

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.

Ort, Datum Personenkennzeichen Unterschrift des/der Studierenden

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 bzw. 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 wurden 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: Media Redundancy Protokoll (MRP), Parallel Redundancy Protokoll (PRP) und das High Availability Seamless Redundancy Protokoll (HSR). Durch die Testumgebung werden die Protokolle bezüglich Paketübertragungszeit, Paketverlustrate, Jitter und Regenerationszeit, aus welcher sich auch Hochverfügbarkeit erkennen lässt, getestet. Die Ergebnisse der Messungen haben gezeigt, dass sich HSR bezüglich industriellen 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 will be tested for stability and speed. A test installation are realized with three computers and three industrial Ethernet switches. The tested protocols are: Media Redundancy Protocol (MRP), Parallel Redundancy Protocol (PRP) and the High Availability Seamless Redundancy Protocol (HSR). With the testing environment these protocols are tested concerning packet transmission time, packet loss rate, jitter and regeneration time. The results have shown, that HSR meets all requirements concerning industrial environments.

Inhaltsverzeichnis

[1 Einleitung 1](#_Toc442552758)

[2 Theoretischer Teil 2](#_Toc442552759)

[2.1 HiPER- Ring Protocol 3](#_Toc442552760)

[2.2 Media Redundancy Protocol 4](#_Toc442552761)

[2.3 Parallel Redundancy Protocol 6](#_Toc442552762)

[2.4 High Availability Redundancy Seamless Protocol 8](#_Toc442552763)

[2.5 Link Aggregation Control Protocol 11](#_Toc442552764)

[2.6 Zeitsynchronisation 12](#_Toc442552765)

[3 Praktischer Teil 14](#_Toc442552766)

[3.1 Vorbereitungen 14](#_Toc442552767)

[3.2 Verwendete Materialien 15](#_Toc442552768)

[3.3 Konfiguration der Switches und PCs 16](#_Toc442552769)

[3.3.1 Konfiguration der Zeitsynchronisation 17](#_Toc442552770)

[3.3.2 Konfiguration der Link Aggregation (LACP) 17](#_Toc442552771)

[3.3.3 Verwendung des PathEval Tools 17](#_Toc442552772)

[3.4 Untersuchung des Media Redundancy Protocols (MRP) 18](#_Toc442552773)

[3.5 Untersuchung des High Availability Seamless Redundancy Protocols (HSR) 19](#_Toc442552774)

[3.6 Untersuchung des Parallel Redundancy Protocols (PRP) 20](#_Toc442552775)

[4 Ergebnisse 22](#_Toc442552776)

[4.1 Messung des Media Redundancy Protocols 22](#_Toc442552777)

[4.2 Messung des Parallel Redundancy Protocols 25](#_Toc442552778)

[4.3 Messung des High Availability Seamless Redundancy Protocols 27](#_Toc442552779)

[4.4 Résumé 30](#_Toc442552780)

[5 Literaturverzeichnis 31](#_Toc442552781)

[6 Anhang 34](#_Toc442552782)

[6.1 Switch Konfigurationen 34](#_Toc442552783)

[6.1.1 MRP Switch Konfiguration 34](#_Toc442552784)

[6.1.2 HSR Switch Konfiguration 38](#_Toc442552785)

[6.1.3 PRP Switch Konfiguration 44](#_Toc442552786)

[6.2 Matlab Script zum Auswerten der Daten 49](#_Toc442552787)

[6.3 6.3 Java Programm zum rausfiltern der Paket Verluste 51](#_Toc442552788)

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

NTP Network Time Protocol

Abbildungsverzeichnis

[Abbildung 1: Zeitspanne der Protokolle 2](#_Toc442552728)

[Abbildung 2: Herkömmlicher Backbone Ring 3](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552729)

[Abbildung 3: HiPER-Ring Verbindung 4](#_Toc442552730)

[Abbildung 4: Media Redundancy Protocol geschlossen [3] 4](#_Toc442552731)

[Abbildung 5: Media Redundancy Protocol offen [3] 5](#_Toc442552732)

[Abbildung 6: Netzwerk mit Parallel Redundancy Protocol [8] 7](#_Toc442552733)

[Abbildung 7: HSR Ringnetzwerk [9] 9](#_Toc442552734)

[Abbildung 8: HSR Multicast Prinzip [10] 10](#_Toc442552735)

[Abbildung 9: HSR Unicast Prinzip [10] 11](#_Toc442552736)

[Abbildung 10: Link Aggregation [16] 12](#_Toc442552737)

[Abbildung 11: PTP Master Slave Prinzip [18] 13](#_Toc442552738)

[Abbildung 12: Management Netzwerk 16](#_Toc442552739)

[Abbildung 13: MRP Topologie 18](#_Toc442552740)

[Abbildung 14: HSR Topologie 19](#_Toc442552741)

[Abbildung 15: PRP Topologie 20](#_Toc442552742)

[Abbildung 16: MRP Messung Szenario A 22](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552743)

[Abbildung 17: MRP Messung Szenario B 23](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552744)

[Abbildung 18: MRP Messung Szenario C 23](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552745)

[Abbildung 19: PRP Messung Szenario A 25](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552746)

[Abbildung 20: PRP Messung Szenario B 25](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552747)

[Abbildung 21: PRP Messung Szenario C 26](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552748)

[Abbildung 22: HSR Messung Szenario A 27](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552749)

[Abbildung 23: HSR Messung Szenario B 28](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552750)

[Abbildung 24: HSR Messung Szenario C 28](file:///C:\Users\wiela\Downloads\BAC_RealtimeEthernetProtokolle.docx#_Toc442552751)

[Abbildung 25: Paketverlustrate der Messungen 29](#_Toc442552752)

Code-Snippet-Verzeichnis

[Script 1: MRP Switchkonfiguration 38](#_Toc442552753)

[Script 2: HSR Switchkonfiguration 44](#_Toc442552754)

[Script 3: PRP Switchkonfiguration 49](#_Toc442552755)

[Script 4: Code zur Auswertung der Messdaten 51](#_Toc442552756)

[Script 5: Java Code zur Ermittlung des Paketverlustes 52](#_Toc442552757)

Tabellen-Verzeichnis

[Tabelle 1: Verwendete Switches 15](#_Toc442552716)

[Tabelle 2: Verwendete Rechner 1+3 15](#_Toc442552717)

[Tabelle 3: Verwendeter Rechner 2 15](#_Toc442552718)

[Tabelle 4: IP Adressen der PCs 16](#_Toc442552719)

[Tabelle 5: IP Adressen der Switches 16](#_Toc442552720)

[Tabelle 6: MRP Messungen 19](#_Toc442552721)

[Tabelle 7: HSR Messungen 20](#_Toc442552722)

[Tabelle 8: PRP Messungen 21](#_Toc442552723)

[Tabelle 9: Anzahl der übertragenen und verlorenen Pakete mit MRP 24](#_Toc442552724)

[Tabelle 10: Paketverlustraten der einzelnen Unterbrechungen mit MRP 24](#_Toc442552725)

[Tabelle 11: Anzahl der übertragenen und verlorenen Pakete mit PRP 27](#_Toc442552726)

[Tabelle 12: Anzahl der übertragenen und verlorenen Pakete mit HSR 29](#_Toc442552727)

# Einleitung

Netzwerke sind im alltäglichen Leben nicht mehr wegzudenken, wobei diese nicht nur privat, sondern auch in vielen großen 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 Netzwerk einer Industrie zu simulieren und die dafür entsprechenden Redundanz Protokolle zu konfigurieren, um eine stabile Leitung mit im besten Fall geringster Verzögerung zu erzeugen.

Realisiert wird das mit den heutzutage gängigsten und neuesten Ethernet Protokollen. Diese Protokolle werden auf unserer simulierten industriellen Umgebung konfiguriert und getestet. Die Ergebnisse werden dann verglichen und sich für die beste Konstellation von Aufbau und Protokoll entschieden.

# 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 der Layer 2 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, Verzögerungszeit und der Regenerationszeit verglichen. Zur Zeitsynchronisation wird ein *Precision Time Protocol* (PTP) angestrebt.

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

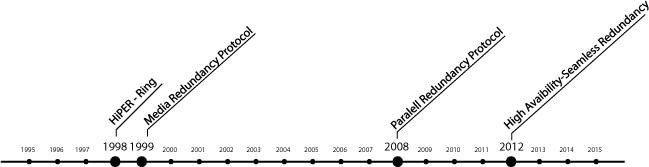


Abbildung 1: Zeitspanne der Protokolle

## HiPER- Ring Protocol

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

Das „High Performance Redundancy“ HiPER-Ring Protokoll wurde von der Firma Hirschmann & Siemens in Kooperation entwickelt. In normalen Backbone Netzen wird eine Verbindung zwischen einer Anzahl N Switches aufgebaut in Linien- Struktur (siehe Abbildung 1).

Wenn nun 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.

Redundanz - Manager

S1

S4

S3

S2

Abbildung 2: Herkömmlicher Backbone Ring

Sollte ein Fehler entstehen, Wird über die redundante Verbindung ebenfalls eine Mit

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

Redundanz - Manager

S3

S2

S4

S1

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 300ms.

## Media Redundancy Protocol

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

MRP arbeitet auf Layer 2 und ist eine direkte Abwandlung bzw. Erweiterung des HiPER-Rings, welches von Hirschmann weiterentwickelt wurde. Hier handelt es sich um einen IEC Standard 62439-2 beschriebenen 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 angezeigt.

MRM

MRC3

MRC1

MRC2

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 weiterübertragen und bei *disabled* wird gar 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 nun beispielsweise zu einem Verbindungsabbruch kommt, dann senden die MRCs ein „LinkChange Frame“ an den MRM, damit der Media Redundancy Manager den *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 diesbezügliches Szenario veranschaulicht.

MRC2

MRC1

MRC3

MRM

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 200ms und 500ms andauern. Während dieses „Switch-overs“ kann es zu Frame Verlusten kommen.

## 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 Protokoll 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 Protokoll 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.

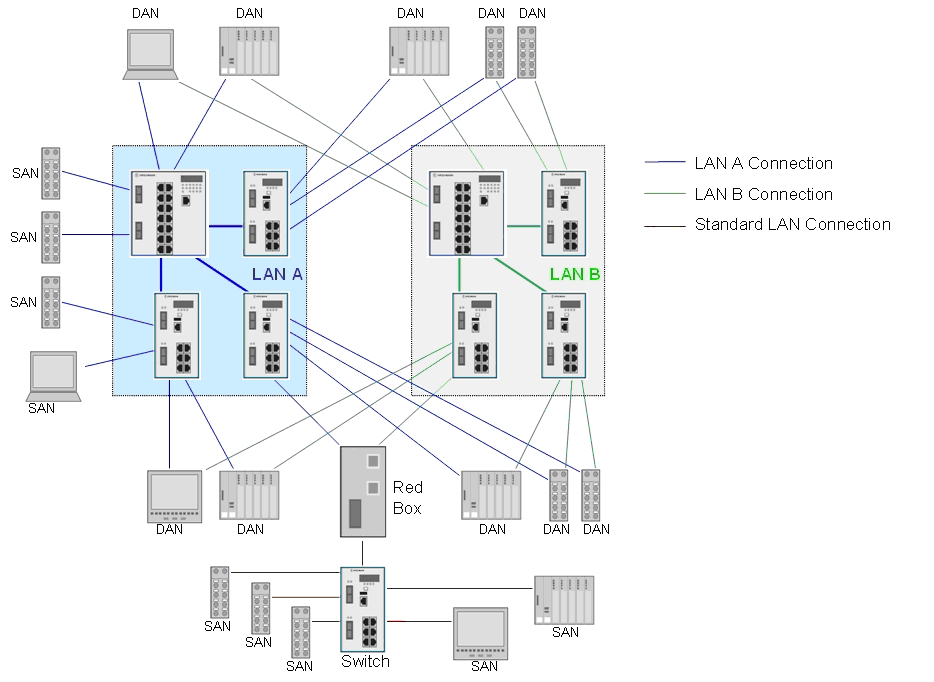


Abbildung 6: Netzwerk mit Parallel Redundancy Protocol [8]

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].

## 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 4 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].

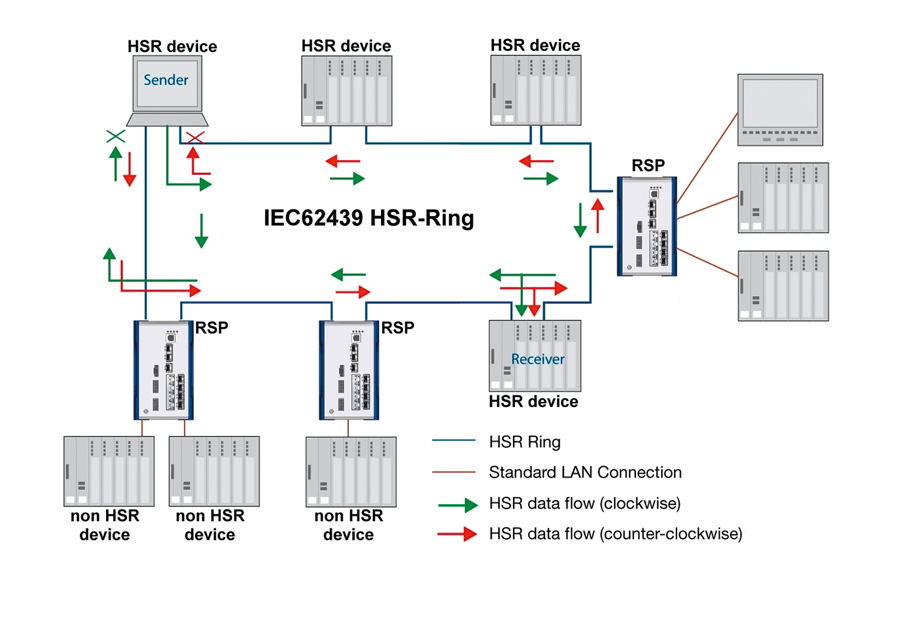


Abbildung 7: HSR Ringnetzwerk [9]

Beim HSR Protokoll ist es nicht möglich, Standardgeräte (SAN), 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 Protokoll 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 dessen 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].

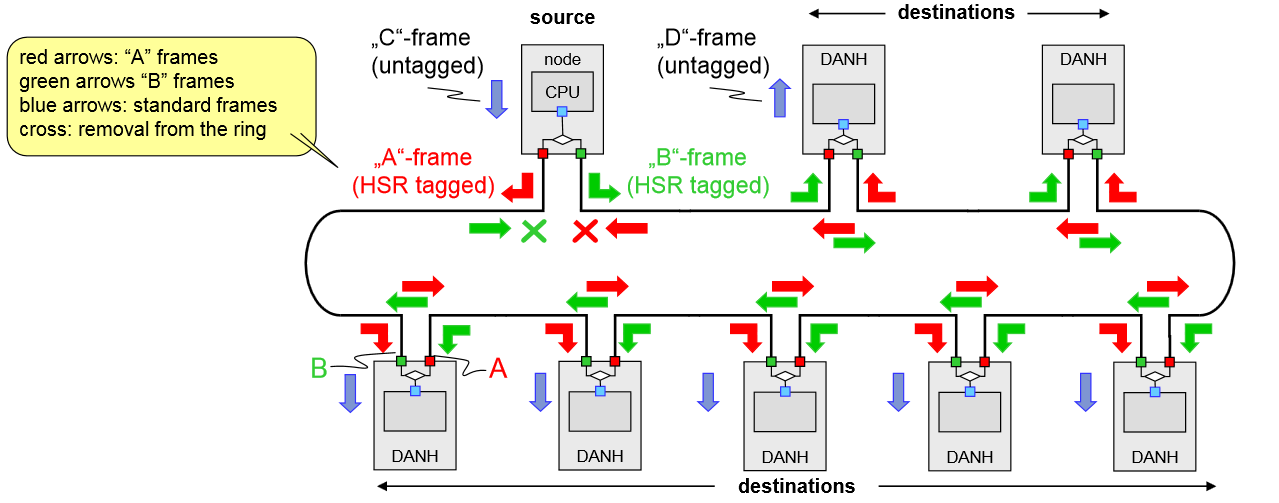


Abbildung 8: HSR Multicast Prinzip [10]

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].

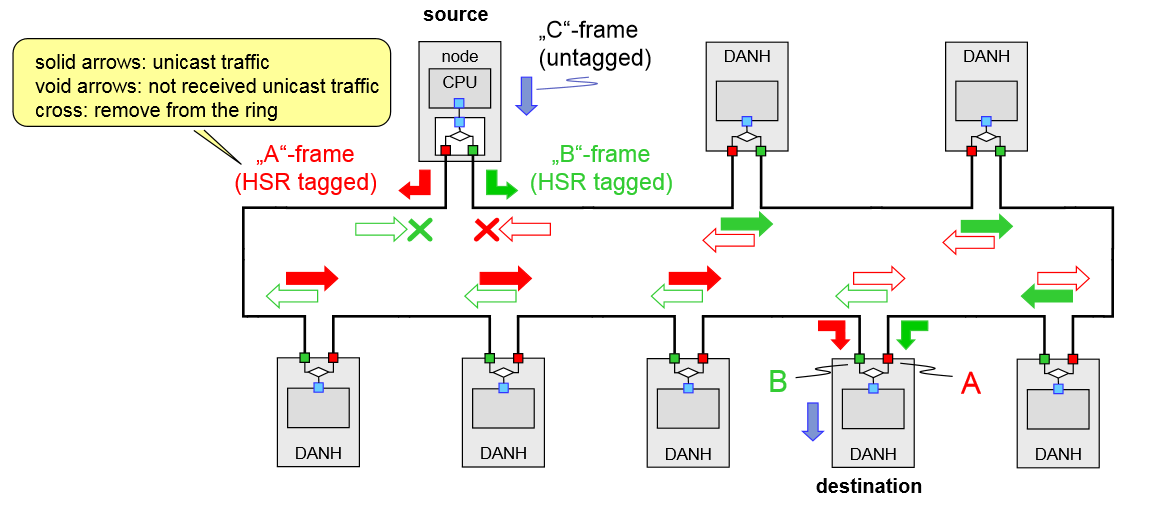


Abbildung 9: HSR Unicast Prinzip [10]

Das High Availability Seamless Redundancy Protokoll 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].

## 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 2 Verbindungen, oder auch Trunks genannt, gebündelt zu einer logischen Verbindung. Die Lasten der Datenübertragung werden in diesem Fall automatisch verteilt sodass es zugleich 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.

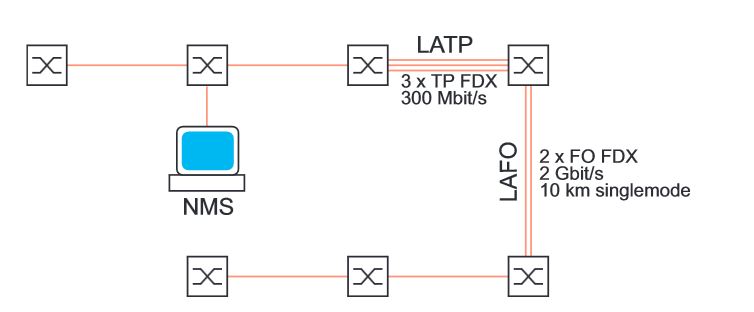


Abbildung 10: Link Aggregation [16]

Es können maximal 8 Trunk Verbindungen miteinander gebündelt werden, wobei eine Verbindung von 4 Optimal wäre. Ein Vorteil hier ist auch, dass die Verbindungen gemixt werden können mit Fiber Kabel und Twisted Pair Kabeln.

## Zeitsynchronisation

Zur Zeitsynchronisation wird das Precision Time Protocol (PTP) angestrebt, welches auch in der IEEE 1588 und in der 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 lokalen Uhren nicht aufeinander abgestimmt, kann es zu unnötigen Wartezeiten kommen oder es könnten Messungen gefälscht werden [18].

Es gibt 2 Möglichkeiten, die lokalen Uhren der Geräte aufeinander abstimmen zu lassen: Man macht eine sogenannte *Offset-correction*, hierbei werden alle Uhren auf die präziseste Uhr „Grandmaster“ eingestellt. Es ist ebenso möglich, eine *Drift-correction* durchzuführen. Hier werden die Zeigergeschwindigkeiten der Uhren angepasst, um dieselbe Zeit an allen Uhren zu erzielen [18].

Die meist verbreitetsten Methoden, eine Zeit 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 aufgeteilt in *Master* und *Slaves*. Der *Master* wird von einer Radio Uhr oder von einem GPS Signal gesteuert und der *Master* synchronisiert anschließend seine Zeit mit der der *Slaves* [18].

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

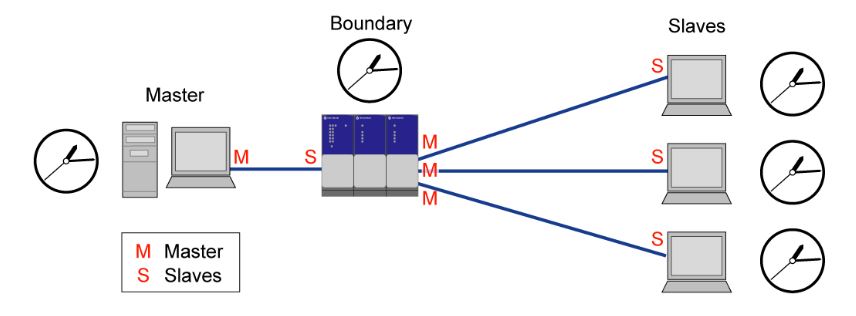


Abbildung 11: PTP Master Slave Prinzip [18]

Die Synchronisation verläuft in 2 Schritten. Im ersten Schritt wird den *Slaves* die Uhrzeit des *Masters* mit einer Synchronisationsnachricht „SYNC message“ mitgeteilt. Vom Absenden 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 2 Schritte erledigt, wurden die Uhrzeiten aufeinander richtig synchronisiert [18].

# Praktischer Teil

Im praktischen Teil der Arbeit wurden die Eigenschaften der Protokolle HSR, PRP und MRP untersucht. Zur 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, die daraus resultierende Stabilität und die Verzögerung bei simuliertem Ausfall einer Switch-Anbindung.

## Vorbereitungen

Bevor der praktische Teil der Arbeit realisiert werden konnte, musste erst 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.

## 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.

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

|  |  |
| --- | --- |
| **Rechner 1+3** | |
| Modell | HP Compaq Elite 8000 Business |
| OS | Linux Ubuntu 15.10 |

Tabelle 2: Verwendete Rechner 1+3

|  |  |
| --- | --- |
| **Rechner 2** | |
| Modell | HP Compaq Elite 8200 Business |
| OS | Linux Ubuntu 15.10 |

Tabelle 3: Verwendeter Rechner 2

## Konfiguration der Switches und PCs

Das grundlegende Netzwerk besteht aus drei PCs, welche alle mit zwei Netzwerken verbunden sind. Beim ersten Netzwerk handelt es sich um das „Management“ Netzwerk, wobei alle drei 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 und wird für jedes getestete Protokoll dementsprechend verändert. 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

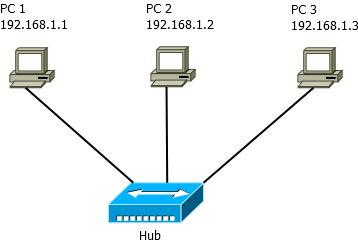


Abbildung 12: Management Netzwerk

|  |  |  |  |
| --- | --- | --- | --- |
| **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. Aufgrund der Simplizität der Testumgebung, welche keine groben Störungen des Netzwerkes verursachen sollte, wurde entschlossen, dass auch das Network Time Protocol für die Messungen geeignet ist.

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 (LACP)

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

## Untersuchung des Media Redundancy Protocols (MRP)

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. In diesem Fall wir 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.

Während den Messungen wird die reguläre Route zehn Mal für einen Zeitraum von 20 Sekunden unterbrochen, sodass der Redundanzmanager dazu gezwungen wird die redundante Route für die Paketübertragung zu verwenden.

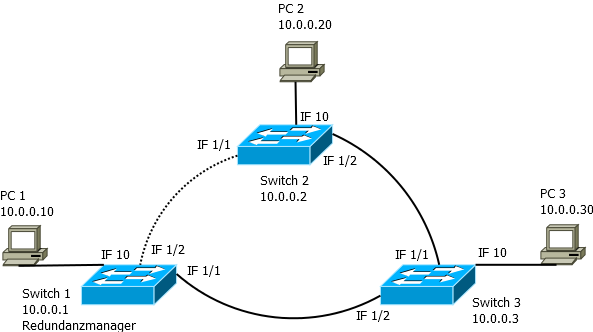


Abbildung 13: MRP Topologie

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Messung** | **Übertragungsdauer in Sekunden** | **Größe der zu übertragenden Pakete in Byte** | **Zeitraum der Unterbrechung/ des Ausfalls** |
| **Messung 1** | 600 | 200 | Sekunde 30 - 50 |
| **Messung 2** | 600 | 700 | Sekunde 30 - 50 |
| **Messung 3** | 600 | 1200 | Sekunde 30 - 50 |

Tabelle 6: MRP Messungen

## Untersuchung des High Availability Seamless Redundancy Protocols (HSR)

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.

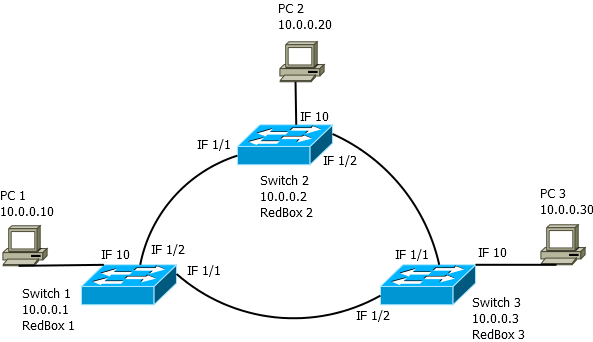


Abbildung 14: HSR Topologie

Die Messungen wurden unter denselben Bedingungen durchgeführt, welche auch bei MRP herrschten.

|  |  |  |  |
| --- | --- | --- | --- |
| **Messung** | **Übertragungsdauer in Sekunden** | **Größe der zu übertragenden Pakete in Byte** | **Zeitraum der Unterbrechung/ des Ausfalls** |
| **Messung 1** | 600 | 200 | Sekunde 30 - 50 |
| **Messung 2** | 600 | 700 | Sekunde 30 - 50 |
| **Messung 3** | 600 | 1200 | Sekunde 30 - 50 |

Tabelle 7: HSR Messungen

## Untersuchung des Parallel Redundancy Protocols (PRP)

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.

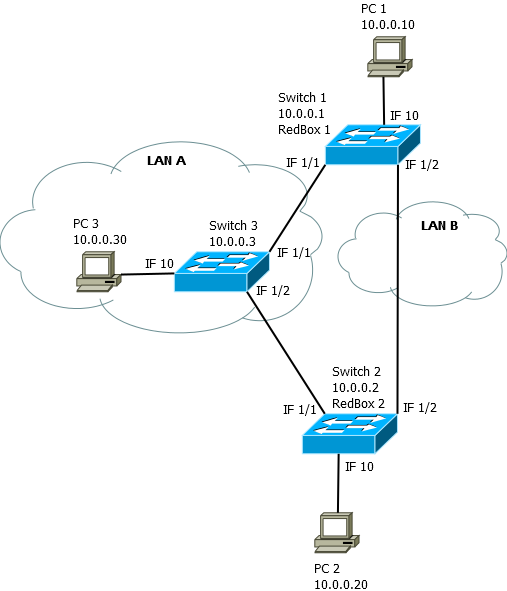


Abbildung 15: PRP Topologie

|  |  |  |  |
| --- | --- | --- | --- |
| **Messung** | **Übertragungsdauer in Sekunden** | **Größe der zu übertragenden Pakete in Byte** | **Zeitraum der Unterbrechung/ des Ausfalls** |
| **Messung 1** | 600 | 200 | Sekunde 30 - 50 |
| **Messung 2** | 600 | 700 | Sekunde 30 - 50 |
| **Messung 3** | 600 | 1200 | Sekunde 30 - 50 |

Tabelle 8: PRP Messungen

# Ergebnisse

Die Messungen wurden alle unter denselben Umständen durchgeführt. Mit Hilfe der Messungsergebnisse wurden anschließend die Anzahl der übertragenen Pakete, die Paketverlustrate, der Jitter, welcher Abweichungen der durchschnittlichen Übertragungszeit beschreibt, und die Dauer der einzelnen Paketsendungen analysiert.

Es wurden, wie bereits im praktischen Teil erwähnt, 3 Messungen durchgeführt mit Szenario A: 200 Bytes / Paket, Szenario B: 700 Bytes / Paket und Szenario C: 1200 Bytes / Paket. Die Messungen betrugen 10 Minuten, wobei bei jeder Messung 10-mal für jeweils 20 Sekunden eine Unterbrechung des Netzwerkes simuliert wurde. In den folgenden Abbildungen werden die Einzelnen Szenarien und deren Paketverzögerung zur gemessenen Zeit dargestellt.

## Messung des Media Redundancy Protocols

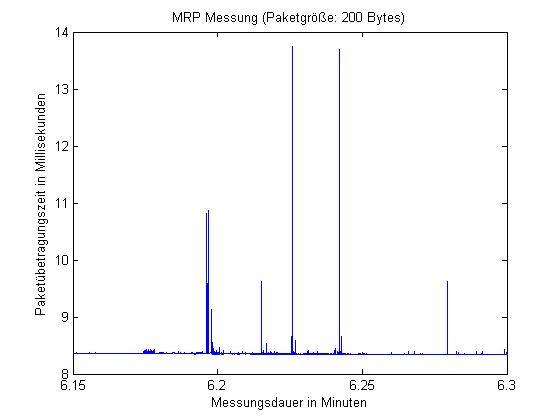
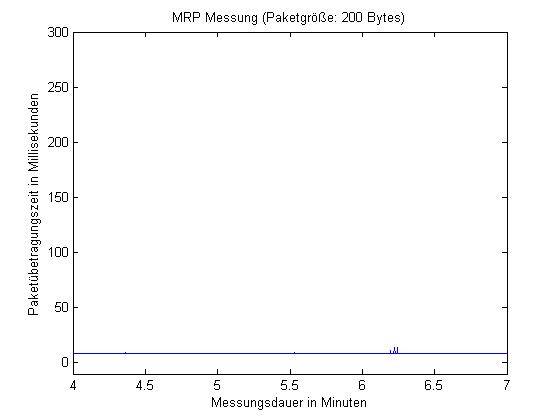
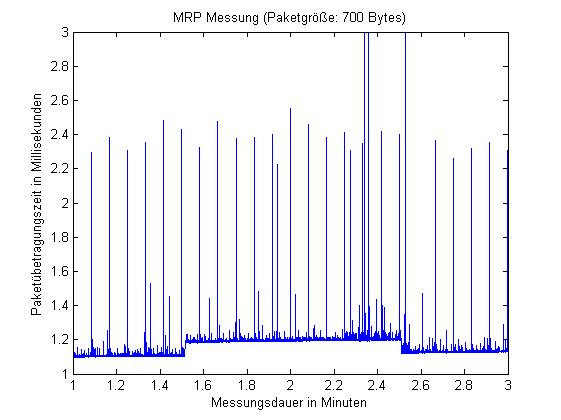
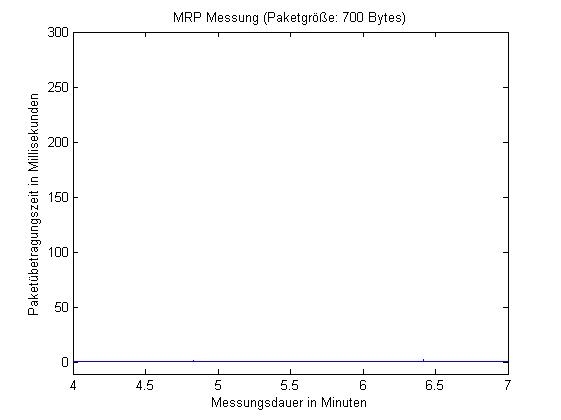


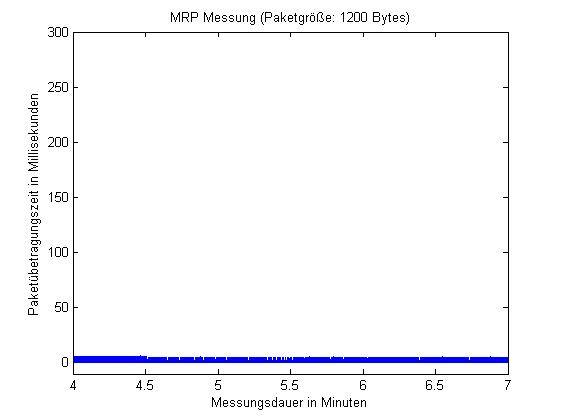
Abbildung 16: MRP Messung Szenario A

Die erste Messung, welche 200 Bytes große Pakete simulierte, ergab eine durchschnittliche Paketübertragungszeit von 8,737 ms. Dabei wurde ein maximaler Jitter von 5,48ms erreicht. Die linke Grafik, welche oberhalb ersichtlich ist, veranschaulicht einen Bereich der Messung, der zum Vergleich mit den anderen Messungen dient. Die rechte Grafik zeigt genau eine Unterbrechung des Netzwerks, welche den Redundanzmanager zum Umschalten der Route zwingt.



1200 Bytes:

Abbildung 17: MRP Messung Szenario B

Die Messung mit 700 Bytes großen Paketen ergab eine Paketübertragungszeit von 1,224 ms. Der maximale Jitter erreichte einen Wert von 6,32 ms. Die Abbildung links zeigt einen Bereich der Messung mit Hilfe der gleichen Skala wie für die anderen Messungen um einen Vergleich zu veranschaulichen. Die rechte Abbildung lässt zwei Unterbrechungen erkennen, wobei der erste bei ca. 1,5 Minuten passiert. Die gewollte Unterbrechung hatte eine Dauer von 20 Sekunden, wobei die Messung bei erneutem Verbinden der unterbrochenen Strecke, bis zur nächsten Unterbrechung auf derselben Strecke weitergeführt wurde. Zur Minute 1,9 wurde die unterbrochene Strecke wieder verbunden und bei 2,5 wurde die nächste Unterbrechung herbeigeführt. Die Sprünge der Paketübertragungszeiten lassen sich anhand der unterschiedlichen Längen der Strecken sichtbar erkennen.

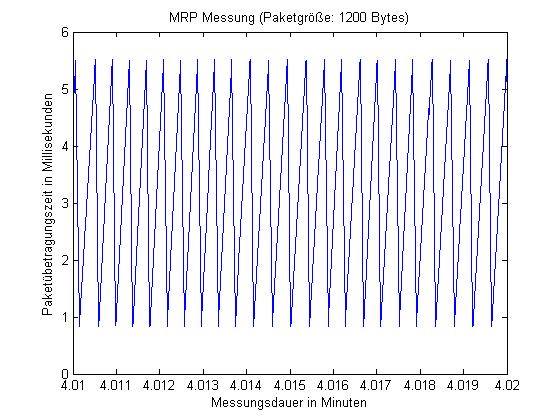


Abbildung 18: MRP Messung Szenario C

Die Messung mit 1200 Bytes Paketgröße stellte auch eine sichtliche Belastung für das MRP Protokoll dar. Die durchschnittliche Übertragungszeit der Pakete betrug 2,943 ms. Ein maximaler Jitter von 5,61 ms wurde festgestellt. Anhand der linken Grafik kann man im Vergleich zu den vorher durchgeführten Messungen eine eher unruhige Übertragung erkennen. Die nähere Betrachtung mit Hilfe der rechten Grafik zeigt deutliche Schwankungen, welche sich über die ganze Messung hinweg ziehen.

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

Tabelle 9: Anzahl der übertragenen und verlorenen Pakete mit MRP

|  |  |  |  |
| --- | --- | --- | --- |
| **Unterbrechungen** | **Anzahl der verlorenen Pakete bei 200 Byte** | **Anzahl der verlorenen Pakete bei 700 Byte** | **Anzahl der verlorenen Pakete bei 1200 Byte** |
| 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 10: Paketverlustraten der einzelnen Unterbrechungen mit MRP

Anhand der Tabelle 9 kann man die Gesamtanzahl der übertragenen und verlorenen Pakete der jeweiligen Messungen erkennen. Die Tabelle 10 beschreibt die genauen Paketverluste pro Unterbrechung, wobei die Messungen mit 200 Bytes einen Verlustwert von 0,313% ergaben. Die Messung mit 700 Bytes großen Paketen ergab 0,299% Verlust, und bei 1200 Bytes Paketen war betrug die Verlustrate 0,294%.

## Messung des Parallel Redundancy Protocols

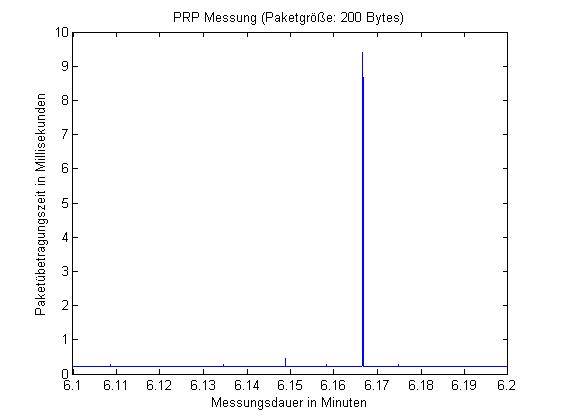
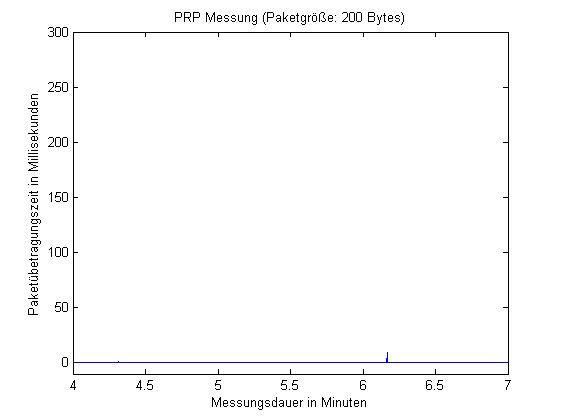


Abbildung 19: PRP Messung Szenario A

Die Messung des PRP Protokolls mit 200 Byte großen Paketen ergab eine durchschnittliche Paketübertragungszeit von 0,219 ms. Man kann anhand der Grafik erkennen, dass dieser Wert bis auf zwei Mal vorkommende Abweichungen trotz Unterbrechungen ziemlich konstant bleibt. Der maximale Jitter erreichte einen Wert von 9,35ms. Die Paketverlustrate dieser Messung betrug wie erwartet 0%.

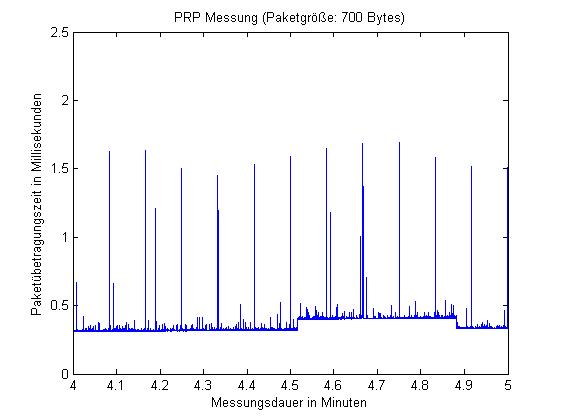
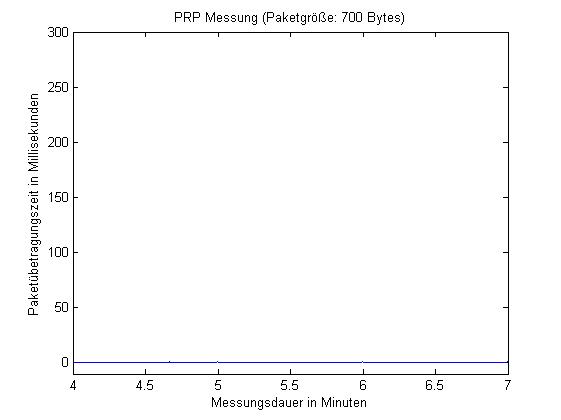


Abbildung 20: PRP Messung Szenario B

Die Messung mit 700 Bytes lässt anhand der linken Grafik, welche zum Vergleich für die anderen Messungen dient, keine größeren Verzögerungen erkennen. Die Paketübertragungszeit beträgt im Schnitt 0,369 ms. Der Jitter erreichte einen maximalen Wert von 1,23 ms. Die rechte Grafik stellt einen genaueren Bereich dar, in welchem eine Unterbrechung durchgeführt wurde. Hier kann man erkennen, dass die Route geändert wird, im Gegensatz zum MRP jedoch, bei erneuter Verfügbarkeit der eigentlich benützten Strecke, wieder auf diese zurückgesprungen wird. Auch bei dieser Messung gingen keine Pakete verloren.

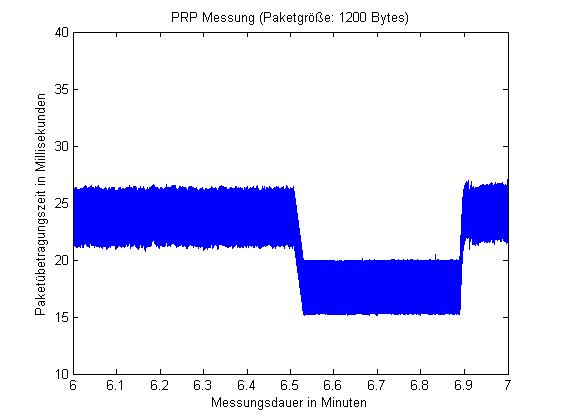
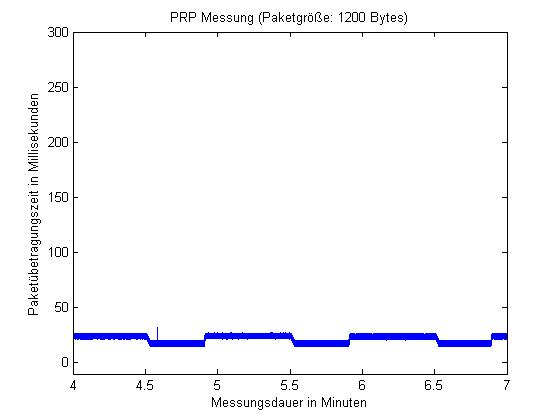


Abbildung 21: PRP Messung Szenario C

Eine belastende Messung mit Paketgrößen von 1200 Bytes wurde ebenfalls durchgeführt. Hierbei betrug die Zeit welche zur Übertragung eines Paketes benötigt wurde im Schnitt 21,732 ms. Die maximale Abweichung des durchschnittlichen Übertragungswertes betrug 10,547 ms. Bei dieser Messung lassen sich bereits anhand der linken Abbildung, welche weniger Details veranschaulicht, die Unterbrechungen erkennen. Die Grafik rechts lässt die Unterbrechung der Messung deutlich erkennen, wobei die 1200 Bytes Pakete eine große Belastung darstellten.

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

Tabelle 11: Anzahl der übertragenen und verlorenen Pakete mit PRP

Die Messungen des PRP Protokolls ergaben eine schnelle Paketübertragung als das MRP Protokoll. Wie erwartet ergaben die ersten beiden Messungen mit 200 und 700 Bytes Paketen eine Verlustrate von 0%. Die Messung mit 1200 Bytes ergab jedoch einen Verlustwert von 0,613% der Pakete, wobei dies eigentlich nicht passieren dürfte.

## Messung des High Availability Seamless Redundancy Protocols

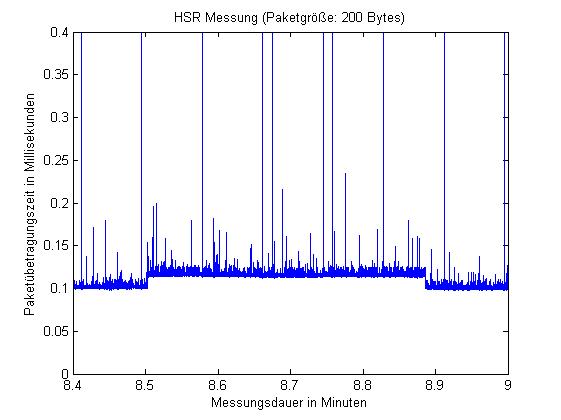
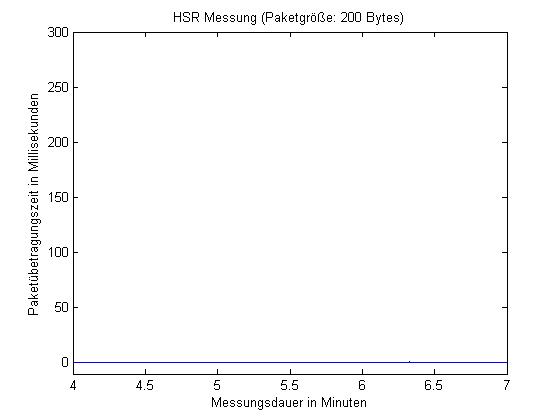
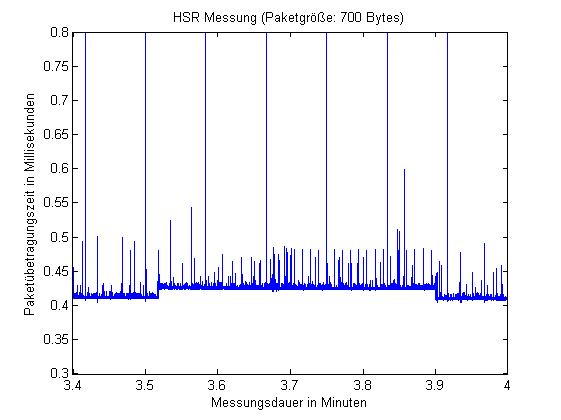
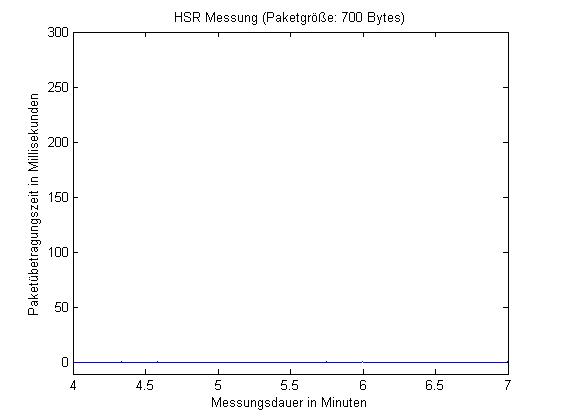


Abbildung 22: HSR Messung Szenario A

Die Messung des HSR Protokolls mit 200 Bytes ergab eine erneute Geschwindigkeitssteigerung im Vergleich zum PRP und MRP Protokoll. Mit einer durchschnittlichen Übertragungszeit von 0,1423 ms ist es eindeutig am schnellsten, wobei ein maximaler Jitterwert von 1,444 ms vorkam. Erneut lässt sich anhand der rechten Grafik eine Unterbrechung erkennen. Im Gegensatz zu PRP sendet das HSR Protokoll nicht in zwei unabhängige Netzwerke sonder in beide Richtungen eines Netzwerkes. Hierbei wird auch wie beim PRP immer wieder die schnellere Strecke bevorzugt, sobald diese verfügbar ist. Ein Paketverlust von null Paketen konnte auch hier, wie erwartet, festgestellt werden.

Abbildung 23: HSR Messung Szenario B



1200 Bytes

Die Messung mit 700 Bytes benötigte pro Paket durchschnittlich 0,427 ms für die Übertragung. Der maximale Jitter erreichte einen Wert von 8,385 ms. Die rechte Abbildung verdeutlicht erneut eine Unterbrechung, wobei auch während dieser Messung die schnellere Route bevorzugt wurde. Auch bei dieser Messung wurden keine Pakete verloren.

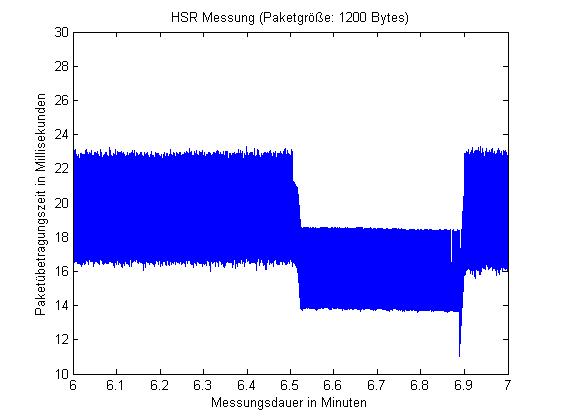
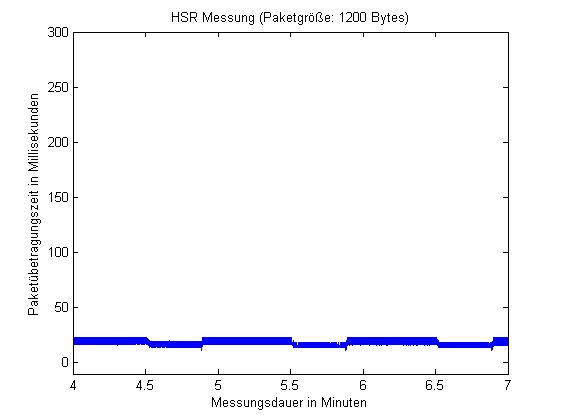


Abbildung 24: HSR Messung Szenario C

Wie zuvor beim PRP, wurde auch beim HSR Protokoll eine belastende Messung mit 1200 Bytes durchgeführt. Hierbei wurde eine durchschnittliche Übertragung von 18,315 ms gemessen. Der maximale Jitter betrug 17,816 ms. Auch während dieser Messung, wie zuvor bei PRP, kam es zu einem unerwarteten Paketverlust von 1,11%.

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

Tabelle 12: Anzahl der übertragenen und verlorenen Pakete mit HSR

Anhand der Tabelle 12 kann man sehen, 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 zustande kam. Wie zuvor bei der Belastungsmessung des PRP Protokolls, wird davon ausgegangen, dass es zu einer Überlastung aufgrund der Paketgrößen gekommen ist.

Abbildung 25: Paketverlustrate der Messungen

Die Abbildung veranschaulicht anschließend die erzielten Ergebnisse der Messungen, wobei alle Messungen samt ihrer Paketverlustraten in % dargestellt werden.

## 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 ergab schnelle Paketübertragungszeiten und geringe Jitter-Werte.

Das Media Redundancy Protokoll erweist sich bezüglich der Stabilität und Ausfallsicherheit als weniger geeignet, da es in jeder Messung, bei Paketen der Größe 200 Bytes, 700 Bytes und 1200 Bytes, zu einem Paketverlust kam. Das Parallel Redundancy Protokoll ergab ähnliche Ergebnisse wie das HSR Protokoll. Obwohl in diesem Testszenario die Ergebnisse des PRP den des HSR nahe kommen, und diese manchmal 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 der Industrie in Betracht gezogen, muss jedoch bedacht werden, dass PRP und HSR jeweils eine hohe Anzahl an Redundanz mit sich bringen, wobei MRP mit zwei zur Verfügung stehenden Pfaden weniger überflüssige Informationen bereitstellt. 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 sogar mehr Paketverlust aufweisen als das MRP Protokoll unter denselben Voraussetzungen. 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 verdeutlichen. Werden somit die Messungen der kleineren Pakete für den Vergleich in Betracht gezogen, wirken das HSR und PRP Protokoll am überzeugendsten, wobei das PRP Protokoll, wie bereits erwähnt, vermutlich aufgrund der geringen Netzwerkstrukturen, nahezu ähnliche Ergebnisse lieferte.

# 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. |

# Anhang

## Switch Konfigurationen

### 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

### 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

### 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

## 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 Programm zum rausfiltern der Paket Verluste

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.Scanner;  
  
public class BAC1 {

    public static void main(String[] args) throws FileNotFoundException, IOException {  
        // TODO code application logic here  
          
        ArrayList<Long> first = new ArrayList<>();  
        ArrayList<Long> second = new ArrayList<>();  
        String[] splitted;  
        ArrayList<Long> breaks = new ArrayList<>();  
        ArrayList<Long> breaks\_pos = new ArrayList<>();  
          
        int index = 0;  
  
      BufferedReader in = new BufferedReader(new FileReader("/Users/Lisa/Documents/NetBeansProjects/BAC1/src/bac1/PRP\_200bytes.dat"));  
      String str;  
      while ((str = in.readLine()) != null) {  
             //System.out.println(str);  
             splitted = str.split("\t");  
             first.add(Long.parseLong(splitted[0]));  
             second.add(Long.parseLong(splitted[1]));  
      }  
       in.close();  
        int file\_size = first.size();  
       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);  
           }  
           }  
       }  
       for(int i=0;i<breaks.size();i++){  
        System.out.println("Pos: " + breaks\_pos.get(i));  
        System.out.println(breaks.get(i));  
    }  
    }  
}

Script : Java Code zur Ermittlung des Paketverlustes