TCP/IP Grundlagen
verfasst von wintools4free.dl.am
visiT:
www.tgss.dl.am
www.wintools4free.dl.am

Das Internet ist ein Heute weit verbreitetes Medium, das auf eine große Resonanz stößt. War das Internet vor einigen Jahren nur großen Firmen und staatlichen Instituten vorbehalten, so gehört es Heute fast zum alltäglichen Leben dazu. Die Möglichkeiten der Nutzung sind hierbei sehr vielfälltig und werden auf unterschiedlichste Art und Weise genutzt. Sei es um Informationen und Dateien auszutauschen, zu chatten oder sich selbst mit einer eigenen Homepage zu presentieren. Im folgenden Text erfahren Sie, was genau passiert wenn Sie beispielsweise einen Link anklicken. Der unerfahrene Benutzer würde sagen: „ Es öffnet sich eine Web-Seite“. Dies stimmt zwar, ist aber nur sehr oberflächlich beschrieben. Bis die angeforderte Seite letztendlich auf Ihrem Bildschirm erscheint werden einige Prozeduren durchlaufen. Zuständig für die Übertragung der Daten ist die sogenannte TCP/IP Protokollfamilie, die sich wiederrum aus verschiedenen mehr oder weniger sicheren Protokollen zusammensetzt. Diese „Familie“ wird auch TCP/IP Stack genannt. Was soviel wie Stapel bedeutet. Dieser Stapel ist in vier Schichten gegliedert, die gleich etwas genauer erläutert werden sollen. Werden nun Daten zu einem entfernten System geschickt, so müssen diese jede dieser Schichten durchlaufen. Auf Seite des Absenders, wie auch auf der Seite des Empfängers.

Erläuterung der Schichten:

➔ Anwendungsschicht

In dieser Schicht befinden sich die vom Anwender genutzten Dienste. Beispielsweise das FTP (File Transfer Protocol) Protokoll, das wie der Name sagt für das Übertragen von Dateien zuständig ist. Oder das HTTP (Hyper Text Transfer Protocol) Protokoll, das für die Übertragung von Webseiten zuständig ist. Hierzu gehören auch andere Protokolle wie z.B. Telnet oder SMTP (Smart Mail Transfer Protocol), die hier jetzt aber an dieser Stelle nicht näher erklärt werden. (Später vielleicht noch!)

➔ Transportschicht

Diese Schicht setzt sich aus 2 Protokollen zusammen. Nämlich dem TCP (Transmission Control Protocol) und dem UDP (User Datagram Protocol) Während das TCP Protokoll eine Aufteilung der Daten vornimmt, um sicherzustellen, dass die Daten auch ihr Ziel erreichen arbeitet das UDP Protokoll „verbindungslos“. Das heißt, dass die Daten verschickt werden, ohne daruf zu achten, ob die Daten tatsächlich Ihr Ziel erreichen. Dies hat natürlich mehrere Nachteile.

➔ Internetschicht

Hier ist eigentlich das IP (Internet Protocol) gemeint, das die von TCP oder UDP verpackten Daten an Ihrem Bestimmungsort überträgt. Darin enthalten ist noch das ICMP Protokoll, das für die Übermittlung von Fehlermeldungen dient, die wiederum über das IP Protokoll ihr Ziel erreichen. Die Datenkontrolle übernimmt das TCP Protokoll. Womit sichergestellt wird, dass die Fehlermeldungen auch wirklich ankommen.

➔ Netzwerkschicht

In dieser Schicht tauchen je nach Verbindung eine Menge andere Protokolle auf. Das Ethernet und eine serielle Verbindung zu einem Modem sind zwar völlig verschiedene Arten von Verbindungen, erfüllen aber beide die von der Internetschicht erfordeten Anforderungen und sind somit TCP/IP fähig.

Die Anwendungsschicht:
Wie oben zu lesen ist, besteht die Anwendungschicht aus einer Vielzahl von Protokollen, die für bestimmte Dienste zuständig sind. Dabei kann die Anzahl der Protokolle schwanken, da neue Dienste hinzukommen und andere wiederum mit der Zeit aussterben (wie z.B. Telnet, das wohl mit der Zeit vom sicheren SSH abgelöst wird, da es unverschlüsselt übertragen wird), weil sie z.B. die Anforderungen nicht mehr gerecht werden können. Diese Protokolle laufen meist auf bestimmten so genannten Well Known Ports. Den Standard Ports. (Dies kann aber vom Dienstanbieter geändert werden und ist kein muss.) So findet man das FTP Protokoll meist auf dem Port 20 und 21. Wobei hier der Port 21 für die Kommunikation zwischen den Rechnern und der Port 20 für die Datenübertragung dient.

Die Transportschicht:
Die Transportschicht bekommt die Daten von der Anwendungschicht. Diese Schicht ist für das Übertragen von einem zum anderen Host zuständig. Hier finden sich die beiden Protokolle UDP und TCP. TCP dient einer gesicherten Übertragung, während UDP keine Rücksicht auf verlorene Pakete nimmt. Sollte unter dem TCP Protokoll mal ein Datenpaket in der Unendlichkeit des Netztes verloren gehen, wird das jeweilige Paket einfach noch mal verschickt. Mehr dazu gleich.

Internetschicht:
Dies ist die dritte Stufe, die ausgehende Daten passieren müssen. In der übergeordneten Schicht (Transportschicht) werden Verbindungen zwischen den Rechnern aufgenommen. Die Internetschicht hält das Gesammtgerüst sozusagen zusammen.

Netzwerkschicht:
Diese Schicht bildet sozusagen das Fundament, indem die Protokolle der Internetschicht arbeiten können.

TCP (Transmission Control Protocol)

Das wohl wichtigste Protokoll der Transportschicht ist das TCP Protokoll, das ein verbindungsorientiertes Protokoll ist. Verbindungen bestehen zwischen 2 Teilnehmern, wobei einer der Client eine Verbindung anfordert, der Server diese bewilligt und der Client diese dann bestätigt. Diese Art der Verbindung nennt man auch Three Way Handshake. Auf die selbe Art und Weise wird diese Verbindung auch wieder getrennt. Man kann sich das ganze auch als Gespräch vorstellen. Der Client sagt: „So, ich bin weg.“ worauf der Server antwortet: „Ja alles klar. Man sieht sich.“ Das letzte Wort hat dann der Client: „Bis dann.“ Das TCP Protokoll ist im Vergleich zum UDP Protokoll sehr zuverlässig. Nachdem ein Datenpaket versendet wurde wartet das Protokoll auf eine Nachricht, ob das Paket tatsächlich angekommen ist oder nicht. Sollte dies nicht der Fall sein, so wird das jeweilige Paket nochmal versendet. Die Datenpakete werden in kleine Pakete aufgeteilt, und mit einigen Informationen versehen, dem sogenannten Header. Dieser Header ist im ersten Teil des Datenstroms enthalten und 20 Byte groß. Die gesammte Größe der Pakete kann unterschiedlich

sein, darf aber 64 KB nicht überschreiten. Im Header finden sich neben der Portinformation der übergeordneten Anwendungschicht auch noch weitere Daten und der eigentliche Inhalt der Paketes. Schauen wir uns die im Header enthaltenden Informationen mal genauer an:

- Quell/Zielport: (Source Port/Destination Port)
  Hier sind die Ports gemeint, die von den Diensten der Anwendungsschicht genutzt werden.

- Sequenznummer: (Sequence Number)
  Diese Nummer gibt die Position des Segments innerhalb des Datenstroms an. Mit dieser Information kann der Empfänger die Daten wieder an der richtigen Stelle zusammensetzen.

- Bestätigungsnummer: (Acknowledgment Number)
  Der Empfänger bestätigt dem Sender den Erhalt der bereits verschickten Daten, bzw. gibt die Nummer als nächsten erwarteten Datenpakets an.

- Offset:
  In diesem Teil wird die genaue Größe des Headers angegeben.

- Fenstergröße: (Window)
  Da die Übertragung nicht wirklich Byte für Byte gschieht, vereinbaren beide Seiten ein Intervall oberhalb der vom Empfänger bestätigten Daten, indessen Bereich der Sender keine Bestätigung über den Erhalt benötigt um dennoch Daten zu verschicken.Wird der Erhalt der Daten nun vom Empfänger bestätigt, wird der Fensterausschnitt verschoben.

- Prüfsumme: (Checksum)
  Dieses Feld erhält eine Prüfsumme des Paketinhaltes, die über einen sogenannten Cycle Redundancy Check (CRC) erzeugt wird.

- Dringlichkeitsanzeiger: (Urgency Pointer)
  Hier wird auf ein anderes Paket verwiesen, dessen Übertragungspriorität erhöht wird.

Neben den sogenannten Flags gibt es noch weitere „Markierungen“ innerhalb des TCP Headers, wovon jede ein Bit groß ist.

- URG: (Urgent/Dringend)
  Ist dieses feld markiert, wird der Dringlichkeitsanzeiger ausgelesen.

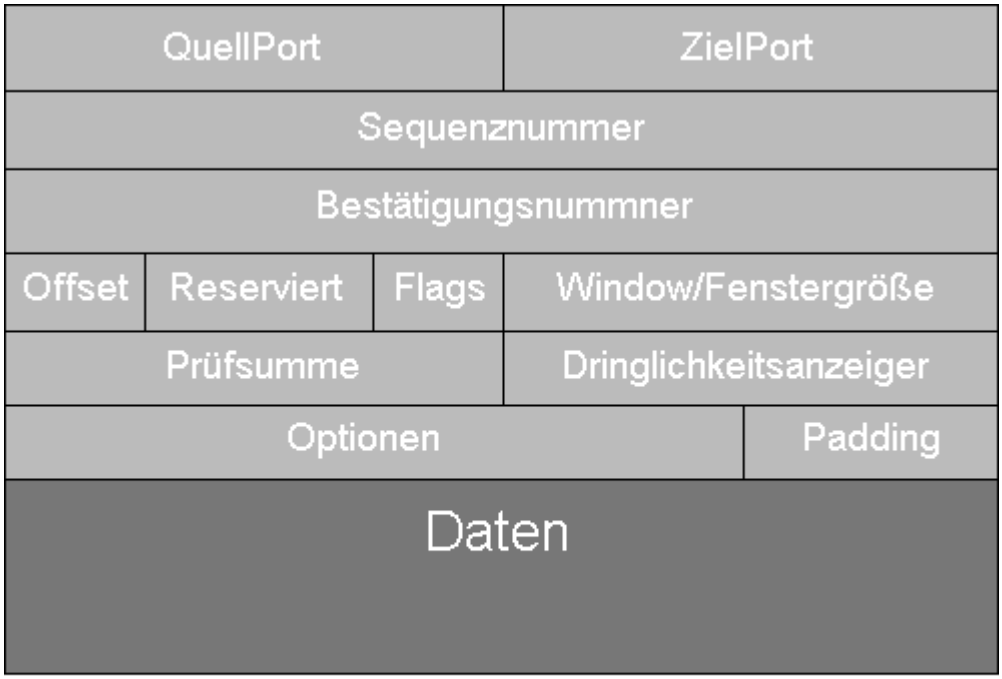
- ACK: (Acknowledgment/Bestätigung)
  Ist dieses Feld markiert, wird die Bestätigungsnummer ausgelesen. Ansonsten nicht.

- PSH: (Push/Durchreichung)
  Bei Aktivierung dieser Option wird das Segment auf Empfängerseite sofort ohne Zwischenpufferung an die Anwendungschicht übergeben.

- RST: (Reset/Zurücksetzen)
  Bei Übertragungsfehlern wird die Verbindung mittels dieses Feldes zurückgesetzt.

- SYN: (Synchronize Sequence Number/ Synchronisations- Sequenznummer)
  Bei Anforderung einer Verbindung ist dieses Bit gesetzt. Die Bestätigung erfolgt dann mit einem ACK-Bit.

- FIN: (Finish/Ende)
  Das Gegenstück zum SYN-Bit, das bei Beendigung einer Verbindung zum Einsatz kommt.



(Der TCP-Header)

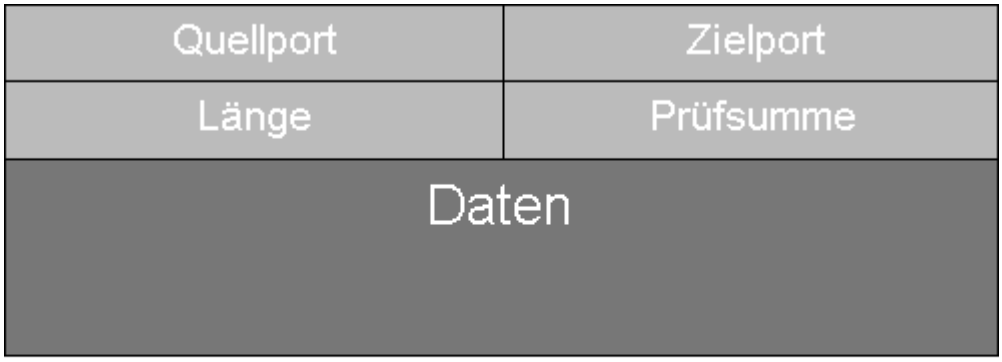
UDP (User Datagram Protocol)

Eine UDP Übertragung ist eine eher unzuverlässige Übertragung. Es findet kein Handshake oder sonst eine Absprache zwischen den Rechnern statt. Die Daten werden auch nicht wie beim TCP Protokoll in kleinere Pakete aufgeteilt. Die Erzeugung einer Prüfsumme ist auch nicht zwingend nötig, kann aber mit angegeben werden. Für eine reibungslose Übertragung sind die im Stapel tiefer angelegten Protokolle verantwortlich. Auch der Header fällt deutlich kleiner aus und liegt bei 8 Byte. Dadurch, dass einige Informationen mit denen TCP zu kämpfen hat wegfallen, eignet sich UDP besser für die Übertragung von Videosequenzen oder Onlinespielen. Also für die Echtzeitübertragung. Wie beispielweise das bekannte Kommunikationsprogramm Skype.

- Quell/Zielport: (Source Port/Destination Port)
  Diese Felder erfüllen die gleiche Funktion wie bei TCP. Auch UDP kommuniziert über die Ports, die in der Anwendungsschicht angegeben wurden, wobei UDP Ports nicht genau denen von TCP entsprechen müssen.

- Länge: (Lenght)
  Dieses Feld gibt die genaue Größe des gesamten Paketes an. Auch UDP Datagramme werden wie TCP Segmente nicht größer als 64 Kilobyte.

- Prüfsumme: (Checksum)
  Dieses Feld erhält eine Prüfsumme, die aber nicht mit angegeben werden muss. Bleibt dieses Feld leer, wird es vom Empfänger ignoriert.



(Der UDP-Header)

IP (Internet Protocol)

Das Internet Protokoll ist für die Adressierung sowie der Übertragung zwischen den Rechnern zuständig. Rechner, die sich im Internet befinden sind nicht mit allen anderen verbunden, sondern die Daten werden von einem zum anderen weitergereicht: (geroutet). Das Internet besteht eigentlich aus vielen kleineren Netzen, die alle über ein großes Netz miteinander verbunden sind. Dieses große Netz wird von Hochleistungsleitungen getragen, den sogenannten Backbones. Schnittstellen sind hierbei die Gateways. Das sind Rechner die an den kleinen Netzen als auch an den Backbones angeschlossen sind. IP hat hier jetzt die Aufgabe den jeweils kürzesten Weg für die Daten zu wählen, um an das Ziel zu gelangen. Wie beim UDP Protokoll gibt es auch hier keine direkte Verbindung zwischen den Rechnern. Datenpakete, die auf diese Weise versendet werden nennt man auch Datagramme. Diese Datagramme können nochmals in kleinere Pakete fragmentiert werden. Ein solches Datagramm setzt sich aus einem 20 Byte großem Header und den eigentlichen Daten

zusammen.Sämmtliche Zustellinformationen wie beispielsweise die IP Adressen vom Absender und Empfänger sind im Header enthalten. Wenn ein solches Datagramm fragmentiert werden muss, erhält jedes Fragment eine modifizierte Kopie des Headers. Auch hier schauen wir und den Header und die darin enthaltenen Informationen mal genauer an:

- Version: (Version)
  Hier wird die Protokollversion angegeben. Host´s, die mehrere Versionen dieses Protokolls unterstüzen können die Information hier auslesen.

- Länge: (Lenght)
  In diesem Feld wird die Größe des IP Headers angegeben.

- Diensttyp: (Type of Service)
  Dieses Feld enthält Kriterien, anhand derer das Internet Protokoll mit verschiedenen Daten verschieden verfahren kann. Die ersten Bits des Feldes geben eine Priorität für die nachfolgenden Bits, welche die bevorzugte Behandlung des Paketes angebeben.

- Gesamtlänge: (Total Lenght)
  Hier wird die gesammt Länge des IP Datagramms angegeben. Auch IP- Pakete sind auf 64KB beschränkt.

- Identifikation: (Identification)
  Wird ein IP Datagramm auf seinem Weg geteilt, gibt dieses Feld die ursprüngliche Zugehörigkeit eines solchen Teiles an.

- Markierungen: (Flags)
  In diesem Feld kann ein Datagramm so markiert werden, dass entweder durch das DF Bit (Dont Fragment) eine Fragmentierung auf jeden Fall verhindert wird, oder, dass der Empfänger durch ein gesetztes MF Bit (More Fragments) weitere Fragmente des Datagramms erhalten soll.

- Fragment Offset: (Fragment Versatz)
  Dieses Feld gibt die Position eines Fragmentes innerhalb des Datagramms an.

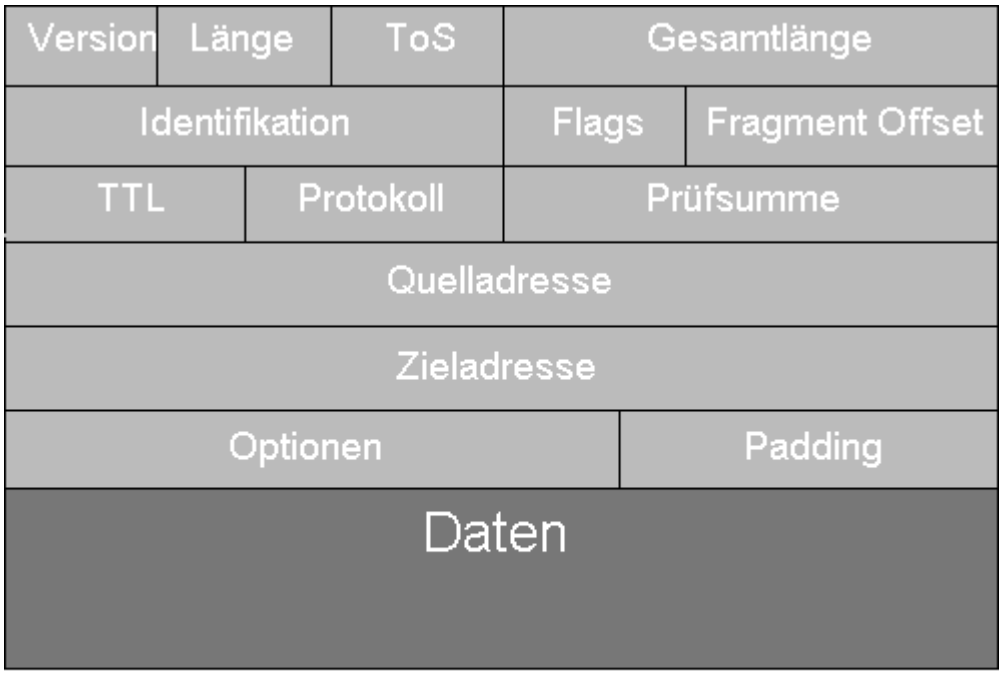
- Lebensdauer: (Time To Live)
  Hier ist ein Zähler enthalten, der bei jedem Passieren eines Netzknotens, eines sogenannten Nodes, mindestens um den Wert 1 verringert wird.Wenn der Zähler bei 0 angekommen ist, wird dieses Paket verworfen.

- Transportprotokoll-Adresse: (Protocol)
  Dieses Feld enthält eine Kennung, mit der das entsprechende Protokoll der übergeordneten

Transportschicht angesprochen wird, wie z.B. TCP oder das UDP Protokoll.

- Headerprüfsumme: (Checksum)
  Wie schon erwähnt handelt es sich bei m Interet Protokoll um ein unzuverlässiges Protokoll. Die Prüfung der übermittelten Daten obliegt das TCP Protokoll in der Transportschicht, an welche die Daten weitergeleitet werden. Die Prüfsumme für den Inahlt des Headers.
- Herkunftsadresse: (Source Address)
  Die IP Adresse des Absenders findet man in diesem Feld.

- Zieladresse: (Destination Address)
  In diesem Feld befindet sich die Adresse des Empfängers.



(Der IP-Header)

B2T: (Back 2 Topic)

Kommen wir nun zum obigen Beispiel zurück, bei dem der Link angeklickt wurde. Was genau passiert jetzt, und vor allem wann? Als erstes werden alle Schichten des Stapels durchlaufen. In der Anwendungsschicht wird nun festgestellt, dass es sich um den Aufruf einer Webseite handelt. Folglich handelt es sich um das HTTP Protokoll, auf dem Well Known Port 80. Diese Informationen werden nun an die Transportschicht übergeben. Hier werden die Daten in TCP Pakete „verpackt“ und an die Internetschicht weitergereicht. Die Pakete werden adressiert und an das LAN (Local Area Network) übergeben. Der nächste Halt wäre der Router. Dieser sendet die Datenpakete in das Internet und somit an den in dem Fall Web-Server. Dieser öffnet die Pakete, liest die angekommenen Daten aus und sendet dann die angeforderte URL an den Client. Wobei das auch noch recht oberflächlich ist. Proxy´s, Firewall´s und Switches wurden hier der Einfachheit wegen ignoriert. Dies soll natürlich nur ein kleiner Einblick in die Welt des Internets sein. Ich hoffe in dem ein oder anderen das Interesse für die Materie geweckt zu haben. Lernt man die

Zusammenhänge der Protokolle zu verstehen, kann man sich und seinen Rechner besser schützen. Bekannte Angriffsmethoden, wie DoS (Denial of Service) können besser verstanden werden. (Bei einem solchen Angriff werden eine Unzahl an SYN Anfragen gesendet, die der Server zu beantworten versucht. Dadurch kann er andere Anfragen nicht mehr bearbeiten und gibt seine Arbeit auf) Zu guter letzt sehen wir uns noch einige Protokolle der Anwendungschicht genauer an:

DNS (Domain Name System)

Dies ist das Protokoll, dass wahrscheinlich einen großen Beitrag zur Beliebtheit des Internet´s beigetragen hat. Dem Domain Name System haben wird es zu verdanken, dass wir uns Webadressen wie z.B. 213.156.321.122 nicht merken müssen. Stattdessen nutzt man Adressen wie: www.google.de, die sogenannten Domains. In diesem System wird zunächst die aufgerufene Seite vom Client bei einem Name Server erfragt. Dieser gibt dann die übersetzte IP- Adresse zurück. Wie bei IP Adressen besteht eine gewisse Hierarchie, die sich auch in der Schreibweise bemerkbar macht. Die einzelnen Teile werden durch Punkte getrennt. Das Domain Name System bildet ein verteiltes hierarchisches System. Es gibt kein Gesamtverzeichnis der aufgelösten Webadressen. Stattdessen gibt es in jedem Netz ein oder mehrere Name Server, die die Namen aller Rechner aus dem eigenen sowie dem eines übergeordneten Name Servers kennen. Um all die aber besser verwalten zu können, wird der gesammte Adressraum in Zonen aufgeteilt. Dies sind meist Second Level Domains oder deren Subdomains. In einer solchen Zone existieren mindestens 2 Name Server. Einmal der primäre Server und ein sekundärer Name Sever, der seine Daten nur vom ersten bezieht. Der primäre Server hat eine vollständige Auflistung aller Rechner, die sich in seiner Zone befinden. Bekommt er nun eine Anfrage über eine unbekannte Adresse, kontaktiert er einen Root Name Server. Das sind die in der hierarchie am höhsten stehenden Server und verwalten die Zonen der Top Level Domains. DNS kann sowohl TCP als auch UDP benutzen. Doch aus Gründen der Effizienz wird TCP nur bei sogenannten Zone Transfers eingesetzt, wenn z.B. ein sekundärer Name Server seine Datenbank durch die eines primären ersetzt. Hierbei macht eine Fehlerkontrolle großen Sinn.

Telnet

Telnet bietet eine unverschlüsselte Verbindung und ist ein Protokoll der Anwendungschicht. Deswegen ist es auch nicht besonders schwer die Daten zu „erstehen“. Mit Telnet lässt sich eine Verbindung zu einem Rechner aufbauen, und mit diesem arbeiten, als wäre man vor Ort. Man nutzt die Programme des Remote-Computers und auch dessen Rechenleistung. Da Telnet bertriebssystemunabhängig arbeitet gibt es keine grafische Oberfläche. Unter Windows nutzt man vielmehr die Eingabeaufforderung. Der zu benutzende Computer muss einen Telnet Server laufen haben, auf den man dann mit einem Client zugreift. Dieser Zugriff erfolgt über eine Anmeldung beim Server. Hierbei werden die Anmeldedaten unverschlüsselt übertragen.