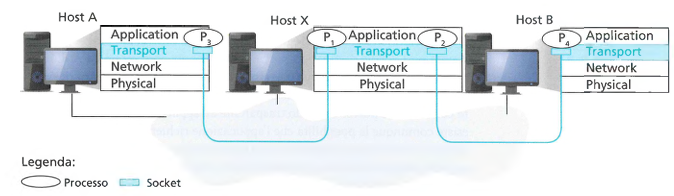
**Multiplexing e Demultiplexing nel livello Transport.**

**Definizione delle funzionalità:**

* **Multiplexing**: in trasmissione, il livello Transport riceve dati da più processi (tramite socket) e li incapsula con un header, unendo tutto in un flusso unico verso la rete.
* **Demultiplexing**: in ricezione, il livello Transport legge l’header dei segmenti ricevuti e li smista al processo corretto, identificato dalla porta di destinazione.

**Analogia postale:**

* L’indirizzo IP dell’host è come l’indirizzo di casa.
* La porta (socket) è come il nome del destinatario sulla busta.
* **Demultiplexing**: smistare le lettere ai membri della famiglia.
* **Multiplexing**: raccogliere lettere da vari membri e spedirle insieme.



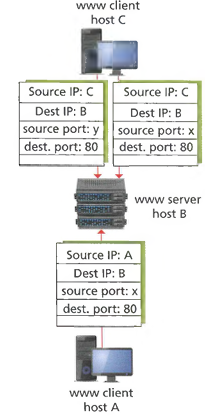
**Tipi di multiplexing/demultiplexing:**

1. **Connectionless (senza connessione) – UDP**

* Nessuna connessione stabile.
* La socket è identificata da: <Server IP address, Server Port number>
* Più client possono inviare dati allo stesso server sulla stessa porta.

1. **Connection-oriented – TCP**

* Connessione stabile tra client e server.
* La socket è identificata da una **quadrupla**: <Client IP, Server IP, Client Port, Server Port>
* Permette più connessioni simultanee allo stesso servizio, anche dallo stesso client.

**Esempio pratico: Web server (porta 80)**

* Host A e Host C (client) si connettono al server B (web server).
* Entrambi inviano segmenti con **destination port = 80** (HTTP).
* Ogni client usa una **source port** diversa (es. x), scelta automaticamente o tramite bind ().

**Scambio dei segmenti:**

* Da A a B: source port = x, destination port = 80
* Da B a A (risposta): source port = 80, destination port = x

Il server distingue le connessioni grazie alla **combinazione unica di IP e porte**, evitando conflitti anche se due client usano la stessa porta sorgente.