PEER5

Peer5 is a peer-to-peer Content Delivery Network (CDN) which is powered by WebRTC.

It is a serverless streaming CDN.

What is a CDN ?

-

A Content Delivery Network is a network of geographically distributed servers that delivers web content to different users based on their geo location.
This results in speeding of delivery of the content of websites with heavy traffic.
Server nearest to the website visitor responds to the request.
CDN copies the page of a website to a network of servers that are dispersed at different geo locations.
When a user request a webpage which is a part of the CDN, the request is redirected to the server in CDN which is nearest to the user.
CDN also communicate with the origin server to cache all the content to different servers in the CDN. If the content is not available in CDN, the request goes to the origin server.

Basic working of Peer5 :

-

Peer5 works side-by-side the origin server and CDN.
It uses WebRTC to create a peer-to-peer network among the users which help one another for their content loading.
Peer5 switches in between Peer5's p2p network and origin server or content delivery system to load the content.
As a result, it reduces origin server's bandwidth usage.
Peer5 p2p network is built on top of HTTP client-server model for serving requests, so it increases reliablity.
Peer5 CDN fetches a stream from other people watching the same content. By retrieving streams from closer to the viewer, Peer5 loads the content faster.
Simultaneously a stream is also being transmitted to the Peer5's p2p network.

Peer5’s service is activated using their JavaScript API in the browser.
It connects multiple peers which are viewing the same content using the basic WebRTC p2p logic.
It connects peer also to the origin server(http) to ensure delivery of the content in case no peers are found. So, we can say that the data stream is delivered by the origin server and the peers connected both.
Peer5 dynamically creates a mesh network of peers and the origin server which improves its delivery power.

Basics of streaming :

-

Streaming protocols like HLS (HTTP Live Streaming) and MPEG-DASH (Dynamic Adaptive Streaming over HTTP) works on the concept of segmentation.
A piece of content is cut into small slices of video called segments. The basic idea is that you are downloading a large number of small files instead of a single large file. There is a manifest file which tells the video player from where to download the next segment. It contains the URL of all the segments.
The video player makes http request to fetch these segments which goes to the nearest CDN server or the origin server.
Once downloaded, these segments go into buffer and are extracted by the video player as needed.

Every user in the world who is watching that content needs to connect to the CDN or the origin to download those segments.

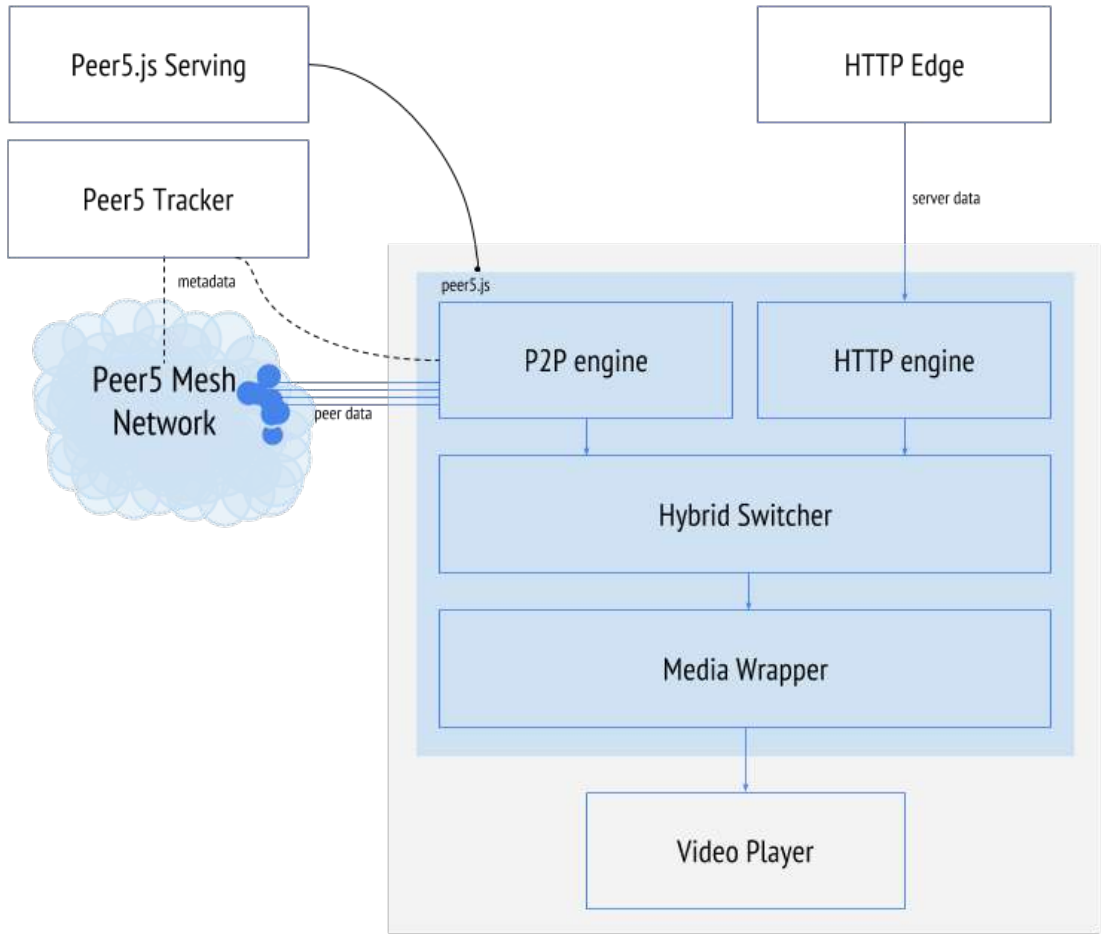
What's interesting in Peer5 :

-

In Peer5, every user doesn’t need to connect to the CDN server or the origin server to fetch the same segments of the content which his or her peer is watching.
In Peer5 p2p network, a user can fetch those segments from the other connected peer and can also help other by transmitting stream from his downloaded content as well.
Peers can share video segments to each other.

PEER5

How Peer5 makes this possible ?

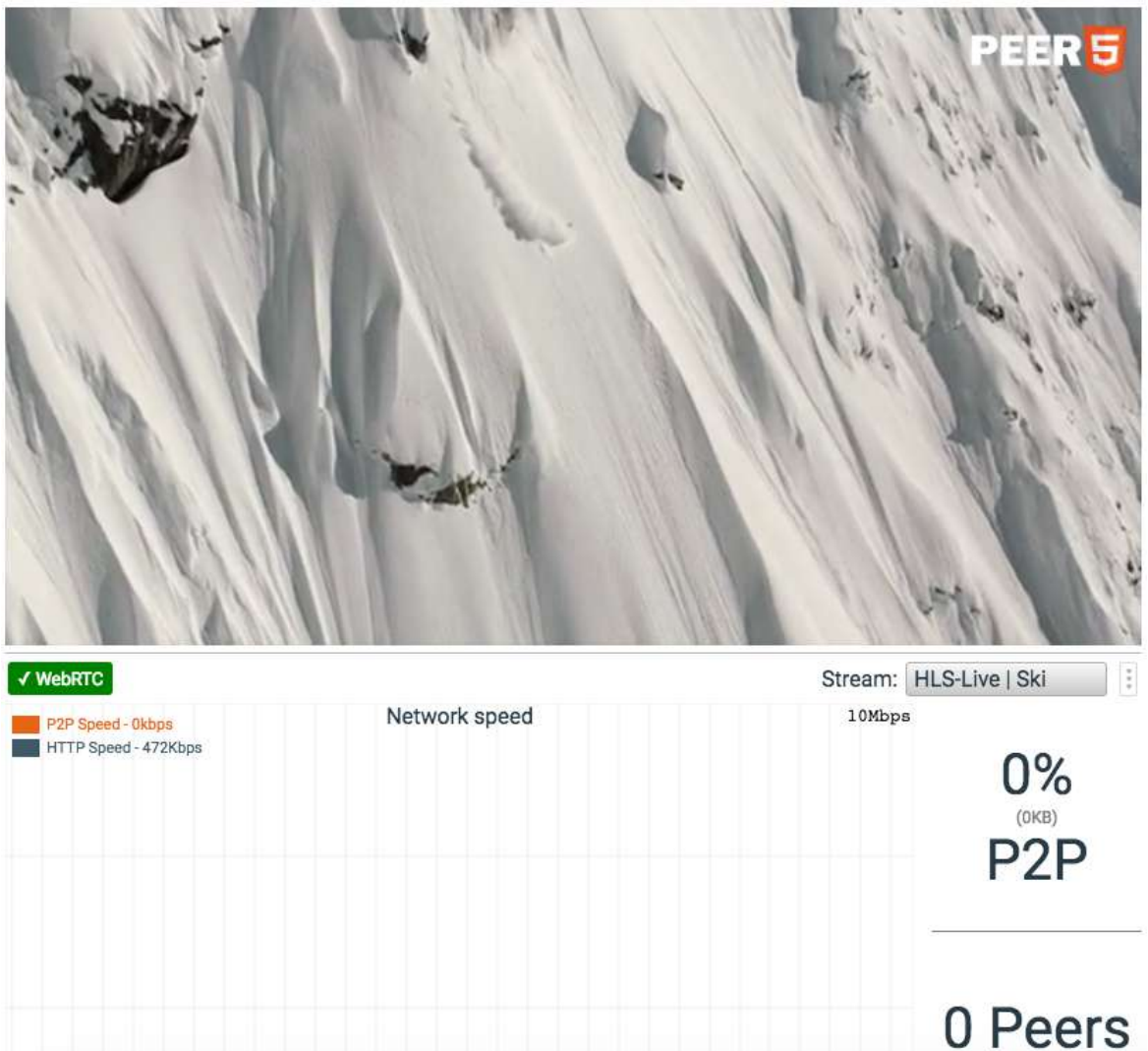


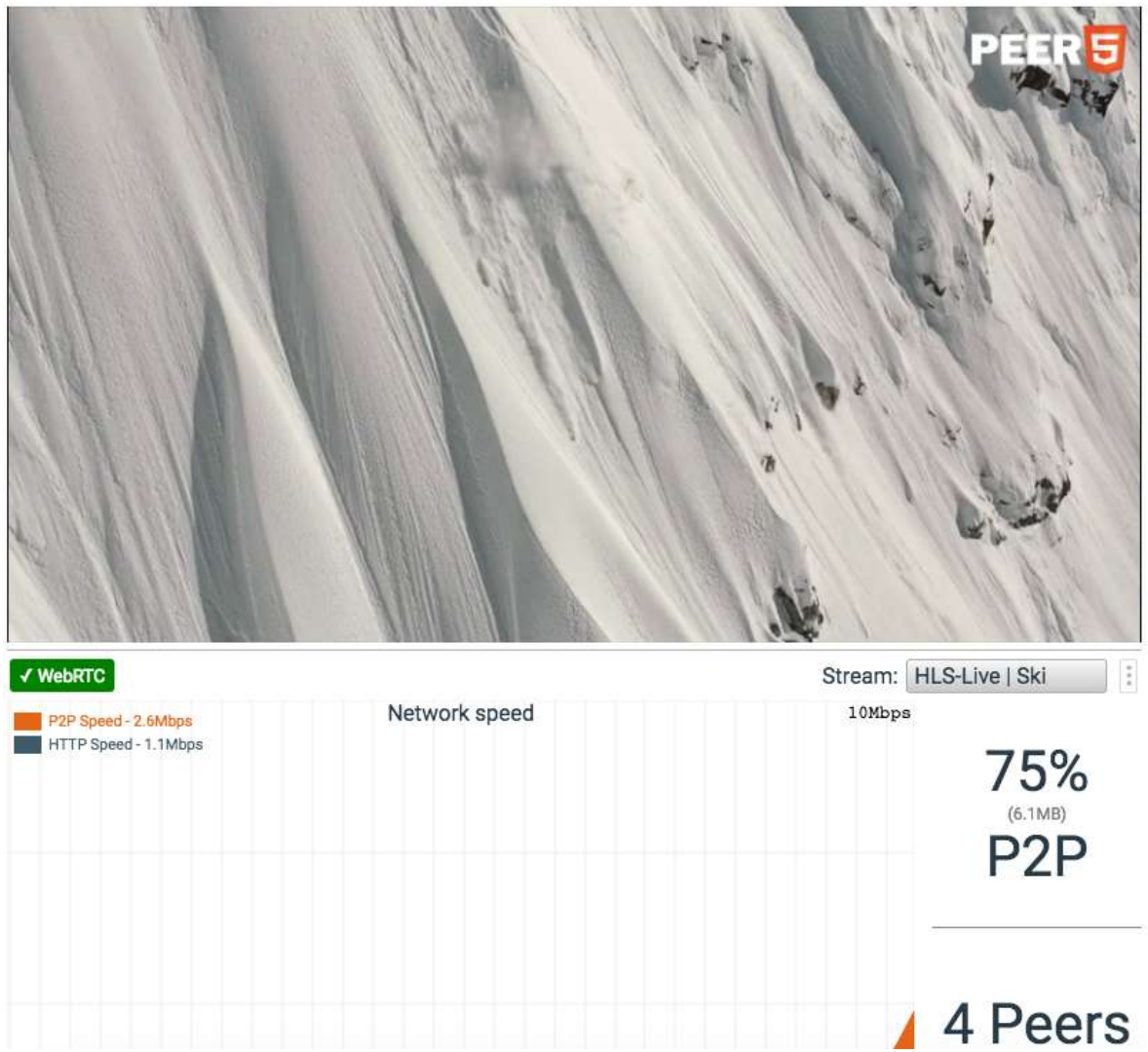
Peer5 works by creating a full mesh p2p network of users who are watching the same content.

First the Peer5 JS API is loaded in the browser which then communicates with the back-end code telling it that there is a new user who wants to watch this stream.
At this point, Peer5 creates a swarm of users who are watching that particular stream.
For p2p transfer, a user's closest peer is determined by its geo location and ISP.

Next, the hybrid switcher decides on that the content should be loaded from the p2p network or the HTTP request or its combination and then passes the stream to the media wrapper which directly connects to the video player on the page user is viewing.

In Peer5, more the number of peers, higher the speed of content loading.





THANK YOU

- HITESH KUMAR