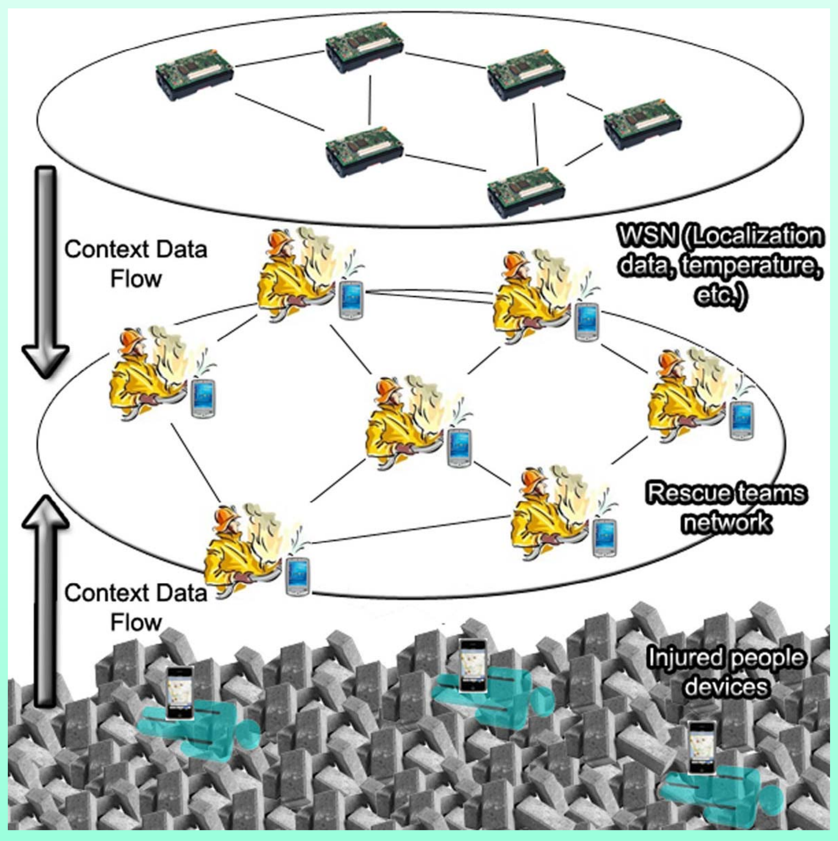
Towards Efficient and Reliable Context Data Distribution in Disaster Area Scenarios

Mario Fanelli[1,2], Luca Foschini[2], Antonio Corradi[2], Azzedine Boukerche[1]
[1] PARADISE Research Laboratory, SITE, University of Ottawa, Canada
[2]Dipartimento di Elettronica, Informatica e Sistemistica (DEIS), University of Bologna, Italy
{mario.fanelli, luca.foschini, antonio.corradi}@unibo.it, {mfanelli, boukerch}@site.uottawa.ca

Disaster Area Scenarios



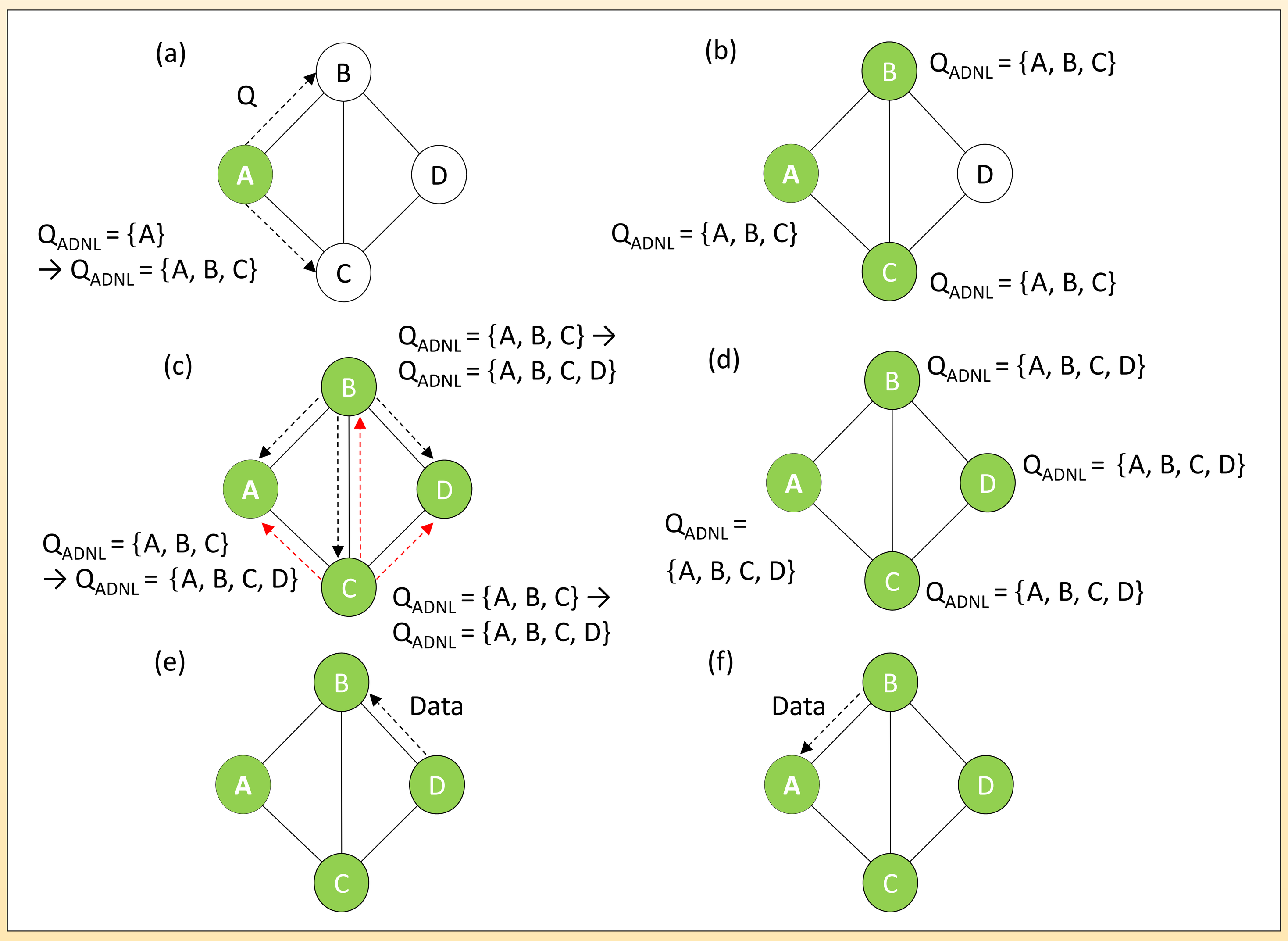
Disaster areas deal with unexpected and sudden disasters, such as *terrorist attacks* or *earthquakes*.
To increase the probability of saving human lives, different rescue teams need to coordinate in a timely, efficient and reliable manner.
If rescue teams are equipped with mobile devices, novel **context-aware services** can be realized to improve rescue operations efficiency.
In our main scenario, mobile devices compose a **M**obile **A**d-hoc **NET**work (**MANET**) useful to distribute and to retrieve interesting context data. At the same time, injured people can have mobile devices that spread useful context information, such as localization information, medical records, etc.

Context Data Management

To enable **context awareness in disaster areas**, we need to distribute and to retrieve **huge amounts of context data**. **Context data management** has to tackle different main issues :

- **Bandwidth-constrained** and **unreliable wireless** communications
- Congestion-prone communications due to the **high node density** and to the **huge amount of traffic**
- **Unreliable and distributed context data repositories** into the MANET

Context Data Routing With Time Guarantees



RECOWER context distribution is based on **context queries**. **Context queries** are distributed by using a **one-hop broadcast-based** approach, while **context data** are routed by using a **unicast-based** approach. If node D produces context data node A is interested in, the context distribution in above figure can be summarized as follows:

a) Node A broadcasts the query Q (with a TTL equal to 2) to all its current neighbors

b) Both B and C match Q against memorized data

c) Assuming that b) had not supplied responses, B and C store Q, and distribute it again after a **random delay less than a fixed Query Routing Delay (QRD)**

d) Query distribution ends due to the associated TTL. All the nodes have a stored copy of Q

e) When D receives the query from B, it matches Q with local data and, due to the positive match, schedules a data distribution after a **random delay less than a fixed Data Routing Delay (DRD)**

f) After another random delay, data are finally sent to A

Given a **maximum data retrieval time** and a **TTL,** a straightforward mechanism is applied to calculate both **QRD** and **DRD**. The random data (respectively, query) delay applied by each node is uniformly distributed into the range **[α; β]*DRD** (respectively, **QRD**), with **0 ≤ α < β ≤ 1**

Context Query Distribution Optimization

Each query has an **Already Disseminated Nodes List (ADNL)** parameter (called Q_{ADNL} in the figure below) to contain all the identifiers associated with the nodes that had already received the query. **Before distributing a query,** each node:

1. retrieves the list of its own one-hop neighbors
2. checks if at least one of them has not already received the query
3. if yes, appends the current neighbors' identifiers in Q_{ADNL}, and distributes the query again
4. otherwise, aborts the query distribution since all the current neighbors had already received the query

Context Data Replication Control

To increase **repositories diversity** in the same physical area, RECOWER associates each context data with a **unique key**, and **disseminates the list of locally memorized data keys**. Each node stores collected lists, and uses them to estimate data replication. **If a data has to be removed**, **we prefer the elimination of the data with the highest replication degree** in the one-hop physical neighborhood. In case of more data with the same replication degree, a simple Least Recently Used approach is used.

RECOWER

Reliable and **E**fficient **CO**ntext-aware data dissemination middle**W**are for **E**mergency **R**esponse

`http://www.lia.deis.unibo.it/Research/RECOWER/`

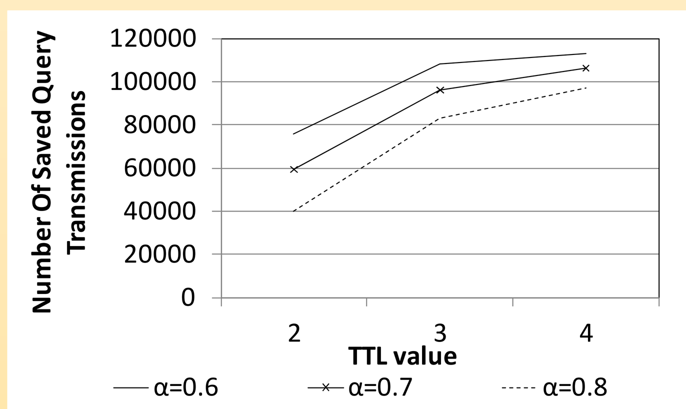
RECOWER eases the realization of context-aware services in disaster areas. The provided **context data distribution process** is mainly driver by **context data retrieval times**. RECOWER uses these constraints to self-adapt the routing delays, so to increase context data distribution efficiency and reliability**.**

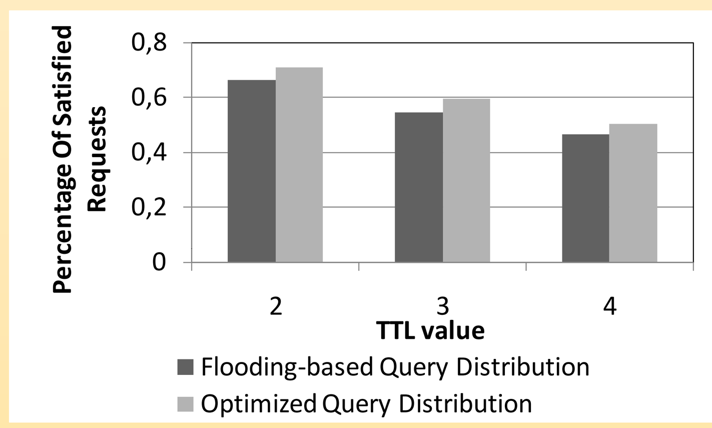
Both Q_{ADNL} and data keys lists are used to support simple membership tests. To reduce the run-time overhead, we exploit a probabilistic data structure, i.e., **Bloom filters,** to represent the needed management information.

Experimental Results

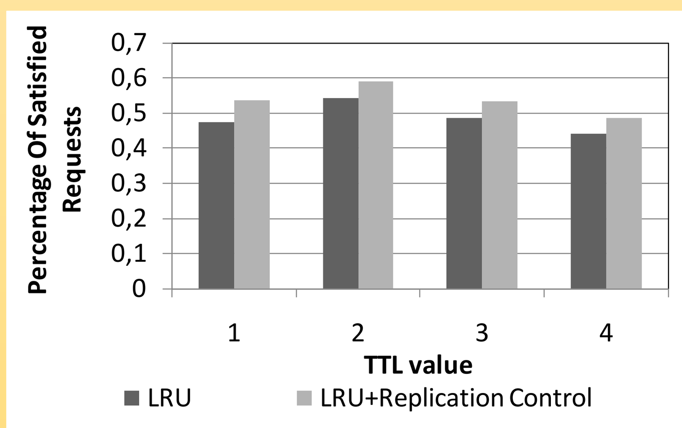
We implemented RECOWER and proposed distribution protocol in the simulator **ns2**. The main simulations parameters are:

- Simulation area of 350x350m, 50 mobile nodes, transmission range of 100m
- WiFi channel with 11 Mbits/s bandwidth
- Random Waypoint (RWP) with uniform speed in [1; 2] m/s and uniform distributed pause in [0; 10] seconds
- Each node has 10 local sources, and a context data repository of 30 elements. In addition, it emits 1 query/sec, and requires one of the 500 context data following a uniform distribution
- Each simulation lasts 600 seconds





The **optimized distribution process** reduces the number of distributed queries (figure on the left). This, in its turn, reduces the wireless channel congestion and increases the reliability of the data distribution process (figure on the right).



By means of **replication control,** our caching approach reduces the number of replicas and increases the context data cached in the same area. This, in its turn, positively affects the reliability of the distribution process, i.e., the total number of collected data.

Ongoing work

Encouraged by our results, we are currently working on **role-based techniques** useful to differentiate both the data and the distribution process quality. In fact, at run-time, different information, for instance associated with rescue teams and their own mobile devices, could be used to drive system reconfigurations, so to better tailor both data memorization and distribution.