Gossip Based Dissemination

R. Friedman[1] A.-M. Kermarrec[2] H. Miranda[3]
L. Rodrigues[4]

[1] Technion, Israel [2] INRIA, France [3] U. of Lisbon, Portugal
[4] INESC-ID/IST, Portugal

MiNEMA Winter School
March 25th, 2009

Gossip Example
The Election Rumour

Case study
In 2005 campaign for the elections for the Portuguese parliament, one of the candidates spent 10m of the only debate on national TV:

Denying a rumour

- All candidates condemned the spreading of the rumour
- Media have frequently referred to it as "the rumour"
- Neither ever mentioned what the rumour was about
  But...
- The whole country knew what the rumour was about

Curious ?

Gossip Example
The Election Rumour

How to address your curiosity?

The test of the three questions (Socrates, 469-399 B.C) Are you sure? Is it something good? Is it useful?

The class would be over

Gossip Example
The Election Rumour

How to address your curiosity?

The test of the three questions (Socrates, 469-399 B.C) Are you sure? Is it something good? Is it useful?

The class would be over

The push approach I tell you the rumour

Gossip Example
The Election Rumour

How to address your curiosity?

The test of the three questions (Socrates, 469-399 B.C) Are you sure? Is it something good? Is it useful?

The class would be over

The push approach I tell you the rumour

The pull approach I told you that I know a rumour

You'll ask if you want to know

Gossip Example
The Election Rumour

How many times will I have to say it?

$$\left.\begin{array}{l} 92 \text{ participants} \\ 12 \text{ Portuguese} \end{array}\right\} \frac{92-12}{12} \approx 7?$$

Or

- Each Portuguese tells to 3 other participants

$$12 \times 3 = 36$$

Gossip Example
The Election Rumour

How many times will I have to say it?

$$\left.\begin{array}{l} 92 \text{ participants} \\ 12 \text{ Portuguese} \end{array}\right\} \frac{92-12}{12} \approx 7?$$

Or

- Each Portuguese tells to 3 other participants

$$12 \times 3 = 36$$

- Each of the 36 tells to 3 other

$$36 \times 3 = 108 > 92$$

Gossip Example
The Election Rumour

How hard is it to get 10 millions?

$$12 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 = 19131876$$

13 rounds

Epidemic Dissemination Examples

- The Great Plague (14th Century)
  - Killed 75M people worldwide
    - In Europe: 25-50M (30%-60% of Europe's population)
  - Started in Asia
  - Transported to Europe by merchants over the route of silk
  - Used as weapon by Mongols
- The Spanish Flu (1918-1920)
  - Killed 50M-100M people worldwide
    - India, North America, Europe, Australia, Western Samoa, Fiji Islands
  - Started in Fort Riley, Kansas, US
  - Infected soldiers were deployed in different places in Europe

Source: Wikipedia

Epidemic Dissemination Pattern

- Infection occurs upon spontaneous contacts between individuals
- Infected individuals become infection agents

Properties

- Scalable
- Resilient to failures
  - See measures to control birds flu

May Not Work

What would have happened if. . .

- The silk route didn't existed?

  – or –

  American soldiers remained in Fort Riley?
- People used the Socratic approach on rumours?

  – or –

  The diseases killed instantly?

$$\Downarrow$$

Dissemination would have been interrupted

Computers don't gossip,,,
do they?

Some computer virus algorithm

- At each computer
  1. Peek a few IP addresses chosen at random
  2. Infect them (if possible)
  3. Do some harm on this computer

Note: the order is important

Gossip algorithms
Outline

Sender thread

Forever

1. peak **fanout** nodes from local node list **(the view)**
2. send
   [push] the messages in the buffer
   [pull] the ID of the messages in the buffer
3. purge buffer from old messages
4. sleep

Receiver thread

Forever

1. receive message
2. compare buffer with message
3. for each message m not in the buffer
   1. If "pull": request m
   2. Deliver m
   3. Add m to the buffer

Questions

- What is an “old” message?
  - FIFO
  - Random based
  - Semantic based
- How to build a view in large scale?
  - Random walk of “join” messages
  - Use a gossip protocol

Robust?

What would happen if...

- nodes were clustered, sharing similar views?
- all infected nodes are affected by a failure?
- message is removed from buffer at all infected nodes?

More likely in early stages of dissemination

$$\Downarrow$$

Bimodal behaviour

Applications
On Wired Networks

Large scale applications. . .

- Database consistency
- Data fusion/aggregation
- Data dissemination
- Peer sampling

Is Gossip Applicable in Infrastructure-less Networks?

Not directly:

- Gossip is resource consuming
  - *rounds*×*fanout* messages per node
- Each message may trigger a route request
  - Each route request may require a flooding

Route Driven Gossip Populates the view with nodes at the route cache of reactive routing protocols [536]

Tailoring Gossip to Infrastructure-less Networks

Redefining a view

A view of node n is the list of nodes within transmission range of n

$$\text{fanout} = \#\text{view}$$

Redefining sleep time

Nodes sleep while their buffer doesn't change

Tailoring Gossip to Infrastructure-less Networks

Advantages

- All nodes in the view can be reached by a single transmission

Implications

- Changes in the view are mostly dictated by node movement
- The view size is outside the control of the algorithm
- Clustering

Problem Solved?

Not yet! We're using a shared medium:

- All nodes will wake to retransmit the new message
- Compete for the medium
- Produce collisions

$$\Downarrow$$

Broadcast Storm [818]

Selecting Retransmitters

- Infection can be propagated with a small number of retransmissions
- Different approaches for selection of retransmitters
  - Probabilistic
  - Counter-based
  - Distance-based
  - Mix of the above

Requirement: decision should be local to each node

Probabilistic Selection

GOSSIP1(p) [353]

Upon reception of a message for the first time, nodes retransmit with probability p

- Flooding: $p = 1$
- Highly vulnerable in the neighbourhood of the source

Biased Probabilistic Selection

GOSSIP1(p,k) [353]

A node retransmits with $p = 1$ if located at less than k hops from the source or with p otherwise

- Vulnerable to low densities elsewhere

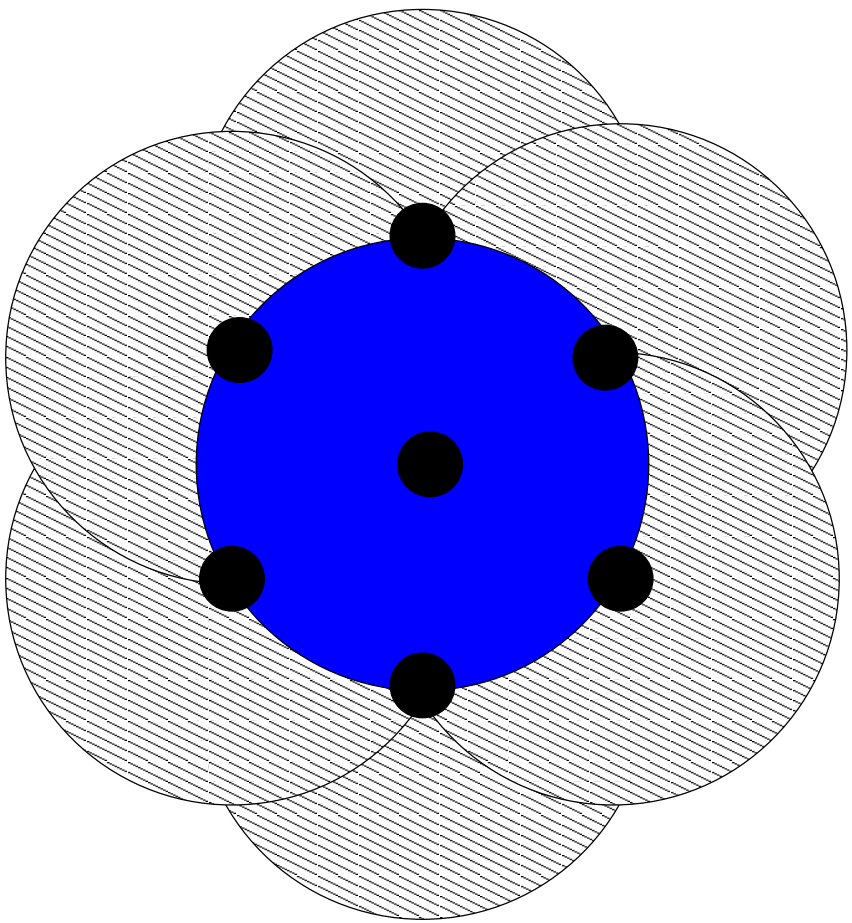
GOSSIP2(p1,k,p2,n) [353]

A node retransmits with $p2$ if it has less than n neighbours. Uses GOSSIP1(p,k) otherwise.

- How to learn n?
- What is a good value for $p1$?

Counter-based Selection

A good number of transmitters does not depend of the size of the view.
In ideal conditions, 6 retransmissions is close to optimal [818]

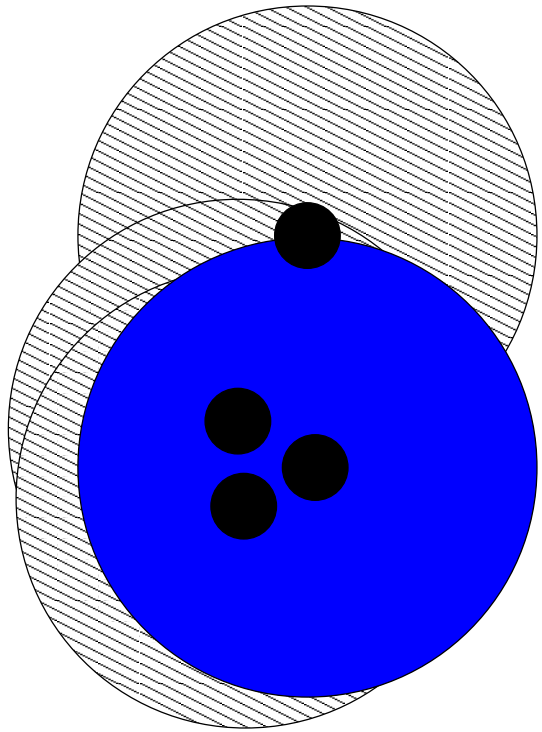


Counter-based Selection

Counter-Based(c) [818]

After sleeping for a random amount of time, a node retransmits if less than c retransmissions were received

- Random node selection

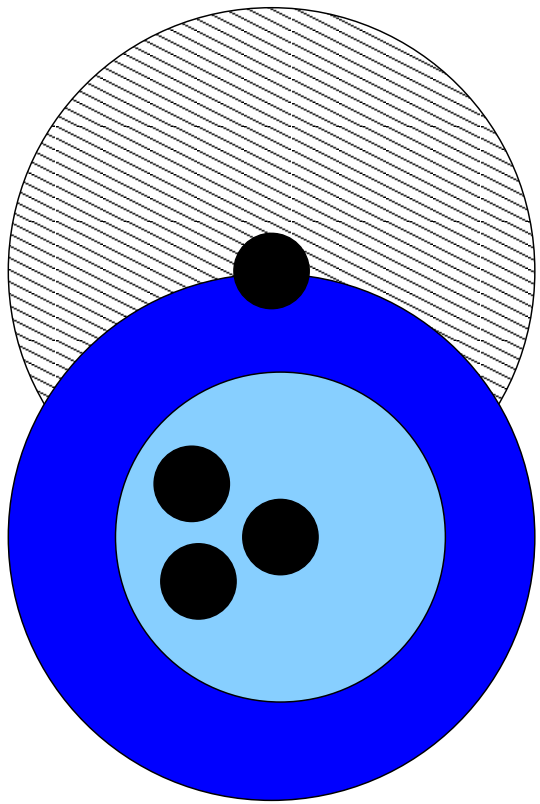


Distance-based Selection

Distance-Based(d) [818]

A node retransmits if more distant than d of any previous retransmission

- Some retransmission is better than none



Probabilistic+Counter based Selection

GOSSIP3(p,k,m) [353]

Apply GOSSIP1(p,k). Retransmit if after a short period, less than m retransmissions were received

- p can be hill tuned and result in spurious retransmissions

RAPID [236]

Retransmit with probability $\min\{\frac{3.5}{\#view}, 1\}$ *or if after a short period less than 2 retransmissions were received*

- Random node selection

Distance+Counter based Selection

PAMPA(c) [583]

Sleep for $k \times RSSI$. Transmit if less than c retransmissions were received.

- Depends of unreliable and coarse-grained RSSI
- Node selection may stop propagation for other clusters

Pull-based Approaches

EraMobile Source pushes to 1-hop neighbours, pull-based afterwards

- Large latency
- Spurious requests
- Periodic transmission

RAPID Complements push-based with periodic broadcast of the state of buffers

- Periodic transmission
- Also used for counting neighbours

Pull-based Approaches

Autonomous Gossip survival of the fittest approach [213]

- Nodes as "habitats"
- Messages as living entities
  - Migrate, Reproduce and Die
- Messages have an "utility" for each node
  - Dictates the suitability of the habitat for the message
  - Derived from user preferences

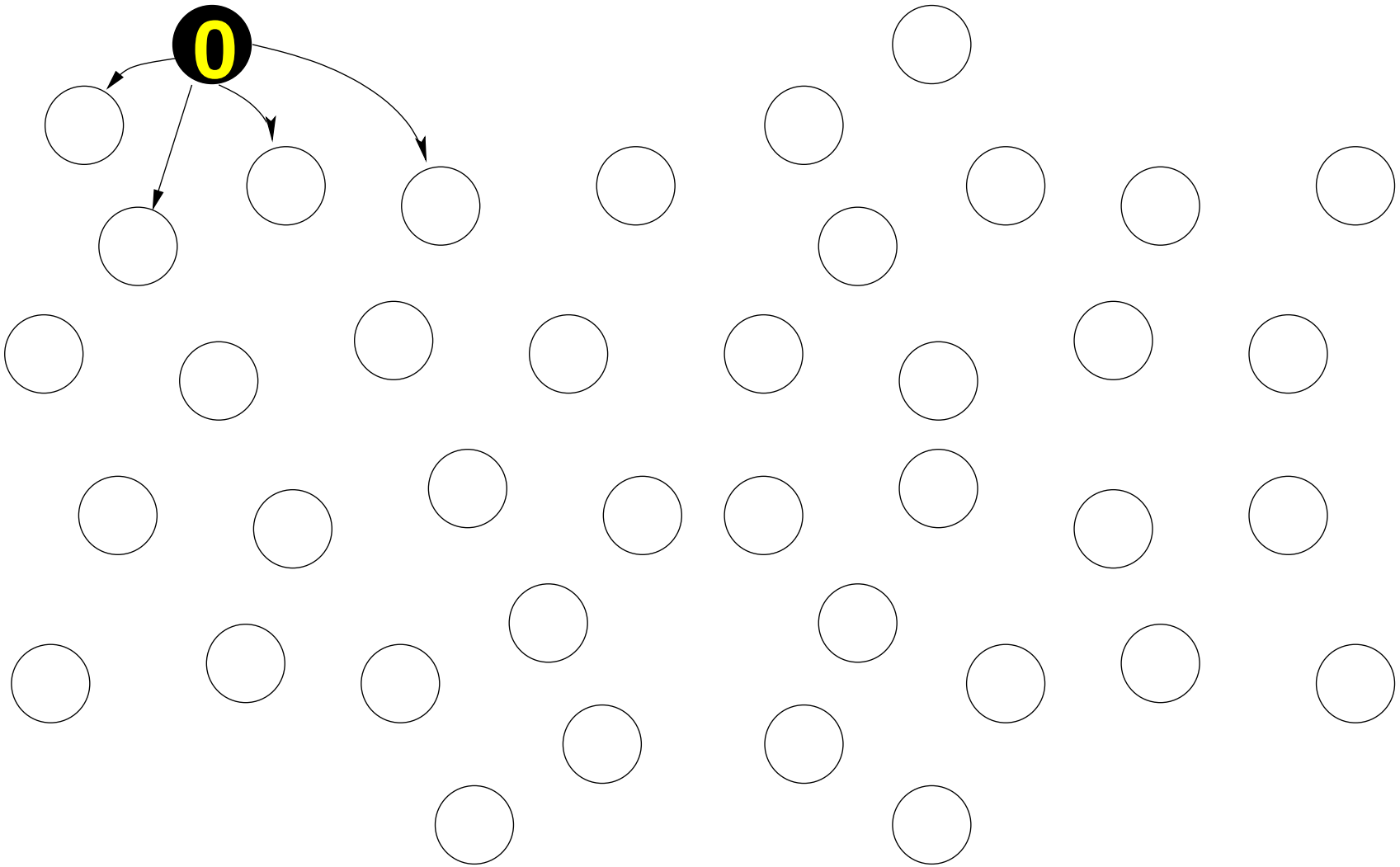
PCache [584]

- Replicates small sized data items in infrastructure-less networks
- Geographically distribute the replicas
  - Reduces latency
  - Improves resilience to localised failures (e.g. interference)
  - Saves bandwidth and battery
- Assumes that nodes:
  - are not aware of their location
  - have limited storage space

Distributed Data Storage
The PCache Approach

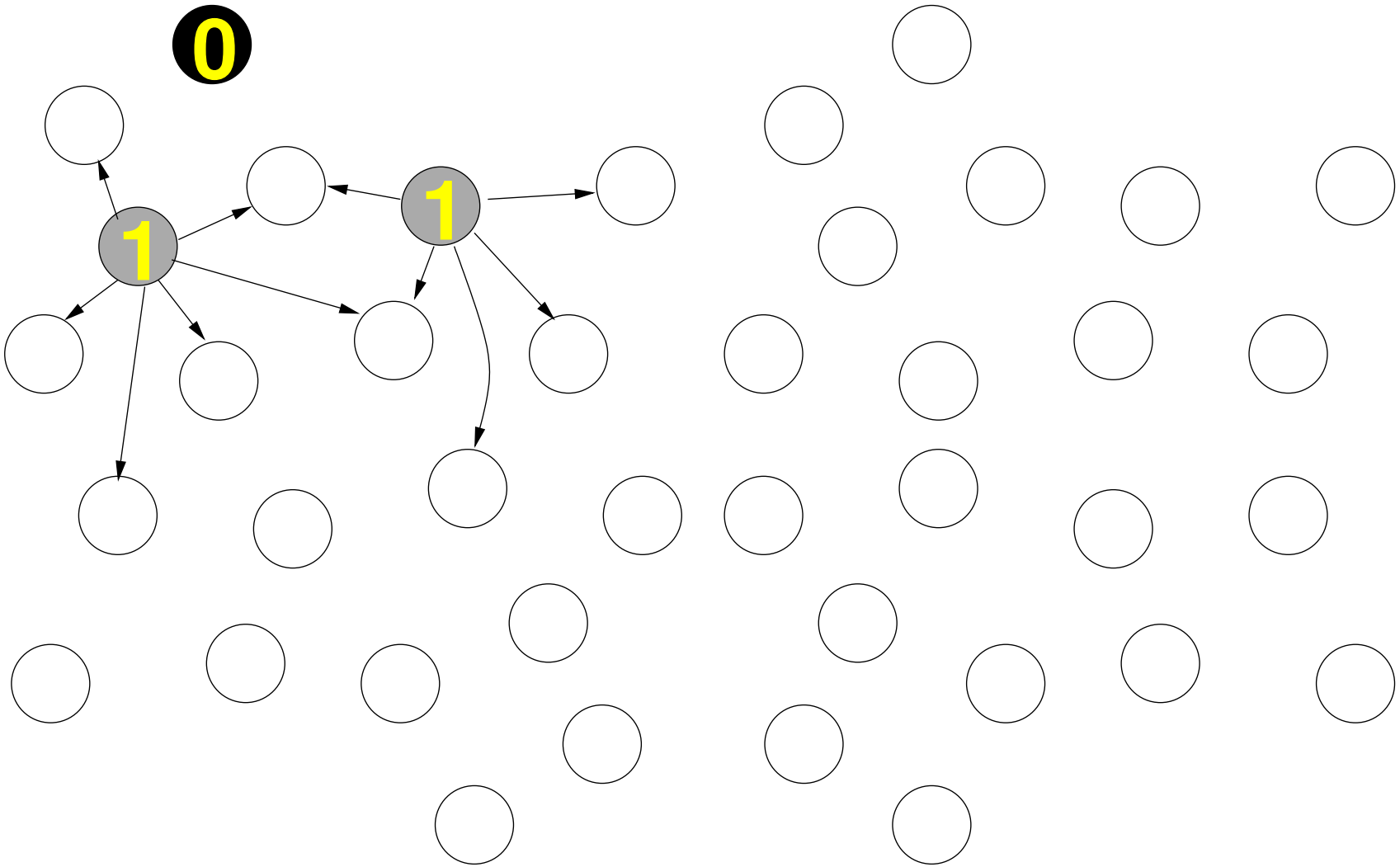
- Replica distribution: PADIS (PAMPA++)
  - Distribution message carries a counter (TFS)
  - Counts the distance (in hops) to the closest copy
  - Item is stored when distance reaches some threshold

Dissemination Example



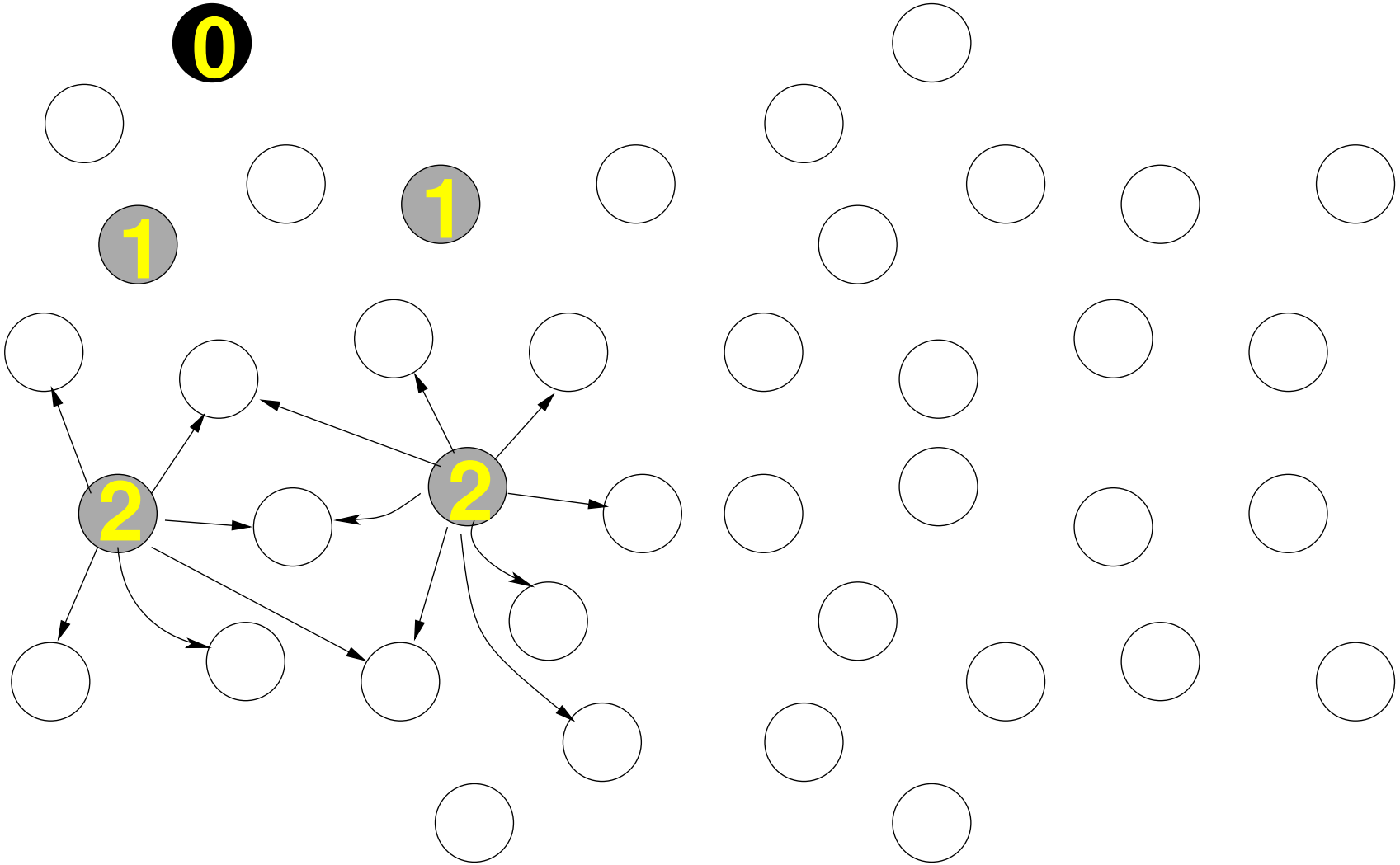
Numbers show TFS of the message. Threshold=2

Dissemination Example



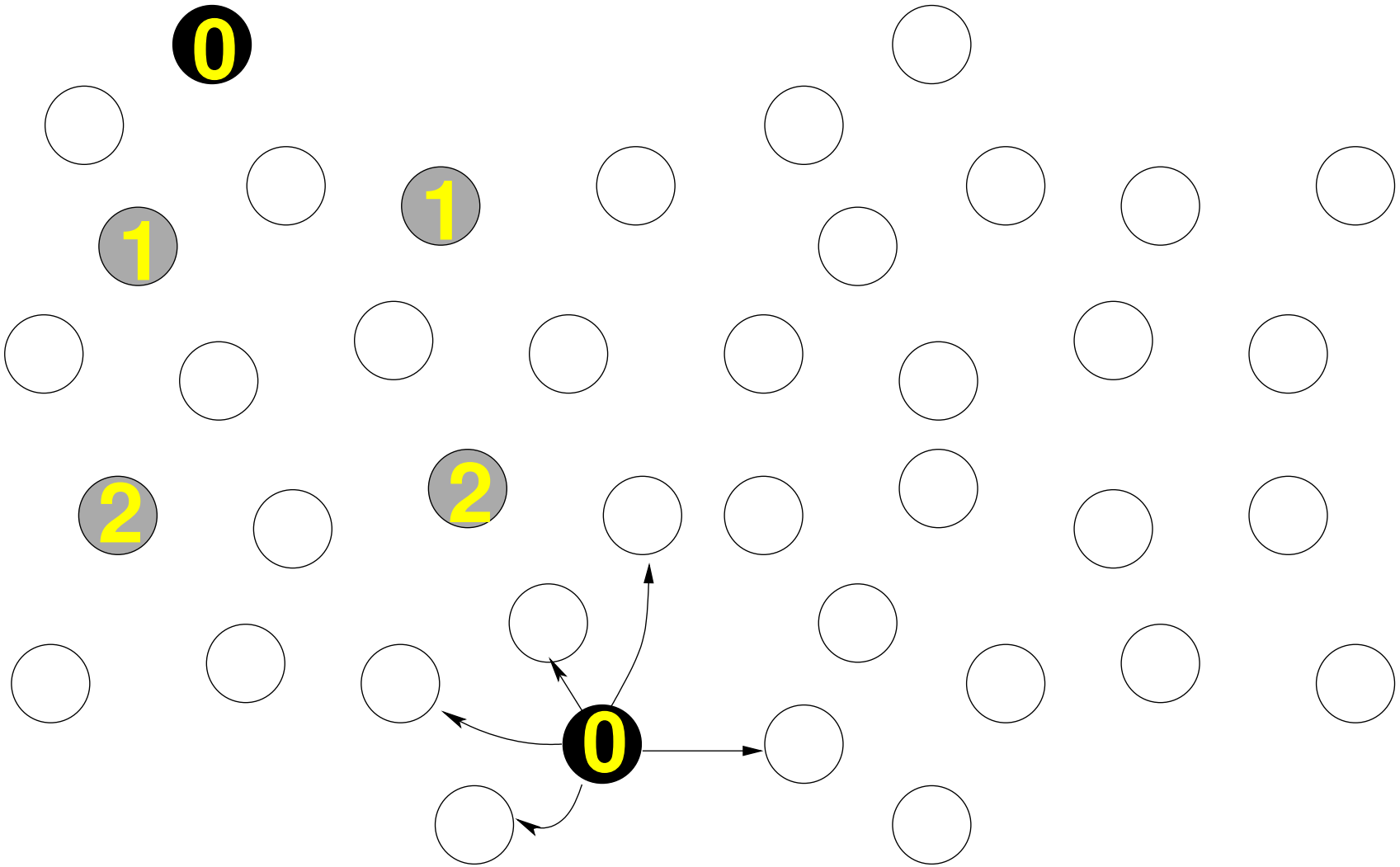
Numbers show TFS of the message. Threshold=2

Dissemination Example



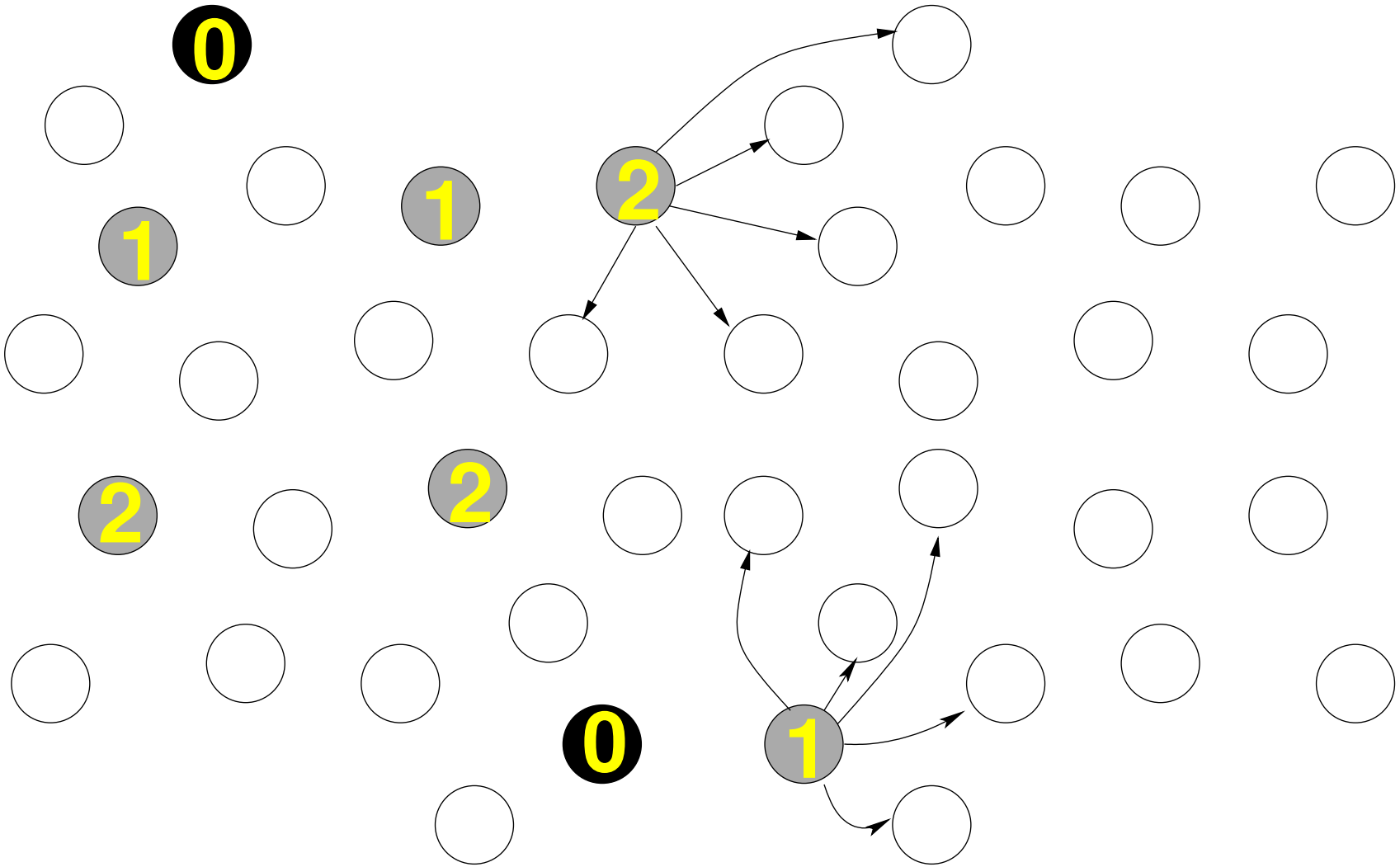
Numbers show TFS of the message. Threshold=2

Dissemination Example



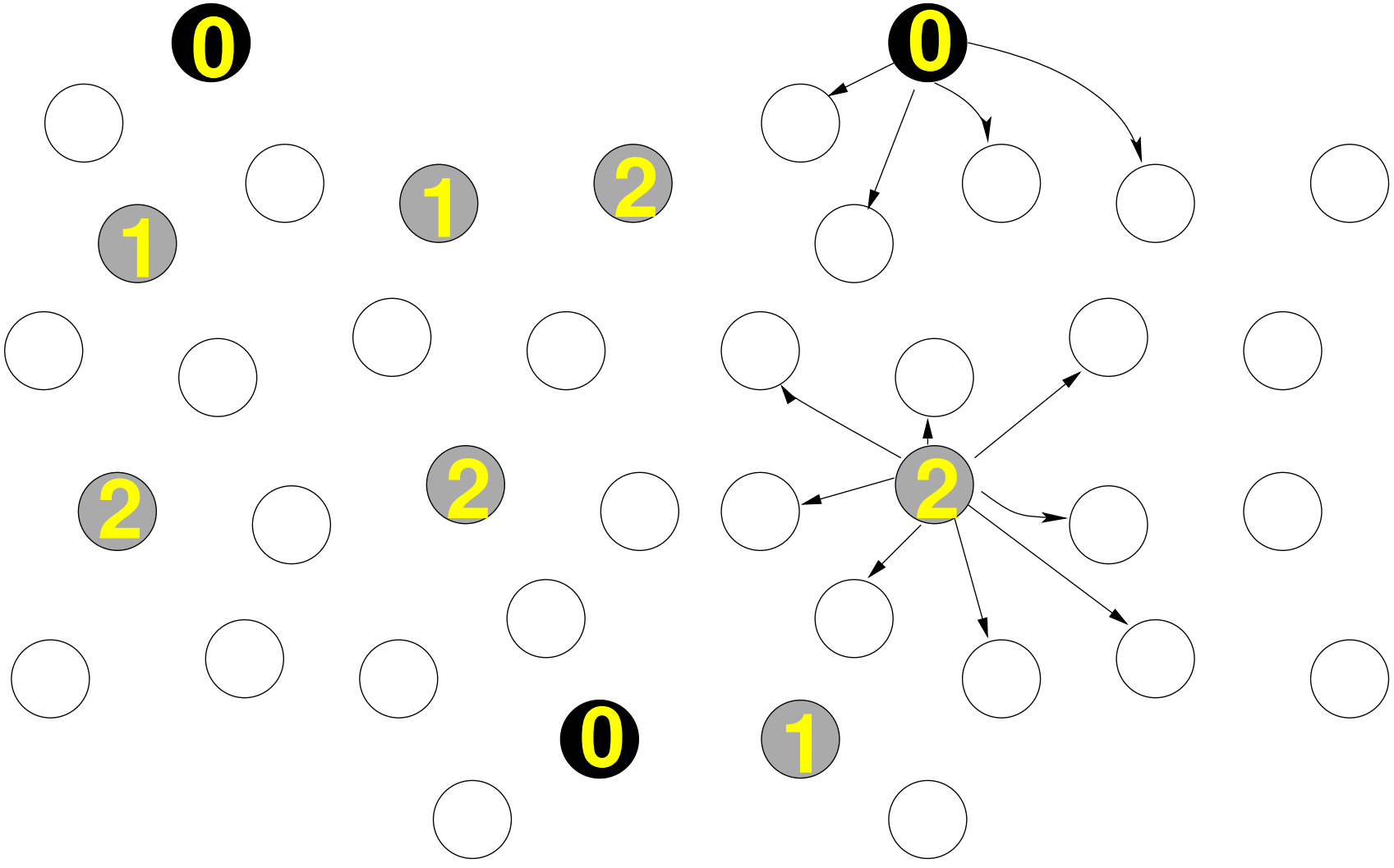
Numbers show TFS of the message. Threshold=2

Dissemination Example



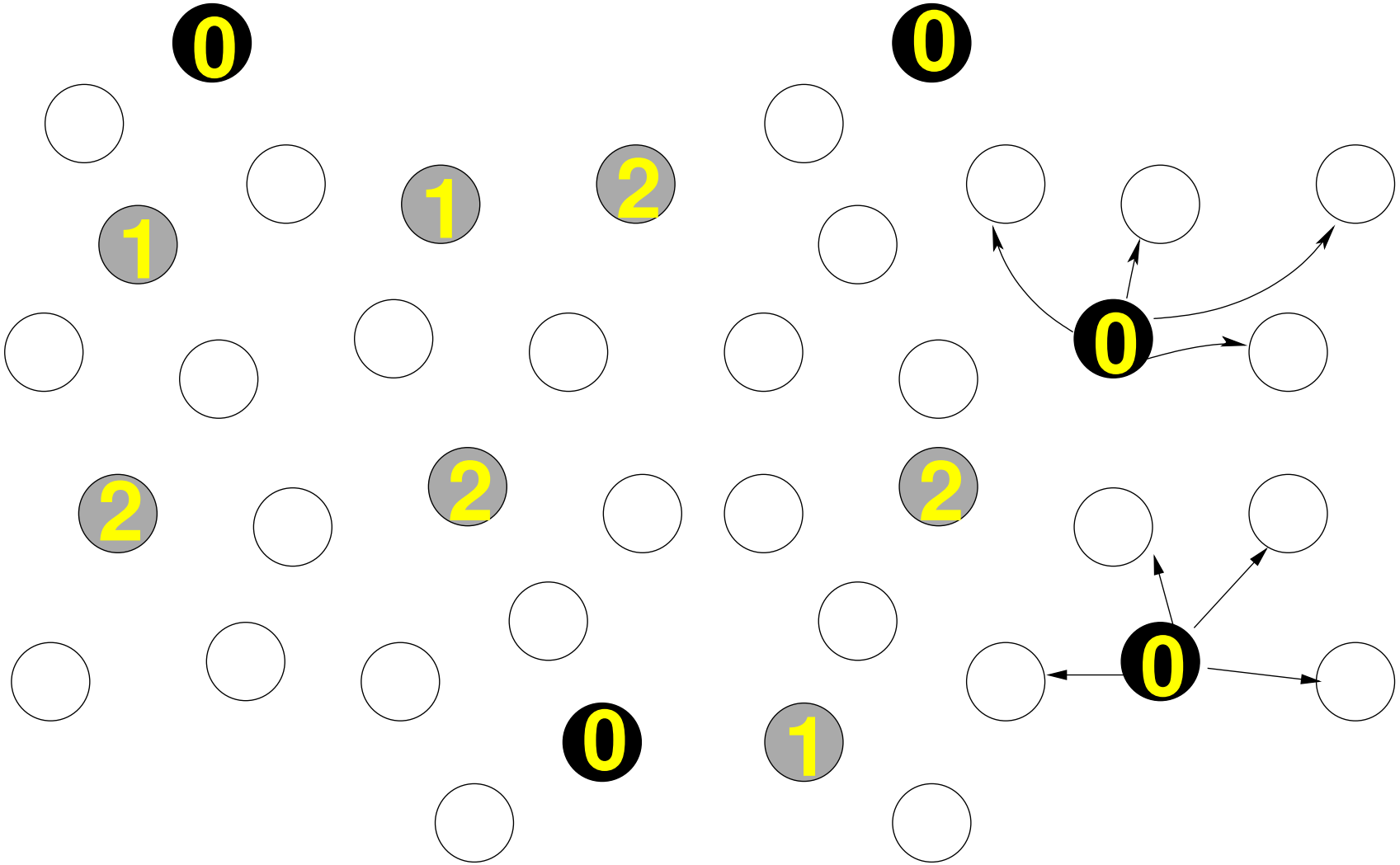
Numbers show TFS of the message. Threshold=2

Dissemination Example



Numbers show TFS of the message. Threshold=2

Dissemination Example



Numbers show TFS of the message. Threshold=2

Distributed Data Storage
The PCache Approach

Replica shuffling

- Keep geographic distribution in the presence of node movement
- Experimented two approaches
  - **Baseline** Nodes store items they query
    - If the reply came from far away
  - **Probabilistic** Nodes piggyback a subset of their caches in queries
    - Low-cost gossip
    - Messages can be edited by nodes that retransmit
    - Nodes probabilistically decide to store piggybacked data
      - Depending on the number of hops travelled

Results

- Probabilistic makes distance to "converge" faster
  - in the presence of node movement
  - to leverage an irregular distribution
- With a more irregular distribution of the number of copies
  - Not necessarily bad

Conclusions

- Gossip is an efficient dissemination mechanism
- Suitable for infrastructure-less networks after some adaptations
- Application examples
  - Message dissemination
  - Distributed data storage