TCP/IPの全体像と「4つの階層」

なぜネットワークはレイヤーで分かれているのか？

ネットワーク基礎 #01

本日のゴール

- **TCP/IPの4層モデル**の役割を説明できる
- 通信データが包まれる**カプセル化**のイメージがつかめる
- 通信障害が起きたとき、**階層ごとに切り分ける視点**が身につく

1. TCP/IP 4層モデル（全体像）

4. アプリケーション層 (Webブラウザ・メール)

↓ データを渡す

3. トランスポート層 (信頼性・速度管理)

↓ 届ける

2. インターネット層 (IPでのルーティング)

↓ 物理的に送る

1. リンク層 (有線LAN・Wi-Fi)

2. 各層の役割とプロトコル

階層	主なプロトコル	役割のイメージ
4. アプリケーション層	HTTP, HTTPS, DNS	ユーザーやアプリが直接触れるデータ
3. トランスポート層	TCP, UDP	データ漏れがないか・どのアプリ宛か（ポート番号）
2. インターネット層	IP, ICMP	どのサーバーへ届けるか（IPアドレス）
1. リンク層	Ethernet, Wi-Fi	電気信号や光信号で物理的に届ける（MACアドレス）

3. なぜ「4つの階層」に分けるのか？

巨大な仕組みを単一で作らないための「役割分担」

- 各層は**隣の層との決まった接点**だけでやり取りする
- ある層の変更は**その層の中にとどまりやすい**（影響の局所化）
  - 回線を変えても: LANケーブル → Wi-Fi でもブラウザのコードは変更不要
  - アプリを変えても: Webアプリを作り直しても物理回線の仕組みは変更不要

4. データの旅を支える「カプセル化」

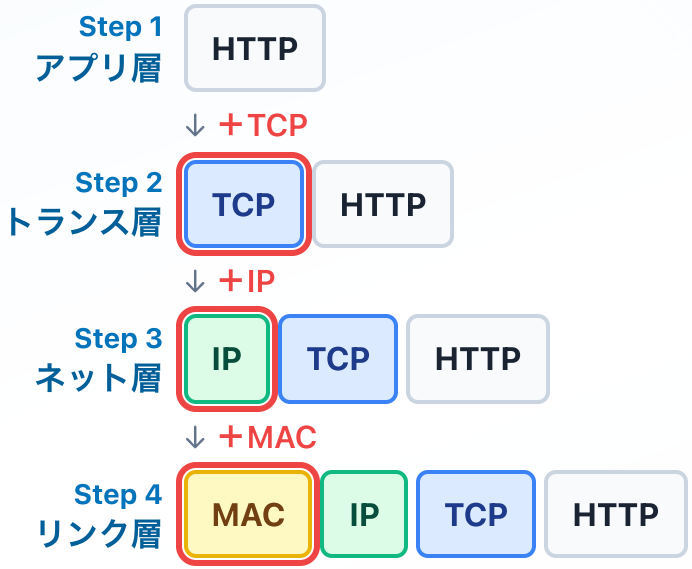
送信時は上から下に「包み紙（ヘッダー）」を追加し、受信時は逆順に剥がす。

- **DATA** → **TCP**（Segment）→ **IP**（Packet）→ **MAC**（Frame）
- 各層がヘッダーを付与して、下の層へ渡す
- 「包み紙 = ヘッダー」と覚える

5. カプセル化の流れ

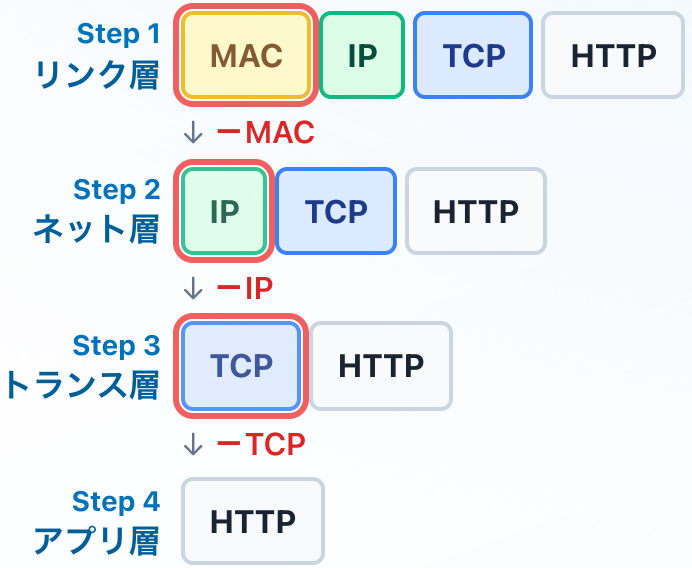
送信時（ヘッダーを足す）

上の層 → 下の層へ



受信時（ヘッダーを剥がす）

下の層 → 上の層へ



▢赤枠 = その層で付与／剥がすヘッダー

6. 階層ごとの障害切り分け

「Webサイトが開かない！」とき、どの階層から疑うか？

【4】アプリケーション層	HTTPエラーは返っていないか？ (500等)
【3】トランスポート層	ポートは開いているか？ (nc / telnet)
【2】インターネット層	IPで相手に届いているか？ (ping)
【1】リンク層	物理的に接続されているか？ (ケーブル等)

⬆ 下（物理）から順に確認していくのが基本！ ⬆

まとめ

1. TCP/IPは通信をシンプルに保つための**役割分担の知恵**
2. データは各層のヘッダーで**カプセル化**されて届く
3. 通信トラブル時は**どの階層の問題か**を意識して切り分けよう

次回予告

ネットワーク基礎 #02

IPアドレスとネットワークの範囲

IPv4 / サブネット / CIDR プライベートIPとNAT