

Internet intern

Inhalt:

-
- Ports und Sockets
- Proxies
- IP-Adressen und DNS
- Protokolle
- Ping

Ob Ports, Sockets, TCP, UDP oder Proxy, Fachchinesisch bietet auch das Internet reichlich. Hier steht, was hinter den wichtigsten Begriffen steckt, und wie die Geschichte überhaupt funktioniert.

Ein paar hundert Anwender greifen zugleich auf den Feierabend-Server zu und tun dabei ganz verschiedene Dinge. Das ist erstaunlich, denn der Server hängt nur über eine einzige Leitung am Internet. Wie bringt es diese Maschine fertig, jeden Anwender glauben zu lassen, der ganze Feierabend-Server gehöre ihm ganz alleine? Etwas präziser: Wieso können viele Anwender eine Leitung gleichzeitig nutzen, und wie unterscheidet der Server die einzelnen User?

Die erste Frage ist einfach zu beantworten. Der Datenstrom wird in kleine Portionen aufgeteilt, Pakete genannt, und diese werden in Abständen nacheinander verschickt. Innerhalb der Lücken zwischen zwei Paketen eines Anwenders werden die Pakete anderer Anwender übertragen. Gibt es also beispielsweise 10 Anwender, wird das erste Paket von Anwender 1 übertragen, dann folgen 9 andere und nun erst das zweite Paket von Anwender 1. Sind es 100 Teilnehmer, passieren zwischenzeitlich schon 99 fremde Pakete diese eine Leitung. Dass diese Leitung sehr schnell sein muss, ist klar.

Ports und Sockets

Es liegt also an der Geschwindigkeit (der Bandbreite) der Leitungen, die uns ausbremst und nicht (oder nur ganz wenig) am Feierabend-Server. Der muss das zweite Problem lösen, nämlich jeden User individuell zu bedienen. Dafür nutzt er zwei Dinge und zwar die Ports und die Sockets. Beginnen wir mit letzteren.

Der Server ist eine Multitasking-Maschine, er kann viele Tätigkeiten (Tasks)

parallel ausführen. Für jeden, der sich bei Feierabend einloggt, startet der Server einen neuen Task, und der legt einen so genannten Socket an. Ein Socket ist eine Software-Schnittstelle zwischen dem jeweiligen Task und der Außenwelt. Auch der Client (also der PC) hat einen Socket, praktisch kommunizieren die Sockets miteinander.

Ports

Wie schon gesagt, je ein Client und der Server kommunizieren über die Sockets miteinander. Jetzt gilt es nur noch, die richtigen Sockets miteinander zu verbinden, und das ist Sache der Ports. Ein Port ist eine 16-Bit-Zahl im Bereich von 0 bis 65.535. Die Ports mit den Nummern 0 - 1023 sind für bestimmte Dienste bzw. Protokolle reserviert. Die wichtigsten davon:

Port Dienst

21 FTP (Datei-Übertragung)

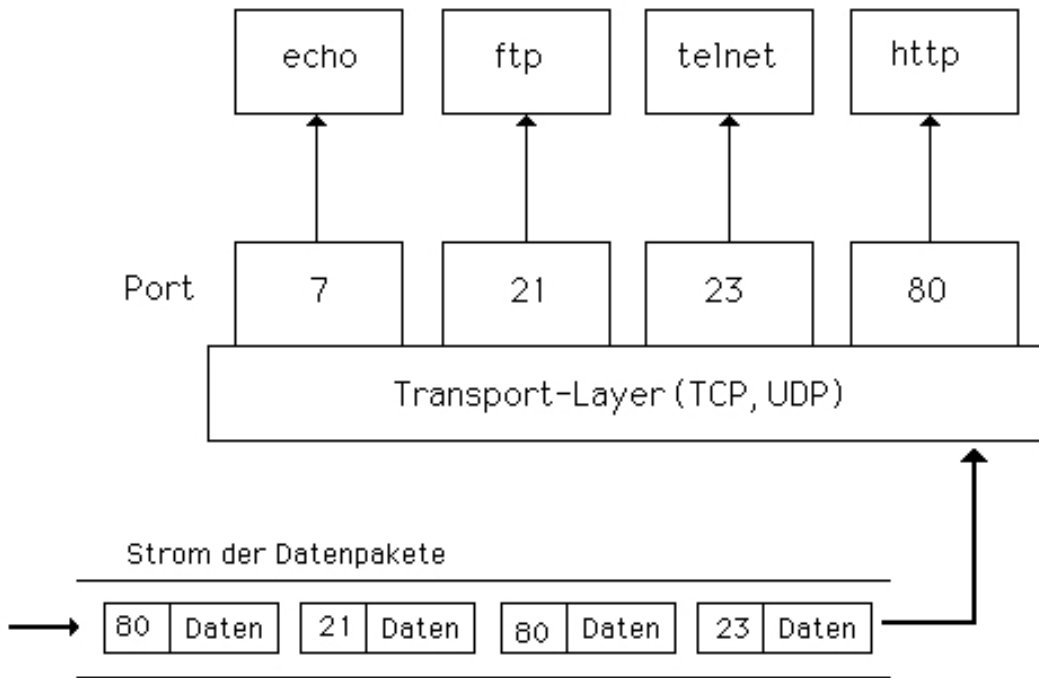
25 SMTP (Postversand)

80 HTTP (Web)

110 POP (Email-Abholung)

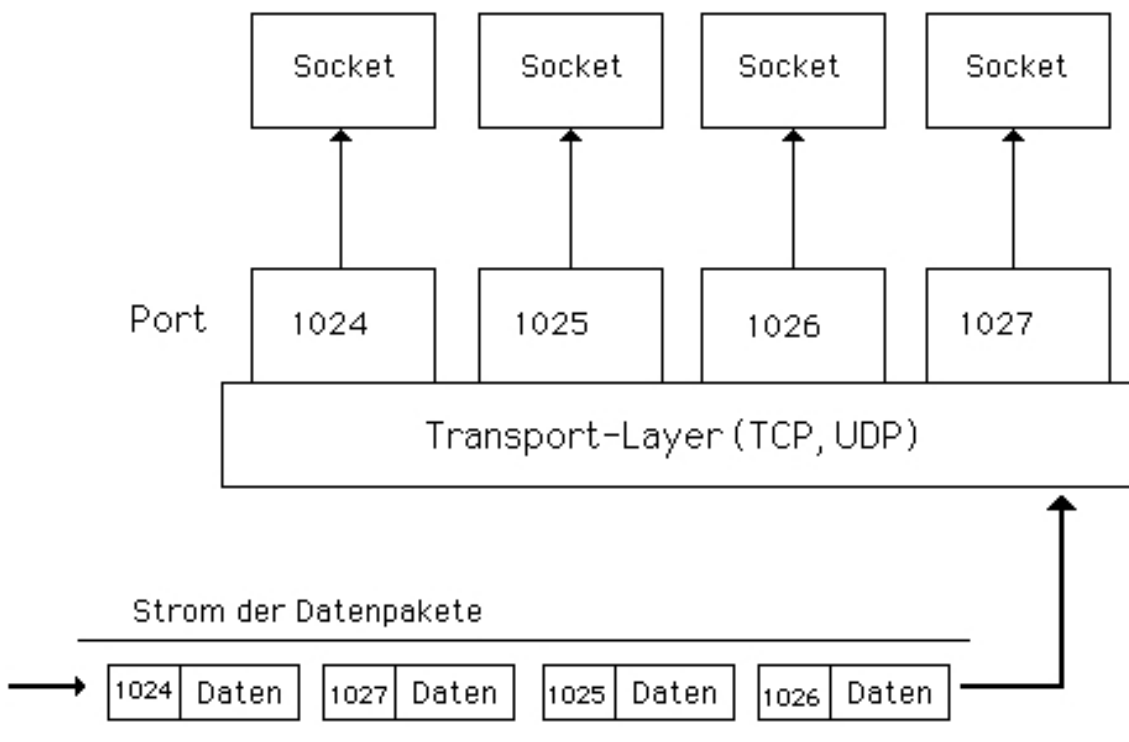
Wenn Sie einen Server ansprechen, wählen Sie dessen IP-Adresse (siehe unten) sowie den passenden Port an, fragt sich nur wie? Ganz einfach: Sie senden im Datenpaket neben der IP-Adresse auch die Portnummer.

Die folgende Abbildung zeigt aus Sicht des Servers, was dabei passiert. Daraufhin wird eine Verbindung aufgebaut und dieser eine neue, so genannte lokale Portnummer (ab 1024) zugeteilt. Das ist schon deshalb wichtig, weil beispielsweise der HTTP-Port (80) sofort wieder freigegeben werden muss, um den nächsten Anwender erwarten zu können.



Um eine bestimmte Anwendung, wie HTTP, zu adressieren, müssen Datenpaket die passende Portnummer gesendet werden.

Der neue Port hingegen wird mit einem Socket verbunden. Damit besteht jetzt eine eindeutige Beziehung zwischen dem Client (unserem PC) und dem Server. Die folgende Abbildung verdeutlicht diese Aktion.



Nachdem die erste Verbindung über den Applikationsport hergestellt wurde, werden Verbindungen mit anderen Portnummern aufgebaut. Damit lassen sich die einzelnen User auseinanderhalten.

Proxies

Wenn Sie Ihren Provider anwählen, erwartet dessen Server bzw. sein Proxy-Server Ihren Anruf auch auf einem bestimmten Port, jetzt aber (meistens) für alle Dienste (HTTP, FTP, Gopher u.s.w.) immer auf demselben, zum Beispiel auf Port 8080.

Wenn der Provider mit einem Proxy-Server arbeitet, müssen Sie dessen Namen und seinen Port im zugehörigem Browser-Dialog eintragen. Prinzipiell läuft jetzt die gleiche Aktion ab, d.h. es gibt eine eindeutige Verbindung zwischen Ihnen und dem Proxy mittels einer lokalen Portnummer und außerdem die oben geschilderte Verbindung zwischen dem Proxy und dem (beispielsweise) Feierabend-Server. Oder auch nicht, denn Proxy heißt (auch) Stellvertreter. In dieser Funktion spiegelt (kopiert) der Proxy diverse Sites (manche hält er auch in seinem Cache), um dann diese anzubieten, wenn der Anwender sie anfordert.

Wenn der Proxy gut ist und ständig spiegelt, ist dagegen nichts einzuwenden, zumal, wenn er nach einem Klick auf "Aktualisieren", dann wirklich verbindet. Es gibt jedoch billige Internet-Zugänge, deren Proxies einfach die wichtigsten Server spiegeln und die User überhaupt nicht in die freie Internet-Welt verbinden. Das spart teure breitbandige Standleitungen, doch hierfür mit Sprüchen wie "Internet für 1,99" zu werben, grenzt schon an Betrug.

Ein Proxy kann aber auch selektiv verhindern, dass bestimmte Seiten angewählt werden können, was zu zwei Ansätzen führt. Im Intranet (einem firmeninternen Internet) meldet sich der Proxy, wenn ein Anwender in das externe Internet will, und verlangt eine User-ID und ein Passwort, jedenfalls dann, wenn nicht jeder Mitarbeiter diese Rechte haben soll. Das geht ja noch. Schon schwieriger ist es für die Provider, "unanständige" Server zu sperren, um dem Gesetz zu genügen.

Ein kleiner Tip zum Thema Proxy: Wenn die Verbindung nicht so schnell ist, können Sie in der Statuszeile des Browsers ablesen, wie bei der Anwahl einer neuen Site immer abwechselnd der Zielsever und Ihr Proxy kontaktiert werden. Wenn Sie dabei so etwas wie "proxy-netxy.de:8080 contacted, waiting for reply" für längere Zeit sichtbar ist, sollten Sie Ihrem ISP auf die Füße treten (oder ihn wechseln).

IP-Adressen und DNS

Das Problem der Provider: Adressiert wird immer mittels IP-Adressen (IP = Internet Protocol). Die IP-Adresse besteht aus vier dreistelligen Zahlengruppen, z.B. 192.168.122.133. Jede Zahlengruppe kann Werte zwischen 000 und 255 einnehmen. Daraus ergibt sich, dass maximal rund vier Milliarden PCs gleichzeitig mit dem Internet verbunden sein können. Jeder Teilnehmer im Internet, Server inklusive, hat eine einmalige IP-

Adresse. Der Provider kann nur eine IP-Adresse sperren, aber nicht verhindern, dass jemand eine Suchmaschine startet, um einen der vielen Server zu finden, welche die unanständigen Seiten gespiegelt haben. Sie können anstatt einer URL (Uniform Resource Locator), wie "http://www.google.de" durchaus die IP-Adresse (66.249.85.104) in das Adreßfeld des Browsers eintragen oder für Feierabend 81.3.61.2. Tun Sie das nicht, dauert es etwas länger. In diesem Fall verbindet sich nämlich Ihr Browser zuerst mit einem DNS-Server (Domain Name Server), um dort nachzulesen, welche IP-Adresse "http://www.google.com" hat. Damit Ihr Browser weiß, wo er suchen muss, müssen Sie im Rahmen der Internet-Konfiguration zwei DNS-Adressen eintragen. Die erste (primäre) Adresse wird im Normalfall genutzt, die zweite, wenn der erste Server nicht oder zu spät antwortet. Die DNS können aber auch von einem Router automatisch bezogen werden.

Protokolle

Bisher haben wir schlicht unterstellt, dass die Rechner im Netzwerk irgendwie über Ports und Sockets miteinander kommunizieren. Doch so, wie sich Amerikaner und Chinesen zuerst auf eine gemeinsame Sprache einigen müssen, bevor Sie miteinander reden können, geht es auch den Computern. Der feine Unterschied: Hier heißt die Sprache "Protokoll", und so wie im wahren Leben gibt es davon viele.

Allgemein bekannt sind die Anwenderprotokolle, wie HTTP, FTP oder Telnet, doch auch in diesen Sprachen können sich zwei Computer nur dann unterhalten, wenn sie über irgendein Medium miteinander verbunden sind. Genauer: Die Daten müssen irgendwie von einem Computer zum anderen transportiert werden. Das muss nicht über das Netz geschehen. Man kann HTML-Seiten auch auf Disketten oder CDs speichern, sie per Briefpost versenden und auf den Zielrechnern abspielen. Wesentlich ist auf dieser Ebene nur, dass beide Computer dasselbe HTML sprechen.

Die zweite Erkenntnis: Es ist sinnvoll, die Anwender- und die Transport-Ebene voneinander zu trennen, denn sicherlich kommunizieren Sie auf eine andere Art mit Ihrem Briefträger, als Ihr Computer mit einem Server. Genau deshalb gibt es auch ein Transportprotokoll für den Fall, dass zwei Computer direkt miteinander verbunden sind und zwar auf der Ebene "Versand von Datenpaketen". Wie diese Datenpakete ihren Weg über das Netzwerk finden, ist Sache noch eines Protokolls, und in welcher Art und Weise die Bits und Bytes durch die Leitungen flitzen, wird durch ein weiteres Protokoll geregelt.

Im Prinzip läuft die Sache so, dass immer ein höher angesiedeltes Protokoll Funktionen im darunter liegenden Protokoll aufruft, und zwar ohne wissen zu müssen, wie dieses funktioniert. Die Protokolle liegen sozusagen

übereinander. Deshalb spricht man auch von einem Stack (Stapel), hier von einem TCP/IP-Stack.

Anwendungen http, ftp, Telnet
Transport TCP, UDP, ...
Netzwerk IP, ICMP, ...
Verbindung Gerätetreiber

Die Protokoll-Stacks des Internets sind in 4 Schichten aufgebaut, und zwar so, dass ein Protokoll Funktionen im darunter liegenden Protokoll aufrufen kann.

Die Abbildung links zeigt den Zusammenhang. Das heißt aber nicht, dass die "unteren" Protokolle die oberen benötigen. Man stelle sich deshalb besser das System so vor, dass die Verbindung zwischen den beiden Rechnern auf allen Schichten parallel laufen.

TCP und UDP

TCP und UDP sind die beiden 4-Schichten-Protokolle, mit deren Hilfe im Internet kommuniziert wird. Wenn zwei Anwendungen zuverlässig miteinander kommunizieren wollen, bauen sie eine TCP-Verbindung auf. TCP heißt "Transport Control Protocol", und zwar im Sinne von kontrolliertem Transport. TCP baut eine feste Verbindung auf, ähnlich wie bei Telefongesprächen. Dabei garantiert TCP, dass die Daten zur Gegenstelle gelangen und zwar in der Reihenfolge, in der sie gesendet wurden. Wenn

nicht, meldet TCP einen Fehler. HTTP oder FTP wären ohne TCP nicht möglich.

Im Gegensatz zum verbindungsorientierten TCP steht das verbindungslose UDP (User Datagram Protocol). UDP sendet einfach seine Datagramme (Datenpakete). Nichts garantiert, dass die Pakete ankommen, und wenn, dann muss die Reihenfolge der Pakete nicht stimmen. Fehlermeldungen unter UDP gibt es nicht. Da fragt mancher nach dem Nutzen dieses Protokolls, aber den muss es wohl geben.

Ein typisches Beispiel ist ein Uhr-Server. Dieser sendet ein Paket der Art "es ist jetzt 13:17:39". Wenn nun in typischer TCP-Systematik eine Rückmeldung der Art "Paket fehlerhaft, sende es

Ping

Sogar ausgesprochen störend wäre TCP für das Kommando (praktisch ein DOS-Programm) namens Ping. Damit lässt sich zuerst feststellen, ob ein Server überhaupt am Netz ist. Dafür sendet Ping Pakete vom Typ "Echo Request" (0x0) an den Zielrechner. Ist dieser in Betrieb, genauer, arbeitet sein IP-Stack, antwortet er mit "Echo Reply" (0x8). Dabei wird die Zeit gemessen, um das Tempo der Verbindung zu testen.

Die folgende Abbildung zeigt, wie auf diese Art ein Server über das Internet "angepingt" wurde.

```
PING 10.0.1.1 (10.0.1.1): 56 data bytes
64 bytes from 10.0.1.1: icmp_seq=0 ttl=255 time=2.818 ms
64 bytes from 10.0.1.1: icmp_seq=1 ttl=255 time=1.457 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=255 time=1.448 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=255 time=1.937 ms
64 bytes from 10.0.1.1: icmp_seq=4 ttl=255 time=2.083 ms

--- 10.0.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.448/1.949/2.818/0.503 ms
```

Bei einer solchen Zeitmessung würde der Overhead von TCP nur stören. Doch Ping wird auch eingesetzt, um die Qualität einer Verbindung zu testen. Dazu werden mehrere Pakete gesendet, um dann zu messen, ob Pakete verloren gingen oder vertauscht wurden. Hierfür wäre nun TCP überhaupt nicht mehr zu gebrauchen.

Ping arbeitet auf der Netzwerk-Ebene (siehe Abbildung 3) und sendet ICMP-Pakete (Internet Control Message Protocol), arbeitet also recht systemnah. Der IP-Stack des Servers gehört praktisch zum Betriebssystem, kurz und schlecht, die Sache ist gefährlich. Tatsächlich hat Ping in der Windows-Welt schon traurige Berühmtheit erlangt. Dort wurde mit übergroßen ICMP-Paketen (anstatt der üblichen 32 Byte) der TCP/IP-Stack von Windows so durcheinander gebracht, dass der "Ping of Death" eintrat, sprich, der Server abstürzte. Theoretisch ist es sogar möglich, in den ICMP-Paketen schädliche Befehle unterzubringen.

Die meisten Firewall-Rechner lassen "Pings" passieren, schließlich ist Ping für viele Netzverwalter ein unverzichtbares Tool. Als Privatmann/frau kann man aber bei einigen Routern die Antwort auf Ping auch abschalten. Bleibt das bekannte Fazit: Wer seinen Rechner mit der Außenwelt verbindet, geht immer ein Risiko ein.

Autor

Peter ([WoSoft](#))