

# One Hop Lookups Plugin for RELOAD

IETF81@Quebec, Canada  
draft-peng-p2psip-one-hop-plugin-00

Jin Peng  
Kai Feng  
Lifeng Le

# Agenda

- Why One Hop Lookups Plugin?
- Peer data structure
- Event notification procedure
- Updates message
- Leader choosing strategies
- Fault tolerance

# Why One Hop Lookups Plugin?

- Topology Plugin of RELOAD
  - Each overlay can select an appropriate overlay algorithm that relies on the common RELOAD core protocols and code
- High demands for the improvement of routing efficiency in real time applications
  - **Chord**:  $O(\log N)$
  - **One Hop Lookups**:  $O(1)$

# Why One Hop Lookups Plugin?

- Requirements for RELOAD
  - The one hop lookups plugin is based on the methods provided by RELOAD which include the framework of commonly-needed methods defined in the Topology Plugin.
  - RELOAD defines three methods for overlay maintenance: Join, Update and Leave. The one hop lookups plugin defines the contents of those message.
  - Based on the architecture of RELOAD to support different usages.

# Peer data structure

- Routing table
  - A full routing table contains information about every node in the overlay
- Predecessor and successor information
  - Neighborhood information
- Unit leader information
- Slice leader information
- Slice leader list



# Updates message

- The definition of update messages based on the event notification
  - *enum { eventUpdate (0), dataStructureUpdate (1), (255) }*  
*UpdateType;*
- Two kinds of update
  - **Event update / notification**
    - Keeping the accuracy of routing tables
  - **Data structure update**
    - Mainly used in the transferring of data structure during the peer joining procedure

# Updates message

```
enum { keepAlive (0), ordinaryPeerChange (1),  
      EventType (2), sliceLeaderChange (3), (255) }  
EventType;
```

```
(255) }  
enum { peerJoin (0), peerLeave (1), peerChange (2),
```

• The `ChangeType` and `ChangeType` can construct all

- Keep-alive
- Ordinary peer join / leave
- Unit leader join / leave
- Slice leader join / leave
- Slice leader join / leave



- Update request

- ~~Update request~~ Update request is composed of two kinds of list
  - The update request is composed of two kinds of list
    - Event notification list
  - Data structure list
    - All the peers in the overlay can use this Update request to inform event notification or transfer routing

information

- 

- It is only has a response number to represent the receiver's attitude which may include success, fail

# Leader choosing strategies

- Default choosing schema
  - Dynamic choosing the successor of the mid-point
    - Default choosing schema
- ~~peer in the slice or unit space~~  
SandStone

SandStone

like schema

- Identify well connected and provisioned peers as “Super Node”, all the super nodes form a parallel ring and do not participate in the routing procedure
- The super node can act as a slice leader whose work is collecting the notifications and spreading them in time

# Fault tolerance

- First hop fail
  - ~~First hop fail~~ If a query fails on its first attempt, the receiver can respond a *RouteQueryAns* message to give a closer
  - If a query fails on its first attempt, the receiver can respond a *RouteQueryAns* message to give a closer
- In most of cases, two hops are enough to locate one peer or resource
  - Leaders fail

Q&A?