P2P Networks – Exercise
Solution For Exercise # 7

Ikram M. Khan, P2P Networks Group,
TU Darmstadt

Date: Dec. 20th, 2011

1. Overlays (1/2)

- What can goes wrong with overlays?
  - Overlay neighborhood is not real neighborhood – induces delay
  - Useless traffic traverse over the same physical link – stress
- How can we handle load-balancing problems in overlays?
  - [Look at DHASH]
- Can we create location aware overlay?
  - We can, by using characteristics of lower layer during overlay constructions

1 Overlays (2/2)

- Can we create a „location aware“ overlay?
  - BTW: what is „location“ on the Internet?
    - Darmstadt (DFN) -> Berlin (DFN) can be a lot „closer“ than Darmstadt (DFN) -> Weiterstadt (DSL)!
  - Common (mis-) used metrics:
    - RTT (ECHO, ping)
      - But: DSL without fastpath has ping times like TU-Darmstadt -> Vanuatu...
    - Bandwidth between end-hosts
      - Which bandwidth? Overall? Available? How do we measure that?
    - IP-Hops
      - Stuttgart is in same distance cmp. to New York (www.dfn.de vs www.ny.com)

2. 1. Is Chord a DHT?

- Chord does not provide complete functionalities of a DHT
  - It does not provide storage of <key, value> pairs
  - Does not associate values with identifiers

- Chord is a scalable lookup service
  - It associates keys with values
  - Successor lookup function is implemented in Chord

2. 2) DHASH

- DHASH is layered Chord application
- Uses Chord as a lookup service
  - Insert ()
  - Lookup ()
- Layer 1: Chord – Maps indentifiers to successor nodes
- Layer 2: DHASH – Associate values with identifiers
- Layer 3: Application – provide a file system interface

2. 3) DoS Attacks

- Pruning links
  - Resistant against DoS attacks because of network locality
- Polluting with large amount of data
  - Flushing legitimate values from distributed storage
  - Limiting the number of blocks a single node stores
  - Limiting relation among nodes in the system
- Picking own identifier
  - Could delete data by positioning themselves as successor of data
  - Using strong ID = hash(IP, ...) could provide prevention
- Arbitrarily incorrect behavior
  - Monitoring and verifying nodes' responses with others
    - Mutual verification to verifying node's responses
  - There is no way to proof malicious activities of a group of nodes

2. 4) Load Balancing

- Avoid to store complete document on a node
  - Splitting files into blocks
  - Insert each block into the DHASH layer with the hash of the block
- Spread a single file among several nodes
- Use meta-data to provide a single name for a distributed, multi-blocked file

Programming Exercise - Routing Table

- RPC Identifiers?
  - Provides safety against forged messages.
  - Eases parsing when many messages travel around.
- Sender Indentifiers?
  - Ping works without.
  - Eases parsing Pong.
  - Enables Backward learning!
- Kademlia design specification
  http://xlattice.sourceforge.net/components/protocol/kademlia/specs.html

Next?

- There will be no theory/programming exercise during winter break

- So, Merry Christmas and Happy New Year