Aplicaciones y Servicios en Redes

Tema 01. Aplicaciones y Servicios Básicos

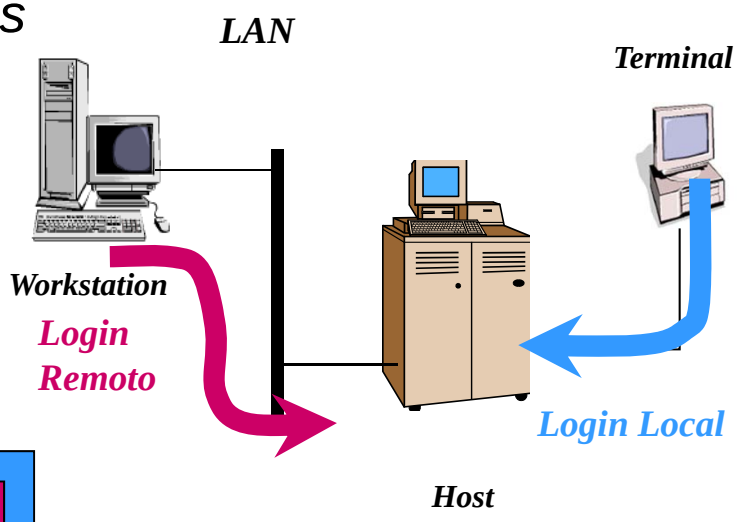
Alberto Eloy García Gutiérrez

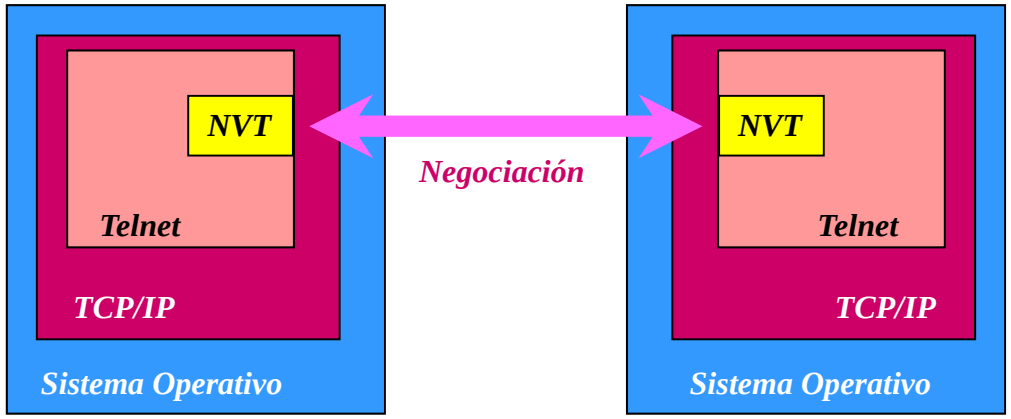
Luis Sánchez González

DPTO. DE INGENIERÍA DE COMUNICACIONES

Capa de Aplicación: TELNET

- *Conexión TCP a servidores remotos (RFC 854, 855)*
- *Interfaz estándar para acceso a recursos*
- *No soporta entornos gráficos*
- *NVT (Network Virtual Terminal):*
  teclado + pantalla
- *Datos de 7 bits (se transmiten 8 bits)*
- *Semiduplex con buffer de línea*
- *Opciones negociadas por los extremos*





TELNET: Descripción

- TELNET es un protocolo basado en tres ideas:
  - *NVT(Network Virtual Terminal):* dispositivo imaginario que posee una estructura básica común a una amplia gama de terminales reales.
  - Una perspectiva simétrica de las terminales y los procesos.
  - Negociación de las opciones de la terminal: mecanismos "DO, DON'T, WILL, WON'T"("hazlo, no lo hagas, lo harás, no lo harás")
- Los dos hosts verifican que existe una comprensión mutua
- Completada la negociación inicial son capaces de trabajar en el nivel mínimo implementado por la NVT
- Pueden negociar opciones adicionales
- Sigue un modelo simétrico: cliente y servidor pueden proponer opciones adicionales.

TELNET: NVT(Network Virtual Terminat)

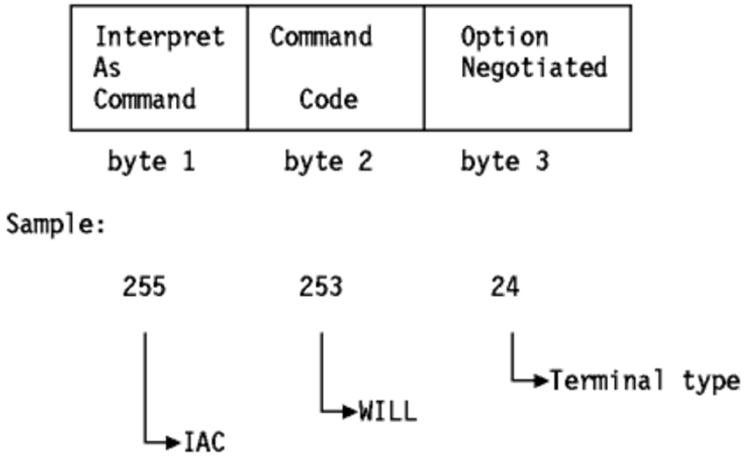
- La NVT cuenta con un monitor o "display" y un teclado.
  - El teclado produce datos de salida, que se envían por la conexión
  - El monitor recibe los datos de entrada que llegan.
- Los datos se representan en código ASCII de 7 bits, transmitido en bytes de 8 bits.
- La NVT es un dispositivo semi-duplex que opera en modo de buffer en línea.
- La NVT proporciona una función de eco local.
- Todas estas opciones pueden ser negociadas por los dos hosts.
  - Ejm.: se prefiere el eco local porque la carga de la red es inferior y el rendimiento superior pero existe la opción de usar el eco remoto, aunque no se le requiera a ningún host.

TELNET: Opciones

Num	Name	State	RFC	STD
255	Extended-Options-List	Standard	861	32
0	Binary Transmission	Standard	856	27
1	Echo	Standard	857	28
3	Suppress Go Ahead	Standard	858	29
5	Status	Standard	859	30
6	Timing Mark	Standard	860	31
34	Linemode	Draft	1184	
2	Reconnection	Proposed		
4	Approx Message Size Negotiation	Proposed		
7	Remote Controlled Trans and Echo	Proposed	726	
8	Output Line Width	Proposed		
9	Output Page Size	Proposed		
10	Output Carriage-Return Disposition	Proposed	652	
11	Output Horizontal Tabstops	Proposed	653	
12	Output Horizontal Tab Disposition	Proposed	654	
13	Output Formfeed Disposition	Proposed	655	
14	Output Vertical Tabstops	Proposed	656	
15	Output Vertical Tab Disposition	Proposed	657	
16	Output Linefeed Disposition	Proposed	658	
17	Extended ASCII	Proposed	698	

Num	Name	State	RFC	STD
18	Logout	Proposed	727	
19	Byte Macro	Proposed	735	
20	Data Entry Terminal	Proposed	1043	
21	SUPDUP	Proposed	736	
22	SUPDUP Output	Proposed	749	
23	Send Location	Proposed	779	
24	Terminal Type	Proposed	1091	
25	End of Record	Proposed	885	
26	TACACS User Identification	Proposed	927	
27	Output Marking	Proposed	933	
28	Terminal Location Number	Proposed	946	
29	TELNET 3270 Regime	Proposed	1041	
30	X.3 PAD	Proposed	1053	
31	Negotiate About Window Size	Proposed	1073	
32	Terminal Speed	Proposed	1079	
33	Remote Flow Control	Proposed	1372	
35	X Display Location	Proposed	1096	
39	TELNET Environment Option	Proposed	1572	
37	TELNET Authentication Option	Experimental	1416	

TELNET: Estructura de Comandos



- La comunicación es manejada por comandos internos, no accesibles a los usuarios
- Todos los comandos internos de TELNET son secuencias de 2 o 3 bytes

Command name	Code	Comments
SE	240	End of sub-negotiation parameters.
NOP	241	No operation.
Data Mark	242	The data stream portion of a synch. This should always be accompanied by a TCP urgent notification.
Break	243	NVT character BRK.
Go ahead	249	The GA signal.
SB	250	Indicates that what follows is sub-negotiation of the option indicated by the immediately following code.
WILL	251	Shows the desire to use, or confirmation that you are now using, the option indicated by the code immediately following.
WON'T	252	Shows the refusal to use, or to continue to use, the option indicated by the code immediately following.
DO	253	Requests that the other party uses, or confirms that you are expecting the other party to use, the option indicated by the code immediately following.
DON'T	254	Demands that the other party stop using, or confirms that you are no longer expecting the other party to use, the option indicated by the code immediately following.
IAC	255	Interpret As Command. Indicates that what follows is a TELNET command, not data.

TELNET: Negociación de opciones

- Se hace uso exclusivamente de comandos internos
- Debe existir un acuerdo mínimo
- Códigos de comando "WILL, WON'T, DO, DON'T"
- Sub-opciones: SB y SE llevan a cabo la sub-negociación.

Send	Reply	Meaning
DO transmit binary	WILL transmit binary	
DO window size	WILL window size	Can we negotiate window size?
SB Window Size 0 80 0 24 SE		Specify window size
DO terminal type	WILL terminal type	Can we negotiate terminal type?
SB terminal type SE		Send me your terminal characteristics.
	SB terminal type IBM-3278-2 SE	My terminal is a 3278-2
DO echo	WON'T echo	

TELNET: Comandos básicos

- Se proporciona una interfaz estándar
- Permite que comience una conexión
- Implementa una representación estándar para algunas funciones:

IP	**Interrumpir proceso**
AO	**Abortar la salida**
AYT	**¿Estás ahí?**
EC	**Borrar carácter**
EL	**Borrar línea**
SYNCH	**Sincronizar**

TELNET: Ejemplo de sesión

```
C:\>telnet pepe.tlmat.unican.es
Conectándose a pepe.tlmat.unican.es...No se puede abrir la conexión al host, en
puerto 23: Error en la conexión

C:\>telnet luna.tlmat.unican.es
Conectándose a luna.tlmat.unican.es...No se puede abrir la conexión al host, en
puerto 23.
No se ha podido establecer conexión ya que el equipo de destino ha denegado acti
vamente dicha conexión.


C:\>telnet luna.tlmat.unican.es
Conectándose a luna.tlmat.unican.es...


login: agarcia
Password:
Login incorrect
login: agarcia
Password:
Please wait...checking for disk quotas
(c)Copyright 1983-1996 Hewlett-Packard Co.,  All Rights Reserved.
(c)Copyright 1979, 1980, 1983, 1985-1993 The Regents of the Univ. of California
(c)Copyright 1980, 1984, 1986 Novell, Inc.
(c)Copyright 1986-1992 Sun Microsystems, Inc.
(c)Copyright 1985, 1986, 1988 Massachusetts Institute of Technology
(c)Copyright 1989-1993  The Open Software Foundation, Inc.
(c)Copyright 1986 Digital Equipment Corp.
(c)Copyright 1990 Motorola, Inc.
(c)Copyright 1990, 1991, 1992 Cornell University
(c)Copyright 1989-1991 The University of Maryland
(c)Copyright 1988 Carnegie Mellon University

                            RESTRICTED RIGHTS LEGEND
Use, duplication, or disclosure by the U.S. Government is subject to
restrictions as set forth in sub-paragraph (c)(1)(ii) of the Rights in
Technical Data and Computer Software clause in DFARS 252.227-7013.


                            Hewlett-Packard Company
                            3000 Hanover Street
                            Palo Alto, CA 94304 U.S.A.

Rights for non-DOD U.S. Government Departments and Agencies are as set
forth in FAR 52.227-19(c)(1,2).


$ logout
sh: logout:  not found.
$ quit
sh: quit:  not found.
$ exit
logout


Se ha perdido la conexión con el host.
```

TELNET: Seguridad

rlogin, rsh, telnet

- Texto plano
- Vulnerable a todos los tipos de ataques

SSH 1

- Creado en respuesta a ataques de packet sniffing

SSH 2

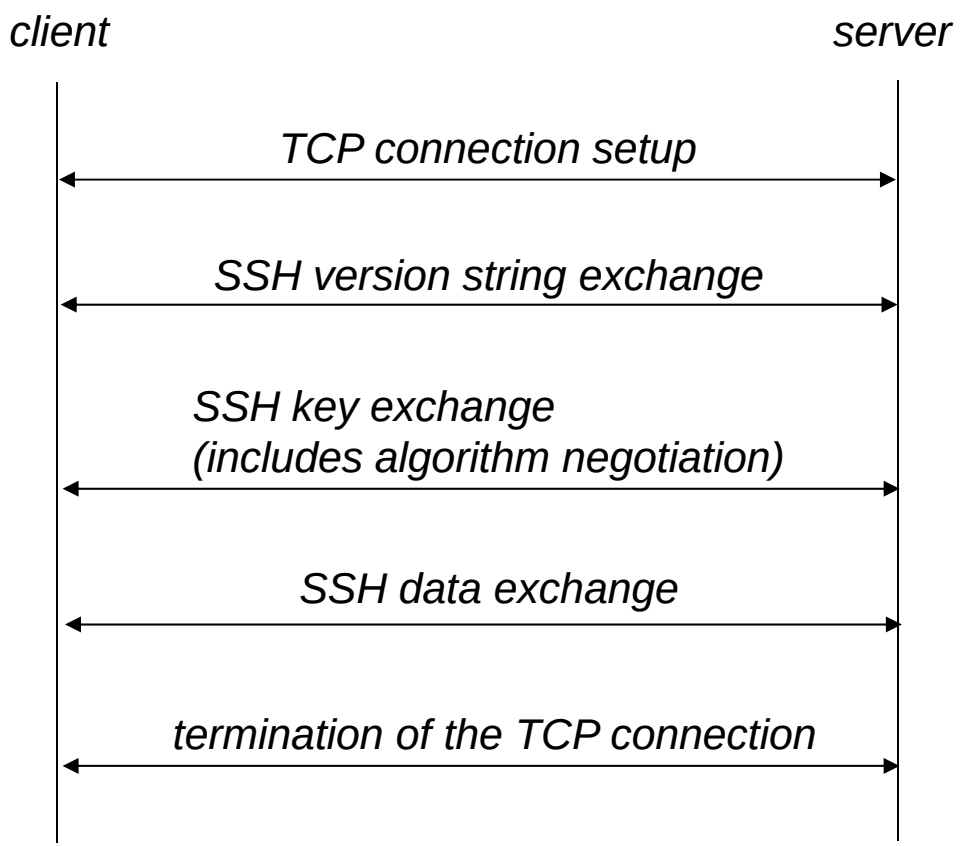
- Mejora del SSH 1 como respuesta a la evolución de los ataques

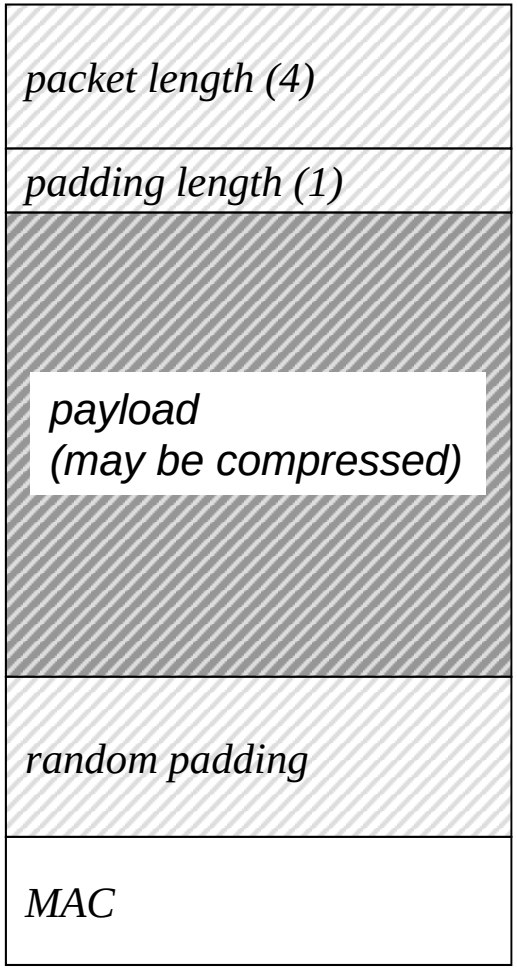
SSH - Secure Shell

- *SSH es un protocolo de acceso remoto seguro y otros servicios de red a través de una red insegura*
- *Desarrollado por SSH Communications Security Corp., Finlandia*
- *Dos distribuciones están disponibles:*
  - *versión comercial*
  - *Freeware (www.openssh.com)*

SSH - Componentes

- *SSH Protocolo de transporte Capa (TLP)*
  - *proporciona autenticación de servidores, la confidencialidad y la integridad de servicios puede proporcionar una compresión en la parte superior de cualquier capa de transporte fiable (por ejemplo, TCP)*
- *SSH Protocolo de autenticación de usuario*
  - *proporciona autenticación de usuario del lado del cliente se ejecuta en la parte superior de la SSH TLP*
- *Protocolo de conexión SSH*
  - *multitúnel seguro proporcionado por la capa de transporte SSH y Protocolos de autenticación de usuario en varios canales lógicos*


client
server
TCP connection setup
SSH version string exchange
SSH key exchange
(includes algorithm negotiation)
SSH data exchange
termination of the TCP connection



- packet length:
  - length of the packet not including the MAC and the packet length field
- padding length:
  - length of padding
- payload:
  - useful contents
  - might be compressed
  - max payload size is 32768
- random padding:
  - 4 – 255 bytes
  - total length of packet not including the MAC must be multiple of max(8, cipher block size)
  - even if a stream cipher is used
- MAC:
  - message authentication code
  - computed over the clear packet and an implicit sequence number

encryption

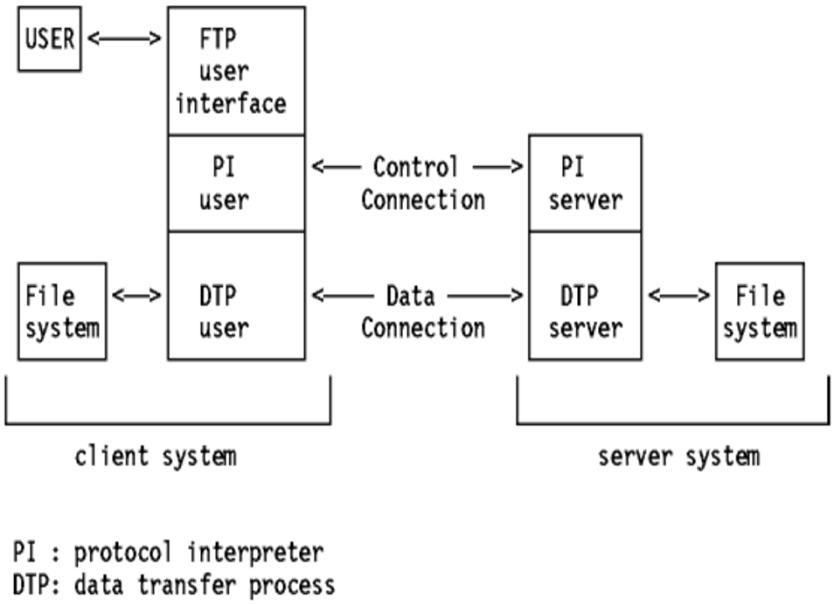
compression

Transferencia de Ficheros: FTP

- Protocolo + Aplicación sobre TCP (*RFC 959)*
- Acceso y transferencia de archivos
- Modelo cliente servidor
- Conexión de control (login + Telnet) + conexión de datos (gestión de transf.)
- Puertos fijos “bien conocidos” (21 + 20)
- El servidor autentifica al cliente antes de permitir la transferencia
- El enlace está orientado a conexión: ambos hosts deben estar activos y ejecutando TCP/IP

FTP: Descripción

- Las conexiones fiables se realizan con TCP
- Emplea dos conexiones: la primera es para el login y TELNET y la segunda es para gestionar la transferencia de datos.
- El usuario que inicia la conexión asume la función de cliente
- En ambos extremos del enlace, la aplicación FTP se construye con intérprete de protocolo(PI), un proceso de transferencia de datos, y una interfaz de usuario
- La interfaz de usuario se comunica con el PI, que está a cargo del control de la conexión.
- En el otro extremo de la conexión, el PI, además de su función de responder al protocolo TELNET, ha de iniciar la conexión de datos.
- Los DTPs se ocupan de gestionar la transferencia de datos.



FTP: Operaciones

- **Conexión a un host remoto**
  - Open: Selecciona el host remoto de inicia la sesión con el login
  - User: Identifica al ID del usuario remoto
  - Pass: Autentifica al usuario
  - Site: Envía información para proporcionar servicios específicos para ese host
- **Selección de un directorio**
  - cd (change directory) y lcd (Local change directory)
- **Listado de ficheros disponibles para una transferencia**
  - dir, ls
- **Especificación del modo de transferencia**
  - Mode: el fichero tiene estructura de registros o es un flujo de bytes.
  - Block: respeta las separaciones lógicas entre registros.
  - Stream: El fichero se trata como un flujo de bytes (por defecto)
  - Type: Conjunto de caracteres usado para los datos.
  - ASCII, EBCDIC: Traducción (si fuera necesario) ASCII-EBCDIC.
  - Image bits contiguos empaquetados en bytes de 8 bits.
- **Copiar ficheros de o al host remoto**
  - Get: Copia del host remoto al host local.
  - Put: Copia del host local al host remoto.
- **Desconectar del host remoto**
  - Quit: Desconecta del host remoto y cierra el FTP
  - Close: Desconecta del host remoto pero deja al cliente FTP ejecutándose

FTP: Códigos de respuesta

- Cliente y el servidor mantienen un diálogo por medio de TELNET.
- El cliente lanza comandos, y el servidor contesta con *códigos de respuesta*.
- Las respuestas incluyen comentarios para el usuario
- Los códigos de respuesta tienen tres dígitos, siendo el primero el más significativo
- Los dígitos primero y segundo proporcionan más detalles de la respuesta

Reply code	Description
1xx	Positive preliminary reply.
2xx	Positive completion reply.
3xx	Positive intermediate reply.
4xx	Transient negative completion reply.
5xx	Permanent negative completion reply.

FTP: Ejemplo

```
[C:\SAMPLES]ftp host01.itsc.raleigh.ibm.com
Connected to host01.itsc.raleigh.ibm.com.
220 host01 FTP server (Version 4.1 Sat Nov 23 12:52:09 CST 1991) ready.
Name (rs60002): cms01
331 Password required for cms01.
Password: xxxxxx
230 User cms01 logged in.
ftp> put file01.tst file01.tst
200 PORT command successful.
150 Opening data connection for file01.tst (1252 bytes).
226 Transfer complete.
local: file01.tst remote: file01.tst
1285 bytes received in 0.062 seconds (20 Kbytes/s)
ftp> close
221 Goodbye.
```

TFTP (Trivial File Transport Protocol

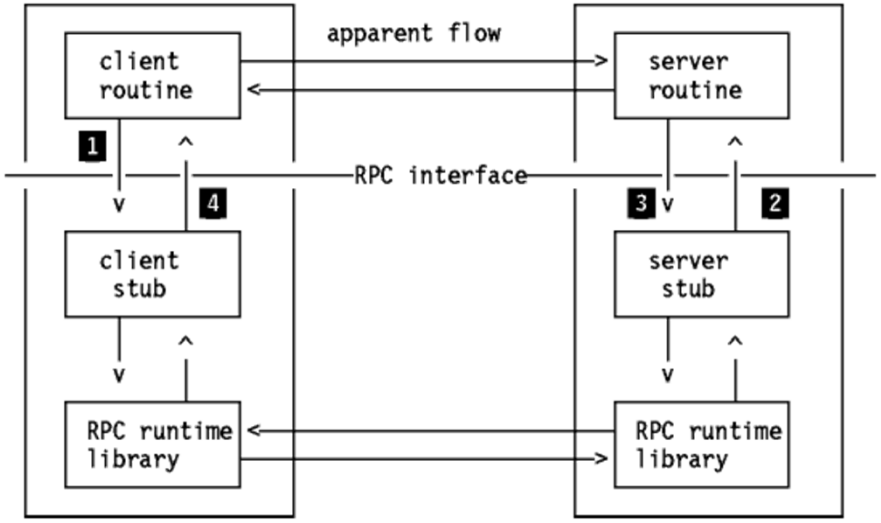
- Basado en UDP, por lo que no requiere conexión
- FTP en redes fiables: No hay autentificación
- Control por timeout y retransmisión
- Modo de Funcionamiento:
  - La transferencia comienza con una solicitud para leer o escribir un fichero.
  - Si el servidor concede la solicitud, se abre la conexión y el fichero se envía en bloques consecutivos de 512 bytes(longitud fija).
  - Los bloques del fichero se numeran correlativamente, comenzando en 1.
  - Cada paquete de datos debe ser reconocido mediante un paquete de reconocimiento antes de que se envíe el siguiente paquete.
  - Se asume la terminación al recibir un paquete de menos de 512 bytes.

TFTP: Limitaciones

- La mayoría de los errores provocarán la terminación de la conexión (falta de fiabilidad)
  - Si un paquete se pierde en la red, se producirá un "timeout", tras el que se efectuará la retransmisión del último paquete (de datos o de reconocimiento)
- En el RFC 783 se describe un error grave: "***Síndrome del Aprendiz de Brujo***": retransmisión excesiva en ambas partes de la conexión

RPC (Remote Procedure Call)

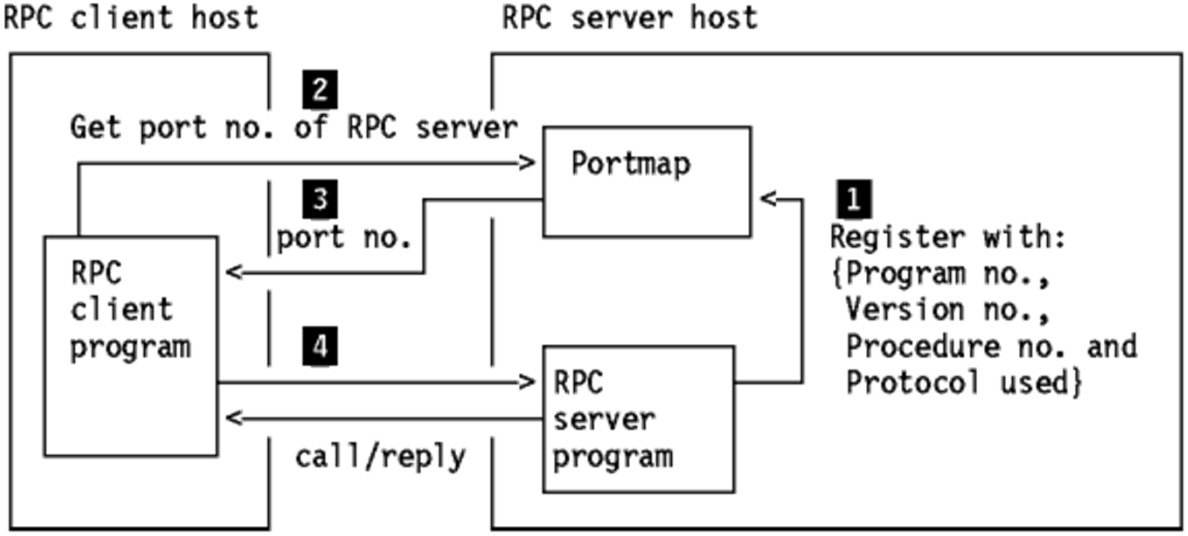
- *API para desarrollo de aplicaciones distribuidas*
- *TCP y UDP*
- *Tres partes:*
  - *RPCGEN: Un compilador de la interfaz del proc. remoto*
  - *XDR ("eXternal Data Representation"): estándar de codif.*
  - *Una librería Run-Time*



- *El proceso llamador envía un mensaje de llamada y espera por la respuesta.*
- *En el lado del servidor un proceso permanece dormido a la espera de mensajes de llamada. Cuando llega una llamada, el proceso servidor extrae los parámetros del procedimiento, calcula los resultados y los devuelve en un mensaje de respuesta.*

RPC: Portmap

- Informa al llamador por el puerto 111 de qué número de puerto ocupa un programa en su host.
- Sólo tiene conocimiento de programas RPC en el host local
- Cada programa RPC debe registrarse cuando arranca y anular su registro al finalizar su ejecución.
- El llamador puede enviar los parámetros del procedimiento al Portmap para que este lo invoque.

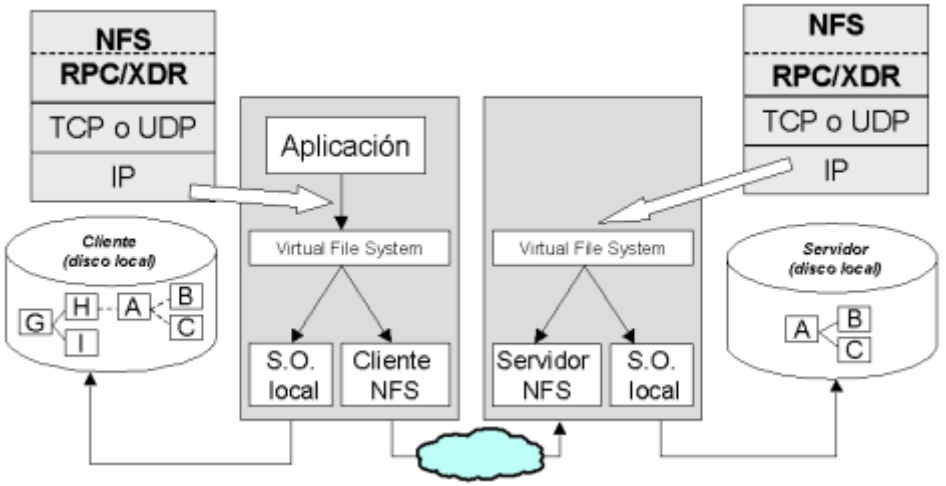


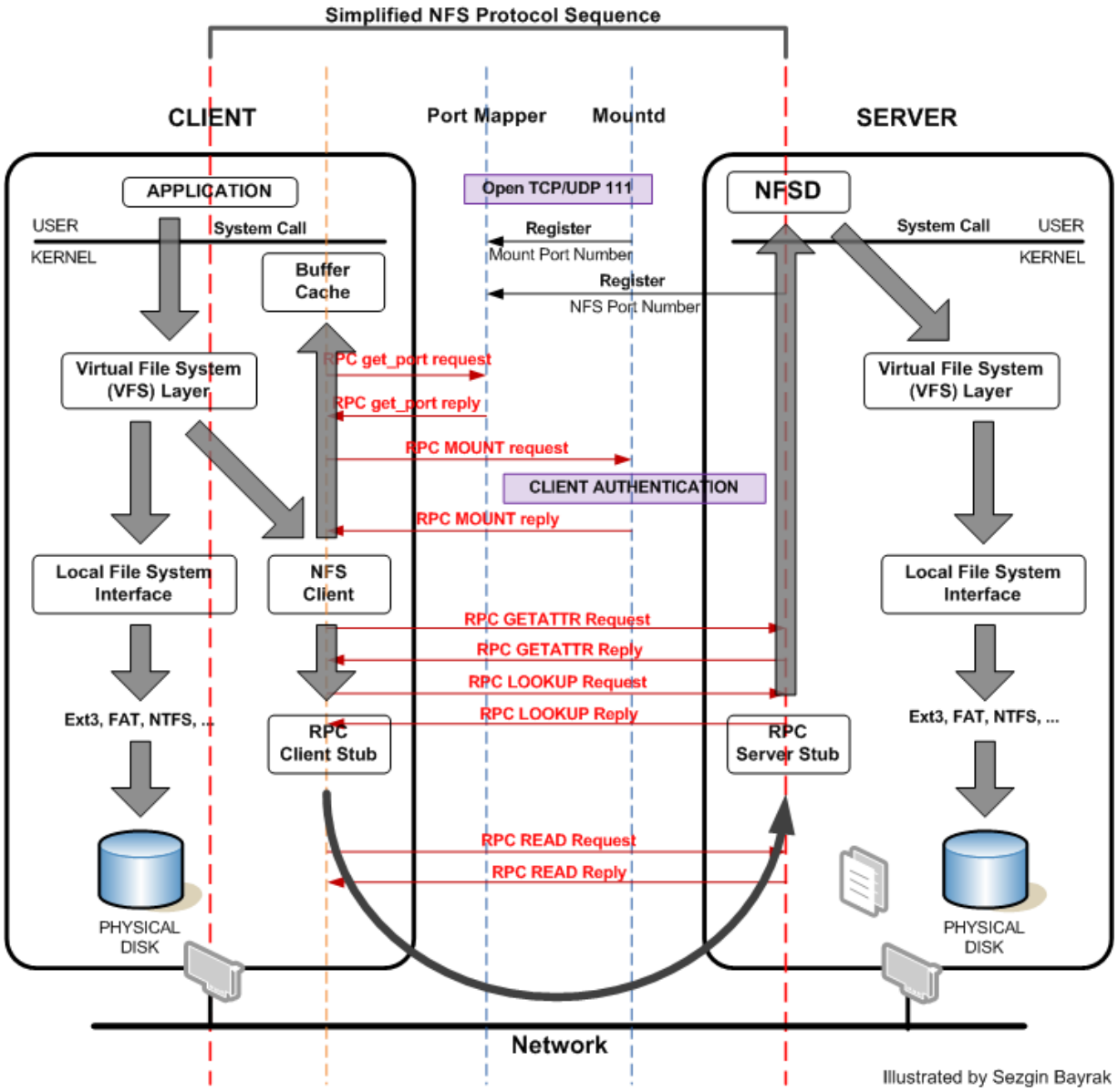
TRANSFERENCIA DE FICHEROS: NFS

- *Compartición de ficheros*
- *Independencia de la máquina, del S.O., y del protocolo de Transporte*
- *Se implementa sobre RPC, que establece el XDR*
- *Almacenamiento local/remoto*
- *Funcionamiento transparente (acceso como en local)*
- *Dos protocolos de RPC sobre UDP:*
  - *Mount : especifica el host remoto y el sistema de ficheros*
  - *NFS : realiza la E/S sobre el sistema remoto*

Sistema de ficheros NFS:

Estructura jerárquica (directorios)
operaciones síncronas para garantizar la integridad
No hay asociación entre cliente y servidor (si el servidor falla, el cliente reintenta hasta que responda, sin realizar otra vez el mount)




Simplified NFS Protocol Sequence
CLIENT
Port Mapper
Mountd
SERVER
APPLICATION
USER
KERNEL
System Call
Buffer Cache
Virtual File System (VFS) Layer
Local File System Interface
Ext3, FAT, NTFS, ..
PHYSICAL DISK
NFS Client
RPC Client Stub
Open TCP/UDP 111
Register
Mount Port Number
Register
NFS Port Number
RPC get_port request
RPC get_port reply
RPC MOUNT request
CLIENT AUTHENTICATION
RPC MOUNT reply
RPC GETATTR Request
RPC GETATTR Reply
RPC LOOKUP Request
RPC LOOKUP Reply
RPC READ Request
RPC READ Reply
NFSD
System Call
USER
KERNEL
Virtual File System (VFS) Layer
Local File System Interface
Ext3, FAT, NTFS, ...
RPC Server Stub
PHYSICAL DISK
Network
Illustrated by Sezgin Bayrak

El sistema de autentificación y autorización Kerberos

Según el Gran Diccionario del Diablo("Enlarged Devil's Dictionary (Ambrose Bierce)"), Cerbero es "el perro guardián del Hades, cuyo deber era guardar la entrada del infierno; se sabe que Cerbero tenía tres cabezas".

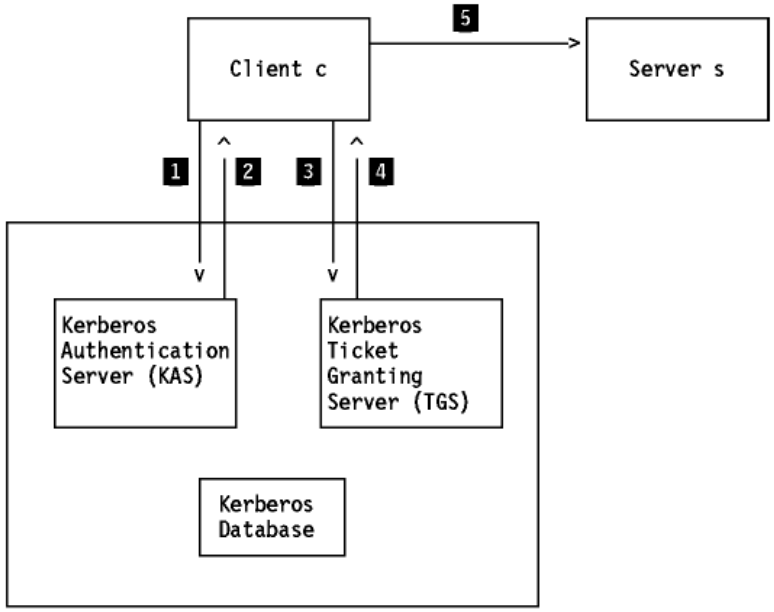
- Sistema de seguridad basado en la encriptación
- Proporciona autentificación mutua entre usuarios y servidores
  - Evita solicitudes/respuestas fraudulentas entre servidores y usuarios
  - Cada servicio puede implementar su propio sistema de autorización independiente
  - Puede asumir que la autentificación usuario/cliente es fiable
- Permite implementar un sistema de contabilidad integrado, seguro y fiable, con estructura modular y soporte para facturación
- Versión 4: amplio uso, implementada en productos comerciales
- Versión 5: está propuesta como borrador de Internet

Kerberos: condiciones

- La red incluye:
  - estaciones de trabajo públicas y privadas localizadas en áreas con seguridad física mínima.
    - Ejm.: una red universitaria sin encriptación de los enlaces que puede estar compuesta de redes locales dispersas conectadas por troncales o pasarelas
  - servidores operados de modo centralizado en habitaciones cerradas con seguridad física moderada o alta.
- Los datos confidenciales o las operaciones de alto riesgo (Ejm. transacciones bancarias) no pueden formar parte de este entorno sin seguridad adicional
  - Peligro de emular ciertas condiciones y los datos normales su protección por encriptación.
- Uno de los sistemas de encriptación utilizados es el DES("Data Encryption Standard")
- Kerberos asume la existencia de un reloj débilmente sincronizado ("Time server").

Kerberos: Autentificación

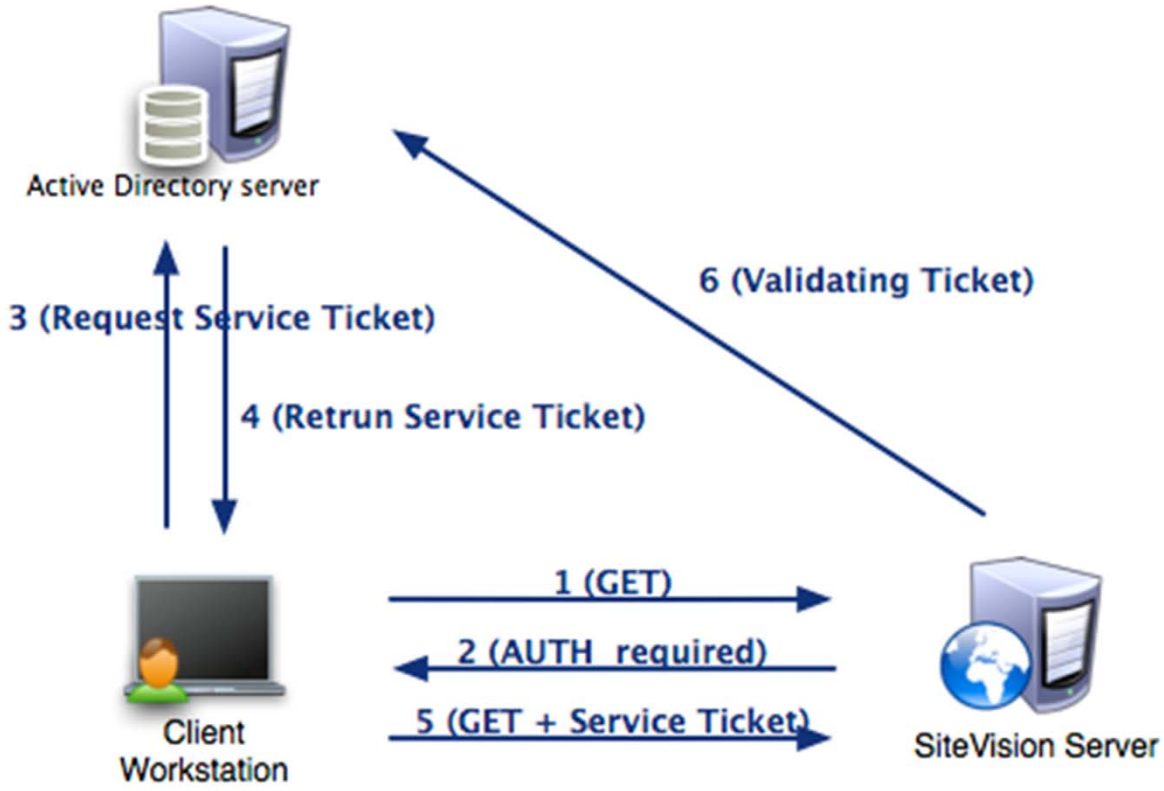
- El cliente que desea contactar con un servidor debe pedir primero un *ticket* de una tercera parte de mutua confianza, el **KAS**("Kerberos Authentication Server").
- Este ticket se obtiene como una función en la que uno de los componentes es una llave privada conocida sólo por el servicio y el KAS,
- El servicio puede estar seguro de que el ticket procede sólo de Kerberos.
- El KAS conoce al cliente por su nombre principal(**c**).
- La llave privada(**K(c)**) es la llave de autentificación conocida sólo por el usuario y el KAS



1 Client —> KAS: c, tgs, n
2 KAS —> Client: $\{K_{c,tgs}, n\}K_c$, $\{T_{c,tgs}\}K_{tgs}$
3 Client —> TGS: $\{A_c\}K_{c,tgs}$, $\{T_{c,tgs}\}K_{tgs}$, s, n
4 TGS —> Client: $\{K_{c,s},n\}K_{c,tgs}$, $\{T_{c,s}\}K_s$
5 Client —> Server: $\{A_cn\}K_{c,s}$, $\{T_{c,s}\}K_s$

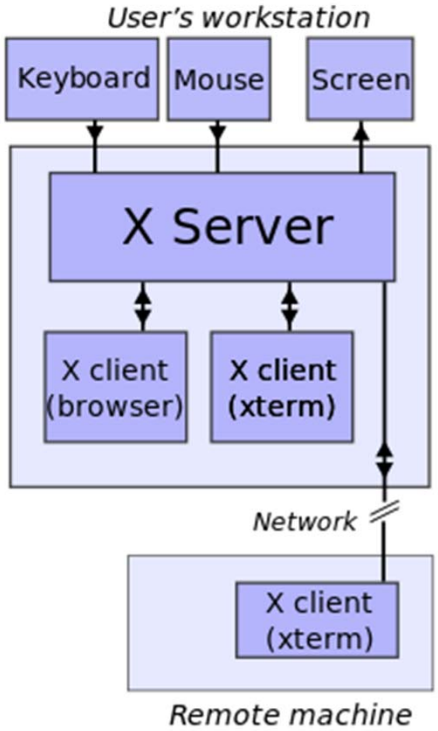
In Kerberos Version 4:

message 2 was: KAS —> Client: $\{K_{c,tgs},n,\{T_{c,tgs}\}K_{tgs}\}K_c$
message 4 was: TGS —> Client: $\{K_{c,s},n,\{T_{c,s}\}K_s\}K_{c,tgs}$


Active Directory server
3 (Request Service Ticket)
4 (Retrun Service Ticket)
6 (Validating Ticket)
1 (GET)
2 (AUTH required)
5 (GET + Service Ticket)
Client Workstation
SiteVision Server

X-WINDOWS

- Desarrollado a mediados de los años 1980 en el MIT
  - Interfaz gráfica a los sistemas Unix.
  - Interacción gráfica en red entre un usuario y una o más computadoras
  - Generalmente se refiere a la versión 11 de este protocolo, X11, totalmente independiente del sistema operativo.
- Para utilizar un programa de cliente X sobre una máquina remota, el usuario hace lo siguiente:
  - En la máquina local, abrir una ventana de terminal
  - usar telnet o ssh para conectarse con la máquina remota
  - solicitar el servicio local de pantalla/entrada (ej., export DISPLAY=[user's machine]:0 si no se está usando SSH con X tunneling activado)
- Sistemas:
  - UNIX: XFree86, Xorg.Server
  - OpenVMS: DECwindows
  - MAC OS: X11.app, eXodus, MacX
  - Windows: Cygwin/X, Xming y WeirdX; productos propietarios Xmanager, Exceed, MKS X/Server, Reflection X, y X-Win32.



BOOTP (Bootstrap Protocol)

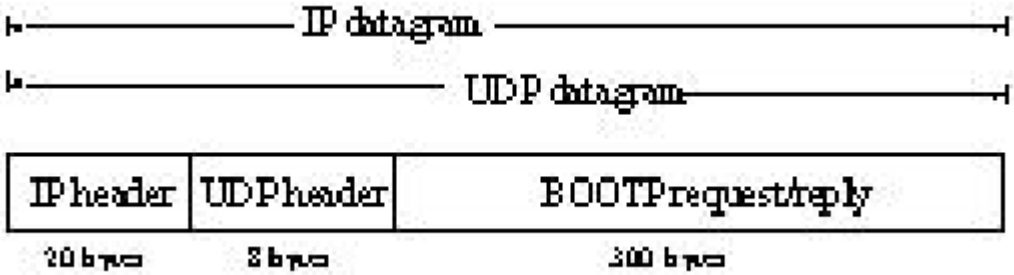
- Función análoga a RARP pero:
  - RARP de estaciones sin disco
  - Basado en UDP
  - Permite suministrar todos los parámetros de configuración al 'cliente'.
    - Obtención de dir. IP de servidores, gateways
    - Obtención de la submáscara de red
- El servidor puede estar en una LAN diferente al cliente.
- Es preciso registrar en el servidor todas las direcciones MAC que vayan a usar el servicio.

Tanto RARP como BOOTP requieren asignación estática biunívoca entre direcciones MAC y direcciones IP: Tantas direcciones IP como ordenadores utilicen TCP/IP

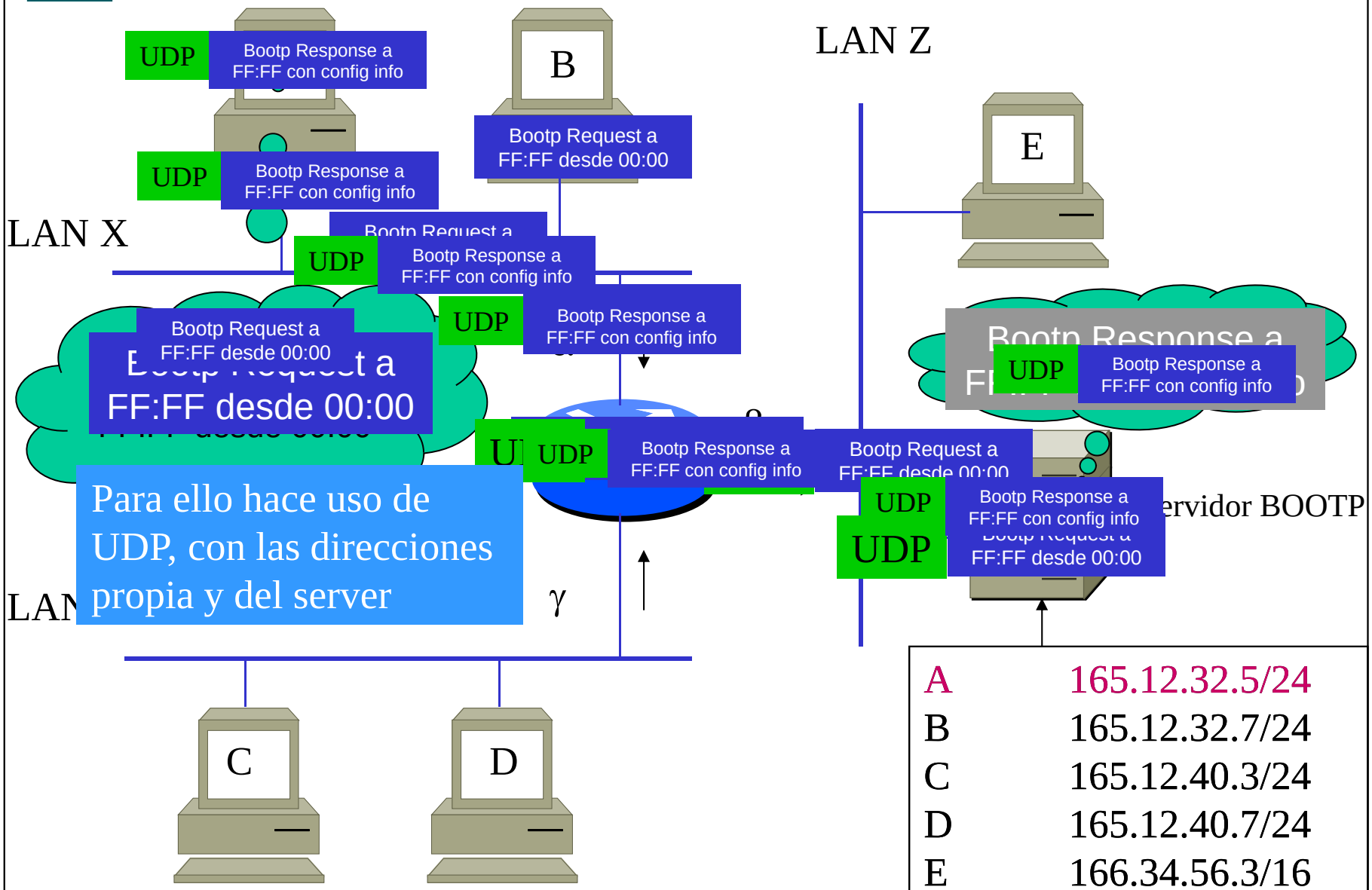
Casos problemáticos:

Una empresa de 500 PC's con conexiones simultáneas del 50 %
Una sala para acceso Internet de portátiles con 50 direcciones IP

Bootp y encapsulado UDP



- Peticiones y respuestas son encapsuladas en datagramas UDP
- Hace uso de puertos “Well-known”:
  - Servidor: 67
  - Cliente: 68



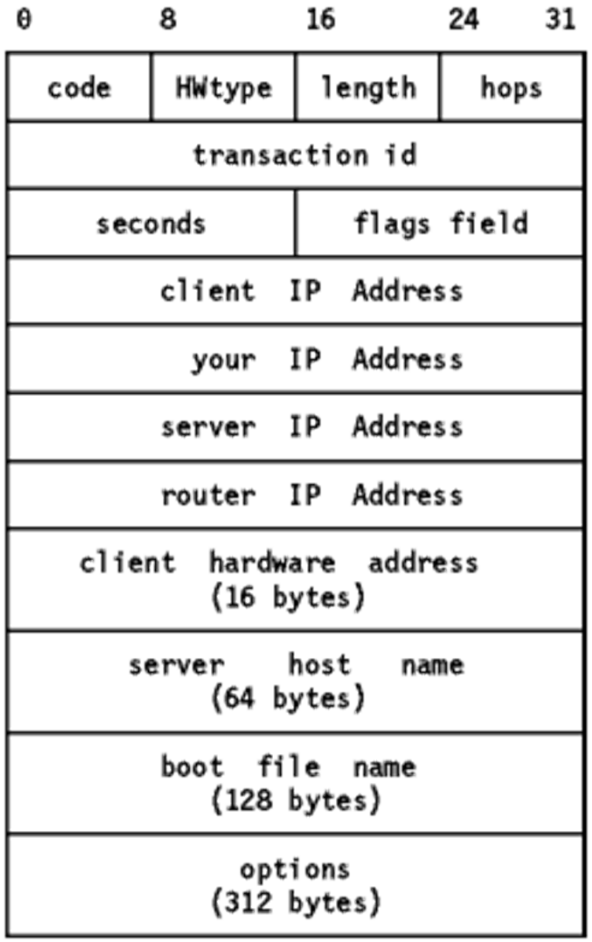
A	165.12.32.5/24
B	165.12.32.7/24
C	165.12.40.3/24
D	165.12.40.7/24
E	166.34.56.3/16

DHCP (Dynamic Host Control Protocol)

- Similar al BOOTP
- Asignación dinámica de direcciones
- El cliente solicita una dir IP que “alquila” un tiempo negociado en el momento de la conexión (minutos/indefinido):
  - Indefinido y estático (por admon.), equivale a BOOTP.
  - Automático (también estático)
  - Dinámico (se asigna la dirección de un pool). Requiere interacción de DHCP con el DNS.
- Mejora la seguridad y fiabilidad de la red, así como su admon.
- Es lo más parecido a la autoconfiguración automática

  - Problema: **detectar errores de equipos concretos**

Formato de mensajes DHCP



- Code: Indica solicitud (1) o respuesta (2)
- Hwtype: tipo de hardware (1 Ethernet, 6 IEEE 802 Networks
- Length: Longitud en bytes de la dirección hardware.
- Hops: El cliente lo pone a 0. Cada router lo incrementa (3 indica un bucle)
- Transaction ID: Nº aleatorio que compara la solicitud con la respuesta que genera.
- Seconds: Fijado por el cliente. Tiempo transcurrido desde el inició de arranque.
- Flags Field: El bit más significativo indica broadcast.
- Client IP adress: Fijada por el cliente. O bien es su dirección IP real , o 0.0.0.0.
- Your IP Address: Fijada por el servidor si el valor del campo anterior es 0.0.0.0
- Server IP address: Fijada por el servidor.
- Router IP address: Fijada por el router retransmisor si se usa retransmisión BOOTP.
- Client hardware address: Fijada por el cliente. El servidor identifica al cliente.
- Server host name: Nombre opcional del host servidor acabado en X'00'.
- Nombre del fichero de arranque
- Options: Los primeros cuatro bytes contienen el cookie(99.130.83.99)

Parámetros BOOTP/DHCP

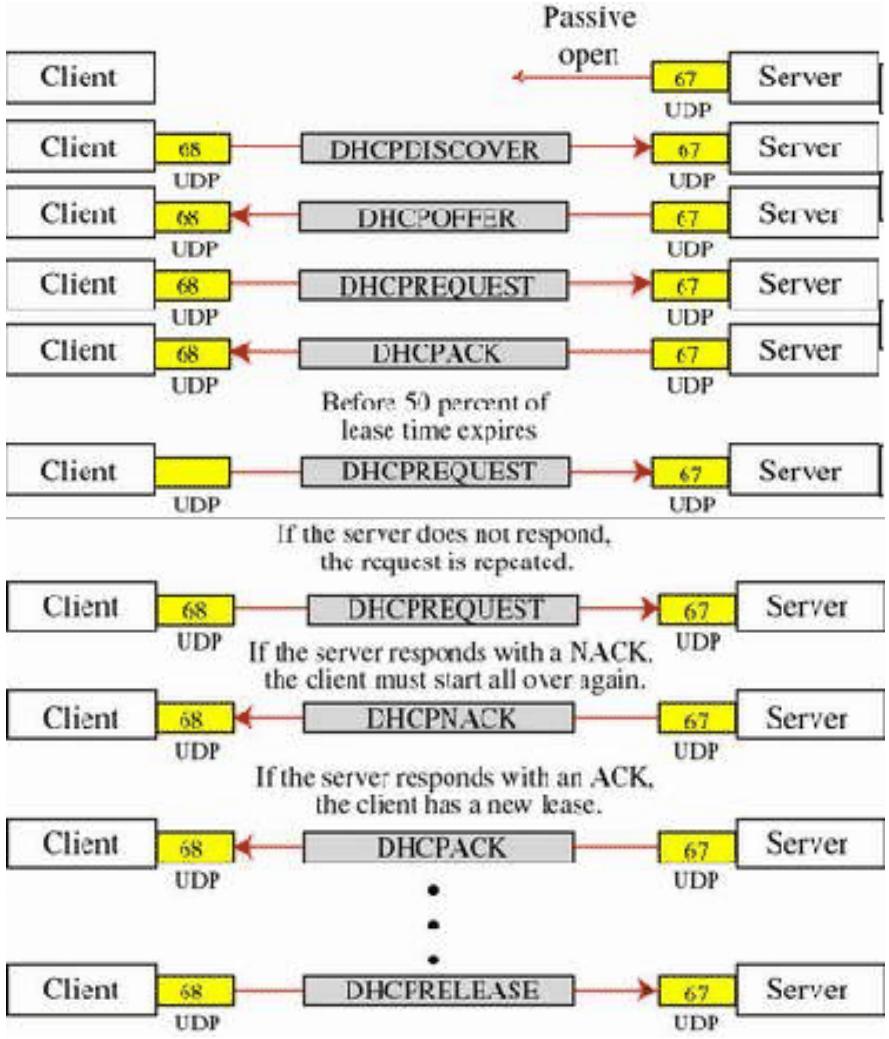
- Dirección IP del cliente
- Hostname del cliente
- Máscara de subred
- Dirección(es) IP de:
  - Routers
  - Servidores de nombres
  - Servidores de impresión (LPR)
  - Servidores de tiempo
- Nombre y ubicación del fichero que debe usar para hacer boot (lo cargará después por TFTP)

Proceso DHCP

1) **Solicitud de la dirección IP (IP lease request):** El cliente inicia una versión "limitada" de TCP/IP, y hace broadcast solicitando la ubicación del servidor DHCP.
2) **Ofrecimiento de una dirección IP (IP lease offers):** Todos los servidores DHCP que tengan direcciones IP disponibles envían una oferta al cliente.
3) **Selección de la dirección IP (IP lease selection):** El cliente selecciona la dirección IP de la primera oferta que recibe, y luego hace broadcast para solicitar que se le "arriende" esa dirección específica.
4) **Reconocimiento de la dirección IP (IP lease acknowledgement):** El servidor DHCP que haya hecho la oferta seleccionada, responde al mensaje, mientras los demás desechan su ofrecimiento. Se asigna al cliente la información relacionada con la dirección IP, y se le envía un reconocimiento de la operación.
5) **El cliente finaliza el proceso**, iniciando el protocolo TCP/IP.

Cuando el proceso de configuración automática finaliza, el cliente puede utilizar todos los servicios y utilidades TCP/IP para establecer comunicaciones de red normales y para conectarse a otros equipos IP.

Proceso DHCP



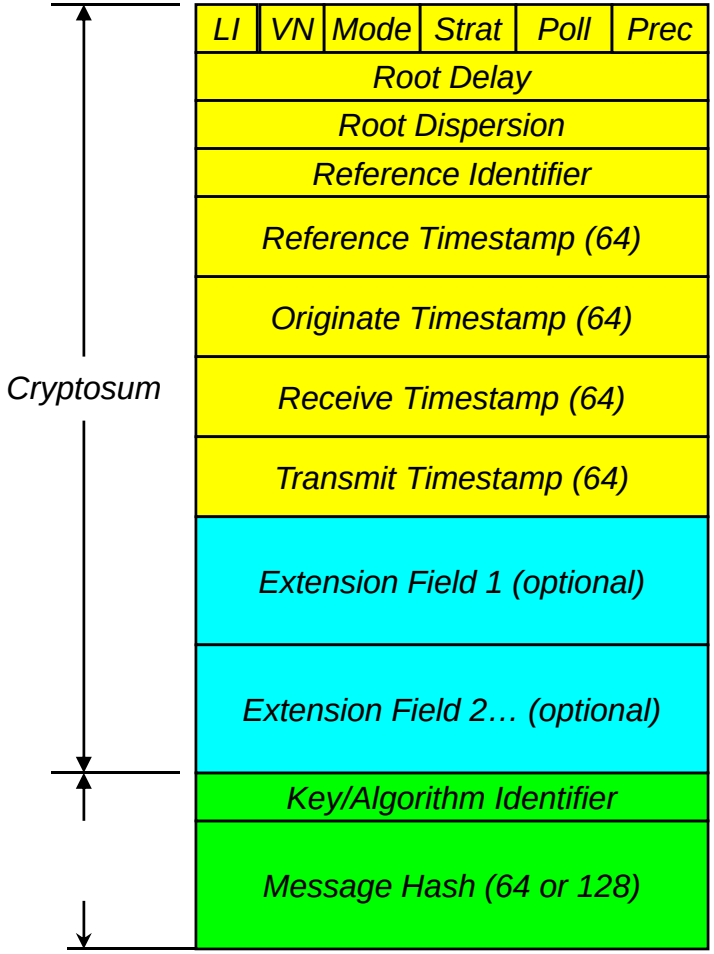
Etiq.	Long.	Valor
53	1	1 DHCPDISCOVER
		2 DHCPOFFER
		3 DHCPREQUEST
		4 DHCPDECLINE
		5 DHCPACK
		6 DHCPNACK
		7 DHCPRELEASE

Network Time Protocol (NTP)

- Sincroniza los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable.
- Diseñado originalmente por David L. Mills de la Universidad de Delaware (1985)
- UDP 123
- NTPv4 puede mantenerse sincronizado con una diferencia máxima de 10 milisegundos (1/100 segundos)
- Sistema de jerarquía de *estratos de reloj*:
  - Sistemas de estrato 1: sincronizados con un reloj externo tal como un reloj GPS o algún reloj atómico.
  - Sistemas de estrato 2: derivan su tiempo de uno o más de los sistemas de estrato 1, y así consecutivamente
- *Protocolo Simple de Tiempo de Red'* ó **SNTP**. Ha ganado popularidad en dispositivos incrustados y en aplicaciones en las que no se necesita una gran precisión

NTP packet/timestamp

NTP Protocol Header Format (32 bits)

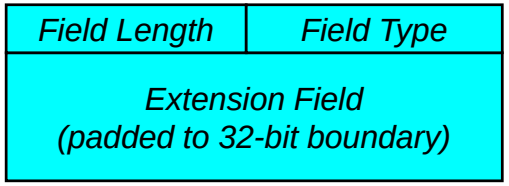


LI	leap warning indicator
VN	version number (4)
Strat	stratum (0-15)
Poll	poll interval (log2)
Prec	precision (log2)

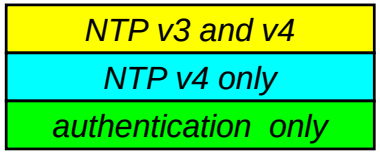
NTP Timestamp Format (64 bits)

Value is in seconds and fraction since 0^h 1 January 1900

NTPv4 Extension Field



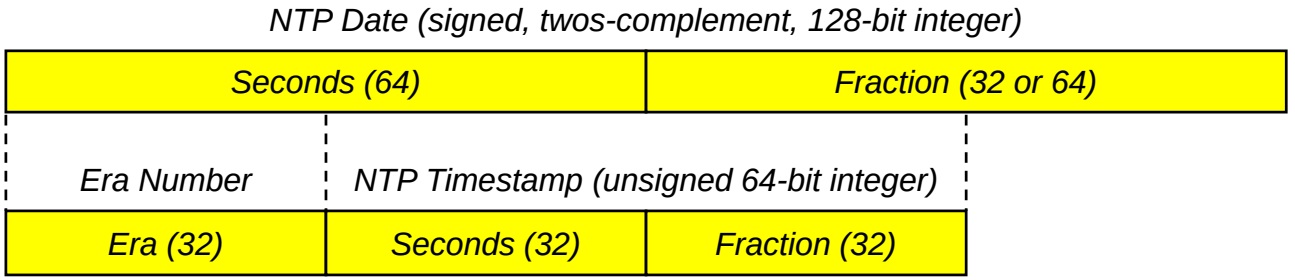
Last field padded to 64-bit boundary



Authenticator uses DES-CBC or MD5 cryptosum of NTP header plus extension fields (NTPv4)

NTP: Algunas fechas interesantes

Year	M	D	JDN	NTP Date	Era	Timestamp		
-4712	1	1	0	-208,657,814,400	-49	1,795,583,104	First day Julian Era	
1	1	1	1,721,426	-59,926,608,000	-14	202,934,144	First day Common Era	
1582	10	15	2,299,161	-10,010,304,000	-3	2,874,597,888	First day Gregorian Era	
1900	1	1	2,415,021	0	0	0	First day NTP Era 0	
1970	1	1	2,440,588	2,208,988,800	0	2,208,988,800	First day Unix Era	
1972	1	1	2,441,318	2,272,060,800	0	2,272,060,800	First day UTC	
2000	1	1	2,451,545	3,155,673,600	0	3,155,673,600	First day 21st century	
2036	2	7	2,464,731	4,294,944,000	0	4,294,944,000	Last day NTP Era 0	
2036	2	8	2,464,732	4,295,030,400	1	63,104	First day NTP Era 1	
3000	1	1	2,816,788	34,712,668,800	8	352,930,432	4294967296	



Extincted Apps

- Name/Finger protocol
  - Por David Zimmerman, RFC 742 (Diciembre 1977)
  - Earnest, autor del 1er progrma Finger (1971), le dió el nombre por la búsqueda en una lista[1]
  - Interfaz al **name y/o finger**
  - Provee información sobre el estado actual de un equipo o de los usuarios de los equipos provide status reports on a particular computer system or a particular person at network sites
  - Anteriormente **who** mostraba el ID y el número de línea de terminales y servidores para el facilitar el log-in.
  - TCP 79, denominado RUIP (Remote User Information Program).
  - Devuelve si un determinado usuario está conectado, la dirección de e-mail, su nombre completo, etc, y el contenido de **.project** y **.plan,** archivos gestionados por el usuario con sus propias informaciones de interés
  - Es el precursor del micro-blogging!!!

Extincted Apps

- rlogin
  - RFC 1282
  - Earnest, autor del 1er programa Finger (1971), le dió el nombre por la búsqueda en una lista[1]
  - Acceso remoto a sistemas UNIX (sistemas BSD)
  - TCP 513, denominado rlongid (Remote Login Daemon).
- IRC (Internet Remote Chat)
  - protocolo de comunicación en tiempo real basado en texto, que permite debates entre dos o más personas (canales IRC)
  - La aplicación IRCd (IRC daemon o servidor de IRC) que gestiona los canales y las conversaciones murales
  - Creado por Jarkko Oikarinen en agosto de 1988
  - Programas independientes, como mIRC, Irssi, Konversation o X-Chat fueron los más populares, o integrados dentro de otros programas, como Chatzilla
  - TCP 194