UNIVERSITY OF CAMBRIDGE

Dynamic Reprogramming of Mobile Wireless Sensor Networks

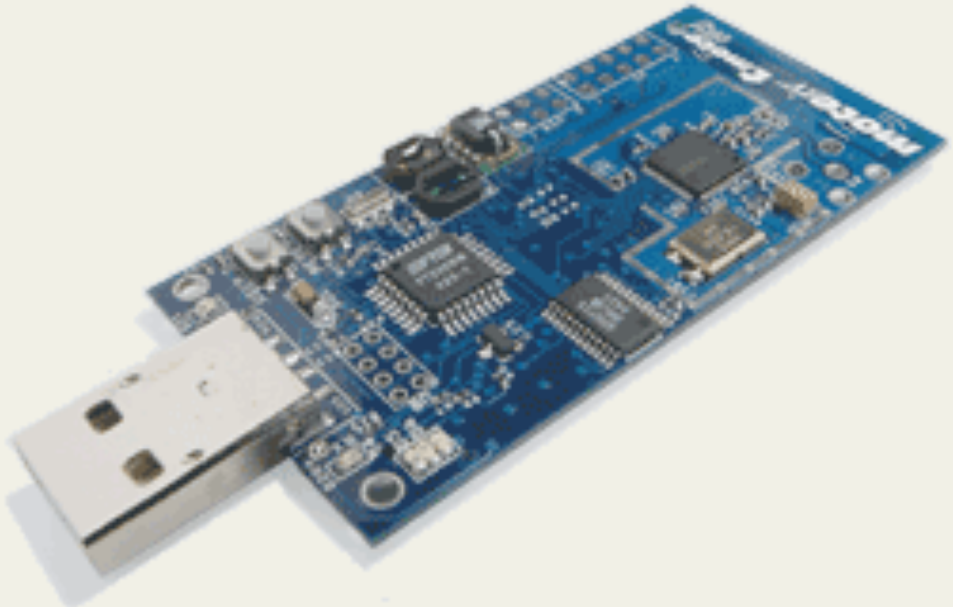
Bence Pásztor, Cecilia Mascolo

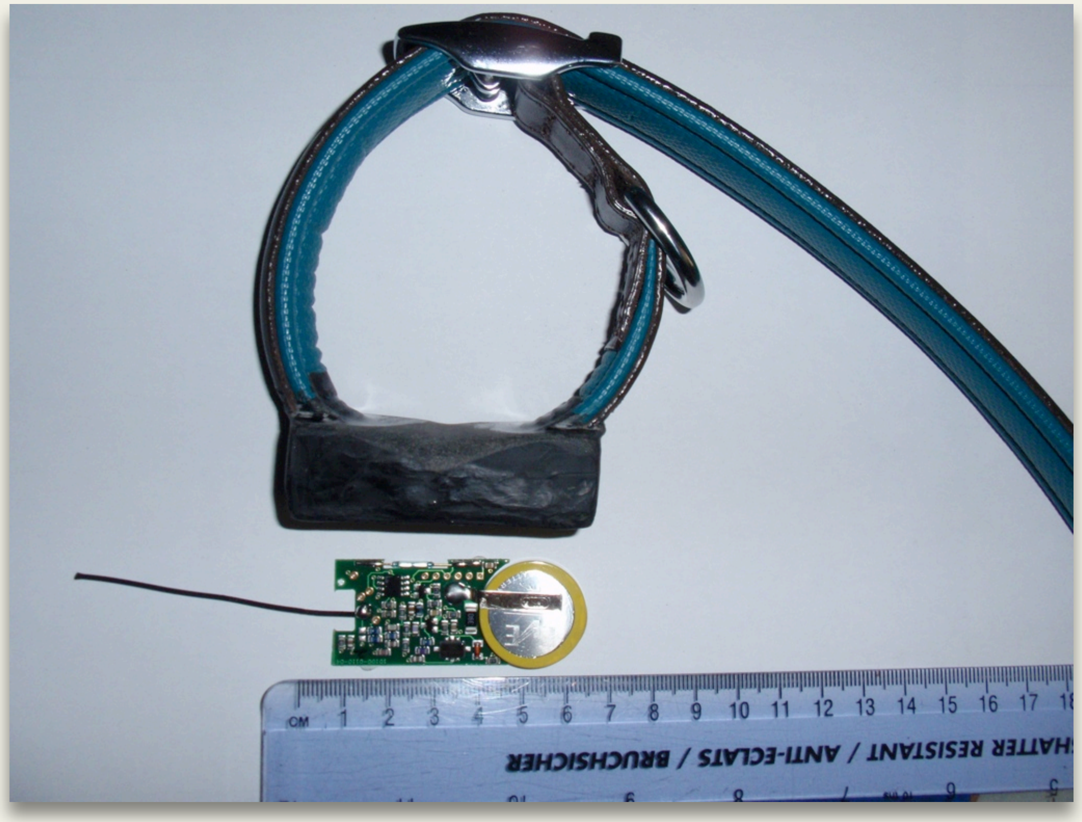
in collaboration with

Gian-Pietro Picco, Luca Mottola, David McDonald
and Bernie McConnell

Motivation

- Wireless Sensors: small, *very* constrained devices collecting information about the environment
- Capable of communicating with each other over short ranges



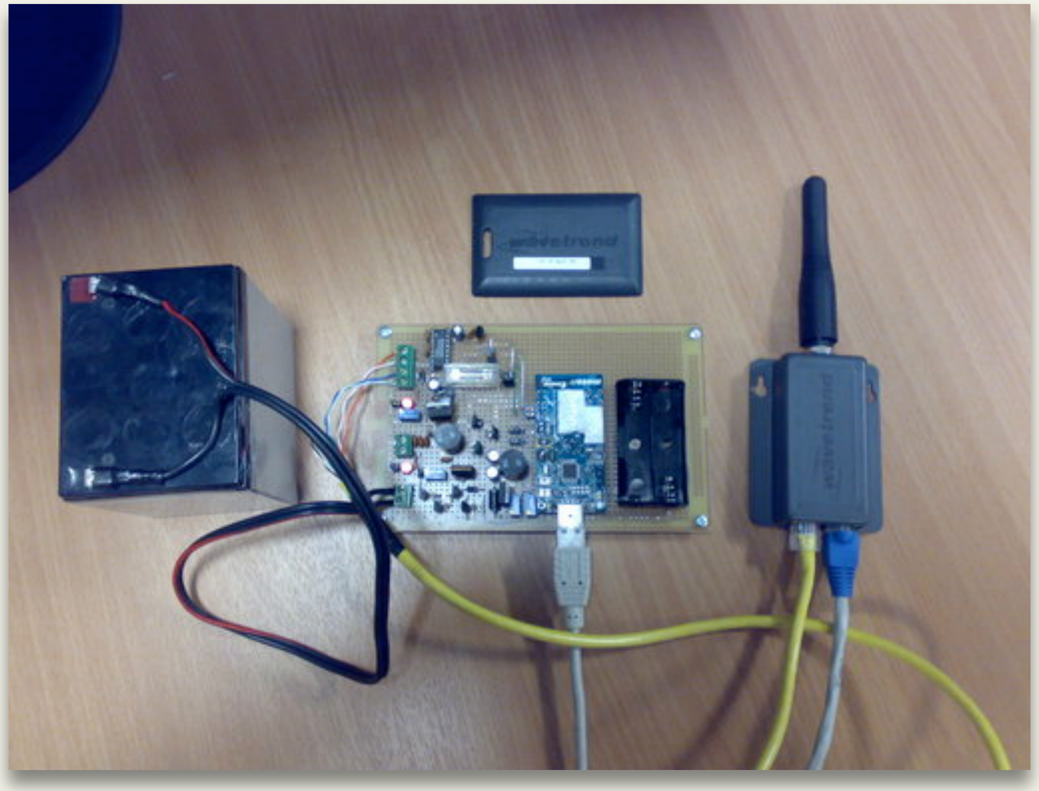


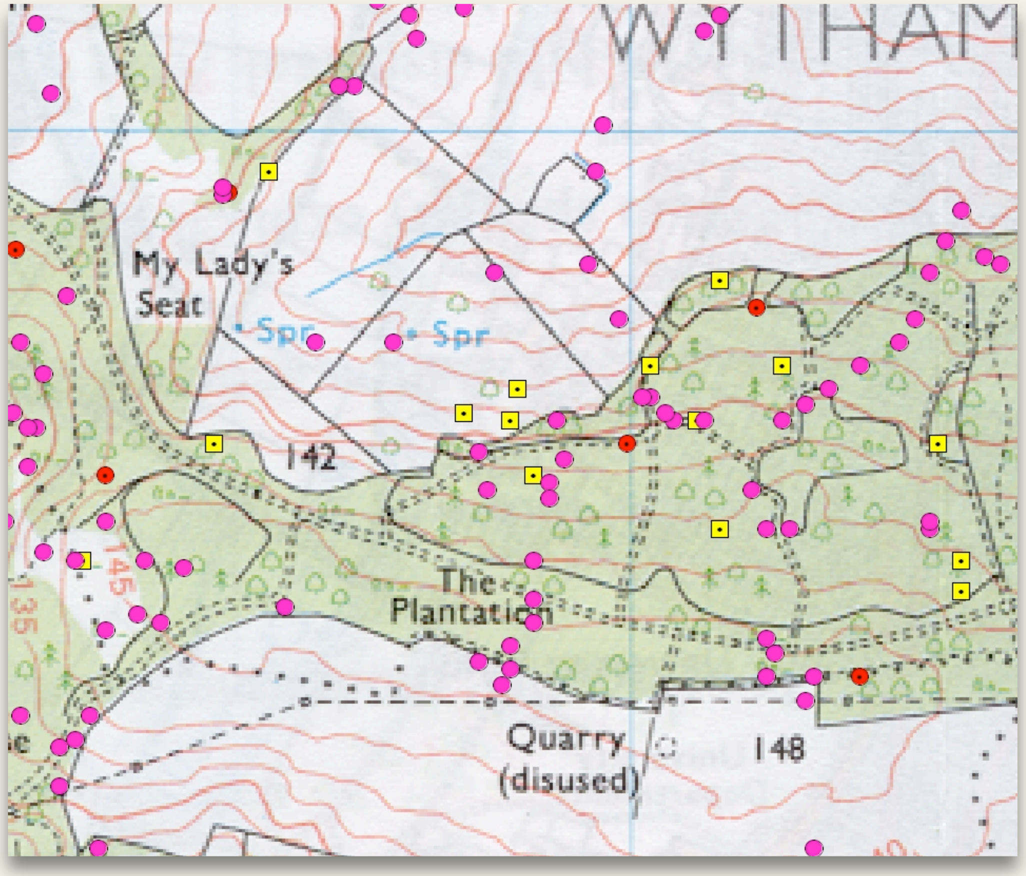
Wildlife monitoring

- Current technology is based on either GPS or VHF tracking
- It has been very difficult to track multiple animals for an extended period of time
- WildSensing Project: track badgers using RFID-WSN technology in Wytham
  - Collaboration with Computing Lab, University of Oxford and Department of Zoology, University of Oxford

WildSensing

- There are 28 RFID readers spread around the forest, capable of detecting a tag from about 20-30 m
- The data is stored on a sensor connected to the reader, and is delivered wirelessly to the enduser (zoologist)





WildSensing

- Currently, about 30 badgers carry active RFID tags in Wytham, Oxford
- RFID tags beacon about twice a second and last for about 2 years

Limitations

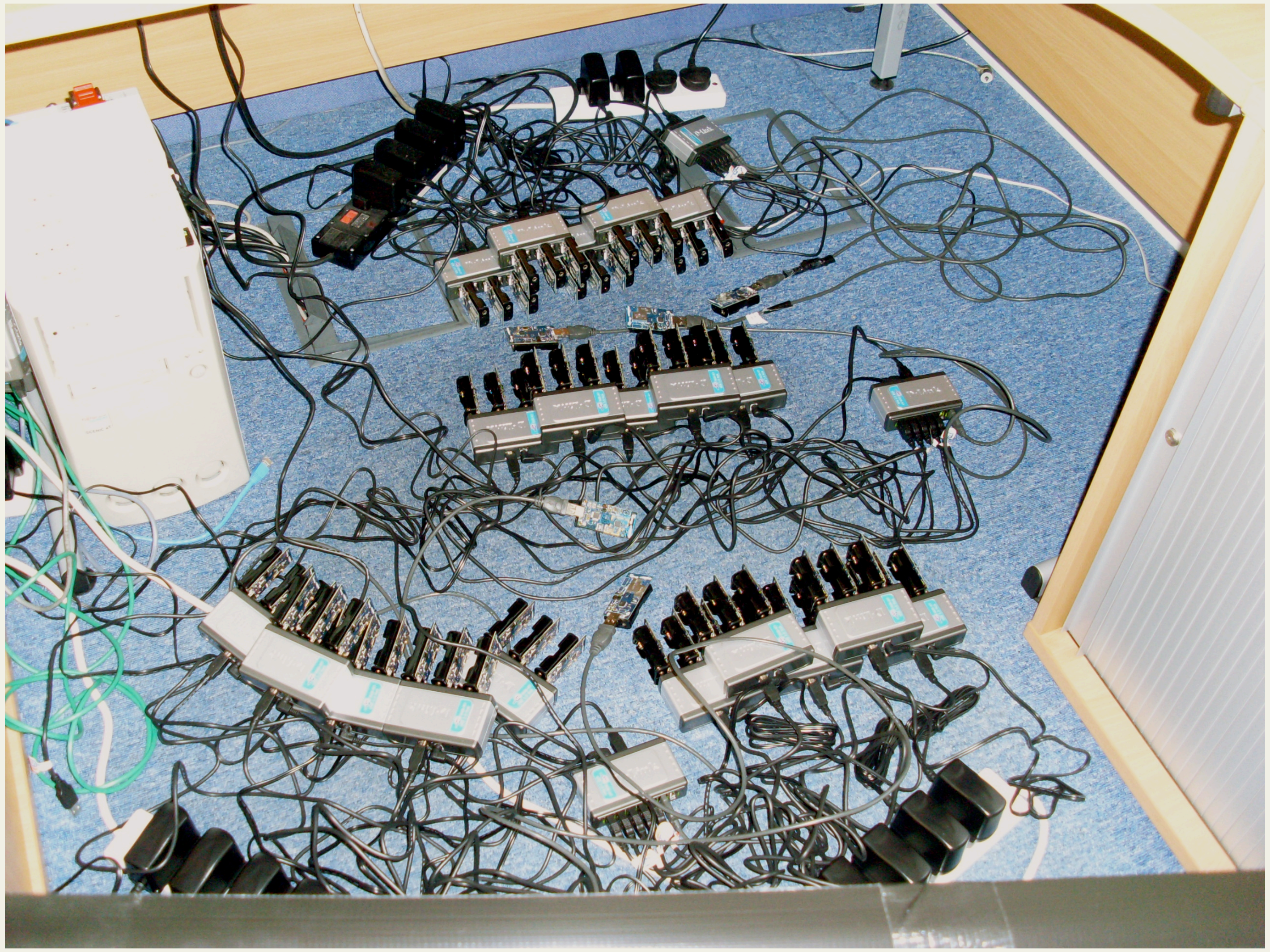
- Energy and memory constraints:
  - –both the memory gets full and the reader battery dies in about 2 weeks
  - –lot of effort to replace these
- not to mention bugs in the code...
- *The system is unable to log contacts between the animals*
  - > sensors are needed on the animals

Reprogramming

- One of the main difficulties with deployed sensor networks is *maintenance*
  - –reprogram sensors to fix bugs
  - –change parameters of a program or
  - –deploy a new program,
    e.g. due to new requirements

Reprogramming

- Usual method
  - –does not scale
  - –not possible when sensors are remote and/or are attached to animals moving around



- Current wireless solutions focus on static networks, and involve some kind of flooding, gossiping to disseminate code {Deluge, MNP, etc}

Mobile WSN

- Sensors are attached to animals, which roam around the forest
- Strictly not random, but predictable movements and colocations!
  - e.g. badgers use paths in the forest

Social Animals!

- Animals are social!
  - –they tend to stick together (better chances of survival)
  - –obvious example: families

- These social groups tend to be stable over time, and more importantly, *they spend a lot of time together, regularly*

Social dissemination

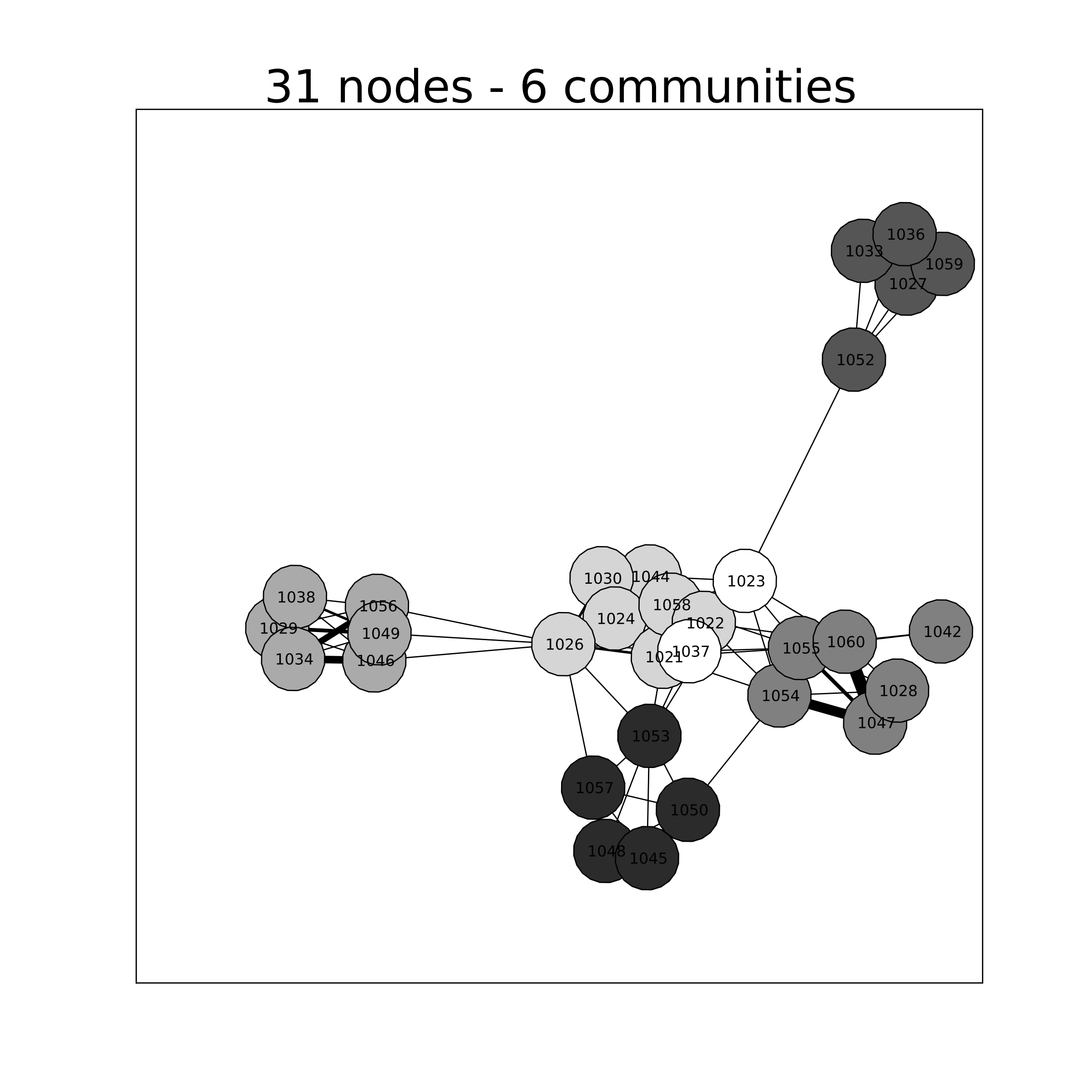
- Instead of flooding the network, let us try to use the social characteristics: *social groups, social links* between nodes, as well as *group leaders*
- Groups tend to stay connected - perfect for maintenance!
- Animals don’t behave the same - some are more active than others
  - *group leaders*, tend to be larger, male members of the community (it is safer for them to roam around...)

Basic Dissemination

- The protocol identifies the *social groups*, and differentiates between *group leaders* and *group members* based on contact-history/change degree of connectivity
- Leaders form the backbone, and deliver the code to the group
- They then wait until the group becomes connected, and broadcast the update

Clustering

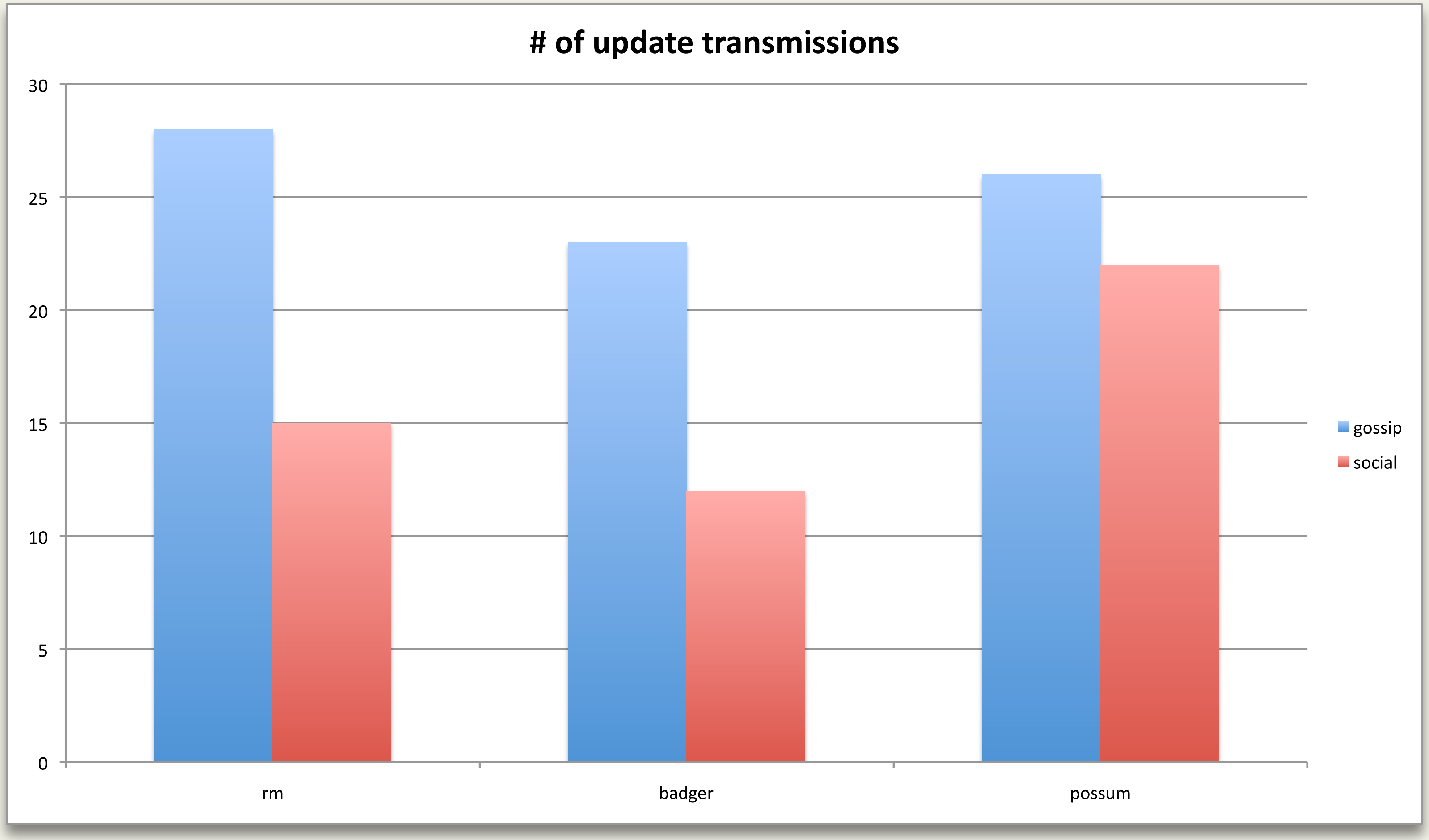
- Two nodes are in the same group if they spend relatively long time together
- Define a threshold: if nodes spend more than 50% of their time together, they belong to the same group
  - we can classify links between nodes!



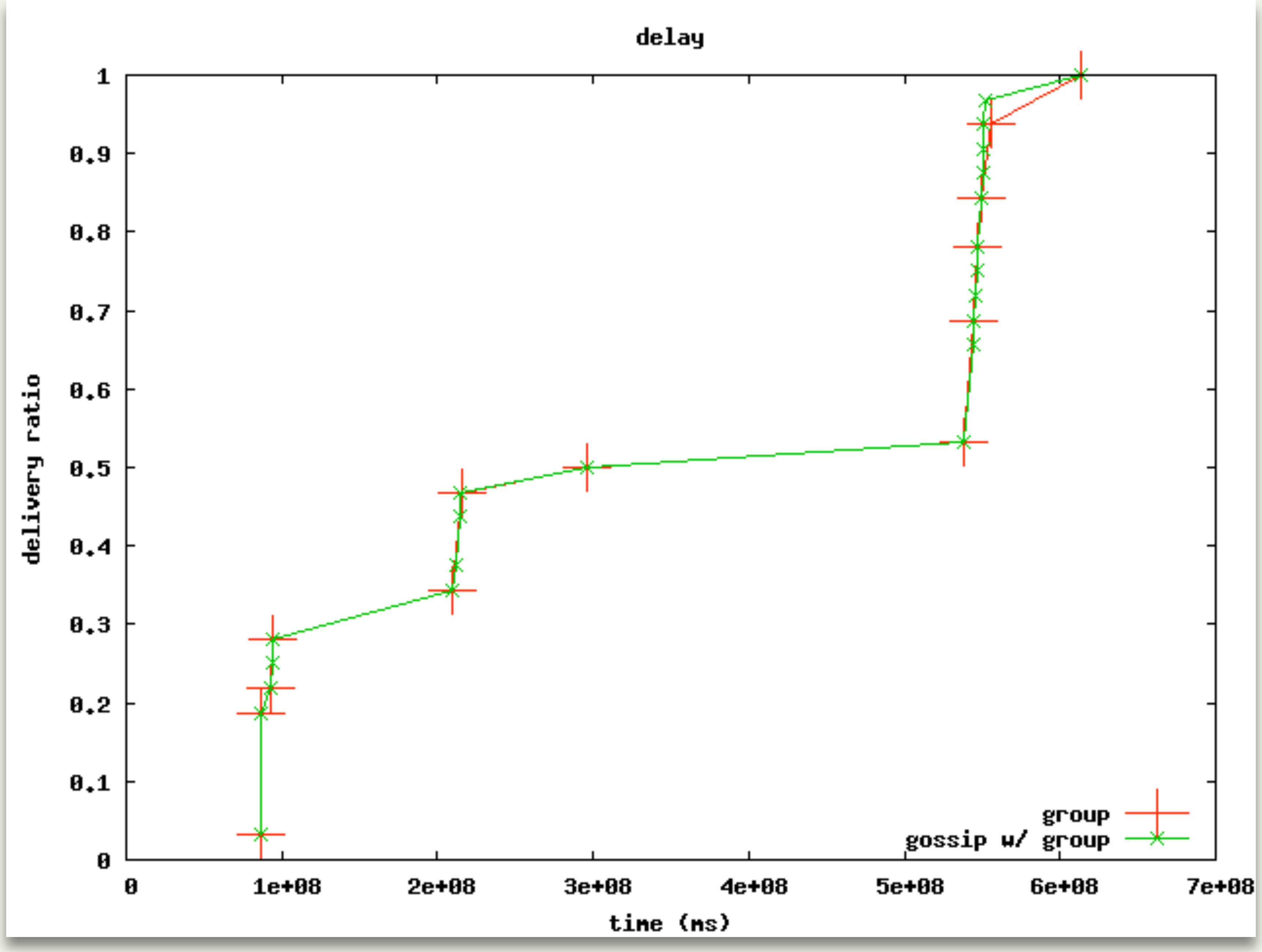
Graph from Salvatore Scellato @ CL

Initial Results

- around 50% less updates than a gossip protocol on badger/rm traces!



Delay

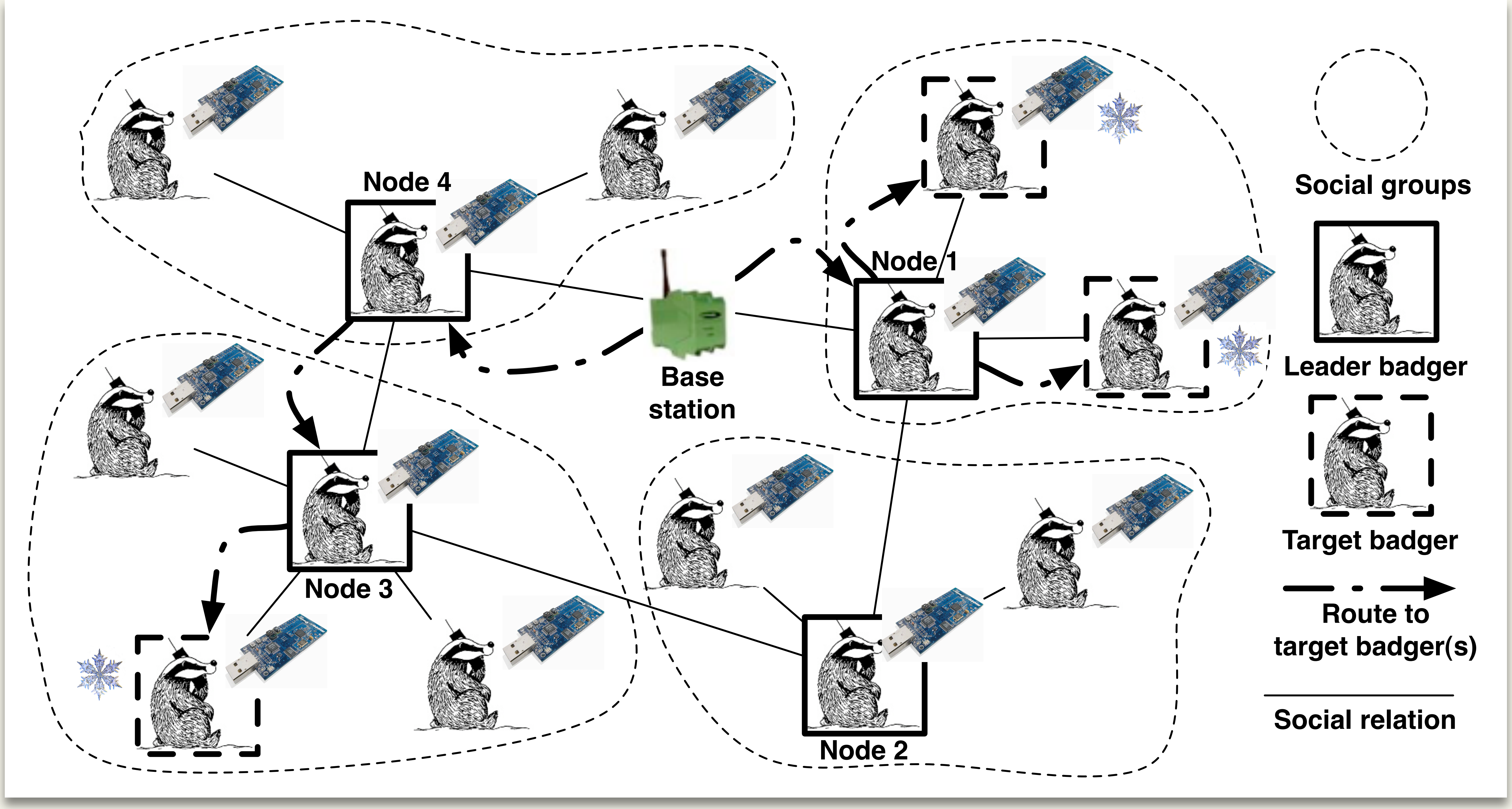

delay
delivery ratio
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1e+08
2e+08
3e+08
4e+08
5e+08
6e+08
7e+08
time (ms)
group
gossip w/ group

Extension: selective dissemination

- Future - deployed sensors should be shared/ reused:
  - a network of 100s of nodes can be shared between users, each running their own program. E.g. one collecting social information, while another environmental data
  - need a way to specify which sensors to update based on the user's interest

Programming model & dissemination

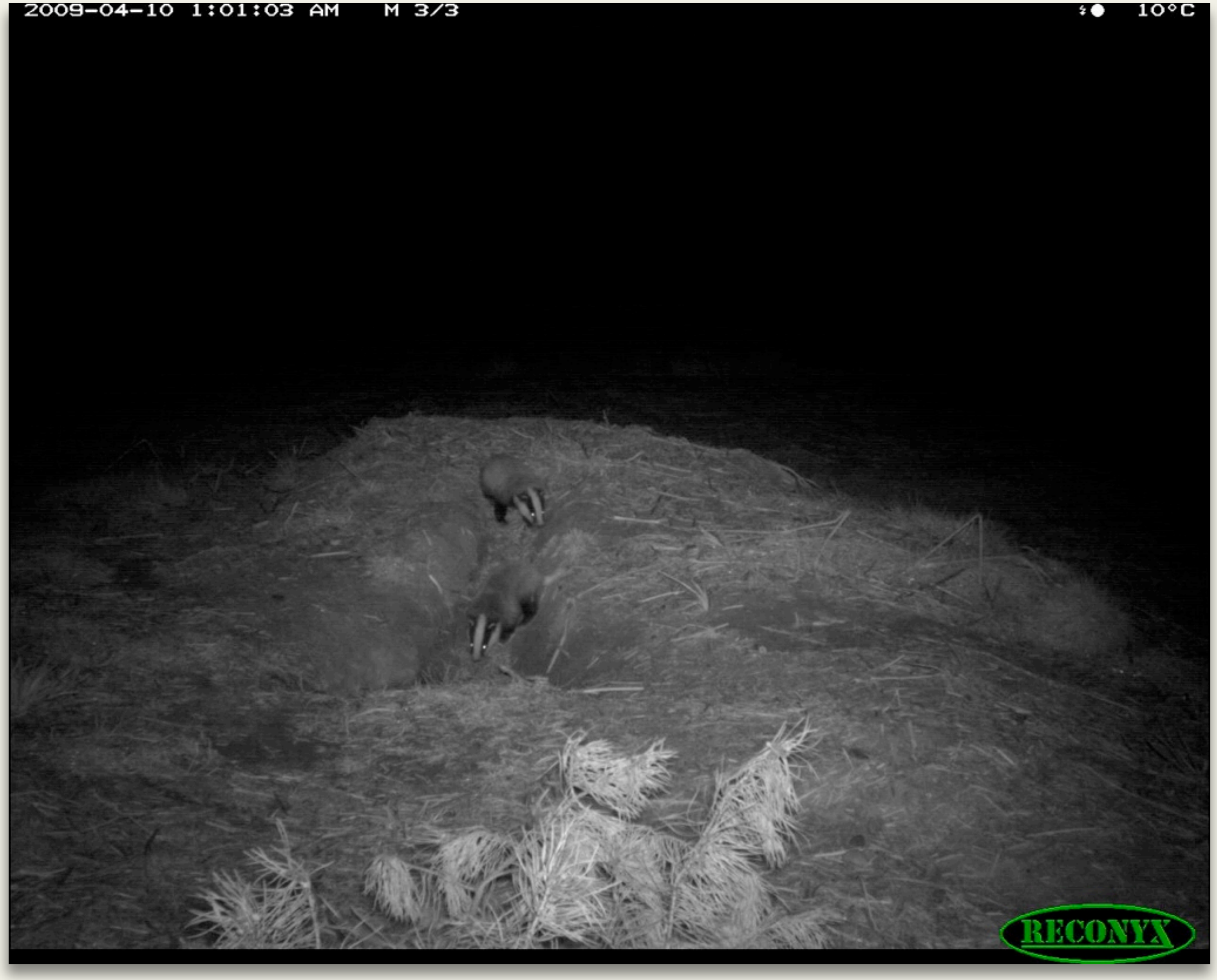
- characterize nodes with *attributes* describing some changing environmental condition (eg. temperature)
- let the user define *constraints* on the attributes to limit the dissemination of new code
  - i.e. only update nodes sensing a daily average temperature below 10 C
- use social dissemination to disseminate only to target nodes


Node 4
Node 1
Base station
Node 3
Node 2
Social groups
Leader badger
Target badger
Route to target badger(s)
Social relation

Current/Future direction

- Study animal traces to understand/improve the clustering algorithm
- Lots of potential in the clustering:
  - –duty cycling
  - –redundant processing detection
  - –routing
- Deploy it on badgers/sheep/seals;)
- Keep WildSensing running

Thanks!



www.cl.cam.ac.uk/~bp296

www.cl.cam.ac.uk/research/srg/netos/wildsensing/index.html

bence.pasztor@cl.cam.ac.uk