

Understanding TCP

A Wireshark trace to radio.garden

November 2025

Luke Poley

What Stages does TCP go through on radio.garden



Think of it like a phone conversation:

- Picking up the phone → Typing the URL
- Dialing → Requesting initial page (SYN)
- Waiting for answer → Sending initial page (SYN-ACK)
- Talking → Page interaction (ESTABLISHED)
- Saying goodbye → Closing page (FIN)

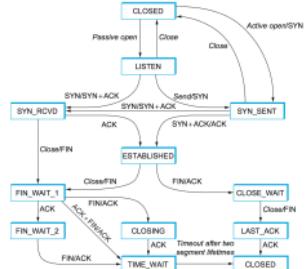


Figure 1: TCP state-transition diagram

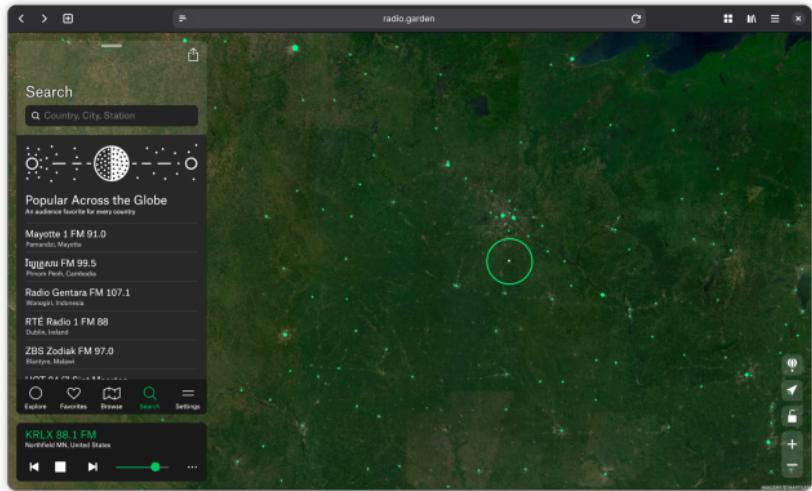


Figure 2: Radio.garden uses a three-tier distribution of: Cloudflare (layout), BunnyCDN (map), Voscast (radio)

1 How does a connection begin? The Three-Way Handshake

After entering the URL, your computer (the client) and the server need to synchronize

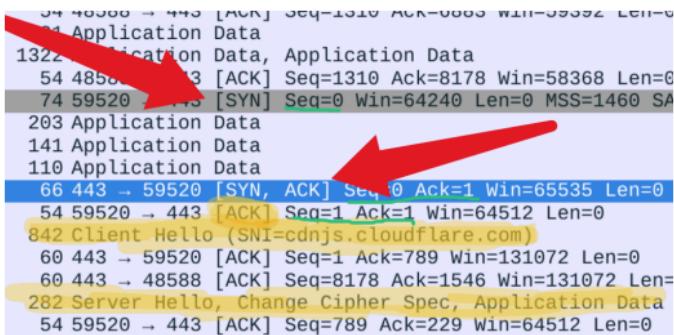


Figure 3: Packets 42-51

Client Initialization: CLOSED → SYN-SENT:

- Client says “I need to connect to this server”
- TCP creates a SYN (synchronize) packet
- Includes initial sequence number (random for security)
- Sends to server and waits for response



The Three-Way Handshake

Server: LISTEN → SYN-RECEIVED

- Was in LISTEN state (waiting for connection requests)
- Receives SYN packet from client
- Allocates resources for this connection and sequence number
- Sends back SYN-ACK (synchronize + acknowledge)

Client: Sends ACK

- Sends ACK to Server with to verify server's sequence number

Connection will encrypt with TLS

The Handshake will repeat for each server: Cloudflare (Layout), BunnyCDN (Map), Voscast (Radio)

2 The ESTABLISHED State

What Happens in ESTABLISHED State?



This is where TCP spends most of its time - transferring data!

In ESTABLISHED state, both sides can:

- Send data in either direction (full-duplex)
 - ACK received data
 - Apply sliding-window flow control, where the advertised receive window limits how much outstanding data the sender may have in flight

What you'll see in Wireshark:

- Many packets flowing back and forth
 - All have ACK flag set
 - Sequence numbers incrementing
 - Window sizes adjusting based on the task being changed, such as $Seq = 1223$ changing the amount of packets sent at an instant

Figure 4: Packets shortly after the audio streaming started, while the map data is still in transit.
 (Dynamic Window Sizes) Packets 622 - 649

Reliable Transport



Reliable Transport prevents Packet Loss

- Many packets have the same ACK number since it gets lost in transport
- The sender, not knowing yet, keeps sending the later cards (1539, ...)
- The receiver keeps saying it needs packet with ACK=1538

Wireshark Blacklines

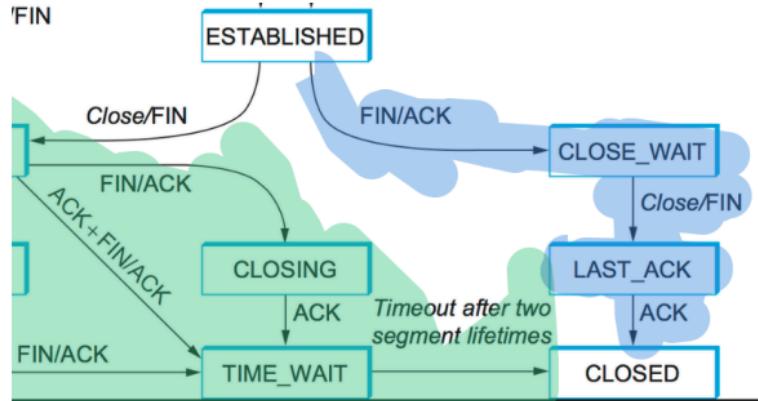
- TCP Out-of-Order: Packet arrived before the missing one
- TCP Dup ACK: Receiver is saying “Same ACK again because we are still missing data.”

```
54 36022 → 10803 [ACK] Seq=1223 Ack=153877 Win=128000 Len=0
107 Application Data
773 Application Data
54 44310 → 443 [ACK] Seq=1538 Ack=54156 Win=35840 Len=0
436 443 → 44310 [ACK] Seq=54156 Ack=1538 Win=71680 Len=1382 [TCP PDU reassembled in
54 44310 → 443 [ACK] Seq=1538 Ack=55538 Win=34816 Len=0
436 443 → 44310 [PSH, ACK] Seq=55538 Ack=1538 Win=71680 Len=1382 [TCP PDU reassembl
818 443 → 44310 [PSH, ACK] Seq=56920 Ack=1538 Win=71680 Len=2764 [TCP PDU reassembl
436 443 → 44310 [ACK] Seq=59684 Ack=1538 Win=71680 Len=1382 [TCP PDU reassembled in
54 44310 → 443 [ACK] Seq=1538 Ack=56920 Win=34816 Len=0
54 44310 → 443 [ACK] Seq=1538 Ack=59684 Win=32768 Len=0
54 44310 → 443 [ACK] Seq=1538 Ack=61066 Win=31744 Len=0
436 443 → 44310 [PSH, ACK] Seq=61066 Ack=1538 Win=71680 Len=1382 [TCP PDU reasemb
54 44310 → 443 [ACK] Seq=1538 Ack=62448 Win=34914 Len=0
436 [TCP Previous segment not captured] 443 → 44310 [PSH, ACK] Seq=63029 Ack=1538 Win=71680 Len=1382 [TCP
436 [TCP Out-Of-Order] 443 → 44310 [ACK] Seq=62448 Ack=1538 Win=71680 Len=1382 [TCP
66 [TCP Dup ACK 820#1] 44310 → 443 [ACK] Seq=1538 Ack=62448 Win=34816 Len=0 SLE=63
54 44310 → 443 [ACK] Seq=1538 Ack=65212 Win=33792 Len=0
818 443 → 44310 [PSH, ACK] Seq=65212 Ack=1538 Win=71680 Len=2764 [TCP PDU reasemb
436 443 → 44310 [ACK] Seq=67976 Ack=1538 Win=71680 Len=1382 [TCP PDU reassembled in
```

Figure 5: Packets 807-825

3 Connection Termination

Termination



Blue: The Normal 4-Way Handshake termination from the layout and map servers.

Green: Server sends RST flag to terminate the audio

1043 198.178.123.8	12.826634729	10.133.23.207	TCP	54 36022 -- 10803 [ACK] Seq=1223 Ack=202640 Win=130048 Len=0
1046 198.178.123.8	12.906694853	10.133.23.207	TLSv1.2	85 Encrypted Alert
1047 198.178.123.8	12.906752336	10.133.23.207	TCP	54 36022 -- 10803 [FIN, ACK] Seq=1254 Ack=202640 Win=130048 Len=0
1048 10.133.23.207	12.925477437	198.178.123.8	TCP	101 [TCP Previous segment not captured] 10803 -- 36022 [PSH, ACK] Seq=204022 Ack=1223 Win=17920 Len=47
1049 198.178.123.8	12.925546816	10.133.23.207	TCP	54 36022 -- 10803 [RST] Seq=1223 Win=0 Len=0
1050 10.133.23.207	12.925752438	198.178.123.8	TCP	1436 [TCP Out-of-Order] 10803 -- 36022 [ACK] Seq=202640 Ack=1223 Win=17920 Len=1382
1051 198.178.123.8	12.925762988	10.133.23.207	TCP	54 36022 -- 10803 [RST] Seq=1223 Win=0 Len=0
1052 10.133.23.207	12.945030002	198.178.123.8	TCP	60 10803 -- 36022 [ACK] Seq=20469 Ack=1254 Win=17920 Len=0
1053 198.178.123.8	12.945086626	10.133.23.207	TCP	54 36022 -- 10803 [RST] Seq=1254 Win=0 Len=0
1054 10.133.23.207	12.974677170	198.178.123.8	TCP	1436 10803 -- 36022 [ACK] Seq=20469 Ack=1255 Win=17920 Len=1382 [TCP PDU reassembled in 1056]
1055 198.178.123.8	12.974739376	10.133.23.207	TCP	54 36022 -- 10803 [RST] Seq=1255 Win=0 Len=0
1056 10.133.23.207	12.976148754	198.178.123.8	TLSv1.2	101 Application Data
1057 198.178.123.8	12.976177918	10.133.23.207	TCP	54 36022 -- 10803 [RST] Seq=1255 Win=0 Len=0
1058 10.133.23.207	13.077295964	198.178.123.8	TCP	60 10803 -- 36022 [RST, ACK] Seq=205408 Ack=1255 Win=17920 Len=0
1060 38.142.94.218	13.329642961	10.133.23.207	TLSv1.3	93 Application Data

Figure 7: The (Green) Reset Termination as it appears in Wireshark