Wireless Embedded Systems and Networking

Foundations of IP-based Ubiquitous Sensor Networks

Low-Power Wireless Communication

David E. Culler
University of California, Berkeley
Arch Rock Corp.
July 11, 2007

What we mean by "Low Power"

- 2 AA => 1.5 amp hours (~4 watt hours)
- Cell => 1 amp hour (3.5 watt hours)

Cell: 500 -1000 mW => few hours active
WiFi: 300 - 500 mW => several hours
GPS: 50 – 100 mW => couple days

WSN: 50 mW active, 20 uW passive
450 uW => one year
45 uW => ~10 years

$$\text{Ave Power} = f_{act} * P_{act} + f_{sleep} * P_{sleep} + f_{waking} * P_{waking}$$

* System design
* Leakage (~RAM)
* Nobody fools mother nature

Where the energy goes

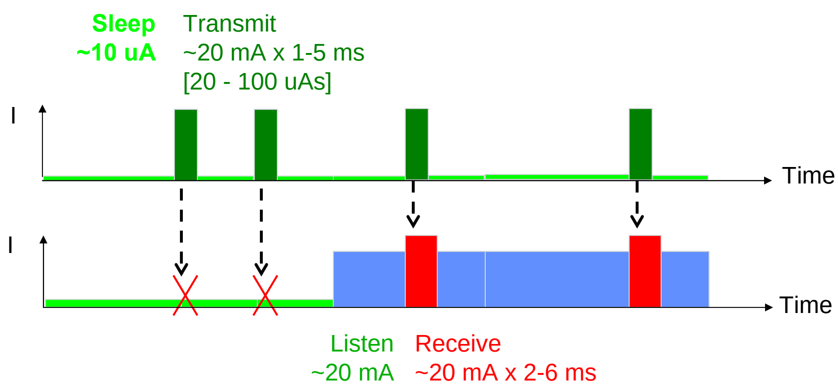
- Sleep
  - 7 uA for TI MSP
- Sensing
- Transmitting results
- Management Traffic
- Routing Structure Maintenance
  - only parent tracking for leaf
- **Listening**
- Forwarding
  - non-leaf
- Overhearing packets destined for others

The "Idle Listening" Problem

- The power consumption of "short range" (i.e., low-power) wireless communications devices is roughly the same whether the radio is transmitting, receiving, or simply ON, "listening" for potential reception
  - includes IEEE 802.15.4, Zwave, Bluetooth, and the many variants
  - WiFi too!
  - Circuit power dominated by core, rather than large amplifiers
- Radio must be ON (listening) in order receive anything.
  - Transmission is infrequent. Reception α Transmit x Density
  - Listening (potentially) happens all the time

⇒Total energy consumption dominated by *idle listening*

Communication Power Consumption



Radio

Type	Narrowband				Wideband		
Vendor	RFM	Chipcon	Chipcon	Nordic	Chipcon	Motorola	Zeevo
Part no.	TR1000	CC1000	CC2400	nRF2401	CC2420	MC13191/92	ZV4002
Max Data rate (kbps)	115.2	76.8	1000	1000	250	250	723.2
RX power (mA)	3.8	9.6	24	18 (25)	19.7	37(42)	65
TX power (mA/dBm)	12 / 1.5	16.5 / 10	19 / 0	13 / 0	17.4 / 0	34(30)/ 0	65 / 0
Powerdown power (μA)	1	1	1.5	0.4	1	1	140
Turn on time (ms)	0.02	2	1.13	3	0.58	20	*
Modulation	OOK/ASK	FSK	FSK,GFSK	GFSK	DSSS-O-QPSK	DSSS-O-QPSK	FHSS-GFSK
Packet detection	no	no	programmable	yes	yes	yes	yes
Address decoding	no	no	no	yes	yes	yes	yes
Encryption support	no	no	no	no	128-bit AES	no	128-bit SC
Error detection	no	no	yes	yes	yes	yes	yes
Error correction	no	no	no	no	yes	yes	yes
Acknowledgments	no	no	no	no	yes	yes	yes
Interface	bit	byte	packet/byte	packet/byte	packet/byte	packet/byte	packet
Buffering (bytes)	no	1	32	16	128	133	yes *
Time-sync	bit	SFD/byte	SFD/packet	packet	SFD	SFD	Bluetooth
Localization	RSSI	RSSI	RSSI	no	RSSI/LQI	RSSI/LQI	RSSI

* Manufacturer's documentation does not include additional information.

- **resilience | performance => slow wake up**
- **Wakeup vs interface level**
- **Ability to optimize vs dedicated support**

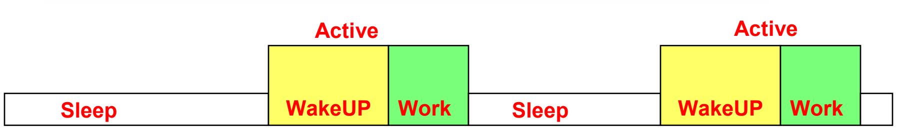
* Polastre, Culler, BMAC, Sensys 2005

Brief Comparison

Vendor	TI	Crossbow	MeshNetics	Ember	DustNetworks
Model	**CC2430**	**IRIS M2110CA**	**ZigBit ZDM-A1281**	**EM250**	**SmartMesh-XT M2030**
Type	**System-on-Chip**	**OEM module**	**OEM module**	**System-on-Chip**	**OEM module**
Link	http://www.ti.com/lit/gpn/cc2430	http://www.xbow.com/Products/productdetails.aspx?sid=264	http://www.meshnetics.com/zigbee-products/	http://www.ember.com/pdf/EM250/120-0082-000H_EM250_Datasheet.pdf	http://www.dustnetworks.com/docs/M2030.pdf
Micro Controller	**8051 core integrated in CC2430**	**Atmega 1281**	**Atmega1281**	**Integrated 16-bt XAP2b MCU**	**Integrated**
Program Memory (KB)	32/64/128	128	128	128	
SRAM (KB)	8	8	8	5	
Nominal Voltage (V)	3	3	3	1.8	3
MCU Active current (mA)	9.5	8	14	8.5	
MCU Active RX current (mA)	26.7	24	19	35.5	22
MCU Active TX current (mA)	26.9 (at 0 dBm)	25 (at 3 dBm)	18 (at 0 dBm)	35.5 (at 0 dBm)	20 (at -2 dBm)
Numer of power saving modes	3	1	1	1	2
Power mode 1 (uA)	190	8 (sleep mode)	6 (power save mode)	1 (deep sleep)	51 (low power networking)
Power mode 2 (uA)	0.5				10 (sleep)
Power mode 3 (uA)	0.3				
Number of timers	4	6	6	3	
Granularity of timers	one general 16-bit timer	four general 16-bit timers	four general 16-bit timers	two 16-bit general timer	
	two general 8-bit timers	two general 8-bit timers	two general 8-bit timers	one 16-bit sleep timer	
	one MAC timer				
ADC					
precision (bits)	12	10	10	12	
# channels	8	8	4	4	
DMA	Yes			Yes	
Radio	**CC2430**	**AT86RF230**	**AT86RF230**	**EM250**	**Integrated**
Data rate (kbps)	250	250	250	250	250
Receiver Sensitivity (dBm)	-92	-101	-101	-97	-90
Current at RX (mA)	17.2	16	16	27	
Current at TX at 0dBm (mA)	17.4 (at 0 dBm)			24.3 (at 0 dBm)	
Current at minimum TX (mA)	18.3 (at -25.2 dBm)	10 (at -17 dBm)	10 (at -17 dBm)	19.5 (at -32 dBm)	
Current at maximum TX (mA)	32.4 (at 0.6 dBm)	17 (at 3 dBm)	17 (at 3 dBm)	27 (at 3 dBm)	
Any link or signal indicator	RSSI, LQI	RSSI, LQI	RSSI, LQI	RSSI, LQI	
Non standard mode					
How high in the stack	802.15.4/ZigBee	802.15.4/XMesh	802.15.4/ZigBee	802.15.4/ZigBee	802.15.4/TSMP

Power States at Node Level

Operation	Telos	Mica2	MicaZ
Minimum Voltage	1.8V	2.7V	2.7V
Module Standby	5.1 μA	19.0 μA	27.0 μA
MCU Idle	54.5 μA	3.2 mA	3.2 mA
MCU Active	1.8 mA	8.0 mA	8.0 mA
MCU + Radio RX	21.8 mA	15.1 mA	23.3 mA
MCU + Radio TX (0dBm)	19.5 mA	25.4 mA	21.0 mA
MCU + Flash Read	4.1 mA	9.4 mA	9.4 mA
MCU + Flash Write	15.1 mA	21.6 mA	21.6 mA
MCU Wakeup	6 μs	180 μs	180 μs
Radio Wakeup	580 μs	1800 μs	860 μs



Telos: Enabling Ultra-Low Power Wireless Research, Polastre, Szewczyk, Culler, *IPSN/SPOTS 2005*

3 Basic Solution Techniques

Goal: listen only when there is likely to be something useful to hear.

- Listen during scheduled time slots
  - Arrange a schedule of possible communication opportunities
  - Maintain appropriately coordinated clocks and schedule
  - Only listen during specific "slots"
  - Many variants: Aloha, Token-Ring, TDMA, Beacons, Bluetooth piconets, … TSMP, …
- Sampled Listening
  - Listen for very short intervals to detect eminent transmissions
  - On detection, listen actively to receive
- Listen after send (with powered infrastructure)
  - Generally, device is not listening and will not receive.
  - After it transmits to a receptive device, it listens for a time window
  - Many variants: 802.11 AMAT, Key fobs, remote modems, …

Approaches

- Powered Router / Duty cycle Leaf
- Coordinator / Beacon
- Network schedule
- Preamble Sampling
- Slotted Preamble Sampling
- Quasi-scheduled

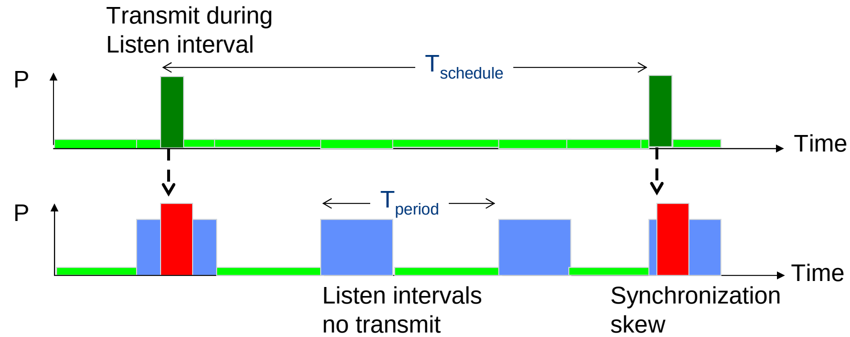
MAC Caution

- The idle listen problem is often associated with Media Access Control (MAC) protocols,
  - TDMA, CSMA, …
- but MACs provide arbitration among multiple transmitters attempting to utilize a shared medium simultaneously.
  - Reduce Contention and associated loss.
  - May involve scheduling (TDMA) or transmission detection (CSMA)
- The problem here is the opposite.
  - Most of the time, nothing is transmitting.
  - Avoid listening when there is nothing to hear.
  - Scheduling and detection are involved, but to determine when to turn on receiver, rather than when to turn off transmission.

Powered Router + Leaf

- Algorithm
  - Leaf nodes never forward, turn off when ever they like
    - » Typically a minimum frequency
  - Forwarding nodes never turn off
  - Special case transmit to leaf
    - » Buffer and wait till hear
- Usage
  - Ember, Sensicast, Millennial, Figure 8
    - » All started with fancier schemes and gave up

Scheduled Listen



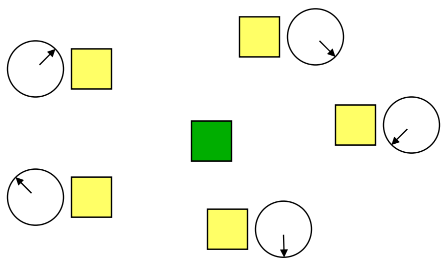
- $P_{ave} \cong P_{sleep} + P_{listen} \bullet T_{listen} / T_{schedule} + P_{xmit} \bullet T_{xmit} / T_{xmit\text{-}interval} + P_{clock\text{-}sync\text{-}ave} + P_{discover\text{-}ave}$
- **Communication to maintain synch. sets lower bound.**
- **Full power listen to discover and join schedule.**

Schedule Mechanisms

- **Compute schedule off-line and distribute it to the nodes**
  - Requires some unscheduled communication mechanism to perform survey of who-communicates-with-whom and who-interferes-with-whom, collect results, and distributed schedule.
  - Easy for star and clique.
  - Changing conditions, additions and deletions are problematic
- **Define set of slots, advertise, resolve**
  - Relatively easy for star, clique, or ring.
  - Complicated for general mesh. Essentially constraint resolution.
  - Typically, coordinator schedules for one-hop neighbors and coordinators (cluster heads) stay powered.
    - » As network grows, leaf-fanout trends to 1
  - Embedded a “scheduling tree”
- **Essentially the traffic-light coordination problem, but the roads and flows are changing.**

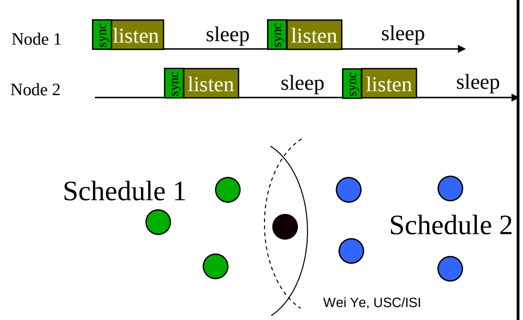
Communication Scheduling: TDMA

- Time Division Media Access
  - Each node has a schedule of awake times
  - Typically used in star around coordinator
    - » Bluetooth, ZIGBEE
    - » Coordinator hands out slots
  - Far more difficult with multihop (mesh) networks
  - Further complicated by network dynamics
    - » Noise, overhearing, interference



S-MAC

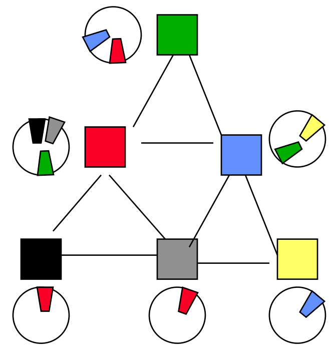
- Carrier Sense Media Access
- Synchronized protocol with periodic listen periods
- Integrates higher layer functionality into link protocol
  - Hard to maintain set of schedules
- T-MAC
  [van Dam and Langendoen, Sensys 2003]
  - Reduces power consumption by returning to sleep if no traffic is detected at the beginning of a listen period



Ye, Heidemann, and Estrin, INFOCOM 2002

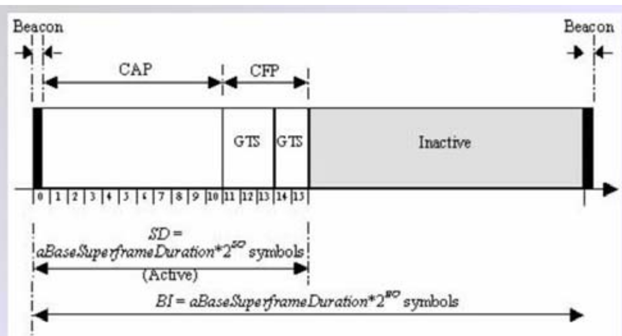
Flexible Power Scheduling

- TDMA-like scheduling of listening slots
  - Distributed allocation
  - CSMA with LPL within a slot
- Node allocates
  - listen slots for each child
  - Transmission slots to parent
  - Hailing slot to hear joins
- To join listen for full cycle
  - Pick parent and announce self
  - Get transmission slot
- CSMA to manage media
  - Allows slot sharing
  - Little contention
- Reduces loss & overhearing
- Connectivity changes cause mgmt traffic



Holte, PhD 2004, IPSN 2004

IEEE 802.15.4 SuperFrame



Beacon Interval (BI) can be a multiple of the Superframe Duration (SD)

Figure 59—An example of the superframe structure

Source: IEEE 802.15.4 specification

Coordinator / Beacon

- Algorithm
  - Coordinator broadcasts periodic beacon assigning speaking/receiving slot for each node
  - Node listens for full beacon period to learn slot
  - Listens / Recv / Transmit in each slot to maintain synch and communicate
- Usage
  - 802.15.4 provision, baked deeply into Zigbee
    - » No Zigbee provider uses it, just powered routers + leaf
- Strengths
  - Localized timesynch, not whole network
- Weaknesses
  - Fragile: can't comm till find coordinator and bind. What happens if none around. Lose coordinator temporarily.
  - Adaptation
    - » Scan when joining network.
    - » When to reclaim slots
  - Utilization
    - » Only fraction of channel available, even when only one node needs to burst
  - Efficiency
    - » Need to participate in beacon coordination even if nothing to send or receive
  - Routing
    - » Within cell easy, need to coordinate across multiple coordinators
- Power bounds
  - Beacons + time synch

TSMP

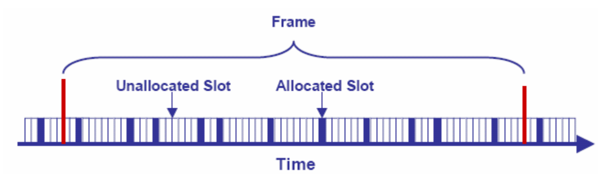


Figure 3. TSMP Slots and Frames

- Connectivity Survey of entire network on every channel.
- Offline computation of a communication opportunity schedule.

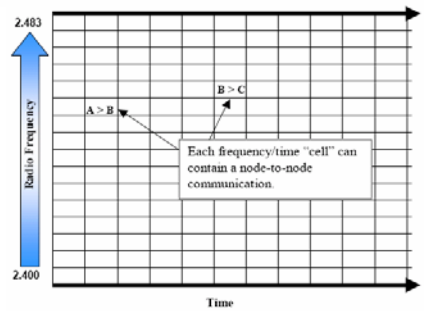


Figure 6. Frequency/Time Matrix

Scheduled Listen Tradeoffs

+ Transmission scheduling that reduces contention, reduces loss, reduces retransmission, saves energy.
  - regardless of whether reception is scheduled.

+ Transmission and Reception costs predictable in steady state (synched and stable)

-- Listening Cost: $P_{listen} \bullet T_{listen} / T_{period}$
  - Pay the cost even when nothing is received.
  - T_{listen} must be greater than the maximum synchronization variation.

- Delay
  - In the worst case, delay is T_{cycle} per hop. Expected $T_{cycle}/2$. Aligning slots for one direction of flow increases delay for others.
  - Jitter in latency is small,
    - » except that loss requires retransmission and whenever fanin exceeds reserved slots for next hop

\- Bandwidth
  - Limited as $1/T_{schedule}$ even if rest is unused

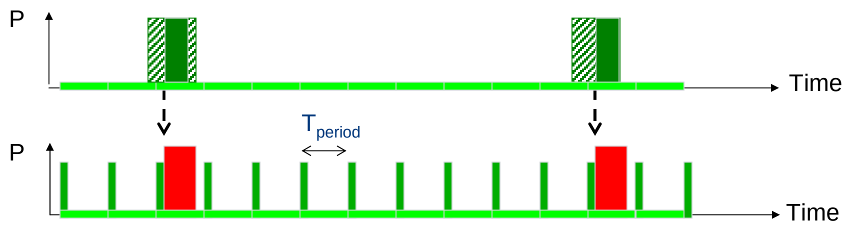
Scheduled Listen (Cont)

-- Time synchronization required
  - Frequency of time-synch protocol communication increases with required accuracy and with variations in clocks.
  - If you lose synch, it is hard to regain it because you don't know when the node is listening. You must transmit continuously or wait for it to go into a watchdog full listen.
  - Reducing listening cost (T_{listen}) requires tightening synch accuracy.
  - Having paid the cost, timestamping data is easy, but generally overkill.
  - Nodes cannot "go silent" as they will lose synch.

-- Discovery/Join cost
  - generally requires long period of full listen or long period of announce transmission.
  - The less the nodes are listening, the more costly the discovery.

- Fragile (with scale)
- Rigid (with scale and with variation)

Sampled Listen



- Relatively frequent, Very short “Listen Samples”
- Transmit (infrequent) increased to cover T_{period}
- $P_{ave} \cong P_{sleep} + P_{listen} \bullet T_{sample} / T_{period} + P_{xmit} \bullet T_{period} / T_{xmit\text{-}interval}$
- No time synchronization required.
- Rapid join and at low power.
- Scheduled sleep, wake times, time synch, etc. built on top for further reductions in power

Sampling Mechanisms

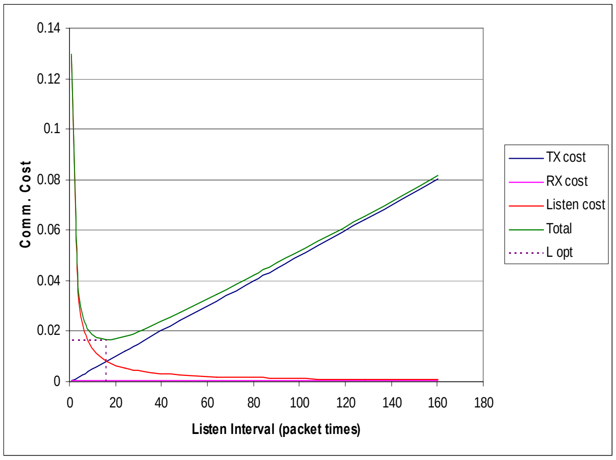
Typical Frame Structure

Preamble	Start	Header	Data Payload	Trailer

Goal: Shift continuous listen burden to infrequent transmit

- **Preamble Sampling (DARPA 1984) and Low-power Listen (UCB, 99)**
  - TX: extend preamble to listen period.
  - RX: samples, hears preamble, Receives till START as usual.
  - Requires low level link access (e.g., RFM1000)
- **Repeated Transmit**
  - Transmit packet multiple times to fill listen period.
  - RX: CCA to sample, ON to receive packet
- **Wake-up Packets**
  - Transmit (small) wake up packets to cover sample, then packet

Optimal Listen Sampling

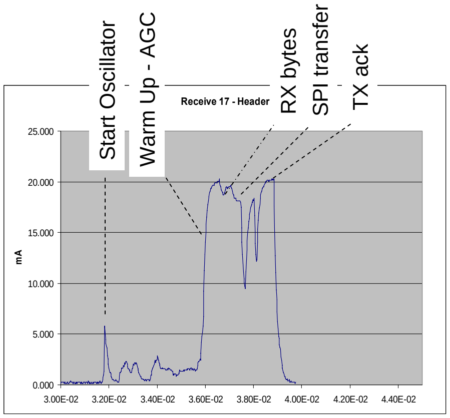


Sampled Listen Trade-offs

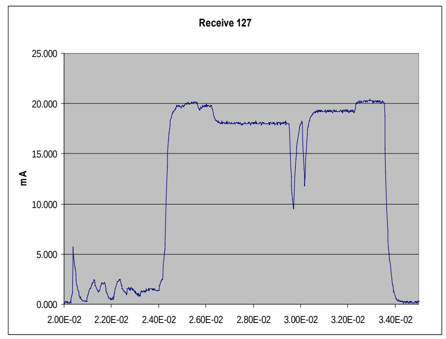
+ Responsive
  – Unscheduled communication at T_{period} per hop, not $T_{schedule}$
+ Adaptive
  – Joins, leaves, changes in connectivity don't require rescheduling
  – Rapid discovery
+ No synchronization cost
  – Can run time synch on top to further reduce costs
  – Don't depend on it for correctness
+ Resilient
  – When forced to listen, do so at low power
+ Flexible
  – T_{period} configurable to workload, even on per-node basis
  – Permits a wide array of further optimizations to eliminate reduce wakeup costs and listening samples
- Bandwidth
  + Node can burst at full link bandwidth, not $1/T_{schedule}$
  - unoptimized bandwidth limited by $1/ T_{period}$

- Transmit cost increases T_{period}
  – So does receive cost if you do it naively

Bottom Up Power Modeling



Receive packet with 15.4 header (17B)
- one EUID64, one Short16



Receive max 15.4 packet (127B)

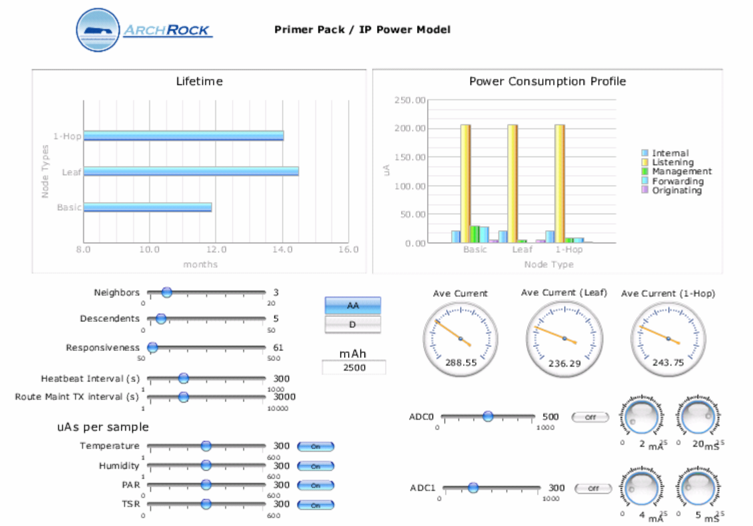
Linear model:

TelosB configuration:
TI MSP430-F1611 + TI CC2420

Communication Power Model

- **Sleep Current = I_s uA**
- **RX packet (n) = R_0 + n* R_{byte} uAs**
- **RX wake and receive = R_w + RXpacket(n)**
- **RX overhear and reject: R_{rjt}**
- **TX packet (n,c) = T_0 + n* T_{byte} + W* I_w uAs**
  - **Radio Wakeup, Clear Channel Assessment, Transmit**
- **Listen = L_w / W uA**

Example – Whole System Modeling



Power-aware Routing

- Cost-based Routing
  - Minimize number of hops
  - Minimize loss rate along the path
  - Perform local retransmissions, minimize number along path
- Energy balance
  - Cost-function prefers nodes with larger energy resources
- Utilize redundancy
  - Nodes near the sink route more traffic, hence use more energy
  - Give them bigger batteries or provide more of them and spread the load
  - Randomize routes
- Utilize heterogeneity
  - Route through nodes with abundant power sources

Example Mesh Routing Implementations

- TinyOS
  - MintRoute (2003) – cost based collection routing over Low Power Listening on CC1000
  - MultiHop LQI (2005) – LQI cost-based collection on 15.4
  - Drain / Drip – (2006)
- ZIGBEE
  - Star, Tree, and Mesh defined
  - Tree generally rejects
  - Only mesh profiles appear to be utilized
  - AODV (build tree from any node with full bcast)
  - No power management defined
- Ember
  - Proprietary extensions / simplifications of zigbee
- Millennial
  - Powered backbone
- Crossbow xmesh / TrueMesh
  - extends TinyOS mint-route with local slots
  - Preamble Chirps on slot edge, listens only on edges
- Dust SmartMesh
  - Survey all pairs connectivity
  - Compute off-line schedule of Source->Dest opportunities
  - Maintain global time slots with single pair per slot
  - Periodic collection to single root at 1 packet per minute
- Arch Rock
  - 15.4 packet-based Sampled Listening
  - IP routing over 6LoWPAN, triply-redundant mesh

In-network Processing

- Best way to reduce communication cost is to not communicate
- Compute at the sensor
  - Only communicate important events
- Compute over localized regions of the network
  - Distributed detection
  - Validation