Program Note

- Today, the role of David Culler will be played by an understudy, Scott Shenker

Creating a Sensornet Architecture:

Motivation and Open Questions

David Culler, Scott Shenker, Ion Stoica

(and the entire community….)

Today’s Sensornet Landscape

Appln

EnviroTrack

Hood TinyDB

FTSP Regions

Dir.Diffusion

Transport SPIN

TTDD Deluge Trickle Drip

Routing MMRP TORA Ascent Arrive MintRoute

CGSR

AODVDSR ARA GSR GPSR GRAD

Scheduling DSDV DBF TBRPF

Resynch

SPAN GAF FPS

Topology PC ReORg Yao

SMAC WooMac

PAMAS BMAC

Link WiseMAC TMAC Pico

802.15.4

Bluetooth

Phy eyes

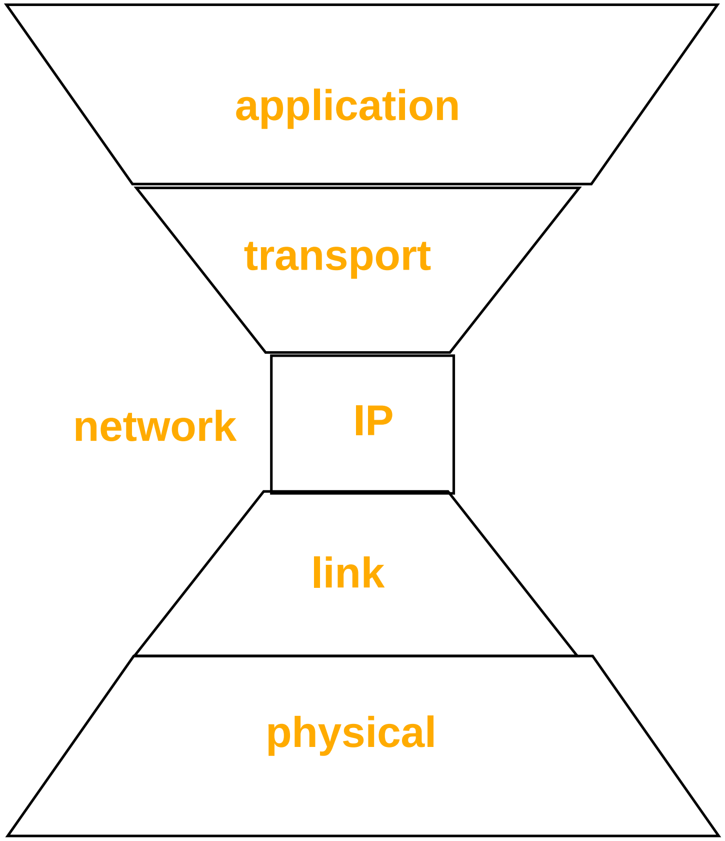
RadioMetrix CC1000 nordic

RFM

Not Just a Messy Picture

- Many components developed in isolation
  - Differing assumptions about overall structure…
- Some vertically integrated systems
  - Not much interoperability between them
- Our conjecture:
  - The biggest impediment to progress is *not* any single technical challenge
  - It is the lack of an overall architecture that would increase composability and interoperability

The “Internet Architecture”



Internet Architecture

- Goal 1: universal connectivity
  - Problem: diversity of network technologies
  - Solution: universal and opaque IP protocol
- Goal 2: application flexibility
  - Problem: application-aware networks limit flexibility *(because network is static)*
  - Solution: end-to-end principle
    - *Put app. functionality in hosts, not network*
    - *Hosts are under our control, and can be changed*

The Internet Architecture

- Shields applications from hardware diversity
- Shields network from application diversity
- Speeds development and deployment of both

How Do Sensornets Differ?

- Apps: data-centric, not host-centric
  - Endpoints not explicitly addressed by apps

  ⇒ Can’t organize around end-to-end abstractions

- Goal: code portability and reuse
  - Not universal connectivity
  - Not application flexibility for static network

  ⇒ End-to-end principle not (automatically) applicable

  In-network processing is often much more efficient

How Do Sensornets Differ (cont’d)?

- Constraints: scarce resources (energy)

- Internet: opaque layers as easy abstraction
  - Willing to tolerate significant efficiency loss

- Sensornets: need *translucent* layers
  - Hide details of hardware underneath
  - But expose abstractions for control

- Goal: trade (small) efficiency loss for (much) less reprogramming

Six Aspects of a Sensor Network Arch.

- Design Principles how to split functionality
- Functional Architecture logical building blocks
- Programming Architecture API/ISA
- Protocol Architecture distributed algortithms, etc.
- System Support Architecture node capabilities
- Physical Architecture hardware

Open Questions

- Only a few of the very many open questions….

Where is the Narrow Waist?

- Internet: best-effort end-to-end packet delivery (IP)

- Sensornets: best-effort single-hop broadcast (SP)?

- Expressive abstraction of a universal link layer
  - Single abstraction for all lower layer technologies
  - Expose mechanisms such as acks, backoffs, FEC,…

- Abstraction should allow higher-layers to optimize without knowing the underlying technology
  - knobs *and* dials

Two Questions about SP

- Can we achieve the necessary efficiency with this common abstraction without having to access the link technology directly?

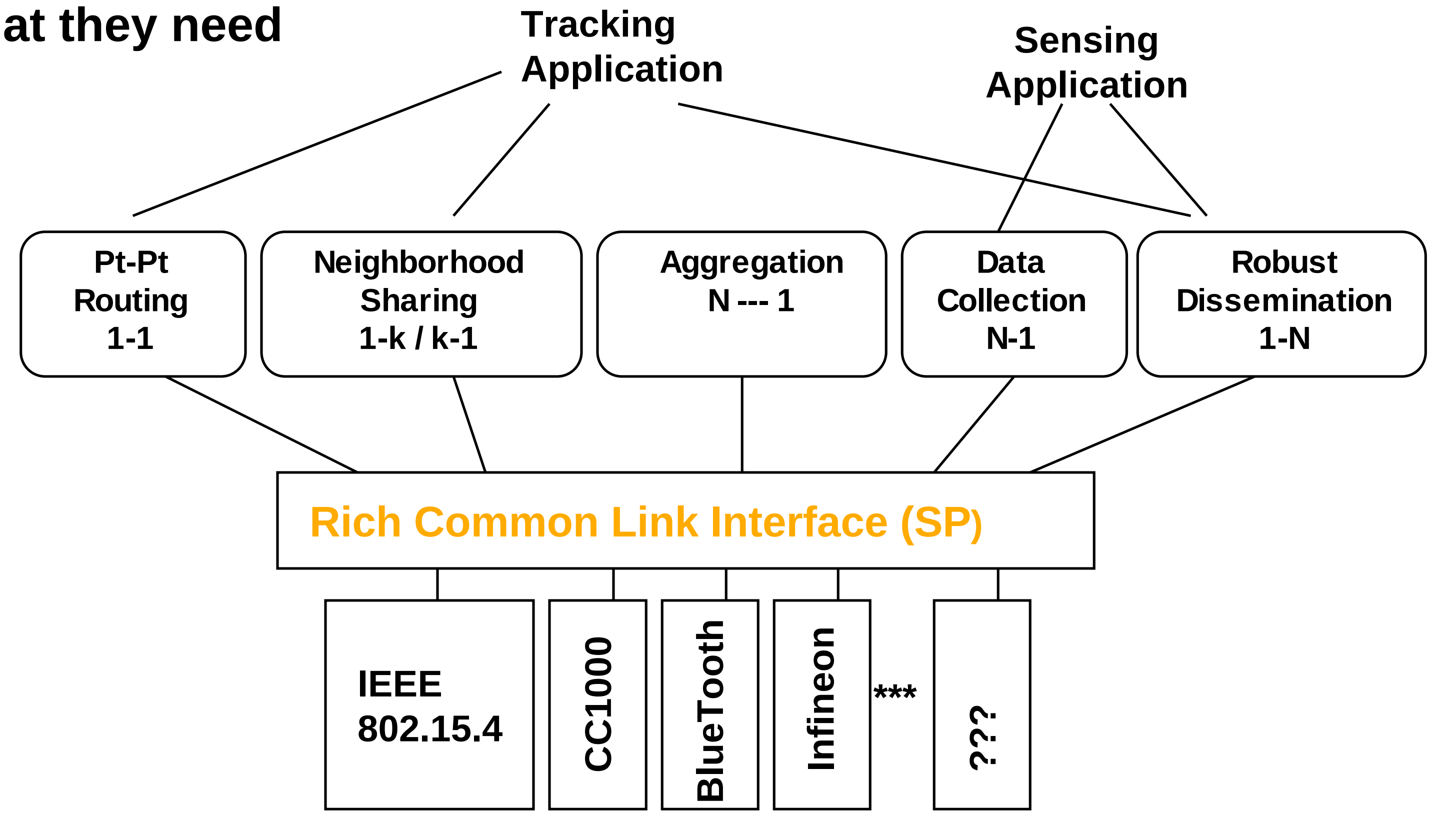
- Where do we want to draw the limits of applicability?
  - Mobility?
  - Actuation?
  - Directed antennae?
  - Cooperative analog communication?

The Sensornet “Hourglass”

Applications
Compose what they need

Multiple
Network
Layer
Protocols

Multiple
Link and
Physical
Layers



Tracking
Application

Sensing
Application

Pt-Pt
Routing
1-1

Neighborhood
Sharing
1-k / k-1

Aggregation
N --- 1

Data
Collection
N-1

Robust
Dissemination
1-N

Rich Common Link Interface (SP)

IEEE
802.15.4

CC1000

BlueTooth

Infineon

***

???

Is There a Transport Layer?

- Internet transport layers provide:
  - Reliable delivery
  - Congestion control

- Their ends-only design is simple and universal

- Sensornets will need these functions too

- With in-network processing and storage, they can’t be done with an ends-only approach

Question about Transport

- Can we achieve the simplicity of the ends-only approach even in the presence of in-network processing and storage?

- Or does each in-network design have to do their own congestion control and/or reliable delivery (if so needed)?

Are Data and Control Different?

- Most current work focuses on data traffic

- Is control traffic qualitatively different?
  - Traffic patterns?
  - Service requirements?

- Do we need an architectural distinction?

Handling Cross-Layer Functions

- Many functions occur at many levels:
  - Discovery, time coordination, power management

- Can one factor them so that these functions are coordinated consistently across “layers”?

Handling Unconstrained Nodes?

- The presence of unconstrained nodes makes system design much easier

- Can we design the architecture so that it can take advantage of, but not count on, unconstrained nodes?

How Might This Effort Fail?

- SP can’t achieve adequate efficiency
- Cross-layer compilers are more efficient
  - Makes programming easy, but compilers are hard
- Unconstrained nodes make everything simple
  - So much easier that people find a way to deploy them
- Rapid technology changes shift basic tradeoffs
- ……..

Height of Arrogance?

- This is a community effort:
  - Annual workshops
  - Informal discussions with various groups

- Push/Pull dynamics
  - Pull in insights and components
  - Push out framework for comments and use

- Not the architecture, just an architecture
  - An experiment in unifying abstractions

Web Site

- http://today.cs.berkeley.edu/SNA/