Detecting and avoiding DOS attacks on the get() operation in DHTs

Emil Ong
<emilong@cs.berkeley.edu>
UC Berkeley
Quick background on DHTs

- Nodes and data keys exist in the same address space
- Nodes whose addresses are closest to a key store data on a put() and serves it on a get()
- We deal with recursive routing
The Problem: Low cost DOS attack on get() in DHTs

- A malicious node:
  1. Waits to be a forwarder for a get()
  2. Sends “does not exist” to querier
  3. Does not forward the request

- The victim does not know if the response is correct!
Existing solution: Alternate routing

- Send get() queries along multiple paths
  - Can be done in case of negative response or proactively
  - Couples detection with resolution
  - Can be expensive, especially when used proactively
New approach: Network size estimation

- **Idea:** If we get a negative response from a node that is far from the key, it is probably lying.
- **Difficulty:** What does “far” mean?
  - Distance between node and key is much greater than $(\text{address space size})/(\text{network size})$
- **Now we need a way to estimate network size**
  - Use leaf set to estimate network size, assume uniform density.
Solution

- All non-existence responses must now include the responder’s leaf set
- Check:
  - Leaf set span and size
  - Distribution
- Even if the responder lies, its leaf set will span too much of the address space
Status

- Currently evaluating through emulation
- Early results:
  - Most honest nodes overestimate the size of the network
  - Dishonest nodes must underestimate the size of the network to make a believable lie
- Still evaluating best threshold for low false-positives/false-negatives
The End

Thank you!