PPS-Seminar
Grundlagen der Internet-Technologie, SS 03

IP Adressierung und Routing fürs Internet

Stephan Senn
ssenn@ee.ethz.ch
29. April 2003

1 Zur Geschichte

In den sechziger Jahren gab die Verteidigungsstelle der amerikanischen Armee, die *Defense Advanced Research Projects Agency,* einer Forschergruppe den Auftrag, ein Netzwerk zu schaffen, das auch beim Ausfall einzelner Bereiche immer noch seine Funktion erfüllt. Zudem sollten Computer verschiedener Hersteller und Bauarten miteinander kommunizieren können. Die Rede war von einem Netzwerk, dessen Kommunikationswege nicht fest bestimmt sind, sondern dynamisch festgelegt werden. Das Projekt *DARPA Internet Program* kam Anfang der achziger Jahre zum Abschluss und gipfelte in der Ausarbeitung eines Dokuments namens *Internet Protocol*. Die 7. überarbeitete Version dieses Dokuments erschien im September 1981 und stellt bis heute den Standard für das Internet Protokoll der Version 4 dar (kurz IPv4 genannt). Gegenwärtig ist die Version 6 (IPv6) in Evaluationsphase und dürfte sich bald als neuer Standard für das Internet Protokoll etablieren.[1] IPv4 ist als E-Dokument mit der Beschreibung RFC791 auch im Internet vorhanden.[1]

2 Das Internet Protokoll

2.1 Standard-Internet-Protokollarchitektur

Die Internet-Protokollarchitektur ähnelt der allgemeinen Standard-Netzwerkarchitektur, die ISO/OSI (International Standard Organization's Open System Interconnect) genannt wird. Das ISO/OSI-Modell [2,3] ist eingeteilt in sieben Schichten. Jede Schicht erfüllt eine bestimmte Aufgabe im Netzwerk. Die Tabelle 1 zeigt kurz die Aufgaben der einzelnen Schichten des ISO/OSI-Modells.

Application Layer (Applikationsschicht)	für den Endbenutzer bestimmte Anwendungen (z.B. FTP, TELNET, SMTP, usw.)
Presentation Layer (Darstellungsschicht)	den semantisch korrekten Datenaustausch zwischen unterschiedlichen Systemen ermöglichen
Session Layer (Sitzungsschicht oder Kommunikationssteuerungsschicht)	Dienste für einen organisierten und synchronisierten Datenaustausch
Transport Layer (Transportschicht)	beschreibt den Ablauf des Transports der Daten im Netzwerk: Optimierung der Benutzung der Netzdienste, Anpassung von schnelleren und langsameren Netzwerken, usw.
Network Layer (Vermittlungsschicht)	Datenpakete vom Sender-Host über die dazwischenliegenden Router zum Empfängerhost zu leiten, Verbindungsauf- und Verbindungsabbau, Multiplexing und Überlastkontrolle
Datalink Layer (Sicherungsschicht)	Erkennen und/oder Beheben von Übertragungsfehlern, Flusskontrolle, Medienzugangskontrolle
Physical Layer (Physikalische Schicht)	zuständig für die Bitübertragung von Daten

Tabelle 1 ISO/OSI-Modell

Eine detailliertere Aufgabenbeschreibung der Schichten des ISO/OSI-Modells sind den Beiträgen ‘Grundlagen des Internet’ sowie ‘Netzwerktechnologien fürs Internet’ zu entnehmen.

[1] Die Versionen 1,2,3 und 5 des Internet Protkolls wurden für Testzwecke gebraucht, die sich aber nie als Standard etabliert haben.

Die Internet-Protokollarchitektur kennt nicht alle Schichten des ISO/OSI-Modells; ist diesem aber sehr ähnlich. Die Tabelle 2 zeigt die Internet-Protokollarchitektur im Vergleich zum ISO/OSI-Modell.

ISO/OSI-Modell	**Standard-Internet-Protokollarchitektur**
Application Layer	Application Layer
Presentation Layer	
Session Layer	
Transport Layer	Transmission Control Protocol (TCP) / User Datagram Protocol (UDP)
Network Layer	Internet Protocol (IP) / Internet Control Message Protocol (ICMP)
Datalink Layer	Link Layer
Physical Layer	

Tabelle 2 Die Standard-Internet-Protokollarchitektur

Die physikalische Schicht sowie die Sicherungsschicht wurden bei der Internet-Protokollarchitektur zusammengefasst. Ihre Aufgabe besteht darin, die Hardware mit den dazugehörigen Treibern zur Übertragung von Bits zur Verfügung zu stellen. Das Error-Handling sowie die Flusskontrolle sind weitere wichtige Bestandteile dieser Schicht. Die Netzwerkschicht wird durch das eigentliche Internet Protokoll (IP) mit dem dazughörigen Fehlerprotokoll ICMP bereitgestellt. Das ICMP hat nichts zu tun mit dem klassischen Error-Handling der untersten Schichten. Es beschreibt Fehler beim Aufbau von Verbindungen. Wird beispielsweise eine Adresse nicht gefunden oder bricht auf einmal die Verbindung zusammen, dann wird das ICMP aktiv. Die Transportschicht wird durch die Transportprotokolle TCP und UDP beschrieben. Die Applikationsschicht ist nahezu identisch mit der Applikationsschicht des ISO/OSI-Modells. Hier sind die bekannten Internetdienste angesiedelt: z.B. HTTP, FTP, TELNET, u.a., die jeder Endbenutzer mit Hilfe eines Browsers oder einer Shell-Applikation auf den meisten Betriebssystemen verwenden kann. Die Darstellungs- sowie die Sitzungsschicht haben bei der Internet-Architektur keine Bedeutung.

2.2 Aufgabe

Die Aufgabe des Internet Protokolls besteht darin, Datenpakete, auch Datagramme genannt, an Subnetze weiterzuleiten. Dabei werden zwei grundlegende Funktionen benötigt: Adressierung und Fragmentierung. Bei der Adressierung geht es um das Prinzip der Adressvergabe und dem systematischen Auffinden von Adressen im Netz. Die Fragmentierung (Fragmentation) eines Datenpakets bezeichnet dessen ‚Aufsplittung' in kleinere Datenpakete. Das spätere Zusammenfügen dieser Datenpakte wird als Assembling bezeichnet. Die Fragmentierung ist sehr wichtig für das Zusammenspiel von leistungsfähigen mit schwachen Netzwerken. Durch die Fragmentierung beim Sender und dem späteren Assembling beim Empfänger kann der Übertragungskanal verkleinert werden. Es wird hier eindringlich darauf hingewiesen, dass die Netzwerkschicht auch beim Internet Protokoll keine Mechanismen zur Fehlerkorrektur oder zur Flusskontrolle beinhaltet. Allerdings besitzt das Internet Protokoll über eine Fehlererkennung. Es kann festgestellt werden, ob ein Datenpaket richtig übermittelt wurde, oder ob es fehlerhaft ist. Die Fehlererkennung erstreckt sich aber nicht auf die eigentlichen Daten, sondern nur auf den Datenheader, der Kopfzeile jedes Datenpakets. Deshalb ist auch keine Fehlerkorrektur möglich.

2.3 Internet Header

Jedes Datenpaket besitzt neben den eigentlichen Daten eine sogenannte Kopfzeile, auch Header genannt. Dort werden alle wichtigen Informationen, ähnlich wie bei einer Etikette auf einem richtigen Paket bei der Post, abgelegt. Die Abbildung 1 zeigt den Aufbau des Internet Headers. Es folgen die Erläuterungen zu den einzelnen Feldern.

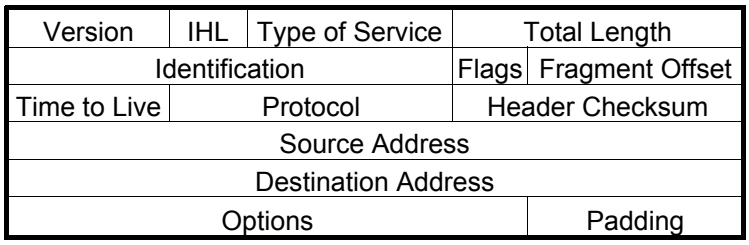

Abbildung 1 Internet Header

Version:	4Bits	bezeichnet die Version des Internet Protokolls (Version 4)
IHL:	4Bits	bezeichnet die Länge des Internet Headers (Internet Header Length)
Type of Service:	8Bits	beschreibt verschiedene Dienste für die Beeinflussung der Übertragungsqualität wie Verzögerung, Durchsatz, Zuverlässigkeit, u.a.
Total Length:	16Bits	Ein Datenpaket hat eine maximale Länge von 65535Bytes. Alle Hosts müssen Datenpakete von mindestens 576Bytes verarbeiten können.
Identification:	16Bits	Wird ein Datenpaket in Fragmente zerlegt, so müssen diese eindeutig gekennzeichnet werden, dass sie zu ein und demselben Datenpaket gehören.
Flags:	3Bits	verschiedene Control Flags
Fragment Offset:	13Bits	Dieses Feld gibt jedem Fragment eine eindeutige Zuordnung. Im Unterkapitel ‚Fragmentierung' wird dieser Prozess ausführlich besprochen.
Time to Live (TTL):	8Bits	Findet beispielsweise ein Datenpaket die angegebene Zieladresse nicht, da diese nicht mehr existiert oder eine andere Adresse hat, dann würde das Datenpaket andauernd im Netzwerk ‚herumirren' und unnötig Ressourcen binden. Damit dies nicht geschieht, beinhaltet jedes Datenpaket einen Zähler, der von einer bestimmten Zahl rückwärts gegen null zählt. Beträgt der Wert dieses Feldes null bevor das Datenpaket an der Zieldadresse angelangt ist, dann wird es vom System gelöscht. Die maximale Länge beträgt 255s (4.25min). Bei jeder Zwischenstation im System, wird der Wert dieses Feldes um 1 dekrementiert. Ursprünglich sollte die Dekrementierung um 1 der Zeiteinheit von einer Sekunde entsprechen. Die gewünschte Startzeit für ein Datenpaket wird beim Routing bestimmt.
Protocol:	8Bits	Dualzahl-Codierung des Next-Level-Protokolls
Header Checksum:	16Bits	Die Prüfsumme Header Checksum dient der Verifikation der korrekten Übertragung. Dabei wird bei jeder Prozessierung des Headers, diese Prüfsumme neu berechnet. Die Prüfsumme erstreckt sich nur auf die Daten des Headers, und nicht auf die eigentlichen Daten im Datenpaket. Es gibt auch keine Fehlerkorrektur. Ist kein Fehler aufgetreten, so hat die Prüfsumme den Wert null; ansonsten ungleich null. Bei fehlerhaften Übertragungen wird ein Fehlerprotokoll erstellt und an die Quelladresse gesendet. Diese Aufgabe übernimmt das Internet Control Message Protocol (ICMP). Eine automatische Neuübertragung des fehlerhaften Datenpakets erfolgt nicht.
Source Address:	32Bits	bezeichnet die Quelladresse

Dest. Address:	32Bits	bezeichnet die Zieladresse
Options:	var	Weitere Optionen wie z.B. Sicherheitaspekte, Route-Tracking, Internet-Time-Stamps, u.a. können hier eingeführt werden.
Padding:	var	Padding ist ein Ausgleichsfeld. Ein Internet Header muss ein Vielfaches von 32Bits sein. Dieses Feld dient dazu, diesen Anforderungen gerecht zu werden. Für das ‚Auffüllen' werden 0-Bits verwendet.

Tabelle 3 Felder des Internet Headers

2.4 Fragmentierung

Bei der Fragmentierung wird ein Datenpaket in kleinere Datenpakete, sogenannte Fragmente, aufgeteilt. Dabei dient das Feld *Identification* als Grundkennung der Fragmente mit dem Ziel, die Zugehörigkeit aller Fragmente zu einem bestimmten Datenpaket sicherzustellen. Die Reihenfolge der Fragmente ist für das Assembling, das Zusammenfügen, unumgänglich. Das Feld *Fragment Offset* dient zur Festlegung der Reihenfolge. Beim ersten Fragment ist dieser Wert null. Beim zweiten Fragment ist er null plus die Grösse des neuen Fragments. Beim dritten Fragment ist er die Grösse des zweiten Fragments plus diejenige des neuen Fragments. Usw. Da alle Fragmente die gleiche Grösse haben ergibt sich daraus:

- 1.Fragment: 0 mal Fragmentgrösse
- 2. Fragment: 1 mal Fragmentgrösse ...
- n. Fragment: (n-1) mal Fragmentgrösse

Das Feld *More Fragments* bezeichnet, ob noch mehr Fragmente folgen. Beim Assembling werden nun die Fragmente der Reihe nach zum ursprünglichen Datenpaket zusammengesetzt. Mit dem Flag *Don't Fragment* ist es möglich, das Fragmentieren von Datenpaketen zu unterbinden. Die Abbildung 2 zeigt eine vereinfachte Darstellung der Fragmentierung. Genauere Informationen zur Fragmentierung und zum Assembling sind dem E-Dokument RFC791 zu entnehmen.[1]

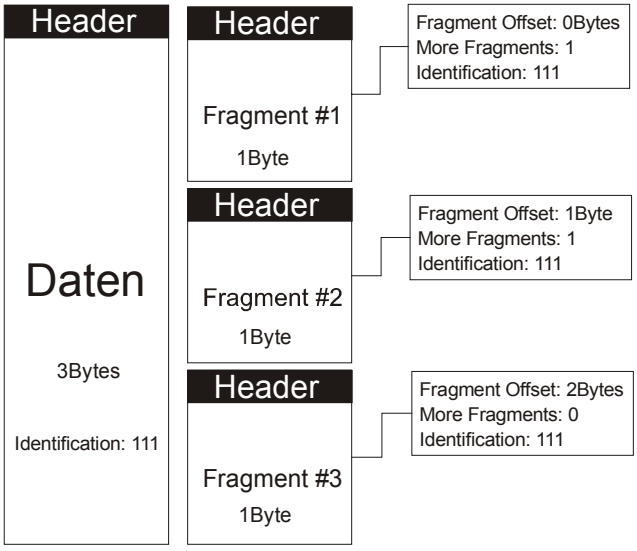

Abbildung 2 Fragmentierung eines Datenpakets

2.5 Adressierung und Adressauflösung

2.5.1 Die Internetadresse

Jede Internetadresse besteht aus vier Zahlen im Bereich von 0 bis 255 (1Byte), die durch einen Punkt voneinander getrennt sind (z.B. 129.132.200.35). Sie hat eine Länge von 32Bits und wird zusätzlich in drei Klassen eingeteilt. Jede Adresse lässt sich charakterisieren durch einen Präfix, eine Netzadresse und eine Hostadresse (auch Endsystemadresse genannt). Die Einteilung der Adressklassen ist in den Tabellen 2 und 3 dargestellt.

Abbildung 3 Format der Internetadresse

Klasse	Präfix	Netzadresse	Anzahl Netzadressen	Hostadresse	Anzahl Hostadressen
A	0	7Bits	126	24Bits	16777214
B	10	14Bits	16382	16Bits	65534
C	110	21Bits	2097150	8Bits	254

Tabelle 4 Adressklassen (1)

Klasse	von	bis	feste Bitstellen
A	1.0.0.0	126.0.0.0	**0**0000000.00000000.00000000.00000000
B	128.0.0.0	191.255.0.0	**10**000000.00000000.00000000.00000000
C	192.0.0.0	223.255.255.0	**110**00000.00000000.00000000.00000000

Tabelle 5 Adressklassen (2)

Wir halten also folgendes fest:

- Die Klasse A eignet sich für grosse Netzwerke mit vielen Hosts.
- Die Klasse B eignet sich für mittlere Netzwerke mit einer mittleren Anzahl von Hosts.
- Die Klasse C eignet sich für kleine Netzwerke mit wenigen Hosts.

Wie man vielleicht bemerkt hat, werden nicht alle Internetadressen verwendet. Die Tabelle 4 zeigt einige spezielle Adressen.

Spezielle Internetadressen	Beschreibung
0.0.0.0	default Route
127.0.0.0	lokales Netzwerk
224.0.0.0 bis 255.255.255.255	reserviert für zukünftige Verwendung

Tabelle 6 Spezielle Internetadressen

Die Abbildung 4 zeigt das Netz der ETH. Wie man leicht sieht handelt es sich um die Adressklasse C. Die Netzadresse lautet 129.132. Der Host www.ee.ethz.ch hat die Hostadresse 2.198. Die vollständige Internetadresse von www.ee.ethz.ch lautet: 129.132.2.198.

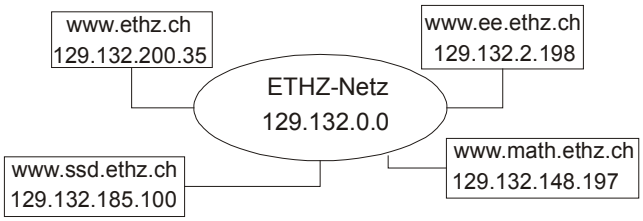

Abbildung 4 Das Netz der ETH (Beispiel)

2.5.2 Subnetze und Netzmaske

Um ein Netzwerk von einem anderen abzugrenzen, besteht die Möglichkeit, Subnetze zu erzeugen. Oft ist dies bei grossen Netzwerken von Vorteil. Eine grosse Firma hat beispielsweise verschiedene Abteilungen. Jede Abteilung soll nun wie ein eigenes Netz operieren. Dazu können Subnetze verwendet werden. Dazu wird eine Art ‚Schablonentechnik', auch Maskierung genannt, verwendet. Die sogenannte Netzmaske ist für die einzelnen Adressklassen in der Tabelle 5 angegeben.

Klasse	Netzmaske	Netzmaske in Binärform
A	255.0.0.0	11111111.00000000.00000000.00000000
B	255.255.0.0	11111111.11111111.00000000.00000000
C	255.255.255.0	11111111.11111111.11111111.00000000

Abbildung 5 Subnetzadressen

Die Maskierung betriff die Netzadresse. Dazu wird die Internetadresse mit der Netzmaskierung bitweise addiert. Dadurch erhalten wir die Netzadresse. Denn nur Hosts mit dieser Netzadresse sollen mit einem anderen Host mit derselben Netzadresse kommunizieren können. Dadurch wird eine Art Subnetz erzeugt: Denn nur Hosts, die dieselbe Netzadresse haben, können miteinander kommunizieren. Die Abbildung 6 zeigt den Vorgang der Maskierung. Dieses spezielle Thema ist auch im Internet zu finden.[4]

Internetadresse	149.76.6.4	10010101.01001100.00000110.00000100
Maskierung	255.255.0.0	11111111.11111111.00000000.00000000
AND	149.76.0.0	10010101.01001100. 00000000.00000000

Abbildung 6 Maskierung

Beispiel

Eine Firma besitzt drei grosse Abteilungen A, B und C mit den Netzadressen 149.76, 149.77 und 149.78 (Adressklasse B). Die einzelnen Abteilungen werden als Subnetze geführt; d.h. ein Host aus der Abteilung A kann nicht mit einem Host der Abteilung B kommunizieren. Die Subnetzmaske lautet dann: 255.255.0.0.

2.5.3 Symbolische Internetadressen

Die numerischen Internetadressen mögen zur internen Adressierung dienen, für den Benutzer eines Browsers wären sie geradezu verwirrend. Wer merkt sich schon Folgen von Zahlen? – Darum entwickelte man eine Art ‚Alias-System'. Wenn man in einem Browser www.ethz.ch eintippt, dann erfolgt zuerst eine Query-Abfrage in einer globalen Datenbank.[5,6,7] Dort wird dem Alias www.ethz.ch die numerische Internetadresse 129.132.200.35 zugeordnet. Der Aufbau dieser symbolischen Internetadressen ist durch das Domain Name System (DNS) geregelt. Ein Domain bezeichnet einen Bereich eines Netzwerks. Jeder Domain wird mittels einer Punktnotation gekennzeichnet. So beschreibt die Internetadresse www.ethz.ch die Domain .ethz.ch. Diese Domain

liegt in der grösseren Domain .ch. Jede Domain besitzt eine globale Datenbank, die die ‚Alias-Zuweisungen' der einzelnen Teildomains beinhaltet. Mit dem Browserbefehl www.ethz.ch wird eine Anfrage (Query-Request) an die nächst höhere Domain .ch gestellt. Die Antwort dieser Anfrage (Query-Answer) beinhaltet die numerische Internetadresse von www.ethz.ch. Sie kann natürlich auch leer ausfallen, wenn die entsprechende Internetseite nicht gefunden wird. In diesem Fall meldet der Browser: Site not found.

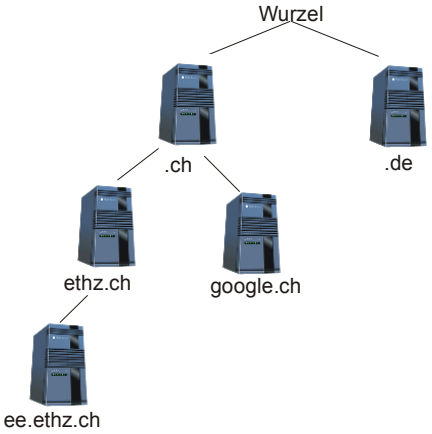

Abbildung 7 Domain Name System (DNS)

2.5.4 Address Resolution Protocol (ARP)

Jede Ethernet- oder Token-Ring-Karte besitzt eine weltweit eindeutige Kennung, die MAC[2]-Adresse. Will nun ein Router ein Datenpaket an eine Internetadresse eines Hosts oder eines Servers schicken, der mit einer Ethernet- oder einer Token-Ring-Karte ausgestattet ist, dann muss zuerst die MAC-Adresse ermittelt werden. Dies geschieht mittels eines Broadcast-Verfahrens. Der Router sendet die Internetadresse mit einem ARP-Paket an alle Hosts oder Server eines Subnetzes (ARP-Request). Derjenige Host oder Server, bei dem die Internetadresse zutrifft, sendet die MAC-Adresse mit einem ARP-Paket zurück (ARP-Reply). Die anderen Rechner löschen das ARP-Paket. Auf diese Weise erhält der Router die MAC-Adresse und kann ein Datenpaket an die entsprechende MAC-Adresse liefern. Die Abbildung 8 zeigt diesen Vorgang nochmals. Eine sehr gute Erklärung zu ARP ist auch im Internet zu finden.[8]

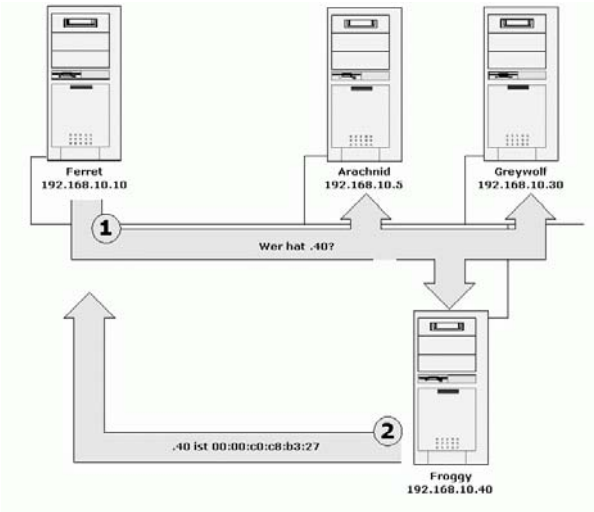

Abbildung 8 ARP-Schema (Beispiel)[8]

[2] steht für **M**edia **A**ccess **C**ontrol

2.5.5 Classless Inter-Domain Routing (CIDR)

Viele kleinere und mittlere Firmen haben das Problem, dass sie mit der Adressklasse C zu wenig Hosts adressieren können. Deshalb wählen sie die Adressklasse B. Das hat aber umgekehrt zur Folge, dass viele Internetadressen unbenutzt bleiben. Heute sind die Internetadressen der Klasse B fast völlig erschöpft. Das Classless Inter-Domain Routing stellt aber nur eine vorübergehende Lösung dar. Eine endgültige Lösung wird nur mit der neuen Adressierung von IPv6, dem Internet Protokoll der Zukunft, erreicht. Ein anderes Problem, das auch nicht unbedeutend ist, betrifft die Routing-Tabellen, die verwendet werden, um Datenpakete korrekt an ihr Ziel zu führen. Diese Tabellen verzeichnen einen enormen Zuwachs.[3] Die Idee besteht nun darin, die feste Einteilung der Adressklassen zu umgehen. Dadurch wird die Einteilung von Host- und Netzadresse variabel. Dem System muss aber mitgeteilt werden, wie viel Bits der Host- und Netzadresse zugeteilt sind, damit eine Unterscheidung von Netzadresse zu Hostadresse möglich wird. Dies geschieht mittels eines Präfix mit der Notation /N. Dabei bezeichnet N die Anzahl Bits der Netzadresse. Eine Internetadresse sieht dann wie folgt aus: 195.5.0.0/22. Für N sind nur Zahlen von 13 bis 27 erlaubt. Internetadressen werden also in sogenannte Adressblöcke aufgeteilt. Die Barriere der 8-Bit-Einteilung wird dabei durchbrochen. Das folgende Beispiel dient der Veranschaulichung. Mehr zu diesem Thema findet man im Internet.[9]

Präfix	Internetadresse	in Binärform
/22	195.5.0.0	11000011.00000101.000000**00**.00000000

Abbildung 9 Classless Inter-Domain Routing (Beispiel)

3 Routing

Ein Datenpaket beinhaltet keine Weginformationen, die beschreiben, auf welche Weise das Datenpaket an sein Ziel kommt. Diese Aufgabe übernimmt das Routing. Es sorgt dafür, dass die einzelnen Datenpakete korrekt weitergeleitet werden.

Zwischen Hosts befindet sich eine elektronische oder optische Vernetzung und sogenannte Verteilungsknoten, genannt Router. Im Unterschied zu Hosts besteht der Router nur aus der Netzwerkschicht (IP und ICMP) und der Link-Schicht. Ein Router hat die Aufgabe, die Daten richtig weiterzuleiten. Dazu werden Routing-Tabellen verwendet. Diese Tabellen beschreiben die Netztopologie oder Teile daraus; je nach Routing-Verfahren. – Die Initialisierung der Routing-Tabelle kann zu Beginn festgelegt sein (statisch) oder zur Laufzeit (dynamisch) festgelegt werden. Man spricht dann von statischem oder dynamischen Routing. Es kann beispielsweise vorkommen, dass ein Host mit einer neuen Internetadresse einem Netzwerk zur Laufzeit hinzugefügt wird. Ein dynamisches Routing-Verfahren würde diesen Host ins Netz einbinden und mitbenutzen. Bei einem statischen Routing-Verfahren muss dem System zuerst beigebracht werden, dass sich dieser neue Host im Netz befindet. Zur Laufzeit würde dieser Host nicht erkannt. Es gibt grundsätzlich zwei Routing-Verfahren:

- Distanz-Vektor-Routing
- Link-State-Routing

Das Routing wird durch sogenannte Routing-Protokolle beschrieben. Da es mehrere solcher Protokolle gibt, bedarf es einem allgemeinen Routing-Protokoll, das alle Protokolle untereinander vereinigt und die Kommunikation gewährleistet. Dieses allgemeine Protokoll heisst Border Gateway Protocol (BGP).

[3] Mehr zum Thema ‚Routing' folgt im nächsten Kapitel.

3.1 Distanz-Vektor-Routing

Beim Distanz-Vektor-Routing kennt jeder Router nur die unmittelbare Umgebung und ist somit auf die Informationen angewiesen, die ihm andere Router in Form von Routing-Tabellen bereitstellen. Jeder Router kennt also nur jeweils einen Teil der Topologie des Netzwerks. Das Austauschen der Routing-Tabellen und somit das Austauschen von Distanzinformationen ermöglicht die Ermittlung der kürzesten Distanz und somit den besten Pfad für das Weiterleiten der Datenpakete; daherDistanz-Vektor-Routing genannt. Dieses Routing-Konzept missachtet aber Parameter wie Datendurchsatz, Wartezeiten, usw., die zu Zeitverzögerungen führen. Eine sehr kurze Verbindung muss nicht zwangsläufig mit einer besseren Verbindung korrelieren. Zudem handelt es sich hier um statisches Routing, denn die Distanzen müssen den Routern zu Beginn bekannt sein. Der Vorteil dieses Routing-Konzeptes liegt in der einfachen Implementierung. Das Distanz-Vektor-Routing wird meist für kleinere Netzwerke verwendet, da die Distanzangaben meist nicht eine bestimmte Grösse überschreiten dürfen. Das Routing Information Protocol (RIP) beschreibt ein Routing-Verfahren auf der Grundlage des Distanz-Vektor-Routing. Es wird im folgenden beschrieben.

3.1.1 Routing Information Protocol (RIP)

Wichtig für das Distanz-Vektor-Routing ist das Festlegen der Distanz von einer Zwischenstation zur nächsten. Die Distanz wird in Hops angegeben. 1Hop bezeichnet das Passieren 1 Zwischenstation im Netz (z.B. Server, Router, usw.). Dementsprechend bezeichnen nHops n Zwischenstationen. Alle 30s sendet ein Router seine Routing-Tabellen an alle Zwischenstationen, die direkt mit dem Router verbunden sind. Diese Router machen wiederum dasselbe und schicken ihre Routing-Tabellen an den Router. Das folgende Beispiel soll dies veranschaulichen.

Beispiel

Gegeben sei das Netzwerk in Abbildung 10, bestehend aus sieben Netzen, sechs Routern, zwei Servern und drei Hosts. Host #1 will eine Datei auf dem Server #2 speichern. Die Datei wird zunächst an den Router #2 gesendet. Die Routing-Tabelle von Router #2 enthalte folgende Einträge:

Netz	Hops	Router	Host/Server	Hops	Router
#1	0	-	Host #1	0	-
#2	0	-	Host #2	1	#1
#3	1	#3	Host #3	3	#1
#4	1	#1	Server #1	1	#1
#5	1	#3	Server #2	2	#1
#6	2	#3			
#7	2	#1			

Abbildung 10 Routing-Tabelle von Router #2

Gemäss dieser Routing-Tabelle wird die Datei also an den Router #1 gesendet. Dieser würde dann die Datei über das Netz #4 zum Router #4 senden und dieser würde dann die Datei an den Server #2 weiterreichen. Wir nehmen nun an, dass der Router #1 defekt sei und keine Antwort mehr gibt. Er fällt also aus. Der Router #2 probiert also mit dem Router #1 zu kommunizieren. Doch er erhält keine Antwort. Router #3 sendet nun seine Routing-Tabelle an Router #2. Dieser passt nun seine Routing-Tabelle wie folgt an:

Netz	Hops	Router	Host/Server	Hops	Router
#1	0	-	Host #1	0	-
#2	0	-	Host #2	2	#3
#3	1	#3	Host #3	3	#3
#4	2	#3	Server #1	2	#3
#5	1	#3	Server #2	2	#3
#6	2	#3			
#7	2	#3			

Abbildung 11 Routing-Tabelle von Router #2

Die Datei wird dann an den Router #3 gesendet. Dieser wird dann, wenn der Router #4 einwandfrei funktioniert, die Datei an den Router #4 liefern und dieser reicht dann die Datei an den Server #2 weiter. Problematisch wird es dann, wenn die Datei anstelle von Server #2 an den Server #1 geliefert werden soll. Diese Aufgabe kann gar nicht erfüllt werden, da Router #1 defekt ist. Router #2 wird dann versuchen, die Datei an Router #3 zu schicken. Dieser wird die Datei an Router #4 weiterschicken. Router #4 kann die Datei nicht an Router #1 senden, also wird Router #4 die Datei an Router #3 zurückschicken. Denn der Weg von Router #4 nach Server #1 ist kürzer als der Weg über Router #6. Die Folge ist ein ewiges Hin- und Herschicken. Dies kann im Extremfall dazu führen, dass das Netzwerk zusammenbricht. Denn wenn mehrere Dateien an den Host #2 oder an den Server #1 gesendet werden, dann zirkulieren immer mehr Dateien im Netzwerk und verstopfen dieses zunehmend. Man bezeichnet dieses Phänomen als *Counting to Infinity*. Damit dies nicht passiert, beinhaltet RIP einen Schutzmechanismus namens *Split Horizon*:

- Wird die Datei von einem Router A zu einem Router B gesendet, so kann sie nicht mehr an den Router A zurückgeschickt werden.
- Wird der Hopzähler grösser als 15, so wird die Datei gelöscht.

Bis die Datei gelöscht ist, vergehen meist über 7min. Dies ist eine sehr lange Zeit. Nun wird auch ersichtlich, warum RIP nur für kleine Netzwerke möglich ist. Der Grund liegt im Hopzähler. Dieser lässt nur Werte bis maximal 15 zu. Bei grossen Netzwerken gibt es aber meist mehr als 15 Zwischenstationen bis zum Ziel. Wichtig ist auch die Einsicht, dass der Router immer aufgrund der Hopzahlen entscheidet. Kleinere Hopzahlen bedeuten kürzere Wege. Daher wird jeder Router immer die kleinsten Hopzahlen auswählen und die Daten an den jeweiligen Router, der kleinere Hopzahlen zur Folge hat, senden. Würde der Router #1 wieder funktionieren, und müsste Router #2 eine Datei an den Host #2 liefern, dann würde, sofern die Routing-Tabelle von Router #1 rechtzeitig bei Router #2 eintrifft, die Datei an Router #1 gesendet, da der Eintrag Host #2,2Hops,Router #3 schlechter ist als der Eintrag Host #2,1Hop,Router#1.

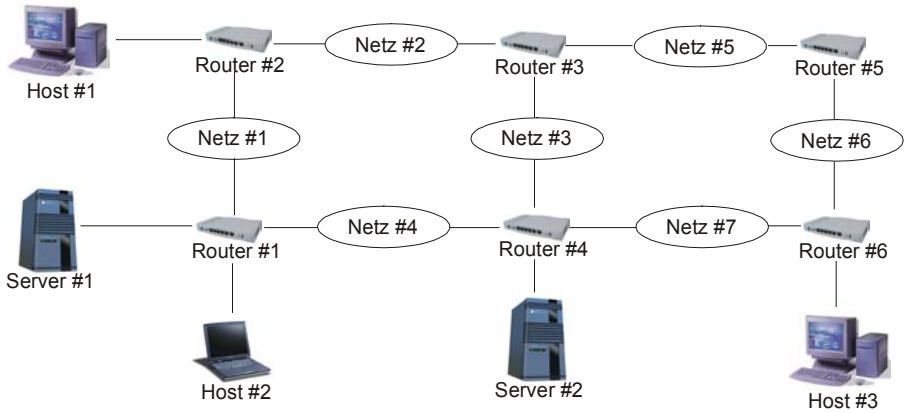

Abbildung 12 Netzwerk (Beispiel)

3.2 Link-State-Routing

Beim Link-State Routing besitzt jeder Router nicht nur Informationen über seine unmittelbare Umgebung, wie dies beim Distanz-Vektor-Routing der Fall ist, sondern besitzt alle Informationen über die gesamte Topologie des Netzwerks. Dazu zählen nicht nur Distanzangaben, sondern auch Angaben über Durchsatz, Verzögerungen, Zuverlässigkeit, finanzielle Kosten, usw.. Man bezeichnet solche Angaben als Routing-Metriken. Link-State-Routing ist ebenfalls ein statisches Routing-Verfahren, da die Verbindungsinformationen zwischen den einzelnen Routern, Servern, usw. zu Beginn bekannt sein müssen. Die Berechnung der gesamten Netztopologie erfolgt allerdings zur Laufzeit. Open Shortest Pass First (OSPF) ist ein Vertreter des Link-State-Routing-Verfahrens.

3.2.1 Open Shortest Pass First (OSPF)

Die Verknüpfungen im Netzwerk wird in den Routern mittels Baumdiagramm dargestellt, ähnlich wie man dies auch in vielen Dateisystemen verwendet. Der jeweilige Router bildet dabei die Wurzel des Baumes. Für die Berechnung der Netztopologie wird das Hello-Protokoll zur Hilfe genommen. Das Hello-Protokoll arbeitet ähnlich wie der Befehl ping. Ein Router fragt alle 30s (oder kürzer) andere Router im Netz an. Dabei werden sogenannte Hello-Nachrichten versendet. Die Laufzeit von solchen Nachrichten ist oft ein gutes Mass für die Güte einer Übertragung. Dadurch können auch Engpässe im Netzwerk lokalisiert werden. Beispielsweise ein langsamer Router hat längere Verzögerungszeiten und wird deshalb weniger bedient als schnellere Router. Es ist auch möglich eine effiziente Kosten-Nutzen-Kalkulation einfliessen zu lassen. Allerdings muss dem Routing-System bekannt gegeben werden, nach welchem Parameter die Optimierung erfolgen soll. Durch das Hello-Protokoll erkennt ein Router zudem, ob ein anderer Router bereit ist. Danach erfolgt in der Regel der Informationsaustausch zwischen den Routern. Die Routing-Metriken werden ausgetauscht. Der Vorteil von OSPF liegt in der Dynamik der Verbindungen. Hello-Protokolle werden häufiger versendet als Routing-Tabellen beim RIP. Zudem wird nicht nur die Distanz zwischen Routern, Servern, usw. im Netz betrachtet, sondern Zeitverzögerungen miteinkalkuliert. Damit wird es auch möglich, Aussagen über die Zuverlässigkeit oder Funktionstüchtigkeit eines Routers abzugeben. Das Phänomen *Counting to Infinity* ist hier nicht möglich, da das Routing-System schnell bemerkt, dass ein bestimmter Router nicht angesprochen werden kann. Hängt an diesem defekten Router ein Host, dann merkt das Routing-System, dass dieser Host nicht erreichbar ist. Daten, die an diesen Host adressiert sind, können vom System gelöscht werden. Aus all diesen Gründen eignet sich OSPF vor allem für grosse Netzwerke. Allerdings hat dieses Verfahren auch einen entscheidenden Nachteil. Die Berechnung der Netztopologie ist im Vergleich zu RIP aufwendig und komplex. Die Routing-Tabellen sind grösser als bei RIP. Dementsprechend ist auch der Informationsaustausch zwischen den Routern grösser.

4 Probleme mit IPv4

Obwohl IPv4 sich sehr bewährt hat, wird der Ruf nach einem neuen Internet Protokoll immer lauter. Die Erfinder des Internet Protokolls zu Beginn der achziger Jahre konnten ja nicht ahnen, dass ihr Protokoll heute von Millionen von Menschen rund um unseren Erdball tagtäglich genutzt wird. Und es werden immer mehr Benutzer! – Nicht nur die Fülle der Informationen hat enorm zugenommen, sondern auch die Menschen, die das Internet nutzen. Schliesslich haben auch neue Medien ins Internet Einzug genommen. Man denke nur an Internettelefonie, Internetradio, Internetfernsehen, u.a.. Alle diese Medien müssen in Echtzeit bearbeitet werden. Es dürfen praktisch keine Unterbrüche erfolgen. Höhere Datenraten werden gefordert. Die Benutzer entdeckt gerade die Vorzüge von Wireless-LAN und damit wird die Forderung nach einem mobilen Internet Protokoll (Mobile IP) laut.[4] Kurzum, eine enorme Steigerung der Datenverarbeitung wird die Zukunft des Internets entscheidend bestimmen. Dies hat aber direkte Einflüsse auf das Internet Protokoll. – Alle diese Einblicke lassen erahnen, dass das jetzige Internet Protokoll dem nicht mehr das Wasser reichen kann. – Im folgenden werden einige Problempunkte von IPv4 behandelt. Mehr zum Thema ‚Probleme mit IPv4' findet man im Internet.[10]

[4] Siehe auch Beitrag ‚Mobile IP'.

4.1 Ausgehende Adressen

Die Quell- und Zieladresse (Source and Destination Address) des Internet Headers haben eine Länge von 32Bit. Daraus ergeben sich 2^{32} mögliche Adressen. Diese Anzahl wird aber noch verkleinert durch die Existenz von Adressklassen. Das Schaffen von solchen Klassen führte also nicht nur zur logischen Adressgebung[5], sondern auch zu einer schlechten Nutzung der Adressen. Deshalb wird in der nächsten Generation des Internet Protokolls 128Bit verwendet. Auch eine neue Aufteilung des Klassenschemas ist denkbar.

4.2 Unterschiedliche Daten

Bei IPv4 wird zwischen gewöhnlichen Daten und mutlimedialen Daten nicht unterschieden. Wenn man also über das Internet beispielsweise telefoniert, dann wird das gesprochene analoge Signal digitalisiert und wie normale Daten über das Internet verschickt. Das Netzwerk weiss nicht, dass es sich um Daten aus einer Internettelefonverbindung handelt. Dementsprechend wird das Netzwerk auch nicht besondere Vorkehrungen treffen; beispielsweise eine niedrige Verzögerungszeit, keine Unterbrechungen, usw.. Die Tabelle 5 zeigt einige wesentlichen Unterschiede zwischen verschiedenen Datentypen. IPv6 könnte solche Eigenschaften im Internet Header als Flag abspeichern. Damit könnten die Daten nicht nur unterschieden, sondern auch – und das ist ja das Ziel – datenspezifisch übertragen werden.

Attribut	**Telefonie**	**Daten**	**Video**
Bandbreite	niedrig	variiert	hoch
Verzögerungstoleranz	niedrig	variiert	mittel
Fehlertoleranz	hoch	niedrig	mittel bis niedrig (wenn komprimiert)
Unterbrechungen	keine	viele	keine oder viele (wenn komprimiert)

Tabelle 7 Anforderungen an verschiedene Datentypen

4.3 Sicherheit

Sicherheit wird überall dort gefordert, wo sensitive Daten vorliegen, die man in der Regel nur innerhalb einer kleinen Gruppe austauschen möchte. Das bekannteste Beispiel ist vermutlich E-Commerce. Zahlungsaufträge, Daueraufträge oder andere Geschäftstätigkeiten können einfach via Mausklick von zuhause aus via Internet erledigt werden. Die heutigen Sicherheitskonzepte sind auf der Applikationsschicht angesiedelt. Protokolle wie SSL oder HTTPS[6] dienen dazu, die nötige Sicherheit zu erhöhen. Doch können sie sie auch gewährleisten? – IPv6 soll schon auf der Netzwerkebene, also auf der Basis des Internet Protokolls, zahlreiche Sicherheitseigenschaften erfüllen. Damit hält eine Art Sicherheitsschicht schon bei der Netzwerkschicht Einzug; und nicht erst auf Applikationsebene. Dies würde es Hackern nahezu unmöglich machen, auf solche Daten zuzugreifen, da diese dann mit einer Art ‚Sicherheitssignierung' versehen wären.

[5] Server mit vielen Hosts haben eine andere Adressklasse als solche mit mittleren bis wenig Hosts.
[6] Siehe Beitrag ‚SSL, SHTTP-sichere Kommunikation'.

4.4 Dynamic Host Configuration Protocol (DHCP)

Portable Geräte der Zukunft werden vermehrt das Internet verwenden. Dies bedingt aber eine drahtlose Kommunikation zwischen einem dieser Geräte und einer Sendestation, die mit einem Server kommuniziert. Das Problem beginnt dann, wenn man sich mit diesem Gerät in einem Gebiet mit mehreren Sendern bewegt. Beim Verlassen eines Sendebereichs muss schon die Verbindung mit einem neuen Sender aufgebaut sein. Es muss also eine Koordination zwischen dem alten und dem neuen Sender erfolgen. Der alte Sender wird verlassen und der neue Sender verwendet. Diese Art des mobilen Internet Protokolls (kurz mobile IP[7]) bedarf auch einer Erweiterung des bisherigen Standards. Der Host muss dann dynamisch den Sender wechseln und somit seine Konfiguration ändern. Deshalb bezeichnet man dieses Protokoll auch Dynamic Host Configuration Protocol (kurz DHCP).

5 Schlusswort

Die Anforderungen an das Internet werden in Zukunft markant steigen. Nur ein ausbaufähiges, flexibles aber auch robustes Internet Protokoll kann dem Parole bieten. Das Internet Protokoll bleibt eines der wichtigsten Kernelemente der Internet-Technologie. Seine Entwicklung wird die Entwicklung des Internets massgeblich prägen. Das IPv6 wird in ein paar Jahren auf allen Netzwerken zum neuen Standard erhoben. Sein Gelingen entscheidet massgeblich über die Weiterentwicklung des Internets. Hoffen wir, dass dieser neue Standard das ist, was seine Anhänger propagieren: ein flexibleres Internet Protokoll, das dem Benutzer mehr Möglichkeiten bietet, aber mit der Zuverlässigkeit von IPv4 agiert.

Referenzen

[1] http://www.faqs.org/rfcs/rfc791.html
[2] http://www-mm.informatik.uni-mannheim.de/veranstaltungen/animation/isoosi/isoosi1/
[3] http://www.ussg.iu.edu/usail/network/nfs/network_layers.html
[4] http://www.linux-praxis.de/linux2/ipaddr.html
[5] http://www.it-academy.cc/content/article_browse.php?ID=10
[6] http://public.pacbell.net/dedicated/cidr.html
[7] http://www.switch.ch/id/search-domain.html
[8] http://www.denic.de/servlet/Whois
[9] http://www.allwhois.com
[10] http://www.acm.org/crossroads/columns/connector/august2000.html

[7] Siehe Beitrag ‚Mobile IP'.