IP-based Sensor Networking with Contiki

Adam Dunkels, PhD, **Fredrik Österlind, Nicolas Tsiftes,**
Niclas Finne, Joakim Eriksson, Zhitao He (SICS)
Oliver Schmidt (SAP AG)
Julien Abeillé, Mathilde Durvy, PhD (Cisco)
Takahide Matsutsuka, Simon Barner, Simon Berg
Colin O'Flynn (NewAE)
Eric Gnoske, Blake Leverett, Michael Vidales (Atmel)
David Kopf

Today

- IP for embedded systems
- The history of IP-based sensor networks
- Contiki
- Hands-on
  - Instant Contiki
  - Contiki for Tmote Sky
  - Tmote Sky as IPv6/6lowpan bridge

IP for Embedded Systems

History of IP for Embedded Systems

- Embedded systems
  - Resource-constrained
  - Autonomous operation
  - Remote configuration, data collection, …
- 1999: EMIT Alliance – remote communication
  - EMbed the InterneT Alliance
  - EmWare, Atmel, Texas Instruments, Zilog, Microchip, …
  - ”IP too heavyweight”
    - Non-IP based wired protocol
    - Gateways

2000: lwIP

- Open source lightweight IP stack
  - Top-down design
    - Similar to BSD, Linux, …
  - 40 kb code, 20 kb RAM
  - IP, ICMP, UDP, TCP
  - Widely used
    - ”Too large”…

2001: uIP

- World’s smallest TCP/IP stack
  - Open source
  - Bottom-up design
  - 4 kb code, 1 kb RAM
  - Fully RFC compliant
  - IP, ICMP, UDP, TCP


N504EA
EXPERIMENTAL


Radio Tagging Hardware
Door Proximity Sensors
GE Security
Door Seal Guides
10 Year battery
Light Sensor
Expansion Serial Port


Micronet


YA07 JYW

2002: IP for Embedded is a Fact

- Non-IP solutions went away
  - Highly optimized custom protocols
  - IP took over
- EMIT Alliance is no more
  - EmWare moved to uIP

The Most Important Lesson Learned

- Ability to communicate more important than performance

Example: uIP-based Pico Satellite

- CubeSat: pico satellite construction kit
  - MSP430-based
- 128 bytes of RAM for uIP

2004: IP-based Sensor Networks

- Lightweight IP stack (uIP)
- Header compression
- Low-power MAC layer
- Sensor data
  - UDP
- Configuration
  - TCP
- 2006: IETF 6lowpan

2004: First IP-based Sensor Network

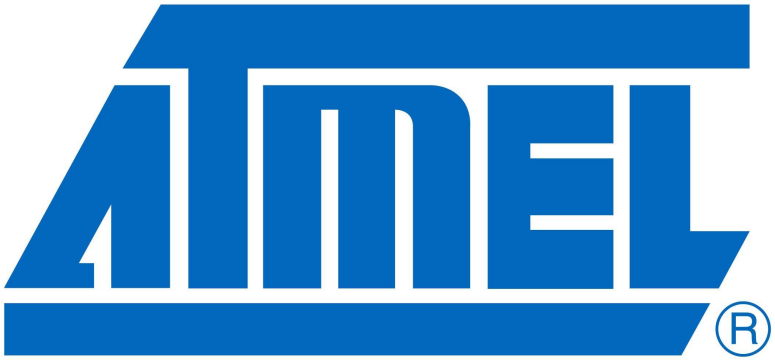
Why IP for Sensor Networks?

- Scalable
- Versatile
- Manageable
- Open and flexible standard
- Widely deployed
- Lightweight
  - As shown by uIP, uIPv6
- Low-power
  - IETF 6lowpan wg
  - Power-saving MAC protocols

September 2008:
IP for Smart Objects Alliance

ARCH ROCK
ATMEL
CISCO
Duke Energy
DUST NETWORKS
eka systems
EDF
ERICSSON
TAKING YOU FORWARD
freescale
semiconductor
GainSpan
Jennic
Kinney Consulting
NATIONAL INSTRUMENTS
NIVIS
PicosNet
Proto6
SAP
sensinode
SICS
Sun
microsystems
WATTECO

October 2008:
uIPv6, first certified IPv6 stack, SICSlowpan



ELEKTRONIK TIDNINGEN

Sökord ...

SÖNDAG 19 OKT 2

NYHETER FÖR ELEKTRONIKPROFFS

ndern | Prenumerera | Annonsera | Om oss » | About » | Sajtkarta

Cisco och Atmel anammar svenskt mini-OS

Av Adam Edström (adam@etn.se)

lt ska ku
Cisco oc
renska
protokol
orskare

heise online

Home | Newsticker | 7-Tage-News | News-A

heise online › News › 2008 › KW 43 › IPv6-S

23.10.2008 14:22

IPv6-Stack für Sensoren

vorlesen / MP3-Download

Cisco, Atmel und das s
den IPv6-Stack uIPv6
läuft auf Sensor-Hardw
Open-Source-Lizenz u

the INQUIRER

Les actualités impertinentes du monde IT

HERCHE

ercher

vres blancs

ÉGORIES

NNEZ VOUS À LA
/SLETTER

ESSOIRES

ARTICLE

Le stack UIPv6 disponible au téléchargement

Internet, Logiciels, Réseaux, Web 2.0

Par : Cyril Fussy - Mardi 14 octobre 2008 à 14:33

Cisco, Atmel et l'Institut Suédois des Sciences Informatiques (*Swedish Institute of Computer Science* ou SICS) ont annoncé la disponibilité d'UIPv6.La pile de protocoles *open source* pour IPv6 est capable d'assigner des adresses *Internet Protocol* pour

Slashdot

NEWS FOR NERDS. STUFF THAT MATTERS.

Log In | Create Account | Help | Subscribe | Firehose

▼ Sections

Main
Apple
AskSlashdot
Books
Developers
Games
Hardware

World's Smallest IPv6 Stack By Cisco, Atmel, SICS

Posted by timothy on Wednesday October 15, @03:25PM
from the **is-beautiful** dept.

B Rog writes

> "Cisco, Atmel, and the Swedish Institute of Computer Science hav
> uIPv6, the world's smallest IPv6 compliant IPv6 stack, as open sou
> Contiki embedded operating system. The intent is to bring IP addr

mor | Annonsering | Kontakt | fikarun

Samarbete för IPv6 i sensornät

Skrivet av Göte Fagerfjäll

2008-10-16

Cisco, Atmel och SICS lanserar tillsammans

November 2008:
30th best invention of 2008

TIME IN PARTNERSHIP WITH CNN

TIME's Best Inventions of 2008

49 Best Inventions

Previous Next

30. The Internet Of Things

ARTICLE TOOLS
Print
Email
Sphere
AddThis
RSS
Yahoo! Buzz

In September, a group of high-tech companies that includes Cisco and Sun formed the IP for Smart Objects Alliance. Simply put, the organization intends to create a new kind of network that will allow sensor-enabled physical objects — appliances in your home, products in a factory, cars in a city — to talk to one another, the same way people communicate over the Internet.

ILLUSTRATION FOR TIME BY CHRISTOPH NIEMANN

Top Stories
- Readying Bus
- More Allegati Governor Cas
- Why Nicarag
- Where the Re
- Europe's Hop Being Dashed

The Contiki Operating System

Contiki as an Open Source Project

- Open source – BSD license
  - C programming language
- Version 1.0 released in March 2003
  - Version 2.2.3 in March 2009
- Highly portable
  - Tmote Sky, JCreate, TelosB, Atmel Raven, MSB, ESB, MicaZ, …
  - Simulators: Cooja, MSPsim, AvroraZ, netsim
  - Native platform
- Actively developed
  - 17 developers from SICS, SAP, Cisco, NewAE, TU Munich, Atmel
- Active mailing list

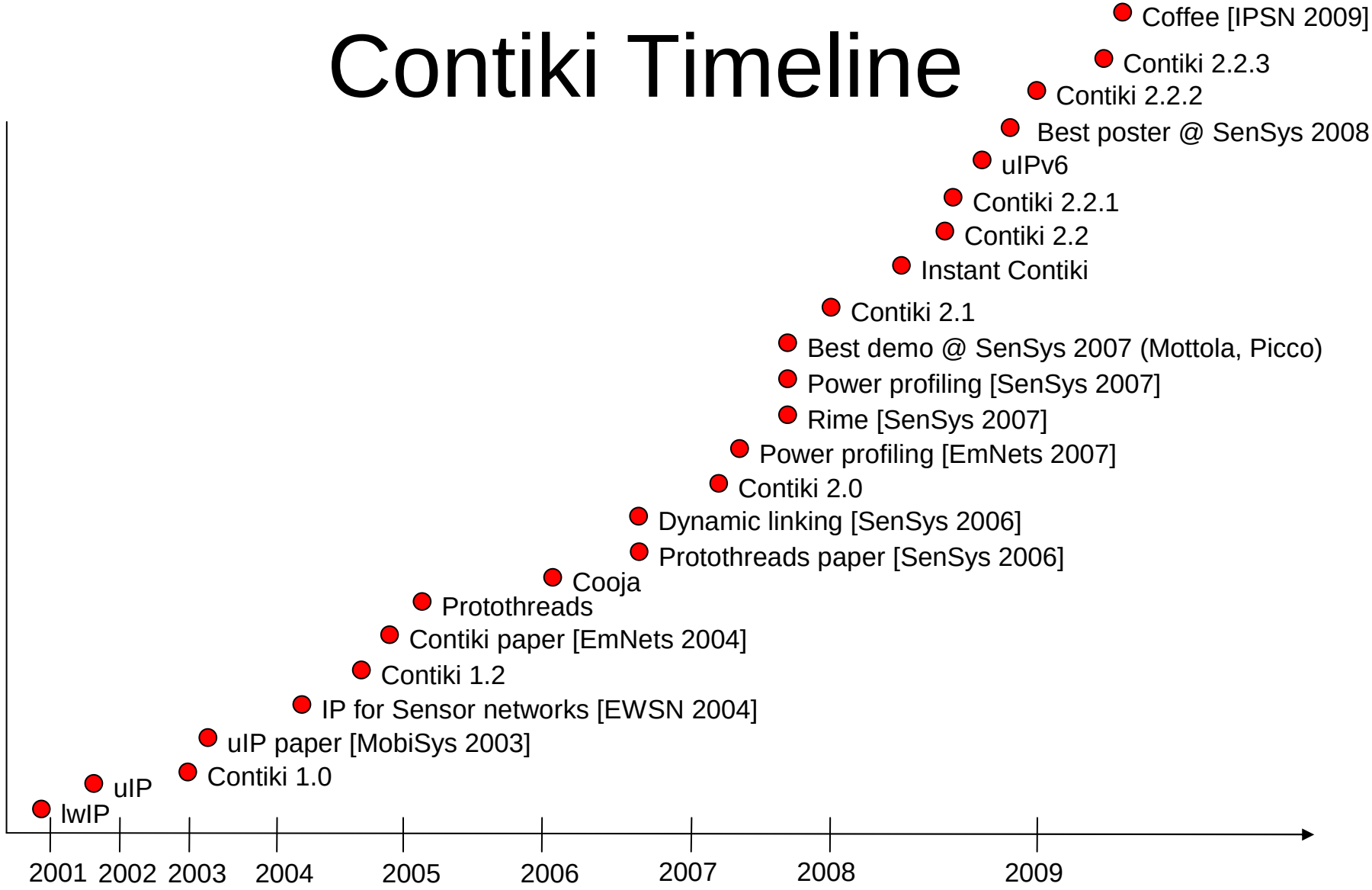
Contiki as a Research Theme

- Exploring successful computer science abstractions, mechanisms for sensor networks
  - Dynamic module loading and linking [ACM SenSys 2006]
  - File system [IEEE/ACM IPSN 2009]
  - Multi-threaded programming [EmNets 2004]
  - Java, scripting, … [ACM SenSys 2006, …]
  - Interactive network shell
  - IP networking for low-power embedded systems [ACM/Usenix MobiSys 2003, ACM SenSys2007, ACM SenSys 2008]
- Pursuing new abstractions
  - Protothreads [ACM SenSys 2006]
  - Low-power radio networking [ACM SenSys 2007]
  - Power profiling [EmNets 2007]
  - Novel communication primitives

Contiki Influence

- Dynamic loading [EmNets 2004]
  - SOS: loadable modules [MobiSys 2005]
  - TinyOS 2.1.0: loadable modules (2008)
- Preemptive threads on top of events [EmNets 2004]
  - TOSThreads: preemptive threads on top of events (TinyOS 2.1.0, 2008)
- Software-based power profiling [EmNets 2007]
  - TinyOS Quanto [OSDI 2008]
- IP-based sensor networks [MobiSys 2003, EWSN 2004, SenSys 2007]
  - IETF 6lowpan: IPv6 over 802.15.4 (2006)
  - Two papers at ACM SenSys 2008
  - Tutorials at IPSN 2009
  - Several demos, posters at SenSys, IPSN, …

Contiki Timeline



The Coffee file system [IPSN 2009]

- Flash-based file system
- open(), read(), seek(), write(), close()
- Constant memory complexity
- Very lightweight
  - 5 kb ROM, < 0.5 kb RAM
- Very fast
  - More than 92% of raw flash throughput

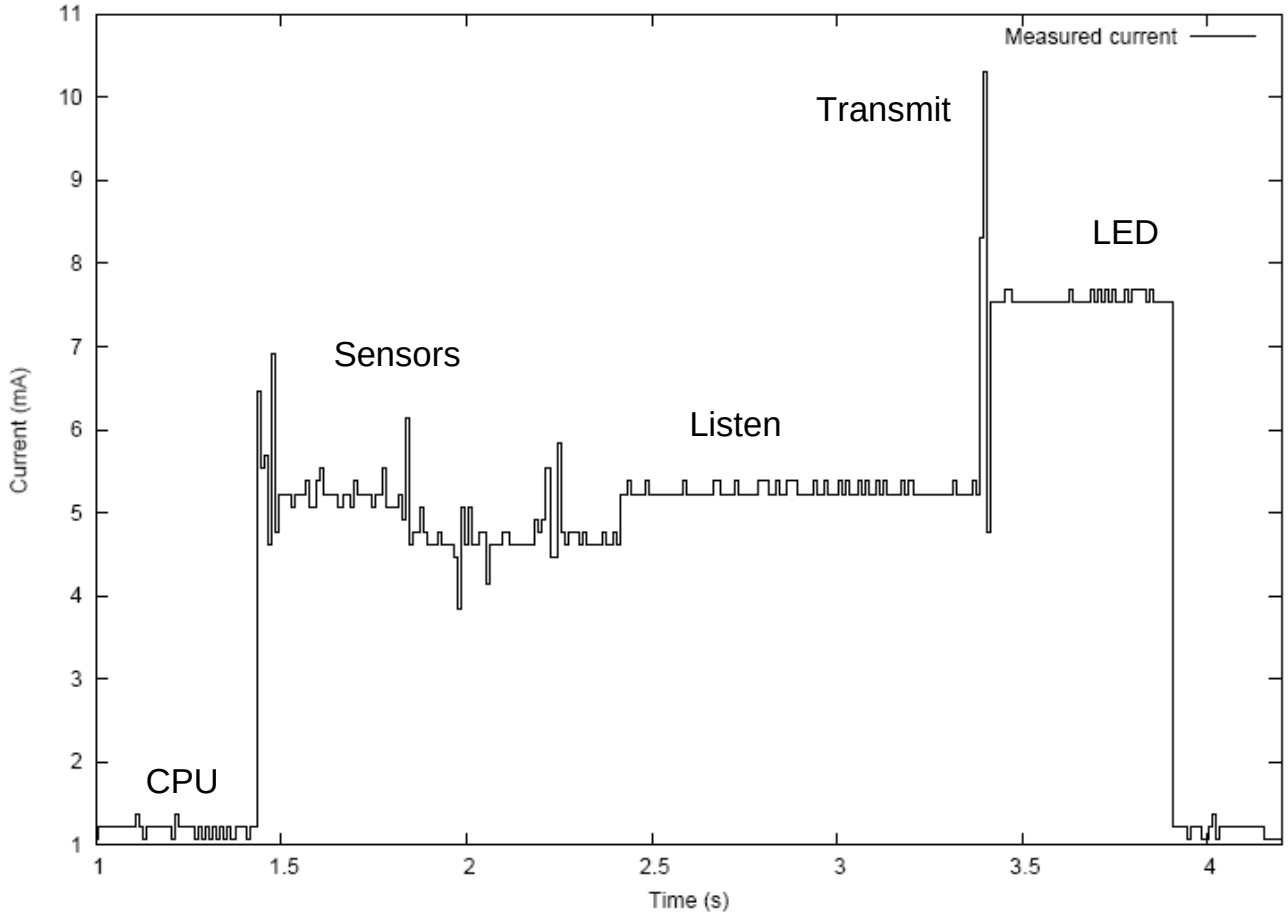
Interactive shell

- Network debugging, performance tuning
- Leverage UNIX-style pipelines
- Network commands
- Direct serial connection, or over Telnet/TCP
- A generic interface for higher level applications
  - Automated interaction, scripting

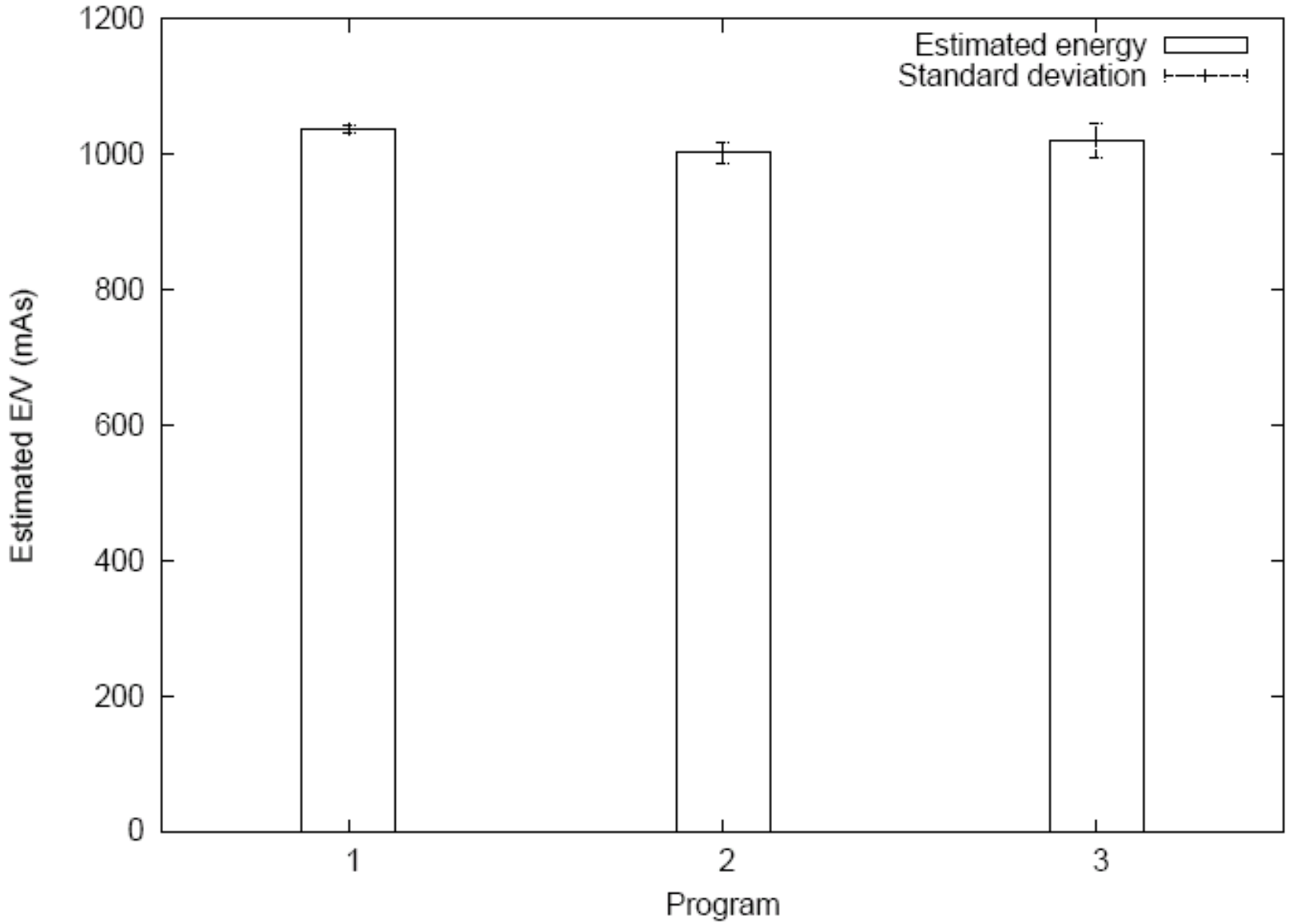
Power Profiling [EmNets 2007]

- Software-based
  - Zero-cost hardware
  - Zero-effort deployment
- Good accuracy, low overhead
- Enables network-scale energy profiling
- Enables energy-aware mechanisms

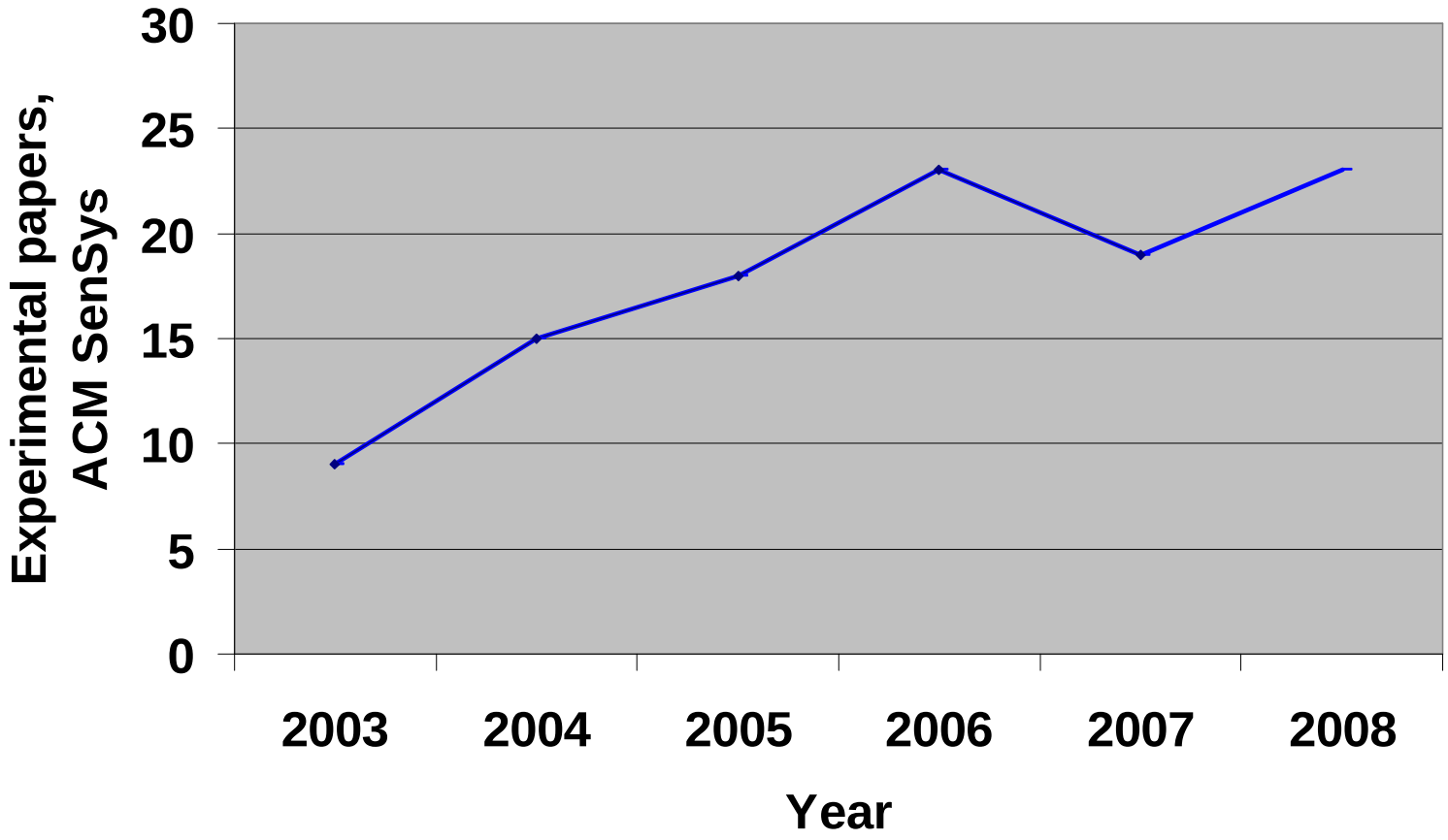
Linear current draw



Accuracy is good [EmNets 2007]



Trend: experimental is increasing



Example: B-MAC [Polastre et al, SenSys 2004]

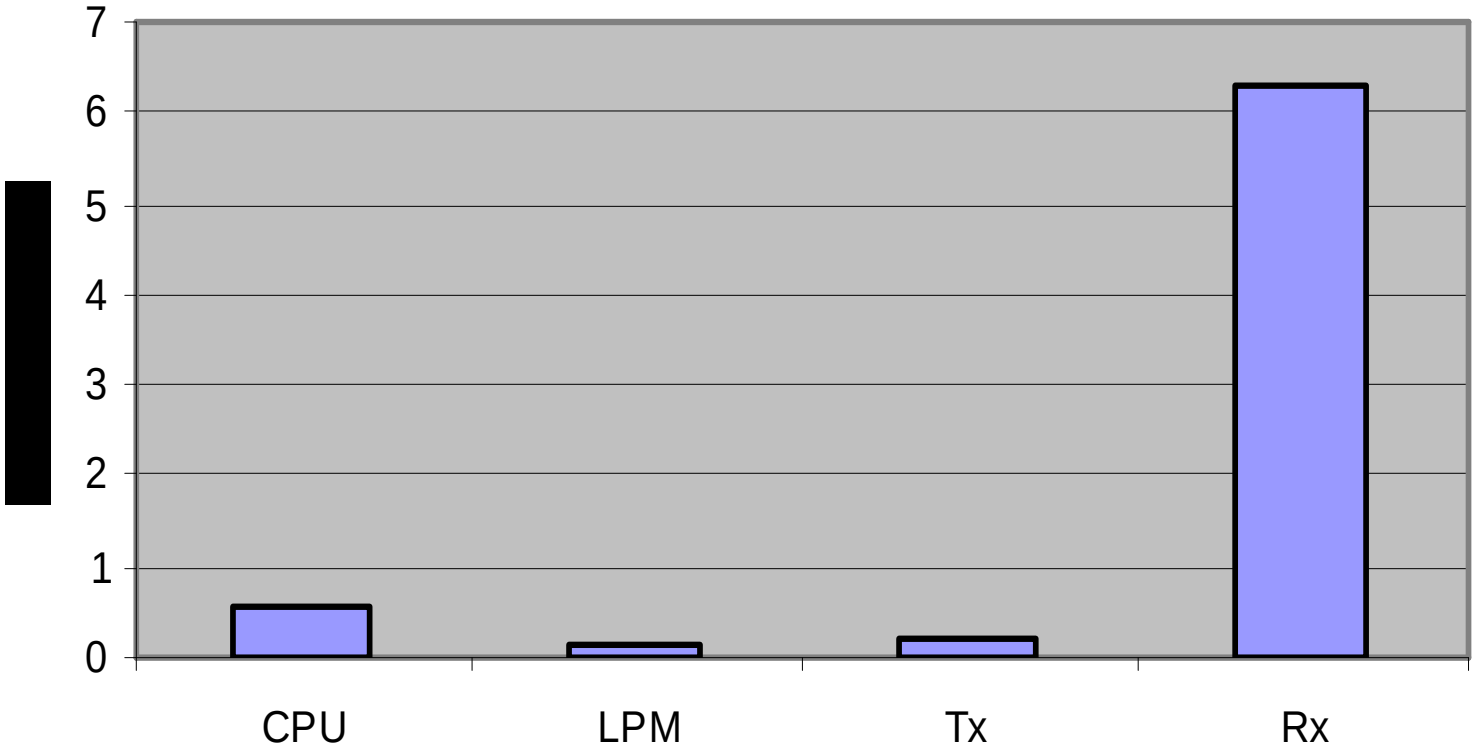
"To [experimentally] determine the power consumption of each protocol, we implemented counters in the MAC protocol that keep track of how many times various operations were performed.

For B-MAC, this includes receiving a byte, transmitting a byte, and checking the channel for activity.

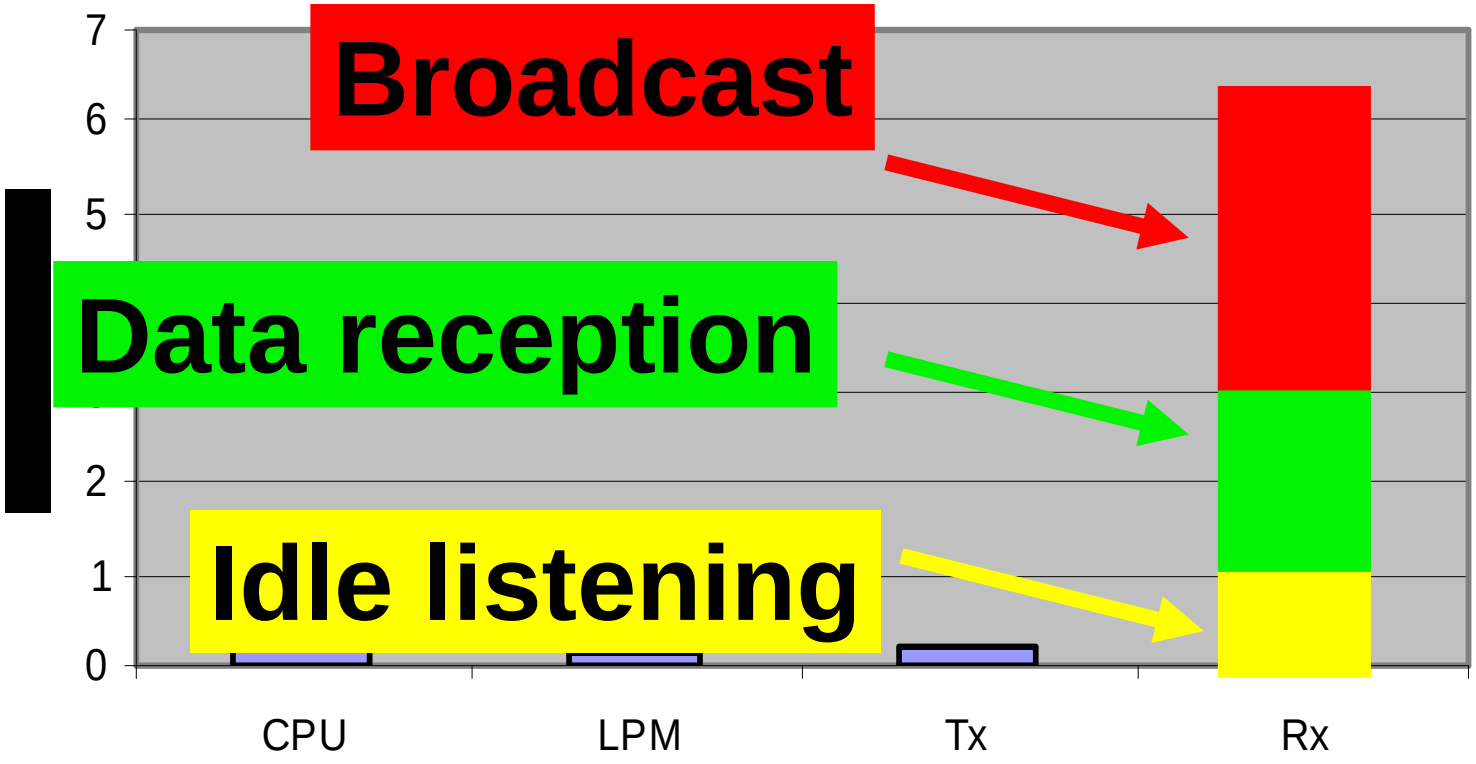
For S-MAC, we count the amount of time that the node is active, number of bytes transmitted and received, and the additional time the node spent awake due to adaptive listening."

Polastre, Hill, Culler. **Versatile low power media access for wireless sensor networks**. SenSys 2004

Per-peripheral Power Profiling



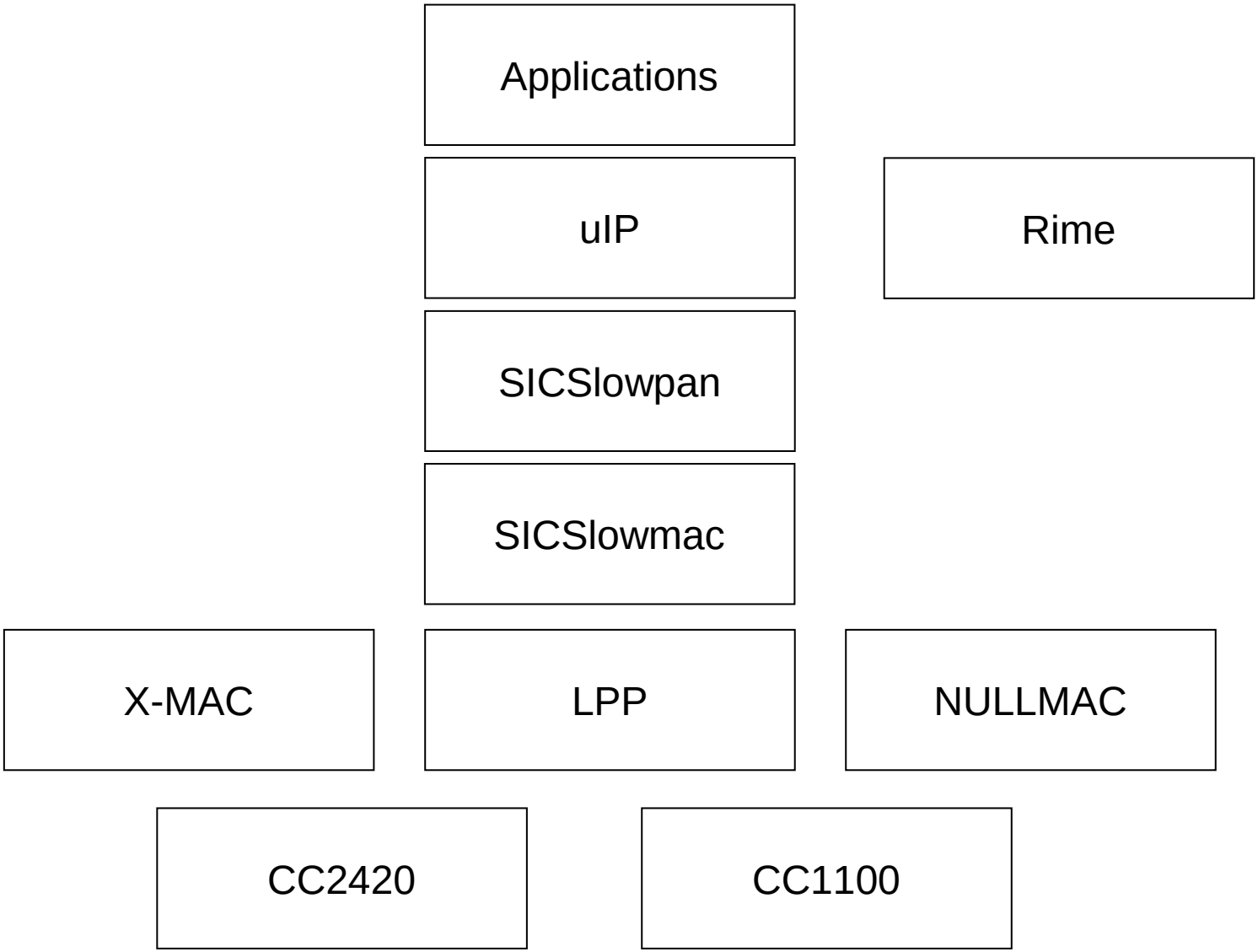
Per-packet Power Profiling



Rime: Communication Primitives [ACM SenSys 2007]

- Makes implementation of sensor network mechanisms easier
- A set of protocols
  - Data collection
  - Data dissemination
  - Unicast multi-hop routing
  - Single-hop bulk transfer
  - …

The Contniki IP Architecture



Instant Contiki

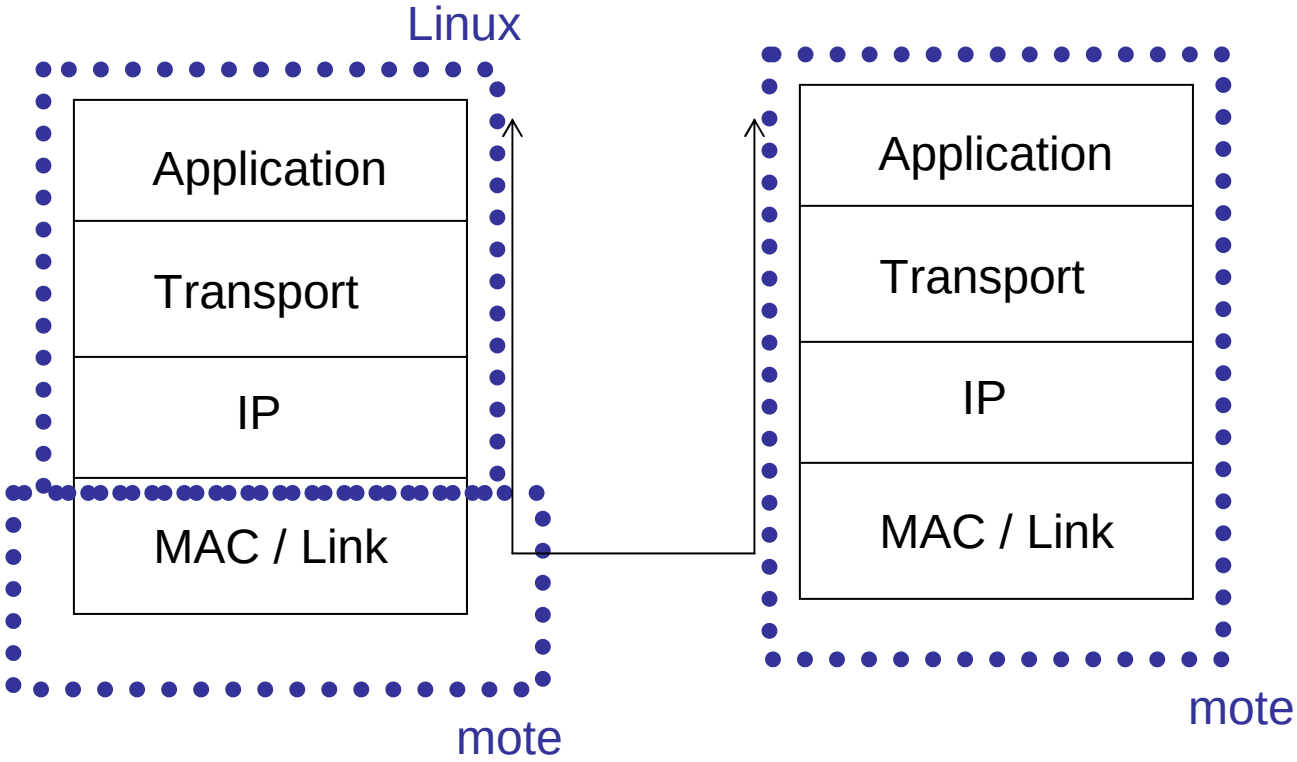
- Contiki development environment
- Single-file download
- Linux system with all Contki tools installed
- Runs in VMWare, VirtualBox
- Removes toolset installation hurdle

Seamless Integration with IP Networks

- Atmel Raven/Jackdaw USB Stick
- Attaches as network card in Windows, Linux [EWSN 2009]
- IPv6 packets sent out over 802.15.4 with 6lowpan header compression

Today: Build IPv6 Bridge with a Mote

- Make Tmote Sky mote a Linux SICSlowpan network card
- No custom hardware needed



Contiki Programming Crash Course

- Hello world
- uIP APIs
  - Raw API
  - Protosockets
- Timers
- Building

Hello, world!

```
/* Declare the process */
PROCESS(hello_world_process, “Hello world”);
/* Make the process start when the module is loaded */
AUTOSTART_PROCESSES(&hello_world_process);

/* Define the process code */
PROCESS_THREAD(hello_world_process, ev, data) {
  PROCESS_BEGIN();              /* Must always come first */
  printf(“Hello, world!\n”);  /* Initialization code goes
   here */
  while(1) {                    /* Loop for ever */
    PROCESS_WAIT_EVENT();     /* Wait for something to
   happen */
  }
  PROCESS_END();                /* Must always come last */
}
```

uIP APIs

- Two APIs
  - The “raw” uIP event-driven API
  - Protosockets – sockets-like programming based on protothreads
- Event-driven API works well for small programs
  - Explicit state machines
- Protosockets work better for larger programs
  - Sequential code

Protosockets: example

```
int smtp_protothread(struct psock *s)
{
  PSOCK_BEGIN(s);

  PSOCK_READTO(s, '\n');

  if(strncmp(inputbuffer, “220”, 3) != 0) {
    PSOCK_CLOSE(s);
    PSOCK_EXIT(s);
  }

  PSOCK_SEND(s, “HELO ”, 5);
  PSOCK_SEND(s, hostname, strlen(hostname));
  PSOCK_SEND(s, “\r\n”, 2);

  PSOCK_READTO(s, '\n');

  if(inputbuffer[0] != '2') {
    PSOCK_CLOSE(s);
    PSOCK_EXIT(s);
  }
```

Four types of timers

- struct timer
  - Passive timer, only keeps track of its expiration time
- struct etimer
  - Active timer, sends an event when it expires
- struct ctimer
  - Active timer, calls a function when it expires
- struct rtimer
  - Real-time timer, calls a function at an exact time
  - Limited function support

Using etimers in processes

```
PROCESS_THREAD(hello_world_process, ev, data) {
  static struct etimer et;          /* Must be static      */
  PROCESS_BEGIN();                  /* since processes are */
                                    /* protothreads        */
  while(1) {
       etimer_set(&et, CLOCK_SECOND);
       PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&et));
  }
  PROCESS_END();
}
```

Example: building hello world

- cd examples/hello-world
- make TARGET=native — Build monolithic system for native
- ./hello-world.native — Run entire Contiki system + app
- make TARGET=netsim — Build netsim simulation
- ./hello-world.netsim — Run netsim simulation
- make TARGET=sky — Build monolithic system image
- make TARGET=sky hello-world.u — Build & upload system image
- make TARGET=sky hello-world.ce — Build loadable module
- make TARGET=esb — Monolithic system image for ESB
- make TARGET=esb hello-world.u — Build & upload image
- make TARGET=esb hello-world.ce — Build loadable module

make TARGET=

- TARGET=name of a directory under platform/
- make TARGET=xxx savetarget
  - Remembers the TARGET
  - Example: make TARGET=netsim savetarget

Hands-on

Hands on

- Install Instant Contiki
  - USB stick with VMWare Player, Instant Contiki, FTDI Driver
- Compilation
  - Hello World for native
  - Blink for Tmote Sky
  - Hello World for Tmote Sky
  - Contiki Shell for Tmote Sky
- IPv6 mote bridge
  - Tmote Sky / JCreate as an IPv6/6lowpan bridge

Conclusions

- IP for embedded systems, sensor networks
- Contiki: many influential, widely used features
  - IP for sensor networks, dynamic loading, protothreads
  - Vision and mission
    - Exploring successful CS abstractions in a novel context, pursing new abstractions
- Instant Contiki simplifies development
- IPv6 bridge mote: no custom hardware needed

Thank You

http://www.sics.se/contiki/