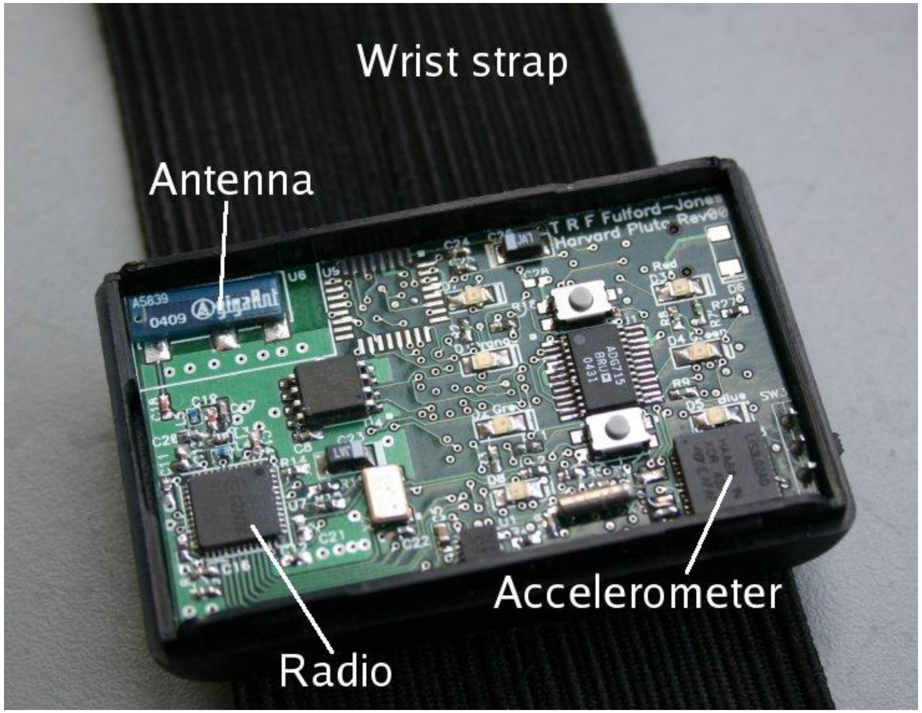
CodeBlue:
A Wireless Sensor Infrastructure for Medical Monitoring

Matt Welsh
Harvard University
School of Engineering and Applied Sciences

Introduction: Wireless Sensor Networks



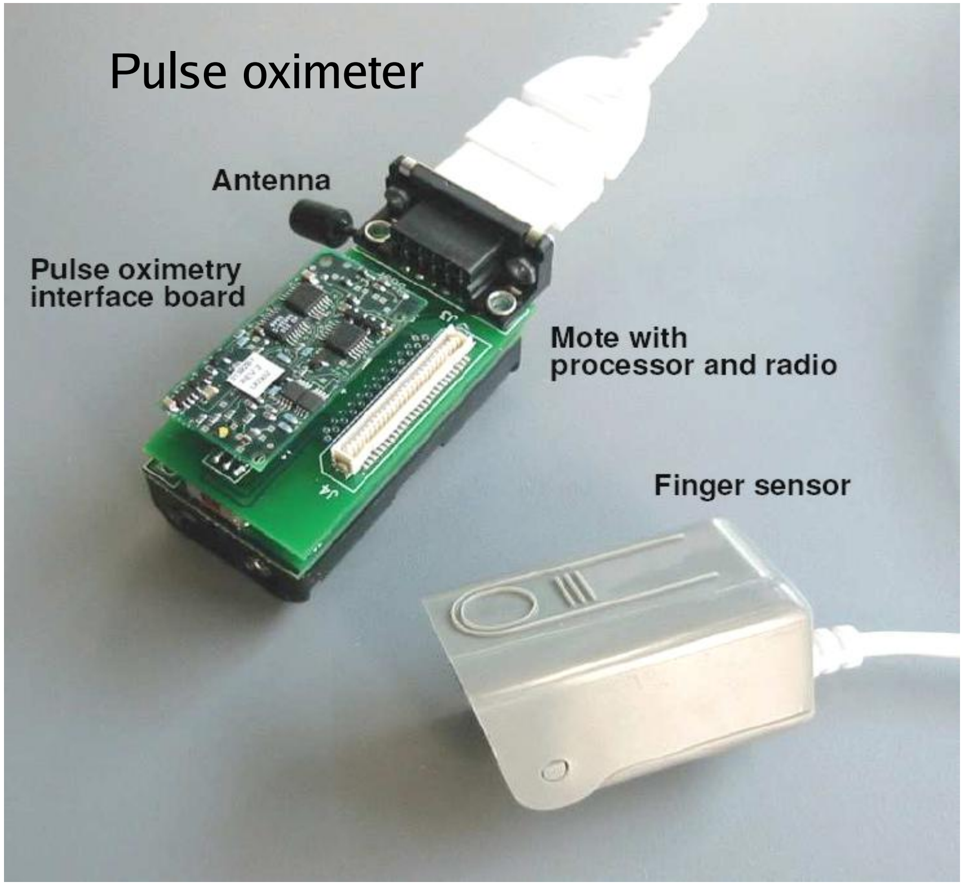
- Tiny, low-power, wireless sensors
- Minimal CPU, memory, and radio
  - 8 Mhz CPU, 10 KB RAM
  - 100 m radio range, 802.15.4/Zigbee
- Extremely low power
  - Battery lifetime of months to years

http://www.eecs.harvard.edu/~mdw/proj/codeblue

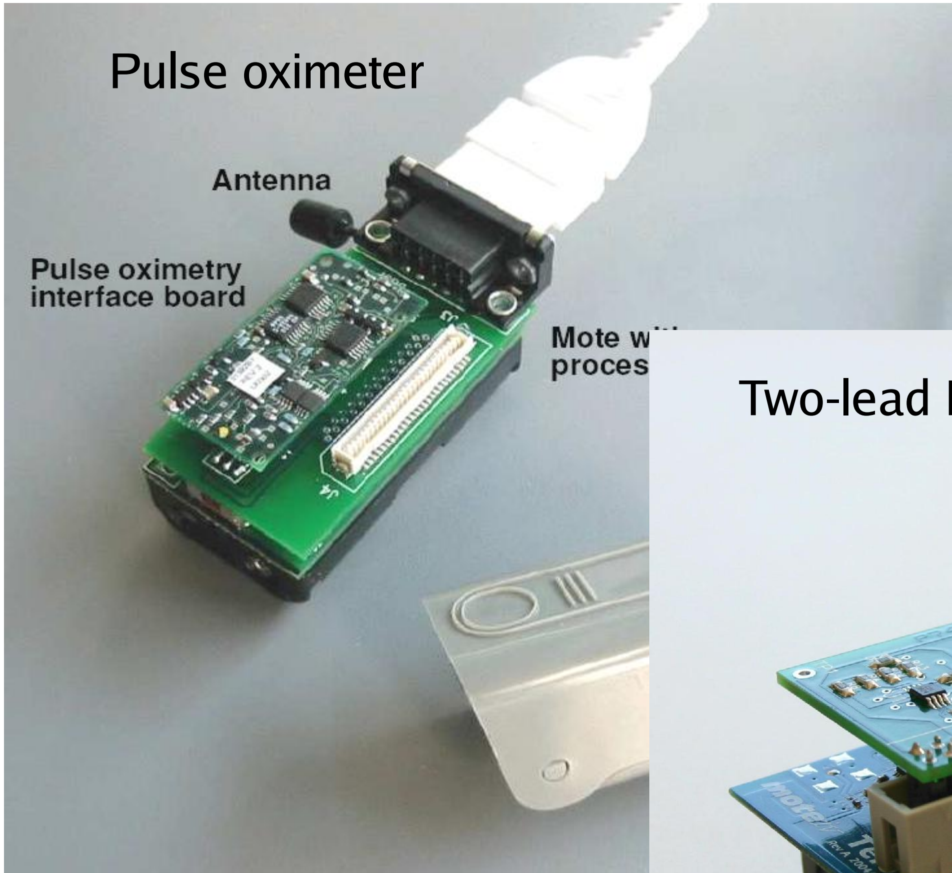
Potential Medical Applications

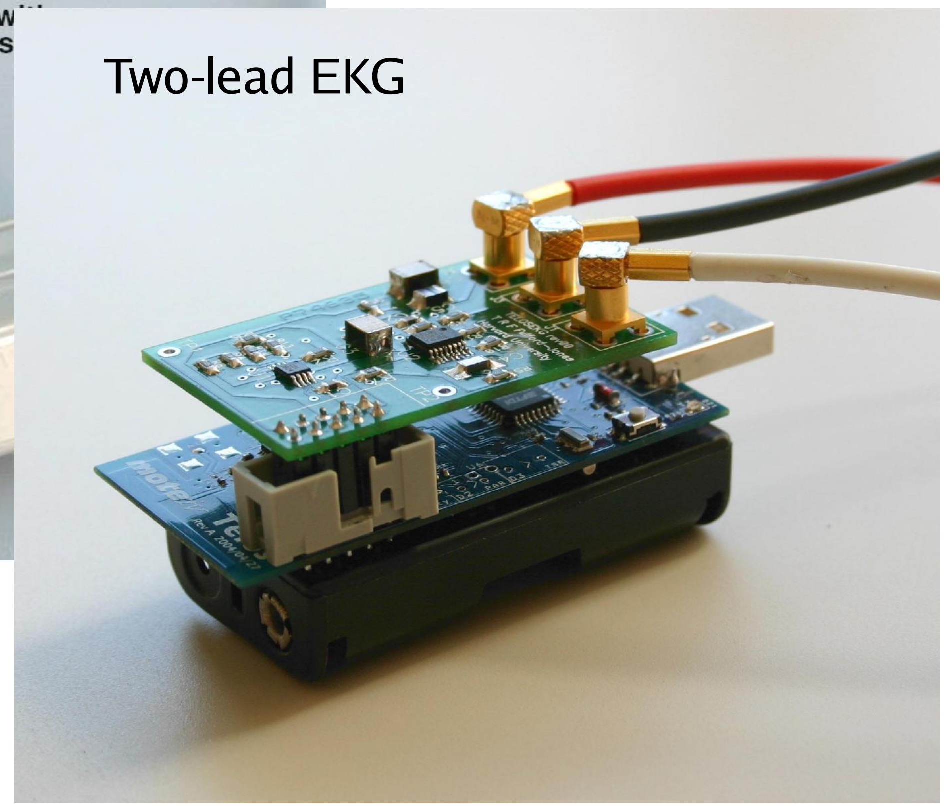
- Real-time, continuous patient monitoring
  - Pre-hospital, in-hospital, and ambulatory monitoring
  - Augment or replace wired telemetry systems
- Home monitoring for chronic and elderly patients
  - Collect periodic or continuous data and upload to physician
  - Allows long-term care and trend analysis
  - Reduce length of hospital stay
- Collection of long-term databases of clinical data
  - Correlation of biosensor readings with other patient information
  - Longitudinal studies across populations
  - Study effects of interventions and data mining

Some wireless sensors we have developed...

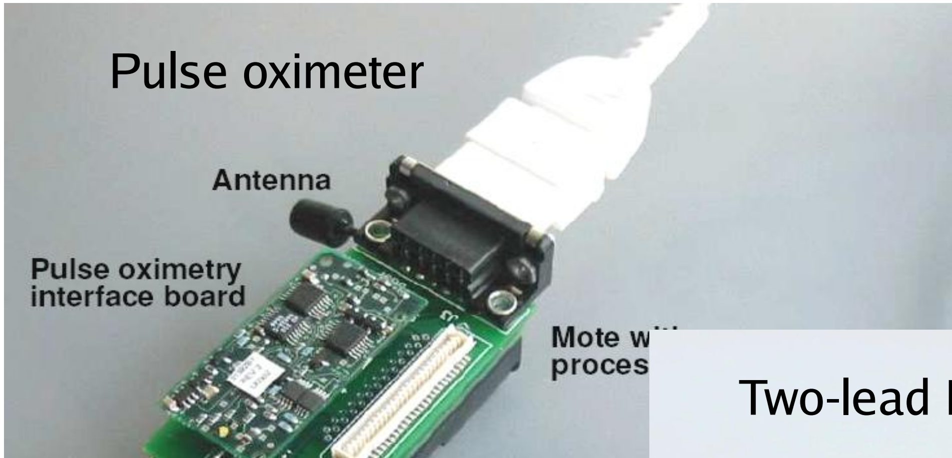


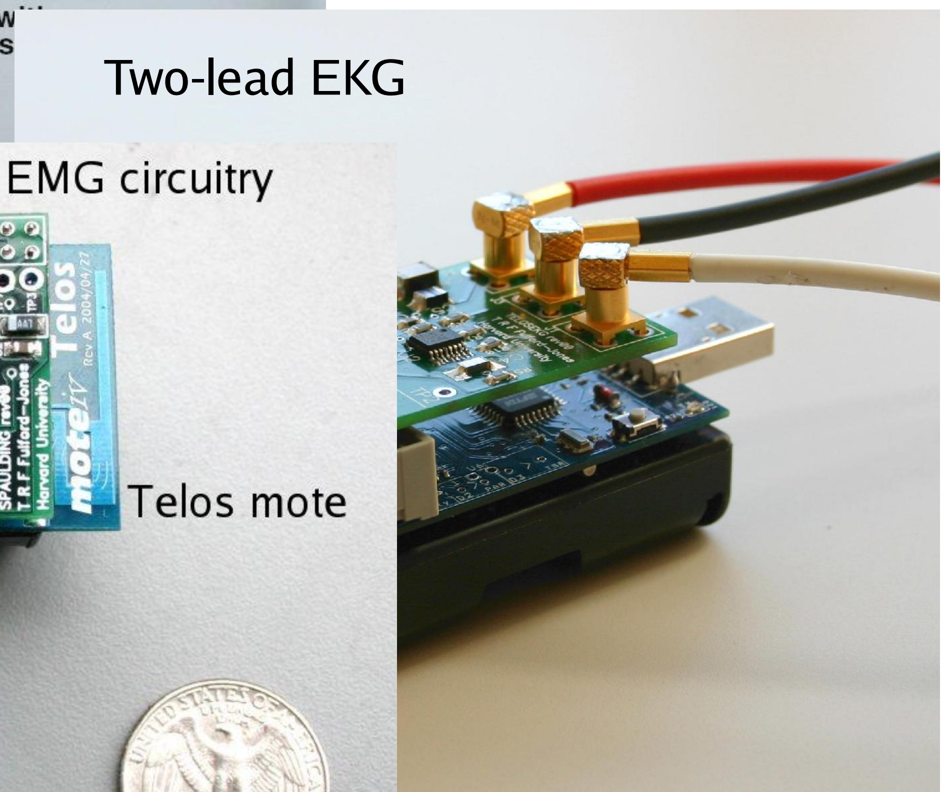
Some wireless sensors we have developed...

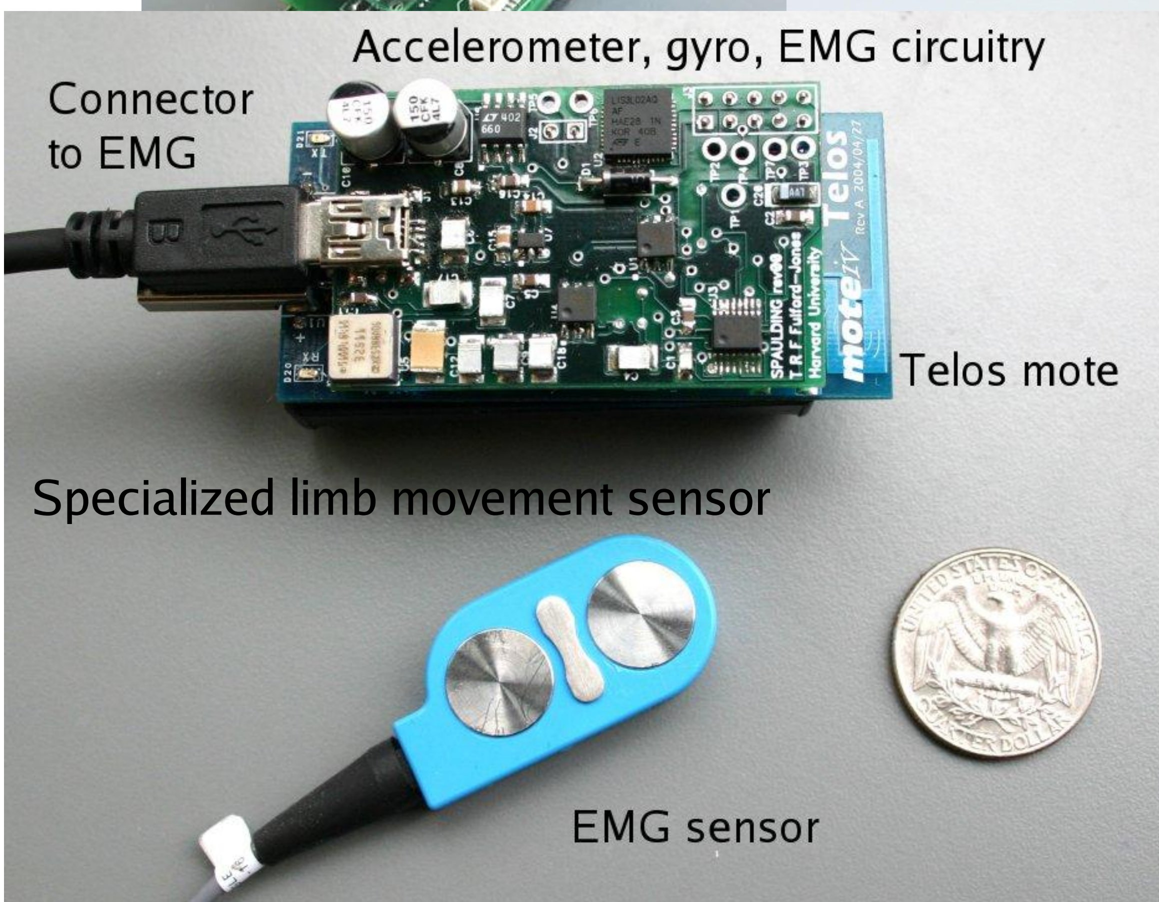




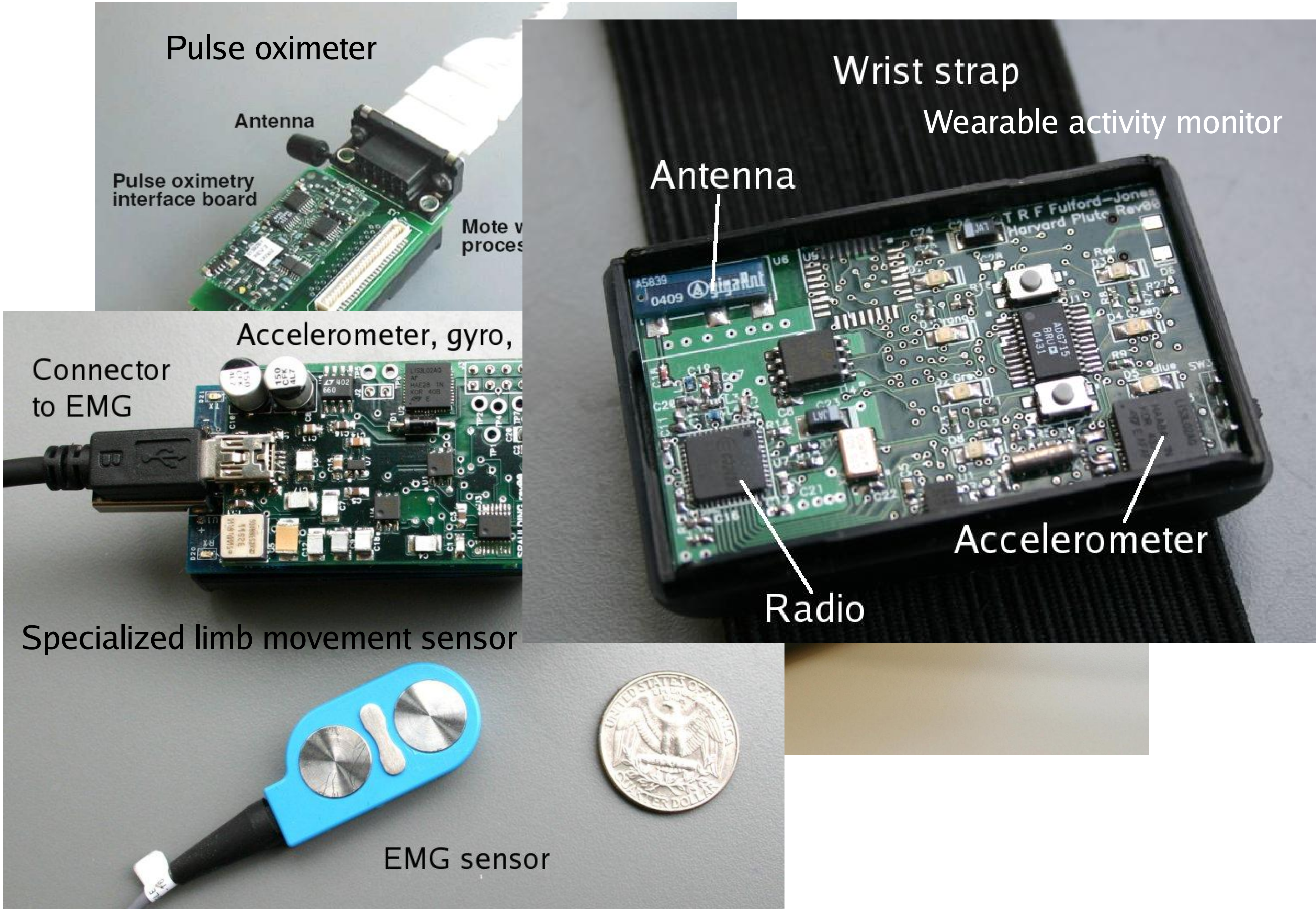
Some wireless sensors we have developed...



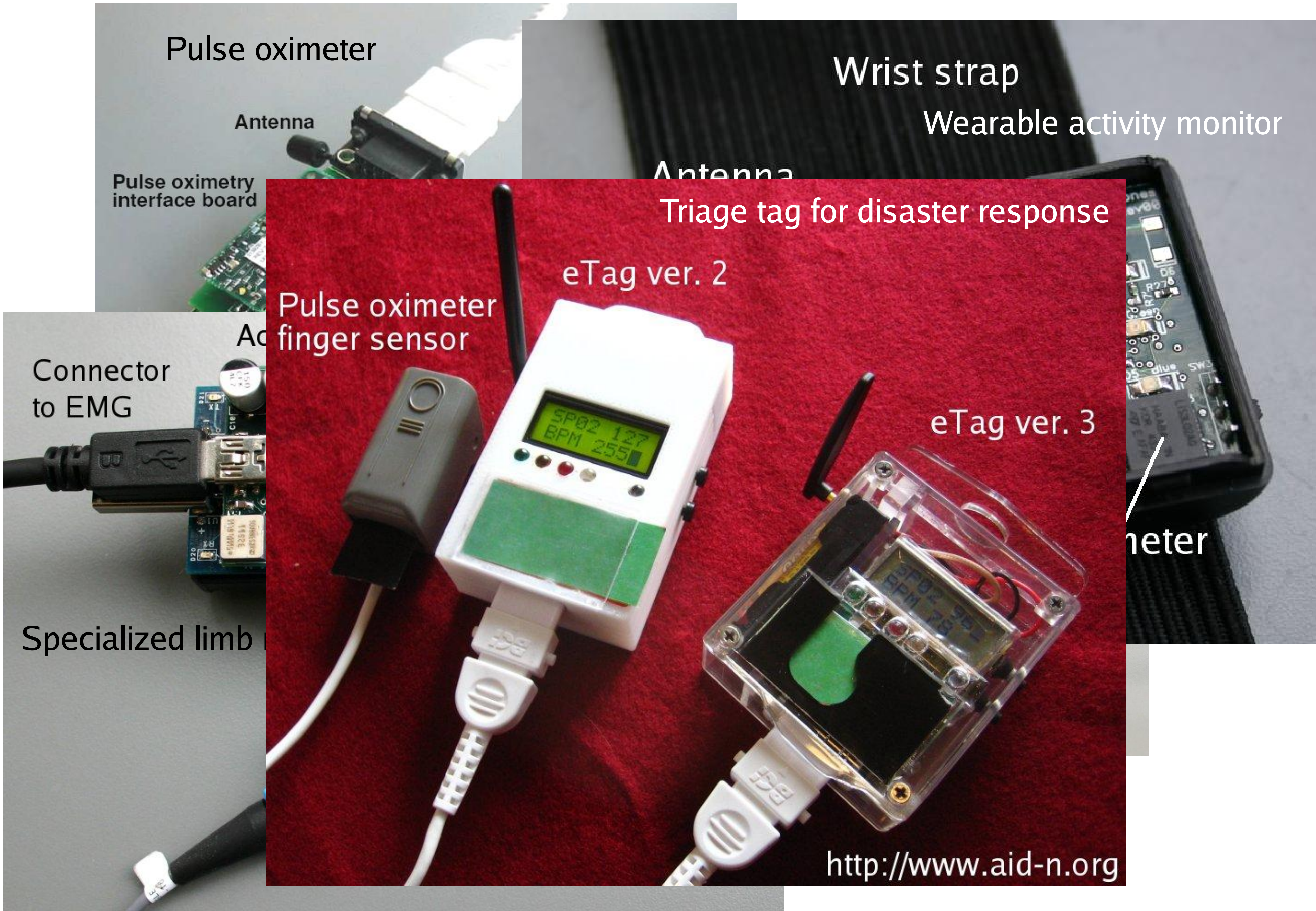




Some wireless sensors we have developed...

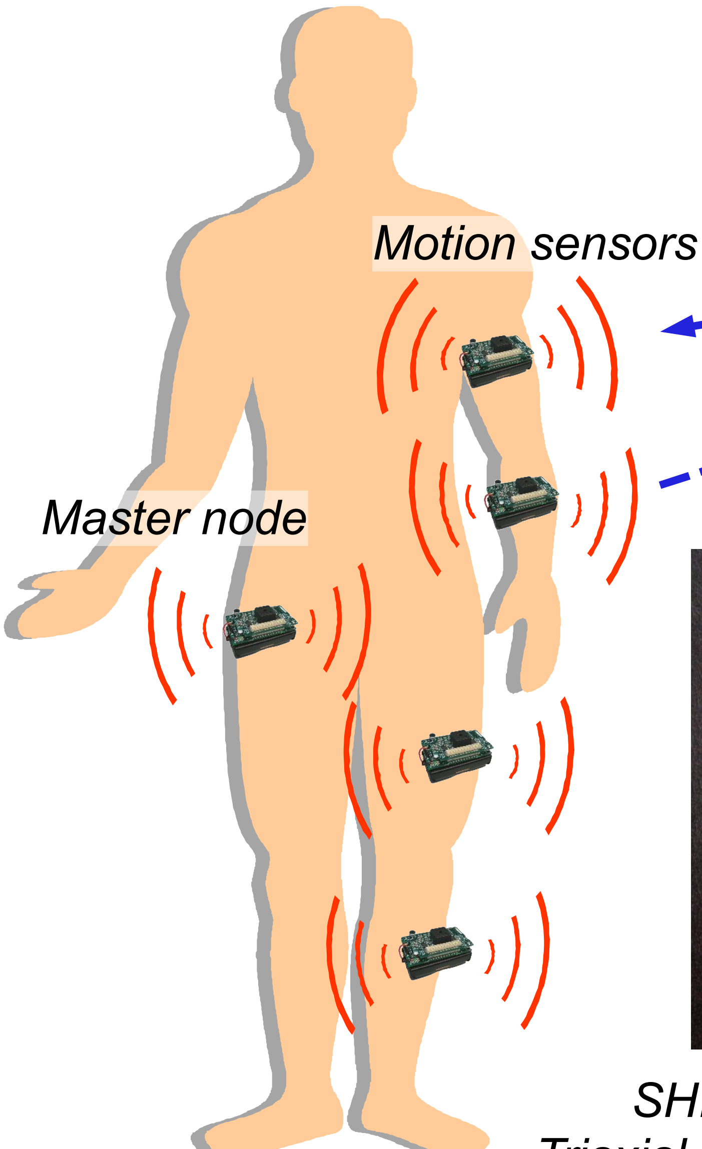


Some wireless sensors we have developed...

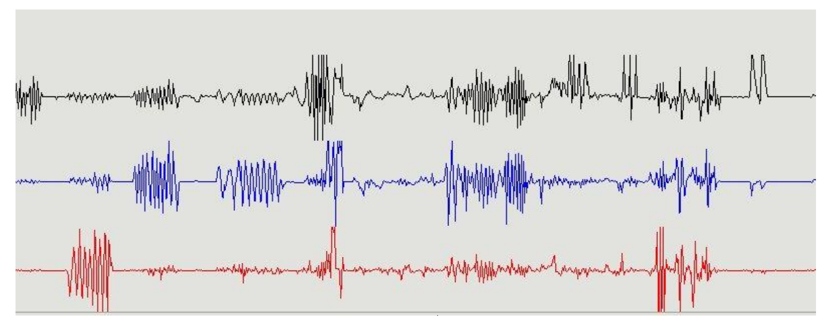


Applications: Parkinson's Disease and Stroke Rehabilitation Monitoring

with P. Bonato, Spaulding Rehabilitation Hospital



Data acquisition at laptop



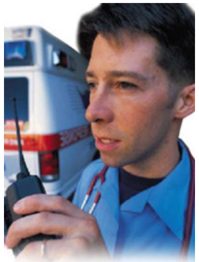
SHIMMER mote (Intel corp.)
Triaxial accelerometer and gyroscope

Feature extraction

↓

Classification (e.g., UPDRS scores)

The challenge

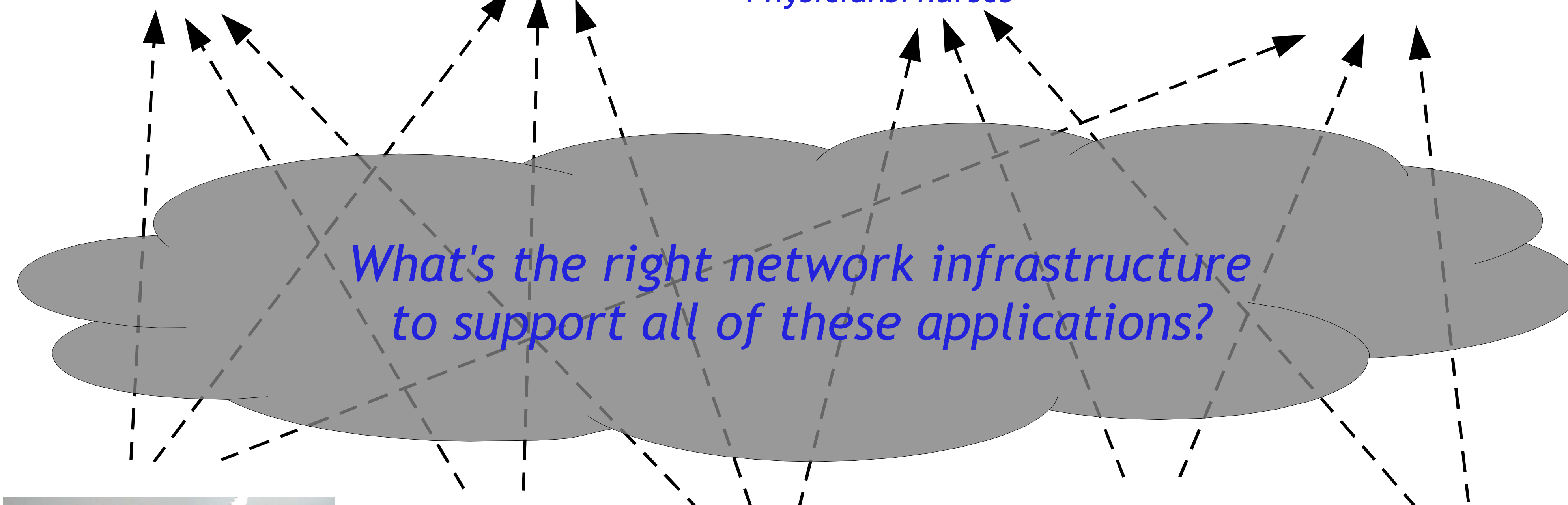


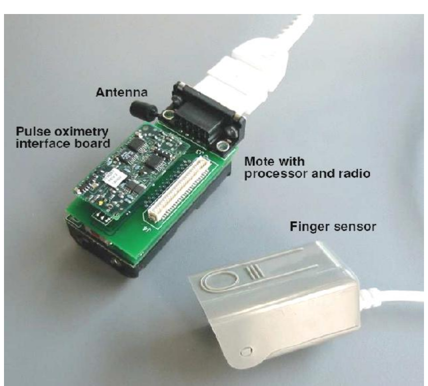
First responders

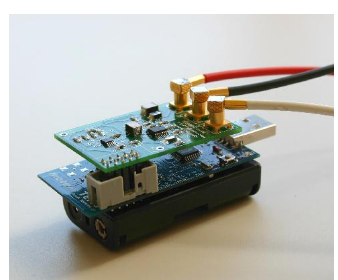
EMS / 911 Dispatch

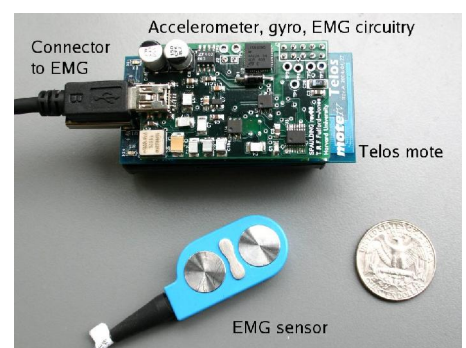
Physicians/nurses

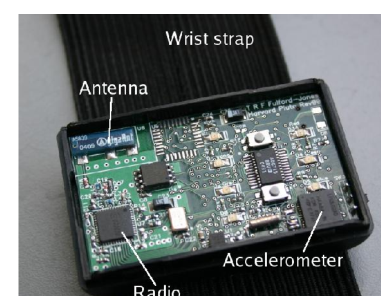
Hospital Information Systems



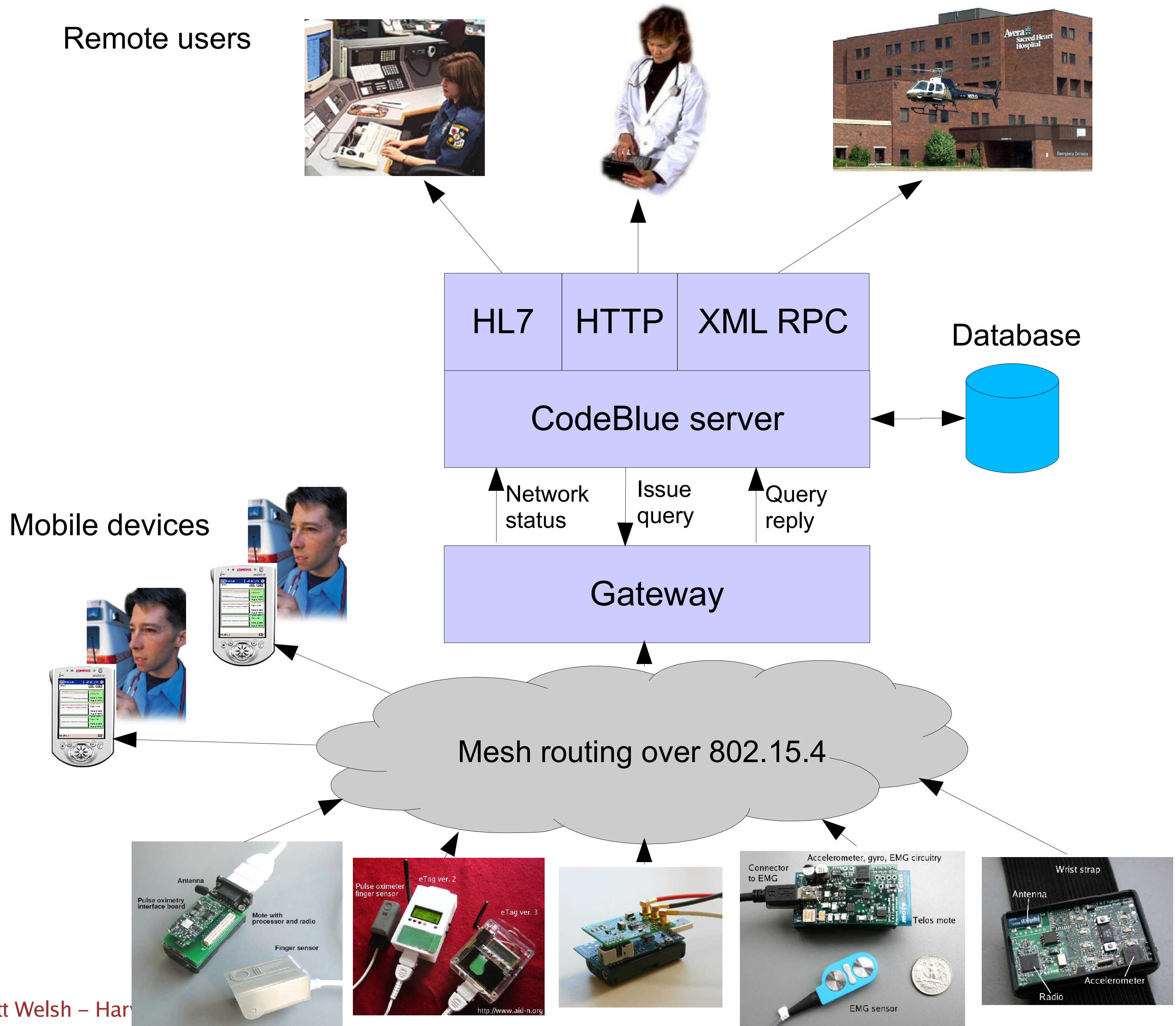








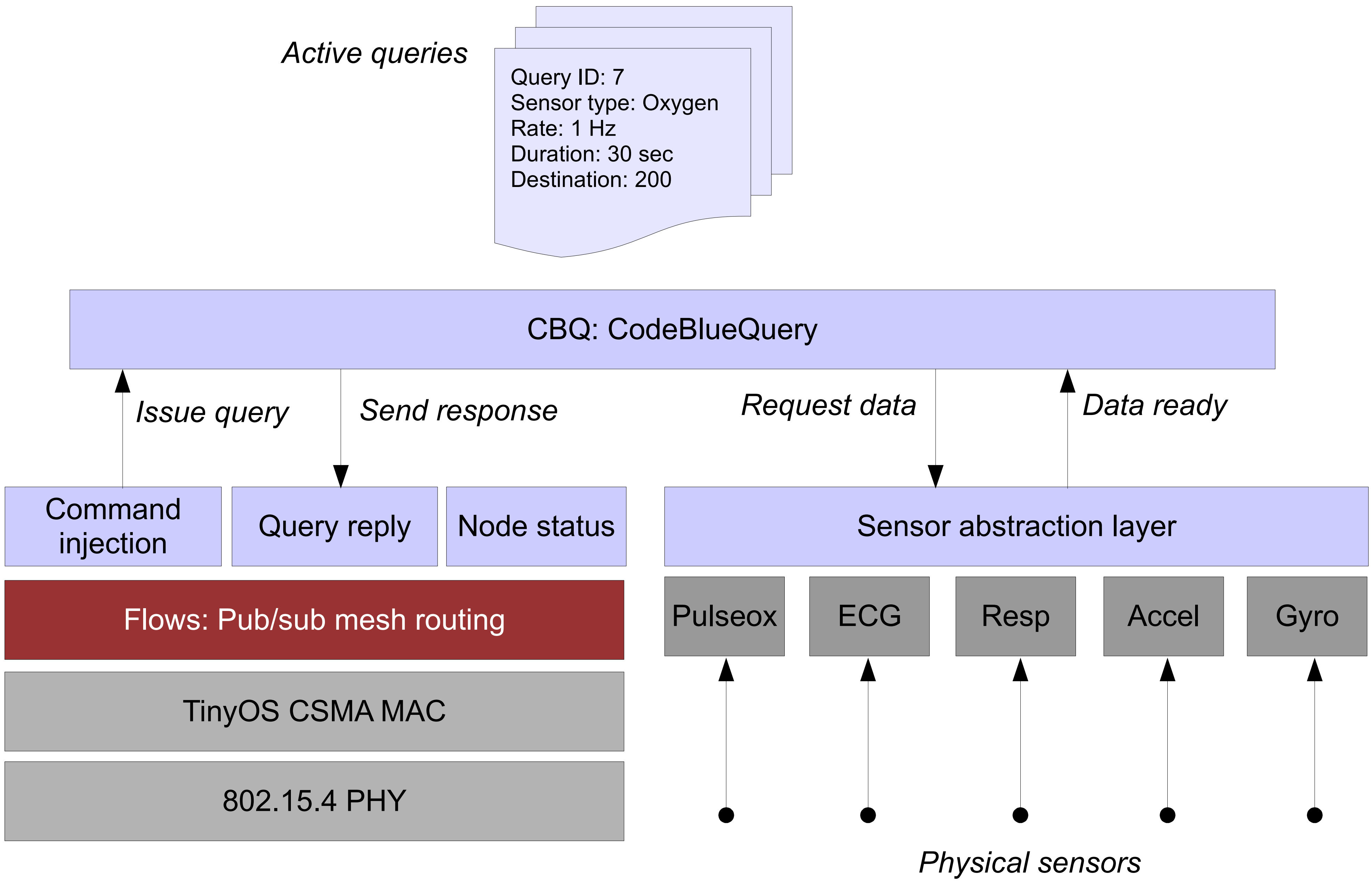
The Harvard CodeBlue Network Architecture



Sensor Network Limitations and Challenges

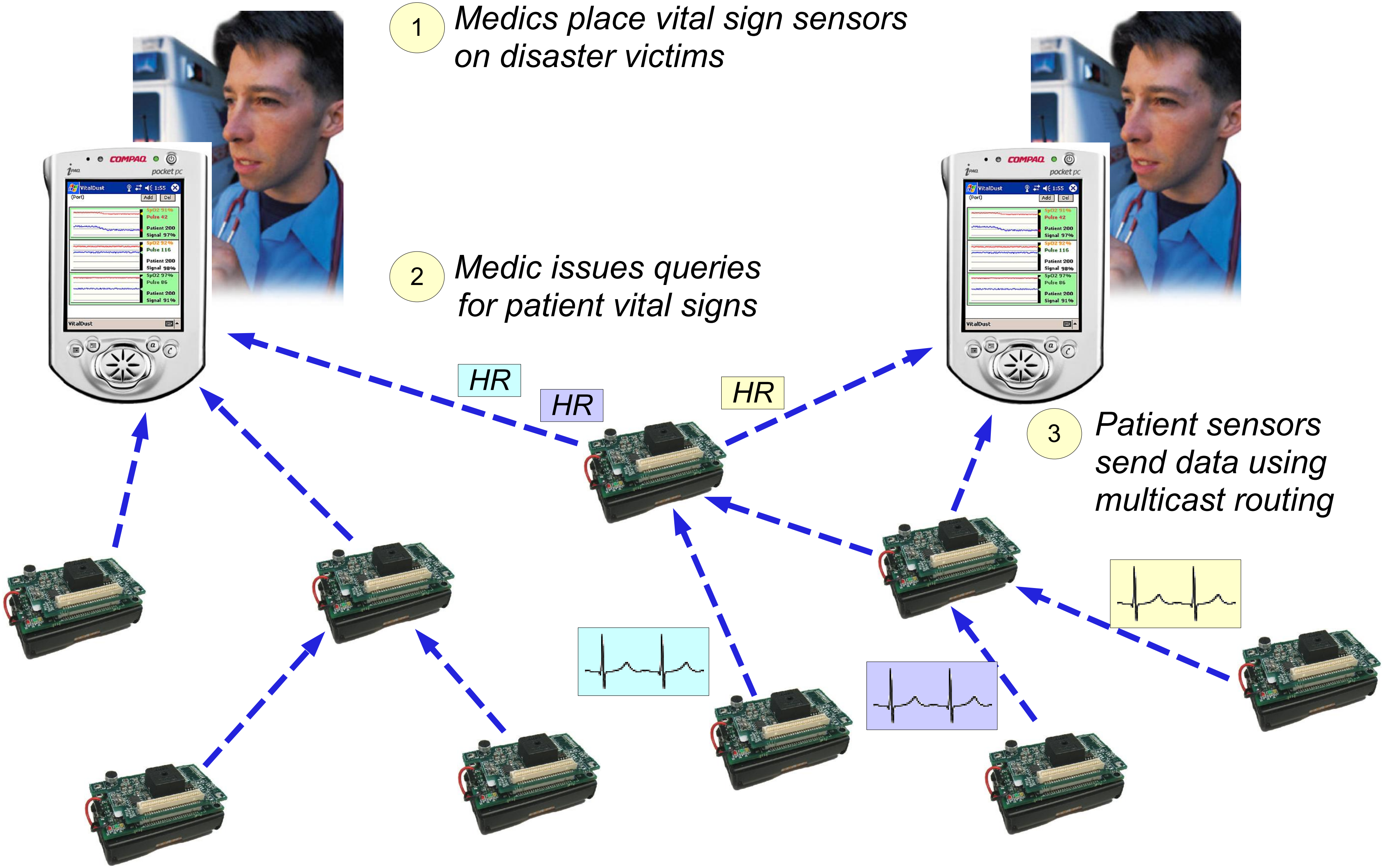
- The low power consumption of these devices implies very limited capabilities – well below that of a cell phone or PDA!
- Extremely limited radio bandwidth
  - Low-power, 802.15.4 radios – max PHY rate of 250 Kbps
  - Drops to 100 Kbps when taking MAC overhead and framing into account
  - A small number of sensors can rapidly saturate the channel
- Device mobility
  - Both patient sensors and receiving devices are moving around
  - Need to maintain good connectivity – in elevators, stairwells, etc.
- Multihop, multicast communications
  - Cannot always assume fixed infrastructure – e.g., disaster response
  - Multiple patient sensors may be monitored by multiple end users
- Limited device capabilities
  - Low CPU power (10 MIPS), tiny memories (10 KB of RAM)
  - Cannot run sophisticated algorithms involving lots of computation or memory usage

CodeBlue Software Architecture

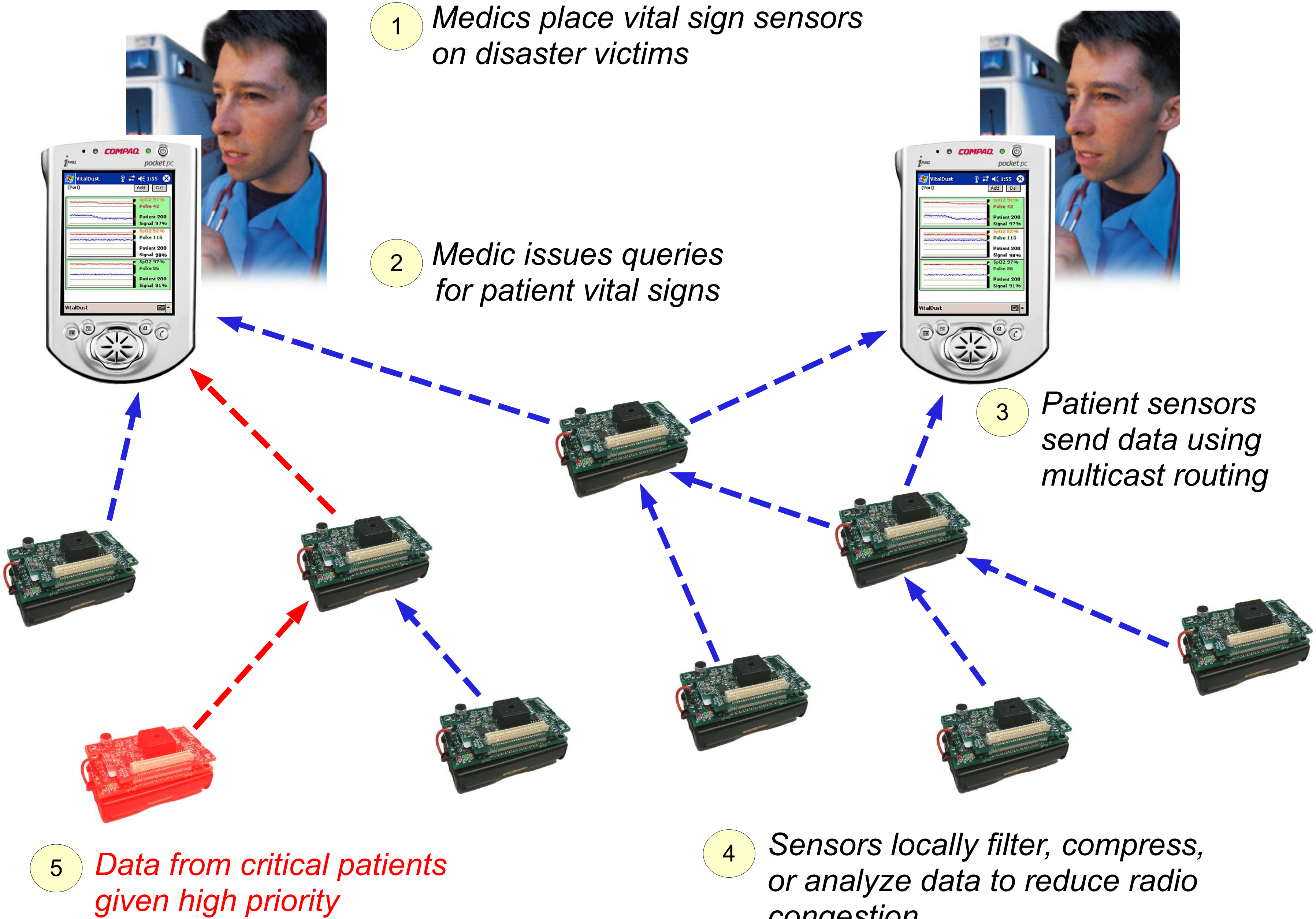


CodeBlue Mesh Routing

1. *Medics place vital sign sensors on disaster victims*
2. *Medic issues queries for patient vital signs*
3. *Patient sensors send data using multicast routing*
4. *Sensors locally filter, compress, or analyze data to reduce radio congestion*

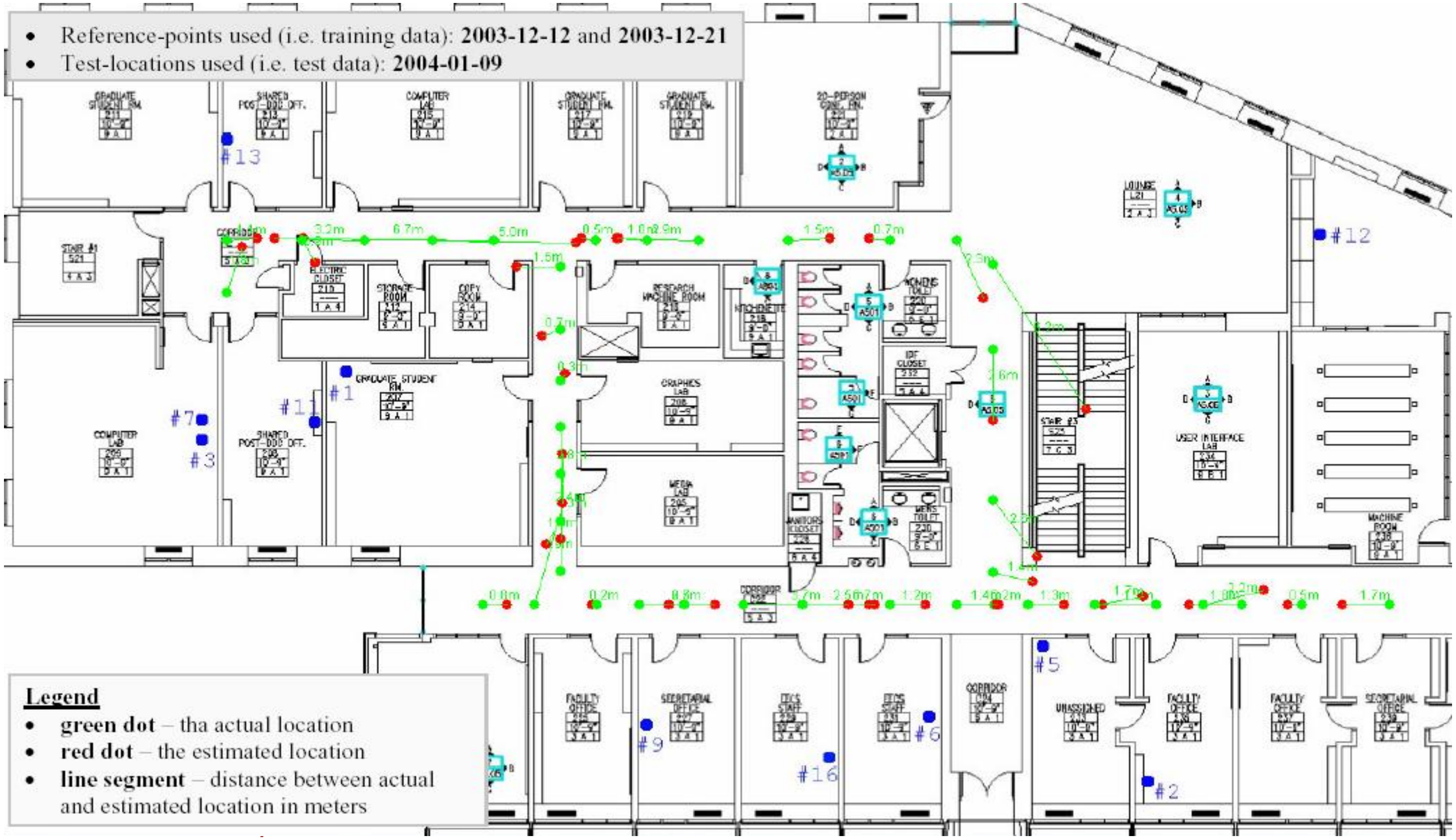


CodeBlue Mesh Routing



MoteTrack: RF-Based Localization

- Collect RF signal "signatures" from various points in building
  - Use MoteLab testbed with 30 beacon nodes
  - Similar to RADAR scheme for 802.11 networks, with much higher density
- Nodes compute location by comparing to stored signatures
  - Centroid of weighted signature distance from known points
- Good results: 80th percentile error of *1 meter*



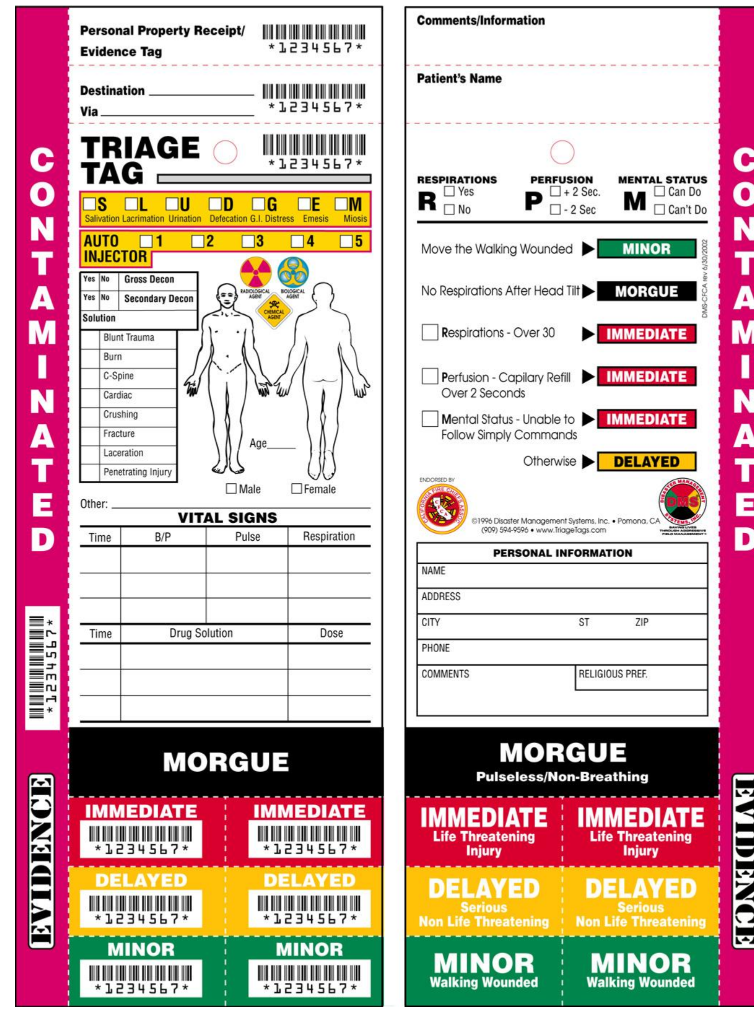
GUI for Real-Time Patient Tracking

Map showing location (MoteTrack)

Applications: Emergency medicine

- Large accidents, fires, terrorist attacks
  - Normal organized community support may be damaged or destroyed
  - Large numbers of patients, severe load on emergency personnel

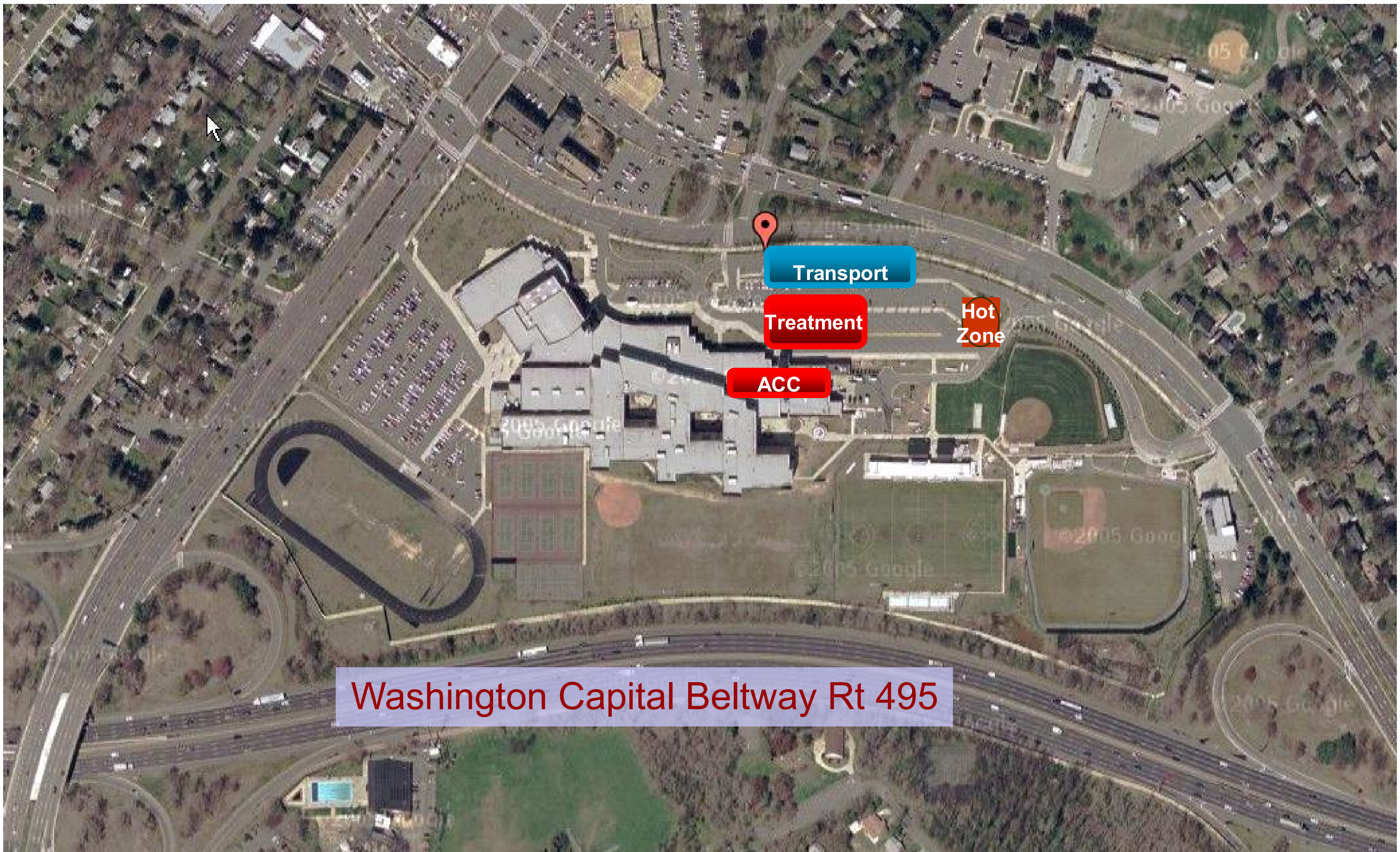
- Manual tracking of patient status is difficult
  - Current systems are paper, phone, radio based
  - No real-time updates on patient condition



DHS Disaster Drill, August 2006

DHS Disaster Drill, August 2006

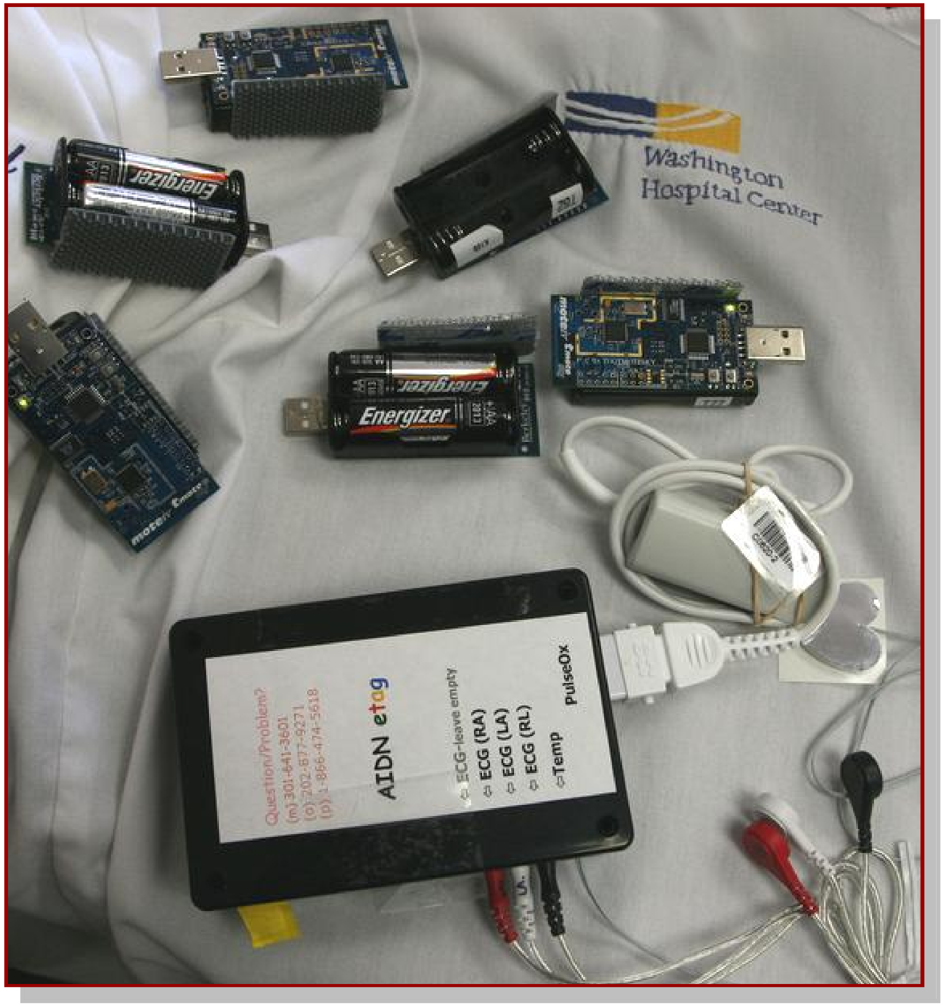
- Bus accident at junction of Washington Capital Beltway Rt 495
- Assumption: Hospitals within 15 mi radius reached surge capacity



Applications: Inpatient Monitoring

Electronic Triage Tag, version 2

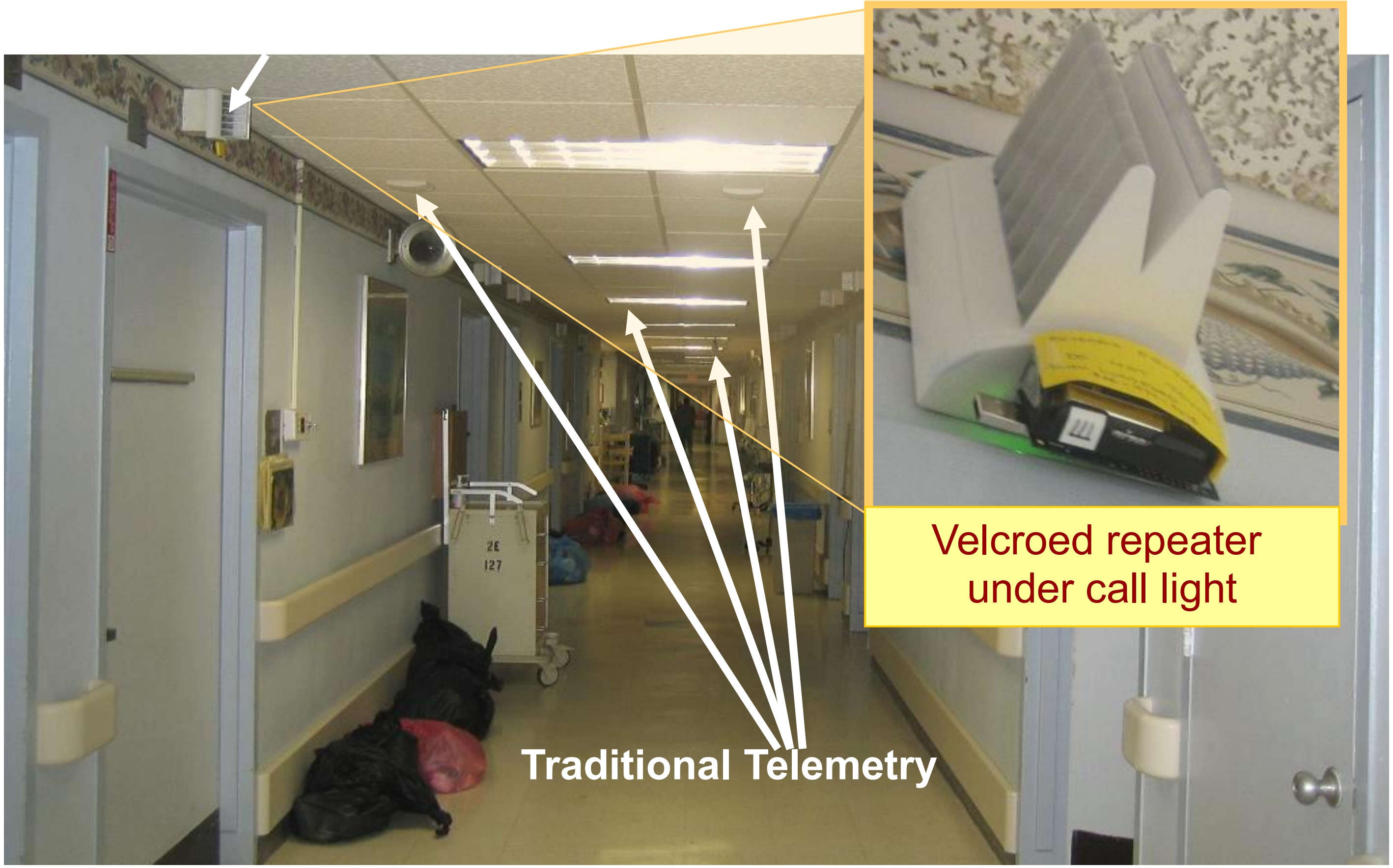
- 250kbps 2.4GHz IEEE 802.15.4
- 10 K RAM, 48K Flash
- Sampling Rates
  - Pulse Oximetry: 1 Hz
  - EKG: 250 Hz
  - Temperature: 1Hz
- Battery life: 2.5 days on 2 AA batteries



Washington Hospital Burn Unit, Feb 2007

Monitoring patients for one week 24 hours per day in a burn center introduces new hurdles

- Departmental approvals needed
  - IT department: wireless interference
  - Clinical engineering: device safety
- Network noise
  - Wireless devices
  - Equipment
  - Doors
  - Walls, hallways
- Sensor noise
  - Moving patients
  - Physiologic artifacts
- Installation of Repeaters



Current Status

- First prototype of CodeBlue protocol framework is complete
  - TinyADMR for multicast routing
  - MoteTrack for indoor localization
  - Java-based GUI for real-time visualization
  - Web services interface for application integration
- Range of medical sensors based on motes
  - Pulse oximeter, EKG, accelerometer/gyro/EMG board
  - Pluto custom mote for wearable applications
- Disaster drill completed in collaboration with JHU AID-N project
- Pilot at Washington Hospital Burn Unit in Feb 2007
- Deployment at Beth Israel Deaconess ED in coming months
- Ongoing collaboration with Spaulding Rehabilitation Hospital

Some Lessons and Take Away Points...

- New wireless sensor technologies have tremendous potential to impact many aspects of healthcare.
  - But, there is still a large gap between the existing technology and the needs of medical practitioners.
  - We aim to close this gap by developing a robust and flexible **sensor network infrastructure** to tie wireless sensors into a range of medical applications.
- Medical monitoring raises new challenges for wireless sensors
  - Hardware platforms, data fidelity, bandwidth requirements, fault tolerance, and security
  - Prior work in mobile ad hoc networking focuses on simulations, as well as very different application requirements (i.e., Web browsing)
  - Should re-examine the fundamental assumptions behind this work for medical use
- Need more real world experience deploying, testing, and evaluating these sensors in real clinical settings.
  - Our disaster drill and hospital tests have been promising, but shed light on many open technology and policy questions
  - Need a better path towards new technology adoption in healthcare to enable innovation

Acknowledgments

- This work is supported by grants from the National Science Foundation, National Institutes of Health, and US Army

- We are also supported by industrial sponsors:
  - Intel Corporation
  - Microsoft
  - Sun Microsystems
  - Siemens

- The CodeBlue hardware deisgns and software are available for download:

http://www.eecs.harvard.edu/~mdw/proj/codeblue

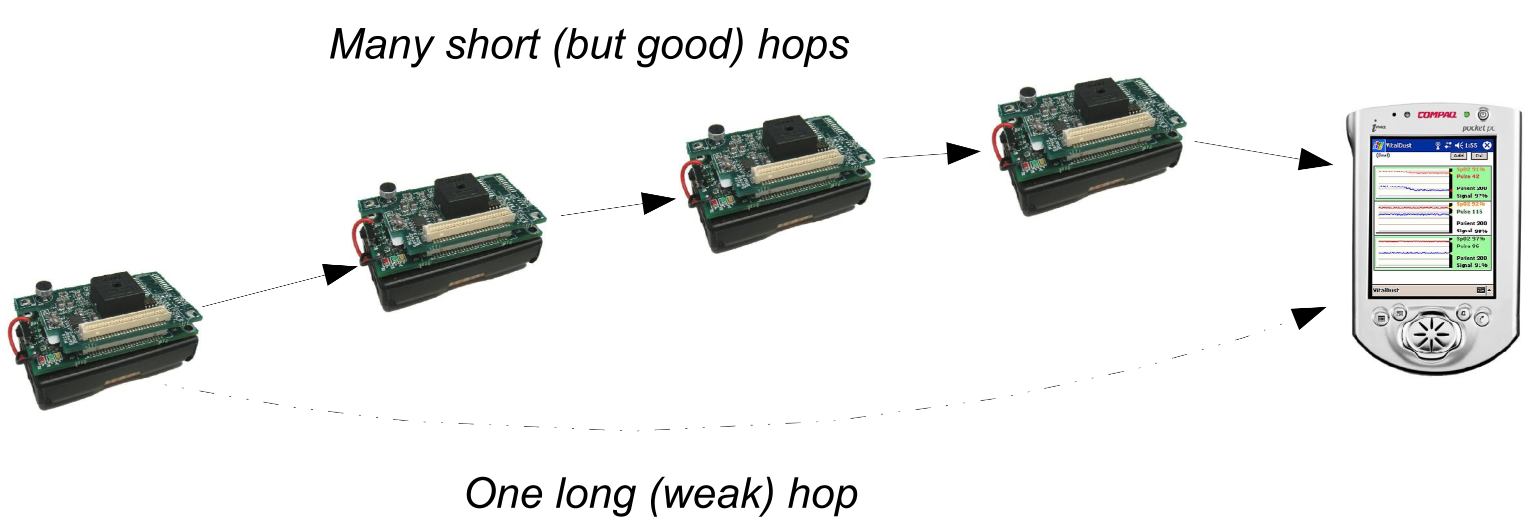
mdw@eecs.harvard.edu

CodeBlue Mesh Routing Protocol

- CodeBlue requires an *ad hoc* multicast routing protocol
  - *Ad hoc*: No need for fixed infrastructure, forms routes “on demand”
  - Multicast: Data from each sensor can be received by multiple end-user devices
- Ad hoc routing has been extensively studied in wireless environments
  - AODV, CSR, DSDV, ODMRP, ADMR, ...
  - Much of this work done in simulation assuming perfect radio links
  - Implementations primarily focus on laptops or PDAs with 802.11 radios
- What's new here?
  - Very limited radio bandwidth: protocol overhead is a big deal
  - Real radios with lossy, asymmetric links
  - Nodes have very small memory (< 10KB) and limited computational power

Our first protocol: TinyADMR

- Based on *Adaptive Demand-driven Multicast Routing (ADMR)*
  - [Jetcheva and Johnson, Proc. MobiHoc 2001]
  - Mature, well-designed multicast protocol for wireless networks
- We implemented the protocol on motes using TinyOS
  - Lots of changes required to get ADMR to work well on this platform
- Route selection metric:
  - Minimum-hopcount path performs poorly (selects short routes with bad links)

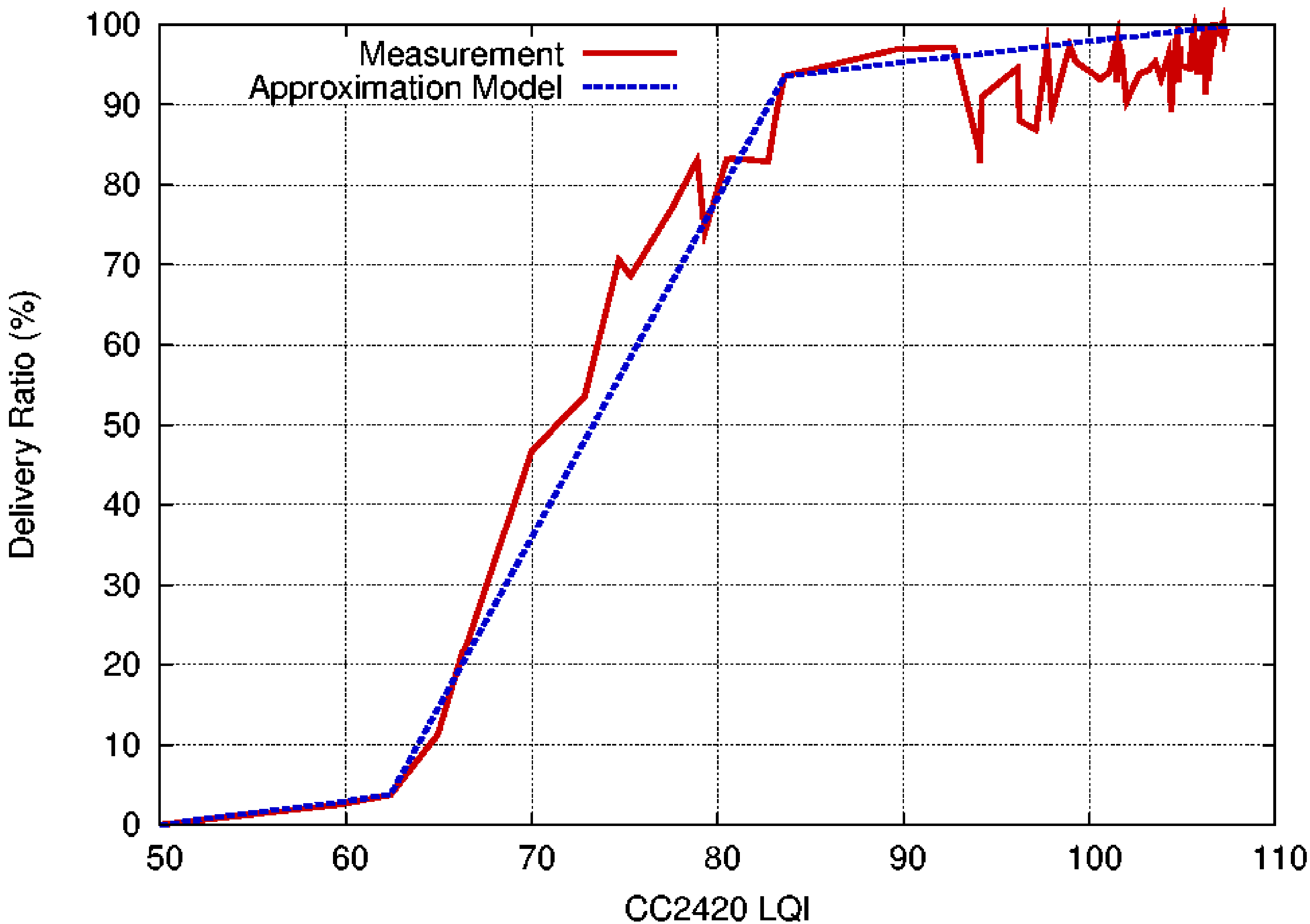


Our first protocol: TinyADMR

- Based on *Adaptive Demand-driven Multicast Routing (ADMR)*
  - [Jetcheva and Johnson, Proc. MobiHoc 2001]
  - Mature, well-designed multicast protocol for wireless networks
- We implemented the protocol on motes using TinyOS
  - Lots of changes required to get ADMR to work well on this platform
- Route selection metric:
  - Minimum-hopcount path performs poorly (selects short routes with bad links)
- Link asymmetry:
  - Node A can hear Node B does not imply that Node B can hear Node A
- Memory constraints:
  - ADMR keeps several tables with state about active paths and network neighbors
  - In a large network this state will rapidly consume available memory

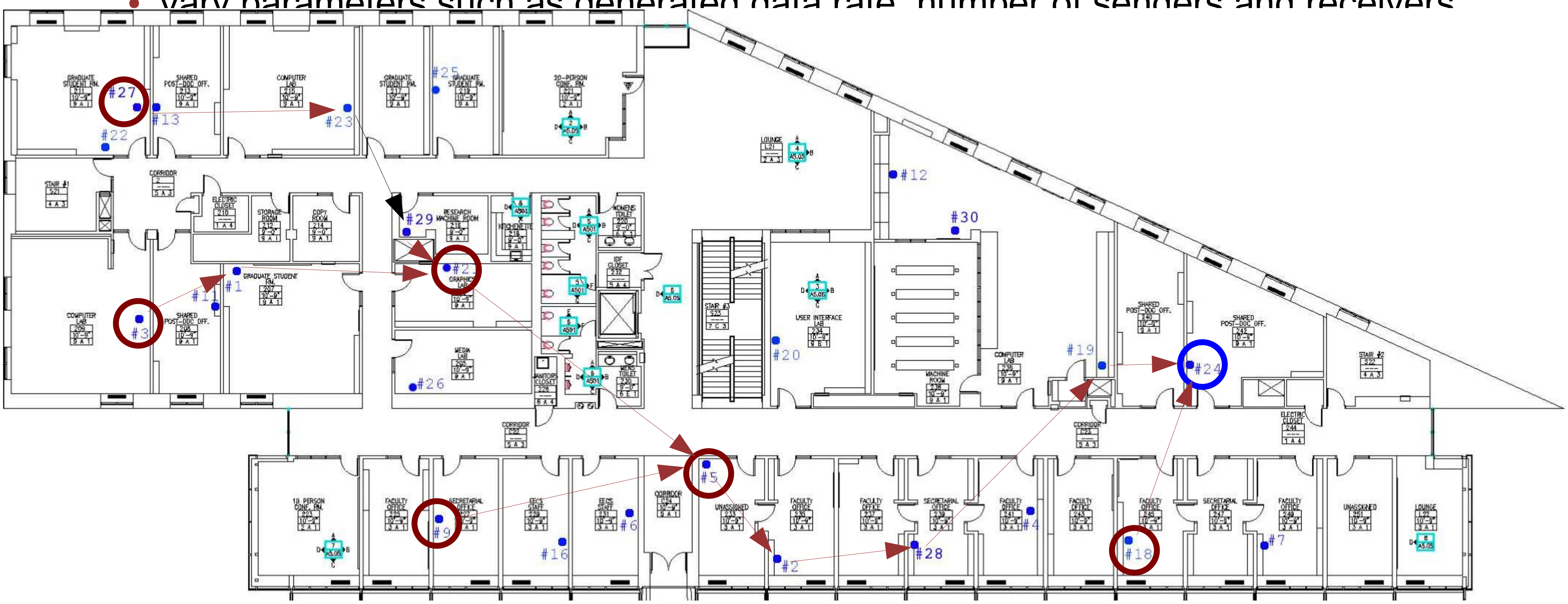
TinyADMR Route Selection

- We make use of CC2420 Link Quality Indicator (LQI) metric:
    - *Indicates ability of radio to decode start symbol of packet*
  - LQI is highly correlated with packet delivery ratio
  - Can be measured with a single packet reception (no probing traffic required)



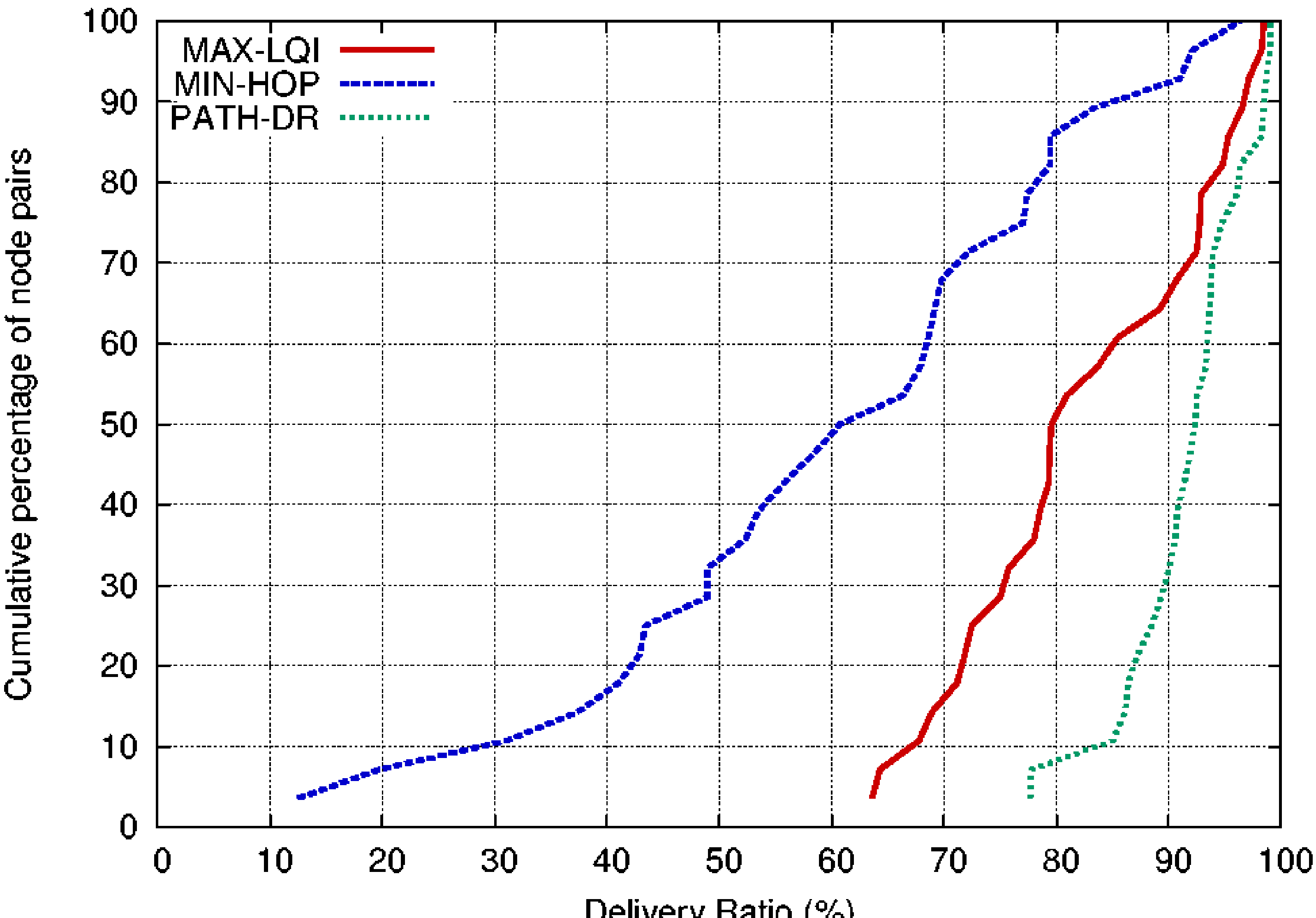
Evaluation Methodology

- Testbed of 30 MicaZ nodes distributed throughout our building
  - Reprogram and debug via web interface at motelab.eecs.harvard.edu
  - Now upgraded to 190 Tmote Sky motes
- Set up certain nodes as “virtual patients” and others as “virtual doctors”
  - Vary parameters such as generated data rate, number of senders and receivers

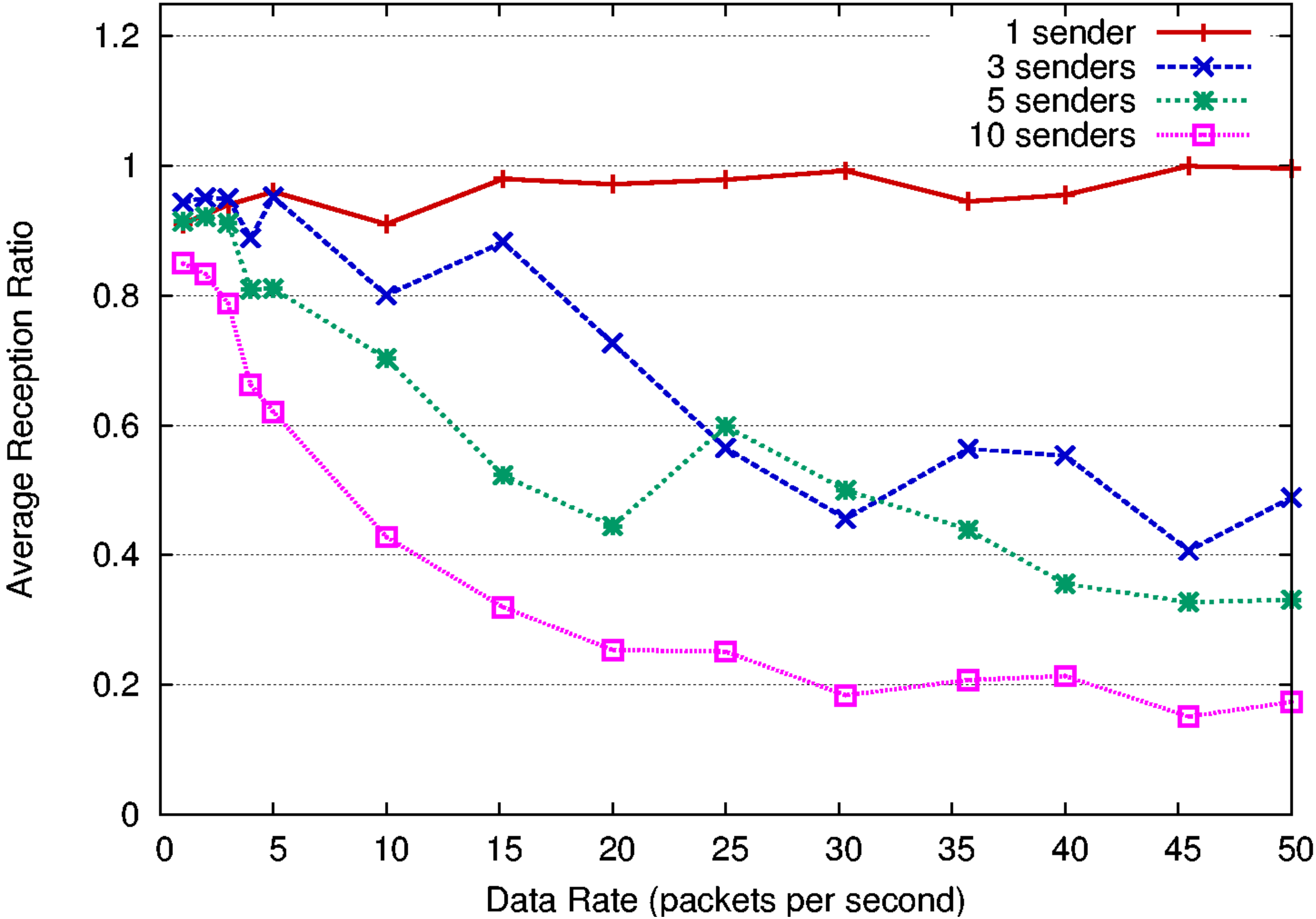


Effect of LQI Based Route Selection

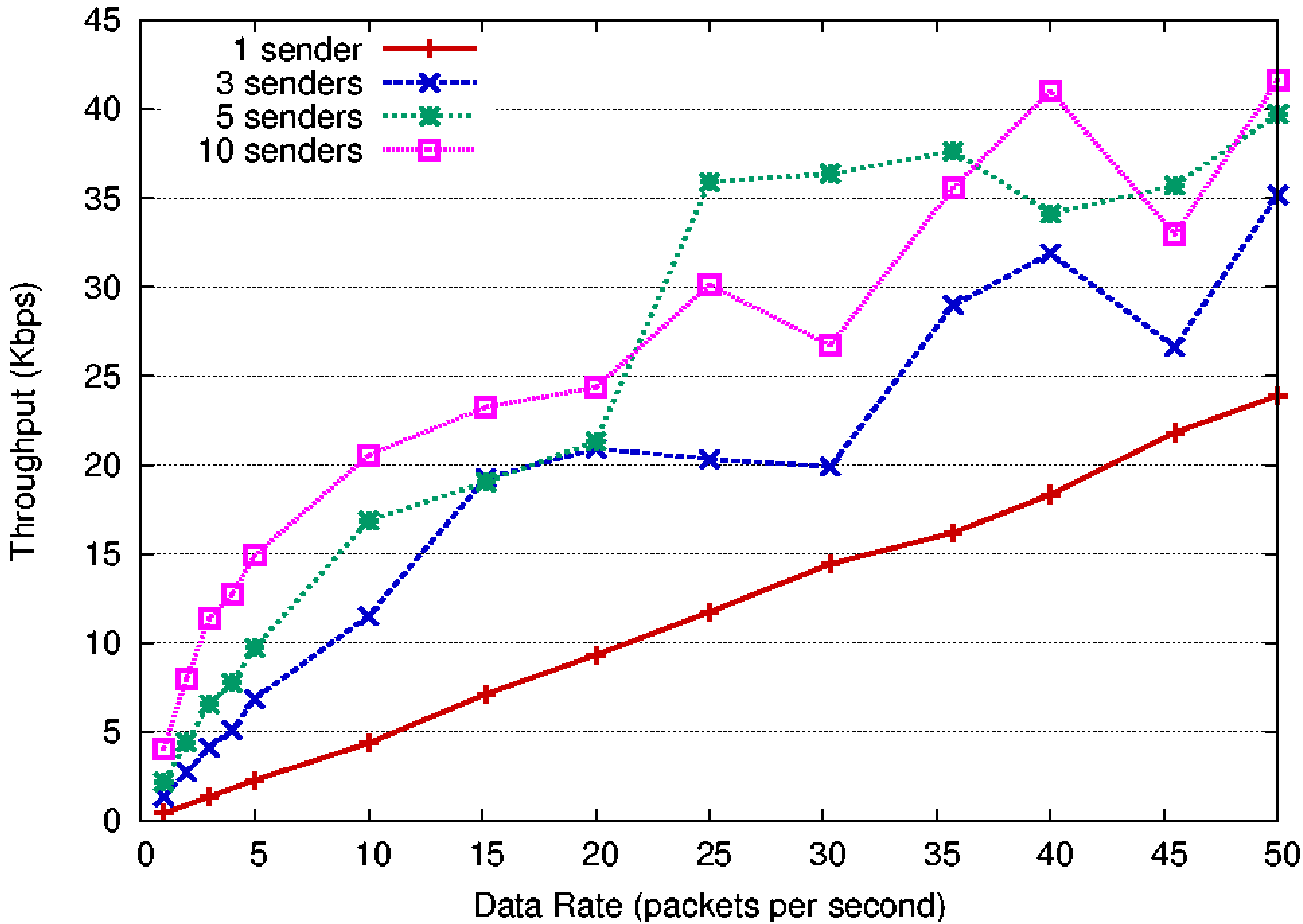
- Comparison to other route selection metrics
  - MIN-HOP: Lowest hopcount path
  - MAX-LQI: Path with worst LQI rating per link
  - PATH-DR: Estimated path delivery ratio from LQI model



Effect of increasing data rate and number of senders



Effect of increasing data rate and number of senders

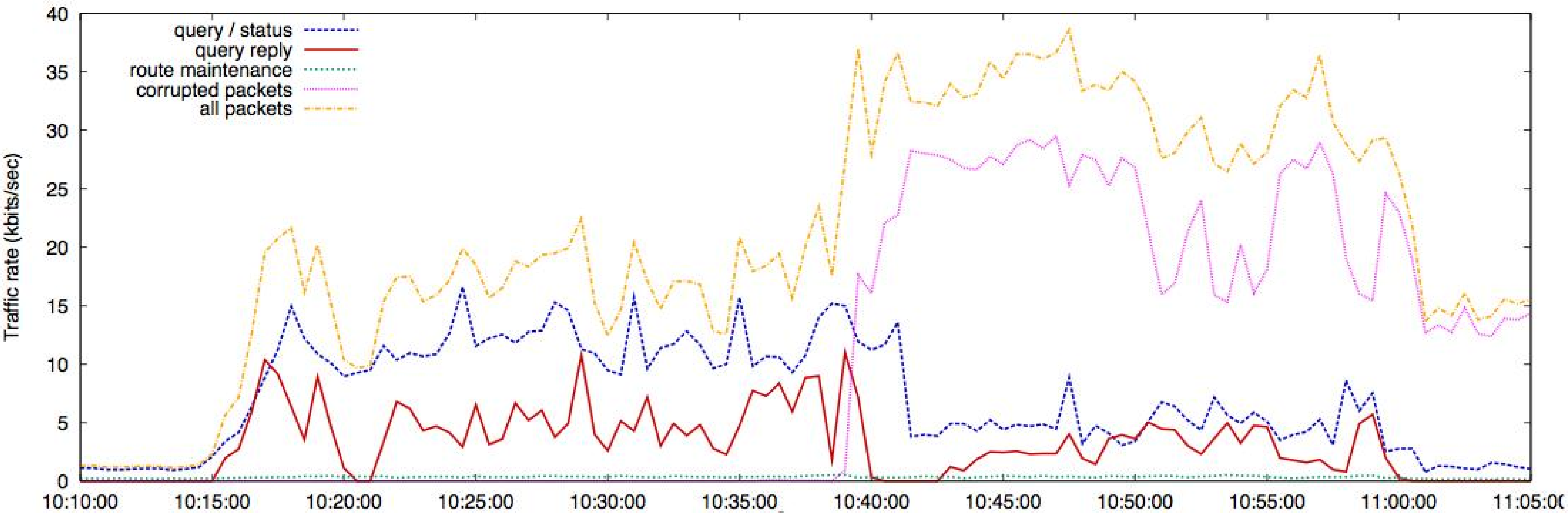


Rethinking multicast

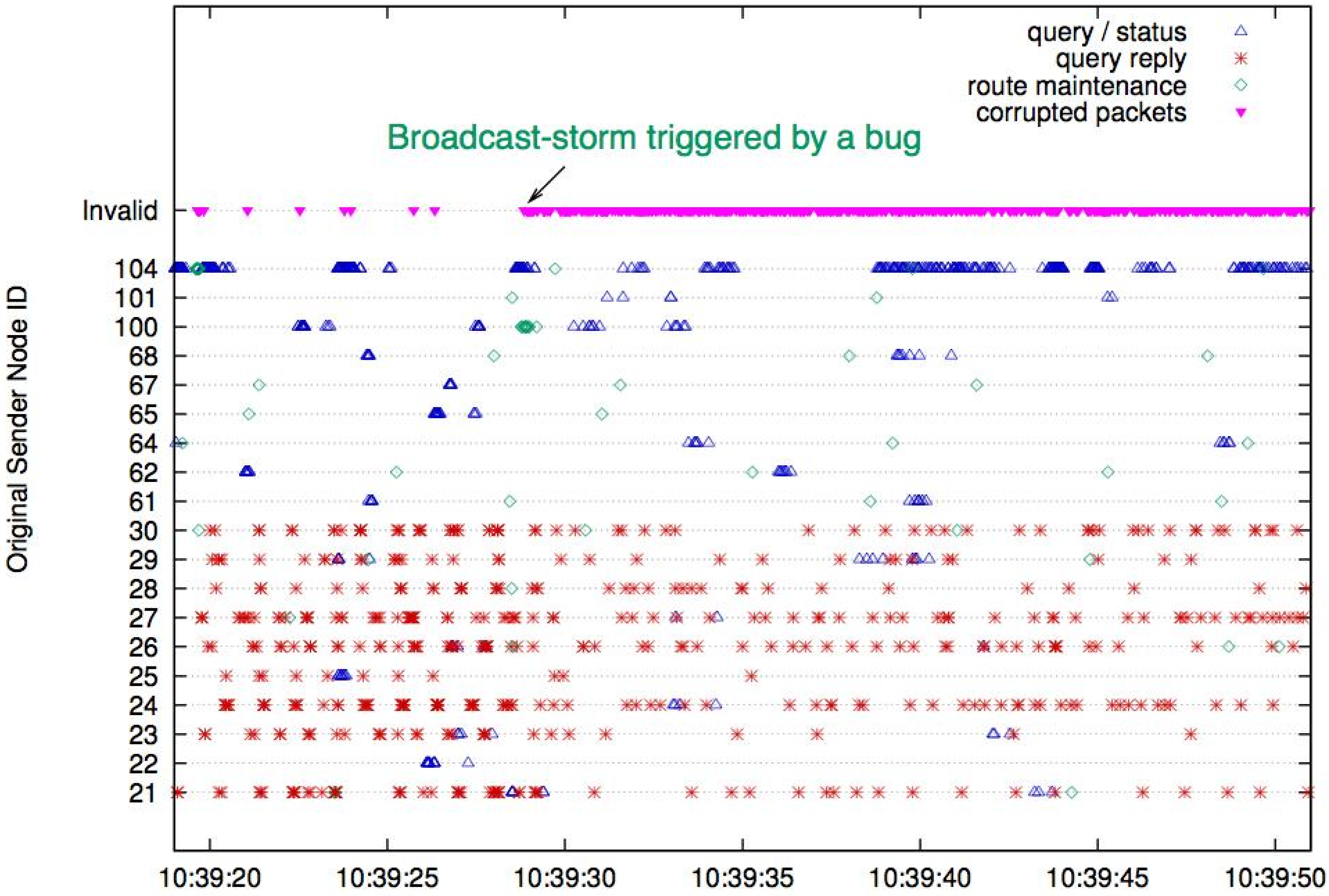
- Multicast routing is fundamentally based on **broadcast** packets
  - Problem: Cannot take advantage of MAC-level ACKs and retransmission
  - 802.15.4 MAC does not support multicast natively
- Original idea: “Best effort” transmission is adequate
  - As long as link to next hop is good – hence the focus on path selection metrics
- Problem: This assumption breaks down under moderate-heavy load
  - TinyOS MAC does not appear to work well with high radio contention
  - Problems with CCA and hidden terminals
- Current direction: Go back to unicast spanning trees
  - Current MAC seems to **require** hop-by-hop ACK and retransmission to ensure good delivery ratios.
  - Form a spanning tree for each receiving device
  - Implies multiple transmissions for multi-destination packets (heavier load)
  - We are currently investigating the break-even point.

Disaster Drill Evaluation

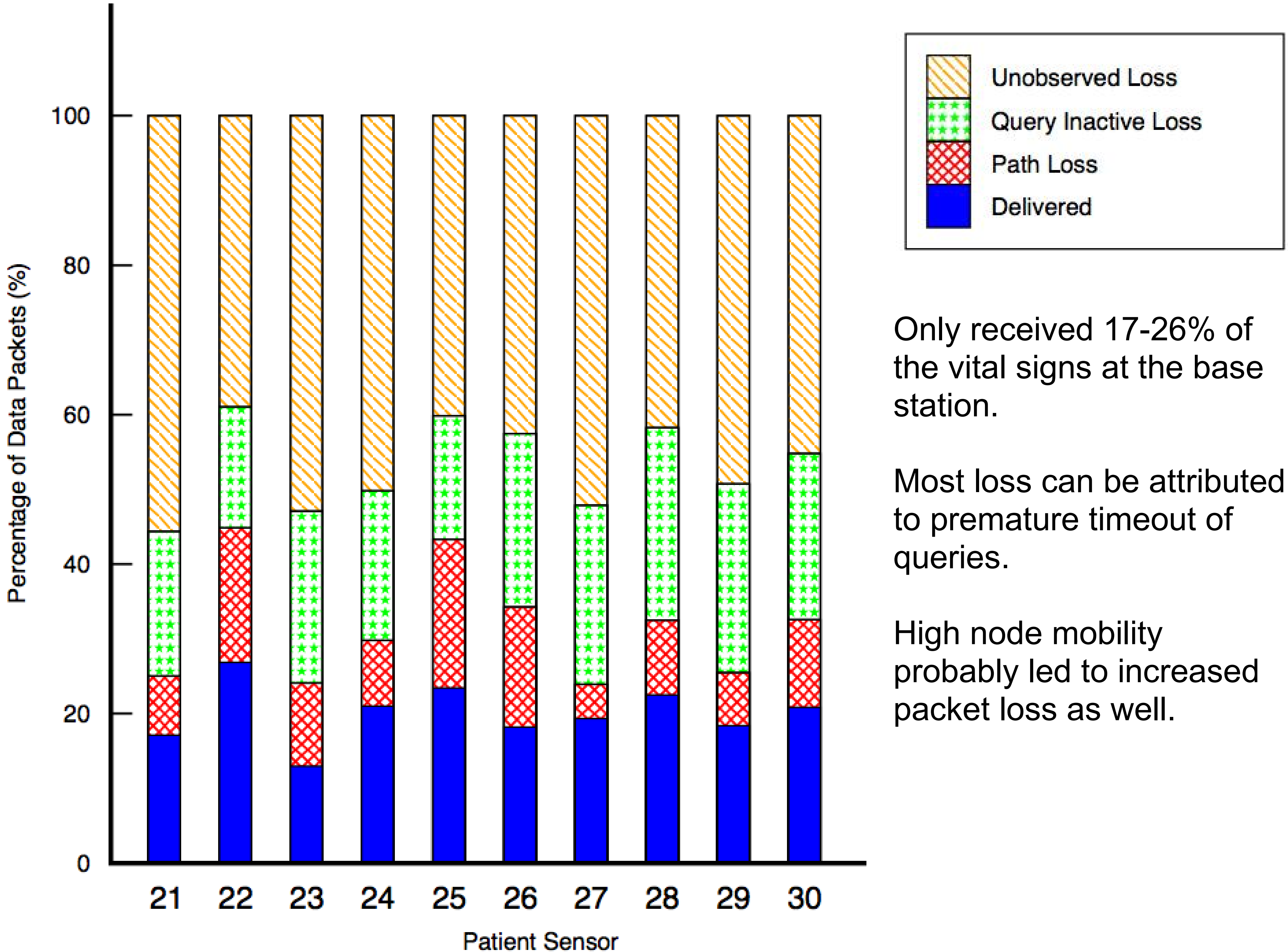
- How well did the system work "in the field?"
  - Deployed six packet sniffers along with the network to record traces of all packets
  - Merged traces to obtain global picture of the network's operation



Disaster Drill: Network flooding bug!



Disaster Drill Query Yield



Only received 17-26% of the vital signs at the base station.

Most loss can be attributed to premature timeout of queries.

High node mobility probably led to increased packet loss as well.