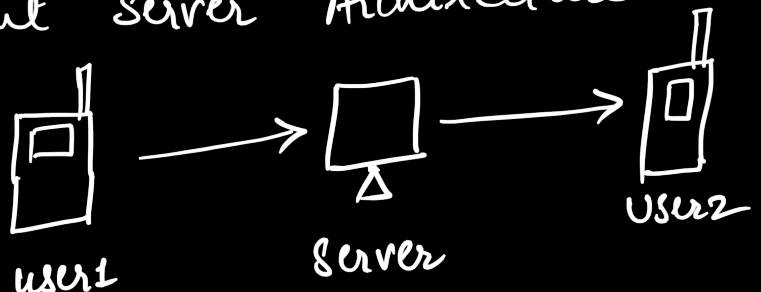
How Discord handles millions of Concurrent voice users?

YouTube Channel- MsDeep Singh

How can people interact on channel?

① Peer to Peer connection

② Client Server Architecture



{interaction with server in between}

Discord uses #2, WHY?

① Peer to peer networking becomes expensive as no of users↑

② Routing network traffic via Discord servers ensures that user IP address is never leaked.

↳ helps in preventing to discover user IP address and launch DDOS attack against user.

③ Moderation. Ex- Admins can disable audio/video for offending users.

Client Architecture →

Discord uses webRTC - web Real Time Communication.
From wikipedia - It provides webbrowsers / mobile apps with real time communication via APIs. It allows audio/video communication to work inside web pages by allowing direct peer to peer communication, eliminating the need to install any plugins/download native apps.

NOTE → we discussed peer to peer communication is SLOW... if userpool is large

↳ so How does webRTC work for DISCORD?

↳ via custom implementation on top of webRTC.

Some Internal implementation details

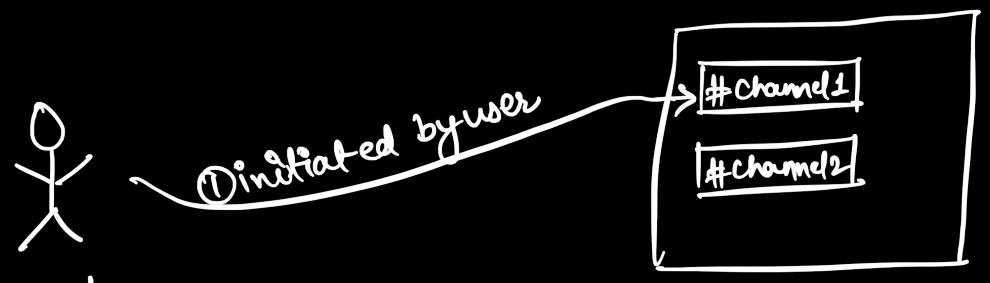
1. WebRTC relies on the Session Description Protocol (SDP) to negotiate audio/video information between users. Its size is ~10KB for round-trip.
   i. Discord utilize lower level API webrtc::Call to create send/receive streams and reduce the information that is to be exchanged - server IP/port, encryption method and keys, codec and stream identification ~1000Bytes.

2. WebRTC uses Interactive Connectivity Establishment (ICE) to determine the best communication path between users. Since every client connects to our media relay server, ICE is not needed.
   i. It helps in providing more reliable connection when you're behind NAT, also keep your IP secret from other parties in channel.
   ii. Clients send ping msgs at fixed interval to ensure firewall remains open.

3. WebRTC use Secure Real-time Transport Protocol (SRTP) for media encryption. The encryption keys are set up using Datagram Transport Layer Security(DTLS), which is based on the TLS protocol. You can also implement own transport layer via webrtc::TransportAPI.
   i. Salsa20 encryption is used.
   ii. No audio data is send in periods of silence, helps in saving bandwidth and CPU usage.

Backend Architecture

How communication is initiated ?



user/ client

Discord Server or Guild
↳ combination of channels & users

1. written in Elixir. Main backend services - Discord Gateway Server[DGWS], Discord Guilds Server[DGS] and Discord Voice Server[DVS].

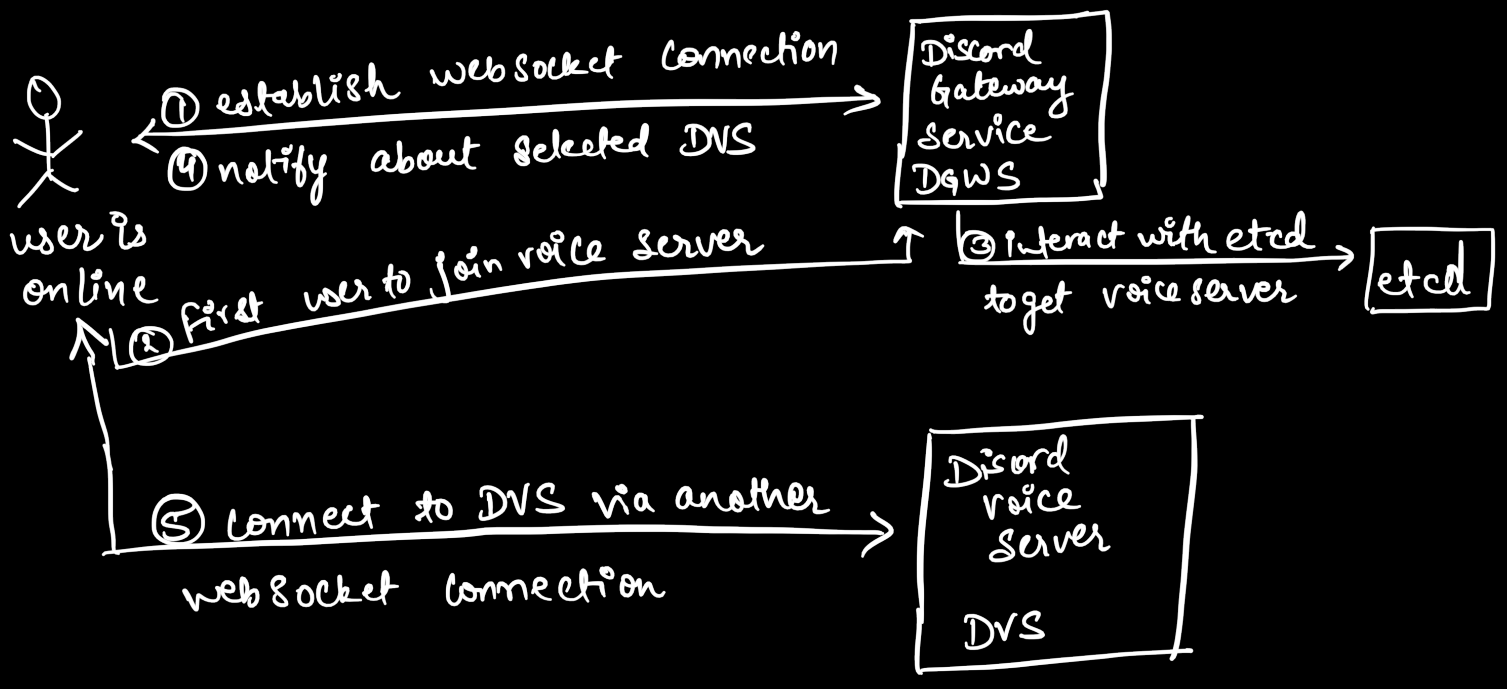
2. When you're online, client maintains websocket connection with Discord Gateway. Any event related to guilds, channels, messages etc is sent via this.

3. When you join voice server, you're assigned Discord Voice Server
   i. [responsible for transmitting every memebers' audio to channel].

   ii. All voice channels within guild are assigned same Discord server. It helps in efficient iteraction with reduced latency, since all users are present on same server.

   iii. If you're first voice participant, DGS assigns Discord Voice Server to guild.

   iv. Once you're connected to DVS, connection status is represented by voice state object.



How voice server is assigned ?

1. DVS health and load is maintained via Service Discovery System [etcd - we discussed the same is our last Discord System Design videos]

2. DGS looks in etcd and assign least utilized server to guild.

3. After DVS selection, voice state objects are pushed to DVS so as it knows how to setup audio/video forwarding.

4. Clients are notified about selected DVS.

5. Client opens second webSocket connection to DVS, used for setting up media forwarding and speaking indication.

More about Discord voice server

It contains two components
1. Signalling component - controls the SFU and generates stream identifiers and encryption keys, forwards speaking indication, etc.
2. Media Relay Component - called Selective Forwarding Unit(SFU).

SFU is also responsible for handling real-time control transport protocol (RTCP), which is used for video quality optimization. It collects and processes RTCP reports from receivers and notifies senders how much bandwidth is available for video.

Discord voice Server Failover

Discord voice server is directly accessible from internet.

1. Signalling Component monitors SFU. If it crashes, it is restarted again, could drop some packets and state is recontructed by Signalling Component.

2. Discord Voice Server dies --
  i. Removed from etcd.
  ii. Client webSocket connection is cut and it requests a voice server ping through the gateway WebSocket connection.
  iii. DGS confirms failure by consulting with etcd and assign new DVS to guild.
  iv. DGS push all voice state objects to new voice server.
  v. Every client is notified about the new DVS and creates a voice WebSocket connection to the new voice server to start media setup.

Since DVS is directly accessible from public internet, quite common to suffer DDoS attacks. - same procedure is followed as of Discord Voice Server dies.

Subscribe YouTube Channel for more such content

YouTube - MsDeep Singh

Happy Learning ☺