Fachhochschule Salzburg University of Applied Sciences

BACHELORARBEIT 1

Evaluierung von Realtime Ethernet Protokollen

durchgeführt am Studiengang

Informationstechnik und System-Management

Fachhochschule Salzburg GmbH

vorgelegt von

Christopher Wieland

Lisa Steiner

Studiengangsleiter: FH-Prof. DI Dr. Gerhard Jöchtl

Betreuer/Betreuerin: DI (FH) Thomas Pfeiffenberger

Puch/Salzburg, 06.10.2015

Eidesstattliche Erklärung

Ich/Wir versichere(n) an Eides statt, dass ich/wir die vorliegende Bachelorarbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und alle aus ungedruckten Quellen, gedruckter Literatur oder aus dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte gemäß den Richtlinien wissenschaftlicher Arbeiten zitiert, bzw. mit genauer Quellenangabe kenntlich gemacht habe(n). Diese Arbeit wurde in gleicher oder ähnlicher Form weder im In- noch im Ausland in irgendeiner Form als Prüfungsarbeit vorgelegt und stimmt mit der durch die Begutachter/Begutachterinnen beurteilten Arbeit überein.

Puch Urstein, 13.02.2016	1210555053	
Ort, Datum	Personenkennzeichen	Unterschrift des/der Studierenden

Puch Urstein, 13.02.2016	1310555080	
Ort, Datum	Personenkennzeichen	Unterschrift des/der Studierenden

Kurzzusammenfassung

Industrielle Netzwerke werden weltweit in Industrien verwendet, um eine vor allem schnelle und gleichzeitig stabile Verbindung zwischen Maschinen und Computern herzustellen. Da es in Industrieumgebungen zu erhöhten, beziehungsweise niedrigen Temperaturen, sowie erhöhter Luftfeuchtigkeit oder Staubbelastung kommen kann, wurden eigens dafür sogenannte Industrial Ethernet Switches entwickelt, welche dort zum Einsatz kommen. Dazu können verschiedene Realtime Ethernet Protokolle verwendet werden. In dieser Arbeit werden Protokolle auf Stabilität und Schnelligkeit für diesen Bereich untersucht. Realisiert wird eine Testinstallation mit drei Computern und drei Industrial Ethernet Switches. Die dazu getesteten Protokolle sind: das *Media Redundancy Protokoll* (MRP), das *Parallel Redundancy Protokoll* (PRP) und das *High Availability Seamless Redundancy Protokoll* (HSR). Durch die Testumgebung werden die Protokolle auf ihre Paketübertragungszeit, Paketverlustrate und auf ihren maximalen Jitter getestet. Die Ergebnisse der Messungen haben gezeigt, dass sich das HSR Protokoll bezüglich industrieller Anforderungen am geeignetsten erweist.

Abstract

Industrial networks are used in industries worldwide to get a stable and fast connection between computers and machines. In industries dust pollution, humidity and high or low temperatures can occur. That is the reason why special developed industrial switches are used. For this purpose, various real time Ethernet protocols can be used. In this paper protocols are tested for stability and speed. A test installation containing three computers and three industrial Ethernet switches is realized. The tested protocols are: *Media Redundancy Protocol* (MRP), *Parallel Redundancy Protocol* (PRP) and *High Availability Seamless Redundancy Protocol* (HSR). With the testing environment these protocols are tested concerning packet transmission time, packet loss rate and maximal jitter. The results have shown that the HSR protocol meets all requirements concerning industrial environments.

Inhaltsverzeichnis

Abkürzungsverzeichnis

PRP	Parallel Redundancy Protocol
HSR	High Availability Seamless Redundancy Protocol
PTP	Precision Time Protocol
LACP	Link Aggregation Control Protocol
NTP	Network Time Protocol
SNTP	Simple Network Time Protocol
MRM	Media Redundancy Manager
MRC	Media Redundancy Client
SAN	Single Attached Nodes
DANP	Double Attached Nodes for PRP
MAC	Media Access Control
RCT	Redundancy Control Trailer
DANH	Double Attached Nodes for HSR
OS	Operating System
PC	Personal Computer
NTP	Network Time Protocol

Abbildungsverzeichnis

Code-Snippet-Verzeichnis

Tabellen-Verzeichnis

1 Einleitung

Netzwerke sind im alltäglichen Leben nicht mehr wegzudenken, wobei diese nicht nur privat, sondern auch in vielen Industrien, in welchen besondere Bedingungen herrschen, genutzt werden. Kommerzielle Switches können unter industriellen Umgebungen nicht ordnungsgemäß arbeiten. Damit unter diesen Konditionen gearbeitet werden kann, muss eine speziell darauf ausgelegte Hardware, sogenannte Industrial Switches, verwendet werden.

Ziel dieser Untersuchung ist es, ein Industrienetzwerk zu simulieren und die dafür entsprechenden Redundanz-Protokolle zu konfigurieren, um eine stabile Leitung, welche bestenfalls geringste Verzögerung aufweist, zu erzeugen.

Realisiert wird dies mit Hilfe von aktuellen Realtime Ethernet Protokollen. Diese Protokolle werden für die simulierte industrielle Umgebung konfiguriert und getestet. Die Ergebnisse werden anschließend verglichen und es wird eine Schlussfolgerung der Messungen gezogen.

2 Theoretischer Teil

In diesem Teil der Arbeit wird die Funktionalität der verwendeten Protokolle erläutert. Die Protokolle operieren alle auf Layer 2 des Open Systems Interconnection (OSI) Modells.

Diese Sicherungsschicht, auch Data Link Layer oder Layer 2 genannt, ist dafür zuständig, Daten zwischen Geräten eines Netzwerkes oder auch verschiedenen Netzwerken zu übertragen. Um das Senden und Empfangen von Datenpaketen zu ermöglichen, ist es die Aufgabe dieser Schicht, eine Verbindung zwischen den Knoten eines Netzwerkes herzustellen [1].

Die Protokolle *Media Redundancy Protocol* (MRP), *High Availability Redundancy Seamless Redundancy Protocol* (HSR), sowie das *Parallel Redundancy Protocol* (PRP) werden miteinander hinsichtlich der Paketübertragungsrate, Paketverlustrate und des maximalen Jitters verglichen. Zur Zeitsynchronisation wird ein *Precision Time Protocol* (PTP) angestrebt.

Mit Hilfe der folgenden Abbildung wird ein Überblick über die Veröffentlichung der getesteten Protokolle verschafft.

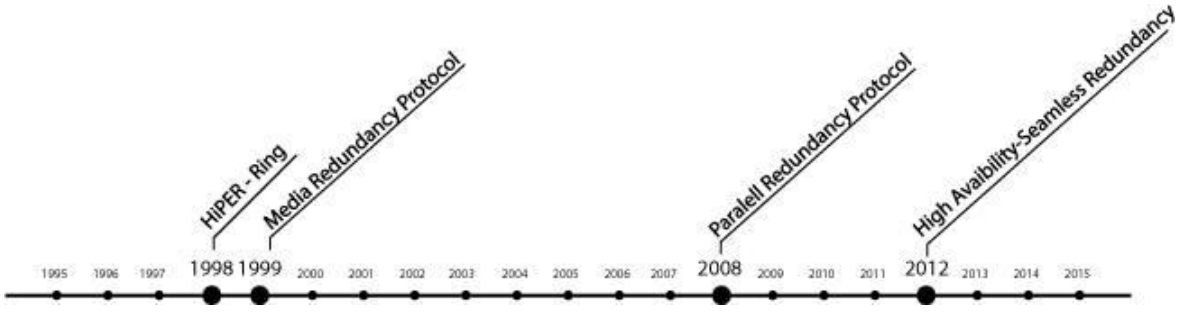

Abbildung 1: Zeitspanne der Protokolle

2.1 HiPER- Ring Protocol

Der folgende Text wurde erarbeitet mit der Referenz [2].

Das *High Performance Redundancy HiPER-Ring Protocol* wurde von den Firmen Hirschmann und Siemens in Kooperation entwickelt. In normalen Backbone Netzen wird in Linien-Struktur eine Verbindung zwischen einer Anzahl von N Switches aufgebaut (siehe Abbildung 1).

Wenn eine Verbindung ausfällt, sind alle darauffolgenden Switches auch vom Netz getrennt. Um das zu verhindern, wird eine Redundanzverbindung zwischen dem ersten und dem letzten Switch hergestellt (siehe Abbildung 2). In dieser redundanten Verbindung werden, sofern es zu keinem Fehler im Netz kommt, keine Daten übertragen, sondern ausschließlich Watchdog Pakete versendet.

Watchdog Pakete werden von allen dazugehörigen Switches übertragen, um ständig zu überprüfen, ob das Netz noch intakt ist oder nicht. Der Switch, welcher mit der redundanten Verbindung angeschlossen ist, wird als Redundanz Manager bezeichnet.

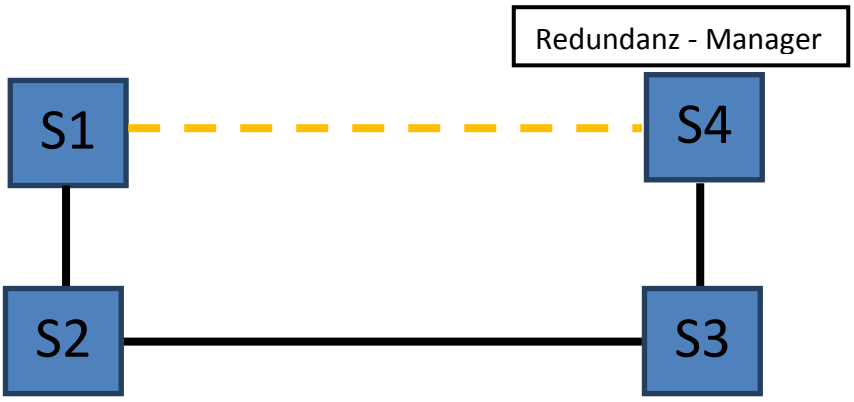

Abbildung 2: Herkömmlicher Backbone Ring

Mit dieser Methode kann die Datenübertragung weiterhin stattfinden, bis das Problem behoben wurde.

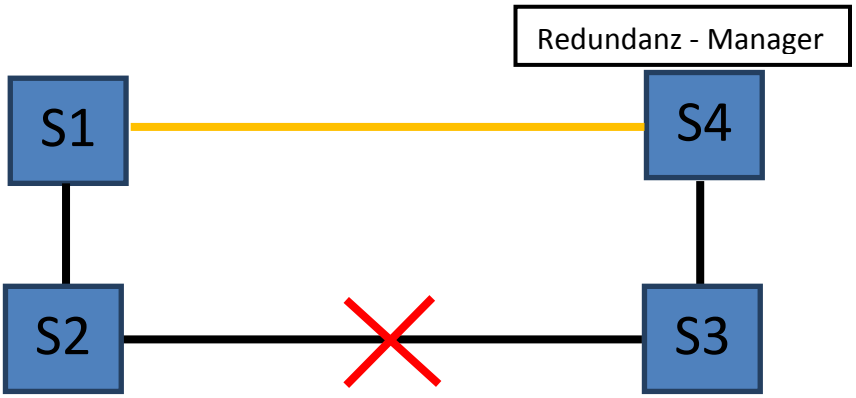

Abbildung 3: HiPER-Ring Verbindung

Dieses „Self-Healing“, das Reagieren auf die defekte Leitung bis hin zur Übermittlung über die redundante Leitung, dauert bis zu 300 ms.

2.2 Media Redundancy Protocol

Der folgende Text wurde erarbeitet mit der Referenz [3].

MRP arbeitet auf Layer 2 und ist eine direkte Abwandlung, beziehungsweise Erweiterung des HiPER-Rings, welche von Hirschmann weiterentwickelt wurde. Hier handelt es sich um ein im IEC Standard 62439-2 beschriebenes *Media Redundancy Protocol*.

Wenn das MRP zum Einsatz kommt, gibt es einen Media Redundancy Manager (MRM) und die Clients (MRC). In Abbildung 4 wird ein solcher Ring dargestellt.

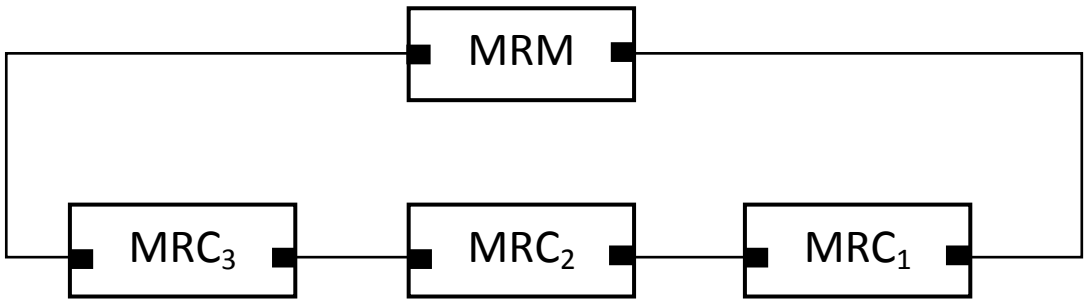

Abbildung 4: Media Redundancy Protocol geschlossen [3]

Die Switches unterstützen 3 Modi: „disabled“, „blocked“ und „fowarding“. Beim Status „fowarding“ werden alle Pakete weitergeleitet, bei „blocked“ wird ausschließlich die Kommunikation des MRM weiter übertragen und bei „disabled“ wird überhaupt keine Übertragung stattfinden. Im Normalfall arbeitet der Ring im geschlossenen Zustand. Alle Ports der MRC Switches sind auf „fowarding“ gestellt.

Beim MRM Switch sind die Status der Ports auf „fowarding“ und auf „closed“, damit eine Schleife verhindert werden kann. Der Media Redundancy Manager sendet außerdem Test-Frames durch das Netz, um sicher zu gehen, dass alles in Ordnung ist.

Wenn es beispielsweise zu einem Verbindungsabbruch kommt, dann senden die MRCs ein „LinkChange Frame“ an den MRM, damit der Media Redundancy Manager den auf „blocked“ gesetzten Port auf „fowarding“ umschaltet und der MRM sendet ein „TopoChange Frame“ an die entsprechenden MRCs, um die betroffenen Ports von „fowarding“ auf „blocked“ zu setzen. So kann es zu keiner Unterbrechung der Verbindung kommen. In Abbildung 5 wird ein solches Szenario veranschaulicht.

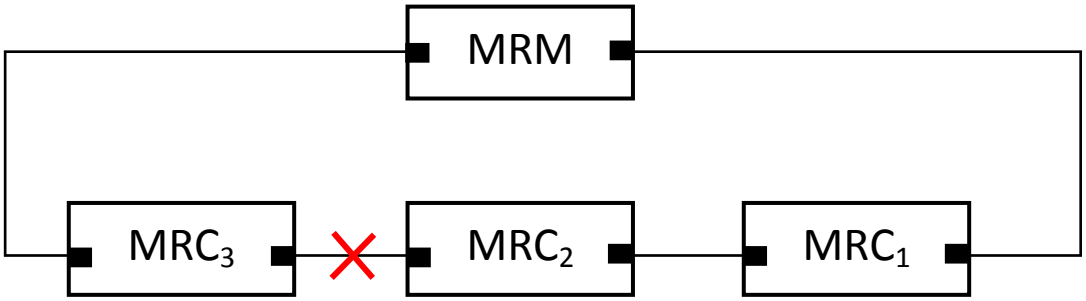

Abbildung 5: Media Redundancy Protocol offen [3]

Die Ports, zwischen denen der Verbindungsabbruch signalisiert wurde, schalten die anliegenden Ports auf „blocked“. Dieses „Switch-Over“ kann zwischen 200 ms und 500 ms andauern. Während diesem „Switch-Over“ kann es zu Verlusten von Frames kommen.

2.3 Parallel Redundancy Protocol

Beim *Parallel Redundancy Protocol* handelt es sich um ein Realtime Ethernet Protokoll, welches von der IEC SC65C WG15 „High Available Automation Networks“ Arbeitsgemeinschaft als eine Redundanzmethode genannt wird [4]. Das im IEC 62439-3 Abschnitt 4 beschriebene Protokoll zählt mit dem *High Availability Seamless Redundancy Protocol* zu den einzigen Protokollen, welche keine Erholzeit benötigen. Damit ist die Wiederherstellung eines Netzwerkes aufgrund einer Unterbrechung oder dem Ausfall einer Netzwerkkomponente gemeint [5].

Das *Parallel Redundancy Protocol* wurde als Layer 2 Ethernet Protokoll eingeführt [6]. Bei Ausfällen ist es in der Lage, diese ohne Unterbrechungen und ohne Umschalten zu bewältigen. Verglichen mit anderen Protokollen, welche ausschließlich Ringstrukturen verwenden, kann dieses Protokoll mit Hilfe von zwei parallelen Netzwerken mehr Ausfallszenarien bewältigen und bietet somit eine hohe Ausfallssicherheit. Diese Störungssicherheit bezieht sich sowohl auf Unterbrechungen des Netzwerkes selbst als auch auf Ausfälle von Komponenten wie beispielsweise Switches oder Netzwerkkarten [7].

Das Konzept dieses Protokolls basiert auf zwei voneinander unabhängigen Netzwerken, welche parallel betrieben werden, aber keine Lastteilung vornehmen [7]. Die gewählten Netze müssen nicht dieselbe Topologie aufweisen, was bedeutet, dass etwa Ringstrukturen aber auch Netzwerke ohne Redundanz verwendet werden können. Es ist jedoch von Vorteil, aufgrund der abweichenden Übertragungszeiten in unterschiedlichen Netzwerken, ähnliche Strukturen zu verwenden [6].

Das *Parallel Redundancy Protocol* wird vor allem in kritischen Endgeräten implementiert, nicht unbedingt aber in den Switches. Die Geräte, welche über eine derartige Funktionalität verfügen, werden auch Double Attached Nodes for PRP (DANP) genannt. Beim Senden werden von diesen DANs zwei unabhängige Netzschnittstellen verwendet, welche nicht nur dieselben Daten zeitgleich in die verschiedenen Netzwerke schicken, sondern auch dieselbe MAC Adresse verwenden [8]. Werden Daten empfangen, geschieht dies ebenfalls über beide Schnittstellen. Aufgrund dieser Vorgehensweise muss ein DAN in der Lage sein, eines der beiden Datenpakete, insofern beide das Ziel erreichen, wieder zu verwerfen [6].

Jedoch gibt es auch Standardgeräte, welche nicht über dieses Protokoll verfügen und nur mit einer Netzschnittstelle ausgestattet sind, auch Single Attached Nodes (SAN) genannt. Für SANs ist lediglich eine Verbindung zu einem der beiden Netzwerke möglich, was bedeutet, dass sie auch nur mit Geräten im selben Netzwerk kommunizieren können [7]. Damit diese Standardgeräte jedoch die Möglichkeit bekommen PRP zu verwenden, gibt es sogenannte Redundancy-Boxen (RedBoxes). Diese RedBoxes, welche das *Parallel Redundancy Protokoll* implementiert haben, können als eine Art Redundanz Proxy dienen, um den SANs eine Verbindung zu beiden Netzen zu verschaffen [8].

In der folgenden Abbildung wird ein Netzwerk mit *Parallel Redundancy Protocol* dargestellt, um die bereits erwähnten Komponenten veranschaulichen zu können.

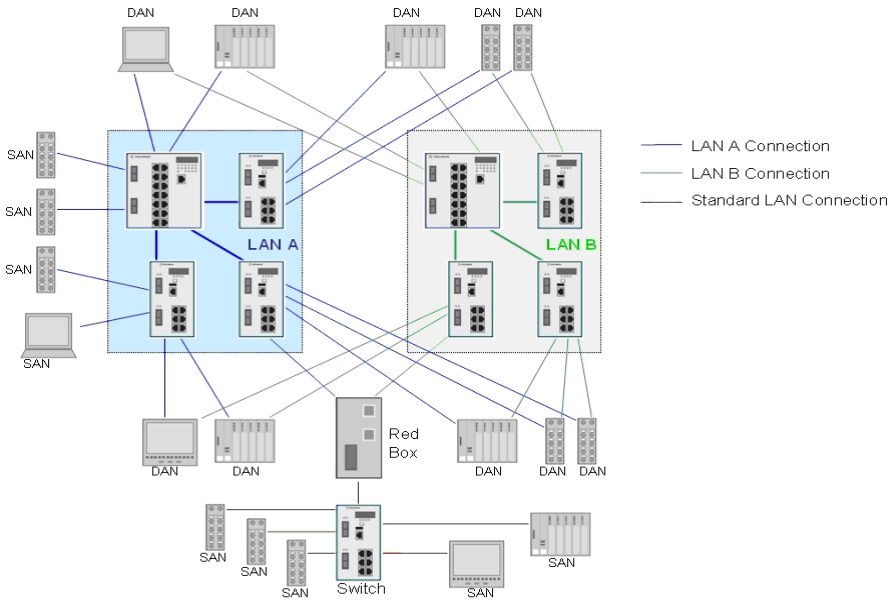

Abbildung 6: Netzwerk mit Parallel Redundancy Protocol [8]

Um erhaltene Duplikate erkennen zu können ist ein Redundancy Control Trailer (RCT) notwendig. Dieser RCT setzt sich aus der Zählnummer, der LAN Bezeichnung, der Größe des Frames und einer bestimmten PRP Endung zusammen. Diese PRP Endung ist für eine ordnungsgemäße Identifizierung notwendig. Jedes Framepaar erhält eine bestimmte Zählnummer, wobei beide Frames dieselbe erhalten. Unter der Voraussetzung, dass mindestens eines der beiden Netze funktioniert, wird das Ziel von einem der beiden Frames erreicht. Jener, der

aufgrund geringerer Verzögerungszeiten zuerst den Empfänger erreicht, wird anhand seiner Zählnummer identifiziert. Kommt es aufgrund der Funktionalität von beiden Netzen zur Ankunft von beiden Frames, wird der zuerst erhaltene angenommen und der darauf folgende verworfen [6].

Die Stärken dieses Protokolls liegen besonders in der Hochverfügbarkeit, welche durch das System zweier unabhängiger Netzwerke garantiert wird. Darüber hinaus bietet die Flexibilität im Netzaufbau unzählige Möglichkeiten um verschiedene Netzwerke realisieren zu können [8]. Andererseits stellt das Vorhandensein zweier Netzwerke auch einen großen Aufwand an Hardware und Rechenintensivität dar [7].

2.4 High Availability Redundancy Seamless Protocol

Das *High Availability Seamless Redundancy Protocol*, welches auch als Weiterentwicklung des *Parallel Redundancy Protocol* gesehen wird [9], stellt ein Ethernet (IEEE 802.3) Redundanzprotokoll dar, welches im Standard IEC 62439 – 3 Abschnitt 5 definiert und beschrieben wird [10]. Verglichen mit dem PRP handelt es sich dabei jedoch um ein Protokoll zur Herstellung von Medienredundanz, wobei PRP Netzwerkredundanz erzeugt [9]. Bei Medienredundanz handelt es sich um mehrfach vorhandene Information, bei Netzwerkredundanz um mehrfach vorhandene Netzwerkkomponenten oder auch Netzwerke [11].

Das *High Availability Seamless Redundancy Protocol* wird hauptsächlich durch Ringnetzwerke realisiert. Dabei ist es nicht zwingend notwendig Switches in die Topologie mit einzubinden, denn es können auch lineare Topologien verwendet werden [10].

In einer HSR Ringstruktur gibt es vier Arten von Knoten. Wie bereits beim PRP handelt es sich bei diesen Knoten um Double Attached Nodes for HSR (DANH), Single Attached Nodes (SAN), Redundancy-Boxen und zusätzlich werden sogenannte Quadboxes eingesetzt [12], bei welchen es sich prinzipiell um zwei Redboxes handelt, welche dazu in der Lage sind, eine Verbindung zu einem weiteren Ringnetzwerk herstellen zu können [10].

Die folgende Abbildung stellt ein einfaches Ringnetzwerk dar, wobei Hirschmann Managed RSP Switches, welche die beiden Redundanzprotokolle HSR und PRP

unterstützen, als RedBoxes dienen, um SANs in das Netzwerk einbauen zu können [13].

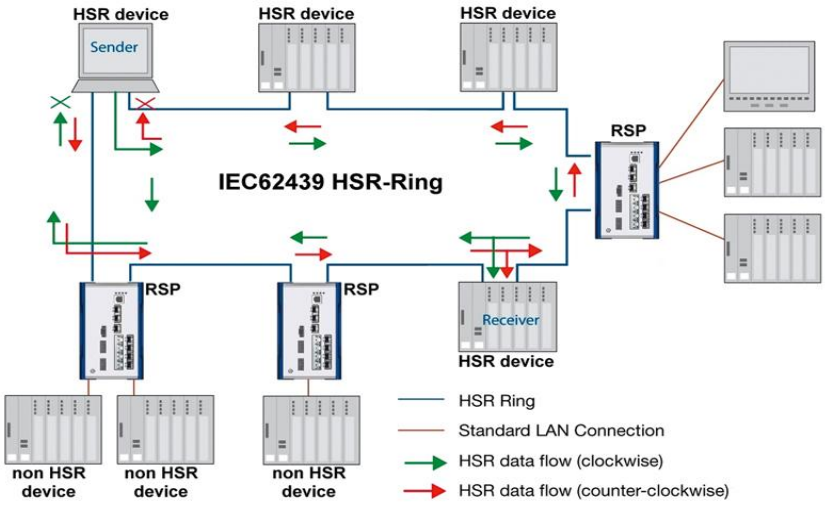

Abbildung 7: HSR Ringnetzwerk [9]

Beim HSR Protokoll ist es nicht möglich, Standardgeräte (SANs), welche nur über eine Netzwerkschnittstelle verfügen, direkt in ein Netzwerk mit einzubinden, weshalb Redundancy-Boxen in derartigen Fällen unbedingt notwendig sind. Der Grund dafür befindet sich in der Struktur des Frames, welcher, bevor er versendet wird, mit einer HSR Markierung versehen wird.

Dieser „Tag“ befindet sich beim High *Availability Seamless Redundancy Protocol* nicht am Ende des Frames, wie es beim PRP der Fall ist, sondern direkt am Beginn, weshalb der Protokollverkehr somit für SANs unkenntlich gemacht wird. Beim PRP hingegen wird dieser Tag einfach als Padding, also zusätzlich vorhandene Füllbits, interpretiert [9].

Zusätzlich befinden sich in diesem „HSR Tag“ noch die Länger der Nutzlast, der Sendeport und die Sequenznummer des Frames [9]. Aufgrund der Position dieses „Tags“ können und müssen alle Geräte (Knoten), welche ein Frame erhalten, unmittelbar nach Erhalt dessen, eine Duplikats Erkennung durchführen [14].

Jeder sich in diesem Ring befindende Knoten besitzt zwei Netzwerkports, welche beide dieselbe MAC Adresse und dieselbe IP Adresse verwenden [10]. Soll ein

Frame versendet werden, so passiert dies stets über beide Ports, wobei eine Kopie im Uhrzeigersinn versendet, und die andere Kopie gegen den Uhrzeigersinn versendet wird [15].

Beim Multicast Prinzip wird ein Frame von jedem Knoten, jeweils in beide Richtungen, weitergeleitet, unter der Voraussetzung, dass er zuvor von diesem Knoten noch nicht weitergleitet wurde. Dabei werden alle Knoten als Empfänger angesehen und jeder Knoten behält jeweils einen Frame und verwirft das Duplikat. Erreicht der Frame nach einer Umrundung aller Knoten die ursprüngliche Quelle wieder, so ist dieser Knoten für die Verwerfung von diesem zuständig [10], um permanente Kreise zu verhindern [9].

Dabei kann erkannt werden, ob eventuelle Unterbrechungen vorliegen, denn erreicht jeweils nur einer der beiden Frames das Ziel und die Duplikate bleiben aus, so lässt sich das auf einen Fehler im Netzwerk zurückschließen [10].

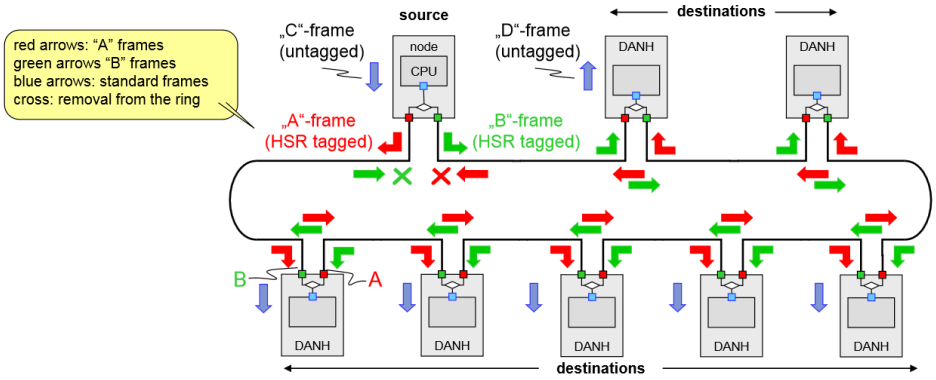

Abbildung 8: HSR Multicast Prinzip [10]

Beim Unicast Prinzip hingegen werden Frames nur dann von Knoten aufgenommen, wenn sie an diese adressiert sind [9]. Dabei gibt es einen bestimmten Empfänger, welche den Frame anschließend tatsächlich an die Anwendung weiterleitet, wobei dies beim Multicast Prinzip jeder Knoten durchführt [10].

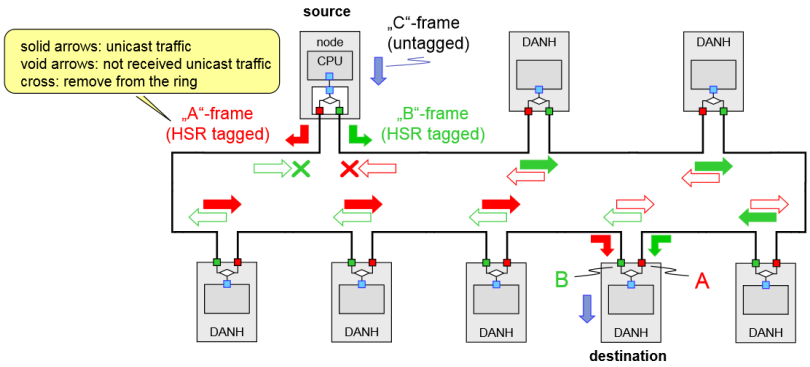

Abbildung 9: HSR Unicast Prinzip [10]

Das High *Availability Seamless Redundancy Protocol* bietet Vorteile bezüglich der Verwendung, da es beispielsweise für jedes Industrial Ethernet verwendet werden kann und zusätzlich keine Duplikation vom Netzwerk, wie es beispielsweise beim PRP der Fall ist, benötigt [10]. Jedoch erhält im fehlerlosen Fall jeder Empfängerknoten zwei identische Frames [14], was einen erheblichen Nachteil bezüglich der Performanz darstellt [12].

2.5 Link Aggregation Control Protocol

Der folgende Text wurde erarbeitet mit der Referenz [16].

Bei dem *Link Aggregation Control Protocol* (LACP) nach IEEE 802.1AX-2008 handelt es sich um ein Protokoll, welches physische Netzwerkverbindungen dynamisch miteinander bündelt, um eine höhere Bandbreite zu erreichen.

Somit sind mindestens zwei Verbindungen, auch Trunks genannt, zu einer logischen Verbindung gebündelt. Die Lasten der Datenübertragung werden in diesem Fall automatisch verteilt, sodass es zu gleichermaßen ausgelasteten Leitungen kommt. Falls nun eine Verbindung abbricht, ist die Redundanz durch die andere Verbindung gegeben.

In der folgenden Abbildung kann man ein Beispiel eines Aufbaus sehen, welches mit LACP konfiguriert wurde.

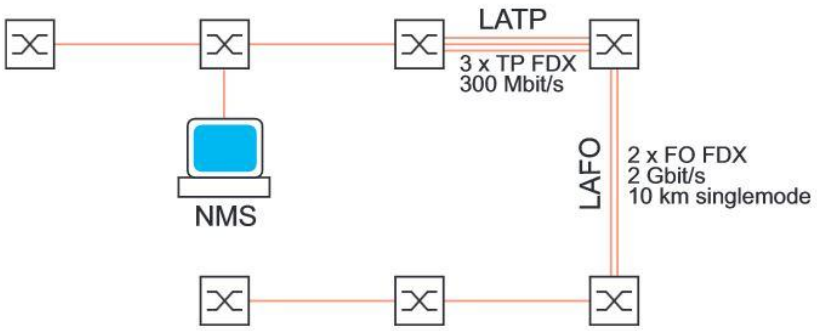

Abbildung 10: Link Aggregation [16]

Es können maximal acht Trunk Verbindungen miteinander gebündelt werden, wobei eine Verbindung von vier optimal wäre. Ein Vorteil hier ist auch, dass die Verbindungen mit Fiber Kabeln und Twisted Pair Kabeln gemixt werden können.

2.6 Zeitsynchronisation

Zur Zeitsynchronisation wird das *Precision Time Protocol* (PTP) angestrebt, welches in IEEE 1588 und in IEC 61588 beschrieben ist. Mit dem PTP ist es möglich, Geräte mit Hilfe von Ethernet Schnittstellen auf Nanosekunden genau zu synchronisieren [17]. Viele Geräte kommunizieren miteinander. Sind deren lokale Uhren nicht aufeinander abgestimmt, kann es zu unnötigen Wartezeiten oder verfälschten Messungen kommen [18].

Es gibt zwei Möglichkeiten, die lokalen Uhren der Geräte aufeinander abstimmen zu lassen. Man kann eine sogenannte „Offset-Correction“ durchführen, wobei alle Uhren auf die präziseste Uhr, den „Grandmaster“, eingestellt werden. Es ist aber ebenso möglich, eine „Drift-Correction“ durchzuführen. Hier werden die Zeigergeschwindigkeiten der Uhren angepasst, um dieselbe Zeit an allen Uhren zu erzielen [18].

Die am meisten verbreiteten Methoden, um Zeiten korrekt einzustellen, sind derzeit das *Network Time Protocol* (NTP) und das daraus abgeleitete *Simple Network Time Protocol* (SNTP), obwohl das PTP genauer ist [18].

Bei dem *Precision Time Protocol* wird zwischen „Master“ und „Slave“ unterschieden. Der „Master“ wird von einer Radio-Uhr oder von einem GPS Signal gesteuert und der „Master“ synchronisiert anschließend seine Zeit mit jener der „Slaves“ [18].

In dem folgenden Abbild kann man ein Beispiel eines solchen Aufbaus sehen.

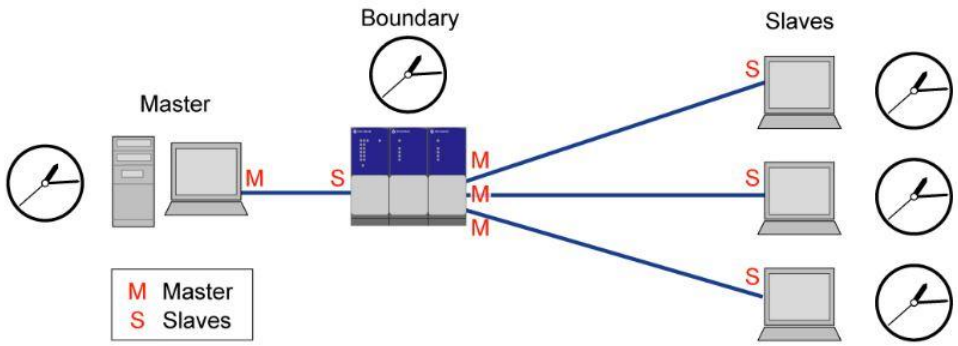

Abbildung 11: PTP Master Slave Prinzip [18]

Die Synchronisation verläuft in zwei Schritten. Im ersten Schritt wird den „Slaves“ die Uhrzeit des „Masters“ mit einer Synchronisationsnachricht, der „SYNC message“, mitgeteilt. Vom Zeitpunkt des Absendens bis zum Empfang der „SYNC message“ wird die Zeit gemessen.

Diese gemessene Zeit wird in einer „follow-up message“ nachgeschickt, um die Uhrzeit noch präziser einstellen zu können. Sind diese zwei Schritte erledigt, wurden die Uhrzeiten richtig miteinander synchronisiert [18].

3 Praktischer Teil

Im praktischen Teil der Arbeit werden die Eigenschaften der Protokolle HSR, PRP und MRP untersucht. Für die Zeitsynchronisation kommt das *Network Time Protocol* (NTP) zum Einsatz, welches an den Testrechnern konfiguriert wurde. Es wurden jeweils drei Messungen pro Protokoll durchgeführt. Gemessen wurden die Paketübertragungsdauer unter verschiedenen Lasten, daraus entstehende Verzögerungen und die Paketverluste bei simuliertem Ausfall einer Switch-Anbindung.

3.1 Vorbereitungen

Bevor der praktische Teil der Arbeit realisiert werden konnte, musste zuerst die passende Hardware gefunden und ausgewählt werden. Dabei wurden mehrere Switches verschiedener Hersteller in Betracht gezogen, anhand der bereitgestellten Informationen verglichen und schlussendlich die Auswahl getroffen. Dabei war es am wichtigsten, auf die Redundanzmethoden der Switches zu achten, da die Protokolle nur von speziellen, für Industrien vorgesehenen Switches, unterstützt werden. Zusätzlich mussten die Switches über Gigabit Ports verfügen, welche für die Realisierung der jeweiligen Netzwerktopologien nötig waren.

Aufgrund der Funktionalitäten, welche den entsprechenden Anforderungen entsprachen, fiel die Entscheidung auf die RSP25 Switches der Firma Hirschmann, welche auch für die Dauer des Projektes von der Firma bereitgestellt wurden.

3.2 Verwendete Materialien

Für sämtliche Messungen kommen drei Switches und drei PCs zum Einsatz. Bei den Switches handelt es sich um RSP25 Industrial Ethernet Switches der Firma Hirschmann, damit der Aufbau eines Netzwerkes, welches einem Industrienetzwerk möglichst ähnlich sein soll, realisiert werden kann. Die Switches werden jeweils an ein Stromversorgungsgerät mit 39 Volt und 0.4 Ampere angeschlossen.

Auf den drei PCs wird das Linux Betriebssystem Ubuntu 15.10 installiert, welches notwendig ist, um die Messungen mit Hilfe des „PathEval Tools“ durchführen zu können. Die drei PCs, welche ebenfalls im Netzwerk angeschlossen sind, werden dazu benutzt, um simulierten Datenverkehr zu senden und zu empfangen.

Für die Darstellung der Topologien wird das Programm „Dia Diagram Editor 0.97.2“ verwendet. Die Messungsergebnisse werden mit Hilfe von „Matlab Version 2013 r2“ grafisch veranschaulicht.

In den folgenden Tabellen werden die verwendeten Geräte detaillierter beschrieben.

Switches	
Bezeichnung	RSP25-11003Z6TT-SCCZ9HDE2A05.0.01
Anschlüsse	8x 100Mbit, 3x 1000MBit
Funktionen	PRP, HSR, MRP, Hiper Ring, LACP

Tabelle 1: Verwendete Switches

PC 1 und PC 3	
Modell	HP Compaq Elite 8000 Business
OS	Linux Ubuntu 15.10

Tabelle 2: Verwendete PCs 1 und 3

PC 2	
Modell	HP Compaq Elite 8200 Business
OS	Linux Ubuntu 15.10

Tabelle 3: Verwendeter PC 2

3.3 Konfiguration der Switches und PCs

Die Netzwerktopologien für die getesteten Protokolle bestehen aus drei Switches und drei PCs, wobei alle PCs mit zwei Netzwerken verbunden sind. Beim ersten Netzwerk handelt es sich um das „Management“ Netzwerk, wobei alle PCs an einem Hub angeschlossen sind. Dieses Netzwerk wird dazu verwendet, um die Konfigurationsdateien des zum Senden verwendeten Programms „PathEval“ an den Sender PC und Empfänger PC zu übertragen. Das zweite Netzwerk, „Switch Anbindung“, beinhaltet die eigentliche Topologie mit den drei Switches und wird für jedes getestete Protokoll dementsprechend angepasst. Die Konfigurationen der einzelnen Switches für jedes Protokoll sind im Anhang angefügt.

Netzwerk	PC 1	PC 2	PC 3
Management	192.168.1.1	192.168.1.2	192.168.1.3
Switch Anbindung	10.0.0.10	10.0.0.20	10.0.0.30

Tabelle 4: IP Adressen der PCs

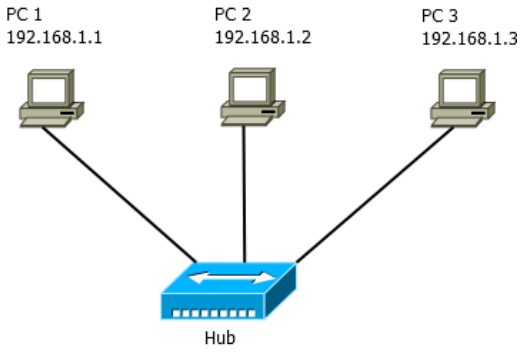

Abbildung 12: Management Netzwerk

Netzwerk	Switch 1	Switch 2	Switch 3
Switch Anbindung	10.0.0.1	10.0.0.2	10.0.0.3

Tabelle 5: IP Adressen der Switches

3.3.1 Konfiguration der Zeitsynchronisation

Zur Zeitsynchronisation wird das *Network Time Protocol* (NTP) verwendet, da es bei erstmaligen Messungsversuchen mit dem *Precision Time Protocol* (PTP) zu Problemen kam und keine Synchronisation der Zeit erreicht werden konnte. Aufgrund der Simplizität der Testumgebung, welche keine schwerwiegenden Störungen der Netzwerke verursachen sollte, wurde entschlossen, dass das *Network Time Protocol* für die Messungen eingesetzt wird.

Das *Network Time Protocol* ermöglicht eine Synchronisation auf wenige Millisekunden genau zwischen sogenannten „Clients“ und „Servern“. Meistens wird dieses Protokoll dazu verwendet, um Geräte, über das Internet, nach internationalen Zeitstandards zu konfigurieren, wobei sich die Clients in gewissen Zeitintervallen immer wieder nach der Zeit des angegebenen Servers konfigurieren [19].

In der verwendeten Testumgebung werden die PCs, welche als Sender und Empfänger der Testpakete dienen, als NTP-Clients konfiguriert. Der PC 3, welcher in diese Sendungen nicht involviert ist, agiert als NTP-Server für die NTP-Clients (PC 1 und PC 2). Somit ist es möglich, dass sich die Clients nach dem lokalen NTP-Server synchronisieren können und für die Messungen eine möglichst geringe Abweichung voneinander aufweisen.

3.3.2 Konfiguration der Link Aggregation

Zusätzlich sollte das *Link Aggregation Control Protocol* (LACP) dazu dienen, um zusätzliche Redundanz zur Verfügung zu stellen. Details bezüglich der Switches besagten, dass Link Aggregation in Verbindung mit dem MRP Protokoll möglich sei, bezüglich HSR und PRP gab es diesbezüglich jedoch zu wenige Informationen. Aufgrund der Idee, die Protokolle unter denselben Umständen zu testen, wurde beschlossen das *Link Aggregation Control Protocol* nicht für die Messungen zu verwenden.

3.3.3 Verwendung des PathEval Tools

Das Programm PathEval Tool ist ein Daemon Programm für Linux, welches, mit Hilfe von Python Skripts, dazu verwendet wird, einen TCP/UDP IP Verkehr

zwischen zwei PCs zu erzeugen. Auf den verwendeten PCs muss dazu das Programm installiert und als Dienst gestartet werden. Um die Messung durchführen zu können muss zuerst das Python Skript des Empfängers und danach das Python Skript des Senders ausgeführt werden. Die verwendeten Python Skripts definieren die benötigten Parameter IP Destination, IP Source und die Paketgröße des Datenverkehrs, welche für die simulierte Übertragung benötigt werden.

3.4 Beschreibung des Media Redundancy Protocols

Die folgende Abbildung stellt die Topologie des MRP Netzwerkes dar. Dabei dient der Switch 1 als Redundanzmanager, was bedeutet, dass dieser Switch dazu zuständig ist, im Falle einer Unterbrechung der standardmäßigen Route, die Route auf den Redundanzpfad umzulenken. Es wird eine Messung simuliert, in welcher PC 1 an PC 2 sendet. Die reguläre Route führt von PC 1 über Switch 1, über Switch 3 und über Switch 2 an PC 2.

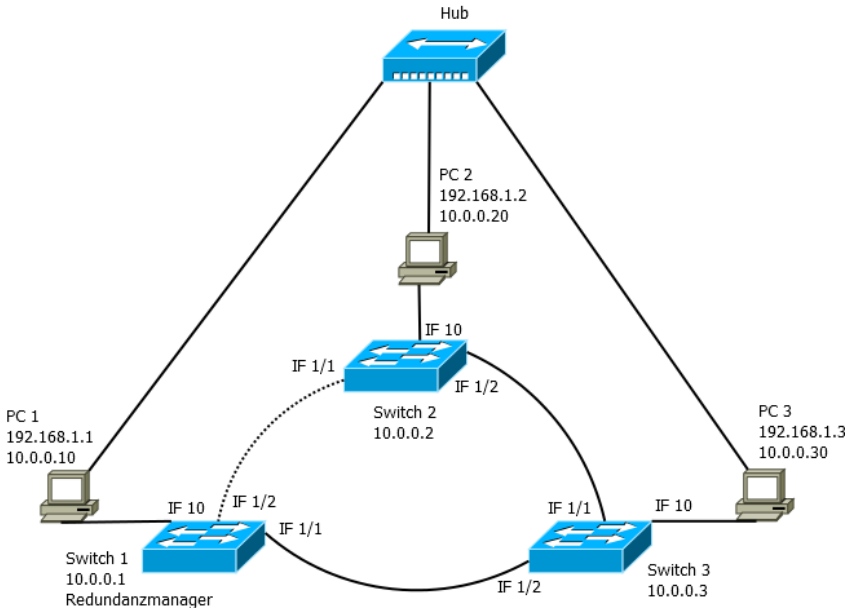

Abbildung 13: MRP Topologie

Um die Topologie für das *Media Redundancy Protocol* zu erstellen, werden bei allen Switches jeweils die zwei Gigabit Interfaces (1/1 und 1/2) verwendet.

3.5 Beschreibung des Parallel Redundancy Protocols

Für die Messungen des *Parallel Redundancy Protokolls* werden zwei Netzwerke benötigt, wobei der Sender (PC 1) gleichzeitig den Datenverkehr über die beiden Netzwerke zum Empfänger (PC 2) sendet. Die Switches 1 und 2 werden hier als Redboxes implementiert und der Switch 3 wird als Dummy-Switch ohne Konfiguration eines Protokolls mit dem Netzwerk verbunden. PC 1 und PC 2 werden jeweils an eine RedBox angeschlossen, da sie mit beiden Netzwerken in Kontakt stehen müssen.

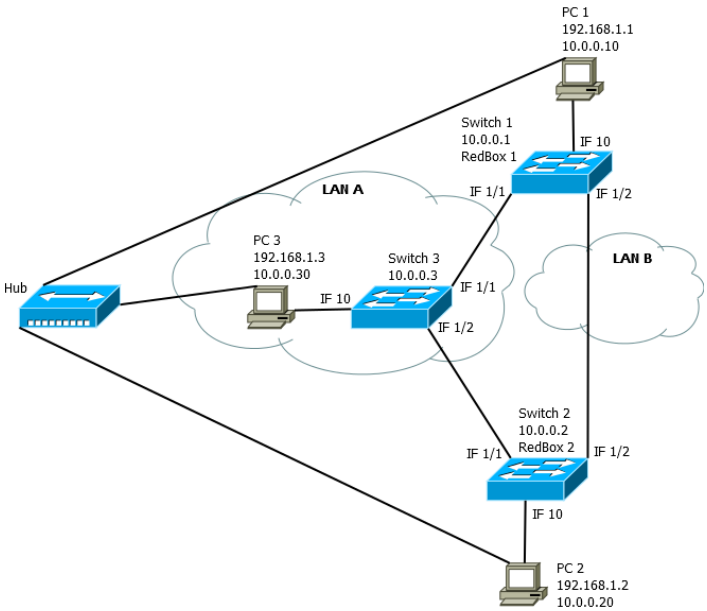

Abbildung 14: PRP Topologie

3.6 Beschreibung des High Availability Seamless Redundancy Protocols

Für die Topologie zur Messung des HSR Protokolls werden alle Switches als sogenannte RedBoxes implementiert. Das ist notwendig, um die PCs mit in das Netzwerk einzubinden, da diese über keine parallele Netzwerkschnittstelle verfügen.

Das HSR Protokoll wird ebenfalls als Ringtopologie aufgebaut, und die RedBoxes benutzen jeweils die Gigabit Interfaces 1/1 und 1/2. Die Messungen werden erneut so durchgeführt, dass PC 1 an PC 2 senden soll, wobei der Datenverkehr in beide Richtungen gesendet wird.

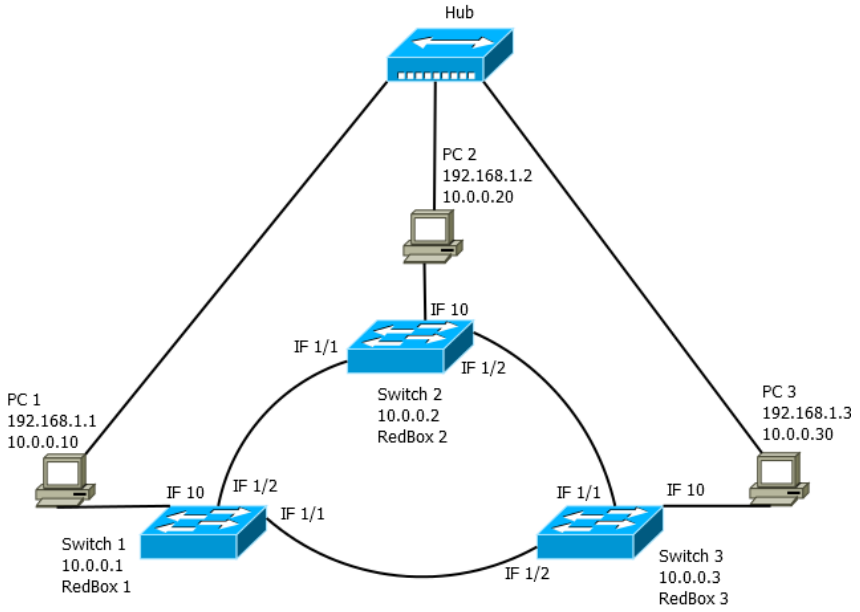

Abbildung 15: HSR Topologie

3.7 Methodik der Messungen

Die Messungen werden unter denselben Umständen durchgeführt. Die folgende Tabelle beschreibt die verschiedenen Messungsszenarien.

Messungs-szenarien	**Übertragungsdauer in Sekunden**	**Größe der zu übertragenden Pakete in Bytes**	**Zeitraum der Unterbrechung/ des Ausfalls pro Minute**
Szenario A	600	200	Sekunde 30 - 50
Szenario B	600	700	Sekunde 30 - 50
Szenario C	600	1200	Sekunde 30 - 50

Tabelle 6: Messungsszenarien

Wie in Tabelle 6 ersichtlich ist, werden pro Protokoll drei Messungen durchgeführt mit Szenario A: 200 Bytes/Paket, Szenario B: 700 Bytes/Paket und Szenario C: 1200 Bytes/Paket. Die Messungen betragen 10 Minuten, wobei bei jeder Messung 10-mal für jeweils 20 Sekunden eine Unterbrechung des Netzwerkes simuliert wird.

4 Ergebnisse

Wie bereits im praktischen Teil erwähnt wurde, herrschten für alle Messungen dieselben Bedingungen. Mit Hilfe der Messungsergebnisse wurden anschließend die Anzahl der übertragenen Pakete, die Paketverlustrate, der maximale Jitter, welcher die größte Abweichung der durchschnittlichen Übertragungszeit beschreibt, und die Dauer der einzelnen Paketsendungen analysiert. In den folgenden Abbildungen werden die einzelnen Szenarien detailliert veranschaulicht.

Für jedes Szenario wird eine Grafik mit derselben Skalierung dargestellt, um alle Messungen einheitlich vergleichen zu können. Für die Verbildlichung wurde ein Ausschnitt aus der zehnminütigen Messung, welcher zwei Unterbrechungen beinhaltet, gewählt.

4.1 Messung des Media Redundancy Protocols

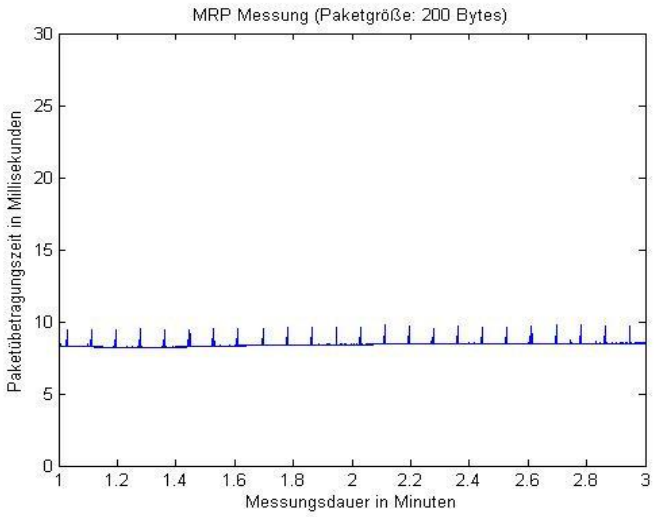

Abbildung 16: MRP Messung Szenario A

Die erste Messung, welche 200 Bytes große Pakete simulierte, ergab eine durchschnittliche Paketübertragungszeit von 8,73 ms. Dabei wurde ein maximaler Jitter von 5,48 ms erreicht.

Vergleicht man diese Übertragungszeit mit jener der Messung mit 700 Bytes großen Paketen, erkennt man, dass die 200 Bytes Pakete mehr Zeit zum Übertragen benötigten, als die größeren 700 Bytes Pakete. Da diese Messung als erste

durchgeführt wurde, wird vermutet, dass sich die Zeitsynchronisation mit NTP zu diesem Zeitpunkt noch nicht eingependelt hatte und daraus die längere Zeitdauer der Paketübertragung entstanden ist.

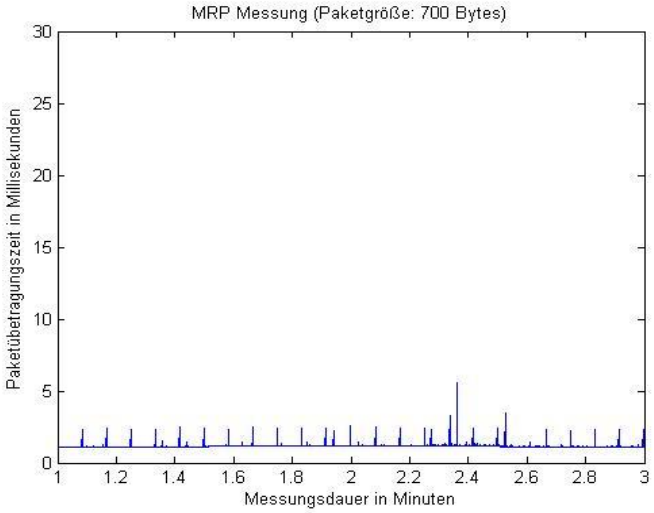

Abbildung 17: MRP Messung Szenario B

Wie in Abbildung 17 ersichtlich ist, ergab die Messung mit 700 Bytes großen Paketen eine Paketübertragungszeit von 1,22 ms. Der maximale Jitter erreichte einen Wert von 6,32 ms.

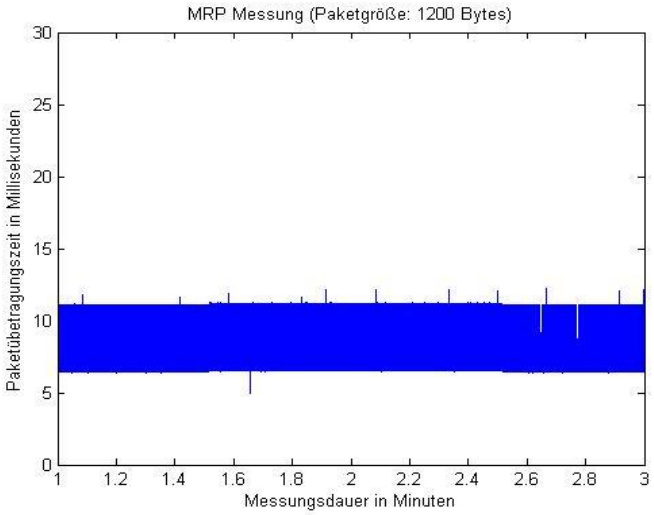

Abbildung 18: MRP Messung Szenario C

Die Messung mit 1200 Bytes Paketgröße stellte eine sichtliche Belastung für das MRP Protokoll dar. Die durchschnittliche Übertragungszeit der Pakete betrug 9,13 ms. Ein maximaler Jitter von 13,96 ms wurde festgestellt.

Paketgröße in Bytes	**Anzahl der übertragenen Pakete**	**Anzahl der verlorenen Pakete**
200	7223254	22664
700	7211926	21591
1200	5902426	17355

Tabelle 7: Anzahl der übertragenen und verlorenen Pakete mit MRP

Unterbrechungen	**Anzahl der verlorenen Pakete bei Szenario A**	**Anzahl der verlorenen Pakete bei Szenario B**	**Anzahl der verlorenen Pakete bei Szenario C**
1. Unterbrechung	2070	2188	1732
2. Unterbrechung	2098	2166	1440
3. Unterbrechung	2180	2073	1718
4. Unterbrechung	2166	2276	1818
5. Unterbrechung	2064	1734	1744
6. Unterbrechung	2277	2265	1715
7. Unterbrechung	2192	2240	1853
8. Unterbrechung	2259	2261	1797
9. Unterbrechung	2156	2098	1800
10. Unterbrechung	2240	2290	1787

Tabelle 8: Paketverlustraten der einzelnen Unterbrechungen mit MRP

Anhand der Tabelle 7 kann man die Gesamtanzahl der übertragenen und verlorenen Pakete der jeweiligen Messungen erkennen. Tabelle 8 beschreibt die genauen Paketverluste pro Unterbrechung, wobei die Messung mit 200 Bytes Paketen einen Verlustwert von 0,31 Prozent ergab. Die Messung mit 700 Bytes Paketen ergab 0,29 Prozent Verlust, und bei 1200 Bytes Paketen betrug die Verlustrate 0,29 Prozent.

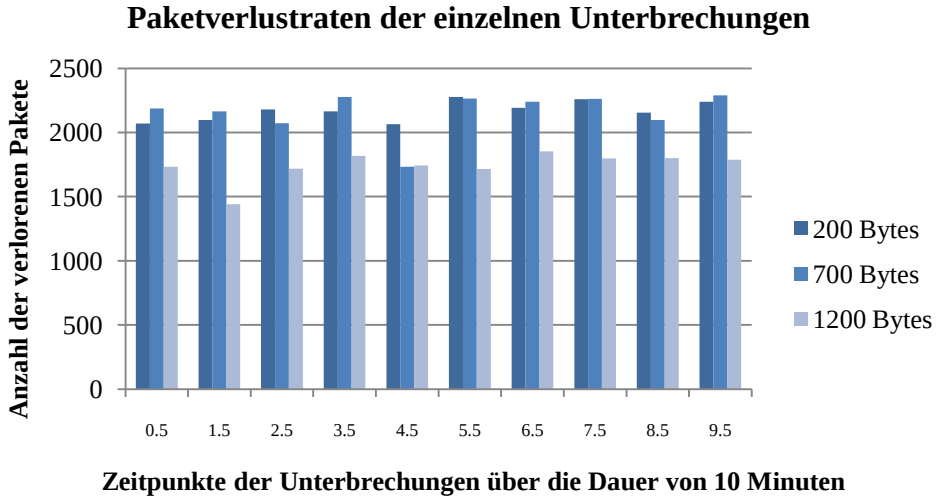

Abbildung 19: Paketverlustraten der einzelnen Unterbrechungen

Die oberhalb ersichtliche Abbildung zeigt die jeweiligen Paketverlustraten der drei MRP Messungen. Es wird die Anzahl der Pakete, welche pro Unterbrechung verloren wurde, veranschaulicht.

4.2 Messung des Parallel Redundancy Protocols

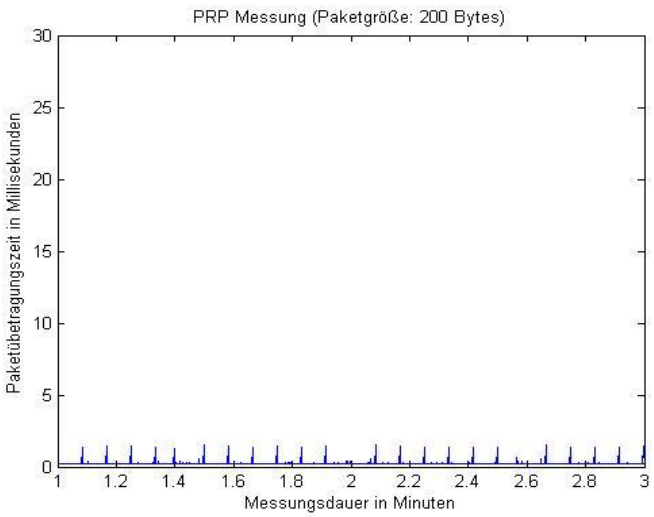

Abbildung 20: PRP Messung Szenario A

Die Messung des *Parallel Redundancy Protokolls* mit 200 Byte großen Paketen ergab eine durchschnittliche Paketübertragungszeit von 0,21 ms. Der maximale Jitter erreichte einen Wert von 9,35 ms. Die Paketverlustrate dieser Messung betrug wie erwartet null Prozent.

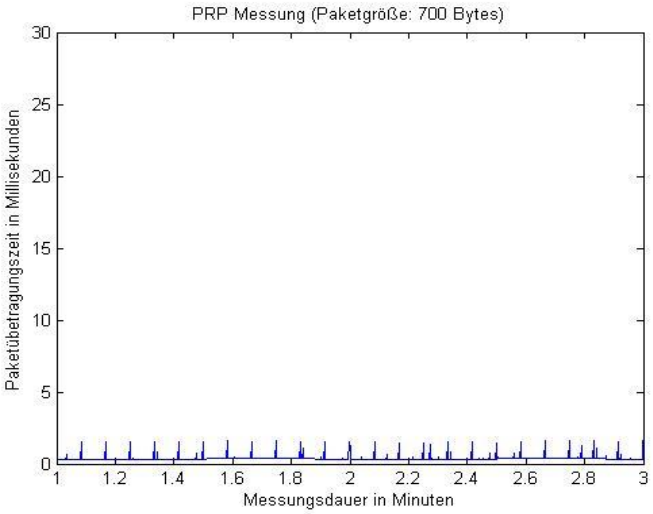

Abbildung 21: PRP Messung Szenario B

Die Messung mit 700 Bytes lässt keine größeren Verzögerungen erkennen. Die Paketübertragungszeit betrug im Schnitt 0,36 ms. Der Jitter erreichte einen maximalen Wert von 1,23 ms. Auch bei dieser Messung gingen keine Pakete verloren.

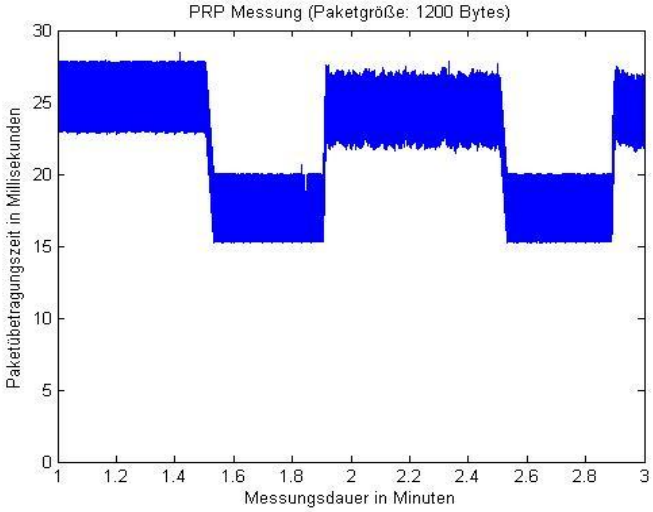

Abbildung 22: PRP Messung Szenario C

Eine belastende Messung mit Paketgrößen von 1200 Bytes wurde ebenfalls durchgeführt. Hierbei betrug die Zeit, welche zur Übertragung eines Paketes benötigt wurde, im Schnitt 21,73 ms. Die maximale Abweichung des durchschnittlichen Übertragungswertes betrug 10,54 ms. Bei dieser Messung lassen sich die Unterbrechungen mit Hilfe der oben ersichtlichen Grafik deutlich erkennen.

Paketgröße in Byte	**Anzahl der übertragenen Pakete**	**Anzahl der verlorenen Pakete**
200	7258593	0
700	7193927	0
1200	5883659	36085

Tabelle 9: Anzahl der übertragenen und verlorenen Pakete mit PRP

Die Messungen des PRP Protokolls erzielten schnellere Paketübertragungen als das MRP Protokoll. Wie erwartet ergaben die ersten beiden Messungen mit 200 und 700 Bytes Paketen eine Verlustrate von null Prozent. Die Messung mit 1200 Bytes ergab jedoch einen Verlustwert von 0,61 Prozent der Pakete, wobei dies nicht passieren dürfte. Es wird eine Überlastung aufgrund der Paketgrößen vermutet.

4.3 Messung des High Availability Seamless Redundancy Protocols

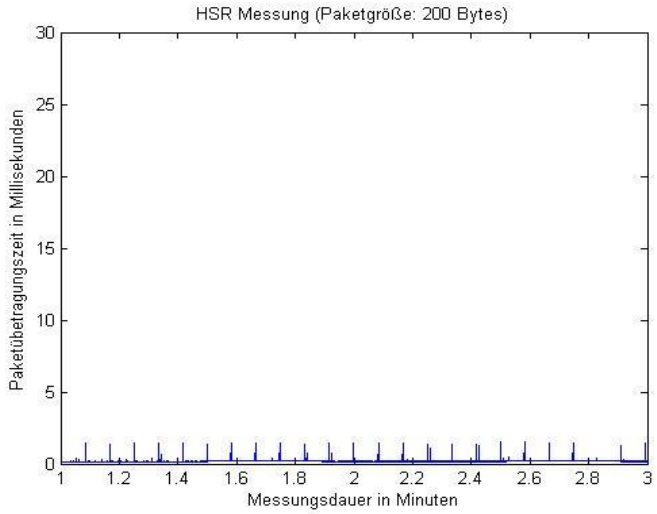

Abbildung 23: HSR Messung Szenario A

Die Messung des HSR Protokolls mit 200 Bytes ergab eine erneute Geschwindigkeitssteigerung im Vergleich zum PRP und MRP Protokoll. Mit einer durchschnittlichen Übertragungszeit von 0,14 ms ist es eindeutig am schnellsten, wobei ein maximaler Jitter von 1,44 ms erreicht wurde. Es konnte kein Paketverlust festgestellt werden.

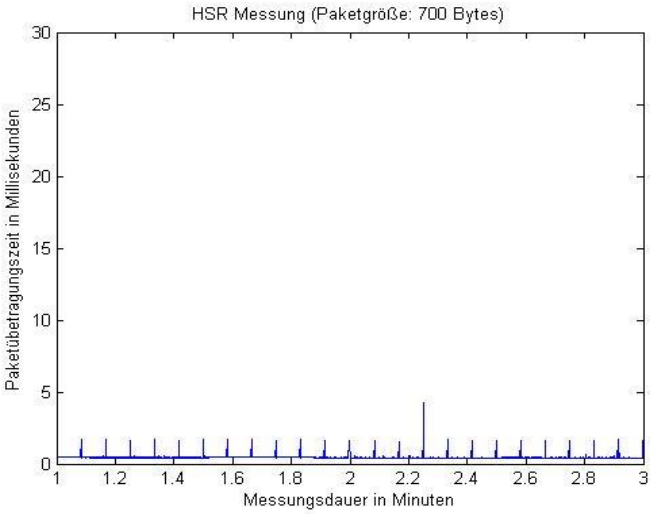

Abbildung 24: HSR Messung Szenario B

Die Messung mit 700 Bytes benötigte pro Paket durchschnittlich 0,42 ms für die Übertragung. Der maximale Jitter erreichte einen Wert von 8,38 ms. Auch bei dieser Messung wurden keine Pakete verloren.

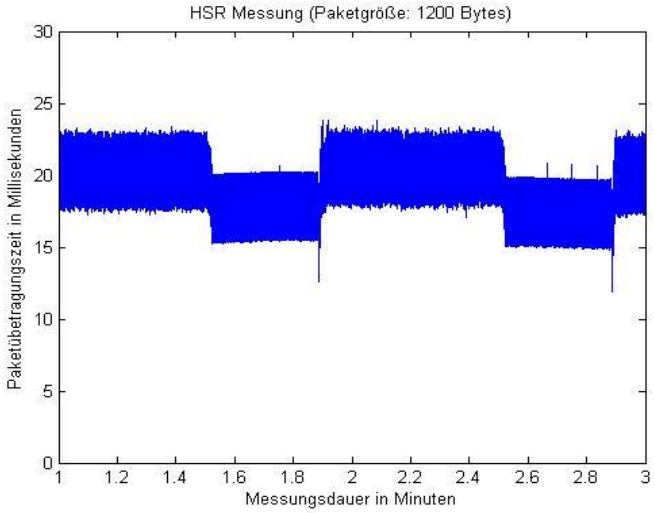

Abbildung 25: HSR Messung Szenario C

Wie zuvor beim PRP, wurde auch beim HSR Protokoll eine belastende Messung mit 1200 Bytes durchgeführt. Hierbei wurde eine durchschnittliche Übertragung von 18,31 ms gemessen. Der maximale Jitter betrug 17,81 ms. Auch während dieser Messung, wie zuvor beim PRP, kam es zu einem unerwarteten Paketverlust von 1,11 Prozent.

Paketgröße in Byte	**Anzahl der übertragenen Pakete**	**Anzahl der verlorenen Pakete**
200	7251230	0
700	7244580	0
1200	5854698	65077

Tabelle 10: Anzahl der übertragenen und verlorenen Pakete mit HSR

Anhand der Tabelle 10 kann man erkennen, dass die Messungen mit den geringeren Paketgrößen keinen Paketverlust aufzeichneten. Wiederum kam es dazu, dass bei der Messung mit 1200 Bytes großen Paketen ein Verlust aufgezeichnet wurde. Wie

zuvor bei der Belastungsmessung des *Parallel Redundancy Protokolls* wird davon ausgegangen, dass es zu einer Überlastung aufgrund der Paketgrößen gekommen ist.

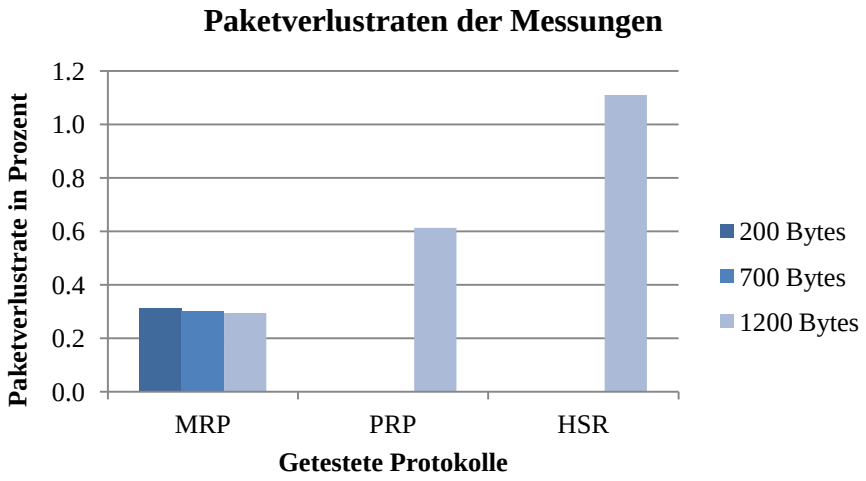

Abbildung 26: Paketverlustraten der Messungen

Die Abbildung veranschaulicht abschließend die erzielten Ergebnisse der Messungen, wobei alle Messungen samt ihren Paketverlustraten in Prozent dargestellt werden.

4.4 Résumé

In Anbetracht des gesetzten Zieles, das sicherste und gleichzeitig schnellste Protokoll herauszufinden, hat sich das *High Availability Seamless Redundancy Protokoll* als performantes Protokoll für Industrieumgebungen bewährt, denn es ergibt schnelle Paketübertragungszeiten und geringe Werte für den maximalen Jitter.

Das *Media Redundancy Protokoll* erweist sich bezüglich der Stabilität und Ausfallsicherheit als weniger geeignet, da es während jeder Messung, bei Paketen der Größe 200 Bytes, 700 Bytes und 1200 Bytes, zu einem Paketverlust kam. Das *Parallel Redundancy Protokoll* ergibt ähnliche Ergebnisse wie das HSR Protokoll. Da in diesen Testszenarien die Ergebnisse des PRP jenen des HSR sehr nahe kommen, und diese manchmal sogar übertreffen, ist dies vermutlich auf die Einfachheit der zwei verwendeten Netzwerke für die Messungen des PRP Protokolls zurückzuführen. Eines davon bestand aus einem einzigen Switch, woran ein PC angeschlossen war, wobei das andere lediglich eine einfache Leitung darstellte.

Wird der Einsatz eines solchen Protokolls in Industrien in Betracht gezogen, muss jedoch bedacht werden, dass das PRP und HSR Protokoll jeweils eine hohe Anzahl an Redundanz mit sich bringen, wobei beim MRP mit zwei zur Verfügung stehenden Pfaden weniger überflüssige Information vorhanden ist. Liegen die Anforderungen aber in Hochverfügbarkeit, Stabilität und Sicherheit, führt seit Einführung der Realtime Ethernet Protokolle PRP und HSR kein Weg an diesen vorbei.

Aufgrund der Tatsache, dass das *Parallel Redundancy Protokoll* und das *High Availability Seamless Redundancy Protokoll* ein Ergebnis versprechen, welches keinen Paketverlust aufweisen soll, verwundern die Messungen mit jeweils 1200 Byte großen Paketen, da diese, unter denselben Voraussetzungen, sogar mehr Paketverlust aufweisen als das *Media Redundancy Protokoll*. Es wurden mehrere Messungen unter denselben Bedingungen durchgeführt, jedoch kam es immer zu ähnlichen Ergebnissen, wobei die Messungen mit den kleineren Paketen die eigentlich vorausgesetzte Paketverlustrate von null Paketen erfüllen.

Werden somit die Messungen der kleineren Pakete für den Vergleich in Betracht gezogen, wirken HSR und PRP am überzeugendsten, wobei das PRP, wie bereits erwähnt, vermutlich aufgrund der unkomplizierten Netzwerkstrukturen, nahezu ähnliche Ergebnisse wie das HSR Protokoll lieferte.

5 Literaturverzeichnis

[1] S. Maurya, V. K. Nayak und A. Nagaraju, „Implementation of Data Link Control Protocols in Wired Network," *International Journal of Engineering Trends and Technology,* pp. 64-68, Dezember 2014.

[2] Hirschmann Automation and Control GmbH, „How does HIPER-Ring Redundancy work?," 2015. [Online]. Available: http://www.dacel.com.tr/upload/data/files/Dokuman/belden/3151_ring_redundancy_de_en.swf. [Zugriff am 16 Oktober 2015].

[3] A. Giorgetti, F. Cugini, F. Paolucci, L. Valcarenghi, A. Pistone und P. Castoldi, „Performance Analysis of Media Redundancy Protocol (MRP)," Februar 2013. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6145654&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F9424%2F4389054%2F06145654.pdf%3Farnumber%3D6145654. [Zugriff am 16 Oktober 2015].

[4] H. Kirrmann, „IEC SC65C WG15 Parallel Redundancy Protocol an IEC standard for a seamless redundancy method applicable to hard-real time Industrial Ethernet," 5 Juni 2012. [Online]. Available: http://lamspeople.epfl.ch/kirrmann/Pubs/IEC_62439-3/IEC_62439-3.4_PRP_Kirrmann.pdf. [Zugriff am 5 November 2015].

[5] H. Kirrmann, „Highly Available Automation Networks Standard Redundancy Methods Rationales behind the IEC 62439 standard suite," 2012. [Online]. Available: http://lamspeople.epfl.ch/kirrmann/Pubs/IEC_62439-1/IEC_62439_Summary.pdf. [Zugriff am 5 November 2015].

[6] H. H. Rentschler M., „The Parallel Redundancy Protocol for Industrial IP Networks," in *IEEE International Conference,* Kapstadt, 2013.

[7] S. Meier, „Doppelt gemoppelt hält besser!," 25 Jänner 2007. [Online]. Available: https://www.zhaw.ch/storage/engineering/institute-zentren/ines/forschung-und-entwicklung/time-synchronisation/doppelt-gemoppelt-haelt-besser.pdf. [Zugriff am 5 November 2015].

[8] Hirschmann Automation and Control GmbH, „Parallel Redundancy Protocol (PRP),“ 2015. [Online]. Available: http://www.hirschmann.com/de/Hirschmann/Industrial_Ethernet/Technologien/PRP_-_Parallel_Redundancy_Protocol/index.phtml. [Zugriff am 5 November 2015].

[9] Hirschmann Automation and Control GmbH, „High Availability Seamless Redundancy (HSR),“ 2015. [Online]. Available: http://www.hirschmann.com/de/Hirschmann/Industrial_Ethernet/Technologien/HSR_uE2u80u93_High_Availability_Seamless_Redundancy/index.phtml. [Zugriff am 18 Oktober 2015].

[10] H. Kirrmann, „HSR - High Availability Seamless Redundancy,“ 2014. [Online]. Available: http://lamspeople.epfl.ch/kirrmann/Pubs/IEC_62439-3/IEC_62439-3.5_HSR_Kirrmann.pdf. [Zugriff am 6 November 2015].

[11] Hirschmann Automation and Control GmbH, „Medienredundanzkonzepte,“ 2014. [Online]. Available: http://belden.picturepark.com/Website/Download.aspx?DownloadToken=c5a2ca65-3090-42b2-a541-b687520c8fe5&Purpose=AssetManager&mime-type=application/pdf. [Zugriff am 27 Jänner 2016].

[12] N. X. Tien, S. Nsaif und J. M. Rhee, „High-availability Seamless Redundancy (HSR) Traffic Reduction Using Optimal Dual Paths (ODP),“ in *International Conference on Green and Human Information Technology (ICGHIT)*, Vietnam, 2015.

[13] Hirschmann Automation and Control GmbH, „Hirschmann™ Managed RSP Switches,“ 2015. [Online]. Available: http://www.hirschmann.com/de/Hirschmann/Industrial_Ethernet/managed_rsp_switches/index.phtml. [Zugriff am 18 Oktober 2015].

[14] H. Kirrmann, I. Sotiropoulos, D. Ilie und C. Hornegger, „Industrial Ethernet seamless redundancy and sub-microsecond clock synchronization with IEC 62439-3 and IEC 61588,“ in *IEEE International Conference on Emerging Technologies and Factory Automation*, Toulouse, 2011.

[15] I. R. Altaha, J. M. Rhee und H.-A. Pham, „Improvement of High-Availability Seamless Redundancy (HSR) Unicast Traffic Performance Using Enhanced Port Locking (EPL) Approach,“ *IEICE Transactions on Information and Systems*, pp.

1646-1656, September 2015.

[16] Hirschmann Automation and Control GmbH, „User Manual Redundancy Configuration," Juli 2010. [Online]. Available: https://www.e-catalog.beldensolutions.com/download/managed/pim/640ed87e-f1eb-4fd8-b38b-fb066a49e391/UM_RedundConfig_L2P_Rel60_en.pdf;jsessionid=13EED78C7286D89D92DB0C3B8ECFF994?type=attachment. [Zugriff am 13 Oktober 2015].

[17] Hirschmann Automation and Control GmbH, „Precision Time Protocol (PTP)," 2015. [Online]. Available: http://www.hirschmann.com/de/Hirschmann/Industrial_Ethernet/Technologien/Precision_Time_Protocol/index.phtml. [Zugriff am 19 Oktober 2015].

[18] A. Dreher und D. Mohl, „Precision Clock Synchronization - IEEE 1588," 2015. [Online]. Available: https://www.belden.com/docs/upload/Precision_Clock_Synchronization_WP.pdf. [Zugriff am 19 Oktober 2015].

[19] J. D. Guyton und M. F. Schwartz, „Experiences with a Survey Tool for Discovering Network Time Protocol Servers," Boulder, 1994.

6 Anhang

6.1 Switch Konfigurationen

6.1.1 MRP Switch Konfiguration

Switch 1 (Redundanzmanager):

no network hidiscovery blinking

network parms 10.0.0.1 255.255.255.0 10.0.0.254

network protocol none

network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable

network management access status 1 enable

vlan database

exit

configure

mrp domain add default-domain

mrp domain modify port primary 1/1

mrp domain modify port secondary 1/2

mrp domain modify advanced-mode enable

mrp domain modify manager-priority 32768

mrp domain modify mode manager

mrp domain modify name ""

mrp domain modify recovery-delay 200ms

mrp domain modify vlan 0

mrp domain modify operation enable

mrp operation enable

dns cache adminstate

no config watchdog admin-state

config watchdog timeout 600

logging email from-addr switch@hirschmann.com

no spanning-tree operation

system name RSP-Eth1

interface 1/1

voice vlan disable

power-state

exit

interface 1/2

voice vlan disable

power-state

exit

interface 1/10

voice vlan disable

exit

interface 1/11

voice vlan disable

exit

Switch 2:

no network hidiscovery blinking

network parms 10.0.0.2 255.255.255.0 10.0.0.254

network protocol none

network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable

network management access status 1 enable

vlan database

exit

configure

mrp domain add default-domain

mrp domain modify port primary 1/1

mrp domain modify port secondary 1/2

mrp domain modify advanced-mode enable

mrp domain modify manager-priority 32768

mrp domain modify mode client

mrp domain modify name ""

mrp domain modify recovery-delay 200ms

mrp domain modify vlan 0

mrp domain modify operation enable

mrp operation enable

dns cache adminstate

no config watchdog admin-state

config watchdog timeout 600

logging email from-addr switch@hirschmann.com

no spanning-tree operation

system name RSP-Eth2

interface 1/1

voice vlan disable

power-state

exit

interface 1/2

voice vlan disable

power-state

exit

interface 1/10

voice vlan disable

exit

interface 1/11

voice vlan disable

exit

Switch 3:

no network hidiscovery blinking

network parms 10.0.0.3 255.255.255.0 10.0.0.254

network protocol none

network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable

network management access status 1 enable

vlan database

exit

configure

mrp domain add default-domain

mrp domain modify port primary 1/1

mrp domain modify port secondary 1/2

mrp domain modify advanced-mode enable

mrp domain modify manager-priority 32768

mrp domain modify mode client

mrp domain modify name ""

mrp domain modify recovery-delay 200ms

mrp domain modify vlan 0

mrp domain modify operation enable

mrp operation enable

dns cache adminstate

no config watchdog admin-state

config watchdog timeout 600

logging email from-addr switch@hirschmann.com

no spanning-tree operation

system name RSP-Eth3

interface 1/1

voice vlan disable

power-state

exit

```
interface 1/2
voice vlan disable
power-state
exit
interface 1/10
voice vlan disable
exit
interface 1/11
voice vlan disable
exit
```

Script 1: MRP Switchkonfiguration

6.1.2 HSR Switch Konfiguration

Switch 1:

```
no network hidiscovery blinking
network parms 10.0.0.1 255.255.255.0 10.0.0.254
network protocol none
network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp
enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable
network management access status 1 enable
vlan database
exit
configure
hsr instance 1 operation enable
hsr instance 1 mode modeu
no config watchdog admin-state
config watchdog timeout 600
logging email from-addr switch@hirschmann.com
ptp v2-transparent-clock delay-mechanism p2p
no spanning-tree operation
system name RSP-Eth1
interface 1/1
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/2
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
```

```
interface 1/10
voice vlan disable
exit
interface 1/11
voice vlan disable
exit
interface hsr/1
no mac notification operation
mrp-ieee global join-time 20
mrp-ieee global leave-time 60
mrp-ieee global leave-all-time 1000
mrp-ieee mvrp operation
mrp-ieee mmrp operation
no ip dhcp-snooping trust
no ip dhcp-snooping log
ip dhcp-snooping auto-disable
ip dhcp-snooping limit -1 1
classofservice trust dot1p
no dhcp-l2relay mode
no dhcp-l2relay trust
dhcp-server operation
no igmp-snooping mode
no igmp-snooping fast-leave
igmp-snooping groupmembership-interval 260
igmp-snooping maxresponse 10
igmp-snooping mcrtrexpiretime 260
no igmp-snooping static-query-port
name ""
storm-control flow-control
no storm-control ingress broadcast operation
storm-control ingress broadcast threshold 0
no storm-control ingress multicast operation
storm-control ingress multicast threshold 0
no storm-control ingress unicast operation
storm-control ingress unicast threshold 0
storm-control ingress unit percent
traffic-shape bw 0
exit
```

Switch 2:

no network hidiscovery blinking
network parms 10.0.0.2 255.255.255.0 10.0.0.254
network protocol none
network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp
enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable
network management access status 1 enable
vlan database
exit
configure
hsr instance 1 operation enable
hsr instance 1 mode modeu
no config watchdog admin-state
config watchdog timeout 600
logging email from-addr switch@hirschmann.com
ptp v2-transparent-clock delay-mechanism p2p
no spanning-tree operation
system name RSP-Eth2
interface 1/1
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/2
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/10
voice vlan disable
exit
interface 1/11
voice vlan disable
exit
interface hsr/1
no mac notification operation
mrp-ieee global join-time 20
mrp-ieee global leave-time 60
mrp-ieee global leave-all-time 1000
mrp-ieee mvrp operation
mrp-ieee mmrp operation
no ip dhcp-snooping trust

no ip dhcp-snooping log
ip dhcp-snooping auto-disable
ip dhcp-snooping limit -1 1
classofservice trust dot1p
no dhcp-l2relay mode
no dhcp-l2relay trust
dhcp-server operation
no igmp-snooping mode
no igmp-snooping fast-leave
igmp-snooping groupmembership-interval 260
igmp-snooping maxresponse 10
igmp-snooping mcrtrexpiretime 260
no igmp-snooping static-query-port
name ""
storm-control flow-control
no storm-control ingress broadcast operation
storm-control ingress broadcast threshold 0
no storm-control ingress multicast operation
storm-control ingress multicast threshold 0
no storm-control ingress unicast operation
storm-control ingress unicast threshold 0
storm-control ingress unit percent
traffic-shape bw 0
exit

Switch 3:

no network hidiscovery blinking
network parms 10.0.0.3 255.255.255.0 10.0.0.254
network protocol none
network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp
enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable
network management access status 1 enable
vlan database
exit
configure
hsr instance 1 operation enable
hsr instance 1 mode modeu
no config watchdog admin-state
config watchdog timeout 600
logging email from-addr switch@hirschmann.com

```
ptp v2-transparent-clock delay-mechanism p2p
no spanning-tree operation
system name RSP-Eth3
interface 1/1
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/2
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/10
voice vlan disable
exit
interface 1/11
voice vlan disable
exit
interface hsr/1
no mac notification operation
mrp-ieee global join-time 20
mrp-ieee global leave-time 60
mrp-ieee global leave-all-time 1000
mrp-ieee mvrp operation
mrp-ieee mmrp operation
no ip dhcp-snooping trust
no ip dhcp-snooping log
ip dhcp-snooping auto-disable
ip dhcp-snooping limit -1 1
classofservice trust dot1p
no dhcp-l2relay mode
no dhcp-l2relay trust
dhcp-server operation
no igmp-snooping mode
no igmp-snooping fast-leave
igmp-snooping groupmembership-interval 260
igmp-snooping maxresponse 10
igmp-snooping mcrtrexpiretime 260
no igmp-snooping static-query-port
name ""
storm-control flow-control
no storm-control ingress broadcast operation
```

```
storm-control ingress broadcast threshold 0
no storm-control ingress multicast operation
storm-control ingress multicast threshold 0
no storm-control ingress unicast operation
storm-control ingress unicast threshold 0
storm-control ingress unit percent
traffic-shape bw 0
exit
```

Script 2: HSR Switchkonfiguration

6.1.3 PRP Switch Konfiguration

Switch 1:

```
no network hidiscovery blinking
network parms 10.0.0.1 255.255.255.0 10.0.0.254
network protocol none
network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp
enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable
network management access status 1 enable
vlan database
exit
configure
prp instance 1 operation enable
no config watchdog admin-state
config watchdog timeout 600
logging email from-addr switch@hirschmann.com
ptp v2-transparent-clock delay-mechanism p2p
no spanning-tree operation
system name RSP-Eth1
interface 1/1
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/2
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/10
voice vlan disable
exit
```

interface 1/11
voice vlan disable
exit
interface prp/1
no mac notification operation
mrp-ieee global join-time 20
mrp-ieee global leave-time 60
mrp-ieee global leave-all-time 1000
mrp-ieee mvrp operation
mrp-ieee mmrp operation
no ip dhcp-snooping trust
no ip dhcp-snooping log
ip dhcp-snooping auto-disable
ip dhcp-snooping limit -1 1
classofservice trust dot1p
no dhcp-l2relay mode
no dhcp-l2relay trust
dhcp-server operation
no igmp-snooping mode
no igmp-snooping fast-leave
igmp-snooping groupmembership-interval 260
igmp-snooping maxresponse 10
igmp-snooping mcrtrexpiretime 260
no igmp-snooping static-query-port
name ""
storm-control flow-control
no storm-control ingress broadcast operation
storm-control ingress broadcast threshold 0
no storm-control ingress multicast operation
storm-control ingress multicast threshold 0
no storm-control ingress unicast operation
storm-control ingress unicast threshold 0
storm-control ingress unit percent
traffic-shape bw 0
exit

Switch 2:

no network hidiscovery blinking
network parms 10.0.0.2 255.255.255.0 10.0.0.254
network protocol none

network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp
enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable
network management access status 1 enable
vlan database
exit
configure
prp instance 1 operation enable
no config watchdog admin-state
config watchdog timeout 600
logging email from-addr switch@hirschmann.com
ptp v2-transparent-clock delay-mechanism p2p
no spanning-tree operation
system name RSP-Eth2
interface 1/1
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/2
power-state
ptp v2-boundary-clock delay-mechanism p2p
exit
interface 1/10
voice vlan disable
exit
interface 1/11
voice vlan disable
exit
interface prp/1
no mac notification operation
mrp-ieee global join-time 20
mrp-ieee global leave-time 60
mrp-ieee global leave-all-time 1000
mrp-ieee mvrp operation
mrp-ieee mmrp operation
no ip dhcp-snooping trust
no ip dhcp-snooping log
ip dhcp-snooping auto-disable
ip dhcp-snooping limit -1 1
classofservice trust dot1p
no dhcp-l2relay mode
no dhcp-l2relay trust

dhcp-server operation
no igmp-snooping mode
no igmp-snooping fast-leave
igmp-snooping groupmembership-interval 260
igmp-snooping maxresponse 10
igmp-snooping mcrtrexpiretime 260
no igmp-snooping static-query-port
name ""
storm-control flow-control
no storm-control ingress broadcast operation
storm-control ingress broadcast threshold 0
no storm-control ingress multicast operation
storm-control ingress multicast threshold 0
no storm-control ingress unicast operation
storm-control ingress unicast threshold 0
storm-control ingress unit percent
traffic-shape bw 0
exit

Switch 3 (Dummy):

no network hidiscovery blinking
network parms 10.0.0.3 255.255.255.0 10.0.0.254
network protocol none
network management access add 1 ip 0.0.0.0 mask 0 http enable https enable snmp
enable telnet enable iec61850-mms enable modbus-tcp enable ssh enable
network management access status 1 enable
vlan database
exit
configure
no config watchdog admin-state
config watchdog timeout 600
logging email from-addr switch@hirschmann.com
no spanning-tree operation
system name RSP-Eth3
interface 1/1
voice vlan disable
exit
interface 1/2
voice vlan disable
exit

```
interface 1/10
voice vlan disable
exit
interface 1/11
voice vlan disable
exit
```

Script 3: PRP Switchkonfiguration

6.2 Matlab Script zum Auswerten der Daten

```
%%Messung der Paketdauer zur Zeit
load MRP_1200bytes.dat
raw = MRP_1200bytes(:,:);
paketpointer = raw(:,1);
paketcounter = size(raw(:,1));
startzeit = raw(:,2);
endzeit = raw(:,3);
%in Minuten
testzeit = (startzeit(:,1) - startzeit(1,1))/60000000;
%in Millisekunden
paketzeit = (endzeit(:,1) - startzeit(:,1))/1000;
%plot der Paketverzögerung
plot(testzeit,paketzeit);
title('Visualisierung der Verzögerung');
xlabel('Messungsdauer in Minuten');
ylabel('Paketverzögerung in Millisekunden');
axis([4 7 -10 300]);

%%Berechnung des Jitters
%in Millisekunden
durchschnittspaketzeit = mean(paketzeit)*-1;
%in Millisekunden
jitter = ((paketzeit(:,1) -
durchschnittspaketzeit)/paketcounter(1,1))*-1;
maximalerjitter = max(jitter(:,1));

%%Paketverlust ermitteln
Paketnummer = paketpointer(size(raw(:,1)),1);
Paketverlust = Paketnummer(1,1)-paketcounter;
durchschnittspaketverlust = Paketverlust(1,1)/10;
```

```
%pro Unterbrechung in Millisekunden
Verlustzeit = durchschnittspaketverlust*durchschnittspaketzeit;

%%Zählen des Reordering
reorder = 1;
number = 1;
i = 1;
%Benötigte Schleife
while (number ~= paketcounter(1,1))

    if(number ~= paketpointer(i))
        reorder = reorder +1;
    end
    number = number +1;
      i = i +1;
end
```

Script 4: Code zur Auswertung der Messdaten

6.3 Java Code zur Ermittlung der Paketverluste

```
package bac1;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class BAC1 {

    public static void main(String[] args) throws FileNotFoundException,
IOException {

        ArrayList<Long> first = new ArrayList<>();
        ArrayList<Long> second = new ArrayList<>();
        ArrayList<Long> third = new ArrayList();
        String[] splitted;
        ArrayList<Long> breaks = new ArrayList<>();
```

```
        ArrayList<Long> breaks_pos = new ArrayList<>();


      BufferedReader in = new BufferedReader(new
FileReader("/MRP_1200bytes.dat"));
      String str;


      while ((str = in.readLine()) != null) {

             splitted = str.split("\t");

             first.add(Long.parseLong(splitted[0]));
             second.add(Long.parseLong(splitted[1]));
             third.add(Long.parseLong(splitted[2]));
      }

       in.close();

        int file_size = first.size();

       //Zur Berechnung der auftretenden Paketverluste bzw. deren Position
(Paket Nr. worauf Verlust folgt)
       for(int i=0; i<file_size;i++){

           if(i>0){
           if(first.get(i) > (first.get(i-1))+1){

               Long diff = first.get(i)-(first.get(i-1));
               breaks.add(diff-1);

               long pos = first.get(i)-diff;
               breaks_pos.add(pos);

           }
           }
       }

       //Ausgabe der Ergebnisse der Paketverluste
       for(int i=0;i<breaks.size();i++){
        System.out.println(i+". "+"Pos: " + breaks_pos.get(i));
        System.out.println(breaks.get(i));
```

```
    }


        long loss = first.get(first.size()-1)-first.size();


        double proc1 = (double)100/(double)(first.size()); // stellt 1% dar
(Anzahl der Pakete)
        double proc_val = proc1*(double)loss;



        System.out.println("Gesamtanzahl der übertragenen Pakete: " +
first.size());
        System.out.println("Gesamtanzahl der verlorenen Pakete: " + loss);
        System.out.println("Prozentsatz der verlorenen Pakete im Vergleich
zu den übertragenen Paketen: " + proc_val + " %");


    }
}
```

Script 5: Java Code zur Ermittlung der Paketverlustrate

6.4 NTP Konfiguration

6.4.1 NTP Client

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help
driftfile /var/lib/ntp/ntp.drift
# Enable this if you want statistics to be logged.
#statsdir /var/log/ntpstats/
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
# Specify one or more NTP servers.
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
server 192.168.1.3
# Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for
# details.  The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
```

```
# up blocking replies from your own upstream servers.
# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust
# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255
# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines.  Please do this only if you trust everybody on the network!
#disable auth
#broadcastclient
```

Script 6: NTP Client Script

6.4.2 NTP Server

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help
driftfile /var/lib/ntp/ntp.drift
# Enable this if you want statistics to be logged.
#statsdir /var/log/ntpstats/
statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable
# Specify one or more NTP servers.
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
server 127.127.1.0
fudge 127.127.1.0 stratum 10
# Allow LAN machines to use you as Time Server:
restrict 192.168.1.1 mask 255.255.255.0 nomodify notrap
restrict 192.168.1.2 mask 255.255.255.0 nomodify notrap
# Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for
# details.  The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
```

```
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.
# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery
# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust
# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255
# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines.  Please do this only if you trust everybody on the network!
#disable auth
#broadcastclient
minpoll 4
```

Script 7: NTP Server Script