A SDN solution for Wireless Sensor Networks

DIEEI
University of Catania

http://sdn-wise.dieei.unict.it
Outline

- Motivations
- Related work
- SDN-WISE
- Prototype and testbed
- Performance evaluation
- Conclusions and future work

http://sdn-wise.dieei.unict.it
Motivations
A few facts about wireless sensor networks

- Mature technology since early 2000s
- Challenging communication & networking environment
- Requirements extremely application specific

The bottom-line...

There is nothing like a one-fits-all solution

**Upsides:**
- Large number of solutions proposed
- Deep understanding of the WSN domain
- Zillions of papers, citations, academic promotions, projects

**Downsides:**
- High solution specialization
- Market fragmentation
- Burden on application developers
- Low reusability

http://sdn-wise.dieei.unict.it
The consequence...

Always there...

It’s not taking off!

http://sdn-wise.dieei.unict.it
Our objectives

1. Overcome fragmentation
2. Ease life of developers
Related work

http://sdn-wise.dieei.unict.it
Software Defined Networking (SDN) clearly separates:

- **Data plane**: run by network Switches
- **Control plane**: implemented by a software program running on a server (the Controller)

Modifying the behavior of the network as easy as it is installing a new piece of software on a PC

**OpenFlow** is the most popular implementation of the SDN paradigm

http://sdn-wise.dieei.unict.it
SDN in WSNs

- Few attempts to extend SDN to WSNs:
  - Software Defined Wireless Networks (SDWN), 2012
  - Sensor OpenFlow, 2012

- Different requirements:
  - **WSNs**
    - Efficiency
    - Flexibility
    - Memory occupancy
  - **Traditional wired networks**
    - Velocity
SDN-WISE

http://sdn-wise.dieei.unict.it
SDN-WISE: Basic concepts

- Directly derived by OpenFlow
- Separation between
  - data plane (executed by sensor nodes)
  - control plane (executed by the Controller)
- When an event (e.g., the arrival of a packet) occurs sensor nodes behave as specified in the WISE Table
- If there is no relevant information in the WISE Table → Ask the Controller
- The Controller replies sending a new entry for the WISE Table
- A simple protocol defined to allow nodes to:
  - Learn the shortest path towards the (closest) sink(s)
  - Discover the neighboring nodes
  - Periodically report local information to the Controller (through the sink)
# WISE Table

<table>
<thead>
<tr>
<th>Matching Rule</th>
<th>Matching Rule</th>
<th>Matching Rule</th>
<th>Action</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>= 2 0 2 B</td>
<td>= 1 1 0 x_{Thr}</td>
<td>= 1 1 0 1</td>
<td>Modify 1 1 0 1</td>
<td>122 23</td>
</tr>
<tr>
<td>= 2 0 2 B</td>
<td>≤ 2 0 10 x_{Thr}</td>
<td>= 1 1 0 1</td>
<td>Modify 1 1 0 0</td>
<td>122 120</td>
</tr>
<tr>
<td>= 2 0 2 B</td>
<td>- 0 - -</td>
<td>- 0 - -</td>
<td>Forward 0 0 0 D</td>
<td>122 143</td>
</tr>
<tr>
<td>= 2 0 2 A</td>
<td>= 1 1 0 0</td>
<td>- 0 - -</td>
<td>Drop 0 0 - -</td>
<td>100 42</td>
</tr>
<tr>
<td>= 2 0 2 A</td>
<td>= 1 1 0 1</td>
<td>- 0 - -</td>
<td>Forward 0 0 0 D</td>
<td>100 32</td>
</tr>
</tbody>
</table>
SDN-WISE Architecture

http://sdn-wise.dieei.unict.it
Major features (compared to OpenFlow)

1. Statefulness
2. Flexible definition of rules
3. Support of duty cycles
4. Support of multitenancy (beyond slicing)
5. Lots of deployment options and programming languages
6. Integration with simulation environments (OMNET++ & OPNET)

http://sdn-wise.dieei.unict.it
Statefulness

- OpenFlow is stateless
- SDN-WISE is stateful: a buffer of memory is reserved for state information
  - Rules can state info to classify packets in flows
  - Actions can modify state info
- Why? Reduce the number of interactions with the Controller if local policies must be applied
- 3 exemplary uses of the state...

http://sdn-wise.dieei.unict.it
Exemplary use of the state (1)

- Conditional forwarding:
  - C must forward packets from A only, if the values coming from B are higher than a threshold
Exemplary use of the state (2)

- Support of QoS:
  - A congested node must give different priorities to different flows
  - Level of congestion stored as state information
  - Different drop probabilities given to different flows in the WISE-table in case of congestion

http://sdn-wise.dieei.unict.it
Exemplary use of the state (3)

- Multipath routing

If $s = 0$, fwd to A and set $s = 1$

If $s = 1$, fwd to B and set $s = 2$

If $s = 2$, fwd to C and set $s = 0$

http://sdn-wise.dieei.unict.it
Flexible definition of rules

- Rules consider:
  - \( \leq 3 \) windows (\( \leq 2 \) byte) in the packet (in any position), or
  - any portion (\( \leq 2 \) byte) of the memory for state
  - any relational operator (\( =, <, >, \leq, \geq, \neq \), Kalman filters)

- Slower than OpenFlow but higher efficiency and more sophisticated programmability

http://sdn-wise.dieei.unict.it
Support of duty cycle

- The set of actions has been enlarged to support duty cycle
- It is possible to turn the radio off for a certain time interval

http://sdn-wise.dieei.unict.it
Multitenancy (beyond slicing)

- Slicing assigns each packet to **only** one tenant.
- In WSN the same piece of data can be of interest of several applications.
- **WISE-Visor** a new layer which abstracts the real network and creates (different) views for different tenants.
- At each node a packet belongs to all tenants that agree on its treatment.
- When there is a disagreement, a new copy of the packet is created.

http://sdn-wise.dieei.unict.it
Lots of deployment options and programming languages

Simple

Complex

IEEE 802.15.4 Tx/Rx (Sink)

PC (Adaptation, Virtualization, Controllers & Simulation)

Regular Nodes

Server (Controller 2)

SOAP

Internet

Server (Controller 1)

RMI

Embedded System (Adaptation for real network)

Sensor Nodes

UMTS Modem

IEEE 802.15.4 Tx/Rx (Sink)
Prototype and testbed

http://sdn-wise.dieei.unict.it
Prototype

- **Sensor nodes:** Embit IEEE 802.15.4 boards (EMB-Z2530PA)
  - 2.4 GHz ISM
  - Texas Instruments CC2530
  - Memory: 8 kB RAM + 256 kB Flash memory
  - 40 kB of memory used for MAC (TIMAC v.1.4.0)
  - 10 kB of memory used for SDN-WiSe

- **Control plane:** WISEVisor + Controllers hosted in the same PC
  - Intel(R) Core(TM) 2 CPU, 2.40 GHz
  - 4GB of RAM
  - Windows 7, 32 bit
  - Controllers implemented Java 7

http://sdn-wise.dieei.unict.it
Testbed

Sink
Sensor node

http://sdn- wise.dieei.unict.it
Performance results

http://sdn-wise.dieei.unict.it
Unicast RTT

Fig. 4. Unicast traffic: RTT as a function of the number of hops when transmitting 20 bytes of payload in static conditions.
Fig. 5. Unicast traffic: RTT as a function of the payload size in the case of one hop and static conditions.
Fig. 7. Multicast traffic: Average RTT as a function of the payload size.
Multicast RTT

Fig. 8. Multicast traffic: Average PLR as a function of the payload size.
Conclusions and future work

http://sdn-wise.dieei.unict.it
Conclusions

- SDN-WISE is a SDN solution for WSNs
- SDN-WISE has several specific features designed to achieve efficiency in WSNs
- A prototype of SDN-WISE has been implemented and is available for download: http://sdn-wise.dieei.unict.it
- EuWIN facility (Newcom#) has been exploited to run experimentation
- Performance comparison has been carried out with respect to ZigBee and 6LOWPAN

http://sdn-wise.dieei.unict.it
Current work

- Implement a framework for in-network processing in SDN-WISE
  - MapReduce approach
- Implementation in Contiki
- Integration with Open Network Operating System (ONOS)

http://sdn-wise.dieei.unict.it