ADHOC-NOW 2013
The 12th International Conference on
Ad Hoc Networks and Wireless
July 8-10, 2013, Wrocław, Poland

Social-Based Routing with Congestion Avoidance in Opportunistic Networks

Alexandru Asandei, Ciprian Dobre, Matei Popovici

University POLITEHNICA of Bucharest, Romania

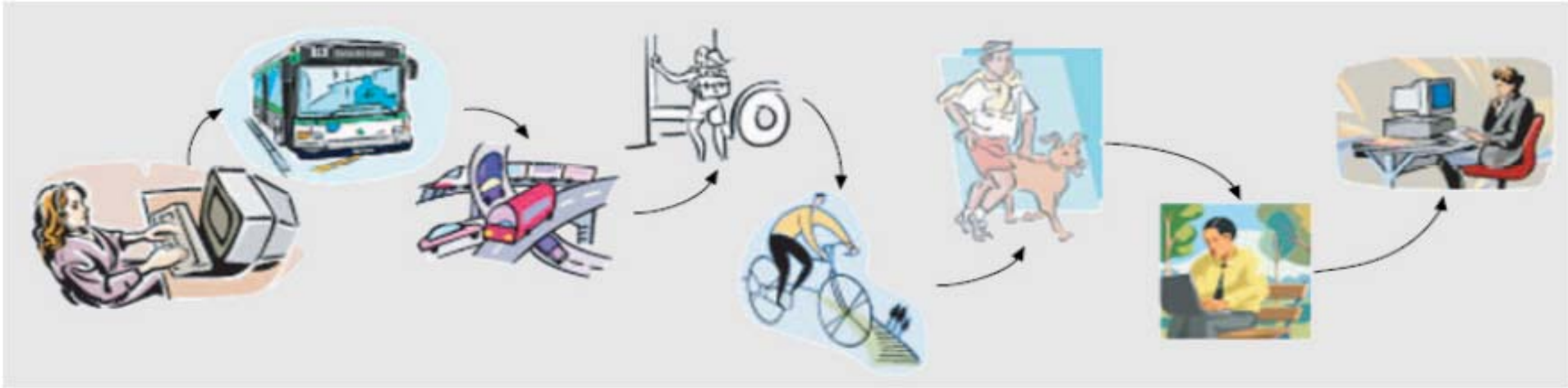
ciprian.dobre@cs.pub.ro

OUTLINE

- Scope and motivation
  - Social-aware routing algorithms
  - Congestions => problem
- Two approaches
  - *Outer* and *LowerEps* vs. BubbleRap
- Conclusions and future work

Opportunistic Networks

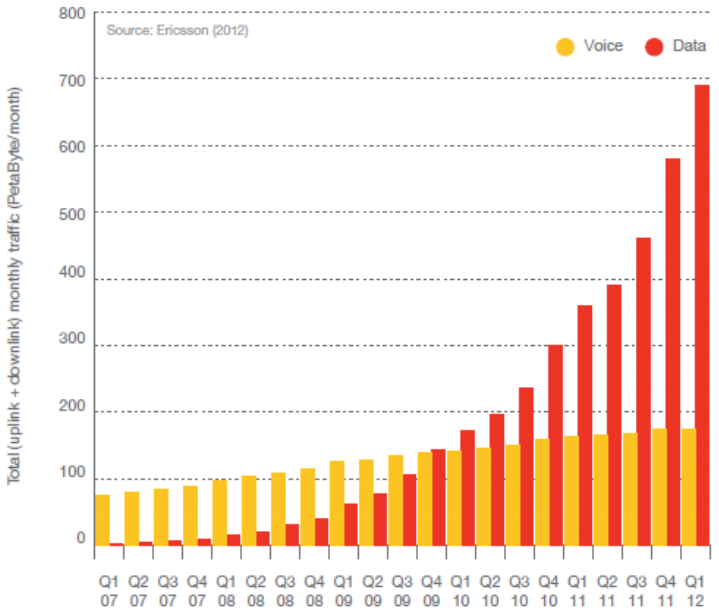
- Evolution of MANET
- Networks composed of mobile devices
- No assumption made on the existence of paths between nodes
- Store-carry-and-forward
- Connectivity opportunities when no direct access to the Internet is available



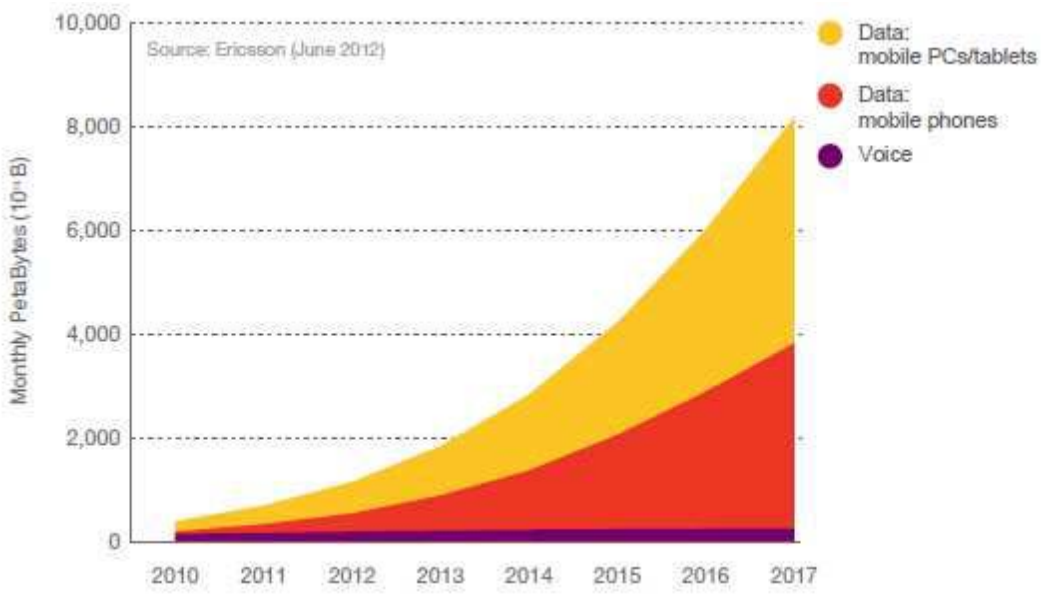
Opportunity in exploiting wide-availability of mobile devices

World Data Traffic: Status & Forecast

- The myth of exponential growth ?



Development 2007-2012

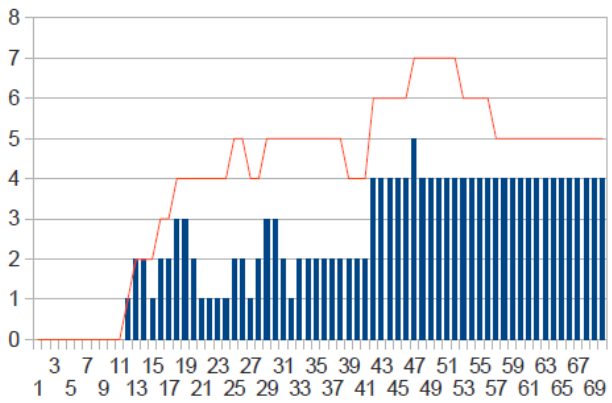


Predictions

By 2025 experts predict a 1000-fold increase of data traffic!

Applications of ONs

- Traditionally applications: data and computation offloading, distributed social networks, e-Post-it
- New applications of interest
  - Data sharing: tile sharing, smart taxi, data sharing in crowded envs., disaster management
  - Sensor sharing: GPS in the bus
  - Computation sharing: HYCCUPS
  - Network sharing: e.g., roaming net sharing



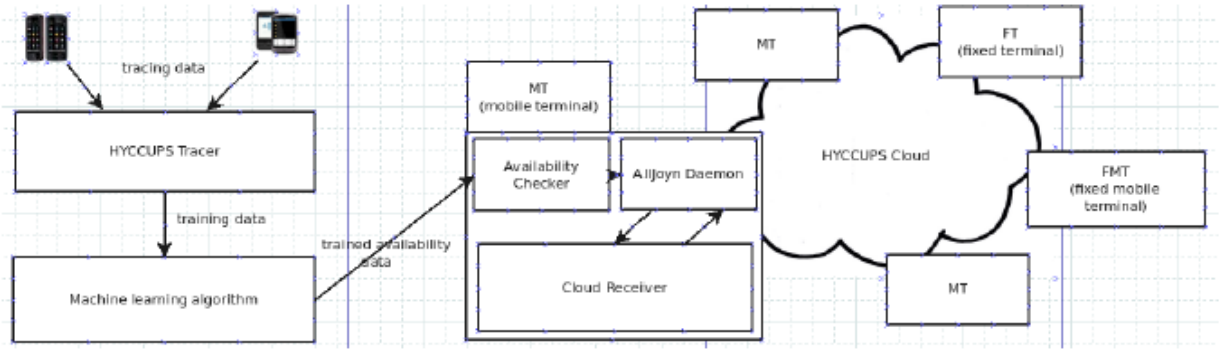


Fig. 3. HYCCUPS architecture

Challenges

- Lack of connectivity → lack of end-to-end paths
- How to select next hop?
- Congestion, traffic overhead → energy consumption
- Long and variable delay
- Asymmetric data rates
- Reliability
- Privacy and security
- Only use locally collected knowledge
- **Achieve real "mobile computing" without the need for a connected network**

The problem…

- Mobility-aware routing algorithms for ONs
  - e.g., PRoPHET
  - history-based: routing decisions based on the number and duration of node encounters
- Social-aware routing algorithms for ONs
  - e.g., Bubble Rap
  - popularity-based: routing decisions based on how people are organized into communities, according to places of living and work, common interests, leisure activities, etc.
- More interest lately on social-based routing algorithms
- Network congestions?
  - More popular ON members become flooded by message forwarding requests
- Can social-based ONs successfully carry messages destined for others, as envisioned by the opportunistic approach?

Bubble Rap

- Nodes are organized into communities
- Each node is assigned a local ranking and a global ranking
  - Measures the importance (centrality) of the node, either in some community, or in the entire ON
- Routing:
  - Messages are forwarded to nodes with higher global ranking until a node from the destination's local community is encountered
  - Next, the same forwarding process occurs inside the local community, based on the local ranking instead of the global one

```
1 begin Classic()
2   foreach EncounteredNode_i do
3     BubbleProcedure()
4 end
5 Obs: for each generated message, InnerTokens = OuterTokens = MAX_TOKENS/2
```

Bubble Rap

- Whenever a message is generated, two values InnerTokens and OuterTokens expressing the message copy limit in the local and global communities, are associated with it.

```
1  begin BubbleProcedure()
2      if (LabelOf(currentNode) == LabelOf(destination)) then
3          if (LabelOf(EncounteredNode_i) == LabelOf(destination))
4              and (LocalRankOf(EncounteredNode_i) > LocalRankOf(currentNode))
5              and (message.InnerTokens > 0)
6          then
7              message.InnerTokens--
8              EncounteredNode_i.addMessageToBuffer(message)
9      else
10         if ((LabelOf(EncounteredNode_i) == LabelOf(destination))
11             or (GlobalRankOf(EncounteredNode_i) > GlobalRankOf(currentNode)))
12             and (message.OuterTokens > 0)
13         then
14             message.OuterTokens--
15             EncounteredNode_i.addMessageToBuffer(message)
16 end
```

- Surpassingly, nobody evaluated the probability of node congestion…

The congestion problem

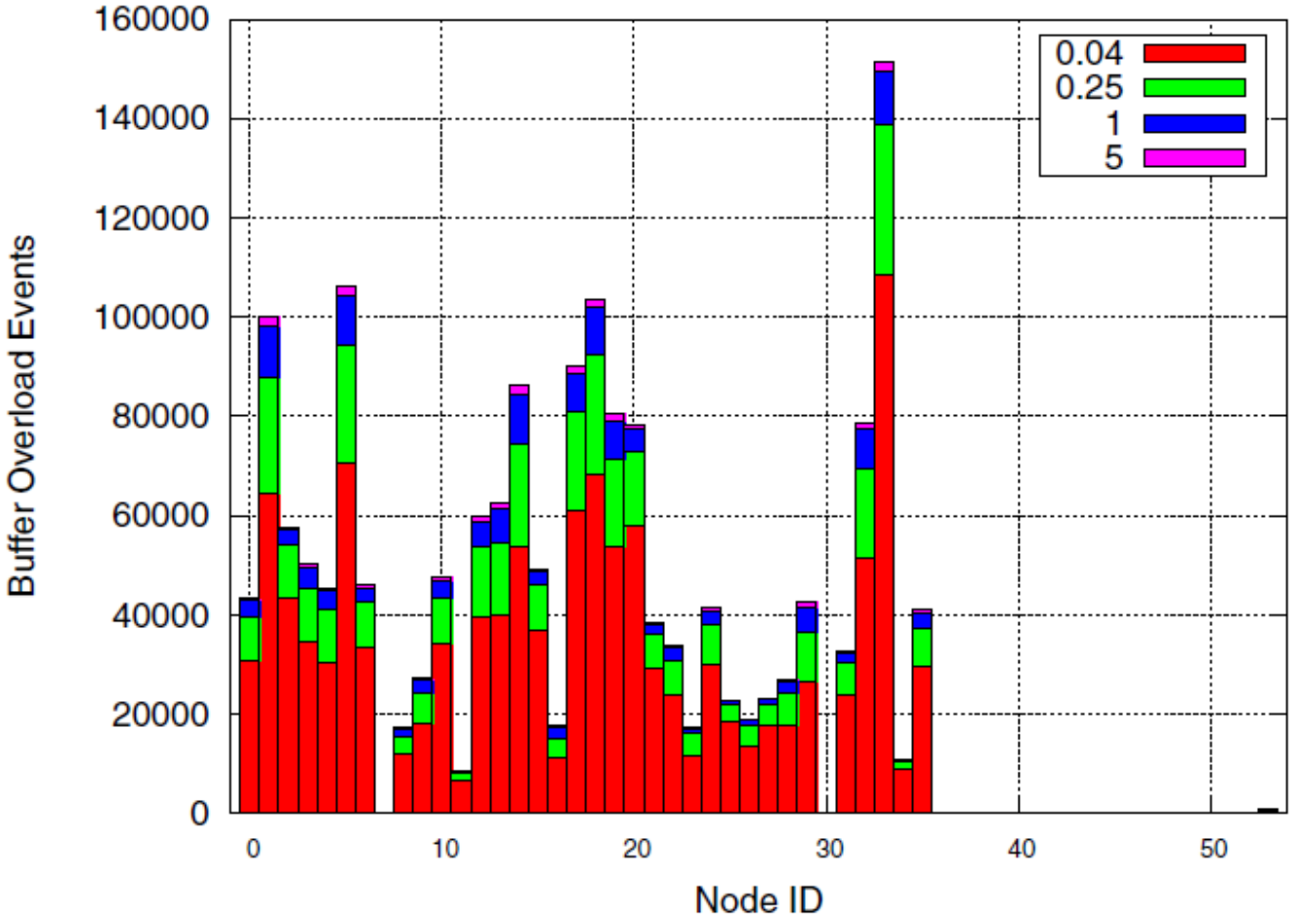
- Evaluated the Bubble Rap algorithm on various interaction densities
- Low density: any two wireless nodes (i.e., cars in traffic, large urban environments) meet with a low probability.
  - *St.Andrews*: trace collected in the city of St. Andrews, UK, with 27 participants, 11 weeks. It reflects mobility in an urban environment.
- Medium density: small communities (i.e. suburbs, small towns, university campuses), where node encounters have a higher probability than in case of *low* density scenarios.
  - *Content :* experiment performed in Cambridge, UK, with 54 participants over 8 weeks. It reflects mobility in a academic environment.
- High density: node encounters are frequent (i.e. conferences, employees within a single company).
  - *Infocom:* trace collected at the IEEE Infocom Conference, involving 41 participants over 3 days.

Experimental setting

- Each node can generate a message for some other node(s)
  - Out of the total of *N* nodes in the network, *N/2* nodes are generators of messages.
- The number of messages each generator sends are distributed according to the normal distribution
  - We used a *community bias* distribution for choosing destinations sent messages
  - This corresponds to the distribution of status messages in a social network
- Messages have an average size of 10KB
  - Each node has a buffer size of approximately 20MB, meaning can store up to 2048 messages
  - The choice of the buffer size is motivated by the current memory limit of most applications on Android devices, which ranges from 16 to 24 MB, on newer generation phones…

Results on congestion…

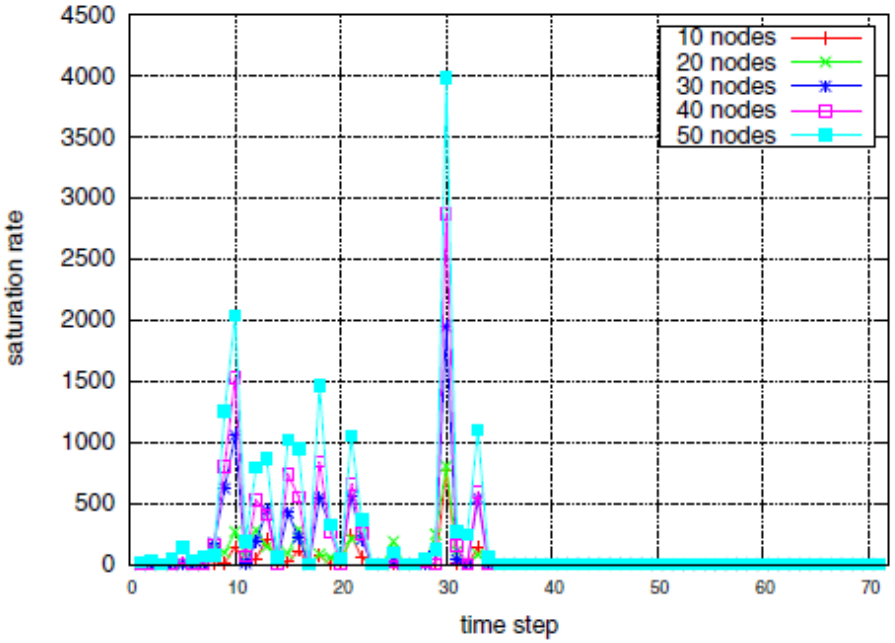
- B*uffer overflow* event: experienced when the buffer capacity is exceeded - for example, when a node receives messages at a higher rate than it is able to forward, thus causing message buffers to be filled



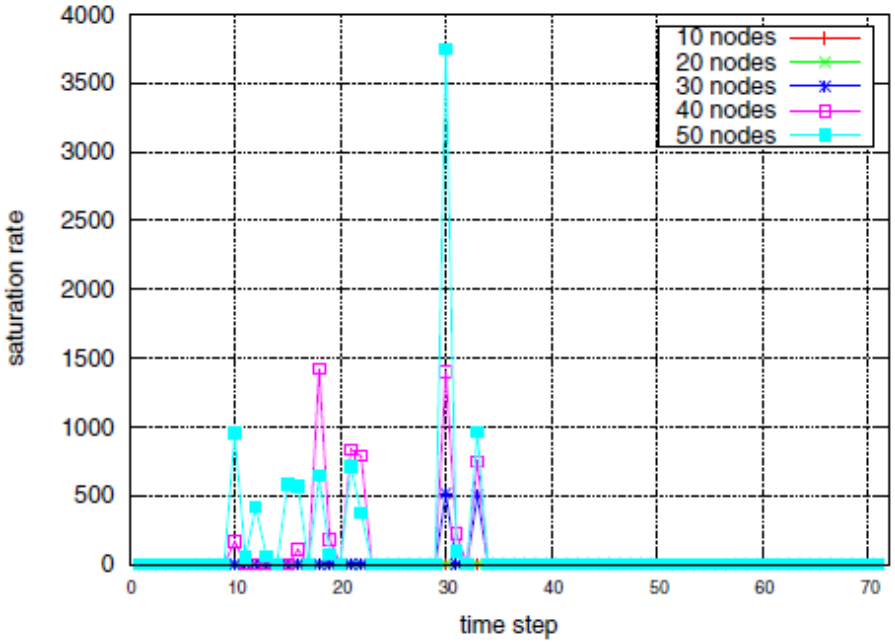
Out-of-memory events, *Content*, 2 message copies

Saturation rate

- Next, we were interested to see how pervasive these events are...
- We monitored the *saturation rate* for each node: the number of saturation events a node experiences, over a fixed period of time.



(a) Initial buffer size.



(b) 20 times increased buffer.

Saturaton rate, *Content*, 2 message copies, 0.25 message generation period

Node saturation is a *general problem* intrinsic to social-based routing algorithms, and does not depend on specific features of the ON at hand!

The *Outer* algorithms

- Outer: to avoid node saturation, after a message copy is forwarded inside the destination's community, all other copies are used for encounters with nodes from other communities…
  - Once a message has reached a node from it's destination community, it has a high delivery probability
- This restriction is fit for relieving message flooding inside local communities…

```
1  begin Outer()
2    foreach EncounteredNode_i do
3      BubbleProcedure()
4      if (message.wasNotForwardedTo(EncounteredNode_i))
5        and (LabelOf(currentNode) !=
6          LabelOf(destination))
7        and (LabelOf(currentNode) !=
8          LabelOf(EncounteredNode_i))
9        and (message.OuterTokens > 0)
10     then
11       message.OuterTokens--;
12       EncounteredNode_i.addMessageToBuffer(message)
13 end
14 Obs: for each generated message, InnerTokens=1 and OuterTokens= MAX_TOKENS-1
```

LowerEps

- LowerEps: exploits randomness
- In Bubble Rap, messages having as destination nodes with low global ranking have to traverse nodes with higher ranking in the local community first, before being finally delivered.
- Approach: For each node encounter there is a probability that a message will be forwarded to the encountered node, regardless of its rank, and a 1− probability that the algorithm will behave precisely as *Classic*.

```
1  begin LowerEps()
2    BubbleProcedure()
3    if  (message.wasNotForwardedTo(EncounteredNode_i))
4      and (random < 0.1)
5      and (message.Tokens > 0)
6    then
7      message.Tokens--;
8      EncounteredNode_i.addMessageToBuffer(message)
9  end
10 Obs: for each generated message, InnerTokens = OuterTokens = MAX_TOKENS/2
```

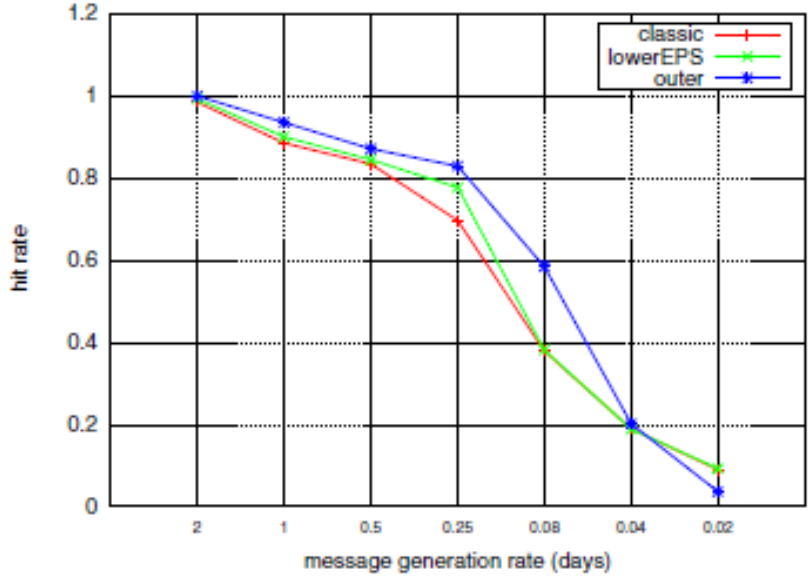
Performance results

- No widely accepted metric for measuring the susceptibility to congestions in ONs...
- We propose *load balancing* as relevant metric:
  - An ON node *i* is less prone to congestion if the number of encounters of node *i* is proportionate to the number of messages sent to node *i*.
- The *load balancing factor* (LBF) for node *i* (denoted LBF(i)), is the difference between the normalized sighting values for node *i* and those for the normalized number of received messages by node *i*:

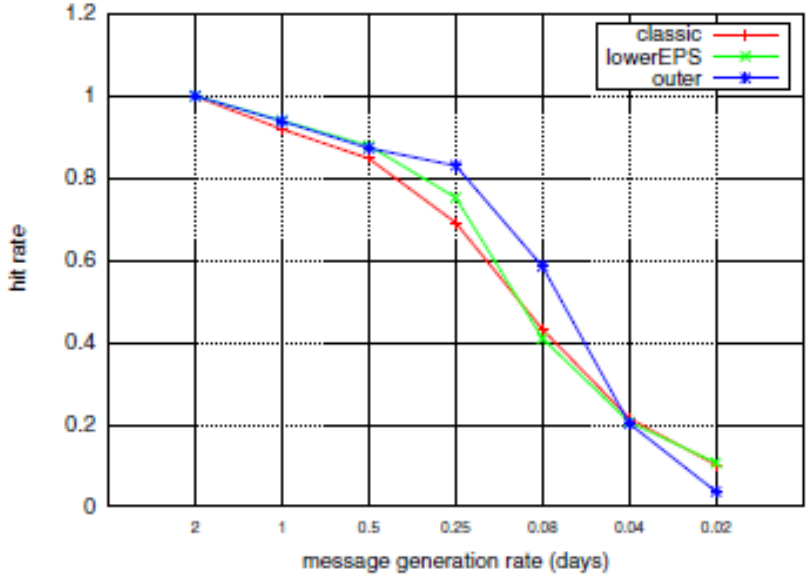
$$\mathrm{LBF(i)} = \frac{\mathrm{sight}(i)}{\max\limits_{i \in \mathrm{nodes(ON)}} \mathrm{sight}(i)} - \frac{\mathrm{msg}(i)}{\max\limits_{i \in \mathrm{nodes(ON)}} \mathrm{msg}(i)}$$

- We use the average value of LBF(i) computed for all nodes *i* in the ON, to measure the exposure to congestion for each algorithm, on a given trace:
  - We denote this value as *ALBF*.

Results

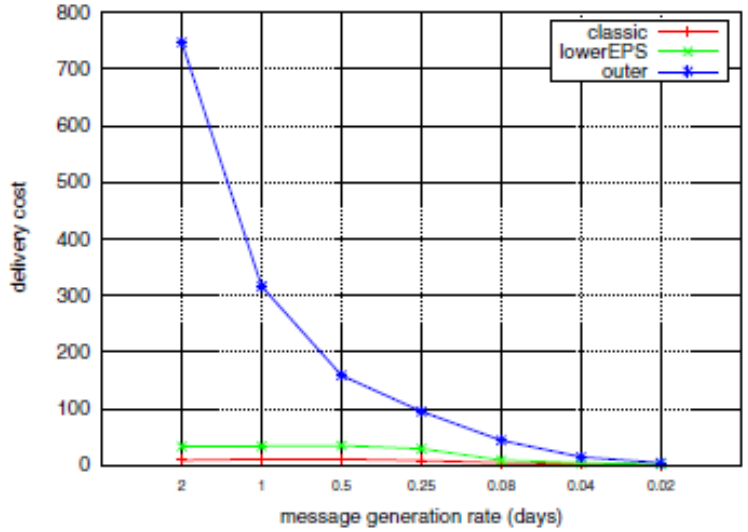


(a) 2 message copies.

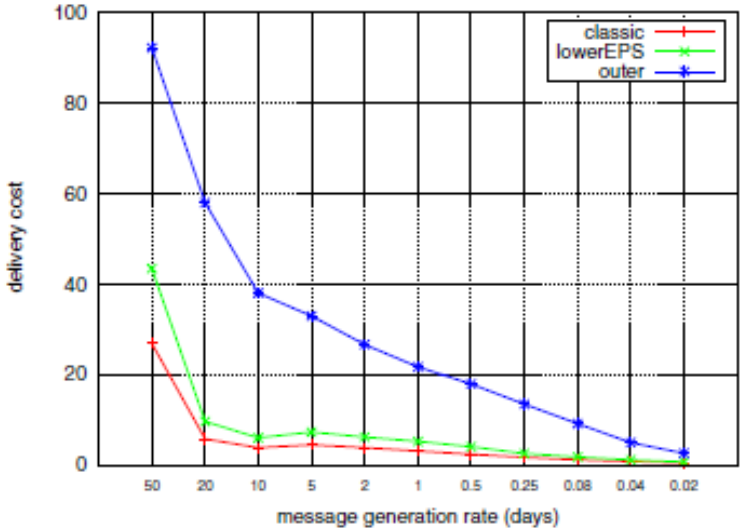


(b) 8 message copies.

Hit rate, *Infocom*



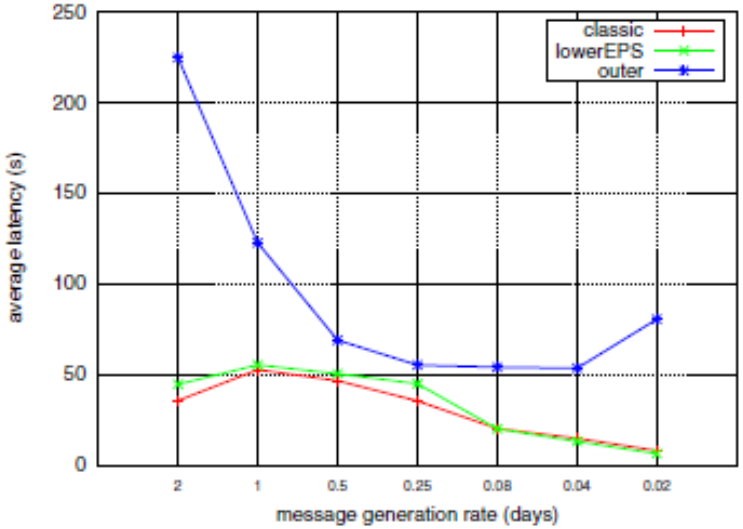
(a) *Infocom.*



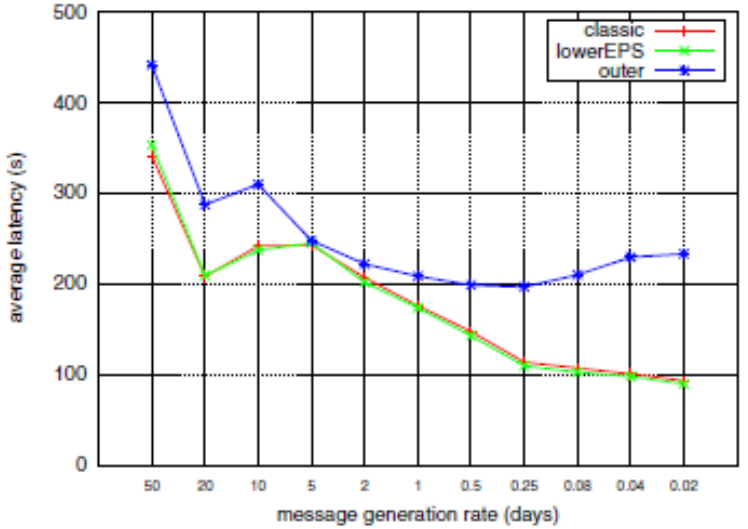
(b) *Content.*

Delivery cost, 2 message copies

Results

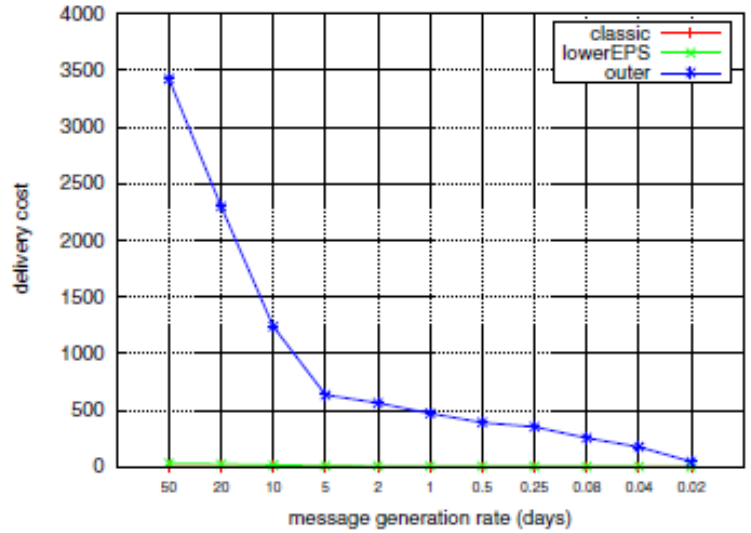


(a) *Infocom.*

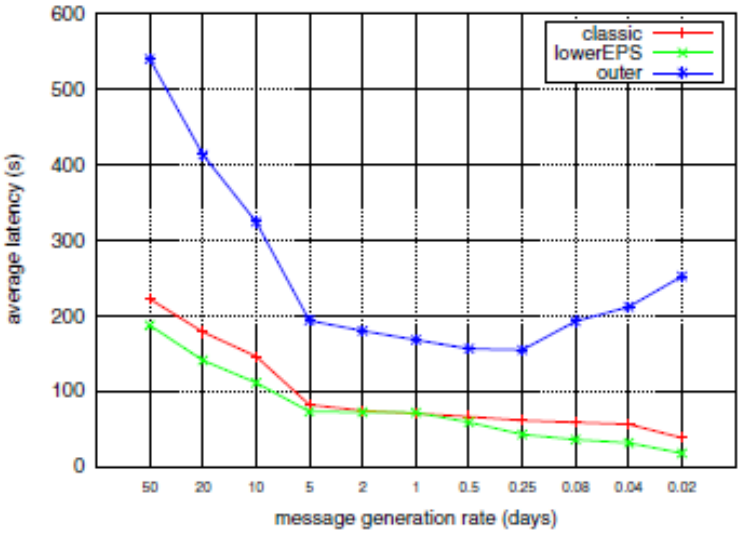


(b) *Content.*

Average latency, 2 message copies



(a) Delivery cost.

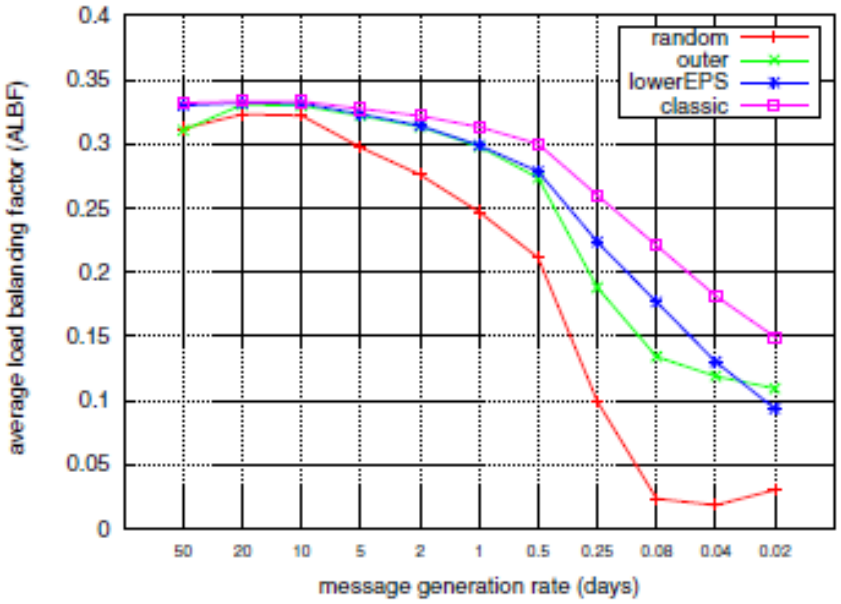


(b) Average latency.

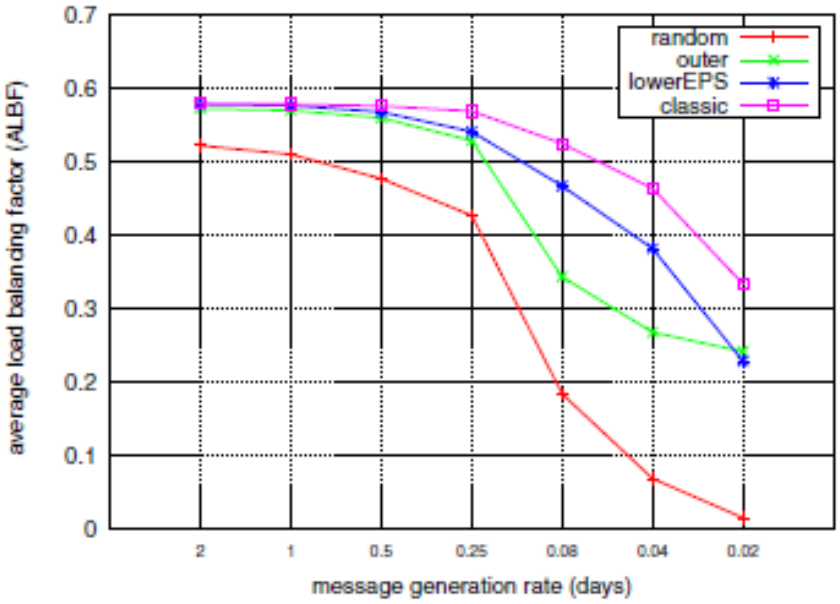
St. Andrews, 2 message copies

Results

- Outer achieves the best load balancing, being closest to the random routing algorithm
- Classic has the worst behaviour, which shows that it is unfit to provide good load balancing.



(a) *Content.*



(b) *Infocom.*

ALBF, 2 message copies

Conclusions

- Demonstrated that congestions are likely to appear in social-based routing algorithms due to buffer overflows
- We identified two solutions to the congestion problem.
  - Results suggest that global rankings are not always a suitable criterion for forwarding messages between nodes of different communities
- The observations were drawn from experiments performed on a variety of traces, with different number of nodes, taken in various environments and having a broad range of sighting distributions.

Q&A

Thank you!

Ciprian Dobre

ciprian.dobre@cs.pub.ro