

# Routing in IP Netzen

Felix von Leitner  
Chaos Computer Club Berlin  
felix@ccc.de

Chaos Communication Congress 2000

## Zusammenfassung

Routing beantwortet die Frage: *Wieso kommt mein Paket nicht an?*

## Wie spricht man das eigentlich aus?

*[rauting]* und *[ruting]* sind beide richtig. Amerikaner benutzen gerne letzteres, Europäer gerne ersteres.

## Router und Routing-Protokolle

Eine **Route** ist eine Information, die einem Gerät sagt, wie es ein Paket in ein bestimmtes Ziel-Subnetz senden kann. Eine Route heißt **asymmetrisch**, wenn die Rückroute einen anderen Pfad benutzt.

Ein **Router** ist ein Gerät, das zwei Subnetze verbindet. Geräte in beiden Subnetzen geben ihre Pakete für das jeweils andere Subnetz beim Router ab.

Ein **Routing-Protokoll** ist ein Mechanismus für das dynamische Entdecken der Pfade für Datenpakete durch das Internet.

Ein **Hop** ist ein Gerät auf dem Pfad zwischen zwei Geräten.

## IPv4-Adressen

- Eine IPv4-Adresse ist ein 32-bit Integer
- Wird gewöhnlich in der Form **192.168.17.23** geschrieben (192 ist das höchstwertigste Byte)
- Im IP-Header steht sie in der *network order* (d.h. big-endian)
- IPv4-Adressen sind im Internet eindeutig

## IPv6-Adressen

- Eine IPv6-Adresse ist ein 128-bit Integer
- Teil der IPv6-Adresse ist die (eindeutige) MAC-Adresse

**Alle Betrachtungen in diesem Vortrag sind bezüglich IPv4!**

## Was ist eine Netzmaske?

Die Netzmaske wird benutzt, um die Netzadresse zu berechnen. Dafür wird eine IP-Nummer mit der Netzmaske logisch **AND**-verknüpft.

Beispiel:

IP-Nummer	192	.	168	.	17	.	23
Netzmaske	255	.	255	.	0	.	0
Netzadresse	192	.	168	.	0	.	0

Tabelle 1: Netzadresse berechnen

## Wozu braucht man die Netzadresse?

Zwei IP-Nummern sind im gleichen Subnetz, wenn sie bezüglich der gleichen Netzmaske die gleiche Netzadresse haben.

Per Konvention haben Netzmasken die Binär-Form  $1*0*$ , d.h. die Einsen sind alle linksbündig, aber technisch ist das nicht notwendig. Wenn Netzmasken in dieser Form sind, schreibt man sie auch in der Form  $10.0.0.0/8$  (für die Netzmaske 255.0.0.0).

Eine Netzadresse und eine Netzmaske zusammen bestimmen ein **Subnetz** eindeutig.

Ein Subnetz und eine IP-Adresse zusammen ergeben eine **Route**, wobei die IP-Adresse die des **Gateways** ist. Das Gateway ist ein Gerät, das einen **Hop** näher am Ziel ist. Lokale Netze (d.h. solche, bei denen der Rechner selbst Mitglied ist) haben kein Gateway.

Eine Route mit einer Netzmaske von 0 wird **default** Route genannt, weil sie bei allen Ziel-Adressen zutrifft.

## Was ist denn ein Subnetz?

Es gab ursprünglich nur 5 Klassen von Netzen:

Klasse	IP-Bereich	Netzmaske
Class A	0-127.*	255.0.0.0
Class B	128-191.*	255.255.0.0
Class C	192-223.*	255.255.255.0
Class D (multicast)	224-239.*	255.0.0.0
Class E (reserviert)	240-255.*	255.0.0.0

Tabelle 2: Netzklassen

Man kann natürlich andere Netzmasken benutzen. Die sich ergebenden Netze heißen (aus historischen Gründen) Subnetze. Heute ist diese Trennung unwichtig und man benutzt Netz und Subnetz synonym.

## Microsoft und Routing

Bei Microsoft hat man das mit den Subnetzen noch nicht so verstanden, weshalb man bei Windows auch ein Default Gateway angeben kann, das *nicht erreichbar ist*. Beispiel: Host 10.2.3.5, Netzmaske 255.255.255.0, Router 10.17.23.42.

Daher ist es für Laptop-User in Windows-Netzen manchmal schwierig, eine gültige IP-Konfiguration zu bekommen.

**Fix:** eine statische Host-Route zum Gateway manuell setzen.

## Cisco und Subnetze

Manche Hersteller optimieren die (sequentielle) Suche in großen Routing-Tabellen, indem sie nur die Netzadressen vergleichen, was bei **10.0.0.0/8** und **10.0.0.0/24** u.ä. nicht funktioniert.

Bei Cisco läßt sich dieser Bug mit **ip subnet-zero** korrigieren.

## Was ist eine Routing-Tabelle?

Jedes IP-Gerät hat eine Liste von Routen. Wenn ein Paket verschickt werden soll, wird für alle Routen geprüft, ob die Ziel-Adresse in dem entsprechenden Netz liegt. Wenn ja, wird es an die IP des Gateways der Route geschickt.

Damit nicht immer die Default-Route angewandt wird, sind Routen noch mit einer **metric** gewichtet. Eine höhere Metrik steht für eine teurere Route. Die Default-Route hat deshalb gewöhnlich Metriken ungleich Null.

Das kann aber auch benutzt werden, um Backup-Routes über langsamere Geräte zu definieren, die nur benutzt werden, wenn die Hauptroute ausfällt.

Zusätzlich steht bei jeder Route noch das Netzwerk-Interface dabei, über das das Paket mit der IP-Nummer des entsprechenden Gateways gesendet werden soll. Diese wird normalerweise anhand früherer Routen und der Gateway-IP bestimmt.

## Des Pudels Kern

Dieser Vortrag behandelt die Frage, wo der Inhalt der Routing-Tabelle herkommt, bzw. wie man Geräte überredet, anders als in der Routing-Tabelle beschrieben zu routen.

Tatsächlich interessiert den Angreifer gewöhnlich nur die Rück-Route, d.h. man möchte gewöhnlich erreichen, daß Pakete bei einem Router vorbeigeroutet werden, auf den man privilegierten Zugriff hat.

## Statisches Routing

Die einfachste Methode für das Bevölkern der Routing-Tabelle sind statische Routen, d.h. von Hand eingetragene. Die Routen für die lokalen Netzwerk-Interfaces müssen immer statisch definiert werden.

Normalerweise können Routing-Protokolle keine statischen Routen ändern.

## Dynamisches Routing

Dynamisches Routing heißt, daß die Routing-Tabelle über das Netzwerk modifizierbar ist. Verbreitete Protokolle dafür sind

- ICMP Redirect
- RIP (Routing Information Protocol)
- OSPF (Open Shortest Path First)
- BGP (Border Gateway Protocol)
- IGRP (Interior Gateway Routing Protocol)
- EIGRP (Enhanced IGRP)

## Was kann man alles falsch machen?

Das offensichtliche Routing-Protokoll wäre, daß alle Rechner periodisch ihre Routen broadcasten und die bekannten Routen mit einem Timeout belegen.

- Yoyo-Effekt
- Routen verschwinden nicht
- Falschrouten propagieren sich

## Yoyo-Effekt

1.
  - Rechner A hat eine Leitung zu Rechner C
  - Rechner A hat eine Route über diese Leitung, Metrik 1
  - Rechner B kennt die Route mit Gateway A und Metrik 2
2. Die Leitung bricht zusammen, Rechner A löscht die Route
3. Rechner B nennt Rechner A die Route
4. Rechner A kennt jetzt eine Route über die tote Leitung mit Rechner B als Gateway und Metrik 3

## Routen verschwinden nicht

Wenn ein Rechner seine Route löscht, bekommt er sie umgehend von einem Nachbarn mit dem nächsten Broadcast wieder mitgeteilt.

- Man könnte periodisch die Metrik verschlechtern bei den alten Routen. Dann würden Schrott-Routen irgendwann schlechter als die Default-Route.
- Man könnte verbieten, daß Routen an den Rechner zurückgemeldet werden, von dem sie kommen. Damit verhindert man aber keine größeren Schleifen (z.B. drei Rechner reichen eine Route im Kreis)
- Man könnte negative Routing-Meldungen einführen, d.h. eine „bitte löschen“ Meldung. Die müßte auch nach einer Weile gelöscht werden, sonst akkumulieren sich Müll-Broadcasts.

## Warum überhaupt dynamisch routen?

- Automatisch auf Backup-Pfad umschalten
- Besten Pfad aus mehreren Alternativen wählen
- Last auf mehrere Pfade verteilen (nur dedizierte Router)
- Weniger administrativer Aufwand

## Source Routing

Es gibt im IP-Header die Möglichkeit, ein paar Hops der Rückroute zu spezifizieren, und dem Zielrechner zu sagen, daß die Rückroute doch bitte so aussehen soll. Weil dafür praktisch keine legitimen Gründe für die Anwendung existieren, lassen praktisch alle Firewalls und die meisten Router solche Pakete auf den Boden fallen.

Mit Source-Routing kann man mit **netcat** herumspielen.

## ICMP Redirect

ICMP ist das IP-Protokoll für Kontrollnachrichten. Es wird für *ping* und für *port unreachable*, *network unreachable*, *host unreachable* und eben *redirect* benutzt.

Mit der letzten Nachricht kann der Gateway einem Host sagen, daß er (im lokalen Netz) doch nicht zuständig ist und für das Ziel doch lieber dieser andere Gateway zuständig wäre.

Leider akzeptieren manche Implementationen auch ICMP redirect Pakete von anderen Rechnern als dem zuständigen Gateway (Router inzwischen nicht mehr). ICMP redirect verstehen auch Nicht-Router, Router unterhalten sich anders.

ICMP redirect Pakete kann man mit libnet von Hand basteln, mit **sendip** und es gibt auch Perl-Module für den Hobby-Paketbastler.

## Klassen von Routing-Protokollen

Man unterscheidet zwischen *internen* und *externen* Routing-Protokollen. Erstere werden innerhalb von *autonomen Systemen* benutzt, letztere verwalten den Verkehr zwischen diesen.

Autonome Systeme sind gemeinsam administrierte Netze mit mehr als einem Zugang zum Internet. Normalen Administratoren begegnen nur Protokolle vom Typ *intern*. Autonome Systeme werden anhand einer 16-bit Zahl unterschieden, die man offiziell beim InterNIC beantragen muß.

Das meistgenutzte *interne* Protokoll ist RIP, man kann aber auch OSPF benutzen.

## Routing Information Protocol

- RIP wurde beim XEROX PARC entwickelt und 1981 für IP formal definiert (RFC 2453; auch 1058, 1388, 1723)
- Es definiert ein Format, mit dem man sagen kann, daß man eine Route zu Rechner XY mit Metrik Z kennt, oder fragen kann, ob jemand eine Route zu Rechner XY kennt
- Nachrichten gehen nur an Nachbarn im LAN
- Die Metrik einer Route wird beim Import inkrementiert, d.h. sie entspricht dem Hop-Count
- Das nennt man „distance vector“
- Nachrichten sind trivial spoofbar
- RIP 2 definiert ein Paßwort-Feld, das natürlich auch snoopbar ist
- RIP definiert ein paar triviale Heuristiken, mit denen Loops verhindert werden sollen
- Der BSD routed benutzt RIP

## RIP Heuristiken

- Hops sind limitiert auf 15, d.h. unbenutzbar für sehr große Netze und das ganze Internet
- Hold-Down (Routen werden nicht gelöscht, sondern als gelöscht markiert und eine Weile eingefroren)
- Split Horizon (Routen nicht an deren Gateway propagieren)
- Poison Reverse Updates (Beim Nachbarn Routen mit mehr Hops löschen)

## RIP Spoofing

Man kann zwar RIP-Pakete einfach spoofen, aber sie werden nur innerhalb des eigenen autonomen Systems verbreitet und pro Hop wird die Metrik inkrementiert.

Es gibt keine negativen Metriken.

Spoofing geht also nur, wenn man näher am Absender sitzt als der eigentlich Empfänger.

## Open Shortest Path First

- Von IETF entwickelt, weil RIP nicht skalierbar genug war (RFC 2328; auch 1131, 1247, 1583, 2178)
- Schickt nicht nur eigene Routen und nicht nur an Nachbarn im LAN
- Jeder Router akkumuliert einen Graphen aus Routen
- In diesem Graphen wird die richtige Route mit Dijkstras Algorithmus gesucht
- Ein autonomes System kann aus mehreren „areas“ bestehen
- Router in mehreren Areas heißen „area border routers“ und halten einen Graphen pro Area
- Nachbarn werden mit dem *OSPF Hello protocol* gefunden, eine Art *ping*, das auch als *keepalive* benutzt wird
- OSPF unterstützt *type of service* Routing mit *delay*, *throughput* und *reliability* als mögliche Anforderungen
- *Equal cost multipath routing* (nur dedizierte Router)
- `gated` implementiert OSPF.

## **OSPF Spoofing**

**OSPFv2 definiert neben Null und Password auch MD5-basierte Authentisierung mit einem gemeinsamen Schlüssel. Außerdem: Sequence Number gegen Replay-Attacken.**

## Interior Gateway Routing Protocol

- Cisco-proprietäres „distance vector“ Protokoll
- Metrik-Vektor: *delay, bandwidth, reliability* und *load*
- Kann *multipath routing* (channel bundling)
- Push-Protokoll, ausgefallene Router werden an fehlenden Updates erkannt
- Zusätzlich zu den RIP-Heuristiken gibt es verschiedene Timer

## Enhanced IGRP

- Merkt sich die Routing-Tabellen aller Nachbarn
- Subnetzmasken variabler Länge
- Keine periodischen Updates, partielle on-demand Updates
- Periodische Hello-Pakete für Discovery
- *Reliable Transport Protocol (RTP)* sorgt für garantierte in-order Auslieferung der IGRP-Pakete

## **Spoofing von IGRP/EIGRP**

**Gute Frage. Wer hat eine Cisco und Zeit zum Spielen?**

## Exterior Protocols

Externes Routing findet nur zwischen den Gateways von autonomen Systemen statt. Zum Spoofen muß man also auf so einen Router Zugriff haben.

Nicht-dedizierte Router benutzen gewöhnlich die **gated**-Implementation. Neuerdings wird für externe Protokolle auch Multicasting benutzt.

## Exterior Gateway Protocol

- Erreichbarkeit, nicht Routing, von 1984 (RFC 0904; auch 0827)
- Periodisches *Hello/I-Heard-You*
- Polling
- Definiert über einen endlichen Automaten
- Nachrichten nur zwischen je zwei direkten Nachbarn
- EGP gibt auch Routen anderer Leute weiter
- Router werden direkte Nachbarn mit einem 3-way handshake
- Direkte Nachbarn pollt man periodisch nach deren Routen

EGP definiert nicht, wie man auf die Nachrichten reagieren soll, oder daß man die Routing-Tabelle überhaupt anfassen soll.

## Border Gateway Protocol

- Ein „Exterior Gateway Protocol“ von 1989 (RFC 1771; auch 1105, 1163, 1267, 1654)
- Routing zwischen autonomen Systemen
- Nachfolger des RFC0904-EGP
- Kann Routing-Loops erkennen
- Keine periodischen Table-Broadcasts
- BSP merkt sich für alle Peers deren aktuelle Routing-Tabelle
- Inkrementelle Update-Nachrichten melden jeweils den optimalen Pfad, nicht alle Pfade
- Pakete beinhalten 16 Bytes „Marker“ für Authentisierung und ein Feld, mit dem man den Authentisierungs-Algorithmus nennen kann, aber keine Algorithmen sind definiert
- Für die Verbindungen wird TCP benutzt

## **BGP Sicherheit**

**BGP bemüht sich zwar um Schutz vor Spoofing von außen, aber prüft die Routen nicht weiter. Wer also einen externen Router administriert, der BGP-Zugang hat, kann beliebige Netze aus dem Internet entfernen, indem er Routen der Metrik 0 für diese Netze announced.**

## Routing Performance: Tag Switching

Tag Switching erhöht die Routing-Performance drastisch.

Idee: man hängt ein Schildchen mit der Gateway-Nummer an jedes Paket.

Router, die so ein Schildchen sehen, nehmen die Nummer als Index in einer Tabelle, in der das Gateway und eine neue Nummer stehen.

Das Paket geht mit einem neuen Schildchen mit der Nummer aus der Tabelle an den Gateway aus der Tabelle.

Das Schildchen implementiert man über das IPv6 Flow Label, über ATM Tags oder über einen kleinen Zwischen-Header zwischen Layer 2 und 3.

## Welches Protokoll soll man denn benutzen?

Intern: RIP. OSPF lohnt sich nur in großen Netzen und verlangt Platz und CPU auf dem Router für den Graphen.

Wenn man schon RIP benutzt, sollte man auf jeden Fall statische ARP-Einträge für die bekannten Router benutzen und IPsec o.ä. benutzen zwischen den Routern.

Extern: BGP. Niemand spricht extern etwas anderes.

## Freie Routing-Software: mrt

Das Merit Routing Toolkit spricht BGP, BGP+, RIP, RIPng, DVMRP, PIM und PIMv6.

RIPng ist RIP für IPv6, DVMRP und PIM sind Routing-Protokolle für Multicast-Traffic. PIMv6 ist PIM für IPv6 und BGP+ ist BGP für IPv6. mrt kann sowohl PIM Dense Mode als auch Sparse Mode.

Für die Anzahl der Features ist mrt relativ klein. Da mrt sowohl Threads als auch select() benutzt, ist der Code weder les- noch wartbar.

Sieht relativ mächtig aber unfreundlich zu bedienen aus.

Zu finden unter <http://www.mrtd.net/>. Kommt leider ohne Dokumentation (die ist im Web).

## Freie Routing-Software: GNU zebra

Zebra spricht BGP, BGP+, OSPF, OSPFv6, RIP und RIPng.

Kooperation von Japanern und Russen (d.h. Dokumentation, Fehlermeldungen und Kommentare sind unverständlich).

Typische GNU-Software (groß, verbose, #definiert Konstanten wie ZEBRA\_ROUTE\_OSPF, kommt mit zebra.el, hat Guile-Binding, aber man findet sich schnell im Code zurecht). Hat sogar eine IOS nachempfundene Konsole.

Zu finden unter <http://www.zebra.org/>.

## Freie Routing-Software: BIRD

BIRD spricht BGP, BGP+, RIP, RIPng und OSPF.

BIRD kommt mit der mit Abstand besten Dokumentation, sogar mit Programmierer-Referenz! Trotzdem ist es mit Doku noch kleiner als mrt ohne und benutzt keine Threads. Der Code ist vorbildlich lesbar.

Startete als Studentenprojekt (d.h. kein Grant wie mrt) und ist mit Abstand am sympathischsten, wenn auch am jüngsten.

Zu finden unter <http://bird.network.cz/>.

## Spoofing mit Routen

Um Traffic für `microsoft.com` bei `ccc.de` vorbeizurouten müßten wir:

1. Per BGP eine falsche Route mit Metrik 0 von `ccc.de` zu `microsoft.com` propagieren
2. Dafür sorgen, daß es keine kürzeren Routen zum Ziel gibt als unsere
3. Eine Möglichkeit haben, den Traffic dann auch tatsächlich zu Microsoft weiterzugeben

## Spooftng mit Routen

Der erste Teil erfordert, daß man Zugang zu einem externen Router hat.

Viele Protokolle löschen Routen per Timeout, nicht auf Zuruf. Bis sich der Timeout propagiert hat, können Stunden vergehen! Selbst wenn man also die BGP-Announcements des `microsoft.com`-Router deaktivieren kann, muß man auf den Timeout warten. Einfacher ist es, wenn man auf dem anderen Router eine Hintertür hat.

Der dritte Teil erfordert einen Tunnel zu einem Teil des Internets, der nahe genug an `microsoft.com` liegt, um die gespoofen Routen noch nicht erhalten zu haben. Auch alle Hops zwischen diesem Rechner und `microsoft.com` dürfen die falsche Route noch nicht erhalten haben!

## Hintertüren in Routern?

Die meisten Router-Hersteller haben Hintertüren und undokumentierte Admin-Zugänge, um im Notfall beim Kunden trotz vergessenem Paßwort den Router reparieren zu können. Bei einigen Herstellern sind solche Zugänge bereits gefunden worden:

- Cisco
- 3com
- Bay Networks

## Denial of Service

Normalerweise braucht man dafür keine Routing-Protokolle, aber ein sehr einfaches DoS ist natürlich das Propagieren von 100.000 neuen Routen. Weil das noch keiner mit böser Absicht gemacht hat, haben Router dagegen keine Schutzmaßnahmen eingebaut.

Ansonsten kann man `microsoft.com` vom Netz holen, wenn man BGP Routen announce kann.

Es gab Mitte 2000 das Gerücht, daß die Musikindustrie mit BGP Raubkopierer-Sites vom Netz holen wollte, aber das war wohl eine Ente.

## Was für ein Protokoll soll ich denn einsetzen?

- RIP ist relativ robust und man kommt nicht sehr weit
- BGP kann auf entfernten Routern Routen einfügen und löschen. Die Routen werden aber gekillt, sobald die TCP-Verbindung weg ist!
- BGP ist sehr leicht DoSbar, weil TCP-basiert.

Man kann bei Routing-Protokollen nur Routen spoofen, wenn `victim.com` von dem attackierten Router weiter weg ist

Man kann bei BGP also den anderen Router DoSsen und dann selber eine Route setzen. Das hilft aber nicht viel und fällt sofort auf.

## Zusammenfassung

Die Routing-Protokolle sind auf Robustheit ausgelegt, weil jede Komponente zu jedem Zeitpunkt ausfallen kann. Daher gibt es auch keinen zentralen Angriffspunkt.

Protokolle gehen immer davon aus, daß sie ungültige Routen angezeigt bekommen, die sie dann wegwerfen.

Spoofting ist schwierig, aber machbar.

**Danke für die Aufmerksamkeit!**

Fragen?

Diese (und andere) Folien gibt es auf <http://www.fefe.de/>.