

COMS 6181 Project

RELOAD P2PSIP

Jehanzeb Khan (jk3305)
Dainis Kiusals (dvk2102)
Alejandro Mesa(am3473)

Overview

Client/Server based protocols are mostly dominant in the end-systems and/or B2B/B2C applications

Peer-to-Peer based protocols are more dominant in the Intermediate systems such as routing protocols

Pros and Cons of each of these architectures

Peer-to-Peer protocols in the end-systems

Motivation and Challenges

Scope of our project

Storage Layer

Simplified basic unit of stored data based on subset of RELOAD
StoredData structure representing the stored time, lifetime and value.

Single value Kind-ID data model implementation.

Security (signatures, access policy) not addressed in current
implementation.

Functionality for Store (adding and removing values) and Fetch
requests and responses handled by current protocol layer
implementation. At this time only non-replica stores are handled by
the Storage Layer protocol.

Stat and Find request and response functionality not currently
implemented.

Topology Plugin

The Internet Layer[OSI] equivalent of P2PSIP Reload

Functions of Topology Plugin and its interfaces to other layers

- Chord Algorithm and its details
 - Joining a Node
 - Distribution of keys to nodes
 - Routing messages
 - Populating finger table
 - Stabilising/ keeping pointers up-to-date
 - Fixing entries in the finger table

Inter-Overlay routing ?

Location/identification issue ?

Message Transport

Simplified Message Transport layer developed as gateway for handling incoming and outgoing messages

All message communication to and from the Forwarding & Link Management Layer goes through the Message Transport

Uses queuing mechanism to distribute incoming message to all layers

Does not provide reliable transfer of message per RELOAD RFC specification

Forwarding & Link Management

Simplified functionality implemented to handle sending and receiving messages through multiple connections

Handles parsing and message validation

Handles all RELOAD corner cases except wild card node logic

API can be easily extended to comply with RELOAD spec

Test Results

Test Scenario	Results
Initiate single-node overlay	Overlay starts on its own without any failures
Join second node to overlay	Second node joins the overlay, finger table is update and stabilization occurs
Join third node to overlay	This was not tested
Perform a Store Request and Answer	Storage data is exchanged as expected
Perform a Store Fetch and Answer	Fetch Request and response are delivered without issues
Route message through overlay	Two clients attach to the overlay and a message from client A is successfully routed to client B

Next Steps

Add Storage functionality for Stat and Find requests/responses, storage replication, handling of array and dictionary Kind-ID data models, and security (signatures).

Add Forwarding and Link Management functionality for handling of wildcard nodes and updates to the connection table based on topology plugin route updates.

Add Message Transport functionality to handle reliable message transfer of messages (multiple retries).

Add Topology Plugin functionality to validate finger table updates and to forward route updates to the Forwarding and Link Management Layer.

Add enhancements to enable functionality behind firewalls.