Chatty Things

Making the Internet of Things Readily Usable for the Masses with XMPP

Roland Hieber

Institute of Operating Systems and Computer Networks, TU Braunschweig

Seminar Communications and Multimedia, WS 2013/14

Contents

Motivation

Technologies for Distributed Networking

Chatty Things

Related Approaches

Conclusion

Motivation

The IoT Vision

- plentitude of smart objects
- interoperability between devices
- easy accessibility for users

Motivation: Subgoals

But how do we...

- configure our devices?
- find other nodes to talk to?
- talk to other nodes or users?
- filter relevant information?

Address Allocation

IPv4 Link-Local Addessing (“APIPA”, “Zeroconf”, RFC 3927)

- subnet `169.254.0.0/16`

IPv6 Stateless Address Autoconfiguration (RFC 4862)

- subnet `fe80::/64` (link-local)
- subnet `fc00::/11` (unique-local, if configured)
- or global address (if configured)

Algorithm

1. choose (random) IP address in subnet
2. ask if anyone uses that address
3. if not, we’re fine
4. else, retry

Motivation: Subgoals

But how do we. . .

- configure our devices?
  - ✓ Link-Local Addressing, Stateless Address Autoconfiguration
- find other nodes to talk to?
- talk to other nodes or users?
- filter relevant information?

Multicast DNS (RFC 6762)

- distributed DNS database
- uses multicast address `224.0.0.251` (IPv4) and `ff02::fb` (IPv6), UDP port 5353
- standard DNS packet format
- hosts announce their own resources
- hosts respond to queries if queried resource is known

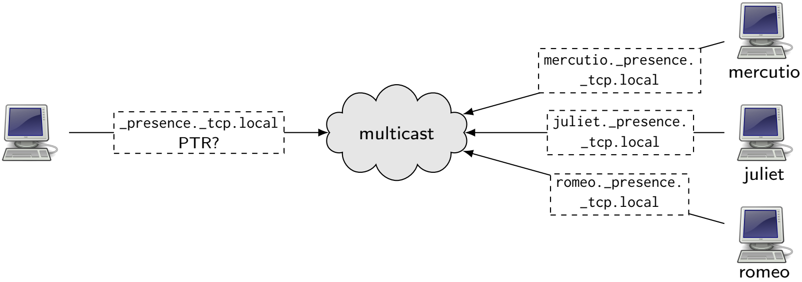
DNS-Based Service Discovery (RFC 6763)

Two-step process:

1. Service Instance Enumeration
   - query PTR records of form `_service._proto.domain`
   - results: instance names of form `name._service._proto.domain`
2. Service Instance Resolution
   - query instance names as SRV records
   - result gives host name, port, priority, weight

Example: mDNS + DNS-SD

1. Service Instance Enumeration



_presence._tcp.local
PTR?

multicast

mercutio._presence.
_tcp.local

juliet._presence.
_tcp.local

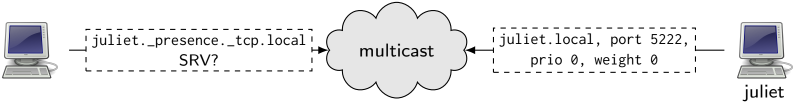
romeo._presence.
_tcp.local

mercutio

juliet

romeo

2. Service Instance Resolution



juliet._presence._tcp.local
SRV?

multicast

juliet.local, port 5222,
prio 0, weight 0

juliet

Motivation: Subgoals

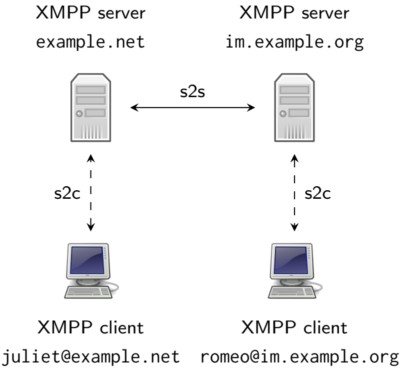
But how do we. . .

- configure our devices?
  - ✓ Link-Local Addressing, Stateless Address Autoconfiguration
- find other nodes to talk to?
  - ✓ DNS-SD + mDNS
- talk to other nodes or users?
- filter relevant information?

XMPP (RFC 6122)

Extensible Messaging and Presence Protocol

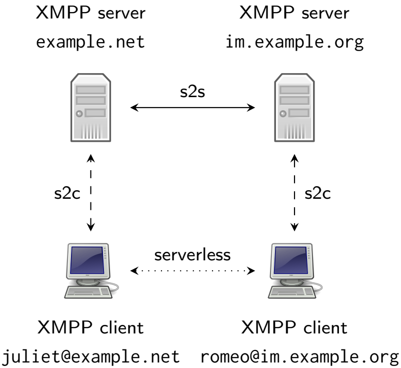
- XML-based
- Federated architecture
    - modeled after E-Mail
- publish-subscribe mechanism
- XMPP Extension Protocols (XEPs), e. g.
    - Multi-User Chats
    - Serverless Messaging
    - recently, also XEPs for the IoT



XMPP (RFC 6122)

Serverless XMPP (XEP-0174)

- Clients communicate directly, no server needed
- service discovery via mDNS/DNS-SD



XMPP (RFC 6122)

XEPs for the Internet of Things

- XEP-0323 Sensor Data
  - format for sensor data, query modes
- XEP-0324 Provisioning
  - defining access rights and user privileges
- XEP-0325 Control
  - get/set control prameters on a (group of) sensor node(s)
- XEP-0326 Concentrators
  - implement proxies for a subnet of the WSN

Motivation: Subgoals

But how do we. . .

- configure our devices?
  - ✓ Link-Local Addressing, Stateless Address Autoconfiguration
- find other nodes to talk to?
  - ✓ DNS-SD + mDNS
- talk to other nodes or users?
  - ✓ XMPP Serverless Messaging
- filter relevant information?

Chatty Things

- use serverless XMPP + mDNS + DNS-SD for communication
- interaction using a standard XMPP client
- prevent information overflow
  - "Traffic lights": status icon in roster represents threshold value
  - *Temporary Subscription for Presence (TSP)*

Prototype

- *uBonjour* for mDNS + DNS-SD
- *uXMPP* for XMPP
- 12 kB of ROM, 0.6 kB of RAM with Contiki on MSP-430

Temporary Subscription for Presence

Problem

- node must manually subscribe to get information
- users can move quickly out of the network
- subscriptions become outdated
- renewing/canceling subscriptions needs bandwidth
- data publishers also get updates

Temporary Subscription for Presence

Solution: Multi-User Chats

- create one chat room per topic
- users subscribe to information by entering the chat room
- server only sends information to nodes who want it
  - Chatty Things send a flag that they're uninterested

Drawbacks

- only works with central XMPP server
  - XEP-0045 is not (yet) specified for serverless XMPP
- XMPP server needs to handle TSP

Bootstrapping

At Boot

1. activate uBonjour
2. try to discover a central XMPP server
    - DNS-SD: `_xmpp-client._tcp.local`
3. if an XMPP server is discovered: *Infrastructure mode*
    - connect with ANONYMOUS login (XEP-0175)
    - join topic-based chats
    - deactivate uBonjour
4. if no server is found: *Ad hoc mode*
    - activate serverless messaging

Bootstrapping

During Runtime

- if server is lost, change to Ad hoc mode
- if new server is found in Ad hod mode, try changing to Infrastructure mode
  - if that fails, stay in Ad hoc mode

Motivation: Subgoals

But how do we...

- configure our devices?
  - ✓ Link-Local Addressing, Stateless Address Autoconfiguration
- find other nodes to talk to?
  - ✓ DNS-SD + mDNS
- talk to other nodes or users?
  - ✓ XMPP Serverless Messaging
- filter relevant information?
  - ✓ "Traffic Lights", Temporary Subscription for Presence

Related Approaches

Chatty Things

Feature	Chatty Things	
application gateways	-	
usable with standard clients	yes	
discovery support	yes	
IPv6/6LoWPAN ready	yes	
asynchronous messages	yes	
protocol overhead	moderate	

Related Approaches

Constrained Application Protocol (CoAP)

- binary mapping to HTTP
- UDP with confirmation and congestion control

Feature	Chatty Things	CoAP	
application gateways	-	yes	
usable with standard clients	yes	-	
discovery support	yes	yes	
IPv6/6LoWPAN ready	yes	yes	
asynchronous messages	yes	yes	
protocol overhead	moderate	low	

Related Approaches

MQ Telemetry Transport (MQTT)

- binary, only 2-byte header
- focused on M2M communication

Feature	Chatty Things	CoAP	MQTT	
application gateways	-	yes	yes	
usable with standard clients	yes	-	-	
discovery support	yes	yes	-	
IPv6/6LoWPAN ready	yes	yes	?	
asynchronous messages	yes	yes	?	
protocol overhead	moderate	low	low	

Related Approaches

Web Service for Devices (WS4D)

- SOAP (XML-based) over HTTP

Feature	Chatty Things	CoAP	MQTT	WS4D
application gateways	-	yes	yes	-
usable with standard clients	yes	-	-	(yes)
discovery support	yes	yes	-	yes
IPv6/6LoWPAN ready	yes	yes	?	partial
asynchronous messages	yes	yes	?	?
protocol overhead	moderate	low	low	high

Conclusion

Advantages of Chatty Things

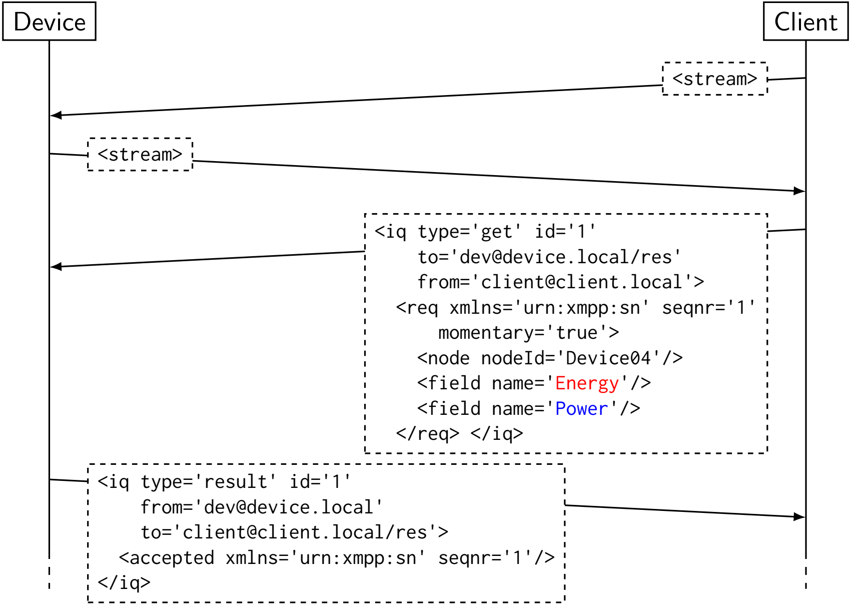
- no need for central infrastructure
- self-configuration and auto-discovery
- interaction over standard chat clients
- protocol flexibility for enhancements

Disadvantages of Chatty Things

- XMPP introduces some complexity
- topic filtering only possible with central server

Questions?

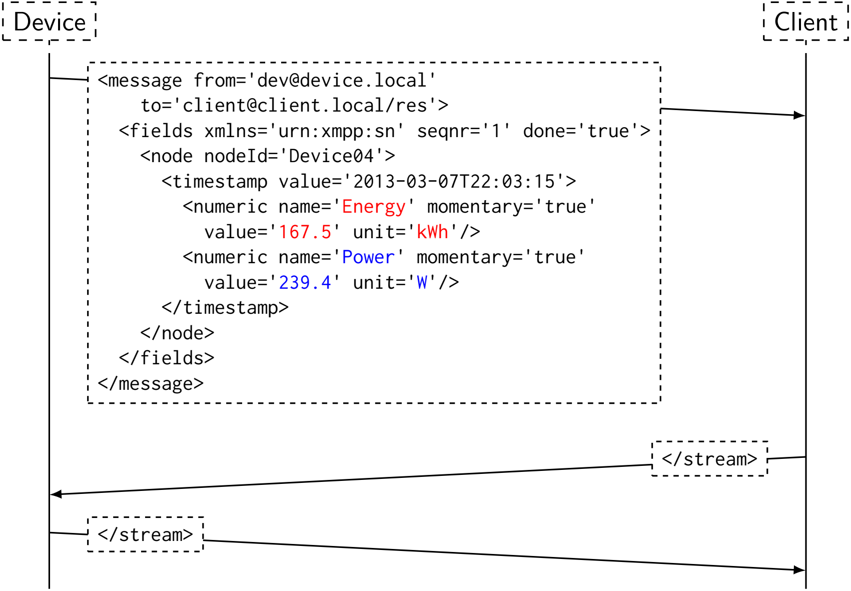
Example: Sensor Data



```
<iq type='get' id='1'
    to='dev@device.local/res'
    from='client@client.local'>
  <req xmlns='urn:xmpp:sn' seqnr='1'
      momentary='true'>
    <node nodeId='Device04'/>
    <field name='Energy'/>
    <field name='Power'/>
  </req> </iq>
```

```
<iq type='result' id='1'
    from='dev@device.local'
    to='client@client.local/res'>
  <accepted xmlns='urn:xmpp:sn' seqnr='1'/>
</iq>
```

Example: Sensor Data (cont.)



```
<message from='dev@device.local'
    to='client@client.local/res'>
  <fields xmlns='urn:xmpp:sn' seqnr='1' done='true'>
    <node nodeId='Device04'>
      <timestamp value='2013-03-07T22:03:15'>
        <numeric name='Energy' momentary='true'
          value='167.5' unit='kWh'/>
        <numeric name='Power' momentary='true'
          value='239.4' unit='W'/>
      </timestamp>
    </node>
  </fields>
</message>
```

```
</stream>
```

```
</stream>
```