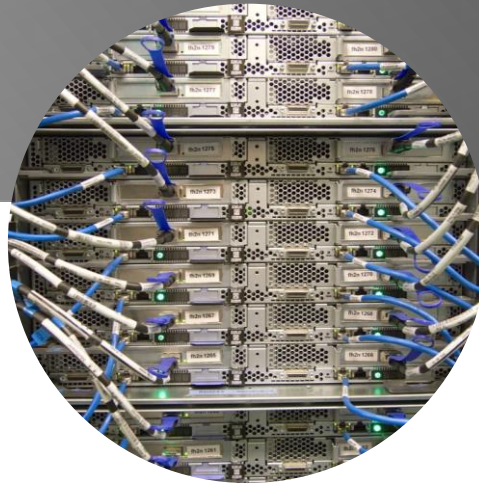


# Redes de computadores 2022 -1 (11310052)

David Felipe Celeita Rodriguez



Universidad del  
**Rosario**

Escuela de Ingeniería,  
Ciencia y Tecnología

---

Probability is expectation founded upon partial knowledge. A perfect acquaintance with all the circumstances affecting the occurrence of an event would change expectation into certainty, and leave nether room nor demand for a theory of probabilities.

George Boole



# Programa

Fecha (Sesión)	Tema
Sesión 1-2 24 Ene – 28 Ene	Introducción a redes de computadores Parte 1
Sesión 3-4 31 Ene – 4 Feb	Introducción a redes de computadores Parte 2
Sesión 5-6 7 Feb – 11 Feb	Capa de aplicación Parte 1
Sesión 7-8 14 Feb – 18 Feb	Capa de aplicación Parte 2
Sesión 9-10 21 Feb – 25 Feb	Capa de transporte Parte 1
Sesión 10 21 Feb – 25 Feb	PARCIAL 1



# Programa

Fecha (Sesión)	Tema
Sesión 11-12 28 Feb – 4 Mar	Capa de transporte Parte 2
Sesión 13-14 7 Mar – 11 Mar	Capa de red Parte 1 (Plano de datos)
Sesión 15-16 14 Mar – 18 Mar	Capa de red Parte 2 (Plano de datos)
Sesión 17-20 21 Mar – 25 Mar	Capa de red Parte 3 (Plano de control)
Sesión 17-20 28 Mar – 1 Abr	Capa de red Parte 4 (Plano de control)
Sesión 20 28 Mar – 1 Abr	PARCIAL 2





# Capítulo 4: Capa de red

Capa de red: descripción general

plano de datos

plano de control

¿Qué hay dentro de un enrutador?

puertos de entrada, conmutación, puertos de salida

gestión de búfer, scheduling

IP: el Protocolo de Internet

formato de datagrama

Direccionamiento

Traducción de Direcciones de RedIPv6

Reenvío generalizado, SDN

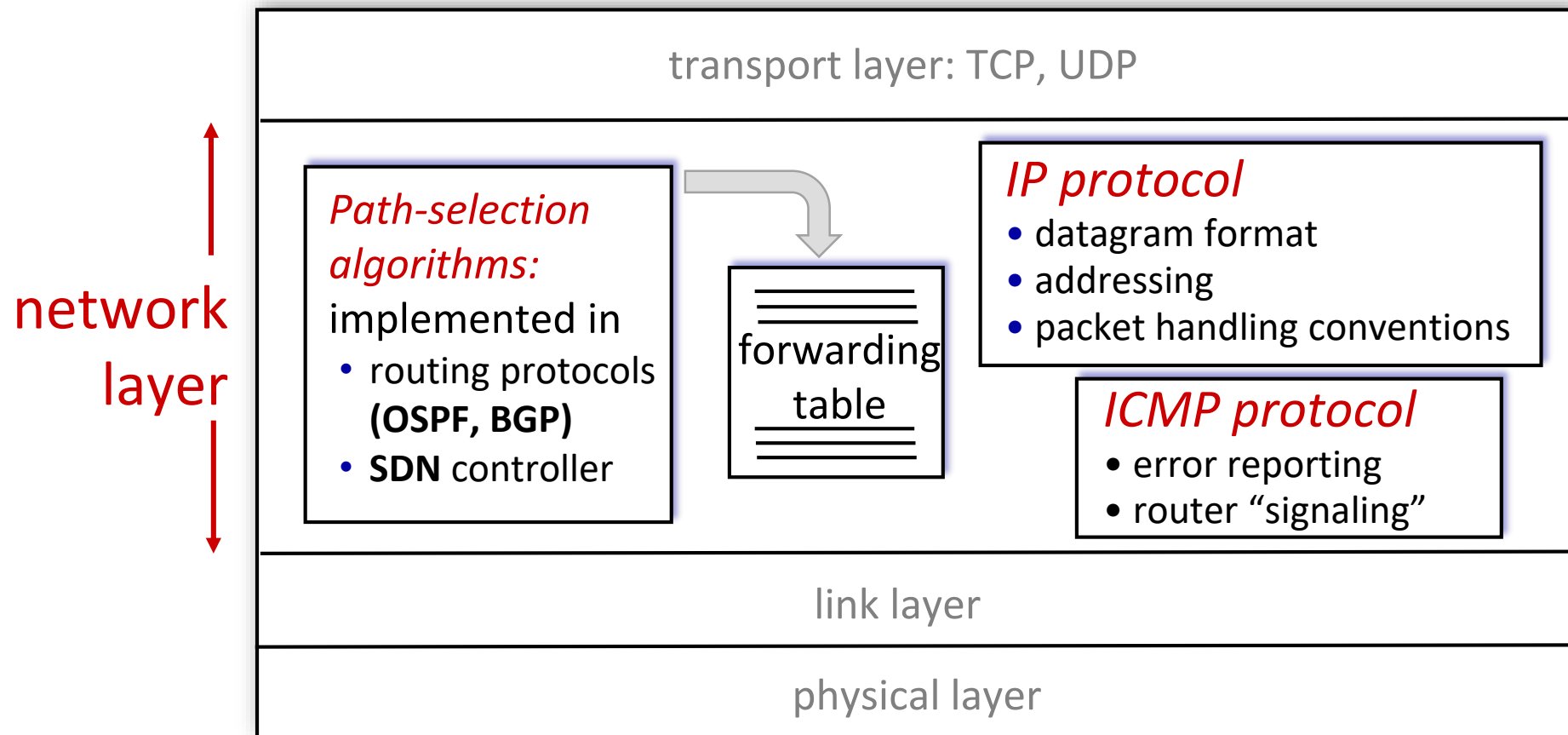
Match+action

OpenFlow: match+action en operación

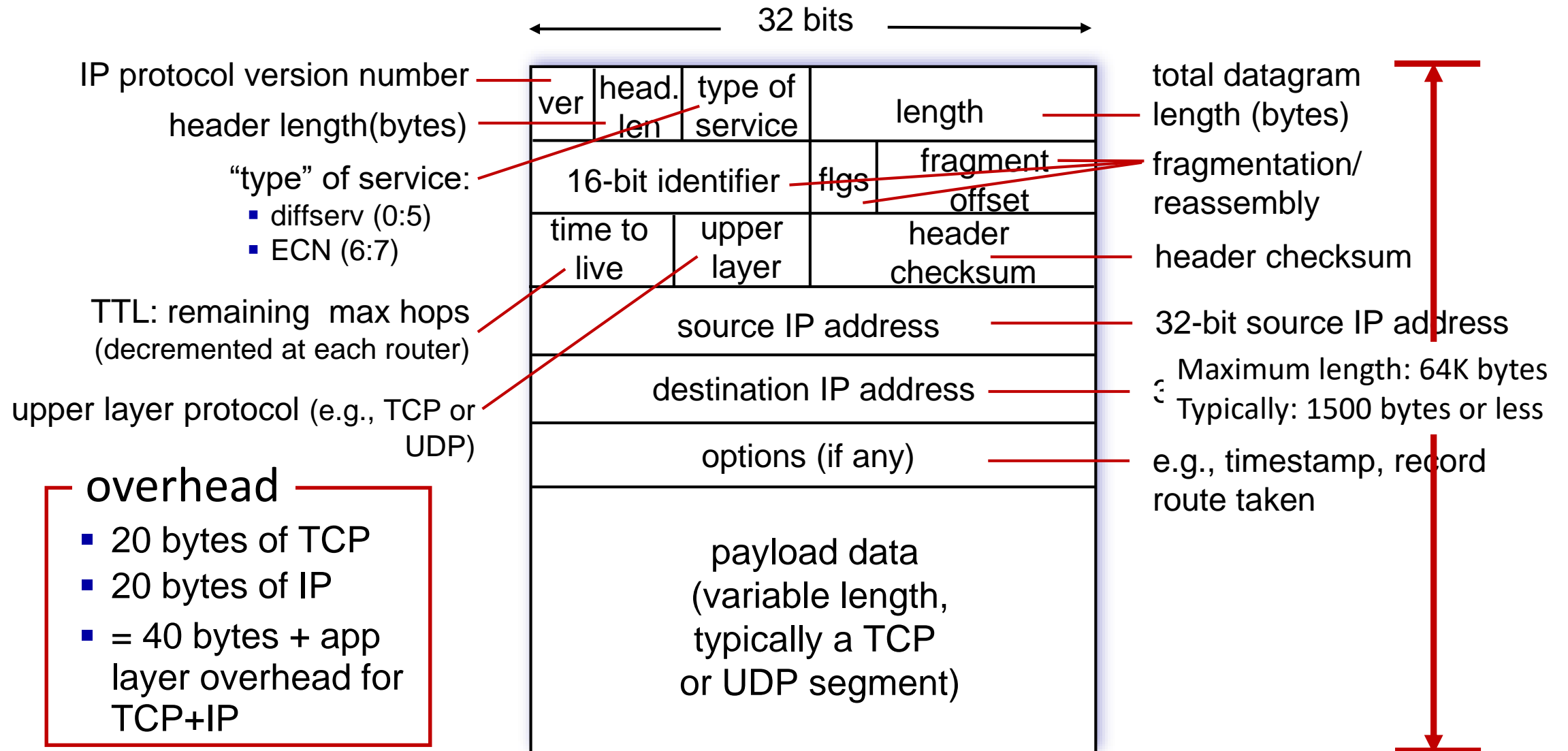
Middleboxes

# Capa de red: Internet (recordando)

Host y router: funciones en la capa de red



# IP Datagram (format)



# Direcciones IP (recordando)

¿Cómo obtiene un ISP un bloque de direcciones?

ICANN: Internet Corporation for Assigned Names and Numbers  
<http://www.icann.org/>

- asigna direcciones IP, a través de 5 registros regionales (RR) (que luego pueden asignar a los registros locales)
- gestiona root zone del DNS, incluida la delegación de la gestión de TLD individuales (.com, .edu,...)

¿Hay suficientes direcciones IP de 32 bits?

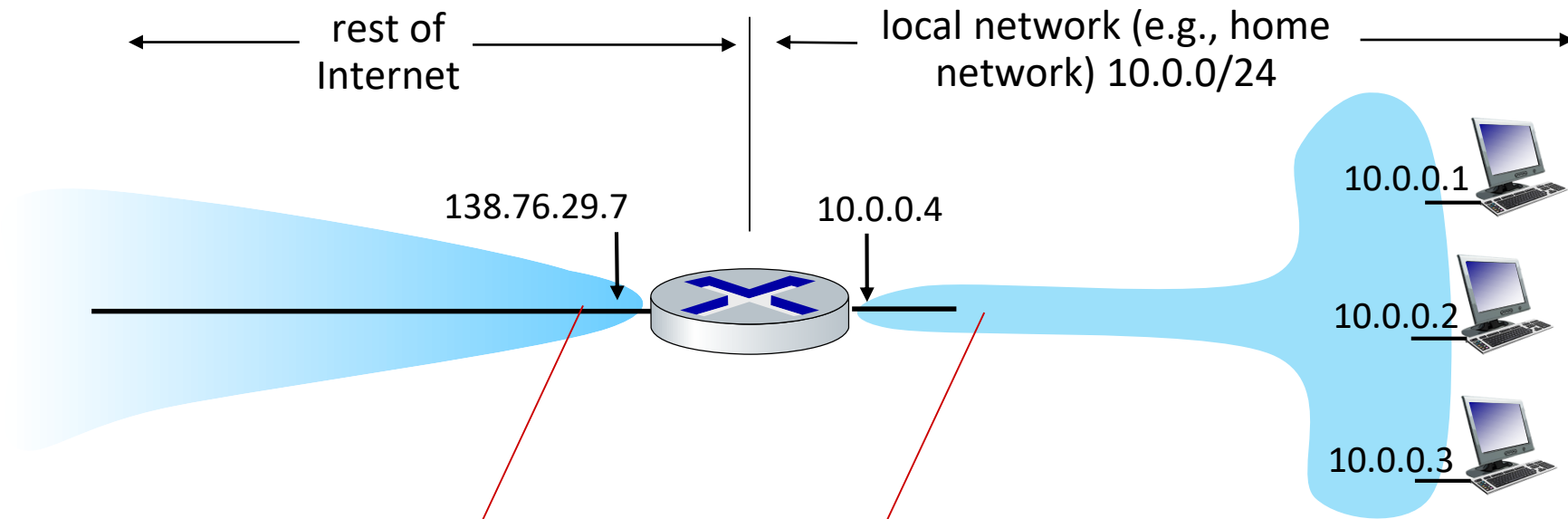
- La ICANN asignó la última parte de las direcciones IPv4 a los RR en 2011
- NAT (siguiente) ayuda al agotamiento del espacio de direcciones IPv4
- IPv6 tiene un espacio de direcciones de 128 bits

"¿Quién diablos sabía cuánto espacio de direcciones necesitábamos?" Vint Cerf (reflexionando sobre la decisión de hacer que la dirección IPv4 tenga una longitud de 32 bits)



# NAT: network address translation

**NAT:** todos los dispositivos en la red local comparten solo **una** dirección IPv4 en lo que respecta al mundo exterior



*todos* los datagramas **saliendo** red local tienen **la misma** dirección NAT IP de origen: 138.76.29.7, pero diferentes puertos de origen

Los datagramas con origen o destino en esta red tienen una dirección 10.0.0/24 para el origen.

# NAT: network address translation

- todos los dispositivos en la red local tienen direcciones de 32 bits en un espacio de dirección IP "privado" (prefijos 10/8, 172.16/12, 192.168/16) que solo se pueden usar en la red local
- Ventajas:
  - solo se necesita una dirección IP del proveedor ISP para todos los dispositivos
  - puede cambiar las direcciones del host en la red local sin notificar al mundo exterior
  - puede cambiar de ISP sin cambiar las direcciones de los dispositivos en la red local
  - seguridad: dispositivos dentro de la red local no directamente direccionables, visibles por el mundo exterior

# NAT: network address translation

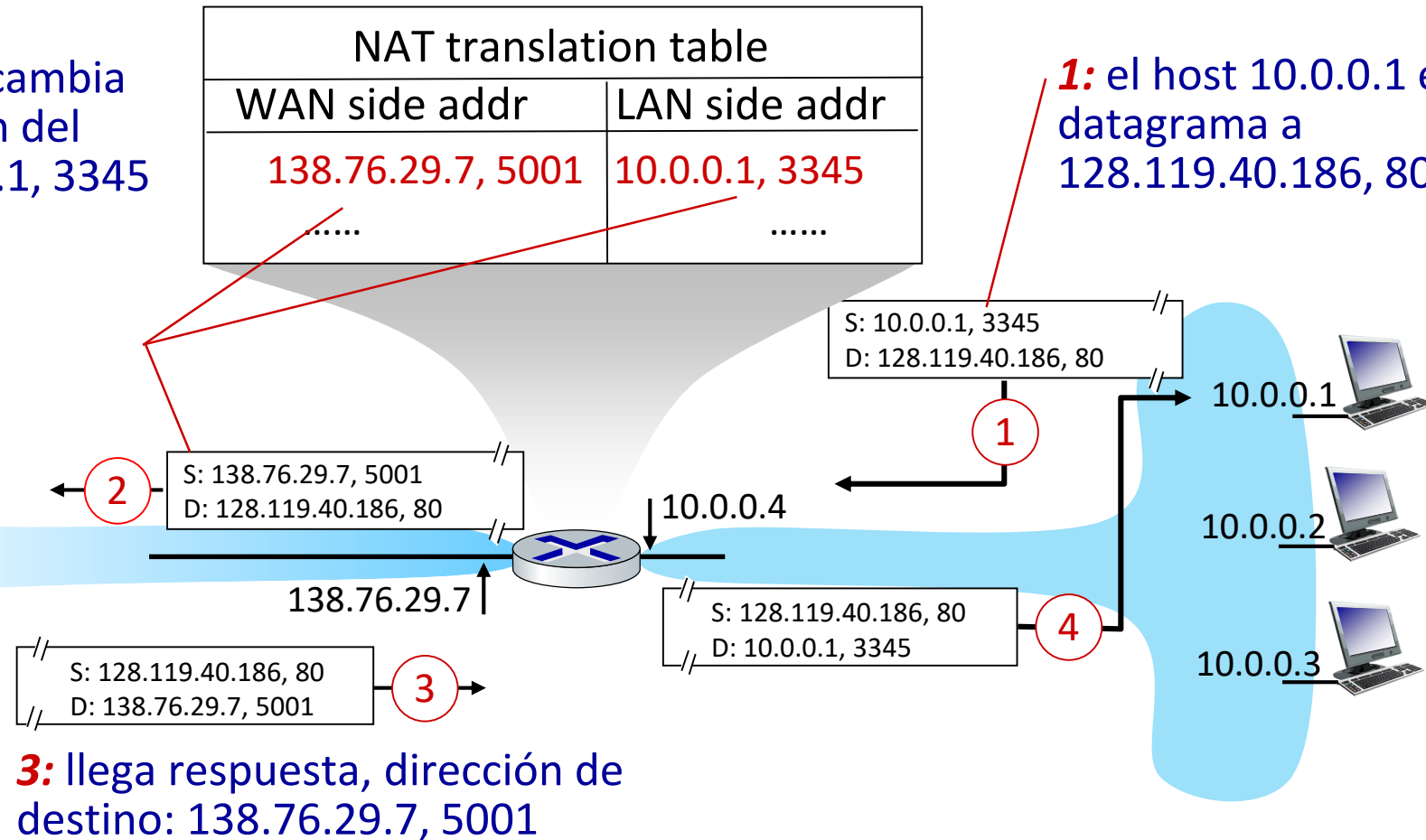
**implementación:** los enrutadores NAT deben (transparentemente):

- **datagramas salientes: reemplazar** (dirección IP de origen, número de puerto) de cada datagrama saliente a (dirección IP NAT, número de puerto nuevo)
  - Los clientes / servidores remotos responderán utilizando (dirección IP NAT, número de puerto nuevo) como dirección de destino
- **recordar (en la tabla de traducción de NAT)** cada par de traducción (dirección IP de origen, número de puerto) a (dirección IP de NAT, número de puerto nuevo)
- **Datagramas entrantes: reemplazar** (dirección IP NAT, número de puerto nuevo) en los campos de destino de cada datagrama entrante con el correspondiente (dirección IP de origen, número de puerto) almacenado en la tabla NAT

# NAT: network address translation

**2:** El enrutador NAT cambia la dirección de origen del datagrama de 10.0.0.1, 3345 a 138.76.29.7, 5001, actualiza la tabla

**1:** el host 10.0.0.1 envía el datagrama a 128.119.40.186, 80





# NAT: network address translation

- NAT ha sido controvertido:
  - los enrutadores "deberían" procesar solo hasta la capa 3
  - La "escasez" de direcciones debe resolverse mediante IPv6
  - viola el argumento end-to-end (número de puerto manipulado por un dispositivo de capa de red)
  - NAT transversal: ¿qué pasa si el cliente quiere conectarse al servidor detrás de NAT?
- pero NAT llegó para quedarse: ampliamente utilizado en redes domésticas e institucionales, redes celulares 4G / 5G

# IPv6: context y motivación

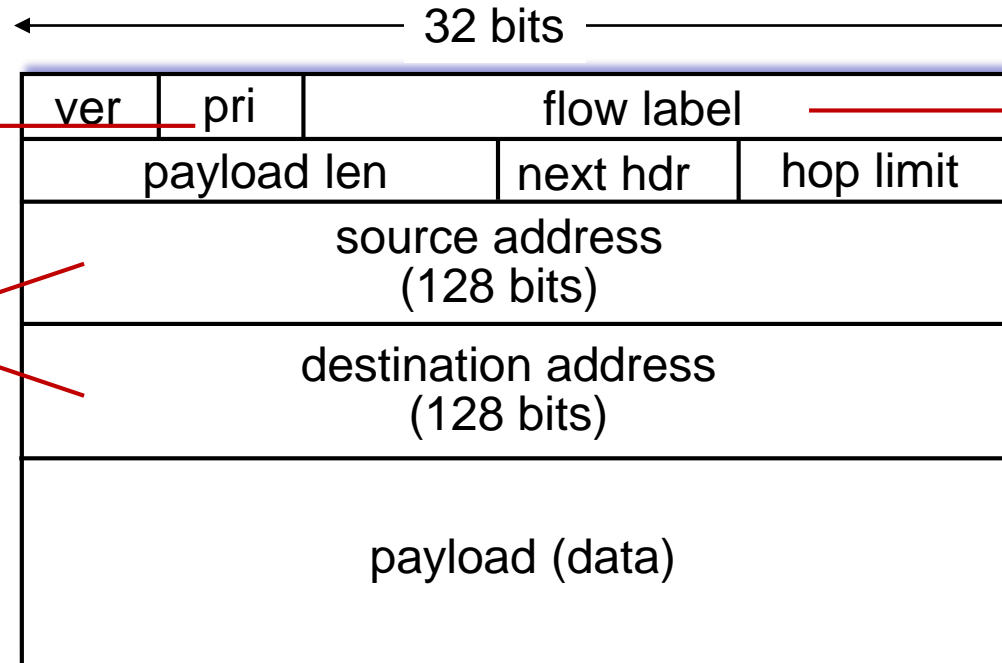
- **Motivación inicial:** el espacio de direcciones IPv4 de 32 bits se asignaría por completo
- Contexto:
- Velocidad de procesamiento/reenvío: encabezado de longitud fija de 40 bytes
- permitir un tratamiento diferente de la capa de red a los "flujos"

# IPv6 Datagram (formato)

**priority:** identificar la prioridad entre los datagramas en flujo

**flow label:** identificar datagramas en el mismo "flujo". (concepto de "flujo" no está bien definido).

**128-bit**  
IPv6 (direcciones)

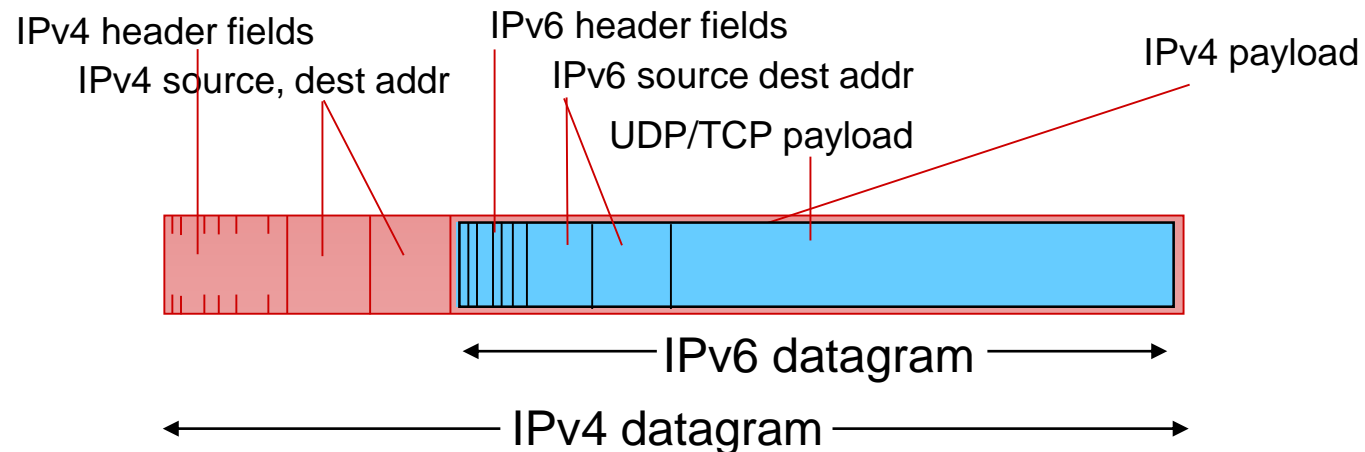


Qué falta (en comparación con IPv4):

- sin checksum (para acelerar el procesamiento en los enrutadores)
- sin fragmentación/reensamblaje
- sin opciones (disponible como protocolo de siguiente encabezado de capa superior en el enrutador)

# De IPv4 a IPv6

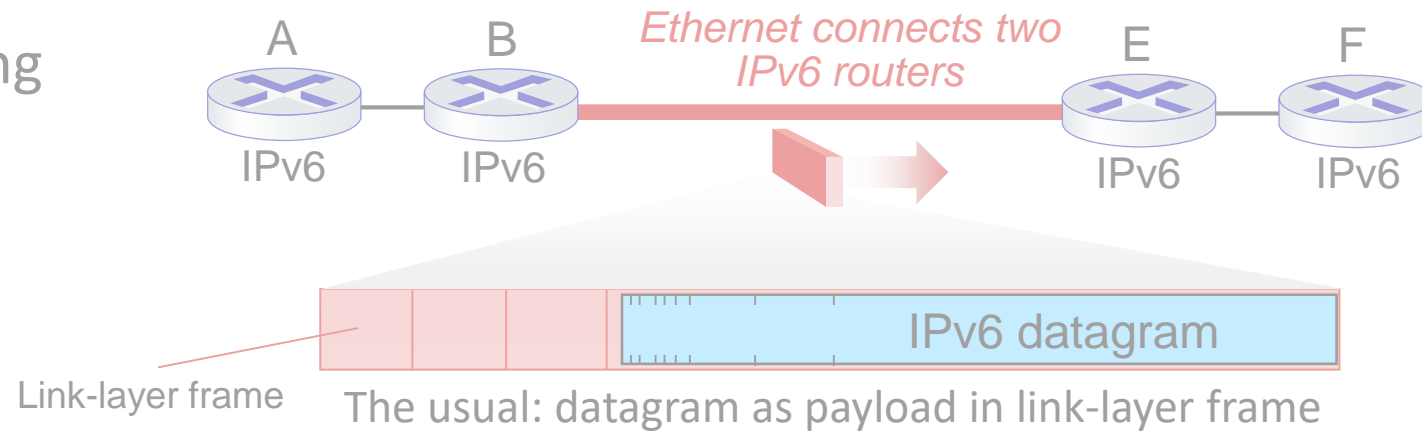
- no todos los enrutadores se pueden actualizar simultáneamente
- sin “flag days”
- ¿Cómo funcionará la red con enrutadores IPv4 e IPv6 mixtos?
- **tunneling**: Datagrama IPv6 transportado como payload en datagrama IPv4 entre enrutadores IPv4 ("paquete dentro de un paquete")
- Túneles utilizados ampliamente en otros contextos (4G/5G)



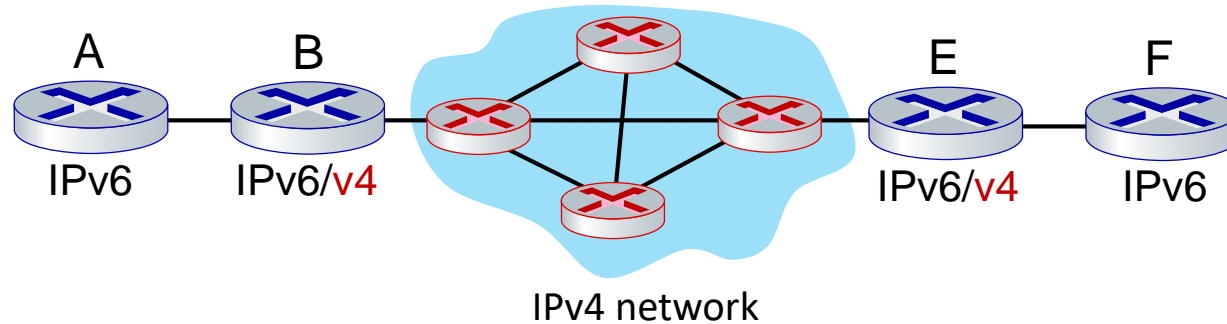


# Tunneling and encapsulation

Ethernet connecting two IPv6 routers:

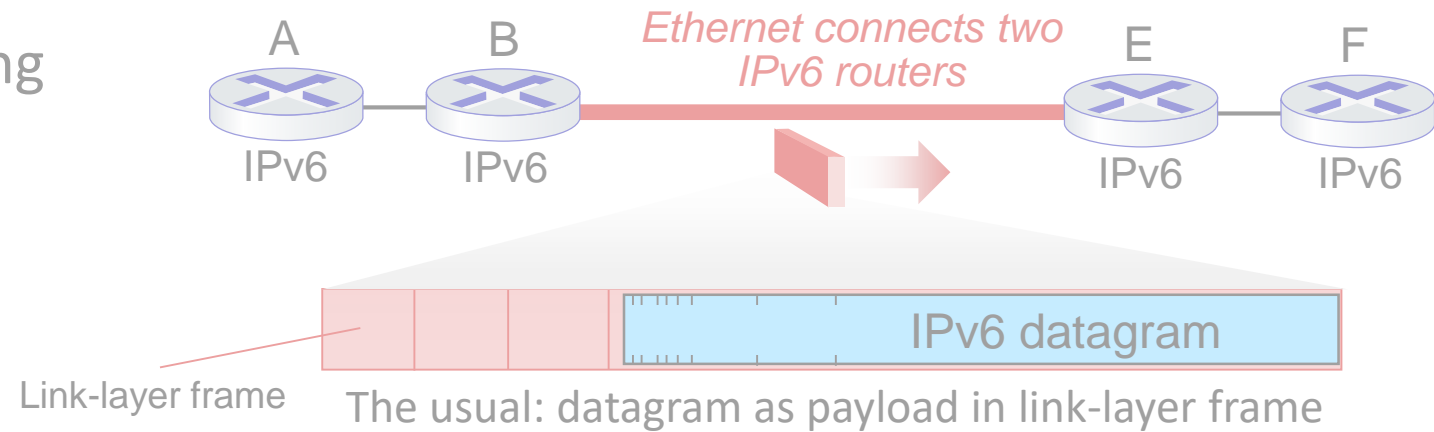


IPv4 network connecting two IPv6 routers

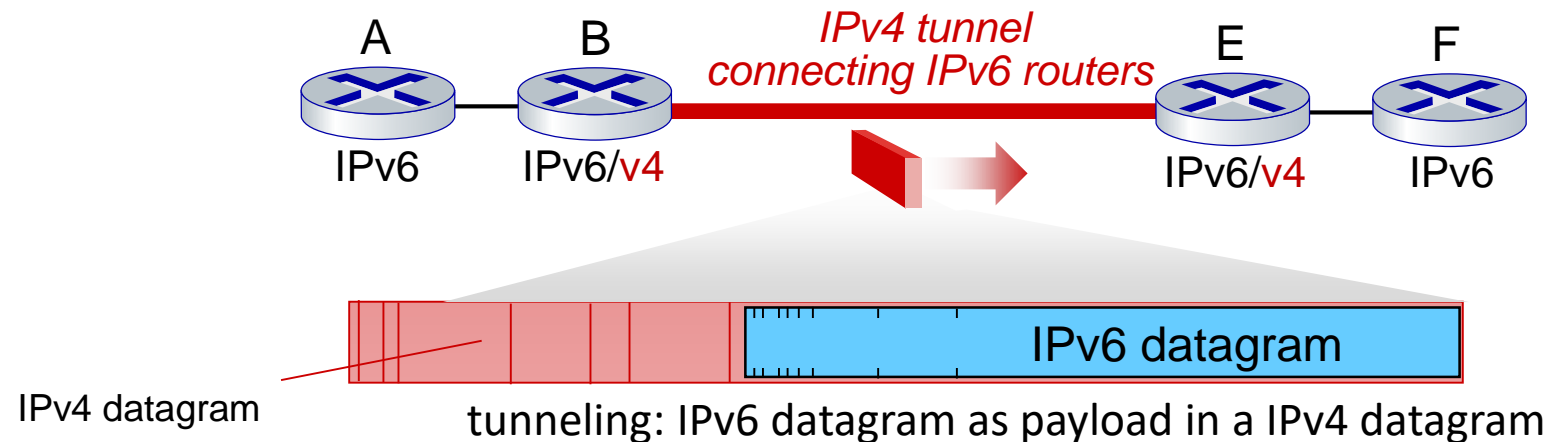


# Tunneling and encapsulation

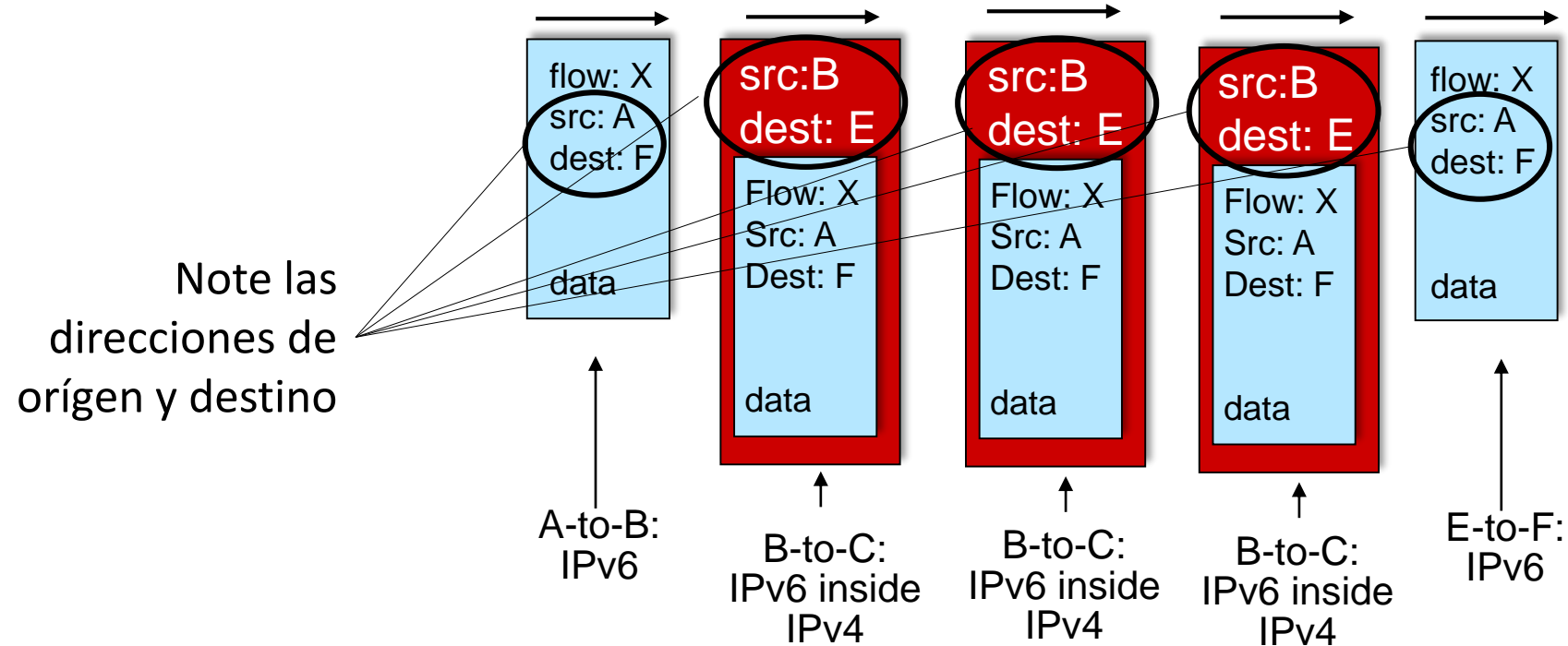
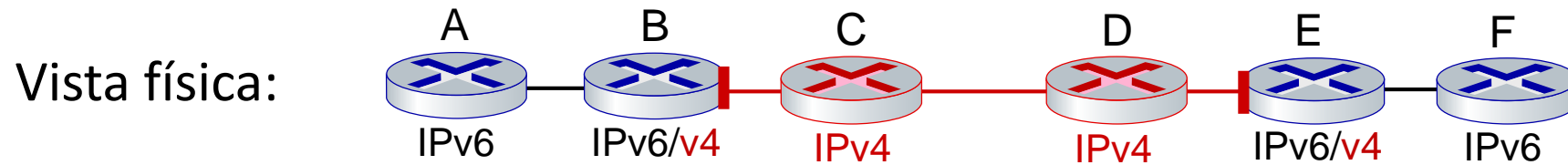
Ethernet connecting two IPv6 routers:



IPv4 tunnel connecting two IPv6 routers

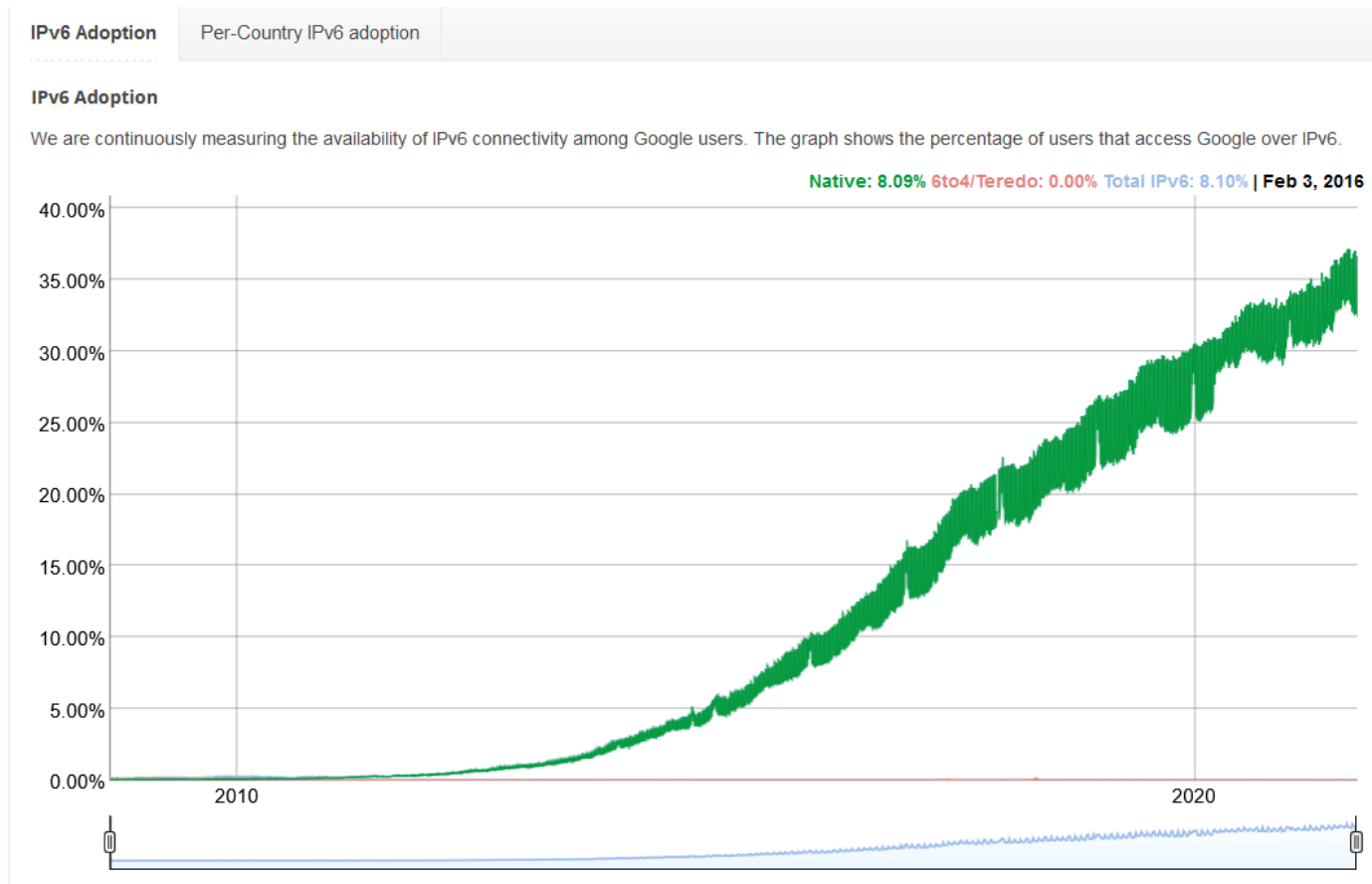


# Tunneling



# IPv6: transición y adopción reciente

- Google<sup>1</sup>: ~ 30% de los clientes acceden por IPv6
- NIST: 1/3 de todos los dominios del gobierno de EE. UU. son compatibles con IPv6



<https://www.google.com/intl/en/ipv6/statistics.html>



# IPv6: transición

- Google<sup>1</sup>: ~ 30% de los clientes acceden por IPv6
- NIST: 1/3 de todos los dominios del gobierno de EE. UU. son compatibles con IPv6
- Tiempo (muy) largo de implementación (25 años y contando)
- piense en los cambios a nivel de aplicación en los últimos 25 años: WWW, redes sociales, transmisión de medios, juegos, telepresencia,...¿Por qué?
- *¿Cómo vamos en Colombia con IPv6?*



# Capítulo 4: Capa de red

Capa de red: descripción general

plano de datos

plano de control

¿Qué hay dentro de un enrutador?

puertos de entrada, conmutación, puertos de salida

gestión de búfer, scheduling

IP: el Protocolo de Internet

formato de datagrama

Direccionamiento

Traducción de Direcciones de RedIPv6

Reenvío generalizado, SDN

Match+action

OpenFlow: match+action en operación

Middleboxes

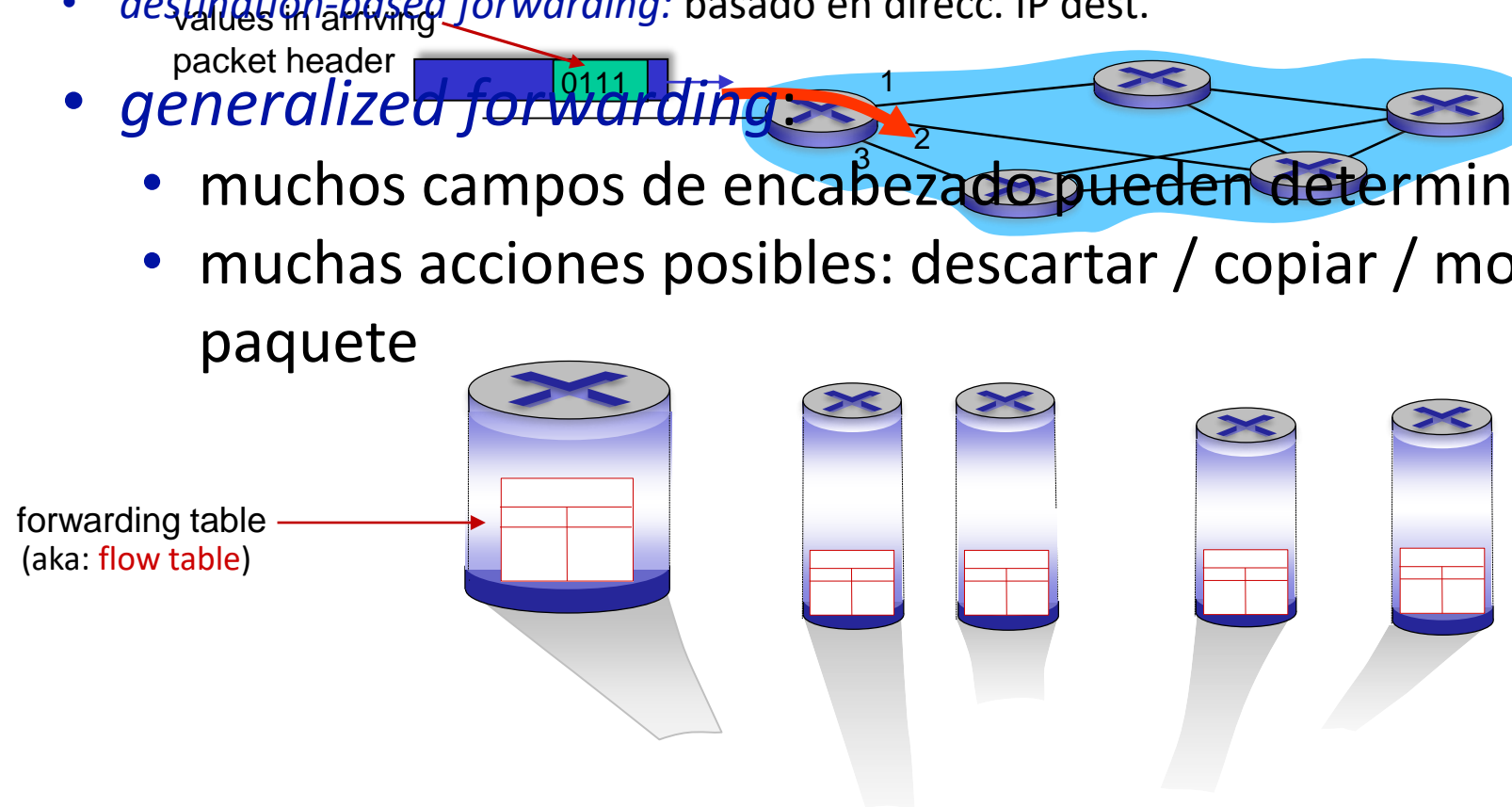
# Generalized forwarding: match plus action

Cada router contiene 1 tabla forwarding table (aka: flow table)

- “match plus action” concordancia de bits en pkt de llegada → acción

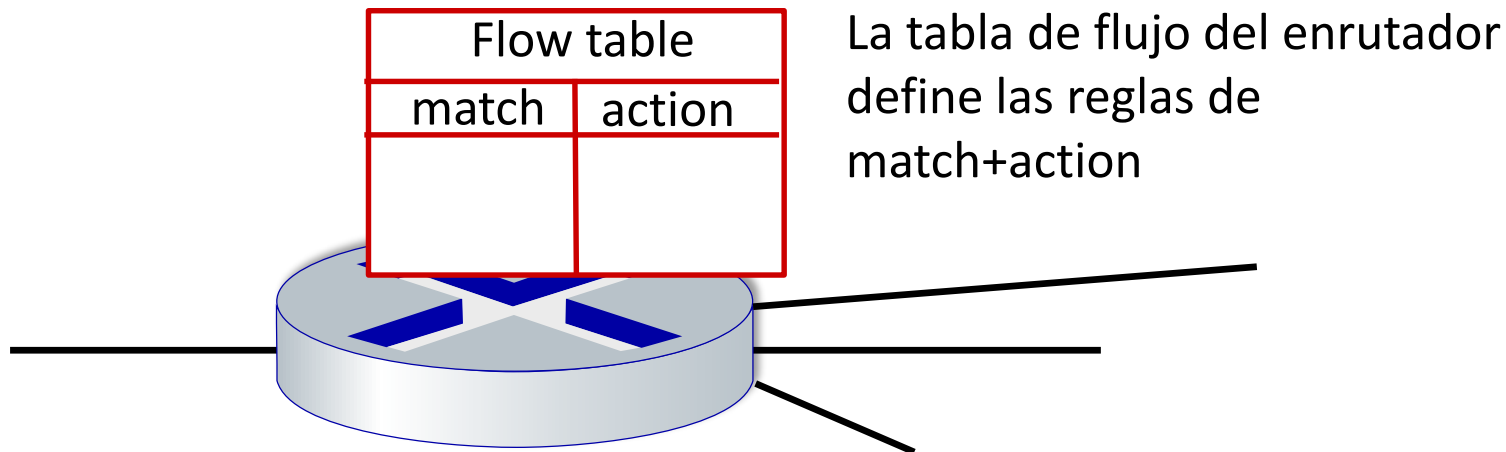
- *destination-based forwarding*: basado en direcc. IP dest.

- *generalized forwarding*:
  - muchos campos de encabezado pueden determinar la acción
  - muchas acciones posibles: descartar / copiar / modificar / registrar paquete



# Flow table

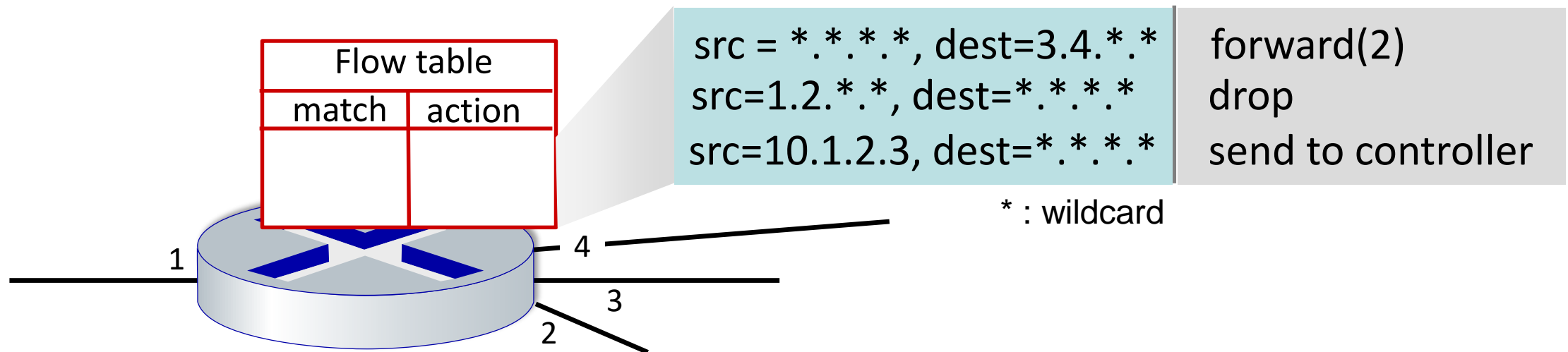
- **flow**: definido por los valores del campo de encabezado (en los campos de capa de enlace, red, transporte)
- **generalized forwarding**: reglas simples de manejo de paquetes
  - **match**: valores de patrón en los campos de encabezado del paquete
  - **actions**: para cada paquete coincidente: descartar, reenviar, modificar, paquete coincidente o enviar paquete coincidente al controlador
  - **priority**: eliminar la ambigüedad de los patrones superpuestos
  - **counters**: #bytes y #packets



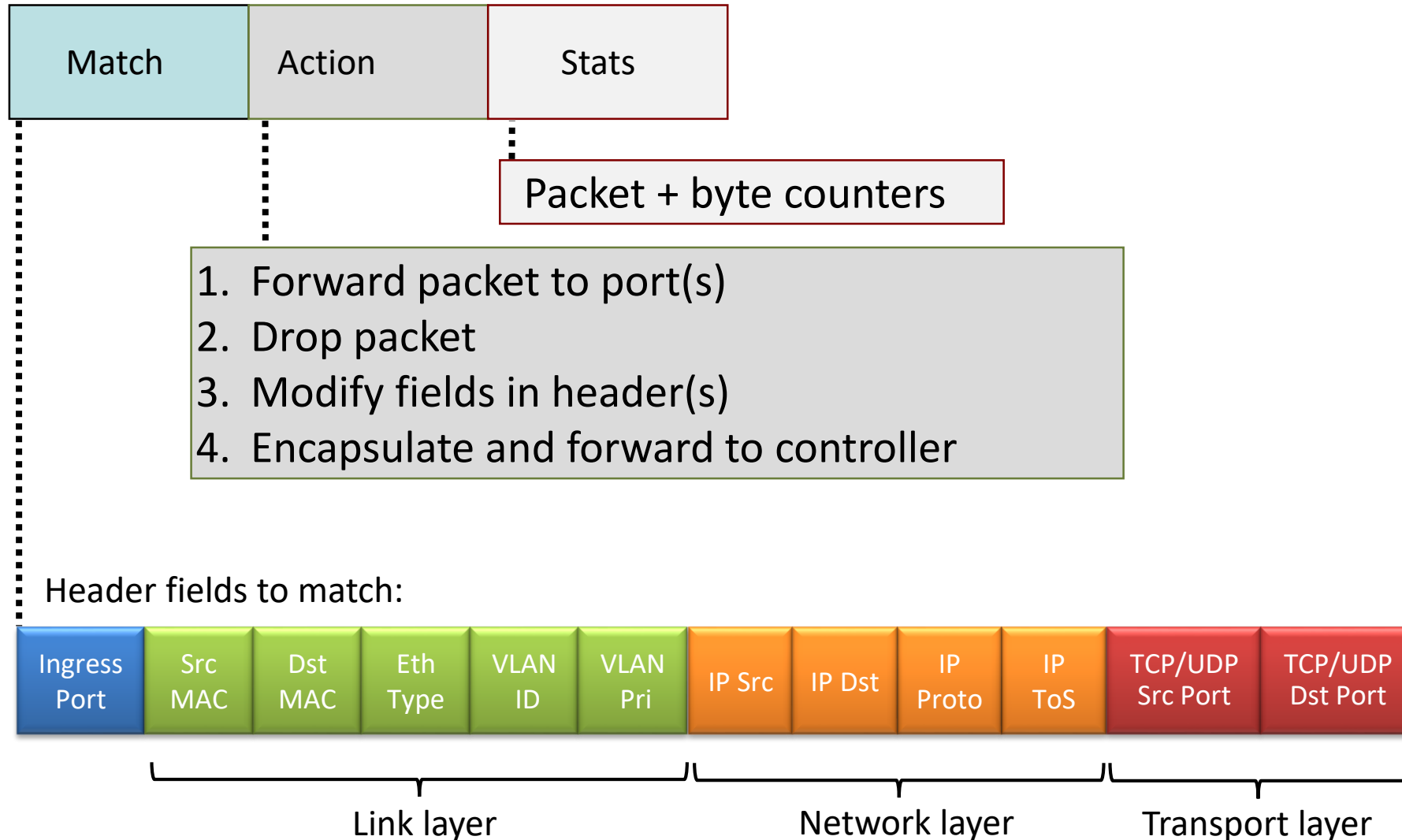


# Flow table

- **flow**: definido por los valores del campo de encabezado (en los campos de capa de enlace, red, transporte)
- **generalized forwarding**: reglas simples de manejo de paquetes
  - **match**: valores de patrón en los campos de encabezado del paquete
  - **actions**: para cada paquete coincidente: descartar, reenviar, modificar, paquete coincidente o enviar paquete coincidente al controlador
  - **priority**: eliminar la ambigüedad de los patrones superpuestos
  - **counters**: #bytes y #packets



# OpenFlow: flow table (entradas)



# OpenFlow: ejemplos

## Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	51.6.0.8	*	*	*	*	port6

Los datagramas IP destinados a la dirección IP 51.6.0.8 deben enviarse al puerto de salida 6 del enrutador

## Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	22	drop

Bloquear (no reenviar) todos los datagramas destinados al puerto TCP 22(ssh port #)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	drop

Bloquear (no reenviar) todos los datagramas enviados por el host 128.119.1.1

# OpenFlow: ejemplos

## Layer 2 destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	*	port3

Las tramas de capa 2 con dirección MAC de destino 22: A7: 23: 11: E1: 02 deben reenviarse al puerto de salida 3

# OpenFlow

- **match+action:** unifica diferentes tipos de dispositivos

## Router

- *match:* prefijo de IP de destino más largo
- *action:* reenviar a capa de enlace

## Switch

- *match:* dirección MAC de destino
- *action:* reenviar o permitir flujo

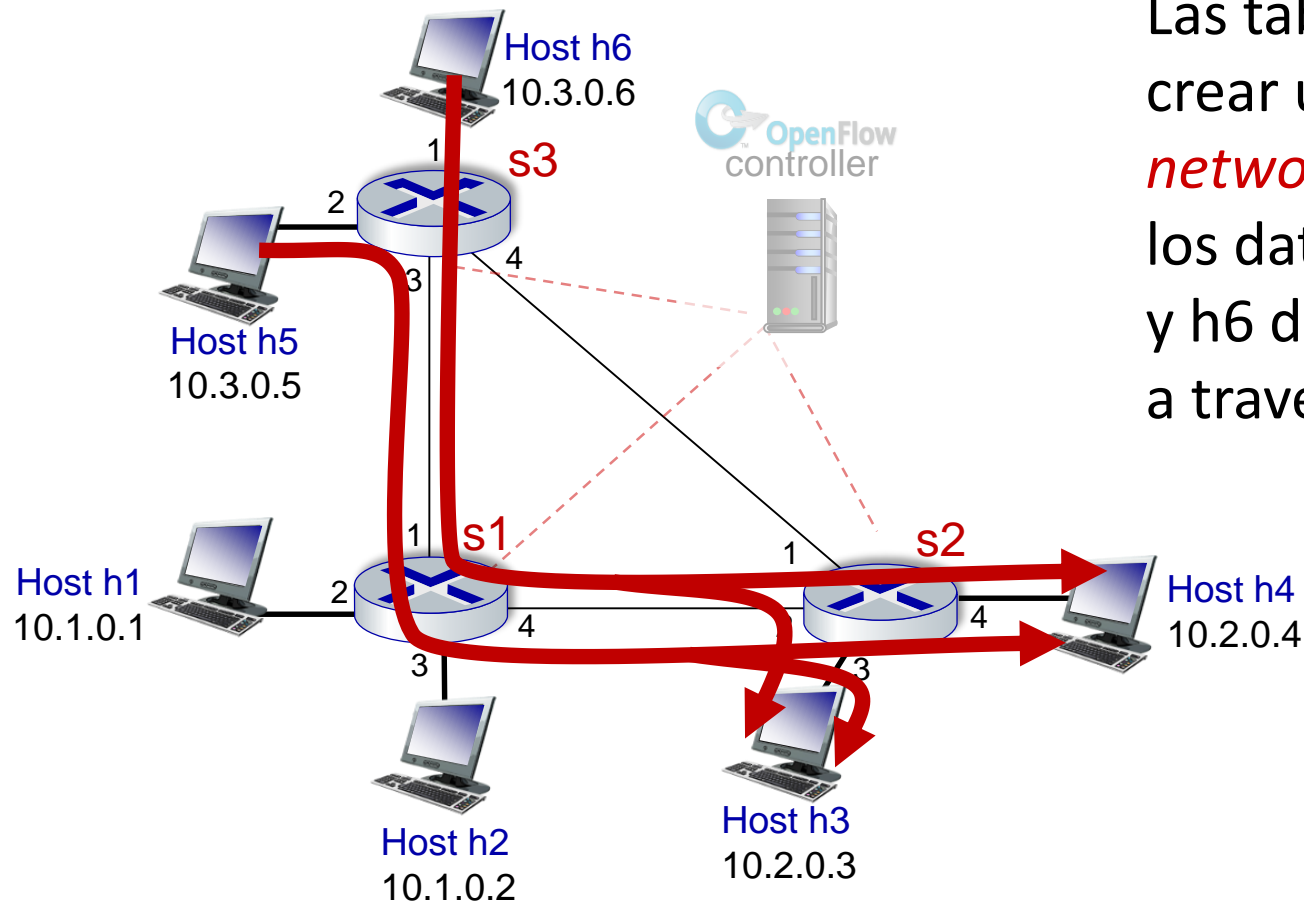
## Firewall

- *match:* Direcciones IP y números de puerto TCP / UDP
- *action:* permite o bloquea

## NAT

- *match:* Direcciones IP y números de puerto
- *action:* reescribir la dirección y el puerto

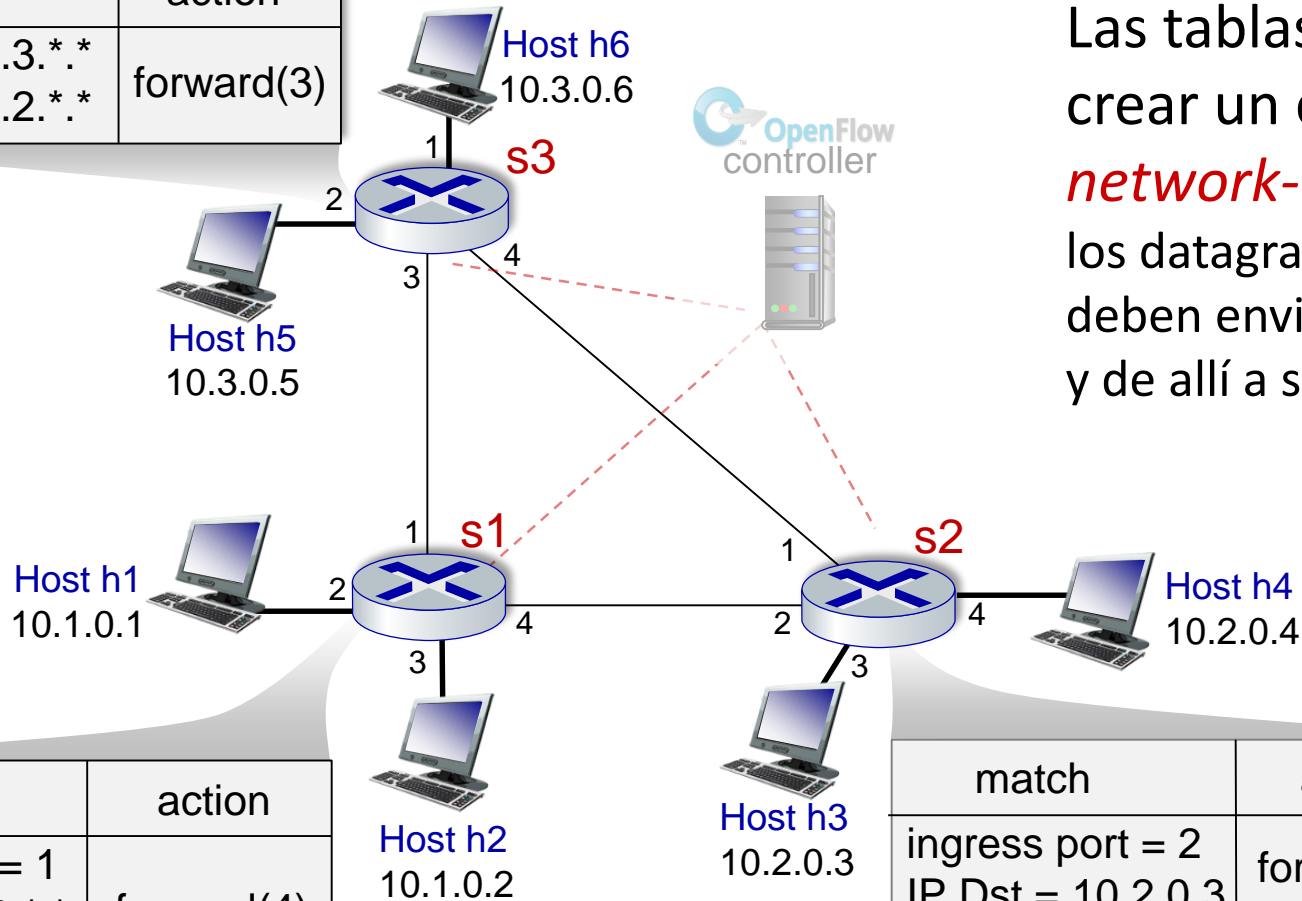
# OpenFlow (ejemplo)



Las tablas orquestadas pueden crear un comportamiento *network-wide*, por ejemplo: los datagramas de los hosts h5 y h6 deben enviarse a h3 o h4, a través de s1 y de allí a s2

# OpenFlow example

match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)



Las tablas orquestadas pueden crear un comportamiento *network-wide*, por ejemplo: los datagramas de los hosts h5 y h6 deben enviarse a h3 o h4, a través de s1 y de allí a s2

match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)

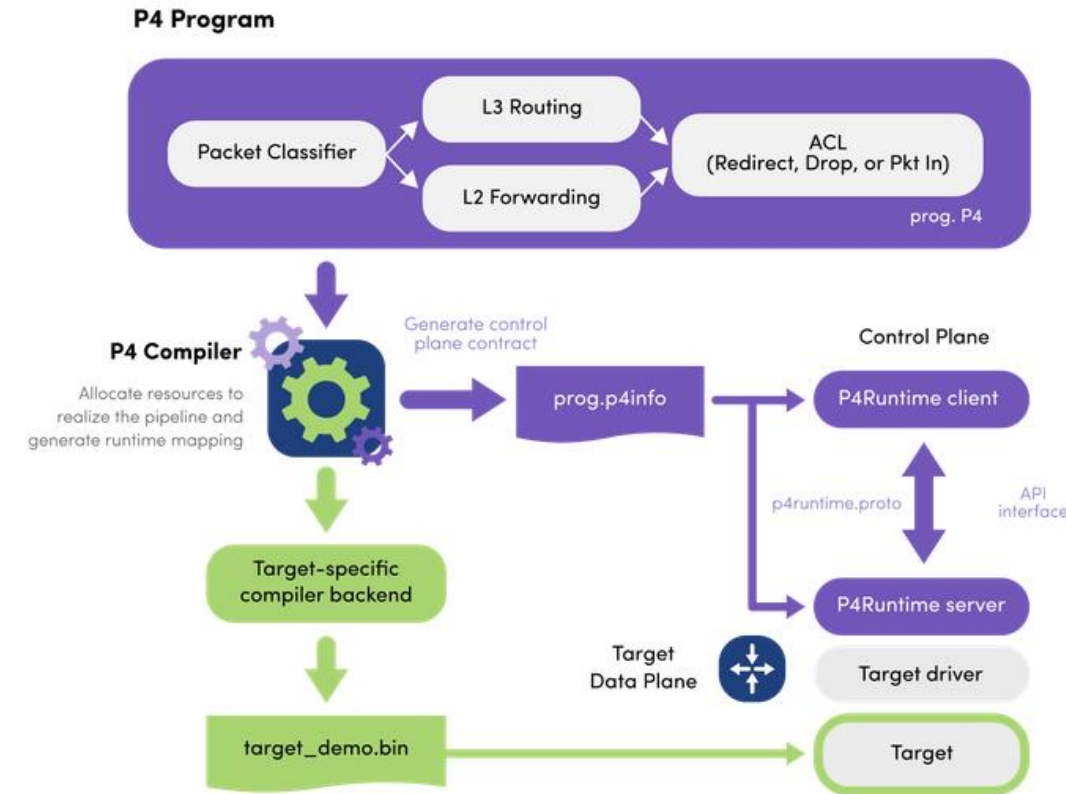


# Generalized forwarding: resumen

- “match plus action” abstraction: match bits in arriving packet header(s) in any layers, take action
  - matching over many fields (link-, network-, transport-layer)
  - local actions: drop, forward, modify, or send matched packet to controller
  - “program” *network-wide* behaviors
- simple form of “network programmability”
  - programmable, per-packet “processing”
  - *historical roots*: active networking
  - *today*: more generalized programming: P4 (see [p4.org](http://p4.org)).

# Generalized forwarding: resumen

- “match plus action” hacer coincidir los bits en los encabezados de los paquetes que llegan en cualquier capa y luego tomar acciones
- coincidencia en muchos campos (enlace, red, capa de transporte)
- acciones locales: descartar, reenviar, modificar o enviar paquetes coincidentes al controlador
- Comportamientos de "programa" en toda la red
- forma simple de "programabilidad de red"
- "procesamiento" programable por paquete
- raíces históricas: trabajo en red activo
- programación más generalizada



P4 (ver [www.p4.org](http://www.p4.org))

# Middleboxes

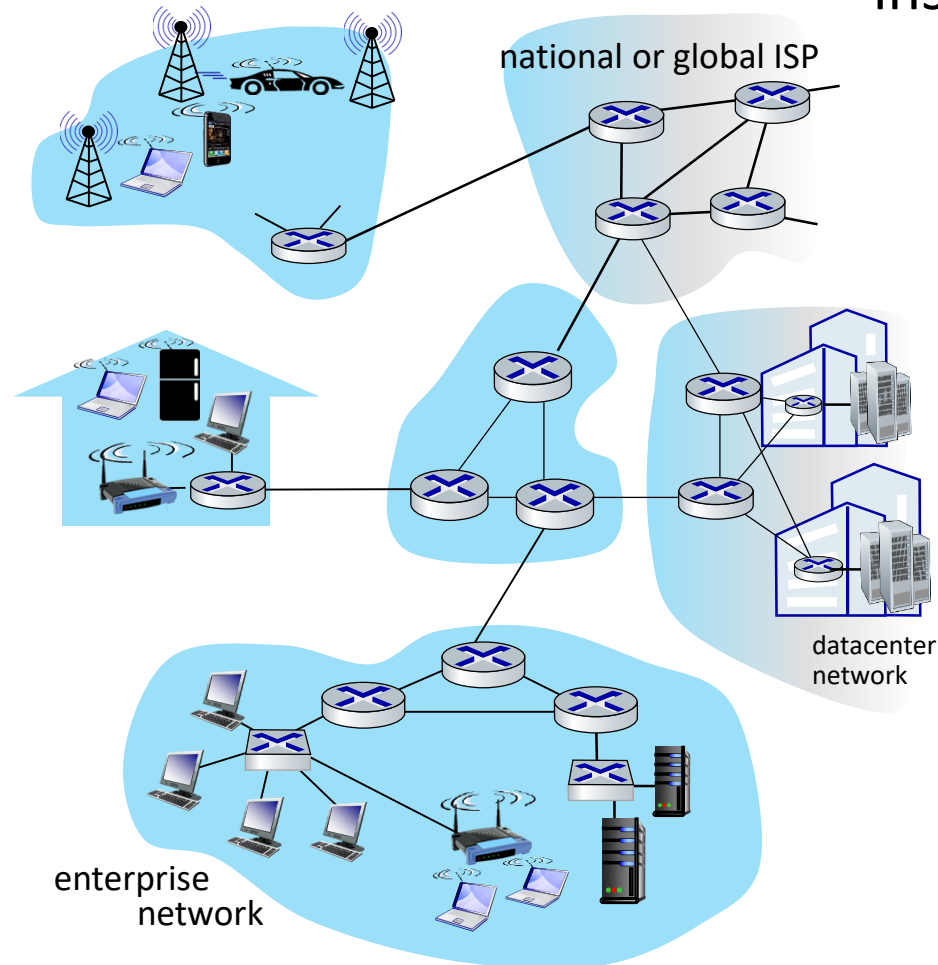
Middlebox (RFC 3234)

“cualquier caja intermedia que realice funciones además de las funciones estándar normales de un enrutador IP en la ruta de datos entre un host de origen y un host de destino”

# Middleboxes en todo lado!

**NAT:** home,  
cellular,  
institutional

**Application-specific:** service  
providers,  
institutional,  
CDN



**Firewalls, IDS:** corporate,  
institutional, service providers,  
ISPs

**Load balancers:**  
corporate, service  
provider, data center,  
mobile nets

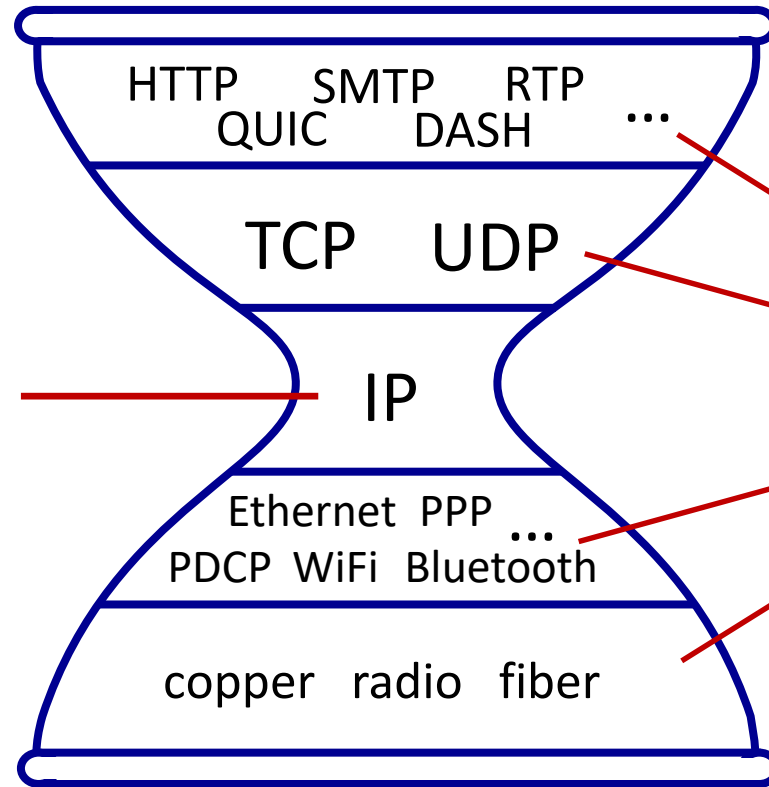
**Caches:** service  
provider, mobile, CDNs

# Middleboxes

- inicialmente: soluciones de hardware patentadas (cerradas)
- avanzar hacia el “whitebox” hardware que implementa una API abierta
- alejarse de las soluciones de hardware patentadas
- programmable local actions a través de match+action
- avanzar hacia la innovación / diferenciación en software
- SDN: (lógicamente) control centralizado y gestión de la configuración a menudo en la nube privada / pública
- network functions virtualization (NFV): servicios programables sobre redes de caja blanca, computación, almacenamiento

# El reloj de arena IP

- *un protocolo de capa de red: IP*
- *debe ser implementado por cada (miles de millones) de dispositivos conectados a Internet*

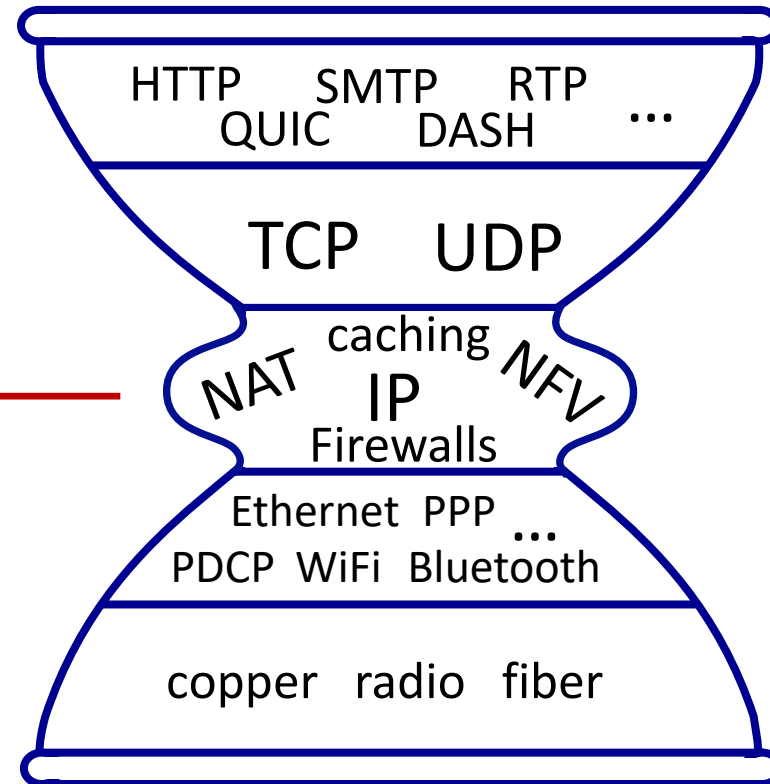


*muchos* protocolos en:  
physical, link,  
transport, and  
application  
layers

# El reloj de arena IP modificado

## Con gestión adicional

- middleboxes, operando internamente en la red





# Principios de arquitectura en Internet

## RFC 1958

“Muchos miembros de la comunidad de Internet argumentarían que no hay arquitectura, sino solo una tradición, que no fue escrita durante los primeros 25 años (o al menos no por la IAB). Sin embargo, en términos muy generales, la comunidad cree que **el objetivo es la conectividad, la herramienta es el Protocolo de Internet y la inteligencia está de extremo a extremo en lugar de ocultarse en la red**”.

Tres creencias fundamentales:

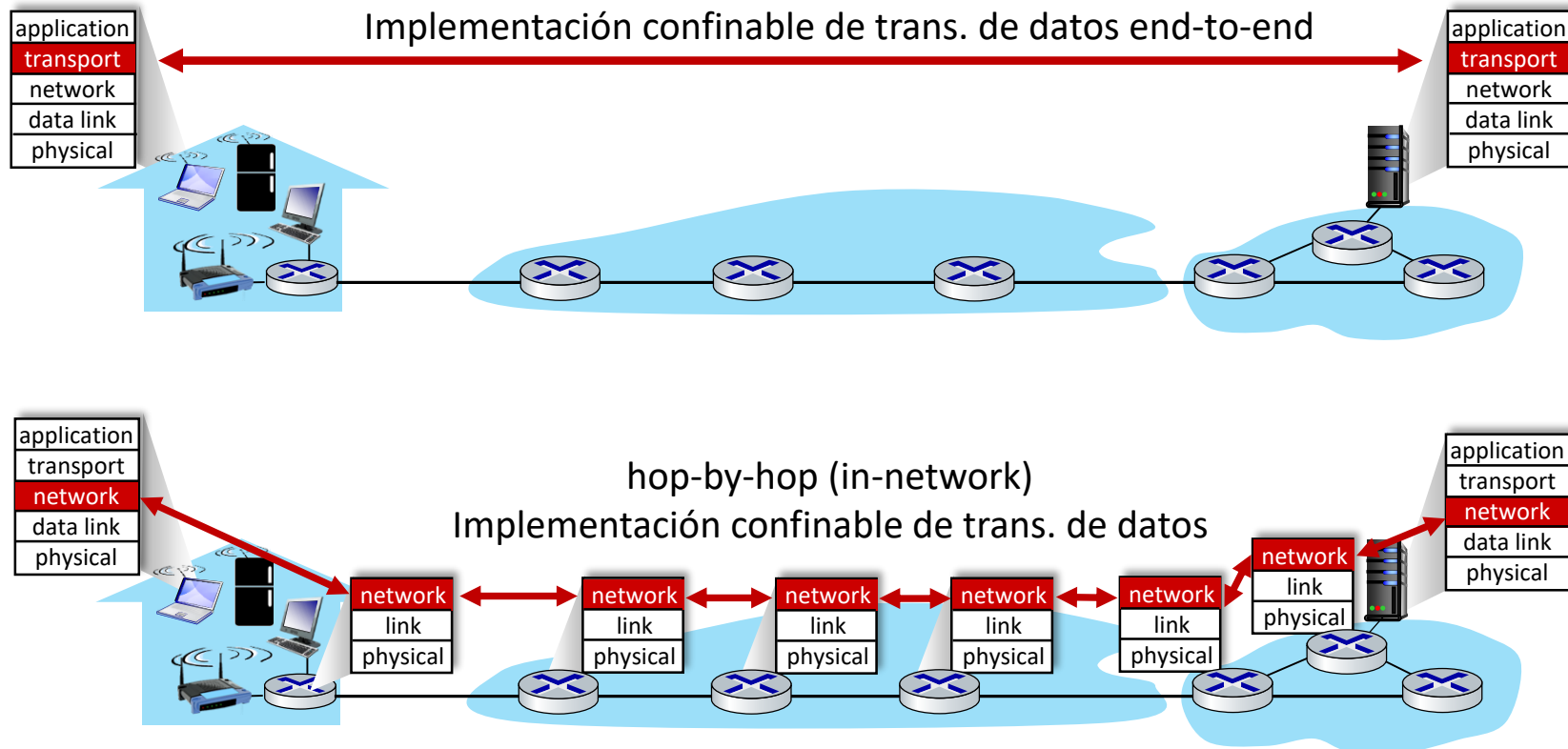
conectividad simple

Protocolo IP: esa cintura estrecha

inteligencia, complejidad en el borde de la red

# El argumento end-to-end

- Se puede implementar alguna funcionalidad de red (por ejemplo, transferencia de datos confiable, congestión) **en la capa de red**, o cerca **a la frontera de red**



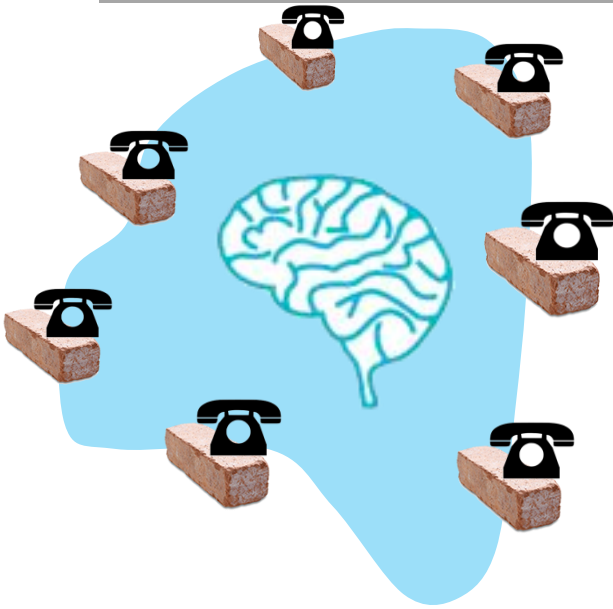
# El argumento end-to-end

- Se puede implementar alguna funcionalidad de red (por ejemplo, transferencia de datos confiable, congestión) **en la capa de red**, o cerca **a la frontera de red**

“La función en cuestión puede implementarse completa y correctamente solo con el conocimiento y la ayuda de la aplicación que se encuentra en los puntos finales del sistema de comunicación. Por lo tanto, no es posible proporcionar esa función cuestionada como una característica del propio sistema de comunicación. (A veces, una versión incompleta de la función proporcionada por el sistema de comunicación puede resultar útil para mejorar el rendimiento).

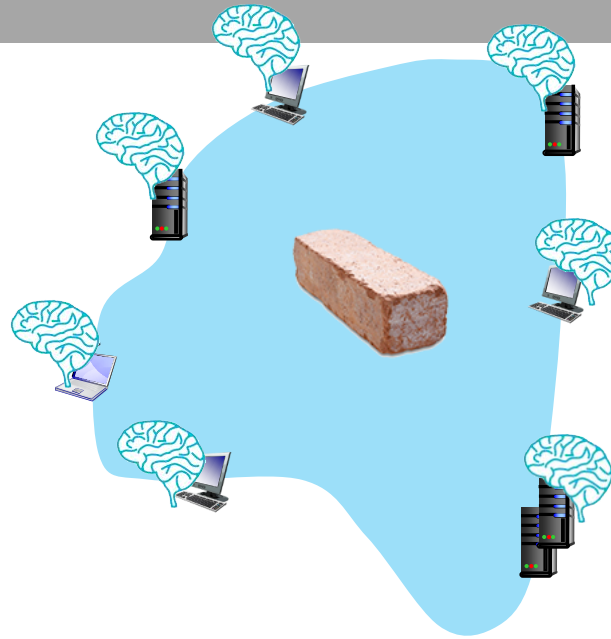
Llamamos a esta línea de razonamiento contra la implementación de funciones de bajo nivel el "argumento end-to-end".

# ¿Dónde está la inteligencia?



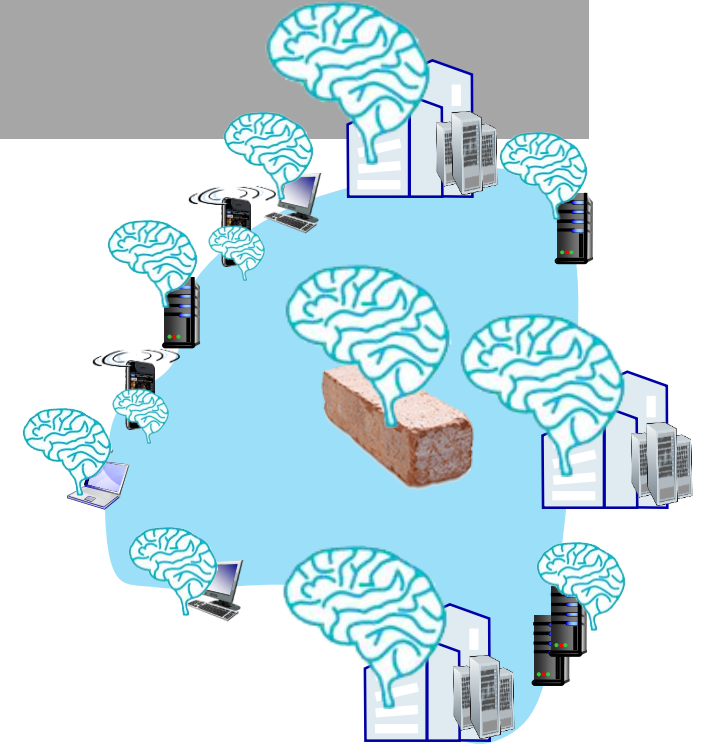
## 20<sup>th</sup> century phone net:

- inteligencia / computación en conmutadores de red



## Internet (pre-2005)

- inteligencia, computación en el borde



## Internet (post-2005)

- dispositivos de red programables  
inteligencia, informática, infraestructura masiva a nivel de aplicaciones en el borde





# Capítulo 4: Capa de red

## Capa de red: descripción general

- plano de datos

- plano de control

## ¿Qué hay dentro de un enrutador?

- puertos de entrada, conmutación, puertos de salida

- gestión de búfer, scheduling

## IP: el Protocolo de Internet

- formato de datagrama

- Direccionamiento

- Traducción de Direcciones de RedIPv6

## Reenvío generalizado, SDN

- Match+action

- OpenFlow: match+action en operación

## Middleboxes