CS 6390 – Advanced Computer Networks

Final Project – High Level Design Document

Members:

Dhruv Arora (dxa141530)

Hemant Malik (hxm142630)

Language to be Used:

Java/Python

Modules to be Used:

Not yet decided

Design:

System has three types of nodes – Fog Nodes, IoT Nodes & a Cloud Node.

1. Fog Node:

- A fog node is a server.
- It will receive requests from client (IoT nodes) and process them.
- They can collaborate with other fog nodes, using various rules.
- Each Fog node has:
  a) An IP address
  b) A Hostname. (optional with above)
  c) A UDP port number.
  d) A TCP port number.
- Communication Rules:
  a) Used UDP for communication with client (IoT nodes).
  b) Uses TCP to communicate with other fog nodes (for tasks c & d).
- Main tasks:
  a) Receive requests from IoT nodes.

b) Processing the requests & sending back response.
c) Forwarding these requests depending on its processing queue & Max_Response_Time to fog nodes or cloud.
d) Periodic response time update messages among the fog nodes.

2. IoT Node

- The IoT node is acting as the client here.
- Will generate requests for the Fog nodes.
- Each request will contain the following:
  a) Packet Sequence No
  b) Request Type
  c) Forward Limit
  d) IP address of the request generator.
  e) UDP port of request generator.
- A request can be forwarded among the fog nodes at most Forward_Limit times.

3. Could Node

- The server nodes.
- Processes each request very fast, about 100x faster.
- Has infinite queue length.

Communication Rules:

- New request $\mathbf{R_{new}}$ arrives at a fog node **A**.
- Each request has a ***processing delay***.
- Decision on processing or forwarding a request by a fog node, will be based on the ***expected response time*** for the request.
- ***Queuing time*** will be given by the summation of the processing times of all the currently queued requests at A.
- ***Expected response time*** of $\mathbf{R_{new}}$ can be calculated by adding up the *queuing delay* and the ***processing time*** of $\mathbf{R_{new}}$.
- When a request is up for processing at a fog node, the processing operation should be simulated by having a thread sleep or wait for ***processing time*** of the request.

- Once this time elapses, a response should be created and sent back to the request originator IoT node.

- When the corresponding response arrives at the IoT node, the IoT node will print it out. The information should clearly indicate which fog nodes are visited and what each fog node did on the request.
- If a fog node decides that it is too busy to accept this new request (which will be defined by an input parameter of ***Max_Response_Time*** at each node), it will check if it can forward the request to a neighbor or not.
- If the request has not reached the ***Forward_Limit***, the fog node will identify the best neighbor among its 1-hop neighbors and forward the request to that neighbor.
- We need to keep the id's of the visited nodes in order to prevent looping.
- Selection of the best neighbor depends on the ***queueing time*** of the 1-hop neighbors.
- This information needs to be periodically exchanged among 1-hop neighbors so that they can use this up to date information to decide on their best neighbors to forward a request when needed.
- If a fog node cannot accept a new request and the ***Forward_Limit*** is reached for the request, the fog node will need to forward this request to the cloud.
- When a fog node needs to forward a request to the cloud, it will put the request in the cloud queue which will incur the necessary processing delay (by using a sleep or wait system call for the duration of processing time recorded in the request message) and then will send the response directly to the request originator IoT node.

Implementation:

The main functions which will be written:

i. Periodic response time update messages among the fog nodes.
ii. IoT node to fog node request communication.
iii. Forward or process the request question answered by the fog nodes.
iv. Fog to Fog offloading.
v. Process the request by creating a thread, and waiting till the time it finishes.
vi. Sending a response back to IoT Client.

vii. Printing of the message body of the response received.

Bean Classes to be created:

i. **Cloud Node:**
   It's fields ->
   a. IP address
   b. TCP port no.
   c. Queue

ii. **Fog Node:**
   It's fields ->
   a. IP address.
   b. TCP port no.
   c. UDP port no.
   d. Queue (of requests)
   e. List of neighbors & IPs + port
   f. Max_Response_Time
   g. Queueing Time

iii. **Request:**
   It's fields
   a. Packet Sequence No
   b. Request Type
   c. Forward Limit
   d. IP address of the request generator.
   e. UDP port of request generator.
   f. Message
   g. Response Time

Some more operations which will be carried out:

a) There will be a single server known as cloud node, whose IP ever fog node will know before-hand.
b) This cloud node processes requests 100 times faster and has an infinite queue.
c) Creating fog nodes or socket server with the requested IP/name.
d) The network topology will be entered either one by one through the console or a network configuration file.
e) Creating the network topology. Each fog node maintains a table of list of its 1-hop neighbors and the IP & port used to connect to them.
f) The requests generated will be given in the console.

g) This line from console is turned into a request object and forwarded to a fog node.
h) The fog node receives the request and calculates the expected processing time of the request and compares it with its Max_Response_Time and decides to process the request or forward it to its 1-hop table.
i) Each fog node decrements the forward limit by 1 when we forward the request.
j) If forward limit is = 0 OR the all the fog nodes have been visited and still request can't be processed, request is given to cloud node.
k) For processing of a request, we create a thread and run it for the processing time of the request.
l) A response is sent back to the IoT node after completion of the thread and its message body is printed.
m) Message information should clearly indicate which fog nodes are visited and what each fog node did on the request.
n) Periodically response time update messages are sent among the fog nodes