



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung - Anwendungen 2 SS 2010

Hermann Dieumo Kenfack

Erzeugung von charakteristischen
Last-Profilen zur simulationsgestützten
Analyse von TTEthernet: Related Works

Hermann Dieumo Kenack

Thema der Ausarbeitung

Erzeugung von charakteristischen Last-Profilen zur simulationsgestützten Analyse von TTEthernet: Related Works

Kurzzusammenfassung:

Ein TTEthernet-Netzwerk ist ein Switched-Ethernet-Netzwerk mit Erweiterungen für Echtzeitanforderungen. Das Designen solch eines Netzwerkes und seiner Dienste in einer Simulationsumgebung ist eine anspruchsvolle Aufgabe. Bei ihrer Lösung ist das Verstehen von Methoden zur (realistischen) Traffic-Generierung und Analyse unabdingbar. Diese Arbeit stellt zwei Herangehensweise vor.

Betreuende Prüfer

Prof. Dr. Luck, Kai von; Prof. Dr. Klemke, Gunter

Betreuer

Prof. Dr. Korf, Franz; Prof. Dr. Schmidt, Thomas

Inhaltsverzeichnis

1	Einleitung	4
1.1	Problemstellung und Ziele	4
1.2	Stand meiner Arbeiten.....	5
2	TTEthernet	6
2.1	Allgemeines	6
2.2	Traffic-Klassen	6
3	RFC 2544 Benchmarking-Methodik	8
3.1	Versuchsaufbauten.....	8
3.2	Test-Durchführung Randbedingungen	8
3.3	Back-to-back (burst) Test.....	9
3.4	Durchsatz Test	10
3.5	Frame- Verlustrate Test.....	11
3.6	Bewertung.....	11
4	Analytische Traffic-Modelle	12
4.1	Einfache Modellierung.....	12
4.2	Poissonverteilung.....	12
4.3	Selbstähnlichkeit.....	13
5	Zusammenfassung und Ausblick	14
5.1	Zusammenfassung.....	14
5.2	Ausblick.....	14
	Abkürzungsverzeichnis	15
	Abbildungsverzeichnis	16
	Literaturverzeichnis	17

1 Einleitung

In einem Kraftfahrzeug steigt die Anzahl der eingesetzten elektronischen Systeme immer weiter an. Diese müssen miteinander kommunizieren, da ihre Funktionen systemübergreifend benötigt werden. So wird beispielsweise die Fahrgeschwindigkeit im ESP (Elektronisches Stabilitätsprogramm) für die Fahrdynamikregelung, im Motormanagement für die automatische Geschwindigkeitsregelung und im Navigationssystem benötigt (Mischo, et al., 2007, S. 16). Moderne Kraftfahrzeuge verfügen mehr und mehr über informations- und unterhaltungselektronische Komponenten. Dadurch entsteht ein steigender Bedarf an breitbandigen Kommunikationsverbindungen. An die elektronischen Systeme werden unterschiedliche Anforderungen an das Zeitverhalten gestellt (harte, weiche und keine Echtzeitanforderung). Um diese Anforderungen zu erfüllen, wird momentan die Vernetzung von Fahrzeugkomponenten in Domänen (Funktionsbereiche) unterteilt. Für jede Domäne wird ein spezielles Bussystem eingesetzt (z. B. LIN, CAN, MOST, etc.). Eine Anwendung wird einer Domäne zugeordnet, die am besten ihren Anforderungen entspricht. Die Bussysteme haben unterschiedliche Protokolle, sodass eine Übersetzung für die Kommunikation zwischen diesen erforderlich ist. Diese wird anhand sogenannter Gateways realisiert. Die verschiedenen Bussysteme machen das Fahrzeugnetzwerk kompliziert und damit in der Entwicklung teuer und unflexibel. Wünschenswert ist ein Backbone, das die schrittweise Konsolidierung dieser Bussysteme ermöglicht. Dafür ist TTEthernet ein guter Kandidat. TTEthernet (Steiner, 2008) ist eine Echtzeit-Variante des Standard Ethernet-Protokolls.

1.1 Problemstellung und Ziele

Es gibt mittlerweile viele Veröffentlichungen, Firmen und Institutionen, die den Fokus auf die Thematik TTEthernet gerichtet haben (TTTech), (Steiner, 2008), (TU Wien - Institut für Technische Informatik). Jedoch keine veröffentlichten Arbeiten, die seine Eignung zur Vernetzung von Kraftfahrzeug-Komponenten bzw. als Backbone zur schrittweisen Konsolidierung der vorhandenen Bussysteme zeigt. Außerdem gibt es bisher keine Werkzeuge zur Modellierung, Simulation und Analyse von TTEthernet-Netzwerken. Aus diesem wurden mehrere Ziele zur Entwicklung einer TTEthernet-Simulationsumgebung definiert.

- Entwicklung eines TTEthernet-Simulators basierend auf dem OMNeT++-INET-Framework (OMNeT++ Community2). Dabei soll der TTEthernet-Protokollstack designt und implementiert, charakteristische Topologien erzeugt und Konzepten für die Konfiguration des Netzwerks entworfen werden. Weiterhin sollen charakteristische Datenströme generiert, erfasst und analysiert werden.
- Entwurf von Modellen für typische Kraftfahrzeug-Netzwerke, welche aus einem Architektur- und Traffic-Modell bestehen.

Der TTEthernet-Protokollstack soll auf der TTEthernet Spezifikation (Steiner, 2008) von TTTech basieren. Diese wurde zur Standardisierung bei der Society of Automotive Engineers (SAE - AS-2D Time Triggered Systems and Architecture Committee, 2009) eingereicht.

1.2 Stand meiner Arbeiten

Der aktuelle Schwerpunkt meiner Arbeiten im Kontext Analyse und Bewertung von TTEthernet im Automobil-Kontext an der HAW Hamburg liegt in der Entwicklung der auf OMNeT++ basierenden TTEthernet-Simulationsumgebung (oder TTEthernet-Simulator).

In dieser Hinsicht wurden einleitende Arbeiten in AW1 erbracht. Dabei wurde ein Überblick über die momentan eingesetzten Techniken zur Vernetzung von Kraftfahrzeug-Komponenten und deren Einschränkungen gegeben. Weiterhin wurden die Anforderungen vorgestellt, die an Bussysteme im Auto gestellt werden. Darüber hinaus wurde die Motivation für den Einsatz von Ethernet gegeben und die Umgebung (OMNeT++) sowie das Framework (INET Framework) zur Entwicklung des TTEthernet-Simulators vorgestellt. Als weiteres Thema von AW1 war die allgemeine Beschreibung von TTEthernet und seiner verschiedenen Traffic-Klassen. Dieses Thema wird aufgrund seiner Wichtigkeit für diese Arbeit (AW2) im Kapitel 2 wiederholt. Für Details zu den anderen AW1-Themen wird auf die Ausarbeitung verwiesen (Dieumo, 2010).

Auf AW1 folgte das Projekt1, welches in Arbeit ist. Im Rahmen des Projekt1 soll der TTEthernet Protokoll-Stack für die Simulation in OMNeT++ entwickelt (Design, Implementierung, etc.) werden. Wichtige zu realisierende Komponente sind, die TTEthernet-Protokollschicht (TTEthernet-Dienst), ein Modell für die Planung der Nachrichten, ein Scheduler, der das Senden der geplanten Nachrichten triggert und eine Schnittstelle zur Anbindung von echten TTEthernet Applikationen in der Simulation. Der Protokoll-Stack soll selbstverständlich getestet werden. Dafür soll Traffic generiert werden können. Hierbei ist zu beachten, dass typische bzw. charakteristische Traffics generiert werden. Somit liegt der Fokus dieser Arbeit (AW2) auf verwandte Methoden in diesem Zusammenhang.

2 TTEthernet

In diesem Kapitel wird TTEthernet allgemein beschrieben und seine verschiedenen Traffic-Klassen vorgestellt.

2.1 Allgemeines

TTEthernet ist eine Echtzeit-Variante des Standard-Ethernet-Protokolls. Die Echtzeitfähigkeit beruht auf einem Verfahren, welches auf Zeitschlitzen aufbaut (TDMA -Time Division Multiple Access-). Dabei wird ein Zeitplan vordefiniert, nach dem alle Teilnehmer operieren müssen. Hierfür wird für jeden Zyklus ein Sende-Zeitintervall für jede (harte) Echtzeit-Nachricht (Time-Triggered Nachricht) festgelegt. Erwähnenswert ist, dass diese Zeitintervalle sich nicht überschneiden dürfen. Somit wird sichergestellt, dass zwei Echtzeit-Nachrichten nicht zur selben Zeit gesendet werden. Bei einem TDMA-Verfahren ist es wichtig, dass alle Teilnehmer miteinander synchronisiert sind. Daher werden diese im TTEthernet über ein ausfallsicheres Synchronisationsprotokoll mit einer globalen Zeit synchronisiert. TTEthernet ermöglicht nicht nur die Übertragung von Daten mit harten Echtzeitanforderungen, sondern auch mit weichen und ohne Echtzeitanforderungen. Dies wird anhand verschiedener Traffic-Klassen realisiert.

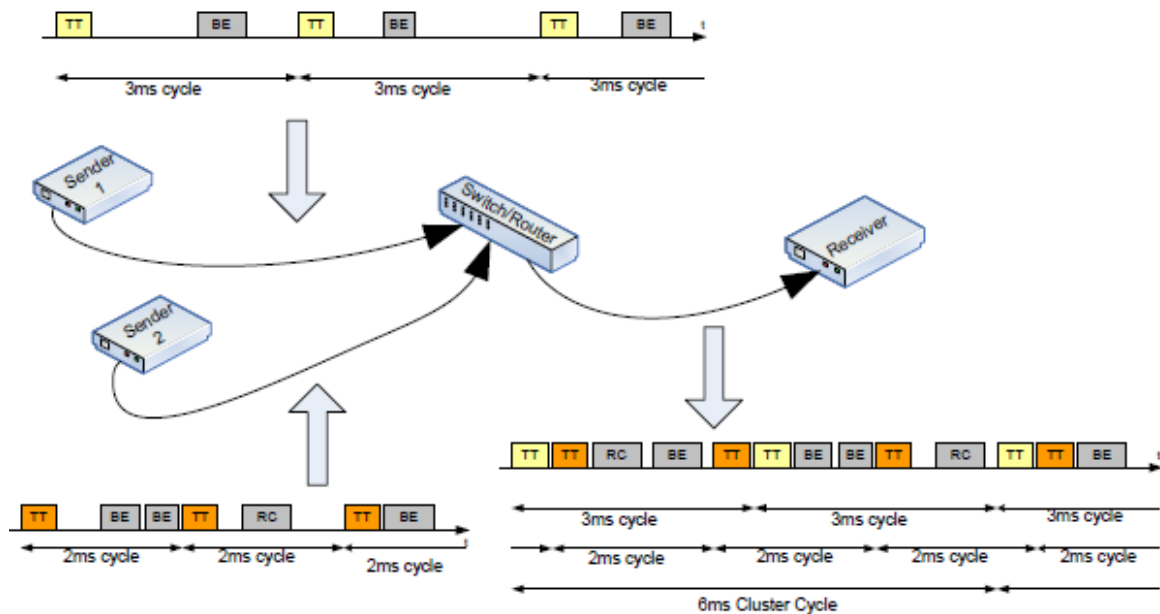


Abbildung 2-1: TTEthernet Traffic-Klassen (vom Switch weitergeleitet nach Priorität) (Wilfried, Günther, Brendan, Michael, & Srivatsan, 2009, S. 4)

2.2 Traffic-Klassen

TTEthernet definiert drei Traffic-Klassen (Abbildung 2-2):

- TT-Traffic (Time-Triggered Traffic): Dient zur Übertragung von Nachrichten mit harten Echtzeitanforderungen (TT-Nachrichten). Diese unterliegen wie bereits gesagt einer festen Planung (Abbildung 2-2) und haben Vorrang vor allen anderen Nachrichten.
- RC-Traffic (Rate-Constrained Traffic): Dient zur Übertragung von Daten mit weichen Echtzeit-Anforderungen. Hierbei wird für eine Anwendung eine feste Bandbreite definiert, indem sogenannte BAGs (Bandwidth Allocation Gap) festgelegt werden. Diese definieren die minimale Zeit zwischen zwei nachfolgende Frames sowie die maximal zu verwendeten Framelänge. Somit wird sichergestellt, dass Nachrichten mit einem „begrenzten Jitter“ übermittelt werden. Dieses Protokoll entspricht dem AFDX (Avionics Full Duplex Switched Ethernet)- Protokoll (AIM, 2008) und wird mit der zweiten Priorität behandelt.
- BE-Traffic (Best-Effort Traffic): Dient zur Übertragung von Daten ohne Echtzeitanforderungen. Dies entspricht dem Standard Ethernet-Traffic und wird mit niedrigster Priorität weitergeleitet. Eine zuverlässige Übertragung wird nicht gewährleistet.

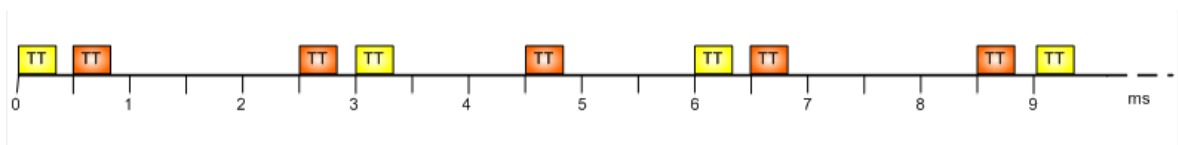


Abbildung 2-2: Beispiel einer TT-Traffic-Planung (extrahiert aus Abbildung 2-1)

Aufgrund der verschiedenen Traffic-Klassen und deren Prioritäten bzw. unterschiedlicher Anforderungen tauchen mehrere Fragen auf:

- Wie können die Zeitintervalle zwischen TT-Nachrichten so verteilt werden, dass noch ausreichende Bandbreite für RC- und BE-Nachrichten zur Verfügung steht?
- Wie kann man feststellen, ob die an den RC-Traffic gestellten Anforderungen erfüllt werden können?
- Wie kann man feststellen, ob noch „genug“ BE-Nachrichten weitergeleitet werden?
- Wie können Design-Fehler bezüglich der Traffic-Planung herausgestellt werden?
- Etc.

Um Aussagen über diese Fragen machen zu können, sind systematische Verfahren zur Planung, Generierung und Analyse von Traffics unerlässlich. Zwei davon werden in den Nachfolgenden Kapiteln vorgestellt. Im Kapitel 3 werden Benchmarking-Methodiken zur Traffic-Generierung und zur Ermittlung von charakteristischen Merkmalen eines Netzwerks und im Kapitel 4 mathematische Modelle zur Beschreibung von realistischen Traffics vorgestellt.

3 RFC 2544 Benchmarking-Methodik

RFC 2544 wurde von der Benchmarking Methodology Working Group (IETF, 2010) der IETF (Internet Engineering Task Force) festgelegt. In diesem Dokument wird eine Reihe von Tests beschrieben, die verwendet werden können, um die Leistungsmerkmale von Netzwerk-Geräten zu ermitteln. Außerdem werden spezifische Formate zur Protokollierung der Testergebnisse vorgeschlagen.

3.1 Versuchsaufbauten

Der typische Versuchsaufbau (Abbildung 3-1, a) besteht aus einem TI (Test Instrument) oder Tester und einem DUT (Device under Test). Das TI generiert Traffic für das DUT. Das DUT leitet den empfangenen Traffic an das TI weiter. Somit kann das TI den empfangenen Traffic analysieren. In einigen Situationen (z. B. testen von Virtual-Links) werden das Sender- und Receiver-TI getrennt betrachtet (Abbildung 3-1, b).

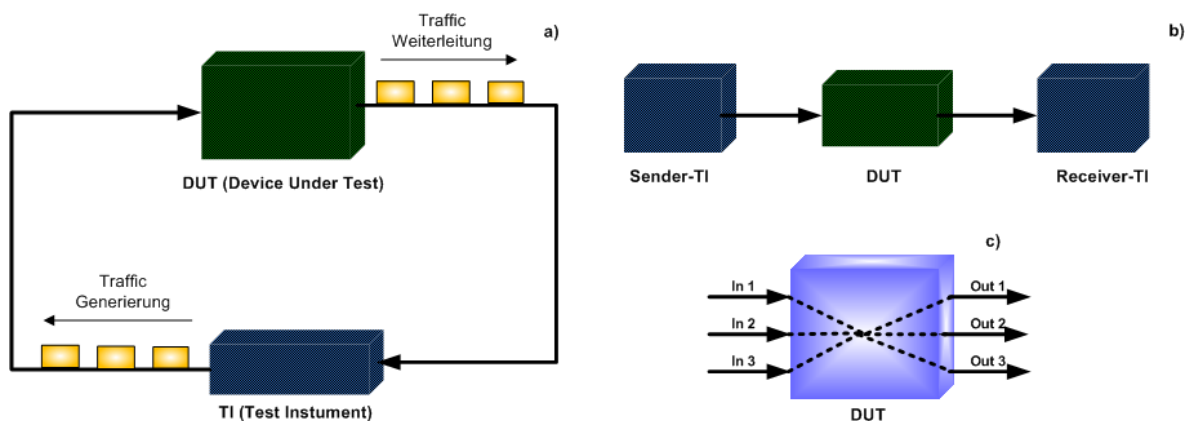


Abbildung 3-1: Versuchsaufbauten

3.2 Randbedingungen der Test-Durchführung

Framegröße

Es sollen verschiedene Framegrößen zum Generieren des Test-Traffics verwendet werden. Für Ethernet werden 64, 128, 256, 512, 1024, 1280 und 1518 Bytes vorgeschlagen. Wobei 64 die minimale und 1518 die maximale Framegröße ist.

Sequenznummer

Das TI soll die zu sendenden Frames mit Sequenznummern versehen. Anhand dieser Nummern kann die Anzahl der verloren gegangenen und der doppelten empfangenen Frames ermittelt werden. Es kann außerdem festgestellt werden, ob die Frames in der gesendeten Reihenfolge empfangen worden sind.

Traffic Modifiers

Neben den Test-Traffic sollen auch sogenannte „Traffic Modifiers“ generiert werden. Diese sind z. B. Netzwerkmanagement Anfragen, Routing-Updates, Diagnose Daten oder Protocol Control Frame (im Falle von TTEthernet zur Synchronisierung der Teilnehmer).

Datenstrom Richtung

Datenströme fließen meist nicht in eine, sondern in zwei Richtungen. Der Test-Traffic soll also in beide Richtungen generiert werden, um die bidirektionale Leistung des DUTs zu testen.

DUT mit mehreren Ports

Hat das DUT mehreren Ports (Switch, Router), wird es empfohlen, die Hälfte der Ports als Eingang und die andere Hälfte als Ausgang einzustellen (Abbildung 3-1, c). Ein Testscenario wäre z. B. den Traffic gleichzeitig an alle Eingangs-Ports zu senden und den an einem Eingangsport eingetroffenen Traffic an alle Ausgangsports weiterzuleiten. Somit kann sichergestellt werden, dass wie „im echten Leben“ mehrere Pakete gleichzeitig an einem Ausgangsport ankommen.

Test Durchführung

Ein Test wird in der Regel mehrmals durchgeführt. Nach jeder Durchführung muss mindestens 2 Sekunden gewartet werden, um sicherzustellen, dass das TI alle vom DUT weitergeleiteten Frames empfangen hat und mindestens 5 Sekunden damit das DUT wieder empfangsbereit ist.

Der generierte Traffic hängt vom durchzuführenden Test ab. Es wird nachfolgend eine Auswahl der Tests beschrieben. Für alle Tests wird auf RFC 2544 referenziert. Ein weiteres wichtiges Dokument ist RFC 1242. Dieses gibt Definitionen der in RFC 2544 verwendeten Begriffe an.

3.3 Back-to-back (burst) Test

Um die Leistung eines DUT unter realistischen Bedingungen zu messen, sollte Burst-Traffic eingesetzt werden. Frames innerhalb eines Bursts werden mit dem minimalen Interframe-Gap gesendet. Die Anzahl der Frames innerhalb eines Bursts soll konstant sein und das Inter-Burst-Intervall soll variieren. Es soll mit den Burstlängen 16, 64, 256 und 1024 Frames getestet werden. Das Ziel eines Burst-Tests ist die Ermittlung der Anzahl der Frames, die das DUT während eines Bursts ohne Verlust weiterleiten kann (Burst-Länge). Sein Ablauf wird durch Abbildung 3-2 beschrieben. Der Test soll mindestens 2 Sekunden dauern und mindestens 50 mal je Framegröße durchgeführt werden. Das Endergebnis ist der Mittelwert über alle Ergebnisse.

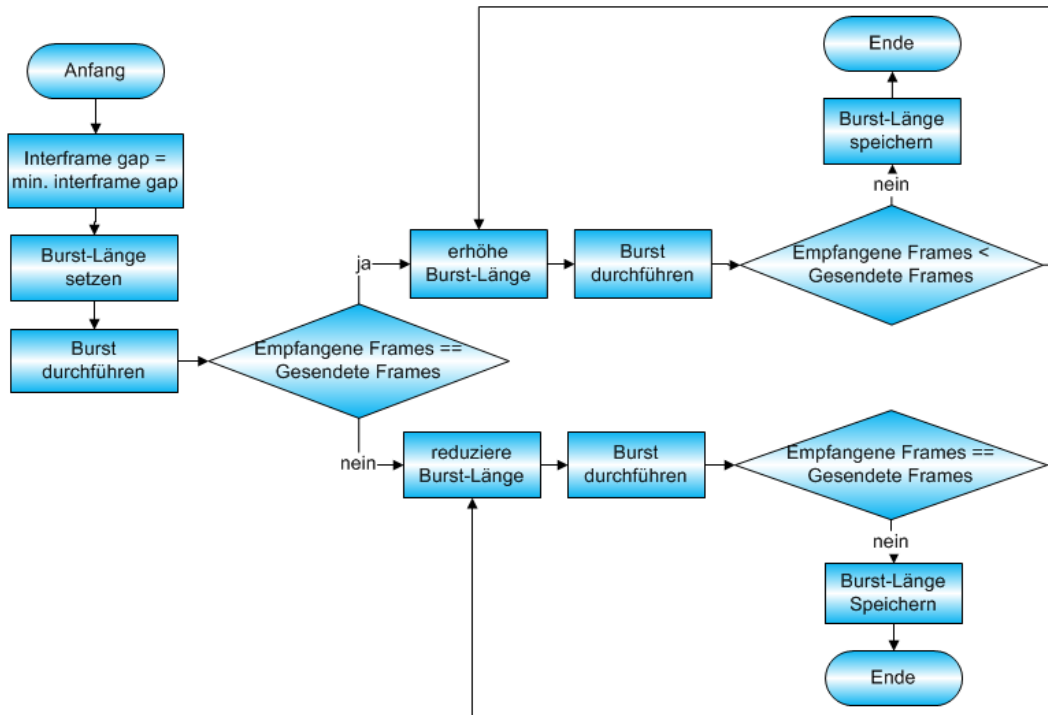


Abbildung 3-2: Ablauf Burst Test

3.4 Durchsatz Test

Beim Durchsatz (throughput) Test wird die maximale Datenrate ermittelt, bei der das DUT ohne Frame-Verlust weiterleiten kann. Der Ablauf eines Durchsatz-Tests wird durch Abbildung 3-3 illustriert. Der Test soll für alle Frame-Größe durchgeführt werden. Das Ergebnis wird als Graf angegeben. In der X-Achse wird die Frame-Größe eingetragen und in der Y-Achse der Durchsatz.

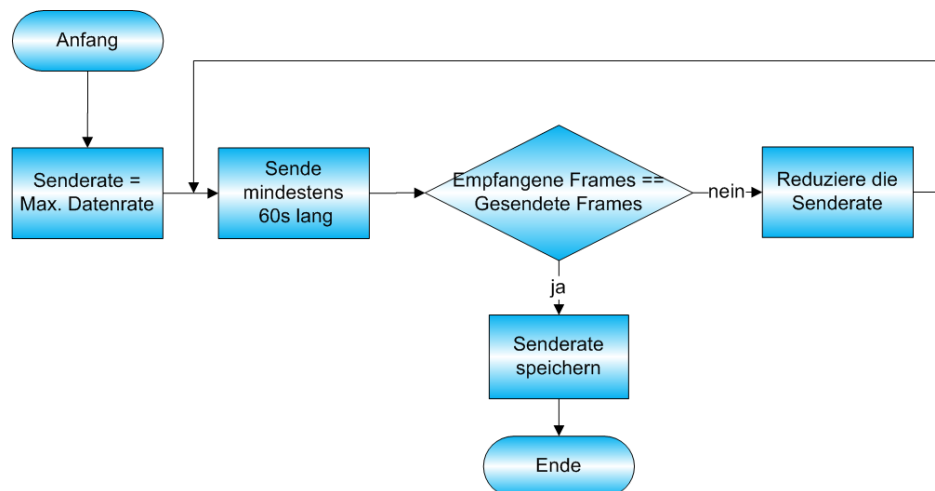
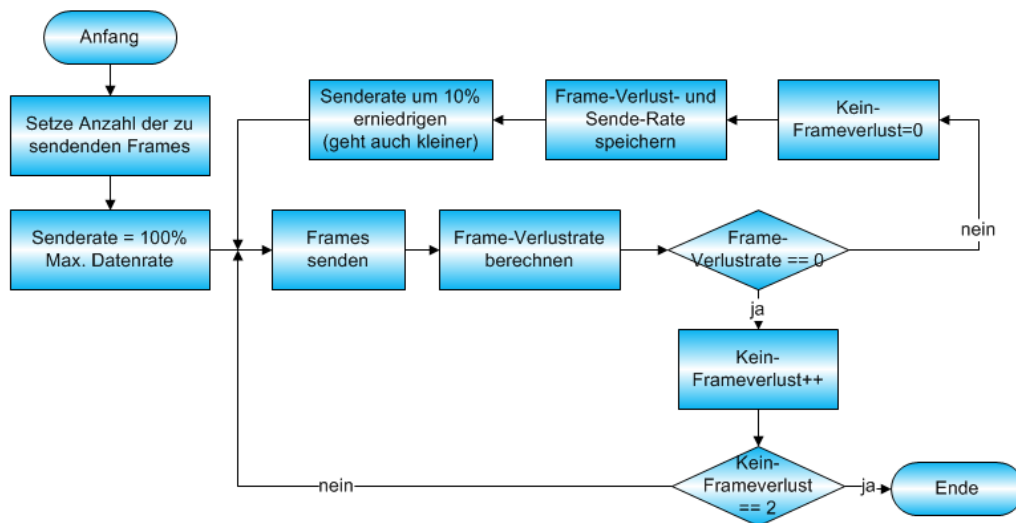


Abbildung 3-3: Ablauf Durchsatz-Test

3.5 Frame- Verlustrate Test

Das Ziel des Frame-Verlustrate-Tests ist die Ermittlung des Prozentsatzes der Frames, die das DUT hätte weiterleiten soll, dies aber aus Ressourcenmangel nicht getan hat. Der Ablauf eines Durchsatz-Tests wird durch Abbildung 3-4 beschrieben. Dabei wird am Anfang die maximale Datenrate verwendet und diese kontinuierlich bei Frame-Verlusten reduziert. Der Test findet nur ein Ende, wenn zwei Versuche nacheinander ohne Frame-Verlust durchgeführt worden sind. Das Ergebnis des Tests wird als Graph angegeben. In der X-Achse wird die Senderate eingetragen und in der Y-Achse die Verlustrate. Da der Test für alle Framegrößen durchgeführt wird, wird je Framegröße ein Graph angegeben.



$$\text{FrameVerlustRate} = \frac{(\text{GesendeteFrames} - \text{EmpfangeneFrames}) * 100}{\text{GesendeteFrames}}$$

Abbildung 3-4: Ablauf und Berechnung des Frame-Verlustrate-Tests

3.6 Bewertung

Die im RFC 2544 vorgeschlagene Verfahren sind geeignet zur Bewertung und Messung der Leistung von Netzwerk-Geräten. Diese können außerdem in einer Simulationsumgebung verwendet werden, um charakteristische Merkmale eines Netzwerks zu ermitteln. Für ein TTEthernet-Netzwerk können zum Beispiel verschiedene Konstellationen bezüglich der Traffic-Konfiguration getestet werden. Darüber hinaus sind für diese Verfahren keine mathematische Kenntnisse notwendig. Sie sind jedoch zeitintensiv. Im RFC 2544 wird nicht angegeben wie spezifische Traffics wie Video-, FTP-, WWW- oder Kraftfahrzeug-Anwendungen-Traffics generiert werden können. Allerdings kann die „Testabdeckung“ eines Netzwerks nicht nur mit den in RFC 2544 vorgeschlagen (Traffic-Generierung-) Methoden erreicht werden. Daher wenden wir uns im nachfolgenden Kapitel den mathematischen Modellen zur Beschreibung von Traffics zu.

4 Analytische Traffic-Modelle

In diesem Kapitel wird eine Auswahl analytischer Traffic-Modelle beschrieben.

4.1 Einfache Modellierung

Ein einfacher Traffic besteht aus dem Eintreffen von diskreten Ereignissen (Datenpakete, Anrufversuche, ...). Er kann mathematisch als Punkt-Prozess beschrieben werden, welcher aus einer Sequenz ankommender Instanzen T_1, T_2, \dots, T_n besteht. Ein Punktprozess kann auch durch das Zeit-Intervall zwischen zwei eingetroffenen Ereignissen $\xi_n = T_n - T_{n-1}$ beschrieben werden (Abbildung 4-1). Um solch einen Prozess in einer Simulationsumgebung zu realisieren, werden zufällige Zwischenankunftszeitpunkte generiert (Frost & Melamed, 1994, S. 72).

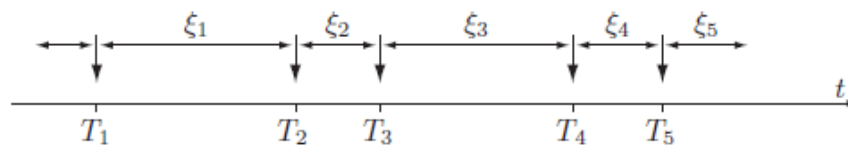


Abbildung 4-1: Einfache Traffic-Modellierung

4.2 Poissonverteilung

Die Poissonverteilung lässt sich gut von der Binomialverteilung ableiten. Die Binomialverteilung wird zur Modellierung von Zufallsexperimenten verwendet, bei denen nur zwei Ergebnisse im Sinne von Treffer und Nicht-Treffer möglich sind. Dabei wird angenommen, dass die eintreffenden Ereignisse voneinander unabhängig sind. Die Binomialverteilung wird beschrieben durch:

$$P(Y = k) = \frac{n!}{(n - k)! k!} p^k (1 - p)^{n - k}$$

mit

$$E[Y] = np \text{ und } \text{VAR}[Y] = np(1 - p)$$

Wobei $P(Y = k)$ die Wahrscheinlichkeit ist, genau k Treffer zu erzielen bei n mal Durchführung eines Experiments (Weigand, 2009, S. 191). Wird die Versuchsanzahl n sehr groß ($n \rightarrow \infty$) und die Trefferchance p eines Einzelexperimentes sehr gering, verwendet man die Poissonverteilung. Nachfolgend wird gezeigt wie die Poissonverteilung sich aus der Binomialverteilung ableiten lässt.

$$P(Y = k) = \frac{n!}{(n-k)! k!} p^k (1-p)^{n-k} \quad \text{mit} \quad E[Y] = \mu = np$$

$$\begin{aligned} P(Y = k) &= \lim_{n \rightarrow \infty} \frac{n!}{(n-k)! k!} \left(\frac{\mu}{n}\right)^k \left(1 - \frac{\mu}{n}\right)^{n-k} \\ &= \lim_{n \rightarrow \infty} \left(\frac{\mu^k}{k!}\right) \left(\frac{(n-k)! (n-(k-1)) \dots (n-2)(n-1)n}{(n-k)! n^k}\right) \left(1 - \frac{\mu}{n}\right)^n \left(1 - \frac{\mu}{n}\right)^{-k} \\ &= \frac{\mu^k}{k!} \lim_{n \rightarrow \infty} \left(\frac{n}{n} * \frac{n-1}{n} * \frac{n-2}{n} \dots \frac{n-k+1}{n}\right) \left(1 - \frac{\mu}{n}\right)^n \left(1 - \frac{\mu}{n}\right)^{-k} \end{aligned}$$

$$P(Y = k) = \frac{\mu^k}{k!} e^{-\mu} \quad \text{mit} \quad E[Y] = V[Y] = \mu \quad (\text{Weigand, 2009, S. 207})$$

Die Poissonverteilung wird eingesetzt um eintreffende Ereignisse innerhalb eines bestimmten Zeitintervalls zu modellieren. Z. B. Zugriffe pro Zeiteinheit auf einem Server oder die Anzahl eintreffender Daten-Pakete innerhalb einer Minute. Sie ist sehr beliebt, weil sie sehr einfach ist und weniger Parameter hat. Man kann jedoch mit ihr kein komplexes Verhalten modellieren. Das dynamische Verhalten des Ethernet-Traffics kann mit der Poissonverteilung nicht widerspiegeln werden.

4.3 Selbstähnlichkeit

Der Ethernet-Traffic lässt sich mit sogenannten selbstähnlichen Prozessen realistischer modellieren. Es gibt mittlerweile mehrere Arbeiten - (LELAND W. M., 1993), (Riedi & Willinger, 2000)-, die sich auf den Selbstähnlichkeitscharakter des Ethernetdatenverkehrs fokussiert haben. Als Initialzündung für den Nachweis des selbstähnlichen Charakters des Ethernet-Trffics waren die Messungen über eine Periode von mehr als vier Jahren (1989-1992) am Bellcore Morris Research and Engineering Center (LELAND W. M., 1993).

Charakteristische Eigenschaften von selbstähnlichen Datenverkehrsmodellen sind z. B. die Langzeitkorrelation, der Hurst-Parameter sowie die Skalierungsinvarianz (Kaage, 1998, S. 16). Bei der letzten wird das gleiche Verhalten bei verschiedenen (Zeit-) Skalen betrachtet. Ein Beispiel von selbstähnlichen Modellen ist das zeitdiskrete FARIMA (Fractional Autoregressive Integrated Moving Average)-Modell. Mit ihm kann sowohl lang- als auch kurzzeitanhängige Traffics beschrieben werden (Fei, Jiakun, Yantai, Lianfang, & Yang, 2002, S. 1). Dieses Model kann außerdem für Sprach, Bursty- und Multimedia-Traffic verwendet werden.

Aufgrund der Komplexität von selbstähnlichen Prozessen wurde eine mathematische Beschreibung ausgelassen. Hierfür wird auf weiterführende Literaturen verwiesen (Grimm & Schlüchtermann, 2004).

5 Zusammenfassung und Ausblick

In diesem Kapitel wird eine Zusammenfassung und ein Ausblick dieser Arbeit gegeben.

5.1 Zusammenfassung

In dieser Arbeit wurden verwandte Methoden zur Planung, Generierung und Analyse des Ethernet-Traffic vorgestellt. Das einführende Kapitel stellte die festgelegten Ziele für die Entwicklung des TTEthernet-Simulators vor. Auf die Ziele folgte der Darlegung des aktuellen Stands meiner Arbeiten. Dabei wurde eine Zusammenfassung der in AW1 behandelten Themen und die Ziele von Projekt1 angegeben. Im Kapitel 2 wurde TTEthernet allgemein beschrieben und seine drei Traffic-Klassen vorgestellt. Kapitel 3 beschäftigte sich mit der Benchmarking-Methodik, womit Traffics generiert und charakteristische Merkmale eines Netzwerks ermittelt werden können. Diese Benchmarking-Methodik definiert aber kein Verfahren, um spezifische Traffics wie Video-, WWW-Traffics, etc. zu generieren. Diese Lücke kann mit analytischen Traffic-Modellen gefüllt werden. Zwei davon wurden in Kapitel 4 vorgestellt. Einerseits die Poissonverteilung, die zwar einfach zu verstehen ist und wenige Parameter hat aber das dynamische Verhalten des Ethernet-Traffic nicht vollständig widerspiegeln kann. Andererseits selbstähnliche Traffic-Modelle, die zwar komplex sind aber Ethernet-Traffic gut beschreiben können.

5.2 Ausblick

Für die Analyse von TTEthernet-Netzwerken in der Simulationsumgebung ist es unerlässlich Netzwerk-Modelle zu erstellen, die reelle Netzwerke so gut wie möglich abbilden. Hierzu sollen realistische Topologien (z. B. typische Fahrzeug-Netzwerk-Topologien) erzeugt werden. Des Weiteren sollen genauere Modelle der Nachrichten-Planung sowie der Traffic-Generierung erstellt werden. Was die Traffic-Generierung angeht, können bestehende Traffic-Modelle auf deren Eignung für das TTEthernet-Netzwerk untersucht werden. Außerdem können neue Traffic-Modelle nach Bedarf erstellt werden. Stehen die Traffic-Modelle fest, sollen diese in der Simulation implementiert werden. Ein weiterer wichtiger Punkt ist das Testen der an den Traffic gestellten Anforderungen.

Abkürzungsverzeichnis

AFDX	Avionics Full Duplex Switched Ethernet
AW1	Anwendungen 1
AW2	Anwendungen 2
BAG	Bandwidth Allocation Gap
BE	Best-Effort
CAN	Control Area Network
DUT	Device Under Test
FARIMA	Fractional Autoregressive Integrated Moving Average
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force
Kfz	Kraftfahrzeug
LIN	Lokal Interconnect Network
MOST	Media Oriented Systems Transport
MPEG	Moving Picture Experts Group
OMNeT++	Objective Modular Network Testbed in C++
RC	Rate-Constrained
RFC	Requests for Comments
SS7	Signalling System 7
TDMA	Time Division Multiple Access
TI	Test Instrument
TT	Time-Triggered
TTEthernet	Time-Triggered Ethernet
WWW	World Wide Web

Abbildungsverzeichnis

<i>Abbildung 2-1: TTEthernet Traffic-Klassen (von Switch weitergeleitet nach Priorität) (Wilfried, Günther, Brendan, Michael, & Srivatsan, 2009, S. 4)</i>	6
<i>Abbildung 2-2: Beispiel einer TT-Traffic-Planung (extrahiert aus Abbildung 2-1)</i>	7
<i>Abbildung 3-1: Versuchsaufbauten</i>	8
<i>Abbildung 3-2: Ablauf Burst Test</i>	10
<i>Abbildung 3-3: Ablauf Durchsatz-Test</i>	10
<i>Abbildung 3-4: Ablauf und Berechnung des Frame-Verlustrate-Tests</i>	11
<i>Abbildung 4-1: Einfache Traffic-Modellierung</i>	12

Literaturverzeichnis

AIM. (2008). *AFDX*. Abgerufen am 08 2010 von <http://www.afdx.com/afdx.html>

Dieumo, H. (2010). *Simulation von Time-Triggered Ethernet im Automobilkontext mit OMNeT++ : Einführung*. HAW Hamburg, Department Informatik.

Fei, X., Jiakun, L., Yantai, S., Lianfang, Z., & Yang, O. W. (2002). Traffic Modeling Based on FARIMA Models. *Electrical and Computer Engineering, 1999 IEEE Canadian Conference*. Canada.

Frost, V., & Melamed, B. (1994). *Traffic Modeling For Telecommunications Networks*.

Grimm, C., & Schlüchtermann, G. (2004). *Verkehrstheorie in IP-Netzen: Modelle, Berechnungen, statistische (1. Auflage Ausg.)*. Bonn: Hüthig Telekommunikation .

IETF. (2010). *Benchmarking Methodology*. Abgerufen am 08 2010 von <http://datatracker.ietf.org/wg/bmwg/charter/>

Kaage, U. (1998). *Hochleistungsnetzwerk in und zwischen Studios: Stochastische Quellencharakterisierung*. Karlsruhe.

LELAND, W. E. (1994). On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*.

LELAND, W. M. (1993). On the Self-Similar Nature of Ethernet Traffic. *SIGCOMM'93*.

Mischo, S., Powolny, S., Zündel, H., Löchel, N., Stuphorn, J., Constapel, R., et al. (2007). *Vernetzung im Kraftfahrzeug*. Germany.

OMNeT++ Community2. (kein Datum). *OMNeT++*. Abgerufen am 08 2010 von INET Framework for OMNeT++: URL- <http://inet.omnetpp.org>

Riedi, R. H., & Willinger, W. (2000). in Self-similar Network Traffic and Performance Evaluation. In *Towards an improved understanding of network traffic dynamics* (S. 507-530). Wiley.

SAE - AS-2D Time Triggered Systems and Architecture Committee. (2009). *Time-Triggered Ethernet (AS 6802)*. Abgerufen am 08 2010 von URL <http://www.sae.org>

Steiner, W. (November 2008). TTEthernet specification. TTTech Computertechnik AG.

TTTech. (kein Datum). *TTEthernet*. Abgerufen am 08 2010 von URL <http://www.tttech.com>

TU Wien - Institut für Technische Informatik. (kein Datum). *Real-Time Systems Group*. Abgerufen am 08 2010 von <http://www.vmars.tuwien.ac.at/projects/tta/index.html>

Weigand, C. (2009). *Statistik mit und ohne Zufall; Eine anwendungsorientierte Einführung (2. Auflage Ausg.)*. Berlin Heidelberg: Springer-Verlag.

Wilfried, S., Günther, B., Brendan, H., Michael, P., & Srivatsan, V. (2009). *TTEthernet Dataflow Concept*.

