TCP/IP-Ethernet y Web-IO

Este documento explica los fundamentos de la tecnología de redes, todos los protocolos de Ethernet importantes, desde TCP/IP hasta SSL/TLS.

www.telsa.com.mx

W&T
www.WuT.de

Prólogo

Este cuaderno es para todos los que sin conocimientos especiales sobre redes de ordenadores quieren poner en funcionamiento terminales Ethernet con TCP/IP. Está estructurado en cuatro partes:

- **Comprender TCP/IP-Ethernet**
  Aquí se encuentra la información fundamental más importante sobre el tema TCP/IP.

- **Más protocolos y servicios**
  En este apartado descubrirá como funciona el correo electrónico E-MAIL; que ocurre al solicitar una página de internet y que otros protocolos y servicios importantes se puede encontrar en relación con TCP/IP-Ethernet.

- **Configurar TCP/IP- Ethernet**
  Aquí se mostrará paso a paso el ajuste (instalación) de TCP/IP-Ethernet en el PC con los sistemas operativos comunes.

- **Pequeño abecedario de redes**
  En este apartado mostramos los términos más importantes y abreviaturas en el manejo relacionado con redes de ordenadores.

Todos los procesos y relaciones importantes se explican fácilmente.

Sin miedo: no nos vamos a perder dentro del último detalle. Nos hemos limitado a propósito en las cosas que verdaderamente son importantes para la comprensión de las tecnologías descritas.

Para la puramente puesta en marcha de componentes TCP/IP no es preciso conocer hasta el último bit de cada protocolo.

Autores:
Frank Thiel y Rüdiger Theis

4ª edición revisada

Traducción:
realizada por Garper Telecomunicaciones S.L., Madrid (España)

Ya que podemos cometer fallos, no se pueden utilizar nuestras afirmaciones sin ser comprobadas. Por favor, comuníquennos todos los fallos o malos entendidos que descubra en este libro, para que así podamos reconocerlos y solucionarlos lo antes posible.

Contenido

Otros Protocolos y Servicios 47

COMPRENDER TCP/IP-ETHERNET

Todavía hace pocos años solamente se encontraban las redes de ordenadores en bancos, en la administración y en empresas grandes. Los componentes utilizados eran la mayoría casi imposibles de pagar, la instalación y administración sólo se podía solventar por técnicos especialistas preparados.

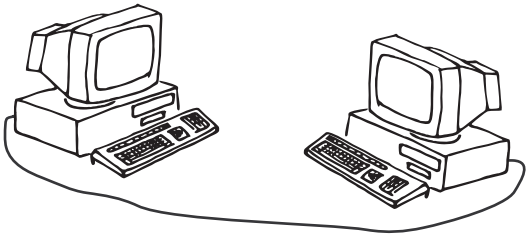
Pero como muy tarde los 90, los ordenadores, sobre todo el PC, tomaron importancia en todas las áreas de la vida diaria; la gran afluencia de datos contribuyó esencialmente a la extensión y al fuerte uso de redes de ordenadores.

Paralelamente a este desarrollo se extendió Internet explosivamente y los usuarios privados lo pueden usar hoy en día sin problemas.

Todo esto ha contribuido a que la posibilidad al acceso de ordenadores en red es hoy una parte esencial y fuerte de los modernos sistemas operativos. Las funciones más importantes se centran en dos cosas: el papel de Ethernet como fundamento físico y TCP/IP como protocolo.

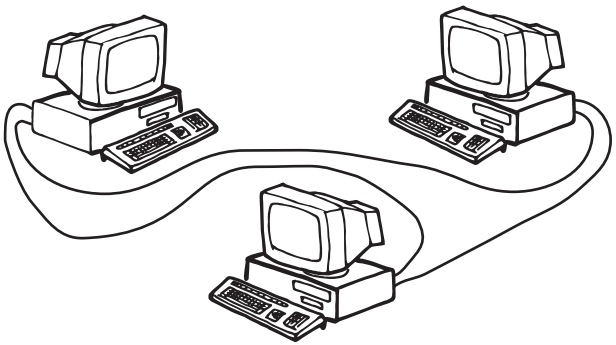
EXIGENCIAS A UNA RED DE COMPUTADORES

Cualquier usuario de ordenadores ha conectado seguramente alguna vez dos terminales entre sí (p.e. PC e impresora, PC y modem, PC y PC). Para la conexión sirve un cable especial de la aplicación deseada, sobre el que los datos entre los dos terminales serán enviados.



Se puede imaginar así: dos amigos por correspondencia se envían mutuamente cartas y un recadero está permanentemente ocupado llevando esas cartas a los buzones de los dos. En este sencillo ejemplo no es necesario ni el sobre, ni la dirección, ni el remitente.

El proceso es sencillo y funciona sin problemas. Se envían sólo puramente los datos útiles. Este tipo de conexión se llama también conexión punto a punto. Se podría utilizar, por supuesto también, la conexión punto a punto para comunicar entre si tres ordenadores. Para ello se tendría que instalar un cable desde cada PC a los otros dos.



Para el envío de cartas entre tres amigos por correspondencia se necesitarían con este proceso tres mensajeros.

Pero ya para cuatro PCs se necesitarían 6 cables y si se quisieran conectar 10 o más PCs de esta forma la consecuencia sería un enmarañado nudo de cables. Además cada cambio en una red de este tipo produciría una avalancha de cambios en el cableado. La realización de una red de este tipo es por lo tanto muy poco práctica.

Una red de ordenadores debería proporcionar a un número indeterminado de usuarios conectados a ella, acceso a las fuentes disponibles (almacenamiento, bases de datos, impresoras y otros equipos terminales) con el menor coste en material y cableado posible. Con ello debe de proporcionarse también la seguridad en los datos y velocidad de transmisión.

A causa de estas exigencias han surgido las corrientes estándares de la actualidad.

Funciones básicas de las redes

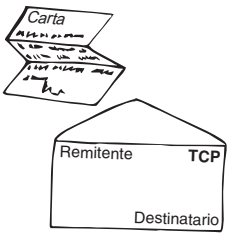
Básicamente tienen todas las topologías de redes algo en común:

Cada participante tiene una dirección propia. Los datos utilizados se „empaquetan“ en un marco con informaciones adicionales (pe. dirección destinatario, dirección remitente, suma de comprobación).

Con ayuda de la información de las direcciones en los paquetes de datos creados, se pueden transmitir los datos por un soporte físico común, al destinatario correcto.

Con una carta no es diferente: Se introduce la carta en un sobre, donde está escrito el destinatario y el remitente. El cartero sabe de esta forma a quién debe entregar la carta;

así el destinatario puede leer de dónde viene la carta y a quién puede responder en caso necesario.



Con la transferencia de datos dentro de una red el destinatario tiene la posibilidad adicional, con ayuda de la suma de comprobación, de asegurarse que todos los datos están completos.

Ethernet y FastEthernet

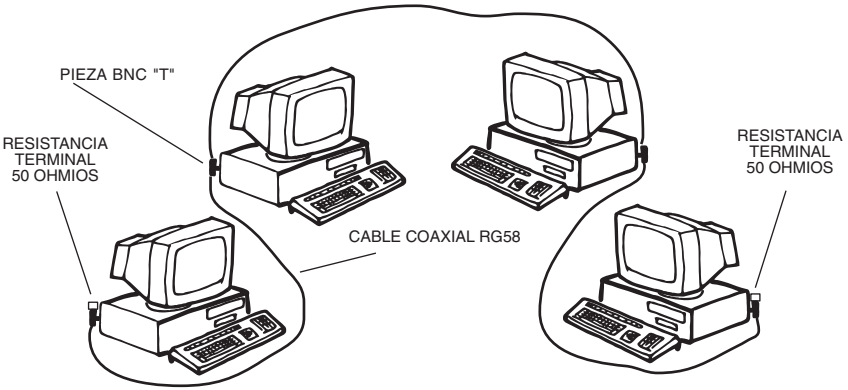
Ethernet es el estándar de redes más extendido en la actualidad. Ya en 1996 estaban implementadas en esta tecnología aprox. 86% de todas las redes existentes.

Ethernet utilizaba al principio una velocidad de transferencia de 10Mbit/s; hay fundamentalmente tres modelos físicos diferentes:

10Base2

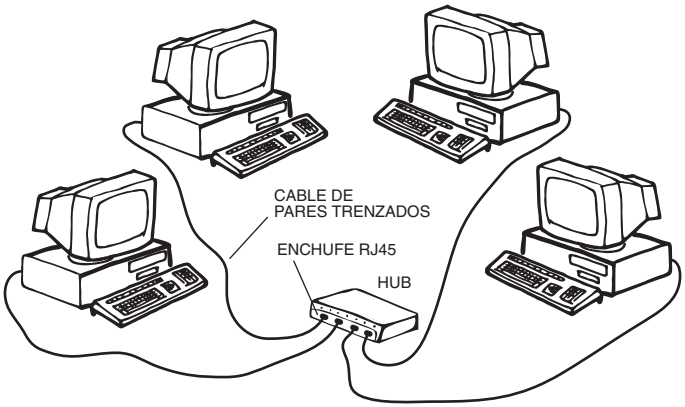
También conocido como Thin Ethernet , Cheapernet o sencillamente como red BNC. Todos los participantes se conectan paralelamente a un cable coaxial (RG58, 50 Ohmios).

El cable debe terminarse en cada final con una resistencia de 50 Ohmios.



10BaseT

Cada participante en la red se conecta con un cable propio de pares de hilos trenzados (Twisted Pair) a un aparato llamado Hub („distribuidor en estrella“), que distribuye a todos los usuarios en igual medida todos los paquetes de datos. 10BaseT trabaja por lo tanto a nivel físico en forma de estrella pero a nivel lógico al igual que 10Base2, como el principio de Bus.



10Base5
También nombrado „Yellow Cable“, presenta el inicio del estándar Ethernet y actualmente no tiene casi importancia.

Con el crecimiento de grandes cantidades de datos se introdujo en los 90 Fast Ethernet con una velocidad de transmisión de 100MBit/s. Hay dos modelos físicos diferentes:

100BaseT4
Igual que 10BaseT se conecta cada usuario con un cable propio de pares de hilos trenzados a un Hub, que distribuye a todos los paquetes de datos. 100BaseT4 no se utiliza normalmente en instalaciones nuevas.

100BaseTX
Presenta el estándar actual para redes de 100Mbit. 100BaseT4 y 100BaseTX se diferencian sólo a nivel físico en la forma de transmitir los datos. Además 100BaseTX necesita unos cables de mayor calidad.

Explicaciones más detalladas sobre Ethernet y las diferentes topologías físicas las puede encontrar en el W&T hojas de aplicaciones.

Cualquier modelo físico que se utilice tiene el mismo formato lógico de los paquetes de datos utilizados para todas las topologías Ethernet. Todos los usuarios en la red local reciben todos los paquetes de datos, incluidos aquellos que son para otros usuarios de la red (excepto el Switch, ver Anexo), pero sólo se procesan los paquetes que realmente están dirigidos a ellos.

Las direcciones de Ethernet, también llamadas MAC-ID o número de nodo, son fijadas irrefutablemente por el fabricante en el adaptador físico de Ethernet (Tarjeta de red, servidor de impresora, Com-Server, Router...), por lo tanto es fijo para cada terminal y no se puede cambiar. La dirección de Ethernet es un valor de 6 Bytes, que normalmente se escribe en Hexadecimal. Una dirección de Ethernet es normalmente así: 00-C0-3D-00-27-8B.

Cada dirección Ethernet es única en el mundo!

Los tres primeros valores hexadecimales corresponden al código del fabricante, los tres últimos calores hexadecimales son dados correlativamente por el fabricante.

Hay cuatro tipos diferentes de paquetes Ethernet, que se usan dependiendo de la aplicación:

Tipo de paquete	*Aplicación*
Ethernet 802.2	Novell IPX/SPX
Ethernet 802.3	Novell IPX/SPX
Ethernet SNAP	APPLE TALK Phase II
Ethernet II	APPLE TALK Phase I, TCP/IP

En conexión con TCP/IP se utilizan normalmente paquetes de datos Ethernet del tipo Ethernet II.

Aquí la estructura de un paquete de datos Ethernet II:

ESTRUCTURA DE UN PAQUETE DE DATOS ETHERNET

SUMA COMPROBACIÓN

⎍⎍⎍...	00C03D00278B	03A055236544	0800	DATOS	
Preamble	Destination	Source	Type	Data	FCS

Cada dirección Ethernet es única en el mundo!

Preámbulo	La secuencia de bits con consecutivos cambios entre o y 1 sirve para el reconocimiento del principio del paquete y para la sincronización. El final del preámbulo se señala con la secuencia de bits „11“.
Destino	Dirección Ethernet del receptor o destinatario.
Fuente	Dirección Ethernet del emisor o remitente.
Tipo	Proporciona el uso previsto para niveles superiores (pe. IP = Internet Protocol = 0800h).
Data	Datos de usuario o información útil transmitida.
FCS	Suma de comprobación.

La estructura de los otros paquetes Ethernet se diferencian sólo en los campos Tipo y Data, a los cuales dependiendo del tipo de paquete se les asigna otra función. Con esto un paquete de datos Ethernet dispone de varias características necesarias para enviar datos en redes locales de un usuario a otro.

No obstante, Ethernet por si sola no dispone de la posibilidad de direccionar diferentes redes. Además Ethernet trabaja sin conexión, es decir, el emisor no recibe del receptor ninguna confirmación si un paquete a llegado o no.

Como muy tarde, cuando una red Ethernet se tiene que conectar con más redes, entonces se tiene que trabajar con protocolos superiores como TCP/IP.

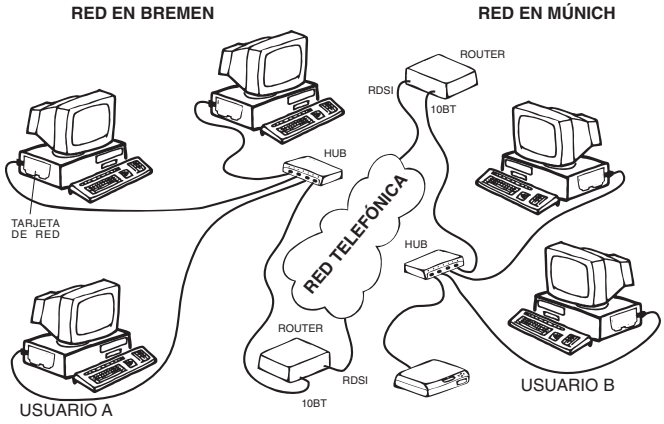
TCP/IP Los protocolos más importantes.

Ya en los años 60 encargó el ejercito americano la realización de un protocolo que permitiera, independientemente del Hardware y Software utilizado, el intercambio de información entre un número cualquiera de diferentes redes. De esta petición surgió en 1974 el protocolo TCP/IP.

Aunque TCP e IP siempre se nombran en conjunto, se trata de dos protocolos sobrepuestos uno encima del otro. El protocolo de Internet IP se hace cargo del correcto direccionamiento y reparto de los paquetes de datos, mientras que el protocolo siguiente „Transport Control Protocol“ TCP es responsable del transporte y seguridad de los datos.

IP Internet Protocol

El Protocolo de Internet hace posible que un número indeterminado de redes únicas se junten en una red conjunta. Posibilita además el intercambio de datos entre dos usuarios cualesquiera que a su vez están en sus redes independientes. La realización física de las redes o los caminos de transmisión, no tienen aquí ninguna importancia (Ethernet, Token Ring, RDSI...). Los datos son transferidos independientemente del medio al destino.



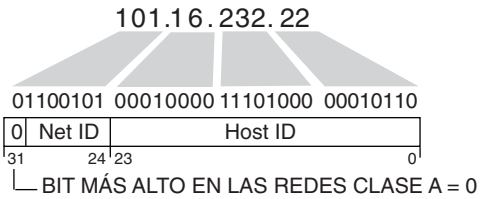
Direcciones IP

Bajo IP tiene cada usuario una única dirección de Internet, que a menudo se conoce como „Número IP“. Esta dirección de Internet es un valor de 32 Bits, que para su mejor comprensión siempre se da en formato de cuatro números decimales (valores de 8 Bits) separados por puntos (Notación Dot).

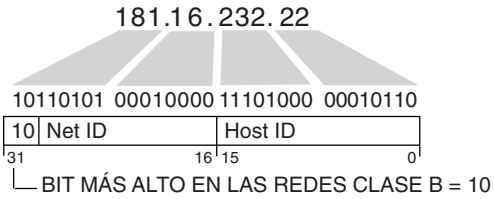
La dirección de Internet se divide en Net-ID y Host-ID, donde el Net-ID sirve para el direccionamiento de las redes y el Host-ID para el del usuario dentro de la red.

Para el direccionamiento de redes normales se diferencian tres clases de redes:

Clase A:
El primer Byte de la dirección IP sirve como direccionamiento de la red, los últimos 3 Bytes direccionan al usuario de la red

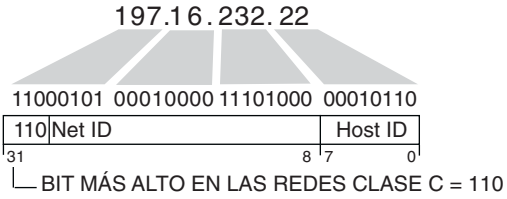


Clase B:
Los dos primeros Bytes de la dirección IP sirven para el direccionamiento de la red, los dos últimos Bytes direccionan al usuario de la red.



Clase C:

Los tres primeros Bytes de la dirección IP sirven para el direccionamiento de la red, el último Byte direcciona al usuario de la red.



En la tabla siguiente se encuentran los datos de referencia de las diferentes clases de redes:

RANGO DE DIRECCIONES DE LA RED		N°. DE REDES POSIBLES		N°. DE HOSTS/RED	
Clase A	1.xxx.xxx.xxx – 126.xxx.xxx.xxx	127	(2^7)	~ 16 Millones	(2^{24})
Clase B	128.0.xxx.xxx – 191.255.xxx.xxx	~ 16.000	(2^{14})	~ 65.000	(2^{16})
Clase C	192.0.0.xxx – 223.255.255.xxx	~ 2 Millones	(2^{21})	254	(2^8)

Aparte de las redes aquí descritas, existen también Clase D y E, que están por encima de la clase C. La red de clase D y E tienen en la práctica poca importancia, ya que sólo se utilizan para investigación y fines especiales. El usuario normal de Internet no tiene ningún contacto con este tipo de redes.

Para las redes que se tienen que conectar directamente con Internet, les asigna una comisión llamada InterNIC un Net-ID libre y decide dependiendo del tamaño planeado para la red de que clase se trata.

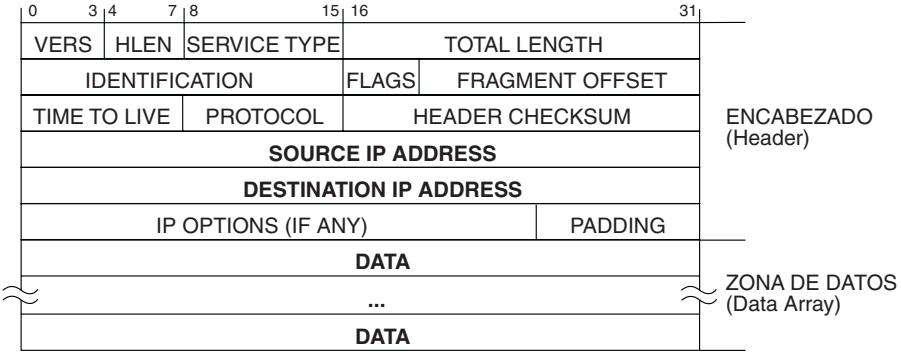
La asignación del Host-ID para el usuario y con ello la dirección IP resultante puede ser designada libremente por el administrador de la red. El debe de preocuparse de que sólo

una única dirección IP puede estar asignada en un mismo momento.

Paquetes de datos IP

También en la transmisión de datos por Internet se empaquetan los datos utilizados en un marco con informaciones de dirección. Los paquetes de datos IP contienen junto con los datos a transportar un montón de información adicional y de direccionamiento, que se incluyen en el llamado „encabezado“ del paquete.

ESTRUCTURA DE UN PAQUETE DE DATOS IP



Nos limitamos aquí a la descripción de las informaciones más importantes:

source IP address: Dirección IP del emisor.
destination IP address: Dirección IP del receptor.

TCP Transport Control Protocol

Porque IP es un protocolo inseguro, sin conexión, trabaja normalmente junto con el protocolo superior TCP, que se encarga de la seguridad y manejo de los datos.

TCP realiza una conexión entre dos usuarios durante la transmisión. Durante el establecimiento de la conexión se fijan condiciones como por ejemplo el tamaño de los paquetes de datos que permanecerá para toda la duración de la conexión.

TCP se puede comparar con una conexión telefónica. El usuario A marca al usuario B, usuario B acepta la conexión al descolgar el teléfono, esta conexión se mantiene hasta que uno de los dos la acaba.

TCP trabaja según el principio llamado *Client-Server*:

Aquel usuario que establece una conexión (el que coge la iniciativa), se le denomina Cliente (client). El cliente toma un servicio ofrecido por el servidor, donde dependiendo de cada servicio, el servidor puede atender al mismo tiempo más clientes.

Aquel usuario, al que se le establece una conexión, se denomina servidor (server). Un servidor no hace por si mismo nada, sino que espera a un cliente que establecerá una conexión con él.

En relación con TCP se habla de cliente TCP y servidor TCP.

TCP asegura los datos transmitidos con una suma de control (Cheksum) y estampa cada paquete enviado con un número secuencial. El receptor del paquete TCP comprueba la recepción correcta de los datos con la suma de control adjunta. Cuando un servidor TCP ha recibido correctamente un paquete, se calcula un número de reconocimiento (Acknowledgement) a partir del número secuencial y con un algoritmo preconcebido.

El número de reconocimiento se mandará de vuelta al cliente con el próximo paquete enviado como confirmación de recepción. El servidor estampa sus paquetes enviados igualmente con sus propios números secuenciales, que de igual manera se confirman por el cliente con un número de reconocimiento.

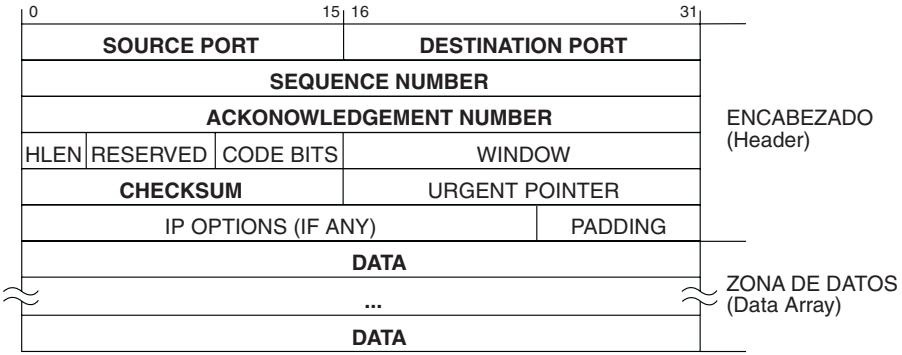
Con ello se consigue, que la pérdida de paquetes TCP sea detectada y esos paquetes perdidos, en caso necesario, en una secuencia correcta se puedan enviar nuevamente.

Además TCP dirige los datos de información al ordenador destino, en el cual hay diferentes aplicaciones, y al programa concreto de aplicación, también llamados servicios, a los que se dirige por diferentes números de puertos. Así es Telnet por ejemplo el puerto 23 y FTP puerto 21.

Se compara un paquete TCP con una carta dirigida a una institución, se puede comparar el número de puerto con el número de la habitación de ese servicio. Se encuentra por ejemplo el despacho de tráfico en la habitación 312 y se direcciona una carta a ese despacho, entonces se supone también al mismo tiempo que se quieren tomar los servicios del despacho de tráfico.

También TCP empaqueta los datos en un marco de informaciones adicionales. Estos paquetes TCP están estructurados como sigue:

ESTRUCTURA DE UN PAQUETE DE DATOS TCP



Source Port: Número de puerto de la aplicación del emisor.

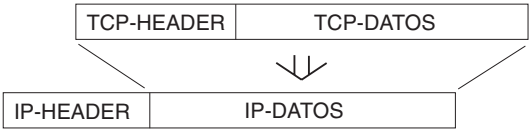
Destination Port: Número de puerto de la aplicación del receptor.

Sequence Nº: Offset de los primeros bytes de datos relativos al principio del flujo TCP

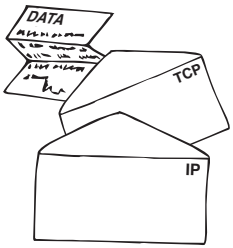
	(garantiza el mantenimiento de la secuencia).
Acknowl. Nº:	Número de Secuencia esperado en el próximo paquete TCP.
Data:	Los datos útiles o la información en si.

El paquete TCP así estructurado se coloca en el espacio de los datos de un paquete IP.

ESTRUCTURA DE UN PAQUETE DE DATOS TCP/IP



Los datos o información se introducen en cierto sentido en un sobre (paquete TCP), el cual a su vez se introduce en un sobre (paquete IP)



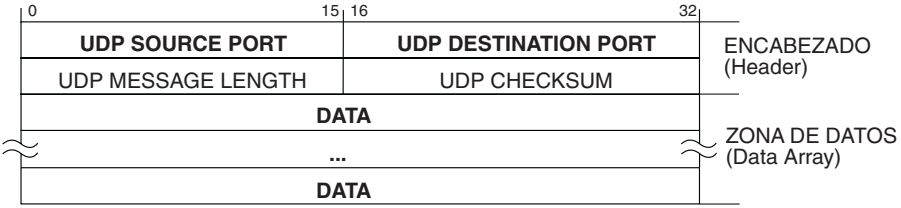
UDP User Datagramm Protocol

UDP es otro protocolo de transporte, que al igual que TCP trabaja por encima de IP. Al contrario de TCP, UDP trabaja sin conexión. Cada paquete de datos se maneja como un único envío y no hay sobre él nigún mensaje de vuelta si el receptor lo ha recibido o no.

Ya que con UDP no se tiene que establecer ninguna conexión o desconexión y por lo tanto no puede ocurrir ninguna situación de Time-out, UDP es más rápido que TCP: Cuando un paquete se pierde, la transmisión de datos sigue sin obstáculos, a no ser que un protocolo superior se ocupe de la repetición.

La seguridad de los datos es bajo UDP en cada caso soportada por el programa de aplicación.

ESTRUCTURA DE UN PAQUETE DE DATOS UDP



Source Port: Número de puerto de la aplicación emisora (puerto de vuelta al enviar el receptor).

Destination Port: Puerto de destino, al que los datos en el receptor deben de ser entregados.

Como pequeña regla se puede decir:

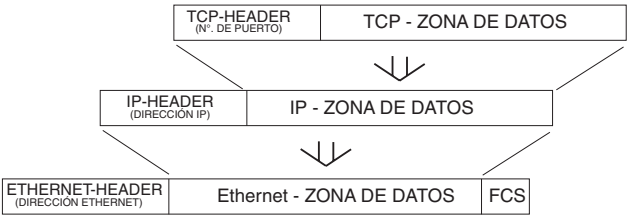
- Para corrientes de datos continuadas o grandes cantidades de datos así como en situaciones en las que una alta medida de seguridad en los datos es exigida, entonces se utiliza por regla general TCP.
- Para repetidos cambios de compañeros de transmisiones así como un soporte de la seguridad de los datos por protocolos superiores se hace favorable y razonable la utilización de UDP.

TCP/IP Ethernet

TCP/IP es un protocolo puramente lógico y necesita siempre un fundamento físico. Como ya se ha mencionado al principio, Ethernet disfruta hoy en día de la mayor extensión de las topologías de red físicas. Así se encuentran también en la mayoría de las redes TCP/IP, Ethernet como fundamento físico.

TCP/IP y Ethernet se funden conjuntamente cuando cada paquete TCP/IP se introduce en el espacio de datos de un paquete Ethernet.

ESTRUCTURA DE UN PAQUETE DE DATOS TCP/IP-ETHERNET



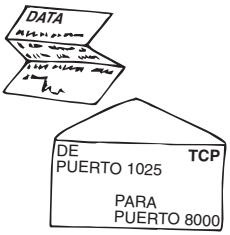
Los datos atraviesan en su camino desde la aplicación en el PC hasta la red por diversos niveles de controladores (drivers):

- El programa aplicación decide a qué otro usuario se debe de enviar la información y pasa la dirección IP y puerto TCP al driver TCP/IP (a menudo también llamado TCP/IP Stack o Pila)

- El driver TCP/IP coordina la construcción de la conexión TCP.

- Los datos entregados por la aplicación se separan por el driver TCP dependiendo del tamaño en pequeños y manejables bloques.

- Cada bloque de datos se empaqueta después por el driver TCP en un paquete TCP.

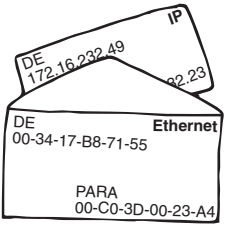


- El driver TCP entrega el paquete TCP y la dirección IP del receptor al driver IP.

- El driver IP empaqueta el paquete TCP en un paquete IP.

- El driver IP busca en la tabla llamada ARP (Address Resolution Protocol) la dirección Ethernet de la dirección IP que se ha entregado como receptor (sobre esto, más tarde más) y entrega el paquete IP junto con la dirección Ethernet conseguida a los driver de la tarjeta Ethernet.

- El driver de la tarjeta Ethernet empaqueta el paquete IP en un paquete Ethernet y lo entrega a través de la tarjeta de red a la propia red.



En el lado del receptor tiene lugar este procedimiento pero en secuencia inversa:

- La tarjeta de red reconoce en la dirección de destino Ethernet, que el paquete es para el usuario de red y lo entrega al driver de ethernet.

- El driver de Ethernet aisla el paquete IP y lo entrega al driver de IP.

- El driver de IP aisla el paquete TCP y lo entrega al driver de TCP.

- El driver de TCP comprueba el contenido del paquete TCP que sea correcto y entrega los datos anexos al número de puerto de la aplicación correcta.

A primera vista parece este procedimiento de transmisión con muchas capas incomprensible y enormemente complicado. Pero la separación de protocolos lógicos (TCP/IP) y protocolos físicos (Ethernet), hace posible el intercambio de datos independientemente de la red y el hardware.

ARP Address Resolution Protocol

Como hemos visto, el driver IP entrega junto al paquete IP también la dirección física Ethernet al driver de la tarjeta Ethernet. Para la emisión de la dirección Ethernet del receptor se sirve el driver IP del protocolo ARP (Address Resolution Protocol).

En cada ordenador con capacidad TCP/IP hay una tabla ARP. La tabla ARP se actualiza en caso de necesidad por el driver TCP/IP y contiene la correspondencia entre dirección IP y dirección Ethernet.

InternetAddress	Physical Address	Type
172.16.232.23	00-80-48-9c-ac-03	dynamic
172.16.232.49	00-c0-3d-00-26-a1	dynamic
172.16.232.92	00-80-48-9c-a3-62	dynamic
172.16.232.98	00-c0-3d-00-1b-26	dynamic
172.16.232.105	00-c0-3d-00-18-bb	dynamic

Cuando se tiene que enviar un paquete IP, mira el driver IP si la dirección IP deseada ya existe en la tabla ARP. Si es así el caso, entonces el driver IP entrega la dirección Ethernet conseguida junto con su paquete IP al driver de la tarjeta Ethernet.

Cuando no se puede encontrar la dirección IP deseada, comienza el driver IP una petición ARP (ARP-Request). Una petición ARP es una llamada general (también llamado Broadcast) a todos los usuarios en la red local.

Para que la llamada general sea reconocida por todos los usuarios de la red, el driver IP entrega como dirección Ethernet FF-FF-FF-FF-FF-FF. Un paquete Ethernet con la dirección FF-FF-FF-FF-FF-FF será leído por todos los usuarios. En el paquete IP se introduce como destino la dirección IP deseada y en el campo Protocol de la cabecera IP (IP-Header) la identificación para el reconocimiento ARP.

Aquel usuario, que en esta petición ARP reconozca su propia dirección IP, lo confirma con una respuesta ARP (ARP-Reply). La respuesta ARP es un paquete tanto a nivel Ethernet como

también a nivel IP direccionado al emisor del ARP-Request con el código ARP en el campo de protocolo.

El driver IP puede ordenar ahora la dirección Ethernet de la dirección IP deseada que el ARP-Reply ha entregado y entonces la incluye en la tabla ARP.

En un caso normal los asientos en la tabla ARP no permanecen para siempre. Cuando un usuario introducido en la tabla no se contacta durante un determinado espacio de tiempo (con Windows aprox. 2 minutos), entonces se borra el correspondiente asiento de la tabla. Esto mantiene la tabla ARP manejable y posibilita el cambio de componentes Hardware bajo la conservación de la dirección IP. También se llama a estos asientos limitados por el tiempo, asientos dinámicos.

Junto a los asientos dinámicos hay también asientos estáticos, que el usuario introduce él mismo en la tabla ARP. Los asientos estáticos se pueden utilizar para que un componente Hardware nuevo, que todavía no tiene una dirección IP, se le pueda asignar la dirección IP deseada.

Esta forma de asignación de direcciones IP es permitida también por los Com-Server: Recibe un Com-Server, que todavía no tiene su dirección IP, un paquete IP que a nivel Ethernet se dirige a él, entonces la dirección IP de ese paquete se analiza y se toma como su propia dirección IP.

Atención: No todos los componentes de red poseen estas capacidades. ¡PCs pe. no se dejan configurar de esta forma!

Ahora ya sabemos, qué informaciones se necesitan para una conexión TCP/IP Ethernet en una propia red local. Lo que todavía falta son las informaciones necesarias que una conexión entre redes diferentes necesitan.

Puerta de enlace (Gateway) y Máscara de subred

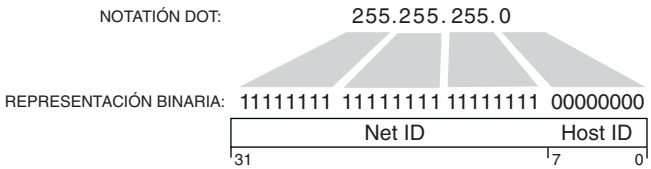
Si el receptor o destinatario, con el cual la conexión debe de ser establecida, se encuentra en la misma red que el emisor, se reconoce en el Net-ID (la parte de la dirección IP que direcciona la red). Si corresponden esas partes de la dirección IP por el emisor y el receptor, entonces se encuentran ambos en la misma red, si no son iguales, entonces está el destinatario en otra red.

Las diferentes redes unitarias se conectan entre ellas mediante Gateways/Routers y así forman en conjunto Internet.

Para las redes de clase A,B y C está claramente definido, que parte de la dirección IP es Net-ID y cuál Host-ID.

Pero también es posible dividir una red, da igual que clase de red sea, en muchas más redes pequeñas. Para el direccionamiento de estas Subredes (Subnets) no es suficiente el Net-ID prefijado de cada clase de red, se tiene que ramificar una parte del Host-ID para el direccionamiento de las subredes. Esto significa, que el Net-ID se agranda y el Host-ID en correspondencia se empequeñece.

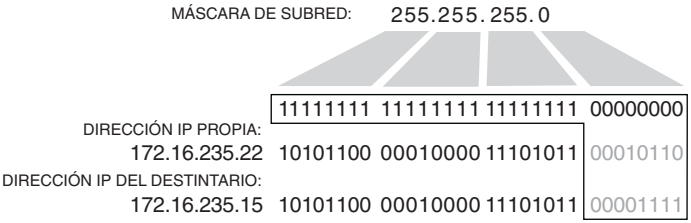
Qué parte de la dirección IP se procesa como Net-ID y cuál como Host-ID lo indica la máscara de subred. La máscara de subred es exactamente como la dirección IP, un valor de 32 bits, que se representa con la notación Dot. Si se observa la máscara de subred en binario, la parte de la Net-ID es con unos y la parte del Host-ID con ceros.



Con cada paquete de datos por enviar, el IP driver compara la dirección IP propia con la del receptor.

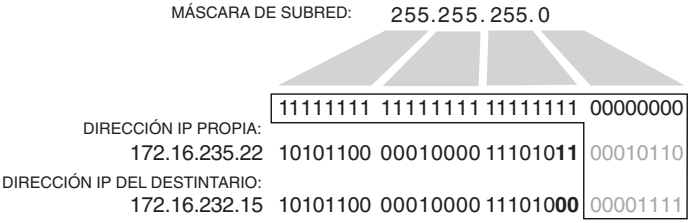
Aquí se suprimen los Bits del Host-ID que quedan cubiertos por los ceros de la parte de la máscara de subred.

Si los Bits procesados son iguales que las dos direcciones IP, entonces se encuentra el usuario elegido en la misma subred.



En el ejemplo representado arriba puede el IP driver emitir la dirección Ethernet con ARP y entregarla al driver de la tarjeta de red para su direccionamiento directo.

Si por el contrario se diferencian en un único de los bits procesados, entonces no se encuentra el usuario en la misma subred. En este caso se tiene que entregar el paquete IP para su siguiente envío en la red destino al Gateway, router o pasarela.



En el paquete IP se introduce la dirección IP del usuario deseado. Pero el IP driver no proporciona con ARP la dirección Ethernet del usuario deseado sino la dirección Ethernet del Router.

Los Gateways, pasarelas así como los routers, en principio, no son nada más que computadoras con dos tarjetas de red. Los paquetes Ethernet, que se reciben en la tarjeta A, se desempaquetan por los driver Ethernet y el paquete IP recibido

se entrega al driver IP. Este comprueba si la dirección IP destino pertenece a la subred que está conectada a la tarjeta B y el paquete se puede entregar directamente o si el paquete IP se entrega a otro Gateway.

Así un paquete de datos puede pasar en su camino de un usuario a otro por varios Gateways / Routers. Mientras que a nivel IP en todo el camino se introduce la dirección del destinatario, a nivel Ethernet siempre se direcciona el siguiente Gateway. Sólo en el último tramo del correspondiente Gateway / Router se coloca como destino en el paquete Ethernet la dirección Ethernet del destinatario.

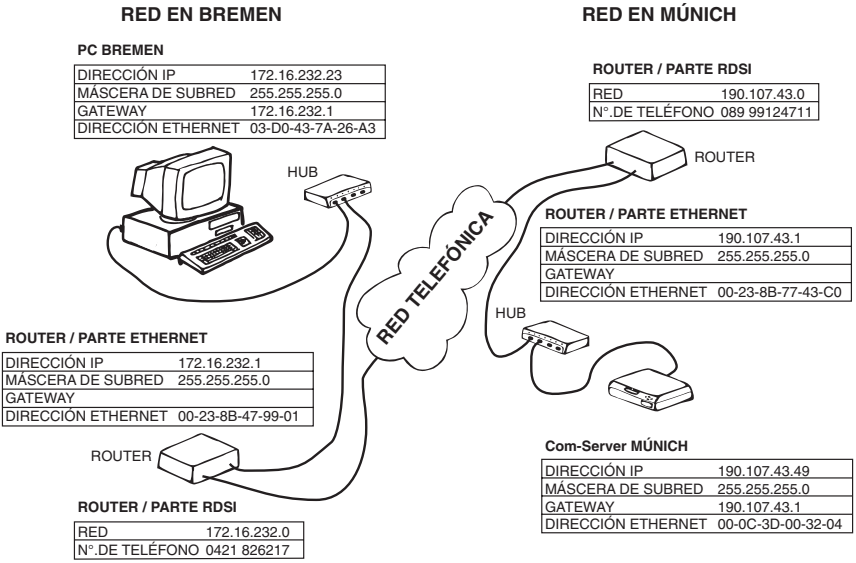
Junto con los Routers, que conexionan una subred Ethernet con otra subred Ethernet, hay también Routers, que cambian el medio físico, pe. de Ethernet a Token Ring o RDSI. Mientras que aquí también el direccionamiento IP permanecería igual sobre todo el camino, es el direccionamiento físico de un router a otro el que se adapta a las condiciones físicas necesarias de estos intervalos.

Entre dos routers RDSI-Ethernet se direccionan por ejemplo con números de teléfono.

Conexiones TCP/IP sobre varias redes

En los siguientes párrafos se va a describir, anexo a una conexión Telnet ya existente, el recorrido de un símbolo sobre una conexión de red enrutada.

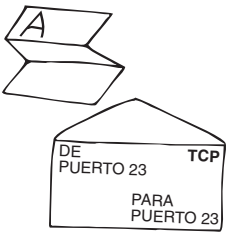
Partimos en nuestro ejemplo de que un Usuario en Bremen ya ha establecido una conexión Telnet a un W&T Com-Server en Munich. La conexión entre las redes de Bremen y Munich consta de una conexión de red enrutada a través de RDSI.



El usuario en Bremen teclea en la aplicación Cliente de Telnet el símbolo „A“.

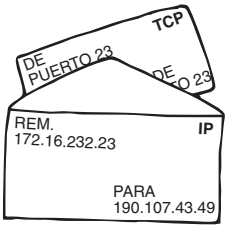
- El programa cliente de Telnet en el PC entrega a la pila TCP/IP la „A“ como los datos de información. La dirección IP del destino (190.107.43.49) y el número de puerto 23 para Telnet se entregaron ya a la pila TCP/IP en el momento del establecimiento de la conexión.

- El driver TCP escribe la „A“ en el espacio para los datos de un paquete TCP y coloca como puerto destino el 23.



- El driver TCP entrega el paquete TCP y la dirección IP del receptor al driver IP.

- El driver IP empaqueta el paquete TCP en un paquete IP.



- El driver IP proporciona mediante la comparación de las partes Net-ID de su dirección IP y de la del destinatario, si el paquete IP se puede entregar en la misma subred o se tiene que entregar a un router.

MÁSCARA DE SUBRED:	255.255.255.0			
	11111111	11111111	11111111	00000000
DIRECCIÓN IP PROPIA:				
172.16.232.23	10101100	00010000	11101000	00010111
DIRECCIÓN IP DEL DESTINTARIO:				
190.107.43.49	10111110	01101011	00101011	00110001

Aquí no son las partes Net-ID de las dos direciones iguales; el paquete IP tiene que ser enviado por lo tanto al router predeterminado.

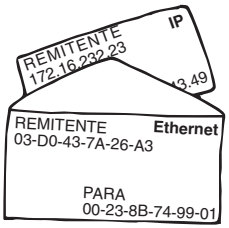
- El driver IP proporciona con ARP la dirección Ethernet del router. Como la conexión TCP ya está constituída,

entonces la dirección IP del router tiene que estar ya resuelta en la tabla ARP.

```
Internet Address      Physical Address   Type
172.16.232.1          00-23-8B-74-99-01  dynamic
172.16.232.49         00-c0-3d-00-26-a1  dynamic
172.16.232.92         00-80-48-9c-a3-62  dynamic
```

- El driver IP coge de la tabla ARP la dirección de Ethernet del router y lo entrega junto con el paquete IP a los drivers de la tarjeta Ethernet.

- El driver de la tarjeta Ethernet empaqueta el paquete IP en un paquete Ethernet y entrega este paquete a través de la tarjeta de red al exterior de ella.



- El router extrae el paquete IP del paquete Ethernet recibido.

- La dirección IP del destinatario se compara con una tabla de enrutamiento. A través de esta tabla de enrutamiento decide el router RDSI, bajo que número se encuentra la red buscada. Ya que la conexión TCP ya está establecida, es muy probable que también la conexión RDSI en este momento esté establecida. Si no fuera éste el caso, el router llama al número extraído de la tabla de enrutamiento y establece la conexión RDSI con el router contrario en la red destino.

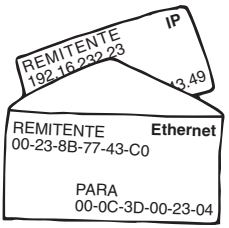
- También en la red RDSI se empaqueta el paquete IP en un marco de informaciones de direcciones. Para nosotros es sólo importante que se recoge sin cambiar su espacio de direcciones y se incluye así en el paquete RDSI.

- El router en la red destino extrae el paquete IP del paquete RDSI recibido. Con las direcciones IP y máscara de subred se decide si el paquete IP recibido se puede entregar en la subred local o se tiene que entregar a otro router.

MÁSCARA DE SUBRED:	255.	255.	255.	0
	11111111	11111111	11111111	00000000
DIRECCIÓN IP PROPIA: 190.107.43.1	10111110	01101011	00101011	00000001
DIRECCIÓN IP DEL DESTINTARIO: 190.107.43.49	10111110	01101011	00101011	00110001

En nuestro ejemplo el paquete IP ha alcanzado la red destino y se puede direccionar en la red local por Ethernet.

- El router, que internamente también tiene una tabla ARP, proporciona con ARP y la dirección IP la correspondiente dirección Ethernet y lo empaqueta con el paquete IP todavía sin cambiar en un paquete Ethernet.



- El Com-Server reconoce en la dirección destino de Ethernet, que el paquete es concretamente para él y extrae el paquete IP.

- El driver IP del Com-Server aisla el paquete TCP y lo entrega al driver TCP.

- El driver TCP comprueba el contenido del paquete TCP para que sea correcto y entrega los datos, en este caso la „A“ al driver Serie.

- El driver Serie entrega la „A“ al interfaz Serie.

En una conexión TCP se confirma la correcta recepción de un paquete de datos con el envío de vuelta de un número de reconocimiento (Acknowledgement nº.) El paquete de confirmación atraviesa el mismo camino de transmisión y con ello también todos los procedimientos a la inversa. Todo esto sucede en un intervalo de tiempo de milisegundos.

DHCP Dynamic Host Configuration Protocol

Como recordatorio: Cada aparato terminal de Ethernet tiene una dirección Ethernet única en todo el mundo (dirección MAC) que viene dada por el fabricante y no se puede cambiar. Para el funcionamiento en redes TCP/IP entrega el administrador de red al aparato terminal una dirección IP acorde a la red.

Si no se utiliza DHCP, se entregan las direcciones IP „clásicamente“:

- Para aparatos, que permiten la interacción directa del usuario (pe. PCs), se puede entregar el número IP directamente en un menú de configuración correspondiente.
- Para aparatos „Black-Box“ cajas negras (pe. Com-Server) existe para unos el procedimiento del ARP a través de la red, para otros existe la posibilidad de introducir la información de configuración a través de un interfaz serie.

Además de la dirección IP se tienen que configurar también otros parámetros como la máscara de subred, Gateway así como un servidor DNS (más sobre esto en el capítulo siguiente). En redes grandes con muchos y diferentes aparatos terminales conlleva todo esto un gran esfuerzo en medios, administración y configuración.

Con DHCP se le ofrece al administrador de red una herramienta, con la que las configuraciones de red de cada aparato son automáticas, únicas y centrales.

Para el uso de DHCP se necesita en la red por lo menos un servidor DHCP, que gestione los datos de configuración para un rango predeterminado de direcciones IP. Aparatos terminales con capacidad DHCP preguntan al arrancar (Booting) a este servidor por su dirección IP y también por los correspondientes parámetros como máscara de subred y Gateway. Los servidores DHCP tienen tres posibilidades básicas de reparto de las direcciones IP y configuración:

Concesión de la dirección IP desde un grupo de direcciones

En el servidor DHCP se fija un rango de direcciones IP del cual se reparte una dirección, en ese momento sin usar, para el usuario que pregunte por una dirección. El reparto es en este procedimiento normalmente limitado en el tiempo, donde la duración de uso (Lease-Time) es fijado por el administrador de red o incluso es totalmente desactivado. De esta forma se permite guardar datos importantes (Lease-Time, máscara de subred, Gateway, Servidor DNS, etc...) en un perfil de configuración, que es válido para todos los aparatos terminales, que se tienen que servir del grupo de direcciones.

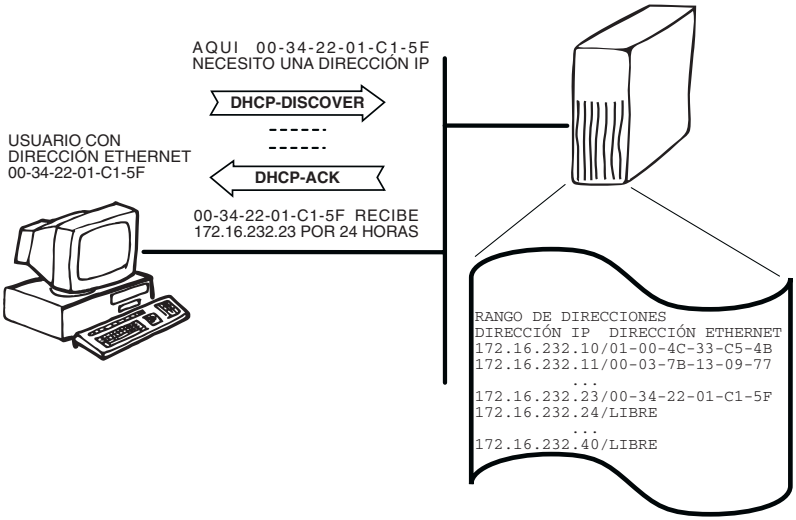
Ventajas pequeño esfuerzo de administración; los usuarios pueden estar en red en diferentes lugares con el mismo terminal sin esfuerzos de configuración.

Mientras que no estén todos los terminales al mismo tiempo activos en la red, puede ser el número de posibles aparatos mayor que el número de direcciones IP disponibles.

Desventajas Un usuario no puede ser identificado por su dirección IP, ya que el orden claro de dirección IP y aparato terminal se pierde.

Puede ocurrir que un terminal en cada comienzo se le asigne una dirección IP diferente.

Ejemplo: Casos típicos para la entrega de direcciones IP de un grupo de direcciones, son las redes de las universidades. Aquí existen redes con un número casi ilimitado de usuarios potenciales pero de los cuales sólo en verdad unos pocos trabajan en red. Gracias a DHCP los estudiantes tienen la posibilidad de llevarse su ordenador portátil sin cambiar la configuración, de un laboratorio a otro y de conectarlo a la red.



Concesión de una dirección IP reservada

El administrador de la red tiene la posibilidad de reservar direcciones únicas IP para determinados terminales. En el servidor DHCP se ordena la dirección IP con la dirección Ethernet del terminal de red; para cada dirección IP reservada se puede además almacenar un perfil de configuración individual. La introducción de un tiempo de uso „Lease-Time“ es en este caso nada razonable (pero de todas formas es posible), ya que la dirección IP siempre estará asignada al uso de este terminal.

Ventajas: Aunque sea una configuración individual, se permite solucionar todas las configuraciones de red en un puesto central y no se tienen que llevar acabo en el terminal mismo.

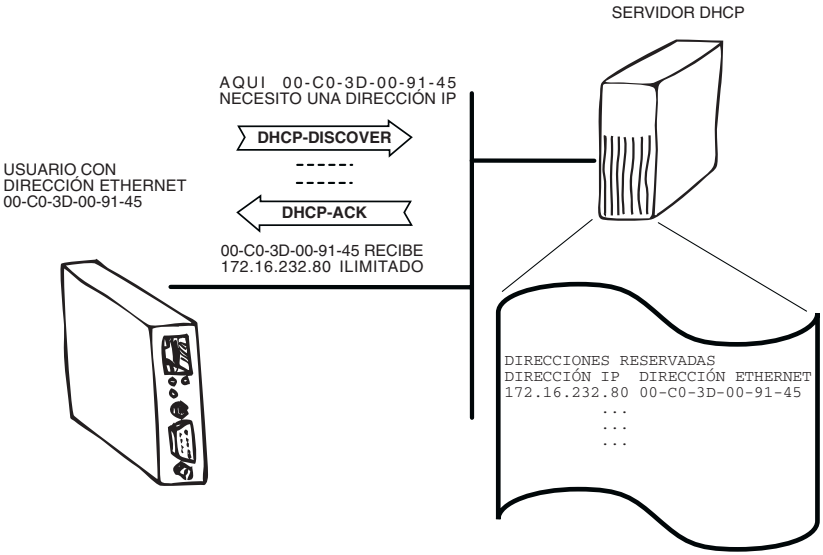
Los Terminales de red pueden ser directamente preguntados por su propia dirección IP.

Desventajas: Ya que para cada terminal hay que introducir configuraciones específicas, aumenta el trabajo de administración.

En intercambio de terminales de red se tienen que introducir de nuevo en el servidor DHCP en el perfil de configuración por lo menos la dirección de Ethernet.

Ejemplo: Configuraciones de terminales con capacidades DHCP como servidores de impresora o Com-Servers, para los cuales dependiendo del uso necesitan un direccionamiento por dirección IP. En el control de DHCP se introduce en la dirección IP reservada la dirección Ethernet del correspondiente terminal de red; el Lease Time debería estar desactivado. Con los Com-Servers se puede introducir como parámetros adicionales máscara de subred y Gateway (Router).

En este punto hay que aclarar que muchos terminales también usan el viejo protocolo BootP para averiguar su configuración. BootP es un predecesor de DHCP y es soportado por servidores DHCP. Pero BootP sólo puede trabajar con direcciones IP reservadas.



Con aparatos „Black Box“ como el Com-Server se debería de utilizar el protocolo BootP, para obligar en cada caso la entrega de una dirección IP reservada. Si en el servidor DHCP no hay ninguna dirección Ethernet para el Com-Server, entonces se ignora la petición de BootP y el Com-Server mantiene la dirección IP que en ese momento está configurada.

Los servidores DHCP con Windows 2000 asignan direcciones IP del rango normal determinado con peticiones BootP.Esta característica se deja desactivar, ¡su administrador de red debería hacerlo a toda costa!

Exclusión de determinadas direcciones IP del configurador DHCP

Para terminales que no son capaces de soportar DHCP ni tampoco BootP, tiene el administrador de red la posibilidad de excluir unas direcciones IP o incluso completos rangos de direcciones para la disponibilidad de entrega con DHCP.

La configuración tiene que llevarse a cabo en este caso en el terminal mismo o con la utilización de herramientas incluídas con el terminal.

Desventajas: Configuración descentralizada y no unitaria; se necesita un increíble esfuerzo en administración.

Ejemplo: PCs con versiones antiguas de DOS o viejos servidores de impresora y viejos Com-Server no son capaces de soportar DHCP y tienen por lo tanto que configurarse „a mano“.

Los tres procedimientos se pueden utilizar en conjunto con la ayuda de DHCP.

DHCP y Router

El intercambio de información entre terminales y servidores DHCP se realiza a nivel físico en forma de UDP-Broadcast (llamada general en la red). Si se extiende la configuración DHCP por varias subredes, se deben de elegir routers apropiados para que estos también permitan el paso a los DHCP-Broadcasts.

DNS el sistema de nombres de dominio

El sistema de nombres de dominio (Domain Name System o DNS) es el directorio o guía de Internet. Aunque por el usuario sólo se utilice en el transfondo, es de todas formas uno de los servicios más importantes de Internet.

A nivel IP se interrelacionan millones de usuarios en Internet sobre las direcciones IP. Pero para los usuarios sería el trato con direcciones IP un poco difícil: ¿quién se puede acordar de que el termómetro Web de W&T se puede encontrar con la dirección IP 195.8.247.225? Mientras que un nombre más conciso como www.klima.wut.de, se puede recordar más facilmente.

Ya en el comienzo de Internet se llevaba la cuenta de la necesidad de una correspondencia entre nombres simbólicos y sus direcciones IP. En cada computadora local se cuidaba una tabla de Hosts, en la cual se almacenaban las correspondencias determinadas. La desventaja era que sólo se podían localizar los usuarios que estaban en la lista local. Además las listas locales engordaron con el increíble crecimiento de Internet y enseguida tomaron dimensiones que no se podían manejar. Se necesitaba por lo tanto encontrar un sistema único de solución de nombres. Por esta razón se sacó en 1984 el estándar DNS, en el cual hasta hoy casi no se ha cambiado nada.

El principio es simple, la ordenación de direcciones IP y nombres de dominio se almacena en los llamados servidores de DNS y allí es donde se pregunta en caso de necesidad. Pero vayamos un poco a los detalles, algunos comentarios para la estructura de los nombres de dominio:

Nombres de dominio

El DNS sigue una concesión uniforme de nombres, por la cual cada Host (usuario en la red), es parte por lo menos de un dominio superior „Top Level Domain“.

Como Top Level Domain se ofrecen nombres de países determinados:

- *.de* para Alemania
- *.at* para Austria
- *.ch* para Suiza
- *.es* para España

El dominio se puede también elegir por el contenido o sociedad:

- *.com* para ofertas comerciales
- *.net* para operadores de red
- *.edu* para organizaciones educativas o formación
- *.gov* para el gobierno de los EEUU
- *.mil* es para los militares de EEUU
- *.org* es para organizaciones

Todos los nombres de dominio inferior o subdominios se pueden elegir por el operador mismo, pero tienen que ser únicos en el dominio superior. Para cada Top Level Domain hay una institución propia de administración, a la cual hay que solicitar los subdominios y así asegurar que no se conceden más de una vez. Para el dominio *.de* es responsable en estas premisas el DENIC (*De*utsches *N*etwork *I*nformation *C*enter; *http://www.denic.de*). Para España es el ES-NIC *http://www.nic.es.*

Un ejemplo: www.klima.wut.de se forma por:

- *de* para Alemania (Deutschland) como Top Level Domain.
- *wut* para Wiesemann y Theis como Sub Level Domain.
- *www.klima* para el termómetro Web en el dominio *wut.de*.

El nombre de dominio completo tiene que tener como máximo 255 caracteres, donde cada nombre de subdominio puede tener máximo 63 caracteres. Los subdominios individuales se separan con puntos. No hay una diferenciación entre mayúsculas y minúsculas. *WWW.WUT.DE* lleva a la página principal de W&T al igual que *www.wut.de* o *www.WuT.de*.

Resolución de nombres en DNS

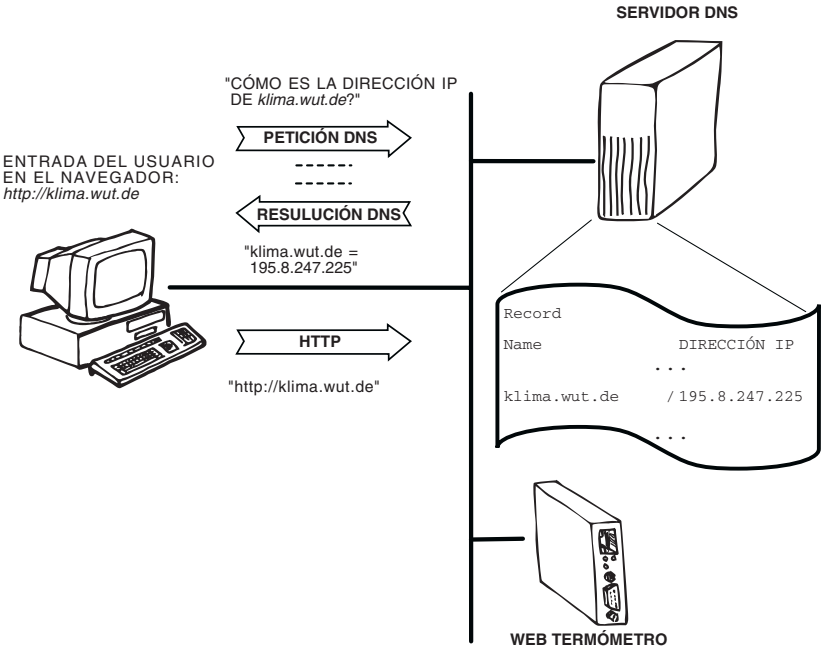
Como ya se ha aclarado, se dirijen listados con la ordenación de nombres de dominio y direcciones IP en los servidores DNS (también llamados servidores de nombres). Si sólo hubiese un único servidor DNS con la densidad actual de Internet, entonces estaría probablemente con la inmensa cantidad de peticiones DNS sobre exigido. Por esta razón se distribuye Internet en zonas, para la cual uno o varios servidores DNS son responsables.

Los usuarios, que quieran usar el DNS, tienen que introducir en su TCP/IP-stack la dirección IP de un servidor DNS de su zona. Para poder trabajar también en caso de fallo de este servidor, exigen los normales TCP/IP-stacks incluso la introducción de un segundo servidor DNS.

Que servidor DNS es responsable del correspondiente usuario se averigua por el proveedor o por el administrador de red.

Para poder resolver nombres de dominio en direcciones IP, disponen los actuales TCP/IP-stacks de un programa de resolución. Introduce el usuario en lugar de una dirección IP un nombre de dominio, comienza el programa de resolución una petición al servidor DNS configurado. Si no hay allí ninguna inscripción para el dominio buscado, se envía la petición hacia el próximo servidor DNS más alto en la jerarquía. Esto ocurre tantas veces hasta que la petición o se resuelve o se averigua que no existe el nombre de dominio solicitado.

La dirección IP perteneciente al nombre de dominio se envía de vuelta de servidor DNS a servidor DNS y finalmente al programa de resolución. El TCP/IP-stack puede tomar ahora el direccionamiento del usuario con toda normalidad con su dirección IP.



La correspondencia entre dirección IP y el nombre de dominio se almacena por el TCP/IP-stack en un Cache. Estas inscripciones en el Cache son dinámicas: si el usuario almacenado no se utiliza en un determinado espacio de tiempo, el stack borra la inscripción. Esto mantiene el cache manejable y posibilita, que una dirección IP con su correspondiente dominio se pueda cambiar.

DNS en sistemas empotrados (embedded)

Los sistemas empotrados (embedded) no ofrecen por regla general la posibilidad de configurar al aparato mismo con un nombre de dominio.

Esto no es de todas formas necesario, ya que el aparato terminal no tiene que conocer su propio nombre. Incluso en estos casos se mantiene también la dirección IP del correspondiente nombre en el servidor DNS. Si se tiene que realizar una conexión, por ejemplo, de un cliente con un servidor en un

sistema empotrado, el cliente realiza la petición de número de IP con el correspondiente nombre al servidor DNS como lo hemos descrito anteriormente.

Los sistemas empotrados trabajan a menudo en conexiones máquina-máquina en lugar de hombre-máquina, ya que aquí un direccionamiento directo con número de IP es más eficiente, debido a que el tiempo de resolución del DNS desaparece.

El direccionamiento con nombres es razonable con sistemas empotrados sólo cuando, o el nombre es lo único conocido (p.e. direcciones de e-mail) o cuando hay que contar con una „mudanza“ del servidor (el nombre se mantiene, la dirección IP cambia, como es el caso de un servidor web).

DHCP y DNS

Mientras que DHCP gestiona la correspondencia entre terminales físicos, es decir la dirección Ethernet con la dirección IP, son los servidores DNS los que almacenan la correspondencia entre direcciones IP y sus nombres de dominio. Aunque en el caso de la cesión dinámica de IP con DHCP sería razonable un ajuste automático con DNS, lamentablemente no se produce.

Para repartos fijos de direcciones IP se pueden sincronizar los asientos o entradas de los servidores DHCP y DNS a mano o con herramientas adicionales. Para el reparto dinámico de direcciones IP sólo se puede realizar el cuidado de los asientos DNS correspondientes mediante la función de herramientas adicionales.

Para los sistemas UNIX hay servidores DHCP, que antes del comienzo de la entrega de direcciones permiten que se realice una ordenación de direcciones Ethernet con nombres en lugar de direcciones Ethernet con números IP. De esta forma se mantiene en el servidor DNS la correspondencia de nombre con dirección IP. La entrega de direcciones se realiza de la siguiente manera:

1. El terminal intenta conseguir una dirección IP del servidor DHCP.
2. Debido a la dirección Ethernet del terminal, el servidor DHCP encuentra el nombre que pertenece a ese terminal de red.
3. El servidor DHCP averigua en el servidor DNS la dirección IP que pertenece a ese nombre.
4. El servidor DHCP entrega al terminal la dirección IP que ha resuelto el servidor DNS.

   Para la siguiente generación de sistemas operativos está planeado la introducción de un DNS ampliado, el DNS dinámico (abreviado DDNS). Con el DDNS tiene que darse un ajuste entre DNS y DHCP, de tal forma que también los terminales reciban su dirección desde un grupo de direcciones y que sean también localizables con nombres de dominios.

Lamentablemente sólo ha habido sobre este tema anuncios pero todavía ninguna información concreta.

Otros Protocolos y Servicios

La función básica de TCP/IP-Ethernet estaría aclarada ya, pero en las redes se puede encontrar con una gran cantidad de otros protocolos y servicios.

En este apartado se encontrará con, cómo funciona un e-mail, qué ocurre al solicitar una página Web y qué otros protocolos y servicios importantes se encontrará en relación con TCP/IP-Ethernet.

WWW World Wide Web

En los primeros 20 años de su existencia, ha sido el uso de Internet para la gente de a pie nada interesante. Un grupo pequeño, en comparación a la actualidad, de personas enteradas tenía que teclear filas de órdenes encriptadas para poder intercambiar información.

Fué entonces con el desarrollo del estándar WWW cuando se abrió Internet a un público cada vez más amplio. Para que el usuario pueda aprovechar las posibilidades de WWW, necesita un navegador de Internet o Internet Browser. Un programa cliente que mediante una ventana gráfica muestra los contenidos de las páginas Web, que están almacenadas en servidores WWW.

Si se traduce la palabra inglesa „browse“ al español, significa „hojear“ y es exactamente esa la filosofía que se esconde en WWW.

El usuario tiene que poder navegar a través de una inmensa telaraña de información con un mínimo de esfuerzo en su manejo. Y todo ésto fácilmente con una pulsación en el ratón: no son necesarios para ello bastos y asentados conocimientos en Computadoras y Redes.

Las páginas Web se presentan como Hipertexto y pueden contener junto a informaciones de texto también referencias a fotos, gráficos y otros contenidos multimedia. Cómo se deben de mostrar todos estos elementos en el navegador, viene descrito en el hipertexto.

Pero el logro más importante de WWW es la „interconexión“ de contenidos. Cada elemento de una página Web se puede equipar con un „hiperlink“, una referencia a otra página Web. Pulsa el usuario con el ratón sobre uno de estos elementos interconectados, se abre en el navegador automáticamente la página Web deseada. El usuario puede así saltar de aquí para allá con el ratón en una red de páginas y otros contenidos.

Los fundamentos básicos de la World Wide Web lo forman tres cosas:

- **URL Uniform Resource Locator**
  Con el URL le indica el usuario al Browser, qué protocolo se va a utilizar, en qué servidor se sitúa la página y dónde en el servidor se encuentra la página.
- **HTML Hypertext Markup Language**
  Un lenguaje descriptivo de páginas, que fija mediante palabras clave cómo se muestran los contenidos en el navegador, dónde se encuentran los elementos multimedia y qué elementos y de qué forma están interconexionados.
- **HTTP Hypertext Transfer Protocol**
  El protocolo HTTP regula las exigencias y transmisión de contenidos Web entre el servidor HTTP y el navegador.

URL Uniform Resource Locator

Una premisa para que el usuario se encuentre a gusto en la WWW es que exista un esquema de direcciones unitario. Esta tarea la toma el URL, que normalmente tiene este formato:

```
protocolo://hostname[:tcp-port][/pfadname][/nombre de fichero][?otros parámetros]
```

La introducción del protocolo y el nombre del Host es en cada caso necesaria; los otros parámetros son opcionales.

Protocolo

De todos aquellos protocolos que la mayoría de navegadores soportan, mostramos aquí sólo los tres más importantes:

HTTP se utiliza para el acceso a páginas Web y es *el* protocolo WWW por autonomasia. Por ejemplo: *http://www.wut.de* abre la página principal de Wiesemann & Theis.

FTP sirve para la transferencia de ficheros y se usa para la carga y descarga (Up-y Download) de

ficheros completos.

Ejemplo: *ftp://www.wut.de/download/anleitg/tcpip_anf.pdf* comienza la descarga del fichero TCPIP_ANF.PDF. Con FTP hay que tener en cuenta, que el usuario en caso necesario debe de poseer permiso de acceso para realizar la acción deseada.

Telnet dispone al navegador para que realice una conexión telnet y abra una sesión Telnet-Cliente con el Host introducido.Telnet se utiliza a menudo para configurar sistemas empotrados a través de la red. Por ejemplo si se introduce en el Browser como URL: *telnet:// <dirección IP de un W&T Com-Server>:1111*, entonces se conecta directamente con el puerto de configuración del aparato.

Nombre del Host

Aquí se introduce el nombre del Host (Hostname) del servidor o su dirección IP, con el cual se

quiere establecer una conexión. El Hostname del servidor W&T es *http://www.wut.de*

Con el protocolo y el nombre del host no es importante si se escriben en mayúsculas o minúsculas.

HTTP://www.wut.de lleva igualmente a la página principal de W&T como *http://www.WuT.de.*

TCP Port

Algunos protocolos estándares acceden normalmente bajo TCP a puertos fijos:

HTTP	Puerto 80
FTP	Puerto 21
TELNET	Puerto 23

Si el usuario quiere realizar la conexión con otro puerto, lo puede hacer en el URL con el parámetro TCP-Port

Ejemplo: *Con telnet://<dirección IP de un W&T Com-Server>:1111* dirijirá el acceso al puerto de configuración de un W&T Com-Server.

Pfadname
En un servidor WWW se pueden archivar los contenidos exactamente igual que en un ordenador local en diferentes directorios y carpetas. El Pfadname muestra por lo tanto, dónde se encuentran en el servidor los contenidos deseados.

Filename
Representa el nombre del fichero, al que se quiere acceder.

¡Con el Pfadname y el nombre del fichero hay diferencias entre ellos si se escriben en mayúsculas o minúsculas!

Renuncia el usuario con el uso del protocolo HTTP a la introducción de un fichero, entonces se accede directamente a un fichero llamado *index.html* o *default.html*, siempre y cuando estos existan. Ejemplo: *http://www.wut.de* corresponde a *http://www.wut.de/index.html*.

Otros parámetros
Todos los datos después del símbolo de interrogación se pasarán como parámetros a la aplicación que esté funcionando en el servidor www (más sobre esto posteriormente).

HTML Hypertext Markup Language

Uno de los problemas en WWW fué al principio la multitud de diferentes ordenadores y sistemas operativos. No existía un único interfaz software a nivel de usuario. De esta necesidad se tenía que desarrollar uno, que también fuera para los profanos fácil de manejar y que en diferentes ordenadores se mostrara lo mismo, surgió HTML.

HTML es un lenguaje de remarcado (Markup Language) que se compone de palabras clave, también llamados rótulos, y del contenido a mostrar. Los rótulos introducen de que forma y cómo se muestra el texto siguiente. Así se permite fijar pe. el tamaño de la letra, fuente y dirección. Los contenidos pueden ser representados en tablas o de forma numérica. El color del texto y fondo se pueden también fijar, etc.

Junto al texto se pueden mostrar también con ayuda de HTML, gráficos e incluso contenidos multimedia como música, lenguaje o secuencias de cine, todo ello se puede integrar con HTML. El documento HTML en si transporta exclusivamente contenidos textuales. Para cada uno de los otros elementos a representar se comunica vía HTML desde dónde se puede cargar, dónde se debe mostrar en la pantalla y en que tamaño debe de representarse.

La característica más importante de HTML es que todos los elementos se pueden configurar con una referencia, también llamado *Hyperlink* o abreviado *Link*. Pulsa el usuario con el ratón sobre uno de estos elementos entonces será automáticamente reconducido a otra página Web, recibe un gráfico o comienza una descarga de fichero.

Con las aclaraciones de los rótulos de HTML existentes se podrían llenar libros enteros. Por ello nos limitamos aquí a los rótulos básicos y características de HTML.

Para los rótulos HTML es válido una estructura fija:

- Cada rótulo está „empaquetado“ en símbolos de mayor y menor. *<Rótulo HTML>*
- El rótulo se puede completar con la introducción de parámetros.
  <Rótulo HTML Attribut = „xy“>
- Para cada rótulo HTML hay un correspondiente final de rótulo, que se señala con una barra de división. *</Rótulo HTML>*
- Las características definidas por un rótulo son válidas para todo aquello que esté entre el rótulo y su final de rótulo.
  <Rótulo HTML> Espacio de validez </Rótulo HTML>
- Con los rótulos HTML no se hace distinción entre mayúsculas y minúsculas.
  <HTML> significa lo mismo que <html>

Estructura básica de un fichero HTML

Cada fichero HTML comienza con *<HTML>* y termina con *</HTML>*. Se diferencia para el resto de la estructura de una página, entre cabecera y cuerpo.

Todos los datos en la cabecera permanecen invisibles para el observador y contienen características de la página, que no influyen directamente con la presentación. La única excepción es el título, que se muestra en la barra de título de la ventana del navegador. Las informaciones de la cabecera se encuentran entre los rótulos *<head>* y *</head>*.

Después de la cabecera le sigue el cuerpo de la página, que es introducido por el rótulo *<body>*. En el cuerpo de la página HTML se encuentran todos los datos que realmente conciernen al contenido de la página y su representación. El final del cuerpo termina con *</body>*.

Aquí un ejemplo fácil:

```
<html>
    <head>
        <title> Willkommen (Bienvenido) </title>
    </head>
    <body bgcolor="#FFFFFF">
        Willkommen bei WUT.de (Bienvenido a WUT.de)
    </body)
</html>
```

Tenga en cuenta que en el rótulo *<body>* se ha introducido el atributo *bgcolor="#FFFFFF"* que es para un fondo blanco. En el navegador se muestra así:

Hyperlinks

La mayor ventaja de HTML es la posibilidad de equipar elementos contextuales únicos con un hyperlink. Pulsa el usuario sobre uno de estos conexionados elementos, entonces será reconducido a otra página Web.

Ampliamos a nuestro Código HTML en un hyperlink:

```
<body bgcolor = „#FFFFFF“>
    Willkommen bei <a href = „http://www.wut.de/index.html“> WuT.de </a>
</body>
```

Con una pulsación en el ratón sobre „WuT.de“ nos conducen ahora a la página principal de W&T.

El atributo de dirección del rótulo *<a href = „Introducción del Pfad“>* puede contener la dirección de forma absoluta o relativa.

Absoluta: Se introduce el URL completo, al que el hyperlink hace referencia.

Relativa: Se introduce sólo el nombre del fichero al que se debe acceder. El fichero se buscará así en el mismo directorio, en el cual se encuentra el fichero HTML actual.

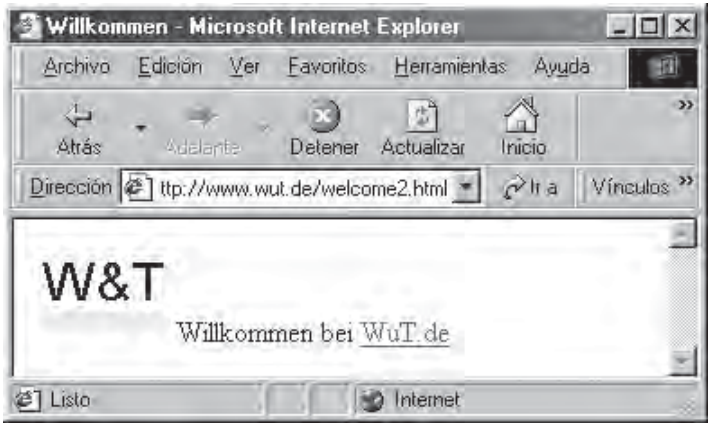
Representación de contenidos multimedia

Como ya se ha comentado, HTML permite la presentación de contenidos que no son componentes del documento HTML, sino que se tienen que cargar desde otro sitio. Para la conexión con ficheros de imágenes, HTML pone a disposición el rótulo *<img>*, donde con el atributo *src* se introducen el nombre y fuente del fichero con las imágenes.

Completaremos nuestro documento HTML con una imagen:

```
<body bgcolor = „#FFFFFF“>
    <img src = „http://www.wut.de/kpics/logo5.gif“>
Willkommen bei <a href  = „http://www.wut.de/index.html“> WuT.de </a>
</body>
```

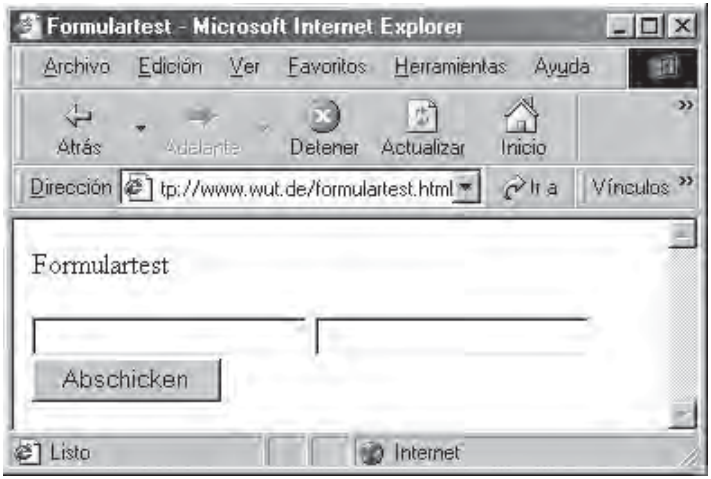
Ahora se representará junto al texto un logotipo en formato GIF, que se cargará desde el directorio *kpics* del servidor web de W&T. El camino del fichero se puede introducir como en el hyperlink en absoluto o relativo.



HTML es un lenguaje puramente representativo, que genera un visor estático en el navegador. Pero ¿cómo funciona cuando el usuario quiere enviar información al servidor WWW?

Como solución a esto, HTML ofrece la posibilidad de mostrar formularios, que se pueden rellenar por el usuario. Estas informaciones que se han introducido, se pueden enviar al servidor WWW con un botón del navegador llamado „Submit-Button“.

Aquí un breve ejemplo:



En el código HTML sería así:

```
<html>
    <head>
        <title> Formulartest</title>
    </head>
    <body bgcolor = „#FFFFFF“>
    Formulartest
    <form method = „post“ action =“Formularauswertung.cgi“ name =“FORMULAR1">
        <input type=“text“ name =“EINGABEFELD1">
        <input type=“text“ name =“EINGABEFELD2">
        <input type=“submit“ name =“submit“ value = „Abschicken“>
    </form>
    </body>
</html>
```

Todos los elementos pertenecientes al formulario se encuentran entre el rótulo *<form>* que lo inicia y el rótulo *</form>* que lo finaliza.

Los atributos del rótulo form son:

method señala cómo http pasará los datos al servidor WWW.
action fija, a qué proceso en el servidor se pasarán los datos.
name se puede pasar arbitrariamente y señala el proceso en el servidor y de qué formulario proceden los datos. (Un proceso puede procesar muchos formularios)

Los elementos de entrada se fijan con el rótulo *<input>*, donde el atributo *type* proporciona de qué tipo de elemento de entrada se está tratando. Posibles atributos pueden ser:

text Campo de entrada de texto.
checkbox Casilla de verificación.
radio Botón de opciones.
submit Botón para enviar o volver a cargar el formulario vacío.

Con el atributo name se le puede asignar al elemento un nombre significativo (comparable con el nombre de una variable); con el atributo value se le puede asignar un valor inicial.

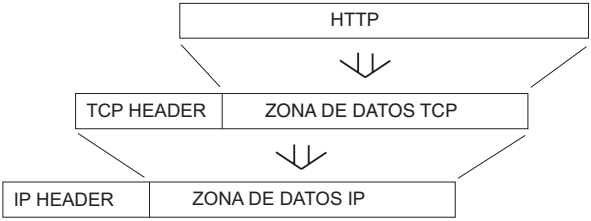
La fila *<input type=„text“ name= „EINGABEFELD1“ value=“test1"*> llevaría por ejemplo a que al abrir el formulario en el navegador, ya estuviera presente en el primer campo de datos el texto *test1*.

Lo que sucede con la información enviada por el formulario, si el usuario recibe una respuesta y cómo es ésta, lo decide sólo el proceso en el servidor WWW, que recibe la información y la procesa.

Como ya se ha dicho, no queremos entrar aquí hasta el último detalle de HTML. El que quiera crear páginas Web, si debería ocuparse entonces con este tema. Una fuente excelente para más información sobre HTML se encuentra en *http://www.teamone.de/selfhtml/selfhtml.htm*; también por supuesto es muy útil la página Web del consorcio W3, que es el cuerpo estándar de normas en temas HTML. (*http://www.W3.org*).

HTTP Hypertext Transfer Protocol

Por el rapidísimo crecimiento de usuarios de www, HTTP es actualmente el protocolo más utilizado con distancia en Internet. HTTP se basa sobre TCP como protocolo base, donde normalmente se utiliza el puerto TCP 80 (es posible el uso de otros puertos diferentes, pero se tienen que especificar en el URL).

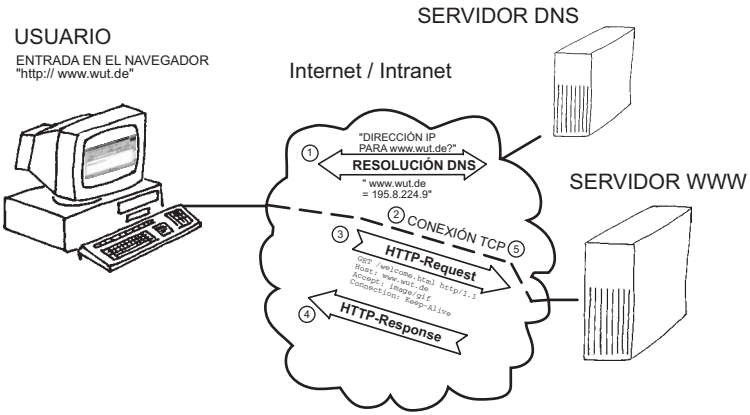


La petición y transmisión de una página Web sucede en cuatro pasos:

1. **Resolución del Host introducido y nombre de dominio en una dirección IP**
   La pila TCP/IP (TCP/IP Stack) comienza una petición DNS para proporcionar la dirección IP del servidor deseado.
2. **Establecimiento de la conexión TCP**
   Recordar: En una conexión TCP es válido el principio Cliente-Servidor. Con HTTP el navegador tiene el papel de cliente y realiza la conexión TCP al servidor WWW deseado.
3. **Envío de la petición HTTP**
   Después del establecimiento exitoso de la conexión TCP, pide el navegador al servidor WWW la página Web deseada. En este punto comienza realmente el protocolo HTTP: El navegador envía el comando GET con los parámetros necesarios al servidor WWW.
4. **Envío de la página Web solicitada**
   El servidor WWW envía primero una confirmación HTTP y entonces la página Web en si misma.
5. **Finalización de la conexión TCP por el servidor WWW**
   Una peculiaridad con HTTP es que la conexión TCP no se termina por el cliente como normalmente sucede, sino que

es finalizada por el servidor. Existen dos razones para este comportamiento:

- El servidor WWW señala al navegador de una forma sencilla que la transmisión se ha terminado. Una página Web recibida se mostrará entonces por lo tanto cuando la conexión TCP esté terminada.
- Los servidores WWW tienen que manejar una cantidad grande de conexiones TCP al mismo tiempo. Por ello, cada conexión abierta en el servidor exige una determinada medida en rendimiento. Para mantener el tiempo de conexión tan corto como sea posible, el servidor termina sencillamente la conexión, tan pronto como todos los datos requeridos se han transmitido.



Los comandos y parámetros más importantes de HTTP

Como ya hemos dicho, HTTP se basa también en el principio Cliente-Servidor: El navegador como cliente puede controlar la comunicación mediante el envío de comandos concretos.

El comando GET

El comando más usado con diferencia es la petición GET, que introduce a cada petición de una página Web. GET exige al servidor HTTP el envío de un documento o elemento y es por ello el comando más importante.

Para el servicio de GET se necesitan algunos parámetros; se habla también de una línea de comandos (inglés Request line).

```
GET /pfadname/filename http-Version
```

Otros parámetros se pueden enviar en otra fila más. Estos parámetros anexos se denominan también cabecera (Header).

Host Nombre del Host (sólo necesario para HTTP 1.1)

Accept fija con qué formatos puede trabajar el navegador. Con *accept:image/gif* da a conocer el navegador pe. que puede mostrar las imágenes en formato GIF.

Connection Con este parámetro (*Connection:Keep-Alive*) se puede fijar por el navegador si la conexión TCP se debe de mantener abierta para cargar otros elementos.

Muchos otros parámetros están descritos en el RFC2616, que se puede ver en *http://www.w3.org/Protocolos/rfc2616/rfc2616.html.*

Un comando típico GET podría tener esta forma:

```
GET       /Welcome.html       http/1.1
Host: www.wut.de
Accept: image/gif
Connection: keep-Alive
```

Como respuesta el servidor HTTP envía una fila de estatus, en la cual está un Header (esta vez con los parámetros del servidor). Separado por una fila vacía <CR LF CR LF> se entrega el elemento solicitado.

```
HTTP/1.1 200 OK                                  |   Fila de estado
Date: Thu, 15 Mar 2001  11:33:41 GMT             |
Server: Apache / 1.3.4 (Unix) PHP / 3.0.6        |
Last-Modified: Thu 15 Mar 2001 11:32:32 GMT      |
...                                              |
...                                              |   Cabecera
```

```
Keep-Alive: timeout = 15                    |
Connection: Keep-Alive                      |
Content-Type: text / html                   |
<html>                                      |
...                                         |   Página html
</html>                                     |
```

La fila de estado recoge las Versiones HTTP que soporta el servidor, un número de código de errores y un comentario. En la cabecera el servidor muestra las características de conexión soportadas y datos.

El comando POST

El comando opuesto a GET es el POST. Este permite al navegador entregar informaciones al servidor HTTP.

El uso clásico del comando POST es la entrega de datos desde un formulario en una página HTML. En el núcleo, es la estructura de la petición POST idéntica a la de GET. Después de los parámetros viene una fila vacía <CR LF CR LF>, a la que le siguen los datos a entregar. Contiene una petición POST más de una única información, entonces se separan unas de otras con un „&“. Como nombre de fichero (*filename*) se tiene que introducir en la primera fila de la petición POST un proceso existente en el servidor, que recoga las informaciones y las pueda procesar.

Para el formulario mostrado en el párrafo HTML, formulartest, podría tener la petición POST la siguiente forma; el parámetro hasta este momento no comentado *Referer* implementa aquí una referencia a la página formulario cargada originalmente.

```
POST /Formularauswertung.cgi HTTP/1.1
Accept: image/gif, image/jpeg
Referer: http://172.16.232.145/formulartest.html
Host: 172.16.232.145
Connection: Keep-Alive

EINGABEFELD1=test1&EINGABEFELD2=test2&submit=Abschicken
```

Sugerencia: La mayoría de los proveedores de internet ofrecen los llamados „CGI-Scripts“ (Programas en el servidor HTTP), que recogen los datos de formularios y los envían como e-mail a una dirección cualquiera. Así se puede proporcionar a sus clientes la oportunidad , directamente desde su página Web, de envío de un pedido o cuestión.

El Comando HEAD

Como tercer comando existe aunque también se le llama la variante de GET. El comando **HEAD** trabaja como el comando GET, pero el servidor HTTP devuelve sólo la fila de estado y la cabecera, pero no el elemento solicitado.

Se utiliza casi exclusivamente para fines de pruebas y por máquinas de búsqueda , que con el mensaje resultante (código de errores) pueden comprobar la existencia de una página.

Versiones de HTTP

HTTP se ha desarrollado desde la introducción de WWW varias veces y aparece en la actualidad en tres Versiones:

HTTP 0.9	en 1989 se presentó por 1ª vez y desde entonces se ha utilizado pero nunca especificado
HTTP 1.0	desde 1996 se especificó HTTP en la versión 1.0 por el RFC 1945 que es en su mayor parte idéntica a HTTP 0.9
HTTP 1.1	se introdujo en 1997 (RFC 2068) y está desde 1999 revisada y en uso (RFC2616).

Todos los navegadores actuales que se puedan conseguir soportan de forma estándar HTTP1.1, pueden también trabajar sin problemas con servidores que utilicen HTTP0.9 0 HTTP1.0.

El cambio fundamental en HTTP1.1 es que para la conexión establecida TCP de la transmisión del documento HTML se sigue utilizando para la posterior carga de otros elementos. HTTP 1.0 así como 0.9 habrían establecido una conexión TCP separada para cada elemento.

Una conexión persistente como en 1.1 eleva el rendimiento de los datos, ya que el tiempo para el establecimiento y desconexión desaparece.

Otra novedad en la versión 1.1 es que un servidor HTTP con una sola dirección IP puede procesar peticiones a diferentes direcciones Host. Por ejemplo: introduce el usuario en el navegador como *URL http://www.wut.de* entonces pregunta el PC por la dirección IP correspondiente al servidor DNS.

El navegador abre la conexión TCP y envía el comando GET. Para poder gestionar en un servidor HTTP las presencias de Internet de varios proveedores, se introdujo con Host un parámetro adicional en el comando GET, el cual entrega también al servidor el nombre del Host junto con la petición GET (pe. Host: *http:www.wut.de*). Gracias a este parámetro adicional puede reconocer el servidor HTTP con la petición GET para que Host es válida la conexión TCP.

Interactividad en WWW

Junto a la presentación puramente estática de informaciones (Páginas Web), hay también diferentes posibilidades, como provocar acciones desde el navegador y mostrar elementos dinámicos.

Para ello es en cualquier caso necesario un programa o un proceso el cual por ejemplo reciba datos del usuario y provoque las correspondientes reacciones.

Se diferencia entre los programas que están activos en el servidor WWW y aquellos que lo están en el navegador, es decir en el ordenador local. También se encuentran combinaciones de ambas muy a menudo.

Interactividad entre programas que están en funcionamiento en el Servidor

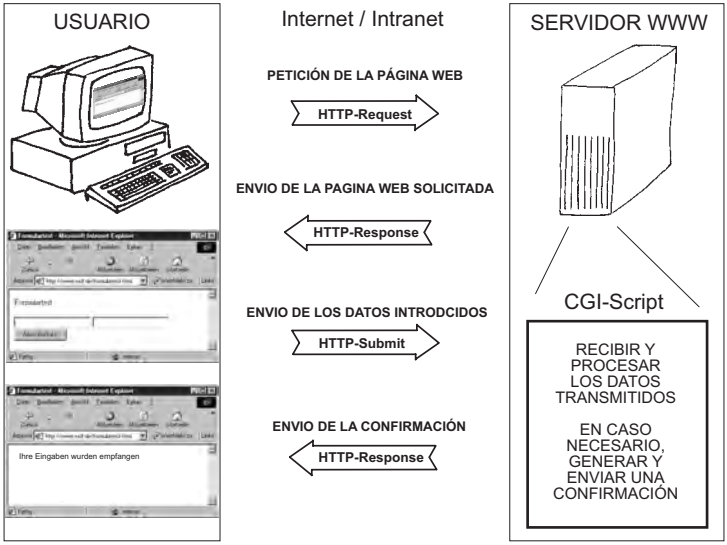
CGI Common Gateway Interface

El servicio de CGI-Scripts es en estos momentos el procedimiento más usado para mostrar en el navegador contenidos interactivos, así como provocar acciones.

Con CGI se pueden ejecutar programas en el servidor desde el navegador.

Con un hiperlink, un botón „Submit“ o datos directos del URL se llama al correspondiente programa y se entregan los parámetros necesarios.

Un ejemplo clásico son los formularios HTML, que son rellenados por el usuario. Pulsa el usuario el botón „submit“ (enviar) entonces se entregan los datos via http con ayuda del comando POST al servidor WWW. El CGI-Script introducido será iniciado y procesará los datos.



Otras posibles aplicaciones son contadores de visitantes, libros de visitas, foros de discusión, accesos a bases de datos o buscadores.

Los CGI-Scripts se pueden generar básicamente en todos los lenguajes de programación comunes. Lo que si es importante es que el servidor WWW soporte el lenguaje elegido.

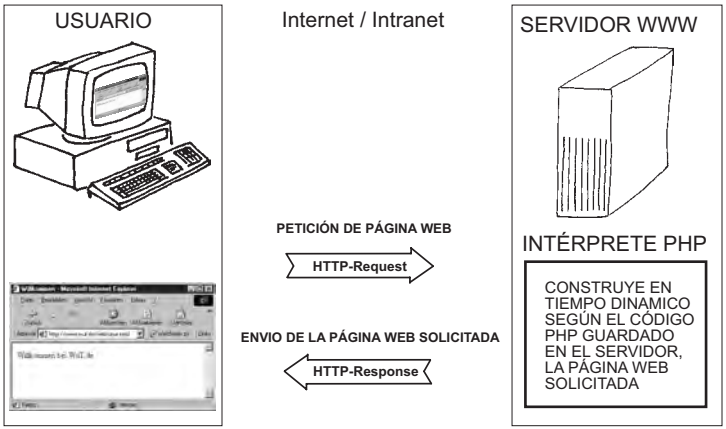
En la práctica se ha consolidado el uso de Perl para la creación de CGI-Scripts.

PHP

PHP también permite el funcionamiento de programas en un servidor WWW. PHP es un lenguaje intérprete, su código fuente está en formato texto, está unido a una página HTML y se almacena en el servidor WWW. Con esto se pueden definir contenidos estáticos de la página en formato HTML y al contrario que HTML se pueden introducir contenidos cambiantes con el código fuente de PHP. PHP puede acceder también a otras fuentes en el servidor, como pe. bases de datos.

Con la petición de la página correspondiente por el navegador, es analizado por el intérprete de PHP el código fuente integrado en la página que está en el servidor.

El intérprete de PHP genera individualmente una página en código HTML. La página Web resultante se envía entonces desde el servidor vía http al navegador.



De esta forma permanece el código fuente PHP invisible para el usuario.

Por ejemplo, para las compras online se podría integrar en una página Web vía PHP un contador dinámico de unidades disponibles almacenadas para los artículos ofrecidos, tiempos de entrega y precios para una aplicación de comercio de mercancía. Esto exige naturalmente, que en el servidor esté disponible un intérprete de PHP activo.

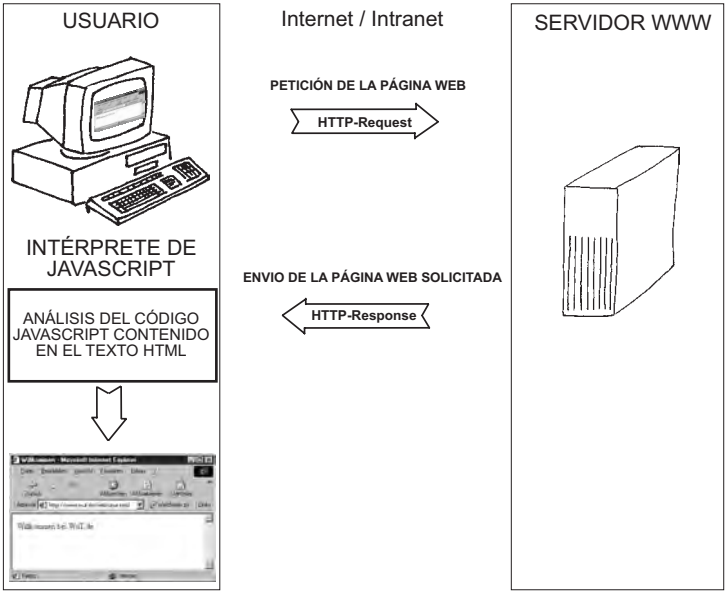
PHP se utiliza actualmente en las versiones PHP3 y PHP 4

Programas que se ejecutan en el navegador

JavaScript

Con JavaScript se integra el código fuente en el texto HTML de la página. El código JavaScript se indica con el rótulo <SCRIPT

language="JavaScript"> y al cargarse una página Web es reconocido, interpretado y ejecutado por el navegador.



Con JavaScript se pueden realizar pe. adaptaciones individuales del contenido mostrado de una página Web. También permite comprobar datos introducidos por el usuario antes de que se envíen al servidor www.

Un ejemplo:
El siguiente código analiza, si una página Web se ha solicitado por un dominio „com" o un dominio „de" y la muestra respectivamente en inglés o en alemán.

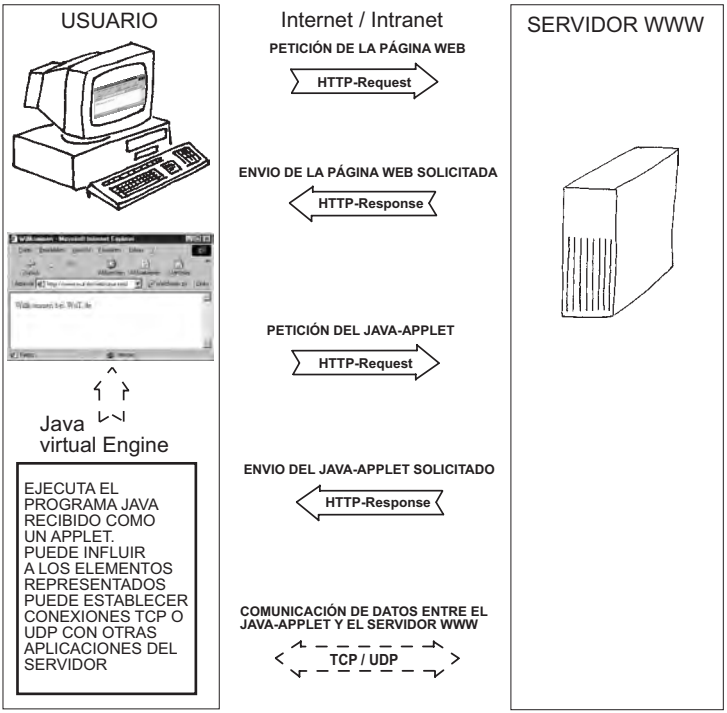
```
<HTML>
 <HEAD>
  <TITLE>urltest</TITLE>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
 </HEAD>
 <BODY>
  <SCRIPT LANGUAGE="JavaScript"><!—
  if (location.hostname == „www.web-io.com") document.write(„welcome at WuT");
  else document.write(„willkommen bei WuT");
```

```
  //--></SCRIPT>
 </BODY>
</HTML>
```

Java Applets

Aquí se trata de programas compilados, que se generaron con el lenguaje de programación Java. Los Java Applets, parecidos a elementos gráficos, se cargan adicionalmente al texto HTML y se ejecutan en el navegador.

Con los Java Applets se pueden realizar también acciones complejas, como accesos a la red en el nivel TCP y UDP.



Por razones de seguridad es sólo posible la comunicación con el servidor desde el cual se ha cargado el Applet. También en el ordenador local del usuario está limitado el acceso a elementos y funciones del navegador. Por ejemplo, un acceso al disco duro del ordenador propio no es posible.

Un ejemplo del uso de Java Applets es el termómetro Web de W&T. Con el termómetro Web se pone a disposición un Applet, que una vez arrancado en el navegador, pregunta la temperatura actual en intervalos periódicos y la muestra en la página web desde la cual se solicitó.

En el código HTML se integran los Applets con el rótulo Applet, donde se introduce con el parámetro „code=“ el nombre del Applet y con *codebase*= el Host desde el cual se carga el Applet.

```
<html>
 <head>
  <title>Schaltschranktemperatur</title>
 </head>
 <body bgcolor="#FFFFFF">
  <p><applet code="A.class" codebase = „http://172.16.232.152/" ></applet> </p>
 </body>
</html>
```

En el navegador aparece así:

E-Mail

La posibilidad de poder enviar correo electrónico en pocos segundos desde una punta del mundo a la otra, es seguramente una de las principales razones de la rapidísima extensión de Internet.

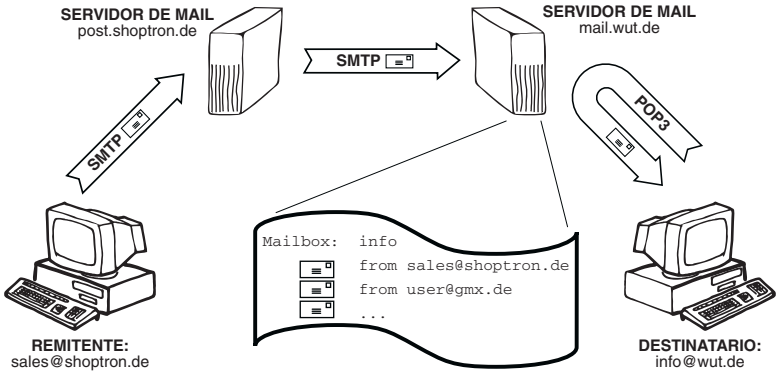
Al contrario que la mayoría de las otras aplicaciones en Internet, es el envío de e-mail un servicio en el cual no existe ninguna conexión directa entre emisor y receptor. Esto parece ser a priori confuso, pero tiene mucho sentido ya que sino fuera así, sólo sería posible el intercambio de e-mail cuando el remitente y el destinatario estuvieran al mismo tiempo activos en la red.

Para proporcionar una independencia temporal, necesita el receptor de e-mail un „Mailbox“, buzón, en un servidor Mail, en el cual los mensajes recibidos se puedan archivar.

Una dirección e-mail se compone siempre de un nombre del buzón y del dominio destino; como símbolo de separación está la „@“ entre estos dos componentes. Un ejemplo: *info@wut.de* define el buzón de Información en el servidor Mail de Wiesemann & Theis.

El camino de un e-mail desde el emisor al receptor se compone de dos partes, en las cuales el transporte está controlado por diferentes protocolos:

- Desde el ordenador del emisor hasta el buzón del destinatario se utiliza el protocolo SMTP.
- Desde el buzón del destinatario hasta el ordenador de él mismo se utiliza el protocolo POP3.



Estructura de un e-mail

Un e-mail se compone de la cabecera del mensaje y del mensaje en sí mismo. Esta cabecera se puede comparar con un sobre; contiene campos para el destinatario, remitente, fecha, asunto y unas cuantas informaciones más.

Aquí están los campos más importantes en un esquema:

Los cuatro campos siguientes forman una cabecera mínima y tienen que estar incluídos siempre:

Campo	Función
FROM	Dirección e-mail del autor
TO	Dirección e-mail del destinatario
DATE	Fecha y hora: Nota: La hora se puede introducir arbitrariamente y es normalmente la hora local del remitente.
SUBJECT	Texto del campo asunto
RECEIVED	El campo received presenta una especialidad ya que no se introduce cuando el e-mail es generado. Cada Mail-Router, que está en el recorrido del e-mail, introduce un campo received y deja de esta forma un sello de paso con fecha y hora.

El uso de los siguientes campos es opcional.

Campo	Función
SENDER	Dirección e-mail del remitente (normalmente siempre igual que los datos en FROM)
REPLY-TO	Dirección e-mail, a la cual el destinatario tiene que contestar en caso necesario.Importante, si los e-mails de un sistema empotrado (embedded) como el W&T IO-Mailer se envían automáticamente, como dirección de respuesta se puede colocar en este caso por ejemplo la dirección e-mail del administrador de la red.
CC	Dirección e-mail de un destinatario más que recibe una copia de la noticia (CC = Carbon Copy)
BCC	Dirección e-mail de un destinatario más pero que permanece invisible para todos los demás, (BCC = Blind Carbon Copy)
MESSAGE-ID	Identificación significativa de un e-mail que será concedido por el Software Mail de una forma arbitraria.
X-"MEINFELD"	Por medio de la colocación de "X-" se pueden generar algunos campos.

En algunos campos es posible una variante „RESENT" (reenviado), que aparece cuando se trata de un e-mail reenviado por el destinatario original.

La estructura formal de la cabecera de la noticia y los campos deben de cumplir las convenciones siguientes:

- Después del nombre del campo, viene un punto doble; después le sigue el correspondiente parámetro.
- Cada campo se sitúa en una única fila, que finaliza con <CR LFC> (Carriage Return Line Feed; hexadecimal 0D 0A).
- La cabecera y el cuerpo del mensaje se separan con una línea vacía extra <CR LF>.
- El cuerpo de la noticia sólo contiene el texto para transmitir así como los ficheros anexos. El final de la noticia se señala con <CR LF . CR LF> (hexadecimal 0D 0A 2E 0D 0A).
- Tanto como la cabecera como el cuerpo de la noticia se componen exclusivamente de símbolos ASCII 7-Bit. Por eso también se pueden transmitir todas las informaciones de control como puro texto.

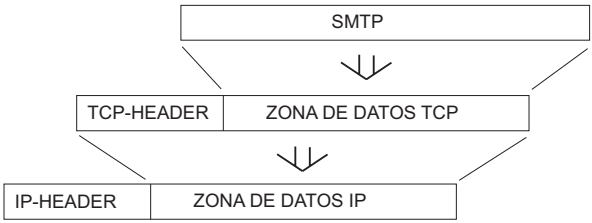
MIME Multipurpose Internet Mail Extensions

Para poder enviar también datos binarios (formato de 8-Bit) vía e-mail, se codifican los datos con el estándar MIME en el formato 7-Bit antes de la unión en el cuerpo del mensaje y en la recepción se vuelve otra vez a decodificar. Como el procesado de datos binarios lo soportan los programas actuales de e-mail automáticamente, renunciamos en este punto a una aclaración detallada del código MIME.

SMTP Simple Mail Transfer Protocol

El SMTP regula el envío de e-mails desde el cliente de mail al servidor de mail (SMTP-Server). El cliente de mail puede ser o el emisor original o un mail router situado en el camino. Los mail routers se utilizan cuando el e-mail en su recorrido tiene que pasar a través de varios dominios. A menudo se encuentra como mail router también la denominación MTA (Mail Transfer Agent).

Para cada tramo del recorrido que un e-mail deja atrás, se ha establecido una conexión TCP. El SMTP se basa sobre esta conexión TCP, donde se utiliza el puerto TCP 25.



El SMTP tiene a disposición algunos comandos de control (pe. login, datos del emisor, del receptor...). Cada comando SMTP se confirma uno por uno por el servidor SMTP. El propio e-mail se envía completamente con cabecera y cuerpo, y es entonces cuando el servidor SMTP lo confirma. Si no hay más e-mails para enviar, entonces también se desconecta la conexión TCP.

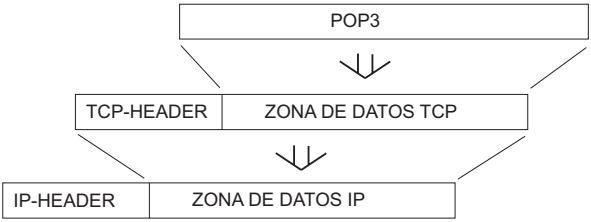
Si el e-mail ha alcanzado el servidor mail destino, se deja en el buzón del destinatario y permanece allí hasta que el destinatario lo recoja.

POP3 Post Office Protocol Version 3

Para recoger los e-mails recibidos del buzón en el servidor de mail, se usa en la mayoría de los casos el protocolo POP3. El destinatario no es informado de los e-mails entrantes. El mismo debe comprobar su buzón sobre los e-mails recibidos y puede recogerlo cuando quiera.

La mayoría de los programas de e-mail usados en la actualidad comprueban en el comienzo automáticamente el buzón del usuario para ver los e-mails recibidos. Muchos programas e-mail ofrecen la posibilidad de introducir un intervalo en el cual el buzón será comprobado cíclicamente. Usuarios típicos, que la mayor parte del tiempo del día están „offline“ (sin conexión), reciben sus e-mails sólo cuando se conectan telefónicamente al proveedor. Pero para los ordenadores que tienen permanente acceso a Internet es cuando tiene verdadero sentido la petición cíclica: el usuario está siempre online y recibe sus e-mails con muy poco retraso, casi en tiempo real.

También el protocolo POP3 se basa sobre una conexión TCP y no es más que un diálogo de texto.



POP3 utiliza el puerto TCP 110. Al igual que SMTP, aquí comienza el diálogo también con un login. Aunque en POP3 se tiene que registrar el destinatario en dos pasos: con nombre de usuario y con contraseña.

Después del login con éxito, proporciona POP3 algunos comandos, con los cuales se pueden mostrar en una lista los e-mails recibidos, recogerlos o borrarlos.

En la actualidad el usuario se enfrenta pocas veces con SMTP y POP3: él tiene que introducir sólo en la configuración del software de mail el nombre del servidor POP3 y SMTP, el desarrollo de los protocolos mismos se encarga invisiblemente en el transfondo por los programas de e-mail.

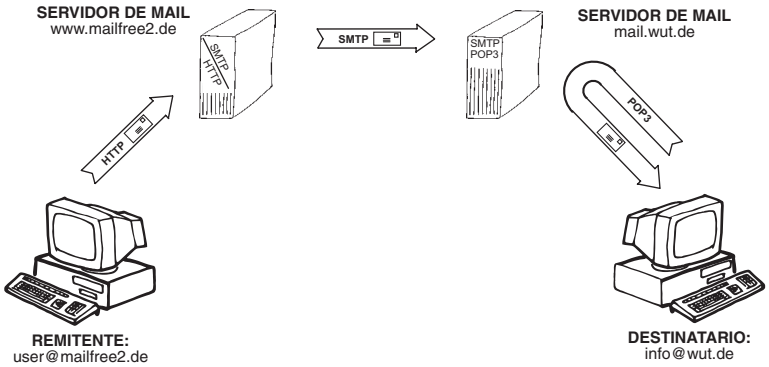
Para comentar también todo completamente, sea mencionado que junto al protocolo POP3 también existen los protocolos POP2 y POP1 (ambos predecesores de POP3) e IMAP4, que se desarrollaron también para recoger los e-mails. Estos protocolos no se impusieron en la práctica o fueron desplazados por POP3.

Enviar y recibir e-mail por HTTP

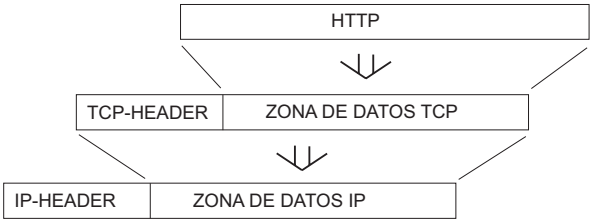
Con el creciente uso del e-mail, surgen cada vez más proveedores de mail gratis, que ponen a libre disposición en sus servidores de mail, buzones gratis. Estos servicios, que cualquiera puede utilizar, se financian normalmente con publicidad.

Para disponer de espacio donde introducir la publicidad, ofrecen la mayoría de los proveedores de mail gratis al usuario la posibilidad de enviar y recibir cómodamente e-mails por HTTP con el navegador, el cual naturalmente está enriquecido de anuncios en banners. Para todo esto están a disposición del usuario los correspondientes formularios HTML.

Para posibilitar la gestión de e-mails con HTTP, el proveedor de mail gratis tiene que tener funcionando una combinación especial de servidor e-mail, el cual para la parte del usuario trabaja como servidor Web y para la otra parte como servidor SMTP. El camino de un e-mail es de la siguiente forma:



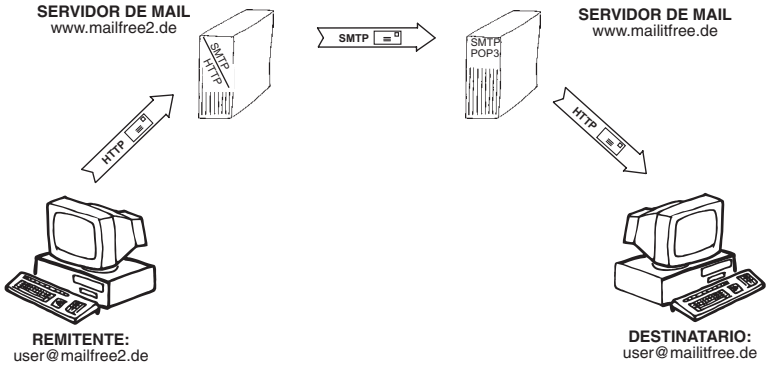
Entre el ordenador del remitente y el servidor del proveedor de mail gratis, se utiliza el protocolo HTTP. Como en otras aplicaciones HTTP, también aquí se utiliza el puerto TCP número 80.



Entre los servidores de e-mail mismos no se cambia nada. Se comunican entre ellos con el protocolo SMTP.

Entre el servidor e-mail destino y el ordenador del destinatario pueden producirse dos variantes diferentes:

- Si el destinatario tiene una cuenta estándar de e-mail, se recogerán los mails entrantes por POP3.
- Si el destinatario usa también los servicios de un proveedor de mail gratis, se pone en servicio otra vez HTTP.



Quien quiera enviar sus e-mails por SMTP y POP3, tiene que asegurarse en la elección del proveedor de mail gratis que también exista a su disposición acceso sobre un servidor SMTP así como POP3.

E-mails y DNS

También al enviar e-mails se trabaja en el nivel IP con direcciones IP. La resolución de nombres con direcciones e-mail funciona también en principio exactamente igual que con los usuarios de la red. Naturalmente no se resuelve la dirección del destinatario de los e-mails, sino la del servidor e-mail, en el cual el destinatario tiene su buzón de correo electrónico.

Recordatorio: Para resolver nombres en direcciones, se sirve la pila TCP/IP (Stack) de un programa de resolución, que al servidor de DNS le pregunta la correspondiente petición.

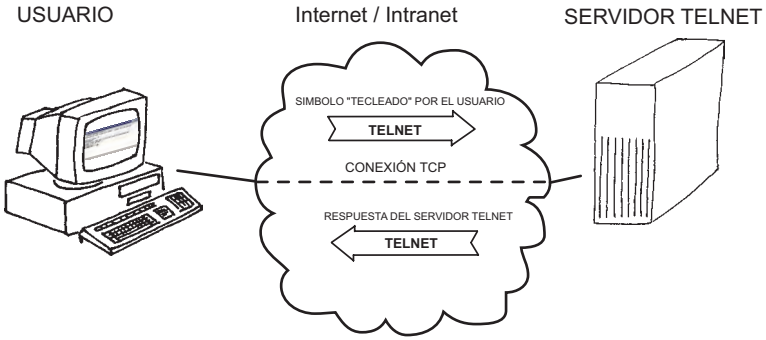
Pero el nombre del Host del servidor destino no es conocido. Conocido es sólo el dominio destino, que está presente en la dirección de e-mail detrás del símbolo @. Para poder resolver peticiones DNS acerca de servidores mail, existen en los servidores DNS secuencias especiales de datos, en las que se señala un servidor e-mail junto su dominio y su correspondiente dirección IP.

El programa Resolver (de resolución DNS) introduce en la petición sólo el nombre de dominio destino y comunica con ello que se trata de un servidor de mail para el usuario buscado. El servidor DNS responde con la dirección IP buscada y la devuelve al programa resolver.

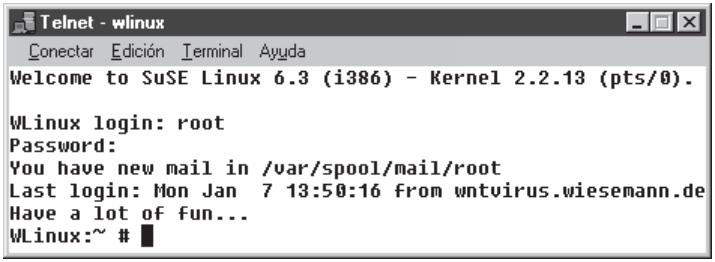
El nombre del buzón no es para nada necesario en la petición DNS. Será procesado por primera vez en la recepción del mensaje en el servidor de mail destino, para que el mensaje se deposite en el buzón de correo correcto.

Telnet Terminal over Network

Expresado sencillamente, es Telnet una ventana de texto así como un programa orientado al texto, con el cual otro ordenador (Host) en la red puede ser controlado a distancia por un usuario.



Una sesión Telnet se puede imaginar como una ventana de DOS en la que no obstante se ejecutan las ordenes introducidas en el ordenador remoto.



Para ello se necesitan varios elementos.

El cliente Telnet

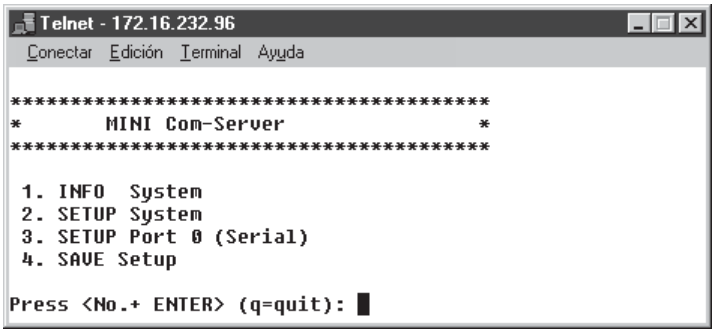
Todos los sistemas operativos modernos disponen hoy en día de un programa cliente de Telnet.

El cliente Telnet construye una conexión TCP con un servidor de Telnet, se hace cargo de las entradas del teclado del

usuario, las entrega al servidor de Telnet y muestra los símbolos enviados por el servidor en el monitor.

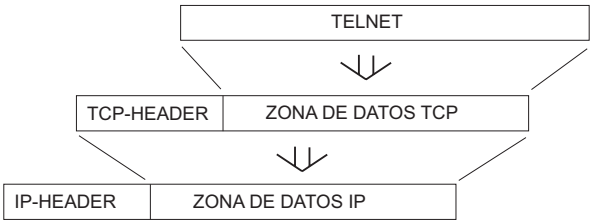
El servidor Telnet

está activo en el ordenador remoto y ofrece la posibilidad a uno o varios usuarios de entrar allí. Con esto es el servidor Telnet (en sistemas UNIX también a menudo llamado Telnet-Deamon) el vínculo de unión entre el acceso a la red vía cliente Telnet y el proceso de servicio. En su origen se utilizó Telnet para conseguir un acceso remoto en sistemas UNIX. En la actualidad muchos sistemas empotrados como Com-Server, Printer-Server, Switches, Hubs y Routers disponen también de un servidor Telnet, que sirve como acceso para su configuración.



El protocolo Telnet

También Telnet se basa sobre TCP como protocolo base.



Aquí se utiliza el puerto nº. 23, si el usuario no ha fijado otro puerto para ello. Se puede también fijar cualquier puerto a elección. Lo importante es que en el puerto elegido haya un servidor Telnet activo.

El protocolo Telnet asume fundamentalmente tres tareas:

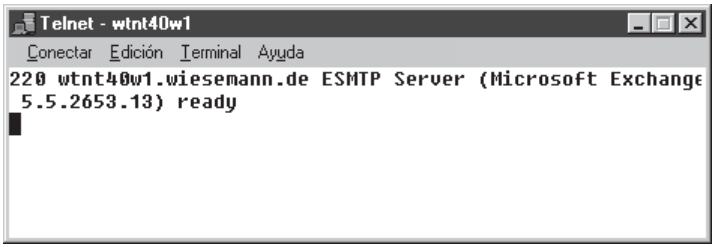
1. Establecimiento del uso de códigos de control y combinaciones de símbolos para el posicionamiento del cursor.
   Como base conjunta para cliente y servidor se usa aquí el estándar NVT (Network Virtual Terminal). El NVT utiliza el código de símbolos 7 Bit ASCII y fija que símbolos se tienen que mostrar y cuales son para el control y posicionamiento.

2. Negociación y configuración de las opciones de conexión. Telnet puede hacer uso con los establecimientos en NTV de una variedad de funciones especiales. El protocolo Telnet ofrece al cliente y servidor la posibilidad de negociar las opciones para la conexión. Por ejemplo: si el servidor tiene que devolver todos los símbolos recibidos desde el cliente como un eco.
   Para ello se utilizan símbolos de control, donde el octavo bit entre en juego, por lo tanto, símbolos por encima del 127 y con ello fuera del conjunto de símbolos NTV.

3. El transporte de los símbolos, que son intercambiados entre cliente y servidor. Todos los símbolos del conjunto NTV introducidos por el usuario o enviados por el servidor se empaquetan 1:1 en el espacio de datos de un paquete TCP y se transportan por la red.
   La sencillez del protocolo Telnet, así como la transparencia en el transporte de los símbolos, han convertido también a Telnet en una herramienta de diagnóstico muy popular. Así se permiten establecer conexiones a servidores HTTP, SMTP o POP3.

Por ejemplo se permite comprobar con la introducción del comando siguiente en una ventana DOS, si el servidor SMTP(Puerto 25) está trabajando:

Si el servidor SMTP está activo, se devolverá una señal de bienvenida.



Se podrían enviar e-mails en teoría por Telnet mediante la introducción consecuente de datos del protocolo SMTP.

También otros protocolos sencillos como HTTP o POP3 permiten entenderse vía cliente Telnet.

FTP File Transfer Protocol

Expresado en palabras sencillas, FTP permite a un usuario el acceso en la red al sistema de ficheros, así como al disco duro de un ordenador distante.

El cliente FTP

FTP trabaja según el principio cliente /servidor. Un cliente FTP es hoy en día un componente de cada sistema operativo. Bajo Windows pe. se arranca el cliente FTP con la introducción del comando c:\ftp en una ventana DOS.

Con el comando OPEN, seguido de la dirección IP o del nombre de Host del servidor FTP, se abre la conexión FTP y el usuario tiene que introducir su nombre de login y su clave.

Después de un login con éxito, son posibles dependiendo de los derechos de acceso, entre algunas, las siguientes operaciones con ficheros:

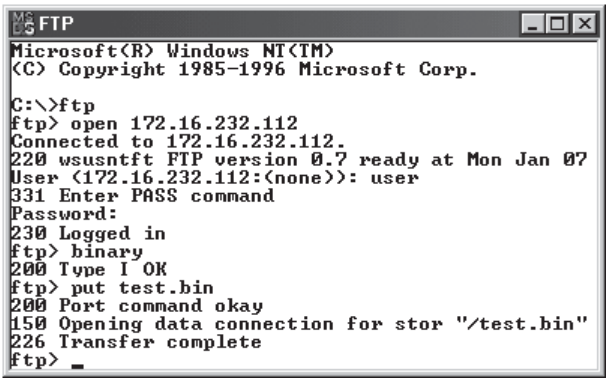
	Comando FTP
Guardar ficheros en el servidor	PUT
Descargar ficheros desde el servidor	GET
Adjuntar datos a un fichero existente	APPEND
Borrar ficheros en el servidor	DELETE
Mostrar el contenido del índice	DIR

Una lista de todos los comandos soportados se consigue con la introducción de una „?“ detrás del símbolo de sistema FTP (FTP Prompt). Una pequeña descripción de un único comando se puede llamar con „? comando“.

Una característica importante de FTP es el diferente manejo de ficheros de texto y binarios. Para elegir el modo deseado, pone FTP a disposición 2 comandos más.

	Comando FTP
Para la transmisión de ficheros de texto	ASCII
Para la transmisión de ficheros binarios	BINARY

Después de la introducción de FTP, el servicio tiene lugar de una forma como si fuera un diálogo. Como se muestra en el ejemplo donde se guarda el fichero „test.bin“ en el servidor „172.16.232.23“:



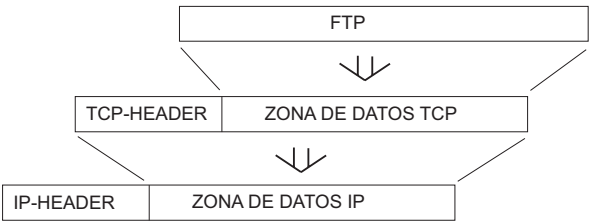
Dependiendo del sistema operativo pueden variar tanto la pantalla como los comandos del cliente FTP.

En sistemas operativos UNIX además hay que tener en cuenta a la escritura minúscula y mayúscula.

Un manejo más confortable de FTP se puede alcanzar con el uso de programas cliente FTP a la venta que presentan pantallas de usuario gráficas.

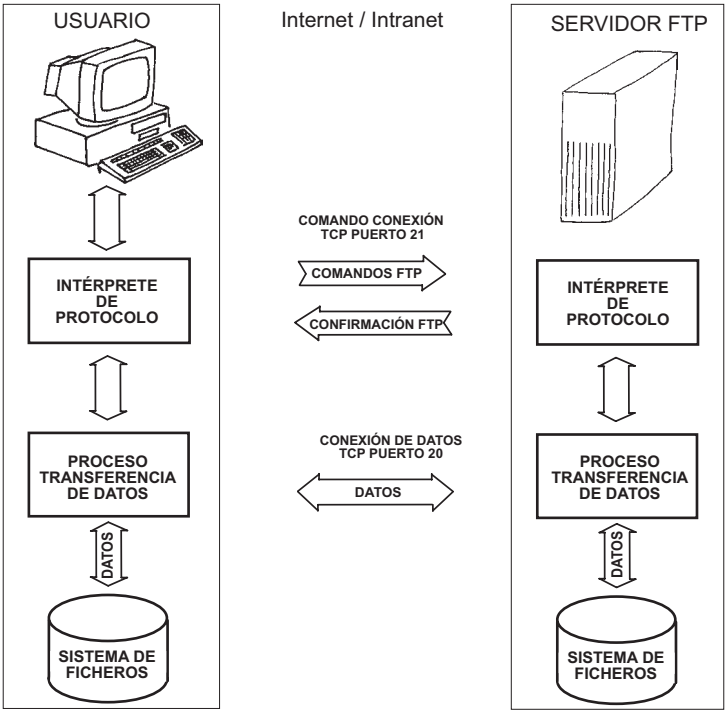
El protocolo FTP

FTP se basa sobre el protocolo básico, orientado a la conexión y seguro, TCP.



Al contrario que otros servicios internos, FTP utiliza dos conexiones TCP y con ello dos puertos TCP.

- Puerto 21 como conexión de comandos.
- Puerto 20 para la transmisión de ficheros.



El control de la transferencia de ficheros entre cliente y servidor se maneja sobre un diálogo de comandos. Esta parte se desarrolla por los intérpretes de comandos a través de la conexión de comando.

La conexión de comandos se mantiene abierta durante la totalidad de la sesión FTP.

La transferencia real del fichero tiene lugar a través de la conexión de datos, que se abre nuevamente por el proceso de transferencia de datos para cada operación de ficheros.

El proceso de transferencia de datos es el nexo entre la red y el sistema de ficheros y es controlado por el intérprete de protocolo.

El servidor FTP

Un servidor FTP está disponible normalmente sólo para sistemas operativos de servidores y tiene que ser iniciado.

Los servidores FTP ofrecen dos posibilidades de acceso:

1. Sólo los usuarios dados de alta tienen acceso y pueden, dependiendo de los derechos de acceso almacenados en una lista de usuarios, ejecutar operaciones de ficheros.
2. Cada usuario puede acceder al servidor. Un login no tiene lugar o se da el nombre de usuario „anonymus“. Entonces se habla de FTP anónimo.

Uno de los usos principales de FTP es hoy en día el almacenamiento de páginas HTML en servidores WWW, que para este fin siempre tienen un acceso FTP.

FTP puede también ser usado para guardar datos serie de terminales en un fichero del servidor mediante clientes FTP empotrados como por ejemplo el Com-Server W&T.

TFTP Trivial File Transfer Protocol

Junto a FTP es también TFTP un servicio más para poder acceder por la red a ficheros de un ordenador remoto.

TFTP es no obstante tanto por la diversidad de las funciones como por el tamaño del código de programa, claramente „más delgado“ que FTP.

Un cliente TFTP no es siempre una parte del sistema operativo y por ejemplo en el entorno de Windows sólo Windows NT y 2000 lo incluyen.

Los servidores TFTP están rara vez en funcionamiento con entornos ofimáticos.

Especialmente optimizado es TFTP para su uso en sistemas embebidos o empotrados, en los cuales sólo hay disponible un espacio limitado de memoria para componentes de sistemas operativos. TFTP ofrece aquí por su mínimo código de programa una gran medida en eficiencia.

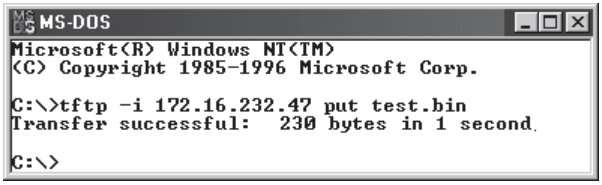
En los Com-Servers, servidores de impresión y mini terminales, se usa para transmitir ficheros de configuración y Firmware.

TFTP ofrece sólo dos operaciones de ficheros:

	Comando TFTP
Almacenar ficheros en el servidor	PUT
Descargar ficheros desde el servidor	GET

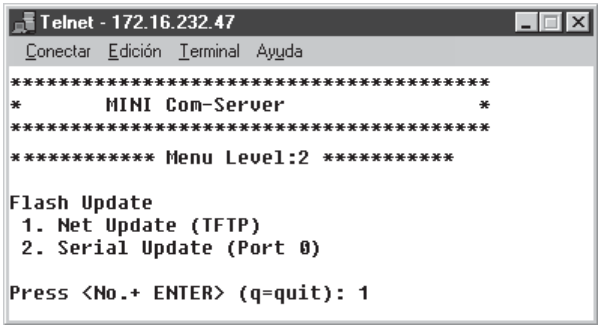
Como FTP, TFTP diferencia entre la transmisión de ficheros de texto y ficheros binarios. Si se tienen que transmitir ficheros binarios, se introduce esto a través del parámetro adicional „-i“.

Un pequeño ejemplo: El fichero binario „test.txt“ se almacena por un ordenador Windows NT en el servidor wlinux.



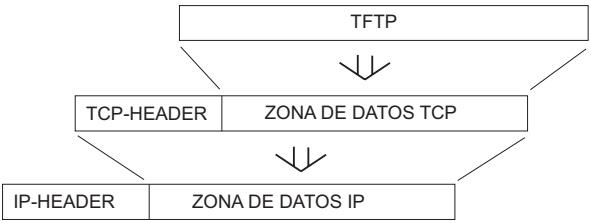
Se renuncia a una autentificación, es decir un login con clave como en FTP.

Una posibilidad, para evitar accesos indeseados, lo queremos mostrar con un ejemplo con el Com-Server. Para guardar en un Com-Server vía TFTP una nueva versión de Firmware, se tiene que habilitar el acceso TFTP a través de Telnet.



Después se comprueba, si los datos recibidos verdaderamente son el Firmware de un Com-Server.

Al contrario que FTP, TFTP utiliza como protocolo base UDP, que a su vez utiliza el puerto 69

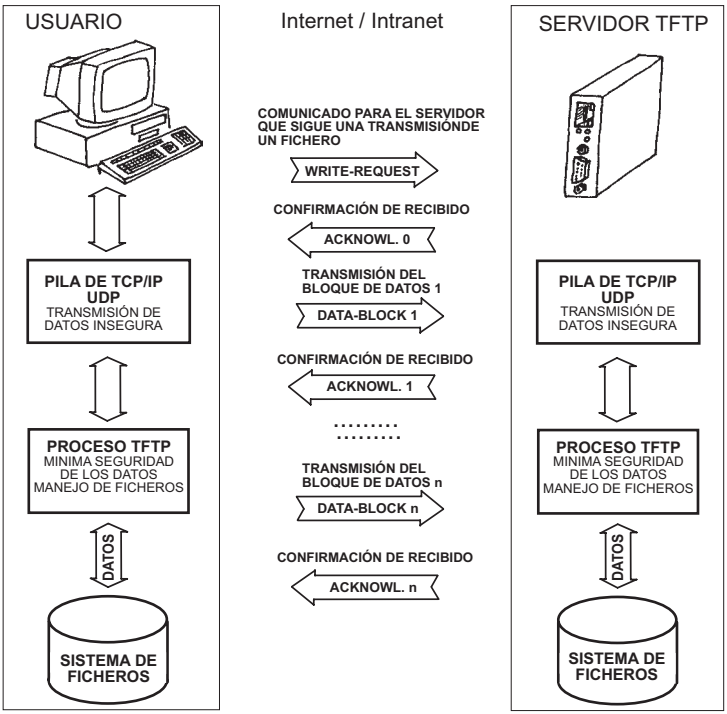


Como recordatorio:

UDP trabaja sin conexión. Se habla también de los paquetes UDP como Datagramas, ya que cada paquete se trata como un envío de datos individual. En el nivel UDP no se acusan de recibo los paquetes recibidos. El emisor no recibe un mensaje de vuelta si un paquete enviado verdaderamente ha llegado al receptor. Paquetes UDP no reciben una secuencia numerada. Un receptor, que recibe más paquetes UDP, no tiene la posibilidad de decidir si los paquetes se han recibido en la secuencia correcta.

Por estas razones TFTP asume la seguridad de la transmisión de datos el mismo.

La transmisión de ficheros tiene lugar en bloques de 512 Bytes cada uno, donde los bloques son señalados con un número correlativo. Cada bloque recibido se confirma por el otro extremo. Después de la recepción de la última confirmación se envía el siguiente bloque.



TFTP reconoce , si los bloques de datos recibidos son correctos aunque una corrección de fallos no existe. Si en la transmisión ocurriera algún fallo, como el tamaño del paquete no es exacto o un paquete se pierde por completo, la transmisión se cancelará. En este caso puede el usuario o una Aplicación Software inteligente comenzar de nuevo el procedimiento.

SNMP Simple Network Management Protocol

En el momento de la impresión de esta edición, este capítulo no estaba, lamentablemente, todavía terminado.

Ampliaciones a este libro lo pueden encontrar en Internet en todas las hojas de datos de los Com-Server en formato pdf.

Aquí tienen también siempre una versión actual para descargarse.

Visítennos en la página http://www.wut.de

Modbus TCP

En el momento de la impresión de esta edición, este capítulo no estaba, lamentablemente, todavía terminado.

Ampliaciones a este libro lo pueden encontrar en Internet en todas las hojas de datos de los Com-Server en formato pdf.

Aquí tienen también siempre una versión actual para descargarse.

Visítennos en la página http://www.wut.de

Programación Socket

Los protocolos y servicios estándares de Internet que se han mostrado en los capítulos anteriores, ofrecen ya posibles soluciones para aplicaciones diversas.

Pero a menudo son también necesarias soluciones especiales de software a medida para una aplicación determinada. Esto puede ser tanto de igual manera a especiales interfaces gráficos de usuario o de introducción de datos a nivel usuario como también conexión técnica a terminales y programas existentes.

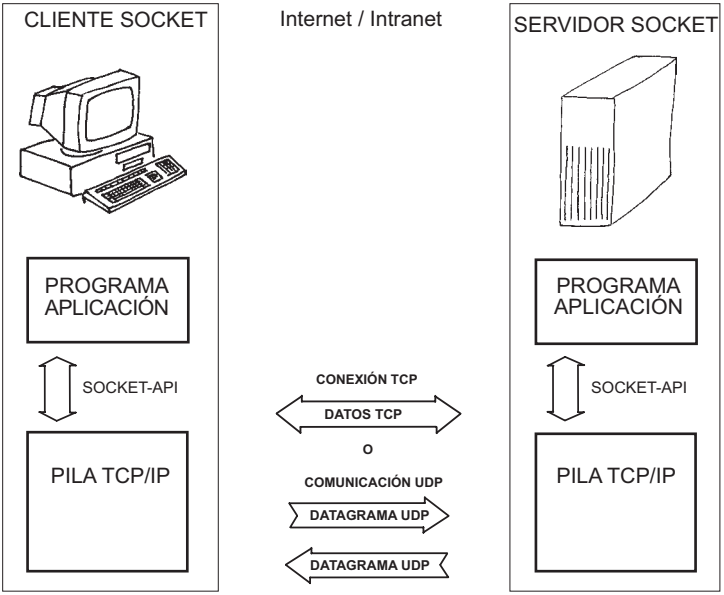
Como recordatorio: La parte de un Sistema operativo que es responsable de la comunicación TCP/IP, se denomina la pila TCP/IP (TCP/IP Stack). El driver o controlador de TCP/IP intercambia con la aplicación las direcciones IP y puertos, así como la entrega de los datos y compone con ello paquetes IP. Este paquete IP se entrega de la pila TCP/IP al nivel físico para su envío a la tarjeta de red.

El orden exacto de una conexión surge de la conjunción dirección IP y número de puerto. Se habla de esta correspondencia como un Socket.

Quien quiera desarrollar una aplicación propia que soporte una comunicación vía TCP/IP, tiene con la programación del Socket posibilidades casi ilimitadas.

Todos los sistemas operativos modernos poseen actualmente de un interfaz de aplicaciones Socket.

El Socket API es una interfaz software definida, que permite dependiendo del lenguaje de programación y sistema operativo el acceso a la pila TCP/IP sobre ficheros DLL y controles.



Los lenguajes de alto nivel, Visual Basic y Delphi ofrecen una plataforma especialmente fácil para la creación de aplicaciones individuales, que presentaremos aquí con algunos pequeños ejemplos.

Naturalmente ofrecen también lenguajes de programación como C++ y Java excelentes requisitos para la programación Socket. Ejemplos y aclaraciones se pueden encontrar en http://www.wut.de

¿Cliente TCP, Servidor TCP o par UDP?

Independientemente del entorno de desarrollo elegido, se debería elegir dependiendo de la tarea y los componentes implicados que parte de la comunicación debe tomar el programa desarrollado.

TCP

Aplicaciones, en las que se intercambiarán grandes cantidades de datos, deben de ser desarrolladas con base en TCP. TCP

tiene aquí la ventaja de una conexión fija, donde la pila TCP/IP se encarga de la seguridad de los datos.

Cliente TCP

Si tiene que decidir el propio programa cuando y con quién se tiene que realizar la conexión, entonces es más práctico programar una aplicación cliente.

La pila TCP/IP necesita del programa aplicación la dirección IP así como el nombre del servidor y el nº. de puerto, en el cual la conexión se debe de realizar.

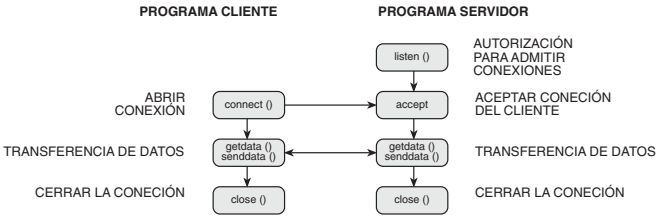
El puerto local en el cual la aplicación cliente recibe los datos del servidor, se asigna arbitrariamente en la aplicación cliente de la pila TCP/IP.

Servidor TCP

Cuando la propia aplicación tiene que poner a disposición datos y servicios a uno o a muchas otras aplicaciones, entonces se programa un servidor.

La pila TCP/IP necesita del programa aplicación la siguiente información, a qué puerto local se desea realizar la conexión de una aplicación cliente.

Si existe alguna conexión deseada, la pila informa al programa. Si la aplicación acepta la conexión, se pueden enviar y recibir los datos.



UDP

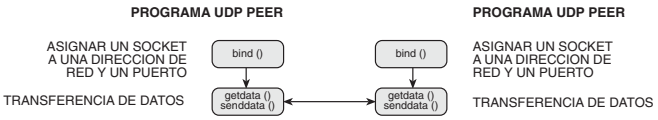
Para aplicaciones de red con participantes cambiantes o envíos de datos cortos, es UDP preferente como protocolo sin conexiones.

En el nivel UDP, tienen ambos participantes de la comunicación los mismos privilegios. No se diferencia entre cliente y servidor. Ya que no se pierde tiempo para la conexión y desconexión, se pueden conseguir en pequeñas cantidades de datos tiempos de acceso más rápidos.

Se tiene que tener en cuenta, que la seguridad de los datos se tiene que realizar en la propia aplicación.

La pila TCP/IP necesita del programa aplicación la dirección IP o el nombre del Host y el nº. de puerto propio.

Por la correspondencia de estos parámetros se genera un Socket, por el cual se pueden enviar y recibir los datos.



Programación Socket en Visual Basic

Los ejemplos mostrados se crearon en Visual Basic 5, que disfruta todavía después del lanzamiento de VB6 en este momento de mayor extensión.

Todos los que tengan unos conocimientos básicos sobre programación en VB, deberían de seguir fácilmente los ejemplos de programación.

Para poder crear en VB programas basados en TCP/IP, se tiene que instalar en la lista de componentes el control Winsock.

- Con el botón derecho del ratón pulsar la barra de componentes.
- Elegir con el botón izquierdo del ratón el punto del menú „componentes“
- En la lista de los elementos de control elegir „Microsoft Winsock Control“

La barra de componentes está ahora enriquecida con el elemento de control Winsock. 

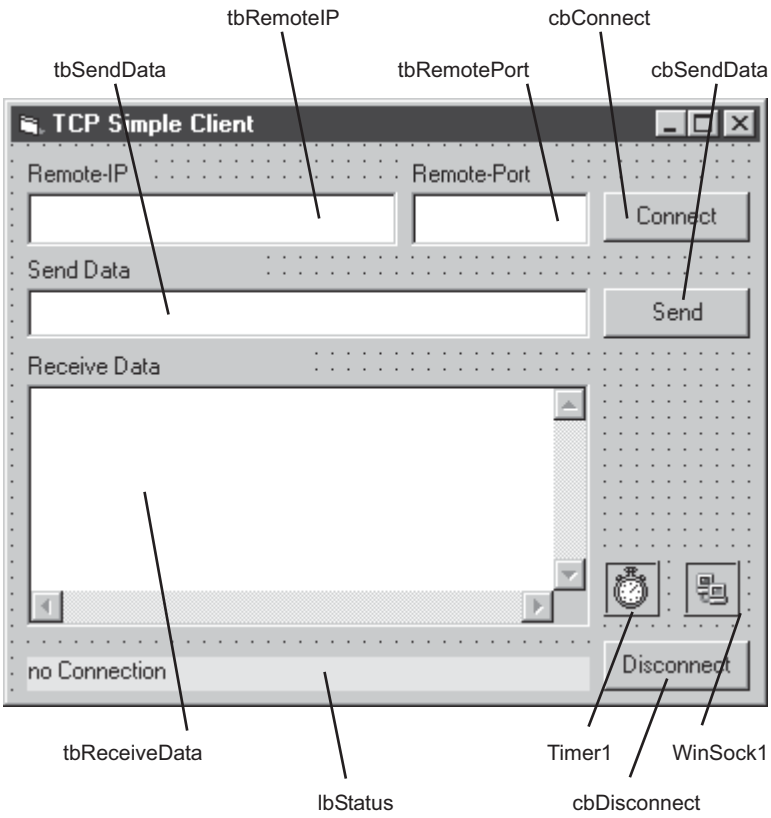
Un cliente TCP en VB

Como principio queremos generar un cliente TCP, que soporte las siguientes tareas:

(El ejemplo completo se puede descargar desde http://www.wut.de)

- Establecimiento de la conexión TCP.
- Envío y recepción de datos de texto.
- Cierre de la conexión TCP.
- Mostrar el estado de la conexión TCP.
- Reconocimiento de fallos.

Para ello se construye un formulario con los siguientes elementos:



Todas las variables y nombres de los elementos deben describirse a sí mismos por el nombre elegido.

Para los elementos del tipo Textbox se han elegido nombres con el comienzo „tb“, los „Command-Buttons“ con el comienzo „cb“.

El código VB siguiente se muestra por lo tanto sin muchos comentarios:

El procedimiento siguiente realiza y construye la conexión TCP después de apretar el botón Connect con los parámetros de direccionamiento que el usuario ha introducido y con ayuda del método Connect del

Winsock-Control.

```
PrivateSubcbConnect_Click()
  If (tbRemotePort.Text <> "") And (tbRemoteIP.Text <> "") Then
    Winsock1.RemotePort = tbRemotePort.Text
    Winsock1.RemoteHost = tbRemoteIP.Text
    Winsock1.Connect
  End If
End Sub
```

Procedimiento para la separación de la conexión TCP con ayuda del método Close.

```
Private Sub cbDisconnect_Click()
  Winsock1.Close
  cbConnect.Enabled = True
  cbConnect.SetFocus
End Sub
```

Mediante el botón Send se envía el texto introducido por el usuario a través de la conexión TCP existente. Aquí se utiliza el método Senddata.

```
Private Sub cbSendData_Click()
  Winsock1.SendData (tbSendData.Text)
  tbSendData.Text = ""
End Sub
```

La rutina Timer observa el estado actual de la conexión sobre las características de estado del Control de Winsock. Un intervalo razonable para el timer es 500ms.

```
Private Sub Timer1_Timer()
  Select Case Winsock1.State
    Case ckClosed
      lbStatus.Caption = "no Connection"
    Case sckResolvingHost
      lbStatus.Caption = "waiting for DNS"
    Case sckHostResolved
      lbStatus.Caption = "get IP from DNS"
    Case sckConnecting
      lbStatus.Caption = "connecting"
    Case sckConnected
      lbStatus.Caption = "Connection to " + Winsock1.RemoteHost
```

```
    Case sckClosing
      lbStatus.Caption = "closing Connection"
    Case sckError
      lbStatus.Caption = "Connection Error"
      Winsock1.Close
    End Select
    If Winsock1.State <> sckConnected Then
      cbSendData.Enabled = False
      cbDisconnect.Enabled = False
      cbConnect.Enabled = True
    Else
      cbSendData.Enabled = True
      cbDisconnect.Enabled = True
      cbConnect.Enabled = False
    End If
End Sub
```

Este procedimiento se llamará automáticamente cuando una conexión desde el otro extremo se finalice y puede utilizarse por ejemplo con el método close para colocar la propia administración de la conexión.

```
Private Sub Winsock1_Close()

  Winsock1.Close
  cbConnect.Enabled = True
  cbDisconnect.Enabled = False
  cbSendData.Enabled = False
  lbStatus.Caption = "no Connection"
  cbConnect.SetFocus
End Sub
```

Este procedimiento se llamará automáticamente cuando el establecimiento de la conexión sea con éxito.

```
Private Sub Winsock1_Connect()

  cbDisconnect.Enabled = True
  cbConnect.Enabled = False
  tbReceiveData.Text = ""
  lbStatus.Caption = "connected to " + Winsock1.RemoteHost
End Sub
```

Este procedimiento funcionará automático al recibir datos.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
  Dim ReceiveData As String
  Winsock1.GetData ReceiveData
  tbReceiveData.Text = tbReceiveData.Text + ReceiveData
End Sub
```

Datos entrantes se cogerán con el método Getdata y se mostrarán en la ventana „Receive Data“.

```
Private Sub Winsock1_Error(ByVal Number As Integer,_
Description As String, ByVal Scode As Long,_
ByVal Source As String, ByVal HelpFile As String,_
ByVal HelpContext As Long, CancelDisplay As Boolean)
  Winsock1.Close
  dummy = MsgBox("Connection Error", vbOKOnly, "TCP simple Client")
End Sub
```

Así se tiene listo un programa con menos de 2 páginas de código fuente con un cliente TCP, señalización de estado y manejo de fallos.

Como parte complementaria, se necesita por supuesto un servidor, que recoja los deseos de conexión del cliente. Esto puede ser un equipo existente como un W&T Com-Server o una aplicación VB-Server más.

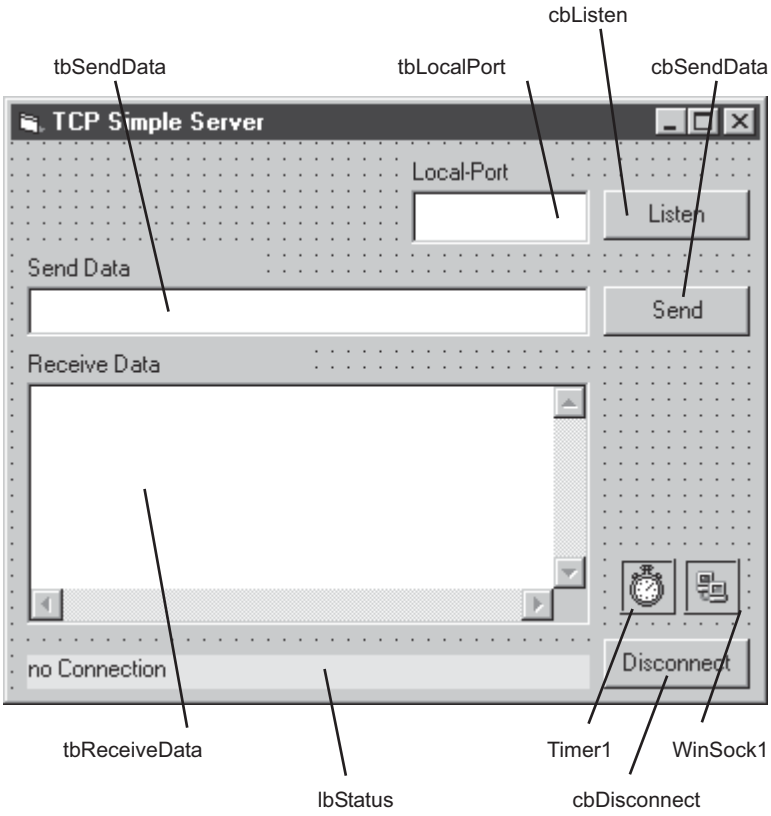
Cómo se puede construir un servidor TCP, se mostrará en el siguiente ejemplo.

Un servidor TCP en VB

La aplicación servidor se encarga de las siguientes tareas:

- „Escuchar“ en la red por si existe una petición de conexión en los puertos soportados.
- Hacerse cargo de las conexiones deseadas.
- Envío y recepción de datos de texto.
- Mostrar el estado de la conexión.
- Mostrar los fallos en la conexión.
- Cerrar la conexión desde la parte del servidor.

Para ello se desarrolla un formulario con los siguientes elementos:



Todas las variables y nombres de los elementos deben describirse a sí mismos por el nombre elegido.

Los procedimientos de VB siguientes serán necesarios para el servidor:

Este procedimiento no es estrictamente necesario para una aplicación de servidor y está presente sólo al principio del código fuente porque VB muestra los procedimientos ordenados alfabéticamente. Permite que con el método Close se cierre una conexión existente. Con el método Listen comienza el control Winsock de nuevo a escuchar posibles peticiones de conexión en el puerto elegido.

```
Private Sub cbDisconnect_Click()
  Winsock1.Close
  Winsock1.Listen
End Sub
```

Con la llamada del método listen, comienza el control Winsock a escuchar posibles peticiones de conexión en el puerto elegido.

```
Private Sub cbListen_Click()

If tbLocalPort.Text <> "" Then
    Winsock1.LocalPort = tbLocalPort.Text
    Winsock1.Listen
    cbListen.Enabled = False
  End If
End Sub
```

Al pulsar el botón de „Send“ se envía el texto introducido por el usuario sobre la conexión TCP existente. Para ello se utiliza el método Senddata.

```
Private Sub cbSendData_Click()

  Winsock1.SendData (tbSendData.Text)
  tbSendData.Text = ""
End Sub
```

La rutina Timer vigila el estado actual de la conexión a través de las características „State“ del control Winsock. Un intervalo razonable para el „timer“ es de 500mS

```
Private Sub Timer1_Timer()

  Select Case Winsock1.State
    Case ckClosed
```

```
      lbStatus.Caption = "no Connection"
    Case sckListening
      lbStatus.Caption = "listening for connection"
    Case sckConnectionPending
      lbStatus.Caption = "Connection Pending"
    Case sckConnecting
      lbStatus.Caption = "connecting"
    Case sckConnected
      lbStatus.Caption = "Connection to " + Winsock1.RemoteHostIP
    Case sckError
      lbStatus.Caption = "Connection Error"
      Winsock1.Close
    End Select
    If Winsock1.State <> sckConnected Then
      cbSendData.Enabled = False
      cbDisconnect.Enabled = False
    Else
      cbSendData.Enabled = True
      cbDisconnect.Enabled = True
    End If
End Sub
```

Este procedimiento se llama automáticamente, cuando una conexión de parte del cliente se cierra. Con el método „Close“ se reinicializa la propia administración de la conexión. Con la llamada del método „Listen“ comienza de nuevo el control Winsock a escuchar posibles peticiones de conexión en el puerto elegido.

```
Private Sub Winsock1_Close()
  Winsock1.Close
  Winsock1.Listen
End Sub
```

Si reconoce el elemento de control Winsock una petición de conexión de un Cliente, entonces este procedimiento se ejecuta automáticamente. Con el método „accept“ se acepta la conexión.

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)

  If Winsock1.State <> sckclose Then Winsock1.Close
  Winsock1.Accept requestID
End Sub
```

Cuando se reciben datos se ejecuta automáticamente este procedimiento. Datos recibidos se aceptan con el método „Getdata“ y se muestran en la ventana „Receive Data“.

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
  Dim ReceiveData As String
  Winsock1.GetData ReceiveData
  tbReceiveData.Text = tbReceiveData.Text + ReceiveData
End Sub
```

Procedimiento para el manejo de fallos en la conexión

```
Private Sub Winsock1_Error(ByVal Number As Integer, Description As String, ByVal Scode As
Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long,
CancelDisplay As Boolean)
  Winsock1.Close
  Winsock1.LocalPort = 0
  dummy = MsgBox("Connection Error", vbOKOnly, "TCP simple Server")
End Sub
```

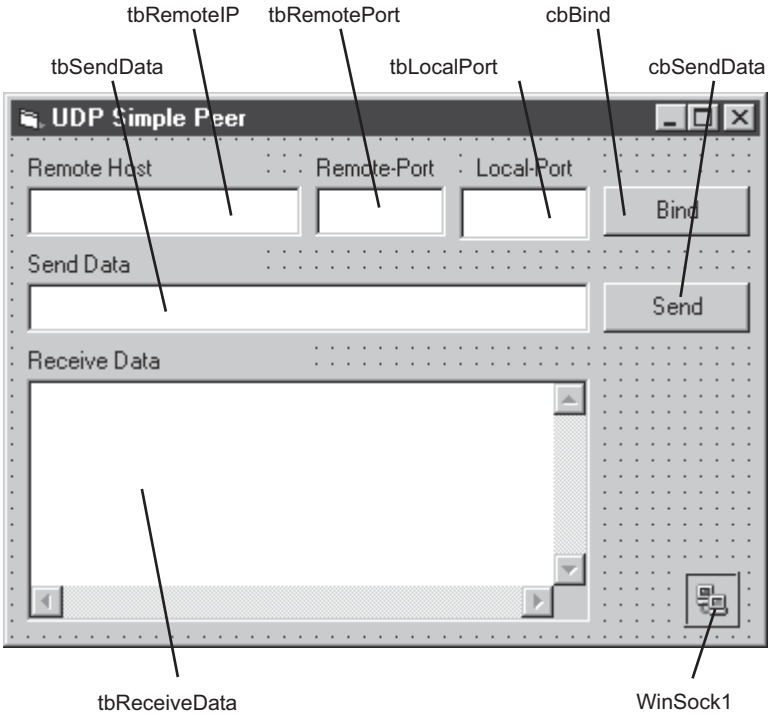
Se puede acceder al servidor que se ha programado como antes con el programa cliente anteriormente mostrado. Pero también otras cualesquiera aplicaciones clientes, por ejemplo, el W&T Com-Server en modo cliente puede realizar conexión con el servidor al elegir el puerto correspondiente.

Un par UDP sencillo en VB

La aplicación UDP se hace cargo de las siguientes tareas:

- Juntar en un Socket la dirección IP y puerto.
- Enviar y recibir datos de texto.

Para ello se desarrolla un formulario con los siguientes elementos:



Todas las variables deberían de aclararse a sí mismas también aquí con la descripción propia de sus nombres.

El siguiente código fuente viene comentado:

Con el método „Bind" se juntan la dirección IP y puerto en un Socket.

```
Private       Sub       cbBind_Click()
  Winsock1.Protocol = sckUDPProtocol
  Winsock1.RemotePort = tbRemotePort.Text
  Winsock1.RemoteHost = tbRemoteIP.Text
  Winsock1.Bind tbLocalPort.Text
  cbBind.Enabled = False
  cbSendData.Enabled = True
End Sub
```

Al apretar el botón send se envía el texto introducido por el usuario como un paquete de datos UDP. Para ello se utiliza el método „senddata“. Una condición es que para ello se haya realizado un Socket con el método „Bind“.

```
Private Sub cbSendData_Click()

  Winsock1.SendData (tbSendData.Text)
  tbSendData.Text = ""
End Sub
```

Al recibir los datos se ejecutará automáticamente este procedimiento.

Los datos recibidos se recogerán con el método „Getdata“ y se mostrarán en la ventana „Receive Data“.

```
Private      Sub      Winsock1_DataArrival(ByVal      bytesTotal      As      Long)

  Dim ReceiveData As String
  Winsock1.GetData ReceiveData
  tbReceiveData.Text = tbReceiveData.Text + ReceiveData
End Sub
```

Para realizar una comunicación de datos con el par UDP, se puede arrancar el mismo par en un segundo ordenador. Igualmente es también posible comunicarse con un W&T Com-Server sobre el par UDP que esté configurado como cliente UDP.

El par UDP aquí mostrado renuncia a cualquier tipo de seguridad en los datos. Esto significa, si se envían datos a una dirección IP no existente o el aparato direccionado no está funcionando entonces los datos caen en el vacío sin que el usuario se de cuenta.

Programación Socket con Delphi

Los ejemplos aquí mostrados se construyeron con la versión estándar de Delphi 5.

Todos los que posean conocimientos básicos de programación Delphi, deben de poder seguir fácilmente los ejemplos de programación.

Delphi 5 pone a disposición en el registro „Internet“ elementos estándares de control para la programación Socket.

Al contrario que Visual Basic, donde sólo con un elemento de control se pueden adaptar las funciones deseadas a través de diferentes parámetros, Delphi ofrece dos elementos de control específicos:

Elemento de control para cliente TCP (Client Socket) 

Elemento de control para servidor TCP (Server Socket) 

Un elemento de control para aplicaciones UDP no está disponible lamentablemente en la versión estándar de Delphi 5.

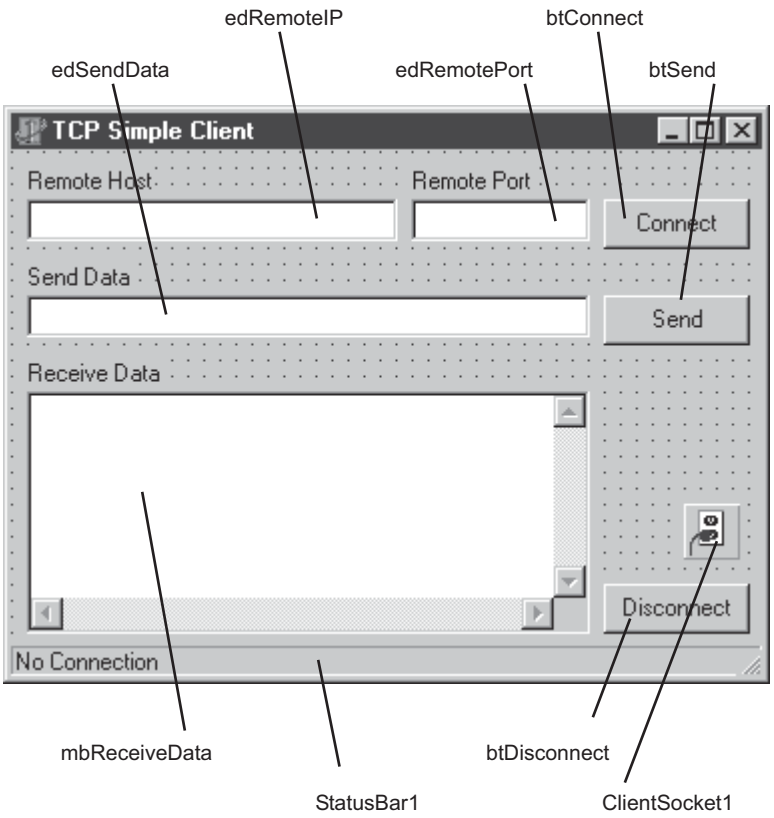
Un cliente TCP en Delphi

Primero queremos construir un cliente TCP, que desarrolle las siguientes tareas:

(El ejemplo completo está a disposición para su descarga en http://www.wut.de)

- Establecimiento de la conexión TCP.
- Envío y recepción de datos de texto.
- Cierre de la conexión TCP.
- Mostrar el estado de la conexión.
- Reconocimiento de fallos.

Para ello se construye un formulario con los siguientes elementos:



Todas las variables deberían de aclararse a sí mismas también aquí con la descripción propia de sus nombres.

Para elementos del tipo „Typ Edit“ se han elegido nombres con el comienzo „ed“, los botones „Buttons“ comienzan con „bt“ y las cajas „Memoboxen“ con „mb“.

El código fuente mostrado viene con comentarios:

La primera parte del código fuente se desarrolla a sí mismo en Delphi al diseñar el formulario, y sirve como declaración de todos los elementos que forman parte de él.

```
unit TCP_Client;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ScktComp, StdCtrls, ComCtrls;

type
  TTCPClient = class(TForm)
    edRemoteIP: TEdit;
    btConnect: TButton;
    edRemotePort: TEdit;
    edSendData: TEdit;
    btSend: TButton;
    mbReceiveData: TMemo;
    btDisconnect: TButton;
    StatusBar1: TStatusBar;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    ClientSocket1: TClientSocket;
    procedure btConnectClick(Sender: TObject);
    procedure btSendClick(Sender: TObject);
    procedure OnConnect(Sender: TObject; Socket: TCustomWinSocket);
    procedure btDisconnectClick(Sender: TObject);
    procedure OnDisconnect(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnRead(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnError(Sender: TObject; Socket: TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  TCPClient: TTCPClient;

implementation
```

```
{$R *.DFM}
```

Aquí comienza el propio programa

El procedimiento siguiente establece la conexión TCP después de apretar el botón Connect con los parámetros de direccionamiento introducidos por el usuario y activa el control Winsock.

```
procedure             TTCPClient.btConnectClick(Sender:             TObject);

begin
   ClientSocket1.Host := edRemoteIP.Text;
   ClientSocket1.Port := strtoint(edRemotePort.Text);
   ClientSocket1.Active := True;
end;
```

Este procedimiento se ejecuta automáticamente al establecerse la conexión con éxito.

```
procedure TTCPClient.OnConnect(Sender: TObject; Socket: TCustomWinSocket);
begin
  btSend.Enabled := True;
  btDisconnect.Enabled := True;
  btConnect.Enabled := False;
  mbReceiveData.Clear;
  Statusbar1.SimpleText := 'Connected to ' + ClientSocket1.Host;
end;
```

Este procedimiento se llama automáticamente al desconectarse la conexión.

```
procedure TTCPClient.OnDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  btSend.Enabled := False;
  btDisconnect.Enabled := False;
  btConnect.Enabled := True;
  Statusbar1.SimpleText := 'No Connection';
end;
```

Procedimiento para el manejo automático de fallos

```
procedure TTCPClient.OnError(Sender: TObject; Socket: TCustomWinSocket;
  ErrorEvent: TErrorEvent; var ErrorCode: Integer);
begin
```

```
  ShowMessage ('Connection Error');
  ClientSocket1.Active := False;
  btSend.Enabled := False;
  btDisconnect.Enabled := False;
  btConnect.Enabled := True;
  Statusbar1.SimpleText := 'No Connection';
end;
```

Al apretar el botón „Send" se enviará el texto introducido por el usuario a través de la conexión TCP existente.

```
procedure TTCPClient.btSendClick(Sender: TObject);

begin
  ClientSocket1.Socket.SendText (edSendData.Text);
  edSendData.Text := '';
end;
```

Al recibir datos se ejecutará automáticamente este procedimiento.
Datos entrantes se aceptarán y se mostrarán en la ventana de „Receive Data".

```
procedure TTCPClient.OnRead(Sender: TObject; Socket: TCustomWinSocket);

begin
  mbReceiveData.Text := mbReceiveData.Text + ClientSocket1.Socket.ReceiveText;
end;
```

Procedimiento para separar la conexión TCP a través de la desactivación del elemento de control „Client Socket"

```
procedure TTCPClient.btDisconnectClick(Sender: TObject);

begin
  ClientSocket1.Active := False;
end;

end.
```

Así se tiene programado también en Delphi con menos de dos hojas de código fuente un cliente TCP con ventana de estatus y manejo de fallos.

Como contrapartida se necesita naturalmente un servidor, que recoja los deseos de conexión del cliente. Esto puede ser un

aparato ya existente como pe. un W&T Com-Server, o una aplicación más en Delphi para servidor.

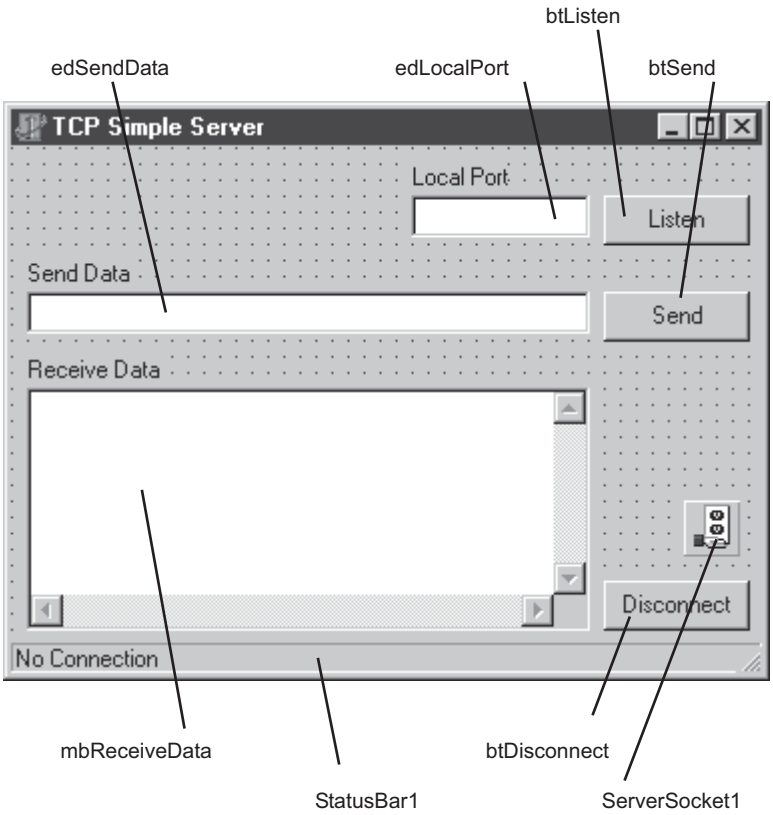
Cómo se puede construir un servidor TCP en Delphi, se mostrará en el siguiente ejemplo.

Un servidor TCP en Delphi

La aplicación servidor se hace cargo de las siguientes tareas:

- „Escuchar“ en la red por si existe una petición de conexión en los puertos soportados
- Hacerse cargo de las conexiones deseadas
- Envío y recepción de datos de texto
- Mostrar el estado de la conexión
- Mostrar los fallos en la conexión
- Cerrar la conexión desde la parte del servidor. (no pertenece a las típicas funciones de un servidor pero también es posible con el programa)

Para ello se construye un formulario con los siguientes elementos:



Todas las variables deberían de aclararse a sí mismas también aquí con la descripción propia de sus nombres.

Los procedimientos Delphi siguientes son necesarios para el servidor:

Como para la aplicación cliente, la primera parte del código fuente de Delphi se crea automáticamente con el diseño del formulario y sirve como declaraciones de los elementos que forman parte.

```
unit TCP_Server;


interface

uses
```

```
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ScktComp, StdCtrls, ComCtrls;

type
  TTCPServer = class(TForm)
    btListen: TButton;
    edLocalPort: TEdit;
    edSendData: TEdit;
    btSend: TButton;
    mbReceiveData: TMemo;
    btDisconnect: TButton;
    StatusBar1: TStatusBar;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    ServerSocket1: TServerSocket;
    procedure btListenClick(Sender: TObject);
    procedure btSendClick(Sender: TObject);
    procedure btDisconnectClick(Sender: TObject);
    procedure OnListen(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnAccept(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnClientRead(Sender: TObject; Socket: TCustomWinSocket);
    procedure OnClientDisconnect(Sender: TObject;
      Socket: TCustomWinSocket);
    procedure OnClientError(Sender: TObject; Socket: TCustomWinSocket;
      ErrorEvent: TErrorEvent; var ErrorCode: Integer);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  TCPServer: TTCPServer;

implementation

{$R *.DFM}
```

Aquí comienza el programa en sí.

Al apretar el botón „listen“ se abre el elemento de control „Server Socket“ y comienza a escuchar las posibles peticiones de conexión en el puerto elegido.

```
procedure TTCPServer.btListenClick(Sender: TObject);

begin
  If edLocalPort.Text <> '' Then
  begin
    ServerSocket1.Port := strtoint(edLocalPort.Text);
    ServerSocket1.Open;
  end
  Else ShowMessage ('No local port!');
end;
```

Este procedimiento se llama automáticamente cuando el elemento de control „Serversocket“ es abierto y espera a las peticiones de conexión.

```
procedure TTCPServer.OnListen(Sender: TObject; Socket: TCustomWinSocket);

begin
  Statusbar1.SimpleText := 'Listening';
  btSend.Enabled := False;
  btDisconnect.Enabled := False;
  btListen.Enabled := False;
end;
```

La aceptación de conexiones es solucionado por el elemento de control „ServerSocket“ automáticamente en el transfondo. Si se acepta una conexión, ejecuta automáticamente el programa este procedimiento:

```
procedure TTCPServer.OnAccept(Sender: TObject; Socket: TCustomWinSocket);

begin
  Statusbar1.SimpleText := 'Connected to ' + Socket.RemoteAddress;
  btSend.Enabled := True;
  btDisconnect.Enabled := True;
end;
```

Este procedimiento se llama automáticamente para la desconexión

```
procedure TTCPServer.OnClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
```

```
          Statusbar1.SimpleText     :=     'Listening';
  ServerSocket1.Open;
  btSend.Enabled := False;
  btDisconnect.Enabled := False;
end;
```

Procedimiento para el manejo automático de fallos

```
procedure TTCPServer.OnClientError(Sender: TObject;
  Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
  var ErrorCode: Integer);
begin
  ShowMessage ('Connection Error');
  ErrorCode := 0;
  ServerSocket1.Close;
  btSend.Enabled := False;
  btDisconnect.Enabled := False;
  btListen.Enabled := True;
  Statusbar1.SimpleText := 'No Connection';
end;
```

Al apretar el botón „Send" se envía el texto introducido por el usuario a través de la conexión TCP existente.

```
procedure TTCPServer.btSendClick(Sender: TObject);
begin
  ServerSocket1.Socket.Connections[0].SendText(edSendData.Text);
  edSendData.Text := '';
end;
```

En la recepción de datos es ejecutado automáticamente este procedimiento. Los datos entrantes se recibirán y mostrarán en la ventana „Receive Data".

```
procedure TTCPServer.OnClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  mbReceiveData.Text := mbReceiveData.Text + Socket.ReceiveText;
end;
```

Procedimiento para la separación de la conexión TCP mediante el cierre del elemento de control „Server Sockets".

```
procedure TTCPServer.btDisconnectClick(Sender: TObject);
begin
  ServerSocket1.Close;
  btListen.Enabled := True;
  btSend.Enabled := False;
  btDisconnect.Enabled := False;
  Statusbar1.SimpleText := 'No Connection';
end;

end.
```

Se puede acceder al servidor programado anteriormente con el programa cliente mostrado. Pero también con otras cualesquiera aplicaciones cliente, pe. el W&T Com-Server en modo cliente, pueden realizar conexiones con el servidor en el puerto elegido correspondiente.

Quien quiera programar con Delphi aplicaciones UDP, tiene la posibilidad de utilizar elementos de control de terceros.

Un ejemplo para ello es el elemento de control en Internet del Belga Francois Piette, que se puede descargar gratis en http://users.swing.be/francois.piette/indexuk.htm.

Los ejemplos mostrados son pensados como sugerencias y deben de invitar para probar y jugar con la transmisión de datos vía TCP/IP. El código fuente puede ser fácilmente adaptado con modificaciones a las aplicaciones finales deseadas.

Configurar TCP/IP-Ethernet

Todos los sistemas operativos actuales ofrecen hoy en día la posibilidad de utilizar TCP/IP como protocolo local de red.

La condición para ello es que el PC disponga de una tarjeta de red ethernet.

Cómo se configura e instala el protocolo TCP/IP en los sistemas corrientes de Microsoft Windows, será descrito paso a paso en las siguientes páginas.

Si su PC ya está conectado en una red Ethernet, debería de averiguar primero, si en esa red ya están en funcionamiento aplicaciones TCP/IP. Pregunte en este caso a su administrador de red, si para su PC ya hay una dirección IP preasignada o que dirección IP puede utilizar para su PC. Además tiene que saber, qué máscara de red, que gateway (puerta de enlace) y que servidores de DNS son válidos para la red.

Por favor anotese los valores utilizados:

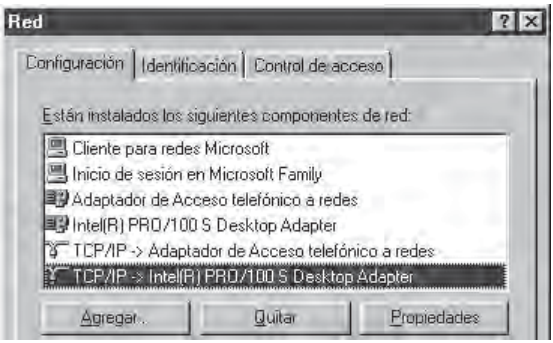
DIRECIÓN IP	_____._____._____._____
MASCARA DE SUBRED	_____._____._____._____
Gateway	_____._____._____._____
SERVIDOR DNS	_____._____._____._____

Instalar y configurar TCP/IP bajo Windows 9x

1- Apretar sobre *Inicio* („Start“) y abrir panel de control.

2- Pulsar dos veces sobre el símbolo de red.

3- Controlar si en la ventana de configuración aparece en la lista, TCP/IP->“tarjeta de red“.

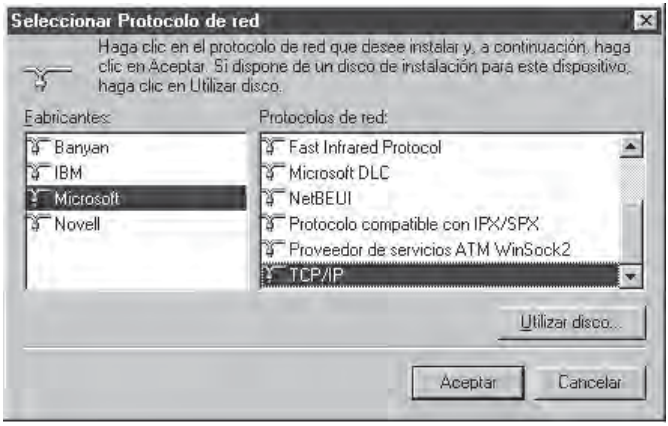


Si la entrada TCP/IP->“tarjeta de red“ existe, continúe con el punto nº. 5.

La entrada TCP/IP-> Adaptador de acceso telefónico a redes no es suficiente para que TCP/IP funcione sobre ethernet correctamente!

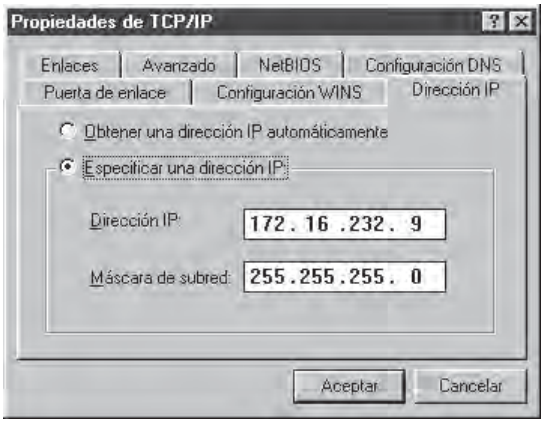
4- Para la entrada inexistente TCP/IP->“tarjeta de red“, apretar el botón „agregar“ y elija el protocolo en la siguiente ventana.

Pulse sobre *agregar* y elija en la siguiente ventana como fabricante *Microsoft* y como protocolo de red TCP/IP



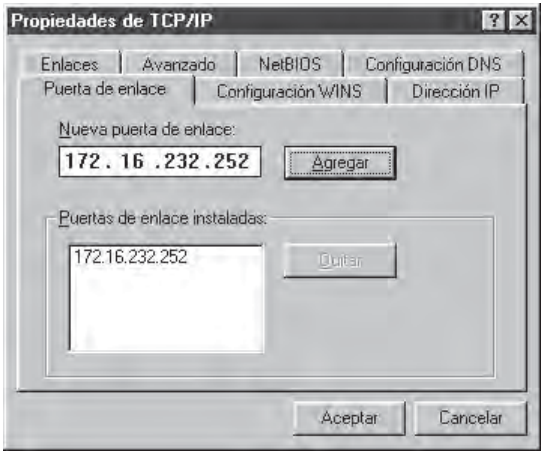
Confirme con Aceptar.
Para la instalación del protocolo se necesita el CD de instalación de su versión de Windows.

5- Marcar TCP/IP->"Tarjeta de red" y elija Propiedades, pregunte a su administrador de red, si la dirección de IP se asigna automáticamente sobre DHCP.



Si no, introduzca la dirección IP y la máscara de subred.

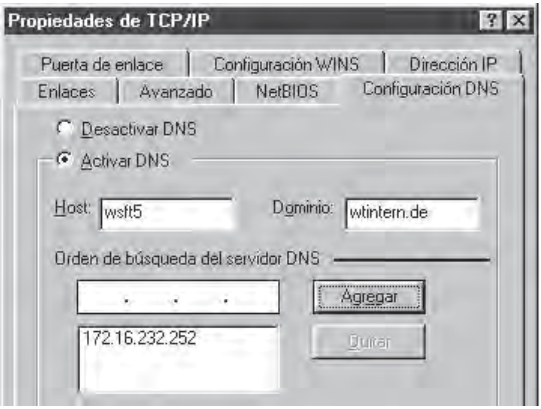
Cambie ahora a la tarjeta de registro, puerta de enlace *gateway*.



Introduzca la dirección IP de la puerta de enlace en el campo *nueva puerta de enlace* y apriete sobre *Agregar*. Sólo cuando la dirección introducida de la puerta de enlace aparezca en el cuadro inferior, permanecerá después de la confirmación sobre *Aceptar*.

Si su red trabaja con soporte de DNS, debería introducir también en el registro configuración DNS la dirección IP del servidor DNS.Sólo cuando la dirección DNS introducida aparezca en el cuadro inferior, permanecerá después de confirmarlo con *Aceptar*.

Además debería de introducir aquí el nombre del host del PC y el dominio, en el cual se administra.



Confirme con *Aceptar*.

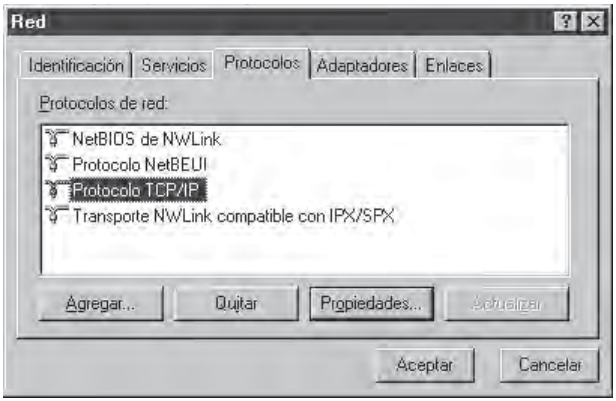
Con esto la instalación de TCP/IP está completa y se le pedirá ahora que reinicialice su PC.

Instalar y configurar TCP/IP bajo Windows NT

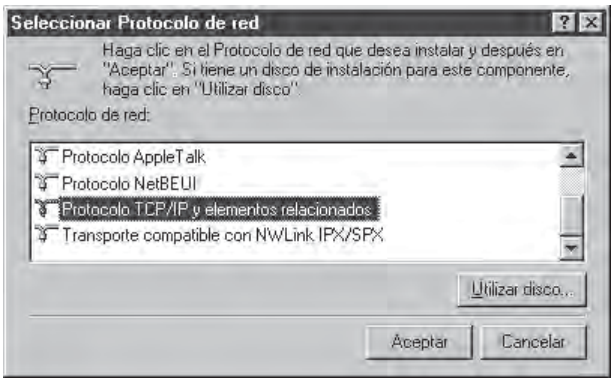
1- Haga clic sobre *Inicio* y abra el panel de control.

2- Doble clic sobre el icono de red 

3- Si en el registro de *Protocolos* existe ya en el listado *TCP/IP –protocolo* entonces puede continuar con el punto 5.



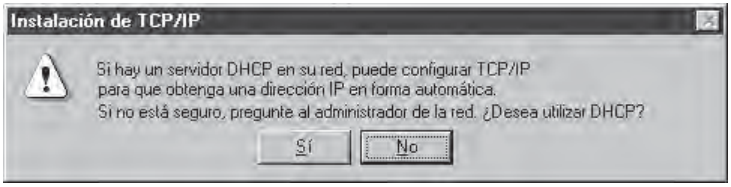
4- Si la entrada *TCP/IP protocolo* no existe, pulse sobre *Agregar* y elija en la ventana siguiente *TCP/IP*.



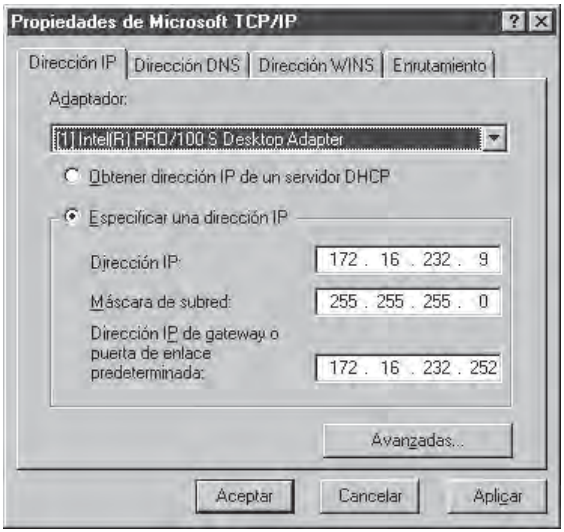
Ahora necesita el CD de instalación de Windows NT. Confirme con *Aceptar*.

5- Para entradas nuevas de soporte TCP/IP apriete sobre *Aceptar* para configurar las propiedades. Si ya existía TCP/IP en su ordenador, marque la entrada *TCP/IP protocolo* y haga clic a continuación sobre propiedades.

Si ha instalado nuevo el soporte de TCP/IP, aparece el siguiente mensaje:



Pregunte a su administrador de red si el servicio DHCP está activado, si no es el caso haga clic sobre *No*.



Introduzca en la ventana siguiente la dirección IP, Máscara de subred y puerta de enlace.

Si su red trabaja con soporte de DNS, debería introducir también en el registro de *configuración DNS* la dirección IP del servidor DNS.

Además debería de introducir aquí el nombre del Host de su PC y el dominio en el cual es administrado.

Confirme con Aceptar.

Con esto está completa la instalación de TCP/IP y ahora se le pedirá que reinicialice su ordenador.

Instalar y configurar TCP/IP sobre WIN2000

1- Haga clic sobre *Inicio* y abra el panel de control.

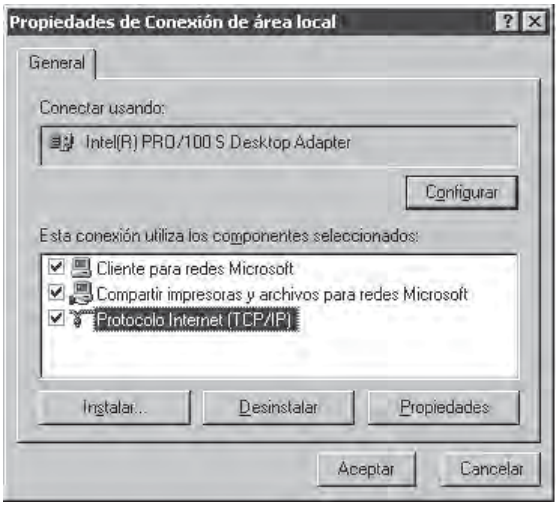
2- Haga doble clic sobre el icono

Conexiones de red y de acceso telefónico

y en la ventana siguiente sobre

Conexión de área local

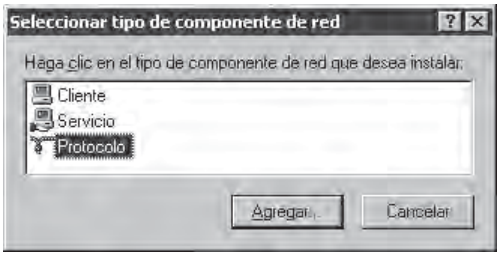
3- Compruebe si el protocolo de internet (TCP/IP) aparece en el listado.



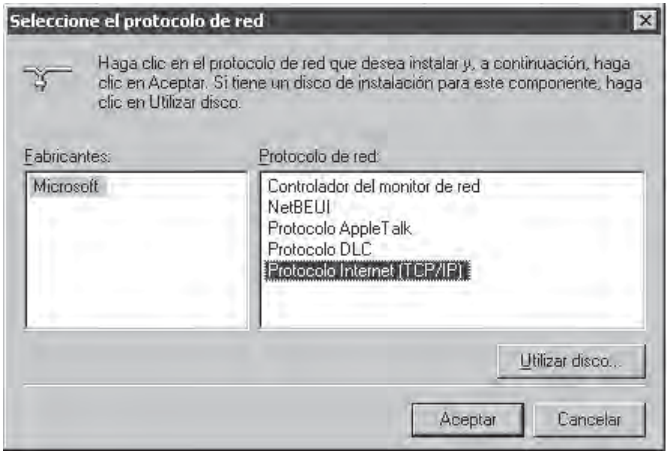
Si la entrada protocolo de internet (TCP/IP) no existe, pulse sobre instalar y elija en la ventana siguiente agregar protocolo

Si la entrada protocolo de internet (TCP/IP) ya existe, contienúe con el punto nº. 5.

4- Si la entrada *protocolo de internet (TCP/IP)* no existe, pulse sobre *instalar* y elija en la ventana siguiente *agregar protocolo*.



seleccione *protocolo de internet (TCP/IP)*

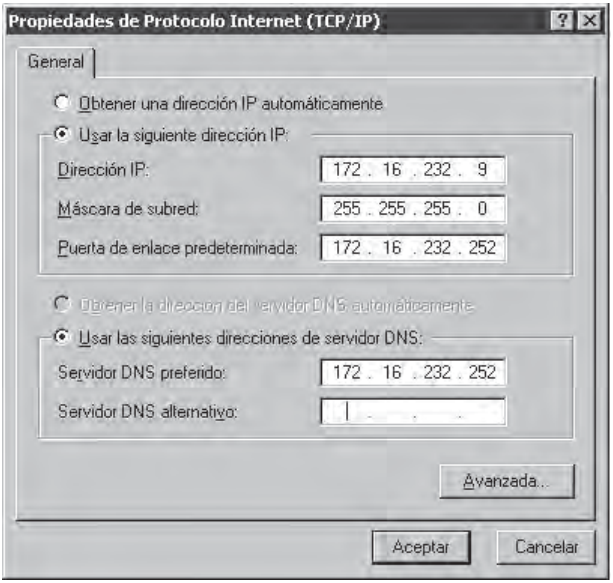


Ahora necesita el CD de instalación de Windows 2000. Después de la confirmación en *aceptar* la lista de protocolos de red se ha aumentado con la entrada *protocolo TCP/IP.*

5- Aparece otra vez la ventana de *propiedades de la red*. Marque la entrada *protocolo de internet (TCP/IP)* y a continuación haga clic sobre *propiedades*.

Si su PC ya está integrado en una red, tiene que averiguar con su administrador de red si el servicio DHCP está funcionando.

Si es éste el caso, entonces elija *obtener una dirección IP automáticamente*.



Si no, introduzca en la ventana siguiente la dirección IP, máscara de subred y puerta de enlace. Si su red trabaja con soporte de DNS, tiene que introducir también la dirección IP del servidor DNS. Confirme con *Aceptar*.

Con esto está completada la instalación de TCP/IP y se le pedirá ahora que reinicialice su ordenador.

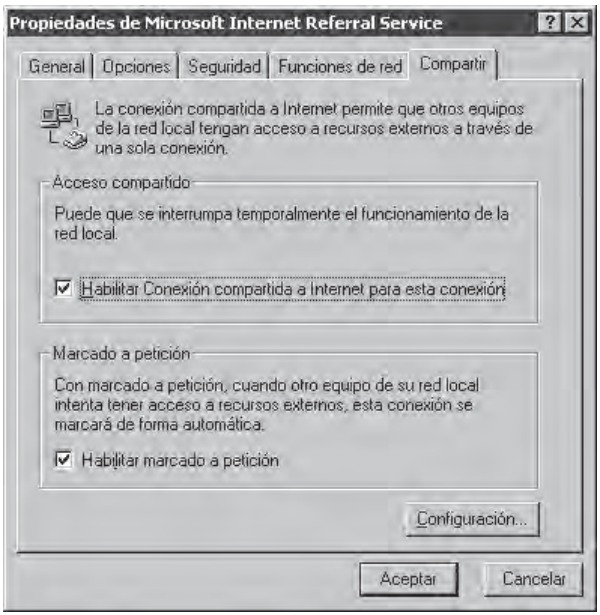
TCP/IP-Ethernet simultáneamente con acceso a internet a través de conexiones telefónicas

Si tiene el ordenador además de un acceso a la red local un acceso a internet a través de conexión telefónica, Windows 2000 ofrece una peculiaridad:

El acceso a internet de este PC puede ser aprovechado por los otros aparatos que están conectados a la red.

Para dar este servicio tiene que realizar doble clic sobre el icono de conexión telefónica que está en el panel de control.

Pulse sobre *propiedades* y elija en la ventana siguiente el registro *compartir*.



Active la *conexión compartida* de la conexión a internet.

El PC trabaja ahora como un router en internet.

Por la activación de este servicio se cambia la dirección IP fija de este PC por sistema a 192.168.0.1

Además derepente este PC aparece como un servidor DHCP y BootP.

Como recordatorio: Un servidor DHCP asigna automáticamente a los participantes de la red bajo petición, una dirección IP. Más información se puede encontrar en el capítulo DHCP.

Los PCs en los cuales está activado el *uso conjunto*, asignan arbitrariamente las direcciones IP del entorno de direcciones 192.168.0 y esto incluso con peticiones BootP.

¡Bajo BootP es normalmente sólo la asignación de direcciones reservadas! por lo tanto

¡hay que tener cuidado! ya que por este comportamiento es posible que a otros usuarios de la red no se les pueda localizar como consecuencia del cambio de direcciones.

Remedio

1. En pequeñas redes donde en un PC con Windows 2000 se ha activado el *uso conjunto*:

   - Desactive en todos los usuarios de la red la función de asignación automática de una dirección IP vía DHCP o BootP.
   - Asigne a los usuarios de la red direcciones fijas IP en el entorno de direcciones 192.168.0.

   Con estas medidas permanecen los usuarios de la red también localizables sobre su dirección IP. Esto es ante todo en sistemas empotrados como el Com-Server muy importante.
   Así de todas formas se permite aprovechar las ventajas del uso conjunto.

2- En redes grandes se tiene que renunciar al uso conjunto. En lugar de esto aconsejamos el uso de un router.

Pequeño abecedario de Redes

10Base2 – *10Mbit/s Baseband 200(185m)/Segmento*
Topología de Ethernet basada en cable coaxial con una velocidad de transferencia de 10Mbit/s.
Otras definiciones corrientes para 10Base2 son también „Cheapernet“ o „Thin-Ethernet“. Se utiliza cable coaxial de 50 Ohmios de impedancia en una versión flexible y delgada para conectar entre si como en un bus las estaciones individuales. El principio y final de un segmento tiene que estar cerrado por impedancias terminales de 50 Ohmios.
Los transceptores están integrados en las tarjetas de red, así que el bus se lleva directamente a cada puesto de trabajo, donde se conectan a cada ordenador con conectores BNC-T. La atenuación del cable, así como en parte el elevado número de conectores limitan un segmento 10Base2 a 185m máximo con 30 conexiones máximas. Entre 2 estaciones no puede haber más de 4 repetidores.
El punto débil de las topologías de bus físicas de ethernet están en el hecho de que una interrupción del cable pe. al separar un conector, tiene como consecuencia la caída de toda la red.

10Base5 - *10Mbit/s BASEband 500m / Segmento*
10Base5 es la especificación original de Ethernet. El cableado consta aquí de un cable coaxial en bus con 50 Ohmios de impedancia y una distancia máxima permitida de 500m (Yellow Cable). Debido a la técnica del cable coaxial con 2 conductores (núcleo y pantalla) permiten tanto 10Base5 como 10Base2 sólo una comunicación semidúplex. Los usuarios de la red se conectan con transceptores externos que recogen las señales directamente del cable bus con unas garras vampiro, sin que lo interrumpa como un conector o algo parecido. Una vez separados los datos de envío, recepción o colisión, se ponen a disposición desde el transceptor en un conector D-SUB de 15 pines. La conexión al aparato final tiene lugar con un cable TP de 8 hilos de una longitud máxima 50m. Entre dos estaciones cualesquiera no debe de haber más de 4 repetidores. Esta regla

sólo concierne a repetidores „seguidos“, en la realización de estructuras de red en forma de árbol se pueden colocar un número mayor de repetidores.
Por la utilización de cables de alta calidad sin ninguna interrupción por conectores se consiguen ventajas en la longitud de los segmentos y en el número máximo de posibles conexiones por segmento (max. 100).
El grosor y la rigidez del Yellow Cable, así como los transceptores externos, costes adicionales del Yellow Cable, son las principales desventajas de 10Base5 y han influido decisivamente a la introducción de 10Base2.

10BaseT - *10Mbbit/s BASEband TwistedPair*
Con la definición de 10BaseT se separa la topología física de la lógica. El cableado se realiza de una forma en estrella, saliendo desde un HUB como componente central activo. Se utiliza como mínimo un cable con dos pares de hilos de la categoría 3 con 100 Ohmios de impedancia, en el cual los datos son transmitidos por separado según la dirección de envío o recepción. Como conector se usa el tipo RJ45 con 8 pines en el que los pares se colocan sobre los pines 1 / 2 y 3 / 6. La longitud máxima de un segmento (igual a la conexión desde el Hub hasta el terminal) está limitada a 100m. Su origen lo tiene la topología 10BaseT en USA porque permitía utilizar los cables típicos de allí para teléfonos también para el uso de la red. En Alemania no existía esa ventaja ya que para la telefonía se colocó cable en estrella 4 que no cumplía las condiciones de la categoría 3.
Ruptura de cables o conectores desconectados, que para todas las estructuras físicas de bus producían una caída del segmento total, se limitan en 10BaseT a un sólo lugar de trabajo.

100BaseT4 – *100Mbit/s BASEband Twisted 4 Pairs*
100BaseT4 especifica una transmisión Ethernet con 100Mbit/s. Al igual que 10BaseT se trata aquí de una estructura física en estrella con Hub en el centro. Se utiliza igualmente un cable categoría 3 con 100 Ohmios de impedancia, conectores RJ45 y una longitud máxima de 100m. Las diez veces más de la velocidad de transmisión de 100Mbit/s con el mismo cable categoría 3 con ancho de banda de 25 MHz se consigue

también con el uso de los 4 pares de hilos. Para cada sentido de datos se utilizan 3 pares al mismo tiempo en 100BaseT4.

100BaseTX – *100 Mbit/s BASEband Twisted 2 Pairs*
100BaseTX especifica la transmisión de 100 Mbit/s sobre 2 pares de hilos de un cableado con componentes de categoría 5. Enchufes RJ45, Patch panel, etc. tienen que cumplir, según esta categoría, una frecuencia de transmisión de 100MHz.

Administrador (Administrator)
El administrador de la red, que tiene en la red local derecho de acceso ilimitado y es responsable de la gestión (administración) y el cuidado de la red. El administrador asigna entre otras cosas las direcciones IP en su red y tiene que mantener la individualidad de cada dirección IP.

ARP - *Address Resolution Protocol*
Con ARP se averigua la dirección de ethernet en una red que corresponde a su dirección IP. El orden resuelto se administra en cada ordenador en su tabla ARP. En sistemas operativos Windows se puede influenciar con ayuda de la orden ARP la tabla de ARP.
Características y parámetros del comando ARP en la ventana de DOS:

- ARP -A muestra un listado de las entradas en la tabla ARP.
- ARP -S <dirección IP> <dirección ethernet> introduce una entrada estática en la tabla ARP.
- ARP -D<dirección IP> elimina una entrada de la tabla ARP.

ARP está definido en el estándar de internet RFC-826 ;
ver 26

Arquitectura Cliente Servidor
Sistema de „inteligencia distribuida“, en el cual el cliente establece una conexión en el servidor para hacer uso de los servicios ofrecidos por el servidor. Algunas aplicaciones del servidor pueden atender a varios clientes al mismo tiempo;
ver 19

AUI – *Attachment Unit Interface*
Interfaz para la conexión de un transceptor ethernet externo. Una vez separados los datos de envío, recepción o colisión, se ponen a disposición desde el transceptor en un conector D-SUB de 15 pines. La conexión al aparato final tiene lugar con un cable TP de 8 hilos de una longitud máxima 50m.
Mientras el interfaz AUI en el pasado se utilizó principalmente para acoplar los aparatos terminales a los transceptores 10Base5 (Yellow Cable), se usan hoy en día para la conexión al transceptor F.O. (fibra óptica).

BNC – *Bayonet Neill Concelmann*
En el conector BNC se trata de un cierre de bajoneta para conectar dos cables coaxiales. Los conectores BNC se utilizan en redes 10Base2 para la conexión mecánica del cable RG-58 (Cheapernet).

BootP – *Boot Protocol*
Este viejo protocolo para arrancar PCs sin disco duro a través de la red, es el predecesor de DHCP. También los servidores DHCP modernos soportan todavía peticiones BootP. En la actualidad se usa BootP principalmente para asignar una dirección IP a sistemas empotrados. Para ello tiene que existir en el servidor DHCP una entrada reservada en la cual hay ordenada una dirección IP fija para la dirección MAC del sistema empotrado.

Broadcast
Como broadcast se designa a una llamada en general a todos los usuarios de la red. Una aplicación típica de broadcast es la petición ARP (ver ARP). También otros protocolos, como RIP o DHCP, utilizan mensajes de broadcast.
Los Mensajes broadcast no se envían a través de un router o puente.

Browser (Navegador)
Es un programa cliente con interfaz gráfico de usuario que proporciona al usuario la posibilidad de ver páginas Web y utilizar otros servicios de internet.
ver 48.

Cheapernet
Otra forma de nombrar a Ethernet configurada en 10Base2.

Cliente (client)
Ordenador o aplicación, que utiliza los servicios de los llamados servidores. El servicio del servidor puede ser por ejemplo la disponibilidad de un interfaz COM o de impresora en la red, pero también Telnet y FTP;
ver 19.

Com Server
Es un aparato terminal de redes TCP/IP Ethernet, que pone a disposición interfaces de aparatos serie y puntos digitales de entradas/salidas en la red.
ver 151.

DHCP – *Dynamic Host Configuration Protocol*
Reparto dinámico de direcciones IP desde un rango determinado de direcciones. DHCP se utiliza para configurar PCs en una red TCP/IP automáticamente, es decir, sin acceso manual, centralmente y con ello unitario. El administrador del sistema decide cómo se asignan las direcciones IP y fija en qué momento se deben de asignar.
DHCP está definido en el estándar de Internet RFC 2131 (03/97) y RFC 2241 (11/97).
ver 36.

Dirección E-Mail
Se necesita una dirección e-mail para poder enviar a un usuario correo electrónico y se compone siempre del nombre del buzón del usuario y el dominio destino, separados por un símbolo @.
Ejemplo: info@wut.de describe el buzón de información en el servidor mail de W&T, ver 71.

Dirección IP
La dirección IP es un número de 32 Bit, que identifica claramente a cada usuario en Internet así como en Intranet. Se compone de una componente de red (Net-ID) y una parte de usuario (Host-ID).

ver 16 pag foto.

DNS - *Domain Name Service*
Los usuarios de la red son reconocidos en Internet por las direcciones numéricas IP. Pero como se recuerdan mejor los nombres que los números, se introdujo el DNS.
DNS consta de un sistema estructurado jerárquicamente: Cada nombre de dirección se identifica por un dominio de nivel más alto (Top Level Domain; „de“, „com“, „net“, „es“...) y dentro de éste por un subdominio (Sub Level Domain). Cada subdominio puede (no tiene porqué) contener otros dominios subordinados. Las partes individuales de esta jerarquía de nombres están separadas unas de otras por puntos.
Si se introduce por un usuario un nombre de dominio para dirigirse a él, entonces pregunta la pila TCP/IP al servidor DNS más próximo por la correspondiente dirección IP.
Los recursos de una red tienen que recibir sensatamente un nombre de dominio, que represente contextualmente a los servicios ofrecidos o al nombre de la empresa del proveedor. Así pe. se despliega wut.de, en el top level de = Alemania (Deutschland) y el sub level wut = Wiesemann & Theis GmbH ;
ver 41.

E-Mail
Correo electrónico a través de Internet e Intranet;
ver 71.

Embedded System (Sistemas empotrados)
Como Embedded System se denomina a un sistema controlado por un microprocesador, que como parte intercalada en un aparato o máquina, procesa a fondo datos o en su caso controla procesos.

Ethernet
Ethernet es actualmente la tecnología en redes locales más utilizada. Existen tres diferentes topologías de Ethernet 10Base2, 10Base5 y 10BaseT ; la velocidad de transmisión es 10Mbit/s.
ver 11.

Dirección Ethernet
La dirección física e invariable de un componente de la red en Ethernet
ver 13.

Fast Ethernet
Fast Ethernet es por así decirlo, una actualización de la topología 10BaseT de 10 Mbit/s a 100 MBit/s, ver 100BaseT4 y 100BaseTX.

Firewall (Corta fuegos)
Bajo Firewall o cortafuegos se entiende un componente de red que adapta como un router una red interna (Intranet) a una red pública (pe. Internet). Aquí se pueden limitar o negar completamente los accesos de una red a la otra dependiendo de la dirección de acceso, el servicio utilizado o la autentificación e identificación del usuario.
Una característica más puede ser el encriptado de datos, cuando pe. la red pública sólo se usa como camino de tránsito entre dos partes de Intranet separadas en el espacio.

FTP- ***File Transfer Protocol***
FTP es un protocolo basado en TCP/IP, que posibilita la transmisión de ficheros completos entre dos usuarios de la red.
Ver 84.

Gateway (Puerta de enlace)
Los gateways, al igual que los puentes y routers, conectan entre sí diferentes redes. Mientras los puentes y routers sólo adaptan el nivel físico de las redes (pe. Ethernet a RDSI) y el propio protocolo (pe. TCP/IP) lo dejan tal cual, los gateways ofrecen la posibilidad de conseguir un acceso a redes de diferentes protocolos (pe. TCP/IP a PROFIBUS). Una puerta de enlace tiene entre otras cosas la tarea de traducir diferentes protocolos de comunicación.
Atención: En la configuración de red en sistemas operativos Windows se exige también la introducción de un Gateway. ¡Este dato se refiere a un router existente en la red!

HTML – *Hypertext Markup Language*
Es un lenguaje de representación que indica con palabras clave, cómo se deben de mostrar los contenidos en el navegador (Browser), dónde se sitúan los elementos multimedia y qué elementos están interconexionados.
ver 49, 52.

HTTP - *Hypertext Transfer Protocol*
El protocolo HTTP está basado en TCP y regula la petición y transmisión de contenidos Web entre el servidor HTTP y un navegador. Por eso es HTTP en la actualidad el protocolo más utilizado en Internet.
ver 49, 50.

Hyperlink (hipervínculo)
Es una referencia a otra página Web o contenidos dentro de una página Web. El usuario alcanza la página Web deseada con un sencillo click sobre el elemento interconexionado.
ver 48, 54.

Hub
Un hub, a menudo nombrado como acoplador en estrella, ofrece la posibilidad de conectar entre sí varios usuarios de la red de una forma en estrella.
Los paquetes de datos, que se reciben en un puerto, se entregan de igual manera a todos los otros puertos. Además de los hubs para 10BaseT (10Mbit/s) y 100BaseT (100Mbit/s) existen los hubs – Autosensing, que reconocen automáticamente, si el aparato conectado trabaja a 10 o 100 Mbit/s. Con los hubs – Autosensing se pueden conectar sin problemas viejos aparatos 10BaseT en las nuevas redes 100BaseT.

ICMP – *Internet Control Message Protocol*
El protocolo ICMP sirve para el envío de información de estado y mensajes de fallos entre nudos de la red IP. El ICMP ofrece además la posibilidad de petición de eco: de esta forma se puede averiguar si un lugar concreto está localizable; ver también PING.

Internet
Internet es en la actualidad la conexión de redes más grande del mundo, que ofrece a los usuarios conectados a ella una infraestructura de comunicaciones casi ilimitada.
A través del uso de TCP/IP, los usuarios pueden disponer independientemente de la plataforma los servicios ofrecidos en Internet como e-mail, FTP, HTTP, etc.

Intranet
Una red cerrada (dentro de una empresa), en la cual dentro de sus fronteras los usuarios pueden utilizar servicios típicos de Internet como e-mail, FTP, HTTP, etc. Normalmente existe paso de una Intranet a Internet a través de un Router o Firewall.

IP – *Internet Protocol*
EL protocolo que posibilita la conexión de usuarios que se encuentran en diferentes redes.
Ver 15.

LAN – *Local Area Network*
Es una red local dentro de unos límites espaciales bajo la aplicación de un medio de transmisión rápido como pe. Ethernet.

MAC – ID
La dirección física inalterable de un componente de red (MAC = Media Access Control). Ver dirección *Ethernet*
y 13.

Máscara de subred
Valor de 32 Bit que fija, que parte de la dirección IP direcciona a la red y cual al usuario. Ver pag. 28.

NAT – *Network Address Translation*
Por la expansión explosiva de Internet en los últimos años, las direcciones IP libres se han convertido en un bien escaso y se asignan sólo de forma cautelosa. Aquí es donde entra en juego NAT, cuando se quieren conectar las redes de empresas a Internet. La red de la empresa se conecta a Internet con un router capaz de soportar NAT, que trabaja internamente, eso sí, en un rango propio de direcciones IP independientes a

Internet. La red está disponible al exterior por sólo una (o algunas pocas) direcciones IP. Dependiendo del número de puerto en el paquete recibido TCP/IP se redirecciona este paquete al determinado usuario interno de la red.

Puente (Bridge)
Los puentes unen partes de redes entre ellas y deciden dependiendo de la dirección ethernet qué paquetes pueden atravesar el puente y cuales no. La información necesaria para ello la tiene la tabla de puente, que dependiendo de cada modelo de administrador, tienen que ser introducidas o son generadas por el propio puente. Ver Router.

PING – *Packet Internet Groper*
Ping sirve para fines de diagnóstico en las redes TCP/IP; con ayuda de esta función se comprueba si un usuario concreto de la red existe y de hecho está accesible. Ping trabaja con el protocolo ICMP, el cual a su vez está basado en IP. Si un usuario envía con el comando PING una petición ICMP-Request, entonces la estación destino contesta con una respuesta ICMP-Reply al remitente.
La ejecución del comando PING dirección IP en la ventana DOS exige, del usuario introducido por la dirección IP, una contestación.
Además se pueden introducir diversos parámetros:

-t	Repite el comando PING en una secuencia continua, hasta que el usuario lo interrumpe con <CTRL> C.
-n count	repite el comando PING el número de veces de „count“.
-l size	„size“ indica con cuántos Byte se llenará el paquete ICMP. En los Com-Servers la configuración por defecto es máximo 512 byte.
-w timeout	„timeout“ especifica, cuánto (en milisegundos) tiempo hay que esperar para la contestación.

Ejemplo:

```
PING 172.16.232.49 -n 50
```

envía 50 comandos PING a la estación 172.16.232.49. Si el usuario está disponible, se muestra este mensaje de contestación:
Reply from 172.16.232.49: bytes = 32 time = 10ms TTL = 32
Si la contestación no existe, el mensaje mostrado es:
Request timed out.
Los paquetes ICMP que se utilizan con PING están definidos en el estándar de Internet RFC-792.

POP3 – *Post Office Protocol Version 3*
Para recoger e-mails recibidos del buzón situado en el servidor de mail, se utiliza en mayor parte el protocolo POP3. También POP3 está basado sobre TCP,
ver 71 y 75.

PPP – *Point to Point Protocol*
PPP es un descendiente ampliado de SLIP y muestra una corrección de fallos mejorada.
Exactamente como SLIP, ofrece PPP la posibilidad de conectar aparatos TCP/IP, que tienen conector LAN, sobre el interfaz Serie a las redes TCP/IP.

RDSI – *Red Digital de Servicios Integrados*
La RDSI es el nuevo estándar de las telecomunicaciones clásicas y ha sustituido completamente en Alemania a la red analógica telefónica interurbana. En RDSI se integran teléfono y fax pero también videoconferencia y datos. Por lo tanto, con RDSI se puede transmitir, dependiendo de los aparatos terminales, Voz, texto, gráficos y otros datos.
RDSI dispone por una conexión básica, interfaz So, de 2 canales básicos (Canales B) con 64 Kbit/s cada uno y de un canal de control (Canal D) con 16 Kbit/s. La conexión de usuario digital tiene una velocidad de transmisión total máxima de 144Kbit/s (2B+D). En los dos canales B se puede tener simultáneamente dos servicios diferentes con una velocidad de transmisión de 64 Kbit/s sobre un cable.

Router RDSI
Los routers RDSI permiten que dos redes locales se conecten entre si por la red RDSI de un proveedor telefónico. Para ello

poseen los routers RDSI además de las funciones de un router, el manejo de conexiones RDSI.

Repeater (Repetidor)
En redes locales, sirve para conectar dos segmentos Ethernet para que se amplíe el alcance de un único segmento. Los repetidores pasan los paquetes de datos de un segmento de red a otro, y „refrescan" las señales eléctricas según la norma; el contenido de los paquetes de datos los dejan igual. Si el repetidor reconoce en uno de los segmentos conectados un fallo físico, entonces se separa la conexión a ese segmento. Esta retención se reestablece automáticamente cuando el fallo ya no existe.
Entre 2 estaciones no se puede tener más de 4 repetidores. Esta regla sólo incluye a repetidores que están uno detrás de otro. En la realización de estructuras de red en árbol se pueden tener muchos repetidores.

Resistencia Terminal
En topologías de red coaxiales con 10Base5 o 10Base2 tiene que terminarse al principio y al final de cada tramo de red con una resistencia terminal (Terminator). El valor de cada resistencia terminal tiene que corresponder al valor de la impedancia del cable; en 10Base5 o 10Base2 es de 50 Ohmios.

RIP – *Routing Information Protocol*
El protocolo de enrutado como RIP sirve para que los cambios de los caminos entre dos sistemas en red sean comunicados a las redes interesadas y así posibilitar los cambios dinámicamente de la tabla de rutas. RIP está definido en el estándar RFC 1058.

Router
Conectan dos redes diferentes, aunque a diferencia de los puentes (Bridge) no se decide dependiendo de la dirección Ethernet sino de la dirección IP para encaminar los paquetes de datos.
Ver *Bridge* y 22.

Router SLIP
Un router SLIP ofrece la funcionalidad y el hardware para conectar a una red, aparatos serie que dispongan de una pila TCP/IP.
Com-Server pe. dispone de un modo de funcionamiento como SLIP-Routing.

SLIP – *Serial Line Internet Protocol*
SLIP ofrece una posibilidad sencilla de transmitir paquetes de datos TCP/IP por una conexión serie punto a punto. Con esto también pueden los terminales, que no disponen de un conector LAN, conectarse por el interfaz serie a la red.
SLIP trabaja según un algoritmo sencillo sin el procedimiento de seguridad de datos: Al propio paquete IP de datos se le coloca delante un símbolo de inicio (decimal 192) y un símbolo de final (decimal también 192) detrás. Para mantener la transparencia binaria, se sustituyen los símbolos de comienzo y final anteriores en el paquete de datos por otras secuencias. SLIP está descrito en RFC 1055.

Servidor DNS
Los servidores DNS ponen a disposición en Internet el servicio de resolver un nombre de dominio en una dirección IP.

Sistemas bus
En un sistema de bus se reparten varios equipos un único cable de datos (el bus). Ya que sólo puede utilizar un equipo al mismo tiempo el cable de datos, los sistemas de bus necesitan siempre un protocolo para regular los derechos de acceso. Sistemas de bus clásicos son las topologías de Ethernet 10Base2 y 10Base5.

SMTP – *Simple Mail Transfer Protocol*
SMTP regula el envío de e-mails del cliente al servidor de mail (servidor SMTP) y entre servidores mail, también está basado en TCP,
ver 71 y 74.

SNMP – *Simple Network Management Protocol*
SNMP funciona con UDP y posibilita la administración central y vigilancia de componentes de red.
SNMP está descrito en los siguientes estándares: RFC 1052, RFC 1155, RFC 1156, RFC 1157, RFC 1213 y RFC1441.

STP – *Shielded Twisted Pair*
Cable de datos apantallado, en el cual cada dos hilos están trenzados, ver twisted pair.

Switch
Un switch ofrece como un Hub la posibilidad de conectar entre sí de forma en estrella varios usuarios de la red. Los Switches perfeccionan la función de un hub con la de un puente: Un switch „aprende“ la dirección ethernet del aparato conectado a un puerto y dirije allí sólo aquellos paquetes de datos que direccionan a ese usuario. Una excepción son los mensajes de Broadcast, que se envían a todos los puertos (aquí se diferencian los switch en su función de la de un puente que generalmente no redirije los mensajes broadcast).
Además de los switches 100BaseT (100MBit/s) existen los llamados Autosensing, que reconocen automáticamente si el aparato conectado trabaja a 10 o 100 Mbit/s. Con switches Autosensing se pueden conectar aparatos antiguos de 10BaseT en las redes nuevas 100BaseT.

TCP – *Transmision Control Protocol*
TCP funciona sobre IP y no sólo proporciona la conexión del usuario durante la transmisión de datos, sino que también asegura la corrección de los datos y la correcta secuencia de los paquetes de datos.
Ver 19.

TCP/IP Stack (la pila TCP/IP)
Es una parte del sistema operativo o una parte del sistema por encima de los controladores, que pone a disposición todas las funciones necesarias y driver para el soporte del protocolo IP.

Telnet – *Terminal Over Network*
En el pasado, se utilizaba Telnet para el acceso remoto por la red en servidores UNIX. Con una aplicación Telnet (Cliente

Telnet) se puede lograr desde un ordenador cualquiera un acceso remoto por la red a otro ordenador (Servidor Telnet). En la actualidad, también se utiliza Telnet para la configuración de componentes de red como pe. Com-Server. Telnet funciona normalmente bajo TCP/IP por el puerto nº. 23; para aplicaciones especiales se pueden utilizar también otros números de puertos. Telnet se basa en TCP/IP como protocolo de transmisión y seguridad.
Ver 80.
Telnet está descrito en el estándar de Internet RFC 854.

TFTP – *Trivial File Transfer Protocol*
El protocolo Trivial file Transfer (TFTP) es junto a FTP un protocolo más para la transferencia de ficheros completos. TFTP ofrece sólo un mínimo de comandos, no soporta ningún mecanismo complicado de seguridad y utiliza UDP como protocolo de transmisión. Ya que UDP es un protocolo inseguro, se implementaron en TFTP algunos mecanismos de seguridad mínimos.
Ver 88.
El trivial File Transfer Protocol está descrito en los estándares 783, 906, 1350 y 1785.

Transceptor (transceiver)
La palabra transceiver es un compuesto de Transmitter (transmisor) y Receiver (receptor). El transceiver realiza el acceso a la red física de una estación en el Ethernet y está integrado en las topologías modernas de Ethernet 10Base2 y 10BaseT en la tarjeta de red. Sólo en 10Base5 (ver también conexión AUI) está implementado el transceptor como componente externo directo en el cable de la red.

Twisted Pair
Cables de datos, que para cada dos hilos del cable están trenzados entre sí. Por la asociación de pares de hilos se logra una clara reducción en el acoplamiento entre los pares de hilos. Se diferencian en cables twisted pair entre no apantallados UTP (Unshielded Twisted Pair) y apantallados STP (Shielded Twisted Pair).

Los cables TP se colocan sobre todo en la tecnología de redes y se clasifican según sus frecuencias máximas de transmisión; en la práctica se instalan en mayor parte dos tipos:

- Cable Categoría 3 permite una frecuencia máxima de transmisión de 25 MHz, suficiente para el uso en 10BaseT, pero también en redes 100BaseT4.
- Cable Categoría 5 permite una frecuencia máxima de transmisión de 100MHz y es suficiente con ello para todas las topologías actuales de red.

UDP – *User Datagram Protocol*
UDP es un protocolo, que como TCP se basa sobre IP, pero al contrario trabaja sin conexión y que no posee ningún mecanismo de seguridad. La ventaja de UDP frente a TCP es la alta velocidad de transmisión.
Ver 22.

URL – *Uniform Resource Locator*
Protocolo de información y dirección para el navegador. Con el URL fija el usuario para el navegador, qué protocolo se va a utilizar, en qué servidor Web está la página y dónde se encuentra en ese servidor Web.
Ver 49.

UTP – *Unshielded Twisted Pair*
Al contrario que Shielded Twisted Pair un cable de datos no apantallado, en el cual cada dos hilos del cable están trenzados entre sí.

Web-Based Management
Como Web-Based Management se entiende la posibilidad, sin software especial, de configurar equipos terminales por la red directamente con la ventana del navegador
(Browser)

WWW – *World Wide Web*
WWW se iguala muy a menudo con Internet. Esto no es correcto del todo: mientras Internet describe los recorridos físicos de conexión, la WWW define un estándar, que proporciona al usuario por un entorno gráfico con los accesos más sencillos de manejo, los servicios de Internet más comunes.

Con un click del ratón se piden páginas Web, se envían e-mails y se descargan ficheros.
Ver 48.

Sistemas numéricos

Junto al sistema numérico decimal (símbolos disponibles: 0-9, nueva posición en 10) se utiliza muy a menudo en la tecnología Computacional el sistema numérico binario (símbolos disponibles: 0-1, nueva posición en 2) y el sistema hexadecimal (símbolos disponibles: 0-9 + A-F).

En la siguiente tabla puede encontrar algunos ejemplos para la representación de valores comunes en los tres sistemas numéricos:

binary	dec.	hex	binary	dec.	hex
0	0	0	11111	31	1F
1	1	1	100000	32	20
10	2	2	...	...	...
11	3	3	111111	63	3F
100	4	4	1000000	64	40
101	5	5	...	...	...
110	6	6			
111	7	7	...	...	...
1000	8	8	1111111	127	7F
1001	9	9	10000000	128	80
1010	10	A	11000000	192	C0
1011	11	B	11100000	224	E0
1100	12	C	11110000	240	F0
1101	13	D	11111000	248	F8
1110	14	E	11111100	252	FC
1111	15	F	11111110	254	FE
10000	16	10	11111111	255	FF

Web-IO

La razón más decisiva para dotar un aparato con un interfaz de red era en el pasado la alta velocidad de transmisión. Con el continuo crecimiento de redes de empresas y crecimiento conjunto de Intranet e Internet, gana peso, la flexibilidad, el uso de infraestructuras existentes en la decisión de no sólo a PCs, File-Server e impresoras dotarles de una conexión de red.

Para finalizar, queremos mostrarles la idea de conectar diferentes señales directamente por la red para analizarlas y controlarlas.

Para esta tecnología hemos elegido el nombre de Web-IO.

Com-Server Ejemplos de Aplicaciones desde la Praxis

Los Com-Server, son pequeñas cajas que poseen por un lado una conexión Ethernet y por el otro de 1 a 4 puertos serie.

La conexión Ethernet trabaja dependiendo del modelo a 10Mbit o 10/100Mbit autosensing (reconocimiento automático). Para la parte serie se puede elegir RS232, RS422, RS485 o 20mA.

El soporte de los protocolos TCP, FTP y Telnet (tanto como Cliente y servidor) así como UDP, SNMP, BootP, RARP y ARP permiten casi cualquier aplicación imaginable.

Como fuentes de alimentación son posibles 230V, 12-24V, 5V (compatible TTL) y 3V. Detalladas hojas de características de los diferentes Com-Server se encuentran en el anexo.

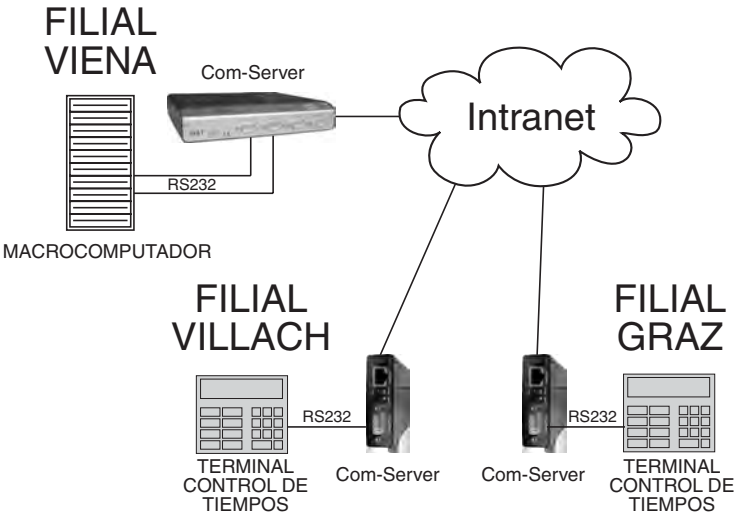
Box-to-Box – El túnel por la red

Dos Com-Server se configuran de tal forma que los datos, que entren en el puerto serie del Com-Server 1, pasen por la red al Com-Server 2. El Com-Server 2 entrega los datos serie otra vez. Por supuesto todo esto funciona en ambas direcciones.

Un ejemplo:
Los datos de control horario en un banco de Viena se transmiten por RS232 desde el terminal de control horario hasta una macrocomputadora UNIX. Los datos de las filiales Villach y Graz se enviaban hasta ahora por correo en un disquete.

En la actualidad, los datos se transmiten sencillamente por un Com-Server en modo Box-to-Box a través de la conexión exis-

tente de Intranet. El primer Com-Server „introduce“ los datos RS232 en paquetes TCP y lo enruta por la red hacia el segundo Com-Server. Este „desempaqueta“ los datos RS232 y los entrega al ordenador central.

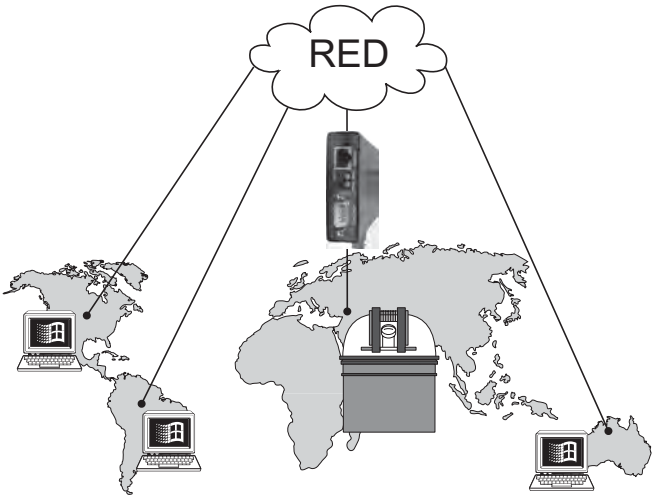


Redireccionamiento COM – El puerto COM „completamente en otro sitio“

Con la ayuda de los driver de redireccionamiento COM y de uno o varios Com-Server permiten los sistemas operativos basados en Microsoft Windows instalar puertos COM adicionales, que pueden estar en cualquier posición de la red.

Un ejemplo:
Un telescopio solar en el sur de Europa entrega sus datos de imágenes por una red TCP/IP a diferentes universidades en todo el mundo. Pero las coordenadas de posición del telescopio sólo se pueden introducir lamentablemente por un interfaz serie directamente en el lugar. Hasta ahora tenían que entregar telefónicamente estos parámetros a un trabajador, que introducía las configuraciones necesarias.

Desde que el puerto de configuración del telescopio solar se ha conectado a un Com-Server, pueden los usuarios de Sydney, Washington y Tegucigalpa con un COM3 virtual de su PC posicionar online el telescopio solar.

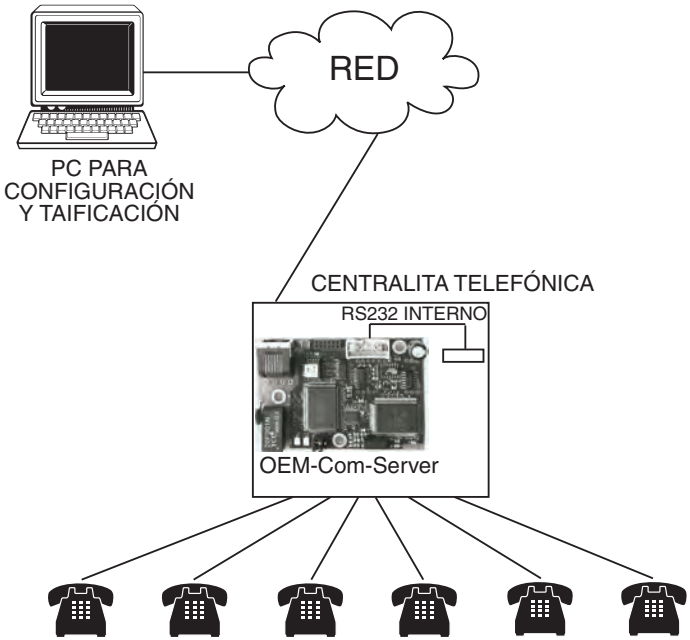


TCP/IP Sockets – Con el propio programa al puerto serie

A través de TCP o también UDP permite el Com-Server una comunicación directa con el puerto serie del Com-Server.

Un ejemplo:
Un fabricante de Centralitas telefónicas conecta el interfaz RS232 de configuración y tarificación a la placa integrada Com-Server OEM. Para poder configurar la centralita y procesar los datos de tarificación se programó un pequeño software,

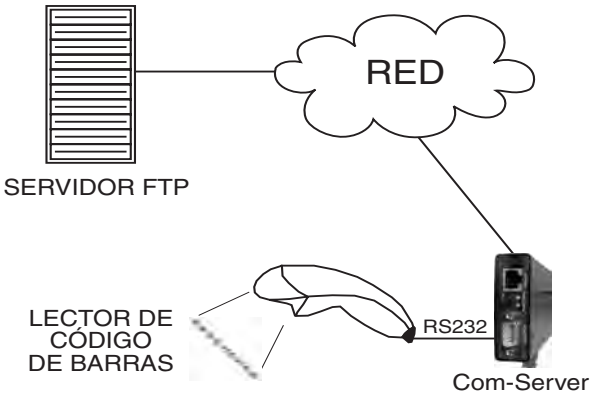
que soluciona estas tareas de una forma cómoda a través de la red.



FTP – Datos Serie directamente a un fichero

El Com-Server soporta entre otros también FTP, tanto cliente como servidor. Por ello se pueden transferir ficheros sin problemas desde o hacia el puerto serie del Com-Server.

Un ejemplo:
En un almacén de una empresa de transportes se tienen que procesar todos los paquetes entrantes y salientes con un código de barras. Para ello se conectaron en la entrada y salida de paquetes un lector de código de barras y un Com-Server, que se configuró como cliente FTP. Los códigos de barras leídos se almacenan ahora automáticamente por FTP en el fichero del File-Server de la empresa.



Por supuesto existen muchos ejemplos más de aplicaciones para el uso de los Com-Servers. CNC, DNC, Obtención de datos, procesado de medidas, administración remota... por nombrar algunos ejemplos.

Com-Server – Los diferentes Modelos

Com-Server Highspeed Industry - #58631

- Red: 10BaseT o 10/100BaseT autosensing.
- Protocolos TCP/IP:
  UDP/TCP-Sockets, FTP, Telnet tanto Server como Cliente, puertos virtuales
  COM con el redireccionamiento COM de Windows, modo Box-to-Box (RS232 túnel), protocolos de ayuda: ARP, RARP, DHCP/BOOTP, PING, RIP, SNMP, HTTP y Web-Based Management (en desarrollo), Inventarización, manejo en grupo.
- Puerto Serie: Conector DB9, configuración pines como PC incluído señales de modem e intercambiable a RS422, RS485, Velocidad 50-230,400 Baudios, formato de los datos 7, 8 Bit de datos; 1, 2 Stopbit; Paridad: none, even, odd; Handshake: Hardware, Xon/Xoff
- Alimentación: 12-24V AC/DC en clemas para funcionamiento industrial o 230V transformador para entornos ofimáticos.
- Pequeña carcasa para montaje en carril DIN 105x75x22 mm.

Com-Server Highspeed - #58031, 58034

- Red y Protocolos: igual que el Com-Server Highspeed Industry
- 1 o 4 puertos Serie: enchufes DB9 con RS232 configuración pines como PC, conmutables independientemente a RS422, RS485, posibilidad 20mA opcional, Velocidad 50-230,400 Baudios, formato de datos: 7,8 bit de datos; 1, 2 Stopbit; Paridad: none, even, odd; Handshake: Hardware, Xon/Xoff.
- Alimentación: fuente de alimentación 230V integrada.
- Carcasa de aluminio para sobremesa.

Placas OEM

- Red: 10BaseT o 10/100BaseT autosensing RJ45, posibilidad de clemas o tira de pinchos.
- Serie: Tira de pinchos con señales TTL RS232 o RS485, RS422 o posibiidad de 20mA.
- Alimentación: 3V, 5V o 24V.

- Diferentes formatos.

Más formatos ver http://www.wut.de

Web – IO – Ejemplos de conexiones desde la Praxis

Web-IO son unos pequeños sistemas, con los cuales se pueden controlar y vigilar señales analógicas y digitales por TCP/IP-Ethernet.

Un servidor HTTP integrado permite un completo manejo Web-Based Management. Configuraciones, Controles y Análisis son posibles de realizar sin software especial para cada equipo, incluso para usuarios nóveles, inmediatamente desde un navegador.

Además la integración en sistemas ya existentes de Control y Visualización no ofrece ningún problema. SNMP y OPC, pero también el acceso directo vía TCP y UDP ofrecen una integración completamente sencilla.

Los SNMP-Traps y el envío de Email, en infraestructuras ya existentes incluso SMS, permiten nuevos caminos del procesado de señales.

Por el lado de la red están dotados todos los Web-IO con una conexión 10/100BaseT autosensing. La alimentación se realiza en un amplio rango entre 12V y 24V con continua o alterna así como la posibilidad de los 230V de red.

Debido a esta flexibilidad, los Web-IO ofrecen todas sus funciones para su uso en las plantas industriales, mantenimiento así como laboratorio o aplicaciones ofimáticas.

Web-IO Termómetro – Vigilancia de la temperatura en la red

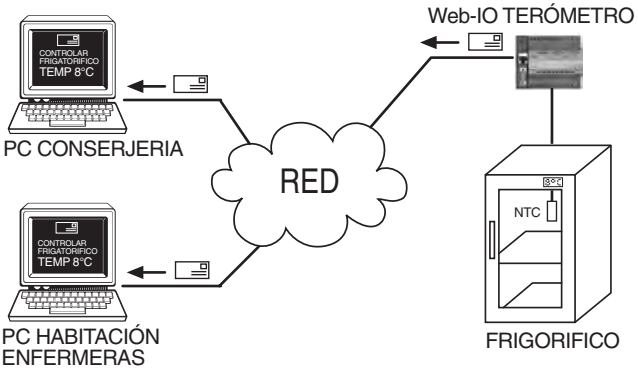
El Web-IO termómetro permite la conexión de hasta 8 sensores de temperatura del tipo NTC o PT100. Las temperaturas se pueden observar en cada momento desde el Browser (navegador); como tablas o en una página creada por si misma. Se pueden fijar valores límite individuales para cada sensor. Se pueden enviar alarmas vía E-Mail o SNMP-Trap provocadas por sobrepasar o descender los valores límite.

Un ejemplo:
En un hospital se tienen que almacenar medicamentos especiales en un frigorífico entre 3ºC y 8ºC.

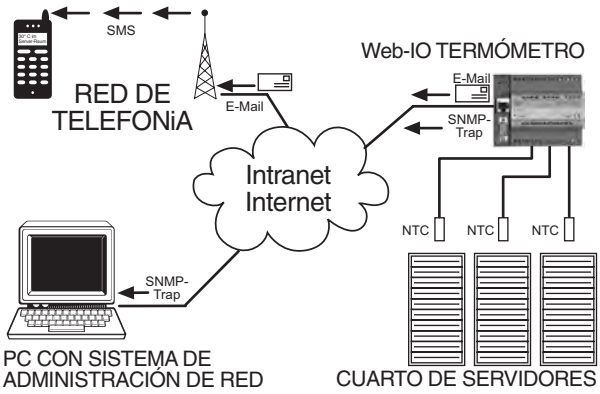
En el pasado, se controlaba la temperatura del frigorífico cada hora por la enfermera de planta y se anotaba en una tabla.

En la actualidad, el Web-IO termómetro vigila el frigorífico. Si la temperatura sube por encima de 7,5ºC recibe la enfermera de planta un email. Si la temperatura sobrepasa incluso los 8ºC se envía adicionalmente un email a la conserjería.

Adicionalmente se guarda una copia de seguridad cada semana del historial de la temperatura con un Download en formato de tabla Excel.



Un segundo ejemplo:
En un centro computacional ya han perecido por sobrecalentamiento varias veces diferentes discos duros, porque por la noche ha fallado el sistema de climatización en el cuarto de servidores. Actualmente el Web-IO termómetro en servicio envía, por email al operador de telefonía móvil, un SMS directo al técnico. Adicionalmente se envía un SNMP-Trap al sistema de administración de red. Así se puede reaccionar a tiempo a cualquier hora.



Web-IO 12xDigital

Por el navegador HTTP (Browser), TCP, UDP, SNMP u OPC se pueden controlar y analizar por la red 12 entradas y 12 salidas digitales. Además se ofrece también un modo Box-to-Box, con el cual una entrada en el Web-IO 1 puede influir en la salida del Web-IO 2.

Las entradas digitales están separadas galvánicamente en grupos de 4 y se permite su control con tensiones de hasta ±30V.

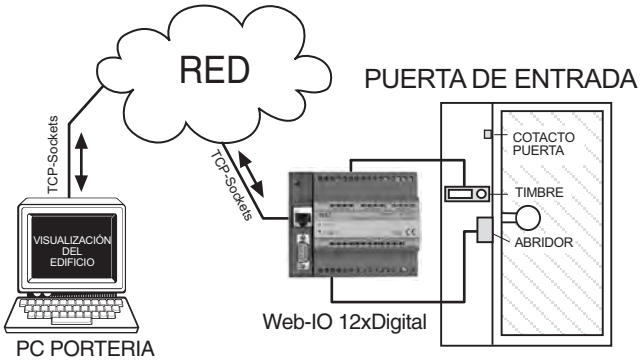
Las salidas conmutan entre un rango de fuentes de alimentación comunes de tensiones entre 6 y 30V. La corriente máxima de salida por cada señal es de 500mA. Una protección de sobrecarga térmica proporciona una seguridad contra cortocircuitos. Las salidas se pueden conectar en paralelo a pares o en grupos de 4 para conseguir corrientes mayores. Diodos libres para conexiones de relés están naturalmente integrados.

Se permiten hasta 12 configuraciones independientes de alarmas que proporcionan las muestras de entrada deseadas para vigilar y en caso de alarma se pueden enviar por e-mail, SNMP-Trap o UDP avisos para alertar.

Poniendo punto y final: Con el Web-IO 12xDigital se puede introducir en la red cualquier aparato que disponga de un contacto o una señal eléctrica.

Un ejemplo:
La puerta trasera de un edificio empresarial tiene un timbre, un abridor y un contacto, que vigila si la puerta está cerrada.

El timbre y el contacto de la puerta están conectados con el Web-IO en las entradas 0 y 1. La salida 0 del Web-IO controla el abridor de la puerta. Con TCP-Sockets se integran estas señales del Web-IO en el sistema de visualización del edificio. De esta forma recibe el portero en la entrada principal una señal acústica desde su PC, cuando alguien pulsa el timbre. El portero puede activar el abridor de la puerta mediante un clic del ratón. Si la puerta está cerrada o abierta también se muestra aquí.



Adicionalmente está conectado a la entrada 4 del Web-IO el contacto de la puerta. En el sótano, junto al sistema de alarma, se encuentra un segundo Web-IO. La salida 4 de este Web-IO controla una entrada en el sistema de alarma. Los Web-IO fueron concebidos de tal forma que la entrada 4 del primer Web-IO Box-to-Box con la salida 4 del segundo Web-IO trabajen juntas. El estado del contacto de la puerta está conectado 1:1 por la red de esta forma con el sistema de alarma. Si por la noche, cuando la alarma está conectada, se abre la puerta, salta inmediatamente la alarma.

W&T

www.WuT.de

W&T distribuido en México por Telsa Mayorista

Tel. 55 5740 2142, 55 5740 4360

ventas@telsa.com.mx

www.telsa.com.mx

telsa®

La Alternativa en Conectividad

Más de 35 años de experiencia en soluciones de TIC