

CS2003: Internet and The Web

Basic Internet Protocol Architecture

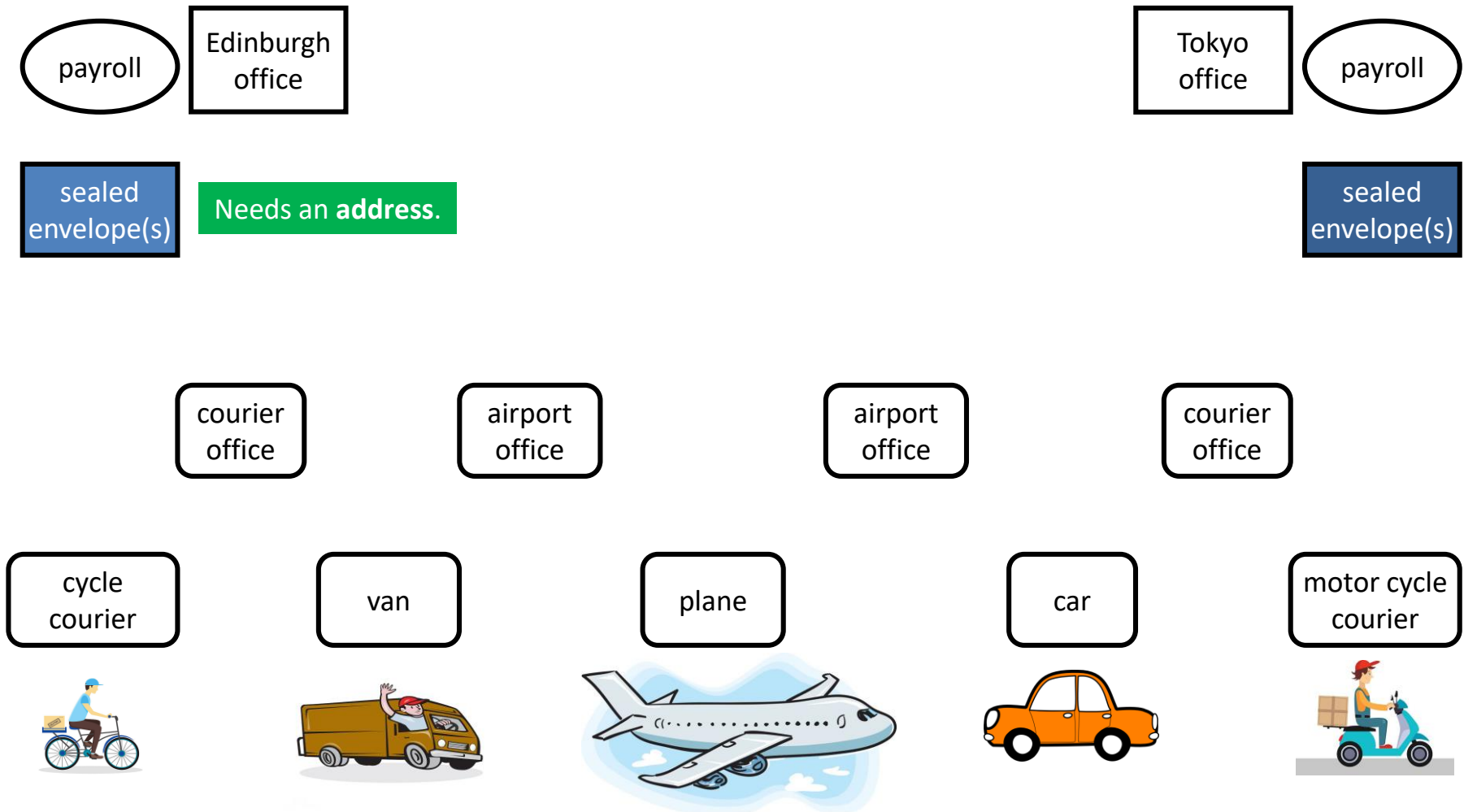
How do we arrange communication?

- Why do we need different network devices?
- Does every application need to worry about how to transmit on fibre / copper / radio?
 - How do we deal with different physical media?
- Are there common services that could be **abstracted** and **provisioned** for the benefit of multiple applications?

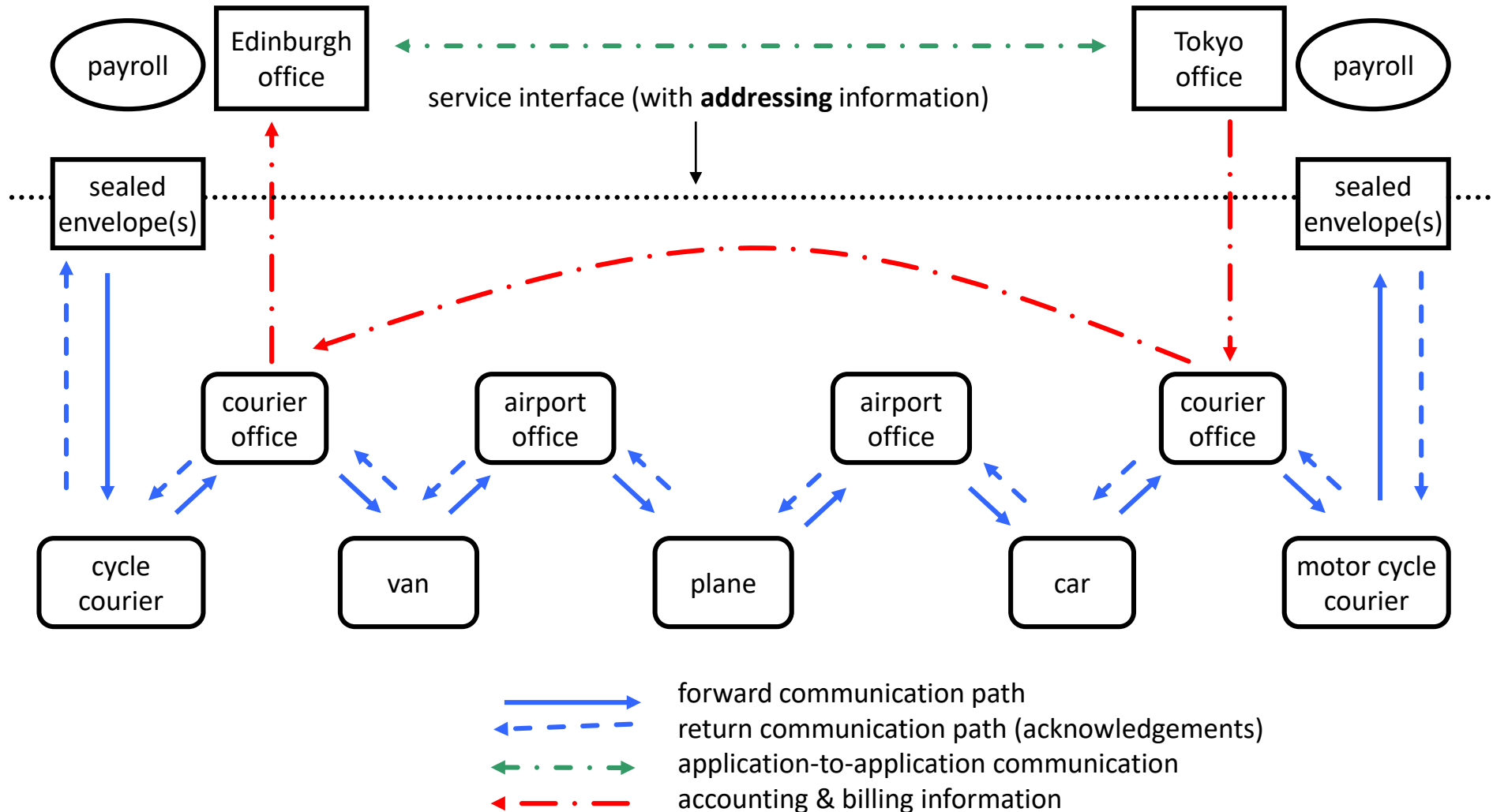
Application example: Payroll (1)

- Payroll:
 - Fluff International Ltd (suppliers of quality fluff).
 - Sites in Edinburgh, Tokyo and New York.
 - Payroll managed centrally (Edinburgh) and distributed to other offices.
- Compare process of:
 - “Traditional” paper-based approach.
 - “Map” to an electronic approach.

Application example: Payroll (2)



Application example: Payroll (3)



Layered architecture

- Application-layer information:
 - format of information, application-level messages.
- End-to-end service interface to communication:
 - well-defined, end-to-end, transparent communication.
- Network service:
 - “common” courier service.
- Link service:
 - each mode of transmission has its own envelope.
- Physical transmission on individual links.

5

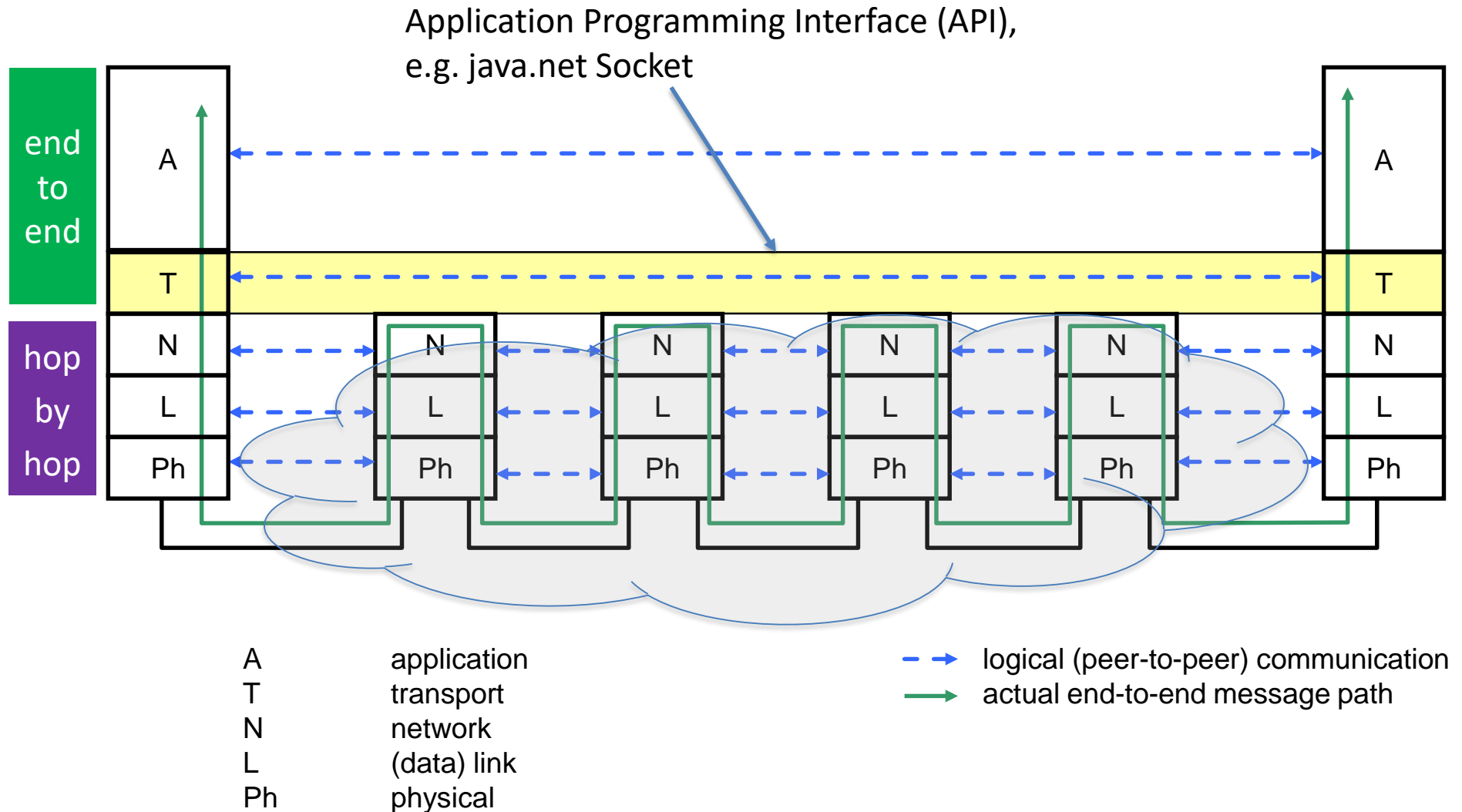
4

3

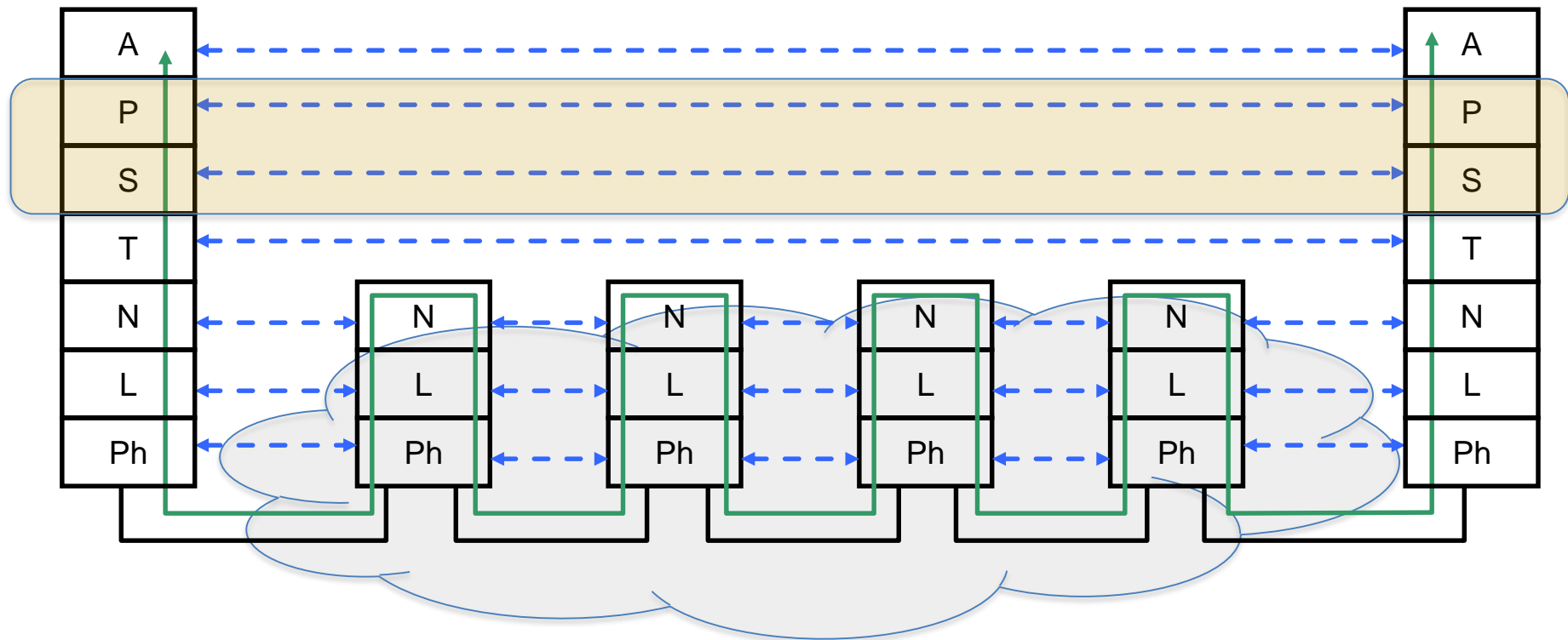
2

1

Internet protocol reference model



OSI protocol reference model

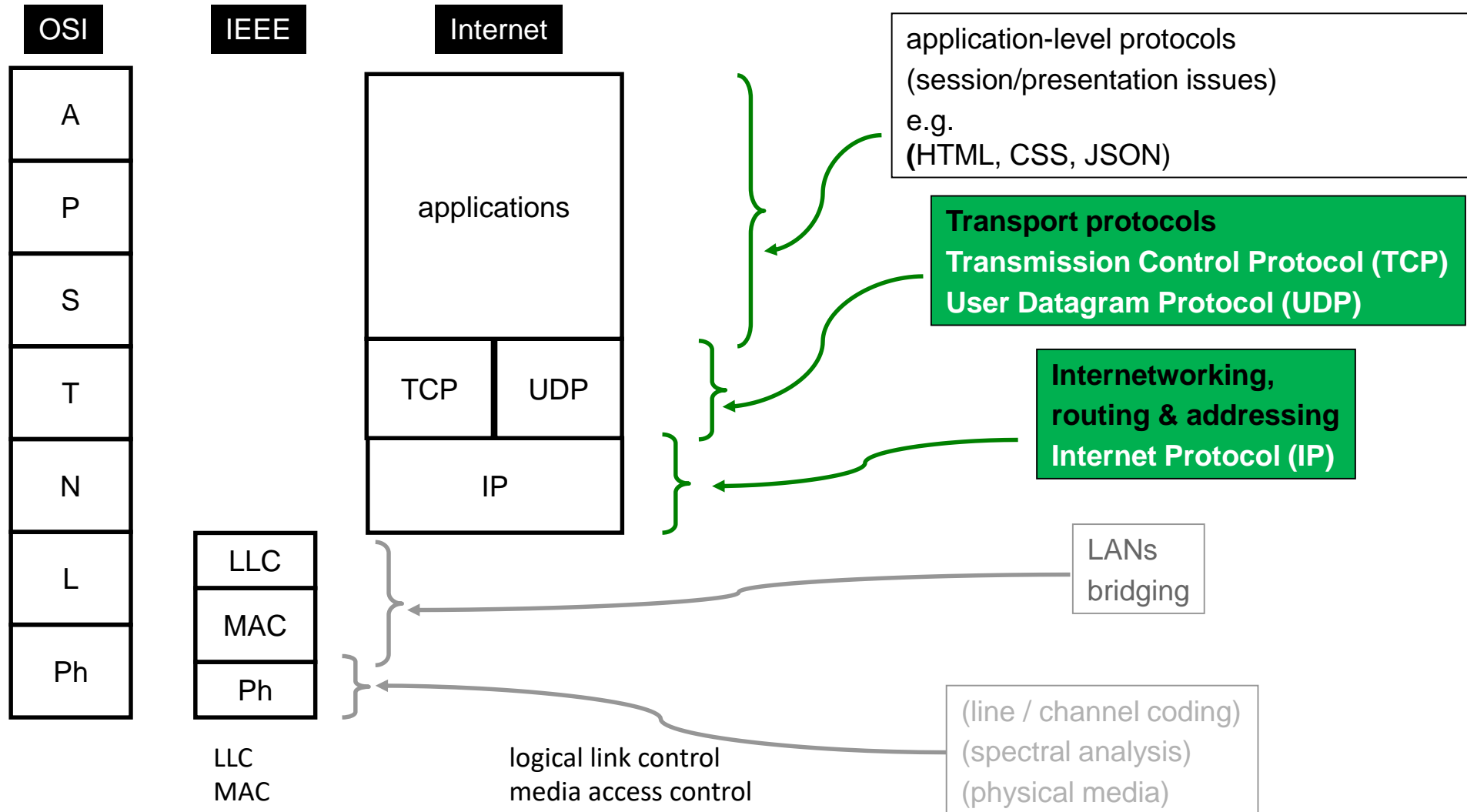


A
P
S
T
N
L
Ph

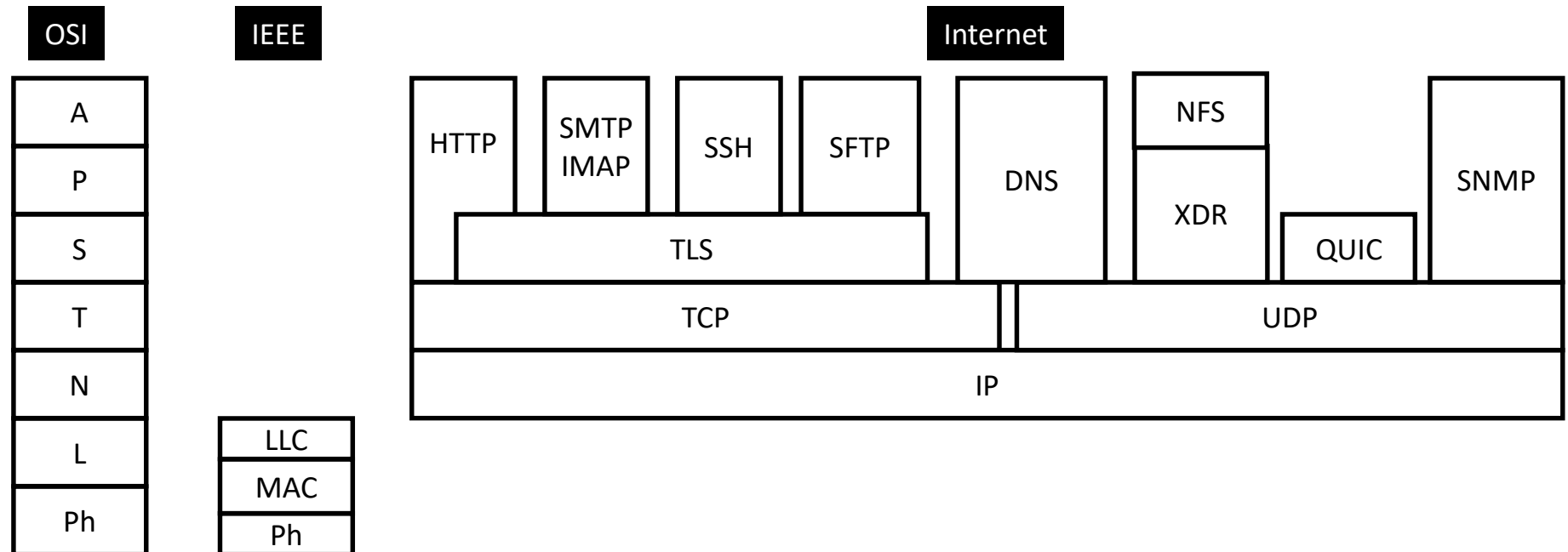
application
presentation
session
transport
network
(data) link
physical

—→ logical (peer-to-peer) communication
—→ actual end-to-end message path

Protocol Reference Models

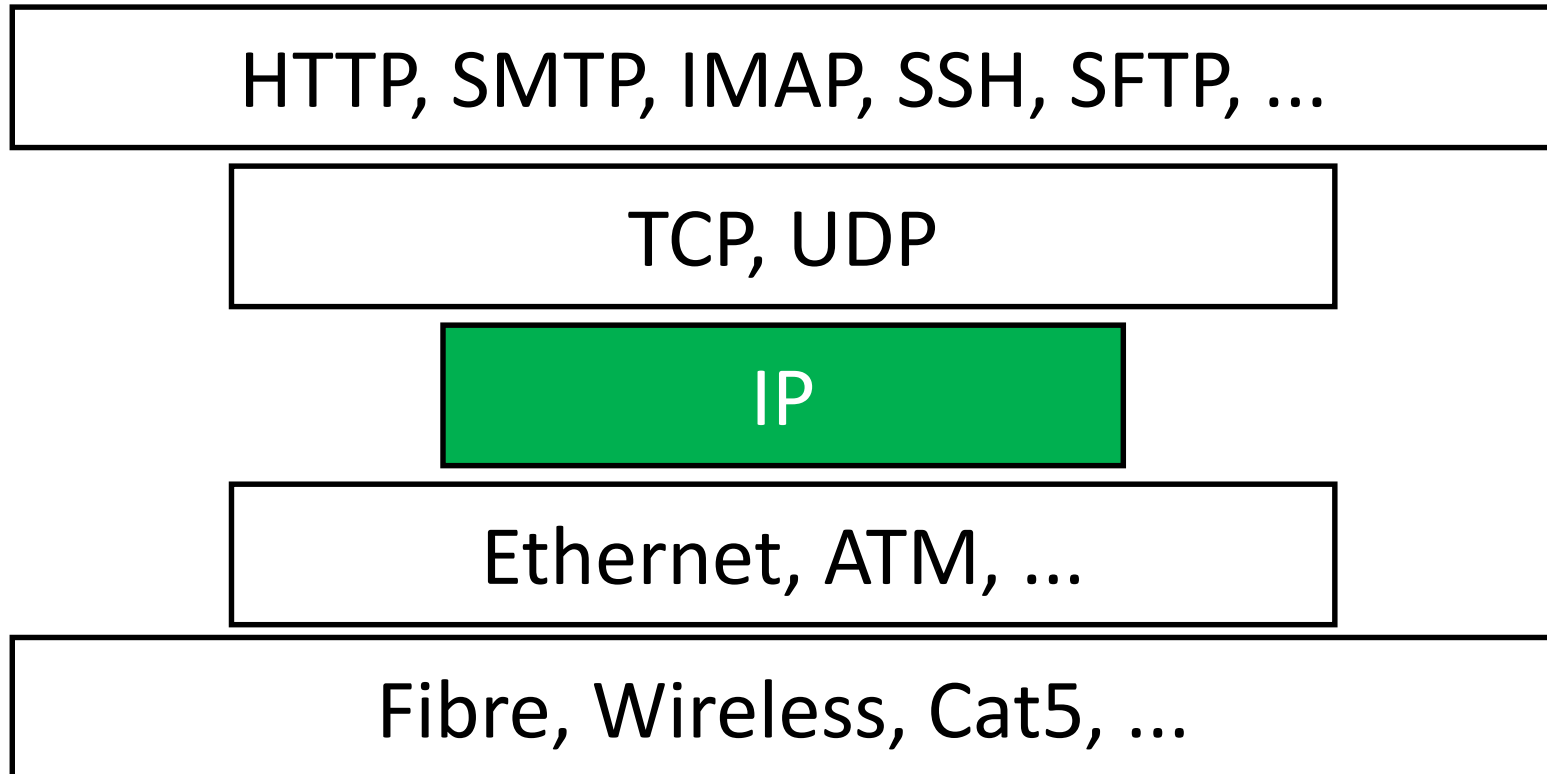


OSI, IEEE & Internet protocol examples



| | | | |
|------|--|------|------------------------------------|
| DNS | Domain Name Service | POP | Post Office Protocol |
| HTTP | HyperText Transfer Protocol | QUIC | Quick UDP Internet Connections |
| SFTP | Secure File Transfer Protocol | SMTP | Simple Mail Transfer Protocol |
| IP | Internet Protocol | SNMP | Simple Network Management Protocol |
| IEEE | Institute of Electrical and Electronic Engineers | SSH | Secure Shell |
| LLC | Logical Link Control | TCP | Transmission Control Protocol |
| MAC | Medium Access Control | TLS | Transport Layer Security (aka SSL) |
| NFS | Network File System | UDP | User Datagram Protocol |
| OSI | Open Systems Interconnection | XDR | eXternal Data Representation |

IP as the “waist” of the hourglass

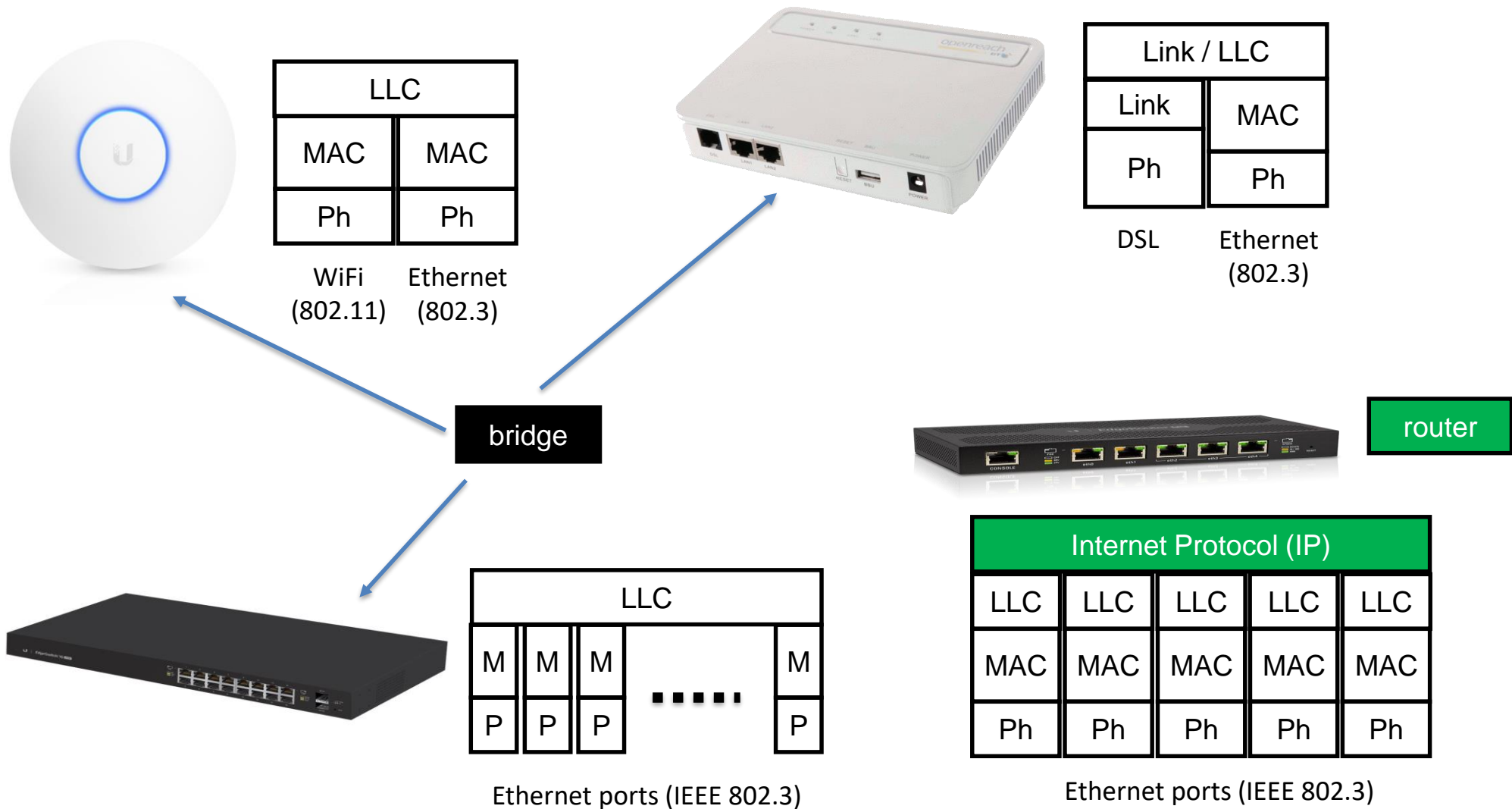


IP is independent of sub-network technology.

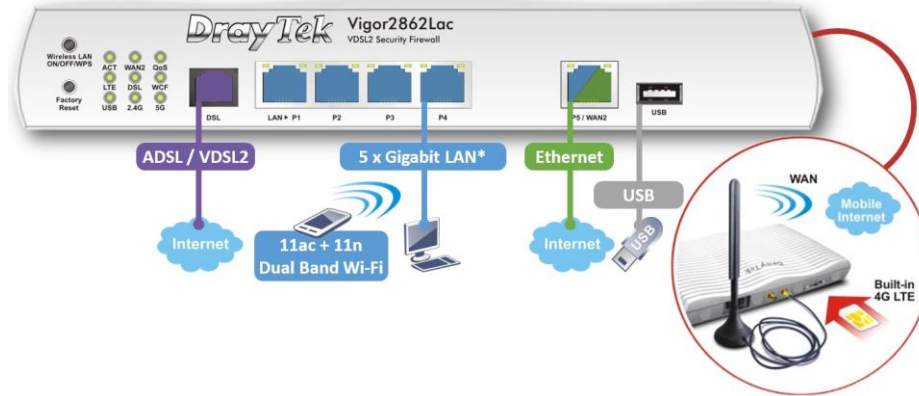
IP is the common layer for internetworking

Need to “map” IP onto sub-network technology.

Equipment and layers: internetworking



Multi-network router



Internetworking.

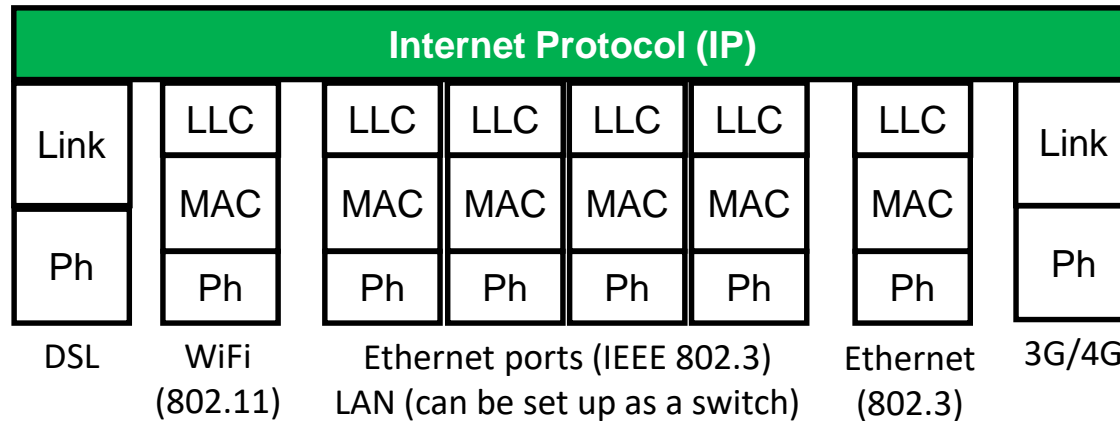
Example configuration shown.

Other configurations possible, e.g.:

- WiFi bridged to Ethernet.
- ISP via DSL.
- ISP via 3G/4G.
- ISP via additional ethernet port.
- (Software configurable.)

Home gateway can also run various other functions and applications, e.g.:

- DHCP
- DNS
- NAT
- Firewall
- Web server



protocol (1)

protocol noun

pro·to·col | \ 'prō-tə-, kōl - , kōl, - , käl, -kəl\

Definition of *protocol*

- 1** : an original draft, minute, or record of a document or transaction
 - 2 a** : a preliminary memorandum often formulated and signed by diplomatic negotiators as a basis for a final convention or treaty
 - b** : the records or minutes of a diplomatic conference or congress that show officially the agreements arrived at by the negotiators
 - 3 a** : a code prescribing strict adherence to correct etiquette and precedence (as in diplomatic exchange and in the military services) a breach of protocol
 - b** : a set of conventions governing the treatment and especially the formatting of data in an electronic communications system network protocols
 - c** : convention sense 3a,b
 - 4** : a detailed plan of a scientific or medical experiment, treatment, or procedure
- source - <http://www.m-w.com/dictionary/protocol>

protocol (2)

- Definitions of **message** formats:
 - **data structures.**
 - **application programming interface (API).**
 - messages for data, signalling & management.
- A set of rules:
 - **algorithm.**
 - finite state machine (FSM).
- Error handling:
 - part of the FSM, API, special messages.

Simple example : phone call (1)

| | User A | Network | User B | |
|----|-----------------|---------------------------------------|-----------------|--|
| 01 | pick up phone → | ← dial-tone | | } set-up (allocate network resources) |
| 02 | dial number → | parse/check number & route the call → | | |
| 03 | | ← ringing → | (→ phone rings) | |
| 04 | | ← connect call → | ← pick up phone | |
| 05 | talk/listen | (data between user A and User B) | talk/listen | } data transfer |
| 06 | hang-up → | | ← hang-up | |
| 07 | | tear-down call | | } release (network resources) |

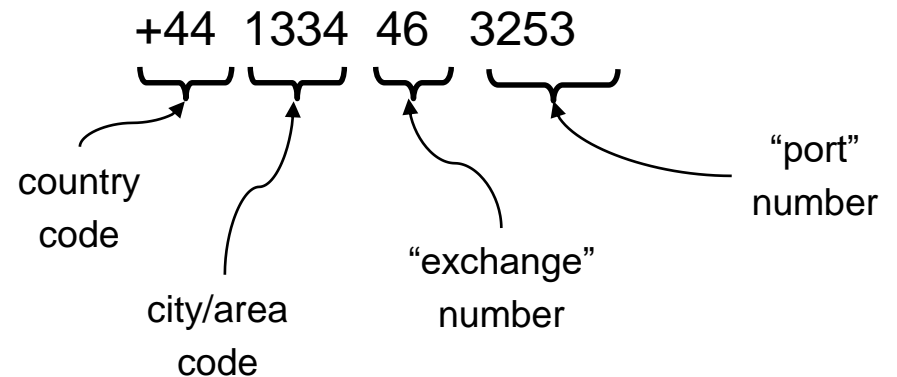
The arrows “→” and “←” indicate **signalling** to/from the network service.

At step 05, a high-layer protocol is used which is **transparent** to this layer.

Note that the protocol described above is **connection oriented**.

Simple example : phone call (2)

- **Addressing:**
 - location (topology)
 - identity (uniqueness)
- Phone number structure:
 - country code
 - city code
 - “exchange” number
 - port number on exchange
 - hierarchical, with respect to topology: routing information **aggregation**



Encapsulation (1)

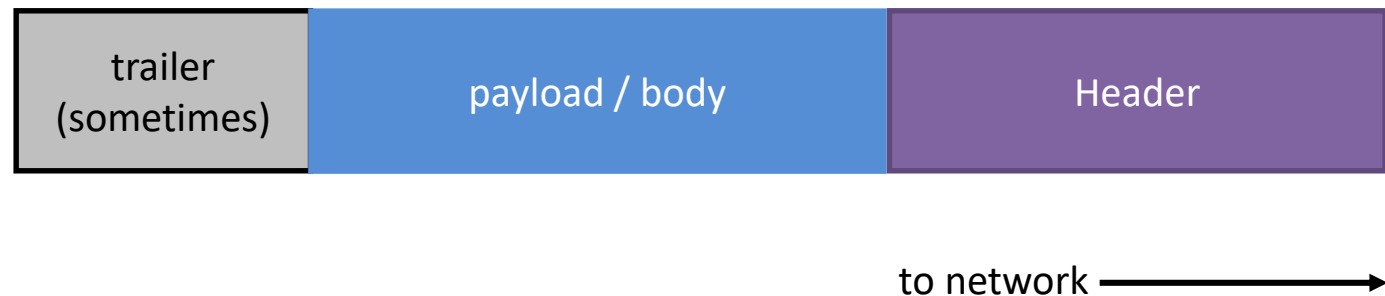
- When we send data / information across the network, it needs to have additional information:
 - An address for destination (and a source).
 - Information about message ordering.
 - Information about message formatting, e.g. document.
 - (plus other information, e.g. protocol, size, error control)
- For example, Fluff Intl Ltd using the courier service:
 - The address of a recipient, and a sender's address.
 - Envelopes marked as 1/3, for example.
 - Labelling to indicate it is for "payroll".

Encapsulation (2)

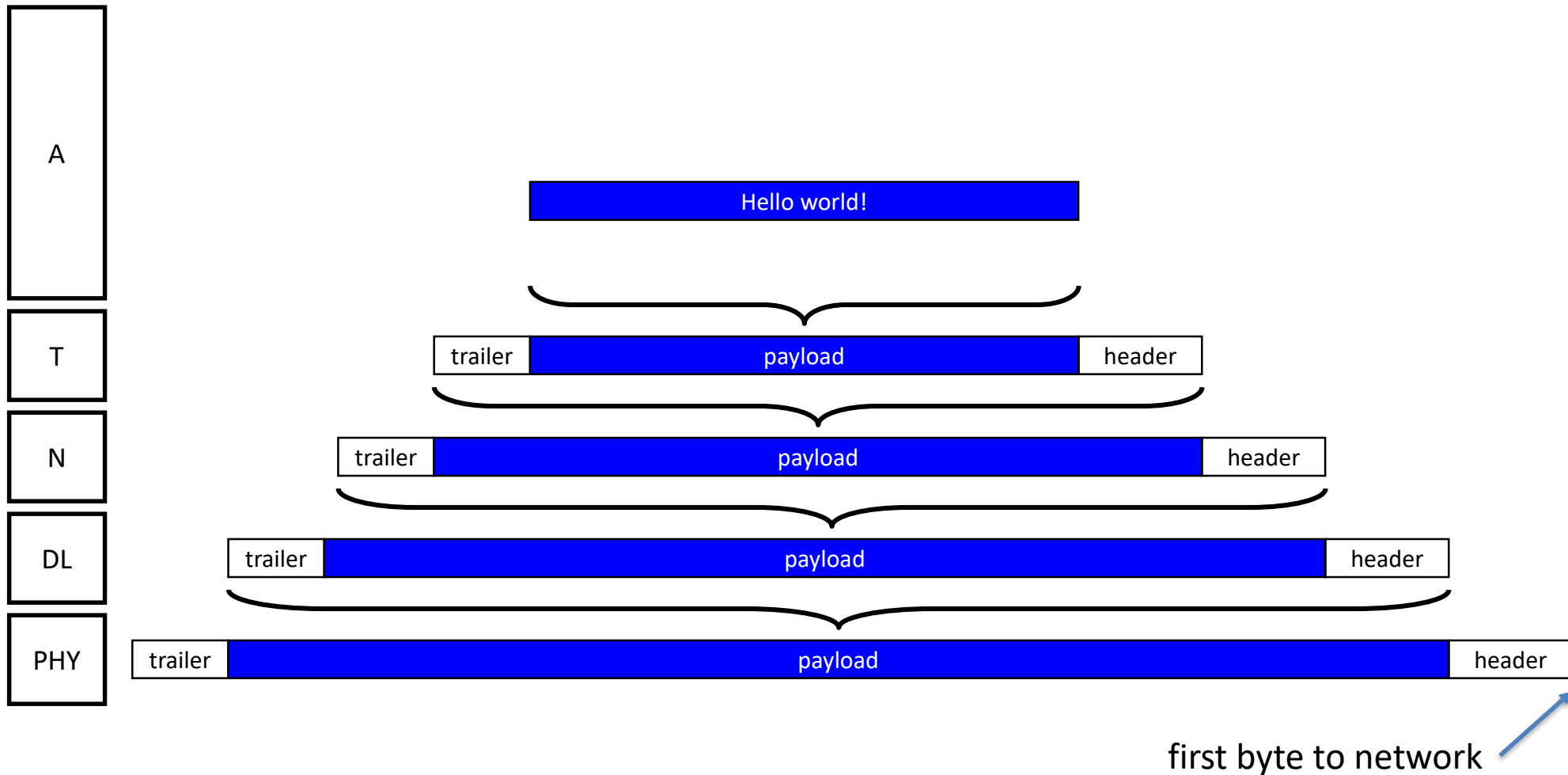
- Message format:
 - **Header** (possible a **Trailer**) + Payload / Body.
- **Header** allows protocol to operate, e.g.:
 - Address.
 - Message numbering.
 - Message type.
 - (also possibly a Trailer).
- Payload / Body is the data from the layer above:
 - What is carried by that layer.

Encapsulation (3)

- Message:
 - Well-defined data-structure as part of a protocol.
 - Header (possibly a trailer).
 - Payload / Body.
- Header (trailer): protocol control information.



Encapsulation (4)



Message format: terminology

- **XPDU**: *X* protocol data unit, e.g. APDU is application data unit (application message).
- **segment**: TCP (segment is part of a stream / flow)
- **datagram**: UDP and IP (self-contained, e.g. has addressing)
- **frame**: link layer (often has trailer as well as header)
- **packet**: general term, context specific.

Summary

- Layered architecture, each layer:
 - Has a specific purpose.
 - Provides a service to the layer above.
 - Has a function that depends on one or more protocols.
- Protocols:
 - Data structures plus algorithm / rules for operation.
 - Connectionless (CL) or connection-oriented (CO).
 - Addressing (much more to come on this).