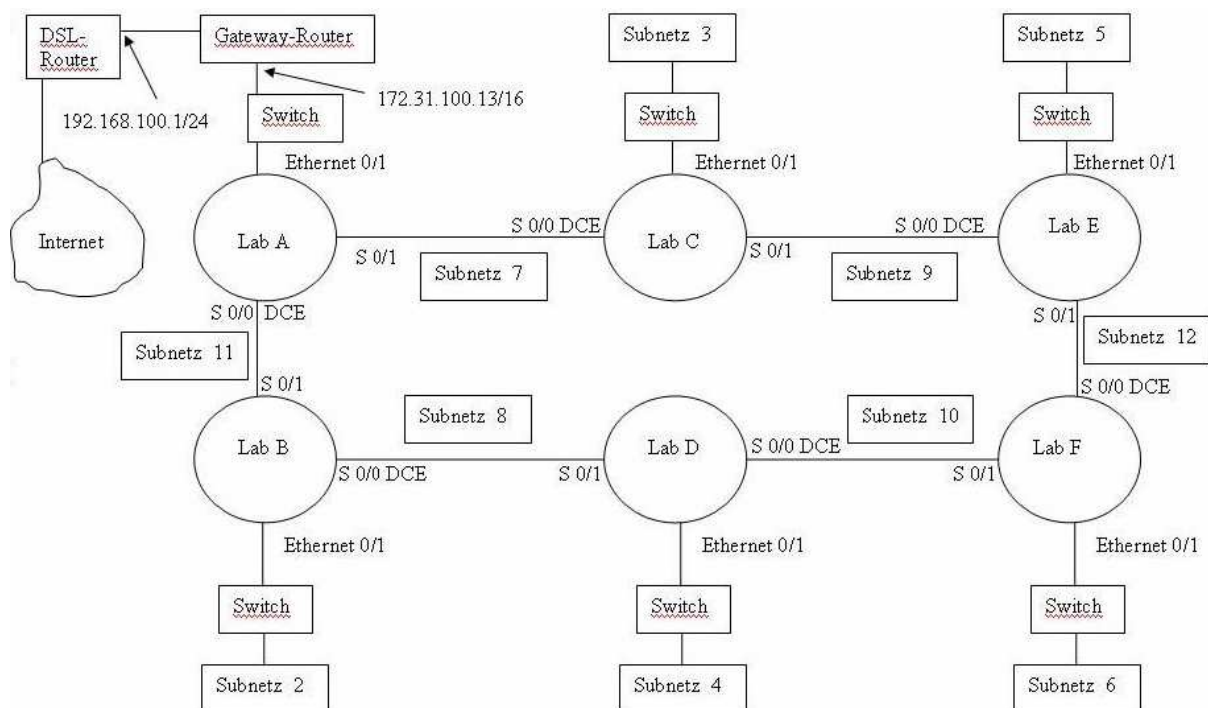


# Projekt

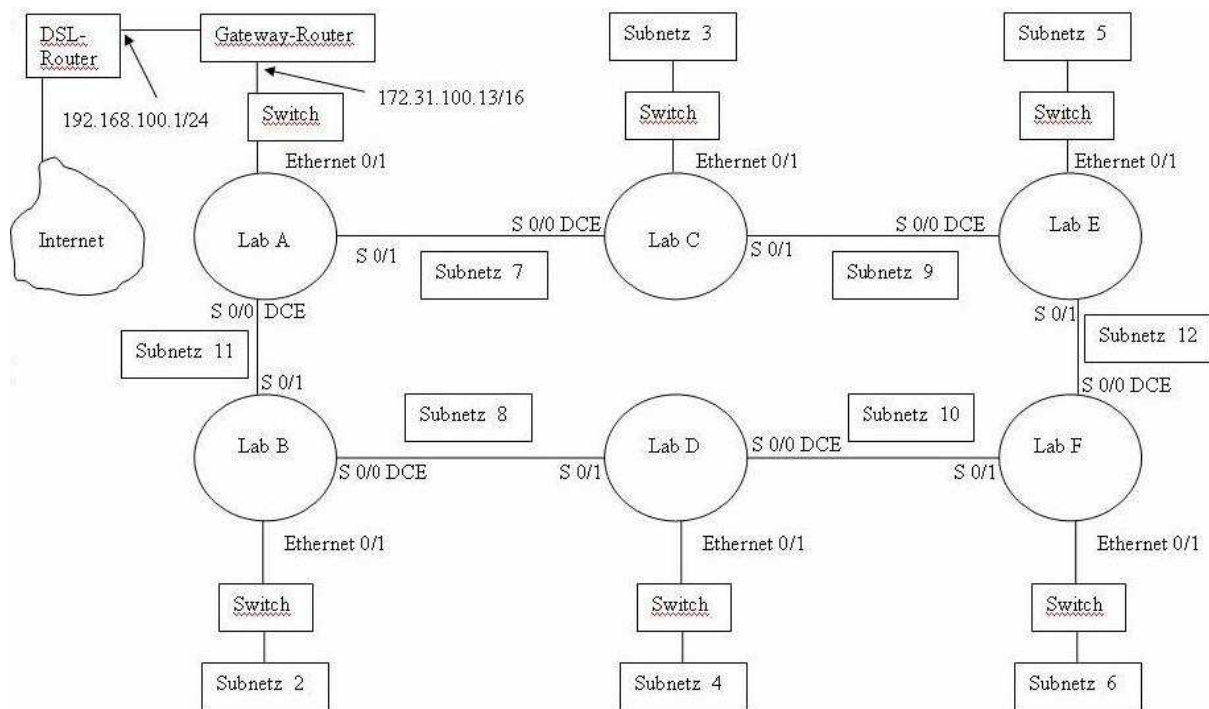
## Routende Netze mit RIP



## Grundkonfiguration

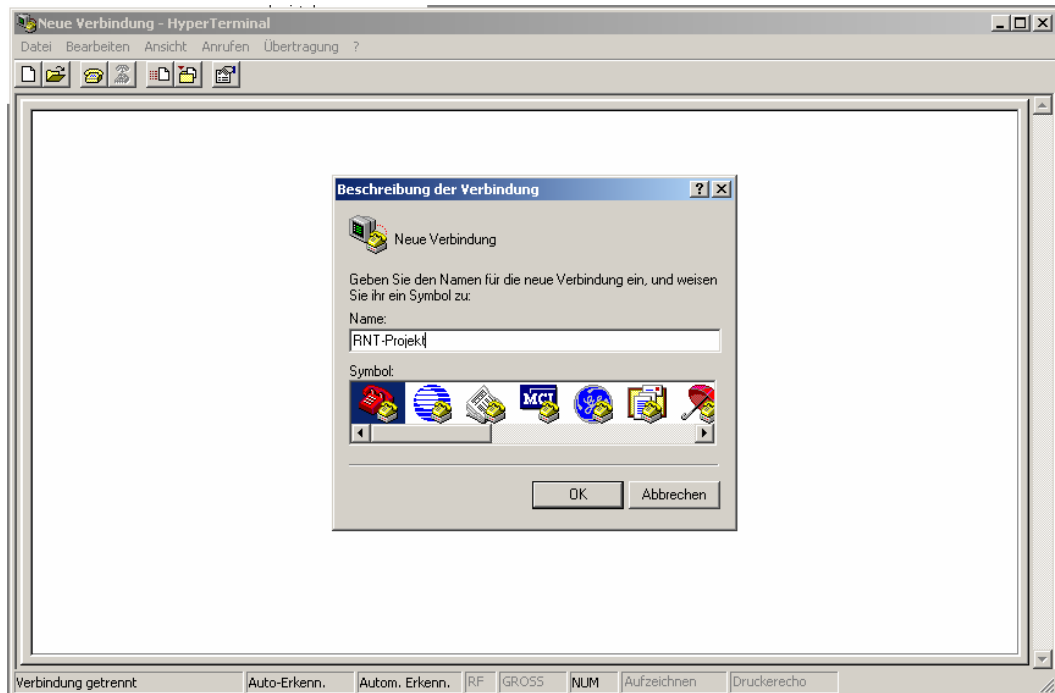
Um ein routendes Netz gemäß den Angaben des Netzplan aus Abbildung 1 zu auf zu bauen, sind zunächst die Router und PCs richtig zu verkabeln.

Abbildung 1:



Anschließend verbindet man sich mit den Routern über ein Rollover-Kabel mit dem COM-Port des Routers mit einer Terminal Software wie Hyperterminal die in Windows standardmäßig zu finden ist.

Hierzu gibt man unter Start → Ausführen den Befehl hypertrm.exe ein, der nun die Software Hyperterminal startet.



Beim Einstellen der Verbindung ist darauf zu achten, dass man die richtige Clockrate( in unserem Fall bei 56000) für den Router wählt.

Hat man sich nun mit dem Router Verbunden (hier LAB\_F) geht es weiter zu den wirklich interessanten Teil.

Damit man mit der Konfiguration beginnen kann, ist zunächst in den so genannten ena-Modus zu wechseln.

Anschließend ist für die Konfiguration in den globalen Konfigurationmodus zu wechseln.

Dort setzt man zunächst einmal den Hostnamen des Routers.

#### Befehl

```
Router>ena
```

```
Router# conf t
```

```
Router(config)# conf t
```

```
Router(config)# hostname LAB_F
```

Um den Router nicht ganz ungeschützt zu lassen, sollte für eine Telnet-Sitzung und das Erreichen des ena-Modus ein Passwort gesetzt werden.

Allerdings kann man das Passwort leicht zurücksetzen, sobald man lokalen Zugriff zu den Routern erhält.

Befehl
<i>LAB_F(config)# line vty 0 4</i>
<i>LAB_F(config-line)# login</i>
<i>LAB_F(config-line)# password cisco</i>
<i>LAB_F(config-line)# end</i>
<i>LAB_F(config)# line console 0</i>
<i>LAB_F(config-line)# login</i>
<i>LAB_F(config-line)# password cisco</i>
<i>LAB_F(config-line)# end</i>
<i>LAB_F(config)#</i>

Zur Erklärung, mit dem Befehl in Zeile 1 wird das Setzen eines Telnet-Passwortes eingeleitet. Anschließend legt man fest, dass der Login geschützt werden soll bzw. es um den Login geht. Nun wird das Passwort gesetzt, hier ist es immer *cisco*, aber im Produktiv-Einsatz sollte natürlich ein sichereres gewählt werden.

Anschließend wird das gleiche für den Konsolen Zugang gesetzt.

### **Serielle Schnittstellen einrichten**

Um die Seriellenschnittstellen einzurichten, ist im globalen Konfigurationmodus die entsprechende Schnittstelle (Interface) zu wählen, und diese mit einer IP-Adresse auszustatten und diese Administrativ „hochzufahren“, also zu aktivieren.

Dabei ist es wichtig das selbiges auch von den angrenzenden Routern ebenfalls gemacht wird, ansonsten besteht keine Verbindung.

Weiterhin legt die Serielle-Schnittstelle Serial0/0 die Clockrate vor und die Serielle-Schnittstelle Serial0/1 nimmt die von der Verbundenen Serielle-Schnittstelle Serial0/0 des Router-Nachbarn an.

Anschließend ist die Konfiguration der Seriellen Schnittstellen abgeschlossen.

Befehl
<i>LAB_F(config)# ip routing</i>
<i>LAB_F(config)# interface serial0/0</i>

```
LAB_F(config-if)#ip address 192.168.12.2 255.255.255.0
LAB_F(config-if)# clock rate 56000
LAB_F(config-if)# no shutdown
LAB_F(config-if)# end
LAB_F(config)# interface serial0/1
LAB_F(config-if)#ip address 192.168.10.1 255.255.255.0
LAB_F(config-if)# no shutdown
LAB_F(config-if)# end
LAB_F(config)#
```

Zur Erläuterung, mit dem Befehl in Zeile 1, wird erstmal IP-Routing auf dem Router ermöglicht.

Anschließend ruft man den Konfigurationsmodus für die Schnittstellenkonfiguration auf. Hier beginnen wir mit der Seriellenschnittstelle Serial0/0, dieser weisen wir zu erst einmal eine IP-Adresse zu in der Form *ip address <ip-adresse> <subnetzmaske>*, dabei muss das ganze dotted decimal erfolgen.

Danach wird die Serial0/0-Schnittstelle als DCE (engl. Data Communication Equipment ; de. Datenübertragungseinrichtung -DÜE ; einfach: taktgebene Seite) aktiviert.

Abgesehen davon, wird das gleiche noch für die zweite Serielle-Schnittstelle gemacht.

Abschließend wird dann noch der administrative, standardmäßig eingeschaltete Shutdown der Schnittstelle aufgehoben.

### **FastEthernet-Schnittstelle einrichten**

Um nun die FastEthernet-Schnittstelle des Routers zu konfigurieren ist zu nächst in den globalen Konfigurationsmodus zu wechseln( dies macht man mit *conf t*).

Aus diesem wechselt man in den Konfigurationsmodus für die Schnittstellenkonfiguration, hier der FastEthernet-Schnittstelle.

Ach hier wird dieser Schnittstelle zu erst einmal eine IP-Adresse zugewiesen in der Form *ip address <ip-adresse> <subnetzmaske>*, dabei muss das ganze auch wieder dotted decimal erfolgen.

Abschließend wird dann noch der administrative, standardmäßig eingeschaltete Shutdown der Schnittstelle aufgehoben.

Befehl
<pre>LAB_F# conf t LAB_F(config)# interface FastEthernet0/0 LAB_F(config-if)# ip address 198.1.1.81 255.255.255.240 LAB_F(config-if)# speed auto LAB_F(config-if)# no shutdown LAB_F(config-if)# end LAB_F(config)#</pre>

Mit dem Befehl *sh ip int brief* kann man sich den Zustand der Schnittstellen überprüfen.

## **RIP (Routing Information Protocol)**

### Theorie

Es gibt dynamische Routen und statische Routen. Statische Routen sind Routen, die durch einen Administrator festgelegt werden.

Dynamische Routen sind hingegen festgelegte Wege/Routen/Pfade vom Router, durch bestimmte Protokolle.

RIP (Routing Information Protocol) ist ein Routing-Protokoll basierend auf den Distanzvektoralgorithmus (auch Bellman-Ford-Algorithmen).

Es ist für manche Netzadministratoren attraktiv, weil es auf offenen Standards basiert und sehr leicht implementiert werden kann, obwohl es nicht über die Leistungsfähigkeit und die Funktionen anspruchsvollerer Routing-Protokolle verfügt.

Aufgrund seiner Einfachheit ist RIP ein gutes Anfangsprotokoll für all jene, die sich noch nicht lange mit Netzen beschäftigen.

Routing-Tabellen werden in regelmäßigen Abständen oder nach einer Änderung der Topologie in einem Netz aktualisiert, in dem ein Distanzvektor-Protokoll verwendet wird. Es ist wichtig, dass ein Routing-Protokoll die Routing-Tabellen effizient aktualisiert. Wie beim Prozess der Netzerkennung, werden Updates zu Änderungen der Topologie systematisch von Router zu Router weitergeleitet. Distanzvektor-Routing-Algorithmen fordern, dass jeder Router die gesamte Routing-Tabelle an jeden Nachbarn sendet.

Beim Routing kommunizieren Router untereinander mit Hilfe von Routingprotokollen, und tauschen Informationen darüber aus, welches die optimalen Wege im Netzwerk sind. Die

Routingtabelle stellt hierbei sozusagen das Herz eines jeden Routingprotokolls dar; in ihr werden typischerweise Ziele, Nachbarn und Wegekosten festgehalten (variiert je nach Protokoll). Da auf einem Router mehrere Routingprotokolle gleichzeitig laufen können, und da jedem Routingprotokoll (mindestens) eine Routingtabelle zugeordnet ist, verfügen viele Router über mehrere Routingtabellen

Routing-Algorithmen auf Distanzvektor-Basis übertragen in periodischen Abständen Kopien einer Routing-Tabelle von Router zu Router. Bei diesen regelmäßigen Aktualisierungen zwischen Routern werden Informationen zu Topologie-Änderungen ausgetauscht.

Jeder Router erhält eine Routing-Tabelle von den benachbarten Routern, mit denen er direkt verbunden ist. Router B empfängt Informationen von Router A. Router B fügt eine Distanzvektor-Nummer (z. B. eine Anzahl von Hops) hinzu, durch die sich der Distanzvektor erhöht. Danach überträgt Router B diese neue Routing-Tabelle an seinen anderen Nachbarn, Router C. Derselbe schrittweise Vorgang läuft in alle Richtungen zwischen benachbarten Routern ab.

Jeder Router, der mit dem Distanzvektor-Routing arbeitet, beginnt mit der Identifikation seiner eigenen Nachbarn. Jede direkte Verbindung zwischen Schnittstelle und Netz hat die Entfernung 0. Im Verlauf des Distanzvektor-Erkennungsprozesses ermitteln Router anhand der Informationen, die sie von jedem benachbarten Router erhalten, den optimalen Pfad zu Zielnetzen. Router A bezieht sein Wissen über andere Netze aus den Informationen, die er von Router B erhält. Jeder der anderen Netzeinträge in der Routing-Tabelle verfügt über einen addierten Distanzvektor, der anzeigt, wie weit entfernt sich das Netz in einer bestimmten Richtung befindet.

Die Routing-Tabelle wird bei Topologie-Änderungen aktualisiert. Wie beim Prozess der Netzerkennung werden Updates zu Änderungen der Topologie schrittweise von Router zu Router weitergeleitet. Distanzvektor-Routing-Algorithmen fordern jeden Router auf, die gesamte Routing-Tabelle an jeden Nachbarn zu senden.

Zu den Nachteilen gegenüber Link-State-Protokollen zählen die schlechten Konvergenzeigenschaften und die mangelhafte Skalierbarkeit.

Eine gute Analogie zu diesen Verfahren, ist im CCNA Semester 2 zu finden:

„Ein Distanzvektor könnte mit den Schildern an einer Straßenkreuzung verglichen werden. Ein Schild gibt ein Ziel und die Entfernung zu diesem Ziel an. Nach einer Weile zeigt ein weiteres Schild das Ziel erneut, aber mit einer geringeren Entfernung an. Solange die Entfernung kleiner wird, folgt der Verkehr dem besten Pfad.“

### **Routen mit RIP dynamisch und statische Routen von Hand**

Auch hier wechselt man direkt in den globalen Konfiguration-Modus. Von diesem wechselt man in den Konfigurationsmodus für RIP.

Dabei wird auch gleichzeitig RIP aktiviert. Nun gibt man dem Router die angeschlossenen Netz-Klassen an, die für das Routing-Protokoll gelten soll.

Anschließend verlässt man diesen Modus wieder, um zurück im globalen Konfigurationsmodus zu gelangen.

Hier werden dann die Netze an den FastEthernet-Schnittstellen statisch bekannt gemacht, da das von RIP in der Version 1 nicht unterstützt wird, dies dynamisch zu machen.

Hingegen die Netze an den seriellen Schnittstellen werden über RIP dynamisch bekannt gemacht, durch den so genannten *network*-Befehl.

#### Befehl

```
LAB_F# conf t
LAB_F(config)# router rip
LAB_F(config-router)# network 192.168.10.0
LAB_F(config-router)# network 192.168.12.0
LAB_F(config-router)# network 198.1.1.0
LAB_F(config-router)# end
LAB_F(config)#ip route 198.1.1.17 255.255.255.240 Serial0/1
LAB_F(config)#ip route 198.1.1.33 255.255.255.240 Serial0/0
LAB_F(config)#ip route 198.1.1.49 255.255.255.240 Serial0/1
LAB_F(config)#ip route 198.1.1.65 255.255.255.240 Serial0/0
```

Mit dem Befehl *router rip* wählen Sie RIP als Routing-Protokoll aus. Der Befehl *network* wird anschließend verwendet, um dem Router mitzuteilen, auf welchen Schnittstellen RIP ausgeführt werden soll. Der Routing-Vorgang weist den Netzadressen anschließend



bestimmte Schnittstellen zu und beginnt damit, RIP-Updates auf diesen Schnittstellen zu senden und zu empfangen.

Der Grund wieso bei den statischen Routen die Schnittstellen einige male wechseln, liegt daran, dass man sich hier selbst für den optimalsten Pfad entscheiden muss. Hier ist unser Routing-Maß der Pfad.

Also der kürzeste Weg zum Gateway ist der optimalste Pfad.

## Der Weg ins Internet

Vorerst müssen dazu erstmal die Grundeinstellungen geändert werde, die aber zum großen Teil, denen von LAB\_F ähneln, nur mit dem kleinen Unterschied, das die IP von der FastEthernet-Schnittstelle sich im selben Netz befinden muss, wie der Gateway fürs Internet.

Befehl
<i>LAB_A # conf t</i>
<i>LAB_A(config)# interface FastEthernet0/0</i>
<i>LAB_A(config-if)# ip address 172.31.100.15 255.255.0.0</i>
<i>LAB_A(config-if)# speed auto</i>
<i>LAB_A(config-if)# no shutdown</i>
<i>LAB_A(config-if)# end</i>
<i>LAB_A(config)#</i>

Gleichzeitig muss noch mit *router rip* die unbekannt Netze via RIP bekannt gemacht werden.

Befehl
<i>LAB_A# conf t</i>
<i>LAB_A(config)# router rip</i>
<i>LAB_A(config-router)# network 192.168.11.0</i>
<i>LAB_A(config-router)# network 192.168.7.0</i>
<i>LAB_A(config-router)# network 172.31.0.0</i>
<i>LAB_A(config-router)# end</i>
<i>LAB_A(config)#</i>

Alle anderen Einstellungen sind mit LAB\_F identisch, nur sollte natürlich als Hostname LAB\_A anstelle von LAB\_F gewählt werden genauso sind die IP-Adressen für die Seriellen-Schnittstellen auch anders.

Befehl
<pre>LAB_A(config)# ip routing LAB_A(config)# interface serial0/0 LAB_A(config-if)# ip address 192.168.7.2 255.255.255.0 LAB_A(config-if)# clock rate 56000 LAB_A(config-if)# no shutdown LAB_A(config-if)# end LAB_A(config)# interface serial0/1 LAB_A(config-if)# ip address 192.168.11.1 255.255.255.0 LAB_A(config-if)# no shutdown LAB_A(config-if)# end LAB_A(config)#</pre>

Nun wollen wir den Router LAB\_A so konfigurieren, dass dieser sich mit dem Internet verbinden kann.

Dazu braucht dieser eine statische Route, die alle unbekanntes IP-Pakete welche als Ziel-IP im IP-Header eine IP haben, zu der es keinen Eintrag in der Routing-Tabelle des Routers von LAB\_A gibt.

Solche eine statische Route wird auch als Default-Gateway bezeichnet.

Befehl
<pre>LAB_A# conf t LAB_A(config)# ip route 0.0.0.0 0.0.0.0 172.31.100.13 LAB_A(config)# ip route 198.1.1.17 255.255.255.240 192.168.11.1 LAB_A(config)# ip route 198.1.1.33 255.255.255.240 192.168.7.2 LAB_A(config)# ip route 198.1.1.49 255.255.255.240 192.168.11.1 LAB_A(config)# ip route 198.1.1.65 255.255.255.240 192.168.7.2 LAB_A(config)# ip route 198.1.1.81 255.255.255.240 192.168.7.2</pre>

Weiterhin muss der Router noch die Möglichkeit haben, einen DNS-Namen in eine IP-Adresse umzuwandeln.

Befehl
--------

<pre>LAB_A# conf t LAB_A(config)# ip name-server 172.31.100.11 LAB_A(config)# ip domain-lookup</pre>
--

Hier ist nun ein DNS-Server angegeben, der sich um die Namensauflösung kümmert. Nach diesen Einstellungen sollte eine Verbindung ins Internet mit LAB\_A möglich sein. Zum testen könnte man nun einen Ping auf google machen.

Befehl
--------

<pre>LAB_A# conf t LAB_A(config)# ping google.de</pre>
--

## Internet für alle

Um nun allen Routern unseres Szenarios eine Verbindung ins Internet zu ermöglichen, müssen alle – außer LAB\_A – nun folgende Befehle am Terminalk absetzen.

Beispielhaft wird das hier auf LAB\_F gezeigt.

Befehl
--------

<pre>LAB_F# conf t LAB_F(config)# ip route 0.0.0.0 0.0.0.0 Serial0/1 LAB_F(config)# ip name-server 172.31.100.11 LAB_F(config)# ip domain-lookup</pre>
--

Beim Festlegen des Default-Gateways ist dabei zu Achten, wie bei jeder anderen statischen Router auch, über welche Schnittstelle man die Pakete weiterschickt.

Hier bei LAB\_F spielt das keine Rolle, da der Weg sich hier gleich verhält, hingegen z.B. bei LAB\_C sollte die Schnittstelle Serial0/0 wählen, da Serial0/1 hier ein unnötiger Umweg wäre.

### **Speichern der Konfigurationsergebnisse:**

Damit wir die Router nicht ständig neu konfigurieren müssen, können wir die Konfiguration des jeweiligen Routers in eine Textdatei abspeichern. Dies bewerkstelligen wir in dem wir in der Menübar von HyperTerminal den Punkt „Text aufzeichnen“ auswählen und daraufhin eine Textdatei auswählen in der die Konfiguration gespeichert werden soll. Nach dem wir diesen Schritt getan haben, müssen wir nun mit Hilfe des Befehls „show run“ die komplette Konfiguration des Routers aufzeigen lassen. Während dessen werden die angezeigten Informationen in der vorher ausgewählten Textdatei abgespeichert. Als letztes wählt man nun in der Menübar den Punkt „Textaufzeichnung beenden“ aus. Somit hat man nun die komplette Konfiguration des Routers in der Textdatei abgelegt.

### **Internetkonnektivität durch RIP**

Nun wollen wir die Internetkonnektivität dynamisch durch RIP setzen. Voraussetzung ist, dass RIP auf allen Routern konfiguriert ist.

Um RIP so zu konfigurieren, dass alle Router über RIP eine Default-Route zu nehmen.

Der Befehl der dies veranlassen kann bewirkt, dass alle RIP fähigen Router, von LAB\_A eine Default-Route zugewiesen wird.

Nun ist eine statische Route nicht mehr notwendig, daher sollte man diese löschen.

Befehl
<i>LAB_A# conf t</i>
<i>LAB_A(config)# default -information originate</i>
<i>LAB_F# conf t</i>
<i>LAB_F(config)#no ip route 0.0.0.0 0.0.0.0</i>