Censorship Resistant Peer-to-Peer Content Addressable Networks

the idea of A. Fiat and J. Saia

Pascal Minnerup

Ferienakademie im Sarntal 2008
FAU Erlangen-Nürnberg, TU München, Uni Stuttgart

September 2008

Overview



Overview



Why are Peer-to-Peer networks being attacked?

Security Threats

Why Peer-to-Peer networks are being attacked

Why are Peer-to-Peer networks being attacked?

- Legal attacks (because of copyright violation)

Security Threats

Why Peer-to-Peer networks are being attacked

Why are Peer-to-Peer networks being attacked?

- Legal attacks (because of copyright violation)
  - Napster file sharing system



Security Threats

Why Peer-to-Peer networks are being attacked

Why are Peer-to-Peer networks being attacked?

- Legal attacks (because of copyright violation)
  - Napster file sharing system
- Totalitarian regimes hide dissident information



Security Threats

Why Peer-to-Peer networks are being attacked

Why are Peer-to-Peer networks being attacked?

- Legal attacks (because of copyright violation)
  - Napster file sharing system
- Totalitarian regimes hide dissident information
- Reputation



Security Threats

Why Peer-to-Peer networks are being attacked

Why are Peer-to-Peer networks being attacked?

- Legal attacks (because of copyright violation)
  - Napster file sharing system
- Totalitarian regimes hide dissident information
- Reputation
- Random Faults



Types of Attack

- Who attacks?
- How does he attack?



Types of Attack

- Who attacks?
  - Random
  - Enemy
- How does he attack?



Types of Attack

- Who attacks?
  - Random
  - Enemy
- How does he attack?
  - Disabling some Peers
  - Sending wrong messages



Overview



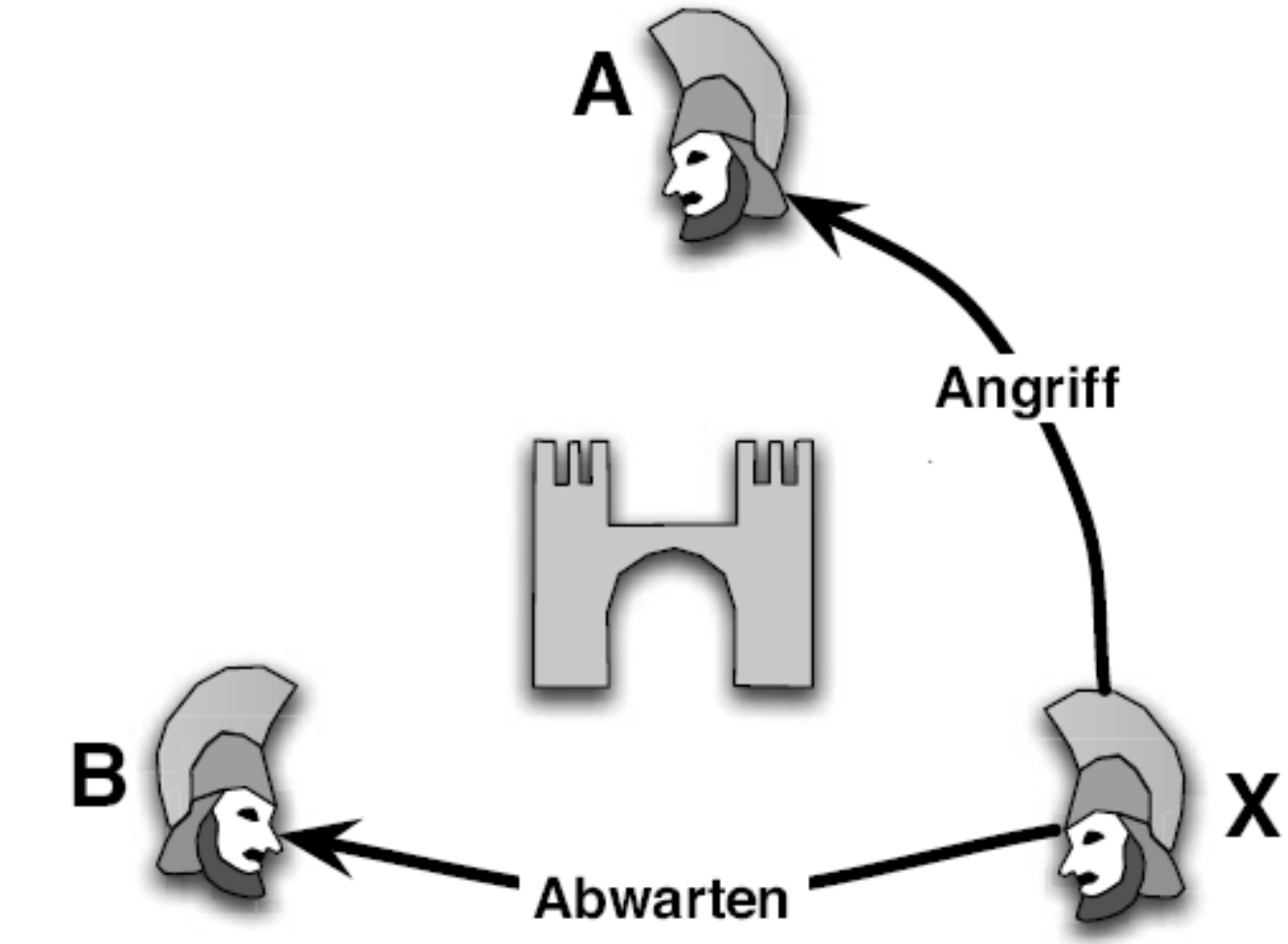
Byzantine Generals

- Model problem for peer-to-peer networks
- Refers to generals of the eastern roman empire
- Some generals wanted to become emporer themselves and therefore did not obey their orders
- Three generals siege a town
- If they want to win, at least two generals have to do the same: wait or attack
- They have to agree what to do
- One general deserted to the enemy town



Byzantine Faults

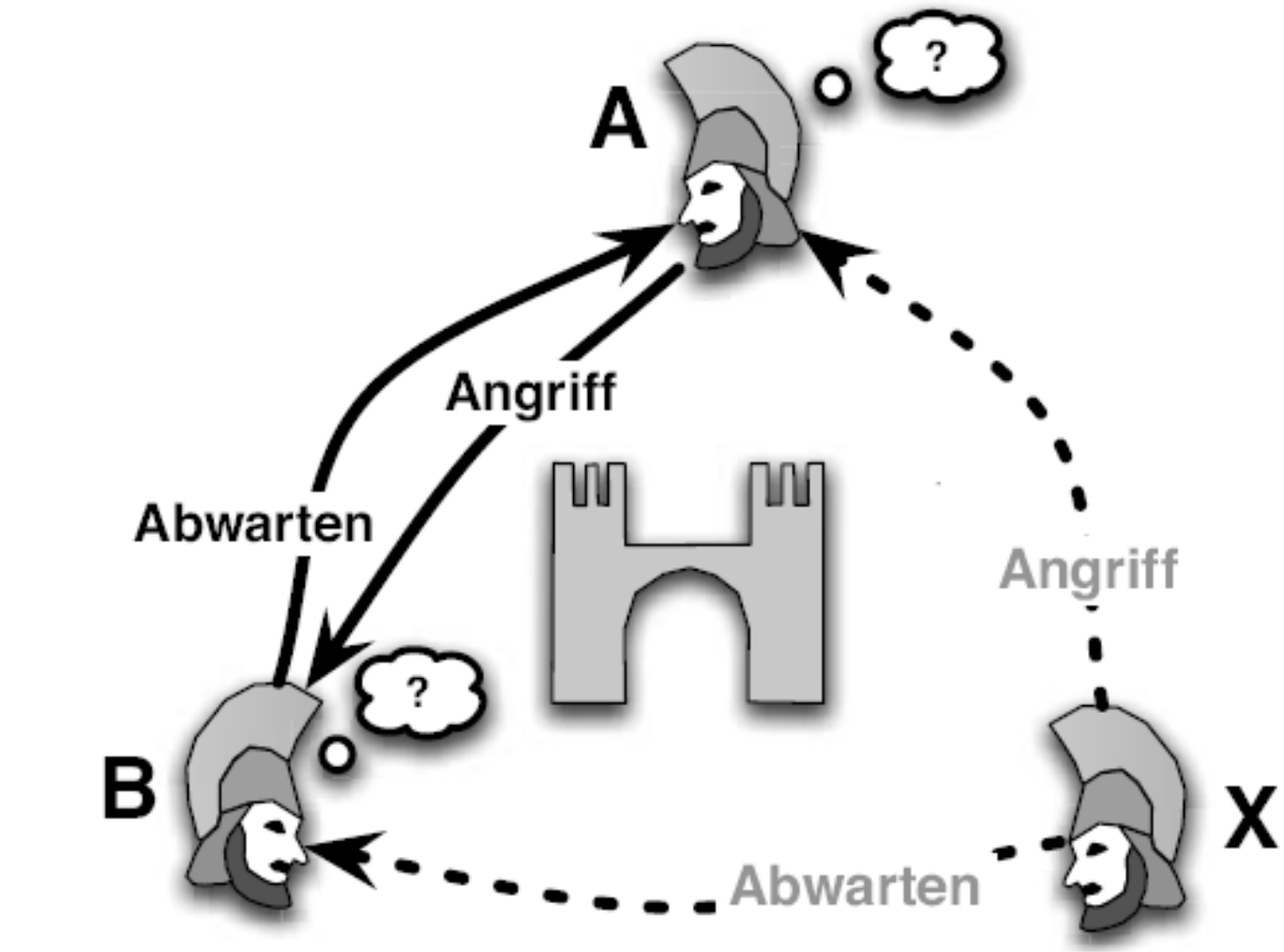
3 Gererals



Source: Christian Schindelhauer - Peer-to-Peer-Netzwerke

Byzantine Faults

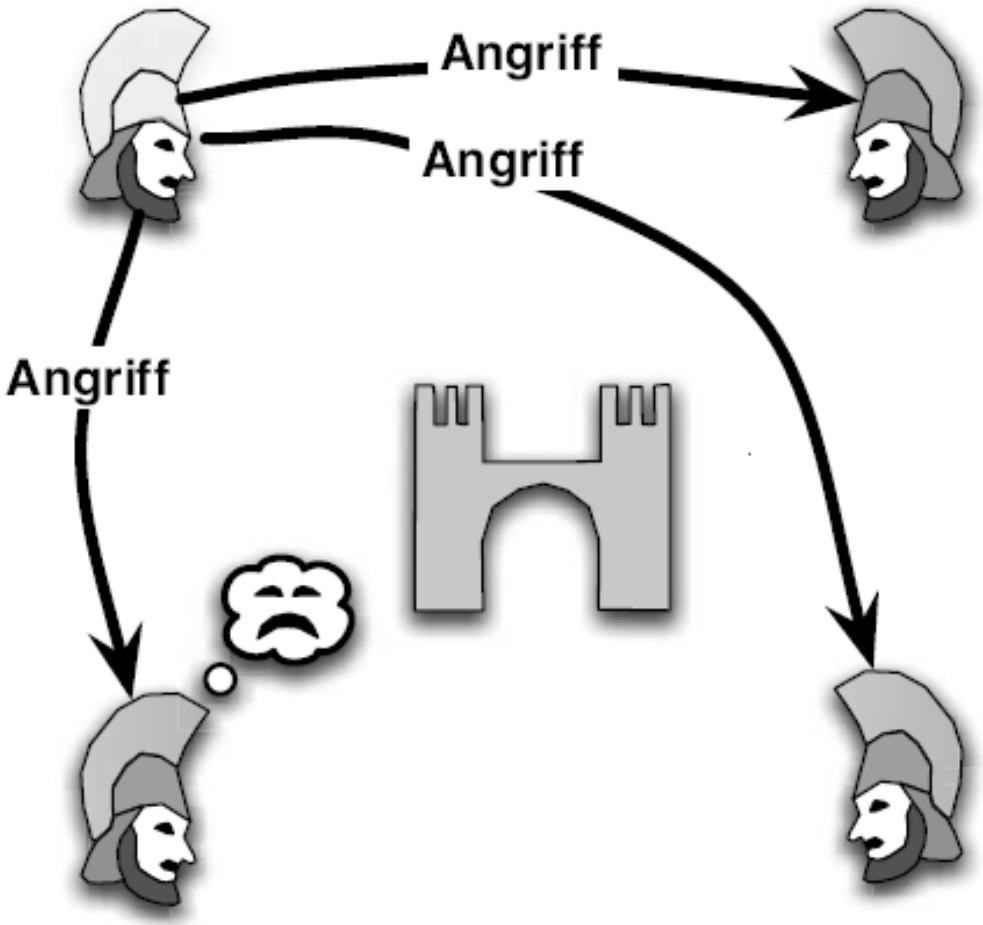
3 Gererals



Source: Christian Schindelhauer - Peer-to-Peer-Netzwerke

Byzantine Faults

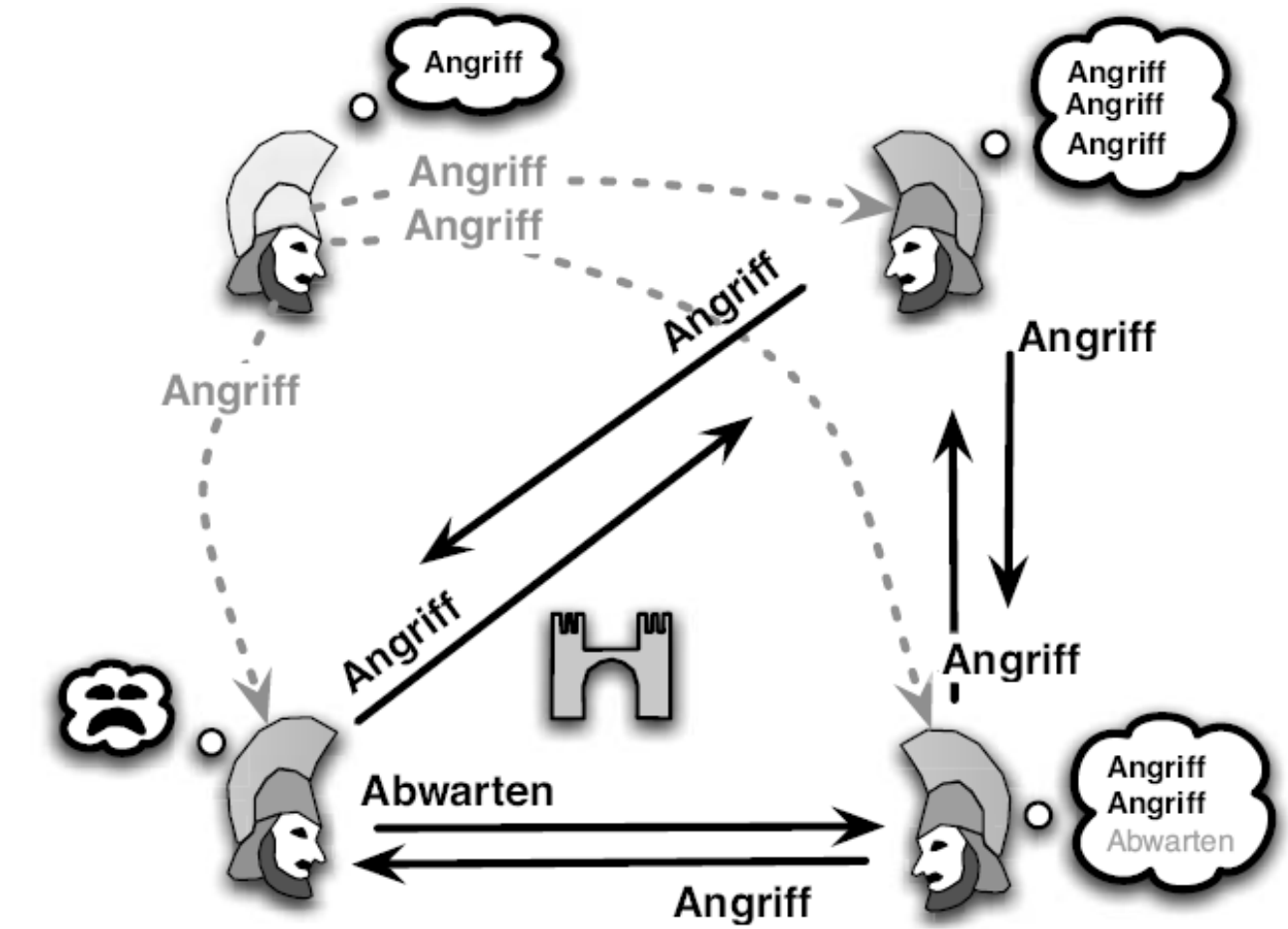
4 Gererals



Source: Christian Schindelhauer - Peer-to-Peer-Netzwerke

Byzantine Faults

4 Gererals



Source: Christian Schindelhauer - Peer-to-Peer-Netzwerke

Network of Amos Fiat and Jared Saia

Amos Fiat and Jared Saia presented a new kind of network

Network of Amos Fiat and Jared Saia

Amos Fiat and Jared Saia presented a new kind of network

- Censorship resistant: Robust against some possible attacks aiming to reduce the data availability



Network of Amos Fiat and Jared Saia

Amos Fiat and Jared Saia presented a new kind of network

- Censorship resistant: Robust against some possible attacks aiming to reduce the data availability
- Content addressable:
  - Uses distributed hashtables
  - Search can be performed using the hashcode of the data item



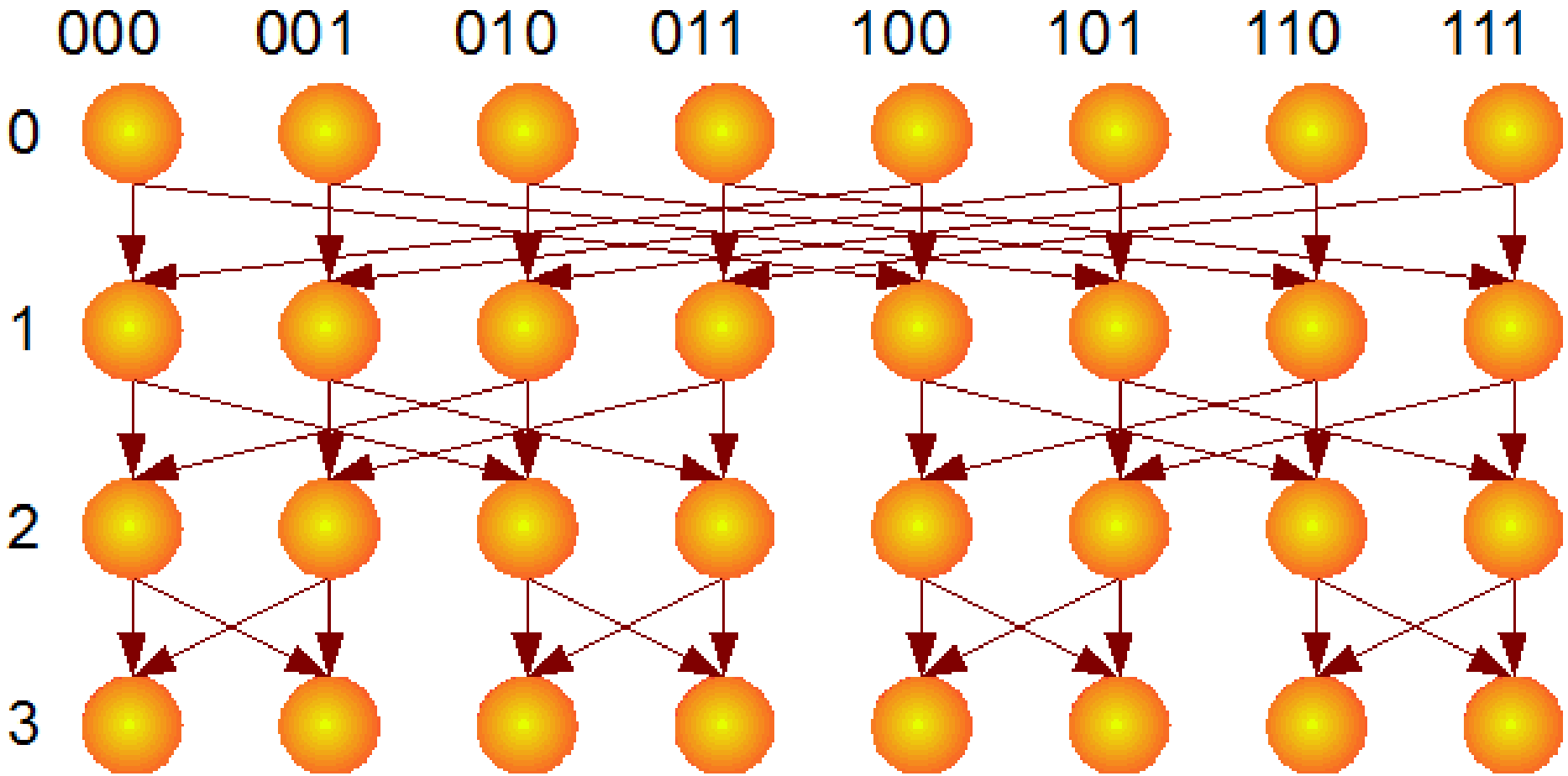
Network of Amos Fiat and Jared Saia

Amos Fiat and Jared Saia presented a new kind of network

- Censorship resistant: Robust against some possible attacks aiming to reduce the data availability
- Content addressable:
  - Uses distributed hashtables
  - Search can be performed using the hashcode of the data item
- Resistant against deletion of peers
- Modification also resistant against sending wrong messages (spam resistance)



Butterfly Network



Overview



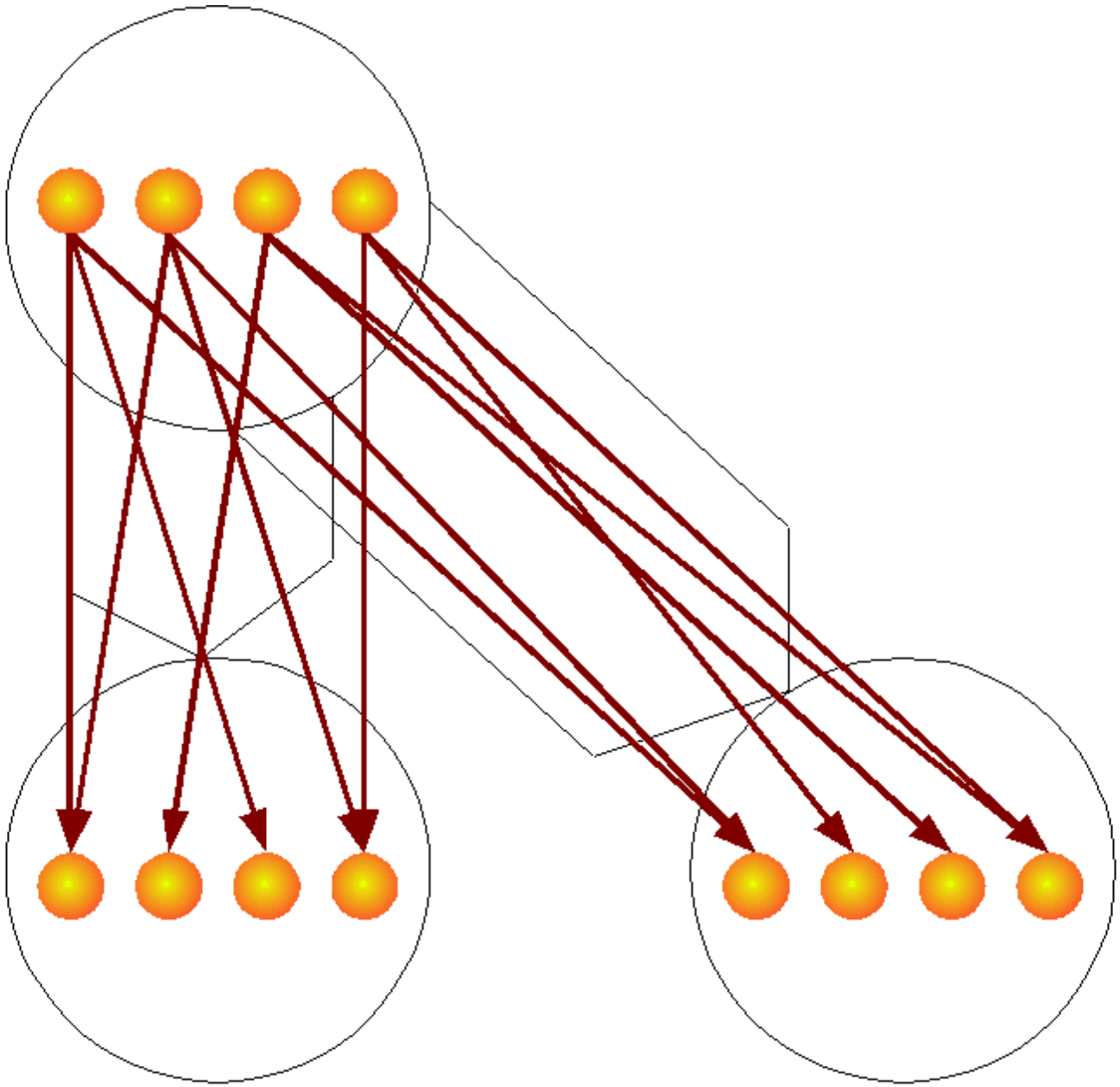
Creation of the network

Each peer chooses

- C top supernodes, C bottom supernodes and C log n middle supernodes
- For each supernode it is part of:
  D nodes in the neighboring supernodes to which it connects
- T top supernodes where searches are started later



Supernodes - Connection



Properties

Half of the participating peers can be deleted without changing the following properties:

Properties

Half of the participating peers can be deleted without changing the following properties:

Theorem (3.1.)

For all $\varepsilon > 0$, there exist constants $k_1(\varepsilon)$, $k_2(\varepsilon)$, $k_3(\varepsilon)$ which depend only on ε such that

Properties

Half of the participating peers can be deleted without changing the following properties:

Theorem (3.1.)

For all $\varepsilon > 0$, there exist constants $k_1(\varepsilon)$, $k_2(\varepsilon)$, $k_3(\varepsilon)$ which depend only on ε such that

- *Every node requires $k_1(\varepsilon)$ log n memory*

Properties

Half of the participating peers can be deleted without changing the following properties:

Theorem (3.1.)

For all $\varepsilon > 0$, there exist constants $k_1(\varepsilon)$, $k_2(\varepsilon)$, $k_3(\varepsilon)$ which depend only on ε such that

- *Every node requires $k_1(\varepsilon)\ \log n$ memory*
- *Search for a data item takes no more than $k_2(\varepsilon)\ \log n$ time*

Properties

Half of the participating peers can be deleted without changing the following properties:

Theorem (3.1.)

For all $\varepsilon > 0$, there exist constants $k_1(\varepsilon)$, $k_2(\varepsilon)$, $k_3(\varepsilon)$ which depend only on ε such that

- *Every node requires $k_1(\varepsilon)\ \log n$ memory*
- *Search for a data item takes no more than $k_2(\varepsilon)\ \log n$ time*
- *Search for a data item requires no more than $k_3(\varepsilon)\ \log^2 n$ messages*

Properties

Half of the participating peers can be deleted without changing the following properties:

Theorem (3.1.)

For all $\varepsilon > 0$, there exist constants $k_1(\varepsilon)$, $k_2(\varepsilon)$, $k_3(\varepsilon)$ which depend only on ε such that

- *Every node requires $k_1(\varepsilon)\ \log n$ memory*
- *Search for a data item takes no more than $k_2(\varepsilon)\ \log n$ time*
- *Search for a data item requires no more than $k_3(\varepsilon)\ \log^2 n$ messages*
- *All but εn nodes can reach all but εn data items*

Creation of the network

- Each data item is hashed to B bottom supernodes



Creation of the network

- Each data item is hashed to B bottom supernodes
- Network can be created in a fully distributed fashion
- Requires n^2 messages



Creation of the network

- Each data item is hashed to B bottom supernodes
- Network can be created in a fully distributed fashion
- Requires n^2 messages
- Each peer
  - chooses its supernodes randomly
  - Informs all other peers which supernodes it belongs to
  - Connects to nodes of neighboring supernodes



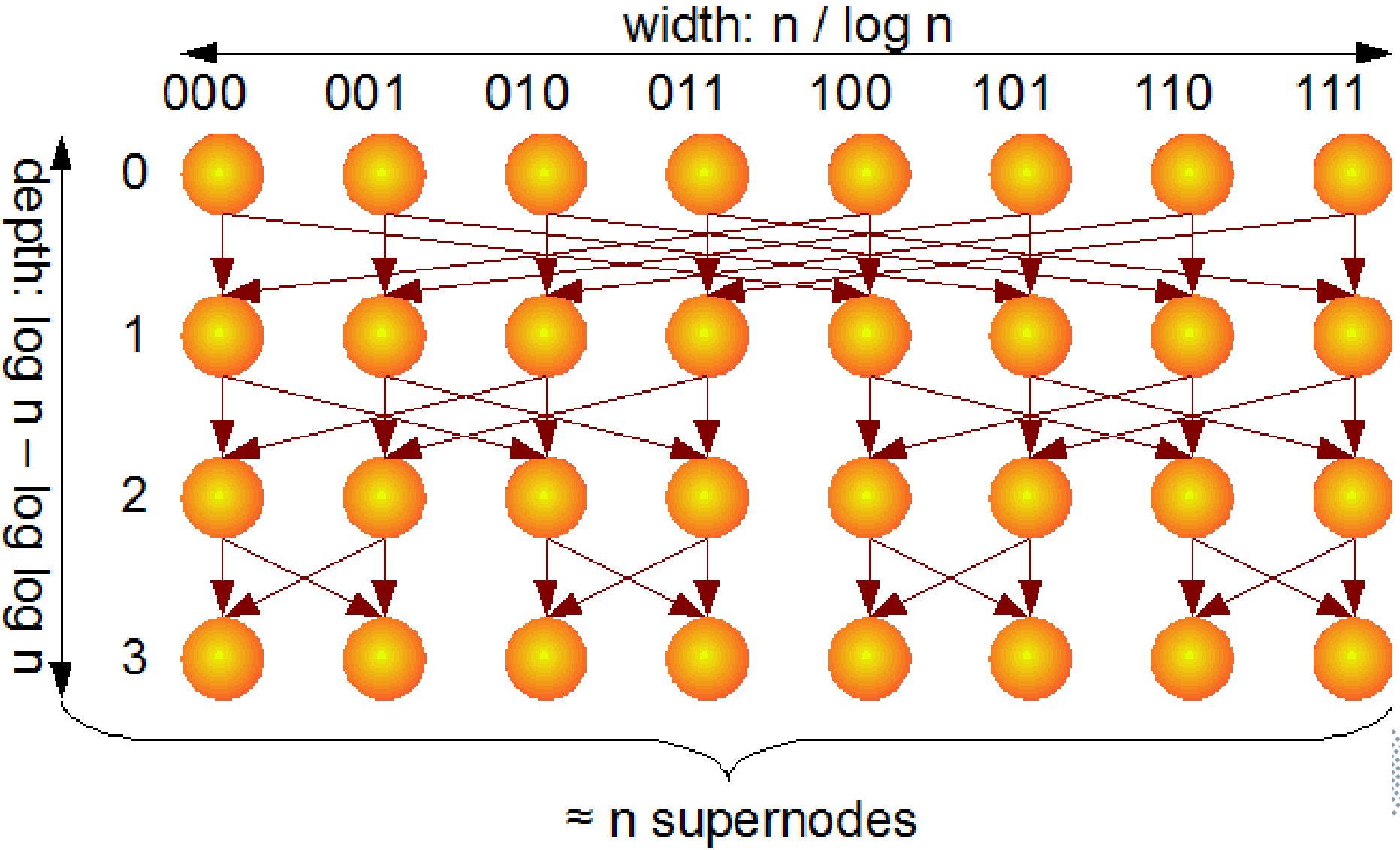
Creation of the network

- Each data item is hashed to B bottom supernodes
- Network can be created in a fully distributed fashion
- Requires n^2 messages
- Each peer
  - chooses its supernodes randomly
  - Informs all other peers which supernodes it belongs to
  - Connects to nodes of neighboring supernodes
- Finally delete supernodes with too few or too many nodes



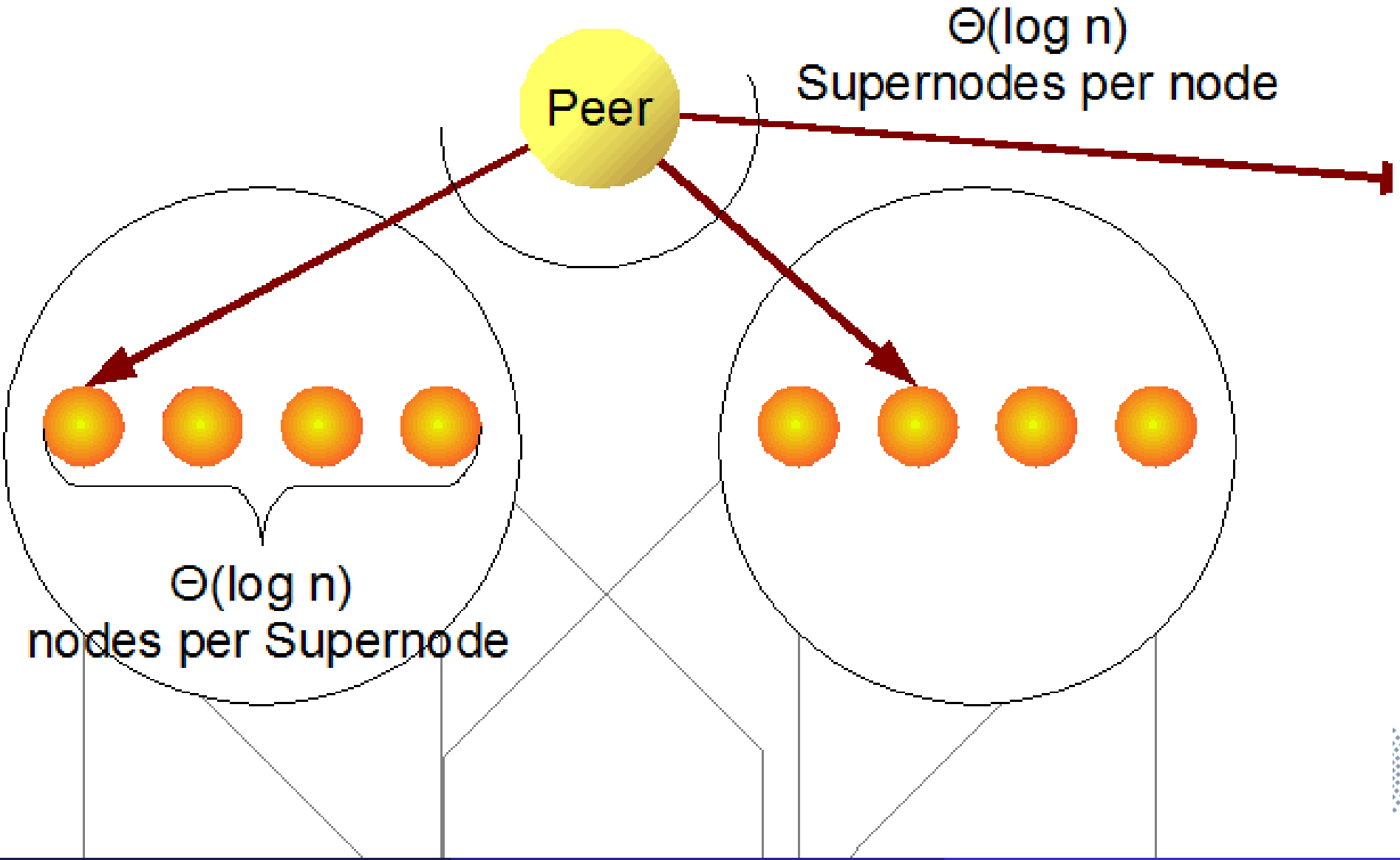
Size of the network

Butterfly graph



Size of the network

Size of a supernode



Size of the network

- n nodes and data items
- ($log\ n - \ log\ log\ n$) depth (nodes per column)
- $\frac{n}{log\ n}$ width (nodes per row)
- (log n) nodes per super node

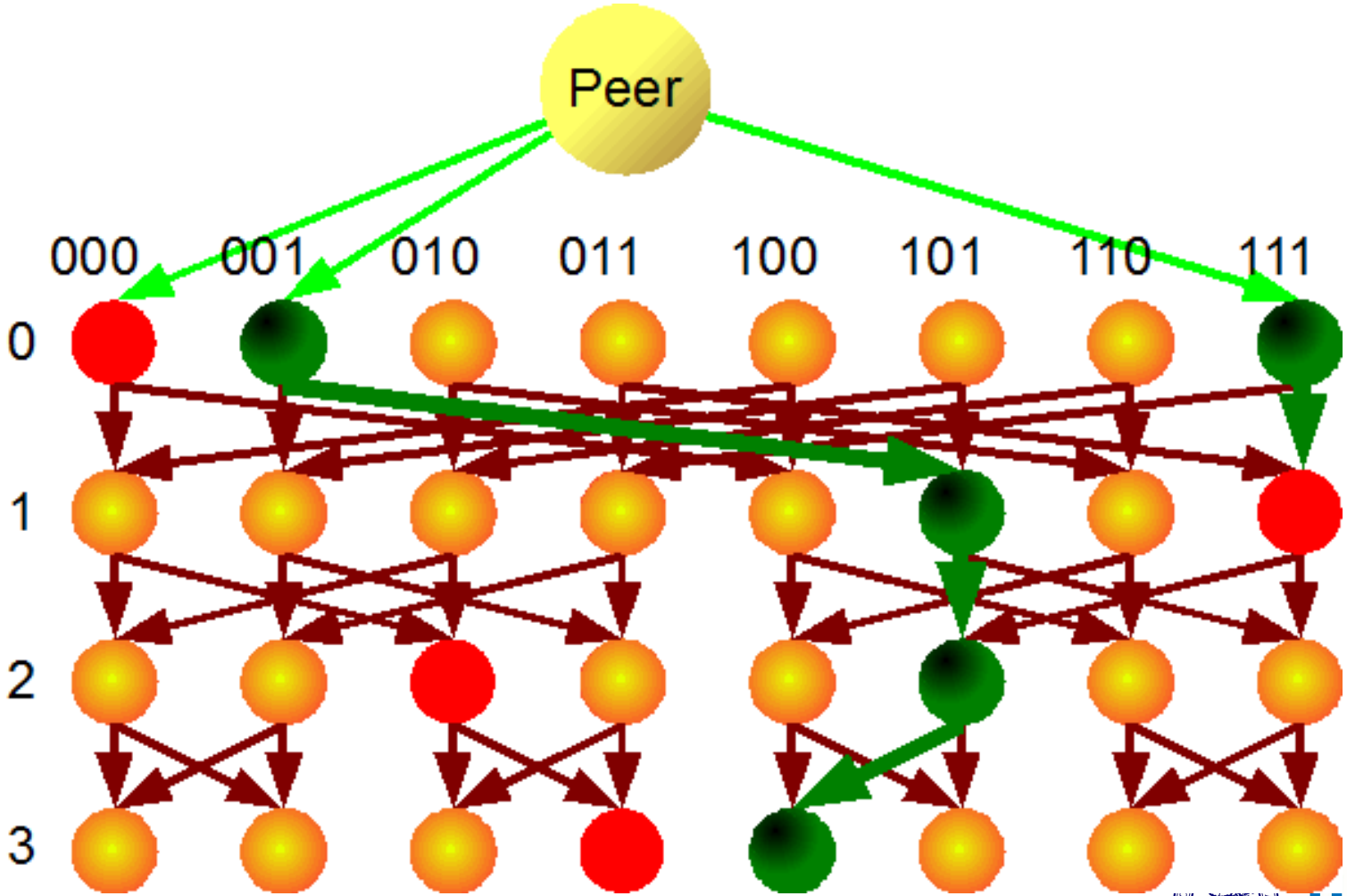


Overview



Launching a search

Example



Launching a search

Description

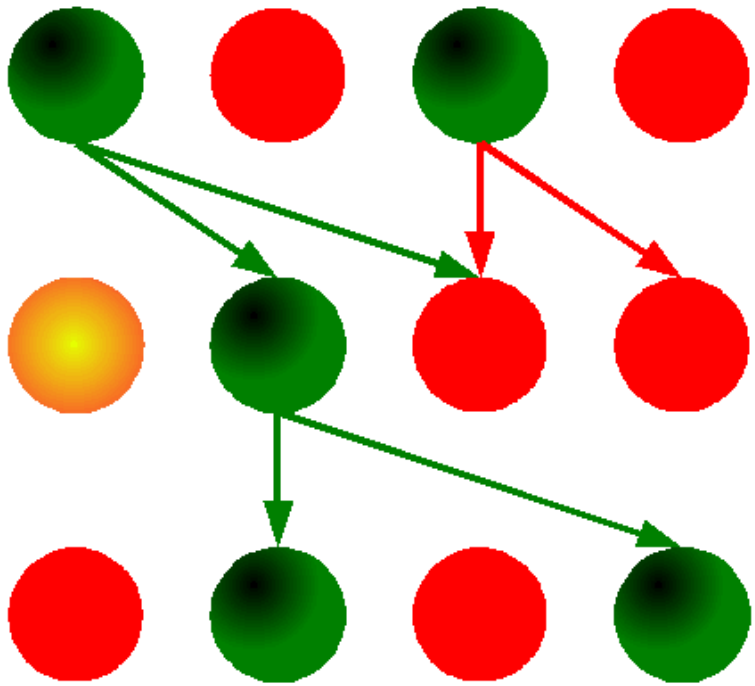
- Starts in all T top supernodes the peer has connected to
- Each peer in each such supernode is informed
- Each peer passes the message to all connected peers in the next supernode down the butterfly path
- The peers in the bottom supernode return the data item up the same path
- If not successful, the procedure is repeated for all bottom supernodes containing the data item



Launching a search

Passing the message

- Each row shows the nodes of one supernode on the path of the previous page



Overview



Proof overview

- Steps of the Proof:



Proof overview

- Steps of the Proof:
  1. Show that most supernodes are good



Proof overview

- Steps of the Proof:
  1. Show that most supernodes are good
  2. Show that most paths are good



Proof overview

- Steps of the Proof:
  1. Show that most supernodes are good
  2. Show that most paths are good
  3. Show that a search using only good paths works



Proof overview

- Steps of the Proof:
  1. Show that most supernodes are good
  2. Show that most paths are good
  3. Show that a search using only good paths works
  4. Show that most peers can reach nearly all data items



Proof

Show that most supernodes are good

Show that most supernodes are good

- A supernode is considered good, if it still contains $\Theta(\log n)$ live nodes
- If you remove half nodes in each supernode this is still the case
- The formal proof has to show that a deletion of peers leads to a mediocre deletion of nodes equally in all supernodes, but not to a big deletion in a large group of supernodes
- Additionally no supernode may contain too many child nodes



Proof

Show that most paths are good

Show that most paths are good

- Only few supernodes are ’bad’ supernodes
- The formal proof has to show that most paths do not use these supernodes



Proof

Show that a search using only good paths works

Show that a search using only good paths works

- Proof uses the expander properties



Proof

Show that most peers can reach nearly all data items

Show that most peers can reach nearly all data items

- Only missing aspect: Show that most peers are connected to the working paths

Proof overview - Repetition

- Steps of the Proof:
  1. Show that most supernodes are good
  2. Show that most paths are good
  3. Show that a search using only good paths works
  4. Show that most peers can reach nearly all data items



Overview



Necessary Modifications

- Spamming in this context means to send wrong messages

Necessary Modifications

- Spamming in this context means to send wrong messages
- Instead of a constant number connections each node connects to all nodes in the neighboring supernode

Necessary Modifications

- Spamming in this context means to send wrong messages
- Instead of a constant number connections each node connects to all nodes in the neighboring supernode
- Each node passes the message it receives by the majority of other nodes



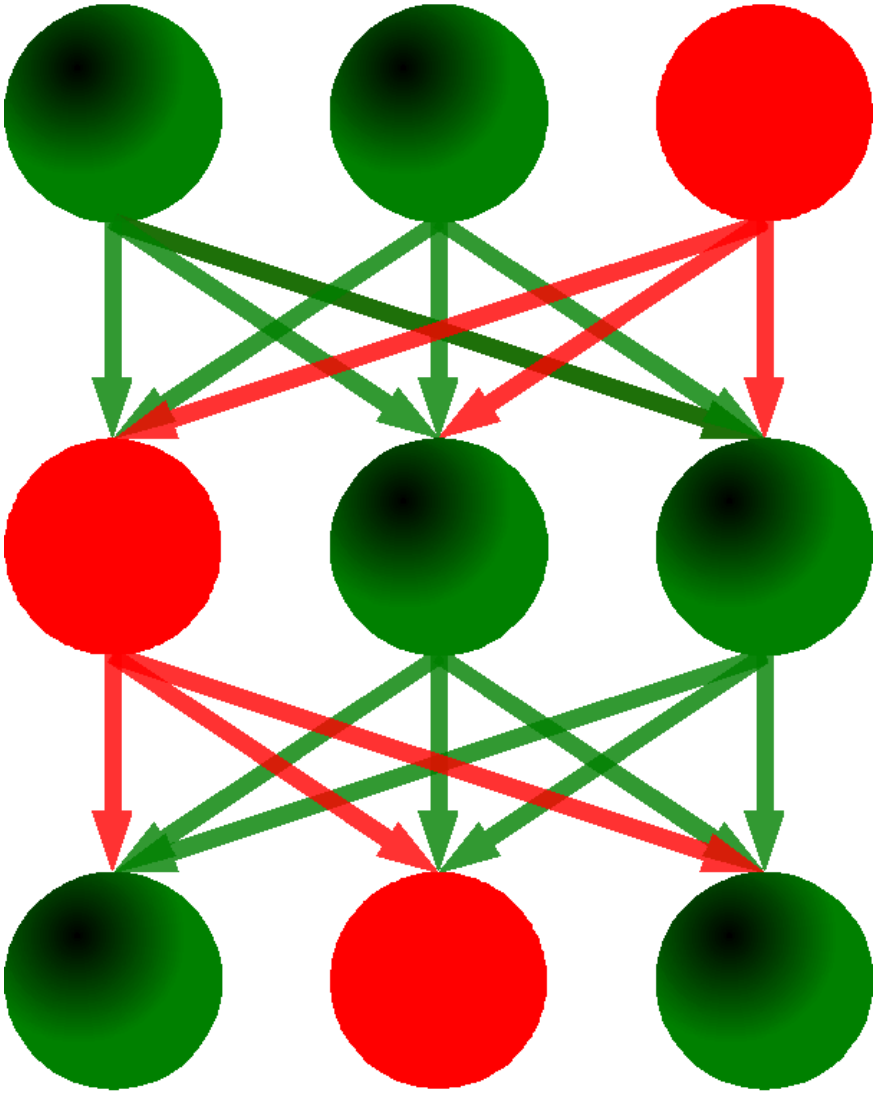
Necessary Modifications

- Spamming in this context means to send wrong messages
- Instead of a constant number connections each node connects to all nodes in the neighboring supernode
- Each node passes the message it receives by the majority of other nodes
- The fraction of adversarial controlled nodes must be strictly less than 50%



Spam Resistant Content Addressable Network

Example



Overview



Alternatives

- The presented network works without encryption
- Alternatives to this approach would be:
  - Access control
  - Encryption and signatures



Open problems of the paper

A. Fiat and J. Saia finished their paper with some open problems:

- Is there a mechanism for dynamically maintaining our network when large numbers of nodes are deleted or added to the network?



Open problems of the paper

A. Fiat and J. Saia finished their paper with some open problems:

- Is there a mechanism for dynamically maintaining our network when large numbers of nodes are deleted or added to the network?
  - According to his own words Mayur Datar solved this problem.
  - His approach is a multi butterfly network



1st problem

Second question

- Is it possible to reduce the number of messages that are sent in a search for a data item from O($log^2\ n$) to O(log n)?



1st problem

Second question

- Is it possible to reduce the number of messages that are sent in a search for a data item from O($log^2\ n$) to O(log n)?
  - Mayur Datar solved this problem, too (according to his own words)
  - Additionally each peer in his network only requires O(1) instead of O(log n) memory
  - The data availability degrades with the number of adversarial deletions. In the CRN some data items might be not available even if there is no enemy action



2nd problem

- 2. Can one improve on the construction for the spam resistant content addressable network?



3rd problem

- Can one deal efficiently with more general Byzantine faults? For example, the adversary could use nodes under his control to flood the network with irrelevant searches, this is not dealt with by either of our solutions.



The Economics of Censorship Resistance

Alternative to CRN: The Economics of Censorship Resistance

- Data items are stored randomly in the CRN
- It perhaps makes more sense if everybody stores the information he is interested in
- George Danezis and Ross Anderson present such a solution in "The Economics of Censorship Resistance"

Conclusion

- You can delete half of the nodes and the network will still work
- Outlook



Conclusion

- You can delete half of the nodes and the network will still work
- With little modifications the network also achieves spam resistance

- Outlook



Conclusion

- You can delete half of the nodes and the network will still work
- With little modifications the network also achieves spam resistance

- Outlook
  - Are there any real networks using this model?



Conclusion

- You can delete half of the nodes and the network will still work
- With little modifications the network also achieves spam resistance

- Outlook
  - Are there any real networks using this model?
  - How can spam resistant networks be made more efficient?



Appendix

Appendix

- In the Appendix the single Lemmas are being explained



Explanation of the parameters

Most of the following lemmas make use of the same parameters. The names of the parameters are:

- l: The left side of the bipartite graph
- r: The right side of the bipartite graph
- l': Subgroup of the left side
- r': Subgroup of the right side
- d: Degree of the nodes. (Number of nodes it is connected to)
- n: Number of nodes in the whole graph and number of data items



Explanation of the parameters - continuation

Greek letters are used as factors which are multiplied with other values

- λ: Only used in Lemma 4.2.
- α: Lower bound of allowed nodes per supernode. It is a factor of the average amount of nodes per supernode.
- β: Upper bound of allowed nodes per supernode.
- γ: Part of bottom supernodes that can be reached by a top supernode
- δ: Factor indicating wrong working supernodes



Lemma 4.1

Preparing Lemma 4.2.

Lemma 4.1.

- Let l,r,l',d and n be any positive values where $l' \leq r$ and $d \geq \frac{r}{r'l'}\left(l' ln\left(\frac{le}{l'}\right) + r' ln\left(\frac{re}{r'}\right) + 2 ln\ n\right)$
  Let G bei a random bipartite multigraph with left side L and right side R where $|L| = l$ and $|R| = r$ and each node in L has edges to d random neigbours in R. Then with probability at least $1 - \frac{1}{n^2}$, any subset of L of size l' shares an edge with any subset of R of size r'.

Relevance of Lemma 4.1

- General Lemma for bipartite graphs
- Later used to show facts about the allocation of peers
- Only one connection guaranteed

Lemma 4.2.

- Let I,r,I',r',d,λ and n be any positive values where $l' \leq l, r' \leq r, 0 < \lambda < 1$ and $d \geq \frac{2r}{r'l'(1-\lambda)^2}\left(l' \ln\left(\frac{le}{l'}\right) + r' \ln\left(\frac{re}{r'}\right) + 2 \ln n\right)$
  Let G be a random bipartite multigraph with left side L and right side R where |L| = l and |R| = r and each node in L has edges to d random neighbors in R. Then with probability at least $1 - \frac{1}{n^2}$, for any set L' $\subset$ L where $|L'| = l'$, there is no set R' $\subset$ R, where $|R'| = r'$ such that all nodes in R' share less than λl'd/r edges with L'.

Relevance of Lemma 4.2.

- Modifies Lemma 4.1. for the use in this proof
- Not only one connection, but a certain amount depending on parameters is guaranteed
- Later used in order to show that there are enough nodes in a supernode



Lemma 4.3.

- Let l, r, r', d, β' and n be any positive values where $l' \leq l, \beta' > 1$ and $d \geq \frac{4r}{r' l(\beta'-1)^2}\left(r' \ ln\left(\frac{re}{r'}\right) + 2 \ ln \ n\right)$
  Let G be a random bipartite multigraph with left side L and right side R where $|L| = l$ and $|R| = r$ and each node in L has edges to d random neighbors in R. Then with probability at least $1 - \frac{1}{n^2}$, there is no set $R' \subset R$,where $|R'| = r'$ such that all nodes in R' have degree greater than $\beta' ld/r$.

Relevance of Lemma 4.3.

- Prepares Lemma 4.5. and 4.7.
- Used in order to show that there are not too many nodes in a supernode

Lemma 4.4.

(α, β)-good supernodes

Lemma 4.4.

- Let α, δ', n be values where $\alpha < 1/2$ and $\delta' > 0$ and let k(δ', α) be a value that depends only on α, δ' and assume n is sufficiently large. Let each node participate in k(δ', α)ln n random middle supernodes. Then removing any set of n/2 nodes still leaves all but δ'n/ ln n middle supernodes with at least αk(δ', α) ln n live nodes.

Relevance of Lemma 4.4.

- Most of the middle supernodes have enough living nodes in them
- If they do not have too many nodes they are also (α, β)-good
- $\Theta\left(\frac{n}{\ln n}\right)$ bad supernodes $<<$ n middle supernodes in total
- (α, β)-good means that there are not too many and not too few nodes in the supernode

Proof of Lemma 4.4.

Prior considerations

- Each peer is connected to k(δ', α)ln n middle supernodes
- This connection could be pictured as a bipartite graph with the peers on the left side and the supernodes on the right side.
- In Lemma 4.2. we showed that under certain conditions there is no subgroup of the right side with less than $\frac{\lambda l' d}{r}$ edges
- This subgroup would be the group with too few living nodes
- $\Rightarrow$We just have to find the right values for the parameters l,r,l',r',d,λ and n

Proof of Lemma 4.4.

Calculation

The right values are:

$l = n, l' = \frac{n}{2}, r = n, r' = \frac{\delta' n}{\ln n}, \lambda = 2\alpha$ and $d = k(\delta', \alpha) \ln n$

Insertion in Lemma 4.2.:

$$d \geq \frac{2r}{r' l' (1-\lambda)^2} \left(l' \ln\left(\frac{le}{l'}\right) + r' \ln\left(\frac{re}{r'}\right) + 2 \ln n \right)$$

$$d \geq \frac{2n}{\frac{\delta' n}{\ln n} \frac{n}{2} (1-2\alpha)^2} \left(\frac{n}{2} \ln\left(\frac{ne}{\frac{n}{2}}\right) + \frac{\delta' n}{\ln n} \ln\left(\frac{ne}{\frac{\delta' n}{\ln n}}\right) + 2 \ln n \right)$$

$$d \geq \frac{4 \ln n}{\delta' n (1-2\alpha)^2} \left(\frac{n}{2} \ln 2e + \frac{\delta' n}{\ln n} \ln\left(\frac{e \ln n}{\delta'}\right) + 2 \ln n \right)$$

$$d \geq \ln n \left[\frac{2 \ln 2e}{\delta'(1-2\alpha)^2} + \frac{4}{\delta'(1-2\alpha)^2} \left(\frac{\delta' + \delta' \ln \ln n + \delta' \ln \delta'}{\ln n} + \frac{2 \ln n}{n} \right) \right]$$

$$d \geq \ln n \left[\frac{2 \ln 2e}{\delta'(1-2\alpha)^2} + o(1) \right]$$

$$\Rightarrow k(\delta', \alpha) \geq \frac{2 \ln 2e}{\delta'(1-2\alpha)^2} + o(1)$$

Lemma 4.5.

Most middle Supernodes do not have too many nodes

Lemma 4.5.

- Let β, δ',n,k be values such that $\beta > 1$, $\delta' > 0$ and assume n is suffciently large. Let each node participate in k ln n of the middle supernodes, chosen uniformly at random. Then all but $\delta' n / ln\ n$ middle supernodes have less than $\beta k\ ln\ n$ participating nodes with probability at least $1 - \frac{1}{n^2}$.

Relevance of Lemma 4.5.

- Most middle supernodes do not have too many nodes
- Together with Lemma 4.4. we know that most middle supernodes are (α, β)-good

Lemma 4.6.

Top and bottom supernodes have enough live nodes

Lemma 4.6.

- Let α, δ', n be values such that $\alpha < \frac{1}{2}$, $\delta' > 0$ and let k(δ', α) be a value that depends only on δ' and α and assume n is sufficiently large. Let each node participate in k(δ', α) top (bottom) supernodes. Then removing any set of $\frac{n}{2}$ nodes still leaves all but $\frac{\delta' n}{ln\ n}$ top (bottom) supernodes with at least αk(δ', α) ln n live nodes.

Relevance of Lemma 4.6.

- Most top and bottom supernodes have enough live nodes

Lemma 4.7.

Lemma 4.7.

- Let β, δ',n,k be values such that $\beta > 1$, $\delta' > 0$ and n is sufficiently large. Let each node participate in k of the top (bottom) supernodes (chosen uniformly at random). Then all but $\frac{\delta' n}{ln\ n}$ top (bottom) supernodes consist of less than $\beta k\ ln\ n$ nodes with probability at least $1 - \frac{1}{n^2}$.

Relevance of Lemma 4.7.

- Most top and bottom supernodes do not consist of too many nodes
- Together with Lemma 4.6. we know that Most top and bottom supernodes are (α, β)-good
- Together with Lemma 4.4 and 4.5 we know that most of all supernodes are (α, β)-good



Corollary 4.1.

Corollary 4.1.

- Let β, δ',n, k be values such that $\beta > 1, \delta' > 0$ and n is sufficiently large. Let each data item be stored in k of the bottom supernodes (chosen uniformly at random). Then all but $\frac{\delta' n}{ln\ n}$ bottom supernodes have less than $\beta k\ ln\ n$ data items stored on them with probability at least $1 - \frac{1}{n^2}$.

Relevance of Corollary 4.1.

- Most of the bottom supernodes do not have too many data items stored on them
- Equates to Lemma 4.7.



Corollary 4.2.

Corollary 4.2.

- Let $\delta' > 0, \alpha < \frac{1}{2}, \beta > 1$. Let k($\delta', \alpha$), be a value depending only on δ' and assume n is sufficiently large. Let each node appear in k(δ', α) top supernodes, k(δ', α) bottom supernodes and k(δ', α) In n middle supernodes. Then all but $\delta' n$ of the supernodes are $(\alpha k(\delta', \alpha), \beta k(\delta', \alpha))$-good with probability 1 - O($\frac{1}{n^2}$).

Relevance of Corollary 4.2.

- Puts the previous Lemmas together
- Most supernodes are (α, β)-good
- Prepares Theorem 4.1.



Theorem 4.1

Definition 4.4.

- A top supernode is called (γ, α, β)-expansive if there exist $\frac{\gamma n}{log\ n}$ (α, β)-good paths that start at this supernode.

Theorem 4.1.

- Let $\delta > 0, \alpha < \frac{1}{2}, 0 < \gamma < 1, \beta > 1$. Let k$(\delta, \alpha, \gamma)$ be a value depending only on δ, α, γ and assume n is sufficiently large. Let each node participate in k(δ, α, γ) top supernodes, k(δ, α, γ) bottom supernodes and k(δ, α, γ) ln n middle supernodes. Then all but $\frac{\delta n}{ln\ n}$ top supernodes are $(\gamma, \alpha k(\delta, \alpha, \gamma), \beta k(\delta, \alpha, \gamma))$-expansive with probability 1 - O$(\frac{1}{n^2})$.

Theorem 4.1.

Most Peers can reach most of the data items

Relevance of Theorem 4.1.

- Most top supernodes can use nearly $\frac{n}{\log\ n}$ paths
- There are $\frac{n}{\log\ n}$ bottom supernodes
- Most top supernodes can reach most bottom supernodes
- The bottom supernodes contain the data items

Lemma 4.8.

Lemma 4.8.

- Let $\delta > 0, \varepsilon > 0$ and n be suffciently large. Then exists a constant k(δ, ε) depending only on ε and δ such that if each node connects to k(δ, ε) random top supernodes then with high probability, any subset of the top supernodes of size $\frac{(1-\delta)n}{ln\ n}$ can be reached by at least $(1 - \varepsilon)$n nodes.

Relevance of Lemma 4.8.

- Most peers can reach at least one (γ, α, β)-expansive top supernode
- Most peers can reach most bottom supernodes

Lemma 4.9.

Lemma 4.9.

- Let γ, n, ε be any positive values such that $\varepsilon > O$, $\gamma > O$. There exists a k(ε, γ) which depends only on ε, γ such that if each bottom supernode holds k(ε, γ)ln n random data items, then any subset of bottom supernodes of size $\frac{\gamma n}{ln\ n}$ holds $(1 - \varepsilon)$n unique data items.

Relevance of Lemma 4.9.

- Most of the data items are mapped to some bottom supernodes which can be reached by the top supernodes
- Together with the previous Lemmas, we know that most peers can reach most data items using only (α, β)-good supernodes

Lemma 4.10.

Connection between (α, β)-good supernodes

Lemma 4.10.

- Let $\alpha, \beta, \alpha', n$ be any positive values where $\alpha' < \alpha, \alpha > 0$ and let C be the number of supernodes to which each node connects. Let X and Y be two supernodes that are both $(\alpha C, \beta C)$-good. Let each node in X have edges to k(α, β, α') random nodes in Y where k(α, β, α') is a value depending only on α, β and α'. Then with probability at least $1 - \frac{1}{n^2}$, any set of $\alpha' C \ln n$ nodes in X has at least $\alpha' C \ln n$ live neighbors in Y.

Lemma 4.10.

Relevance of Lemma 4.10.

- A path consisting only of (α, β)-good supernodes grants connection between both ends with high propability
- Together with the previous Lemmas we can now say that most of the peers can reach most of the data items even after half of the nodes have been deleted

The proof is finished:

- The network is robust against the deletion of half of the peers
- Most peers can still reach most data items