

# Tema 7

# Redes y Sistemas Distribuidos

## Arquitectura TCP/IP (Transporte y Aplicación)



*Grado en Ingeniería Informática  
Escuela Politécnica Superior*



Universidad  
Francisco de Vitoria  
**UFV** Madrid

# Contenidos

## 1. Introducción a las redes de computadores

- Concepto de Red
- Tipos de redes
- Direccionamiento
- Latencia

## 2. Redes de área local

- Concepto y tipos de redes locales
- Medios de transmisión
- Técnicas de contención

## 3. Red Ethernet

- Características
- Protocolos
- Estándares
- Direcciones
- Codificación

## 4. Interconexión de redes

- Modos de interconexión
- Puentes
- *Spanning Tree*
- Switches

## 5. Red WLAN

- Topologías
- Espectro
- Nivel físico
- Protocolos
- Seguridad

## 6.1 Nivel del Red Internet

- Encaminamiento
- Fragmentación y reensamblaje

## 6.2 Direccionamiento IP

## 7. Arquitectura TCP/IP

- Capa de transporte
- Capa de aplicación

## 8. Sistemas distribuidos

- Concepto
- Arquitectura cliente-servidor
- Arquitectura P2P

# Contenidos

## Arquitectura TCP/IP

- **Capa de transporte**
  - Funcionalidades
  - Puertos
  - Protocolos
- **Capa de aplicación**
  - RTP
  - NAT
  - DHCP



# Contenidos

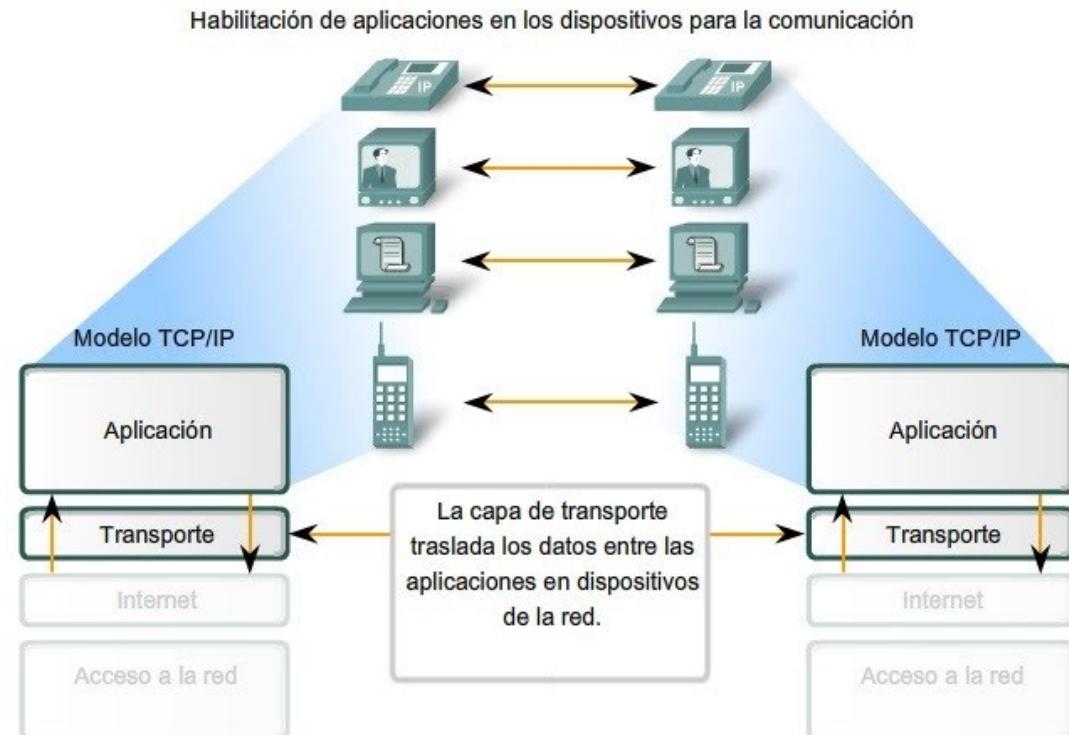
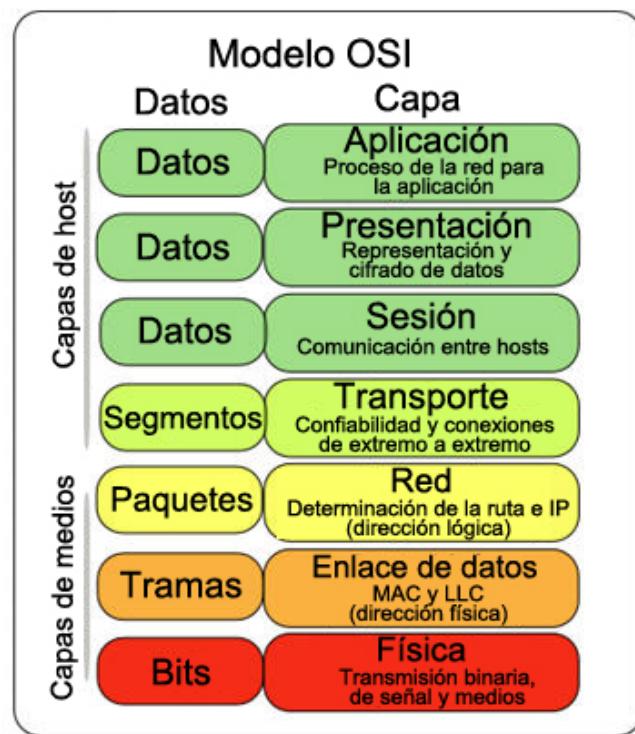
## Arquitectura TCP/IP

- **Capa de transporte**
  - Funcionalidades
  - Puertos
  - Protocolos
  
- Capa de aplicación
  - RTP
  - NAT
  - DHCP

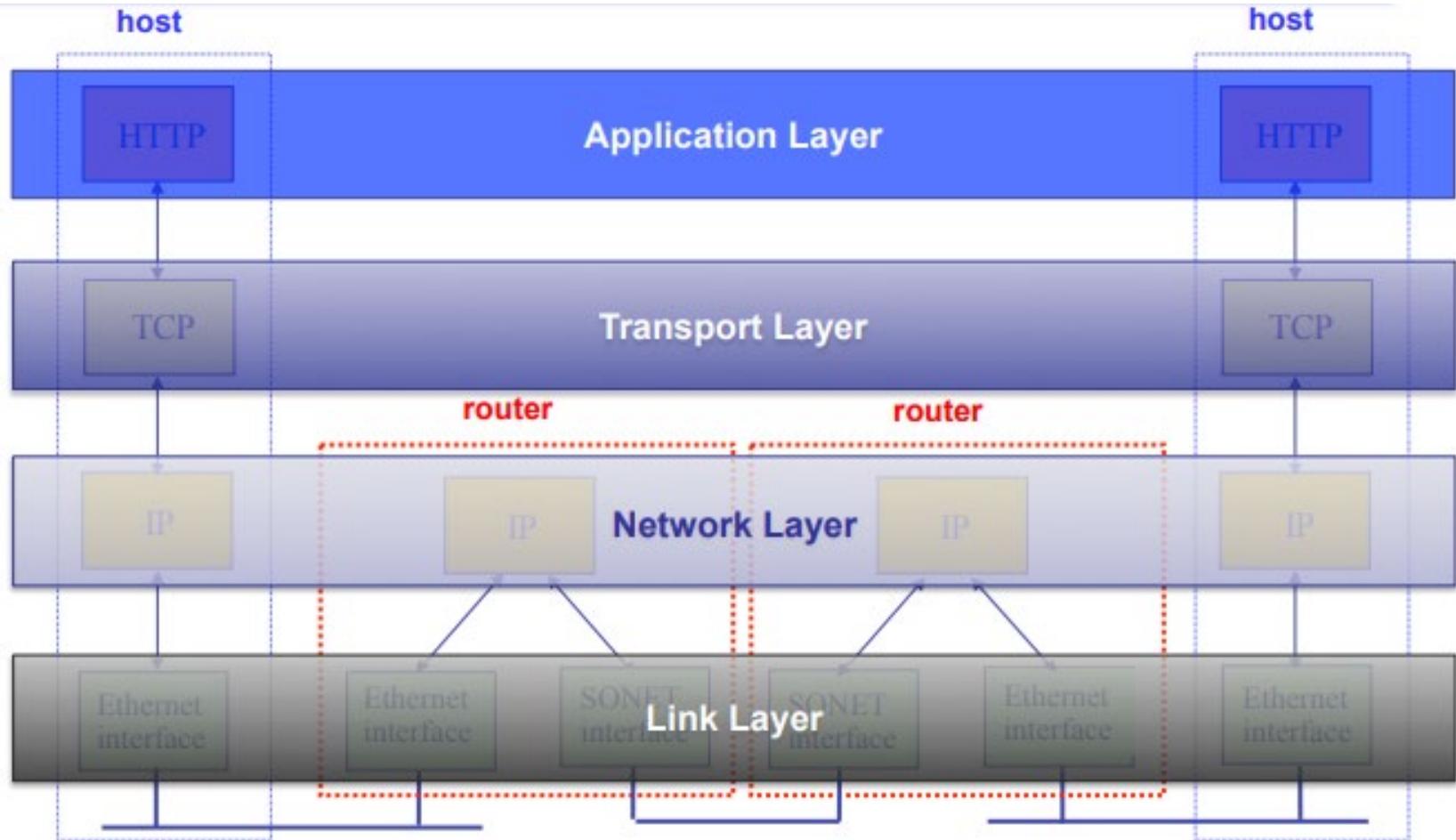


# La capa de transporte

**Capa de transporte** es el 4º nivel del modelo TCP/IP, y está encargado de la transferencia libre de errores de los datos entre el emisor y el receptor, aunque no estén directamente conectados, así como de mantener el flujo de la red.

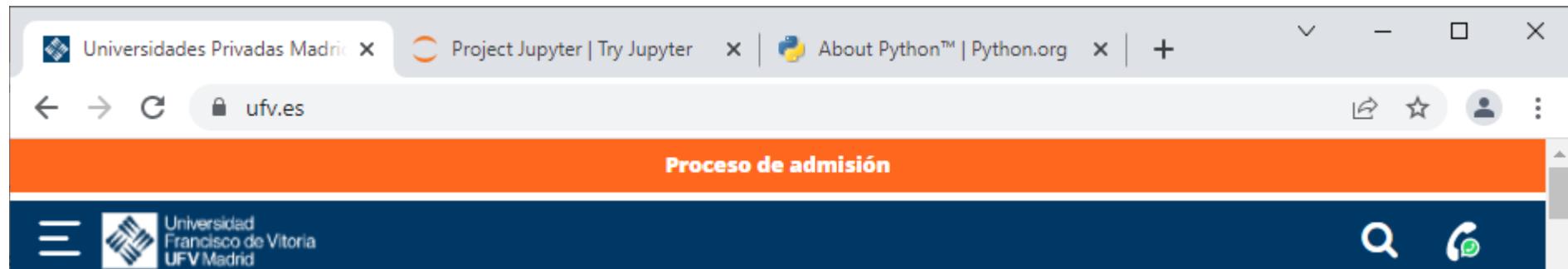


# La capa de transporte



# La capa de transporte

- La capa de **transporte** debe ser capaz **de separar y administrar** varias **aplicaciones**, que a su vez pueden abrir muchas **sesiones** individuales.
- Por ejemplo, al abrir varias pestañas en el navegador web para ver varias páginas web, el protocolo *http* de la capa de aplicación crea una sesión independiente para cada pestaña.



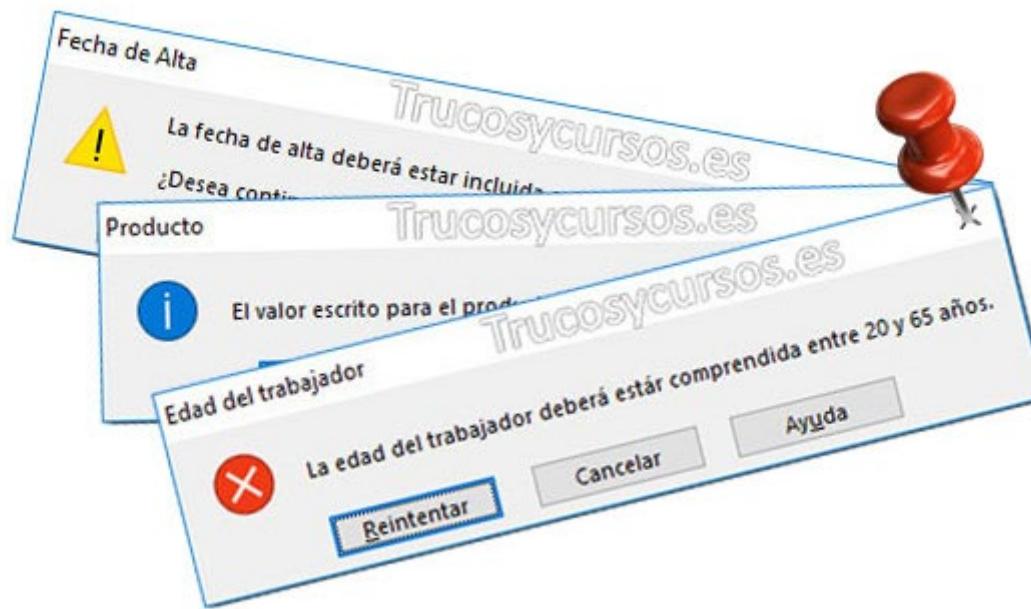
# La capa de transporte

- Al mismo tiempo que se visualizan varias pestañas, también puede enviar correo electrónico, mensajes instantáneos y descargar archivos, etc.
- Cada una de estas actividades requiere establecer el **acceso simultáneo a la red a través de protocolos de transporte TCP / UDP.**



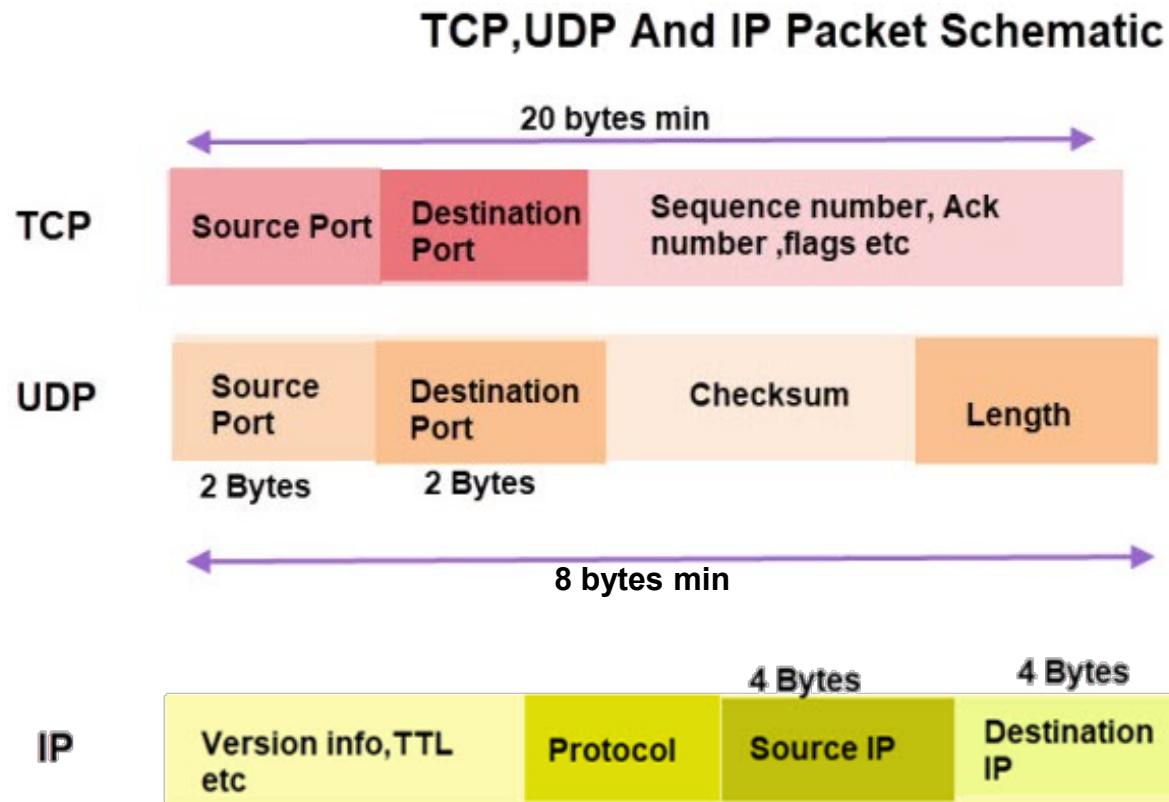
# La capa de transporte

- Los protocolos de la capa de transporte deben realizar un seguimiento de esta actividad y asegúrese de que **los datos recibidos se dirigen a la capa de aplicación correcta**, de lo contrario, por ejemplo, los datos de la página web pueden dirigirse a una aplicación de correo electrónico.

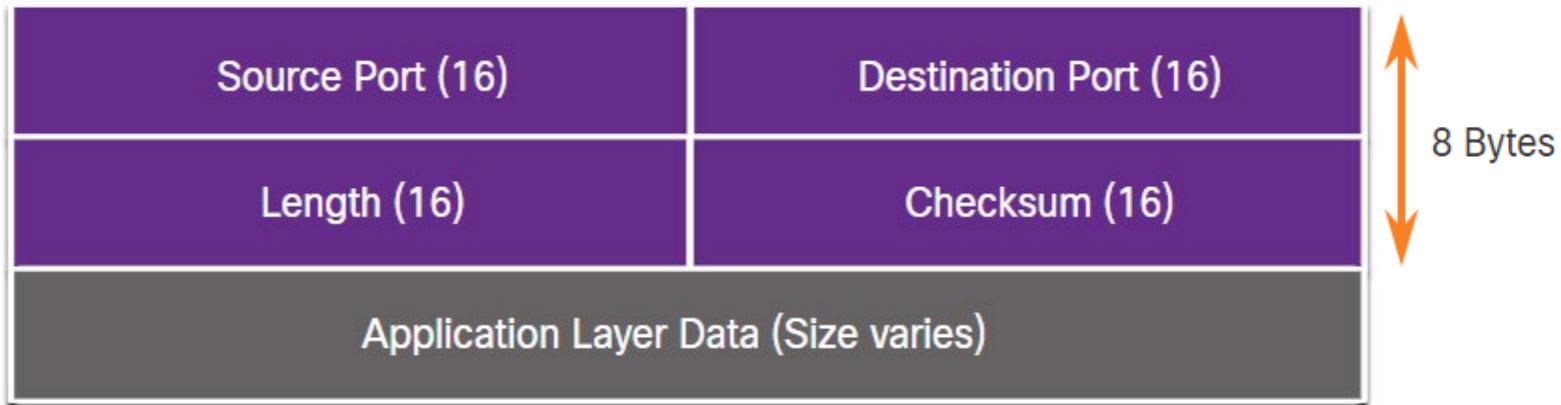


# La capa de transporte

- TCP y UDP administran estos múltiples procesos mediante el uso de **números de puerto únicos** contenidos en el campo de encabezado.

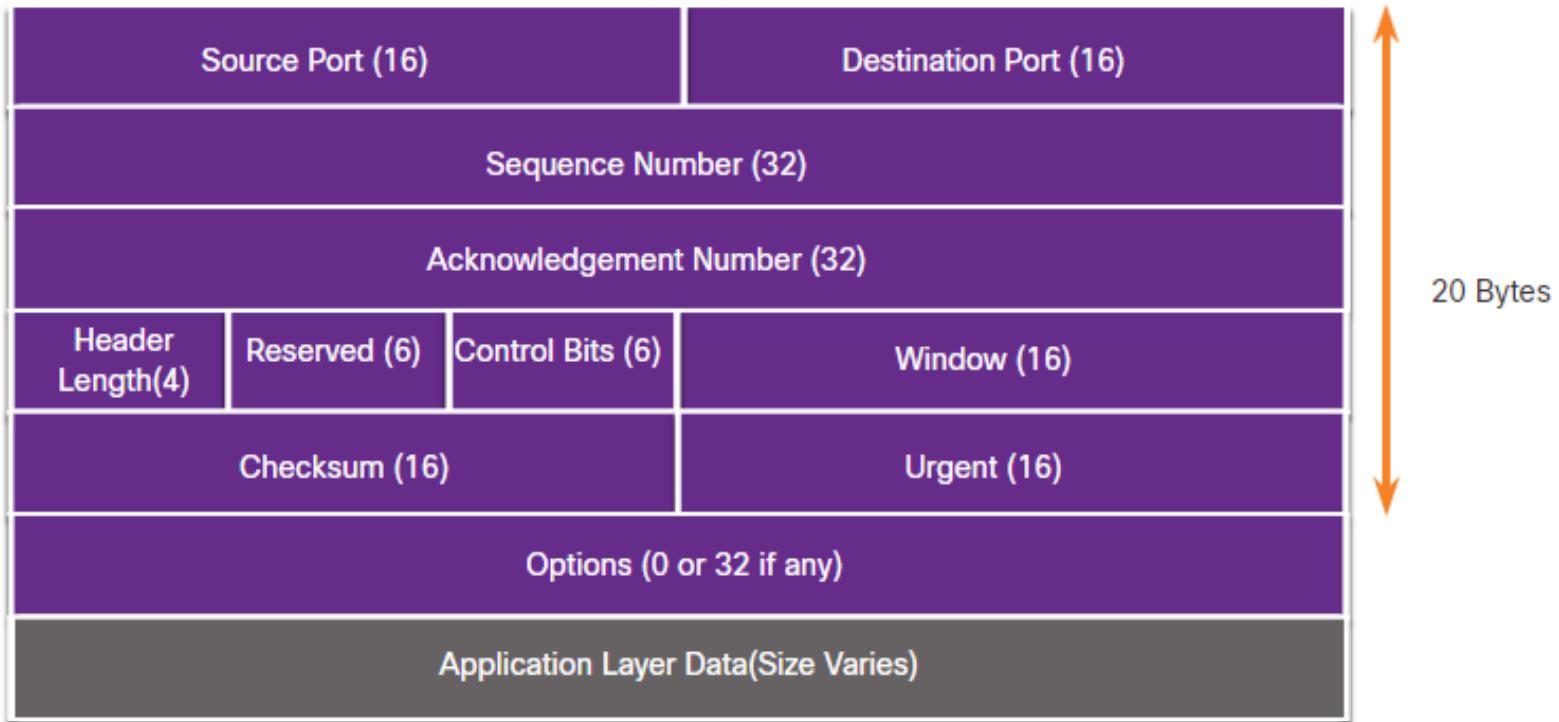


# La capa de transporte: Cabecera UDP



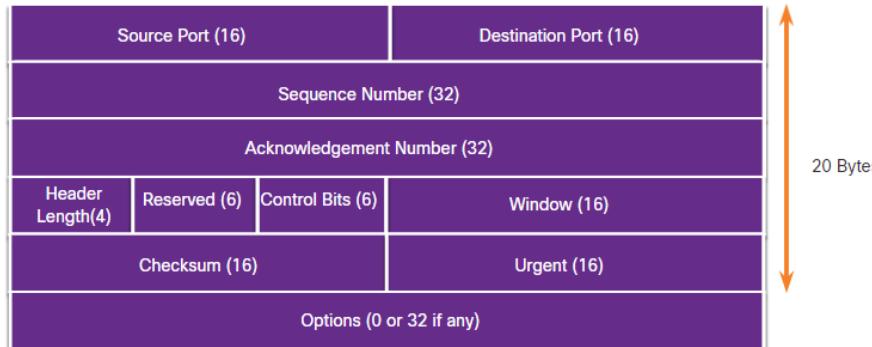
Campo UDP	Descripción
Source port	(16 bits) Identifica a la aplicación de origen por número de puerto.
Destination port	(16 bits) Identifica la aplicación de destino por número de puerto.
Length	(16 bits) Indica la longitud del encabezado y del datagrama UDP.
Checksum	(16 bits) Para comprobación de errores encabezado y datos del datagrama.

# La capa de transporte: Cabecera TCP



# La capa de transporte: Cabecera TCP

Campo TCP	Descripción
<b>Source port</b>	(16 bits) Identifica a la aplicación de origen por número de puerto.
<b>Destination port</b>	(16 bits) Identifica la aplicación de destino por número de puerto.
<b>Sequence number</b>	(32 bits) Para reensamblar datos.
<b>ACK number</b>	(32 bits) Para indicar que se han recibido datos y el siguiente byte esperado.
<b>Header Length</b>	(4 bits) conocido como «desplazamiento de datos» que indica la longitud del encabezado del segmento TCP.
<b>Reserved</b>	(6 bits) Reservados para uso futuro.
<b>Control bits</b>	(6 bits) Incluye códigos de bit, o indicadores, que indican el propósito y la función del segmento TCP.
<b>Window</b>	(16 bits) Para indicar el número de bytes que se pueden aceptar
<b>Checksum</b>	(16 bits) Para comprobación de errores del encabezado y datos del segmento.
<b>Urgent</b>	(16 bits) Indica si los datos contenidos son urgentes.



# La capa de transporte

- Los protocolos de la capa de **aplicación** tienen requisitos diferentes de la capa de transporte
- Algunos requieren una **entrega fiable** de datos (por ejemplo, **HTTP, FTP**), mientras que otros requieren un servicio de **bajo retardo** (por ejemplo, **DNS**).
- Los diferentes protocolos de capa de aplicación están diseñados para funcionar con **TCP o UDP** en función de estos principales requisitos.

## Arquitectura TCP/IP

- **Capa de transporte**
  - Funcionalidades
  - **Puertos**
  - Protocolos
- Capa de aplicación
  - RTP
  - NAT
  - DHCP

# La capa de transporte

- El enlace entre los **protocolos** de la capa de **aplicación** y **transporte** se basa en los **puertos** seleccionados para admitir las sesiones individuales admitidas por la capa de transporte.
- Hay **65.535** números de puerto disponibles, y estos se dividen en tres rangos:
  - **Puertos conocidos (0 a 1.023)**
  - **Puertos registrados (1.024 a 49.151)**
  - **Puertos efímeros (49.152 a 65.535)**

# La capa de transporte

- **Puertos conocidos (0 a 1023).** Están **reservados**, y son comúnmente utilizados por **HTTP(80), SMTP, POP3, FTP(21), DNS**, etc.

- Reservados por la ***Internet Assigned Numbers Authority (IANA)***

**<https://www.iana.org/>**

<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

- Debido a que están reservadas, las aplicaciones cliente se pueden programar para solicitar una conexión a un puerto específico y su servicio de capa de transporte asociado (TCP o UDP).

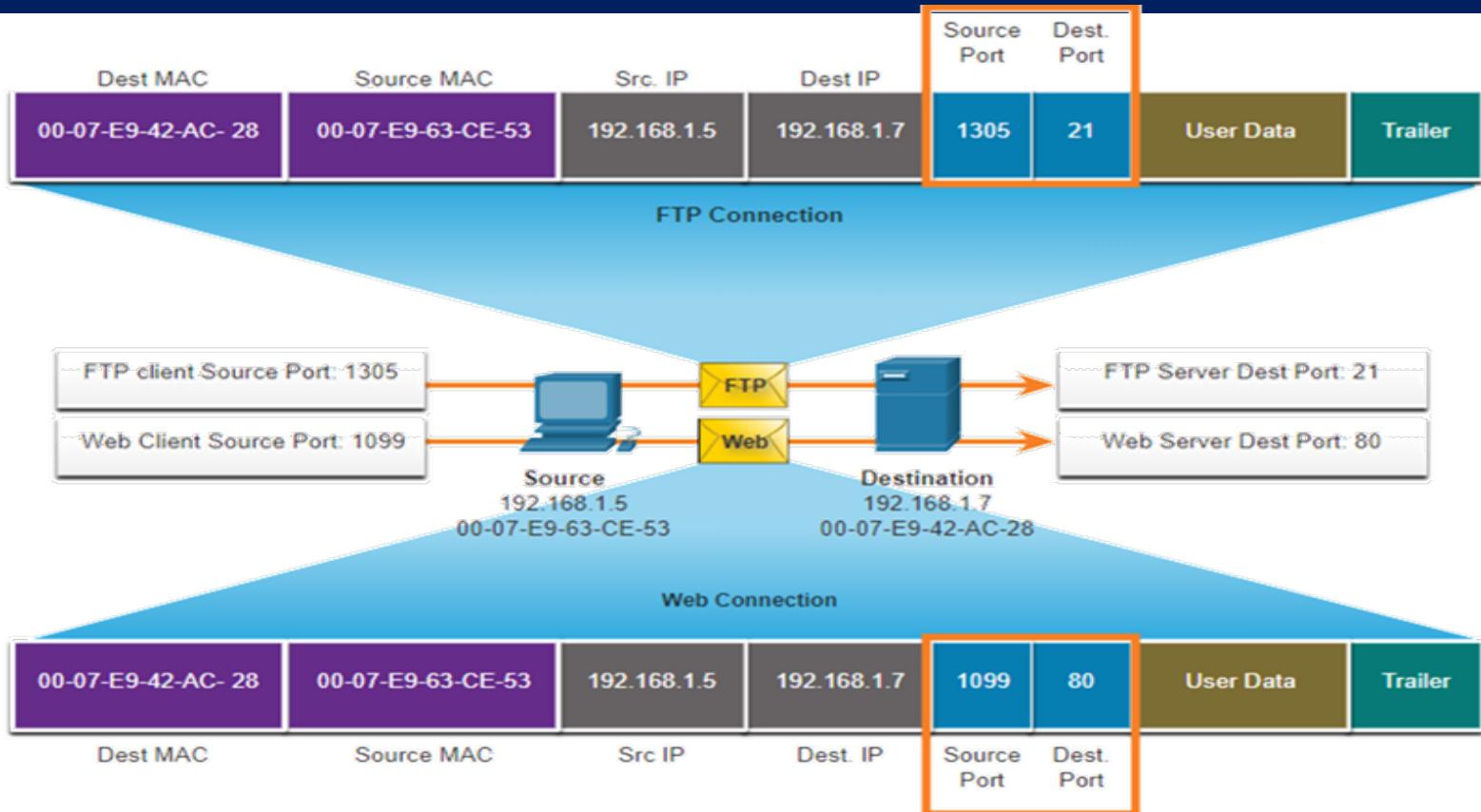
# La capa de transporte

- **Puertos registrados (1024 a 49151)**
  - Estos pueden registrarse para **servicios con la IANA** y deben tratarse como **semi-reservados**.
  - Los programas escritos por el usuario **no deben utilizar** estos puertos.

# La capa de transporte

- **Puertos efímeros (49152 a 65535)**
  - Estos son **libremente utilizados para programas cliente** y usted es libre de usarlos en programas cliente.
  - Ejemplo:
    - Cuando un navegador web se conecta a un servidor web, el navegador se asignará un puerto en este rango.

# Pares de sockets



- **Puertos de origen y destino** se colocan en segmento (TCP)
- Segmentos se encapsulan dentro de paquete IP
- **Socket:** Combinación de **IP + puerto** y permiten:
  - Que los diversos procesos que se ejecutan en un cliente se distingan entre sí.
  - La diferenciación de diferentes conexiones a un proceso de servidor.

# La capa de transporte

¿Cómo descubrir los puertos libres y usados por tu ordenador?  
→ **NETSTAT**

Microsoft Windows [Versión 6.1.7601] Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.			
C:\Users\Alumno> netstat			
Conexiones activas			
Proto	Dirección local	Dirección remota	Estado
TCP	10.228.35.199:49219	213-0-88-085:8080	ESTABLISHED
TCP	10.228.35.199:50255	213-0-88-085:8080	ESTABLISHED
TCP	10.228.35.199:50473	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50479	213-0-88-085:8080	ESTABLISHED
TCP	10.228.35.199:50491	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50493	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50494	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50496	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50502	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50505	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50520	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50524	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50528	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50529	213-0-88-085:8080	ESTABLISHED
TCP	10.228.35.199:50530	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50532	213-0-88-085:8080	ESTABLISHED
TCP	10.228.35.199:50533	213-0-88-085:8080	ESTABLISHED
TCP	10.228.35.199:50534	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50535	213-0-88-085:8080	TIME_WAIT
TCP	10.228.35.199:50536	213-0-88-085:8080	ESTABLISHED
TCP	10.228.35.199:50538	213-0-88-085:8080	ESTABLISHED



# La capa de transporte

¿Cómo descubrir los puertos libres y usados en ordenadores remotos? → **NMAP (Network Mapper)**

*“Nmap (“Network Mapper”) is an open-source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts”.*

Información y descarga:

<https://nmap.org/>



Excelente guía / tutorial (*Linux*):

<https://www.cyberciti.biz/networking/nmap-command-examples-tutorials/>

# La capa de transporte

TCP/UDP Port Numbers			
7 Echo	554 RTSP	2745 Bagle.H	6891-6901 Windows Live
19 Chargen	546-547 DHCPv6	2967 Symantec AV	6970 Quicktime
20-21 FTP	560 rmonitor	3050 Interbase DB	7212 GhostSurf
22 SSH/SCP	563 NNTP over SSL	3074 XBOX Live	7648-7649 CU-SeeMe
23 Telnet	587 SMTP	3124 HTTP Proxy	8000 Internet Radio
25 SMTP	591 FileMaker	3127 MyDoom	8080 HTTP Proxy
42 WINS Replication	593 Microsoft DCOM	3128 HTTP Proxy	8086-8087 Kaspersky AV
43 WHOIS	631 Internet Printing	3222 GLBP	8118 Privoxy
49 TACACS	636 LDAP over SSL	3260 iSCSI Target	8200 VMware Server
53 DNS	639 MSDP (PIM)	3306 MySQL	8500 Adobe ColdFusion
67-68 DHCP/BOOTP	646 LDP (MPLS)	3389 Terminal Server	8767 TeamSpeak
69 TFTP	691 MS Exchange	3689 iTunes	8866 Bagle.B
70 Gopher	860 iSCSI	3690 Subversion	9100 HP JetDirect
79 Finger	873 rsync	3724 World of Warcraft	9101-9103 Bacula
80 HTTP	902 VMware Server	3784-3785 Ventrilo	9119 MXit
88 Kerberos	989-990 FTP over SSL	4333 mSQL	9800 WebDAV
102 MS Exchange	993 IMAP4 over SSL	4444 Blaster	9898 Dabber
110 POP3	995 POP3 over SSL	4664 Google Desktop	9988 Rbot/Spybot
113 Ident	1025 Microsoft RPC	4672 eMule	9999 Urchin
119 NNTP (Usenet)	1026-1029 Windows Messenger	4899 Radmin	10000 Webmin
123 NTP	1080 SOCKS Proxy	5000 UPnP	10000 BackupExec
135 Microsoft RPC	1080 MyDoom	5001 Slingbox	10113-10116 NetIQ
137-139 NetBIOS	1194 OpenVPN	5001 Iperf	11371 OpenPGP



# La capa de transporte

143 IMAP4	1214 Kazaa	5004-5005 RTP	12035-12036 Second Life
161-162 SNMP	1241 Nessus	5050 Yahoo! Messenger	12345 NetBus
177 XDMCP	1311 Dell OpenManage	5060 SIP	13720-13721 NetBackup
179 BGP	1337 WASTE	5190 AIM/ICQ	14567 Battlefield
201 AppleTalk	1433-1434 Microsoft SQL	5222-5223 XMPP/Jabber	15118 Dipnet/Oddbob
264 BGMP	1512 WINS	5432 PostgreSQL	19226 AdminSecure
318 TSP	1589 Cisco VQP	5500 VNC Server	19638 Ensim
381-383 HP Openview	1701 L2TP	5554 Sasser	20000 Usermin
389 LDAP	1723 MS PPTP	5631-5632 pcAnywhere	24800 Synergy
411-412 Direct Connect	1725 Steam	5800 VNC over HTTP	25999 Xfire
443 HTTP over SSL	1741 CiscoWorks 2000	5900+ VNC Server	27015 Half-Life
445 Microsoft DS	1755 MS Media Server	6000-6001 X11	27374 Sub7
464 Kerberos	1812-1813 RADIUS	6112 Battle.net	28960 Call of Duty
465 SMTP over SSL	1863 MSN	6129 DameWare	31337 Back Orifice
497 Retrospect	1985 Cisco HSRP	6257 WinMX	33434+ traceroute
500 ISAKMP	2000 Cisco SCCP	6346-6347 Gnutella	Legend
512 rexec	2002 Cisco ACS	6500 GameSpy Arcade	Chat
513 rlogin	2049 NFS	6566 SANE	Encrypted
514 syslog	2082-2083 cPanel	6588 AnalogX	Gaming
515 LPD/LPR	2100 Oracle XDB	6665-6669 IRC	Malicious
520 RIP	2222 DirectAdmin	6679/6697 IRC over SSL	Peer to Peer
521 RIPng (IPv6)	2302 Halo	6699 Napster	Streaming
540 UUCP	2483-2484 Oracle DB	6881-6999 BitTorrent	

IANA port assignments published at <http://www.iana.org/assignments/port-numbers>



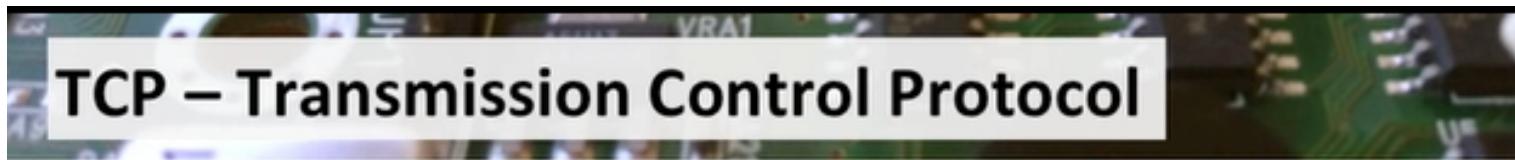
# La capa de transporte

Port number	Process name	Protocol used	Description
20	FTP-DATA	TCP	File transfer—data
21	FTP	TCP	File transfer—control
22	SSH	TCP	Secure Shell
23	TELNET	TCP	Telnet
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP and UDP	Domain Name System
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP and UDP	Hypertext Transfer Protocol
110	POP3	TCP	Post Office Protocol 3
123	NTP	TCP	Network Time Protocol
143	IMAP	TCP	Internet Message Access Protocol
443	HTTPS	TCP	Secure implementation of HTTP 

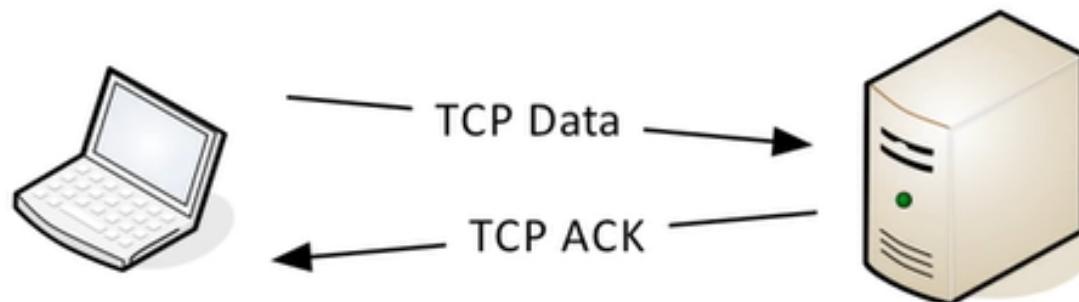


## Arquitectura TCP/IP

- **Capa de transporte**
  - Funcionalidades
  - Puertos
  - **Protocolos**
  
- Capa de aplicación
  - RTP
  - NAT
  - DHCP

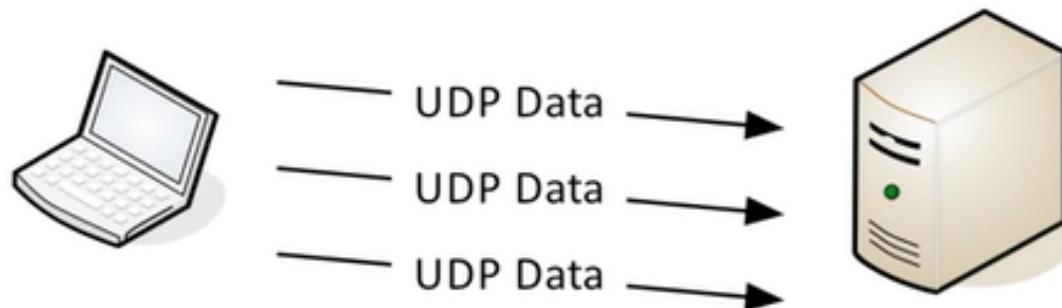


- Connection-oriented
- Reliable delivery
- Can manage out-of-order messages or retransmissions
- Loads and unloads the moving truck
  - Checks for out-of-order or missing cargo

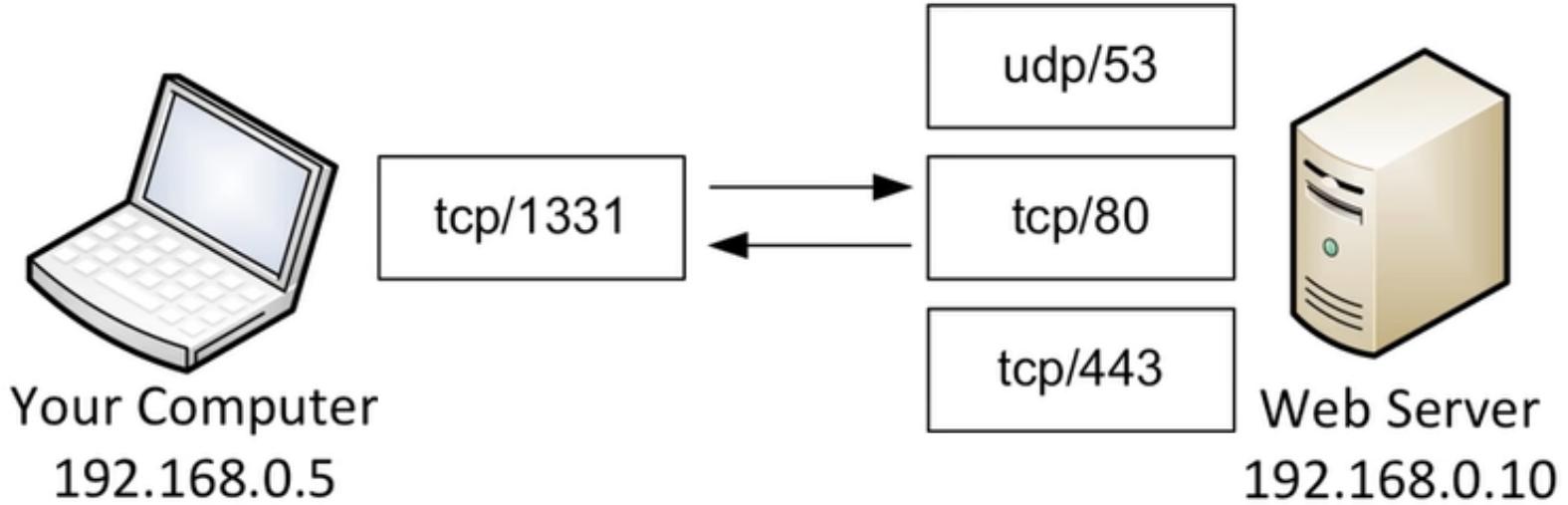


## UDP – User Datagram Protocol

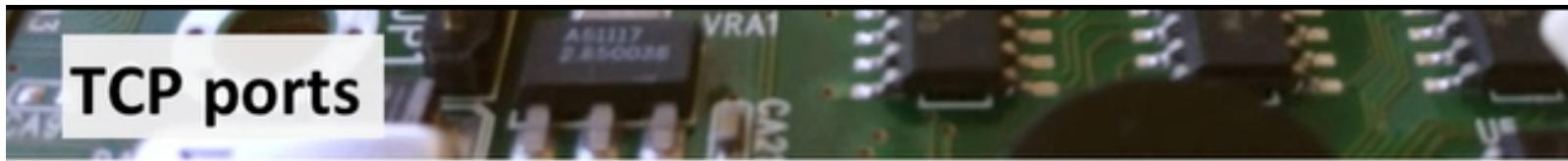
- Connectionless
- Unreliable
- No reordering of data or retransmissions
- Loads and unloads the moving truck
  - Doesn't care about out-of-order cargo or missing cargo



# Conexiones de red



# Principales puertos TCP

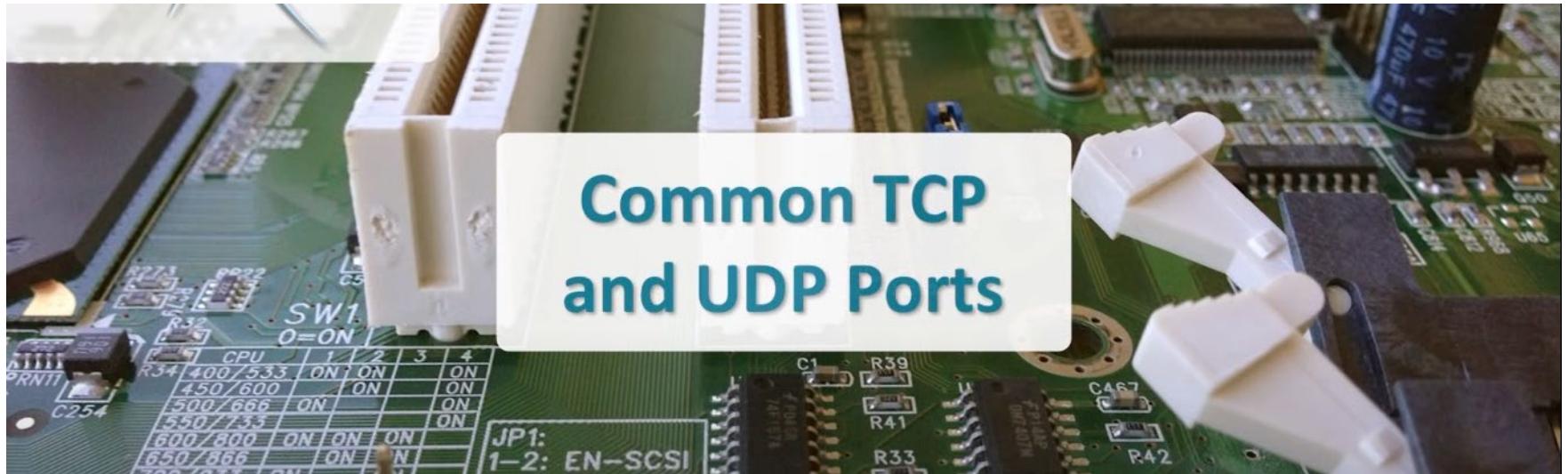


- FTP – File Transfer Protocol –  
tcp/20 (data), tcp/21 (control)
- Telnet – tcp/23
- SMTP – Simple Mail Transfer Protocol – tcp/25
- DNS – Domain Name Services – tcp/53 (zone transfers)
- HTTP – Hypertext Transfer Protocol – tcp/80
- POP3 – Post Office Protocol version 3 – tcp/110
- IMAP – Internet Message Access Protocol v4 – tcp/143
- HTTPS – Hypertext Transfer Protocol Secure – tcp/443
- RDP - Remote Desktop Protocol - tcp/3389

# Lectura: Comparativa TCP vs UDP

## Explicación y comparativa de TCP y UDP

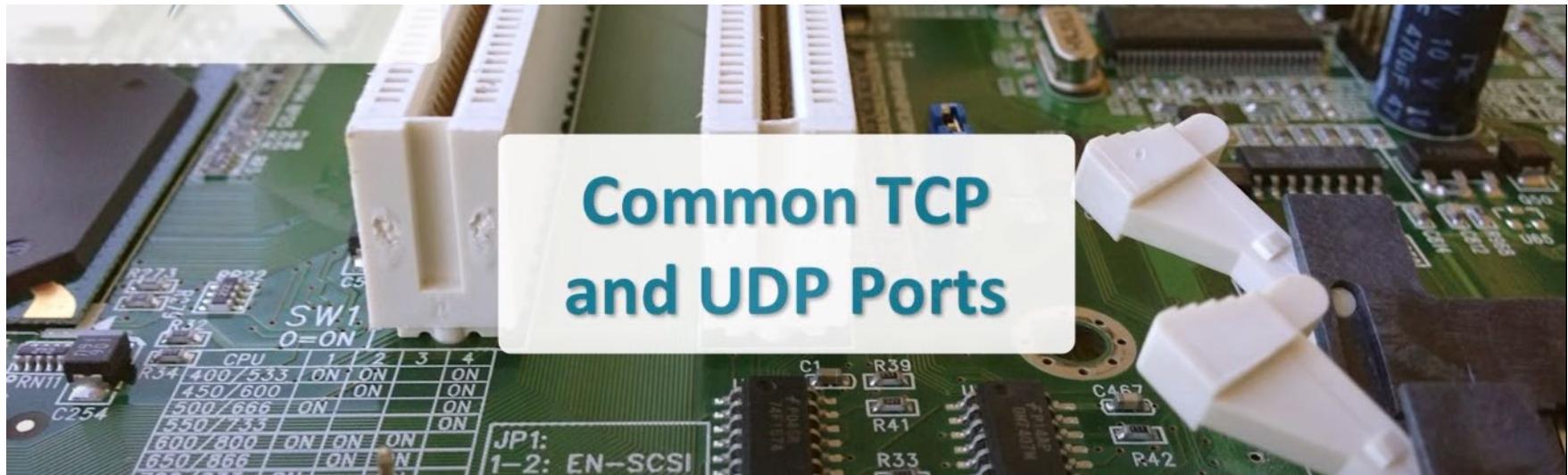
<http://www.steves-internet-guide.com/tcpip-ports-sockets/>



# Video: Comparativa TCP vs UDP

## Explicación y comparativa de TCP y UDP

<https://www.youtube.com/watch?v=r59LJARW8hU>



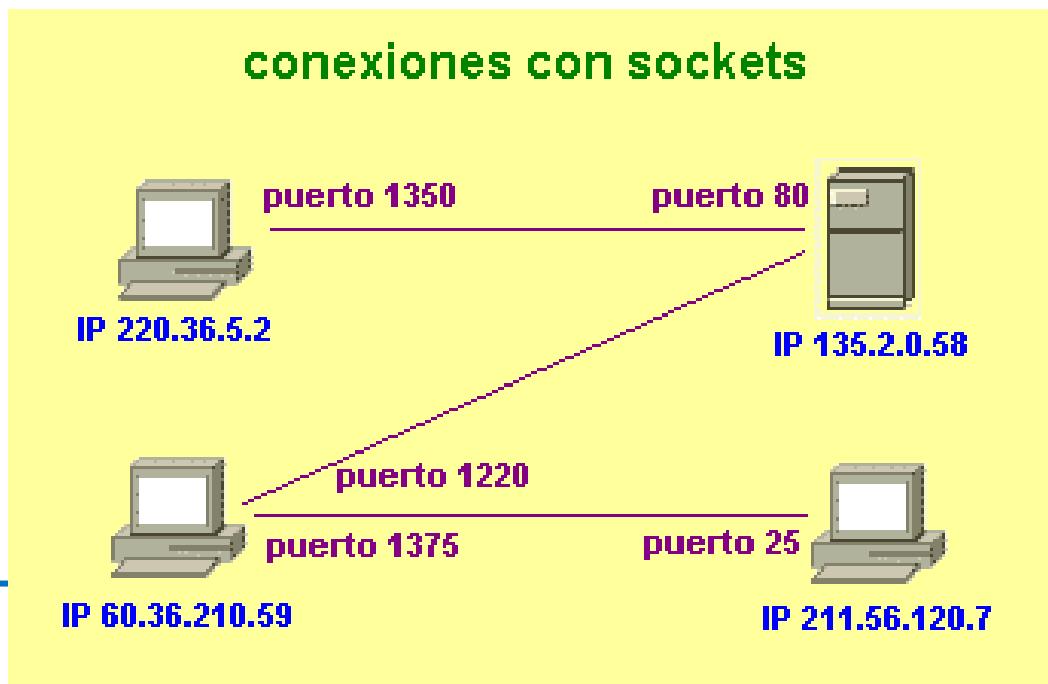
# Protocolo TCP: En detalle

- **TCP = *Transmission Control Protocol***
  - Definido por las RFCs 793, 1122, 1323, 2018 y 2581
  - Protocolo **orientado a conexión**
  - Flujo de datos
    - Stream de bytes
    - Fiable
    - Ordenado
    - Full-dúplex
  - **Además:**
    - Control de flujo, para evitar congestionar al receptor
    - Control de congestión, para evitar congestionar la red

# Protocolo TCP: En detalle

- **Conexiones TCP**

- La conexión se establece entre dos sockets
  - Cada **socket** = **dirección IP**, y **puerto TCP**
- Cada host soporta múltiples conexiones TCP simultáneas

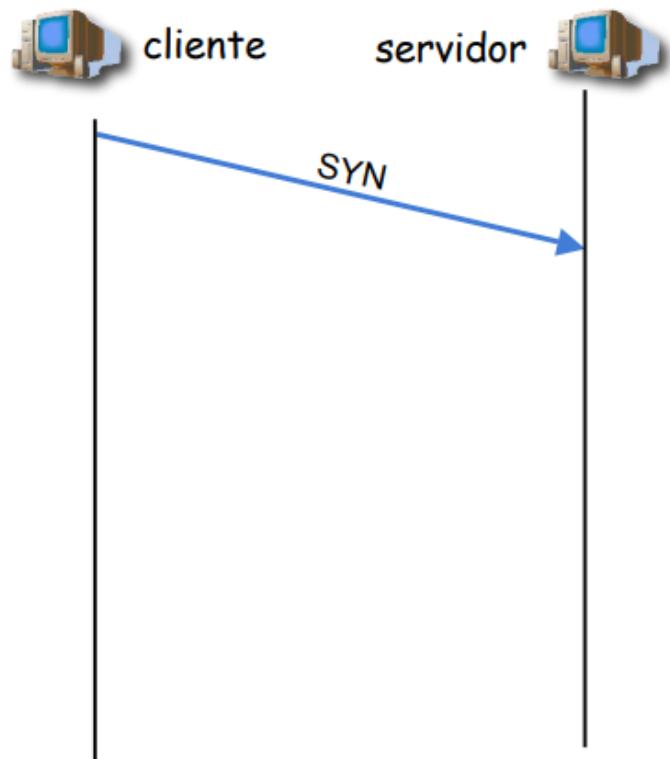


# Protocolo TCP: En detalle

- **Conexiones TCP: Establecimiento**
  - Saludo en tres pasos (*Three way handshake*)

## Paso 1: (SYN)

- El extremo **cliente** envía un segmento **(SYN)** solicitando una conexión al servidor
- El segmento **no tiene datos**, solo cabecera



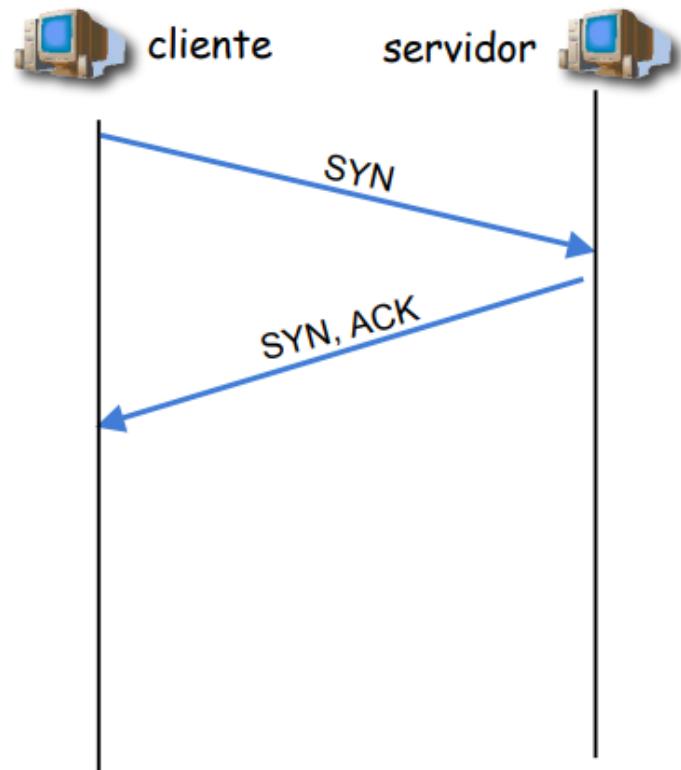
# Protocolo TCP: En detalle

- **Conexiones TCP: Establecimiento**
  - Saludo en tres pasos (*Three way handshake*)

## Paso 2: (SYN → ACK)

- El extremo **servidor** envía un segmento al cliente confirmando (**ACK**) la recepción de SYN.

- El segmento no tiene datos, solo cabecera



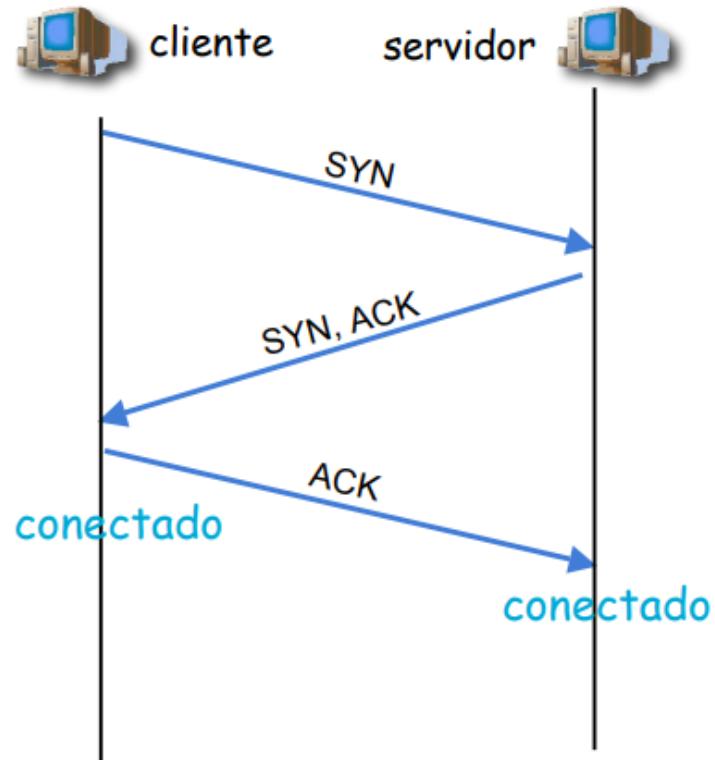
# Protocolo TCP: En detalle

- **Conexiones TCP: Establecimiento**

- Saludo en tres pasos (*Three way handshake*)

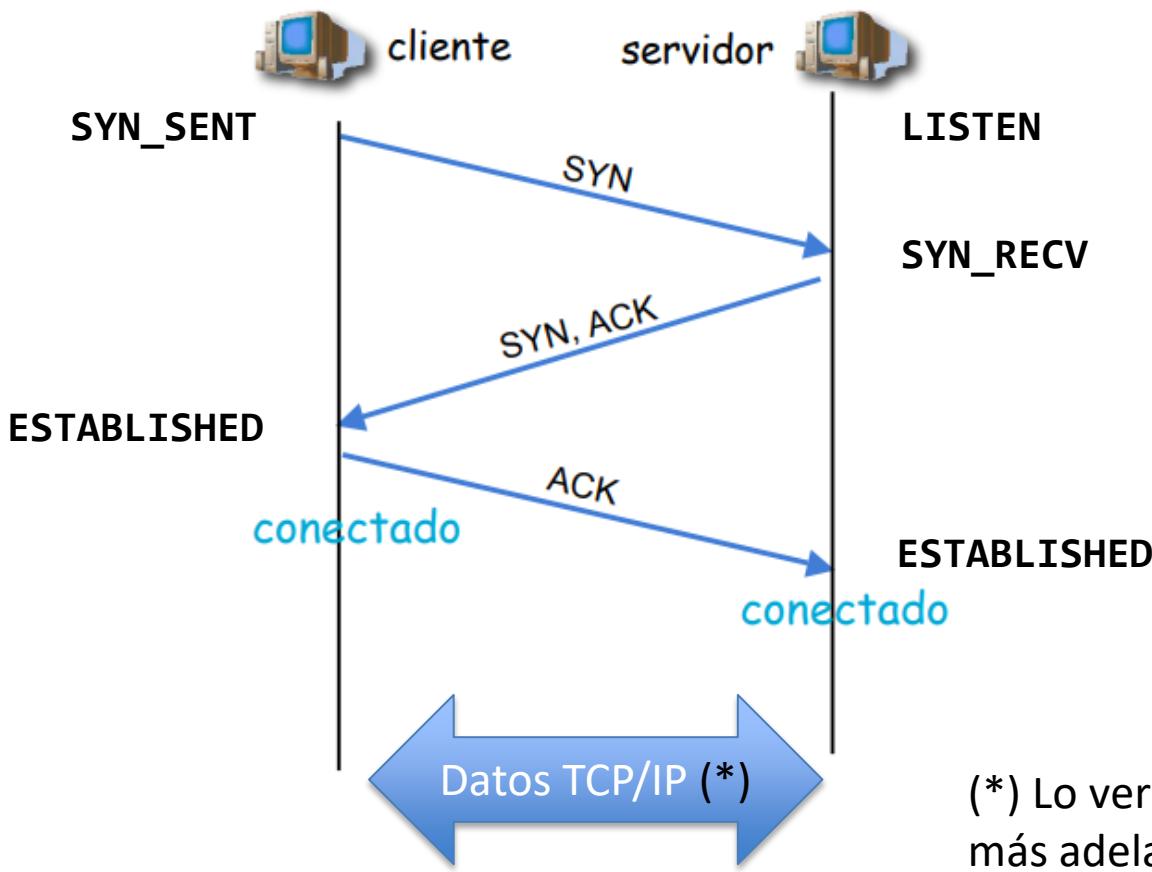
## Paso 3: (SYN → ACK → ACK)

- El extremo **cliente** envía una confirmación (**ACK**) al SYN / ACK del servidor
- El segmento no tiene datos, solo cabecera
- Desde este momento, la conexión está **establecida**!



# Protocolo TCP: En detalle

- **Conexiones TCP: Trasferencia de Datos**

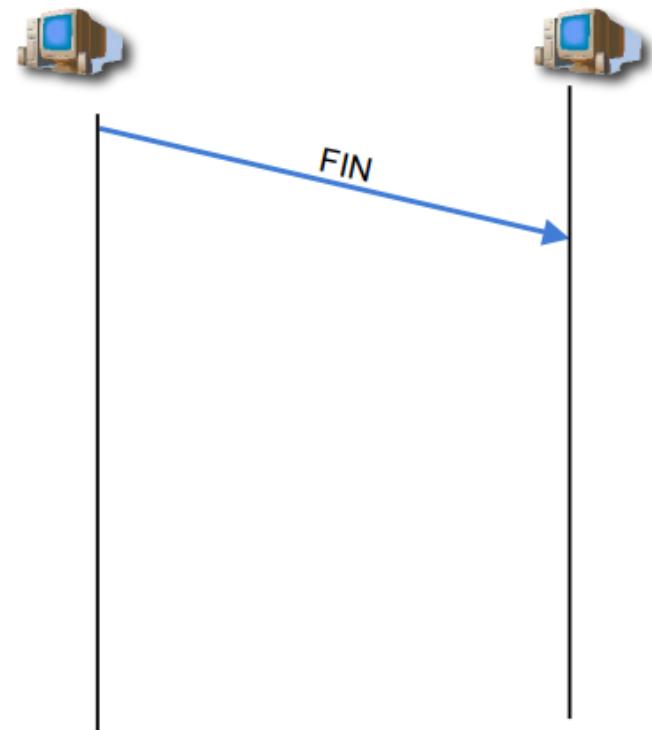


# Protocolo TCP: En detalle

- **Conexiones TCP: Cierre de conexión**
  - Proceso en cuatro pasos

## Paso 1: (**FIN**)

- Un **extremo** envía un segmento al otro solicitando (**FIN**) el cierre de la conexión.
- El segmento no tiene datos, solo cabecera

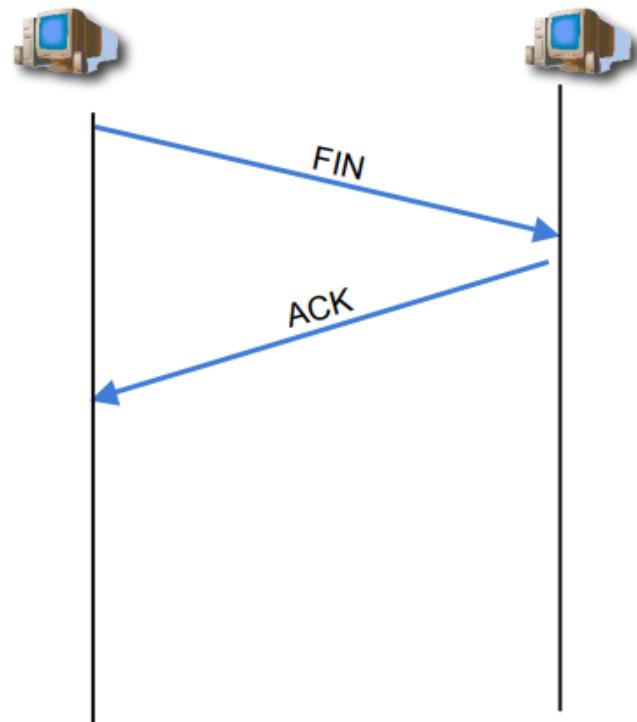


# Protocolo TCP: En detalle

- **Conexiones TCP: Cierre de conexión**
  - Proceso en cuatro pasos

## Paso 2: ( $FIN \rightarrow ACK$ )

- El **otro extremo** envía un segmento al otro confirmando (**ACK**) la recepción del **FIN**.
- Desde ese momento, el extremo que ha enviado el **FIN** ya **no puede** enviar más datos nuevos.
- Se consigue el **cierre sólo en un sentido** de la comunicación.

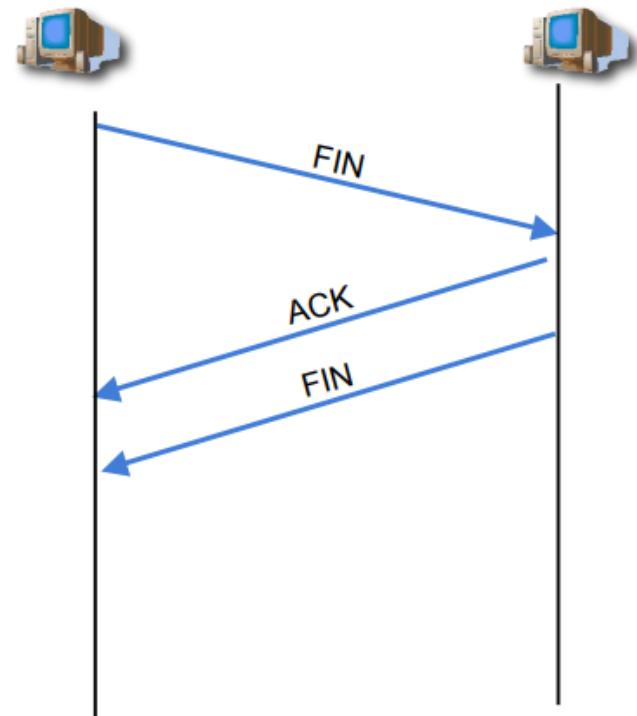


# Protocolo TCP: En detalle

- **Conexiones TCP: Cierre de conexión**
  - Proceso en cuatro pasos

## Paso 3: ( **$FIN \rightarrow ACK \rightarrow FIN$** )

- El **otro extremo** envía un segmento solicitando el cierre de la conexión **( $FIN$ )**
- El segmento no tiene datos, solo cabecera

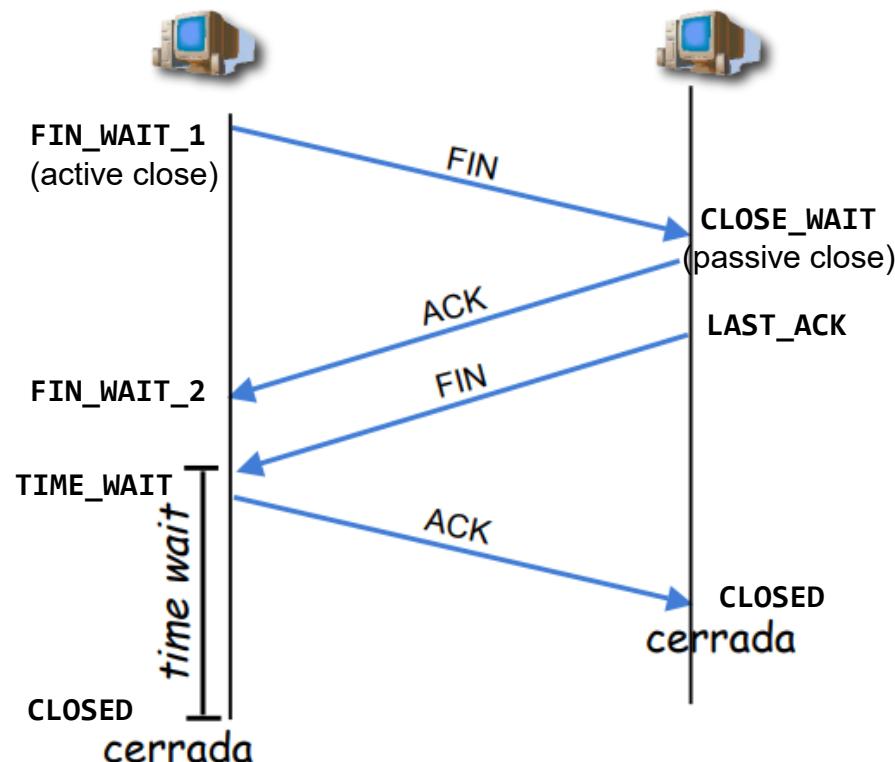


# Protocolo TCP: En detalle

- **Conexiones TCP: Cierre de conexión**
  - Proceso en cuatro pasos

## Paso 4: (***FIN → ACK → FIN → ACK***)

- El **extremo que solicitó el cierre en primer lugar** envía un segmento (***ACK***) confirmando el segundo cierre de la conexión (***FIN***)
- Por si ese último ***ACK* se pierde**, el que lo envió **espera** un tiempo (podría tener que volverlo a enviar)
- Desde este momento, la conexión está **cerrada**!



# Protocolo TCP: En detalle

## TCP: Diagrama de Estados (RFC793)

- Durante su ciclo de vida, una **conexión** puede pasar a través de una serie de estados.
- **El paso** de un estado a otro se produce como **respuesta a eventos**:
  - Llamadas del usuario:
    - **OPEN**
    - **SEND**
    - **RECEIVE**
    - **CLOSE**
    - **ABORT**
    - **STATUS**
  - Segmentos recibidos, en particular los que tengan los *flag* de:
    - **SYN**
    - **ACK**
    - **RST**
    - **FIN**
  - Timeouts

**Video: TCP Connection States**  
[https://www.youtube.com/watch?v=pF3S\\_qAbv54](https://www.youtube.com/watch?v=pF3S_qAbv54)



# Protocolo TCP: En detalle

## TCP: Diagrama de Estados (RFC793)

- Durante su ciclo de vida, una **conexión** puede pasar a través de una serie de **estados**:
  - **LISTEN** represents waiting for a connection request from any remote TCP and port.
  - **SYN-SENT** represents waiting for a matching connection request after having sent a connection request.
  - **SYN-RECEIVED** represents waiting for a confirming connection request acknowledgment after having both received and sent a connection request.
  - **ESTABLISHED** represents an open connection, data received can be delivered to the user. The normal state for the data transfer phase of the connection.
  - **FIN-WAIT-1** represents waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent.



# Protocolo TCP: En detalle

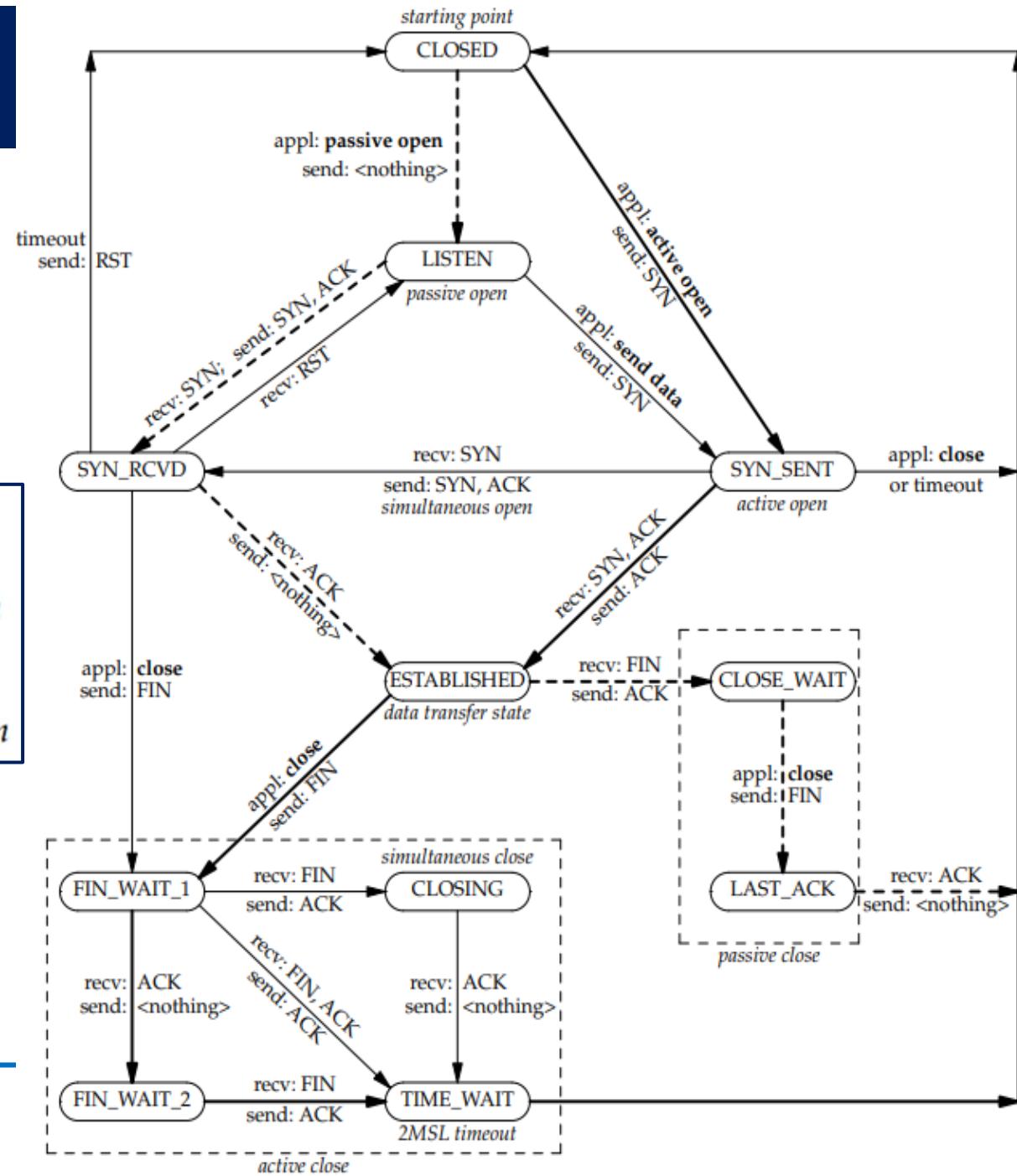
## TCP: Diagrama de Estados (RFC793) (Continuación)

- **FIN-WAIT-2** represents waiting for a connection termination request from the remote TCP.
- **CLOSE-WAIT** represents waiting for a connection termination request from the local user.
- **CLOSING** represents waiting for a connection termination request acknowledgment from the remote TCP.
- **LAST-ACK** represents waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request).
- **TIME-WAIT** represents waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.
- **CLOSED** represents no connection state at all.

# Protocolo TCP: En detalle

## TCP: Diagrama de Estados

➔ normal transitions for client  
 ➔ normal transitions for server  
 appl: state transitions taken when application issues operation  
 recv: state transitions taken when segment received  
 send: what is sent for this transition



TCP/IP Illustrated, Volume 2: The Implementation by Gary R. Wright and W. Richard Stevens, Addison-Wesley Publishing Company, Inc.

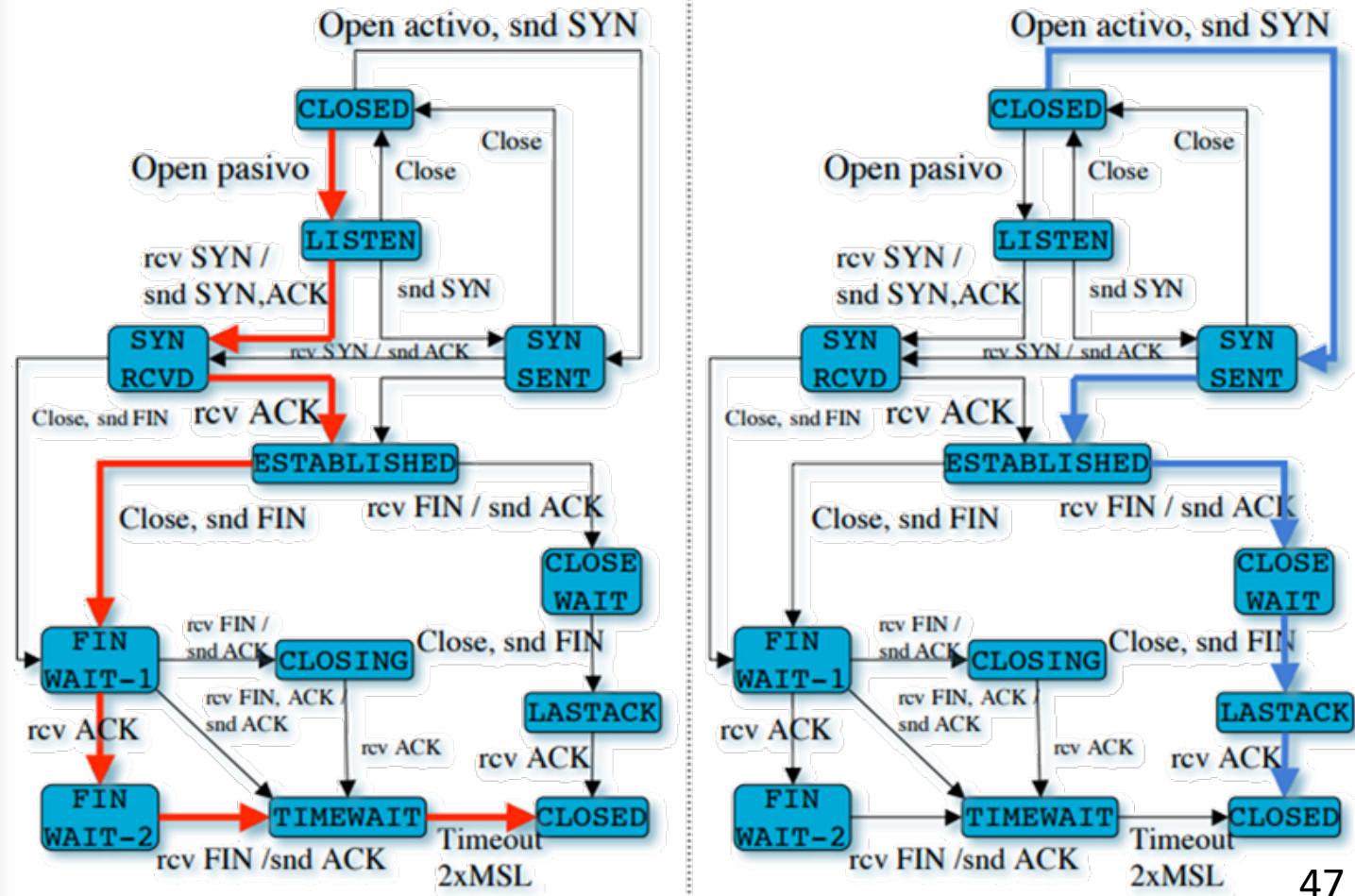


# Protocolo TCP: En detalle

## TCP: Diagrama de Estados. Ejemplo

Servidor

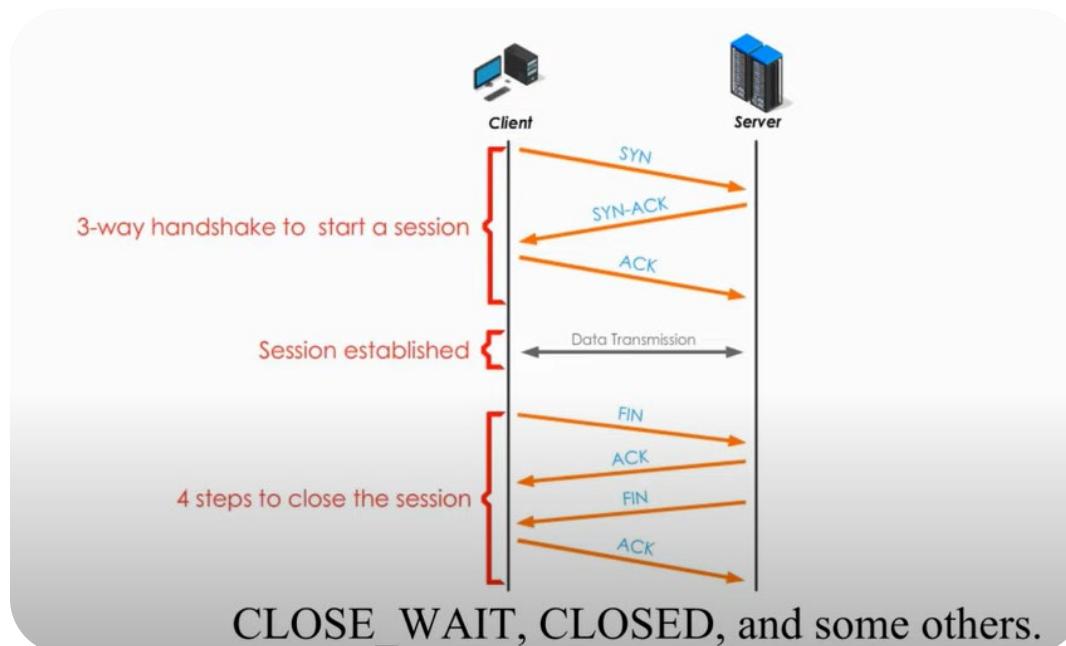
Cliente



# Videos: TCP session: Circle of Life

## TCP session: Circle of Life (*Sunny Classroom*)

<https://www.youtube.com/watch?v=K4BE4Qf2Uf8>

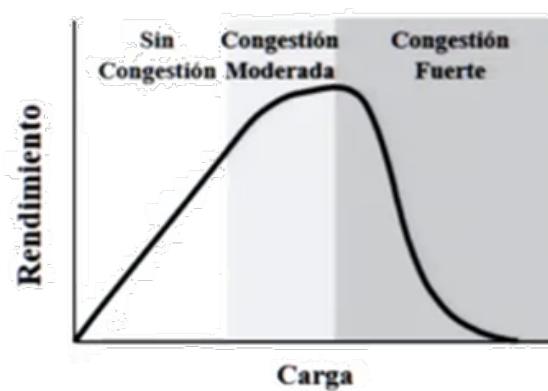
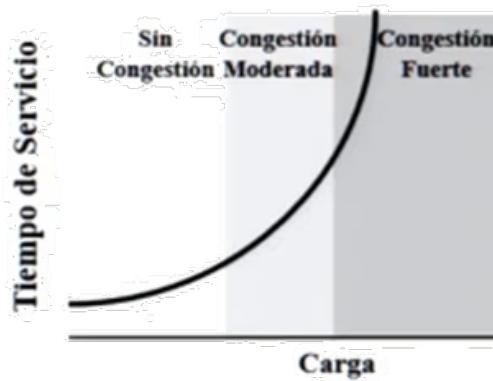


# Protocolo TCP: En detalle

## TCP: Control de Congestión

### Objetivos del control de congestión

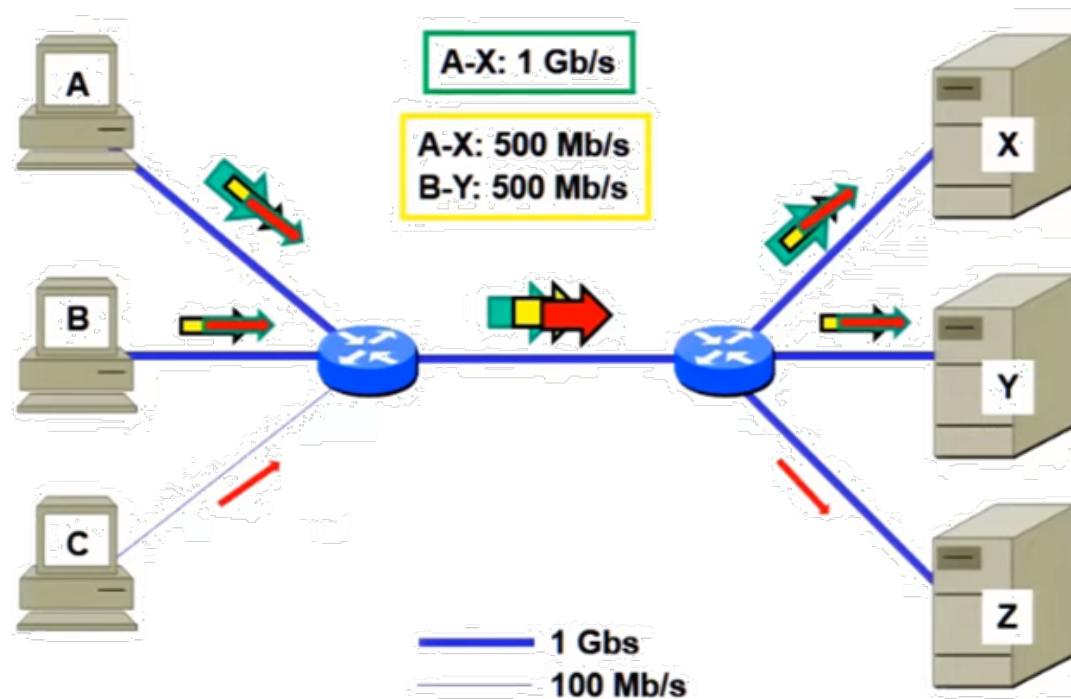
1. Evitar el ‘congestion collapse’
2. Repartir los recursos de manera equitativa entre las conexiones establecidas
3. Aprovechar lo mejor posible los recursos disponibles (acercarse tanto como sea posible al 100% de carga sin incurrir en el colapso por congestión)



# Protocolo TCP: En detalle

## TCP: Control de Congestión

### Ejemplo de reparto equitativo



# Protocolo TCP: En detalle

## **TCP: Control de Congestión**

### Control de congestión en TCP

- TCP detecta la congestión de forma implícita: si se pierden segmentos supone que hay congestión y baja el ritmo.
- Utiliza un algoritmo en dos fases conocidas como *slow-start* (arranque lento) y *congestion avoidance* (evitación de la congestión)
- El TCP emisor mantiene en todo momento una ventana de congestión (congestion window, cwnd) que establece cuantos datos puede enviar en cada momento.
- La cwnd no se anuncia a nadie, es interna del TCP e independiente de la ventana de control de flujo que va en la cabecera TCP. Se usa siempre la más pequeña de ambas.

**Ventana de control de flujo**  
Máximo admite receptor (buffers)

**Ventana de congestión**  
Lo que estima el proceso TCP. No lo comunican al resto.

A la hora de enviar datos, TCP respeta ambas ventanas. No supera el tamaño de la más pequeña de ambas.

Si ventana de control de flujo anuncia que el otro TCP tiene disponible para el 32 Kb para el y mi ventana de congestión dice que tiene 4 Kb, la transmisión se adapta al mínimo que es 4Kb.



# Protocolo TCP: En detalle

## **TCP: Control de Congestión**

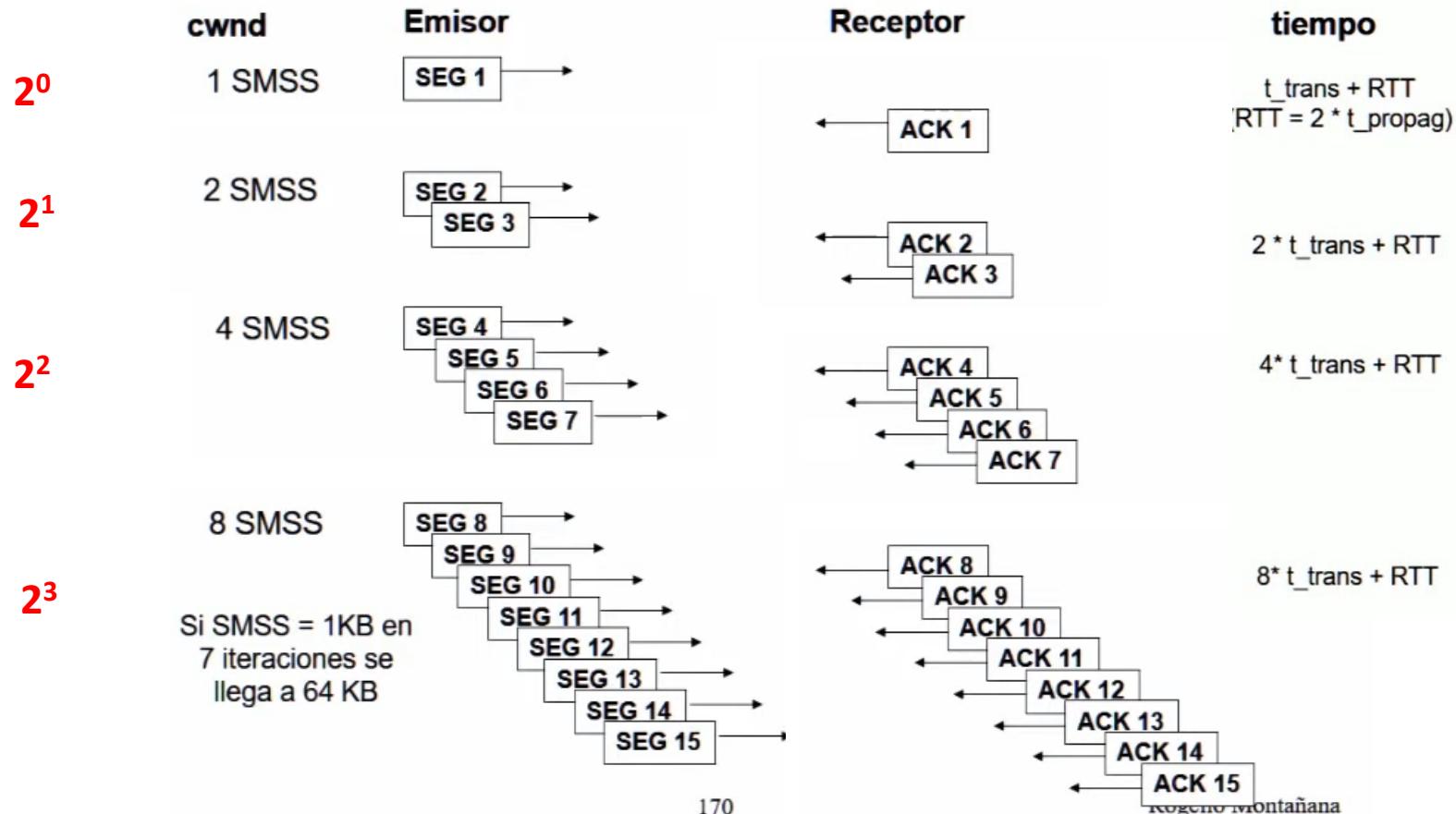
### Arranque lento (Slow Start, fase 1)

- Inicialmente la cwnd (ventana de congestión) tiene un tamaño de entre uno y cuatro SMSS (Sender Maximum Segment Size). El valor exacto depende de la implementación
- Por cada segmento enviado con éxito la cwnd se amplía en un SMSS. El éxito se conoce al recibir el ACK
- En la práctica esto supone un crecimiento exponencial de la cwnd (en potencias de dos)
- Mientras no se pierda ningún segmento la cwnd crece ilimitadamente.

# Protocolo TCP: En detalle

## TCP: Control de Congestión

### Control de congestión, fase 1 (slow start)



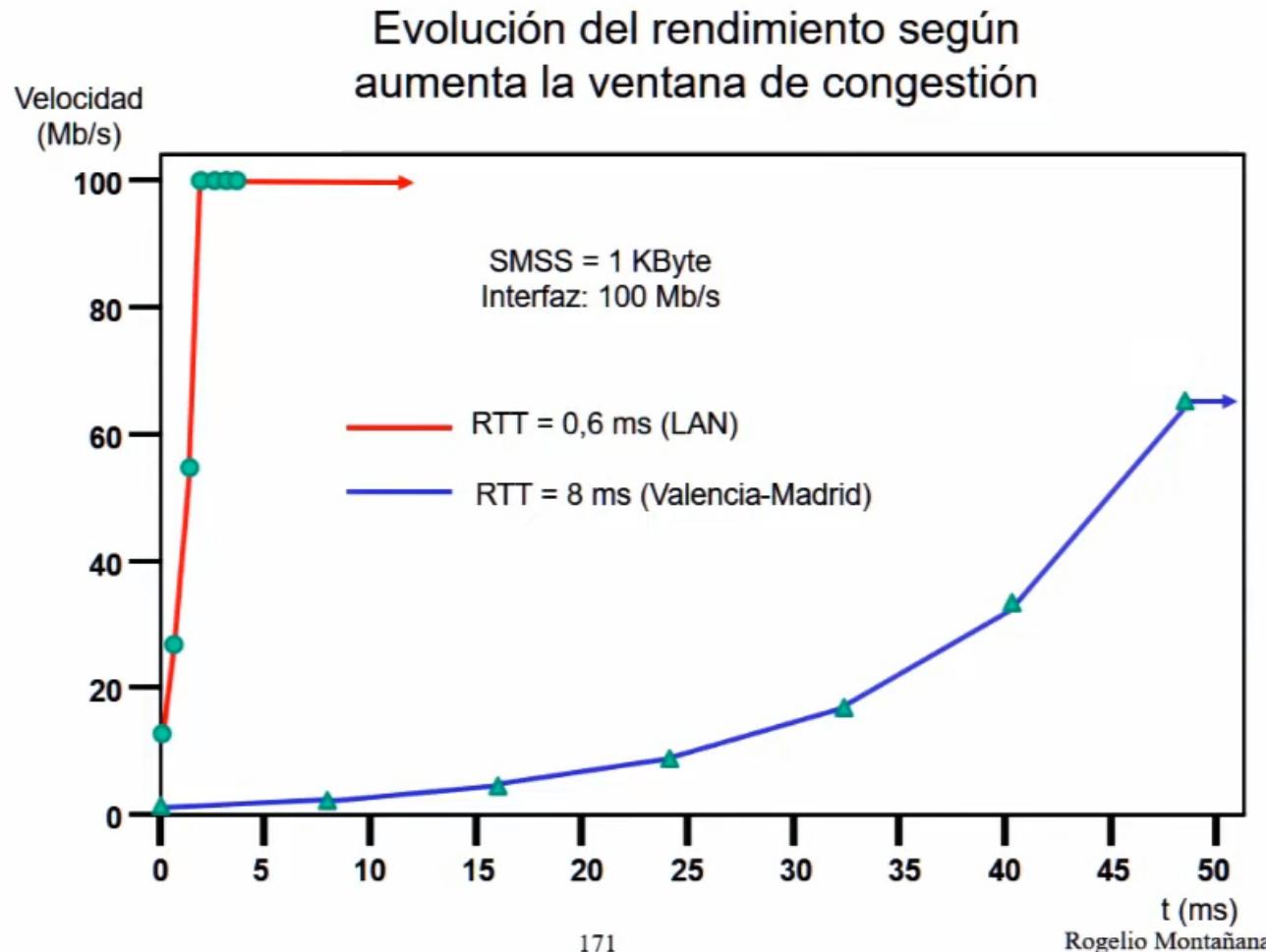
170

Rogelio Montañana



# Protocolo TCP: En detalle

## TCP: Control de Congestión



# Videos: TCP en detalle

## TCP en detalle (Rojelio Montañeda)

Para saberlo todo: Vídeos 10.1 – 10.27

<https://www.aulaclic.es/redes/>

Control de congestión de TCP

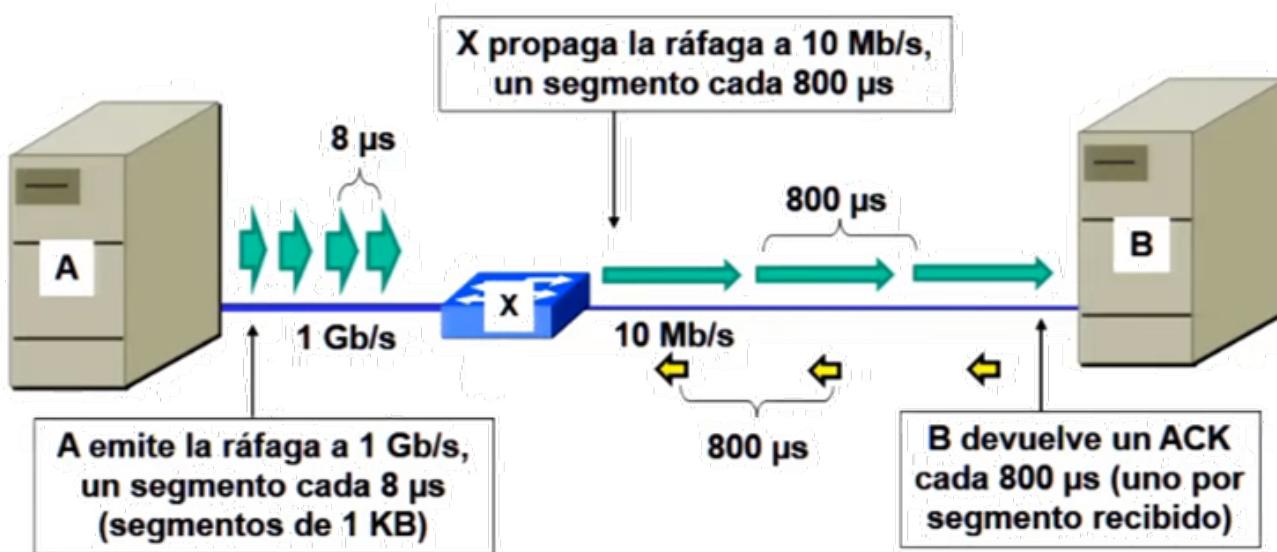
<https://www.youtube.com/watch?v=nS08p-lsbPM>



# Protocolo TCP: En detalle

## TCP: Control de Congestión

### El 'reloj ACK'



Como A espera el ACK de B antes de enviar la siguiente tanda el caudal se ajusta automáticamente al enlace de menor capacidad en la ruta  
Esto es lo que se conoce como el reloj ACK o 'ACK clock'

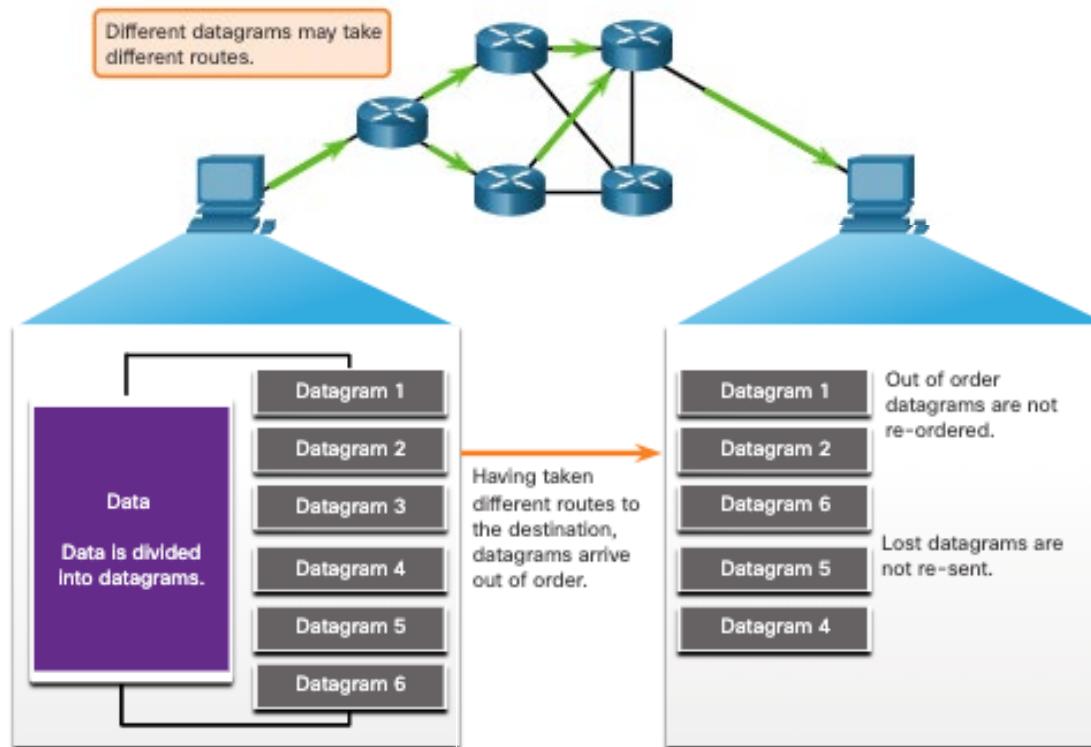
# Protocolo UDP: En detalle

- ***UDP = User Datagram Protocol***
  - Definido por las RFCs 768
  - Protocolo **NO orientado a conexión**
  - Baja sobrecarga debido a su encabezado reducido
  - No gestiona ni resuelve incidencias:
    - NO realiza un seguimiento de los números de secuencia.
    - NO puede reordenar los datagramas en el orden de la transmisión.
    - Simplemente reensambla los datos en el orden en que se recibieron y los entrega a la aplicación.

**Todo queda en manos del desarrollador del software**

# Protocolo UDP: En detalle

UDP depende del mejor esfuerzo de la red, lo cual puede suponer la pérdida de información así como del orden.



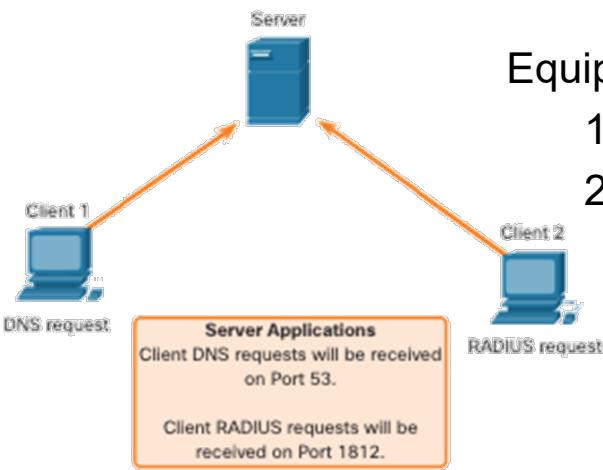
Fuente: Cisco NetAcad

# Protocolo UDP: En detalle

## Situaciones apropiadas para el uso de UDP:

- Intercambios **escasos**. Ej.: Consultas DNS, DHCP, ...
- Requerimiento de **tiempo real**, por lo que no tiene sentido la espera de confirmaciones. Ej.: Videoconferencias, llamadas, retransmisiones audiovisuales, etc.
  - Existen mecanismos para este tipo de aplicaciones como el **protocolo de aplicación RTP** (*Real Time Protocol*)
- Intercambios **regulares**, por los que no importa en exceso la pérdida de alguno. Ej: NTP, SNMP, sensores domóticos, etc.
- Tráfico **broadcast/multicast** (NO PUEDE ENVIARSE TCP)
- Implementaciones **sencillas**, dónde el desarrollador gestiona toda la comunicación. Ej: TFTP, ...

# Protocolo UDP: Ejemplo de uso

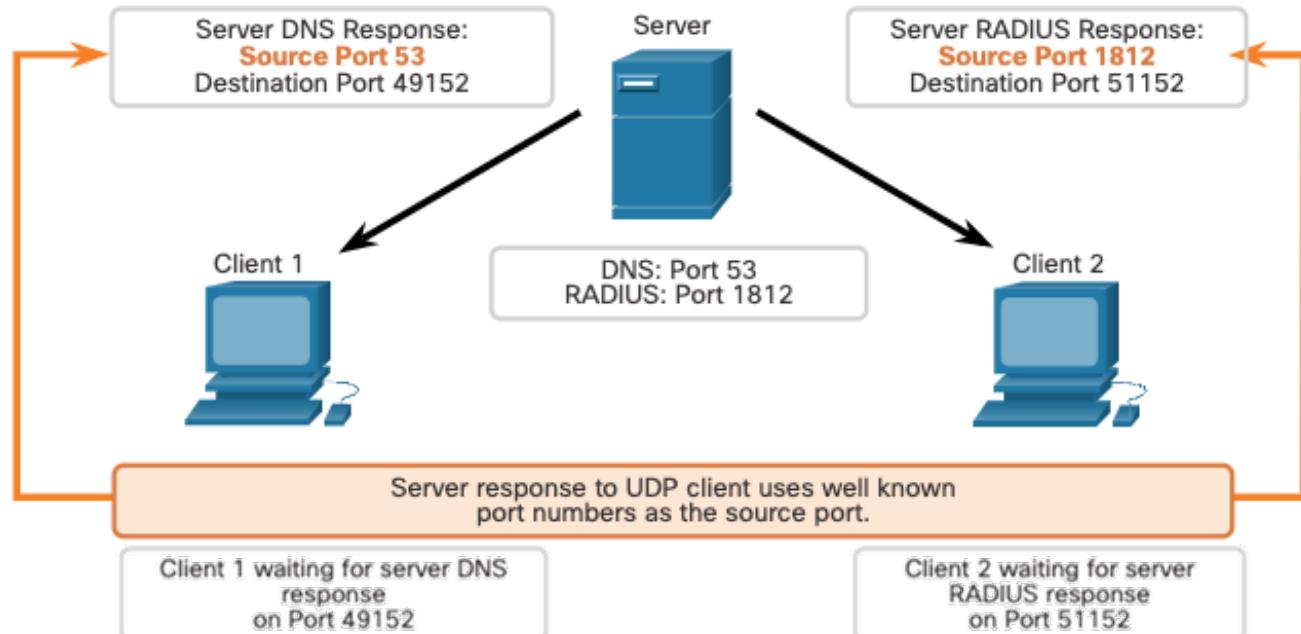


Equipo con dos procesos servidores a la escucha:

1. Servidor DNS (puerto: 53)
2. Servidor de autenticación RADIUS\* (Puerto 1812)

Puerto destino es el puerto bien conocido o registrado que se asigna al proceso de servidor.

Proceso **cliente UDP** selecciona dinámicamente un número de puerto del intervalo de números de puerto y lo utiliza como puerto de origen para la conversación.



# Videos: UDP en detalle

## UDP en detalle (Rojelio Montañeda)

*Protocolo UDP*

<https://www.youtube.com/watch?v=ez82JeEMYjQ>

*Protocolo UDP: Ejemplo llamada IP*

<https://www.youtube.com/watch?v=vqeapa1MWWA>



## Arquitectura TCP/IP

- Capa de transporte
  - Funcionalidades
  - Puertos
  - Protocolos
- Capa de aplicación
  - RTP
  - NAT
  - DHCP

# Protocolo RTP

- **RTP = Real-time Transport Protocol**
  - Definido por las RFC 1889 (1996), 3550 (2003)
  - Utilizado para transmisión de información en tiempo real en **aplicaciones multimedia** como audio y vídeo
  - Trabaja con el protocolo de transporte **UDP**
  - Han surgido otros más recientes, basados en HTTP ([ver más](#))

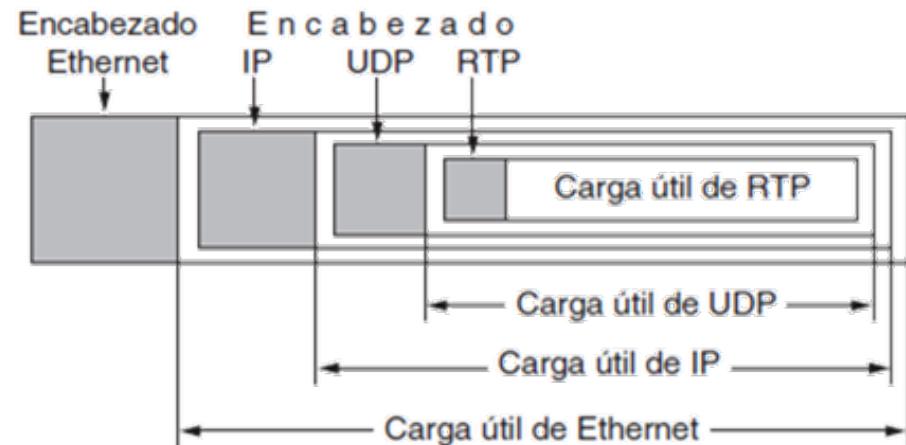
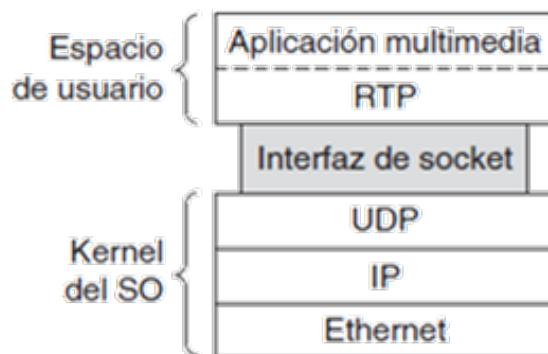


Figura 6-30. (a) La posición de RTP en la pila de protocolos.

(b) Anidamiento de paquetes.

# Protocolo RTP

## Funcionamiento de RTP

- **Multiplexa** varios flujos de datos en un solo flujo de paquetes UDP.
- Flujo puede ser **unidifusión** o **multidifusión** (unicast/multicast).
- Los routers no dan a sus paquetes un trato especial (al ser UDP), a menos que se habiliten características de calidad de servicio.
- **NO** hay **garantías** especiales acerca de la entrega, así que los paquetes se pueden perder, retrasar, corromper, etcétera.
- **NO** tiene **confirmaciones** de recepción ni ningún mecanismo para solicitar retransmisiones.

# Protocolo RTP

## Cabecera de RTP

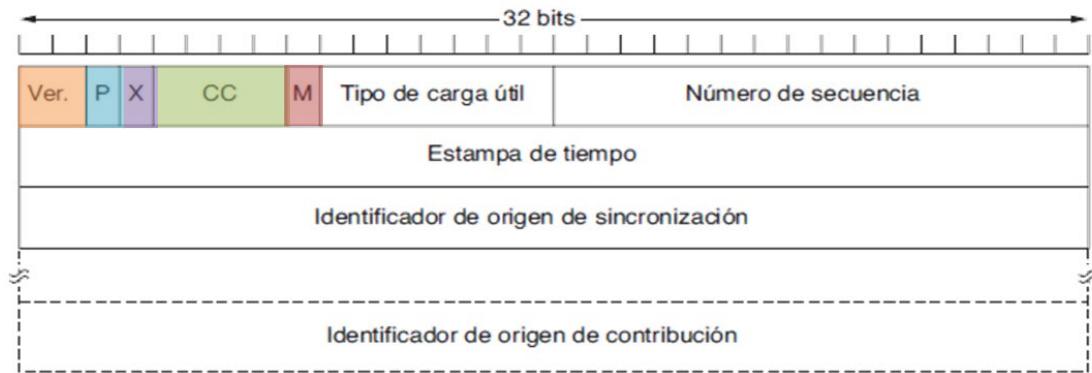


Figura 6-31. El encabezado RTP.

- **Version**: Versión, actualmente la 2
- **Bit P**: Indica que el paquete se ha rellenado para formar un múltiplo de 4 bytes. El último byte de relleno indica cuántos bytes se agregaron.
- **Bit X**: indica que hay un encabezado de extensión. Formato y significado de este encabezado no se definen. Lo único que se define es que la primera palabra de la extensión proporciona la longitud.
- **CC**: Número de fuentes de contribución presentes (0-15).
- **Bit M**: Marcador específico de la aplicación. Puede utilizarse para marcar el inicio de una trama de video, el inicio de una palabra en un canal de audio o algo más que la aplicación entienda.

# Protocolo RTP

## Cabecera de RTP

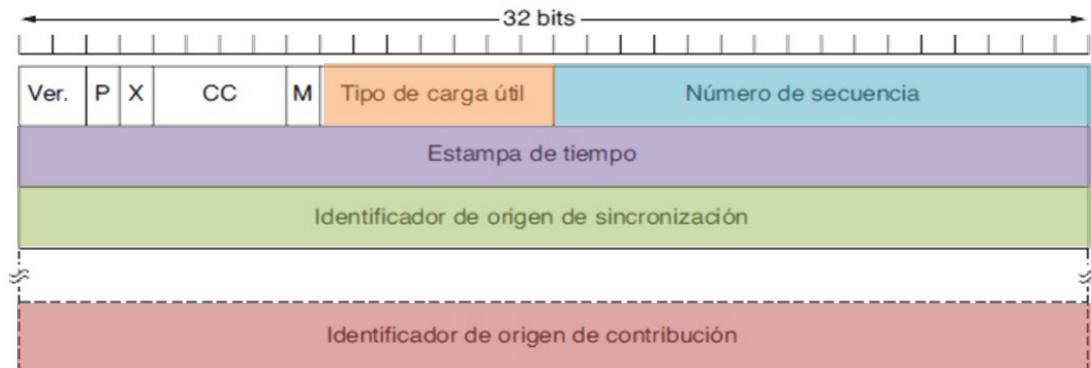


Figura 6-31. El encabezado RTP.

- **Tipo de carga útil**: Algoritmo de codificación utilizado (ej. audio de 8 bits sin compresión, MP3, etc.). Puesto que cada paquete lleva este campo, la codificación puede cambiar durante la transmisión.
- **Nº de secuencia**: Contador de paquetes RTP enviados. Se utiliza para detectar paquetes perdidos.
- **Marca de tiempo**: Cuándo se creó la primera muestra en el paquete. Este valor puede ayudar a reducir la variabilidad de la sincronización conocida como variación del retardo (**jitter**) en el receptor, al desacoplar la reproducción del tiempo de llegada del paquete.
- **Identificador de origen de sincronización**: indica a cuál flujo pertenece el paquete. Es el método utilizado para multiplexar y desmultiplexar varios flujos de datos en un solo flujo de paquetes UDP.
- **Identificadores de origen de contribución** (en caso de que haya): Utilizados cuando hay mezcladoras en el estudio. En ese caso, la mezcladora es el origen de la sincronización y los flujos que se mezclan se listan aquí.

# Protocolo RTP

## Funcionamiento de RTP

Conviene utilizar buffer para poder gestionar el “Jitter”

Debe decidirse que hacer con los paquetes faltantes: ¿Esperar por ellos o seguir con el siguiente?



Figura 6-32. Hay que colocar los paquetes en un búfer para uniformar el flujo de salida.



# Protocolo RTP

## Funcionamiento de RTP

Conviene utilizar buffer para poder gestionar el “Jitter” ...

... y además: Elegir un buen punto de reproducción (tiempo en el que comenzar a reproducir, asociado al tamaño del buffer)

- Si la red está mal gestionada, tiene mucha demanda o no tiene capacidad suficiente puede llegar a hacer imposible la retransmisión
- Suele utilizarse junto a **RTCP** (RTP Control Protocol) que permite controlar la información en una sesión RTP.

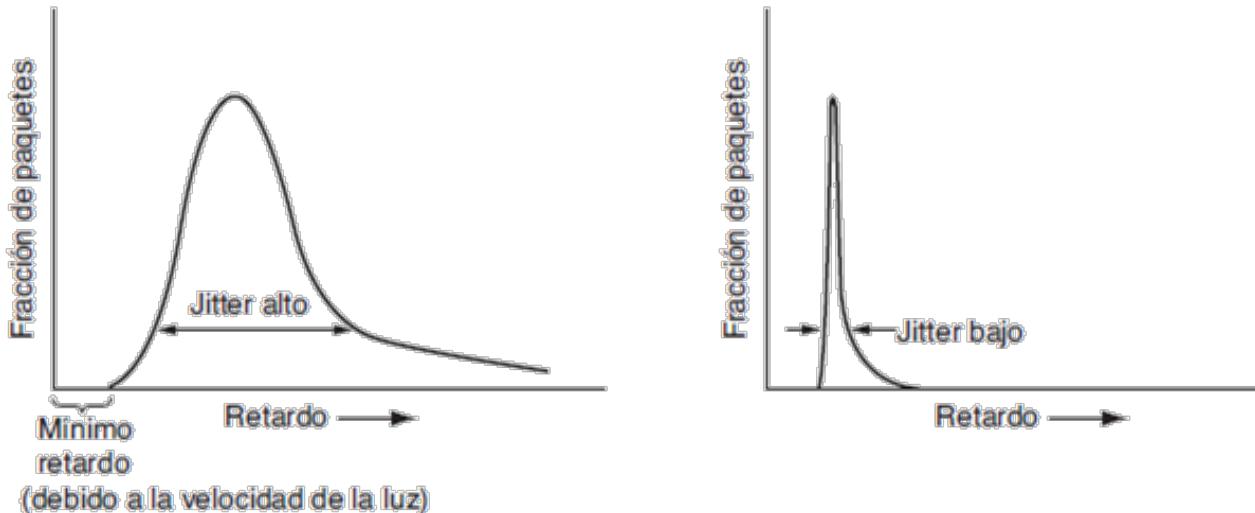


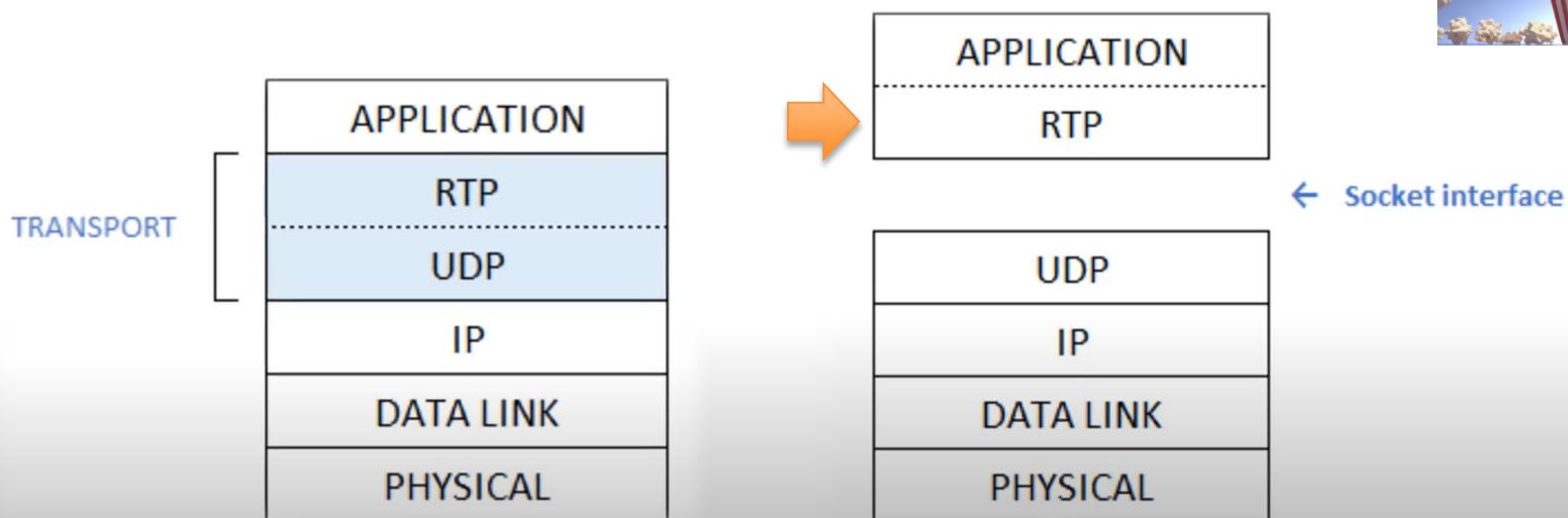
Figura 6-33. (a) Jitter alto. (b) Jitter bajo.

# Videos: RTP

## RTP (*Network Encyclopedia*)

*Real-time Transport Protocol (RTP) and RTCP*

<https://www.youtube.com/watch?v=xqL2mF-dBZs>



## Arquitectura TCP/IP

- Capa de transporte
  - Funcionalidades
  - Puertos
  - Protocolos
- Capa de aplicación
  - RTP
  - NAT
  - DHCP

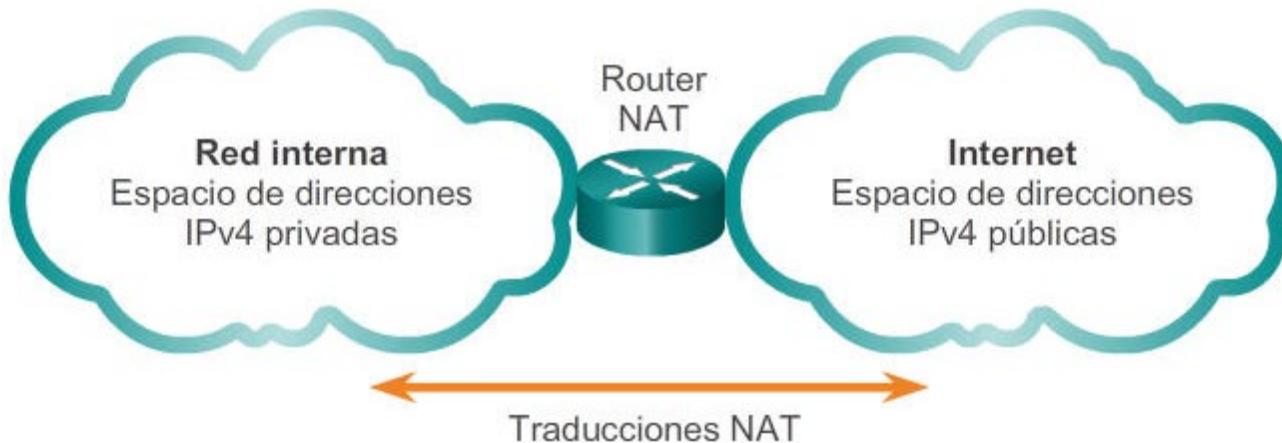
# NAT (Network Address Translation)

## NAT (Network Address Translation)

- Espacio de direcciones IPv4 no es suficientemente grande para que todos los dispositivos puedan conectarse a Internet
- Direcciones privadas (RFC 1918) diseñadas para utilizarse solo dentro de una organización o un sitio
- **Routers de Internet**
  - **No enrutan direcciones privadas**
  - **Sí enrutan direcciones públicas**

**NAT se utiliza para realizar esta traducción**

# NAT (Network Address Translation)



Las direcciones privadas de Internet están definidas en RFC 1918:

Clase	Rango de direcciones internas RFC 1918	Prefijo CIDR
A	10.0.0.0 a 10.255.255.255	10.0.0.0/8
B	172.16.0.0 a 172.31.255.255	172.16.0.0/12
C	192.168.0.0 a 192.168.255.255	192.168.0.0/16

Fuente: Cisco NetAcad



# NAT: Tipos de direcciones

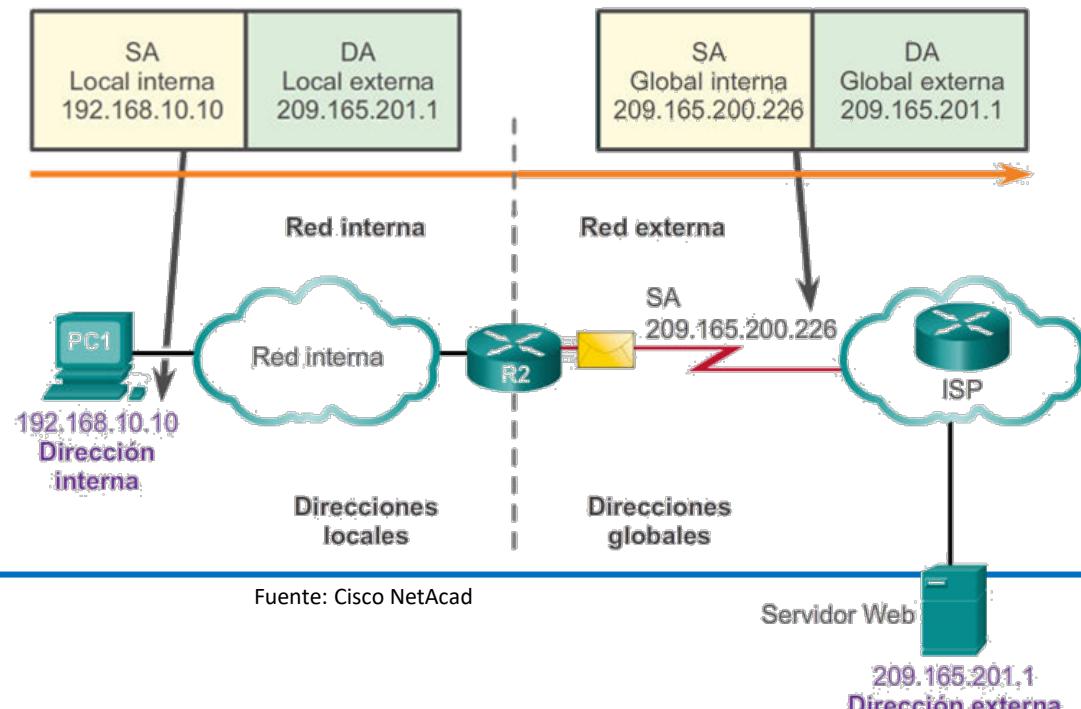
NAT tiene 4 tipos de direcciones:

1. **Local interna:** Dirección **privada** asignada a un host de la red **interna**
2. **Global interna:** Dirección **pública** que el router asigna al mensaje interno para emitirlo por internet
3. **Global externa:** Dirección asignada por su propietario a un host de la red **externa**
4. **Local externa:** Dirección de un host externo desde el punto de vista desde los host de la red **interna**

Términos “interna” y “externa” se combinan con términos “global” y “local” para referenciar a direcciones específicas

**SA:** Source Address

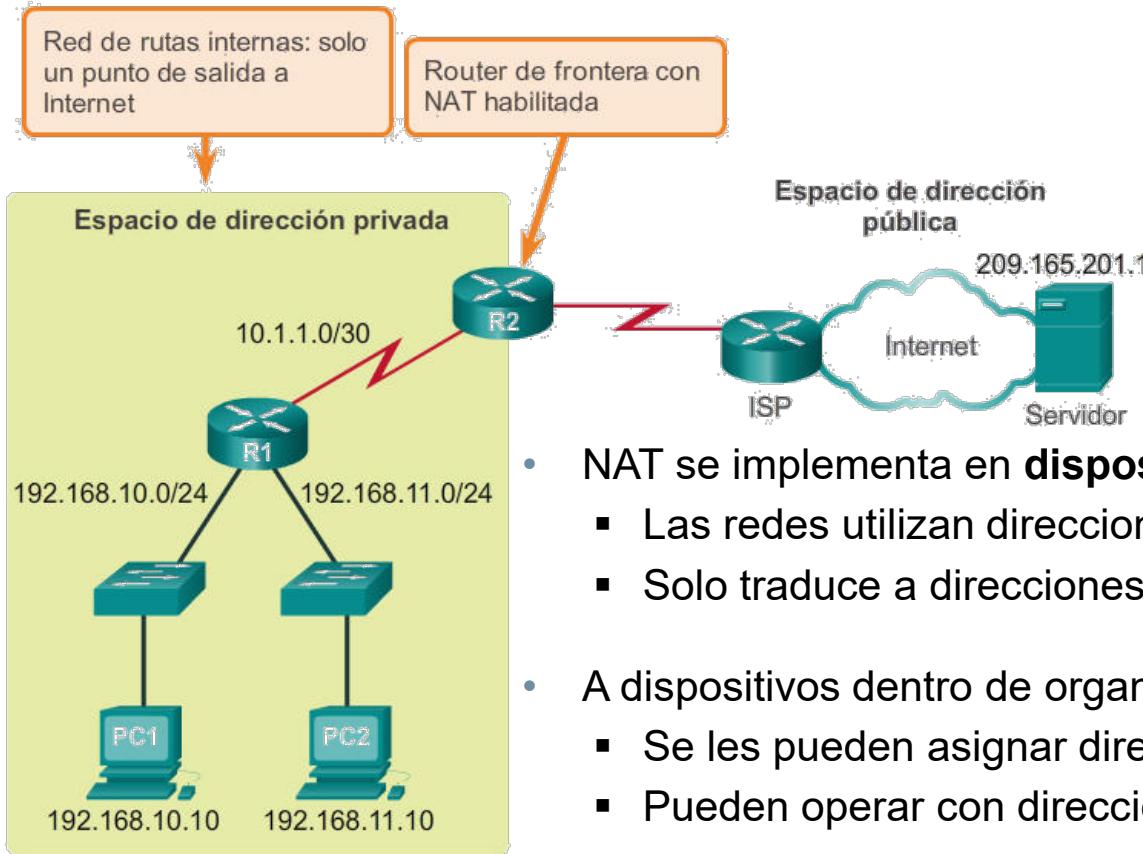
**DA:** Destination Address



Fuente: Cisco NetAcad



# NAT (Network Address Translation)



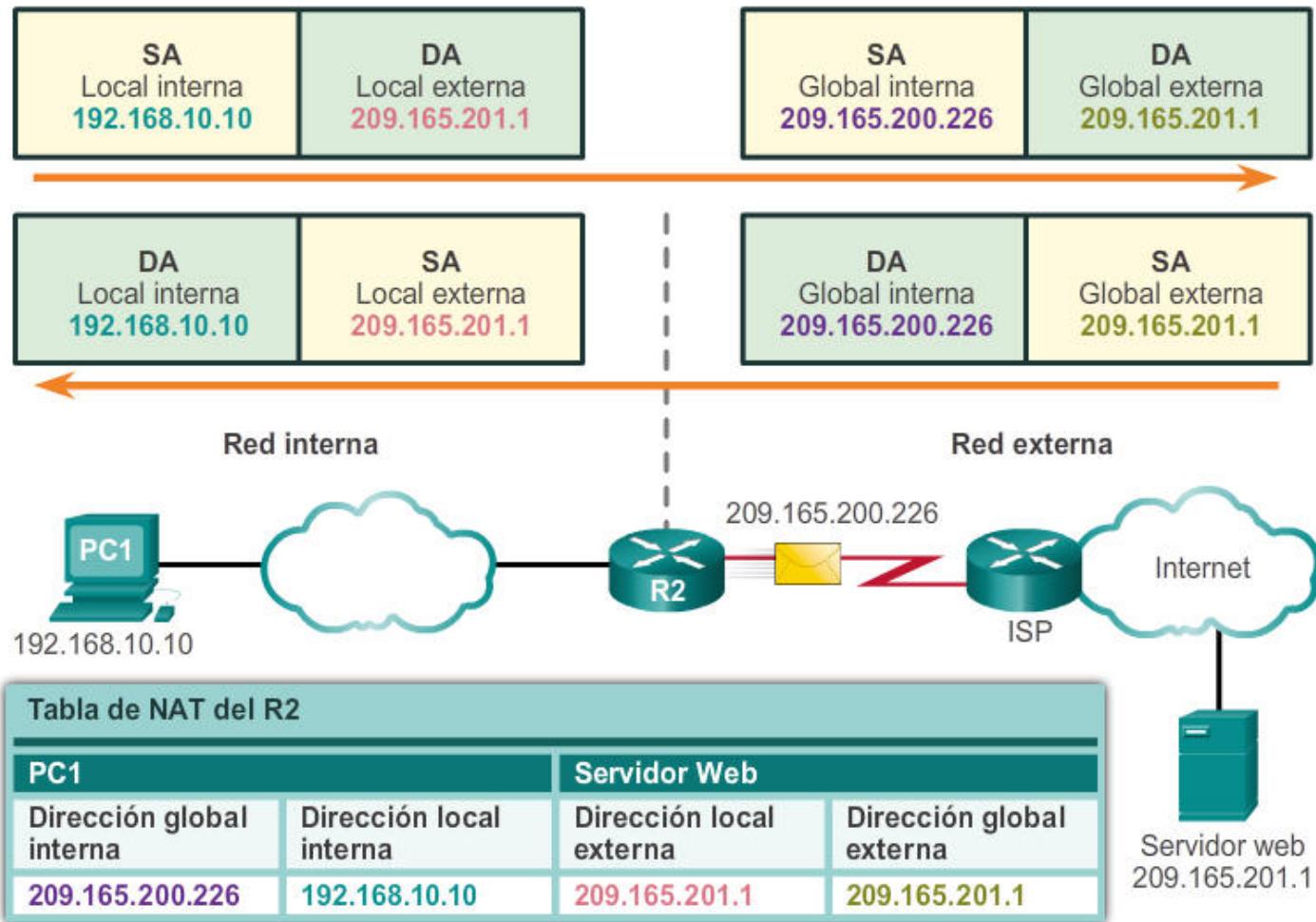
## Router frontera con NAT:

- **Retransmite** paquetes TCP/UDP y algunos ICMP.
- **NO retransmite** info. de routing

- NAT se implementa en **dispositivos** de red **fronterizos**
  - Las redes utilizan direcciones privadas de manera interna
  - Solo traduce a direcciones públicas cuando sea necesario
- A dispositivos dentro de organización
  - Se les pueden asignar direcciones privadas
  - Pueden operar con direcciones locales únicas
- Router traduce direcciones a dirección pública globalmente única
  - Al enviar tráfico a otras organizaciones o Internet
  - Cuando se debe recibir tráfico de estas

Fuente: Cisco NetAcad

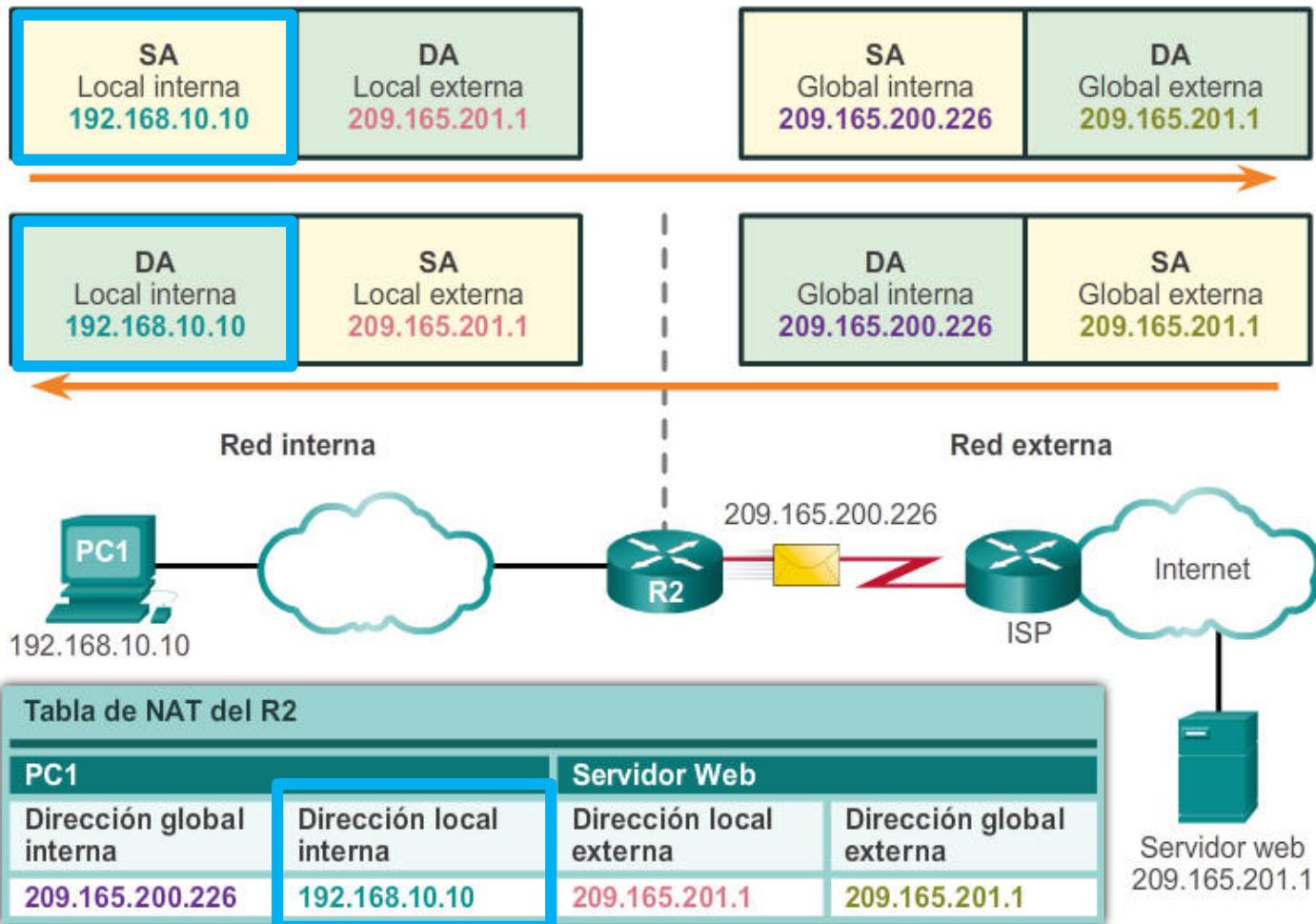
# NAT: Ejemplo



Fuente: Cisco NetAcad

# NAT: Ejemplo

Dirección local interna:  
192.168.10.10 se asignó a PC1  
Esta es la dirección local interna de PC1



Fuente: Cisco NetAcad

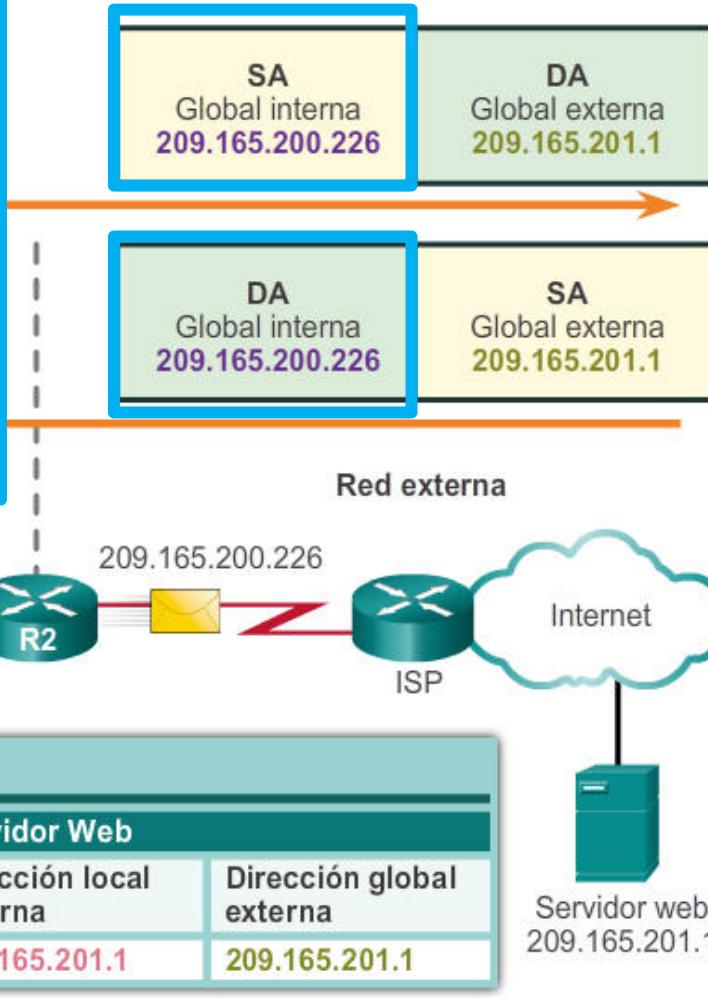
# NAT: Ejemplo

## Dirección global interna:

Al enviar tráfico de PC1 al servidor web en 209.165.201.1, R2 traduce la dirección local interna a una dirección global interna.

En este caso, R2 cambia la dirección IPv4 de origen de 192.168.10.10 a 209.165.200.226.

De acuerdo con la terminología de NAT, la dirección local interna 192.168.10.10 se traduce a la dirección global interna 209.165.200.226.



Fuente: Cisco NetAcad



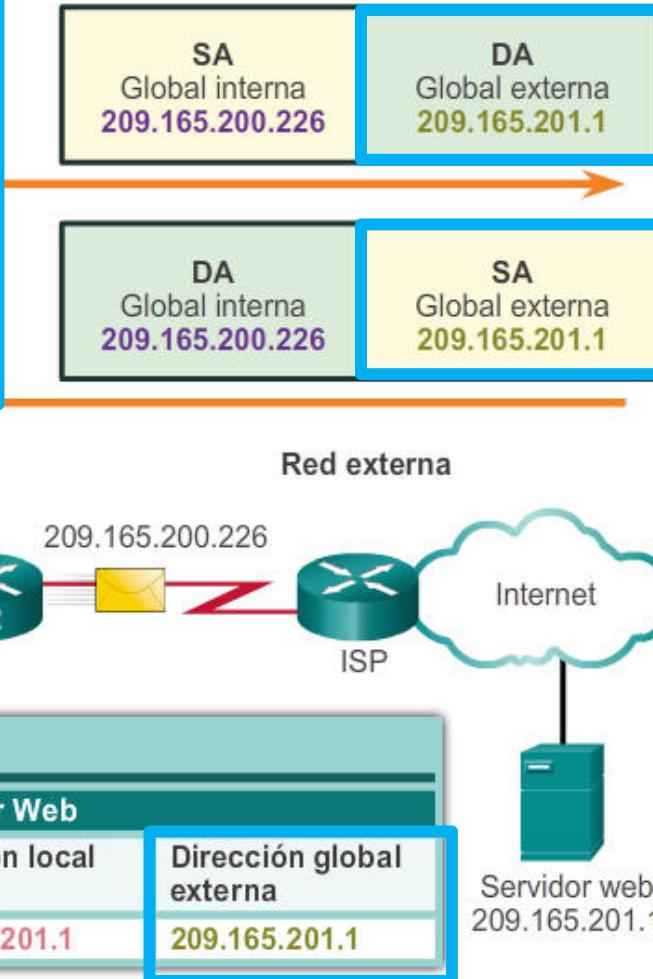
# NAT: Ejemplo

## Dirección global externa:

Dirección IPv4 enrutable globalmente y asignada a un host en Internet.

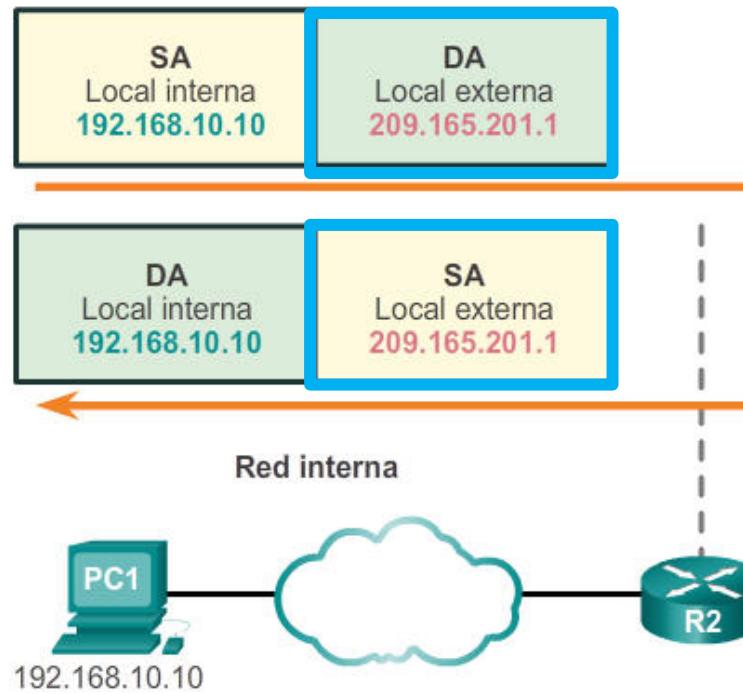
Por ejemplo, se puede llegar al servidor web en la dirección IPv4 209.165.201.1.

Generalmente, direcciones externas globales y locales son iguales.



Fuente: Cisco NetAcad

# NAT: Ejemplo



## Dirección local externa:

PC1 envía tráfico al servidor web en la dirección IPv4 209.165.201.1

Si bien es poco frecuente, esta dirección podría ser diferente de la dirección globalmente enrutable del destino.

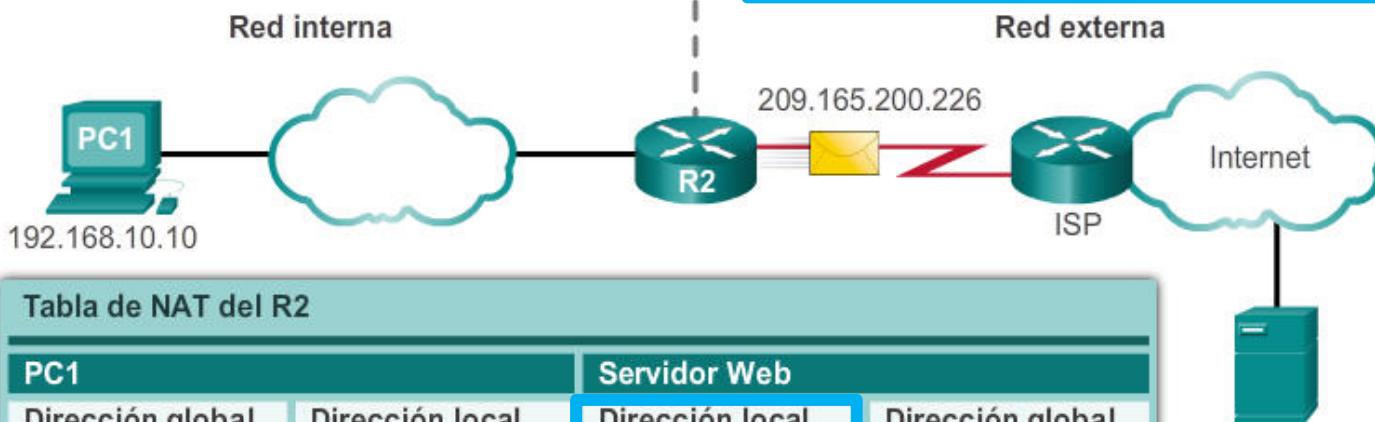


Tabla de NAT del R2

PC1	Servidor Web
Dirección global interna	Dirección local interna
209.165.200.226	192.168.10.10
Dirección local externa	Dirección global externa
	209.165.201.1

Fuente: Cisco NetAcad



# NAT: Ejemplo

## Explicaciones del proceso anterior

El router R2 se configuró para proporcionar NAT con un conjunto de direcciones públicas para asignar a los hosts internos.

- **Dirección local interna:** IPv4 192.168.10.10 se asignó a PC1. Esta es la dirección local interna de PC1.
- **Dirección global interna:** Cuando se envía el tráfico de PC1 al servidor web en 209.165.201.1, el R2 traduce la dirección local interna a una dirección global interna. En este caso, el R2 cambia la dirección IPv4 de origen de 192.168.10.10 a 209.165.200.226. De acuerdo con la terminología de NAT, la dirección local interna 192.168.10.10 se traduce a la dirección global interna 209.165.200.226.
- **Dirección global externa:** Dirección IPv4 enrutable globalmente y asignada a un host en Internet. Por ejemplo, se puede llegar al servidor web en la dirección IPv4 209.165.201.1. Generalmente, direcciones externas globales y locales son iguales.
- **Dirección local externa:** PC1 envía tráfico al servidor web en la dirección IPv4 209.165.201.1. Si bien es poco frecuente, esta dirección podría ser diferente de la dirección globalmente enrutable del destino.

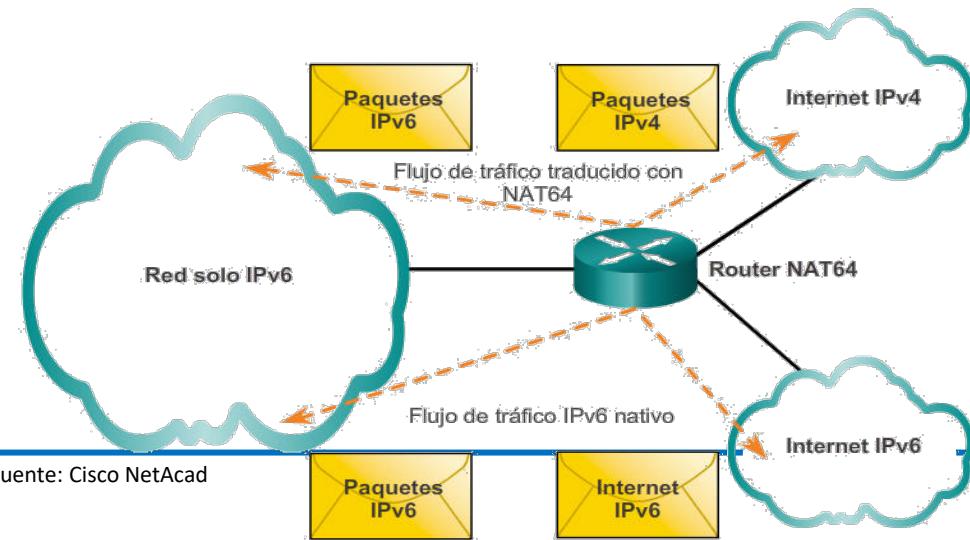
# Tipos de NAT

## Tipos de NAT en IPv4:

Existen tres formas de realizarlo:

1. NAT estático
2. NAT dinámico
3. NAT por traducción de Puerto o PAT

En **IPv6** también existe la posibilidad de hacerlo, mediante NAT64, que pretende ser **transitorio**. No vamos a comentar más sobre este mecanismo.

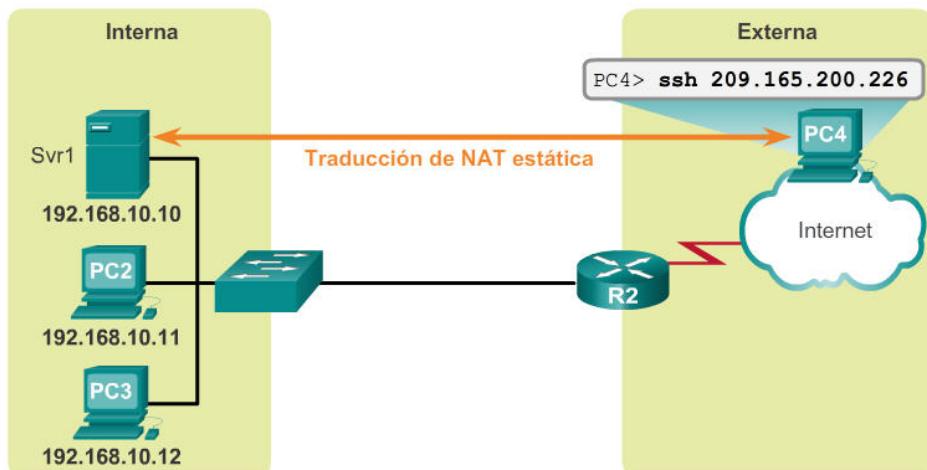


# NAT estático

**NAT estática** consiste en la **asignación 1:1** entre las direcciones **locales** y **globales**

- Administrador configura estas asignaciones y se mantienen constantes.
- Útil cuando se tiene que acceder a los servidores alojados en la red interna desde la red externa
- Administrador puede acceder mediante SSH a servidor de red interna indicando a cliente SSH la dirección global interna
- **NOTA:** Al cambiar direcciones, debe actualizar checksum cabecera IP

NAT estática	
Tabla de NAT estática	
Dirección local interna	Dirección global interna: direcciones a las que se puede llegar a través del R2
192.168.10.10	209.165.200.226
192.168.10.11	209.165.200.227
192.168.10.12	209.165.200.228

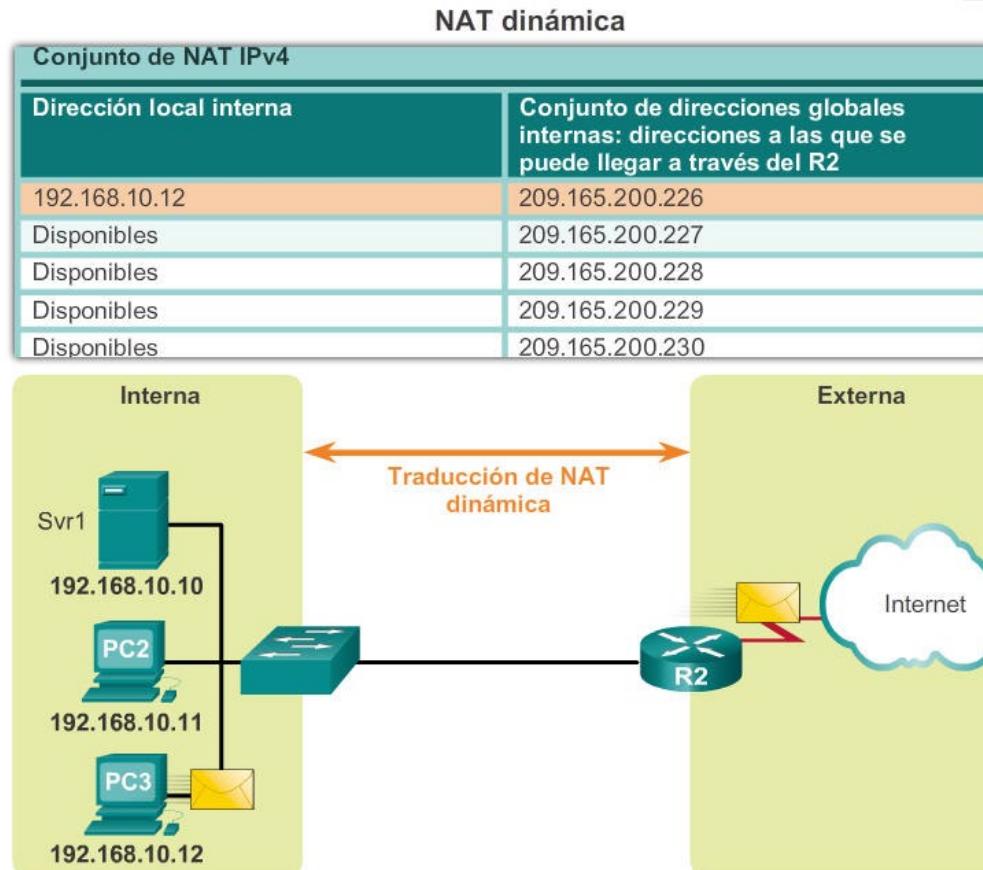


Fuente: Cisco NetAcad

# NAT dinámico

NAT dinámica utiliza **conjunto de direcciones públicas** que asigna según orden de llegada

- Cuando un dispositivo interno solicita acceso a una red externa, la NAT dinámica asigna una dirección IPv4 pública disponible del conjunto
- La NAT dinámica requiere que haya suficientes direcciones públicas disponibles para satisfacer la cantidad total de sesiones de usuario simultáneas
- **NOTA:** Al cambiar direcciones, debe actualizar checksum cabecera IP



# NAT traducción de Puerto (PAT)

**PAT asigna varias direcciones IPv4 privadas a única dirección IPv4 pública o a unas pocas direcciones**

- PAT utiliza la combinación puerto e IP de origen para poder seguir el tráfico que pertenece a cada cliente interno
- PAT también se conoce como NAT con sobrecarga
- Mediante el número de puerto, PAT también puede reenviar los paquetes de respuesta al dispositivo interno correcto
- PAT también valida que los paquetes entrantes se hayan solicitado: **añade un grado de seguridad a la sesión**
- **NOTA:** Al cambiar direcciones y puertos, debe actualizar checksum cabecera IP pero también el de la cabecera de transporte TCP/UDP

## Análisis de PAT de las computadoras a los servidores

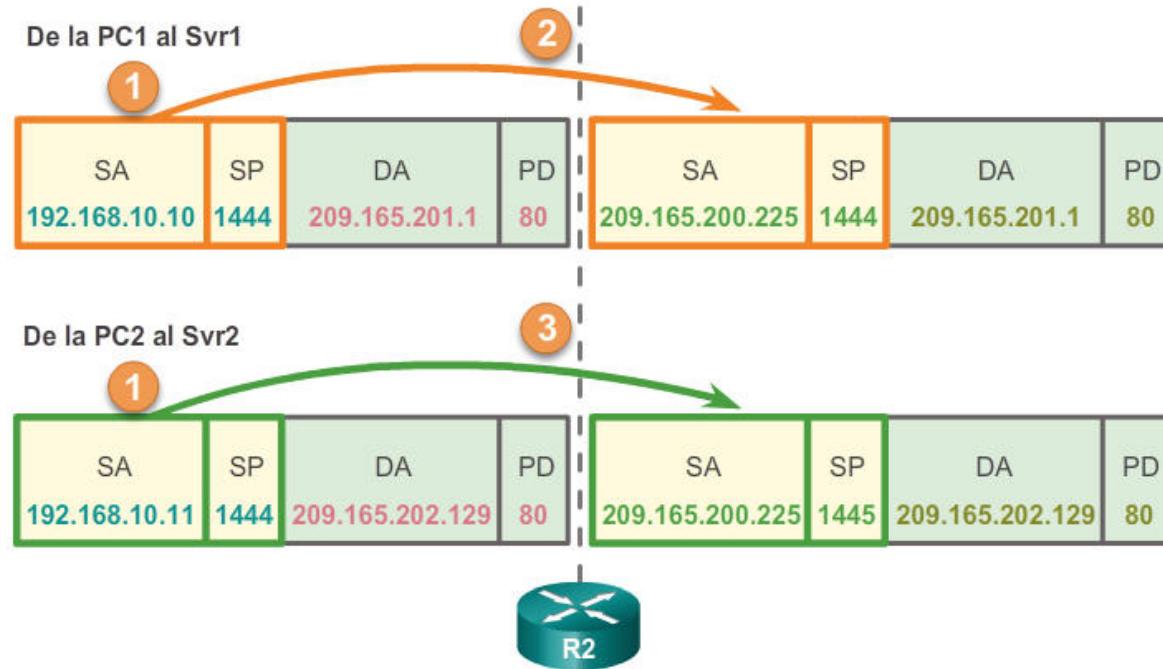


Tabla NAT

Dirección local interna	Dirección global interna	Dirección global externa	Dirección local externa
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1444	209.165.200.225:1445	209.165.202.129:80	209.165.202.129:80

Fuente: Cisco NetAcad

## Análisis de PAT de los servidores a las computadoras

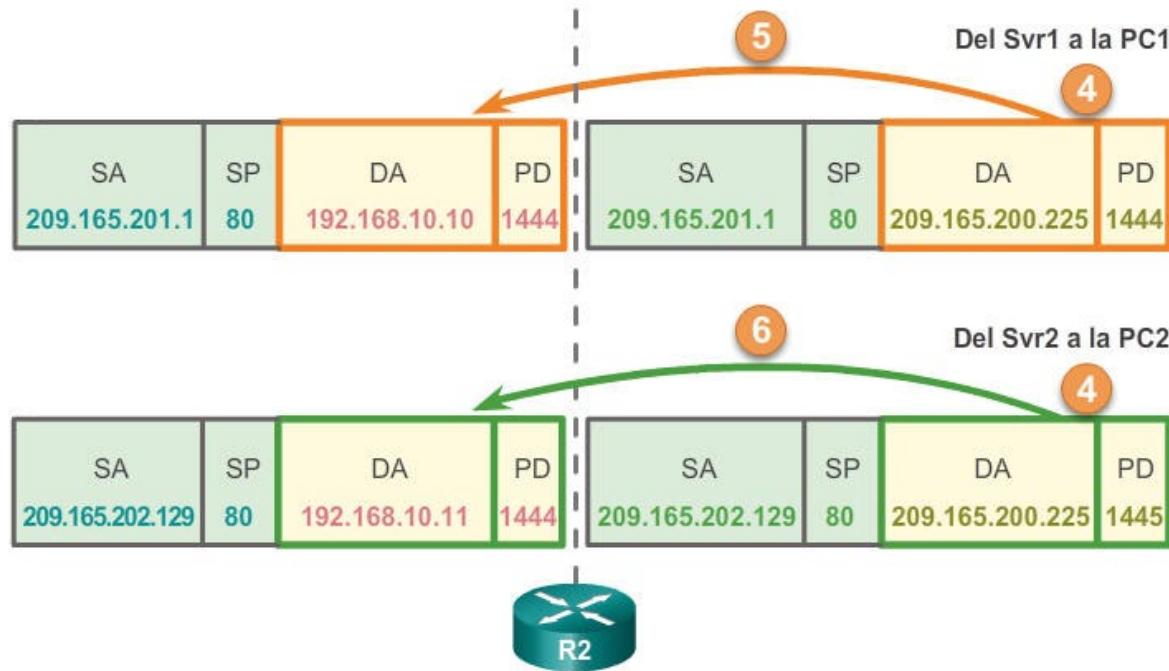


Tabla NAT

Dirección local interna	Dirección global interna	Dirección global externa	Dirección local externa
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1444	209.165.200.225:1445	209.165.202.129:80	209.165.202.129:80

Fuente: Cisco NetAcad

# Tabla NAPT dinámica en router doméstico

## Ejemplo de la tabla NAPT dinámica en un router doméstico

Transport LAN			WAN			NAPT	
Protocol	Port	IP Address	Port	IP Address	Port	IP Address	
<hr/>							
udp	27960	192.168.1.11	27960	212.142.28.226	60218	192.76.100.7	
tcp	1098	192.168.1.11	8000	205.149.163.62	60020	192.76.100.7	
tcp	1661	192.168.1.10	8000	192.160.165.89	60007	192.76.100.7	

El ordenador 192.168.1.11 está jugando al Quake 3 (puerto UDP 27960) y a la vez oye una emisora MP3 (puerto TCP 8000). Al mismo tiempo el ordenador 192.168.1.10 oye otra emisora MP3 (la dirección IP remota es diferente).

Ejemplo obtenido de: <http://adsl.internautas.org/sections.php?op=viewarticle&artid=1>

# Beneficios y desventajas de NAT

## Ventajas de la NAT

- Conserva el esquema de direccionamiento legalmente registrado.
- Aumenta la flexibilidad de las conexiones a la red pública.
- Proporciona coherencia a los esquemas de direccionamiento de red interna.
- Proporciona seguridad de red.

## Desventajas de la NAT

- Se deteriora el rendimiento.
- Se deteriora la funcionalidad de extremo a extremo.
- Se reduce el seguimiento IP de extremo a extremo.
- El tunnelling se torna más complicado.
- El inicio de las conexiones TCP puede interrumpirse.

# Videos: NAT

## NAT (Rojelio Montañeda)

NAT I

<https://www.youtube.com/watch?v=IFylgt7iVKA>

NAT II

[https://www.youtube.com/watch?v=eGgZCJA1X\\_E](https://www.youtube.com/watch?v=eGgZCJA1X_E)



## NAT (Aruma Digital)

NAT *Introducción y configuración estática*

<https://www.youtube.com/watch?v=sMNqYq3lqoQ>

## NAT (Sunny Classroom)

NAT - SNAT, DNAT, PAT & Port Forwarding

<https://www.youtube.com/watch?v=wg8Hosr20yw>

## Arquitectura TCP/IP

- Capa de transporte
  - Funcionalidades
  - Puertos
  - Protocolos
- Capa de aplicación
  - RTP
  - NAT
  - DHCP

# DHCP

- Protocolo de configuración dinámica de host (DHCP) proporciona a los clientes:
  - Dirección IP
  - Máscara de subred (IPv4) o longitud de prefijo (IPv6)
  - Dirección de gateway por defecto
  - Dirección de servidor DNS
- Está disponible tanto para IPv6 como para IPv4

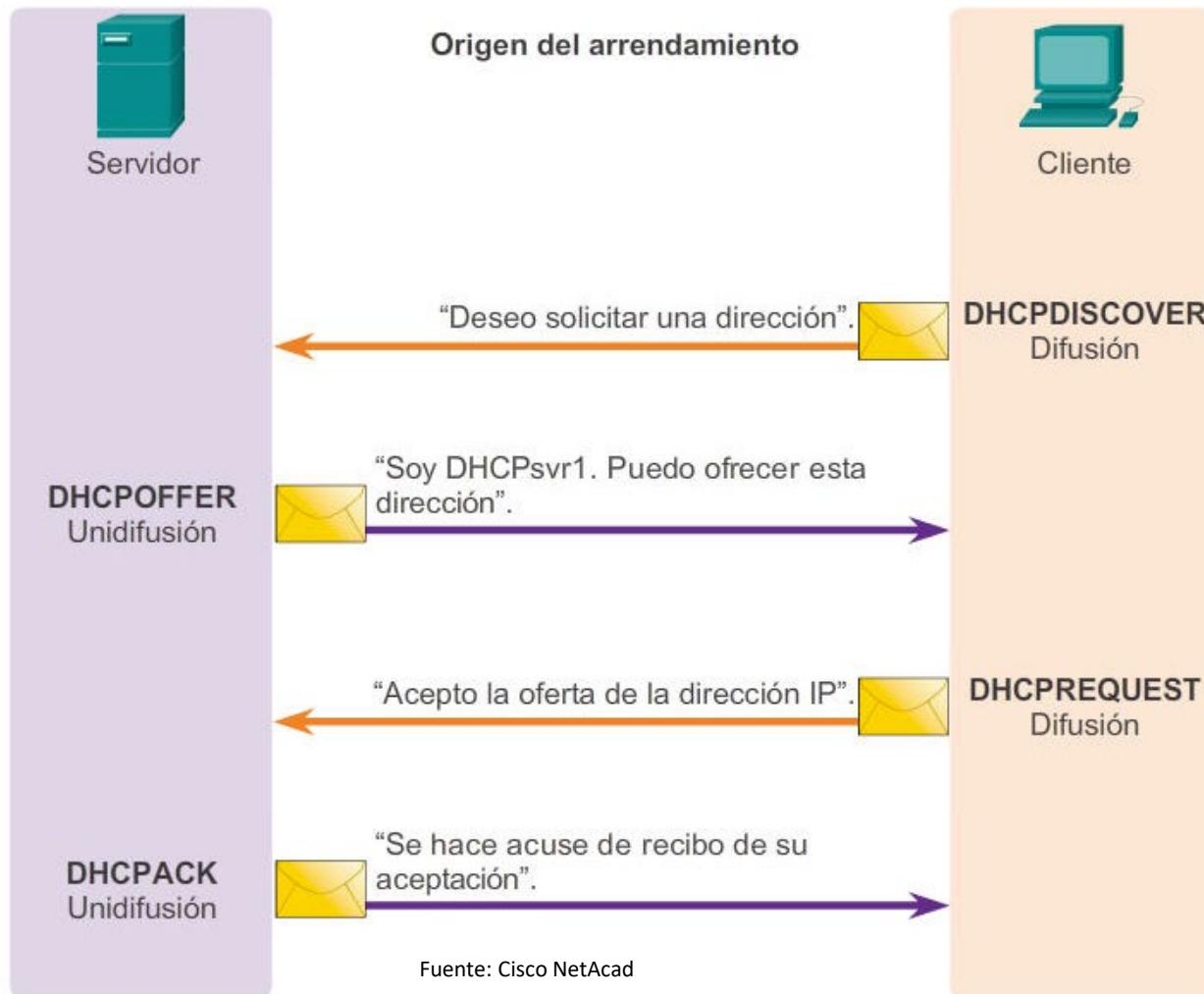
# DHCPv4

**DHCPv4** utiliza 3 métodos diferentes:

1. **Asignación manual:** Administrador asigna una dirección IPv4 preasignada al cliente, y DHCPv4 comunica solo la dirección IPv4 al dispositivo
2. **Asignación automática:** DHCPv4 asigna automáticamente una dirección IPv4 estática de forma permanente a un dispositivo y la selecciona de un conjunto de direcciones disponibles. No hay arrendamiento
3. **Asignación dinámica:** DHCPv4 asigna dinámicamente, o arrienda, una dirección IPv4 de un conjunto de direcciones durante un período limitado elegido por el servidor o hasta que el cliente ya no necesite la dirección. **El más utilizado**

# DHCPv4

## Funcionamiento de DHCPv4



# DHCPv4

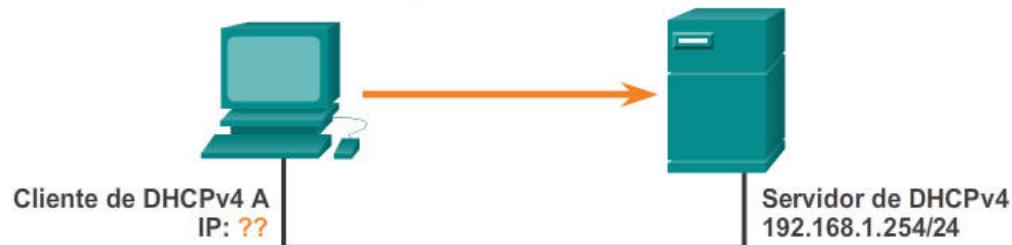
## Formato del mensaje DHCPv4

8	16	24	32		
Código OP (1)	Tipo de hardware (1)	Longitud de dirección de hardware (1)	Saltos (1)		
Identificador de transacción					
Segundos: 2bytes		Indicadores: 2bytes			
Dirección IP del cliente (CIADDR): 4bytes					
Su dirección IP (YIADDR): 4bytes					
Dirección IP del servidor (SIADDR): 4bytes					
Dirección IP del gateway (GIADDR): 4bytes					
Dirección de hardware del cliente (CHADDR): 16bytes					
Nombre del servidor (SNAME): 64bytes					
Nombre del archivo de arranque: 128bytes					
Opciones de DHCP: variable					



# DHCPv4

Mensaje Discover DHCPv4



Trama de Ethernet	IP	UDP	DHCPDISCOVER
MAC de origen: MAC A MAC de destino: FF:FF:FF:FF:FF:FF	IP de origen: 0.0.0.0 IP de destino: 255.255.255.255	UDP 67	CIADDR: 0.0.0.0 GIADDR: 0.0.0.0 Máscara: 0.0.0.0 CHADDR: MAC A
<p>MAC: dirección de control de acceso a los medios CIADDR: dirección IP del cliente GIADDR: dirección IP del gateway CHADDR: dirección de hardware del cliente</p>			

El cliente de DHCP envía una difusión IP dirigida con un paquete DHCPDISCOVER. En este ejemplo, el servidor de DHCP se encuentra en el mismo segmento y capta esta solicitud. El servidor advierte que el campo GIADDR está en blanco, de manera que el cliente está en el mismo segmento. El servidor también observa la dirección de hardware del cliente en el paquete de solicitud.

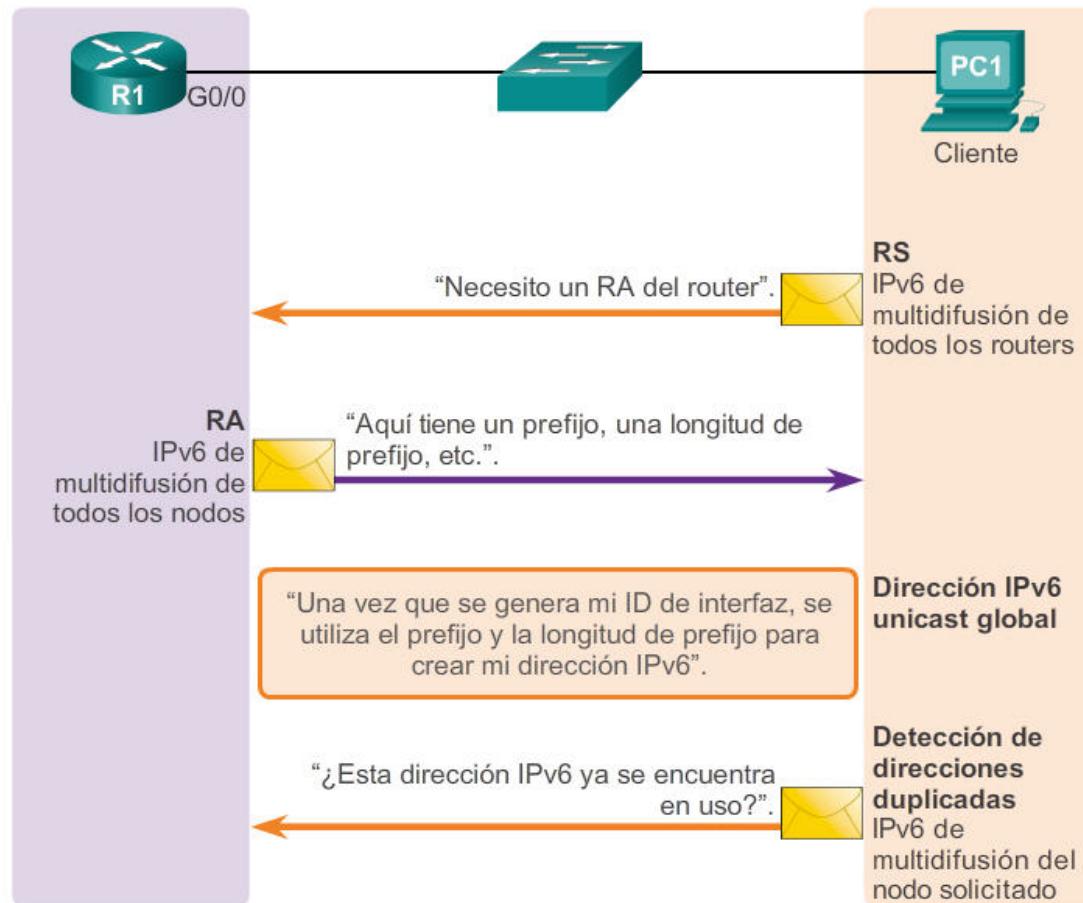
# DHCPv4

**DHCPv6** utiliza 3 métodos diferentes:

1. **SLAAC (Configuración automática de dirección sin estado):** Un dispositivo puede obtener una dirección IPv6 de unidifusión global sin los servicios de un servidor de DHCPv6
2. **DHCP sin estado:** El servidor no mantiene información de estado del cliente (es decir, una lista de direcciones IPv6 asignadas y disponibles). El servidor de DHCPv6 sin estado solo proporciona parámetros de configuración para los clientes, no direcciones IPv6.
3. **DHCP con estado:** Esta opción es la más similar a DHCPv4.

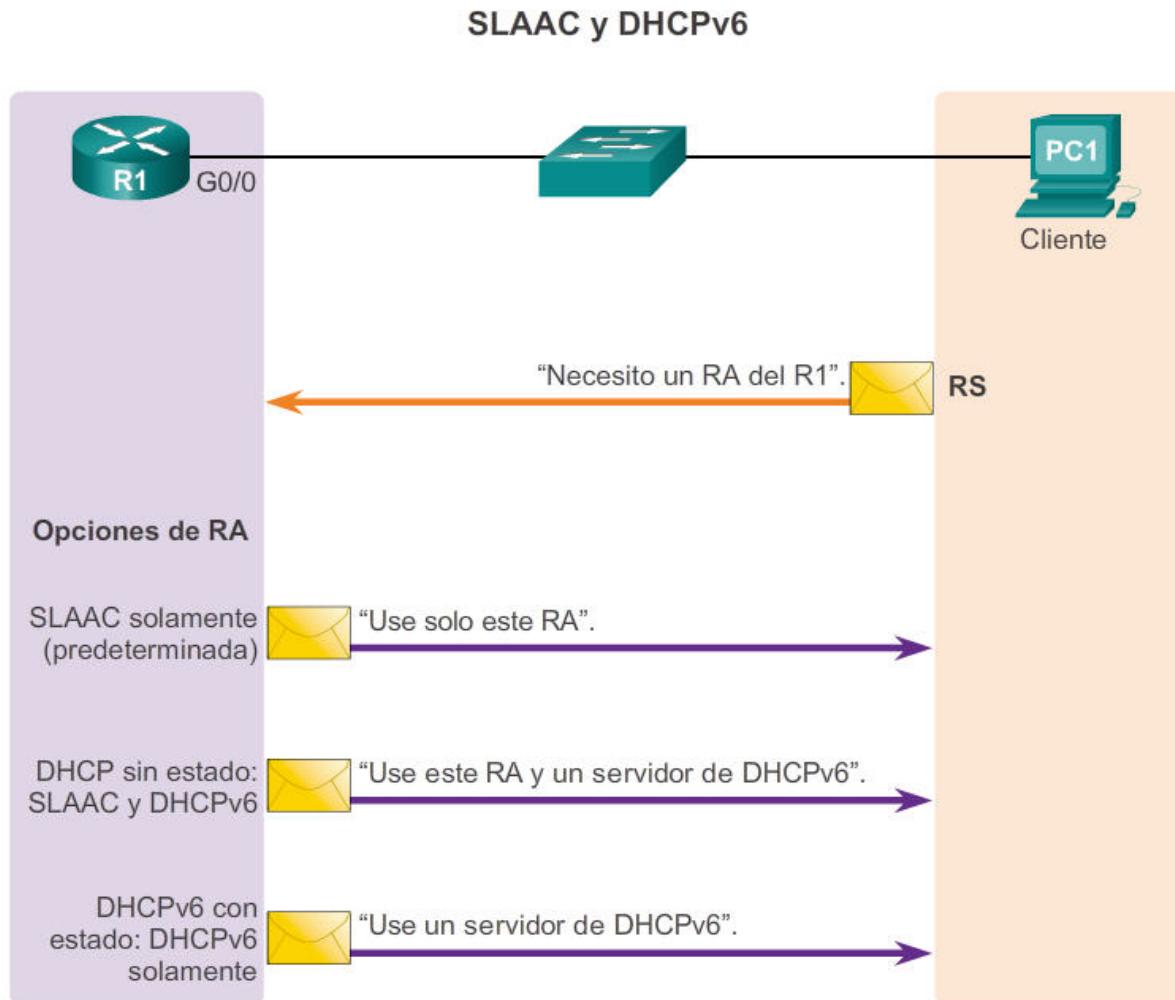
# DHCPv6 – SLAAC

El cliente realiza la detección de direcciones duplicadas



Fuente: Cisco NetAcad

# DHCPv6 – SLAAC y DHCPv6



# Videos: DHCP

## DHCP (Rojelio Montañeda)

*Resolución inversa de direcciones: protocolo DHCP*

[https://www.youtube.com/watch?v=r\\_8YCvfcNM4](https://www.youtube.com/watch?v=r_8YCvfcNM4)



*Ataques relacionados con DHCP*

<https://www.youtube.com/watch?v=mmuvHcGK-8M>

## DHCP (Aruma Digital)

*DHCP Proceso y Agente relay*

<https://www.youtube.com/watch?v=ARa0hZgWRxY>