

# Architecting Energy Efficient Computing Platforms

Rajesh Gupta, UC San Diego

<http://mesl.ucsd.edu>

**Science of Power Management, April 9, 2009**

# Credits: Projects and Teams

- Completed Efforts
  - Power Aware Distributed Systems (PADS)
    - Mani Srivastava, UCLA
    - Cristiano Pereira, Arun Kejariwal.
  - Formal Methods in Power Management
    - Sandy Irani, UC Irvine
    - Sandeep Shukla, Virginia Tech
    - Ravindra Jejurikar, Dinesh Ramanathan, Zhen Ma
- Ongoing
  - System level Power Management
    - Yuvraj Agrawal, Zhong Yi Jin
  - Location, Location infrastructure, Software for Real World
    - SPATIAL PROGRAMMING
    - Thomas Weng, Ryo Sugihara, R. K. Shyamasundar, TIFR, India
    - DYNAMIC RESOURCE DISCOVERY
    - Kaisen Lin, Jeffrey Namkung , Chalermek Intanagonwiwat, Amin Vahdat
  - GreenLight: Coherent Coprocessing for Energy Efficient Computing
    - Joel Coburn, Arup De, Gerald Clark, M. Florea, ....Tom DeFanti
- Launching: Non-Volatile Data Intensive Supercomputing NV-DISC

---

  - Arup De, Steve Swanson

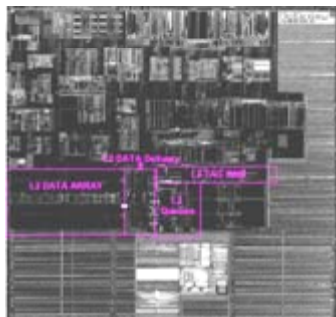
---

# Outline

- Energy and Computing
  - Three Observations
  - Approach and Lessons Learnt
    - Architectural Design for Low Power
    - Algorithm Design for Power Management
  - Cross-layer optimization and awareness
    - For aggressive duty-cycling
  - Takeaways
-

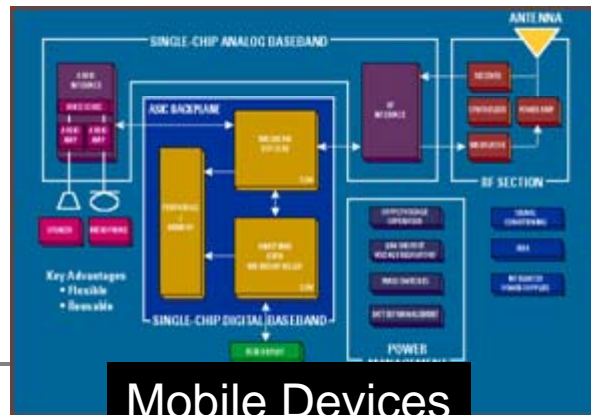
# Energy Efficiency is at the front & center of all forms of computing

- Current architectural offerings range from  $300\mu\text{W}$  to  $30\text{mW}$  per (reasonable) MIPS.



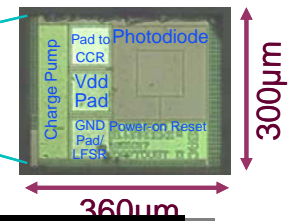
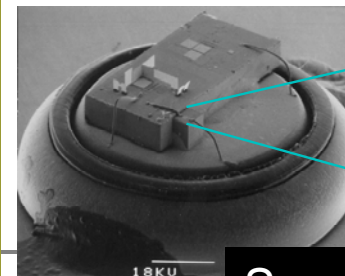
Stationary Devices

mW



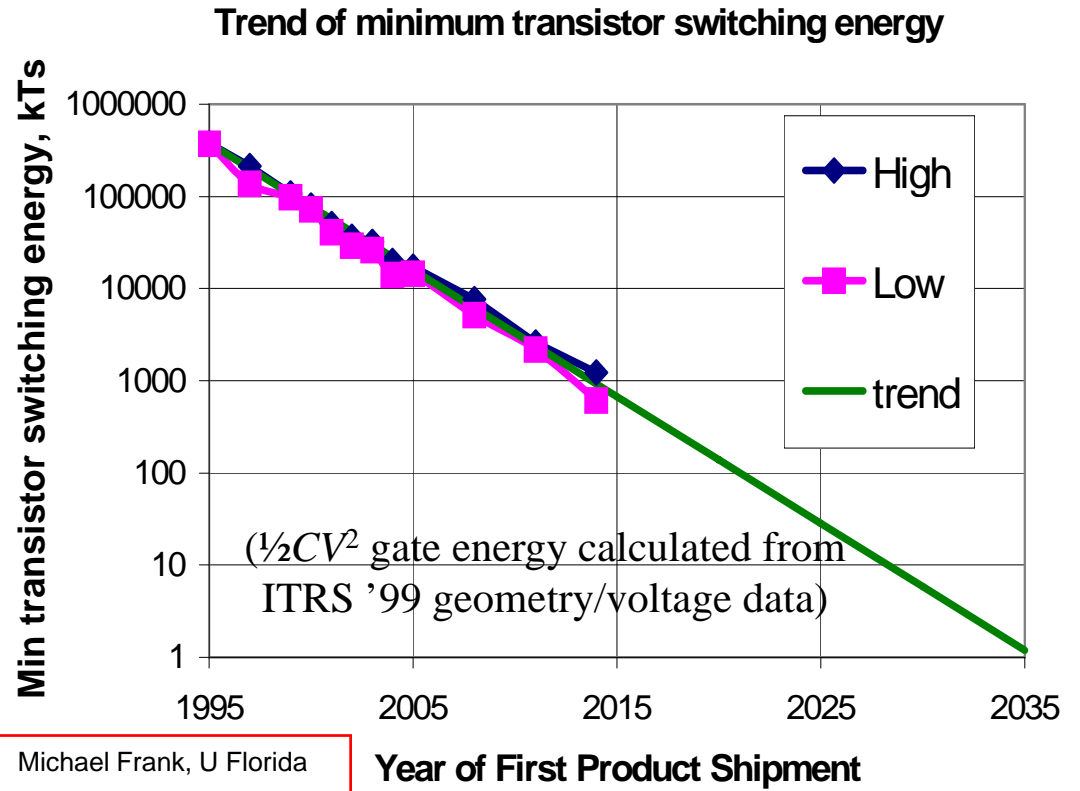
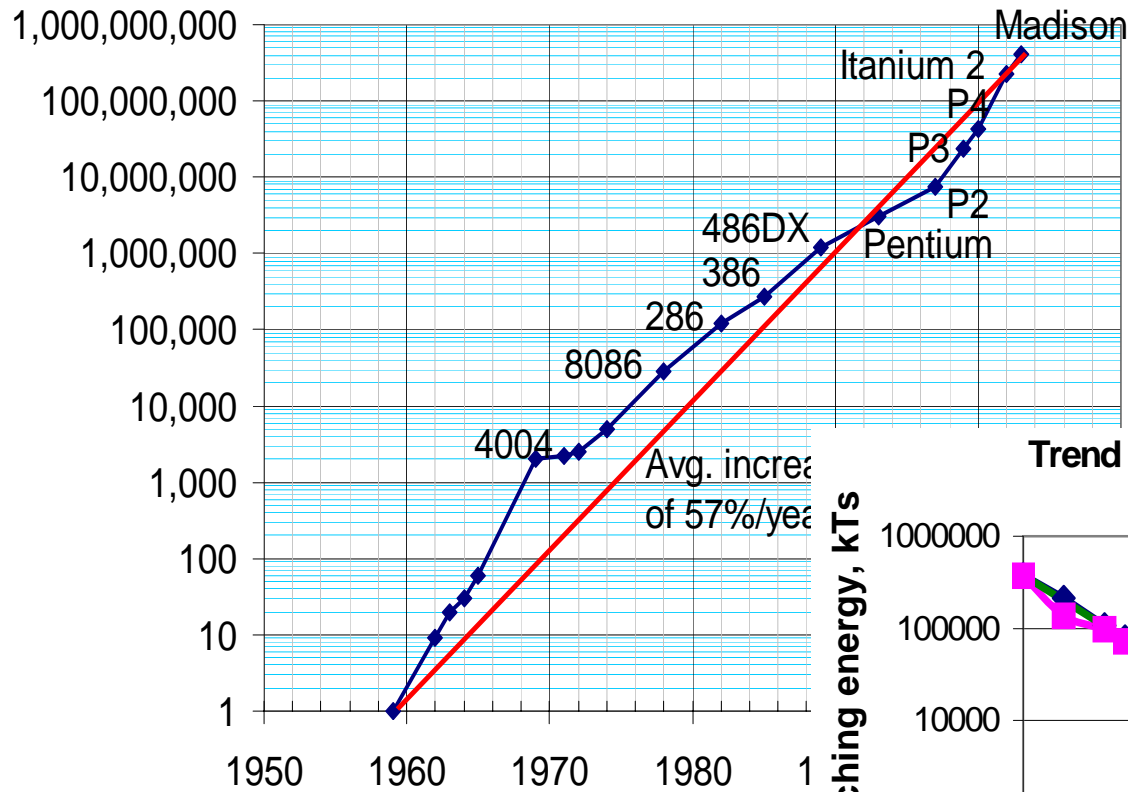
Mobile Devices

$\mu\text{W}$



Sensor Devices

# Our Famous Scaling Curves



Michael Frank, U Florida

Year of First Product Shipment

# Physicists and Computer Scientists Have Been Here Before

- Confirmed physical theories define limits
  - Relativity: speed of light: latencies, bandwidth
  - Quantum: uncertainty: information capacity
  - Quantum: energy, reversibility: processing rate, energy/op
- Newton, Einstein:
  - Energy and mass are the same thing in different units
  - Energy, matter can not exceed SOL. If you do, there exists a FOR in which causality is violated
- Thermodynamics relates heat, temperature and work
  - Entropy = heat/temperature = log (#states)
- Feynman, von Neuman, Shannon, Landauer
  - Entropy = amount of unknown or incompressible information in a physical system
  - Information loss equates heat generation
  - Minimum energy per op same as min energy per bit
  - Energy lost to heat,  $S.T = kT \ln 2$  per bit loss, 18eV at 300K

---

**Minimum Vdd of 48mV (with 30mV swing) verified by several groups.  
Realistically approaching 200mW.**

---

# Our Work: Know or Find Limits, Architectural Design to Reach Limits

- Hardware:

- What is the right choice and combinations of components?  
Processors, Radios, Storage, Networking. [Mobisys 07-08, NSDI 09]

- Power System States and Transitions

- What is the right choice of power states and methods to move among these? Dynamic power management, Speed Scaling.  
[TCAS-I 09, TOA 07, TCOMP 06, TCAD 06]

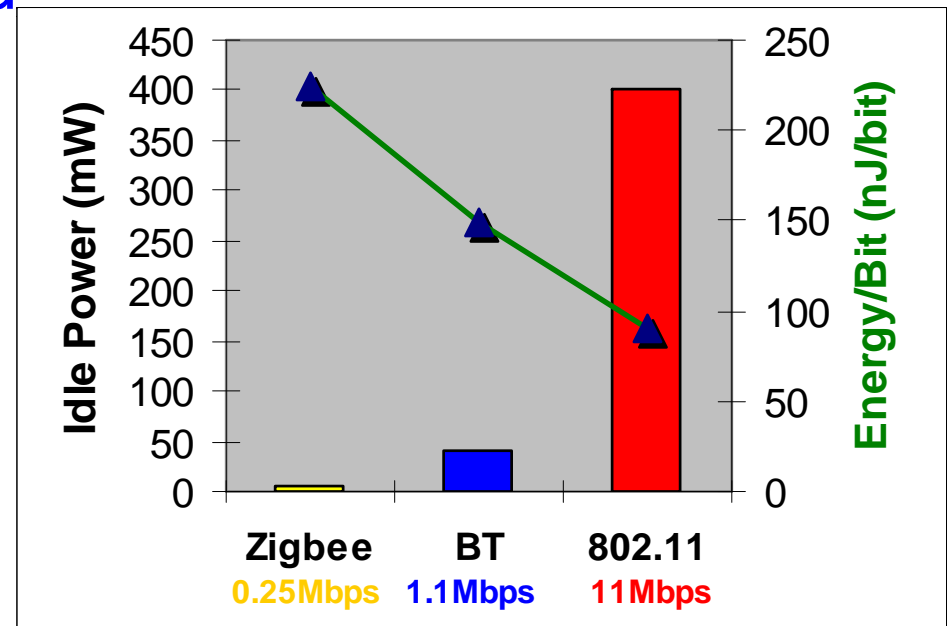
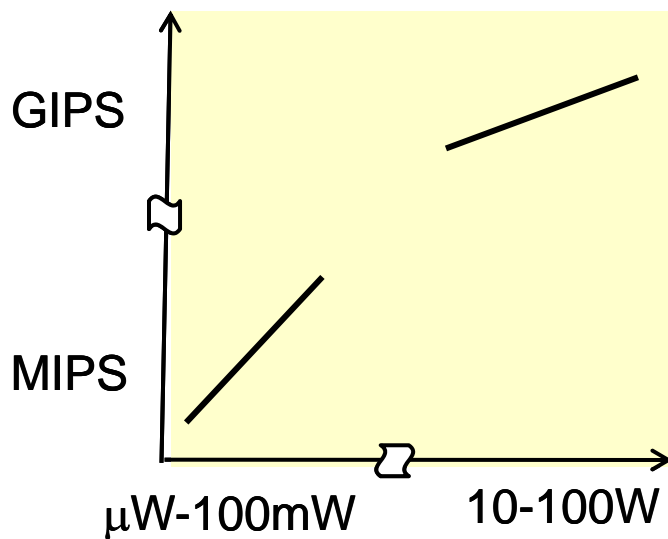
- Software

- How to manage power-related decisions across abstraction layers (more in software than hardware)? Metadata methods, reflection, introspection. [TVLSI 06, IPDPS 05]
-

# Three Important Observations

## O1. Hardware is increasingly heterogeneous

- Component efficiency rated against absolute performance delivered



Medium range, High power (400mW-1W), Higher bit-rate (54Mbps)



Short range, low power (20mW-100mW), lower bit rate (2Mbps)



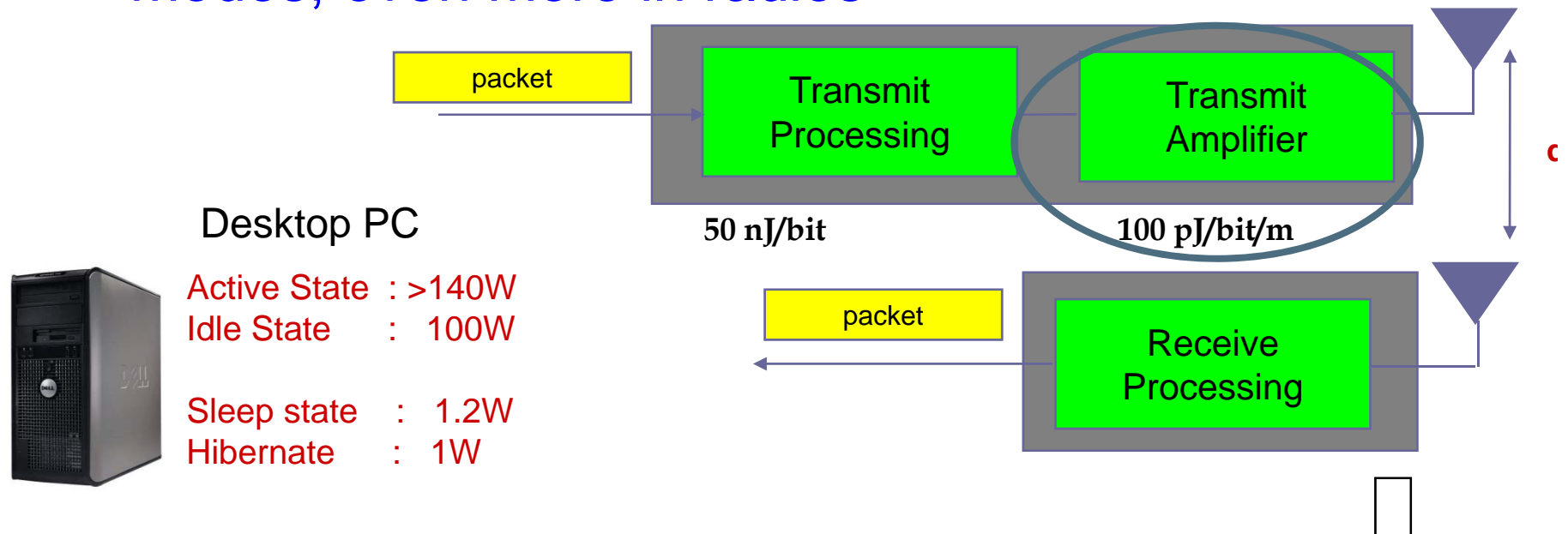
Long Range, very low power (<10mW), voice only



# Three Important Observations

## O2. Tremendous dynamic variation in power use

- 6-10x variation in power from active to sleep modes, even more in radios



## O3. Abstraction stack has a real (high) cost for energy.

---

# Improving Energy Efficiency: Three Approaches

Reduce distance (O1)

- Physical, logical

Minimize wasted work (O2)

- Shutdown, slowdown, procrastinate

Specialized *heterogeneous* processing (O3)

- In a generalized execution environment

---

**Apply these lessons to build better architectures, power management algorithms.**

---

# Introduce & Exploit Heterogeneity

- Exploit the wide range of power consumption
  - Duty-cycle higher power consumers
  - ...in lieu of low power alternatives when possible
- To do this well, three things must happen
  - Subsystems must be “functionally similar”
    - Radios – fundamentally send bits across the air
  - Subsystems must be “heterogeneous”
    - Operate in different power performance regimes
  - Subsystems must “collaborate”

---

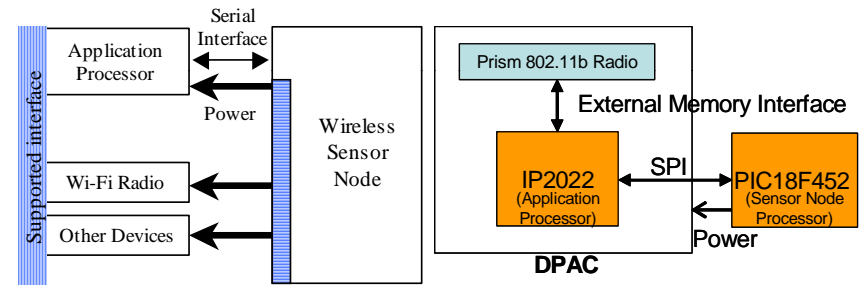
**Solves the Receiver Side Problem (RSP)**

# Architectural Collaboration

- Duty cycle the more power consuming resource using the other

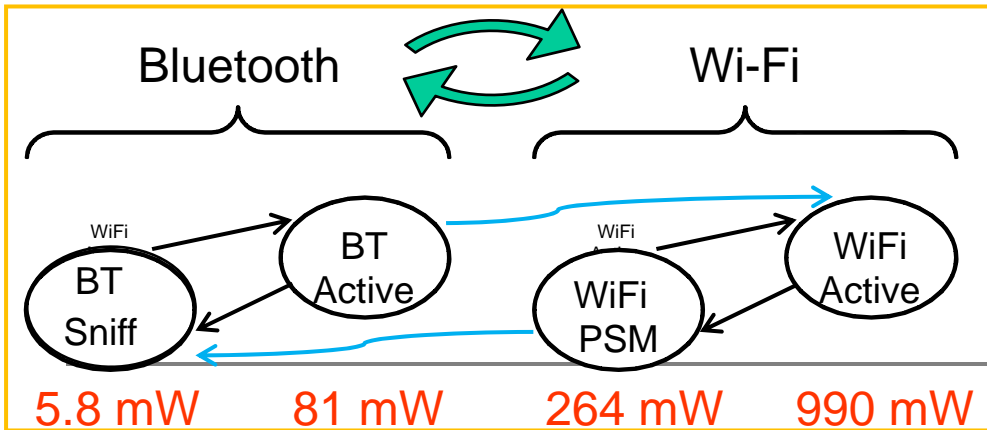


**Sleep-talking Processors**



WGN Block Diagram

WGN Architecture

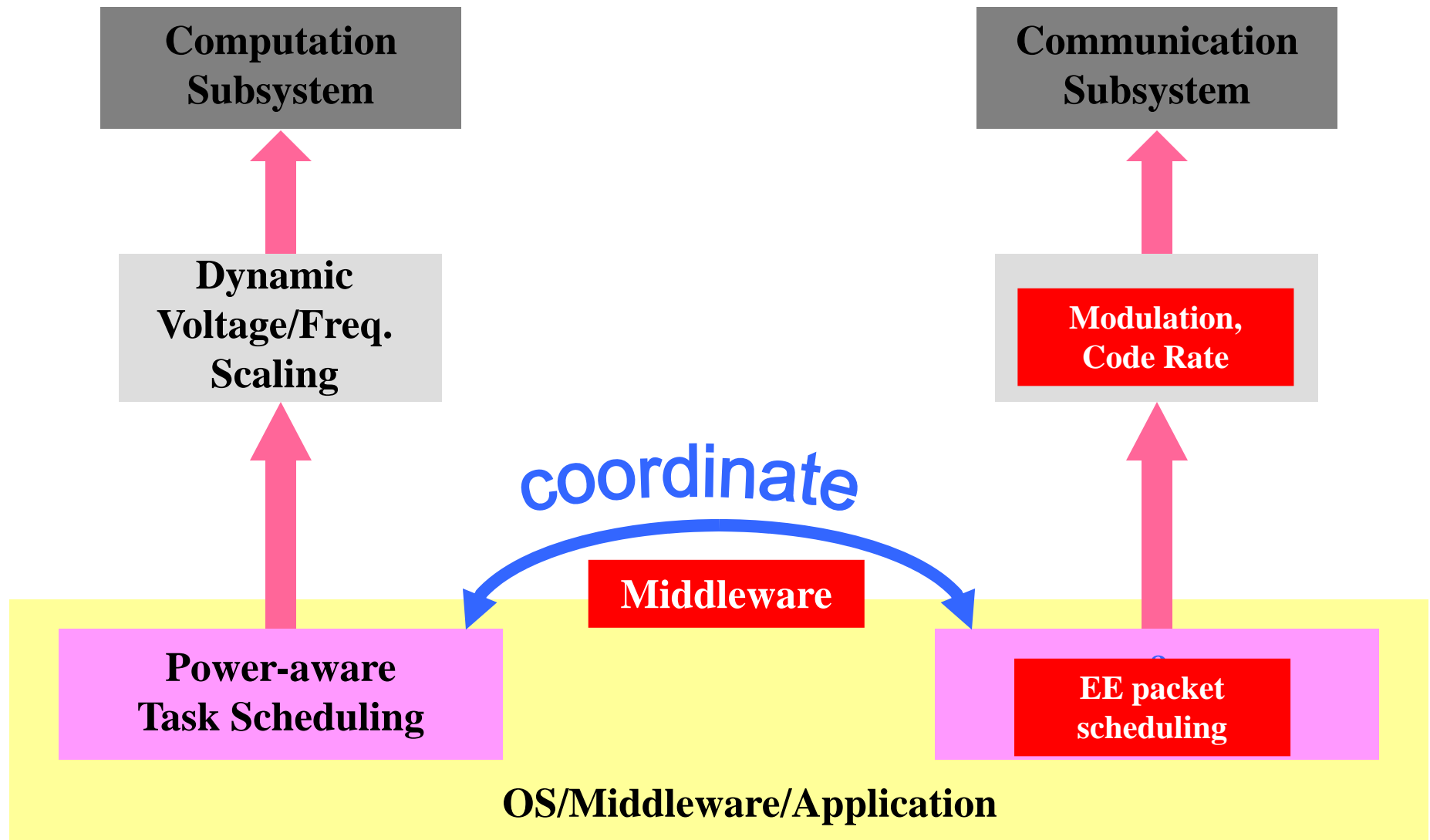


1. Use a low power radio to wake up higher power radio
2. Build a radio-switching hierarchy

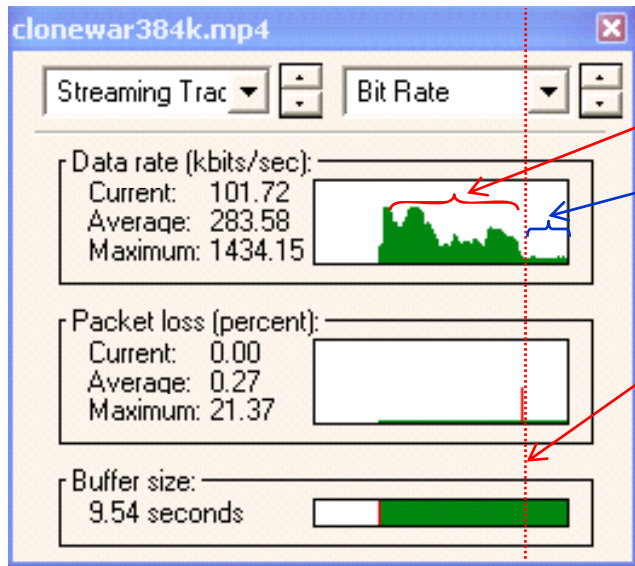
**Effectively expand the power states at a system level**

**E.g. consider a system with Bluetooth and Wi-Fi radios**

# Collaborate and Coordinate

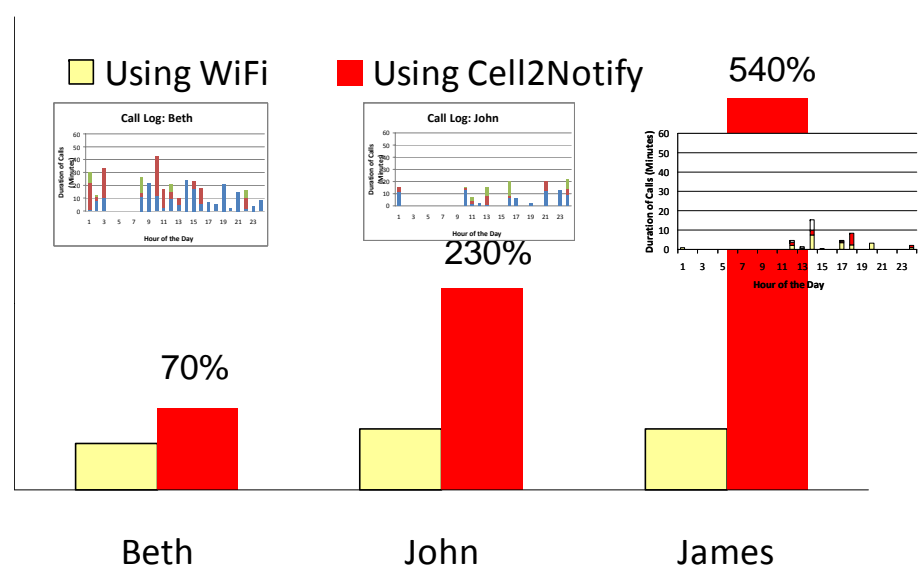


# Collaborating Radios

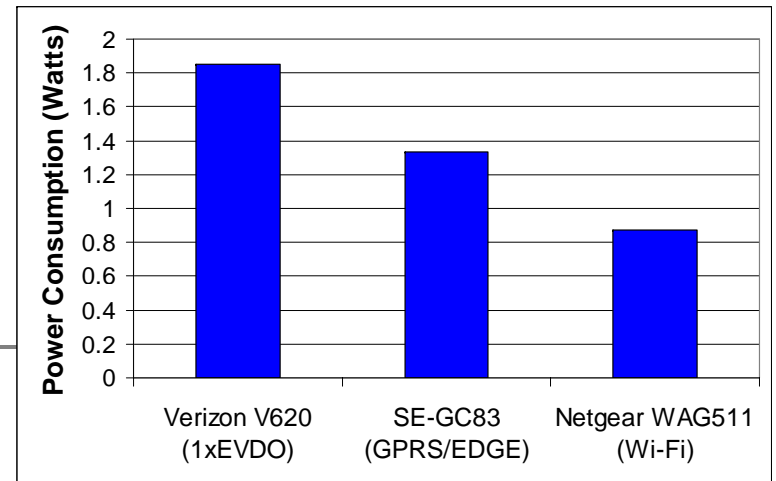


Wi-Fi  
 Bluetooth  
 Switch :  
 Wi-Fi -> BT

Lifetime (Hours of Usage)

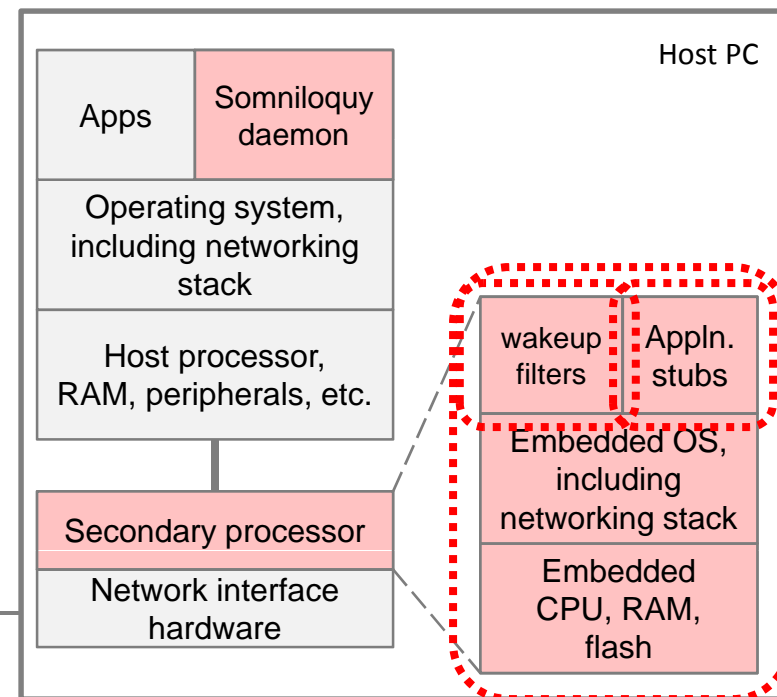


- 50% energy reduction with CoolSpots
- VOIP with Cell2Notify can reduce power 1.7-6.4x over WiFi and better than Cellular radios!



# Collaborating Processors

- ❑ Problem: Power State Design Runs Into Use Models
  - ❑ Hosts (PCs) are either Awake (Active) or Sleep (Inactive)
  - ❑ Power consumed when Awake = 100X power in Sleep!
  - ❑ Network: Assumes hosts are always “Connected” (Awake)
- ❑ Users want machines with the availability of active machine, power of a sleeping machine.



# Prototypes

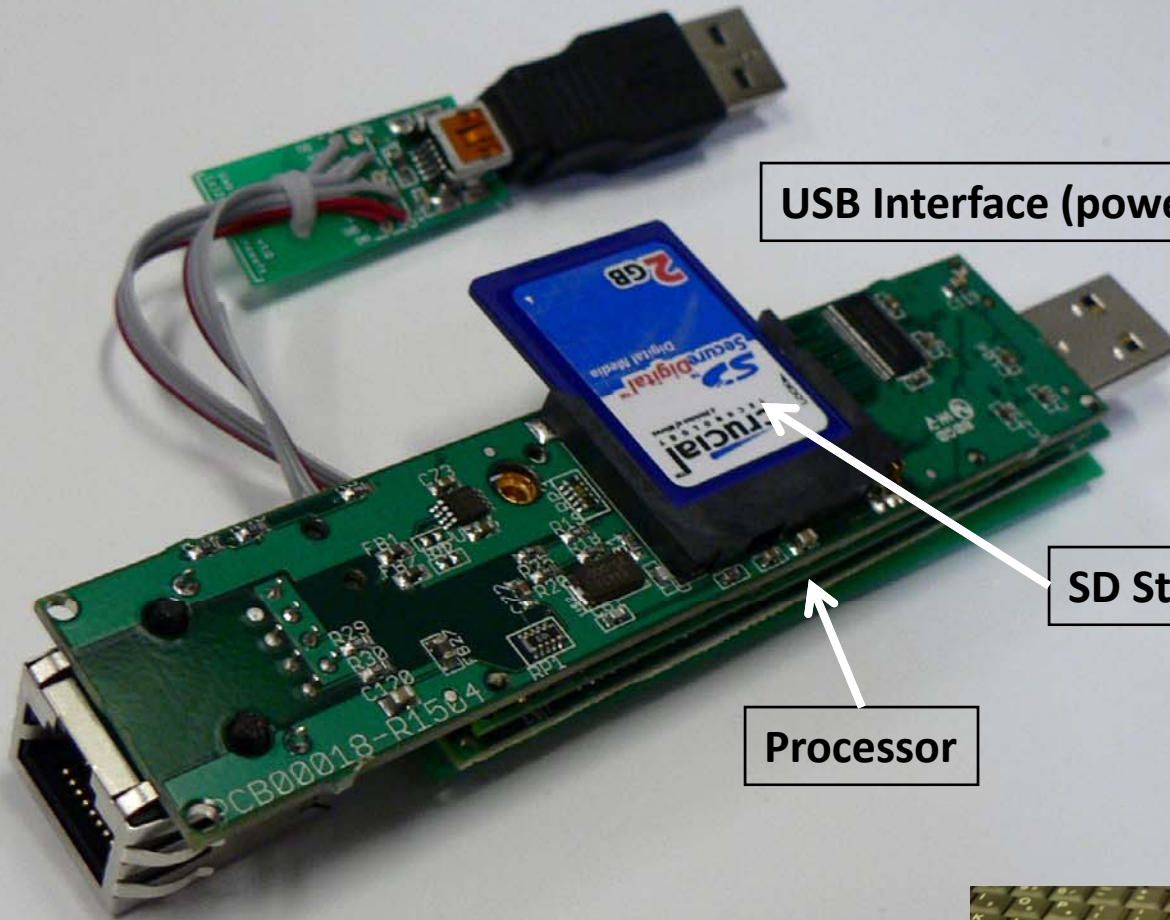
USB Interface (Wake up Host + Status + Debug)

USB Interface (power + USBNet)

SD Storage

Processor

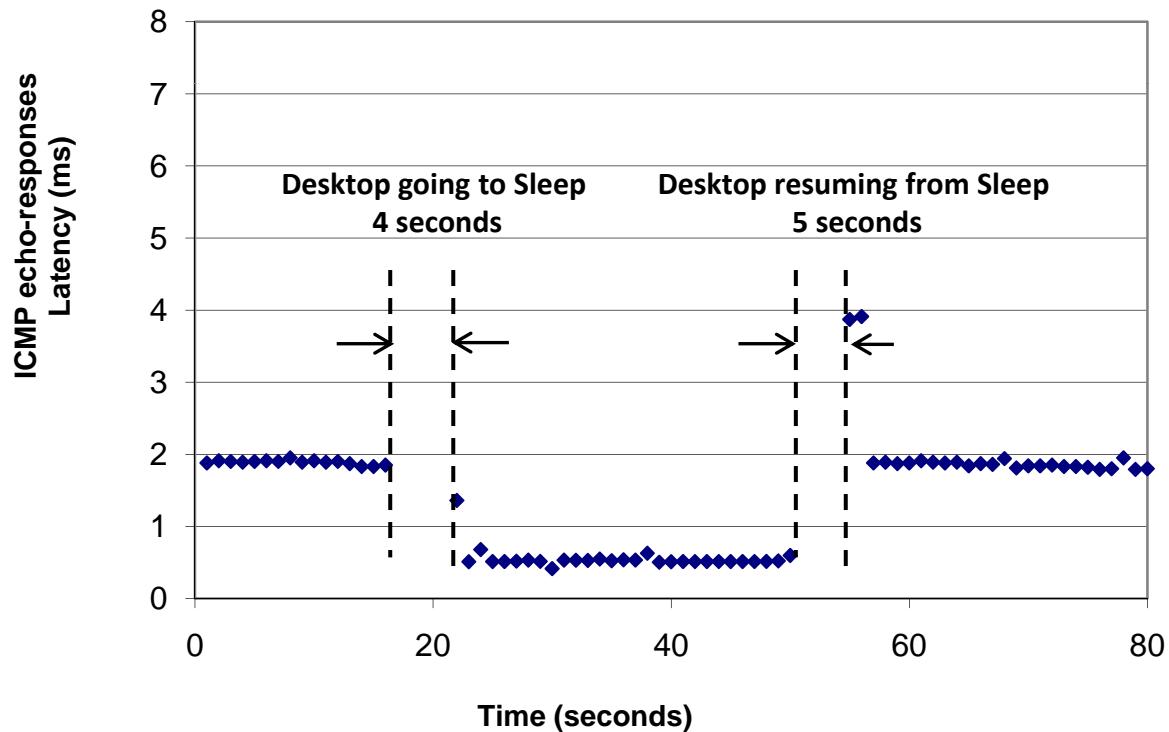
100Mbps Ethernet Interface





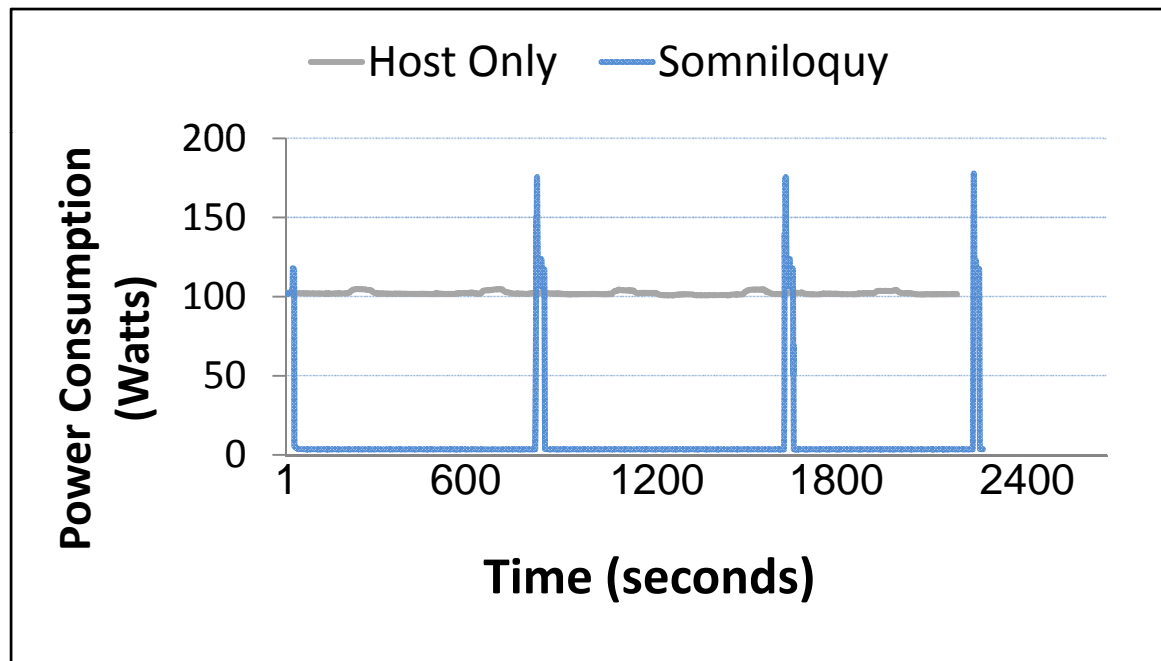
# Network, Application Level Reachability

- Respond to “ping”, ARP queries, maintain DHCP
- Maintain availability across the entire protocol stack
- E.g. ARP(layer 2), ICMP(layer 3), SSH (Application layer)



# Web downloads

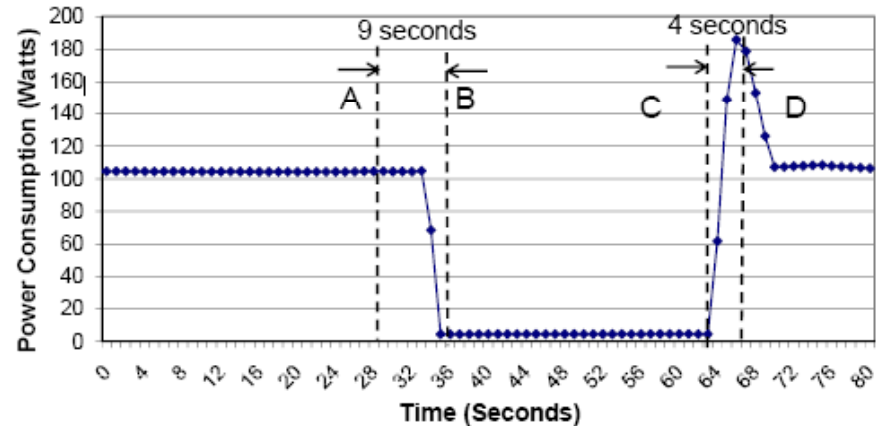
- 200MB flash storage, download when PC is asleep
  - Wake up PC and upload to PC when needed



**92% less energy than using the host PC for download**

# Desktops: Power Savings

State	Power
Normal Idle State	102.1W
Lowest CPU frequency	97.4W
Disable Multiple cores	93.1W
"Base Power"	93.1W
<b>Suspend state (S3)</b>	<b>1.2W</b>



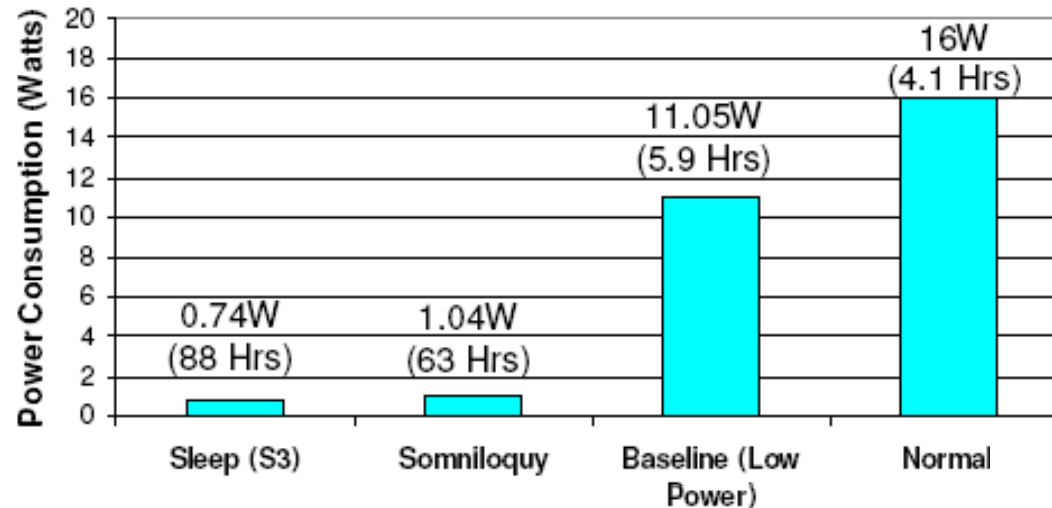
Dell Optiplex 745 Power Consumption and transitions between states

## Using Somniloquy:

- Power drops from  $>100W$  to  $<5W$
- Assuming a 45 hour work week
  - 620kWh saved per year
  - US \$56 savings, 378 kg CO<sub>2</sub>

# Laptops: Extends Battery Lifetime

IBM X60 Power Consumption



## Using Somniloquy:

- Power drops from  $>11W$  to  $1W$ ,
  - Battery life increases from  $<6$  hours to  $>60$  hours
- Provides functionality of the “Baseline” state
  - Power consumption similar to “Sleep” state

---

# Improving Energy Efficiency

Reduce distance (O1)

- Physical, logical

Minimize wasted work (O2)

- Shutdown, slowdown, procrastinate

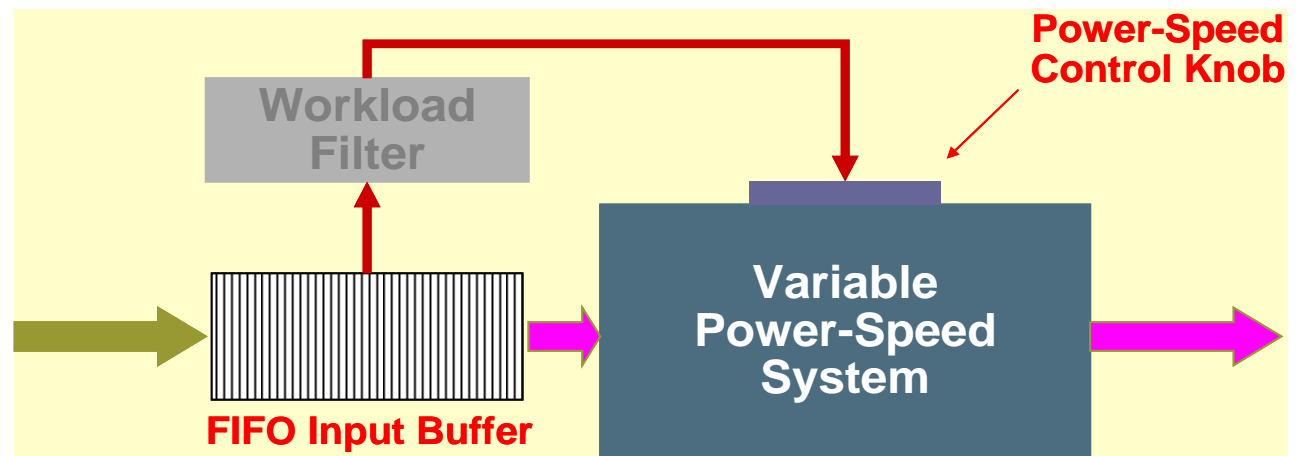
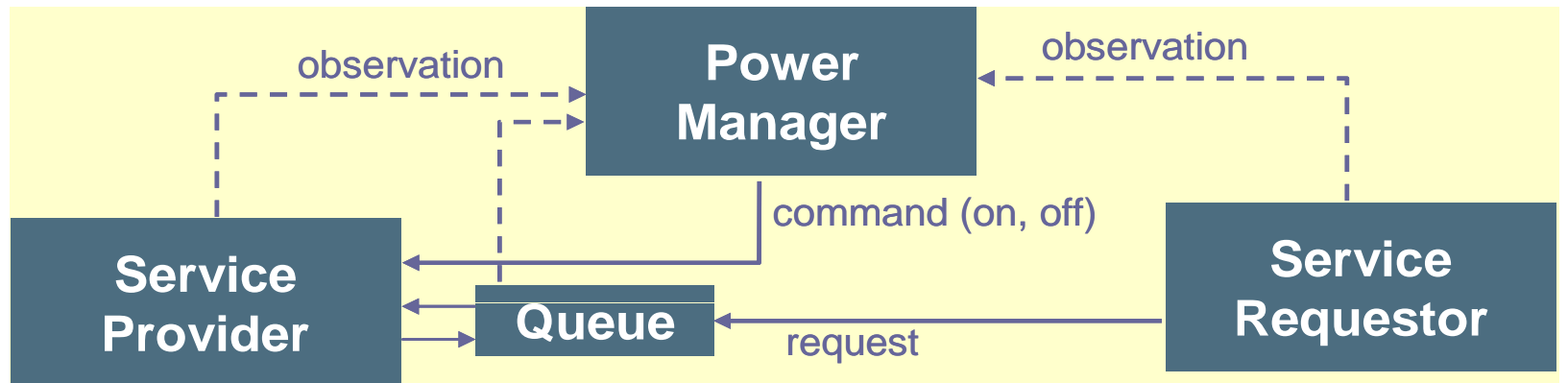
Specialized *heterogeneous* processing (O3)

- In a generalized execution environment

---

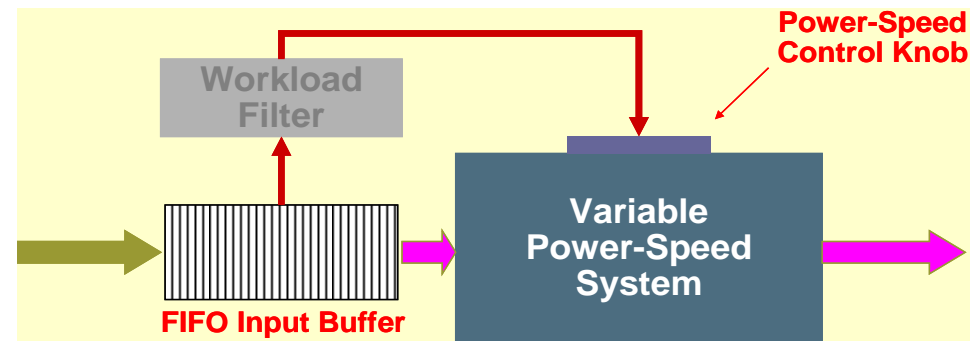
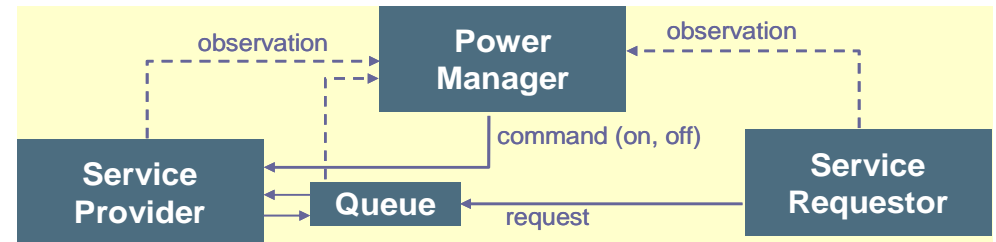
**Apply these lessons to build better architectures, power management algorithms.**

Algorithmically, there are basically two ways to save power



# Algorithmically, there are basically two ways to save power

- **Shutdown** through choice of right system & device states
  - Multiple sleep states
  - Also known as Dynamic Power Management (DPM)
- **Slowdown** through choice of right system & device states
  - Multiple active states
  - Also known as Dynamic Voltage/Frequency Scaling (DVS)
- DPM + DVS
  - Choice between amount of slowdown and shutdown



Competitive and Adversarial Approaches using Probabilistic Model Checking  
Machine Learning Techniques  
Convex Optimization for Thermally Efficient Chip Design

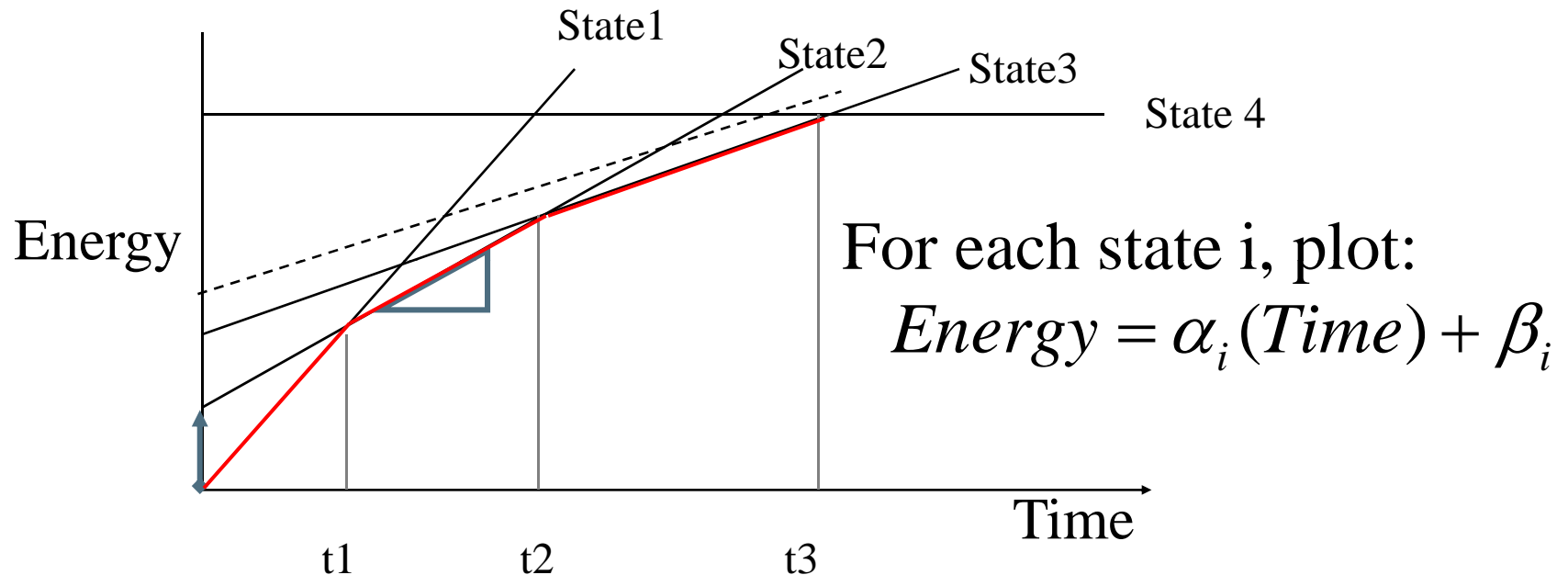
---

# Our Work In This Context

- Quantitative bounds on the quality of DPM algorithms based on Competitive Analysis [TCAD 01]
  - DPM strategies for devices with both multiple active and multiple sleep states [TCAD 02]
  - **Critical speed** when using DPM + DVS [SODA 03, TECS02]
  - **Optimized slowdown** methods under various timing scenarios [TCOM 06, TCAD 06, DAC 05-06, ECRTS 04-05]
  - Model the system as a game between DPM algorithm and an **non-deterministic adversary** to verify competitive ratio [TVLSI 05]
  - **Parameterized** job scheduling problems [DCOSS 08, INFOCOM 09]
-



# Multi-state DPM: Lower Envelope



- LEA can be deterministic or probabilistic

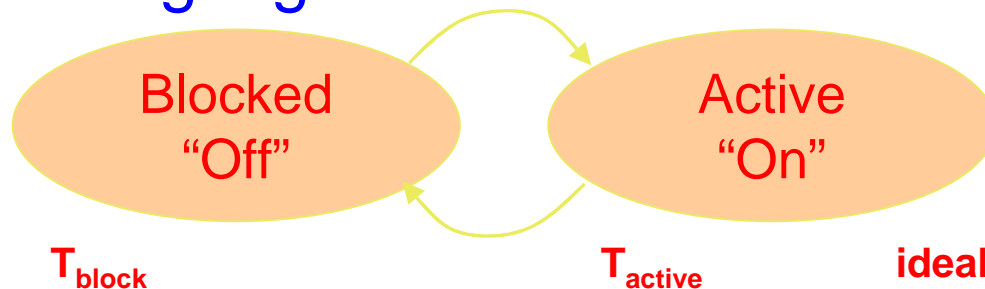
$$T_i = \arg \min_T \int_0^T [\alpha_{i-1}t + \beta_{i-1}] p(t) dt$$

$$+ \int_T^\infty [\alpha_{i-1}T + \alpha_i(t - T) + \beta_i] p(t) dt$$

- PLEA is  $e/(e-1)$  competitive.

# Lessons from Slowdown, Shutdown

- Slowdown eventually reaches a limit w.r.t. to work done, quality, timing
- Shutdown keeps giving *if*
  - There is heterogeneity: large difference between “on” and “off” power
  - Keep finding opportunities to duty-cycle actions by using higher level semantics.

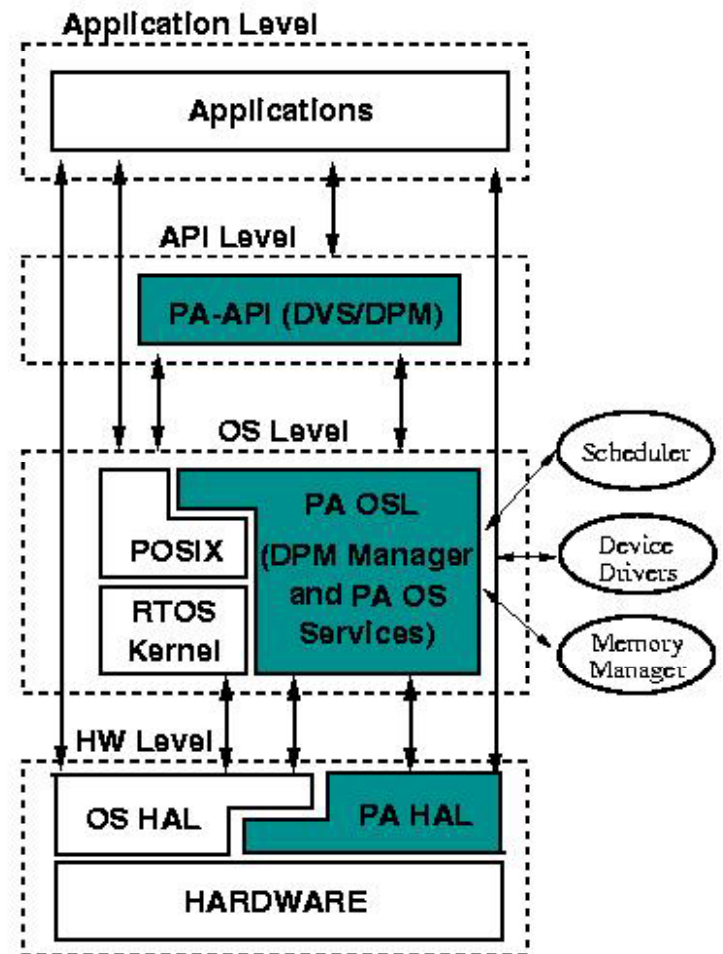


ideal improvement =  $1 + T_{\text{block}}/T_{\text{active}}$

**Need to reach higher layers for shutdown → power/energy awareness.**

# What does it mean to be 'aware'?

- That the application and the services **know** about energy, power
  - File system, memory management, process scheduling
  - Make each of them energy aware
- How does one make software to be "aware"?
  - Use "reflectivity" in software to build adaptive software
  - Ability to reason about and act upon itself (OS, MW)

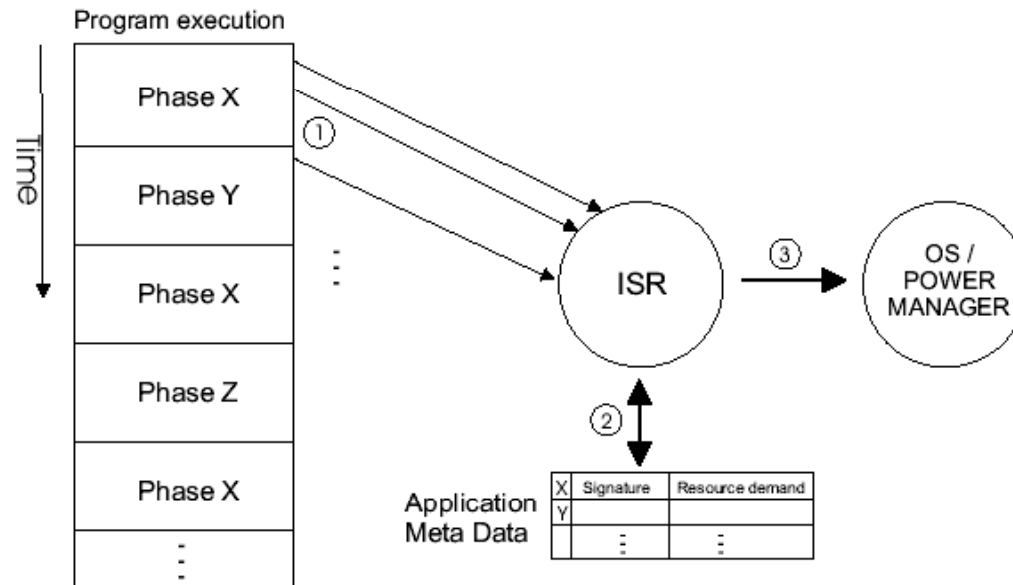


---

# Example: Program Phases & Power Control

1. Characterize application offline
    - Divide an application into phases of execution
      - A group of program intervals executing similar code
    - Each phase has similar demand on resources, energy use
      - Similar code, similar resource demands (memory, IPC)
  2. Annotate source code
    - Phase signatures
  3. Enable OS (and hardware) to recognize signature
    - Smart hardware and/or online learning techniques
  4. Dynamically tune the power manager
    - As application moves from one phase to another.
-

# Matching Signatures at Runtime

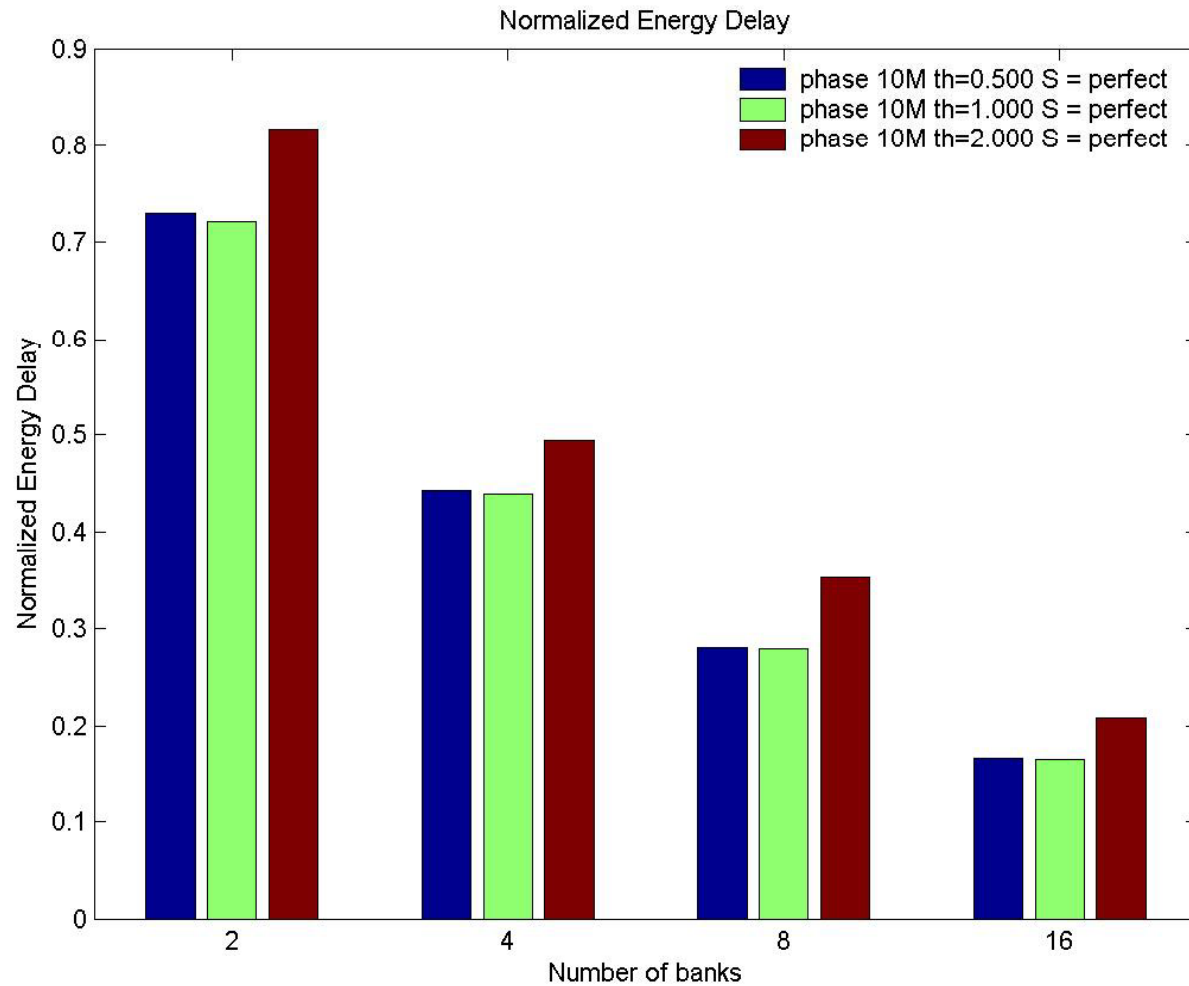


- Use performance counters:
  - Can be programmed to generate an interrupt on specified counts
- ISR provides matching with the meta data and mode changes
  - Every  $S \cdot 10,000$  loop branches try a match
  - Phase matching can also be done in hardware

---

- Notify power manager to trigger proper action (memory bank shutdowns)

# Results – Normalized to NAP



Average among bzip, mpeg, ghostscript and ADPCM

# Results - overheads

- Approx. 350K instructions for every 10,000 loop branch instructions
- Number of instructions executed by the match algorithm at every 10,000 loop branches to match a partial signature (500 instructions per phase)

# of phases	# instructions	overhead
5	2,580	0.7%
10	4,500	1%
20	8,280	2%
30	12,060	3%

- Size overhead. 4 bytes per inter arrival estimate per bank / phase. ~~4 x 16 x 10 = 640 bytes assuming 16 banks and 10 phases.~~
- The signatures take 1280 bytes for 10 phases. Total of 2KB of meta data

# Takeaways

- Algorithmically we look for the right combination of slowdown and shutdown strategies
  - Driven by increasingly real, accurate and timely sensor data that push the available slack to thermal limits
- Architecturally we look for the right organization of components for maximal duty cycling
- Future increases in energy efficiency lie in architectures that enable aggressive duty cycling
  - By continually reaching to the higher levels of decision making, capturing intent.

**“Future lies in system architectures built for aggressive duty-cycling”**



# Power Management in Mixed Use Buildings

- 500 occupants, 750 machines (nom.)
- Detailed instrumentation to measure macro and micro-scale power use
  - 39 sensor pods, 156 radios, 70 circuits
  - Subsystems: Air Conditioning, Light

