COLLABORATIVE SELFISH NODE DETECTION WITH AN INCENTIVE MECHANISM FOR OPPORTUNISTIC NETWORKS

RADU-IOAN CIOBANU, CIPRIAN DOBRE, **MIHAI DASCĂLU**, ŞTEFAN TRĂUŞAN-MATU, VALENTIN CRISTEA

University Politehnica of Bucharest

INTRODUCTION

Problem:

- **Opportunistic networks (ONs)** – mobile networks based on the store-carry-and-forward paradigm
- **Selfish nodes** – nodes that don't want to participate in the routing process for various reasons:
  - Low resources
  - Fear of malicious data
  - Lack of interest in helping non-community nodes
- Delay/loss of messages

INTRODUCTION (2)

Proposed solution:

- Novel social-based collaborative content and context-based selfish node detection algorithm
- An **incentive** mechanism that rewards active nodes and punishes selfish ones
- Based on **gossiping**
- **Context-based:** social knowledge, battery level, etc.
- **Content-based:** message content-based decisions

RELATED WORK

Altruism model:

- **Community-biased distribution model** – people in a community have greater incentives to carry messages for other community members
- Altruism values distributed uniformly inside a community
- Two altruism values:
  - intra-community
  - inter-community

RELATED WORK (2)

Compare our algorithm with IRONMAN:

- Each node stores a perceived altruism value for other nodes
- A node is considered selfish if it doesn't relay a message it has been given
- Uses incentive mechanisms – messages belonging to nodes considered selfish aren't relayed anymore

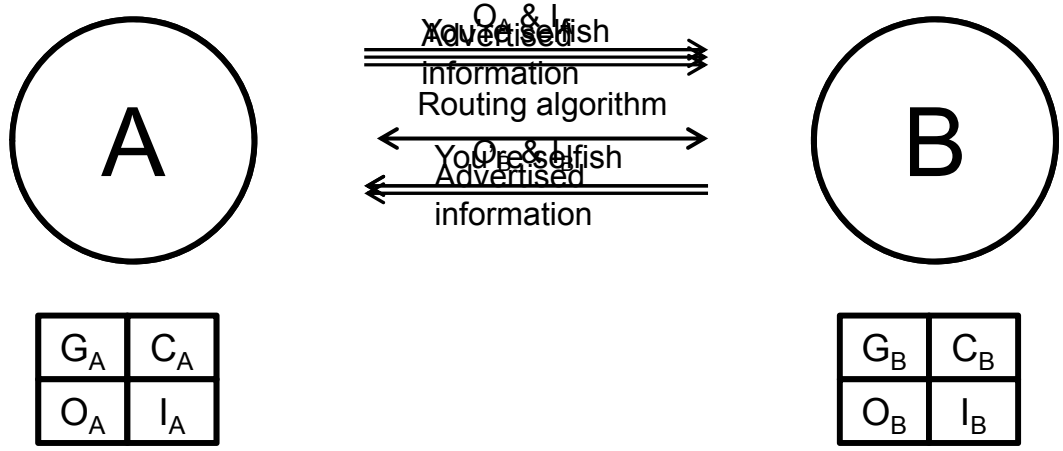
ALGORITHM

A node contains:

- Its own ID
- IDs of nodes in its own community
- Battery level
- Data memory split into four sections

G Messages generated by the node	**C** Messages carried by the node
O List of past forwards	**I** List of past receives

ALGORITHM (2)



1. Compute altruism towards B
2. Send O and I
3. Update O and I
4. Advertise specific information (battery level, messages metadata)
5. Compute perceived altruism for B
6. If perceived altruism is acceptable, apply routing algorithm
7. Else, consider B selfish and notify it
8. If considered selfish, increase altruism (optional)

1. Compute altruism towards A
2. Send O and I
3. Update O and I
4. Advertise specific information (battery level, messages metadata)
5. Compute perceived altruism for A
6. If perceived altruism is acceptable, apply routing algorithm
7. Else, consider A selfish and notify it
8. If considered selfish, increase altruism (optional)

ALGORITHM (3)

An altruism value for a node *N* and a message *m*:

$$altruism(N,m) = \prod_{o \in O, i \in I, o.m=i.m}^{N.id=o.d, N.id=i.s} type(m, o.m) \times thr(o.b)$$

A past encounter *x*:

- *x.m* – message that was sent
- *x.s/d* – source/destination node ID
- *x.d* – destination node ID

type -> 1 if two messages are of the same type, else 0

thr -> 1 if the battery ≥ preset threshold, else 0

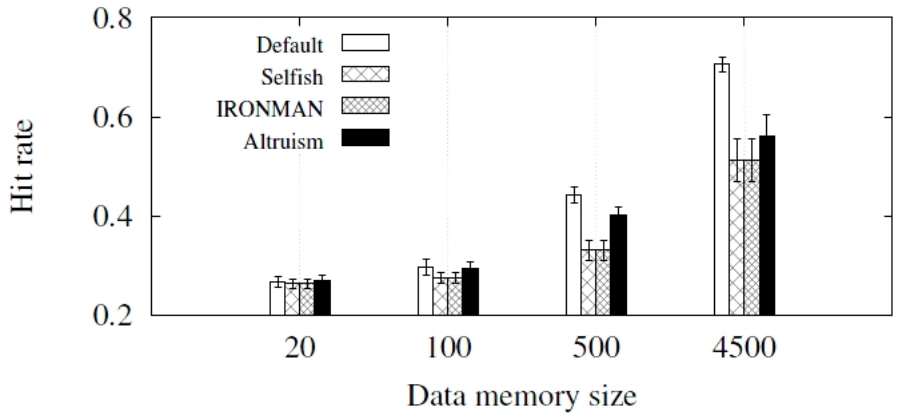
EVALUATION

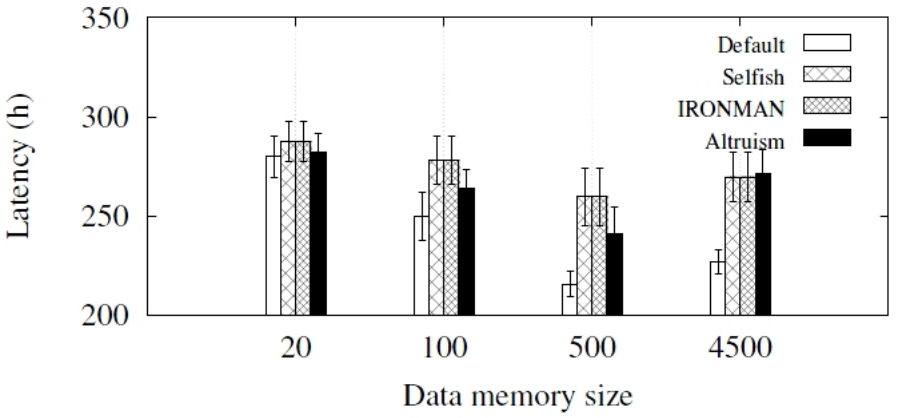
- MobEmu emulator for testing
- One real-life mobility trace (UPB 2012) and a synthetic mobility model (HCMM)
- Spray-and-Wait as the opportunistic routing algorithm (fixed number of copies)
- Variable size for *C*
- *O* and *I* with maximum 1000 entries
- Messages generated using a Zipf distribution with an exponent of 1 (30 messages per week-day)

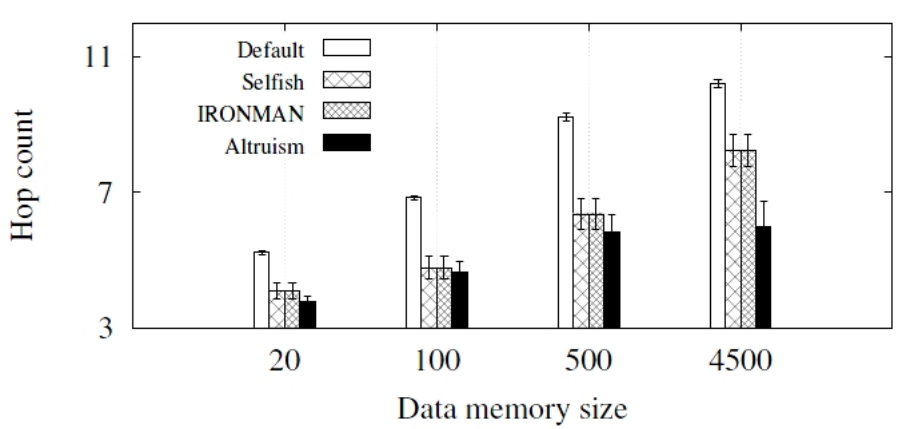
EVALUATION (2)

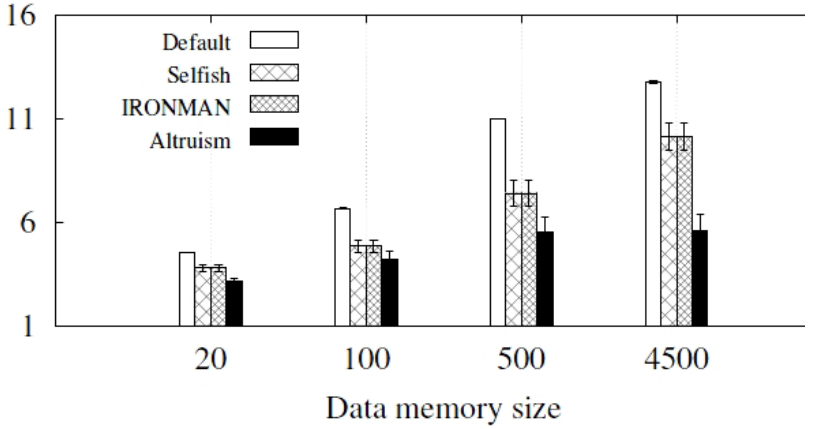
- Altruism between 0 and 1 (fuzzy), increased by 0.1 when the node is told it's selfish
- Inter and intra-community altruism values distributed normally with a mean of 0.4 for inter-community and 0.6 for intra-community
- **Two sets of tests:**
  - Analyze **hit rate**, **latency**, **delivery cost** and **hop count**
  - Analyze **community-biased detection accuracy** (percentage of nodes that end up with altruism 1)

EVALUATION (3)



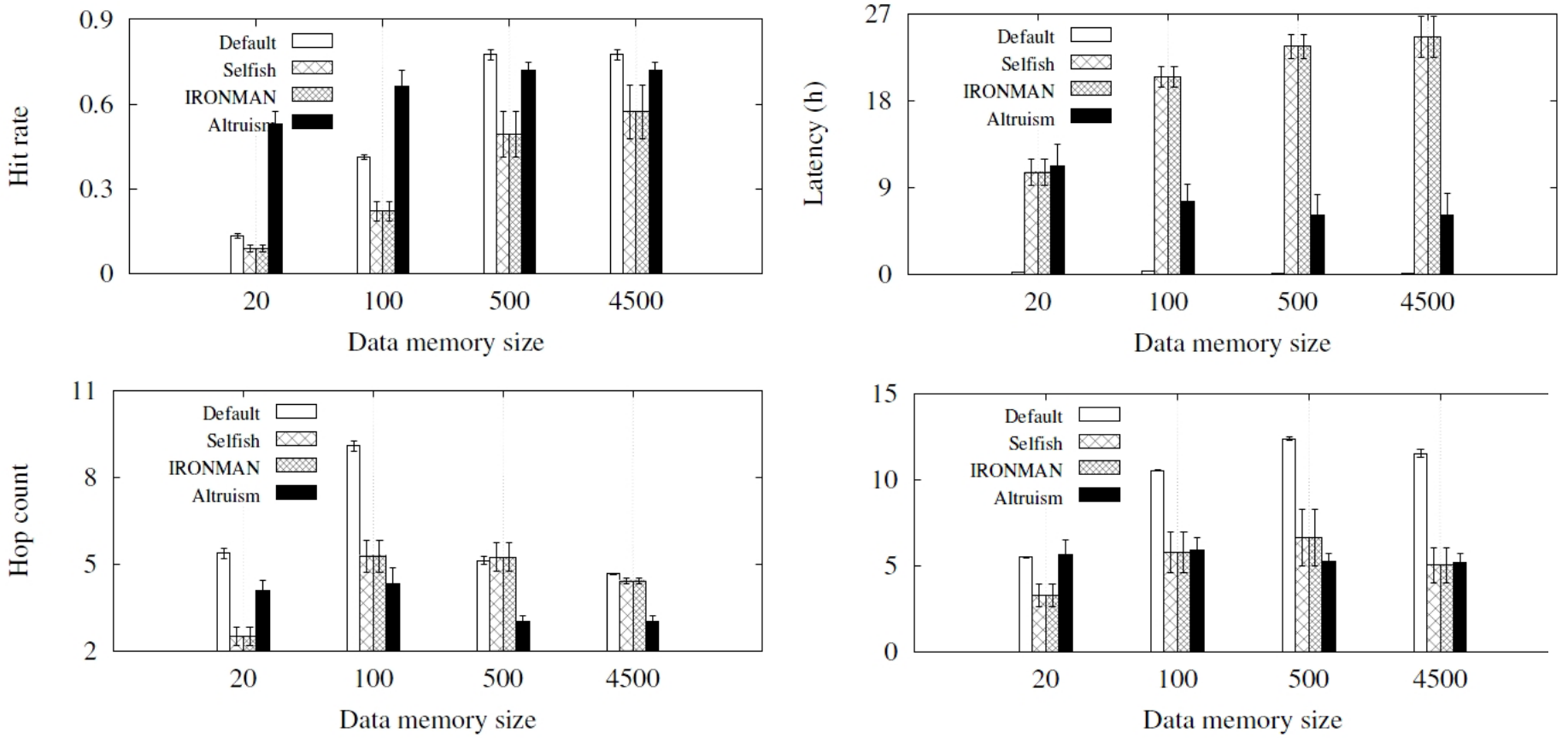






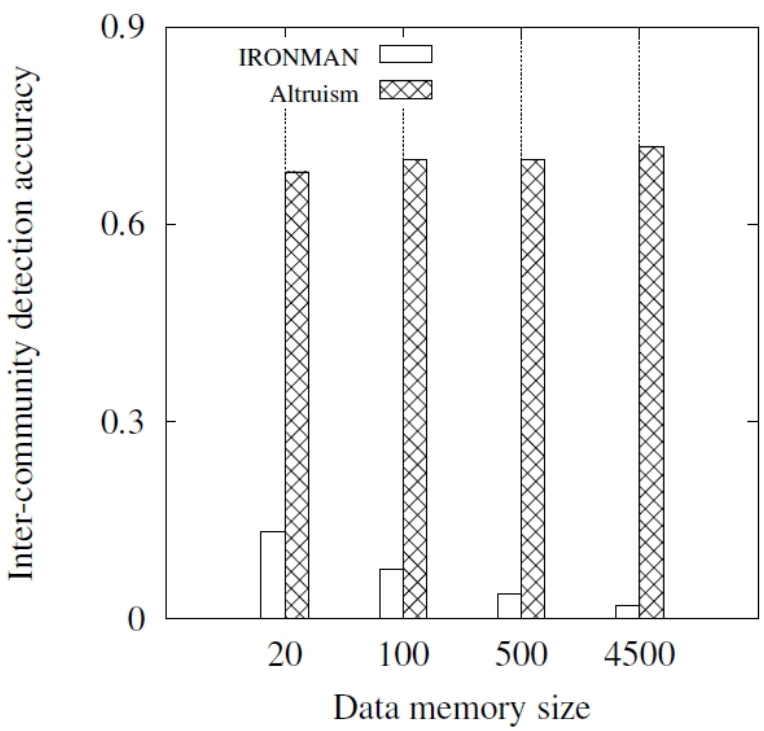
UPB 2012 Results

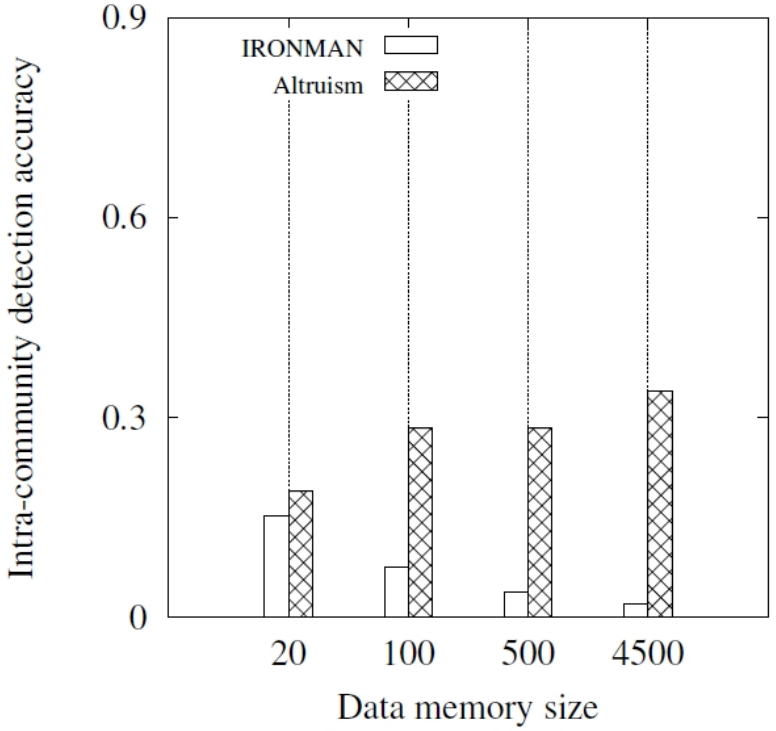
EVALUATION (4)



HCMM Results

EVALUATION (5)





Community-biased detection accuracy (UPB 2012)

CONCLUSIONS

- Novel social-based collaborative content and context-based selfish node detection algorithm with incentive mechanisms for Ons
- Gossiping & incentive mechanisms
- Tested on a social trace and a mobility model
- Outperforms IRONMAN in terms of hit rate and latency, and it fares better than the default case for hop count and delivery cost
  - Motive: messages are sent in a selective manner, only to nodes that have already successfully delivered messages of that type
- Better detection accuracy than IRONMAN