CS 591, Lecture 19

Jared Saia
University of New Mexico

Class Outline

Message send under Byzanting fault model

- Reducing blow up in number of messages sent from $O(\log^2 n)$ to $O(\log n)$ (Review)
- Reducing blow up in number of messages sent from $O(\log n)$ to $O(1)$

The Problem

- Some fraction less than half of the peers suffer Byzantine faults
- These peers will try to disrupt communication
- Q: Can we still ensure that communication will proceed as expected?
- Note: it helps us that we can say that with high probability, each informer cloud has a majority of good peers in it

Review

Recall from last time:

- Each peer has a unique digital signature and id that is verifiable by everyone but forgeable by no one
- The digital signature is used to "sign" an entire message, so that each message can be verified to be generated by a unique entity at a given time.
- There is a function I which maps every id to $k \log n$ random peers currently in the network

Review

Peer Join:

- Let G be some given network
- We can assign random id's to each
  For any node x in G, let $I(x)$ be the set of informers that x's will maps to
- For any underlying network G, there is a protocol to create

Review

- 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 (the network without informers)
- For each edge (x, y) in G_p, all peers in $I(x)$ and in $I(y)$ are connected in a full bipartite graph

Review

- Let $\alpha > 0$ be chosen such that we know the fraction of good peers in the network is always $1/2 + \alpha$
- Last time, we showed that the probability that there is some x such that $I(x)$ does not have a majority of good peers is "small".
- In particular, for any $\alpha > 0$ and $k' > 0$, we can choose k dependent only on α and k' such that each set has greater than a 1/2 fraction of good peers is at least $1 - n^{-k}$
- Note: we can do even better. We can say that for any value $\alpha' < \alpha$ and $k' > 0$, we can choose k dependent only on α' and k' such that the probability that every set has greater than a $1/2 + \alpha'$ fraction of good peers is at least $1 - n^{-k'}$

Majority preserving graph

- Let $G = (L, R)$ be bipartite graph with $|L| = |R|$.
- G is a *(α, β)-MPG* if it satisfies the following properties:

- G is regular with degree $d(\alpha, \beta)$ where $d(\alpha, \beta)$ is a value depending only on α and β
- For every $S \subset L$ such that $|S| \geq (1/2+\alpha)|L|$, at least $(1-\beta)|R|$ nodes in R have a majority of their neighbors in S

MPG Intution

- Having the informer clouds connected by such graphs would be great. Why?
- 1) Constant number of links between nodes
- 2) Majority Preserving property ensures secure communication. Why?
- What do we want? We want α and β as small as possible

Creating MPG's

- Turns out that a random, regular bipartite graph will be a "good" MPG
- Let α, β be any fixed constants greater than 0 with $\alpha > \beta$ and let l be sufficiently large
- Then there exists a value $d(\alpha, \beta)$ such that a random $d(\alpha, \beta)$-regular bipartite multigraph over $2l$ vertices is a (α, β)-MPG with high probability
- Specifically, the probability of failure is exponentially small in l
- (The proof of this is omitted)

How does an MPG help?

- Assume that the fraction of good peers is $1/2 + \alpha'$ where $\alpha' > 0$
- Let the informer clouds be connected using an (α, β)-MPG where α and β are chosen such that $\alpha' - \beta \geq \alpha$
- Let messages be sent in the informer clouds along all the links of the MPG

How does an MPG Help?

- Fact: $O(\log n)$ messages are sent in each hop
- Fact: If $1/2 + \alpha$ good peers have the right message in the sending informer cloud, then at least $1/2 + \alpha$ good peers in the receiving informer cloud will get the right message.
- Why? Because in worst case, β of the good peers in the receiving informer cloud get the wrong message. Hence at least $1/2 + \alpha' - \beta$ good peers got the message and $\alpha' - \beta \geq \alpha$.
- (Note: We could have the same exact MPG between all informer clouds, and even if the adversary knows the single MPG topology, it can do nothing)

Can we do better?

- So we've gotten done to $O(\log n)$ messages sent for every hop from informer cloud to informer cloud
- Can we do any better than that?
- Key ideas: split the message into pieces, error-correcting codes and finger-printing

Encoding the message

- Assume we are sending some message x with $|x|$ bits in it and that there are $m = k * \log n$ peers in every informer cloud
- Let $e_1, e_2, \ldots, e_m$ be encoded pieces of x which have the following properties
- 1) Each e_i is of size $O(|x|/m)$
- 2) x can be reconstructed from *any* subset of $e_1, e_2, \ldots, e_m$ of size at least $m/2$

Fingerprinting

- Let $f_1, \ldots, f_m$ be fingerprints of $e_1, \ldots, e_m$ respectively
- We assume that the fingerprint f_i can be used to determine if some untrusted message is in fact e_i
- Note that if the fingerprint f_i is long enough, we can make it arbitrarily difficult for an adversary to generate a different message which has the fingerprint f_i

The Setup

- Let π be a mapping from the integers $\{1, \ldots, m\}$ to random subsets of size c of the pieces $\{e_1, \ldots, e_m\}$
- The i-th peer in an informer cloud will be "responsible" for the pieces $\pi(i)$
- At the end of a stage of the algorithm, we will ensure that a constant fraction of the good peers know all of the pieces they're "responsible" for

The Setup

A peer set A is said to be α-*aware* of the message x if the following properties hold:

- All good peers in A know $f_1, f_2, \ldots, f_m$
- There is some set S of αm good peers in A such that for all peers $a_i \in S$, a_i knows the encoded pieces $\pi(i)$.

Note that an α-aware peer set can reconstruct the message for α and c sufficiently large.

The Algorithm (I)

- Each peer in A sends all known fingerprints to peers in B according to a certain MPG.
- The peers in A choose a peer, r, uniformly at random from all peers in B using the distributed algorithm for choosing a random number (discussed two lectures ago)
- Each peer a_i in A sends the pieces $\pi(i)$ to r (if possible)

Algorithm (II)

- r reconstructs the encoded pieces as follows:
  1. r removes all the pieces it receives that don't match one of the fingerprints that r has
  2. r reconstructs a message x' from the remaining pieces
  3. r computes encoded pieces, $e'_1, \ldots, e'_m$, for x'
- r sends each peer b_i the encoded pieces specified by $\pi(i)$

Algorithm (III)

- Each peer b_i in B sends the message "Pieces Ok" to all peers in A if it received all the pieces in $\pi(i)$ and all of these pieces have correct fingerprints.
- If a good peer in A receives the message "Pieces Ok" from less than $(1/2+\gamma')m$ peers in B, it loops back to the second step of the algorithm.
- Else the peer sends the message "Success" to all peers in B.

Analysis

- Claim: If the informer cloud A is α-aware of the message x, then at the end of the procedure, informer cloud B will be α-aware of x
- Why?
- Q1: What is the total number of bits sent?
- Q2: What is the total number of messages sent?