



**Leopold – Franzens - Universität
Innsbruck**

Institut für Informatik

Analyse des Netzwerkverkehrs von Computerspielen

Bakkalaureatsarbeit

eingereicht bei Dr. Ing. Michael Welzl

Autor: Florian Larch und Markus Pabst

Innsbruck 28.03.2007

Inhaltsverzeichnis

1. Abkürzungsverzeichnis	4
2. Allgemeines	5
2.1. Einleitung	5
2.2. Begriffserklärung	6
2.2.1. Client-Server Technologie	6
2.2.2. Game-Engine	7
2.2.3. Netcode	7
3. Aufgabenstellung	8
4. Netzwerkprotokolle	9
4.1. IP	9
4.2. UDP	10
5. Versuchsdurchführung	11
5.1. Allgemeines zur Teststrecke	11
5.2. Aufbau der Teststrecke	12
5.3. Ablauf des Testverfahrens	12
5.4. Testszenarien	13
5.5. Spezifikation der Teststrecke	14
5.5.1. Hardwarespezifikation	14
5.5.2. Softwarespezifikation	15
5.6. Tools für die Durchführung der Tests	16
5.6.1. Netzwerkspezifische Tools	16
5.6.2. Weitere Tools	18
6. Anwendungen	21
6.1. Strategiespiele	21
6.1.1. Age of Empire	21
6.1.2. Anno 1701	23
6.1.3. Down of War	24
6.2. Ego – Shooter	26
6.2.1. Quake 4	26
6.2.2. Battlefield 2	28
6.3. Sportspiele	29
6.3.1. Fifa07	29

7. Skripte und Programme	31
7.1. tc-Skripte	31
7.2. mgen-Skripte	32
7.3. Java Programme	34
7.3.1. Stapelverarbeitungsdatei	34
8. Testergebnisse	35
8.1. Age of Empire	35
8.1.1. Kein Hintergrundverkehr	35
8.1.2. Wenig Hintergrundverkehr	35
8.1.3. Mittlerer Hintergrundverkehr	36
8.1.4. Viel Hintergrundverkehr	36
8.2. Anno 1701	36
8.2.1. Kein Hintergrundverkehr	36
8.2.2. Wenig Hintergrundverkehr	37
8.2.3. Mittlerer Hintergrundverkehr	37
8.2.4. Viel Hintergrundverkehr	37
8.3. Dawn of War	38
8.3.1. Kein Hintergrundverkehr	38
8.3.2. Wenig Hintergrundverkehr	38
8.3.3. Mittlerer Hintergrundverkehr	38
8.3.4. Viel Hintergrundverkehr	39
8.4. Quake 4	39
8.4.1. Kein Hintergrundverkehr	39
8.4.2. Wenig Hintergrundverkehr	40
8.4.3. Mittlerer Hintergrundverkehr	40
8.4.4. Viel Hintergrundverkehr	40
8.5. Battlefield 2	41
8.5.1. Kein Hintergrundverkehr	41
8.5.2. Wenig Hintergrundverkehr	41
8.5.3. Mittlerer Hintergrundverkehr	41
8.5.4. Viel Hintergrundverkehr	42
8.6. Fifa 07	43
8.6.1. Kein Hintergrundverkehr	43

8.6.2. Wenig Hintergrundverkehr	43
8.6.3. Mittlerer Hintergrundverkehr	44
8.6.4. Viel Hintergrundverkehr	44
9. Zusammenfassungen	45
9.1. Zusammenfassung Age of Empire	45
9.2. Zusammenfassung Anno 1701	45
9.3. Zusammenfassung Dawn of War	45
9.4. Zusammenfassung Quake 4	46
9.5. Zusammenfassung Battlefield 2	46
9.6. Zusammenfassung Fifa 07	46
Schlusswort	47
Quellenverzeichnis	49
Abbildungsverzeichnis	50
Danksagung	50

1. Abkürzungsverzeichnis

Abkürzung	Bedeutung
ARP	Address Resolution Protocol
DNS	Domain Name System
DHCP	Dymanic Host Configuration Protocol
IP	Internet Protocol
LAN	Local Area Network
MAC	Medium Access Control
TCP	Transmission Control Protocol
OSI-ISO	Open Systems Interconnection Reference Model
UDP	User Data Protocol
Trpr	Tcpdump Rate Plot Real-time

2. Allgemeines

2.1. Einleitung

Computerspiele entwickelten sich in den 50er Jahren aus technischen Versuchen an Universitäten. Bereits auf den ersten Computern gab es Versuche, bekannte Brettspiele umzusetzen. Als erstes modernes Computerspiel wird meist das von William Higinbotham entwickelte „Tennis for two“ angesehen. Durch die rasante technische Weiterentwicklung wurden Computerspiele immer unterhaltsamer und vor allem für die breite Masse zugänglich. Schon bald wurden Mehrbenutzerspiele entwickelt, die via Internet bzw. LAN (Local Area Network) miteinander kommunizierten. Diese Spiele ermöglichen also ein Spielen mit realen Gegenspielern, unabhängig von Wohnort, Sprache und Alter.

Mehrbenutzerspiele, die über das Internet mit anderen realen Mitspielern kommunizieren, basieren auf der Client-Server Technologie. Der Server kann dabei mit einer entsprechenden Internetverbindung je nach Spiel, 8 bis 32 Clients bedienen.

Das Spielen dieser Art von Computerspiele ist zurzeit eine der beliebtesten Freizeitaktivitäten. Durch die große Anzahl an Clients und faszinierenden Spielwelten steigt auch stetig der notwendige Datenaustausch zwischen Server und Client. Bei zu hoher Last kann es auch zu Stau zwischen Server und Client kommen. Der Stau könnte sich daher negativ auf dem Spielfluss auswirken. Viele Mehrbenutzerspiele haben daher eine Staukontrolle, damit die Kommunikation zwischen Server und Client aufrechterhalten bleibt und den Spielfluss nicht beeinträchtigt.

Da stellt sich die Frage, welche Protokolle und welche Strategien verwendet werden müssen, um eine sichere Kommunikation zu gewährleisten.

2.2. Begriffserklärung

Im folgenden Kapitel wird die vorherrschende Technologie bei Computerspielen genauer erklärt.

2.2.1. Client-Server Technologie

Die analysierten Computerspiele basieren auf einer Client-Server Architektur.

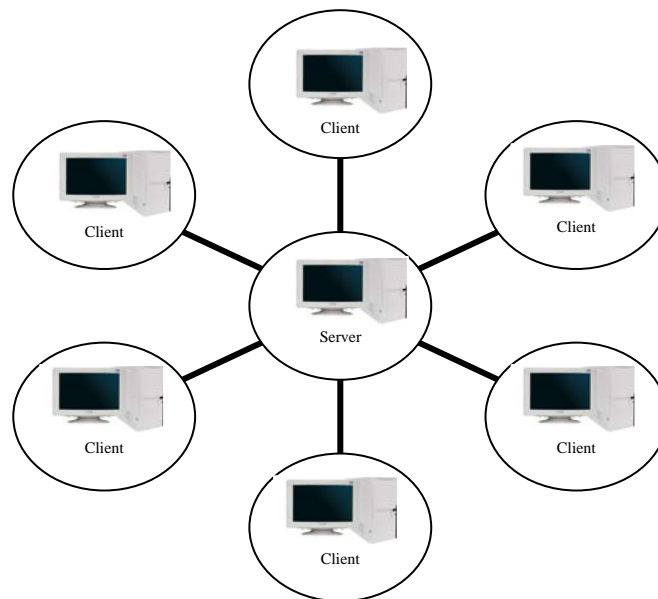


Abbildung 1: Client-Server Architektur

Der wegen seiner Struktur auch „Stern-Architektur“ genannte Aufbau, siehe Abbildung 1, enthält einen Server, welcher mit mehreren Clients verbunden ist. Der Server speichert sämtliche Nachrichten, die er von den angehängten Clients bekommt. Der Server verarbeitet diese Daten und schickt sie an die Clients zurück. Der Grund für diese Kommunikation ist das Synchronisieren der Spielstände auf allen Clients.

Der Vorteil des Servers ist, dass er als reine Rechenanlage keine grafischen Darstellungen des Spielgeschehens benötigt und daher viel Rechenzeit spart. Dagegen haben die Clients einen hohen Rechenaufwand, den sie zum Rendern der verschiedenen Szenarien brauchen.

Allerdings hat die Stern-Architektur auch Nachteile. Sind bei einem Server zu viele Clients eingeloggt, so kann es zu Verzögerungen oder sogar zur Trennung der Verbindung kommen (Time Out). Ein synchrones Ablaufen der Spiele kann auch nicht garantiert werden, da die Verbindungen vom Server zu den einzelnen Clients unterschiedlich gut ausgelegt sind (unterschiedliche Hardware der Clients & unterschiedliche Qualität der Datenübertragung).

Trotz allem gehört die Stern-Architektur derzeit zu den effizientesten Lösungen für das vernetzte Spielen und wird daher von vielen Spielherstellern verwendet.

2.2.2 Game-Engine

Als Game-Engine wird der eigenständige Teil des Computerspiels bezeichnet. Dieser Teil beinhaltet alle nötigen Routinen, Objekte und Bewegungen des Spiels zu berechnen und auf dem Bildschirm darzustellen. Eine Game-Engine besteht aus einer Programmbibliothek, die den Entwicklern von Computerspielen, die meist benutzten Werkzeuge zur Verfügung stellt.

2.2.3 Netcode

Der Netcode ist ein Teil der Game Engine, der die Aufgabe besitzt, sämtliche Spielinformationen im Netzwerk zu koordinieren. Es werden Objekte oder auch Bewegungen und andere n Spielinformationen eines Netzwerkspiels in Informationspakete zusammengefasst. Diese werden mittels Netcode vom Client zum Server oder vom Server notwendige um Client geschickt. Somit ist ein Zusammenspiel mehrere Clients möglich. Außerdem bestimmt der Netcode die Voraussetzungen für den Multiplayerteil eines Spiels, z. B. wie viele Spieler gleichzeitig am Spiel teilnehmen können, oder ob ein 56k-Modem ausreichend ist, oder ob DSL benötigt wird.

3. Aufgabenstellung

Ziel dieser Arbeit war die Untersuchung des Netzwerkverhaltens von Computerspielen mit zunehmendem Stau (Stau: Verzögerung und oder Verlust von Paketen). Dazu wurde ein kleines Netzwerk mit fünf Computern verwendet. Ein Computer soll ein Linux - Router sein, welcher das Netz in zwei Netze aufteilt sprich zwei Computer sind in einem IP-Netz und die anderen zwei Computer sind in einem anderen IP-Netz. Auf jeweils einen Computer pro Netz wird dann ein Computernetzwerkspiel gestartet. Mit den anderen zwei Computern wird der auftretende Netzwerkverkehr aufgezeichnet. Am Router wird eine Bandbreiteneinschränkung erzeugt, damit man sehen kann, wie das Spiel auf Stau im Netzwerk reagiert.

Mit dieser Arbeit wurden folgende Punkte abgedeckt:

- Vermittlung der Netzwerkgrundkenntnisse
- Vermittlung einiger Grundkenntnisse der beiden Betriebssysteme, Windows und Linux
- Installation von allen notwendigen Tools und Computerspielen
- Erstellen von normierten Testfällen
- Aufzeichnen des auftretenden Netzwerkverkehrs
- Analyse des auftretenden Netzwerkverkehrs mit graphischer Darstellung
- Interpretation der Testergebnisse

Folgenden Fragen wurden in dieser Arbeit beantwortet:

- Welche Protokolle verwenden Computer für die Kommunikation?
- Wie reagieren Computerspiele auf Stau bzw. Engpässen bei der Datenübertragung?
- Welche Strategie verwenden Computerspiele um die Kommunikation zu gewährleisten?
- Welche Bandbreite verwenden Computerspiele, für die Kommunikation?
- Welche Paketgrößen werden für die Datenübertragung benutzt?

4. Netzwerkprotokolle

Computerspiele verwenden für die Kommunikation die Protokolle IP und UDP.

Das unten angeführte ARP Protokoll ist ein Netzwerkgrundlegendes Protokoll, welches hier erklärt wird, um das Netzwerkverständnis zu erhöhen.

UDP (User Datagram Protocol)	
Familie:	Internetprotokollfamilie
Einsatzgebiet:	Verbindungslose Übertragung von Daten über das Internet
UDP im TCP/IP-Protokollstapel	
<i>Anwendung</i>	DNS DHCP NTP ...
Transport	UDP TCP
<i>Internet</i>	IP
<i>Netzzugang</i>	Ethernet Token Ring FDDI ...
Standards:	RFC 768 (1980)

Abbildung 2: UDP (User Datagram Protocol)

4.1. IP

Das Internet Protokoll (IP) ist ein, in Computernetzen, weit verbreitetes Netzwerkprotokoll. Es ist die Implementierung der Internet-Schicht des TCP/IP-Modells bzw. der Vermittlungsschicht (Network Layer) des OSI-Modells.

IP bildet die vom Übertragungsmedium unabhängige Schicht der Internetprotokollfamilie. Das bedeutet, dass mittels IP-Adresse und Subnetzmaske 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. Das Internet Protokoll stellt die Grundlage des Internets dar.

4.2. UDP

Das User Datagram Protokoll (Abk. UDP) ist ein minimales, verbindungsloses Netzprotokoll, das zur Transportschicht der Internetprotokollfamilie gehört. Aufgabe von UDP ist es, Daten, die über das Internet übertragen werden, die richtigen Anwendung zukommen zu lassen.

Die Entwicklung von UDP begann 1977, als man für die Übertragung von Sprache ein einfacheres Protokoll benötigte, als das bisherige verbindungsorientierte TCP. Es wurde ein Protokoll benötigt, das nur für die Adressierung zuständig war, ohne die Datenübertragung zu sichern, da dies zu Verzögerungen bei der Sprachübertragung führen würde.

Funktionsweise:

Um die Daten, die mit UDP versendet werden, dem richtigen Programm auf dem Zielrechner zukommen zu lassen, werden bei UDP so genannte Ports verwendet. Dazu wird bei UDP die Portnummer des Dienstes mit gesendet, der die Daten erhalten soll. Diese Erweiterung der Host-zu-Host- auf eine Prozess-zu-Prozess- Übertragung wird als Anwendungsmultiplexen und - demultiplexen bezeichnet.

Zusätzlich bietet UDP die Möglichkeit einer Integritätsüberprüfung an, indem eine Prüfsumme mit gesendet wird. Dadurch kann eine fehlerhafte Übertragung erkannt werden.

5. Versuchsdurchführung

5.1. Allgemeines zu Teststrecke

Die Teststrecke bestehend aus fünf Rechnern, sie ist wie folgt zusammengesetzt:

- 2 Windowsrechner (PC1, PC5 bzw. PC6)
- 2 Linuxrechner (PC2, PC4)
- 1 Router(PC 3)

Im Versuchsaufbau wird ein Windowsrechner (PC5 oder PC6) als Server eingesetzt. Dieser ist über einem Hub mit einen Linuxrechner, der als Messstation eingesetzt ist, und dem Router verbunden. Der Windowsrechner (PC5 oder PC6) simuliert somit einen Spielserver im Internet, auf den zusammen mit dem 2.Windowsrechner (PC1) die Anwendungen gestartet und gespielt wurden. Weiters ist hinzufügen, dass auch der 2. Windowsrechner (PC1) über einen Hub mit den Router und mit einem Linuxrechner (PC2) verbunden ist (Siehe Abbildung 3).

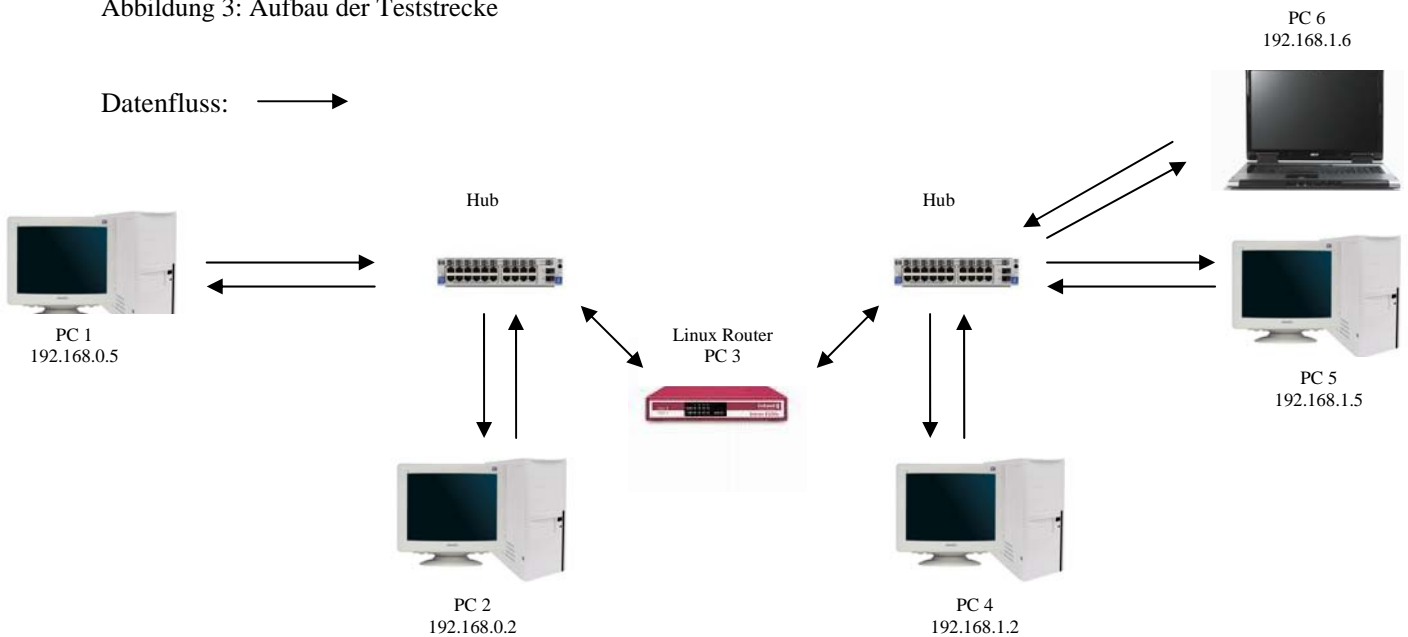
Auf den Windowsrechnern (PC1, PC5 bzw. PC6) werden die Anwendungen gestartet und ausgeführt. Diese Anwendungen werden mit Hilfe der 2 Linuxrechner (PC2, PC4) analysiert und ausgewertet. Die Linuxrechner (PC2, PC4) messen den Datenfluss zwischen Server und Client. Der Linuxrouter (PC3) leitet die Daten weiter, die protokolliert werden. Die Aufgabe des Routers liegt darin, den eingehenden und abgehenden Verkehr der jeweiligen Netzwerkkarten zu beschränken. Er wird als Bottleneck verwendet, d.h. der Datenfluss hat hier einen Engpass zu überwinden.

Einer der Linuxrechner (PC4) simuliert erhöhtes Datenaufkommen, sodass verschiedene Szenarien getestet werden können.

Der Datenfluss zwischen Server und Client wird durch die Linuxrechner protokolliert. Ab dem Switch wird dem Datenfluss in Richtung Client zusätzlicher UPD-Verkehr hinzugefügt. Dabei entsteht mit dem erhöhten Verkehr und dem Bottleneck eine Stausituation. Vom Client zum Server wird kein zusätzlicher UDP-Verkehr hinzugefügt.

5.2. Aufbau der Teststrecke

Abbildung 3: Aufbau der Teststrecke



5.3. Ablauf des Testverfahrens

Bevor mit dem Testen begonnen werden kann, müssen alle nötigen Anwendungen auf den jeweiligen PCs installiert und konfiguriert werden.

1. Auf beiden Windows XP Rechnern die jeweiligen Spiele starten. PC5 (Server) richtet ein Netzwerkspiel ein und PC1 (Client) verbindet sich zu diesem Spiel.
2. Auf PC2 den Netzwerksniffer Wireshark starten, welcher den Netzwerkverkehr von eth1 protokolliert.
3. Auf PC4 den Netzwerksniffer Wireshark starten, welcher den Netzwerkverkehr von eth1 protokolliert.
4. Auf dem Router (PC3) wird das Shell –Skript tcein.sh ausgeführt und somit tc - Konfiguration aktiviert.

5. Mgen auf PC2 mit dem Script Aufruf “mgen input <mgenscript.mgen>“ in der Shell starten

6. Nach 120 Sekunden beendet mgen auf PC2 die Aktion. Danach können alle Anwendungen gestoppt werden.

Die zwei dump-Files von Wireshark auf PC2 und PC4 werden dort gespeichert und stehen später für die Auswertung zur Verfügung. Das dump-File auf PC4 hat den Serververkehr vor dem Router protokolliert und das dump-File auf PC2 hat den Serververkehr nach dem Router protokolliert.

5.4. Testszzenarien

Im Vorfeld wurden 4 Testszzenarien festgelegt, die sich durch die Menge des Hintergrundverkehrs unterscheiden.

1. **Szenario:** Kein Hintergrundverkehr
2. **Szenario:** Wenig Hintergrundverkehr
3. **Szenario:** Mittlerer Hintergrundverkehr
4. **Szenario:** Viel Hintergrundverkehr

Es wurde dabei nicht unterschieden in welcher Spielsituation sich das Spiel gerade befand. Dies wurde vernachlässigt, da davon ausgegangen wurde, dass über die Dauer des Tests nicht nur eine einzelne Spielsituation auftritt und somit dass Testergebnis nicht einer Spielsituation zugeordnet werden kann.

5.5. Spezifikation der Teststrecke

5.5.1. Hardwarespezifikation

Der Testablauf wurde auf folgend Personalcomputern bzw. auf einem Laptop durchgeführt. Diese Rechner hatten folgend Eigenschaften:

PC1 (Windowsrechner links):

Prozessor:	Intel Pentium 4 1,5 GHz
Grafikkarte:	Nvidia GeForce 6600 AGP 4x 256MB
RAM:	1024 MByte
Netzwerkkarte:	Realtek RTL 8139 – Familie - PCI – Fast Ethernet - NIC

PC2 (Linuxrechner links):

Prozessor:	Intel Pentium 4 1,8 GHz
Grafikkarte:	Video Cart Intel Cooperation 82845G/GL/GE Integrated Device 128 MByte
RAM:	1024 MByte
Netzwerkkarte:	Intel Cooperation 82540EM

PC3 (Router):

Prozessor:	AMD Athlon(TM) XP 2200+
Grafikkarte:	ATI Radeon 7000 64 MB AGP 4x
RAM:	512 MB
Netzwerkkarte:	
eth1:	Realtek Semiconductor co.,ltd.RTI-8139/8139c/8139c+
eth2:	Realtek Semiconductor co.,ltd.RTI-8139/8139c/8139c+

PC4 (Linuxrechner rechts):

Prozessor:	Intel Pentium 4 1,7 GHz
Grafikkarte:	Nvidia GeForce 2Gts Pro 128 MByte
RAM:	512 Mbyte
Netzwerkkarte:	3Com 3c906C

PC5 (Windowsrechner rechts):

Prozessor: Intel Pentium 4 1,5 GHz
Grafikkarte: Nvidia GeForce 3 Ti 300 64MB
RAM: 1024 MByte
Netzwerkkarte: 3Com EtherLink XL PCI

PC6 (Windowsrechner):

Prozessor: Intel Core 2 T5500 1,66 GHz
Grafikkarte: NVidia GeForce Go 7300 256 MB
RAM: 2 GByte
Netzwerkkarte: Broadcom 440 10/100Integrated Controller

5.5.2. Softwarespezifikation

Die Rechner in der Teststrecke hatten folgende Software installiert:

PC1:	Betriebssystem:	Microsoft Windows Professional SP2
	Verwendungszweck:	Rechner, auf den die Spiele installiert sind und laufen.
PC2:	Betriebssystem:	Linux Fedora Core 5 Kernel: 2.6.18
	Verwendungszweck:	Aufzeichnung des Netzwerkverkehrs
PC3:	Betriebssystem:	Linux Fedora Core 4 Kernel: 2.6.15
	Verwendungszweck:	Router (Bottleneck)
PC4:	Betriebssystem:	Linux Fedora Core 5 Kernel: 2.6.18
	Verwendungszweck:	Aufzeichnungen des Netzwerkverkehrs und Erzeugung des Hintergrundverkehrs
PC5:	Betriebssystem:	Microsoft Windows Professional SP2
	Verwendungszweck:	Rechner, auf den die Spiele installiert sind und laufen.

PC6:	Betriebssystem:	Microsoft Windows Professional SP2
	Verwendungszweck:	Rechner, auf den die Spiele installiert sind und laufen.

Anmerkung: PC6 wurde zum Testen der Spiele Quake4 und Battlefield 2 eingesetzt, da PC5 die Hardwarespezifikationen nicht erfüllte.

5.6. Tools für die Durchführung der Tests

5.6.1. Netzwerkspezifische Tools

Zur Analyse des Netzwerkes verwendeten wir verschiedene Tools. Mit Hilfe derer Tools war es möglich den Datenfluss zwischen den Rechner auszuwerten.

- **Wireshark**

Name: Wireshark

Verwendungszweck: Protokollierung des Netzwerkverkehrs

Firma: Open Source

Quelle: <http://www.wireshark.org/>

Beschreibung:

Wireshark (zu Deutsch „Kabelhai“ oder „Draht-Hai“, ehemals Ethereal genannt) ist ein so genannter Netzwerksniffer, welcher den Netzwerkverkehr protokolliert, und für das menschliche Auge lesbar macht. Dabei stellt Wireshark die Datenpakete so genau dar, dass man auch tiefe Einblicke in die verschiedenen Protokollheader nehmen kann. Netzwerksniffer werden verwendet, um den Netzwerkverkehr aufzuzeichnen und zu analysieren.

Als Gerald Combs, Entwickler von Ethereal, von Ethereal Software Inc. zu CACE Technologies wechselte, startete er ein eigenes Folgeprojekt und nannte es Wireshark.

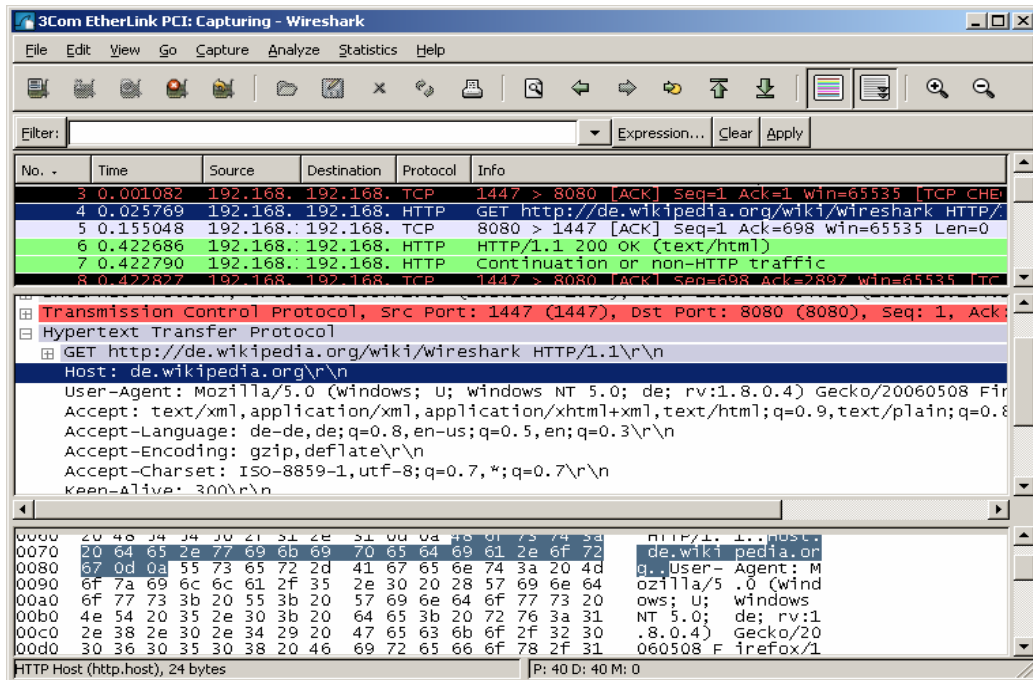


Abbildung 4: Screenshot des Programms Wireshark

- **Tc**

Name: tc

Verwendungszweck: Beschränkung der Bandbreite

Firma: Open Source

Quelle: <http://lartc.org/lartc.html>

Beschreibung: Das Traffic Control Programm dient zum Management von Queuing Disciplines, Verkehrsklassen und Trafficfiltern. In dieser Arbeit wurde es eingesetzt um die vorhandene Bandbreite zu reduzieren und eine Leitung mit einer Bandbreite von 50 KB bzw. 100 KB zu simulieren. Mit anderen Worten, es wurde ein Bottleneck erzeugt, durch den sowohl der von mgen erzeugte UDP Datenstrom, als auch die eigentlichen Daten des Anwendungsprogramms geschickt wurden.

- **Mgen**

Name: mgen

Verwendungszweck: Erzeugen von zuvor definiertem Hintergrundverkehr

Firma: Naval Research Laboratory

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. Dieses Tool wurde verwendet um den Hintergrundverkehr bei den verschiedenen Testfällen zu erzeugen. Das Programm verlangt als Eingabe ein Skriptfile, auf welches später noch genauer eingegangen wird.

5.6.2. Weitere Tools

- **WinPcap**

Name: WinPcap

Verwendungszweck: für WinDump nötig um auf Schnittstellen zu zugreifen.

Firma: Open Source

Quelle: [http:// www.winpcap.org](http://www.winpcap.org)

Beschreibung:

WinPcap ist eine als Open Source nutzbare Programmbibliothek.

WinPcap besteht aus einem Treiber, welcher hardwarenahen Zugriff auf die Netzwerkkarte für Windows-basierte Betriebssysteme ermöglicht und einer Sammlung von Programmen, die den bequemen Zugriff auf die einzelnen für Netzwerke relevanten Schichten des OSI-Modells bieten.

Die Programmbibliothek basiert auf der von Unix her bekannten Bibliothek „libpcap“.

Die über das Netzwerk transportierten Pakete werden durch die WinPcap-Module unter Umgehung des Protokollstacks entgegengenommen und weitergeleitet. Somit können Statistiken über die Netzwerkauslastung, die verschiedenen Paketarten und deren Inhalt protokolliert und analysiert werden.

Konkrete Anwendung findet WinPcap in Netzwerküberwachungssoftware wie zum Beispiel Wireshark, Nmap, Snort, Cain & Abel, WinDump und ntop.

In dieser Arbeit wurde die Auswertung auf einem Windowsrechner durchgeführt. Damit WinDump richtig arbeiten kann benötigt man WinPcap.

- **WinDump**

Name: WinDump

Verwendungszweck: Für Auswertung nötig

Firma: Dieses Programm ist Freeware. Es darf sowohl privat als auch kommerziell kostenlos eingesetzt werden.

Quelle: <http://www.tcpdump.org/>

Beschreibung:

Tcpdump/WinDump ist die bekannteste Software zur Überwachung und Auswertung von Netzwerkverkehr. Sie wurde von Van Jacobson, Craig Leres und Steven McCanne geschrieben, Sie wird aber mittlerweile von vielen anderen weiterentwickelt. Tcpdump/WinDump arbeitet im Textmodus und wird über die Kommandozeile gesteuert.

Das Programm liest Daten in Form von Paketen, die über das Netzwerk gesendet werden und stellt diese auf dem Bildschirm dar oder speichert sie in Dateien.

Zusätzlich ermöglicht Tcpdump/WinDump die Auswertung von vorher in Dateien gespeicherten Paketen. Mittels Parametern, die bei Programmstart auf der Kommandozeile angegeben werden müssen, steuert der Benutzer das Verhalten von WinDump und übergibt Filter an das Programm, wodurch die Pakete ausgewertet werden.

- **Trpr (Tcpdump Rate Plot Real-time)**

Name: Tcpdump Rate Plot Real-time

Verwendungszweck: Erzeugen von Plotfiles

Quelle: <http://pf.itd.nrl.navy.mil/protocols/trpr.html>

Beschreibung

Trpr ist ein Auswertungsprogramm, das aus tcpdump/WinDump Output graphische Darstellungen verschiedener Flows erzeugt. Als Plotter wird extern auf das Programm Gnuplot zurückgegriffen.

- **Gnuplot**

Name: Gnuplot

Verwendungszweck: Grafische Darstellung des Netzwerkverkehrs

Quelle: <http://www.gnuplot.info/>

Beschreibung:

Gnuplot ist ein so genannter Funktionenplotter, ein skript- bzw. kommandozeilengesteuertes Computerprogramm zur grafischen Darstellung von Funktionen und Daten. In dieser Arbeit wurde Gnuplot, verwendet um die gemessenen Daten graphisch darzustellen, um eine bessere Verständlichkeit zu erreichen und um die Daten besser interpretieren zu können.

- **Microsoft Office 2003 SP2**

- **Name:** Microsoft Office 2003 SP2

Verwendungszweck:

Word: Dokumentation dieser Arbeit

Excel: Auswertung der Paketgröße und der Paketanzahl

- **Adobe Acrobat Writer 7.0**

Verwendungszweck: Konvertierung der Dokumentation in PDF

6. Anwendungen

Für unsere Versuchsreihe verwendeten wir eine Reihe von Spielen aus allen Kategorien:

- Strategiespiele:
 - Age of Empires 3
 - Anno 1701
 - Dawn of War
- Shooterspiele
 - Battlefield 2
 - Quake 4
- Sportspiele
 - Fifa 07

6.1. Strategiespiele

Strategiespiele stellen eine Computerspielart dar, bei der alle Spieler oder Computergegner ihre Handlungen gleichzeitig ausführen und in dessen Vordergrund das wirtschaftliche, strategische und taktische Agieren gegen einen Gegner steht. Das Spielprinzip von Strategiespielen wird oft als Echtzeit bezeichnet. Sie entspricht aber keineswegs der harten Definition von Echtzeit: „Denn ein Jahr im Spiel entspricht meist nur wenigen Minuten für den Spieler“.

6.1.1. Age of Empires III

Age of Empires III ist Anfang November 2005 erschienen und beginnt inhaltlich ungefähr im Jahre 1500 und endet etwa um 1860 mit dem amerikanischen Bürgerkrieg.

Es benutzt eine 3D-Grafikengine, die DirectX-9-Effekte und Shader 3.0 nutzt und den Namen Bang trägt. Es handelt sich hierbei um eine erweiterte Version der Engine von Age of Mythology, welche laut Herstellerangaben zu über sechzig Prozent neu geschrieben wurde und zahlreiche Effekte und sonstige Neuerungen (wie beispielsweise höhere Texturauflösung) eingeführt hat.

Zudem wurde eine Havok-Physik-Engine hinzugefügt. Diese soll unter anderem genutzt werden, um den Schaden zu berechnen, der mit Schüssen verursacht wird. So wird beispielsweise die Flugbahn von Kanonenkugeln realistisch und exakt berechnet und dargestellt wie auch das Einstürzen eines Turmes. Diese Physik dient aber nur der Optik, spielerisch wirkt sie sich (im Gegensatz zu anderen Echtzeitstrategiespielen, wie z.B. bei C&C) nicht aus. Für unser Testszenario starteten wir ein Multiplayerspiel mit zwei realen Spielern. Die Tests wurden nicht gleich am Spielanfang durchgeführt, sondern mitten im Spiel. Dort wurde der größte Verkehr beobachtet.



Abbildung 5: Age of Empires III

Entwickler:	Essamble Studios
Verleger:	Microsoft
Publikation:	4. November 2005
Plattform(en):	PC (Windows & Mac, OS) PS2
Spielmodi:	Einzelspieler; Mehrspieler über IPX, TCP/IP, Modem oder Ensemble Studios Online (ESO)

Steuerung:	Maus, Tastatur
Systemminima:	CPU: 1400 MHz, SSE-fähig; RAM: 256 MB; HDD: 2,5GB; GPU: 64-MB-3D-Karte; OS: Windows XP
Medien:	3 CDs
Sprache:	Deutsch

6.1.2. Anno 1701

Anno 1701 ist am 26. Oktober 2006 erschienen. Im Gegensatz zu früheren Anno-Spielen ist keine Kampagne enthalten, stattdessen sind im Endlosspiel Ziele wählbar. Für bestimmte Leistungen, etwa den Abschluss der zehn Szenarien oder das Erreichen einer bestimmten Spielzeit, werden darüber hinaus Medaillen vergeben.

Anno 1701 verwendet ein 3D Engine, die von Related Designs entwickelt wurde.

Bei unserem Testscenario waren nicht nur reale Spieler beteiligt, sondern auch von Computer aus gesteuerte Spieler.



Abbildung 6. ANNO 1701

Entwickler:	Related Designs, Sunflowers
Verleger:	Sunflowers
Publikation:	26. Oktober 2006
Plattform(en):	PC (Microsoft Windows 2000, XP)
Spielmodi:	Einzelspieler, Mehrspieler (bis zu 4 Spieler; Internet & LAN)
Steuerung:	Tastatur, Maus
Systemminima:	2,2 GHz CPU (3+ GHz CPU empfohlen)Grafikkarte mit mindestens 64MB Grafikspeicher und Shader-1 Effekten (Shader2- Grafikkarte mit 256+ MB Grafikspeicher empfohlen)512 MB Arbeitsspeicher (2048 MB RAM empfohlen)
Medien:	1 DVD
Sprache:	Deutsch

6.1.3. Dawn of War

Das Spiel Dawn of War wurde von Relic Entertainment 2004 entwickelt. Es ist ein Echtzeitstrategiespiel das in der Zukunft spielt.

Dawn of War verwendet seine selbst entwickelte Essence Engine. Diese besitzt sehr viele neue graphische Effekte, einschließlich High Dynamic Range und dynamisches Leuchten. Sie benutzt zusätzlich die Havok Physik Engine.

Das Testszenario bei Dawn of War ist ähnlich wie bei ANNO 1701, zwei reale Spieler und bis zu 16 von Computer gesteuerte Spieler.

Als Sieger geht jene Mannschaft hervor, die es schafft zuerst drei strategische Punkte, auf der Zufallskarte insgesamt sieben Minuten zu halten.



Abbildung 7: Dawn of War

Entwickler:	Relic Entertainment
Verleger:	THQ
Publikation:	September 2004
Plattform(en):	PC Windows
Spielmodi:	Einzelspieler, Mehrspieler
Steuerung:	Maus, Tastatur
Systemminima:	CPU: 1,4 GHz; RAM: 256 MB; HDD: 2 GB; Grafik: 3D-Karte mit T&L, 32 MB
Medien:	3 CDs
Sprache:	Deutsch

6.2. Ego-Shooter

Als Ego-Shooter bezeichnet man eine Kategorie der Computerspiele, bei der die Darstellung einer frei begehbaren, dreidimensionalen Spielwelt durch die Augen eines Spielercharakters, also in Egoperspektive erfolgt. Der Spielverlauf ist schwerpunktmäßig geprägt durch den Kampf mit verschiedenen Schusswaffen gegen eine Vielzahl von unterschiedlichen Gegnern.

Typische Merkmale eines Ego-Shooter sind das Blickfeld, Bewegung und Steuerung und Waffen.

6.2.1. Quake 4

Quake 4 ist der vierte Teil der Quake-Serie. Es wurde von id Software für den PC 2005 entwickelt. Quake 4 ist das erste Spiel der Quake-Reihe, welches keine eigens für das Spiel entwickelte Engine nutzt. Stattdessen wird eine überarbeitete Version der Doom-3-Engine verwendet, die unter anderem die Darstellung von Gelände und Hintergrund verbessert.

Getestet wurde ein Szenario ohne Bots. Es waren nur zwei reale Spieler beteiligt.



Abbildung 8: Quake 4

Entwickler:	Ravensoftware, id Software
Verleger:	Activision
Publikation:	18. Oktober 2005 (PC), 2. Dezember 2005 (Xbox 360), 16. Mai 2006 (Macintosh)
Plattform(en):	Mac OS X, PC (Windows, Linux), Xbox 360
Spielmodi:	Einzelspieler, Mehrspieler
Steuerung:	Tastatur und Maus, Gamepad (Konsole)
Systemminima:	Betriebssystem: Windows 2000/XP, Linux ab Kernel 2.4, Mac OS X ab 10.3.9, Prozessor: 2 GHz (Windows, Linux), 1.67 GHz G4 (Mac) Arbeitsspeicher: 512 MB 3D-Karte: 64 MB/ DirectX 9.0c Festplattenspeicher: 2,8 GB DVD-ROM: 4x DVD
Medien:	4 CD-ROMs, 1 bzw. 2 DVDs
Sprache:	Englisch

6.2.2. Battlefield 2

Bei dem Spiel Battlefield 2 handelt es sich um ein taktisches 3D First-Person-Shooter Spiel, das sowohl im Einzel- und Mehrspielermodus spielbar ist.

Es ist am 23. Juni 2005 erschienen. Die Ziffer 2 im Titel steht für den zweiten Teil der Reihe, der von DICE Schweden entwickelt wurde.

Die Grafikengine wurde stark überarbeitet und weiterentwickelt. Das Spiel verwendet die RenderWare-Engine. Das ist 3D-API- und eine Grafikübertragungengine, die bei Computerspielen wie Battlefield 2 verwendet wird und von Criterion Software (mittlerweile Electronic Arts) entwickelt wurde.



Abbildung 9: Battlefield 2

Entwickler:	DICE Schweden
Verleger:	EA
Publikation:	23. Juni 2005
Plattform(en):	PC (Windows XP)
Spielmodi:	Einzelspieler (begrenzt), Mehrspieler (Internet, LAN)

Steuerung:	Maus, Tastatur, Joystick (optional, bessere Steuerung in Flugzeugen)
Systemminima:	Windows XP (32bit Version), 1,7 GHz Prozessor, 512 MB RAM, Grafikkarte mit 128 MB RAM und ShaderModell 1.4, 2,3 GB Festplattenplatz, , DirectX Version 9.0c
Medien:	1 DVD, DVD-Box, 24-Seiten-Handbuch
Sprache:	Deutsch

6.3. Sportspiele

Ein Sportspiel ist ein Computerspiel, das eine bestimmte oder mehrere Sportarten simuliert.

6.3.1. Fifa 07

Fifa 07 ist eine Fußballsimulation für einen echten Fußball-Fan. Das Spiel erschien am 28. September 2006 von Electronic Arts. Seit dem Jahre 1994 erscheint jährlich eine neue FIFA-Version auf dem Markt. Stets werden die Spielsituationen realistischer, die Grafiken Detail getreuer und der Sound beeindruckender. Das Spiel wurde von den Kanadiern der Firma EA-Sports erarbeitet und für folgende Plattformen entwickelt: Playstation2, Xbox, Gamecube und PC.

Für das Testszenario waren zwei Spieler beteiligt die gegeneinander angetreten sind.



Abbildung 10: Fifa 07

Entwickler:	EA Sports
Verleger:	Electronic Arts
Publikation:	28. September 2006
Plattform(en):	PC (Windows 2000, XP)
Spielmodi:	Einzelspieler, Mehrspieler
Steuerung:	Tastatur, Maus, Gamepad
Systemminima:	Betriebssystem: Windows 2000/XP, Prozessor: 1.3 GHz oder schneller, Arbeitsspeicher: 256MB RAM, Festplatte: min. 1.2 GB frei, DVD- Laufwerk: 8x, Grafikkarte: 64MB, Soundkarte: DirectX 9.0c kompatibel, DirectX: Version 9.0c, LAN-Multiplayer: TCP/IP Protokoll, Online Multiplayer: 512 Kb/s oder schneller
Medien:	4 CD-ROMs, 1 bzw. 2 DVDs
Sprache:	Deutsch

7. Skripte und Programme

Es wurden mehrere Skripte verwendet, damit die Tests durchgeführt werden konnten. Zum Einem bediente man sich tc – Skripte, um den Datenfluss zu regulieren und zu beschränken. Weiteres wurden mgen – Skripte verwendet, um Stau zu erzeugen.

7.1. tc – Skripte

Das folgende tc Skript wurde für alle Spiele, außer für Battlefield und Quake 4, verwendet:

```
tc qdisc add dev eth1 root handle 1: cbq avpkt 512 bandwidth 100kbit
```

```
tc class add dev eth1 parent 1: classid 1:1 cbq bandwidth 50kbit rate 50kbit allot  
512 bounded
```

```
tc filter add dev eth1 parent 1: protocol ip prio 16 u32 match u16 0 0 at 0 flowid  
1:1
```

In der ersten Zeile wird zuerst eine Queueing Disziplin des entsprechenden Devices (die Netzwerkkarte eth1) festgelegt. Die verwendete Disziplin ist cbq (class based Queueing). In der zweiten Zeile wird eine Klasse erstellt und die entsprechende Bandbreite auf 50 KBit festgelegt. In der letzten Zeile wird ein Filter erstellt, welcher die Pakete den entsprechenden Klassen zuweist.

```
tc qdisc add dev eth1 root handle 1: cbq avpkt 512 bandwidth 200kbit
```

```
tc class add dev eth1 parent 1: classid 1:1 cbq bandwidth 100kbit rate 100kbit allot  
512 bounded
```

```
tc filter add dev eth1 parent 1: protocol ip prio 16 u32 match u16 0 0 at 0 flowid  
1:1
```


Das ist dasselbe Skript wie zuvor, nur ist hier die Bandbreite auf 100 KBit beschränkt, also doppelt so groß wie zuvor. Dieses Skript wurde für die Spiele Battlefield und Quake 4 verwendet.

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

7.2. mgen – Skripte

Mgen erzeugte den Hintergrundverkehr und wurde somit als Staugenerator verwendet. Wir errechneten den nötigen Hintergrundverkehr, um den gewünschten Stau zu erzeugen. Anschließend konnten die benötigten mgen Skripte erstellt werden.

Zum besseren Verständnis wird hier die Datei „Heavytraffic.mgen“ angeführt.

```
0.0 ON 2 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 30]
```

```
1 OFF 2
```

```
30.0 ON 1 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [500 30]
```

```
90.0 OFF 1
```

```
120.0 ON 3 UDP SRC 5001 DST 192.168.0.2/5000 PERIODIC [1 30]
```

```
121 OFF 3
```

Da die Testdauer immer 120 Sekunden beträgt, war es nötig am Anfang und Ende jeden Testfalls ein (Markierungs-)Paket zu schicken, um später bei der Auswertung die Daten zu synchronisieren. Zwischen der 30 und 90 Sekunde werden dann pro Sekunde 500 Pakete mit jeweils 30 Byte Nutzdaten verschickt (d.h. 58 Byte inklusive IP- und UDP- Header).

Folgende mgen - Skripts wurden erstellt

- **Nottraffic.mgen**

Kein Hintergrundverkehr, es werden nur am Anfang und Ende die Markierungspakete geschickt.

- **Lowtraffic.mgen**

Zwischen der 30 und 90 Sekunde werden 200 Pakete der Größe 30 Byte pro Sekunde versendet

- **Middletraffic.mgen**

Zwischen der 30 und 90 Sekunde werden 300 Pakete der Größe 30 Byte pro Sekunde versendet

- **Heavytraffic.mgen**

Zwischen der 30 und 90 Sekunde werden 500 Pakete der Größe 30 Byte pro Sekunde versendet

Da für Quake und Battlefield eine andere Bandbreite festgelegt wurde, mussten für Quake 4 andere mgen - Skripte erstellt werden. Die folgenden Skripte sind dieselben Pakete wie bisher, nur versenden sie doppelt so viele Pakete.

- **Quake4lt.mgen**

Zwischen der 30 und 90 Sekunde werden 400 Pakete der Größe 30 Byte pro Sekunde versendet

- **Quake4mt.mgen**

Zwischen der 30 und 90 Sekunde werden 600 Pakete der Größe 30 Byte pro Sekunde versendet

- **Quake4ht.mgen**

Zwischen der 30 und 90 Sekunde werden 1000 Pakete der Größe 30 Byte pro Sekunde versendet

7.3. Java Programme

Mit einem selbstentwickelten Java-Programm wurden die Daten ausgewertet. Damit dieses Java- Programm richtig arbeiten konnte, musste man mit Hilfe eines WinDump und trpr Aufrufs die Trace - files bzw. Plotfiles erstellen. Diese Files dienen dem Java-Programm als Eingabefiles. Das Java-Programm ermittelt die Paketanzahl und Paketgröße.

7.3.1. Stapelverarbeitungsdatei

Die unten angegebene Datei erstellt aus den Dump-File (Ausgabedatei von Wireshark) den trace - file und aus den trace – file einen Plot – File. Die letzte Zeile ruft das Programm gnuplot auf, welches den Netzwerkverkehr graphisch darstellt.

```
windump -x -r name.dump >name.trace
```

```
trpr input name.trace key auto X output name.pl
```

```
pgnuplot -persist name.pl
```

8. Testergebnisse

Die Testergebnisse ergaben sich aus den vier getesteten Szenarien und die darauf ausgeführten mgen-Skripte. Dies führte zu unterschiedlichsten Reaktionen bei den Spielen. Diese Situationen wurden während des Spiels protokolliert und ausgewertet.

Im Voraus ist festgestellt worden, dass bei allen getesteten Spielen ausschließlich UDP-Pakete verwendet und versendet wurden.

In nächsten Kapiteln werden die Testergebnisse von den einzelnen Spielen beschrieben und erklärt.

8.1. Age of Empire

8.1.1. Kein Hintergrundverkehr

Der Verkehr zwischen Server und Client ist sehr niedrig, die nötige Bandbreite beträgt nur in etwa 4 KBit. Der Spielserver schickt sehr kleine Pakete in der Größenordnung von 10-15 Byte an Nutzdaten und etwa 10 Pakete pro Sekunde. Diese Werte bleiben während des gesamten Testlaufs konstant. Age of Empires verwendet die kleinsten Pakete, die wir im Rahmen dieser Arbeit gemessen haben. Wie man aus folgendem Diagramm erkennen kann, ist die benötigte Bandbreite minimal und der Datenfluss weist auf Schwankungen hin. Dies bedeutet, dass die Informationsdichte vom Server zum Client von der jeweiligen Spielsituation abhängig ist.

8.1.2. Wenig Hintergrundverkehr

Wir konnten beobachten, dass die benötigte Bandbreite leicht zunimmt wenn es zu einer Engstelle bzw. einem Stau kommt. Daraus kann man schließen, dass Age of Empires eine Netzwerkengine besitzt, welche auf Stau reagiert. Die Paketanzahl ist größtenteils konstant geblieben und die Paketgröße hat sich nicht verändert. Da bei wenig Hintergrundverkehr der Stau zu klein war, reagierte das Spiel ohne den Spielfluss einzuschränken. Das Spiel stockte nicht und zeigte sonst auch keine Veränderungen. Um das Verhalten der Netzwerkengine genauer zu analysieren haben wir noch folgende zwei Tests durchgeführt.

8.1.3. Mittlerer Hintergrundverkehr

Im Vergleich zu den vorherigen Testfällen zeigte das Spiel eine viel deutlichere Reaktion auf den auftretenden Stau. Das Spiel vergrößerte die benötigte Bandbreite deutlich; dies bedeutet, dass der Server mehr Daten zum Client schickt. Aus dem Diagramm AoE MT 3 kann man deutlich sehen, dass die Paketanzahl zunimmt und die Paketgröße in etwa gleich bleibt. Das Spiel begann leicht zu stocken, dennoch war ein normales Weiterspielen möglich.

8.1.4. Viel Hintergrundverkehr

Bei diesem Testfall zeigt „Age of Empires III“ die größte Reaktion. Die Bandbreite nahm stark zu, wie man aus den Diagrammen AoE HT1 und AoE HT2 erkennen kann. Das Spiel reagierte auf dieselbe Art und Weise, wie bereits aus dem vorhergehenden Testfall ersichtlich war. Die Paketgröße blieb konstant und die Paketanzahl nahm sehr stark zu. Spielen war in diesen Testfall nicht mehr möglich, da das Spiel sehr stark stockte. Ungewöhnlich war jedoch, dass die Verbindung nicht unterbrochen wurde, wie bei anderen Spielen beobachtet wurde.

8.2. Anno 1701

8.2.1. Kein Hintergrundverkehr

Wie auch bei Age of Empires ist hier die nötige Bandbreite sehr gering. Anno 1701 kommt auch mit einer Bandbreite von 4 bis 6 Kbit aus.

Aus den Diagrammen von Age of Empires und Anno 1701 kann der Rückschluss gezogen werden, dass Strategiespiele im Allgemeinen eine geringe Bandbreite benötigen. Für den normalen Spielverlauf benötigt das Spiel pro Sekunde nur etwa 5 Pakete mit einer Nutzdatengröße von etwa 65 bis 135 Byte. Aus den Diagrammen ANT 1 und ANT 2 kann man schön erkennen, dass Anno 1701 keinen konstanten Datenfluss hat, d. h. die Bandbreite schwankt sehr.

Aus dem Diagramm ANT 4 lässt sich erkennen, dass die Nutzdatengröße stark schwankt.

8.2.2. Wenig Hintergrundverkehr

Eine Reaktion kann man bereits erkennen, jedoch lässt sich bis zu diesem Zeitpunkt keine eindeutige Strategie ablesen. Die Daten kommen bereits mit einer bestimmten Verzögerung beim Empfänger an. Die Paketgröße sowie die Paketanzahl schwanken (siehe Diagramm ALT 3 und ALT 4), dies lässt vermuten, dass Anno 1701 eine andere Strategie verwendet als Age of Empires. Dies werden wir auch in den folgenden Testszenarien sehen. Was man bis jetzt erkennen kann ist, dass die Bandbreite um 1 KBit verringert wurde (siehe Diagramm ALT 2). Das Spiel begann bereits stark zu stocken, sodass fast nicht mehr an ein vernünftiges Weiterspielen fast undenkbar war.

8.2.3. Mittlerer Hintergrundverkehr

Im Diagramm AMT 1 und AMT 2 wurde beobachtet, dass die Bandbreite höher liegt als im normalen Spielfluss ohne Hintergrundverkehr. Das Spiel reagiert eindeutig auf Stau. Aus dem Diagramm AMT 3 ist ersichtlich, dass die Paketanzahl zu schwanken anfängt. Die Paketgröße (Diagramm AMT 4) nimmt stark zu, bis zu 350 Byte.

Daraus lässt sich schließen, dass das Spiel versucht, den normalen Spielfluss zu gewährleisten, indem es die Pakete vergrößert. Auffällig war, dass bereits bei diesem Testszenario das Spiel so stark stockte, dass ein normales Spielen nicht mehr möglich war.

8.2.4. Viel Hintergrundverkehr

Die deutlichste Reaktion auf Stau war in diesem Testszenarien zu verzeichnen. Das Spiel benötigte teilweise mehr als die doppelte Bandbreite als beim normalen Spielfluss. Die nötige Bandbreite schwankte zudem ziemlich stark (Diagramm AHT 1, AHT 2). Die Paketgröße stieg um das Vierfache (Diagramm AHT 4), wobei die Paketanzahl nicht wesentlich anstieg. Aus diesem Testszenario kann man eindeutig ablesen, dass das Spiel Anno 1701 auf Stau mit Vergrößerung der Pakete reagiert. Bei diesem Testszenario war auch an ein Spielen nicht zu denken, da das Spiel während des Staus überhaupt nicht reagierte. Bei diesem Testszenario

stand das Spiel kurz vor einem Verbindungsabbruch. Kurz bevor sich der Stau auflöste kam beim Client (PC1) eine Meldung mit einem Countdown, dass die Verbindung in 10 sec unterbrochen wird, wenn die Verbindung zum Server nicht wieder aufgebaut werden kann. Der Stau löste sich aber auf bevor die Verbindung unterbrochen wurde.

8.3. Dawn of War

8.3.1. Kein Hintergrundverkehr

„Dawn of War“ ist das Dritte getestete Strategiespiel; wie die anderen beiden Spielen zuvor, benötigt auch Dawn of War eine geringe Bandbreite von 4,4 Kbit (Diagramm DoWNT 1, DoWNT 2). Der normale Spielfluss benötigt eine Paketanzahl von 7 und eine Paketgröße von 48 Byte Nutzdaten (Diagramm DoWNT 3 und DoWNT 4).

8.3.2. Wenig Hintergrundverkehr

Das Ungewöhnliche in diesem Test war, dass die benötigte Bandbreite nicht zugenommen, sondern sogar abgenommen hat. Dies lässt sich aus dem Diagramm DoWLT 4 ablesen. Hingegen bleibt die Paketanzahl konstant (siehe Diagramm DoWLT 3), welches eine weitere unerwartete Reaktion des Spiels war. Bei den Spielen zuvor nahm die Bandbreite stets zu, bei diesem Spiel konnten wir das Gegenteil beobachten. Es konnte noch kein Stocken des Spiels festgestellt werden.

8.3.3. Mittlerer Hintergrundverkehr

In den Diagrammen DoWMT 1-4 ist die Reaktion des Spiels auf Stau sichtbar. Die Bandbreite verringert sich um etwa 1 Kbit und erreicht eine Bandbreite von 3 KBit. Die Paketanzahl bleibt konstant bei 7 Pakete die Sekunde; die Paketgröße verringert sich hingegen um 20 Byte auf 25 Byte. In diesem Testszenario wurde von den Testern ein deutliches Stocken wahrgenommen, jedoch kam Dawn of War besser mit dem Stau zu Recht als die anderen bereits getesteten Spiele.

8.3.4. Viel Hintergrundverkehr

Mit stark auftretendem Stau war Dawn of War überfordert, der Spielfluss wurde unterbrochen. Während der gesamten Stausituation war kein Spielen möglich. Aus dem Diagramm DoWHT 2 kann man ablesen, dass der Server um etwa 1 Kbit weniger Daten schickte und am Client so gut wie keine Daten ankamen. Die Mindestgröße der Pakete scheint sich bei 18 Byte Nutzdaten einzupendeln. Da die Paketanzahl nicht abnimmt kann davon ausgegangen werden, dass die Strategie des Spiels bei Stau darin besteht, nur die Paketgröße zu verändern (Diagramm DoWHT 3, DoWHT 4).

8.4. Quake 4

Für diese vier Tests wurde ein anderes TC-Skript (tcein2.sh) verwendet, da Quake 4 zum normalen Spielverhalten die größere Bandbreite benötigt als alle anderen getesteten Spiele. Speziell für dieses Spiel wurden die mgen-Skripte (quakelt.mgen, quakemt.mgen, quakeht.mgen) spezifiziert und angewendet.

Bei Quake 4 haben nur 2 Spieler in einem Netzwerkspiel gegeneinander gespielt. Es haben keine vom Computer gesteuerten Gegner mitgespielt. Das Spiel beginnt sobald beide Spieler in das Netzwerkspiel eingeloggt sind.

8.4.1. Kein Hintergrundverkehr

Aus den Diagrammen QNT 1 und QNT 2 kann man sofort beobachten, dass die nötige Bandbreite höher liegt als bei den Strategiespielen. Dies lässt erahnen dass der Datenaustausch größer sein muss, um den Spielfluss zu gewährleisten. Infolgedessen sollten Ego-Shooter-Spiele eine bessere Staustrategie verwenden als Strategiespiele, da es in der Praxis öfters zu einer Stausituation kommen könnte.

Das Spiel benötigt eine Bandbreite von etwa 15 Kbit und schickt in etwa 15 Pakete der Größe 110 Byte pro Sekunde.

8.4.2. Wenig Hintergrundverkehr

Bereits bei diesem Szenario kann man aus dem Diagramm QLT 1 und QLT 2 erkennen, dass die benötigte Bandbreite zugenommen hat. Sie hat zeitweise sogar 40 Kbit erreicht. Aus den Diagrammen QLT 3 und QLT 4 ist ersichtlich, dass die Paketanzahl gleich geblieben ist, sich jedoch die Paketgröße vervierfacht hat. Das Diagramm weist zwei gut erkennbare Spitzen auf. Daraus lässt sich schließen, dass zu diesen Zeitpunkten ein Gefecht stattgefunden hat. Wenn geschossen wird, steigt der Verkehr an, da sich viele Objekte (Projektilen usw.) in der Spielumgebung ändern und Synchronisierung erforderlich wird.

8.4.3. Mittlerer Hintergrundverkehr

In diesem Testszenario stieg die Bandbreite wieder deutlich an. Im Diagramm QMT 1 und QMT 2 ist wieder ersichtlich, dass zwei Gefechte stattgefunden haben. Die Paketgröße ist wie beim Testszenario zuvor stark angestiegen. Ein normales Spielen war nicht mehr möglich, die Spielfigur des Clients konnte nicht mehr gesteuert werden. Der Clientspieler konnte sich in der Spielwelt nicht mehr fortbewegen, hingegen konnte der Serverspieler sich ganz normal bewegen. Wenn der Client Spieler getötet wurde, gab es eine mehrsekundige Verzögerung bis der Spieler wieder in das Spiel eintreten konnte. Weiteres war zu beobachten, dass sich nach dem Stau für einige Zeit der Serverspieler auf der Clientseite viel schneller bewegte. Dies ist darauf zurückzuführen, dass der Client die gesamten Bewegungen des Servers, welche er während der Stausituation tätigte, nachholte.

8.4.4. Viel Hintergrundverkehr

Auch hier kann man die Reaktion des Spiels auf Stau vom Diagramm QHT 1 und QHT 2 ablesen. Der Spielfluss wurde gleich wie beim Szenario mit mittlerem Hintergrundverkehr beeinflusst. Die Paketgröße war auch in etwa gleich wie beim Szenario mit mittlerem Hintergrundverkehr. Somit hat Quake 4 dieselbe Staustrategie wie Anno 1701. Allerdings liegt die benötigte Bandbreite bei Quake doch um einiges höher.

8.5. Battlefield 2

Bei Battlefield 2 waren am Spielgeschehen 2 Spieler und 14 Bots beteiligt. Aus diesem Grund benötigte Battlefield 2 von allen getesteten Spielen die größte Bandbreite. Es wurden dieselben Mgen-Skripte (quakelt.mgen, quakemt.mgen, quakeht.mgen) verwendet wie bei Quake 4.

Hier wird wie beim Spiel Quake 4 das TC – Skript (tcein2.sh) verwendet.

8.5.1. Kein Hintergrundverkehr

Bei den Ego-Shooterspielen kann man beobachten, dass sie eine größere Bandbreite benötigen als andere Spielarten. Bei einer Bandbreite von ungefähr 80 Kbit benötigt Battlefield 2 eine deutlich höhere Bandbreite als Quake 4. Ein wesentlicher Unterschied ist auch, dass Battlefield 2 weniger, aber dafür größere Pakete schickt. Der Server schickt in etwa 20 Pakete der Größe 400 Byte zum Client.

8.5.2. Wenig Hintergrundverkehr

Fügt man dem Spiel ein wenig Hintergrundverkehr hinzu (90 Kbit), so kann man an der Bandbreite keinen Unterschied erkennen. Sie beträgt, gleich wie bei dem Szenario mit keinem Hintergrundverkehr, 80 Kbit. Aus dem Diagramm erkennt man, dass die Paketgröße sehr leicht nach unten korrigiert wurde. Die Paketgröße wird anfangs von 460 Kbit auf 420 Kbit herabgesetzt.

8.5.3. Mittlerer Hintergrundverkehr

Nach einer weiteren Erhöhung des Hintergrundverkehrs um 50 Kbit auf 140 Kbit war ein flüssiges Spielen unmöglich geworden. Das anfängliche Ruckeln des Spiels hat sich in ein unbewegliches Bild verwandelt. Der Client konnte, solange der Hintergrundverkehr aktiv war, ins Spielgeschehen nicht mehr aktiv eingreifen.

Die Bandbreite blieb relativ konstant bei 80 Kbit bis zum Hinzufügen des Hintergrundverkehrs. Interessant zu beobachten war die Anzahl der Pakete: die Pakete wurden nur mehr schubweise abgeschickt. Die Anzahl reduzierte sich bis fast auf 3 Pakete pro Sekunde und stieg bis zu 35 Pakete in der Sekunde an. Die Paketgröße veränderte sich von 500 Byte auf 5 Byte. Aus dem Diagramm BMT 2 kann man herauslesen, dass der Server, sobald er eine Stausituation bemerkt, versucht er in zufälligen Intervallen, Pakete zu schicken. Der Server schickt während der Stausituation einige Zeitintervalle lang fast keine Pakete, sodass die Bandbreite auf 0 Byte sinkt.

8.5.4. Viel Hintergrundverkehr

In diesem Szenario wurde das Verhalten bei Stau noch deutlicher. Das Spiel stockte, und es kam die Fehlermeldung, dass Verbindungsprobleme aufgetreten sind. Der Server schickte schubweise Pakete, doch der Durchsatz sank auf 0. Die Verbindung wurde nicht abgebrochen, allerdings wurde ein flüssiges Spielen unmöglich.

In diesem Testfall kann man aus dem Diagramm BHT 1 erkennen, dass der Servers in unregelmäßigen Abständen immer wieder Pakete zu versenden versucht. Diese Sendepausen vom Server konnten wir bei keinem anderen Spiel beobachten.

Die anfängliche Durchsatzquote von 20 Paketen pro Sekunde wurde sehr schnell fast auf 0 reduziert. Eine Kommunikation zwischen Server und Client wurde unmöglich. Die Größe der Pakete fing bei 400 Byte an und fiel fast auf 0 Byte.

8.6. Fifa 07

8.6.1. Kein Hintergrundverkehr

Da hier auch sehr viel synchronisiert werden musste z.B. die Position des Balls oder die Position von den 22 Spielern, die sich ständig ändert, muss eine große Menge von Daten ausgetauscht werden.

Die nötige Bandbreite für Fifa 07 sind 12 KBit. Man kann auch aus dem Diagramm FNT 1 beobachten, dass ab und zu die Bandbreite ganz kurz nach oben schießt, um ein flüssiges Spielen zu gewährleisten.

Ähnliches ist bei der Paketgröße zu beobachten. Meist ist sie konstant bei 43 Byte mit gelegentlichen Spitzen, wobei die Paketgröße bis zu 130 Byte ansteigen kann. Ganz anders ist es mit der Anzahl der Pakete. Der Server schickt konstant 20 Pakete pro Sekunde.

8.6.2. Wenig Hintergrundverkehr

Eine erste Reaktion kann man hier schon feststellen. Der Client ist im Spielgeschehen ein paar Sekunden hinter dem Server (zeitlich). Ein leichtes Stocken des Spiels konnte man auch beobachten. Sobald der Hintergrundverkehr beendet war, wurde das Spiel beim Client für eine kurze Zeit schneller. Der Stau wurde aufgelöst und man bekam wieder volle Bandbreite, der Client erhielt viele Pakete auf einmal.

Die Bandbreite schießt auf 100 Byte hinauf sobald man Hintergrundverkehr von 90 Byte pro Sekunde hinzufügt. Zu beobachten war, dass die Anzahl der Pakete sich um das Vierfache verringerte und anschließend stark variierte. Die Größe der Pakete erhöhte sich um das 11fache, von 30 Byte auf 330 Byte.

Das Spiel versucht mit sehr wenigen, aber dafür sehr großen Paketen den Stau zu verringern.

8.6.3. Mittlerer Hintergrundverkehr

Beim Hinzufügen vom mittleren Hintergrundverkehr (140 Byte pro Sekunde) ist ein flüssiges Weiterspielen unmöglich geworden. Die Paketgröße wurde, wie im Szenario mit wenig Hintergrundverkehr, um das 11fache erhöht. Die Paketanzahl variiert sehr stark. Dagegen wurde die Bandbreite auf 70 Kbit erhöht, um 30 Kbit weniger als wenn man weniger Hintergrundverkehr hat.

8.6.4. Viel Hintergrundverkehr

Bei diesem Szenario mit viel Hintergrundverkehr konnte man sehr gut beobachten, dass sich die Bandbreite nur schubweise erhöhte. Sie variierte zwischen 10 Kbit und 140 Kbit. Die Anzahl und die Größe der Pakete verändern sich, wie in dem vorherigen gegangenen Testszenario. In diesem Szenario war ein Spielen nicht mehr möglich

9. Zusammenfassungen

9.1. Zusammenfassung Age of Empires 3

Age of Empire 3 war das erste getestete Spiel. Die Testergebnisse zeigen eine eindeutige Strategie der Staubewältigung: Die Anzahl der Pakete wird kontinuierlich erhöht, wobei die Paketgröße konstant bleibt. Age of Empire kam mit dem Stau relativ gut zurecht, d.h. das normale Spielverhalten blieb aufrecht.

Bei zu großem Stau bleibt die Verbindung bestehen und das Spiel wird „eingefroren“.

Auffällig war auch, dass bei größeren Spielszenarien, in denen viele Objekte sichtbar waren, ein Anstieg des Netzwerkverkehrs erkennbar war. Age of Empires kommt mit einer sehr geringen Bandbreite aus, nur im Staufall wird diese vergrößert.

9.2. Zusammenfassung Anno 1701

Anno 1701 ist auch wie Age of Empires 3 ein Aufbaustrategiespiel und benötigt daher fast dieselbe Bandbreite für den regulären Spielverlauf. Diese beträgt zwischen 4 bis 6 KBit.

Auch bei Anno 1701 kann man die Staustrategie gut erkennen, es reagiert auf Stau indem es die Pakete vergrößert. Dabei ist auffällig, dass die Anzahl der Pakete zu schwanken beginnen. Das Spiel stockte bei den Testszenarien sehr stark, beim letzten Testszenario kam es fast zu einem Verbindungsabbruch.

9.3. Zusammenfassung Dawn of War

Bei Dawn of War wurde beobachtet, dass die benötigte Bandbreite bei Stau abnahm. Die Anzahl der Pakete blieb konstant, die Paketgröße hingegen nahm über die Hälfte ab. Dabei war auffällig, dass dieses Spiel von den getesteten Strategiespielen den normalen Spielfluss am längsten aufrechterhalten konnte. Beim letzten Testszenario kam das Spiel genau wie die anderen Strategiespiele zum Stillstand.

9.4. Zusammenfassung Quake 4

Das Ego-Shooterspiel Quake 4 reagiert auf Stau, indem es die Pakete vergrößert und die Anzahl der Pakete leicht variiert. Hier konnte man erkennen, dass sich bei auftretendem Stau, der Spieler auf der Clientseite nicht mehr bewegen konnte. Der Spieler beim Server konnte hingegen normal weiterspielen, und es traten auf dieser Seite keine Verzögerungen auf. Quake 4 vergrößert bei Stau die Bandbreite auf das Doppelte.

9.5. Zusammenfassung Battlefield 2

Battlefield 2 ist das Spiel, welches die größte Bandbreite benötigt etwa 80 Kbit. Ein flüssiges Spielen wird schon bei geringem Hintergrundverkehr fast unmöglich. Das Spiel versucht Stausituationen durch Verringerung der Anzahl der Pakete mit mäßigem Erfolg entgegenzuwirken. Der Server versucht schubweise die Pakete zu übertragen, wenn ihm dies nicht gelingt, unterbricht er die Verbindung. Dieses Spiel hatte augenscheinlich die schlechteste Staustrategie von allen getesteten Spielen.

9.6. Zusammenfassung Fifa 07

Aus den Testszenarien mit Fifa 07 war eine deutliche Strategie zu erkennen, wie das Spiel auf Stausituationen reagiert. Sobald im Netzwerk Stau auftritt, senkte das Spiel die Anzahl der Pakete und erhöhte die Paketgröße auf das 11fache. Dies war bei allen drei Testszenarien (wenig, mittel und viel Hintergrundverkehr) zu beobachten.

Die Bandbreite hingegen war bei den Tests sehr unterschiedlich. Bei wenig und viel Hintergrundverkehr stieg die Bandbreite am meisten und mit mittlerem Hintergrundverkehr stieg sie am geringsten an. Das Spiel variiert die Bandbreite mit zunehmendem Stau. Sobald sehr viel Datenverkehr im Netzwerk zu erkennen war, erhöhte es leicht die Bandbreite. Sobald der Datenverkehr nachließ, verringerte das Spiel die Bandbreite.

Schlusswort

Das Ziel dieser Arbeit bestand darin herauszufinden ob netzwerkfähige Spiele Staukontrolle besitzen. Dazu wurde ein Versuchsaufbau entwickelt, der den Datenverkehr zwischen zwei Spielen aufzeichnet und unterschiedliche Stausituationen erzeugt. Die unterschiedlichen Stausituationen sollten eine Reaktion der Spiele erzwingen. Die Aufzeichnungen der verschiedensten Testdurchläufe und die anschließende grafische Auswertung gaben Aufschluss darüber, wie ein Computerspiel auf eine auftretende Stausituation reagiert.

Tests waren für eine Reihe von Spielen aus allen Kategorien geplant:

- Age of Empire
- Anno 1701
- Battlefield 2
- Quake 4
- Dawn of war
- Fifa 07
- Medieval 2 Total War
- Need for Speed Most Wanted
- Splinter Cell
- Doom 3
- Halo
- Far Cry
- Spellforce 2
- Counterstrike

Die Spiele, die auf eine Stausituation reagierten, wurden in verschiedenen Szenarien getestet. Die Spiele Medieval 2 Total War, NFS MostWanted, Splinter Cell und Doom 3, Halo, Far Cry, Spellforce 2 und Counterstrike wurden aus hardwarespezifischen Gründen nicht getestet.

Aus den Beobachtungen der verschiedenen Testszenarien kann man schließen, dass die meisten Spiele Staubewältigungsstrategien besitzen. Jedoch besitzen Spiele derselben Kategorie nicht dieselben Staubewältigungsstrategien.

Es hängt vom Netcode der einzelnen Spiele, ab wie sie auf Stau reagieren, wenn er also sehr stabil ist kann es nur zu geringen Verzögerungen kommen. Es ist heute meistens so, dass derselbe Netcode in mehreren Spielen implementiert wird. Erstaunlicher Weise gibt es keine Standardstrategie, welche von den Spielfirmen verfolgt wird um auf Stau zu reagieren. Der eine Netcode kümmert sich nicht um Stau, der andere verdoppelt oder verdreifacht die Packetanzahl und ein andere wiederum vergrößert die Packetgröße um das x-fache.

Abschließend ist zu erwähnen, dass es nicht gattungsabhängig ist, ob ein Spiel eine Staukontrolle besitzt oder nicht.

Quellenverzeichnis

<http://www.gnuplot.info>
<http://lartc.org/lartc.html>
<http://mgen.pf.itd.nrl.navy.mil/mgen.html>
<http://www.tcdump.org>
<http://pf.idt.nrl.navy.mil/protocols/trpr.html>
<http://www.wikipedia.org>
<http://www.wincap.org>
<http://wirshark.org>

Abbildungsverzeichnis

Abbildung 1: Client-Server Architektur
Abbildung 2: <http://de.wikipedia.org/wiki/UDP>
Abbildung 3: Aufbau der Teststrecke
Abbildung 4: Screenshot von Programm Wireshark
Abbildung 5: Screenshot vom Spiel Age of Empires III
Abbildung 6: Screenshot vom Spiel ANNO 1701
Abbildung 7: Screenshot vom Spiel Dawn of War
Abbildung 8: Screenshot vom Spiel Quake 4
Abbildung 9: Screenshot vom Spiel Battlefield 2
Abbildung 10: Screenshot vom Spiel Fifa 07

Danksagung

Für die Betreuung und Unterstützung während der Arbeit danken wir Dr.-Ing Michael Welzl. Für die Bereitstellung der Teststrecke danken wir Sven Hessler und dem Institut für Informatik in Innsbruck.