Teamprojekt zum Thema: „Aufbau eines Experimentalkastens für Communication Systems“

Konrad Meier
Holger Bertsch
Dennis Wehrle

Rechenzentrum Universität Freiburg
Lehrstuhl für Komunikationsysteme
Prof. Dr. Gerhard Schneider

Betreuer:
Dirk von Suchodoletz

Wintersemester 2008/09

Inhaltsverzeichnis

1 Einleitung und Motivation

Aufbau eines „Experimentalkastens“ für Communication Systems

Ziel dieses Gruppenprojektes bestand in der Auswahl, Vorbereitung und Einrichtung einer Experimentalumgebung für die Vorlesung „Communication Systems“. Die Planung der Veranstaltung sah eine Aufteilung in einen theoretischen und einen umfangreichen praktischen Teil vor. Dabei war für den praktischen Teil die Hälfte der gesamten Vorlesungszeit vorgesehen. Dieser diente vor allem dazu, den Theorieteil der Vorlesung zu veranschaulichen und den zuvor vermittelten Stoff durch konkrete Anwendungsbeispiele besser verständlich zu machen. Der Schwerpunkt der Übungsaufgaben lag dabei auf klassischen Internetdiensten und ihren Grundlagen. War eine praktische Übung auf Grund der Thematik nicht sinnvoll, wurde diese durch eine Präsentation ersetzt.
In der Vorlesung wurden aktuelle Kommunikationssysteme auf Daten und Telefoniebasis näher betrachtet. Zu Beginn der Vorlesung wurden allgemeine Netzwerktechnik-Grundlagen vermittelt, die durch Themen wie VLAN, Bridges, ARP, NAT und PPPoE vertieft wurden. Anschließend folgte eine Einführung in die Internetprotokolle IPv4, IPv6 und in die Routing-Protokolle. Auf Anwendungsebene wurden unter anderem Protokolle wie DNS, Voice over IP, SSL/TLS und SIP genauer beleuchtet.
Im zweiten Teil der Vorlesung lag der Schwerpunkt auf Themen wie Netzwerksicherheit, Zertifikate, Firewalls, sichere Kommunikation und Themen wie „Quality of Service“ (QoS). Die Vorlesung endete mit einem Überblick über kabelgebundene und kabellose Telefonsysteme. Dazu gehörten Themen wie ISDN, GSM, GPRS und auch der neue Mobilfunkstandard UMTS.

2 Infrastruktur

Der Abbildung 2.1 kann die für den praktischen Übungsteil zur Verfügung stehende und eingesetzte Netzwerk-Infrastruktur entnommen werden. Hierzu gehören insgesamt 30 PC Arbeitsplätze und ein zentraler Server (ComSys Server). Als Betriebssystem kommt ein speziell angepasstes Xubuntu zum Einsatz. Es handelt es sich hierbei um ein VMware Image, das mit Hilfe des VMware Players in einer virtuellen Umgebung gestartet wird und vor jeder Übung angepasst wurde. Die für die Übungsaufgaben entsprechend benötigten Programme oder Pakete wurden bereitgestellt und entsprechend vorkonfiguriert. Der Einsatz eines VMware Images bietet einige Vorteile. Die Möglichkeit das System in einem nicht persistenen Modus zu starten, stellt einen der wichtigsten Gründe für ein VMware Image dar. Dies bedeutet, dass es den Studenten nicht möglich ist, das Übungssystem zu verändern. Somit konnte den Studenten der Zugang als „root“ zum System erlaubt werden. Dies war für die meisten Übungsaufgaben auch zwingend erforderlich.
Um sämtliche Netzwerkkommunikation vom restlichen Universitätsnetzwerk zu trennen findet eine Vernetzung der einzelnen Rechner über VLANs (Virtual Local Area Network) statt. Mit Hilfe von VLANs können innerhalb der Universität über einen Switch oder über mehrere Switches hinweg virtuell getrennte Netze betrieben werden. Hierbei tragen die Pakete eine Markierung, welche die Zugehörigkeit zu einem VLAN festlegt. Innerhalb der Universität wurden hierfür für die Poolräume die VLANs mit den IDs 260 bis 295 eingerichtet.
Der ComSys Server stellt Dienste wie DHCP, HTTP(S), PPPoE, SIP und DNS für die entsprechenden Übungsaufgaben zur Verfügung. Gleichzeitig dient er als Gateway für die einzelnen VLAN-Netze und ermöglicht den Zugang zum Internet. Als Betriebssystem kommt Ubuntu 8.04 zum Einsatz.

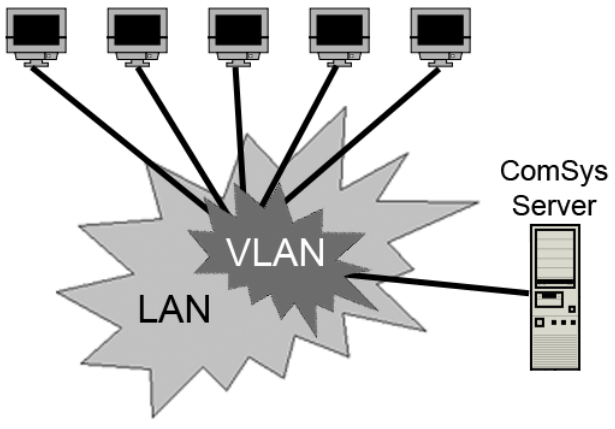


Abbildung 2.1: Auschnitt Netzwerk Infrastruktur mit VLAN

Alle erstellten Übungsblätter, das Xubuntu VMware Image sowie das mit Hilfe der SystemRescue CD 1.1 erstellte Backup Image des ComSys Servers sind auf der beigelegten DVD abgespeichert. Des Weiteren befindet sich im Anhang eine Beschreibung der für die Übungsblätter benötigten Tools.

3 Grundlagen Blatt 1

Bei der ersten praktischen Übung sollen die Studenten sich mit dem Linux-System (Xubuntu) vertraut machen und grundlegende Kenntnisse über Netzwerke erlangen. Zu den Aufgaben gehört es, die IP und MAC Adressen der einzelnen Systeme herauszufinden und mit Hilfe der Kommandos *ifconfig*, *ping* und *traceroute* die grundlegenden Befehle für die weiteren praktischen Übungen zu erlernen.
Das speziell angepasste Xubuntu Image ist mit allen dafür benötigten Tools in einer VMware vorinstalliert.

3.1 Aufgaben

Question 1
Login to the machine using your standard computer center account (the ID of the faculty of computer science will not work on these systems) and explore the standard Linux environment. This should give you access to a range of standard tools for network analysis but without system administrators privileges. You might start a console window (the place to enter commands) by using "ALT-F2" and typing "konsole" in KDE3 and/or "gnome-terminal" in Gnome graphical environment. Within the menu you will find other ways to open the shell.

Question 2
Try to find out the IP address of both machines (host and guest Linux system) you are using. Which differences you observe?

Question 3
Try to find out the MAC address of your network adaptors (both real and virtual).

Question 4
What is the full qualified domain name of your machine? Which program gives you information on IP addresses, host names of other machines, e.g. your neighbors?

Question 5
What information the commands `ping` and `traceroute` produce?

Question 6
Playing around a little bit with `ping`: Ask your neighbor for the IP address of its machine and exchange pings! Try to ping your neighbors machine from your host and your guest system (any differences). Send ICMP messages (simply means pinging) from your guest system to the host system and vice versa. Try to ping your neighbors guest machine from your outside and guest environment. What do you observe?

Question 7
Use the `traceroute` command on the tasks in question 6! Which additional, useful information it produces? Deploy this command on IP addresses like *134.76.10.46* or *132.230.200.200*.

3.2 Lösungsskizze

Question 1
Keine Lösung notwendig.

Question 2
Befehl: *ifconfig*
Mangels fehlenden root Rechten am Hostsystem muss ifconfig über */sbin/ifconfig* aufgerufen werden.

Question 3
Befehl: *ifconfig*
Siehe Question 2.

Question 4
Befehl: *hostname*

Question 5
Ping sendet ein „Echo Request“ ICMP-Paket an die Zieladresse. Die Zieladresse antwortet mit einem „Echo-Reply“ ICMP-Paket.
Traceroute sendet mehrfach Pakete mit jeweils um 1 erhöhten TTL-Wert (Time-to-live) an das Zielsystem. Jeder Host, der das Datenpaket empfängt zählt den Wert der TTL um eins herunter. Empfängt ein Router ein Paket mit TTL-Wert = 1 wird das Paket verworfen und eine ICMP-Antwort Typ 11: „Time-to-live exceeded“ an den Absender zurückgeschickt. Die Sequenz der so gesammelten Adressen kennzeichnet den Weg zum Ziel durch das Netz.

Question 6
Es kann beobachtet werden, dass sich Gast und Host-System in MAC und IP-Adresse unterscheiden. Je nachdem ob VMware im „bridged“ oder „NAT“ Modus läuft, kann das Ergebnis variieren.

Question 7
Das Programm „traceroute“ gibt aus, über welche Router ein Rechner zu erreichen ist.
Läuft die VMware im „NAT“-Modus wird der NAT-Router zwischen Gast und Host-System ausgegeben.
Beispiel „traceroute“ zu heise.de:

```
traceroute to heise.de (193.99.144.80), 30 hops max, 40 byte packets
 1  132.230.4.254 (132.230.4.254)  0.746 ms  1.045 ms  0.993 ms
 2  nsc-rz-gwn.fun.uni-freiburg.de (132.230.0.141)  0.333 ms  0.295 ms  0.698 ms
 3  Freiburg1.belwue.de (129.143.15.141)  2.057 ms  1.843 ms  1.763 ms
 4  Karlsruhe1.belwue.de (129.143.1.4)  7.632 ms  7.599 ms  7.430 ms
 5  Frankfurt1.belwue.de (129.143.1.178)  8.941 ms  8.899 ms  8.818 ms
 6  te3-1.c302.f.de.plusline.net (80.81.193.132)  10.759 ms  9.898 ms  9.814 ms
 7  82.98.98.110 (82.98.98.110)  10.828 ms  10.777 ms  10.696 ms
 8  82.98.98.110 (82.98.98.110)  10.621 ms !X * *
```

4 Grundlagen Blatt 2

Mittels den Befehlen *ifconfig* und *route* werden zunächst die bisher erlangten Kenntnisse weiter vertieft, indem eine eigene IP und ein gegebener „default gateway“ gesetzt werden. Zusätzlich wird das Tool Wireshark eingeführt, mit dessen Hilfe erstmalig die generierten Pakete sichtbar werden. Somit können die Auswirkungen verschiedener MTU Größen der Ethernet Schnittstelle anhand der Fragmentierung von Paketen veranschaulicht werden.

4.1 Aufgaben

Question 1
If you have started Xubuntu successfully, you will find the DHCP client disabled. Therefore you have to set an IP-address *10.230.4.X* and *10.230.4.1* as default gateway. But watch out: The number X should not be equal to a number from anyone in this room :) Try to ”ping” your default gateway and some neighbor first and then proceed to *ping google.de* to check if everthing (testing name resolution that way) works fine. If the name of the destination is not resolved properly the resolver is not setup right (wrong DNS server or unreachable because of incomplete, wrong routing).

Question 2
Now set your default gateway to a virtual machine from your neighbours. Check your access to the Internet. What happend? (Use *wireshark* for in depth exploration)

Question 3
Start the *wireshark* program on your Ethernet interface (typically ”ethN”, N some number starting from 0)! *ping 127.0.0.1*. Do you see any packets? Why/not?

Question 4
Ping again: point your *wireshark* to your loopback interface and *ping -s 8192 127.0.0.1* How many packets are generated? After that: point your *wireshark* to your Ethernet interface and *ping -s 8192 132.230.4.2* How many packets are generated? Why?

Question 5
Configure an alternate MTU size to your Ethernet interface, e.g. 1000. Ping your neighbor with a ICMP packet using exactly the maximum of packet size without beeing split. What is the size you have to give for the payload, why? Let ping your neighbor without changing his MTU size: Sent a standard ICMP packet and a packet filling up his MTU. What happens? (If your neighbor is pinging you at the same time, you might want to distinguish your packets easily. Check the man page of the ping command to get an idea how to apply a pattern to the packets payload.)

Question 6
Find out with *wireshark* or similar tools what kind of transport layer protocols are used by ICMP, SSH, HTTP, FTP, DNS. How to trigger the use of these networks – example: open some SSH connection (login2.ruf.uni-freiburg.de) and use wireshark to analyze. Alike should be easy

for the other protocols too. (You might be astonished to find out, how many different sites are connected additionally if opening some commercial web site.)

Question 7
If you are really fast: Try to find out how to change the MAC address of your virtual Ethernet card in the Xubuntu system!

4.2 Lösungsskizze

Question 1
ifconfig eth0 10.230.4.10
route add default gw 10.230.4.1

Question 2
route del default
route add default gw 10.230.4.X
Die Pakete werden vom Host zur Nachbarmaschine gesendet. Aufgrund des fehlenden IP-Forwarding der Nachbarmaschine wird das Paket jedoch verworfen. Somit besteht für den Host keine Internetverbindung.

Question 3
Der Ping an die lokale IP-Adresse 127.0.0.1 ist in Wireshark durch das eth0 Interface nicht sichtbar, da das Ziel der eigene Host ist und somit keine Pakete am eth0 Interface generiert werden.

Question 4
Die MTU Größe des lo Interfaces beträgt 16436 Bytes und die des eth0 Interfaces 1500 Bytes. Somit wird bei dem lo Interface 1 Paket und bei dem eth0 Interface 6 Pakete generiert.

Question 5
Bei einer MTU Größe von 1000 Byte darf der maximale Payload 972 Byte nicht übersteigen. Der Grund hierfür ist der 20 Byte große IP-Header und der 8 Byte große ICMP-Header
Wird ein Datenpaket an einen Host gesendet das größer ist als dessen MTU, kann der Host das Paket nicht entgegennehmen und verwirft das Paket.

Question 6
Folgende „transport layer“ Protokolle werden verwendet:
ICMP: -
SSH: TCP
HTTP: TCP
FTP: TCP
DNS: UDP

Question 7
ifconfig eth0 down hw ether MAC-ADRESSE
ifconfig eth0 up

5 Grundlagen Blatt 3

Ziel des letzten Grundlagenblattes ist es einen Einblick in VLANs, Bridges und dem Spanning Tree Protocol (STP) zu geben. Mit Hilfe von VLANs (Virtual Local Area Network) können innerhalb der Universität über einen Switch oder über mehrere Switches hinweg virtuell getrennte Netze betrieben werden. Hierbei tragen die Pakete eine Markierung, welche die Zugehörigkeit zu einem VLAN festlegt. Innerhalb der Universität wurden hierfür für die Poolräume die VLANs mit den IDs 260 bis 295 eingerichtet. Die VLANs bilden die Grundlage für zahlreiche weiteren praktischen Übungen.

5.1 Aufgaben

Question 1
Power up the *dummy0* interface in your machine and configure a MAC address to it just taking the Ethernet MAC and change the second block number. After these preliminary steps, check that the interface is visible. Start setting up a simple two ports bridge named *br0*.

Question 2
Enable STP on the bridge and analyse the generated pakets with wireshark. Do you see any packets from neighboured machines? Why/why not? What is the effect of changing the bridge priority?

Question 3
Then try to *ping* some neighbors Ethernet and then the dummy0 interface ... Check for the ARP messages exchanged. Which kind of destination MAC addresses you see? Will you see every exchanged packet on the eth0, br0, dummy0 interface?

Question 4
Prepare your system to be used with Ethernet VLANs. It will be used for more sophisticated network setups in future experiments. It will allow to setup virtual links between different hosts "not seen" by others in the same physical LAN. Use the VLAN ID 260. Run the *dhclient3 ethN.260* command on it and try to obtain an IP address.

Question 5
Run *wireshark* on eth0 und eth0.N (N VLAN number)! Do you observe any differences?

Question 6
Configure some additional VLAN using an ID between 261 and 279. Coordinate with your neighbors and use the same ID. Setup a *10.231.4.X/24* (or any alike) network and exchange ICMP packets. Check with *wireshark* on the different interfaces for the ARP messages exchanged! Have a look at your kernel routing table!

5.2 Lösungsskizze

Question 1
modprobe dummy0
ifconfig dummy0 down hw ether MAC-ADRESSE
ifconfig dummy0 up
ifconfig eth0 0
ifconfig br0 up
ifconfig br0 IP-ADRESSE

ACHTUNG: Die Übungsaufgabe funktioniert leider nicht wie erwartet. Sendet man Pakete von einem anderen Rechner an das br0 Interface sind diese nicht auf dummy0 sichtbar.

Question 2
Die STP-Pakete sind auf auf beiden Interfacen zu beobachten. STP bildet eine Spannbaum in der der Rechner mit der niedrigsten Priorität (priority) der Wurzelknoten wird.

Question 3
Als MAC-Adresse sollte die Adresse des br0 Interfaces zu sehen sein. Die zwischen zwei Rechnern ausgetauschten Pakete sind nicht auf dem dummy Interface zu sehen.

Question 4
Keine Lösung notwendig.

Question 5
Auf dem Interface eth0.N ist nichts auffälliges zu beobachten. Bei eth0 sieht man jedoch VLAN spezifische Informationen.

Question 6
Keine Lösung notwendig.

6 ARP

Das Address Resolution Protocol (ARP) ist ein Netzwerkprotokoll, dass die Zuordnung von Hardwareadressen (MAC) zur entsprechenden IP Adresse ermöglicht. Hierzu wird eine ARP-Broadcast mit der IP Adresse des gesuchten Computers gesendet. Ein gesuchte Host, sendet die gesuchte MAC Adresse zurück. Sobald der Host eine ARP-Anforderung oder Antwort erhält, wird der ARP-Cache automatisch aktualisiert.
Nachdem sich die Studenten die Funktionsweise von ARP durch Wireshark verdeutlicht haben, sollen mögliche Angriffspunkte besprochen werden. Mit Hilfe der Tools *ettercap* und *arpspoof* werden anschließend diese Angriffe durchgeführt. Hierbei sollen Vor- und Nachteile von statischen beziehungsweise dynamischen ARP-Mechanismen verdeutlicht werden.

Die Verteilung der VLAN IDs auf die Rechner geschieht wie folgt:

VLan ID's

261	264	267	270
262	265	268	271
263	266	269	272

6.1 Aufgaben

Question 1
Use *wireshark* to analyse how ARP works. Try to ping each other and analyse the generated packets. After that take a look in the ARP table.

Question 2
With the command *arp* it is possible to show the actual ARP table (if the table is empty try *ping google.de* and look again). Change the entry of the Gateway so that your VLAN MAC address is used. Is it still possible to *ping google.de* Why?/Why not? After that delete the Gateway entry *arp -d <gateway-ip>*.

Question 3
From now on work together with your neighbour. Use *ettercap* for sniffing. Can you find any HTML fragments? Is it possible to see what your neighbour is doing? Is it possible with ARP replies to tie a network?

Question 4
Start an ARP poison attack with *arpspoof*. Analyse the generated traffic with *wireshark*. Is it possible to detect the attacker?

Question 5
Switch off the kernel ARP mechanism by using the *ip* or *ifconfig* command! Try to ping the subnet gateway and other machines in your subnet! What happens? Add the MAC address of the gateway manually to the ARP table. Try to ping again. Try the *ettercap* exploit again - what do you observe? Which (dis)advantages does this concept have?

Question 6
Start the command *arpwatch*. Ping some machines in the subnet. What happens and what does it tell you?

6.2 Lösungsskizze

Question 1
Keine Lösung notwendig.

Question 2
arp -s <gateway-ip> <vlan-mac-addresse>
Wird der die MAC-Adresse der Gateway geändert, ist diese nicht mehr erreichbar. Somit ist ein ping zu google auch nicht möglich da im Ethernet-Frame die falsche MAC-Adresse steht.

Question 3
Mittels einer Man-in-the-middle Attacke können alle Fragmente mitgesnifft werden. Es ist möglich den gesamten Datenverkehr des Nachbarn mitzuschneiden. Des Weiteren ist es mit ARP Spoofing / Poisoning möglich das Netzwerk so zu verändern, dass der Datenverkehr zwischen verschiedenen Rechnern abgehört oder manipuliert werden kann. Es ist auch denkbar, dass ein Netzwerk durch einen ARP-Angriff lahmgelegt wird. Hierzu werden falsche MAC-Adressen im Netzwerk verbreitet, was zur Folge hat, das Verbindungen zwischen Rechnern unmöglich werden (siehe Question 2).

Question 4
Es ist möglich einen Angreifer zu erkennen, wenn dieser eine große Anzahl von ARP-Informationen im Netzwerk verbreitet. Die Anzahl der ARP-Packete ist dann im Vergleich zu den andren Rechnern auffällig. Es ist auch möglich mit speziellen Programmen den ARP-Datenverkehr zu überwachen. Ein Beispiel hierzu ist ArpWatch in Question 6.

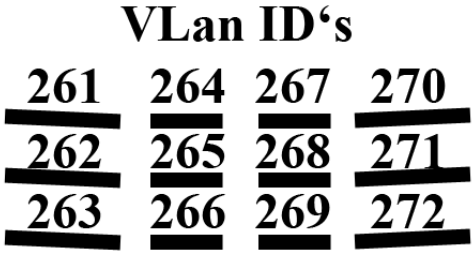
Question 5
ifconfig eth0 -arp
Ohne arp kann das Gateway nicht gepingt werden. Nachdem die MAC Adresse des Gateway manuell eingetragen wurde funktioniert ping wieder. Das „ettercap exploit“ funktioniert nicht mehr, da die MAC Adressen nun nicht mehr automatisch eingetragen werden.
Vorteil: Kein arp spoofing / poisoning mehr möglich
Nachteil: MAC Adresse muss bekannt sein und manuell eingetragen werden; Unflexibel;

Question 6
Arpwatch überwacht alle ARP Aktivitäten im Netzwerk und speichert diese. ArpWatch kann mit Hilfe von Scripten so konfiguriert werden, dass es ARP-Angriffe erkennt.

7 PPPoE

Das PPPoE (Point-to-Point Protocol over Ethernet) wird zurzeit in Deutschland vor allem bei ADSL-Anschlüssen verwendet. Für den Verbindungsaufbau zu einem PoP (Point-of-Presence) wird mittels Ethernet-Broadcast das PPPoE-Discovery (PPPoED) benutzt. Um diesen Verbindungsaufbau zu verdeutlichen wird daher eine PPPoE Verbindung zu unserem Comsys-Server aufgebaut und mit Wireshark näher betrachtet. Als weiteren Schritt setzen die Studenten in Gruppenarbeit zunächst einen PPPoE-Server ohne Authentifizierungsmechanismen auf. Anschließend wird die Konfiguration des PPPoE-Servers um pap (Password Authentication Protocol) oder chap (Challenge Handshake Authentication Protocol) Authentifizierung erweitert.

Die Verteilung der VLAN IDs auf die Rechner geschieht wie folgt:



7.1 Aufgaben

Question 1
Setup a VLAN with ID 274: *vconfig add eth0 274*.
Try to establish a PPPoE-Connection on your eth0.274 Interface. The command *pppoe-setup* and *pppoe-start* can be useful. Use as username: comsys and PPPoE password: comsys08 and DNS-Server: *132.230.200.200*. To see all configuration parameters take a look in */etc/ppp/pppoe.conf*. Analyse the generated PPPoE pakets with wireshark. *google.de* should be available ;) (if not check DNS local resolver configuration)

Question 2
Setup a VLAN with ID X: *vconfig add eth0 X*. See the graphics for ID distribution. Try to work with your neighbour(s), one configure a PPPoE server and the other try to connect with that PPPoE server without authentification. Analyse in each step the generated traffic with wireshark. Which specific headers are added either for PPP and PPPoE? How the whole protocol stack of an average HTTP packet traveling over the PPPoE connection looks like?

a) Use the *man pppoe-server* command to learn the main options of the PPPoE server. Take a look at -m; -N; -I; -L; -R; -F etc. Try to start and connect to a PPPoE Server on your eth0.VLAN interface.

b) Try to add a PPPoE server name with the option *S name*, and now try to connect with the servername.

c) Modify the */etc/ppp/pppoe-server-options*, so that a connection with pap or chap is possible. Add two lines to the configuration file:
auth
require-pap or *require-chap*
... and specify a username and password. Restart the PPPoE server and reconfigure your client. Is the username / password exchange seen in any packet captures?

Question 3
PPPoE operates with dedicated serial links. Is the Address Resolution Protocol still needed (Why/not)?

Question 4
Check the MTU of your different interfaces! What do you observe? Try to re-configure your PPPoE connection to a different MTU, e.g. of 1000 Bytes! Why would it be desirable to set a MTU of 1500 Bytes to your PPP interface too? Would it be possible to use the MTU of 1500 Bytes there (Why/not)?

Question 5
Calculate roughly the overhead introduced by PPPoE compared to standard Ethernet connection taking ARP and PPP control protocols and headers into account!

7.2 Lösungsskizze

Question 1
Keine Lösung notwendig.

Question 2

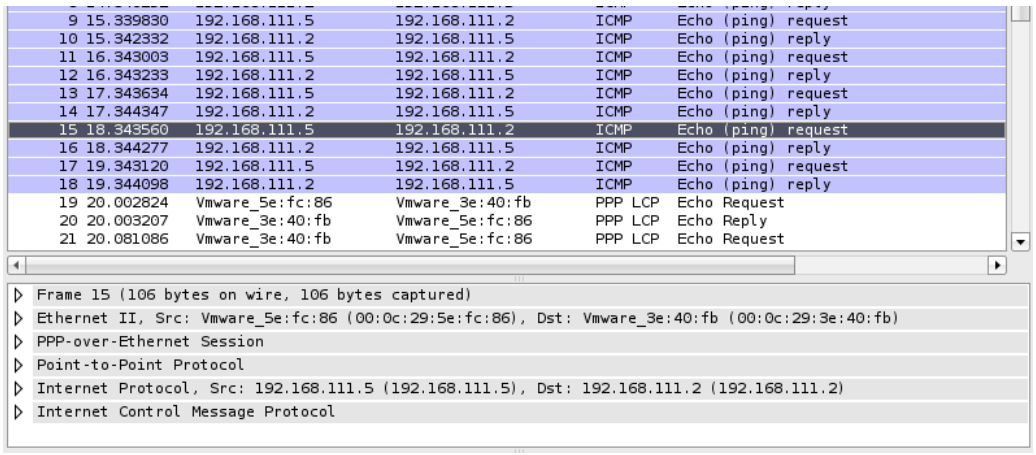


Abbildung 7.1: Protokollstack eines ICMP Datenpakets über eine PPPoE Verbindung

a) *pppoe-server -m 1412 -N 330 -I ethX.274 -L IP (Server) -R IP (Client) -F*

b) *pppoe-server -S name -m 1412 -N 330 -I ethX.274 -L IP (Server) -R IP (Client) -F*

c) Siehe Grafik 7.2. Es wird ein „challenge response“ Verfahren benutzt. Ein Passwort wird nicht übertragen.

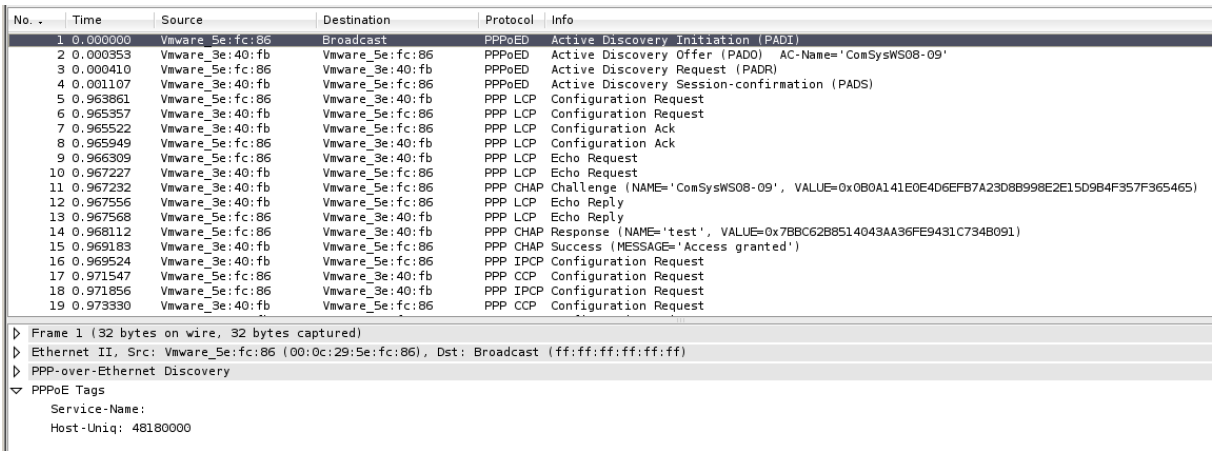


Abbildung 7.2: Protokoll eines PPPoE-Verbindungsaufbaus mit Chap

Question 3
Da eine Direktverbindung besteht wird kein ARP benötigt. Für den Verbindungsaufbau wird PPPoE Discovery (PPPoED) benutzt.

Question 4
Beispielberechnung der MTU anhand eines ICMP-Pakets über eine PPPoE Verbindung mit aktivem VLAN:

```
                1500 Byte
- VLAN:            4 Byte
- PPPoE-Header:    6 Byte
- PtP-Header:      2 Byte
- IP-Header:      20 Byte
- ICMP-Header:     4 Byte
----------------------------
          MTU = 1464 Byte
```

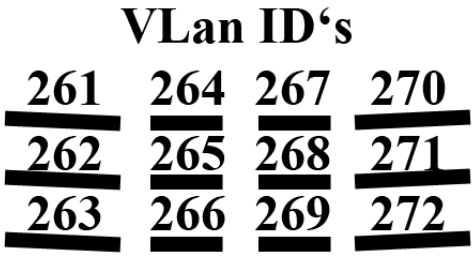
In diesem Beispiel ist der ICMP-Header relativ klein. Ein TCP-Header wäre mit 20 Byte wesentlich größer. Deshalb wird in der „manpage“ des pppoe-Servers eine MTU von 1412 Byte empfohlen.

Question 5
Wie in der Lösungsskizze zu Aufgabe 4 zu sehen ist, beträgt der Overhead durch eine PPP-Verbindung 8 Byte (PPPoE-Header+PtP-Header).

8 IP / DHCP

Für die Vernetzung verschiedener Rechner auf Basis des Internet Protocol (IP) werden hier zwei Methoden vorgestellt. Zum einen das Tool „avahi“ das eine konfigurationsfreie Vernetzung ermöglicht, und zum anderen ein DHCP Server. Avahi wird für die automatische Konfiguration der Netzwerkschnittstellen ohne manuelles Eingreifen oder zentralen Systemen, wie beispielsweise ein DHCP Server, eingesetzt. Hierzu generiert jeder Host per Zufallsgenerator – beeinflusst durch die eigene MAC Adresse – eine IP Adresse. Unter Zuhilfenahme von ARP Paketen muss anschließend ein möglicher IP Adressen Konflikt aufgedeckt und falls notwendig eine neue IP Adresse berechnet werden. Des Weiteren wird unter Anwendung von multicast DNS (mDNS) die Auflösung von Hostnamen zu IP Adressen ohne DNS Server und das automatische Auffinden von Diensten im lokalen Netzwerk ohne einen zentralen Directory-Server ermöglicht. Im Gegensatz dazu ermöglicht das Dynamic Host Configuration Protocol (DHCP) die Netzwerkschnittstellenkonfiguration durch einen zentralen Server. Von dem DHCP Server kann zu diesem Zwecke die IP Adresse, die Netzmaske, das Gateway, DNS-Server und weitere Informationen bezogen werden. Der entscheidende Vorteil besteht bei großen Netzwerken vor allem darin, dass Änderungen der Netzwerktopologie lediglich am Server eingestellt werden müssen.

Die Verteilung der VLAN IDs auf die Rechner geschieht wie folgt:



8.1 Aufgaben

Question 1
Setup a VLAN with ID 274: *vconfig add eth0 274*
Start wireshark and analyse the command: *avahi-autoipd eth0.274*. This command should give you an auto-ip setup for your local machine. Try to analyse the generated packets. What kind of packets do you see? How does it work?

Question 2
Start the *avahi* daemon (background service to handle auto-ip, mDNS and service discovery). What are the commands to explore the network? Do you see anything in wireshark?

Question 3
Setup a VLAN with ID 260: *vconfig add eth0 260*. Try to request via DHCP client a certain IP the server does not provide in the eth0.260 interface! (see client configuration file dhclient.conf) How to configure the client to request certain options, e.g. a certain host name!?

Question 4
Setup a VLAN with ID x: *vconfig add eth0 x*. See the graphics above for ID distribution. Try to work with your neighbour(s), one configure an starts a DHCP server (configuration file: */etc/dhcp3/dhcpd.conf*) and the other(s) try to connect with that DHCP server. Analyse in each step the generated traffic with wireshark.

a) Lease time: set the lease time of your server to a real small value and observe the renew procedure! Who is starting the procedure? Why does not the DHCP server checks if a certain client is alive? What is the difference to the keep alive packets of the PPP? Why the concept of a "lease" was introduced to DHCP evolving from the BOOTP? Discuss the security implications of DHCP - do you trust the answer from your server?

b) Please check the lease file of your client and the amount of data transferred. By now you have used a rather small subset of the DHCP options available. Take a look in the manpage of the server and add a few options (e.g. ntp, wins, etc.) and get a new lease on the client!

c) Reconfigure the server so that clients get always the same IP.

d) Configure your DHCP server to provide a ntp server (check *man dhcp-options*.

e) The DHCP Server allows to set vendor specified options: Define a TEXT field and transfer a simple message. Try to transfer a message of 1024 and then of 2048 Byte. Does it work? Hint: A new option string must be defined in server and client configuration file *dhcpd.conf, dhclient.conf* (check *man dhcp-options*.

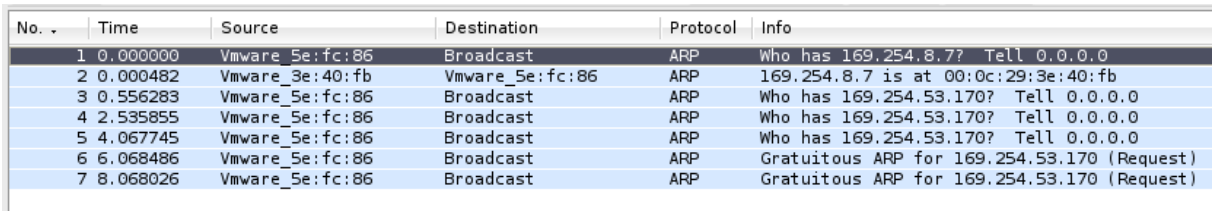
Question 5
Homework/to be tried at home: Write a (dhcp client) script which writes the content of your vendor provided TEXT field to some file! Or start some action or set some other dhcp option in your systems ...

8.2 Lösungsskizze

Question 1
Beispiel für Avahi auf Vlan 260:

```
root@ComSysWS08-09:~# avahi-autoipd eth0.260
Found user 'avahi-autoipd' (UID 105) and group 'avahi-autoipd' (GID 113).
Successfully called chroot().
Successfully dropped root privileges.
Starting with address 169.254.8.7
Received conflicting normal ARP packet.
Trying address 169.254.53.170
Callout BIND, address 169.254.53.170 on interface eth0.260
Successfully claimed IP address 169.254.53.170
```

In der Konsolenausgabe und Grafik 8.1 ist zu sehen, dass Avahi versucht die IP-Adresse 169.254.8.7 dem Netzwerkinterface zuzuweisen. Jedoch ist diese IP-Adresse schon vergeben, da Avahi eine Antwort auf sein ARP-Broadcast Paket bekommt. Daraufhin wird eine neue Adresse generiert und überprüft ob diese vergeben ist (Ist sie in diesem Fall nicht).



No.	Time	Source	Destination	Protocol	Info
1	0.000000	Vmware_5e:fc:86	Broadcast	ARP	Who has 169.254.8.7? Tell 0.0.0.0
2	0.000482	Vmware_3e:40:fb	Vmware_5e:fc:86	ARP	169.254.8.7 is at 00:0c:29:3e:40:fb
3	0.556283	Vmware_5e:fc:86	Broadcast	ARP	Who has 169.254.53.170? Tell 0.0.0.0
4	2.535855	Vmware_5e:fc:86	Broadcast	ARP	Who has 169.254.53.170? Tell 0.0.0.0
5	4.067745	Vmware_5e:fc:86	Broadcast	ARP	Who has 169.254.53.170? Tell 0.0.0.0
6	6.068486	Vmware_5e:fc:86	Broadcast	ARP	Gratuitous ARP for 169.254.53.170 (Request)
7	8.068026	Vmware_5e:fc:86	Broadcast	ARP	Gratuitous ARP for 169.254.53.170 (Request)

Abbildung 8.1: Screenshot Wireshark: von Avahi generierte Pakete

Question 2
avahi-daemon -D
avahi-autoipd eth0.260 -D
Mit dem Befehl *avahi-browse -a -t* können Services im Netzwerk angezeigt werden.
In Wireshark sind MDNS Pakete zu sehen.

Question 3
In */etc/dhcp3/dhclient.conf* beispielsweise folgende Zeile einfügen:
interface eth0.260 send dhcp-requested-address 10.230.4.255;

Question 4

a) Zur Vermeidung der Serverlast und um den DHCP-Server nicht unnötig kompliziert zu machen, ist der Client für den renew-Prozess zuständig. Es gibt auch keine „keep alive packets“ und auch keine aktive Verbindung zwischen DHCP-Server und Client. Der DHCP-Server vergibt eine „lease“ an einen Client für eine bestimmt Zeit. Sollte sich nach Anlauf dieser Zeit der Client nicht gemeldet haben, geht der Server davon aus, dass der Client nicht mehr im Netzwerk ist. Bei Bedarf wird die Adresse dann an andere Clients vergeben.
Einer Antwort von einem DHCP-Server kann im allgemeinen nicht getraut werden, da nicht sicher gestellt werden kann das die Antwort vom entsprechenden Server stammt. Es sind Angriffsszenarien denkbar, in dehnen ein Angreifer einen DHCP-Server in eine Netzwerk einschleust und über diesen einen neuen „default-gateway“ den Clients mitteilt. Das neue Gateway wäre in der Lage den gesamten Datenverkehr im Netzwerk zu überwachen.(Man in the middle Szenario)

b) Es gibt sehr viele DHCP-Options. Eine Übersicht ist hier zu finden:
http://www.iana.org/assignments/bootp-dhcp-parameters
http://spblinux.de/2.0/doc/dhcp-options.html
Beispiel: option domain-name “isc.org”;

c) In die Datei */etc/dhcp3/dhcpd.conf* folgende Zeile einfügen:
host test hardware ethernet 00:00:00:00:00:00; fixed-address 1.1.1.1;

d) option ntp-server IP-ADRESSE;

e) Server: *dhcpd.conf*
Im header: option my-string code 200 = string;
Im subnetz: option my-string „test_text“

Client: */etc/dhcp3/dhclient.conf*
Im header:
option my-string code 200 = string;

request my-string;

Question 5
Keine Lösung notwendig.

9 Static Routing

In diesem Kapitel soll die Idee des statischen Routing anhand eines einfachen Setups veranschaulicht werden (siehe Grafik 9.1). Dabei steht das Verständnis und der Umgang mit der Routingtabelle im Vordergrund. Es soll klar werden, welche Bedeutung einzelnen Einträgen in der Routingtabelle zukommt und welchen Einfluss diese auf das globale Routing haben.

9.1 Aufgaben

Question 1
Work together to build the network shown in the picture. Within your group one computer should operate as a router. In the first step, you have to setup a Vlan with ID X (X = your ID shown in the picture). Afterwards you have to choose a proper ip address with correct netmask within your Vlan ID (also shown in the picture).

Check your configuration:
You can use the *ip* command to show your
network interfaces: *ip link show*
ip adress: *ip address show*
the routing table: *ip route show*

Question 2
Until now, you can only ping your direct neighbours. To ping someone within your group, you have to activate ip forwarding!
Use the command: *echo '1' > /proc/sys/net/ipv4/ip_forward*
To test if everything works fine, try to ping someone within your group.

Question 3
Everyone except the „Router“ have to set a proper default gateway and the „Router“ has to set *10.230.Y.241* as default gateway. (See the picture ;))

Question 4
Analyse with *traceroute* several routes to other hosts. Use *traceroute* to check how many routers are between you and the default gateway. Check again traceroute with *www.uni-freiburg.de*. Is there a difference? Why?

Question 5
Have a look at the picture again and write down the routing table for the default gateway „ComSys Server“.

Question 6
Setup a dummy0 interface with IP address 192.168.Y.X and netmask 255.255.255.0 (24 Bit). Edit your routing table. Try to ping dummy interfaces from other people.
(Hint: command to setup dummy0 interface: *ip link set up dev dummy0*)

9.2 Lösungsskizze

Question 1
vconfig add eth0.X
ifconfig eth0.X 10.230.4.Z\24

Question 2
Keine Lösung notwendig.

Question 3
route add default gw 10.230.Y.241

Question 4
Die Anzahl der Router zwischen dem eigenen Rechner und der Gateway variiert je nachdem wo man sich im Netzwerk befindet.

Question 5

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
10.230.0.241	0.0.0.0	255.255.255.240	U	0	0	0	eth0.290
10.230.1.241	0.0.0.0	255.255.255.240	U	0	0	0	eth0.291
10.230.2.241	0.0.0.0	255.255.255.240	U	0	0	0	eth0.292
10.230.3.241	0.0.0.0	255.255.255.240	U	0	0	0	eth0.293
10.230.4.241	0.0.0.0	255.255.255.240	U	0	0	0	eth0.294
10.230.5.241	0.0.0.0	255.255.255.240	U	0	0	0	eth0.295
132.230.4.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
10.230.0.0	10.230.0.254	255.255.255.0	UG	0	0	0	eth0.290
10.230.1.0	10.230.1.254	255.255.255.0	UG	0	0	0	eth0.291
10.230.2.0	10.230.2.254	255.255.255.0	UG	0	0	0	eth0.292
10.230.3.0	10.230.3.254	255.255.255.0	UG	0	0	0	eth0.293
10.230.4.0	10.230.4.254	255.255.255.0	UG	0	0	0	eth0.294
10.230.5.0	10.230.5.254	255.255.255.0	UG	0	0	0	eth0.295
0.0.0.0	132.230.4.254	0.0.0.0	UG	0	0	0	eth0

Question 6
Keine Lösung notwendig.

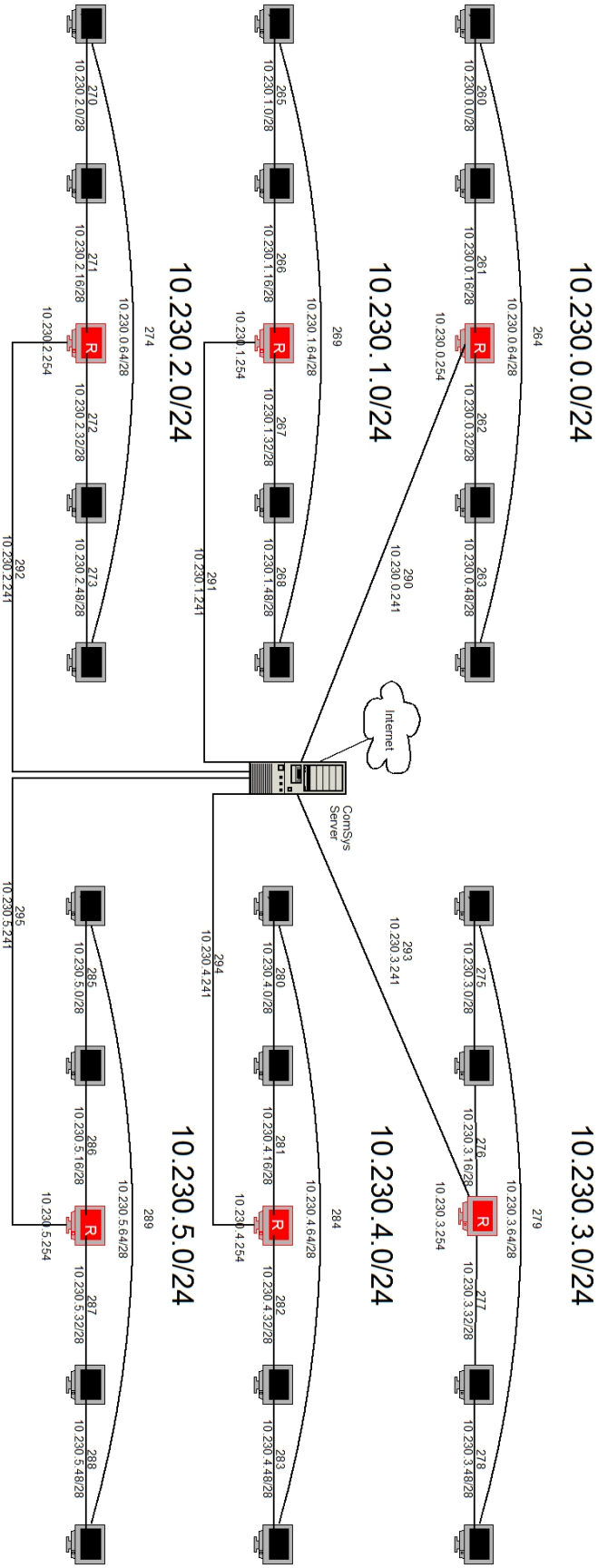


Abbildung 9.1: Netzwerksetup für statisches Routing

10 NAT

NAT oder auch als „Masquerading“ bezeichnet stellt den Sammelbegriff für Verfahren dar um Adressinformationen durch andere zu ersetzen. NAPT stellt mittlerweile die häufigste Form des NAT dar und wird daher oft als synonym gebraucht. Da es neben der Umsetzung von IP-Adressen auch eine Umsetzung von Port-Nummern gestattet, wird es oft eingesetzt, eine Reihe von (privaten) IP-Adressen und zugeordneter Port-Nummern zur Nutzung nur einer (öffentlichen) IP-Adresse zu verwenden.
Die Studenten sollen in Gruppen zu je drei oder vier Personen zusammenarbeiten. Ein Gruppenmitglied muss als Router agieren und ip_forwarding und NAT konfigurieren (siehe Grafik 10.2 und Aufgabe 1). Die anderen Gruppenmitglieder sind vor und hinter dem NAT. Um die einzelnen Netzwerksegmente zu trennen wurden verschiedene VLANs verwendet (VLAN A und VLAN B). Um die Gruppeneinteilung zu vereinfachen und die Vergabe der VLANs zu veranschaulichen werden diese in Grafik 10.1 vorgegeben. In Grafik 10.2 ist das vorgeschlagene NAT Szenario für ein einfacheres Verständnis dargestellt.

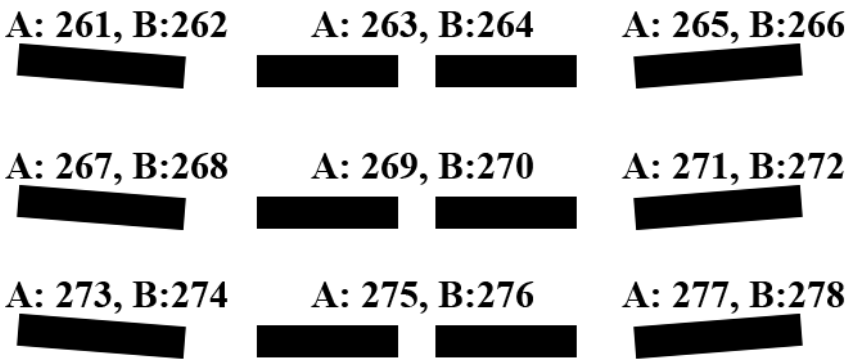


Abbildung 10.1: Gruppeneinteilung und Vergabe der VLAN IDs

10.1 Aufgaben

Question 1
Setup the network shown in the picture with VLAN ID A and B: *vconfig add eth0 A/B*. See the graphics for ID distribution. Router and Client 2 should use *dhclient3* command to get an IP from the DHCP Server. Client 1/(3) has to setup its IP address manually and set its default route to the router.
Activate IP forwarding on the router (use the command: *echo '1' > /proc/sys/net/ipv4/ip_forward*)
Configure NAT in the router machine.
(use command: *iptables -t nat -A POSTROUTING -o eth0.x -j MASQUERADE*)

Question 2
Try to test your network. Client 1/(3) sould be able to reach Client 2 and the Router, and Client 2 should be able to reach the Router.

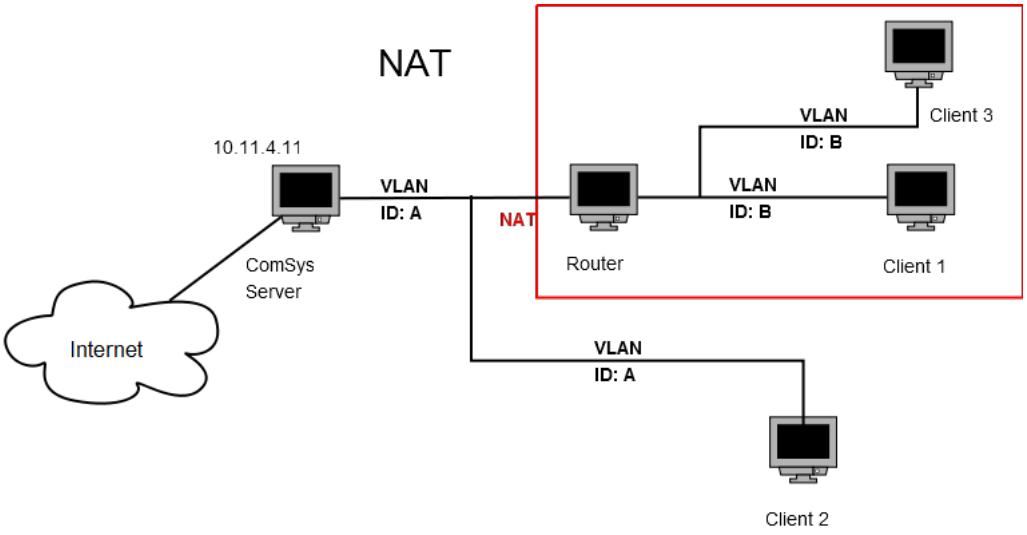


Abbildung 10.2: NAT Szenario mit vorgegebener Netzwerkstruktur

Now Client 1/(3) tries to ping client 2 and client 2 trys to ping Client 1/(3). Does it work? Why/why not? Explain your observations.

Question 3
Try to solve the problem from question 2. Configure port forwarding for SSH (port 22) and client 1 on the router (Which transport layer protocol does SSH uses? Try google.de for hints on the forwarding and the manpage iptables).
Start the ssh daemon on client 1 (use command: */etc/init.d/ssh start*) and change the root password before to avoid unsolicited remote administration! Client 2 should establish a SSH connection to client 1.

Question 4
Is it posible to configure the port forwarding for client 3 in a similar way? Wich conflict may occur? How can you solve this problem? Could you apply the same solution to allow ICMP packets to travel over the NAT router?

Question 5
IP masquerading NAT obfuscates the network structure behind the router. How it would be possible to get an idea of the network behind nevertheless? Generate traffic originating in client 1 and client 3. Look at the IP fragment numbers and the TCP sequence numbers on the outgoing interface of the router. Is it possible to see how many clients are attached behind the NAT router?

Question 6
Which ways do you have to distinguish IP packets generated directly within the masquerading router and originating from some hosts (further) behind that router? Look into the header fields of packets leaving the router into direction of the default router / Internet!

Question 7
Configure the MTU in VLAN B to 900, the router should change the MTU to 500 on the outgoing interface, same as client 2. Try to ping each other and analyse the generated traffic with wireshark. Try to ping with paket size 1400 and analyze the generated traffic again. How many

packets would arrive in client 2 if client 3 (the origin) is hidden behind a NAT.

10.2 Lösungsskizze

Question 1
Lösungsweg ist bereits in der Aufgabenstellung skizziert.

Question 2
Zum Testen der Verbindung kann das Tool „ping“ verwendet werden. Ein Ping zwischen Host 1/3 und Host 2 und dem Router ist möglich. Da NAT verwendet wird ist jedoch keine Verbindung von Host 2 zu Host 1/3 möglich.

Question 3
Da in Aufgabe 2 von außen keine Verbindung zu Hosts hinter einem Router mit aktiviertem NAT möglich ist, soll ein Portforwarding am Beispiel einer SSH Verbindung eingerichtet werden.
Befehl: *iptables -t nat -A PREROUTING -p tcp -i eth0.271 -d IPROUTER −−dport 22 -j DNAT −−to IPCLIENT:22*

Question 4
Würde das Forwarding für Client 3 auf gleiche Weise konfiguriert werden, würde es zu einem Konflikt kommen, da der Router nicht wissen kann an welchen Host er die ankommenden Pakete weiterleiten soll. Der Grund hierfür ist, dass als Kriterium für das forwarding in Aufgabe 3 nur der eingehende Port verwendet wird. Mehrere Lösungen dieses Problems sind denkbar:

- Es wird ein weiteres Kriterium für das forwarding hinzugenommen, beispielsweise die Quelladresse.
- Es wird ein anderer Port für SSH verwendet. So könnte beispielsweise Port 22 zu Client 1 und Port 222 an Client 3 weitergeleitet werden.
- Es sind auch aufwändige Mechanismen denkbar, die jedoch über ein simples Port-Forwarding hinausgehen. Denkbar wäre ein Login direkt auf dem Router und von dort eine weitere SSH Verbindung zu Client 1/3. Es ist auch eine automatische Delegation der SSH Verbindung nach einem erfolgreichen Login auf dem Router denkbar. Hierbei müsste anhand des Login-Namens unterschieden werden, ob eine Delegation auf Client 1 oder 3 stattfinden soll.

Question 5
Falls es möglich ist den Traffic vor dem NAT abzuhören, kann anhand der Felder „fragment number“ und „sequence number“ festgestellt werden ob sich mehrere Rechner hinter einem NAT verbergen.

Question 6
Pakete von Rechnern aus einem Netzwerk hinter einem Router mit aktiviertem NAT haben eine um eins verringerte TTL.

Question 7
Client 2 empfängt vier Pakete. Client 3 sendet zwei Pakete zum Router da er durch die gesetzte MTU von 900 Bytes gezwungen wird das Paket zu fragmentieren. Auf dem Router werden die Pakete wegen der MTU von 500 Bytes erneut fragmentiert.

11 ICMP

Das Internet Control Message Protocol (ICMP) dient in Netzwerken zum Austausch von Informations- und Fehlermeldungen über das Internet-Protokoll (IP).
Bei dieser Aufgabe soll versucht werden verschiedene ICMP Fehler-Codes zu erzeugen. Dabei soll auch die Bedeutung des TTL Felds im IP-Header verdeutlicht werden.

11.1 Aufgabe

Question 1
Try to ping a neighbour in the same VLAN with TTL of "1" and after that try to ping 132.230.200.200 with same TTL. Observe the generated traffic with wireshark. You should now try to produce an ICMP error packet with "Destination unreachable". Check in wireshark the field "typ" and "code". Try to produce two ICMP error packets with "Destination unreachable" and different "code" field. Explain the value in the code field.

11.2 Lösungsskizze

Question 1
Beispiel für TTL=1: *ping 10.230.4.180 -t 1*
Es kann beobachtet werden, dass beim Pingen der IP 132.230.200.200 folgende Meldung zurückgegeben wird: „Time to live exeeded". „Destination Host unreachable" kann erzeugt werden wenn einen nicht existierende IP-Adresse als Ziel angegeben wird. Es sind folgende Type 3 Codes möglich:
0 Netzwerk nicht erreichbar
1 Host (Zielstation) nicht erreichbar
2 Protokoll nicht erreichbar
3 Port nicht erreichbar
4 Fragmentierung nötig, DF gesetzt
5 Route nicht möglich (die Richtung in IP-Header-Feld Option falsch angegeben)

Beispielsweise wird eine Code 0 ICMP von dem Gateway erzeugt wenn ein Netzwerk gepingt wird, das nicht existiert.

12 Dynamic Routing

In diesem Kapitel wird das Thema Dynamic Routing anhand der beiden Routing-Protokolle RIP und OSPF veranschaulicht. Dynamische Routing-Protokolle sind notwendig, da ein manuelles Umkonfigurieren der Routing-Tabellen im Fehlerfall zu aufwendig ist. Ein Fehler im Netzwerk kann durch den Ausfall eines Routers oder einer Verbindung zwischen zwei Routern hervorgerufen werden. Dynamische Routing-Protokolle ermöglichen eine schnelle automatisierte Reaktion auf solche Situationen.
Ziel dieser Übung ist es, die grundlegende Funktion von dynamischen Routing-Protokollen zu verdeutlichen. Dabei werden alle Teilnehmer in Gruppen zu je vier eingeteilt. Jede dieser Gruppen hat die Aufgabe eine Ringstruktur und einen Uplink zum ComSys Server herzustellen. Das Netzwerk und die Gruppeneinteilung wird in Grafik 12.1 und Grafik 12.2 dargestellt. Alle wichtigen Informationen sind den Grafiken zu entnehmen. Bei OSPF wurden im Übungssystem Scripte angelegt, um die grundlegende Konfiguration zu vereinfachen. Bei beiden Routing Szenarien sollen künstliche Fehler im Netzwerk hergestellt werden, um die Reaktion der Routing-Protokolle zu beobachten.

12.1 Aufgaben

12.1.1 Routing - RIP

Question 1
First start wireshark to analyse all later produced traffic. To configure the RIP routing, connect via telnet to the rip daemon.
telnet localhost 2602, password is: ”comsys”
Type in the comands (and replace appropriatly):
enable
configure terminal
router rip
network IP_VLAN_X/24
network eth0.X
network IP_VLAN_Y/24
network eth0.Y
quit
quit

Now you can check if everything is correct with: *show ip rip*
Analyse the generated RIP packets with wireshark and take a look in your routing table. What do you observe?

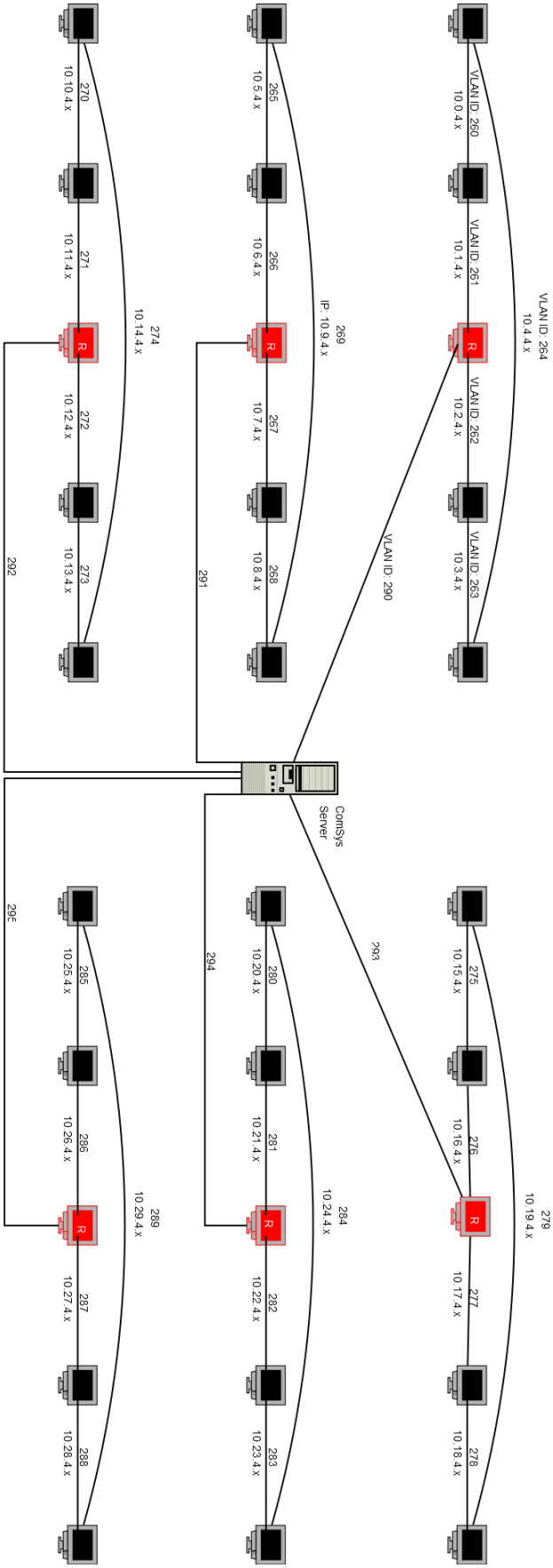


Abbildung 12.1: Netzwerkplan zum Thema Dynamic Routing mit RIP

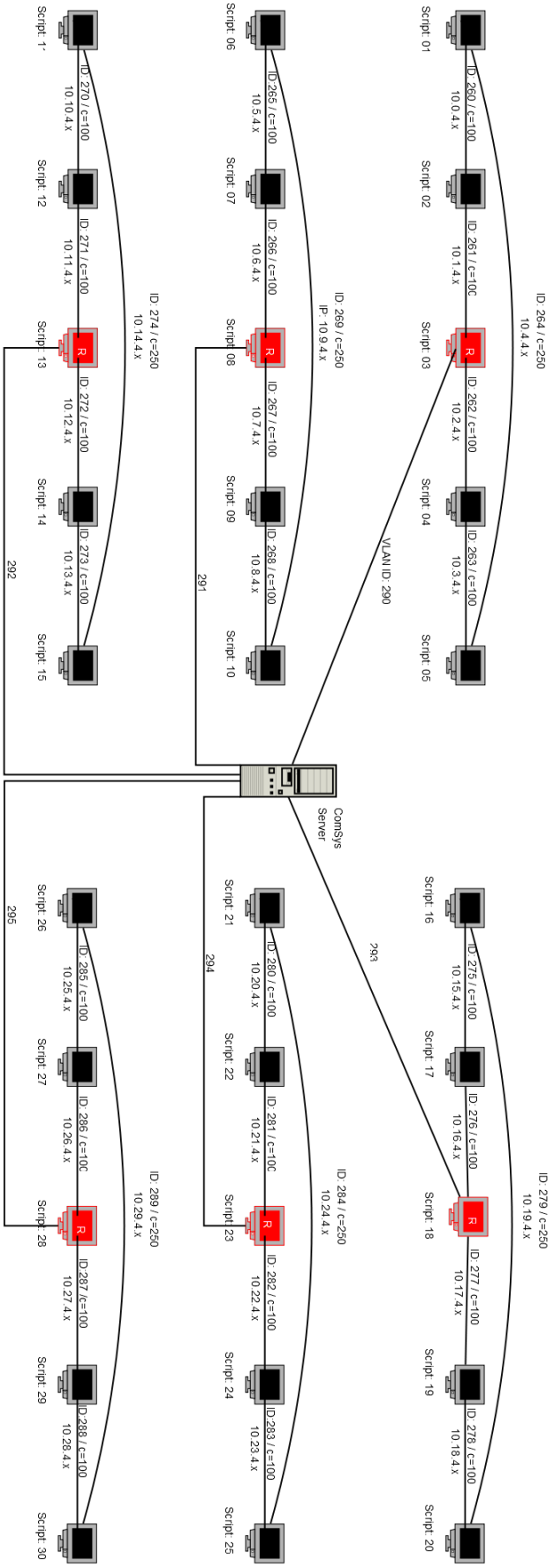


Abbildung 12.2: Netzwerkplan zum Thema Dynamic Routing mit OSPF

Question 2
Use traceroute to check the connection to your neighbors network crossing the comsys central router! Add some additional networks to your dummy interface. Use some network like 192.168.X.0/24 to distinguish from the first setup (everybody in a group should use a subnet so that it takes the whole class C network). Do not forget to switch on the kernel packet forwarding(!) See how the information propagates through your routers and check if the entries in the routing table aggregated! Check the reachability with *traceroute*. Observe the ICMP messages generated.

Question 3
Plug out one cable virtualy(!) in your network. This means one person in your group should take one eth0.VLAN down for about three minutes (use the standard commands like *ip* or *ifconfig* for it). Refresh in short intervals the routing table or in the telnet console the command *show ip rip*. What kind of changes do you observe? What happend?

12.1.2 Routing - OSPF

Question 1
First start wireshark to analyse all later produced traffic. To configure the OSPF routing, connect via telnet to the ospf daemon. (Same procedure like last lectures setup with RIP.)
telnet localhost 2604, password is: "comsys"
Type in the comands (and replace appropriatly):
enable
configure terminal
router ospf
network IP_VLAN_X/24 area 1 (don't use another area!)
network IP_VLAN_Y/24 area 1
interface eth0.X
ip ospf cost c (See path cost in the grafic)
ip ospf hello-interval 10
interface eth0.Y
ip ospf cost c (See path cost in the grafic)
ip ospf hello-interval 10
quit
quit

As router configure additional interface to the comsys server.

Now you can check if everything is correct with: *show ip ospf route* .
Analyse the generated OSPF packets with wireshark and take a look in your routing table. What do you observe?

Question 2
Use traceroute to check the connection to your neighbors network crossing the comsys central router! Add some additional networks to your dummy interface. Use some network like 192.168.X.0/24 to distinguish from the first setup (everybody in a group should use a subnet so that it takes the whole class C network). See how the information propagates through your routers and check if the entries in the routing table aggregated! Check the reachability with *traceroute*. Observe the ICMP messages generated. See the IP addresses of the lecturers machine to ping for testing!

Question 3
Play a bit with the ospf cost metric parameter. Everyone should change the cost parameter on every interface to a random number. Observe the changes in the routing table.

Question 4
Plug out one cable virtualy(!) in your network. This means one person in your group should take one eth0.VLAN down for about one minute (use the standard commands like *ip* or *ifconfig* for it). Refresh in short intervals the routing table or in the telnet console the command *show ip ospf*. What kind of changes do you observe? What happend?

12.2 Lösungsskizze

12.2.1 Routing - RIP

Question 1 & 2
Lösungen werden nicht benötigt da Lösungsweg in der Aufgabenstellung beschrieben wird.

Question 3
Es kann beobachtet werden das sich die Einträge in der Routingtabelle nach einiger Zeit ändern.

12.2.2 Routing - OSPF

Question 1 & 2 & 3
Lösungen werden nicht benötigt da Lösungsweg in der Aufgabenstellung beschrieben wird.

Question 4
Es kann beobachtet werden das sich die Einträge in der Routingtabelle nach einiger Zeit ändern.

13 IPv6

IP Version 6 ist der Nachfolger des bekannten Internet Protokolls Version 4. Der Hauptgrund für den Umstieg auf IPv6 ist die begrenzte Anzahl von IPv4 Adressen. In IPv4 stehen zur Adresskodierung 32 Bit zur Verfügung. Dies entspricht etwas über vier Milliarden Adressen. Viele dieser Adressen sind aber für spezielle Aufgaben reserviert oder bereits vergeben. Daraus ergibt sich heutzutage eine Adressknappheit die mit Einführung von IPv6 behoben wird (128 Bit Adressen). Bei den folgenden Übungsaufgaben wird der Umgang mit IPv6 Adressen verdeutlicht. Weiter soll eine Verbindung mit dem Belwü-IPv6 Tunnel hergestellt werden um eine Verbindung in das globale IPv6 Netz zu bekommen. Die Unterschiede zwischen IPv4 und IPv6 werden in einigen theoretischen Fragen behandelt.

13.1 Aufgaben

Question 1
What is the IPv6 address on your loopback (lo) interface and Ethernet interface? Compare your standard desktop system settings to the settings in your virtual machine? Which are the differences? Show where your MAC adress is in automatically assigned IPv6 address.

Question 2
What distinguishes link-local and global IPv6 addresses? How are the IPv6 addresses for the link-local type composed? Try to ping (send an ICMP packet) your neighbors machine and capture the network traffic with wireshark!

Question 3
How a global routeable address for the Ethernet interface might look like for a machine here (The IPv6 address range for the campus network is *2001:07C0:0100::/48*) ? Try to assign such an address to your Ethernet interface!(**don't use 2001:07C0:0100::1 and 2001:07C0:0100::2!**) Ask your neighbor for his/her address and try to **ping6** to that machine.

Question 4
Neighbouring discovery is the IPv6 successor for ARP in IPv4, find a command to show IPv6 neighbouring discovery! Try to ping some of your neighbors before and take a look in wireshark!

Question 5
There are several ways of getting connected to the „IPv6 world", i.e.the parts of the Internet that already use IPv6. In order to reach outside networks, IPv6 packets are sent directly over ethernet to an IPv6 router. This router tunnels them into IPv4 packets and sends them to a tunnel end-point at the universitie's internet service provider, „belwue". From there, they are sent to IPv6 desination hosts.
To connect to the IPv6 world, you have to add a valid global IPv6 address to your host and add a route to IPv6 world.

- Add the IPv6 address, type: *ip -6 addr add 2001:07C0:0100::XXX/48 dev eth0* (Replace XXX with the last block of your local ipv6 address!)
- Add a route to IPv6 gateway 2001:07C0:0100::2,
  type: *ip -6 route add 2001:07C0:0100::2 dev eth0*
- Add a route to IPv6 world via IPv6 gateway,
  type: *ip -6 route add 2000::/3 via 2001:07C0:0100::2 dev eth0*

DNS isn't working yet, because your system trys to resolve the domain name over ip version 4 (look at */etc/resolve.conf*). Add a new line: nameserver 2001:4070:101:1::2. Just try to open or ping some ip version 6 adresses like:

- 2001:7c0:0:fffg::1 (belwue gateway)
- 2001:4860:0:1001::68 (ipv6.google.com)
- 2001:200:0:8002:230:47ff:fea5:3085 (kame.net) Do you see the „dancing" turtle? ;)

Question 6
Setup a Vlan with ID 260 and use *dhclient3* command to get an ip from the dhcp server.
There is another way to get connected to IPv6 world with a public service called freenet6 (www.freenet6.net). Freenet6 sets up a tunnel between your host and freenet6 tunnel server, by using a pseudo-device sit1, which tunnels IPv6 to IPv4. The software is already installed. Do the following to make it work:

- disable the old IPv6 address: *ip -6 addr del yourip dev eth0*
- *cd /etc/tsp*, where you can find tspc.conf (everything should be configured)
- Run: *tspc -vf tspc.conf*

Now check what IPv6 address do you get from freenet6? Check: *www.kame.net*, do you see the „dancing turtle"? Or take a look at *www.ipv6.org*. Use wireshark to capture IPv6 traffic on the pseudo-device sit1 (or sit0, depends on your machine). Can you see the IPv6 packets? How is the HTTP datagram encapsulated. Now capture directly on eth0.260. Can you see the difference?

Question 7
Do a traceroute6 from your host to an remote IPv6 host. What is your route to the remote IPv6 host?

Theory questions
Homework: here some theory questions to think about!

- What are the differences in IPv4 and IPv6 fragmentation? Why were they introduced? Explain the strategies how IPv4 and IPv6 adapt to different MTU sizes!
- Explain the concept of jumbograms! Why were they introduced? Which problems might occur with higher level protocols?
- In IP v6 header, header length, type of service and header checksum were removed. Why? How does the host know the payload in the IPv6 packet should be delivered to TCP, UDP or other protocols?

13.2 Lösungsskizze

Question 1
Die Adresse kann mit *ifconfig* ausgegeben werden. Der gleiche Befehl gibt auch die MAC-Adresse aus. In der virtuellen Maschine ist zu sehen das sich die automatisch zugewiesene Adresse aus der MAC-Adresse zusammensetzt.

Question 2
Lokale Adressen beginnen mit „fe80“. Link-Local Adresse: „::1/128“. Zum pingen wird „ping6“ verwendet. Dabei ist es manchmal notwendig mit Parameter „-I“ das Interface zu spezifizieren.

Question 3
Beispieladresse: „2001:07C0:0100::a7“

Question 4
„Neighbor Discovery Protocol“ Tabelle anzeigen: *ip -6 neigbour show*

Question 5
Lösungsweg ist bereits in der Aufgabenstellung skizziert.

Question 6
Lösungsweg ist bereits in der Aufgabenstellung skizziert.

Question 7
Beispiel traceroute zu 2001:4860:0:1001::68 (google.de) Gateway: freenet6

```
root@ComSysWS08-09:/# traceroute6 2001:4860:0:1001::68
traceroute to 2001:4860:0:1001::68 (2001:4860:0:1001::68) from 2001:5c0:1000:a::567, 30 hops max, 16 byte packets
 1  2001:5c0:1000:a::566 (2001:5c0:1000:a::566)  209.305 ms  204.345 ms  207.54 ms
 2  * if-5-0-1.6bb1.mtt-montreal.ipv6.as6453.net (2001:5a0:300::5)  185.23 ms  177.602 ms
 3  if-1-0.mcore3.mtt-montreal.ipv6.as6453.net (2001:5a0:300:100::1)  191.114 ms  146.734 ms  152.828 ms
 4  if-13-0.mcore4.nqt-newyork.ipv6.as6453.net (2001:5a0:300:100::2)  165.163 ms  139.738 ms  137.349 ms
 5  2001:5a0:400:200::1 (2001:5a0:400:200::1)  145.688 ms  346.557 ms  206.894 ms
 6  2001:5a0:400:200::6 (2001:5a0:400:200::6)  175.029 ms  181.268 ms  190.165 ms
 7  2001:5a0:600:100::11 (2001:5a0:600:100::11)  195.312 ms  177.778 ms  169.471 ms
 8  2001:5a0:600::5 (2001:5a0:600::5)  168.456 ms  154.626 ms  154.955 ms
 9  pr61.iad07.net.google.com (2001:504:0:2:0:1:5169:1)  164.139 ms  180.27 ms  175.926 ms
10  * * *
11  2001:4860:0:1001::68 (2001:4860:0:1001::68)  335.251 ms  298.005 ms *
```

Theory questions

- Eine Fragmentierung der Datenpakete im Router gibt es nicht mehr. Die Größe der MTU sollte über eine „Path MTU Discovery“ ermittelt werden.
- Über eine Erweiterung des Headers ist es möglich Pakete mit einer Größe von bis zu 4.294.967.335 Byte zu verschicken, sogenannte „Jumbograms“. Dies kann zu Problemen in höheren Protokollschichten führen da dort oft nur 16 Bit für die Paketgröße vorgesehen sind.
- Das Feld „header length“ wird nicht benötigt, da der Header eine feste Länge von 40 Byte hat. Zusätzlich wurde auch das Feld „header checksum“ komplett weggelassen. Hieraus resultiert eine geringere Belastung der Router. Es gibt aber Prüfsummen in Layer 2 und 4. Das Feld „type of service“ wurde durch das Feld „Flow Label“ ersetzt. Durch das Feld „Next Header“ wird die Zuordnung von Paketen zu höheren Protokollschichten ermöglicht.

14 DNS

Aufgabe des Domain Name System ist es, eine Zuordnung zwischen Domainnamen und dazugehörigen IP-Adressen zu ermöglichen. Diese Aufgabe wird von DNS-Severn weltweit erledigt. Ein solches System ist notwendig, da sich Menschen IP-Adressen nur schwer merken können. Die DNS-Sever stellen einen hierarchischen Verzeichnissdienst dar, der durchsucht werden kann. Die meisten Anfragen an DNS-Server sind in der Form Domainnamen in IP-Adressen aufzulösen. Eine Auflösung von IP-Adressen zu Domainnamen ist jedoch auch möglich. Dieser Vorgang wird als Reverse Lookup bezeichnet.
Ziel dieser Übung ist es, die grundlegende Arbeitsweise eines DNS-Servers zu verstehen. Die Aufgaben zu Beginn sind zunächst einmal ausschließlich aus der Sicht des Clients. Im weiteren Verlauf wird von den Studenten ein Bind9 Server aufgesetzt und konfiguriert, um die an einfachen Beispielen die Funktionsweise der Zonefiles zu veranschaulichen. Dabei soll vor allem die Funktion der verschiedenen Resource Records klar werden.

14.1 Aufgaben

Question 1
First take a look in */etc/resolv.conf* of your primary host system and try to understand the entrys. (The resolv.conf in your vmware image may be empty). In your virtual machine add the DNS Server 10.230.4.1 to the file /etc/resolv.conf. The entry should look like: "nameserver 10.230.4.1" Now use the dig command in your virtual machine to trace out the domain: *informatik.uni-freiburg.de*. How many domain name server it need to query?

Question 2
Using nslookup and host command to find out the IP of the mail server of the www.uni-freiburg.de.

Question 3
Find a command to show all DNS root servers. (Don't use wikipedia ;)). What about IPv6 adresses of the nameservers?

Question 4
The comsys server has a Bind9 DNS Server for a local domain called: local.vlan. The server is configured to delegate the subdomains pXXX.local.vlan to the IPs shown in the figure above. (XXX are the 3 last character of your IP) Your exercise is to setup an bind9 server on your machine so that the comsys server can delegate the requests to you.

Step by step:
1. Install Bind with command: *apt-get install bind9*
2. Add a zone file in your */etc/bind/named.conf* file. You can use any editor you like to do that.

```
zone "pXXX.local.vlan" in {
  type master;
  file "/etc/bind/pXXX.local.vlan";
```

```
};
```

Please replace XXX with the last 3 character of your IP. (XXX musst be 3 character)
Example: 10.230.4.11 => XXX = 011
3. Then create the zone file ppXXX.local.vlanïn the directory: */etc/bind*
Here is an example zone file you can use:

```
$TTL 180000
$ORIGIN PXXX.local.vlan.
@  IN  SOA    ns.pXXX.local.vlan.   root.local.vlan.(
              2008101001            ; serial
              10000                 ; refresh
              1800                  ; retry
              604800                ; expiry
              180 )                 ; minimum

@   IN  NS    ns1
    IN  A     10.230.4.XXX ;(your own IP)
ns1 IN  A     10.230.4.XXX ;(your own IP)
```

4. After your configuration. Using the command */etc/init.de/bind9 restart* to restart the bind.
5. Now resolve the address *pxx.local.vlan* over the comsys server. Explain what happens with your request. Use wireshark to see the packets exchanged.
6. On your Bind9-Server you can setup a subdomain. Try to setup a address like:
host1.pXXX.local.vlan. Later on you might want to add one or more aliases like your given name. Check how such an entry should look like.

Question 5
Try to add reverse resolving to your DNS server configuration! (By now only the server was able to find IPs from names only.) You might want to add other resource records like MX or TXT too.

Question 6
Resolve the domain doubleclick.net over the comsys DNS Server(10.230.4.1 or 132.230.4.11). What can you see? Compare the result with the answer from the DNS Server 132.230.200.200. Try to imagine what you can do if you have the control over a DNS-Server of an big Internet Service Provider. (censorship, man in the middle attack ...)

Theory questions
Homework: Here are some theory questions to think about (might help for the preparation for the exam)!

- Question 1:
  Why do we have the maximum of 13 root name-servers (number did not change since the introduction of the DNS)? Where are these servers located (rough global position)?
- Question 2:
  What is the difference between iterative and recursive DNS queries. What kind of requests (iterative vs. recursive) do the root-servers support (why)?
- Question 3:
  What distinguishes master and slave name-servers, which role play caching name-servers? What kinds of Resource Records (RR) exist? What meanings do they have?

- Question 4:
  What extensions are made on DNS to support IPv6? What problems still exist?

14.2 Lösungsskizze

Question 1
In der Datei „resolv.conf“ werden die Nameserver angegeben, die bei der Namensauflösung gefragt werden sollen. Die Anzahl der zu fragenden Nameserver kann mit *dig +trace informatik.uni-freiburg.de* ausgegeben werden.

Question 2
host -t MX www.uni-freiburg.de

Question 3
dig +nocmd . NS +noall +answer +additional

Question 4
Lösungsweg ist bereits in der Aufgabenstellung skizziert.

Question 5
Beispiel für eine „reverse zone“:

- named.conf:

```
zone "XX.4.230.10.in-addr.arpa" {
type master;
file "/etc/bind/XX.4.230.10.in-addr.arpa";
};
```

- XX.4.230.10.in-addr.arpa:

```
$TTL 180000
@ IN SOA pXXX.local.vlan. root.pXXX.local.vlan.(
2008121603 ; serial
10000 ; refresh
1800 ; retry
604800 ; expiry
180 ) ; minimum

IN NS pXXX.local.vlan.
IN PTR pXXX.local.vlan.
```

Question 6
Wird die Adresse „doubleclick.net“ über den ComSys Server aufgelöst, wird als Antwort „127.0.0.1“ zurückgegeben. In diesem Fall kann eine Werbeeinblendung auf Webseiten verhindert werden. Es sind jedoch auch Szenarien denkbar in denen das Manipulieren von DNS Einträgen zu Zensur oder Betrugszwecken missbraucht werden kann.

Theory questions

- Question 1
  Die Anzahl der Root-Nameserver ist auf 13 begrenzt. Grund dafür ist die konservative MTU Annahme von 512 Byte, sowie bei alten Netzwerkgeräten eine nicht vorhandene Fragmentierung. 10 der Root-Server befinden sich in den USA, die anderen drei sind in Europa und Asien.

- Question 2
  Bei der iterativen Abfrage antwortet der gefragte Server mit einem Verweis auf den nächsten Server. Im Gegensatz dazu holt der gefragte Server bei einer rekursiven Anfrage die Daten selbst ab und sendet dann die Antwort an den Client zurück.

- Question 3
  Auf einem Master-Name-Sever liegen die eigentlichen Informationen. Slave-Name-Server dienen der Redundanz und zur Entlastung des Master-Severs. Caching-Nameserver speichern die Antworten der Nameserver zwischen, um die eigentlichen Nameserver zu entlasten und eine geringere Latenz zu ermöglichen.
  Es existieren viele Resource Records. Die wichtigsten sind: A (IPv4-Adresse eines Hosts), AAAA (IPv6-Adresse eines Hosts), CNAME (Alias für einen Host), MX (der für diese Domain zuständige Mailserver), NS (Hostname eines autoritativen Nameservers), PTR (Domain Name Pointer für reverse Einträge).

- Question 4
  siehe Wikipedia: *http://de.wikipedia.org/wiki/Ipv6#IPv6_und_DNS*

15 Advanced DNS

Diese Übungsaufgaben vertiefen das DNS Thema aus Kapitel 14. Dabei wird IPv6 aus Kapitel 13 aufgegriffen, um den Resource Record „AAAA“ noch einmal genauer zu betrachten. Weiter werden „Reverse-Lookups“ und „Zone Transfers“ behandelt. Beim Thema „Zone Transfers“ soll eine verschlüsselte Verbindung zwischem Master- und Slave-Server hergestellt werden.

15.1 Aufgaben

Question 1
Resolve the Domain „local.vlan“ in such a way that the result gives you the IPv6 address of the ComSys Server.

Question 2
Now try to get the domain for the address from question 1 via a reverse-lookup.

Question 3
Setup a Bind Server that resolves IPv6 addresses for the domain „host.comsys“. This time the Bind9-Server is already installed. But you have to configure the IPv6 address resolution in the file: „/etc/bind/host.comsys“. Don't forget to restart your bind-server: */etc/init.d/bind9 restart*
Now test your configuration: *dig host.comsys @127.0.0.1 -t aaaa*
If something is wrong you might take a look at the logfile: „/var/log/daemon.log“

Question 4
Now we want to have a IPv4 reverse-lookup entry in the configuration. To do so you have to add a new zone in */etc/bind/named.conf*. The zone entry should look like the following example:

```
zone "XX.4.230.10.in-addr.arpa" {
  type master;
  file "/etc/bind/XX.4.230.10.in-addr.arpa";
};
```

Replace XX with the last characters of your IP.
Now you have to create the config-file „/etc/bind/XX.4.230.10.in-addr.arpa“. Here is an example file:

```
$TTL 180000
$ORIGIN XX.4.230.10.in-addr.arpa.
@ IN  SOA host.comsys. root.host.comsys.(
      2008121603      ; serial
      10000           ; refresh
      1800            ; retry
      604800          ; expiry
      180 )           ; minimum
  IN  NS  host.comsys.
  IN  PTR host.comsys.
```

You can find this file in your vmware in */etc/bind/example.4.230.10.in-addr.arpa*
Restart your bind-server and test your configuration: *host 10.230.4.XX 127.0.0.1*

Question 5
With version 9 of the popular Bind daemon came crypto support for zone transfers. Work together in teams to build a master-slave relationship to permit only signed zone transfers.

Initial Situation
Everyone has it's own DNS servers, which we will refer to as ns1 (10.230.4.x) and ns2 (10.230.4.Y). ns1 serves as the master for the 'host.comsys' zone, and ns2 slaves this zone off the primary. Modify your configuration (*/etc/bind/named.conf*) so that it looks something like this:
Master:

```
acl "xfer" {
        10.230.4.Y;
   };

   zone "host.comsys" {
        type master;
        file "/etc/bind/host.comsys";
        notify yes;
        allow-update { none; };
        allow-query { any; };
        allow-transfer { xfer; };
   };
```

and edit */etc/bind/host.comsys*

```
$TTL 10
$ORIGIN XX.4.230.10.in-addr.arpa.
@ IN  SOA host.comsys. root.host.comsys.(
      2008121615      ; serial
      10           ; refresh
      1800            ; retry
      60          ; expiry
      60 )           ; minimum
  IN  NS  host.comsys.
  IN  PTR host.comsys.
```

Slave:
create a directory */etc/bind/slave* and *chmod 777 /etc/bind/slave* and change *named.conf* to:

```
   zone "host.comsys" {
        type slave;
        masters { 10.230.4.X; };
        file "/etc/bind/slave/host.comsys";
        notify no;
        allow-query { any; };
        allow-transfer { none; };
   };
```

Step 1 - Generate a key
The first thing that must be established is a shared secret. This is in the form of an HMAC-MD5 key that is generated using the Bind9 tool 'dnssec-keygen' as follows:

```
bash-2.03# dnssec-keygen -a HMAC-MD5 -n HOST -b 128 signed_comms
Ksigned_comms.+157+56812
bash-2.03# ls -l Ksigned_comms.+157+56812.*
-rw-------   1 root    other    56 Jul 16 21:31 Ksigned_comms.+157+56812.key
-rw-------   1 root    other    81 Jul 16 21:31 Ksigned_comms.+157+56812.private
```

...this will create two files, a .key and a .private. The secret we are interested in appears in the .key file:

```
bash-2.03# cat Ksigned_comms.+157+56812.key
signed_comms. IN KEY 512 3 157 s1PBD3jCHmteOzN80LBqVg==
```

Step 2 - Add keys to the named.conf file
The keys are simply added in a line such as this in the same way you would add any global configuration command:

```
key signed_comms { algorithm hmac-md5; secret "s1PBD3jCHmteOzN80LBqVg=="; };
```

This key must be added in the same manner to both master and slave.

Step 3 - Enabled signed transfers
Finally, you must add an entry such as this to your master:

```
server 10.230.4.Y {
  transfer-format many-answers;
  keys { signed_comms.; };
};

and similarly on the slave:

server 10.230.4.X {
  transfer-format many-answers;
  keys { signed_comms.; };
};
```

To enable secure zone transfer add to */etc/bind/named.conf.options* a new line: *dnssec-enable yes;*
Transfers between the two servers will now use this authorization and signing technique, known as TSIG. Use wireshark to analyse the traffic. The Master server can edit the *host.comsys* file and restart the bind server. After that the *host.comsys* file should be syncronised again.

Question 6
Take a look at *http://tinyurl.com/gvlpu* to get an unicode/punycode IDN converter. Play a bit around with domains like www.müler.de or some other „nice“ domains.

15.2 Lösungsskizze

Question 1
dig local.vlan -t aaaa

Question 2
host 10.230.4.1

Question 3
Beispiel Eintrag in der Datei „host.comsys“:
IN AAAA 2001:db8::6

Question 4
Lösungsweg ist bereits in der Aufgabenstellung skizziert.

Question 5
Lösungsweg ist bereits in der Aufgabenstellung skizziert.

Question 6
Keine Lösung notwendig.

16 SSH

SSH wird üblicherweise benutzt um auf sichere Art und Weise eine verschlüsselte Netzwerkverbindung mit einem entfernten Computer herzustellen. Eine SSH Verbindung bietet eine komplett verschlüsselte Alternative zu Telnet oder rlogin. Somit kann in einem unsicheren Netzwerk eine sichere, verschlüsselte und authentifizierte Verbindung zwischen zwei Rechnern aufgebaut werden. Der Server identifiziert sich dem Client gegenüber mit einem RSA Zertifikat, um möglichen Manipulationen vorzubeugen.
In dieser Übung soll durch Erstellen eines RSA Zertifikates ein automatischer Login ohne Verwendung eines Passwortes realisiert werden. Anschließend geht es darum mit Hilfe von SSH sämtlichen TCP Traffic über SSH zu tunneln, mit Hilfe von Wireshark die generierten Datenpakete genauer zu analysieren und diese mit einer unverschlüsselten Telnet Verbindung zu vergleichen.

16.1 Aufgaben:

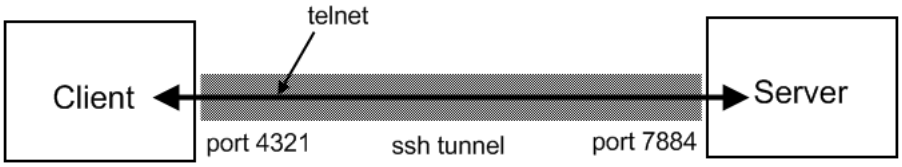
Question 1
Create an RSA key pair for SSH. This allows to autologin on an remote machine without a password or run a command remotely without password interaction.
Work together with your neighbor! One of you as server and one as client.

Setup step by step:

- CLIENT only:
  - Create an RSA-key pair: *ssh-keygen -t rsa* The default filename is ok. Don't insert a password. The created files are in /root/.ssh/ There should be two files. The public-key file id_rsa.pub and the private key file id_rsa
  - Now copy the public key file to directory /root/.ssh/ on the Server. The directory .ssh may not exist. (Use an USB stick or send it by mail or do whatever is necessary ...)
- SERVER only:
  - Create the file /root/.ssh/authorized_keys2 and insert the public-key:
    cat id_rsa.pub » /root/.ssh/authorized_keys2
  - Before starting the SSH-Deamon you should change the root password using *passwd*. Then start the SSH deamon: */etc/init.d/ssh start*
- Now test your SSH autologin on your client.

Take a look at the generated packets via wireshark. Compare the packets to the ssh login without RSA. Explain the Diffi Hellman key exchange!

Question 2
Tunneling TCP Data over SSH:
Sometimes it is useful to secure services via a secure channel (e.g. if no TLS implementation is available). To do so you can use a SSH tunnel. In this example you should tunnel cleartext data from a telnet session over a secure ssh connection. (telnet is a dumb example - just a demonstration of a clear text interaction service with insecure password transfers.) Have a look at a telnet session with password interaction (use the ”follow TCP session” feature inwireshark)!



Setup:

- User: uebung
  Password: comsys
- Open the ssh tunnel: *ssh -N -L 4321:localhost:23 uebung@10.230.4.1 -p 7884*
  This will forward the local port 4321 on your machine to the port 23 on the Server 10.230.4.1.
- Open a telnet session: *telnet localhost 4321*
  Because of the ssh tunnel all data send to your local port 4321 is send to the port 23 of the remote machine.

Take a look at the generated packets via wireshark. Compare the generated packets to the packets from a unencrypted telnet session.
Is it possible to tunnel a ppp over a ssh connection? Explain what happens in the protocol stack.

16.2 Lösungsskizze

In diesen SSH Aufgaben ging es vor allem darum, nachzuvollziehen wie ein automatischer Login mittels SSH möglich ist. Sämtliche dafür benötigten Arbeitsschritte sind detailliert beschrieben und mussten lediglich nachvollzogen werden. In Aufgabe 2 wird der Unterschied zwischen einer verschlüsselten und unverschlüsselten Verbindung in Wireshark verdeutlicht.
Eine ppp-over-ssh Verbindung ist möglich, in dem der gesamte Netzwerkstack getunnelt wird.

17 Open VPN

OpenVPN ist ein Programm zum Aufbau eines Virtuellen Privaten Netzwerkes (VPN) über eine verschlüsselte TLS Verbindung. Zur Authentifizierung stehen in OpenVPN mehrere Methoden zur Verfügung. In den folgenden Aufgaben geht es zunächst einmal darum eine VPN Verbindung mit Hilfe eines „pre-shared key“ (einem statischen Schlüssel/Passwort) aufzubauen. Anschließend soll eine Zertifikat basierte Authentifizierung erfolgen. Der Server und die jeweiligen Nutzer besitzen je ein eigenes Zertifikat (öffentlich/privat). Der OpenVPN-Server lässt nur Verbindungen zu, die mit Hilfe von einer ihm bekannten Zertifizierungsstelle signiert wurden.

17.1 Aufgaben

Question 1
In the first part of the exercise a vpn connection is established using a pre-shared key. In the second part of the exercise certificated authentification should be used.

a) pre-shared key
The necessary config files for openvpn are located in */usr/share/doc/openvpn/*

- The Server generates a secret key: *openvpn −−genkey −−secret secret.key*
- Give the generated file to the client.
- Server and client have to edit there config files as following:
  x = server IP
  y = client IP
  - Server: */.../openvpn/examples/sample-config-files/server.conf*

    ```
    secret /.../secret.key
    ifconfig 10.8.0.x 10.8.0.y
    ```
  - Client: */.../openvpn/examples/sample-config-files/client.conf*

    ```
    remote SERVERIP
    secret /.../secret.key
    ifconfig 10.8.0.y 10.8.0.x
    ```
- Use the following commands to start the server and client:
  - Server: *openvpn /usr/share/doc/openvpn/examples/sample-config-files/server.conf*
  - Client: *openvpn /usr/share/doc/openvpn/examples/sample-config-files/client.conf*

Now test your setup. Explain what packets are transmitted. Which protocol is used? Which interface is used for the secure channel? Have a look at the MTU size of your secure channels interface!

b) certificate based
Now you have to create certificates to secure the communication.

- First you should edit the file /.../openvpn/examples/easy-rsa/2.0/vars . Here is an example:

```
export KEY_COUNTRY="DE"
export KEY_PROVINCE="BW"
export KEY_CITY="Freiburg"
export KEY_ORG="vpn-test"
export KEY_EMAIL="mail(at)vpn.rz.uni-freiburg.de"
```

- Now you have to add the vars file to the source files: *source ./vars*
- Clean previous keys *./clean-all* and build the master certificate and master key *./build-ca*
- This master certificate is now needed to sign the server certificate. To do this follow the setup *./build-key-server server*
- Now you have to create the client keys.

  ```
  ./build-key client1
  ./build-key client2
  etc.
  ```

  But watch out: during the setup the "common name" have to be client1 or client2 etc.
  To transfer secure keys over insecure communications channel the Diffie Hellman protocol is used: *./build-dh*
- Take a look at the files in */.../openvpn/examples/easy-rsa/2.0/keys* This folder should have the following files now:

  ```
  ca.crt
  client1.crt
  client1.key
  client2.crt
  client2.key
  etc.
  ```

- At the next step, you have to configure the openvpn-server. The configuration files are in */.../openvpn/examples/sample-config-files/server.conf* . Server: First remove the added lines from exercise part a: secret ... and ifconfig... After that add the following lines to the config file:
  x = network range to assign IP addresses to the clients.

  ```
  server 10.8.x.0 255.255.255.0
  ca /.../openvpn/examples/easy-rsa/2.0/keys/ca.crt
  cert /.../openvpn/examples/easy-rsa/2.0/keys/server.crt
  key /.../openvpn/examples/easy-rsa/2.0/keys/server.key
  dh /.../openvpn/examples/easy-rsa/2.0/keys/dh1024.pem
  ```

- Excange the files: *ca.crt, clientX.crt and clientX.key* with your clients.
- Client: First remove the added lines from exercise part a: secret ... and ifconfig... After that add the following lines to the config file:

```
client
remote OPENVPNSERVER-IP 1194
ca ...ca.crt
cert ... clientX.crt (x = 1,2....)
key ... clientX.key
nobind
```

- Start server and client:
  Server: *openvpn /usr/share/doc/openvpn/examples/sample-config-files/server.conf*
  Client: *openvpn /usr/share/doc/openvpn/examples/sample-config-files/client.conf*

If everything works fine a vpn connection is established now!? Test your connection. Can you see a difference to question a)? Explain the certificate-based authentication.
Work in groups: Configure your OpenVPN server to allow multiple clients to connect! Check connectivity between the several clients - how the packets are exchanged? What is the delay difference compared to plain IP operation?

17.2 Lösungsskizze

Question 1
Diese Aufgabe kann in Zweiergruppen (Client und Server) gelöst werden. Für die „pre-shared-key“ Authentifizierung und der anschließend zertifikatbasierten Authentifizierung sind sämtliche Arbeitsschritte angegeben und können schrittweise umgesetzt und nachvollzogen werden.

a) Zur Kommunikation wird von OpenVPN ein eigenes tun0 Device erstellt. Der Datenaustausch erfolgt über das UDP Protokoll. Folgende Abbildung veranschaulicht diesen Vorgang genauer.

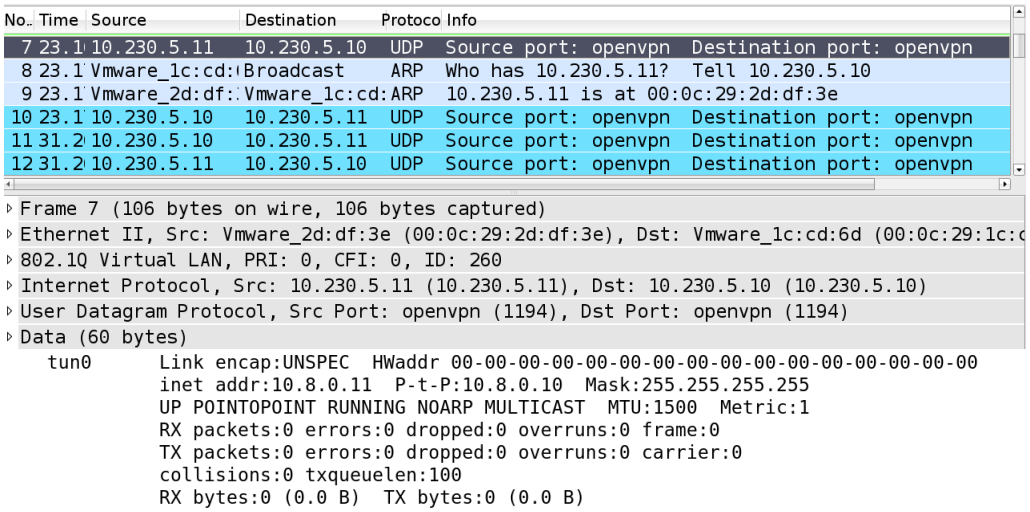


Abbildung 17.1: Wireshark Screenshot und tun0 Device

b) Hauptunterschied: mehr Pakete, da mit Hilfe von Diffie Hellman Protokoll die Schlüssel ausgetauscht werden. Anschließen erfolgt eine Zertifikatsbasierte sichere Kommunikation.

18 SSL

Bei SSL handelt es sich um ein hybrides Verschlüsselungsprotokoll zur Datenübertragung im Internet, welches primär bei HTTPS eingesetzt wird. SSL wird inzwischen standardisiert unter dem Namen TLS weiterentwickelt.
Zunächst einmal geht es darum, sich mit der grundlegenden Funktionsweise von SSL vertraut zu machen, um anschließend eine Client-Server Verbindung, auf Basis von SSL Zertifikaten, aufzubauen.

18.1 Aufgaben

Question 1
With openssl it is possible to interact in a SSL handshake with an arbitrary server. The following comand is used:

openssl s_client -connect <host>:<port>

The client conencts to the server and waits for inputs (like telnet). You can exit the client with Q + <Return>.
Establish a SSL handshake with the following servers and take a look at the output:
www.verisign.com:443
132.230.4.11:443
Connect with a browser to the given servers. (https://www.verisign.com and https://132.230.4.11:443) What can you see? Is it nesecarry to accept the certificate from www.verisign.com? Why/Why not? Would you trust the certificate from 132.230.4.11?

HTTPS, Certificates and PKI

Question 2
We are going to create a client-server authentificatin for a HTTPS connection. A both side SSL certificated authentification is used.
Create certificates for new users
1. We create a new key

```
openssl genrsa -des3 -out secret.key 1024
Generating RSA private key, 1024 bit long modulus
....++++++
..........++++++
e is 65537 (0x10001)
Enter pass phrase for secret.key:
Verifying - Enter pass phrase for secret.key:
```

We produced the key file secret.key. Use a random password.
2. We create a new request for certificate (Certificate Signing Request CSR)
openssl req -new -key secret.key -out secret.req

Enter pass phrase for secret.key: You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank, for some fields there will be a default value. Use as organisation name: RZ.
The just created request file *secret.req* have to be signed by the CA. The Comsys server (132.230.4.11) will act as CA. Rename the *secret.req* to *yourname.req*. Therefore you have to upload the *yourname.req* via ftp. (ftp 132.230.4.11, user: uebung, password: comsys, type ? for further help). Now contact a hiwi and wait... After several time you will find a file *certificateyourname.pem*. Download this file. Now change the format of the certificate into PKCS12 (format for entering a certificate into a browser). This format combines both the certificate (to be sent to the partner) and the private key (that remains secret). Use the follwoing command:

openssl pkcs12 -export -in certificateyourname.pem -inkey secret.key -out cert-yourname.p12

Import the file cert-yourname.p12 into the browser (procedure depends on the browser, in firefox take a look at *Edit, Preferences, Advanced, Encryption, View Certificates, ...*).
Now open the adress 132.230.4.11 in firefox. What happens? Take a look at the certificates. Can you explain what happens?

18.2 Lösungsskizze

Die genauen Lösungswege sind bereits in den Aufgabenstellungen skizziert.
In der ersten Aufgabe geht es darum sich mittels SSL zu zwei verschiedenen Servern zu verbinden und den entstehenden SSL Handshake genauer zu analysieren. Mittels eines Webbrowsers sollen anschließend die zwei verschiedene SSL Zertifikate dieser Server genauer betrachtet werden. Dabei sollte auffallen, dass dem Zertifikat von *verisign.com* – im Gegensatz zu einem selbst generierten Zertifkat – standardmäßig von den meisten Browsern vertraut wird.
Die zweite Aufgabe beschreibt Schritt für Schritt wie eine beidseitige Client-Server Authentifizierung realisiert werden kann. Hierzu wurde eine Testwebseite erstellt, die nur mittels gültigem Zertifikat aufgerufen werden kann. Ohne gültiges Zertifikat ist das Aufrufen dieser Testwebseite nicht möglich. Um ein gültiges Zertifikat zu erstellen, wird zunächst von jedem Studenten ein Schlüssel erzeugt, welcher anschließend manuell signiert wird. Nach dem importieren des Zertifikates in einen entsprechenden Webbrowser, wie zum Beispiel Firefox, sollte eine authentifizierte Verbindung zu der entsprechenden Webseite möglich sein.

19 GnuPG

Mittels GnuPG ist die Aufgabe Dateien zu signieren und zu verschlüsseln. GnuPG oder GPG ist ein freies Kryptographiesystem. Es dient zum Ver- und Entschlüsseln von Daten sowie zum Erzeugen und Prüfen elektronischer Signaturen.

19.1 Aufgaben

GnuPG allows to encrypt and sign your data and communication, features a versatile key managment system as well as access modules for all kind of public key directories. For the commands take a look in the man page. All generated files could be found in */root/.gnupg*.

Question 1
Create with the gpg command a key pair (public and private key). It should be a DSA Key with a length of 1024 Bit for signing and crypting data. You can decide how long the key should be valid.

Question 2
Create a text file. Sign this file with a cleartext signature. Take a look at the created *file.asc*. Verify the just created file.

Question 3
Export your public key and exchange it with your neighbour. Your neighbour should import the key. Now encrypt the text file from question 3 with the public key from your neighbour. (Use the signed or unsigned File). Is there a advantage to encrypt a signed message additional? Decrypt the file with your private key.

19.2 Lösungsskizze

Question 1
Ein gültiges Schlüsselpaar kann mit folgendem Befehl erzeugt werden: *gpg –gen-key*

Question 2
Zum Erstellen und signieren einer Datei sind diese Befehle notwendig:
echo „Das ist eine zu signierende Testdatei“ > signed.txt
gpg −−clearsign text.txt
gpg −−verify text.txt.asc

Question 3
Das Ver- und Entschlüsseln dieser Datei erfolgt durch folgende Befehle:
gpg −−output mykey.key −−export
gpg −−import other.key
gpg −−recipient „Empfänger“ −−encrypt nachricht.txt
\\Empfänger = Name bei gpg −−list-public-key
gpg −−output nachricht.klartxt −−decrypt nachricht.txt.gpg

20 IPsec

IPSec ist eine Sicherheitsarchitektur, die das IP-Protokoll mittels unterschiedlicher Protokolle und Techniken hinsichtlich verschiedener Sicherheitsaspekte erweitert. Diese setzen aber nach wie vor das IP-Protokoll auf der Transportschicht ein. Zur Authentifizierung können das sog. „Manual Keying“, Pre-Shared Keys (PSK) oder Zertifikate verwendet werden. Zur Verschlüsselung des IP-Pakets verwendet IPsec symmetrische Verschlüsselungsalgorithmen, wie zum Beispiel DES, 3DES, BLOWFISH oder AES.
Um die Funktionsweise von IPsec nachvollziehen zu können soll zunächst einmal eine Verbindung zum zentralen IPsec Universitätsserver aufgebaut und mittels Wireshark analysiert werden. Anschließend wird eine verschlüsselte Verbindung zwischen zwei PCs mittels manuell ausgetauschten Schlüsseln hergestellt. Alternativ dazu geht es in Aufgabe 3 darum mittels dem Tool racoon eine Pre-Shared Key Verschlüsselung umzusetzen und in Aufgabe 4 den Schlüsselaustausch durch Zertifikate zu realisieren. In der letzten Aufgabe geht es darum eine IPsec Verbindung zwischen einem PC mit Linux als Betriebssystem und einem PC mit Windows als Betriebssystem aufzubauen.

20.1 Aufgaben

Question 1
The aim is to setup an IPsec connection to the university IPsec server. You find a program called kvpnc in your virtual machine. Start it an create a new profile with the following configuration parameters:

```
Typ: Cisco (free)
enter data manually
username + password (your RZ account/Ras password)
IPSec-ID: home
grouppassword: home
network device: vlan 260
VPN-Gateway:  home-rz.vpn.uni-freiburg.de
```

Start wireshark on your VLAN device and connect to the IPsec server. What do you observe? Take a look at your routing table and network devices. Have a look what different kind of packets generated during session setup and operation. Why does this concept does not offer perfect security? You could try to run the same experiment with the original Cisco protocol driver (homework).

Question 2
Now we want to secure a connection between two computers with a simple manually exchanged key file (shared secret). Use the standard Linux kernel IPsec implementation. Work together in teams of two.

- Create keys for ”manual key exchange” (only one of you has to do this)
  - For AH SAs we need two 128 bit keys; create them with follwing command: *dd if=/dev/random count=16 bs=1| xxd -ps*

  - And for ESP SAs we need two 192 bit keys:
    dd if=/dev/random count=24 bs=1| xxd -ps

- Configuring the connection:
  Edit the configuration file */etc/setkey.conf* and insert the correct IP adresses and just generated keys.

See the following example configuration files for 2 computers with ip adresses 10.37.13.15 and 10.37.13.16.

```
# Configuration for 10.37.13.15
# Flush the SAD and SPD
flush;
spdflush;

# Attention: Use this keys only for testing purposes!
# Generate your own keys!

# AH SAs using 128 bit long keys
add 10.37.13.15 10.37.13.16 ah 0x200 -A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;
add 10.37.13.16 10.37.13.15 ah 0x300 -A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;
# ESP SAs using 192 bit long keys (168 + 24 parity)
add 10.37.13.15 10.37.13.16 esp 0x201 \
  -E 3des-cbc 0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 10.37.13.16 10.37.13.15 esp 0x301 \
  -E 3des-cbc 0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;
# Security policies
spdadd 10.37.13.15 10.37.13.16 any -P out ipsec # on PC2 change "out" to "in"
  esp/transport//require
  ah/transport//require;

spdadd 10.37.13.16 10.37.13.15 any -P in ipsec # on PC2 change "in" to "out"
  esp/transport//require
  ah/transport//require;
```

Attention: Revert the configuration on the second machine: Switch ”out” to ”in” and ”in” to ”out” on the other PC!
Now use the command: *setkey -f /etc/setkey.conf* (on both PCs) and try to ping each other. Analyse the generated traffic with wireshark. What do you observe? Compare it to the packets generated in the first question!

Question 3 - Pre Shared Keys (PSK)
To established a PSK connection we use racoon. You will need the configuration files: */etc/setkey.conf* , */etc/racoon/psk.txt* and */etc/racoon/racoon.conf*

- edit */etc/racoon/psk.txt*:
  10.230.4.x meinpassword #(x = IP from your neighbour)

- edit */etc/setkey.conf:*

  ```
  # Configuration for 10.37.13.15
  # Flush the SAD and SPD
  flush;
  spdflush;
  # Security policies
  spdadd 10.37.13.15 10.37.13.16 any -P out ipsec # on PC2 change "out" to "in"
  ```

```
esp/transport//require
ah/transport//require;

spdadd 10.37.13.16 10.37.13.15 any -P in ipsec # on PC2 change "in" to "out"
esp/transport//require
ah/transport//require;
```

Attention: Revert the configuration on the second machine: Switch ”out” to ”in” and ”in” to ”out” on the other PC!

- edit */etc/racoon/racoon.conf:*

```
path pre_shared_key "/etc/racoon/psk.txt";

remote 10.37.13.16 {
  exchange_mode main;
  proposal {
    encryption_algorithm 3des;
    hash_algorithm md5;
    authentication_method pre_shared_key;
    dh_group modp1024;
  }
}

sainfo address anonymous {
  pfs_group modp768;
  encryption_algorithm 3des;
  authentication_algorithm hmac_md5;
  compression_algorithm deflate;
}
```

Restart racoon with */etc/init.d/racoon restart* and initial SAD und SPD with the command *setkey -f /etc/setkey.conf*. Try to ping each other and analyse the generated traffic with wireshark again. What do you observe ?

Question 4 - Using Certificates
For advanced/fast participants: One of the uses of certificates is to secure IPsec connection. With the standard Linux and Windows IPsec implementations it is possible to configure secure that way too. Change your setup above and configure your same two Linux machines to use certificates for the key exchange. Look for online resources on configuration and setup!

Question 5 - Interoperability
For advanced/fast participants or homework: Try to configure an IPsec connection between a Linux and a Windows machine. You might want to use one of the several WindowsXP images available on your experiments machine. The Windows system should act as a road warrior because is configured to run in NAT mode (remember earlier practical sessions with NAT). One of the experimenter should start a machine with: `run-vmware.sh /var/lib/vmware/vmconfigs/winxppro-****.xml`. Please be careful with reboots which might be needed for Windows - do not switch off the virtual machine. Look for online resources on configuration and setup!

20.2 Lösungsskizze

Question 1
Mittels dem Programm kvpnc kann ein neues Verbindungsprofil angelegt und mit Hilfe der angegebenen

Konfigurationsparameter eine IPsec Verbindung aufgebaut werden.

Question 2
Mittels angegebener Arbeitsschritte kann Schritt für Schritt eine sichere Verbindung zwischen zwei Computern aufgebaut werden. Hierzu wird zunächst manuell ein Schlüssel generiert, der anschließend untereinander ausgetauscht wird und mit dessen Hilfe die Verbindung aufgebaut werden kann.

Question 3
Nach Anpassung der Konfigurationsdateien des Programms racoon, kann mittels eines „Pre shared keys“ eine verschlüsselte Verbindung aufgebaut werden. Die dafür notwendigen Konfigurationsparameter sind exemplarisch für zwei PCs angegeben und müssen lediglich leicht modifiziert werden.

Question 4
Wie es möglich ist mittels Zertifikaten einen gemeinsamen Schlüssel auszutauschen und anschließend eine IPsec Verbindung aufzubauen kann folgender Quelle entnommen werden:
http://www.bndlg.de/kursangebote/aglinux/index.php?title=IPSEC#Tunnel_Mode_mit_Zertifikaten

Question 5
Eine genaue Konfigurationsanleitung kann unter folgender Adresse eingesehen werden:
http://www.bndlg.de/kursangebote/aglinux/index.php?title=IPSEC#Tunnel_Mode_mit_Zertifikaten

21 IPTABLES

Ziel der nachfolgenden Aufgaben ist es, sich mit grundlegenden Funktionen von IPTABLES vertraut zu machen, verschiedene Firewall Szenarien durchzuspielen und Pakete zu manipulieren. Hierzu sollen notwendige Regeln selbständig erstellt werden.
Netfilter ist eine Freie Software innerhalb des Linux-Kernels, die es erlaubt, Netzwerkpakete abzufangen und zu manipulieren. Es bildet damit das Herzstück einer Linuxfirewall. Xtables stellt die Tabellenstruktur zur Regelmanipulation bereit und kann durch das Programm Iptables konfiguriert werden.

21.1 Aufgaben

For references please have a look at the web page put online at:
http://www.ks.uni-freiburg.de/download/inetworkSS05/practical/references/pra-ref06.html

Reminder: Please note that wireshark listens on the interface before the packets are filtered. Thus, incoming packets that get dropped are still shown in wireshark.

Question 1
First use command iptables -L, iptables -t nat -L, to look at the network traffic control rules in your host. Now block ping (ICMP) by using iptables. Try to ping anyone. Which additional consequences you or hosts connecting could suffer, if you block ICMP? Can you figure out how to configure iptables so that you are able to ping, but can't be pinged?

Question 2
Use command iptables to block all outgoing web traffic (TCP, destination port 80). Try to connect to any website. Can you get connected? Now try to connect to http://eldorado.uni-dortmund.de:8080/, why can you connect? Is there a way to block this? Delete the blocks you did, so that you can access web again.

Question 3
Start a ssh server (*/etc/init.d/ssh start*). Try to block incoming connections to that server (port 22) using iptables! What is the difference between DROP and REJECT? Check it by connecting from another computer to the host that blocks the connections. Setup a firewall rule that just your partner(s) in the experiments session is able to connect to your machine but all the other "administrators" are blocked (this could be a trivial alternative to keep unwanted interference out instead of changing the password). Which alternatives you could think of to use instead of the source IP address?

Question 4
Now you should only allow one single ssh connection/host at the same time (the next session can connect from the moment on the first ended only).

Question 5
Now let's be evil ;) and fake our address when we ping. Remember IP Spoofing? For example, use iptables to modify your IP address to 10.230.4.x when sending out ICMP requests. Ping a friend and see what happens. (Your friend should take a look in wireshark, too). After that use as IP address 10.230.4.1 and analyse the generated traffic with wireshark. Are there any differences if you take an IP address which is in use elsewhere in the experiments subnet compared to an unclaimed address?

Question 6
Use nmap to scan for open ports on your neighbours machines. Which open ports do you observe? You should create a whitelist that blocks all incoming traffic except all allowed traffic.

1. Create an exception that allows only incoming traffic if there exists an outgoing connection.
2. Block all incoming traffic on any port and any protocol.

Question 7
Try to find out how many packets matched a certain rule of your firewall setup!

21.2 Lösungsskizze

Question 1
block ICMP: *iptables -A INPUT -p icmp -icmp-type ping -j DROP*

Question 2
iptables -A OUTPUT -p tcp −−dport 80 -j DROP
IPTABLES löschen: *iptables -D OUTPUT -p tcp −−dport 80 -j DROP*

Question 3
iptables -A INPUT -p tcp −−dport 22 -j DROP/REJECT

Question 4
iptables -A INPUT -p tcp −−dport 22 -m connlimit −−connlimit-above 1 -j DROP/REJECT

Question 5
iptables -t nat -A POSTROUTING -o eth0.260 -p icmp -j SNAT −−to 10.230.4.x

Question 6
zu 1.: *iptables -A INPUT -p all -i eth0.260 -m state −−state RELATED,ESTABLISHED -j ACCEPT*
zu 2.: *iptables -A INPUT -p all -i eth0.260 -j DROP*

22 QoS

QoS steht für Qualitiy of Service und soll gewährleisten, dass alle Bestandteile eines Telekommunikationsnetzes gut miteinander zusammen arbeiten. Unterschiedliche Dienste haben unterschiedliche Anforderungen. Für reine Dateitransfers ist üblicherweise der Gesamtdurchsatz der entscheidende Parameter, die individuelle Latenz und Verlustrate hingegen sind hier weniger von Bedeutung. Für Echtzeitkommunikation wie z. B. Voice-over-IP hingegen spielen die Latenz, der Jitter und die Verlustrate eine weitaus größere Rolle, weil sie maßgeblich die Sprachverständlichkeit beeinflussen. Ziel von QoS ist es, bestimmt Pakete zu priorisieren um ein definiertes Ziel zu erreichen. Die kann zum Beispiel die minimierung der Latenz sein.
Die folgenden Aufgaben zu QoS können von zwei Studenten bearbeitet werden. Ein PC dient dabei als Sender, der andere als Empfänger von Daten. Die Bandbreite soll dabei auf 100Kbps eingeschränkt werden. In den weiteren Aufgaben geht es um die Priorisierung bestimmter Dienste bzw. Datenpakete, unter anderem SSH und ICMP. Abschließend soll ein gleichzeitiger up- und download mittels QoS ermöglicht werden.

22.1 Aufgaben

Tc (Traffic Control) allows advanced bandwidth control in Linux systems. It supports few queuing disciplines and is capable of traffic shaping. In this way we can bring Quality of Service guarantees into our system.
Reference information about tc utility and queues from Iproute2 toolset in on *http://ds9a.nl/lartc/*, chapter 9 (Queuing Disciplines for Bandwidth Management).

Question 1
You will need two machines for this one, A and B. For testing bandwidth we'll use netcat utility (issue man netcat for info).

For testing throughput you can do following - on Machine A (sender):
dd if=/dev/zero bs=1024k count=1024 | time netcat DST 1234
where DST is IP address of machine A.

On Machine B (reciever):
netcat -l -p 1234 > /dev/null

We want to limit traffic so that download speed from machine A to around 100Kbps. For this you should use Hierarchy Token Bucket (HTB) which is a queuing discipline that allows shaping (type man tc-htb for details). For just limiting speed, you can attach qdisc to the root node and set rate for HTB to „100kbps“.

To do this use the following commands:

```
//we build this structure:
                      1:0   root qdisc
                       |
                      1:1
                       |
                       |
                       |
                       |
                     1:10


//add root node

tc qdisc add dev eth0.260 root handle 1:0 htb default 10

// add children nodes
tc class add dev eth0.260 parent 1:0 classid 1:1 htb rate 100kbit ceil 100kbit
tc class add dev eth0.260 parent 1:1 classid 1:10 htb rate 100kbit ceil 100kbit

//mark packets
iptables -A POSTROUTING -t mangle -o eth0.260 -p all -j MARK --set-mark 10

//Zuordnung markierter Pakete zu den Klassen
tc filter add dev eth0.260 parent 1: prio 0 protocol all handle 10 fw classid 1:10
```

Start the ssh daemon on Machine B (/etc/init.d/ssh start). Now generate a lot of traffic with command:
On Machine A:
dd if=/dev/zero bs=1024k ***count=10240*** *| time netcat DST 1234*

On Machine B (reciever):
netcat -l -p 1234 > /dev/null
Now machine A should try to ping machine B or open a ssh connection. What do you observe?
Now delete everything:
tc qdisc del dev eth0.260 root
iptables -t mangle -F

Question 2
In */root* you'll find 2 scripts: script.sh (will set QoS on Machine A) and optionally stat.sh (gives statistics about qdisc, classes, filters and iptables). For testing bandwidth we'll use netcat utility and ssh. The script will prefer ssh packets.
Enable QoS on Machine A: *./script.sh*
Generate Traffic:
On Machine A: *dd if=/dev/zero bs=1024k* ***count=10240*** *| time netcat DST 1234*
On Machine B (reciever): *netcat -l -p 1234 > /dev/null*

And now Machine A should try again to open a ssh connection and ping Machine B. Machine A can execute the *./stat.sh* script to show some informations. Play a bit with ssh connections. Take a look in the script.sh to explain the difference to exercise 1.

Question 3
Your ping should be about 100-500 ms. Edit the ./script.sh to prefer ICMP packets. Execute the script again. If everthinmg works fine you should have ping times about 1 ms.

Question 4
First delete all settings again.
Downloading and Uploading at the same time can be a problem. We are going to solve it. But first try to understand what is causing the problem.
1. Use the wget command to download a file from the comsys server.
(*wget http://10.230.4.1:8111/file.txt*)
2. Machine A should send data to Machine B (use the netcat command again).
3. Now you should observe some ßpeed problemsïn wget. Can you explain why?
4. To achive a good troughput for incoming data it is necessary to prior the *ack-Packets*, so do it ;).
5. Now test your connection with priored ack packets again.

22.2 Lösungsskizze

Question 1
Ziel ist es die Downloadgeschwindigkeit auf „100kbps“ zu limitieren. Der Lösungsweg dazu ist bereits in der Aufgabenstellung skizziert. Eine zusätzliche SSH Verbindung oder ein ping wird bei gleichzeitigem hohem Traffic nicht möglich sein.

Question 2
Mittels eines vorgegeben Scriptes werden SSH Pakete priorisiert. Ziel ist es diese Priorisierung genauer anzuschauen und nachzuvollziehen. Der Lösungsweg ist bereits in der Aufgabenstellung skizziert.

Question 3
ICMP Packete können mittels folgender iptables Regel priorisiert werden:
iptables -A OUTPUT -t mangle -p icmp -j MARK −−set-mark 1

Question 4
Im vorliegenden Szenario sendet ein PC eine große Datei an einen weiteren PC und läd gleichzeitig von diesem eine große Datei herunter. Bei einem gleichzeitigen up- und download entsteht das Problem, dass irgendwann eingehende TCP-Packete nicht mehr schnell genug mit einem „Acknowlage“-Paket (Bestätigung) beantwortet werden können, da sämtliche Bandbreite durch das Senden einer Datei bereits verbraucht wird. Um dieses Problem lösen zu können, können bestimmte ACK-Pakete mittels einer iptables Regel priorisiert werden. Dies ist mit folgendem Befehl möglich:
iptables -A OUTPUT -t mangle -p tcp −−tcp-flags SYC,ACK,FIN,RST ACK -j MARK −−set-mark 1

23 Voice over IP

Unter Voice over IP (kurz VoIP) versteht man das Telefonieren über Computernetzwerke mittels des Internet Protokolls (IP). Je nach Kontext wird von IP, Internet, DSL oder LAN Telefonie gesprochen. Die zugrunde liegende Technik ist dabei immer dieselbe und wird als Voice over IP bezeichnet.
In Zweierteams soll eine direkte SIP Verbindung über die IP Adresse hergestellt werden. Hierzu kann eines der vorgegebenen Programme wie Zoiper oder Edika verwendet werden. Mit Hilfe von Wireshark sollten alle durch das Sip Protokoll generierten Pakete genauer analysiert und der genaue Verbindungsaufbau nachvollzogen werden. Anschließend erfolgt ein Verbindungsaufbau zu einem zentralen Server, über den ein Telefonat ins deutsche Festnetz geführt wird. Drei Personen sollen am Ende ein klassisches man-in-the-middle Szenario durchspielen, wobei zwei davon ein Gespräch führen und die dritte Person mittels eines Voice over IP Sniffers das unverschlüsselte Gespräch abhört und aufzeichnet.

23.1 Aufgaben

Question 1
Check for **twinkle** or **ekiga** in your desktop environment! Try to configure one of these tools or a tool on your personal/mobile machine to create a direct SIP connection to your neighbor (using the IP addresses). Which problems might occur in NAT scenarios?
Try to make a phone call after the setup! Capture the traffic and check for the several SIP messages exchanged between the two machines! How is the connection setup done? What standard is used for the audio encoding in your setup?

Question 2
We have setup an SIP registrar server/proxy for SIP at the IP address `132.230.4.8`. You might logon to this server using the credentials comsysNN, cseNN (username and password, `NN` equals to `01` to `10`). Use the Comsys Server as proxy (10.230.4.1). Why is this necessary? You might place a call to the German landline telephony network. You have to add the prefix ”10” to the number you want do dial (e.g. 100761-2034631 - telephone in room -111. You might use the number 1557106 with Freiburg prefix too).

Question 3
Try to find out which codec is used in your SIP connection! Which codec you would expect for a connection to the PSTN? Why? Which data rates are generated for each direction in a VoIP call using this codec? Which (dis)advantages might have the exchange of this codec for the GSM codec?
Change the codecs in a way that a-law or gsm codec is used for the call! Compare the generated data rates. Which role does the SDP plays in session setup? In which kind of packets it occurs? Which problems you could expect with SIP in NAT scenarios - why?

Question 4
You'll find a VoIP sniffer installed in your VMware image called `ucsniffer` (use the man page for further help: *man ucsniff*). Start ucsniffer in an empty directory and take a look at the generated files. Work together in teams of three and try to establish a voice over IP connection. One of you should use the ucsniffer tool to spy on this connection. Play a bit around with this sniffing tool and take a look at the arp table and use wireshark to analyse the traffic. If everything works fine there should be a pcap file generated, that you can open and analyse with wireshark. Can you imagine a solution to avoid an attacker listening to your call?
Imagine two people within the Freiburg campus network using a public SIP service. What would you expect for your in-call voice traffic? Why wouldn't you see that kind of packet exchange in real (why are the packets whould be routed further on over the provider)?

23.2 Lösungsskizze

Question 1
Die Programme Zoiper und Edika können intuitiv konfiguriert werden. Es erfolgt eine Direktverbindung zwischen zwei Personen mittels der IP Adresse. Eine Analyse des entstehenden Traffics erfolgt mit Hilfe von Wireshark. Hierbei wird der genaue Verbindungsaufbau durch das SIP Protokoll verdeutlicht.

Question 2
Mittels eines zentralen Servers als SIP Proxy kann eine Verbindung zu dem zentralen Asterisk Server der Universität aufgebaut werden. Durch das Präfix „10“ vor einer Rufnummer ist es nach erfolgreichem Verbindungsaufbau möglich eine beliebige deutsche Festnetznummer anzurufen.

Question 3
Als Sprachcodec kann G.711a verwendet werden. Die entsprechende Bandbreite beträgt 64 kbit/s. GSM Codecs haben eine wesentlich geringere Datenrate. In einem Netzwerk mit NAT kommt es zu dem Problem, dass Signalisierung und Sprachübertragung getrennt erfolgen. Für die Audioübertragung wird ein beliebiger zufälliger Port gewählt. So kann es passieren, dass trotz erfolgreichem Gesprächsaufbau ein Gesprächspartner nichts hören kann, da die Audioübertragung auf einem anderen Port fehlschlägt.

Question 4
Der uc-sniffer kann mit folgendem Befehl im Monitor Mode gestartet werden: *ucsniff -i eth0 -M*. Alternativ im Man in the middle Mode (Mitm) mit folgendem Befehl: *ucsniff -i eth0 -c 1 // //*
Würde anstelle von RTP ein verschlüsseltes Protokoll zur Sprachübertragung verwendet werden, wäre es egal, wenn das Gespräch mitgeschnitten wird.

24 Asterisk

Asterisk ist eine freie Software, die alle Funktionalitäten einer herkömmlichen Telefonanlage abdeckt. Ursprünglich war Asterisk als Telefonanlage für analoge Telefonanschlüsse konzipiert worden. Später kam dann ISDN hinzu, gefolgt von Voice over IP mit unterschiedlichen Protokollen.
Zunächst einmal soll von den Studenten ein eigener Asterisk Server als Telefonanlage eingerichtet und untereinander Telefongespräche mittels einer Voice over IP Software geführt werden. Anschließend soll dieser Server als Client an einen zentralen Asterisk Server angebunden werden, um damit Gespräche ins Festnetz führen zu können. Abschließend sollen mehrere Asterisk Server als Kaskade miteinander verbunden werden und IAX als alternatives Verbindungsprotokoll genutzt werden.

24.1 Aufgaben

Asterisk is a software implementation of a telephone private branch exchange (PBX) originally created in 1999 by Mark Spencer of Digium. Like any PBX, it allows attached telephones to make calls to one another, and to connect to other telephone services including the public switched telephone network (PSTN) and Voice over Internet Protocol (VoIP) services. Its name comes from the asterisk symbol, „*“.

Everything you'll need to solve these exercises you'll find on *www.das-asterisk-buch.de* in chapter 2/3.

Question 1
Work together in teams of 2. Try to configure an Asterisk server on one of your machines. Asterisk is allready preinstalled in your experiments environment. You might work in groups here: One setting up the Asterisk and the others try to connect via SIP. Edit the configuration file */etc/asterisk/sip.conf*. Try to create 2 user accounts, one for both of you. After that start a softphone software (Zoiper, Edika, Ekiga, Twinkle, etc.) and establish with your just created account a connection to your Asterisk server. To start Asterisk in the debug mode use the follwing command: *asterisk -vvvvr*. If you make some changes in the configuration files you can reload the configuartion with: *module reload*. Alternative you can stop the Asterisk server: *stop now* and start the Asterisk server: asterisk -vvvvvc. After that try to establish a call and check if everything works fine. For testing it might be helpful to setup 2 simple services ("echo" and "demo-thanks" in your */etc/asterisk/extensions.conf*), which could be called with the numbers given:

```
exten => 111,1,Answer()
exten => 111,2,Wait(1)
exten => 111,3,Echo()

exten => 212,1,Answer()
exten => 212,2,Playback(demo-thanks)
exten => 212,3,Hangup
```

Question 2
You should configure your Asterisk as client to the main Asterisk Server (132.230.4.8). Are there any differences if connecting an Asterisk server or a SIP client? You might logon to this server using the credentials comsysNN, cseNN (username and password, `NN` equals to `01` to `10`). To login successful you'll need the Comsys Server (132.230.4.1) as outbound proxy. Edit the configuration files */etc/asterisk/sip.conf* to establish a connection to the Asterisk Server (132.230.4.8).
Prove if everything works correct and your Asterisk server is connected as client to the Comsys server. Take a look in the Asterisk console or wireshark.

Question 3
Create a dialing rule with pattern matching to make external calls. The prefix 10 will be necessary, because the main Asterisk server has a dialing rule, which only accepts external calls if this prefix is used. Use your softphone to establish an external call. Try to call the following external number: 10 0761 2034631 - telephone in room -111 (or any other land-line number of yours). Try to configure your Asterix server, so that you can dial an external number without this "10"- Prefix.

Question 4
Extend the task in question 2 and setup a cascade of Asterisk servers (work with your neighbors: Connect your machine to your neighbors and this one to the central server). Would it be possible to link your servers in both directions (every partner logging on to the other machine)? Connect two SIP clients to different ends of your Asterisk line and call each other. Observe the SIP messages exchanged for call setup and the RTP packages exchanged. Are they taking the same route (why/not)?

Question 5 (expert level)
The way to setup connections between Asterisk servers as required in question 2 and 4 is not very practical for many connections. Look up the concept of IAX (Inter Asterisk Exchange protocol) and try to connect two of your servers this way. Work in pairs for this task. What is the major difference of IAX compared to SIP?

Hint: If you want to use Asterisk at home or in a company take a look at *www.freepbx.org* for a configuration GUI called FreePBX.

24.2 Lösungsskizze

Question 1,2,3
Um diese Aufgaben lösen zu können sind Änderungen in den zwei Konfigurationsdateien *sip.conf* und *extension.conf* notwendig. Es werden zunächst zwei User Accounts: 2000 und 2001 angelegt. Mittels des register Befehls kann der Asterisk Server als Client zu dem zentralen Asterisk Server verbunden werden. Die entsprechenden Wahlregeln können in der *extension.conf* konfiguriert werden. Hier ein Auszug aus den zwei benötigten Konfigurationsdateien *sip.conf* und *extension.conf*:

```
;sip.conf
[general]
port=5060
bidaddr=0.0.0.0
language=de

;register => comsys03:cse03@132.230.4.8/comsys03
register => comsys03@132.230.4.8:cse03:comsys03@10.230.4.1

[2000]
type=friend
secret=1234
host=dynamic
context=meine-telefone

[2001]
type=friend
secret=4321
host=dynamic
context=meine-telefone

[ext-sip-account]
type=friend
```

```
context=von-voip-provider
username=comsys03
fromuser=comsys03
secret=cse03
host=132.230.4.8
;fromdomain=132.230.4.8
outboundproxy=10.230.4.1
qualify=yes
insecure=very
nat=yes

;extension.conf
[default]
exten => 1001,1,Answer()
exten => 1001,2,Playback(hello-world)
exten => 1001,3,Hangup()

[meine-telefone]
include => default
exten => 50,1,Dial(SIP/2000)
exten => 51,1,Dial(SIP/2001)
exten => 51,2,VoiceMail(2001,u)
exten => 2999,1,VoiceMailMain(${CALLERID(num)),s)
exten => _10.,1,Dial(SIP/${EXTEN}@ext-sip-account)
exten => _0.,1,Dial(SIP/10${EXTEN}@ext-sip-account)
```

Question 5
Bei Sip und IAX handelt es sich um zwei unterschiedliche Übertragungsprotokolle. Ein besonderer Vorteil von IAX ist, dass es nur einen Port braucht, was für Verbindung über Firewalls sehr von Vorteil ist. IAX arbeitet während des Gesprächs unabhängig von der Anzahl der Anrufe und des verwendeten Codecs effizienter als RTP. Außerdem ist IAX nicht ASCII-, sondern Datenelementkodiert. Dies macht Implementierungen wesentlich leichter und zudem robuster gegenüber Pufferüberlaufangriffen.

25 Beschreibung einiger nützlicher Tools

- **arp**
  arp -n
  arp -d <gateway-ip>
- **avahi**
  Avahi is a system which facilitates service discovery on a local network. This means that you can plug your laptop or computer into a network and instantly be able to view other people who you can chat with, find printers to print to or find files being shared. This kind of technology is already found in Apple MacOS X (branded Rendezvous, Bonjour and sometimes Zeroconf) and is very convenient. Avahi is mainly based on Lennart Poettering's flexmdns mDNS implementation for Linux which has been discontinued in favour of Avahi.
- **brctl**
  Tool to create, modify and shutdown software bridges in Linux. Check *man brctl* for further help and options!
- **dhcpd**
  see configfile: */etc/dhcp3/dhcpd.conf*
- **dig**
  Domain information groper is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried
- **ettercap**
  Ettercap is a suite for man in the middle attacks on LAN. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols (even ciphered ones) and includes many features for network and host analysis.

  Textmode:
  ettercap -Tq -M arp:remote //
  T = Text mode
  q = quiet - without this you'll see the raw packets which can be noisy and annoying.
  M = Man-In-The-Middle
  arp: = ARP Poisoning
  remote = forward packets destined for the WAN (Internet, whatever)
  // = Empty target specification; i.e. all hosts

  Curses mode:
  This is the one with the cursed interface.
  Use: ettercap -C
  and then select the following options:
  Sniff -> Unified Sniffing
  eth0 (or whatever you i/f spec is)
  Hosts -> Scan for hosts
  Start -> Start Sniffing
  Mitm -> Arp poisoning
  remote
  View -> Connections Mitm -> Stop Mitm
  Start -> Stop Sniffing
  Start -> Exit to quit.

  GTK+ mode:

This is the one with the GTK+ interface.
Use: ettercap -G
Then follow the same steps as for the Cursed interface.

- **hosts**
  The static table lookup for host names.
- **ifconfig** *<interface> [options]*
  ifconfig serves to configure and control TCP/IP network interfaces. If no arguments are given, ifconfig displays the status of the currently active interfaces. Otherwise you can for example configure the (MAC, IP) address, the MTU size etc. of a given interface.
  Examples:
  ifconfig -a (Shows all Interfaces)
  ifconfig eth0 (Displays the status of the interface eth0)
- **iptables**
  For references please have a look at theman page or web page put online at:
  http://www.ks.uni-freiburg.de/download/inetworkSS05/practical/references/pra-ref06.html
- **nslookup**
  Sends queries to Internet domain name servers. It has two modes: interactive and non-interactive. Interactive mode allows the user to contact servers for information about various hosts and domains or to display a list of hosts in a domain. Non-interactive mode is used to display just the name and requested information for a host or domain.
- **openvpn**
  OpenVPN is a full-featured open source SSL VPN solution that accommodates a wide range of configurations, including remote access, site-to-site VPNs, Wi-Fi security, and enterprise-scale remote access solutions with load balancing, failover, and fine-grained access-controls.
  (see: http://openvpn.net)
- **ping** *<destination>*
  Example: *ping 192.168.1.1*
- **ping6** *<IPv6 destination>*
  read more about the paramters in the manpage
- **pppoe-server**
  see configfile: */etc/ppp/pppoe-server-options*
- **route**
  Route manipulates the kernel's IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig program. When the add or del options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables.
- **ssh**
  ssh (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel. ssh connects and logs into the specified hostname (with optional user name). The user must prove his/her identity to the remote machine using one of several methods depending on the protocol version used.
- **traceroute** *<destination>*
  Traceroute is a computer network tool used to determine the route taken by packets across an IP network. Traceroute is often used for network troubleshooting. By showing a list of routers traversed, it allows the user to identify the path taken to reach a particular destination on the network. This can help identify routing problems or firewalls that may be blocking access to a site.
- **traceroute6** *<IPv6 destination>*
  read more about the paramters in the manpage

- **TSPC**
  Tunnel Server Protocol Client (tspc), is a daemon to automate the setup and maintenance of an IPv6 tunnel. This client will connect to any migration broker which uses Hexago's implementation.
- **ucsniffer**
  UCSniff is a VoIP/UC Sniffer / Assessment / Pentest tool with some useful new features, such as IP Video Sniffing. UCSniff is a Proof of Concept tool to demonstrate the risk of unauthorized VoIP eavesdropping on voice and video - it can help you understand who can eavesdrop, and from what parts of your network. It is intended for next generation enterprise VoIP/UC Infrastructures that rely on Voice VLANs to segment UC applications for QoS requirements.
- **vconfig** *<interface> [options]*
  vfconfig serves to configure VLANs on Ethernet interfaces
- **wireshark** *[interface]*
  Wireshark is an Open Source packet sniffer computer application. It is used for network troubleshooting, analysis, software and communications protocol development. It allows the user to see all traffic being passed over the network (usually an Ethernet network) by putting the network interface into promiscuous mode. Promiscuous mode is a configuration of a network card that makes the card pass all traffic it receives to the CPU rather than just packets addressed to it. Each packet includes the hardware (Media Access Control) address. When a network card receives a packet, it checks if the address is its own. If not, the card normally drops the packet. But in promiscuous mode, the card doesn't drop the packet, thus allowing the computer to read all packets.
  Example: *wireshark eth0* (starts wireshark to listen on interface eth0)