Multiplayer Online Games over Scale-Free Networks: a Viable Solution?

Gabriele D’Angelo

<gda@cs.unibo.it>

http://www.cs.unibo.it/gdangelo/

joint work with:

Stefano Ferretti

Torremolinos, Malaga (Spain)

DIstributed **SI**mulation & **O**nline gaming (DISIO), 2010

Presentation **outline**

- Multiplayer Online Games: scalability and responsiveness
- MOGs: a peer-to-peer approach
- Scale-free networks
- Gossip protocols
- Performance evaluation: simulation-based
- Performance evaluation: metrics
- Experimental evaluation
- Conclusions and future work

MOGs: scalability and responsiveness

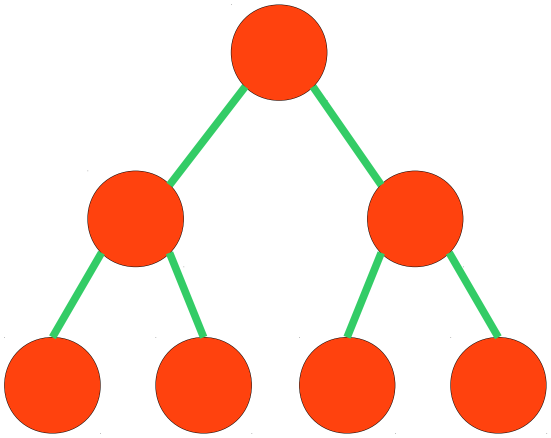
- **Scalability** and **responsiveness** are open problems in Multiplayer Online Games (MOGs)
- Several architectures have been proposed to support MOGs:
  - client/server
  - mirrored servers
  - peer-to-peer
- Given the distributed nature of this kind of applications, the **dissemination** of game events can be very **costly**
- Under the **scalability** viewpoint the **peer-to-peer approach** is very promising

MOGs: a **peer-to-peer** approach

- Each peer locally manages its copy of the **game state**
- The peers are organized in some form of **overlay network**
- The dissemination of game events is obtained by passing messages through the overlay
- What is the **best form of overlay**?
  - tree
  - fully connected graph
  - random graph
  - scale-free network

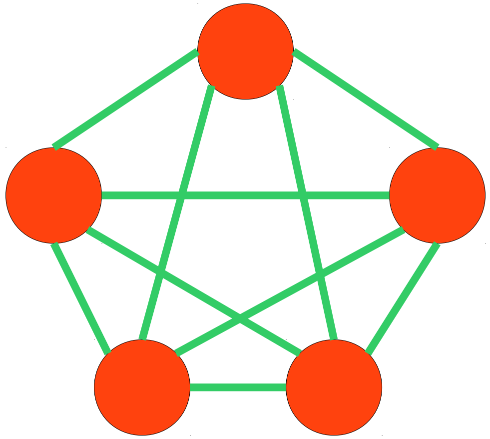
MOGs: a **peer-to-peer** approach

- Each peer locally manages its copy of the **game state**
- The peers are organized in some form of **overlay network**
- The dissemination of game events is obtained by passing messages through the overlay
- What is the **best form of overlay**?
  - tree
  - fully connected graph
  - random graph
  - scale-free network



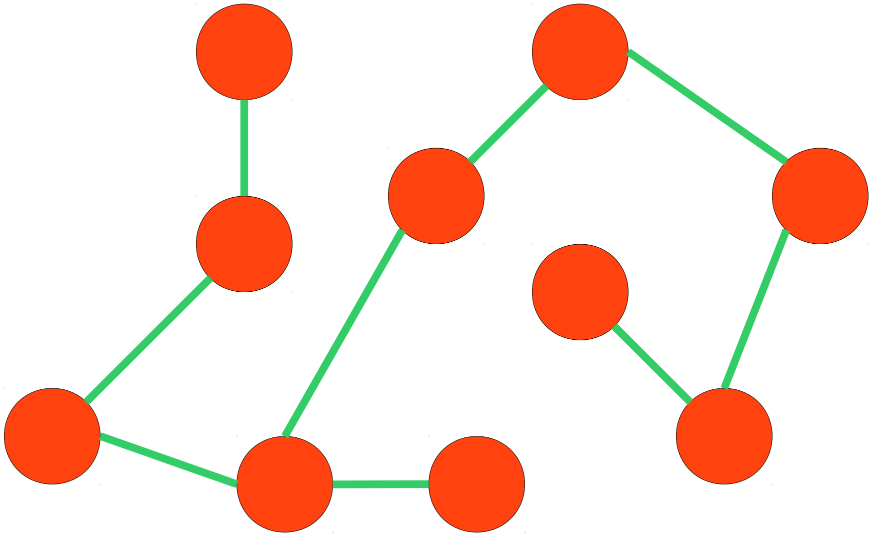
MOGs: a **peer-to-peer** approach

- Each peer locally manages its copy of the **game state**
- The peers are organized in some form of **overlay network**
- The dissemination of game events is obtained by passing messages through the overlay
- What is the **best form of overlay**?
  - tree
  - fully connected graph
  - random graph
  - scale-free network



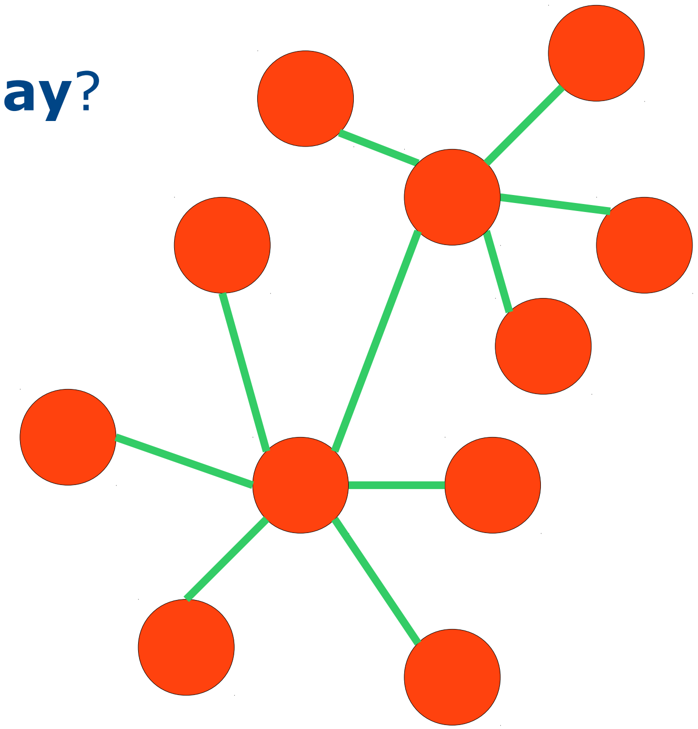
MOGs: a **peer-to-peer** approach

- Each peer locally manages its copy of the **game state**
- The peers are organized in some form of **overlay network**
- The dissemination of game events is obtained by passing messages through the overlay
- What is the **best form of overlay**?
  - tree
  - fully connected graph
  - random graph
  - scale-free network

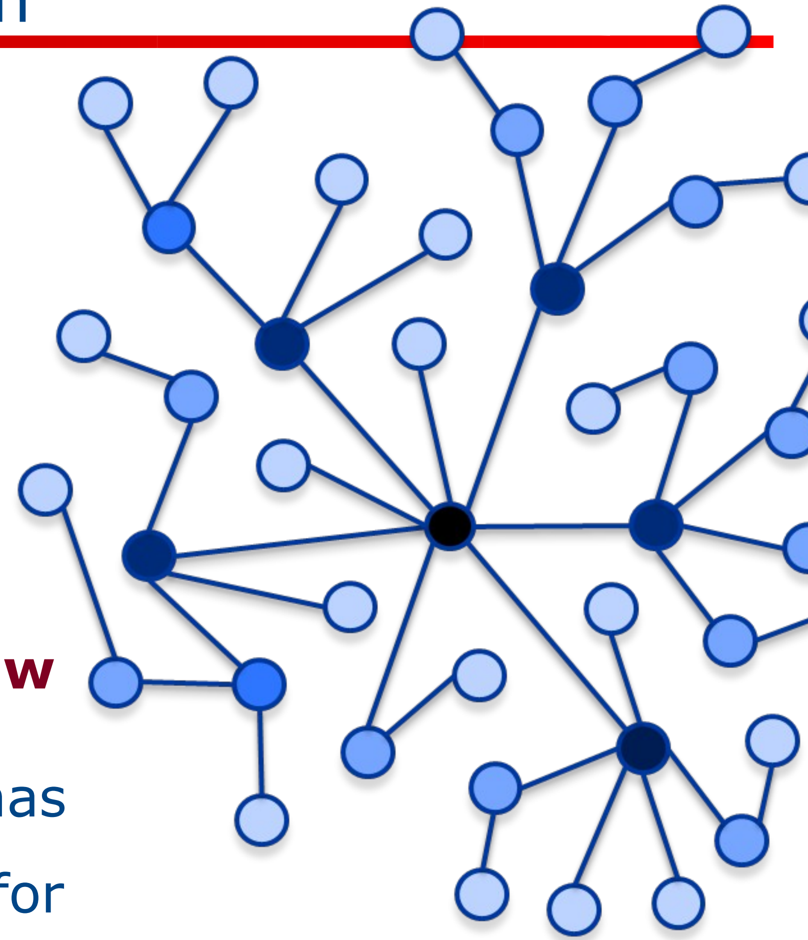


MOGs: a **peer-to-peer** approach

- Each peer locally manages its copy of the **game state**
- The peers are organized in some form of **overlay network**
- The dissemination of game events is obtained by passing messages through the overlay
- What is the **best form of overlay**?
  - tree
  - fully connected graph
  - random graph
  - scale-free network



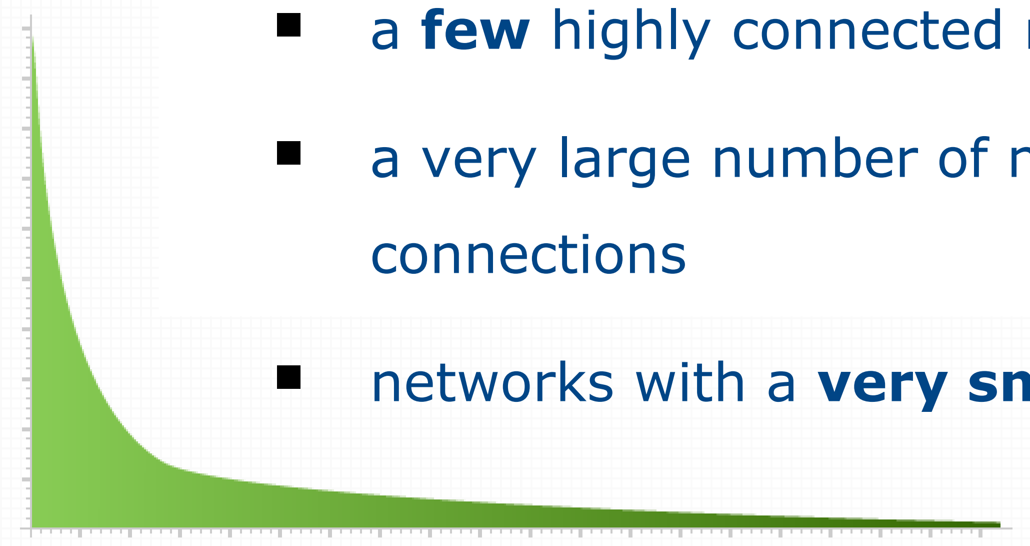
Scale-free networks: definition



- A graph can be used to represent a network and its connectivity
- **Degree of a node** = number of neighbor nodes attached to it
- A scale-free network has a degree distribution that follows a **power law**
- If $\mathbf{p^k}$ is the probability that a node has a degree equal to **k** then: $\mathbf{p^k \sim k^{-\alpha}}$ for some constant value $\boldsymbol{\alpha}$ (usually: $2 < \boldsymbol{\alpha} < 3$)
- **Many examples** in the real world: the Web, transmission of diseases, citation graphs, social networks interactions etc.

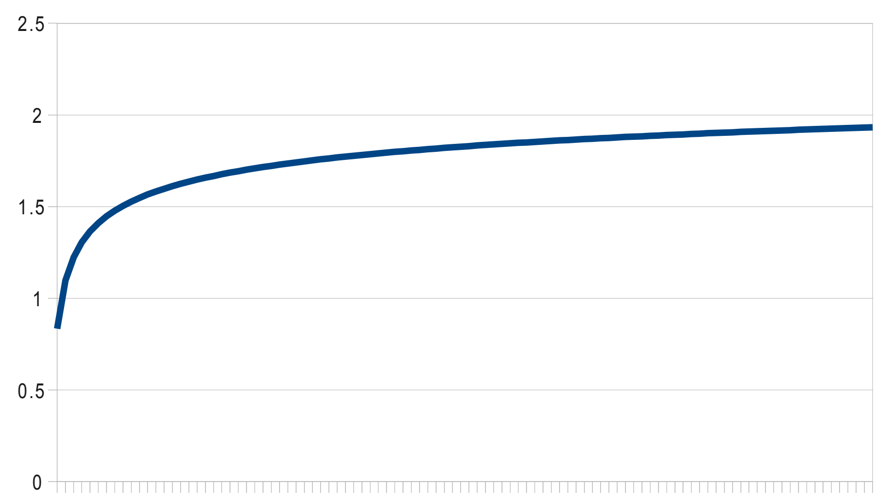
Scale-free networks: properties

- **This means:**
  - a **few** highly connected nodes (**hubs**)
  - a very large number of nodes with a few connections
  - networks with a **very small diameter**



- **For example:**
  - with **2 < α < 3**

    the diameter of a network with **N** nodes is **~ ln ln (N)**



Scale-free networks for MOGs

- Our proposal is to implement MOGs using a peer-to-peer architecture that is partially "unstructured" and "spontaneous"
- Some properties of **scale-free networks** (such as the diameter) can be very valuable in supporting scalability and responsiveness
- In our view, the **dissemination of game events** will be obtained through probabilistic approaches, for example using **gossip protocols**
- The game events generated at peers are disseminated to the whole network, using very simple gossip protocols and **without any form of centralization** or **predefined routing**

Gossip protocol #1: **probabilistic broadcast**

- If the message is locally generated then it is broadcasted to all neighbors, otherwise it is decided at random if it will be broadcasted or ignored

PARAMETERS:

- $\mathbf{p_b}$ = probability to broadcast a message

ADDITIONAL MECHANISMS:

- time to live (**ttl**) in each message
- local **cache** in each node

ALGORITHM

```
function INITIALIZATION()
p_b ← PROBABILITY_BROADCAST()


function GOSSIP(msg)
if  (RANDOM() < p_b or
    FIRST_TRANSMISSION())
    then
        for all n_j in Π_j do
                  SEND(msg, n_j)
        end for
end if
```

Gossip protocol #2: **fixed probability**

- For each received message, the node randomly selects those edges through which the message must be propagated

PARAMETERS:

- **v** = threshold value

ADDITIONAL MECHANISMS:

- time to live (**ttl**) in each message
- local **cache** in each node

ALGORITHM

```
function INITIALIZATION()
v ← CHOOSE_PROBABILITY()

function GOSSIP(msg)
for all n_j in Π_j do
    if RANDOM() < v then
        SEND(msg, n_j)
    end if
end for
```

Gossip protocol #3: **fixed fanout**

- Each message is sent only to a limited number of nodes, the receivers are selected at **random** among the neighbors

PARAMETERS:

- **fanout** = total number of receivers

ADDITIONAL MECHANISMS:

- time to live (**ttl**) in each message
- local **cache** in each node

ALGORITHM

```
function INITIALIZATION()
fanout ← RETRIEVE_FANOUT()

function GOSSIP(msg)
if  fanout ≥ |Πj| then
    toSend ← Πj
else
    SELECT_NODES()
end if
for all nj in toSend do
    SEND(msg, nj)
end for
```

Performance evaluation: simulation-based

- The following performance evaluation is based on **simulation**
- New features implemented in the Parallel and distributed Scale-free network Simulator (**PaScaS**): http://pads.cs.unibo.it

Parameter	Value
number of **nodes**	0-500
message generation	*exponential distribution* *mean* = 50 *time-steps*
cache size (local to each node)	256 *slots*
message Time To Live (**ttl**)	6
probability of dissemination (**v**)	0.5, 0.8, 1 *(i.e. 50-80-100%)*
fanout value	5 (# of nodes)
probability of broadcast ($\mathbf{p_b}$)	0.5, 0.8, 1 *(i.e. 50-80-100%)*
simulated time (gaming time)	1000 *time-steps (after building)*

Performance evaluation: **metrics**

- **Coverage**
  - percentage of nodes that have received all the messages that have been produced during the whole game execution

    "are the game events received by all gamers?"

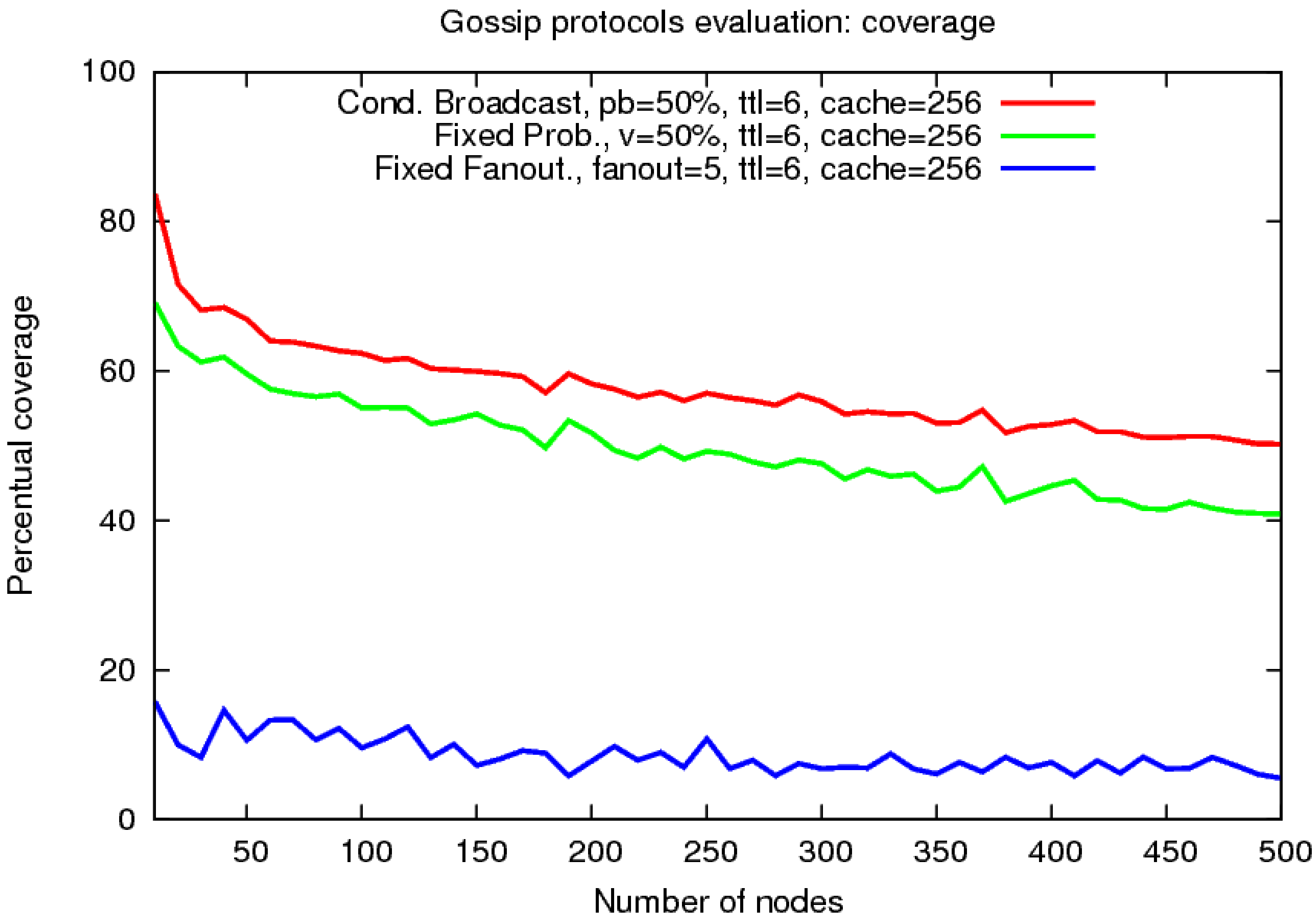
- **Delay**
  - average number of hops (that is time-steps) necessary to receive a message after its creation

    "is the data dissemination really responsive?"

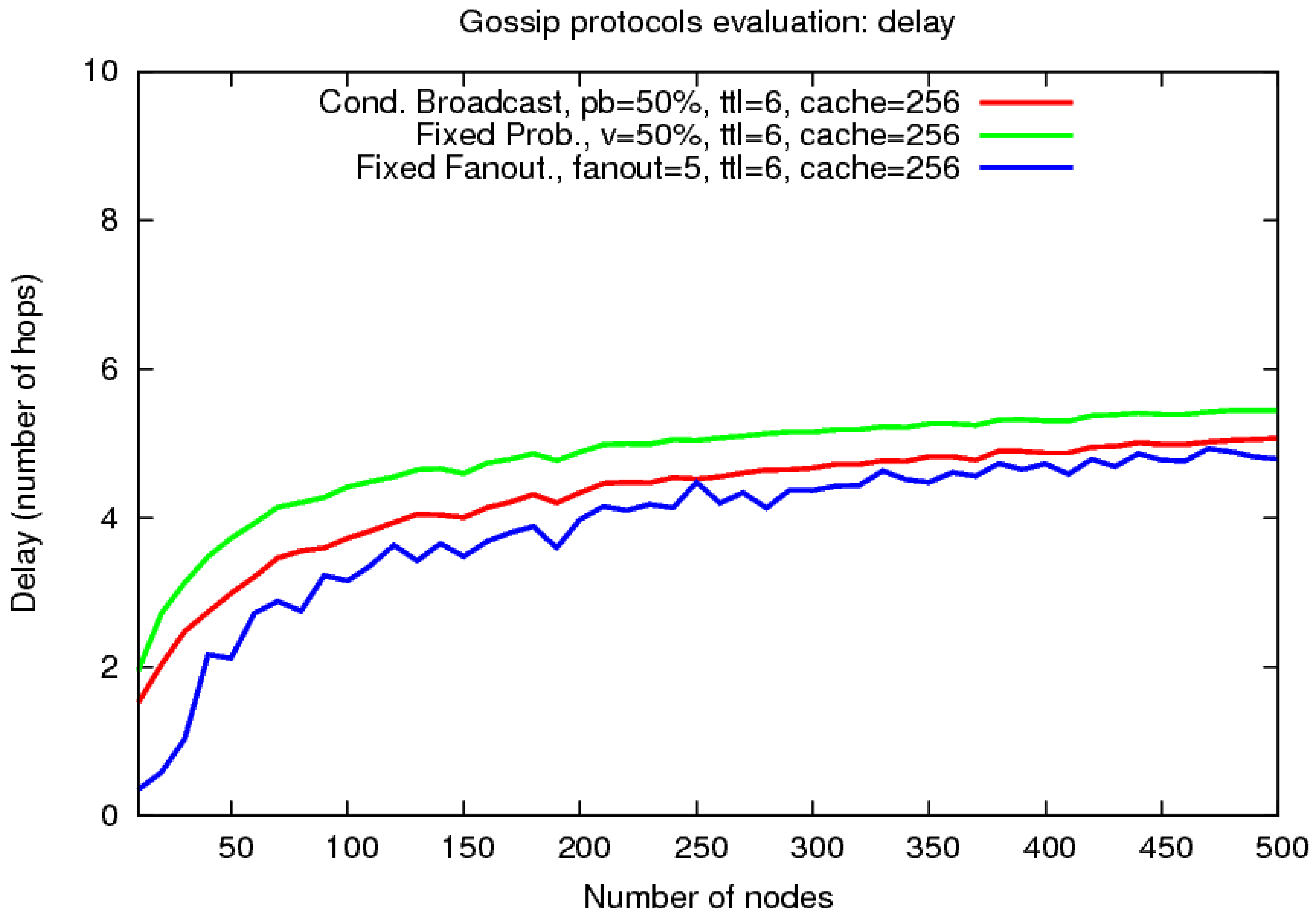
- **Messages**
  - total number of messages routed in the game execution

    "what is the overhead due to the events dissemination?"

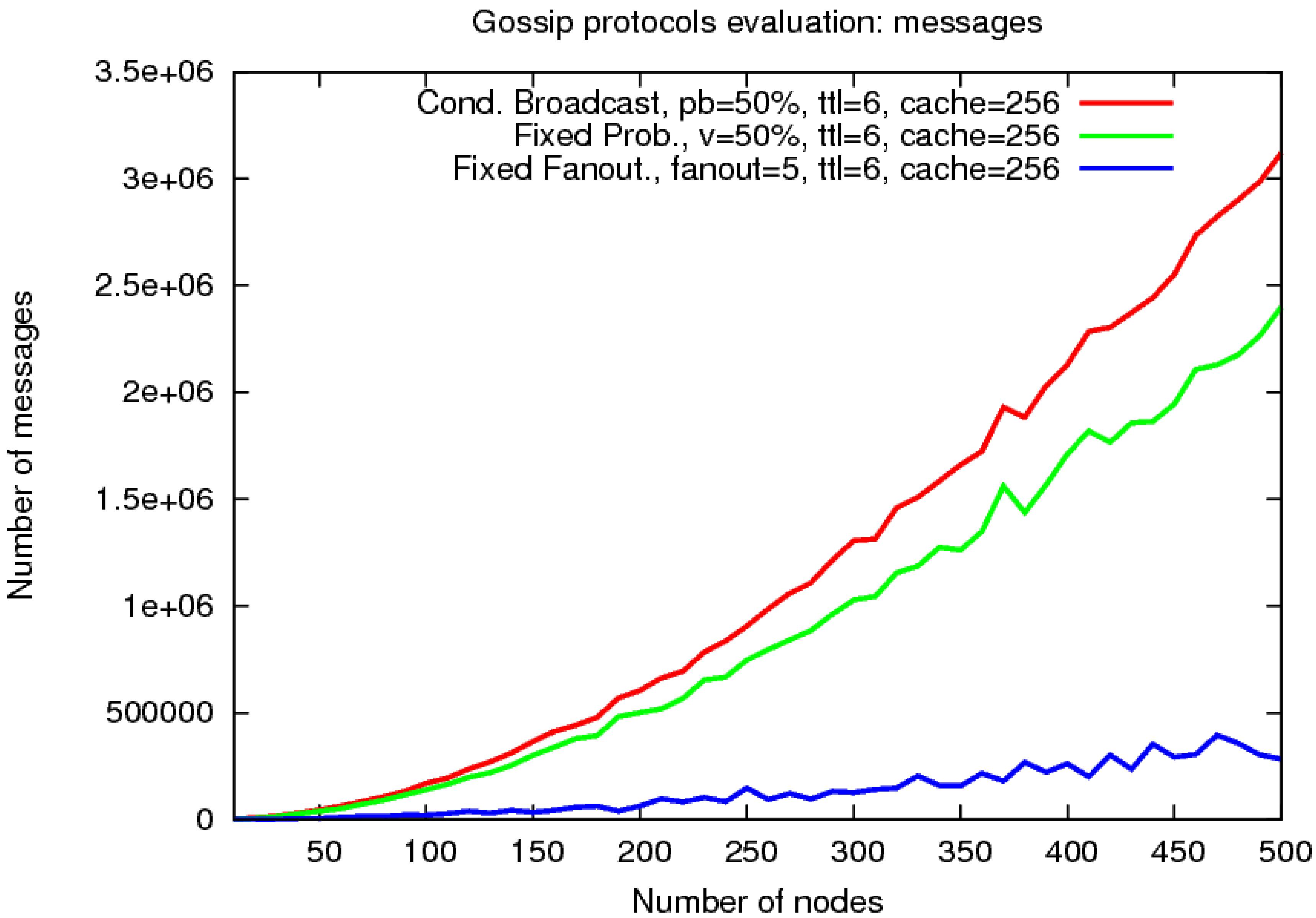
Evaluation: **coverage** rate (%)



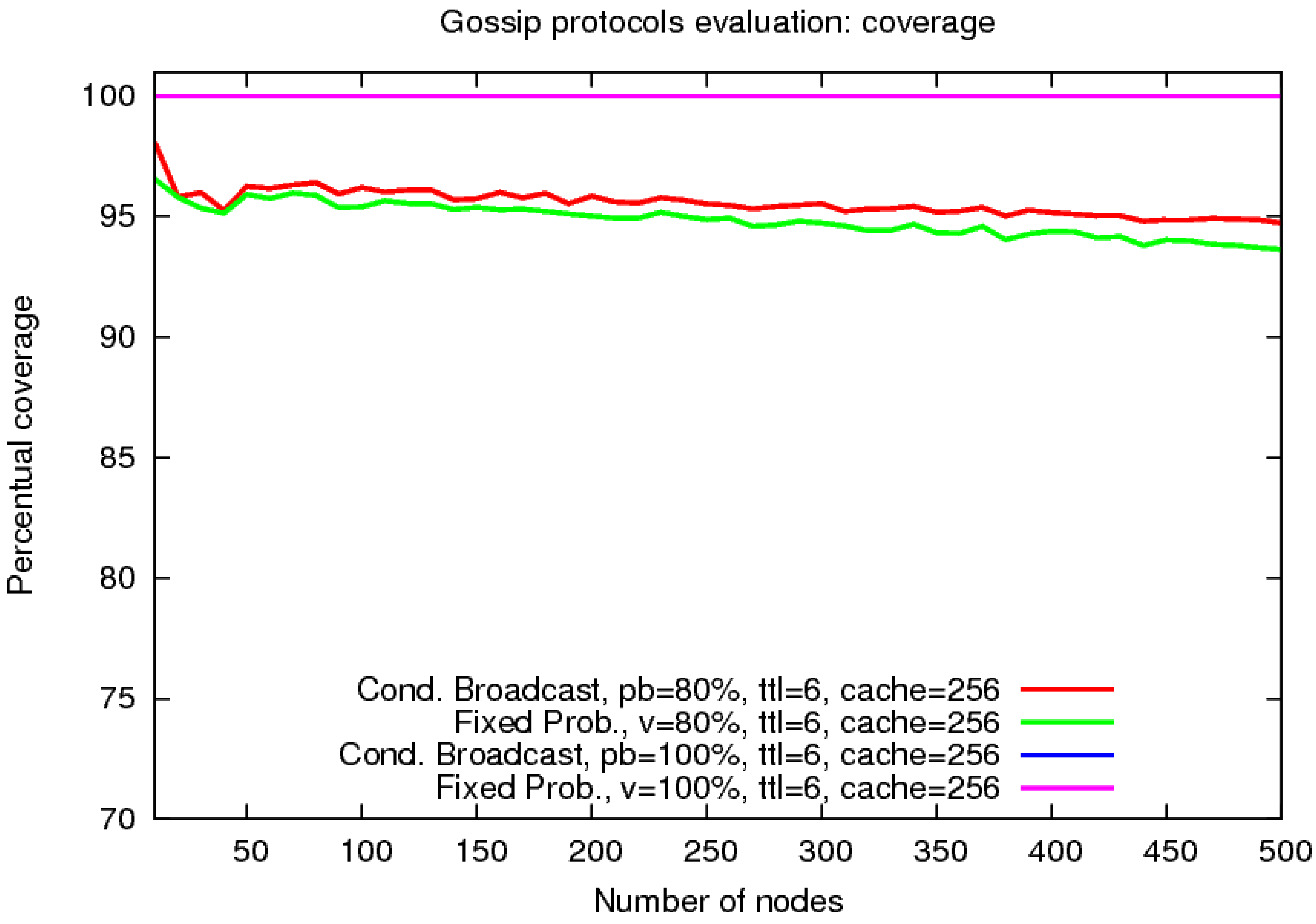
Evaluation: number of **hops**



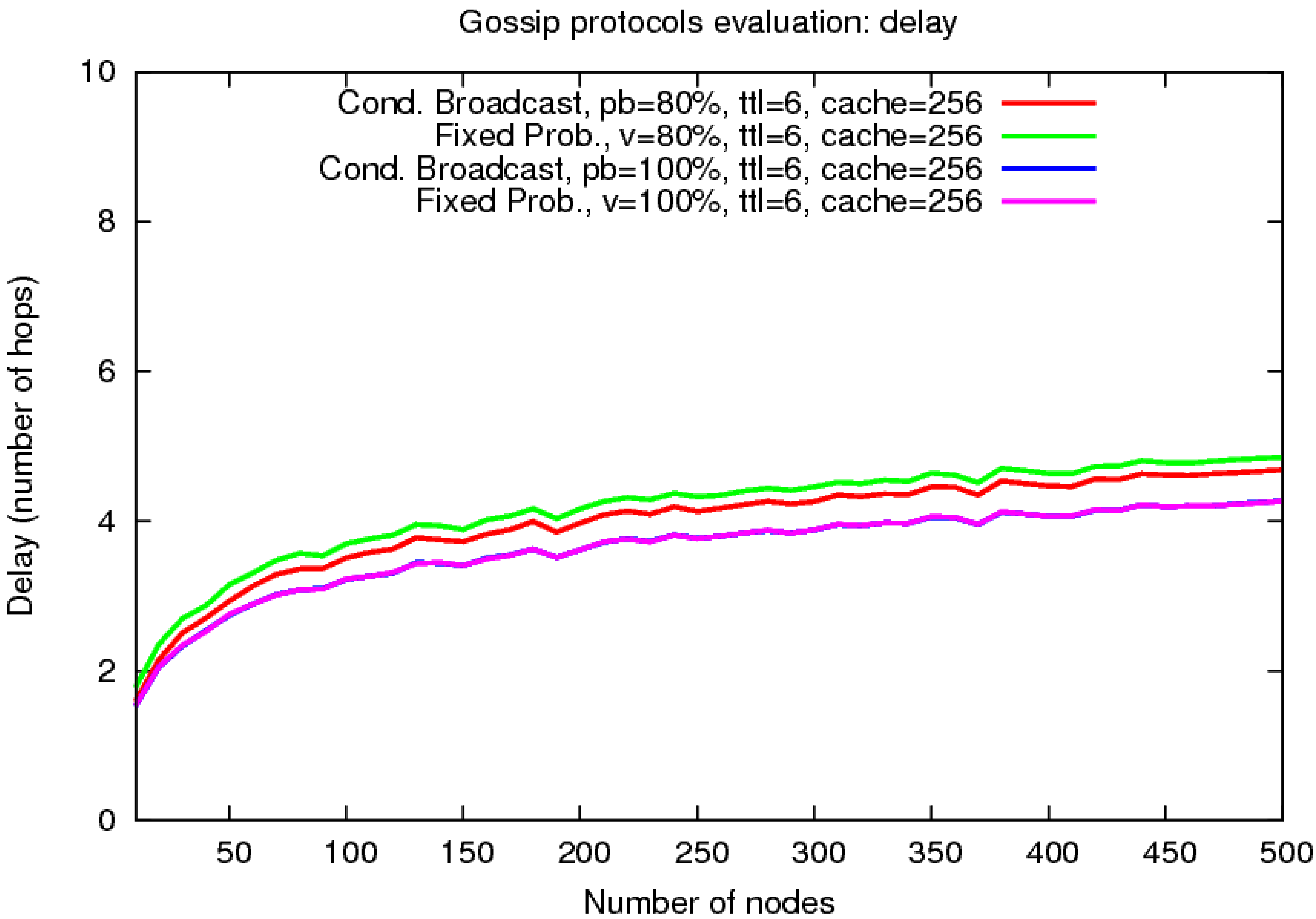
Evaluation: total number of **messages**



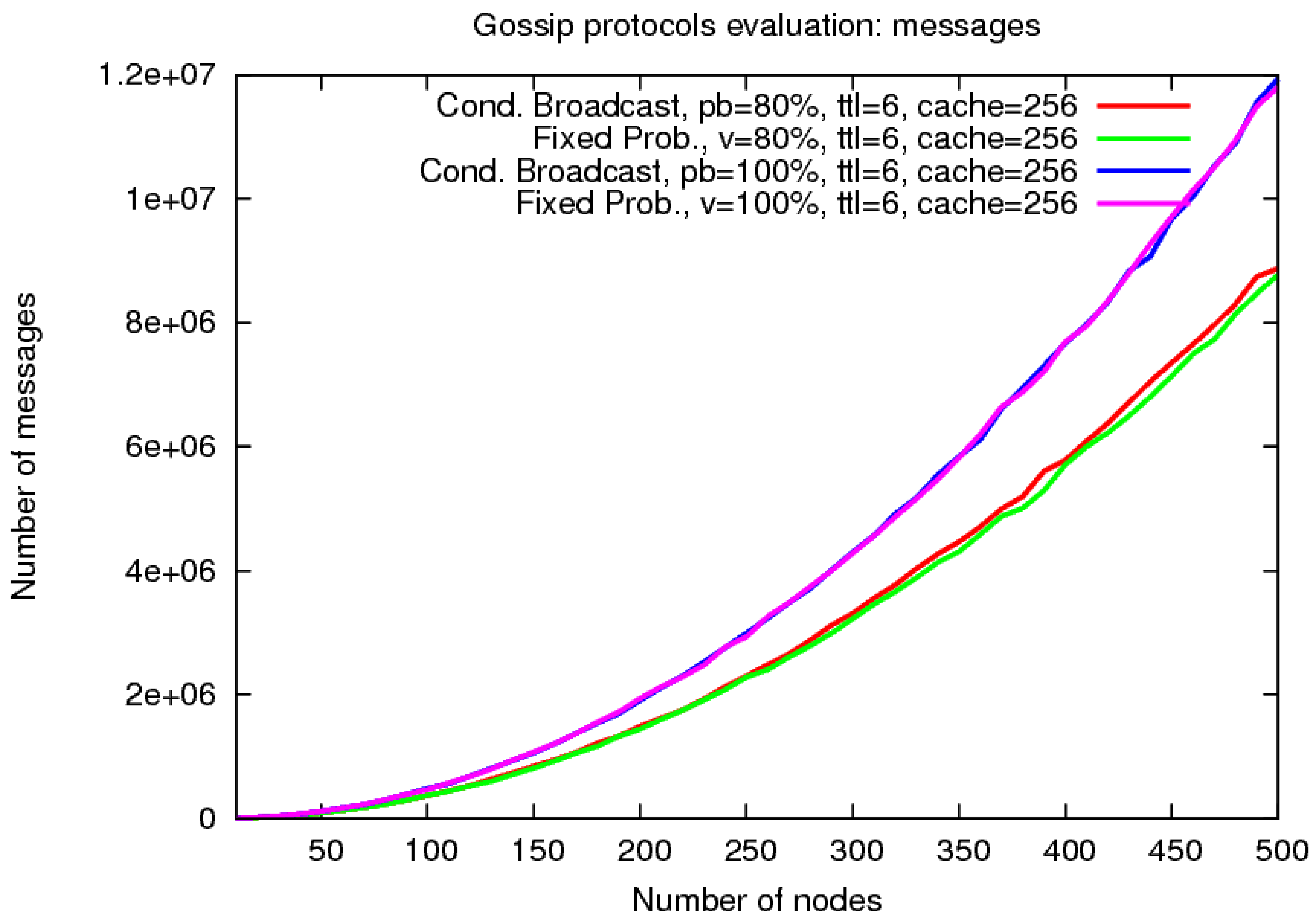
Evaluation: **coverage** rate (%)



Evaluation: number of **hops**



Evaluation: total number of **messages**



Conclusions and **Future work**

- The **low diameter** of scale-free networks is **very good** for fast data dissemination
- Common gossip protocols are **unable** to disseminate the whole event trace and their **overhead is very high**
- Simple mechanisms such as **caching** of packets, **ttl** and **protocols tweaking** are quite ineffective or with limited impact on performances (*e.g. # of routed messages*)
- **Smarter protocols** are necessary:
  - push / pull approaches for data dissemination
  - adaptive protocols and behaviors
  - more information shared among network nodes

Multiplayer Online Games over Scale-Free Networks: a Viable Solution?