An Algorithm for Dissemination and Retrieval of Information in Wireless Ad Hoc Networks

Euro-par 2007

Hugo Miranda[1] Simone Leggio[2] Luís Rodrigues[1] Kimmo Raatikainen[2]

[1] University of Lisbon and LaSIGE - Portugal

[2] University of Helsinki - Finland

August 29th, 2007

Motivation

The problem

- To make data items available to all nodes in a MANET
  - posts to a white board
  - SIP/SLP records
  - data collected in field surveys

Knowing that

- Nodes may fail⇒Data should be replicated
- Storage space is limited⇒It is not possible/desirable to replicate everything at all nodes
- Nodes have limited batteries⇒Use as few transmissions as possible

Related Work

SAF, DAFN, DCG On each round, nodes negotiate with their neighbours the items to store.

- Number and location of the replicas depends of an estimated access pattern
- No guarantee of geographical distribution

Data centric storage Items are stored at the nodes closer to a location extracted from an hash to the item (GHTs)

- Nodes must be aware of their location
- Some versions, geographically distribute the replicas

Data Distribution Framework

Assumptions

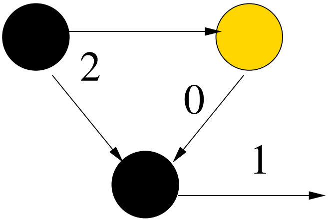
- Nodes are not aware of their location
  - I.e. do not have a GPS device
- Nodes cannot anticipate the data they will require

Goals

- To place a replica close to every node, to:
  - Reduce access latency
  - Save messages on data retrieval
  - Increase resilience to localised failures (e.g. interference)

The Dissemination Algorithm
The brief version

- Data is broadcasted with a counter field (TFS)
- TFS counts the distance from the transmitting node to the closest copy
    - Before forwarding the message, nodes compute the local minimum of TFS from the redundant transmissions
- Retransmissions
    - Nodes storing the data set TFS to zero
    - Nodes not storing the data set TFS to *minimum + 1*



The Dissemination Algorithm

Some Details

- Who stores the data?
    - The nodes whose local minimum exceeds a threshold (DbC)

stored $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ stored

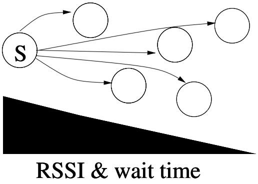
$$\Downarrow \quad <0,\text{data}> \quad <1,\text{data}> \quad <2,\text{data}> \quad \Downarrow \quad <0,\text{data}>$$

$$\bigcirc \quad \longrightarrow \quad \bigcirc \quad \longrightarrow \quad \bigcirc \quad \longrightarrow \quad \bigcirc \quad \longrightarrow \quad \ldots$$

- Who forwards the message?
    1. Any node that stores the data
    2. Some of the remaining (Pampa)
- Wait time is dictated by Pampa

Pampa

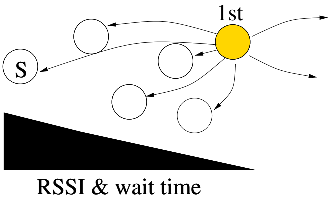
- An all-purpose broadcast algorithm
  - Selects the nodes required to retransmit a broadcast message
    - Reduces redundancy
    - Saves bandwidth and power
  - Wait time is proportional to RSSI
    - Nodes more distant from the source wait less
  - Nodes do not retransmit if they hear a predefined number of retransmission



Pampa *stretches* the hop length

Pampa

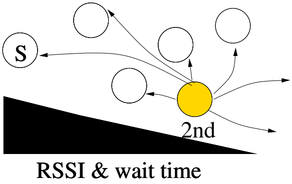
- An all-purpose broadcast algorithm
  - Selects the nodes required to retransmit a broadcast message
    - Reduces redundancy
    - Saves bandwidth and power
  - Wait time is proportional to RSSI
    - Nodes more distant from the source wait less
  - Nodes do not retransmit if they hear a predefined number of retransmission



Pampa *stretches* the hop length

Pampa

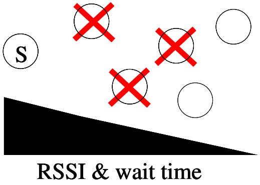
- An all-purpose broadcast algorithm
  - Selects the nodes required to retransmit a broadcast message
    - Reduces redundancy
    - Saves bandwidth and power
  - Wait time is proportional to RSSI
    - Nodes more distant from the source wait less
  - Nodes do not retransmit if they hear a predefined number of retransmission



Pampa *stretches* the hop length

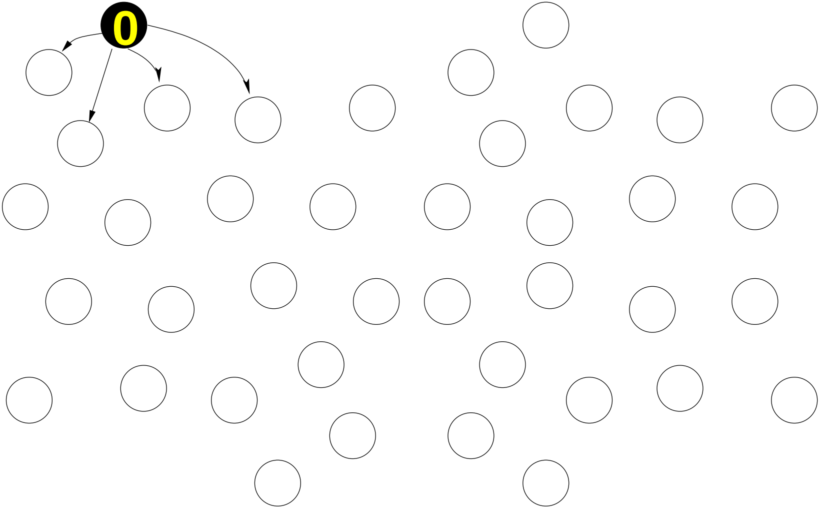
Pampa

- An all-purpose broadcast algorithm
  - Selects the nodes required to retransmit a broadcast message
    - Reduces redundancy
    - Saves bandwidth and power
  - Wait time is proportional to RSSI
    - Nodes more distant from the source wait less
  - Nodes do not retransmit if they hear a predefined number of retransmission



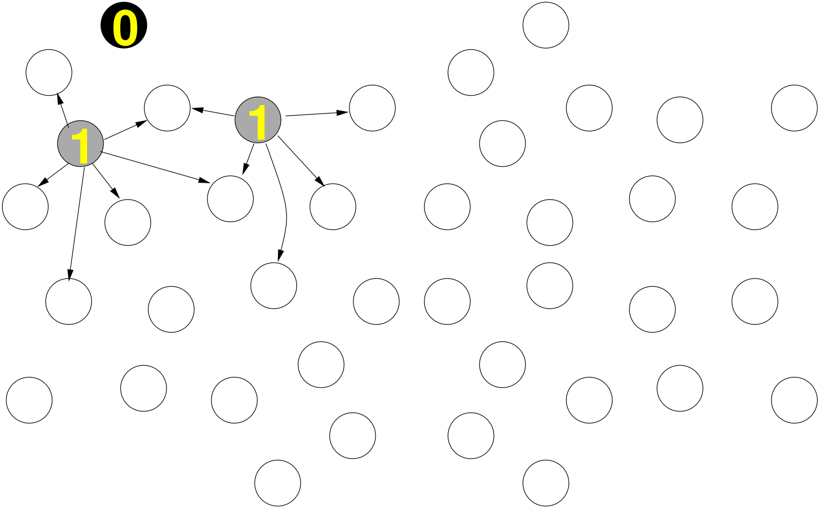
Pampa *stretches* the hop length

Dissemination Example



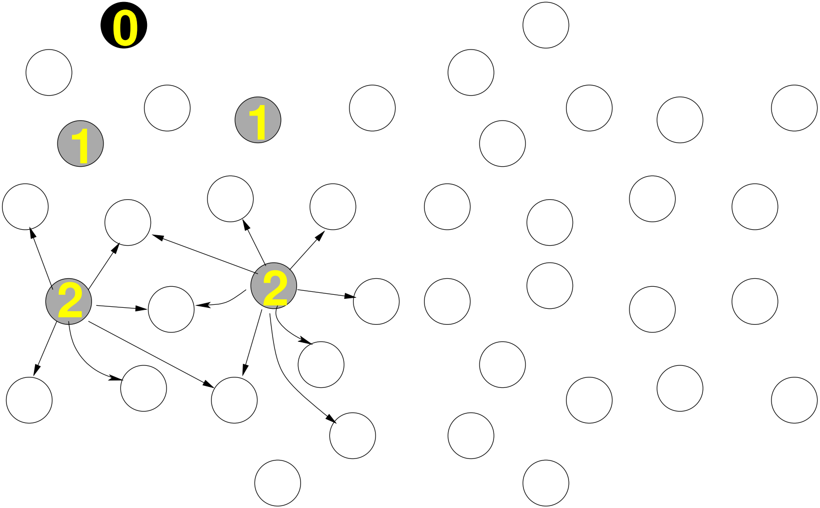
Numbers show TFS of the message. DbC=2

Dissemination Example



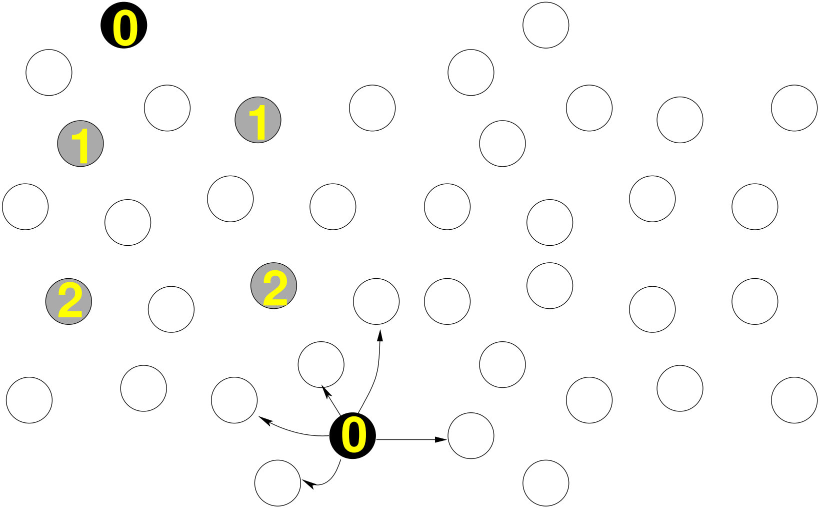
Numbers show TFS of the message. DbC=2

Dissemination Example



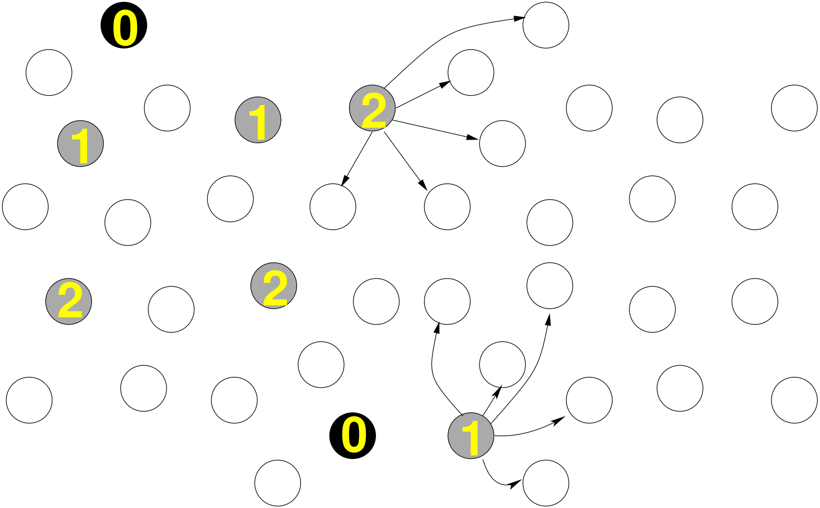
Numbers show TFS of the message. DbC=2

Dissemination Example



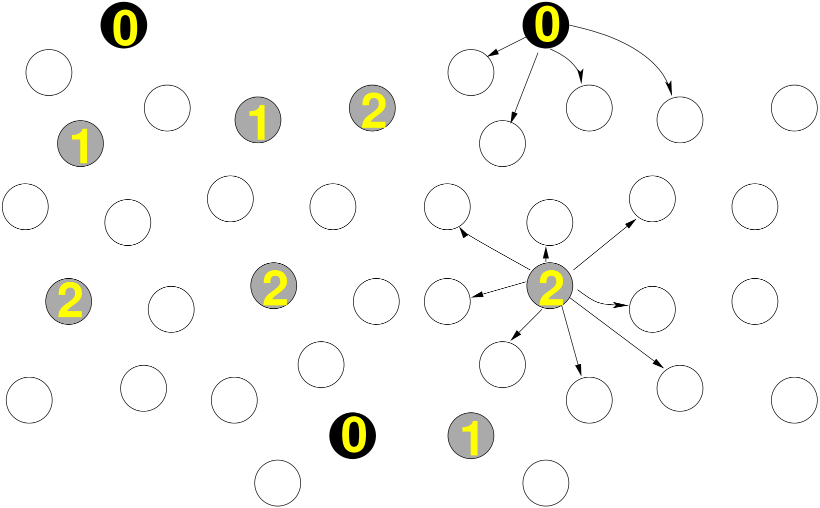
Numbers show TFS of the message. DbC=2

Dissemination Example



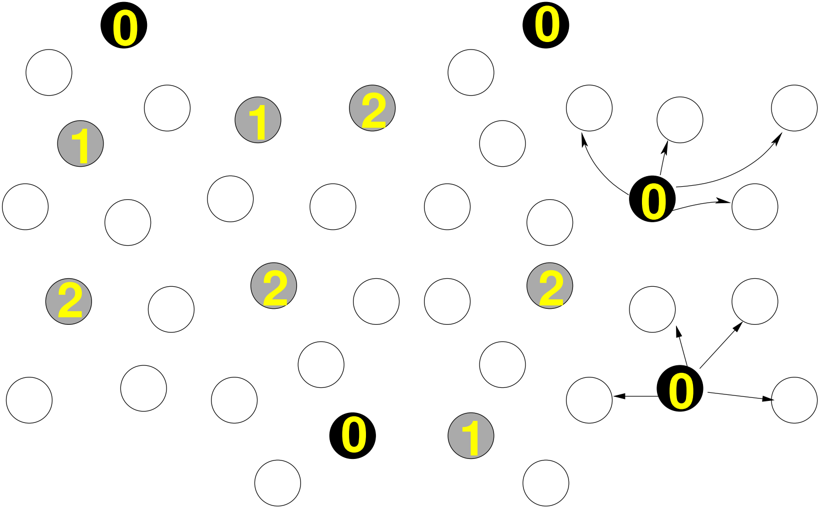
Numbers show TFS of the message. DbC=2

Dissemination Example



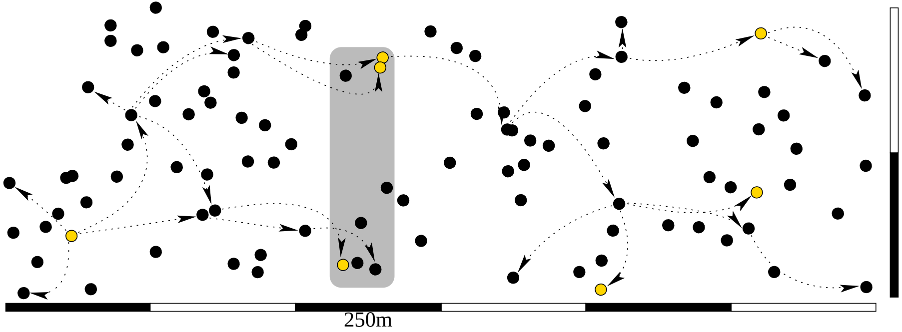
Numbers show TFS of the message. DbC=2

Dissemination Example



Numbers show TFS of the message. DbC=2

An example of data distribution



- ns–2
- 1500m×500m
- 100 nodes

- Arrows indicate devices that retransmited
- 7 copies
- 26 retransmissions

Query

- Two attempts
- Nodes first broadcast the query with a small TTL
  - The TTL adapts from past experiences
- If no reply is received, broadcast to all nodes
- Replies are sent point-to-point
  - Use the route constructed during query propagation (like DSR)
- Mitigation of a bad distribution
  - If a reply is received from more than DbC hops away, store the item

Evaluation

Factors Considered

Experiments varied:

- Node density
- Storage space at each node

$\Rightarrow$ Free space in the neighbourhood of each node

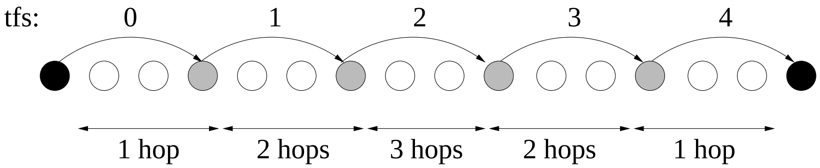
- Number of data items advertised

- Evaluated

Average Reply Distance How many hops away are the data items

Number of Messages How many messages are required for getting a reply

Analytical Properties



- If there is enough space available:
  - All nodes will have a copy at most DbC hops away
  - The majority will have a copy at most $\lceil \frac{DbC+1}{2} \rceil$ hops away

Expected Distance

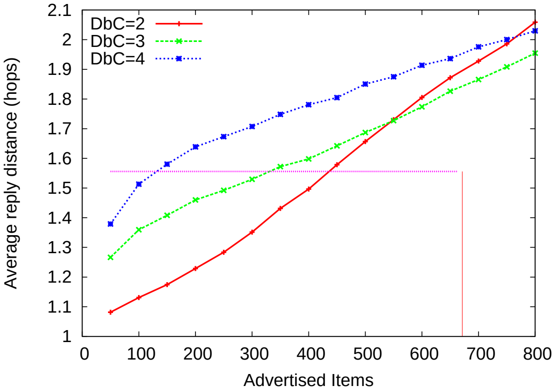
- Assuming:
  - An homogeneous distribution of the nodes
  - Sufficient storage space at the nodes
- The "average distance" at which a reply will be found is given by

$$\tau\,(\mathrm{DbC}) = \frac{\sum_{i=0}^{\mathrm{DbC}} \left(\left\lceil \frac{i+1}{2} \right\rceil (2i+1) \right)}{(DbC+1)^2}$$

- $\tau(2) = 1.55, \tau(3) = 1.75$ and $\tau(4) = 2.2$

Simulation Results

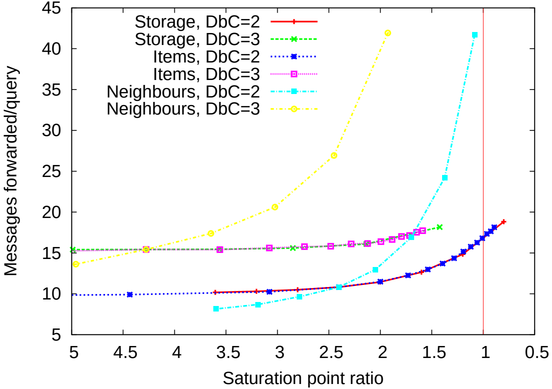
Distance of the replies



- Simulations in *ns-2*, 100 nodes
- Square defines the theoretical limit for DbC=2

Simulation Results

Traffic



- Changes in node density affect number of forwarders
- Degrades gracefully with the number of items and storage space

Conclusions

- We presented an algorithm for disseminating replicas of data items that:
  - Geographically distribute the replicas
    - Even when nodes are not aware of their location
  - Creates a number of replicas that depends of the region covered by the network
  - Requires a limited number of retransmissions

Future Work

- Leverage the distribution
  - Improve original distribution
  - Attend to node movement
  - Addition/removal of nodes
- Apply the algorithms to different applications