A snapshot of my scientific career

Iva Bojic
University of Zagreb, Croatia
Faculty of Electrical Engineering and Computing
Collaborative downloading

Distributed self-organized energy-efficient model of service provisioning process for mobile network users
Outline of the presentation
Collaborative downloading

- Research motivation
- Main idea & system architecture
- Advantages & disadvantages
- Self-Organized Market Algorithm (SOMA)
- Case study
- Experimental evaluation
- Conclusions
- References
Research motivation
New business opportunities in telecommunication market

- Operators must shift their thinking from viewing connection to customers as a *dumb pipe* to a *smart pipe* [1]

- Services in Next Generation Network (NGN)
  - personalized
  - group-oriented and
  - energy efficient

Main idea
Efficient collaboration among mobile users

- Collaborative Downloading – service for peer-to-peer data exchange among mobile users
- Idea is to identify a set of $n$ similar mobile users and enable collaboration among them
- Only a part of requested content is downloaded using mobile network, the rest is shared using Bluetooth
System architecture
Individual vs. collaborative approach
Case study
Efficient collaboration among mobile users

- Service server
- Propose service
- Collaborative downloading
Advantages (1)
Energy efficiency on the client side

- Bluetooth-based communication consumes less energy than GPRS-based communication [2]

Advantages (2)
Energy efficiency on the server and network sides

- Service server processes less queries and is less loaded
- Server CPU and memory consumption is lower – service server consumes less energy and is more environmentally friendly
- Through GPRS network is transmitted less data – less network resources are used
Disadvantages
Longer time of the service provisioning process

➢ Service response time can be longer:
  o Bluetooth-based communication consumes more time than GPRS-based communication
  o decelerated speed for Bluetooth is higher, but test results show speed of only 10 - 11 KB/s [3]

➢ Semantic matchmaking and group formation – additional delay in the service provisioning process

Self-organized market algorithm

Two steps – market formation & auctions

- Market formation
  - every user downloads content parts from server and calculates the number of parts to download using \( u_j = w_1 * bc + w_2 * ss \) where \( w_1 \) and \( w_2 \) are constants, \( bc \) is battery and \( ss \) signal strength level

- Auctions
  - user possessing at least one content part acts as a seller and bids together with the other sellers
  - user missing at least one content part acts as a buyer and starts auctioning for the missing parts
Experimental evaluation
Evaluated in a real system environment

- Service server developed in Java
- Sony Ericsson W910i mobile phones
  - Java platform 8 (JP8)
  - Connected Limited Device Configuration 1.0
  - Mobile Information Device Profile 2.0
  - JSR180 (SIP API)
  - JSR82 (Bluetooth API)
  - JSR256 (Mobile Sensor API)
Almost 70% savings of the mobile devices’ energy

![Bar chart showing energy consumption comparison between individual and collaborative approaches for different numbers of users.]

- Individual approach:
  - One mobile user: 3.91 J
  - Two mobile users: 7.81 J
  - Three mobile users: 11.72 J
  - Four mobile users: 15.63 J

- Collaborative approach:
  - One mobile user: 3.91 J
  - Two mobile users: 4.33 J
  - Three mobile users: 4.74 J
  - Four mobile users: 5.16 J
Time consumption
Individual vs. collaborative approach

- 60% longer time of the service provisioning process

![Graph showing time consumption comparison between individual and collaborative approaches]

- Individual approach:
  - One mobile user: 40.25
  - Two mobile users: 45.13
  - Three mobile users: 53.08

- Collaborative approach:
  - Four mobile users: 187.25
  - Three mobile users: 115.83
  - Two mobile users: 98.88
  - One mobile user: 40.25
CPU consumption variance
Individual vs. collaborative approach

- $\text{CPU}_{\text{on_graph}} = \text{CPU}_{\text{individual\_approach}} - \text{CPU}_{\text{collaborative\_approach}}$
Memory consumption variance
Individual vs. collaborative approach

- Memory savings 550 Mb during 80 seconds

![Graph showing memory consumption variance over time for different numbers of mobile users.](image)
Conclusions
Collaborative downloading – advantages and disadvantages

- Distributed self-organized energy-efficient model of service provisioning process for mobile network users
  - Bluetooth-based vs. GPRS-based communication
  - Self-Organized Market Algorithm

- Collaborative downloading saves energy of mobile devices but also prolongs service time provisioning

- Collaborative downloading reduces CPU and memory consumption on the server side
Possible improvements
Substituting Bluetooth communication technology

➢ Using Bluetooth low energy (version 4)
  o 1 Mbit/s
  o <15 mA

➢ Using WI-FI
  o there is no consistent evidence to date that exposure to radio signals from Wi-Fi and WLANs adversely affects the health of the general population [4]

Energy consumption (data)
Different communication technologies

![Graph showing energy consumption for WiFi, BLE, and 3G downloads and uploads.](image)
Energy consumption (time)
Different communication technologies

![Graph showing energy consumption over time for WiFi download, BLE download, 3G download, WiFi upload, BLE upload, and 3G upload.]
References

Collaborative downloading


5) Bojic, Iva; Podobnik, Vedran; Petric, Ana: Swarm-oriented mobile services: Step towards green communication, Expert systems with applications, 39( 9), pp. 7874-7886 (2012)

New identification scheme that allows M2M devices to establish communication with M2M server or M2M devices
Outline of the presentation
Machine-to-machine management

- Introduction
- Research motivation
- M2M system architecture
- M2M standardization
- Technologies for M2M communication
- Proposed identifier
- Conclusions
- References
Introduction (1)
Machine-to-machine systems

- 50 billion devices in M2M systems until 2020
- Communication among different M2M devices
  - computers, mobile phones, tablets, sensors, smart grid networks, vehicles, medical equipment...
- Various applications based on M2M devices
  - smart metering
  - eHealth
  - connected consumer
  - automotive applications
  - city automation
Introduction (2)
Machine-to-machine systems
Motivation
Uniquely identifying M2M devices

- Various communication technologies
  - IP networks
  - ZigBee
  - Bluetooth
  - M-BUS
  - Wireless M-BUS

- Some M2M devices support more than one communication technology

- How to uniquely identify an M2M device?
Standardization organization ETSI’s goals

- defining common service layer for various M2M devices
M2M system architecture
Defined by ETSI
Identifiers in M2M system
Defined by ETSI

- Application Identifier (App-ID)
- SCL Identifier (SCL-ID)
- M2M Node Identifier (M2M-Node-ID)
- M2M Service Connection Identifier (M2M-Connection-ID)
- M2M Service Provider Identifier (M2M-SP-ID)
- M2M Subscription Identifier
### Communication technologies

**Used in M2M area network**

<table>
<thead>
<tr>
<th>Standard</th>
<th>Rate</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECT ULE</td>
<td>&lt; 500 kb/s</td>
<td>around 1.9 GHz</td>
</tr>
<tr>
<td>Z-Wave</td>
<td>&lt; 40 kb/s</td>
<td>around 900 MHz</td>
</tr>
<tr>
<td>ZigBee</td>
<td>&lt; 250 kb/s</td>
<td>around 900 MHz and 2.4 GHz</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>&lt; 2 mb/s</td>
<td>around 2.4 GHz</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>&lt; 54 mb/s</td>
<td>around 2.4 GHz and 5 GHz</td>
</tr>
<tr>
<td>Wireless M-BUS</td>
<td>&lt; 100 kb/s</td>
<td>around 870 MHz</td>
</tr>
<tr>
<td>NFC</td>
<td>&lt; 424 kb/s</td>
<td>around 13.56 MHz, 106 MHz, 212 MHz</td>
</tr>
</tbody>
</table>
Proposed identifier
Used in M2M area network

- Uniquely identifies an M2M device regardless of communication technology it uses
  - one sensor with multiple communication technologies
  - sends the same data by using different technologies
  - it is of variable length

![Diagram of identifier structure]

1st byte, technologies indicator

Address of 1st technology (optional)

Address of nth technology (optional)
Proposed identifier – example (1)
Used in M2M area network

- Libelium Waspmote sensor support of 8 different radio technologies including
  - ZigBee
  - Bluetooth

- Proposed technologies in indicator (1st byte)
  - Bit 0: Bluetooth – 48-bit IEEE 802 address
  - Bit 1: Wi-Fi – 48-bit MAC address
  - Bit 2: Wireless M-BUS – 6-bit address
  - Bit 3: ZigBee – 64-bit EUI address
Proposed identifier – example (2)
Used in M2M area network

- Bluetooth address: 48 bits
  - 0x0003190D0D7C

- ZigBee address: 64 bits
  - 0x0013A200406937A0

- Technologies indicator (1st byte): 8 bits
  - 10010000

- Total universal identifier length: 120 bits
  - 0x900003190D0D7C0013A200406937A0
Conclusions
M2M management – advantages and disadvantages

➢ Advantages
  o uniquely identifies M2M device regardless of the used communication technology
  o contains information for all available access technologies
  o small information overhead since it is variable

➢ Disadvantages
  o specifying order for all known technologies
  o constantly updating order when new communication technology appears
References

M2M management

1) Galetic, Vedran; Bojic, Iva; Kusek, Mario; Jezic, Gordan; Desic, Sasa; Huljenic, Darko: Basic principles of Machine-to-Machine communication and its impact on telecommunications industry, Proceedings of MIPRO, Rijeka, Croatia, pp. 89-94 (2011)

2) Kuna, Matko; Kolaric, Hrvoje; Bojic, Iva; Kusek, Mario; Jezic, Gordan: Android/OSGi-based Machine-to-Machine Context-Aware System, Proceedings of ConTEL, Graz, Austria, pp. 95-102 (2011)


5) Katusic, Damjan; Skocir, Pavle; Bojic, Iva; Kusek, Mario; Jezic, Gordan; Desic, Sasa; Huljenic, Darko: Universal Identification Scheme in Machine-to-Machine Systems, Proceedings of ConTEL, Zagreb, Croatia, pp. 71-78 (2013)

6) Bojic, Iva; Granjal, Jorge; Monteiro, Edmundo; Katusic, Damjan; Skocir, Pavle; Kusek, Mario: Communication and Security in Machine-to-Machine Systems, in press.

Dr. Iva Bojić

Milan, 23 June, 2014
Machine-to-machine synchronization

Biologically inspired model of self-synchronization and its modification to be applicable in M2M systems
Logical questions
How hard is it to give the answers?

- How hard it is to synchronize different clocks?

- Why do we need time synchronization?

- How can we achieve time synchronization?
Outline of the presentation

Machine-to-machine synchronization

- Research motivation
- Biologically-inspired computing
- Firefly-inspired synchronization
- Pulse-coupled oscillators (PCO) model
- Modification of the PCO model
- Experimental evaluation
- Conclusions
- Future work
- References
Research motivation

No global notion of time

- In distributed systems each node has its own internal clock and its own notion of time
- In practice these clocks drift apart accumulating errors over time
- Global notion of time needed for:
  - common resource sharing (e.g. channel)
  - depend events tracking (e.g. distributed databases consistency)
  - simultaneous events detection (e.g. data collection)
Time synchronization provides a common time scale for local clocks of nodes in distributed systems

<table>
<thead>
<tr>
<th>Name</th>
<th>Adjustability</th>
<th>Decentralization</th>
<th>Robustness</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Clock Reading</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global Position System (GPS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Time Protocol (NTP)</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Continuous clock synchronization</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Reference Broadcast Synchronization (RBS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pulse Couple Oscillators (PCO)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>


Dr. Iva Bojić

Milan, 23 June, 2014
Biologically-inspired computing
Biology applied in distributed systems

- Applying natural phenomena in artificial distributed systems using only a three step process

Investigating natural phenomena
- Bee dance [Von Frish, 1953]
- Ant behaviour [Van Vorhis Key and Baker 1982]
- Firefly synchronization [Buck and Buck, 1968]

Modeling natural phenomena
- Artificial Bee Colony [Karaboga, 2005]
- Ant Colony Optimization [Dorigo, et al. 1991]
- Phase coupled oscillators model [Kuramoto, 1975]
- Pulse coupled oscillators model [Mirollo and Strogatz, 1990]

Implementing models in distributed systems
- BeeAdHoc [Wedde and Farooq 2005]
- AntHocNet [Caro et al. 2004]
- Heterogeneous PCO [Bojic, 2013]
If oscillators are not coupled, their state variables change following only their own excitations

\[ x_i(t) = f(\phi_i) \]

- \( x_i(t) \) – state variable \( x_i \) value at time \( t \)
- \( f(\phi_i) \) – excitation evolution of oscillator \( i \)
- \( t_i^* \) – time when oscillator \( i \) flashes

If oscillators are coupled, state variable $x_i$ is adjusted upon the reception of flashes from the others.

$$x_i(t) = f(\varphi_i) + \sum_{j=1}^{N} \varepsilon_{i,j} g_{i,j}(t)$$

$\varepsilon_{i,j}$ – coupling constant

$$g_{i,j}(t) = \begin{cases} 
1, & \text{if } i \text{ is coupled with } j \text{ and } t = t_j^* \\
0, & \text{otherwise}
\end{cases}$$

Pulse coupled oscillators model

Limitations

➢ Pulse coupled oscillators model assumptions
  o oscillators are connected in a **fully-connected network**
  o oscillators cannot join or leave the network nor change their positions in the network (i.e. no **mobility**)
  o no oscillators with a **faulty** behaviour that desynchronizes the network
  o oscillators are the same (i.e. have **same frequencies**)
  o no delays in the message exchange among oscillators
Forms of time synchronization

Real synchrony

Time (seconds) - Oscillator number

Machine-to-machine synchronization
Forms of time synchronization

Frequency locking

- Oscillators run at approximately the same intrinsic frequencies, but that differences between their state variables values are not constant because of frequency fluctuations.

Synchronization window

- Maximal difference between all oscillator state variable values at the moments when they “fired”
Synchronization window
Frequency locking

Time (seconds)

Oscillator number

Machine-to-machine synchronization

Synchronization cycle length
Synchronization window width

Dr. Iva Bojić
Milan, 23 June, 2014
Heterogeneous systems
Machine-to-machine systems
Implementing PCO in real-world

Everything is perfect until it isn't

- Libelium Waspmotes
  - 10 sensors
  - 6 Xbee modules fully-connected
  - 2 Xbee + Bluetooth modules acting as gateways between two networks
  - 2 Bluetooth modules connected to "gateways"
int MAX_STEPS = 20;
int step = 0;

void loop() {

    step += 1;

    if(receiveBlinks()) {
        step += 1;
    }

    if (step >= MAX_STEPS) {
        sendBlink();
        Utils.setLED(LED0, LED_ON);
        delay(2000);
        Utils.setLED(LED0, LED_OFF);
        step = 0;
    }
}
void setup()
{
    while (connected == false && connectCounter < 5)
    {
        if (BT.createConnection("0003190D0D9C", "02") == 1)
        {
            connected = true;
            connectCounter = 0;
            USB.println("Connection established \n");
        }
        else
        {
            connectCounter++;
            USB.println("Connection not established \n");
        }
    }
}
Problem #2
How to collect distributed data?
Problem #3
Different synchronization cycle lengths

Oscillator number vs. Time (seconds)
Dynamic frequency adjustment

The main idea

a) time synchronization cycle length in theory
b) time synchronization cycle length in reality
c) time synchronization cycle length after the proposed mechanism for dynamic frequency adjustment was used
Successful synchronization
Mechanism for dynamic frequency adjustment
Synchronization precision
Not comparable with other mechanisms

- Implementation of the PCO model on Waspmotes
  - approximate precision of 48 milliseconds
  - implemented on app layer - Waspmote platform (8 MHz)
    - Bluetooth
      - receiving → 40 - 60 milliseconds
      - sending → up to 300 milliseconds
    - Xbee
      - receiving → 10 - 20 milliseconds
      - sending → 10 - 20 milliseconds

- Reference Broadcast Synchronization
  - approximate precision of 30 microseconds
  - implemented on phy layer - Berkeley Mica2 (10 MHz)
    - time needed for sending and receiving messages 0 do 100 milliseconds
Problem #4
Is this algorithm truly robust enough?
Cryptographic mechanism
Using simple XOR protection
Problem #5
Is this algorithm truly scalable enough?
Selective coupling mechanism
Implemented on the sender side

- Set a `thresholdSyn` value that is used as an indicator whether to send synchronization messages or not

**Algorithm 1** The proposed mechanism pseudo code

1. `probability = random.uniform(0,1)`

2. `synRatio = \frac{\text{synchronizedNeighborsNumber}}{\text{allNeighborsNumber}}`

3. **if** `(synRatio > thresholdSyn)` or `(probability < thresholdProbability)` **end if**

4. `sendMessages()`

5. **end if**
Quality measure
Short synchronization and low network traffic

- Quality measure

\[ q_m = \alpha \, t_s + (1 - \alpha) \, n_t, \]

- \( t_s \) – synchronization time scaled to \([0, 1]\)
- \( n_t \) – network traffic scaled to \([0, 1]\)
- \( \alpha \) – which of those two values is more important to reduce
Programming for real-world devices is like watching Photoshopped picture of celebrities

- everybody knows the reality is different but most of the time we just aren’t aware to which extent
Future work (I)
Research challenges

- Better representation of M2M systems in simulations
  - interdependent networks
  - multilayer networks

Future work (II)
Research challenges

- Better understanding of links between mathematical and physical models, simulations and real-world environments
  - interpreting mathematical and physical models
  - understanding their limitations
  - translating models into algorithms
  - implementing algorithms into simulated environments
  - implementing algorithms into real-world environment

Dr. Iva Bojić
Milan, 23 June, 2014
Machine-to-machine synchronization
Future work (III)

Research challenges

- Perhaps skip the steps described on the previous slide by using digital evolution
  - form of evolutionary computation
  - population of self-replicating computer programs exists in a user-defined computational environment
  - subject to instruction-level mutations
  - subject to natural selection

References

M2M synchronization


