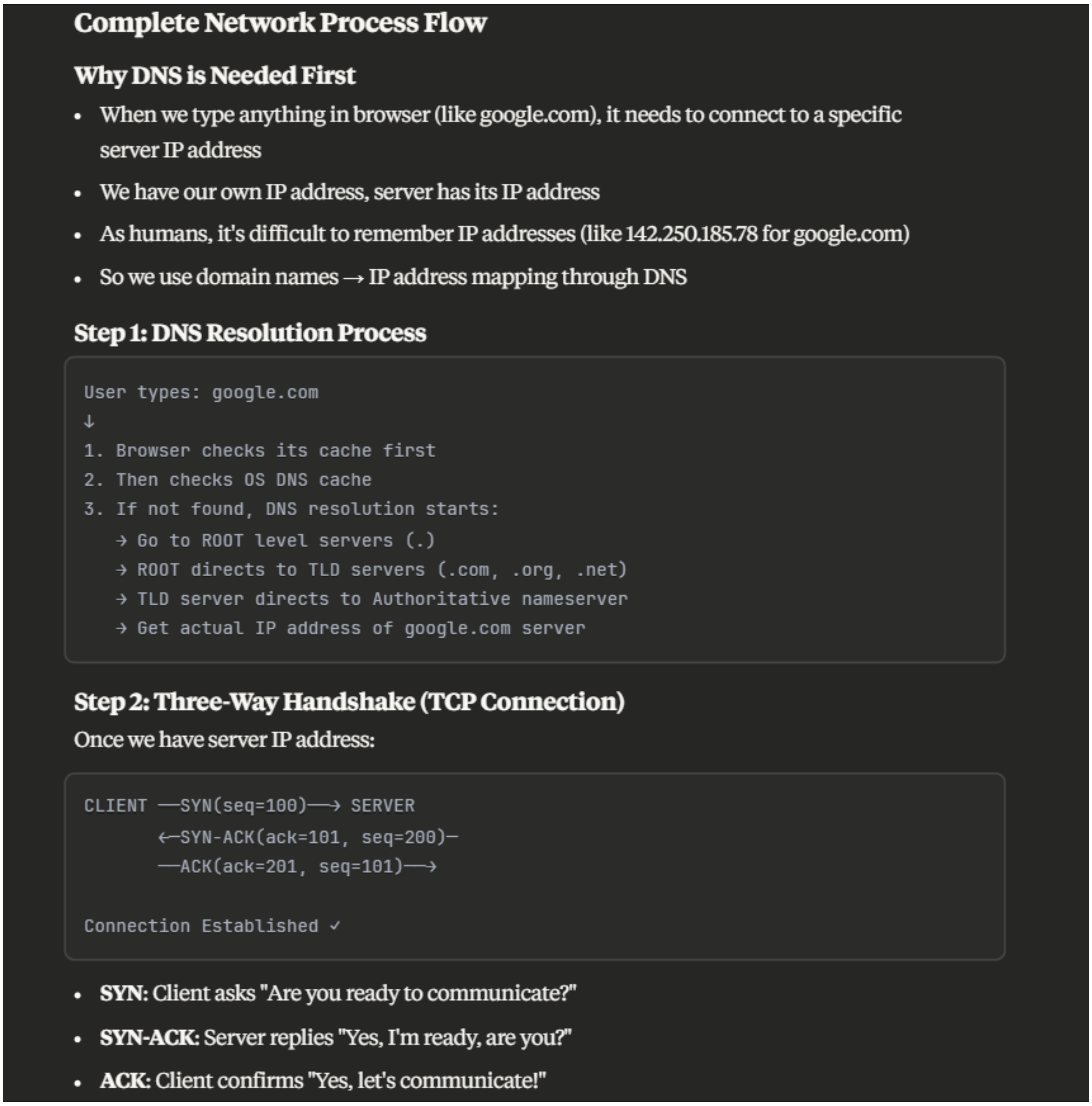


Complete Network Process Flow

Why DNS is Needed First

- When we type anything in browser (like google.com), it needs to connect to a specific server IP address
- We have our own IP address, server has its IP address
- As humans, it's difficult to remember IP addresses (like 142.250.185.78 for google.com)
- So we use domain names → IP address mapping through DNS

Step 1: DNS Resolution Process

```
User types: google.com
↓
1. Browser checks its cache first
2. Then checks OS DNS cache
3. If not found, DNS resolution starts:
   → Go to ROOT level servers (.)
   → ROOT directs to TLD servers (.com, .org, .net)
   → TLD server directs to Authoritative nameserver
   → Get actual IP address of google.com server
```

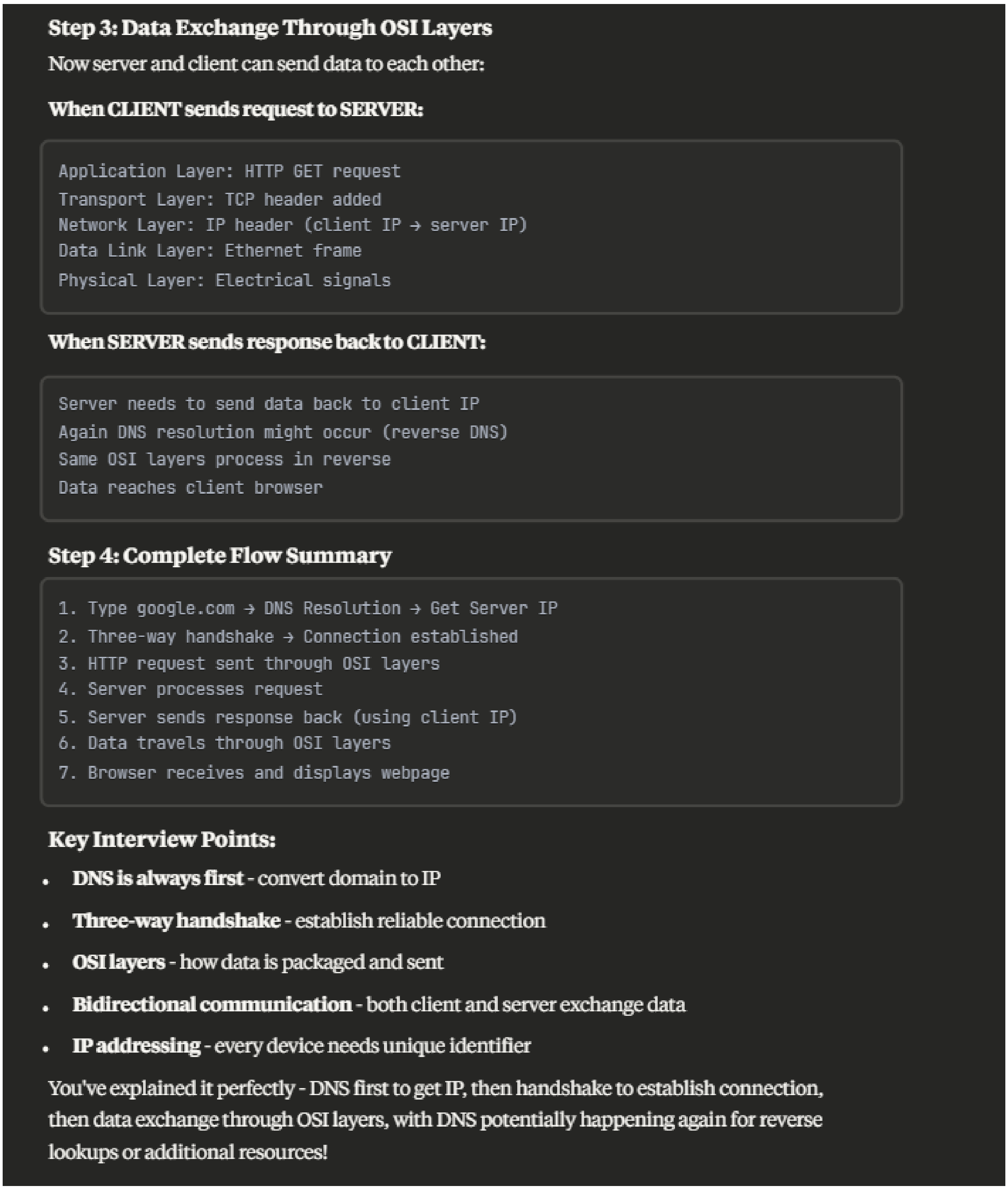
Step 2: Three-Way Handshake (TCP Connection)

Once we have server IP address:

```
CLIENT ——SYN(seq=100)——→ SERVER
       ←—SYN-ACK(ack=101, seq=200)—
       ——ACK(ack=201, seq=101)——→

Connection Established ✓
```

- **SYN:** Client asks "Are you ready to communicate?"
- **SYN-ACK:** Server replies "Yes, I'm ready, are you?"
- **ACK:** Client confirms "Yes, let's communicate!"



Step 3: Data Exchange Through OSI Layers

Now server and client can send data to each other:

When CLIENT sends request to SERVER:

```
Application Layer: HTTP GET request
Transport Layer: TCP header added
Network Layer: IP header (client IP → server IP)
Data Link Layer: Ethernet frame
Physical Layer: Electrical signals
```

When SERVER sends response back to CLIENT:

```
Server needs to send data back to client IP
Again DNS resolution might occur (reverse DNS)
Same OSI layers process in reverse
Data reaches client browser
```

Step 4: Complete Flow Summary

```
1. Type google.com → DNS Resolution → Get Server IP
2. Three-way handshake → Connection established
3. HTTP request sent through OSI layers
4. Server processes request
5. Server sends response back (using client IP)
6. Data travels through OSI layers
7. Browser receives and displays webpage
```

Key Interview Points:

- **DNS is always first** - convert domain to IP
- **Three-way handshake** - establish reliable connection
- **OSI layers** - how data is packaged and sent
- **Bidirectional communication** - both client and server exchange data
- **IP addressing** - every device needs unique identifier

You've explained it perfectly - DNS first to get IP, then handshake to establish connection, then data exchange through OSI layers, with DNS potentially happening again for reverse lookups or additional resources!