



**Leopold-Franzens-Universität
Innsbruck**

Institut für Informatik

Analyse des Netzwerkverhaltens

von Echtzeit-Multimedia-Internetanwendungen

Detailanalyse von MSN & Skype (VoIP)

Bakkalaureatsarbeit

eingereicht bei Dr. Ing. Michael Welzl

Autor: Thomas Rammer

Innsbruck, 05.12.2004

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis.....	2
2	Allgemeines.....	4
2.1	Einleitung	4
2.2	Aufgabenstellung	4
3	Theoretisches.....	6
3.1	Allgemeines.....	6
3.2	Fehlererkennung & Fehlerkorrektur.....	6
3.2.1	Rückwärtsfehlerkorrektur.....	6
3.2.1.1	Stop – and – Wait	6
3.2.1.2	Go – Back – N.....	6
3.2.2	Vorwärtsfehlerkorrektur.....	7
4	Visualisierung bisher gewonnener Ergebnisse.....	9
4.1	Gesamtverhalten von ICQ.....	9
4.2	Gesamtverhalten von MSN	9
4.3	Gesamtverhalten von Roger Wilco	10
4.4	Gesamtverhalten von Skype.....	11
4.5	Visualisierung des prozentuellen Datenverlustes.....	11
4.6	Analyse gestaffelt nach Szenarien:	12
4.7	Auffälligkeiten:	13
5	Versuchsdurchführung	14
5.1	Allgemeines zur Teststrecke	14
5.2	Aufbau der Teststrecke.....	14
5.3	Spezifikation der Teststrecke	15
5.3.1	Hardwarespezifikation	15
5.3.2	Softwarespezifikation.....	15
5.4	Tools für die Durchführung der Tests	16
5.4.1	Netzwerkspezifische Tools	16
5.4.2	weitere Tools	17
5.5	Testszenarioszenarien.....	18
5.5.1	Allgemeines.....	18
5.6	Skripte	18
5.6.1	tc – Skripte	19
5.6.2	mgen – Skripte	19
5.6.3	Perl – Skripte.....	20
5.6.4	Gnuplot – Skripte	21
5.7	Testfile.....	21
5.7.1	Allgemeines.....	21
5.7.2	Files mit einmaliger Pause	22
5.7.3	Files mit mehrmaliger Pause	23
5.8	Ablauf des Testverfahrens.....	24
6	Testergebnisse	27
6.1	MSN	27
6.1.1	Multi traffic steps 05 – 25	27
6.1.2	Multi traffic steps 30 – 50	27
6.1.3	Multi traffic steps 55 – 75	27
6.1.4	Multi traffic steps 80 – 100	28
6.1.5	Multi traffic steps 105 – 125	28
6.1.6	Multi traffic steps 130 – 150	28
6.1.7	Multi traffic steps 155 – 175	28

6.1.8	Multi traffic steps 180 – 200	28
6.1.9	Multi traffic steps 200 – 180	29
6.1.10	Multi traffic steps 175 – 155	29
6.1.11	Multi traffic steps 150 – 130	29
6.1.12	Multi traffic steps 125 – 105	29
6.1.13	Multi traffic steps 100 – 80	29
6.1.14	Multi traffic steps 75 – 55	30
6.1.15	Multi traffic steps 50 – 30	30
6.1.16	Multi traffic steps 25 – 05	30
6.2	Skype	30
6.2.1	Single pause 10 sec	30
6.2.2	Single pause 20 sec	30
6.2.3	Single pause 30 sec	31
6.2.4	Single pause 40 sec	31
6.2.5	Single pause 50 sec	31
6.2.6	Single pause 60 sec	31
6.2.7	Repeated pause 10 sec	31
6.2.8	Repeated pause 20 sec	32
6.2.9	Repeated pause 30 sec	32
6.2.10	Repeated pause 40 sec	32
6.2.11	Repeated pause 50 sec	32
6.2.12	Repeated pause 60 sec	32
6.3	Zusammenfassung der gewonnenen Ergebnisse	33
6.3.1	Zusammenfassung MSN	33
6.3.2	Zusammenfassung Skype	33
6.4	Datentransfer Tabelle	34
7	Schlusswort	35
7.1	Zusammenfassung der Arbeit	35
7.1.1	MSN	35
7.1.2	Skype	35
7.2	Danksagung	36
7.3	Literaturverzeichnis	36

2 Allgemeines

2.1 Einleitung

Diese Testreihe baut auf meine erste Arbeit [1], welche das Thema „Analyse des Netzwerkverhaltens von Echtzeit-Multimedia-Internetanwendungen - detaillierte Analyse von VoIP“ trägt, auf.

Da die letzte Auswertung mehrere interessante Gesichtspunkte zu Tage brachte und damit auch neue Fragen aufwarf, sollte mit einer erweiterten Testreihe noch genauer auf die Eigenheiten der einzelnen Tools eingegangen werden, um deren Verhalten in Bezug auf Stau oder Übertragungspausen exakter spezifizieren zu können. In beiden Fällen, welche in dieser Arbeit genauer unter die Lupe genommen werden, erschien die Reaktion der Software auf die verschiedenen Szenarien auf Anhieb genau dem Gegenteil von dem zu Entsprechen was man sich hätte erwarten können (siehe erste Arbeit). Interessant ist diese Analyse vor allem deswegen, weil in beiden Fällen, also sowohl bei Skype als auch bei MSN, ein größerer Datentransfer in den jeweils relevanten Szenarien vorherrschte als zuvor angenommen.

2.2 Aufgabenstellung

Die Arbeit teilt sich in folgende Teilbereiche auf:

1. Visualisierung bisher gewonnener Erkenntnisse zum leichteren Vergleich der einzelnen VoIP Tools.
2. Detailanalyse des Sprunges im Datendurchsatz von MSN
3. detaillierte Analyse des Verhaltens von Skype gegenüber auftretenden Pausen in der Übertragung

zu 1.) Um die grundlegenden Ergebnisse der ersten Arbeit auf einen Blick sichtbar zu machen bietet Kapitel 4 eine graphische Darstellung des Datendurchsatzes pro Szenario sowie einen Vergleich des Datenverlustes.

zu 2.) In der ersten Testreihe zeigt MSN eine interessante Verhaltensweise. Ab dem „Middletraffic – Szenario“, bei dem pro Sekunde 87 Pakete mit je 972 Byte pro Paket gesendet werden, wird ein Sprung von 10 KByte/s auf 18 KByte/s im Datendurchsatz ersichtlich. Dieser resultiert aus einer Erhöhung der Paketgröße, wobei die Paketanzahl annähernd konstant bleibt. Folgende Fragen werden mit dieser Arbeit geklärt:

- Sind die Erhöhung der Paketgröße und der damit verbundene Sprung im Datendurchsatz einmalig oder wiederholt er sich bei weiterer Steigerung des Hintergrundverkehrs?
- Wird der Sprung im Datendurchsatz immer durch eine Erhöhung der Paketgröße ausgelöst oder beginnt auch die bisher annähernd konstante Paketanzahl zu variieren?
- Beeinflusst zu- oder abnehmender Hintergrundverkehr das bisher beobachtete Verhalten in irgendeiner Weise?

zu 3.) Ein Szenario der ersten Testreihe beschäftigt sich mit dem Verhalten von VoIP Tools in Bezug auf Stille in der Übertragung. Dabei wird im Test ein Soundfile übermittelt, in welchem in der Mitte 10 sec Stille aufgezeichnet sind. ICQ, MSN sowie Roger Wilco reagieren auf dieses Szenario mit einer Verminderung der Paketanzahl sowie der Paketgröße. Bei Skype hingegen bleibt die Anzahl der Pakete pro Sekunde konstant. Ob der sehr geringe Abfall der Paketgröße während des entscheidenden Zeitbereichs von 10 sec Zufall ist, oder ob Skype auf Stille in der Übertragung reagiert, wird in dieser Arbeit geklärt. Dabei werden folgende Fragen beantwortet:

- Reagiert Skype auf einmalige Pausen in der Übertragung, wenn ja wie?
- Reagiert Skype auf mehrmalige Pausen in der Übertragung, wenn ja wie?

3 Theoretisches

3.1 Allgemeines

Da das bisher gezeigte Verhalten von MSN, aus der ersten Arbeit, auf eine Art der Vorwärtsfehlerkorrektur (Forward Error Correction) schließen lässt, werden in diesem Abschnitt die Grundlagen von Fehlerkorrekturverfahren erläutert.

3.2 Fehlererkennung & Fehlerkorrektur

Unter Fehlererkennung und – korrektur [3] versteht man Verfahren um etwaig anfallende Übertragungsfehler zu erkennen bzw. auszubessern. Die Grundidee stellt dabei eine um Prüfdaten erweiterter Übertragung dar. Folglich kommt es bei einem solchen Übertragungscode immer zu einer Erhöhung der eigentlichen, zu transferierenden Datenmenge. Die im Code integrierten Prüfdaten können an sich beliebig gestaltet sein, sollten jedoch möglichst viel Aufschluss über die zu übermittelnden Daten geben. Zum Beispiel kann man auch eine wiederholte Übertragung derselben Daten als Senden von Prüfdaten ansehen. Inwiefern das jedoch sinnvoll wäre, sei dahingestellt. Zudem ist zu bedenken, dass ein Mehraufwand an Datentransfer mit einem höheren Leistungsaufwand einhergeht.

3.2.1 Rückwärtsfehlerkorrektur

Bei der Rückwärtsfehlerkorrektur erkennt der Sender keine drohenden Übertragungsstörungen. Folglich erfolgt die notwendige Fehlerbehandlung beim Empfänger, welcher eine Sendewiederholung beim Sender beantragt. Nach diesem Prinzip funktionieren z.B. die ARQ – Protokolle (Automatic Repeat – reQuest), welche in TCP zur Anwendung kommen. Bei diesen Protokollen verständigen sich Empfänger und Sender mittels ACK/NAK Signalen (acknowledgement bzw. negative acknowledgement) um einen korrekten Paketeneingang zu bestätigen bzw. Übertragungsfehler zu melden. Dabei unterscheidet man mehrere Protokollarten, welche sich meist durch ihren erzielbaren Datendurchsatz unterscheiden.

3.2.1.1 Stop – and – Wait

Beim „Stop – and – Wait“ Verfahren wartet der Sender so lange mit dem Versenden des nächsten Paketes, bis der Empfänger seine Empfangsbestätigung zurückschickt.

3.2.1.2 Go – Back – N

Das „Go – Back – N“ Verfahren hingegen arbeitet effizienter. Der Empfänger bestätigt korrekt empfangene Dateneinheiten, und kann auch mehrere Dateneinheiten auf einmal quittieren. Bei einem Fehler muss der Sender alle noch nicht quittierten Daten noch einmal senden. Er geht also zurück zur zuletzt bestätigten Sequenznummer (siehe Abbildung 1).

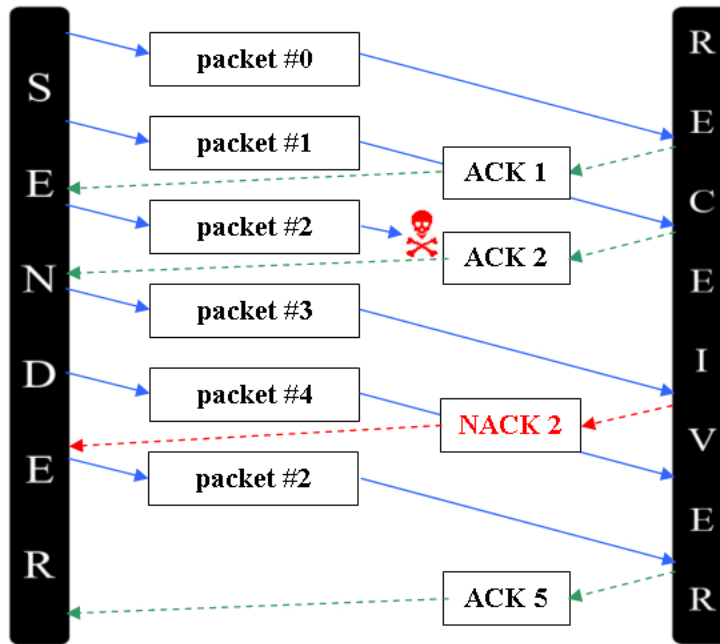


Abbildung 1: „Go – Back – N“ Verfahren

 ... Übertragungsfehler

3.2.2 Vorwärtsfehlerkorrektur

Bei der Vorwärtsfehlerkorrektur versucht der Sender selbst die Möglichkeit von auftretenden Übertragungsfehlern zu minimieren, wobei er äußeren Störeinflüssen entgegenwirkt, indem er die Übertragung durch das Hinzufügen relevanter Prüfdaten ergänzt. Diese Prüfdaten sollten es dem Empfänger ermöglichen, die Ausgangsnachricht im Falle eines Störfalles, bzw. fragmentarischen Datenverlustes zu rekonstruieren und somit einen reibungslosen Ablauf zu gewährleisten (siehe Abbildung 2).

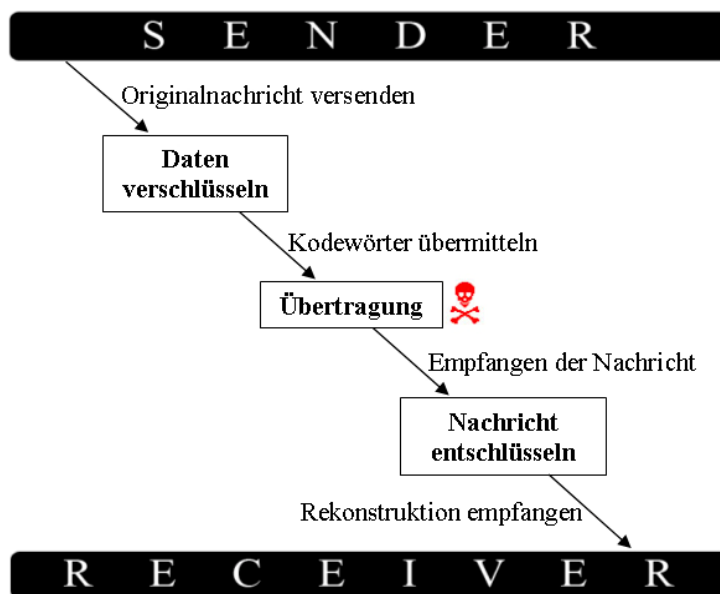


Abbildung 2: Grundidee der Vorwärtsfehlerkorrektur

 ... möglicher Fehler in der Übertragung

Wie Abbildung 2 zeigt, wird dabei eine zu übertragende Nachricht vor der Übermittlung vom Sender kodiert und vom Empfänger nach der Übertragung wieder dekodiert. Die Grundidee, welche hinter einem so genannten Fehlerkorrekturcode steckt, kann an sich relativ einfach sein. Um eine Möglichkeit der Kodierung und Dekodierung aufzuzeigen, folgt ein kurzer Exkurs in die Mathematik [6][5]:

Zwei Zahlen 100111000 und 001111100 seien in unserem Fall „Codewörter“, da sie mittels eines Fehlerkorrekturcodes übertragen werden sollen.

Die Distanz dieser beiden Zahlen sei definiert als die Anzahl jener Stellen an denen die Ziffer i des ersten ungleich der Ziffer i des zweiten Codewortes ist ($0 < i < 10$, für neunstellige Codewörter). In unserem Fall wäre die Distanz = 3, da sich die beiden Codewörter an den Stellen $i=1$, $i=3$ und $i=7$ unterscheiden.

In Abbildung 3 sind Codewörter mit einem schwarzen Punkt dargestellt und die Kreise um die Codewörter stellen die maximale Distanz dar, die ein empfangenes Wort zu einem Codewort haben kann, um eindeutig jenem zugewiesen, bzw. als jenes identifiziert zu werden. Dabei ist es wichtig, dass die Kreise disjunkt sind, damit jedes ankommende Wort (für den Fall, dass es innerhalb eines Kreises liegt) exakt dem zugehörnden Codewort zugewiesen und damit rekonstruiert werden kann. Liegt ein empfangenes Wort außerhalb eines solchen Kreises, war die Störung in der Übertragung zu groß um eine Rekonstruktion zu gewährleisten. Folglich geht die Information verloren.

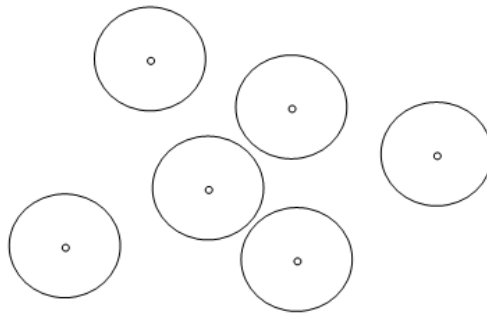


Abbildung 3: Codewörter und Distanzen

Heutzutage gängige Vorwärtsfehlerkorrekturverfahren bauen auf dieser oder ähnlichen Ideen auf, wodurch Daten welche durch die Übertragung zu Schaden gekommen sind, durch Zusatzinformationen in der Übermittlung wieder rekonstruiert werden können.

4 Visualisierung bisher gewonnener Ergebnisse

In diesem Abschnitt werden bisher gewonnene Ergebnisse, bzw. Zusammenhänge zur Veranschaulichung und zum besseren Verständnis graphisch dargestellt.

4.1 Gesamtverhalten von ICQ

* (1) 10 sec Stille, kein Verkehr (2) Kein Verkehr (3) Wenig Verkehr (4) Mittlerer Verkehr (5) Viel Verkehr

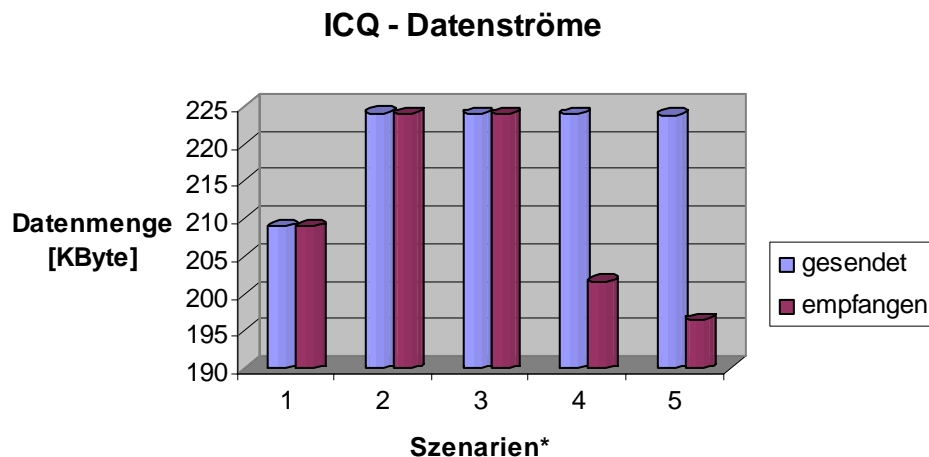


Abbildung 4: Verhalten von ICQ

Bei ICQ zeigen sich zwei charakteristische Merkmale. Zum einen wird im ersten Testszenario (10 sec Pause) eine starke Verringerung des Datendurchsatzes deutlich. Zum anderen erkennt man in den beiden Szenarien mit höherem Hintergrundverkehr, dass sich die empfangene Datenmenge merklich verringert. ICQ zeigt jedoch ansonsten keine Eigenschaft, welche darauf schließen lassen würde, dass das Tool den auf ihn einwirkenden Hintergrundverkehr wahrnehmen würde (dies wird durch die unveränderten „gesendet Balken“ der Szenarien 2 – 6 verdeutlicht).

4.2 Gesamtverhalten von MSN

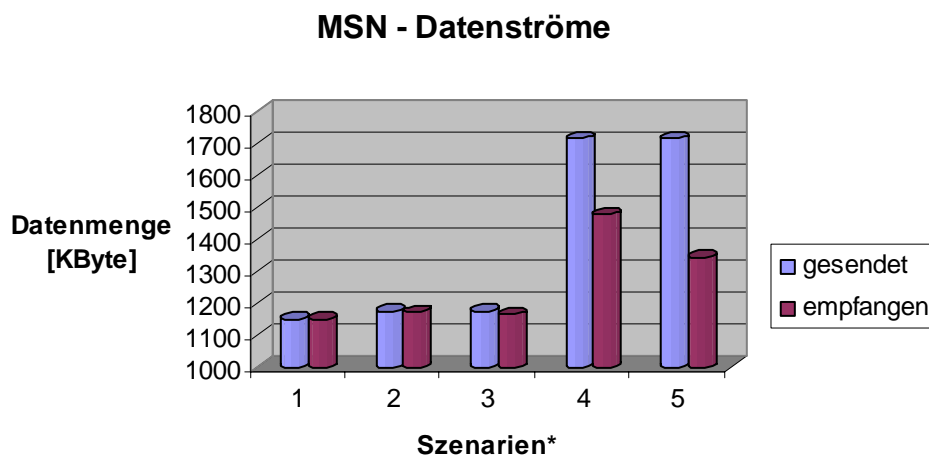


Abbildung 5: Verhalten von MSN

Bei MSN scheint im ersten Szenario keine wesentliche Veränderung im Datendurchsatz, der über die ganze Laufzeit von 120 sec gerechnet wird, sehbar. Jedoch täuscht diese Darstellung, da MSN, wie aus den throughput Diagrammen der letzten Arbeit ersichtlich, sehr wohl auf auftretende Übertragungspausen reagiert. Diese fälschlich anmutende Darstellung (siehe Abbildung 5) resultiert aus dem höchsten Datendurchsatz aller VoIP Tools, den MSN in durchgehend allen Szenarien aufweist und der dadurch resultierenden, gedrängten Darstellung auf der Ordinate und das trotz eines bereits sinnvoll gewählten Wertebereichs. Jedoch ist die Reaktion von MSN sehr gering und daher nur sehr schlecht darstellbar. Weiters ist zu bemerken, dass das Tool aus dem Hause Microsoft als einziges auf Hintergrundverkehr reagiert (wenngleich auch nicht wie erwartet). Aus den beiden Szenarien 4 und 5 ablesbar steigert MSN seinen Datendurchsatz um etwa 600 KByte – wieder gerechnet auf die komplette Testlaufzeit von 120 sec.

4.3 Gesamtverhalten von Roger Wilco

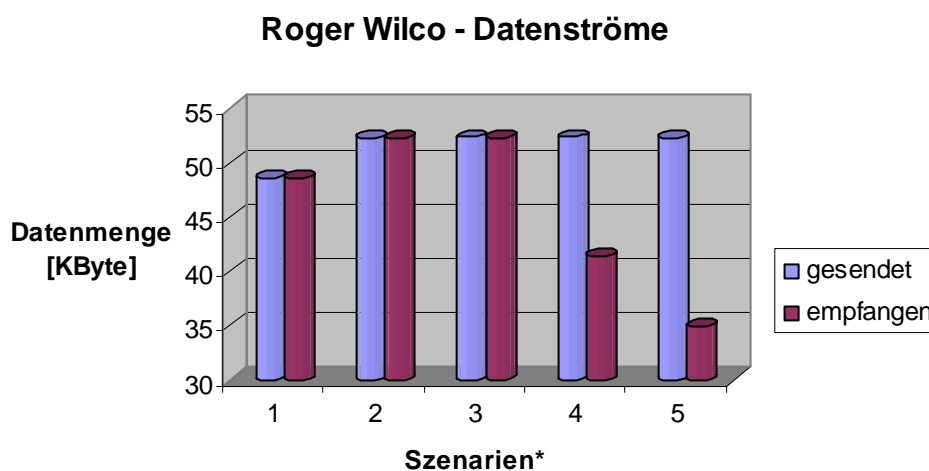


Abbildung 6: Verhalten von Roger Wilco

Mit im Höchstfall knapp über 50 KByte (in 120 sec.) zeigt Roger Wilco die geringste anfallende Datenmenge. Das Tool reagiert, wie aus Szenario 1 ersichtlich, auf anfallenden Übertragungspausen, aber nicht auf Hintergrundverkehr. Auch sofort zu erkennen ist, dass Roger Wilcos Übertragungsrate in Szenario 4 sowie 5 durch den hohen Hintergrundverkehr um bis zu 33% reduziert wird.

4.4 Gesamtverhalten von Skype

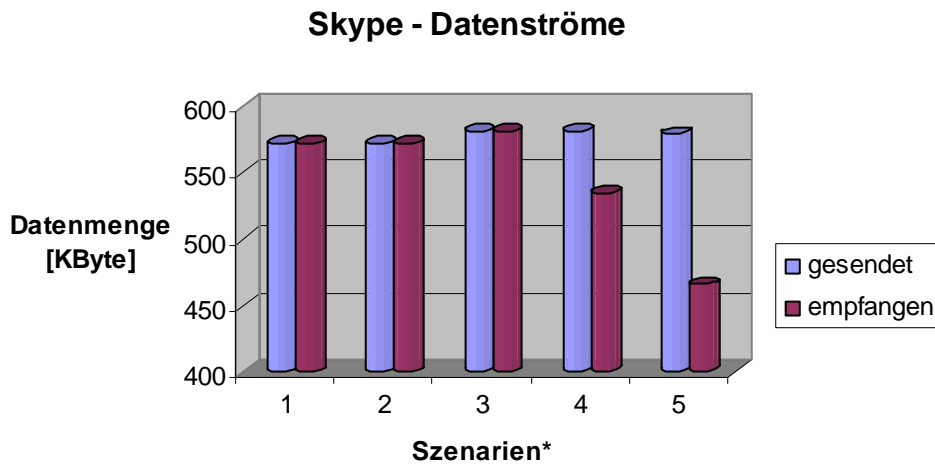


Abbildung 7: Verhalten von Skype

Skype zeigt sich, im Vergleich zu allen anderen getesteten VoIP Tools, von anfallenden Übertragungspausen am wenigsten beeinflusst, zeigt aber wie beinahe alle anderen auf die beiden Szenarien mit mittel und viel Hintergrundverkehr einen nicht unbeachtlichen Datenverlust in der Übertragung. Allerdings sei noch einmal zu erwähnen, dass Skype die beste (subjektive) Übertragungsqualität der getesteten Anwendungen zu Tage legte.

4.5 Visualisierung des prozentuellen Datenverlustes

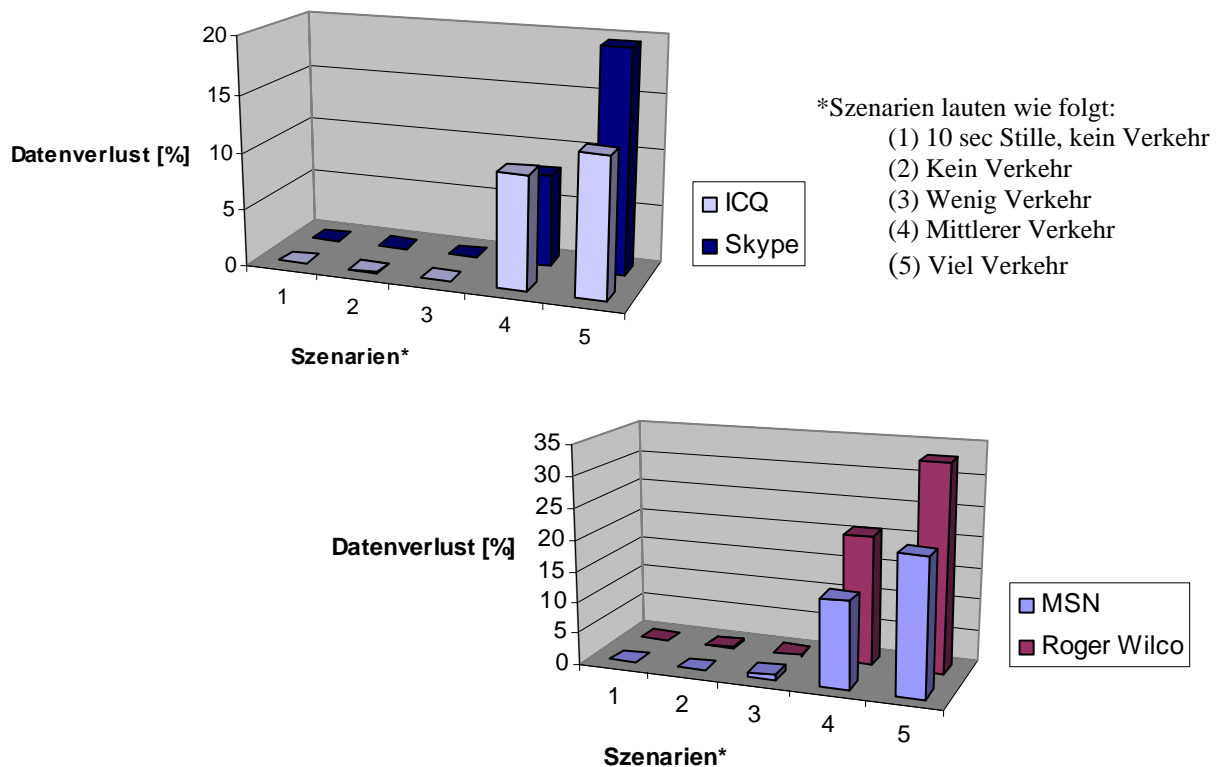


Abbildung 8: prozentueller Datenverlust

Die beiden oberen Diagramme visualisieren den prozentuellen Datenverlust der getesteten VoIP Tools. Da die Darstellung in einer einzelnen Graphik zu unübersichtlich war, wurden die beiden Tools mit höherem Datenverlust (untere Abbildung) getrennt von jenen mit geringerem Verlust (oberes Diagramm) dargestellt.

4.6 Analyse gestaffelt nach Szenarien:

- Szenario 1 (10 sec Stille, kein Verkehr):
Wie zu erwarten ist hier kein Datenverlust ersichtlich. Die VoIP Tools laufen in diesem Test ohne jegliche Belastung durch Hintergrundverkehr und werden lediglich auf ihre Reaktion auf anfallende Übertragungspausen getestet.
- Szenario 2 (Kein Verkehr):
Auch hier bestätigt sich die Annahme: Keine einwirkenden Störfaktoren, folglich auch kein Datenverlust.
- Szenario 3 (Wenig Verkehr):
Dieses Szenario zeigt erste merkliche Datenverluste bei MSN. Alle anderen VoIP Tools reagieren in nur sehr geringem (hier nicht ersichtlichem) Ausmaß auf den gestarteten Hintergrundverkehr von 72 Paketen pro Sekunde.
- Szenario 4 (Mittlerer Verkehr):
Ab einem Hintergrundverkehr von 87 Paketen pro Sekunde steigt der prozentuelle Datenverlust aller getesteten Anwendungen sprunghaft an. Dabei reicht die Spannweite von noch akkuraten 8,0% bei Skype bis hin zu beachtliche 20,8% Verlust bei Roger Wilco.
- Szenario 5 (Viel Verkehr):
Während bei den restlichen drei Anwendungen praktisch im selben Maße die Verlustrate steigt, versäumt es Skype mit einem Zuwachs von 11,5% gegenüber dem vorherigen Testfall nur denkbar knapp MSN von Platz zwei zu verdrängen. Die höchste Verlustrate zeigt aber wiederum Roger Wilco, bei dem ein Drittel aller gesendeten Daten bei der Übertragung verloren gehen.
- Szenario 6 (Ruckartiger Verkehr):
Dieses Szenario zeigte gleich wie die ersten, wenn überhaupt – nur eine sehr geringe Reaktion. Jedoch zeigt diese Arbeit später noch, dass ein solcher plötzlicher ruckartiger Verlust auch ohne das Einwirken äußerer Störfaktoren vorkommen kann.

4.7 Auffälligkeiten:

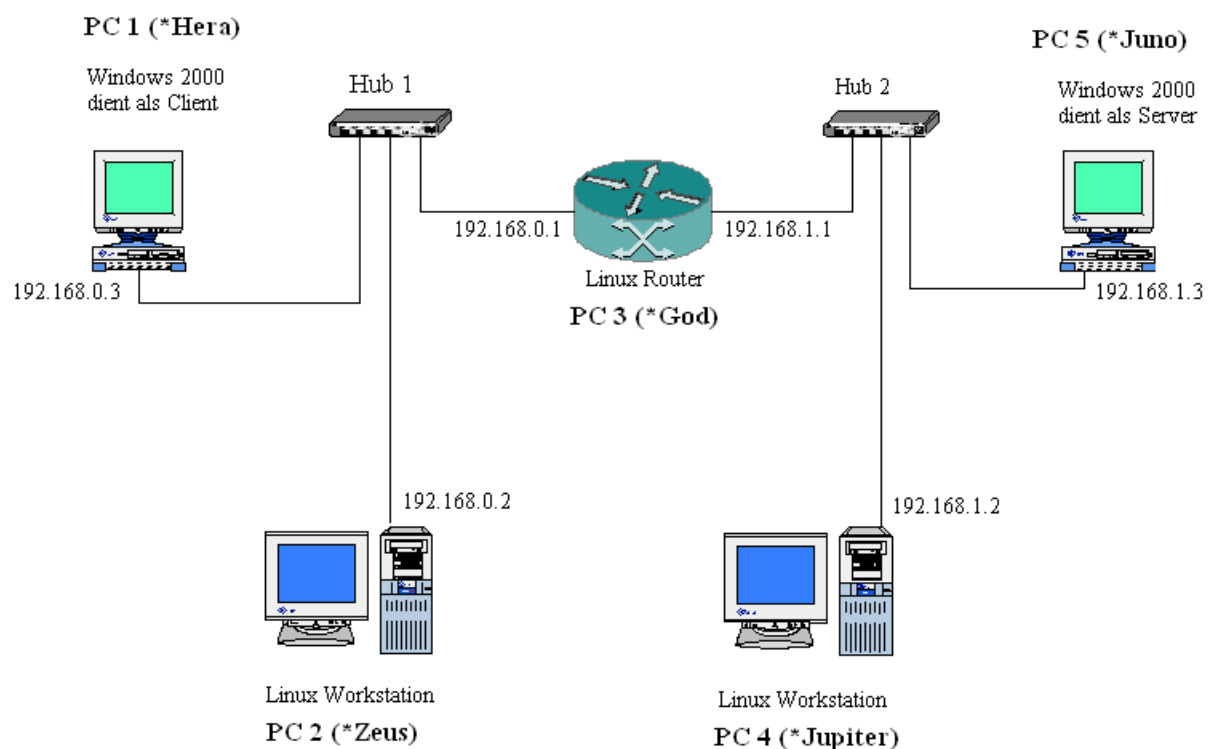
Wie oben beschrieben kann das Verhalten der einzelnen Tools nachvollzogen werden, wenngleich auch oft in keiner Weise auf auftretenden Hintergrundverkehr reagiert wird. Allerdings stellt sich immer noch die Frage warum MSN, wie in Abbildung 5 zu sehen, eine Erhöhung des Datendurchsatzes bei steigendem Hintergrundverkehr hervorruft und nicht „TCP friendly“ sprich mit einer Verminderung des Datendurchsatzes reagiert. Zudem scheint es mysteriös, dass Skype mit der subjektiv besten Übertragungsqualität in keiner Weise auf auftretende Übertragungspausen reagiert. Aus diesem Grund behandelt diese Arbeit genau jene beiden noch fragwürdigen Sachverhalte.

5 Versuchsdurchführung

5.1 Allgemeines zur Teststrecke

Die Teststrecke wurde vom Institut für Informatik zur Verfügung gestellt und ermöglichte eine genaue 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 und während der Laufzeit der anfallende Netzwerkverkehr zwischen diesen beiden PCs gemessen wird. Dabei wird ein zuvor genau festgelegtes Protokoll durchlaufen.

5.2 Aufbau der Teststrecke



* Namen der Computer in der verwendeten Teststrecke

Abbildung 9: 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 stellt sich wie in Abbildung 9 abgebildet dar [4].

5.3 Spezifikation der Teststrecke

5.3.1 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 (100MBit)
RAM: 512MB DDR RAM

5.3.2 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: Aufzeichnung 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

5.4 Tools für die Durchführung der Tests

5.4.1 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 anderer sehr bekannter Sniffer wäre zum Beispiel Ethereal, welcher 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 ein Paket vom localhost zum Zielhost und zurück braucht.

Name: mgen
Verwendungszweck: Erzeugung 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 [2] 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.

5.4.2 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: Audacity 1.2.2
Verwendungszweck: Generierung der benötigten Audiofiles für die verschiedenen Testszenerarien (z.B.: mehrfache Pausen in einem wav – File).

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

5.5 Testszzenarien

5.5.1 Allgemeines

Wurden in [1] alle Programme denselben Tests unterzogen, so werden in dieser Arbeit die verschiedenen Szenarien genau auf die jeweiligen Programme zugeschnitten. Genauer gesagt, sind die Tests nicht auf die Programme zugeschnitten, sondern vielmehr auf deren Eigenheiten, die sich nach der Auswertung der ersten Versuchsergebnisse offenbaren.

Da sich in der ersten Arbeit zeigte, dass MSN ab einer gewissen Hintergrundverkehrsmenge anfängt den Datendurchsatz durch einen ruckartigen Sprung in der Paketgröße zu erhöhen, werden folgende zwei Testläufe generiert:

1. multi traffic steps – increasing
2. multi traffic steps – decreasing

Sie sollen zeigen:

- bei welcher Menge an Hintergrundverkehr der bereits bemerkte Anstieg des Datendurchsatzes liegt.
- ob noch andere Erhöhungen bzw. Schwankungen der Paketgröße gemessen werden können.
- ob ein kontinuierlich ansteigender bzw. abfallender Hintergrundverkehr einen Einfluss auf die Steigerung (oder mögliche Senkung) des Datendurchsatzes hat.

Bei Skype drängte sich nach den ersten Testläufen die Frage auf, ob dieses VoIP Tool auf Pausen reagiert oder nicht. Im Gegensatz zu allen anderen Programmen konnte dies nicht mit Sicherheit festgestellt werden. Zwar zeigte sich beim „10 sec Stille übertragen – kein Hintergrundverkehr“ – Szenario während der fraglichen 10 sec ein geringer Abfall des Datendurchsatzes, jedoch wäre es möglich, dass eine so geringe Schwankung auch durch Zufall in genau diese untersuchte Zeitspanne von 10 sec, gefallen ist und nichts mit der Pause in der Übertragung zu tun hat. Um das Verhalten genauer zu betrachten werden folgende Szenarien generiert:

1. single pause
2. repeated pauses

Diese beiden Testszzenarien sollen zeigen:

- ob Skype auf Pausen in der Übertragung reagiert.
- falls es reagiert, in welcher Art
- ob ein Unterschied bei verschiedenen Pausenlängen zu bemerken ist.
- ob sich mehrmalige Pausen derselben Länge anders auswirken als einmalige.

5.6 Skripte

Bei dieser Arbeit wurden Skripte [4] 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

5.6.1 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 unserem Fall 512kbit.

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

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

5.6.2 mgen – Skripte

Mgen wurde in den Versuchszwecken sowohl als Staugenerator als auch für das Setzen von timestamps verwendet. Im Falle der MSN Tests wurden beide Eigenschaften von mgen benötigt. Bei Skype hingegen wurde nur die Möglichkeit des Timestampssetzens verwendet um die Versuche reibungslos und einheitlich durchzuführen.

Zur Verdeutlichung ein mgen Skript aus der MSN Testreihe mit Namen „3-MultiTrafficSteps_75-55.mgn“:

```
#####
# MGEN Skript
#####

#Timestamppakete:
0.0 ON 10 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
20.0 ON 11 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
40.0 ON 12 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
60.0 ON 13 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
80.0 ON 14 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
100.0 ON 15 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
120.0 ON 16 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]
140.0 ON 17 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 100]

#UDP - Strom Init & Anfang:
20.0 ON 1 UDP SRC 5000 DST 192.168.0.2/5000 PERIODIC [75 972]
40.0 ON 2 UDP SRC 5000 DST 192.168.0.2/5000 PERIODIC [70 972]
60.0 ON 3 UDP SRC 5000 DST 192.168.0.2/5000 PERIODIC [65 972]
80.0 ON 4 UDP SRC 5000 DST 192.168.0.2/5000 PERIODIC [60 972]
100.0 ON 5 UDP SRC 5000 DST 192.168.0.2/5000 PERIODIC [55 972]

#Ende aller Timestamps sowie UDP - Ströme
0.1 OFF 10
```

```
20.1 OFF 11
40.0 OFF 1
40.1 OFF 12
60.0 OFF 2
60.1 OFF 13
80.0 OFF 3
80.1 OFF 14
100.0 OFF 4
100.1 OFF 15
120.0 OFF 5
120.1 OFF 16
140.1 OFF 17
```

3-MultiTrafficSteps_75-55.mgn

Als erstes werden 8 Markierungspakete in einem Zeitabstand von je 20 Sekunden gesetzt. Danach folgen 5 UDP – Datenströme mit unterschiedlichen Paketanzahlen (von 75 – 55 Pakete – in 5er Schritten erniedrigt). Am Ende des Files werden der Reihe nach alle Datenströme beendet. Nach der Abarbeitung der letzten Zeile beendet sich mgn.

Die Files für die verschiedenen Skype Szenarien ähneln sehr jenen der MSN Tests, jedoch mit dem einen Unterschied, dass bei den mgn Files für Skype der komplette Block, welcher mit „#UDP - Strom Init & Anfang:“ gekennzeichnet ist, entfällt, da bei den Skype Tests, wie oben schon erwähnt, nur Markierungspakete, jedoch kein Hintergrundverkehr benötigt wurden.

5.6.3 Perl – Skripte

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

Folgende PerlSkripts wurden verwendet:

- **divideproto.pl**

Dem Skript 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 mgn 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 geparst werden. Dem Skript 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 der Pakete pro Sekunde

5.6.4 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.

5.7 Testfile

5.7.1 Allgemeines

Als Ausgangsfile für die erweiterten Tests wurde dasselbe Testfile wie in der ersten Testreihe verwendet. Dieses wurde mit Hilfe des Programms VirtualDub Version 1.5.10, 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 Zusammenschnitt mehrerer, schneller Szenen des Matrix Reloaded Trailers bestehendes Videofile darstellt. Das Sourcevideo, aus dem der eigentliche Zusammenschnitt besteht ist unter folgendem Link abrufbar: <http://www.movie-infos.net/> Dateiname: „Matrix Reloaded – Trailer Deutsch (CinemaTV).avi“.

Das Ausgangsfile wurde für die Skype Tests insofern modifiziert, als dass einmalige, bzw. mehrmalige, periodisch auftretende Pausen generiert wurden. Für die MSN Tests wurden keine Änderungen am besagten Ausgangsfile vorgenommen.

Hier eine Liste der für diese Arbeit verwendeten Audiofiles:

Dateiname	Dateilänge [sec]	Dateigröße [MB]
matrix_reloaded.wav*	140	20,2
10sec_pause_einmalig.wav	150	12,6
20sec_pause_einmalig.wav	150	12,6
30sec_pause_einmalig.wav	150	12,6
40sec_pause_einmalig.wav	150	12,6
50sec_pause_einmalig.wav	150	12,6
60sec_pause_einmalig.wav	150	12,6
10sec_pause_mehrfach.wav	100	8,4
20sec_pause_mehrfach.wav	175	14,7
30sec_pause_mehrfach.wav	245	20,6
40sec_pause_mehrfach.wav	315	26,4
50sec_pause_mehrfach.wav	380	31,9
60sec_pause_mehrfach.wav	450	37,8

Tabelle 1: Audiotestfiles

*beschreibt die Ausgangsdatei aus der alle weiteren Files generiert wurden

5.7.2 Files mit einmaliger Pause

Wie aus Tabelle 1 ersichtlich besitzen alle Testfiles, die eine einmalige Pause beinhalten, dieselbe Abspieldauer von 150 sec und somit auch die gleiche Dateigröße. Für die Tests relevant waren jedoch nur die ersten 120 sec.

Abbildung 10, sowie Abbildung 11 zeigen zwei Files mit einer einmaligen Pause. Diese Files kamen in der detaillierteren Analyse von Skype zum Einsatz.

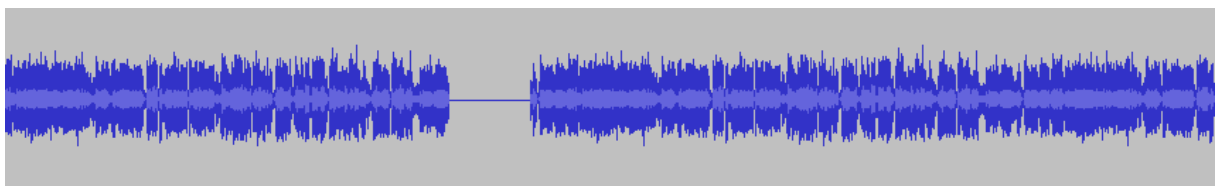


Abbildung 10: Tonspur von „10sec_pause_einmalig.wav“



Abbildung 11: Tonspur von „60sec_pause_einmalig.wav“

5.7.3 Files mit mehrmaliger Pause

Diese Audiofiles wurden gleich wie jene mit einfacher Pause für die Detailanalyse von Skype herangezogen. Ebenso wie oben sieht man auch hier die Audiospuren der Files mit den kürzesten, sowie mit den längsten Pausen. Natürlich variiert hier die Testdauer, da diese immer aus einem Vielfachen der Pausenlänge plus Auslaufzeit resultiert. Genauer gesagt aus dem Siebenfachen. Das Zeitintervall in dem eine Audioübertragung stattfindet entspricht genau dem der jeweiligen Pause. Folglich bedeuten 10 sec Pause auch 10 sec Übertragung – schön zu sehen in den unteren beiden exemplarischen Abbildungen (Abbildung 12 und Abbildung 13).

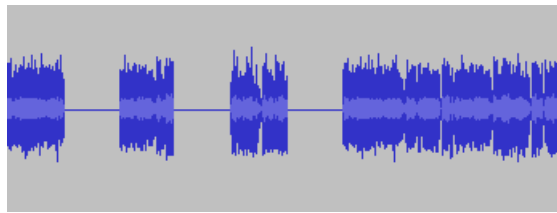


Abbildung 12: Tonspur von „10sec_pause_mehrfach.wav“



Abbildung 13: Tonspur von 60sec_pause_mehrfach.wav

5.8 Ablauf des Testverfahrens

Abbildung 14: Testablauf und Datenflussrichtung zeigen eine prinzipielle Struktur [4] des Testverfahrens und des Verkehrsflusses.

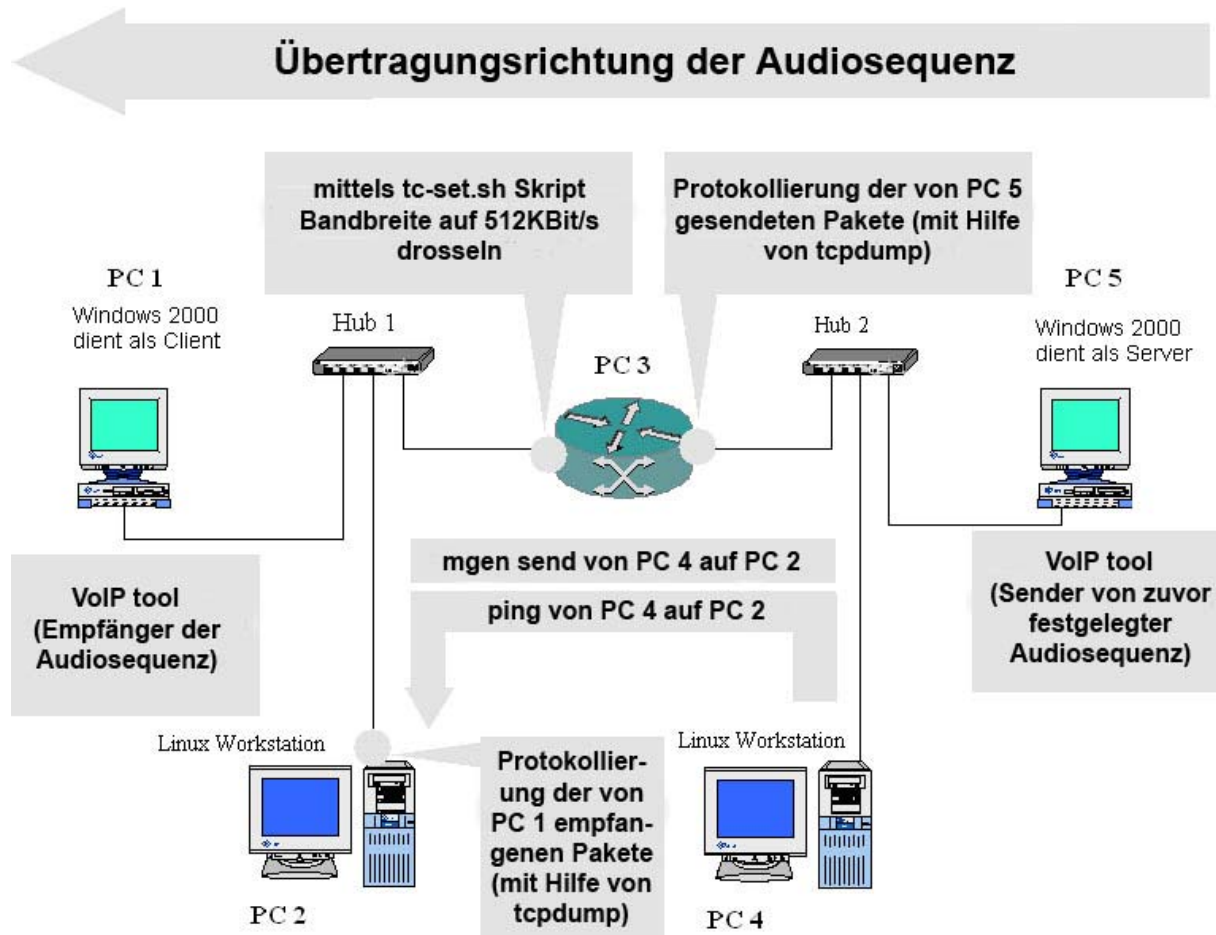


Abbildung 14: Testablauf und Datenflussrichtung

Vorerst müssen alle Tools und Anwendungen (wie in Abbildung 14 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 Skript 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 Skript Aufruf `mgen input <mgenSkript.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 Auswertung, sowie Visualisierung 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 Skriptfiles werden mit dieser Arbeit mitgeliefert Nähere Erläuterungen dazu sind unter dem Punkt Gnuplot – Skripte weiter oben angeführt.

Da PC 2 leider zwischenzeitlich anderwärtig gebraucht wurde, musste ein Teil der Testläufe ohne denselben durchgeführt werden. Dabei wurde sowohl das ping Signal als auch der mgen Traffic auf PC 1 umgeleitet und der Teststreckenaufbau stellte sich für die restlichen Versuchsabläufe wie folgt dar:

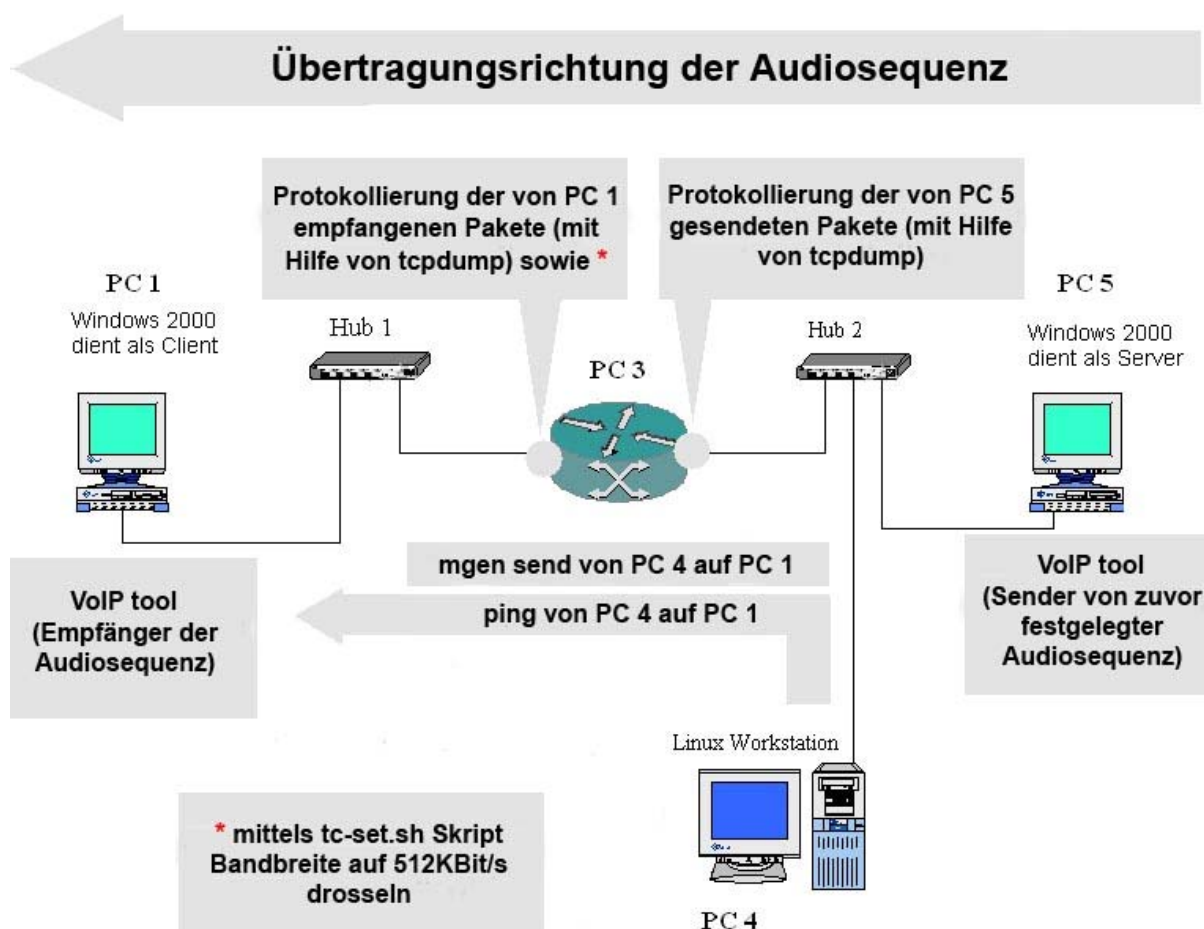


Abbildung 15: Veränderter Testablauf

Aus der geänderten Versuchsanordnung ergeben sich nur zwei Änderungen im zuvor beschriebenen Ablauf.

- Zum einen ändert sich Punkt 2, da beide tcpdump Protokollierungen nun auf PC 3 von statten laufen. Folglich muss bei Punkt 2 tcpdump auch auf PC 3 gestartet werden und zwar mit dem Aufruf `tcpdump -i eth1 -vvtttnn > tcpout.txt`.
- Weiters ändert sich Punkt 6, da ping nicht wie vorher auf PC 2 sondern auf PC 3 abgesetzt wird. Daraus resultiert die Änderung der IP im ping Befehl (`ping 192.168.0.3`).

6 Testergebnisse

Die graphische Ausarbeitung der gewonnenen Testergebnisse für alle Programme und Szenarien befindet sich, wie in [1], im Anhang dieses Dokuments.

6.1 MSN

6.1.1 Multi traffic steps 05 – 25

Dieses Szenario zeigt, dass der generierte Hintergrundverkehr, wie in Diagramm M.05-25.5 dargestellt, mit einem Höchstwert von 25 KByte/sec zwischen der 120. und 140. Sekunde des Tests, nicht genügt, um eine Erhöhung des Datendurchsatzes (Diagr. M.05-25.1) zu erreichen. Dieser bleibt während der gesamten Laufzeit konstant. Auch ergeben sich keine wesentlichen Veränderungen in der Paketgröße, siehe Diagramm M.05-25.2, sowie der Paketanzahl pro Sekunde (Diagr. M.05-25.4).

6.1.2 Multi traffic steps 30 – 50

Auch hier ist nach der ersten Testreihe keine wesentliche Veränderung im throughput zu erwarten. Was sich, wie aus Diagramm M.30-50.1 ersichtlich, bewahrheitet. Paketgröße und –anzahl bleiben weiterhin annähernd konstant (Diagr. M.30-50.2 & M.30-50.4). Die Round Trip Time aus Diagramm M.30-50.3 beginnt minimal zu fluktuieren. Allerdings bewegt sich dieselbe in einem Bereich von maximal 10 Millisekunden.

6.1.3 Multi traffic steps 55 – 75

In Diagramm M.55-70.1 erkennt man, dass ab der 100. Sekunde ein merkbarer Datenverlust auftritt. In diesem Zeitraum von der 100. bis zur 120. Sekunde wirkt der stärkste Hintergrundverkehr dieses Szenarios auf MSN ein (Diagr. M.55-70.5). Die Round Trip Time, aus Diagramm M.55-75.3, steigt dabei auch markant von zuerst unter 200 Millisekunden auf nunmehr 0,9 Sekunden an. Paketlänge und –anzahl (Diagr. M.55-75.2 & M.55-75.3) bleiben davon aber weiterhin unbeeinflusst.

6.1.4 Multi traffic steps 80 – 100

Ähnlich wie sich im vorigen Test die letzte Staustufe verhält, verhält sich nun MSN bei der ersten dieses Versuches. Ein merkbarer Datenverlust (Diagr. M.80-100.1) innerhalb der ersten 20 Sekunden des erzeugten Hintergrundverkehrs (Diagr. M.80-100.5) geht einher mit einer sich auf etwa eine Sekunde steigernde Round Trip Time (Diagr. M.80-100.3). Noch wichtiger ist aber die nächste Hintergrundverkehrserhöhung, welche ab der 40. Sekunde in Kraft tritt. Ab 85 KByte/sec generierter mgen – Staubelastung erhöht MSN seine durchschnittliche Paketgröße von durchschnittlich 200 auf ca. 360Byte, wie aus Diagramm Diagr. M.80-100.2 ersichtlich. Daraus resultiert zum einen eine Erhöhung der gesendeten Daten von ca. 8 KByte/sec, zum anderen scheint es, als gingen beim Empfänger ab diesem Zeitpunkt zumindest gleich viele Daten ein wie vor der Staubelastung. Zu sehen ist dieses Verhalten in Diagramm M.80-100.1.

6.1.5 Multi traffic steps 105 – 125

Genau wie beim vorigen Szenario steigt auch hier die Paketgröße, aus Diagramm M.105-125.2 ablesbar, auf denselben Wert von ca. 360 KByte/sec an. Dabei ist zu beachten, dass entgegen dem letzten Verhalten MSN aufgrund des weiter steigenden Hintergrundverkehrs (Diagr. M.105-125.5) nur noch sehr schwer einen Fall der ankommenden Daten unter die normalen 10 KByte/sec verhindern kann, siehe dazu Diagramm M.105-125.1.

6.1.6 Multi traffic steps 130 – 150

MSN beginnt immer mehr und mehr mit Datenverlusten zu kämpfen, gut zu sehen in Diagramm M.130-150.1, da der Hintergrundverkehr (Diagr. M.130-150.5) kontinuierlich in 5 KByte Schritten pro 20 sec weiter steigt.

6.1.7 Multi traffic steps 155 – 175

Hier zeigt sich in Diagramm M.155-175.3 gut, wie enorm die Round Trip Time aufgrund des ständigen Hintergrundverkehrs auf nunmehr 1,2 sec angestiegen ist. Der Paketgrößensprung von ca. 160Byte, aus Diagramm M.155-175.2, ist nach wie vor die einzige Erhöhung. Weder eine weitere Steigerung der Paketgröße, noch der Paketanzahl (Diagr. M.155-175.4) ist in diesem Testfall zu bemerken.

6.1.8 Multi traffic steps 180 – 200

Dieser Test zählt zu jenen beiden, welche die größte Belastung simulieren (Diagr. M.180-200.5). Dabei ist gut zu erkennen, dass MSN, bezüglich des empfangenen Datenverkehrs, teilweise nahe an der 0 KByte/sec Grenze vorbeischrämmt (Diagr. M.180-200.1). Jedoch bleibt es nach wie vor nur bei einer Erhöhung der Paketgröße, wie man Diagramm M.180-200.2 unschwer entnehmen kann.

6.1.9 Multi traffic steps 200 – 180

Auch hier erkennt man in Diagramm M.200-180.1 und M.200-180.2, dass MSN auf den enormen mgen – Verkehr von 200 KByte/sec reagiert. Allerdings ist bei diesem Szenario mit abfallendem Hintergrundverkehr (welches bewirkt, dass die höchste Verkehrsstufe als erstes auf das VoIP Tool einwirkt – siehe Diagramm M.200-180.5) bemerkbar, dass MSN um eine Spur schneller auf die Staubelastung reagiert als beim vergleichbaren „Multi traffic steps 180 – 200“ Test.

6.1.10 Multi traffic steps 175 – 155

Legt man Diagramm M.175-155.1 sowie M.155-175.1 übereinander, so erkennt man auch hier, dass MSN im Szenario mit sinkendem Hintergrundverkehr und draus resultierender, höherer Belastung bei Testbeginn, wiederum ein wenig schneller auf den Hintergrundverkehr reagiert, als das dies beim 155-177er Szenario der Fall ist.

6.1.11 Multi traffic steps 150 – 130

Vergleicht man Diagramm M.150-130.1 mit M.130-150.1 und M.150-130.2 mit M.130-150.2, so erkennt man bisher an diesem Beispiel am anschaulichsten, dass der zu Beginn höhere Hintergrundverkehr zu einem früheren Sprung in der Paketgröße führt. Dabei dürfte es sich bei dem oben angesprochen Vergleich um einen Unterschied von etwa 5 Sekunden handeln.

6.1.12 Multi traffic steps 125 – 105

Bei diesem Szenario scheinen die Veränderungen in throughput (Diagr. 125-105.1) und Paketgröße (Diagr. 125-105.2) annähernd gleich wie in seinem Pedant mit zunehmendem Hintergrundverkehr.

6.1.13 Multi traffic steps 100 – 80

Aus Diagramm M.100-80.2 ist ein steil ansteigender Sprung in der Paketgröße bei einem Hintergrundverkehr (Diagr. M.100-80.5) von 100 KByte/sec ersichtlich. Wichtiger ist jedoch die Umkehrung dieses Vorganges nach der 135. Sekunde. Dabei geht MSN wieder auf die Ausgangspaketgröße von 200Byte zurück. Lohnend ist noch ein Blick auf den mgen – Verkehr (Diagram. M.100-80.5). Während in Diagramm M.80-100.1 bei einem mgen – Verkehr von 80 KByte/sec, also während der 20. und 40. Sekunde, MSN erst am Ende dieser Hintergrundverkehrsstufe eine Erhöhung im throughput (Diagr. M.80-100.1) bewirkt, bleibt das Tool während des entsprechenden Zeitraumes im Szenario mit abnehmenden Hintergrundverkehr (100. bis 120. Sekunde) durchgehend auf dem Niveau mit erhöhter Paketgröße von 360Byte. Sogar über den Zeitraum des mgen – Verkehrs hinaus, also ab der 120. Sekunde bis hin zur 135., bleibt das VoIP Tool auf diesem erhöhten Niveau, wie man Diagramm M.100-80.2 entnehmen kann.

6.1.14 Multi traffic steps 75 – 55

Bei diesem Test ist in den beiden Diagrammen M.75-55.1 sowie M.75-55.5 schön zu beobachten, wie die in der ersten Stauphase verloren gegangenen Daten zwar während der nächst niedrigeren mgen – Verkehrsstufe von 70 KByte/sec noch nicht, dann aber mit dem Beginn der 65 KByte/sec Stufe nachgereicht werden können. Der maximale Hintergrundverkehr von 75 KByte/sec reicht nicht aus um eine Erhöhung der Paketgröße, in Diagramm M.75-55.2 zu sehen, zu bewirken. Die Round Trip Time aus Diagramm M.75-55.3 nimmt in diesem Szenario mit Beginn der ersten mgen – Verkehrsstufe linear zu und hält sich anschließend bis zum Ende der 70 KByte/sec Stufe annähernd auf demselben Niveau, bevor sie wieder linear zu fallen beginnt.

6.1.15 Multi traffic steps 50 – 30

Dieses Szenario zeigt einen annähernd linearen Anstieg der Round Trip Time ab der ca. 75. Sekunde. Zu sehen ist dies in Diagramm M.50-30.3. Jedoch bewegt man sich dabei noch in einem Bereich von unter 15 Millisekunden.

6.1.16 Multi traffic steps 25 – 05

Sehr ähnlich dem Testfall 05 – 25 zeigt sich auch dieser hier. Der Hintergrundverkehr (Diagr. M.25-05.5) reicht bei weitem nicht aus, um eine Veränderung im Sendeverhalten von MSN zu bewirken, siehe dazu Diagramm M.25-05.1, M.25-05.2 und M.25-05.4. Kleine Fluktuationen in der Round Trip Time (Diagr. M.25-05.3) scheinen von geringem Interesse, da sich diese in einem Bereich von max. 10 Millisekunden bewegen.

6.2 Skype

6.2.1 Single pause 10 sec

Dieser Test entspricht demselben der schon in der ersten Arbeit durchgeführt wurde. Die beiden mgen – Paket aus Diagramm S.sp10.5 markieren die entscheidenden 10 Sekunden, in denen die Pause im übertragenen Audiofile, wie zuvor weiter oben beschrieben, zum Tragen kommt. Wie bereits in der ersten Arbeit ist ein geringfügiger Abfall in der Paketlänge (Diagr. S.sp10.2), sowie folglich auch im Datendurchsatz (Diagr. S.sp10.1) zu bemerken. Allerdings ist immer noch fraglich, ob ein so geringer Abfall wirklich von der Übertragungspause herrührt.

6.2.2 Single pause 20 sec

Wie im vorherigen Fall markieren die beiden mgen – Pakete, nun in der 50. und 70. Sekunde, den Start und das Ende der Übertragungspause, siehe dazu Diagramm S.sp20.5. Paketgröße (Diagr. S.sp20.2), sowie throughput (Diagr. S.sp20.1) verharren nun eine Spur länger auf dem gleichen Niveau, auf das Skype schon im „Single pause 10 sec“ Test gefallen war.

6.2.3 Single pause 30 sec

Bei einer Pausenlänge von 30 Sekunden pendeln sich nicht nur wie eben noch Durchsatz (Diagr. S.sp30.1) und Paketgröße auf das gleiche Niveau ein, sondern des weiteren kommt in der 60. Sekunde ein kurzzeitiger Abfall der Paketgröße um etwa 20Byte hinzu. Diese Minderung der Paketgröße geht einher mit einer Steigerung der Paketanzahl, wie man Diagramm S.sp30.2 entnehmen kann. Aus der Senkung der Paketgröße und der Erhöhung der Paketanzahl resultiert ein kurzzeitiger Datenverlust (Diagr. S.sp30.1, 6.4 Datentransfer Tabelle) für den besagten Zeitraum.

6.2.4 Single pause 40 sec

Aus Diagramm S.sp40.1 ersichtlich, pendelt sich der Datendurchsatz hier im entscheidenden Zeitraum auf etwa 4,2 KByte/sec und die Paketgröße (Diagr. S.sp40.2) auf etwa 125Byte ein. Eine ruckartige Änderung in der Paketgröße und eine damit verbundene Veränderung in der Paketanzahl (Diagr. S.sp40.4) ist nicht festzustellen.

6.2.5 Single pause 50 sec

Gleich wie bei allen Testfällen zuvor reagiert Skype auf die Pause in der Übertragung mit einer Senkung der Paketgröße und einer daraus resultierenden Minderung des throughput (Diagr. S.sp.50.2 u. S.sp.50.1). Allerdings tritt hier, gleich wie zuvor im „Single pause 30 sec“ Szenario, eine ruckartige Verminderung der Paketgröße, eine ebenso schnelle wie kurzzeitige Erhöhung der Paketanzahl (Diagr. S.sp.50.4) und ein daraus resultierender Datenverlust von etwa 0,54 KByte auf (6.4 Datentransfer Tabelle).

6.2.6 Single pause 60 sec

60 Sekunden Pause in der Übertragung haben annähernd 55 Sekunden gleich bleibende Paketgröße und –anzahl zur Folge, wie man den Diagrammen S.sp.60.1 sowie S.sp.60.2 entnehmen kann.

6.2.7 Repeated pause 10 sec

Bei diesem ersten Test mit mehrmaliger Pause bemerkt man sofort die variierende Testlänge, welche in diesem Fall bei 70 Sekunden liegt. Während der ersten Pause von der 10. bis zur 20. Sekunde zeigt Skype im throughput – Verlauf (Diagr. S.sp10.1) relativ wenig Reaktion im Vergleich zu den ersten 10 Testsekunden. Nach der ersten Pause steigt der Datendurchsatz jedoch an und erreicht kurz vor einem abermaligen Abfall, der von der zweiten Übertragungspause herrühren dürfte, ein lokales Maximum. Von der 40. Sekunde bis hin zum Szenarioabschluss wird der steigende und fallende throughput, der mit den Pausen in der Übertragung einhergeht, noch deutlicher.

6.2.8 Repeated pause 20 sec

Aus den Diagrammen S.rp20.1, S.rp20.2 und S.rp20.3 lässt sich unschwer erkennen, dass Veränderungen im throughput rein von der variierenden Paketgröße herrühren. Auch erkennt man, dass das bereits im vorherigen Test festgestellte Verhalten in Bezug auf Übertragungspausen verstärkt wird.

6.2.9 Repeated pause 30 sec

Diagramm S.rp30.1 sowie S.rp30.2 zeigen jeweils, dass Skype in dem Moment in dem es auf die Übertragungspause reagiert, seine Paketgröße ruckartig um bis zu 40Byte senkt. Dabei bleibt die Anzahl der Pakete unverändert (Diagr. S.rp30.4).

6.2.10 Repeated pause 40 sec

Die Darstellungen ab diesem Test bis hin zum Ende dieser Testreihe wirken sehr gedrängt, was durch die lange Testdauer und die gleich bleibenden Fenstergrößen der Diagramme resultiert.

Das aus den vorigen Szenarien beschriebene Verhalten von Skype setzt sich hier, mit einem kleinen Unterschied, fort. Dieser Unterschied ist in Diagramm S.rp40.1, bzw. S.rp40.2 zu erkennen. Das VoIP Tool beginnt sich in den 40 Sekundenbereichen, in denen ein normaler Geräuschpegel übertragen wird, auf ein normales Niveau einzupendeln. Sprich es ist kein ständiger Anstieg mehr bis zur nächsten Pause bemerkbar (siehe hierzu z.B. Diagr. S.rp40.2 – 180. Sekunde).

6.2.11 Repeated pause 50 sec

Diagramm S.rp50.2 bestätigt die Aussage zum vorigen Szenario. Während der Pausen, die nunmehr 50 Sekunden dauern, pendelt sich Skype, wie allen Tests gemein, auf ein Level von etwa 125 – 130Byte bei der Paketgröße ein. Die Größe der Pakete hingegen schwankt während der Zeiträume der normalen Übertragung extremer denn je zuvor (siehe Diagr. S.rp50.1). Die Paketanzahl aus Diagramm S.rp50.4 bleibt wie bisher schon beobachtet konstant.

6.2.12 Repeated pause 60 sec

Dieser Test stellt mit einer Laufzeit von 420 Sekunden das längste Szenario dar. An sich bleibt das Verhalten von throughput (Diagr. S.rp60.1) und Paketgröße (Diagr. S.rp60.2) dasselbe, wie schon zuvor beobachtet. Nur ein kleiner Unterschied in der 236. Sekunde macht sich bemerkbar. Hier sinkt, wie in zwei Testfällen mit einfacher Pause bereits beobachtet, die Paketgröße plötzlich ruckartig ab, obwohl sie sich bereits auf dem niederen Niveau von ca. 125Byte befände. Gleichzeitig springt die Paketanzahl (Diagr. S.rp60.4) von bisher ca. 34 Paketen auf nunmehr über 40. Wiederum bewirkt dieses Verhalten einen geringen Datenverlust – ersichtlich aus Diagramm S.rp.60.1 sowie aus der Datentransfer Tabelle auf Seite 34.

6.3 Zusammenfassung der gewonnenen Ergebnisse

6.3.1 Zusammenfassung MSN

Wie in Kapitel 3 (im Speziellen Kapitel 3.2.2) geschildert, scheint es sich bei der throughput – Erhöhung von MSN um eine Art der Vorwärtsfehlerkorrektur zu handeln. Hierbei versucht das VoIP Tool drohenden Paketverlust durch eine zusätzliche Übertragung von „Stützdaten“ zu kompensieren.

Die zu Beginn dieser Arbeit noch offenen Fragen können nun wie folgt beantwortet werden:

- Sind die Erhöhung der Paketgröße und der damit verbundene Sprung im Datendurchsatz einmalig, oder wiederholt er sich bei weiterer Steigerung des Hintergrundverkehrs?

Die Erhöhung der Paketgröße auf fast das Doppelte des Ausgangswertes ist einmalig und kann durch Steigerung des Hintergrundverkehrs nicht wiederholt werden.

- Wird der Sprung im Datendurchsatz immer durch eine Erhöhung der Paketgröße ausgelöst, oder beginnt auch die bisher annähernd konstante Paketanzahl zu variieren?

Die Anzahl der Pakete bleibt unter allen Testfällen konstant, daher wird eine Änderung des Datendurchsatzes immer durch eine Schwankung in der Paketgröße bewirkt.

- Beeinflusst ein zu- oder abnehmender Hintergrundverkehr das bisher beobachtete Verhalten in irgendeiner Weise?

Der Sprung in der Paketgröße wird sowohl durch ein Zunehmen als auch durch ein Abnehmen des Hintergrundverkehrs bewirkt. Kleinere Unterschiede zwischen zu- und abnehmenden Verkehrsszenarien sind in den jeweiligen Detailerläuterungen für die einzelnen Szenarien selbst beschrieben.

6.3.2 Zusammenfassung Skype

Skype reagiert, wie aus beinahe allen Szenarien mit einer Pausenlänge über 20 sec ersichtlich, auf Übertragungspausen. Dabei werden sowohl die Paketgröße, als auch der resultierende Datendurchsatz während der Pause beinahe konstant auf einem relativ geringen Niveau gehalten. Je länger die Pausen und damit auch die Zeit zwischen zwei Pausen wird (für Testfälle mit mehreren Pausen pro Szenario), desto mehr beginnt Skype in der Zeit der eigentlichen Übertragung zwischen hoher und niedriger Datenrate zu schwanken.

Erwähnenswert ist Skypes Verhalten, welches sich in den Testfällen „single pause 30 sec“, „single pause 50 sec“ und „repeated pause 60 sec“ zeigte. In diesen Szenarien ist ein kurzzeitiger Rückgang der Paketgröße und ein Anstieg der Paketanzahl zu beobachten. Aus diesem Verhalten resultierte in all diesen Beispielen ein kurzzeitiger Datenverlust.

6.4 Datentransfer Tabelle

In der unten abgebildeten Transfertabelle sind sowohl die MSN als auch 3 Skype Szenarien dargestellt. Diese drei Szenarien von Skype entsprechen genau jenen, bei welchen eine kurzzeitige Verminderung der Paketgröße und eine damit einhergehende Erhöhung der Paketanzahl zu bemerken waren (siehe Anhang Diagramme S.sp30.4, S.sp50.4 sowie S.rp60.4).

Anwendung	gesendet [KByte]	empfangen [KByte]	Verlust [KByte]	Anteil [%]
MSN				
Multi traffic steps 05-25	1436,16	1404,71	31,45	2,19
Multi traffic steps 30-50	1420,26	1403,34	16,92	1,19
Multi traffic steps 55-75	1414,01	1386,90	27,11	1,92
Multi traffic steps 80-100	2247,40	1753,28	494,12	21,99
Multi traffic steps 105-125	2380,04	1549,84	830,20	34,88
Multi traffic steps 130-150	2320,86	1339,01	981,85	42,31
Multi traffic steps 155-175	2355,02	1231,41	1123,61	47,71
Multi traffic steps 180-200	2341,98	1140,71	<u>1201,27</u>	<u>51,29</u>
			<i>gesamt</i>	<i>203,48</i>
MSN				
Multi traffic steps 200-180	2419,35	1123,39	1295,96	53,57
Multi traffic steps 175-155	2357,45	1212,64	1144,81	48,56
Multi traffic steps 150-130	2427,32	1412,60	1014,72	41,80
Multi traffic steps 125-105	2319,68	1435,28	884,40	38,13
Multi traffic steps 100-80	2284,45	1811,24	473,21	20,71
Multi traffic steps 75-55	1406,07	1381,17	24,90	1,77
Multi traffic steps 50-30	1483,45	1456,82	26,63	1,80
Multi traffic steps 25-05	1419,06	1400,81	<u>18,25</u>	<u>1,29</u>
			<i>gesamt</i>	<i>207,62</i>
Skype				
Single pause 30sec	558,80	558,26	0,54	0,10
Single pause 50sec	546,51	545,97	0,54	0,10
Repeated pauses 60sec	1912,79	1909,84	2,95	0,15

*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

7 Schlusswort

7.1 Zusammenfassung der Arbeit

War in der ersten Arbeit noch ein Herantasten an die einzelnen VoIP Tools von Nöten, um die Grundeigenschaften der Programme etwas näher kennen zu lernen, so blieb einem dies aufgrund des bereits gewonnen Wissens in dieser Testreihe erspart. Jedoch brachten die Änderung des Teststreckenaufbaues, sowie einige Eigenheiten von mgen (welches bei sehr großen Skripten seinen Dienst verweigerte) Verzögerungen mit sich.

7.1.1 MSN

In [1] zeigte MSN als einziges Tool eine Reaktion auf den mittels mgen generierten Hintergrundverkehr. Dabei stieg die Paketgröße in den Szenarien „Middletraffic“ sowie „Hightraffic“ sprunghaft an, was auf eine Art der Vorwärtsfehlerkorrektur schließen lässt. Aus diesem Grund wurde in dieser Arbeit das Grundprinzip von Vorwärtsfehlerkorrekturverfahren in Kapitel 3 bzw. 3.2.2 erläutert. Weiters wurde das Verhalten von MSN genauer analysiert. Dazu generierte man verschiedenste Szenarien, in welchen durch mgen generierter Hintergrundverkehr stufenweise erhöht bzw. vermindert wurde. Diese erneute Analyse ergab, dass MSNs Sprung in der Paketgröße einmalig ist und durch weitere Erhöhung des Hintergrundverkehrs nicht wiederholt werden kann. Weiters blieb, wie in der ersten Arbeit schon beobachtet, die Paketanzahl in allen Testfällen auf einem konstanten Niveau.

7.1.2 Skype

Skype zeigte in [1] keine Reaktion auf Hintergrundverkehr. Überraschend war zudem, dass das Tool, auf Pausen in der Übertragung, nicht wie alle anderen VoIP Anwendungen mit einer Verminderung des Datendurchsatzes reagierte bzw. nur in so geringem Ausmaße, dass nicht eindeutig festzustellen war, ob der Abfall im Datendurchsatz von der Übertragungspause oder anderen Faktoren abhängt. Deshalb wurden in dieser Arbeit weitere Szenarien erzeugt aus welchen dies klar hervorgehen sollte. Die Analyse dieser Testfälle ergab, dass Skype wie alle anderen zuvor getesteten Anwendungen auch auf Pausen in der Übertragung reagiert, indem es den Datendurchsatz vermindert. Erzielt wird diese Verminderung durch ein Variieren der Paketgröße. Merkwürdig war allerdings, dass in manchen Szenarien eine weitere Verminderung der Paketgröße, in eben genau jener Zeit, in welcher sich die Paketgröße schon auf einem geringen Niveau befand, zu einer ebenso kurzen Erhöhung der Paketanzahl führte. Aus diesem Verhalten resultierte schlussendlich ein kurzzeitiger Datenverlust (siehe 6.4 Datentransfer Tabelle).

7.2 Danksagung

- Mein Dank gilt Herrn Dr.-Ing. Michael Welzl für seine tatkräftige Unterstützung und die Betreuung dieser Arbeit und vor allem für die Möglichkeit mich auf diesem Interessensgebiet weiter einarbeiten zu können.
- 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.

7.3 Literaturverzeichnis

- [1] Meine erste Arbeit zu diesem Thema - Analyse des Netzwerkverhaltens von VoIP
Thomas Rammer
<http://www.welzl.at/teaching/index.html>
- [2] 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/>
- [3] Enzyklopädie verwendete Erläuterung von Fehlerkorrektur
<http://de.wikipedia.org/wiki/Hauptseite>
- [4] Teststreckenaufbau und Vergleichswerte
Akdag Muhlis
<http://www.welzl.at/teaching/baks/akdagmuhlis/BakArbeitAkdagMuhlis.pdf>
- [5] Codewörter und Vorwärtsfehlerkorrektur
Felix Holderied
<http://www.wipr.ira.uka.de/~lehre/RS/uebungen/ss97/zuverlaessigkeit.loesung.html>
- [6] Fehlerkorrektur
<http://www.christoph-haas.de/uni/ta-lernskript-daten%FCbertragung.pdf>