Modeling, prediction, designing for LiFTinG

Semester project presentation

Supervised by: Maxime Monod, Prof. Rachid Guerraoui

Utkarsh Upadhyay

September 26, 2010

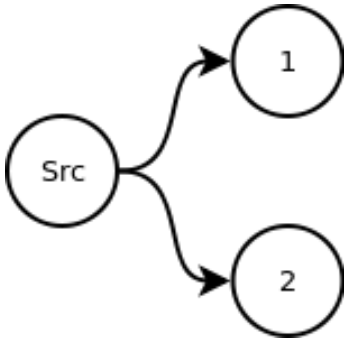
The main problems

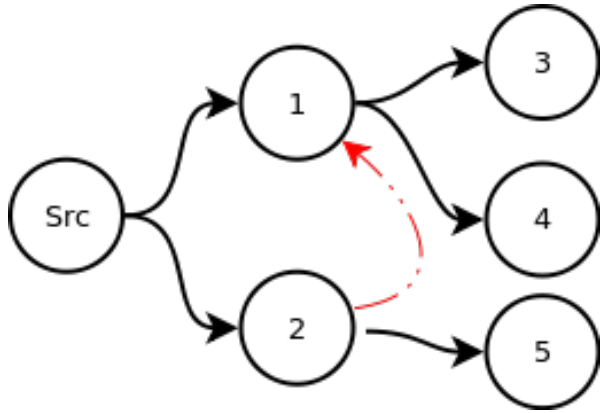
Objectives

- Modeling Gossip for outcast nodes
- Designing an efficient score management system

Outline

Epidemic data dissemination

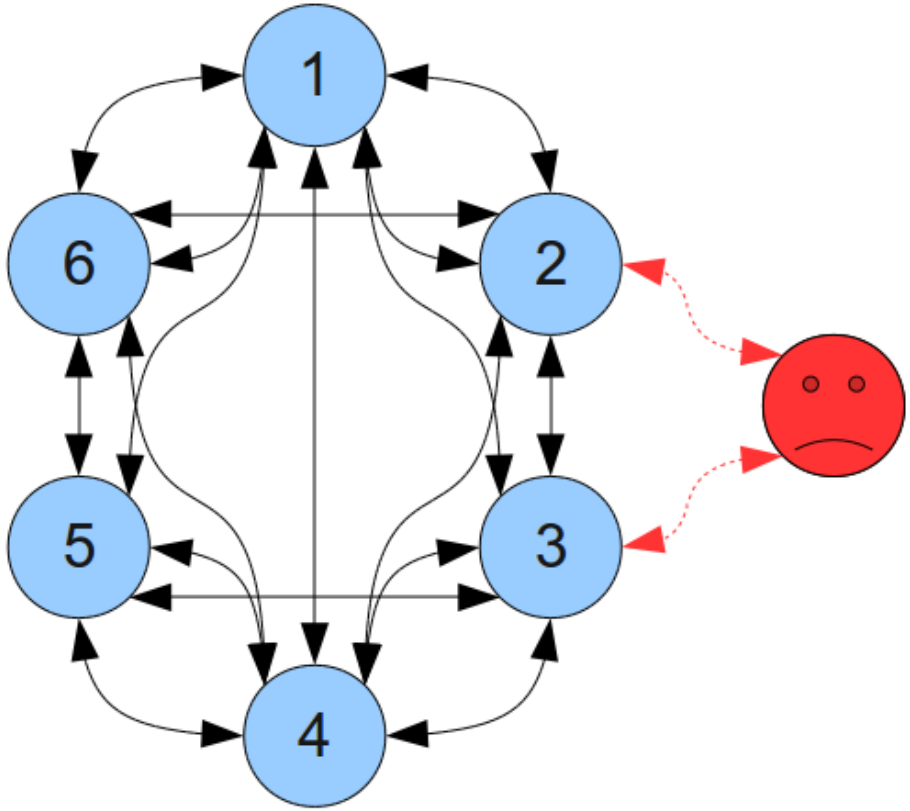




Gossiping with $f = 2$

- Push-pull
- Infect-and-die
- Random
- Packet drops?

Who is an outcast?



One outcast

- Nodes *almost* no-one gossips with

The main question

- Is 100% reception necessary? File sharing or Live streaming
- What about FEC-ARQ? $x\%$ $\Longrightarrow$ 99% reception

The problem

How many nodes should outcast a node so that it receives less than $x\%$ of data?

- Spread of revoke messages
- Better score management system?
- But why a model, why not just simulate it?
  - Theoretical guarantees (at least on mean values)
  - Faster
  - Insights

The main question

- Is 100% reception necessary? File sharing or Live streaming
- What about FEC-ARQ? x% $\Longrightarrow$ 99% reception

The problem

How many nodes should outcast a node so that it receives less than x% of data?

- Spread of revoke messages
- Better score management system?
- But why a model, why not just simulate it?
  - Theoretical guarantees (at least on mean values)
  - Faster
  - Insights

The main question

- Is 100% reception necessary? File sharing or Live streaming
- What about FEC-ARQ? x% $\Longrightarrow$ 99% reception

The problem

How many nodes should outcast a node so that it receives less than x% of data?

- Spread of revoke messages
- Better score management system?
- But why a model, why not just simulate it?
  - Theoretical guarantees (at least on mean values)
  - Faster
  - Insights

The main question

- Is 100% reception necessary? File sharing or Live streaming
- What about FEC-ARQ? x% $\Longrightarrow$ 99% reception

The problem

How many nodes should outcast a node so that it receives less than x% of data?

- Spread of revoke messages
- Better score management system?
- But why a model, why not just simulate it?
  - Theoretical guarantees (at least on mean values)
  - Faster
  - Insights

The main question

- Is 100% reception necessary? File sharing or Live streaming
- What about FEC-ARQ? x% $\Longrightarrow$ 99% reception

The problem

How many nodes should outcast a node so that it receives less than x% of data?

- Spread of revoke messages
- Better score management system?
- But why a model, why not just simulate it?
  - Theoretical guarantees (at least on mean values)
  - Faster
  - Insights

The main question

- Is 100% reception necessary? File sharing or Live streaming
- What about FEC-ARQ? x% $\Longrightarrow$ 99% reception

The problem

How many nodes should outcast a node so that it receives less than x% of data?

- Spread of revoke messages
- Better score management system?
- But why a model, why not just simulate it?
  - Theoretical guarantees (at least on mean values)
  - Faster
  - Insights

Models

- **Erdös-Rényi model**: Good for connectivity study, exact numbers, show $f \approx \log_e N$ for atomic broadcasts, not ideal for behavior of one individual (outcast) with respect to a random graph
- **Macro-discrete Models:** Keep track of exact numbers in each state, intractable because of complex combinatorics
- **Epidemic model** : Good for approximations, averages, fractions

Models

- **Erdös-Rényi model**: Good for connectivity study, exact numbers, show $f \approx \log_e N$ for atomic broadcasts, not ideal for behavior of one individual (outcast) with respect to a random graph
- **Macro-discrete Models:** Keep track of exact numbers in each state, intractable because of complex combinatorics
- **Epidemic model** : Good for approximations, averages, fractions

Models

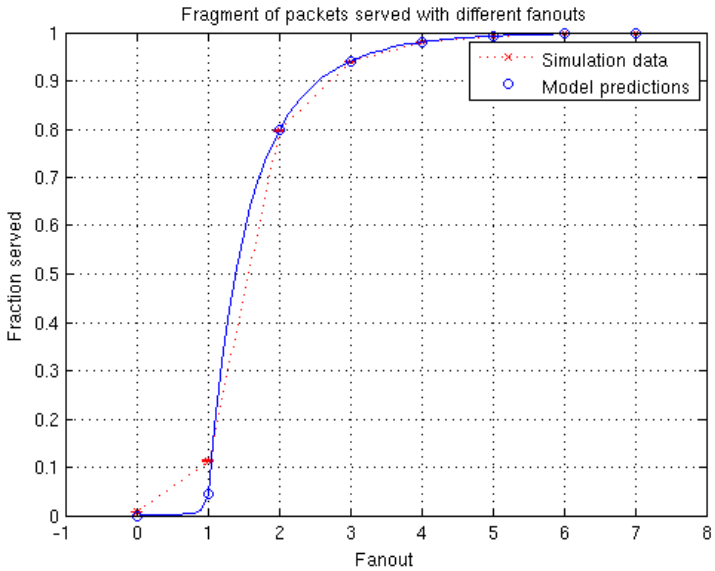
- **Erdös-Rényi model**: Good for connectivity study, exact numbers, show $f \approx \log_e N$ for atomic broadcasts, not ideal for behavior of one individual (outcast) with respect to a random graph
- **Macro-discrete Models:** Keep track of exact numbers in each state, intractable because of complex combinatorics
- **Epidemic model** : Good for approximations, averages, fractions

SIR model

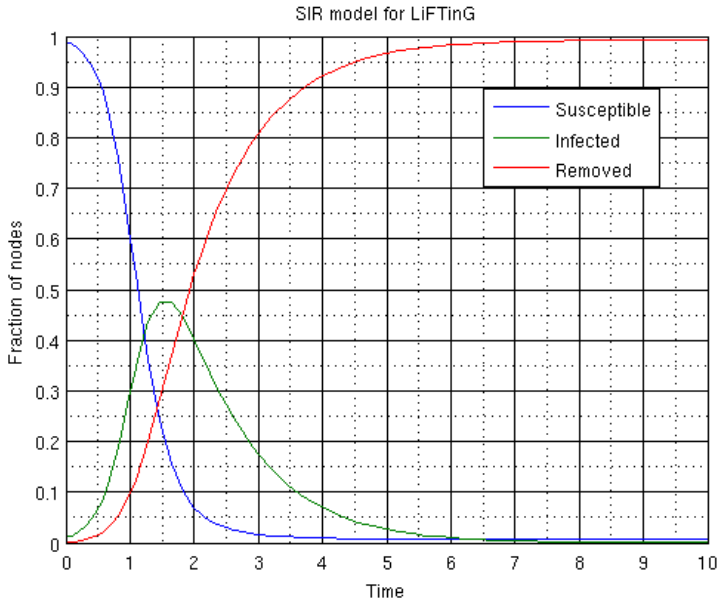
Development

- Basic model for epidemic spread
- Modeling individual outcasts using mean field theory
- Modeling a fraction of outcast by extending the basic model

Basic model



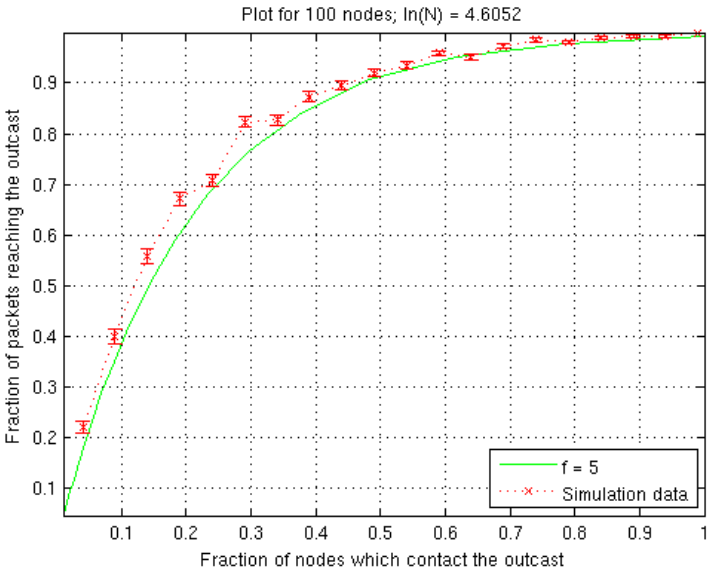
(a) Coverage with different f



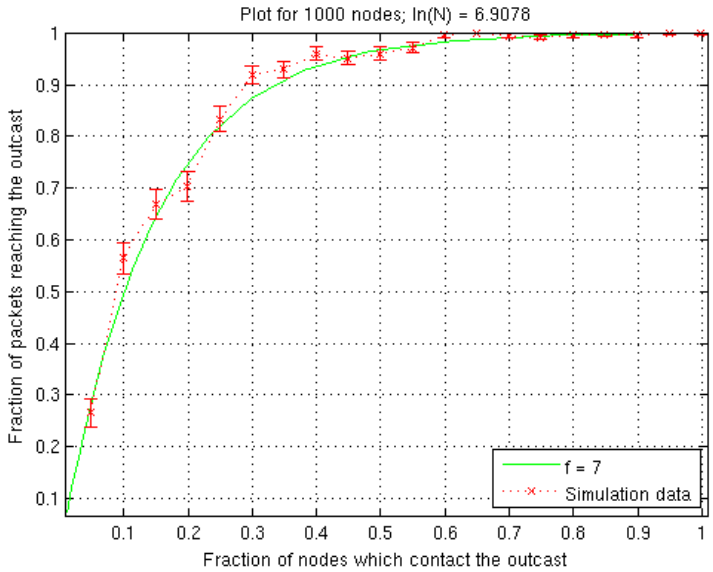
(b) Temporal Evolution

Predictions of the basic SIR model

One outcast



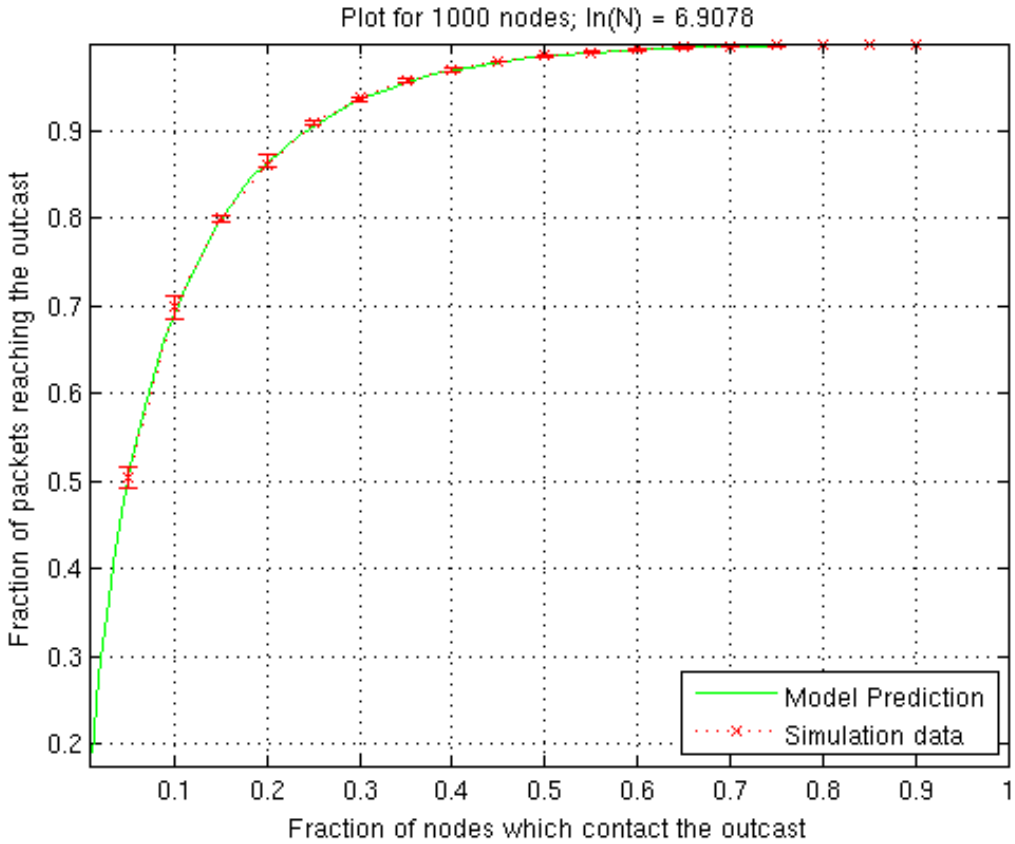
(a) 100 nodes, 1 outcast



(b) 1000 nodes, 1 outcast

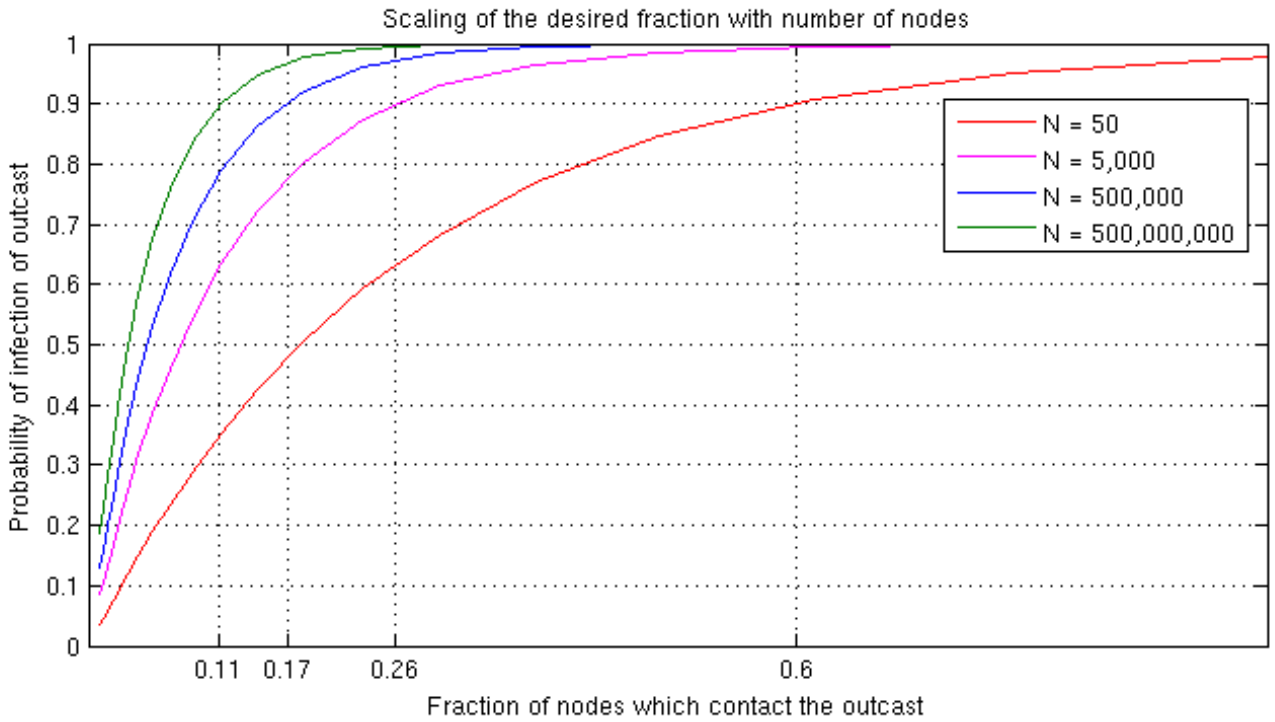
Verification of model predictions

More than one outcast



10% nodes are outcasts

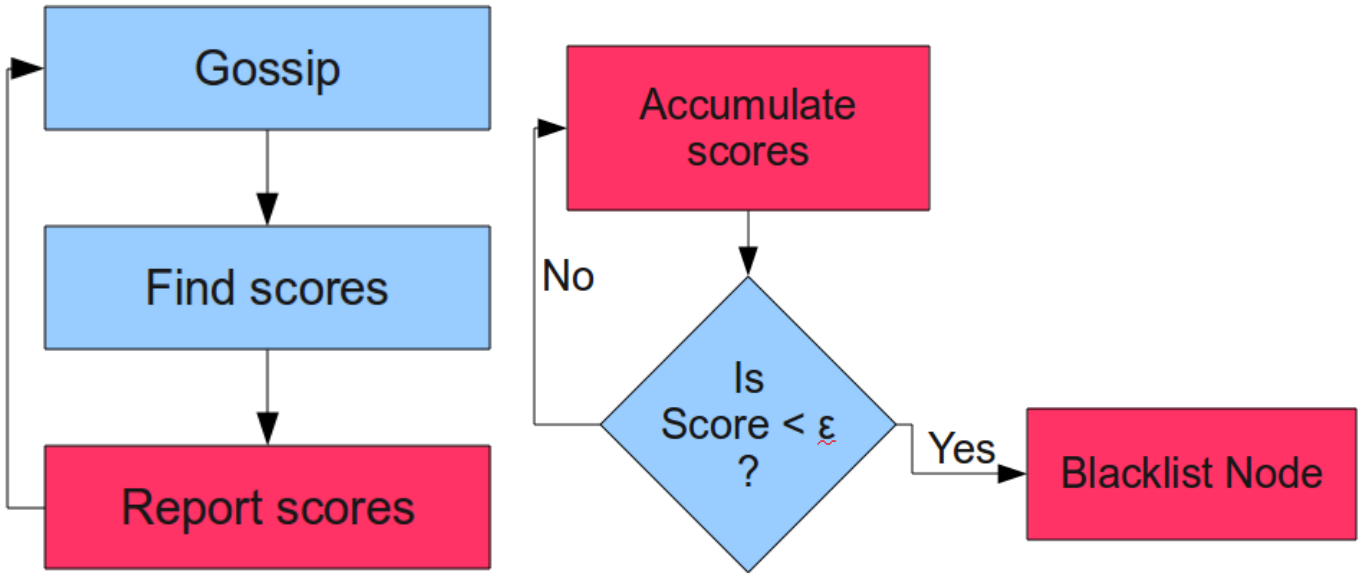
Does this scale?



With 1 outcast and different N

Outline

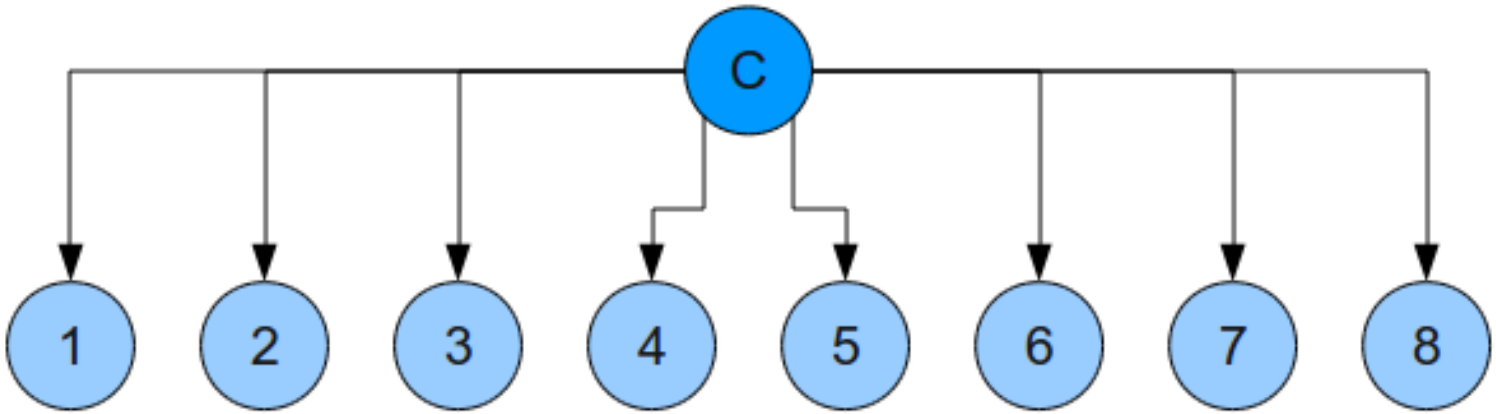
What is score management?



(a) On each reporter (b) For each blamed node

LiFTinG protocol

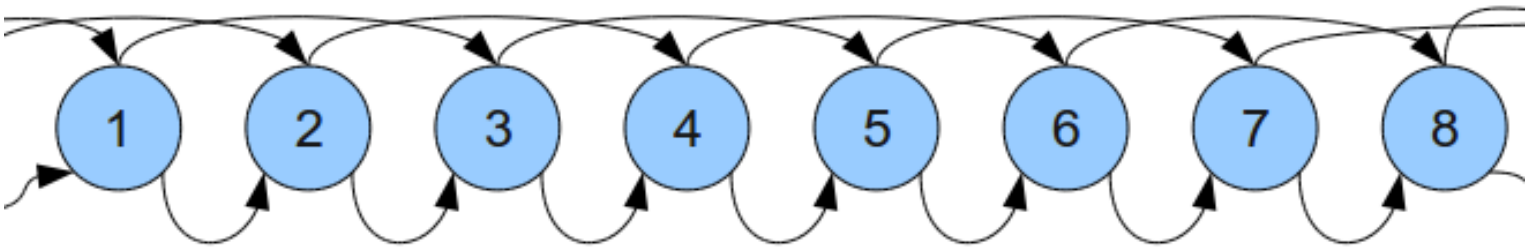
Central blaming entity



Central server managing all nodes in the system

- Scalability!

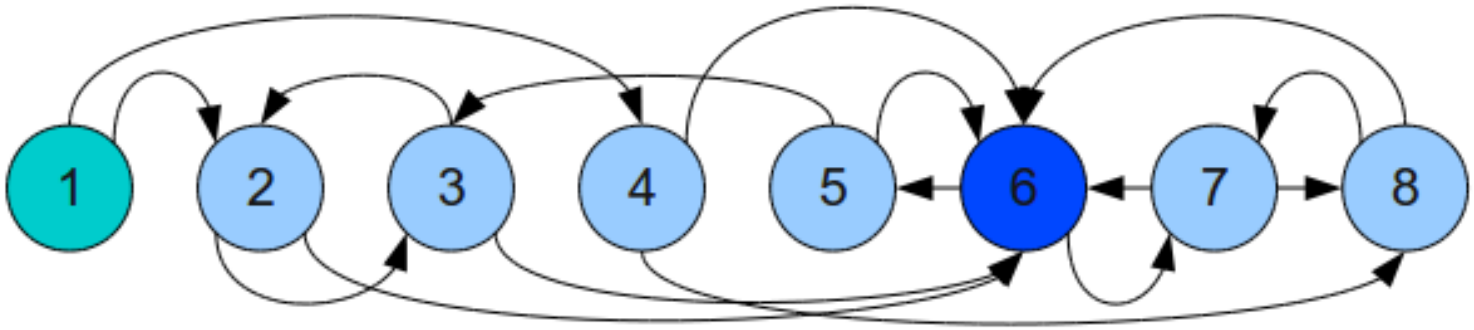
AVMON or AVCast



Hash based manager selection

- Structure!
- Complicated to implement

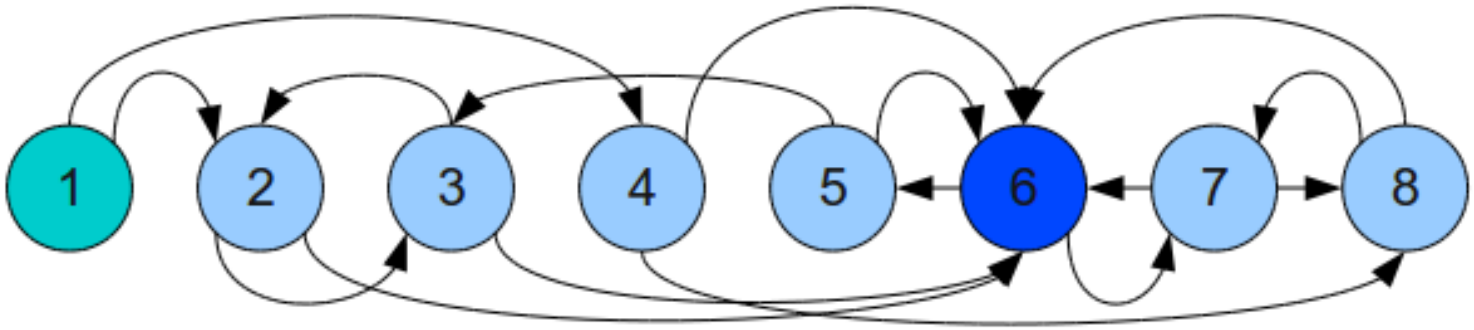
Random managers



Choose two nodes to manage at random

- Will each node be covered? $k = \log_e n + m \cdot c$
- How to report scores? Gossiping blames does not scale!

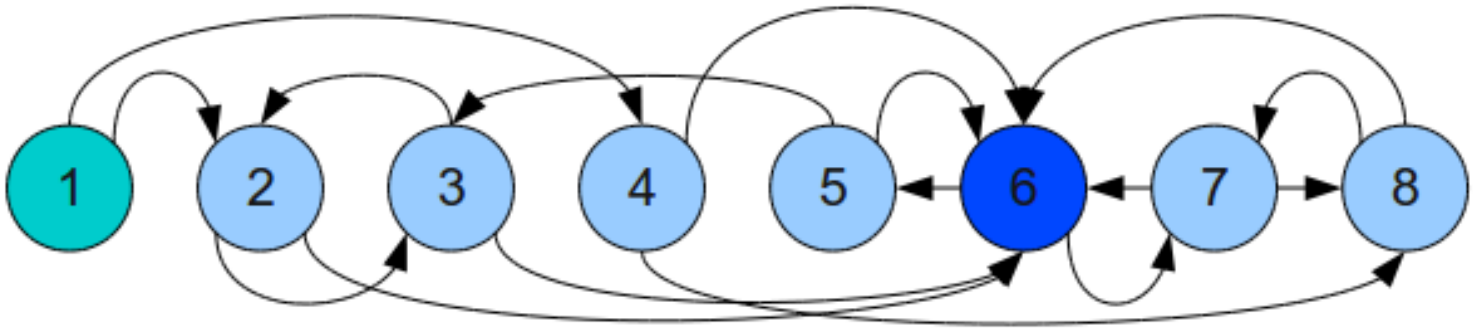
Random managers



Choose two nodes to manage at random

- Will each node be covered? $k = \log_e n + m \cdot c$
- How to report scores? Gossiping blames does not scale!

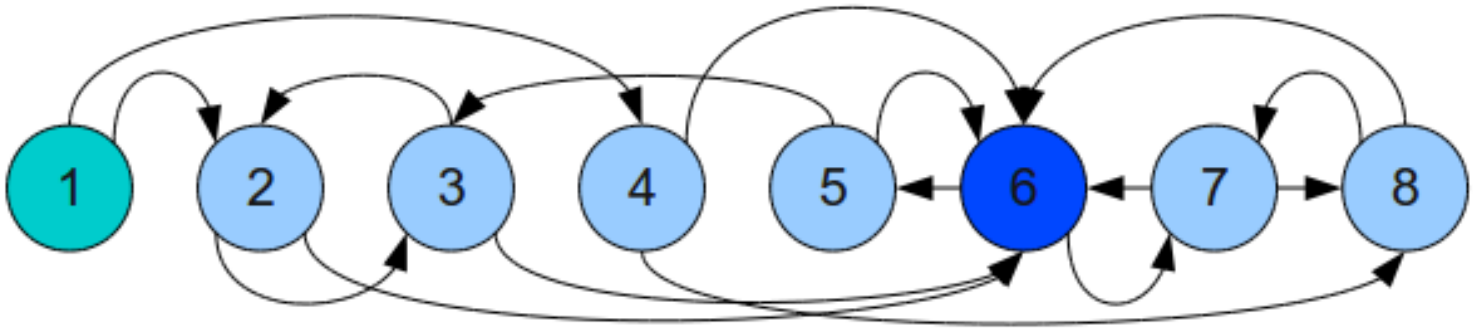
Random managers



Choose two nodes to manage at random

- Will each node be covered? $k = \log_e n + m \cdot c$
- How to report scores? Gossiping blames does not scale!

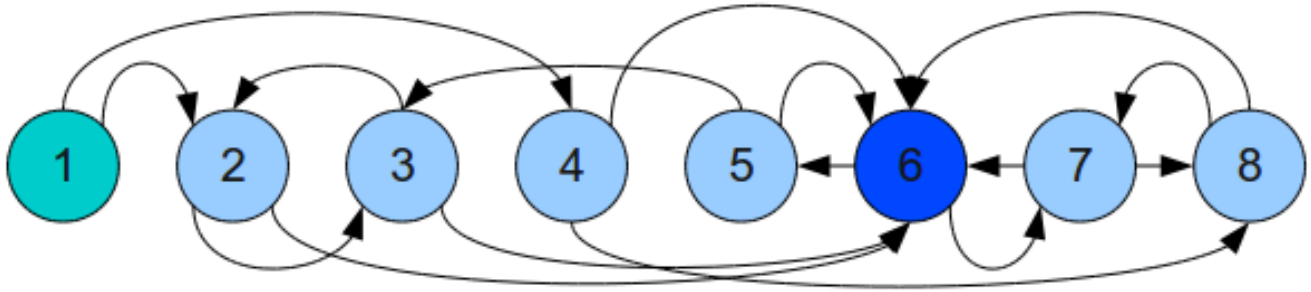
Random managers



Choose two nodes to manage at random

- Will each node be covered? $k = \log_e n + m \cdot c$
- How to report scores? Gossiping blames does not scale!

Caching addresses



Choose two nodes to manage at random

(a) For Node 3

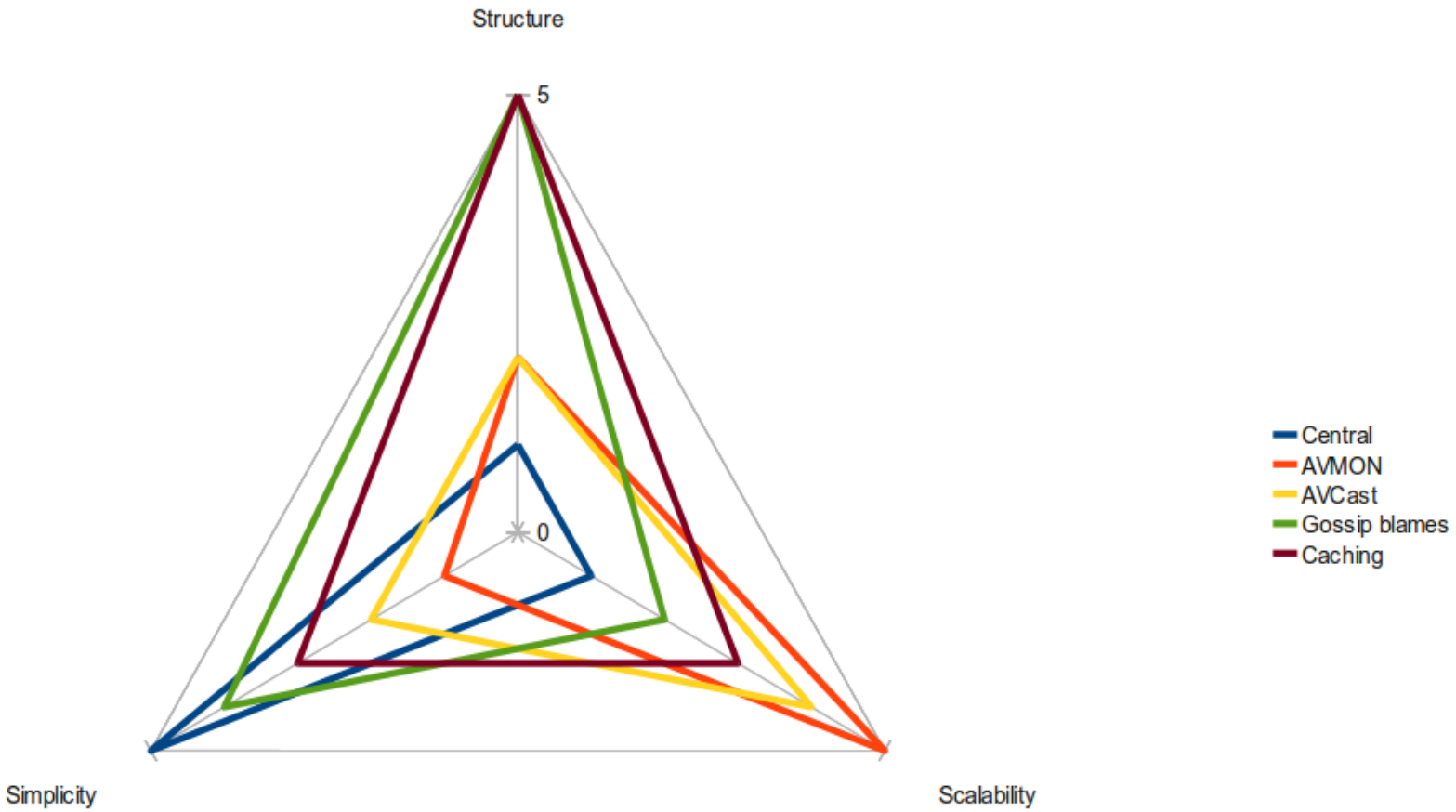
Of:	List
1	-
4	{1}
8	{4, 7}

(b) For Node 4

Of:	List
2	{1,3}
8	{4,7}
6	{2,3,4,5,7,8}

$O(\sqrt{N})$ managers stored

Deciding the metric

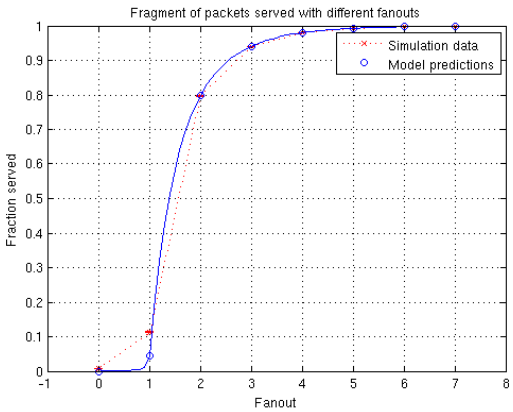


Comparison of different management systems

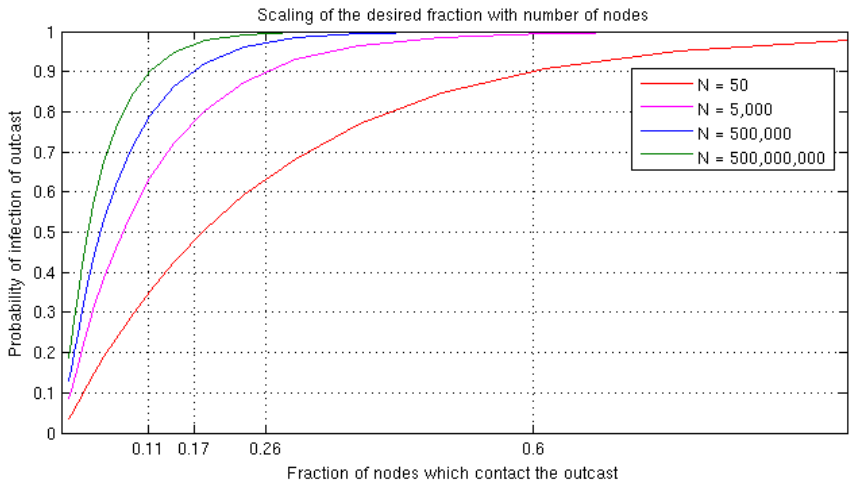
Outline

Take home message: Modeling Gossip

- Revoke message with fanout of 3 will reach 95% nodes: works for more than 500,000,000 nodes



(a) Frac. reached



(b) Needed revoking fraction

Putting together the results

Take home message: Modeling Gossip

- Turning perspective around: A new-coming node will need to contact a small fraction of nodes to get 90% data:
  - Useful for probing sensor networks?
  - Hierarchal gossiping?
- Very general model, can model:
  - Heterogeneous gossiping behaviors
  - Mean value of jitter, temporal evolution?
  - Variance, second order analysis?

Take home messages: Score management

- AVMON is too strong and structured
- Properties of LiFTinG which should be used:
  - Knowledge of own managers is not necessary
  - No positive scores, no false negative feedback
- STRUCTURE v/s SCALABILITY: can do better?

Thank you

Questions?

Utkarsh Upadhyay
utkarsh.upadhyay@epfl.ch