15-441/641 Homework #4
Due November 18, 2019 at 5PM to Gradescope
October 29, 2019

Part 1: Token Bucket

In your new job at an ISP you are responsible for managing the traffic enforcer infrastructure.

On each ingress link of the ISP, there is a traffic enforcer that makes sure customers only send traffic at an agreed upon rate. This enforcer is implemented with a token bucket, as presented in class, where each token gives you the right to send a packet.

There are three ingress links (each with their own token bucket) that connect up to a shared link with a queue served in FIFO order. The shared link is served at a rate of one packet every 10ms (100 packets/sec).

There are three flows with the following Token Rates (Tokens/sec) and Bucket Sizes (packets):

- Flow 1: 1 token/sec and 8 packets
- Flow 2: 2 tokens/sec and 5 packets
- Flow 3: 4 tokens//sec and 2 packet

The space below is left open so you can draw a figure (not graded).

1. (5 points) What is the maximum delay a packet might incur waiting in the FIFO queue?

2. (10 points) You reduce the bandwidth serving the FIFO queue by a factor of 10, i.e, you can forward a packet every 100 milliseconds (10 packets/sec).

Now consider the following scenario: at time T, packets are arriving at all three token buckets at a very high rate -- faster than the token buckets can release. In the two second time window that starts at time T, i.e., [T, T + 2 second], what is the minimum number of packets (or a small range) from the third sender that will make it all the way through the system – i.e., get released from the FIFO queue?

Part 2: HTTP 2.0

HTTP 2.0 has four new features compared with earlier versions of HTTP:

- Responses are multiplexed across multiple prioritized streams inside one TCP flow.
- HTTP headers are compressed
- PUSH allows the server to send web objects to the client without waiting for a GET request.

For each of the following types of web transfer, please list the HTTP 2.0 feature that is most likely to speed up the transfer of the web page, compared with using a single HTTP 1.1 session (1 TCP connection). Explain why if none of the features is going to help.

1. (5 points) Transfer of a video chunk

2. (5 points) A simple, static web page that includes a small number of large images.

3. (5 points) Same as 2., but the web page is customized to the screen size and resolution of the client device by adapting the size and encoding of the images. This is done using a javascript running in the browser.

4. (5 points) A web page that shows a photo album as a large grid of thumbnails (very small images of the photos in the album).

Part 3. A peer-to-peer DNS service

A CDN has its content strategically distributed across 100s of CDN server clusters distributed across the Internet to minimize RTT to many clients. It uses DNS redirect but decides this is too slow and instead wants you to explore using a peer-to-peer architecture for name resolution and for redirecting clients to a CDN cluster that has the content.

The idea is simple: web objects of customers (content providers) are randomly placed on one or more CDN cluster across the Internet to achieve load balancing. Newer objects that are likely to be more popular are replicated on 10-20 nodes, while older objects are replicated on a smaller number of nodes, e.g., 1-5. The CDN clusters are organized as a peer-to-peer (overlay) network.

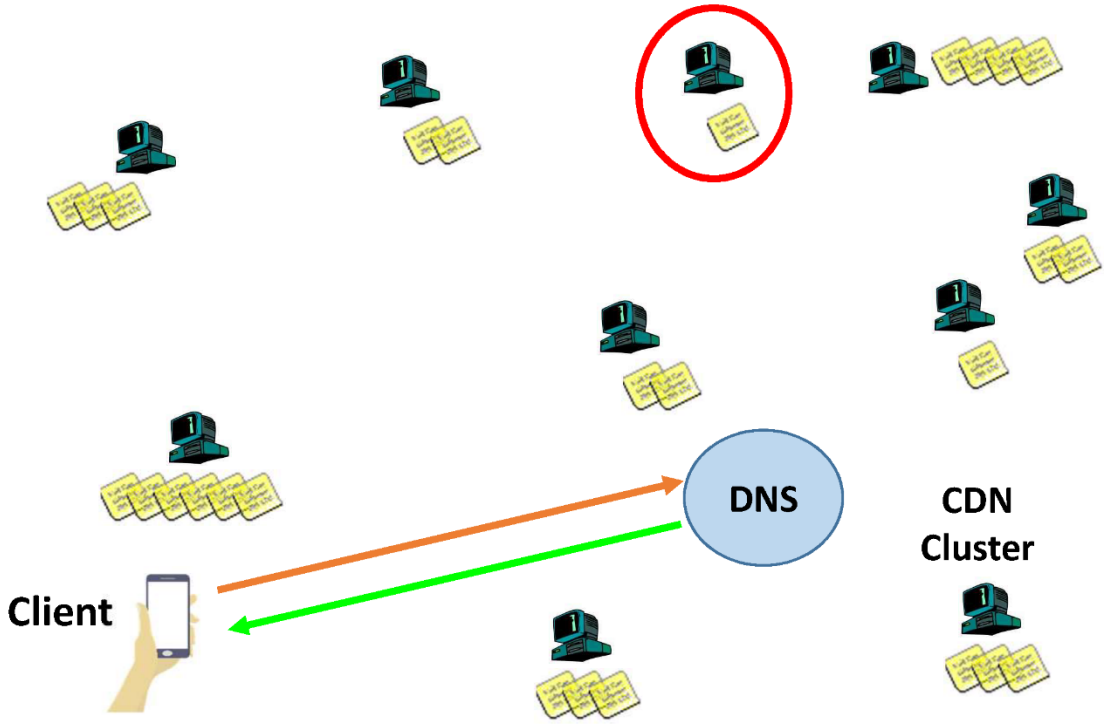
Clients are associated with a random CDN cluster (preconfigured), and any GET request for a web objects of one of the CDN's customers (which can be recognized based on the URL) are forwarded to that CDN cluster. That CDN cluster will serve the content if it has it. Otherwise, it does name resolution by using the peer-to-peer network to identify a peer CDN cluster that has the web object (circled in red in the figures) and returns its IP address to the client.

The CDN is considering three peer-peer designs, Napster, Gnutella, and KaZaA. The content provider asks you to compare the performance of these three peer-peer designs and its DNS redirect solution using two metrics:

(1) The *number of RTTs* needed before a client node can start establishing a TCP session with the CDN server cluster that has the object. You should assume that the RTTs to the DNS server, the contact cluster and to any node managing the peer-peer network are all roughly the same.
(2) Degree to which the selected cluster is the best choice, given the example criteria discussed in class.

Please comment on these two metrics for the three peer-peer architectures and DNS redirect. For the first metric, list the possible RTT counts, describe when they happen, and mark them on the figure. For the second metric, describe qualitatively how likely it is that the system will pick a good cluster.

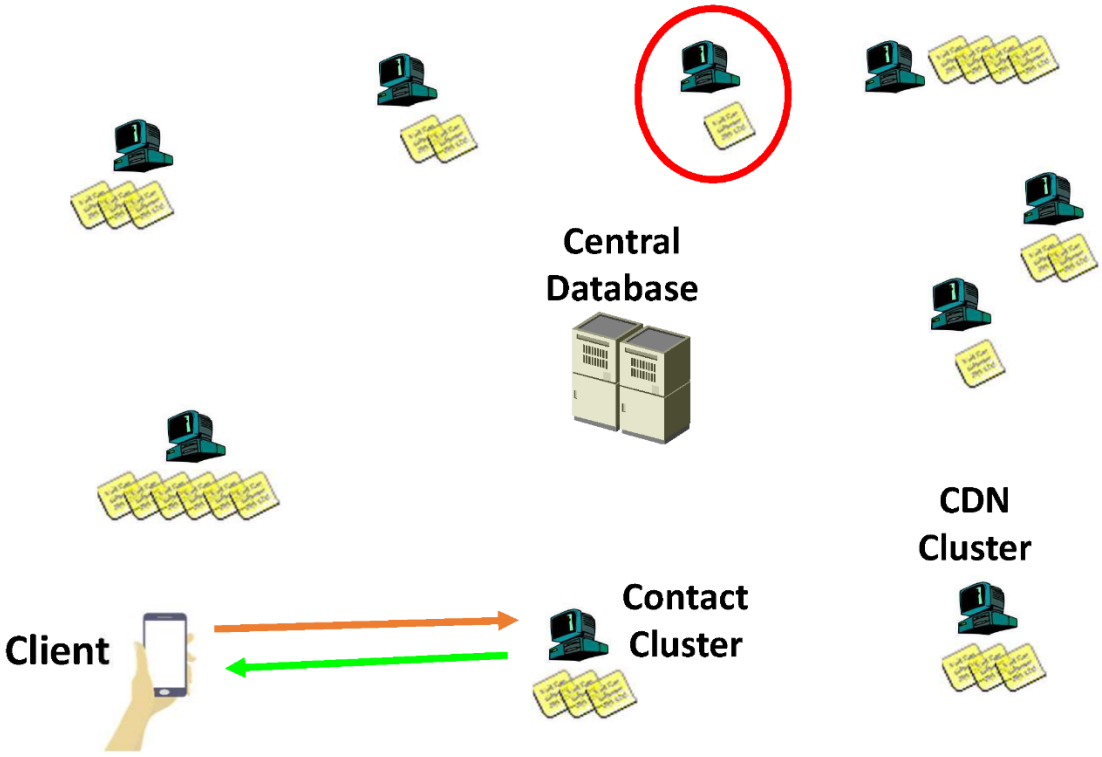
a. DNS redirect.



1. (3 + 2 points) Number of RTTs

2. Quality of the selected cluster

b. Napster.

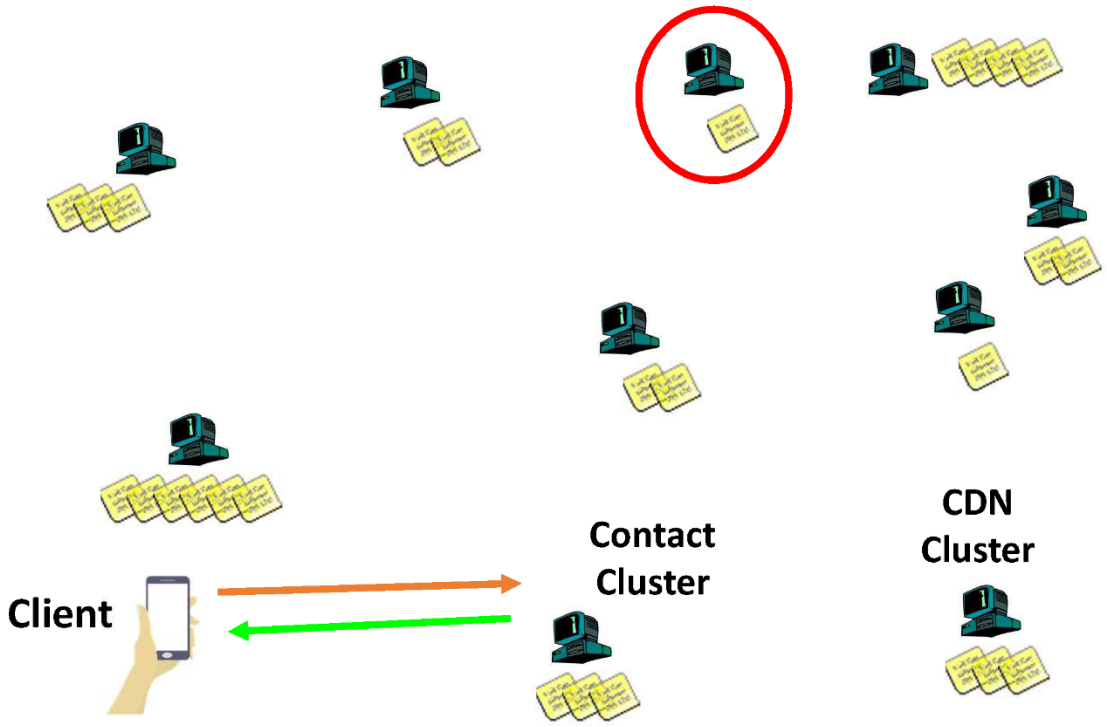


1. (3 + 2 points) Number of RTTs

Solution:

Solution:

c. Gnutella.



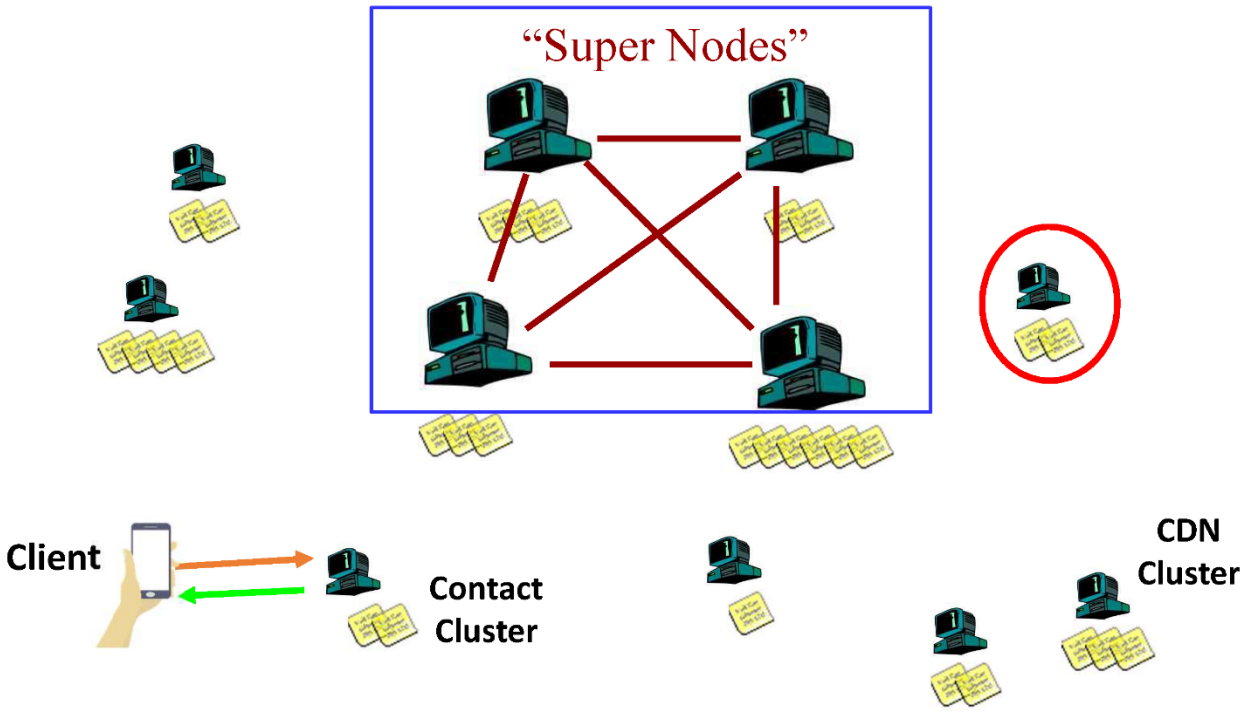
1. (3 + 2 points) Number of RTTs

Solution:

2. Quality of the selected cluster

Solution:

d. Kazza.



1. (3 + 2 points) Number of RTTs

Solution:

2. Quality of the selected cluster

Solution: