Middle ware and Programming Paradigms for WSN

Budwell Tamuka Masaiti

JNTUCEH(CSE Department)

Middle ware and Programming Paradigms for WSN

Agenda

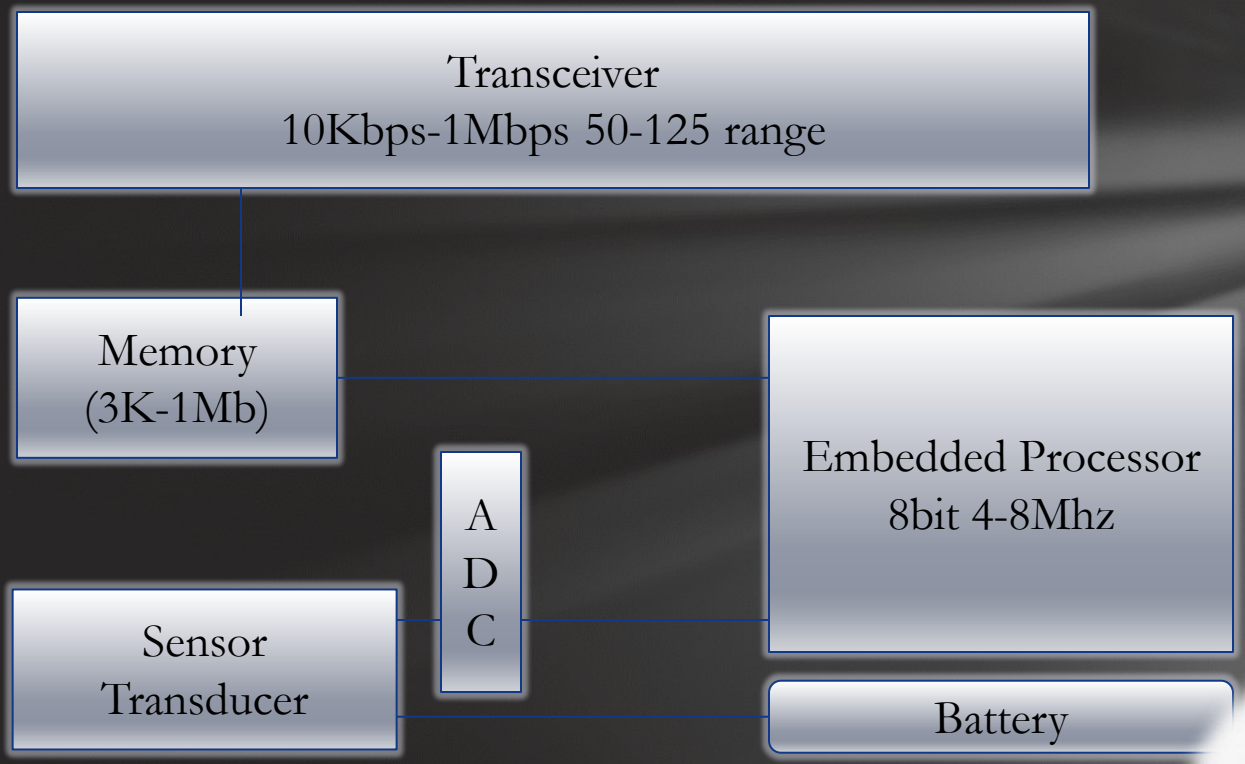
- Wireless Sensor Networks :Introduction
- Applications of WSNs
- Main Issues in WSN
- Middleware and Programming Abstractions for WSN
- Goals for Middleware and Programming abstractions
- Contiki: Event driven abstractions
- TinyOs: Event driven abstractions
- Conclusion

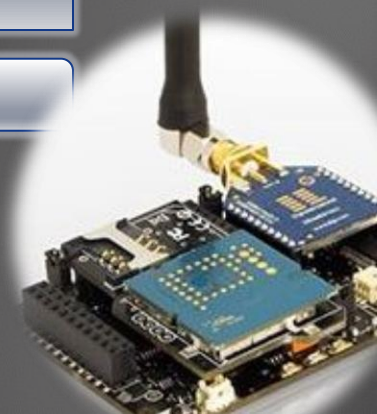
Wireless Sensor Networks :Introduction

A **wireless sensor network (WSN)** consists of spatially distributed autonomous sensors to

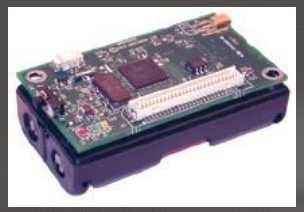
monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location.

Sensor Node

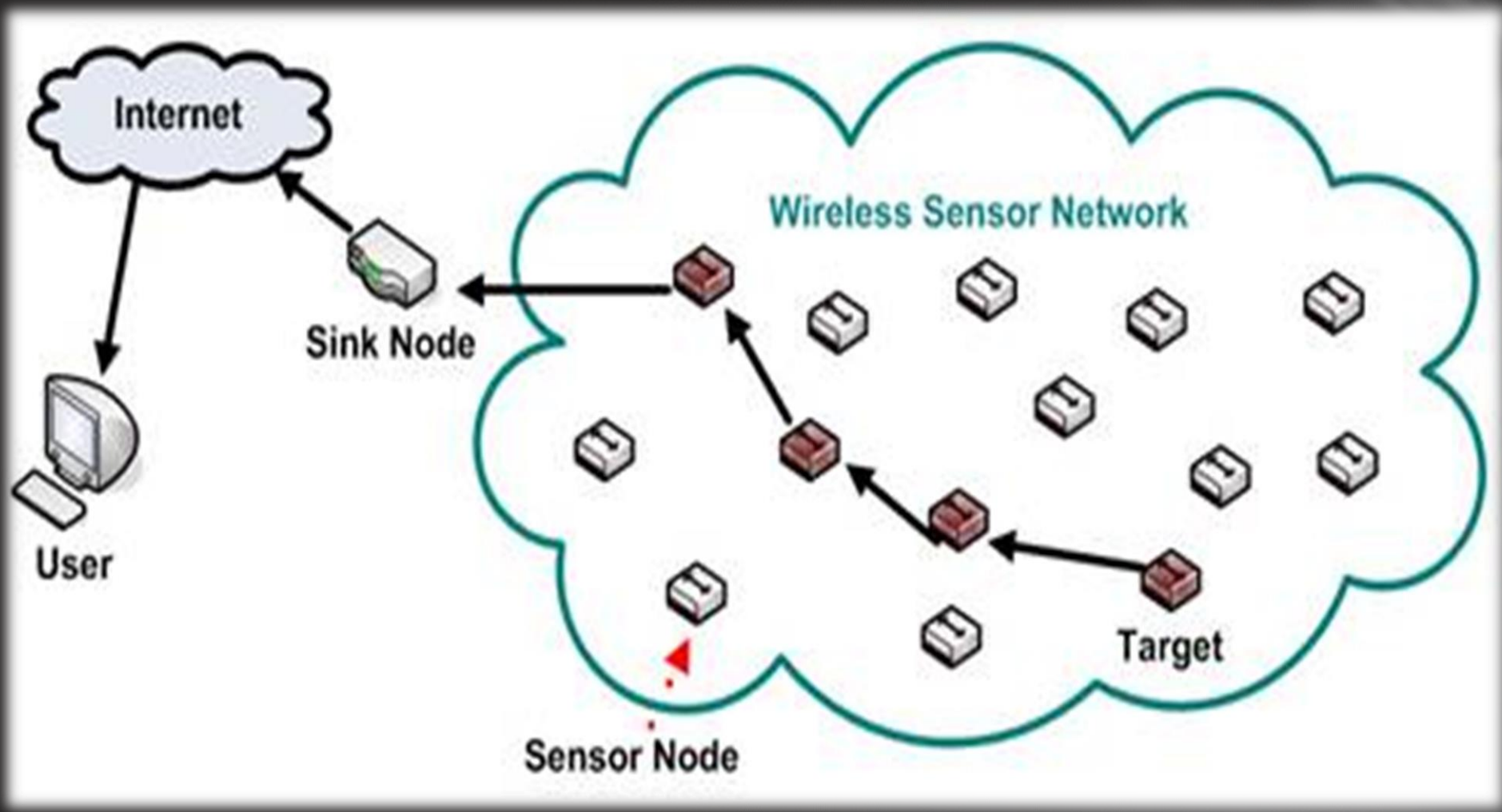

Transceiver
10Kbps-1Mbps 50-125 range
Memory
(3K-1Mb)
Embedded Processor
8bit 4-8Mhz
A
D
C
Sensor
Transducer
Battery



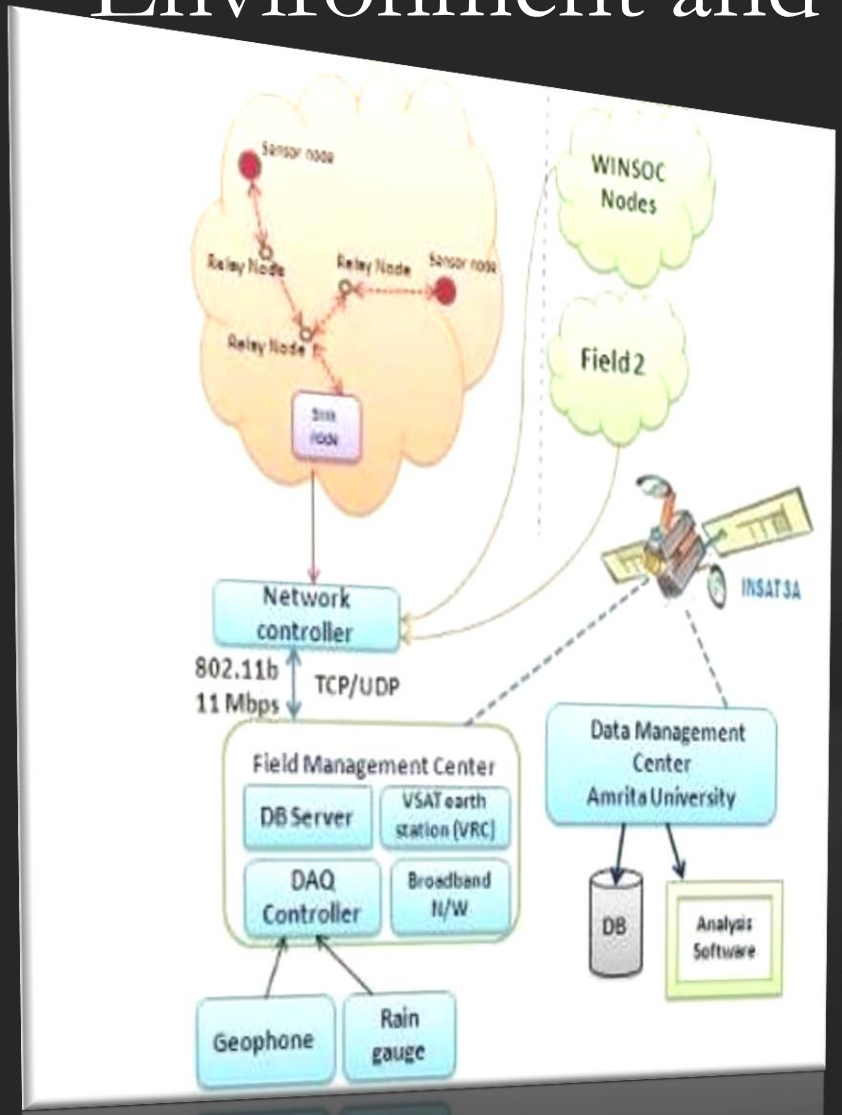
Popular Implementations

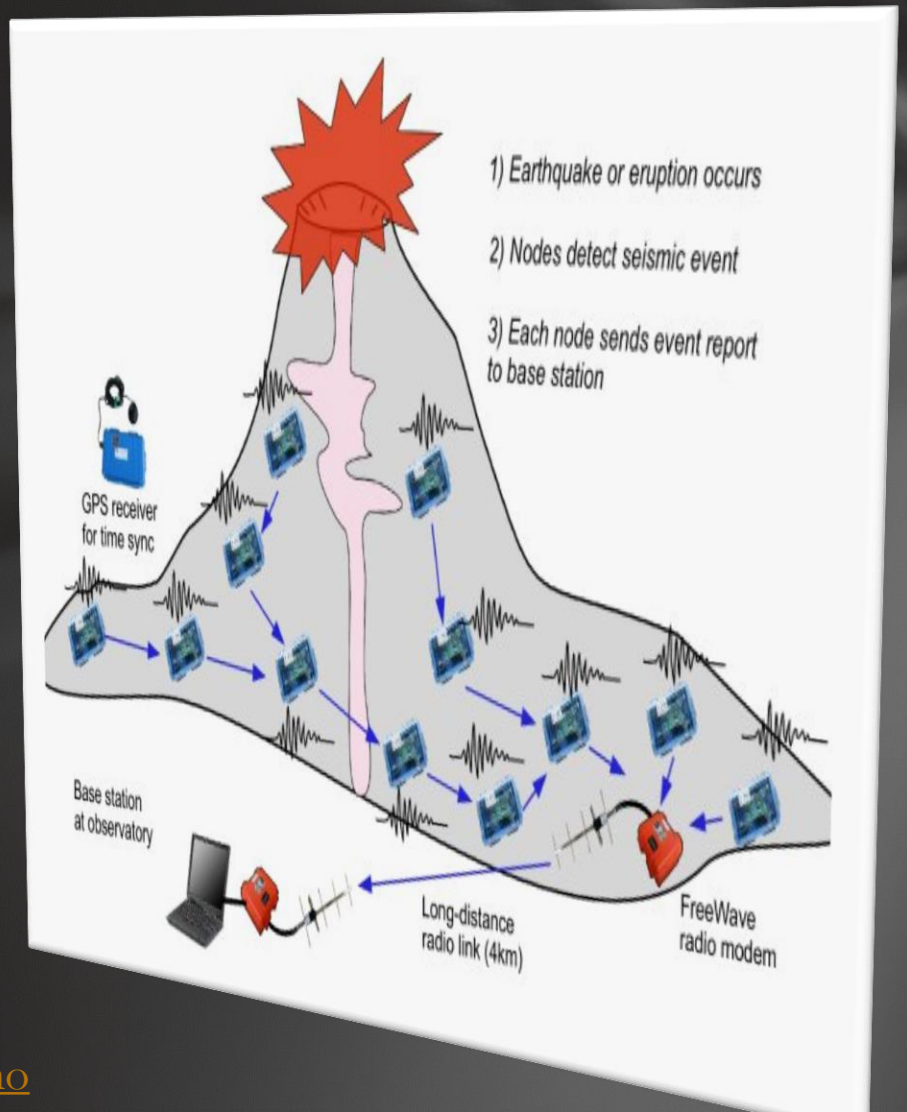
mica	mica2	mica2dot	micaz
			
telos	**telosb**	**rene2**	**waspmote**
			

WSN Architecture



Environment and Habitat Monitoring

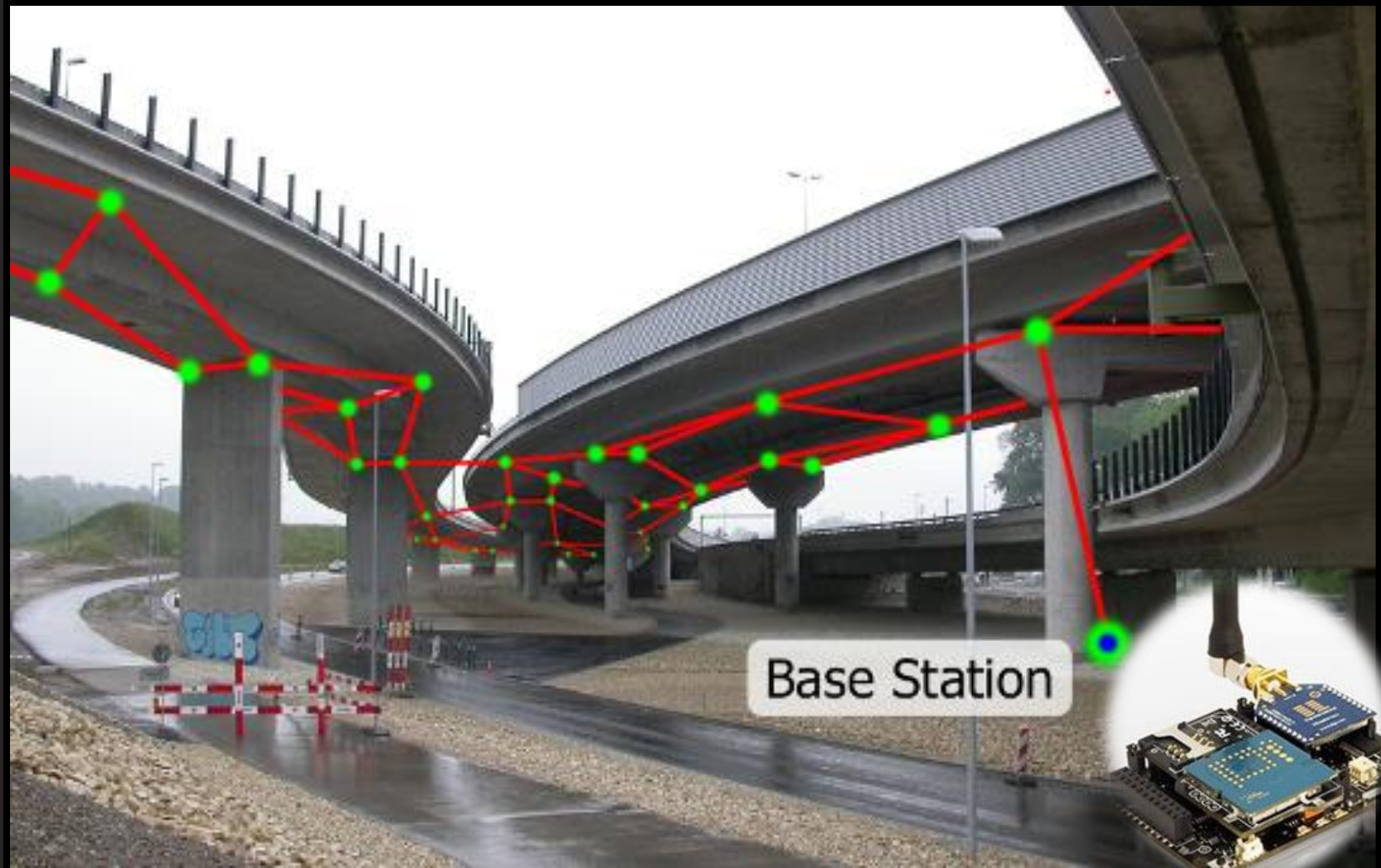




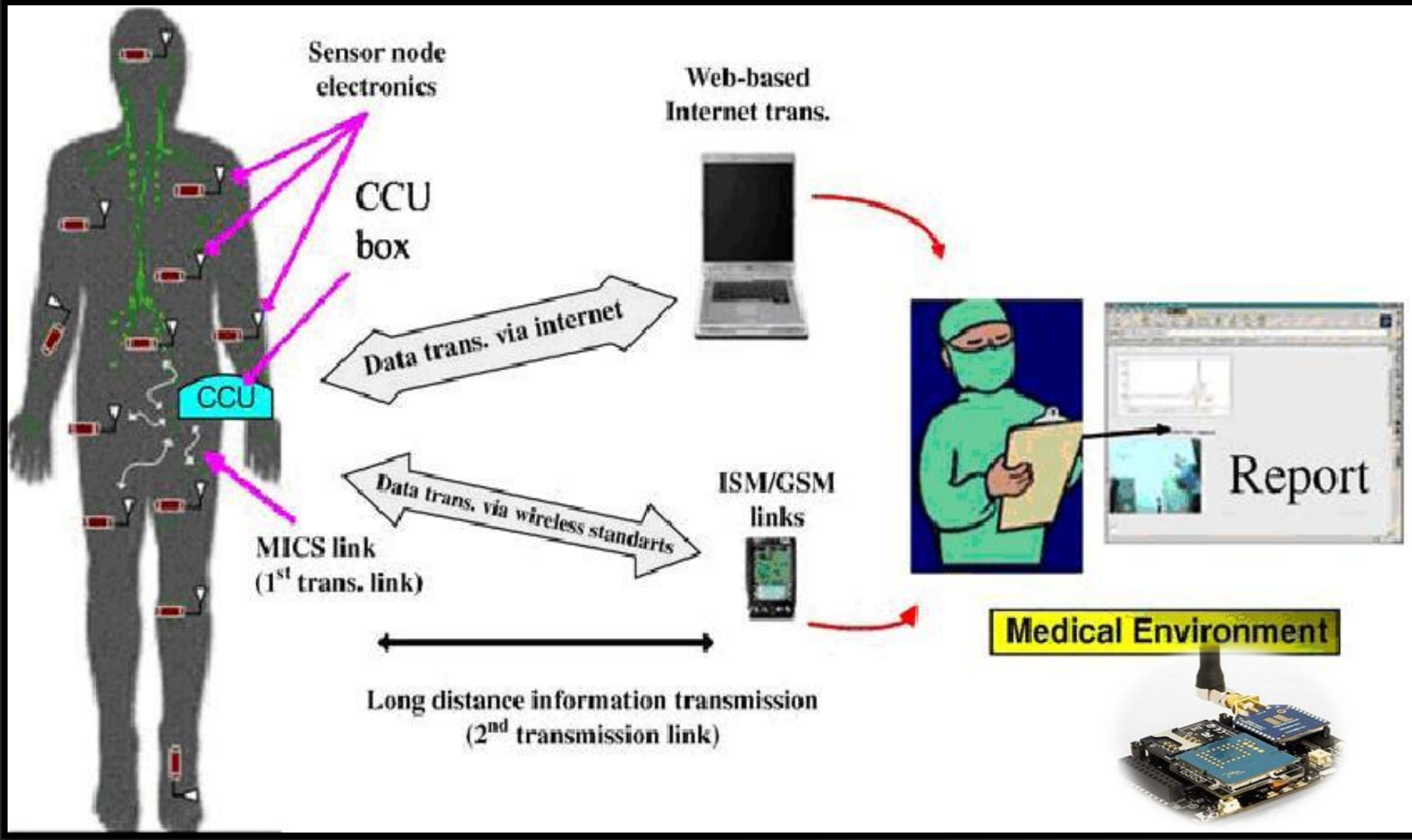
http://fiji.eecs.harvard.edu/Volcano

Infrastructure Health Monitoring

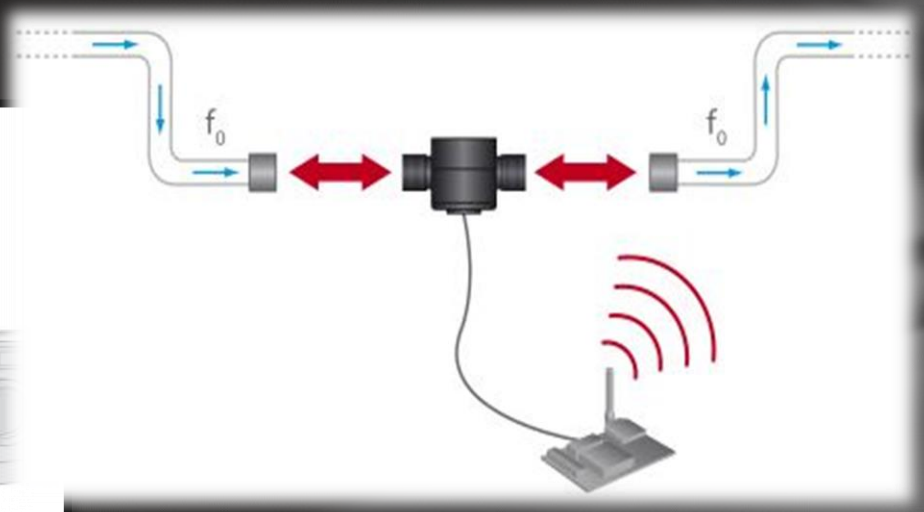
- Monitoring bridge health

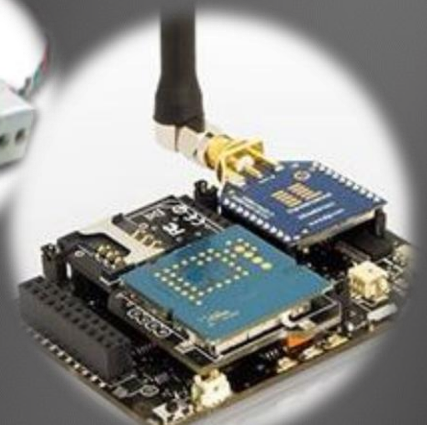


HealthCare Monitoring(BAN,PAN)



Smart Metering





WSN Middleware and Programming Abstractions

- Middleware provides the software layer interfacing the hardware and the application programs in an OS less system
- Or the software layer between the OS and the application programs.
- Programming abstractions refer to the interfaces and data structures provided for a programming model and language

Goals for WSN Middle and Programming Abstractions

- Automatic node management
- Concurrency management
- Resource management
- Developer friendly abstractions

Contiki Adam Dunkels

Background:

- Contiki – pioneering open source operating system for sensor networks

- Development started in 2001

Got its name from Kon Tiki

Across the Atlantic with resources

Contiki Key Features

- Small memory footprint
- IP networking
- Hybrid threading model, protothreads
- Power profiling

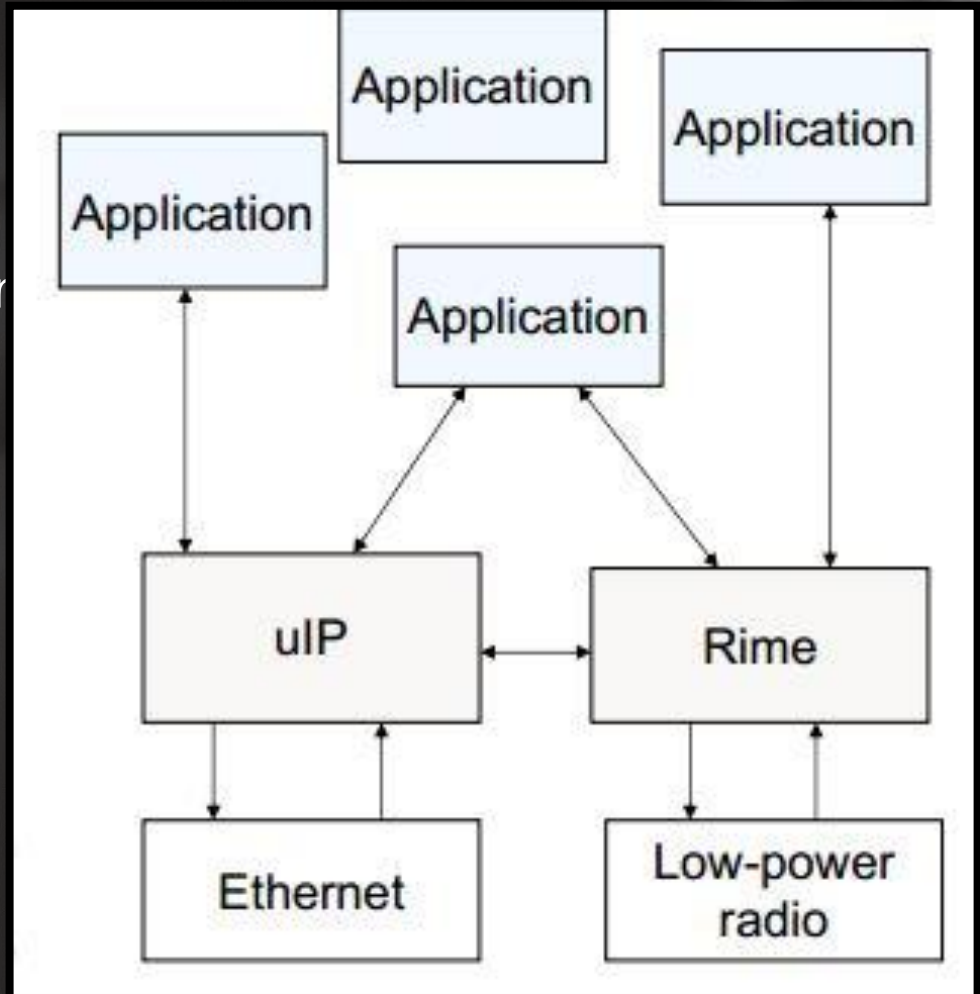
  -measure power consumption at network scale
- Network shell

  - for easy command line interaction
- Designed for portability
- Dynamic loading

Contiki Communication Stack

- Two communication stacks
- uIP TCP/IP
- Rime low overhead

- Can run on top of each other



Contiki :Event Driven Paradigm

- The Contiki kernel is event-based

- Invokes processes whenever something happens

  Sensor events, processes starting,

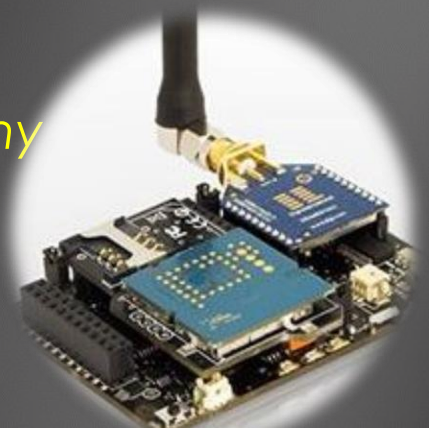
- Process invocations must not block

- *Protothreads provide sequential flow of control in Contiki processes*

- Protothreads extremely lightweight and stackless

Contiki : Event driven abstractions

- Event-driven vs multithreaded
- Event-driven requires less memory
- Multithreading requires per-thread stacks
- *Threads require per-thread stack memory*
- *Events require one stack*
- *Protothreads like events require one stack for as many there are protothreads running*



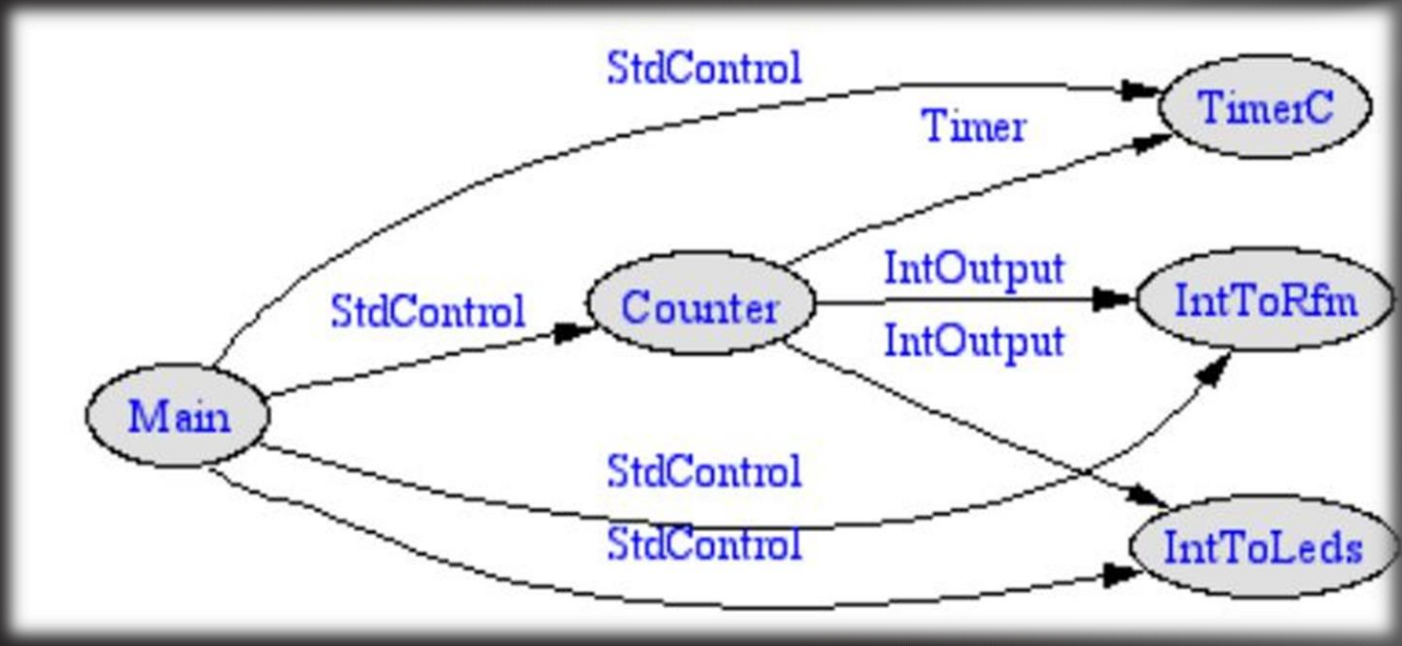
TinyOs:Event Driven Abstractions

- An event driven OS specifically designed for WSN
- Completely non-blocking
- Programs are built out of software components
- Tasks are non –pre-emptive and run in FIFO order
- Code is statically linked

TinyOS :Scheduling

- Two level scheduling: events and tasks
- Scheduler is simple FIFO
- A task cannot pre-empt another task
- Events pre-empt tasks (higher priority)

- Provides a specific Service
- Message Handling, Signal Processing
- Implemented in a Module(code)
- Wired up of other components in a Configuration



Comparison: Contiki and TinyOs

COMPARISON

TINY OS	CONTIKI OS
❖Event –driven OS with **non-preemptive** multitasking.	❖ Event-driven OS with **optional preemptive multitasking**.
❖ **Static** linking.	❖**Dynamic** linking.
❖Written using **nesC** programming language.	❖Written using **C** Programming language.

Conclusion

- Unique challenges and opportunities
- Expanding areas of applications
- Contiki and TinyOs
- New research directions

references

- Geoff Werner-Allen, Konrad Lorincz, and Matt Welsh et al .Monitoring Volcanic Activity at Reventador Volcano, Ecuador with a Wireless Sensor Network
- Oliver Kansten and Kay Romer. Beyond Event Handlers:Programming Wireless Sensors with Attriunuted State machines
- A Dunkels,Bjorn Gronvall ,and Thiemo Voigt ,Contiki –a Lightweight and Flexible Operating System for Tiny Networked Sensors"inEmNeTSI,Tampa USA 2005
- http://www.wsnmagazine.com/tinyos-tutorial/
- http://fiji.eecs.harvard.edu/Volcano
- https://www.sics.se/search/content/contiki
- http://www.tinyos.net