

Extending the Internet Architecture to Sensor Networks: Some Open Questions

Matt Welsh

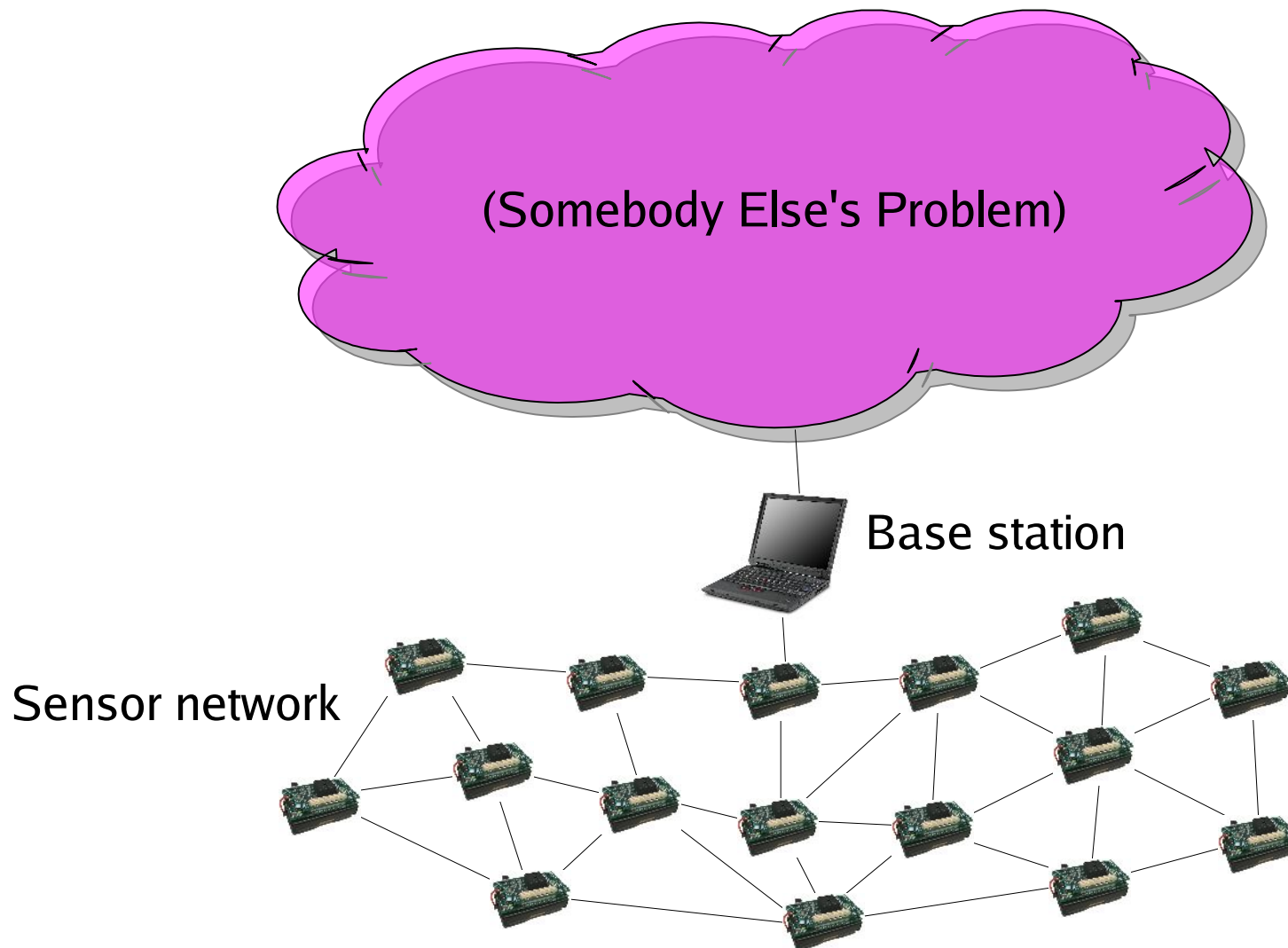
Harvard University

Division of Engineering and Applied Sciences



The Problem, in a nutshell...

Traditional view of sensor networks:



The Crisis of Connectivity

The great strides in sensor networks have led
to a looming “crisis” ---

***How are sensor networks going to be integrated
with rest of the Internet?***

The Crisis of Connectivity

The great strides in sensor networks have led
to a looming “crisis” ---

*How are sensor networks going to be integrated
with rest of the Internet?*

This problem is very sticky, and the answer is not at all obvious.

Sensor nets are not just passive instruments.

- They can (and should be) remotely reprogrammed and retasked.

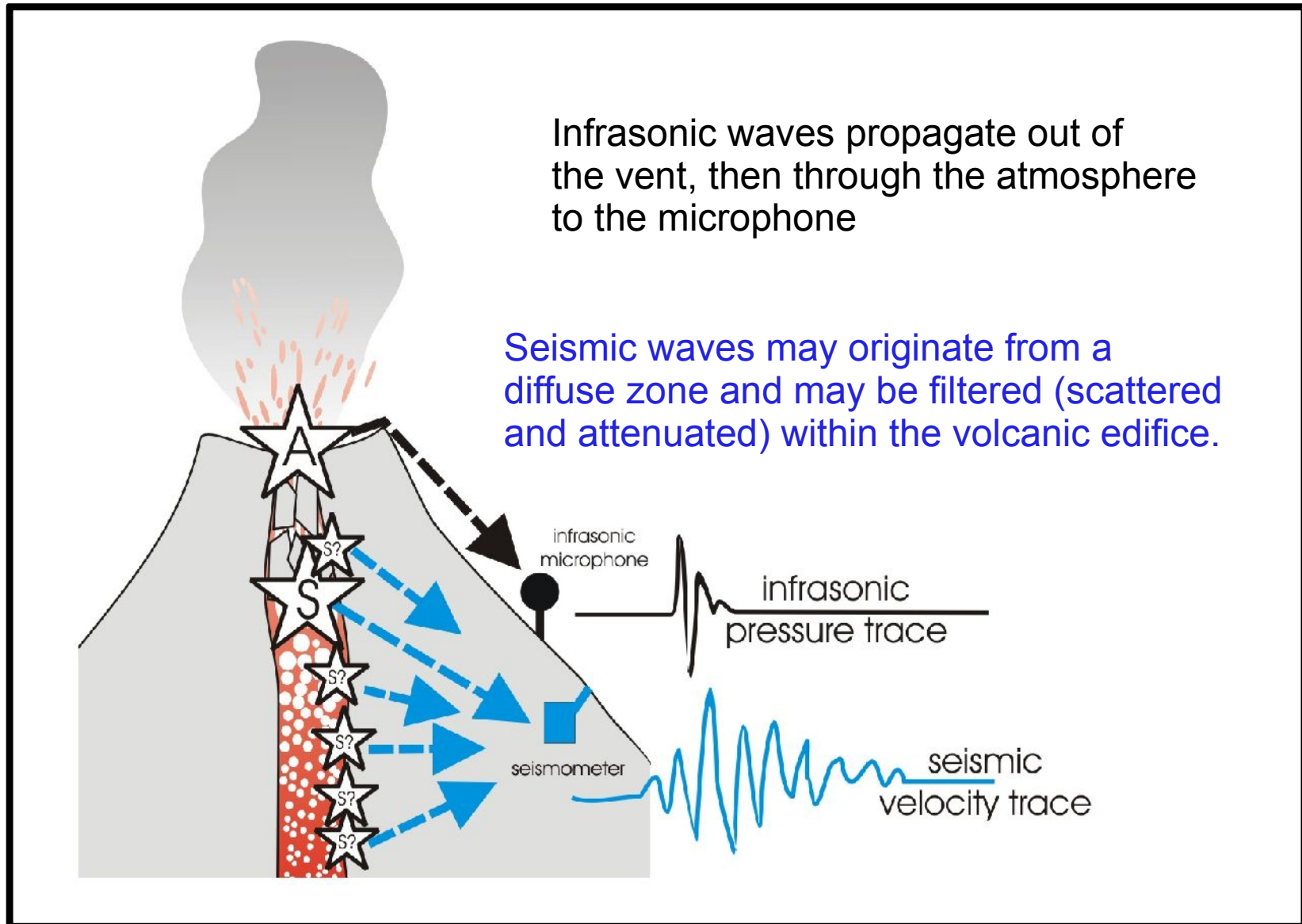
Sensor nets are unlike traditional Internet “services.”

- Cannot be provisioned in the same way: Inherently limited resources and, by extension, capacity.

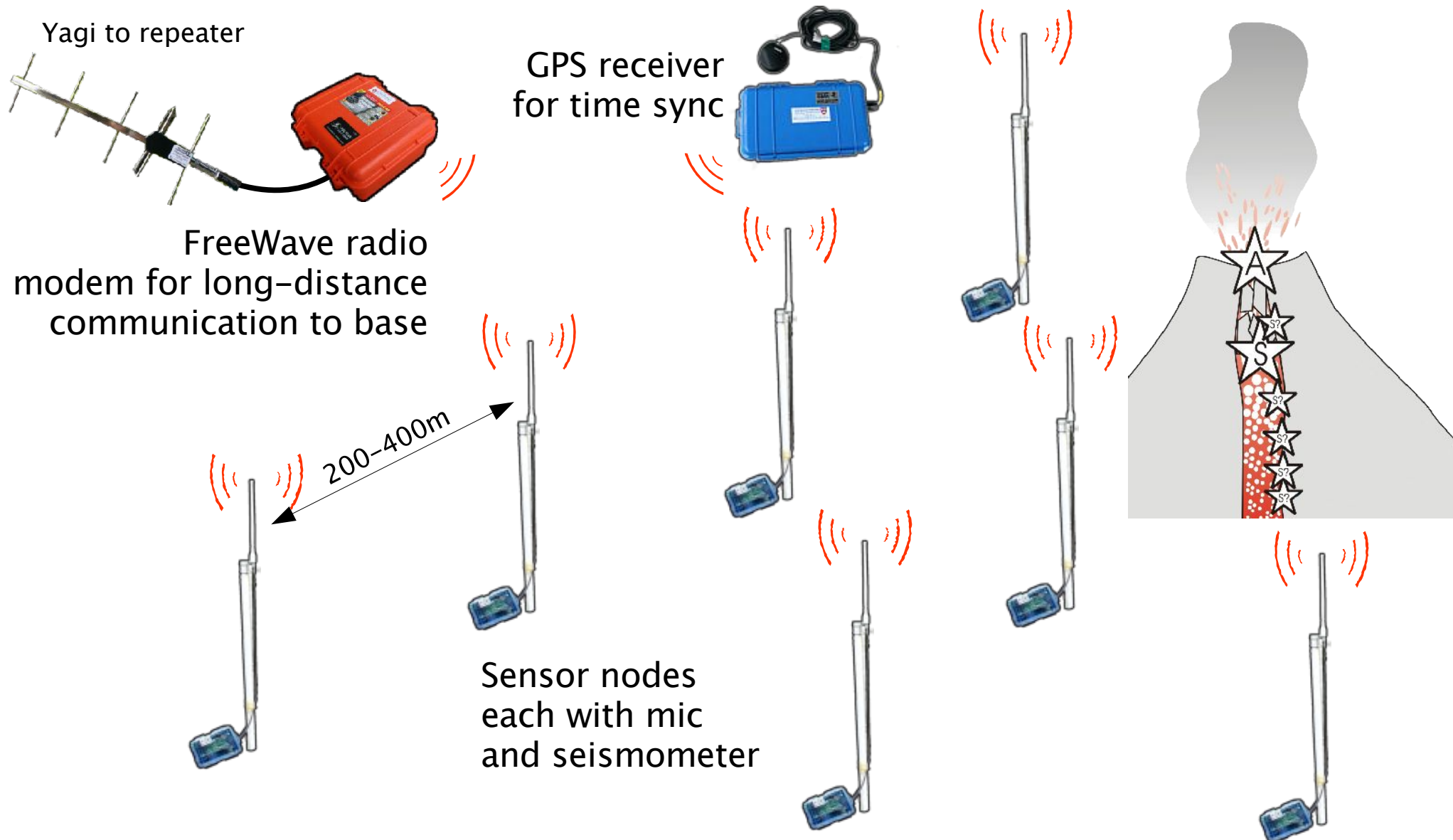
It is unclear how to extend the current Internet Architecture to support large numbers of wireless sensors in the wild.

Example: Monitoring Volcanic Eruptions

joint work with UNH, UNC, Instituto Geofisico Ecuador



The Volcano Monitoring Macroscope

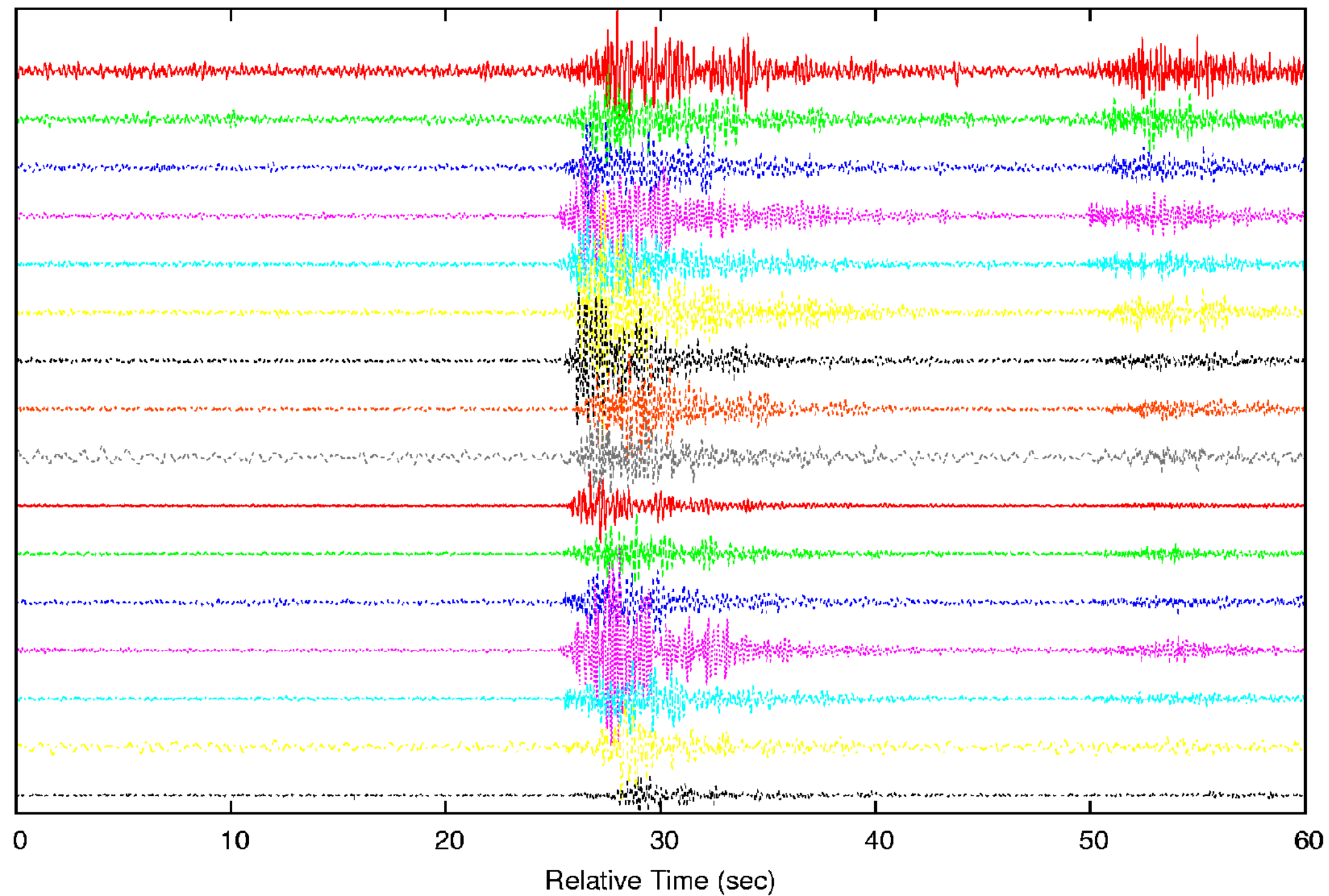


Volcán Reventador, Ecuador

July/August 2005



Seismic Event - 2005-08-16_{04.05.56}



Some Immediate Issues...

Data collected was logged on our laptop.

- Over 40 GB of raw data logged over 3 weeks from hundreds of earthquakes.

Well, we *tried* to make the data available via the Internet...

- Goal was to post an activity report following large events
- NASA/JPL ready to trigger imaging via EO-1 satellite from our reports!

But, the “last 484 miles problem” got in the way

- Satellite connectivity via Iridium was extremely unreliable!

Even if we had succeeded, what is the right communications model?

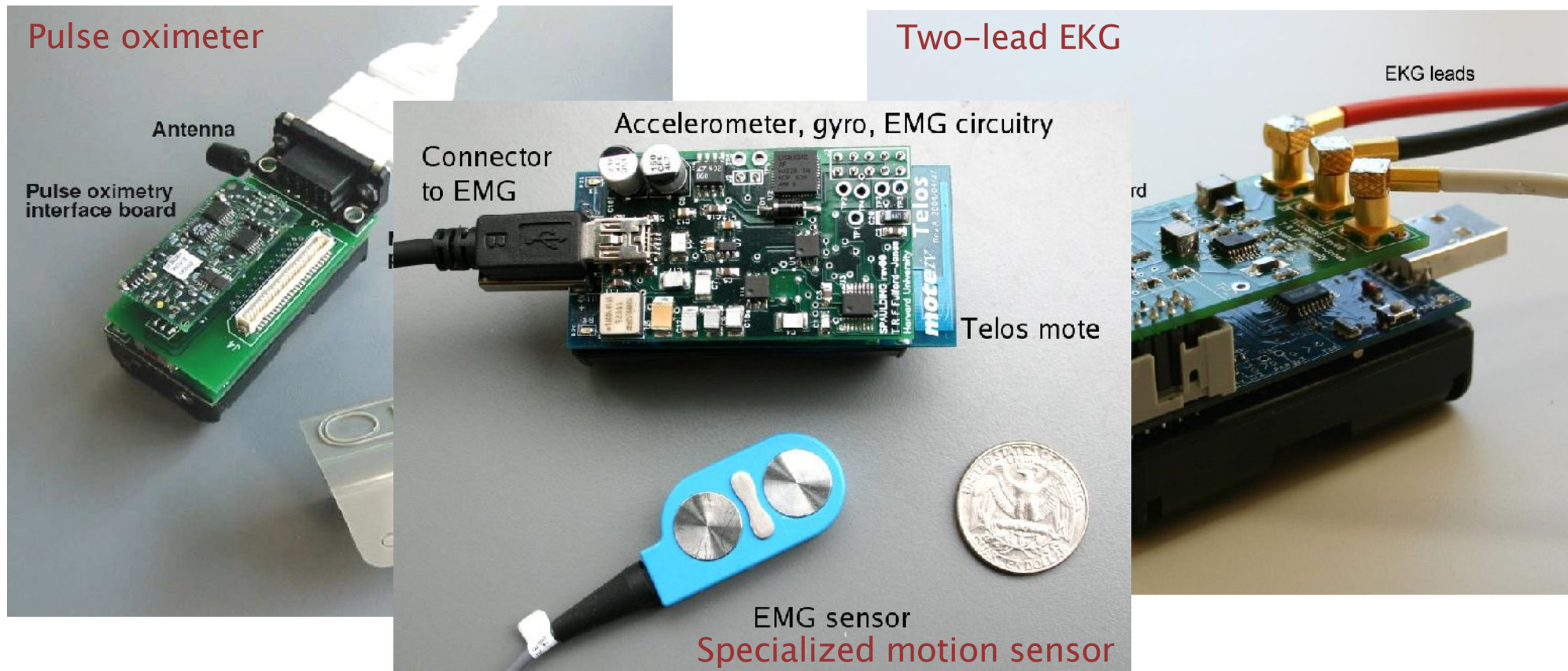
- Sensor net was not continuously connected...
- And there's no way we could have uploaded all of our signals in real time!

Another Frontier: Medical Care

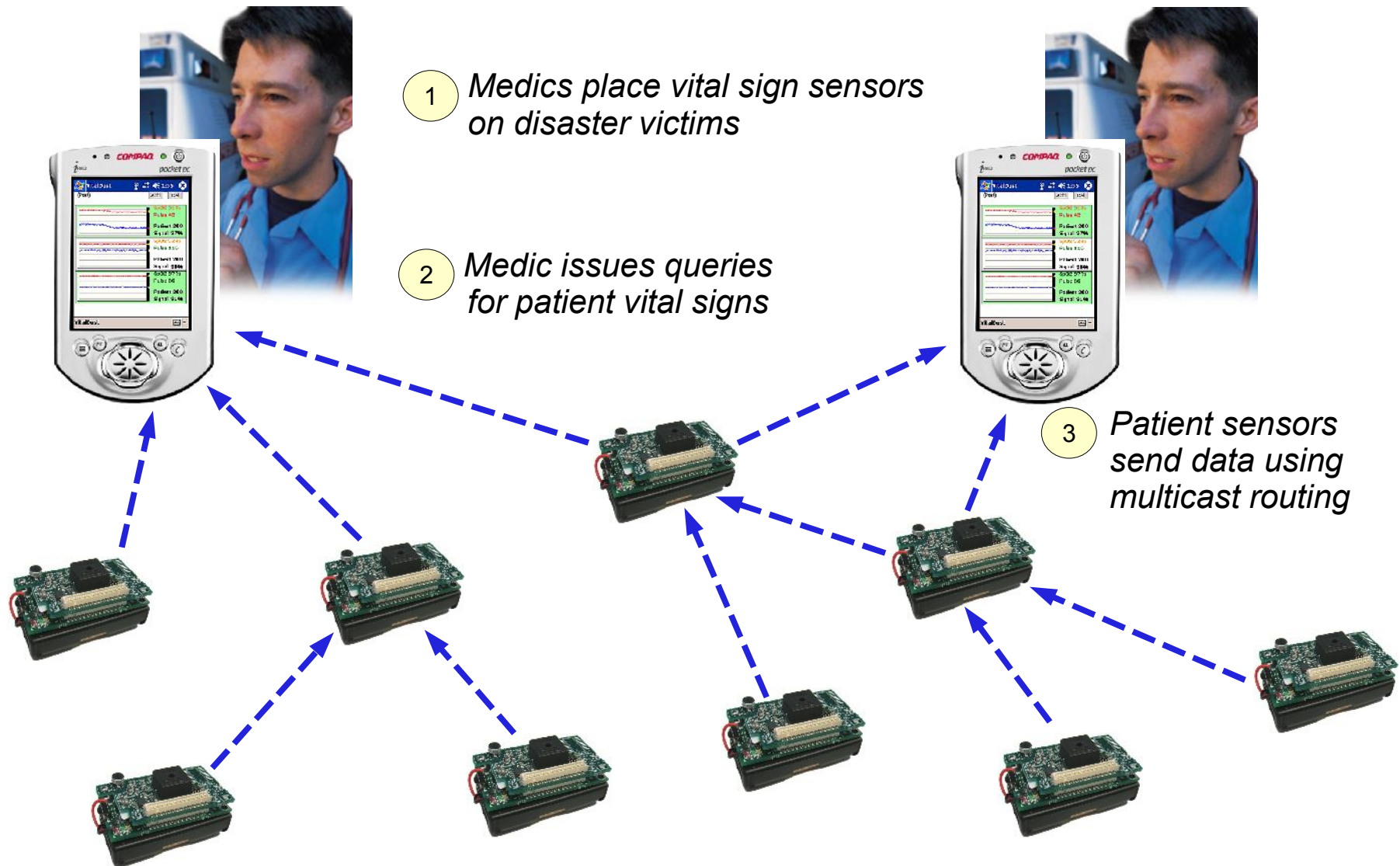
Harvard CodeBlue Project

Sensor networks have enormous potential in medicine:

- Pre-hospital, in-hospital, and ambulatory monitoring
- Tracking many patients in a disaster or mass casualty event
- Monitoring entire populations (e.g., chronic or elderly patients at home)



Example: Disaster Response Scenario



CodeBlue Web Services Interface

EMS / 911 Dispatch



Hospital staff



Hospital Information Systems

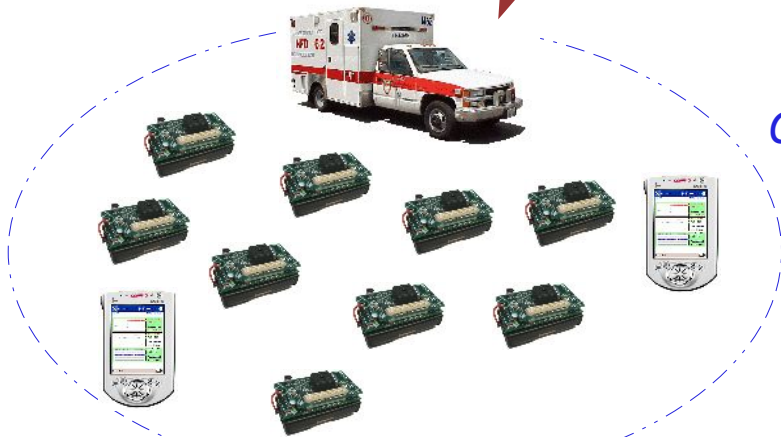


Web Services Proxy Server

Vital signs, location

Triage, transport

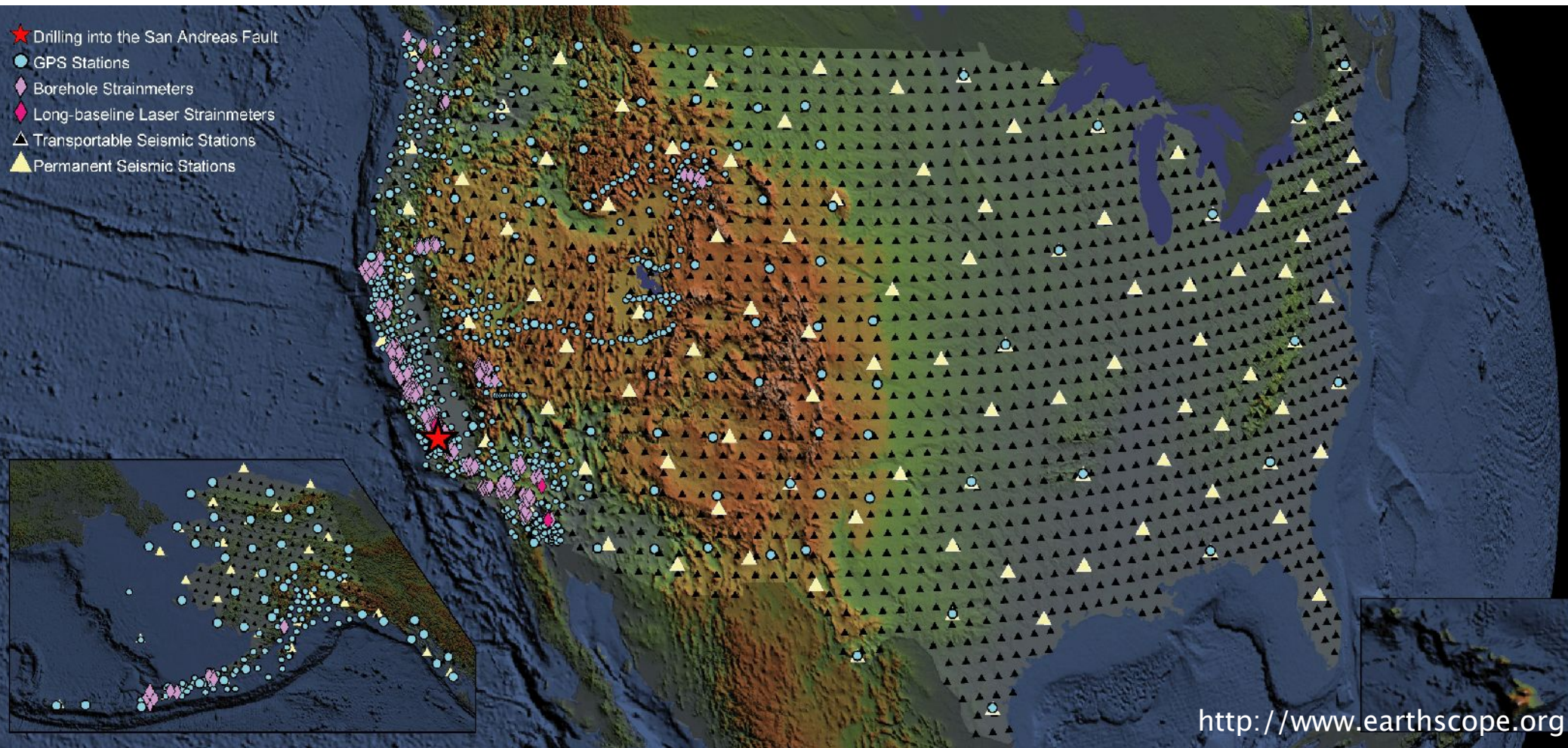
CodeBlue networks at disaster sites



Global Scale Sensor Networks

We often think of sensor networks as relatively small, and local.

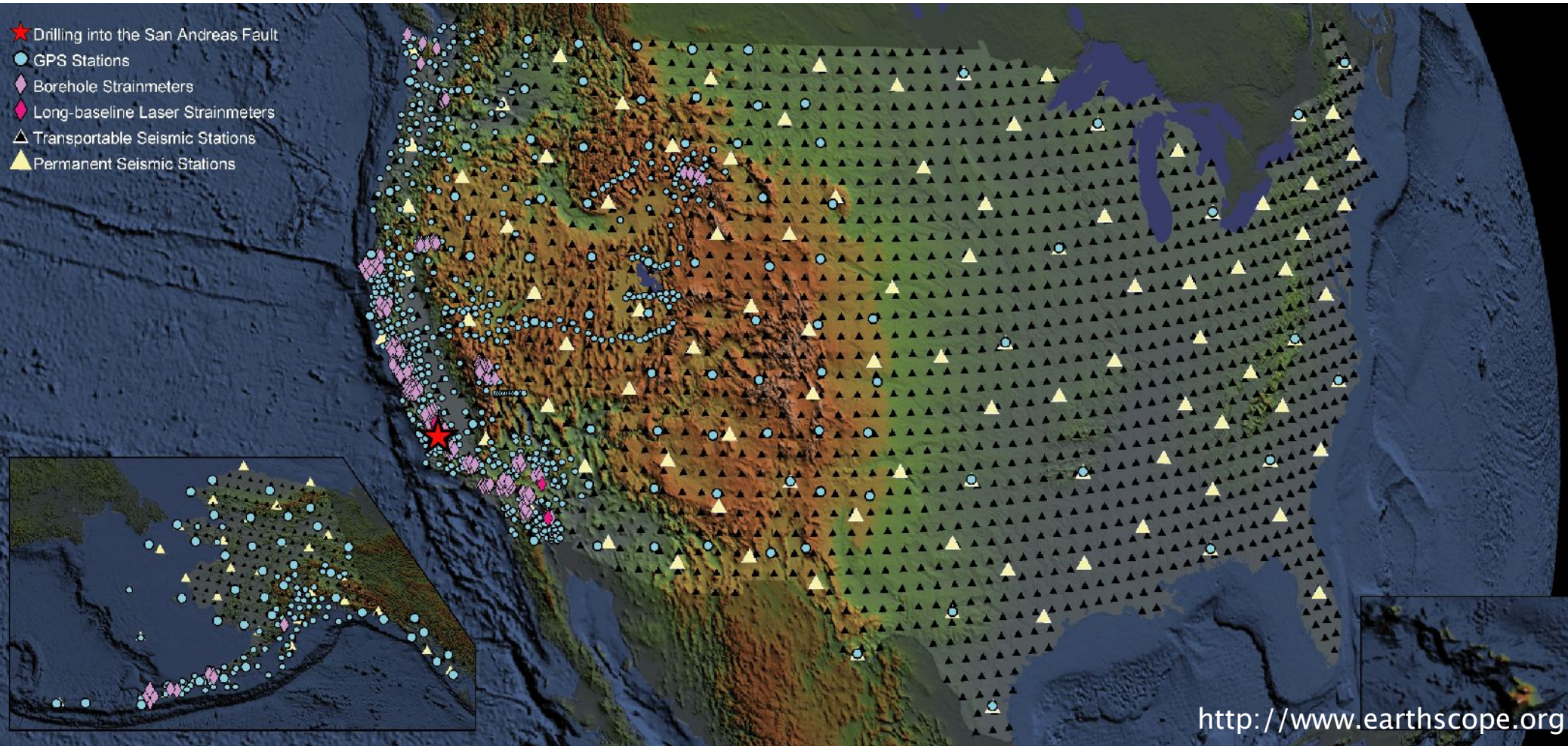
- What about connecting thousands of sensors all over the planet?



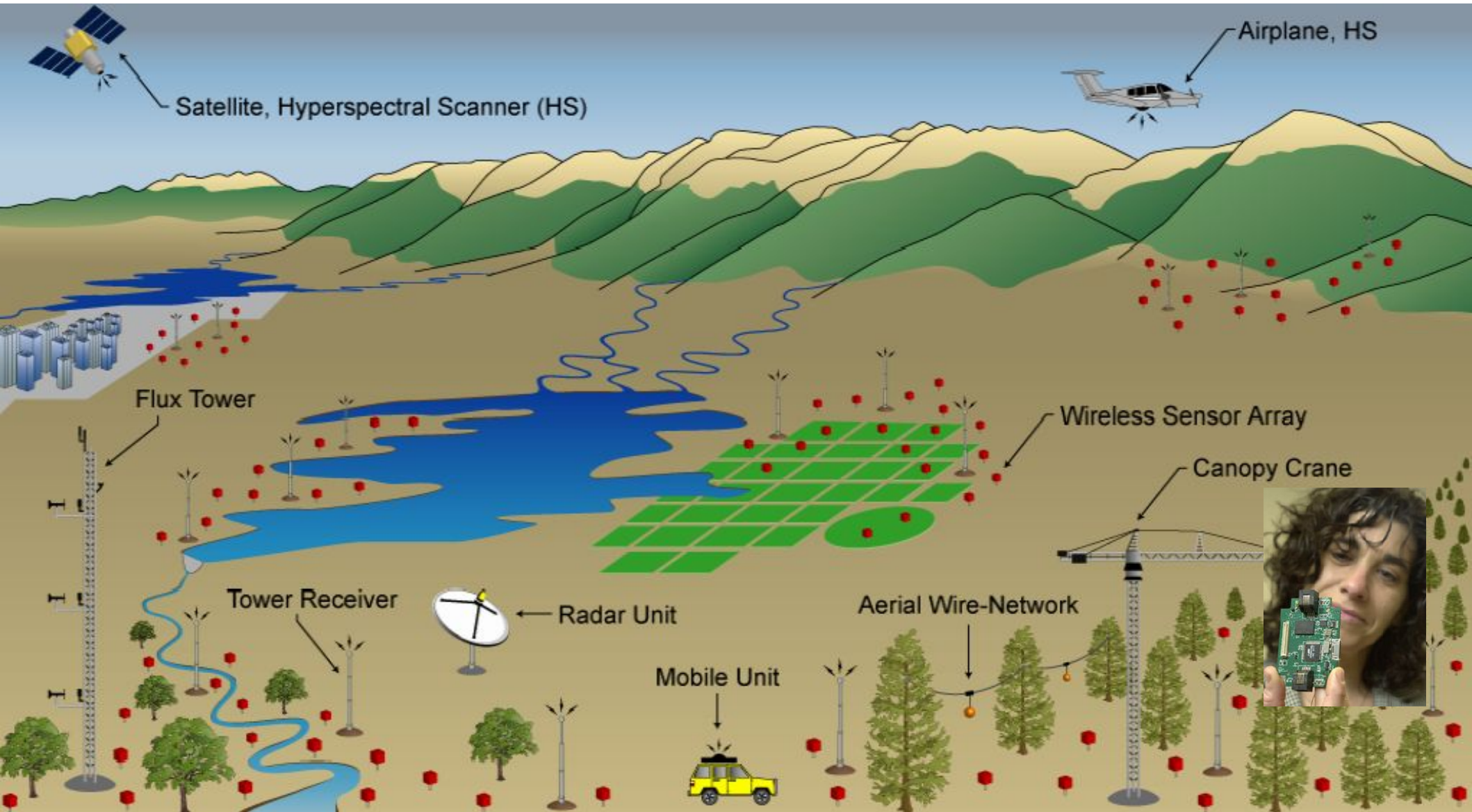
EarthScope (NSF)

15-year effort to understand earthquakes, volcanism, and plate movements in N. America

- 400 seismometers, 1000 GPS stations, 180 strainmeters



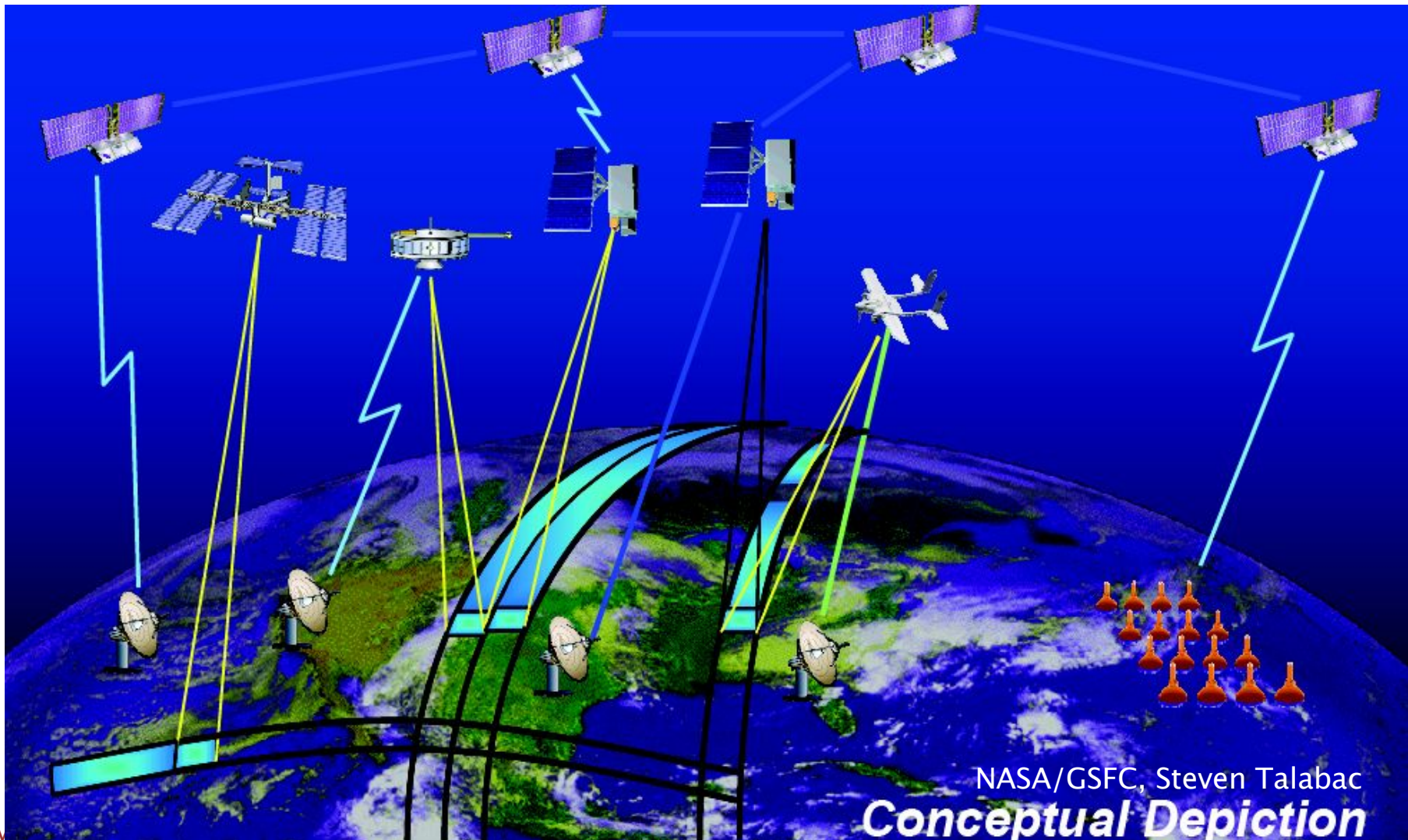
NEON (National Ecological Observatory Network)



SensorWebs

NASA Goddard Space Flight Center

Adaptive, model-based sensing with ground and space-based sensors



Some Open Questions

How do we harness vast numbers of real-time data sources on the Internet to undertake broad scientific studies?

- Enormous numbers of sensors are coming online.
- Leveraging them requires solving many hard systems problems.

Domain scientists demand a much richer interface than TCP/IP:

- Interchange between varying data formats.
- Discovering and naming data using physical/spatiotemporal/logical attributes.
- Annotation and tracking of data provenance.

How do we expose sensor data to the Internet?

- HTTP? SOAP? RDF? Raw bytestreams?

How do we open up sensor sources for queries or reprogramming?

- Need mechanisms to manage access, protect privacy, support multiple applications

Some Open Questions - continued...

How do we distribute query processing across the Internet?

- Sucking down all data to a central location is unlikely to work.
- Need to push query logic into the infrastructure.
- Must handle vast numbers of data sources and many simultaneous queries.

How does one discover and tap into real-time data sources?

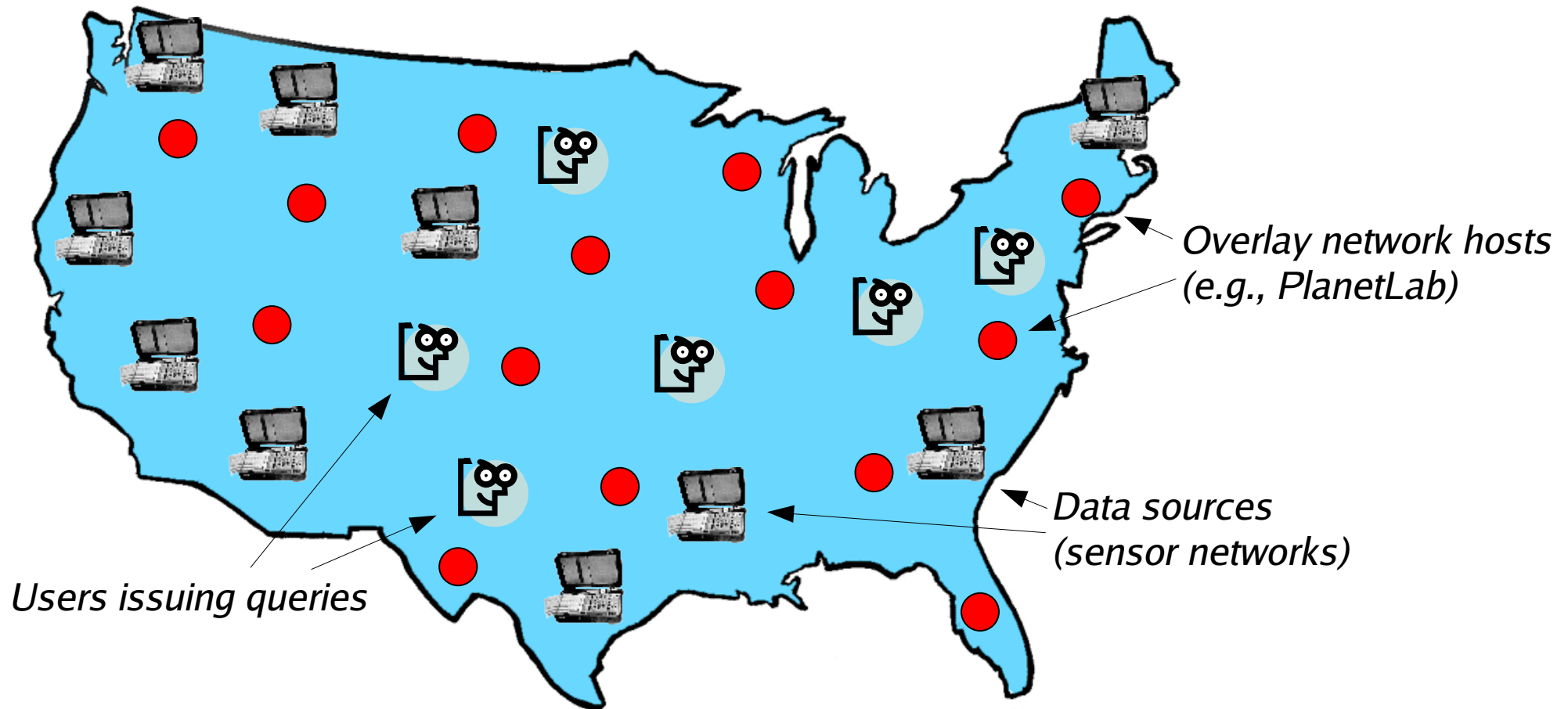
- Remember the canonical FTP site list of old?
- “sensors.google.com” -- Now serving 8,197,626,324,425 sensor networks!

The computer science systems community should be partnering with the “hard sciences” to address these problems.

- Those communities are building their own solutions anyway.
- This is a rich opportunity for us that could reinvigorate large-scale systems research.

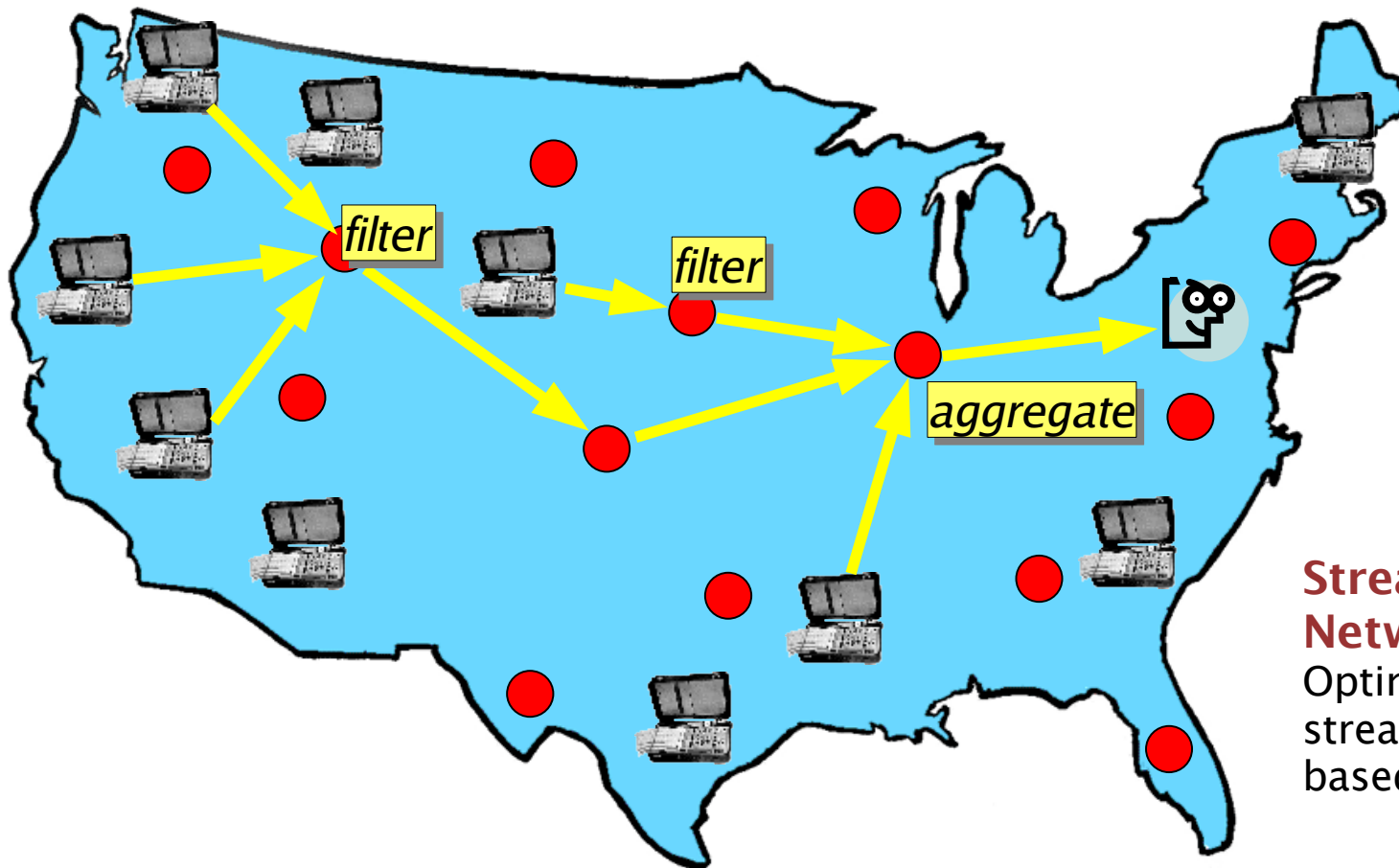
Hourglass and SBONs

Hourglass: Set of services to harness real-time sensor data across many geographically-distributed sources



Hourglass and SBONs

Hourglass: Set of services to harness real-time sensor data across many geographically-distributed sources



Stream-Based Overlay Network (SBON):
Optimize placement of stream-processing services based on network conditions

IP as the “fat narrow waist”

IP has amazed us with its versatility

- Runs on everything from huge data centers to PDAs and cell phones.

Ultimately, IP offers a degree of interoperability that is hard to ignore.

- Leads us to question whether a “parallel” (*but connected!*) universe of sensor networks is viable in the long term.
- History teaches us that lack of IP compatibility forebodes obsolescence
 - *Recall AppleTalk, DECNet, ...*

IP is probably not for *all* sensor networks

- But could be appropriate for an important class of them.

Some potential issues

Code size

- IP (and TCP, UDP, etc.) are relatively memory hungry
 - *Though TCP/IP runs on the Telos motes!*

Packet overhead

- TCP/IP Headers, footers, etc. eat into packet payload considerably
 - *Though 29 byte limit in TinyOS is probably antiquated*
- RTS/CTS-based MAC can mitigate this concern somewhat.
- More serious concern: **memory use** for buffering packets

Multiple addressing modes

- IP “mostly” supports unicast (at least in the wide-area Internet).
- IP defines the *semantics* for routing, not the implementation!

Breaking the end-to-end principle?

- Arguably the Internet has not followed this principle for some time now...

Where could IP fit in well?

Sensors with larger memory and CPU capacity

- Not all applications demand mote-sized platforms
- Much scientific instrumentation is already very heavyweight.

Applications where remote access is essential

- Direct IP connectivity to individual sensors, not just via a proxy

IP for routing within a sensor network cloud

- Truly bridge the gap from sensor nets to the rest of the world.

Keeping up with dominant technology trends

- What happens to SP when the next cool technology comes around?

Some questions worth pondering

Where do we think sensor node platforms are headed?

- Smaller, cheaper, lower power?
- Or moderate size, cost, but with greater CPU/memory capacity?

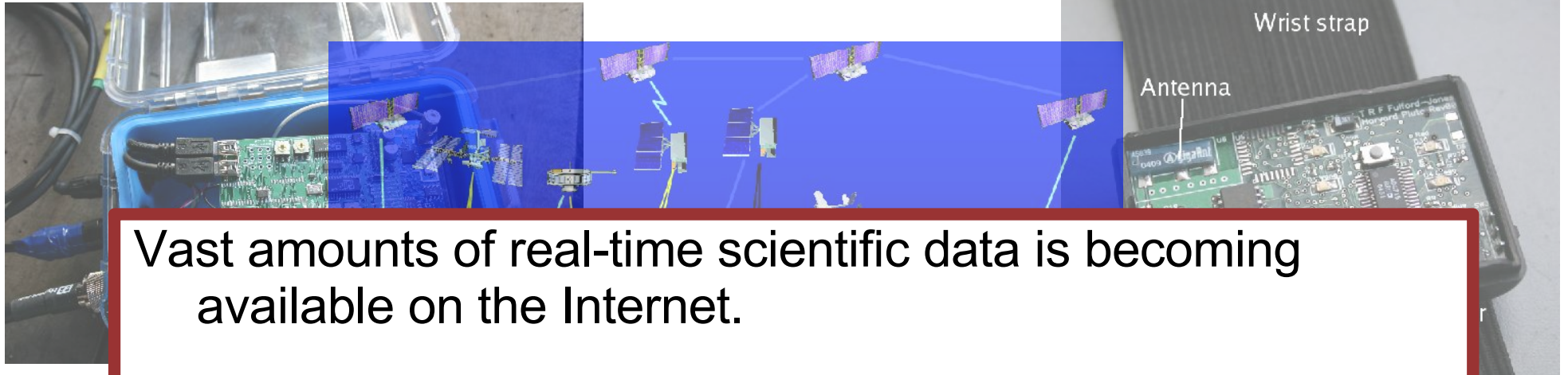
Where do we think applications are headed?

- Thousands of tiny disposable nodes scattered in every field?
- Supplanting smoke detectors and light switches?
- Replacing more conventional wired instrumentation?

Claim: These disparities will lead to a *divergence* of platforms.

- Industry clearly has certain applications and platforms in mind (evidence: Zigbee)
- Academics focused on more diverse range of requirements
- “Real scientists” may need something other than motes

Take Away Points...



Vast amounts of real-time scientific data is becoming available on the Internet.

Yet, domain scientists are currently forced to build up significant infrastructure to tap into it.

Now is the time to extend the Internet Architecture to incorporate sensor networks as a first-class citizen.

