Peer-to-Peer Networks

17 Kelips and Epidemic Algorithms

Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

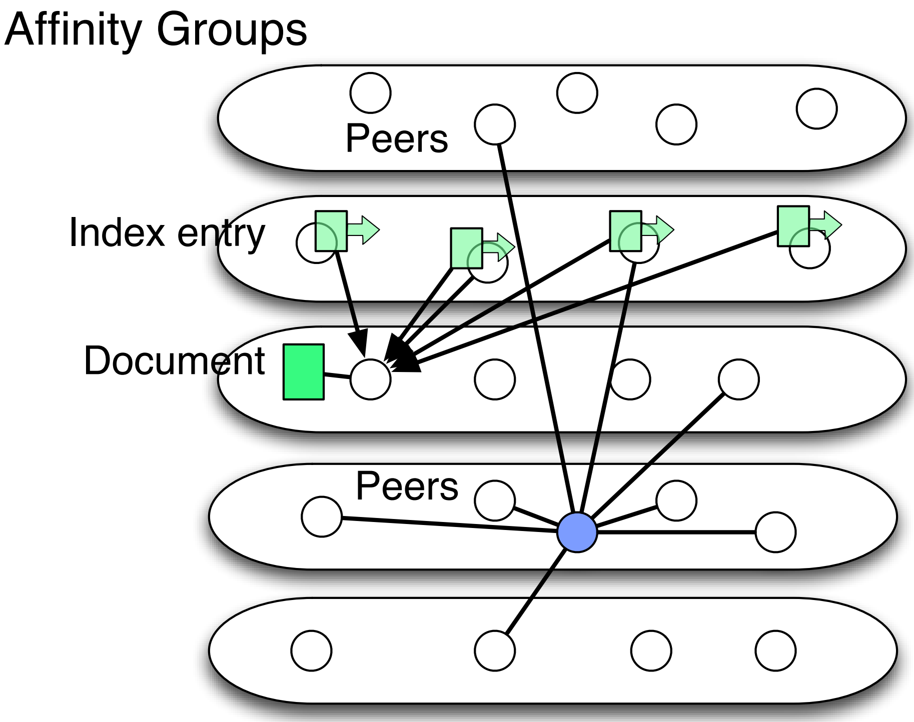
Kelips

- Indranil Gupta, Ken Birman, Prakash Linga, Al Demers, Robbert van Renesse
  - Cornell University, Ithaca, New York
- Kelip-kelip
  - malay name for synchronizing fireflies
- P2P Network
  - uses DHT
  - constant lookup time
  - $O(n^{1/2})$ storage size
  - fast and robust update

Copyrights @ 1998 - 2008 by TourMalaysia

Kelips Overview

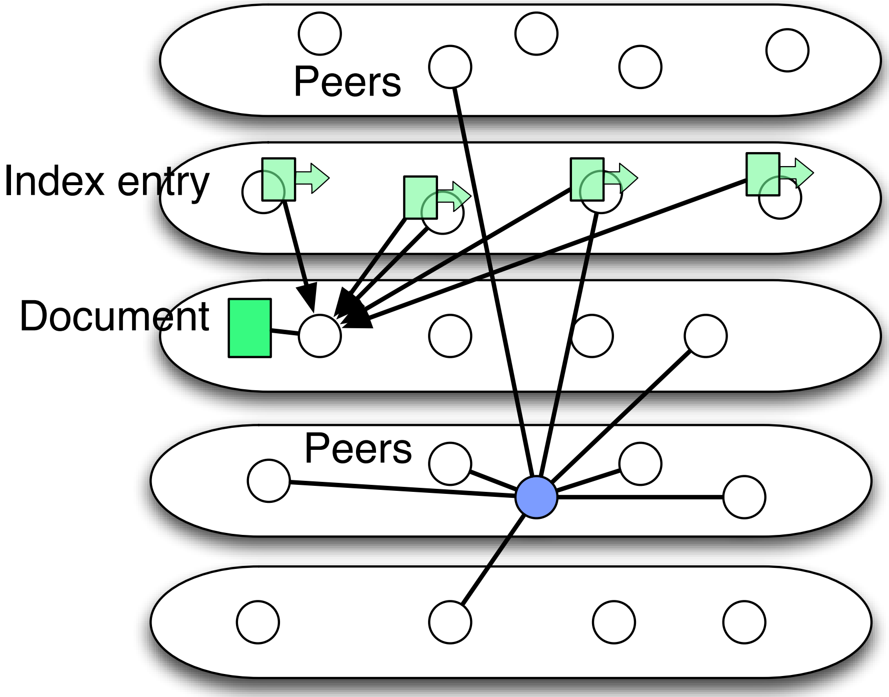
- Peers are organized in k affinity groups
  - peer position chosen by DHT mechanism
  - k is chosen as $n^{1/2}$ for n peers
- Data is mapped to an affinity group using DHT
  - all members of an affinity group store all data
- Routing Table
  - each peer knows all members of the affinity group
  - each peer knows at least one member of each affinity group
- Updates
  - are performed by epidemic algorithms



Routing Table

- Affinity Group View
  - Links to all O(n/k) group members
  - This set can be reduced to a partial set as long as the update mechanism works
- Contacts
  - For each of the other affinity group a small (constant-sized) set of nodes
  - O(k) links
- Filetuples
  - A (partial) set of tuples, each detailing a file name and host IP address of the node storing the file
  - O(F/k) entries, if F is the overall number of files
- Memory Usage: O(n/k + k + F/k)
  - for $\mathrm{k} = \mathrm{O}\left(\sqrt{n+F}\right)$

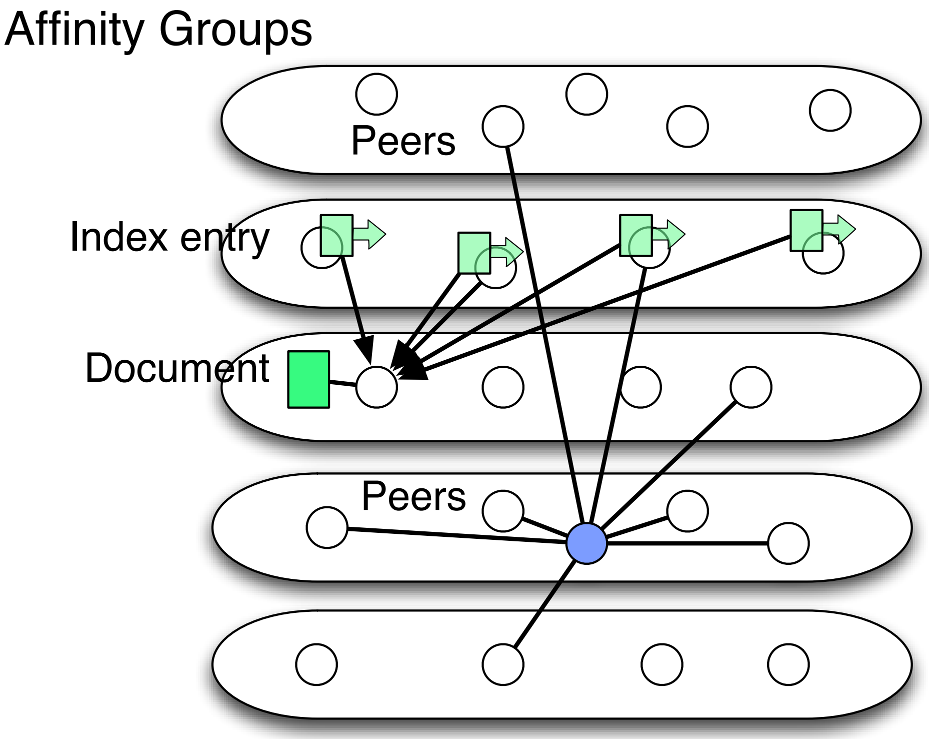
Affinity Groups



$$\mathrm{O}\left(\sqrt{n+F}\right)$$

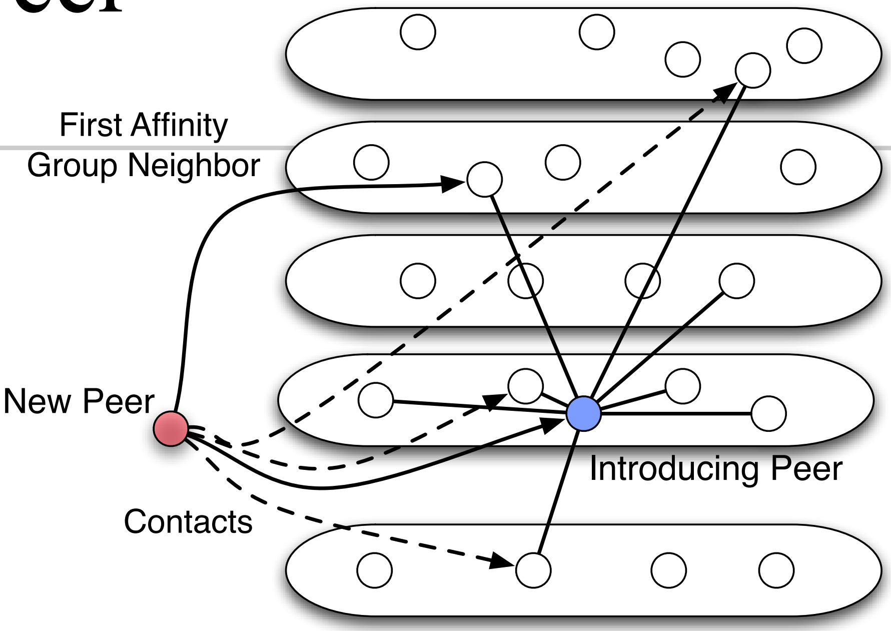
Lookup

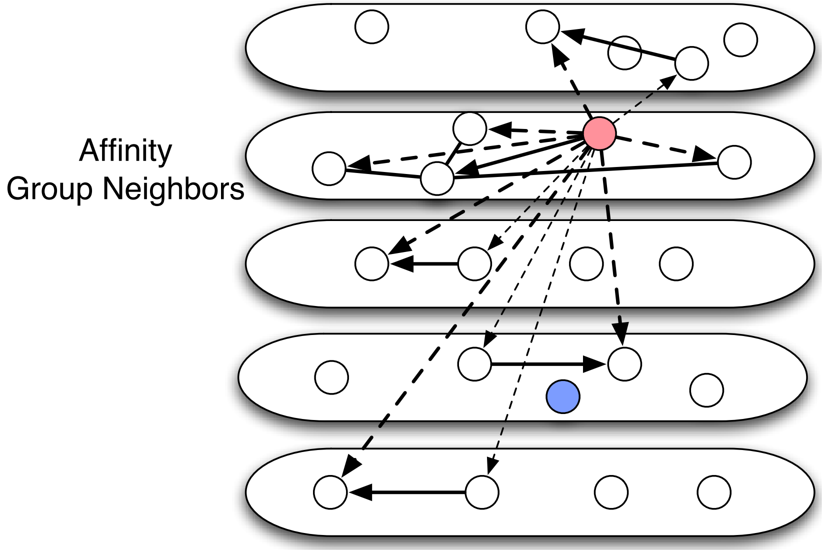
- Lookup-Algorithm
  - compute index value
  - find affinity group using hash function
  - contact peer from affinity group
  - receive index entry for file (if it exists)
  - contact peer with the document
- Kelips needs four hops to retrieve a file



Inserting a Peer

- Algorithm
    - Every new peer is introduced by a special peer, group or other method,
        - e.g. web-page, forum etc.
    - The new peer computes its affinity group and contacts any peer
    - The new peer asks for one contact of the affinity group and copies the contacts of the old affinity group
    - By contacting a neighbor node in the affinity group it receives all the necessary contacts and index filetuples
    - Every contact is replaced by a random replacement (suggested by the contact peer)
    - The peer starts an epidemic algorithm to update all links
- Except the epidemic algorithm the runtime is O(k) and only O(k) messages are exchanged





How to Add a Document

- Start an Epidemic Algorithm to Spread the news in the affinity group
- Such an algorithm uses O(n/k) messages and needs O(log n) time
- We introduce Epidemic Algorithms later on

How to Check Errors

- Kelip works in heartbeats, i.e. discrete timing
- In every heartbeat each peer checks one neighbor
- If a neighbor does not answer for some time
  - it is declared to be dead
  - this information is spread by an epidemic algorithm
- Using the heartbeat mechanisms all nodes also refresh their neighbors
- Kelips quickly detects missing nodes and updates this information

Discussion

- Kelips has lookup time O(1), but needs O($n^{1/2}$) sized Routing Table
  - not counting the O(F/$n^{1/2}$) Filetuples
- Chord, Pastry & Tapestry use lookup time O(log n) but only O(log n) memory units
- Kelips is a reasonable choice for medium sized networks
  - up to some million peers and some hundred thousands index entries

To Do

- What is an Epidemic Algorithm

Epidemic Spread of Viruses

- Observation
  - most viruses do not prosper in real life
  - other viruses are very successful and spread fast
- How fast do viruses spread?
- How many individuals of the population are infected?
- Problem
  - social behavior and infection risk determine the spread
  - the reaction of a society to a virus changes the epidemy
  - viruses and individuals may change during the infection

Mathematical Models

- SI-Model (rumor spreading)
  - susceptible → infected
- SIS-Model (birthrate/deathrate)
  - susceptible → infected → susceptible
- SIR-Model
  - susceptible → infected → recovered
- Continuos models
  - deterministic
  - or stochastic
- Lead to differential equations
- Discrete Models
  - graph based models
  - random call based
- Lead to the analysis of Markov Processes

Infection Models

- SI-Model (rumor spreading)
  - susceptible → infected
  - At the beginning one individual is infected
  - Every contact infects another indiviual
  - In every time unit there are in the expectation ß contacts
- SIS-Model (birthrate/deathrate)
  - susceptible → infected → susceptible
  - similar as in the SI-Model, yet a share of δ of all infected individuals is healed and can receive the virus again
  - with probability δ an individual is susceptible again
- SIR-Model
  - susceptible → infected → recovered
  - like SI-Model, but healed individuals remain immune against the virus and do not transmit the virus again

SI-Model

- Variables
  - n: total number of individuals
    - remains constant
  - S(t): number of (healthy) susceptible individuals at time t
  - I(t): number of infected individuals
- Relative shares
  - s(t) := S(t)/n
  - i(t) := I(t)/n
- At every time unit each individual contacts ß partners
- Assumptions:
  - Among ß contact partnres ß s(t) are susceptible
  - All I(t) infected individuals infect ß s(t) I(t) other individuals in each round
- Leads to the following recursive equations:
  - I(t+1) = I(t) + ß s(t) I(t)
  - i(t+1) = i(t) + ß i(t) s(t)
  - S(t+1) = S(t) – ß s(t) I(t)
  - s(t+1) = s(t) – ß i(t) s(t)

SI-Model

- i(t+1) = i(t) + ß i(t) s(t)
- s(t+1) = s(t) – ß i(t) s(t)
- Idea:
  - i(t) is a continuous function
  - i(t+1)-i(t) approximate first derivative

$$\frac{i(t+1) - i(t)}{1} \approx \frac{di(t)}{dt}$$

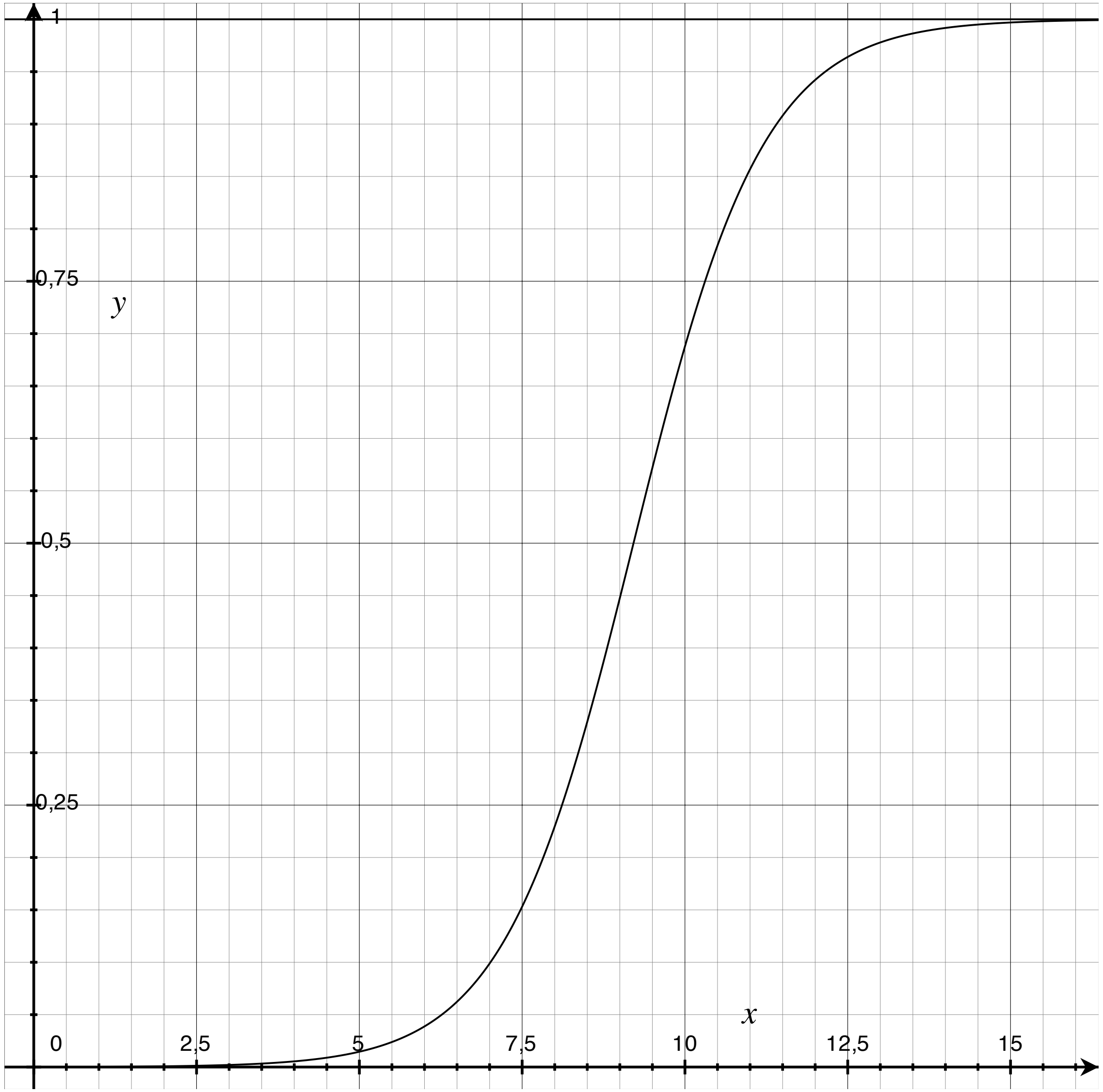
$$\frac{di(t)}{dt} = \beta \cdot i(t)(1 - i(t))$$

- Solution:

$$i(t) = \frac{1}{1 + \left(\frac{1}{i(0)} - 1\right) e^{-\beta t}}$$

SI-Model

- The number of infected grows exponentially until half of all members are infected
- Then the number of susceptible decrease exponentially



SIS-Model

- Variables
  - n: total number of individuals
    - remains constant
  - S(t): number of (healthy) susceptible individuals at time t
  - I(t): number of infected individuals
- Relative shares
  - s(t) := S(t)/n
  - i(t) := I(t)/n
- At every time unit each individual contacts ß partners

- Assumptions:
  - Among ß contact partnres ß s(t) are susceptible
  - All I(t) infected individuals infect ß s(t) I(t) other individuals in each round
  - A share of δ of all infected individuals is susceptible again
- Leads to the following recursive equations:
  - $I(t+1) = I(t) + ß\, i(t)\, S(t) - \delta\, I(t)$
  - $i(t+1) = i(t) + ß\, i(t)\, s(t) - \delta\, i(t)$
  - $S(t+1) = S(t) - ß\, i(t)\, S(t) + \delta\, I(t)$
  - $s(t+1) = s(t) - ß\, i(t)\, s(t) + \delta\, i(t)$

SI-Model

- i(t+1) = i(t) + ß i(t) s(t) – δ i(t)
- s(t+1) = s(t) – ß i(t) s(t) + δ i(t)
- Idea:
  - i(t) is a continuous function
  - i(t+1)-i(t) approximate first derivative

$$\frac{i(t+1)-i(t)}{1} \approx \frac{di(t)}{dt}$$

$$\frac{di(t)}{dt} = \beta \cdot i(t)(1-i(t)) - \delta i(t)$$

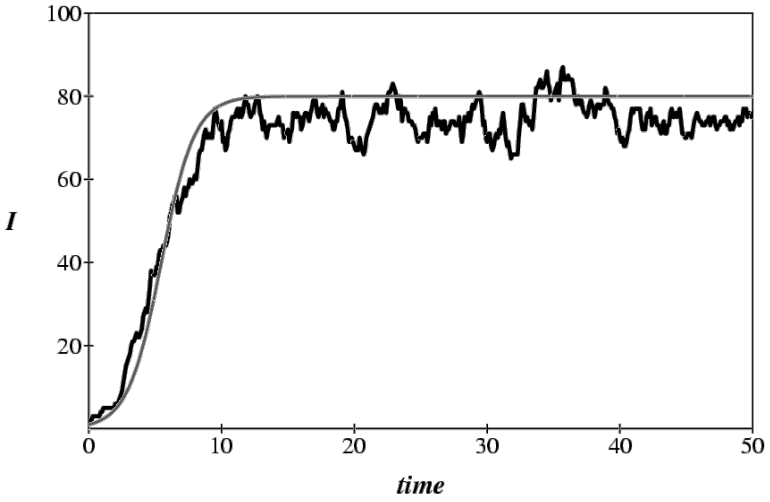
- Solution:
  - for $\rho = \frac{\delta}{\beta}$

$$i(t) = \frac{1-\rho}{1+\left(\frac{1-\rho}{i(0)}-1\right)e^{-(\beta-\delta)t}}$$

SIS-Model
Interpretation of Solution

$$i(t) = \frac{1-\rho}{1+\left(\frac{1-\rho}{i(0)}-1\right)e^{-(\beta-\delta)t}} \qquad \rho = \frac{\delta}{\beta}$$

- If ß < δ
  - then i(t) is strictly decreasing
- If ß > δ
  - then i(t) converges against $1 - \rho = 1 - \delta/ß$
- Same behavior in discrete model has been observed
  - [Kephart,White'94]



SIR-Model

- Variables
  - n: total number of individuals
    - remains constant
  - S(t): number of (healthy) susceptible individuals at time t
  - I(t): number of infected individuals
  - R(t): number or recovered individ.
- Relative shares
  - s(t) := S(t)/n
  - i(t) := I(t)/n
  - r(t) := R(t)/n
- At every time unit each individual contacts ß partners
- Assumptions:
  - Among ß contact partnres ß s(t) are susceptible
  - All I(t) infected individuals infect ß s(t) I(t) other individuals in each round
  - A share of δ of all infected individuals is immune (recovered) and never infected again
- Leads to the following recursive equations:
  - $I(t+1) = I(t) + ß\, i(t)\, S(t) - \delta\, I(t)$
  - $i(t+1) = i(t) + ß\, i(t)\, i(t) - \delta\, i(t)$
  - $S(t+1) = S(t) - ß\, i(t)\, S(t)$
  - $s(t+1) = s(t) - ß\, i(t)\, s(t)$
  - $R(t+1) = R(t) + \delta\, I(t)$
  - $r(t+1) = r(t) + \delta\, i(t)$

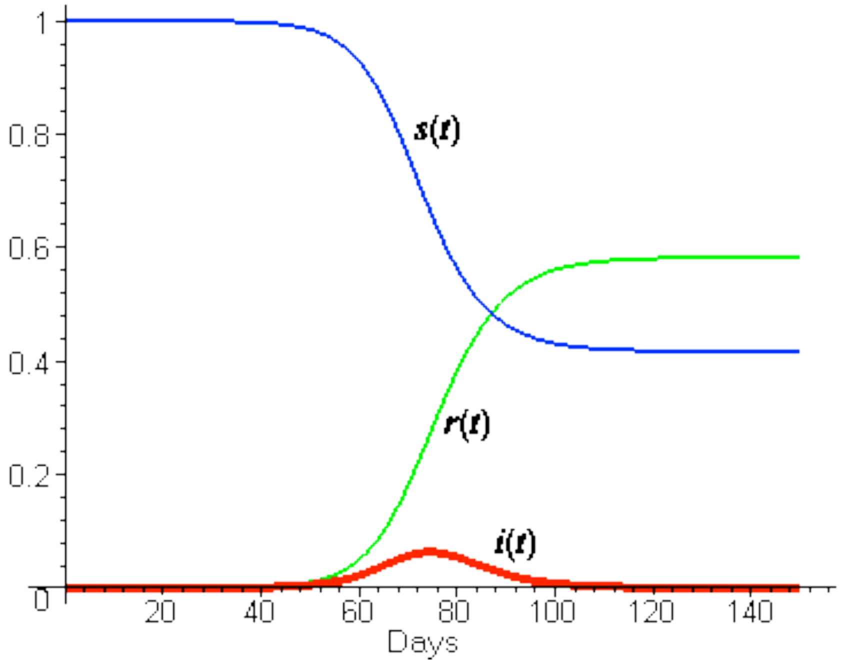
SIR-Model

- The equations and its differential equations counterpart
  - i(t+1) = i(t) + ß i(t) i(t) – δ i(t)
  - s(t+1) = s(t) – ß i(t) s(t)
  - r(t+1) = r(t) + δ i(t)
- No closed solution known
  - hence numeric solution
- Example
  - s(0) = 1
  - i(0) = 1,27 10-6
  - r(0) = 0
  - ß = 0,5
  - δ = 0,3333

$$\frac{ds(t)}{dt} = -\beta \cdot i(t)s(t)$$

$$\frac{di(t)}{dt} = \beta \cdot i(t)s(t) - \delta i(t)$$

$$\frac{dr(t)}{dt} = -\delta i(t)$$



Replicated Databases

- Same data storage at all locations
  - new entries appear locally
- Data must be kept consistently
- Algorithm is supposed to be decentral and robust
  - since connections and hosts are unreliable
- Not all databases are known to all
- Solutions
  - Unicast
    - New information is sent to all data servers
  - Problem:
    - not all data servers are known and can be reached
  - Anti-Entropy
    - Every local data server contacts another one and exchanges all information
    - total consistency check of all data
  - Problem
    - comunication overhead
- Epicast …

Epidemic Algorithms

- Epicast
  - new information is a rumor
  - as long the rumor is new it is distributed
  - Is the rumor old, it is known to all servers
- Epidemic Algorithm [Demers et al 87]
  - distributes information like a virus
  - robust alternative to BFS or flooding
- Communication method
  - Push & Pull, d.h. infection after $\log_3 n + O(\log \log n)$ rounds with high probability
- Problem:
  - growing number of infections increases comunication effort
  - trade-off between robustness and communication overhead

SI-Model for Graphs

- Given a contact graph G=(V,E)
  - n: number of nodes
  - I(t) := number of infected nodes in round t
  - i(t) = I(T)/n
  - S(t) := number of susceptible nodes in round t
    - I(t)+S(t)=n
  - s(t) = S(T)/n
- Infection:
  - If u is infected in round t and (u,v) $\in$ E, then v is infected in round t+1
- Graph determines epidemics
- Complete graph:
  - 1 time unit until complete infection
- Line graph
  - n-1 time units until complete infection

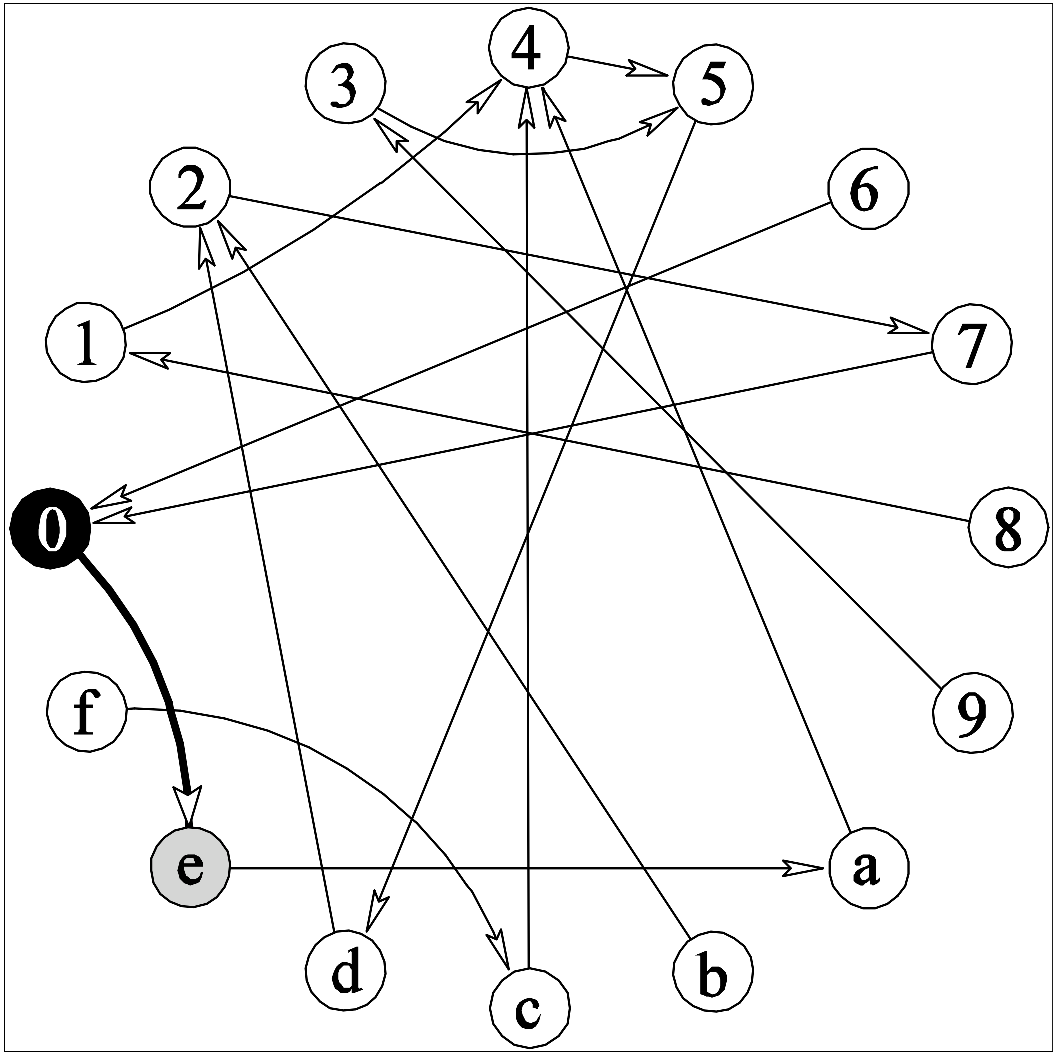
Epidemics in Static Random Graphs

- Random graph $G_{n,p}$
  - n nodes
  - Each directed edge occurs with independent probability p
- Expected indegre $\gamma = p\,(n-1)$
- How fast does an epidemic spread in $G_{n,p}$, if $\gamma \in O(1)$?
- Observation für n>2:
  - With probability $\geq 4^{-\gamma}$ and $\leq e^{-\gamma}$
    - a node has in-degree 0 and cannot be infected
    - a node has out-degree 0, and cannot infect others
- Implications:
  - Random (static) graph is not a suitable graph for epidemics

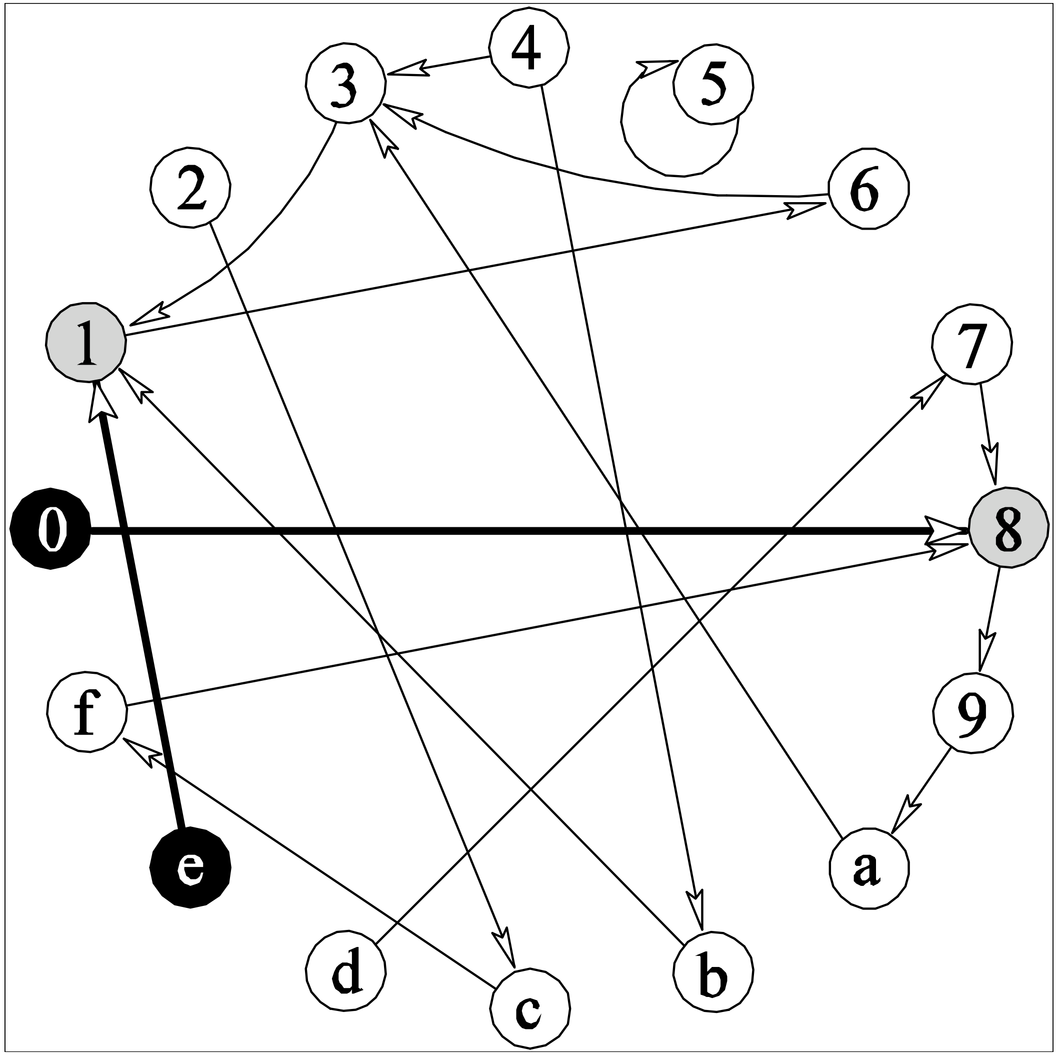
Random Call Model

- In each round a new contact graph $G_t=(V,E_t)$:
  - Each node in G_t has out-degree 1
    - chooses random node v out of V
- Infection models:
  - Push-Model
    - if u is infected and $(u,v) \in E_t$, then v is infected in the next round
  - Pull-Modell:
    - if v is infected and $(u,v) \in E_t$, then u is infected in the next round

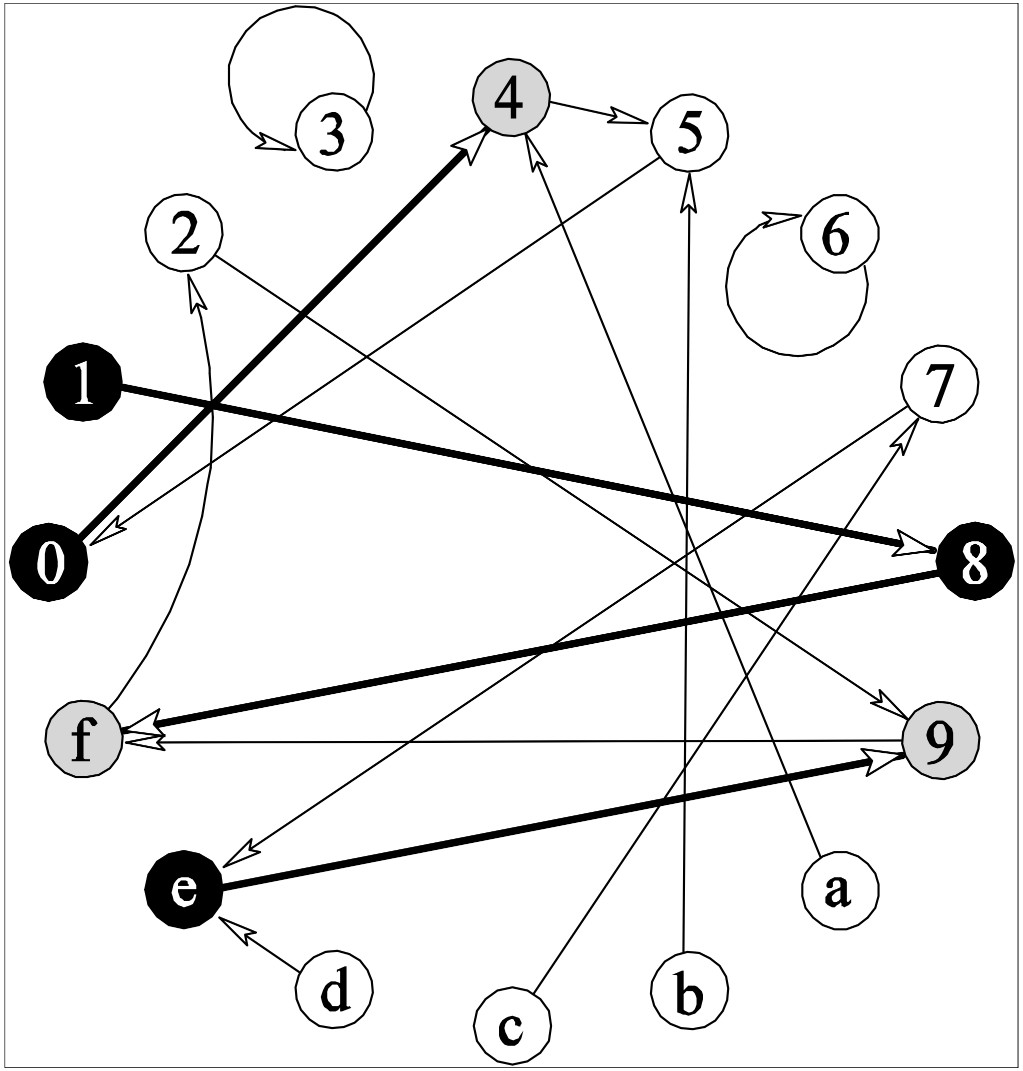
Push Model



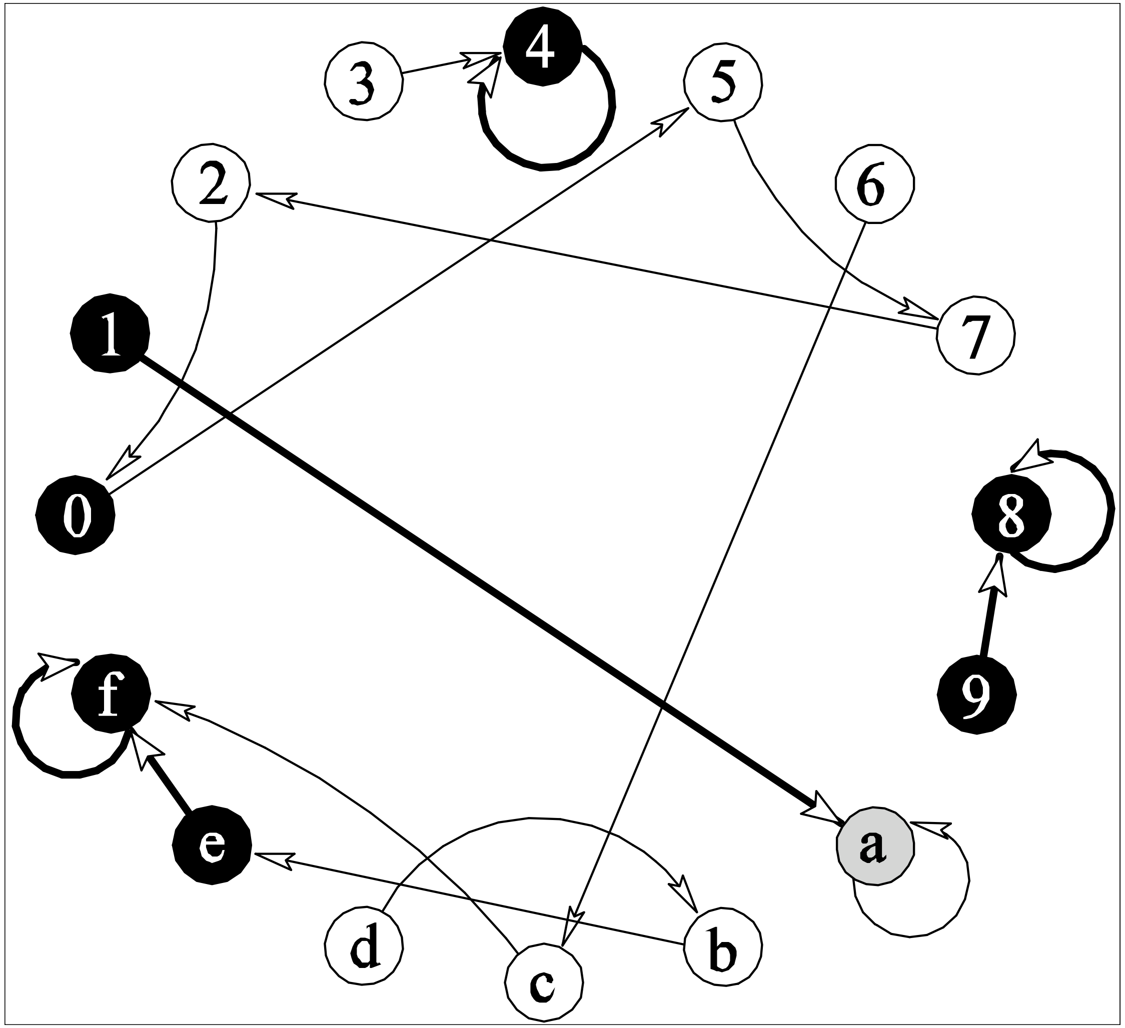
Push Model



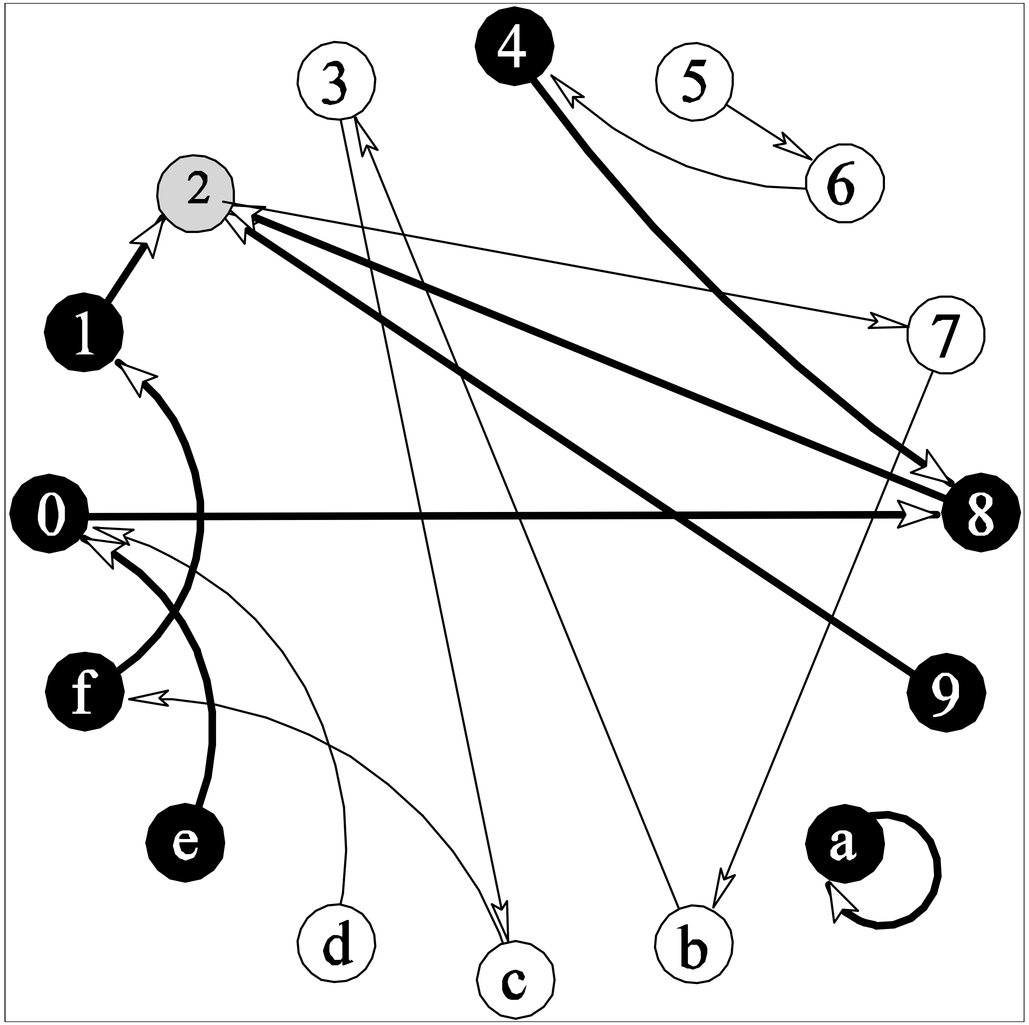
Push Model



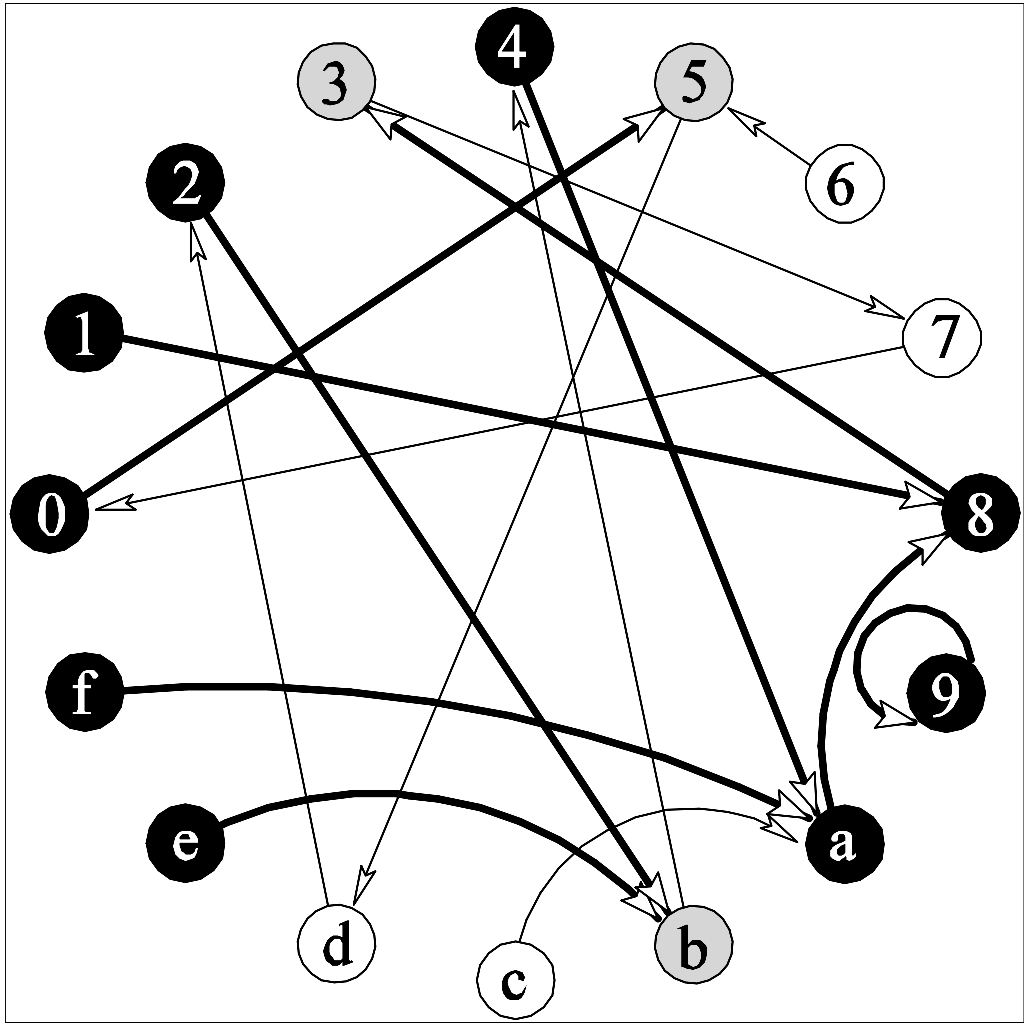
Push Model



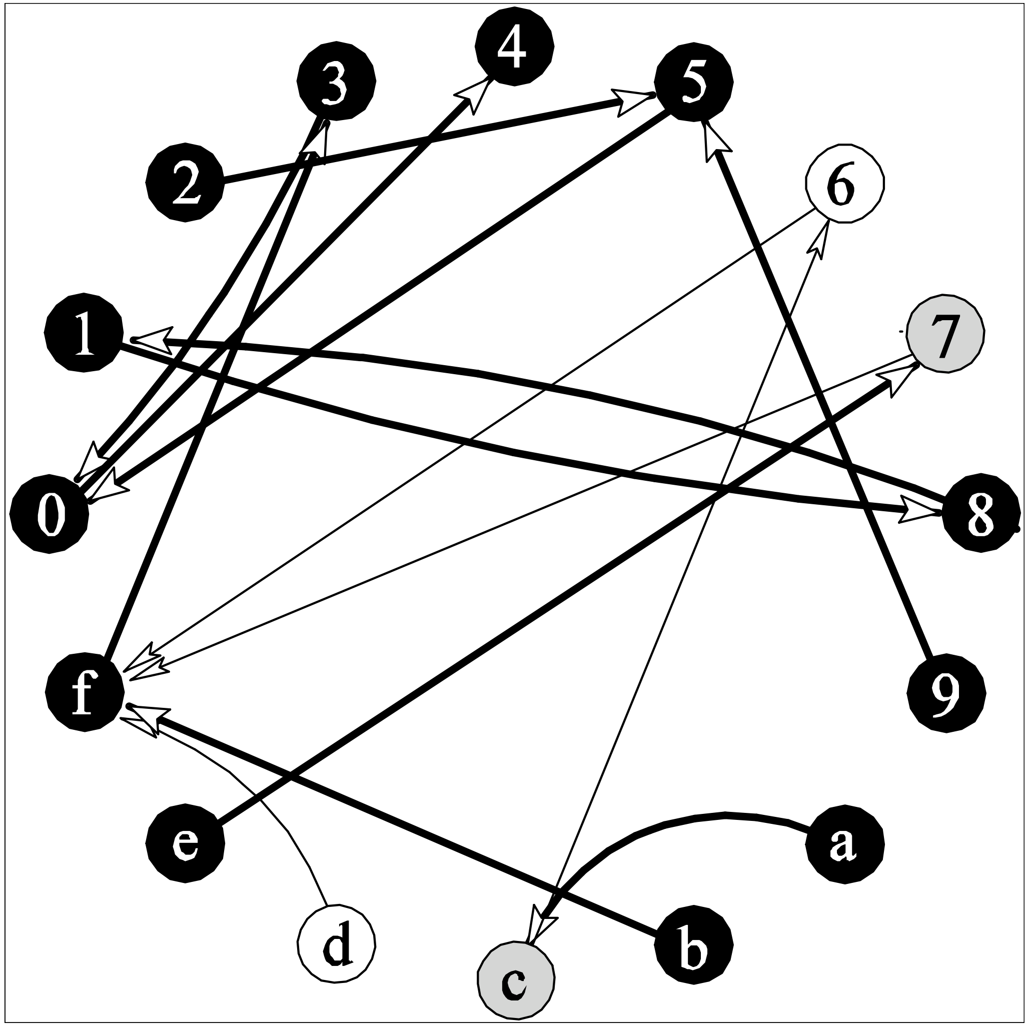
Push Model



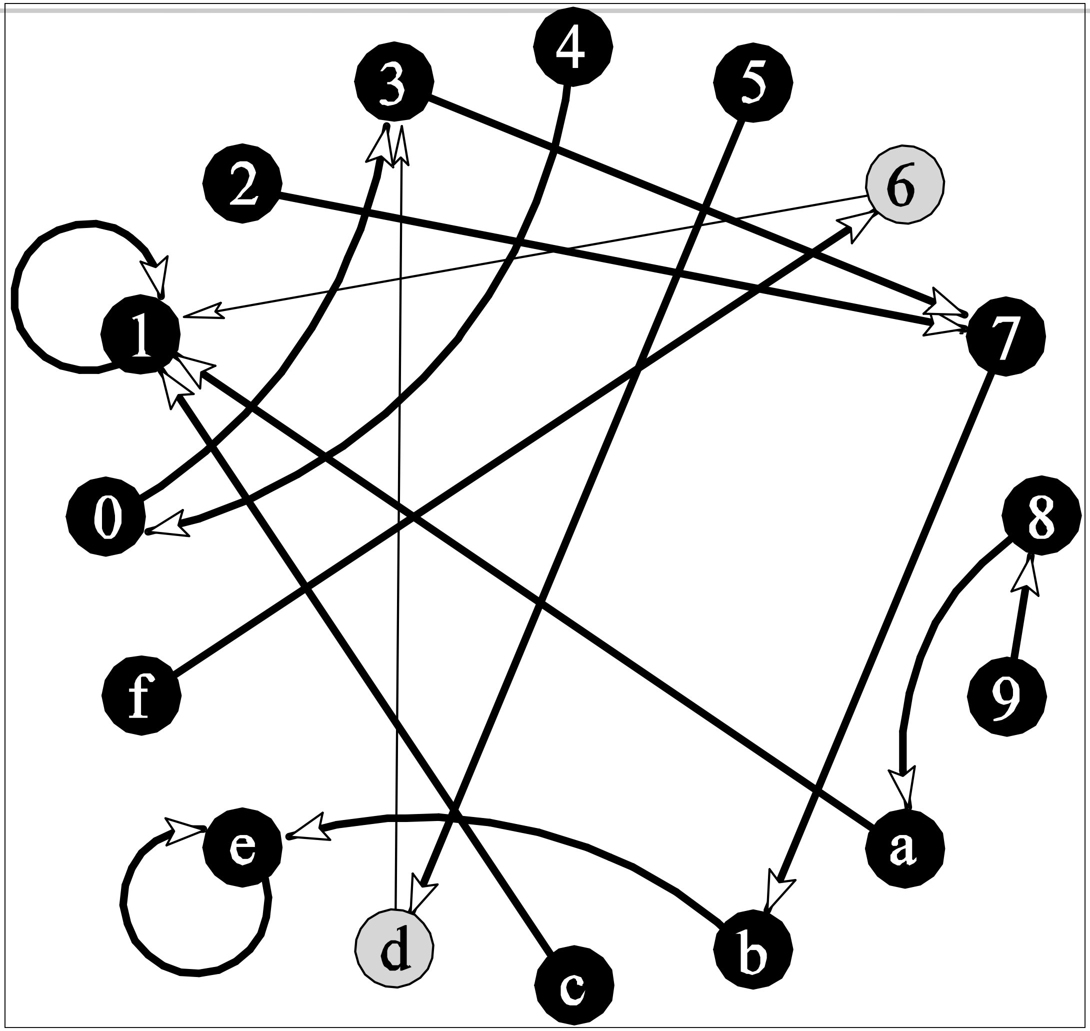
Push Model



Push Model



Push Model



Push Model
Start Phase

- 3 cases for an infected node
  1. it is the only one infecting a new node
  2. it contacts an already infected node
  3. it infects together with other infected nodes a new node
     - this case is neglected in the prior deterministic case

  - Probability for 1st or 3rd case $s(t) = 1-i(t)$
  - Probability for 2nd case $i(t)$
  - Probability for 3rd case is at most $i(t)$
    - since at most $i(t)$ are infected
- Probability of infection of a new node, if $i(t) \leq s(t)/2$:
  - at least $1 - 2i(t)$
- $E[i(t+1)] \geq i(t) + i(t)(1 - 2\, i(t)) = 2i(t) - 2i(t)^2 \approx 2\, i(t)$

Push Model
Start phase & Exponential Growth

- If i(t) ≤ s(t)/2:
  - E[i(t+1)] ≥ 2 i(t) – 2i(t)2 ≈ 2 i(t)
- Start phase: I(t) ≤ 2 c (ln n)^2
  - Variance of i(t+1) relatively large
  - Exponential growth starts after some O(1) with high probability
- Exponential growth: I(t) ∈ [2 c (ln n)^2, n/(log n)]
  - Nearly doubling of infecting nodes with high probability, i.e. 1-O(n^{-c})

- Proof by Chernoff-Bounds
  - For independent random variables X$_i$∈{0,1} with $X_m = \sum_{i=1}^m X_i$
  - and any $0 \leq \delta \leq 1$

$$\mathrm{P}[X_m \leq (1-\delta)\mathbf{E}[X_m]] \ \leq \ e^{-\delta^2\mathbf{E}[X_m]/2}$$

  - Let δ = 1/(ln n)
  - **E**[X$_m$] ≥ 2 c (ln n)^3
  - Then δ^2 **E**[X$_m$] /2 ≥ c ln n
  - This implies

$$\mathrm{P}[X_m \leq (1-\delta)\mathbf{E}[X_m]] \ \leq \ e^{-\delta^2\mathbf{E}[X_m]/2} \ \leq n^{-c}$$

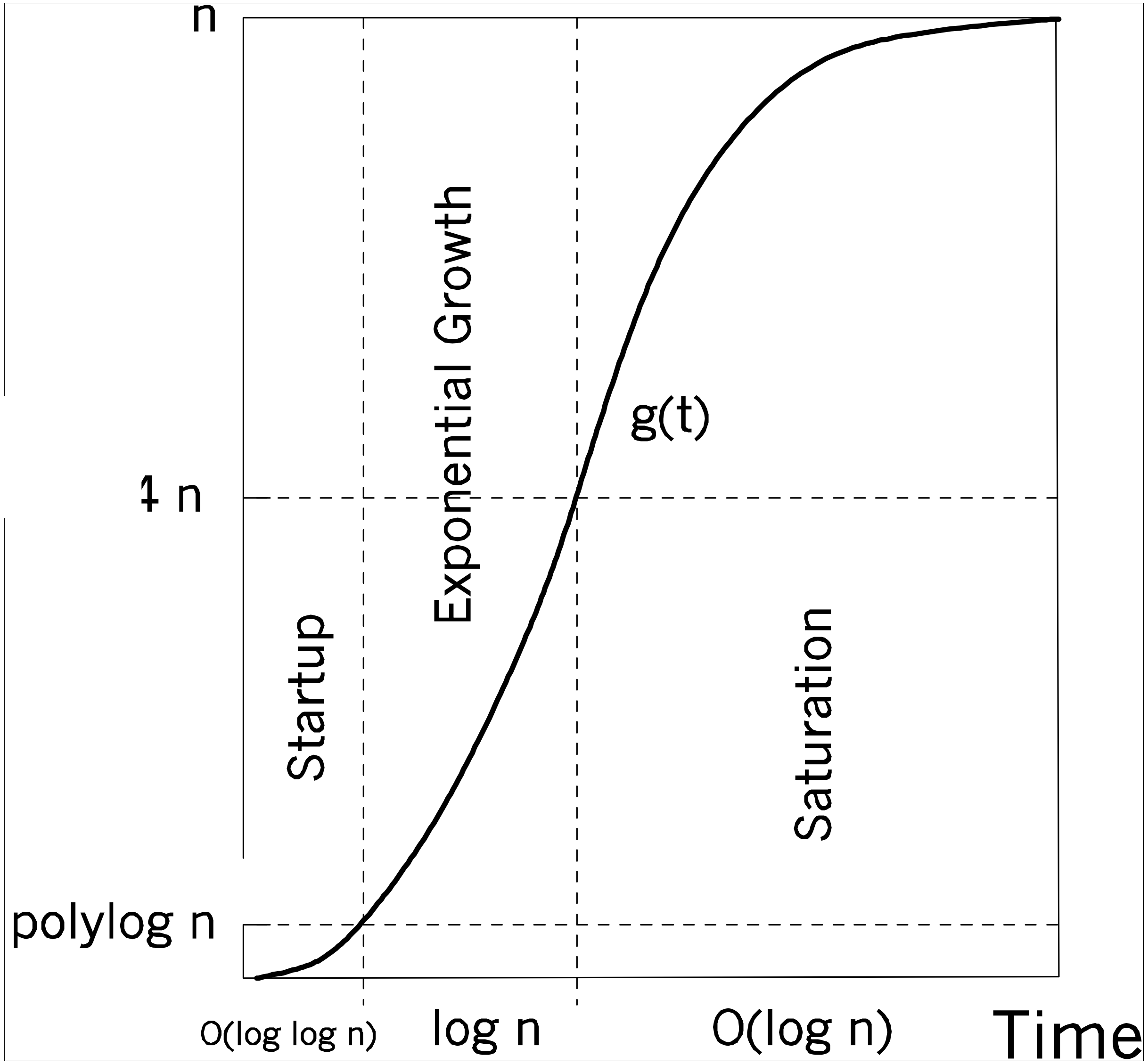
Push Model
Middle Phase & Saturation

- Probability of infections of a new node if $i(t) \leq s(t)/2$: $1 - 2i(t)$
  - $\mathbf{E[i(t+1)] \geq 2\, i(t) - 2i(t)^2 \approx 2\, i(t)}$
- Middle phase $\mathbf{I(t) \in [n/(\log n), n/3]}$
  - term $\mathbf{2i(t)^2} \geq \mathbf{2i(t)/(\log n)}$ cannot be neglected anymore
  - Yet, $\mathbf{2i(t) - 2i(t)^2} \geq \mathbf{4/3\, i(t)}$ still implies expontential growth, but with base < 2
- **Saturation:** $\mathbf{I(t)} \geq \mathbf{n/3}$
  - Probability that a susceptible node is not contacted by $I(t) = c\, n$ infected nodes:

$$\left(1-\frac{1}{n}\right)^{cn} = \left(\left(1-\frac{1}{n}\right)^{n}\right)^{c} \leq \frac{1}{e^c}$$

- This implies a constant probability for infection $\geq$ $\mathbf{1 - e^{-1/3}}$ und $\leq$ $\mathbf{1 - e^{-1}}$
- Hence
  $\mathbf{E[s(t+1)]} \leq \mathbf{e^{-i(t)}\, s(t)} \leq \mathbf{e^{-1/3}\, s(t)}$
- Chernoff-bounds imply that this holds with high probability
- Exponential shrinking of susceptible nodes
- Base converges to 1/e

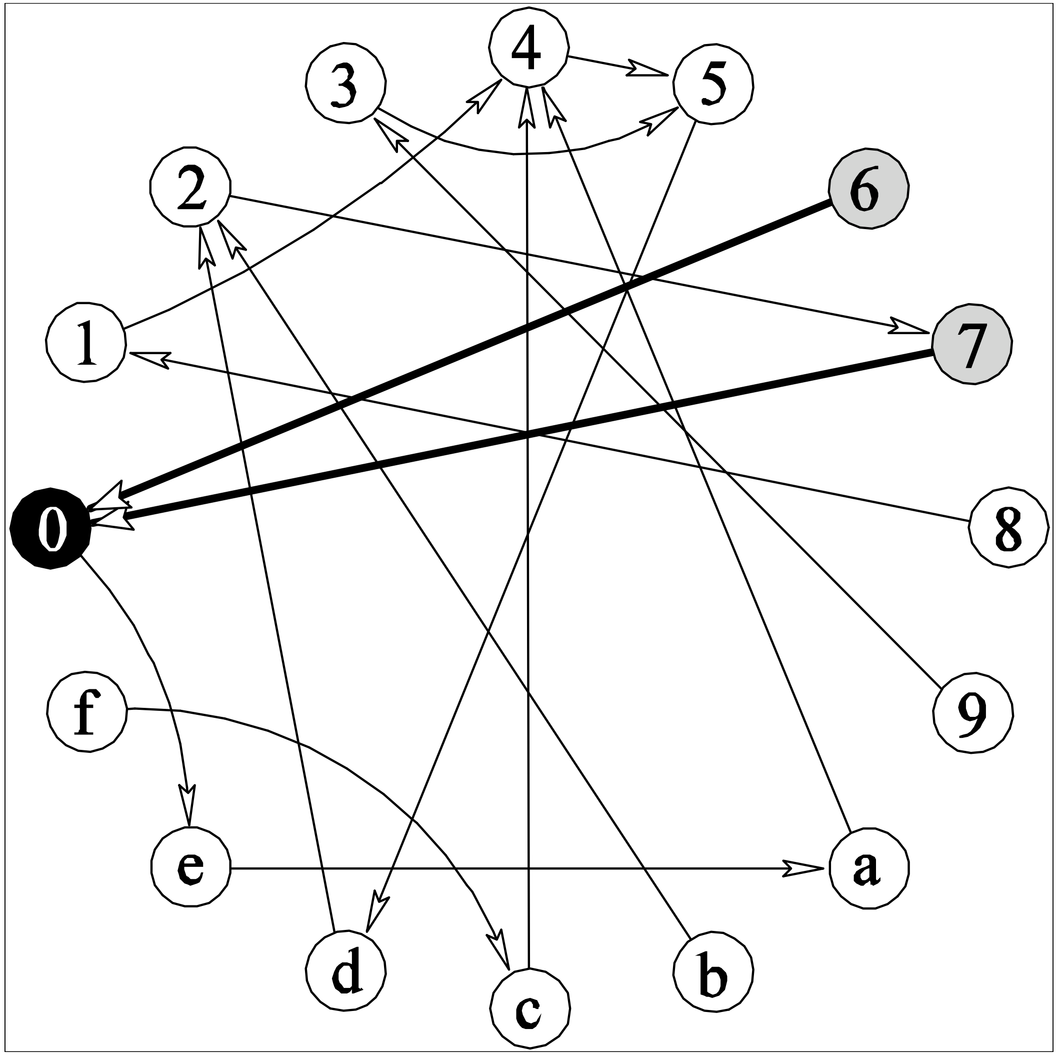
Push Model


n
1 n
polylog n
Startup
Exponential Growth
Saturation
g(t)
O(log log n)
log n
O(log n)
Time

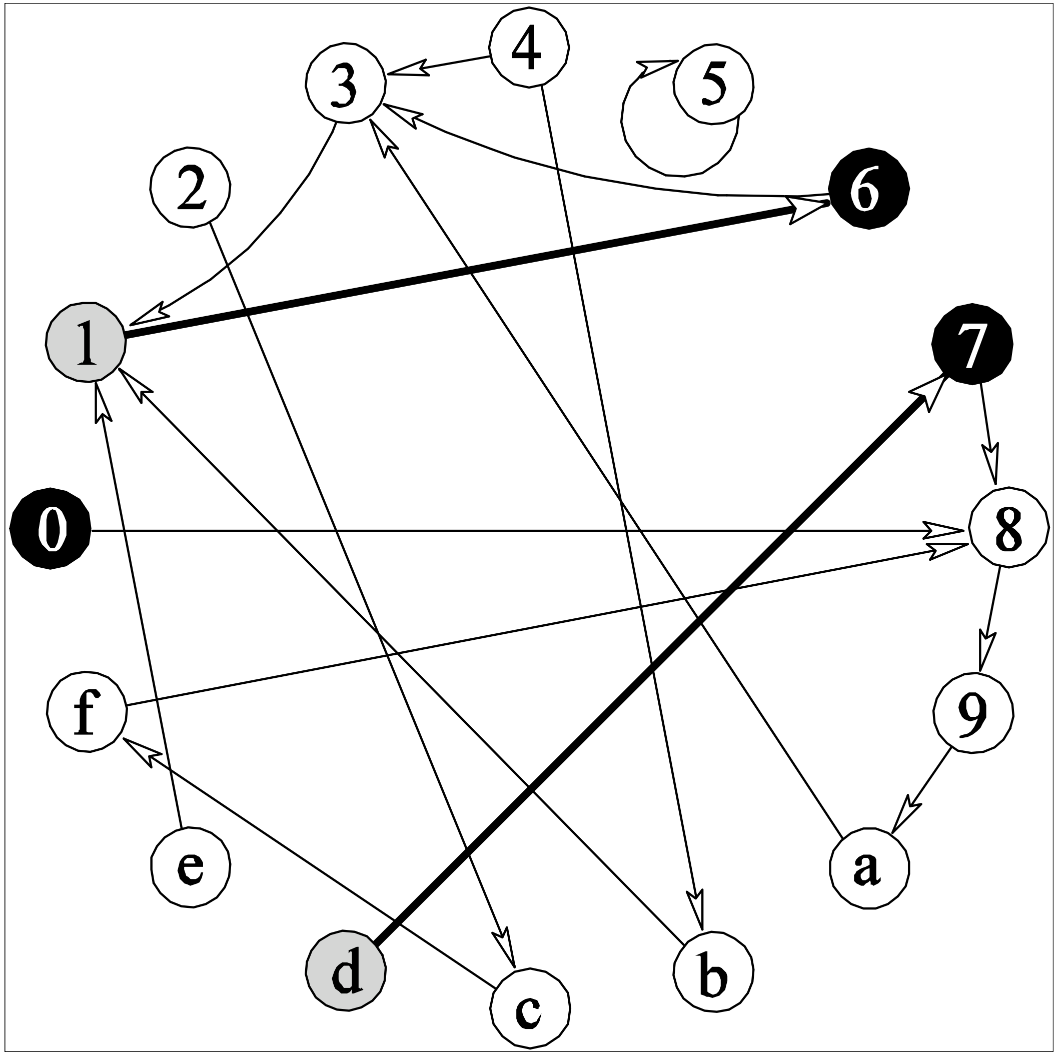
Random Call Model

- Infection models:
  - Push Model
    - if u is infected and $(u,v) \in E_t$, then v is infected in the next round
  - Pull Model
    - if v is infected and $(u,v) \in E_t$, then u is infected in the next round

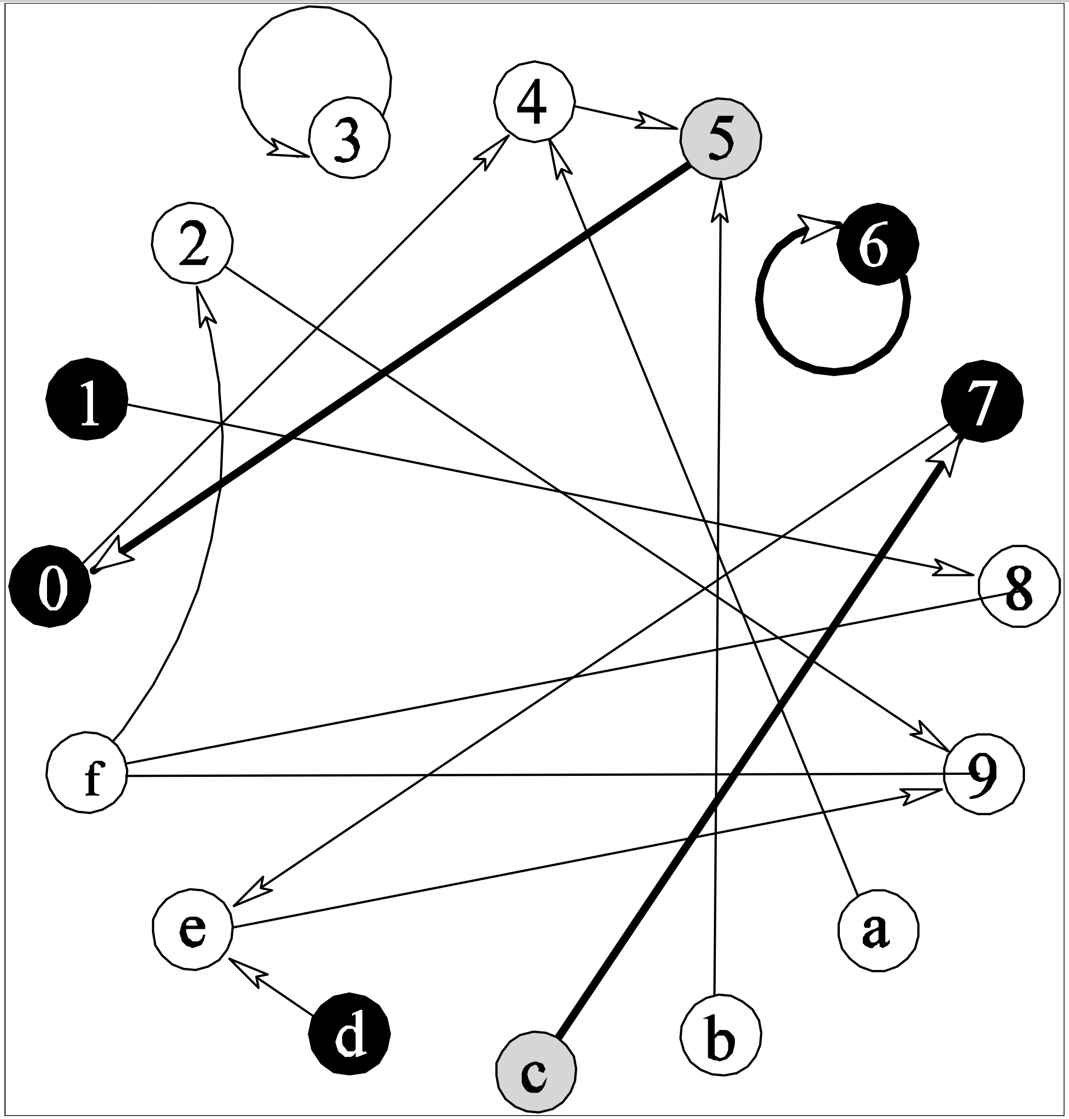
Pull Model



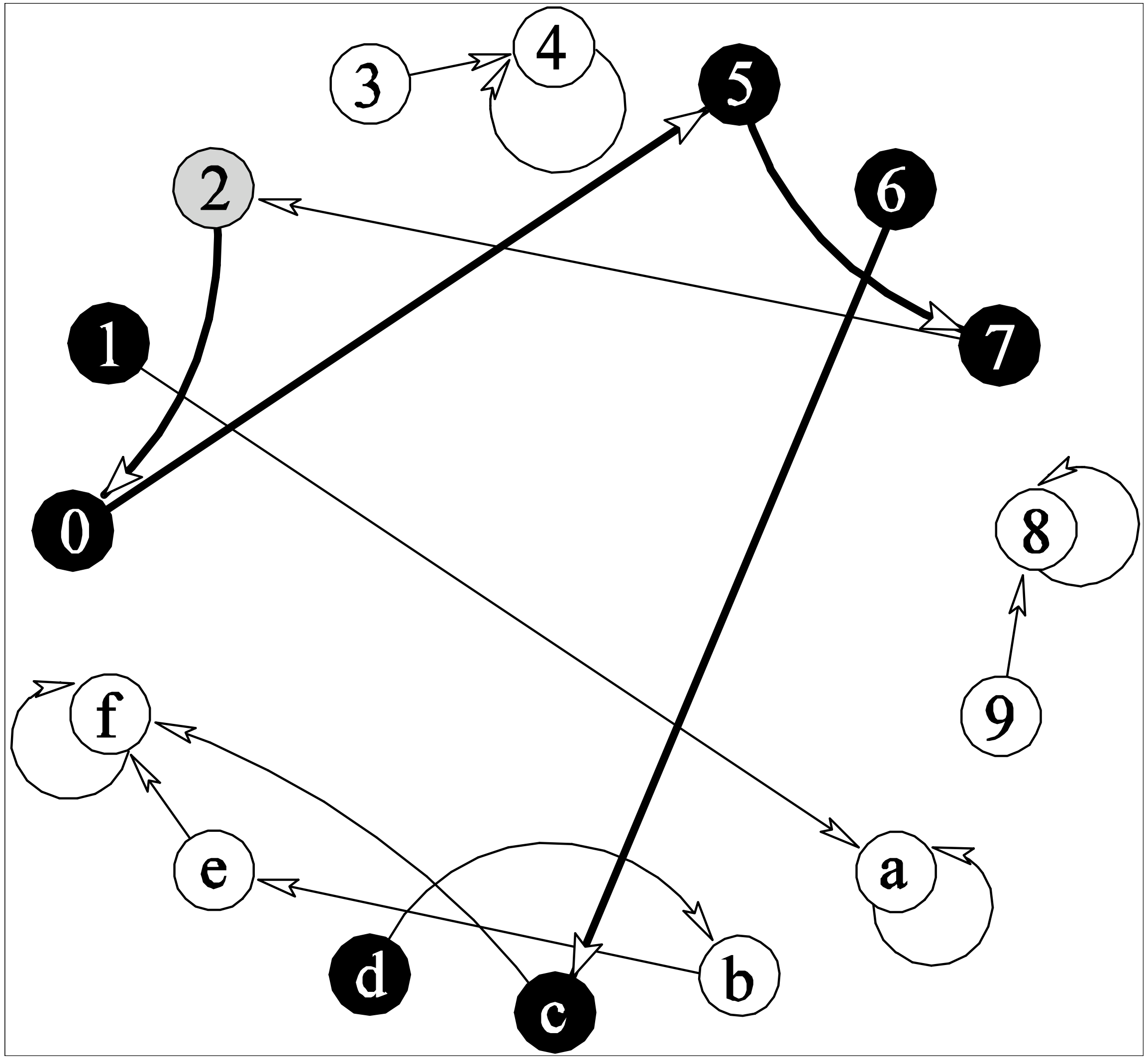
Pull Model



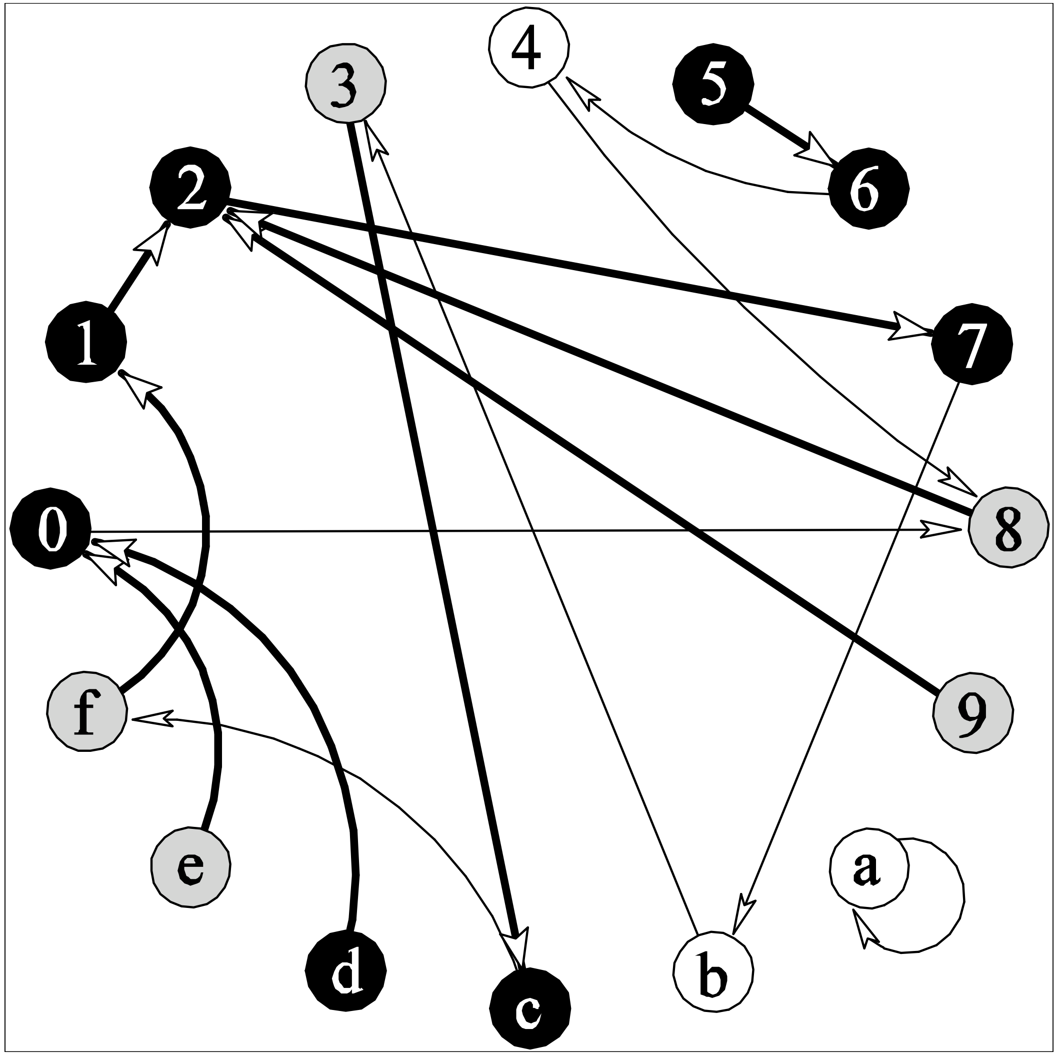
Pull Model



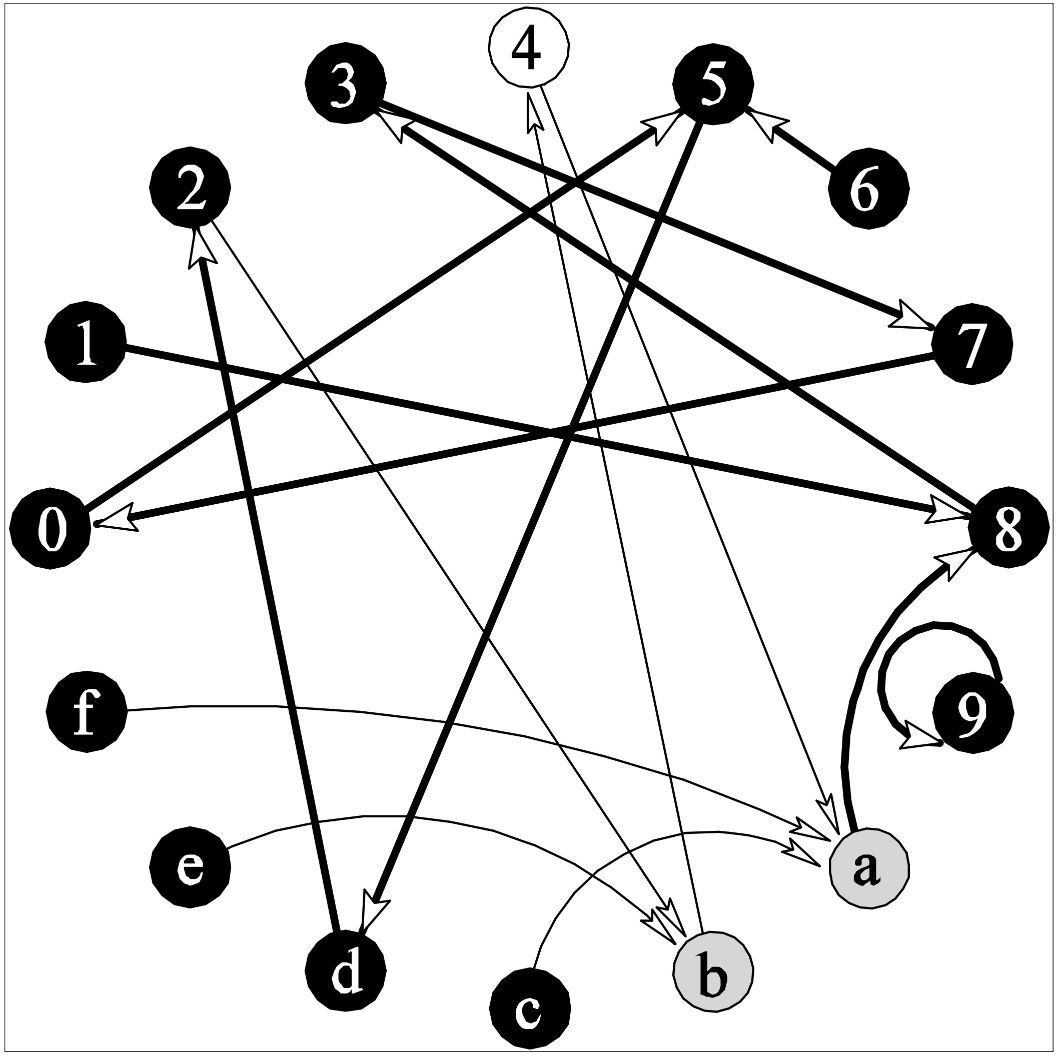
Pull Model



Pull Model



Pull Model

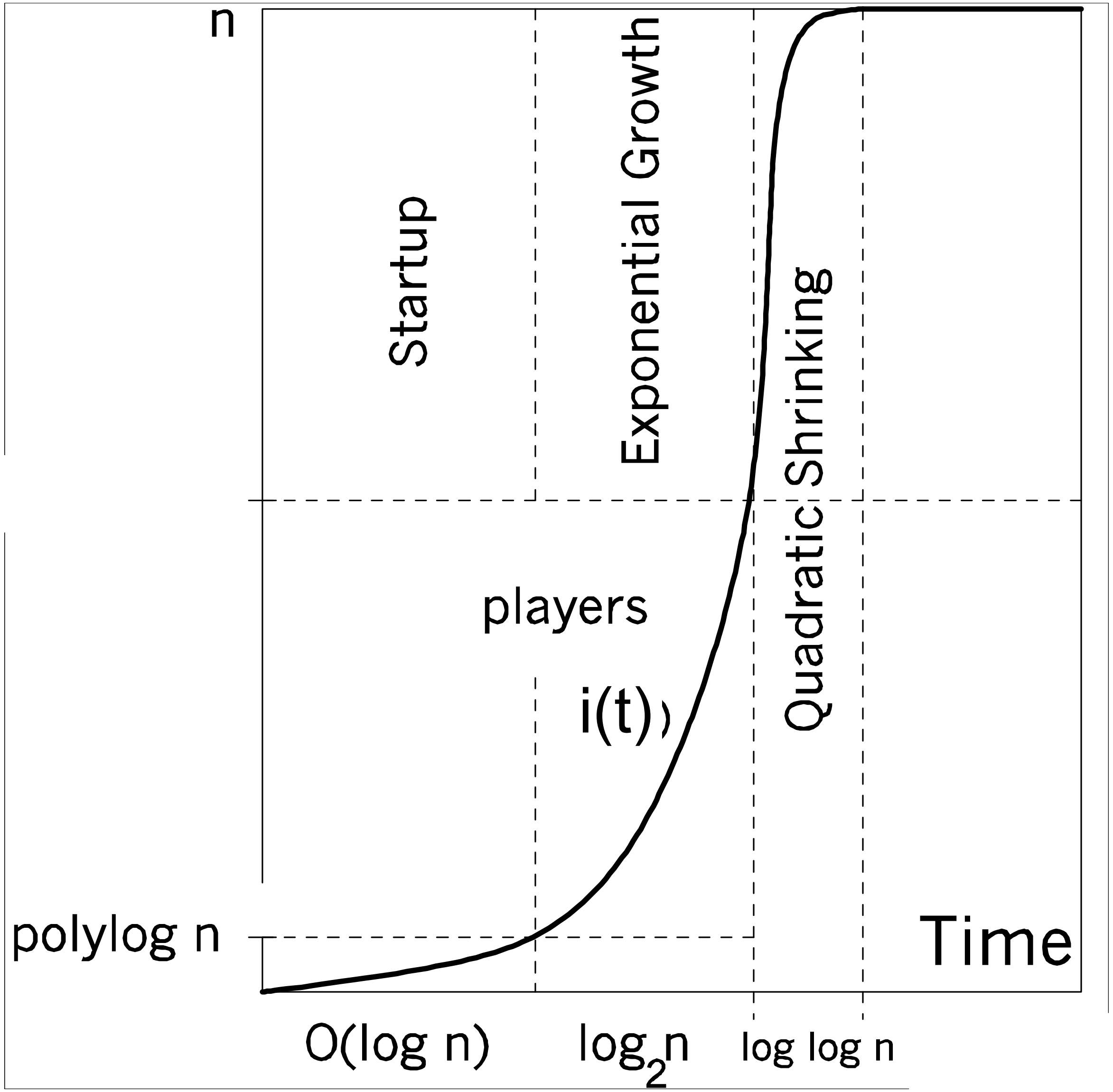


Pull Model

- Consider
  - an susceptible node and I(t) infected nodes
- Probability that a susceptible node contacts an infected node: i(t)
  - E[s(t+1)]
$= s(t) – s(t)\, i(t)$
$= s(t)\, (1 – i(t)) = s(t)^2$
  - E[i(t+1)]
$= 1\text{-}s(t)^2$
$= 1 – (1 – i(t))^2$
$= 2\, i(t) – i(t)^2 \approx 2\, i(t)$
    - for small i(t)

- Problem
  - if $i(t) \leq (\log n)^2$ then exponential growth is not with high probability
  - O(log n) steps are needed to start eh growth with high probability
    - yet in the expectation it grows exponentially
- After this phase
  - If $s(t) \leq ½$
    - then the share of susceptible nodes is squared in each step
  - This implies $E[s(t+ O(\log \log n))] = 0$,
  - If $i(t) \geq ½$ then after O(log log n) steps all nodes are infected with high probability

Pull Model

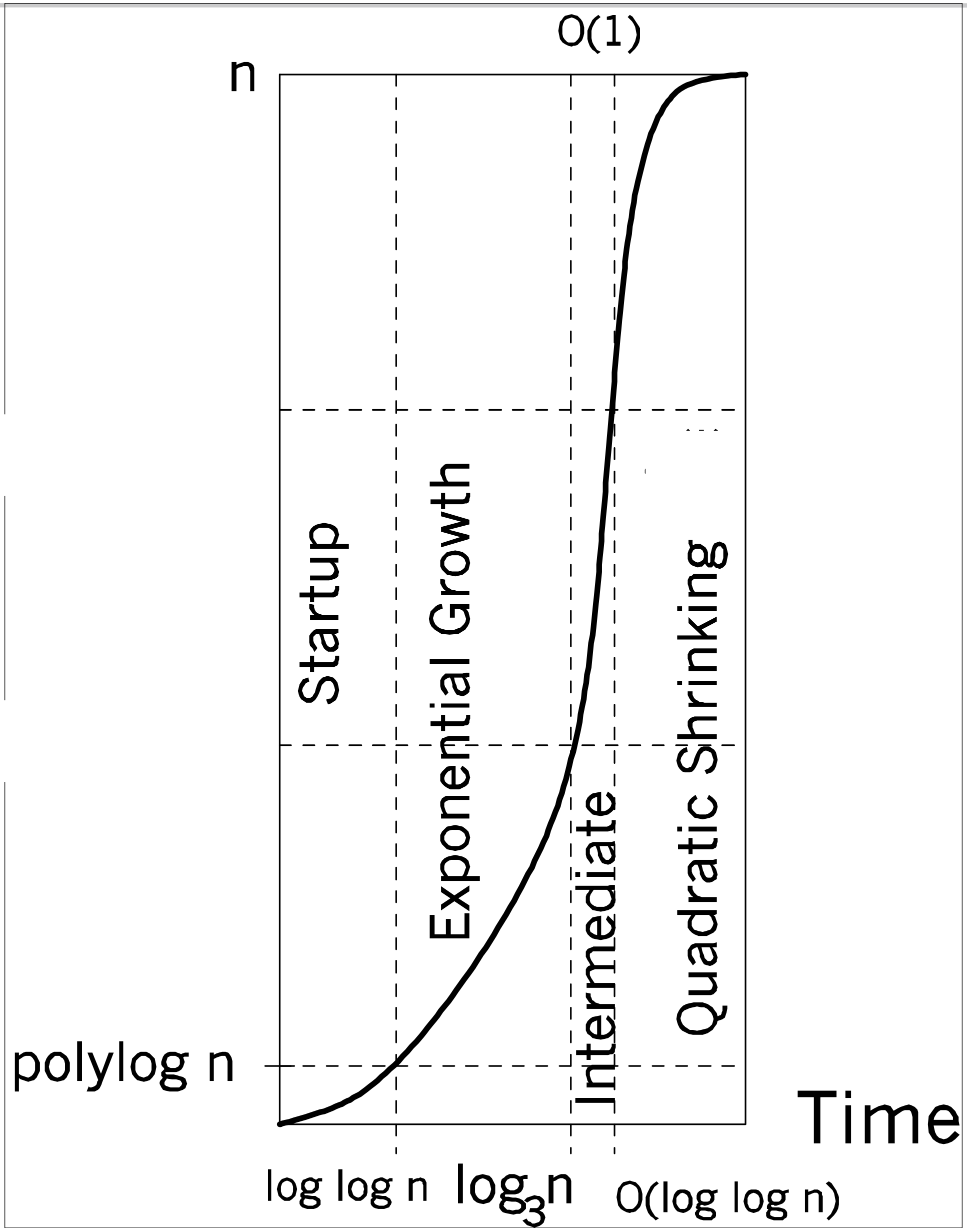


n
Startup
Exponential Growth
Quadratic Shrinking
players
i(t)
polylog n
Time
O(log n)
$\log_2 n$
log log n

Push&Pull Model

- Combines growth of Push and Pull
- Start phase: $i(t) \leq 2\ c\ (\ln n)^2$
  - Push causes doubling of $i(t)$ after $O(1)$ rounds with high probability
- Exponential growth: $I(t) \in [2\ c\ (\ln n)2, n/(\log n)]$
  - Push and Pull nearly triple in each round with high probability:
    - $i(t+1) \geq 3\ (1-1/(\log n))\ i(t)$
- Middle phase: $I(t) \in [n/(\log n), n/3]$
  - Push and Pull
    - slower exponential growth
- Quadratic shrinking: $I(t) \geq n/3$
  - caused by Pull:
  - $E[s(t+1)] \leq s(t)^2$
  - The Chernoff bound implies with high probability
  - $s(t+1) \leq 2\ s(t)^2$
  - so after two rounds for $s(t) \leq 1/2^{1/2}$
    - $s(t+2) \leq s(t)^2$ w.h.p.

Push&Pull Model



Max-Counter Algorithm

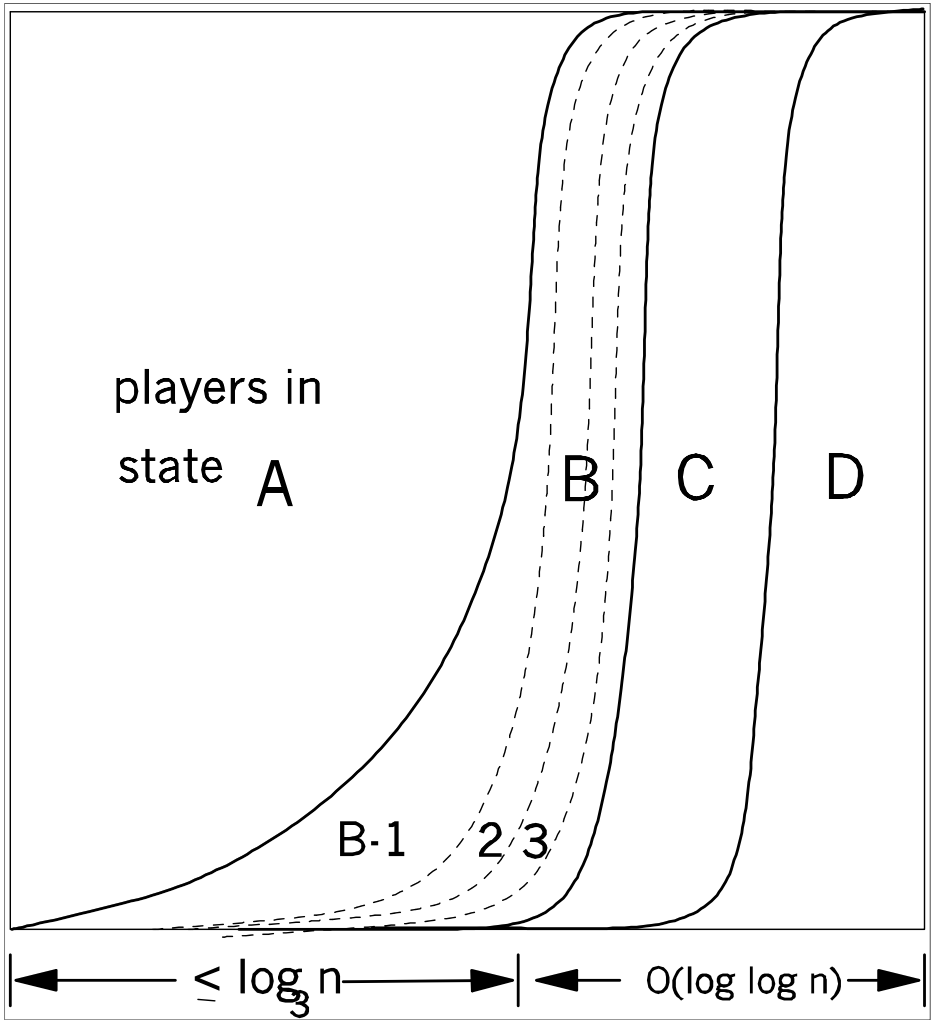
- Simple termination strategy
  - If the rumor is older than max_{ctr}, then stop transmission
- Advantages
  - simple
- Disadvantage
  - Choice of max_{ctr} is critical
  - If max_{ctr} is too small then not all nodes are informed
  - If max_{ctr} is too large, then the message overhead is $\Omega(n\ max_{ctr})$
- Optimal choice for push-communication
  - $max_{ctr} = O(\log n)$
  - Number of messages: $O(n \log n)$
- Pull communication
  - $max_{ctr} = O(\log n)$
  - Number of messages: $O(n \log n)$
- Push&Pull communication
  - $max_{ctr} = \log_3 n + O(\log \log n)$
  - Number of messages: $O(n \log \log n)$

Shenker‘s Min-Counter Algorithm

- Only is the rumor is seen as old then contact partners increase the age-counter
- Shenkers Min-Counter-Algorithmus für $\max_{ctr} = O(\log\log n)$
  - Every player P stores age-variable $ctr_R(P)$ for each rumor R
  - A: player P does not know the rumor:
    - $ctr_R(P) \leftarrow \mathbf{1}$
  - B: If player P sees rumor for the first time
    - $ctr_R(P) \leftarrow 1$
  - B: If partners $Q_1, Q_2, \ldots, Q_m$ communicate with P in a round
  - If $\min_i\{ctr_R(Q_i)\} \geq ctr_R(P)$ then
    - $ctr_R(P) \leftarrow ctr_R(P) + 1$
  - C: If $ctr_R(P) \geq \max_{ctr}$ then
    - tell the rumor for $\max_{ctr}$ more rounds
    - then D: stop sending the rumor
- Theorem
  - Shenkers Min-Counter algorithms informs all nodes using Push&Pull-communication in $\log_3 n + O(\log\log n)$ rounds with probability $1-n^{-c}$, using at most $O(n \log\log n)$ messages.

Shenker‘s Min-Counter-Algorithm

- Theorem
  - Shenkers Min-Counter algorithms informs all nodes using Push&Pull-communication in $\log_3 n + O(\log \log n)$ rounds with probability $1-n^{-c}$, using at most $O(n \log \log n)$ messages.



Peer-to-Peer Networks

17 Kelips and Epidemic Algorithms

Christian Schindelhauer

Technical Faculty

Computer-Networks and Telematics

University of Freiburg