Never replicate a successful experiment

-Fett's law.

A volcano is a mountain with hiccups strong enough to blow things away…

Authors:

Geoffrey Werner-Allen, Konrad Lorincz, and Matt Welsh
Harvard University
Jeff Johnson
University of New Hampshire
Jonathan Lees
University of North Carolina

Presentation By*:
Rahul Potharaju
EECS 494: Distributed Systems in Challenging Environments
* Presentation inspired from the original version

My Assumptions

Everyone knows what a volcano is

Everyone's familiar with sensor networks

Ok… Even if you're not, you'll realize it in a few minutes!

What we are going to cover…

A small overview

System Architecture

Network Robustness

Event Detector Accuracy

Data Collection Performance

Time Rectification & Accuracy

Data Fidelity

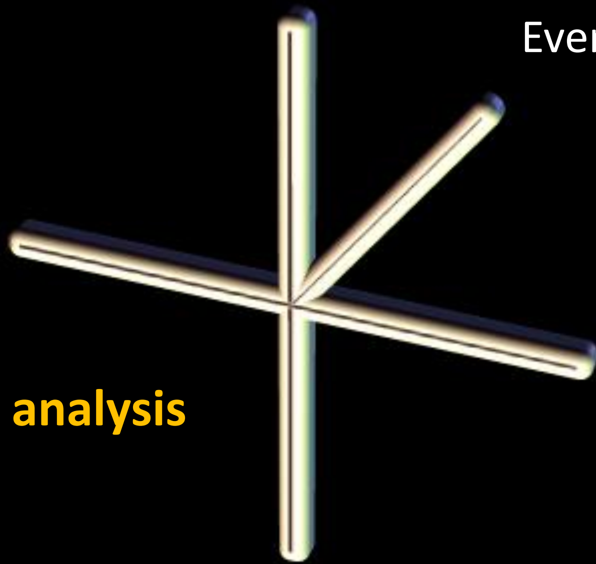
Summary

Overview of the Experiment

Evaluation of data collected from a **19-day** field deployment of **16 wireless sensors**

Collected data for hundreds of earthquakes, eruptions and tremor events

Location: Reventador volcano, Ecuador



The five axes of analysis

Overview of the Existing Scenario

Why monitor volcanoes in the first place?

to monitor hazards by assessing the level of volcanic unrest

to understand physical processes occurring within the volcano (magma migration)

Instrument used:

Seismometer → Measures ground propagating elastic radiation from source internal to volcano and on the surface

Existing Volcano instrumentation:

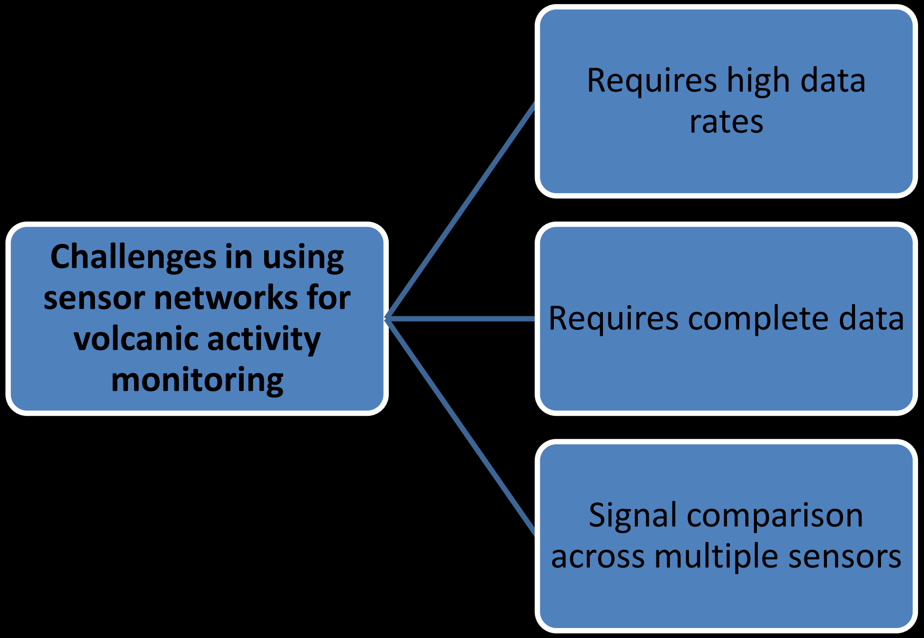
Standalone data loggers → record on a flash drive BUT power hungry, bulky

Overview of various techniques

Challenges in today's sensor networks:

- Node failure
- Message Failure
- Sensor Calibration
- Inaccurate Time Synchronization

Typical Volcano Studies employ: GPS synchronized data loggers



Why should we bother about it anyways?

We are concerned with a few typical challenges as students of EECS

High data rate sampling

- Seismometers and microphones typically sampled at 100 Hz, 24 bits per channel
- Total data rate **exceeds** radio bandwidth!

Automatic seismic event detection

- Detect signals of interest – earthquakes, eruptions, etc.
- Detector uses a ratio of two EWMA filters (Exponentially Weighted Moving Average)

Reliable data collection

- Download 60 sec. of data from each node following a detected event
- Uses a reliable multi-hop data transfer protocol, called Fetch

Fine grained time synchronization

- Must correlate signals across the network (e.g., for computing wave arrival times)
- Requires accuracy of one sample time (10 ms)
- Uses the Vanderbilt FTSP1 protocol and a single GPS receiver

What we are going to cover…

A small overview

System Architecture

Network Robustness

Event Detector Accuracy

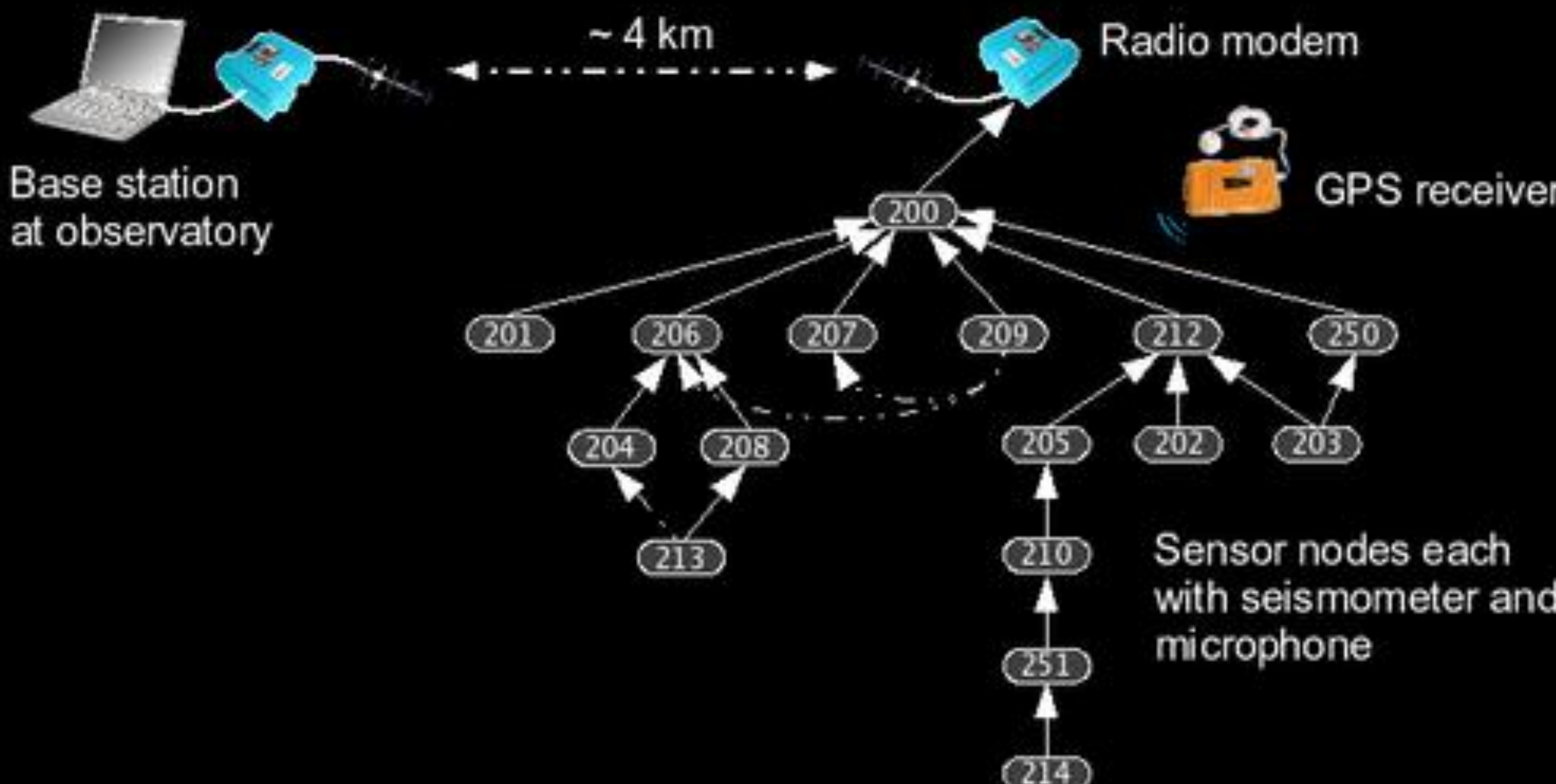
Data Collection Performance

Time Rectification & Accuracy

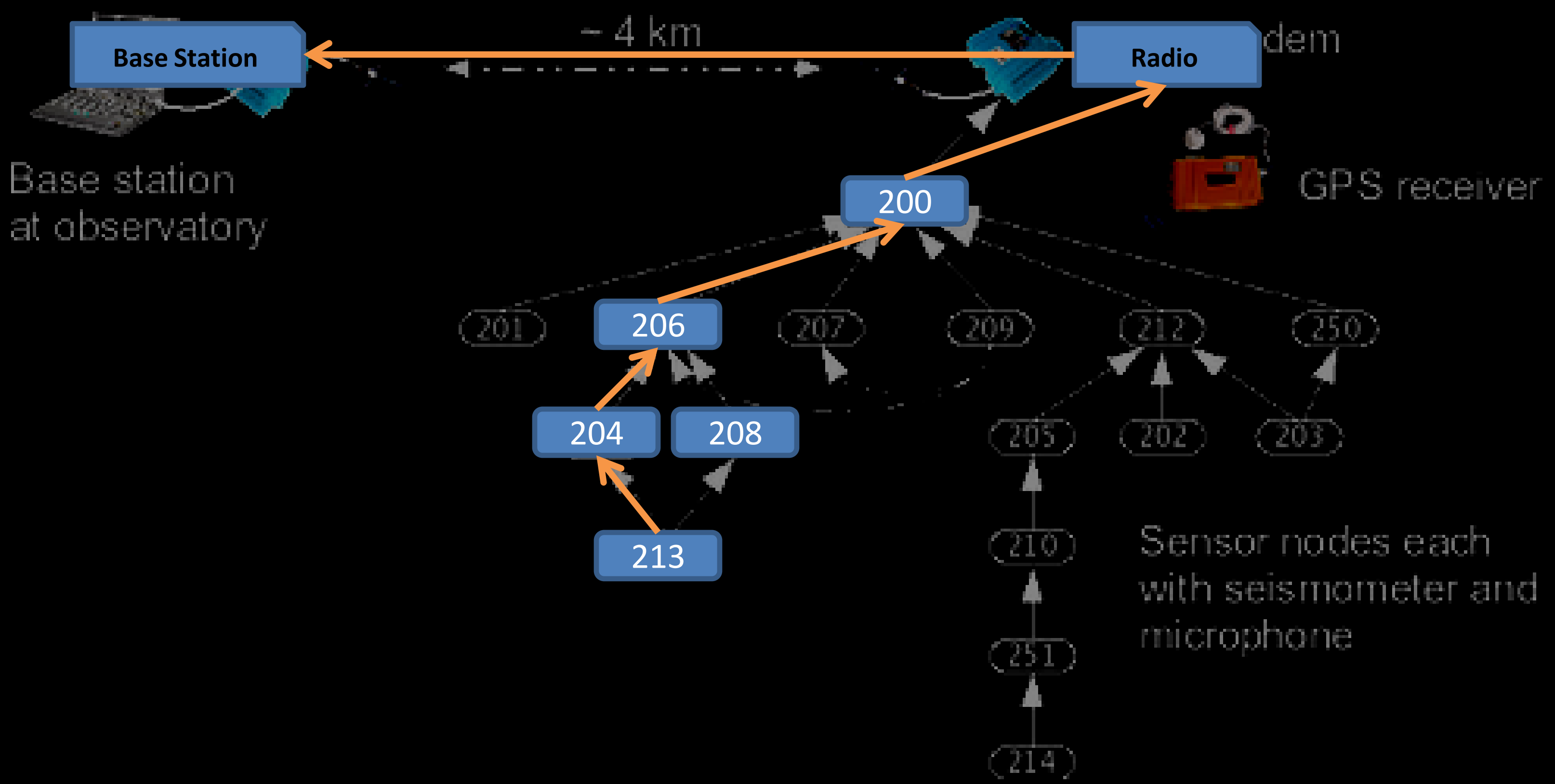
Data Fidelity

Summary

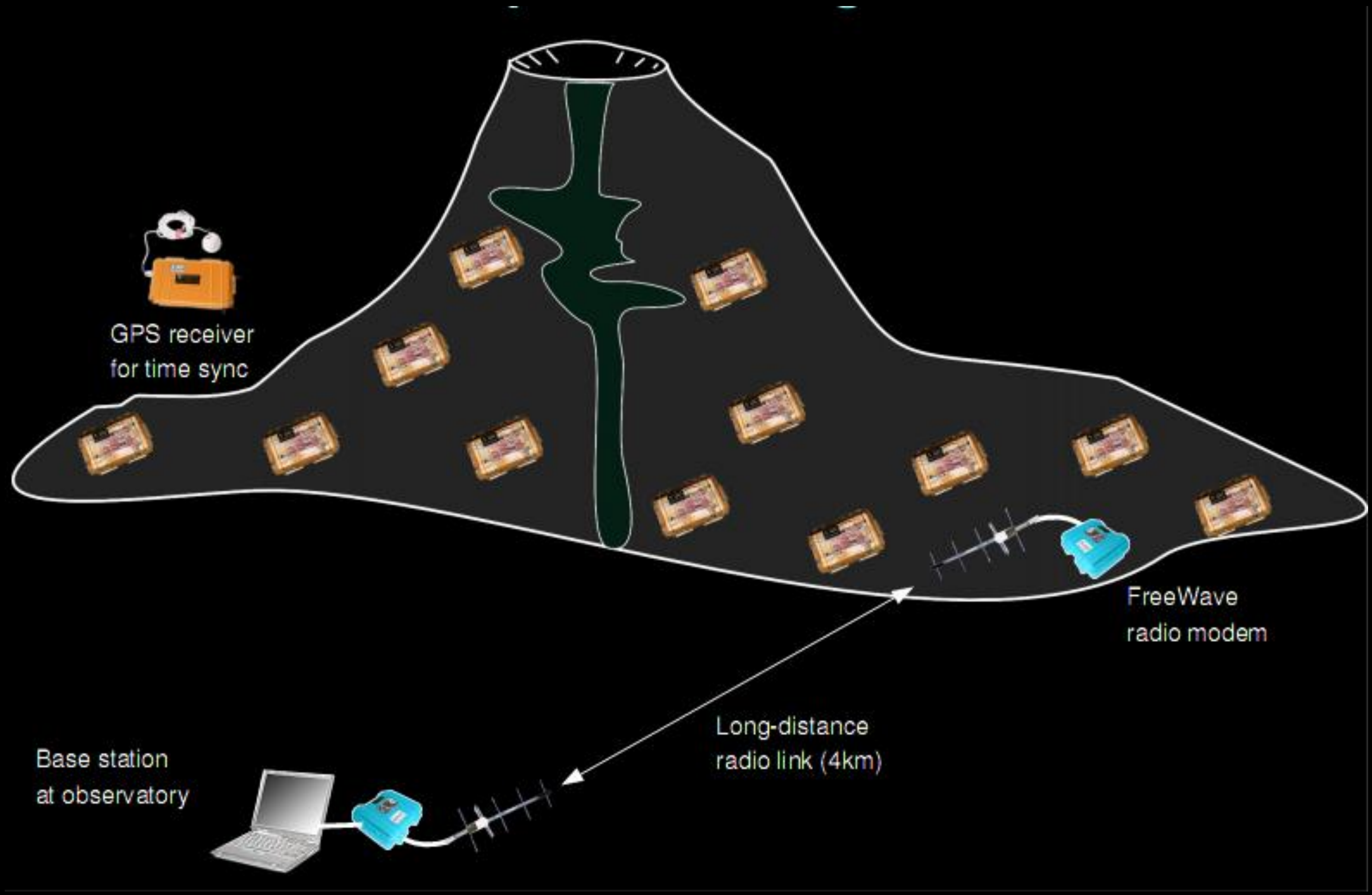
System Architecture



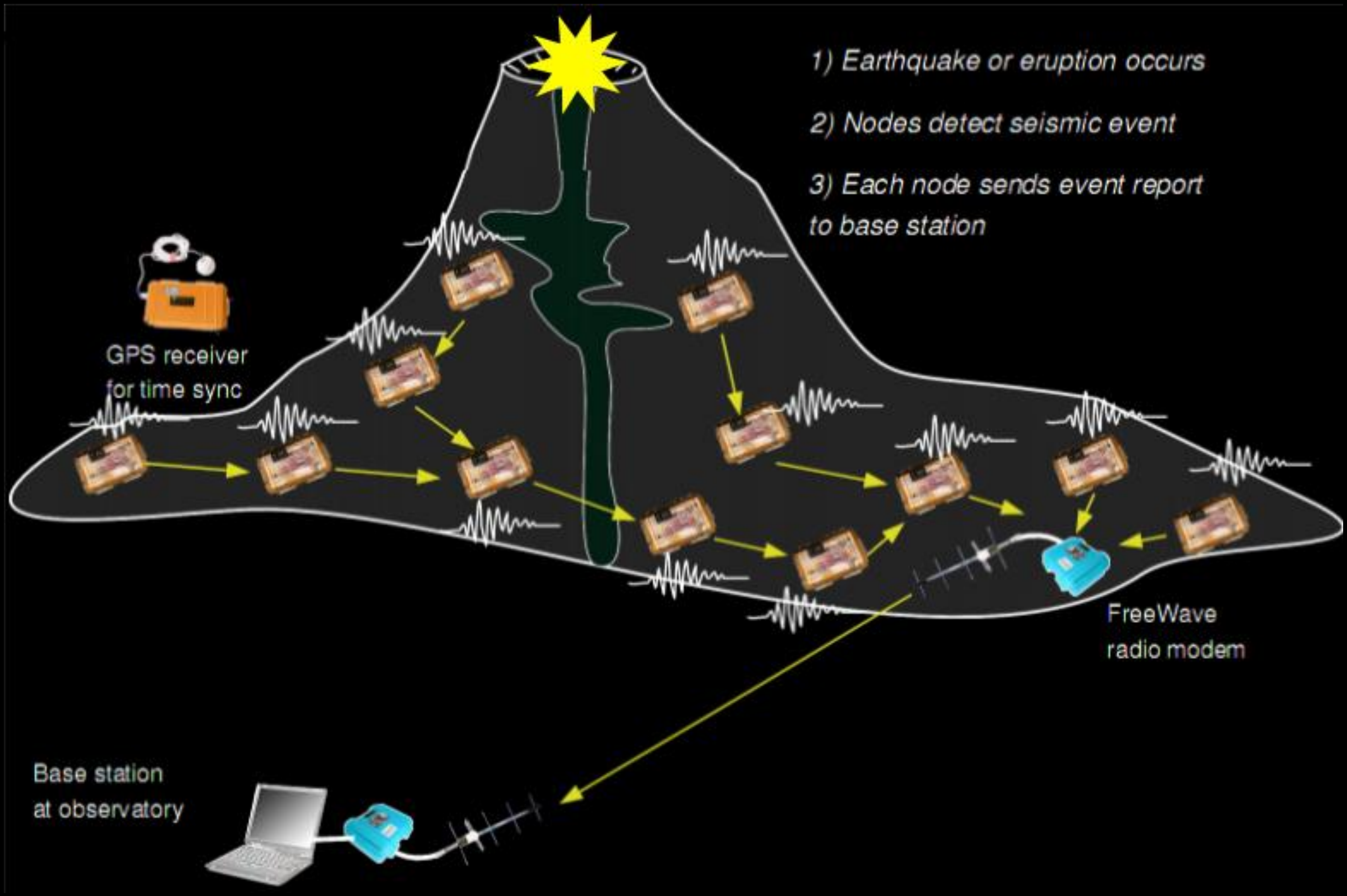
System Architecture



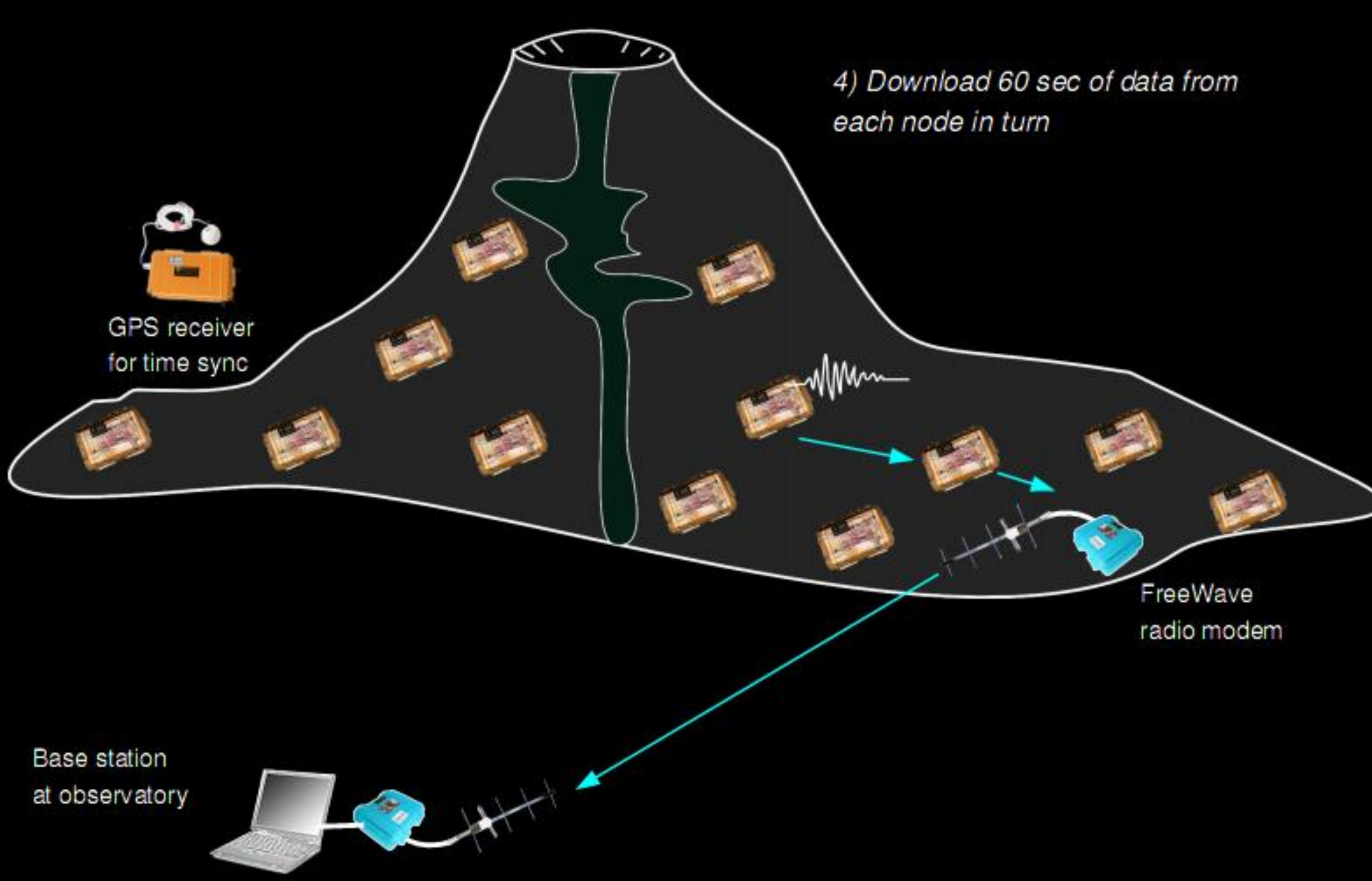
System Architecture



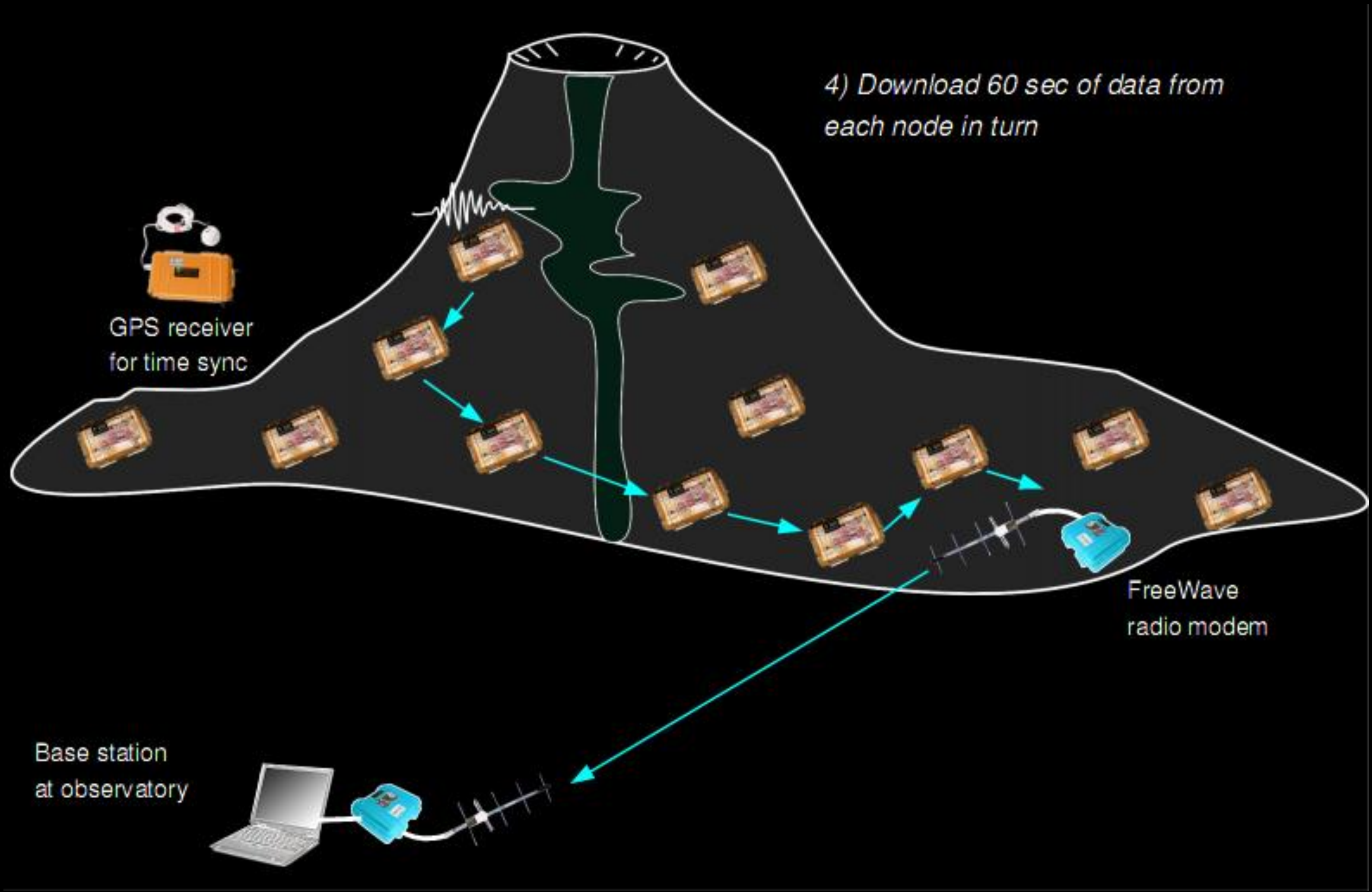
System Architecture



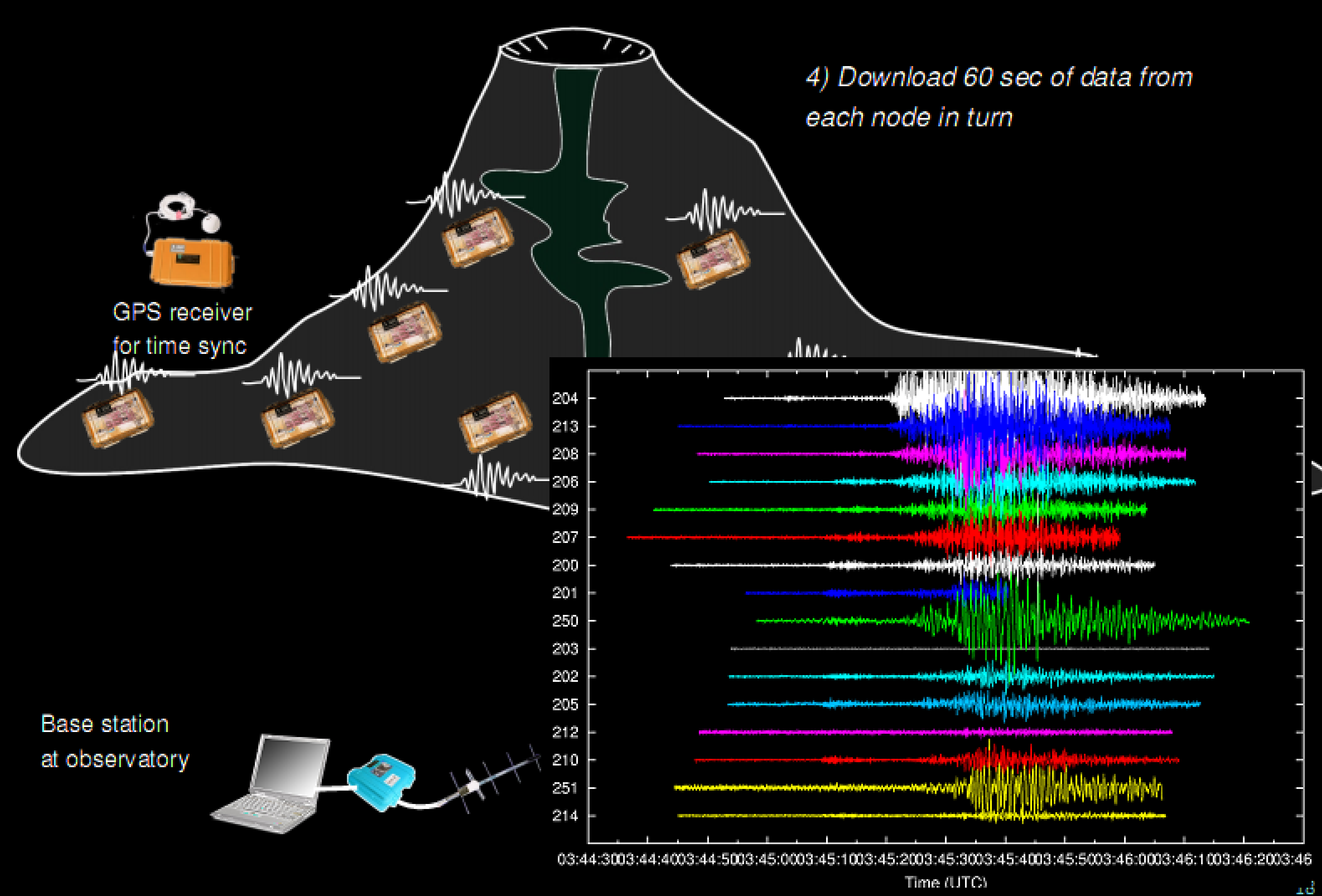
System Architecture



System Architecture



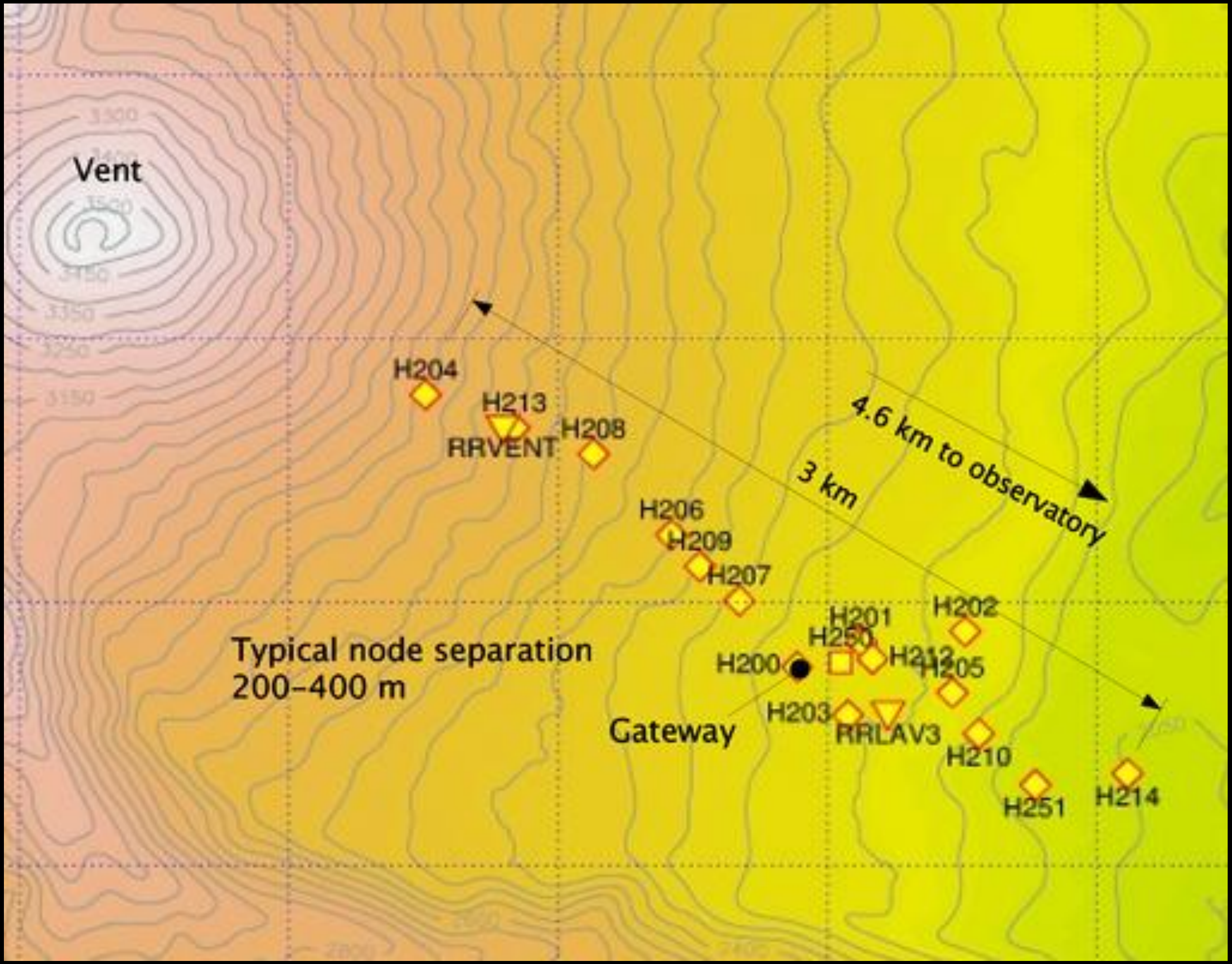
System Architecture



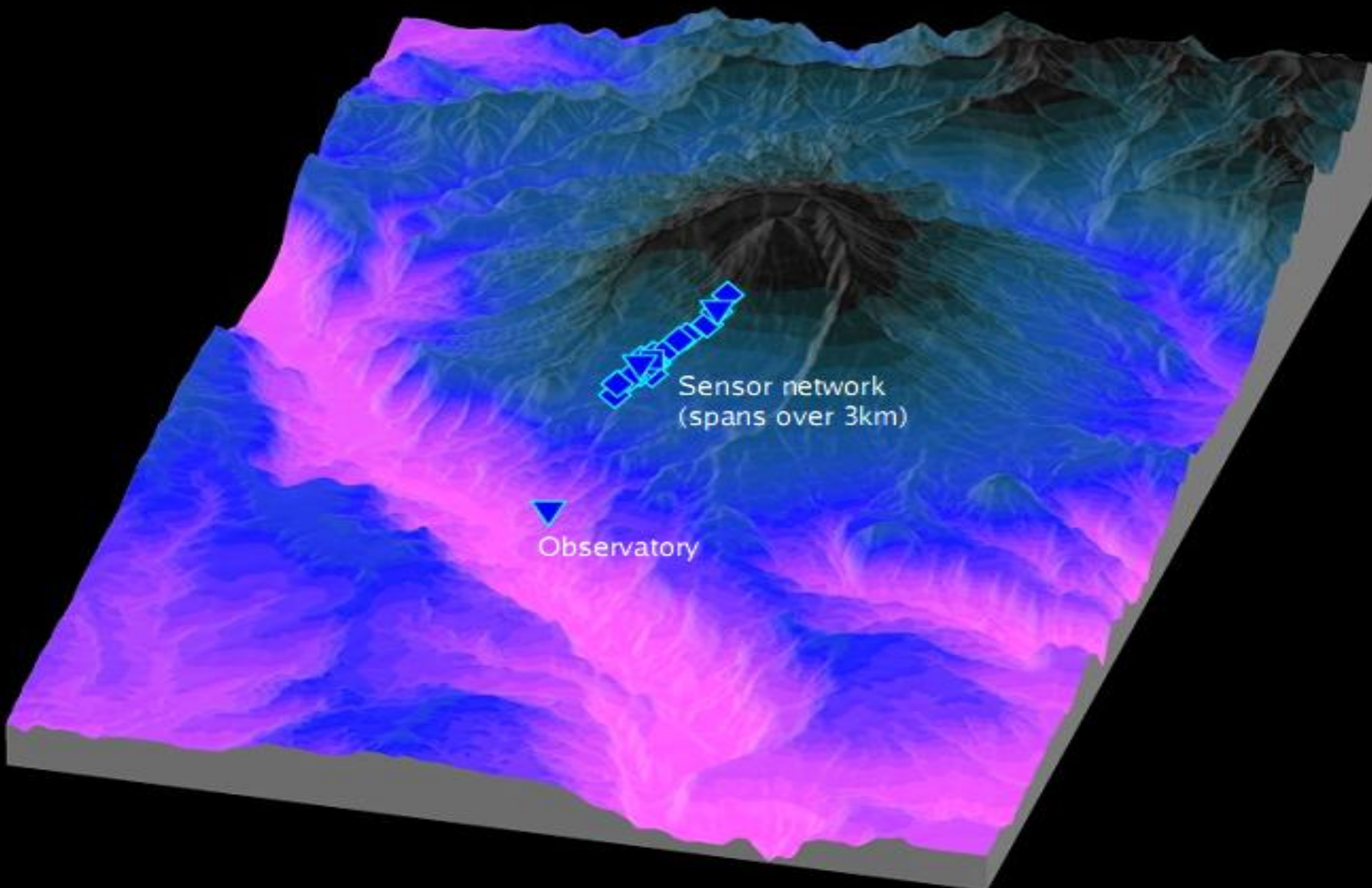
Deployment Site: Reventador Volcano, Ecuador

Deployment Site: Team responsible

Sensor Deployment Map



Sensor Deployment Map



If an experiment works, something has gone wrong.

What we are going to cover…

A small overview

System Architecture

Network Robustness

Event Detector Accuracy

Data Collection Performance

Time Rectification & Accuracy

Data Fidelity

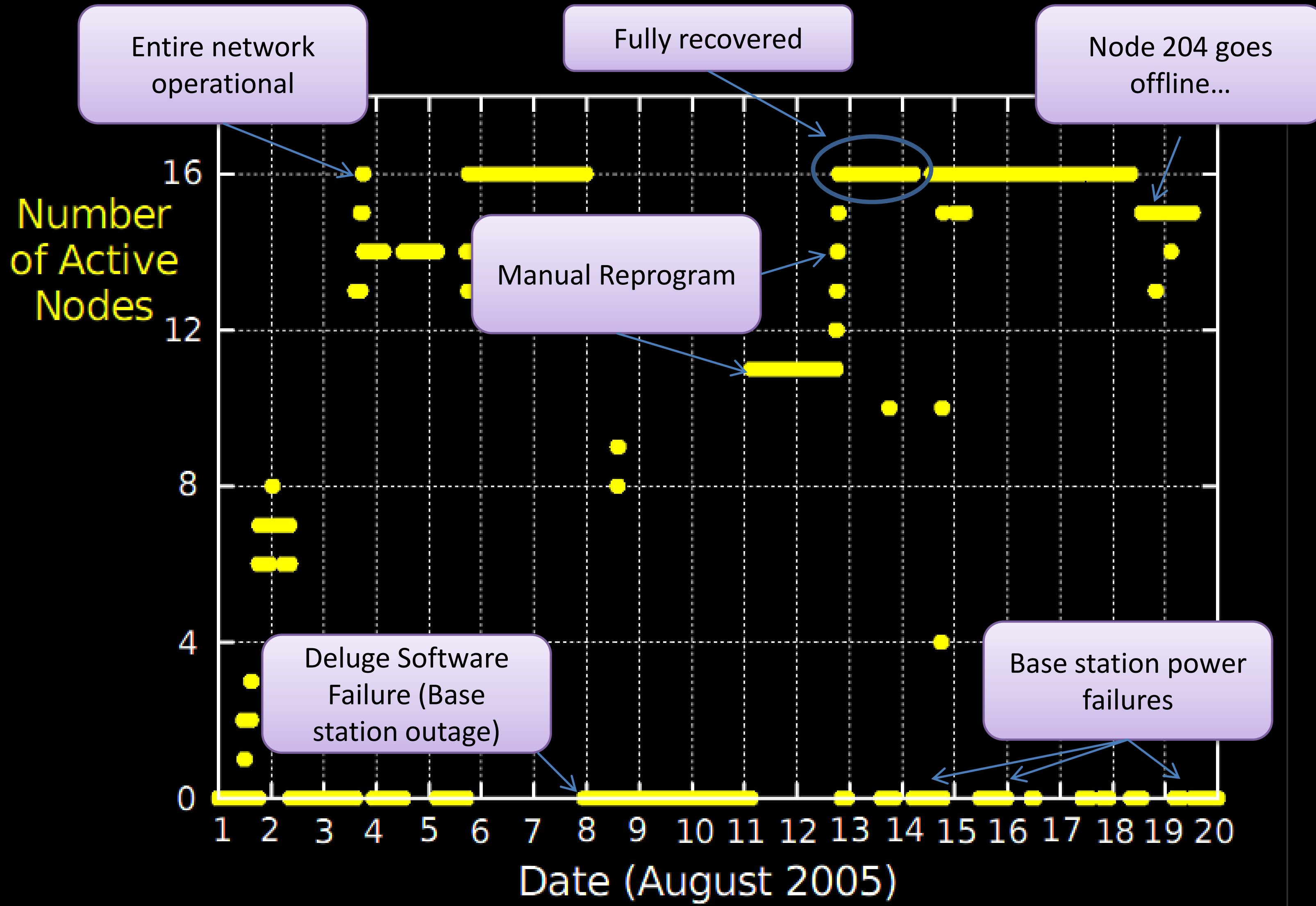
Summary

So what went wrong?

Nodes exhibited high uptimes but the base station was giving problems – something that was unexpected. Why?

Reason: A single bug affecting the deluge protocol caused a three-day outage of the entire network

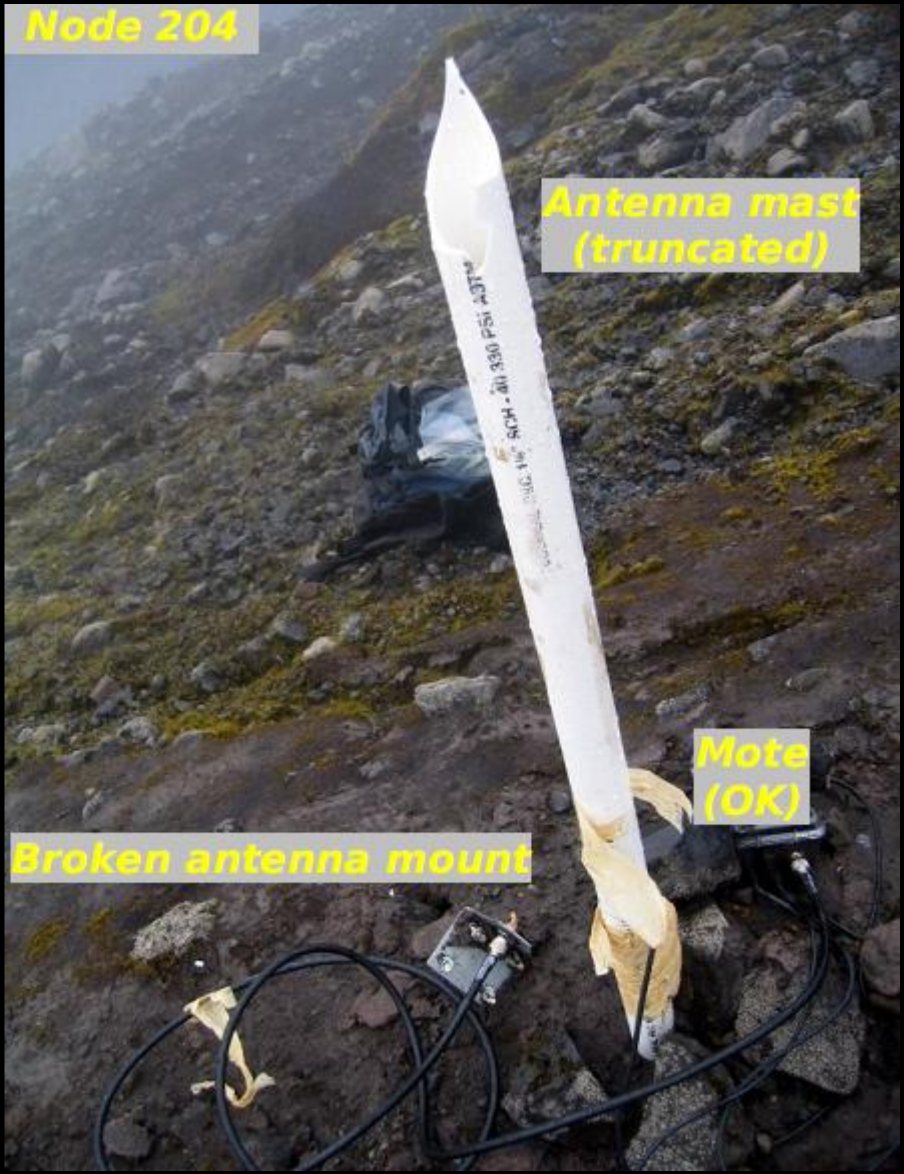
Overall Network Uptime



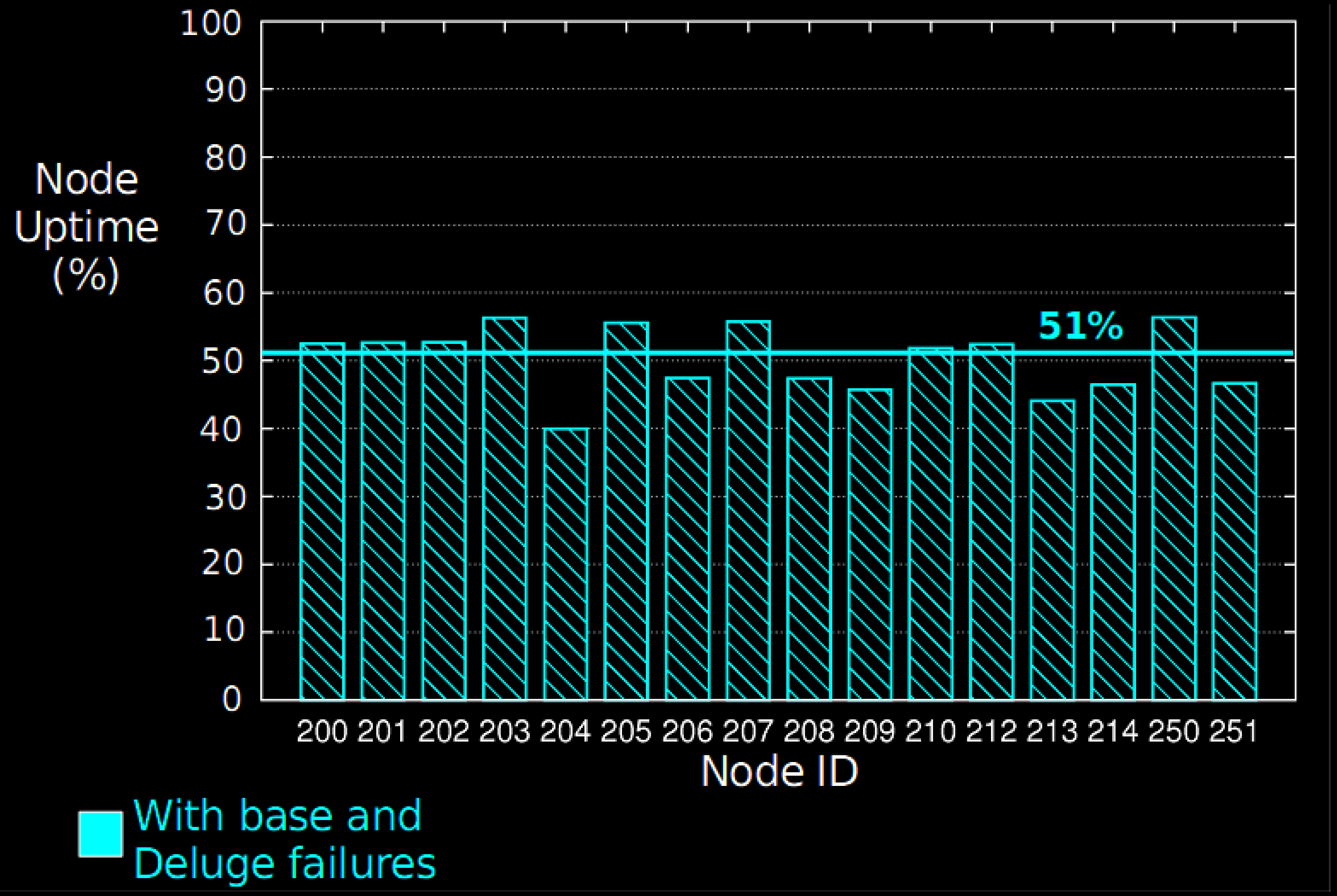
What happened to Node 204?

It was closest to the volcano vent

The funny thing was that the node remained intact but the antenna mast was destroyed (by a bomb?)

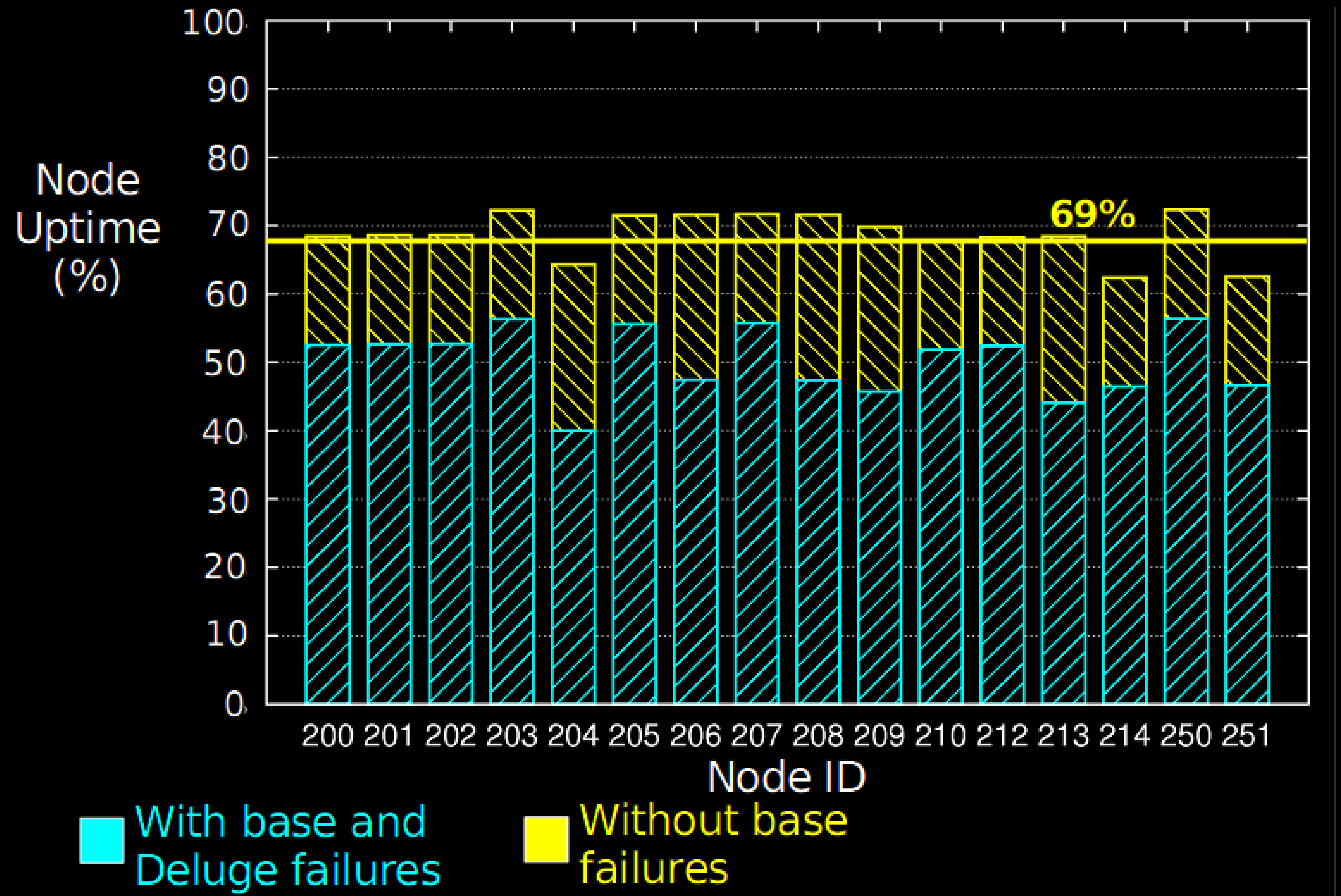


Node Uptime

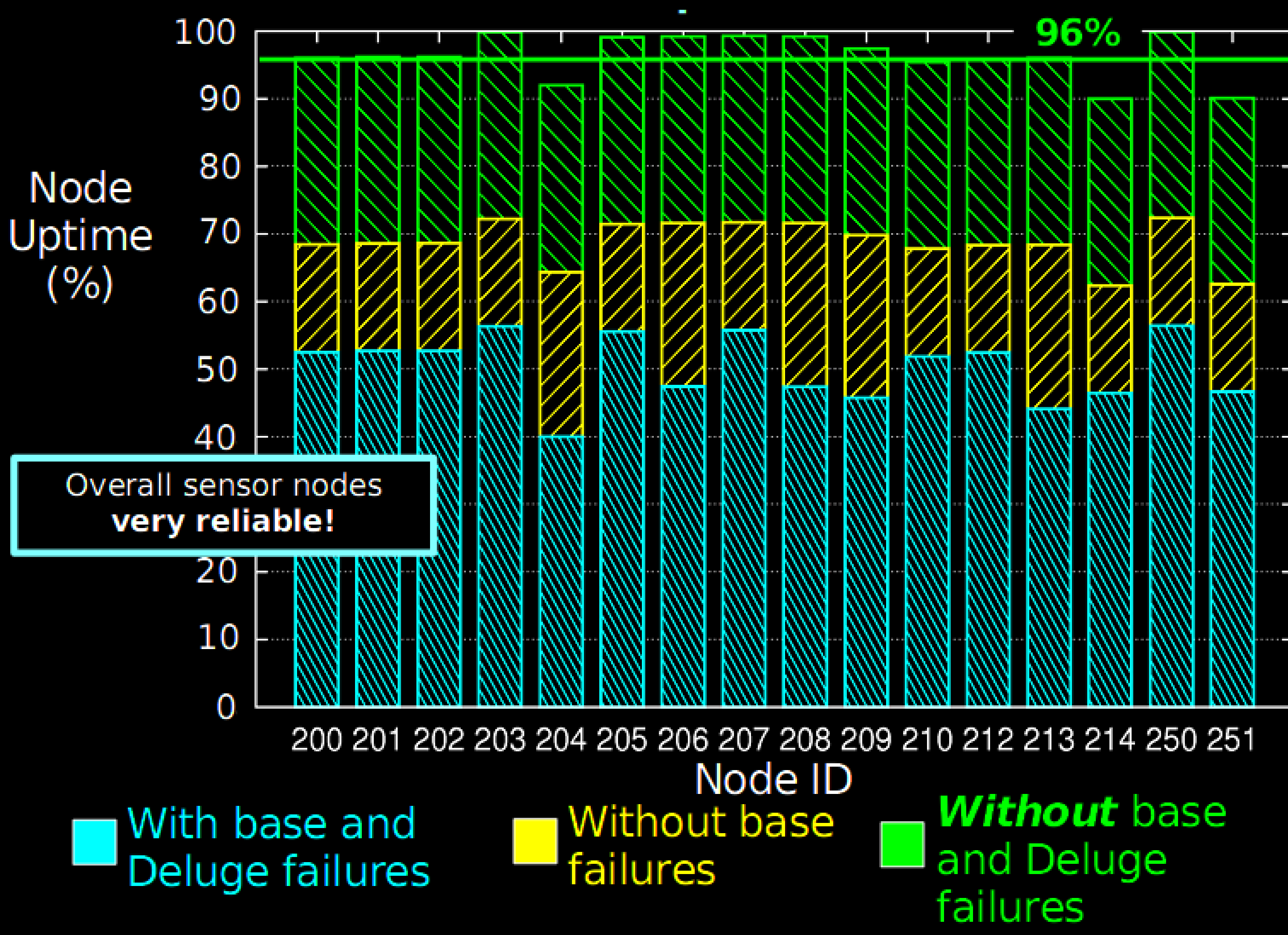


100
90
80
70
60
50
40
30
20
10
0
Node
Uptime
(%)
51%
200 201 202 203 204 205 206 207 208 209 210 212 213 214 250 251
Node ID
With base and
Deluge failures

Node Uptime


100
90
80
70
60
50
40
30
20
10
0
Node
Uptime
(%)
69%
200 201 202 203 204 205 206 207 208 209 210 212 213 214 250 251
Node ID
With base and
Deluge failures
Without base
failures

Node Uptime


100
90
80
70
60
50
40
20
10
0
96%
Node Uptime (%)
Overall sensor nodes
very reliable!
200 201 202 203 204 205 206 207 208 209 210 212 213 214 250 251
Node ID
With base and Deluge failures
Without base failures
Without base and Deluge failures

What we are going to cover…

A small overview

System Architecture

Network Robustness

Event Detector Accuracy

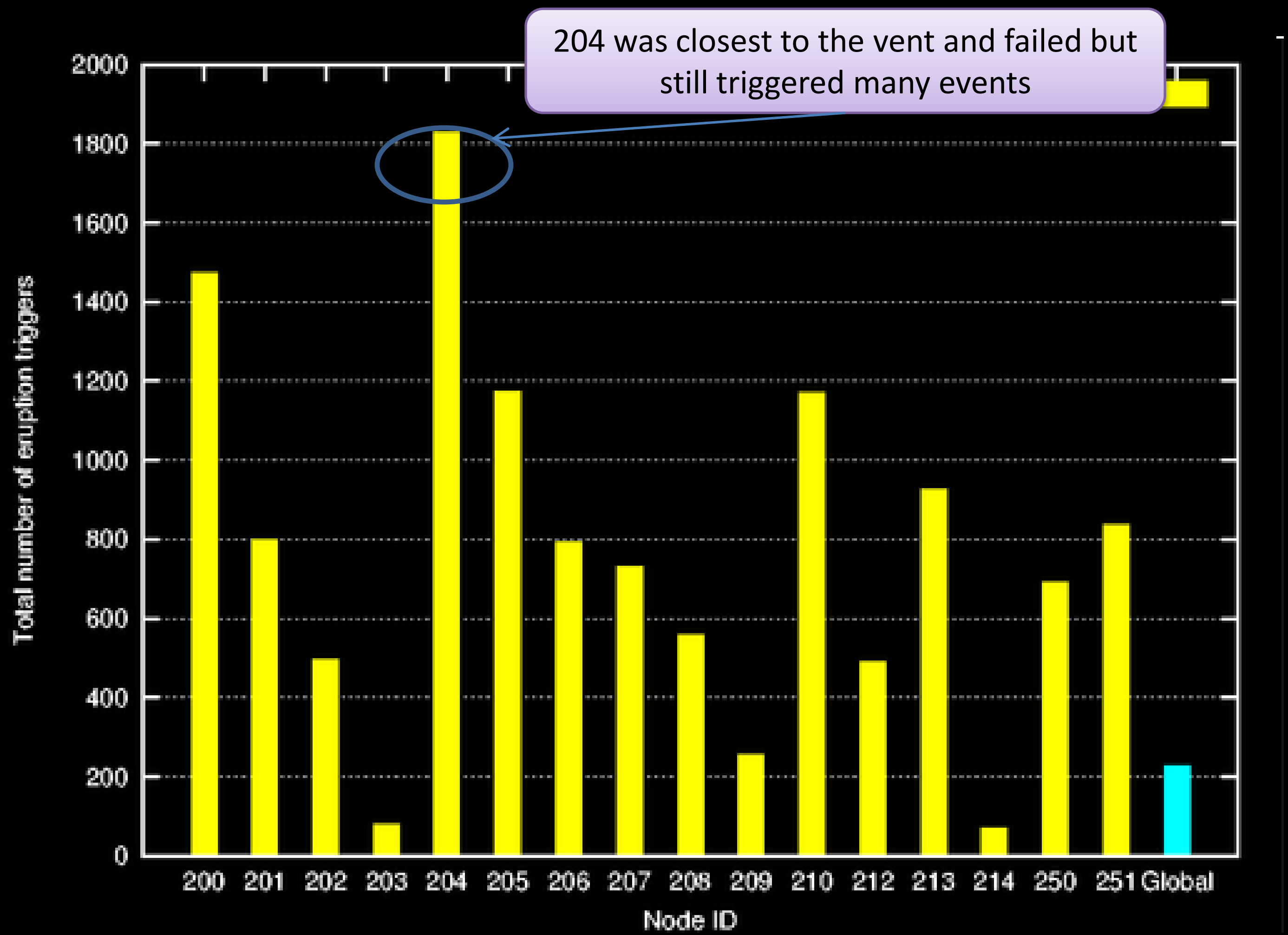
Data Collection Performance

Time Rectification & Accuracy

Data Fidelity

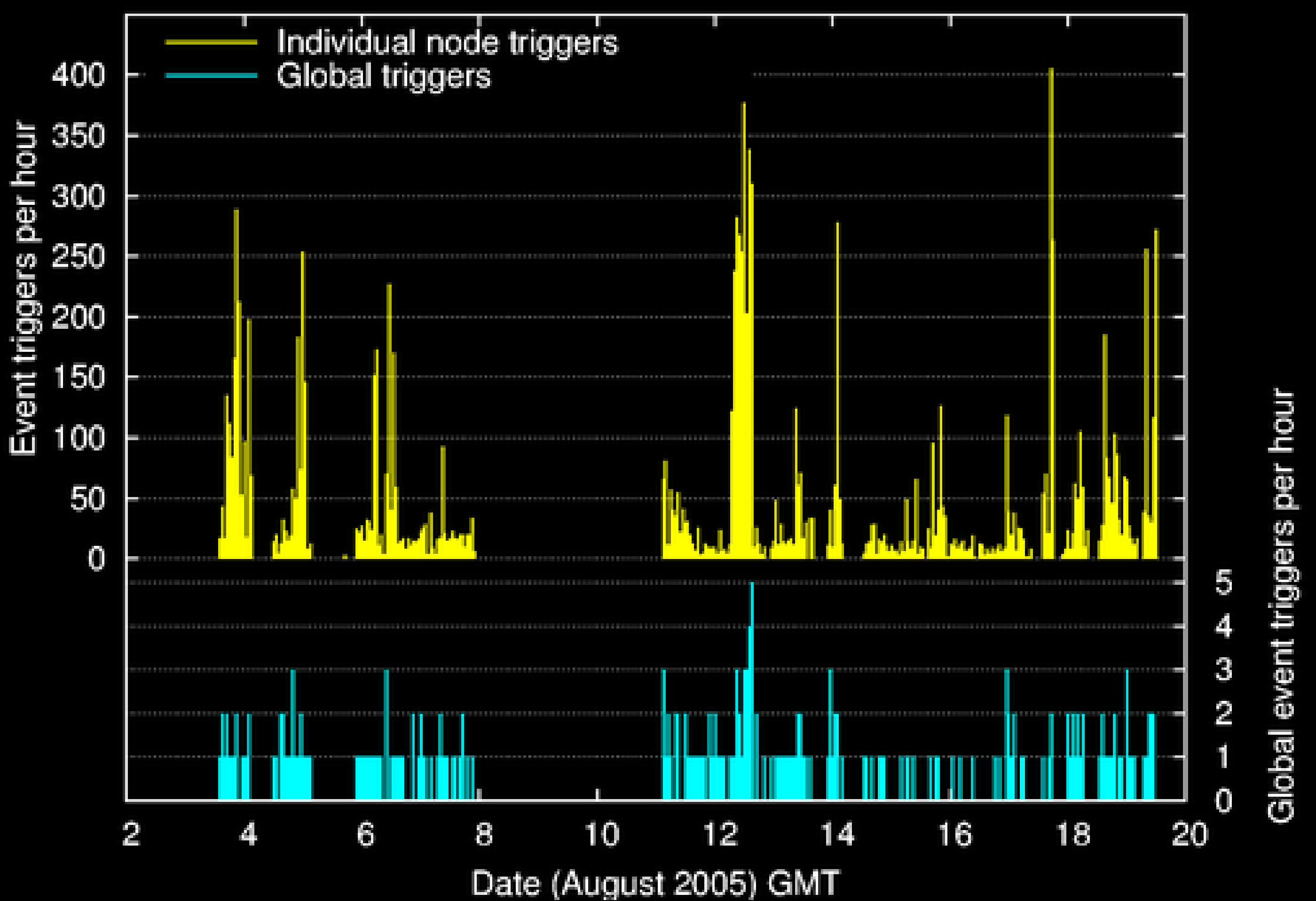
Summary

Event triggers per node



Wide variation - From 70 triggers for node 213 to 1830 triggers for node 204.
Attributed to location of the node, orientation of the seismometer, quality of seismometer to ground coupling.

Event triggers over time



Unpredictable! There is so much of variation in the volcano's activity – But look at the global activity…
That was because of stopping signal recording was stopped while downloading event related data.

Event detector accuracy

Given the **high degree of coherency** required by the global event detector (30% active nodes), recording false events could be difficult.

Comparing the event detection algorithm to that of broadband seismic stations was difficult due to:

- Intermittent power outages at the base stations
- Sensitivity of the seismometer

Initially, **detection accuracy was around 1%** !

- Failure of individual nodes
- Failure of base station or radio modem
- Low sensitivity of the seismometer
- Failure of the event detection algorithm itself

Reason: Sensor network could have detected approximately 24% more events if the protocol was able to sample and download simultaneously.

Anything that can go wrong, will—at the worst possible moment

What we are going to cover…

A small overview

System Architecture

Network Robustness

Event Detector Accuracy

Data Collection Performance

Time Rectification & Accuracy

Data Fidelity

Summary

Data collection performance

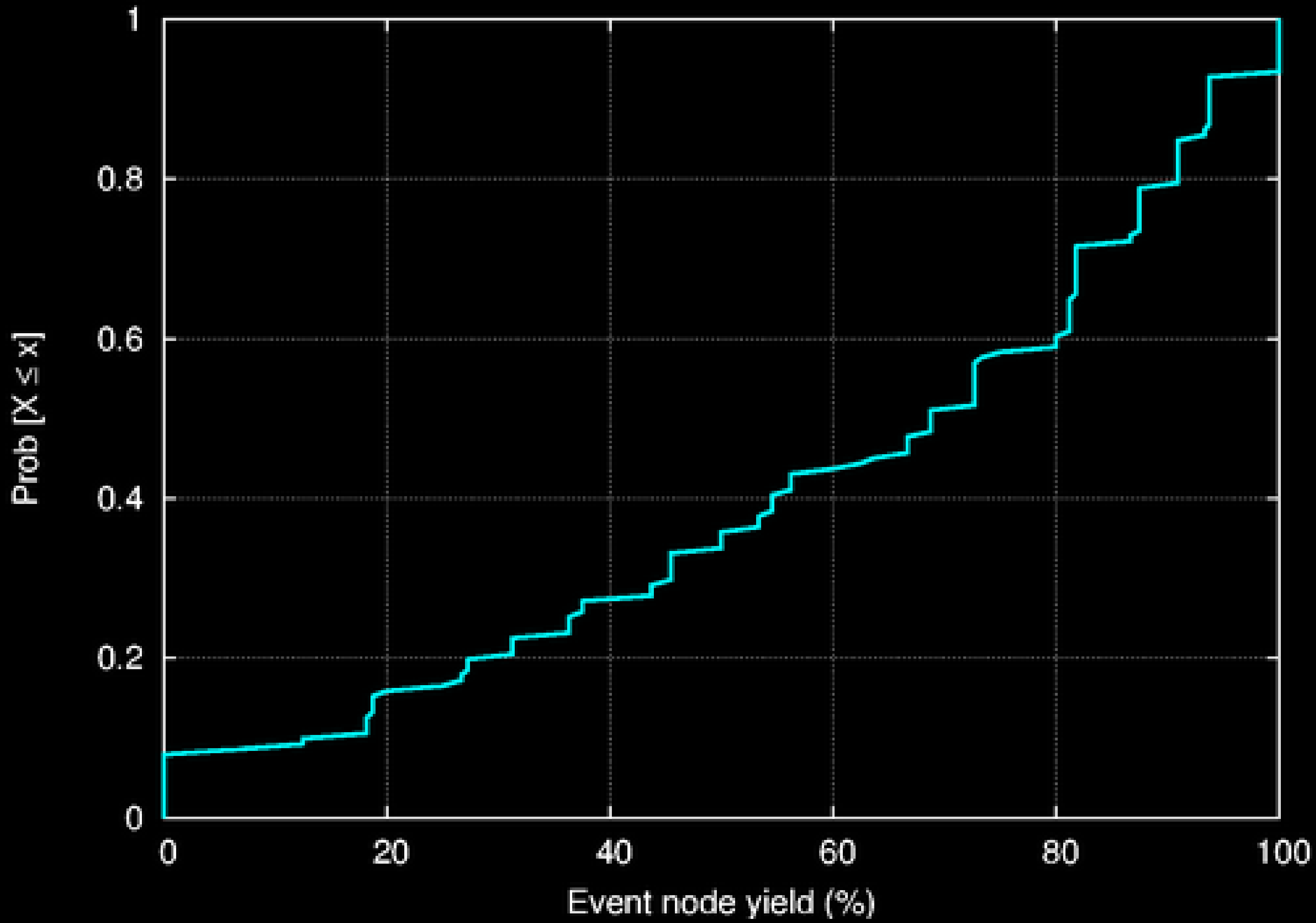
Evaluating Fetch data collection protocol in terms of

- its ability to collect requested data
- Its latency – time to download events from the network

Two aspects:

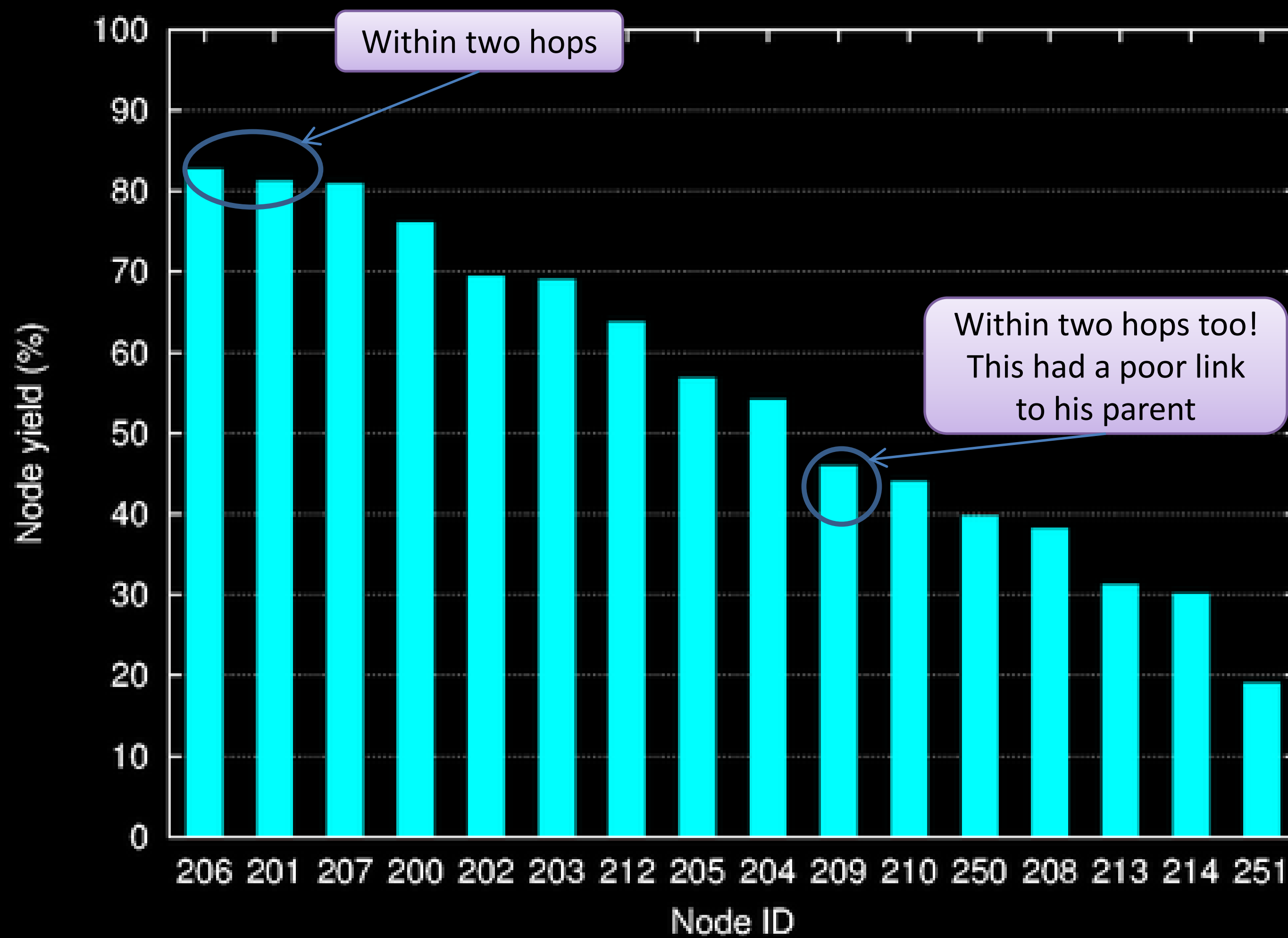
- Event Yield – Fraction of nodes for which the entire 60 sec data was downloaded
- Node Yield – Probability that an event was successfully downloaded

Event yield



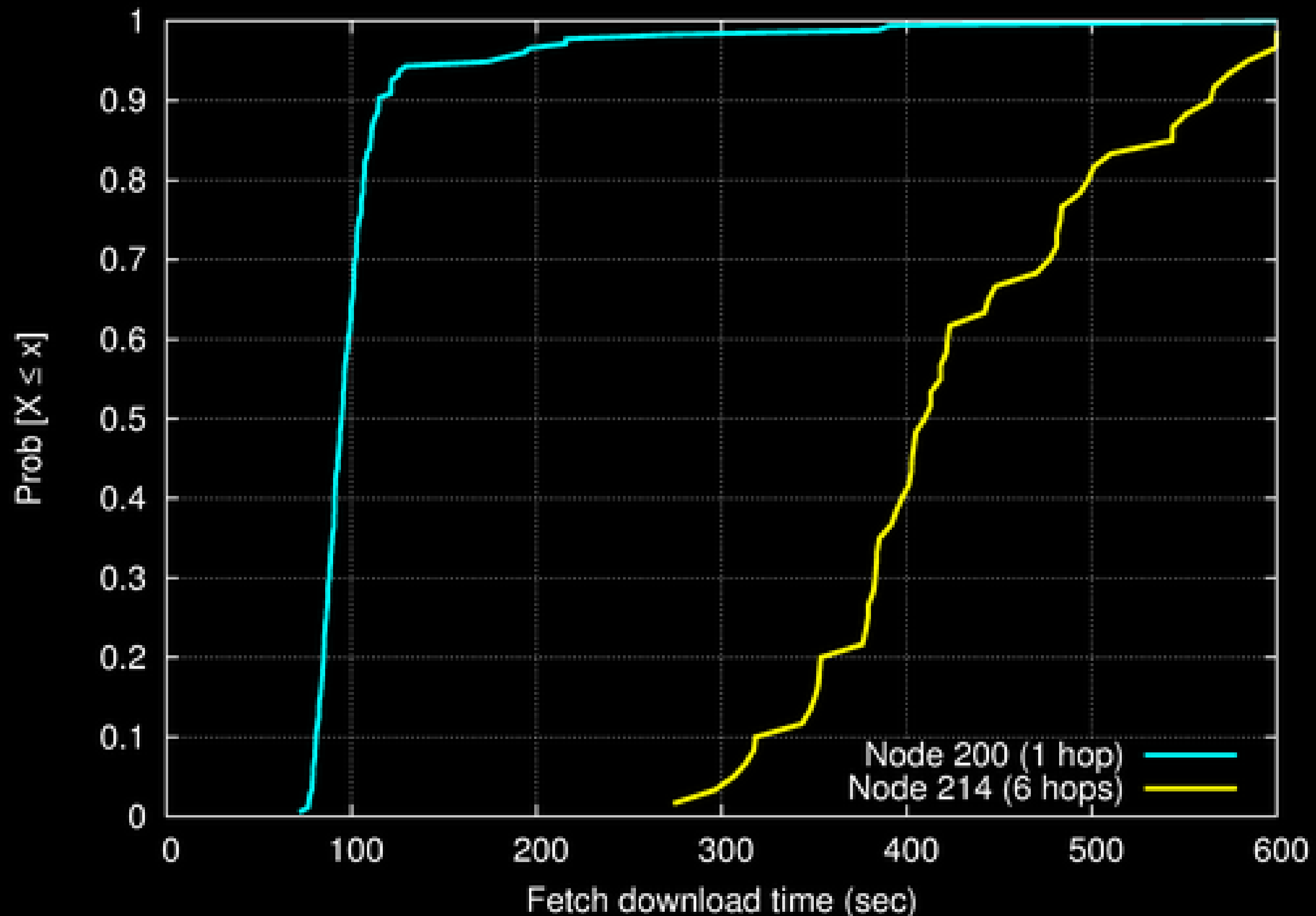
CDF of the event yield – Median event yield was 68.5% and the 90th percentile was 94%
Affected by: Transfer timeout (Request 20 times from a node and abort), Simultaneous transfer

Node yield



Affected by: depth and radio link quality of node's routing path to the base station, Packet loss rate

Fetch Latency



Latency varies with the number of hops.

No matter what the result, there is always someone eager to misinterpret it.

What we are going to cover…

A small overview

System Architecture

Network Robustness

Event Detector Accuracy

Data Collection Performance

Time Rectification & Accuracy

Data Fidelity

Summary

Time rectification and accuracy

When dealing with sensitive signals, timing accuracy is required

Uses Flooding Time Synchronization Protocol, an existing protocol for WSN

Three relevant time bases:

- Local time at each node
- Global time established by the FTSP protocol
- GPS time recorded by the FTSP root

FTSP: Flooding Time Synchronization Protocol

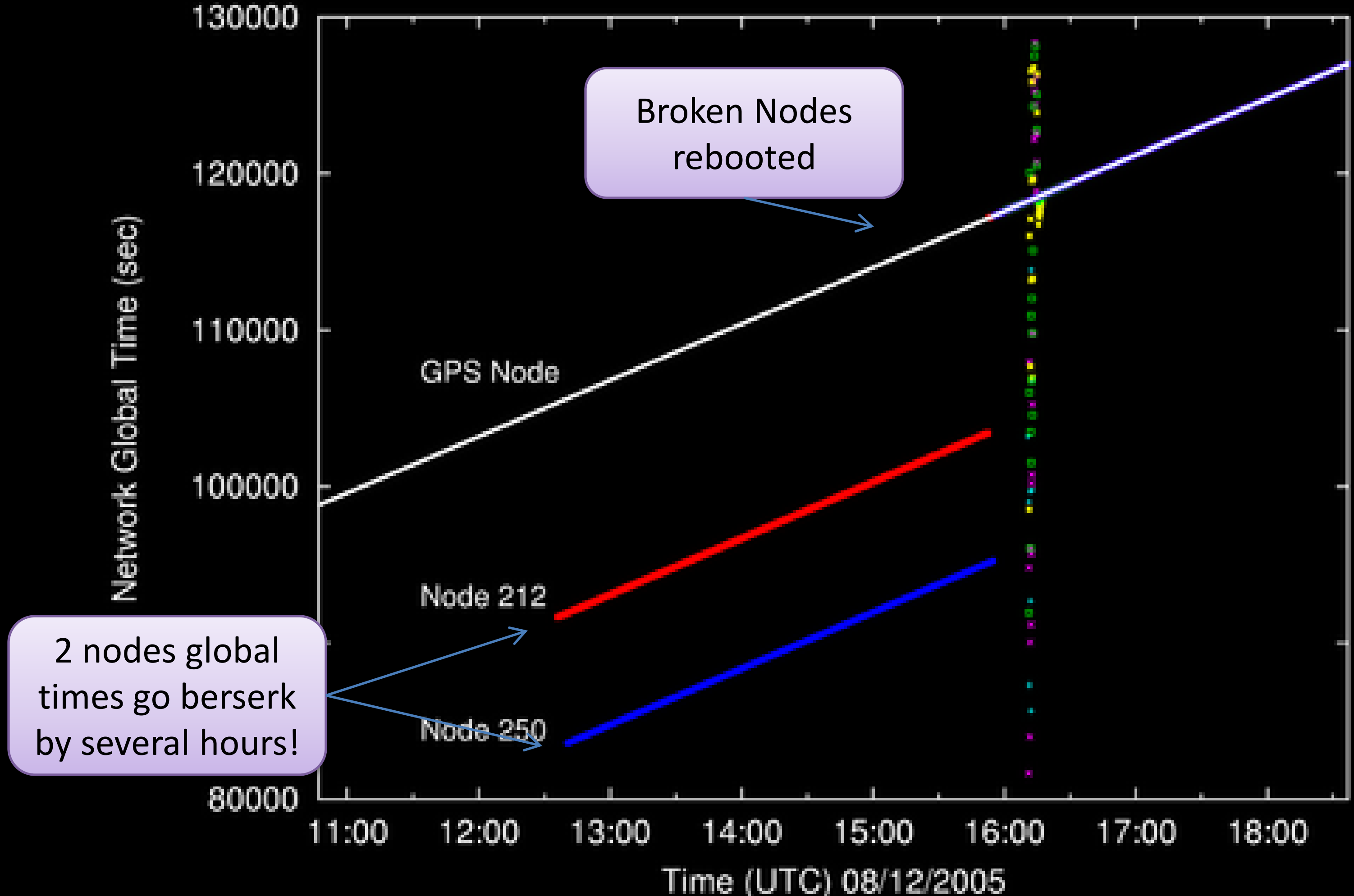
In FTSP, nodes periodically exchange information on global time

- Each node sends heartbeat messages with MAC-delay corrected global timestamp
- Nodes use this information to calculate local clock skew w.r.t. global clock

Very good pre-deployment test results:

- 90th percentile error of 2.1 ms on a 5-hop linear testbed

FTSP Failures in the field



Affected by: A bug in TinyOS clock driver – returns bogus local timestamps & FTSP does not check the validity of synchronization messages – One node can corrupt others!

FTSP Stability Issues

Problem: Nodes can produce corrupted FTSP messages

Bug in MSP430 clock driver caused nodes to sometimes read wrong time.

FTSP trusts this information and includes it in local skew/offset calculations

Also, bad information can propagate throughout the network.

Lesson: FTSP should perform internal consistency checks

e.g., Avoid radically updating local phase/skew if well-synchronized already

Post hoc time rectification technique

Must correct global timestamps in recorded data set

1) Filter out obviously wrong global timestamps

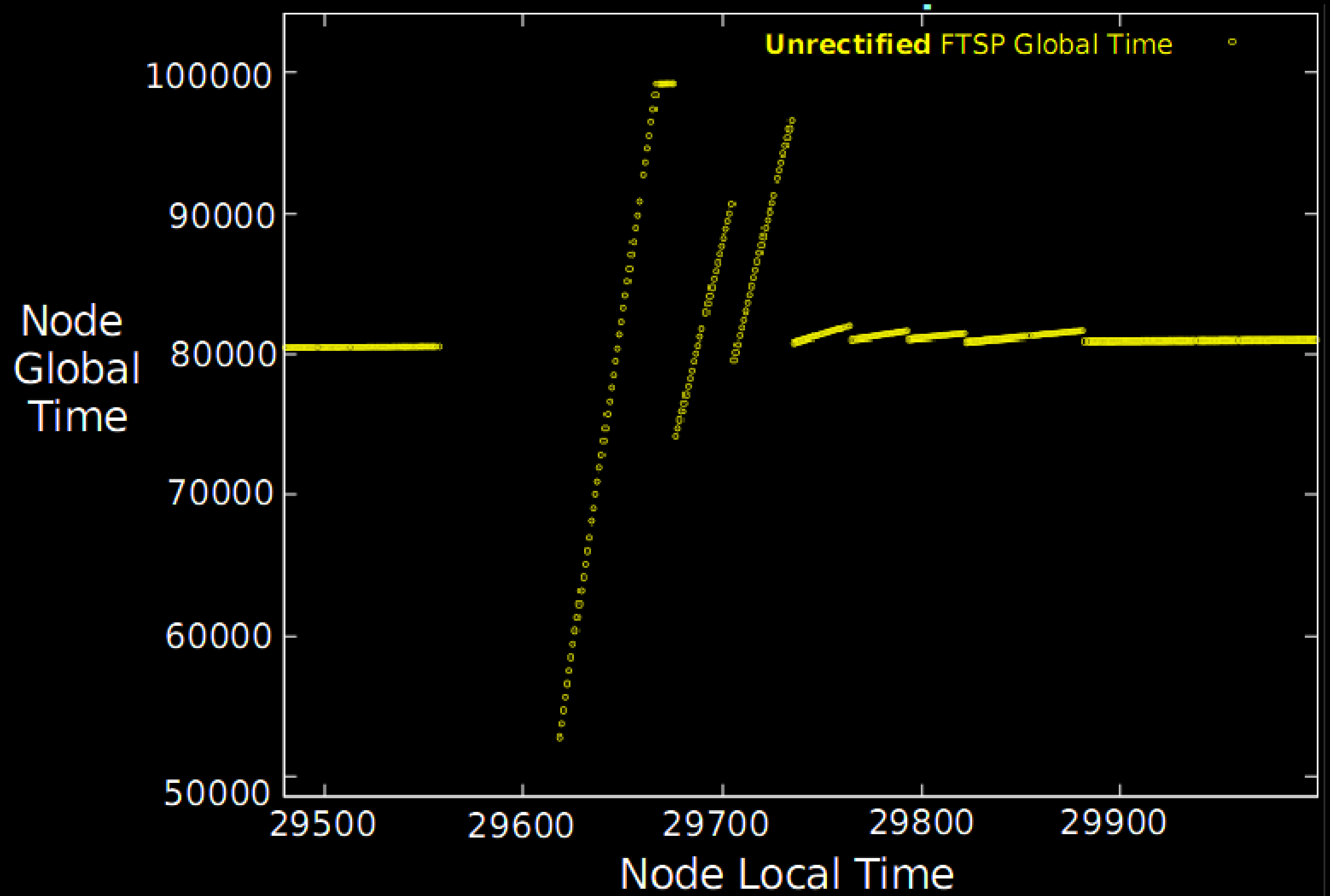
e.g., Those that differ significantly from global time recorded by FTSP root
Threshold used: 1 sec.

2) Perform piecewise linear regression on remaining data set

Produce a mapping from node's local time to “correct” global time
Linear regression extends for no more than 30 minutes
Check that clock skew produced by model is sane (within error tolerance of crystal frequency)

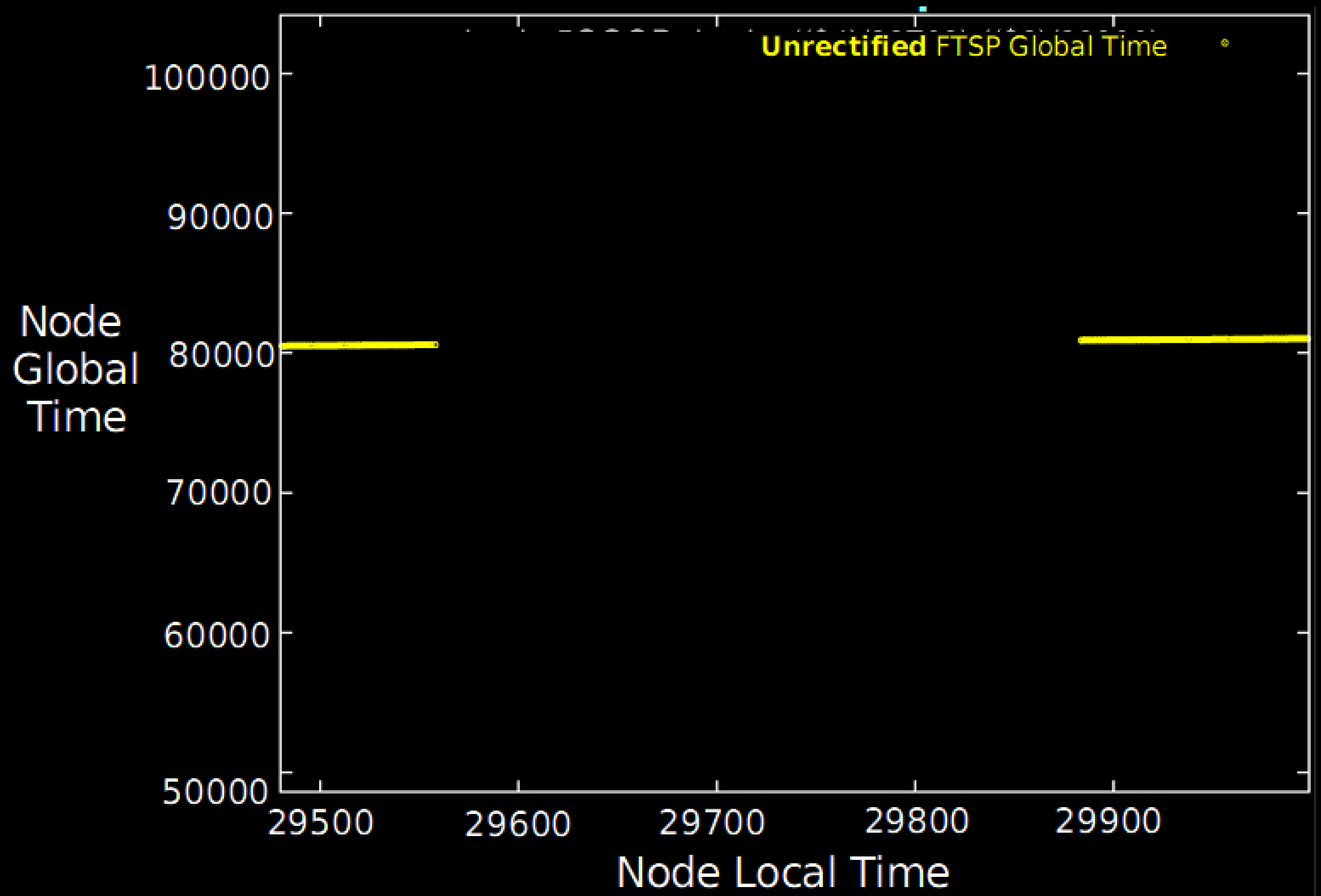
3) Apply this mapping to the recorded data for all events

Time Rectification Table



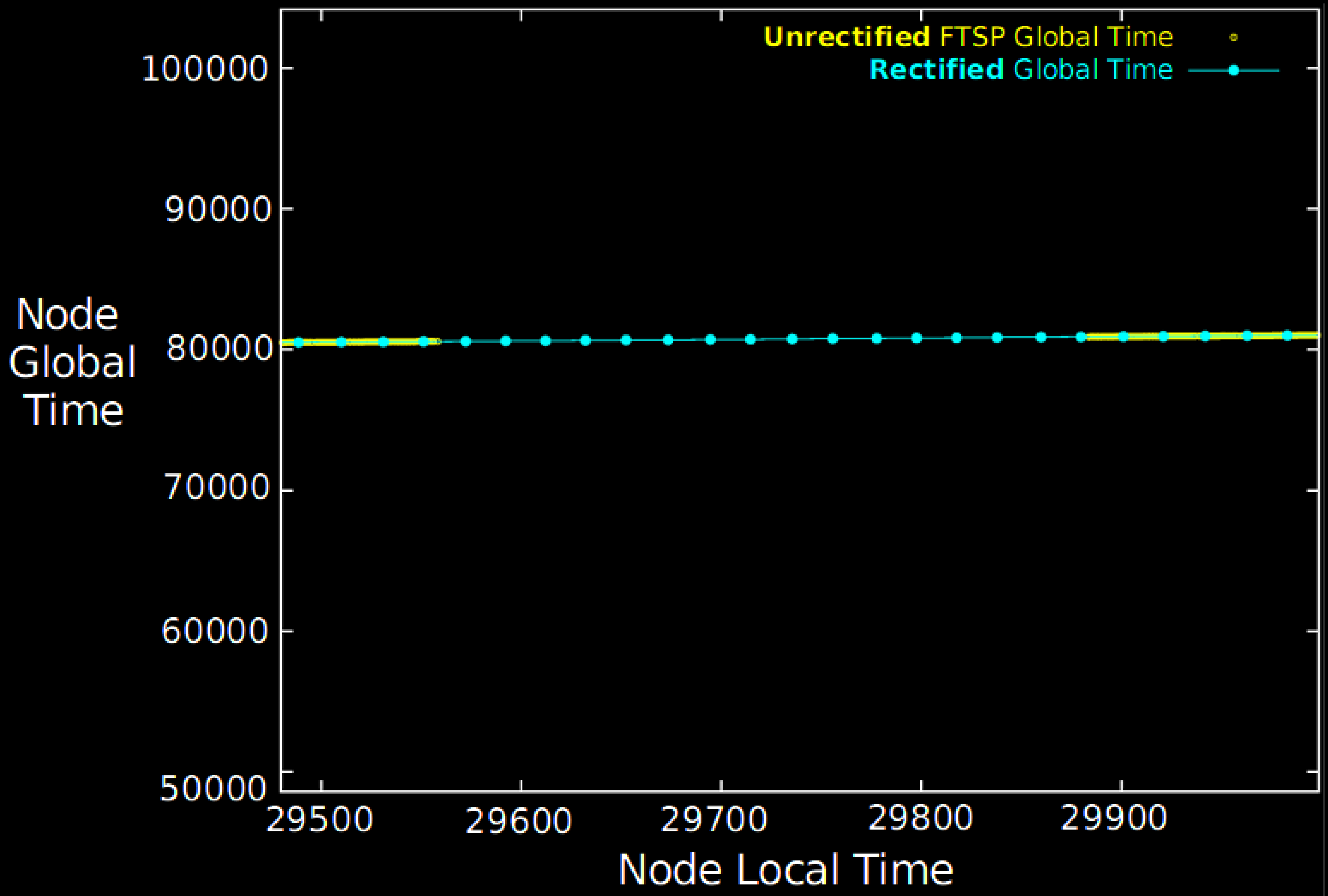
The time rectification process removes the errant timestamps creating an accurate mapping between LT and GT created using linear regression on the remaining timestamps.

Time Rectification Table



The time rectification process removes the errant timestamps creating an accurate mapping between LT and GT created using linear regression on the remaining timestamps.

Time Rectification Table



The time rectification process removes the errant timestamps creating an accurate mapping between LT and GT created using linear regression on the remaining timestamps.

How to know if the timing was right?

Compare data captured by one of the sensor nodes to a nearby wired seismic station

Data timestamped by **FTSP**

Node 213
(3 hops from FTSP root)

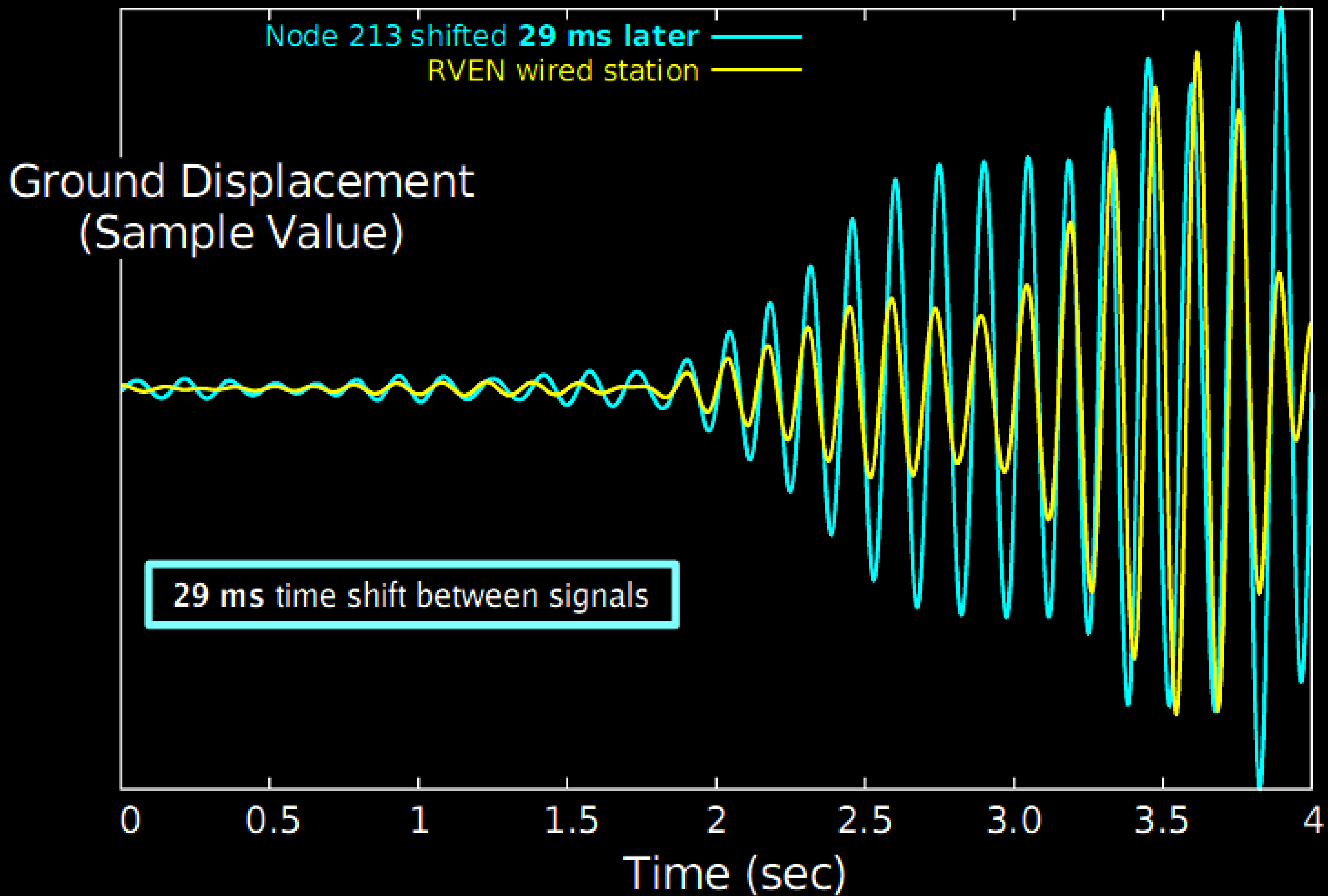
56 meters

Data timestamped by **Internal GPS**

RVEN
Reventador **VEN**t
standalone seismic station

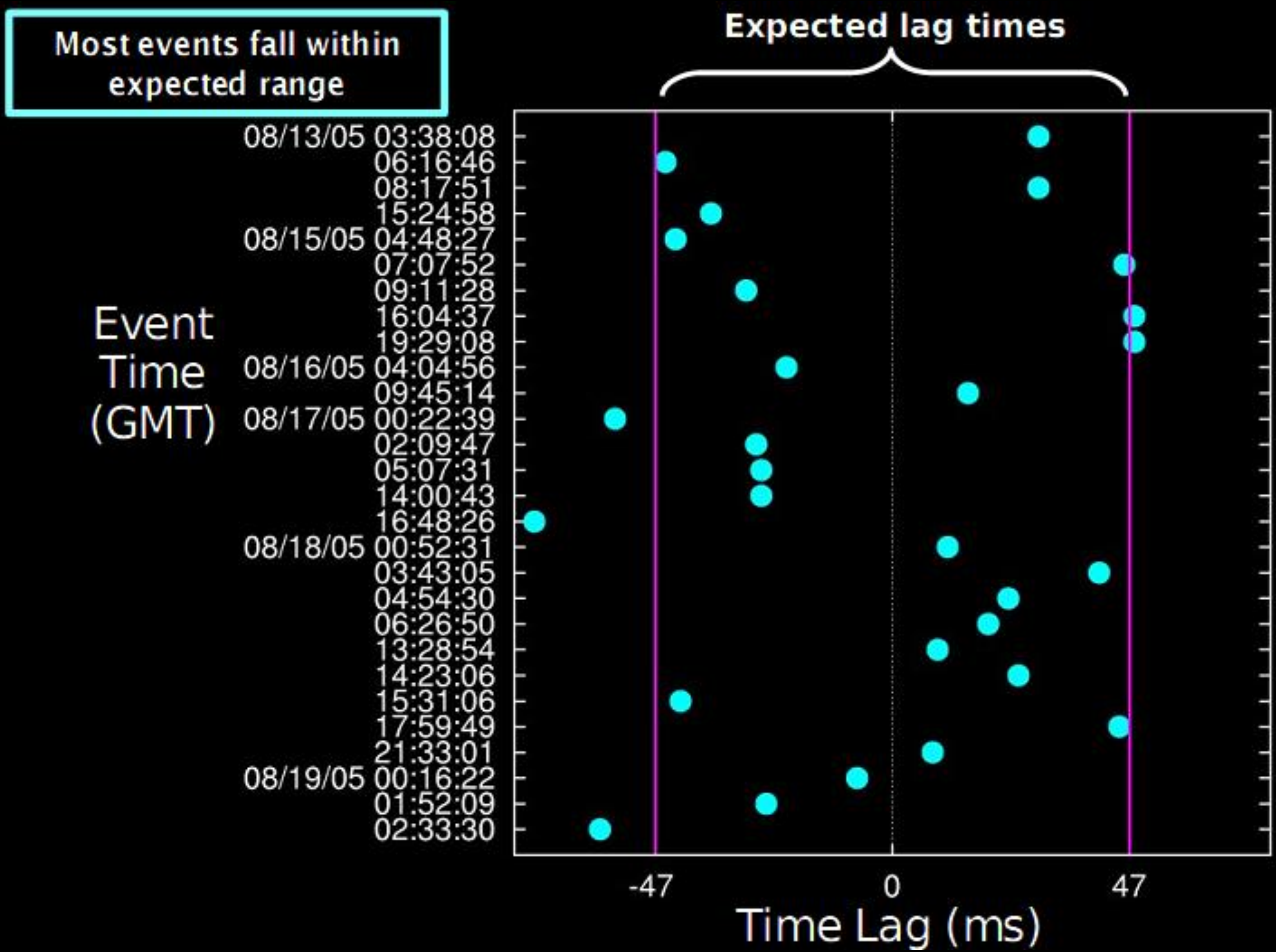
Largest expected offset: 47ms
(based on distance between stations)

Comparison with Reftek wired station



Only a 29ms time shift between signals was observed. After time rectification, 99.9% of the errors for the one-hope Node and 93.1% of the errors for the six hop node fall within 10 ms error envelope.

Time lags across multiple events



In any collection of data, the figure most obviously correct, beyond all need of checking, is the mistake.

What we are going to cover…

A small overview

System Architecture

Network Robustness

Event Detector Accuracy

Data Collection Performance

Time Rectification & Accuracy

Data Fidelity

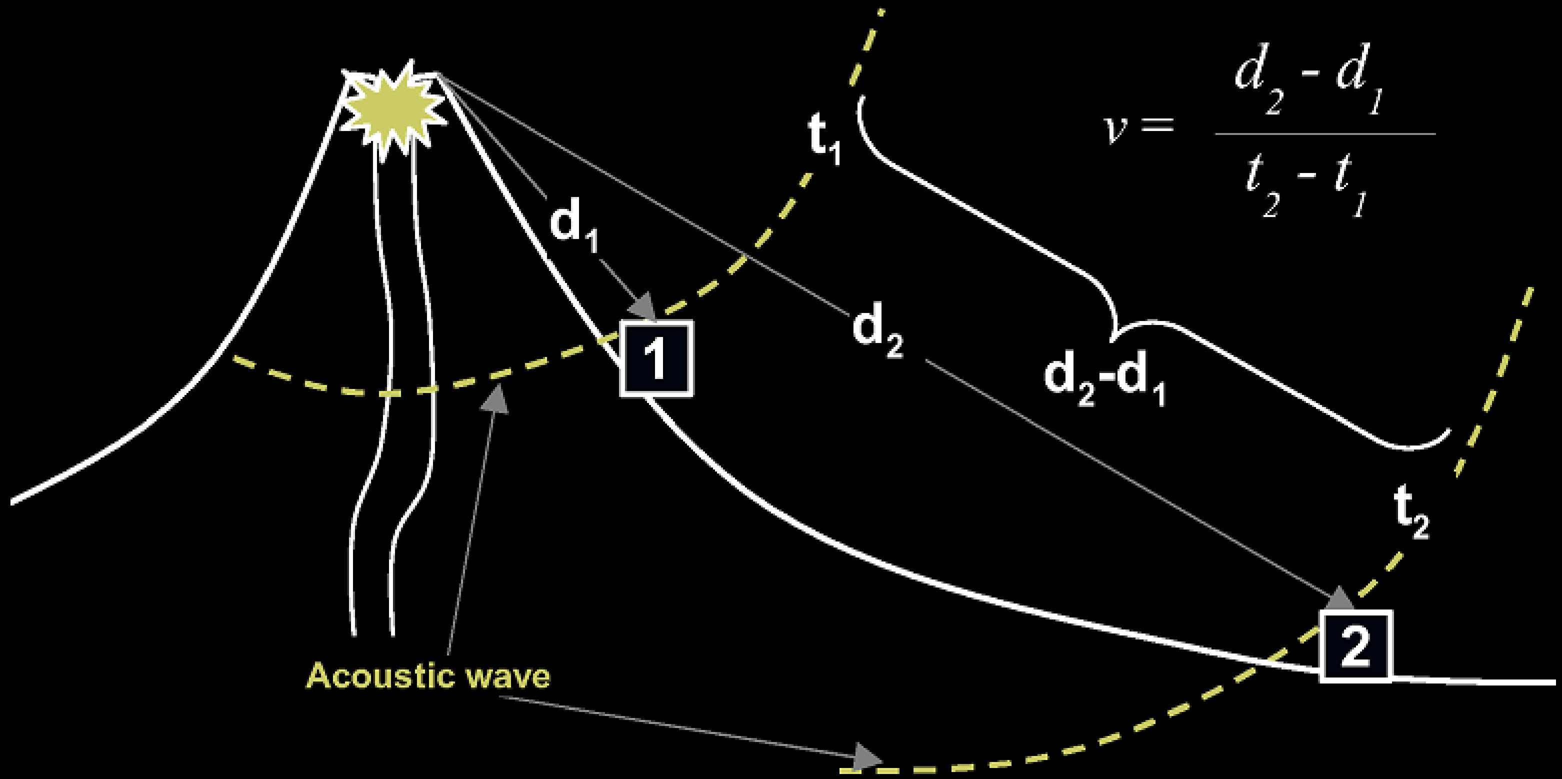
Summary

Data Fidelity

Ability to provide scientifically meaningful data on the volcano's activity.

Initial analysis on seismic and acoustic signals from a seismological perspective

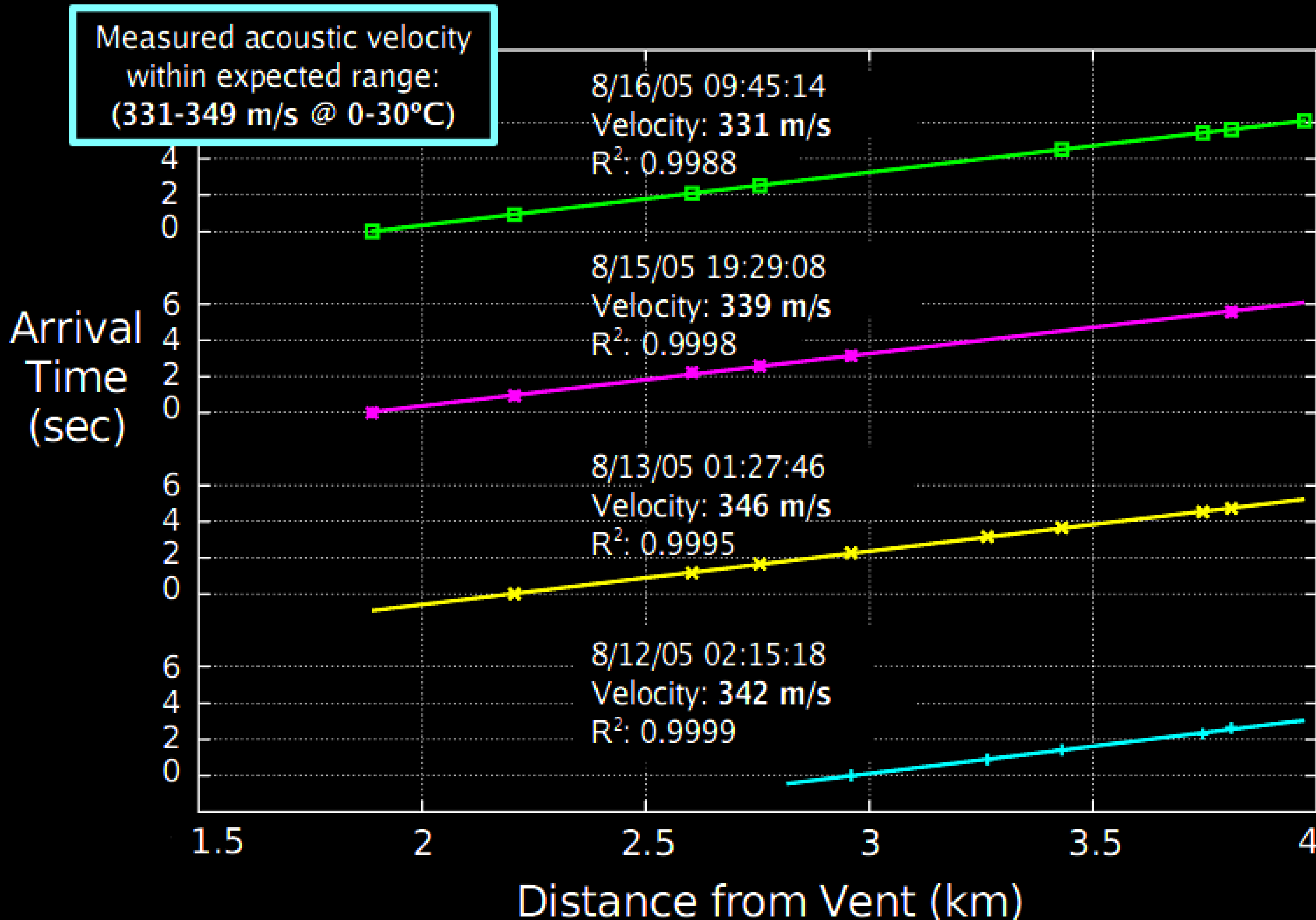
Data Validation: Acoustic Wave Traversal



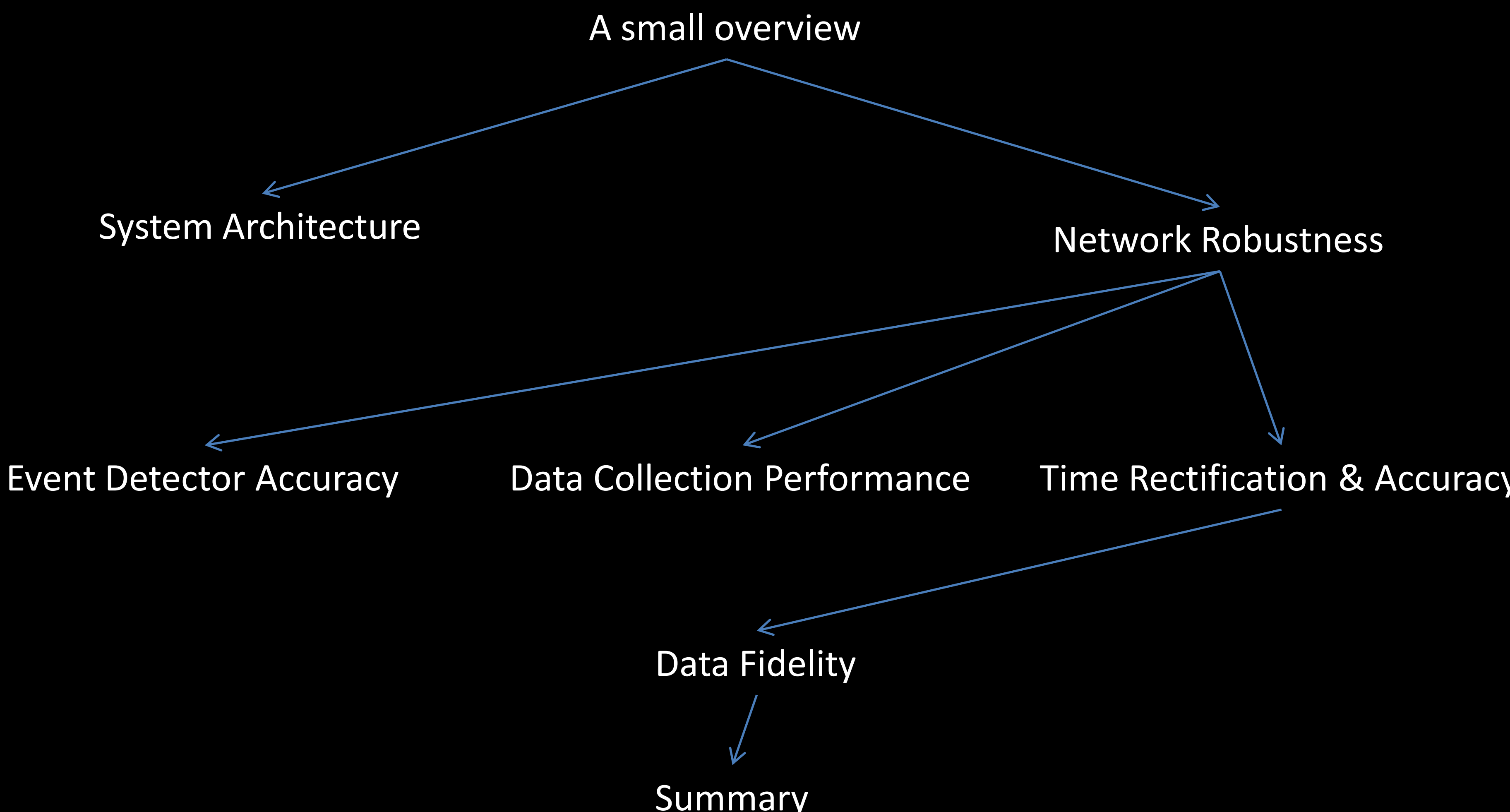
$$v = \frac{d_2 - d_1}{t_2 - t_1}$$

The velocity of the acoustic wave is calculated based on the distance of each station from the vent, di, and the arrival time of the wave at each station ti

Data Validation: Acoustic Wave Traversal



What we are going to cover…



Summary

Validation of sensor network as a scientific instrument

Robustness

- How available was the system during the deployment?
- Surprisingly, the sensor nodes themselves were far more reliable than the infrastructure

Timing accuracy and data fidelity

- How accurate were the timestamps assigned to each sample?
- After a great deal of work accuracy was achieved within the scientific tolerances in most cases
- Was the data consistent with ground truth and internally consistent?
- Although ground truth was difficult to ascertain, well…

And finally…

If mathematically you end up with the wrong answer, try multiplying by the page number.

NOTE:

I didn’t make up these snippets. In fact, these (except the last one) are known by the name Finagle's laws

These have nothing to do with the actual research mentioned in this presentation though!

Whew… So much for a volcano ☺
That's it folks!
Thanks for your patience