CS 591, Lecture 17

Jared Saia
University of New Mexico

Class Outline

- Using Informers to Restrain Free Riders
- Peer Join
- Message send under fail-stop fault model
- Message send under Byzanting fault model

Class Presentations

- Aaron - Routing in Small World Networks, Vishal - An Application of Chebyshez's Inequality, Jun - Online Machine Learning, Jijune - Phylogenitic/Graph Theory Applications, Vinay - the Byzantine General's problem, Ravi - power-aware disk scheduling
- By November 5th (this coming Tuesday), you must turn in a 1 page detailed summary of what you will present. Give a description of the abstract problem, the algorithm(s), and all major theorems (you should try to prove no more than 1 or 2 theorems).

Presentation Signup

- 11/19 Vinay and Jijun
- 11/21 Ravi and Vishal
- 11/26 Jun and Aaron

"Free Rider" Problem

The "Three" ruled with the help of an elaborate network of agents and informers. The city was divided up, with a set number of informers for each ward and parish - Donald Phau in "Venice: The Methodology of Evil"

- The problem: how can we enforce good social behavior for peers?
- Question: What is "good social behavior"?
- More generally: how can we maintain "law and order" on the p2p frontier?

Informers

- A key idea: Let the peers "inform" on each other
- Each peer, in addition to playing its normal roles in the network, also plays the role of an "informer" for some peers
- An informer for some peer p, watches the way that p interacts with the network and ensures that if p is not interacting appropriately, that p is cut off from the network informing on a fixed number of other peers
- All communication that any peer has with the p2p network first passes through the informers for that peer.

Informers

- Some of the informers will be untrustworthy
- They may try to collude to let bad peers get away with things
- They may try to collude to unfairly punish good peers

We use the following tools to handle bad informers:

- Require a large number of informers - massive redundancy
- Informers are assigned "at random" to inform on someone

Random Assignment of Informers

- Since there is no centralized authority, getting a random assignment is tricky
- Improper that peers perform the random selection themselves since
- 1) "bad" informers could all assign themselves to some "bad" peer and thus protect that peer
- 2) "bad" informers could all assign themselves to some "good" peer and thus punish that peer unfairly

Unique ID

- In order to have any hope of success, each peer must have a unique, verifiable id determined by some outside authority
- This id could be the ip-address of the peer
- Without such id's, it's possible for one bad peer to pretend that it is a very large number of separate peers and thus take over the network
- For a peer p, let i_p be the unique id for p

Random Assignment

- Let n be the total number of peers in the network
- We make use of a "random oracle" function f which maps Z to $\{0, \ldots, n\}^{\text{polylog (n)}}$
- This function essentially maps integers to random tuples of length polylog (n) over the set $\{1, \ldots, n\}$
- For a peer p with id, i_p, $f(i_p)$ will give the ids of the informers of p

Join Protocol for joining peer p

- p broadcasts message that i_p is joining to all peers in network
- Let S be the set of peers p receives replies from that claim to be p's informers. Let S' be the set of verified informer in S (i.e. $S' = \{s \in S | i_s \in f(i_p)\}$
- p sets it's informer set $I(p)$ to be S'
- p establishes connections with all peers in $I(p)$

Join Protocol for non-joining peer o

- o receives join message from p and obtains p's id, i_p
- If $i_o \in f(i_p)$, o broadcasts a claim that it is an informer for p to all other peers. It also adds p to its informant set, $I^{-1}(o)$
- o receives messages from all peers which claim to be informers for p. Let $I(p)$ be the verified set of all such peers
- Let $I^{-1}(o)$ be the set of peers that o informs on. For each $x \in I^{-1}(o)$ such that (x, p) is an edge in the underlying network, o establishes an informer-to-informer link to each peer in $I(p)$

Analysis

- Assume we are given some protocol for creating a p2p network without informers (e.g. the gnutella protocol or the CRN or DRN protocol)
- This protocol will define some graph G_p over the peers which would be the topology of the overlay network without informers
- The join algorithm creates a network which mirrors G_p

Fail-stop faults

- First consider the situation where bad peers suffer "fail-stop" faults
- Recall "fail-stop" means that bad informers simply stop communicating
- We give a light-weight scheme for informing and communicating through the network in this case

Analysis

- Each peer x has a set $I(x)$ of informers chosen randomly from all the peers. $|I(x)| = k * \log(n)$
- The join algorithm creates a network which mirrors G_p
- For each edge (x, y) in G_p, all peers in $I(x)$ and in $I(y)$ are connected in a full bipartite graph

Communication: Peer to Informer

- Peer protocol: When a peer p requests a data item or serves a data item, it sends out a message to a random informer in $I(p)$
- Informer protocol: When an informer receives a message from its peer, it does the following
  - If the message is a data item requested by a previously logged request, it logs this response and broadcasts information on the response to all other informers in $I(p)$
  - Otherwise, with probability p_i, it logs information on the message sent out and broadcasts this information to all other peers in $I(p)$
  - Finally, it sends the message to the next appropriate informer cloud using the protocol described under *Informer to Neighboring Informer Cloud*

Informer to Peer

- When an informer receives a message to its peer, it does the following:
- If the message is a request for data, with probability p_i, it logs information on the request and broadcasts this information to all other informers in the informer cloud.
- Next, it sends the message to the peer

Informer to Neighboring Informer Cloud

- When an informer has a message which is to be sent to a neighboring informer cloud, it does the following:
- Until it gets a response, it chooses a random informer in the neighboring informer cloud and pings that informer
- Once a live informer is found, the message is sent to that live informer

Analysis

- Let p_f be the ratio of the number of failed peers to the total number of peers
- Assume we want to communicate a message from some peer a to some peer b in the original network
- Let p_i be any number in the range 0 to 1
- Let m be the number of informers in each informer cloud (m is polylog in n)

Analysis

Theorem: The above protocol allows for communication along the path from a to b such that

- The message sent from a is known by all peers in $I(a)$ with probability p_i, and is known to all peers in $I(b)$ with probability p_i
- The expected time to send the message is $(1/p_f)l + 2 * p_i$
- The expected number of messages sent is $(1/p_f)l + 2 * p_i * m$
- The time to send and number of messages to send are sharply concentrated around their expected values

Byzantine Faults

- Now we'll consider the case of Byzantine faults
- Recall that these Byzantine are strictly worse than fail-stop faults
- We'll assume that at any time no more than a p_f fraction of the peers are bad for some $p_f < 1/2$
- To handle Byzantine faults, we'll first consider the probability that any informer cloud contains a majority of bad peers.

Analysis

- A2: Let x be some fixed peer. Note that $I(x)$ is formed by making $log^k(n)$ independent random choices from the n peers in the network
- Let X_i be 1 if this choice is a bad peer and 0 otherwise
- Note that $E(X_i) = p_f$.
- Let $X = \sum_{i=1}^{k*\log n} X_i$. Note by linearity of expectation that $E(X) = k * \log n * p_f$

Analysis

- Let p_f be worst case ratio of bad peers in the network to good peers
- Q1: What is the probability that there is some peer x, such that $I(x)$ does not have a majority of good peers?
- Q2 (Easier): What is the probability that for some *fixed* peer x, $I(x)$ does not have a majority of good peers?

Chernoff Bounds

- Assume p_f is a fixed constant less than 1/2
- Then we want to find:

$$P(X \geq (1/2) * k * \log n) = P(|X - E(X)| \geq \delta E(X))$$

where $\delta = (1/2 - p_f)/p_f$
- By Chernoff Bounds,

$$P(|X - E(X))| \geq \delta E(X) leq 2e^{\frac{-E(X)\delta^2}{4}}$$

Chernoff Bounds

- In this case, that means that for $\delta = (1/2 - p_f)/p_f$,

$$P(|X - E(X))| \geq \delta E(X) leq 2e^{\frac{-E(X)\delta^2}{4}}$$

- This probability is $2e^{-Ck * \log n}$ where $C = (p_f * \delta^2)/4$ (note that C depends only on p_f)
- Thus for any $k' > 0$, k can be chosen as a function of k' and p_f so that this probability is no more than $n^{-k'}$

Union Bounds

- Now we can answer the more difficult Q1:
- Q1: What is the probability that there is some peer x, such that $I(x)$ does not have a majority of good peers?
- We know the probability that for a *fixed* peer x, the probability of failure is no more than $n^{-k'}$
- By Union Bound, we can say that the probability of failure for any of the peers is no more than n^{-k+1}