



**School of  
Engineering**

InES Institute of  
Embedded Systems

## **Projektarbeit Informatik**

### **PA14 wlan 1**

Performance-Evaluation

Ethernet für Echtzeit-Datenerfassung

---

**Autoren**

Mauro Guadagnini (guadamau@students.zhaw.ch)

Prosper Leibundgut (leibupro@students.zhaw.ch)

---

**Hauptbetreuung**

Hans Weibel (wlan@zhaw.ch)

---

**Datum**

19.12.2014

# Abstract

Blubb blubb blaa brrrrr, ...

# Inhaltsverzeichnis

<b>Abstract</b>	<b>2</b>
<b>Vorwort</b>	<b>6</b>
Ausgangslage . . . . .	6
 <b>I. Vorarbeiten, Planung</b>	 <b>7</b>
<b>1. Aufgabenstellung</b>	<b>8</b>
1.1. Anforderungen . . . . .	8
1.1.1. Modell für HSR-Knoten erweitern . . . . .	8
1.1.1.1. Anforderung #1.1 . . . . .	8
1.1.1.2. Anforderung #1.2 . . . . .	8
1.1.1.3. Anforderung #1.3 . . . . .	8
1.1.1.4. Anforderung #1.4 . . . . .	9
1.1.1.5. Anforderung #1.5 . . . . .	9
1.1.2. Lastmodell beschreiben und implementieren . . . . .	9
1.1.2.1. Anforderung #2.1 . . . . .	9
1.1.2.2. Anforderung #2.2 . . . . .	9
1.1.2.3. Anforderung #2.3 . . . . .	9
1.1.3. Simulationen durchführen und Resultate interpretieren . . . . .	9
1.1.3.1. Anforderung #3.1 . . . . .	9
 <b>2. Pflichtenheft</b>	 <b>10</b>
2.1. Ausgangslage . . . . .	10
2.1.1. Arbeitspakete und Zeitplanung . . . . .	10
2.2. Grundlagen HSR . . . . .	10
2.2.1. Funktionsweise . . . . .	10
2.2.1.1. Duplikat Erkennung . . . . .	11
2.2.2. Gerätetypen . . . . .	11
2.2.2.1. DANH (Doubly Attached Node HSR) . . . . .	12
2.2.2.2. RedBox (Redundancy Box) . . . . .	12
2.2.2.3. QuadBox . . . . .	12
2.2.2.4. VDAN (Virtual Doubly Attached Node) . . . . .	12

2.2.2.5. DANP (Double Attached Node PRP) . . . . .	12
<b>II. Engineering</b>	<b>13</b>
<b>3. Konzeptpapier</b>	<b>14</b>
3.1. Einleitung . . . . .	14
3.2. Projekt-Risikoanalyse . . . . .	14
<b>III. Realisierung</b>	<b>15</b>
<b>4. Umsetzung der Software/Spezifikation</b>	<b>16</b>
4.1. Diagramme und Beschrieb . . . . .	16
4.1.1. Beispiel für Source-Code-Listing . . . . .	16
<b>5. Testing</b>	<b>17</b>
5.1. Teststrategie . . . . .	17
5.1.1. Blackbox-Test . . . . .	17
5.1.2. Whitebox-Tests . . . . .	17
5.1.3. Performance-Tests . . . . .	17
5.2. Testkriterien . . . . .	17
5.3. Schlüsselfunktionen . . . . .	17
5.4. Testumgebung . . . . .	18
5.5. Kurzbeschreibung der Tests . . . . .	18
5.5.1. Anforderung #1.1 . . . . .	18
5.5.1.1. Systemanforderungen/Voraussetzungen . . . . .	18
5.5.1.2. Testdaten . . . . .	18
5.5.1.3. Resultat . . . . .	18
5.5.1.4. Interpretation . . . . .	18
5.5.1.5. Weiteres Vorgehen . . . . .	18
5.6. Testabweichungen . . . . .	19
5.7. Zusammenfassung der Testresultate . . . . .	19
<b>IV. Anhang</b>	<b>20</b>
<b>Fremdprodukte</b>	<b>21</b>
OMNeT++ . . . . .	21
<b>Bedienungsanleitung</b>	<b>22</b>
Systemvoraussetzungen . . . . .	22
Kompilierung/Installation . . . . .	22

Verwendung . . . . .	22
Bekannte Probleme . . . . .	22
Deinstallation . . . . .	22
<b>Glossar</b>	<b>23</b>
<b>Abbildungsverzeichnis</b>	<b>24</b>
<b>Tabellenverzeichnis</b>	<b>25</b>
<b>Listings</b>	<b>26</b>
<b>Literaturverzeichnis</b>	<b>27</b>

# Vorwort

## Ausgangslage

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

# **Teil I.**

## **Vorarbeiten, Planung**

# 1. Aufgabenstellung

Durch das Institute of Embedded Systems der ZHAW wurde den Autoren am 24. September 2014 eine Aufgabenstellung[3] zugestellt, welche die nachfolgenden Hauptanforderungen umfasst:

## 1.1. Anforderungen

### 1.1.1. Modell für HSR-Knoten erweitern

Das betrachtete Netzwerk ist ein HSR-Ring. Die bestehende Simulationsumgebung [1] soll so erweitert bzw. angepasst werden, dass folgende Funktionen/Mechanismen simuliert werden können:

#### 1.1.1.1. Anforderung #1.1

Der Knoten soll zwei Prioritäten unterstützen, d.h. zwei Warteschlangen pro Interface bewirtschaften.

#### 1.1.1.2. Anforderung #1.2

Der Knoten soll Interspersing Express Traffic (IET) unterstützen, d.h. Express Frames können die aktuell ablaufende Übertragung eines Frames unterbrechen.

#### 1.1.1.3. Anforderung #1.3

Der in den Ring einflussende Traffic kann limitiert werden.



#### **1.1.1.4. Anforderung #1.4**

Die Vortrittsregeln bezüglich der im Ring zirkulierenden Frames und den Frames, die in den Ring einfließen, können variiert werden (z.B. «zirkulierende Frames haben immer Vortritt» oder «minimaler Zufluss wird garantiert»).

#### **1.1.1.5. Anforderung #1.5**

Der Knoten implementiert ein Zeitschlitzverfahren, welches dem zeitkritischen Traffic und dem Bulk Traffic je eine Phase zuordnet.

### **1.1.2. Lastmodell beschreiben und implementieren**

Das durch die Anwendung generierte Verkehrsaufkommen ist zu studieren und zu beschreiben. Lastgeneratoren sollen implementiert werden, die das Verkehrsaufkommen für die Simulation generieren durch die Überlagerung von Strömen mit folgender Charakteristik:

#### **1.1.2.1. Anforderung #2.1**

Lastgenerator mit konstanter Framerate.

#### **1.1.2.2. Anforderung #2.2**

Lastgenerator mit zufälliger zeitlicher Verteilung der Frames.

#### **1.1.2.3. Anforderung #2.3**

Lastgenerator, der spontane Einzelmeldungen erzeugt.

### **1.1.3. Simulationen durchführen und Resultate interpretieren**

Das Zeitverhalten der verschiedenen Weiterleitungsvarianten soll durch entsprechende Simulationen ermittelt werden. Die Resultate sind zu vergleichen und zu interpretieren.

#### **1.1.3.1. Anforderung #3.1**

Simulationsresultate vergleichen und interpretieren.

## 2. Pflichtenheft

### 2.1. Ausgangslage

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

#### 2.1.1. Arbeitspakete und Zeitplanung

Blubb, blaa blaa

### 2.2. Grundlagen HSR

Die Grundlagen zum Beschrieb und zur Funktionsweise des HSR-Protokolls wurden alle der bestehenden Vertiefungsarbeit [1] entnommen.

HSR ist ein Redundanzprotokoll für Ethernet basierte Netzwerke. Im Vergleich zu weit verbreiteten Redundanzprotokollen wie z.B. Rapid Spanning Tree, zeichnet sich HSR dadurch aus, dass auch im Fehlerfall keine Pakete verloren gehen. Im Gegensatz zu HSR braucht Rapid Spanning Tree im Fehlerfall einige Millisekunden bis zu mehreren Sekunden, bis das Netzwerk wieder Daten übermitteln kann. Für gewisse industrielle Anwendungen wie beispielsweise in der Automatisierung sind solche Unterbrüche nicht tolerierbar.

#### 2.2.1. Funktionsweise

In einem HSR Netzwerk besitzen alle Geräte mindestens zwei Ethernet Ports. Eine Quelle sendet jedes Paket auf beide Ports. Im fehlerfreien Zustand kommen immer zwei oder mehr Pakete

beim Ziel an. Anhand einer Sequenznummer und der Quelladresse werden Duplikate erkannt und verworfen. Falls ein Link ausfällt, geht nur eines der Pakete verloren und die anderen erreichen trotzdem noch das Ziel. Auf diese Weise wird verhindert, dass es im Fehlerfall zu einem Paketverlust kommt.

### 2.2.1.1. Duplikat Erkennung

Die Duplikat Erkennung ist ein zentrales Element von HSR. Sie sorgt einerseits dafür, dass jedes Frame nur einmal zu den höheren Protokollschichten weitergegeben wird und andererseits sorgt sie dafür, dass die Frames von den Netzwerkknoten nur einmal über jeden Netzwerkport versendet werden.

Die zweite Aufgabe ist kritisch. Versagt die Duplikat Erkennung, werden Frames unendlich lange im Netzwerk weitergeleitet. Je nach Anzahl solcher Frames kann das Netzwerk dadurch völlig ausgelastet werden, so dass keine Nutzdaten mehr übertragen werden können. Die Duplikat Erkennung benutzt die Absender-MAC-Adresse und eine 16-Bit-Sequenznummer, welche vom Sender für jedes Frame um eins inkrementiert wird. Ein Knoten muss für jedes Frame 64 Bit Informationen speichern, solange sich das Frame im HSR Netzwerk befindet, um Duplikate erkennen zu können.

### 2.2.2. Gerätetypen

Die folgende Abbildung zeigt ein Beispielnetzwerk mit allen Gerätetypen, die mit einem HSR Netzwerk verbunden werden können.

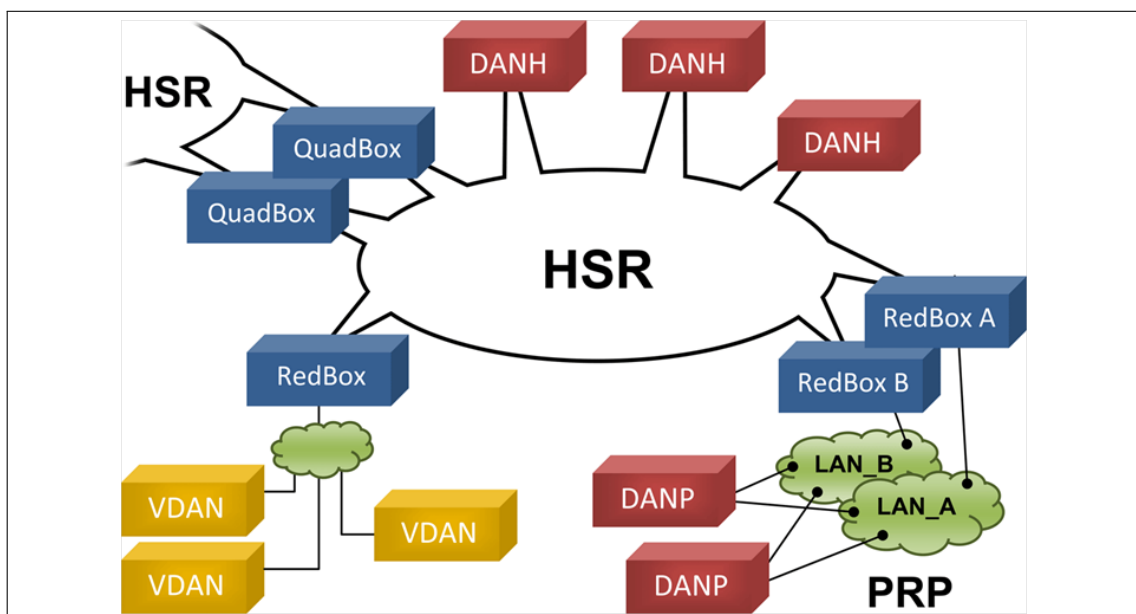


Abbildung 2.1.: HSR-Ring mit allen möglichen Gerätetypen[2]

#### 2.2.2.1. DANH (Doubly Attached Node HSR)

Ein DANH ist ein direkt und redundant an das HSR Netzwerk angeschlossener Netzwerkknoten mit zwei Ethernet Ports.

#### 2.2.2.2. RedBox (Redundancy Box)

Eine RedBox wird benötigt, um das HSR Netzwerk mit anderen Netzwerken zu verbinden. Mit RedBoxen können entweder Standard-Ethernet, PRP oder andere HSR Netzwerke an ein HSR Netzwerk gekoppelt werden.

#### 2.2.2.3. QuadBox

Quadboxen sind Geräte mit vier Ethernet Ports. Sie werden benötigt, um zwei HSR Netzwerke miteinander zu verbinden. Es sind mindestens zwei Quadboxen nötig, um zwei Netze redundant zu verbinden, da sonst ein Single Point of Failure entsteht.

#### 2.2.2.4. VDAN (Virtual Doubly Attached Node)

VDAN sind Knoten, die über eine Redbox mit dem redundanten Netzwerk verbunden sind. VDAN kennen das HSR Protokoll nicht und sind auch nicht redundant ans Netzwerk angeschlossen, können aber trotzdem transparent mit jedem redundanten Gerät im HSR Netzwerk kommunizieren.

#### 2.2.2.5. DANP (Double Attached Node PRP)

DANP sind redundant angeschlossen Knoten, welche das PRP Protokoll verwenden. Das PRP Protokoll arbeitet grundsätzlich mit dem gleichen Funktionsprinzip wie HSR, es gibt aber einige Unterschiede, wie beispielsweise das Frameformat. Durch die Ähnlichkeit können PRP Netzwerke über RedBoxen mit HSR Netzwerken gekoppelt werden. Die Kommunikation zwischen DANP und DANH ist transparent und redundant. Ein DANP muss das HSR Protokoll für die Kommunikation mit einem DANH nicht kennen.

# Teil II.

# Engineering

## 3. Konzeptpapier

### 3.1. Einleitung

Dieses Dokument geht konkret auf die funktionalen Anforderungen, welche im Pflichtenheft gestellt werden ein und versucht anhand von verschiedenen Varianten die geeignete Lösung für deren Umsetzung herauszufiltern. Zunächst werden mögliche Projektrisiken aufgezeigt.

### 3.2. Projekt-Risikoanalyse

#### Risikoidentifikation

Risiko	Mögliche Massnahmen
wenn die Polizei vorbei fährt ...	<ul style="list-style-type: none"><li>• ... halt ich erst mal an</li><li>• ja</li><li>• ... nimm ne Ziese aus der Schachtel</li><li>• ja</li></ul>

Tabelle 3.1.: Projekt-Risikoanalyse

## **Teil III.**

# **Realisierung**

## 4. Umsetzung der Software/Spezifikation

### 4.1. Diagramme und Beschrieb

Blaaaaaaaaa ...

#### 4.1.1. Beispiel für Source-Code-Listing

Listing 4.1: Hello World in C.

```
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int main( void )
5  {
6      printf( "Hello, World!\n" );
7      return EXIT_SUCCESS;
8  }
```



## 5. Testing

### 5.1. Teststrategie

Test me, I'm famous ...

#### 5.1.1. Blackbox-Test

Test me, I'm famous ...

#### 5.1.2. Whitebox-Tests

Test me, I'm famous ...

#### 5.1.3. Performance-Tests

Test me, I'm famous ...

### 5.2. Testkriterien

Test me, I'm famous ...

### 5.3. Schlüsselfunktionen

Test me, I'm famous ...

## 5.4. Testumgebung

Test me, I'm famous ...

## 5.5. Kurzbeschreibung der Tests

Gegenüberstellung der Anforderungen aus der Aufgabenstellung und den «Abnahmetests».

### 5.5.1. Anforderung #1.1

TicToc TicToc TicToc ...

#### 5.5.1.1. Systemanforderungen/Voraussetzungen

TicToc TicToc TicToc ...

#### 5.5.1.2. Testdaten

TicToc TicToc TicToc ...

#### 5.5.1.3. Resultat

TicToc TicToc TicToc ...

#### 5.5.1.4. Interpretation

TicToc TicToc TicToc ...

#### 5.5.1.5. Weiteres Vorgehen

TicToc TicToc TicToc ...



# **Teil IV.**

# **Anhang**

# Fremdprodukte

## OMNeT++

OMNeT++ ist ein C++-Framework, welches es erleichtert, Netzwerke und all deren Komponenten mit einem sehr hohen Detaillierungsgrad zu modellieren und den Netzwerk-Datenverkehr zu simulieren. Die Simulationen können grafisch dargestellt werden. Zur Auswertung der Simulationen steht eine grosse Auswahl an verschiedenen Diagrammtypen zur Verfügung. In dieser Projektarbeit wird die Version 4.5 verwendet.

# Bedienungsanleitung

Blubb blubb blubb ...

## Systemvoraussetzungen

Das Problem in dem System ja ...

## Kompilierung/Installation

<3 gcc gcc gcc gcc gcc gcc gcc <3

## Verwendung

GUI starten, run Simulation bla bla bla ...

## Bekannte Probleme

- Wir raten von einer Verwendung unter Windows ab. Windows ist ein GUI mit einem Betriebssystem. Und nicht ein Betriebssystem mit optionaler Wahl eines GUIs.
- Blaaaaaaaaaaaaa

## Deinstallation

shred -n 27000 OMNeT++

## Glossar

**HSR** High-availability Seamless Redundancy. Redundanzprotokoll für Ethernet basierte Netzwerke. HSR ist für redundant gekoppelte Ringtopologien ausgelegt. Die Datenübermittlung innerhalb eines HSR-Rings ist im Fehlerfall gewährleistet, wenn eine Netzwerkschnittstelle ausfallen sollte.

# Abbildungsverzeichnis

2.1. HSR-Ring mit allen möglichen Gerätetypen[2] . . . . .	11
--	----



# Tabellenverzeichnis

3.1. Projekt-Risikoanalyse . . . . .	14
--------------------------------------	----

# Listingverzeichnis

4.1. Hello World in C. . . . .	16
--------------------------------	----

## Literaturverzeichnis

- [1] GEMPERLI, ALFRED: *Vertiefungsarbeit: HSR Simulation mit OMNeT++*. Technischer Bericht, Institute of Embedded Systems, ZHAW School of Engineering, Juli 2011.
- [2] INSTITUTE OF EMBEDDED SYSTEMS: *HSR-Konzept* @<http://ines.zhaw.ch/de/engineering/institute-zentren/ines/forschung-und-entwicklung/high-availability/hsr.html>, September 2014.
- [3] WEIBEL, HANS: *PA14\_wlan\_1: Projektarbeit im Fachgebiet Kommunikation*. Aufgabenstellung, September 2014.