Internet Connectivity for Wireless Sensor Networks

Joel J. P. C. Rodrigues
Instituto de Telecomunicações, Portugal
University of Beira Interior, Covilhã, Portugal
joeljr@ieee.org

Outline

• Introduction
• Background
  • Architectural perspective
• Challenges
• Internet of Things
• Related research efforts
• Resources
  • Software
  • Hardware
• Conclusions
• Future work
Introduction

- Wireless sensor networks
  - Smart sensors with processing core, memory, sensors, wireless transceiver and battery
  - Energy concerns/constraints
- Need for Internet connection
  - Make sensing services widely available
  - Power ubiquitous computing
- IP - Internet Protocol
  - Protocol used on the Internet (network layer)
  - Heterogeneous network interconnection
  - Map WSNs into public subnet, provide global connectivity

Background - Architectural Perspective

- Two main approaches
  - **Proxy-based**
  - **IP integration**
- Proxy-based
  - Adaptable to almost any WSN
  - Needs protocol conversion - e.g. sink
    (uses dedicated WSNs protocols)
- IP integration
  - IP inside the smart sensor node
  - Enables technology homogeneity
  - Takes advantage of years of research on IP
Challenges

- Proxy-based approach preferred
  - IP not suited for WSNs
- Large header overhead
  - IPv4 - minimum of 20 bytes
  - IPv6 - 40 bytes
- Global addressing scheme
  - WSNs are typically data-centric
  - IP is address centric, needs global addressing scheme at the network level

Challenges (cont.)

- Limited bandwidth
  - Very limited bandwidth of WSN wireless communication - up to 250kbps
  - More bits -> more MAC delays and transmission times
- Limited node energy
  - Very important on many WSN scenarios
  - Bits cost energy
- Implementation
  - IP complex for 8/16 bits micro-architectures and very low memory devices
  - IEEE 802.15.4 has 102 bytes payload, IP can reach 1280 bytes or more
- Transport protocol
  - TCP - even more communication overhead
  - UDP - no reliability guarantees
Motivation

- **Why IP support on all LoWPAN nodes?**
  (Low-power wireless personal area networks)
- IP-based technologies already exist, are well known and proven to be working
- The pervasive nature of IP networks allows use of existing infrastructure
- Take advantage of existing tools for network management
- Technology homogeneity with the Internet
  - Seamless connection
  - “Internet of Things”
- Individual smart sensor node addressing
  - Adequate for some scenarios - e.g. BSNs

Internet of things approaches

It is foreseeable that any object in the near future will have an Internet connection - this is the Internet of Things vision. All these objects will be able to exchange and process information, most of them characterized by small size, power constrained, small computing and storage resources. In fact, connecting embedded low-power devices to the Internet is considered the biggest challenge and opportunity for the Internet. There is a strong trend of convergence towards an Internet based solution and the 6LoWPAN may be the convergence solution to achieve the Internet of things vision.

- ZigBee
- Machine-to-machine communications
- The Future Internet
- Web of things
- Wireless sensor networks
Internet of things approaches

- There are a strong trend of convergence towards a Internet-based solution to connect all Internet of things solutions.
- The 6LoWPAN may be the convergence solution

Related Research Efforts on IP-over-WSNs

- Challenges exist to be tackled
- IP is widely used and known
- Several published papers
- Operating systems support
  - TinyOS
  - ContikiOS
- IPv6 advantages
  - Larger addressing space
  - Stateless auto-configuration
The beginning...

- Breaking the implementation myths
  - Header compression
  - Link-layer addresses (IPv6)
  - Stateless auto-configuration (IPv6)
  - Fragmentation support
  - uIP - IP for 8-bit micro-architectures
  - lwIP - IP for 16-bit architectures

Takeoff

- Some networks
  - among others...
Dedicated Stacks

- Research around dedicated IP stacks for sensor nodes
- Using 6LoWPAN
  - blip - TinyOS
  - sicslowpan - ContikiOS
  - Atmel, Jennic and Sensinode, Arch Rock implementations
- Other approaches

IPv6 vs IPv4

- IPv6
  - Low impact on bits transmitted
  - Added addressing space
  - New features
  - IPv6 wins!

6LoWPAN

- IETF working group
- Bring IPv6 to WSNs powered by IEEE 802.15.4 radios
- Specification of an intermediate layer
  - Between IPv6 (network) and IEEE 802.15.4 (MAC)
- Features
  - Header compression
  - Fragmentation support
- Mailing list presents great activity

G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, Transmission of IPv6 Packets over IEEE 802.15.4 Networks, RFC number 4944, IETF, 2007
  - Information on header compression, frame format, IPv6 local-link addresses

Resources - Software

- Main operating systems
  - TinyOS
  - ContikiOS
- Both support IP
  - blip - TinyOS
  - uIP(v6) - ContikiOS
- Both support 6LoWPAN
- Several hardware available
TinyOS

- Open source
- Berkley university - http://www.tinyos.net
- nesC programming language
  - Component-based
  - Learning curve for C programmers
- TOSIM simulator
- One of the most used operating systems for embedded devices worldwide
- blip - Berkley IP
  - IP over WSNs, with or without 6LoWPAN
  - Formerly b6lowpan
  - Linux virtual machine

blip

- Performs IP over WSNs on TinyOS
- With 6LoWPAN support
- Features
  - IPv6 neighbor discovery
  - Default route selection
  - Point-to-point routing
- Support for
  - ping6, tracert6, and nc6
  - TCP still experimental
- BSD sockets API
- Tested on Micaz, Telos and epic platforms
ContikiOS

- Open source
- Swedish Institute of Computer Science – http://www.sics.se/contiki
- C programming language
- Protothreads
  - Thread-like approach with shared stack
  - Memory efficient
  - Auto-start, start on request
- Events
  - System and programmer-defined
  - Wait for an event
- Timers
  - Event timers

ContikiOS (cont.)

- Two communication stacks
  - Rime and uIP(v6) (the world’s smallest open source compliant IPv6 stack, for Contiki)
- COOJA Java simulator
  - recently updated with dramatic performance increase
- MSPSim standalone MSP430-based simulation
  - suitable for one node firmware testing
  - several resources namely memory usage, stack view and
- Linux virtual machine - over Ubuntu 8.04LTS - more flexible
uIP6

- IPv6 over WSN
  - More than IEEE 802.15.4 - interface independent architecture
- With 6LoWPAN support - sicslowpan
  - IPv6 over IEEE 802.15.4
  - Fragmentation and header compression support
- Features
  - TCP, UDP
  - IPv6 addressing
  - ICMPv6
  - Neighbor Discovery
- Tested on Atmel Raven, sky/Telos, Micaz, Sensinode and others
- From ContikiOS 2.2.3 (currently 2.3)

Resources - Hardware

- Crossbow as a “core” manufacturer
- Other manufacturers
  - Sensinode
  - Jennic
  - Sentilla
- Two main processor platforms
  - TI MSP430
  - Atmel ATMega
- IEEE 802.15.4 radios dominate
Resources - Hardware (some motes)

<table>
<thead>
<tr>
<th>Name</th>
<th>Processor</th>
<th>RAM(Program)</th>
<th>Transceiver</th>
<th>Operating System</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chirp4IRIS</td>
<td>Atmel ATmega128</td>
<td>8 KB/128 KB/128 KB</td>
<td>Arrow F230</td>
<td>TinyOS</td>
<td>$15.75</td>
</tr>
<tr>
<td>Chirp4TEL42A</td>
<td>Atmel ATmega89C52</td>
<td>4 KB/128 KB/128 KB</td>
<td>TI CC2420</td>
<td>TinyOS</td>
<td>$20/30</td>
</tr>
<tr>
<td>Chirp4TEL43B</td>
<td>Atmel ATmega89C52</td>
<td>4 KB/128 KB/128 KB</td>
<td>TI CC2420</td>
<td>TinyOS</td>
<td>$115/125/75</td>
</tr>
<tr>
<td>Crossbar</td>
<td>Microchip</td>
<td>32 MB/32 MB/32 MB</td>
<td>TI CC2420</td>
<td>TinyOS</td>
<td>$290/300</td>
</tr>
<tr>
<td>TelosB2</td>
<td>Texas Instruments</td>
<td>10 KB/8 KB/1 MB</td>
<td>TI CC2420</td>
<td>TinyOS</td>
<td>$90-129</td>
</tr>
<tr>
<td>Amet. AVR</td>
<td>Atmel ATmega128</td>
<td>128 KB+250 KB</td>
<td>TI CC2420</td>
<td>TinyOS</td>
<td>$8 (6k)</td>
</tr>
<tr>
<td>Semtech</td>
<td>Texas Instruments</td>
<td>5 KB/5 KB/5 KB</td>
<td>TI CC2420</td>
<td>Proprietary</td>
<td>$220/270</td>
</tr>
<tr>
<td>Crossbar</td>
<td>Texas Instruments</td>
<td>32 MB/32 MB/32 MB</td>
<td>TI CC2420</td>
<td>Proprietary</td>
<td>$220/270</td>
</tr>
<tr>
<td>Crossbar</td>
<td>Texas Instruments</td>
<td>64 MB/64 MB/64 MB</td>
<td>TI CC2420</td>
<td>Proprietary</td>
<td>$220/270</td>
</tr>
<tr>
<td>San JOFT</td>
<td>ARM922T</td>
<td>512 KB/14 MB</td>
<td>Integrated in</td>
<td>Proprietary</td>
<td>$259/100</td>
</tr>
</tbody>
</table>

Resources - TelosB

- All-in-one platform
  - micro-controller (MSP430)
  - IEEE 802.15.4 compliant radio (CC2420)
  - USB port - FTDI chip
  - 10KB RAM/48KB program/1MB general purpose
  - autonomous power supply - 2AA batteries
  - 3LEDs - red, green and blue
  - sensors - temperature, humidity and light
- Previous versions
  - sky
  - Telos
- Supported by both TinyOS and ContikiOS
Resources - Summary

- RAM
  - 4KB-10KB (typically)
- Flash ROM
  - 48KB-128KB (typically)
- Several combinations of software/hardware
  - TelosB supports both OS’s
- ContikiOS gaining popularity
  - uIPv6 contribution
  - Works very well with TelosB

Deploying IPv6-enabled WSNs

- Choose an operating system
  - TinyOS, ContikiOS or other
  - Examples available to start playing with
- Choose an hardware platform
  - Many platforms available
- Develop solution
  - Linux OS preferred for development - virtual machines available
- Use suitable simulation tools
  - COOJA/MSPSim/TOSIM
- Test/deploy on target hardware
Conclusions

• IP on WSNs is a reality!
• Internet of Things has emerged
• Crossbow TelosB hardware
  • Support from both TinyOS and ContikiOS
  • All-in-one platform
  • A good platform to start with
• Other platforms emerge (Raaven Atmel, ...)
• New platforms
  • Sometimes, port is not mature enough

Conclusions (cont.)

• Deployment of IP-based WSN possible today
  • With off-the-shelf components and open source OS
• What’s the best approach?
  • Available WSNs - proxy-based
  • New deployments - depends on the network scenario
• IPv6 over WSN
  • ContikiOS takes the lead
  • But TinyOS also supports it
Some more references


Internet Connectivity for Wireless Sensor Networks

Joel Rodrigues
joelr@ieee.org

Thanks for your attention!