Energy-aware adaptation for mobile applications

Original slides prepared by
Shravan Rayanchu

Energy efficiency

- In the previous class, we covered
  - WiFi energy consumption issues
  - IEEE 802.11 PowerSave Mode
    - Static PSM, Adaptive PSM
  - Issues with PSM
    - Naïve usage can cost more energy! (e.g., NFS)
    - *STPM*: Energy Vs. Performance tradeoff
    - Background traffic can increase energy
      - Priority queuing, Dynamic beacon periods (*NAPman*)

Energy-aware adaptation for mobile applications

Jason Flinn and M. Satyanarayanan

Motivation

Energy is a vital resource for mobile computing.

- Battery technology improving slowly.
- Improvements from hardware power management.

Can the Operating System help?

Yes!   Through **energy-aware adaptation:**

- Applications dynamically change their behavior to conserve energy (trade data fidelity for energy usage)
- System helps strike appropriate balance.

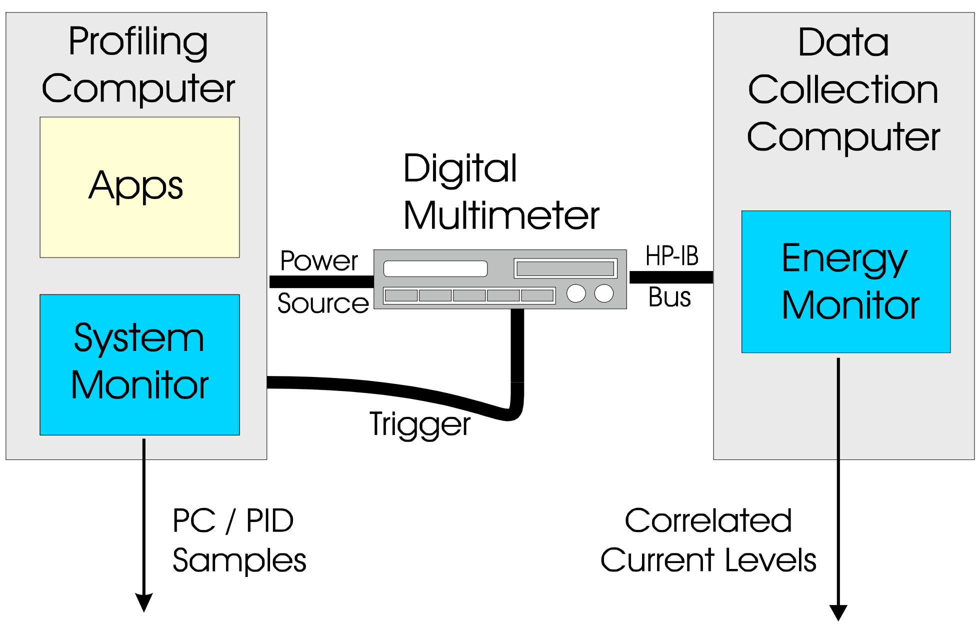
PowerScope: Profiling Energy Use

First stage: **sample collection**

- Digital multimeter samples power levels.
- Kernel instrumentation samples system activity.

Second stage: **off-line analysis**

- Generate profile from samples and symbol tables.



Energy Profiling

Process	CPU Time(s)	Total Energy(J)	Average Power(W)
/usr/odyssey/bin/xanim	66.57	643.17	9.66
/usr/X11R6/bin/X	35.72	331.58	9.28
Kernel	50.89	328.71	6.46
Interrupts-WaveLAN	18.62	165.88	8.91
/usr/odyssey/bin/odyssey	12.19	123.40	10.12
Total	183.99	1592.75	8.66

Energy Usage Detail for process /usr/odyssey/bin/odyssey

Procedure	CPU Time(s)	Total Energy(J)	Average Power(W)
_Dispatcher	0.25	2.53	10.11
_IOMGR_CheckDescriptors	0.17	1.74	10.23
_sftp_DataArrived	0.16	1.68	10.48
_rpc2_RecvPacket	0.16	1.67	10.41
_ExaminePacket	0.16	1.66	10.35

Fraction of Energy Consumed by:

- Process
- Procedure

Samples:

(PID, PC, Power)

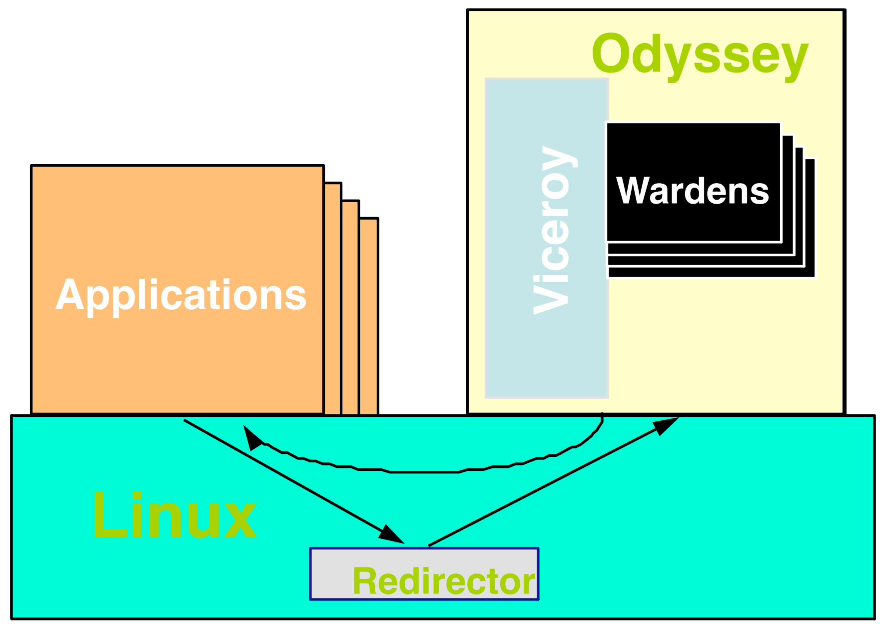
Use Symbol Table

Odyssey

Applications trade data fidelity for resource usage.
(Bandwidth drop: Video Color to Black & White, Maps with lesser detail)

Fidelity: degree to which data matches reference copy.

Odyssey periodically:
• predicts energy demand.
• measures energy supply.
• triggers adaptations.



Viceroy: Monitors resource usage
Wardens: Encapsulates
type-specific functionality (for different fidelity levels)

Odyssey

Measured the impact of reducing data fidelity for:

- Video player.
- Speech recognizer.
- Map viewer.
- *Web browser.*
- Concurrently executing applications.

Applications modified to interact with Odyssey

- Odyssey notifies through **upcall**
- Applications adjust fidelty, communicate new set of expectations
- Web browser not modified (proxy used)

PowerScope profiles energy usage (at 633 Hz.).

Methodology

- Power Management

  Power down as many components as possible

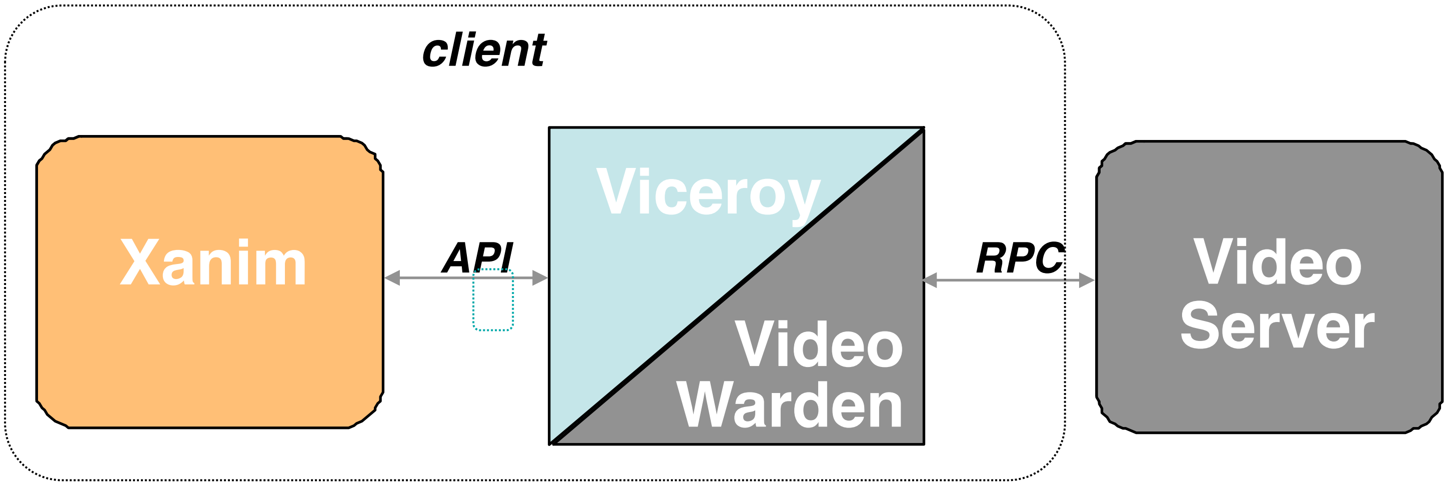
  – Disk in Standby mode after 10 sec of inactivity
- Experimental set up:
  – 233 MHz IBM ThinkPad, 64MB memory
  – 2Mbps wireless card (WaveLAN)
  – 200 MHz Pentium Pro (servers)

Power Consumption

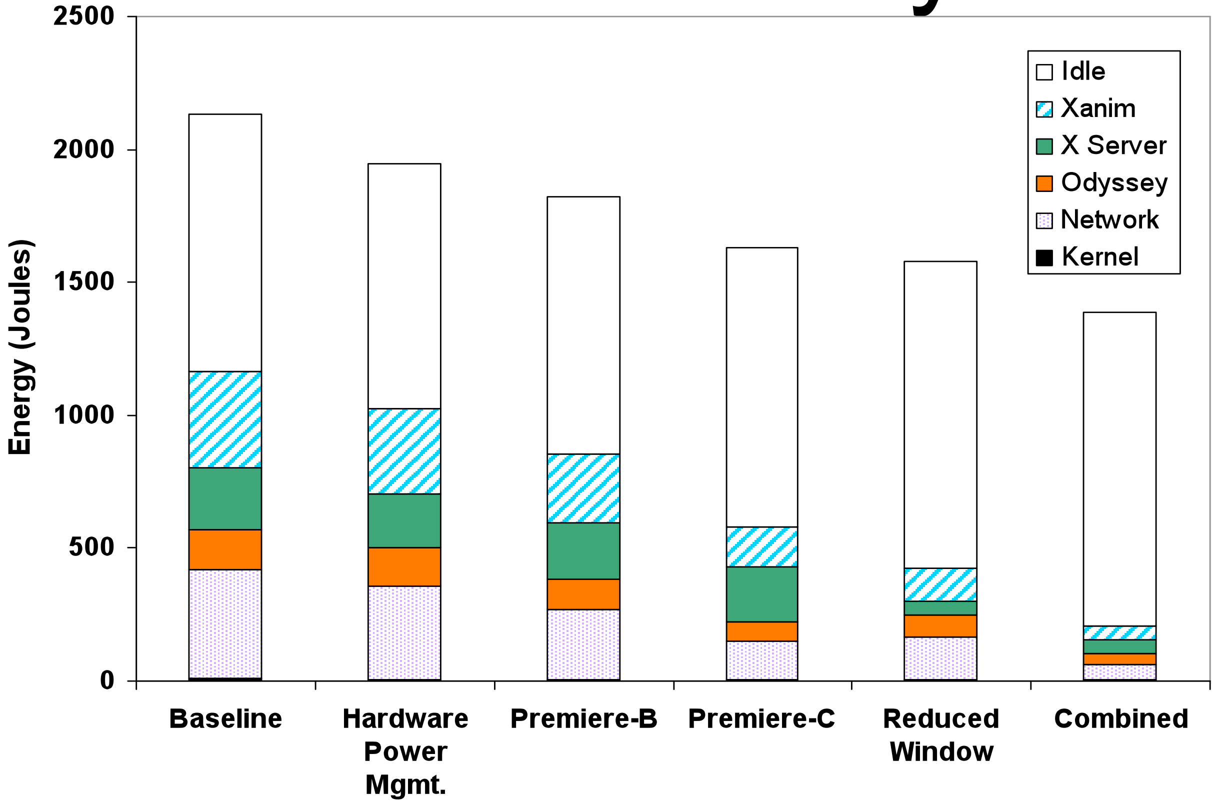
Component	State	Power (W)
Display	Bright	4.54
	Dim	1.95
WaveLAN	Idle	1.46
	Standby	0.18
Disk	Idle	0.88
	Standby	0.24
Other	Idle	3.20

Application: Video Player

- Fetches video data from remote server.
- Two dimensions of data fidelity:
  - Compression (baseline, Premiere-B, Premiere-C).
  - Display window size (baseline, half width/height).



Impact of Fidelity on Video Player



Network : Bottleneck

Most energy:

Processor Idle state

Power Mgmt: 9-10%,

Only Disk is off

Fidelity:

Compression: 17%

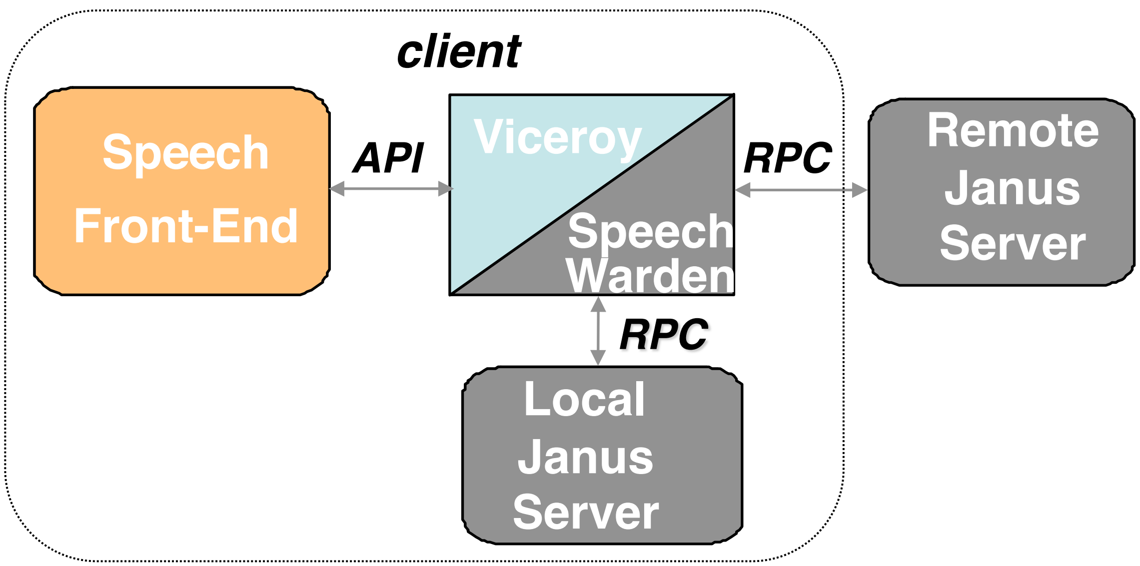
Window size: 20%

Combined: 35%

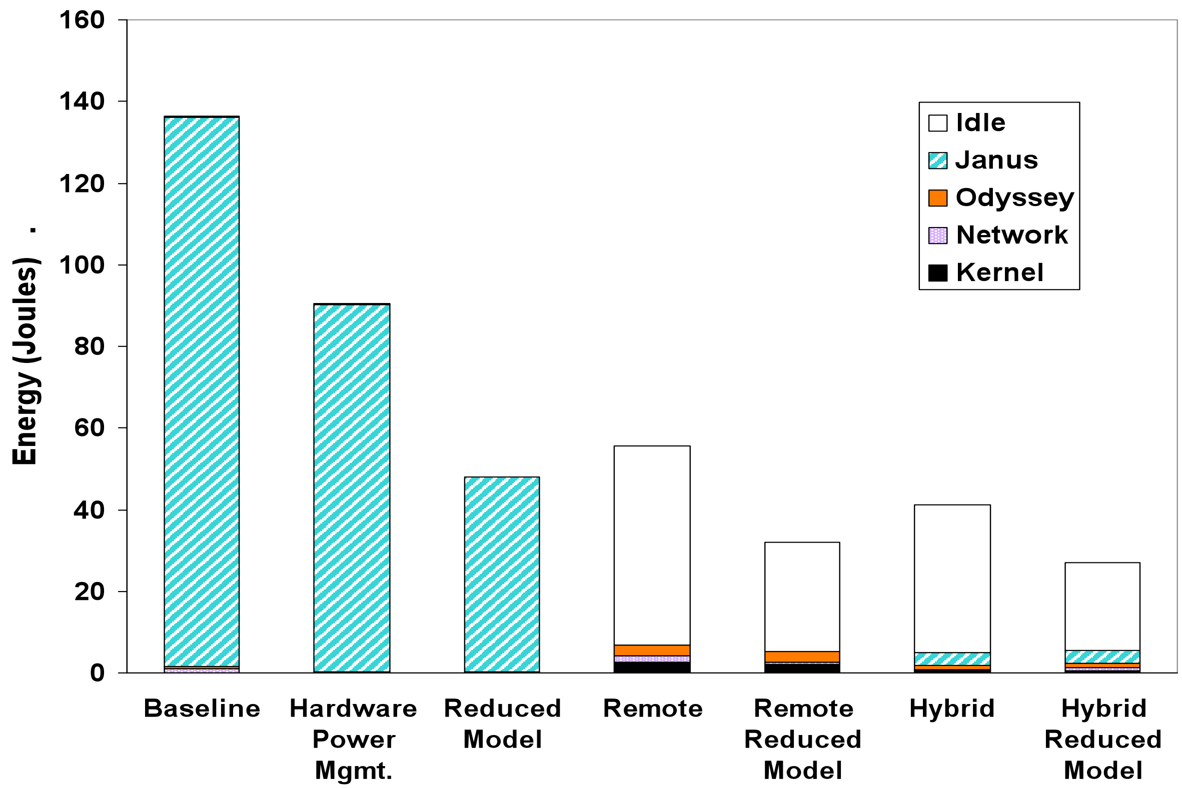
(Further reduction less : mostly “Idle”)

Application: Speech Recognition

- Speech-to-text translation of spoken utterances.
- Two dimensions of fidelity:
  - Full or reduced speech model.
  - Local, remote, or hybrid recognition.



Impact of Fidelity on Speech Recognition



Janus : Bottleneck (computation)

Power Mgmt: 34%,
Display, Disk, n/w off (local)

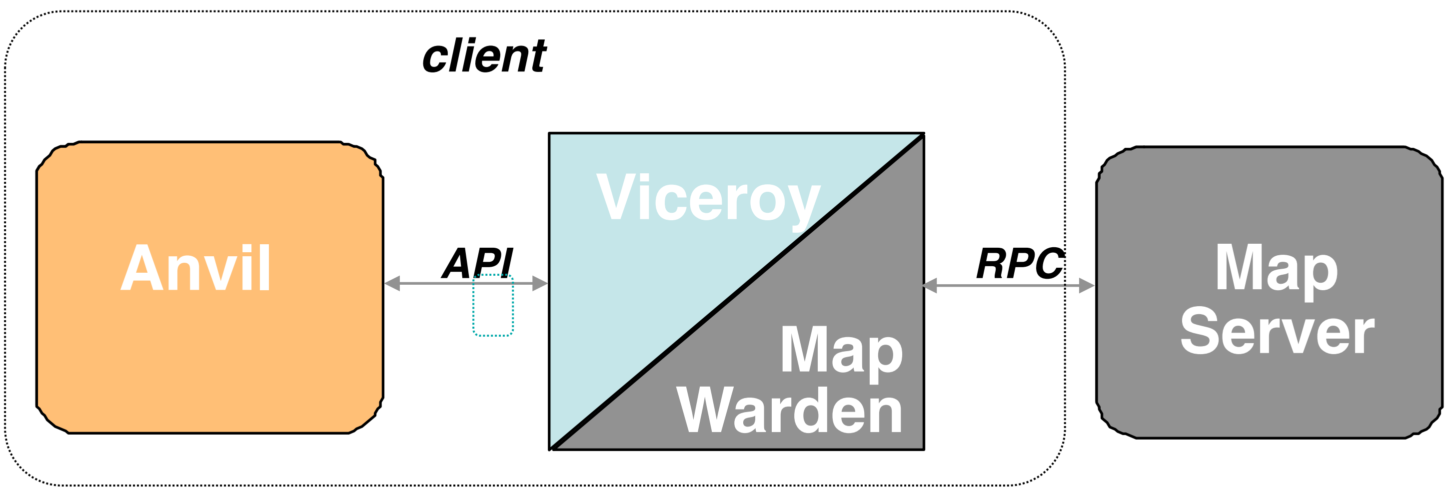
Fidelity: 50-65% reduction

Remote: 44%
65% (less fidelity)
“Idle” mostly

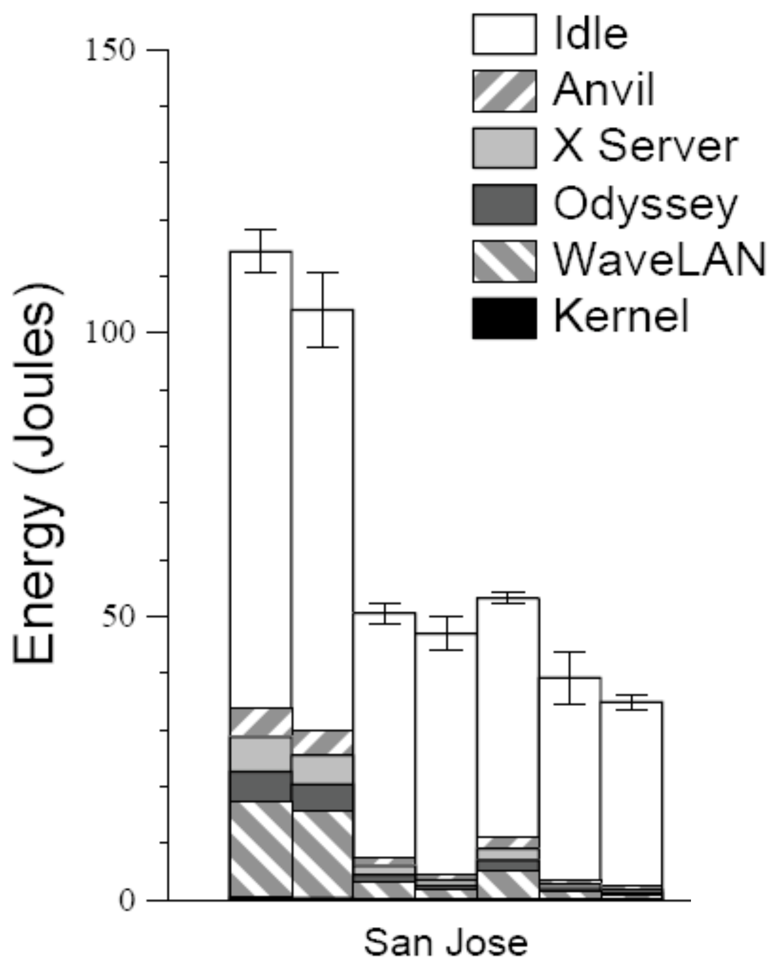
Hybrid: 55%
75% (less fidelity)

Application: Map Viewer

- Fetches maps remote server.
- Two dimensions of data fidelity:
  - Filtering (eliminate secondary roads etc).
  - Cropping (restrict data to a geographic subset of original map).



Impact of Fidelity on Map Viewer

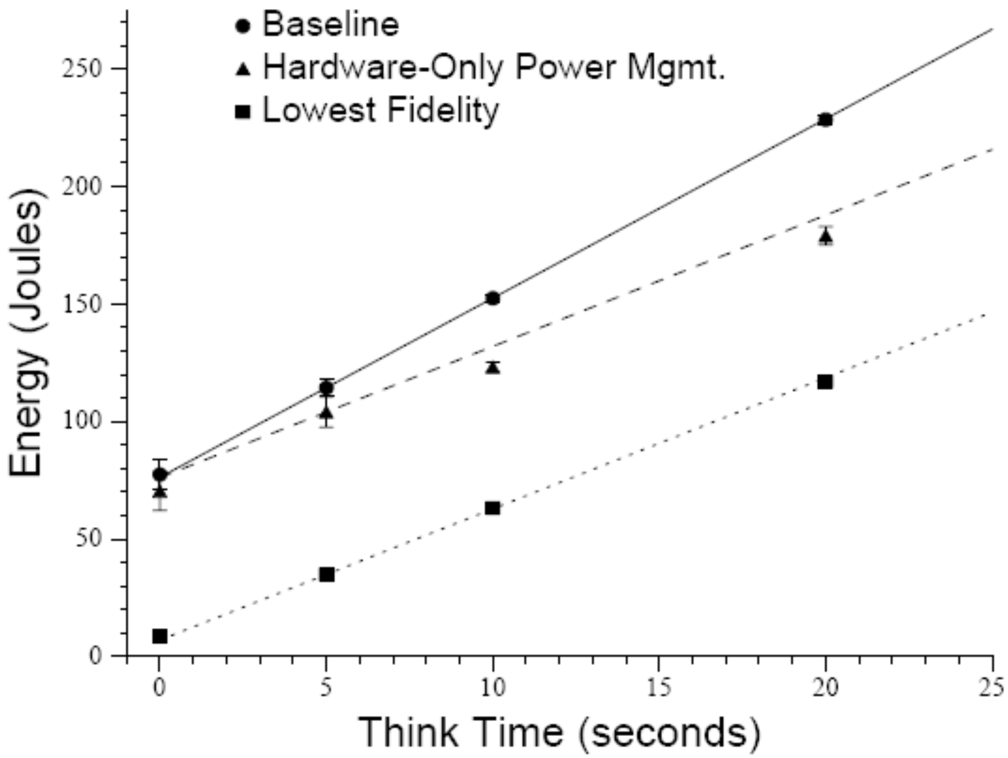


Power Mgmt: 20%
(Think time, n/w off)

Cropping + Filtering:
45-70% reduction

Baseline, Power mgmt, filters (minor, secondary), cropping, crop+filter

Effect of User Think Time



$$E_T = E_O + t \,.\, P_B$$

Divergent lines: Power Mgmt scales linearly with think time

Parallel lines: Fidelity reduction independent of think time

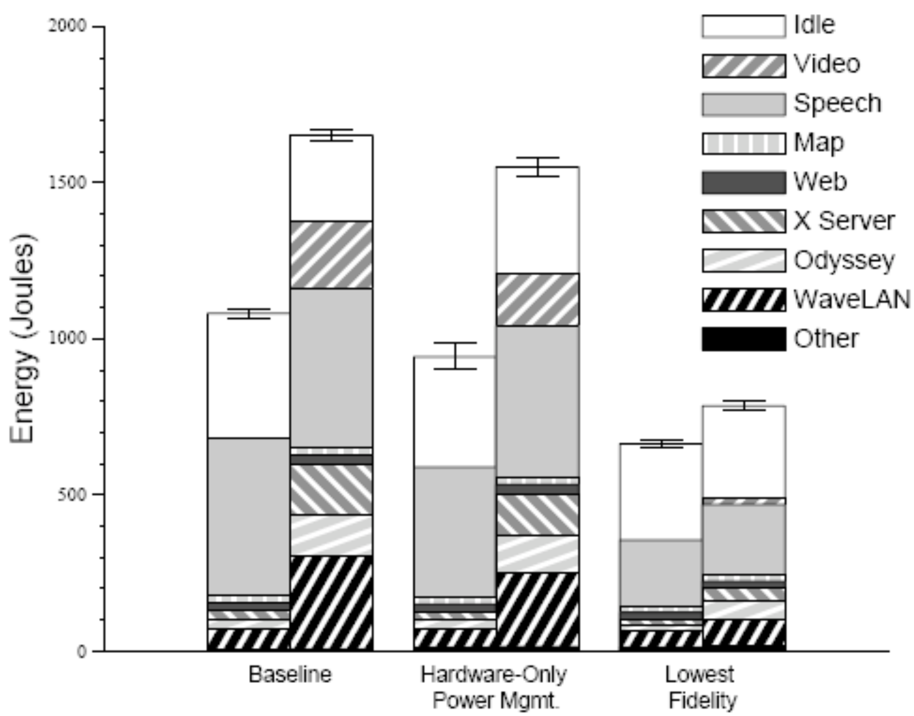
Summary: Impact of Fidelity

Application	Baseline	Hardware Power Mgmt.	Fidelity Reduction	Combined
Video	1.00	0.91	0.84	0.65
Speech	1.00	0.66	0.28	0.24
Map	1.00	0.85	0.51	0.40
Web	1.00	0.76	0.93	0.69

Conclusions about the impact of reducing data fidelity:

- Can significantly reduce application energy usage.
- Complementary to hardware power management.

Effect of Concurrent Applications



Composite application, video application

Energy goes up/down ??

Addition of video app -

Baseline: 53% more

Power Mgmt: 64% more!

(Reduced opportunities for power mgmt)

Min. Fidelity: 18% more

Background power usage

amortized by second application

Zoned Backlighting

The display is the Achilles heel of power management.

What if we could selectively illuminate different areas?

- Divide screen into independently controlled zones.
- When battery is critical, dim or disable unused zones.
- Applications could adapt to use fewer zones.

What is the possible benefit of zoned backlighting?

- Assume energy usage is proportional to zone area.

	Combined Energy Usage			
Application	Baseline	No Zones	4 Zones	8 Zones
Video	1.00	0.65	0.49	0.47
Map	1.00	0.40	0.33	0.31

Goal-Directed Adaptation

User can often estimate needed battery duration:

- Length of a meeting, flight, etc.

Applications provide multiple data fidelities.

- At run time, system directs adaptation.

System directs adaptation with the following goals:

- Meet the specified duration whenever possible.
- Maximize application fidelity.
- Minimize number of adaptations.

Predicting Future Energy Demand

Use smoothed observations of past power usage:

$$New = (1-\alpha) \bullet (sample) + \alpha \bullet Old$$

Multiply by time remaining to predict energy demand.

α **varies as energy drains**:

- When goal is distant, large α yields stability.
- When goal is near, small α yields agility.

Calculate α so that half-life of decay function is 10% of time remaining.

Determining Energy Supply

Energy supply is residual energy in battery.

Prototype currently uses external equipment.

- Assumes known initial value.
- Digital multimeter samples power usage (10 Hz.)

Alternative implementations:

- Gas-gauge ICs (Smart Battery).
- PCMCIA multimeter.
- Built-in monitoring capability.

Triggering Adaptation

When demand exceeds supply:

- Applications adapt to conserve energy usage.

When supply significantly exceeds demand:

- Applications increase data fidelity.

Hysteresis prevents frequent adaptations.

(difference in supply and demand = level of hysteresis)

$$5\% \text{ of residual energy} + 1\% \text{ of initial energy}$$

When multiple applications are executing:

- Static priorities determine which adapts.

Evaluating Goal-Directed Adaptation

Client: 233 MHz Pentium laptop.
Servers: 200 MHz Pentium Pro desktops.
Network: 2 MB/s campus wireless WaveLAN.

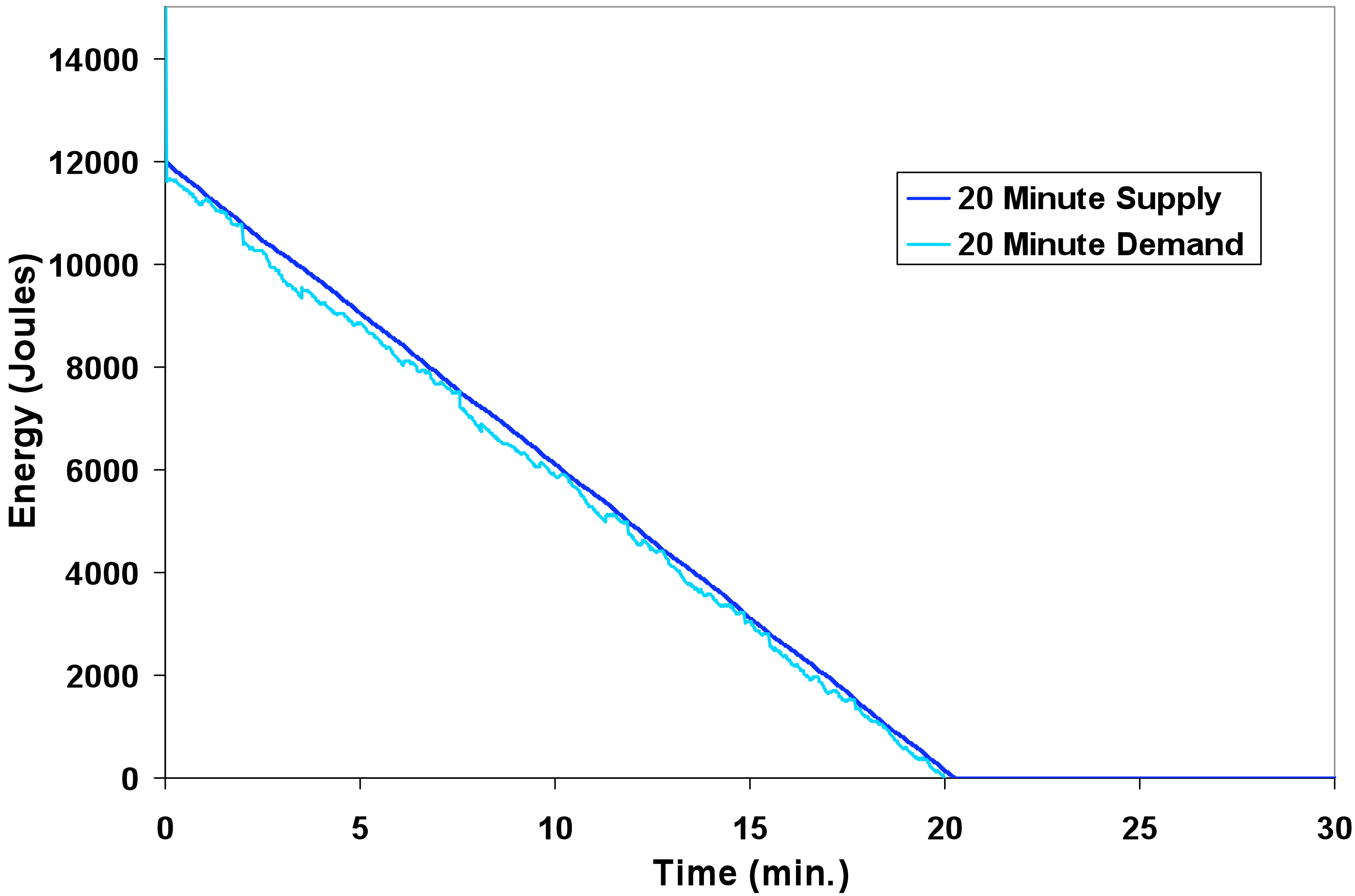
Multiple energy-aware applications run concurrently:

- Speech recognizer.
- Video player.
- Map viewer.
- Web browser.

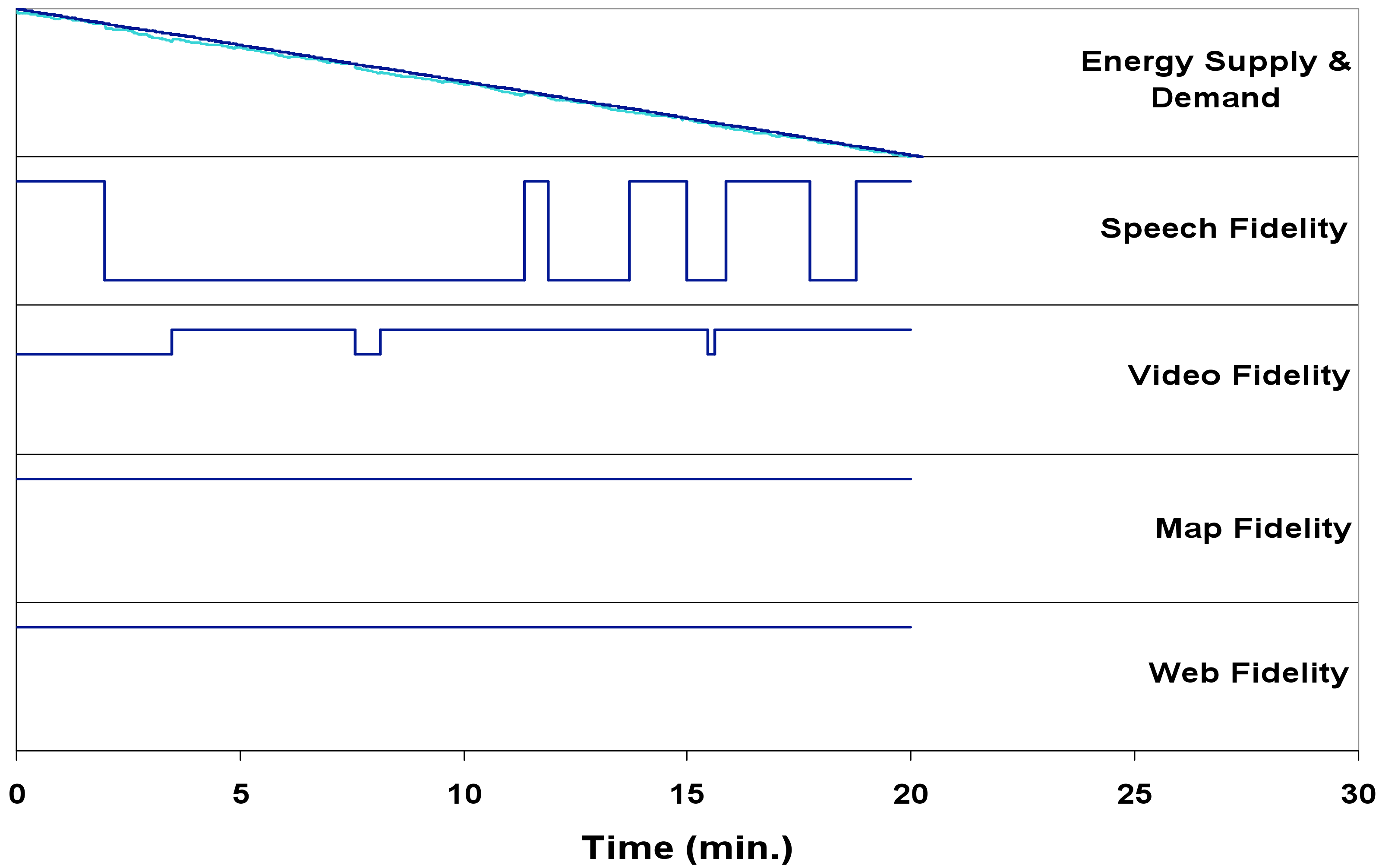
Emulate 12 KJ. energy supply (14% of laptop battery).

- Lasts 19:27 at maximum fidelity, 27:06 at minimum.
- Specify time goals of 20, 22, 24, and 26 minutes.

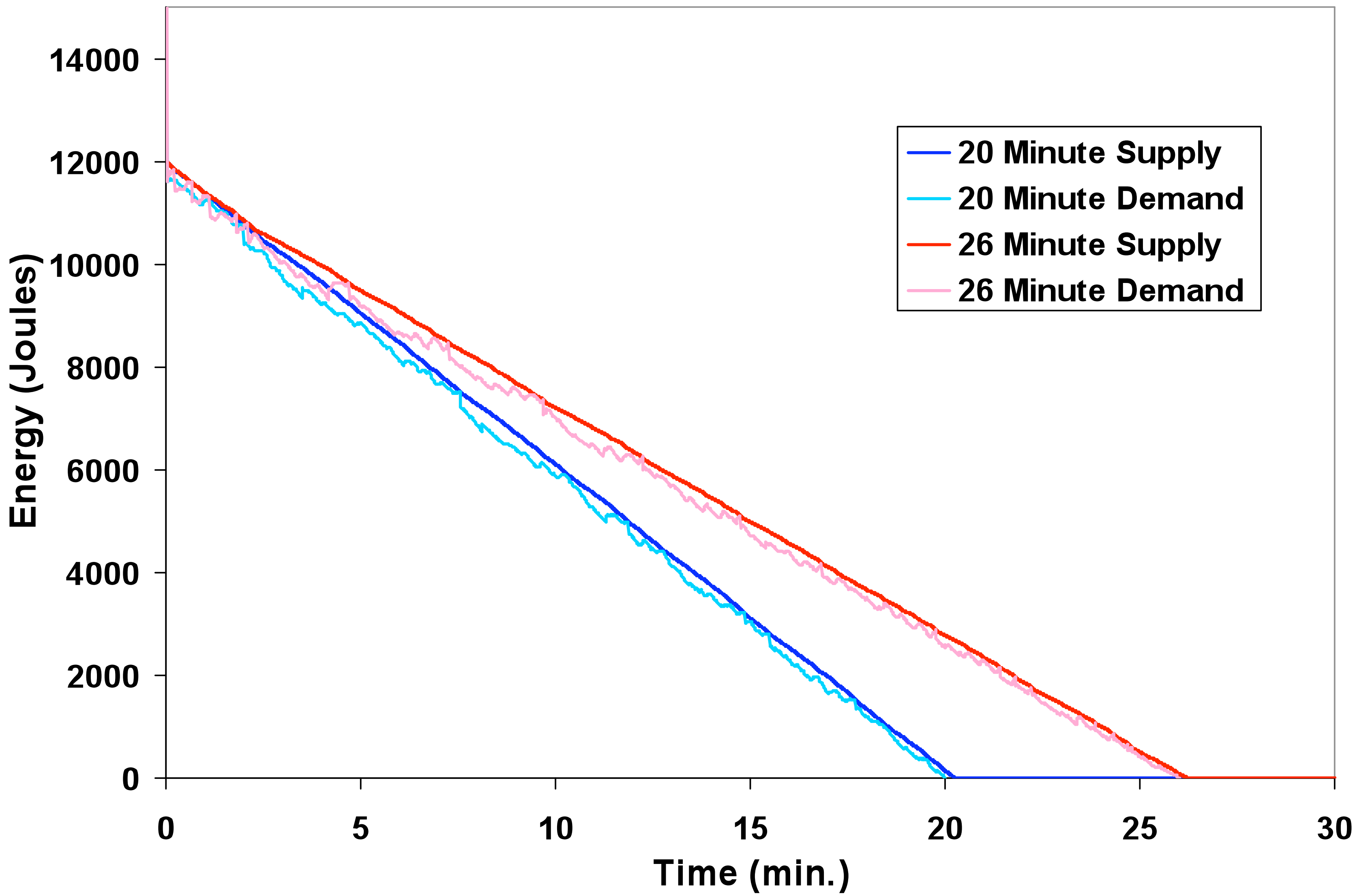
Energy Supply & Demand



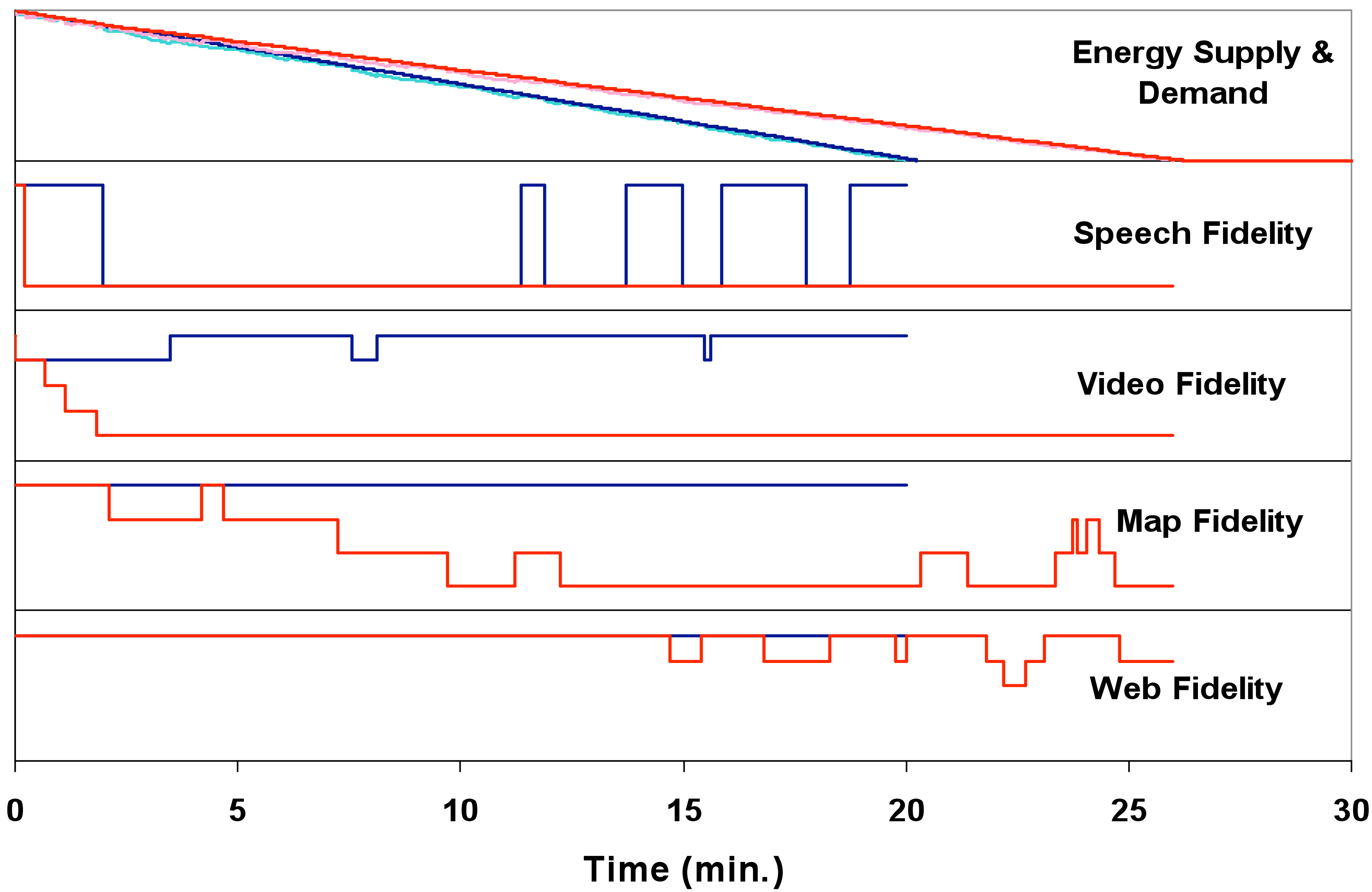
Time Goal: 20 Minutes



Energy Supply & Demand



Time Goal: 26 Minutes



Results

Specifed Duration (s.)	Goal Met	Residue	
		Energy (%)	Time (s)
1200	100%	1.21%	15.3
1320	100%	0.90%	12.9
1440	100%	0.84%	13.0
1560	100%	0.50%	8.7

Goal is met in every trial.

Residual energy is low.

Other experiments show similar results for larger energy supply, modified time goal, and bursty workload.

Questions?