

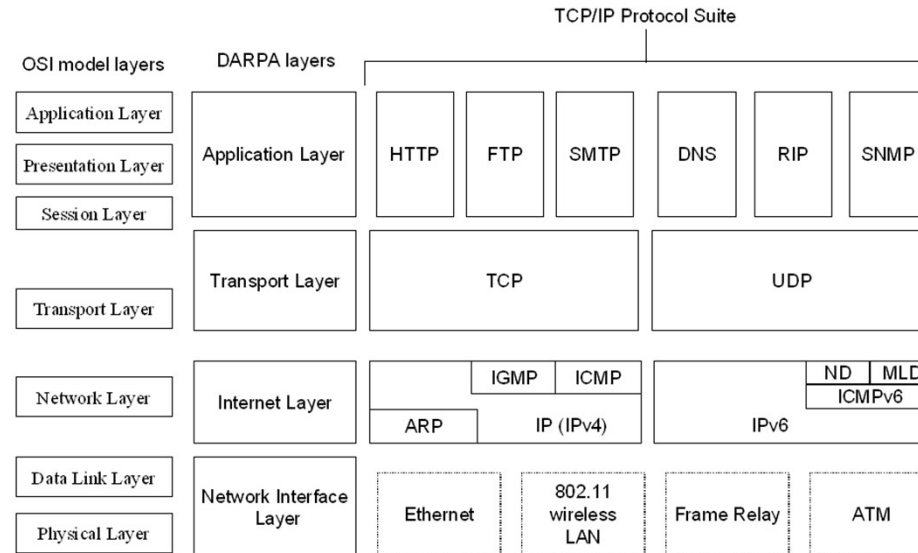
The background features abstract green geometric shapes, including triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

TCP/IP

Sergio Nicolás Santana Sánchez, ITESO - 2018

TCP/IP

- Conjunto de estándares TCP/IP, es un grupo de protocolos de comunicación utilizados para interconectar dispositivos de red en Internet.
- TCP/IP también se puede usar como un protocolo de comunicaciones en una red privada (una intranet).
- Estándares abiertos, proveedor neutral: patrón exitoso para la infraestructura.



TCP/IP

- ▶ Creada en los 70s por DARPA, para su uso en ARPANET, una red de área amplia que precedió a Internet.
- ▶ TCP / IP se diseñó originalmente para el sistema operativo Unix, y se ha integrado en todos los sistemas operativos posteriores.
- ▶ El modelo de TCP / IP y sus protocolos relacionados ahora son mantenidos por el Grupo de trabajo de ingeniería de Internet IETF. <https://www.ietf.org/>

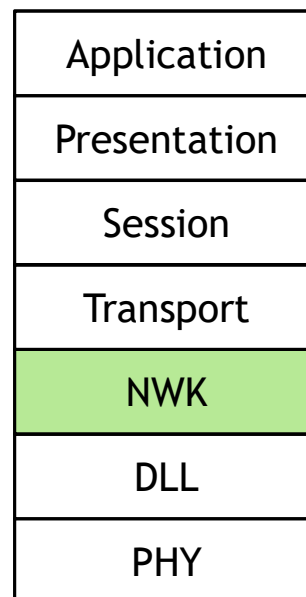


The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern, layered effect. The shapes are positioned on the left and right sides of the slide, framing the central text.

Internet Protocol - IP

IP

- ▶ Internet, es la colección de redes que permite la comunicación entre la mayoría de las instituciones de investigación, universidades y muchas otras organizaciones de todo el mundo.
- ▶ IP(Internet Protocol), es el protocolo primario de Capa 3 en el Stack de Internet.
- ▶ IP representa el corazón del conjunto de protocolos que componen el Internet.
- ▶ IP es definido por el estándar RFC 791.

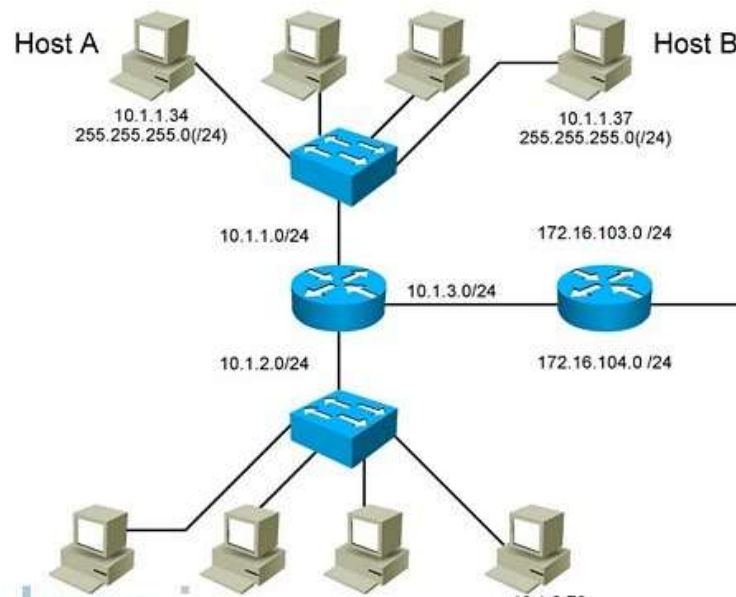


Internet Protocol - IP



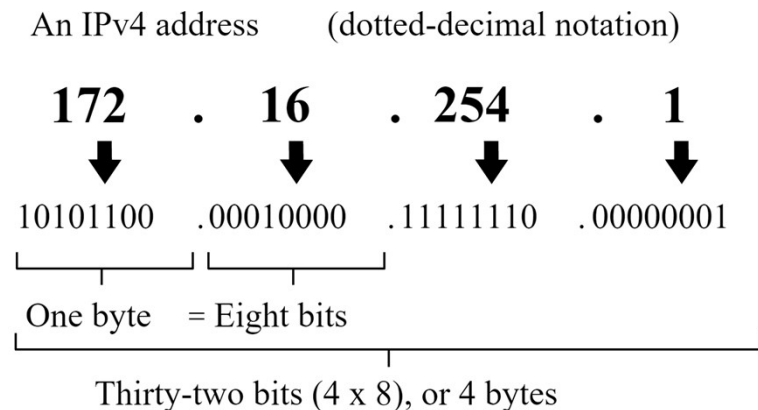
IP

- Su trabajo principal es encontrar la mejor ruta a través de Internet desde el host origen hasta el host destino.
- Cada enrutador de la red verifica esta dirección IP para determinar dónde reenviar el mensaje.



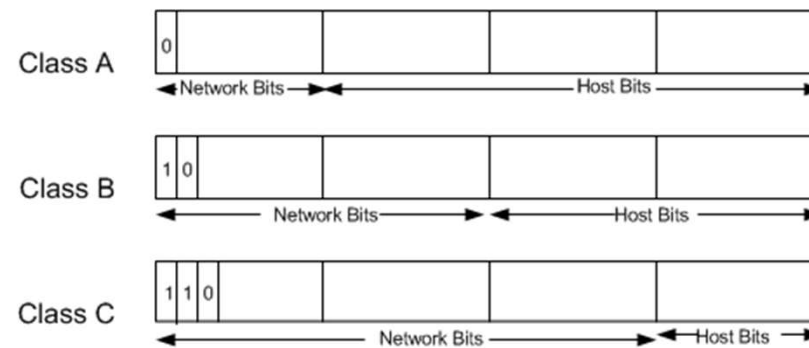
Direcciones IP

- ▶ Las direcciones IP son únicas en el mundo.
- ▶ Son números de 32 bits asignados por el Centro de información de red (NIC - Network Information Center).
- ▶ Las direcciones únicas a nivel mundial permiten que las redes IP de cualquier parte del mundo se comuniquen entre sí.
- ▶ Se escriben en la notación decimal de puntos, que consta de los cuatro octetos de la dirección expresados individualmente en números decimales y separados por puntos.
 - ▶ Por ejemplo, la dirección IP 192.0.2.235 representa el número decimal de 32 bits 2886794753, que en formato hexadecimal es 0xAC10FE01 y en binario 1010110000010000111111000000001b.



Direcciones IP

- Una dirección IP se divide en dos partes. La primera parte designa la dirección de red, mientras que la segunda parte designa la dirección de host.
- El espacio de direcciones IP se divide en diferentes clases de red. Las redes de clase A están destinadas principalmente para su uso con algunas redes muy grandes, ya que proporcionan solo 8 bits para el campo de dirección de red. Las redes de clase B asignan 16 bits, y las redes de clase C asignan 24 bits para el campo de dirección de red. Sin embargo, las redes de clase C solo proporcionan 8 bits para el campo de host, por lo que la cantidad de hosts por red puede ser un factor limitante. En los tres casos, los bits más a la izquierda indican la clase de red.



Direcciones IP - Reservadas

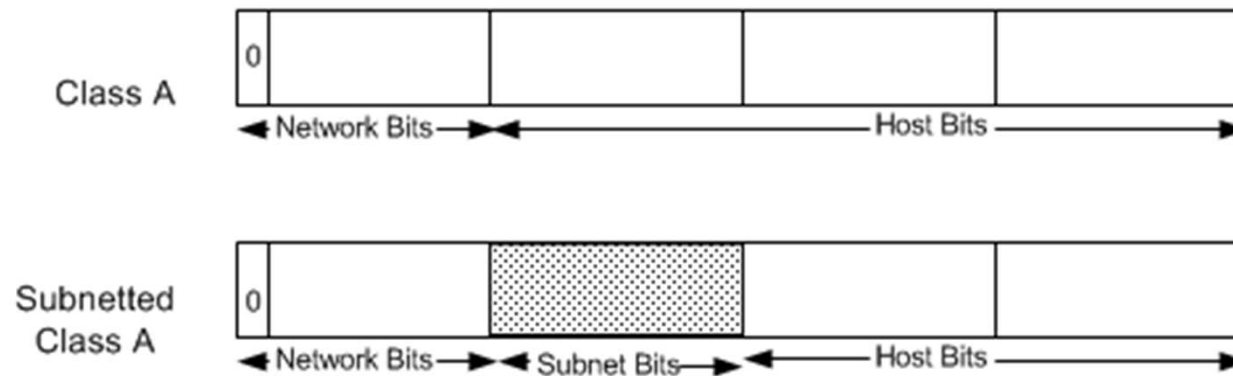
- ▶ En lugar de tener dispositivos dentro de una red doméstica o comercial, cada uno usando una dirección IP pública, de la que hay un suministro limitado, las direcciones IP privadas proporcionan un conjunto de direcciones totalmente independientes que aún permiten el acceso en una red pero sin ocupar un espacio de direcciones IP públicas.
 - ▶ La Autoridad de números asignados de Internet (IANA) reserva los siguientes bloques de direcciones IP para usar como direcciones IP privadas:
 - ▶ 10.0.0.0 a la 10.255.255.255
 - ▶ 172.16.0.0 a la 172.31.255.255
 - ▶ 192.168.0.0 a la 192.168.255.255
- ▶ El rango de la 127.0.0.0 a la 127.255.255.255 esta reservado para loopback.
- ▶ El rango de la 0.0.0.0 a la 0.255.255.255 esta reservado pero no se usa para nada.
- ▶ <https://www.lifewire.com/what-is-a-private-ip-address-2625970>

Direcciones IP - Especiales

- ▶ Existen algunas direcciones IP especiales:
- ▶ **IP 127.0.0.1**
 - ▶ Dirección reservada en todos los host IP para la función loopback.
- ▶ **IP: 0.0.0.0**
 - ▶ Dirección de IP cuando el host no esta conectado a la red.
- ▶ **IP: 10.0.0.1**
 - ▶ Dirección por defecto asignada a algunos enrutadores CISCO

IP - Subnets

- Las redes IP también se pueden dividir en unidades más pequeñas llamadas subredes o “subnets”. Las subredes proporcionan flexibilidad adicional para el administrador de red.



IP - Subnets

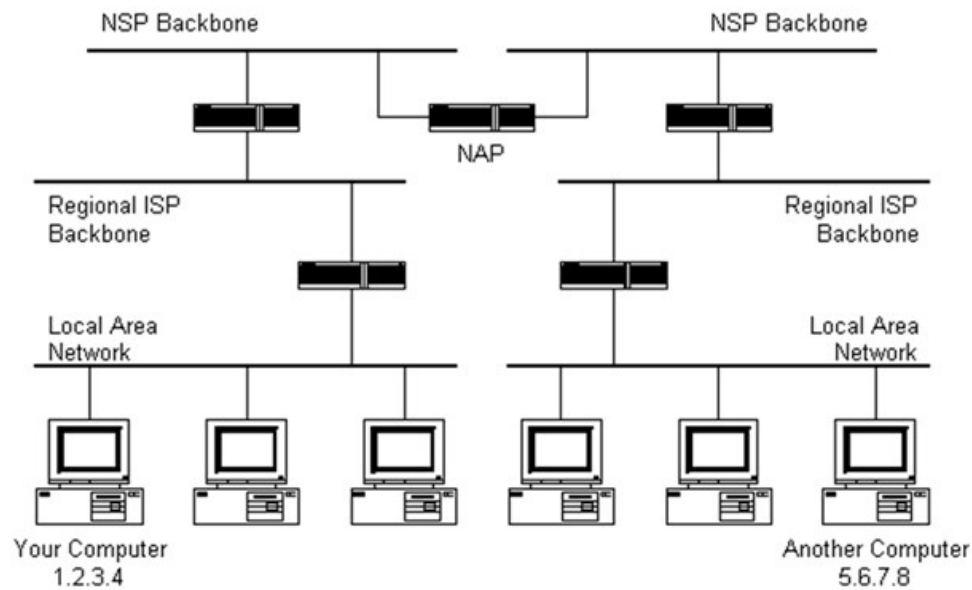
- Cada subnet cuenta con una dirección de subnet y una dirección broadcast de la subnet.

Address:	192.168.0.1	11000000.10101000.00000000	.00000001	
Netmask:	255.255.255.0 = 24	11111111.11111111.11111111	.00000000	
Wildcard:	0.0.0.255	00000000.00000000.00000000	.11111111	
=>				
Network:	192.168.0.0/24	11000000.10101000.00000000	.00000000	(Class C)
Broadcast:	192.168.0.255	11000000.10101000.00000000	.11111111	
HostMin:	192.168.0.1	11000000.10101000.00000000	.00000001	
HostMax:	192.168.0.254	11000000.10101000.00000000	.11111110	

- <http://jodies.de/ipcalc>

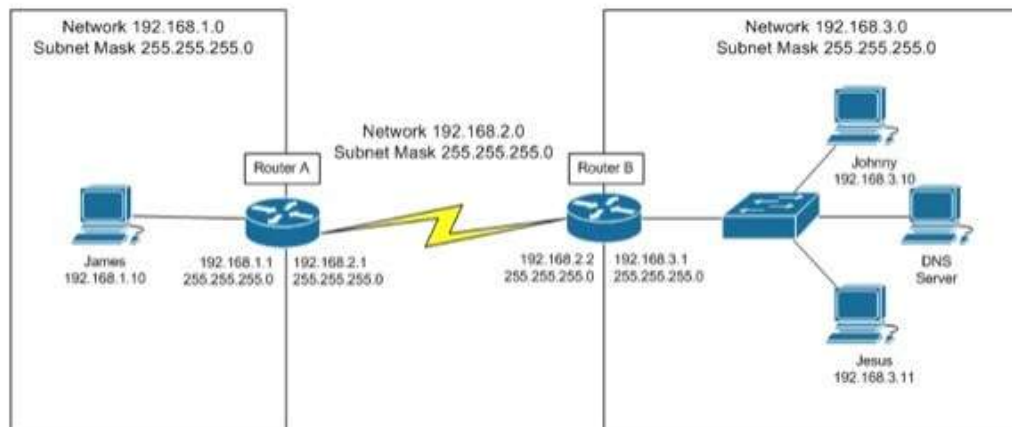
Routing en redes IP

- Los enrutadores dentro de Internet están organizados jerárquicamente. Algunos enrutadores se usan para mover información a través de un grupo particular de redes bajo la misma autoridad y control administrativo. (Dicha entidad se denomina sistema autónomo).



Routing en redes IP

- Los protocolos de enrutamiento utilizados con IP son de naturaleza dinámica. El enrutamiento dinámico requiere que el software en los dispositivos de enrutamiento calcule las rutas.
- Los algoritmos de enrutamiento dinámico se adaptan a los cambios en la red y automáticamente seleccionan las mejores rutas.
- A diferencia del enrutamiento dinámico, el enrutamiento estático requiere que el administrador de la red establezca rutas. Las rutas estáticas no cambian hasta que el administrador de la red las cambia.

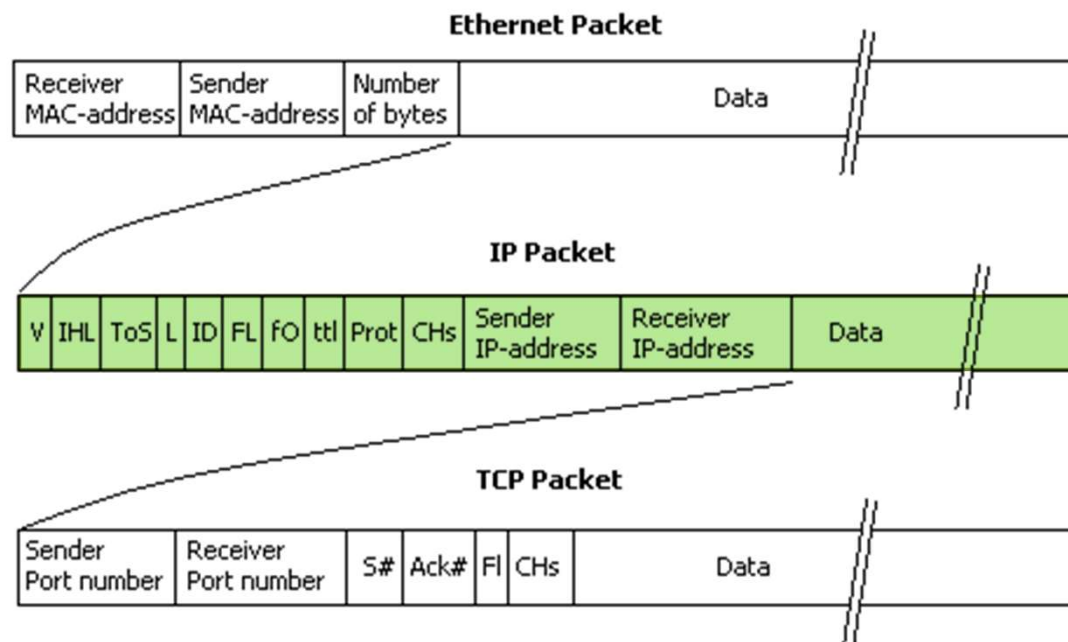


Routing en redes IP

- El enrutamiento de IP especifica que los datagramas de IP viajan a través de una red de un salto de enrutador a la vez.
- La ruta completa no se conoce al comienzo del viaje. En cambio, en cada parada, el siguiente salto del enrutador se determina al hacer coincidir la dirección de destino dentro del datagrama con una entrada en la tabla de enrutamiento del nodo actual.
- La participación de cada nodo en el proceso de enrutamiento consiste solamente en reenviar paquetes basados en información interna.

Destination	Gateway
169.254.0.0	0.0.0.0
192.168.200.0	0.0.0.0
192.168.200.1	192.168.200.7
192.168.200.4	192.168.200.4
192.168.200.7	192.168.200.7

La trama de IP



El header de IP

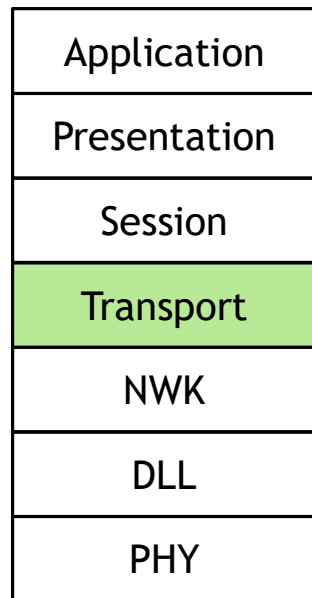
0	4	8	16	19	31
Version	Header Length	Service Type	Total Length		
Identification			Flags	Fragment Offset	
TTL	Protocol		Header Checksum		
Source IP Addr					
Destination IP Addr					
Options				Padding	

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

Transmission Control Protocol- TCP

TCP

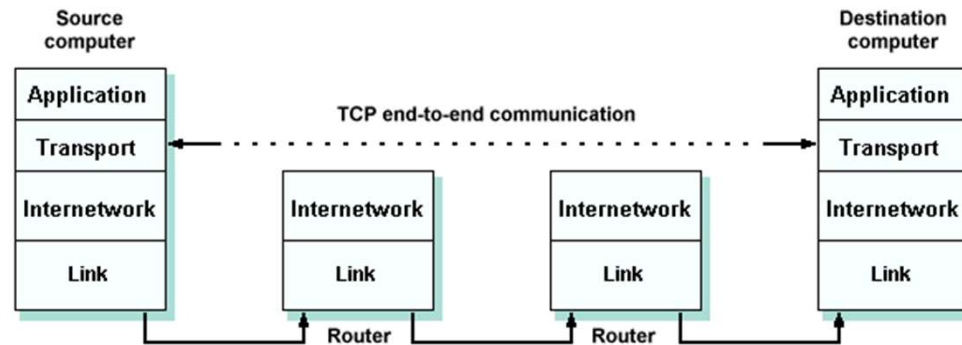
- TCP(Transmission Control Protocol), es un protocolo de transporte orientado a la conexión. Es una implementación de la capa 4 del Modelo OSI.



← Transmission Control Protocol - TCP

TCP

- ▶ Mediante el uso de números de secuencia y mensajes de acuse de recibo (Acks), TCP puede proporcionar un nodo de envío con información de entrega sobre los paquetes transmitidos a un nodo de destino.
- ▶ Cuando se pierden datos en tránsito desde el origen hasta el destino, TCP puede retransmitir los datos hasta que se alcanza una condición de tiempo de espera o hasta que se alcanza la entrega exitosa.
- ▶ TCP también puede reconocer mensajes duplicados y los descartará adecuadamente.



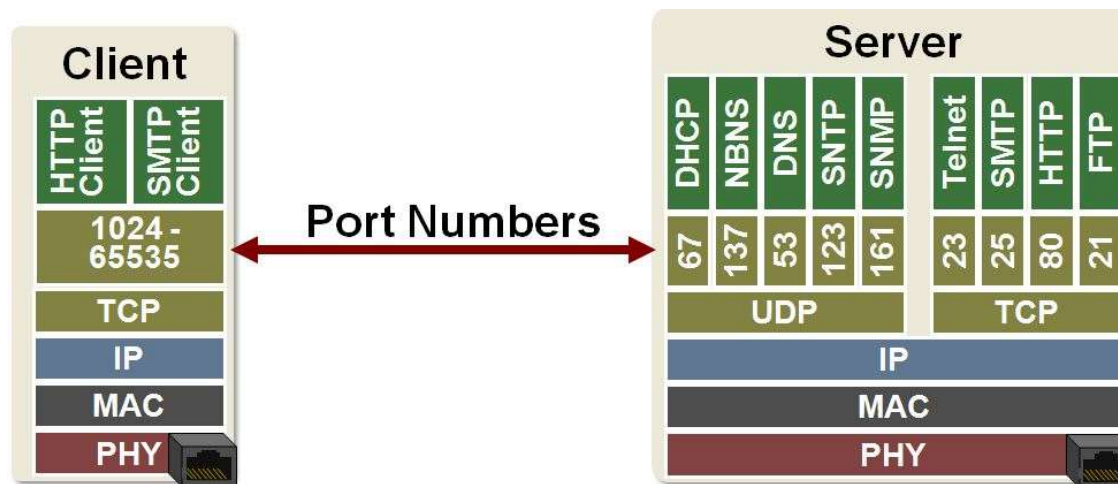
<http://www.technologyuk.net/telecommunications/internet/transport-layer-protocols.shtml>

TCP

- ▶ Si el dispositivo que envía está transmitiendo demasiado rápido para el dispositivo receptor, TCP puede emplear mecanismos de control de flujo para ralentizar la transferencia de datos.
- ▶ TCP también puede comunicar información de entrega a los protocolos y aplicaciones de capa superior.
- ▶ TCP se especifica en el estándar RFC 793.

Puertos TCP

- Los puertos se utilizan para identificar procesos que se ejecutan en las aplicaciones en un host.



<http://microchipdeveloper.com/tcpip:tcp-ip-ports>

Puertos TCP - Reservados

- ▶ Los números de puerto se asignan de varias maneras, en función de tres rangos (RFC6335):
 - ▶ Puertos del sistema (0-1023).
 - ▶ Puertos de usuario (1024-49151).
 - ▶ Puertos dinámicos y / o privados (49152-65535).
- ▶ Los procedimientos de registro para nombres de servicio y números de puerto se describen en [RFC6335].
- ▶ Los puertos asignados tanto del sistema como del usuario NO DEBEN ser utilizados sin o antes del registro de IANA.

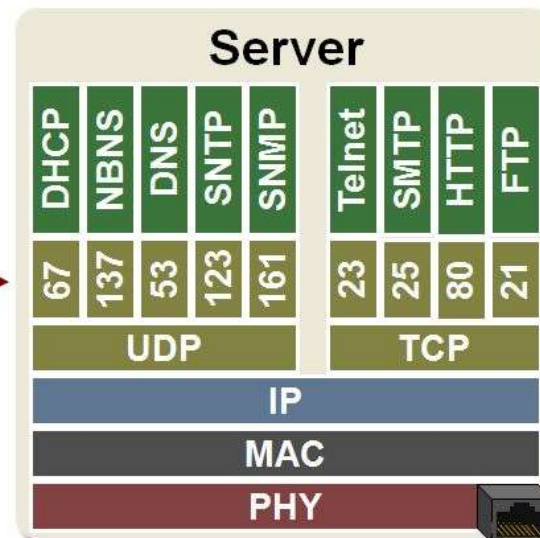


Internet Assigned Numbers Authority

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

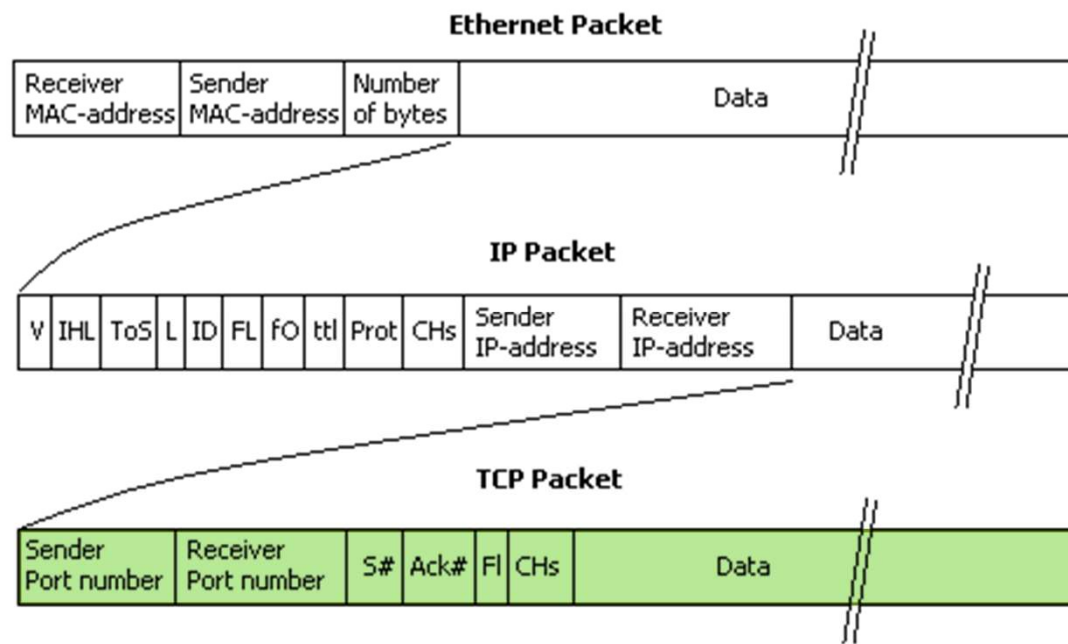
Puertos TCP - Reservados

Puertos bien conocidos →



https://microchip.wdfiles.com/local--files/tcpip:well-known-ports/well_known_ports.JPG

La trama de TCP



Encabezado TCP

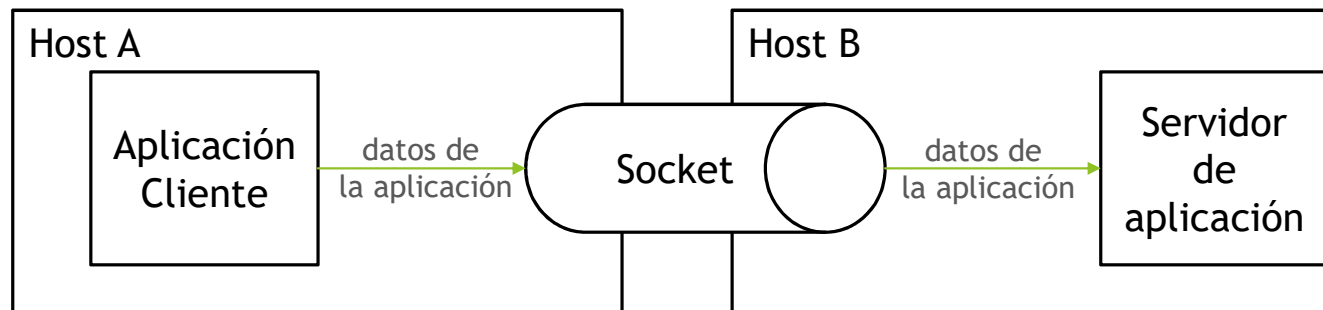
Transmission Control Protocol (TCP) Header 20-60 bytes

source port number 2 bytes				destination port number 2 bytes			
sequence number 4 bytes							
acknowledgement number 4 bytes							
data offset 4 bits	reserved 3 bits			control flags 9 bits			window size 2 bytes
checksum 2 bytes				urgent pointer 2 bytes			
optional data 0-40 bytes							

<https://www.lifewire.com/tcp-headers-and-udp-headers-explained-817970>

Sockets

- ▶ Un socket es un concepto de software para una conexión. Los sockets permiten que las aplicaciones se conecten a una red TCP/IP.
- ▶ Una interfaz de programación de socket proporciona las rutinas necesarias para la comunicación entre procesos de aplicaciones, ya sea en el sistema local o distribuidas en un entorno de red distribuido, basado en TCP/IP.
- ▶ Una vez que se establece una conexión punto a punto, se utiliza un descriptor de socket para identificar de manera única la conexión. El descriptor de socket en sí mismo es un valor numérico específico de la tarea.



TCP/UDP Sockets

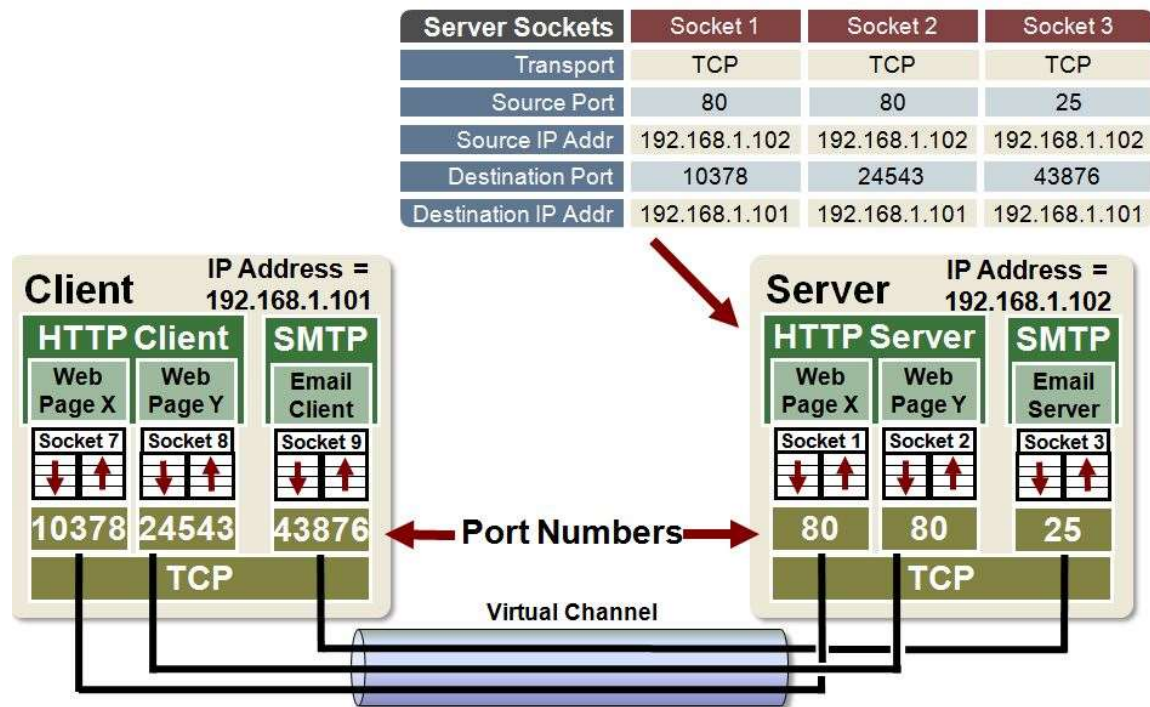
- ▶ El extremo de una conexión punto a punto de una aplicación de red distribuida basada en TCP/IP descrito por un socket está definido de forma única por:
 - ▶ Su dirección de Internet, por ejemplo, 127.0.0.1 (en una red IPv4).
 - ▶ Protocolo de comunicación.
 - ▶ Protocolo de datagrama de usuario (UDP).
 - ▶ Protocolo de control de transmisión (TCP).
 - ▶ Numero de puerto que identifica una aplicación.
 - ▶ Distinguimos entre puertos "bien conocidos", por ejemplo, el puerto 23 para Telnet y puertos definidos por el usuario.

TCP/UDP Sockets

Source port numbers assigned by local host →	Server Sockets	Socket 1	Socket 2	Socket 3
	Transport	TCP	TCP	TCP
	Source Port	80	80	25
	Source IP Addr	192.168.1.102	192.168.1.102	192.168.1.102
	Destination Port	10378	24543	43876
Destination port numbers assigned by remote host →	Destination IP Addr	192.168.1.101	192.168.1.101	192.168.1.101
		↑ Web server web page X	↑ Web server web page Y	↑ Email server

<http://microchipdeveloper.com/tcpip:tcp-ip-socket-creation>

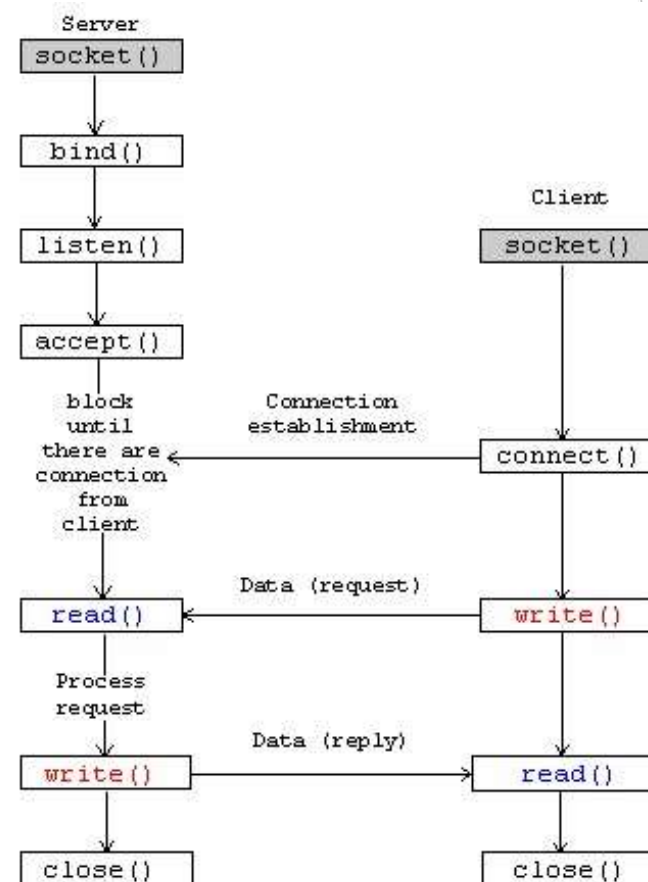
TCP/UDP Sockets



<http://microchipdeveloper.com/tcpip:tcp-ip-socket-creation>

Uso de Sockets

- El API de sockets es una interfaz de programación de Comunicación entre Procesos (IPC) proporcionada originalmente como parte del sistema operativo Berkeley UNIX.
- Los llamados a función necesarios para el Cliente y el Servidor pueden ser algo diferentes, pero ambos implican la construcción básica de un socket.
- Un socket es un extremo de un canal de comunicación entre procesos. Los dos procesos establecen su propio socket.



The background features abstract green geometric shapes. On the left, a solid green trapezoid points upwards. On the right, a complex arrangement of overlapping translucent green triangles and polygons creates a layered, crystalline effect. The text 'Stack LwIP' is centered in a green, sans-serif font.

Stack LwIP

LwIP

- ▶ lwIP es una pequeña implementación independiente del conjunto de protocolos TCP/IP que ha sido desarrollado inicialmente por Adam Dunkels.
- ▶ El objetivo de la implementación de lwIP TCP/IP es reducir el uso de recursos al mismo tiempo que se tiene un stack de TCP a escala completa. Esto hace que lwIP sea adecuado para su uso en sistemas embebidos con decenas de kilobytes de RAM libre y espacio para alrededor de 40 kilobytes de ROM de código.

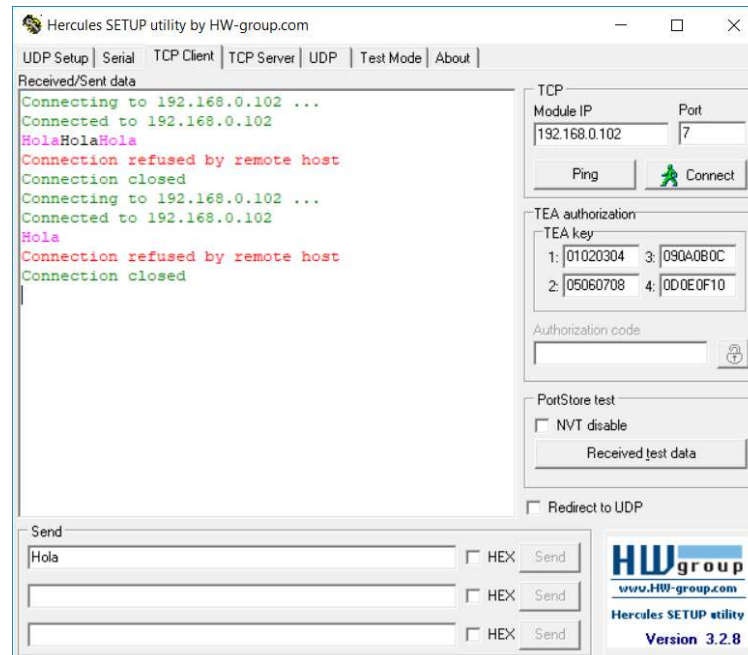


Sockets en LwIP

```
#define accept(s, addr, addrlen) lwip_accept(s,addr,addrlen)
#define bind(s, name, namelen) lwip_bind(s,name,namelen)
#define shutdown(s, how) lwip_shutdown(s,how)
#define getpeername(s, name, namelen) lwip_getpeername(s,name,namelen)
#define getsockname(s, name, namelen) lwip_getsockname(s,name,namelen)
#define setsockopt(s, level, optname, opval, optlen) lwip_setsockopt(s,level,optname,opval,optlen)
#define getsockopt(s, level, optname, opval, optlen) lwip_getsockopt(s,level,optname,opval,optlen)
#define closesocket(s) lwip_close(s)
#define connect(s, name, namelen) lwip_connect(s,name,namelen)
#define listen(s, backlog) lwip_listen(s,backlog)
#define recv(s, mem, len, flags) lwip_recv(s,mem,len,flags)
#define recvfrom(s, mem, len, flags, from, fromlen) lwip_recvfrom(s,mem,len,flags,from,fromlen)
#define send(s, dataptr, size, flags) lwip_send(s,dataptr,size,flags)
#define sendmsg(s, message, flags) lwip_sendmsg(s,message,flags)
#define sendto(s, dataptr, size, flags, to, tolen) lwip_sendto(s,dataptr,size,flags,to,tolen)
#define socket(domain, type, protocol) lwip_socket(domain,type,protocol)
#define select(maxfdp1, readset, writeset, exceptset, timeout) lwip_select(maxfdp1,readset,writeset,exceptset,timeout)
#define ioctlsocket(s, cmd, argp) lwip_ioctl(s,cmd,argp)
#define read(s, mem, len) lwip_read(s,mem,len)
#define write(s, dataptr, len) lwip_write(s,dataptr,len)
#define writev(s, iov, iovcnt) lwip_writev(s,iov,iovcnt)
#define close(s) lwip_close(s)
#define fcntl(s, cmd, val) lwip_fcntl(s,cmd,val)
#define ioctl(s, cmd, argp) lwip_ioctl(s,cmd,argp)
#define inet_ntop(af, src, dst, size)
#define inet_pton(af, src, dst)
```

Testing Sockets on the PC

- Let's use Hercules



<https://www.hw-group.com/software/hercules-setup-utility>