On the broadcast of segmented messages in dynamic networks

Rajib Ranjan Maiti (rajib.maiti@gmail.com)

Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, India

In collaboration with:

Marcin Bodych, Tyll Krueger (Wroclaw University of Technology)

Sandipan Sikdar, Biswajit Paria, Niloy Ganguly, Animesh Mukherjee (IIT Kharagpur)

Outline

1) Problem definition
2) Setup
3) Algorithms of Message Transfer
4) Deeper analysis of the algorithms
5) Conclusions

1) Problem definition
2) Setup
3) Algorithms of Message Transfer
4) Deeper analysis of the algorithms
5) Conclusions

Problem Definition

- Consider a dynamic and a distributed system of agents
- Connections between any pair of agents are intermittent
- Information dissemination occurs through epidemic protocols due to its dynamic nature
- A piece of information can be passed on when two nodes meet (i.e., establish communication link)

Problem Definition

- Consider a dynamic and a distributed system of agents
- Connections between any pair of agents are intermittent
- Information dissemination occurs through epidemic protocols due to its dynamic nature
- A piece of information can be passed on when two nodes meet (i.e., establish communication link)
- But what if the information is too large to be sent over in a single contact?

Split the Message

- A simple solution is to split the message into packets
- Hence, during a communication opportunity transfer only a part of the message
- But, how will message splitting influence the speed of dissemination?
- Will the underlying topology have any effect?
- Will it be possible to control resource wastage by devising efficient message spreading policy, especially, in a distributed environment?

Split the Message

- A simple solution is to split the message into packets
- Hence, during a communication opportunity transfer only a part of the message
- But, how will message splitting influence the speed of dissemination?
- Will the underlying topology have any effect?
- Will it be possible to control resource wastage by devising efficient message spreading policy, especially, in a distributed environment?
- We consider the case of broadcast and try to answer these questions
- We also propose an efficient broadcast algorithm

1) Problem definition
2) Setup
3) Algorithms of Message Transfer
4) Deeper analysis of the algorithms
5) Conclusions

Agent and Message Configuration

- Agent configuration
  - A graph G=(V,E) represents a dynamic network
    - A node in V ➔ an agent, and
    - An edge in E ➔ a possible link between a pair of agents
    - In case, a link is not available at a particular time due to dynamic nature of the network, the link is considered as existing but inactive at that time

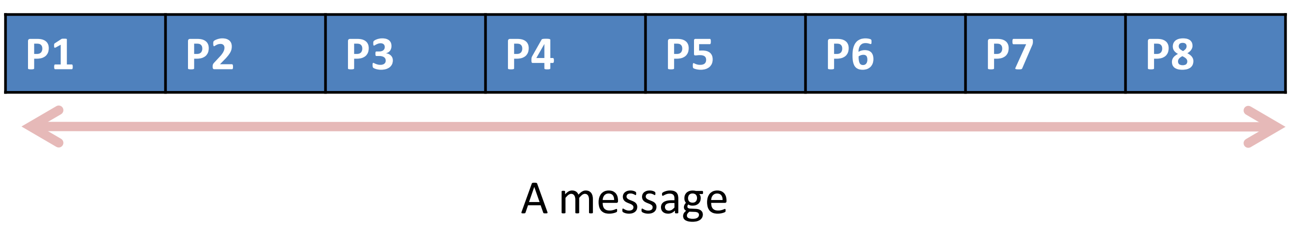
Agent and Message Configuration

- Agent configuration
  - A graph G=(V,E) represents a dynamic network
    - A node in V ➔ an agent, and
    - An edge in E ➔ a possible link between a pair of agents
    - In case, a link is not available at a particular time due to dynamic nature of the network, the link is considered as existing but inactive at that time
- Message configuration
  - A message is split into a set of m packets and the packets are grouped into s segments, s.t., *m mod s = 0*

A message

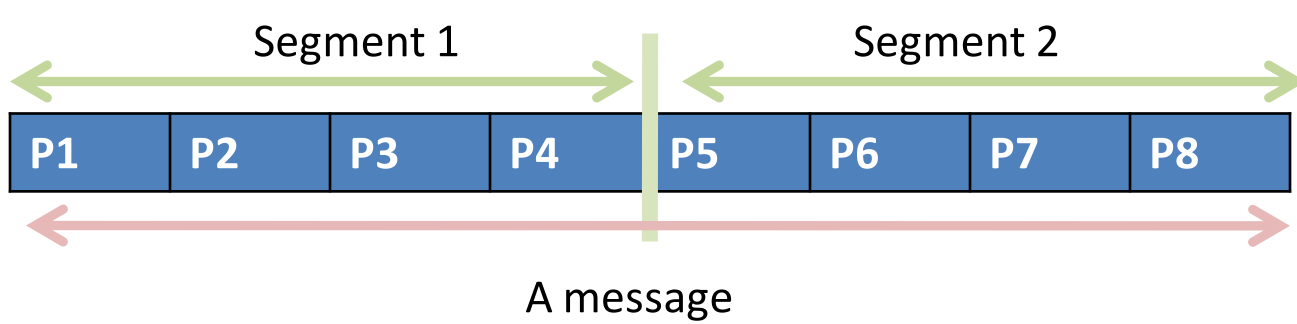
Agent and Message Configuration

- Agent configuration
  - A graph G=(V,E) represents a dynamic network
    - A node in V ➔ an agent, and
    - An edge in E ➔ a possible link between a pair of agents
    - In case, a link is not available at a particular time due to dynamic nature of the network, the link is considered as existing but inactive at that time
- Message configuration
  - A message is split into a set of m packets and the packets are grouped into s segments, s.t., *$m \bmod s = 0$*



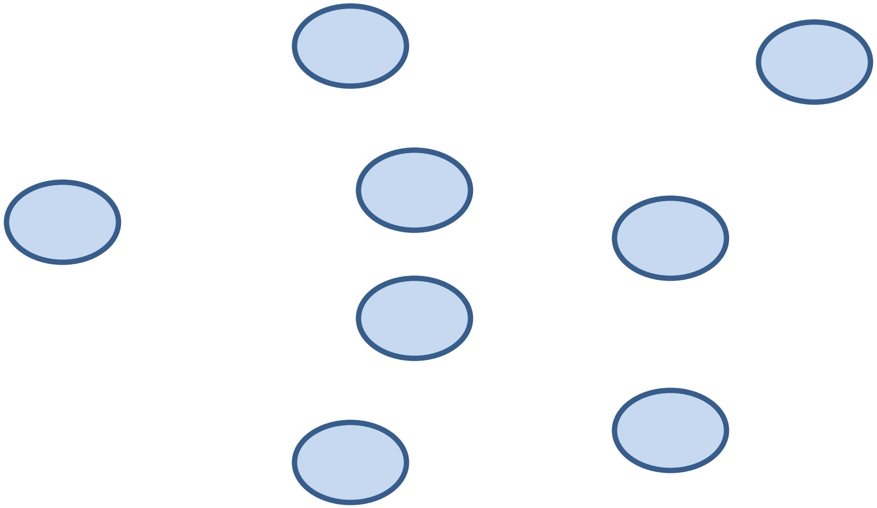
Agent and Message Configuration

- Agent configuration
  - A graph G=(V,E) represents a dynamic network
    - A node in V ➔ an agent, and
    - An edge in E ➔ a possible link between a pair of agents
    - In case a link is not available at a particular time due to dynamic nature of the network, the link can be considered as existing but inactive
- Message configuration
  - A message is split into a set of m packets and the packets are grouped into s segments, s.t., $m \bmod s = 0$



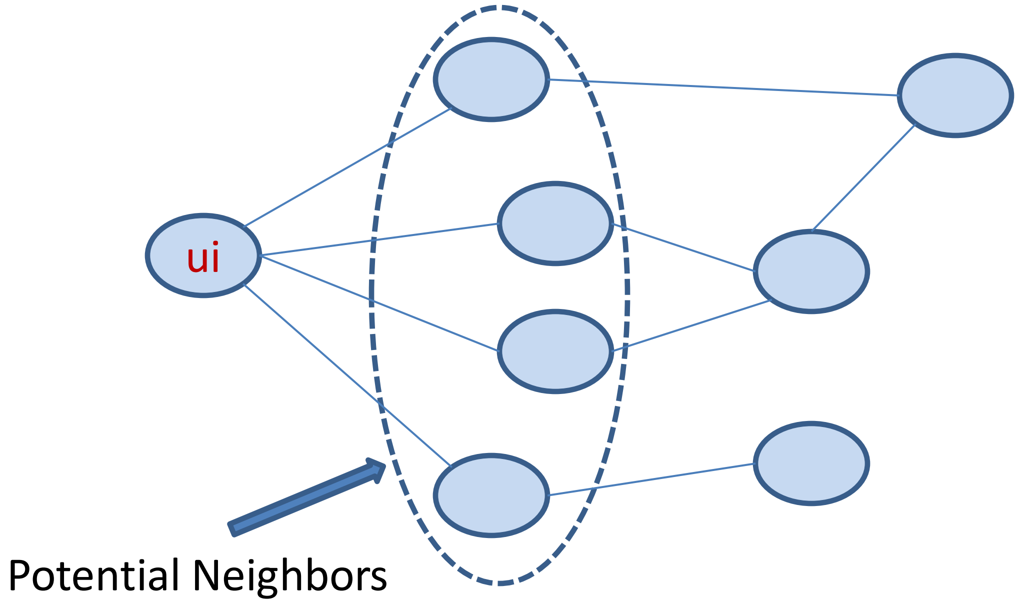
Dynamic Network Topology

- Topology specifies potential neighbors for each node
- At each time step, a node tries to connect with one of the neighbors selected randomly



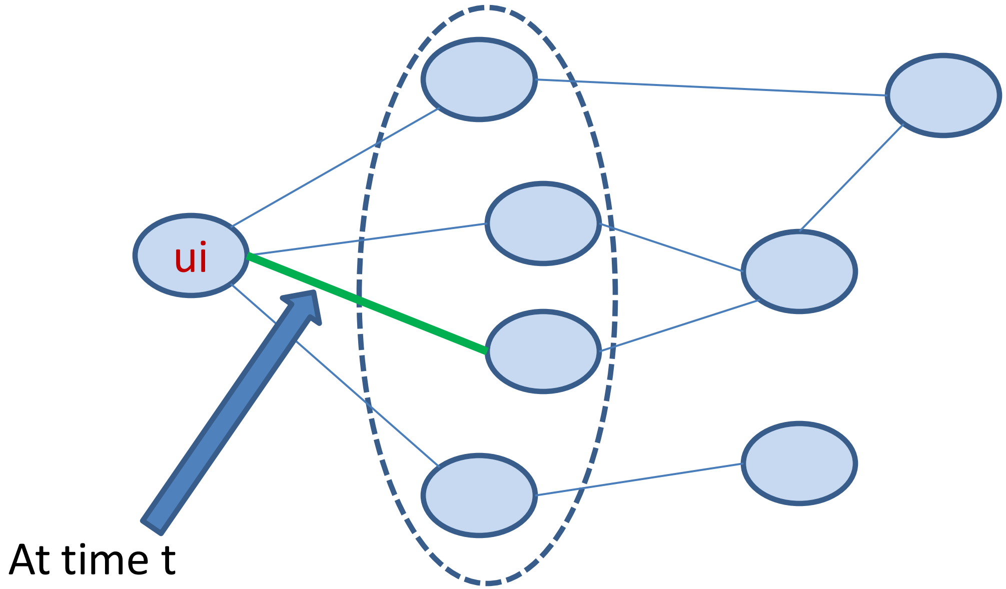
Dynamic Network Topology

- Topology specifies potential neighbors for each node
- At each time step, a node tries to connect with one of the neighbors selected randomly



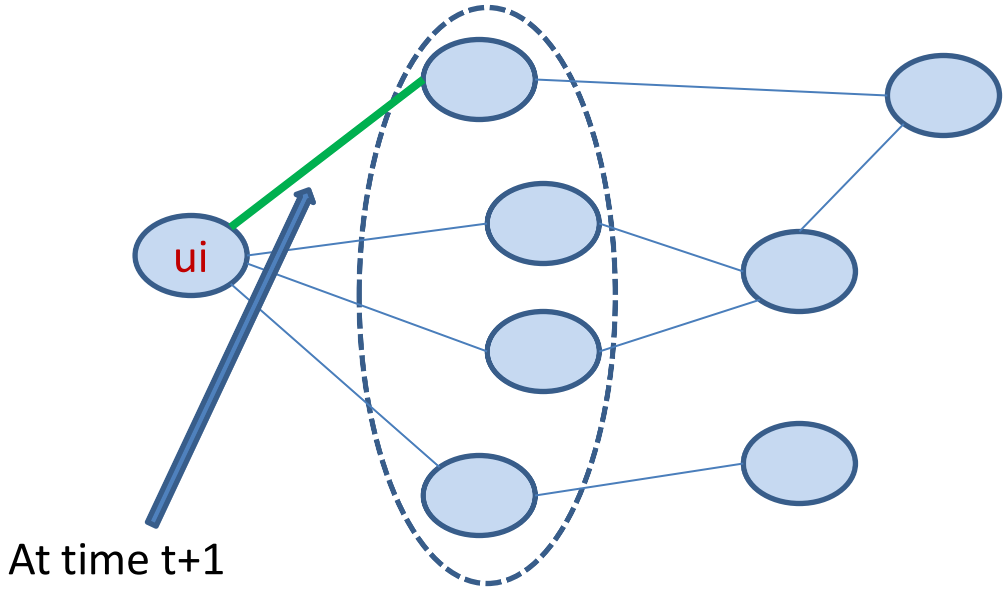
Dynamic Network Topology

- Topology specifies potential neighbors for each node
- At each time step, a node tries to connect with one of the neighbors selected randomly



Dynamic Network Topology

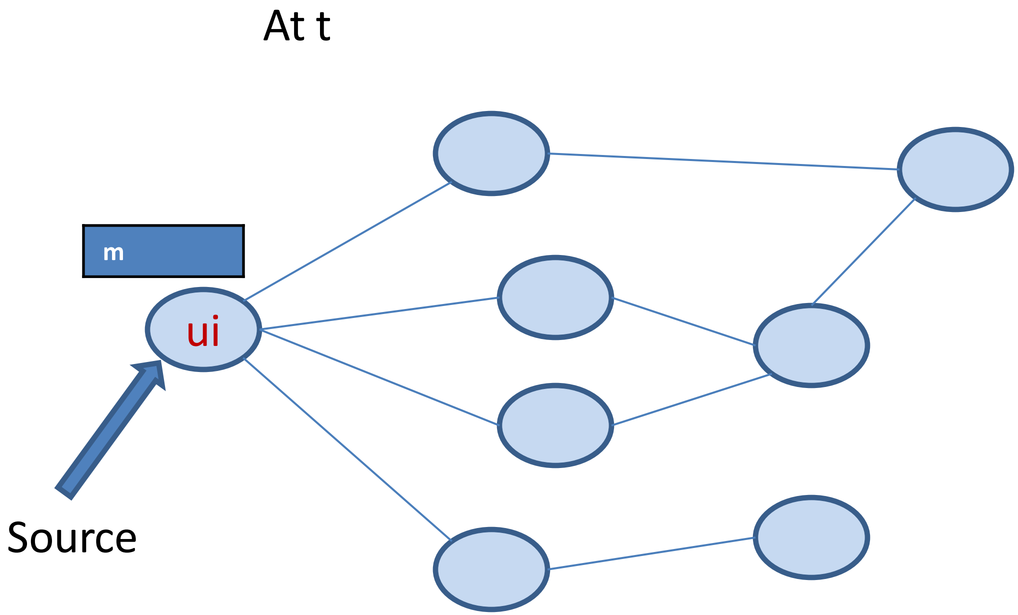
- Topology specifies potential neighbors for each node
- At each time step, a node tries to connect with one of the neighbors selected randomly



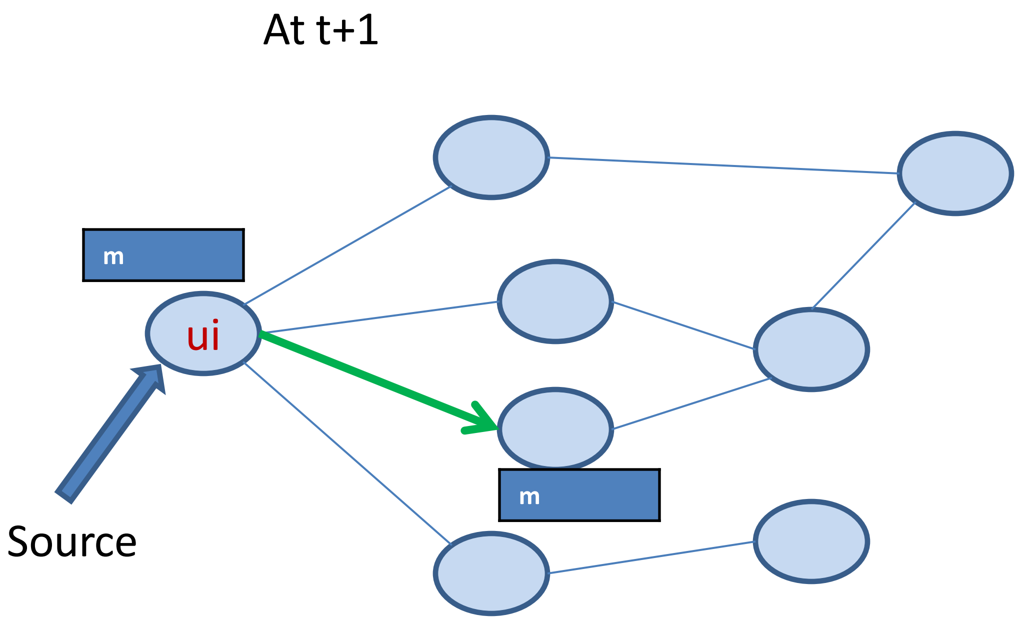
Transfer Protocols

- Consider a node as a sender when it has all the packets of a message
  - Initially, source is the only sender having all the packets
- And, the other nodes as non-senders that has no or not all packets of the message
  - Hence, non-senders won't start forwarding any packet
  - They do forward only when they become sender
- Two basic forwarding techniques
  - Push technique
  - Pull technique

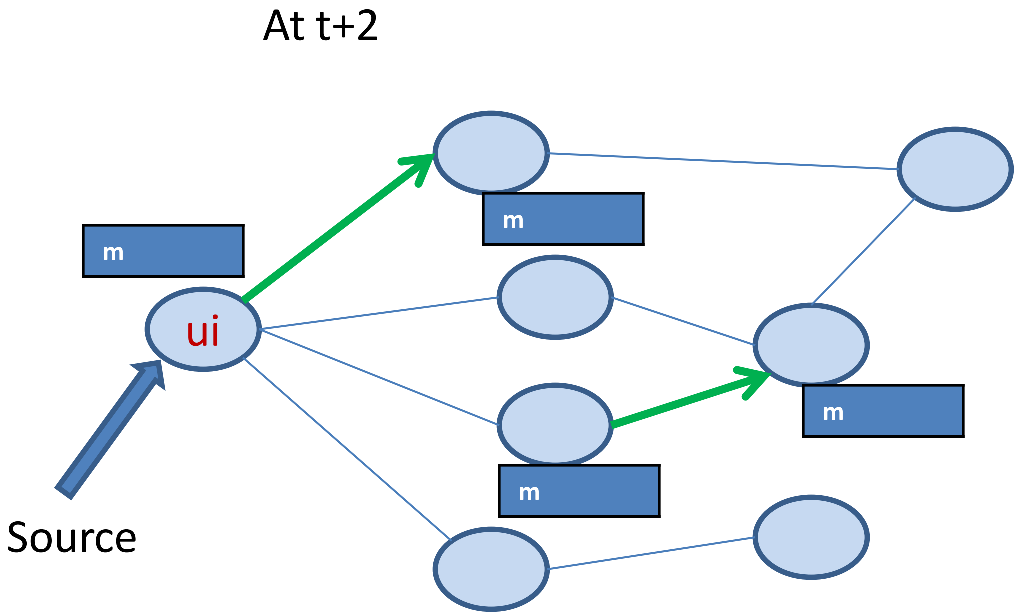
Push Technique



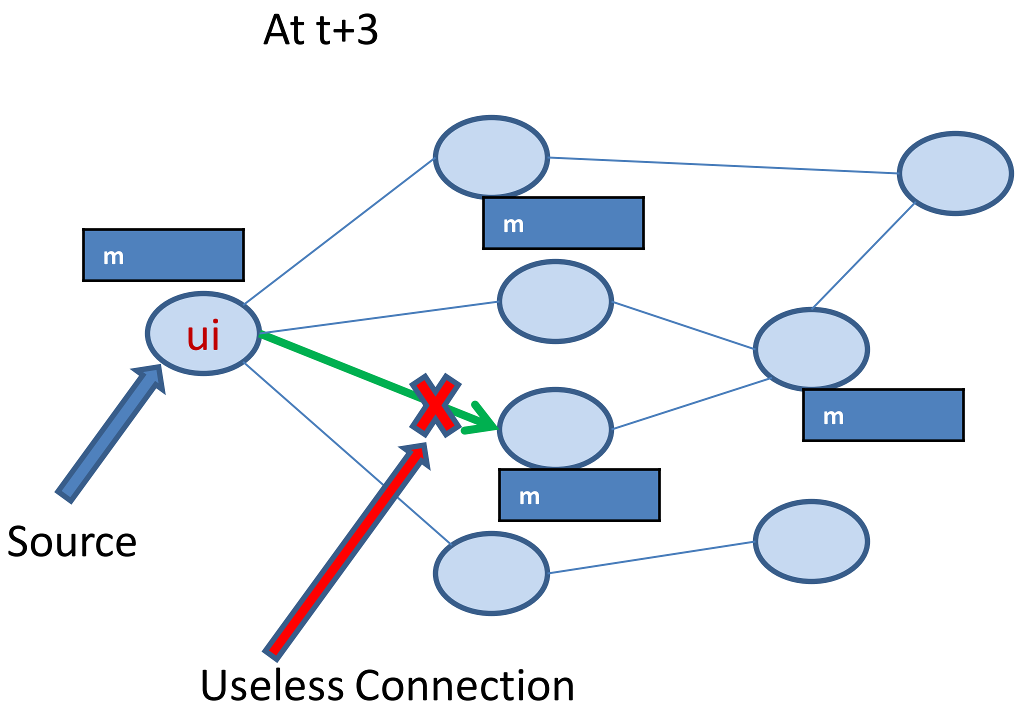
Push Technique



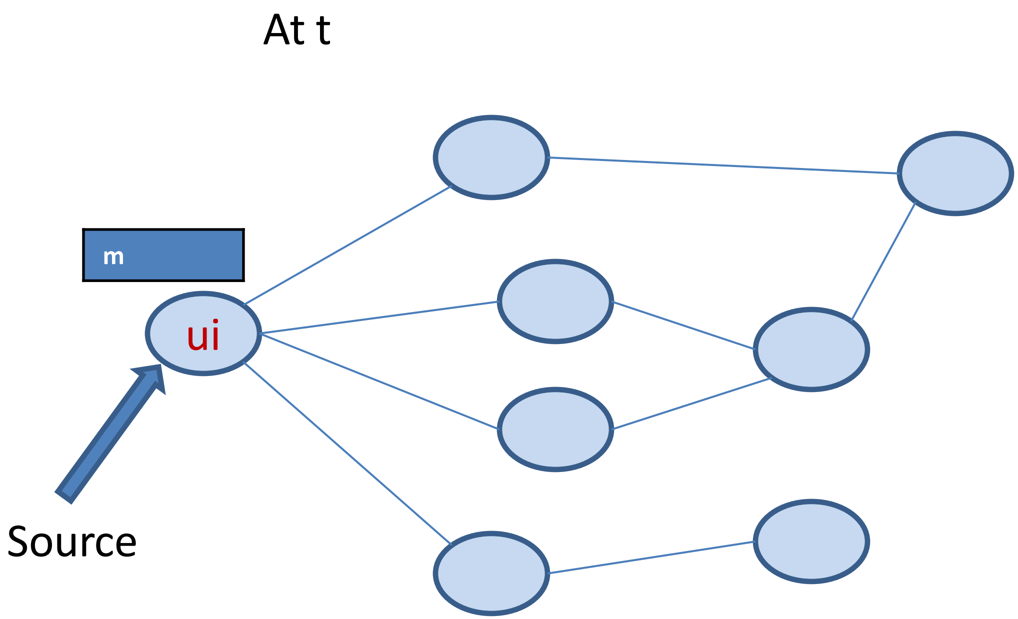
Push Technique



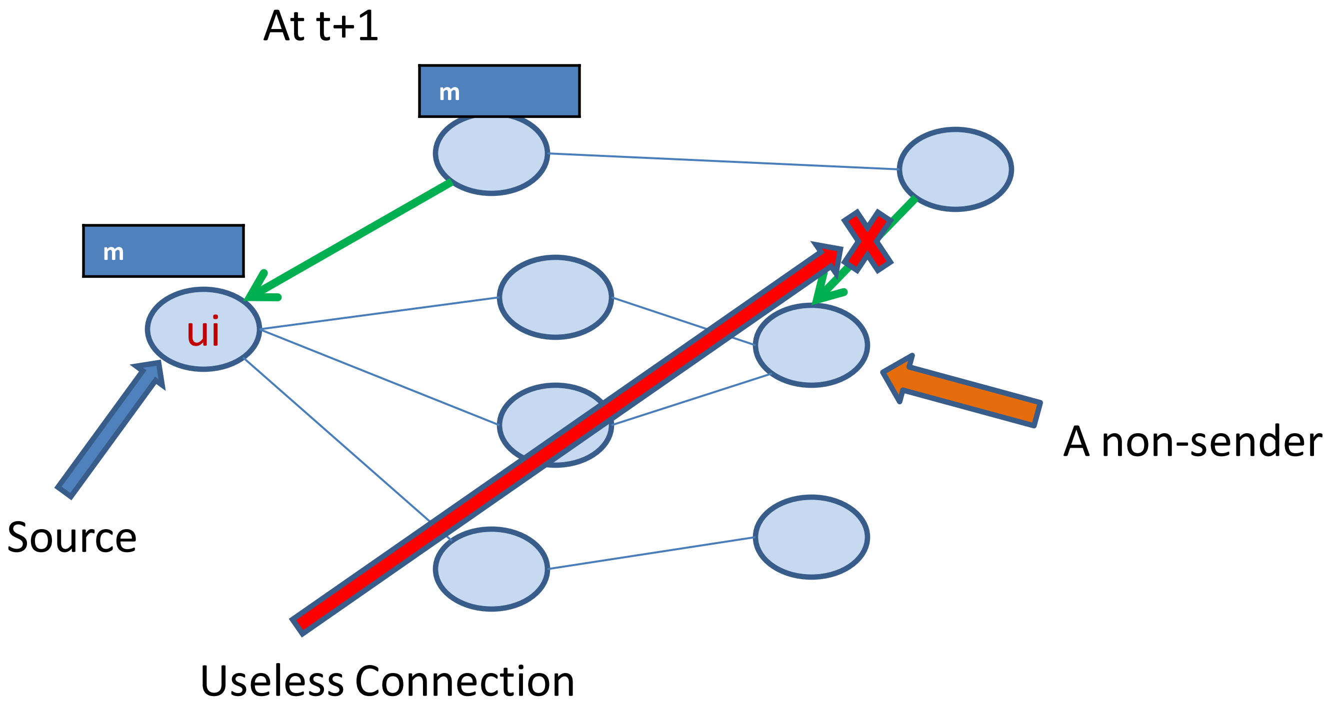
Push Technique



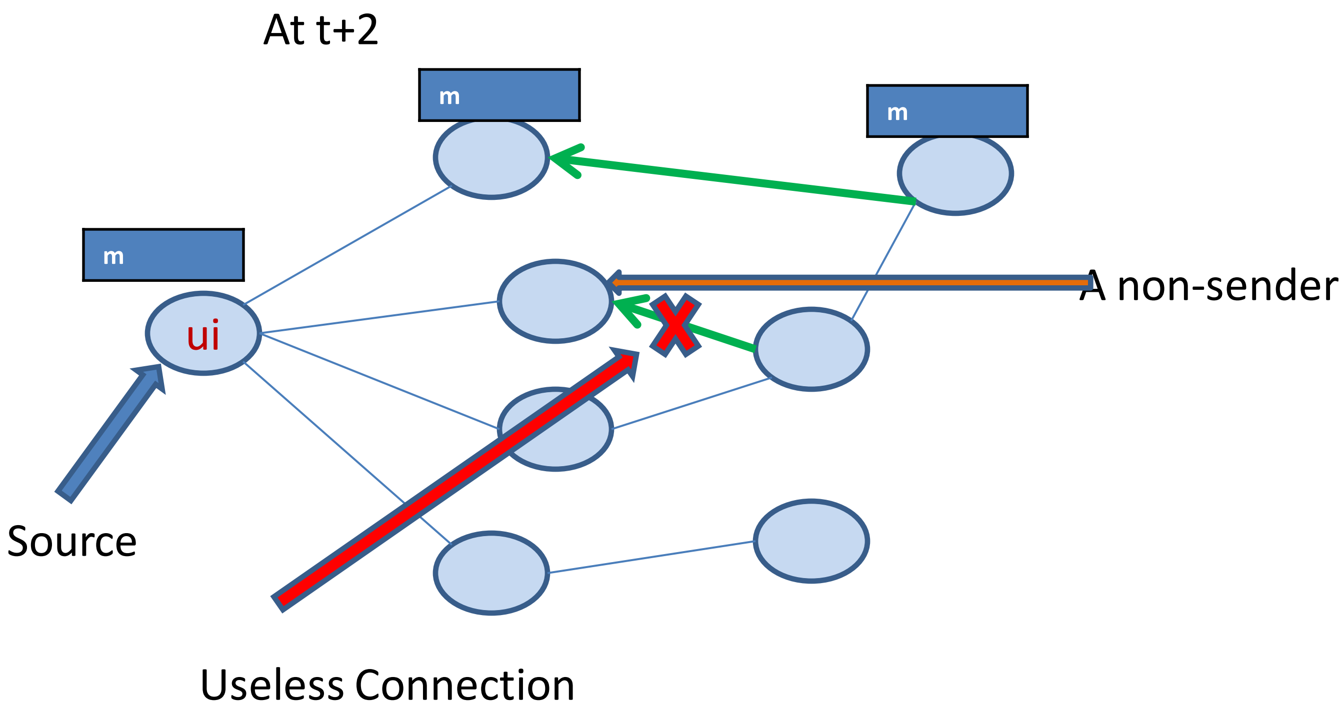
Pull Technique



Pull Technique



Pull Technique



Metrics

- Broadcast Time T*
  - Time from the point the source starts sending the first packet to the point when all the agents in the network have received the entire message
    - Expected broadcast time: E(T)
  - T_i : minimum time at which the system has *i* senders

- Broadcast Wastage C^*_m
  - Measures the number of useless contacts, i.e., the contacts when no packet is transferred
  - $C^*_m = (C_l - C_p)/C_l$, where
    - C_l ➔ the total number of links
    - C_p ➔ the number of successful links

1) Problem definition
2) Setup
3) Algorithms of Message Transfer
4) Deeper analysis of the algorithms
5) Conclusions

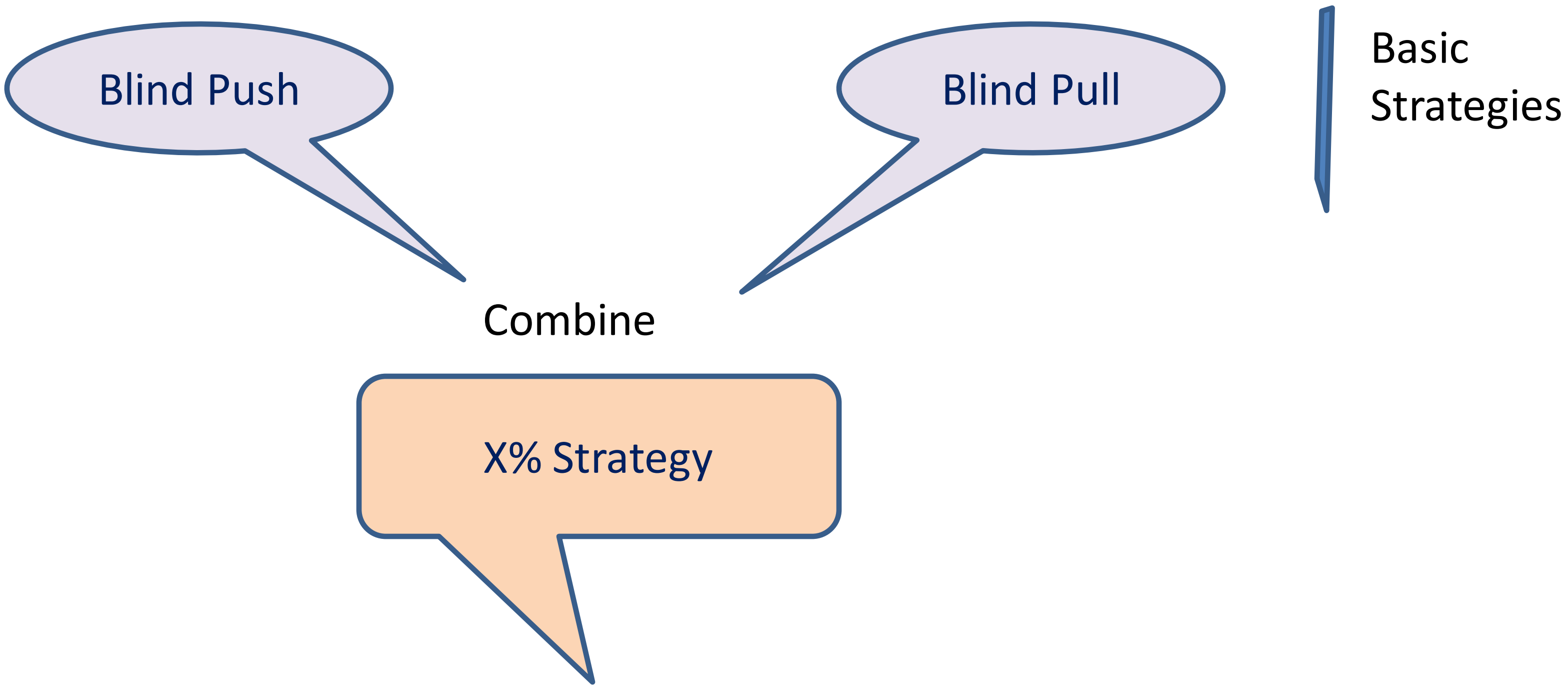
Algorithms for Message Transfer

Blind Push

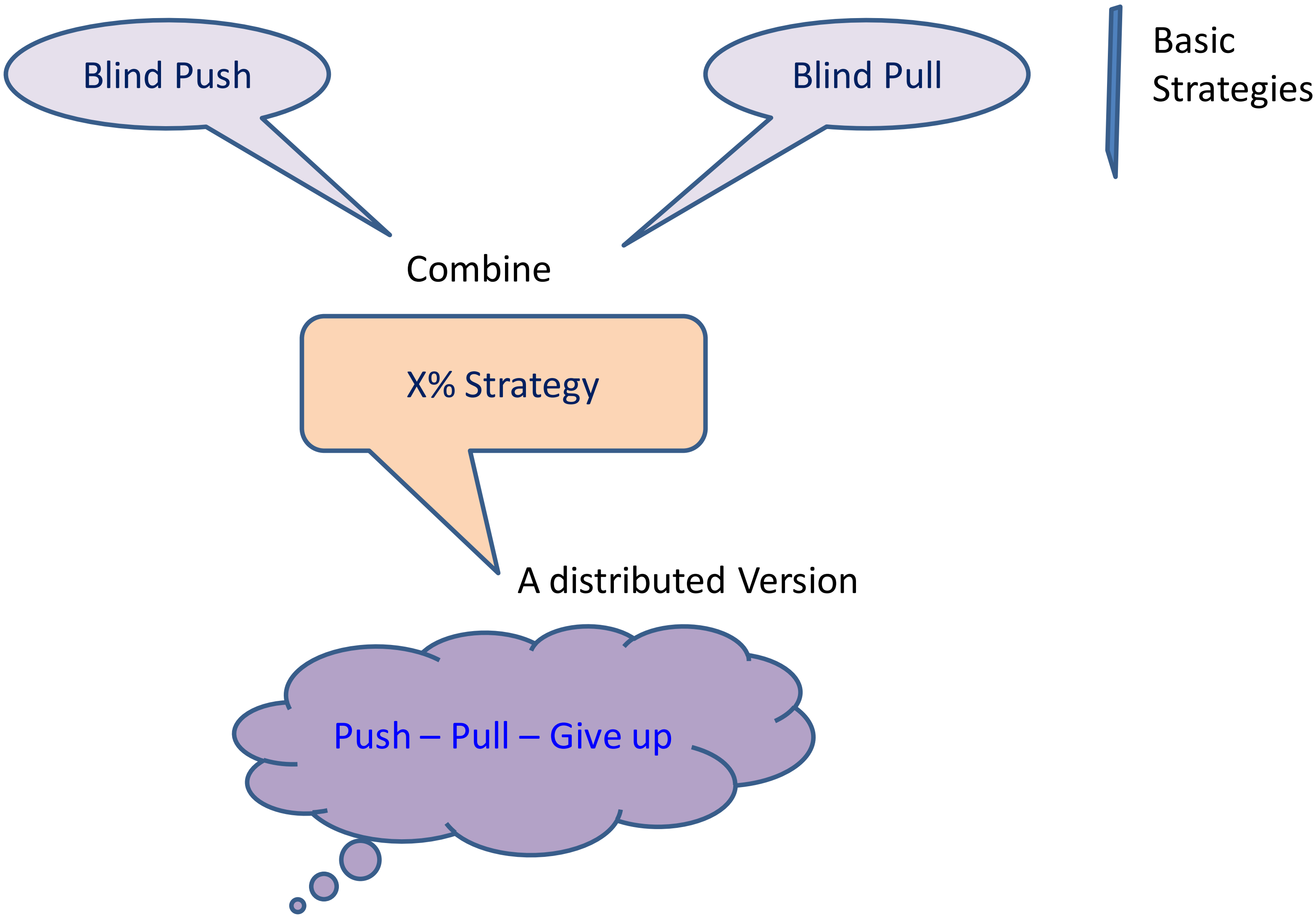
Blind Pull

Basic Strategies

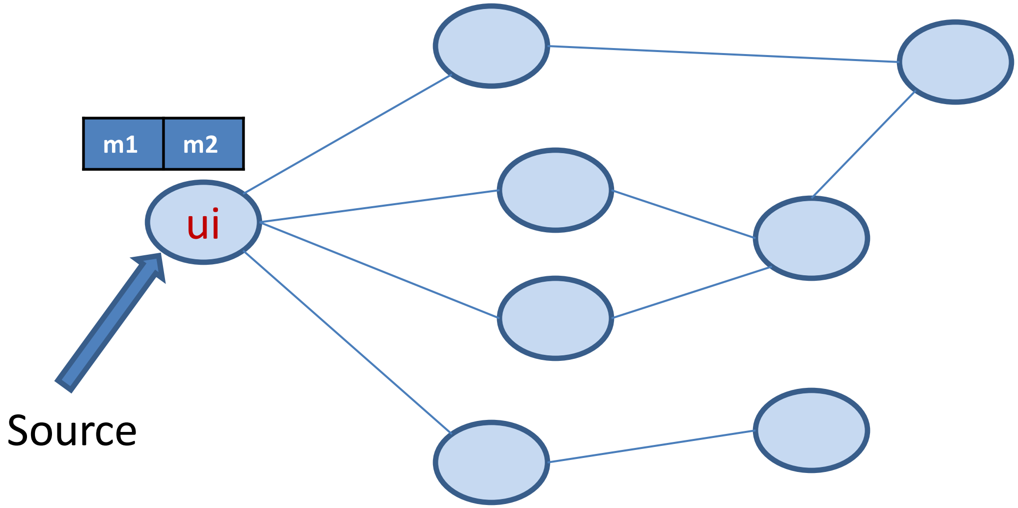
Algorithms for Message Transfer



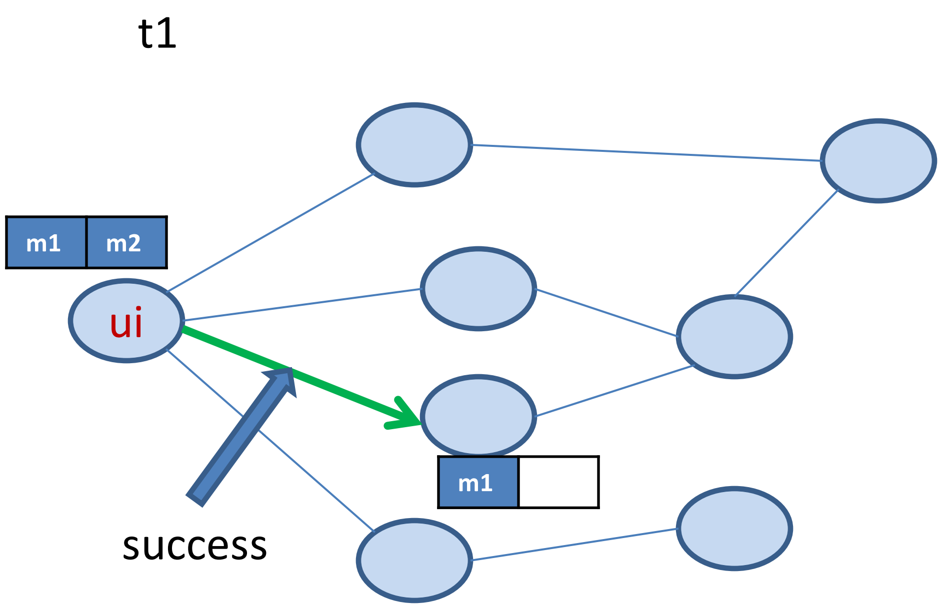
Algorithms for Message Transfer



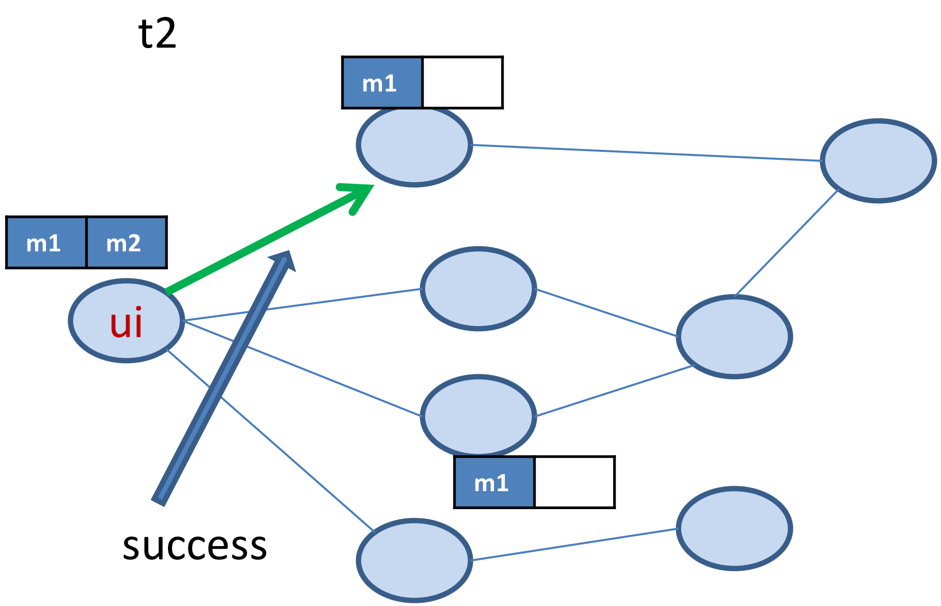
Blind Push (B-P) Technique



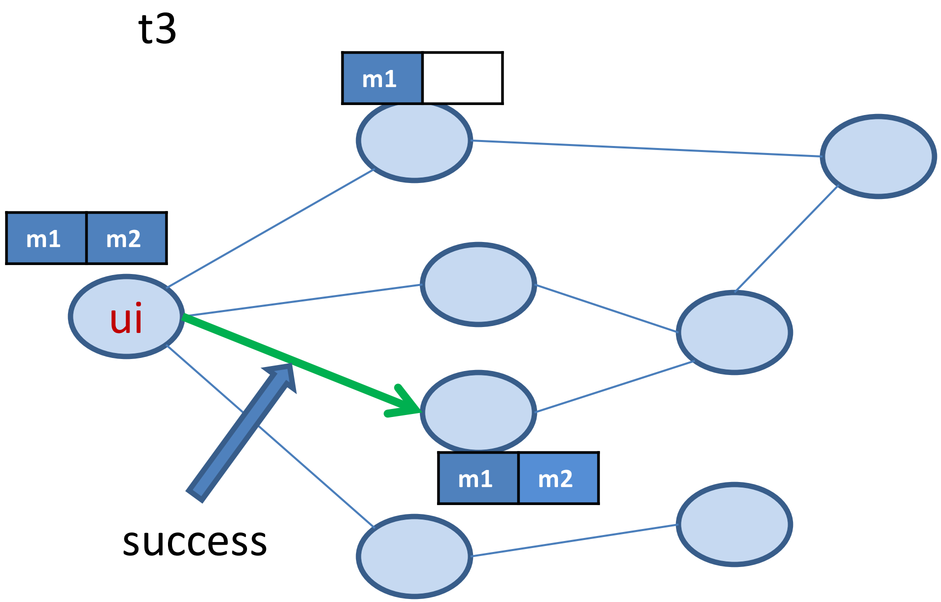
Blind Push (B-P) Technique



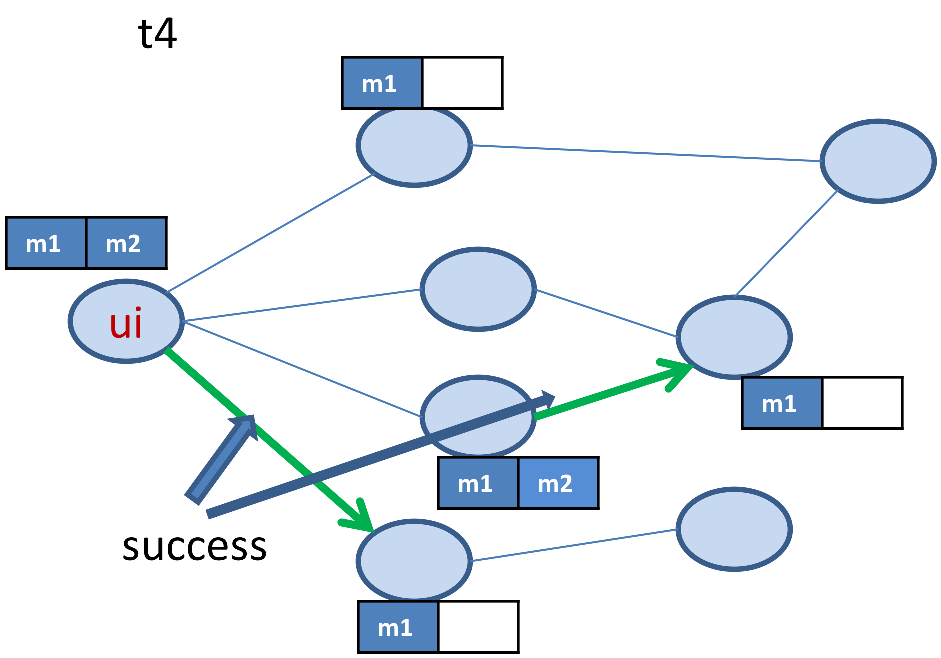
Blind Push (B-P) Technique



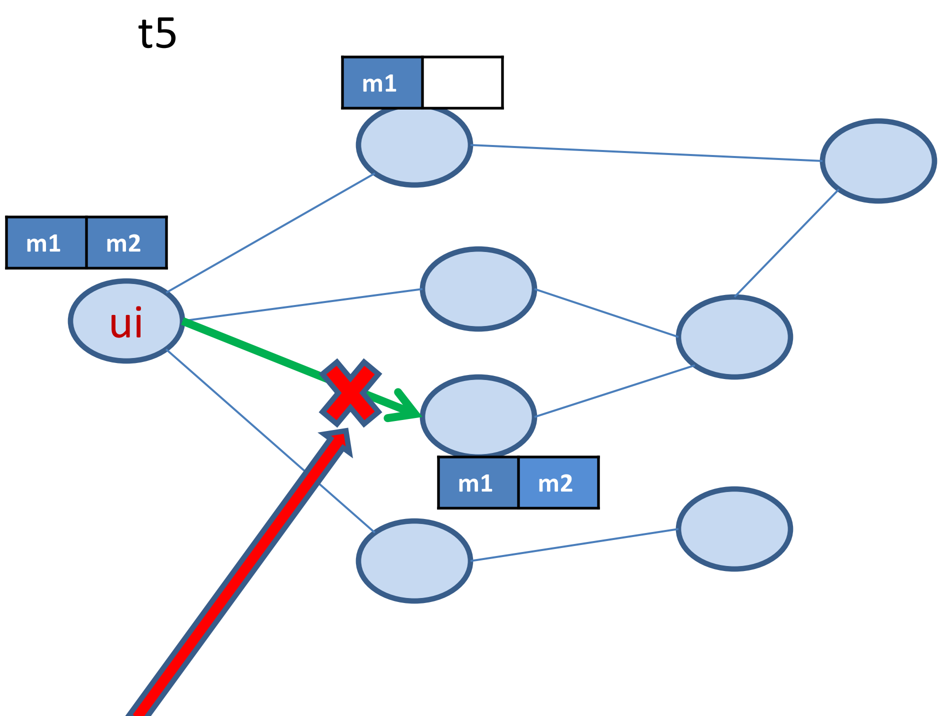
Blind Push (B-P) Technique



Blind Push (B-P) Technique

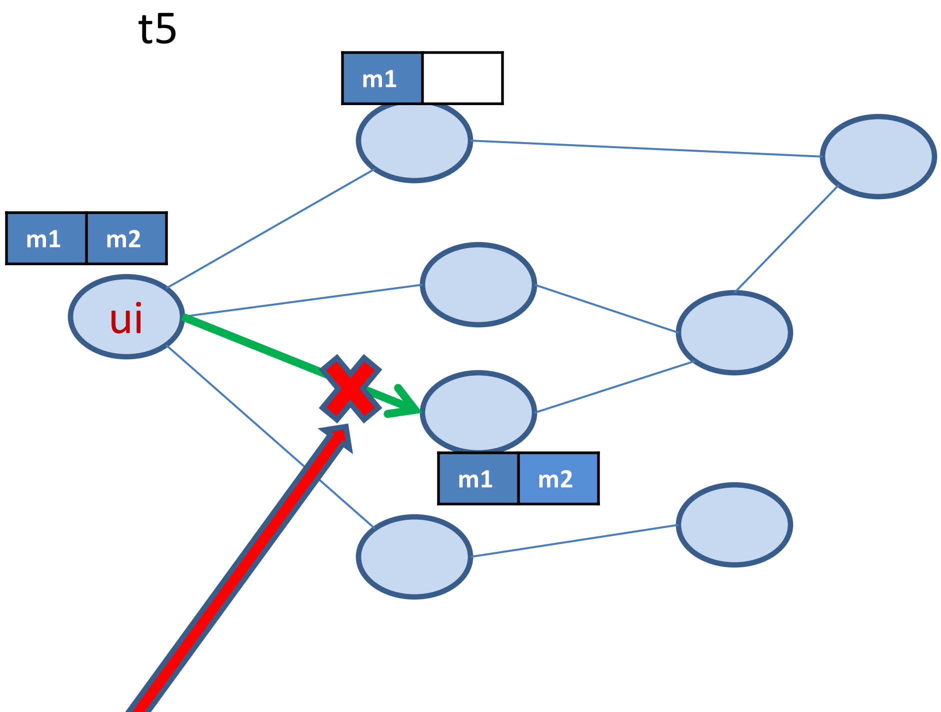


Blind Push (B-P) Technique



No transfer, hence useless contact

Blind Push (B-P) Technique



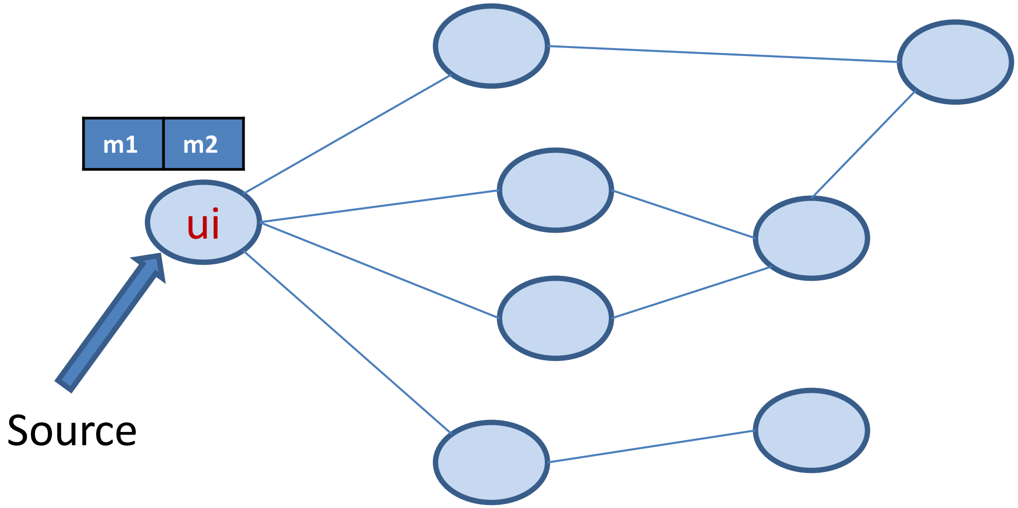
No transfer, hence useless contact

Such failures will be significant towards the end of the broadcast

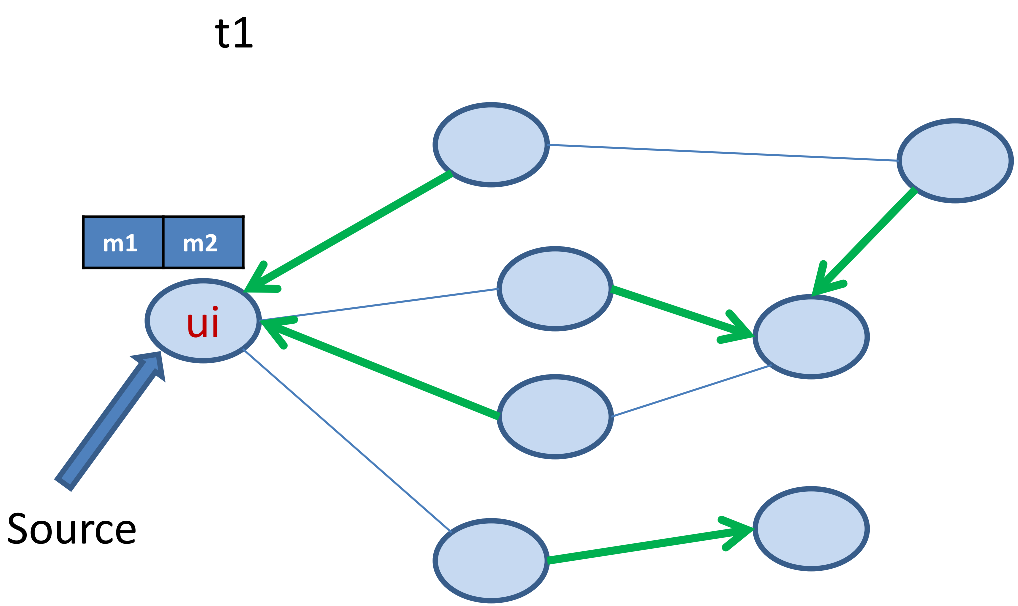
Blind Push (B-P) Technique

How to reduce such failures towards the end?

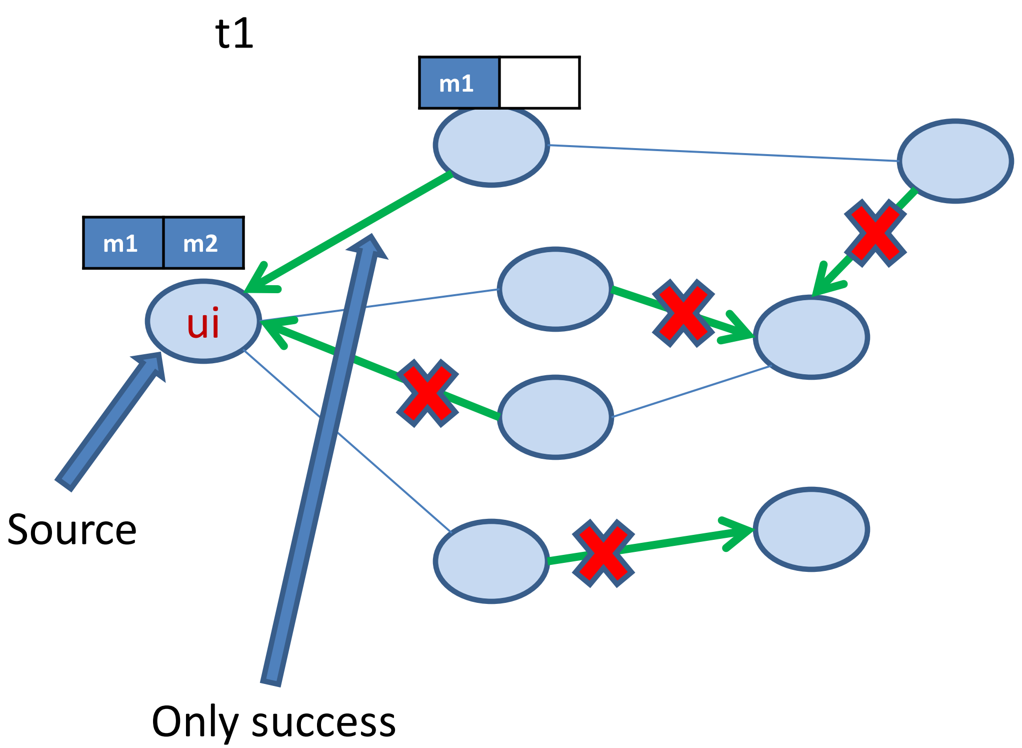
Blind Pull Technique



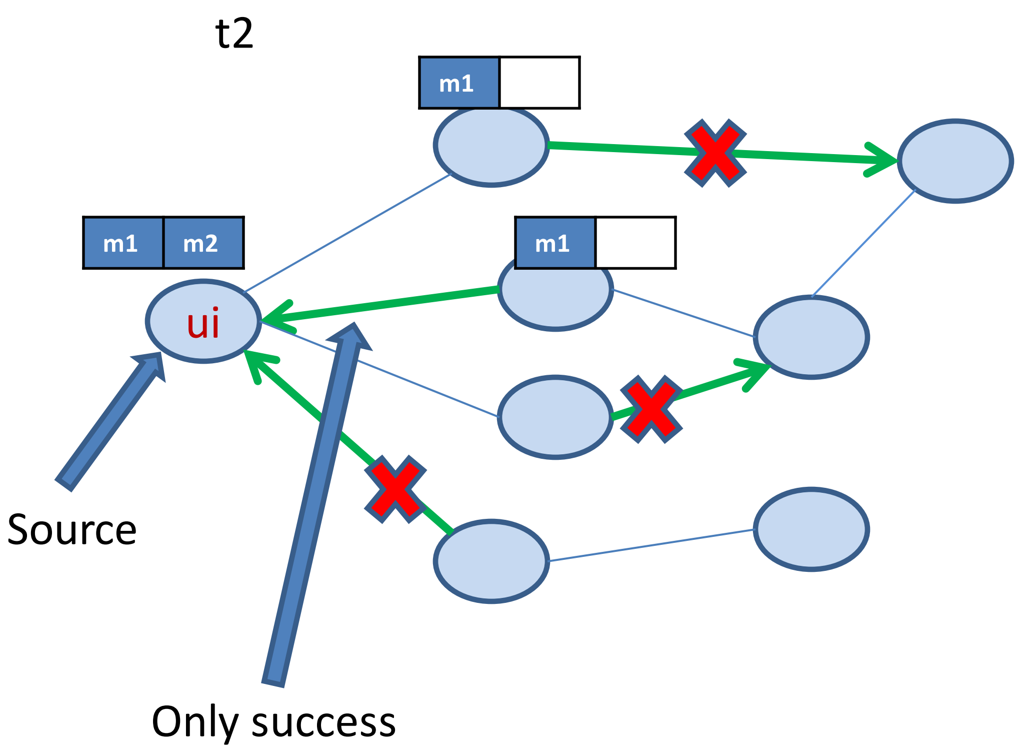
Blind Pull Technique



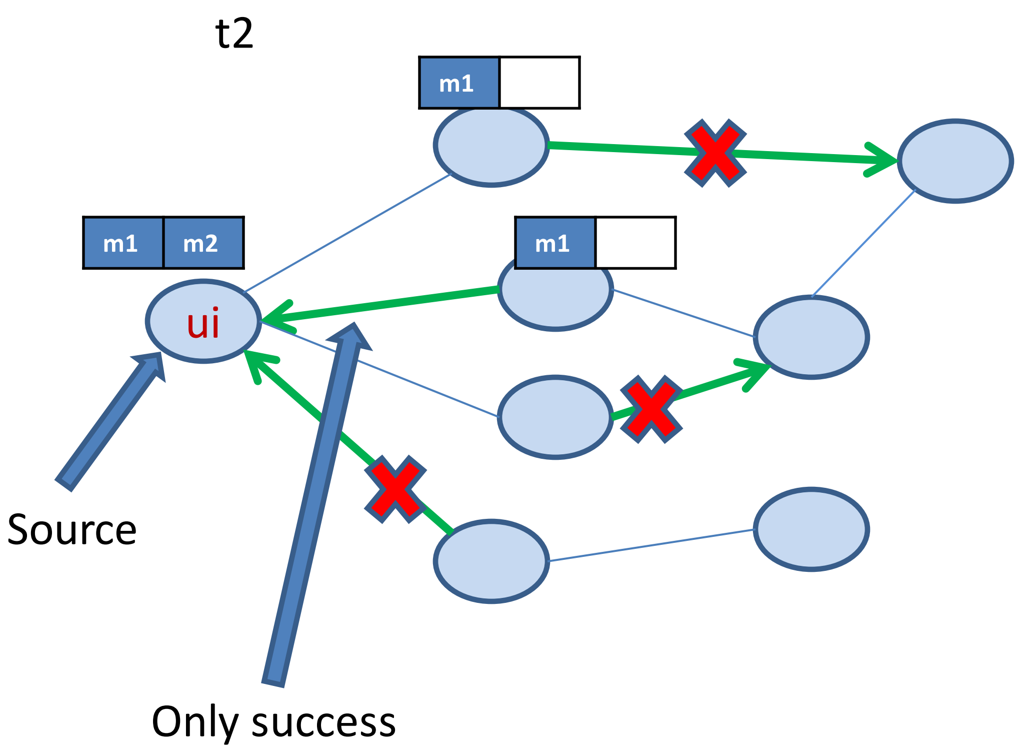
Blind Pull Technique



Blind Pull Technique

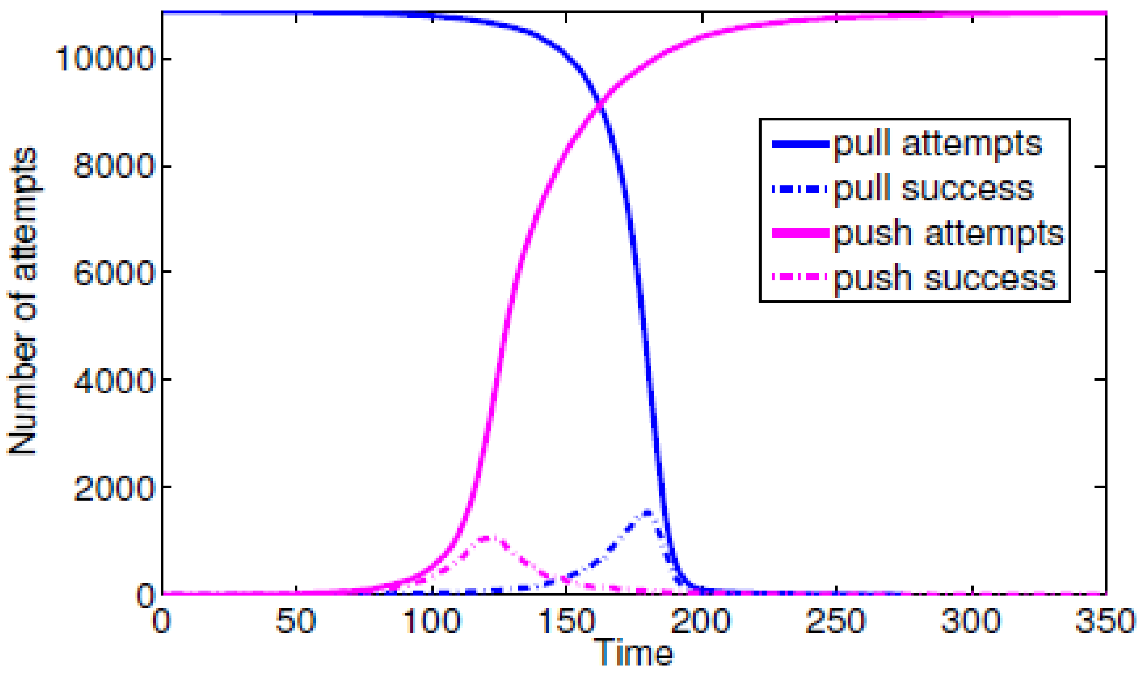


Blind Pull Technique



How to reduce initial failures?

Deeper analysis of the algorithms



- Blind push and blind pull are complementary to each other
- How to take advantages of them and combine?

X% Push-Push Strategy (X-P-P)

- X-P-P combines the advantages of blind push and blind pull

- The process
  - Starts with blind push (B-P)
    - to avoid potential wastage at the beginning
  - After some x% of agents becomes sender, the process converts to blind pull
    - To avoid wastage at the ending

X% Push-Push Strategy (X-P-P)

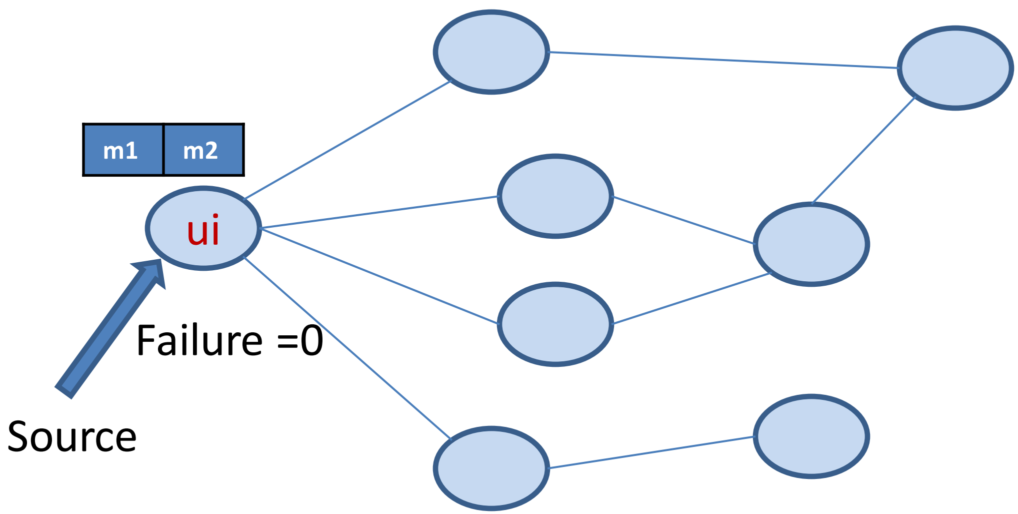
- X-P-P combines the advantages of blind push and blind pull

- The process
  - Starts with blind push (B-P)
    - to avoid potential wastage at the beginning
  - After some x% of agents becomes sender, the process converts to blind pull
    - To avoid wastage at the ending

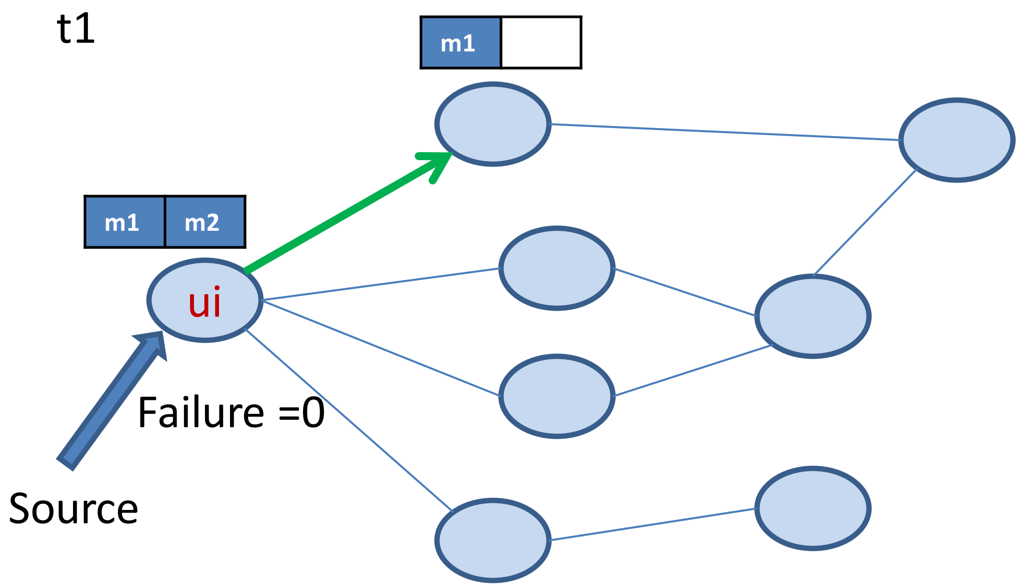
- But, how to estimate x in X-P-P because the system is a envisioned as distributed system?

Push-Pull-with-Giveup (P-P-G)

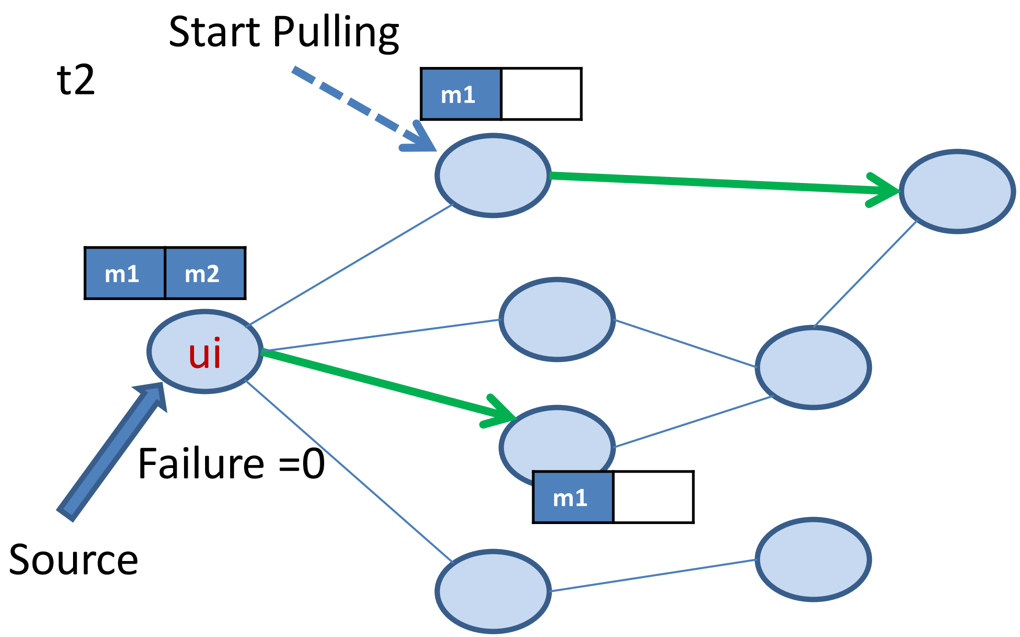
- An approximation of X-P-P to fit distributed environment
- Sender nodes stops "pushing" after a threshold number of failure "pushes"
- Receiving (i.e., non-sender) nodes starts pulling once a packet got received



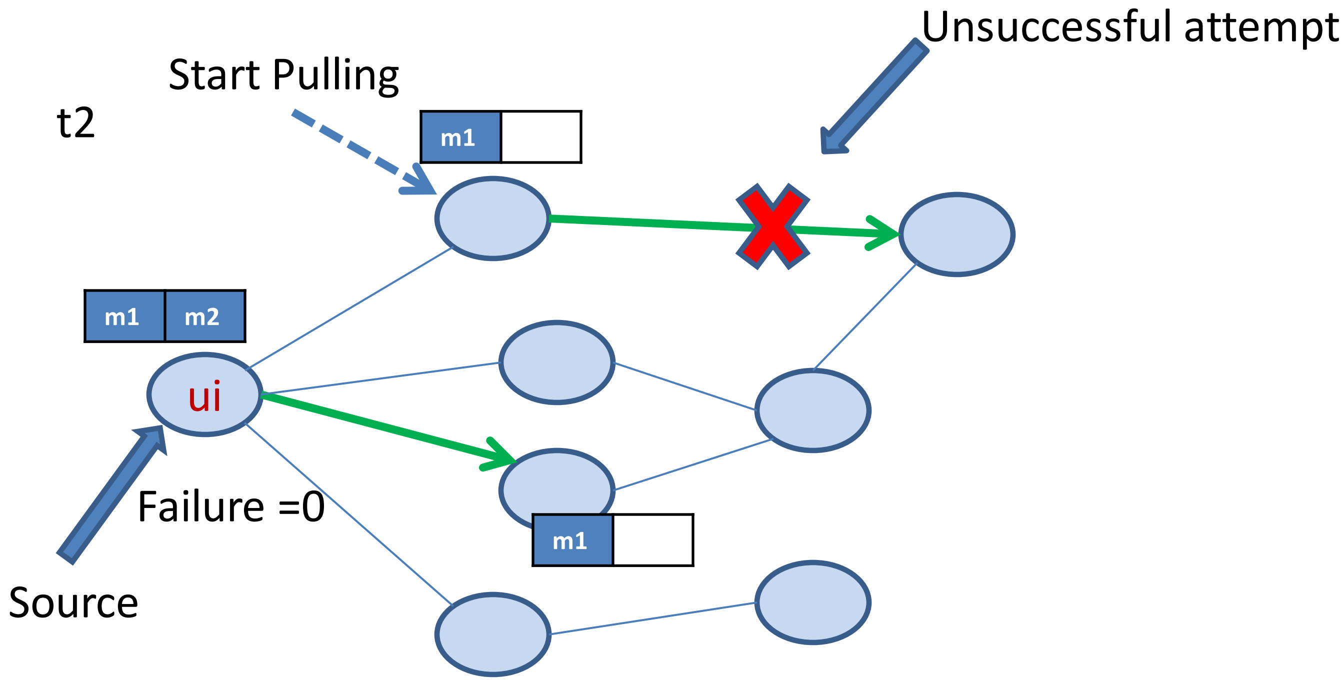
Push-Pull-with-Giveup (P-P-G)



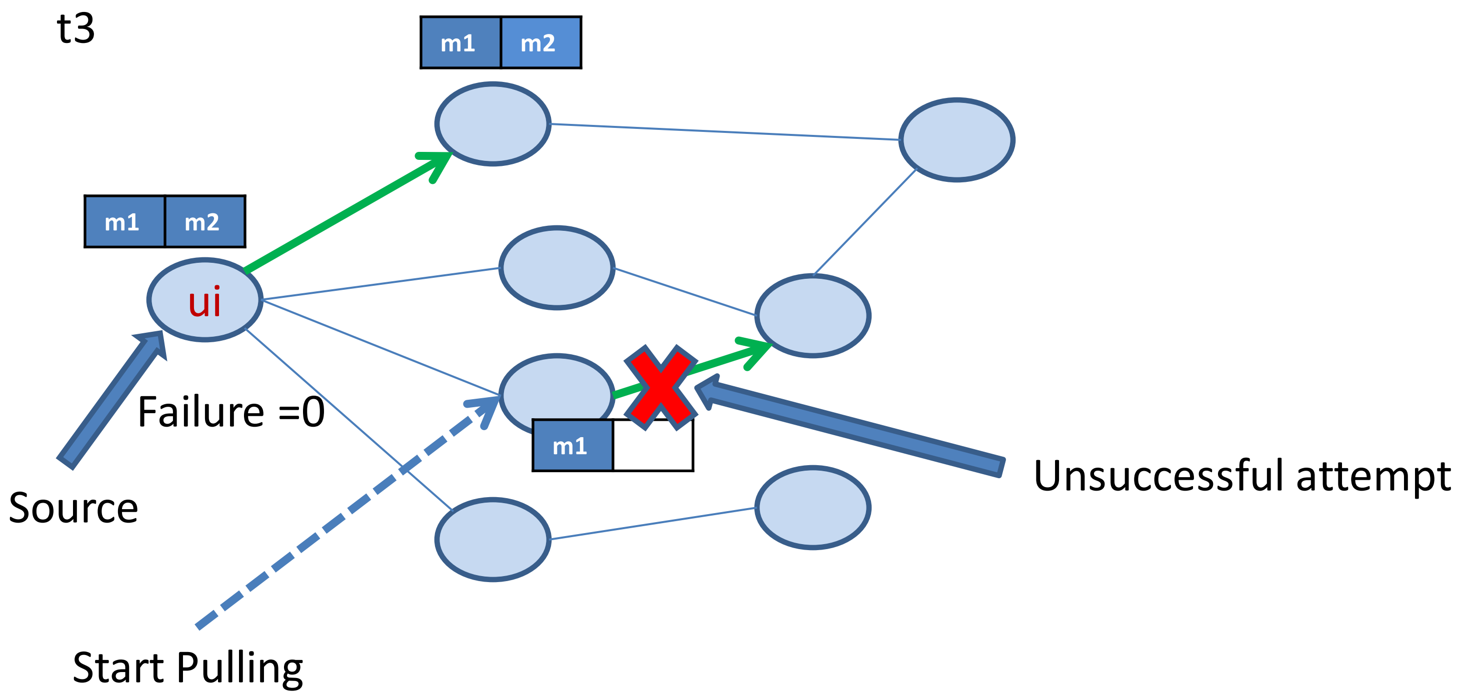
Push-Pull-with-Giveup (P-P-G)



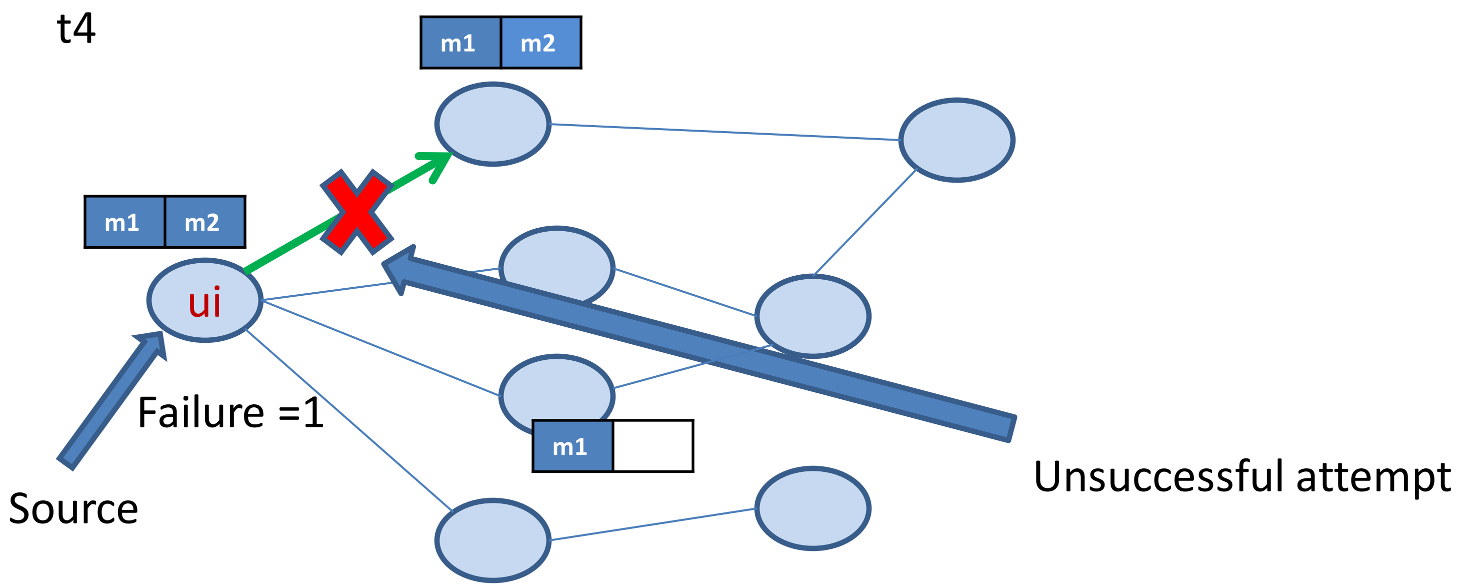
Push-Pull-with-Giveup (P-P-G)



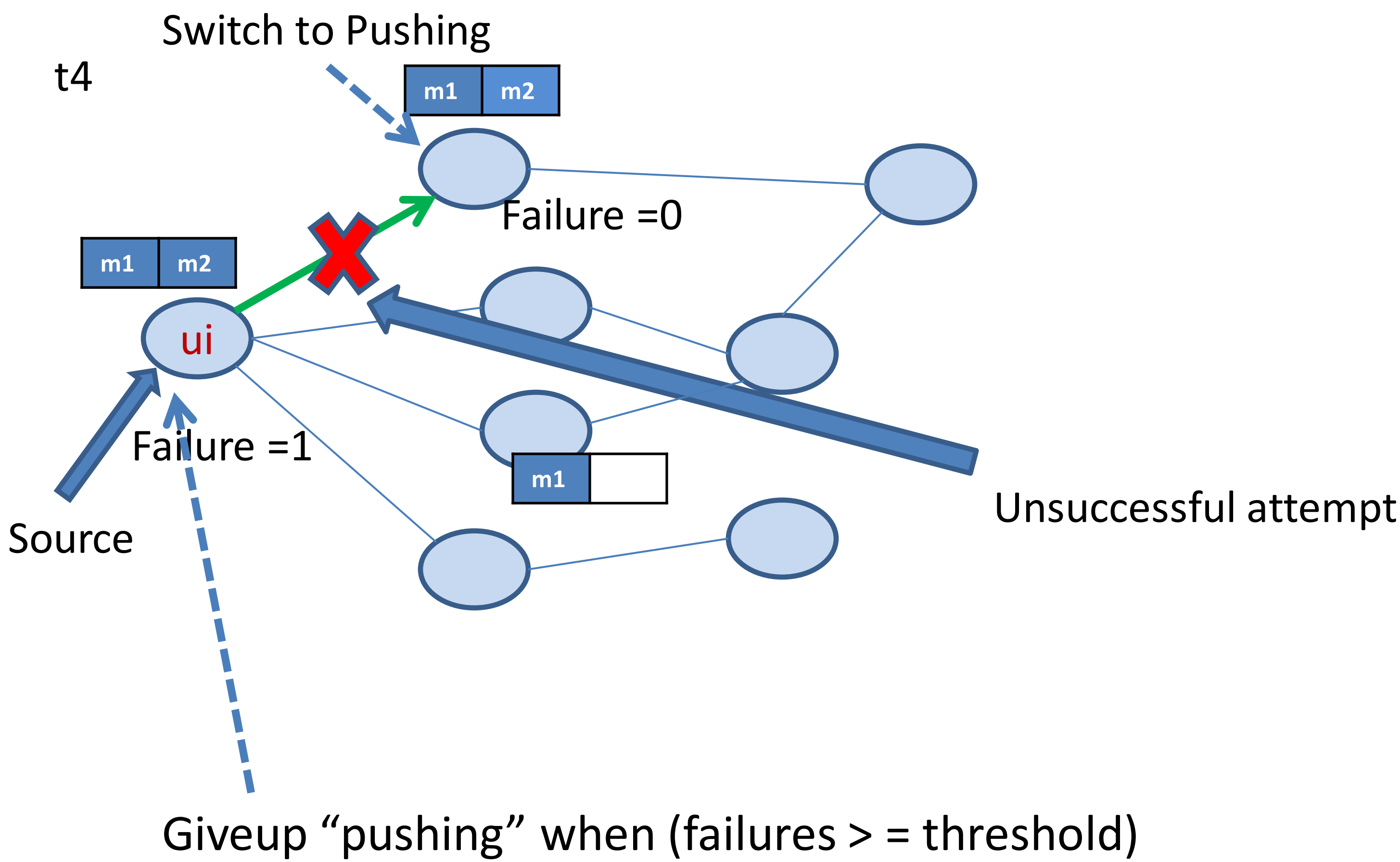
Push-Pull-with-Giveup (P-P-G)



Push-Pull-with-Giveup (P-P-G)



Push-Pull-with-Giveup (P-P-G)



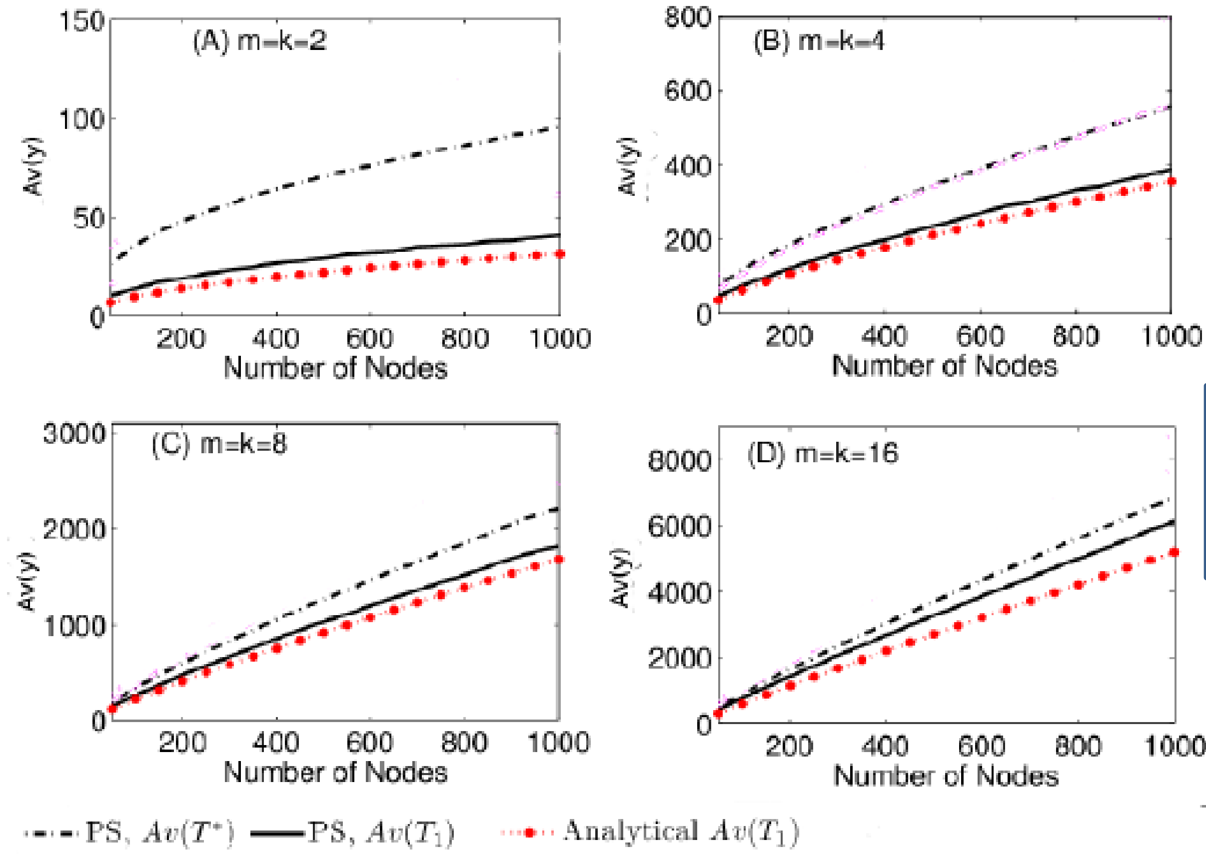
1) Problem definition
2) Setup
3) Algorithms of Message Transfer
4) Deeper analysis of the algorithms
5) Conclusions

Deeper analysis of the algorithms

- Blind Push (B-P) is simple and analytically tractable
  - lets try to see analytical results
- Analyze B-P on a synthetic graph topology of dynamic graphs, like
  - Complete graph, regular graph, regular tree
- Then, find some simulation results on real network topologies, like
  - Gnutella network

Blind Push (B-P) on Complete Graph

- For a complete graph topology we report
  - Expected broadcast time E(T)
  - Expected time $E(T_1)$ to get first sender other than source



Note: $E(T_1)$ is an indicator of E(T)

Estimation of T_1

- Ratio of T* and T_1 converges to a constant
- Estimation of T_1

Estimate T_1 as:

$$\Pr\{T_1 = t\} = \frac{t-1}{n}\prod_{l=1}^{t-2}\left(1 - \frac{l}{n}\right), t \geq 2$$

- Sketch: consider a message has 2 packets (for simplicity)
    - Minimum time required to create first sender = 2
    - Pr(T_1 = t) implies that up to t- 1 time only those nodes are selected that has no packet, and at t, a node out of t-1 nodes are selected

Estimation of T_1

- Thus,

$$\Pr\{T_1 = t\} = \underbrace{(1 - \frac{1}{n})(1 - \frac{2}{n})(1 - \frac{3}{n})...(1 - \frac{t-2}{n})}(\frac{t-1}{n})$$

This part ensures that none of t-1 nodes got selected up to time t-1

Estimation of T_1

- Thus,

$$\Pr\{T_1 = t\} = (1 - \frac{1}{n})(1 - \frac{2}{n})(1 - \frac{3}{n})...(1 - \frac{t-2}{n})(\frac{t-1}{n})$$

This part ensures that none of t-1 nodes got selected up to time t-1

This part ensures that a node out of t-1 nodes got selected at time t

Estimation of T_1

- Thus,

$$\Pr\{T_1 = t\} = \underbrace{(1 - \frac{1}{n})(1 - \frac{2}{n})(1 - \frac{3}{n})...(1 - \frac{t-2}{n})}(\frac{t-1}{n})$$

This part ensures that none of t-1 nodes got selected up to time t-1

This part ensures that a node out of t-1 nodes got selected at time t

$$= \frac{t-1}{n} \prod_{l=1}^{t-2} (1 - \frac{l}{n}), t \geq 2$$

Estimation of T_1

- Thus,

$$\Pr\{T_1 = t\} = \underbrace{(1-\frac{1}{n})(1-\frac{2}{n})(1-\frac{3}{n})...(1-\frac{t-2}{n})}(\frac{t-1}{n})$$

This part ensures that none of t-1 nodes got selected up to time t-1

This part ensures that a node out of t-1 nodes got selected at time t

$$= \frac{t-1}{n}\prod_{l=1}^{t-2}(1-\frac{l}{n}), t \geq 2$$

Estimation of T_1

- Thus,

$$\Pr\{T_1 = t\} = (1 - \frac{1}{n})(1 - \frac{2}{n})(1 - \frac{3}{n})...(1 - \frac{t-2}{n})(\frac{t-1}{n})$$

$$= \frac{t-1}{n} \prod_{l=1}^{t-2} (1 - \frac{l}{n}), t \geq 2$$

- Generalized for k-packets message

$$\Pr(T_1 = t) \sim \frac{t^{k-1}}{(k-1)!n^{k-1}} \prod_{i}^{t} \left(1 - \frac{i^{k-1}}{(k-1)!n^{k-1}}\right)$$

$$\sim \frac{t^{k-1}}{(k-1)!n^{k-1}} e^{-\frac{t^k}{k!n^{k-1}}}$$

Estimation of T_1

- Thus,

$$\Pr\{T_1 = t\} = (1 - \frac{1}{n})(1 - \frac{2}{n})(1 - \frac{3}{n})...(1 - \frac{t-2}{n})(\frac{t-1}{n})$$

$$= \frac{t-1}{n} \prod_{l=1}^{t-2} (1 - \frac{l}{n}), t \geq 2$$

- Generalized for k-packets message

$$\Pr(T_1 = t) \sim \frac{t^{k-1}}{(k-1)!n^{k-1}} \prod_i^t \left(1 - \frac{i^{k-1}}{(k-1)!n^{k-1}}\right)$$

$$\sim \frac{t^{k-1}}{(k-1)!n^{k-1}} e^{-\frac{t^k}{k!n^{k-1}}}$$

$$E(T_1) = \sum t * \frac{t^{k-1}}{(k-1)!n^{k-1}} e^{-\frac{t^k}{k!n^{k-1}}} \Rightarrow n^{\frac{k-1}{k}}$$

Blind Push on Sparse Networks

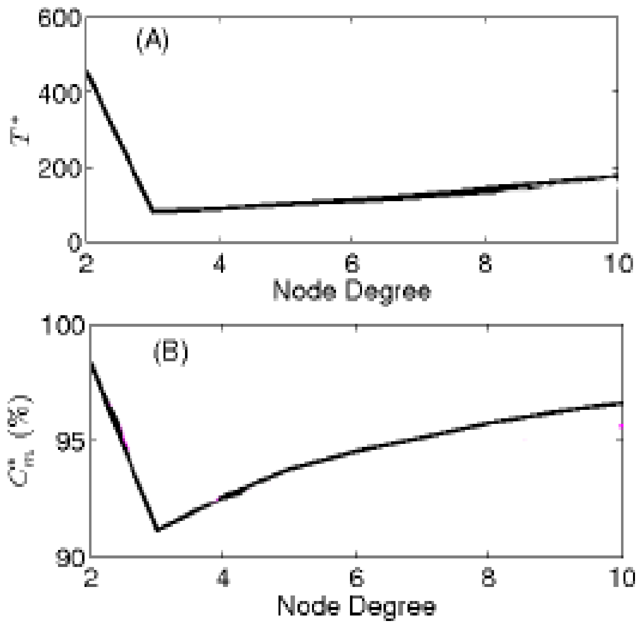


Figure : Regular tree

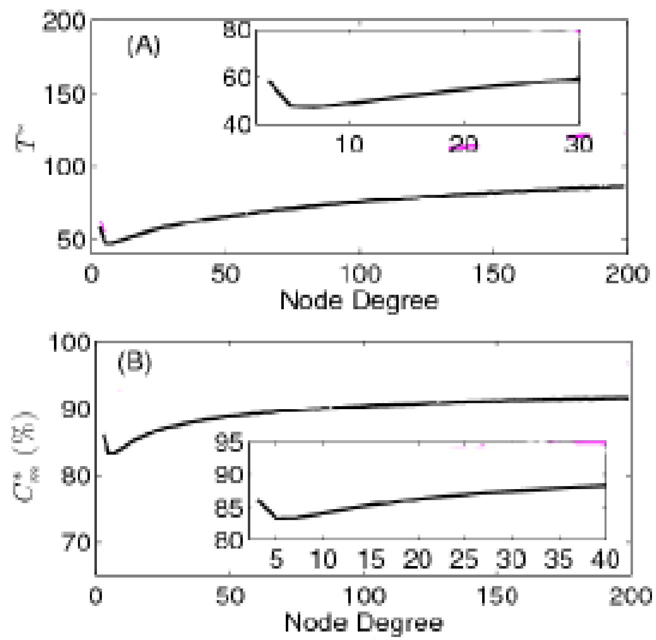


Figure : Regular graph

- An optimal degree d can be found where broadcast delay and wastage can be minimum

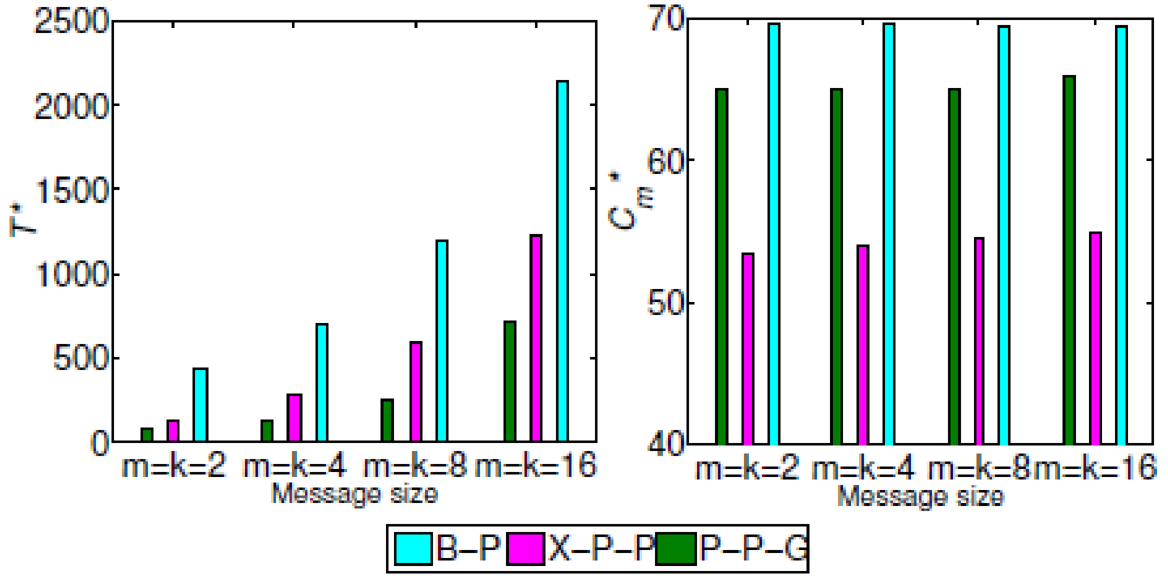
B-P, X-P-P, & P-P-G on Real Network Topology

- Three snapshots of Gnutella network (taken in 2002)

Data set	#Nodes	#Edges	Avg. degree
Gnutella_1	10876	39994	7.34
Gnutella_2	8717	31525	7.24
Gnutella_3	22663	54693	5.82

Comparison of the Broadcast Algorithms

- The optimal X (point where the system switches to Pull) for the three networks are 50, 50 and 60 respectively



	Gain of P-P-G in Delay		Gain of P-P-G in Wastage	
	Over B-P	Over X-P-P	Over B-P	Over X-P-P
Gnutella_1	5.45	1.5	1.10	0.75
Gnutella_2	5.9	1.6	1.06	0.72

Observations

- With respect to B-P, we gain in both broadcast time and wastage while with respect to X-P-P we gain only in broadcast time

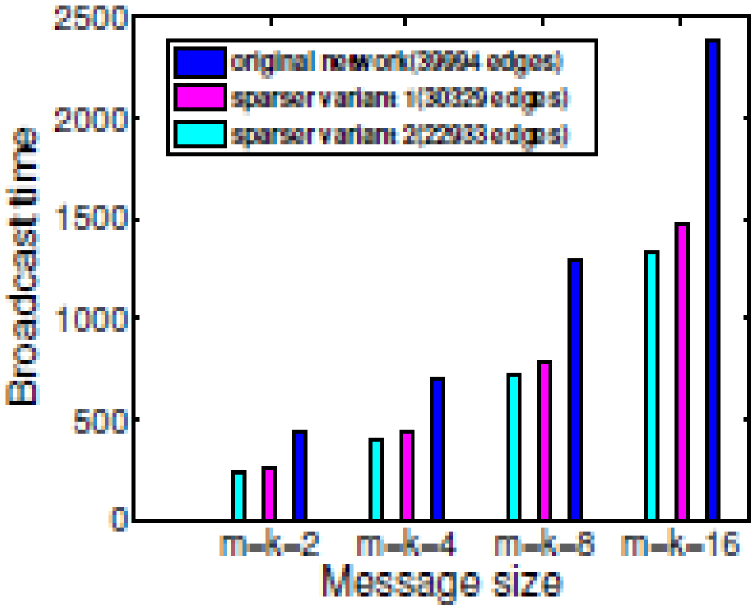
- X-P-P provides optimal trade-o between time and wastage but is not practically implementable

- P-P-G gives best trade-o and can be implemented in distributed fashion with negligible computational overhead

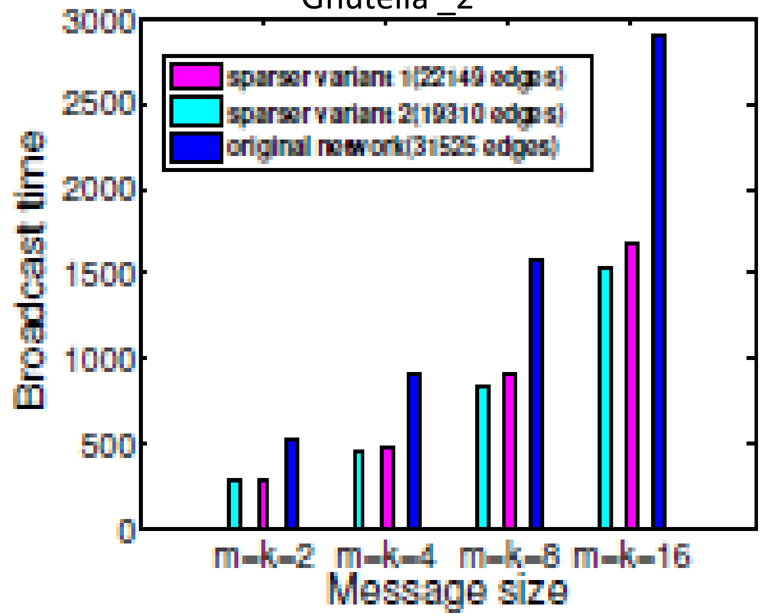
Effect of Sparsity on Broadcast Delay on Real Data Set

- Already observed presence of an optimal degree in case of synthetic networks
- To perform simulations on the sparser variants of the Gnutella networks
  - removed a set of the network edges keeping the network connected

Gnutella _1



Gnutella _2



The broadcast time reduces significantly for the sparser variants

1) Problem definition
2) Setup
3) Algorithms of Message Transfer
4) Deeper analysis of the algorithms
5) Conclusions

Conclusion

- Define a new problem in the space of information dissemination and broadcast in dynamic networks
- The speed of broadcast for segmented message differs from single message case
- Presence of an optimal d (average degree) irrespective of topology where the broadcast is most efficient
- Sparseness leads to faster broadcast

Thank You