



**Leopold-Franzens-Universität  
Innsbruck**

**Institut für Informatik**

*Analyse des Netzwerkverhaltens  
von Echtzeit-Multimedia-Internetanwendungen  
detaillierte Analyse von VoIP*

Bakkalaureatsarbeit

eingereicht bei Dr. Ing. Michael Welzl

Autor: Thomas Rammer

Innsbruck, 07.07.2004

# 1 Inhaltsverzeichnis

1	Inhaltsverzeichnis.....	2
2	Allgemeines.....	4
2.1	Einleitung .....	4
2.2	Aufgabenstellung .....	4
3	Überblick über VoIP .....	5
3.1	Herkömmliche Telephonie.....	5
3.2	Voice over Internet Protokoll.....	5
	Sprachqualität.....	6
3.2.1	Delay (Verzögerung).....	6
3.2.2	Packet Loss (Paketverluste) .....	6
3.2.3	Compression (Kompression).....	6
3.2.4	Jitter (Laufzeitschwankung).....	6
3.3	wichtige Netzwerkprotokolle .....	7
3.3.1	TCP.....	7
3.3.2	IP .....	7
3.3.3	UDP.....	7
3.3.4	H.323 (Protokollfamilie) .....	7
3.3.5	SIP .....	8
3.3.6	SDP.....	8
3.3.7	RTP.....	8
4	Versuchsdurchführung .....	9
4.1	Allgemeines zur Teststrecke .....	9
4.2	Aufbau der Teststrecke.....	9
4.3	Spezifikation der Teststrecke .....	10
4.3.1	Hardwarespezifikation .....	10
4.3.2	Softwarespezifikation.....	10
4.4	Tools für die Durchführung der Tests .....	11
4.4.1	Netzwerkspezifische Tools .....	11
4.4.2	weitere Tools .....	12
4.5	Testszenarien.....	13
4.6	Skripte .....	13
4.6.1	tc – Skripte .....	13
4.6.2	mgen – Skripte .....	14
4.6.3	Perl – Skripte.....	15
4.6.4	Gnuplot – Skripte .....	15
4.7	Testfile.....	16
4.8	Ablauf des Testverfahrens.....	17
5	Testergebnisse .....	19
5.1	ICQ.....	19
5.1.1	10sec Stille übertragen – kein Hintergrundverkehr.....	19
5.1.2	Kein Hintergrundverkehr .....	19
5.1.3	Wenig Hintergrundverkehr .....	19
5.1.4	Mittlerer Hintergrundverkehr .....	19
5.1.5	Viel Hintergrundverkehr .....	20
5.1.6	Ruckartig auftretender Hintergrundverkehr .....	20
5.2	MSN.....	20
5.2.1	10sec Stille übertragen – kein Hintergrundverkehr.....	20
5.2.2	Kein Hintergrundverkehr .....	20

5.2.3	Wenig Hintergrundverkehr .....	21
5.2.4	Mittlerer Hintergrundverkehr .....	21
5.2.5	Viel Hintergrundverkehr .....	21
5.2.6	Ruckartig auftretender Hintergrundverkehr .....	21
5.3	Roger Wilco .....	21
5.3.1	10sec Stille übertragen – kein Hintergrundverkehr.....	21
5.3.2	Kein Hintergrundverkehr .....	22
5.3.3	Wenig Hintergrundverkehr .....	22
5.3.4	Mittlerer Hintergrundverkehr .....	22
5.3.5	Viel Hintergrundverkehr .....	22
5.3.6	Ruckartig auftretender Hintergrundverkehr .....	22
5.4	Skype .....	23
5.4.1	10sec Stille übertragen – kein Hintergrundverkehr.....	23
5.4.2	Kein Hintergrundverkehr .....	23
5.4.3	Wenig Hintergrundverkehr .....	23
5.4.4	Mittlerer Hintergrundverkehr .....	23
5.4.5	Viel Hintergrundverkehr .....	23
5.4.6	Ruckartig auftretender Hintergrundverkehr .....	24
5.5	Zusammenfassung der gewonnenen Ergebnisse .....	25
5.5.1	Zusammenfassung ICQ .....	25
5.5.2	Zusammenfassung MSN .....	25
5.5.3	Zusammenfassung Roger Wilco .....	25
5.5.4	Zusammenfassung Skype .....	25
5.6	Datentransfer Tabelle .....	26
6	Schlusswort .....	27
6.1	Zusammenfassung der Arbeit.....	27
6.2	Wissenswertes .....	27
6.3	Danksagung .....	27
6.4	Literaturverzeichnis.....	28

## 2 Allgemeines

### 2.1 Einleitung

Die Erfindung des Grundprinzips des Telefons, Sprache auf elektrischem Wege zu übertragen, geht auf den Deutschen Philipp Reis im Jahre 1860 zurück. Verbessert und patentiert wurde diese Technik der Sprachübertragung von dem Engländer Alexander Graham Bell im Jahre 1876, welcher bis heute im amerikanischen und angelsächsischen Raum als Erfinder dieser Technologie gehandelt wird.

Neuere Ansätze von Multimediaanwendungen zielen auf eine Sprachübertragung über das Medium Internet ab, welche unter dem Kürzel VoIP (Voice over IP tools) bekannt sind. Dabei stellt sich im Moment eine Reihe von Fragen zu den Massen an Anwenderprogrammen die zum Zweck der Internettelephonie zur Verfügung stehen.

All diese Fragen zielen auf eine Kernfrage ab! Die Frage nach der Dienstgüte (Quality of Service – kurz QoS) der jeweiligen Anwendung.

### 2.2 Aufgabenstellung

Folgende Punkte wurden mit dieser Arbeit abgedeckt:

- Verständnis der Netzwerkgrundkenntnisse
- Verständnis der Grundprinzipien von Voice over IP Anwendungen
- Installation der VoIP Anwendersoftware auf den zur Verfügung gestellten Rechnern
- Analyse des auftretenden Netzwerkverkehrs
- Evaluierung der gewonnen Testergebnisse
- Auswertung und graphische Darstellung der Testergebnisse
- Interpretation der Testergebnisse

Folgende Fragen wurden mit dieser Arbeit beantwortet:

- Wie viel Bandbreite wird für eine VoIP Übertragung aufgewendet?
- Wird UDP oder TCP vom Anwenderprogramm benutzt?
- Reagiert ein eventueller UDP – Strom auf Bandbreitenschwankungen?
- Ist mit einem permanenten oder einem sporadisch unterbrochenen Verkehrsstrom zu rechnen?
- Werden große oder kleine Pakete zur Übermittlung der Nachrichten benutzt?

## 3 Überblick über VoIP

### 3.1 Herkömmliche Telephonie

Die althergebrachte Telephonie arbeitet verbindungsorientiert. Das bedeutet, dass die direkte Verbindung der beiden Endgeräte, welche nach dem Initialisieren besteht, bis zum Ende des Gespräches aufrechterhalten bleibt. Wie viele Daten übertragen werden ist dabei egal, es bleibt immer die gleiche Verbindungsstärke erhalten. Folglich ändert sich an der Verbindung nichts, egal ob man nun am Telefon singt, redet oder nur stumm den Hörer in der Hand hält. Der zu Beginn des Gespräches initialisierte Sprachkanal bleibt während der gesamten Laufzeit blockiert.

Der große Vorteil der herkömmlichen Telephonie gegenüber neueren Techniken ist, dass eine sehr gute Sprachqualität durch eine ständig konstante Bandbreite der Verbindung gewährleistet wird.

### 3.2 Voice over Internet Protokoll

Bei dieser Technologie [2] wird das analoge Sprachsignal des Menschen mittels eines AD – Wandlers (Kürzel für Analog – / Digitalwandler) in einen kontinuierlichen Datenstrom umgewandelt. Nach dieser Umwandlung wird das digitale Signal komprimiert, damit möglichst wenig Bandbreite beansprucht werden muss – was dazu führen sollte, dass die Verbindungsqualität erhöht wird (kann aber bei schlechter Realisierung auch genau das Gegenteil bewirken).

Die komprimierten Daten werden dann wie üblicher Netzwerkverkehr behandelt, sprich zuerst in einzelne Datenpakete zerlegt und anschließend verschickt. Nach dem Einlangen beim Empfänger gelangen die Pakete in den so genannten De – Jitter – Buffer, in welchem Laufzeitunterschiede ausgeglichen werden sollen. Der Buffer setzt dann die einzelnen Pakete wieder zusammen, woraufhin die Daten dekodiert und anschließend wieder von digital in analog umgewandelt werden (siehe Abbildung 1: Grundprinzip von VoIP).

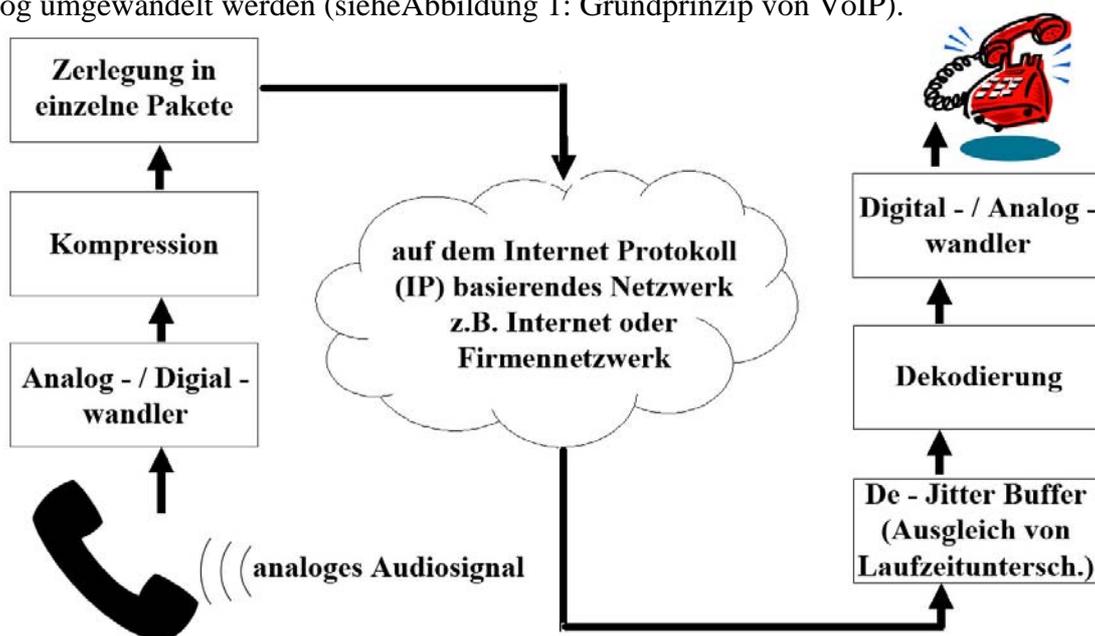


Abbildung 1: Grundprinzip von VoIP

## **Sprachqualität**

Die Sprachqualität von VoIP ist im Wesentlichen von den folgenden 4 Faktoren abhängig:

### **3.2.1 Delay (Verzögerung)**

Die bei einem herkömmlichen Telefongespräch anfallende Verzögerungszeit bewegt sich zwischen 25 und 50ms. Diese Zeitspanne ist so kurz, dass sie vom menschlichen Gehör nicht wahrgenommen wird. Da bei einem herkömmlichen Gespräch immer die gleiche Bandbreite zur Verfügung steht, ändert sich an diesem Verzögerungswert nichts. Anders bei VoIP, meist schwankt hier die verfügbare Bandbreite, weil sie von anderen Anwendungen benötigt wird. Dies birgt zwar zum einen den Vorteil der Kostenersparnis in sich, zum anderen kann es aber unter Umständen erhebliche Verzögerungen in der Übertragung bewirken.

### **3.2.2 Packet Loss (Paketverluste)**

Vor kurzem fand man heraus, dass das Gehirn nur den Anfang und das Ende eines geschriebenen Wortes benötigt, um es genau zu identifizieren. So kann aus einem totalen Wirrwarr an Information immer noch ein sinnvoller Zusammenhang erkannt werden. Ebenso wie mit der Schrift verhält es sich mit der Sprache. Geringe Paketverluste werden vom Gehirn ohne weitere Probleme ausgeglichen. Steigt die Verlustrate aber an, so kann auch unser Gehirn keine Wunder mehr bewirken und man versteht nur noch Kauderwelsch.

### **3.2.3 Compression (Kompression)**

Wie zuvor schon angeschnitten verwendet man Komprimierverfahren um die anfallende Datenmenge zu schmälern und damit Bandbreite zu sparen. Im Allgemeinen scheint dieser Lösungsansatz logisch und unumgänglich zu sein. Allerdings kann dies auch negative Auswirkungen auf die Sprachqualität zur Folge haben. Zum einen ist Komprimierung nicht immer verlustfrei, zum anderen benötigt man je nach Komprimierungsgrad unter Umständen relativ lange um die Daten zu komprimieren und nach erfolgter Übertragung wieder zu dekomprimieren.

### **3.2.4 Jitter (Laufzeitschwankung)**

Der Jitter – Buffer beim Empfänger dient dazu, dem Empfänger einen kontinuierlich gleich bleibenden Datenstrom zu vermitteln. Da der Datenstrom in einem paketorientierten Netzwerk diese Eigenschaften meist nicht aufweist, müssen die übermittelten Daten zwischengespeichert werden und werden erst nach einer kurzen Verzögerungszeit, in welcher später eintreffende Pakete noch in die richtige Position vorgereiht werden, ausgegeben. Dieses ungleichmäßige Einlangen von Paketen entsteht einerseits durch Paketverluste (neuerlicher Versand) sowie durch unterschiedlich lange Wege, welche von den Paketen im Netzwerk zurückgelegt werden müssen.

### **3.3 wichtige Netzwerkprotokolle**

Die für diese Arbeit wichtigsten Netzwerkprotokolle [4] werden im Folgenden in aller Kürze besprochen:

#### **3.3.1 TCP**

Das Transmission Control Protocol ist ein zuverlässiges, verbindungsorientiertes Transportprotokoll in Computernetzwerken. Es ist Teil der TCP/IP – Protokollfamilie.

TCP stellt einen virtuellen Kanal zwischen zwei Rechnern (genauer: Endpunkten zwischen 2 Anwendungen auf diesen Rechnern) her. Auf diesem Kanal können in beide Richtungen Daten übertragen werden. TCP setzt in den meisten Fällen auf das IP – Protokoll auf. Es ist in Schicht 4 des OSI – Netzwerkschichtenmodells angesiedelt.

#### **3.3.2 IP**

Das Internet Protocol ist ein in Computernetzen weit verbreitetes Netzwerkprotokoll. Es ist eine Implementation der Internet-Schicht des TCP/IP – Modells bzw. der Netzwerk-Schicht des OSI – Modells.

IP bildet die erste vom Übertragungsmedium unabhängige Schicht der Internet-Protokoll-Familie. Im Gegensatz zu der physikalischen Adressierung der darunter liegenden Schicht, bietet IP logische Adressierung. Das bedeutet, dass mittels IP – Adresse und Subnetzmaske (subnet mask) Computer innerhalb eines Netzwerkes in logische Einheiten, so genannte Subnetze, gruppiert werden können. Auf dieser Basis ist es möglich, Computer in größeren Netzwerken zu adressieren und Verbindungen zu ihnen aufzubauen, da logische Adressierung die Grundlage für Routing ist. IP stellt also die Grundlage des Internets dar.

#### **3.3.3 UDP**

Das User Datagram Protocol ist ein minimales, verbindungsloses Netzwerkprotokoll. Es gehört zur Transportschicht der TCP/IP – Protokollfamilie und ist im Gegensatz zu TCP nicht auf Zuverlässigkeit ausgelegt.

#### **3.3.4 H.323 (Protokollfamilie)**

H.323 ist eine übergeordnete Empfehlung der ITU-T (International Telecommunication Union), in der Protokolle definiert werden, die eine audio-visuelle Kommunikation auf jedem Netzwerk, das Pakete überträgt, ermöglichen. Es ist zurzeit in verschiedenen Anwendungen wie z.B. NetMeeting und OpenH323 (für Linux) implementiert.

### **3.3.5 SIP**

Das Session Initiation Protocol ist ein Netzwerkprotokoll zum Aufbau einer Kommunikationssitzung zwischen Benutzern. Das Protokoll wird in RFC 3261 spezifiziert und wird bei der IP – Telefonie verwendet.

Im Gegensatz zu H.323 wurde SIP mit Blick auf das Internet von der IETF entwickelt und orientiert sich an der Architektur gängiger Internet-Anwendungen. Dabei wurde von Beginn an auf leichte Implementierbarkeit, Skalierbarkeit, Erweiterbarkeit und Flexibilität geachtet. Benutzt werden kann es, um beliebige Sessions mit einem oder mehreren Teilnehmern zu verwalten. Dabei ist es nicht auf Internet – Telefonie beschränkt, sondern Sessions können beliebige Multimediaströme, Konferenzen, Computerspiele, usw. sein.

### **3.3.6 SDP**

Das Session Description Protocol (RFC 2327) dient dazu Kommunikationssitzungen zu verwalten und wird beispielsweise zusammen mit SIP bei der IP – Telefonie eingesetzt. SDP dient hier dazu, die zwischen den Endpunkten zu verwendenden Codecs, Transportprotokolle, usw. auszuhandeln.

### **3.3.7 RTP**

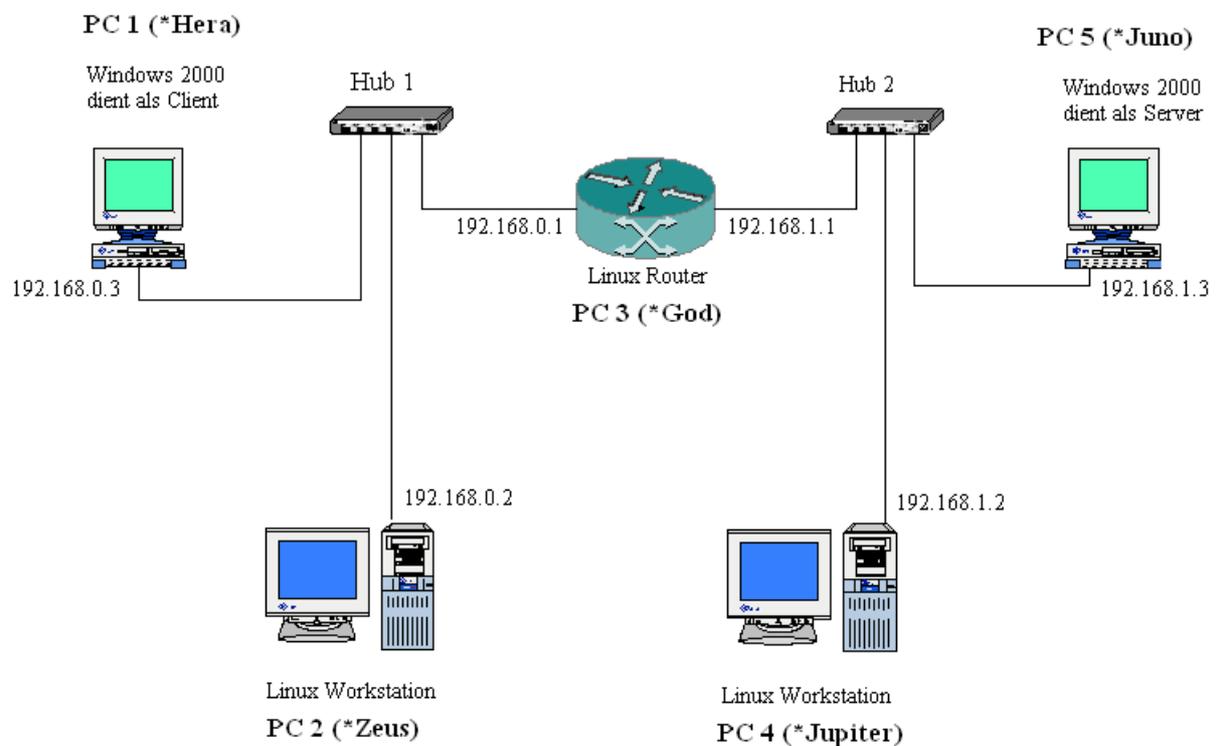
Das Realtime Transport Protocol (RFC 1889) dient dazu, Multimedia – Datenströme (Audio, Video, Text, etc.) über Netzwerke zu transportieren, d.h. die Daten zu kodieren, zu paketieren und zu versenden. RTP ist ein Paket-basiertes Protokoll und wird normalerweise über UDP betrieben. Es findet Anwendung in vielen Bereichen, unter anderem wird es bei den IP – Telefonie – Technologien H.323 und SIP dazu verwendet bei die Audio-/Videostreams des Gespräches zu übertragen.

## 4 Versuchsdurchführung

### 4.1 Allgemeines zur Teststrecke

Die Teststrecke wurde vom Institut für Informatik zur Verfügung gestellt und ermöglichte eine genau Messung und Auswertung der zuvor festgelegten Testszenarien, welche in diesem Kapitel später noch genauer beschrieben werden. Vorab, allen Szenarien ist gemein, dass ein VoIP Tool auf zwei Rechnern innerhalb der Teststrecke gestartet wird und während der Laufzeit der anfallende Netzwerkverkehr zwischen diesen beiden PCs gemessen wird. Dabei wird ein zuvor genau festgelegtes Protokoll durchlaufen.

### 4.2 Aufbau der Teststrecke



\* Namen der Computer in der verwendeten Teststrecke

Abbildung 2: Teststreckenaufbau

Der Aufbau sowie der störungsfreie Betrieb wurden durch die Mitarbeiter des Institutes für Informatik gewährleistet. An den Grundspezifikationen wurde nichts geändert, da sie sich für diesen Versuchszweck als gut geeignet erwiesen. Der Aufbau an sich stellte sich wie in Abbildung 2: Teststreckenaufbau abgebildet dar [3].

## **4.3 Spezifikation der Teststrecke**

### **4.3.1 Hardwarespezifikation**

Die Hardwarespezifikation ist für alle 5 Rechner der Versuchsstrecke dieselbe wie unten angeführt:

Prozessor: AMD Athlon XP 2200+  
Mainboard: VIA Apollo VT8366/A  
Graphikkarte: GeForce Ti 4400  
Soundkarte: VIA-AC'97  
Netzwerkkarte: Surecom EP320-R-100/10/M-PCI (100 MBit)  
RAM: 512 MB DDR RAM

### **4.3.2 Softwarespezifikation**

Die Softwarespezifikation stellte sich folgenderweise dar:

#### **PC 1:**

Betriebssystem: Windows 2000  
Verwendungszweck: Rechner auf dem die zu testende Applikation läuft

#### **PC 2:**

Betriebssystem: Red Hat Linux 8.0 Kernel 2.4.18-24  
Verwendungszweck: Aufzeichnen des Netzwerkverkehrs zwischen PC 1 und PC 3.

#### **PC 3:**

Betriebssystem: Red Hat Linux 8.0 Kernel 2.4.18-24  
Verwendungszweck: Router

#### **PC 4:**

Betriebssystem: Red Hat Linux 8.0 Kernel 2.4.18-24  
Verwendungszweck: Erzeugung von Hintergrundverkehr von PC 4 zu PC 2 sowie „pingen“ auf PC 1 (für Staukontrolle)

#### **PC 5:**

Betriebssystem: Windows 2000  
Verwendungszweck: Rechner auf dem die zu testende Applikation läuft

## 4.4 Tools für die Durchführung der Tests

### 4.4.1 Netzwerkspezifische Tools

Name: tcpdump  
Verwendungszweck: Protokollierung des Netzwerkverkehrs  
Firma: Open Source  
Version: 3.8  
Quelle: <http://www.tcpdump.org/>  
Beschreibung: Tcpdump ist ein Open Source Netzwerksniffer der standardmäßig in fast allen Linux Distributionen mitgeliefert wird. Ein Sniffer ist im Prinzip ein Netzwerk-Monitor oder Netzwerk- Diagnoseprogramm, welches es ermöglicht den anfallenden Netzwerkverkehr an einem fix bestimmten Punkt paketgenau zu protokollieren. Ein andere sehr bekannter Sniffer wäre zum Beispiel Ethereal, welches aufgrund seiner graphischen Oberfläche sehr handlich zu bedienen ist.

Name: ping  
Verwendungszweck: um auftretende Verzögerungen zu beobachten  
Firma: Betriebssystemtool  
Version: -  
Quelle: auf jedem Rechner  
Beschreibung: Der Packet InterNet Groper oder kurz ping ist ein in beinahe allen Betriebssystemen gängiger Befehl. Er führt dazu, dass ein bzw. mehrere ICMP Echo – Request – Paket(e) vom localhost an den angegebenen Zielhost gesendet werden. Der Empfänger dieser Nachricht(en) muss laut Protokollspezifikation eine Antwort zurückschicken: ICMP Echo – Reply. Das Programm führt eine Statistik über die empfangenen bzw. verloren gegangenen Pakete und zeigt die round trip time der einzelnen Pakete an, jene Zeit die das Paket vom localhost zum Zielhost und zurück braucht.

Name: mgen  
Verwendungszweck: Erzeugen von zuvor definiertem Hintergrundverkehr  
Firma: Naval Research Laboratory  
Version: 4.1  
Quelle: <http://mgen.pf.itd.nrl.navy.mil/mgen.html>  
Beschreibung: Mgen ist ein so genannter Traffic – Generator. Das Programm ist Open Source und erzeugt einen definierbaren UDP Netzwerkverkehr zu einem frei wählbaren Zielrechner im Netzwerk. Praktisch an diesem Tool ist, dass es Skripte verarbeiten kann und damit ein exakter, immer wieder reproduzierbarer Ablauf der Testversuche gewährleistet werden kann.

Name: tc  
Verwendungszweck: Beschränkung der Bandbreite  
Firma: Linux eigenes Programm  
Version: -.  
Quelle: <http://lartc.org/lartc.html>  
Beschreibung: Das Traffic Control Programm [1] dient zum Management von Queuing Disciplines, Verkehrsklassen und Trafficfiltern. In dieser Arbeit wurde es eingesetzt um die vorhandene Bandbreite zu reduzieren und eine ADSL Leitung mit einer Bandbreite von 512KB zu simulieren. Mit anderen Worten, es wurde eine Bottleneck erzeugt, durch welchen sowohl der von mgen erzeugte UDP Datenstrom als auch die eigentlichen VoIP Daten der Anwendungsprogramme geschickt wurden.

#### 4.4.2 weitere Tools

Andere verwendete Tools dienten hauptsächlich der graphischen Auswertung und Darstellung der Versuche. Da diese Tools für diese Arbeit nur eine nebensächliche Rolle spielen, werden sie nur in aller Kürze aufgelistet:

Name: VirtualDub  
Verwendungszweck: extrahieren des Tons des in anderen Arbeiten verwendeten Matrix Trailers und speichern als .wav – Datei.

Name: Microsoft Audiorecorder v 5.1  
Verwendungszweck: 10sec Stille generieren und in wav – Testdatei einfügen.

Name: Gnuplot  
Verwendungszweck: graphische Darstellung der Testergebnisse

Name: Active Perl  
Verwendungszweck: Ausführung der Perlskripte

Name: Microsoft Word  
Verwendungszweck: Dokumentation dieser Arbeit

Name: Adobe Acrobat Writer  
Verwendungszweck: Dokumentation in plattformunabhängiges Format konvertieren

## 4.5 Testszzenarien

Die Testszzenarien, welche im Vorfeld der Versuche festgelegt wurden, können grob in zwei Gruppen unterteilt werden. Zum einen jene Szenarien, welche sich durch die Menge des auftretenden Hintergrundverkehrs unterscheiden. Zum anderen in Szenarien, die sich durch die Eingangsdaten die das VoIP Tool zu senden hat unterscheiden.

Zum ersten Punkt, welcher genauer in dieser Arbeit behandelt wird, gehören folgende Einzelszenarien:

1. Kein Hintergrundverkehr
2. Wenig Hintergrundverkehr
3. Mittlerer Hintergrundverkehr
4. Viel Hintergrundverkehr
5. Ruckartig auftretender Hintergrundverkehr

Beim zweiten Punkt unterscheiden sich beim Test die Eingangsdaten des VoIP Tools, sprich im ersten Fall wurde eine Audiosequenz mit möglichst wenigen bis gar keinen Pausen übertragen. Hingegen im zweiten Fall wurde eine längere Pause übermittelt, um zu sehen, ob diese das Verhalten des VoIP Tools beeinflusst. Dieser Testfall (mit Pause) wird in dieser Arbeit als „10sec Stille übertragen – kein Hintergrundverkehr“ bezeichnet.

## 4.6 Skripte

Bei dieser Arbeit wurden Skripte [3] für folgende Teilaufgaben verwendet:

- tc – Skripte: um die Bandbreite zu beschränken
- mgen – Skripte: um einen zuvor genau definierten Hintergrundverkehr zu erzeugen
- Perl – Skripte: um die erhaltenen Messergebnisse auszuwerten
- Gnuplot – Skripte: um die aufbereiteten Messergebnisse graphisch darzustellen

### 4.6.1 tc – Skripte

```
tc qdisc add dev eth1 root handle 1: cbq avpkt 1000 bandwidth 100mbit
tc class add dev eth1 parent 1: classid 1:1 cbq rate 512kbit allot 1500
prio 5 bounded isolated
tc filter add dev eth1 parent 1: protocol ip prio 16 u32 match u32 0 0 at 0
flowid 1:1
```

*tc-set.sh*

Als erstes wird für device eth1 (zu behandelnde Netzwerkkarte) eine Queuing Strategie festgelegt, in diesem Fall cbq (Class Based Queueing), mit **tc qdisc**.

Dann kann mit **tc class** eine Klasse angelegt, und die dazugehörige Bandbreite eingestellt werden, in diesem Fall 512kbit.

Schließlich können noch Paketfilter, welche die einzelnen Pakete ihren Klassen zuweisen mit **tc filter** gesetzt werden.

Mit "tc qdisc del dev eth1 root" kann die erstellte Konfiguration wieder entfernt werden.

## 4.6.2 mgen – Skripte

Mgen wurde in den Versuchszwecken sowohl als Staugenerator als auch für das Setzen von timestamps verwendet. Mithilfe von ping wurde als erstes der Staubereich (Pakete pro Zeiteinheit) durch Variieren des mit mgen erzeugten Hintergrundverkehrs herausgefunden. Nachdem die Daten ermittelt wurden konnten die vorliegenden Skripte umgeschrieben werden.

Hier zur Verdeutlichung das „middletraffic.mgen“ Script:

```
0.0 ON 10 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
30.0 ON 11 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
90.0 ON 12 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
120.0 ON 13 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]

30.0 ON 1 UDP SRC 5000 DST 192.168.0.2/5000 PERIODIC [87 972]

0.1 OFF 10
30.1 OFF 11
90.0 OFF 1
90.1 OFF 12
120.1 OFF 13
```

Als erstes werden je fünf Transmission Events (Übertragungsereignisse) mit den FlowId's 10, 11, 12, 13 und 1 definiert. Dabei sind die ersten vier Events, welche die 100 Byte Markierungspakete schicken. Das fünfte Event, das ab der 30. Sekunde eintrifft, schickt dann in der Sekunde 87 Pakete mit jeweils 972 Byte Nutzlast (d.h. 1000 Byte inklusive UDP – und IP – Header).

Folgende Scripts wurden für mgen erzeugt:

- **nottraffic.mgen**  
Kein Hintergrundverkehr, es werden nur die Markierungspakete (timestamps) gesendet.
- **lowtraffic.mgen**  
Es werden jeweils ab der 30. Sekunde 72 Pakete pro Sekunde mit je 972 Byte pro Paket gesendet.
- **middletraffic.mgen**  
Es werden jeweils ab der 30. Sekunde 87 Pakete pro Sekunde mit je 972 Byte pro Paket gesendet.
- **hightraffic.mgen**  
Es werden jeweils ab der 30. Sekunde 102 Pakete pro Sekunde mit je 972 Byte pro Paket gesendet.
- **burst.mgen**  
Ein Burstverkehr in der 60. Sekunde 720ms lang mit jeweils 200 Paketen (972 Byte pro Paket) wird generiert.

### 4.6.3 Perl – Skripte

Um die mit tcpdump erzeugten outputfiles zu parsen wurden Perlscripts eingesetzt. Das Ziel war dabei die outputfiles in ein Format zu bringen, das von Gnuplot visualisiert werden kann.

Folgende Perlscripts wurden verwendet:

- **divideproto.pl**

Dem Script wird als Argument das zu parsende tcpdump outputfile übergeben. Dieses wird dann nach verschiedenen Kriterien geparkt und in sieben Files geschrieben:

*UDP100*

Hier werden alle UDP Pakete gespeichert, die der Server zum Client schickt.

*UDP250*

Hier werden alle mgen Pakete gespeichert.

*UDPRest*

Hier werden alle anderen UDP Pakete gespeichert

*RTT*

Hier werden die ping Pakete gespeichert, berechnet als RTT.

Die weiteren drei Files TCP100, TCP250 und TCPRest waren nicht von Bedeutung, da kein TCP Verkehr beobachtet wurde.

- **parseTraffic.pl**

Die von divideproto.pl erzeugten Files, außer RTT, können dann Gnuplot gerecht weiter geparkt werden. Dem Script werden je zwei Argumente übergeben, das zu parsende File und der Name des zu erstellenden Files. So wird ein File im folgenden Format erstellt:

1.Spalte	Zeit in Sekunden
2.Spalte	Durchsatz in KByte/s
3.Spalte	Anzahl Pakete pro Sekunde

### 4.6.4 Gnuplot – Skripte

- **throughput.plt**

Visualisiert den Durchsatz der UDP – Pakete die vom Sender gesendet werden und derer, die beim Empfänger ankommen. Dabei werden 2 Kurven übereinander gelegt. So kann visualisiert werden wie die Anwendungen reagieren und wie viele Pakete verloren gehen.

- **ping.plt**

Visualisiert die Round Trip Time die mit ping gemessen wurde.

- **packetInSec.plt**

Visualisiert die in der Sekunde gesendeten UDP – Pakete vom Sender.

- **mgentraffic.plt**  
Visualisiert den Durchsatz der mgen UDP – Pakete, die vom Sender gesendet werden und derer, die beim Empfänger ankommen. Dabei werden 2 Kurven übereinander gelegt geplottet.
- **avPacketLenght.plt**  
Visualisiert die durchschnittliche UDP – Paketgröße die in der Sekunde vom Sender übermittelt wird.

## 4.7 Testfile

Als Testfile wurde eine Wavedatei verwendet. Diese wurde mit Hilfe des Programms VirtualDub Version 1.5.10 (freeware), welches eigentlich zur Videobearbeitung dient, erstellt. Dabei wurde der Sound aus einem Video – Testfile der Arbeit über „Netzwerkverhalten von Videostreaming Anwendungen“ extrahiert. Als Vorlage wurde das File „test\_stream\_max512.rm“ verwendet, welches ein aus einem Zusammchnitt mehrerer, schneller Szenen des Matrix Reloaded Trailers bestehendes Videofile darstellt. Das Sourcevideo, aus dem der eigentliche Zusammchnitt besteht ist unter folgendem Link abrufbar: <http://www.movie-infos.net> / Dateiname: „Matrix Reloaded – Trailer Deutsch (CinemaTV).avi“.

Die für diese Arbeit verwendeten Audiodateien hatten eine Abspielzeit von 4min bzw. 4:10min (File mit 10sec. Stille) und eine Dateigröße von 20,2MB bzw. 21,0MB. Für die verschiedenen Testfälle waren aber immer nur die ersten 120 Sekunden von Bedeutung.

## 4.8 Ablauf des Testverfahrens

Abbildung 3: Testablauf und Datenflussrichtung zeigt eine prinzipielle Struktur [3] des Testverfahrens und des Verkehrsflusses.

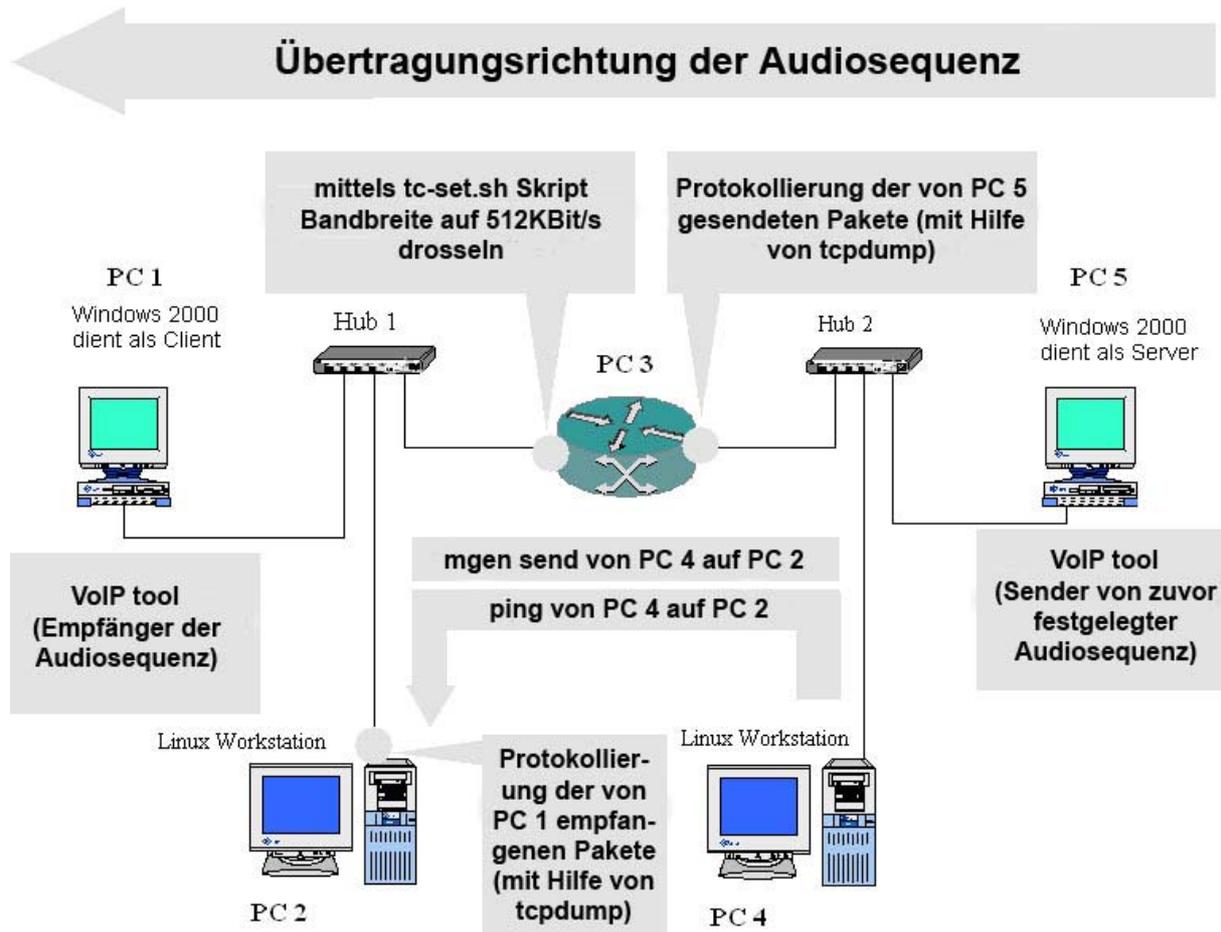


Abbildung 3: Testablauf und Datenflussrichtung

Vorerst müssen alle Tools und Anwendungen (wie in Abbildung 3 dargestellt) installiert und konfiguriert sein. Das für die Tests generierte Audiofile muss auf dem Windows 2000 Rechner, der die Daten versenden soll (PC 5), gespeichert sein.

### Ablauf:

1. Auf beiden Windows 2000 Rechner die zu testende Voice over IP Anwendung starten und eine Verbindung zwischen PC 1 zu PC 5 etablieren.
2. Tcpdump auf PC 2 mit dem Aufruf `tcpdump -i eth0 -vtttnn > tcpout.txt` in der shell starten.
3. Tcpdump auf PC 3 mit dem Aufruf `tcpdump -i eth2 -vtttnn > tcpin.txt` in der shell starten;

4. Tc auf PC 3 mit dem Script Aufruf `./tc-set.sh` setzen
5. Das eigens für die Tests generierte .wav – File auf PC 5 abspielen und mittels einem Koaxialkabel die analogen Audiodaten vom Audioausgang auf den Mikrophoneingang umlenken.
6. Ping auf PC 4 mit dem Aufruf `ping 192.168.0.2` in der shell starten
7. Mgen auf PC 4 mit dem Script Aufruf `mgen input <mgenscript.mgen>` in der shell starten
8. Nach 120 Sekunden beendet mgen auf PC 4 die Aktion. Danach können alle Anwendungen gestoppt werden.

Die zwei outputfiles tcpout.txt und tcpin.txt bleiben auf PC 2 bzw. PC 3 gespeichert und stehen zur späteren Visualisierung und Auswertung zur Verfügung. Dabei sind in tcpin.txt die vom Server gesendeten Pakete und in tcpout.txt die nach der Staustelle vorhandenen Pakete protokolliert. Die verwendeten Gnuplot Scriptfiles werden mit dieser Arbeit mitgeliefert, nähere Erläuterungen dazu sind unter dem Punkt Gnuplot – Skripte weiter oben angeführt.

## 5 Testergebnisse

Die graphische Ausarbeitung der gewonnenen Testergebnisse für alle Programme und Szenarien befinden sich im Anhang dieses Dokuments.

### 5.1 ICQ

#### 5.1.1 10sec Stille übertragen – kein Hintergrundverkehr

Beim ersten Test, bei dem von der 55. bis zur 65. Sekunde des Versuches nur Stille übertragen wurde, ist ersichtlich, dass ICQ auf diese Art der Veränderung reagiert.

Wie aus Diagr. I.S.1 ersichtlich sinkt der throughput in genau jenem Zeitraum auf zwischenzeitlich bis zu 0 KByte/s ab. Dafür sind sowohl eine Veränderung der Paketgrößen (Diagr. I.S.4), als auch eine Anpassung der gesendeten Pakete pro Sekunde (Diagr. I.S.4) ausschlaggebend.

#### 5.1.2 Kein Hintergrundverkehr

Hier gab es bei keinem der getesteten Voice over IP tools irgendwelche speziellen Vorkommnisse. Erwähnenswert sind jedoch die Unterschiede der annähernd konstanten Werte von Throughput (Diagr. I.N.1), Paketen pro Sekunde (Diagr. I.N.3) und der durchschnittlichen Paketlänge (Diagr. I.N.4). Bei einem durchschnittlichen Throughputwert von ca. 1.8 – 1.9 liegt ICQ im unteren Mittelfeld der getesteten Anwendungen.

#### 5.1.3 Wenig Hintergrundverkehr

In diesem Testfall bemerkte man zum ersten Mal einen minimalen Anstieg der Round Trip Time oder kurz RTT (Diagr. I.L.2). In allen anderen Bereichen blieb alles praktisch unverändert im Vergleich zum Testfall „kein Hintergrundverkehr“.

Der von mgen generierte Hintergrundverkehr von PC 4 wurde bis auf vernachlässigbare Ausreißer zur Gänze von PC 2 empfangen, wie man in Diagr. I.L.5 schön erkennen kann.

#### 5.1.4 Mittlerer Hintergrundverkehr

Ersichtlich aus Diagr. I.M.2 scheint hier die Round Trip Time zwischen etwas über 1 Sekunde und ca. 0,1 Sekunde hin und her zu pendeln, wobei ab der 30. Sekunde für etwa 5 – 10 Sekunden ein fast linearer Anstieg vorherrscht. Man bemerkt auch sehr schön, dass ICQ hier zum ersten mal Einbußen beim Empfangen von Paketen hinnehmen muss (Diagr. I.M.1). Ab dem Zeitpunkt, an dem der Hintergrundverkehr gestartet wird, sinkt die empfangen Datenmenge auf im Schnitt unter 1,5 KByte/s ab, wohingegen der Sender weiterhin versucht seine Informationen mit knapp unter ca. 1,9 KByte/s zu verschicken. Die Folge ist ein prozentuell schon relativ hoher Datenverlust von 9,98%.

Wie aus Diagr. I.H.5 gut ersichtlich werden längst nicht mehr alle von mgen gesendeten Daten bei PC 2 auch wieder empfangen.

### **5.1.5 Viel Hintergrundverkehr**

Dieser Fall ist sehr schön mit dem zuvor besprochenen „Mittlerer Hintergrundverkehr“ - Szenario vergleichbar, da sich im Wesentlichen nichts gravierend ändert. Da der Hintergrundverkehr zunimmt (Diagr. I.H.5) steigt die Round Trip Time (Diagr. I.H.2), deren Verhalten aber ansonsten dasselbe darstellt wie schon zuvor, und die Diskrepanz zwischen gesendeter und empfangener Datenmenge sowohl bei mgen (Diagr. I.H.5), als auch bei ICQ selbst (Diagr. I.H.1) an.

### **5.1.6 Ruckartig auftretender Hintergrundverkehr**

Diagr. I.B.1 zeigt wie ICQ auf einen ruckartigen auftretenden Hintergrundverkehr reagiert bzw. nicht reagiert. Zwar steigt kurzzeitig die Round Trip Time (Diagr. I.B.2), jedoch veranlasst dies ICQ nicht dazu von der normalen Anzahl der gesendeten Paketen pro Sekunde (Diagr. I.B.3) und oder der durchschnittlichen Paketgröße (Diagr. I.B.4) abzuweichen. Lediglich ein kurzer Sprung beim Empfangen der Pakete in Diagram I.B.1 ist zu erkennen. Dabei werden direkt nach dem Burst zuerst weniger Pakete empfangen als gesendet wurden. Dieser Umstand gleicht sich aber innerhalb der nächsten Sekunde wieder aus, sprich alle überfälligen Pakete scheinen ihr Ziel, wenn auch verspätet, aber doch, zu erreichen. Diese Verzögerung lässt sich durch Queuing erklären.

## **5.2 MSN**

### **5.2.1 10sec Stille übertragen – kein Hintergrundverkehr**

MSN reagiert auf dieses Testszenario mit einer Minderung der Anzahl von Paketen pro Sekunde (Diagr. M.S.3) sowie einer Reduktion der Paketgrößen (Diagr. M.S.4) während der wichtigen 10 Sekunden. Im Vergleich zu ICQ reagiert das Microsoft Tool langsamer auf diese Änderung, dann aber praktisch gleich. Der Throughput geht auch hier nach einigen Sekunden auf 0 (Diagr. M.S.1).

Auffallend bei dem ersten Test dieser Software sind die hohe Paketanzahl und deren Größe. Beide Kennwerte fallen deutlich höher aus, als bei allen anderen getesteten Programmen, wodurch in der Folge die hohen Throughputwerte aus Diagram M.S.1 resultieren.

### **5.2.2 Kein Hintergrundverkehr**

Hier sind außer den bereits auffallenden hohen Messwerten, wie zum Beispiel für den Throughput (Diagr. M.N.1), keine Besonderheiten zu erkennen. Auch die Round Trip Time bleibt auf einem konstanten Wert von beinahe 0, wie Diagramm M.N.2 zu entnehmen ist.

### **5.2.3 Wenig Hintergrundverkehr**

Der Testfall „wenig Hintergrundverkehr“ zeigt in Diagramm M.L.2 ab der 30. Sekunde eine linear steigende Round Trip Time, welche ab der 60. Sekunde ein geringfügiges Pendeln zwischen 0.8 und 1.8 Sekunden aufweist. Die Paketanzahl sowie die Paketgröße bleiben über die Zeit wieder konstant (Diagr. M.L.3 u. Diagr. M.L.4).

Aus Diagramm M.L.1 kann man erkennen, dass bereits bei wenig Hintergrundverkehr nicht mehr alle gesendeten Daten zeitgerecht übertragen werden können.

### **5.2.4 Mittlerer Hintergrundverkehr**

Diagramm M.M.1 zeigt, dass MSN, nach dem Starten des Hintergrundverkehrs in der 30. Sekunde, nach der 40. Sekunden auf den Paketverlust reagiert. Dabei wird der Durchsatz des Senders gesteigert, indem die Paketgröße (Diagr. M.M.4) von 200 Byte auf über 360 Byte erhöht wird. Die Round Trip Time verhält sich ähnlich wie im Test mit „wenig Hintergrundverkehr“ (Diagr. M.M.2), nur dass der Pendelzustand zwischen knapp unter 1 Sekunde und knapp unter 2 Sekunden schneller erreicht wird als zuvor.

### **5.2.5 Viel Hintergrundverkehr**

Wie aus Diagramm M.H.4 ersichtlich erhöht sich auch hier die Paketgröße im selben Zeitraum wie im vorherigen Testfall, und zwar auch auf knapp über 360. Natürlich gehen hier mehr Daten verloren (M.H.1).

MSN scheint ab einer bestimmten Hintergrundverkehrsmenge immer gleich zu reagieren, egal wie hoch der eigentliche Hintergrundverkehr dann genau ist.

### **5.2.6 Ruckartig auftretender Hintergrundverkehr**

Der Throughputgraph M.B.1 zeigt einen kurzzeitigen Datenverlust, der aber sofort nach dem Ende des ruckartigen Hintergrundverkehrs wieder endet. Die Round Trip Time schnell zwischenzeitlich auf 1,3 Sekunden (Diagr. M.B.2).

An der Paketanzahl (Diagr. M.B.3) sowie der Paketgröße (Diagr. M.B.4) scheint der ruckartige Verkehr spurlos vorüberzugehen. Hier sind keine Änderungen erkennbar.

## **5.3 Roger Wilco**

### **5.3.1 10sec Stille übertragen – kein Hintergrundverkehr**

Roger Wilco reagiert ebenso wie ICQ und MSN auf die Übertragung der 10 Sekunden Stille. Auch hier macht sich ein Rückgang des gesendeten Throuputs (Diagr. R.S.1) durch eine Reduktion der Paketanzahl (Diagr. R.S.3) sowie der Paketlänge (Diagr. R.S.4) bemerkbar. Auffällig ist dabei, dass die Anwendung für relativ lange Zeit wirklich gegen 0 KByte/s geht und dort dann auch verharrt, bis wieder Daten übermittelt werden.

### 5.3.2 Kein Hintergrundverkehr

Auf den ersten Blick auffällig bei diesem Testszenario ist die geringe Anzahl an versendeten Paketen, die Roger Wilco benötigt siehe Diagramm R.N.3. Die Anzahl liegt während des ganzen Testzeitraums von 120 Sekunden unter 5 Paketen und ist somit weit niedriger als bei allen anderen getesteten tools, bei denen der Wert zwischen 20 und 50 schwankt (je nach Programm). Da sich die Paketgröße, wie aus Diagramm R.N.4 ersichtlich, von ca. 120 Byte in etwa im Mittelfeld aller getesteten Anwendungen befindet, resultiert ein sehr niedriger Durchsatz (Diagr. R.N.1).

### 5.3.3 Wenig Hintergrundverkehr

In diesem Test, bei dem zum ersten Mal zusätzlich Hintergrundverkehr generiert wird, sind keine gravierenden Änderungen gegenüber dem vorherigen Testfall bemerkbar. Außer der geringfügig schwankenden Round Trip Time in Diagr. R.L.2 bleiben alle Werte praktisch dieselben wie zuvor.

### 5.3.4 Mittlerer Hintergrundverkehr

Hier fällt in Diagr. R.M.1 eine starke Differenz zwischen dem gesendeten und empfangenen Throughput auf. Zwischen der 30. und 60. Sekunde sind teilweise Stellen bemerkbar, an denen der Empfänger keine Daten mehr vom Sender erhält. Die Round Trip Time schnellst ab der 30. Sekunde linear auf 1,3 – 1,4 Sekunden und scheint dann zwischen diesem und einem Wert von etwa 0,35 hin und her zu pendeln, siehe dazu Diagramm R.M.2.

### 5.3.5 Viel Hintergrundverkehr

Die Round Trip Time aus Diagramm R.H.2 bleibt bis auf einen schnelleren Anstieg ab der 30. Sekunde praktisch dieselbe wie im Szenario mit mittlerem Hintergrundverkehr. Auch der Durchsatz (Diagr. R.H.1) verhält sich sehr ähnlich, nur mit dem Unterschied, dass noch weniger gesendete Daten den Empfänger erreichen.

### 5.3.6 Ruckartig auftretender Hintergrundverkehr

Der ruckartige Hintergrundverkehr bewirkt einen kurzzeitigen Anstieg der Round Trip Time auf mehr als 1 Sekunde (Diagr. R.B.2). Dies macht sich auch in Diagramm R.B.1 durch einen Datenverlust im selben Zeitraum bemerkbar. Die Lage pendelt sich jedoch nach 1 – 2 Sekunden wieder in den Normalzustand ein.

## **5.4 Skype**

### **5.4.1 10sec Stille übertragen – kein Hintergrundverkehr**

Wie aus Diagramm S.S.1 ersichtlich ist Skype das einzige getestete Voice over IP tool, welches nicht massiv auf die 10 Sekunden Stille in der zu übermittelnden Wave Datei reagiert. Zwar ist ein sehr geringer Abfall der Paketgröße in Diagramm S.S.4 zu bemerken, jedoch ist dieser so gering, dass es fraglich ist, ob er wirklich von der Übertragung der 10 Sekunden Stille herrührt, oder nur rein zufällig an dieser Stelle aufgetreten ist.

### **5.4.2 Kein Hintergrundverkehr**

Im Gegensatz zu allen anderen getesteten Anwendungen weißt Skype in Diagramm S.N.1 einen doch ziemlich unruhigen Bandbreitenverlauf auf. Dieser resultiert aus der schwankenden Paketlänge (Diagr. S.N.4), hingegen bleibt die Paketanzahl auf einem beinahe konstanten Wert von 33 (Diagr. S.N.3).

Skype befindet sich mit einem über die Zeit gemittelten Throughputwert von knapp unter 5 KByte/s ziemlich exakt im Mittelfeld aller Testprogramme.

### **5.4.3 Wenig Hintergrundverkehr**

Wie aus Diagramm S.L.1 ersichtlich reagiert Skype kaum auf den erzeugten UDP Hintergrundverkehr, zwar steigt die Round Trip Time in Diagramm S.L.2 geringfügig an, jedoch sind ansonsten keine weiteren Auswirkungen zu erkennen. Sowohl die gesendeten mgen Daten (Diagr. S.L.5) als auch die gesendeten Skype Daten erreichen zeitgerecht ihren jeweiligen Empfänger.

### **5.4.4 Mittlerer Hintergrundverkehr**

Das getestete Voice over IP tool zeigt hier in Diagramm S.M.2 einen raschen Anstieg der Round Trip Time ab der 30. Sekunde. Sogar einige Ausreißer von bis zu 2 Sekunden sind in diesem Diagramm zu erkennen. Im Gegensatz zum vorigen Szenario werden hier weder der mgen Verkehr, wie in Diagramm S.M.5 dargestellt, noch der Skype Verkehr (Diagr. S.M.1) zeitgerecht bzw. verlustfrei übertragen.

### **5.4.5 Viel Hintergrundverkehr**

Dieser Testfall unterscheidet sich nicht wesentlich vom vorigen. Aus den beiden Diagrammen S.H.1 sowie S.H.5 ist zu erkennen, dass von beiden Datenströmen weniger beim Empfänger einlangt als zuvor. Die Spitze im Thoughputdiagramm, die nach der 90. Sekunde auftritt, zeigt Daten an die zeitverzögert beim Empfänger eingehen.

#### **5.4.6 Ruckartig auftretender Hintergrundverkehr**

Hier zeigt sich bei Skype eine ähnlich starker Ausschlag im Throughputdiagramm (Diagr. S.B.1) wie im Vergleichsdiagramm von MSN (Diagr. M.B.1), wobei die Round Trip Time, welche in Diagramm S.B.2 zu sehen ist, weit unter der von MSN liegt. Skype verliert im Zeitraum des ruckartig auftretenden Hintergrundverkehrs mehr als 50% seiner gesendeten Daten.

## **5.5 Zusammenfassung der gewonnenen Ergebnisse**

### **5.5.1 Zusammenfassung ICQ**

ICQ weist im Schnitt über alle Testfälle den geringsten relativen Verlust auf, jedoch wurde das Programm rein von der subjektiv wahrgenommenen Audioqualität, die während der Tests beobachtet wurde, von Skype geschlagen, welches eine minimal bessere Qualität zu Tage legte. ICQ reagierte beim „10 Sekunden Stille“ Versuch sehr schön und senkte sowohl die Paketanzahl als auch die Paketgröße.

Der durchschnittliche Durchsatz war mit ca. 2 KByte/s der zweit – niedrigste im Test.

Leider reagierte ICQ auf aufkommenden Stau weder mit einer Variierung der Paketanzahl noch einer Änderung der Paketlänge und sendete unverändert mit derselben Throughputrate weiter.

### **5.5.2 Zusammenfassung MSN**

Bei MSN bemerkte man beim „10 Sekunden Stille“ Testfall eine Verminderung der Paketlänge und –größe. Dieses Voice – over – IP Tool erwies sich im subjektiven Audiovergleichstest als eines der beiden schwächeren Programme. Selbiges sagt auch die summierte prozentuelle Verlustrate aus.

Aufgefallen ist bei MSN, dass es als einzige Anwendung im Versuch auf Stau zu reagieren begann, jedoch wurde dabei die Throughputrate nicht gesenkt, wie es eine TCP friendly Anwendung machen würde, sondern zusätzlich erhöht. Dies erzielte MSN durch eine (beinahe) Verdoppelung der Paketgröße, wo hingegen die Anzahl der Pakete pro Zeiteinheit unberührt blieb.

### **5.5.3 Zusammenfassung Roger Wilco**

Roger Wilco beeindruckte weder im subjektiven Audiotestvergleich noch im aufsummierten Verlustprozentsatzvergleich. Das Tool reagierte zwar beim „10 Sekunden Stille“ Test in gleicher Art und Weise wie MSN, jedoch vermisste man auch hier eine Änderung des Sendeverhaltens bei auftretendem Stau.

Roger Wilco ist bei Online Gamern sehr beliebt, weil diese Anwendung sehr wenig Bandbreite beansprucht, jedoch belegt es in diesem Test leider den letzten Platz.

### **5.5.4 Zusammenfassung Skype**

Skype überraschte in dieser Versuchsreihe. Das Tool reagierte weder auf Stille\* in der Übertragung noch auf auftretenden Hintergrundverkehr und schaffte es dennoch, die beste Audioqualität zu liefern (subjektiver Eindruck!).

Mit einem durchschnittlichen Durchsatz von ca. 5 KByte/s, konstanter Paketanzahl pro Zeiteinheit und variierender Paketgröße schaffte es Skype im Vergleich des prozentuellen Verlustes auf den zweiten Platz.

\*es war zwar eine Reaktion bei diesem Test zu bemerken, allerdings war diese so gering, dass die Änderung in diesem Zeitraum auch Zufall sein könnte.

## 5.6 Datentransfer Tabelle

Anwendung	gesendet [Kbyte]	empfangen [Kbyte]	Verlust [Kbyte]	Anteil [%]
<b>FTP Download [3]*</b>				
Kein Verkehr	13977,92	13977,92	0,00	0,00
Wenig Verkehr	10433,03	10433,03	0,00	0,00
Mittlerer Verkehr	8069,77	8069,77	0,00	0,00
Viel Verkehr	6314,65	6303,77	10,88	0,17
Ruckartiger Verkehr	13833,83	13833,83	<u>0,00</u>	<u>0,00</u>
			<i>gesamt</i>	<i>0,17</i>
<b>ICQ</b>				
10 sec Stille, kein Verk.	208,92	208,92	0,00	0,00
Kein Verkehr	223,98	223,90	0,08	0,04
Wenig Verkehr	223,90	223,90	0,00	0,00
Mittlerer Verkehr	223,90	201,56	22,34	9,98
Viel Verkehr	223,82	196,44	27,38	12,23
Ruckartiger Verkehr	223,90	223,90	<u>0,0</u>	<u>0,00</u>
			<i>gesamt</i>	<i>22,25</i>
<b>MSN</b>				
10 sec Stille, kein Verk.	1150,54	1149,29	1,25	0,11
Kein Verkehr	1175,20	1173,63	1,57	0,13
Wenig Verkehr	1174,69	1164,84	9,85	0,84
Mittlerer Verkehr	1714,40	1476,84	237,56	13,86
Viel Verkehr	1714,31	1342,65	371,66	21,68
Ruckartiger Verkehr	1176,60	1174,44	<u>2,16</u>	<u>0,18</u>
			<i>gesamt</i>	<i>36,8</i>
<b>Roger Wilco</b>				
10 sec Stille, kein Verk.	48,61	48,61	0,00	0,00
Kein Verkehr	52,31	52,24	0,07	0,13
Wenig Verkehr	52,32	52,20	0,12	0,23
Mittlerer Verkehr	52,32	41,45	10,87	20,78
Viel Verkehr	52,20	34,93	17,27	33,08
Ruckartiger Verkehr	52,32	52,32	<u>0,00</u>	<u>0,00</u>
			<i>gesamt</i>	<i>54,22</i>
<b>Skype</b>				
10 sec Stille, kein Verk.	572,49	572,49	0,00	0,00
Kein Verkehr	572,48	572,37	0,11	0,02
Wenig Verkehr	581,45	581,45	0,00	0,00
Mittlerer Verkehr	581,77	534,93	46,84	8,05
Viel Verkehr	579,13	465,90	113,23	19,55
Ruckartiger Verkehr	576,98	576,62	<u>0,36</u>	<u>0,06</u>
			<i>gesamt</i>	<i>27,68</i>

\*Achtung! Die gemessenen Werte der einzelnen Szenarien des FTP Downloads beziehen sich auf eine andere Bandbreitenbeschränkung sowie anderen Hintergrundverkehr. Wichtig ist nur die letzte Spalte mit dem Verlustanteil in Prozent, der wie zu sehen bei „TCP friendly“ Anwendungen möglichst nahe bei Null liegen sollte.

## **6 Schlusswort**

### **6.1 Zusammenfassung der Arbeit**

Dem Messen sprich, der Gewinnung von exakten Messergebnissen, folgten eine Auswertung und eine möglichst genau Interpretation der Testläufe. Dazu war es zu Beginn erforderlich, sich in die netzwerktechnischen Grundlagen einzuarbeiten und das Grundprinzip von Voice – over – IP basierten Anwendungen zu verstehen. In den ersten Tests, in welchen die Anwendungen auf ihre Verträglichkeit bzw. Messbarkeit hin auf der Teststrecke untersucht wurden, stellte sich heraus, dass es sehr große Diskrepanzen in Hinsicht auf die grundlegende Funktionsweise gibt.

### **6.2 Wissenswertes**

Vor dem Start der eigentlichen Tests wurde eine kleine „Qualifikationsrunde“ gestartet, in welcher festgestellt wurde, ob die Anwendungen auf die zuvor festgelegte Art und Weise getestet werden können. Dabei wurde vor allem auf 2 Punkte Wert gelegt:

- die Datenübertragung des VoIP tools muss mittels UDP erfolgen und
- es muss eine direkte Datenübertragung (innerhalb der Teststrecke) von PC 5 auf PC 1 etabliert werden.

So zeigte sich bereits nach den ersten Tests, dass z.B. der Yahoo Messenger seinen UDP Audiostrom immer über einen speziellen Internetserver laufen lässt. Da der UDP Strom nicht direkt von PC 5 auf PC 1 lief, sondern über den Router (PC 3) ins Internet und zurück führte, wurde der Yahoo Messenger von den folgenden Tests ausgeschlossen, weil mit dieser Eigenheit keine gleich bleibenden Testbedingungen gewährleistet waren.

### **6.3 Danksagung**

- Mein Dank gilt Herrn Dr.-Ing. Michael Welzl für seine tatkräftige Unterstützung und die Betreuung dieser Arbeit.
- Weiters danke ich dem Institut für Informatik der Universität Innsbruck für die Bereitstellung aller nötigen Utensilien.
- Bedanken möchte ich mich auch bei den folgenden Studenten: Muhlis Akdag, Andreas Radinger und Marcus Fischer, welche mit ihrer Pionierarbeit auf dem Gebiet des Messens des Netzwerkverhaltens den gesamten Testablauf erleichterten. Besonders zu erwähnen wären dabei die für diese Arbeit benötigten Skriptfiles.

## 6.4 Literaturverzeichnis

- [1] Linux Advanced Routing and Traffic Control  
Bert Hubert, Thomas Graf, Gregory Maxwell, Remco van Mook, Martijn van Oosterhout, Paul B Schroeder, Jasper Spaans, Pedro Larroy – 31.03.2004  
<http://lartc.org/howto/>
- [2] VoIP Wissenswertes  
Roberto Arcomano – 07.08.2002  
<http://www.tldp.org/HOWTO/VoIP-HOWTO.html>
- [3] Teststreckenaufbau und Vergleichswerte  
Akdag Muhlis – 29.04.2004  
<http://www.welzl.at/teaching/baks/akdagmuhlis/BakArbeitAkdagMuhlis.pdf>
- [4] Enzyklopädie verwendet für Protokollerläuterungen  
<http://de.wikipedia.org/wiki/Hauptseite>