Design, Implementation and Evaluation of Elements of BriMon: A Sensor Network Based Railway Bridge Monitoring System

Phani Kumar Valiveti

Department of Electrical Engineering,

IIT Kanpur

under guidance of

Prof. Kameswari Chebrolu & Prof. Bhaskaran Raman

Outline

- Introduction
- Thesis contributions
- Architecture overview
- Past & related work
- Design and implementation of elements
- Experiments and results
- Conclusion and future work

Outline

- Introduction
- Thesis Contributions
- Architecture Overview
- Past & Related Work
- Design and Implementation of Elements
- Experiments & Results
- Conclusion and Future Work

Introduction

- Motivation
- Problem statement
- Constraints and challenges

Introduction

- Motivation
- Problem statement
- Constraints and challenges

Motivation

- Bridges – a crucial part of railways
  - Indian Railways has over 127,000 bridges
- Railway bridges – safety assurance
  - 40% of Indian railway bridges are 100 yrs old
  - Indian Railways budgets about Rs.7 Billion per annum
- Existing techniques
  - Wired
  - Require technical support on-site
  - Bulky equipment and long setup time

Introduction

- ✓ Motivation
- Problem statement
- Constraints and challenges

Problem Statement

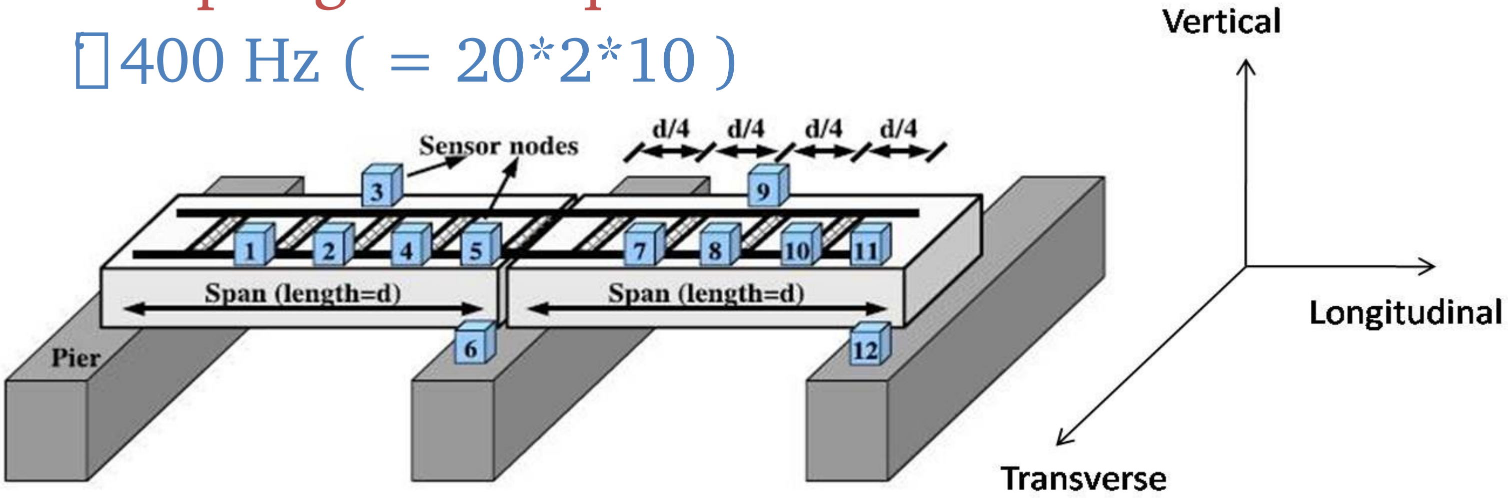
“To build an easily deployable, scalable and maintenance-free system that measures the structural vibrations of a bridge located at a remote place and transfer the data to a repository”

Introduction

- ✓ Motivation
- ✓ Problem statement
- Constraints and challenges

Constraints and Challenges

- Constraints on vibration data
  - □ *Frequency* Content
    - □ 0.25 – 20 Hz
  - □ Sampling *rate* requirement
    - □ 400 Hz (= 20*2*10)



Constraints and Challenges

- Time *synchronization* across readings
  - Within 5ms
- *Duration* of data collection
  - 5 cycles each of *free* and *forced* vibrations
    => 10 cycles (of 0.25 Hz) = 40 seconds

Constraints and Challenges

- Limited power supply
  - □ Calls for *duty cycling*
- Event detection
- Node failures
  - □ Need for a *dynamic topology*
- Limited platform capabilities – sensor motes
  - □ Program and data memory
  - □ Communication range
  - □ Computational power

Outline

✓ Introduction
- Thesis Contributions
- Architecture Overview
- Past & Related Work
- Design and Implementation of Elements
- Experiments & Results
- Conclusion and Future Work

Thesis Contributions

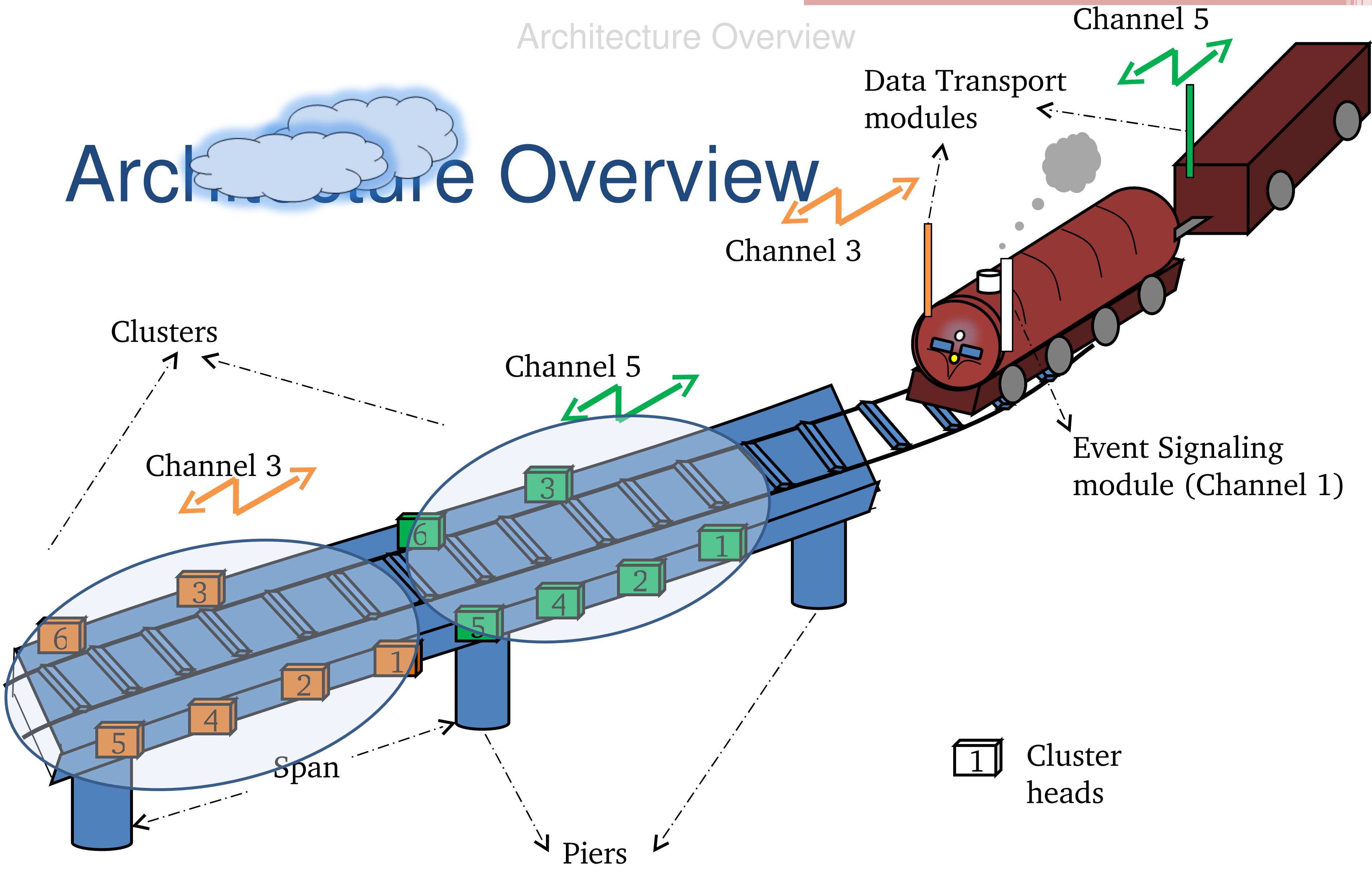
- Routing Protocol – Topology formation
- Time synchronization
- Sleep/Wakeup – Duty cycling
- Event Detection – Train arrival detection
- Integration

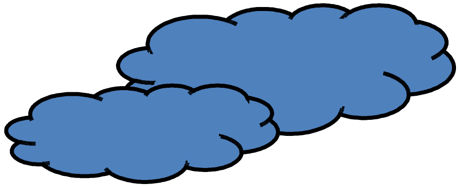
Outline

- ✓ Introduction
- ✓ Thesis Contributions
- Architecture Overview
- Past & Related Work
- Design and Implementation of Elements
- Experiments & Results
- Conclusion and Future Work

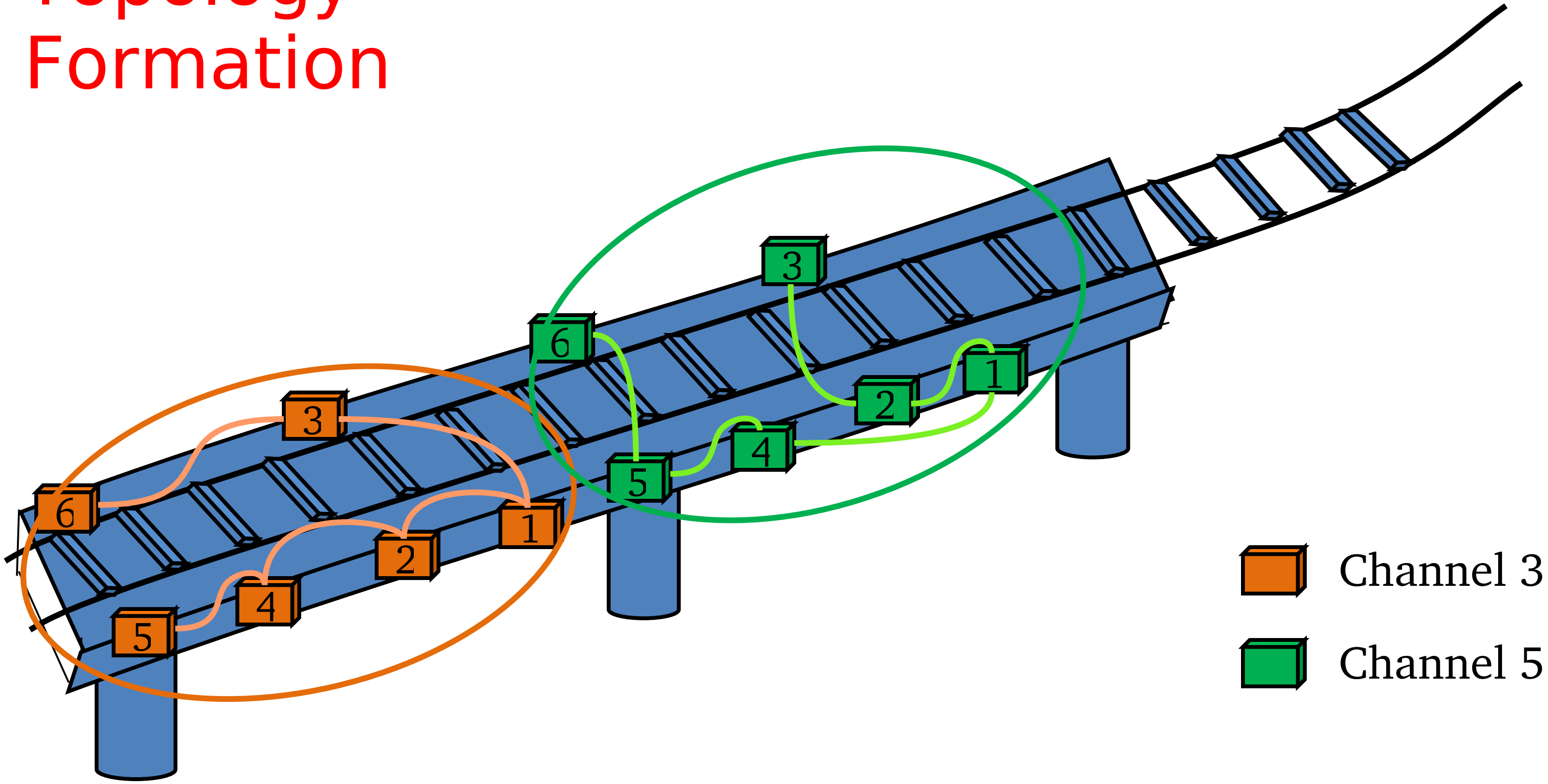
Architecture Overview

Architecture Overview

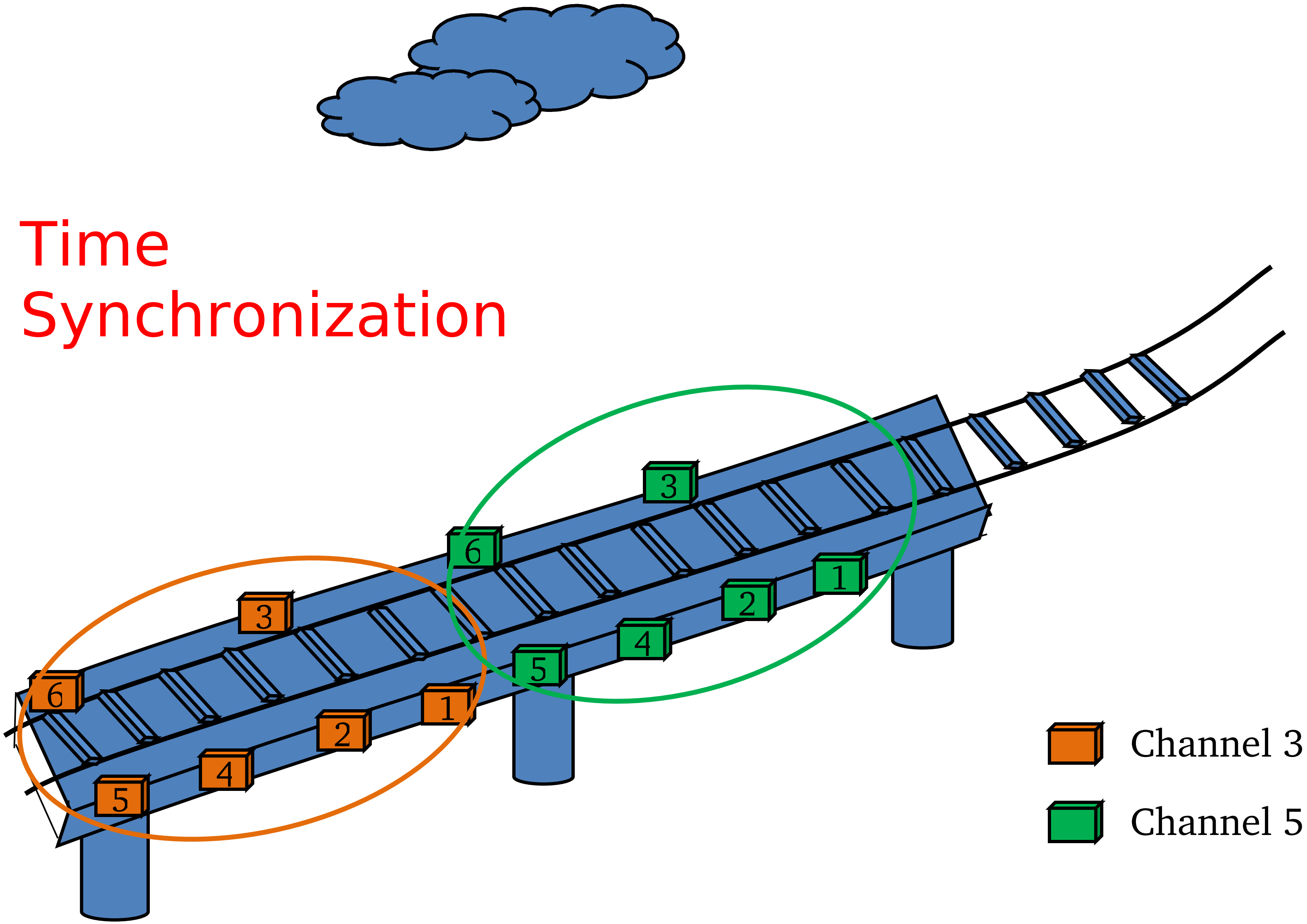


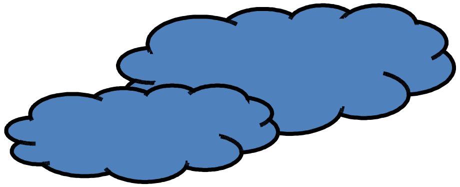


Topology Formation

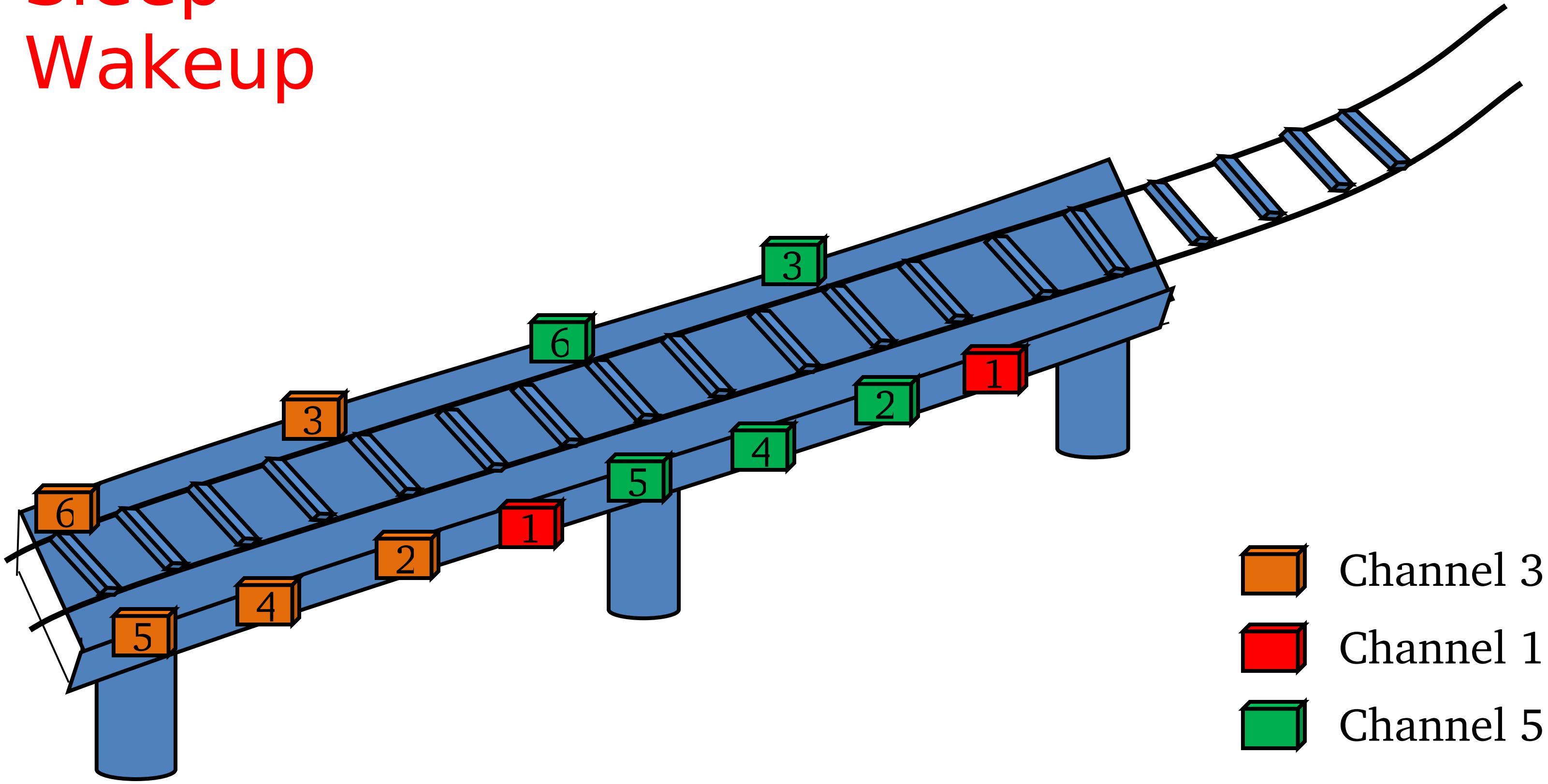


Time Synchronization



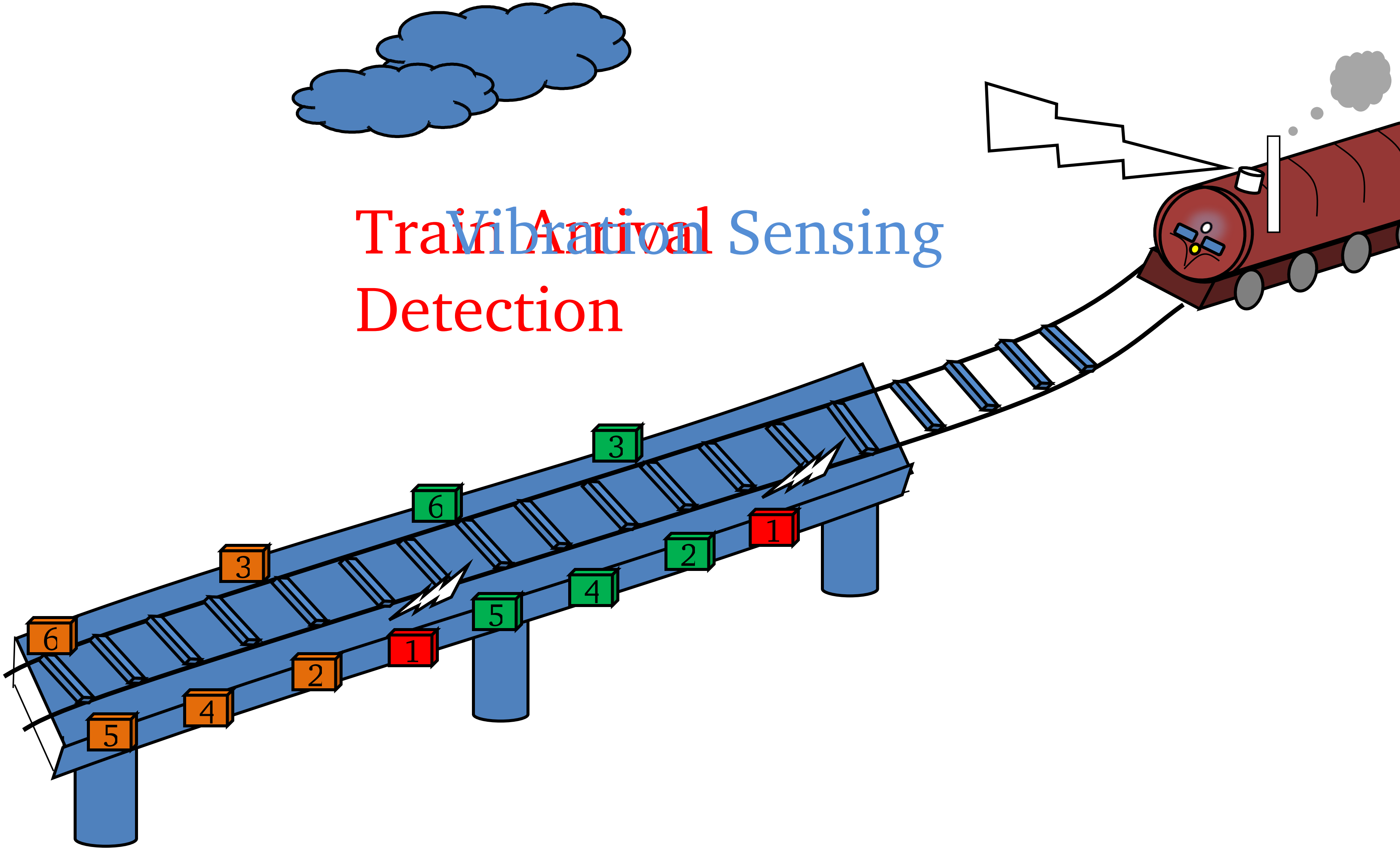


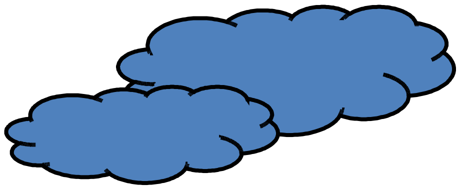
Sleep-Wakeup



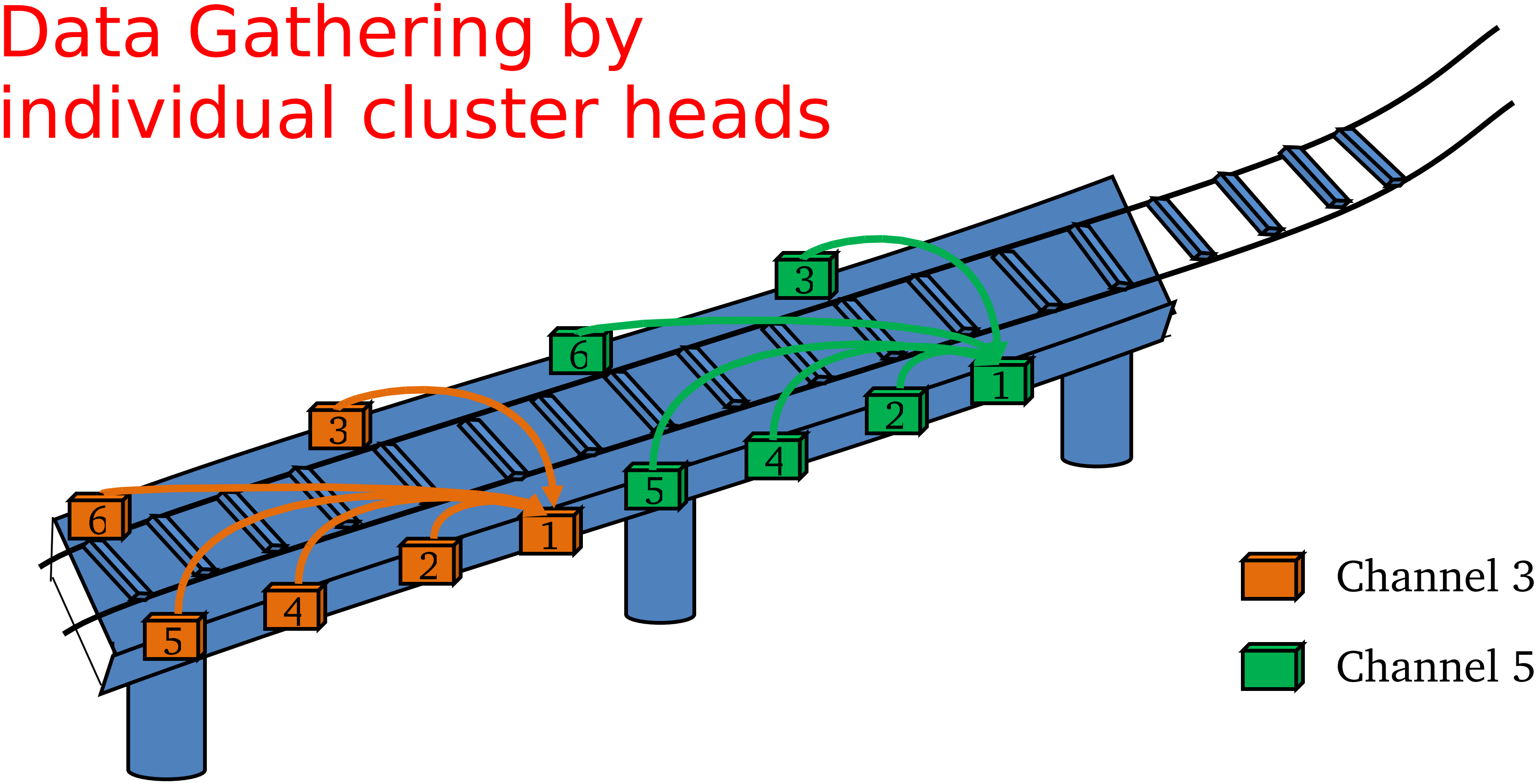
Train Arrival Detection

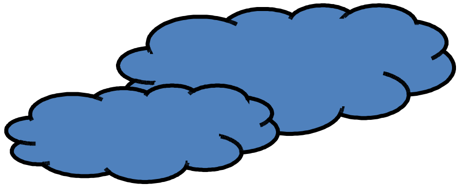
Vibrational Sensing



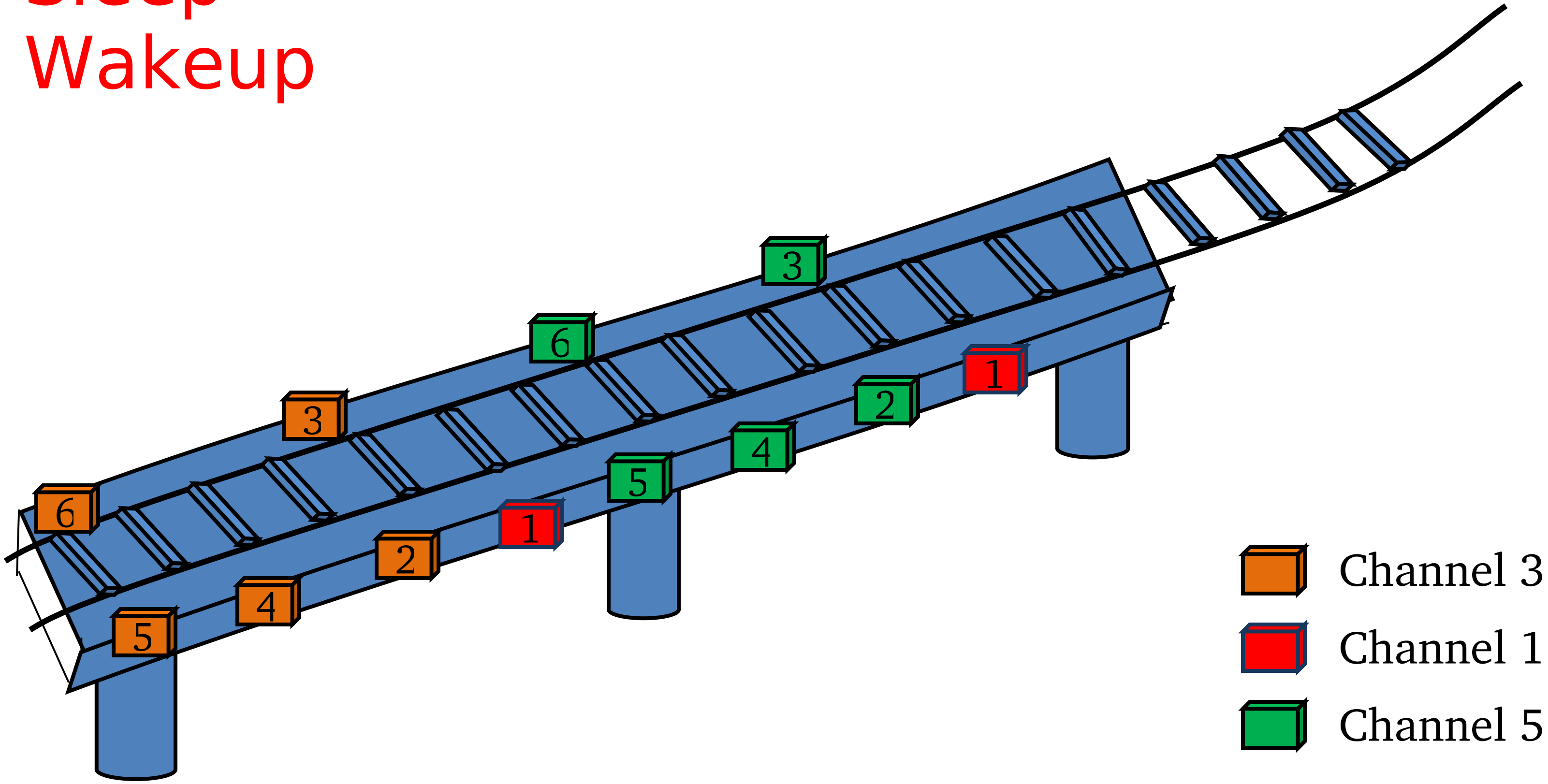


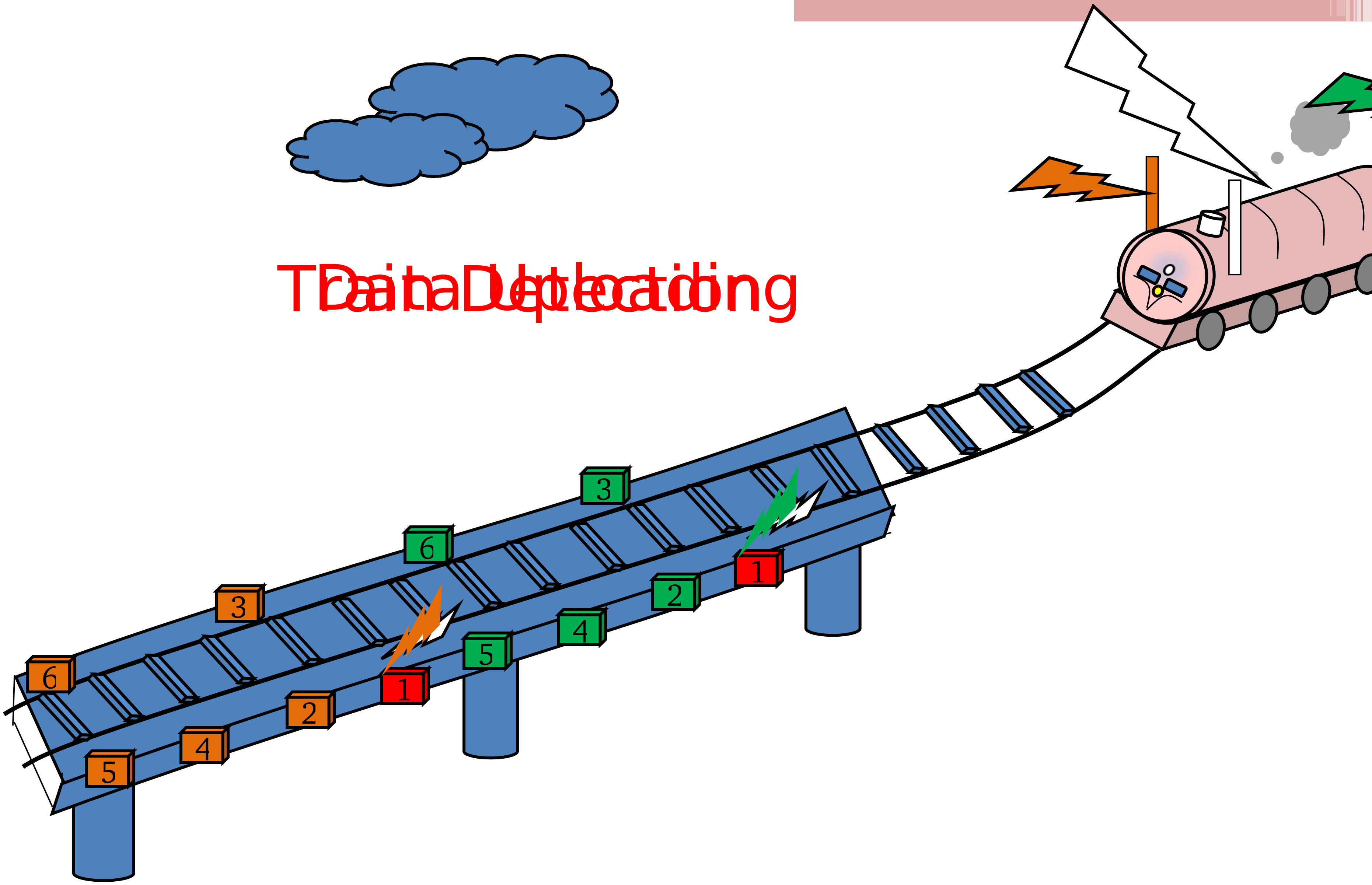
Data Gathering by individual cluster heads





Sleep-Wakeup

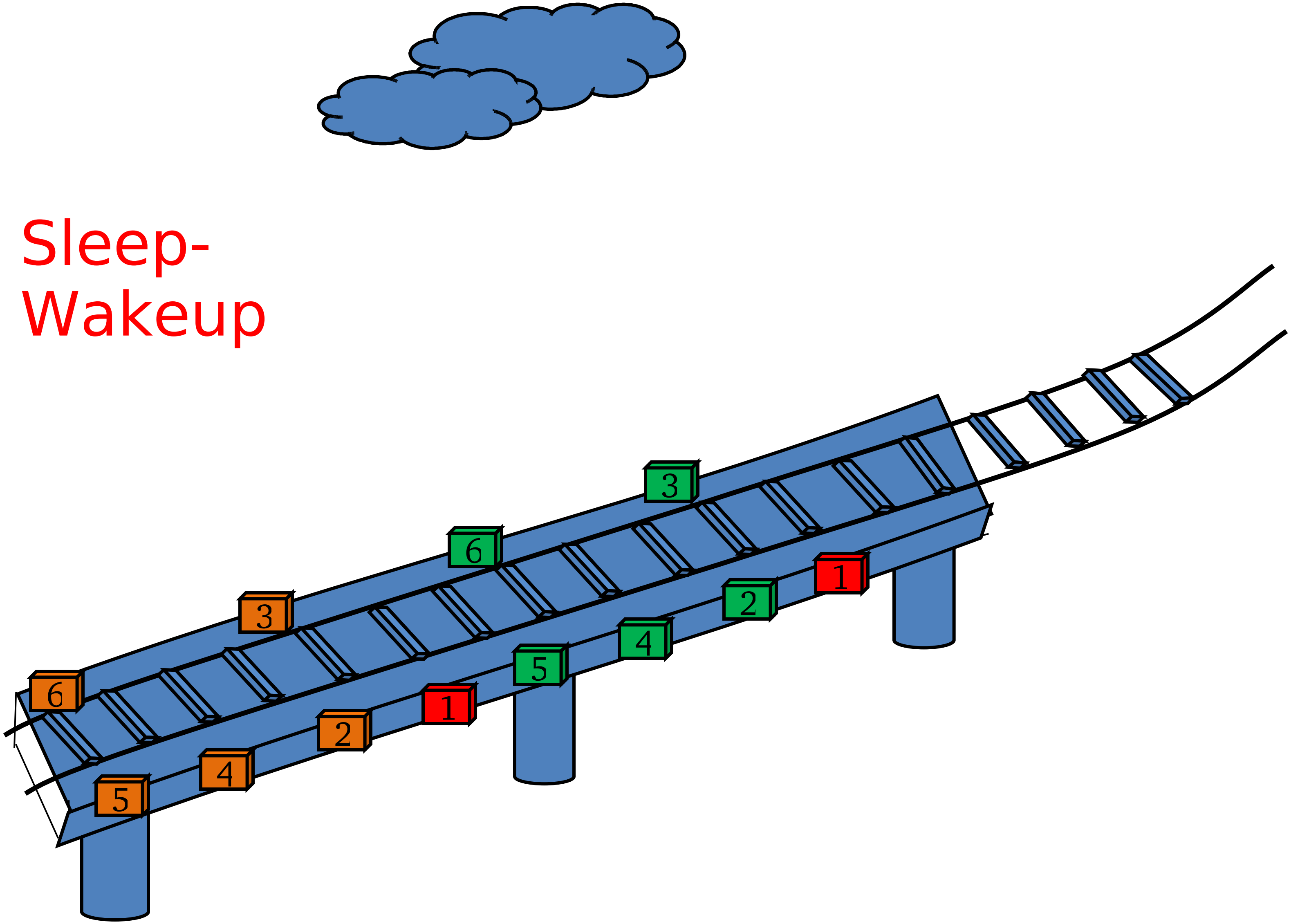


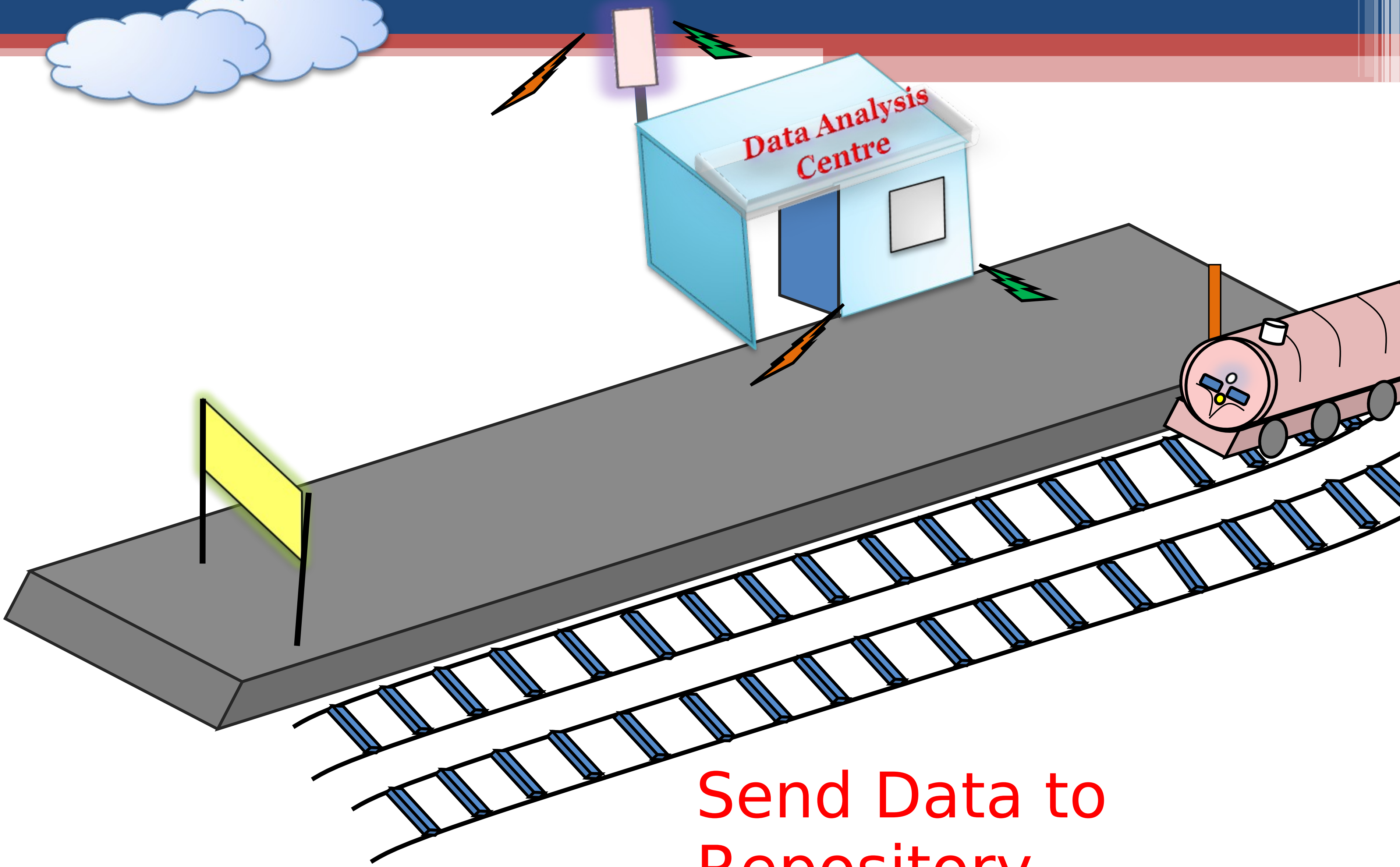


Train Uploading

Data Duplication

Sleep-Wakeup





Send Data to Repository

Architecture Overview

- Salient components
  - □ Network of sensor nodes – Routing protocol
  - □ Time synchronization protocol
  - □ Sleep/Wakeup
  - □ Event Detection
  - □ High fidelity data acquisition
  - □ Reliable Data Transfer – Transport protocol
  - □ Multi-channel mobile data transfer
  - □ Data Analysis

Outline

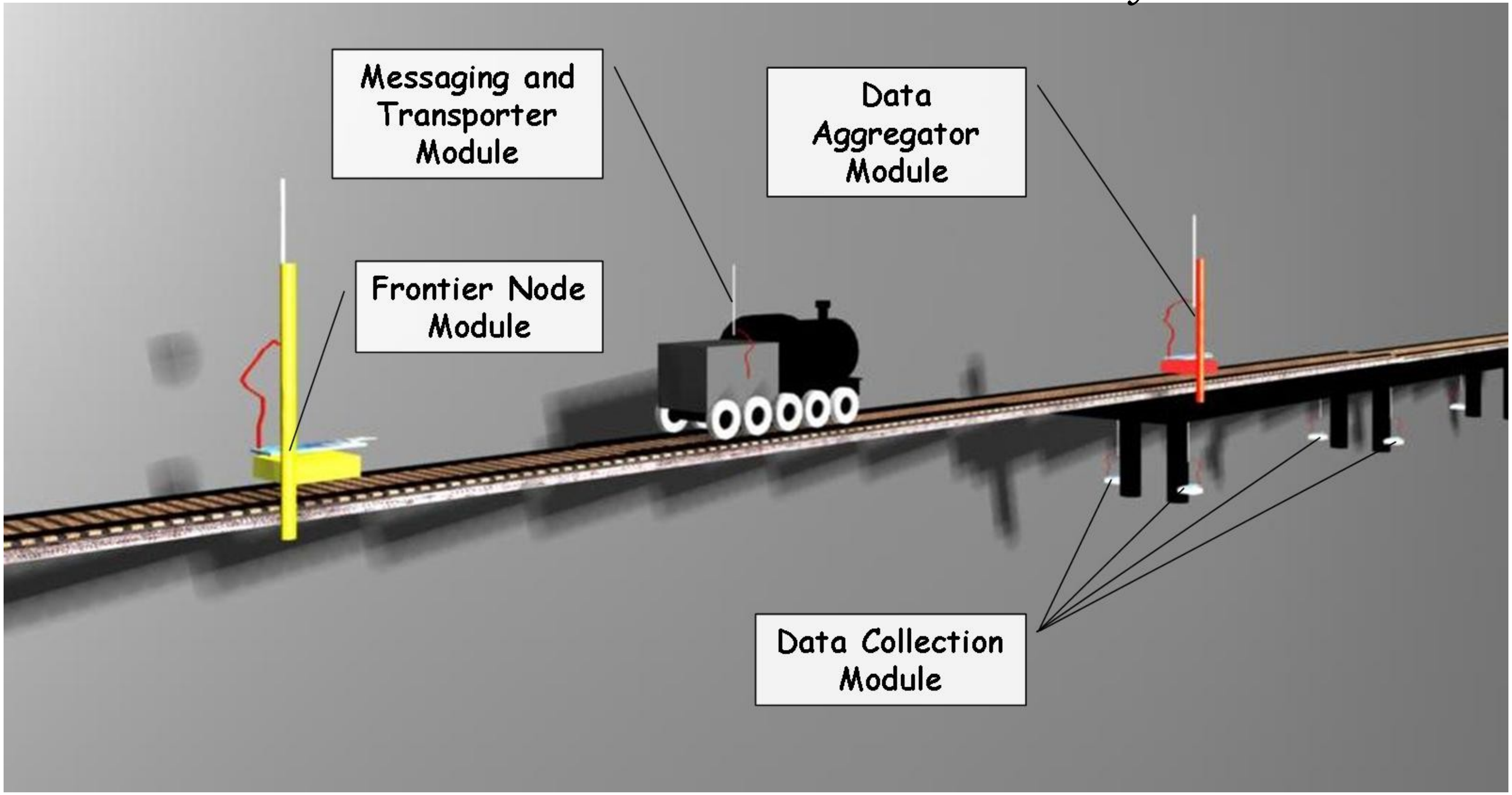
- ✓ Introduction
- ✓ Thesis Contributions
- ✓ Architecture Overview
- Past & Related Work
- Design and Implementation of Elements
- Experiments & Results
- Conclusion and Future Work

Past and Related Work

- Past Work by Hemanth & Nilesh [Fig]
  - □ Protocols not tested
  - □ Design had loopholes
- Disadvantages
  - □ Use of different types of nodes (hardware)
  - □ No scalability
- Past Vs. Present Architecture
  - □ All the nodes are identical in hardware
  - □ Clustering
  - □ Scalability

Past Architecture

Courtesy: Nilesh Mishra



Past and Related Work

- Routing protocols
  - □ LEACH – Adaptive Clustering
  - □ SPIN – Information via Negotiation
  - □ MintRoute
- Irrelevance to BriMon
  - □ Most protocols not implemented
  - □ Nodes are always ON
  - □ No *end* phase

Past and Related Work

- Time synchronization protocols
  - RBS – Broadcast Synchronization
  - TPSN – message *exchange*
  - FTSP – Flooding
- Irrelevance to BriMon - Long synchronization intervals

Outline

- ✓ Introduction
- ✓ Thesis Contributions
- ✓ Architecture Overview
- ✓ Past & Related Work
- Design and Implementation of Elements
- Experiments & Results
- Conclusion and Future Work

Design and Implementation of Elements

- Routing protocol
- Time synchronization protocol
- Sleep/Wakeup

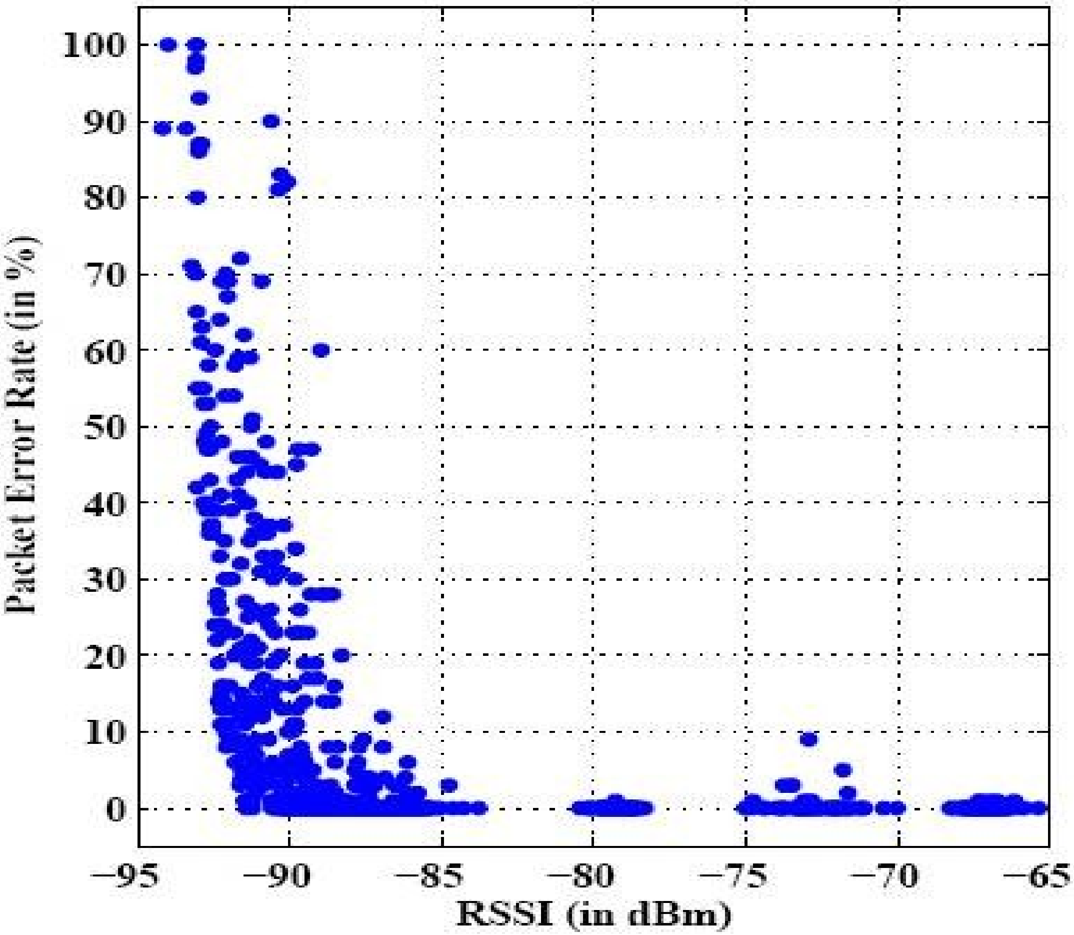
Design and Implementation of Elements

- Routing protocol
- Time synchronization protocol
- Sleep/Wakeup

Routing Protocol

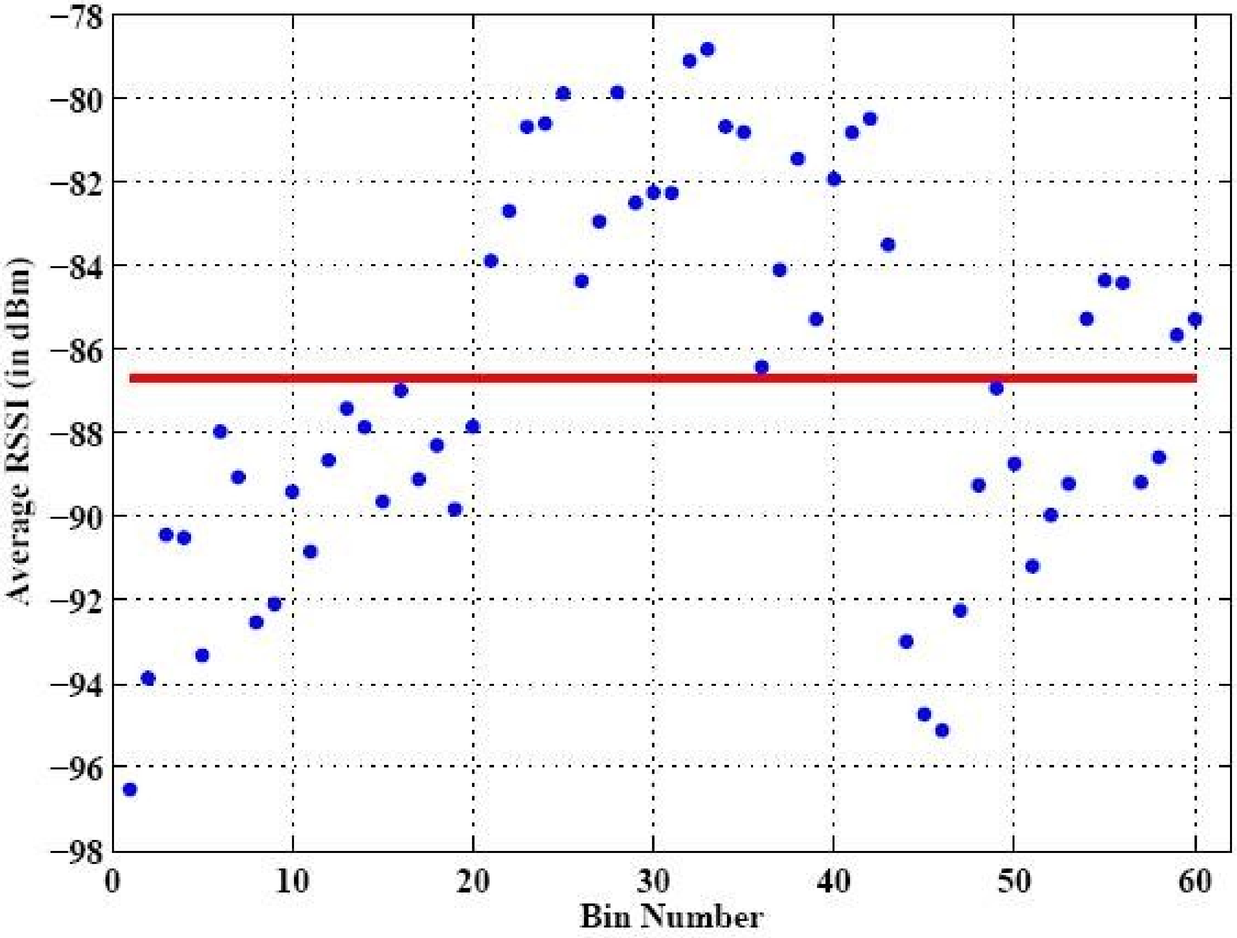
- Purpose – recap
- Basis – Link stability experiments
  - □ RSSI Vs Error rate [graph]
  - □ RSSI stability [graph]
  - □ Error rate stability [graph]
  - □ Takeaway
    - Below a threshold, RSSI experiences temporal variation
    - Temporal variation in error rate
    - Use stable links
- Challenges to be addressed
  - □ Node failures
  - □ Support for low duty-cycle
  - □ Children information at every node needed

RSSI Vs Error rate



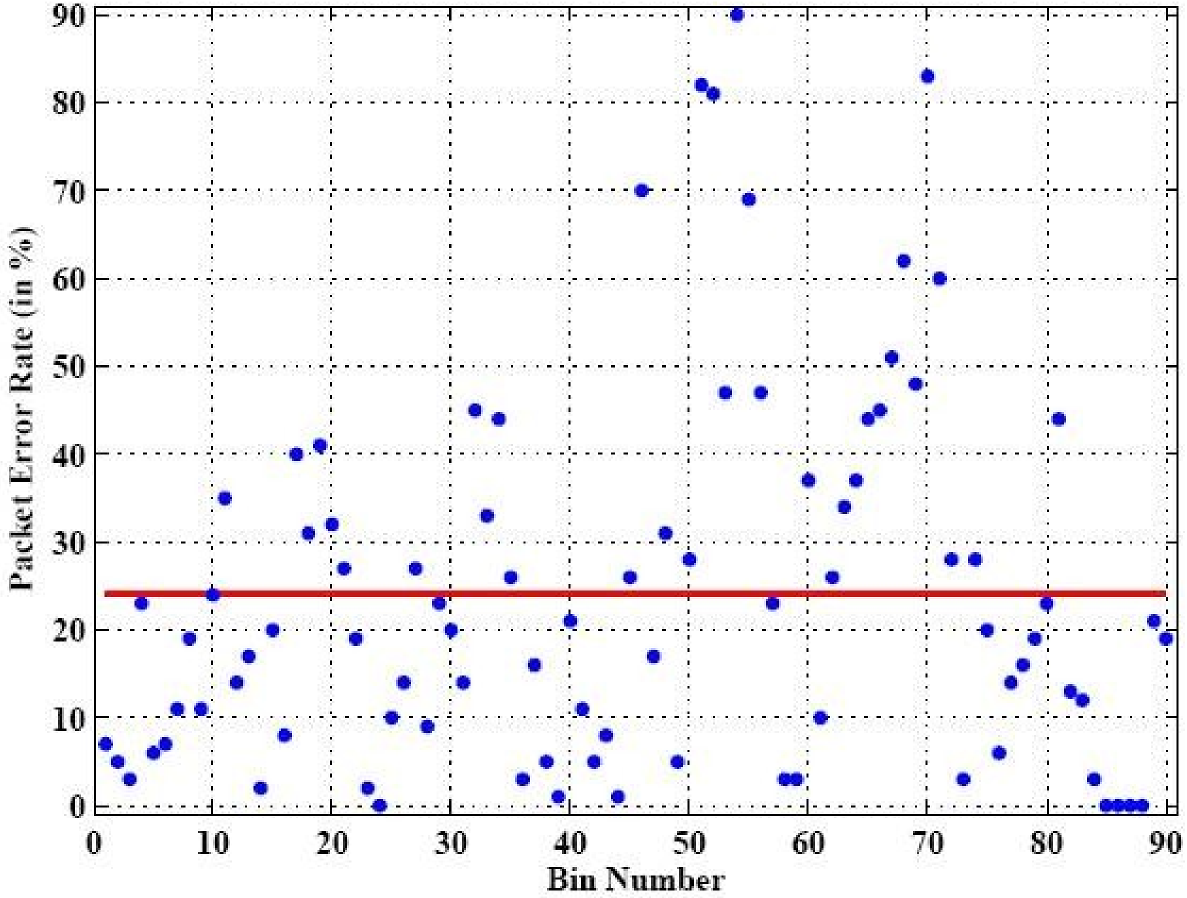
[Back]

RSSI stability



[Back]

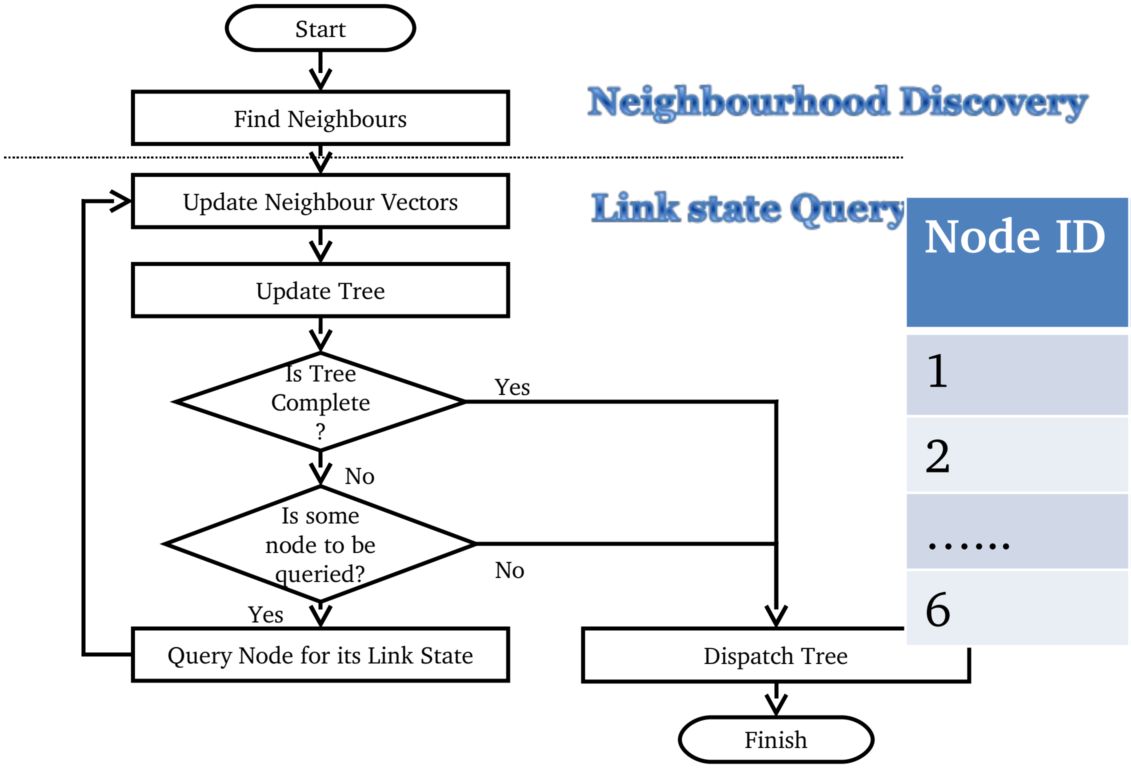
Error Rate Stability



[Back]

Routing Protocol

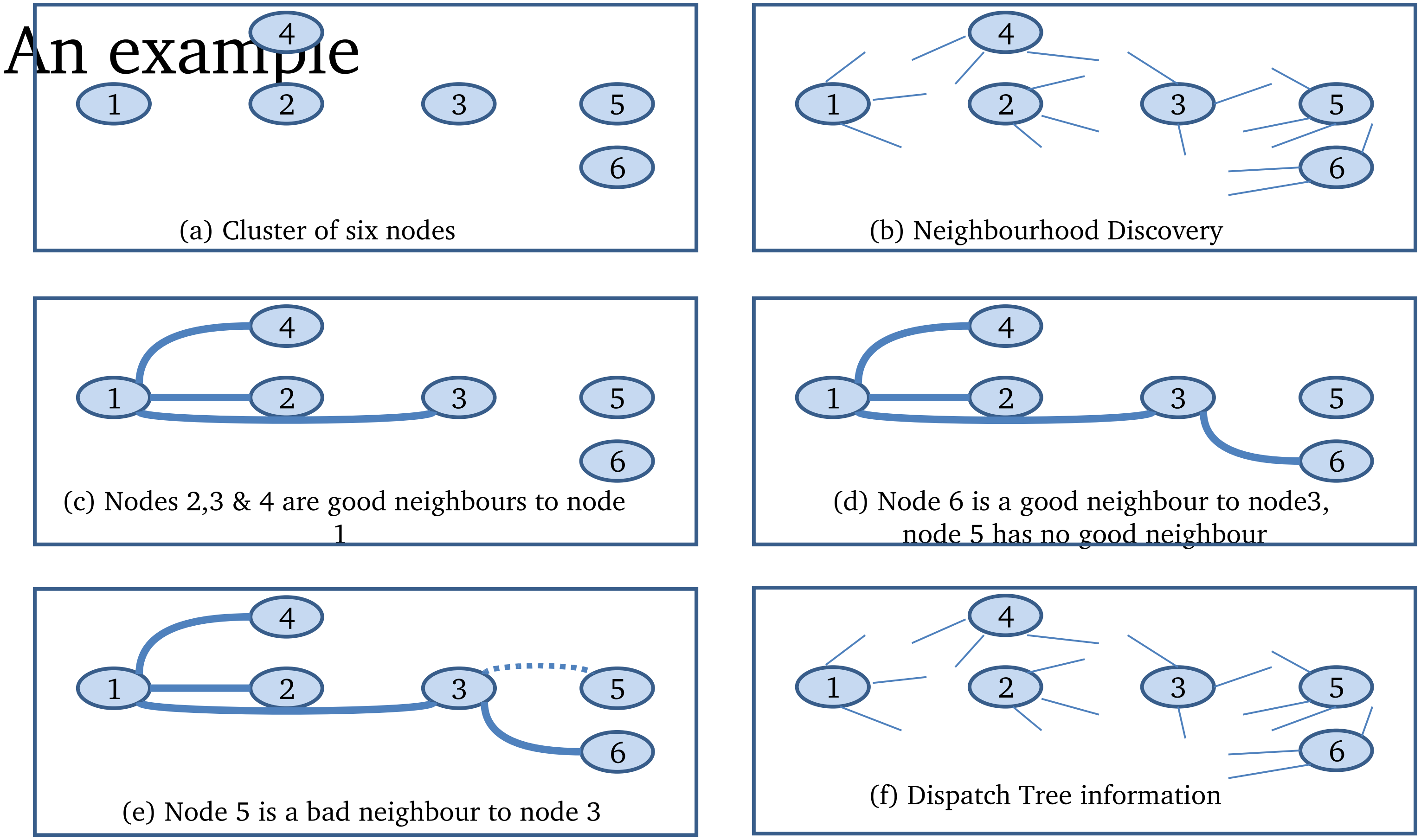
- Algorithm – *Centralized 2-Phase Routing*



Node ID	RSSI
1	-70 dBm
2	…..
……	……
6	……

Routing Protocol

- An example



(a) Cluster of six nodes

(b) Neighbourhood Discovery

(c) Nodes 2,3 & 4 are good neighbours to node 1

(d) Node 6 is a good neighbour to node3, node 5 has no good neighbour

(e) Node 5 is a bad neighbour to node 3

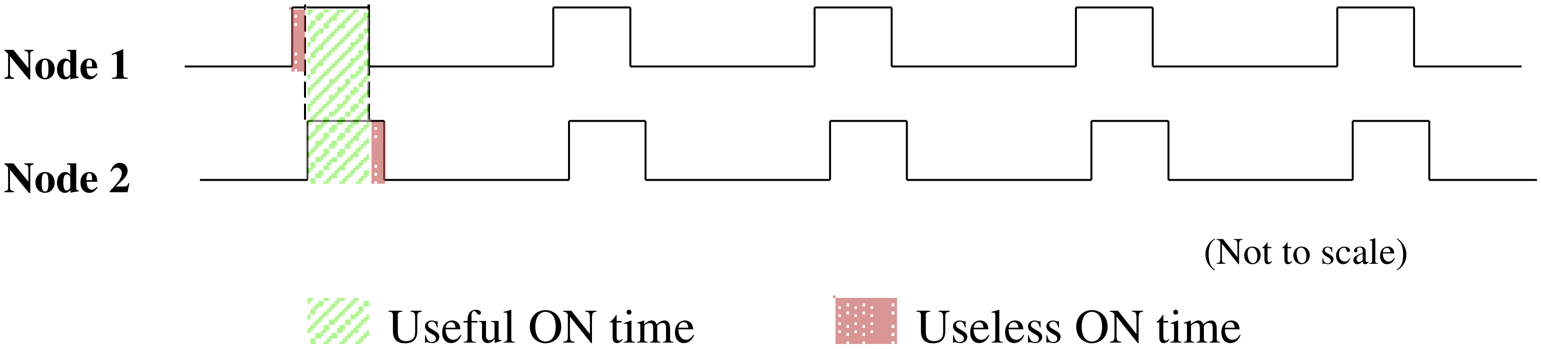
(f) Dispatch Tree information

Design and Implementation of Elements

- ✓ Routing protocol
- Time synchronization protocol
- Sleep/Wakeup

Time Synchronization Protocol

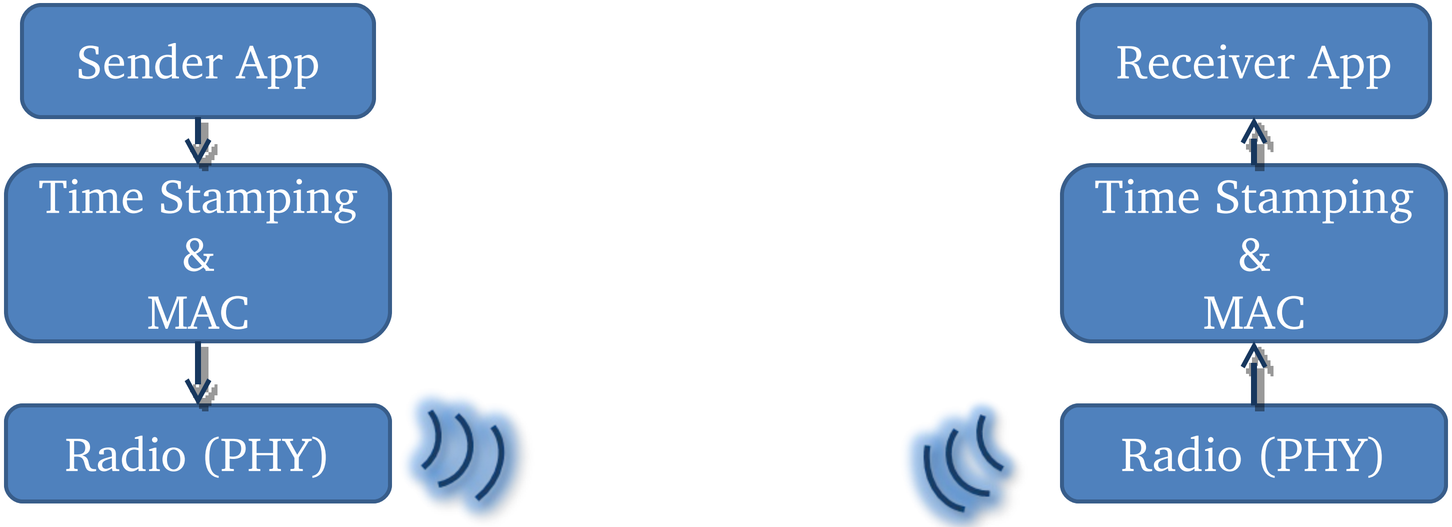
- Purpose – recap
  - □ Application need
  - □ Sleep/Wakeup



(Not to scale)

Time Synchronization Protocol

- Synchronizing two nodes

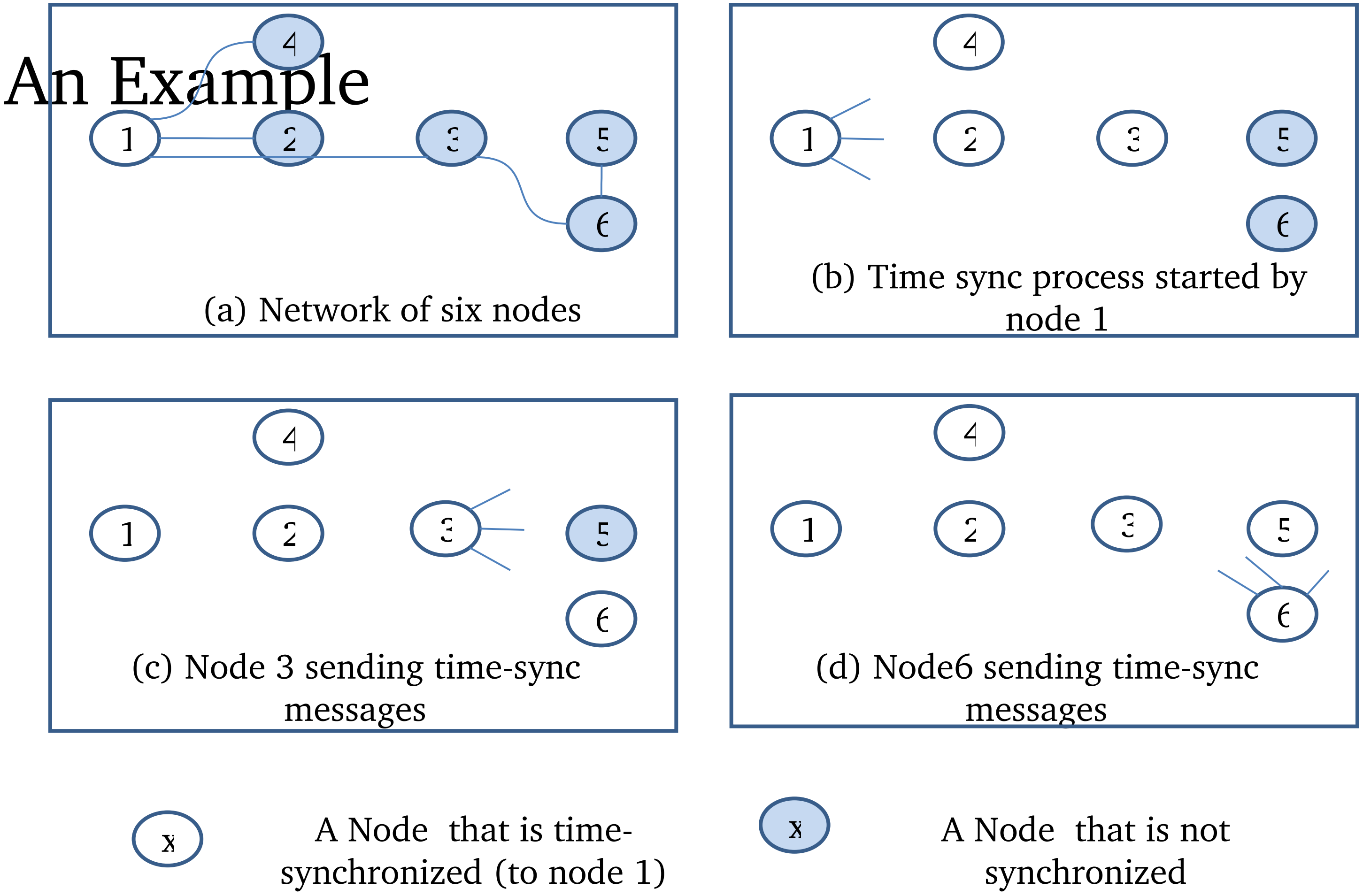


Time Synchronization Protocol

- Synchronizing a cluster
  - □ Problem with random flooding – Flow control
  - □ TDMA scheme
    - Nodes know about their slots – from routing tree
    - Synchronization on receiving *time-sync* message from parent
    - Only non-leaf nodes in the tree need to transmit time-sync messages

Time Synchronization Protocol

- An Example



(a) Network of six nodes

(b) Time sync process started by node 1

(c) Node 3 sending time-sync messages

(d) Node6 sending time-sync messages

A Node that is time-synchronized (to node 1)

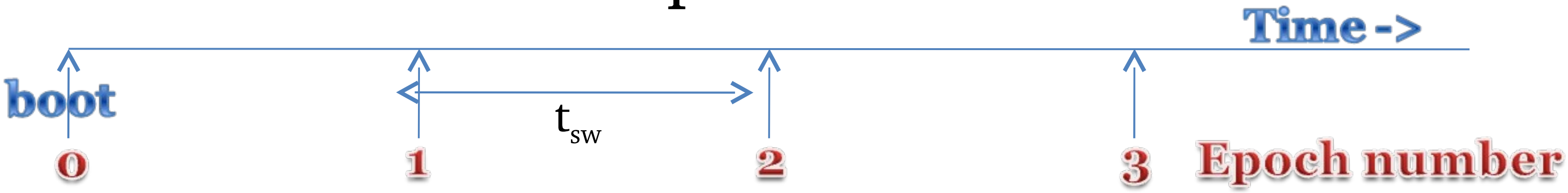
A Node that is not synchronized

Design and Implementation of Elements

- ✓ Routing protocol
- ✓ Time synchronization protocol
- Sleep/Wakeup

Sleep/Wakeup

- Division of time into Epochs



- Wakeup alarm timer value:

$$X = \left(\frac{currentGlobalTime}{SleepWakeupPeriod} + 1\right) * SleepWakeupPeriod - currentGlobalTime$$

$$currentGlobalTime = LocalTime + Offset.$$

- Wakeup times *should* be synchronized

Outline

- ✓ Introduction
- ✓ Thesis Contributions
- ✓ Architecture Overview
- ✓ Past & Related Work
- ✓ Design and Implementation of Elements
- Experiments & Results
- Conclusion and Future Work

Experiments and Results

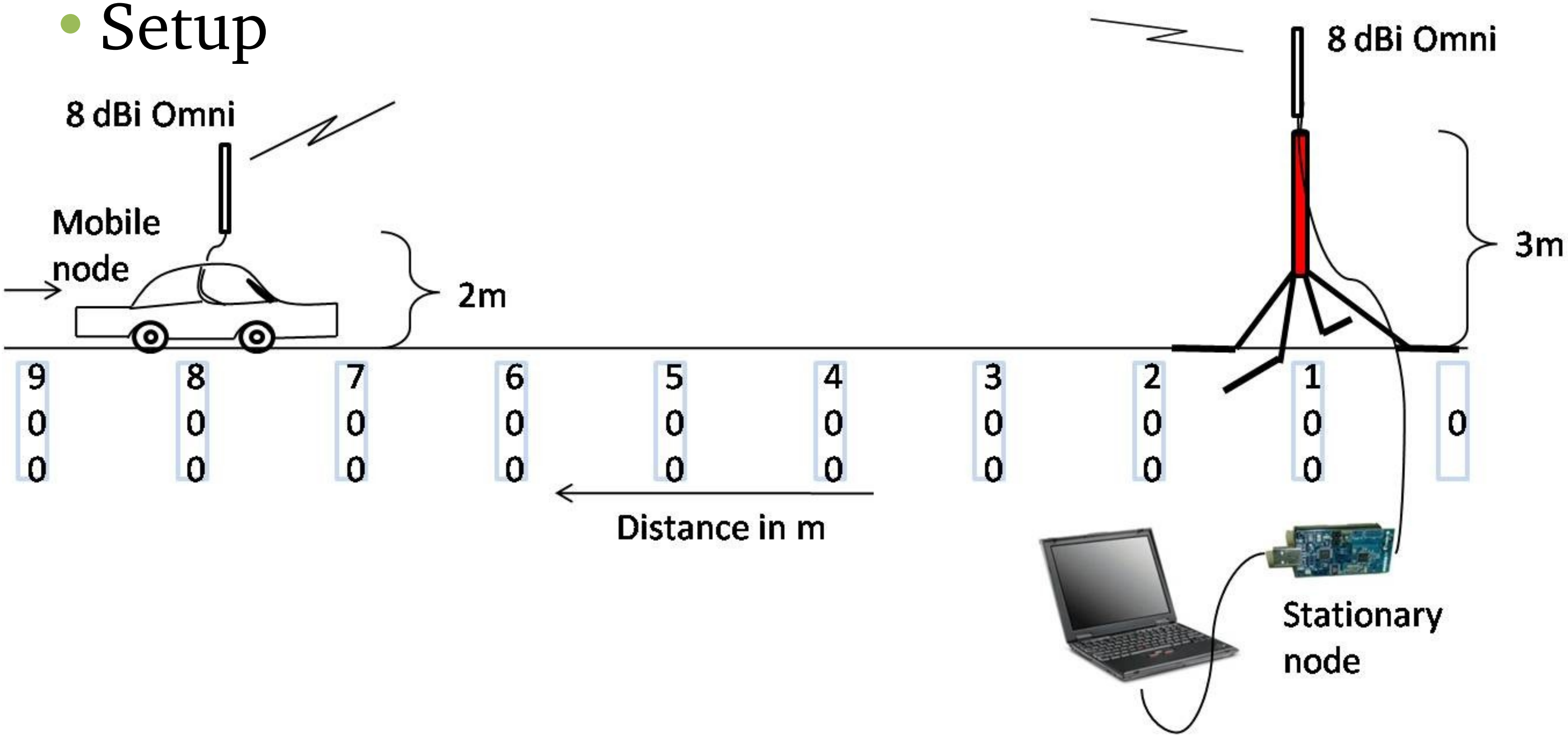
- Event detection experiments
- Evaluation of Routing protocol
- Evaluation of Time synchronization protocol

Experiments and Results

- Event detection experiments
- Evaluation of Routing protocol
- Evaluation of Time synchronization protocol

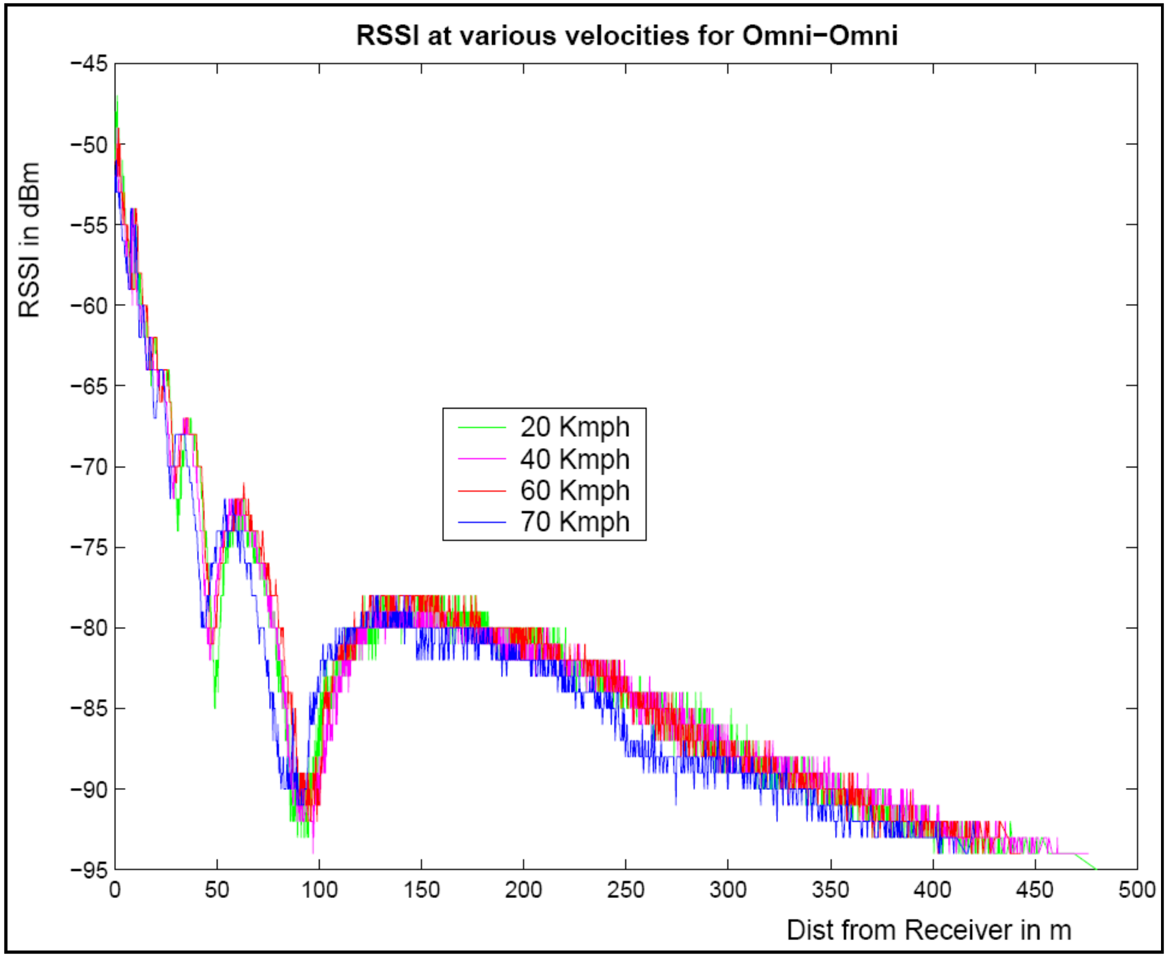
Event detection experiments

- Motive
- Setup



Event detection experiments

- Results
  - □ Maximum range
  - □ Results independent of speed



Event detection experiments

- Implication – Sleep Duration

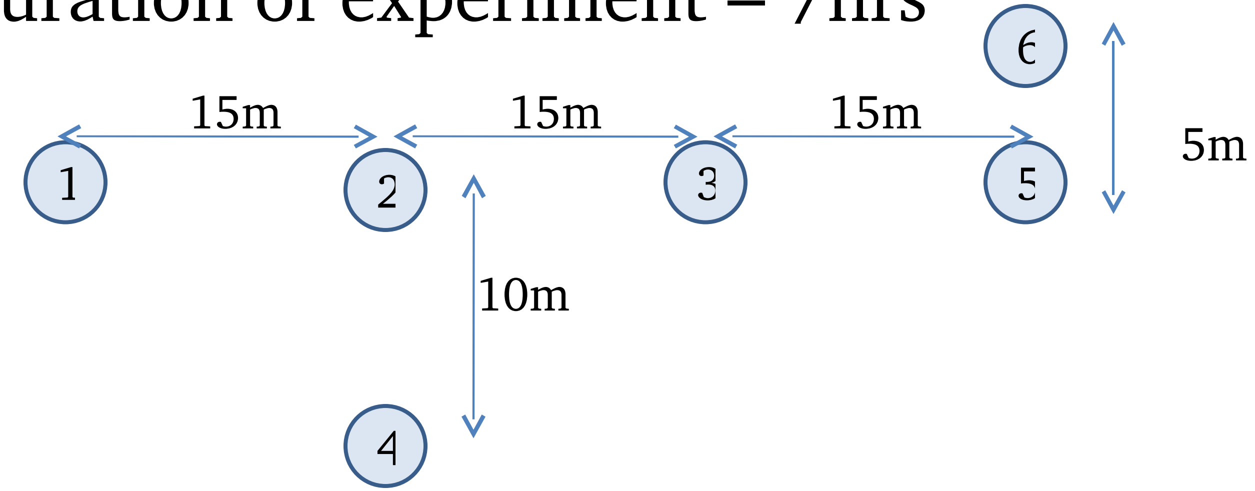
Event Signaler type	Max Range obtained	Sleep Duration at 54 KMPH	Sleep Duration at 80 KMPH
802.15.4	450 m	30.0 s	20.2 s
802.11	800 m	53.3 s	36 s

Experiments and Results

- ✓ Event detection experiments
- Evaluation of Routing protocol
- Evaluation of Time synchronization protocol

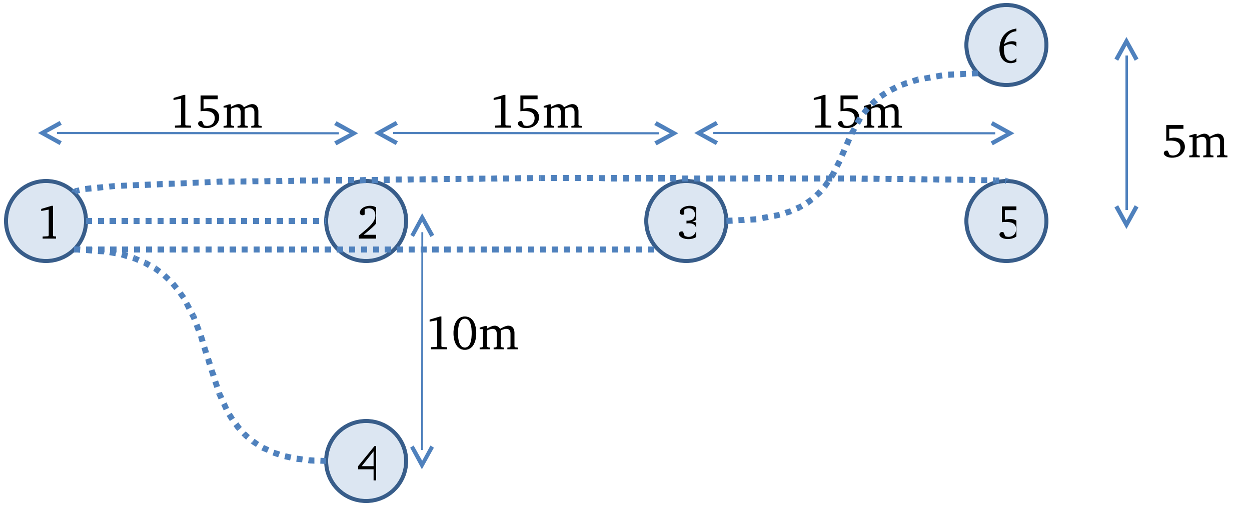
Evaluation of Routing protocol

- Motive
- Setup
- Threshold of -75dBm for good links
- Duration of experiment – 7hrs



Evaluation of Routing protocol

- Frequently formed tree



Most frequently formed links during routing

Evaluation of Routing protocol

- Results & Implications
  - □ Same topology for 99% of the time
    - □ Infrequent routing would suffice
  - □ All the links are good (RSSI > -75dBm)
    - □ Error rates close to 0%
  - □ Most of the nodes have direct link to root
    - □ Low values of t_{ps}
  - □ Average duration of routing : 567ms
    Maximum duration : 819ms, Std Dev : 51ms
    - □ Supports low duty cycle operation

Experiments and Results

✓ Event detection experiments

✓ Evaluation of Routing protocol

- Evaluation of Time synchronization protocol

Evaluation of Time synchronization protocol

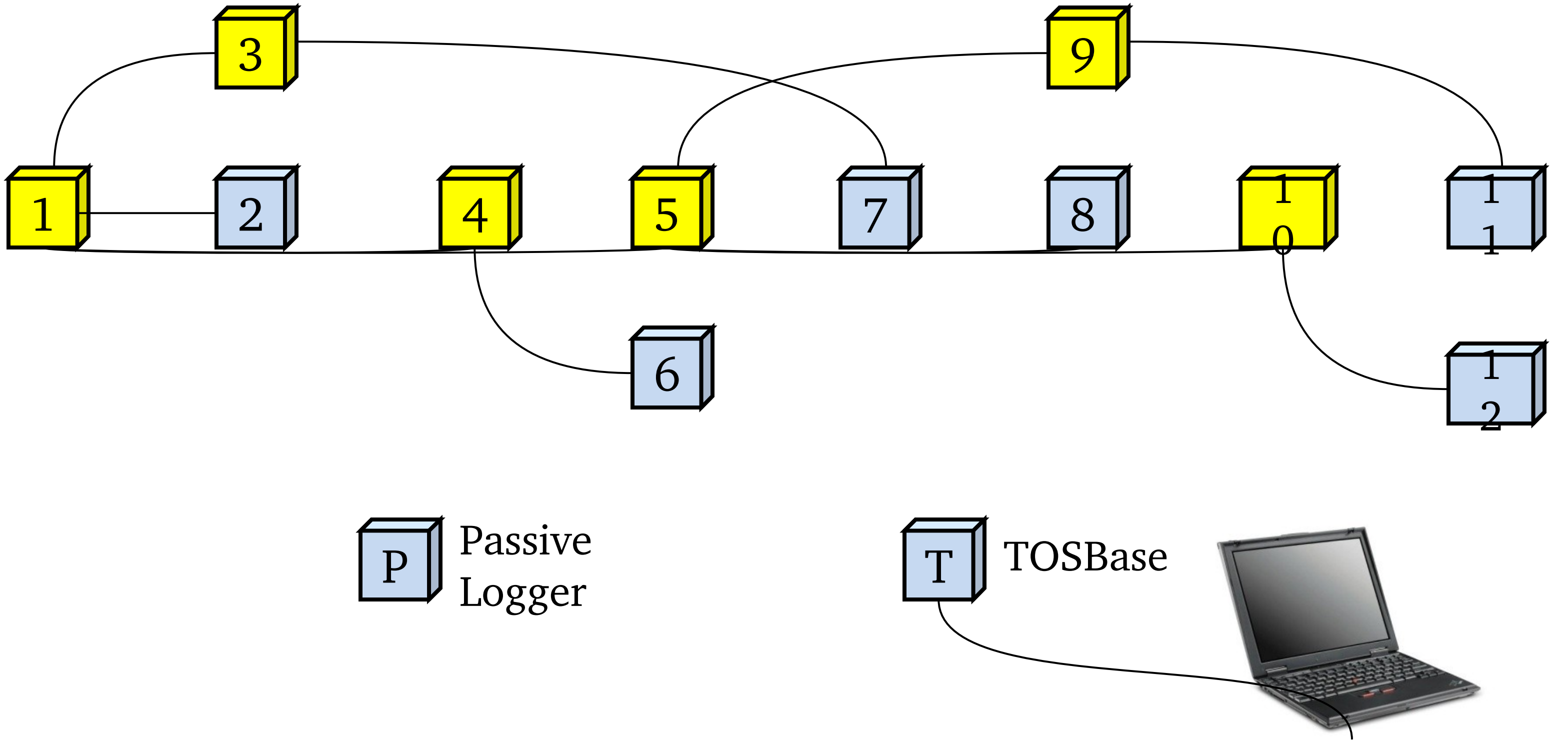
- Measurement of t_{ps}
- Measurement of t_{Δ}

Evaluation of Time synchronization protocol

- Measurement of t_{ps}
- Measurement of t_{Δ}

Measurement of t_{ps}

- Setup



Measurement of t_{ps}

- 6 non-leaf nodes
- Slot time 12ms
- 3 *time-sync* messages from each node
- t_{ps} - 80ms
- $t_{pc} = t_{ps} = 80\text{ms}$

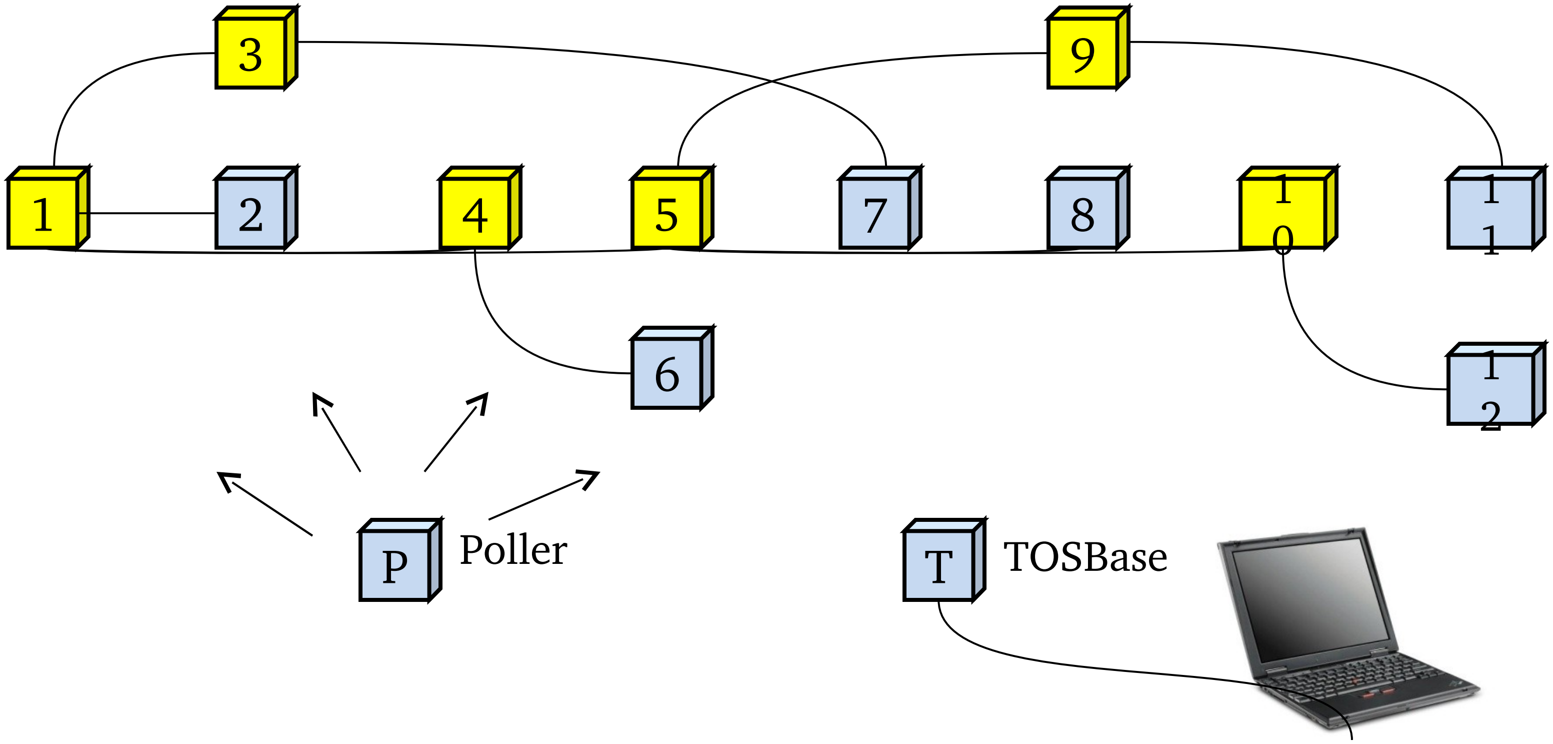
Sender Address	Time Stamp at Logger (in seconds)	Time after start (in seconds)
1	1330.662811	0.000000
1	1330.667725	0.004914
1	1330.672394	0.009583
5	1330.677185	0.014374
5	1330.681915	0.019104
5	1330.686432	0.023621
10	1330.691284	0.028473
10	1330.695953	0.033142
10	1330.700684	0.037873
9	1330.705353	0.042542
9	1330.710083	0.047272
9	1330.714783	0.051972
4	1330.719604	0.056793
4	1330.724426	0.061615
4	1330.729126	0.066315
3	1330.733826	0.071015
3	1330.738403	0.075592
3	1330.742737	0.079926

Evaluation of Time synchronization protocol

- ✓ Measurement of t_{ps}
- Measurement of t_{Δ}

Measurement of t_Δ

- Setup



Measurement of t_Δ

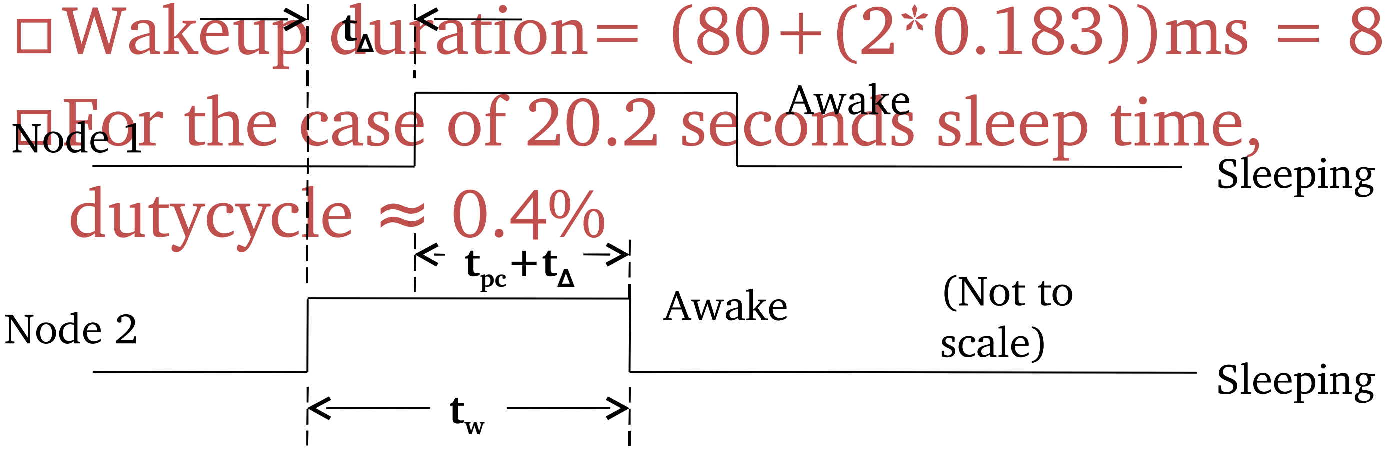
- Slot time of 12ms
- 3 *time-sync* messages from each node
- Tree depth of 3
- t_Δ - 0.183 ms

= 6 clock ticks of 32KHz clock

Node	Global Time Stamp (in seconds)	Node's Local Time (in seconds)
1	24.841461	24.841461
2	**24.841431**	24.812500
3	24.841522	23.534668
4	24.841461	24.792267
5	24.841492	23.577576
6	24.841492	24.845734
7	24.841431	24.816315
8	24.841553	17.672333
9	24.841522	16.461029
10	24.841492	17.670959
11	**24.841614**	17.658264
12	24.841461	17.631683

Evaluation of Time synchronization protocol

- ✓ Measurement of t_{ps}
- ✓ Measurement of t_{Δ}
- Implication on duty-cycling
  - □ Wakeup duration= (80+(2*0.183))ms = 80ms
  - □ For the case of 20.2 seconds sleep time, dutycycle ≈ 0.4%



t_{Δ} : Time Sync error

t_{pc} : Time for command propagation

t_w : Awake duration

Outline

- ✓ Introduction
- ✓ Thesis Contributions
- ✓ Architecture Overview
- ✓ Past & Related Work
- ✓ Design and Implementation of Elements
- ✓ Experiments & Results
- Conclusion and Future Work

Conclusion and Future Work

- Protocol set – generic
  - Design
  - Implementation
  - Experimental Evaluation
- Event Detection
  - Implementation
  - Experimental Evaluation
- Sleep/Wakeup – implementation

Conclusion and Future Work

- Event Detection – Speed of train
- Ambient vibrations
- Earth-quake vibrations

Questions

Thank you !!