Terms

P2P - Peer to Peer
CDN - Content Delivery Network
CAN - Content Addressable Network
Sub-overlay - group of peers with the same channel/content

Constants

k -number of CDN servers
CDN-Bandwidth = 100 Mbps
SuperPeer/Regular Peer Bandwidth Range =

Overview of Protocols

Location-Aware – use of CAN which employs the binning technique; uses RTT as the basis for location
Content-Aware – sub-overlays for peers in the same bin that stream the same content
RTT and bandwidth – select peer with lowest RTT and compatible bandwidth

Entities

Node

- nodeTag(int)
  - 0 = CDN
  - 1 = SuperPeer
  - 2 = Regular
- peerlist – who it has connections with (list of node ids of its peers)
- clientlist – applicable to its superpeer (list of clients in its CAN zone)
- CID (CDN ID)
  - CID = nodeID if the node itself is a CDN
  - CID = nodeID of the CDN the node is connected to
- SID (SuperPeer ID)
  - SID = nodeID if the node itself is a CDN
  - SID = nodeID of the SuperPeer the node is connected to
- maxBandwidth – randomized, depends on the nodeType
- currentBandwidth – currently used bandwidth
- RTTlist – list of RTT wrt to each CDN (landmark?)
- vidList - contains the content the node has
- * clientVidList – contains what content each of its client is watching so that when a new peer comes, sends a list of the clients/peers that stream the same category (or is it the specific content?)

Content/Video

- vidID (int)
- catID (int) – which category it belongs to
- size (int) – size of the vid/content

Simulation Set-Up

[CDN Initialization]

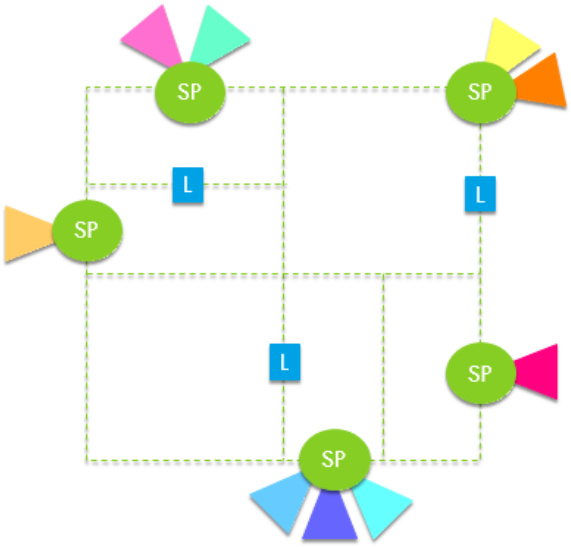
Choose k number of CDNS. Set the nodeTag for each to 0. The first k nodes, with ids 0, 1, 2 … k are chosen as the CDN servers. Set the bandwidth of the CDN nodes to 100 Mbps.

[Regular Peer Initialization]

Set bandwidth to 125 Kbps, upload speed to ___ and download speed to _____. Randomize RTT values ($30 \leq RTT \leq 100$) with respect to each CDN (landmark?). Set its peerlist to empty first. Randomize a video list for the node (i.e. the content it has; for each content assign a category)

[Binning]

Use the RTT values of each node to each CDN to determine which CDN it belongs to. Those that have the same CDN as a result of their RTT belong to the same bin.



[Initial Connections]

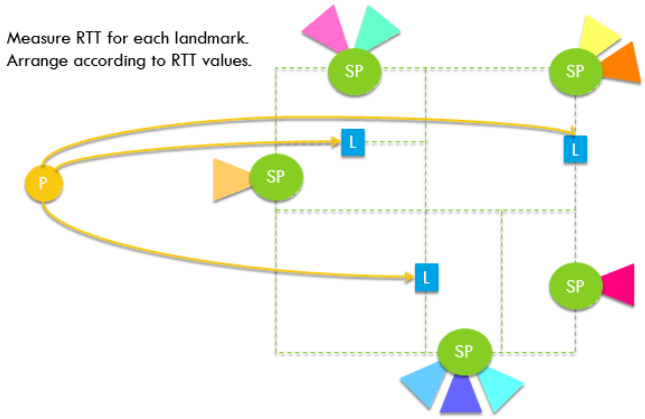
Eottoke? T_T Assign content first then, make the connections in the bin based on that or randomize connections then assign content for those that are connected.

[SuperPeer Initialization]

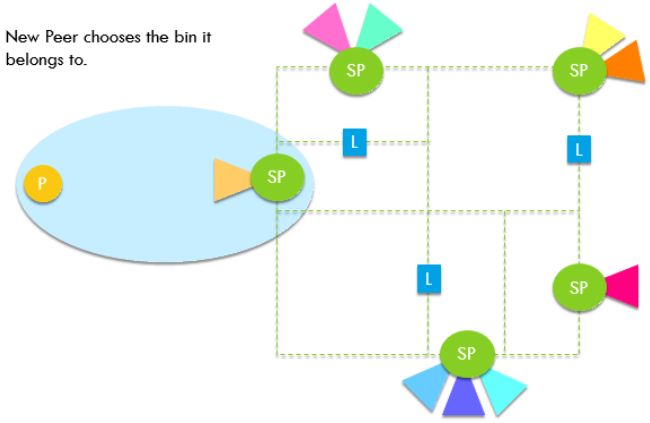
For each bin, order the RTT values of the peers in it wrt to the CDN of the bin, the one with the lowest RTT is then the SuperPeer. Change its nodeTag from 2 to 1. Populate its clientlist with the peers in its bin. Create the clientVidList.

[Protocol]

- If a new peer joins

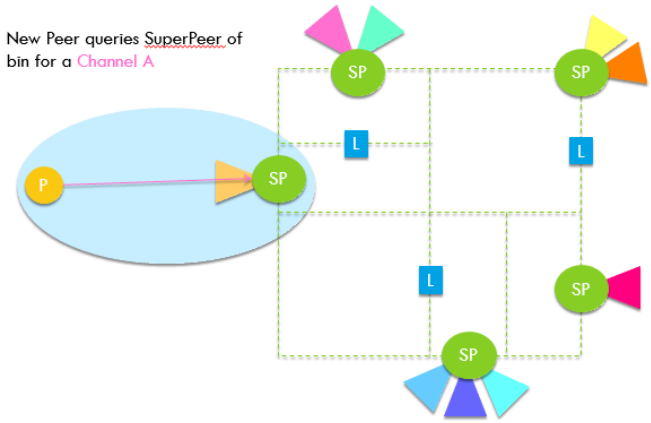


  - Check its RTT wrt to the CDNs to see which bin it belongs to.



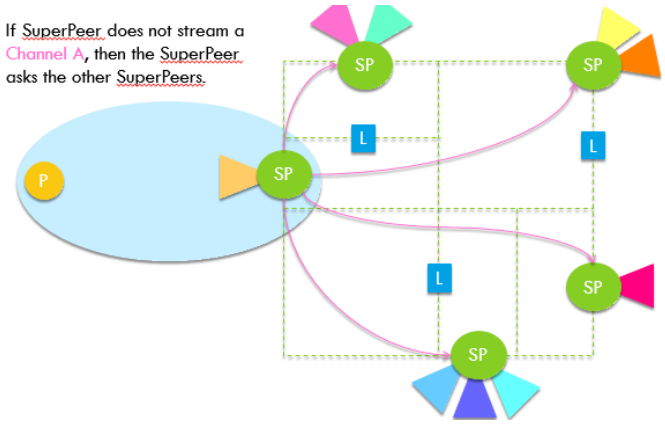
  - Send a message to check if its RTT wrt to the CDN is smaller compared to the current SuperPeer. If yes, make it the SuperPeer.Send message to all peers in the bin that there is a new SuperPeer. Send a message to the old SuperPeer to become a RegularPeer .

- A Peer requests for content



Send message to SuperPeer about content requested. SuperPeer returns a message containing the list of peers that have the content. Use AEPS to select peers from there.

For AEPS, ho out the RTT to other peers? T_T



If the content is not shared by the peers in the bin, forward its request to other SuperPeers.

Simulation Notes: Bryan Version

If multiple CDN. node 0,1,2 ang id
pag sa main CDN ng peer nakaconnect, 30-100RTT otherwise 1k-1.5k
pagkajoin ng new peer, send siya ng message to CDN ng RTT niya. then reaction ng CDN e icheck kung siya smallest, if yes send ng message na siya yung new superpeer and send din sa old SP na may bago ng superpeer

Landmarks??? -- presolutiom: randomized RTT ng peers sa landmarks then compute kung saan ba siyang bin. So parang grid.

Protocol:

so una, random kung saang CDN yung peer

tas maglalagay ng random RTT sa CDN server niya.

si peer e magsesend ng RTT value niya kay CDN, ilolog ni CDN yun and ichecheck niya kung pinakamababa ba yun

if yes, new SP. send siya ng SP signal kay peer. iaacknowledge ni peer, then send ni CDN yung list ng peers. Send din pala kay old SP na hindi na siya SP. tas send sa peers yung id ng new SP

compute landmarks, each bins dapat may id. so depende sa nacompute niya e makukuha niya yung id ng bin, then ask kay CDN kung sino ano yung id ng SP

contact kay SP na magjoin siya. then acknowledge ng SP, tas yung request vid na

problem: paano yung RTT niya to each peer?

check ng SP kung meron ba siyang vid na ganun? if meron, return ng list. else send ng fail, then request uli si peer, this time ipapasa ni SP yung request to other SPs.

take note: so pinakakailangan talaga dito e yung message passing protocol(EDP)

so after makakuha ng list, measure ng AEPS.

globals ng protocol:
max bandwidth niya.
used bandwidth
RTTs
vid ID ng pinapanood(problem: multicasting?)
length ng pinapanood (in bytes)

another problem: quality of vids? so pwedeng imeasure to kung hanggang anong speed yung inaabot e yun yung max quality na kayang iprovide ng architecture

CDN:
sa CDN same dito, node chuchu^
connect kay CDN, check kung may nagsstream na nun. kung wala kay CDN.
una, request kay CDN ng list ng peers
ang ibibigay lang ni CDN e random 20 peers
kung ilan ang kulang ng peer na download speed e connect kay CDN

problem: ilang bandwidth iaallot ni CDN per vid

facts, yung mga nakaconnect kay CDN e kapag tapos na sila magstream, hindi sila kailangang necessarily tanggalin.

problem: after nila magstream e gaano pa sila katagal magcocontribute?