ Features 34 digital I/O pins, one SPI and 3 UART communication ports.
- ✓ Can be configured to run at 96MHz, 48MHz, 24MHz or 2MHz.
- ✓ Operates at 3.3v.
- ✓ Supports sleep mode to minimize power consumption.
- ✓ Arduino compatible:
  - ✓ Ease of firmware development (high-level language).
  - ✓ Several libraries provided (communication with peripherals).

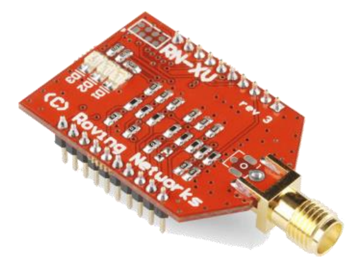
Wireless Interfaces

- ✓ The mobile node features **two sockets where communication modules** can be plugged, in order to provide with sufficient flexibility in terms of networking.

- ✓ Xbee Series 2 (Zigbee compatible):
  - ✓ Implements the **ZigBee** protocol on top of **802.15.4** in the ISM 2.4GHz.
  - ✓ Operates at 3.3V.
  - ✓ Communicates with the Teensy over UART.
  - ✓ Supports sleep mode.
  - ✓ Provide a “HIGH” signal whenever associated.

Xbee Series2 Interface

- ✓ WiFly (WiFi compatible):
  - ✓ Implements the **802.11 b/g** radio protocol in the ISM 2.4GHz.
  - ✓ Operates at 3.3V.
  - ✓ Communicates with the Teensy over UART.
  - ✓ Supports sleep mode.
  - ✓ Provide a “HIGH” signal whenever associated.
  - ✓ Used to **scan for available WiFi networks**.



WiFly Interface

Sensing Modules

- GPS D2523T:
  - Provides with the exact **location** of the node.
  - Also with the exact **time/date** information.
  - Operates at 3.3V.
  - Communicates with the Teensy over **UART**.
  - Arduino compatible – **TinyGPS** library.
  - Teensy controls its operational state (on/off) through a voltage regulator.

GPS D2523T

- Sht11 temperature & humidity:
  - Provides with **temperature & humidity** measurements.
  - Operates at 3.3V.
  - Communicates with the Teensy over a **2-wire port**.
  - Arduino compatible – **Sensirion** library.

Environmental sensor

- Photo-cell sensor:
  - Provides with **luminosity** measurements.
  - Communicates with the Teensy over an **analog** pin.

Photo-Resistor

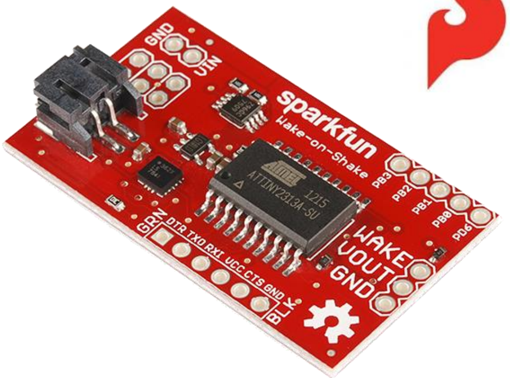
Rest Peripherals

- <u>MicroSD slot/card:</u>
  - It is used to **persistently log sensor measurements**, until they are uploaded to the RC.
  - This way the node provides **long-term** data logging.
  - The node can be put in sleep mode or be completely turned-off without any data loss.

microSD slot

Card

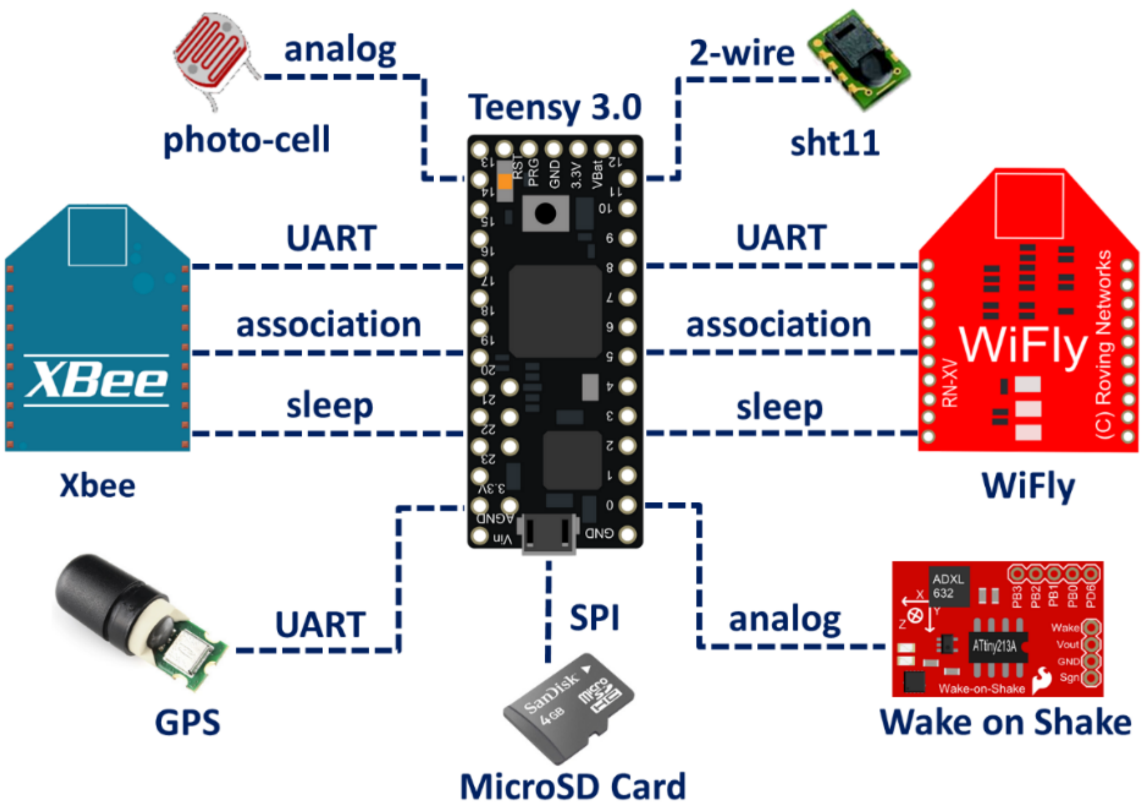
- <u>Wake on shake:</u>
  - Integrates a low-power micro-controller and the ADXL362 accelerometer.
  - Provides a "HIGH" signal whenever it senses an **abrupt shake**.
  - It has **extremely low-power consumption** (<2uA).
  - Note that Teensy cannot acquire vibration measurements by itself when in sleep mode.



Wake on shake board

Components Diagram

✓ Teensy is connected and communicates with all the peripherals:



Components Diagram

UNIVERSITY OF THESSALY

NITLAB NETWORK IMPLEMENTATION TESTBED LABORATORY

Firmware

- Active/Sleep:
  - Teensy periodically executes a **full measurement cycle** as long as the node is **moving**.
  - If the node stays **immobile** for a specific period of time, Teensy puts itself and all the peripherals in **sleep mode** apart from the WoS.
  - Normal operation is resumed when Teensy is woken up by WoS.

- Sensing Procedure/WiFi scan:
  - Teensy puts the WiFly in command mode and **instructs** it to perform a **scan** operation.
  - When the scan is completed, the WiFly responds with the list of networks found.
  - It also reads the values from the rest peripherals.

- Communication with the RC:
  - Teensy monitors the **association pin** of the communication interface (WiFly or ZigBee) to detect that a connection to a gateway has been established.
  - If so, it initiates a conversation with the RC. (TCP connection).

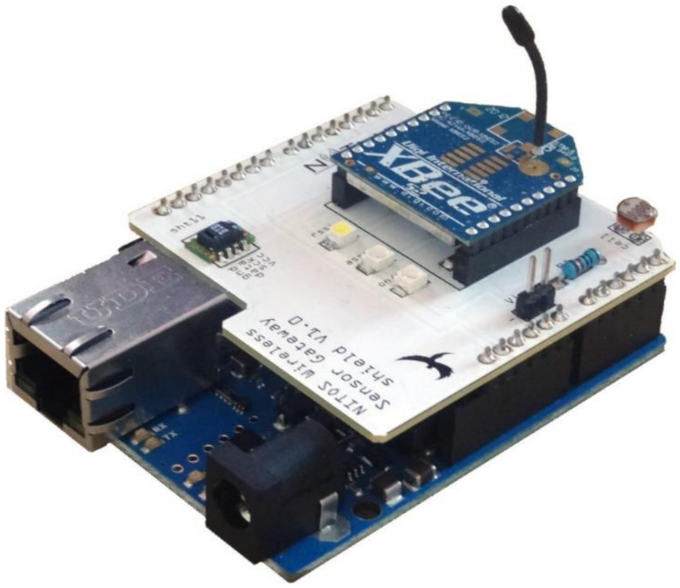
ZigBee-like Gateway

- The ZigBee-like Gateway is based on an **Arduino Ethernet board**, on top of which we mount a **custom shield with an Xbee module**.

- The Ethernet is connected to a **backbone** network through which it communicates with the RC.

- The Xbee is configured as a **coordinator** for a given **PAN id**.

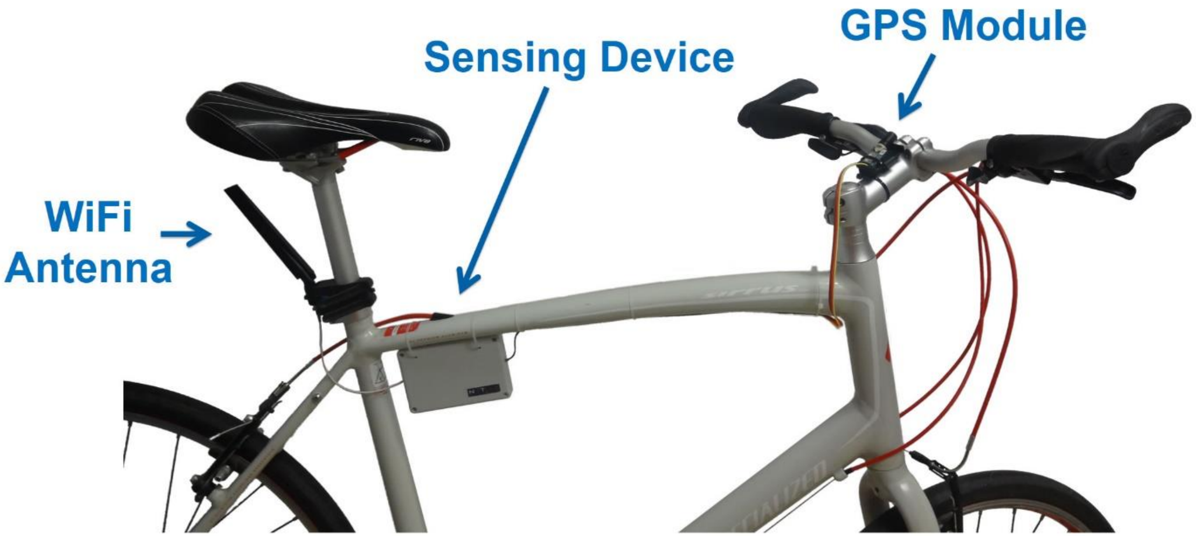
- The gateway initiates a **TCP connection to the RC on behalf of the node**, acting as a proxy.



ZigBee-like Gateway

Installation on Bike

- ✓ The sensor node is **enclosed** in a **waterproof case** and is attached to the bicycle using tire-ups.
- ✓ The **antenna** for the WiFi radio is fixed beneath the saddle, and is connected via a pigtail to the WiFly interface of the node.
- ✓ The **GPS** unit is placed on the **handlebar** of the bicycle in order to have good signal reception.

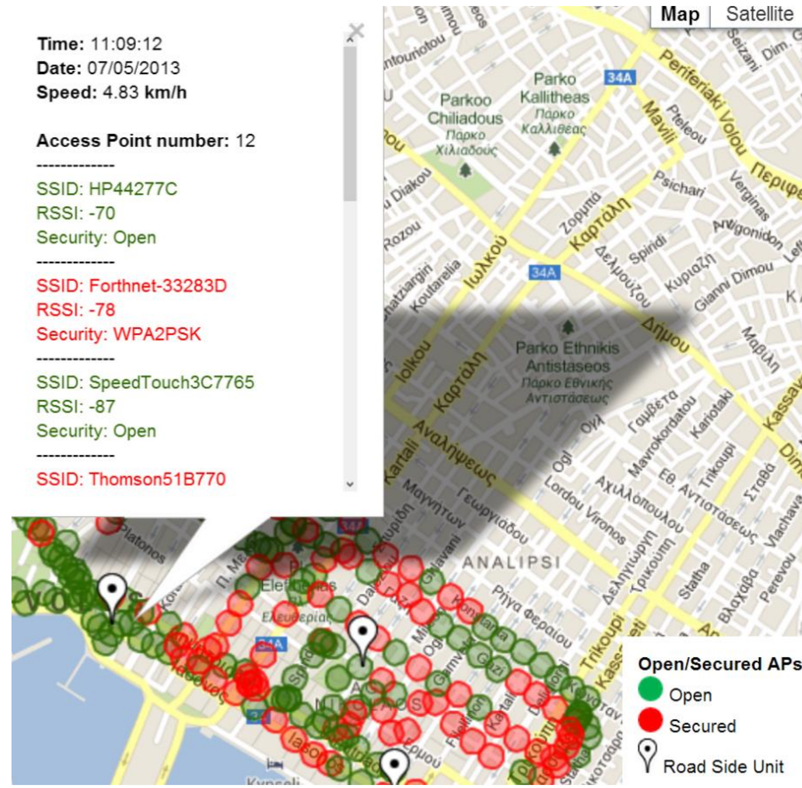


Bike Installation

Visualization Tools

- ✓ We developed a web-based tool based on google maps API that illustrates the collected measurements.

Available WiFi Networks in the city of Volos



Open/Secured WiFi Networks

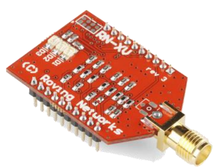
Performance characteristics

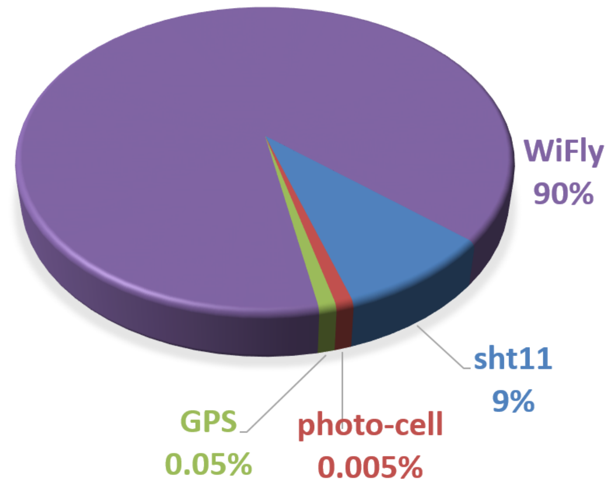
- We evaluated the performance of the NITOS mobile sensor node in terms of:
  - Sensing Latency.
  - Connectivity & Transmission Latency.
  - Power Consumption.

Sensing Latency

- The time required to **acquire** a measurement:
  - **Sht11: 319 ms**
  - **Photo-cell: 230us**
  - **GPS: 2ms**
  - **WiFly: 3180ms**
    (varies on the number of WiFi found)

Limited UART baud rate





Sensing latency per module

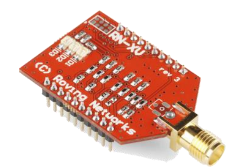
- To **store** the acquired measurements into the **microSD** card, roughly **13ms** are required.
  - in the case that 10 WiFi networks have been found.
- The total time required for **a full sensing cycle** and **data storage** is approximately **3.5 seconds**.
  - means it can perform **15 full sensing cycles per minute**.

Connectivity & transmission latency

- <u>Association Latency:</u>
  - Xbee: 7 secs
  - WiFly: 2-4 secs

- <u>Communication Latency:</u>
  - Although, Xbee module has a physical Tx rate of **250 Kbps**, it only supports the maximum UART baud rate of 57600.
  - It achieves up to **46 Kbps** transmission rate.

  - The Wifly, supports the much higher baud rate of 460800 over UART.
  - Thus, it achieves up to **358 Kbps** transmission rate.

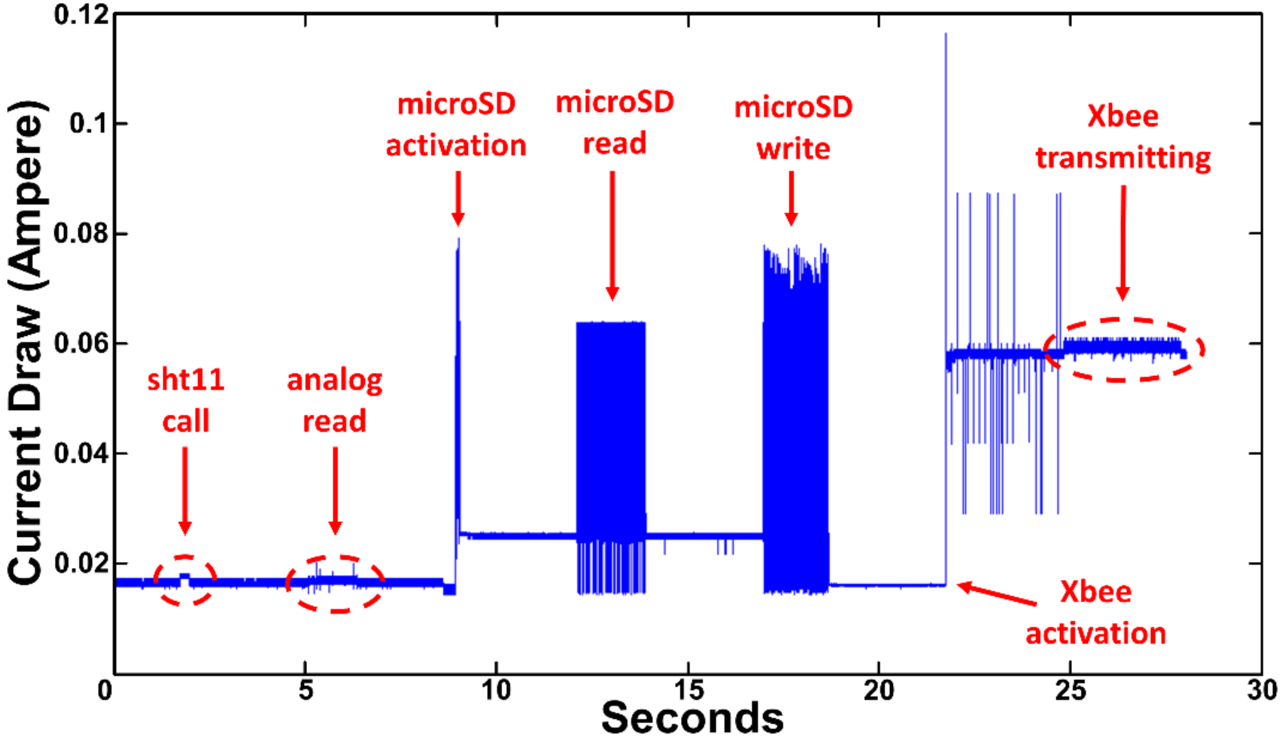


- Assuming we have collected data with size of **90KBytes**:
  - Xbee requires **16 seconds** to upload them,
  - While WiFly only **2 seconds**.

Power Consumption Profile

- ✓ Through the NITOS Energy Monitoring Framework (developed in a previous work), we measure and characterize the power consumption of each individual module as well as the total power consumption.

Developed Energy Monitoring Device



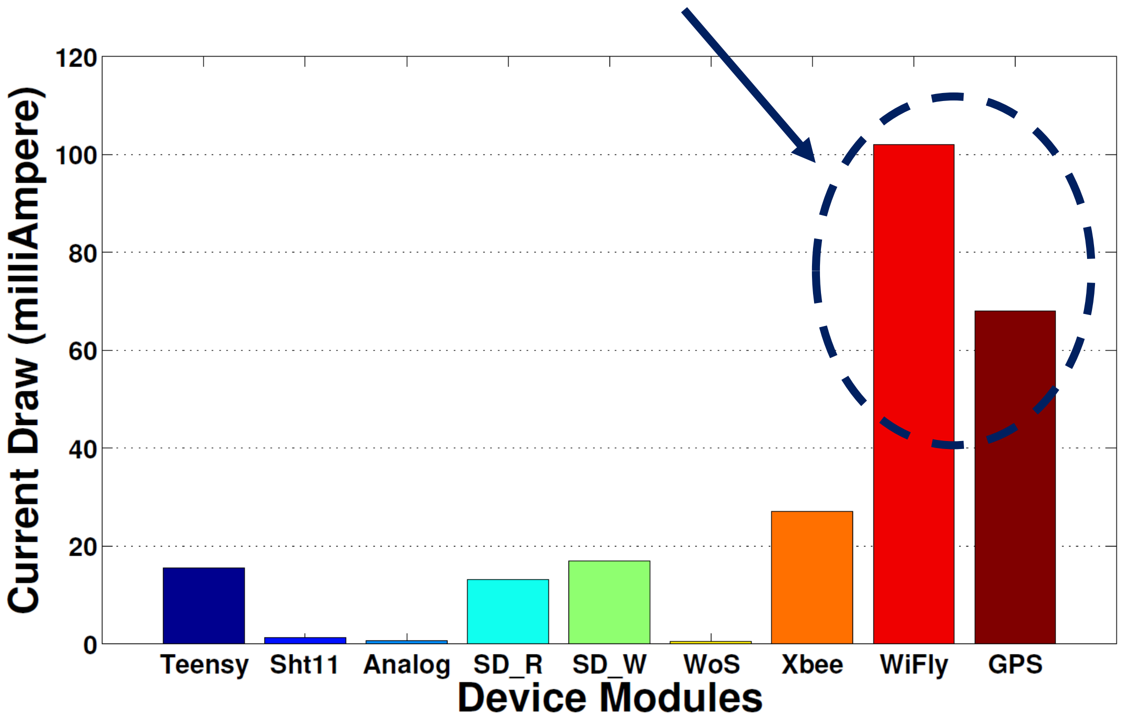
Power Consumption Snapshot of Bicycle's Mobile Device

Power Consumption Profile

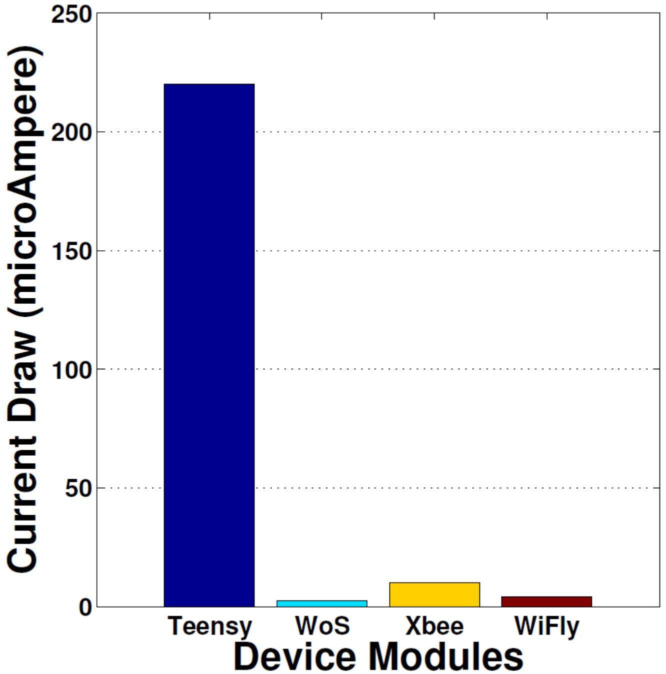
✓ We depict the individual measurements of each module in active and in sleep mode.

Most expensive modules in terms of power consumption

Smart sleeping mechanisms are required to save energy



Active Operation



Sleep Operation

UNIVERSITY OF THESSALY

NITLAB NETWORK IMPLEMENTATION TESTBED LABORATORY

Power Consumption Profile

- Based on repeated measurements:
  - in **active mode** (when the bicycle is moving),
  - involving **all the onboard sensors** and with a sensing/logging period of **10 seconds**,
- we estimate that the device drains **256 mA on average**.

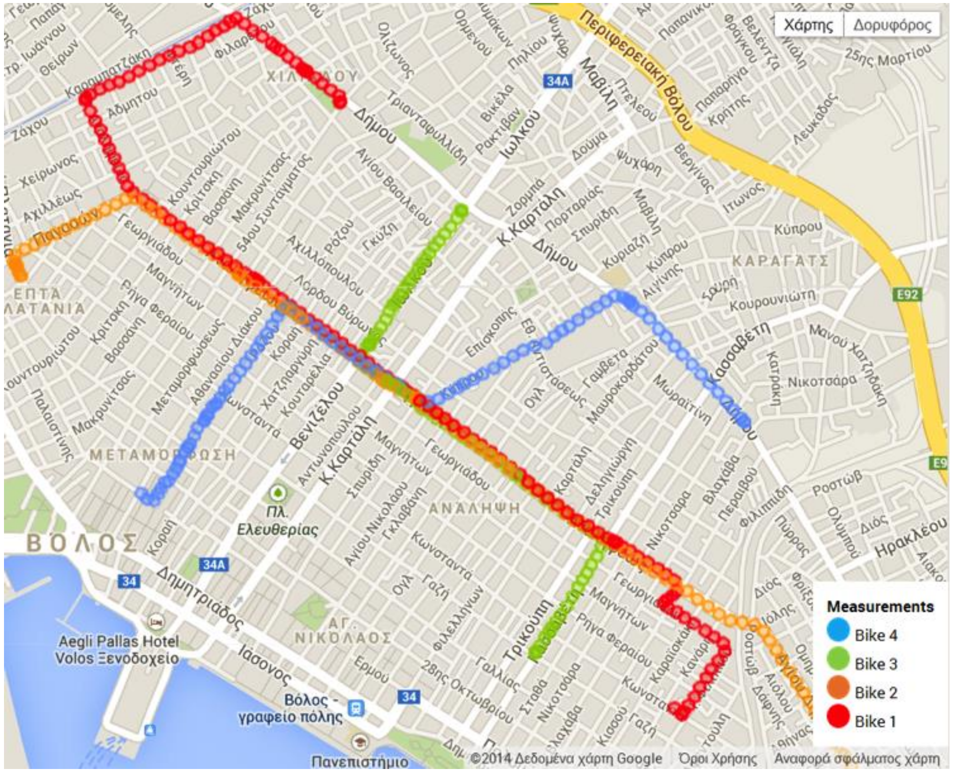
- To note that when in sleep mode (the bicycle does not move) the node requires negligible energy (less than 250uA).

- Thus, we have equipped the node with **3 type AAA** batteries.
- With a total capacity of **6600 mAh.**
- Achieving **more than 25 hours** of continuous operation.

- Assuming a bicycle is **on the move for less than 40 minutes** a day.
- The mobile node will has a lifetime of **one month**.

A Use Case/Experimentation Scenario

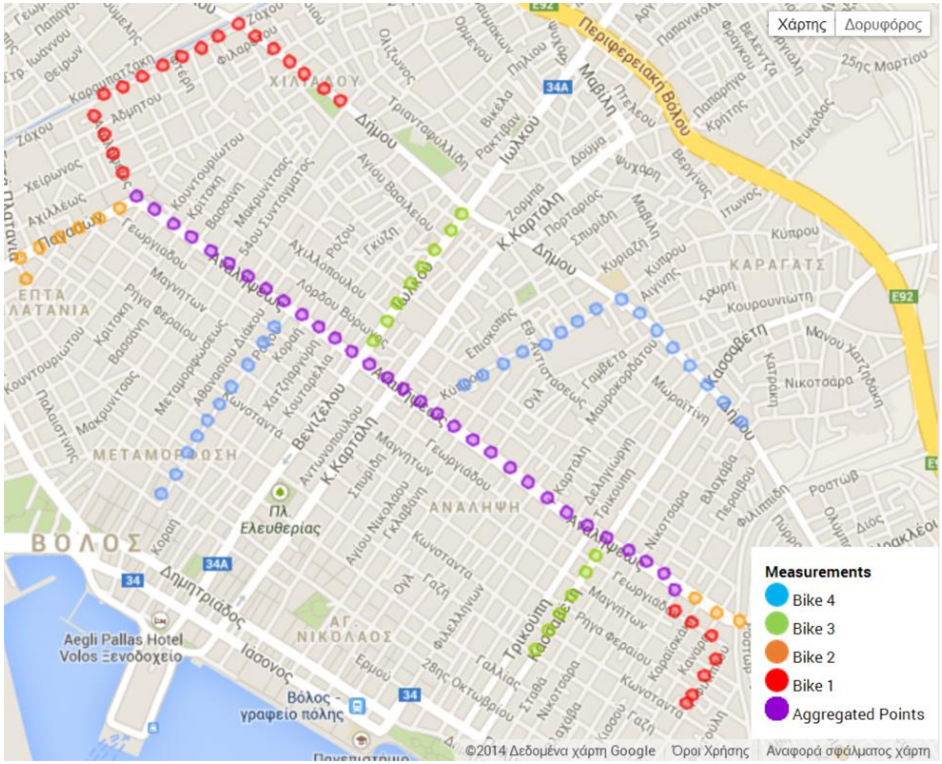
- ✓ In this use case:
  - ✓ We employed 4 bicycles.
  - ✓ Each bike follows a different route.
  - ✓ All the bicycles feature a WiFly and monitor for WiFi networks.
  - ✓ Each bicycle has 5 seconds sensing interval.
  - ✓ The collected measurements were uploaded to the NITOS server at the end of the route.



- ✓ The figure depicts the individual measurements collected by each bicycle.

Aggregated Measurement Points

- ✓ The figure shows an aggregated view of the collected data.
- ✓ Adjacent measurements from different bicycles are merged into a single data point in the map.
- ✓ The mobile nodes took a measurement every 20-27 meters.
- ✓ The mobile nodes generated data at rates between 1,28 and 1,36 Kbps.



Bike id	Samples	Duration	Distance	Avg. speed	Data Size
1	136	11':15"	3.7 Km	19.7 Km/h	106 kB
2	82	6':46"	1.7 Km	15.1 Km/h	67,7 kB
3	127	10':28"	3 Km	17.2 Km/h	97,9 kB
4	102	8':25"	2.2 Km	15.7 Km/h	84,7 kB

Conclusions

- By exploiting the OMF we managed to develop a framework that supports:
  - Experimentation Capability.
  - Remote Configuration.
  - Disconnected Operation.
- Support different networking technologies:
  - By building a **modular** system and evaluating both WiFi and ZigBee technologies.

Conclusions

- ✓ Low-cost:
  - By developing our own solution with open-source components.
- ✓ Low-power:
  - By using low-power components.
  - And by properly set them in sleep mode whenever possible.
- ✓ Small-size:
  - By exploiting tiny modules.
- ✓ Extensibility:
  - By using Arduino platform that supports a vast variety of sensing modules and components.

Future Work

- Enhance the NITOS sensor node to support **additional types of sensors**:
  - measuring noise / air pollution.
  - spectral scan of the wireless bands.

- Alternative **wireless access technologies** for the communication with the RC:
  - Cellular / Bluetooth / Combination.

- More **sophisticated power management** policies at the firmware level, to further increase the autonomy of the mobile nodes.
  - Activate communication interface whenever close to the gateway (based on GPS measurements).

Future Work

- Exploit user's **smartphone**:
  - In order to act as a gateway for the mobile sensor.
  - Use smartphone's GPS for localization.
  - Exploit smartphone's powerful CPU for computations.

- Further application scenarios that could be supported:
  - **Environmental monitoring.**
  - Detection of **potholes** on roads.
  - Inferring **traffic jams** to propose alternative routes.

UNIVERSITY OF THESSALY

NITLAB NETWORK IMPLEMENTATION TESTBED LABORATORY

References

- ✓ NITOS/NITlab site: http://nitlab.inf.uth.gr/NITlab/
- ✓ OMF: http://mytestbed.net/projects/omf/wiki/OMFatNICTA
- ✓ NITOS BikesNet: http://nitlab.inf.uth.gr/NITlab/papers/NITOS_BikesNet.pdf
- ✓ NITOS Energy Monitoring Framework: http://nitlab.inf.uth.gr/NITlab/papers/Keranidis_NITOS_EMF_WinTECH_2013_paper.pdf

Thank You!

Giannis Kazdaridis
iokazdarid@gmail.com

Date: 15/07/2014

CERTH

NITLAB
NETWORK IMPLEMENTATION TESTBED LABORATORY

UNIVERSITY OF
THESSALY