Publish/Subscribe Architecture for Mobile Ad hoc Networks

Authors: C. Rezend, B. Rocha and A. Loureiro

Presented by: Luis Oleas-Chávez

Agenda

- Publish/Subscribe – Concepts & Applications
  - Dissemination tree
  - Beacon exchanges
  - Gossip algorithm
- Related Work
- Architecture
  - Publication Buffer
  - Subscription Table
- Experiments and Results
  - Tuning
  - Performance Analysis
- Conclusions

Publish/Subscribe

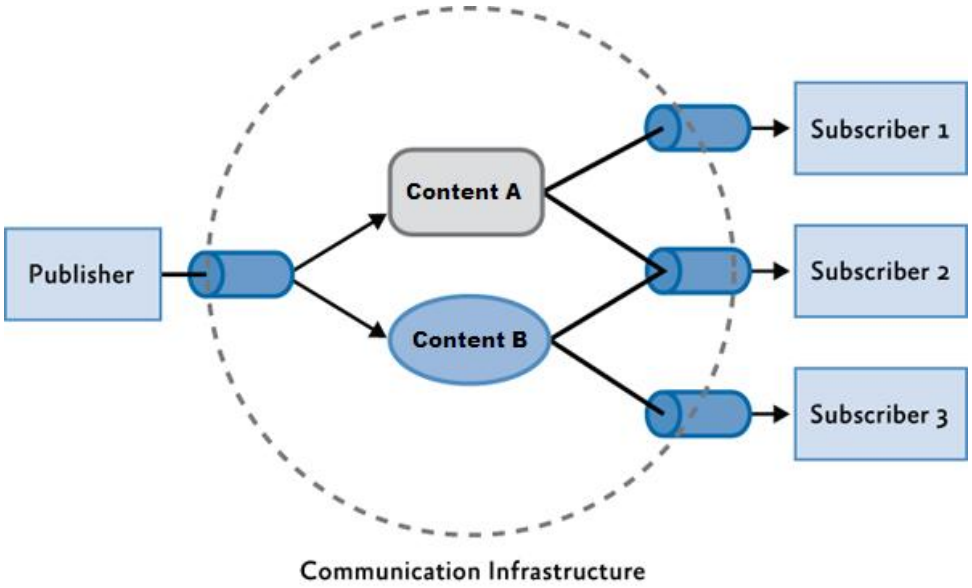
- Publish/Subscribe is a one-to-many communication paradigm
- *Publisher*: Publishes structured events to an event service.
- *Subscribers*: Register their interest in an event, or pattern of events, and are asynchronously notified of events generated by publishers.
- *Event Notification Service*: Provides storage and management for subscriptions and efficient delivery of events.

Variants Publish/Subscribe

- *Topic-Based*: It is based in the notion of groups (many-to-many relationship).
- *Content-Based*: Subscriptions are related to specific information by specifying filters (many-to-one relationship).
- *Type-Based*: Events are filtered by their type. It provides guarantees such as type safety and encapsulation.

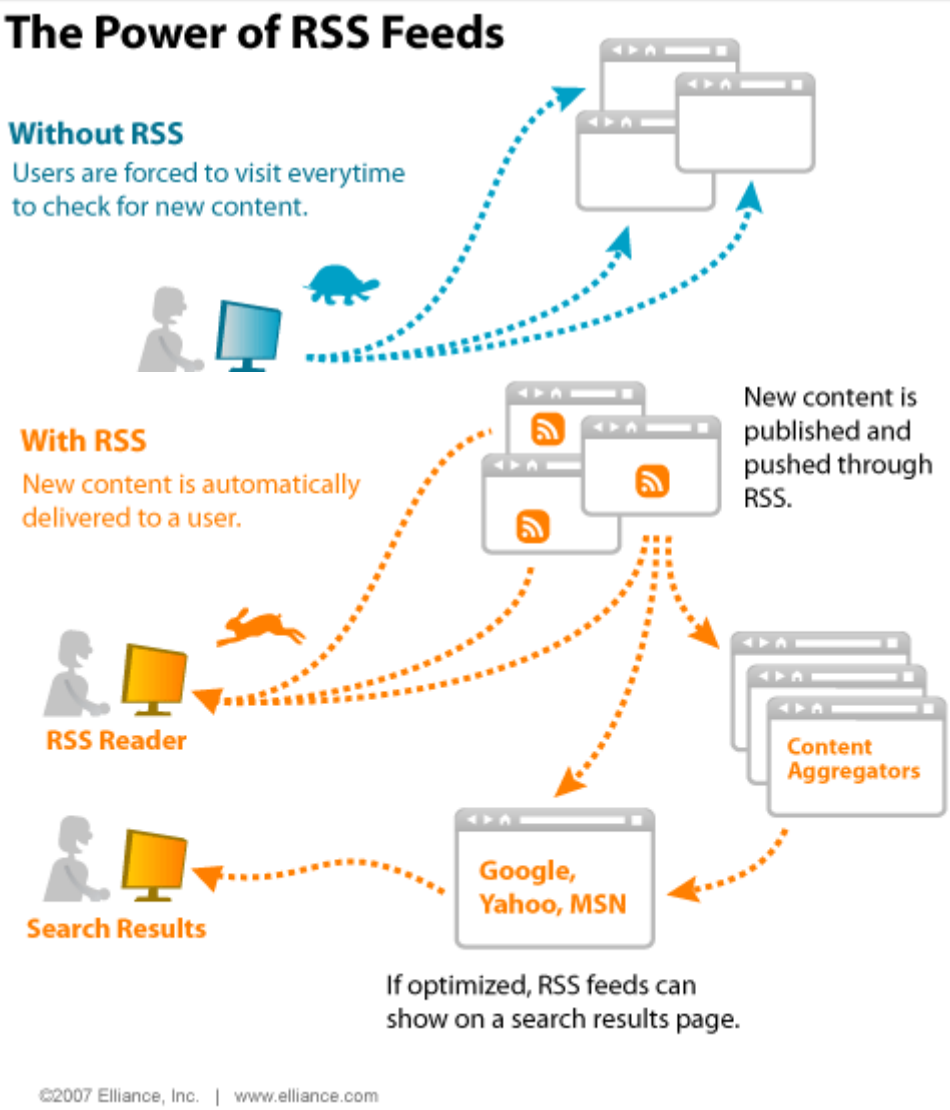
Publish/Subscribe (Cont'd)

- Key properties:
  Full decoupling of the communication entities
  - Space
  - Time
  - Synchronization



Applications

- Financial information systems
- Live feeds of real-time data (including RSS feeds)
- Cooperative working environments with many participants with shared interests in events
- Ubiquitous computing
- Monitoring applications

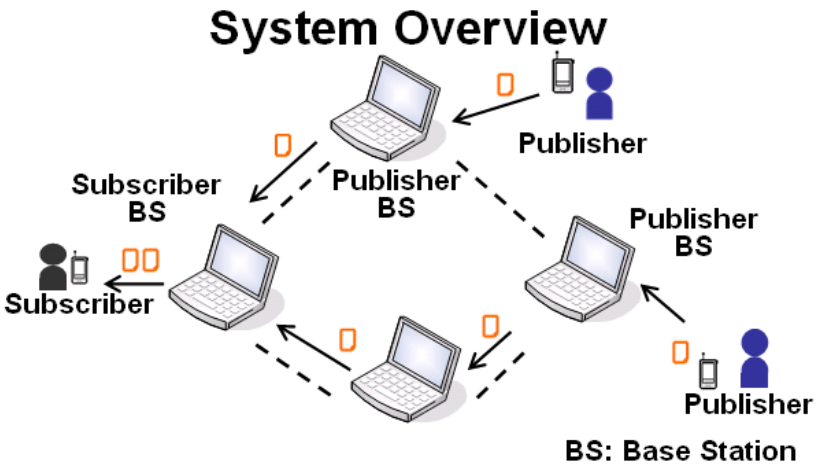


Mobile Ad hoc Network

- Publish/Subscribe architecture has been deeply studied and applied in wired networks.
- The incursion of a mobile Ad hoc Network (MANET) presents challenges which can be minimized using the nodes movement to disseminate publications to the whole network with a reduced number of transmissions.
- These nodes will be responsible for disseminating locally received publications to different areas of the network after they move

Related work

- Previous proposals that make reference to wireless networks, in fact were adaptations of existing wired networks architectures.
- Propagation techniques discarded:
  - Dissemination trees (Huang)
  - Beacons exchanges (Baehni)
  - Gossip-based algorithm



Architecture

- The purpose of this architecture is the use of mobility to notify interested subscribers of messages sent by publishers in a mobile ad hoc network using a minimum amount of broadcasts.
- The developed PSAMANET is based on signatures which describe the interested structured content and publications in a way that its content is readable.

Architecture (Cont'd)

- The number of publications are predefined by a value R; thus, subscriptions are defined in a range of interested values $[v_{min}, v_{max}]$.
- Subscriptions are sent to neighbors when:
  - A node has been created
  - A node stops after it had moved
- Publications are broadcast when:
  - A node has been created
  - A node stops
  - A node has a publication which matches a new incoming subscription
  - A node receives a publication which matches any stored and still valid subscription

Publication Buffer

- A PB is a FIFO queue that stores the incoming publications in every node.
- The PB stores the PB_{size} more recent received publications. PB_{size} is given by:
  - Increases the number of reached nodes
  - Impacts the number of messages sent
- PB considers an upper bound to the number of publications forwarded each time PB_{max} in the case more than PB_{max} publications have to be sent.

Subscription Table

- A node must hold no only the FIFO queue but also a subscription table which stores the nodes subscriptions.
- An important observation is that subscriptions do not stay for a long time in the ST. Then, two types of subscriptions are considered:
  - Own (ST is cleaned when it moves)
  - Foreign (Subscription valid for an amount of time ST_{ttl})
- Every time the ST is used, it first checks for expired subscriptions and remove them.
- Duplicate received publications are avoid with the help of S_{prob}. It reduced the amount of bandwidth and energy wasted

Experiments

- In the experiment, the density of the network is such that a node has an average of four neighbors. It was implemented on NS-2 simulator
- The movement used was Random Way Point.

Parameter	Value
Radio Propagation Model	FreeSpace
RXThresh_	7.69113e-08 (50m)
Routing	DumbAgent
Number of Nodes	200
Area	600m × 606m
Simulation Time	3600s

Parameter	Value
Speed	[1.5,4.0]m/s (normally)
Pause	[0, 600]s (uniformly)

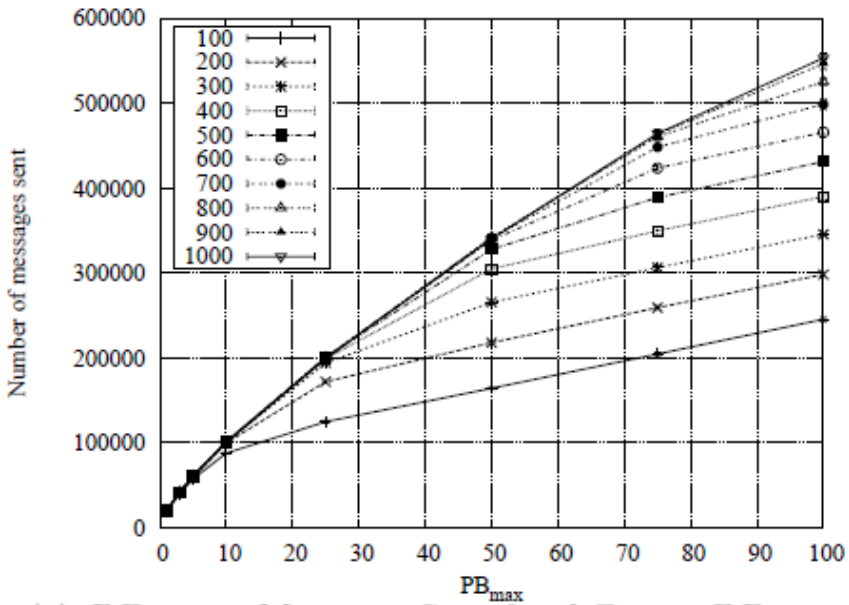
Experiments (cont'd)

- Each node subscribe to two different ranges which are generated randomly.
  - S_{range}: Set to a value which makes a publication to be of the interest to 20% of nodes (R/400)
  - v_{min}: Positive number smaller or equal to R
  - v_{max}: Minimum value plus a constant range S_{range}
  - Total number of publications: 1000
- Experiment phases:
  - Tuning (PB_{size}, PB_{max}, S_{ttl} and S_{prob})
  - Performance analysis

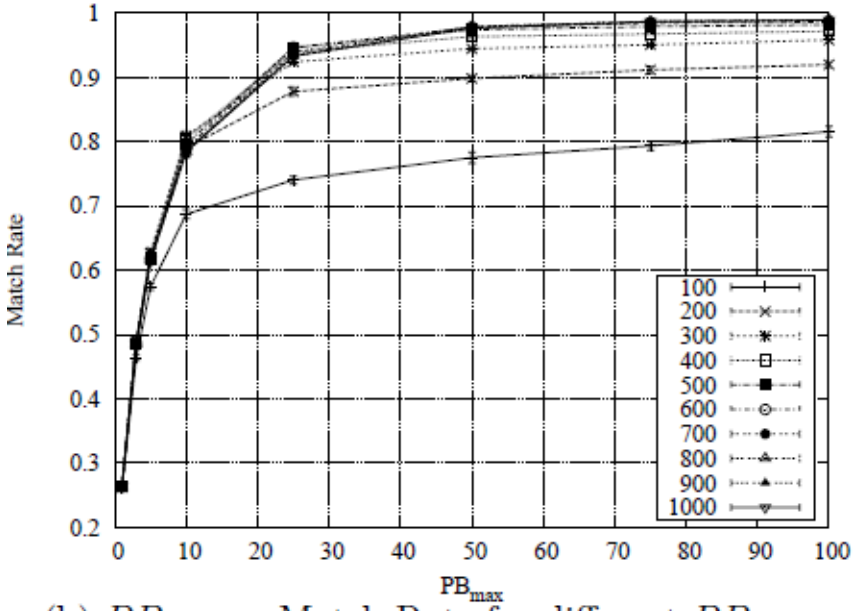
Tuning

- Initial values
  - PB_{size}: influences how long a publication lasts in the network
  - PB_{max}: how many times a publication is sent
  - S_{ttl}: 300
  - S_{prob}: 0.4
- There are 1000 publications during each simulation
  - PB_{size}: 1000 (unlimited buffer)

Tuning (cont'd)



(a) PB_{max} × Messages Sent for different PB_{size}

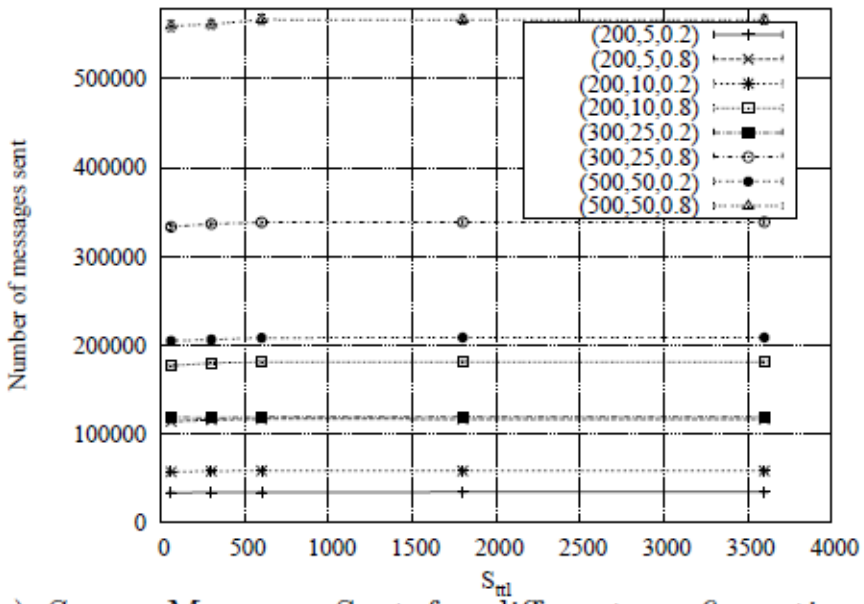


(b) PB_{max} × Match Rate for different PB_{size}

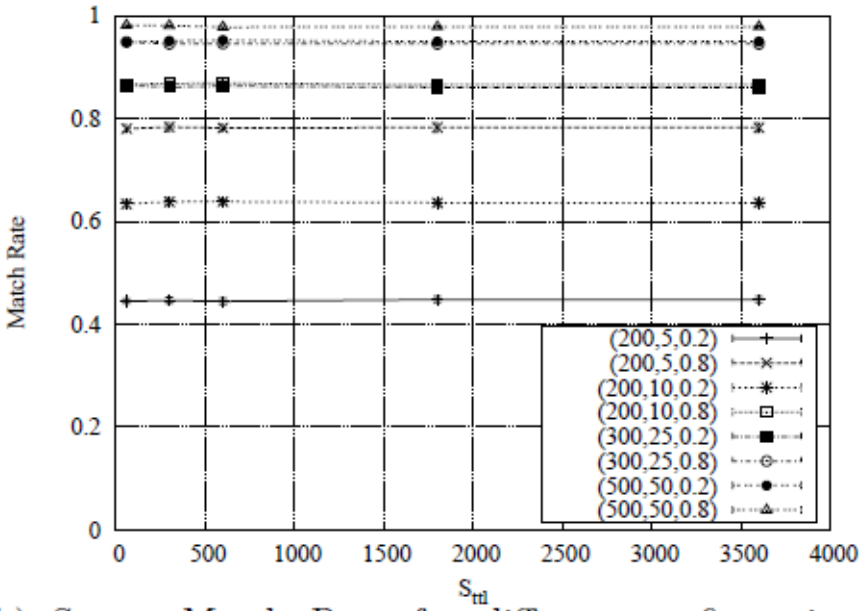
Tuning (cont'd)

- Evaluate configurations with a small number of messages and others with high match rates.
- Subscriptions parameters with four PB configurations with (PB_{max}, PB_{size}).
  - (5, 200), (10,200), (25, 300), and (50, 500)
  - Foreign subscriptions last for 60 sec.
- The first observation was that the S_{ttl} does not have a great impact on the number of messages sent nor on the match rate.

Tuning (cont'd) S_{ttl}

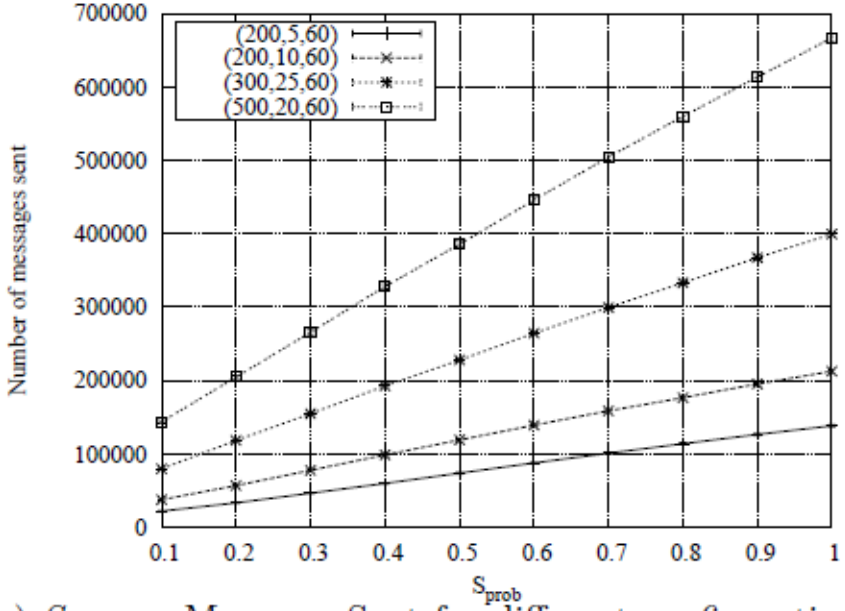


(a) S_{ttl} × Messages Sent for different configurations $(PB_{size}, PB_{max}, S_{prob})$

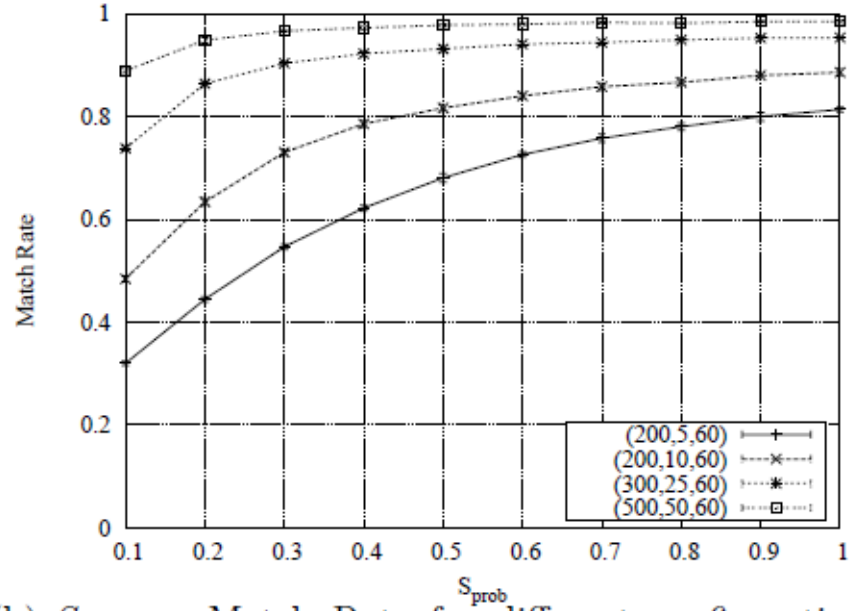


(b) S_{ttl} × Match Rate for different configurations $(PB_{size}, PB_{max}, S_{prob})$

Tuning (cont'd) S_{prob}



(a) S_{prob} × Messages Sent for different configurations $(PB_{size}, PB_{max}, S_{ttl})$



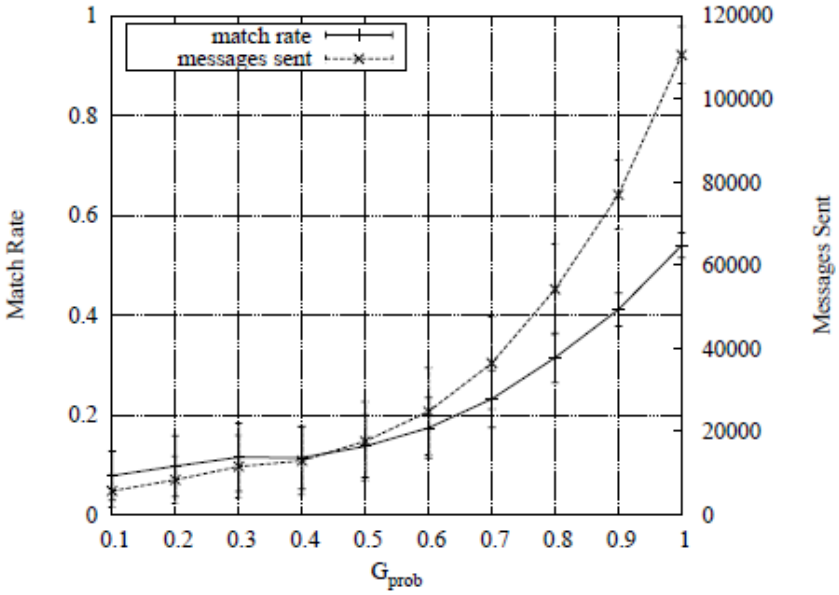
(b) S_{prob} × Match Rate for different configurations $(PB_{size}, PB_{max}, S_{ttl})$

Performance Analysis PSAMANET vs. Gossip-based

- This experiment was tested against a gossip-based routing algorithm for ad hoc networks, where every subscriber stores its subscriptions.
- The publisher broadcasts its publications and whenever a node receives any publication it has a probability of G_{prob} to forward it.

Performance Analysis (cont'd) PSAMANET vs. Gossip-based

Configuration	Messages Sent	Match Rate
I	37,869 ± 234	0.4896 ± 0.0036
II	99,375 ± 632	0.7892 ± 0.0032
III	195,729 ± 991	0.8774 ± 0.0033
IV	119,013 ± 931	0.8659 ± 0.0035
V	193,974 ± 1,181	0.9242 ± 0.0024
VI	207,611 ± 1,508	0.9539 ± 0.0020
VII	330,740 ± 2,137	0.9741 ± 0.0017



Conclusions

- This work presents a solution to the problem of developing a PSAMANET which properly adapts to the highly dynamic topology of such networks using nodes' movement to disseminate publications.
- It considers a totally asynchronous communication in such a way that end-to-end delays of minutes are not a problem and moving nodes can connect distant regions of the network with fewer transmissions.
- The results showed a much better performance than the Gossip algorithm. For instance, it was possible to reach a 46% higher match rate sending 10% less messages.

Future Work

- To analyze of PSAMANET behavior as a complex network to observe properties like degree distribution and diameter.
- To study the Publisher/Subscriber paradigm in MANETs for real-time applications and also try to use mobility without incurring in a high delay.