Zusammenfassung "Internet" bei Prof. Dr. Gheis (WS 1998)

Michael Jaeger

7.10.2000

Einführung

Das Internet entstand aus dem ARPANET, das noch nach dem *store and forward-Prinzip* funktionierte: *interface message processors (IMPs)* waren für die Weiterleitung der Nachrichten zu den *Hosts* zuständig. Dabei war jeder IMP mit mindestens zwei anderen verbunden, um die Ausfallsicherheit zu erhöhen.

Die regulären Netzanwendungen zu dieser Zeit waren E-Mail, FTP, und TELNET.

Mit der Entstehung des Internet wurden auch Gruppen gegründet, die für eine Standardisierung sorgen sollten: die *internet engineering taskforce (IETF)* mit ihren *special workgroups (IESGs)*. Informationen rund um das Internet sind in sog. *requests for comments (RFCs)* abgelegt und werden vom *network information center (NIC)* gespeichert.

Neu sepzifizierte Standards durchlaufen folgenden Weg:

Prüfung durch die IAB → **proposed standard**	Erfahrungen sammeln → **draft standard**	≥2 unabh. Implement. → **internet standard**

Architektur

ISO/OSI

Dieses Referenzmodell wurde zur Standardisierung der Netzwerkprogrammierung entworfen. Es entstand über einen sehr langen Zeitraum hinweg und da sehr viele Personen an seinem Entstehen mitgewirkt haben, war es am Ende derart kompliziert und umständlich, dass es in seiner ursprünglichen Form nie implementiert wurde.

Trotz aller Kritikpunkte muss man ihm zu Gute halten, dass es *gut strukturiert* ist. Dem gegenüber steht aber die *zähe Entwicklung*, die *aufwendige und komplexe Implemlentierung*, seine *Ineffizienz* und die *Redundanz* einiger Schichten.

Anwendung	Verarbeitung		Gateway
Darstellung			(Router)
Sitzungssteuerung			
Transport	Transport		
Vermittlung	Internet		**Vermittlung**
Sicherung	Host-an-Netz		**Sicherung**
Bitübertragung	— " —		**Bitübertragung**

TCP/IP

Das Hauptstandbein, auf dem das Internet beruht ist bis heute das TCP/IP-Protokoll, oder genauer das IP-Protokoll (wie der Name schon sagt). Es realisiert ein *4-Schichten Modell* und hat sich aufgrund seiner *Einfachheit* und der *Verbindungslosigkeit* durchgesetzt. Der *bottom-up Ansatz* machte es gegenüber dem ISO/OSI-Referenz *praktikabler*. Leider besitzt das Protokoll *Schwächen im Modell- und in der Ebenenstruktur* und es gibt *keine klare Trennung zwischen Dienst, Schnittstelle und Protokoll.*

Verarbeitung	TELNET, FTP, SMTP, DNS		Gateway
Transport	TCP, UDP		(Router)
Internet	IP		**Internet**
Host-an-Netz	ARPANET, SATNET, Funk, LAN		**Host-an-Netz**

IPv4

Das *internet protocol* bietet eine eindeutige Adressierung aller Rechner durch die Vergabe von *32 Bit-Adressen*. Die Adresse wird dazu in einen *Netzwerk- und einen Rechnerteil aufgeteilt*, was bereits heute zu Problemen führt, da zum einen die *Adressen weltweit knapp* werden und zum anderen die *Vergabe für heutige Belange zu statisch* erfolgt.

Die Aufgabe der Vermittlungsschicht, auf der IP arbeitet, ist die Übertragung von Datenpaketen, inklusive dem Finden eines Weges und einer kleinen Fehlerbehebung. Da IP *verbindungslos* arbeitet, kann es nicht gewährleisten, dass alle Pakete (in der richtigen Reihenfolge) beim Empfänger ankommen. Dafür stellt es aufgrund seiner niedrigen Ansiedlung im Schichtenmodell den *kleinsten gemeinsamen Nenner vieler Netze* dar und ermöglichte erst das Internet. Im Header eines IP-Paketes wird der *type of service*, die *Fragmentierung* und die *time to live (TTL)* beschrieben.

Zusammenfassend die wichtigsten Eigenschaften von IP:

- ▷ einheitliches Adressierungsschema
- ▷ verbindungslos, einfach heterogen
- ▷ höherwertige Kommunikation in höherwertigen Schichten
- ▷ Hierarchiebildung, Dezentralisierung → skalierbar

Ports

Um verschiedene Dienste auf einem Rechner über eine Protokoll (TCP oder UDP) ansprechen zu können, wurde das *Port-Konzept* entwickelt. Jede Anwendung "lauscht" dazu auf einer bestimmten Port-Nummer (16 Bit Zahl) nach Anfragen. Die Ports mit Nummern <256 sind dabei sog. *well-known ports*, die für bestimmte Anwendungen reserviert sind (TELNET, FTP, SMTP etc.). Über diese Ports werden dann ähnlich dem Arbeiten mit Dateien Verbindungen auf- und abgebaut und Daten übertragen. Eine sog. *Socket-Schnittstelle* stellt dabei eine Abstraktion ähnlich dem I/O-Konzept von UNIX zur Verfügung.

TCP

Die Aufgabe der Transportschicht ist die Garantie einer *fehlerfreien Ende-zu-Ende-Kommunikation* zwischen Prozessen. Dazu werden per *Multiplexing* mehrere Transportbeziehungen auf einmal abgehandelt. Als *verbindungsorientierter* Transportdi-

enst muss er für *fehlerfreie Übertragung, Einhaltung der Reihenfolge, Vermeidung von Paketverlusten oder Duplikaten* und *Flußsteuerung* sorge tragen.

Die Transportschicht ist zwischen der Verarbeitungsschicht und der Vermittlungsschicht eingebettet und kommuniziert mit diesen über einen *transport service access point (TSAP)* bzw. einen *network service access point (NSAP)*. Zwei Transportinstanzen kommunizieren wiederum über *transport protocol data units (TPDUs)* miteinander.

TCP-Pakete sind in der Regel kleiner als 64 KB (1500 Byte). Eine TSAP-Adresse besteht bei TCP aus (IP-Adresse, TCP-Port).

Befehle für den Aufbau einer verbindungsorientierten Schnittstelle sind z.B. `CONNECT, LISTEN, SEND, RECEIVE, DISCONNECT`.

UDP

Das *user datagram protocol* stellt einen *verbindungslosen Transportdienst* zur Verfügung und überläßt der Anwendung die Fehlerbehandlung und die Flußsteuerung. Es unterstützt Multicast und Broadcoast und ist deshalb besonder gut für sog. *streaming* geeignet. Eine TSAP-Adresse ist (IP-Adresse, UDP-Portnummer).

ARP

Mittels des *address resolution protocol* werden die logischen IP-Adressen in physische Netzwerkadressen umgewandelt. Jeder Netzwerkadapter besitzt eine weltweit eindeutige 48 Bit MAC-Adresse, die fest in der Netzwerkarte gespeichert ist. In diesem Zusammenhang fungiert ein Router als Proxy-ARP und antwortet mit seiner Adresse auf Anfragen an entfernte Rechner. Für das Erfragen einer physischen Adresse nutzt ARP die Broadcast-Eigenschaft von LANs.

Mittels *RARP (reverse ARP)* können sich Rechner zu ihrer physischen Adresse eine logische erfragen. Dieses Protokoll wurde jedoch im wesentlichen von *BOOTP* verdrängt, das auf UDP basiert.

ICMP

Für die Übermittlung von Fehlermeldungen auf IP-Ebene wird das *internet control message protocol (ICMP)* verwendet. Es unterteilt zwischen verschiedenen Nachrichtentypen wie z.B. "Datagramm nicht zustellbar", "Zeit um", "nehme einen anderen Weg" oder "lebe noch" (beim PING).

Routing

Das Routing ist eine ganz zentrale Funktion von IP. Ziel ist das "Lenken" von Netzwerkpaketen bis zu seinem Ziel, wobei die Adressen von Quelle und Ziel nicht abgeändert werden sollen. Soll nun ein Paket von Host H_1 in Netz N_1 zu Host H_2 in Netz N_2 wandern, so sucht H_1 ersteinmal nach H_2 in seinem Netzwerk. Befindet sich H_2 in einem anderen Netz, so übernimmt ein Router R_1 aus N_1 das Paket und leitet es immer weiter an andere Router, bis das Paket zuletzt Router R_2 in Netz N_2 erreicht hat, der es dann an den Ziel-Host H_2 schickt. Jedes Weiterreichen von einem Router zum nächsten wird als *Hop* bezeichnet und wird zur Berechnung der *time to live (TTL)* verwendet. Das Weiterleiten von einem Router zum nächsten geschieht auf Basis sog. *Routing-Tabellen*. Hierbei unterscheidet man das *intra-* und das *interdomain routing*, wobei ersteres vorrangig abgeschlossene Netzwerke wie z.B. von Firmen oder Universitäten betrifft. Beim intradomain routing werden vorrangig die Protokolle RIP und OSPF eingesetzt.

RIP

Router gleichen über spezielle Protokolle wie dem *routing information protocol (RIP)* ihre Routing-Informationen ab, wobei die Tabellen natürlich auch statisch eingestellt werden können, was jedoch nur bei sehr kleinen Netzwerken Sinn macht. Ein dynamischer Abgleich der Routing-Informationen unter den Routern ist vor allem wegen der gewünschten Robustheit gegenüber Ausfällen wünschenswert. Die Router tauschen untereinander sog. *Distanzvektoren* aus, und teilen einander alle 30s mit, wie weit es zu den von ihnen erreichbaren Netzen ist (in Hops). Auf Basis dieser Informationen kann ein Router beim Verschicken von IP-Paketen immer den kürzesten Weg suchen.

Die Berechnung des kürzesten Weges wird nur anhand der Anzahl der Hops berechnet, die Geschwindigkeit spielt keine Rolle. Auch ist die Konvergenz beim Ausfall von Routern sehr schlecht und Subnetz-Markierungen sind mit RIP ebenfalls nicht möglich.

OSPF

Wegen der Schwächen von RIP wurde die Entwicklung eines neuen Routing-Protokolls begonnen: *open shortest path first*. Dabei wollte man eine *Authentisierung* von Routing-Nachrichten, eine *Lastbalancierung*, den *Aufbau von Hierarchien* (sog. "areas"), ein *neues Maß für Entfernung* und die Unterstützung verschiedener *types of services* ermöglichen.

BGP

An das Routing zwischen Netzwerken unterschiedlicher Parteien werden ganz andere Anforderungen gestellt, als an das Routing innerhalb eines Netzwerkes, so möchte man u.U. bestimmten Transitverkehr unterbinden, bestimmte Datagramme (nicht) übertragen, internes- und externes Routing trennen und die Erreichbarkeit dem optimalen Weg voranstellen. All diese Anforderungen erfüllt das *border gateway protocol (BGP)*, das aus dem *exterior gateway protocol (EGP)* entstanden ist, was es zum Einsatz beim interdomain routing prädestiniert.

IP Subnetze

Eng verbunden mit dem Begriff "routing" ist der des "Subnetzes". Durch den Einsatz von Netzmasken, ebenfalls 32 Bit-Zahlen, die mit der IP-Adresse UND-verknüpft werden, wird der IP-Adressraum in sog. C-, B- und A-Netze eingeteilt, mit jeweils 255, 255^2 bzw. 255^3 Rechnern. Dies führt heutzutage zu Problemen, das sich diese Aufteilung als sehr unflexibel erwiesen hat und so in einigen Netzen viele Adressen "brachliegen" während woanders Adressenmangel herrscht. Ebenfalls nicht berücksichtigt wurde der Umzug von Rechnern in neue Subnetze und der damit verbundene Aufwand zum Ändern der IP-Adressen und der Bekanntmachung selbiger. Auch der organisatorische Aufwand, der mit dem Beantragen einer neuen IP-Adresse verbunden ist führte zu großer Unzufriedenheit.

Ein Router kann anhand der Subnetzmaske durch einfache UND-Verknüpfung prüfen, zu welchem Netzwerk eine Adresse gehört und Pakete dementsprechend weiterleiten. Dazu werden in der Routingtabelle Einträge gespeichert, die die Subnetznummer, die Subnetzmaske und den nächsten Router enthalten.

IPv6

Die Unzulänglichkeiten von IPv4, das ungeheure Wachstum des Internets und neue Anwendungsanforderungen führten 1990 zu der Entwicklung eines Nachfolgers von

IP - IPv6 oder *IPNG (IP next generation).* Ziel bei der Entwicklung waren

- ein *größerer Adressbereich* (durch 128 Bit-Adressen)
- eine *Reduktion des Umfangs der Routing-Tabellen*
- *schnelles Routing* (durch Header fester Länger, Fragmentierung beim Anwender und Weglassen von Checksum-Berechnungen)
- *erhöhte Sicherheit* (durch Integration von IPSEC: Authentizität, Integrität und Vertraulichkeit mittels DES und MD5)
- Unterstützung *neuartiger Datenströme* (durch Datenströme mit Prioritäten und einer QoS) und mobiler Hosts
- eine *Koexistenz mit IPv4* (durch *tunneling*, sog. IPv4/IPv6-Router packen IPv6-Pakete in IPv4-Pakete und umgekehrt)
- *verbessertes Multicasting*

Adressbereiche lassen sich in IPv6 aufteilen in *registry* (für die Registrierungsorganisationen), *provider* (für die Internetprovider), *subscriber* (für den Kunden) und *subnet* (für das Subnetz des Kunden). Diese Vorgehensweise ermöglichkt sowohl verbindungsspezifische als auch standortspezifische lokale Adressen.

Anwendungen

Basis-Dienste

Bei Internet-Anwendungen kann man zwischen Basisdiensten und spezialisierteren Anwendungen unterscheiden.

DNS

Der *domain name service* sorgt für eine Übersetzung von sprechenden Namen in IP-Adressen und ist damit einer der wesentlichen Dienste des Internets. Der DNS ist hierarchisch und verteilt organisiert. Eine Anfrage wird, wenn sie vom lokalen DNS-Server nicht beantwortet werden kann, an einen sog. *root-server* weitergereicht und dann an die DNS-Server der betreffenden Domain- bzw. Subdomain-Server weitergereicht. DNS-Anfragen erfolgen per UDP und jeder DNS-Server kennt mindestens einen Vorgänger Name-Server.

Jeder Eintrag in der DNS-Datenbank ist ein 5-Tupel bestehend aus *domain name, type, class* und *value.* In manchem Fällen kommt noch eine *TTL* hinzu. Der Namensraum wird in nicht überlappende Zonen unterteilt, die wiederum primäre und evtl. sekundäre DNS-Server enthalten.

Im Zuge von IPv6 wird auch der DNS erneuert werden, da er für das dynamische IPv6 zu unflexibel ist.

Anwendungen

E-Mail

Die Übertragung von E-Mails erfolgt nach dem *store and forward*-Prinzip über das *simple message transfer protocol (SMTP).* Dabei wird die E-Mail von einem *mail user agent (MUA)* zu der nächsten *message transfer unit (MTU)* übertragen, die die Nachricht dann wieder an die nächste MTU weiterleitet, bis die Ziel-MTU erreicht ist. Jede E-Mail besitzt einen "Umschlag", auf dem wichtige Informationen wie der

Empfänger, Absender etc. gespeichert sind. Im "Briefkopf" der E-Mail stehen dann Daten wie das Erstellungsdatum, der Betreff etc.

Um mit E-Mail auch komplexere Daten verschicken zu können, wurde der *multipurpose internet mail extension (MIME)* Standard eingeführt, der die Probleme mit Sonderzeichen und binären Daten behebt. Dazu wird die E-Mail-Nachricht in verschiedene Teile unterteilt, die jeweils über einen eigenen MIME-Header verfügen, der z.B. eine Inhaltsbeschreibung, die Kodierung und der Inhaltstyp enthält.

TELNET

Dieser Dients diente von Anfang an der Fernbedienung von Rechnern und war eine der ersten Internet-Anwendungen überhaupt. Dazu baut der Client eine TCP-Verbindung zum Server auf und leitet alle Eingaben an den Server weiter. Verwendet werden dafür spezialisierte Protokolle wie das *network virtual terminal (NVT)*.

FTP

Ebenfalls unter den ersten Internetanwendungen war das *file transfer protocol*, das der Übertragung von Dateien von/zu entfernten Hosts dient. Da es auf TCP aufsetzt (es gibt auch eine Version, die auf UDP aufsetzt - das sog. *TFTP*) ist die Datenintegrität gewährleistet.

USENET

Über die sogennanten Internet News können Nachrichten in sog. Diskussionsforen ausgetauscht werden. Entstanden ist es aus dem *unix to unix copy (uucp)* Protokoll, das früher für Datenübertragung verwendet wurde. Heute wird stattdessen das *network news transfer protocol (NNTP)* verwendet. News-Server gleichen sich untereinander nach dem *pull-push-Prinzip* ab und Clients können bei Servern nach neuen Artikeln fragen oder welche veröffentlichen. Für jeden Artikel wird ein Verfallsdatum vorgehalten, nachdem dieser gelöscht werden kann.

WWW

Die Anwendung, die das Internet erst richtig populär machte, war das *world wide web*. Es stellt ein verteiltes Hypermedia-System dar, in dem Dokumente bestehend aus Text, Grafik und Multimedia-Daten strukturiert veröffentlicht werden können. Die Einführung des HTTP-Protokolls und von HTML schafften einen Standard, der plattformunabhängig und einfach gestaltet war.

Web-Server sind über den Port 80 erreichbar, und kommunizieren verbindungslos über das HTTP-Protokoll. Auf eine Ressource des Web-Servers wird mittels eines *uniform resource locators (URL)* zugegriffen, der aus dem Namen des Web-Servers und dem Pfad zu dem gewünschten Dokument besteht.

Um die dynamische Gestaltung von HTML-Seiten zu ermöglichen wurde das *common gateway interface (CGI)* eingeführt. es definiert eine Schnittstelle zum Aufruf von Programmen auf dem Server und wird durch fast alle Programmiersprachen unterstützt. Leider stellen sie ein gewisses Sicherheitsrisiko dar, da sie von jedem Benutzer aufgerufen werden können und unter der Berechtigung des Web-Servers laufen.

Suchmaschinen

Aufgrund der Dezentralität des Webs und dem ungeheuren Wachstum wurde das Finden von Seiten immer schwerer, was die Entwicklung von Suchmaschinen erforderlich machte. Diese durchforsten das Netz und indexieren die Seiten nach ge-

fundenen Wörtern und sog. *Meta-Tags*. Die hohe Dynamik des WWW führt leider oft zu veralteten Links und zu unbrauchbaren Ergebnissen. Damit die Server von den Suchmaschinen nicht überlastet werden wurde der sog. *robot standard* eingeführt, nach dem ein Administrator bestimmte Pfade auf seinem Server freigeben oder sperren kann.

Servlets

Die Java-Antwort auf CGI heißt *servlet* bezeichnet eine Menge von Java-Klassen, die vom Web-Server ausgeführt werden. Jedes Servlet stellt für sich eine Art "mini-Webserver" dar und kann auch komplette Sessions verwalten. Durch Datenbankunterstützung eignen sich Servlets auch zur Programmierung von n-Tier Anwendungen z.B. mit CORBA.