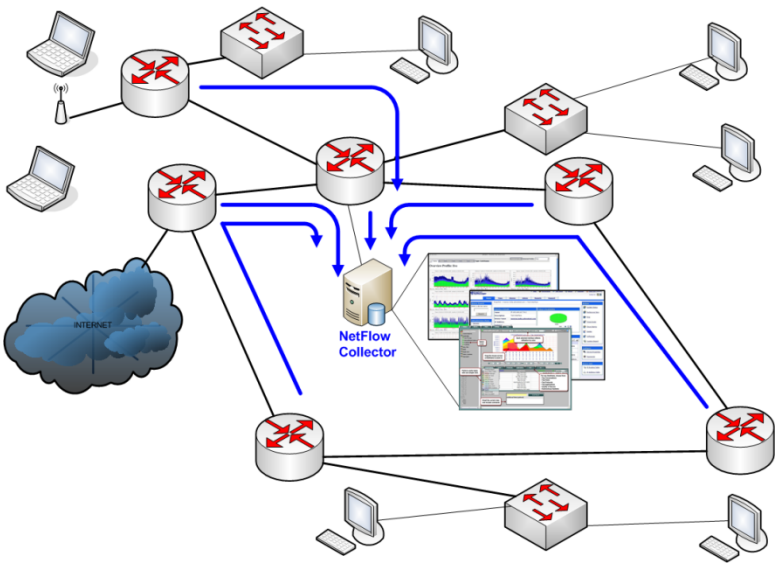
HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

INFORMATIK

Diplom- / Bachelorarbeit

Router Forensics: Challenging NetFlow Accuracy



Herbstsemester 2009

HSR Hochschule für Technik Rapperswil

Abteilung Informatik

Studenten: Marcel Jakopic
Christian Jung

Betreuer: Prof. Eduard Glatz

Gegenleser: Prof. Dr. Adreas Rinkel

Projektpartner: Bernhard Tellenbach,
ETH Zürich

Experte: Roberto Pajetta

1 ABSTRACT

Bei J-Flow von Juniper Networks hat eine Arbeit der UPMC Universität in Paris (Université Pierre et Marie Curie) nachgewiesen, dass bei Langzeitmessungen Lücken in der Auswertung der J-Flow‘s entstehen.

Diese Diplom- / Bachelorarbeit untersucht nun die Genauigkeit von Cisco‘s NetFlow gegenüber dem realen Datenverkehr, ob auch bei dieser Technologie die gleichen oder andere Phänomene auftreten.

Das NetFlow-Verhalten wird auf einem Cisco Router 2621 getestet. Dabei wird untersucht wie der Router reagiert wenn die Ressourcen, also die CPU oder die Bandbreite, knapp werden.
Es wird ermittelt was passiert wenn die Flow-Table gefüllt wird und ob dieses Verhalten der Definition von Cisco entspricht.
In einem weiteren Test werden die Zeitstempel analysiert. Dabei wird errechnet, ob die Zeiten in NetFlow eine Abweichung zum realen Datenstrom beinhalten. Die Computer-Uhren werden initial mit NTP synchronisiert und bei einem zusätzlichen Testaufbau mit einer speziellen Karte ausgestattet, welche es erlaubt, die Messung sehr exakt durchzuführen (auf 3-5 Mikrosekunden genau).

Ein Resultat war, dass der Router das NetFlow-Paket verwirft, sobald das entsprechende Interface ausgelastet ist, auf welchem der Router das NetFlow-Paket senden sollte.
Hingegen behält der Router die NetFlow’s wenn die CPU ausgelastet ist, bis wieder Kapazität der CPU vorhanden ist und die NetFlow’s in einem oder mehreren NetFlow-Paketen versendet werden können.
Die NetFlow’s werden frühzeitig exportiert, respektiv abgeschlossen, sobald der Platz in der Flow-Table zu Ende geht. Das gemessene Verhalten entspricht nicht der Cisco-Definition. In der Gesamtdauer aller Flows ergeben sich deswegen Abweichungen zum realen Datenverkehr.
Bei den Timestamps wird aufgezeigt, dass NetFlow für alle TCP-Verbindungen die Dauer im Mittelwert um 0.993 Sekunden länger anzeigt, als die realen Verbindungen bestehen.

2 MANAGEMENT SUMMARY

2.1 Ausgangslage

Cisco entwickelte NetFlow um statistische Aussagen in einem Netzwerk zu errechnen. Ein Netzwerkgerät (meistens ein Router oder ein Layer 3 Switch) sammelt Informationen über die einzelnen Verbindungen und schickt diese als NetFlow-Paket zu einem NetFlow-Kollektor. Dieser Kollektor speichert die Informationen in einer Datenbank oder einem spezifischen Dateiformat ab. Aus diesen Daten lassen sich danach Statistiken errechnen.

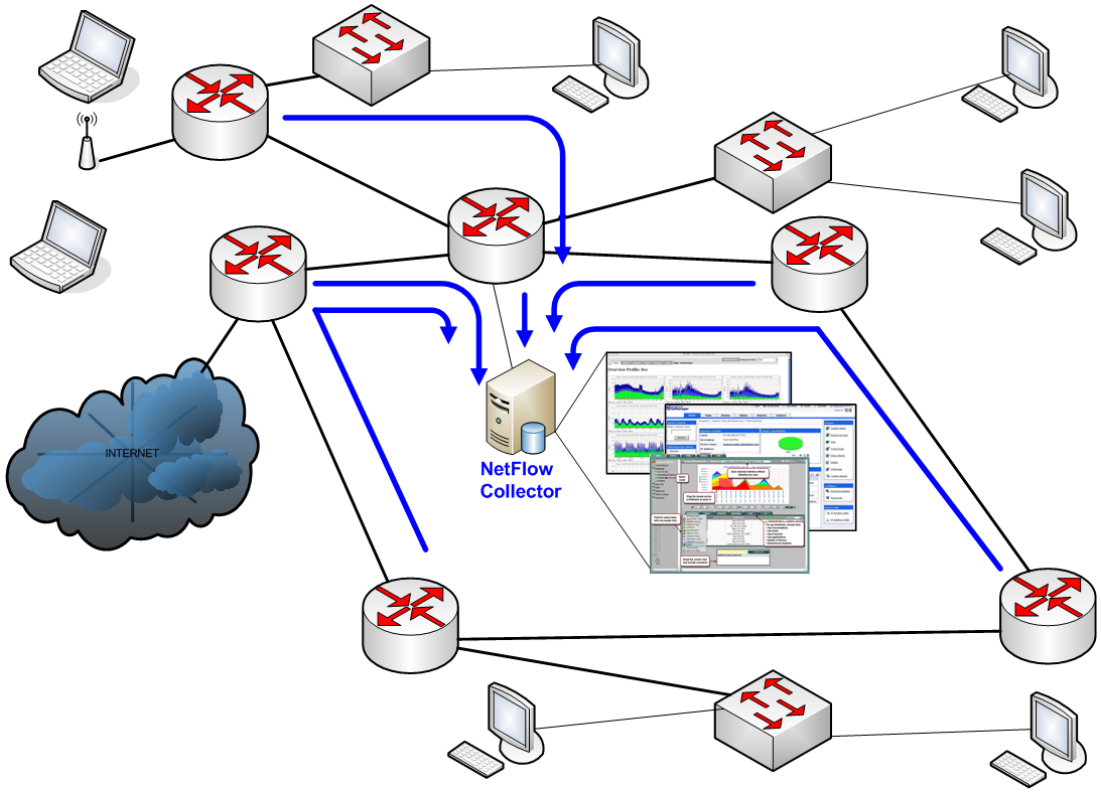


Abbildung 2-1 Beispielaufbau für NetFlow

Die ETH Zürich erhält NetFlow-Datenexporte von SWITCH, um ihnen bei neuen Auswertungsverfahren behilflich zu sein.
J-Flow von Juniper Networks, ein ähnliches Protokoll wie Cisco's NetFlow, wurde in einer Arbeit der Universität UPMC in Paris näher untersucht. Dabei wurden Lücken in den Paketdaten nachgewiesen.
Die Aufgabe dieser Arbeit ist, in verschiedenen Untersuchungen, herauszufinden, wie zuverlässig diese NetFlow-Pakete verschickt werden und ob Abweichungen zum realen Datenstrom entdeckt werden können.

2.2 Ergebnisse

Die verschiedenen Untersuchungen sind in Labs mit dem jeweiligen Versuchsaufbau aufgeteilt. Aus diesen Labs sind folgende Ergebnisse entstanden:

CPU-Auslastung durch die Aktivierung von NetFlow

In einem Lab wird die Prozessoraktivität gemessen, wenn NetFlow-Daten erzeugt werden. Die Messungen ergeben, dass die Aktivierung von NetFlow ca. 2 Prozent CPU-Auslastung beansprucht. Dabei gibt es kleinere Unterschiede je nach Protokoll (1.9% für UDP und 2.29% für TCP).

Die Anzahl der NetFlows pro Sekunde hat keinen Einfluss auf die CPU-Auslastung. Eine Erklärung dafür ist, dass der Router, um die Pakete zu routen, alle Pakete überprüfen muss. Ob nun das Paket zu einem bestehenden NetFlow gehört oder ein neuer NetFlow ist, hat für die CPU-Auslastung keine Relevanz.

NetFlow-Verhalten bei hoher Auslastung des Interfaces zum NetFlow-Kollektor

Der Router sendet die NetFlow-Pakete relativ unregelmässig. Abweichungen von mehreren Sekunden sind durchaus nichts Ungewöhnliches. Der konfigurierte Wert des Export-Timeouts beträgt 15 Sekunden und gemessen werden Werte zwischen 11 und 30 Sekunden.

Wenn im selben Moment die Bandbreite des Interfaces, auf welchem ein NetFlow-Paket verschickt wird, ausgelastet ist, wird dieses NetFlow-Paket verworfen.
Anhand der Sequenznummer der NetFlow-Pakete kann ermittelt werden, dass die NetFlows verloren gegangen sind und der Inhalt ebenfalls verloren geht.
In der Praxis bedeutet dies, dass ein eigenes Interface, an welchem nur der NetFlow-Kollektor angeschlossen ist, genutzt werden soll.

NetFlow-Verhalten bei hoher CPU-Auslastung

Der Router speichert trotz kritischer CPU-Auslastung die NetFlow-Informationen. Wenn jedoch keine CPU-Kapazität zum Verarbeiten zur Verfügung steht, wird solange gewartet, bis wieder Kapazität vorhanden ist. Die NetFlow-Pakete werden zurück gehalten und erst wieder gesendet, wenn der Router genügend Kapazität zur Verfügung stellt. Es gehen dabei keine Informationen verloren.

Die Exportzeit kann beliebig verlängert sein. Bei über 10 Minuten konstanter CPU-Auslastung werden auch danach noch die korrekten NetFlow-Pakete verschickt.

In der Praxis sollte kein Router so lange unter kritischer CPU-Auslastung stehen. In einem solchen Fall müsste der Router neu dimensioniert, resp. ersetzt werden.

NetFlow-Verhalten bei Erreichung der Limite der NetFlow-Table

Nach der Definition von Cisco sollte der Router genau 30 NetFlows frühzeitig exportieren, wenn nur noch zwischen 1 und 30 freie Plätze in der NetFlow-Table sind. Dies kann nicht bestätigt werden, da eine unterschiedliche Anzahl an NetFlows frühzeitig exportiert wird, sogar eine ungerade Anzahl ist möglich.
Dies beantwortet auch die Frage, ob der Router auch immer die zusammengehörenden NetFlows bei einer TCP-Verbindung exportiert.

Wenn NetFlows frühzeitig exportiert werden, stimmt die Flow-Dauer nicht mehr mit der Realität überein, was bei Abrechnungen anhand der Flows Fehler verursachen kann.

Genauigkeit der Zeitstempel in NetFlow

In einem Computer wird zusätzlich eine Turbo CAP Karte installiert, welche 2 Netzwerkinterfaces besitzt und die Timestamps auf 3-5 Mikrosekunden genau setzen kann. Dies ermöglicht eine sehr genaue Zeitmessung, da die Synchronisation über NTP im Windows sehr unzuverlässig funktioniert.

Bei jeder TCP-Verbindung ist die gemessene Zeitdauer der Flows im Mittelwert um 0.993 Sekunden grösser als sie real ist. Bei vielen kurzen TCP-Verbindungen können in der Analyse gravierende Abweichungen auftreten.

2.3 Ausblick

Die Analyse der NetFlow-Pakete hat ergeben, dass die gesammelten Daten vorwiegend von den NetFlow-Exporteinstellungen des Routers abhängen und somit die Genauigkeit stark beeinflusst.

Aus den gesammelten Erfahrungen mit den verschiedenen NetFlow-Kollektoren wird für die zukünftigen Arbeiten NFDump empfohlen. Die grafische Anzeige mit NFSen lässt auch ein Monitoring für den Netzwerk-Administrator zu, welches frühzeitig mögliche Probleme entdecken kann.

Als weitere Messung, welche noch genauer untersucht werden sollte, kann ein repräsentativer Datenstrom genannt werden. Dieser sollte aus verschieden grossen Paketen, Protokollen, sowie aus synchronen und asynchronen Verbindungen bestehen.
Ein solcher Traffic Generator müsste noch gefunden, selbst programmiert oder teuer eingekauft werden.

Die Zeitmessung mit Wireshark unterliegt stark der Genauigkeit der verwendeten Rechner und Netzwerkkomponenten. Diese korrekt zu synchronisieren ist keine triviale Aufgabe und könnte durch einen komplexeren Aufbau und durch die Verwendung von geeigneteren Programmen oder Komponenten die Genauigkeit der Messung steigern.

3 INHALTSVERZEICHNIS

4 EINLEITUNG

4.1 Ausgangslage

Cisco Systems[1] ist Marktführer im Bereich von Computernetzwerken. Entwickelt werden neben Netzwerkprodukten auch Protokolle, welche die Administration von Netzwerken vereinfacht. Zur Überwachung von Netzwerken haben Sie ein Protokoll entwickelt, welches die Verbindungen der einzelnen Peers auflistet. Dieses heisst NetFlow[2] und wurde bis zur Version 9 weiterentwickelt. Das Protokoll wird im Standard RFC3954[3] beschrieben. Die IETF[4] entwickelte auf Grund von NetFlow Version 9 einen offenen Standard weiter, welcher genau im RFC3917[5] beschrieben wird und nennt sich IPFIX. NetFlow Version 5 von Cisco ist der am häufigsten genutzte Standard, weil die meisten Applikationen zur Analyse der NetFlows im Internet frei erhältlich sind.

Router oder andere Layer 3 Netzwerkgeräte sammeln Informationen über Netzwerkverbindungen. Diese Informationen werden mittels dem Protokoll NetFlow Version 5 an einen NetFlow-Kollektor gesendet. Der NetFlow-Kollektor speichert die Daten in einer Datenbank oder einem anderen spezifischen Format ab. Aus den Daten lassen sich Statistiken ableiten und geben einen guten Überblick über den Zustand des Netzwerkes. Mit NetFlow können Echtzeit-Statistiken oder auch Langzeit-Trends nachgewiesen werden. Beliebte Echtzeit-Statistiken sind TopTalkers, Kommunikationsverbindungen, Bandbreitenauslastung, VLAN-Analysen oder verwendete Applikationen.
TopTalkers liefert die Peers, welche am meisten Bandbreite nutzen. Zudem zeigen die Kommunikationsverbindungen an, welche Ziele häufig angewählt werden und wo mögliche Engpässe im Netzwerk entstehen können. Dadurch lassen sich oft gewählte Netzwerkverbindungen präventiv ausbauen. VLAN-Analysen geben Informationen preis, die angeben, ob Broadcast-Domänen weiter unterteilt werden müssen. Mit den Applikationsanalysen kann festgestellt werden, welche Programme auf dem Netzwerk miteinander kommunizieren. Bei den Langzeit-Trends können Bandbreitenzunahmen über längere Zeiträume aufgezeigt und so einen möglichen Ausbau des Netzwerkes begründet und geplant werden.

SWITCH[6] nutzt ebenfalls NetFlow, um die Internetverbindungen aus und in die Schweiz aufzuzeichnen. Die ETH Zürich [7]unterstützt SWITCH bei den Auswertungen, respektive die ETH Zürich erhält die NetFlow-Datenexporte von der SWITCH zur Entwicklung von neuen Analysealgorithmen.

Andere Hersteller von Netzwerkgeräten haben eigene Lösungen zur Netzwerkanalyse entwickelt. Juniper Networks[8] setzen bei ihren Routern J-Flow ein, was ein ähnliches Protokoll wie Cisco‘s NetFlow ist. J-Flow wurde in einer Arbeit[9] der UPMC Universität [10]in Paris näher untersucht. Dabei wurden Lücken in Langzeitmessungen der J-Flow-Paketdaten nachgewiesen.

Die Aufgabe dieser Arbeit ist es, das Protokoll NetFlow von Cisco in diversen Messungen detailiert zu untersuchen, um eventuelle Schwachstellen zu erkennen. Dabei wird die Zuverlässigkeit der zu sendenden NetFlow-Pakete eruiert. Die Genauigkeit der NetFlow-Informationen wird ebenfalls gemessen.

4.2 Zielsetzung

Die rechtsgültige Aufgabenstellung ist im Anhang nachzulesen.

Die Aufgabe besteht darin die gesammelten Informationen von NetFlow mit den Informationen aus den IP-Paketdaten zu vergleichen und zu analysieren.

Dies geschieht mit mehreren Messungen unter verschiedenen Betriebszuständen des Routers. Dabei kann ermittelt werden, ob auch NetFlow wie J-Flow gewisse Ungenauigkeiten besitzen.

4.3 Vorgehensweise

Zuerst werden generelle Informationen zum Thema NetFlow gesammelt.

Die Projektplanung wird ebenfalls zu Beginn der Diplom- / Bachelorarbeit lanciert. Dabei wird ein Projektplan mit den einzelnen Arbeitspaketen erstellt und die Meilensteine mit deren Zeitpunkt definiert.

Um NetFlow-Pakete zu erhalten muss der Router entsprechend konfiguriert, ein NetFlow-Kollektor installiert und ein Traffic Generator benutzt werden. Diese werden nach, noch zu bestimmten Kriterien, begutachtet und getestet.

Nach der Auswahl der Software werden die Messversuche begonnen. Dabei müssen der Aufbau sowie die Messung selbst genau definiert werden, damit sie nachvollziehbar und reproduzierbar sind.

4.4 Abgrenzung

Die Abgrenzungen werden anhand der Labs definiert. Da es sich um eine Forschungsarbeit handelt, können gewisse Teile wegfallen oder neue dazustossen.

Gegeben ist das Equipment, welches von der HSR gestellt wird. Diese sind ein Cisco Router 2621 [11]und 3 Fujitsu Siemens Computer als Arbeitsplätze, welche nachfolgend noch genauer spezifiziert werden.

5 AUFBAU

5.1 Namenskonventionen

Die Labs wurden der Reihe nach nummeriert (Bsp: Lab 1, Lab 2, ...).
Der Inhalt der Labs wurde in der Aufgabenstellung der einzelnen Labs genau definiert.

Gab es aus der ursprünglichen Messung heraus weitere Messungen, welche Bereiche detaillierter untersuchten, wurde die Untermessungen mit einer fortlaufenden Nummer ergänzt (Bsp: Lab 1-1, Lab 1-2, usw.).

Wenn ein Fehler während einer Messung auftrat und die gesamte Messung nochmals wiederholt werden musste, wurde die Labbezeichnung mit einem Buchstaben aus dem Alphabet ergänzt. (Bsp: Lab 1b, Lab 1c, usw.).

5.2 Aufbau

Für die Arbeit sind 2 Computer und 1 Router vorhanden. Diese sind wie in der folgenden Abbildung miteinander verbunden.

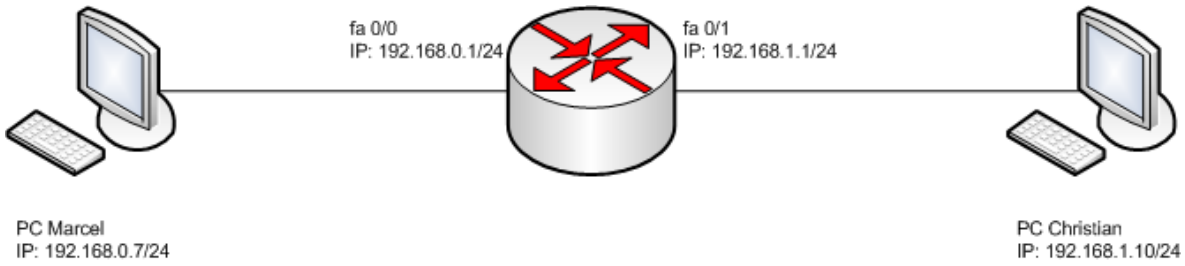


Abbildung 5-1 Netzwerkaufbau

Der Router besitzt ein Consolen-Port, zwei FastEthernet-Ports und einen Auxiliary-Port, welcher nicht verwendet wurde. Zusätzlich zu den vorhandenen FastEthernet Interfaces ist ein Netzwerkmodul eingebaut worden, um den Router mit weiteren 4 Ethernet Interfaces zu bestücken.

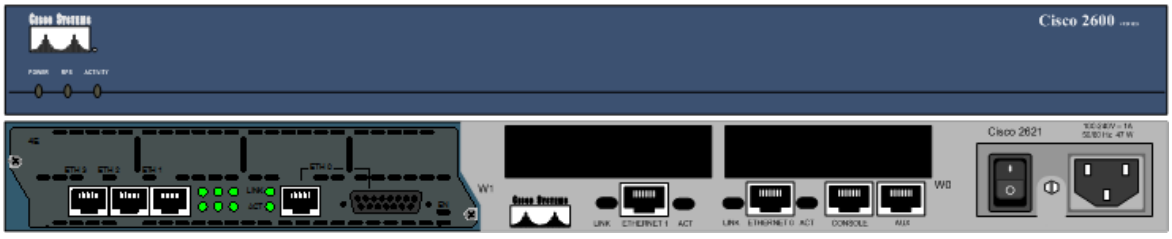


Abbildung 5-2 Front- und Rückansicht des Routers

Der Cisco Router ist vom Typ 2621 und besitzt folgende in Tabelle 5-1 aufgelisteten Eigenschaften.

Typ	CISCO 2621
SW-Version	C2600-J1S3-M
File	C2600-j1s3-mz.123-26.bin
Compiled Date	17.03.2008, 15:23
IOS-Version	12.3 (26)
Processor	M860
Memory	64 MBytes
Module	Network Modul 4E

Tabelle 5-1 Router

Die Computer sind von der HSR und installiert ist ein Dualbootsystem mit Windows XP Professional (SP3) und Fedora 11.

Typ	Fujitsu Siemens
SW-Version	Windows XP Professional SP3/ Fedora 11
CPU	Intel Core 2 Duo @ 2.66 GHz
Memory	3 GB

Tabelle 5-2 Computer

Für die Testumgebung werden Veränderungen am System vorgenommen. Diese dienen dazu, dass das Betriebssystem oder eine Software keine ungewollten Netzwerkdaten sendet und somit die Messungen verfälschen kann. Die Veränderungen sind in der Tabelle 5-3 aufgelistet.

Typ	Software / Dienst
Betriebssystem	Windows XP Professional
Service Pack	Service Pack 3
Deaktivierte Netzwerkdienste	NetBIOS
	Client für Microsoft-Netzwerke
	QoS Paketplaner
	Datei und Druckerfreigabe für Microsoft-Netzwerke
	Microsoft Firewall
Deaktivierte Services	Windows Update Services
	Computer Broswer
	McAfee Framework-Dienst
	Remote Registry
	Workstation
	Server
Deinstallierte Software	Gizmo
	Symantec Ghost Client
	PML Driver HPz12
	McAfee

Tabelle 5-3 Windows Konfiguration

Zusätzlich wird weitere Software benötigt und auf den Systemen installiert.

Software	Version	Website
Wireshark	1.2.2	http://www.wireshark.org/download.html
WinPcap	4.1 beta 5	http://www.winpcap.org
VMWare Player	2.5.3 build-185404	http://www.vmware.com/de/products/player/
SVN Tortoise	1.6.5, Build 16974	http://tortoisesvn.net/downloads
Mozilla Firefox	3.5.3	http://www.mozilla-europe.org/de/firefox/
Iperf	1.7.0	http://www.softliste.de/iperf.html
D-ITG	2.6.1	http://www.grid.unina.it/software/ITG/index.php

Tabelle 5-4 Software

Auf der VMWare läuft als virtuelles Betriebssystem Ubuntu[12] mit der Desktopversion 9.04.

5.3 Router-Konfiguration

5.3.1 Grundkonfiguration Router

Die Grundkonfiguration des Routers ist hier abgebildet. Alle Abweichungen von dieser Konfiguration werden separat in den Labs erwähnt. Zum Einloggen wurde der Router bereits vorkonfiguriert.

Benutzername: stud
Passwort: stud

```
Current configuration : 1785 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname DA_BA
!
boot-start-marker
boot-end-marker
!
enable password 7 1047070A251B1309
!
memory-size iomem 10
clock timezone CST 1
clock summer-time CDT recurring
no aaa new-model
ip subnet-zero
!
no ip domain lookup
!
ip cef
!
username stud privilege 15 password 7 071C35594A
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 no cdp enable
```

```
!
interface FastEthernet0/1
 ip address 192.168.3.1 255.255.255.0
 no cdp enable
!
interface Ethernet1/0
 ip address 192.168.0.1 255.255.255.0
 no cdp enable
!
interface Ethernet1/1
 ip address 192.168.202.1 255.255.255.128
 no cdp enable
!
interface Ethernet1/2
 ip address 152.96.193.206 255.255.252.0
 no cdp enable
!
interface Ethernet1/3
 no ip address
 no cdp enable
!
 ip http server
ip http authentication local
ip classless
!
!
no cdp run
!
snmp-server community public RO
snmp-server ifindex persist
!
line con 0
 password 7 000D1D1524570A04
 logging synchronous
 login local
line aux 0
line vty 0 4
 logging synchronous
 login local
!
!
end
```

5.3.2 NetFlow aktivieren

Damit der Router die Neflow-Pakete schickt, muss auf jedem Port Netflow aktiviert sein:

```
Router-cache flow
```

Folgende globale Einstellungen sind nötig:

```
!Definition an welche IP und Port die NetFlow Daten gesendet
werden
ip flow-export destination 192.168.0.7 9999
!Definition des Absenders
ip flow-export source FastEthernet 0/1
!Definition der Version
ip flow-export version 5
!sendet spätestens nach 1 Minute ein NetFlow Paket
ip flow-cache timeout active 1
!sichert, dass abgeschlossene Flows nach 15 Sekunden geschickt
werden
ip flow-cache timeout inactive 15
!fixiert die Interface Namen global (auch nach Reboot)
snmp-server ifindex persist
```

5.3.3 SNMP aktivieren

Damit der Router die SNMP-Anfragen beanworten kann, muss eine Community für Lesezugriff definiert werden:

```
snmp-server community public ro
```

6 SOFTWARE – EVALUATION

Nachfolgend sind die verschiedenen Softwaregruppen (Monitoring, NetFlow-Kollektor, Flow und Traffic Generator) mit den jeweiligen Anforderungen beschrieben. Eine Auswertung wird im letzten jeweiligen Unterkapitel vorgenommen.

Die Gewichtung der Kriterien und die Bewertung der Funktionen in den einzelnen Applikationen liegen zwischen 1 und 10.

Die Gewichtung ist in der Tabelle 6-1 nach 3 Kategorien unterteilt. Um eine klare Abgrenzung zu erreichen, wird jeweils ein Gewichtungswert zwischen den Kategorien ausgelassen.

Kategorie	Gewichtung
Killerkriterium	10
Wünschenswerte Funktionen	5-8
Nice to have	1-3

Tabelle 6-1 Kategorien mit Gewichtung

Für die Gesamtbewertung der Software wird der Gewichtungswert mit der Bewertung multipliziert. Die Bewertung ist in Funktionale und Organisatorische Kriterien gegliedert.

Die Gewichtung der einzelnen Funktionen ist in einer gemeinsamen Diskussion festgelegt und beruht auf Erfahrungswerten des Teams.

6.1 Auswahl der Monitoring Software

Das Ziel dieser Softwaregruppe ist, die Bandbreite beider Ethernet-Schnittstellen, die CPU-Auslastung und das Memory des Routers zu überwachen.
Wichtig ist die Möglichkeit für die Automatisierung der Software, damit der manuelle Aufwand so klein als möglich gehalten wird. Das Intervall muss selbst definiert werden können und die Software muss gratis erhältlich sein.
Wünschenswert ist eine Speicherung der empfangenen Daten und eine möglichst einfache Installation.
Eine grafische Darstellung ist vorteilhaft, aber nicht nötig.

6.1.1 MRTG

MRTG[13] von Tobi Oetiker stellt Graphen dar, welche die Informationen über SNMP bekommen. Die Software wurde in Perl geschrieben und läuft unter Linux/Unix, Windows und sogar unter NetWare-Systemen.
MRTG ist gratis erhältlich (lizenziert unter Gnu GPL) und wird in vielen Firmen eingesetzt. Als Referenz und Demo gilt die Installation bei Switch.

6.1.1.1 Installation

Für die Installation wird PERL[14] vorausgesetzt. MRTG kann einfach in das gewünschte Verzeichnis entzippt werden.

Damit MRTG funktionsfähig wird, muss ein Konfigurations-File erstellt werden:

```
Perl cfgmaker public@192.168.0.1
-- global „WorkDir: c:\www\mrtg“
–- output mrtg.cfg
```

Damit beim Start von MRTG die Applikation umgehend die Daten entgegennimmt und neu auswertet, muss definiert werden, dass sie als Daemon läuft. Im Windows besteht zudem die Möglichkeit, MRTG als Service zu konfigurieren:

```
RunAsDaemon: yes
```

Nun lässt sich MRTG mit der vorher kreierten Konfigurationsdatei wie folgt starten:

```
Perl mrtg mrtg.cfg
```

Die Command Line darf jetzt nicht mehr geschlossen werden.

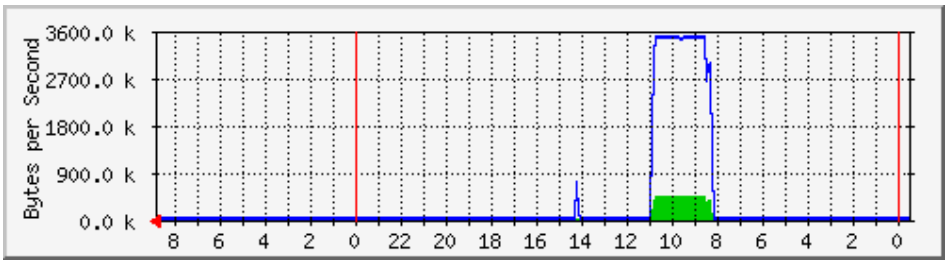


Abbildung 6-1 MRTG: Fa 0/0 in 5 Minuten Abständen

Die Zusätzlichen Messbereiche müssen manuell hinzugefügt werden. Hier ein Beispiel für das Monitoring der CPU:

```
# Router CPU load %
Target[cpu.1]:1.3.6.1.4.1.9.2.1.58.0&1.3.6.1.4.1.9.2.1.58.0:public
@192.168.0.1
RouterUptime[cpu.1]: public@192.168.0.1
MaxBytes[cpu.1]: 100
Title[cpu.1]: CPU LOAD
PageTop[cpu.1]: <H1>CPU Load %</H1>
Unscaled[cpu.1]: ymwd
ShortLegend[cpu.1]: %
XSize[cpu.1]: 380
YSize[cpu.1]: 100
YLegend[cpu.1]: CPU Utilization
Legend1[cpu.1]: CPU Utilization in % (Load)
Legend2[cpu.1]: CPU Utilization in % (Load)
LegendO[cpu.1]:  Usage
Options[cpu.1]: gauge
```

6.1.1.2 Pro und Contra

+ Die Software ist gratis
+ Einfache Erstellung der Basis-Konfigurationsdatei (automatische Interface-Erkennung)

- Konfiguration komplex, besitzt eigene Syntax
- Braucht Webserver
- standardmässige HTML-Generierung pro Abfragewert, manuelle Änderung der Konfigurationsdatei
- Kleinstes Intervall der SNMP-Abfragen ist 5 Minuten (mit rrdtool[15] bei 1 Sekunde)

6.1.1.3 Fazit

Die Messung für die CPU-Auslastung und den Speicher hat nicht funktioniert, obwohl verschiedene Parameter verwendet wurden. Der Router liefert bei keinem Parameter einen brauchbaren Wert bei den abgefragten OIDs zurück.

Nach einer grösseren Einarbeitungszeit ist diese Applikation sicher sehr zweckmässig und da viele Parameter einstellbar sind, äusserst flexibel.

Für diese Arbeit weist diese Applikation viele Funktionen auf, welche nicht benötigt werden.

6.1.2 PRTG

PRTG wird von der Firma Paessler[16] angeboten. Es existiert eine freie und eine kostenpflichtige Version. Die Gratisversion kann auch in Firmen verwendet werden, allerdings nur mit 10 Sensoren, also 10 verschiedene Schnittstellen können gleichzeitig überwacht werden.
Die Überwachung erfolgt über SNMP oder WMI, je nach zu überwachender Schnittstelle.
Weiterhin können noch mehr Technologien (z.B. NetFlow) verwendet werden, allerdings müsste dann zur Verkaufsversion gewechselt werden.

6.1.2.1 Installation

Die Installation ist GUI-basiert. Für die Anzeige der Daten wird ein freier Port benötigt. Auf diesem können die Informationen über einen Browser abgerufen werden.

Zusätzlich lässt sich ein Windows-GUI installieren, welches direkt auf den Port verbindet und der Browser überflüssig wird.

PRTG nennt die einzelnen Messpunkte Sensoren. Ein Sensor kann zum Beispiel die Bandbreite oder die CPU-Auslastung messen. Es stehen schon viele vorkonfigurierte Sensoren zur Verfügung.

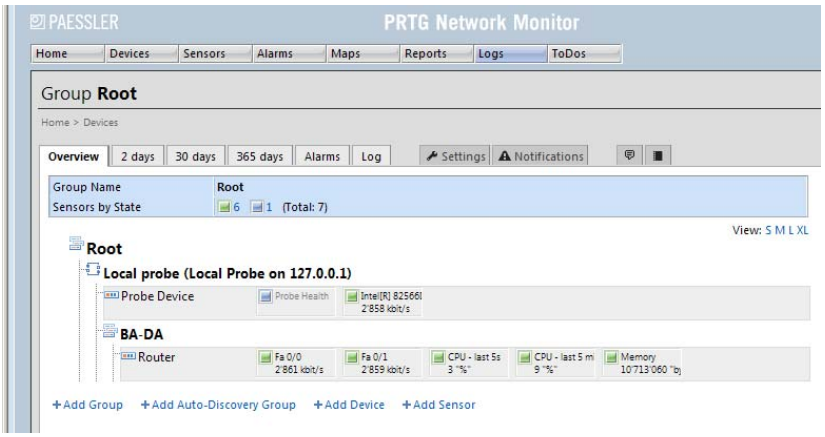


Abbildung 6-2 PRTG: Übersicht aller vorhandenen Geräte (hier nur der Router)

Die Installation erstellt 2 Services (PRTG 7 Core Server Service / PRTG 7 Probe Service).

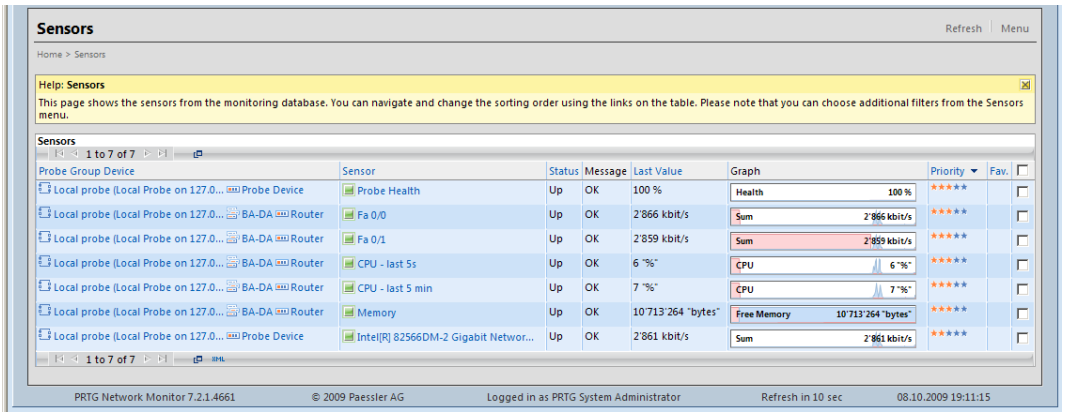


Abbildung 6-3 PRTG: Übersicht aller Sensoren

6.1.2.2 Pro und Contra

+ Einfache Installation
+ Sensoren können beliebig konfiguriert werden
+ Abfrage der einzelnen OIDs
+ Viele Sensoren schon vorkonfiguriert, können ausgewählt und hinzugefügt werden.

- Freeware-Edition unterstützt nur bis 10 Sensoren
- NetFlow in der Freeware-Version nicht unterstützt
- SNMP-Intervall bei 1 Minute (Vollversion bei 1 Sekunde)

6.1.2.3 Fazit

Diese Applikation war ursprünglich als NetFlow-Kollektor vorgesehen. Einen Sensor ist bereits vorkonfiguriert. Allerdings ist dieser in der Freeware-Version nicht funktionsfähig.

Die Überwachung der CPU-Auslastung und des Speichers haben am Anfang Probleme verursacht, es schien, als wären die OIDs falsch.

Entgegen der CISCO-Dokumentation befindet sich der CPU-Wert (letzten 5 Sekunden) bei 1.3.6.1.4.1.9.9.109.1.1.1.1.6.1 (CISCO-Doku: 1.3.6.1.4.1.9.9.109.1.1.1.1.6) und der freie Speicher bei 1.3.6.1.4.1.9.2.1.8.0 (CISCO-Doku: 1.3.6.1.4.1.9.2.1.8)

6.1.3 Command Line Interface (CLI) des Routers

Beim Verbinden mit dem Router (oder andere Cisco Geräte) über Telnet lassen sich Befehle und Konfigurationen durchführen.
Die CLI kennt 3 Betriebsmodi:

- User Mode:
  Nach dem Einloggen erscheint dieser, er ist über den Prompt `>` zu erkennen. In diesem Modus lassen sich nur Basis-Kommandos ausführen. Das Gerät kann nicht konfiguriert oder neu gestartet werden. Ein Beispiel für ein Kommando ist der `Show`-Befehl. Mit dem Befehl `enable` (und dem dazugehörenden Passwort)gelangt man in den nächsten Modus.
- Priviledged Mode:
  In diesem Modus sind alle Befehle freigeschalten. Es lässt sich zum Beispiel die Konfiguration anzeigen (`show running-configuration`). Der Privileged Mode lässt sich über den Prompt `#` erkennen. Damit das Gerät konfiguriert werden kann, muss in den nächsten Modus mit `configuration terminal` geschalten werden.
- Configuration Mode:
  Hier lässt sich das Gerät konfigurieren. Es ist unterteilt in verschiedene Sub-Modes. Der Einstieg ist im Global Configuration Mode, dies ist am `(config)#` Prompt zu erkennen. Ein Sub-Mode ist z. B. die Konfiguration eines Interfaces, dann ändert sich der Prompt in `(config-if)#`.

6.1.3.1 Pro und Contra

+ Keine Installation, da standardmässig vorhanden
+ Einfache Abfrage CPU-Auslastung (Befehl `show processes cpu`; letzte 5 Sekunden, letzte Minute, letzte 5 Minuten)

- Kann nicht gespeichert werden

6.1.3.2 Fazit

Die CLI ist zum Abfragen ideal, wenn der momentane Wert entscheidend ist. Für unser Projekt benötigen wir jedoch eine Aufzeichnung der Daten. Somit fällt die CLI aus dem Rennen, obwohl es sicher eine Möglichkeit gäbe, auf dem Router ein Script auszuführen, welches die Werte ein eine Datei schreiben würde.

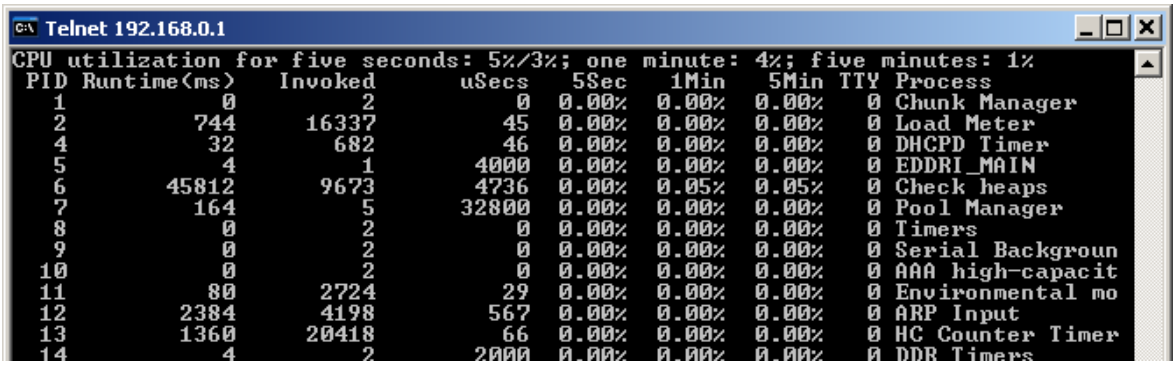


Abbildung 6-4 Beispiel CLI: CPU-Auslastung der letzten 5 Sekunden, 1 Minute und 5 Minuten

6.1.4 iReasoning MIB-Browser[17]

Diese Software unterstützt die SNMP-Abfragen mit einem GUI. Es können die OIDs direkt angegeben werden oder sie lassen sich in einem MIB-Tree, welche geladen werden können, sehr einfach suchen. Mit einem MIB-Tree lassen sich die gewünschten Werte abfragen ohne, dass man die exakte OID kennt.

Die Gratisversion lässt bis zu 10 verschiedene MIB-Tree's zu, welche geladen werden können.

6.1.4.1 Installation

Sehr einfache GUI-basierte Installation.

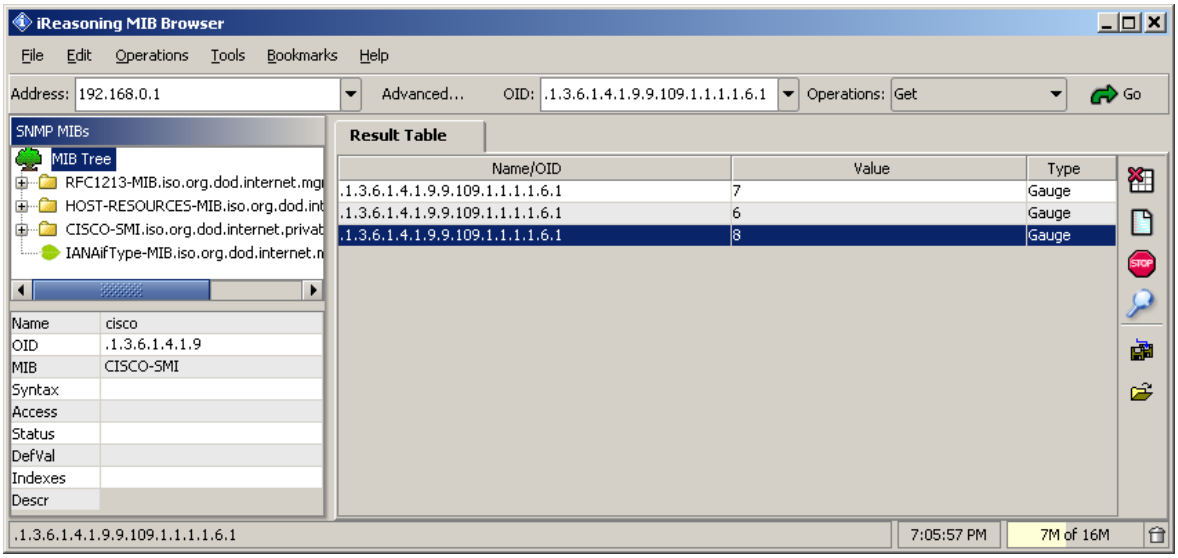


Abbildung 6-5 Beispiel IReasoning: SNMP-Abfrage nach der CPU-Auslastung (letzte 5 Sekunden)

6.1.4.2 Pro und Contra

+ Freeware
+ 10 MIBs gleichzeitig geladen (unbeschränkt bei Vollversion)
+ Einfache Installation
+ Abfrage-Methode auswählbar (Get, GetNext, Bulk, Walk)

- Speichert keine Daten
- Keine Diagramme / Verlauf

6.1.4.3 Fazit

Diese Applikation ist ideal um schnell einen Wert abzufragen. Für unsere Bedürfnisse allerdings zu wenig umfangreich, da ein Speichern der Antworten nicht vorgesehen ist.

6.1.5 WhatsUp Gold

IpSwitch[18] behauptet, dass mehr als 100 000 Netzwerke mit What's Up überwacht werden. Es gibt 4 Varianten zum Kaufen, je nach Grösse des Unternehmens.

Es ist eine Trial-Version erhältlich. Dazu können noch viele Plugins mitinstalliert werden.

6.1.5.1 Installation

Diese Applikation ist schon auf einem Lab-Notebook installiert, welches für ein Modul verwendet wird. Daher gibt es keine Installation von unserer Seite.

6.1.5.2 Pro und Contra

- Kostenpflichtig
- Unübersichtliches GUI
- komplexe Konfiguration

6.1.5.3 Fazit

Um das Problem wegen den falschen OIDs zu erörtern, gibt diese Applikation die Bestätigung, dass die von Cisco angegebene OIDs nicht funktionieren.

Das Notebook mit der vorinstallierten Software steht während der ganzen Arbeit nicht zur Verfügung. Die Software müsste daher erworben werden.

6.1.6 SNMPWalk

SNMPWalk[19] ist eine kleine Applikation, welche es ermöglicht in der Kommandozeile SNMP-Abfragen zu generieren.
Das Programm ist frei verfügbar und die Konfigurationsmöglichkeiten sind gross.

6.1.6.1 Installation

Ubuntu hält ein Paket bereit, welches den SNMPWalk installiert:

```
sudo apt-get intall apache2
```

Anschliessend kann dieser benutzt werden:

```
Snmpwalk –v1 –c public 192.168.0.1 .iso
```

Dieser Befehl durchläuft alle Einträge ab dem Startpunkt `iso`.

6.1.6.2 Pro und Contra

+ Gratis
+ Scriptfähig
+ Leichte Installation
+ Grosser Befehlsumfang

6.1.6.3 Fazit

Die einfache Handhabung und die Scriptfähigkeit lassen SNMPWalk vielseitig einsetzen. Die vielen Parameter begünstigen diese Software und ermöglichen einen Einsatz für jedes Netzwerkgerät. Der OID muss entweder bekannt sein oder lässt sich aus einer Datei einlesen.

6.1.7 Auswertung

In der Gesamtbewertung hat SNMPWalk die meisten Punkte. Durch die Einbettung in Scripts lassen sich die Werte in Dateien schreiben, womit diese Software alle Anforderungen am Besten erfüllt.

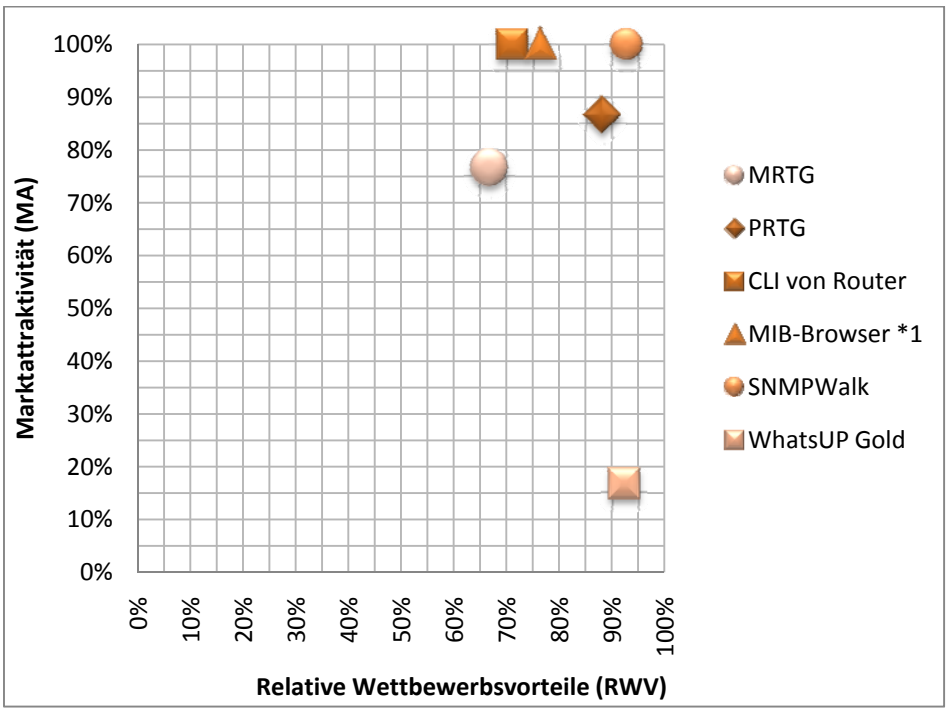


Abbildung 6-6 Auswertung Monitoring

Die detaillierte Auswertung ist in der Portfolio Analyse[20] zu finden.

6.2 Auswahl der NetFlow-Kollektor / Analyzer Software

Die Anforderungen dieser Applikation sind die NetFlow Daten zu empfangen, abzuspeichern und zu analysieren. Die Analyse kann z.B. eine Auswertung der verschiedenen Protokolle beinhalten.
Das Empfangen der NetFlow-Pakete und ein Exportieren der einzelnen Flows sind zwingend.
Das Speichern der NetFlows, die Scriptfähigkeit und ein Konvertieren von PCAP-Files in Flows sind wünschenswerte Eigenschaften.
Eine grafische Darstellung der NetFlows ist nicht unbedingt eine Voraussetzung.

6.2.1 Scrutinizer 7

Die Firma Plixer International[21] vertreibt die Software Scrutinizer und hält die Software als Trial zum Download bereit. Allerdings muss man sich registrieren, um die Software zu erhalten.

Auf der Kundenliste stehen sehr viele Firmen aus verschiedenen Branchen, wie Gesundheit, Finanzbranche, Transportunternehmen, Schulen usw.

6.2.1.1 Installation

Das Installationsfile ist fast 250 MB gross, was bedeutet, dass relativ viel Code verwendet wurde (im Vergleich zur zweitaktuellsten Version). Die Installation ist GUI-basiert. Die Applikation benötigt die Information auf welchem Port die NetFlow Daten entgegengenommen werden sollen.

Der Browser zeigt einen Fehler im Script an. Der Fehler im `auth.js` Script konnte nicht gefunden werden, da die Zeit für eine detaillierte Fehlersuche nicht ausreichte.
Die Software ist nicht kompatibel mit dem Browser Firefox[22], was ein schneller Quertest verunmöglicht.

Eine Neuinstallation hat auch nichts gebracht. Daher wird versucht eine ältere Version zum Laufen zu bringen.

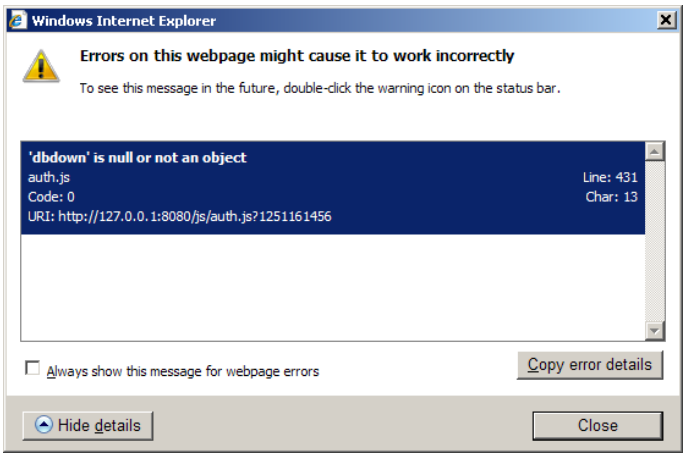


Abbildung 6-7 Scrutinizer 7: Fehlermeldung

6.2.1.2 Pro und Contra

Eine Bewertung ist nicht möglich.

6.2.1.3 Fazit

Es wird versucht die ältere Version zum Laufen zu bringen.

6.2.2 Scrutinizer 6

Auf der Internetseite von Scrutinizer existiert kein direkter Link um die Version 6 herunterzuladen. Wählt man jedoch die Version 7, wird die Möglichkeit gegeben auch die Version 6 herunterzulanden.

6.2.2.1 Installation

Das Installationspaket von Scrutinizer 6 ist etwas mehr als 70 MB gross. Auch diese Version ist GUI-geführt. Die Applikation benutzt ein Port, auf welchem von einem Browser zugegriffen werden kann.

Die Freeware-Version kann 99999 Geräte verwalten. Allerdings sind viele Features gesperrt. Für eine erste Analyse genügen diese Möglichkeiten.

Der Router kann hinzugefügt und die empfangenen NetFlow-Pakete können danach zugeordnet und dargestellt werden.

Grafisch können die verschiedenen Protokolle, Hosts, Applikationen, Top Conversations usw. angezeigt werden. Der Zeitraum kann minutengenau eingestellt werden.

Die Applikation löscht automatisch alle Informationen nach 1 Tag, wer die Daten länger behalten möchte, muss die Vollversion erwerben.

6.2.2.2 Pro und Contra

+ Übersichtlich
+ Grafische Analyse

- Daten sind nach 24h weg
- Viele Features nur mit Vollversion benutzbar
- Registrierung für Download nötig

6.2.2.3 Fazit

Scrutinizer 6 läuft stabil und sammelt die NetFlow-Pakete. Ein Netzwerkadministrator kann damit sein Netzwerk durchwegs gut überwachen.

Für unsere Bedürfnisse fehlt ein Speichern der Netflows, damit man selber die Analyse durchführen kann. Die Auswertungen sind vorgegeben.

6.2.3 NetFlow Analyzer 7

Der Hersteller von NetFlow Analyzer 7 heisst ManageEngnine[23] und hat seinen Sitz in den USA. Die Website bietet eine 30 Tage-Trial-Version mit vollem Umfang.

6.2.3.1 Installation

Der NetFlow Analyzer wird auch über ein GUI installiert. Wie Scrutinizer benötigt diese Applikation auch einen Port, auf welchem über den Browser zugegriffen werden kann.

Das Web-GUI ist übersichtlicher gestaltet als Scrutinizer. Die Geräte werden automatisch anhand der NetFlow Daten aufgezeigt.

Die deutsche Übersetzung ist schlecht, es empfiehlt sich die englische Installation.

Abbildung 6-8 NetFlow Analyzer 7: Übersicht der Geräte, welche NetFlow schicken

Mit nur einem Klick, kann z.B. in eine Grafik hineingezoomt werden. Die Benutzerführung ist einfach gehalten.

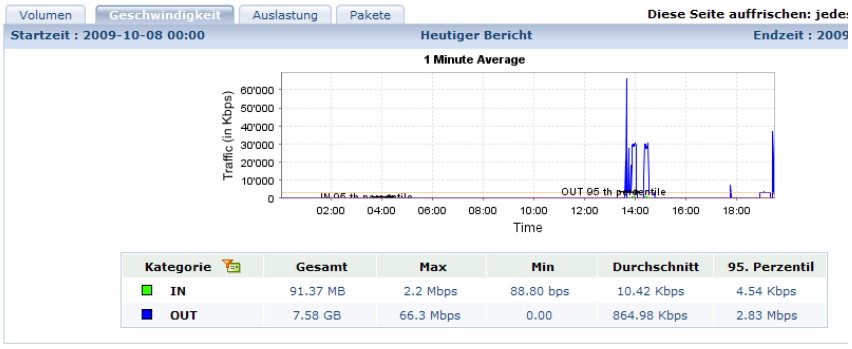


Abbildung 6-9 NetFlow Analyzer 7: Übersicht Fa 0/0

6.2.3.2 Pro und Contra

+ 30 Tage voll funktionsfähig

- Nur 30 Tage freie Nutzung
- Sehr schlechte deutsche Übersetzung
- NetFlow Daten können nicht weiterverwendet werden (keine Exportmöglichkeit)

6.2.3.3 Fazit

Ein gutes Programm für den Netzwerkadministrator.
Die NetFlows können wie auch bei Scrutinizer nicht weiter verwendet werden.

6.2.4 Flowalyzer

Flowalyzer ist ebenfalls von Plixer International wie schon Scrutinizer. Diese Software dient vor allem dazu, schnell zu überprüfen, ob NetFlow-Pakete versendet werden. Es listet lediglich auf, von welchem Gerät wie viele NetFlows empfangen werden.

6.2.4.1 Installation

Diese Applikation benötigt keine Installation. Der Netflow-Port muss angegeben werden. Beim Klick auf den Start-Button bindet die Applikation den Port und listet die empfangenen Netflow-Pakete auf.

6.2.4.2 Pro und Contra

+ Keine Installation
+ Kann auch NetFlow-Pakete generieren

- Keine Speicherung
- Minimale Funktionen

6.2.4.3 Fazit

Ein simples Programm um schnell die Kommunikation testen zu können. Sehr leicht lassen sich NetFlow-Pakete generieren und anschauen. Allerdings ist die Ansicht auf Exporter-IP, Hostname, Port, Type und Anzahl Pakete beschränkt.

6.2.5 NFDump / NFSen

NFDump[24] ist ein Linux-Tool auf OpenSource Basis, welches erlaubt, die NetFlow-Pakete zu speichern. Die Speicherung geschieht standardmässig alle 5 Minuten in ein eigenes binäres Format.

NFSen[25] analysiert die Dateien von NFDump und speichert die Grafiken in Dateien, welche von einem Web-Server dargestellt werden können.

Der grosse Vorteil dieser Kombination ist die Trennung der Sammlung der Daten (NFDump) und die Analyse mit der Darstellung (NFSen). Diese Applikationen können bei einem grossen System auf verschiedenen Servern laufen. Bei NFDump ist lediglich der Diskspeicher die Limitierung.

Auch NFDump kann für die Analyse benutzt werden, was im Kapitel 6.2.6 genauer beschrieben wird.

6.2.5.1 Installation

Die Installation unter Windows ist nicht vorgesehen, daher haben wir auf eine Ubuntu-Installation in einer virtuellen Umgebung zurückgegriffen.

NFDump und NFSen benötigen PERL, FLEX und einen laufenden Apache-Webserver für die Darstellung.

```
apt-get update
apt-get install flex bison librrd2-dev
apt-get install librrds-perl libmailtools-perl
```

Um NFDump zu installieren sind folgende Schritte nötig:

```
sudo ./configure –-prefix=/opt/nfdump --enable-nfprofile
make
make install
```

Damit NFSen installiert werden kann, muss zuerst ein Konfigurationsfile erstellt werden, unten sind die Änderungen zum Standard-Konfigurationsfile aufgelistet:

```
$BASEDIR = "/opt/nfsen";
$HTMLDIR = "/var/www/nfsen/";
# nfdump tools path
$PREFIX = '/opt/nfdump/bin';
$USER = "netflow";
$WWWUSER = "www-data";
$WWWGROUP = "www-data";
$SUBDIRLAYOUT = 7;
%sources = (
'router'=> { 'port' => '9999','col' => '#ff0000' },
);
```

Dieses Konfigurationsfile wird unter /etc/nfsen.conf gespeichert. Ein User „netflow“ wird kreiert und die Gruppe „www-Data“ gepackt. Anschliessend kann das Installations-Script aufgerufen werden:

```
./install.pl /etcnfsen.conf
```

NFSen wird mit folgendem Befehl gestartet:

```
sudo /opt/nfsen/bin/nfsen start
```

Damit die generierten Seiten von NFSen angezeigt werden können, muss der Apache-Webserver mit dem zugehörigen PHP-Modul installiert werden.

```
apt-get install apache2
apt-get install php5
```

Mit einem Dateieditor müssen folgende Einträge in den Konfigurationsfiles erstellt, resp. angepasst werden:

```
sudo nano /etc/apache2/conf.d/fqdn
ServerName localhost
```

```
sudo nano /etc/apache2/sites-enabled/00-default
DocumentRoot /var/www/nfsen
```

Der Apache lässt sich anschliessend starten:

```
sudo /etc/init.d/apache2 start
```

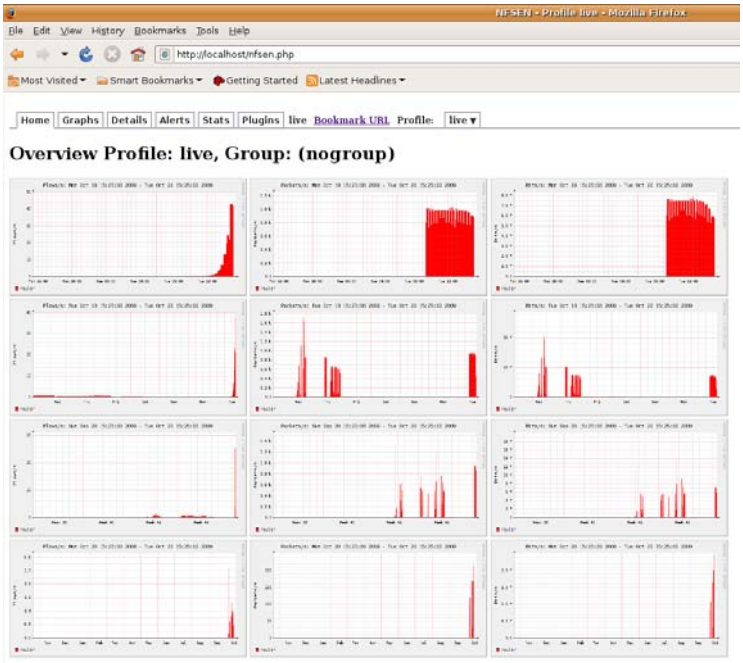


Abbildung 6-10 NFSen-Page mit Apache

6.2.5.2 Pro und Contra

+ NFSen speichert NetFlows als Binaries im 5 Minutentakt
+ Binaries können mit NFDump weiterverarbeitet werden
+ Komplette Webseite wird automatisch erstellt
+ Viele Funktionen

- Aufwändige Installation / Konfiguration

6.2.5.3 Fazit

Diese beiden Applikationen sind ideal für unsere Messungen. NFDump lässt sich über die Kommandozeile bedienen und kann somit in Scripts eingebaut werden.

6.2.6 NFDump für die Analyse

NFDump lässt sich auch dafür verwenden, dass die gespeicherten Binaries, in welchen die NetFlows gespeichert sind, analysiert werden können.

6.2.6.1 Installation

Da NFDump schon als NetFlow-Kollektor installiert ist, können die Optionen einfach beim Aufruf als Parameter mitgegeben werden.

NFDump liest aus einem bestehenden Binary-File und stellt die NetFlows in der Konsole dar:

```
./nfdump
–r /opt/nfsen/profiles-data/live/router/2009-10-
07/nfcapd.200910071315
–o long
```

Es lassen sich auch alle Files von z.B. einem ganzen Tag einlesen:

```
./nfdump –R /opt/nfsen/profiles-data/live/router/2009-10-07 –o
long
```

Die Ausgabe auf der Konsole lassen sich in ein File umleiten:

```
> /home/hsr/Desktop/flows
```

6.2.6.2 Fazit

NFDump kann sowohl die NetFlows in den Binaries speichern und diese wieder bearbeiten und z.B. auf der Konsole darstellen. Diese Applikation ist vielseitig einsetzbar.

6.2.7 Auswertung

Nach Punkten ist der Testsieger in dieser Softwaregruppe die Kombination von NFDump und NFSen. Diese beiden Applikationen erfüllen zusammen die meisten Anforderungen.

Aus diesem Grund wird für die Auswertung der NetFlows auch NFDump verwendet. NFSen könnte auch weggelassen werden, da das Interesse am Exportieren der Flows in eine Textdatei besteht. Trotzdem ist NFSen hilfreich, wenn es darum geht, die ankommenden NetFlow-Pakete schnell zu überblicken.

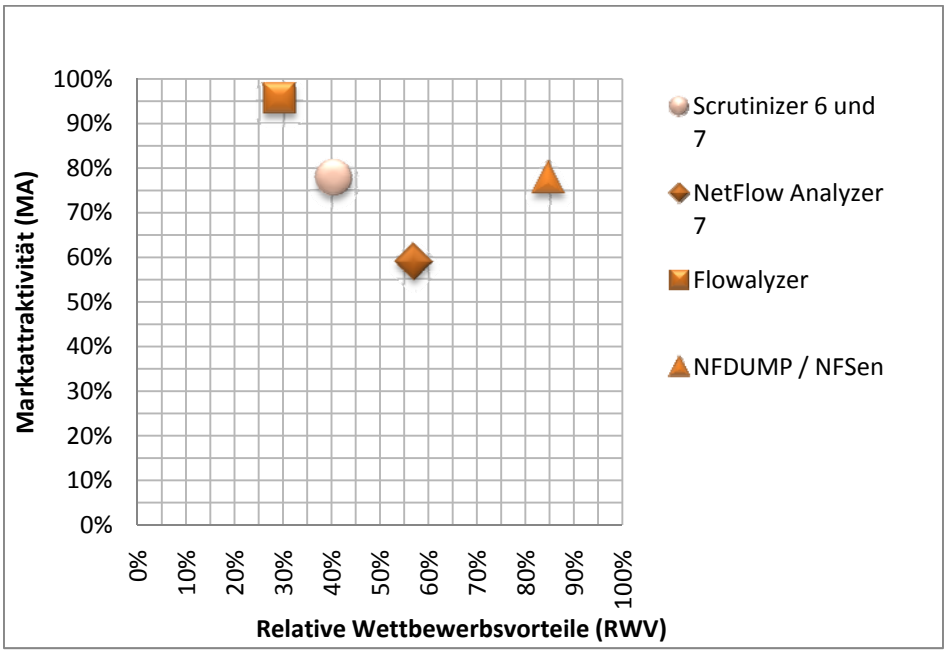


Abbildung 6-11 Auswertung NetFlow Analyzer

Die detaillierte Auswertung ist in der Portfolio Analyse[26] zu finden.

6.3 Auswahl der Flow Generator Software

Ein Flow Generator berechnet aus den aufgezeichneten Netzwerk-Paketen die NetFlows.

6.3.1 pcap2flow

Die Firma Unleash Networks[27] stellt die Software pcap2flow als Open Source bereit. Ursprünglich als interne Test-Applikation gedacht, entschied sich die Firma, die Source für die Allgemeinheit freizugeben.
Die Installation benötigt zwingend eine Linux-Umgebung.

Die einzige Funktion, welche die Software beherrscht, ist das Senden von NetFlow-Pakete an einen Kollektor anhand der aufgezeichneten Verbindungen in einem PCAP-File.

6.3.1.1 Installation

Die Installation von pcap2flow ist wie bei den meisten Unix-Programmen:

```
./configure
make
make install
```

Um die NetFlow-Pakete zu erkennen, benötigt pcap2flow den Port für den NetFlow-Kollektor als Parameter, welcher identisch sein muss wie der Port im PCAP-File. Weiter kann der Abstand zwischen den zu versendenden NetFlow-Paketen gesetzt werden, was nicht funktioniert.

Der entsprechende Aufruf ist wie folgt:

```
Pcap2flow sample_netflow.pcap 192.168.0.7 9999 –ports 9999 –gaus
1000
```

6.3.1.2 Pro und Contra

+ Liest NetFlow-Pakete aus PCAP-Files

- Kleiner Funktionsumfang
- Eingestellter Abstand zwischen NetFlow-Pakete funktioniert nicht

6.3.1.3 Fazit

Diese Applikation ist wie ein fehlendes Puzzle-Teil, welches noch gefehlt hat, da noch keine Applikation von einer PCAP-Datei die NetFlows ermitteln konnte.

6.3.2 Softflowd

Softflowd ist eine Open Source Applikation. Es kann unter mindrot.org[28] heruntergeladen werden.

Softflowd kann NetFlows aus PCAP-Files generieren oder anhand der ankommenden und abgehenden Verbindungen der Netzwerkkarte erstellen.
Ein Anwendungsbeispiel ist ein Netzwerkgerät, welches Linux oder OpenBSD nutzt und mit dieser Software fähig wird, NetFlows zu versenden. Somit ist diese Konstellation eine Alternative zu einem Cisco-Gerät.

6.3.2.1 Installation

Diese Software benötigt zusätzlich die PCAP-Library und wird folgendermassen installiert:

```
apt-get install libpcap-dev
./configure –prefix=/opt/softflowd
make
make install
```

Entweder kann Softflowd die NetFlow-Pakete von einem PCAP-File auslesen oder die NetFlows werden aus dem vorhandenen Netzwerkverkehr errechnet und verschickt.

```
sudo /opt/softflowd/sbin/softflowd –r sample_netflow.pcap
–n 192.168.0.7:9999

sudo /opt/softflowd/sbin/softflowd –v 5 –i eth2 –n
192.168.0.7:9999
```

Mit dem Parameter `–D` kann in den Debug-Modus gewechselt werden. Es kann verfolgt werden, wann ein Flow erstellt und beendet ist und zu welcher Zeit ein NetFlow-Paket versendet wird.

6.3.2.2 Pro und Contra

+ Liest NetFlows aus einem PCAP-File
+ Generiert NetFlow-Pakete anhand des Verkehrs auf Netzwerkkarte

6.3.2.3 Fazit

Ein weiteres Programm für die Generierung der NetFlow-Pakete aus einem PCAP-File.

6.3.3 Auswertung

Zu Beginn war geplant die PCAP-Dateien mit einer Software aus dieser Gruppe zu analysieren.

Bevor jedoch die oben genannten Applikationen benutzt werden können, müsste getestet werden, was ein NetFlow-Kollektor mit den gesendeten NetFlow-Paketen durchführt. Die offenen Punkte sind:

- Welche Timestamps werden gesetzt
- Ordnet der NetFlow-Kollektor die neuen NetFlows in die bestehenden ein
- Werden die bereits erstellten Diagramme neu angepasst, wenn neue NetFlows mit einem älteren Datum empfangen werden
- Ist das NetFlow-Paket standardkonform
- Parameter für NetFlow-Export müssen den Einstellungen des Routers entsprechen

Um die NetFlows miteinander zu vergleichen, müssen die exakten Exporteinstellungen des Routers auch in der Software eingestellt werden können, was von der Software nicht vorgesehen ist.
NFDump kann die NetFlows in einem Textdatei exportieren, welches wieder in Excel importiert werden kann. Excel bietet die Möglichkeit ohne grossen Aufwand Tabelleneinträge zu filtern. Deswegen fällt der Entscheid auf Excel.

6.4 Auswahl der Traffic Generator Software

Traffic Generatoren erzeugen diverse Netzwerkpakete, die von einem Sender zu einem Empfänger geschickt werden. Oft werden sie eingesetzt, um Netzwerkkomponenten in Testumgebungen einem Belastungstest zu unterziehen. Es gibt auch Traffic Generatoren die spezifische Applikationspakete versenden. Diese Pakete können genutzt werden, um den Netzwerkverkehr zu priorisieren und so das gesamte Netzwerk zu optimieren.

Die aufgelisteten Traffic Generatoren werden angeschaut, ob sie für diese Zwecke geeignet sind. Die Auswertung ist im Kapitel 0 beschrieben.

6.4.1 NSASoft Traffic Emulator

NSASoft[29] stellt aus dem NSAuditor diverse kleine Applikationen zur freien Verfügung. Zu diesen Freewaretools gehört auch der Traffic Emulator.

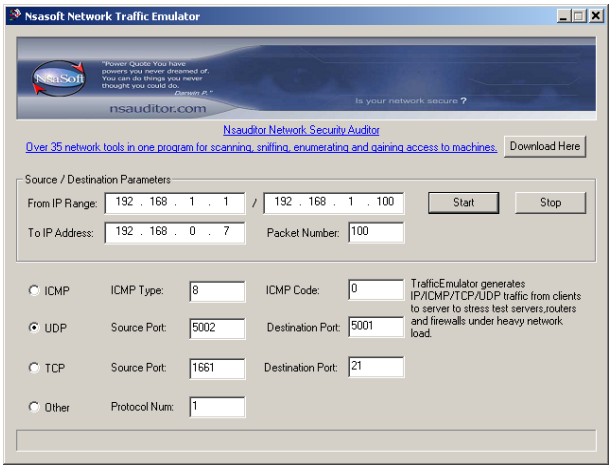


Abbildung 6-12 NSASoft Traffic Emulator

6.4.1.1 Installation

Die Installation wird mit Hilfe eines Wizards geleitet. Es muss nur der Installationspfad und die Verknüpfungen angegeben werden. Die Dateien und Registrierungseinträge werden automatisch vorgenommen.

6.4.1.2 Bedienung

Zuerst müssen die Source und Destination Parameter gesetzt werden. Dazu wird ein IP-Range bei der Source und eine IP bei der Destination angegeben. Die Anzahl der zu versendenden Pakete muss ebenfalls eingetragen sein.
Es kann zwischen ICMP, TCP oder UDP ausgewählt und die jeweiligen Parameter eingestellt werden.

6.4.1.3 Pro und Contra

Der Traffic Emulator von NSASoft liefert alles Nötige für die Erzeugung von Paketen. Leider kann die Paketgrösse oder Sendedauer nicht variiert werden. Zusammengefasst ist der Traffic Emulator ein kleiner Paketgenerator, der die grundlegendsten Eigenschaften unterstützt.

+ Unterstützt drei Protokolle (UDP, TCP, ICMP)
+ Protokollnummer im IP-Header wählbar
+ Portnummer für UDP oder TCP einstellbar
+ Anzahl der zu sendenden Pakete wählbar
+ Freeware
+ Einfache Installation, Begleitung durch Wizard

- Nicht skriptfähig
- Payload nicht variierbar
- Keine variierbaren Sendezeiten von Paketen
- Keine parallelen Verbindungen zu Empfänger möglich

6.4.2 LANforge Fire

LANforge[30] von Candela Technologies bietet ein TrafficGenerator welcher sehr unterschiedliche Pakete generieren kann. Es können FileIO, VoIP-Transfers oder auch simple Crosstransfers erstellt werden.

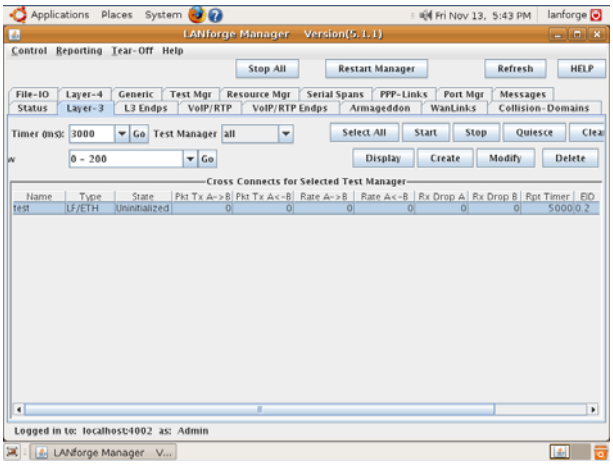


Abbildung 6-13 NSASoft Traffic Emulator

6.4.2.1 Installation

Die Installation wird bei einem virtuellen Computer mit dem Betriebssystem Ubuntu durchgeführt. Bei der Installation kommt es zu Problemen mit den Ubuntu-Installationspaketen oder dem Kernel.
LANforge bietet als weitere Möglichkeit eine bootfähige Live CD, welche vom Internet zur freien Verfügung steht. Von der Boot-CD aus kann die Demoversion benutzt werden.

6.4.2.2 Bedienung

Das Tool liefert ein User-Interface, welche typische Charakterzüge von Java aufweist. In der Demoversion können nur die simplen Generatoren genutzt werden. Dabei können einfache Verbindungen mit den beiden Layer 4 Protokollen UDP und TCP erstellt werden. Zusätzliche Funktionen um Daten über eine serielle Schnittstelle oder eine PPP-Verbindung können simuliert werden. Für die Messungen sind diese Funktionen nicht von Nutzen.
Die Erstellung von Netzwerkverkehr verlangt fundiertes Wissen über den Paketaufbau und ein genaues Einlesen in die Bedienungsanleitung.

6.4.2.3 Pro und Contra

LANforge Fire ist ein vielversprechendes Tool. Jedoch kann die Demoversion keine komplexen Pakete generieren.

+ LANforge Fire automatisierbar
+ Unterstützt Layer 4 Protokolle (UDP, TCP)
+ Wählbare Portnummer für UDP oder TCP
+ Payload einstellbar und variierbar
+ Anzahl der zu sendenden Pakete konfigurierbar
+ Parallele Verbindungen können erstellt werden

- Erzeugt keinen Applikationsspezifischer Netzwerkverkehr
- Sendezeiten der einzelnen Verbindungen nicht einstellbar
- Grosser Installationsaufwand unter Ubuntu, Anleitung existiert
- Komplexe Bedienung, fundierte Paketkenntnisse notwendig

6.4.3 Colasoft Packet Builder

Colasoft[31] stellt mit dem Packet Builder eine Software zur freien Verfügung mit der die einzelnen Headerwerte des Ethernet-Frames, IP-Headers und des Layer 4 Protokoll-Headers gesetzt werden können.

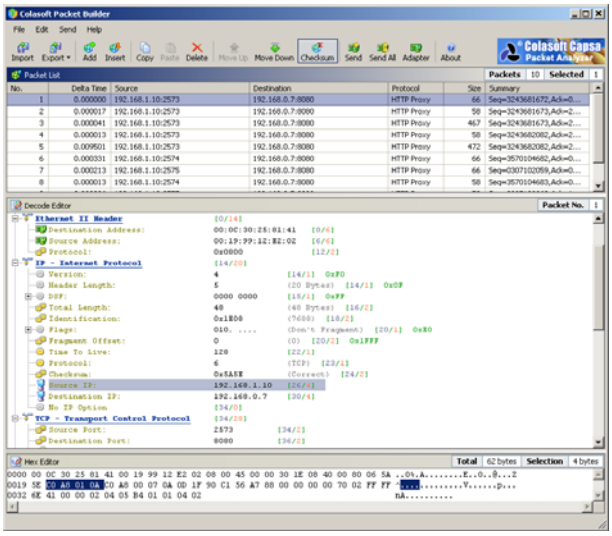


Abbildung 6-14 Benutzeroberfläche vom Colasoft Packet Builder

6.4.3.1 Installation

Die Installation wird mit einem Wizard unterstützt. Darin muss nur der Installationspfad festgelegt werden. Alle benötigten Dateien, sowie auch die Registrierungseinträge werden automatisch gespeichert und registriert.

6.4.3.2 Bedienung

Für die Applikation gibt es eine kleine Dokumentation, welche die einzelnen Screens beschreibt. Für die Erstellung von Paketen sind fundierte Netzwerkkenntnisse notwendig, weil jeder einzelne Wert im Ethernet-Frame, IP-Frame und Transportprotokolls eingestellt werden muss.

Colasoft Packet Builder bietet die Möglichkeit Datenpakete, welche mit Wireshark aufgezeichnet wurden, in die Software zu importieren und von dort auszuführen.

6.4.3.3 Pro und Contra

Mit dem Colasoft Packet Builder kann jeder Parameter in den einzelnen PDUs verändert werden. Es wird viel Zeit für die Erstellung von Paketen benötigt. Dadurch, dass Capturefiles von Wireshark importiert werden können, kann jegliche Art von Netzwerktraffic erstellt werden. Der Aufwand ist aber auch dem entsprechend gross und es existiert kein Server der alle Netzwerkpakete entgegen nimmt.

+ Unterstützt Layer 4 Protokolle
+ Portnummer für UDP oder TCP einstellbar
+ Erzeugt Applikationsspezifischer Netzwerkverkehr
+ Payload pro Paket einstellbar
+ Sendezeit pro Paket einstellbar
+ Colasoft Packet Builder frei erhältlich
+ Einfache Installation

- Keine parallelen Verbindungen möglich
- Importe aus Wireshark müssen exakt gefiltert werden
- fundierte Paketkenntnisse notwendig

6.4.4 Iperf

Iperf[32] ist eine Kommandozeilenapplikation, welches frei erhältlich ist. Es wurde von NLANR/DAST[33] entwickelt, welche es nicht mehr weiterführt. Statt dessen gaben sie den Code frei. Eine stabile Version der Software ist die Version 1.7.0, welche bei Softpedia[34] erhältlich ist.

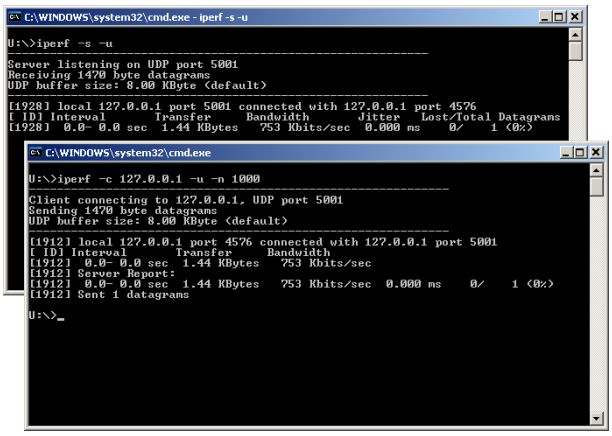


Abbildung 6-15 Iperf im Commandfenster

6.4.4.1 Installation

Unter Windows muss die Datei "iperf.exe" ins Verzeichnis "C:\Windows\System32" hinein kopiert werden. Falls nicht muss die Path-Variable mit dem Verzeichnispfad von "iperf.exe" ergänzt werden. Die Datei kann dann von jedem Pfad aus gestartet oder verwendet werden.

Bei Ubuntu kann im Terminal der `Code apt-get install iperf` eingegeben werden. Das Installationspaket wird vom Internet heruntergeladen und installiert.

6.4.4.2 Bedienung

Die Bedienung ist anfangs ein wenig gewöhnungsbedürftig. In einem Kommandofenster kann mit `iperf -c` für Client und mit `iperf -s` für Server die entsprechende Applikation gestartet werden.
Zudem kann mit `iperf /?` die zugehörenden Parameter eingesehen werden. So kann zwischen den Transportprotokollen TCP und UDP ausgewählt werden und die Belastung per Zeit oder Sendegrösse eingestellt werden.

6.4.4.3 Pro und Contra

Iperf lässt sich durch die Commandline Parametern gut einstellen. Es wird vorwiegend bei Durchsatzgeschwindigkeiten benutzt. Diese werden am Ende einer Messung angezeigt.

+ Skriptfähig
+ Unterstützt Layer 4 Protokolle
+ Portnummer für UDP oder TCP einstellbar
+ Anzahl Pakete kann in Sekunden angegeben werden
+ Payload pro Paket einstellbar
+ Sendezeit per Script einstellbar
+ Gratis
+ Einfache Installation
+ Parallele Verbindungen möglich

- Erzeugt keinen Applikationsspezifischen Netzwerkverkehr
- Keine intuitive Bedienung

6.4.5 Traffic 0.1.3

Traffic 0.1.3[35] wurde von Robert Sandilands entwickelt, um das Verhalten von Routern und Firewalls unter grösserer Last zu testen. Er stellt sein Tool unter die GPL Lizenz und somit ist es frei erhältlich.

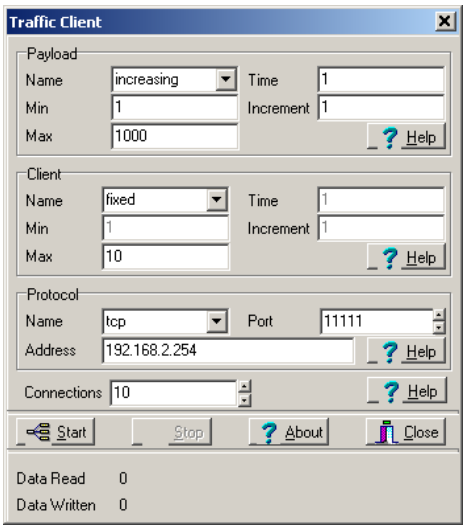


Abbildung 6-16 Benutzeroberfläche von Traffic 0.1.3

6.4.5.1 Installation

Die Installation von Traffic 0.1.3 wird über einen Wizard gesteuert und kopiert alle Dateien und Verknüpfungen automatisch auf den Computer. Die Registrierungseinträge werden ebenfalls automatisch vorgenommen. Es gibt eine Windows und eine Linux-Version.

6.4.5.2 Bedienung

Die Bedienung wird über die Benutzeroberfläche gesteuert. Man kann entweder den Traffic Client starten, bei dem die Pakete erzeugt und gesendet werden oder den Traffic Server, welcher die Datenpakete empfängt.

6.4.5.3 Pro und Contra

Traffic 0.1.3 ist ein guter Helfer, um Netzwerklast zu generieren.

+ Unterstützt Layer 4 Protokolle
+ Portnummer für UDP oder TCP einstellbar
+ Payload pro Paket einstellbar
+ Gratis
+ Einfache Installation
+ Parallele Verbindungen möglich
+ Bedienung ist intuitiv

- Nicht skriptfähig
- Erzeugt keinen Applikationsspezifischen Netzwerkverkehr
- Anzahl Pakete nicht einstellbar, sendet konstanten Verkehr
- Sendezeit nicht einstellbar

6.4.6 D-ITG

D-ITG[36] heisst Distributed Internet Traffic Generator und ist von der Universita degli Studi di Napoli ''Federico II'' in Italien entwickelt worden. Eine Benutzeroberfläche wurde von Volmer Semken in Java entwickelt und wird ebenfalls zur Verfügung gestellt.

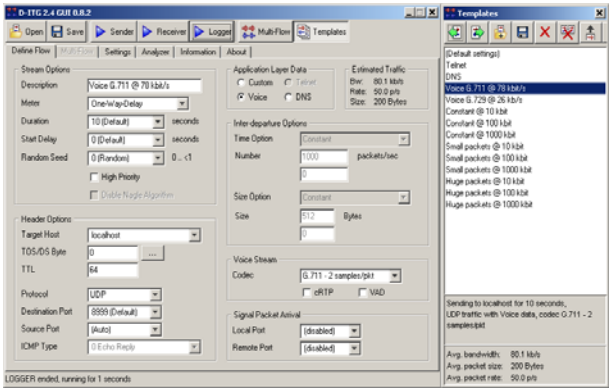


Abbildung 6-17 Benutzeroberfläche von D-ITG

6.4.6.1 Installation

Auf der Internetseite werden die Binary Files der Applikation zum Download angeboten und auf die Website von Volmer Semken für das GUI verwiesen.

Die Dateien werden in ein Verzeichnis hinein kopiert. Der Verzeichnispfad muss der Path-Variable hinzugefügt werden. Die Dateien für das GUI werden ebenfalls ins gleiche Verzeichnis kopiert. Danach lässt sich der Traffic Generator von jedem Pfad in der Kommandozeileneingabe starten.

6.4.6.2 Bedienung

Die Bedienung mit dem GUI ist intuitiv gestaltet. Es können schnell Pakete versendet werden. Vom GUI kann direkt der Empfänger gestartet werden. Es existieren schon vorgefertigte Paketvorlagen welche eingesetzt werden können.

Mehr Möglichkeiten als mit dem GUI kann mit der Kommandozeileneingabe realisiert werden. Es können ähnlich wie in Iperf diverse Parameter gesetzt werden. Wenn mehr als ein Paket gesendet werden soll, kann eine Multiflowdatei erzeugt und abgearbeitet werden. Diese lässt sich nach Belieben erweitern. Ebenfalls kann auch eine Batchdatei eingesetzt werden um die Paketsendung zu automatisieren.

Ein Logfile, welches beim Senden generiert wird, kann ausgewertet werden und liefert auch die gesendeten Flows. Nicht aufgezeichnet wird eine TCP Verbindung vom Client zum Server, welche zur Vorbereitung der Socket beim Server verwendet wird.

6.4.6.3 Pro und Contra

D-ITG lässt sich gut verwenden und ist vielseitig einstellbar. Die Auswertung liefert die Dauer der einzelnen Flows und gibt am Schluss eine Übersicht über alle gesendeten Flows

+ Skriptfähig
+ Unterstützt Layer 4 Protokolle
+ Portnummer für UDP oder TCP einstellbar
+ Erzeugt keinen applikationsspezifischen Netzwerkverkehr
+ Anzahl Pakete wählbar
+ Payload pro Paket einstellbar
+ Sendezeit nicht einstellbar
+ Gratis
+ Einfache Installation
+ Parallele Verbindungen möglich
+ Intuitive GUI Bedienung

6.4.7 Auswertung der Traffic Generatoren

In unseren Labs wird ebenfalls ein Traffic Generator eingesetzt.
Zur Analyse der Traffic Generatoren werden die wichtigsten Funktionen aufgenommen. Diese sind in der Portfolio Analyse zu sehen. Die Bewertung entspricht der gegebenen Vielseitigkeit der Kriterien. Bei den Killerkriterien sind die Bewertungen wie folgt spezifiziert worden.

Scriptfähigkeit / Automatisierbarkeit

Kriterien	Bewertung
Pakete in Endlosschleife senden	3
Definierte Wiederholungen	3
Unterschiedliche Paketfolge	3
Unterbrechung / Pause in fortlauf	1

Tabelle 6-2 Kriterium Scriptfähigkeit

Unterstützte Protokolle

Kriterien	Bewertung
Protokoll UDP	3
Protokoll TCP	3
Protokoll ICMP	2
Andere Protokolle	2

Tabelle 6-3 Kriterium Layer-4-Protokolle

Portnummer

Kriterien	Bewertung
Eine fixe Portnummer	5
Variierbare Portnummern	10

Tabelle 6-4 Kriterium Portnummern

Parallele Verbindungen

Kriterien	Bewertung
Keine parallelen Verbindungen	1
<10 parallele Verbindungen	3
<100 parallele Verbindungen	6
>100 parallele Verbindungen	10

Tabelle 6-5 Kriterium Parallele Verbindungen

Mit den getroffenen Kriterien und Bewertungen stechen die zwei Applikationen D-ITG und Iperf hervor. Sie erfüllen alle Killerkriterien und fügen weitere Funktionen hinzu. Die beiden Applikationen werden in den Labs verwendet, um Pakete zu senden.

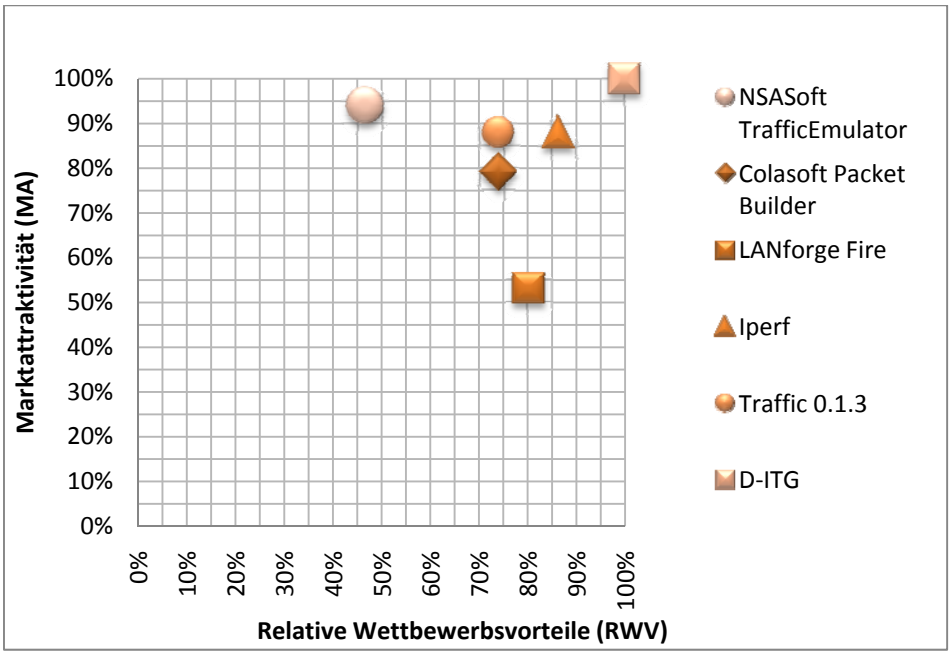


Abbildung 6-18 Portfolio Traffic Generator Analyse

Die detaillierte Auswertung ist in der Portfolio Analyse[37] zu finden.

7 TRAFFICMODELLE

Traffic Modelle sollen den typischen Netzwerkverkehr eines Users definieren. Dabei werden die meist benutzten Applikationen des entsprechenden Users berücksichtigt. Für den Test werden zwei unterschiedliche Modelle verwendet. Eines für den Home-User und eines für den Business-User. Die Angaben in Prozent entsprechen dem ungefähren Datenvolumen.

7.1 Modell Home-User

Der Home-User surft regelmässig auf dem Computer oder Notebook und nutzt die gängigen Internettechnologien, für seine Freizeit oder Unterhaltung. Er verwendet ebenfalls den Computer, um mit seinen Verwandten, Kollegen oder Freunden in Kontakt zu stehen.

Anwendung / Technologie	Prozentualer Anteil
Streaming (Bsp: Youtube)	30 %
Filesharing	25 %
Webtraffic (Bsp: Facebook)	20 %
Skype	10 %
Mail	5 %
Games	5 %
E-Banking	3 %
Services (Bsp: Virenscanner, Windowsupdate)	2 %

Tabelle 7-1 Traffic eines Homeusers

Die Werte unter dem prozentualen Anteil werden aus unseren eigenen Erfahrungen zusammengestellt.

7.2 Modell Business-User

Der Business-User arbeitet in einem modernen Unternehmen. Sie nutzen VoIP und Webmeetings für ortsunabhängige Konferenzen. Die Computer werden zur Informationsbeschaffung und Informationsaustausch verwendet.

Anwendung / Technologie	Prozentualer Anteil
SMB (Bsp: Fileserverzugriffe)	35 %
Webtraffic	20 %
Webmeeting (Bsp: Office Communicator)	10 %
Application Server (Bsp: SAP, DB)	10 %
Mail (Bsp: Exchange)	10 %
Telefon (Bsp: VoIP)	10 %
Services (Bsp: Virenscanner, WindowsUpdate, Telnet, RDP)	5 %

Tabelle 7-2 Traffic eines Businessusers

Die Werte unter dem prozentualen Anteil werden aus unseren eigenen Erfahrungen zusammengestellt.

7.3 Traffic Mix Realisierung

Der Traffic Mix kann mit Hilfe des Tools D-ITG erstellt werden. Insbesondere kann Netzwerkverkehr für Spiele, VoIP, Telnet und DNS mit den entsprechenden Charakteristiken gesendet werden.
Der Netzwerkverkehr der einzelnen Applikationen wurde nach Protokoll- und Porteigenschaften betrachtet. Die Pakete wurden in einem Multiflow-File für D-ITG zusammengestellt.

Die Datei "trafficmix-businessuser_mitComment.itg" zeigt die einzelnen Flows eines Business Users an. Zur einfacheren Auswertung wurde die gesamte Sendedauer auf 1 Minute herunter gebrochen. Dabei entstanden folgende Sendezeiten:

Traffic	Prozent	Sekunden
SMB	35%	21
Streaming	30%	18
Filesharing	25	15
Webtraffic	20%	12
Skype, Webmeeting, Application Server, Mail, VoIP	10%	6
Services	5%	3
E-Banking	3%	1.8
Services	2%	1.2

Tabelle 7-3 Verteilung Traffic Mix

trafficmix-businessuser_mitComment.itg

```
// 35% SMB (Fileserver) 1)TCP:445
-a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
-a 192.168.3.10 -T TCP -rp 445 -t 20920 -d 520

// 20% Webtraffic 1)UDP/DNS:53 2)TCP/HTTP:80
-a 192.168.3.10 -rp 53 -T UDP -t 20 -d 21940 DNS
-a 192.168.3.10 -rp 80 -T TCP -t 380 -d 22460
-a 192.168.3.10 -rp 80 -T TCP -t 6000 -d 23340
-a 192.168.3.10 -rp 80 -T TCP -t 2000 -d 29840
-a 192.168.3.10 -rp 80 -T TCP -t 600 -d 32340
-a 192.168.3.10 -rp 80 -T TCP -t 1000 -d 33440
-a 192.168.3.10 -rp 80 -T TCP -t 1500 -d 34940
-a 192.168.3.10 -rp 80 -T TCP -t 500 -d 36940

// 10% Webmeeting 1)HTTP:80, TCP:2000, TCP/HTTPS:443
-a 192.168.3.10 -rp 80 -T TCP -t 20 -d 37940
-a 192.168.3.10 -rp 443 -T TCP -t 20 -d 38460
-a 192.168.3.10 -rp 2000 -T TCP -t 5960 -d 38980

// 10% Applikationsserver (SAP DB)
-a 192.168.3.10 -rp 53 -T UDP -t 20 -d 45440 DNS
-a 192.168.3.10 -rp 3306 -T TCP -t 5980 -d 45960
```

```
// 10% Mail(Exchange) 1)LDAP TCP:389, 2)IMAP TCP:143 3)POP3
TCP:110 4)NNTP TCP:119 5)SMTP TCP:25
-a 192.168.3.10 -rp 389 -T TCP -t 20 -d 52440
-a 192.168.3.10 -rp 143 -T TCP -t 4000 -d 52960
-a 192.168.3.10 -rp 119 -T TCP -t 980 -d 57460
-a 192.168.3.10 -rp 25 -T TCP -t 1000 -d 58940

// 10% Telefon(VoIP)
-a 192.168.3.10 -rp 53 -T UDP -t 20 -d 60440 DNS
-a 192.168.3.10 -rp 5000 -t 5980 -d 60960 VoIP -x G.711.2 -h RTP -
VAD

//5% Services (NTP/WindowsUpdate/Antivirus/Telnet/RemoteDesktop)
-a 192.168.3.10 -rp 123 -t 100 -d 67440
-a 192.168.3.10 -rp 23 -t 400 -d 67440
-a 192.168.3.10 -rp 23 -t 500 -d 67440
-a 192.168.3.10 -rp 23 -t 1000 -d 67440 Telnet
-a 192.168.3.10 -rp 3389 -t 2000 -d 68940
```

Das Senden der Pakete verursacht im Multiflow-File diverse Abstürze. Eine Regelmässigkeit wird nicht festgestellt. Deshalb werden die Pakete einzeln in einem Batchscript aufgerufen.

lab7_business.cmd

```
taskkill /IM ITGSend.exe /T /F

set file="lab7_business_01.txt"

echo Messstart Lab 7 >> %file%
echo ======================================== >> %file%
echo %date% >> %file%
echo %time% >> %file%

echo // 35% SMB (Fileserver) 1)TCP:445
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -T TCP -rp 445 -t 20920

echo // 20% Webtraffic 1)UDP/DNS:53 2)TCP/HTTP:80
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 380
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 6000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 2000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 600
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 1000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 1500
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 500

echo // 10% Webmeeting 1)HTTP:80, TCP:2000, TCP/HTTPS:443
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 20
itgsend -a 192.168.3.10 -rp 443 -T TCP -t 20
itgsend -a 192.168.3.10 -rp 2000 -T TCP -t 5960

echo // 10% Applikationsserver (SAP DB)
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -rp 3306 -T TCP -t 5980
```

```
echo // 10% Mail(Exchange) 1)LDAP TCP:389, 2)IMAP TCP:143 3)POP3
TCP:110 4)NNTP TCP:119 5)SMTP TCP:25
itgsend -a 192.168.3.10 -rp 389 -T TCP -t 20
itgsend -a 192.168.3.10 -rp 143 -T TCP -t 4000
itgsend -a 192.168.3.10 -rp 119 -T TCP -t 980
itgsend -a 192.168.3.10 -rp 25 -T TCP -t 1000

echo // 10% Telefon(VoIP)
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -rp 5000 -t 5980 VoIP -x G.711.2 -h RTP -
VAD

echo //5% Services
(NTP/WindowsUpdate/Antivirus/Telnet/RemoteDesktop)
itgsend -a 192.168.3.10 -rp 123 -T UDP -t 100
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 400
itgsend -a 192.168.3.10 -rp 443 -T TCP-t 500
itgsend -a 192.168.3.10 -rp 23 -t 1000 Telnet
itgsend -a 192.168.3.10 -rp 3389 -T TCP -t 2000

echo %date% >> %file%
echo %time% >> %file%
echo ======================================== >> %file%
echo Messende Lab 7 >> %file%
```

Die Zusammenstellung für den Home User ist ebenfalls in einem Batchscript eingefügt worden.

lab7_home.cmd

```
taskkill /IM ITGSend.exe /T /F

set file="lab7_home_01.txt"

echo Messstart Lab 7 >> %file%
echo ======================================== >> %file%
echo %date% >> %file%
echo %time% >> %file%

echo // 30% YouTube 1)UDP/DNS:53 2)TCP/HTTP:80
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 17980

echo // 25% BitTorrent nutzt TCP:6881-6999
itgsend -a 192.168.3.10 -rp 6881 -T TCP -t 15000

echo // 20% Webtraffic HTTP:80
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 480
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 6000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 2000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 1000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 1000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 1500
```

```
echo // 10% Skype
itgsend -a 192.168.3.10 -rp 443 -t 6000 VoIP -x G.711.1 -h RTP

echo // 5% Mail 1)IMAP TCP:143 3)POP3 TCP:110 5)SMTP TCP:25
itgsend -a 192.168.3.10 -rp 53 -t 20 DNS
itgsend -a 192.168.3.10 -T TCP -rp 110 -t 1980
itgsend -a 192.168.3.10 -T TCP -rp 25 -t 1000

echo // 5% Game
itgsend -a 192.168.3.10 -rp 53 -t 20 DNS
itgsend -a 192.168.3.10 CSa -t 2580
itgsend -a 192.168.3.10 CSi -t 400

echo // 3% e-Banking
itgsend -a 192.168.3.10 -rp 53 DNS -t 20
itgsend -a 192.168.3.10 -rp 443 -T TCP -t 1780

echo // 2% Services(NTP/WindowsUpdate/Antivirus)
itgsend -a 192.168.3.10 -rp 123 -T UDP -t 100
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 300
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 800

echo %date% >> %file%
echo %time% >> %file%
echo ======================================== >> %file%
echo Messende Lab 7 >> %file%
```

8 LAB 1 – CPU-BELASTUNG DURCH NETFLOW-EXPORT

8.1 Aufgabenstellung

8.1.1 Ziel

Es ist zu ermitteln, welche CPU-Ressourcen die Aktivierung von NetFlow-Export auf dem Router verbrauchen. Zu unterscheiden ist der Datenverkehr mit UDP und TCP.

Der Datenverkehr wird mit Iperf erzeugt. In der ersten Phase wird mit TCP ermittelt, welche maximale Datenrate möglich ist, damit diese für UDP angewendet werden kann.

Jede Messung wird je eine Stunde lang durchgeführt. Über SNMP wird regelmässig der Wert der CPU-Belastung ausgelesen (über die letzten 5 Sekunden).

8.1.2 Bedingungen

Die Netzwerkinterfaces des Routers sind auf 10 Mbit/s zu konfigurieren, damit die CPU des Routers nicht allzu fest beansprucht wird, wie wenn der Verkehr mit 100 Mbit/s generiert wird.

8.1.3 Risiken / Challenges

Falls der Router die NetFlow-Daten hardwaremässig berechnen kann, ist es möglich, dass die Unterschiede zu klein sind, um gemessen werden zu können. Die Genauigkeit der CPU-Auslastung kann nur in Ganzzahlen (1-100) abgefragt werden, was der prozentualen Auslastung entspricht.

8.2 Konfiguration

8.2.1 Aufbau

Der Router wird mit 4 zusätzlichen Interfaces (Ethernet, 10 MBit/s) bestückt, damit die Abfragen über SNMP über ein separates Interface (Ethernet 1/0) durchgeführt werden können.

Vom PC-Client werden Datenströme zum PC-Server erzeugt. Diese Datenströme sind entweder UDP oder TCP.
Beim Router ist entweder NetFlow aktiviert oder deaktiviert. Dies ergibt also 4 verschiedene Konstellationen.
Vom MessPC aus wird über SNMP-Abfragen die CPU-Auslastung gemessen.

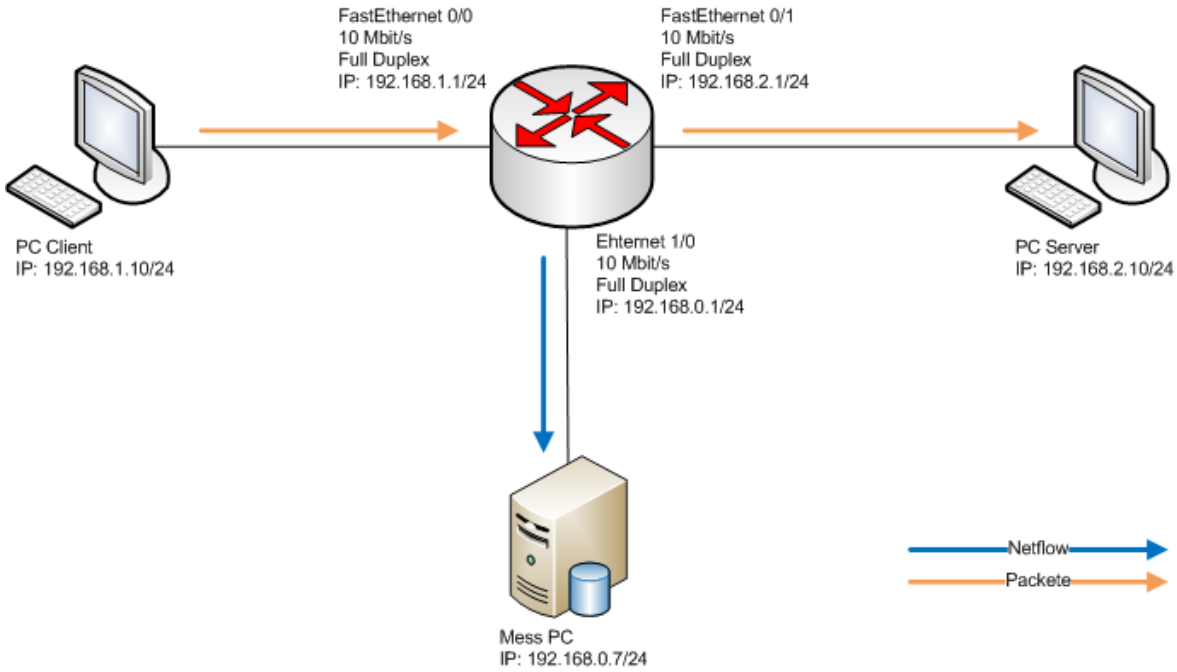


Abbildung 8-1 Aufbau

8.2.2 Router-Konfiguration

Messungen mit NetFlow

Der Router wird mit folgenden Parametern konfiguriert:

```
ip flow-cache timeout active 1
Interface FastEthernet0/1
ip route-cache flow
Interface Ethernet1/0
ip route-cache flow
ip flow-export source FastEthernet0/1
ip flow-export version 5
ip flow-export destination 192.168.0.7 9999
```

Messungen ohne NetFlow

Die NetFlow-Parameter werden entfernt:

```
no ip flow-cache timeout active 1
Interface FastEthernet0/1
no ip route-cache flow
Interface Ethernet1/0
no ip route-cache flow
no ip flow-export source FastEthernet0/1
no ip flow-export version 5
no ip flow-export destination 192.168.0.7 9999
```

8.2.3 PRTG-Konfiguration

Das Programm überwacht mit SNMP-Abfragen im Intervall von 60 Sekunden die beiden folgenden Werte:

CPU-Auslastung (Mittelwert der letzten 5 Sekunden)

```
OID: 1.3.6.1.4.1.9.9.109.1.1.1.1.6.1
```

CPU-Auslastung (Mittelwert der letzten 60 Sekunden)

```
OID: 1.3.6.1.4.1.9.9.109.1.1.1.1.7.1
```

8.2.4 Scripts

Für die Erzeugung des Datenverkehrs wird Iperf verwendet. Dabei wird das Layer 4 Protokoll unterschieden. TCP erhöht die Windowsize fortwährend und kann somit die Bandbreite ausnutzen. Bei UDP muss die zu belegende Bandbreite manuell gesetzt werden.
Anfangs wird ein Test mit 15 TCP Verbindungen durchgeführt und die mittlere Bandbreite ausgewertet. Diese Bandbreitenangabe wird für die UDP-Verbindungen benutzt.

Iperf-Script für TCP Verbindugen:

```
@echo off

echo Messstart TCP > ergebnis_tcp_.txt
echo %date% >> ergebnis_tcp_.txt
echo %time% >> ergebnis_tcp_.txt
echo ======================================== >> ergebnis_tcp_.txt

for /L %%n in (1,1,300) do (
echo Messrunde %%n von 300
time /t >> ergebnis_tcp_.txt
call iperf -c 192.168.2.10 -P 1 -n 10M >> ergebnis_tcp_.txt
)

echo ======================================== >> ergebnis_tcp_.txt
echo Messende >> ergebnis_tcp_.txt
echo %date% >> ergebnis_tcp_.txt
echo %time% >> ergebnis_tcp_.txt
```

Iperf Script für UDP Verbindungen:

```
@echo off

echo Messtart UDP > ergebnis_udp_.txt
echo %date% >> ergebnis_udp_.txt
echo %time% >> ergebnis_udp_.txt
echo ======================================== >> ergebnis_udp_.txt

for /L %%n in (1,1,300) do (
echo Messrunde %%n von 300
time /t >> ergebnis_udp_.txt
call iperf -c 192.168.2.10 -P 1 -u -b 7.19M -n 10M >>
ergebnis_udp_.txt
)

echo ======================================== >> ergebnis_udp_.txt
echo Messende >> ergebnis_udp_.txt
echo %date% >> ergebnis_udp_.txt
echo %time% >> ergebnis_udp_.txt
```

8.3 Testresultate

8.3.1 Erwartet

Es wird erwartet, dass die CPU mehr belastet wird, wenn NetFlow-Export aktiviert ist.

Es sollte keinen Unterschied zwischen TCP und UDP geben.

8.3.2 Gemessen

Die 4 verschiedenen Konstellationen (NetFlow ein oder aus und UDP oder TCP) werden je 3 Mal gemessen und die Resultate in den folgenden Tabellen dargestellt.

Messung 1

TCP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	13.2034	1.1303
NetFlow Aus	10.3729	0.6861
CPU Auslastung durch NetFlow	2.8305	-

Tabelle 8-1: Messung 1 - TCP-Verbindungen

UDP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	9.4746	0.3226
NetFlow Aus	7.1356	0.1882
CPU Auslastung durch NetFlow	2.339	-

Tabelle 8-2: Messung 1 - UDP-Verbindungen

Messung 2

TCP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	14.5593	1.1128
NetFlow Aus	11.0545	0.8303
CPU Auslastung durch NetFlow	3.5048	-

Tabelle 8-3: Messung 2 - TCP-Verbindungen

UDP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	10.5862	0.3195
NetFlow Aus	8.3793	0.2396
CPU Auslastung durch NetFlow	2.2069	-

Tabelle 8-4: Messung 2 - UDP-Verbindungen

Messung 3

TCP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	13.3729	0.6861
NetFlow Aus	11.5636	0.9912
CPU Auslastung durch NetFlow	1.8092	-

Tabelle 8-5: Messung 3 - TCP-Verbindungen

UDP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	9.9298	0.1766
NetFlow Aus	8.7458	0.3308
CPU Auslastung durch NetFlow	1.184	-

Tabelle 8-6: Messung 3 - UDP-Verbindungen

Auswertung

Es fällt auf, dass die Messung 3 andere Resultate aufweist als die anderen Messungen. Dementsprechend haben die Messwerte eine hohe Varianz.

	CPU Auslastung durch NetFlow	Varianz der CPU Auslastung
UDP	1.90998	0.3996
TCP	2.71484	0.7287

Tabelle 8-7: Auswertung CPU Last

8.4 Fazit

Wie erwartet braucht der Router CPU-Ressourcen für die Berechnungen der Flows, allerdings hält sich die CPU-Auslastung in Grenzen. Diese Auslastung wurde höher geschätzt.

Da die Abfrage über SNMP nur alle 60 Sekunden gesendet wird und der Wert der CPU-Auslastung nur die letzen 5 Sekunden betrifft, wird ab der Messung 2 noch der Wert der CPU-Auslastung der letzten Minute ausgelesen. Diese beiden Werte werden verglichen und haben einen Kontrollcharakter. Es könnte per Zufall einen Peak zur Zeit der Abfrage geben, welcher die Messung verfälschen würde.

Die ganze Messung 2 wird wiederholt, da die Bandbreite für das Interface des PC-Clients beim Reboot des Routers wieder den Standardwert bei 100 Mbit/s annahm. Erstaunlicherweise ist die Bandbreite etwa 10% grösser als vorher, obwohl der PC-Server nur 10 Mbit/s verarbeiten kann. Da die Bandbreite auf dem Interface der Serverseite eine Ethernet- und keine FastEthernetschnittstelle ist, dürfte es keinen Unterschied geben, wenn Daten schneller als 10 Mbit/s ankommen, da der Router Pakete verwirft, wenn die Bandbreite und alle Queues ausgelastet sind.
Es wird vermutet, dass der Router diesen Netzwerkverkehr noch zeitlich optimiert. Dieses Phänomen wird nicht weiter betrachtet.

8.4.1 Lessons Learned

Für die SNMP-Abfragen wird PRTG benutzt. Da diese Software nur als Freeware verwendet wird, ist das kleinste Abfrage-Intervall bei 60 Sekunden. Etwa alle 5 Sekunden wären von Vorteil gewesen.

Die Versuche müssen besser automatisiert werden. Das Abfrage-Tool hätte direkt in eine Datei schreiben sollen, damit wäre die Bearbeitung von Hand entfallen und die Versuche hätten über Nacht laufen können und somit schon am nächsten Tag ausgewertet werden können.

Die einzelnen Messungen hätten anders aufgeteilt werden sollen. Anstatt in einer Serie TCP und UDP mit und ohne NetFlow durchzuführen, hätten zuerst alle Messungen mit NetFlow und danach alle ohne NetFlow stattfinden müssen. In dieser Konstellation wäre die Umkonfiguration des Routers nach jeder Messung erspart geblieben.

9 LAB 1B – CPU-BELASTUNG DURCH NETFLOW-EXPORT

9.1 Aufgabenstellung

9.1.1 Ziel

Die Ziele dieses Ergänzungs-Labs sind identisch mit denjenigen von Lab1. Mit den gewonnen Erkenntnissen aus dem Kapitel Lessons Learned wird die Messung wiederholt.

Ermittlung der Bandbreite für UDP, anhand der TCP-Verbindungen. TCP ermittelt selber welche maximale Bandbreite zur Verfügung steht. Dieser Wert kann anschliessend für die UDP-Verbindungen genommen werden.

Während der Messung wird die CPU-Auslastung mit SNMPWalk in eine Datei geschrieben, damit diese ausgewertet werden kann. Im Unterschied zum Lab1 geschieht die SNMP-Abfrage nun alle 5 Sekunden.

Der Verkehr wird nochmals mit Iperf generiert.

Es gibt 2 Messreihen. Jede beinhaltet eine Stunde lang TCP-Verkehr und nach einer halben Stunde Wartezeit UDP-Verkehr. Nach dem Deaktivieren von NetFlow werden die Messreihen wiederholt.

9.1.2 Bedingungen

Wie in Lab1.

9.1.3 Risiken / Challenges

Die Genauigkeit der CPU-Auslastung kann weiterhin nur in Ganzzahlen abgefragt werden.

9.2 Konfiguration

9.2.1 Aufbau

Vom PC-Client werden Datenströme zum PC-Server erzeugt. Diese Datenströme sind entweder UDP oder TCP.
Beim Router ist entweder NetFlow aktiviert oder deaktiviert. Dies ergibt also 4 verschiedene Konstellationen.
Vom MessPC aus wird über SNMP-Abfragen die CPU-Auslastung gemessen.

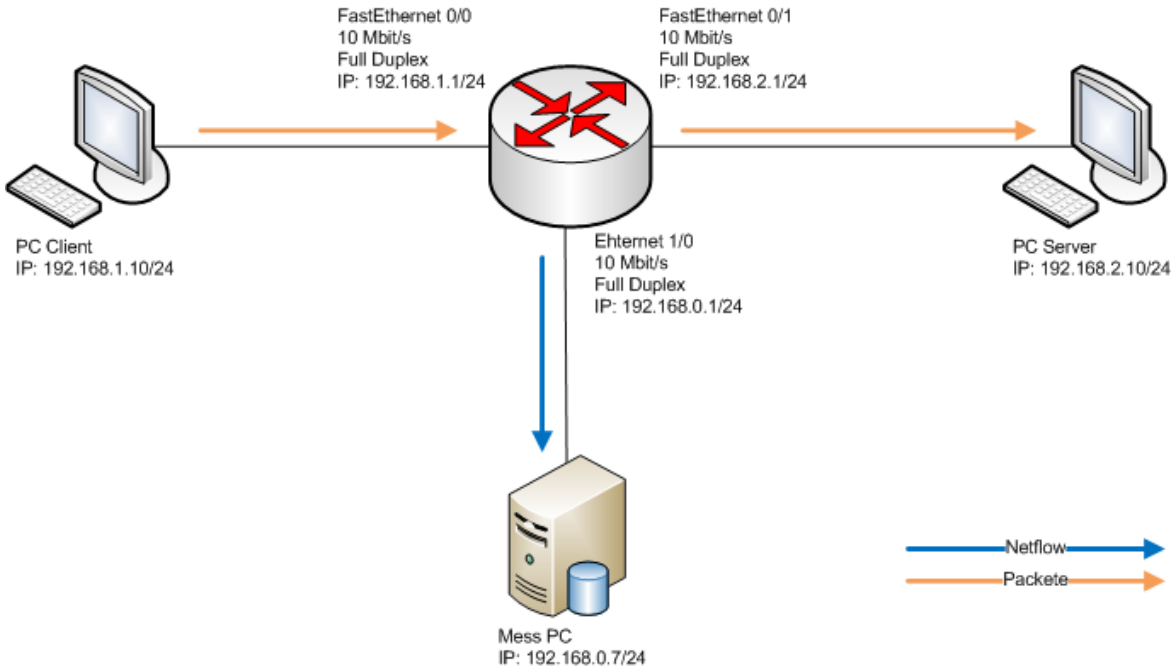


Abbildung 9-1 Aufbau

9.2.2 Router-Konfiguration

Messungen mit NetFlow

Der Router wird mit folgenden Parametern konfiguriert:

```
ip flow-cache timeout active 1
Interface FastEthernet0/1
ip route-cache flow
Interface Ethernet1/0
ip route-cache flow
ip flow-export source FastEthernet0/1
ip flow-export version 5
ip flow-export destination 192.168.0.7 9999
```

Messungen ohne NetFlow

Die NetFlow-Parameter werden entfernt:

```
no ip flow-cache timeout active 1
Interface FastEthernet0/1
no ip route-cache flow
Interface Ethernet1/0
no ip route-cache flow
no ip flow-export source FastEthernet0/1
no ip flow-export version 5
no ip flow-export destination 192.168.0.7 9999
```

9.2.3 SNMPWalk-Konfiguration

Mit folgendem Befehl kann SNMPWalk den Wert der CPU-Auslastung der letzten 5 Sekunden auslesen:

```
snmpwalk -v1 -c public 192.168.0.1 1.3.6.1.4.1.9.9.109.1.1.1.1.6.1
```

9.2.4 Scripts

Shell-Script für SNMPWalk

```
date > /home/hsr/Desktop/answers
for((i=0; i < 12*60*15; i++))
do
snmpwalk -v1 -c public 192.168.0.1 1.3.6.1.4.1.9.9.109.1.1.1.1.6.1
>> /home/hsr/Desktop/answers
sleep 5
done
```

Iperf-Script

```
@echo off
REM Variablen
set max=300
set wait=1200
set bw=7.09M
set file_tcp=Lab1_tcp_night.txt
set file_udp=Lab1_udp_night.txt

echo Messstart TCP1 > %file_tcp%
echo %date% >> %file_tcp%
echo %time% >> %file_tcp%
echo ======================================== >> %file_tcp%
for /L %%n in (1,1,%max%) do (
 echo Messrunde %%n von %max%
 time /t >> %file_tcp%
 call iperf -c 192.168.2.10 -P 1 -n 10M >> %file_tcp%
)
echo ======================================== >> %file_tcp%
echo Messende >> %file_tcp%
```

```
echo %date% >> %file_tcp%
echo %time% >> %file_tcp%
echo ========== >> %file_tcp%
echo == wait == >> %file_tcp%
echo ========== >> %file_tcp%
choice /n /d y /t %wait%
echo Messstart TCP2 >> %file_tcp%
echo %date% >> %file_tcp%
echo %time% >> %file_tcp%
echo ======================================== >> %file_tcp%
for /L %%n in (1,1,%max%) do (
 echo Messrunde %%n von %max%
 time /t >> %file_tcp%
 call iperf -c 192.168.2.10 -P 1 -n 10M >> %file_tcp%
)
echo ======================================== >> %file_tcp%
echo Messende >> %file_tcp%
echo %date% >> %file_tcp%
echo %time% >> %file_tcp%
echo ========== >> %file_tcp%
echo == wait == >> %file_tcp%
echo ========== >> %file_tcp%
choice /n /d y /t %wait%
echo Messtart UDP1 > %file_udp%
echo %date% >> %file_udp%
echo %time% >> %file_udp%
echo ======================================== >> %file_udp%
for /L %%n in (1,1,%max%) do (
 echo Messrunde %%n von %max%
 time /t >> %file_udp%
 call iperf -c 192.168.2.10 -P 1 -u -b %bw% -n 10M >> %file_udp%
)
echo ======================================== >> %file_udp%
echo Messende >> %file_udp%
echo %date% >> %file_udp%
echo %time% >> %file_udp%
echo ========== >> %file_udp%
echo == wait == >> %file_udp%
echo ========== >> %file_udp%
choice /n /d y /t %wait%
echo Messtart UDP2 >> %file_udp%
echo %date% >> %file_udp%
echo %time% >> %file_udp%
echo ======================================== >> %file_udp%
for /L %%n in (1,1,%max%) do (
 echo Messrunde %%n von %max%
 time /t >> %file_udp%
 call iperf -c 192.168.2.10 -P 1 -u -b %bw% -n 10M >> %file_udp%
)
echo ======================================== >> %file_udp%
echo Messende >> %file_udp%
echo %date% >> %file_udp%
echo %time% >> %file_udp%
```

9.3 Testresultate

9.3.1 Erwartet

Durch das SNMP-Polling im 5 Sekunden Intervall sollten nun genügend Daten vorhanden sein, um eine bessere Genauigkeit zu erreichen.

9.3.2 Gemessen

In diesem Lab werden 2 Durchgänge gemessen.

Messung 1

TCP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	13.5726	1.0582
NetFlow Aus	11.3652	1.075
CPU Auslastung durch NetFlow	2.2075	-

Tabelle 9-1: Messung 1 - TCP-Verbindungen

UDP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	9.6932	0.3435
NetFlow Aus	7.7875	0.3576
CPU Auslastung durch NetFlow	1.9061	-

Tabelle 9-2: Messung 1 - UDP-Verbindungen

Messung 2

TCP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	13.5499	1.1294
NetFlow Aus	11.1726	0.963
CPU Auslastung durch NetFlow	2.3772	-

Tabelle 9-3: Messung 2 - TCP-Verbindungen

UDP-Verbindungen:

	Mittelwert CPU-Auslastung	Varianz des Mittelwertes
NetFlow Ein	9.6936	0.3378
NetFlow Aus	7.7972	0.3267
CPU Auslastung durch NetFlow	1.8965	-

Tabelle 9-4: Messung 2 - UDP-Verbindungen

Auswertung

Durch die hohe Anzahl an Messwerten wird eine sehr kleine Varianz erreicht, was bestätigt, dass in diesem Lab die Genauigkeit markant erhöht wird.

	CPU Auslastung durch NetFlow	Varianz der CPU Auslastung
UDP	1.9013	0.00005
TCP	2.2924	0.0144

Tabelle 9-5: Auswertung CPU Last

9.4 Fazit

Durch je ca. 700 Werte (SNMP-Abfragen) pro Messreihe und pro Protokoll kann eine genügend genaue Auswertung errechnet werden. Die Abfragen über die CPU-Auslastung sind nun lückenlos, da alle 5 Sekunden der Wert der letzten 5 Sekunden abgefragt wird.

Die Anzahl NetFlows halten sich pro gesendetes NetFlow-Paket unter 10 Stück. Ein Test über die Auswirkungen der Anzahl NetFlows wird im Lab3 folgen.

Da der empfangene Wert zwischen 0 und 100 liegt, kann die Zahl als Prozentzahl interpretiert werden. Die CPU-Auslastung bei aktiviertem NetFlow-Export beträgt also für UDP 1.9% und für TCP 2.29%.

10 LAB 2 – AUSLASTUNG INTERFACE DURCH HOHE BANDBREITE

10.1 Aufgabenstellung

10.1.1 Ziel

Es ist zu ermitteln welche Auswirkungen ein überlastetes Interface im Zusammenhang mit NetFlow-Daten hat. Das Interface, an welchem die NetFlow-Daten gesendet werden, soll soweit ausgelastet werden, damit keine NetFlow-Daten mehr gesendet werden können.

Wie reagiert der Router auf diese Situation. Werden die NetFlow-Pakete zwischengespeichert oder werden sie irgendwann verworfen?

Welche Unterschiede sind zu beobachten im Vergleich zu einem Interface ohne eine solche Last?

10.1.2 Bedingungen

Die CPU darf nicht auf 100% ausgelastet sein, es darf nur das Interface auf Vollast sein. Damit kann der Router wie gewohnt die NetFlows berechnen und wenn nötig zwischenspeichern.
Die Interfaces müssen deswegen auf 10 Mbit/s eingestellt werden.

10.1.3 Risiken / Challenges

Die NetFlow-Pakete sind in der Regel klein (ca. 216 Bytes). Es besteht die Möglichkeit, dass diese trotzdem noch gesendet werden können. Daher muss Netzwerkverkehr generiert werden, welcher die Bandbreite auslasten kann.

Eine andere Herausforderung könnte die Ungenauigkeit der Messverfahren sein. Wenn der Unterschied zu klein ist, kann dieser gar nicht erst ermittelt werden.

10.2 Konfiguration

10.2.1 Aufbau

Der Router soll die NetFlow-Pakete noch zusätzlich an den PC Server schicken. Dieses Interface ist mit UDP-Traffic auszulasten.

Auf dem Mess PC und dem PC Server wird der Netzwerkverkehr aufgezeichnet. Danach kann ermittelt werden, ob die Anzahl der NetFlow-Pakete übereinstimmt.

Daraus lässt sich schliessen, dass ein Paket verspätet gesendet wird, wenn wieder Kapazität vorhanden ist, ob das Paket ganz verworfen wird oder ob es gespeichert wird, bis die Netzwerklast nachlässt.

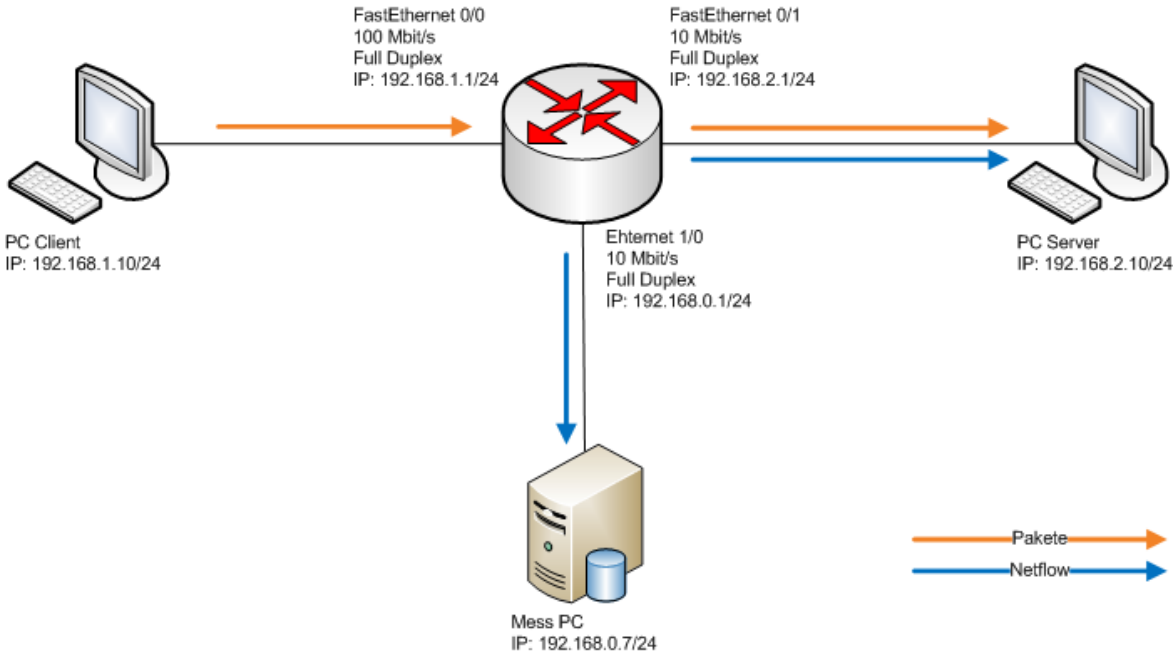


Abbildung 10-1 Aufbau

10.2.2 Wireshark - Konfiguration

Damit nicht alle Pakete gespeichert werden, kann ein Capture Filter gesetzt werden. Diese veranlasst Wireshark nur die gewünschten Pakete einzufangen. Dadurch wird nur gewünschter Netzwerkverkehr aufgezeichnet, was viel Speicherplatz einspart.

Es werden die NetFlow-Pakete und für die Intervall Messung noch die Pings, welche als Startzeichen dienen aufgezeichnet.

```
udp port 9999
udp port 9999 or icmp[icmptype]==icmp-echo
```

10.2.3 Scripts

Das Script stellt eine UDP-Verbindung mit Iperf her. Die Verbindung belegt eine Bandbreite von 20 Mbit/s und übermittelt Daten eine Stunde (resp. 3600 Sekunden) lang.

```
@echo off

REM //Variablen
set max=1
set file="ergebnis_udp_04.txt"

echo Messtart UDP > %file%
echo %date% >> %file%
echo %time% >> %file%
echo ======================================== >> %file%

for /L %%n in (1,1,%max%) do (
echo Messrunde %%n von %max%
time /t >> %file%
call iperf -c 192.168.2.10 -P 1 -u -b 20M -t 3600 >> %file%
)

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

10.3 Testresultate

10.3.1 Erwartet

Wenn der Router auf dem Interface keine Bandbreite mehr zur Verfügung hat, sollte er die NetFlow-Pakete für eine bestimmte Zeit zwischenspeichern. Es ist zu erwarten, dass diese bestimmte Zeit ermittelt werden kann.

Die Hauptaufgabe des Routers besteht darin Netzwerkpakete zu vermitteln, daher sollte er den Netzwerkverkehr gegenüber allem anderen bevorzugt behandeln. Es kann angenommen werden, dass einzelne NetFlow-Pakete zwar verloren gehen, jedoch die meisten beim NetFlow-Kollektor ankommen. Das Interface des Mess PC wird nicht belastet, daher wird der NetFlow-Kollektor alle Netflow-Pakete erhalten.

10.3.2 Gemessen

Während das Interface ausgelastet wird, empfängt der Server kein einziges NetFlow-Paket. Hingegen hat der Mess PC alle NetFlow-Pakete erhalten.

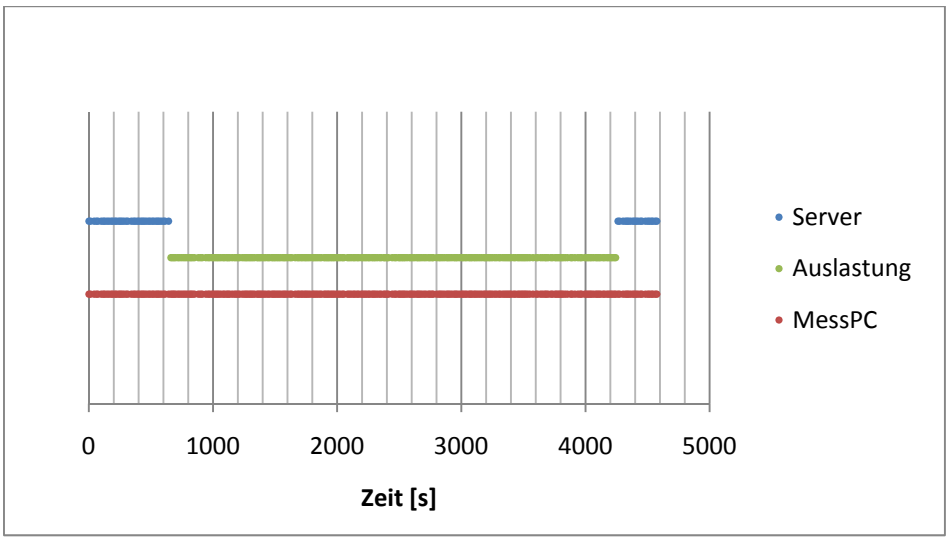


Abbildung 10-2 Ankommende NetFlowdaten

Bei der Analyse der Netflows zeigt sich, dass die Netflow Sequenznummer nicht fortgesetzt, sondern die Pakete einfach ausgelassen werden.

Durch dieses Resultat wird nun untersucht, zu welchen Zeiten der Router die NetFlow-Pakete an den Server und an den Mess PC schickt.

Dadurch, dass beide Computer identisch sind, kann angenommen werden, dass das Zeitverhalten durch die Hardware auch identisch sein muss. Trotzdem lässt sich eine Abweichung messen.
Eine konstante Abweichung könnte dadurch erklärt werden, dass z.B. die Weglänge nicht identisch ist, sprich, unterschiedlich lange Kabel verwendet werden.
Eine andere Möglichkeit ist, dass der Router zuerst das NetFlow-Paket an eine Schnittstelle schickt und anschliessend an die andere, dies also nicht gleichzeitig erledigen kann.

Die Messung hat jedoch ergeben dass die zeitliche Abweichung nicht linear und steigend ist. Damit die Nichtlinearität erkennbar ist, wird zusätzlich ein linearer Graph in die Abbildung 10-3 eingeblendet.

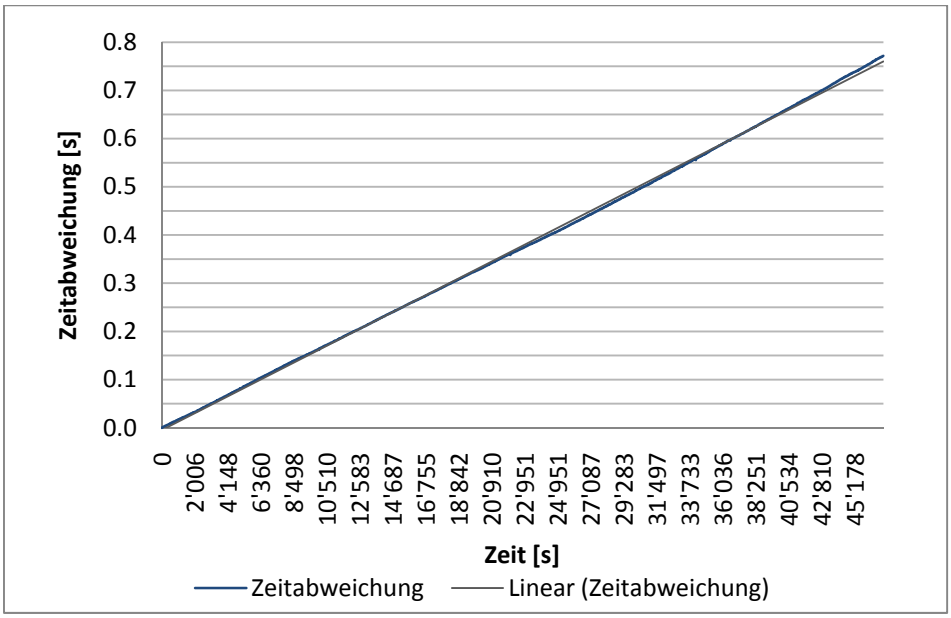


Abbildung 10-3 Zeitunterschied ausgehend MessPC im Vergleich mit dem Server

Die Schlussfolgerung dieses Phänomens basiert auf den unterschiedlichen Zeiten auf den Systemen. Es ist bekannt, dass die verschiedenen Clocks nie identisch laufen. Diese Aussage ist so zu verstehen, dass bei so genauer Messung sich der Unterschied des Clocks bemerkbar macht. Im Klartext heisst dies, dass der Clock auf dem Server schneller läuft als auf dem Mess-PC.
Bei verteilten Systemenen ist die Zeitsynchronisation, resp. das Auseinanderlaufen der Uhren eine grosse Herausforderung[38].
Alle 13.112783 Stunden (entspricht 47206 Sekunden) läuft der Clock 0.772 Sekunden voraus (relativ gesehen zum Mess PC).

Zeit (in Sekunden / Minuten)	Abweichung (in Sekunden)
120 / 2	0.001893
600 / 10	0.009730
6000 / 100	0.098129
12001 / 200.0167	0.196298
24003 / 400.05	0.392651
47206 / 786.767	0.771783

Tabelle 10-1 Abweichung des Clocks des Servers

Die weitere Herausforderung ist nun, herauszufinden, wie lange der Router ein NetFlow-Paket zwischenspeichert. Für diese Aufgabe wird eine Intervall-Messung durchgeführt, da aus der Tabelle 10-2 ersichtlich ist, dass der Router sich nicht strikt an die eingestellten 15 Sekunden hält, bei welchen er das NetFlow-Paket versendet.

	Wert (in Sekunden)
Eingestellter Wert	15
Mittelwert	15.4822
Minimum	11.0042
Maximum	30.999
Varianz	10.7356

Tabelle 10-2 Abweichungen zwischen den Einstellungen und Effektiv

Eine weitere Analyse ergibt eine leichte Korrelation im Zusammenhang mit der Paketgrösse und der Zeitdifferenz vom Senden des NetFlow-Paketes zum vorhergehenden.
In der Abbildung 10-4 ist dies anhand der linearen Trendlinie ersichtlich. Die Paketgrösse ist mindestens 80 Bytes. Der kürzeste gemessene Zeitabstand beträgt 11 Sekunden.
Dies lässt sich in der Grafik gut erkennen. So beträgt der Mindestabstand von den Messpunkten zur Abszisse jeweils 80 Bytes und der Abstand zur Ordinate 11 Sekunden.

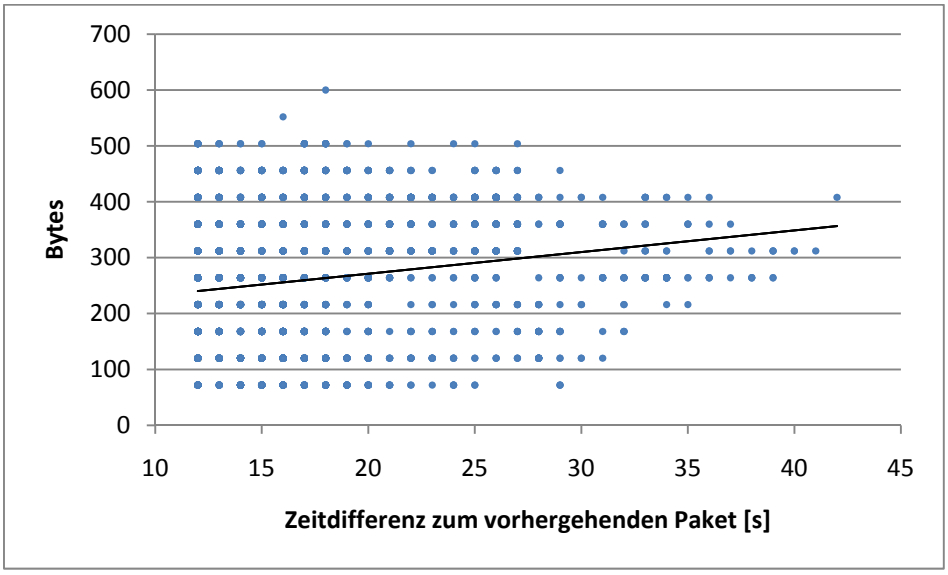


Abbildung 10-4 Korrelation Zeitdifferenz und Paketgrösse von NetFlow

10.4 Fazit

Der Router wirft ein NetFlow-Paket einfach weg, wenn genau in diesem Moment die Bandbreite ausgelastet ist. Es ist daher dringend zu empfehlen für den NetFlow-Export ein separates Interface zu verwenden.

Der Timeout (inactive), welcher auf 15 Sekunden eingestellt ist, hält sich nur im Mittelwert an die definierte Einstellung. Es gibt sehr grosse Unterschiede zwischen dem Sendezeitpunkt der einzelnen NetFlow-Paketen (entsprechend gross ist die Varianz in der Tabelle 10-2).

Eine leichte Korrelation zum Sendezeitpunkt mit der Paketgrösse ist zu erkennen.

11 LAB 3-1 - CPU AUSLASTUNG ANHAND DER FLOW-MENGE

11.1 Aufgabenstellung

11.1.1 Ziel

Es ist zu messen ob eine Korrelation zwischen der Anzahl generierter NetFlows und der CPU-Auslastung besteht.
Die CPU-Auslastung wird wiederum über SNMP gemessen.

11.1.2 Bedingungen

Es werden verschieden grosse Pakete mit Iperf generiert, bei welchen genau bekannt ist, wie viele NetFlows der Router anhand dieser generieren sollte.

11.1.3 Risiken / Challenges

Die Messgenauigkeit spielt eine grosse Rolle. Falls keine Abweichung bei der CPU-Auslastung zu messen ist, könnte diese einfach kleiner als 1% sein. Für diesen Schluss wäre allerdings eine Korrelation uninteressant und hätte im praktischen Aufbau keine relevante Auswirkung.

11.2 Konfiguration

11.2.1 Aufbau

Bei der anfänglich geführten Besprechung wurden gleich zwei Messungen vorgeschlagen, weshalb die Labbezeichnung gleich zu Beginn mit Untermessungen startet.

Vom PC-Client aus werden pro Durchlauf immer je 1 GByte Daten über den Router zum PC-Server geschickt. Nach jedem Durchlauf werden die Anzahl NetFlows verdoppelt und die Datenmenge halbiert. Somit soll gewährleistet werden, dass die CPU-Auslastung für den Netzwerkverkehr über die ganze Messreihe konstant bleibt und so eine Differenz anhand der NetFlow-Anzahl entsteht.

Die CPU-Auslastung wird vom MessPC über SNMP ermittelt.

Ein NetFlow-Kollektor auf einer VMWare empfängt die NetFlow-Pakete vom Router auf dem Port 9999.

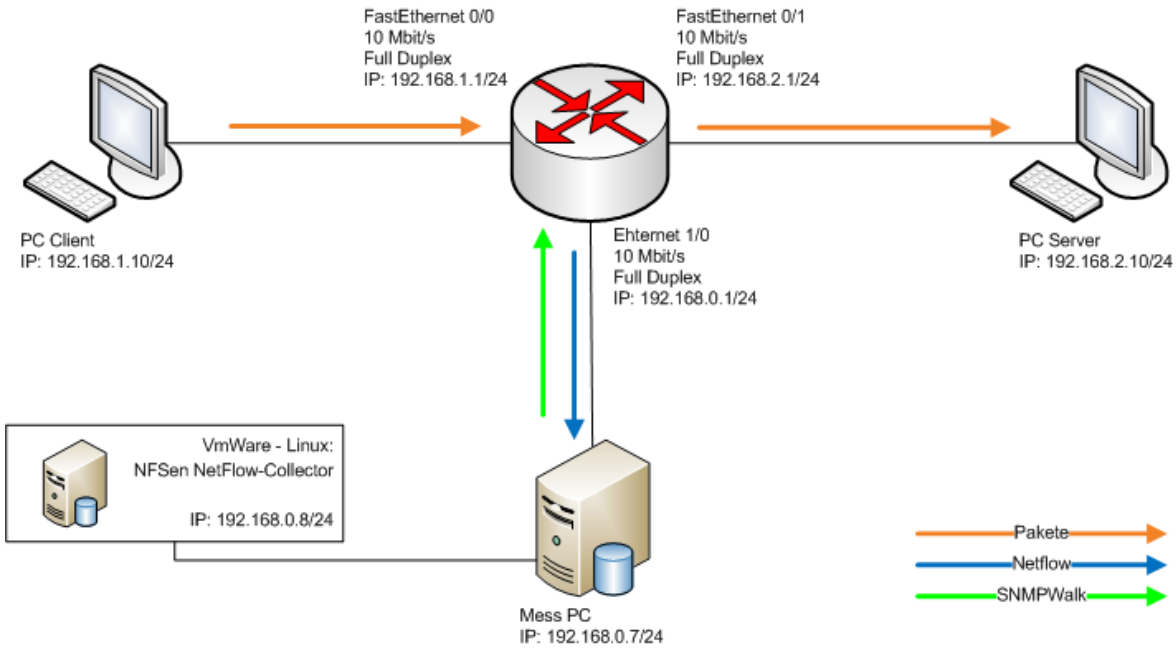


Abbildung 11-1 Aufbau

11.2.2 Iperf-Script

Das Script "IPERF_Lab3_1.cmd" erzeugt 1 Gigabyte Daten, welche übers Netzwerk versendet wird. Bei jedem Durchgang wird die Datengrösse halbiert und die Anzahl seriellen Verbindungen verdoppelt. Dabei bleibt die gesendete Gesamtdatenmenge konstant.

Bei Iperf kann die Datengrösse in Bytes, Kilobytes oder Megabytes angegeben werden. Der Parameter ist auf 32 Bits beschränkt, was 512 Megabytes entspricht (2^32 / 8 Bit / 1024 / 1024 = 512 [MB]).

Im ersten Teil werden die Angaben der Datengrösse in Megabytes geschrieben. Im letzten Durchlauf des ersten Teils werden 512 serielle Verbindungen mit 2 Megabyte erzeugt. Anschliessend läuft der zweite Teil mit den Angaben der Datengrösse in Kilobytes.

```
@echo off

REM //Variablen
set size=1024
set runs=1
set wait=60
set file="Lab3_tcpbig_01.txt"

echo Messtart TCP > %file%
echo %date% >> %file%
echo %time% >> %file%
REM echo ======================================== >> %file%

:START1
if %size% LSS 2 goto :NEXT1
echo ======================================== >> %file%
echo size=%size% >> %file%
echo runs=%runs% >> %file%
echo %time% >> %file%

for /L %%n in (1,1,%runs%) do (
iperf -c 192.168.2.10 -P 1 -n %size%M >> %file%
set /a size=%size% / 2
)

set /a runs=%runs% * 2

REM Waittime(%wait% Sec)
choice /n /d y /t %wait%

goto :START1
:NEXT1

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%


REM Waittime(%wait% Sec)
choice /n /d y /t %wait%

REM //Variablen
set size=1024
set runs=1024
set file="Lab3_tcpsmall_01.txt"

echo Messtart TCP > %file%
echo %date% >> %file%
echo %time% >> %file%
REM echo ======================================== >> %file%
```

```
:START2
if %size% LSS 32 goto :NEXT2
echo ======================================== >> %file%
echo size=%size% >> %file%
echo runs=%runs% >> %file%
echo %time% >> %file%

for /L %%n in (1,1,%runs%) do (
iperf -c 192.168.2.10 -P 1 -n %size%K >> %file%
set /a size=%size% / 2
)

set /a runs=%runs% * 2

REM Waittime(%wait% Sec)
choice /n /d y /t %wait%

goto :START2
:NEXT2

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

Beim Versuch kleinere Datenmengen als 32 Kilobytes mit Iperf zu senden, stürzt Iperf ab. Aus diesem Grund wird die Anzahl von parallelen Verbindungen mit dem Script "IPERF_Lab3_1_tcpthreads.cmd" erhöht. Dieses stösst an die Grenze der maximalen Anzahl von gleichzeitigen Verbindungen, was ebenfalls im Absturz von Iperf resultiert.

```
@echo off

REM //Variablen
set size=128
set runs=10
set threads=32
set wait=120
set file="Lab3_1_1_sub.txt"

echo Messtart TCP Small > %file%
echo %date% >> %file%
echo %time% >> %file%
REM echo ======================================== >> %file%

:START
if 10 LSS 1 goto :NEXT
echo ======================================== >> %file%
echo size=%size% >> %file%
echo runs=%runs% >> %file%
echo threads=%threads% >> %file%
echo %time% >> %file%

REM //Rundenstart
ping 192.168.0.7 -n 1

for /L %%n in (1,1,%runs%) do (
```

```
REM echo %%n >> %file%
start iperf -c 192.168.2.10 -P %threads% -p 4001 -t 2
start iperf -c 192.168.2.10 -P %threads% -p 5001 -t 2
start iperf -c 192.168.2.10 -P %threads% -p 6001 -t 2
REM set /a size=%size% / 2
)

REM set /a runs=%runs% * 2
set /a threads=%threads% * 2

REM Waittime(%wait% Sec)
choice /n /d y /t %wait%

goto :START
:NEXT

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

11.2.3 Wireshark - Konfiguration

Es werden nur die NetFlow-Pakete und die Pings als Marker aufgezeichnet (Capture Filter).

```
udp port 9999 or icmp[icmptype]==icmp-echo
```

11.2.4 CPU-Auslastung über SNMP

Über SNMPWalk wird alle 5 Sekunden die CPU-Auslastung gemessen und in eine Datei gespeichert.

```
date > /home/hsr/Desktop/Lab3-1CPU
for ((i=0; i< 60*12*15; i++))
do
 snmpwalk -v1 -c public 192.168.0.1
1.3.6.1.4.1.9.9.109.1.1.1.1.6.1 >> /home/hsr/Desktop/Lab3-1CPU
 sleep 5
done
```

11.2.5 NFDump

NFSen speichert die Flows in Abständen von 5 Minuten in eine neue binäre Datei. Um die einzelnen Flows darzustellen, müssen diese lesbar in eine Textdatei mit NFDump extrahiert werden. Dies geschieht mit folgendem Befehl:

```
./nfdump -R /opt/NFSen/ -o "fmt:%ts %td %pr %sa %sp %da %dp %pkt
%bps %pps %bpp" >> /opt/NFSen/lab3-1_Flows
```

Der Parameter `-o` definiert eine benutzerdefinierte Ausgabe in folgender Reihenfolge: Datum, Zeit, Dauer, Protokoll, Sende-IP, Sende-Port, Ziel-IP, Ziel-Port, Anzahl Pakete, Datenrate, Pakete pro Sekunde, Bytes pro Paket.

11.3 Testresultate

11.3.1 Erwartet

Mit steigender Anzahl NetFlows generiert der Router mehr NetFlow-Pakete. Auch wird erwartet, dass die CPU-Belastung proportional ansteigt.

11.3.2 Gemessen

Die Messung ergibt, dass sich die CPU-Auslastung nicht verändert, wenn die Anzahl NetFlows pro Sekunde steigt. In der Abbildung 11-2 ist sogar ein kleiner Abfall zu erkennen.

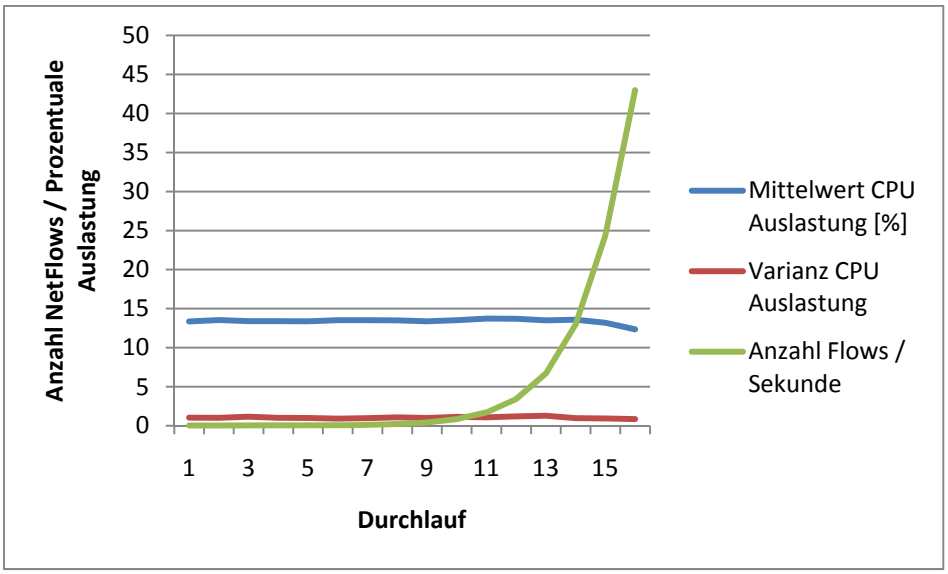


Abbildung 11-2 CPU-Belastung pro Anzahl NetFlows

Diese Erkenntnis überrascht und deswegen wird nochmals eine Messung durchgeführt, bei welcher noch viel mehr NetFlows pro Sekunde generiert werden sollen.

In der Abbildung 11-3 scheint sich die Vermutung nicht zu bestätigen, allerdings ist zu erwähnen, dass die Varianz für die CPU-Auslastung zu hoch ist, um genaue Aussagen schreiben zu können. Es kann daher immer noch angenommen werden, dass die CPU-Auslastung bei höherer Anzahl NetFlows pro Sekunde sich nicht verändert.

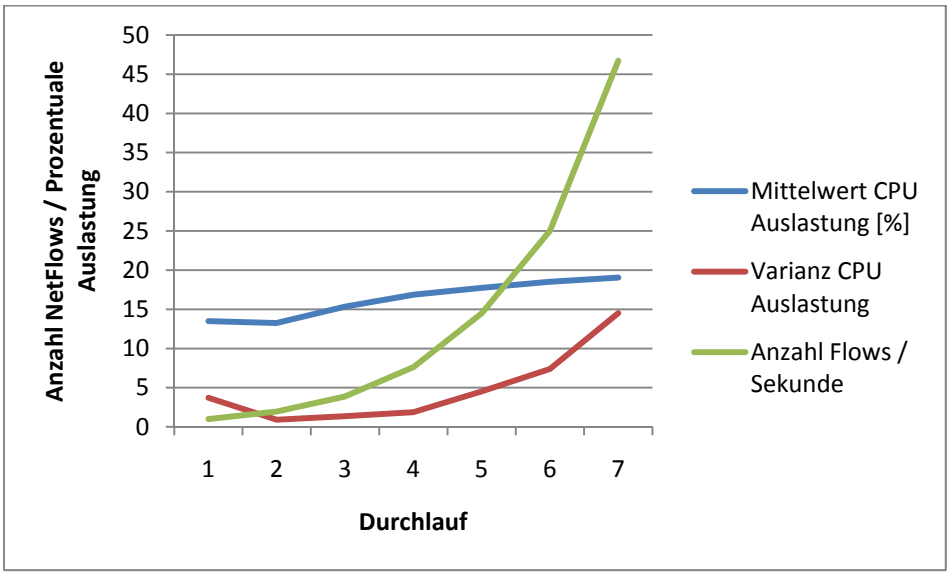


Abbildung 11-3 CPU-Belastung pro Anzahl NetFlows Lab 3-1-1

Es wird nochmals eine Messung durchgeführt. Diese bricht nach ca. 50 NetFlows pro Sekunde ab. Nach einer Analyse dieses Problems, scheint es wahrscheinlich, dass die Ports für Iperf ausgehen, da die einzelnen Datenströme nun in Sekundenbruchteile starten und enden. Das Betriebssystem blockiert einen Port nach der Nutzung eine Weile und kann somit nicht sofort weiterverwendet werden.
Dies ist auch auf der Abbildung 11-4 zu erkennen. Eine lineare Vorhersage bestätigt aber wiederum, dass sich die CPU-Auslastung mit grosser Wahrscheinlichkeit nicht ändern wird.

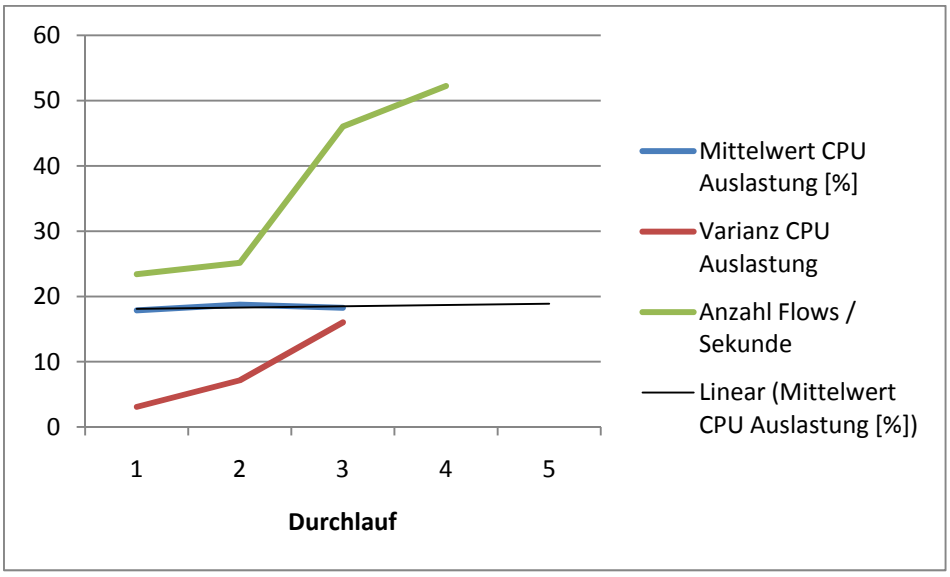


Abbildung 11-4 CPU-Belastung pro Anzahl NetFlows Lab 3-1-2

11.4 Fazit

Die NetFlow-Pakete sind minimal 80 Byte gross und beinhalten somit einen NetFlow. Pro weiteren NetFlow werden zusätzliche 48 Byte benötigt. Es sind also maximal 30 NetFlows in einem NetFlow-Paket erlaubt, um die IP-Paketgrösse nicht zu überschreiten.

Bei grosser Anzahl an NetFlows werden die NetFlow-Pakete, welche die maximalen 30 NetFlows enthalten, einfach in kürzeren Abständen hintereinander gesendet.

12 LAB 3-2 - CPU AUSLASTUNG

12.1 Aufgabenstellung

12.1.1 Ziel

Netflows werden mittels Softwareberechnungen auf dem Cisco Router errechnet. Nun soll die CPU des Routers voll ausgelastet werden, dass die Netflow-Berechnungen beeinflusst werden. Dabei kann ermittelt werden wie der Router reagiert, ob die Netflows ausgerechnet, aber diese wegen der Überlastung der CPU erst verspätet versendet oder die Netflowdaten-Berechnung nicht mehr durchgeführt und somit keine Pakete versendet werden.

12.1.2 Bedingungen

Die CPU muss mit Berechnungen, wie zum Beispiel der Erstellung der Routing Tables, mittels einem Routing Protokoll oder mit dem Routen von vielen Paketen belastet werden. Bei der CPU Belastung muss darauf geachtet werden, dass die Bandbreite die Kapazität des Ausgangsinterfaces nicht ausreizt, so dass die Pakete immer noch gut versendet werden können.

12.1.3 Risiken / Challenges

Die erste Challenge besteht darin die CPU des Routers auf konstante 99.99% auszulasten. Welcher Datenstrom dazu benötigt wird muss ebenfalls ermittelt werden.
Zu der erzeugten CPU Belastung kommt die Netflowdaten-Berechnung hinzu, welche die CPU an den Anschlag bringt.

Zur Auswertung müssen beide Netflows miteinander verglichen werden. Zum einen die Netflows des Routers und zum anderen die Flows aus der Aufzeichnung der Paketdaten.

12.2 Konfiguration

12.2.1 Aufbau

Der Übertragungsrage der Interfaces 0/0 und 0/1 wurde auf 100 Mbit/s erhöht. Vom PC Client aus wird eine stetige UDP-Verbindung von 35 Mbit/s zum PC Server gesendet. Eine TCP-Verbindung wird zum PC Server gesendet, welche bei jedem Durchgang die Datenmenge halbiert und die Anzahl der parallelen Verbindungen verdoppelt. Durch die UDP- und TCP-Verbindungen wird die CPU ausgelastet, was mit SNMP überprüft wird. Ist nur die UDP-Verbindung aktiv liegt die CPU-Auslastung unter 100%.

Die Daten der CPU-Auslastung wird vom Mess PC ermittelt und ein NetFlow-Collector auf einer VMWare empfängt die NetFlow-Pakete vom Router auf dem Port 9999.

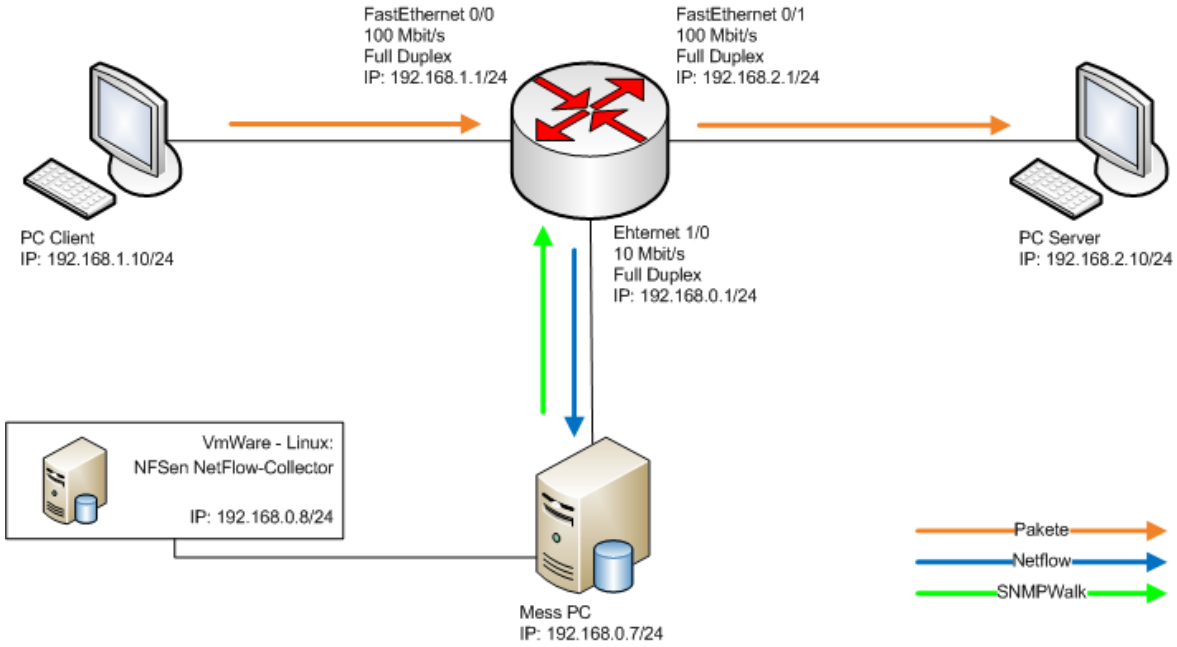


Abbildung 12-1 Aufbau

12.2.2 Wireshark - Konfiguration

Es werden nur die NetFlow-Pakete und die Pings als Marker aufgezeichnet (Capture Filter).

```
udp port 9999 or icmp[icmptype]==icmp-echo
```

12.2.3 CPU-Auslastung über SNMP

Über SNMPWalk wird alle 5 Sekunden die CPU-Auslastung gemessen und in eine Datei gespeichert. Zusätzlich verhindert die Option `–r 1`, dass standardmässig 5 Mal ein Neuversuch stattfindet, wenn der erste Versuch keine Antwort liefert.

```
date > /home/hsr/Desktop/Lab3-1CPU
for ((i=0; i< 60*12*15; i++))
do
 snmpwalk -v1 –r 1 -c public 192.168.0.1
1.3.6.1.4.1.9.9.109.1.1.1.1.6.1 >> /home/hsr/Desktop/Lab3-1CPU
 sleep 5
done
```

12.2.4 NFDump

Die NetFlows in den Binärdateien von NFDump werden in eine Textdatei geschrieben.

```
./nfdump -R /opt/NFSen/ -o "fmt:%ts %td %pr %sa %sp %da %dp %pkt
%bps %pps %bpp" >> /opt/NFSen/lab3-2_Flows
```

12.2.5 Iperf Scripts

Lab 3-2

Das erste Script "IPERF_Lab3_2.cmd" startet eine UDP Verbindung, welche genau eine Bandbreite von 35 Mbit/s belegt. Diese Verbindung wird für 20 Stunden aufrecht erhalten. Diese Verbindung sorgt für eine stetige CPU-Belastung. Sobald die Verbindung startet, wird das zweite Script "IPERF_Lab3_2_tcp02.cmd" ausgeführt.

```
@echo off

REM //Variablen
set wait=180
set file="Lab3_2_main.txt"

echo Messtart UDP > %file%
echo %date% >> %file%
echo %time% >> %file%
REM echo ======================================== >> %file%

REM //UDP TestPC (Permanent)
start iperf -c 192.168.2.10 -u -P 1 -b 35M -p 4001 -t 72000

REM //TCP Startscripts
start IPERF_Lab3_2_tcp02.cmd

REM Waittime(%wait% Sec)
REM choice /n /d y /t %wait%

echo ======================================== >> %file%
echo Messende >> %file%
```

```
echo %date% >> %file%
echo %time% >> %file%
```

Das zweite Script "IPERF_Lab3_2_tcp02.cmd" wird vom vorhergehenden Script aufgerufen. Dabei werden zuerst grössere Datenpakete gesendet und pro Durchlauf die Anzahl der parallelen Verbindungen verdoppelt.

```
@echo off

REM //Variablen
set size=1024
set runs=1
set threads=1
set wait=30
set file="Lab3_2_sub4.txt"

echo Messtart TCP Small > %file%
echo %date% >> %file%
echo %time% >> %file%
REM echo ======================================== >> %file%

:START2
if %size% LSS 4 goto :NEXT2
echo ======================================== >> %file%
echo size=%size% >> %file%
echo runs=%runs% >> %file%
echo threads=%threads% >> %file%
echo %time% >> %file%

REM //Rundenstart
ping 192.168.0.7 -n 1

for /L %%n in (1,1,%runs%) do (
iperf -c 192.168.2.10 -P %threads% -n %size%K >> %file%
set /a size=%size% / 2
)

REM set /a runs=%runs% * 2
set /a threads=%threads% * 2

REM Waittime(%wait% Sec)
choice /n /d y /t %wait%

goto :START2
:NEXT2

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

Lab 3-2-1

Das Iperf Script "IPERF_Lab3_2_1_tcp.cmd" erzeugt einen TCP Paketdatenstrom, welcher immer 20 Sekunden länger Pakete überträgt, als der UDP Paketdatenstrom. Dieses Skript wird über ein Wochenende ausgeführt damit eine grössere Messreihe entsteht.

```
@echo off

REM //Variablen
set runs=15
set threads=1
set duration=30
set duration2=50
set wait=100
set file="Lab3_2_1_tcp.txt"

echo Messtart TCP > %file%
echo %date% >> %file%
echo %time% >> %file%
REM echo ======================================== >> %file%

:START
if %duration% GTR 18000 goto :NEXT
echo ======================================== >> %file%
echo runs=%runs% >> %file%
echo threads=%threads% >> %file%
echo %time% >> %file%

REM //Rundenstart
ping 192.168.0.7 -n 1

for /L %%n in (1,1,%runs%) do (

REM //TCP:5001
echo dauerTCP=%duration2% >> %file%
start iperf -c 192.168.2.10 -P 1 -t %duration2%

REM //UDP:4001
echo dauerUDP=%duration% >> %file%
iperf -c 192.168.2.10 -u -P %threads% -p 4001 -b 70M -t %duration%
>> %file%
echo Ende UDP

REM Waittime(%wait% Sec)
choice /n /d y /t 30
)

REM set /a runs=%runs% * 2
REM set /a threads=%threads% * 2
set /a duration=%duration% + 20
set /a duration2=%duration2% + 20

REM Waittime(%wait% Sec)
choice /n /d y /t %wait%

goto :START
```

```
:NEXT

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

12.2.6 CPU-Auslastung mit SNMPWalk

Mit SNMPWalk wird die ganze MIB durchlaufen. Dies generiert sehr viele SNMP-Requests und belastet entsprechend die Router-CPU.

```
date > /home/hsr/Desktop/Lab3-2_Rounds
for ((i=0; i< 60*2*20; i++))
do
 snmpwalk -v1 -c public 192.168.0.1 .iso
 echo $i >> /home/hsr/Desktop/Lab3-2_Rounds
 sleep 30
done
```

12.2.7 CPU-Auslastung mit TCL (Tool Command Language)

Mit TCL gibt es von Cisco die Möglichkeit auf dem IOS Scripts auszuführen.

Gemäss Hersteller Spezifikation verfügt das verwendete Cisco IOS Image nicht über das Feature Cisco IOS Scripting /w Tcl. Diese Tatsache wurde im Cisco Feature Navigator[39] nachgeschaut, womit wir TCL nicht für die CPU-Auslastung nutzen können

12.3 Testresultate

12.3.1 Erwartet

Es wird erwartet dass die CPU des Router die Netflowdaten nicht vollständig berechnen kann und so einige Netflows nicht angezeigt werden. So können in der Netflow-Analyse Lücken entstehen.

12.3.2 Gemessen

Mit Iperf stellt sich ein Problem ein, welches sich ab dem 11. Durchlauf bemerkbar machte. Die CPU ist mit dem Script nicht mehr voll ausgelastet, womit die nachfolgenden Durchläufe fehlerhafte Messungen liefern.

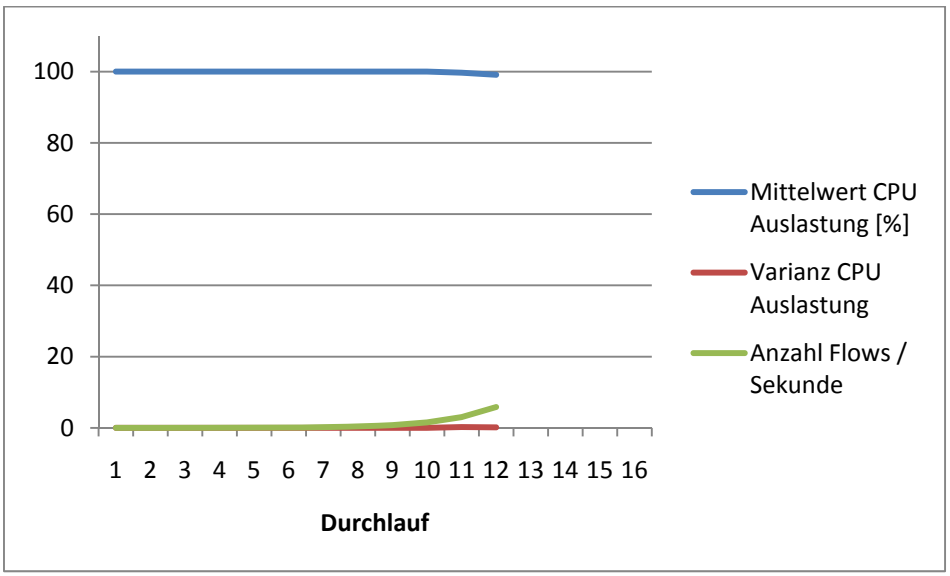


Abbildung 12-2 Auslastung der CPU während dem Versuch

Diese Erkenntnis hat aber für dieses Lab keine weitere Bedeutung.
Es wird etwas Unerwartetes beobachtet, das mehr interessieren könnte und darum genauer analysiert wird.

In den Durchläufen, bei welchen die Dauer der Verbindungen grösser als 60 Sekunden beträgt, wartet der Router solange mit dem Verschicken der NetFlow-Pakete, bis er wieder Rechenkapazität hat, er verwirft die NetFlows nicht.

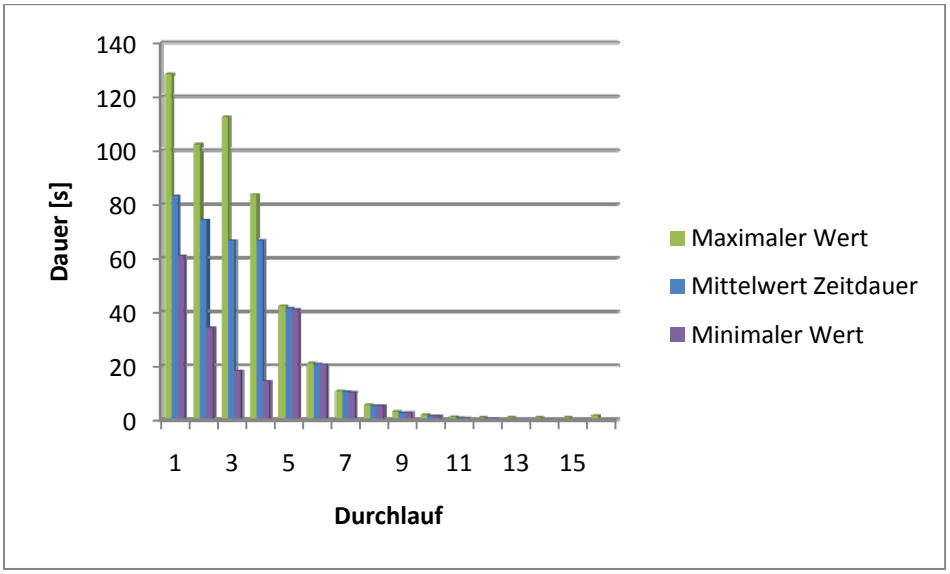


Abbildung 12-3 Flow-Dauer pro Durchlauf

In der Abbildung 12-4 sind noch zur Vollständigkeit die Durchläufe 9-16 ersichtlich.

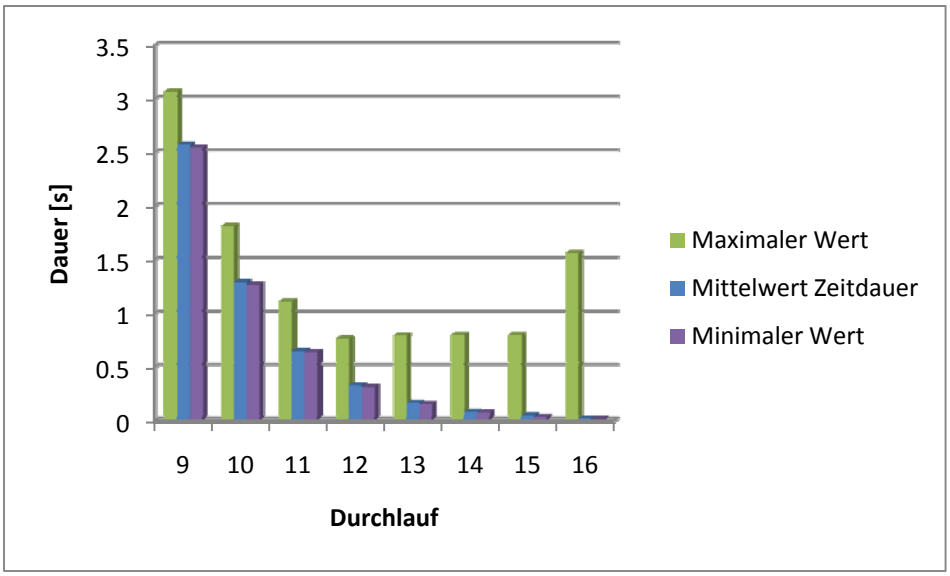


Abbildung 12-4 Flow-Dauer für Durchläufe 9-16

Nun wird gemessen, ob es eine Grenze gibt, wie lange der Router diese Flow-Informationen speichert. Theoretisch müsste er nach der Konfiguration die aktiven Flows alle 60 Sekunden beenden.

Wie in der Abbildung 12-5 ersichtlich, speichert der Router tatsächlich die Netflows und schliesst sie nicht ab, bevor wieder genügend CPU-Kapazität vorhanden ist, um diese zu bearbeiten und danach zu verschicken.

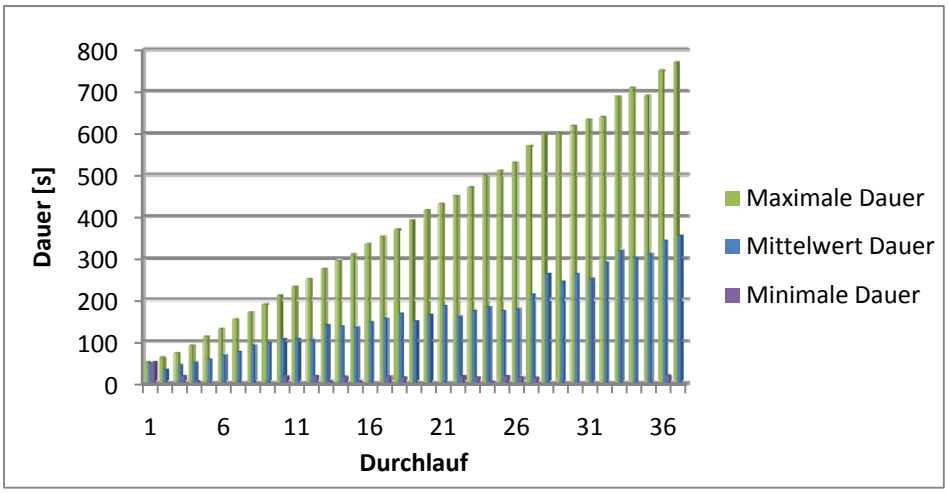


Abbildung 12-5 Flow-Dauer mit langen Flows (Messung 3-2-1)

Beim Lab 3-2-1 wird die Messung nach einem ganzen Wochenende abgebrochen, da fest steht, dass die NetFlows nicht verloren gehen, wenn die CPU ausgelastet ist.
In den letzten Messrunden ist die Auslastung der CPU länger als 10 Minuten, was in der Realität kaum der Fall sein dürfte.

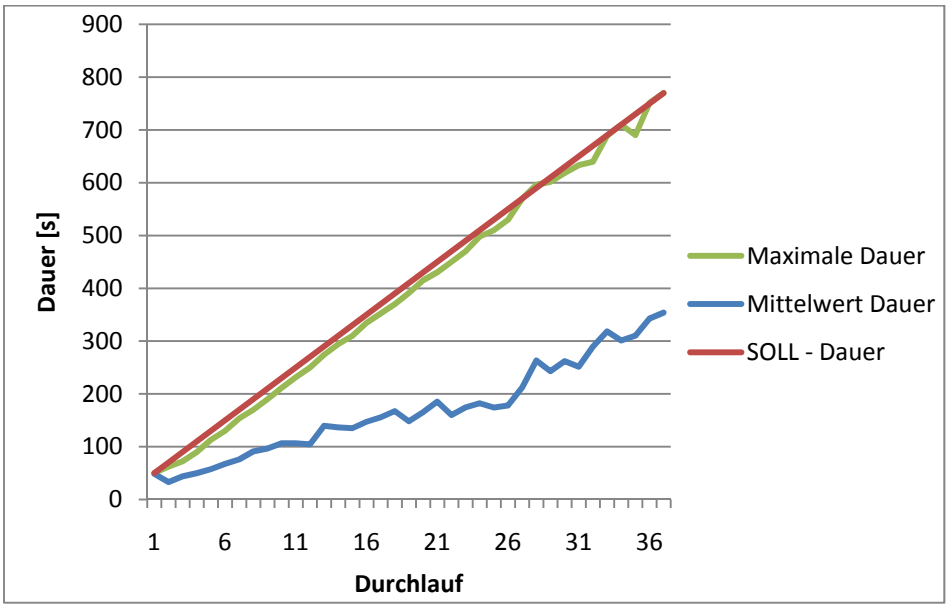


Abbildung 12-6 Flow-Dauer mit langen Flows (Messung 3-2-1) inkl. Der Soll-Dauer

Selbst der Mittelwert der Dauer beträgt im letzten Durchlauf etwa 350 Sekunden. Die Annahme kann also mit dieser Messung bestätigt werden.

12.4 Fazit

Der Router speichert trotz kritischer CPU-Auslastung die NetFlow-Informationen. Wenn jedoch keine CPU-Kapazität zum Verarbeiten vorhanden ist, wird einfach solange gewartet, bis dies wieder zutrifft. Es gehen also keinerlei Informationen der Verbindungen verloren.
Nach der Konfiguration des IOS sollten spätestens alle 60 Sekunden die aktiven Flows exportiert werden. Diese Einstellung wird jedoch ignoriert, wenn die CPU ausgelastet ist.

Mit der Langzeitmessung wird erreicht, dass der Router die Flows sogar über 10 Minuten lang speichert und die CPU keine Rechenzeit zur Verfügung stellt, um die NetFlows zu bearbeiten.

13 LAB 4 – NETFLOW-VERHALTEN BEI MEMORYAUSLASTUNG

13.1 Aufgabenstellung

13.1.1 Ziel

Es ist herauszufinden, was mit dem Netflow-Export oder der Netflow-Berechnung geschieht, wenn das Memory des Routers völlig überfüllt wird, so dass keine weiteren Einträge mehr gespeichert werden können (die Flow-Table wird im Memory gespeichert).

13.1.2 Bedingungen

Dem Router soll das Memory beinahe gefüllt werden, so dass dieser kein Platz mehr zur Verfügung hat, um die NetFlow-Daten zwischenzuspeichern.

Um den vorhandenen Speicherplatz des Memorys zu füllen kann ein grösseres Netzwerk angefügt werden um möglichst viele Routing Table Einträge zu erhalten, welche wiederum im Memory abgelegt werden.

Eine andere Möglichkeit ist das Memory physisch zu entfernen, dadurch sollte nur noch ein Minimum an Memory vorhanden sein.

13.1.3 Risiken / Challanges

Es ist nicht bekannt was der Router mit den Daten im Memory macht, wenn diese momentan nicht benötigt werden. Er könnte diese nach der Häufigkeit sortiert aus dem Memory entfernen (eine Art Garbage Collector). Somit wäre es praktisch nicht möglich, das Memory zu füllen, da immer wieder Bereiche mit der geringsten Häufigkeit gelöscht werden und somit wieder Speicherplatz zur Verfügung steht. Diese Reaktion könnte sich in der Stabilität des IOS bemerkbar machen.

13.2 Konfiguration

13.2.1 Aufbau

Vom PC-Client werden mit Iperf 497 Datenströme zum PC-Server erzeugt. Dies generiert 994 NetFlows im Router, welche 50 Minuten in der NetFlow-Table bleiben, da diese während dieser Zeit aufrechterhalten werden.

Der MessPC empfängt auf der VMWare die gesendeten NetFlow-Pakete vom Router. Gleichzeitig schickt dieser 1 Datenstrom an einen Virtuellen Server auf einer ESX-Umgebung der HSR, wartet 1 Minute und beginnt dann 2 Datenströme an den Server zu schicken. Dies wiederholt der MessPC, bis 31 Datenströme erreicht worden sind.

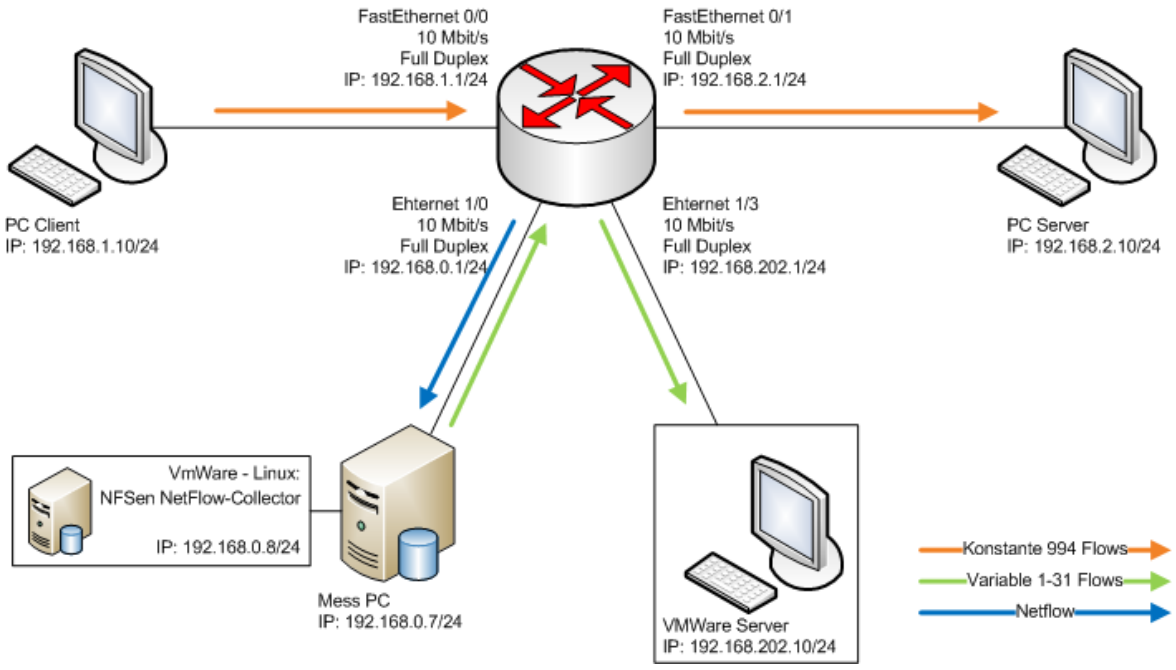


Abbildung 13-1 Aufbau

13.2.2 NetFlow-Konfiguration

Nach der Cisco Command Reference[40] können die Einträge der Flow-Table manuell eingestellt werden. Auf den meisten Router ist dieser Range zwischen 1024 und 524288 (ausser für Cisco ASR 1000 Series Aggregation Services Router).

```
ip flow-cache entries <1024-524288>
```

Ein Eintrag in der Flow-Table benötigt ca. 64 Bytes.

Vor jedem neuen Flow-Eintrag wird überprüft, wie viele Einträge noch in die NetFlow-Table passen.
Falls nur noch ein paar leere Einträge (Definition von Cisco selbst) vorhanden sind, werden 30 vorzeitig entfernt, resp. es gibt ein früheres Timeout der ältesten NetFlows.

Wenn nur noch 1 Eintrag zur Verfügung steht, werden sofort 30 NetFlows aus der NetFlow-Table entfernt, ungeachtet des Alters (!).

Die NetFlow-Einstellungen werden wie folgt geändert, damit die grösstmögliche Flow-Dauer (in Minuten) und die kleinstmögliche Flow-Table (Anzahl Flows) resultieren:

```
ip flow-cache entries 1024
ip flow-cache timeout active 60
```

13.2.3 SNMP-Abfrage

Für die automatische Wert-Abfrage von der Flow-Table ist SNMP prädestiniert.
Auf der Internetseite der Firma ByteView[41] lassen sich die OID's finden. Die Firma stellt auch die MIB-Tree's zum Herunterladen zur Verfügung. Die benötigte Tree heisst CISCO-NETFLOW-MIB.
Die OID für die Anzahl der ActiveFlows ist:

```
1.3.6.1.4.1.9.9.387.1.1.2.1.4
```

Leider existiert diese OID (resp. der ganze Ast ab 387) nicht. Der Fehler ist in der Abbildung 13-2 zu erkennen.

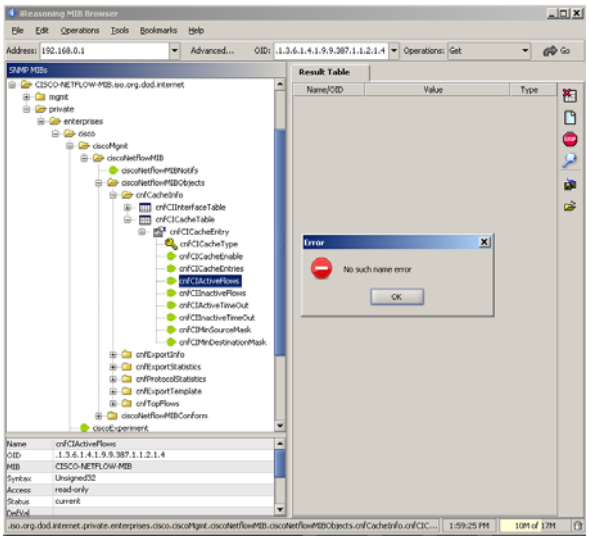


Abbildung 13-2 Fehlermeldung NetFlow OID

13.2.4 CLI-Abfrage

Auf der CLI des Routers lassen sich die Anzahl Flows (active, inactive, Konfigurationswerte, usw.) und sogar die einzelnen NetFlows anzeigen:

```
show ip cache flow
```

Um die NetFlow-Statistik zu löschen, wird folgender Befehl benötigt:

```
clear ip flow stats
```

13.2.5 Iperf-Scripts

Die grosse Herausforderung der Scripts ist, dass diese synchron ablaufen können. Nur so kann sichergestellt werden, dass keine abweichende Konstellation gemessen wird.

Mit dem DOS-Befehl `at` lassen sich Aufträge für den Scheduler erstellen. Dies hat leider nicht über das Netzwerk funktioniert und es muss eine Zeit für den Start übermittelt werden.

Ein weiteres Tool ist `soon.exe`, welches von Microsoft zum Download[42] angeboten wird. Mit diesem ist es möglich, die Startzeit relativ zu setzen, also z.B. nach 5 Sekunden nach absetzen des Befehls. Leider funktionierte es nicht wie gewünscht, da es automatisch einen ganzen Tag in der Zukunft den Startzeitpunkt ansetzte.

Der Entscheid ist auf den internen Scheduler gefallen. Von diesem lassen sich stündlich die Scripts ausführen, nachdem die Systemzeit manuell innerhalb der Toleranz (einige Sekunden) eingestellt wurde.

PC Client

Der Client sendet stündlich mittels IPerf 497 parallele TCP Verbindungen zum PC Server 1. Diese Verbindung besteht während 50 Minuten. Da es sich um TCP Verbindungen handelt, werden in der FlowTable 994 NetFlow-Einträge erstellt.

```
@echo off

REM //Variablen
set file="Lab4c_PC02.txt"

echo Messtart TCP >> %file%
echo %date% >> %file%
echo %time% >> %file%
echo ======================================== >> %file%

choice /d y /n /t 20
iperf -c 192.168.2.10 -p 4001 -P 497 -t 3000 >> %file%

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

PC Server

Auf dem PC Server 1 wird manuel ein IPerf Server gestartet. Damit die Verbindungen unterschiedlich zu den Verbindungen zwischen PC Mess und dem VMWare Server ist, wird Port 4001 benutzt. Die Befehlszeile muss wie folgt im Command Line Fenster eingegeben werden.

```
iperf -s -p 4001
```

PC Mess

Der Client startet das Script per Task Scheduler von Windows. Damit die kleine Ungenauigkeiten der Systemuhren vom Client und Server nicht stören, wird der Client für fünf Minuten angehalten. Erst danach wird eine Verbindung zum VMWare Server erstellt, welche während 40 Sekunden Daten sendet. Nach dem Senden wird eine Pause von 20 Sekunden dazwischen geschaltet bis der Client mit dem Senden fortfährt. Bei jedem Durchgang wird beim Client ein Thread mehr erzeugt. So werden immer mehr parallele Verbindungen zwischen Client und Server erzeugt.

```
@echo off
set threads=1
set file="lab4_MessPC.txt"
echo %date% >> %file%
echo %time% >> %file%
echo ========================= >> %file%
choice /n /d y /t 300


:START
if %threads% GTR 31 goto NEXT
echo threads=%threads% >> %file%
iperf -c 192.168.202.10 -t 40 -p 5001 -P %threads% >> %file%
choice /n /d y /t 20
set /a threads=%threads% + 1

goto START

:NEXT
```

VMWare Server

Der VMWare Server muss nur die gesendeten TCP-Verbindungen entgegennehmen. Bei ihm wird manuell ein IPerf Server auf dem Port 5001 eröffnet. Dazu muss folgender Befehl ins Command Line Fenster eingegeben werden.

```
iperf -s -p 5001
```

13.3 Testresultate

13.3.1 Erwartet

Es wird erwarten, dass die NetFlows unvollständig erfasst werden.

13.3.2 Gemessen

Über die 20 Durchläufe mit je 1-31 Verbindungen haben sich insgesamt 74222 Flows ergeben. Rein rechnerisch sollten jedoch nur 1459 pro Durchlauf, also gesamthaft 29180 Flows resultieren. Die 1459 setzen sich durch die 994 konstanten Flows und den variablen ([n * (n-1)/2)] wobei n=31, entspricht 465) zusammen.

Nach der Analyse wird festgestellt, dass der Verbindungsaufbau sehr viele zusätzliche NetFlows generiert. Dieser lässt sich für die konstanten Flows einfach herausfiltern, wohingegen der Verbindungsaufbau der variablen Flows das Resultat verfälscht.
Aus diesem Grund kann die Auswertung nicht automatisch, sondern muss manuell ermittelt werden.

13.4 Fazit

Nach der Dokumentation von Cisco kann davon ausgegangen werden, dass für die NetFlows in der NetFlow-Table Platz reserviert wird. Diese Grösse lässt sich konfigurieren. Somit erübrigen sich ein Ausbau des Memorys sowie auch die Simulation eines grossen Netzes mit GNS3[43], um die Routing-Table zu vergrössern und somit Memory zu verbrauchen.

Ebenfalls wird während der Messung bemerkt, dass die Verbindungen zum VMWare Server ab und zu unterbrochen und wieder neu aufgebaut werden. Da unbekannt ist, wie das Netzwerk zwischen Router und VMWare Server und auch die Netzwerkimplementation von VMWare programmiert wurde, bildet es eine Blackbox, in der die Störungsursache nicht eruiert werden kann.

Aus der Erkenntnis vom Kapitel Testresultat, wird beschlossen, die Messung im Lab 4b zu wiederholen. Es sollten andere Tools eingesetzt werden, damit der Aufwand für die Analyse in einem vernünftigen Rahmen bleibt.

14 LAB 4B – NETFLOW-VERHALTEN BEIM ERREICHEN DER FLOW-TABLEGRENZE

14.1 Aufgabenstellung

14.1.1 Ziel

Es wird überprüft, ob die Konfiguration der NetFlow-Einstellungen, wie in der Dokumentation beschrieben, funktioniert.

Vor jedem neuen NetFlow-Eintrag in die NetFlow-Tabelle wird überprüft, wie viele Einträge noch aufgenommen werden können.
Falls nur noch ein paar leere Einträge (Definition von Cisco selbst) vorhanden sind, werden 30 vorzeitig entfernt, resp. es gibt ein früheres Timeout der ältesten NetFlows.
Wenn nur noch 1 Eintrag zur Verfügung steht, werden sofort 30 NetFlows aus der NetFlow-Table entfernt, ungeachtet des Alters.

14.1.2 Bedingungen

Die NetFlow-Table soll bis vor den 30 letzten NetFlow-Einträge gefüllt werden. Somit kann danach mit weiteren Flows überprüft werden, wie sich der Router verhält.

14.1.3 Risiken / Challenges

Es wird keine Alternative für Iperf gefunden oder der Verbindungsaufbau verhält sich gleich wie bei Lab 4 und damit wird die Analyse sehr aufwendig.

14.2 Konfiguration

14.2.1 Aufbau

Vom PC-Client 1 werden mit Iperf 497 TCP-Datenströme zum PC-Server erzeugt. Dies generiert 994 NetFlows im Router (da TCP hin und zurück aus 2 NetFlows besteht), welche 50 Minuten in der NetFlow-Table bleiben, da diese während dieser Zeit aufrechterhalten werden.

Der MessPC empfängt auf der VMWare die gesendeten NetFlow-Pakete vom Router. Gleichzeitig schickt dieser mit D-ITG zwischen 1 und 33 Datenströme.

PC-Client2 ist über einen unmanagable Switch angeschlossen. Da eine direkte Verbindung fehlgeschlagen ist. Es herrscht eine Inkompatibiltät des Interfaces am Router und der Netzwerkkarte des Computers. Auf Layer 1 ist eine Verbindung vorhanden, während auf Layer 2 keine ARP's beantwortet werden.

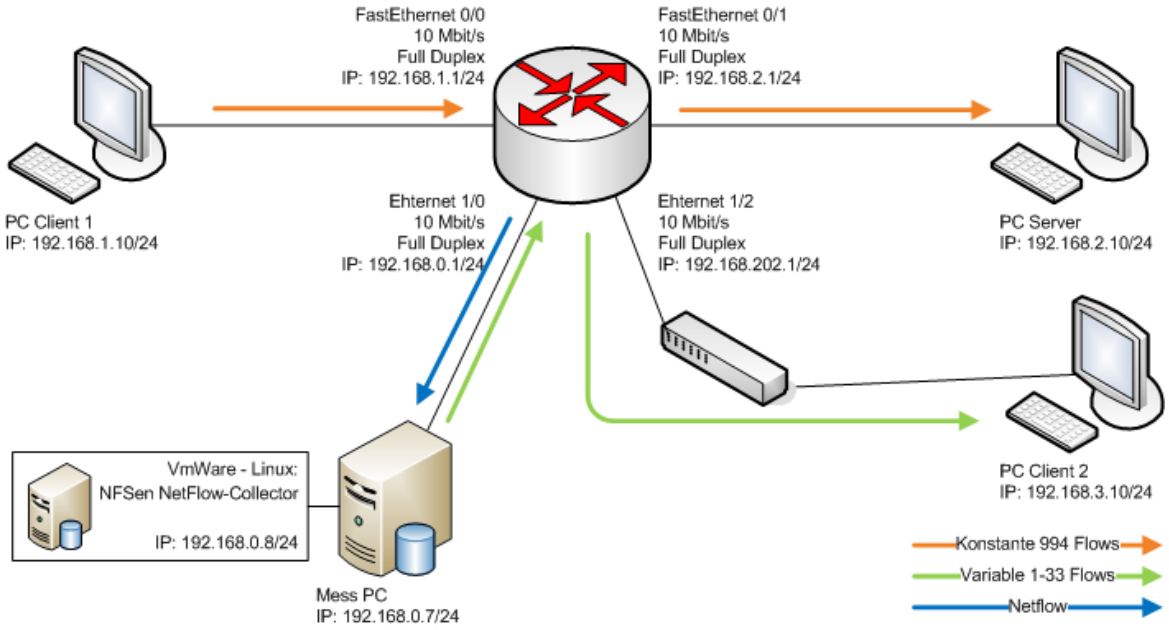


Abbildung 14-1 Aufbau

14.2.2 NetFlow-Konfiguration

Die maximale Grösse der NetFlow-Table wird auf 1024 beschränkt. Der aktive Timeout, welcher noch aktuelle NetFlows in der NetFlow-Table hält, wird auf 60 Minuten eingestellt.

```
ip flow-cache entries 1024
ip flow-cache timeout active 60
```

14.2.3 Scripts

PC Client 1

Es wird das gleiche Script wie beim Lab 4 benutzt. Der PC Client 1 sendet 497 TCP-Verbindungen mit Iperf zum PC Server. Womit im Router insgesamt 994 NetFlow-Einträge aufgenommen werden.

```
@echo off

REM //Variablen
set file="Lab4c_PC02.txt"

echo Messtart TCP >> %file%
echo %date% >> %file%
echo %time% >> %file%
echo ======================================== >> %file%

choice /d y /n /t 20
iperf -c 192.168.2.10 -p 4001 -P 497 -t 3000 >> %file%

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

PC Server

Der Iperf Server nimmt die TCP Verbindungen des PC Clients 1 entgegen. Falls es während der Messung zu abstürzen kommt, wird der Server stündlich neu gestartet und ein noch laufender Prozess mit dem Namen "iperf.exe" abgeschossen.

```
@echo off

REM //Variablen
set file="Lab4c_PC03.txt"

echo Messtart TCP >> %file%
echo %date% >> %file%
echo %time% >> %file%
echo ======================================== >> %file%

taskkill /IM iperf.exe /T /F
choice /d y /n /t 5
iperf -s -p 4001

echo ======================================== >> %file%
echo Messende >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

PC Client 2

Der PC Client 2 sendet mit dem Script "lab4_send.cmd" immer mehr UDP Verbindungen. Im ersten Durchgang wird nur eine UDP Verbindung erstellt. Das Script erhöht die Verbindungen jeweils um 3 weitere UDP Verbindungen pro Durchgang. Die Verbindungen werden bis 33 UDP Verbindungen erhöht. Bei 30 Verbindungen ist die NetFlowtable komplett mit 1024 Einträgen gefüllt.

```
@echo off

REM \\Variablen
set file="lab4c_PC01.txt"
set runs=0

choice /n /d y /t 300

echo Start Messung >> %file%
echo %date% >> %file%
echo %time% >> %file%

:START
if %runs% GTR 33 goto NEXT

echo ======================================== >> %file%
echo lab4c_PC01_%runs% >> %file%
echo %time% >> %file%
taskkill /IM ITGSend.exe /T /F
choice /n /d y /t 5
ITGSend.exe lab4-udp_%runs%.itg -l lab4_sendlog

set /a runs=%runs% + 3
choice /n /d y /t 60
ITGDec.exe lab4_sendlog >> %file%

goto START
:NEXT

echo ======================================== >> %file%
echo ENDE Messung >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

MessPC

Der Mess PC nimmt die inkrementell erhöhten UDP Verbindungen des PC Client 2 entgegen. Dieser braucht die Datenpakete nur zu empfangen, weshalb ein scheduled Task eingerichtet wird, welcher die Batchdatei jede Stunde startet.

```
taskkill /IM itgrecv.exe /T /F
choice /n /d y /t 20
itgrecv.exe
```

14.3 Testresultate

14.3.1 Erwartet

Die Definitionen von Cisco, im Kapitel Ziel aufgeführt, werden eingehalten. Die NetFlows, welche in die NetFlow-Table aufgenommen werden sollen, veranlassen, dass ältere NetFlows frühzeitig exportiert werden.

14.3.2 Gemessen

Da die Abfrage über die NetFlow-Table über SNMP nicht funktioniert, fehlt dadurch eine Referenz. Die SNMP-Abfrage kann nicht gesendet werden, weil die entsprechende OID nicht vorhanden ist. Wie dies schon in Lab4 beschrieben wurde.
In Wirklichkeit sind immer 994 konstante Datenströme vorhanden, jedoch muss der Router diese NetFlow-Einträge aus der Tabelle entfernen, damit die zusätzlich gesendeten (variablen) NetFlows eingetragen werden können. Danach müssen aber die konstanten Datenströme, welche frühzeitig exportiert wurden, wieder in der Tabelle eingetragen werden, da diese ja noch vorhanden sind. Durch diese Konstellation sind nun 2 unbekannte Variablen vorhanden, welche nicht genau erörtert werden können.
Somit lassen sich die Messungen über die Richtigkeit des Exportes anhand des Alters des NetFlows nicht mit vernünftigem Aufwand realisieren. Allerdings können andere Erkenntnisse gewonnen werden.

Laut Dokumentation werden, sobald der Platz in der NetFlow-Table mehr als 994 von 1024 NetFlow-Einträgen übersteigt, 30 NetFlows frühzeitig exportiert. Dies kann nicht bestätigt werden. Der Router hat nie exakt 30 NetFlows exportiert, sondern eine beliebige Anzahl.

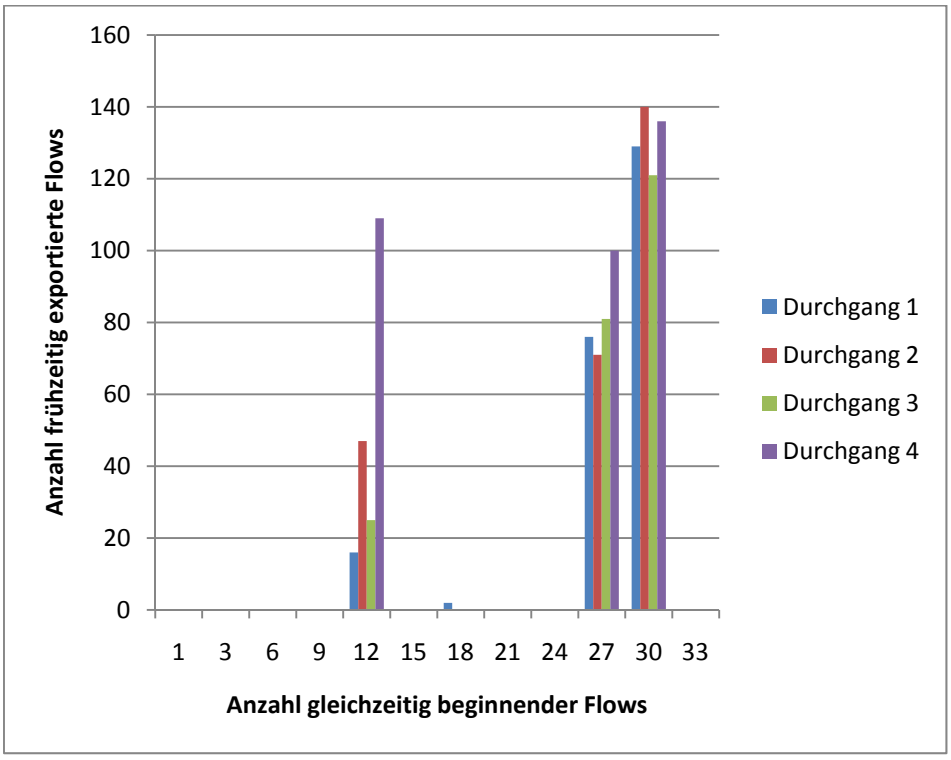


Abbildung 14-2 Anzahl frühzeitig exportierte NetFlows

In der Abbildung 14-2 ist zu erkennen, dass der Exportzeitpunkt ähnlich ist, allerdings ist die Anzahl der exportierten NetFlow-Einträgen nie 30 oder ein Vielfaches davon.

Eine wichtige Eigenschaft wäre, dass bei einem frühzeitigen Export von einem NetFlow, welches eine TCP-Verbindung beschreibt, auch beide dazugehörenden NetFlows entfernt werden, damit wenigstens die anderen der Realität entsprechen.
Diese Eigenschaft kann nur für die ersten beiden Exporte vom Durchgang 1 bestätigt werden. Alle weiteren Exporte liefern verschiedene Verbindungen und sogar eine ungerade Anzahl an exportierten NetFlows.

Das frühzeitige Exportieren führt zu einer falschen Auswertung über die Anzahl der NetFlows als in Wirklichkeit vorhanden sind. Das gleiche Problem taucht auch auf, wenn der Wert des Aktive-NetFlow-Timeout zu kurz gewählt wird.

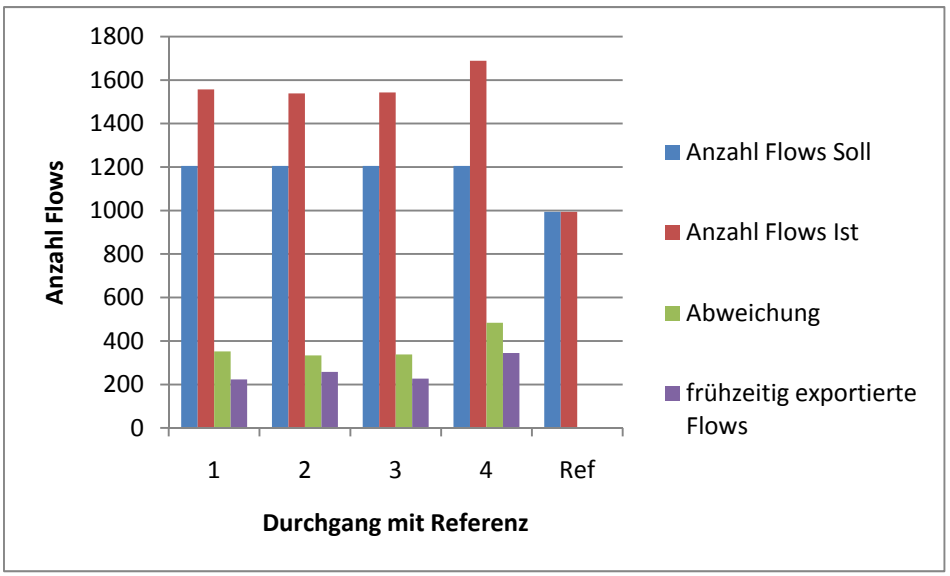


Abbildung 14-3 Anzahl NetFlows verglichen mit den Flows in Realität

In der Abbildung 14-3 ist dies gut ersichtlich. Bei der Referenz-Messung stimmt der Wert überein, da nur die konstanten 994 Datenströme vorhanden sind und somit die NetFlow-Table noch 30 freie Einträge hat und das IOS in diesem Fall nichts unternimmt. Dies entspricht auch der Definition von Cisco.

Interessant ist nun auch die Gesamtdauer der NetFlows gegenüber der Realität zu messen, da es wohl Unterschiede geben muss, wenn viele NetFlows frühzeitig exportiert werden müssen.

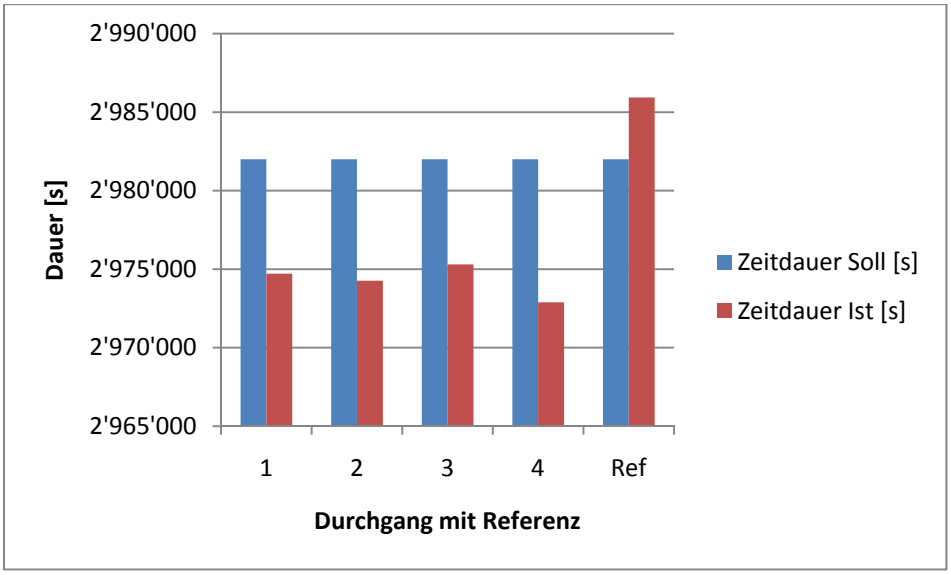


Abbildung 14-4 Gesamtdauer NetFlows verglichen mit den Flows in Realität

In der Abbildung 14-4 ist zu erkennen, dass die NetFlows zu wenig Zeit reporten, als in Wirklichkeit sein müsste. Bei der Referenz-Messung ist aufgefallen, dass die effektive Zeit höher liegt, als bei der Software Iperf eingestellt wurde. Dies kann durch die Ungenauigkeit von Iperf erklärt werden.
Die Differenz wird dadurch jedoch noch grösser und ist in der Tabelle 14-1mit dem Unterschied in Prozent aufgeführt.

Durchgang:	1	2	3	4	Referenz
Zeitdauer Soll (994*50min *60) [s]	2982000	2982000	2982000	2982000	2982000
Zeitdauer Ist [s]	2974707	2974265	2975301	2972885	2985934
Abweichung [s]	7293	7735	6699	9115	-3934
Abweichung zur Referenz [%]	0.244	0.259	0.224	0.305	-

Tabelle 14-1 Abweichung der Sendezeitdauer

Da die Abweichung sich im Promillebereich befindet ist die Überwachung und Analyse des Netzwerkes mit NetFlow genügend exakt.

14.4 Fazit

In den Messungen von Lab4b geht hervor, dass die Auswertung stark von den Einstellungen vom NetFlow-Export abhängig ist. Die Standardeinstellung von Cisco (4096 Einträge) sollte je nach Grösse des Netzwerkgerätes und der erwarteten Flows geändert werden.
Ebenfalls sollte der inactive NetFlow-Timeout (Standard: 15 Sekunden) sowie der active NetFlow-Timeout (Standard: 30 Minuten) den Bedürfnissen angepasst werden.
Falls die Auswertung möglichst aktuell gehalten muss, sollten kleinere Werte gewählt und falls die Auswertung möglichst genau ausfallen soll, grössere Werte gewählt werden.
Wenn also basierend auf den NetFlows Abrechnungen erstellt werden sollen, ist zu empfehlen, den active Timeout auf den Maximalwert von 60 Minuten und die NetFlow-Table auf den Maximalwert von 524288 Einträge zu stellen (auf den Cisco ASR 1000 Series Aggregation Services Router ist sogar ein Wert von 2‘000‘000 möglich).

Wenn der Router nun NetFlows frühzeitig exportieren muss, weil kein Platz für einen neuen NetFlow-Eintrag mehr vorhanden ist, können bei TCP-Verbindungen Unregelmässigkeiten bei der Auswertung entstehen, da der Router auch nur den NetFlow in eine Richtung frühzeitig exportieren kann.
Im schlimmsten Fall können in eine Richtung korrekterweise ein NetFlow für eine TCP-Verbindung existieren und für die andere Richtung fälschlicherweise mehrere.
Dies führt zu einer Inkonsistenz der Gesamtdauer, da Teile der Verbindung nicht aufgezeichnet sind.

15 LAB 5 – ZEITVERHALTEN DER NETFLOW-PAKETE

15.1 Aufgabenstellung

15.1.1 Ziel

Es ist herauszufinden, ob die Timestamps der Wirklichkeit entsprechen. Wann wird eine solche Timestamp in das Paket geschrieben.

Wenn ein NetFlow abgeschlossen wird, steht der Timestamp vom Anfang des Paketes oder vom Schluss, wenn das letzte Byte des Paketes geroutet wurde, drin.

15.1.2 Bedingungen

Die Zeit muss exakt auf allen beteiligten Geräten stimmen. Dies kann über eine Einschubkarte erreicht werden, welche das Funk-Uhr Signal (DCF77) empfangen kann, über NTP oder sonstige Technologien zur Uhrensynchronisation.

15.1.3 Risiken / Challenges

Die Zeiten können nie exakt auf allen Geräten synchron laufen. Diese Ungenauigkeit wird den Wert des Resultates beeinflussen.

Die Zeitunterschiede können zu ungenau erfasst werden. Dadurch können Differenzen nicht erkannt und ausgewertet werden.

Es stellt sich die Kostenfrage, falls eine Karte oder sonst etwas angeschafft werden muss.

15.2 Konfiguration

15.2.1 Zeit-Systeme

Um die lokale Zeit zu synchronisieren gibt es verschiedene Systeme, welche nachfolgend kurz beschrieben sind.

DCF77 (Funkuhr)
Der Langwellensender ist in Mainflingen (D) stationiert und übermittelt für die meisten europäischen Funkuhren[44] die UTC-Zeit.

HBG (Zeitzeichensender der Schweiz)
Das Bundesamt für Metrologie[45] (METAS) hat den Auftrag, unter anderem die grundlegende Masseinheit Sekunde zur Verfügung zu stellen.
Mehrere Atomuhren werden dazu betrieben und als Langwellen die Mitteleuropäische Zeit (MEZ) bzw. Mitteleuropäische Sommerzeit (MESZ) übermittelt.
Der Sender wird allerdings auf Ende 2011 eingestellt[46].

GPS (Global Positioning System)[47]

Das bekannteste System läuft seit 1990 und seit Mai 2000 mit der Abschaltung der künstlichen Signalverschlechterung. GPS ersetzt viele Systeme (z.B. Navigationssystem der US-Marine, Ortung von Atombombenexplosionen).
Die Genauigkeit der Positionsbestimmung lässt sich mit der Differenzmethoden (dGPS) auf Zentimeter steigern.
Jeder Satellit strahlt Signale mit der aktuellen Position und die genaue Uhrzeit aus. Aus den Signallaufzeiten können die GPS-Empfänger ihre eigene Position und die Geschwindigkeit errechnen.

GOES (Geostationary Operational Environmental Satellite)[48]

Dieses System ist eine Serie von geostationären Wettersatelliten der amerikanischen Wetterberhörde[49] (NOAA). Es ermöglicht eine Überwachung des Wetters und wird zur Vorhersage von Hurrikans und zur meteorologischen Forschung benutzt.

LORAN (Long Range Navigation)[50]

Dieses Funknavigationssystem wird vorwiegend für die Luft- und Seefahrt verwendet.
Jede Sendestation übermittelt ein bestimmtes Schema und aus der zeitlichen Differenz der Signallaufzeit, kann die Position errechnet werden.

NTP (Network Time Protocol)[51]

NTP ist ein Standard zur Synchronisation der Uhren in Computersystemen. Die Kommunikation läuft über UDP.
Anhand des Zeitstempels im NTP kann die interne Uhr synchronisiert werden. Dieser ist 64 Bit lang und hat eine Auflösung von 0.25ns. Meistens werden mehrere NTP-Server angefragt.
Server sind z.B.:

- pool.ntp.org (resp. ch.pool.ntp.org)
- metas.ch
- time.ethz.ch

System	Genauigkeit	Weitere Genauigkeit
DCF77	± 1 ms	± 10 ms bei atmosphärischer Störung
GPS	± 1 ms	± 1 µs vom Satellit aus
GOES	± 0.1 ms	
NTP	± 10 ms (Internet)	± 200 µs und genauer im LAN

Tabelle 15-1: Genauigkeiten Zeitsynchronisationssysteme

Wie in der Tabelle 15-1 ersichtlich ist die Genauigkeit innerhalb des LAN mit NTP sehr gross. Die Problematik ist die richtige Zeit von anderen Systemen (mit einer Atomuhr) zu erfassen. Auch wenn NTP den schlechtesten Wert besitzt, lohnt es sich kaum Hardware anzuschaffen, welche eine bessere Genauigkeit bringen könnte.
Selbst eine günstige Einbaukarte für das DCF77-Signal hätte eine Toleranz von 50 ms, was bedingt, teureres Equipment anzuschaffen. Der Nutzen wäre allerdings relativ klein.

15.2.2 Aufbau

Vom PC-Client wird eine TCP-Verbindung mit D-ITG zum PC-Server eröffnet. Der MessPC empfängt auf der VMWare die gesendeten NetFlow-Pakete vom Router.

Der Router wird als NTP-Server eingerichtet und fragt über das Internet-Interface die aktuelle Zeit von einem externen NTP-Server ab, welcher seine Zeit von einer Atomuhr erhält.
Alle anderen beteiligten Geräte erfragen die Zeit beim Router.

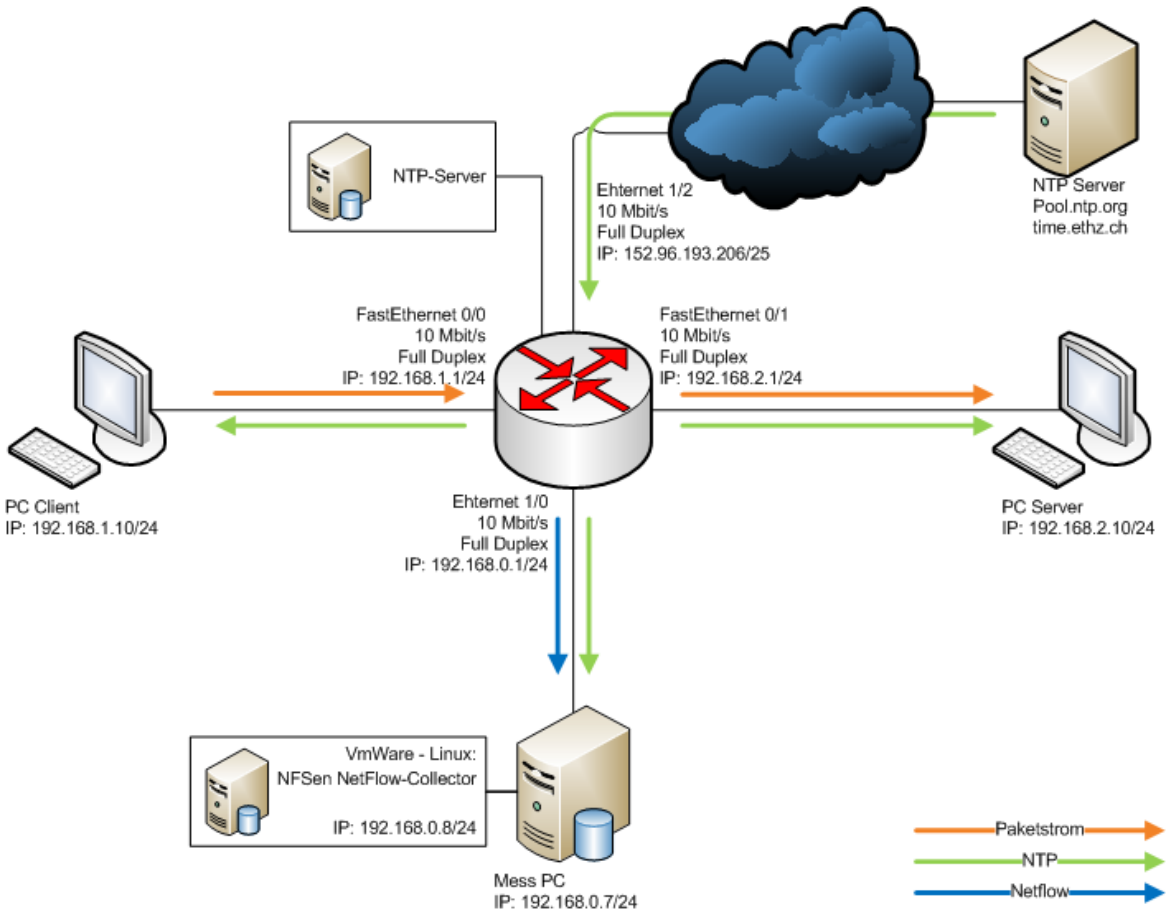


Abbildung 15-1 Aufbau

15.2.3 NTP-Konfiguration

Router

Der Router soll seine Zeit von einem externen Server holen und diese dann selbst als NTP Server an die Computer weitergeben. Wichtig ist, dass das Interface, welches am HSR-LAN angeschlossen ist, als Quelle definiert wird, da die anderen Adressen der Interfaces nicht geroutet werden.

```
ntp server 192.33.96.102
ntp server 129.132.97.15
ntp source Ehternet 1/2
```

Mit `show ntp associations` lassen sich die Konfiguration und der Status abfragen.

```
DA_BA#show ntp associations

      address         ref clock     st  when  poll reach  delay  offset    disp
*~192.33.96.102    .PPS.            1    16    64  377     3.4   10.07     0.2
 ~129.132.97.15    0.0.0.0         16     -  1024    0     0.0    0.00  16000.
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured
DA_BA#_
```

Abbildung 15-2 show ntp associations

Feld	Bedeutung
St	Stratum des NTP-Servers
When	Zeit, wann das letzte NTP-Paket empfangen wurde.
Poll	Sind die Anzahl Sekunden zwischen den einzelnen NTP-Abfragen. Je besser die Synchronisierung, desto höher wird der Wert (bis zu 1024).
Reach	Ist eine Binärdarstellung (377 = 11111111). Heisst, dass die letzten 8 Abfragen vom NTP-Server beantwortet wurden.
Delay	Round-Trip-Delay zum NTP-Server
Offset	Berechnete Differenz in Milisekunden zwischen der Zeit im Router und der NTP-Zeit.
Disp	Dispersion: Repräsentiert den maximalen Fehler relativ zum NTP-Server

Tabelle 15-2: Bedeutung der Felder bei ntp associations

Windows

Alle beteiligten Computer sollen die Zeit vom Router abfragen. Dies soll alle 5 min geschehen, dafür ist ein Eingriff in die Registry nötig (da sonst der kleinste Wert 60 min beträgt).

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Time
Providers\NtpClient]
"SpecialPollInterval"=dword:0000012c
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Para
meters]
"NtpServer"="192.168.0.1,0x1"
```

Manuell muss noch eingestellt werden, dass Windows einen NTP-Server ansprechen soll.

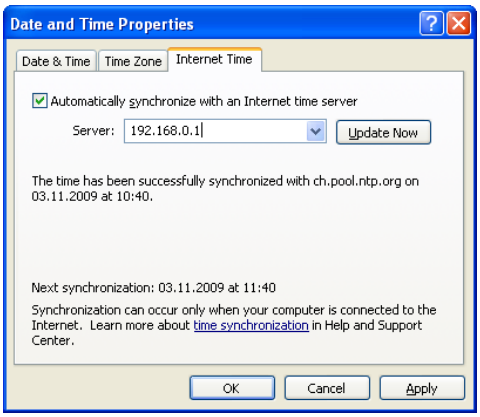


Abbildung 15-3 NTP bei Windows aktivieren

Linux

Mit folgendem Befehl lässt sich die Zeit an den Router anpassen.

```
ntpdate 192.168.0.1
```

Das Problem bei `ntpdate` ist, dass die Systemzeit direkt neu gesetzt wird. Es wäre also möglich dass bei grösseren Sprüngen, die gleiche Zeit 2 Mal existiert, was zu Konflikten z.B. in Log-Dateien führen kann.
Der Dienst `ntpd` würde dies korrekt darstellen, indem er die Uhrzeit beschleunigt oder abbremst, bis sie zur Referenzzeit identisch ist.

Bei diesem Lab wird trotzdem `ntpdate` genutzt, da dieses in Scripte eingebaut werden kann.

15.2.4 Scripts

PC Client

Der PC Client sendet mit D-ITG eine UDP-Verbindung zum PC Server. Die Verbindung wird um eine Minute verzögert, damit sichergestellt ist, dass der PC Server bereit für den Empfang ist. Der PC Client sendet genau ein UDP-Paket pro Minute zum PC Server.

```
@echo off

REM \\Variablen
set file="lab5_tcp.txt"
set runs=0

echo Start Messung > %file%
echo %date% >> %file%
echo %time% >> %file%

:START
if %runs% GTR 65535 goto NEXT

echo ======================================== >> %file%
```

```
echo lab5_%runs% >> %file%
echo %time% >> %file%

ITGSend.exe -a 192.168.2.10 -T UDP -t 1000 -d 60000 -C 1 -c 100 -l
lab5_sendlog

set /a runs=%runs% + 1

ITGDec.exe lab5_sendlog >> %file%

goto START
:NEXT

echo ======================================== >> %file%
echo ENDE Messung >> %file%
echo %date% >> %file%
echo %time% >> %file%
```

PC Server

Der PC Server startet Wireshark. Danach wird der Reciever von D-ITG gestartet und die Zeit mit dem Befehl `w32tm /resync /nowait` aktualisiert. Das ganze Script wird per Scheduled Task auf dem PC zeitlich gesteuert.

```
@echo off

REM \\Variablen
set file="lab5_4_receiver.txt"
set runs=1

echo Start Zeitmessung

REM taskkill /IM ITGRecv.exe /T /F

echo = %time% ====================================
start wireshark.exe -i 4 -b duration:900 -w lab5_2_server.pcap -k
-Q

REM \\ITG Receiver
start itgrecv.exe

choice /d y /n /t 10
w32tm /resync /nowait
choice /d y /n /t 10
w32tm /resync /nowait
choice /d y /n /t 10
w32tm /resync /nowait
choice /d y /n /t 10
w32tm /resync /nowait
choice /d y /n /t 10
w32tm /resync /nowait
choice /d y /n /t 10


echo = %time% ====================================
echo Ende Zeitmessung
```

15.3 Testresultate

15.3.1 Erwartet

Die Timestamps, welcher der Router setzt sollten nach der internen Uhr des Routers entsprechen.

Bei den NetFlows sind wahrscheinlich die Zeiten aufgeführt, wenn das letzte Paket eines Flows den Router verlassen hat.

15.3.2 Gemessen

Laut Unterlagen von Cisco[52] wird ein TCP-Paket anhand des FIN-Bits erkannt und somit der Flow exportiert.

Bei der Messung werden die Startzeit vom PC2 und die Endzeit vom Server mit den NetFlow-Zeiten, welche am MessPC ermittelt werden, verglichen.

Hinzugerechnet wird noch ein Delay, welcher in der Tabelle 15-3 eruiert wird.

Art des Delays	Zeit	Einheit	Anzahl	Kommentar
Transmission (Serialization) Time	51.2	µs	2	Für ein 64 Byte Paket mit 10 MBit/s für das Interface am PC und am Router
Queuing / Buffering beim Router (variabel, worst case)	8	ms	1	Als Durchschnitt wird 1 ms angenommen
Kabellänge (Signallaufzeit)	87.179	µs	15	Formel: 0.65 * c bei 10-100 MHz ergibt die Zeit pro Meter (insgesamt 15m (10 + 5))
Gesamt-Delay	**2.41**	**ms**		

Tabelle 15-3: Delay-Berechnung

Die Analyse ergibt, dass der Unterschied der Abweichung sich unregelmässig vergrössert. Bei diesem Versuch ist die Zeitdauer zu gering, um Korrekturen von NTP zu erkennen.

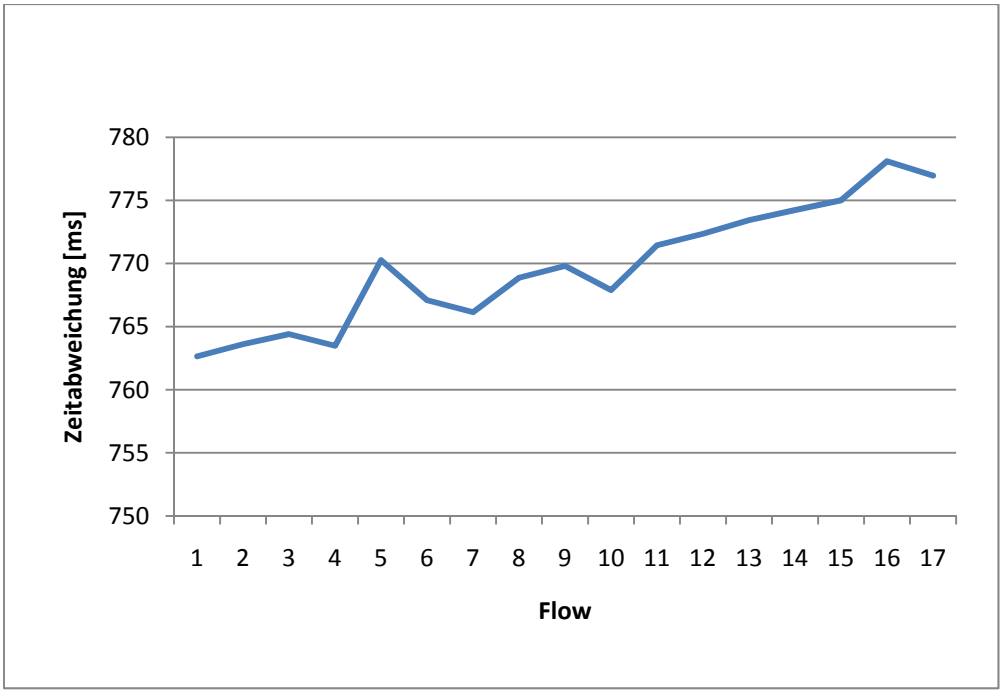


Abbildung 15-4 Unterschied der Timestamps von NetFlow und der Capture-Files

Der konstante Unterschied der Start- und Endzeit beträgt 762 ms. Es wird angenommen, dass dieser Unterschied sich durch die Zeitabweichung ergibt. Dass der Graph unregelmässig steigt, kann mit der unterschiedlichen Geschwindigkeit der Zeit auf den Computern erklärt werden. Um diese Annahmen zu stärken wird eine weitere Messung über eine längere Zeit (ca. 7 Stunden) durchgeführt.

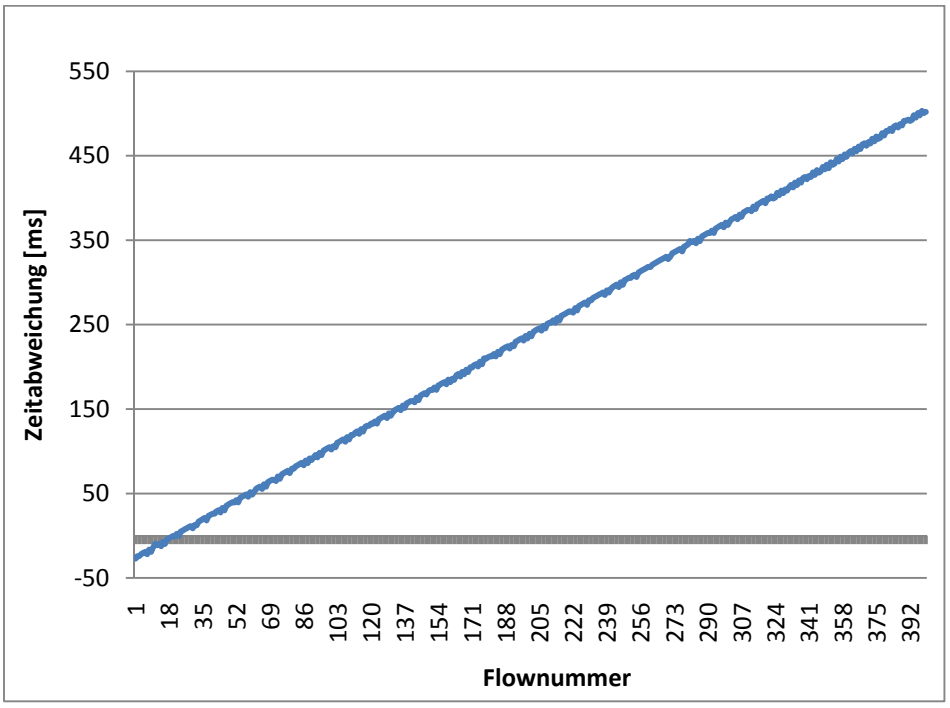


Abbildung 15-5 Messung 5-1 Zeitunterschied über 400 Minuten

Entgegen unseren Annahmen läuft die Zeit weiter konstant auseinander. NTP sollte eigentlich die Zeit auf dem jeweiligen Computer synchronisieren, jeweils jede halbe Stunde.

Es könnte folgende Gründe dafür geben:

- NTP-Server gibt auf die Requests keine Antwort
- Windows synchronisiert die Zeit nicht zuverlässig (nach Erhalt der NTP Antwort)
- Wireshark nimmt nicht die Systemzeit, sondern arbeitet mit einer anderen

Diese Aussagen werden nun genauer analysiert:

- Mit Wireshark lässt sich ansehen, ob die Pakete beantwortet werden. Dieses Problem kann nun ausgeschlossen werden, da die Requests und die Antworten mit Wireshark ersichtlich sind.
- Das GUI der Systemzeit reagiert sehr träge und arbeitet unzuverlässig. Bei einem manuellen Update über NTP meldet das GUI sehr oft Fehler (An error occurred while Windows was synchronizing with 192.168.0.1. The Peer is unreachable). Die Antworten sind jedoch bei Wireshark ersichtlich und kommen auch an.
  Es ist damit zu rechnen, dass Windows zwar die neue Zeit erhalten hat, diese aber nicht zur Synchronisation nutzt. Nach Anpassung des `UpdateInterval` auf 300 in der Registry passt Windows die Zeit zuverlässiger an.
  Mit dem Befehl `w32tm /stripchart /computer:192.168.0.1` lassen sich die Ungenauigkeiten anschauen, welche zum Teil mehrere Sekunden betragen können.
  Nach einem forcierten Synchronisieren braucht Windows ca. 30 Sekunden, bis die Zeit angepasst ist.
- Während einer Aufzeichnung der Pakete mit Wireshark wird die Zeit manuell verändert. Die aufgezeichneten Pakete sind allerdings mit der alten Zeit versehen. Dies bedeutet, dass Wireshark nicht die Systemzeit nimmt.
  Um dies zu bestätigen wird auch WinDump[53] getestet. Jedoch auch bei dieser Software ergibt sich das gleiche Problem. Beide basieren auf der PCAP-Library[54], möglicherweise liegt das Problem dort.

Um zu bestätigen, dass Wireshark den Timestamp aus der PCAP-Library nimmt und diese nur beim Beginn einer Aufzeichnung die Zeit vom Betriebssystem nimmt, wird eine zusätzlich Messsung 5-2 vorgenommen. Bei dieser werden zuerst die Uhren synchronisiert, danach werden in einem Abstand von 1 Minute 10 Pakete verschickt. Anschliessend wird das Capture-File geschrieben und Wireshark beendet. Nach 20 Minuten beginnt es von vorne, dies für 15 Durchführungen.

In der Abbildung 15-6 ist nun sehr gut ersichtlich, wann die Uhr wirklich synchronisiert wird und auch die einzelnen Durchführungen sind zu erkennen, weil bei einer neuen Wireshark-Instanz die neue Zeit vom Betriebssystem ausgelesen wird.

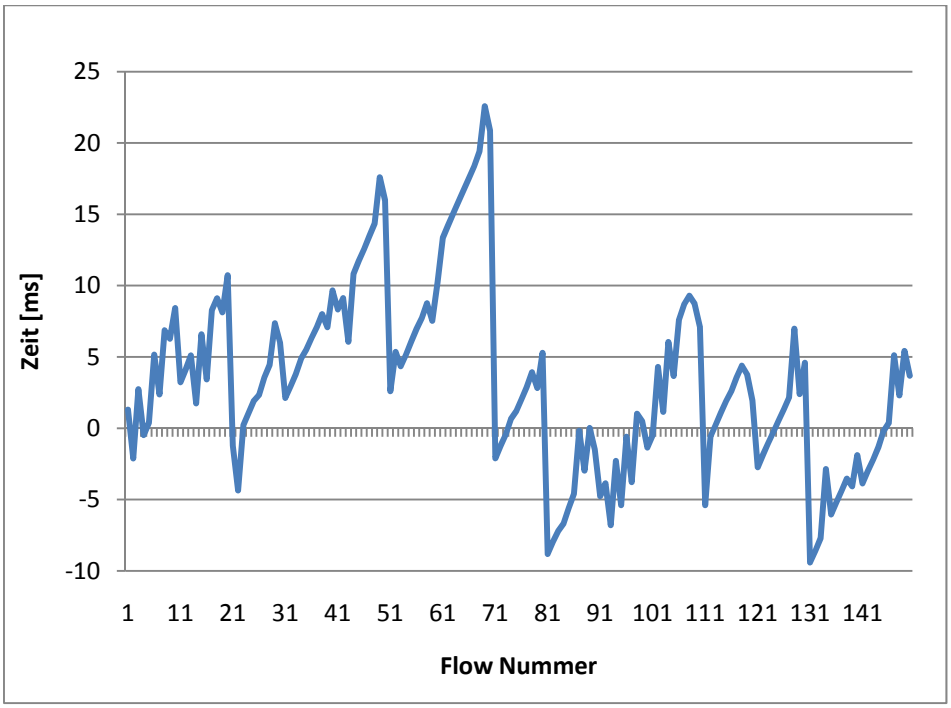


Abbildung 15-6 Zeitdifferenz zwischen NetFlow und Effektiv mit NTP und forcierter Zeitsynchronisation beim OS (Messung 5-2)

Anhand dieser Grafik kann man erkennen dass zwischen dem 1. Und 2. Durchlauf (bei Flownummer 11) keine Zeitsynchronisation stattgefunden hat, allerdings die Zeit vom Betriebssystem neu abgefragt wird.
Bei der Flownummer 21 hat das Betriebssystem die Zeit angepasst, wie auch bei 51, 71 und bei 81 sehr gut ersichtlich ist.

15.4 Fazit

Der Abschluss eines NetFlows ist in der Cisco-Dokumentation beschrieben. Sobald das FIN-Bit bei TCP gesetzt wird, kann der Router den NetFlow abschliessen. Auch bei einem Fehler (RST-Bit) wird auf diese Art verfahren.

Eine wichtige Erkenntnis ist, dass die Zeitsynchronisation bei Windows sehr schlecht implementiert ist. Die Systemuhr weicht teils mehrere Sekunden von der Referenzzeit ab, obwohl mit NTP eine hohe Genauigkeit erreicht werden könnte. Windows überlässt es sich auch selber, wann die Systemuhr wieder synchronisiert wird, obwohl die Referenzzeit mit NTP in kurzen Abständen bekannt sein sollte.
Dies lässt sich damit erklären, dass für den Normalanwender die Uhr genügend genau ist, auch wenn die Systemzeit eine halbe Minute daneben liegt, hat dies keinen Einfluss auf den Büroalltag.

Eine zusätzliche Erkenntnis hat sich beim Aufzeichnen der Pakete mit Wireshark ergeben. Wireshark greift auf die PCAP-Library zu und diese holt die Zeit nur ein einziges Mal beim Start der Aufzeichnung. Danach arbeitet PCAP mit einer eigenen Uhr, welche für genaue Messungen nicht ausreicht.

16 LAB 6 – ANALYSE NETFLOW-EXPORT FÜR FAST / NORMAL / LONG

16.1 Aufgabenstellung

16.1.1 Ziel

Es ist zu ermitteln welche unterschiede es bei der Auswertung der NetFlows gibt, wenn die folgenden 3 Typen des NetFlow-Exportes angewendet werden:

- Normal
- Fast
- Long

Verschiedene Protokolle (ab Layer 4) sind zu erfassen und deren Auswirkungen mit den 3 Typen zu beschreiben.

16.1.2 Bedingungen

Ein Traffic-Mix wird als Standard angenommen und dieser während der ganzen Analyse verwendet.

16.1.3 Risiken / Challenges

Messungenauigkeiten können Fehlmessungen oder Fehlschlussfolgerungen verursachen.

16.2 Konfiguration

16.2.1 Aufbau

Keinen

16.2.2 Router-Konfiguration

Normal

Die inaktiven Flows werden nach einer bestimmten Zeit (in Sekunden) exportiert (wenn keine Daten mehr empfangen werden und der Flow somit als abgeschlossen gilt).

```
mls aging normal <32-4092>
```

Fast

Beim Fast-Export kann angegeben werden, wie viele Pakete (threshold) in einer gewissen Zeit (time) vorkommen müssen, damit der Flow in der Flow-Table bestehen bleibt und nicht exportiert (resp. geschlossen) wird.

```
mls aging fast time <1-128> threshold <1-128>
```

Long

Diese Zeit gibt an, wann der NetFlow spätestens exportiert und in der NetFlow-Table gelöscht wird. Der NetFlow wird auch unterbrochen, wenn er in der Realität noch nicht beendet ist.

```
mls aging long <64-1920>
```

16.2.3 Scripts

keine

16.3 Testresultate

16.3.1 Erwartet

Es werden Unterschiede zwischen den verschiedenen Modi erwartet, welche per Definition sein müssen. Eventuell wird eine Unregelmässigkeit entdeckt.

16.3.2 Gemessen

Diese Messung kann leider nicht durchgeführt werden, da die 3 Modi nur für einen Router, auf welchem MLS läuft, verfügbar sind.

16.4 Fazit

Unten aufgeführt ist das Original Statement von Cisco[55], was es für ein MLS-Netzwerk braucht, welches wir für unsere Arbeit nicht zur Verfügung haben.

An IP MLS network topology consists of these components:
Multilayer Switching-Switching Engine (MLS-SE)—Catalyst 5000 family switch with Supervisor Engine III or III F with the NetFlow Feature Card (NFFC) or NFFC II, or Supervisor Engine II G or III G, or a Catalyst 2926G series switch. The MLS-SE provides Layer 3 LAN-switching services.
Multilayer Switching-Route Processor (MLS-RP)—A Catalyst 5000 family Route Switch Module (RSM) or Route Switch Feature Card (RSFC), or an externally connected Cisco 7500, 7200, 4700, 4500, or 3600 series router with software that supports IP MLS. The MLS-RP provides Cisco IOS-based multiprotocol routing and network services.

17 LAB 7 – GENAUIGKEIT DER TIMESTAMPS MIT TURBOCAP UND TRAFFIC MIX

17.1 Aufgabenstellung

17.1.1 Ziel

Es soll mit einer speziellen Karte gemessen werden, ob die Timestamps richtig gesetzt werden oder Unregelmässigkeiten zu beobachten sind. Die TurboCAP Karte besitzt zwei 1 GBit-Interfaces und wird von der Firma CACE Technologies[56] vertrieben.
Es gibt für die Timestamps 3 Betriebsmodi, der exakteste davon ist der Polling-Mode. Dieser benötigt ein CPU-Core für sich, da die Software in einem Busy-Wait laufen muss. Dafür garantiert die Firma eine Genauigkeit zwischen 3-5 Mikrosekunden.

17.1.2 Bedingungen

Wireshark muss die aufgezeichneten Pakete mit den Timestamps von TurboCAP versehen.

Damit ein möglichst repräsentativer Datenstrom analysiert werden kann, wird der zuvor erstellte Traffic-Mix verwendet und mit D-ITG erzeugt und empfangen.

17.1.3 Risiken / Challenges

Die TurboCAP Karte kann aus irgendwelchen Gründen nicht zum Messen gebraucht werden. Sei es, weil die Computer alle mit einem Schloss versehen sind und somit weder den Standort wechseln, noch Karten herausgenommen, resp. eingesetzt werden können oder ob die speziellen Treiber sich nicht installieren lassen, resp. nicht kompatibel sind oder ob die Hardware sich nicht für diese TurboCAP Karte eignet.

Mit den vorhandenen Ressourcen (Router, 4 Computer, TurboCAP Karte) kann kein vernünftiger Messaufbau erstellt werden.

17.2 Konfiguration

17.2.1 Aufbau

Auf dem Router läuft ein NTP-Server, der seine Zeit mit einem Referenz-System abgleicht.

Der MessPC ist direkt am Interface Ethernet 1/0 am Router angeschlossen. Auf diesem läuft der NetFlow-Kollektor in einer virtuellen Umgebung.

Der Aufbau mit dem TurboCAP PC im Passthrough-Mode, also zwischen Router und einem PC, hat wegen einer Inkompatibilität nicht funktioniert. Für die TurboCAP Karte ist es zwingend notwendig im Duplex-Modus zu operieren, was nicht erreicht wird.

Die TurboCAP Karte arbeitet auf dem Layer 2 und kann daher keine eigene IP erhalten. Somit kann der Router nicht mit dem TurboCAP PC kommunizieren, wenn dieser direkt am Interface des Routers angeschlossen ist.

Dieses Problem kann mit einem Switch behoben werden. Der Switch arbeitet auf dem Layer 2 und kann somit mit der Turbo-CAP anhand der Mac-Adresse kommunizieren.

Damit der Verkehr zwischen dem PC-Client und dem PC-Server abgehört werden kann, muss auf dem Switch ein Port Monitoring eingerichtet werden. Das Port Monitoring veranlasst den Switch alle Pakete, welche für den PC-Server bestimmt sind, auf dem Monitoring Port, auf welchem die TurboCAP Karte angeschlossen ist, zu duplizieren.

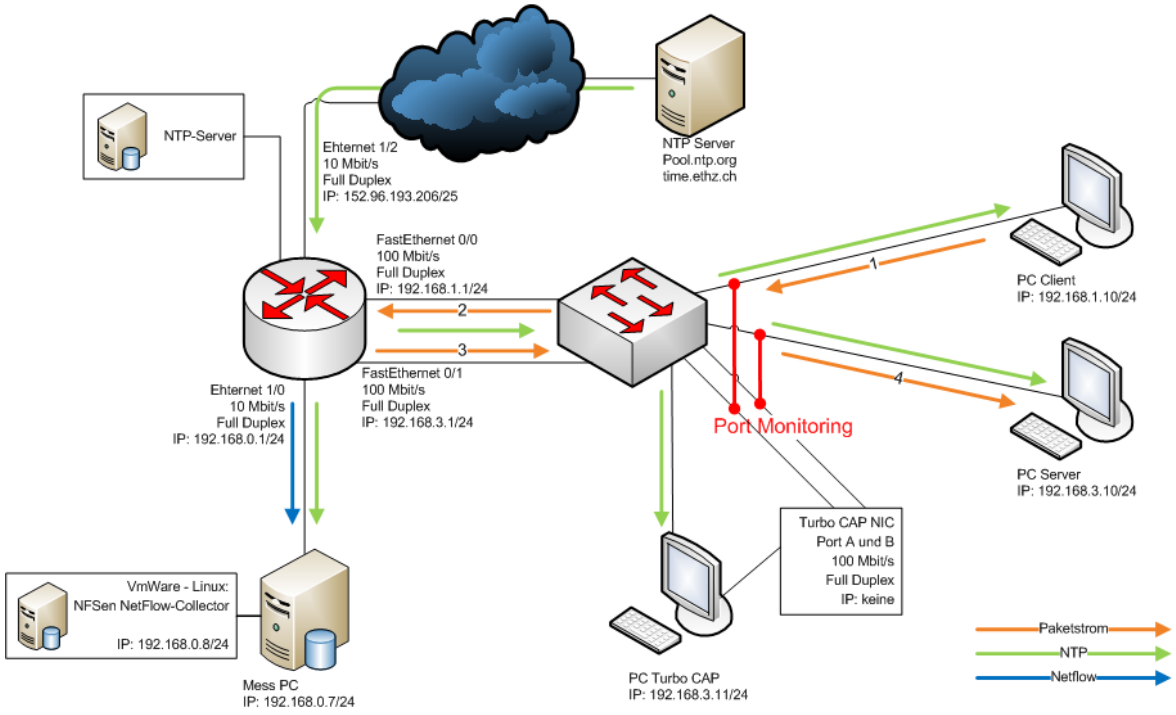


Abbildung 17-1 Aufbau

17.2.2 Router-Konfiguration

Für die Benutzung des Switch im Lab 7 muss das Netzwerk um diesen erweitert und auch entsprechend konfiguriert werden. An den Interfaces wird eine andere IP vergeben.

```
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip route-cache flow
 speed auto
 full-duplex
 no cdp enable
!
interface FastEthernet0/1
 ip address 192.168.3.1 255.255.255.0
 ip route-cache flow
 speed auto
 full-duplex
 no cdp enable
!
interface Ethernet1/0
 ip address 192.168.0.1 255.255.255.0
 full-duplex
 no cdp enable
!
interface Ethernet1/1
 ip address 192.168.202.1 255.255.255.128
 full-duplex
 no cdp enable
!
interface Ethernet1/2
 ip address 152.96.193.206 255.255.252.0
 full-duplex
 no cdp enable
```

Die NetFlow-Exporteinstllungen bleiben wie zuvor bestehen.

```
ip flow-export source Ethernet1/0
ip flow-export version 5
ip flow-export destination 192.168.0.8 9999
```

Dem Router wird für die Zeitsynchronisation ein NTP-Server angegeben. Der Router selbst stellt seine Zeit den Clients im Lab zur Verfügung.

```
ntp clock-period 17180328
ntp source Ethernet1/2
ntp server 192.33.96.102
ntp server 129.132.97.15
```

17.2.3 Switch-Konfiguration

Der Switch besitzt auf der Rückseite einen Konsolen-Port, Stromanschluss und einen Anschluss für eine redundante Stromversorgung. Auf der Frontseite stehen 24 FastEthernet-Port sowie zwei Gigabit-Ports zur Verfügung.

Abbildung 17-2 Switch 3500

Der Switch hat die Modelbezeichnung WS-C3524-XL und besitzt die in der Tabelle 17-1 notierten Eigenschaften.

Typ	WS - CISCO 3524 XL
SW-Version	C3500XL-C3H2S-M
File	c3500xl-c3h2s-mz.120-5.WC10.bin
Compiled Date	28.05.2004, 09:19
IOS-Version	12.0 (5)WC10
Processor	PowerPC403
Memory	9 MBytes
Interfaces	2 Gigabit Ethernet Interfaces
	24 FastEthernet Interfaces

Tabelle 17-1 Switch

Grundkonfiguration

Da der Switch den Auslieferungszustand enthält, muss die Grundkonfiguration selber eingestellt werden. Zusätzlich werden noch Spanning-Tree und das Cisco Discovery Protocol abgeschalten. Der Switch erhält eine IP-Adresse, damit auch über Telnet die Konfiguration angepasst werden kann (und nicht nur über die Konsole).

```
hostname Switch
no spanning-tree
no cdp run
!
username stud privilege 15 password stud
!
ip default-gateway 192.168.2.1
!
interface VLAN 1
   ip address 192.168.3.2 255.255.255.0
!
line console 0
   login local
   logging synchronous
!
vty 0 4
```

```
    login local
    logging synchronous
```

Port-Monitoring
Damit die Pakete auch auf den beiden Ports des Turbo-CAP Computers landen, muss ein Port Mirroring konfiguriert werden.

```
interface FastEthernet 0/23
    port monitoring FastEthernet 0/13
!
interface FastEthernet 0/24
    port monitoring FastEthernet 0/14
!
```

17.2.4 Windows Konfiguration

SMB-Port deaktivieren (unbind Port 445)
Damit der SMB-Port für D-ITG benutzbar wird, darf Windows nicht mehr auf dem Port hören (Listening). Um diesen zu deaktivieren (unbind[57]) kann wie folgt vorgegangen werden:

Im `Device Manager` kann unter `View` das Häckchen bei `Show hidden devices` gesetzt werden. Unter `Non-Plug and Play Drivers` erscheint der Eintrag `NetBios over Tcpip`. Dieses Gerät kann deaktiviert (disable) werden. Nach einem Neustart ist der Port nicht mehr gebunden. Die Überprüfung erfolgt durch `nbtstat –a`.

17.2.5 Scripts

PC Client

Auf dem Client wird gemäss des definierten Trafficmix mit Hilfe von D-ITG eine Batchdatei erstellt, welche die jeweiligen Charakteristiken des Netwerkverkehrs an den PC Server sendet. Der Netzwerkverkehr kann anhand des Layer 4 Protokoll und dessen Port identifiziert werden.
Die Batchdatei wird als scheduled Task erfasst und von dort alle 10 Minuten ausgeführt. Falls der Sendprozess von D-ITG "ITGSend.exe" sich aufhängt wird er zu Beginn mittels taskkill entfernt. Für die Zeitsynchronisation zwischen PC Client und PC Server wird NTP eingesetzt.

lab7.cmd

```
taskkill /IM itgsend.exe /T /F
choice /d y /n /t 30
lab7_business.cmd
```

lab7_business.cmd

```
taskkill /IM ITGSend.exe /T /F

set file="lab7_business_01.txt"

echo Messstart Lab 7 >> %file%
echo ======================================== >> %file%
echo %date% >> %file%
echo %time% >> %file%

echo // 35% SMB (Fileserver) 1)TCP:445
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -T TCP -rp 445 -t 20920

echo // 20% Webtraffic 1)UDP/DNS:53 2)TCP/HTTP:80
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 380
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 6000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 2000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 600
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 1000
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 1500
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 500

echo // 10% Webmeeting 1)HTTP:80, TCP:2000, TCP/HTTPS:443
itgsend -a 192.168.3.10 -rp 80 -T TCP -t 20
itgsend -a 192.168.3.10 -rp 443 -T TCP -t 20
itgsend -a 192.168.3.10 -rp 2000 -T TCP -t 5960

echo // 10% Applikationsserver (SAP DB)
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -rp 3306 -T TCP -t 5980

echo // 10% Mail(Exchange) 1)LDAP TCP:389, 2)IMAP TCP:143 3)POP3
TCP:110 4)NNTP TCP:119 5)SMTP TCP:25
itgsend -a 192.168.3.10 -rp 389 -T TCP -t 20
itgsend -a 192.168.3.10 -rp 143 -T TCP -t 4000
```

```
itgsend -a 192.168.3.10 -rp 119 -T TCP -t 980
itgsend -a 192.168.3.10 -rp 25 -T TCP -t 1000

echo // 10% Telefon(VoIP)
itgsend -a 192.168.3.10 -rp 53 -T UDP -t 20 DNS
itgsend -a 192.168.3.10 -rp 5000 -t 5980 VoIP -x G.711.2 -h RTP -
VAD

echo //5% Services
(NTP/WindowsUpdate/Antivirus/Telnet/RemoteDesktop)
itgsend -a 192.168.3.10 -rp 123 -t 100
itgsend -a 192.168.3.10 -rp 23 -t 400
itgsend -a 192.168.3.10 -rp 23 -t 500
itgsend -a 192.168.3.10 -rp 23 -t 1000 Telnet
itgsend -a 192.168.3.10 -rp 3389 -t 2000

echo %date% >> %file%
echo %time% >> %file%
echo ======================================== >> %file%
echo Messende Lab 7 >> %file%
```

PC Server

Der Server nimmt die gesendeten Daten des Clients entgegen. Ein scheduled Task ist eingerichtet, welcher alle 10 Minuten die Batchdatei neu ausführt und dafür sorgt, dass ein allfällig hängengebliebener Prozess zuerst beendet und danach wieder eröffnet wird.

```
taskkill /IM itgrecv.exe /T /F
itgrecv.exe
```

PC TurboCap

Auf dem PC mit der TurboCap-Karte für die Paketmessung wird Wireshark eingesetzt. Um die Daten an den beiden Interfaces der TurboCap-Karte zu empfangen, wird ein Capturefile von 50 MB erstellt. Sobald dieses voll ist, wird ein neues erstellt, dies dient als Schutz gegen ein OutOfMemory-Fehler. Diese Lösung ist über die Ringbufferfunktion realisiert.

```
start wireshark.exe -i 5 -s 64 -b filesize:51200 -w Lab7_A.pcap -k
start wireshark.exe -i 6 -s 64 -b filesize:51200 -w Lab7_B.pcap -k
```

17.3 Testresultate

17.3.1 Erwartet

Es wird erwartet, dass die Timestamps korrekt eingetragen werden.

17.3.2 Gemessen

Wie erwartet gibt es keine Steigung des Graphen mehr. Die Timestamps sind demnach keinen Unregelmässigkeiten der Computer-Uhr mehr unterworfen.

Trotzdem erstaunt es, dass eine Zeitdifferenz von beinahe einer ganzen Sekunde resultiert, wie in der Abbildung 17-3 ersichtlich ist.

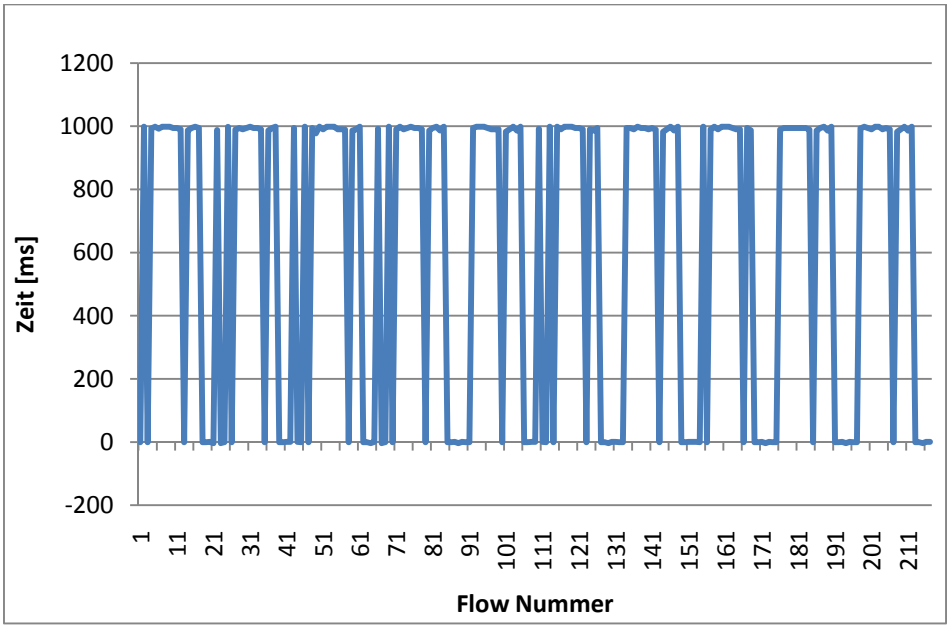


Abbildung 17-3 Zeitdifferenz NetFlow und Effektiv

Ein Durchlauf besteht aus ca. 23 Flows, je nachdem welche Anzahl der Pakete richtig übertragen wurden. Es ist eine Regelmässigkeit in der Achse der Flow Nummern zu erkennen.

Nach einer Aufschlüsselung der Zeitdifferenzen auf den Layer 4 Protokollen wird ersichtlich, dass nur die TCP-Verbindungen eine solche Zeitdifferenz von fast 1 Sekunde haben.
Es kann also gesagt werden, dass alle TCP-Verbindungen von NetFlow als zu lang angezeigt werden.

Ein umgekehrter Fall, wenn also die reale Verbindung länger dauern würde, könnte damit erklärt werden, dass die Vermittlungszeit im Router, die Weiterleitungszeit im Switch und die Signallaufzeit noch dazugerechnet werden müssen. Diese Zeitdifferenz würde aber nicht in dieser Dimension liegen.

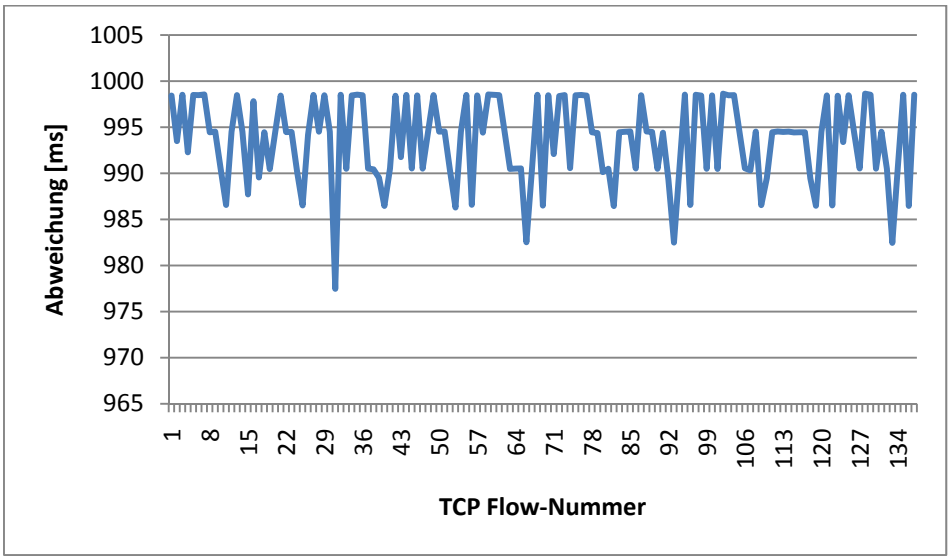


Abbildung 17-4 Zeitdifferenz TCP-Verbindungen

In der Tabelle Tabelle 17-2 ist ersichtlich wie gross die durchschnittliche Zeitdifferenz ist, welche der Router mit NetFlow zu lang protokolliert und somit falsche Angaben in der Auswertung ergeben.

Angabe	Wert in Milisekungen [ms]
Minimum	998.635
Maximum	977.462
Mittelwert	993.5816
Varianz des Mittelwertes	20.11736

Tabelle 17-2: Zeitdifferenz bei TCP-Flows

Für UDP ergibt sich ein anderes Bild. Bei einer UDP-Verbindung mit einem einzelnen Paket ist zu beobachten, dass die realen Verbindungen länger dauern als in den NetFlows angegeben. Dies überrascht nicht, da der Router dieselbe Start- und Endzeit einfügt, was eine Verbindungsdauer von 0 (Sekunden) ergibt. In der Realität beinhaltet eine solche Verbindung zusätzlich die Zeit, welche das Paket von einem zum anderen Computer benötigt.

Wenn die NetFlows teilweise eine kürzere Zeit aufzeigen, handelt es sich immer um einen Wert, welcher kürzer als 1 Millisekunde ist, also um eine ganze Dimension (von Milli- auf Mikrosekunden) kleiner.

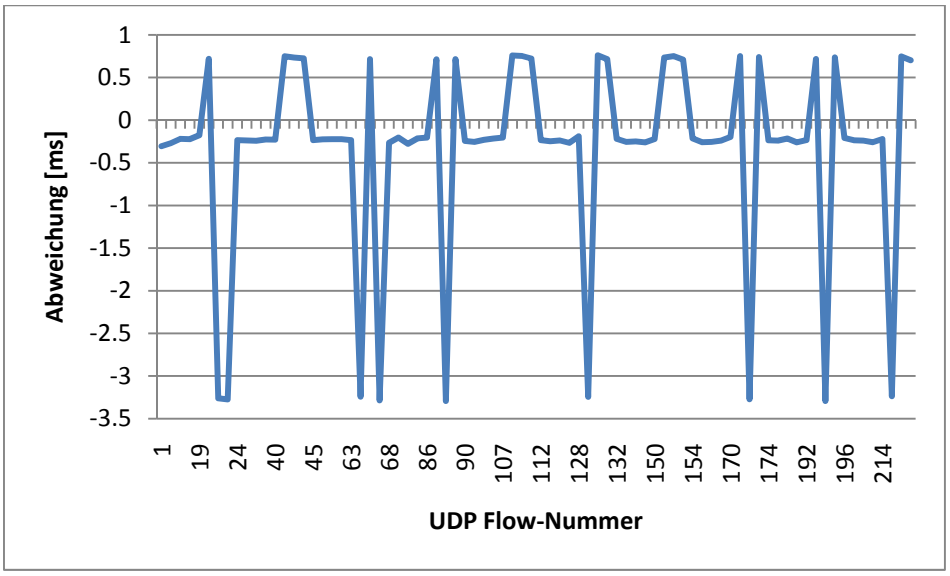


Abbildung 17-5 Zeitdifferenz UDP-Verbindungen

17.4 Fazit

Die grosse Erkenntnis bei diesem Versuch ist bei den TCP-Verbindungen zu finden. NetFlow ermittelt im Mittelwert um 0.993 Sekunden einen zu grossen Wert, als die reale Verbindung in Wirklichkeit ist.
Dies kann bei vielen kurzen Verbindungen zu einem massiven Messfehler führen.

Bei den UDP-Verbindungen ergibt sich die gleiche Erkenntnis, aber in einer viel kleineren Dimension, welche sogar vernachlässigt werden kann.
Wenn eine UDP-Verbindung nur aus einem Paket besteht, ist natürlich die reale Dauer länger, als von NetFlow reportet, denn die Signallaufzeit und die Verzögerungen in den Netzwerkgeräten müssen noch dazuaddiert werden.

18 PROJEKTMANAGEMENT

Dieser Abschnitt beschreibt die Tätigkeiten des Projektmanagement zur Diplom- / Bachelorarbeit mit dem Titel Router Forensics: Challenging NetFlow Accuracy.

18.1 Projektangaben

18.1.1 Projektdauer

Die Diplom- / Bachelorarbeit wird im Herbstsemester 2009 an der HSR durchgeführt. Dabei ist die Projektdauer von 8 Wochen vorgegeben, welche vom 28.09.2009 bis zum 20.11.2009 dauert.

18.1.2 Projektbeteiligte

Das Team besteht aus den zwei Informatikstudenten Marcel Jakopic und Christian Jung. Die Diplom- / Bachelorarbeit wird als 2er Team ausgelegt.
Betreut wird die Arbeit von unserem Dozenten Herr Eduard Glatz.
Der Auftraggeber ist Herr Bernhard Tellenbach von der ETH Zürich.

18.1.3 Beschreibung des Projektes

Die Labbeschreibungen, sowie auch die Resultate und Erkenntnisse werden im Bericht festgehalten. Diese Projektdokumentation wird in gebundener Form an den Betreuer und der Abteilung Informatik überreicht. Der Bericht, sowie alle Messdaten und Auswertungen werden in digitaler Form auf einer DVD-ROM abgegeben.

18.2 Meilensteine

Die folgenden Meilensteine beschreiben den Fortschritt bezüglich der gesamten Projektarbeit. Sie enthalten die wichtigsten Arbeiten. Die Termine der Meilensteine sind im Projektplan eingezeichnet und werden nicht verschoben. Allfällige Abweichungen sind dokumentiert.

18.2.1 Soll

MS1 - Initialdokumente erstellt / Evaluation beendet

Zu erfüllende Punkte:

- Projektplanung erstellt
- Dokumentationsstruktur erstellt
- Evaluation von Software für Trafficgeneratoren, Netflowanalysetools abgeschlossen und dokumentiert.
- Einfache Messung von NetFlows erstellt und dokumentiert
- Lab1 abgeschlossen

MS2 - Lab2 fertig

Zu erfüllende Punkte:

- Definition aller Labs, überprüft von Herrn Glatz und Herrn Tellenbach
- Labaufbau erstellt und dokumentiert
- Bedingungen und erwartete Resultate für Lab2 festgelegt und dokumentiert
- Lab2 abgeschlossen

MS3 - Lab3 fertig

Zu erfüllende Punkte:

- Labaufbau erstellt und dokumentiert
- Bedingungen und erwartete Resultate für Lab3 festgelegt und dokumentiert
- Lab3 abgeschlossen

MS4 - Lab4 fertig

Zu erfüllende Punkte:

- Labaufbau erstellt und dokumentiert
- Bedingungen und erwartete Resultate für Lab4 festgelegt und dokumentiert
- Lab4 abgeschlossen

MS5 - Lab5 fertig

Zu erfüllende Punkte:

- Labaufbau erstellt und dokumentiert
- Bedingungen und erwartete Resultate für Lab5 festgelegt und dokumentiert
- Lab5 abgeschlossen

MS6 - Lab6 fertig / Abstract abgegeben

Zu erfüllende Punkte:

- Kurzbeschreibung der DA/BA abgegeben
- Labaufbau erstellt und dokumentiert
- Bedingungen und erwartete Resultate für Lab6 festgelegt und dokumentiert
- Lab6 abgeschlossen

MS7 - Schlussabgabe

Zu erfüllende Punkte:

- Dokumentation ausgedruckt und abgegeben
- Plakat aufgehängt
- 3 CDs mit Inhalt fertiggestellt

18.2.2 Ist

MS1 - Initialdokumente erstellt / Evaluation beendet

Abweichungen:

- Die Evaluation kann noch nicht abgeschlossen werden, da es sehr viele Programme zu testen und auszuwerten gibt.
- Die Dokumentationsstruktur ist teilweise erstellt.
- Es konnte noch keine Software gefunden werden, welche es erlaubt, die Inhalte der NetFlow-Pakete zu analysieren.

MS2 - Lab2 fertig

Abweichungen:

- Definitionen aller Labs noch nicht vollständig.
- Lab2 kann noch nicht abgeschlossen werden.

MS3 - Lab3 fertig

Abweichungen:

- Lab3 kann noch nicht abgeschlossen werden, da bei den Messungen Fehler aufgetreten sind und nochmals durchgeführt werden mussten.

MS4 - Lab4 fertig

Abweichungen:

- Lab4 ist sehr aufwendig zum Auswerten und daher noch nicht abgeschlossen worden.

MS5 - Lab5 fertig

Abweichungen:

- Mit den Messungen gibt es grosse Unregelmässigkeiten, welche analysiert werden. Dies ergibt einen grösseren Zeitaufwand und dadurch eine Verzögerung des Abschlusses für das Lab5.

MS6 - Lab6 fertig / Abstract abgegeben

Abweichungen:

- Mit dem Abstract der DA/BA muss noch das Management Summary und das Poster dem Betreuer zur Überprüfung geschickt werden.
- Lab6 ist vorzeitig abgeschlossen worden, da der Router MLS nicht unterstützt. Es wird ein neues Lab erstellt, bei welchem nochmals die Timestamps mit einer speziell exakten Karte gemessen werden können.

MS7 - Schlussabgabe

Abweichungen:

- keine

18.3 Risikoanalyse

18.3.1 Ausblick

In der folgenden Tabelle sind mögliche Risiken zum Projekt aufgelistet. Daraus ist ersichtlich welche Rückstellungen und Reserven zum Beheben oder Minimieren der Risiken in der Planung notwendig sind.

Risiko	Kommentar	Auswirkung	p	Std	p * Std
Ressourcen					
Krankheit Marcel	ca. 1-2 Tage pro Jahr	grosse Verzögerung	0.01	40	0.40
Krankheit Christian		grosse Verzögerung	0.01	40	0.40
Erreichbarkeit Betreuer / Auftraggeber	es wird Zeit verbraucht, um zu kontaktieren	kleinere Verzögerung	0.40	20	8.00
Hilfe von anderen Instituten (z.B. INS) / Personen nötig	auf Hilfe / Tips von anderen Sites angewiesen	mittlere Verzögerung	0.50	20	10.00
Finanziell					
Software oder Hardware wird benötigt, welche etwas kosten	wer soll dies bezahlen	mittel, evtl. kann ein LAB nicht durchgeführt werden	0.30	40	12.00
Technisch					
Router defekt	Hauptarbeitsgerät, Neubeschaffung, Reparatur	hoch	0.01	80	0.80
Datenverlust (von 1 Tag)	Daten auf SVN, Notebook Christian, Notebook Marcel	Dokumente überarbeiten, anpassen	0.10	8	0.80
Datenverlust (komplett)	Daten auf SVN, Notebook Christian, Notebook Marcel	alle Dokumente neu erstellen	0.001	80	0.08
Software-Evaluation aufwändiger	keine geeignete SW gefunden	mittel, evtl. kann ein LAB nicht durchgeführt werden	0.50	40	20.00
Messung eines Labs ist aufwändiger	benötigt mehr Zeit	Kleinere Verzögerung	0.40	20	8.00

Killerkriterien					
Pandemie	Ausbruch der Schweinegrippe, Schule muss geschlossen werden	grosse Verzögerung oder Abbruch	0.01	640	6.40
Total					**66.88**

Tabelle 18-1 Risikoanalyse

18.3.2 Auswertung

In der Tabelle 18-2 ist ersichtlich welche Risiken eingetroffen sind.

Risiko	Eingetreten	Benötigte Zeit
Krankheit Marcel	Nein	0
Krankheit Christian	Ja	8
Erreichbarkeit Betreuer / Auftraggeber	Ja	4
Hilfe von anderen Instituten (z.B. INS) / Personen nötig	Ja	12
Software oder Hardware wird benötigt, welche etwas kosten	Nein	0
Router defekt	Nein	0
Datenverlust (von 1 Tag)	Nein	0
Datenverlust (komplett)	Nein	0
Software-Evaluation aufwändiger	Ja	32
Messung eines Labs ist aufwändiger	Teilweise	16
Pandemie	Nein	0
Total		**72**

Tabelle 18-2 Auswertung Risikoanalyse

Ein Teammitglied war einen Tag krankheitshalber abwesend, die Arbeit ist jedoch am Abend oder am Wochenende erledigt worden.

Die Erreichbarkeit des Betreuers und des Auftraggebers war akzeptabel. Wenn mal ein Mail längere Zeit nicht beantwortet wurde, konnten andere Aufgaben während dieser Zeit erledigt werden.

Auf die Hilfe von Drittpersonen ist regelmässig zurückgegriffen worden. Es gab kleinere Verzögerungen, weil die Computer angekettet und verschlossen sind.
Die verschiedenen Labs bedingten immer wieder weitere Hardware, wie z.B. einen zusätzlichen Switch oder zusätzliche Ports für den Router.

Die Software-Evaluation war viel aufwendiger als geplant. Die Herausforderung war, die am besten geeigneten zu finden.

Einzelne Labs waren zum Auswerten sehr aufwendig. Das Lab 6 war sehr kurz, da die zu testende Funktion für diesen Router nicht vorhanden ist. Mit der vorhandenen Zeit konnte ein zusätzliches Lab 7 bearbeitet werden.

18.4 Projektüberwachung

18.4.1 Zeitmanagement

Jedes Teammitglied führt ein persönliches Zeiterfassungsdokument. In diesem werden alle erledigten Arbeiten mit dem dafür aufgewendetem Zeitaufwand und der Zuordnung der entsprechenden Kategorie festgehalten.

Es wird von einem wöchentlichen Zeitaufwand von ca. 42 Stunden ausgegangen. Dieser kann je nach Arbeit von Woche zu Woche unterschiedlich ausfallen.

Die aufgelisteten Diagramme über die Verteilung der Arbeitszeit soll aufzeigen, welche Tätigkeiten mehr oder weniger Ressourcen als geplant in Anspruch genommen haben.

18.4.2 Zeitaufwand nach Kategorien und Team-Mitglieder

In der Abbildung 18-1 sind die verschiedenen Kategorien aufgeschlüsselt. Es ist somit ersichtlich, welches Teammitglied, wieviel Zeit für die jeweilige Kategorie benötigt hat.

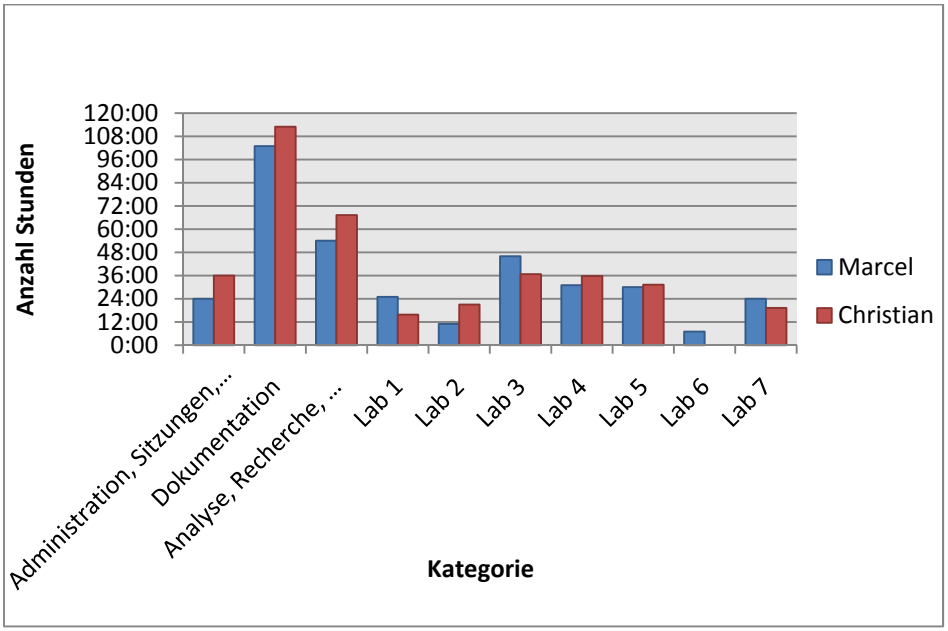


Abbildung 18-1 Zeitaufwand nach Kategorien und Team-Mitglieder

18.4.3 Prozentualer Anteil der Kategorien

In der Abbildung 18-2 ist ersichtlich, welche Kategorie welchen Gesamtanteil benötigt hat.

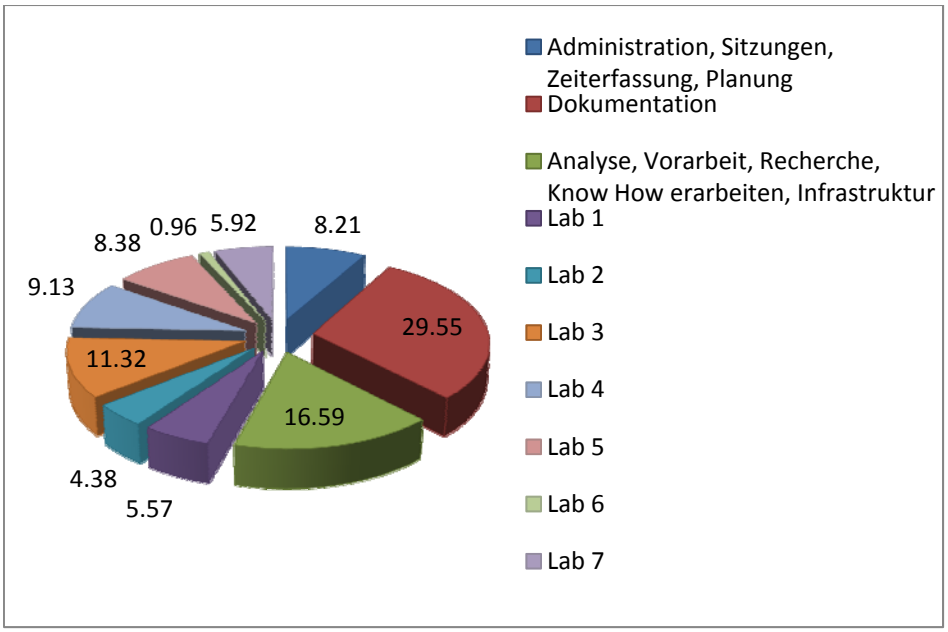


Abbildung 18-2 Prozentualer Anteil der Kategorien des Teams

18.4.4 Wochendiagramm des Teams

In der Abbildung 18-3 ist ersichtlich, wie viel in den einzelnen Wochen gearbeitet wurde.

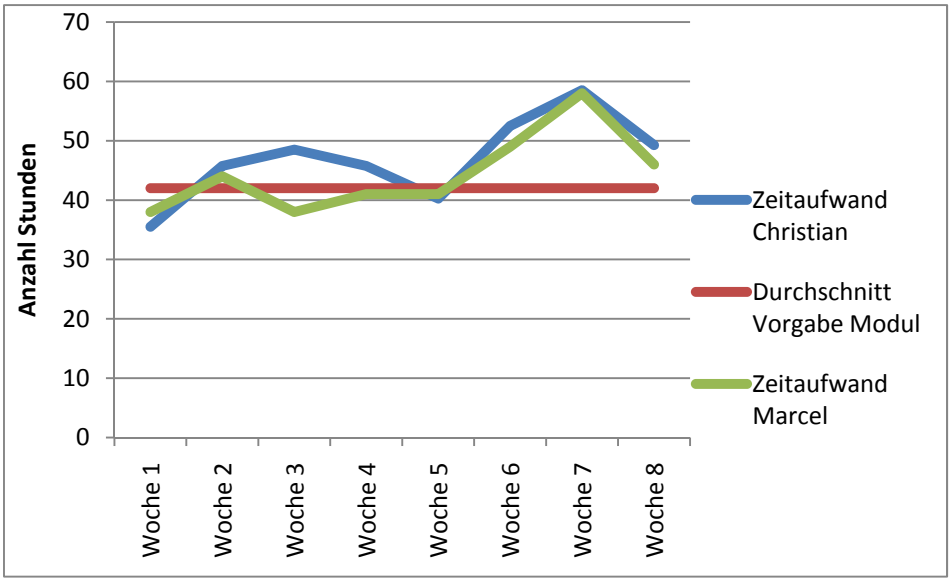


Abbildung 18-3 Zeitaufwand nach Wochen

18.5 Sitzungsübersicht

Es werden insgesamt 8 Sitzungen abgehalten, welche den aktuellen Projektstand aufzeigen. Die detaillierten Sitzungsprotokolle sind im Anhang zu finden.

Woche	Datum	Teilnehmer	Ort	Bemerkungen
1	28.09.2009	Betreuer, Auftraggeber und Studenten	ETH Zürich	Kickoff Meeting
2	02.10.2009	Betreuer und Studenten	Rapperswil	
3	09.10.2009	Betreuer und Studenten	Rapperswil	MS 1
4	16.10.2009	Betreuer und Studenten	Rapperswil	MS 2
5	23.10.2009	Betreuer und Studenten	Rapperswil	MS 3
6	30.10.2009	Betreuer und Studenten	Rapperswil	MS 4
7	06.11.1998	Betreuer und Studenten	Rapperswil	MS 5
8	13.11.2009	Betreuer und Studenten	Rapperswil	MS 6

Tabelle 18-3 Sitzungen

18.6 Projektplan

Der Projektplan wird am Anfang der Arbeit erstellt. Dieser dient als Indikator, wie weit die einzelnen Arbeiten vorangeschritten sein sollten. Durch den Projektplan kann, falls nötig, schnell festgestellt werden, welche Aufgaben eine höhere Priorität erhalten sollen.

18.6.1 SOLL Projektplan

Projektplanung

Arbeiten	KW 40							KW 41							KW 42							KW 43							KW 44							KW 45							KW 46							KW 47							
	28.09.2009	29.09.2009	30.09.2009	01.10.2009	02.10.2009	03.10.2009	04.10.2009	05.10.2009	06.10.2009	07.10.2009	08.10.2009	09.10.2009	10.10.2009	11.10.2009	12.10.2009	13.10.2009	14.10.2009	15.10.2009	16.10.2009	17.10.2009	18.10.2009	19.10.2009	20.10.2009	21.10.2009	22.10.2009	23.10.2009	24.10.2009	25.10.2009	26.10.2009	27.10.2009	28.10.2009	29.10.2009	30.10.2009	31.10.2009	01.11.2009	02.11.2009	03.11.2009	04.11.2009	05.11.2009	06.11.2009	07.11.2009	08.11.2009	09.11.2009	10.11.2009	11.11.2009	12.11.2009	13.11.2009	14.11.2009	15.11.2009	16.11.2009	17.11.2009	18.11.2009	19.11.2009	20.11.2009	21.11.2009	22.11.2009
Projekteinarbeitung	4	1	1																																																					
Projektplanung / Zeitplanung		3	2	2																																																		
Einarbeitung Netflow	2	4	4	4																																																		
Einarbeitung Traffic Generators	2	4	4	4																																																	
Alternative Tools				2	8			10	10	10	10	6																																										
Lab1: Auslastung der CPU mit und ohne NetFlow																																																					
Infrastruktur vorbereiten		1	1																																																	
Inbetriebnahme Router		1																																																		
Messversuche			1	2				4	4	2																																											
Messdatenanalyse und Auswertung				2	4					4	4	3																																									
MS1: Einarbeitung, Lab1 abgeschlossen												MS																																								
Lab2: Auslastung der IF mit viel Bandbreite																																																				
Infrastruktur vorbereiten															6	5																																					
Scripting															6	6	6																																					
Messversuche																2	8	8																																			
Auswertung																		6	8																																	
Reserve																		2	2																																	
MS2: Lab2 Bandbreite abgeschlossen																			MS																															
Lab3: CPU Auslastung (z.B. Full Routing Table)																																																			
Infrastruktur vorbereiten																					6	5																													
Scripting																					6	6	6																												
Messversuche																						2	8	8																											
Auswertung																								6	8																										
Reserve																								2	2																										
MS3: Lab3 Routing abgeschlossen																									MS																									
Lab4: Memoryauslastung																																																			
Infrastruktur vorbereiten																												6	5																							
Scripting																												6	6	6																					
Messversuche																													2	8	8																				
Auswertung																															6	8																		
Reserve																															2	2																		
MS4: Lab4 kein Memory abgeschlossen																																MS																		
Lab5: Analyse Fastexport - Normalexport																																																			
Infrastruktur vorbereiten																																			6	5														
Scripting																																			6	6	6												
Messversuche																																				2	8	8											
Auswertung																																						6	8										
Reserve																																						2	2										
MS5: Lab5 Export abgeschlossen																																							MS										
Lab6: Zeitverhalten																																																		
Infrastruktur vorbereiten																																										6	5								
Scripting																																										6	6	6							
Messversuche																																											2	8	8					
Auswertung																																												6	8					
Reserve																																												2	2				
MS6: Lab6 Zeit, Kurzbeschreibung																																													MS				
Administration	1																																									2		2						
Sitzungen	2				2							2						2							2						2									2					2					
Dokumentation		2	4	2	4			4	4	2	2	6			4	4	2	2	6			4	4	2	2	6			4	4	2	2	6			4	4	2	2	6			4	4	2	2	6			17	17	17	17	8		
MS7: Final Abgabe																																																	MS			
Total Arbeitsstunden	11	16	17	18	18	0	0	18	18	18	16	17	0	0	16	17	16	18	18	0	0	16	17	16	18	18	0	0	16	17	16	18	18	0	0	16	17	16	18	18	0	0	18	17	18	18	18	0	0	17	17	17	17	8	0	0	672
Legende:																																																			
Angaben in Stunden																																																			

Tabelle 18-4 Projektplan SOLL

18.6.2 IST Projektplan

Projektplanung

Arbeiten	KW 40							KW 41							KW 42							KW 43							KW 44							KW 45							KW 46							KW 47							
	28.09.2009	29.09.2009	30.09.2009	01.10.2009	02.10.2009	03.10.2009	04.10.2009	05.10.2009	06.10.2009	07.10.2009	08.10.2009	09.10.2009	10.10.2009	11.10.2009	12.10.2009	13.10.2009	14.10.2009	15.10.2009	16.10.2009	17.10.2009	18.10.2009	19.10.2009	20.10.2009	21.10.2009	22.10.2009	23.10.2009	24.10.2009	25.10.2009	26.10.2009	27.10.2009	28.10.2009	29.10.2009	30.10.2009	31.10.2009	01.11.2009	02.11.2009	03.11.2009	04.11.2009	05.11.2009	06.11.2009	07.11.2009	08.11.2009	09.11.2009	10.11.2009	11.11.2009	12.11.2009	13.11.2009	14.11.2009	15.11.2009	16.11.2009	17.11.2009	18.11.2009	19.11.2009	20.11.2009	21.11.2009	22.11.2009	
Projekteinarbeitung	4.3	5	4.3																																																					
Projektplanung / Zeitplanung	1	3.5		2				1.3				2			0.5																																								
Einarbeitung Netflow	3	6								10	1											2																																		
Einarbeitung Traffic Generators			9.5					5.8		8.5	9.8				3	1.8																																							
Alternative Tools		7	6	7	7			4	8.8			1			2		1.3																								5.8															
Lab1: Auslastung der CPU mit und ohne NetFlow				2				0.8				6	1		10	6.5	1	3.5	3.8			2																																			
Lab2: Auslastung der IF mit viel Bandbreite																1	13	7.8	3.8			6																																		
Lab3: CPU Auslastung (z.B. Full Routing Table)																						7.3	12	18	10	10			15	9	1.8																										
Lab4: Memoryauslastung																									2.3				6	8	4.3	8.8			15	6.8			5.5			6	4											
Lab5: Zeitverhalten																														4.3	3.3					9.8	21	6.5	9.3			6.8	0.5											
Lab6: Analyse Fastexport - Normalexport																															4	3																						
Lab7: Genauigkeit der Timestamps mit Turbo-Cap																																											10	13	5	11	2							
Administration								1				1				1.8	2.5	2	2.5		0.5	0.8	3.3			0.5		0.3	0.5				0.8	0.8			1.5	0.8	0.8	0.5	0.5			1.3		1	1	0.5						11		
Sitzungen		2			1							2				0.5			4.3							2							2							2.3							1.8							
Dokumentation	3.8	2.8	2.3		2			4.3	7.5	3.3	4	1	1	1	1.8	8.3	2		1				1.5		2.5	6.5			1.3	2	4		3					3.5		12	6	3.5	6.5	10	7.5	4	13	24	16	16	20	8		
MS7: Final Abgabe																																																					
Total Arbeitsstunden	14	24	22	11	10	0	0	17	16	22	15	13	2	1	18	20	19	13	15	0	0.5	18	16	18	15	19	0	0.3	17	17	18	12	18	0	0	17	17	22	16	18	0	12	20	18	21	16	21	6	13	24	16	16	20	19	0	0	731
Legende:																																																					
Angaben in Stunden																																																				

Tabelle 18-5 Projektplan IST

18.6.3 Abweichungen

Im Grossen und Ganzen wird der Projektplan eingehalten. Bei der Suche nach einem geeigneten Traffic Generator wird massiv mehr Zeit benötigt. Dafür wird beim Lab 2 Zeit eingespart.
Das Lab 3 ist aufwendiger als geplant. Beim Lab 4 und 5 gibt es nur minimale Abweichungen. Das Lab 6 wird beendet, nachdem sich herausstellt, dass mit dem zur Verfügung stehenden Equipment keine Messung durchgeführt werden kann. Diese Zeit wird im Lab 7 investiert.
Die Administration wird klar unterschätzt und benötigt markant mehr Zeit.
Für die Projektdokumentation wird auch mehr Zeit aufgewendet.

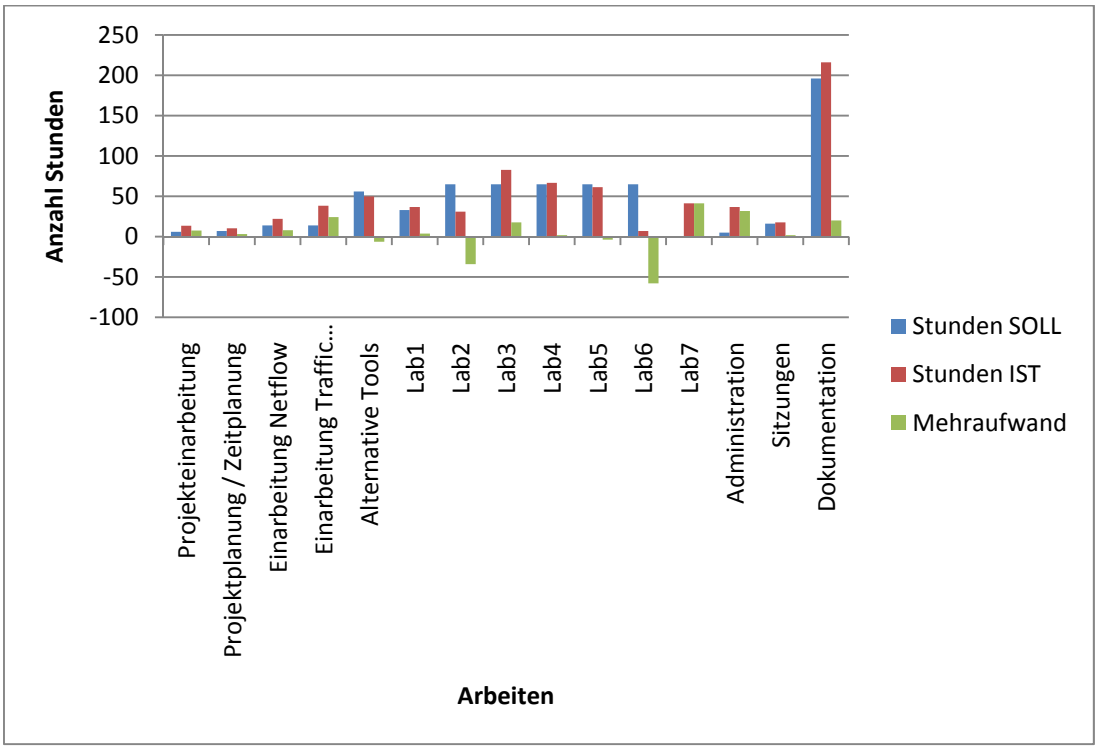


Abbildung 18-4 Abweichungen Soll / Ist Projektplan

19 PERSÖNLICHE BERICHTE

19.1 Marcel Jakopic

19.1.1 Allgemein

Bei der Auswahl der Arbeiten ist mir sofort diese Arbeit aufgefallen, obwohl wir schon vor der offiziellen Bewerbungszeit verschiedene Arbeiten angeschaut und mit den jeweiligen Dozenten besprochen haben. Es entspricht genau meinen Vorstellungen, was ich gerne für den Erhalt meines zukünftigen Bachelor-Titels leisten möchte.

Schon bald nach dem Beginn meines Studiums war mir klar, dass ich nicht der klassische Programmierer sein werde, den die Hochschule ausbildet. Mein grosses Interesse gilt vor allem den Computernetzwerken und alles was damit zu tun hat.

Die Arbeit „Router Forensics“ beinhaltet auch einen grossen Teil an Forschung, was mir zuerst ein bisschen Kopfzerbrechen bereitete. Wie es in der Forschung üblich ist, kann man erst nach mehreren Anläufen ins Ziel gelangen und muss dabei ziemlich viel Zeit investieren. Diese ist bei uns relativ beschränkt und der Abgabetermin steht von Anfang an fest und kann nicht verschoben werden.
Das Argument, dass eben auch ein misslungener Versuch oder Test trotzdem ein Resultat darstellt, stimmte mich zuversichtlich.

Etwas speziell ist die Konstellation, dass diese Arbeit eine Bachelor-Arbeit für mich und eine Diplomarbeit für Christian ist. Diese Tatsache bedingt, dass nur 8 Wochen Vollzeit zur Verfügung stehen, anstatt 16 Wochen Teilzeit wie bei einer regulären Bachelor-Arbeit.
Da ich das Studium berufsbegleitend absolviere und noch einen 80% Arbeitsvertrag habe, stand ich vor dem Problem, was wohl mein Arbeitgeber dazu meint, wenn ich einfach für fast 2 Monaten nicht mehr zur Verfügung stehe.
Mein Vorgesetzter war damit einverstanden, was mich überaus freute und der Arbeit somit nichts mehr im Weg stand.

19.1.2 Erfahrungen

Meine Studienarbeit konnte ich als Einzelarbeit für meinen Arbeitgeber schreiben. Die Bachelor-Arbeit wollte ich daher als Abwechslung in einem Team schreiben.

Ich bin überzeugt mit Christian einen guten und zuverlässigen Teampartner gefunden zu haben. Wir haben uns gut verstanden und konnten uns bei Diskussionen ergänzen. Diese Eigenschaft ist bei einer Forschungsarbeit sehr wichtig, da das zu Testende und der ganze Versuchsaufbau zuerst ausgiebig diskutiert und aufgezeichnet werden müssen.
Während der Diskussion zeigten sich bereits Herausforderungen, welche gemeistert werden mussten. Ohne diese zuerst zu lösen, wäre es überflüssig gewesen einen Testaufbau zu erstellen. Manchmal entdeckten wir auch Probleme, die erst beim Testaufbau aufgetaucht waren, entweder durch eine andere Erwartung oder eine Tatsache die wir nicht bedacht hatten.
Je komplexer der Versuchsaufbau, desto mehr an Diskussionszeit war erforderlich.

Spannend war auch der ganze Verlauf der Messungen. Es war am Anfang nur die Richtung klar, nicht aber die einzelnen Schritte. Durch diese Grobplanung konnten interessante Entdeckungen nochmals gemessen und bestätigt oder andere, welche nach der Spezifikation arbeiteten, zu den Akten gelegt werden.

19.1.3 Verbesserungen

Bei einem Versuchsaufbau ist es sehr wichtig, sich vorher zu überlegen, wie die Messdaten anschliessend vorliegen und wie diese so einfach wie möglich ausgewertet werden können. Manchmal lohnt sich sogar ein anderer Versuchsaufbau oder eine andere Reihenfolge der Teilmessungen.
Dies haben wir nicht immer geschafft und mussten dann viel Zeit für die Auswertung aufwenden.

Ein grosses Problem war die Unzuverlässigkeit gewisser Programme, welche immer wieder und ohne Grund abgestürzt sind. Die langwierigen Messungen, welche wir in der Nacht laufen lassen haben, waren dann am nächsten Tag unvollständig und mussten nochmals ausgeführt werden. Über die ganze Arbeit gesehen und diese Zeit kummuliert, hätte es sich wahrscheinlich doch noch gelohnt, selber ein Programm zu schreiben, welches Sockets öffnen und Daten darüber verschicken kann.

Zu Beginn der Arbeit brauchten wir relativ viel Zeit um uns einzuarbeiten. Es gibt sehr viele Programme, welche NetFlow empfangen oder mit diesen arbeiten können. Einige davon nicht wirklich zuverlässig, was wir zuerst herausfinden mussten.
Unser Fokus waren Programme für Windows. Erst nach einiger Zeit benutzten wir auch Programme für Linux.
Die Linux-Installation, basierend auf der Fedora-Distribution, liess sich auf den vorinstallierten Rechnern nicht starten. Wir haben dann eine vorinstallierte Ubuntu-Distribution für VMWare erhalten, ab diesem Zeitpunkt konnten wir auch Programme für Linux ausprobieren und waren positiv überrascht. Die meisten Programme auf Linux arbeiteten zuverlässiger oder lieferten mehr Informationen.
Wenn ich zukünftig mit NetFlow arbeiten werde, wird meine Entscheidung zu Gunsten von NFDump und NFSen ausfallen.

19.2 Christian Jung

19.2.1 Allgemein

Für die Diplomarbeit suchte ich einen Partner, der wie ich gerne eine Netzwerkarbeit tätigen würde. Ich erfuhr, dass auch kombinierte Diplom- und Bachelorarbeiten in der Übergangszeit vom Fachhochschul- zum Bachelorstudiengang möglich sind. Dabei fragte ich Marcel Jakopic an, ob er gerne mit mir die Arbeit bestreiten würde. Er willigte ein musste dies aber noch mit seinem Arbeitgeber klären. Kurz darauf erhielt er grünes Licht, dass er die Arbeit innert 8 Wochen Vollzeit mit mir beginnen kann.

Marcel und ich versuchten schon früh eine geeignete Arbeit zu finden. Wir fragten bei etlichen Dozenten um die ausgeschriebenen Arbeiten an und hatten auch schon ein paar Favoriten.
Als die Arbeiten öffentlich ausgeschrieben wurden, fiel uns die Arbeit Router Forensics: Challenging NetFlow Accuracy von Herr Glatz auf, welche unser sofortiges Interesse weckte. Wir fragten nach welche Themen in der Arbeit behandelt werden. Bei der Arbeitszuweisung erhielten wir den Zuschlag.

Wir konnten vorab eine kurze Sitzung an der ETH Zürich abmachen und den Themeninhalt der Arbeit genauer besprechen. Dort lernten wir auch Herrn Tellenbach, der Initiant der Arbeit, kennen. Er gab uns eine Einführung in das Monitoring mit NetFlow.
Herr Glatz und Herr Tellenbach schilderten Ihre Vorstellungen der Arbeit und gaben uns den Bericht der Universität von Paris zum Durchlesen. Dieser beinhaltete Untersuchungsergebnisse der Studie über J-Flow.

Nach dem Gespräch mit Herrn Glatz und Herrn Tellenbach war ich guten Mutes die Arbeit mit Marcel in Angriff zu nehmen.

19.2.2 Erfahrungen

In den bisherigen Studienarbeiten arbeitete ich immer in einem Team. Ich kannte die Vorteile einer Teamarbeit, jedoch waren mir auch die Nachteile dessen bewusst. Wichtig für den reibungslosen Ablauf war eine gute Aufgabenteilung.

Mit Marcel Jakopic habe ich einen zuverlässigen Partner für die Arbeit gefunden. Wir haben uns gut verstanden und bei den Diskussionen gegenseitig ergänzt. Die Zusammenarbeit mit ihm war für mich sehr konstruktiv.

Gerade in einer solchen Forschungsarbeit, war es wichtig die Labs miteinander zu besprechen und gut zu planen. Zu Beginn war es einfacher die einzelnen Arbeiten zuzuweisen. Wer welche Software sucht und beurteilt oder die Skripte erstellt und den Aufbau der Labs erstellt.
Mit der Komplexität musste auch genauer im Detail geschaut werden, wie die Labs aufgebaut sein sollten. Dabei halfen die anfangs gehaltenen Diskussionen über die Messungen.

Eine Schwierigkeit war die einzelnen Messungen auseinander zu halten, da manche Messungen wiederholt werden mussten und die Planung der neuen Labs bereits am Laufen waren. Mit der Zeit merkten wir, dass die Messungsresultate genau gleich beschriftet sein müssen, um den Überblick über die einzelnen Labs nicht zu verlieren.

Was uns ebenfalls zu schaffen machte, war die Unzuverlässigkeit der Programme. Sie stürzten ab oder blieben hängen, wodurch sie einige Messungen vereitelten und wir diese wiederholen mussten. Dieser Zustand war vorwiegend bei längeren Messungen unglücklich, welche wir über Nacht laufen liessen.

Bisher machte ich mir nie Gedanken, wann welches Betriebssystem besser sein würde. Im Nachhinein musste ich feststellen das jedes Betriebssystem seine Vorzüge hat. Ein Linuxsystem, wie Ubuntu, reagiert schneller bei Veränderungen, jedoch sind spezielle Installationen mühsamer zu realisieren. Bei Windows ist meist die Installation sehr einfach, dafür reagiert das System träge, was wir bei NTP festgestellt hatten. Ebenfalls wichtig ist, ob virtuelle Computer verwendet werden oder nicht. Bei zeitkritischen Messungen würde ich davon abraten, ebenfalls senden die Dienste von VMWare Broadcasts worauf bei Netzwerkanalysen zu achten ist. Bei anderen Messungen ist die Verwendung von Vorteil, da ein System sehr leicht und schnell importiert werden kann.

Zusammenfassend fand ich es eine sehr interessante Arbeit. Sie entsprach voll und ganz meinen Interessen. Von der Arbeit habe ich viel profitiert im Zusammenhang mit Wireshark. Ich hatte bisher nur die graphische Oberfläche genutzt. Diesen per Kommandozeile zu starten, kann in manchen Situationen von grossem Nutzen sein.
Auch lernte ich viel übers Monitoring und frischte meine Kenntnisse über das Monitoring mit SNMP wieder auf. NetFlow kannte ich bisher nicht, werde es später hoffentlich an vielen Orten einsetzen können.
Die Zusammenarbeit mit Marcel Jakopic war sehr angenehm und ich würde jederzeit wieder mit ihm eine Arbeit antreten.

19.3 Dank

Während der Diplom- / Bachelorarbeit waren wir sehr auf die Unterstützung von Dritten angewiesen. Insbesondere bei der Beschaffung der Hardware konnten wir immer auf die Leute vom INS zählen. Speziellen Dank möchten wir gerne Herbert Fritschi, Maurin Egler und Oliver Rehmann aussprechen. Immer wenn etwas neues in einem Lab dazu kam, konnten wir uns mit der Bitte an die Herren wenden. Innerhalb kürzester Zeit konnten wir mit der Hardware rechnen, was uns im Ablauf der Diplom- / Bachelorarbeit sehr entgegen kam.

Ebenfalls möchten wir uns bei Herr William Boye vom INS bedanken für die Hintergrundinformationen zu NetFlow und den Tipps zur Softwarebeschaffung.

Ein Dank möchten wir auch an Herrn Bernhard Tellenbach widmen, der uns diese Arbeit beschaffte und uns ebenfalls mit Informationen versorgte.

20 ANHANG

20.1 Unterschriebene Aufgabenstellung

20.1.1 Diplomarbeit

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

INFORMATIK

Aufgabenstellung zur Diplomarbeit HS 2009

„Router Forensics: Challenging NetFlow Accuracy"

Gruppe: Ch. Jung / M. Jakopic

Ausgangssituation

Der Umfang heutigen Netzwerkverkehrs bedingt die Benutzung von summarischen Monitoringverfahren um den Umfang der gesammelten Daten auf ein handhabbares Mass zu reduzieren. Ein verbreitetes Monitoring-Datenformat stellt CISCO NetFlow dar, das in Form des IPFIX Formats inzwischen auch von der IETF aufgegriffen wurde. NetFlow, wie auch ähnliche Verfahren, können jedoch die tatsächlichen Verkehrsdaten nur in limitierter Form darstellen. Zudem hat es sich gezeigt, dass der Datensammmlungsprozess infolge von Verarbeitungskonflikten mit der Kernfunktion eines Routers bzw. Switches Fehler aufweisen kann.

Aufgabe

Das Hauptthema dieser Arbeit besteht darin, durch Vergleich von Paketdaten und NetFlow-Daten, die auf den gleichen Verkehrsdaten gesammelt wurden, den Informationsgehalt von NetFlow unter verschiedenen Betriebszuständen kritisch zu beurteilen.
Als Nebenprodukt kann bei ausreichend Zeit ein Datensatz zusammen gestellt werden, der die Beurteilung von Applikationsklassifikations-Verfahren erlaubt, die auf NetFlow-Daten basieren. Solche Datensätze sind für die Forschung notwendig, aber infolge einschränkender Datenschutzbestimmungen nicht allgemein zugänglich. Der Datensatz besteht einerseits aus den Verkehrsdaten im NetFlow-Format und andererseits aus Referenzinformationen über die enthaltenen Applikationsklassen. Die Referenzdaten werden aus den Paketdaten mit Hilfe einer spezialisierten Software abgeleitet. Der so erstellte Datensatz erlaubt es Forschern an der ETH neuartige Klassifikationsverfahren zu beurteilen.

Arbeitsumgebung

Versuchsaufbau bestehend aus einem CISCO Router, PCs als Datenquelle und als Datensenke und einem zusätzlichen PC als Monitoring-Station. Je nach Erfordernissen und verfügbaren Gerätschaften ist dieser Versuchsaufbau um weitere Elemente zu ergänzen.

Erwartete Resultate

Als Resultat dieser Arbeit soll ein Bericht entstehen, der die Analyse der gestellten Aufgabe, die gewählte Versuchsmethodik, die Resultate und ihre Interpretation beschreibt.

Ergänzend soll ein nach den unten stehenden Anforderungen aufgebauter Bericht entstehen. Abzugeben sind **drei** CD-ROMs (MS Windows kompatibel), die Rohdaten und aufbereitete Daten der Messungen, eventuell erstellte Software und den Bericht in elektronischer Form enthalten (Gesamtbericht als PDF-Datei und Originaldateien des benutzten Textsystems). Der Bericht ist zusätzlich **zweimal** in gedruckter Form mit Ringbindung (kein Ordner!) bereitzustellen. Basis für die Bewertung der Arbeit stellt primär der gedruckte Bericht dar.

Geforderte Berichtsinhalte:

1. *Resultatdokumentation:*
   Sie beschreibt alle Resultate der Arbeit und soll projektbegleitend gemäss dem für die Arbeit gewählten Vorgehensmodell erstellt werden. Sie soll alle wichtigen Informationen enthalten, die ein Ingenieur benötigt, der die Arbeitsresultate ohne vorgängige Kenntnis des Projekts weiterverwenden möchte.
2. *Projektdokumentation:*
   Sie umfasst jegliche Dokumentation, die sich auf die Durchführung der Diplomarbeit bezieht. Dazu gehören auch ein *nachgeführter Projektplan* (Arbeitspakete, Zeitplan mit Meilensteinen, Plan- und Ist-Aufwände), *Kurzprotokolle aller Besprechungen* und ein *Projektschlussbericht* (was wurde erreicht bzw. nicht erreicht, was ist in der Durchführung gut/schlecht gelaufen, Schlussfolgerungen, Ausblick auf mögliche Zusatz-und Nachfolgearbeiten). Im Bericht ist die Aufteilung der Arbeit innerhalb der Gruppe auszuweisen (thematisch und in Anzahl Arbeitsstunden).

Im weiteren soll die Gliederung und der Inhalt des Berichts den Regelungen der Abt. Informatik entsprechen (siehe Richtlinien unter den Web-Seiten der Abt. Informatik [1]). Teildokumente sollen derart in den Bericht integriert werden, dass ein hierarchisches Gesamt-Inhaltsverzeichnis mit durchgängiger Seitennummerierung ermöglicht wird. Alle obligatorischen Berichtsinhalte müssen im gedruckten Bericht vorhanden sein (nicht nur auf Abgabe-CD).
Kopien anderer Dokumente, die nicht selbst erstellt wurden, aber für weitere Arbeiten nützlich sind, sollen *separat* zum Gesamtbericht abgegeben werden (Abgabeform frei; evtl. auch nur auf Abgabe-CD).
Berichtsinhalte, die nicht selbst erarbeitet wurden [2], sind mit ihrer Quelle zu bezeichnen [3] (gilt auch für Bilder!).

[1] URL: https://www.hsr.ch/fileadmin/user_upload/hsr.ch/abt_I_crm/downloads/DokuAnleitung.pdf
[2] URL: http://www.plagiarism.org/learning_center/what_is_plagiarism.html
[3] URL: http://www.plagiarism.org/learning_center/preventing_writing.html#write

Literaturhinweise

[4] Cisco Whitepaper; „NetFlow Performance Analysis", May 2007
[5] Italo Cunhal et. al.; „Uncovering Artifacts of Flow Measurement Tools", Passive and Active Measurement Conference (PAM) 2009
[6] Cisco; „NetFlow Services Solution Guide", 22. Jan. 2007
Erhältlich unter:
http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.html

Termine

28. September 2009	Arbeitsbeginn
13. November 2009	Abgabe derKurzbeschreibung an das Abteilungssekretariat (per Email an cfurrer@hsr.ch)
20. November 2009	Abgabe des Berichts an den Betreuer

Weitere Termine siehe Terminangaben auf dem HSR-Web.

Betreuung

Betreuer: Prof. Eduard Glatz, Email: eglatz@hsr.ch
Während der Durchführung der Arbeit findet nach Möglichkeit regelmässig jede Woche eine Besprechung mit dem Betreuer statt. Dazu werden entsprechende Termine bei Arbeitsbeginn festgelegt.

Wichtige Kontaktpersonen:

Bernhard Tellenbach, Inst.f.Techn.Informatik u.Kommunik.Netze
ETZ G 97
Gloriastrasse 35
8092 Zürich

Phone: +41 44 632 70 06
E-Mail: tellenbach@tik.ee.ethz.ch

Rapperswil, den 27. Sept. 2009

20.1.2 Bachelorarbeit

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

INFORMATIK

Aufgabenstellung zur Bachelorarbeit HS 2009

„Router Forensics: Challenging NetFlow Accuracy"

Gruppe: Ch. Jung / M. Jakopic

Ausgangssituation

Der Umfang heutigen Netzwerkverkehrs bedingt die Benutzung von summarischen Monitoringverfahren um den Umfang der gesammelten Daten auf ein handhabbares Mass zu reduzieren. Ein verbreitetes Monitoring-Datenformat stellt CISCO NetFlow dar, das in Form des IPFIX Formats inzwischen auch von der IETF aufgegriffen wurde. NetFlow, wie auch ähnliche Verfahren, können jedoch die tatsächlichen Verkehrsdaten nur in limitierter Form darstellen. Zudem hat es sich gezeigt, dass der Datensammmlungsprozess infolge von Verarbeitungskonflikten mit der Kernfunktion eines Routers bzw. Switches Fehler aufweisen kann.

Aufgabe

Das Hauptthema dieser Arbeit besteht darin, durch Vergleich von Paketdaten und NetFlow-Daten, die auf den gleichen Verkehrsdaten gesammelt wurden, den Informationsgehalt von NetFlow unter verschiedenen Betriebszuständen kritisch zu beurteilen.
Als Nebenprodukt kann bei ausreichend Zeit ein Datensatz zusammen gestellt werden, der die Beurteilung von Applikationsklassifikations-Verfahren erlaubt, die auf NetFlow-Daten basieren. Solche Datensätze sind für die Forschung notwendig, aber infolge einschränkender Datenschutzbestimmungen nicht allgemein zugänglich. Der Datensatz besteht einerseits aus den Verkehrsdaten im NetFlow-Format und andererseits aus Referenzinformationen über die enthaltenen Applikationsklassen. Die Referenzdaten werden aus den Paketdaten mit Hilfe einer spezialisierten Software abgeleitet. Der so erstellte Datensatz erlaubt es Forschern an der ETH neuartige Klassifikationsverfahren zu beurteilen.

Arbeitsumgebung

Versuchsaufbau bestehend aus einem CISCO Router, PCs als Datenquelle und als Datensenke und einem zusätzlichen PC als Monitoring-Station. Je nach Erfordernissen und verfügbaren Gerätschaften ist dieser Versuchsaufbau um weitere Elemente zu ergänzen.

Erwartete Resultate

Als Resultat dieser Arbeit soll ein Bericht entstehen, der die Analyse der gestellten Aufgabe, die gewählte Versuchsmethodik, die Resultate und ihre Interpretation beschreibt.

Ergänzend soll ein nach den unten stehenden Anforderungen aufgebauter Bericht entstehen. Abzugeben sind **drei** CD-ROMs (MS Windows kompatibel), die Rohdaten und aufbereitete Daten der Messungen, eventuell erstellte Software und den Bericht in elektronischer Form enthalten (Gesamtbericht als PDF-Datei und Originaldateien des benutzten Textsystems). Der Bericht ist zusätzlich **zweimal** in gedruckter Form mit Ringbindung (kein Ordner!) bereitzustellen. Basis für die Bewertung der Arbeit stellt primär der gedruckte Bericht dar.

Geforderte Berichtsinhalte:

1. *Resultatdokumentation:*
   Sie beschreibt alle Resultate der Arbeit und soll projektbegleitend gemäss dem für die Arbeit gewählten Vorgehensmodell erstellt werden. Sie soll alle wichtigen Informationen enthalten, die ein Ingenieur benötigt, der die Arbeitsresultate ohne vorgängige Kenntnis des Projekts weiterverwenden möchte.
2. *Projektdokumentation:*
   Sie umfasst jegliche Dokumentation, die sich auf die Durchführung der Bachelorarbeit bezieht. Dazu gehören auch ein *nachgeführter Projektplan* (Arbeitspakete, Zeitplan mit Meilensteinen, Plan- und Ist-Aufwände), *Kurzprotokolle aller Besprechungen* und ein *Projektschlussbericht* (was wurde erreicht bzw. nicht erreicht, was ist in der Durchführung gut/schlecht gelaufen, Schlussfolgerungen, Ausblick auf mögliche Zusatz-und Nachfolgearbeiten). Im Bericht ist die Aufteilung der Arbeit innerhalb der Gruppe auszuweisen (thematisch und in Anzahl Arbeitsstunden).

Im weiteren soll die Gliederung und der Inhalt des Berichts den Regelungen der Abt. Informatik entsprechen (siehe Richtlinien unter den Web-Seiten der Abt. Informatik [1]). Teildokumente sollen derart in den Bericht integriert werden, dass ein hierarchisches Gesamt-Inhaltsverzeichnis mit durchgängiger Seitennummerierung ermöglicht wird. Alle obligatorischen Berichtsinhalte müssen im gedruckten Bericht vorhanden sein (nicht nur auf Abgabe-CD).
Kopien anderer Dokumente, die nicht selbst erstellt wurden, aber für weitere Arbeiten nützlich sind, sollen *separat* zum Gesamtbericht abgegeben werden (Abgabeform frei; evtl. auch nur auf Abgabe-CD).
Berichtsinhalte, die nicht selbst erarbeitet wurden [2], sind mit ihrer Quelle zu bezeichnen [3] (gilt auch für Bilder!).

[1] URL: https://www.hsr.ch/fileadmin/user_upload/hsr.ch/abt_I_crm/downloads/DokuAnleitung.pdf
[2] URL: http://www.plagiarism.org/learning_center/what_is_plagiarism.html
[3] URL: http://www.plagiarism.org/learning_center/preventing_writing.html#write

Literaturhinweise

[4] Cisco Whitepaper; „NetFlow Performance Analysis", May 2007
[5] Italo Cunhal et. al.; „Uncovering Artifacts of Flow Measurement Tools", Passive and Active Measurement Conference (PAM) 2009
[6] Cisco; „NetFlow Services Solution Guide", 22. Jan. 2007
Erhältlich unter:
http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.html

Termine

28. September 2009	Arbeitsbeginn
13. November 2009	Abgabe derKurzbeschreibung an das Abteilungssekretariat (per Email an cfurrer@hsr.ch)
20. November 2009	Abgabe des Berichts an den Betreuer

Weitere Termine siehe Terminangaben auf dem HSR-Web.

Betreuung

Betreuer: Prof. Eduard Glatz, Email: eglatz@hsr.ch
Während der Durchführung der Arbeit findet nach Möglichkeit regelmässig jede Woche eine Besprechung mit dem Betreuer statt. Dazu werden entsprechende Termine bei Arbeitsbeginn festgelegt.

Wichtige Kontaktpersonen:

Bernhard Tellenbach, Inst.f.Techn.Informatik u.Kommunik.Netze
ETZ G 97
Gloriastrasse 35
8092 Zürich

Phone: +41 44 632 70 06
E-Mail: tellenbach@tik.ee.ethz.ch

Rapperswil, den 27. Sept. 2009

20.2 Erklärung

Die Diplom- / Bachelorarbeit Router Forensics: Challenging Netflow Accuracy von Marcel Jakopic und Christian Jung während des Herbstsemester 2009 an der HSR Hochschule für Technik Rapperswil wurde selber und ohne fremde Hilfe durchgeführt, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde.

In der Diplom- / Bachelorarbeit Router Forensics: Challenging Netflow Accuracy sind sämtliche verwendeten Quellen erwähnt und im Literaturverzeichnis aufgelistet.

Die Autoren bestätigen hiermit, dass die gemachten Angaben korrekt sind und die Arbeit ohne fremde Hilfe, ausser den angegebenen Quellen, erstellt wurde.

Rapperswil, 20. November 2009

Marcel Jakopic Christian Jung

20.3 Glossar

Begriff	Beschreibung / Erklärung
Active Timeout	Diese Einstellung für NetFlow veranlasst das Netzwerkgerät, die Flows ab diesem Zeitpunkt abzuschliessen, auch wenn der reale Flow noch nicht zu Ende ist.
Capturefile	Wireshark empfängt alle Netzwerkpakete und listet diese auf. Die Liste kann aus Wireshark gespeichert werden und beinhaltet alle Headerinformationen der jeweiligen Protokolle und den Payload. Diese Liste wir oft auch Capturefile genannt.
Capturefilter	In Wireshark können Filter gesetzt werden, um nur spezifische Paketdaten zu empfangen und aufzuzeichnen. Diese Filter nennt man Capturefilter.
Cisco IOS	Internetwork Operating System Software ist das Betriebssystem von Cisco Routern und Cisco Switches.
CLI – Command Line Interface	Das Interface erwartet Befehle über einen Promt. Es kommt komplett ohne GUI aus. Beispiele sind DOS (Eingabeaufforderung) und das Cisco-CLI, über welches Switches und Router konfiguriert werden können.

Filesharing	Filesharing bedeutet gemeinsamer Dateizugriff und bezeichnet das direkte Weitergeben von Dateien zwischen Benutzern des Internets.
Flow	Ein Flow bezeichnet eine Verbindung, welche zwischen 2 Geräten stattfindet. Es wird durch den Quell- und Ziel-Port, Quell- und Ziel-IP und durch das Protokoll definiert. Ein Flow ist unidirektional, das heisst, dass eine TCP-Verbindung aus 2 Flows besteht. UDP wird anhand der Definition unterschieden.
Freeware	Ist eine Software, welche vom Urheber kostenlos zur Benutzung zur Verfügung gestellt wird.
Frühzeitiger Export	Die Flows werden exportiert, wenn der active Timeout Zähler überschritten wurde, auch wenn die tatsächliche Verbindung immer noch weiter existiert.
GUI	GUI ist die Abkürzung von Graphical User Interface. Es wird damit die Benutzeroberfläche einer Applikation angesprochen
IETF	Die Internet Engineering Task Force ist eine Organisation, die sich mit der technischen Weiterentwicklung des Internets befasst.
LED	LED ist die Abkürzung von Light Emitting Diode und beschreibt eine Leuchtdiode.
MIB – Management Information Base	In dieser Datenbasis sind beliebig viele Objekte enthalten, welche über OID's angesprochen werden können. Der Standard wird in vielen RFC's von IETF (http://en.wikipedia.org/wiki/IETF) beschrieben.
MIB-Tree	Jeder Hersteller kann innerhalb von seinem Ast im MIB eigene Strukturen definieren. Diese können bei geeigneten Programmen importiert werden und Helfen durch eine Beschreibung zum richtigen OID zu kommen.
NetFlow	Cisco hat diese Technik entwickelt und hat sie Cisco NetFlow benannt. Siehe Flow
NetFlow-Kollektor	Ist eine Applikation auf einem Server, welche auf dem Port horcht, auf dem die Netzwerkgeräte die NetFlow-Pakete senden. Meistens werden die NetFlows in eine Datenbank geschrieben, damit sie anschliessend analysiert werden können.
NetFlow-Paket	Ein NetFlow-Paket wird als UDP-Paket versendet. In einem Paket können bis zu 30 NetFlows stehen, mindestens jedoch 1 NetFlow.

OID – Object Identifier	Die Objektidentikation wir verwendet, um ein Objekt absolut oder relativ zu identifizieren. Dies wird z.B. für LDAP, X.509 usw. gebraucht. In diesem Bericht wird OID für SNMP verwendet, indem die OID ein Objekt in der MIB beschreibt.
Passive Timeout	Diese Einstellung für NetFlow veranlasst das Netzwerkgerät, nach dieser Zeit das NetFlow-Paket zu versenden.
PDU	PDU ist die Abkürzung von Protocol Data Unit. Da Netzwerkprotokolle ineinander verschachtelt werden ist die PDU die integrierten Protokoll-Headerinformationen mit den eigentlichen Daten.
Scheduler	Ein Scheduler startet eine Applikation oder ein Script. Die Ausführbare Datei kann somit zeitabhängig gesteuert werden und kann zur gewünschten Zeit ablaufen.
SNMP – Simple Network Management Protocol	Dieses Protokoll wird gebraucht, um auf möglichst einfachem Weg, Daten von einem Gerät abfragen zu können. Es lassen sich praktisch alle Details von einem Netzwerkgerät, solange in der MIB vorhanden, abfragen und bei genügender Berechtigung, sogar verändern.
Streaming	Streaming ist die kontinuierliche Abfolge von Datensätzen, deren Ende nicht im Voraus abzusehen ist.
TCL	TCL heisst Tool Command Language und wird in Cisco IOS verwendet, um Befehle oder auch Scripts auszuführen.
Timestamps	Der Zeitstempel ist ein definierter Wert, der es ermöglicht, dass verschiedene Systeme die Zeit daraus ermitteln können. Anhand dieser kann eine Reihenfolge erstellt werden. Die grosse Schwierigkeit bei verteilten Systemen ist, diese korrekt zu synchronisieren.
Trial-Version	Ist eine Art des Softwarevertriebes. Die Software kann während einer begrenzten Zeit benutzt werden, danach muss sie registriert und bezahlt werden. Meistens ist der volle Umfang freigeschaltet.
Varianz	Ist ein mathematisches Mass, welches beschreibt, wie stark die Streuung der Messgrösse ist. Sie wird berechnet indem man alle Abstände von der Messung zum Mittelwert quadriert und zusammenzählt und anschliessend durch die Anzahl Werte dividiert.
VoIP	Mit Voice over IP ist das Telefonieren über Computernetzwerke gemeint.
Wireshark	Wireshark ist ein frei erhältliches Programm, welches zur Analyse von Netzwerkpaketdaten dient und alle empfangenen Pakete aufzeichnet.

WMI – Windows Management Instrumentation	Die Idee von WMI ist eine umgebungsunabhängige Schnittstelle anzubieten, welche es erlaubt, auf die Einstellungen lesend und schreibend zuzugreifen. Es funktioniert wie SNMP für Netzwerkgeräte, einfach für Windows Betriebssysteme.

Tabelle 20-1 Glossar

20.4 Abbildungsverzeichnis

20.5 Tabellenverzeichnis

20.6 Index

K

L

M

N

O

P

R

S

T

U

V

W

20.7 Sitzungsprotokolle

Protokoll 01 vom 28.09.2009

Anwesende

Betreuer:	Eduard Glatz	(eg)
Studenten:	Marcel Jakopic	(mj)
	Christian Jung	(cj)

Sitzung

Sitzungsbeginn: 09:00 Uhr Ende: 10:00 Uhr

Meetingminutes

Thema	Wer	Detail
Arbeiten	mj/cj	Inbetriebnahme des Routers
	mj/cj	CPU Auslastung testen - Messwerte ablegen (sollten sicher 20 Messpunkte sein)
	mj/cj	IOS kennenlernen und Skriptfähigkeit testen
	mj/cj	Netflow kennenlernen exportieren und analysieren - Was wird erfasst, was nicht - Was passiert mit fragmentierten Netflowdaten
	mj/cj	Grobevaluation von Trafficgenerator - Aufgabenanalyse (Anforderung an die Tg.)
	mj/cj	Berichte vorbereiten - Diagramm erzeugen und importieren - Ideenkatalog notieren - Vergleich Paket-/Netflow export)
	eg	Aufgabenstellung senden

Diskussion

keine

Nächste Sitzung

Datum:	02.10.2009	
Raum:	1.111 (HSR)	
Uhrzeit:	17:00 Uhr	Ende: ca. 17:30 Uhr

Protokoll 02 vom 02.10.2009

Anwesende

Betreuer:	Eduard Glatz	(eg)
Studenten:	Marcel Jakopic	(mj)

Sitzung

Sitzungsbeginn: 16:10 Uhr Ende: 17:25 Uhr

Meetingminutes

Thema	Wer	Detail
Sitzungsprotokoll	bt	Protokoll soll ausführlicher sein, es sollten Ziele bestimmt werden, welche überprüft werden können
Aufbau Technischer Bericht	mj/eg	Auf der Internetseite (systemsoftware.hsr.ch) gibt es viele Beispiele von SA/BA zu begutachten. Projektmanagement kann demnach im Anhang oder als eigenes Kapitel im Bericht stehen.
Messungsaufbau	eg	Die einzelnen Messungen sollen mehrmals wiederholt werden. Richtzahl: 15-30 Muster
	eg	Für die Gesamtmessung (Lab) sollen Minimum, Maximum und Varianz (evtl. noch die Standardabweichung) angegeben werden.
Softwareauswahl	mj/cj	Liste mit möglichen Tools erstellen, klassifizieren und mit den Erwartungen bewerten (einfache Gegenüberstellung nach Beispiel Q-Feedback)
	mj	Lanforge-Trial herunterladen und ausprobieren
	mj/cj	Konvertierung Netflow zu Daten und vice versa ermitteln
Administration	mj	Projektplan per Mail schicken
	mj/cj	Labs definieren, was soll genau getestet werden, ausführliche Beschreibung
	bt/eg	Definition Labs überprüfen und Wünsche / Verbesserungsvorschläge mitteilen
	eg	Tools für Datengenerierung per Mail schicken
	mj	Es stehen SLOTS zum Eintragen der nächsten Sitzung bereit. Die Zettel hängen bei 1.208

Diskussion

Sitzungsprotokoll: Die 1. Woche war eine Einarbeitungsphase. Kennenlernen der Umgebung, Installationen verschiedener Programme, kein direktes Ziel geplant.

Problematik der Softwareauswahl: Im Internet sind sehr viele Tools erhältlich. Diese alle zu installieren und zu testen ist ziemlich aufwändig. Es kann Herr Tellenbach angefragt werden, ob SW empfohlen werden kann.
Von Freeware über sehr teure Produkte ist alles vorhanden. Trial-Versionen testen und sich mit Tricks behelfen (Installation nach Ablauf auf anderem Computer / Datum zurückstellen)

Es gibt in Wireshark eine Einstellung, welche es erlaubt, nur die Header mitzuschneiden, somit kann das Problem der grossen Datenspeicherung bei einem Bandbreitentest behoben werden. Zusätzlich liegen zwei 10 Mbit Interface Karten für den Router, aus dem Privatbestand eines Kollegen von Herr Jung, im Arbeitsraum bereit.

Auf das Know-How vom INS kann zurückgegriffen werden.

Nächste Sitzung

Datum:	09.10.2009	
Raum:	1.111 (HSR)	
Uhrzeit:	15:00 Uhr	Ende: ca. 16:00 Uhr

Protokoll 03 vom 09.10.2009

Anwesende

Betreuer:	Eduard Glatz	(eg)
Studenten:	Marcel Jakopic	(mj)
	Christian Jung	(cj)

Sitzung

Sitzungsbeginn: 15:00 Uhr Ende: 16:00 Uhr

Meetingminutes

Thema	Wer	Detail
Netflow Spezifikation	mj / cj	Gemäss Spezifikation werden Netflowexporte von TCP- oder UDP-Verbindungen zu Flows zusammengefasst. Herausfinden wie es in der Spezifikation beschrieben ist und mit der Realität vergleichen.
	mj / cj	Hardware und Softwareberechnungen für Netflows gemäss Spezifikation des Routers nachschauen.
	mj /cj	Exporteinstellungen für Netflowdaten variieren und Unterschiede festhalten.
Messversuche	mj /cj	Messungen für Labs genau definieren (Ziel, Randbedingungen, Challanges/Risiken) und die Beschreibung an eg und bt senden. (Was will gemessen werden? Welche Bedingungen müssen dafür erfüllt sein? Was ist zu erwarten? Was sind die Resultate der Messung?)
	eg / bt	Definitionen für Messversuche überprüfen. Ev. Änderungen an Messversuche oder andere Messversuche melden.

	mj /cj	Variationen für Trafficgenerierung erstellen (Verkehrsmix / realistischer Netzwerkverkehr). Asymetrisches Verkehrsvolumen (Web, Game,...) Heterogenität des Verkehrs (Auswirkung von regelmässigem Traffic zu variierendem Trafficmix? Welche Werte im Trafficmix müssen variieren?)
Softwareauswahl	mj/cj	Evaluation von Software (Traffic Generatoren) fertig erstellen und als Dokumentation an eg übergeben.

Diskussion

Der Fokus für die kommende Woche liegt bei den Definitionen der Messversuche. Diese sollen anfangs Woche an Herrn Glatz und Herrn Tellenbach übermittelt werden. So könnte allenfalls auf unklare Messversuche hingewiesen werden und/oder wünschenswerte Messversuche hinzugefügt werden.

Für die Traffic Generation soll ein Trafficmix erstellt werden, welcher der Realität nahe kommt. Dabei muss auf asynchrone Verkehrsvolumen geachtet werden. Beim surfen im Web stellt der Client eine kurze Anfrage (Get- oder Post-Request) und erhält vom Server eine grössere Menge an Informationen, wie Dokumente, Bilder, Video und Text (Response). Weitere Beispiele für asynchrone Kommunikation sind VoiceoverIP, Gamemessages.

Meist genutzter Traffic Generator ist Iperf, da dieser einfach mit Scripts zu bedienen ist. Bei LANforge war die Installation recht knifflig und leider verfügt die Freeware-Version nicht über alle Tools für die Generierung von Traffic, wie die Vollversion. Weswegen Iperf auch mit LANforge mithalten kann. Ein weiteres Tool kann verschiedene Protokolle verwenden, leider kann dieses nur manuell über das GUI gesteuert werden.

Hardware für unseren Router (Cisco Router 2621) wurde erweitert. Mitarbeiter vom INS hatten uns ein Ethernet-Modul für den Router gesucht, welches weitere 4 Ethernet-Ports zur Verfügung stellt.

Ungewollten Verkehr vom Client kann anstelle des Services abstellen, auch der Dienst aus dem entsprechendem Interface entfernt werden. So liegen weniger ungewollte Pakete für die Messung vor.

Nächste Sitzung

Datum:	16.10.2009	
Raum:	1.111 (HSR)	
Uhrzeit:	13:30 Uhr	Ende: ca. 14:30 Uhr

Protokoll 04 vom 16.10.2009

Anwesende

Betreuer:	Eduard Glatz	(eg)
Studenten:	Marcel Jakopic	(mj)
	Christian Jung	(cj)

Sitzung

Sitzungsbeginn: 13:30 Uhr Ende: 15:15 Uhr

Meetingminutes

Thema	Wer	Detail
Messversuche Lab 1	mj / cj	Messungen werden wiederholt mit den definierten Einstellungen.
Messversuche Lab2	mj / cj	Test 1 (Bandbreitenbelastung) hat gezeigt, dass Messungen nicht an einem produktiven Interface gemacht werden sollen. Dies entspricht aber nicht der Praxis. (Lab2 nicht mehr weiter verfolgen)
	mj / cj	Im Test 1 wurden nur die Datenpakete angeschaut. Jedoch kann somit nicht gesagt werden, ob der Inhalt der Datenpakete vollständig ist. Sequenznummer der Netflowdaten auslesen (Herr Tellenbach dafür anfragen).
	mj /cj	Test 2 (Zeitdifferenz) immer grösser werdender Unterschied der ankommenden Netflowdatenpakete könnte mit der Clockrate der PC zu tun haben. Um dies zu belegen werden beide PC vertauscht und die Messung erneut gemacht.
Messversuche Lab3	mj /cj	Inhalt der Netflowdaten überprüfen in Lab3. Stimmen Paketinhalte bei Fragmentierten Daten. Dabei auch die Flowlänge messen und prüfen, ob maximale Exportzeit ausgeschöpft wird.
	mj /cj	Stimmen PCAP und Netflow überein: - bei keiner Fragmentierung - bei Fragmentierung - bei länger werdenden Flows

	mj /cj	Ist Netflowexport exakt? - Wann bricht Monitoring ab bei CPU-Belastung? - Wann ist Netflowexport nicht mehr vollständig? - Gibt es Erkennungssignale dafür (Signale für Unterbruch)?
Traffic Mix	mj/cj	Datenerzeugung muss definiert werden Primäre Charakteristik: - Grösse - Wiederholrate - Verkehrsteilnehmer Realitätsnähe definieren, Vereinfachung muss begründet und Nutzen aufgelistet werden. Tagesablauf definieren und daraus Traffic ableiten. Definition an eg und bt senden für Feedback

Diskussion

Bei Lab1 wurden Störungen festgestellt, welche nicht sein sollten. Da Messung nicht optimal durchgeführt wurde, wird diese nochmals durchgeführt.

Lab2 hat sich mit der Messung von Netflowdaten beschäftigt, jedoch nicht mit dem Inhalt der Daten. Herr Glatz hätte gewünscht, dass die Daten miteinbezogen werden.

In Lab3 werden die Netflowdaten mit dem Inhalt genauer betrachtet. Der ganze Prozess der Datenüberprüfung wurde aufgestellt, dabei wurden noch fehlende Software festgestellt. Eventuell kann Herr Tellenbach dafür angefragt werden.

Unsere erste Trafficmixeinteilung ist bereits komplex. Diese sollen nun genau definiert werden und auch umgesetzt werden. Die Definition soll an Herrn Glatz und an Herrn Tellenbach gesendet werden.

Nächste Sitzung

Datum:	Freitag 23.10.2009	
Raum:	Cafeteria (HSR)	
Uhrzeit:	13:30 Uhr	Ende: ca. 14:30 Uhr

Protokoll 05 vom 23.10.2009

Anwesende

Betreuer:	Eduard Glatz	(eg)
Studenten:	Marcel Jakopic	(mj)
	Christian Jung	(cj)

Sitzung

Sitzungsbeginn: 13:30 Uhr Ende: 14:30 Uhr

Meetingminutes

Thema	Wer	Detail
Messversuche Lab 1b	mj / cj	Die Daten von der neuen Messung sollen an Herrn Tellenbach gesendet werden.
Messversuche Lab 2	mj / cj	Messung wurde gemäss letzter Sitzung nochmals untersucht und festgestellt, dass die NetflowSequenzNummer während der Auslastung wirklich unterbrochen war.
Messversuche Lab 3	mj / cj	Im Test 1 wurden immer kleinere Datenmengen erzeugt und somit viele Flows generiert. Dabei kamen wir an die Grenze von Iperf. Damit wir genügend Messdaten erhielten, wurden anstelle kleineren Datenpaketen mehrere Verbindungen eingerichtet.
	mj /cj	Im Test 2 wurde die CPU Auslastung ausgegereizt. Dabei wurde unter die Lupe genommen, ob Netflows zwischengespeichert und diese vollständig übermittelt werden. TCL stand nicht zur Verfügung, obwohl das IOS dies unterstützen sollte.
	mj / cj	Dokumentation von Lab 3 an Herrn Glatz und Herrn Tellenbach senden. Zur Überprüfung und Vollständigkeit.
Messversuche Lab 4	mj /cj	Für Lab 4 müssen folgende Themen berücksichtigt werden: - Memorytests - Speicherausbau - Sinn (wie Realitätsnah) ist Speicherausbau
Messversuche Lab 5	mj /cj	Lab 5 (Zeitgenauigkeit) mit Lab 6 tauschen.

	Bereits Gedanken machen wie die Genauigkeit gemessen werden kann. - über NTP (zweite Netzwerkkarte einbauen und diesen PC als Referenzuhr nehmen) - Mit Funkuhr - Was gilt als Referenzuhr (Atomuhr/Funkuhr) - Wie können Abweichungen gemessen werden.

Diskussion

Eine Vorabversion unserer Dokumentation an Herr Glatz und an Herr Tellenbach senden. Dabei sollten abgeschlossenen Labs vollständig erfasst werden. In jedem Lab sollte die Situation nachgestellt werden können. Auch die Daten müssen reproduzierbar sein.
Es soll die Erwartungen und die Resultate beinhalten.

Nächste Sitzung

Datum:	Freitag 30.10.2009	
Raum:	Cafeteria (HSR)	
Uhrzeit:	13:30 Uhr	Ende: ca. 14:30 Uhr

Protokoll 06 vom 30.10.2009

Anwesende

Betreuer:	Eduard Glatz	(eg)
Studenten:	Marcel Jakopic	(mj)
	Christian Jung	(cj)

Sitzung

Sitzungsbeginn: 13:30 Uhr Ende: 14:30 Uhr

Meetingminutes

Thema	Wer	Detail
Messversuche Lab 4	mj /cj	Im Lab 4 wurde die FlowTable auf die minimale Grösse von 1024 Einträgen konfiguriert.
	mj /cj	Die Auswertung erwies sich als sehr schwierig, da es ein paar Fehler gab. Wir vermuten dass die Verbindung zu den VMWare Servern, sowie die virutellen Server selbst, diese verursachten. Wir kennen leider das Netzwerk nicht, welche zwischen unserem Router und den VMWare Servern zwischengeschaltet ist.
	mj / cj	Die Messung wird nochmals wiederholt als Lab 4b, mit der Änderung, dass ein zusätzlicher PC direkt am Router angeschlossen wird.
Messversuche Lab 5	mj /cj	Genauigkeiten von Zeitabgleich kann mit NTP erfolgt werden. Die Zeiten sollten gemäss Spezifikationen besser sein, als die von Funkuhren. NTP-Abgleich sollte über separaten Port des Routers geschehen.
	mj / cj	Als NTP Server möchten wir den Router selbst nehmen. Cisco hat eventuell in den Spezifikationen des Routers etwas über die Genauigkeit notiert.
	mj / cj	Im Lab 5 soll nun die Genauigkeit von Netflow unter die Lupe genommen werden. Eventuell können wir in der Abweichung der Start- und Endzeit ein Schema erkennen.

Diskussion

Herr Glatz empfehlt uns die Dokumentation anderen Kollegen zu zeigen. Sie könnten den Inhalt und deren Aussage begutachten und uns ein Feedback geben, ob der Text korrekt und verständlich ist.

Beim Lab4 muss in der Analyse auf die Verteilung geachtet werden (Flows im "Normalzustand (Cisco oder Switch Empfehlung)" zur kleinen FlowTable mit 1024 Einträgen).

Herr Glatz kündigte uns als Gegenleser Herr Heinzmann an. Eventuell ändert sich dies noch. Herr Glatz wird uns Bescheid geben.

Wegen der Abgabe der Dokumentation müssen neue Vorlagen beachtet werden, wenn diese zwei Wochen vor Ende der Diplom- / Bachelorarbeit erscheinen. Herr Glatz möchte alles am Montag in seinem Postfach haben (3 CDs und 2 Dokumentationen).

Nächste Sitzung

Datum:	Freitag 06.11.2009	
Raum:	Cafeteria (HSR)	
Uhrzeit:	13:30 Uhr	Ende: ca. 14:30 Uhr

Protokoll 07 vom 06.11.2009

Anwesende

Betreuer:	Eduard Glatz	(eg)
Studenten:	Marcel Jakopic	(mj)
	Christian Jung	(cj)

Sitzung

Sitzungsbeginn: 13:30 Uhr Ende: 14:45 Uhr

Meetingminutes

Thema	Wer	Detail
Dokumentation	mj /cj	Die Aufgabenstellung muss als Original von Herr Glatz in unsere Dokumentation aufgenommen werden. (Dieses kann zuvorderst im Anhang genommen werden)
	mj /cj	Begriffe in Bilder und Text definieren, damit keine Unklarheiten entstehen. Im Glossar Abkürzungen und spezielle Ausdrücke aufnehmen.
	mj / cj	Zu Abgaben: Abstract, Kurzfassung und Kurzbeschreibung ist das gleiche. Im Reglement den Inhalt der Broschüre nachlesen, ob es dem Management Summary entspricht. Ebenfalls informieren über Inhalt des A0-Posters.
Bewertung	mj / cj	Herr Rinkel ist Gegenleser
	mj /cj	Bewertung erfolgt mit Gewichtung zu gleichen Teilen für Projektorganisation, Bericht, Analyse, Design, Durchführung.
Lab 4	mj / cj	Auf dem Router (zusätzliche Steckkarte) haben wir lost carrier Fehler. Um Fehler zu beheben, kann Interface gewechselt oder Steckkarte gereinigt werden.
Lab 7	mj / cj	Zeitgenauigkeit besser testen mit spezieller Netzwerkkarte vom INS. PC ist gleichzeitig Sender und Empfänger. Schauen welches OS geeigneter ist (Win/Linux). Netzwerkkartenspezifikation durchlesen.

Mündliche Prüfung / Präsentation	mj / cj	Eine Präsentation von ca. 20 Minuten über die Arbeit. Danach folgen Fragen von Herr Glatz, Herr Rinkel und einem externen Experten.
	mj / cj	Mehrere Terminvorschläge für die mündliche Prüfung / Präsentation an Herr Glatz senden.

Diskussion

Wie wir die Aufgabenstellung in unsere Dokumentation nehmen, muss dem Reglement entsprechen der Abt. I entsprechen. Deshalb müssen wir die Originalaufgabenstellung einfügen und nicht diese hineinkopieren. Wo wir die Aufgabenstellung einfügen (Vorne oder im Anhang) sei uns überlassen.

Bei der Punkteverteilung gibt es 5 gleich grosse Bewertungsthemen. Dazu gehören Projektorganisation, Bericht, die Analyse (d.h. die Aufgabenanalyse, die grundsätzliche Versuchsplanung), das Design (die Planung der einzelnen Labs / Experimentplanung) und die Durchführung (erfasste Resultate und Interpretation).

Einschubkarte bei Router wegen lost carrier Fehlern reinigen. Goldkontakte mit Radiergummi reinigen und Versuch nochmals starten.

Nächste Sitzung

Datum:	Freitag 13.11.2009	
Raum:	Cafeteria (HSR)	
Uhrzeit:	13:30 Uhr	Ende: ca. 14:30 Uhr

Protokoll 08 vom 13.11.2009

Anwesende

Betreuer:	Eduard Glatz	(eg)
Studenten:	Marcel Jakopic	(mj)
	Christian Jung	(cj)

Sitzung

Sitzungsbeginn: 13:30 Uhr Ende: 14:15 Uhr

Meetingminutes

Thema	Wer	Detail
Abstract	mj /cj	Das Abstract muss erweitert werden. Darin enthalten muss die Motivation sein, welche beschreibt wieso die Arbeit aufgenommen wird.
	mj /cj	Jedes Lab soll im Abstract mit zwei Sätzen beschrieben werden.
	mj / cj	Ebenfalls muss das Abstract die Resultate und die neuen Erkenntnisse beinhalten.
	mj / cj	Als Schema fürs Abstract kann nach SPSE vorgegangen werden. S - Situation (beschreibt Motivation) P - Problem (allgem. Aufgabenbeschreibung) S - Solution (Lösung) E - Evaluation (Resultate und Erkenntnisse)
Management Summary	mj /cj	Bis auf die Aufgabenstellung ist alles enthalten. Als Bild empfiehlt uns Herr Glatz die 4 Border Router von Switch.
	mj / cj	Im Management Summary können Fussnoten drin sein. Herr Glatz rät uns aber diese heruaszunehmen.
	mj / cj	Der Experte der Credit Suisse heisst Roberto Pajetta. Das Themagebiet unserer Arbeit gehört keinem Institut an, deshalb soll sie unter "Verschiedenes" laufen.
Lab 7	mj / cj	Das letzte Lab wurde durchgeführt. Die Analyse fehlt noch, weil sie sehr aufwändig ist. Sollte aber noch bis am Abend fertig werden.

A0-Poster	mj / cj	Posterbilder sind in einer sehr hohen Auflösung von 1600x1200 zu liefern. Herr Glatz rät uns, sie zu vergrössern, obwohl die Bilder dann unschärfer werden. Sie sollten aber immer noch gut erkennbar sein.
Präsentation	mj /cj	Das Poster kann an der Präsentation genutzt werden. Es ist aber grundsätzlich uns überlassen, ob wir es einsetzen möchten. In der Regel wird eine PowerPoint-Präsentation verwendet. Diese soll unseren Vortrag nur unterstützen (keine Sätze)
	mj /cj	Die Präsentation soll an Herr Tellenbach gesandt werden, damit er sich einen kurzen Überblick der Arbeit verschaffen kann.

Diskussion

Herr Glatz erklärte uns wie die ETH mit SWITCH zusammen arbeitet. Die ETH entwickelt für die SWITCH eine Analysealgorithmen um schnellere Diagnosen zu erzielen. Dafür erhält die ETH NetFlow-Datensätze der SWITCH.
In der Dokumentation soll diese Zusammenarbeit zwischen der ETH und der SWITCH nur kurz erwähnt werden.

Nächste Sitzung

Es findet keine Sitzung mehr statt.

20.8 Literaturverezichnis

[1] Cisco Systems, [Online] http://www.cisco.com
[2] NetFlow, [Online] http://en.wikipedia.org/wiki/NetFlow
[3] RFC Standard 3954, [Online] http://www.ietf.org/rfc/rfc3954.txt
[4] Internet Engineer Task Force, [Online] http://www.ietf.org
[5] RFC Standard 3917, [Online] http://www.ietf.org/rfc/rfc3917.txt
[6] SWITCH, [Online] http://www.switch.ch
[7] ETH Zürich, [Online] http://www.ethz.ch
[8] Juniper Networks, [Online] http://www.juniper.net
[9] S.B. Moon et al., [Dokument] Uncovering_Artifacts_of_Flow_Measurement_Tools_PAM2009.pdf
[10] Université Pierre et Marie Curie, [Online] http://www.upmc.fr
[11] Cisco Systems, [Dokument] DataSheet Cisco 2600.pdf
[12] Ubuntu, [Online] http://www.ubuntu.com
[13] MRTG, [Online] http://oss.oetiker.ch/mrtg
[14] PERL, [Online] http://www.activestate.com/store/activeperl/download
[15] RRDtool, [Online] http://oss.oetiker.ch/rrdtool/
[16] Paessler, [Online] http://www.paessler.com
[17] iReasoning MIBBrowser, [Online] http://www.ireasoning.com/mibbrowser.shtml
[18] IPSwitch WhatsUpGold, [Online] http://www.whatsupgold.com
[19] SNMPWalk, [Online] http://net-snmp.sourceforge.net/docs/man/snmpwalk.html
[20] Portfolio Auswertung, [Dokument] Portfolio Analyse_Monitoring.xlsx
[21] Plixer, [Online] http://www.plixer.com
[22] Mozilla Firefox, [Online]http://www.mozilla-europe.org/de/firefox
[23] Cisco Systems, [Online] http://www.manageengine.com/products/netflow/cisco-netflow.html
[24] NFDump, [Online] http://nfdump.sourceforge.net
[25] NFSen, [Online] http://nfsen.sourceforge.net
[26] Portfolio Auswertung, [Dokument] Portfolio Analyse_NetFlow Analyzer.xlsx
[27] PCAP2Flow, [Online] http://www.unleashnetworks.com/open-source/pcap2netflow.html
[28] Softflowd, [Online] http://www.mindrot.org/projects/softflowd
[29] NSASoft, [Online] http://www.nsasoftware.com
[30] CandelaTech LANforge Fire, [Online] http://www.candelatech.com
[31] Colasoft Packet Builder, [Online] http://www.colasoft.com/packet_builder/
[32] Iperf, [Online] http://www.softpedia.com/get/Network-Tools/Misc-Networking-Tools/Iperf.shtml
[33] Iperf Entwickler, [Online] http://en.wikipedia.org/wiki/Iperf
[34] Download Plattform, [Online] http://www.softpedia.com
[35] Traffic 0.1.3, [Online] http://wareseeker.com/Network-Tools/network-traffic-generator-0.1.3.zip/335500
[36] D-ITG, [Online] http://www.grid.unina.it/software/ITG
[37] Portfolio Auswertung, [Dokument] Portfolio Analyse_Traffic Generator.xlsx
[38] Verteilte Systeme, [Online] http://de.wikipedia.org/wiki/Verteilte_Systeme
[39] Cisco Feature Navigator, [Online] http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp

[40] Cisco Systems, [Dokument] Cisco_command_reference_book.pdf
[41] Cisco Systems, [Online] http://www.oidview.com/mibs/9/CISCO-NETFLOW-MIB.html
[42] Microsoft Ressource Kit, [Online] http://download.microsoft.com/download/win2000platform/soon/1.00.0.1/nt5/en-us/soon_setup.exe

[43] GNS3, [Online] http://www.gns3.net
[44] PTB, [Online] http://www.ptb.de/de/org/4/44/442/dcf77_weite.htm
[45] METAS, [Online] http://www.metas.ch
[46] Bundesverwaltung, [Online] http://www.news.admin.ch/message/?lang=de&msg-id=28671
[47] GPS, [Online] http://de.wikipedia.org/wiki/Global_Positioning_System
[48] GOES, [Online] http://de.wikipedia.org/wiki/Geostationary_Operational_Environmental_Satellite
[49] NOAA, [Online] http://www.goes.noaa.gov
[50] LORAN, [Online] http://de.wikipedia.org/wiki/LORAN
[51] NTP, [Online] http://de.wikipedia.org/wiki/Network_Time_Protocol
[52] Cisco Systems, [Dokument] Netflow Services Solution Guide.pdf, Seite 4
[53] WinDump, [Online] http://www.mirrorservice.org/sites/ftp.wiretapped.net/pub/security/packet-capture/winpcap/windump/default.htm
[54] PCAP-Library, [Online] http://www.mirrorservice.org/sites/ftp.wiretapped.net/pub/security/packet-capture/winpcap/
[55] Cisco Systems, [Dokument] mls.pdf, Seite 2
[56] CACE Technologies, [Online] http://www.cacetech.com/products/turbocap.html
[57] Guide: Unbind NetBIOS ports, [Online] http://www.alkasis.com/?action=netbios