A Censorship Resistant Contant Addressable Peer-to-Peer Network

Pascal Minnerup

7.9.2008

Contents

1 Introduction

There are several reasons which can threaten the integrity of a peer-to-peer network. These events typically make some peers of the network acting in a wrong way. This could be for legal reasons. Amos Fiat and Jared Saia[FS02] give the example of Napster, which has been dismembered by legal attacks on the central server. Another possibility would be an totalitarian regime which wants to control the news an documents which its population may receive. This regime could try to control the data items and the participants of a peer to peer network.

Making peer-to-peer networks resistant against such attacks has been a central effort of various designs of such networks. One quite important paper to this topic is that of Amos Fiat and Jared Saia: Censorship Resistant Peer-to-Peer Content Addressable Networks[FS02]. In my work I will explain their ideas and solutions and sometimes give some extra information. Therefore I'll first show, how their network works, then I will describe why their networks meets the requirements and finally I will show how they proof the rightness of their solution.

2 Background

To understand the presented network and estimate its relevance it might be useful to know some facts about censorship and often used models.

2.1 What does censorship mean?

Censorship means that for some reason the peer-to-peer network doesn't provide the full and the right data which it has been filled by. This could be for several random or adversarial reasons. One is the deletion of nodes. Because of computer failures or by adversarial influence some peers do not respond to any demands any more. If this deletion is caused by computer failure it will probably be a random deletion. That means that each node has the same probability to be deleted. The other possibility is more dangerous: Adversarial deletion means that an enemy chooses those nodes which would damage the network as much as possible. Even harder to control is the problem of spamming. Spamming in this context means to send some false information instead of the demanded one. As the receiver cannot judge whether he got the right information he can't just ignore this messages.

2.2 Byzantine Faults

Byzantine faults are a group of possible faults in a peer to peer network caused by some enemy peer in the network. The name Byzantine faults goes back to eastern Roman empire. As the generals there wanted to become emperor themselves you could never be sure whether they obeyed the orders or acted against the other generals to reach their own aims. [MS07]

Three generals have the order to attack a town. They are stationed on three different locations and can only communicate by mounted messengers. If at least two of them attack the town together they will win the battle. If two armies keep waiting they win the battle, too, because the town does not get enough food under the siege and has to surrender. So the three generals only have to decide whether to attack or not. The only important point is that they do the same. But one of the generals cooperates with the enemy town and tries to make the other two generals do different things.
Provided that they have no authentication or encryption protocol the two loyal generals have no possibility possibility to identify the disloyal general or the get a common solution. Each loyal general gets two conflicting messages. Which one should he trust?
Now lets alter the situation. There are now four generals with the order to attack the city. Three generals are loyal one is not. The generals win if three generals perform the same action.
According to the book Peer-to-Peer Netzwerke this situation could be reduced to the problem of one general and three officers. The general gives an order and all three officers repeat this order to the other officers. Each officer then performs the action which most of the other officers voted for. If the general is loyal each officer gets one right message from the general and one from the other officers. This makes a majority of two right message against one probably wrong one. If the general is disloyal he sends conflicting messages to the officers. As there are only two possible messages there is one message he has to send to two officers. This message will be repeated and will outvote the other message. As result in both cases all three loyal persons will perform the same action.

At home I did not find out how to reduce the problem of four generals to the problem of one general and three officers. The four generals would have to appoint one leader. But this election could be disturbed by the enemy general, too. The only possibility I could find out was to wait until there is an overwhelming majority of all three loyal generals voting for the same person. Provided you have enough time this would happen once. But still the disloyal general could send vote messages such that some generals think there is an overwhelming majority and others see a 50%-50% situation.

3 Censorship Resistant Peer-to-Peer Network

3.1 General Structure

A&J make use of a butterfly graph for their network. Each node of this graph is a super node which has Teta(log n) peers connected to it. Each peer connects to C*log(n) super nodes. Unlike in the normal butterfly graph the version of A&J does not connect the 0th level and the bottom level. By this you can differ between top supernodes and bottom supernodes. The bottom supernodes contain all the data items. The top supernodes are used to start search requests. The middle supernodes pass the search request to the right supernode in the next level according to the routing rules. Each peer connects to some top supernodes, some middle supernodes and some bottom supernodes. Thus each peer can start a search using the top supernodes it is connected to. Additionally each peer stores some data items, because it is connected to some bottom supernodes. Now we have a structure of nodes assigned to several supernodes, but no connection between the nodes. This connection is created by the following rule: A supernode in the butterfly network has two neighbouring supernodes it the next level. Each child node connects do a constant number of child nodes of these two supernodes. The resulting bipartite graph has to be an expander graph. In simple word this means that each subgroup of nodes has at least a constant factor times as many neighbours as elements. Note that a node is not connected to all nodes of the next supernode.

3.2 Size of the network

Attribute	**Calculationsteps**	**Value**
Number of Peers	Given	n
Number of Data items	Given	n
Depth	Given	log n - log log n
Width	$2^{depth} = 2^{log\ n - log\ log\ n} = \frac{2^{logn}}{2^{log\ log\ n}}$	$\frac{n}{log\ n}$
Number of Supernodes	$Width * Depth = (\frac{n}{logn}) \cdot (log\ n$ $-log\ log\ n) = n \cdot \left(1\frac{log\ log\ n}{logn}\right) \approx n$	$\approx\ n$
Number of middle supernodes	$\Theta\left(Number of Supernodes\right)$	$\approx n$
Number of top supernodes	$= width$	$\frac{n}{log\ n}$
Number of bottom supernodes	$= width$	$\frac{n}{log\ n}$
Number of supernodes per peer	Given	$\Theta\left(log\ n\right)$
Number of child nodes	$[Number\ of\ peers] \cdot$ $[Number\ of\ supern.\ per\ peer] =$ $= \Theta(log\ n \cdot n)$	
Number of nodes per supernode	$\frac{Number of childnodes}{Number of supernodes} = \frac{n\ log\ n}{n} = log\ n$	log n

3.3 Lauching a Search

In order to search a specific data item A&J present an algorithm. The whole algorithm can be found on the fourth page of the their paper. I will not copy it here, but only explain in my own words how it works.

To perform the search a peer needs the hash-code of the data item. A&J note that this is typically not the hash-code of the whole item, but only the title like "Singing in the rain". Using this hash code the peer starts the search at all top supernodes it has connected to. These are Teta(log n) top supernodes. The peer informs all nodes which belong to the supernode. All informed nodes pass the request to all nodes they can reach in the next supernode down the unique butterfly path. These nodes pass it to the next level and so on. In the end the nodes of the bottom supernodes return the data item up the same path the search request used.

3.4 Censorship Resistance

In a network with only well working peers it would be enough to use one path for the search request. The network of A&J is additionally still working when half of the peers are removed. Working means: With high probability, all but an arbitrarily small fraction of the nodes can find all but an arbitrarily small fraction of the data items. In other words: It is nearly sure that you can find the item you are searching for.

A&J proof this property on four pages of their paper. Here I will show why this network works and then briefly explain how A&J proof the properties. The crucial point of the network is that each peer is connected to many supernodes. A&J assume that an enemy can delete half of the peers. One option would be to delete all peers of some bottom supernodes. A&J now demand that only a arbitrarily small constant factor of the data items get lost. Lets say this is 1%. If each peer would only connect to one supernode you would be able to disable half of the supernodes by deleting half of the peers. But in the network of A&J each peer connects to a constant number of bottom supernode. So the higher this constant is the more supernodes keep working. If the constant is high enough there will still be 99% bottom supernodes working. Additionally each data item is not only stored in one supernode, but in a constant number of supernodes. By increasing this constant you would also increase the percentage of data items available after deleting half of the peers. A&J also highlight that you could as well delete more than half of the peers if you choose the other constants high enough.

4 Proof

A. Fiat and J. Saia use 10 Lemmas for proving their theorem. These lemmas can be arranged to the following steps:

- Show that most supernodes are good
- Show that most paths are good
- Show that a search using only good paths works
- Show that most peers can reach nearly all data items

On the following pages I will explain the used Lemmas and say what they are good for. As an example I will show the calculation steps for one proof. These steps are not part of the source paper. For the other Lemmas you can look up the proof in the paper of A. Fiat and J. Saia.

4.1 Technical Lemmas

The first three Lemmas are general Lemmas which are used for several proofs. Lemma one is a Lemma which has already been proven in other papers. Lemma two and three are modifications of the first Lemma.

Lemma 4.1.

This is a general Lemma which is used to determine statements about connections in bipartite graphs. These graphs can be parts of the butterfly graph or just virtual constructions. A. Fiat and J. Saia use this Lemma to show facts about the allocation of some nodes to others. This includes especially the allocation of peers to supernodes. But this Lemma grants only one connection. As this is not enough for the whole proof, the lemma will be modified in Lemma 4.2:

Lemma 4.2.

This modification of Lemma 4.1. grants more than one connection. The actual amount of connections depends on the parameters. Later this Lemma will be used to show that most supernodes contain enough live nodes.

The last technical Lemma is again a modification of Lemma 4.1. It is some kind of an opposite Lemma to Lemma 4.2. Whereas Lemma 4.2. shows that there are enough connections, Lemma 4.3. shows that there are not too many connections.

Lemma 4.3.

In the proof of A.Fiat and J.Saia, this Lemma is used to show that there are not too many nodes in a supernode. Too many nodes would be bad for the network, because more nodes would mean that there can be more dead nodes. This would decrease the probability that enough live nodes in two neighbouring supernodes are connected.

4.2 (α, β)-good supernodes

The next four Lemmas show that most of the supernodes are (α, β)-good. (α, β)-good Means that the supernode includes enough live nodes but does not have too many nodes mapped to it.

Lemma 4.4.

Lemma 4.4. shows that there are enough live nodes in most of the middle supernodes. Enough means enough for being (α, β)-good. Of course this does not proof that there are few enough nodes so not all middle supernodes including enough live nodes are (α, β)-good. Most of the supernodes in this case are all but only very few supernodes. $\Theta\left(\frac{\delta' n}{In\ n}\right)$ is much smaller than the amount of n middle supernodes in total. In fact this would only be a linear percentage of one row of the butterfly graph.

As this is perhaps a surprising fact, I consider this Lemma as quite important. This is why I chose it to calculate the proof of the Lemma. As for all other Lemmas it consists mainly of inserting the right values in previous Lemmas. For this reason I do not show the calculation steps for all other Lemmas, too. If you want to know how the proof of a certain Lemma works, you can look it up in the source paper.

Lets start the proof with some prior considerations:

- Each peer is connected to k(δ', α)In 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. We have to proof that this group does not exist.
- $\Rightarrow$We just have to find the right values for the parameters l,r,l',r',d,λ and n

The right values are:
$l = n, l' = \frac{n}{2}, r = n, r' = \frac{\delta' n}{ln\ n}, \lambda = 2\alpha$ and $d = k\left(\delta', \alpha\right) ln\ n$
Insertion in Lemma 4.2.:

$$d \geq \frac{2r}{r'l'(1-\lambda)^2}\left(l'\ ln\left(\frac{le}{l'}\right) + r'\ ln\left(\frac{re}{r'}\right) + 2\ ln\ n\right)$$
$$d \geq \frac{2n}{\frac{\delta' n}{ln\ n}\frac{n}{2}(1-2\alpha)^2}\left(\frac{n}{2}\ ln\left(\frac{ne}{\frac{n}{2}}\right) + \frac{\delta' n}{ln\ n}\ ln\left(\frac{ne}{\frac{\delta' n}{ln\ n}}\right) + 2\ ln\ n\right)$$
$$d \geq \frac{4ln\ n}{\delta' n(1-2\alpha)^2}\left(\frac{n}{2}\ ln\ 2e + \frac{\delta' n}{ln\ n}\ ln\left(\frac{e\ ln\ n}{\delta'}\right) + 2\ ln\ n\right)$$
$$d \geq ln\ n\left[\frac{2\ ln\ 2e}{\delta'(1-2\alpha)^2} + \frac{4}{\delta'(1-2\alpha)^2}\left(\frac{\delta' + \delta' ln\ ln\ n + \delta' ln\ \delta'}{ln\ n} + \frac{2\ ln\ n}{n}\right)\right]\ d \geq ln\ n\left[\frac{2\ ln\ 2e}{\delta'(1-2\alpha)^2} + o(1)\right]$$
$$\Rightarrow k\left(\delta', \alpha\right) \geq \frac{2\ ln\ 2e}{\delta'(1-2\alpha)^2} + o(1)$$

So we now have a value for $k\left(\delta', \alpha\right)$ which makes the inequation become true. If we insert it in the parameter $d = k\left(\delta', \alpha\right) ln\ n$ this fact becomes obvious. In this Lemma we have now prooven that most of the middle supernodes still have enough live nodes allthough half of the peers have been deleted.

Lemma 4.5.

Lemma 4.5. shows that most middle supernodes do not have too many supernodes mapped to them. Together with Lemma 4.4. we can now diagnose that most of the middle supernodes are (α, β)-good.

Lemma 4.6.

As Lemma 4.4. showed for middle supernodes, Lemma 4.6. shows that most of the top and bottom supernodes have enough live nodes.

Lemma 4.7.

As Lemma 4.5. showed for middle supernodes, Lemma 4.7. shows that most of the top and bottom supernodes do not have too many nodes mapped tothem. We can now say that most top supernodes are (α, β)-good. For the bottom supernodes we still need corollary 4.1.

Corollary 4.1.

Most bottom supernodes do not have too many data items stored on them. This was the missing point to indentify most of the bottom supernodes as (α, β)-good. All in all we can no say that most of all supernodes are (α, β)-good. This is done in Corollary 4.2. in a more formal way.

4.3 (γ, α, β)-expansive

Knowing that most of the supernodes are (α, β)-good, the next step is to show that most paths are good.

The next theorem uses it:

Theorem 4.1.

This theorem says that most of the top supernodes can use nearly all their $\frac{n}{log\ n}$ possible paths to bottom supernodes. As the searches are started at the top supernodes and the bottom supernodes contain the data items, most such searches would be succesful.

Lemma 4.8.

Not all of the top supernodes are (γ, α, β)-expansive. Lemma 4.8. shows that anyway most peers are connected to at least one (γ, α, β)-expansive top supernodes. (γ, α, β)-expansive means that most of the paths starting in this top supernode contain only (α, β)-good supernodes. Hence the peers can reach most bottom supernodes, where the data items are stored using only (α, β)-good supernodes. The last step from the bottom supernodes to the data items is analysed in Lemma 4.9:

Lemma 4.9.

Not all bottom supernodes can be reached by a specific (γ, α, β)-expansive top supernode. Lemma 4.9. shows that anyway most of the data items can be reached using only the other bottom supernodes. This finishes the path between the peer which lauched the search and the data item.

4.4 Connection between (α, β)-good supernodes

Only one last point is remaining: Is a $\alpha, \beta)$-good path good enough to grant a high probability for the success of the search? This point is analysed in the last Lemma:

Lemma 4.10.

This Lemma grants that a path consisting only of (α, β)-good supernodes grants connection between both ends with high propability. This is because for each supernodes on the way we know that at least a lower bound of live nodes will be able to pass the message. 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. This is what we wanted to show.

5 Final Thoughts

A. Fiat and J. Saia finished their paper with a modification of their network and some open problems. In this last part I will present these further points and additionally mention two other papers about censorship resistant peer-to-peer networks.

5.1 Spam Resistant Content Addressable Network

Spamming in this context means to send wrong messages. A modification of the Censorship Resistant Peer-to-Peer network is able to recognise and block these messages. The difference to the CRN is that instead of connecting to a constant amount of nodes in the next supernode each node connects to all log n nodes. This makes it possible to only pass the majority of messages. The premise is that less than half of the nodes are controlled by the enemy. If more nodes would be controlled by him he would obviously have the majority.

5.2 Alternatives an open problems

A censorship resistant network could also be realised using authentification, encryption and access control. None of these techniques is being used by the presented Censorship Resistant Peer-to-Peer network.

Additionally 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[Dat02]. His approach is a multy butterfly network.

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)[Dat02]. 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.

I did not find a solution to the other two problems:

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

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.

5.3 The Economics of Censorship Resistance

The Economics of Censorship Resistance[DA04] is an alternative to the Censorship Resistant Content Addressable Peer-to-Peer network of Amos Fiat and Jared Saia (CRN). In the CRN data items are stored randomly. In the real world it would perhabs make more sense to prefer those data items which one peer is interested in. As this peer would probably be willing to provide more storage for data items it stores anyway, this would increase the total amount of available storage and therefore increase the positive effects as censorship resistance. George Danezis and Ross Anderson present such a solution in 'The Economics of Censorship Resistance'

6 Conclusion

The network of A. Fiat and J. Saia was one of the first models to provide censorship resistance. Allthough later models have improved this solution it has been a pathbreaking paper for many later works. The distinctive feature of the network is that you can delete half of the peers or with little modifications even control nearly half of the peers and still the network will handle its service.

References

[DA04] George Danezis and Ross Anderson. The Economics of Censorship Resistance. *The Third Annual Workshop on Economics and Information Security*, June 2004. `http://homes.esat.kuleuven.be/~gdanezis//redblue.pdf`.

[Dat02] Mayur Datar. Butterflies and Peer-to-Peer Networks. *Conference or Journal Paper*, March 2002. `http://dbpubs.stanford.edu:8090/pub/showDoc.Fulltext?lang=en&doc=2002-33&format=pdf&compression=&name=2002-33.pdf`.

[FS02] Amos Fiat and Jared Saia. Censorship Resistant Peer-to-Peer Content Addressable Networks. *13th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2002. `http://www12.informatik.uni-erlangen.de/edu/fa/K1/lit/p94-fiat-1.pdf`.

[MS07] Peter Mahlmann and Christian Schindelhauer. *Peer-to-Peer-Netzwerke*. Springer-Verlag, Berlin Heidelberg, 2007.