Facilitating sensor deployment, discovery and resource access using embedded web services

Isam Ishaq, Jeroen Hoebeke, Jen Rossey, Eli De Poorter, Ingrid Moerman, Piet Demeester
{isam.ishaq, jeroen.hoebeke, jen.rossey, eli.depoorter, ingrid.moerman, piet.demeester}@intec.ugent.be
www.ibcn.intec.ugent.be

Internet Based Communication Networks and Services (IBCN)
Department of Information Technology (INTEC)
Ghent University - IBBT
Using the Things of the Internet

What devices can I use?
What services do they offer?

Internet

Sensor Network
CoAP
UDP
6LoWPAN

Proprietary Protocols

HTTP
TCP
IP

GW
GW
GW
Deployment Challenges

Integration with standards (DNS, etc)

User friendly access

Addressing & Subnet assignment

Routing

Resource Discovery

Manual interventions
Use of embedded web services

• Task:
  • Resource Discovery.
  • Self-organizing, minimum human intervention.
  • User-friendly (e.g. from a web browser).
  • Enable machine-to-machine (M2M) communication.

• Solution:
  • Use embedded web services for discovery and self-configuration of devices
  • Create browsable hierarchy:
    • networks
    • sensors
    • sensor resources
Advent of Embedded Web Services - CoAP

Q. Why can’t we just use Internet Protocols inside the Sensor networks?
A. Too complex for constrained devices.

- IETF Constrained RESTful Environments (CoRE) working group: Realize a Representational State Transfer (REST) architecture suitable for constrained nodes.
- Constrained Application Protocol (CoAP):
  - realizes a subset of REST that is common with the HTTP.
  - optimized for M2M applications.
Resource Discovery using CoRE Link Format

1. Default entry-point: "/.well-known/core" returns list of resources on a CoAP server.
2. Client can send further requests to obtain the value of a certain resource.
Requirements

Internet IPv6

LAN
test.ibbt.be
/48

CoAP

SGW

WSN

Sensors know:
Short Name & Address

Reserved for sensor networks:
sensnetX.iot.test.ibbt.be
/64 subnets

Sensor Network
Sensor Discovery

- SGW sends a multicast (ff02::1) CoAP GET request for the resource /.well-known/serverInfo
- SGW discovers (short) addresses and names of all sensor nodes present in the network.
- SGW constructs complete IPv6 addresses and FQDN of all sensors and adds them to local DNS.
GET
[ff02::1]/.well-known/serverInfo

Answer
CONTENT_TYPE: COAP_MEDIATYPE_TEXT_PLAIN
PAYLOAD: AL|2|A|5|N|T|S|N|Sensor5
STATUS: RESPONSE RECEIVED
RESPONSE_CODE: 2.00 OK
CONTENT_TYPE: COAP_MEDIATYPE_TEXT_PLAIN
PAYLOAD: AL|2|A|3|N|T|S|N|Sensor3

Request
METHOD: GET
URI: [ff02::1]/.well-known/serverInfo
APP_ID: 0
MESSAGE_TYPE: NON

Response
COAP_OPTION_TOKEN: 141
STATUS: RESPONSE RECEIVED
CONTENT_TYPE: COAP_MEDIATYPE_TEXT_PLAIN
COAP_OPTION_CONTENT_TYPE: 0
PAYLOAD: AL|2|A|5|N|T|S|N|Sensor5
RESPONSE_CODE: 2.00 OK
APP_ID: 0
GET
sensorgw1.iot.test.ibbt.be/.well-known/servers

sensor3.sensnet1.iot.test.ibbt.be|sensor2.sensnet1.iot.test.ibbt.be|sensor5.sensnet1.iot.test.ibbt.be|sensor4.sensnet1.iot.test.ibbt.be
• IGW adds addresses & names of discovered SGWs to its local DNS.
• IGW installs a route to the sensor subnet.
• IGW adds SGW as the name server for the sensor network.
• If SGW have no subnet prefix, domain suffix or name, IGW will send them as a CoAP POST request to SGW.
Example:
Sensor name: sensor3
⇒ FQDN: sensor3.sensnet1.iot.test.ibbt.be
16-bit sensor address: 3
⇒ IPv6: 2001:6a8:1d80:201::3
**Self-organization process**

1. **Add SGW to DNS**
   - Add it as DNS server
   - Add route to sensor subnet

2. **Create unique name**
   - Store info in local DNS

3. **List of known CoAP server addresses**
   - `coap://[ff02::1]/.well-known/ServerInfo`

4. **List of known CoAP server addresses**
   - `coap://[ff02::1]/.well-known/Servers`

5. **Discovered SGWs**
6. **Discovered sensors**

7. **Sensor Access**
   - `coap://[sensor]/temperature`
   - `temperature`

8. **Sensor Discovery**
   - `coap://[SensorGW]/.well-known/Servers`
   - `coap://[IGW]/.well-known/Servers`

9. **Self-configuration**
   - `coap://[IGW]/.well-known/ServerInfo`
HTTP Access

1. HTTP-CoAP proxy functionality
   a) Non-Transparent:  
      http://gw_name:8080/sensor_name/resource => coap://sensor_name/resource
   b) Transparent:  http://sensor_name:8080/resource => coap://sensor_name/resource

2. Translate CoRE Link Format to HTML
Enabling HTTP Access

- `well-known/serverInfo`: GET, d="AddressLength|Address|NameType|Name"
- `cpuinfo`: GET, d="COAP Server CPU Info"
- `time`: GET, d="COAP Server time"
- `well-known/servers/coapFQDN`: GET, d="Discovered COAP Servers reachable via COAP link containing FQDN"
- `well-known/servers/coapIPv6`: GET, d="Discovered COAP Servers reachable via COAP link containing IPv6 address"
- `well-known/servers/addressIPv6`: GET, d="IPv6 Address of Discovered COAP Servers"
- `well-known/servers`: GET, obs, d="FQDN of Discovered COAP Servers"
- `well-known/servers/httpproxyIPv6`: GET, d="Discovered COAP Servers reachable via HTTP proxy containing IPv6 address"
Implementation and Deployment

Gateways implemented using **click router**. Sensor implemented using **IDRA** on **TinyOS**.

Footprint (MAC, AODV routing and CoAP)
- ROM: 37092 bytes
- RAM: 5923 bytes

Publicly reachable testbed (IPv6 only):
- http://coap.iot.test.ibbt.be:8080
- coap://coap.iot.test.ibbt.be/.well-known/core
Related work

- Subnet assignment
  - Dom. name assign.
  - routing to sensor GW
- Integration into DNS of the discovered sensors
- Discovery of the sensors in the subnet
- Usage & Access from the Internet
  - Pachube (no direct access)
  - OGC SWE framework
  - OSIRIS
  - Lightweight mDNS-SD
  - CoAP
  - [Schor et al.] REST-based, non-CoAP
  - [Schneider et al.] Proprietary, no direct access
Conclusion and Outlook

Conclusion:
The proposed approach provides a feasible and flexible solution to achieve hierarchical self-organization with a minimum of pre-configuration.

Outlook:
• Analysis of the overhead.
• Pull vs. Push.
• Port to Erbium/Contiki.
Questions?

Isam Ishaq, Jeroen Hoebeke, Jen Rossey, Eli De Poorter, Ingrid Moerman, Piet Demeester
{isam.ishaq, jeroen.hoebeke, jen.rossey, eli.depoorter, ingrid.moerman, piet.demeester}@intec.ugent.be
www.ibcn.intec.ugent.be

Internet Based Communication Networks and Services (IBCN)
Department of Information Technology (INTEC)
Ghent University - IBBT