Multipath TCP: Breaking the Single-Path Limit

Why settle for one path when you can use all of them?

SECTION 1

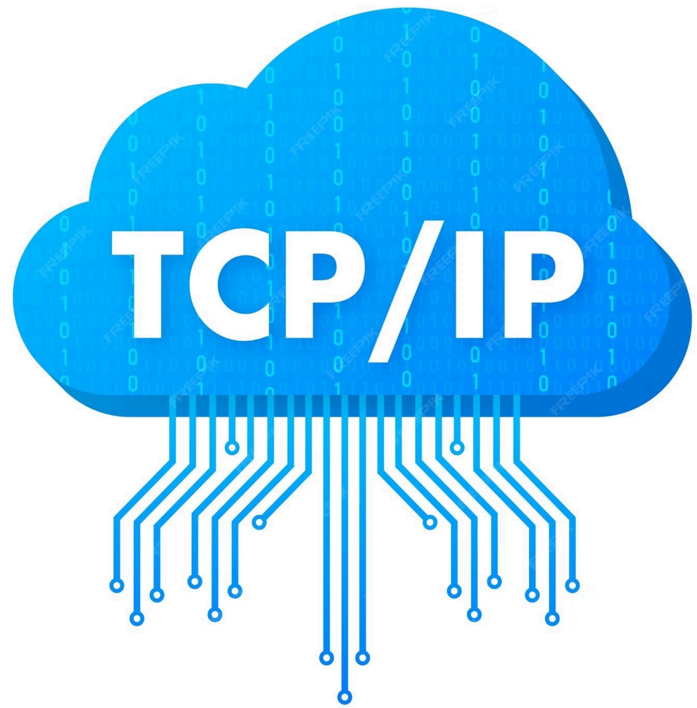
The Problem with Standard TCP

In a world of constant connectivity, standard TCP's limitations create bottlenecks and unreliable experiences, especially for mobile users.

The "One Connection, One Path" Limitation

Standard TCP connections are **rigidly defined** by a 5-tuple: Source IP, Source Port, Dest IP, Dest Port, and Protocol.

- If *any* of these parameters change (e.g., your IP address changes as you move), **the connection breaks**.
- Imagine a single-lane road: if that road is blocked, all traffic stops. Standard TCP operates on this principle, building a **single "pipe"** between two devices.



Why This Matters for Modern Devices

Wasted Bandwidth

Modern devices often have multiple network interfaces simultaneously active (4G/5G and WiFi). Standard TCP can only utilize one at a time, leaving valuable bandwidth unused.

Poor Reliability

When the active link (e.g., a weak WiFi signal) degrades, the connection either hangs or drops entirely, instead of seamlessly switching to a stronger available network.

Handover Latency

Switching between networks (e.g., from WiFi to cellular) typically requires terminating the existing TCP connection and establishing a brand new one, causing noticeable delays and service interruptions.

SECTION 2

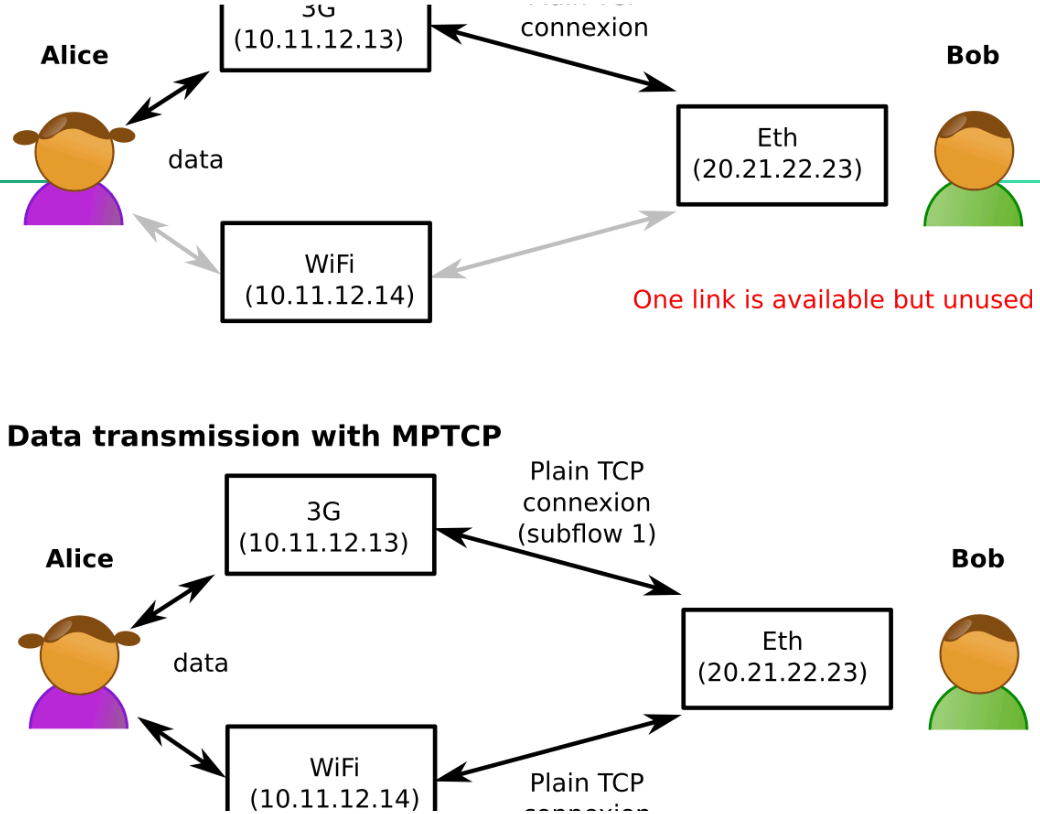
What is Multipath TCP?

Multipath TCP (MPTCP) is an extension to TCP that allows a single connection to use multiple paths simultaneously, enhancing performance and reliability.

Where MPTCP Fits in the OSI Model

Standard TCP

Application → TCP → IP (single path)

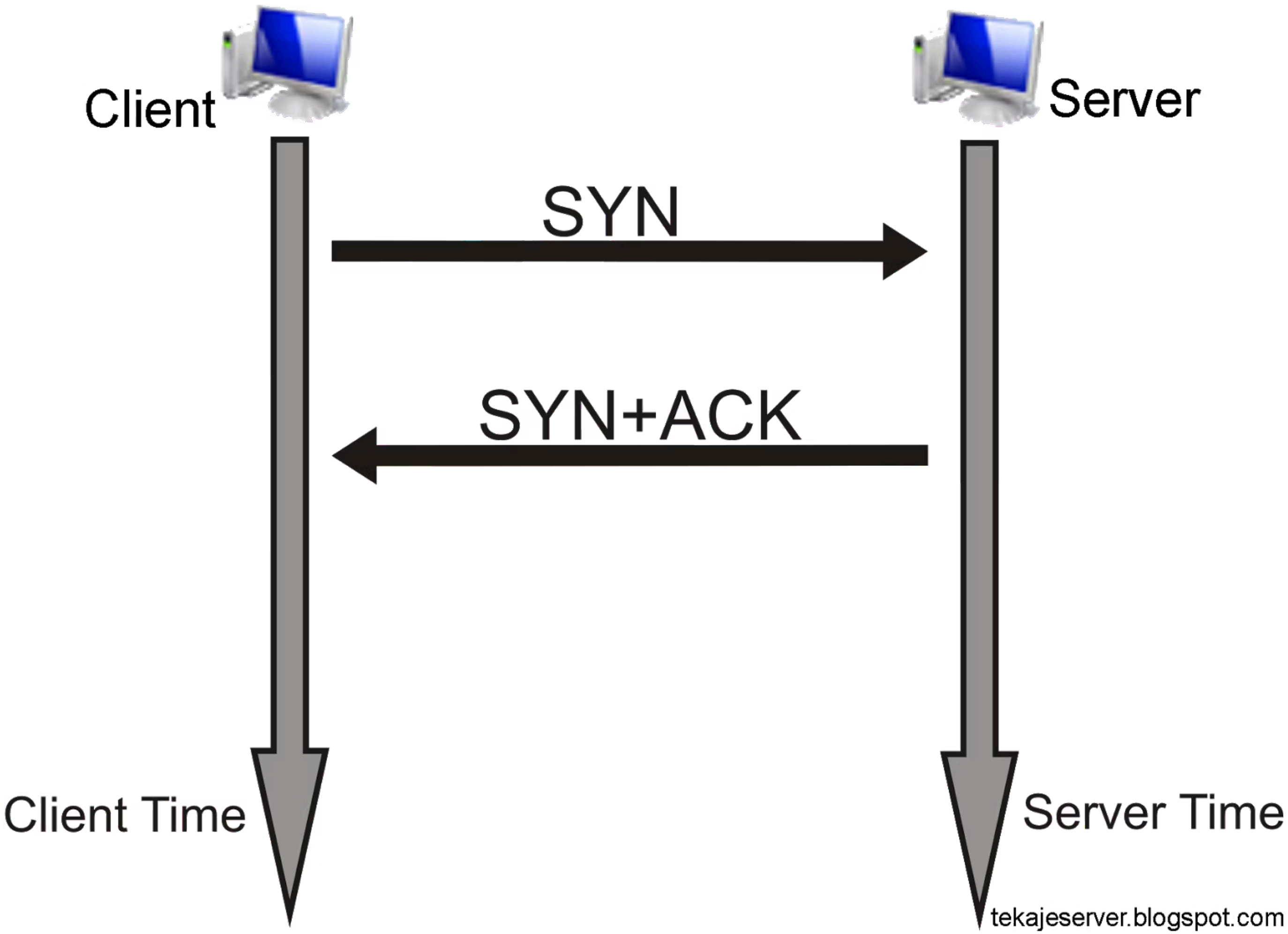


MPTCP Stack

Application → MPTCP → TCP subflows → IP (multiple paths)

MPTCP introduces an abstraction layer above standard TCP, managing multiple "subflows" to provide a unified, resilient connection to applications.

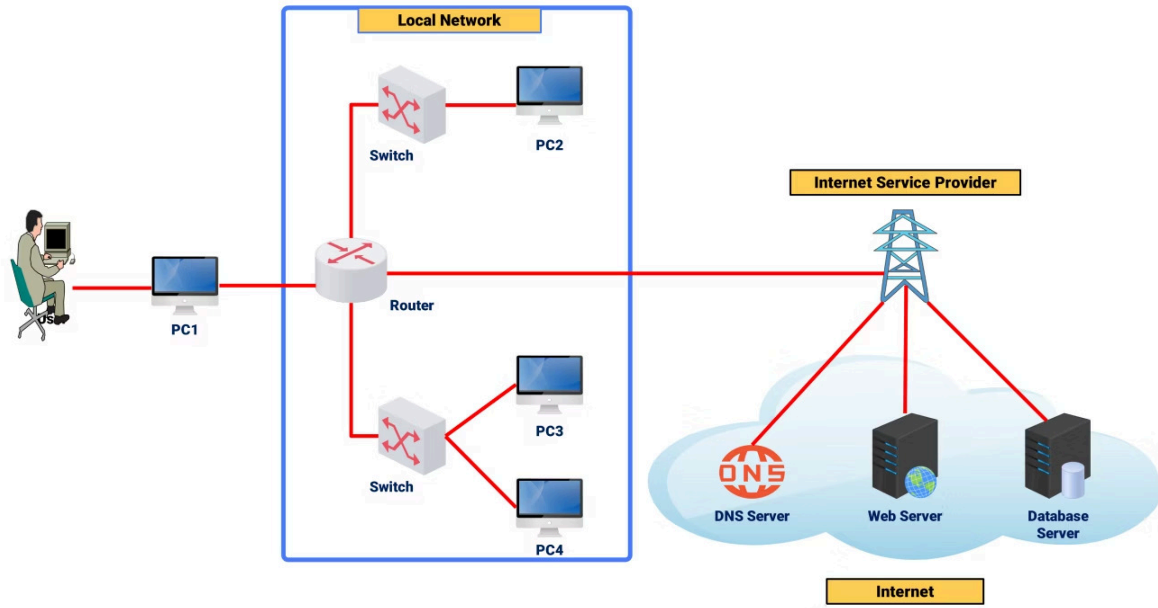
Establishing the Connection: The MP_CAPABLE Handshake



- **Step 1:** The client initiates a connection by sending a `SYN` packet with the `MP_CAPABLE` option.
- **Step 2:** If the server supports MPTCP, it responds with a `SYN/ACK` packet also containing the `MP_CAPABLE` option.
- **Result:** A standard TCP connection is established, but both endpoints are now aware that additional paths can be added later to enhance connectivity.
- **Fallback:** If the server does not support MPTCP, it simply ignores the `MP_CAPABLE` option, and the connection proceeds as a standard single-path TCP session.

Seamlessly Adding New Paths: MP_JOIN

- **Scenario:** A device detects a new network interface becoming available (e.g., an LTE connection activates while WiFi is in use).
- **Action:** The MPTCP client initiates a new standard TCP handshake on this newly available interface.
- **The Token:** During this handshake, the client includes the `MP_JOIN` option, carrying a unique token that identifies it as part of an *existing* MPTCP session.
- **Result:** The server associates this new connection with the ongoing MPTCP session, allowing data to flow over both paths concurrently, leveraging the aggregated bandwidth and improving resilience.



SECTION 3

Project Implementation: Application-Layer Emulation

To demonstrate MPTCP's failover capabilities, we implemented a Python-based, user-space simulation with a primary (WiFi) and backup (LTE) path for streaming data and simulating link failures.

The "Scheduler" Algorithm: Failover Logic

Primary Attempt

The system defaults to Path A for initial data transmission, simulating optimal conditions (e.g., WiFi).

Failure Detection

If a connection drop or timeout is detected (simulated after a specific number of packets, e.g., Packet #10), an exception is raised by the network layer.

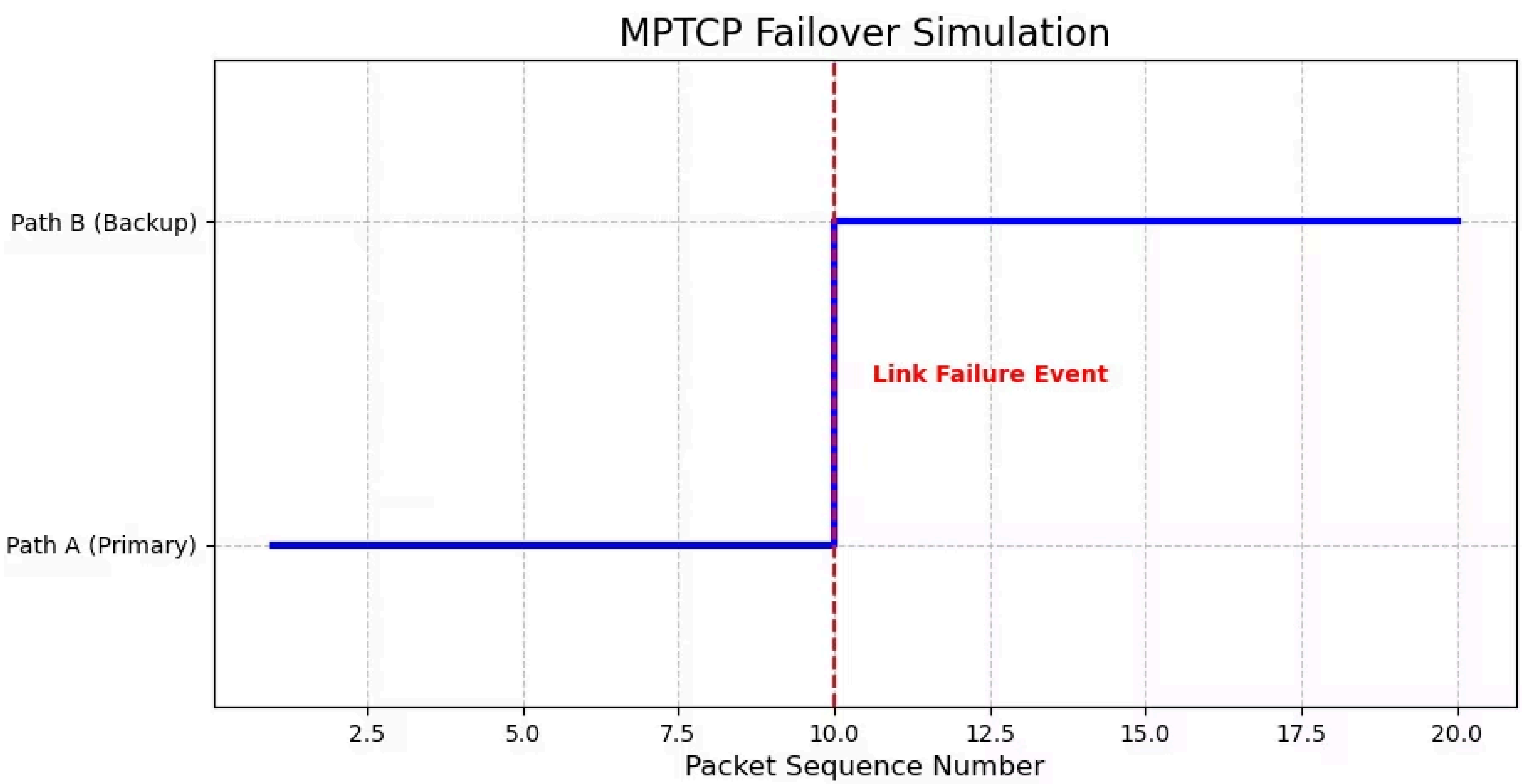
Instant Switch

Our logic immediately catches the exception and reroutes the subsequent data packets to the Backup Path (Path B).

Zero Data Loss

The application continues to stream data without crashing or experiencing noticeable service interruption, demonstrating seamless failover.

Demonstration Results: Seamless Recovery



01

Active Path Indication

The blue line on the graph indicates the currently active data path, with Level 1 representing the Primary (WiFi) and Level 2 representing the Backup (LTE).

02

Simulated Failure Event

Precisely at **Packet 10**, marked by a vertical red line, we simulated a failure of the Primary path.

03

Immediate Path Switching

The graph demonstrates an **immediate and sharp jump** to Level 2 (the backup path) right after the primary path's failure, showing no interruption in data flow.

04

The MPTCP Advantage

In a standard TCP connection, this graph would abruptly end at the red line. MPTCP's intelligent scheduler allows the connection to **seamlessly continue** over the secondary path, ensuring continuous service.

SECTION 4

Who Uses MPTCP Today?

Apple iOS Integration

Siri, Apple Music, and Maps leverage MPTCP to maintain responsiveness even when **WiFi signals are weak**, ensuring commands and streams go through without interruption.

5G Networks (ATSSS)

Access Traffic Steering, Switching, and Splitting (ATSSS) within 5G architectures uses MPTCP to **bond WiFi and cellular connections** for enhanced performance and reliability.

Hybrid Access Broadband

Internet Service Providers (ISPs) deploy MPTCP to combine fixed lines like DSL with **4G/5G mobile connections**, delivering boosted home internet speeds and greater resilience.

CONCLUSION

Summary

Breaks Single-Path Constraints

MPTCP overcomes the inherent limitations of traditional TCP by leveraging multiple network paths simultaneously.

Delivers Core Advantages

Enjoy significantly higher throughput, seamless device mobility, and superior connection reliability.

Validates Failover Logic

Our simulation successfully showcased MPTCP's "**Seamless Failover**" mechanism for uninterrupted data flow.