**Option A — Direct peer-to-peer (recommended for just 2 people)**

**0) One-time prep on both PCs**

# If you ever re-generated keys before with the same names, clear old pins:

Remove-Item .\keys\trustmap.json -ErrorAction SilentlyContinue

# Make keys (run on each PC, with that person’s own name)

python tools\gen\_keys.py --peer-id himanshu # on Himanshu's PC

python tools\gen\_keys.py --peer-id shubham # on Shubham's PC

**1) Pick who will be dialed (listen) and who dials (connect)**

Let’s have **Himanshu** listen on port **9001**, and **Shubham** dial him.

On **Himanshu’s PC** (Terminal 1):

# open firewall port once (needs admin; do once)

netsh advfirewall firewall add rule name="chat-9001" dir=in action=allow protocol=TCP localport=9001

# start node

python -m src.main --name himanshu --port 9001 --keys keys/himanshu

Find Himanshu’s IP:

ipconfig

Note the IPv4 address (e.g., 192.168.1.23 on the same LAN).

On **Shubham’s PC** (Terminal 2):

python -m src.main --name shubham --port 9002 --keys keys/shubham --peers ws://<HIMANSHU\_IP>:9001

Replace <HIMANSHU\_IP> with the actual IPv4 you saw above. (Do **not** use localhost across machines.)

**2) Test**

On either console:

/list

/msg --to himanshu --text "hello from shubham"

…and vice versa:

/msg --to shubham --text "hello from himanshu"

For a broadcast (single hop in direct mode):

/group --text "hi team!"

If you’re **on different networks / over the internet**: forward TCP port **9001** on Himanshu’s router to his PC, then Shubham dials ws://<PUBLIC\_IP>:9001. If port-forwarding isn’t possible (CGNAT), use an overlay like Tailscale/ZeroTier to get a private LAN IP and dial that.

**Option B — One PC acts as a hub (works great if you add more teammates later)**

Pick **alpha-hub** to run on Himanshu’s PC; both nodes connect to it.

**0) One-time prep**

Remove-Item .\keys\trustmap.json -ErrorAction SilentlyContinue

python tools\gen\_keys.py --peer-id alpha-hub

python tools\gen\_keys.py --peer-id himanshu

python tools\gen\_keys.py --peer-id shubham

**1) Start the hub on Himanshu’s PC**

netsh advfirewall firewall add rule name="chat-9001" dir=in action=allow protocol=TCP localport=9001

python -m src.main --name alpha-hub --port 9001 --keys keys/alpha-hub

Grab Himanshu’s IPv4 via ipconfig.

**2) Start members and connect to the hub**

On **Himanshu’s PC** (self as a member too, optional):

python -m src.main --name himanshu --port 9101 --keys keys/himanshu --peers ws://<HIM\_IP>:9001

On **Shubham’s PC**:

python -m src.main --name shubham --port 9102 --keys keys/shubham --peers ws://<HIM\_IP>:9001

**3) Nudge sessions, then chat**

From the **hub** console (optional but helps form link sessions fast):

/msg --to himanshu --text "hub->him"

/msg --to shubham --text "hub->shub"

Now DMs via hub:

# on Himanshu

/msg --to shubham --text "hello via hub"

Group fan-out via hub:

# on either member

/group --text "hello everyone"

**Common pitfalls & quick fixes**

* **Connection fails (refused/timeouts):**
  + Use the **IPv4** address of the listener, not localhost.
  + Ensure Windows firewall rule exists for the listening port (netsh above).
  + If over the internet, confirm router **port forwarding** to the correct PC/IP.
* **“fingerprint mismatch … refusing session”:**
  + You regenerated keys for the same name after it was pinned. Delete keys\trustmap.json on both sides and reconnect.
* **“No session with …” or “No session for …”:**
  + In direct mode, just send a message both ways once; sessions usually form automatically on connect.
  + In hub mode, send a tiny /msg from the hub to each member to ensure link sessions exist, then try DM again.
* **Ben/Shubham sees “[PM/GROUP from alpha-hub] …” instead of from the real sender:**
  + Make sure you kept the small patches that:
    - add from=<sender> inside MSG\_PRIVATE and MSG\_GROUP on send, and
    - print payload.get("from", remote\_name) on receive.