POPWIN
Parallel Object Remote Programming for Wireless Network over IPv6

Pierre Leone
University of Geneva

Project in collaboration with Pierre Kuonen, EIA-FR/Fribourg
POP-C++ (POP-Java) is an **object-oriented** system for programming parallel applications.
## Project goal

The goal of the project is to develop POPWIN an **object-oriented system to program wireless sensor network**.

<table>
<thead>
<tr>
<th>EIA-FR/Fribourg</th>
<th>University of Geneva</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pierre Kuonen</td>
<td>Pierre Leone</td>
</tr>
<tr>
<td>Yao Lu</td>
<td>Cristina Muñoz</td>
</tr>
<tr>
<td>Model of programming</td>
<td>Network primitives</td>
</tr>
<tr>
<td>Implementation of POPWIN</td>
<td>Energy balance mechanisms</td>
</tr>
<tr>
<td>Application</td>
<td>IPv6 – 6LowPan</td>
</tr>
</tbody>
</table>
Network primitives

Executing an object requires some resources

- Computing resources
- Sensory resources – temperature, pressure, etc.

Classical resources might be extended resources particular to sensor network

- Location
- Geographical constraints
First problem: Given a set of sensors that communicate wirelessly, find the appropriate resources in the network.

General approach:

- Use the context of Publish/Subscribe system:
  - Publishers advertise about the resources they offer
  - Subscriber search for resources

- Cooperative system
  - Publishers and subscribers participate to the process of matching demands
  - All the nodes in the network participate
Matching Pub/Sub

The sensors are located in a plane
Matching Pub/Sub - Heuristic
Matching Pub/Sub - Heuristic
Matching Pub/Sub - Heuristic

Pub/sub matching
Directional Random Walk
Directional Random Walk
1. Select the node $v \in N_y$ such that the number of 2-hops paths from $x$ to $v$ is minimal, i.e.

$$\text{argmin}_{v \in N_y} |N_v \cap N_x|$$

2. Introduce a penalty to the nodes that are in $N_x$

3. Once a node is chosen add a random penalty
Directional Random Walk

The first time the node $y$ receives the notification it:

- Memorizes the trace of the path
- Memorizes the node $x$ from which it (first) receives the notification

Properties:

- Covers the entire graph because of the random penalty
- The backward path is loop-erased
3000 nodes, mean number of neighbors 15
3000 nodes, mean number of neighbors 15
Time to intersection

**Measure:** Two nodes start the process and synchronously process until intersection.
Time to intersection

- 1500 nodes, $r=0.04$, ratio=1
  - Mean: 132
  - Median: 96

- 2000 nodes, $r=0.04$, ratio=1
  - Mean: 94
  - Median: 72

- 2500 nodes, $r=0.04$, ratio=1
  - Mean: 77
  - Median: 58

- 3000 nodes, $r=0.04$, ratio=1
  - Mean: 65
  - Median: 51
Time to intersection
Comparison with RW

**Measure:** Comparison of the time to intersection against a pure Random Walk strategy
Time to intersection - RW

Mean: 921
Median: 633

Mean: 663
Median: 478

Mean: 132
Median: 96

Mean: 94
Median: 72
Time to intersection - RW

Mean: 578
Median: 415

Mean: 77
Median: 58
**Conclusion 1:** Directionality shorten the time to intersection
Time to intersection

Asynchrony

Measure: The publisher and subscriber are not synchronous, i.e. they are working at different speeds.
Impact of asynchrony on the intersection time

Mean: 264
Median: 192

Mean: 364
Median: 267

Mean: 602
Median: 431
Impact of asynchrony on the intersection time

Comparison with RW
Impact of asynchrony on the intersection time

**POPWIN project**

**Mean:** 1842  
**Median:** 1267

**Mean:** 264  
**Median:** 192

**Mean:** 2579  
**Median:** 1699

**Mean:** 364  
**Median:** 267
Impact of asynchrony on the intersection time

**Conclusion 2:** Cooperation shorten the time to intersection
Routing efficiency

**Measure:** We consider a permutation of the nodes and for each permutation we execute the process and compute the number of paths that pass through each node (the load).

Node number: 0 1 2 3 ... n

Nodes load
Nodes load

Max: 366  
Mean: 62  
Median: 54

Max: 330  
Mean: 64  
Median: 56

Max: 364  
Mean: 64  
Median: 56

Max: 463  
Mean: 63  
Median: 55
Nodes load
Comparison with RW
Nodes load - RW

- **Max:** 366
  **Mean:** 62
  **Median:** 54

- **Max:** 330
  **Mean:** 64
  **Median:** 56

- **Max:** 253
  **Mean:** 65
  **Median:** 62

- **Max:** 201
  **Mean:** 72
  **Median:** 72
Nodes load - RW

Max: 364  Mean: 64  Median: 56

Max: 463  Mean: 63  Median: 55

Max: 205  Mean: 79  Median: 79

Max: 200  Mean: 83  Median: 85

SmartWold Hearings - Hasler Stiftung March 2013

POPWIN project
Nodes load
Comparison with Shortest path
Nodes load - RW

Max: 366  
Mean: 62  
Median: 54

Max: 330  
Mean: 64  
Median: 56

Max: 899  
Mean: 79  
Median: 3

Max: 1068  
Mean: 65  
Median: 2
Nodes load - RW

Max: 364
Mean: 64
Median: 56

Max: 463
Mean: 63
Median: 55

Max: 1348
Mean: 65
Median: 2

Max: 1725
Mean: 65
Median: 2
**Conclusion 3:** The mechanism balances the number of path that pass through the nodes as efficiently as a Random Walk.

**Remarks:**

- In the classical setting routing a permutation is a tool to estimate lower/upper bound on the performance of routing algorithms.

- In our setting routing the permutation makes possible to match publication/subscription. A subscription from a node $x$ follows the path $x \rightarrow \text{perm}[x] \rightarrow \text{perm}[\text{perm}[x]] \rightarrow \ldots$ and, check at all intermediate nodes if the path to the publication is known.
Further short term research directions

• Investigate the performance where there are one publisher many subscribers.

• We need to find efficient heuristic to stop/restart exploration of the graph.

• Alternatively we plan to investigate the routing of a permutation with a local algorithm. One path starts and stops at the ‘right time’.

• Consider different graph structures, not necessarily modeling wireless networks.
POPWIN
Parallel Object Remote Programming for Wireless Network over IPv6

Pierre Leone
University of Geneva

Project in collaboration with Pierre Kuonen, EIA-FR/Fribourg
Comparison with previous works

- Directional Rumor Routing in Wireless Sensor Networks, the nodes are localized
- A Directional Gossip Protocol for Path Discovery in MANETs., estimation of the critical probability.
- Directed diffusion, ...
- Directional Gossip: Gossip in a wide Area Network
- Lightweight tracking algorithm
- Techniques based on Rendez-vous
- Directional work, gossip and broadcast
- Centrifugal random walk, node Sampling, uses a spanning tree