

Jerarquía de protocolos

Adaptación

Juan Felipe Muñoz Fernández

Necesidad de modularidad

- Funciones
 - Iniciar y terminar las conexiones
 - Encontrar un camino a través de toda la red
 - Transportar información de manera confiable
 - Transportar información de cualquier tamaño
 - Enviar información tan rápido como se pueda
 - Compartir el ancho de banda entre los usuarios
 - Asegurar la información mientras está en tránsito
 - Permitir que nuevos nodos sean añadidos a la red
 - Otras...

(Wetheral)

Necesidad de modularidad

- Funciones

- Iniciar y terminar las conexiones
- Encontrar un camino a través de toda la red
- Transferir datos de un nodo a otro
- Transferir datos de un nodo a otro
- Encuentra un camino a través de toda la red
- Controlar el acceso a los recursos
- Asignar recursos a los usuarios
- Permitir que nuevos nodos sean añadidos a la red
- Otras...

Necesitamos una forma de dividir la responsabilidad de todas estas funciones para ayudar a manejar la complejidad de las mismas

(Wetheral)

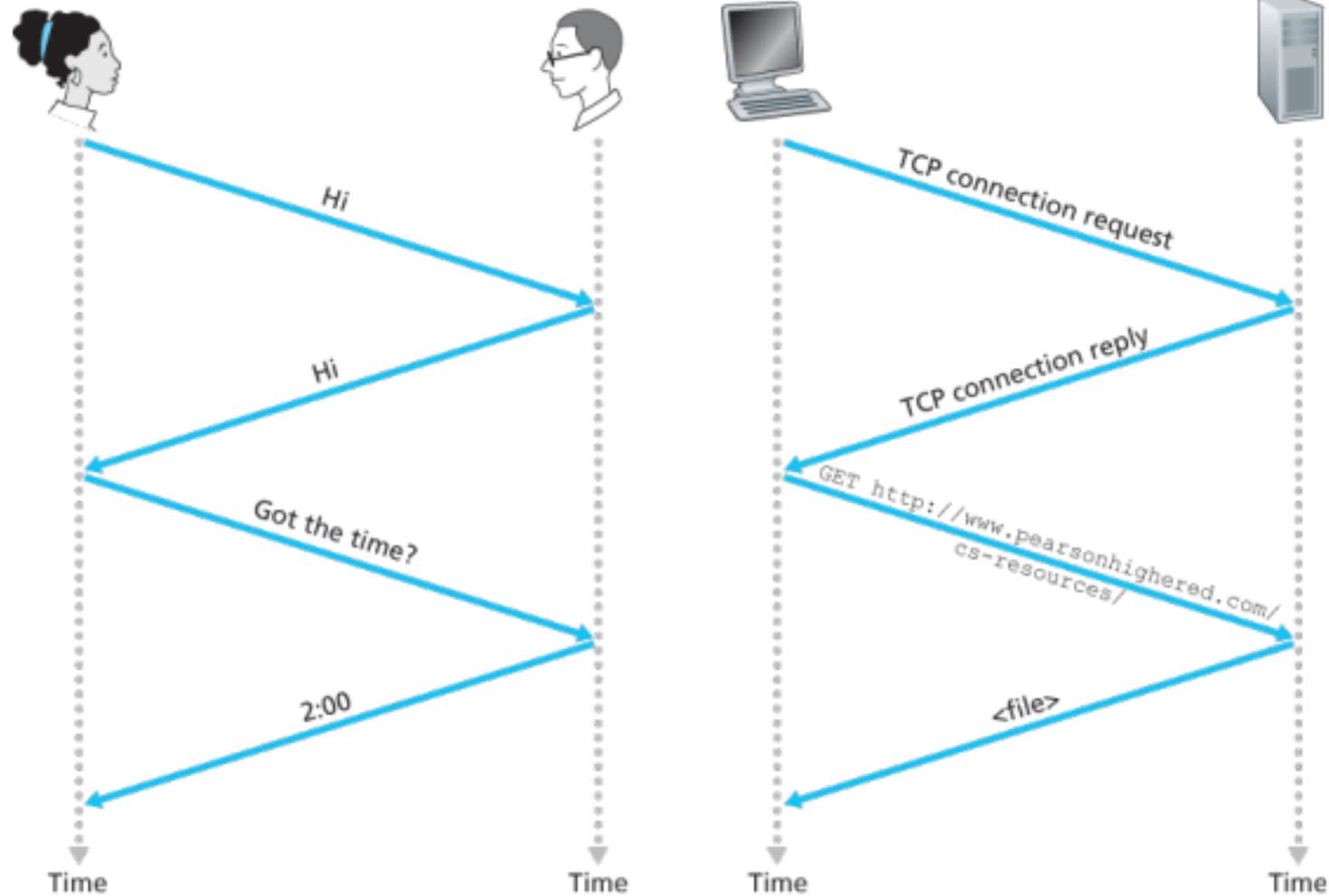
Protocolos y capas

- Estructura que permite **dividir** las funciones de la red
 - Cada instancia de un protocolo **conversa** virtualmente con su **par** (capa) usando el **protocolo** de dicha capa.
 - Cada instancia de un protocolo **usa** únicamente los **servicios** de la **capa inferior**.

Protocolos

- Definen el **formato** de los **mensajes** y el **orden** en el que son intercambiados los mensajes entre dos entidades (Kurose & Ross, 2017).
- Definen las **acciones** que se toman en la transmisión y/o recepción de los mensajes (Kurose & Ross, 2017).
- Algunos ejemplos
 - **Control de flujo de los bits** entre dos tarjetas de red.
 - Protocolos **de control de congestión**.
 - Protocolos de enrutamiento de paquetes.
 - Otros... ¡muchos!

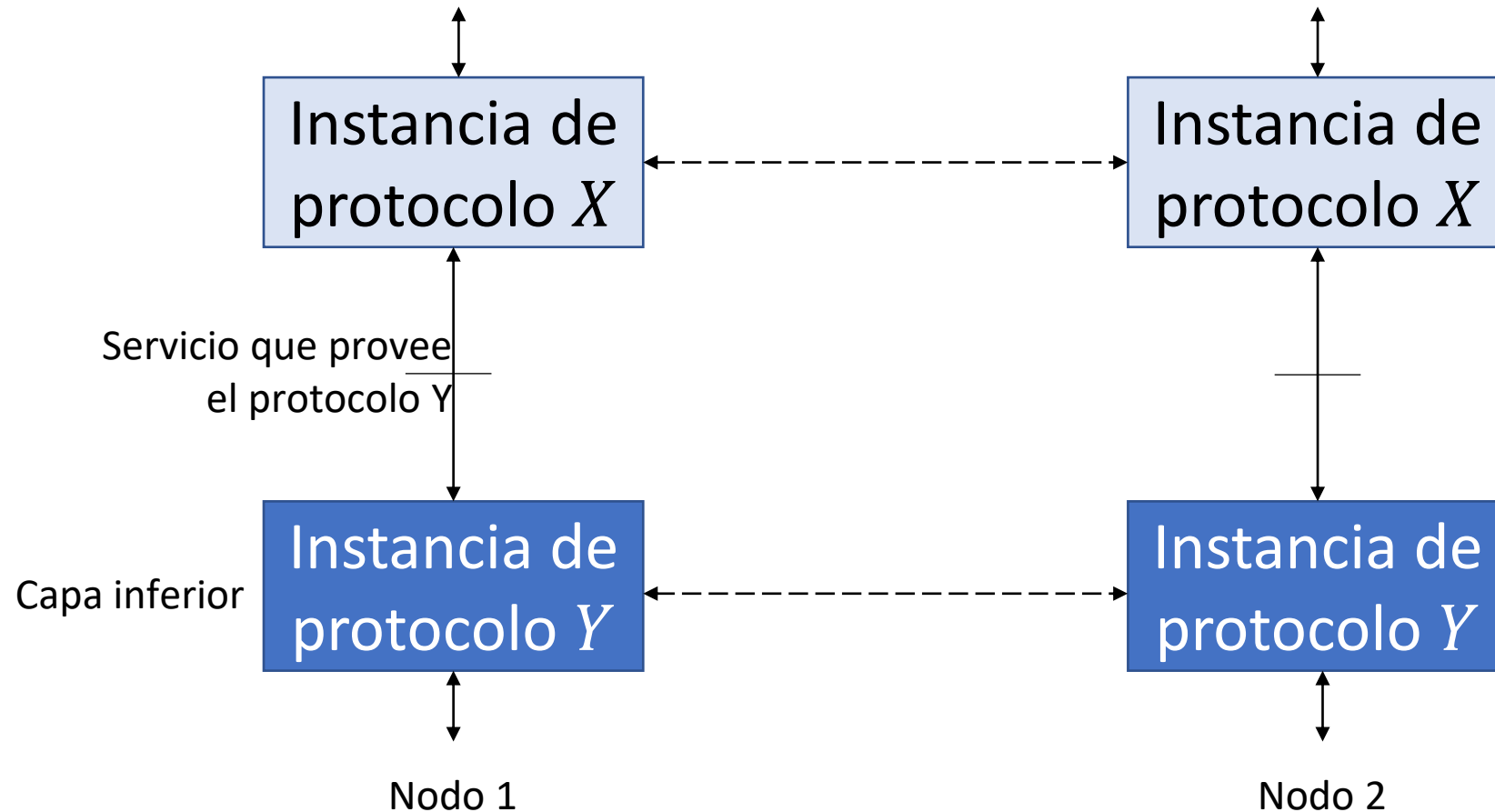
Protocols



Protocolos y capas

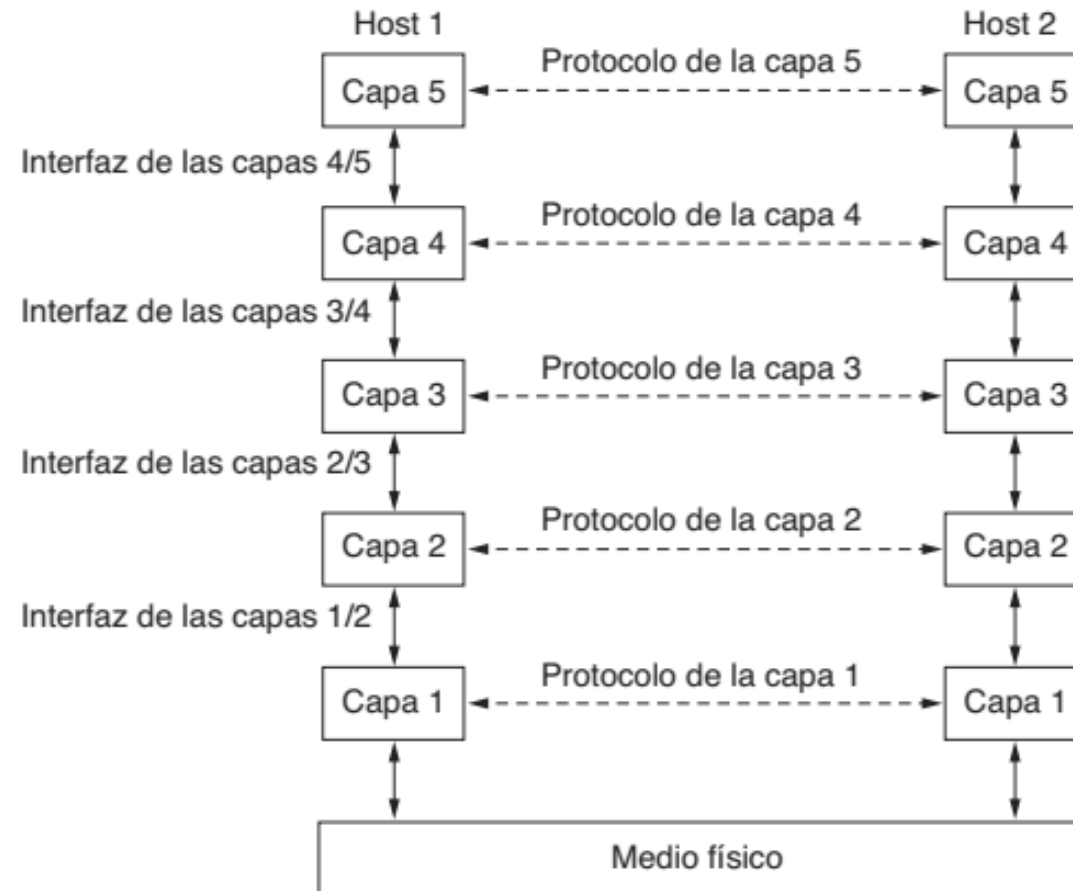
Protocolos y capas

- Protocolos horizontales, capas verticales



Protocolos y capas

- **Pila de protocolos** (*protocol stack*): Conjunto de protocolos en uso



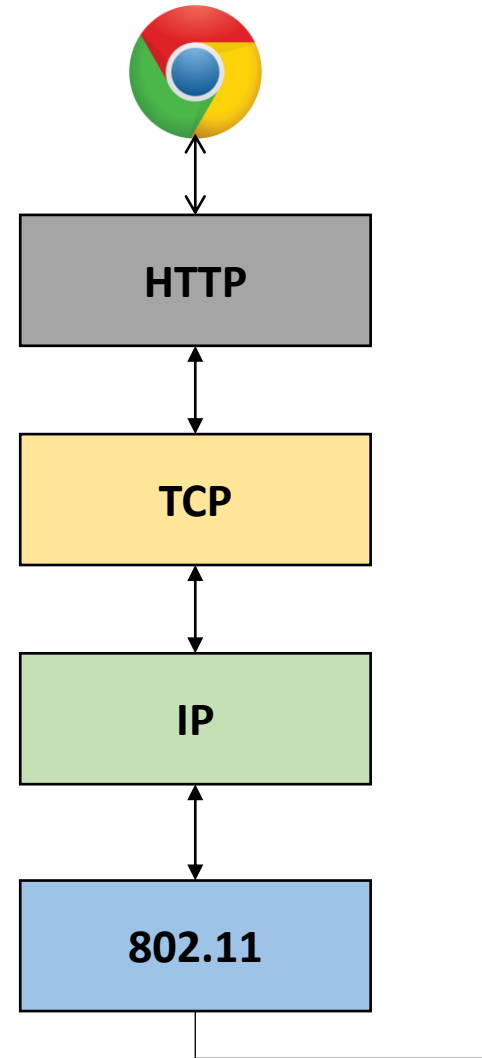
(Tanenbaum & Wetherall, 2012)

Protocolos y capas

- Algunos protocolos
 - HTTP, DNS, TCP, IP, ARP, SIP, Ethernet, 802.11 entre muchos, muchos otros.
- Ejemplo de pila de protocolos
 - Un navegador que se ejecuta sobre un host conectado de manera inalámbrica a la red.

Protocolos y capas

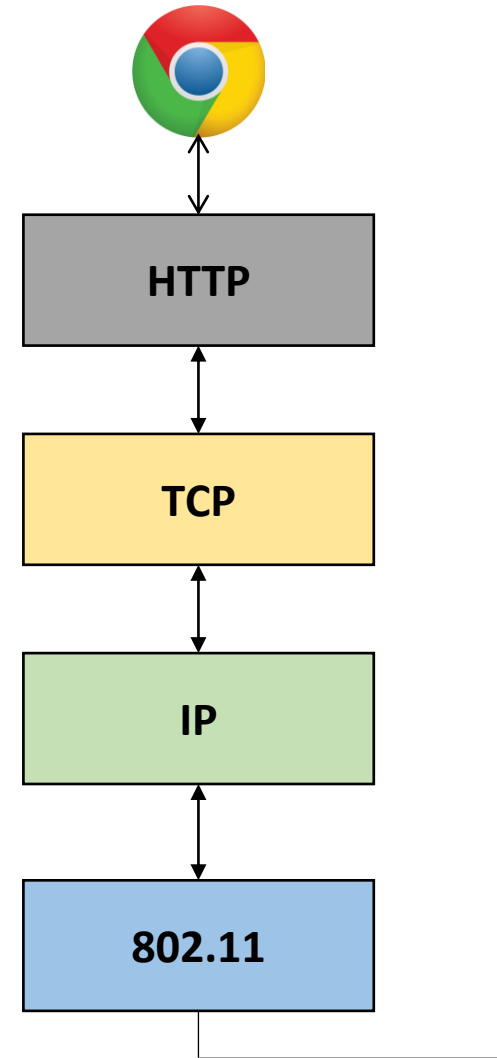
- Ejemplo de pila de protocolos



Encapsulamiento

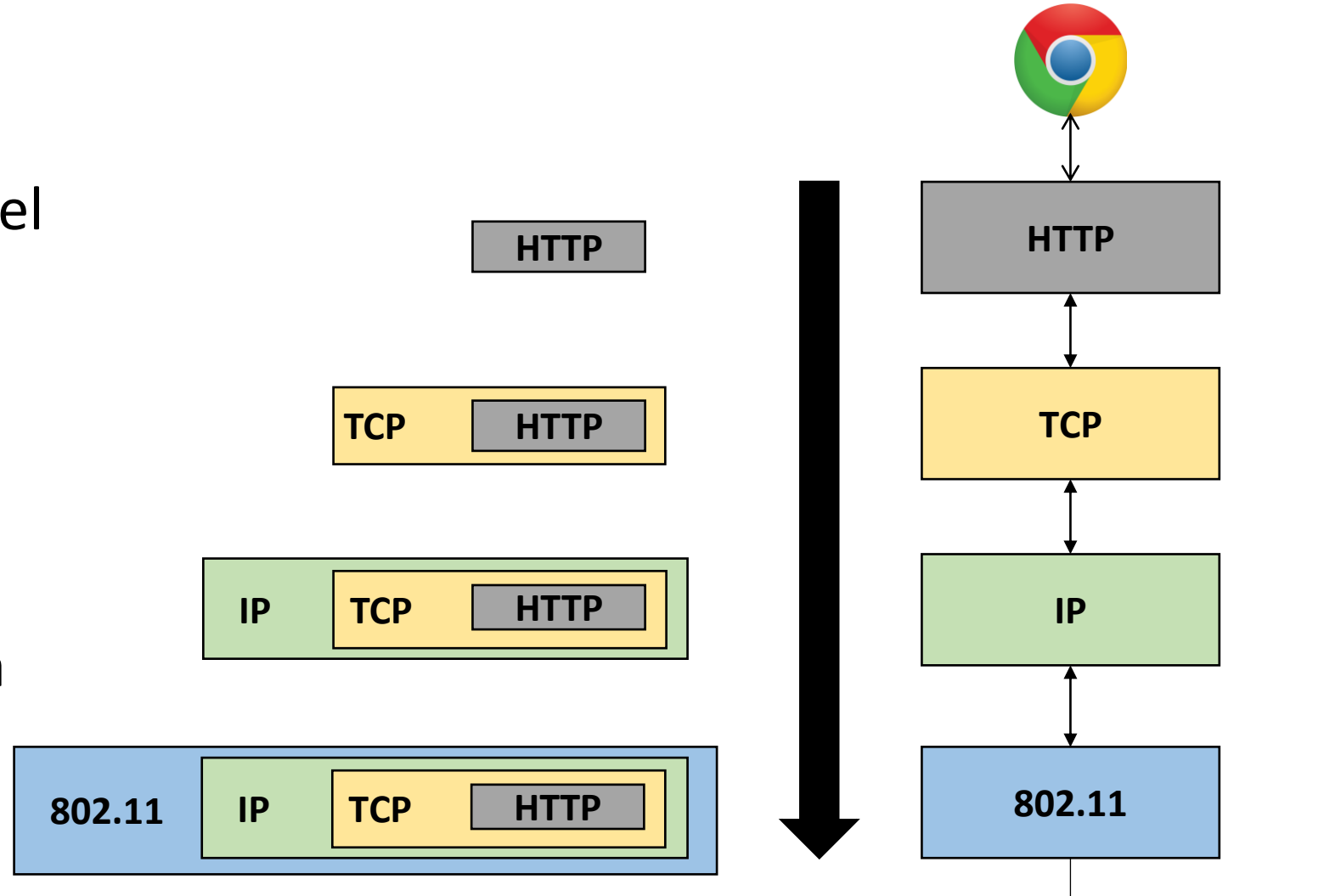
- Operación que se realiza en el nodo emisor
 - Capa inferior **envuelve** el mensaje de la capa superior
 - La capa inferior **le añade** información **de control** al envoltorio para generar un **nuevo mensaje**
 - Cada mensaje de cada capa: **PDU → Protocol Data Unit**
- Analogía: enviar un documento por correo postal
 - Otros procesos intermedios

Encapsulamiento



Encapsulamiento

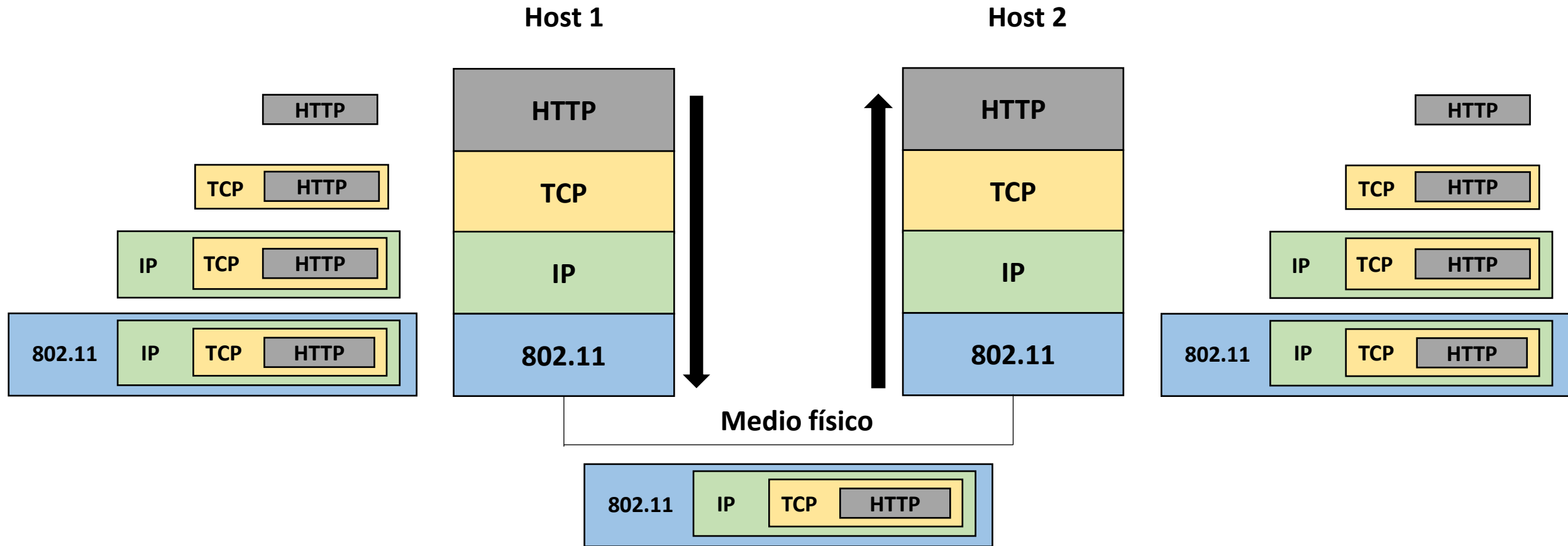
- Mensaje sobre el medio físico parece una cebolla
- Las capas más exteriores corresponden a las capas inferiores



Encapsulamiento

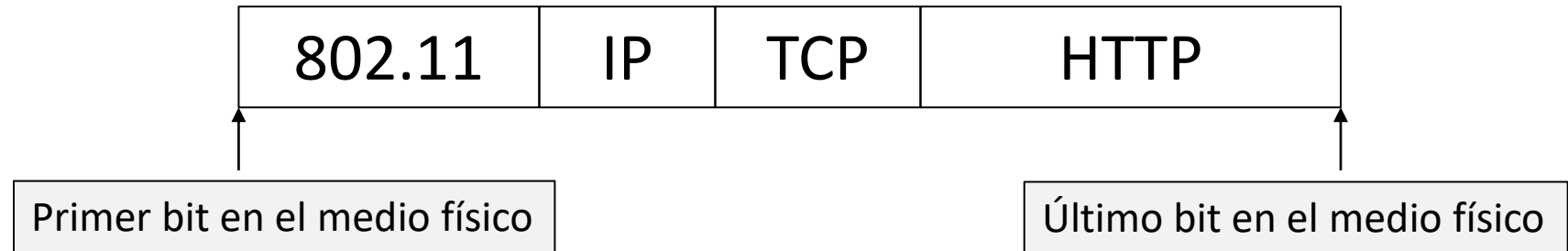
```
Frame 215: 1053 bytes on wire (8424 bits), 1053 bytes captured (8424 bits) on int  
Ethernet II, Src: Technico_90:45:8b (44:32:c8:90:45:8b), Dst: HewlettP_53:aa:8f (  
Internet Protocol Version 4, Src: 188.184.100.82, Dst: 192.168.0.16  
Transmission Control Protocol, Src Port: 80, Dst Port: 59931, Seq: 1461, Ack: 415  
[2 Reassembled TCP Segments (2459 bytes): #214(1460), #215(999)]  
Hypertext Transfer Protocol  
Line-based text data: text/html (73 lines)
```

Encapsulamiento



Encapsulamiento

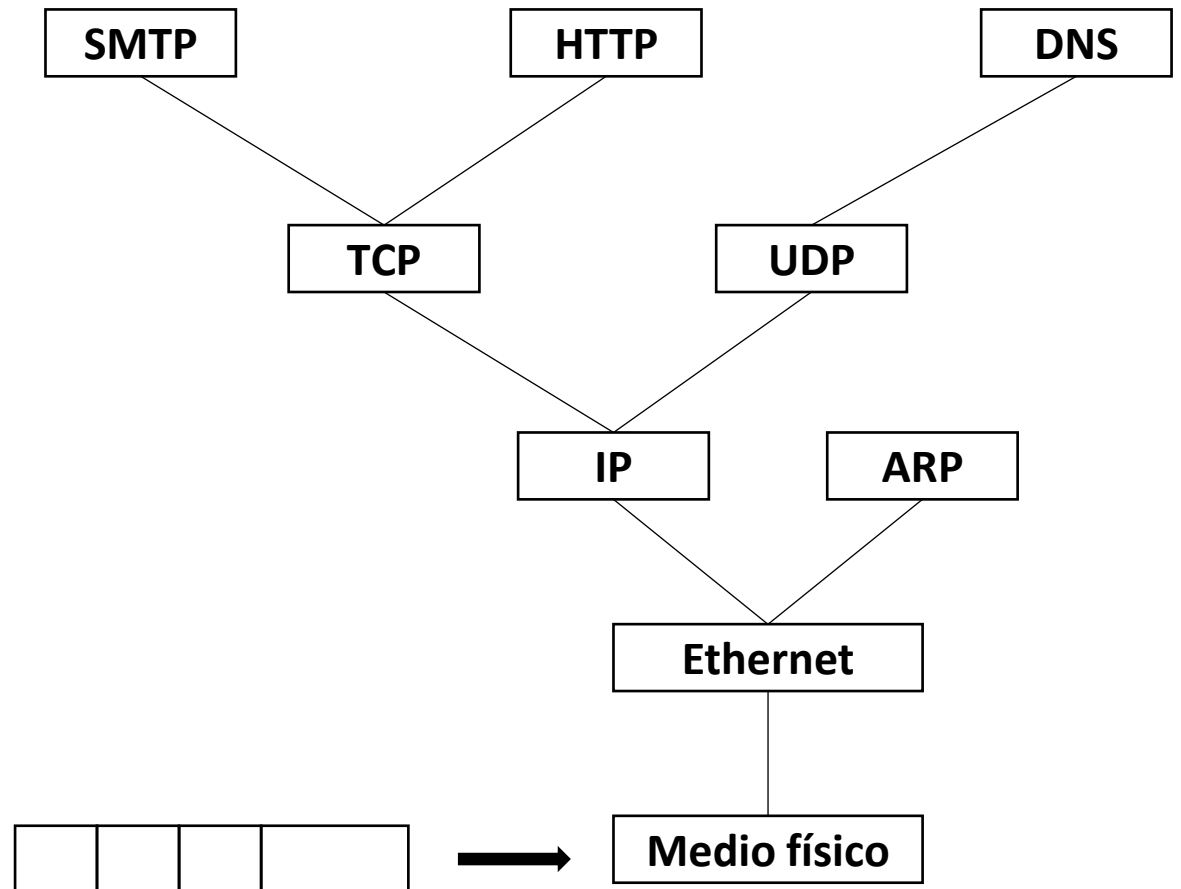
- Por simplicidad
 - Cada capa añade su propio encabezado



- En la práctica hay mas elementos
 - *Trailers*, encriptación, compresión de contenido del mensaje
 - Fragmentación y reensamblado

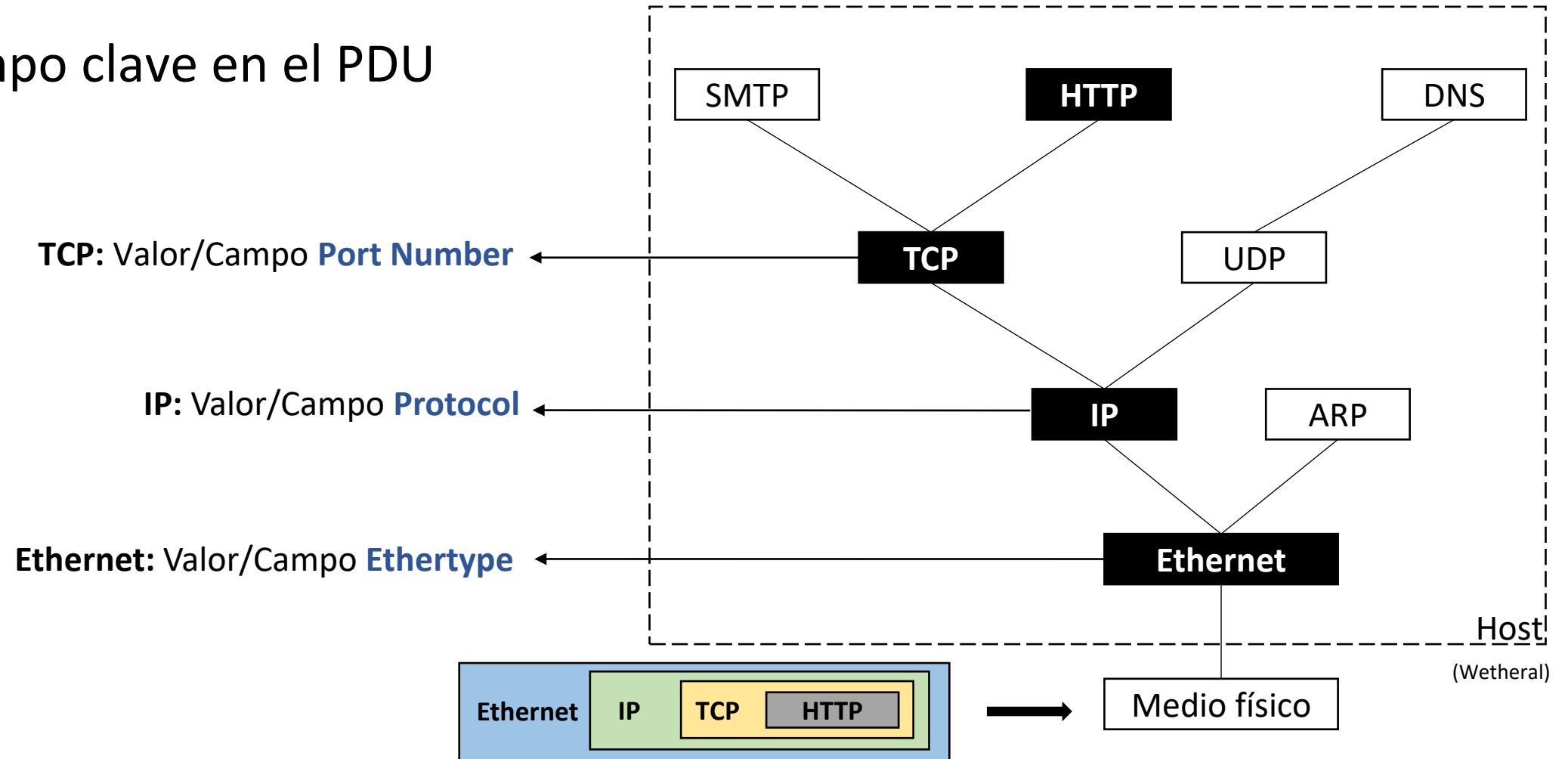
Demultiplexado

- Los mensajes que recibe un nodo deben entregarse a los protocolos que los usan
- Múltiples protocolos por capa



Demultiplexado

- Campo clave en el PDU



Hypertext Transfer Protocol

```
> GET /hypertext/WWW/TheProject.html HTTP/1.1\r\nHost: info.cern.ch\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\nAccept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3\r\nAccept-Encoding: gzip, deflate\r\nConnection: keep-alive\r\nReferer: http://info.cern.ch/\r\nUpgrade-Insecure-Requests: 1\r\n\r\n
```

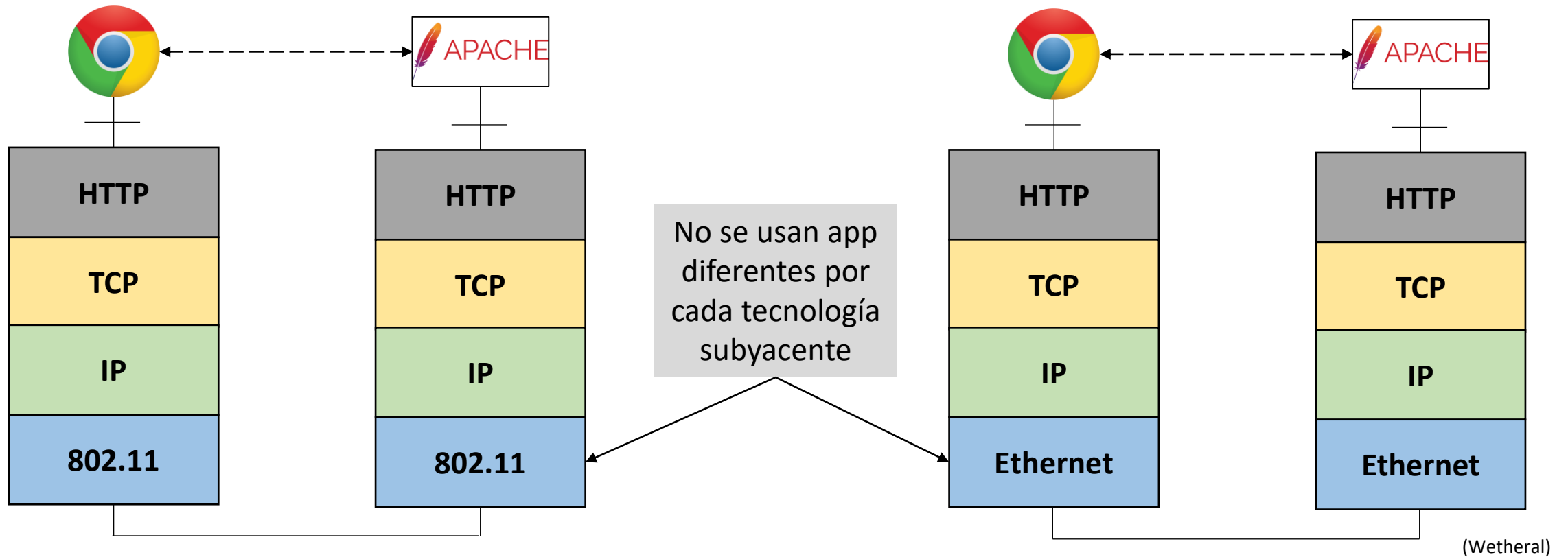
Transmission Control Protocol, Src Port: 59931, Dst Port: 80
Source Port: 59931
Destination Port: 80
[Stream index: 13]
[TCP Segment Len: 414]

Internet Protocol Version 4, Src: 192.168.0.16, Dst: 188.184.100.82
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 454
Identification: 0xdc3a (56378)
> Flags: 0x4000, Don't fragment
...0 0000 0000 0000 = Fragment offset: 0
Time to live: 128
Protocol: TCP (6)
Header checksum: 0x0000 [validation disabled]

Frame 209: 468 bytes on wire (3744 bits), 468 bytes captured (3744 bits) on interface \Device\NPF_{E...}
Ethernet II, Src: HewlettP_53:aa:8f (ec:8e:b5:53:aa:8f), Dst: Technico_90:45:8b (44:32:c8:90:45:8b)
> Destination: Technico_90:45:8b (44:32:c8:90:45:8b)
> Source: HewlettP_53:aa:8f (ec:8e:b5:53:aa:8f)
Type: IPv4 (0x0800)

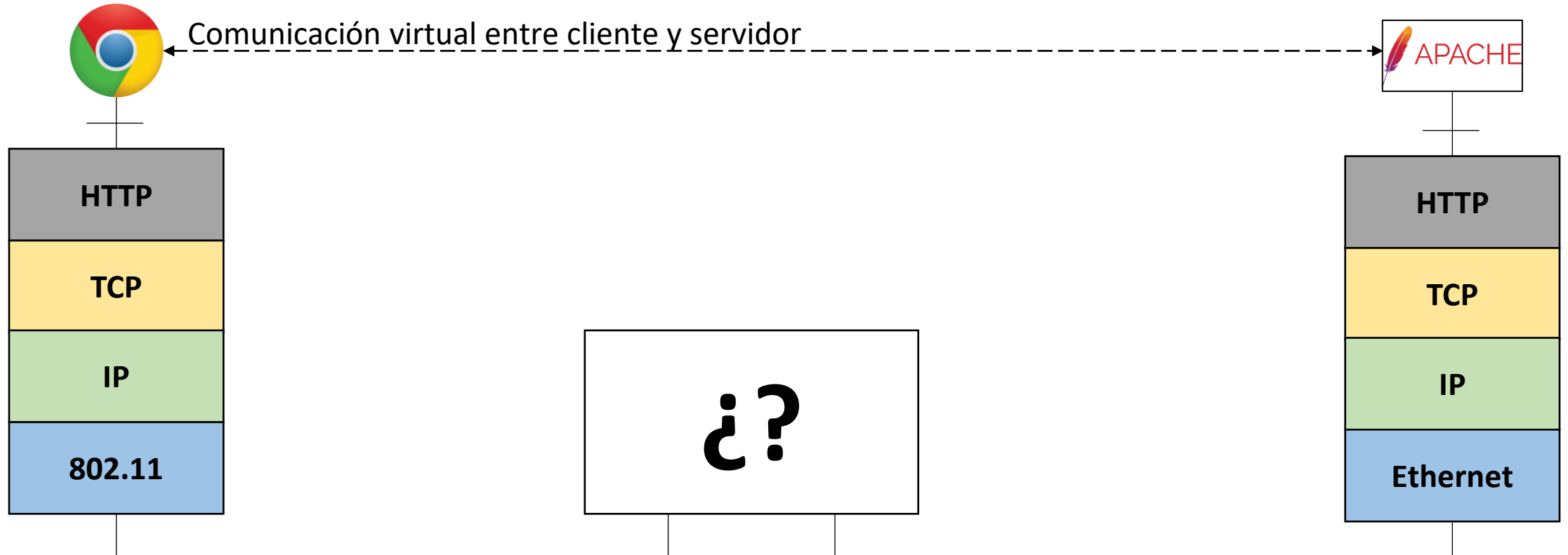
Ventajas

- Ocultamiento y reutilización: la capa superior no sabe de la tecnología subyacente



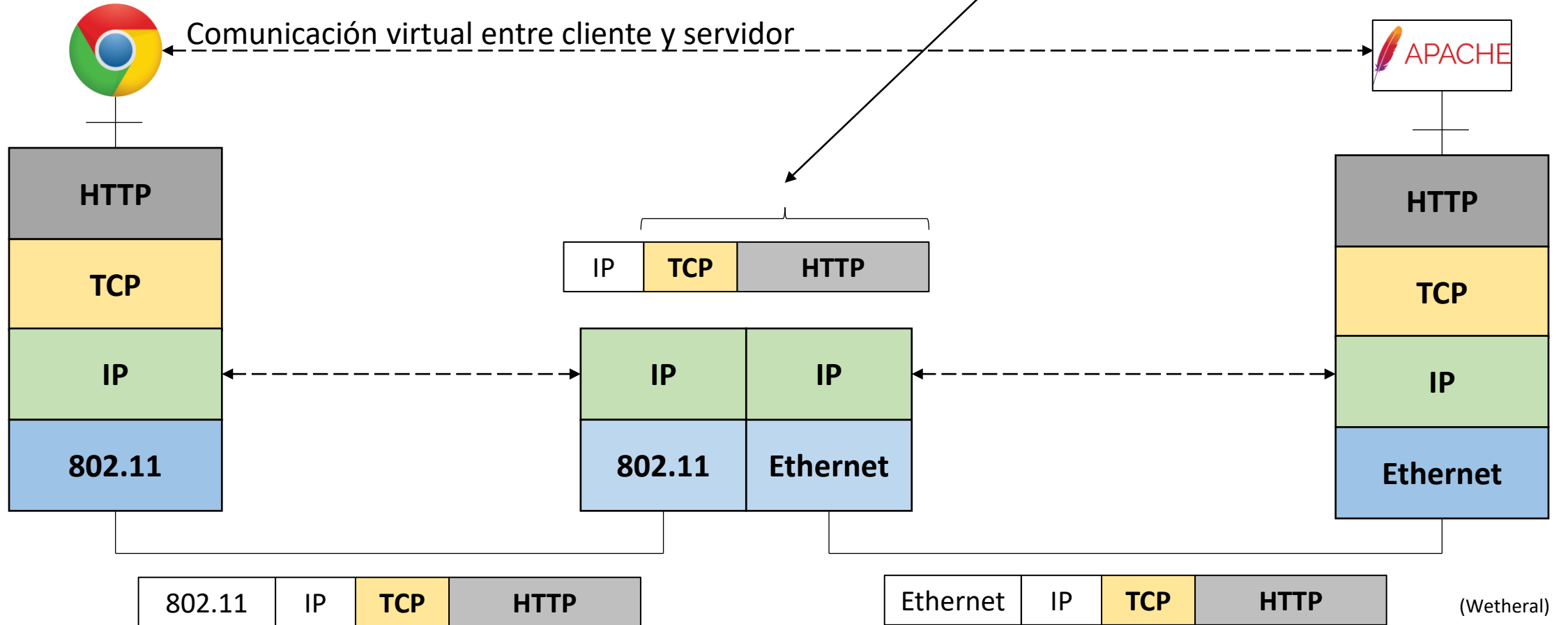
Ventajas

- Interacción entre diferentes tecnologías subyacentes



Ventajas

- Interacción entre diferentes tecnologías subyacentes



Desventajas

- **Añade *overhead***
 - En mensajes de gran tamaño sería menor el *overhead*
 - Enviar un byte → Mucho *overhead*
- **Ocultamiento de información**
 - Las aplicaciones podrían necesitar comportarse de cierta manera dependiendo de la tecnología subyacente
 - Adaptación: Gmail interfaz simple (no detecta recurso subyacente directamente)

Referencias

- Kurose, James F., and Keith W. Ross. “Chapter 1 Computer Networks and the Internet ” en *Computer Networking: a Top-down Approach*. 7th ed., Pearson, 2017.
- Tanenbaum A., and Wetherall D. “Introducción” en *Redes De Computadoras* 5th ed., Pearson Educación, México, 2012
- Wetherall, David. *1-6 Protocol Layers*.
www.youtube.com/watch?v=9d4kFRK0Rwg.