Network, I/O and Peripherals: Device-Specific Power Management

Selected Chapters of System Software Engineering: Energy-Aware System Software

Timo Hönig, Christopher Eibel

Department of Computer Science 4
Distributed Systems and Operating Systems

Friedrich-Alexander University
Erlangen-Nuremberg

19. Juli 2013

http://www4.cs.fau.de/Lehre/SS13/MS_AKSS/

Organisatorisches, Noten

- Seminar{termin,raum,themen}
  - Donnerstag, 17:30 (c. t.) – 19:00 Uhr
  - Raum 0.035-113
  - Themen:
    `http://www4.cs.fau.de/Lehre/SS13/MS_AKSS/`
- Organisatorisches
  - LaTeX-Vorlagen für Ausarbeitung und Präsentation bekommt ihr vom jeweiligen Betreuer (per E-Mail)
  - Abgabetermine bitte *selbstständig* einhalten
- Zusammensetzung der Noten
  - Vortrag (35 %)
  - Ausarbeitung (35 %)
  - Arbeitsweise (30 %)
    Aktive Teilnahme, Diskussionsbeiträge, Vorbereitung von Vortrag und Ausarbeitung

Demystifying 802.11n Power Consumption: Overview

- Paper
  - ,,*Demystifying 802.11n Power Consumption*"
  - Workshop on Power-Aware Computing and Systems 2010 (HotPower'10)

  → co-located with USENIX Symposium on Operating Systems Design and Implementation (OSDI'10)

- Authors
  - University of Washington (2x)
  - Intel Labs Seattle (2x)

  → joint work between academia and industry

  → often implies practical work

- Overview
  - 802.11n WiFi (,,Draft N")
  - **Measurement** paper

Demystifying 802.11n Power Consumption: Abstract

Abstract. We report what we believe to be the first measurements of the power consumption of an 802.11n NIC across a broad set of operating states (channel width, transmit power, rates, antennas, MIMO streams, sleep, and active modes). We find the popular practice of racing to sleep (by sending data at the highest possible rate) to be a useful heuristic to save energy, but that it does not always hold. We contribute three other useful heuristics: wide channels are an energy-efficient way to increase rates; multiple RF chains are more energy-efficient only when the channel is good enough to support the highest MIMO rates; and single antenna operation is always most energy-efficient for short packets.

Demystifying 802.11n Power Consumption: Abstract

Abstract. We report what we believe to be the first measurements of the power consumption of an 802.11n NIC across a broad set of operating states (channel width, transmit power, rates, antennas, MIMO streams, sleep, and active modes). We find the popular practice of racing to sleep (by sending data at the highest possible rate) to be a useful heuristic to save energy, but that it does not always hold. We contribute three other useful heuristics: wide channels are an energy-efficient way to increase rates; multiple RF chains are more energy-efficient only when the channel is good enough to support the highest MIMO rates; and single antenna operation is always most energy-efficient for short packets.

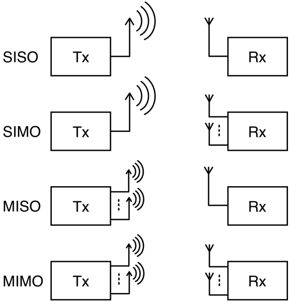
Demystifying 802.11n Power Consumption: Paper Details

- Paper contributions
  1. Energy measurements of 802.11 NICs
  2. Disprove today's best practice (partially)
  3. Suggest new approaches
- Paper structure
  - Motivation
  - Background on 802.11n
  - Measurements
  - Racing to Sleep
  - New Heuristics
- Remarks
  - No related work section, partially merged into first section (Introduction)
  - Possible follow-up conference paper:

D. Halperin, W. Hu, A. Sheth, D. Wetherall
Predictable 802.11 packet delivery from wireless channel measurements
ACM Special Interest Group on Data Communication (SIGCOMM'10), 2010.

Demystifying 802.11n Power Consumption: Motivation

- Up to 50% power consumption caused by WiFi
- 802.11n radio: 2.1 Watt (multiple-input and multiple-output, MIMO)



- Changes 802.11a/b/g → 802.11n: rates, antennas, channel width

→ Software designers need assistance to efficiently use 802.11 radios

Demystifying 802.11n Power Consumption: Strategy

- Strategy: Race to sleep vs. Shannon capacity

- **Race to sleep**: transmit at highest bit rate possible
  - Transmit all pending data as quick as possible → requires high bit rate
  - Pro: sleep for a longer period of time
  - Contra: consume a lot of energy during high bit rate transmission

- **Shannon capacity**: energy consumption per bit grows with bit rate
  - Transmit all pending data at a low speed → requires low bit rate
  - Pro: Low power consumption during transmission time
  - Contra: No idle time to enter sleep states

Demystifying 802.11n Power Consumption: Measurements

- Evaluation Setup
  - Intel WiFi Link 5300 a/b/g/**n**
  - 3x3 MIMO (3x TX, 3x RX)
  - Linux 2.6.33-rc7
  - Driver: `iwlagn`
  - Measuring voltage drop across a shunt resistor → energy consumption
- Scenarios
  - Channel width 20 MHz and 40 MHz
  - Factors: varying number of...

    ...spatial streams

    ...link rates

    ...transmit power
  - Customized driver to allow quick reconfiguration
- Evaluation

→ How do the above factors effect energy consumption?

→ Suggestions how to react given work loads.

Demystifying 802.11n Power Consumption: Racing to Sleep

- When is racing to sleep *not* optimal?
  - Fast single stream configurations are better than other operation modes
  - Cases where fast single stream is *not* the most efficient operation mode are likely to be artificial scenarios
  - Depending on packet size other configurations are more efficient
- Bottom line
  - Fastest **single stream** operation available is most energy efficient
  - Use **multiple streams** only for large packets on strong links
- Findings and conclusions
  - **Cheap** (wrt. energy consumption):
    Doubling the bandwidth to double the bit rate
  - **Expensive** (wrt. energy consumption):
    Adding an additional transmit chain to increase data throughput
  - Commonly, **SISO is more energy efficient** than MIMO (surprisingly)

Demystifying 802.11n Power Consumption: Comments

- Pro
  - Well structured, overall good presentation
  - Easy to follow
  - Extensive evaluation section (workshop paper!)
  - Timely topic (standard was ratified at the time of publication)
  - Presentation of best practice based on evaluation results
- Contra
  - No related work (just a few references in the introduction)
  - „*New heuristics*" fall short
  - Open source driver modified, no details on changes (e.g. patches)
  - Measurement method prone to errors (sampling)

- Paper
  - *"The Synergy between Power-aware Memory Systems and Processor Voltage Scaling"*
    Xiaobo Fan, Carla S. Ellis, Alvin R. Lebeck
  - In *Proceedings of the Workshop on Power-Aware Computing Systems 2003*, San Diego, CA, USA
  - All authors from Duke University, Durham, USA
- **Evaluation** paper
- Paper structure
  - Motivation
  - Background and Related Work
  - The Synergy between DVS and Power-Aware Memory
  - DVS and Standard Memory
  - DVS and Power-Aware Memory
  - Summary and Conclusions

Motivation

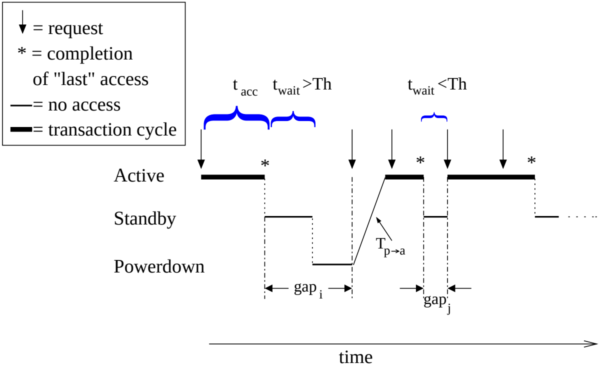
- Power consumption varies…
  - …linearly with frequency
  - …**quadratically** with voltage
- **Dynamic voltage and frequency scaling** (DVFS) has become a popular technique for decreasing energy consumption
  - Plenty of work available that proposes DVS algorithms
  - Running processors at lowest frequency **does not necessarily minimize overall energy consumption**
  - **Problem:** DVS algorithms do not work as expected because of other components' effects; particularly: **memory influences**
- **Observation**: memory energy costs may dominate CPU energy costs

Power-aware Memory

- Proposed **solution**: exploiting synergistic effect between DVS and power-aware memory to enable lower power states
- Memory's energy consumption **highly depends** on the efficiency the OS can manage available hardware power states
- **Power-aware memory**:
  - Memory that can transition into states that consume **less energy**
  - Transition adds additional **latency costs**
  - The lower the energy state, the higher the latency for switching back
- **Three-state model**:
  - active
  - standby
  - power down

Three-State Model

Power State or Transition	Power (mW)	Time (ns)
Active	$P_a = 275$	$t_{access} \approx 90$
Standby	$P_s = 75$	-
Powerdown	$P_p = 1.75$	-
Stby $\rightarrow$ Act	-	$T_{s \rightarrow a} = 0$
Pdn $\rightarrow$ Act	$P_{p \rightarrow a} = 138$	$T_{p \rightarrow a} = +7.5$



Source: [1]

Policies

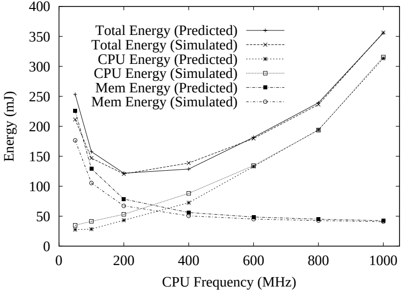
- **Page allocation policy**: Sequential vs. random page allocation
  $\rightarrow$ Keeping number of referenced DRAM chips at a minimum
- Powerdown policy
  - **Naive powerdown** policy:
    Powering down memory chips to the lowest power state after task completion (before the end of the period)
  - **Aggressive powerdown** policy:
    Immediately powering down memory chips in conjunction with application of sequential allocation

Evaluation

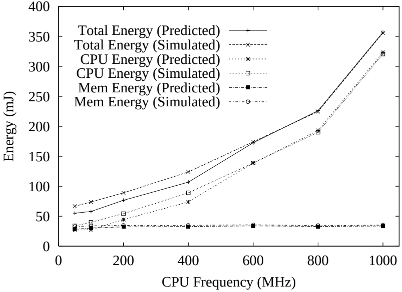
- Evaluation setup
  - Modified version (detailed Mobile-RAM memory model) of the PowerAnalyzer simulator
  - Simulated processor based on the Intel XScale
  - Frequency range: 50 MHz to 1000 MHz
  - Voltage range: 0.65 V to 1.75 V
- Workload generation
  - MediaBench suite
  - MPEG2dec
  - PEGWIT (public key encryption program)
  - G721 (voice compression)
- Varying computation times and **cache miss ratios**
- No **real measurements** but deriving energy values by means of performance counters

Evaluation (2)

- Observations. . .
  - . . . for naive powerdown:
    Lowest energy consumption is achieved with 200 MHz
  - . . . for aggressive powerdown:
    Lowest/Highest energy consumption is achieved with lowest/highest frequency (50 MHz)



a) Naive



b) Aggressive

From: [1]

References

[1] Fan, X. ; Ellis, C. S. ; Lebeck, A. R.:
The synergy between power-aware memory systems and processor voltage scaling.
In: *Proceedings of the Third International Conference on Power-Aware Computer Systems*.
Berlin, Heidelberg : Springer-Verlag, 2004 (PACS'03), S. 164–179

[2] Halperin, D. ; Greenstein, B. ; Sheth, A. ; Wetherall, D. :
Demystifying 802.11n Power Consumption.
In: *Proceedings of the 2010 International Conference on Power-Aware Computing and Systems*.
Berkeley, CA, USA : USENIX Association, 2010 (HotPower'10), S. 1–

[3] Lu, Y.-H. ; De Micheli, G. :
Comparing System-Level Power Management Policies.
In: *IEEE Design and Test of Computers* 18 (2001), März, Nr. 2, S. 10–19