Introduction on energy consumption in computing devices

Igor Neri
July 10, 2015

NiPS Summer School 2015
ICT-Energy: Energy consumption in future ICT devices

Outline

- The power problem and general energy aware techniques

- Power saving techniques on Wireless Sensor Network

- Computer architecture performance vs energy efficiency

- HPC: GPU vs CPU energy efficiency

Motivation

Energy-aware design, sometimes called energy-efficient design, is the design of a system to meet a given performance constraint with the minimum energy consumption.

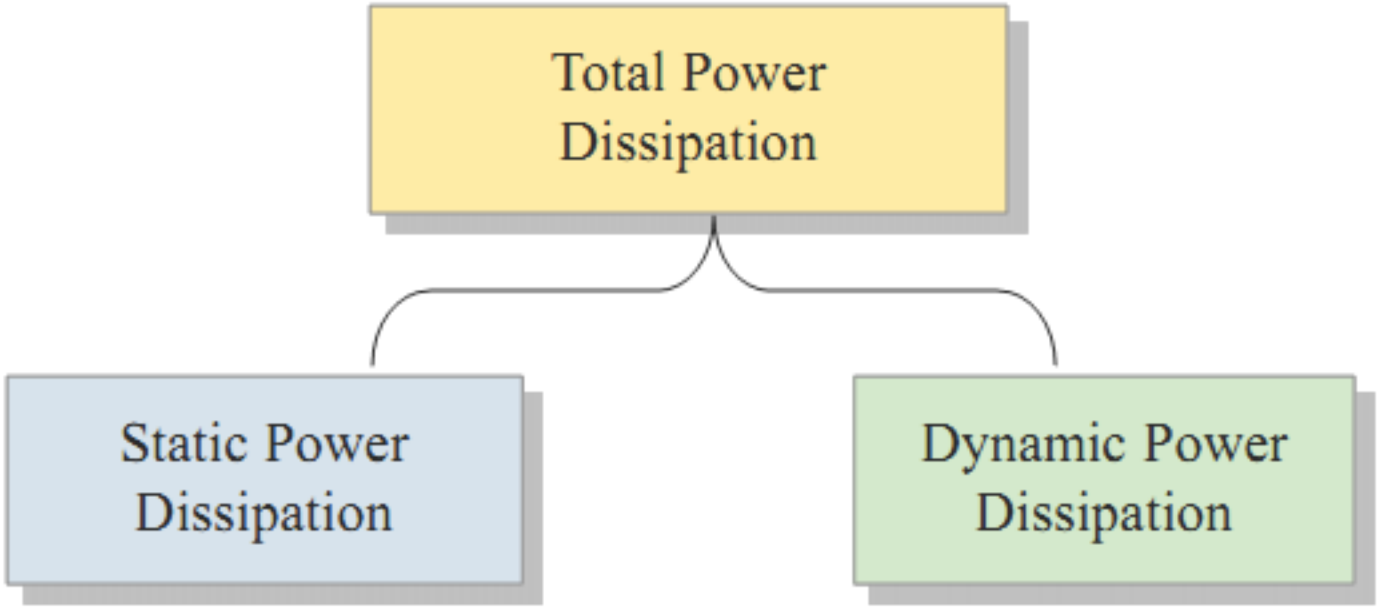
- Critical in **battery-operated** devices.
- Critical in terms of **cost** (computer centers).
- Critical since energy is converted into **heat**.

How

- Algorithm: scheduling, power-down strategies
- Data management: memory-aware software optimization, routing protocol
- Architecture: instruction set selection, dynamic voltage and frequency scaling
- Virtualization: power saving of corporate data centers
- Circuit: device sizing, exploiting of transistor stacking to reduce leakage power

Source of dissipation on CMOS

$$E = \int_0^t (V_{DD} I_{leak} + C V^2_{DD} f_c) dt$$



Total Power Dissipation

Static Power Dissipation

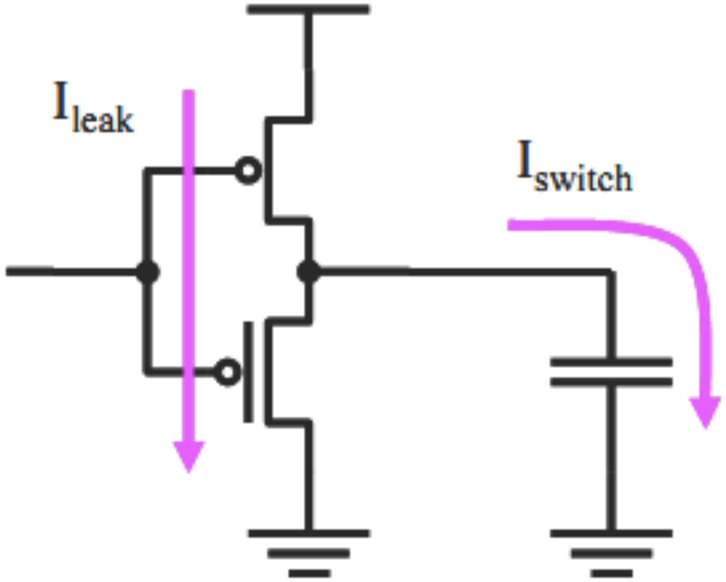
Dynamic Power Dissipation

$$\int_0^t V_{DD} I_{leak} dt$$

$$\int_0^t C V^2_{DD} f_c dt$$

Minimize I_{leak} by:

- Lower operating voltage
- Fewer leaking transistors



Minimize I_{switch} by:

- Lower operating voltage
- Less switching capacitance
- Less switching activity

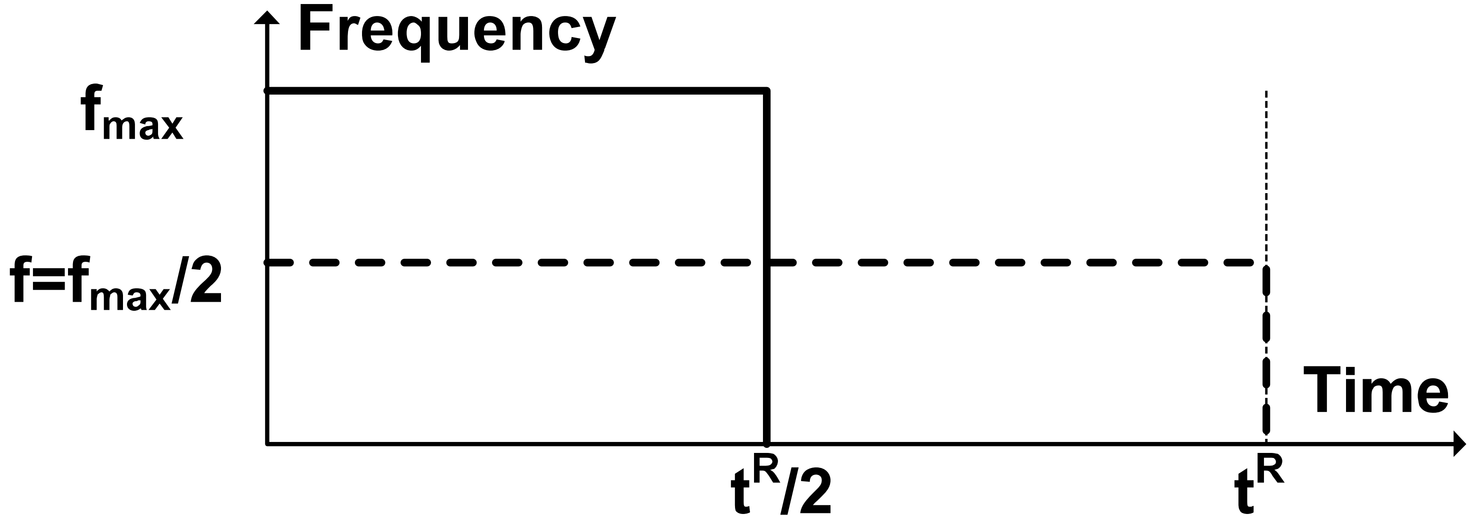
Dynamic Voltage and Frequency Scaling

$$P_{sw} = \tfrac{1}{2}\alpha C_L V_{dd}^2 f$$

- $\mathbf{P_{sw}}$ average switch power consumption
- α probability of output switch
- $\mathbf{C_L}$ load capacitance
- $\boldsymbol{f}$ clock frequency
- $\mathbf{V_{dd}}$ operating voltage

Dynamic Voltage and Frequency Scaling

$$f \propto \frac{(V_{\text{dd}} - V_{\text{th}})^{\alpha}}{V_{\text{dd}}}$$



- V_{dd} operating voltage
- V_{th} threshold voltage
- α is a measure of velocity saturation (1 ≤ α ≤ 2)

$$P_{\text{dyn}} \propto f^3$$

C.M. Kyung, S. Yoo - Energy-Aware System Design

Power down mechanism

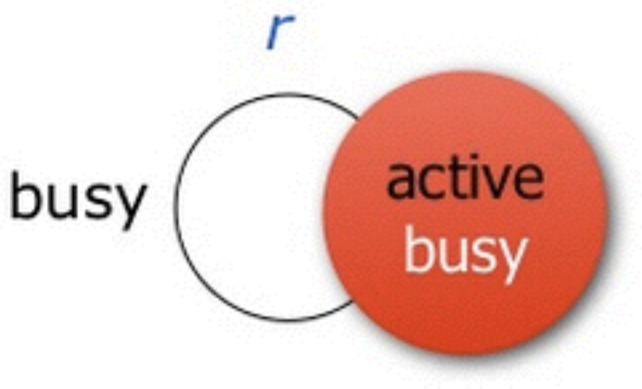
A system in idle state can be transitioned to **low power modes**

The goal is to develop transition schedules in order to minimise **energy consumption**

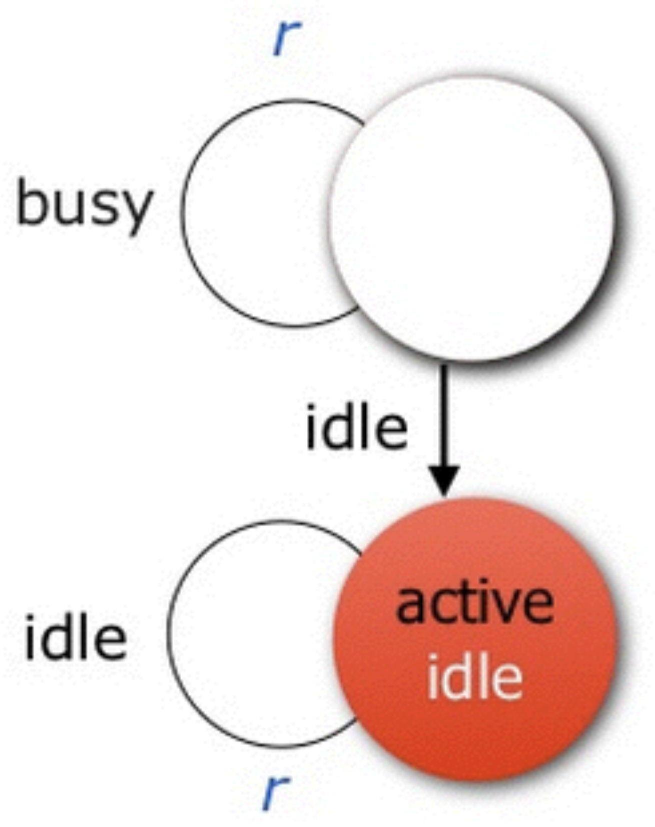
Power down mechanism:

- Two states system: **ON** - **OFF**

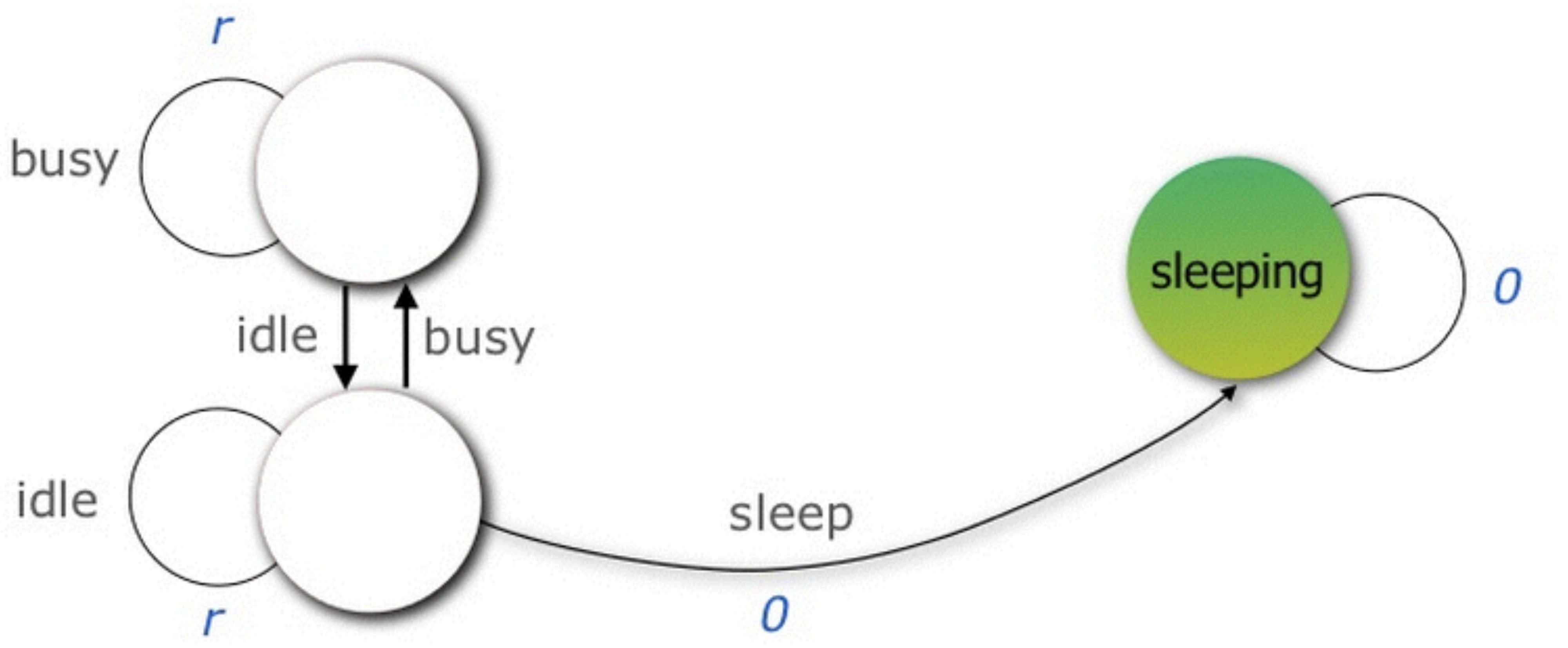
Power down mechanism: two states system



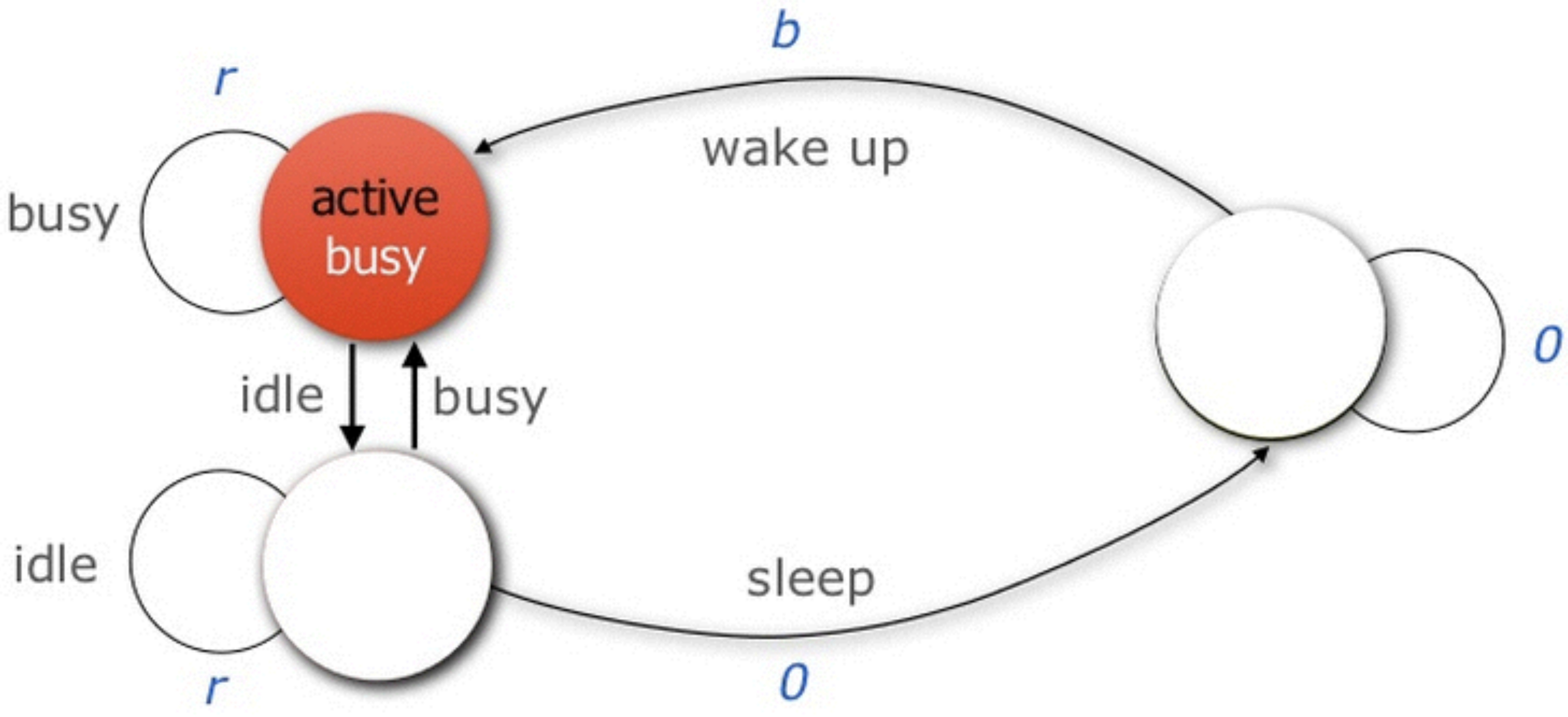
Power down mechanism: two states system



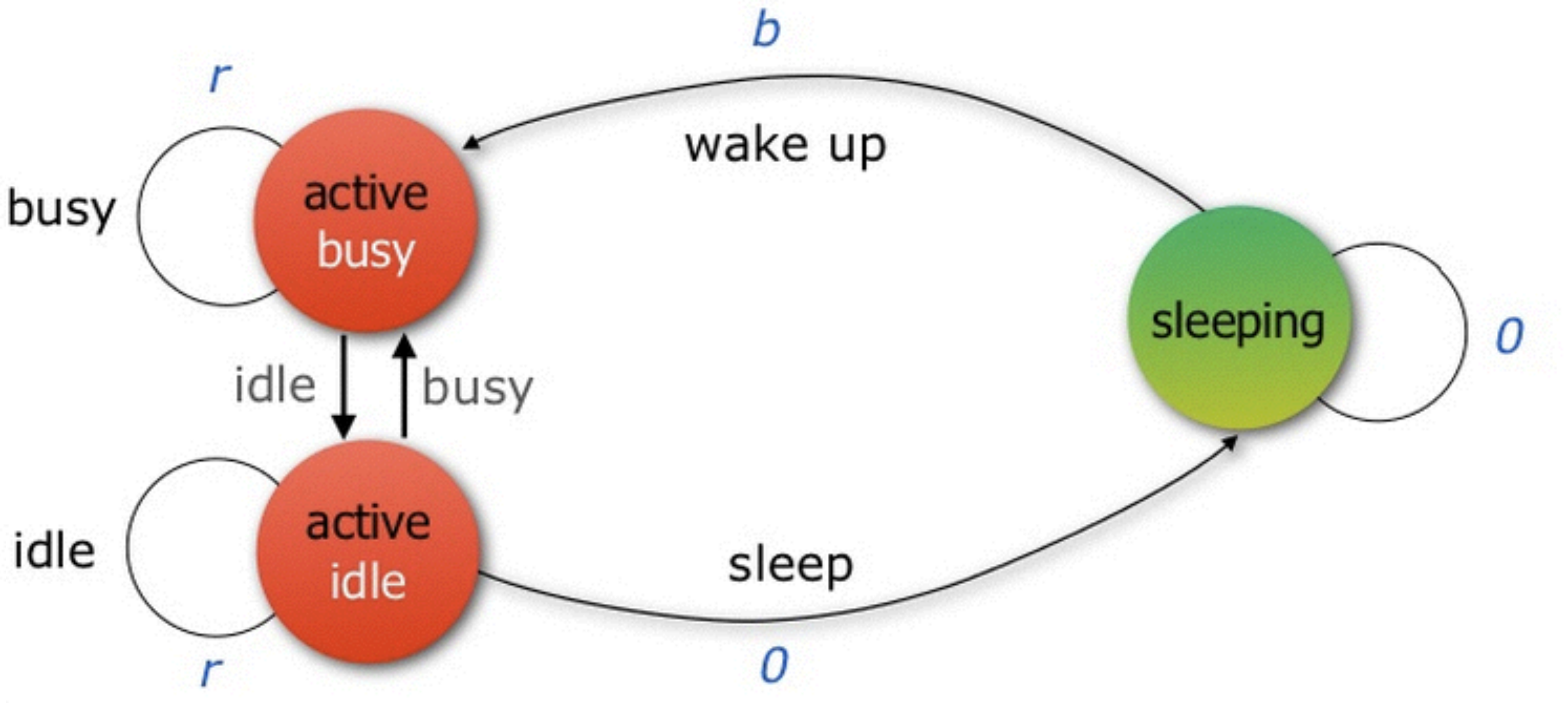
Power down mechanism: two states system



Power down mechanism: two states system



Power down mechanism: two states system



Problem: determine when to transition to the sleep state in order to minimize energy consumption.

Cees Witteveen, Energy-Efficient Algorithms

Power down mechanism

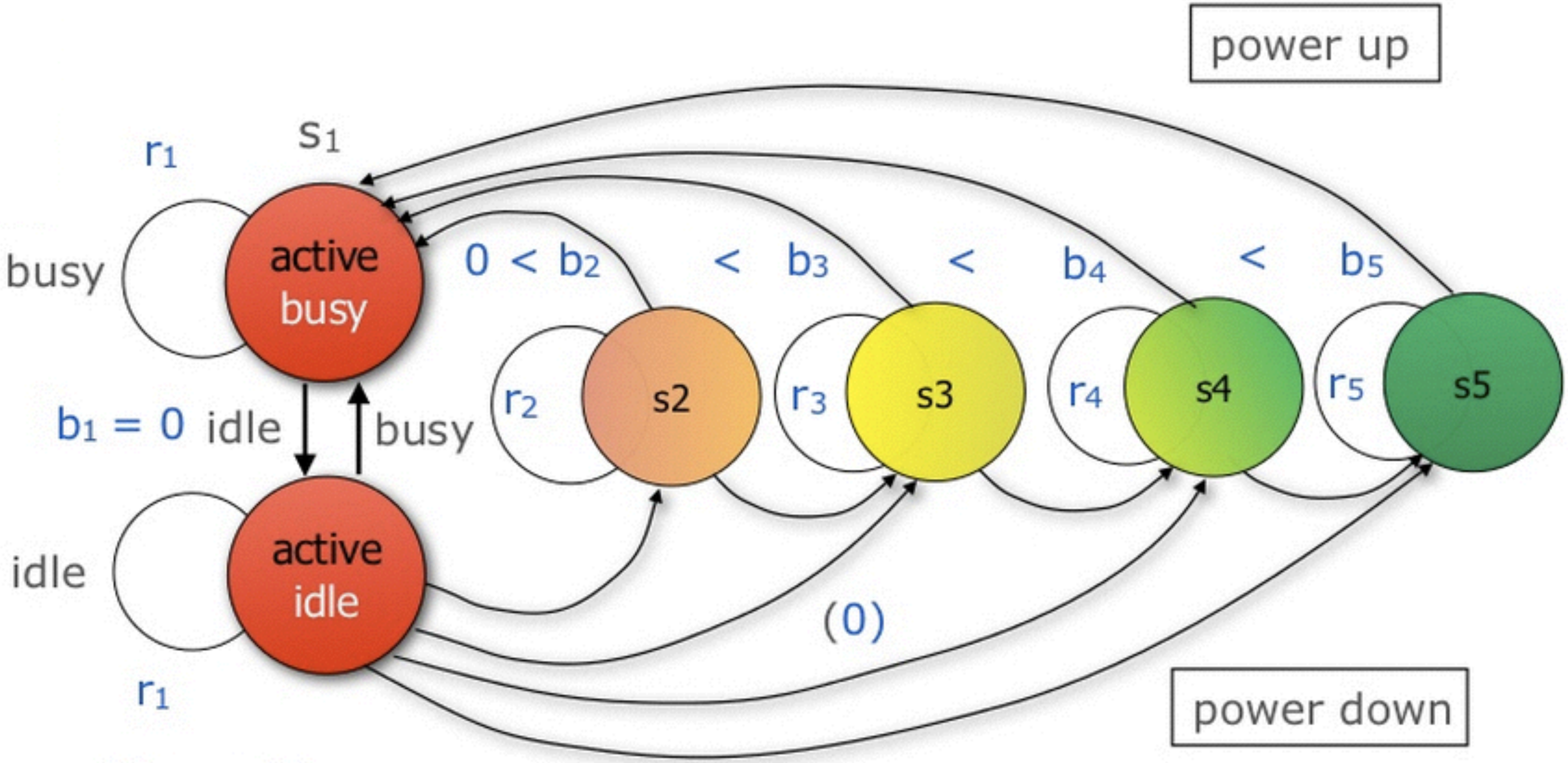
A system in idle state can be transitioned to **low power modes**

The goal is to develop transition schedules in order to minimise **energy consumption**

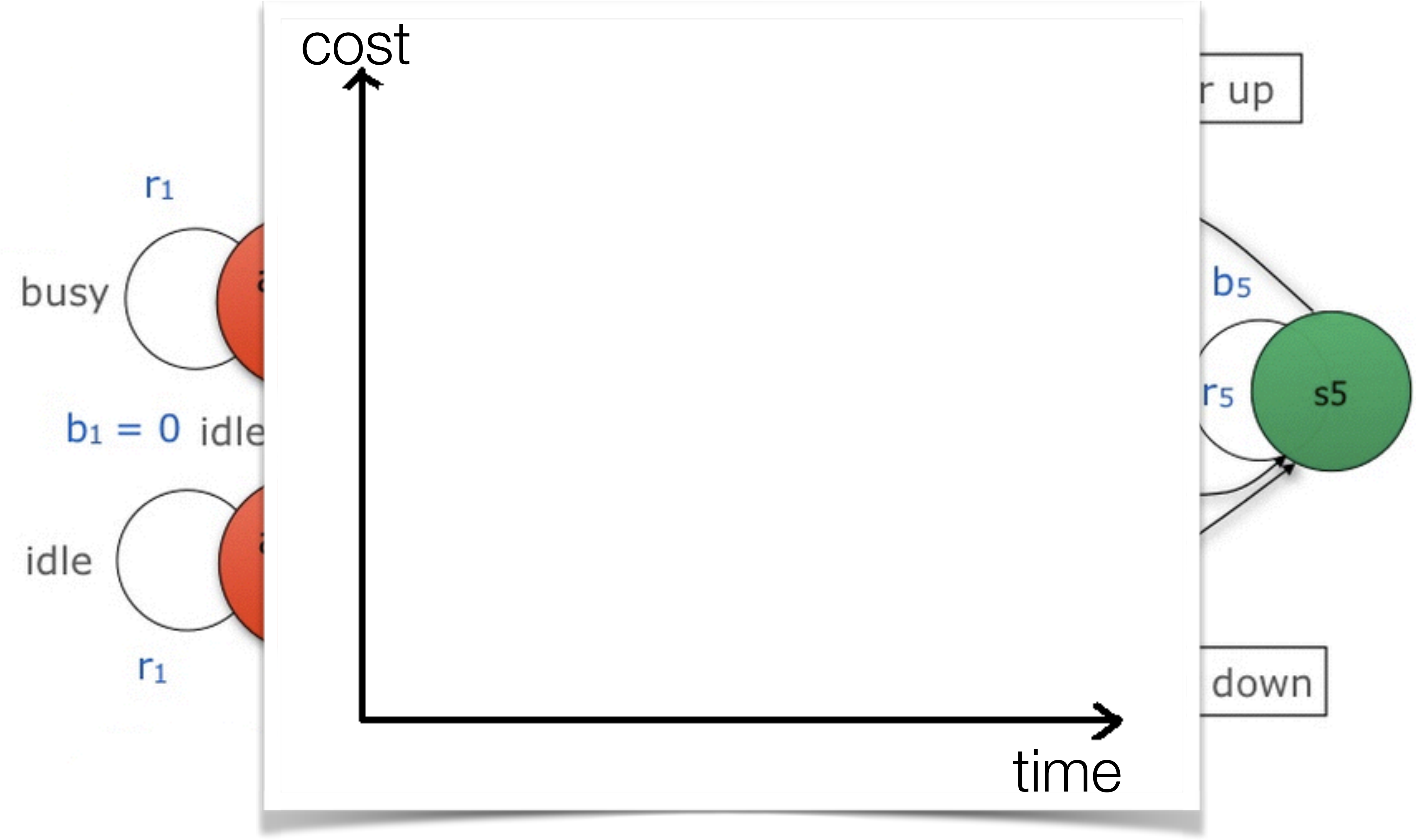
Power down mechanism:

- Two states system
- Multiple states system

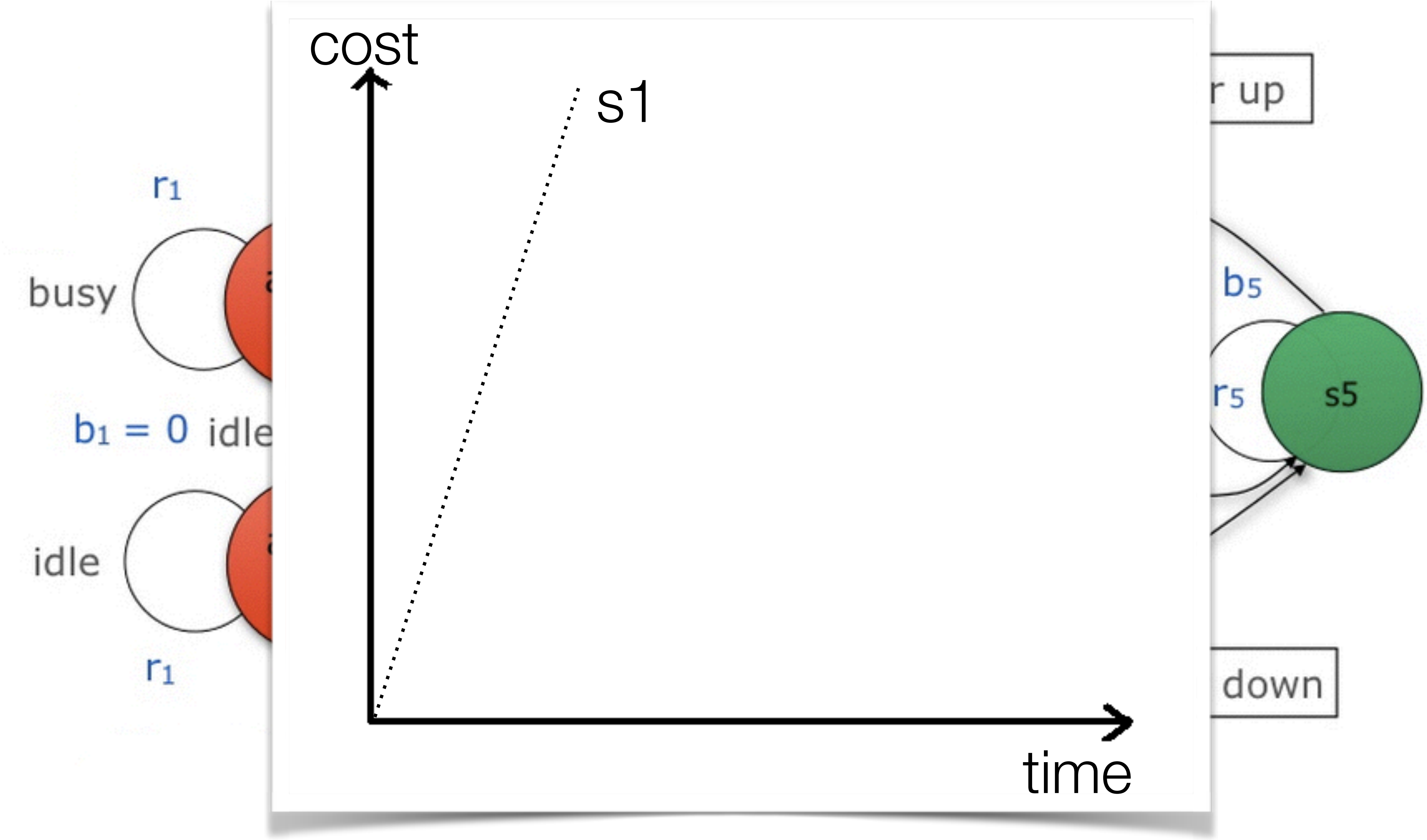
Power down mechanism: multiple states system



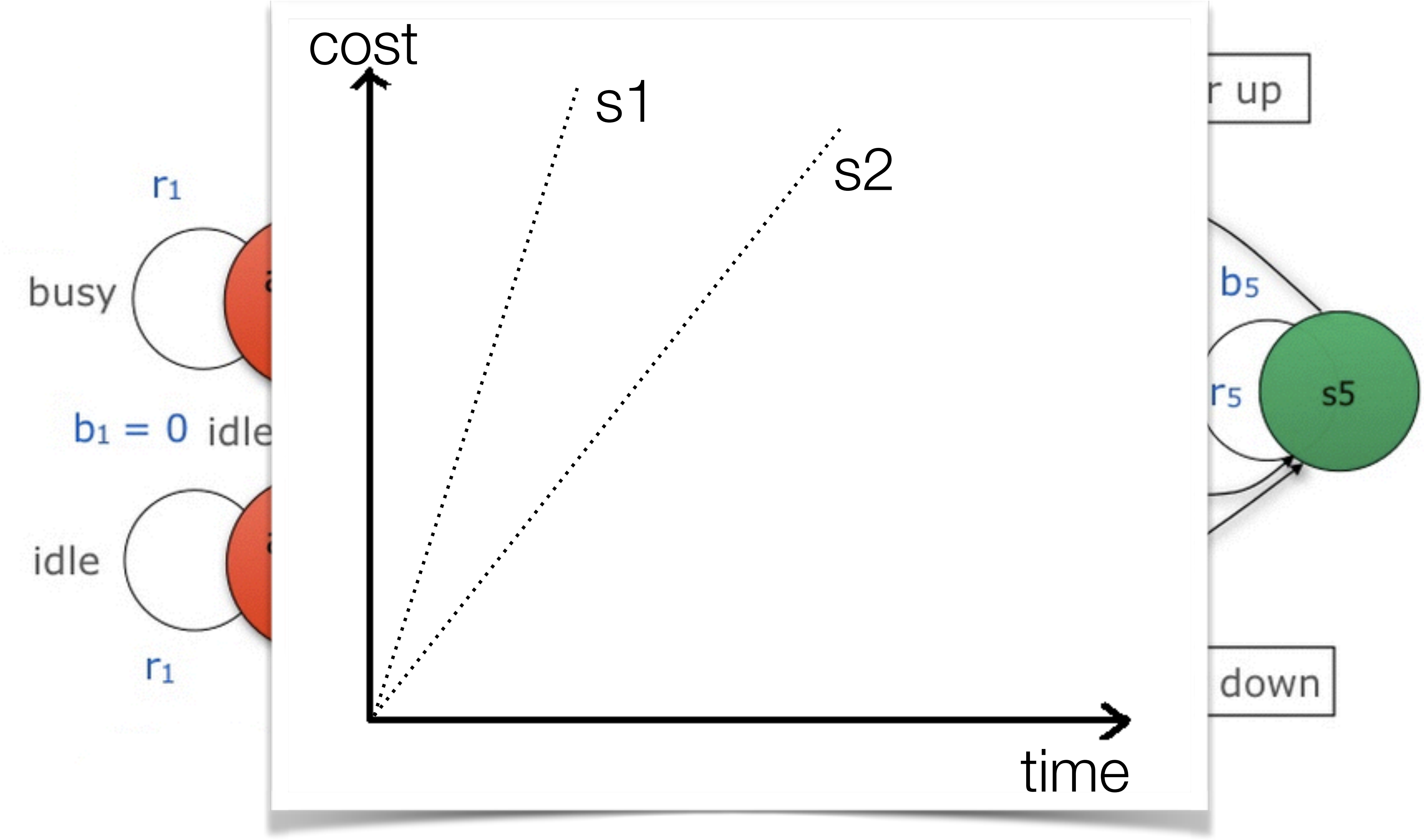
Power down mechanism: multiple states system



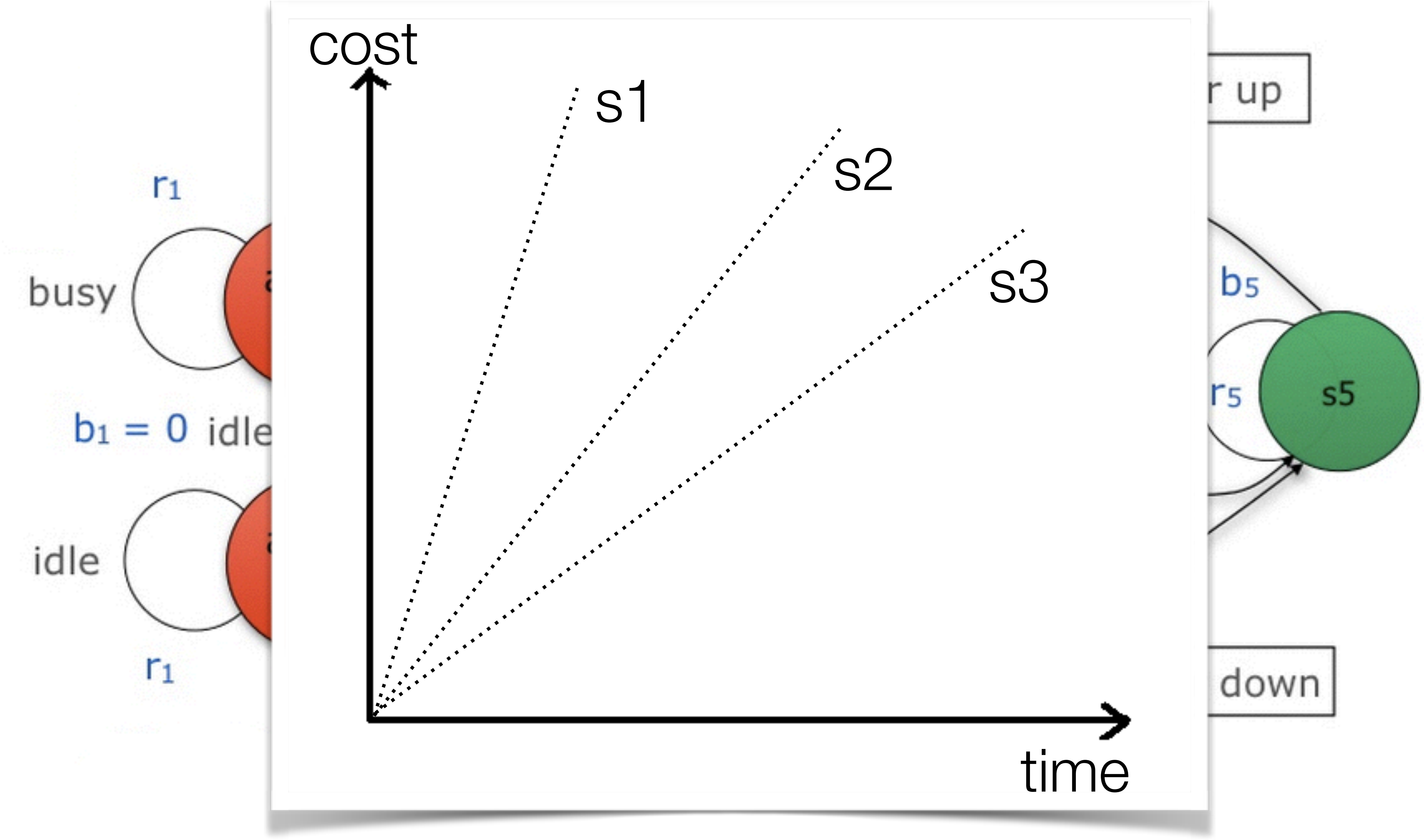
Power down mechanism: multiple states system



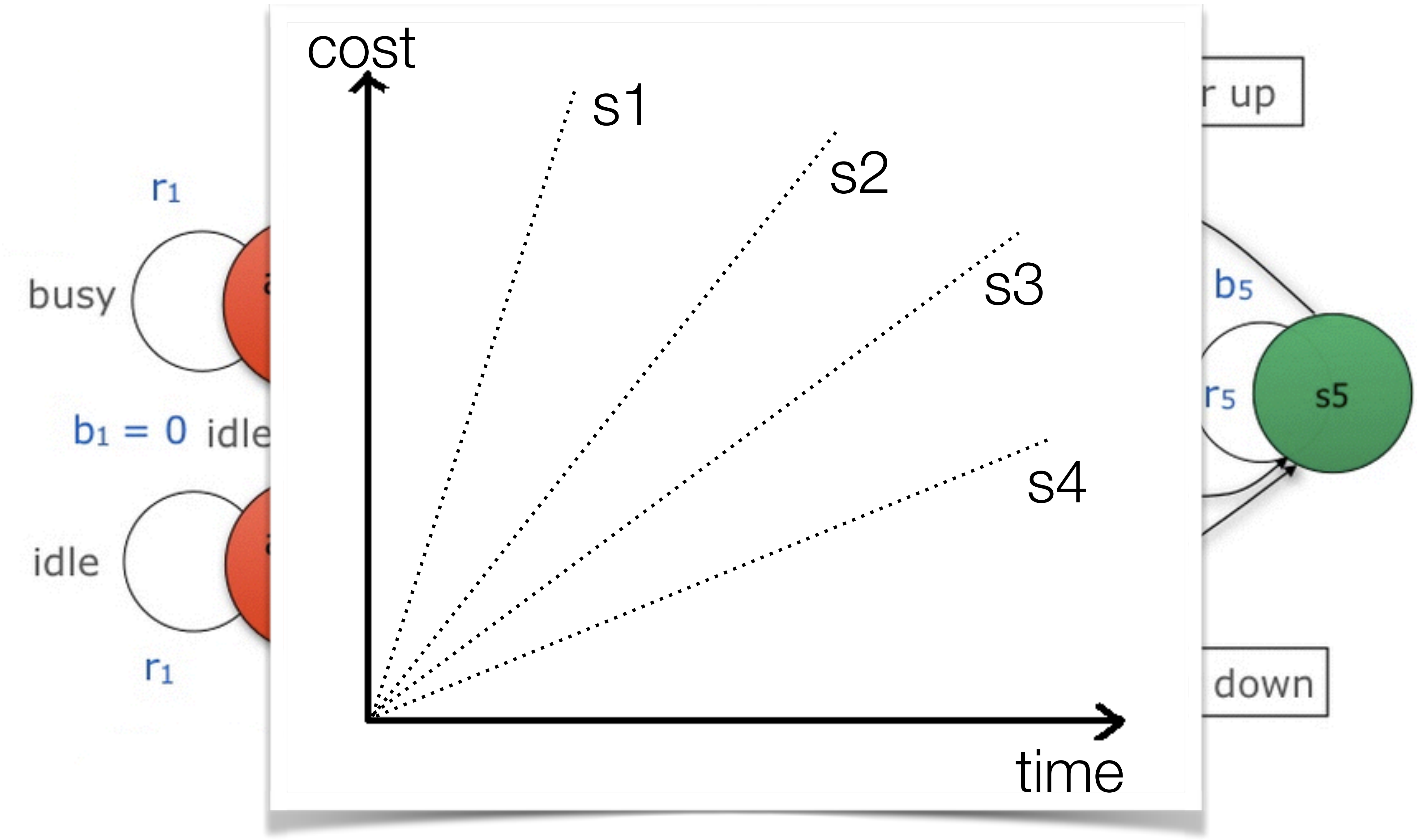
Power down mechanism: multiple states system



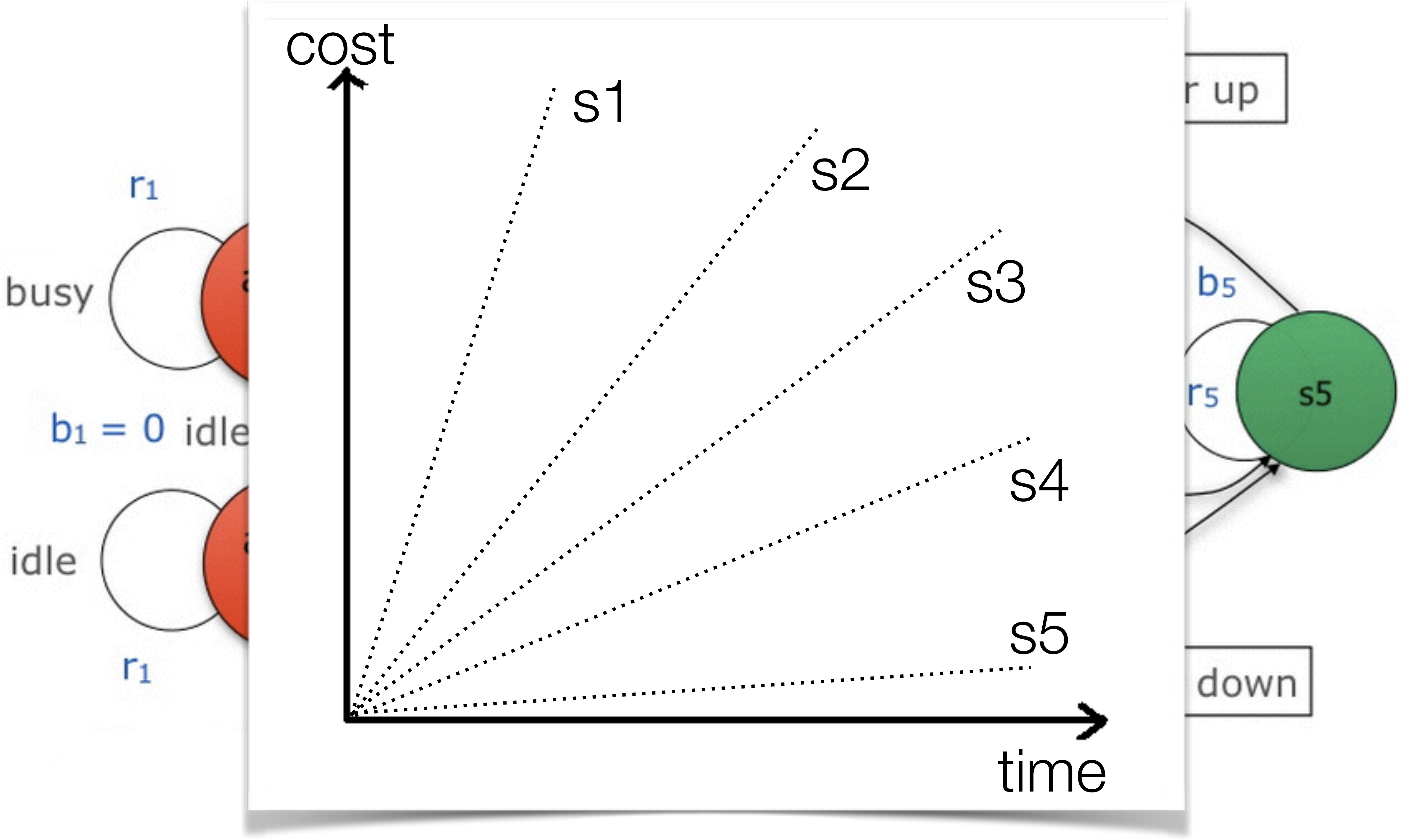
Power down mechanism: multiple states system



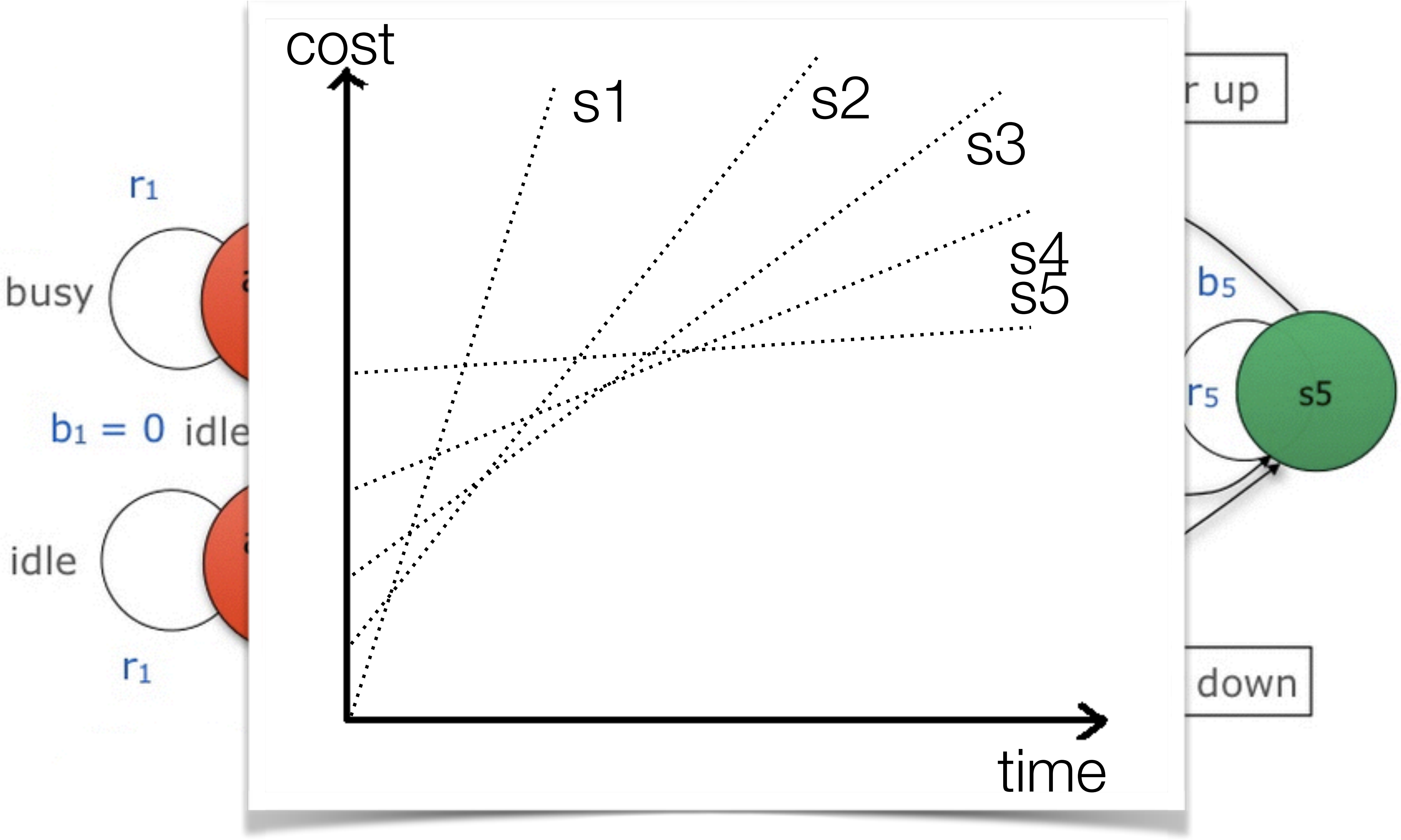
Power down mechanism: multiple states system



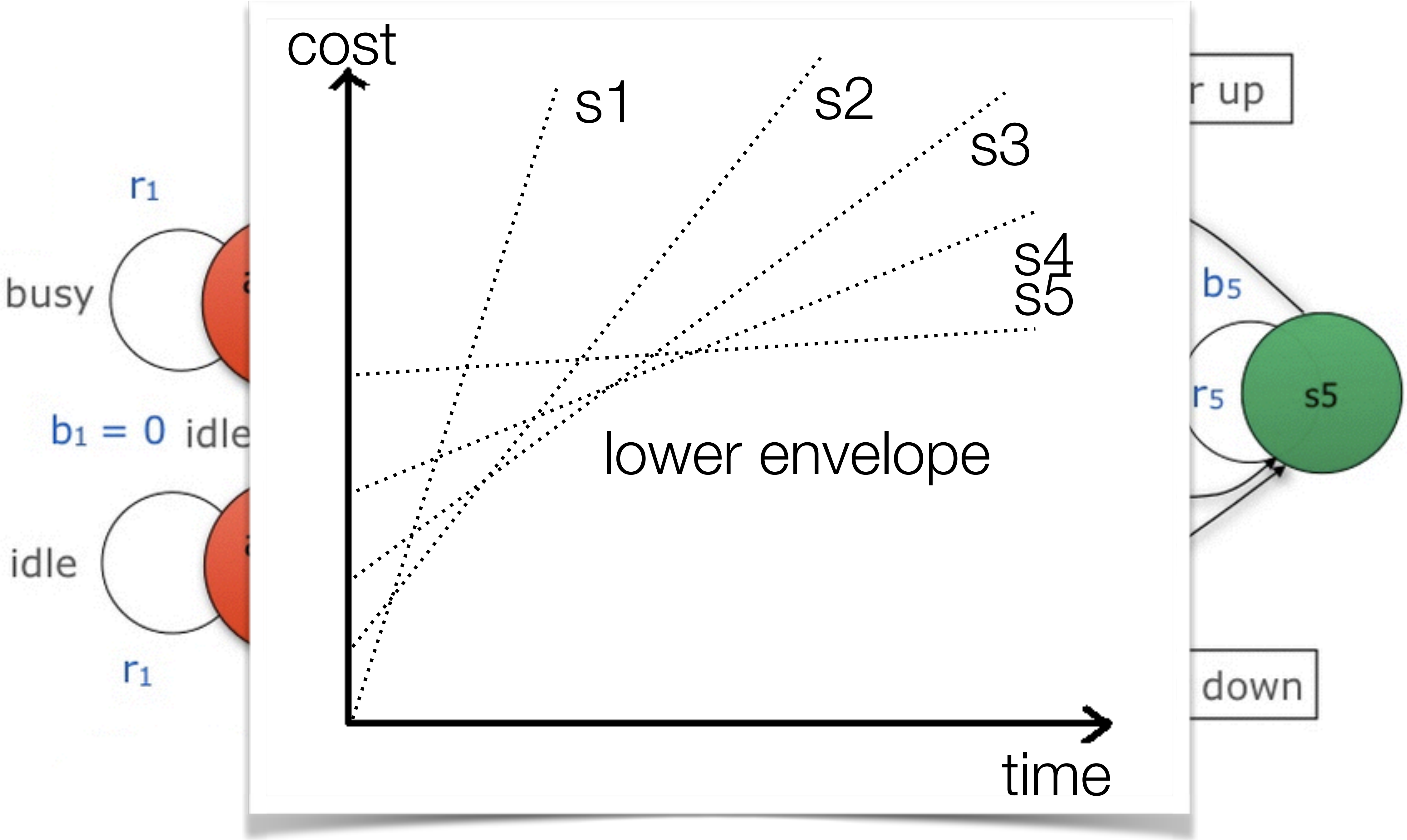
Power down mechanism: multiple states system



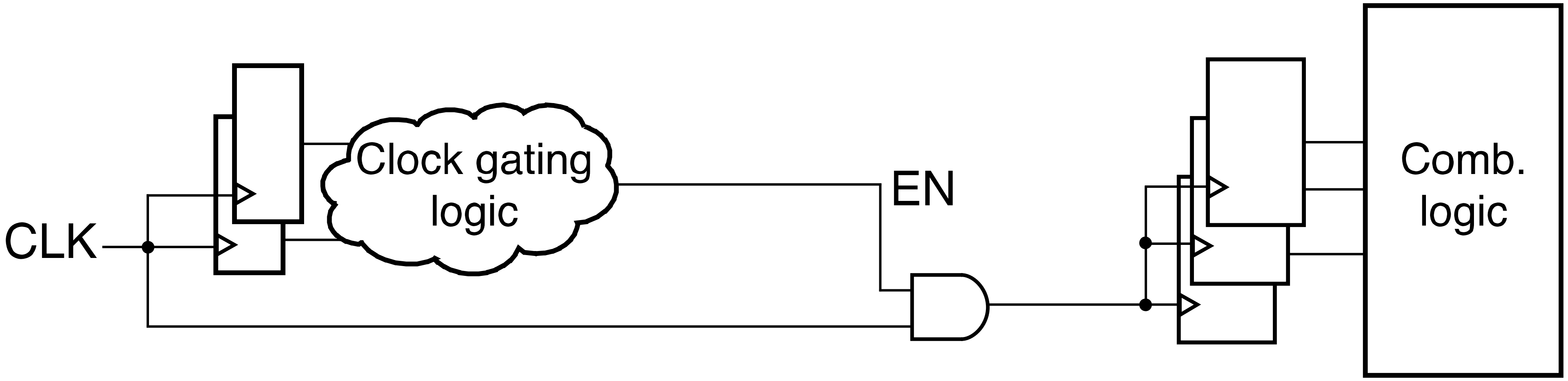
Power down mechanism: multiple states system



Power down mechanism: multiple states system



Clock Gating



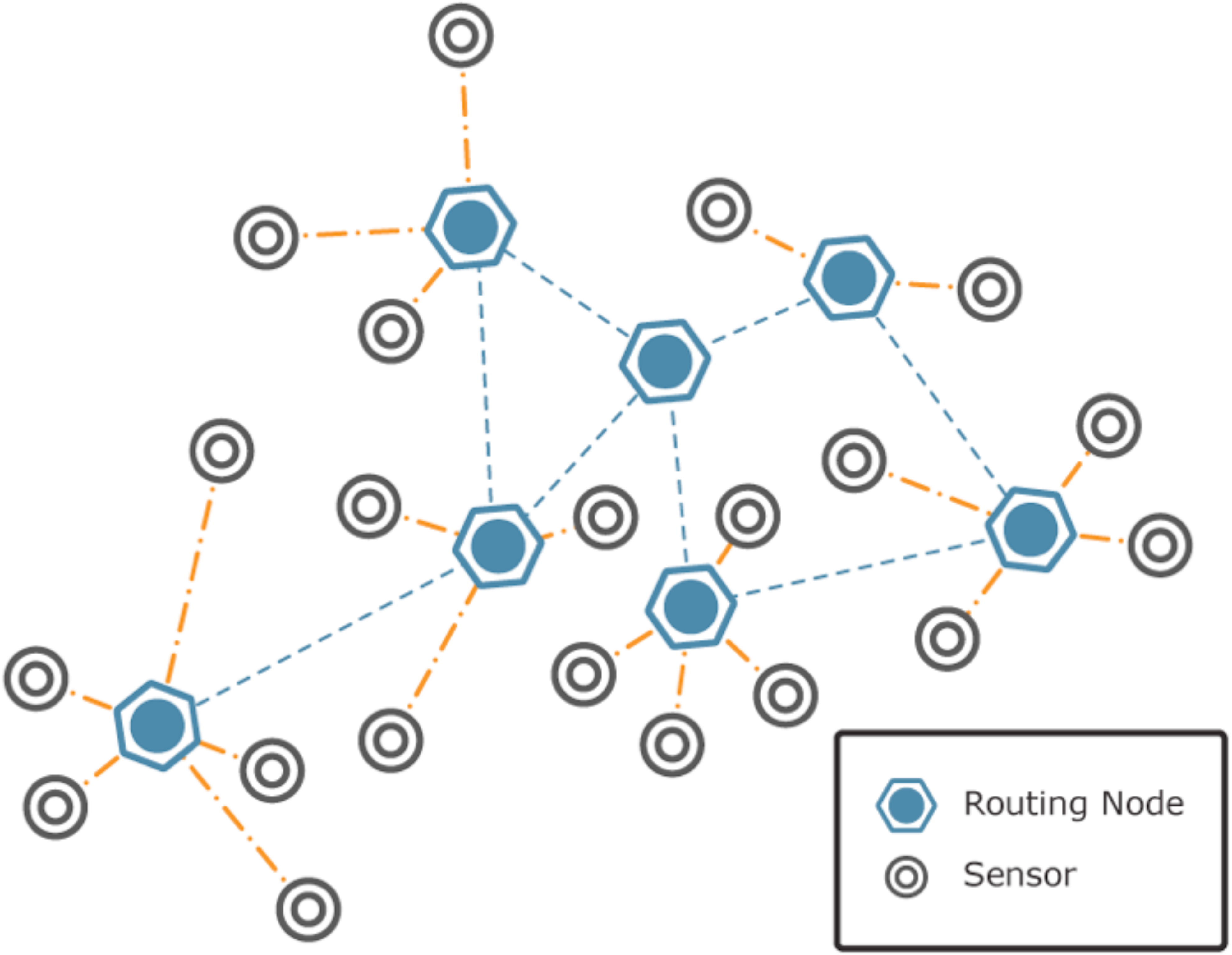
- Today all processors design use clock gating to some degree

- Non only low-power design but also many high-performance processors utilise clock gating because of non extant impact on performance:

    - IBM’s Power 5

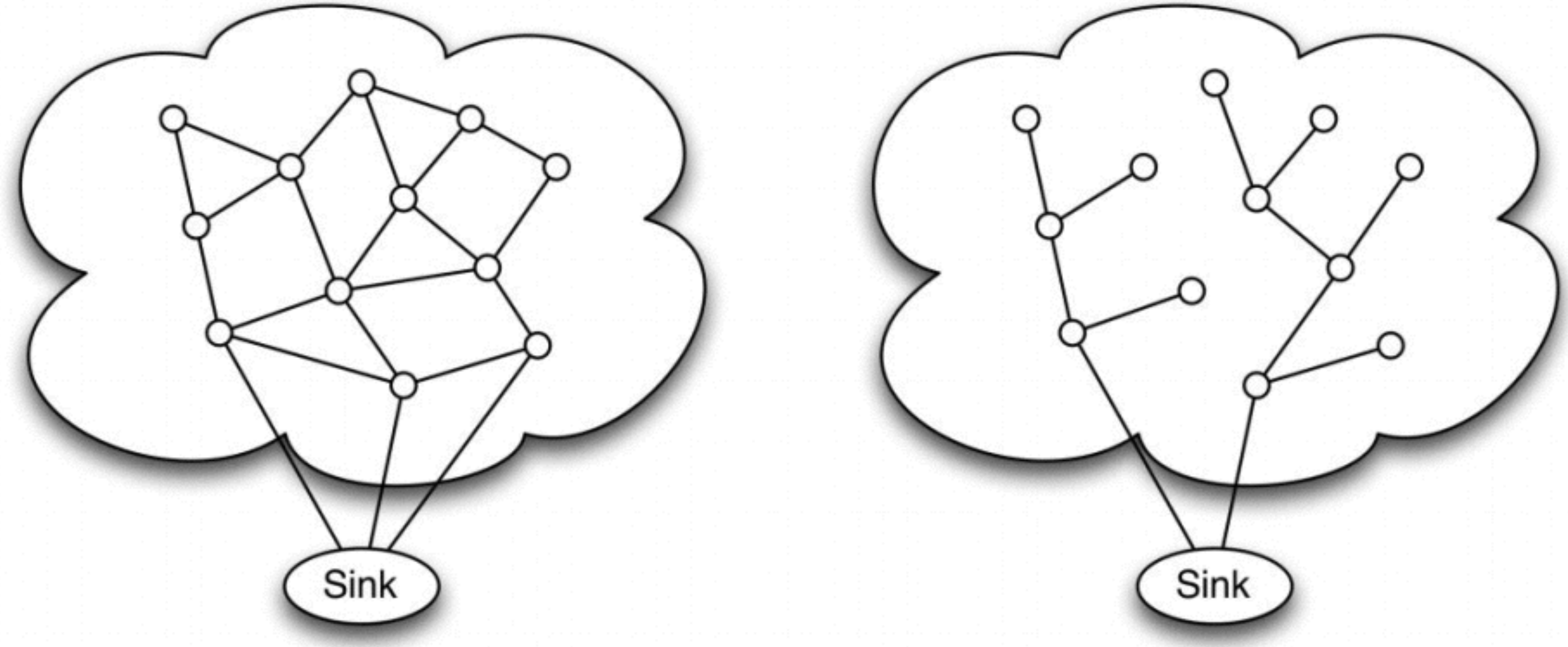
    - Intel’s X-Scale

Power saving techniques on Wireless Sensor Network



Where: Wireless Sensor Network

WSN

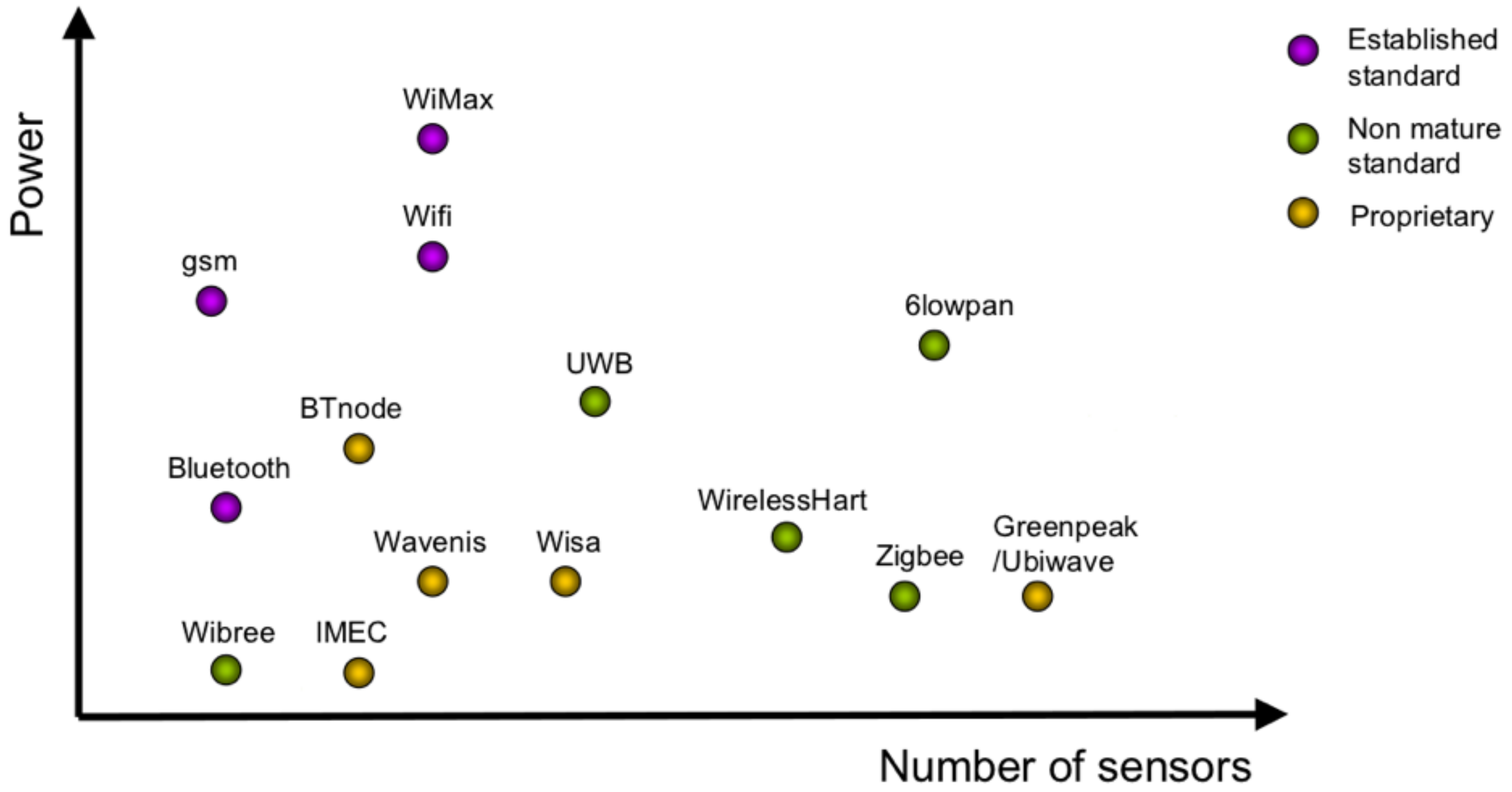


Flat architecture

Cluster architecture

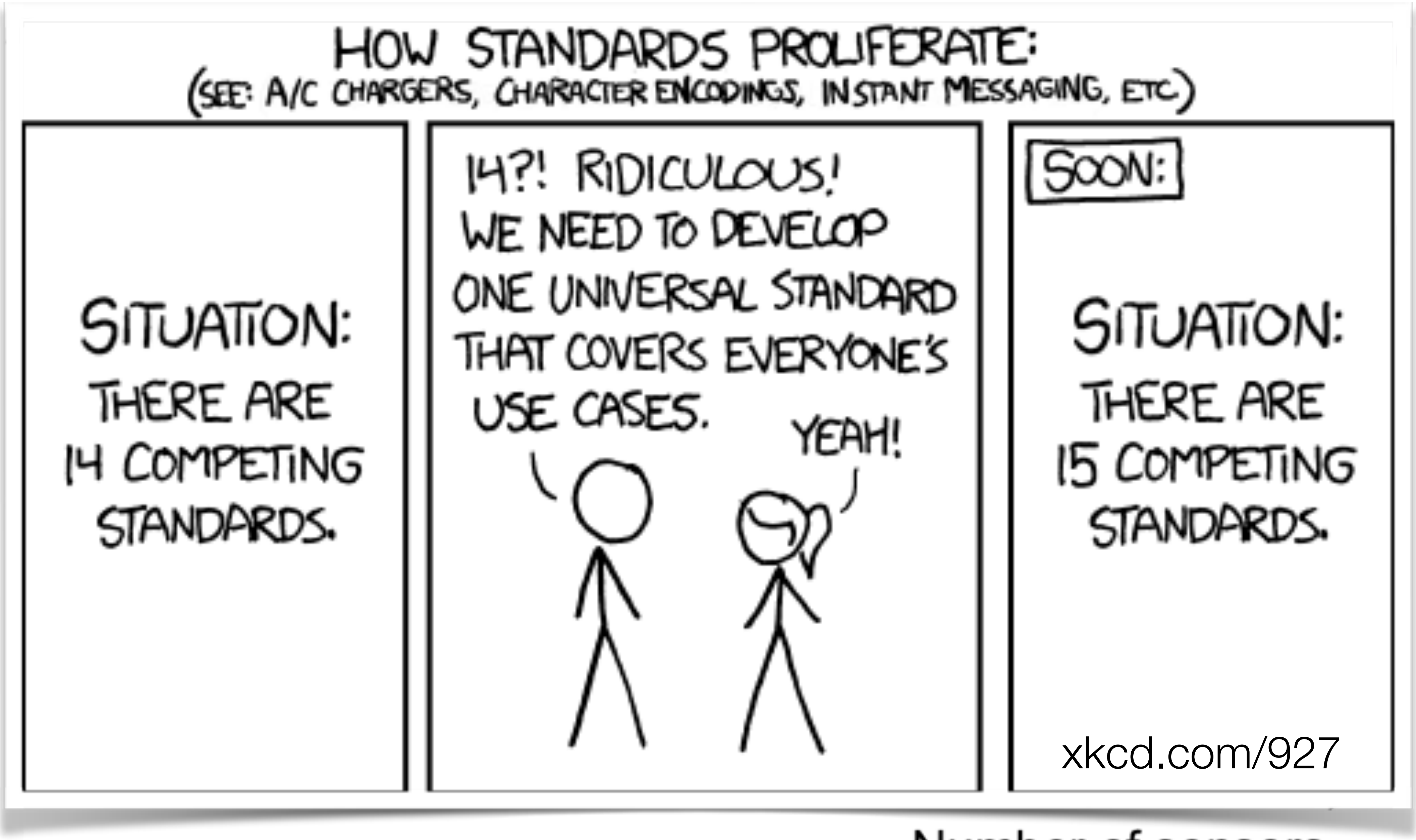
Where: Wireless Sensor Network Topology

WSN



Network | Standard?

Power


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

xkcd.com/927

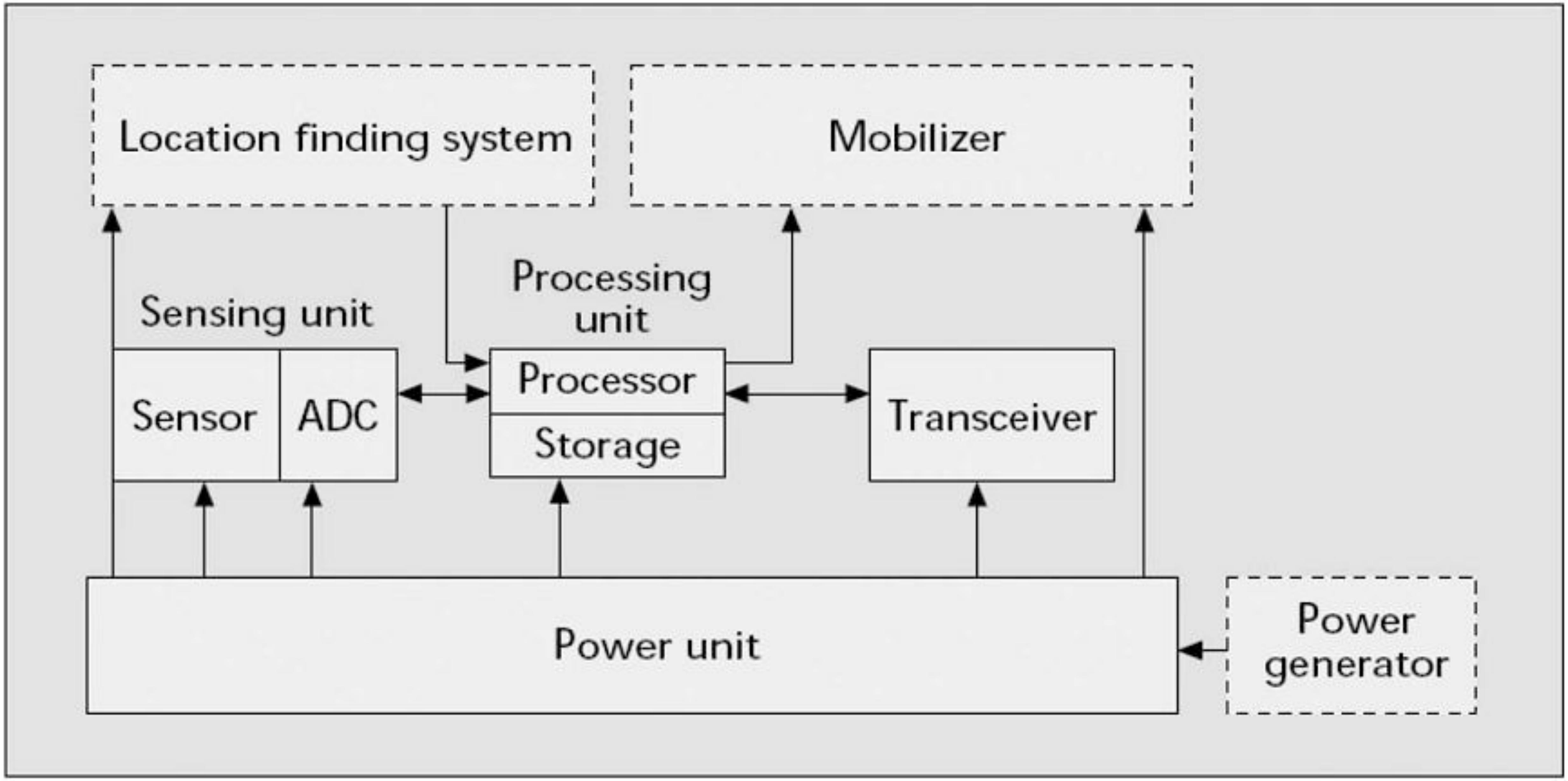
Established standard

Non mature standard

Proprietary

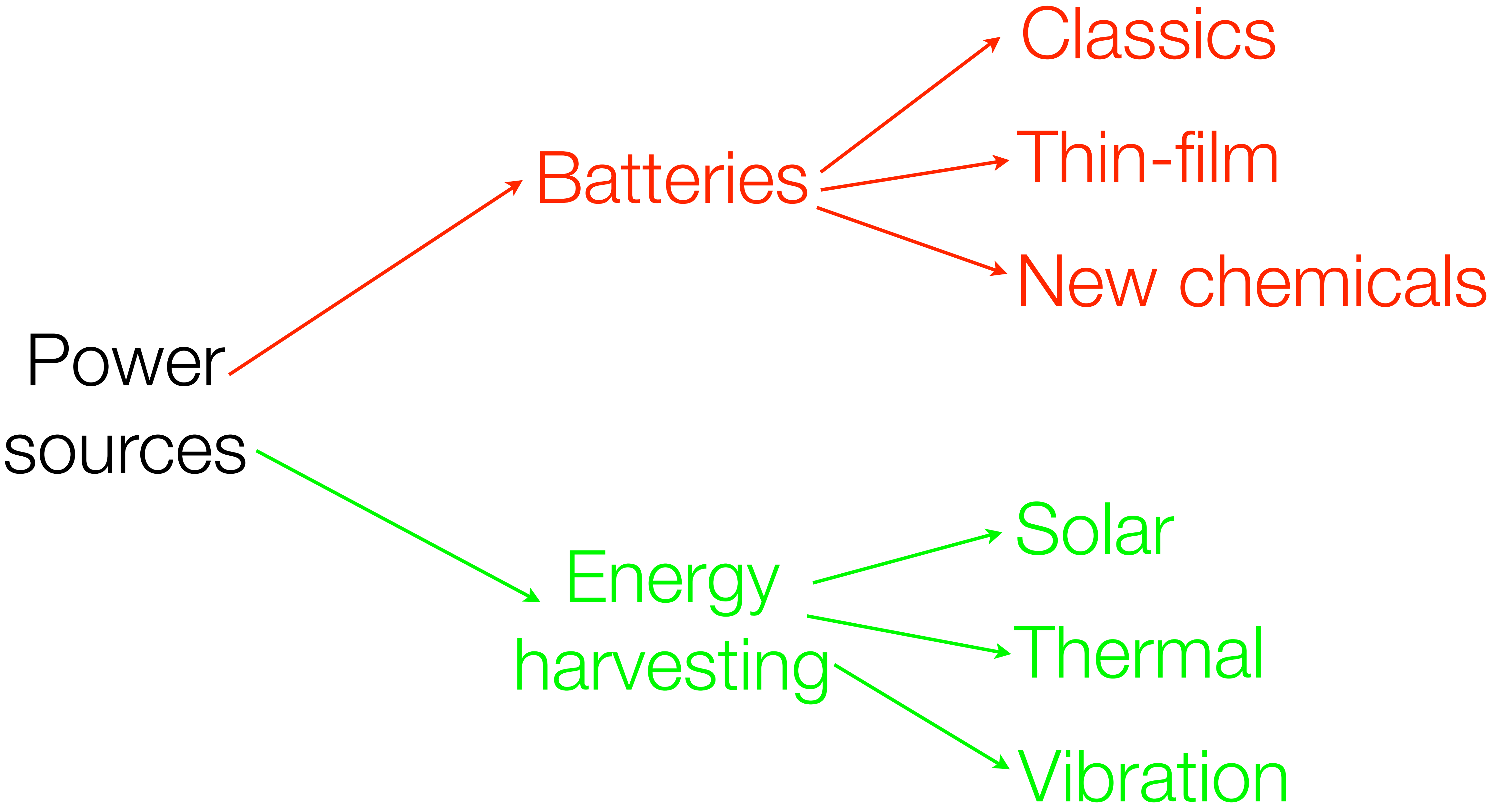
Number of sensors

Network | Standard?



Where: Sensor Node

Hardware



Power sources

Batteries

Classics

Thin-film

New chemicals

Energy harvesting

Solar

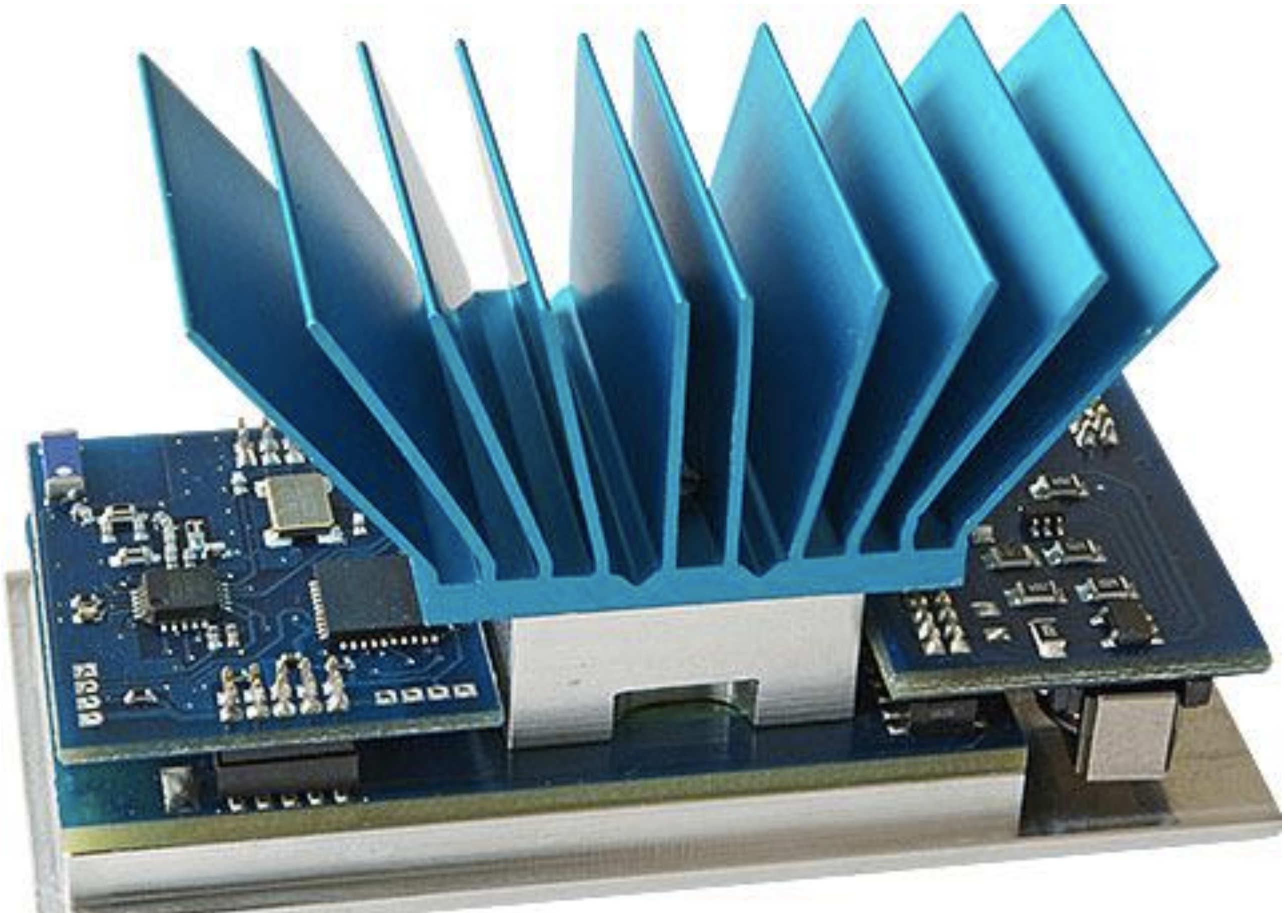
Thermal

Vibration

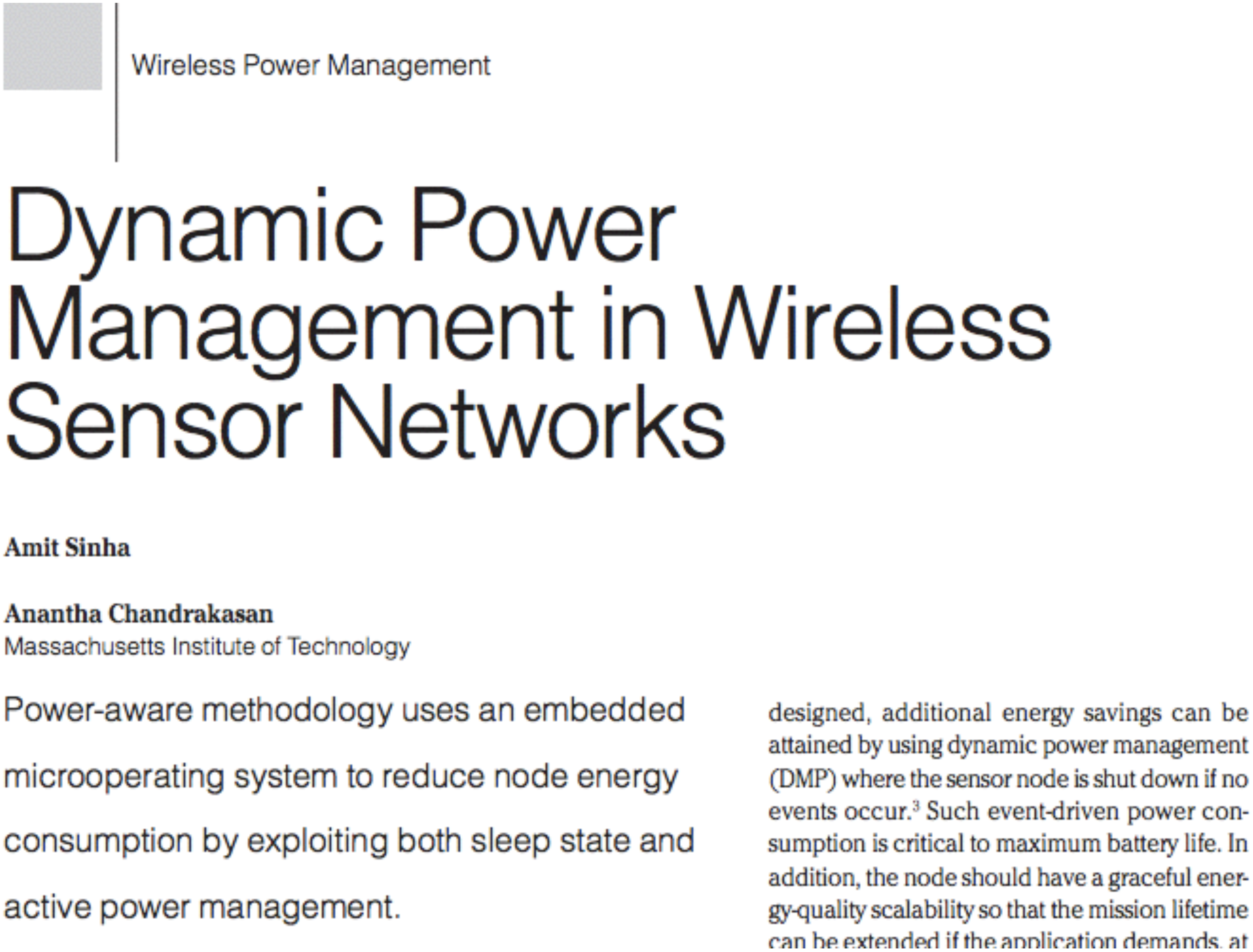
Where: Sensor Node

Power Source

ENERGY SAVING IN WSN



micropelt TE-Power PLUS



Wireless Power Management

Dynamic Power Management in Wireless Sensor Networks

Amit Sinha

Anantha Chandrakasan
Massachusetts Institute of Technology

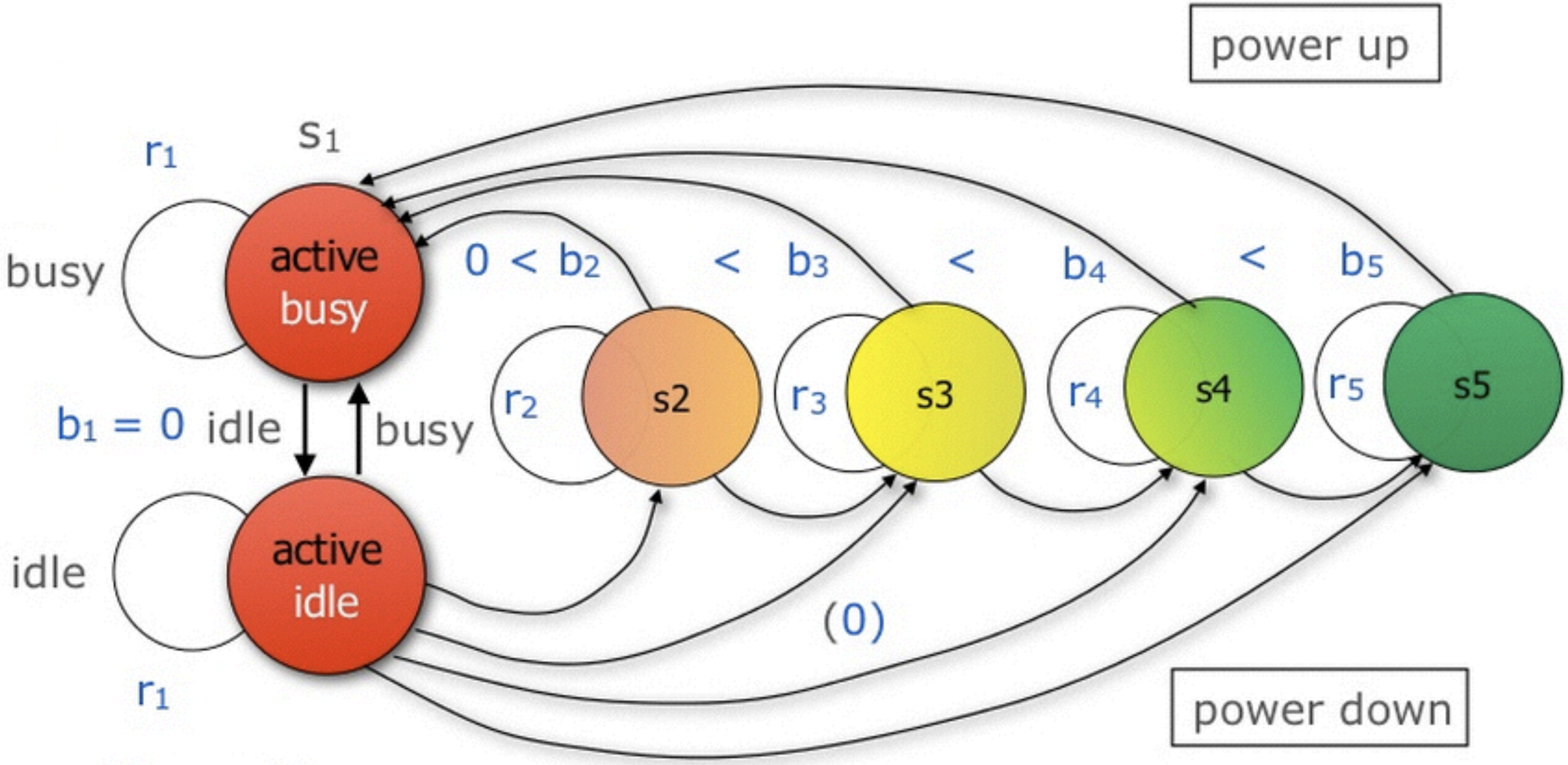
Power-aware methodology uses an embedded microoperating system to reduce node energy consumption by exploiting both sleep state and active power management.

designed, additional energy savings can be attained by using dynamic power management (DMP) where the sensor node is shut down if no events occur.[3] Such event-driven power consumption is critical to maximum battery life. In addition, the node should have a graceful energy-quality scalability so that the mission lifetime can be extended if the application demands, at

Dynamic Power Management

power-aware methodology

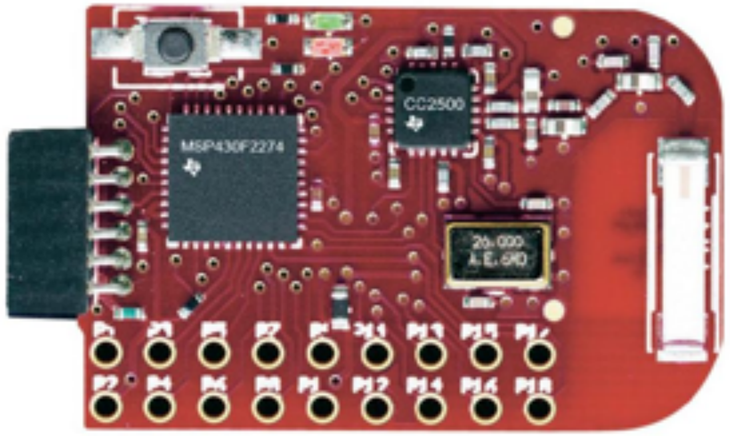
Power down mechanism: multiple states system



Cees Witteveen, Energy-Efficient Algorithms

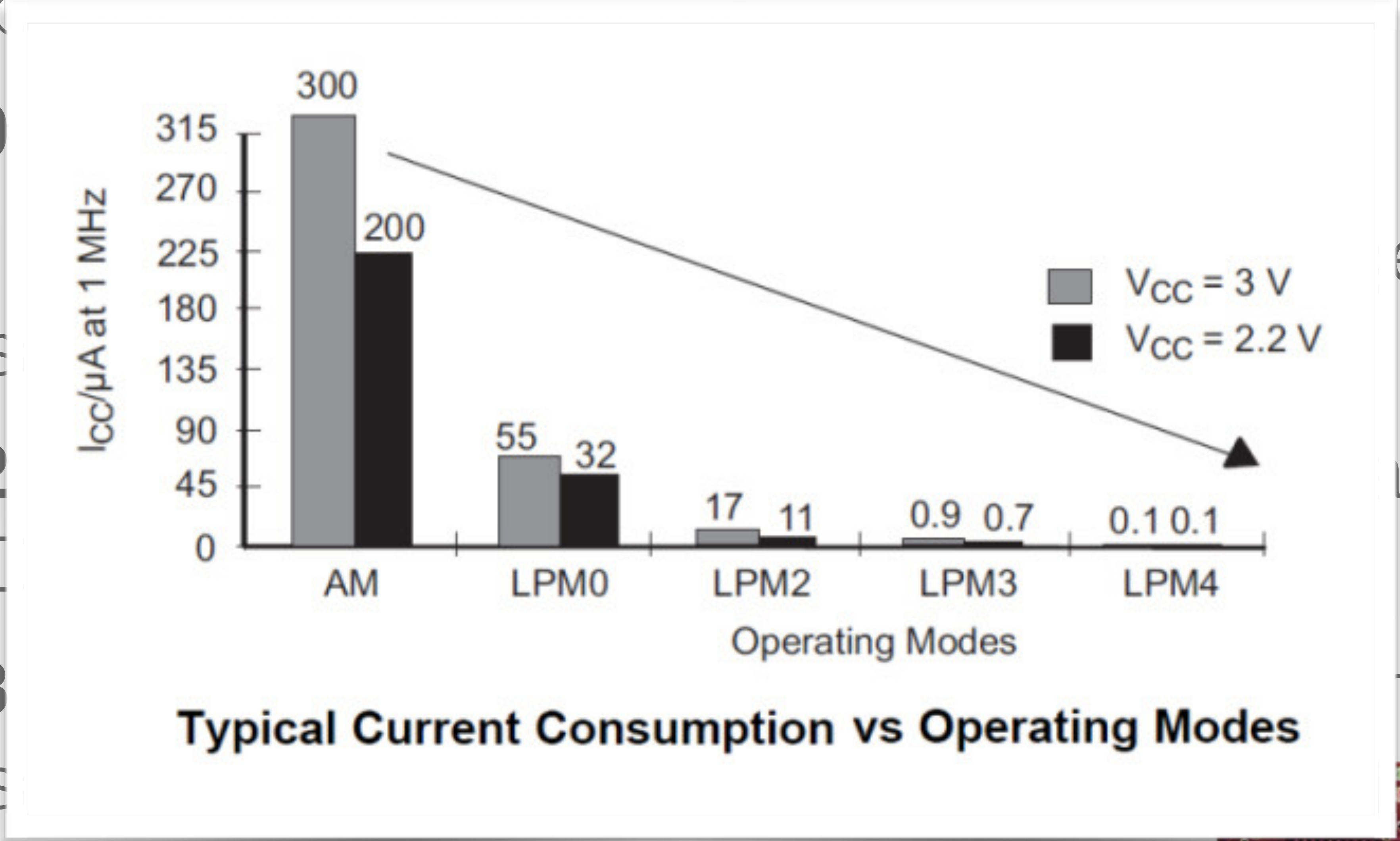
Power down mechanism: msp430

- **Active Mode** (AM) - Everything is turned on, except perhaps for some peripherals
- **LPM0** - CPU and MCLK are shutoff
- **LPM1** - CPU and MCLK are off, DCO and DC generator are disabled if the DCO is not used for SMCLK
- **LPM2** - CPU, MCLK, SMCLK and DCO are disabled, while DC generator is still enabled
- **LPM3** - CPU, MCLK, SMCLK, DCO and DC generator are disabled
- **LPM4** - CPU and all clocks disabled

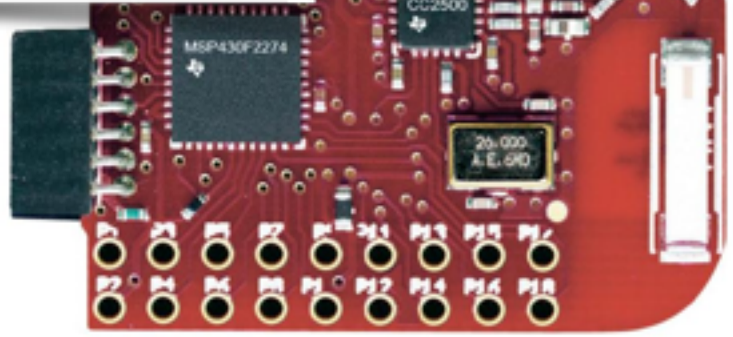


Power down mechanism: msp430

- **Active Mode** (AM) - Everything is turned on, except perhaps for some peripherals
- **LPM0**
- **LPM1** enerator are dis
- **LPM2** bled, while D
- **LPM3** nerator are dis
- **LPM4** - CPU and all clocks disabled



Typical Current Consumption vs Operating Modes



Media Access Control (MAC) Layer

Transmission and reception are energy expensive operations

Objective:

- minimise worthless transmission

How:

- minimise collisions
- minimise cost of collisions

MAC Layer: MACA and MACAW

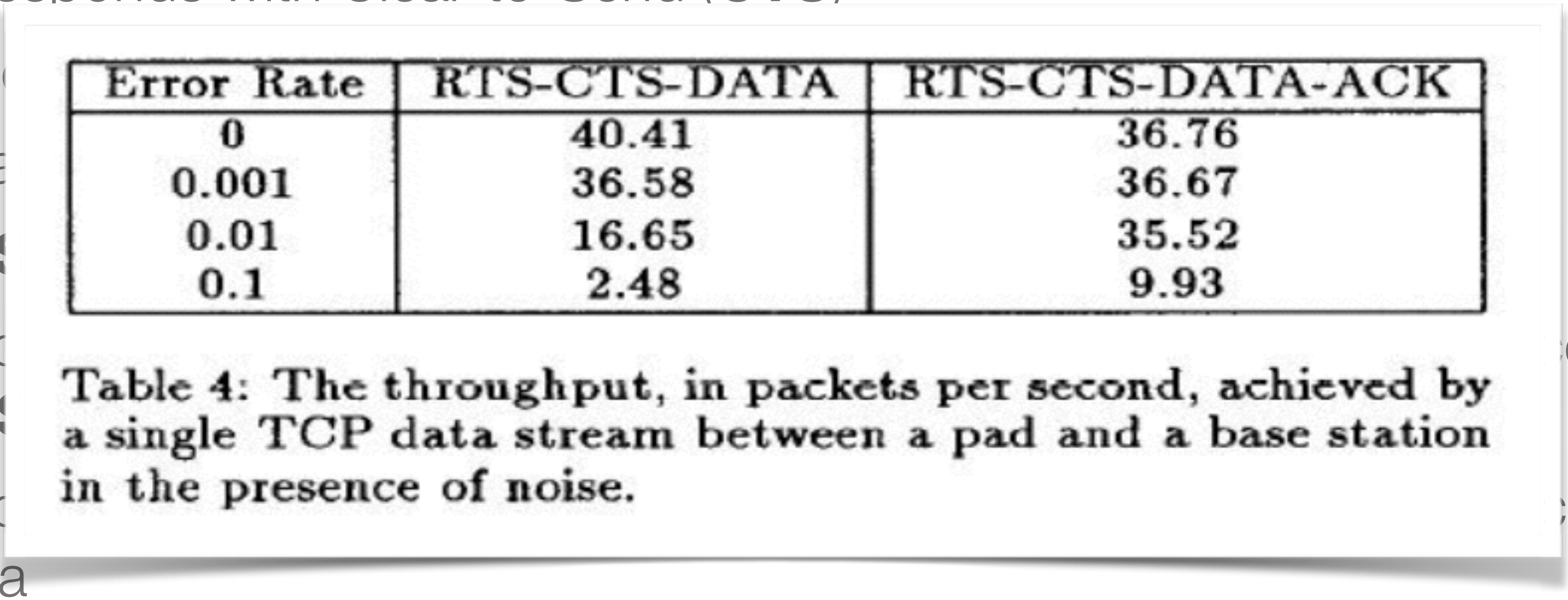
- **A** sends Ready-to-Send (**RTS**)
- **B** responds with Clear-to-Send (**CTS**)
- **A** sends **DATA PACKET**
- (**B** acknowledge with ACK)
- **RTS** and **CTS** announce the duration of the data transfer
- Nodes overhearing **RTS** keep quiet for some time to allow **A** to receive **CTS**
- Nodes overhearing **CTS** keep quiet for some time to allow **B** to receive data
- (**A** will retransmit **RTS** if no **ACK** is received)

P. Karn, “MACA - A new channel access method for packet radio”, in Proceedings of the ARRL CRRL Amateur Radio 9th Computer Networking Conference, Redondo Beach, CA, Apr. 1990, pp. 134-140.

V. Bharghavan, A. Demers, S. Shenkar, and L. Zhang, “MACAW: A media access protocol for wireless LANs”, in Proceedings of ACM SIGCOMM’94, London, UK, Sept. 1994, pp. 212-225.

MAC Layer: MACA and MACAW

- **A** sends Ready-to-Send (**RTS**)
- **B** responds with Clear-to-Send (**CTS**)
- **A** s
- (**B** a
- **RT**
- Noc eive **CT**
- Noc eive data
- (**A** will retransmit **RTS** if no **ACK** is received)



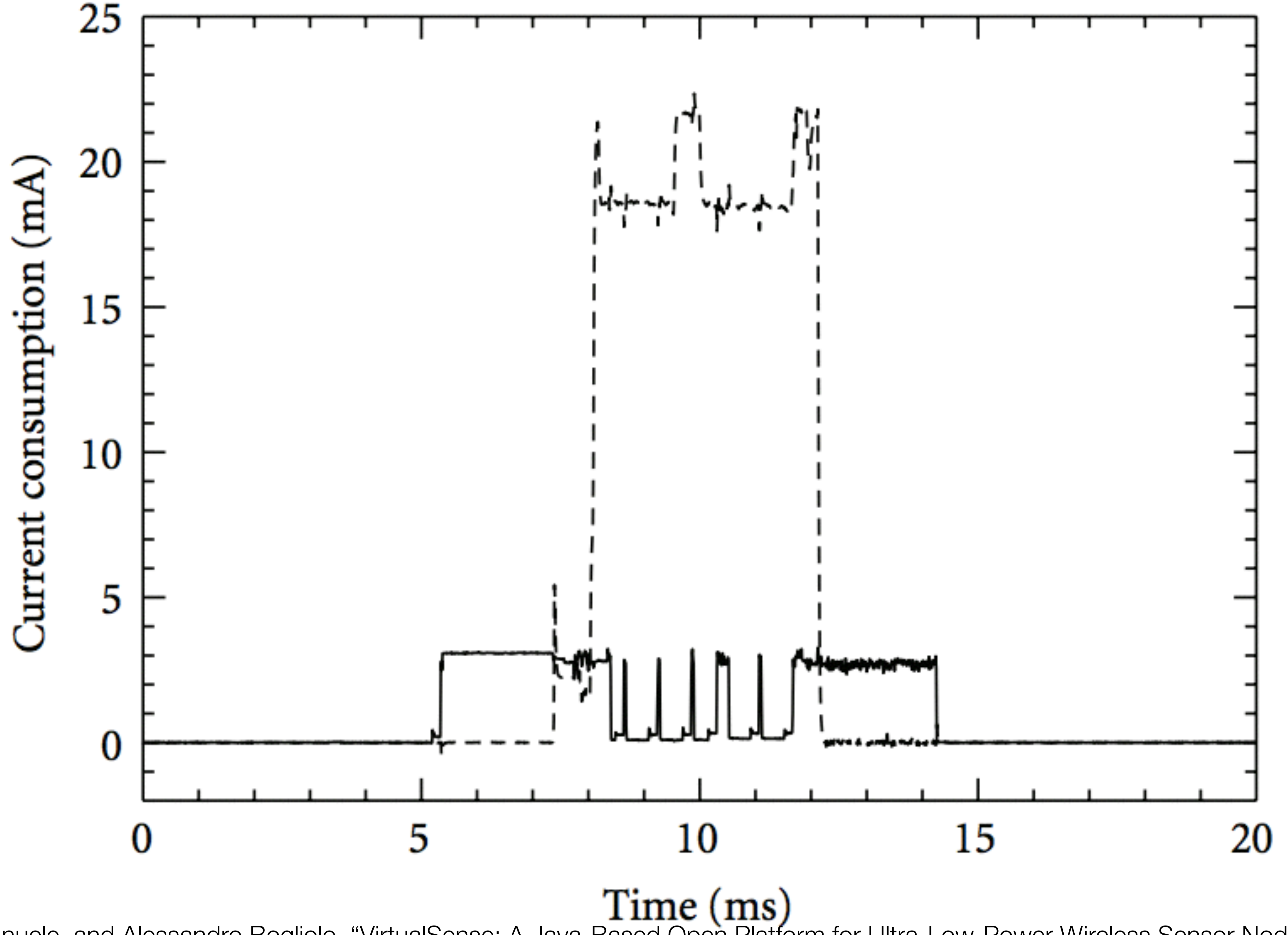
Error Rate	RTS-CTS-DATA	RTS-CTS-DATA-ACK
0	40.41	36.76
0.001	36.58	36.67
0.01	16.65	35.52
0.1	2.48	9.93

Table 4: The throughput, in packets per second, achieved by a single TCP data stream between a pad and a base station in the presence of noise.

P. Karn, "MACA - A new channel access method for packet radio", in Proceedings of the ARRL CRRL Amateur Radio 9th Computer Networking Conference, Redondo Beach, CA, Apr. 1990, pp. 134-140.

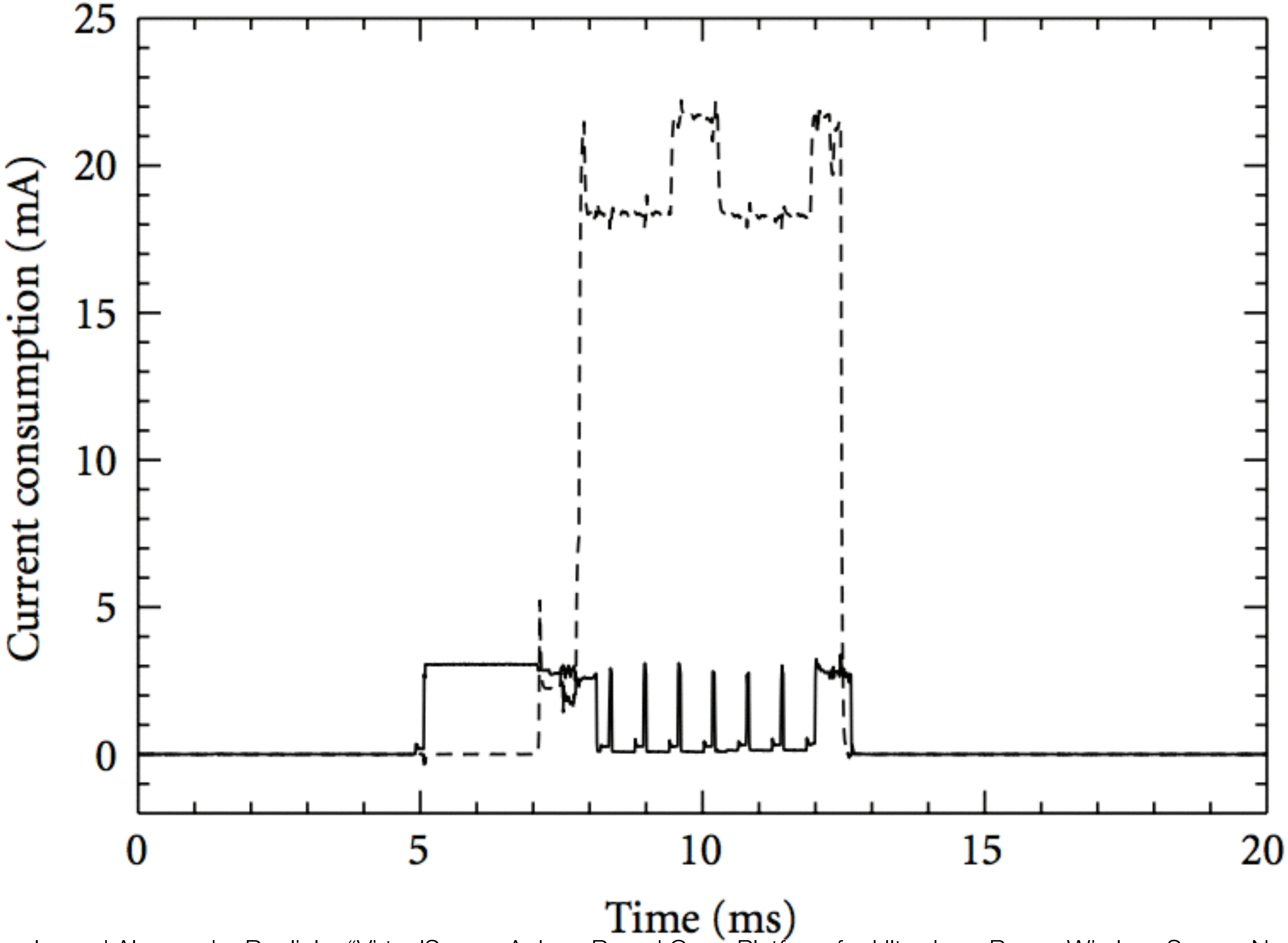
V. Bharghavan, A. Demers, S. Shenkar, and L. Zhang, "MACAW: A media access protocol for wireless LANs", in Proceedings of ACM SIGCOMM'94, London, UK, Sept. 1994, pp. 212-225.

Broadcast reception and processing



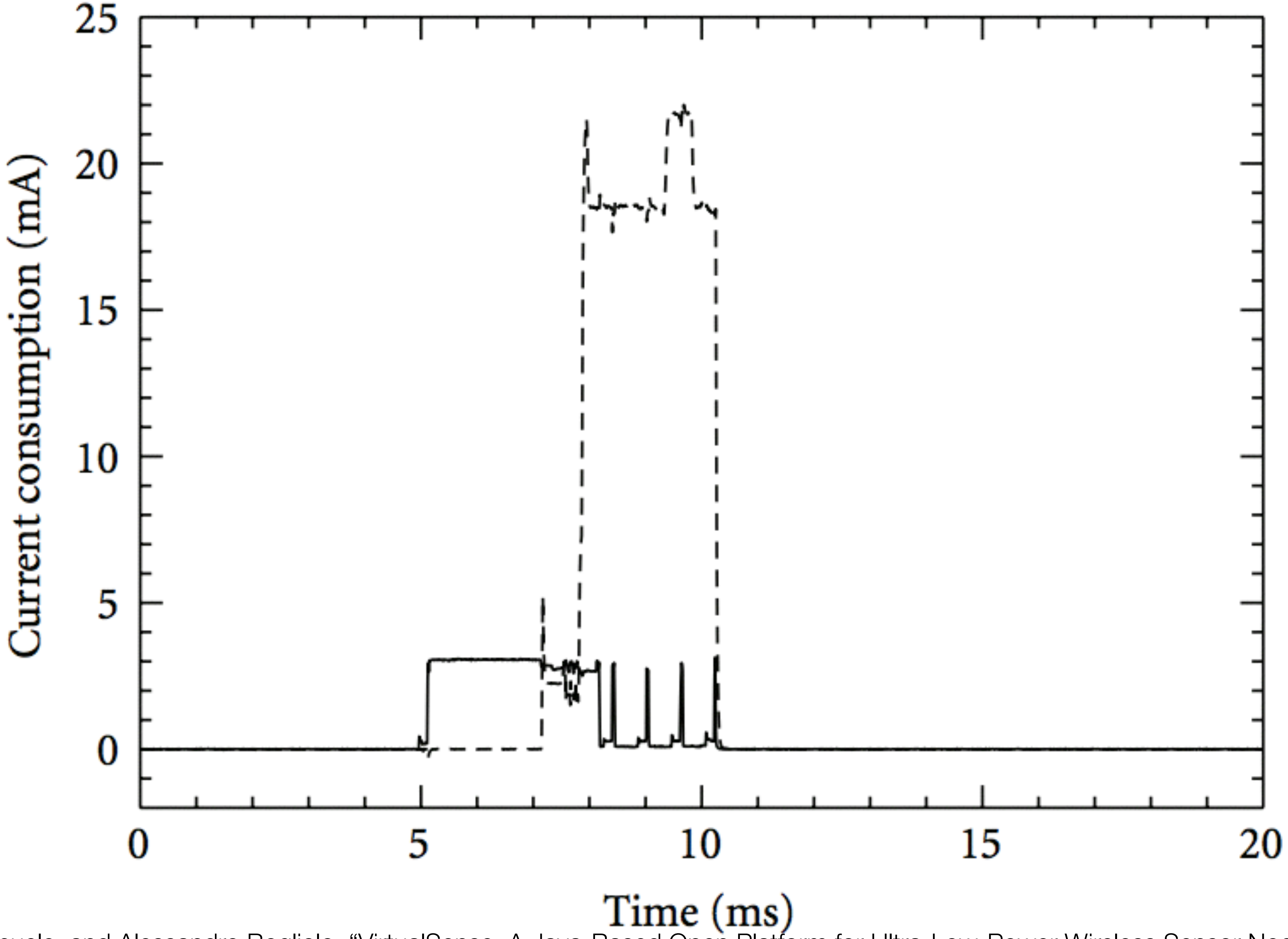
Lattanzi, Emanuele, and Alessandro Bogliolo. “VirtualSense: A Java-Based Open Platform for Ultra-Low-Power Wireless Sensor Nodes.” International Journal of Distributed Sensor Networks 2012 (2012)

Broadcast reception



Lattanzi, Emanuele, and Alessandro Bogliolo. “VirtualSense: A Java-Based Open Platform for Ultra-Low-Power Wireless Sensor Nodes.” International Journal of Distributed Sensor Networks 2012 (2012)

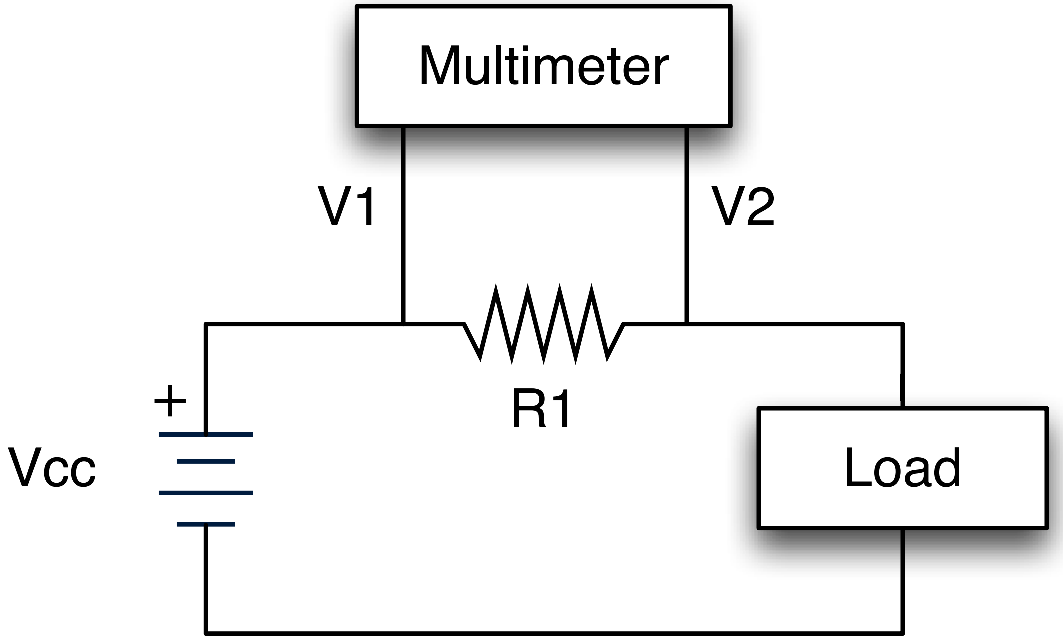
Frame filtering



Lattanzi, Emanuele, and Alessandro Bogliolo. “VirtualSense: A Java-Based Open Platform for Ultra-Low-Power Wireless Sensor Nodes.” International Journal of Distributed Sensor Networks 2012 (2012)

ENERGY CONSUMPTION MEASUREMENT

Shunt resistor



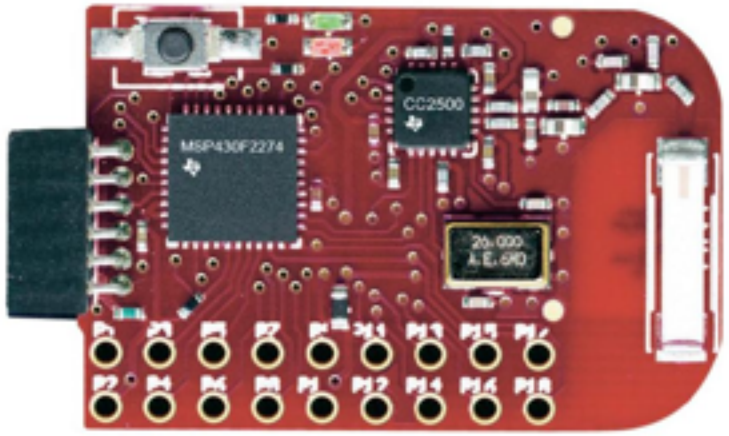
$$\hat{I} = \frac{\Delta V}{R_1} = \frac{V_2 - V_1}{R_1}$$

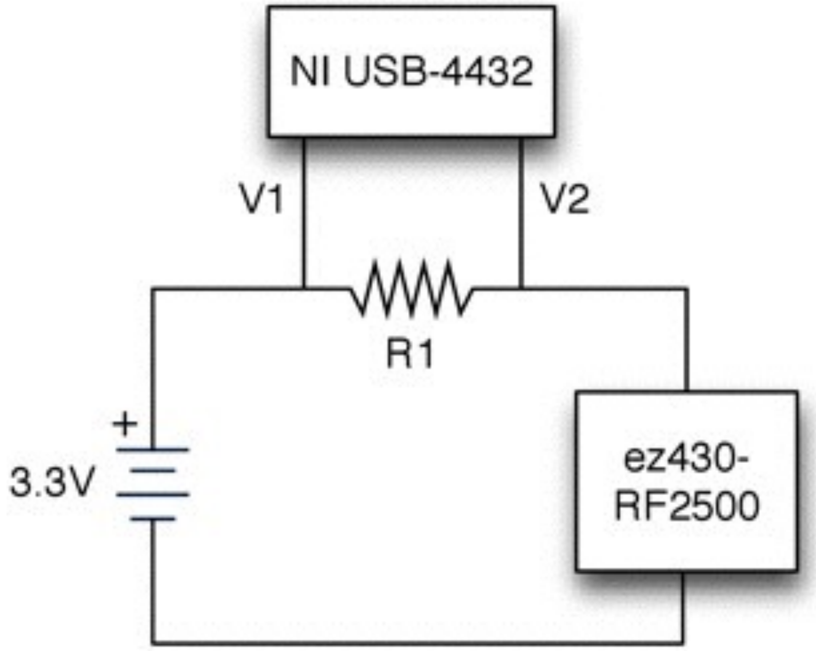
$$\hat{P} = \hat{I} V_0$$

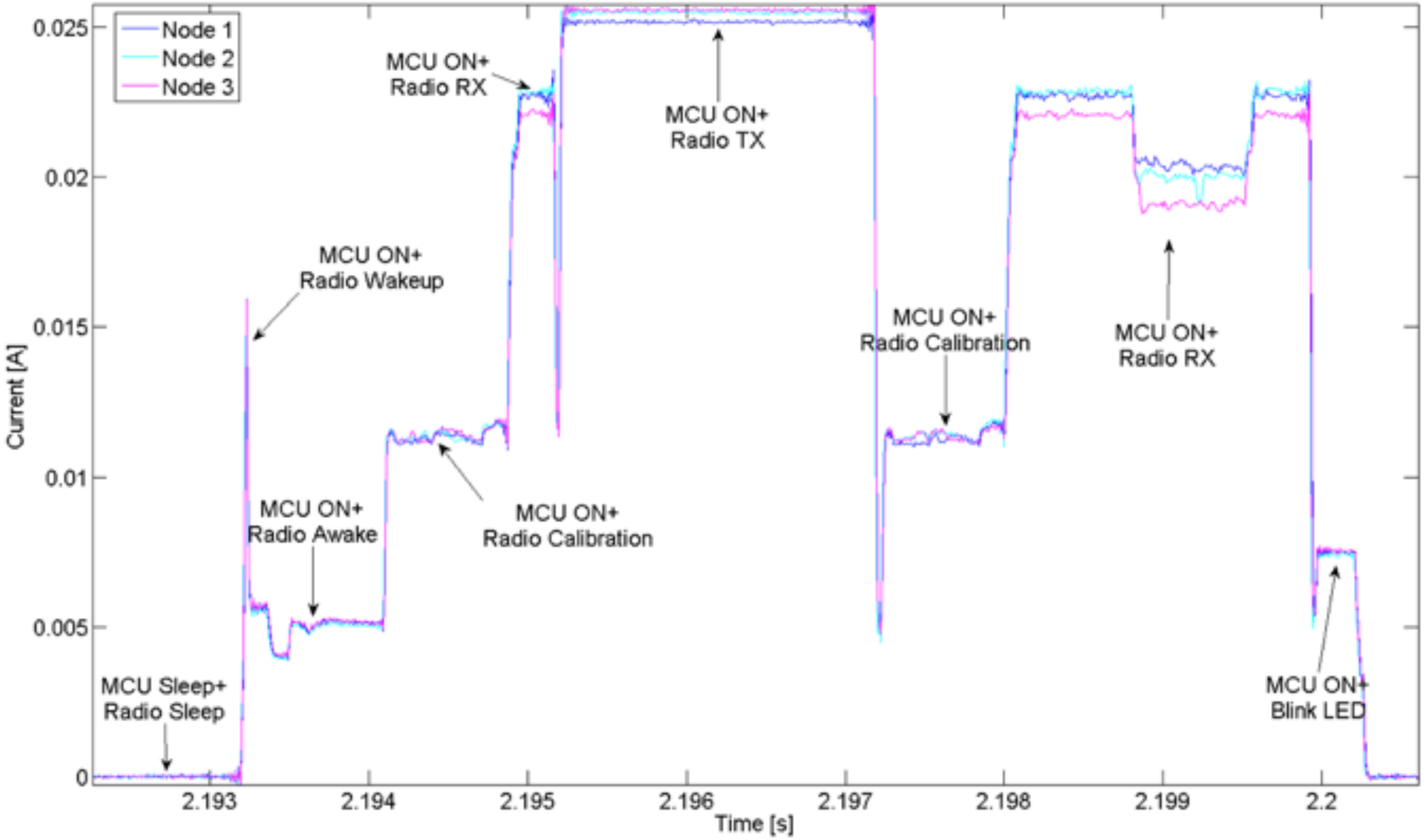
$$u(I) = \sqrt{u(\Delta V)^2 \frac{1}{R_1^2} + u(R_1)^2 \left(\frac{\Delta V}{R_1}\right)^2}$$

$$u(P) = \sqrt{u(\Delta V)^2 \left(\frac{V_0}{R_1}\right)^2 + u(R_1)^2 \left(\frac{V_0 \Delta V}{R_1^2}\right)^2 + u(V_0)^2 \left(\frac{\Delta V}{R_1}\right)^2}$$

ez430-RF2500 energy consumption

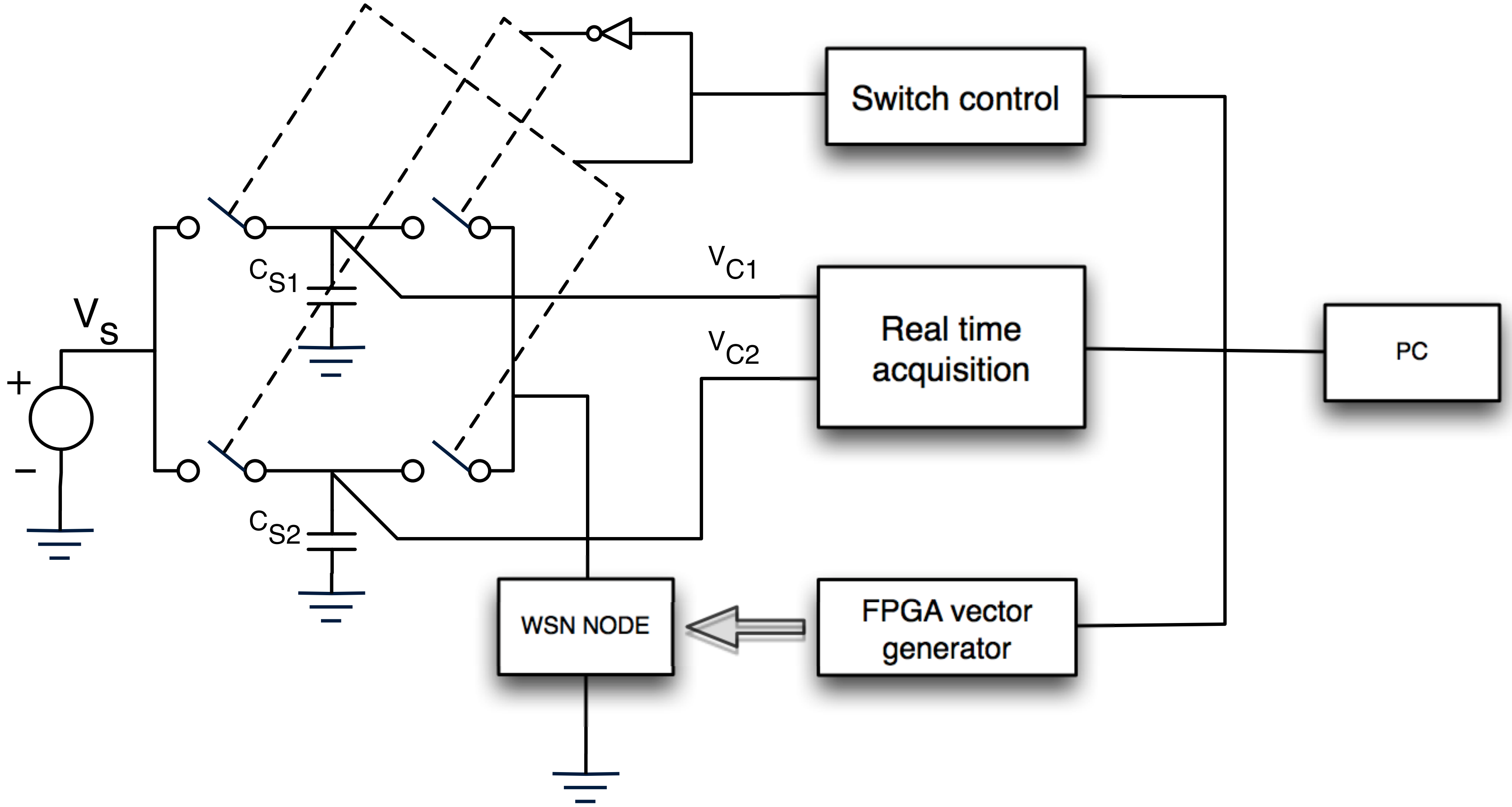






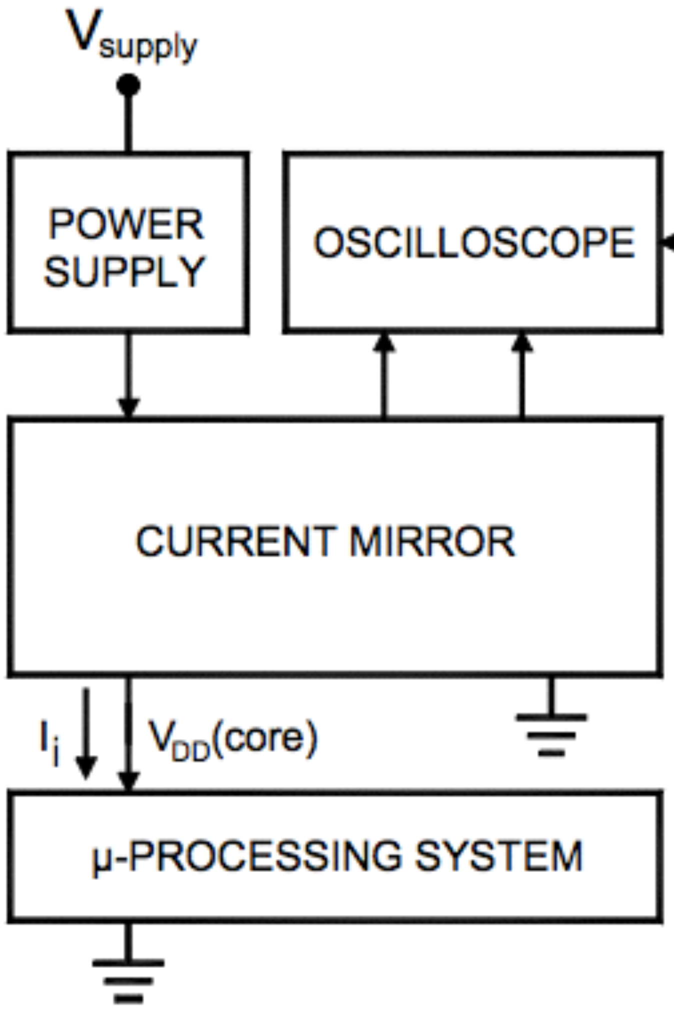
Power consumption assessment in wireless sensor networks. Antonio Moschitta and Igor Neri, ICT- Energy - Nanoscale Energy Management Concepts Towards Zero-Power Information and Communication Technology

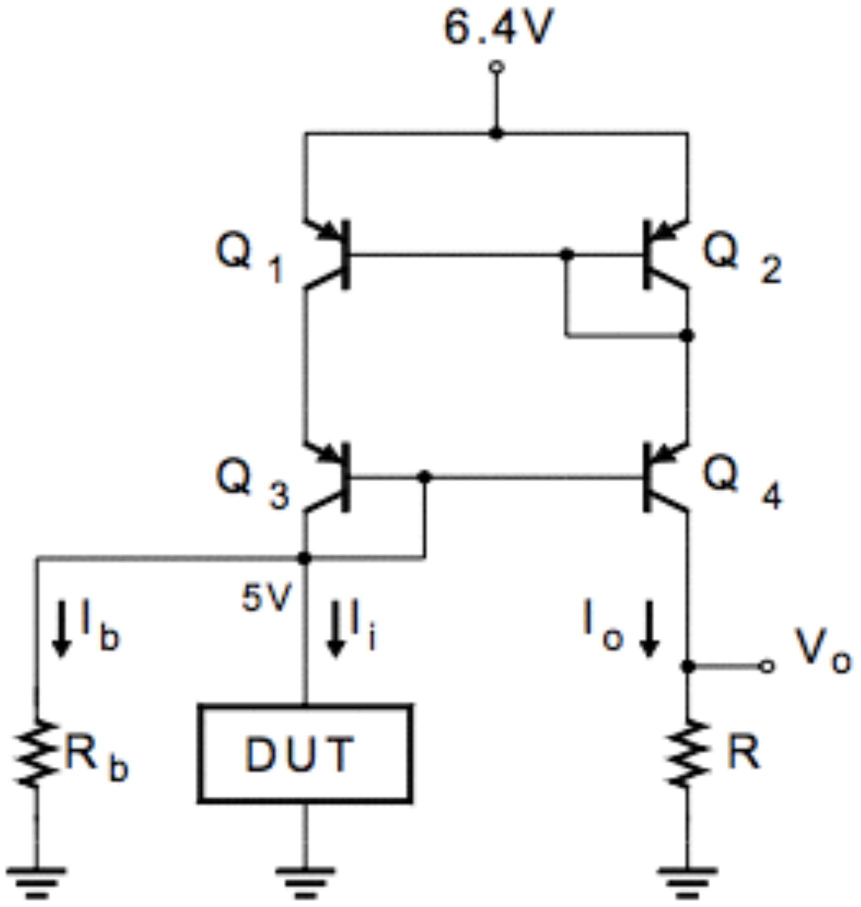
Switch capacitor



Chang, Naehyuck, Kwanho Kim, and Hyung Gyu Lee. “Cycle-accurate energy measurement and characterization with a case study of the ARM7TDMI [microprocessors].” Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 10.2 (2002): 146-154.

Current mirror

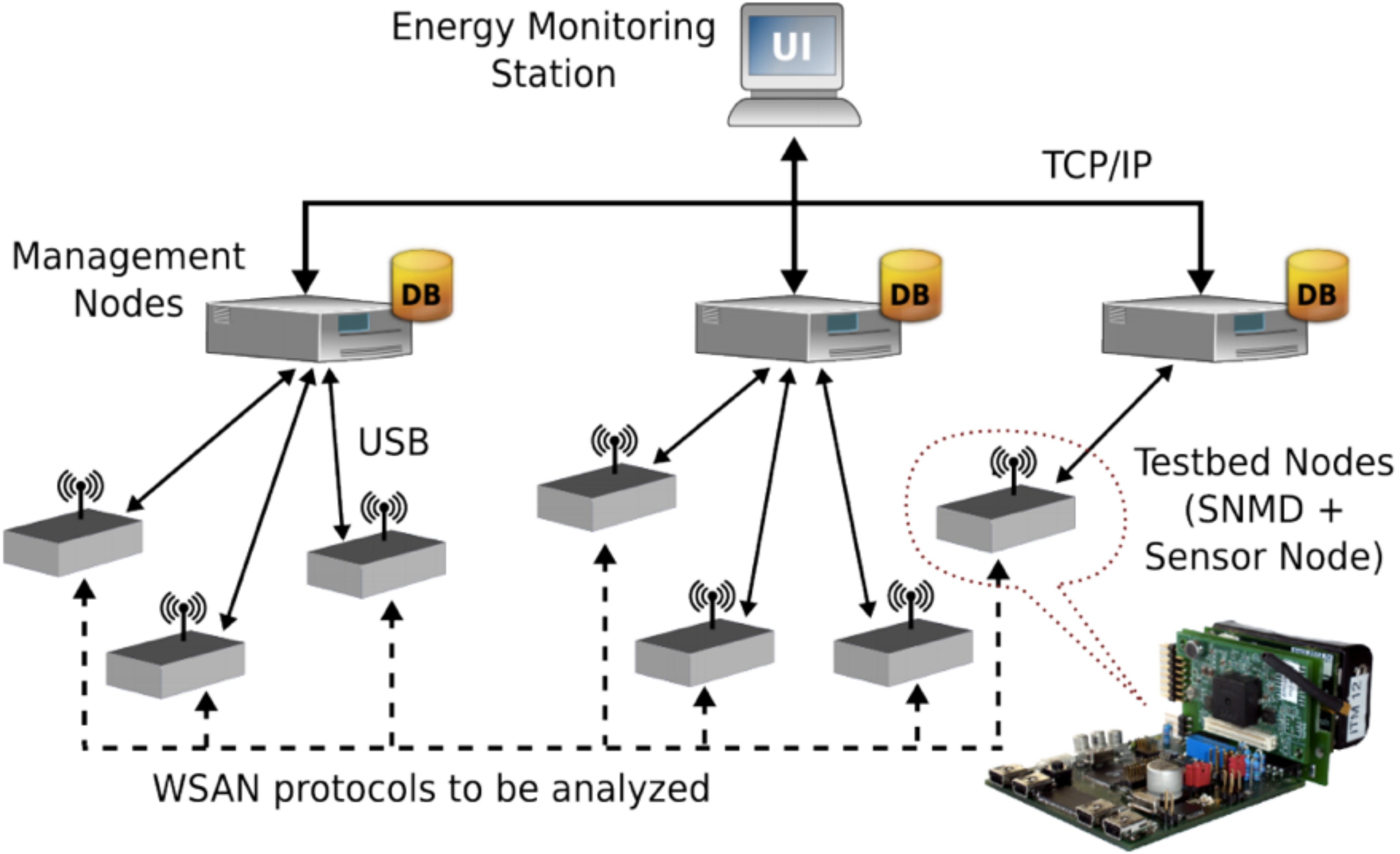




- Copying accuracy
- Time (frequency) response

T. Laopoulos, P. Neofotistos, C. A. Kosmatopoulos, and S. Nikolaidis," Measurement of Current Variations for the Estimation of Software-Related Power Consumption," IEEE Transactions on Instrumentation and Measurements, Vol. 52, no. 4, August 2003

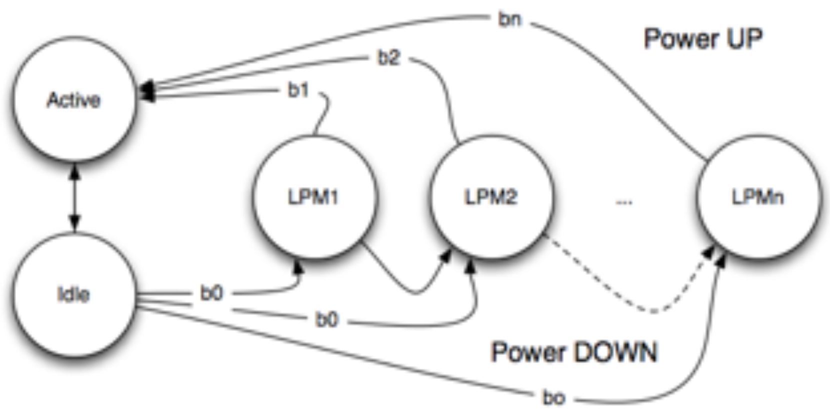
SANDbed: Distributed Energy Measurements


Energy Monitoring Station
UI
TCP/IP
Management Nodes
DB
DB
DB
USB
Testbed Nodes (SNMD + Sensor Node)
ITM 12
WSAN protocols to be analyzed

A. Hergenroder, J. Horneber, and J. Wilke. SANDbed: A WSAN Testbed for Network Management and Energy Monitoring. Hamburg, Germany, Aug. 2009. 8. GI/ITG KuVS Fachgesprach “Drahtlose Sensornetze”.

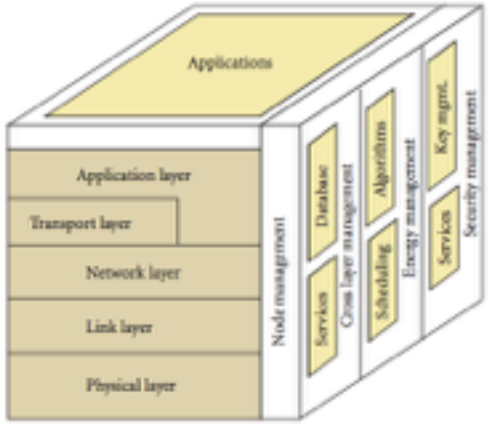
ENERGY CONSUMPTION MODELLING

Modelling strategies: Finite state machine



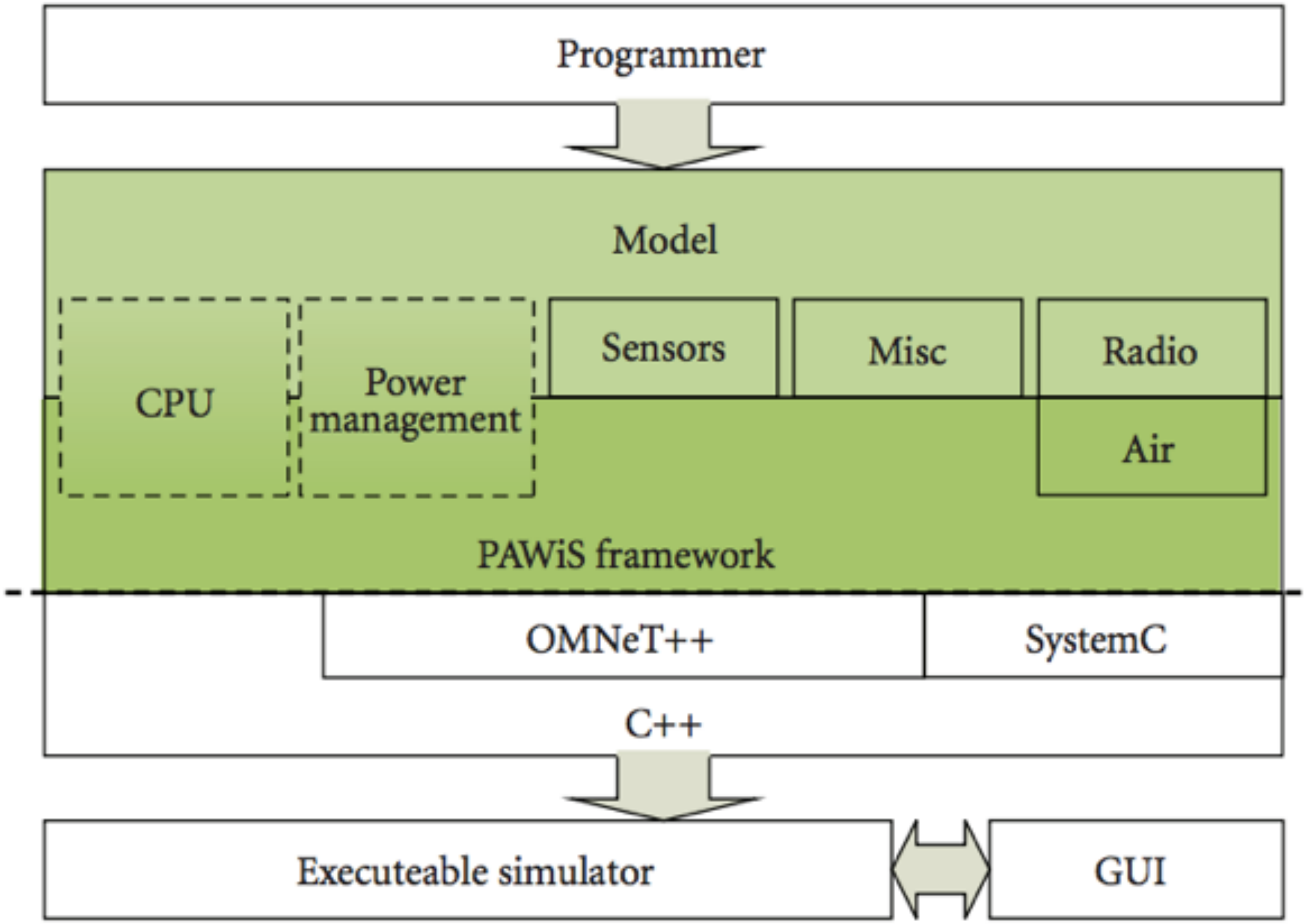
- **Pro**:
  - Easy to implement
  - Easy to simulate
- **Cons**:
  - Rough estimate of energy consumption
  - Complex to address lot of peripherals

Modelling strategies: Network focused simulation frameworks



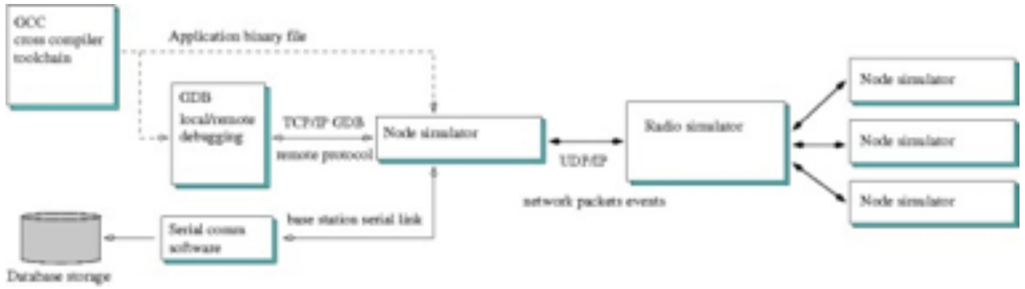
- **Pro**:
  - Fast simulation time
  - Includes network and MAC layers
- **Cons**:
  - Coarse representation of node state
  - Inaccurate energy consumption estimate

Modelling strategies: Network focused simulation frameworks - PAWiS



Johann, Glaser, et al. "Power aware simulation framework for wireless sensor networks and nodes." EURASIP Journal on Embedded Systems 2008 (2008).

Modelling strategies: Instruction-level simulators



- **Pro**:
  - Accurate energy consumption estimation
  - Tracking of node and peripheral states
  - Fine-grained timing
- **Cons**:
  - Strictly dependant on the platform
  - Need of accurate calibration
  - Simulation time can be long

Avrora

- Cycle accurate execution times.
- Online monitoring of program behaviour.
- The profiling utilities allow users to study their program's behavior in simulation.
- Detailed observation of program behavior without disturbing the simulation, and without modifying the simulator source code.
- The GDB debugger hooks allow source-level debugging and integrated development and testing.
- Graphical representation program's instructions that is useful for understanding how it is structured and what the compiler does with your code.
- The energy analysis tool can analyze energy consumption.
- The stack checker tool can be used to bound the maximum stack size used by your program.

http://compilers.cs.ucla.edu/avrora/

Worldsens Framework

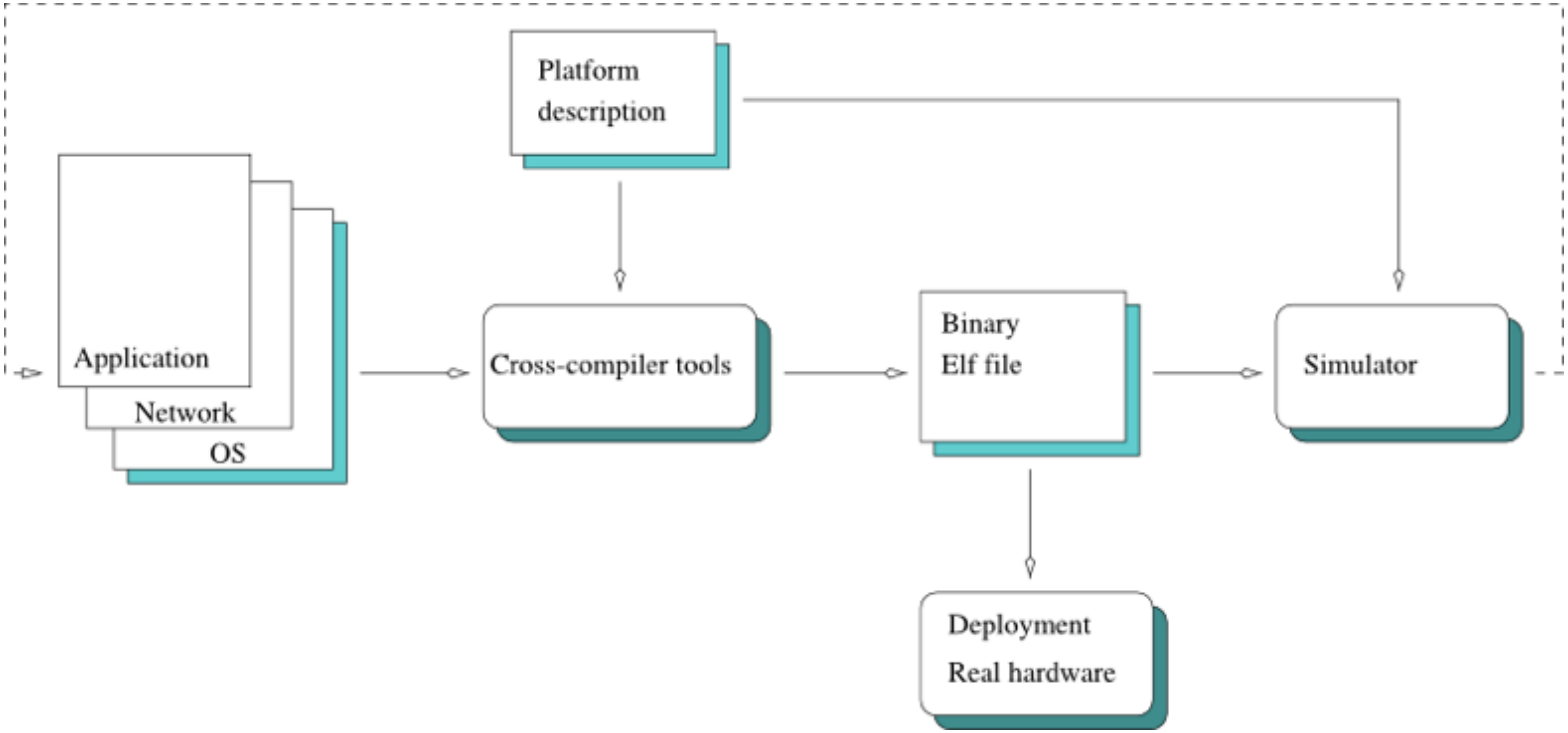
- **WSim**: node instruction-level and peripherals simulator
- **WSNet**: event based network simulator
- **eSimu**: energy consumption analysis and estimate

http://wsim.gforge.inria.fr/

Worldsens Framework: WSim

WSim is a full platform simulator that can run the target platform object code without modification

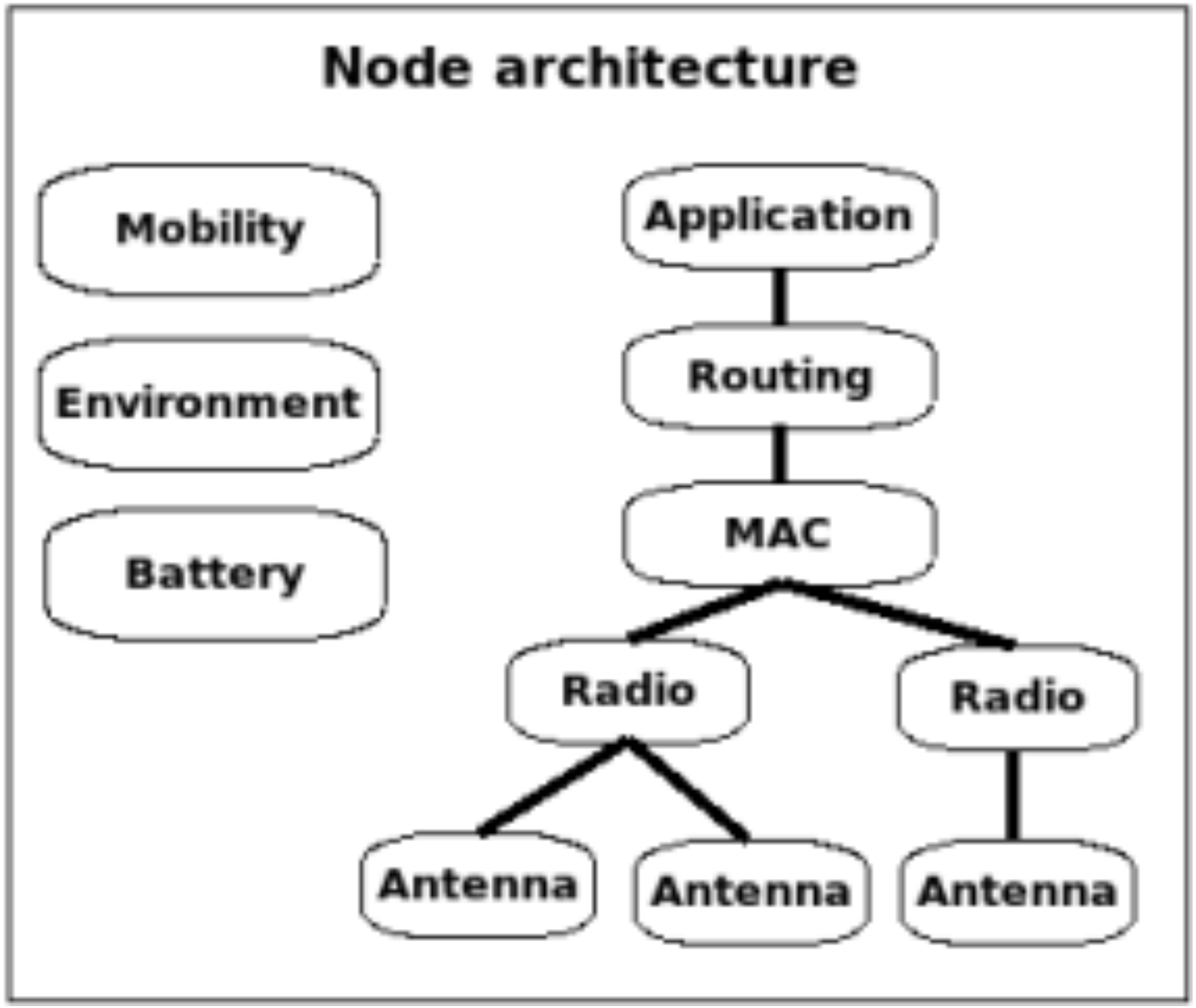
- **debugging**, **profiling** and **performance evaluation**



Worldsens Framework: WSNet

WSNet is an event-driven simulator for wireless networks

- mobility
- energy source
- routing protocols
- mac protocols
- radio interface
- antenna

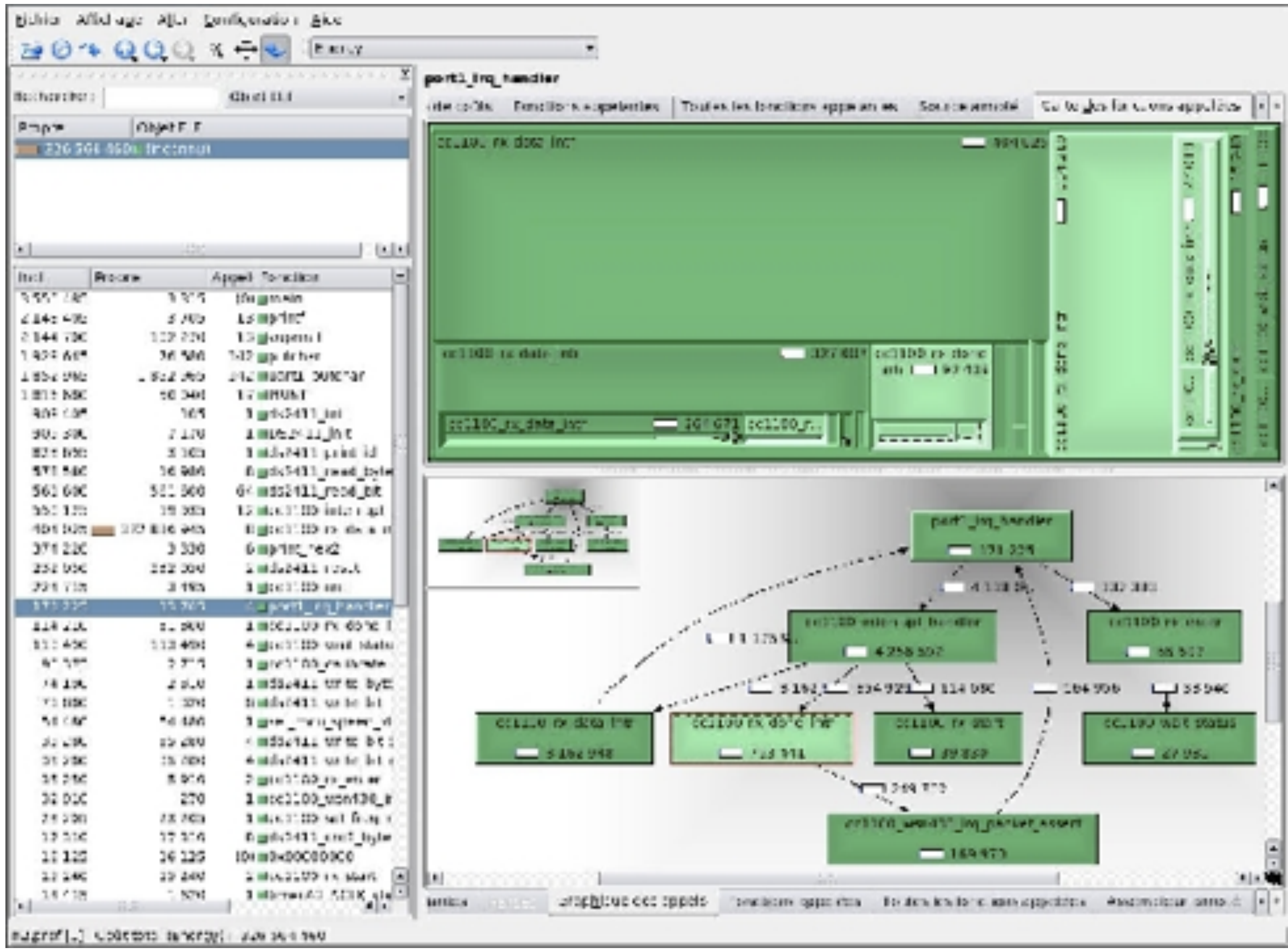


Worldsens Framework: eSimu

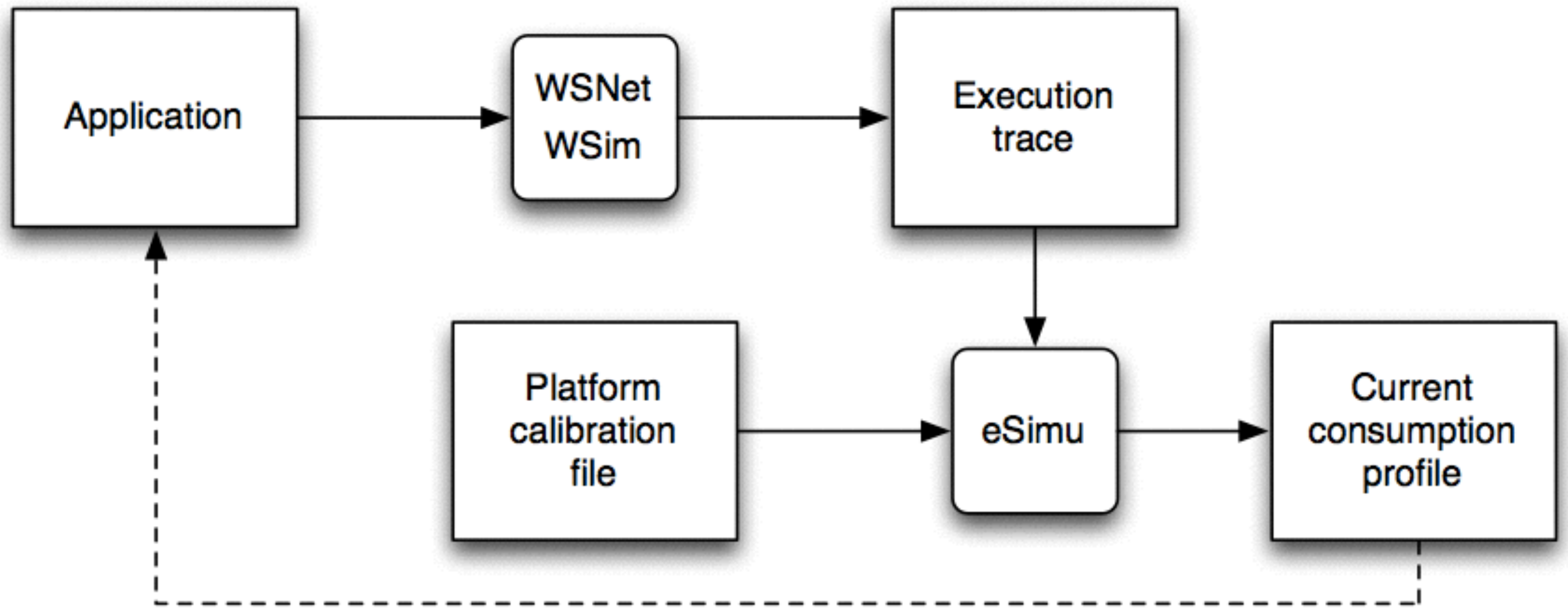
eSimu is a complete system energy model based on non-intrusive measurements

- cycle accurate simulation tools to give energy consumption feedback for embedded systems software programming
- whole system consumption including peripherals

$$\xi_{\text{slot}} = \xi_{\text{CPU}} + \sum_{\text{blocks}} \xi_{\text{bl}}$$



Worldsens Framework: workflow



Application

WSNet
WSim

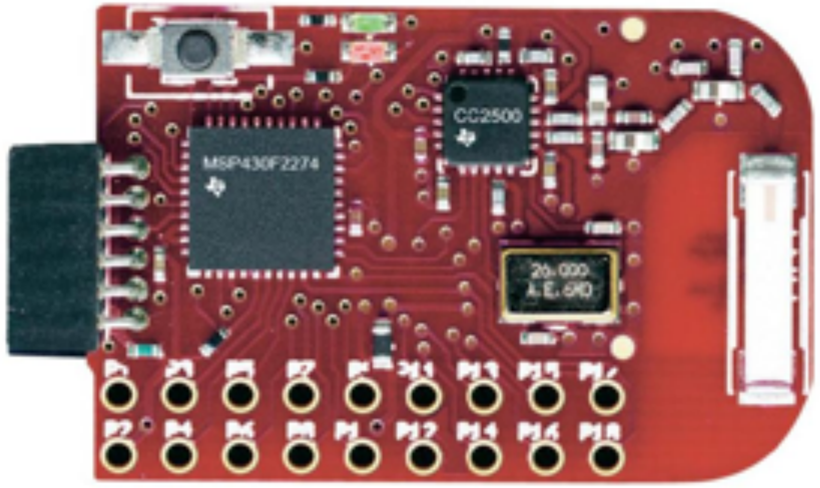
Execution trace

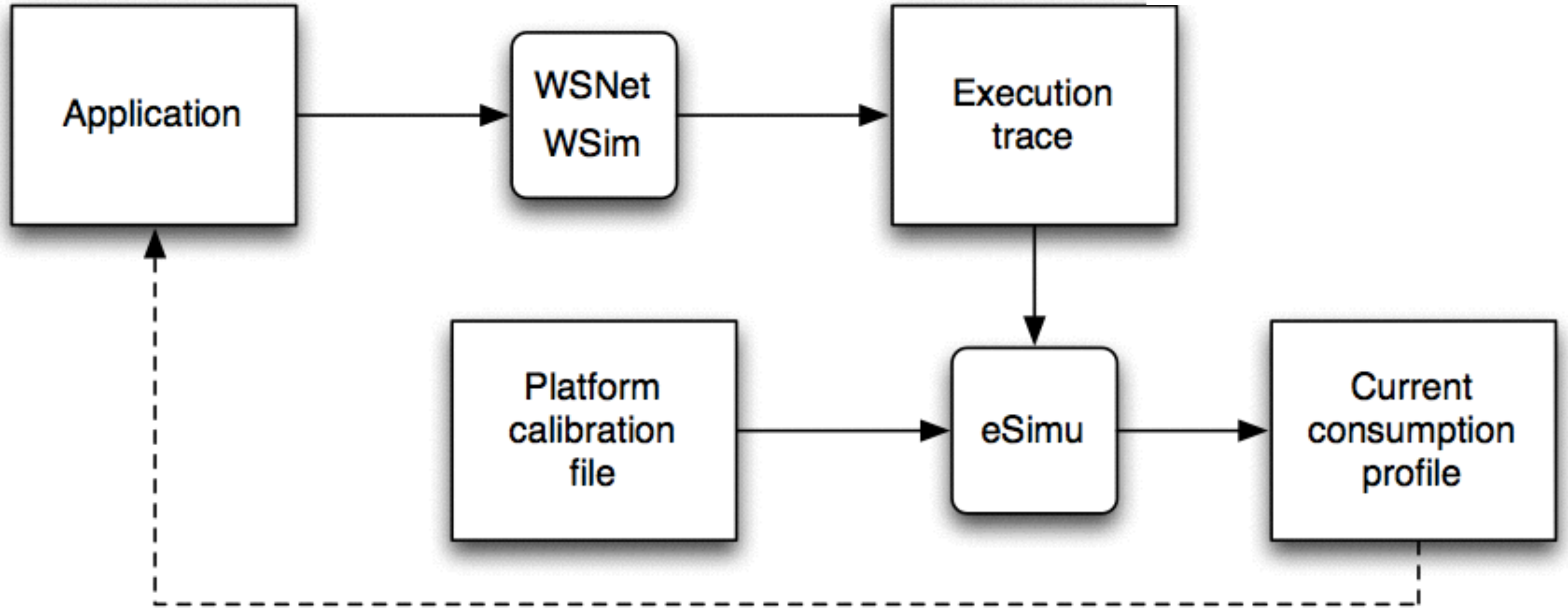
Platform calibration file

eSimu

Current consumption profile

Worldsens Framework: ez430-RF2500





Application

WSNet
WSim

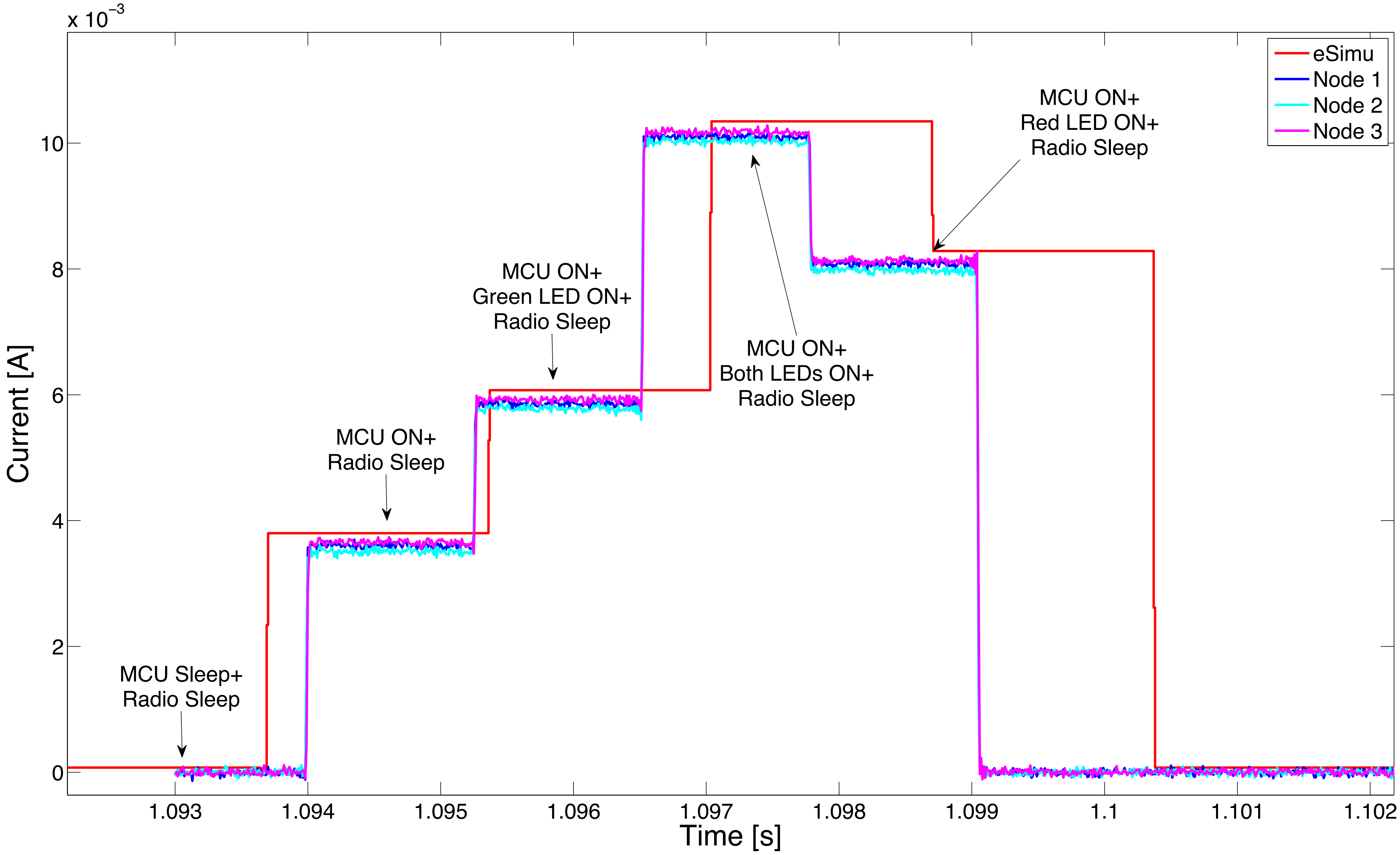
Execution trace

Platform calibration file

eSimu

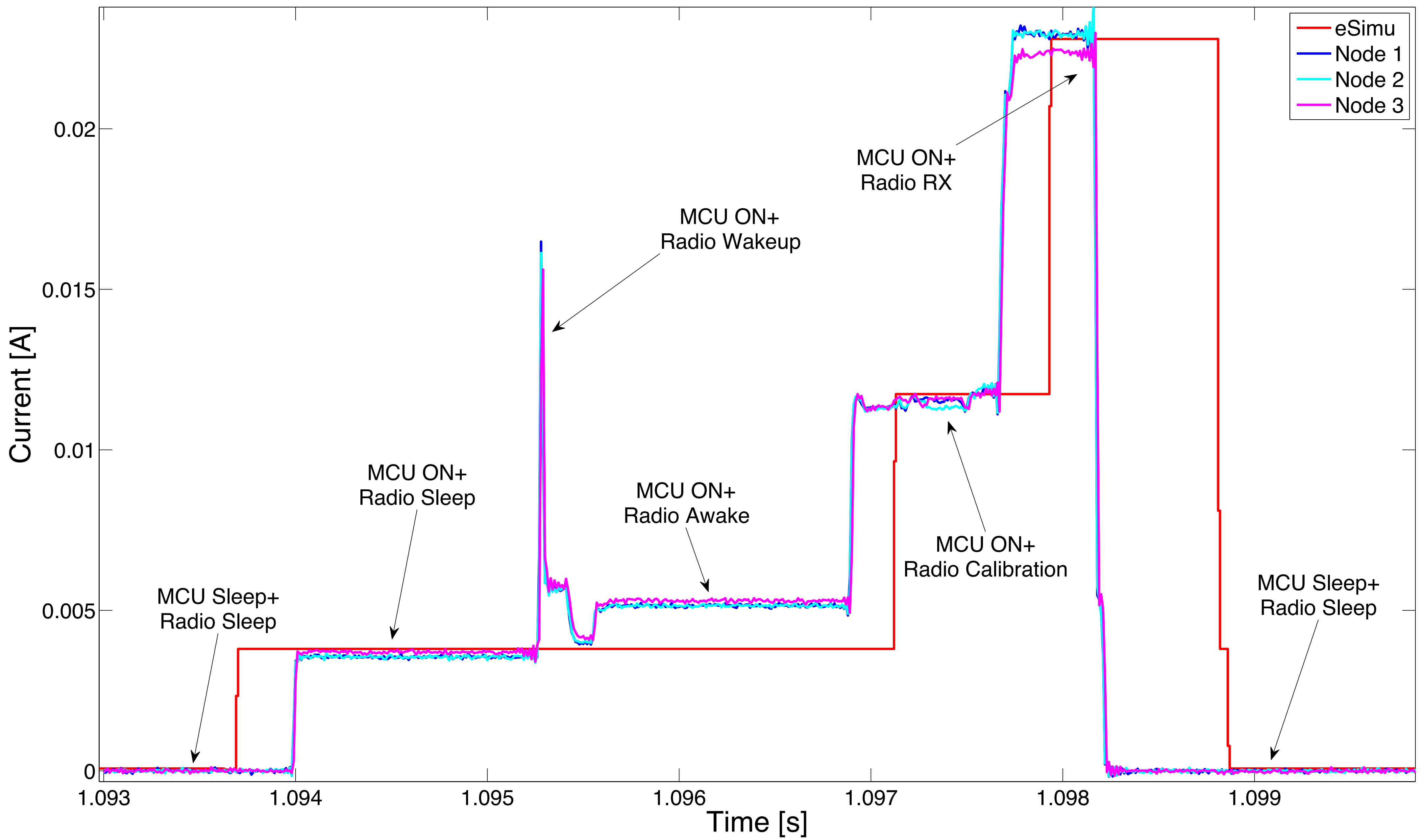
Current consumption profile

ez430-RF2500 current consumption: measurements and simulations: LEDs



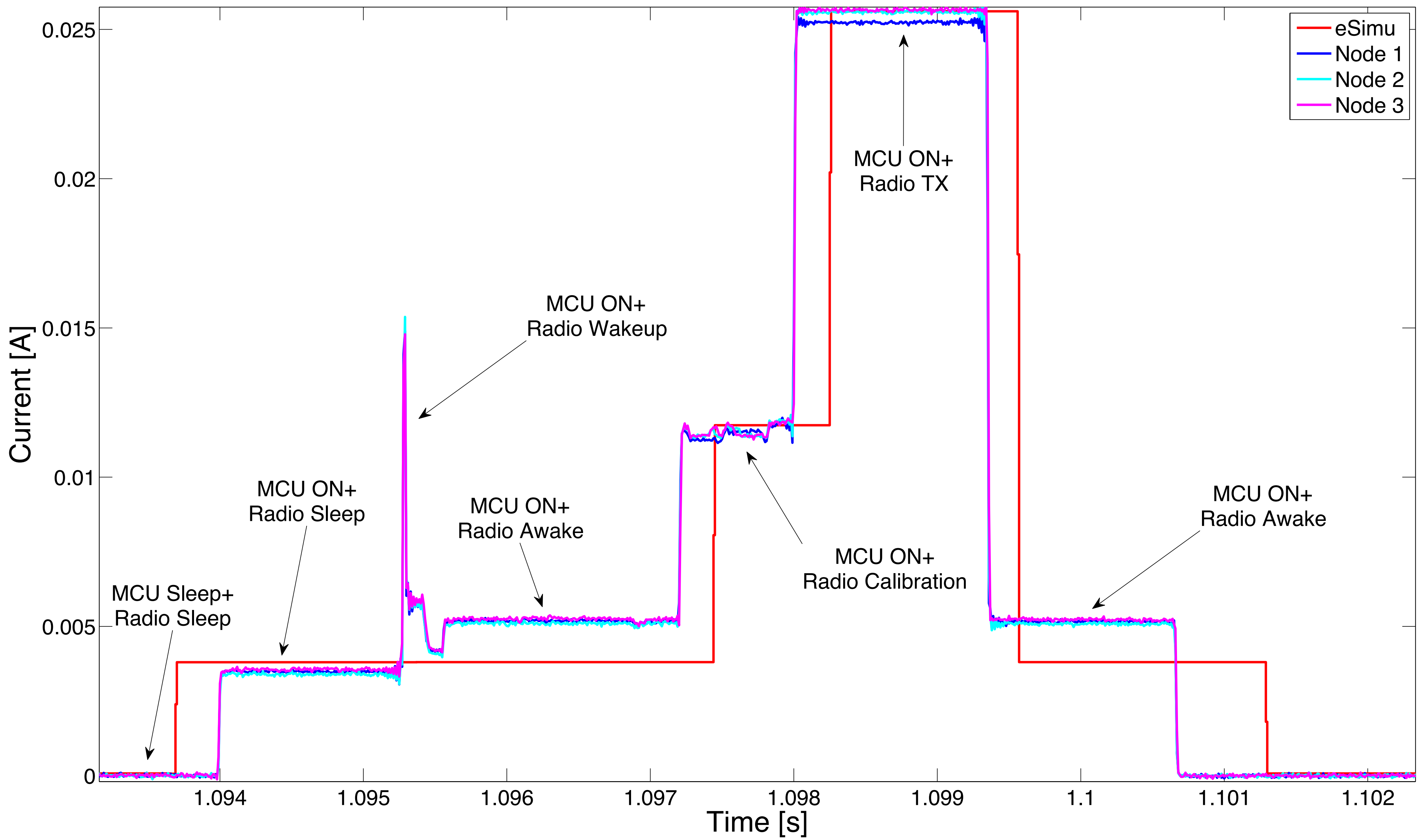
Power consumption assessment in wireless sensor networks. Antonio Moschitta and Igor Neri, ICT- Energy - Nanoscale Energy Management Concepts Towards Zero-Power Information and Communication Technology

ez430-RF2500 current consumption: measurements and simulations: RX



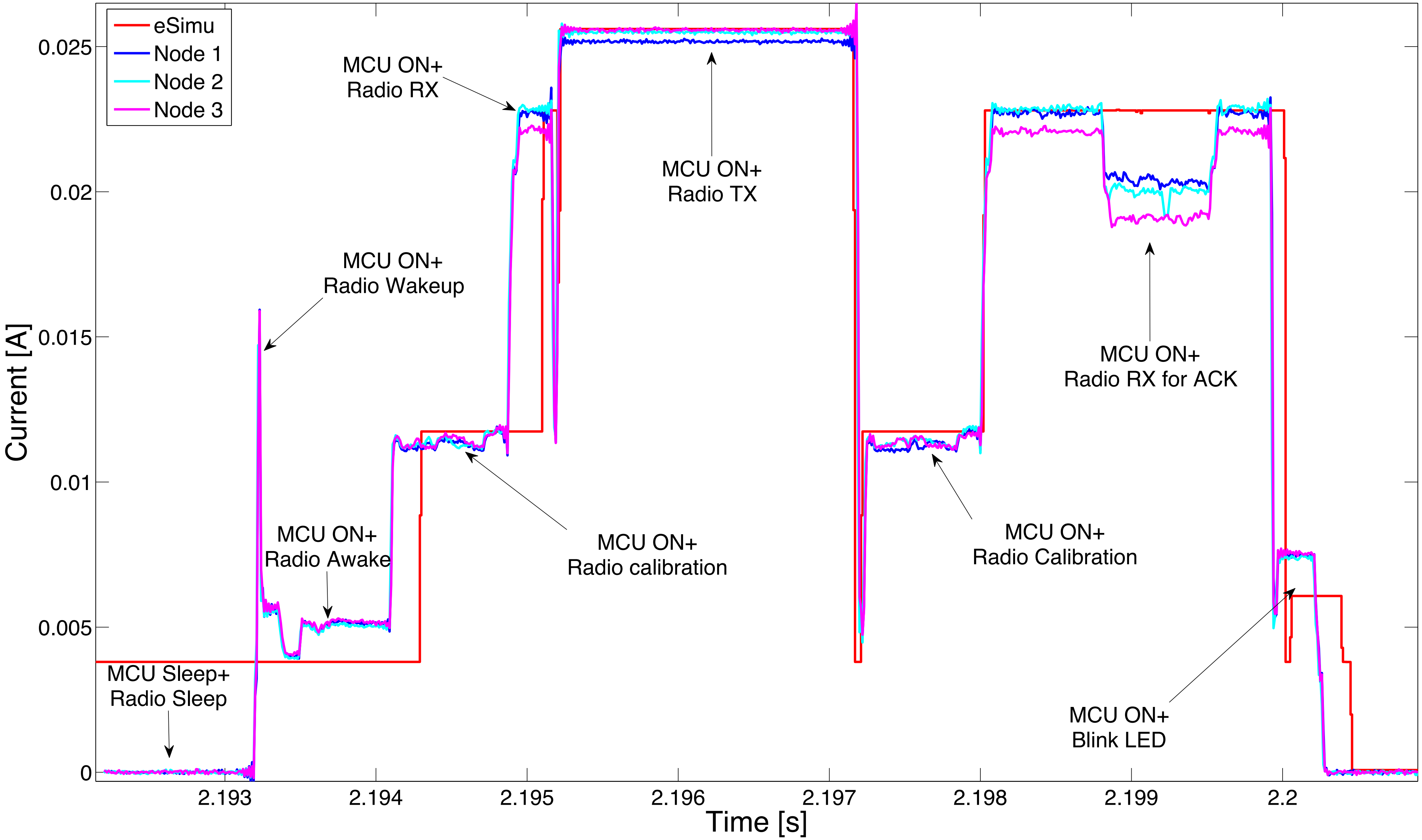
Power consumption assessment in wireless sensor networks. Antonio Moschitta and Igor Neri, ICT- Energy - Nanoscale Energy Management Concepts Towards Zero-Power Information and Communication Technology

ez430-RF2500 current consumption: measurements and simulations: TX



Power consumption assessment in wireless sensor networks. Antonio Moschitta and Igor Neri, ICT- Energy - Nanoscale Energy Management Concepts Towards Zero-Power Information and Communication Technology

ez430-RF2500 current consumption: measurements and simulations: TX with ACK

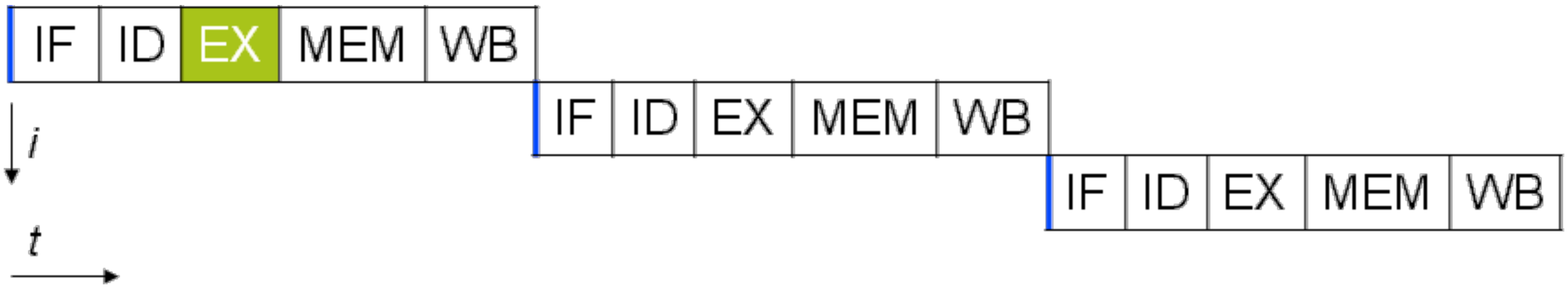


Power consumption assessment in wireless sensor networks. Antonio Moschitta and Igor Neri, ICT- Energy - Nanoscale Energy Management Concepts Towards Zero-Power Information and Communication Technology

Computer architecture: performances vs energy efficiency

Pipeline

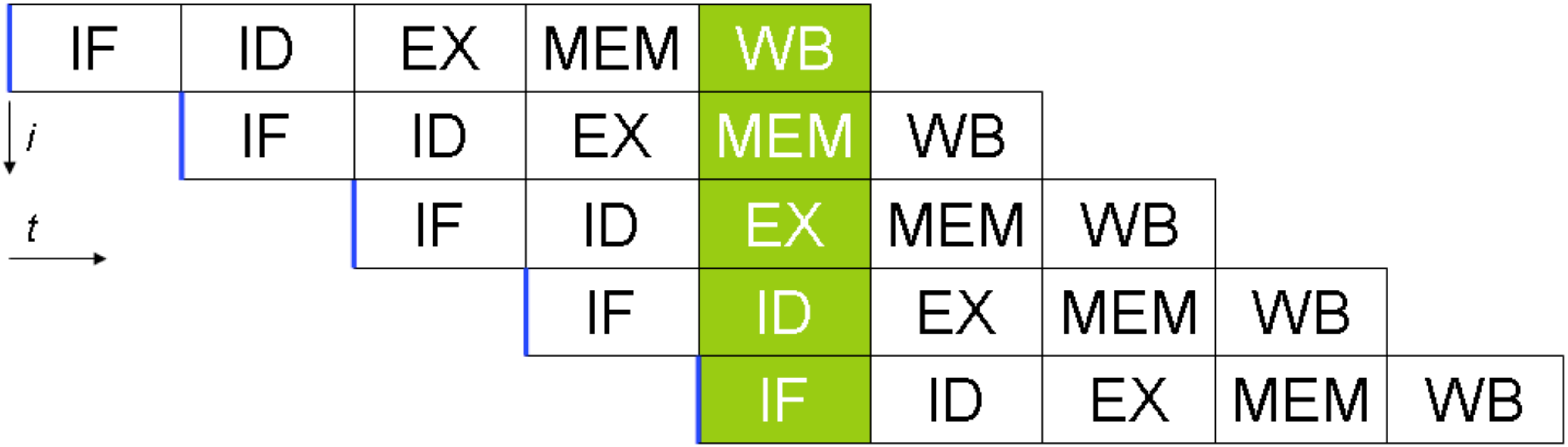
- Programmer assumes sequential execution of each instruction
- Instruction execution: sequential use of proc. logic
    - Instruction fetch
    - Instruction decode and register fetch
    - Execute
    - Memory access
    - Register write back
- When a given structure is used, the others are idle
- If instruction must complete before executing the next one, the resources of the processor are under-utilised



Assembly line

Pipeline

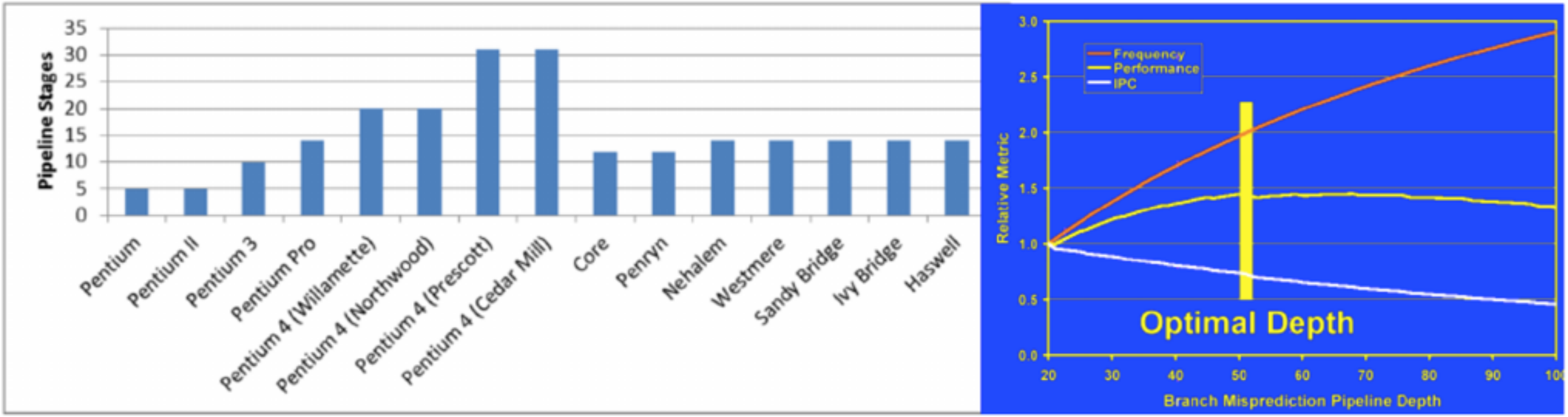
- Divide execution in several stages (pipeline)
- Instructions progress through the pipeline
- So overlapped execution of several instructions



IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB

Superpipelining: more stages

- Superpipelining: more stages. Higher frequency.
- Several fetch stages, etc. Pentium 4: 20 stages (31 stages in Prescott)
- Drawbacks: latch overhead, tight loops
- Superscalar: process several instructions per cycle.
- Drawbacks: more complex logic, hazards, bypasses

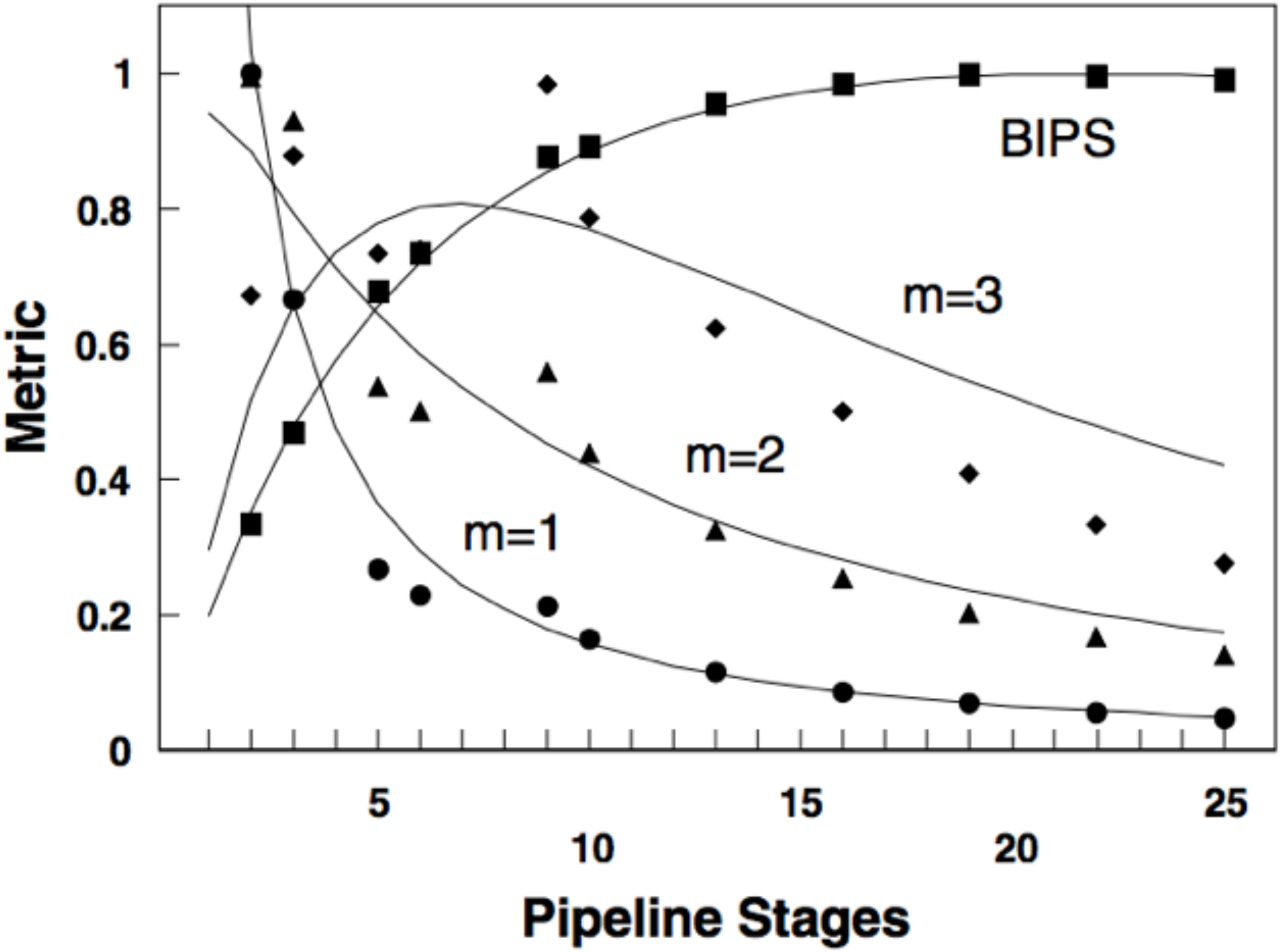


"Runtime Aware Architectures", Mateo Valero, HiPEAC CSW 2014, Barcelona.

Pipeline: performances vs energy

- Depends on the metric
- BIPS/W
- $BIPS^2/W$
- $BIPS^3/W$

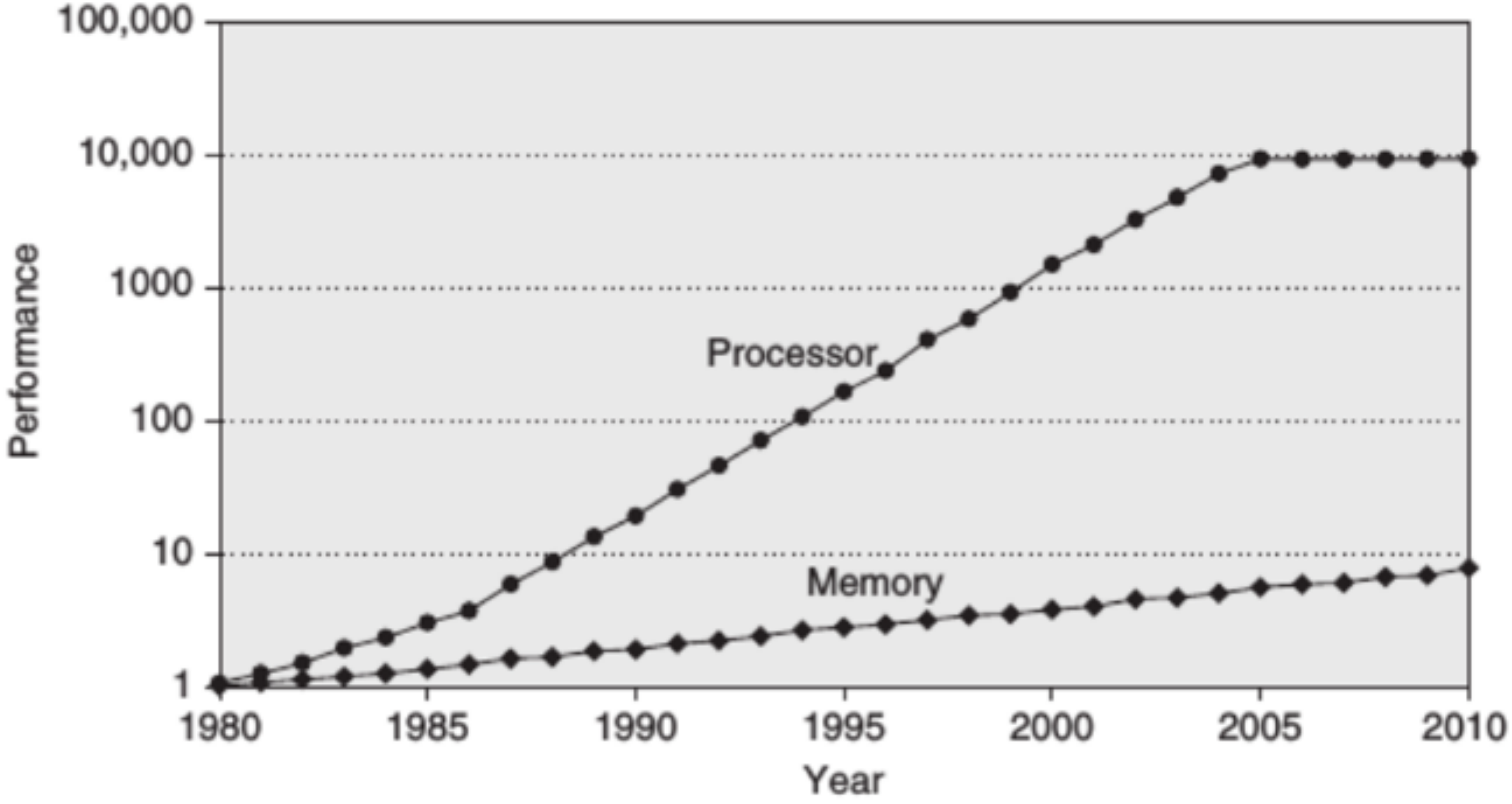
BIPS (Billion Instructions/ sec)



Hartstein, Allan, and Thomas R. Puzak. "Optimum power/performance pipeline depth." Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture. IEEE Computer Society, 2003.

Memory issue

Memory issue



"Computer Architecture: A Quantitative Approach", H. and P., 5th Ed, 2012.

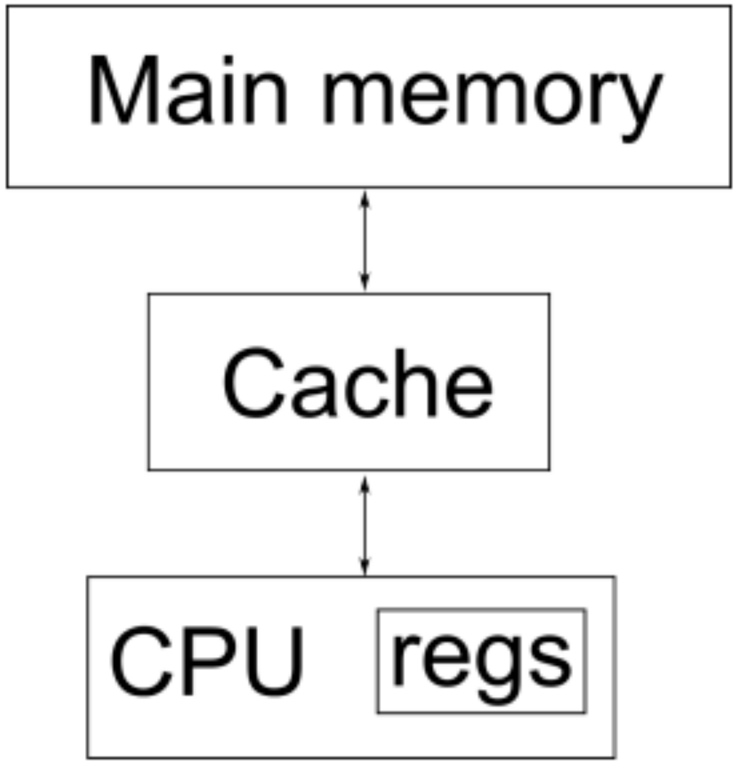
- Increasing gap in latency, many cycles to do an access
- T-cycle$_{cpu}$ << T-access$_{mem}$
- We want large memories, which have higher latencies

Memory: locality

- Memory instructions amongst the most used
- But some @ are accessed more
- For instance, inst in a loop
- This property is called locality
- Temporal locality: reuse data
  - After accessing @x, it is likely to access @x again soon
- Spatial locality: use neighbor data
  - After accessing @x, it is likely to access @x+1 soon
- 90/10 rule of thumb: 90% of accesses to 10% of @

Memory: exploiting locality

- Put most accessed data in fast (but small) SRAM (cache)
- Place the rest in large (but slow) DRAM (main memory)
- Detection of most accessed data? Locality
- Temp loc. -> on first access to @ copy data to cache
- Spatial locality -> store also neighbour data
- Memory is divided in blocks of consecutive words

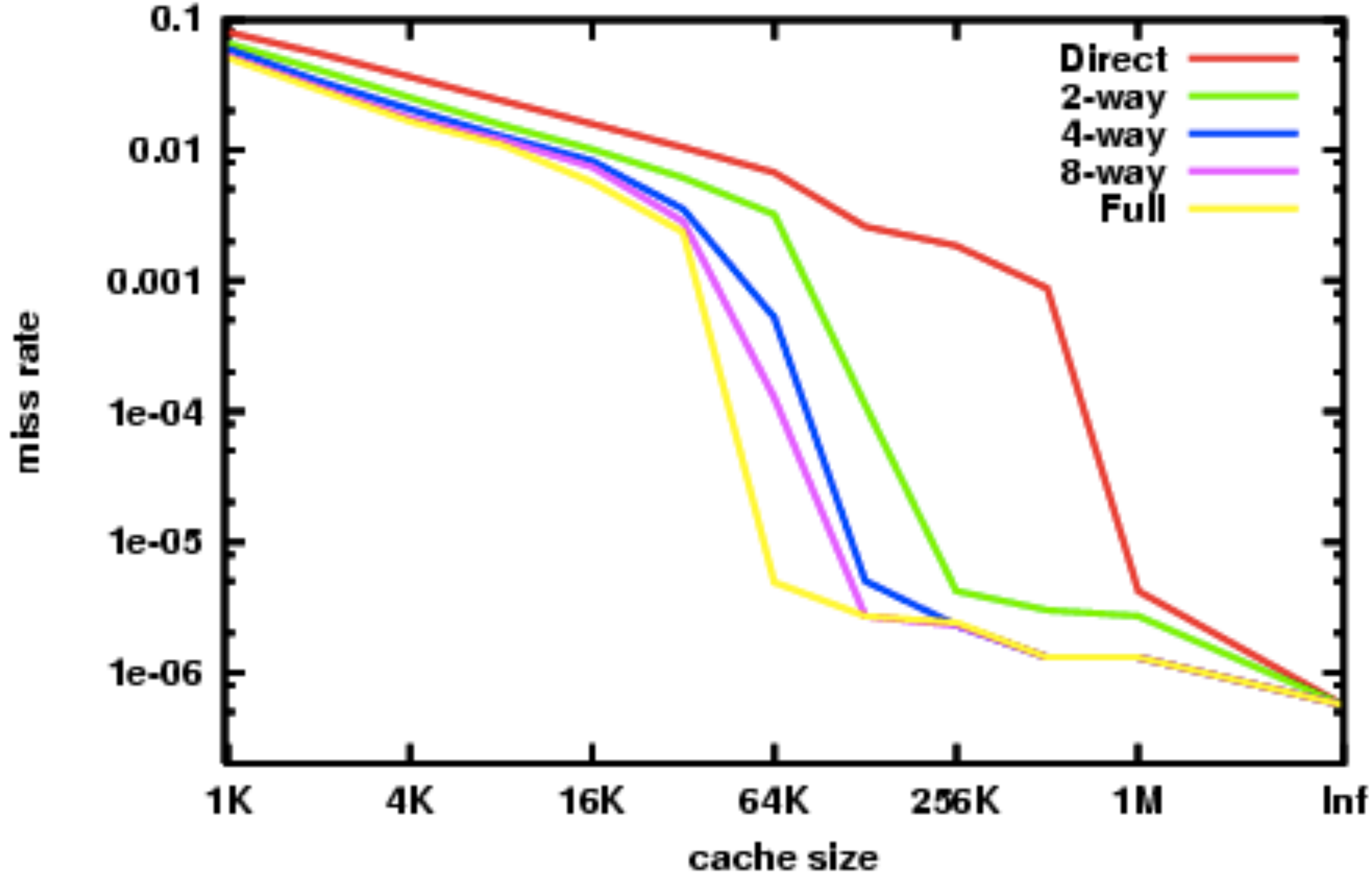


Cache

- Two possibilities when accessing the cache
  - Desired block is in cache -> Cache Hit: read data
  - Otherwise -> Cache miss: bring it from next level

- MissRate = Misses/Accesses: as low as possible
- Different techniques depending on miss class
- Larger cache to store more blocks
- Pre-fetch: access blocks before actually needed (SW/ HW)
- Associativity: allow placing block in different lines

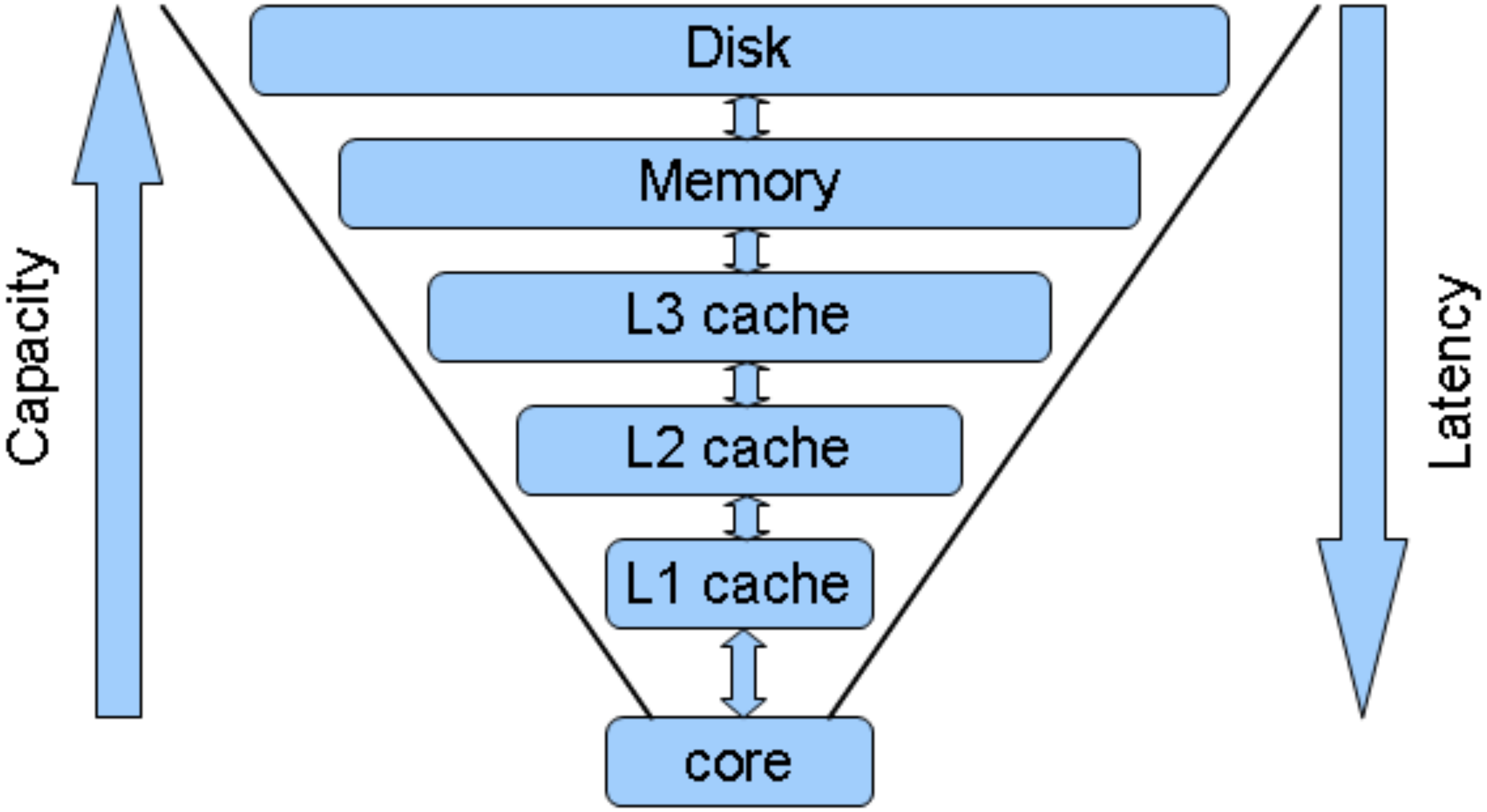
Cache miss



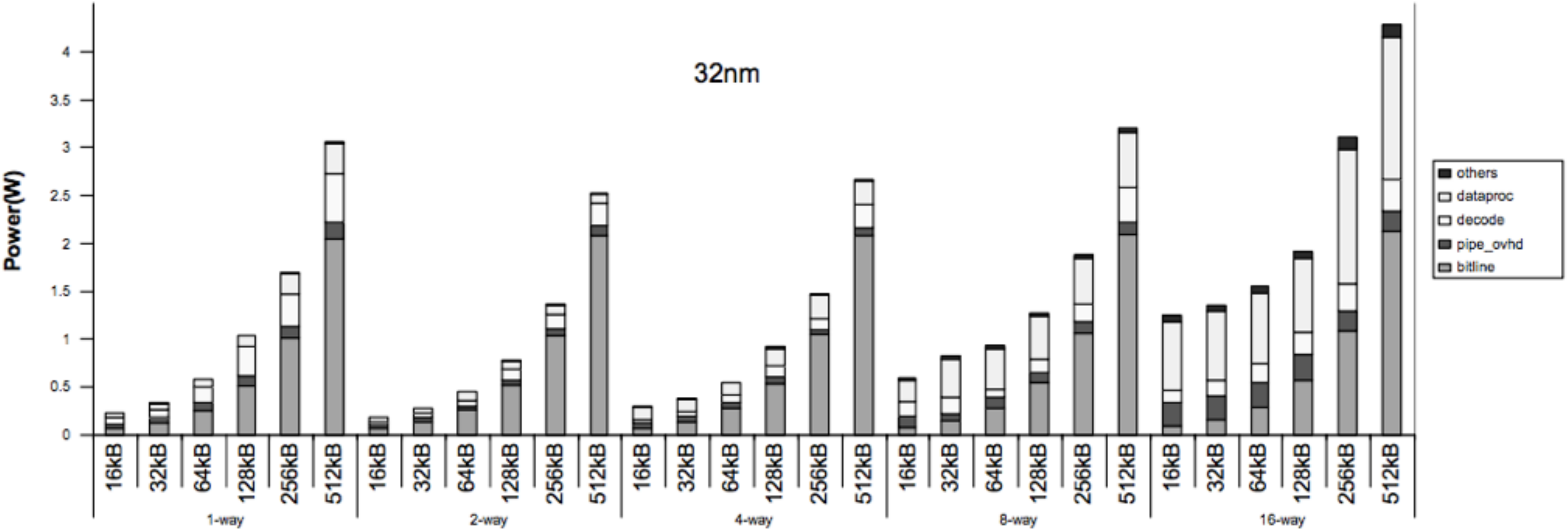
0.1
0.01
0.001
1e-04
1e-05
1e-06
miss rate
1K
4K
16K
64K
256K
1M
Inf
cache size
Direct
2-way
4-way
8-way
Full

Cache: can we do better?

- Place another cache level (L2)
- Larger and slower than L1
- But still much faster than memory
- Eventually repeat



Cache: how much power?

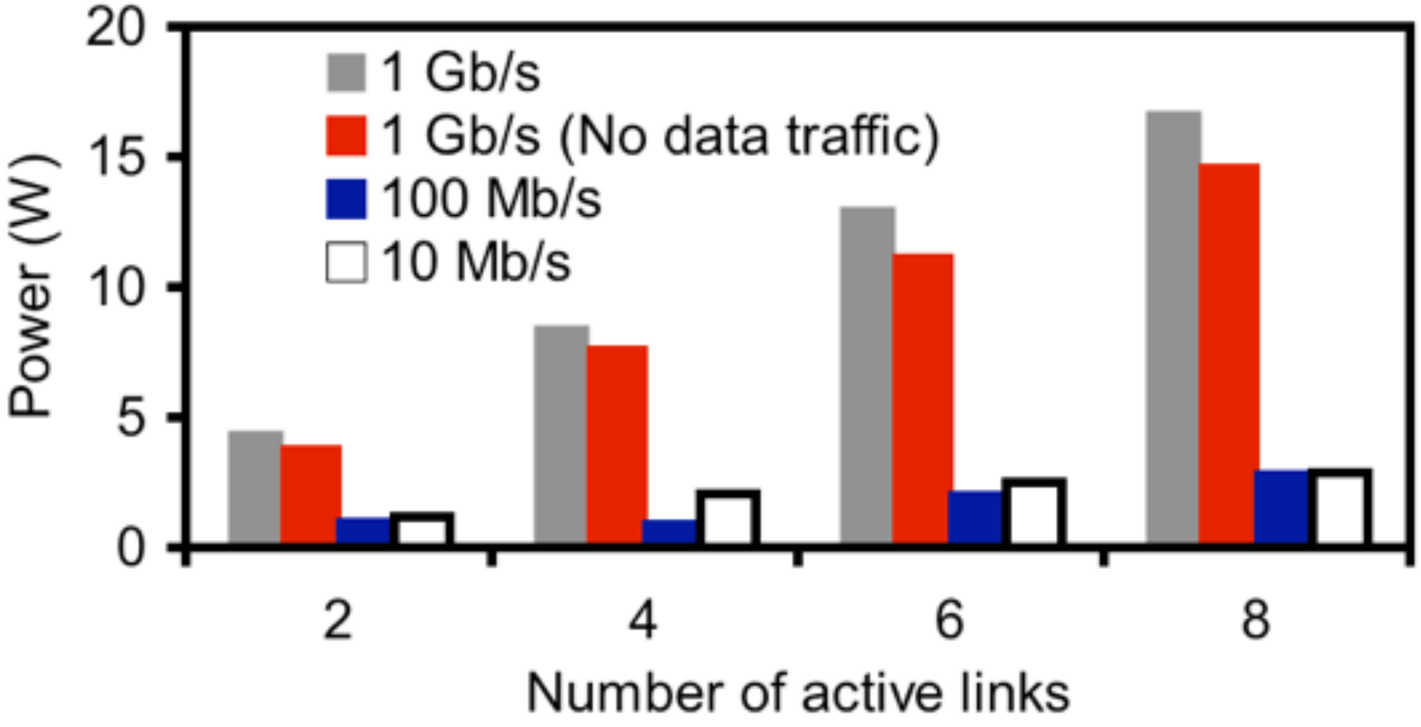


Energy-Efficient Ethernet

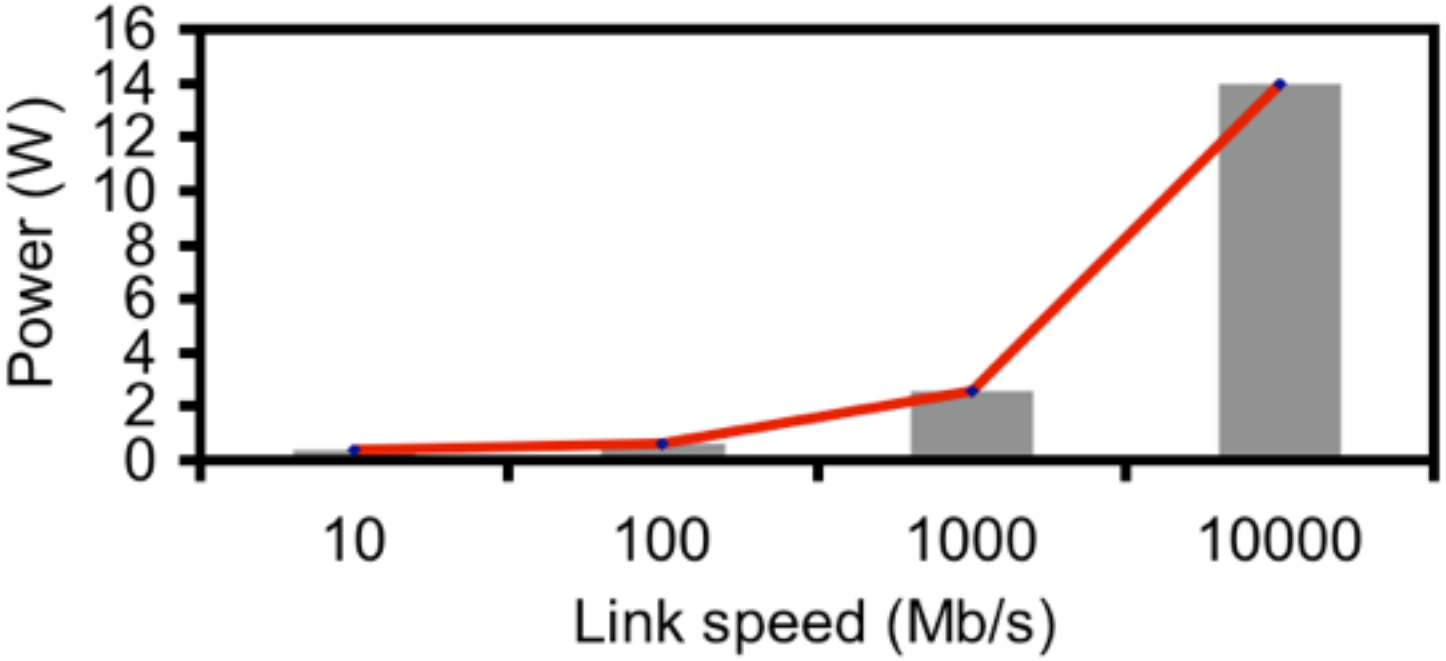
Energy
Efficient
Ethernet

Ethernet energy consumption

- **Typical switch with 24 ports 10/100/1000 Mb/s**



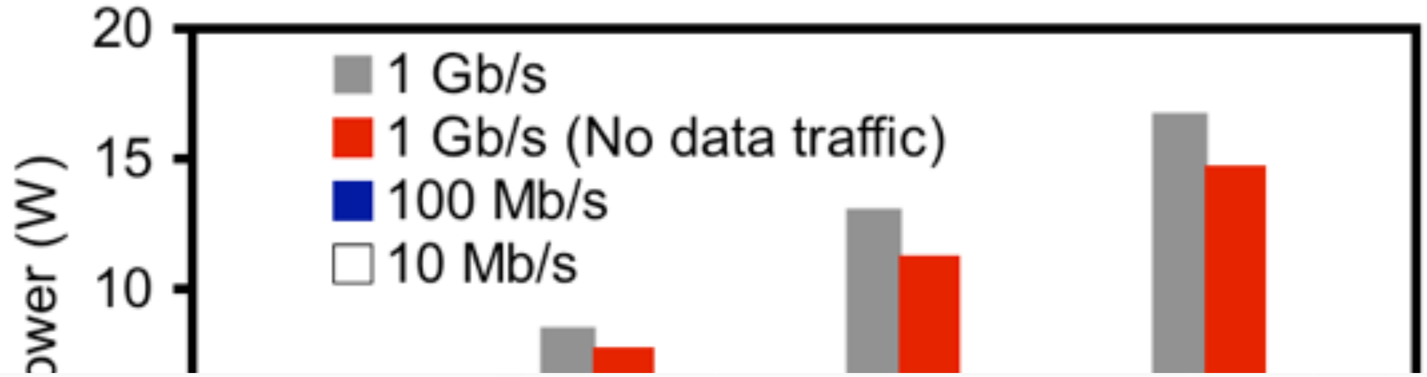
- **Various computer NICs averaged**



Energy Efficient Ethernet An Overview - Mike Bennett Lawrence Berkeley National Lab

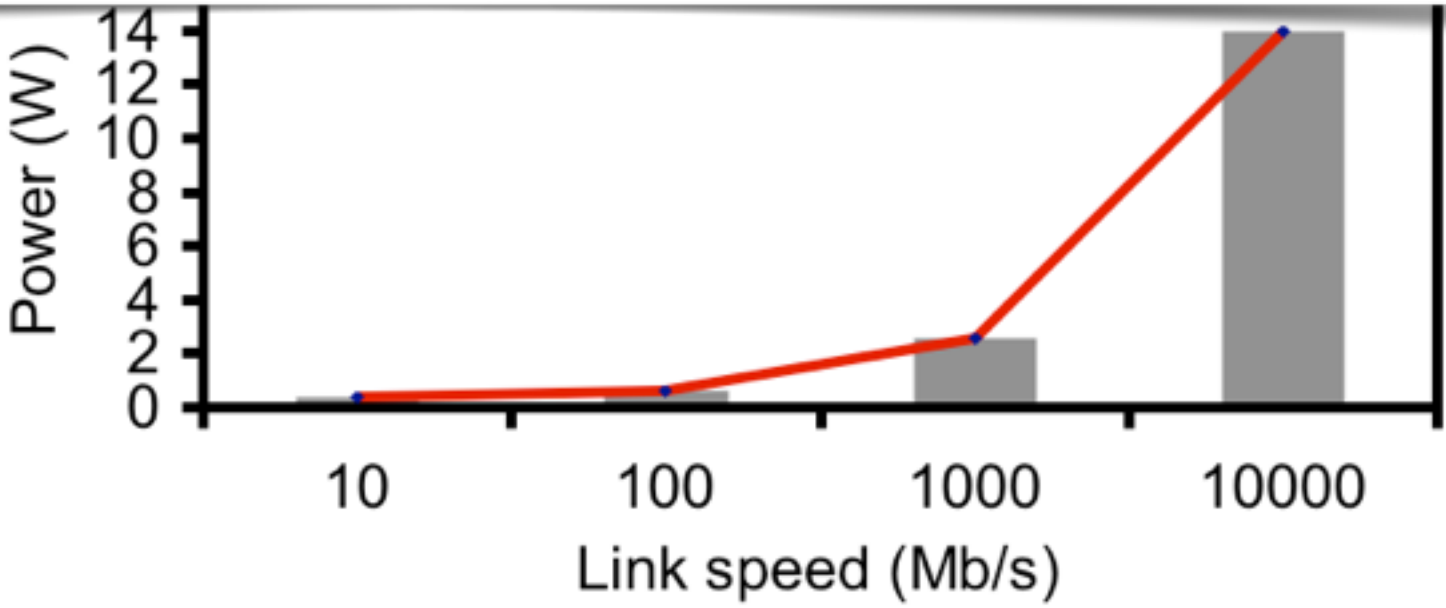
Ethernet energy consumption

- **Typical switch with 24 ports 10/100/1000 Mb/s**



In 2005, all the network interface controllers in the United States (in computers, switches, and routers) used an estimated 5.3 terawatt-hours of electricity.

- **Various computer NICs averaged**



Energy Efficient Ethernet An Overview - Mike Bennett Lawrence Berkeley National Lab

Ethernet energy consumption

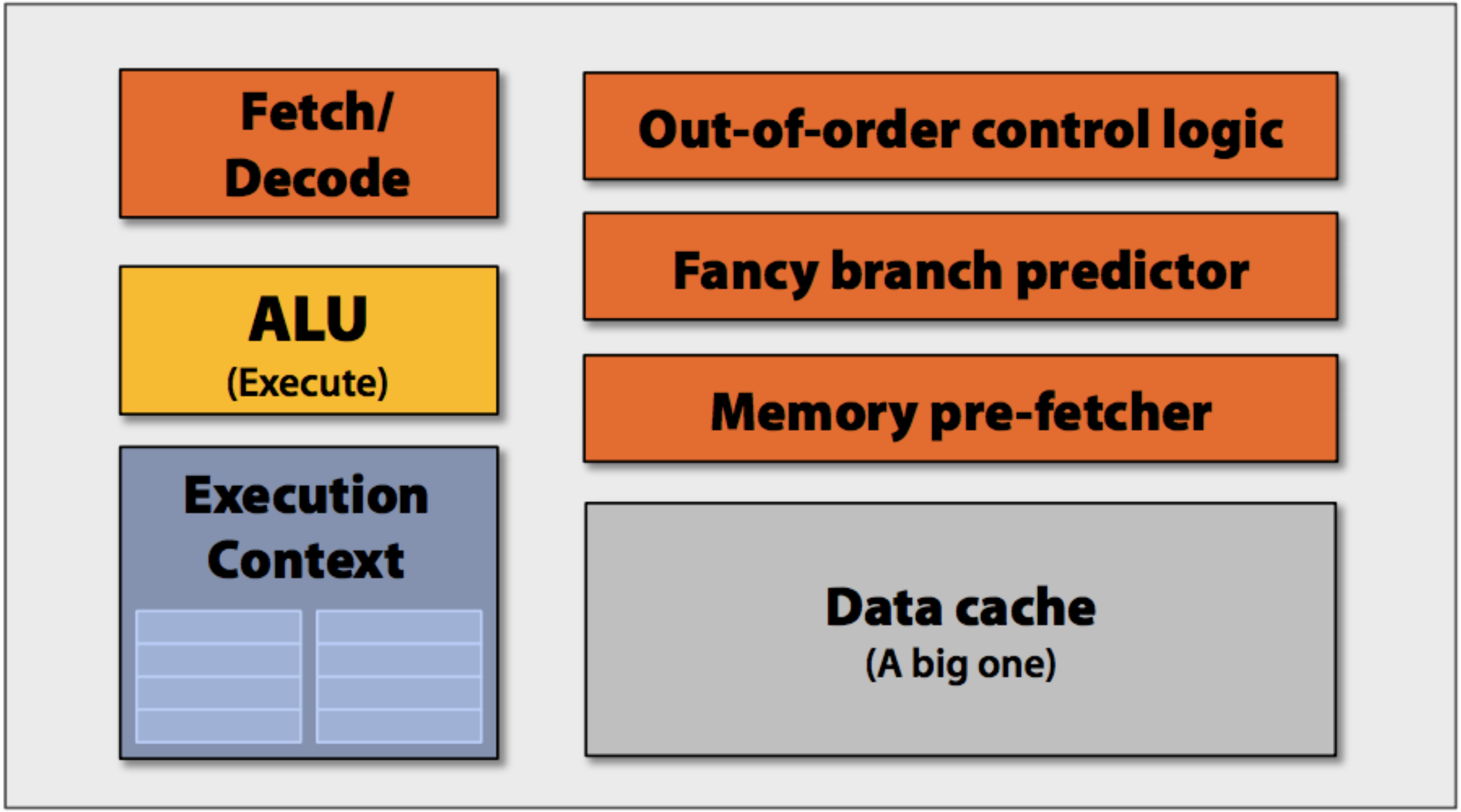
- An IEEE 802.3 Study Group
  - Formed in November 2006 to study the idea
  - Technical and economic feasibility
  - Compatibility and distinct identity
  - Broad market potential
- The Institute of Electrical and Electronics Engineers (IEEE), through the IEEE 802.3az task force developed the standard ratified in September 2010
  - Some companies introduced technology to reduce the power required for Ethernet before the standard was ratified, using the name Green Ethernet.

HPC: GPU vs CPU energy efficiency

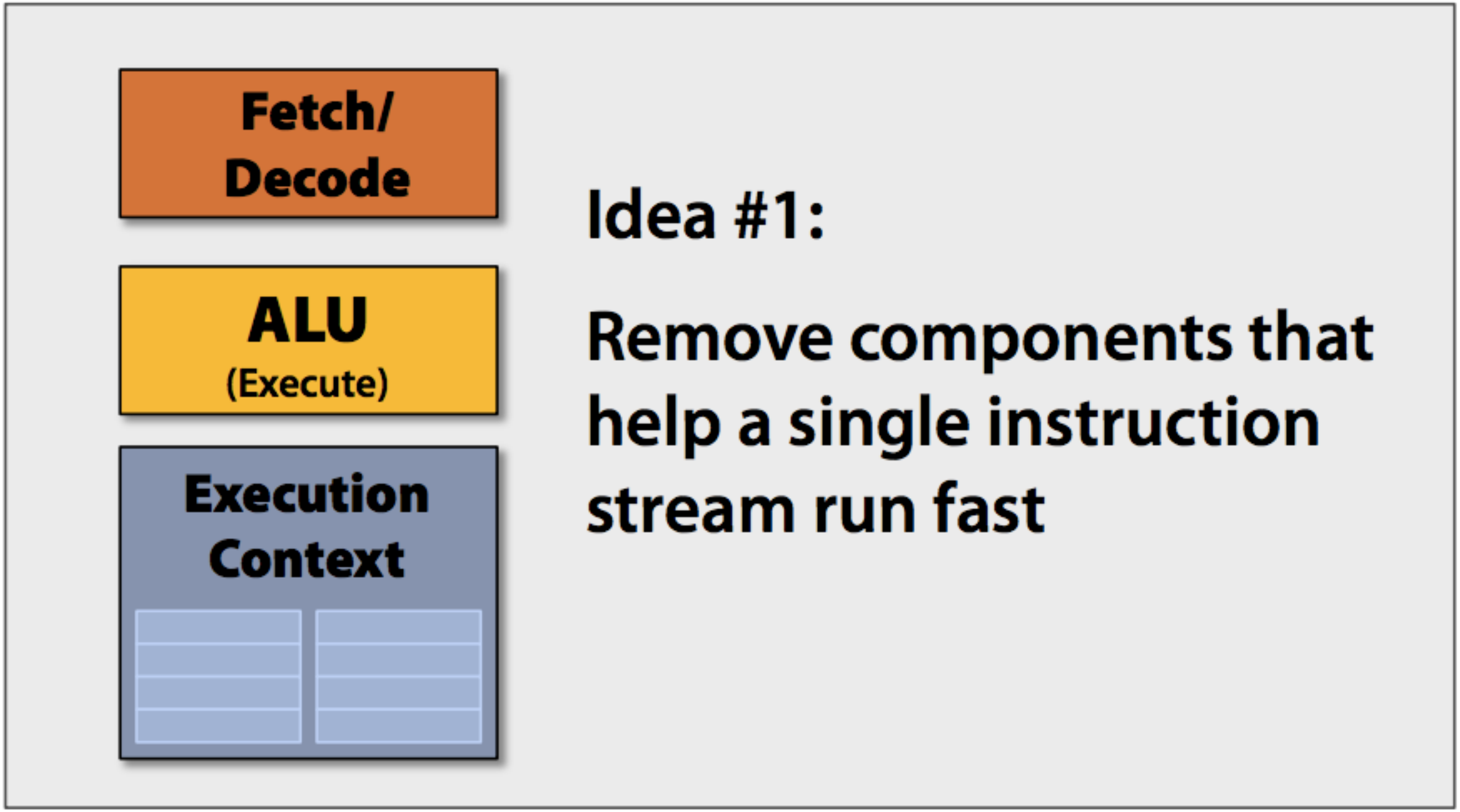
What is a GPU

Graphical **P**rocessing **U**nit

GPU architecture



GPU architecture



Fetch/ Decode

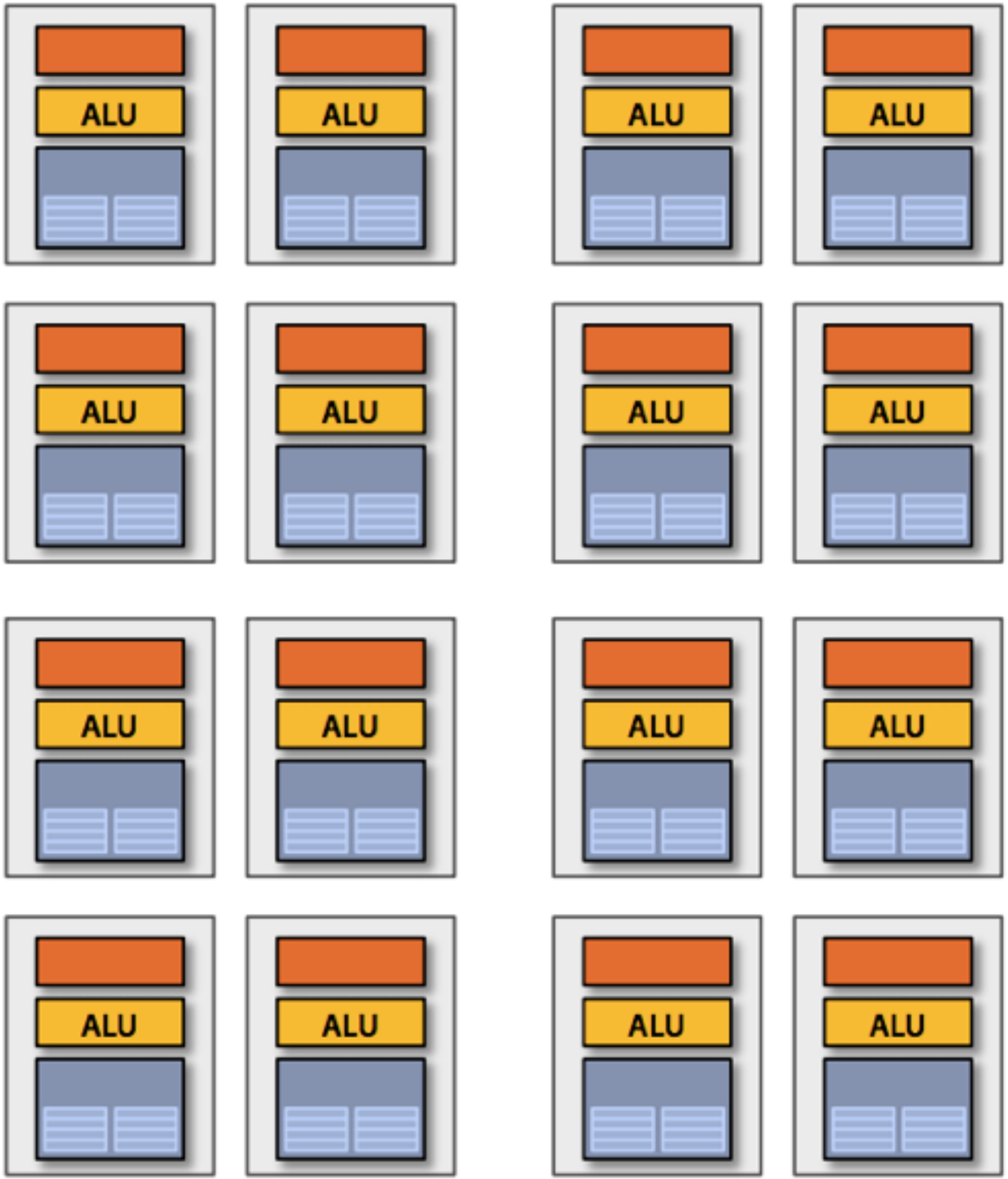
ALU (Execute)

Execution Context

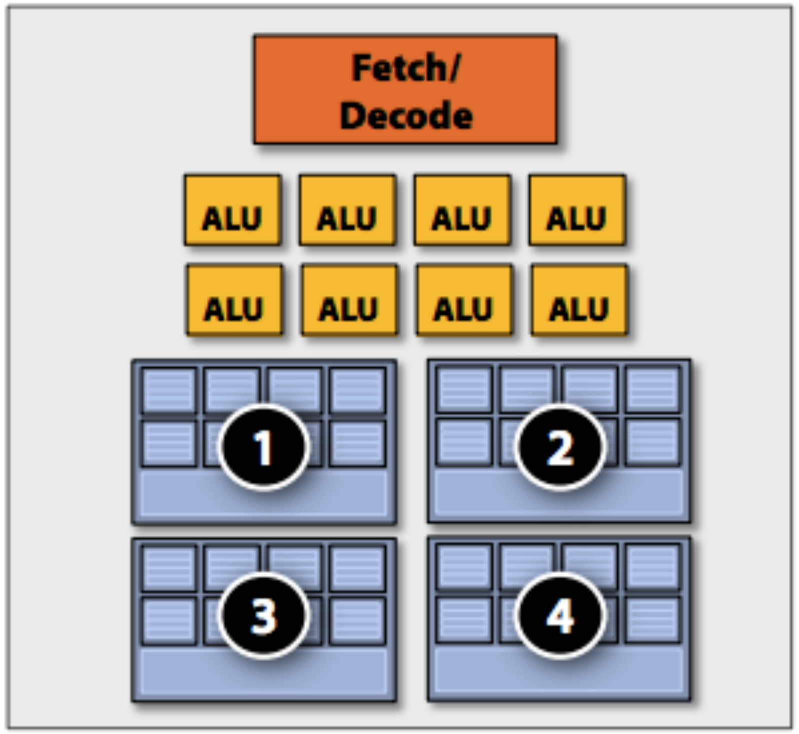
Idea #1:

Remove components that help a single instruction stream run fast

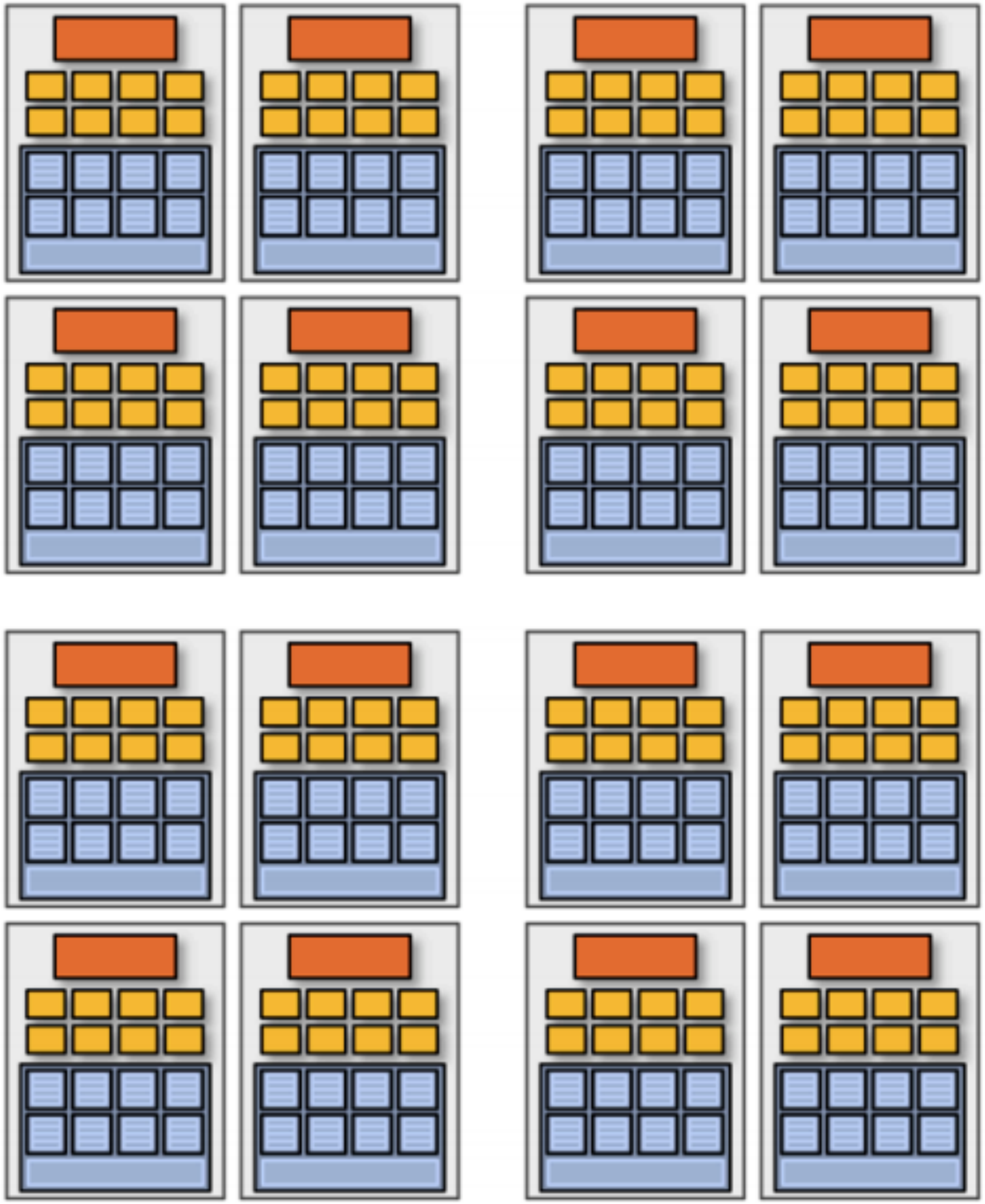
GPU architecture



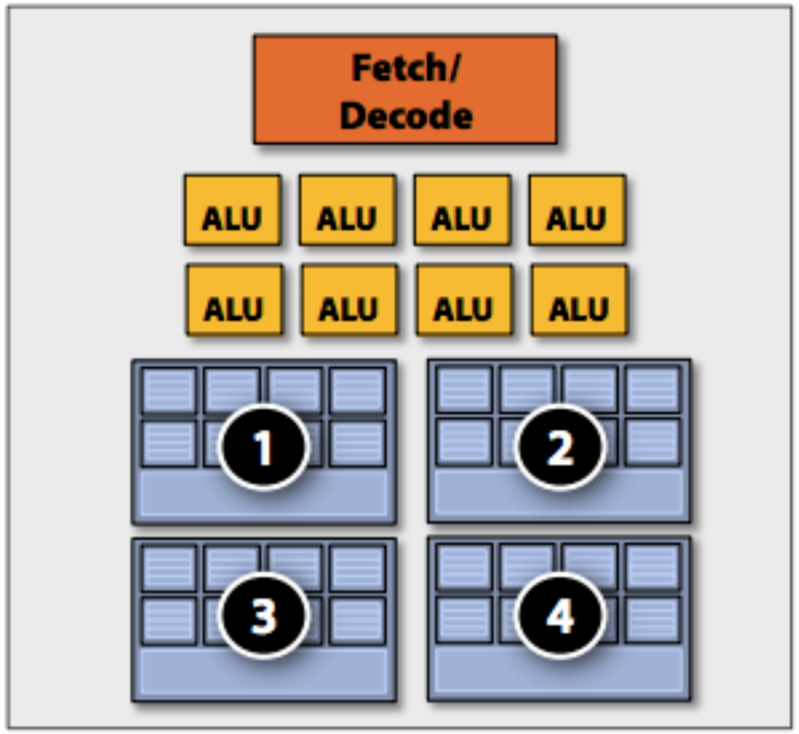
GPU architecture



SIMD

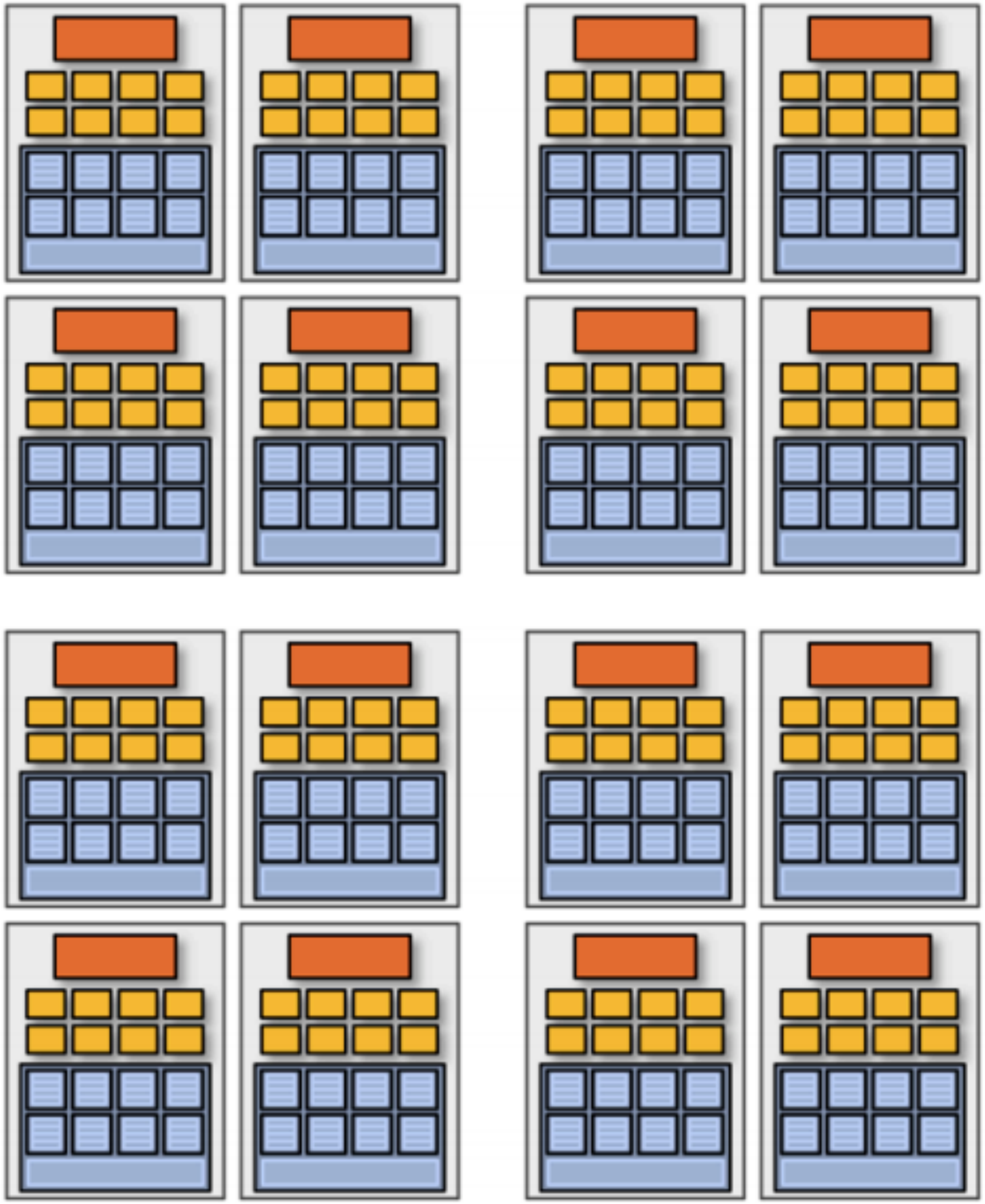


GPU architecture



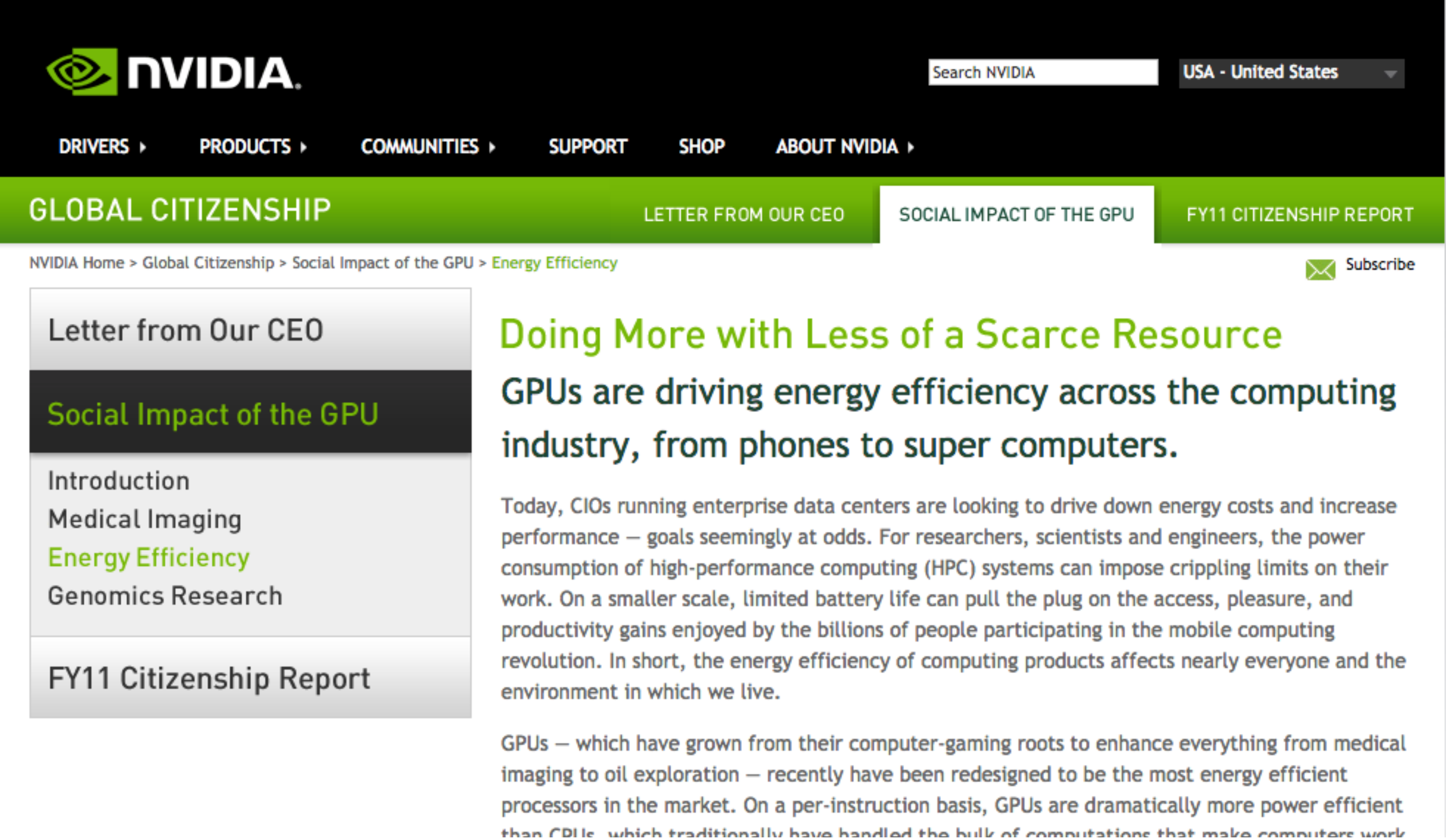
SIMD

Back to the ‘70s



GPU programming

- Old School Style: write program in form of vector graphic problem
- CUDA (Compute Unified Device Architecture): framework developed by NVIDIA
- OpenCL (Open Computing Language): is a framework for writing programs that execute across heterogeneous platforms (CPUs, GPUs, DSPs) maintained by the Khronos Group



Doing More with Less of a Scarce Resource

GPUs are driving energy efficiency across the computing industry, from phones to super computers.

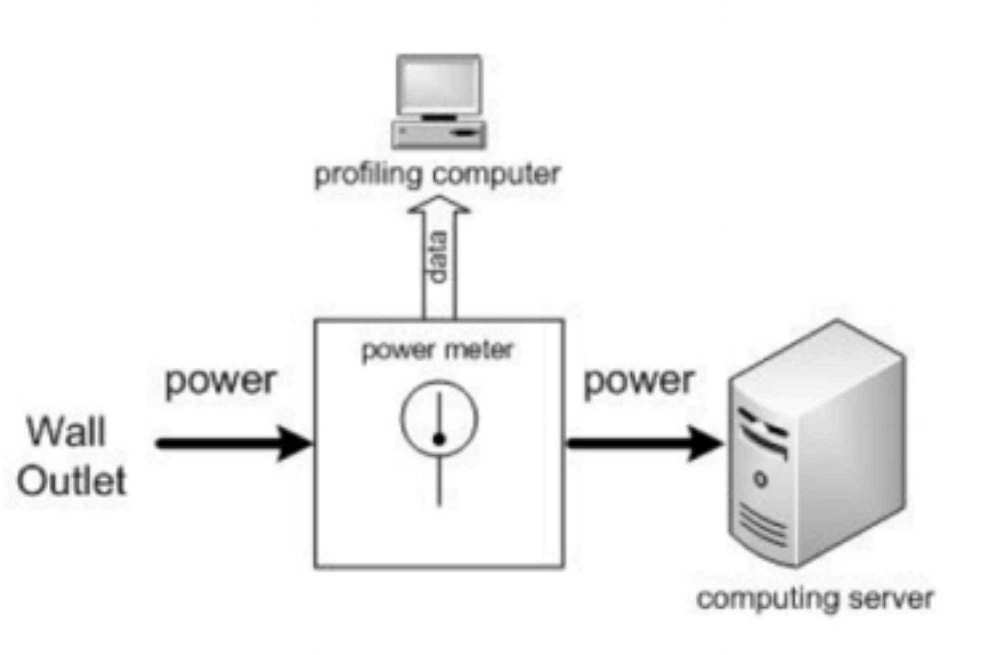
Today, CIOs running enterprise data centers are looking to drive down energy costs and increase performance — goals seemingly at odds. For researchers, scientists and engineers, the power consumption of high-performance computing (HPC) systems can impose crippling limits on their work. On a smaller scale, limited battery life can pull the plug on the access, pleasure, and productivity gains enjoyed by the billions of people participating in the mobile computing revolution. In short, the energy efficiency of computing products affects nearly everyone and the environment in which we live.

GPUs — which have grown from their computer-gaming roots to enhance everything from medical imaging to oil exploration — recently have been redesigned to be the most energy efficient processors in the market. On a per-instruction basis, GPUs are dramatically more power efficient than CPUs, which traditionally have handled the bulk of computations that make computers work

GPU

what about energy?

GPU power consumption



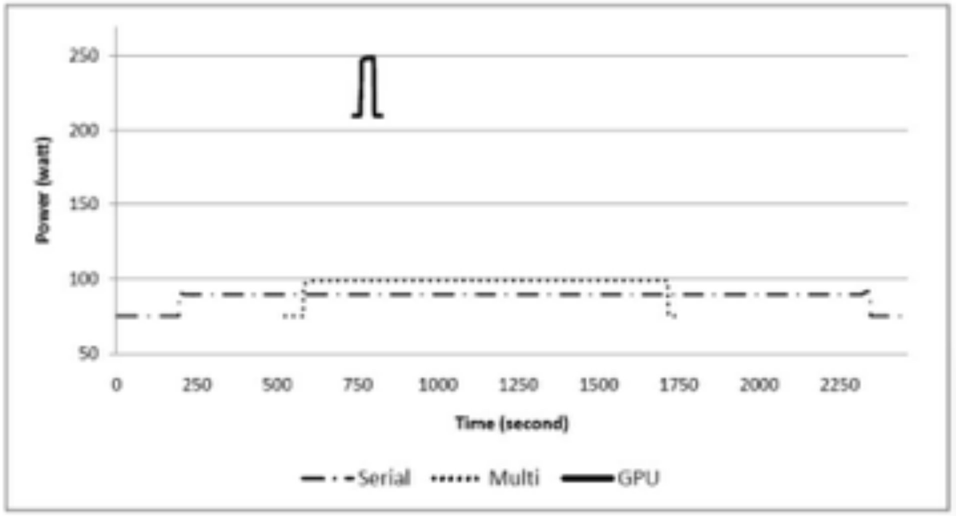


Figure 3. The Execution Time and Power Consumption of the CPU and GPU

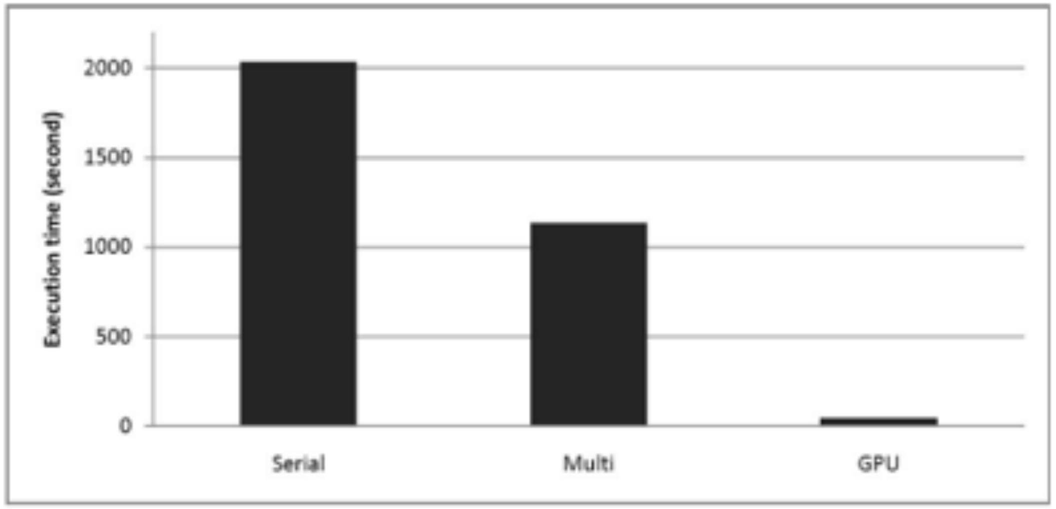


Figure 4. Execution Time of GEM

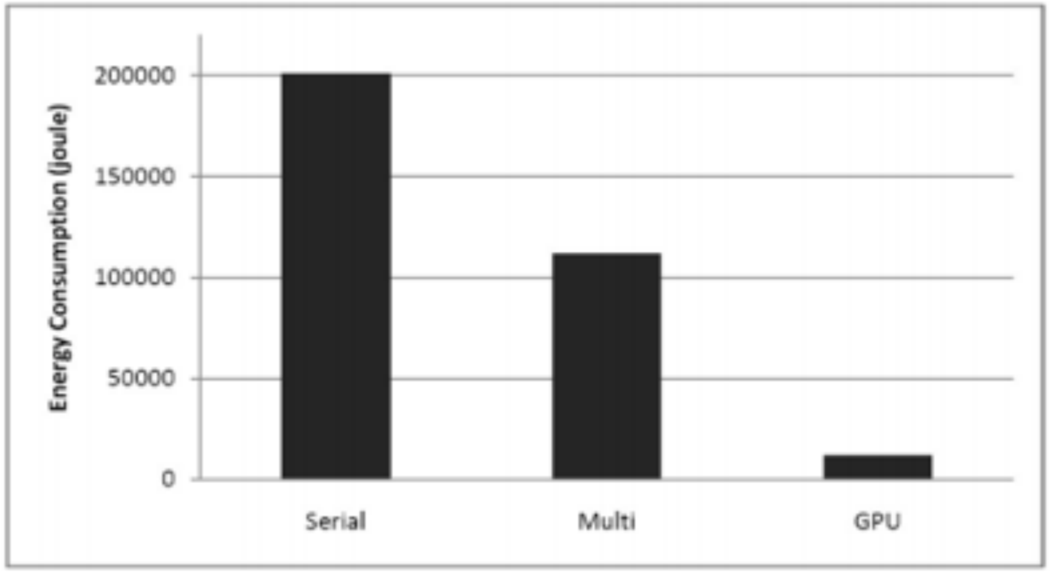
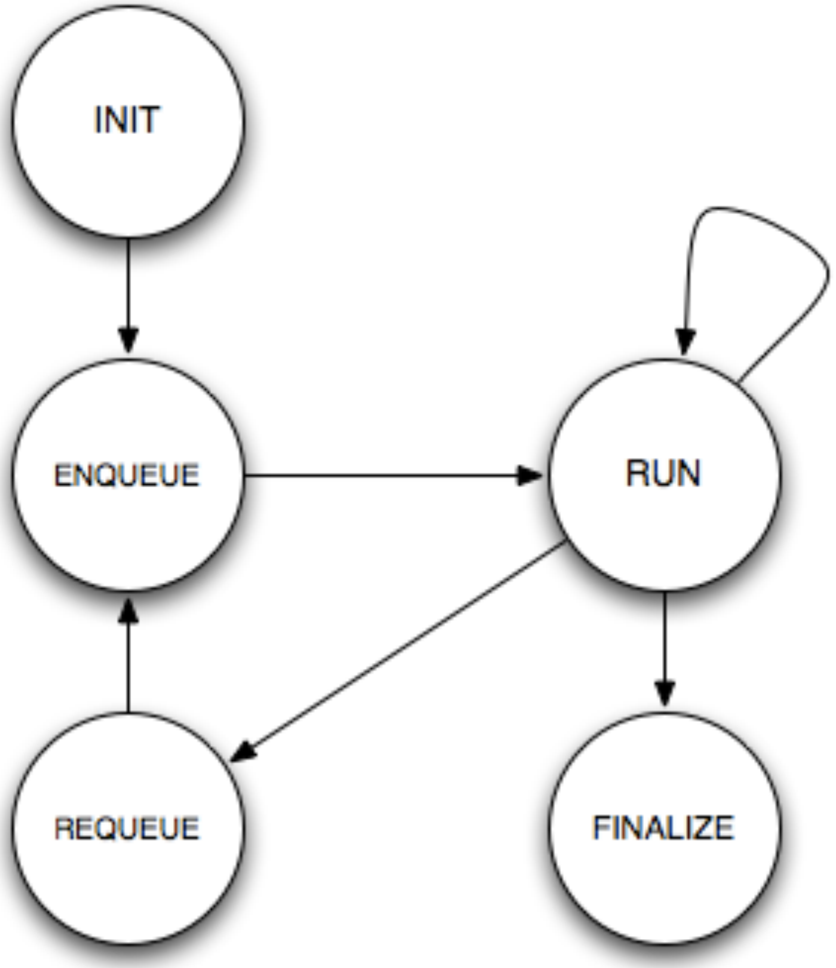


Figure 5. Energy of GEM

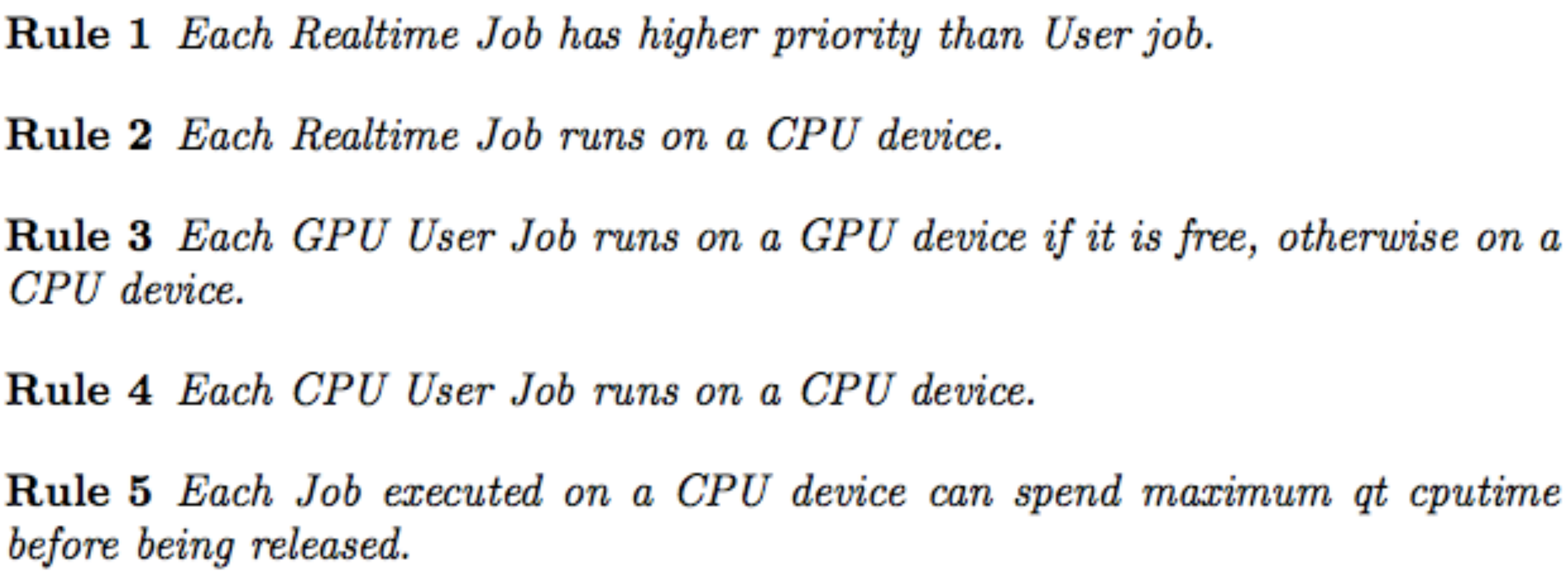
Huang, Song, Shucai Xiao, and Wu-chun Feng. "On the energy efficiency of graphics processing units for scientific computing." Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on. IEEE, 2009.

Simulation framework for scheduling performance evaluation on CPU-GPU heterogeneous system

$$\mathcal{H}_{sm} = (\mathcal{S}, \mathcal{D}, \mathcal{J}, \mathcal{E}, \delta)$$



Single non-preemptive priority queue



Rule 1 *Each Realtime Job has higher priority than User job.*

Rule 2 *Each Realtime Job runs on a CPU device.*

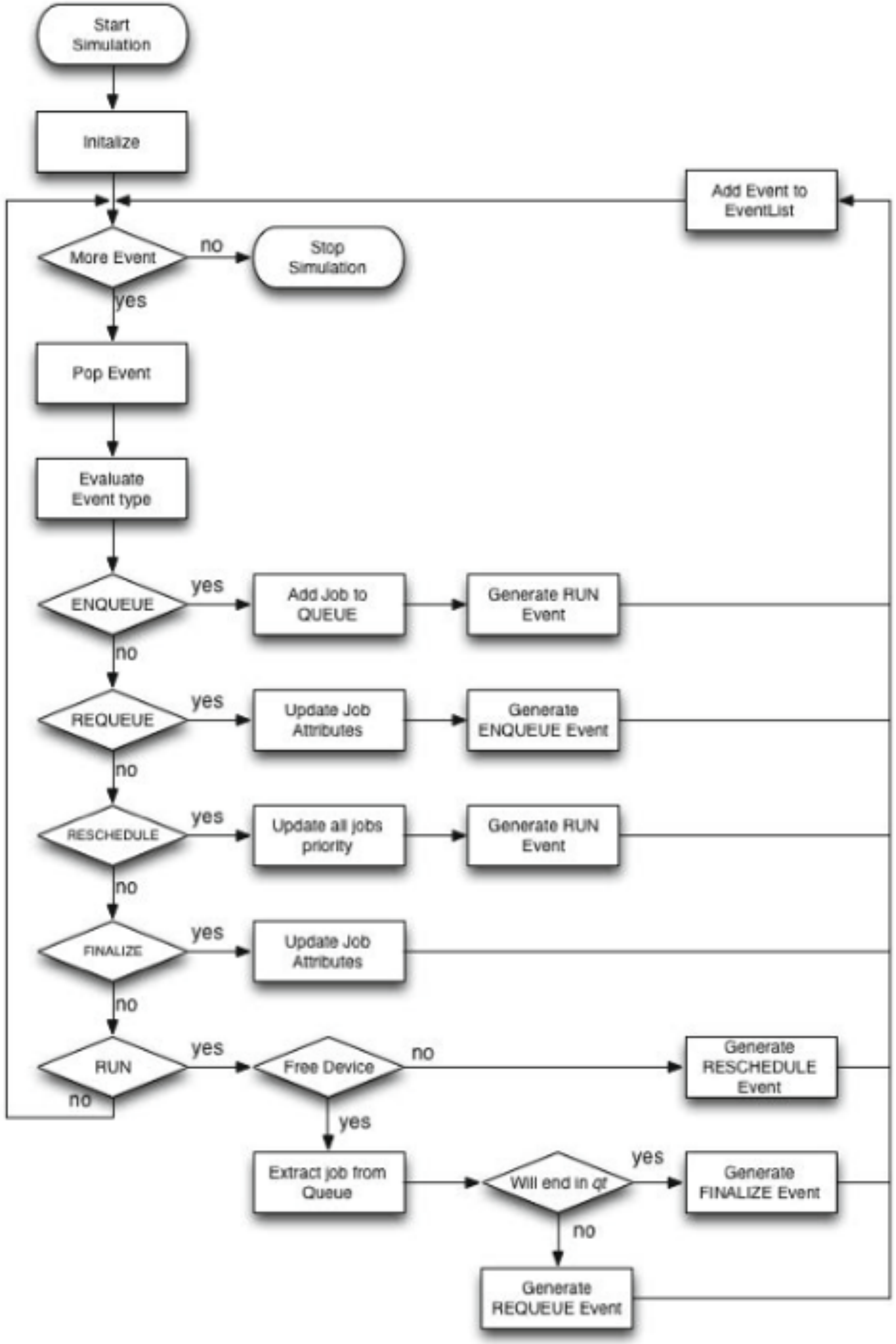
Rule 3 *Each GPU User Job runs on a GPU device if it is free, otherwise on a CPU device.*

Rule 4 *Each CPU User Job runs on a CPU device.*

Rule 5 *Each Job executed on a CPU device can spend maximum qt cputime before being released.*

Vella, Flavio, et al. "A simulation framework for scheduling performance evaluation on CPU-GPU heterogeneous system." Computational Science and Its Applications–ICCSA 2012. Springer Berlin Heidelberg, 2012. 457-469.

Simulator work flow



Vella, Flavio, et al. "A simulation framework for scheduling performance evaluation on CPU-GPU heterogeneous system." Computational Science and Its Applications–ICCSA 2012. Springer Berlin Heidelberg, 2012. 457-469.

Thank you for your attention!

igor.neri@nipslab.org