Towards a Lazy yet Efficient Dynamic DHT

Diksha Gupta
UNM STEM Symposium 2020

Joint work with Prof. Jared Saia & Prof. Maxwell Young
What is a DHT?
**What is a DHT?**

**Distributed Hash Table** is a storage system that provides efficient lookup using (key,value) pairs.

Source: [https://www.educative.io/edpresso/what-is-a-distributed-hash-table](https://www.educative.io/edpresso/what-is-a-distributed-hash-table)
Distributed Hash Table is a storage system that provides efficient lookup using (key,value) pairs.

- Decentralized & Autonomous
What is a DHT?

Distributed Hash Table is a storage system that provides efficient lookup using (key,value) pairs.

- Decentralized & Autonomous
- Fault Tolerant
What is a DHT?

**Distributed Hash Table** is a storage system that provides efficient lookup using (key,value) pairs.

- Decentralized & Autonomous
- Fault Tolerant
- Scalable
Popular Applications

- BitTorrent
- Akamai
- cassandra
Vulnerability - Join/Leave Attacks
Some peers depart
Some peers depart
Malicious peers join
Lookups fail
Problem Statement

• System consists of \( n \) peers, at most \( k < 1/3 \) fraction are malicious (bad) peers.
• Peers can join and depart from the system.
• Non-malicious (good) peers follows the protocols.
• Bad peers can deviate arbitrarily under control of an adversary.
• Adversary cannot evict good peers.
Problem Statement

To define a join strategy that maintains following properties in the neighborhood of every peer in the system:

- **Balancing** - Consists of $c \log n$ peers.
- **Majority** - Has majority of honest peers.
Previous Work

Cuckoo Rule by Awerbuch and Scheidler (SPAA-2006)
Previous Work

Cuckoo Rule by Awerbuch and Scheidler (SPAA-2006)

Setup:

Peers mapped onto unit ring.

Neighborhood: Peers in region of size $(c \log n) / n$ in clockwise direction.
Previous Work

**Cuckoo Rule** by Awerbuch and Scheidler (SPAA-2006)

**Setup:**
- Peers mapped onto unit ring.
- Neighborhood: Peers in region of size $(c \log n) / n$ in clockwise direction.

**Join rule:**
- Choose a random position on ring.
- Shuffle - Move all peers in $(\log n)/n$ neighborhood to random positions on ring.
Previous Work

**Cuckoo Rule** by Awerbuch and Scheidler (SPAA-2006)

Setup:
Peers mapped onto unit ring.
Neighborhood: Peers in region of size $(c \log n)/n$ in clockwise direction.

Join rule:
Choose a random position on ring.
Shuffle - Move all peers in $(\log n)/n$ neighborhood to random positions on ring.

Limitation: Requires unnecessary shuffling.
Our Goal

Reduce the number of shuffles
Lazy Cuckoo Rule (LCR)

Join rule:

Choose a position on the ring.

If the number of peers in some peer’s neighborhood is not in interval:

\[
(1 - \epsilon)c \log n, (1 + \epsilon)c \log n
\]

Move all peers in this neighborhood to random locations on the ring, and bring in \(c \log n\) peers chosen uniformly at random to this neighborhood.
Key Idea

Assuming that peers join independently and uniformly at random, then at any time:

\[
\text{Expected number of peers in region of size } \frac{c \log n}{n} : c \log n
\]
Key Idea

Assuming that peers join independently and uniformly at random, then at any time:

Expected number of peers in region of size \((c \log n)/n\) : \(c \log n\)

At most \(k\)-fraction of peers are bad i.e., not independent and uniformly at random joins.
Key Idea

Assuming that peers join independently and uniformly at random, then at any time:

Expected number of peers in region of size \( \frac{c \log n}{n} \): \( c \log n \)

At most \( k \)-fraction of peers are bad i.e., not independent and uniformly at random joins.

If a region has more than \( (1 + \epsilon)c \log n \) peers, then this region has some bad peers join \( \Rightarrow \) Shuffle.
Key Idea

Assuming that peers join independently and uniformly at random, then at any time:

\[
\text{Expected number of peers in region of size } \frac{c \log n}{n} : c \log n
\]

At most \( k \)-fraction of peers are bad i.e., not independent and uniformly at random joins.

If a region has more than \( (1 + \epsilon)c \log n \) peers, then this region has some bad peers join \( \Rightarrow \) Shuffle.

Only shuffle when there have been some bad peers join in a peer’s neighborhood.
Future Work

- Extend this work to polynomially varying system size.
- Implement and compare performance with state-of-the-art.
Thank you!
Questions