Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network.

Ping is an **application utility (user program).**

It uses **ICMP**, a **Layer 3 protocol.**
Therefore, **ping is not an L7 protocol**, but a **user-level tool that exercises L3 behavior.**

You (user) run **ping** — that's a *software application*.

- Ping talks directly to the **ICMP protocol** — which lives at *Layer 3*.
- ICMP then rides over **IP** and below that **Ethernet.**

So, **ping = user-space tool, ICMP = Layer 3 protocol.**
So the **ping program runs at user level**, but the **packets it generates operate at Layer 3.**
When we say **ping is an application**, we mean it's a **user-space program** (a software utility you can run from the shell).

That doesn't mean it operates at **OSI Layer 7 (Application layer).**

q). How ping is special ?

Ping is different because it doesn't use **TCP or UDP.**
It uses **ICMP (Internet Control Message Protocol)**, which itself is a **Layer 3 protocol**, just like IP.

So the flow looks like:

```
User (ping app)
   ↓
Kernel (creates ICMP Echo Request packet)
   ↓
IP Layer (encapsulates ICMP into IP packet)
   ↓
Ethernet Layer (adds MAC header)
   ↓
Network
```

q) How ping does this "jump"
ans)
When you run:
`ping 8.8.8.8`

The following happens internally:

Ping (user process) calls the OS kernel using raw sockets:
```
socket(AF_INET, SOCK_RAW, IPPROTO_ICMP);
```

- AF_INET → IPv4 family
- SOCK_RAW → raw socket type
- IPPROTO_ICMP → specifies ICMP protocol

2. This bypasses TCP and UDP layers entirely.
   Instead, ping directly constructs the ICMP Echo Request packet and gives it to the kernel.

3. The kernel IP stack then:
   - Adds the IP header (source/destination IP, TTL, etc.)
   - Sends the full IP packet down to the network interface driver (Layer 2)

4. At the receiving side, the kernel recognizes the ICMP Echo Request, generates an ICMP Echo Reply, and sends it back the same way.