Energy Management for Smartphones

Source: K. Naik, A Survey of Software Based Energy Saving Methodologies for Handheld Wireless Communication Devices,

Tech. Report No. 2010-13

Dept. of ECE, University of Waterloo

Outline

2. Overview
3. Smart Batteries
4. Energy Efficient GUI Design
5. Sleep to Save Energy
6. Proxy Assisted Energy Saving
7. Source-level Power Control

Note: The sequence starts with 3 to make it compatible with the notes.

Problem Description

Running apps

Multimedia Streaming

Web Browsing

Social Networking

Gaming

MORE

Resources

Comm. Bandwidth

Processing Power

Sensors

User Interface

MORE

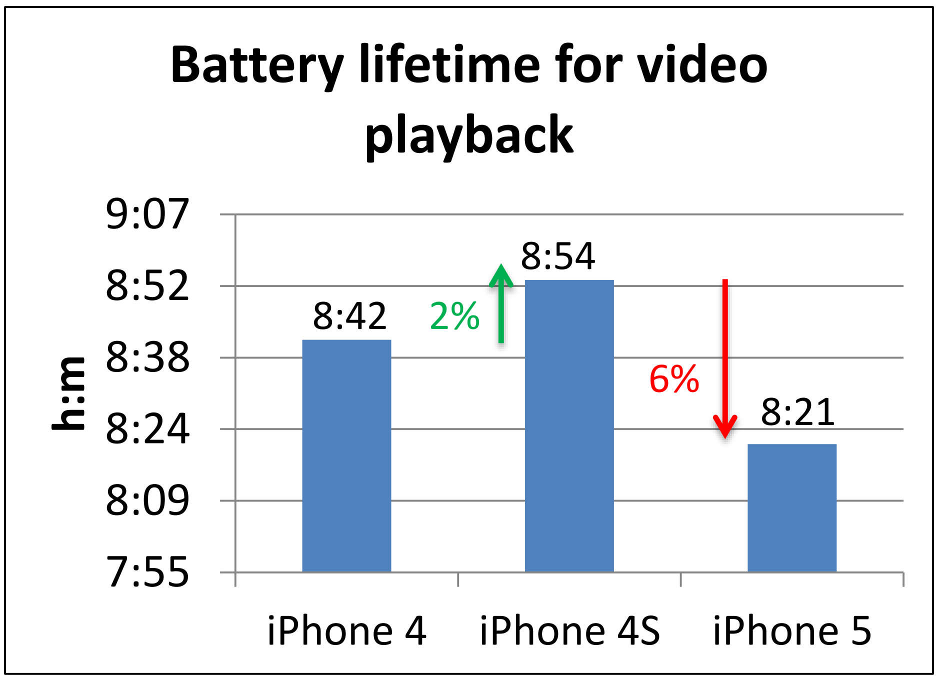
Much Energy Consumption

Introduction

Problem Description

Smartphone Battery

- Battery technology is not keeping up with the rapidly growing energy demands of smartphones.



Phone	Release Year	Battery Capacity	
iPhone 3GS	2009	1250 mAh	
iPhone 4	2010	1420 mAh	↑13 %
iPhone 4S	2011	1432 mAh	↑1%
iPhone 5	2012	1440 mAh	↑0.5 %

Less than **5%** annual growth of battery capacity

Introduction

Solution Strategies

What can be done?

- Hardware
  - Low Power Electronics
  - Different Power States
- Communication
  - Networks
  - Servers
  - Clients
- Software
  - Operating Systems
  - Application Programs

3. Smart Batteries

An ordinary AA battery

A smart battery

The 4 pins of a commonly used smart battery

- Ground
- Positive
- Data
- Temperature

3. Smart Batteries

SoC: State of Charge

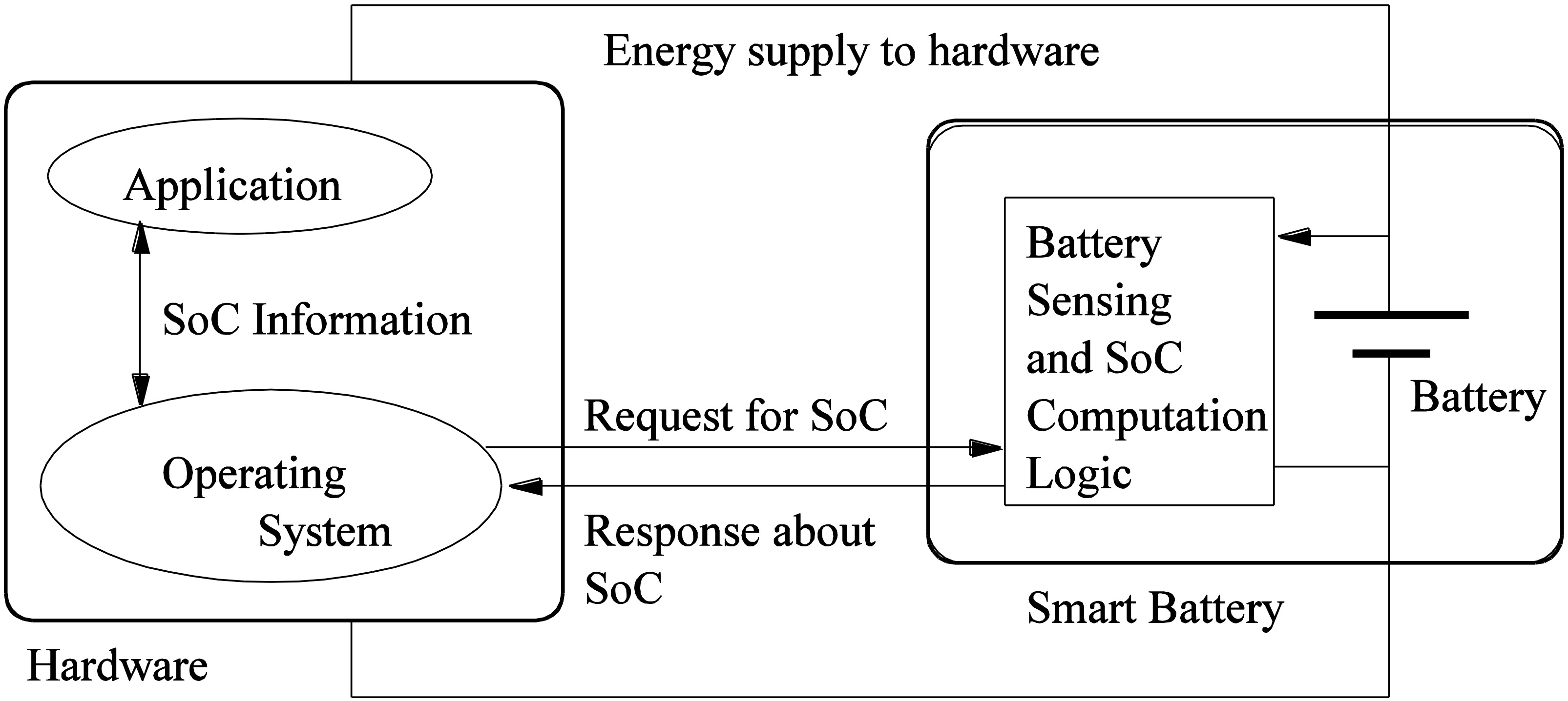


Figure 3.1: Conceptual relationship between a smart battery and an application.

3. Smart Batteries

- **Full Design Capacity:** It is the remaining capacity of a **newly** manufactured battery.

- **Full Charge Capacity:** It is the remaining capacity of a fully charged battery at the beginning of a discharge cycle.
- **Theoretical capacity:** It is the maximum amount of charge that can be extracted from a battery based on the amount of **active materials** it contains.
- **Standard Capacity:** It is the amount of charge that can be extracted from a battery when discharged under **standard load** and temperature condition.
- **Actual Capacity:** It is the amount of charge a battery delivers under given load and temperature condition.

3. Smart Batteries

- Battery discharge behavior is affected by:
  - Discharge rate
    - Higher discharge rate ➔ reduced battery capacity
  - Temperature
    - Below room temp.: Charge capacity decreases
    - At high temp.: Actual delivered capacity reduces
  - # of Charge/Discharge cycles
    - Lithium-Ion batteries lose a portion of their capacity with each Charge/Discharge cycle due to electrolyte decomposition.

➔Capacity fading

Accurate estimation of SoC parameter is a difficult task.

3. Smart Batteries

- Predicting the lifetime of a battery is a difficult problem.
  - The actual capacity is a function of the physical and chemical properties of the battery and the **dynamic load**.

- Examples of battery models
  - Ideal model (L = C/I, where L is lifetime, C is charge capacity, and I is constant current)
  - Electrochemical model: Dualfoil is a battery simulator.
  - Electrical circuit model: PSpice is a circuit simulator.
  - Stochastic model
  - Kinetic battery model
  - Diffusion model

3. Smart Battery

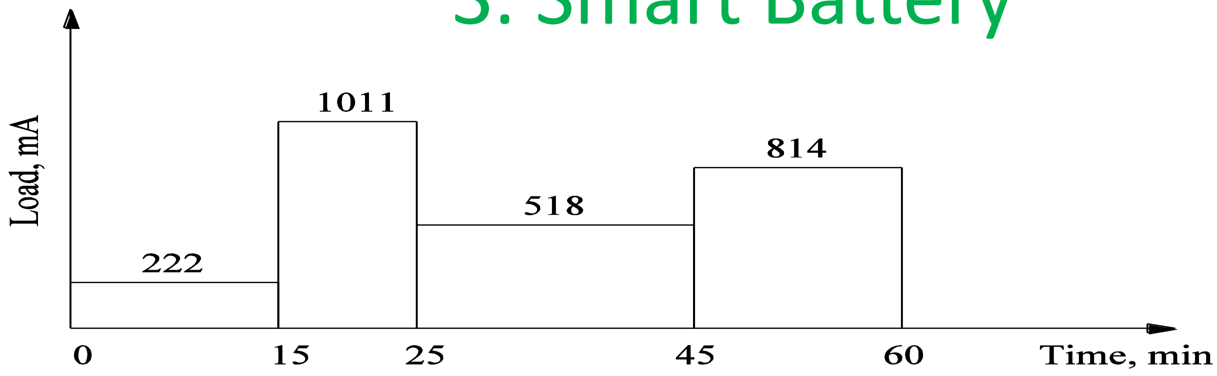
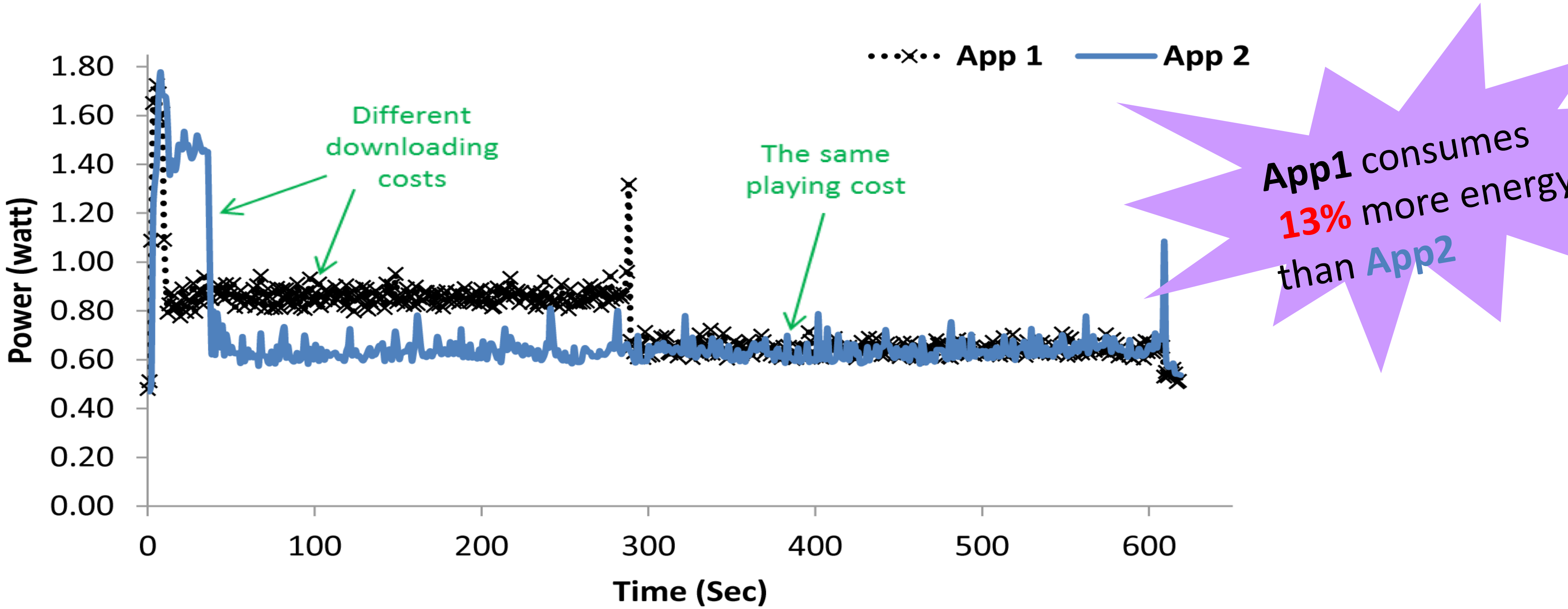
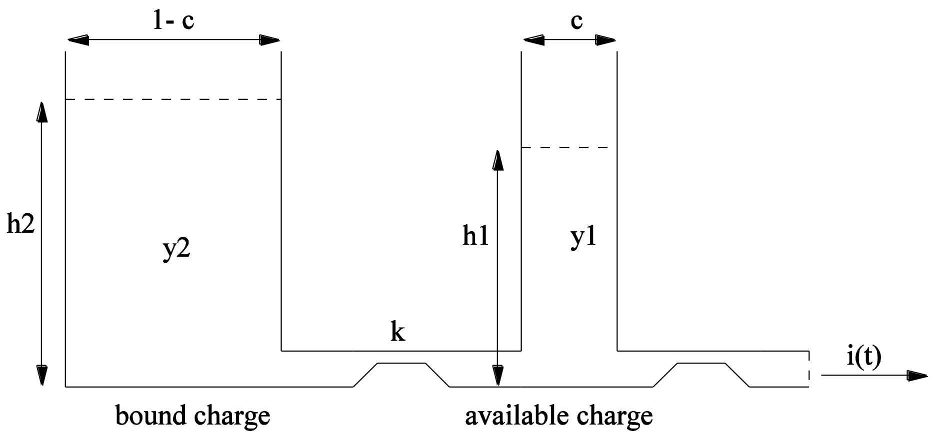


Figure 3.2: An example of load profile.

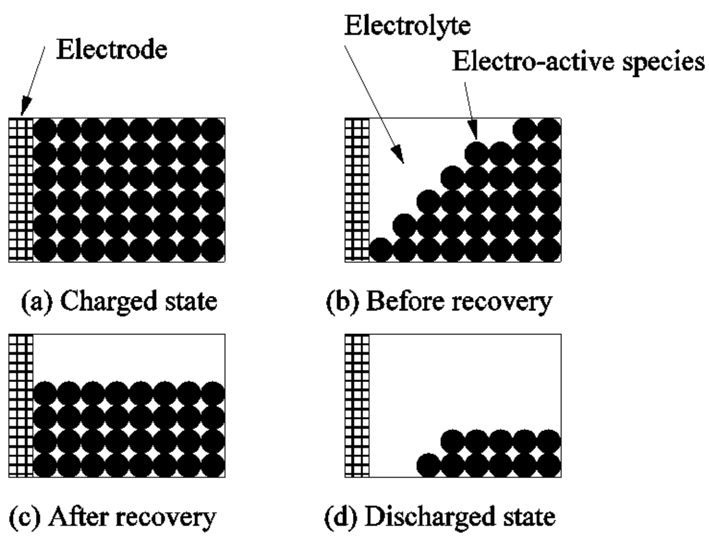


Power consumption of streaming a 10-minute video via App1 and App2

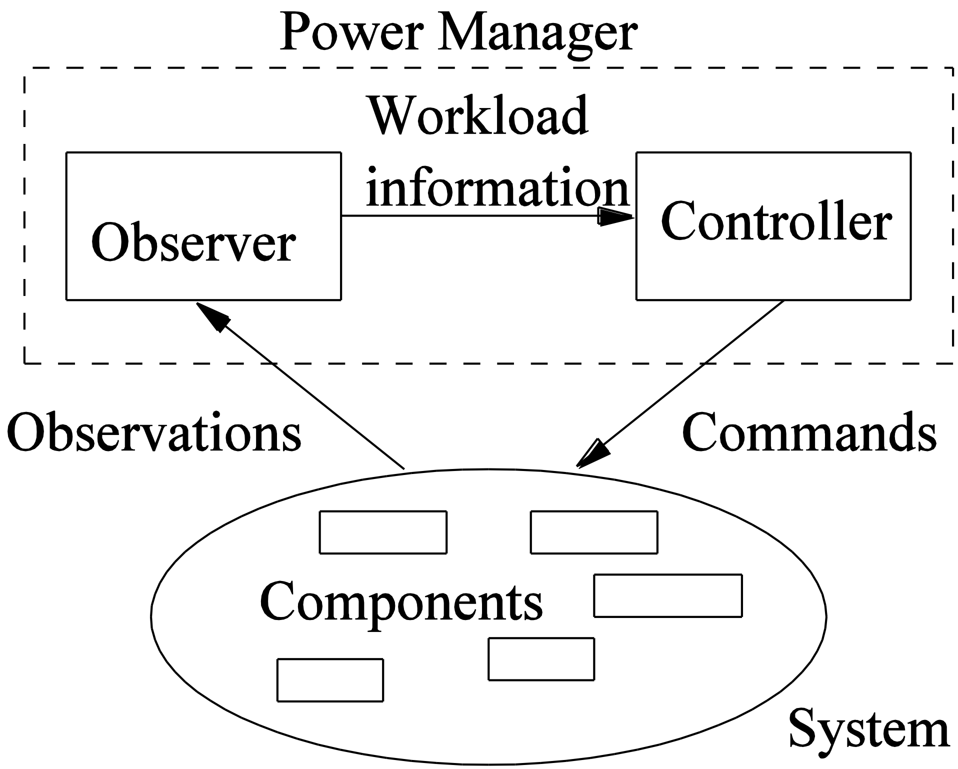
3. Smart Battery



Kinetic battery model



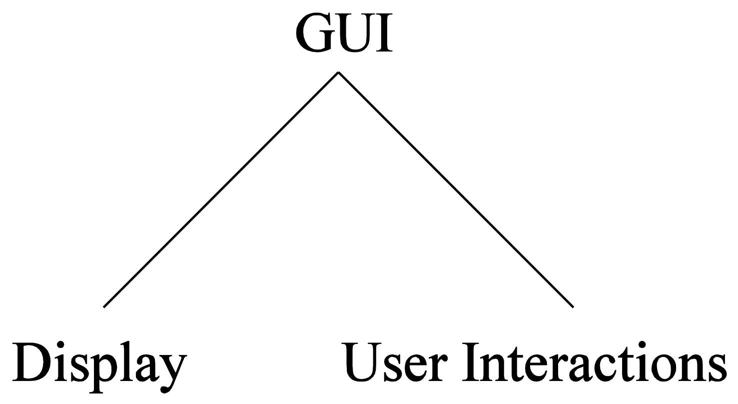
Diffusion model



An architecture of a power manager for smartphones.

4. Energy Efficient GUI Design

A GUI can be broken down into two conceptual components:



4. Energy Efficient GUI Design

- **Energy Efficient Interactions**
  - User interface code comprise a large fraction of the app code. (15 years back it was about 48%.)
  - Energy cost of GUI can be studied from three perspectives:
    - Hardware perspective
    - OS perspective
    - App perspective

4. Energy Efficient GUI Design

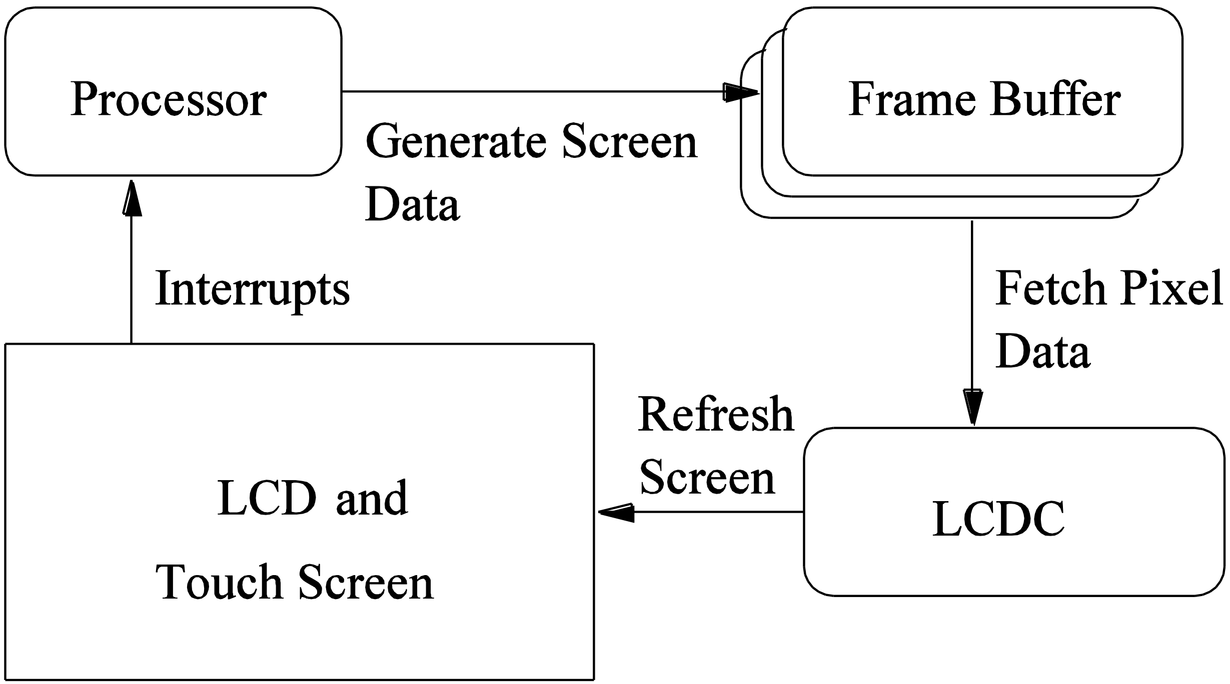


Fig. 4.2: A hardware perspective of GUI energy consumption.

4. Energy Efficient GUI Design

- Efficient Displays
  - For enjoyment and aesthetic use, a display must have enough resolution and color depth.
  - A TFT LCD (Thin-Film Transistor Liquid Crystal Display) is common in smartphones.
  - Components of an LCD display:
    - LCD panel
    - frame buffer memory
    - LCD controller
    - backlight inverter
    - lamp

4. Energy Efficient GUI Design

- Backlight control
  - Dynamic luminance scaling (DLS) technique keeps the perceived contrast of the image as close as possible to the original image while achieving 20-80% power saving from backlight system.
  - Reduced luminance degrades picture quality.
- Other energy saving approaches
  - “Dark window” optimization
  - Dynamic control of color depth and refresh cycle

4. Energy Efficient GUI Design

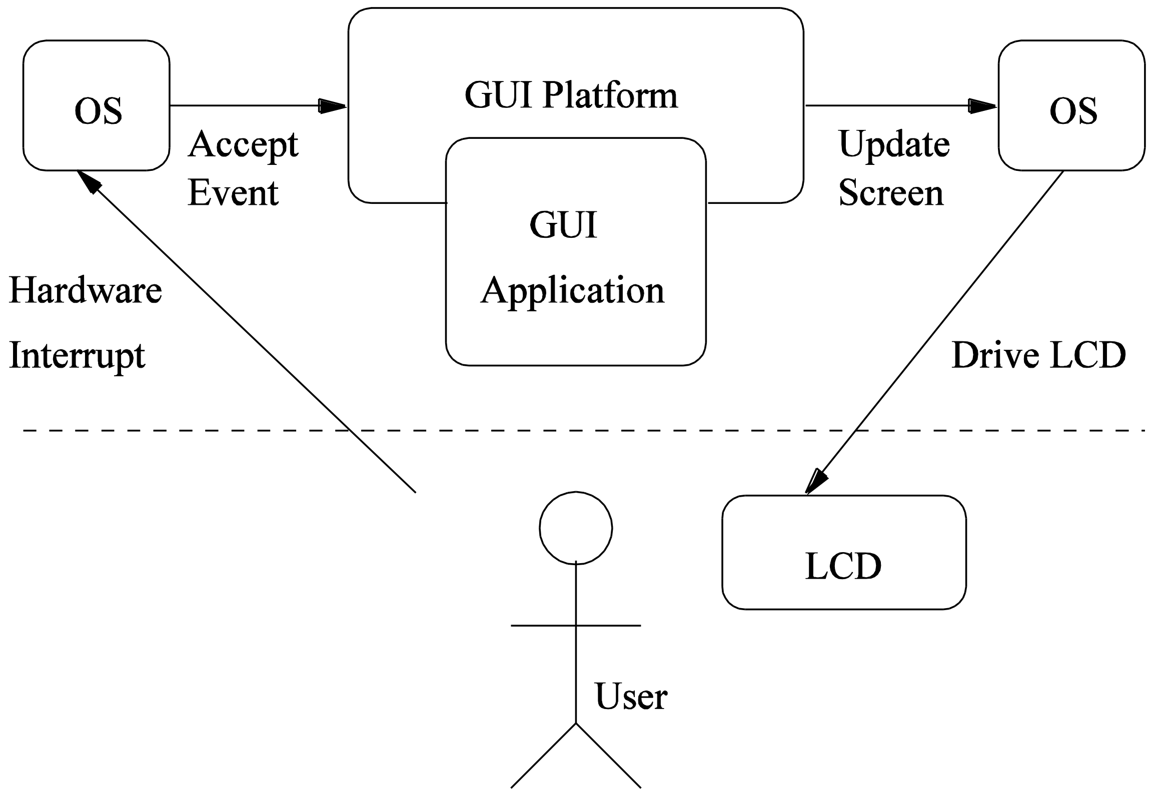


Fig. 4.3: An OS perspective of GUI energy consumption.

4. Energy Efficient GUI Design

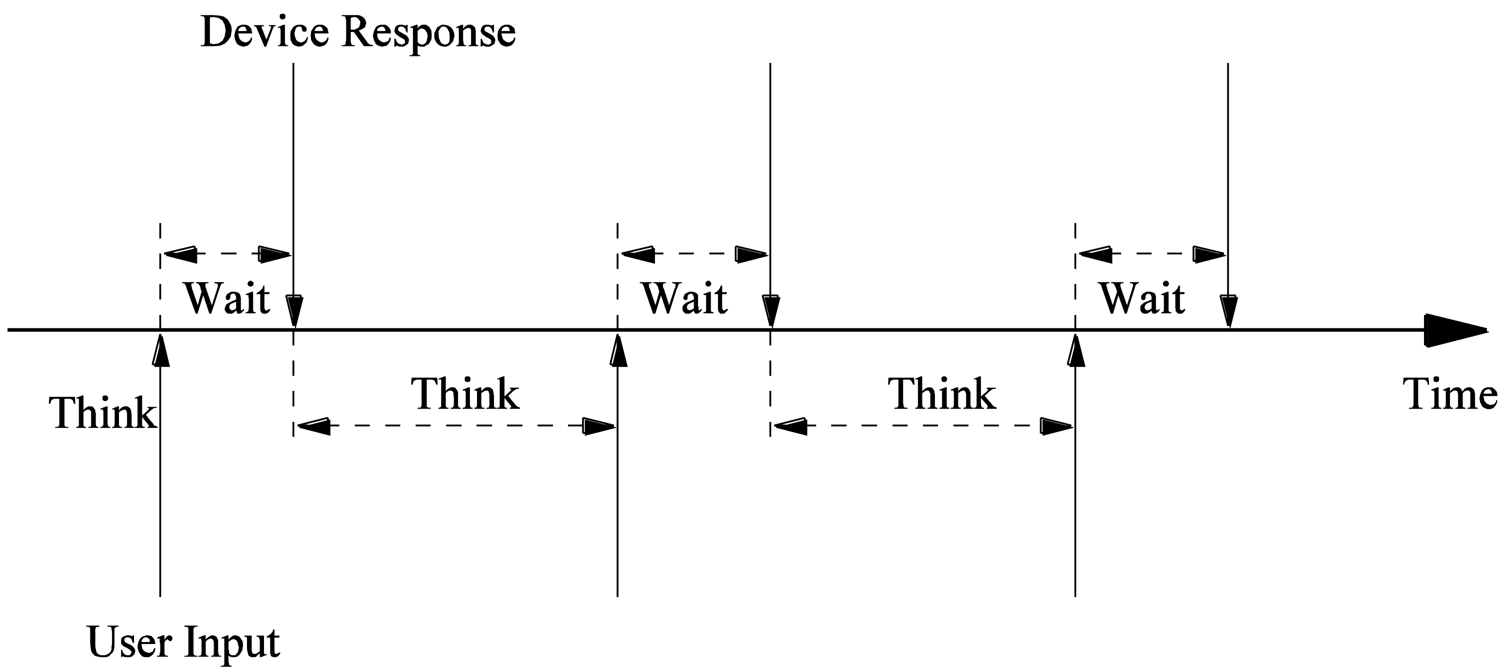


Fig. 4.2: An application perspective of GUI energy consumption.

4. Energy Efficient GUI Design

- **<u>Recommendations to make GUI design energy eff.</u>**
  - Accelerate user interactions
    - Programs and functions should be placed in the GUI so that users can find them quickly.
  - Do something while waiting for user input
    - Example: Speculate user input.
  - Minimize screen changes
    - Animation and window scrolling should be avoided.
  - Avoid or minimize text input
    - Text input is slower
    - Facilitate users to choose from lists

4. Energy Efficient GUI Design

- **Understanding Human-Computer Interactions**
  - If a device assists its user in producing a quick response, then the user's task will be completed quickly. ➔ less energy.
  - The response time of a user is influenced by three fundamental processes:
    - Perception capacity: How quickly one can **sense** (see/read) things
    - Cognitive speed: How quickly one can **react** to it
    - Motor speed: How quickly one can **move** on the display

3. Energy Efficient GUI Design

- Perception capacity
  - A better visibility of the material being read has a positive impact on reading speed.
  - Visibility depends on
    - Font type and size (*Smartphones*)
    - Color scheme
    - Contrast ratio
    - Luminance

4. Energy Efficient GUI Design

- Cognitive speed
  - If there are N distinct and equally possible choices, then the **reaction time** required to make a choice is given by the Hick-Hyman law as:
    - Reaction time = $\mathbf{a + b.\log_2 N}$ (a and b are constants.)
  - Interpretation of the law:
    - A GUI should present as few choices as possible.
    - The concept of **split menus** is useful in realizing this concept.

3. Energy Efficient GUI Design

- Motor speed
  - The motor speed, governed by the Fitt's Law, of human users positively impact their reaction time.
  - User interactions often involve **moving a control point** from one position to another and activating a button at the destination.
  - The time (T) taken to move from the current position to the destination is expressed as (from Fitt's Law):
    - **$T = c1 + c2.\log_2(D/W + 1)$** (c1 and c2 are constants, D is the distance between the two positions, and W is the width of the target).
  - Interpretation: A GUI should utilize as much screen area as available for widgets to be rapidly hit.

4. Energy Efficient GUI Design

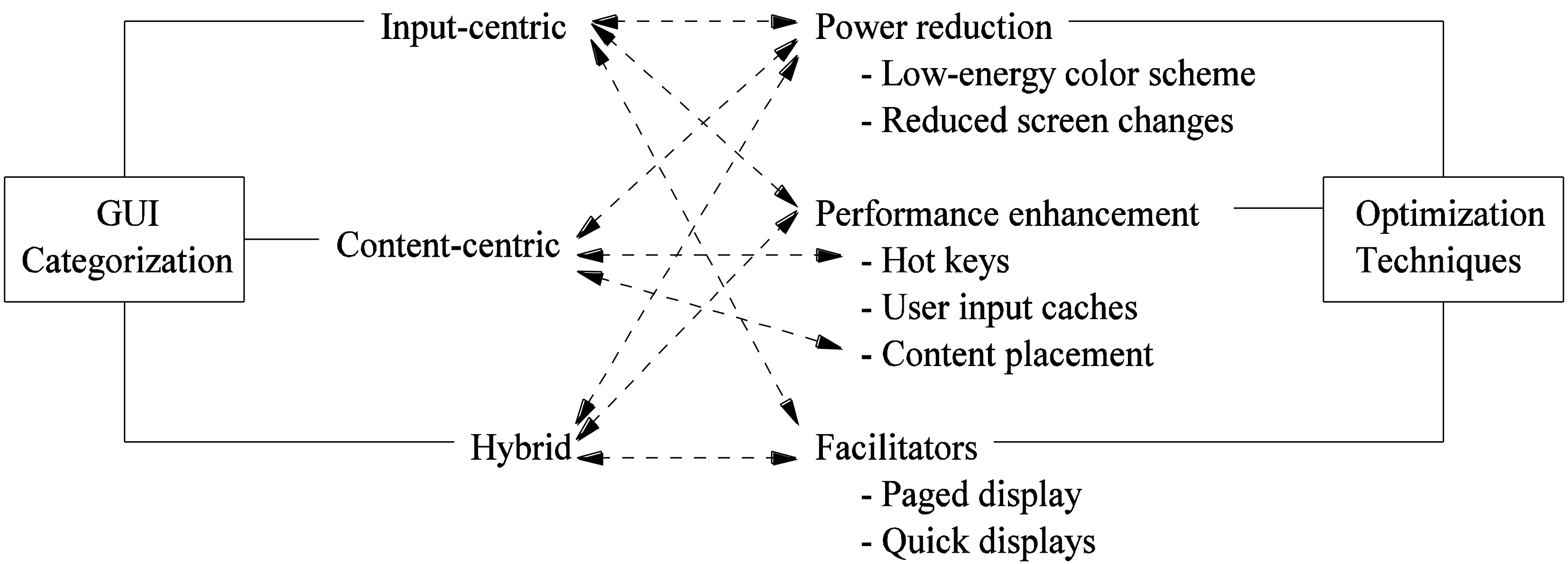
- **Techniques for Designing Energy Efficient GUIs**

- Before applying energy saving techniques, it is useful to classify GUIs based on their primary interactions:
  - **Input-centric**
    - **The main task is to obtain user inputs: messaging and calculator apps**
  - **Content-centric (output-centric)**
    - **Map viewers and browsers have content centric GUIs.**
  - **Hybrid**
    - **Text editors require significant input and display components**

4. Energy Efficient GUI Design

- **Specific techniques to reduce energy consumption by a GUI are divided into three categories**
  - Power reduction
    - Low energy color scheme and reduced screen changes
  - Performance enhancement
    - Hot keys (on laptops), user input caches, and content placement
      - Quick buttons are used instead of hot keys (key combo: alt + ctrl + I)
      - Auto-completion is an example of input cache
      - Content placement reduces perception, cognition, and motor latency.
  - Facilitators
    - Paged displays and quick buttons
      - Facilitators indirectly save energy

4. Energy Efficient GUI Design



Note: An association with Facilitators implies associations with all its components. Similarly, an association with Power reduction and Performance enhancement implies associations with all of their components.

Figure 4.5: GUI categories and energy saving techniques

5. Sleep to Save Energy

- Topics to cover
  - μSleep for OS
  - Standard IEEE 802.11 (WiFi) Power Saving Mode (PSM)
  - Proposed variations of PSM

5. Sleep to Save Energy

- µSleep for OS

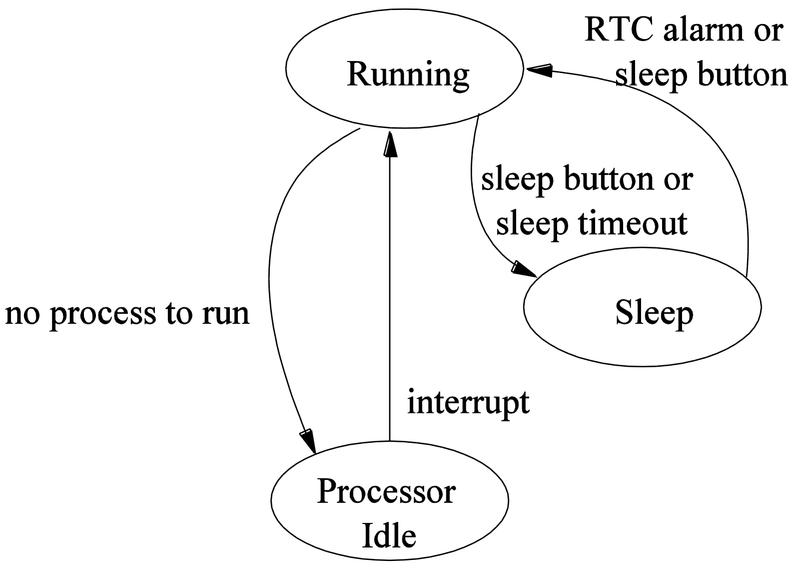


Figure 5.1: State diagram of a processor without PSM.

Power cost ratio:

Sleep: Processor Idle = 1: 6.8 – 1: 12.8

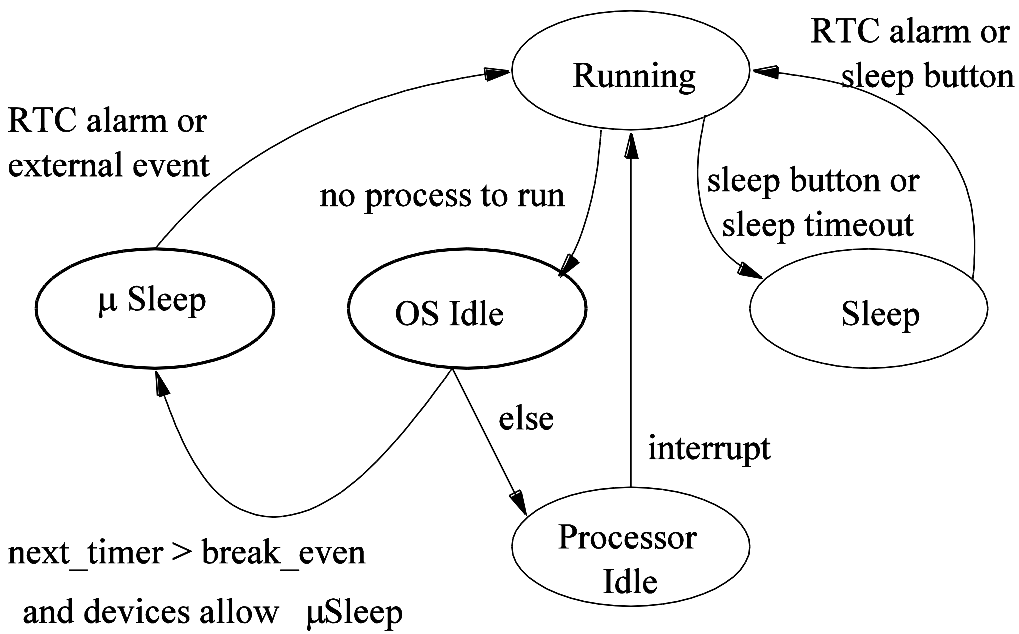
Running state: The OS has executable process, thread, or kernel code to execute.

Idle: The processor has nothing to execute.

Sleep: The processor is put into sleep by user or the code.

5. Sleep to Save Energy

- µSleep for OS



µSleep can educe energy cost by 60% when the Itsy pocket PC is lightly loaded.

Figure 6: State diagram of µSleep.

5. Sleep to Save Energy

- Standard IEEE 802.11 (WiFi) Power Saving Mode (PSM)
  - WiFi protocol has a built-in mechanism to let user devices perform dynamic power management of their WNIC.
  - A WNIC has five major states:
    - Power-off
    - Idle (Ready to transmit and receive)
    - Transmit (Tx)
    - Receive (Rx)
    - Sleep
  - P(Power-off) < P(Sleep) < P(Idle) < P(Rx) < P(Tx)
  - Power saving is achieved by utilizing the central controller mode of the AP (Access Point)

5. Sleep to Save Energy

- Standard IEEE 802.11 (WiFi) Power Saving Mode (PSM)
  - In the PCF mode, each user device informs the AP if the device is utilizing the PSM.
  - The general behaviors of AP and device are as follows:
  - AP:
    - While a user device is sleeping, packets for device are buffered.
    - The AP periodically (100 ms) broadcasts *Beacon* frames.
    - *Beacons* contain a traffic indication map (TIM) that indicates PSM devices with at least one data packet buffered at the AP.
    - The AP sends data to a user after the device makes a request by means of a *PS-Poll* frame.

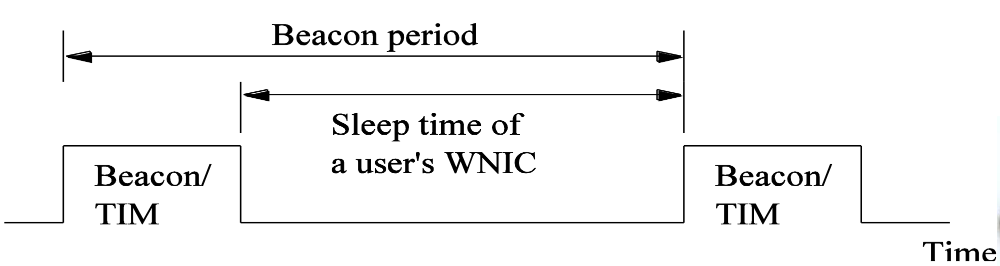
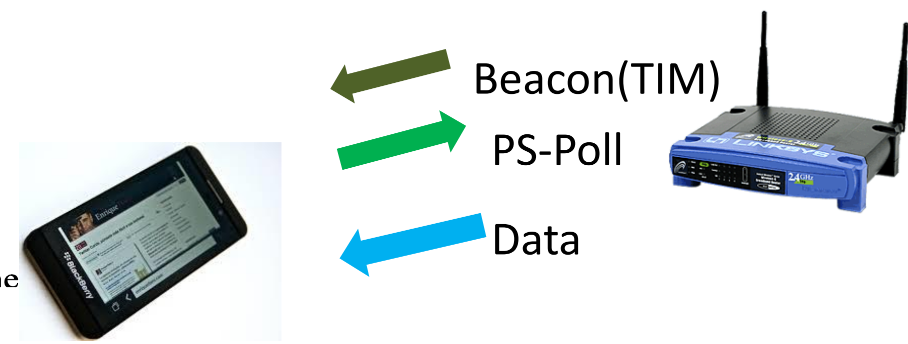
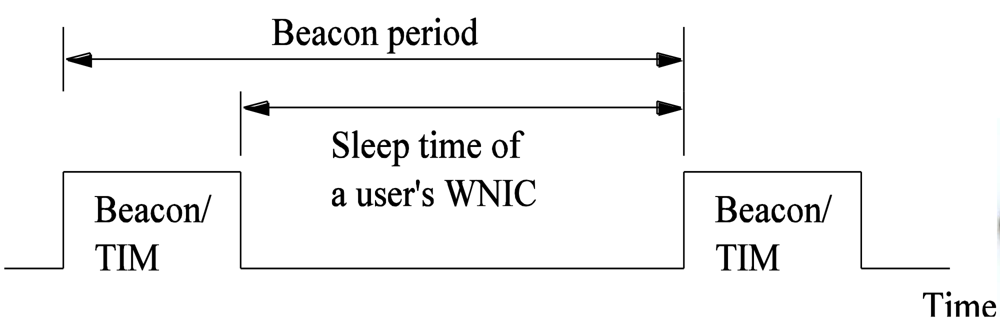


Fig. 5.4: General idea of Beacon period



4. Sleep to Save Energy

- Standard IEEE 802.11 (WiFi) Power Saving Mode (PSM)
  - User's Device:
    - If the PSM feature of the card is enabled, the card sleeps for a fixed period upon the elapse of a fixed timeout since the last received packet.
    - Upon expiration of a sleep period, it wakes up to listen for a Beacon.
    - If the user device is indicated in the TIM of a Beacon, the PSM user device sends a PS-Poll frame (via DCF mode) to the AP to fetch data.



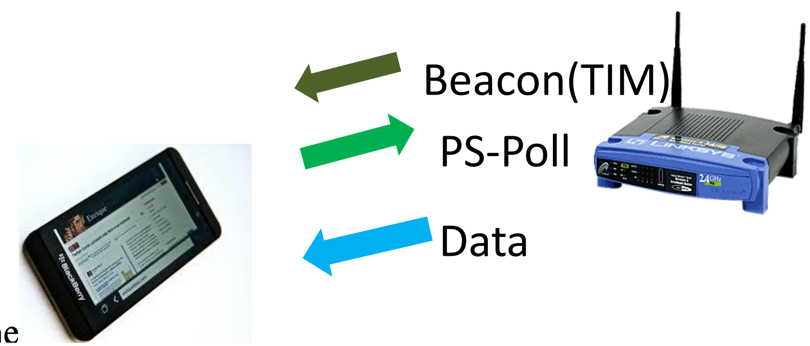
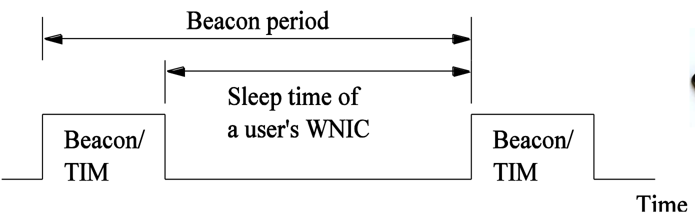
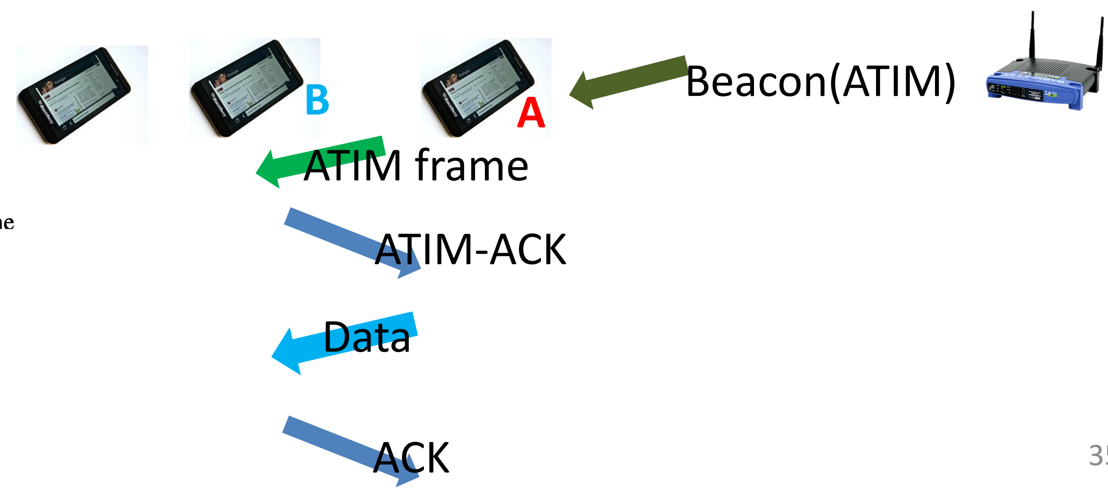


Fig. 5.4: General idea of Beacon period

5. Sleep to Save Energy

- Power saving feature of DCF mode of WiFi
  - At the start of each Beacon interval, devices remain awake for a fixed time interval, called Ad-hoc TIM (ATIM) window.
  - A device (A) advertises data frames pending Tx to others (B).
    - The advertisement is done in the form of an ATIM frame transmitted during ATIM window via the CSMA/CA mechanism.
    - The receiver (B) of an ATIM frame responds with ATIM-ACK and stays awake for the remaining period of the ATIM window. (Other devices simply sleep for the remains of the Beacon interval.)
    - If A receives the ATIM-ACK, it stays awake for the remains of Beacon int.
    - **A sends data** to B during the Beacon interval, after the ATIM window.

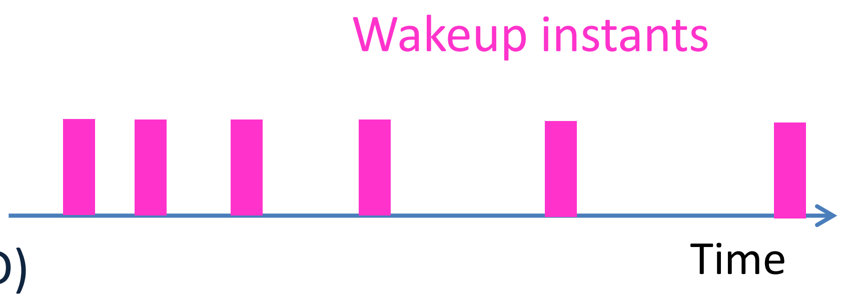




5. Sleep to Save Energy

- Standard IEEE 802.11 (WiFi) Power Saving Mode (PSM)
  - The technique does not completely switch off the WNIC card.
  - The sleep duration is fixed, and it does not reflect variability in received data.
  - The max sleep time usually supported is less than one second, to reduce the risk of AP buffer saturation.

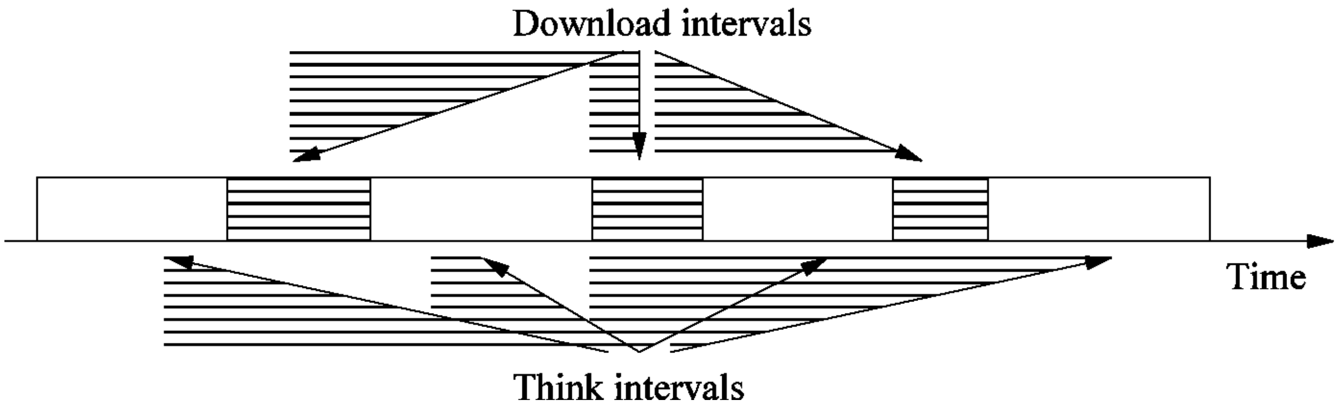
5. Sleep to Save Energy



- Variations of PSM proposed by researchers
  - Krashinsky and Balakrishnan
    - Proposed a Bounded Slowdown (BSD) protocol in which a user device listens for *Beacons* with decreased frequency during idle times.
    - The wake up time is statically determined.
    - The protocol trades off energy for additional delay.
    - If the max tolerable delay is small, it incurs more energy cost than PSM.
  - Qiao and Shin
    - Enhanced BSD – call it Smart PSM (SPSM)
    - They dynamically estimate the time instants when the user device wakes up to listen for Beacons.

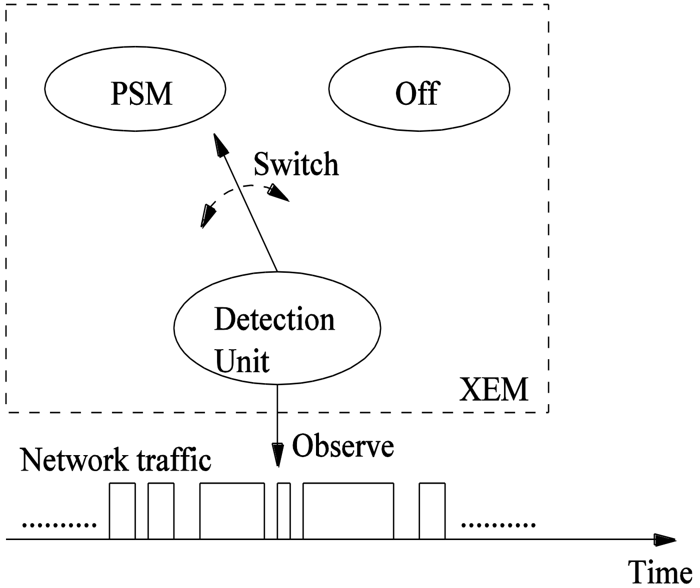
5. Sleep to Save Energy

- Variations of PSM proposed by researchers
  - Nath, Anderson and Seshan
    - Attempt to reduce delay by letting individual devices select their own Beacon interval and having the AP generate separate Beacons for each device.
  - Anand et al.
    - Propose a Self-tuning Power Management (STPM) scheme to exploit ***hints*** provided by network applications.
    - The hints describe the near-future network activities of applications.
    - One can view such hints as knowledge of the download times and think times.
    - Hints are used to manage the states of the WNIC.



5. Sleep to Save Energy

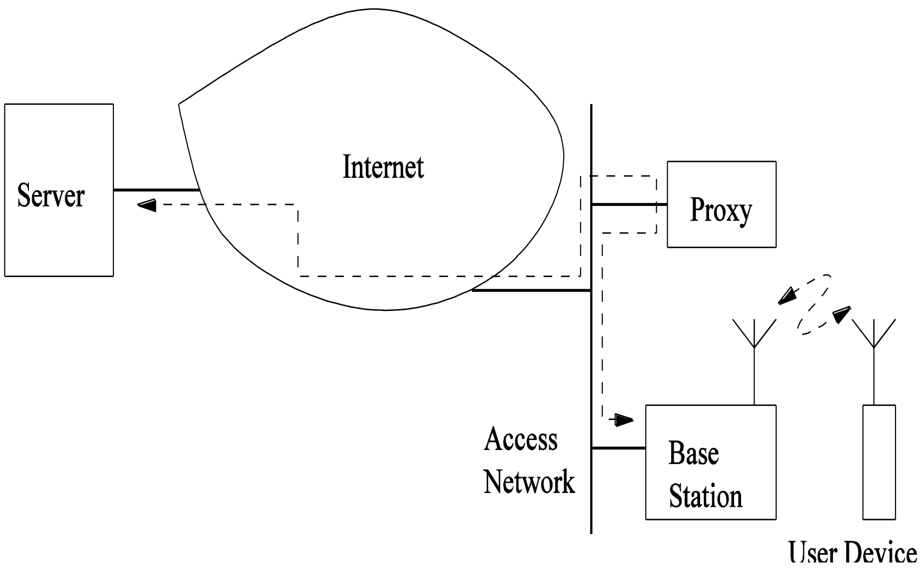
- Variations of PSM ….
  - Anastasi, Conti, Gregori ….
    - Propose a Cross –layer Energy Manager (XEM) that dynamically tunes its strategy based on app layer behavior and network parameters.
    - **Agent based:** An agent spoofs the web traffic generated by the user. For each web page, the agent knows the files to be downloaded. Once all the files are downloaded, a "think" interval is assumed to start so switch to off state.
    - **Timer based:** If no data is received within one TCP RTT (round-trip time), it is assumed that a "think" interval has started.
    - An app level activity switches the WNIC to the PSM mode.
    - 20-96% saving w.r.t. PSM.



PSM: Power Saving Mode
XEM: Cross-layer Energy Manager

6. Proxy Assisted Energy Saving in Handheld Devices

- Proxy
  - It is a request and content processing machine appearing between the server and the base station (aka AP: Access Point).
  - It is generally located close to the Base Station.
  - User devices send requests to the local proxy.



6. Proxy Assisted Energy Saving in Handheld Devices

- **Power Aware Web Proxy**
- Motivation: In PSM, if data packets arrive at an AP with **arbitrary time gaps**, a user device cannot take full advantage of PSM. PAWP allows a device to sleep longer.
- PAWP splits up a client/server TCP connection, buffers web pages requested by the device, and aggressively prefetches all the embedded objects in the pages.
- Result: Data traffic between device and AP becomes bursty and idle periods become longer.
- Note: The proxy works at the HTTP level to exploit app level information.

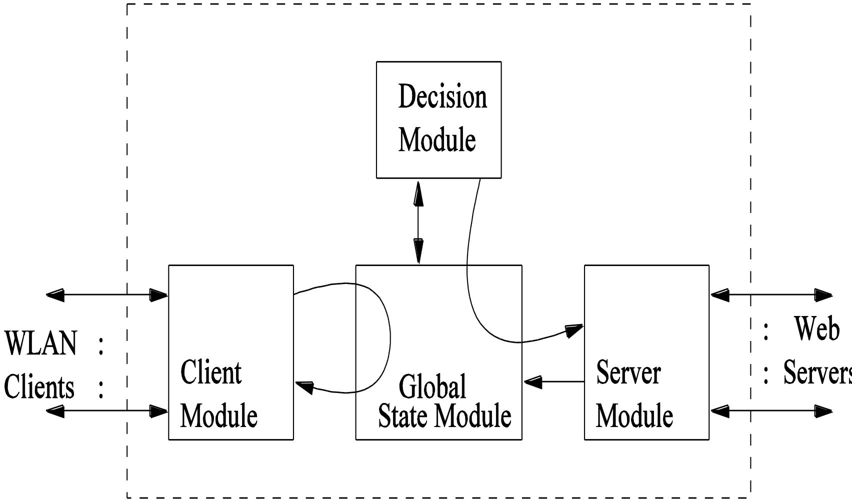
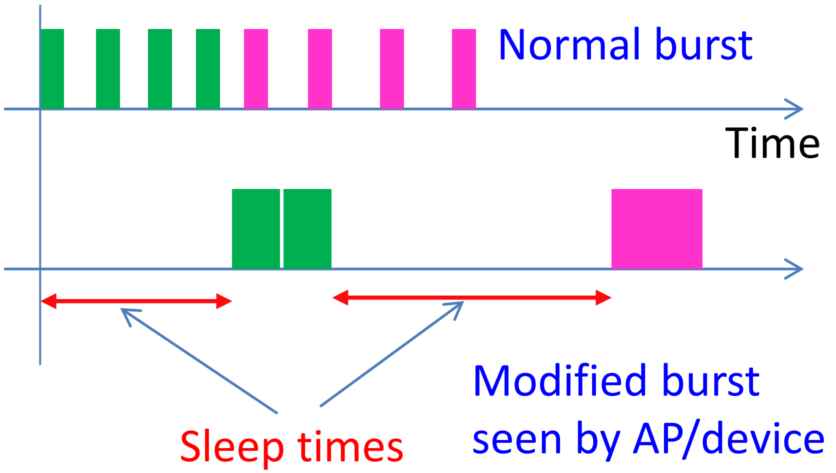


Fig. 6.2: Architecture of a Power Aware Web Proxy.



6. Proxy Assisted Energy Saving in Handheld Devices

- **Power Aware Streaming Proxy**
- **Note:** PAWP **shapes traffic** between AP and device to be bursty, to make the device idle time longer.
- PASP: **Transforms** the server-to-client media stream to adapt it to client capabilities: WLAN bandwidth, screen resolution, computing resources, and battery energy left.
- PASP intercepts and alters the client-to-server stream tunneled on the POST connection.

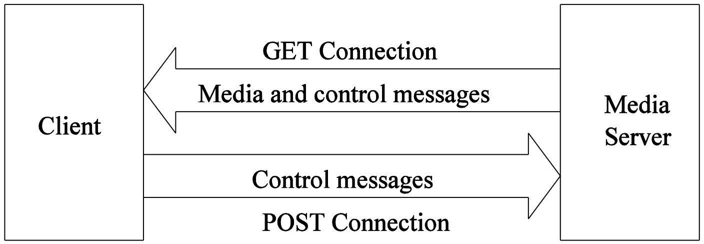


Fig. 6.3: HTTP Connections used for Tunneling. Two TCP conns are used.

6. Proxy Assisted Energy Saving in Handheld Devices

- **Power Aware Streaming Proxy**

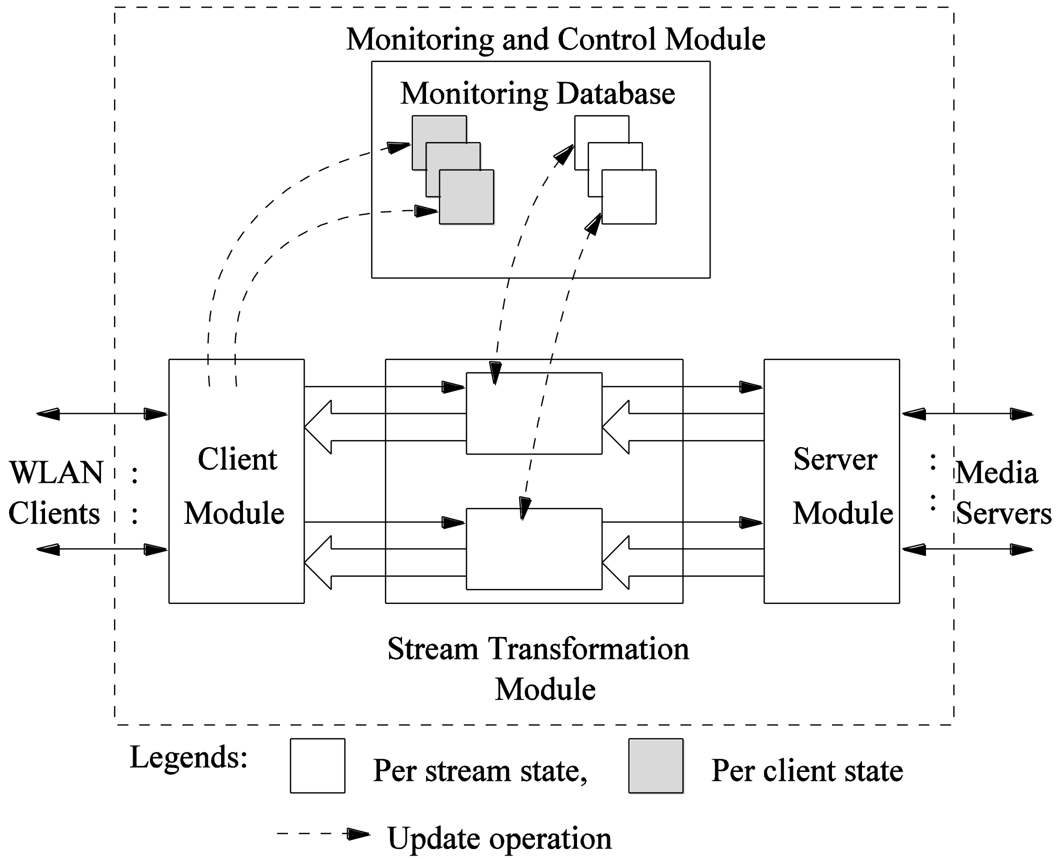


Fig. 6.4: PASP Architecture.

Stream transformation module:

It forwards only the video layer that is most appropriate for the client.

It selectively drops some video layers.

6. Proxy Assisted Energy Saving in Handheld Devices

- **Streaming Audio Proxy**
  - Developed by Anastasi et al. to minimize the energy consumption of WiFi interface of a mobile device.
  - The proxy service runs on AP.
  - The data path between the streaming server and the mobile device is split at the proxy on the AP.
  - Real-time data and control information are handled with two protocols:
    - Data with Real Time Protocol (RTP)
    - Control information with Real Time Streaming Protocol (RTSP)
    - Feedback about quality of data with Real-time Transp. Cont. Pro. (RTCP)
  - The TCP Friendly Rate Control (TFRC) protocol is used to control network congestion, because the underlying protocol is UDP.
  - RTP and RTSP carry TFRC information.
  - The Real Time Power Saving (RT_PS) protocol running on top of UDP and between the device and the AP is key to energy saving.

6. Proxy Assisted Energy Saving in Handheld Devices

- **Streaming Audio Proxy**

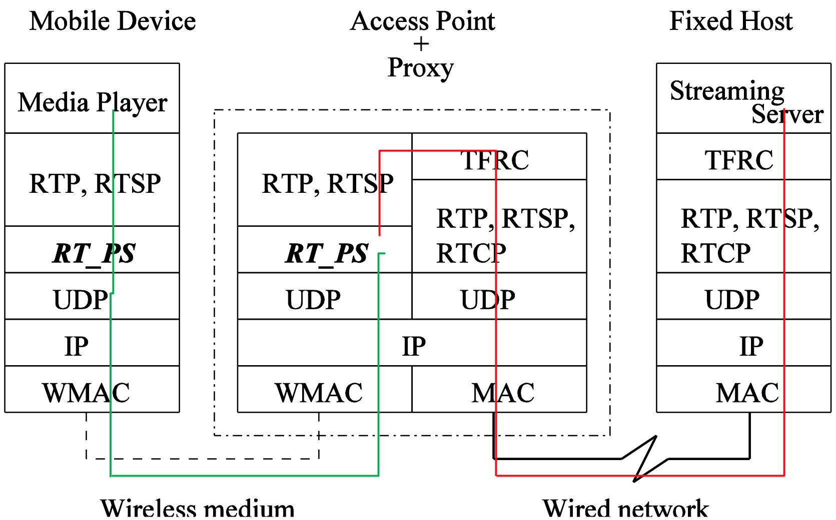


Fig. 6.5: Streaming Proxy Architecture

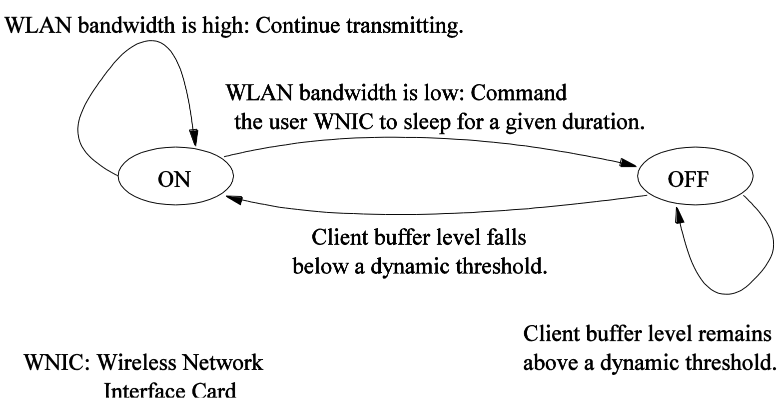


Fig. 6.6: The ON/OFF Model of the Streaming Proxy Architecture

Proxy component of RT_PS

- Calculate the init. playback delay.
- Keep track of WLAN BW and level of user buffer.
- Keep transmitting according to Fig. 6.6.

Client component of RT_PS

- Request the proxy for an audio file.
- Let the proxy know the buffer size.
- Start playback after an initial delay.
- Put the WNIC to **sleep mode** when asked to do so by the proxy.

6. Proxy Assisted Energy Saving in Handheld Devices

- **Streaming Audio Proxy**
  - They implemented the proxy to conduct tests.
  - The technique allows energy saving from 76-91% of the total consumption due to the network interface.
  - Some packets arrived late so those were discarded. Some were lost during transmission.
  - The fraction of packets discarded or lost was less than 5%.

7. Source Level Power Control

- **Focus:** Energy saving strategies applied at the media sources on app servers.

- Strategies
  - Remote Power Control from Servers
  - Application Level Scheduling by Media Servers

7. Source Level Power Control

- **Remote Power Control from Servers**
  - Proposed by Acquaviva, Simunic, Deolalikar, and Roy
  - It is a proxyless approach to enabling a device to move its WNIC card to sleep state.
  - They exploit the **server knowledge of the client workload**, **traffic conditions**, and **feedback information from client** to perform traffic shaping over the wireless link so that the client can put its WNIC to sleep.

7. Source Level Power Control

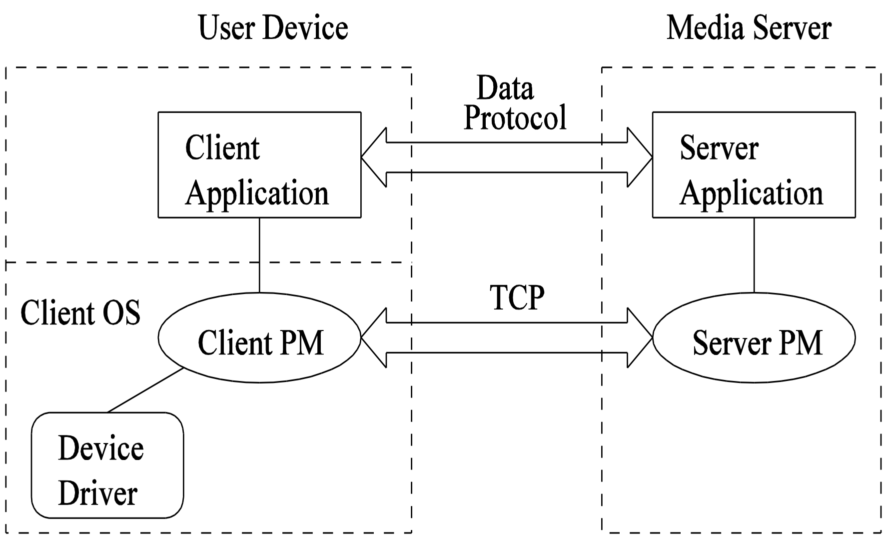


Fig. 7.1: Server Controlled Power Management (PM: Power Manager)

- Server PM and Client PM communicate over 2 channels: Control and Data.
- Server PM sends the following commands to Client PM:
  - Switch off WLAN
  - Set the off time
  - Enable the 802.11 power policy
  - Set the 802.11 power management parameters
    - Period between wake ups
    - Timeout before going to doze mode

7. Source Level Power Control

- Server PM
  - Client adaptation: Get
    - Input buffer size
    - Expected value and variance of service rate (i.e. the rate at which the app empties the buffer)
    - WNIC on/off transition time
    - WNIC card status
  - Traffic adaptation
    - Monitors traffic conditions
    - Decides when to enable PSM (Light traffic → PSM, not off)
- Server PM (contd.)
  - Traffic shaping
    - Schedule Tx in bursts
    - Burst size and delay are precomputed
    - Delay: large enough to almost empty the client buffer
    - Burst size: avoid overflow of client buffer

7. Source Level Power Control

- Client PM
  - Server Interface
    - The Client PM obtains the buffer size, depletion rate, and backlog level from app and forwards them to Server PM. The device drivers report the on/off transition time.
  - Device Interface
    - The Client PM responds to commands from Server PM:
      - Change parameters of the 802.11 power manager, switch WLAN on/off, read interface stat (signal/noise ratio)

- Client PM (contd.)

Delay between bursts,

$D_{burst} = frame\_time(\sum_{i=1}^{M} n_i / i)$

$frame\_time$ is the play time interval between successive frames.

n_i is the # of frames consisting of i packets.

M is the max # of packets per frame.

7. Source Level Power Control

- Evaluation
  - MPEG4 video data was transmitted from server to client.
  - SmartBadge IV was used as a client device.
  - Data Protocol
    - RTSP for session initiation/ termination
    - RTP and UDP to carry data
  - 2 kinds of benchmarks used
    - Benchmark 1: 12 bursts, 12 FPS
    - Benchmark 2: 402 bursts, 30 FPS

      (FPS: Frames per second)

- Evaluation (Contd.)
  - Benchmark 1 yielded a delay of 4 seconds between successive bursts
  - 67% energy saving compared to "always on" WNIC.
  - 50% energy saving compared to the built-in PSM

7. Source Level Power Control

- **Application Level Scheduling by Media Servers**
  - Proposed by Acquaviva, Lattanzi and Bogliolo
  - Two scheduling policies to exploit the information available at the app level to save energy
    - Closed-loop DPM (Dynamic Power Management)
    - Open-loop DPM

7. Source Level Power Control

- **Application Level Scheduling by Media Servers**

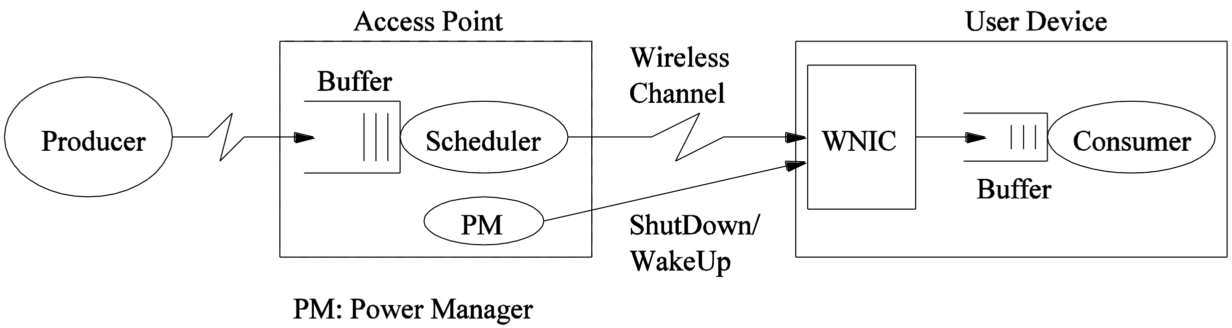


Fig. 7.2: Source-level Scheduling of Streaming Data in WLANs.

Fig. 7.3: Closed-loop Scheduling (to follow)

7. Source Level Power Control

- Elements of the system
  - Producer
    - For each packet, 3 attributes are generated: packet size in bytes, the frame it belongs to, total # of packets belonging to the same frame. Packets in the same frame are put in the same burst.
  - AP
    - Has a buffer with a tunable size and a packet scheduler.
    - If the user device is in PSM, no packets are sent out to the device.
  - Wireless Channel
    - It is represented by channel latency, loss probability, and bandwidth.
  - WNIC
    - It is assumed to have two modes: always-on and power-save protocol (PSP)
  - Power Manager (PM)
    - This generates ShutDown and WakeUp events to **notify** the start and end of sleep periods.
  - User Device
    - It has two main elements: buffer (protocol or app buffer) and consumer.
    - Consumer decides how many packets to read within a frame period.

      Late and incomplete frames are discarded.

7. Source Level Power Control

- Closed-loop DPM
  - The client buffer is monitored wrt HWM and LWM
  - HWM reached
    - Tell server to stop sending.
    - Start a timer for the last packet to receive.
    - Timeout ➔ Shutdown WNIC.
  - LWM reached
    - Client tells server to resume Tx.

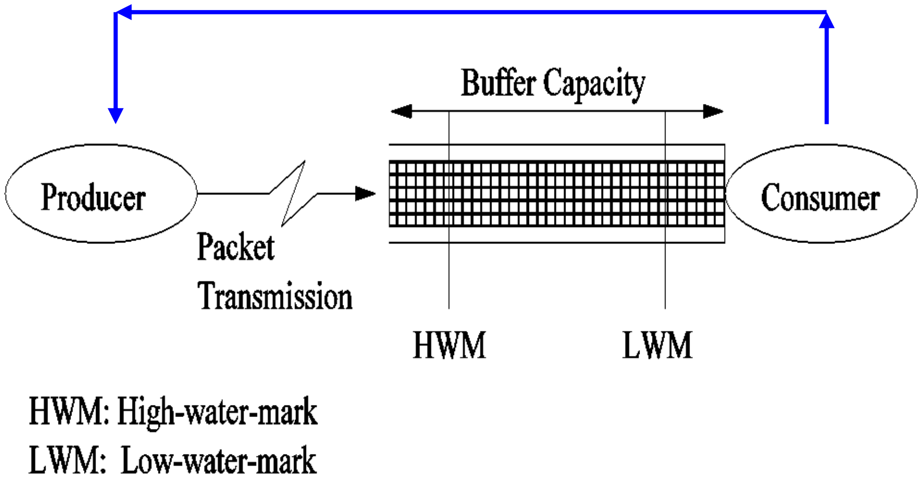


Fig. 7.3: Closed-loop Scheduling

- LWM = (Too + Tc) λs
- HWM = BS(Tm + Tc)(λa - λs)

Closed loop: The client provides feedback Information to the server: LWM and HWM.

7. Source Level Power Control

- Too: Off/on wakeup time for the WNIC.
- Tc: Cushion time used to avoid that the buffer empties completely due to network uncertainties.
- λs: Average packet consumption rate

  = frame_rate x average frame size.
- BS: Client buffer size.
- Tm: Time needed to send a HWM message to source.
- λa: Arrival rate of packet.

7. Source Level Power Control

- Open-loop DPM
  - Client does NOT provide feedback info. to the server.
  - Server predicts when the client buffer will be empty by exploiting its knowledge of workload.
  - Server schedules Tx in bursts to create WNIC idle periods long enough to exploit the radio-off state of WNIC.
  - **Client switches** off WNIC after receiving a burst.
  - Server estimates burst size and idle time (**D_{client}**) and sends this info to client in a control pkt.

- Burst size = BS – Nc

Nc is a constant to avoid buffer overflow.

- $D_{client} = T_{cons} - Too$

T_{cons} is the time needed to consume the burst.

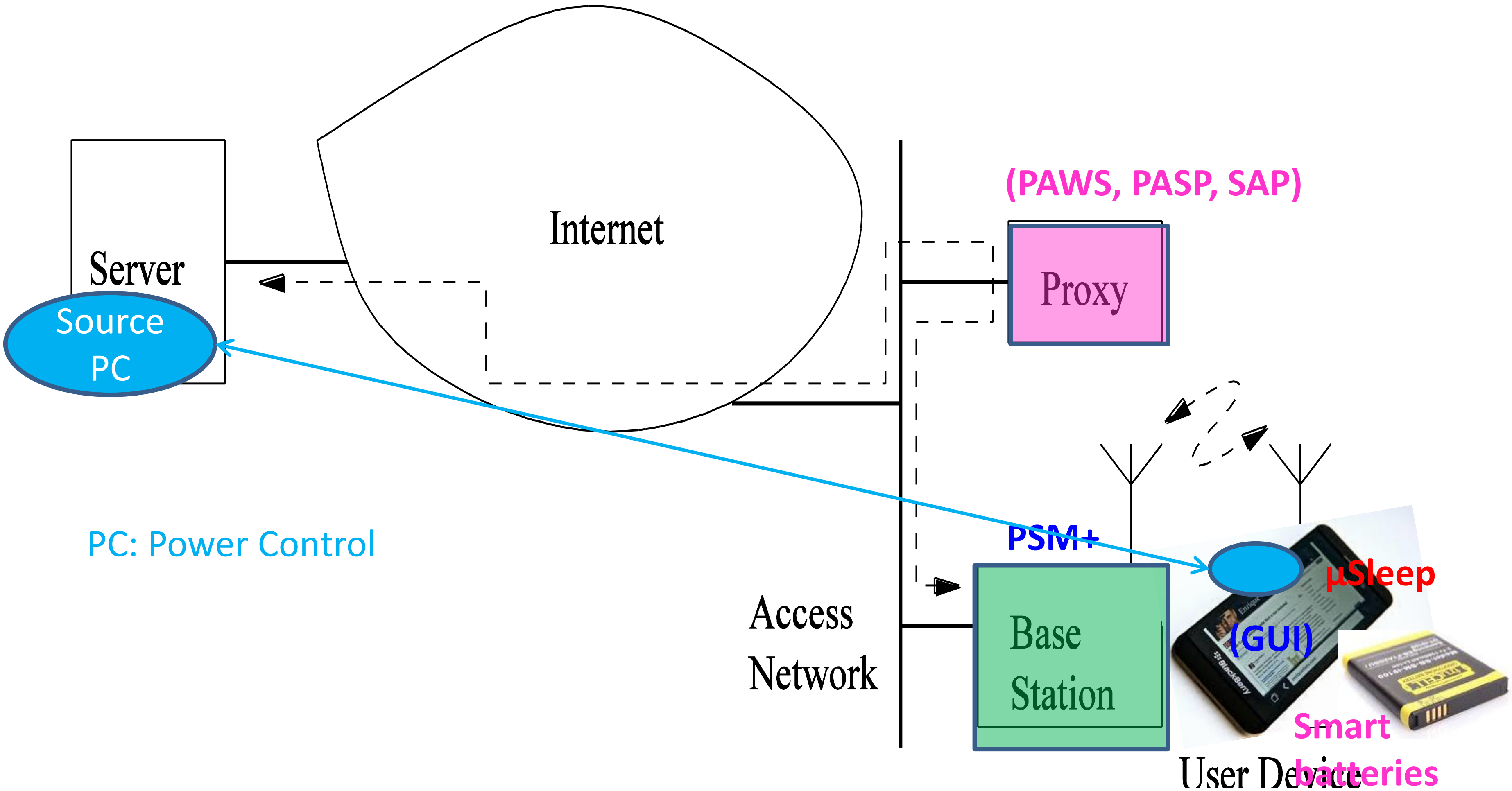
Summary

- Smart battery: OS and app know the remaining SoC
- GUI: Design can lead to energy efficient
- µSleep state for OS
- Standard power saving mode (PSM) in IEEE 802.11/WiFi and its variations
- Proxy assisted power saving
  - PAWP: Split TCP conn, to make traffic bursty at the client
  - PASP: Adapt contents based on device resources
  - SAP (Streaming Audio Proxy): Runs on AP. Asks device to sleep.
- Source Level Power Control

  -- Remote Power Control from Servers: Server PM and Client PM

  -- Application Level Scheduling by Media Servers: Closed loop/ Open loop DPM

Summary


Internet
Server
Source PC
(PAWS, PASP, SAP)
Proxy
PSM+
μSleep
(GUI)
Access Network
Base Station
Smart batteries
User Device
PC: Power Control