



Freie Universität Berlin

# Car2X-Kommunikation für autonom fahrende Modellautos

## Masterarbeit

geschrieben von Severin Junker

Fachbereich für Mathematik und Informatik (Institut für Informatik)  
Arbeitsgruppe *Intelligente Systeme und Robotik*

betreut durch Prof. Dr. Raúl Rojas  
Prof. Dr. Daniel Göhring

gedruckt am 22. Januar 2016

Revision fa530bb «master»  
Fri Jan 22 04:01:51 2016 +0100



# Zusammenfassung

Die menschliche Kommunikation dient dem Austausch von Informationen, die zur Planung zukünftiger Vorhaben genutzt werden können. Damit wird das weitere Vorgehen optimiert und somit Fehler, die von anderen gemacht wurde vermieden. Informationen werden dabei nicht nur von einer Person genutzt. Sollten sich die Informationen für andere als ebenso wichtig herausstellen, werden diese auch an andere weitergegeben. Hierbei handelt es sich um das Prinzip von Memen und deren Verbreitung.

Mit der Verbreitung von immer mehr Informationstechnologien, tritt die Automobilbranche in den nächsten Generations-Zyklus über. Ziel ist es, Erfahrungen, die Autofahrer gewonnen haben, an andere weiterzugeben, um Unfälle vorzubeugen.

Diese Arbeit beschäftigt sich mit den Grundlagen der Fahrzeugkommunikation und der Realisierung einer C2X-Kommunikationsplattform für autonome Modelfahrzeuge, basierend auf dem Berlin-United Framework.

## Abstract

Humans communication exchanges information and uses this for the planning of future events. In this way, they can optimize the procedure and try to avoid the errors made by others in the past. The informations are not only for one person, if the informations are important and useable, they will pass on to the others. This is the principle of memes.

With the increasingly spread of information technologies the automotive industry takes over to the cycle of the next generation. Primary purpose is to benefit from the experiences that motorists have made, to prevent others from accidents.

This work deals with the basics of vehicle communication and the realization of a C2X communication platform for autonomous model cars, based on the Berlin-United Framework.



# Danke

Es war ein langer Weg, der nun sein Ende findet. Während des Studiums ist man durch Höhen und Tiefen gegangen, die ohne die Unterstützung von hilfreichen Geistern nicht zu meistern gewesen wären.

Darum möchte ich auf diesem Wege den folgenden Personen danken und mich bei denjenigen entschuldigen, die nicht hier aufgelistet sind.

Meine erste Dankesbekundung geht an die zwei Professoren Raúl Rojas und Daniel Göhring, die es ermöglicht haben diese Arbeit zu schreiben. Außerdem sollen die lehrreichen Vorlesungen nicht außer Acht gelassen werden.

Bedanken möchte ich mich ebenfalls bei Susanne Schöttker-Söhl, die als Sekretärin das Herz der Arbeitsgruppe Intelligente Systeme und Robotics ist. Ohne sie wären manche bürokratische Höhen unüberwindbar.

Meinen herzlichsten und größten Dank möchte ich Heiko Baumann aussprechen, der mir als Kommilitone auf der schönen Reise durch das Masterstudium mit Rat und Tat beiseite stand. Ebenso seiner Freundin, Katja Vieregge, die für das leibliche Wohl an arbeitsintensiven Wochenenden mit ihrem unvergesslichen Nudelsalat sorgte.

Ein großer Dank geht an die Jungs und Mädels vom FU-RacingTeam, AudiCupTeam, und den FUMANoids, aus dem Robotik-Labor, die mit ihren fachlichen Kompetenzen immer ein offenes Ohr für mich hatten. Besonders erwähnen möchte ich hier: Till-Julius Krüger, Jan Boldt, Tom Bullmann, Benjamin Zengin, Janis Ihrig, Lukas Maischak, Ercan Küçükkaraca, Daniel Krakowczyk, Conrad Lässig, Simon G. Gottlieb, Lutz Freitag, Malte Detlefson, Gregor Barth Daniel Seifert, Zahra Boroujeni, Tobias Höppner und Fritz Ulbrich.

Vor allem möchte ich mich auch bei meiner Freundin Michaela Engert bedanken, die sich von mir immer mein Technik-Geschwafel anhören mußte und dabei immer sehr aufmerksam und interessiert war.

Natürlich will ich auch hier die Unterstützung durch meine Eltern und meinen Bruder erwähnen. Nicht nur während der des Studiums sondern auch in allen anderen Lebenslagen davor.

Und so könnte die Aufzählung ewig weitergehen, doch das würde Bände füllen. Deshalb nochmal mein tiefster Dank an diejenigen die sich hier nicht persönlich wiederfinden.





# Inhaltsverzeichnis

I	Einführung	
1	<b>Vorwort</b> .....	13
1.1	Autonome Fahrzeuge	15
1.2	Der Weg zur Kommunikation	17
2	<b>Zur Arbeit</b> .....	19
2.1	Motivation	20
2.2	Ziele	20
2.3	Aufbau	20
II	Car2X Kommunikation	
3	<b>Hintergrund</b> .....	25
3.1	Car2X-Kommunikation und ITS	26
3.2	Anwendungsbereiche	28
3.3	Europäische Projekte	29
3.4	Der Weg zum Standard	30
4	<b>ETSI Grundlagen</b> .....	31
4.1	ETSI Standards	32

<b>4.2</b>	<b>Kommunikations-Architektur</b>	<b>32</b>
<b>4.3</b>	<b>Die Referenz Architektur</b>	<b>33</b>
4.3.1	Application-Schicht .....	33
4.3.2	Access Layer .....	34
4.3.3	Networking & Transport Layer .....	34
4.3.4	Facilities Layer .....	35
4.3.5	Management Layer .....	36
4.3.6	Security Layer .....	37
<b>4.4</b>	<b>Elemente eines ITS</b>	<b>38</b>
4.4.1	Personal Station .....	38
4.4.2	Central Station .....	38
4.4.3	Vehicle Station .....	38
4.4.4	Roadside Station .....	38
<b>4.5</b>	<b>Kommunikation</b>	<b>38</b>
4.5.1	Kommunikations-Szenarien .....	40
<b>4.6</b>	<b>ASN.1</b>	<b>41</b>
4.6.1	Beispiel .....	42
4.6.2	Datentypen .....	42
4.6.3	Kodierregeln .....	43
<b>5</b>	<b>Cooperative Awareness</b> .....	<b>45</b>
<b>5.1</b>	<b>Cooperative Awareness Message</b>	<b>46</b>
5.1.1	CA Basic Service .....	47
5.1.2	Aufbau einer CAM .....	48
5.1.3	Versand und Empfang von CAM .....	49
<b>5.2</b>	<b>Decentralized Environment Notification Messeage</b>	<b>51</b>
5.2.1	DEN Basic Service .....	52
5.2.2	Aufbau einer DENM .....	53
5.2.3	Versand und Empfang einer DENM .....	54
<b>5.3</b>	<b>Local Dynamic Map</b>	<b>56</b>
<b>6</b>	<b>Zeitpräzision und Synchronisierung</b> .....	<b>57</b>
<b>6.1</b>	<b>Global Positioning System</b>	<b>58</b>
<b>6.2</b>	<b>Synchronisation in der Welt der Verteilten Systeme</b>	<b>58</b>
6.2.1	Happened-Before .....	59
6.2.2	Lamport-Uhr .....	59
6.2.3	Vektor-Uhr .....	60
<b>6.3</b>	<b>Network Time Protocol</b>	<b>61</b>
6.3.1	Simple Network Time Protocol .....	63
<b>6.4</b>	<b>Precision Time Protocol</b>	<b>63</b>

<b>7</b>	<b>Routing-Algorithmen</b>	<b>67</b>
7.1	<b>Flooding-basierte Verfahren</b>	68
7.1.1	Einfaches Flooding	68
7.1.2	Ad-Hoc On Demand Vector Routing	69
7.2	<b>Positions-basierte Verfahren</b>	70
7.2.1	Geographic Unicast Routing	70
7.2.2	Greedy Perimeter Stateless Routing	71
7.2.3	Greedy Perimeter Coordinator Routing	73
7.3	<b>Karten-basierte Verfahren</b>	74
7.3.1	Global State Routing	74
7.3.2	Erweiterungen zum GSR	74

### III

## Projektumgebung

<b>8</b>	<b>Carolocup</b>	<b>77</b>
8.1	<b>Der Wettbewerb</b>	78
8.2	<b>Das FU-RacingTeam</b>	80
8.3	<b>Fuhrpark</b>	81
8.3.1	Gorilla-Klaus	81
8.3.2	Able	82
8.3.3	Miss Baker	82
8.4	<b>Rennstrecken</b>	83
<b>9</b>	<b>Berlin-United</b>	<b>85</b>
9.1	<b>Framework</b>	86
9.1.1	Protobuf	87
9.2	<b>Software-Tools</b>	88

### IV

## Implementierung

<b>10</b>	<b>Architektur</b>	<b>91</b>
10.1	<b>Aufbau</b>	92
10.2	<b>Lokalisierung</b>	92
10.3	<b>Modulübersicht</b>	92
10.4	<b>Datenaustausch</b>	93
10.5	<b>Datenverwaltung</b>	95
10.6	<b>Zeitsynchronisierung</b>	96
<b>11</b>	<b>Anwendungsfälle</b>	<b>97</b>
11.1	<b>Hinderniserkennung</b>	98

11.2	Verkehrsschild Geschwindigkeitsbegrenzung	101
11.3	Ampelszenario	104
11.3.1	Aufbau der Signalanlage .....	106

V

## Auswertung

12	Reflektion .....	109
12.1	Implementierung	110
12.2	Bewertung	111
12.3	Ausblick	112

VI

## Referenzen

Literatur .....	115
Bücher	115
Artikel	116
Technische Berichte	117
Online	119
Abschlussarbeiten	120
Abkürzungsverzeichnis .....	121
Abbildungsverzeichnis .....	124
Tabellenverzeichnis .....	126



# Einführung

<b>1</b>	<b>Vorwort</b>	13
1.1	Autonome Fahrzeuge	
1.2	Der Weg zur Kommunikation	
<b>2</b>	<b>Zur Arbeit</b>	19
2.1	Motivation	
2.2	Ziele	
2.3	Aufbau	





## 1. Vorwort

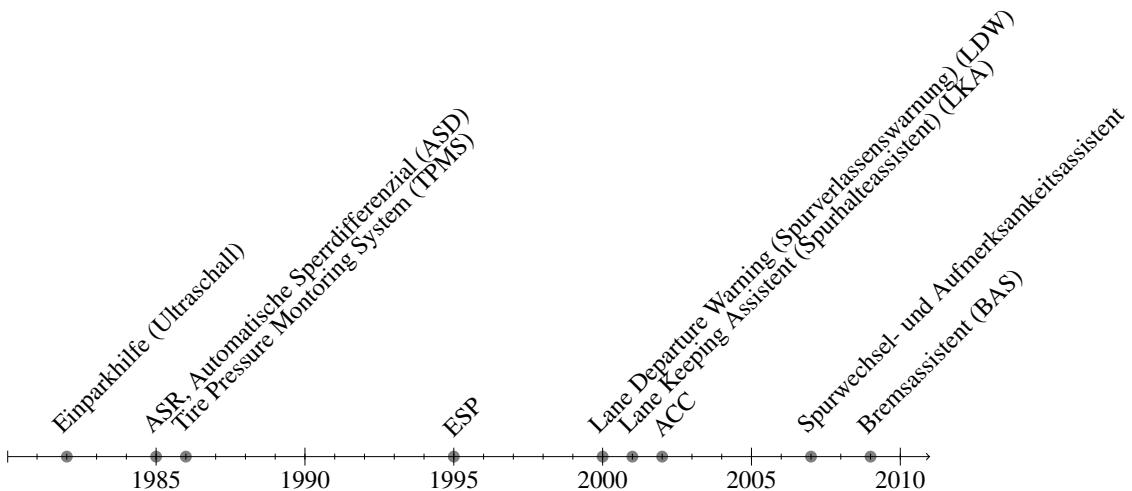
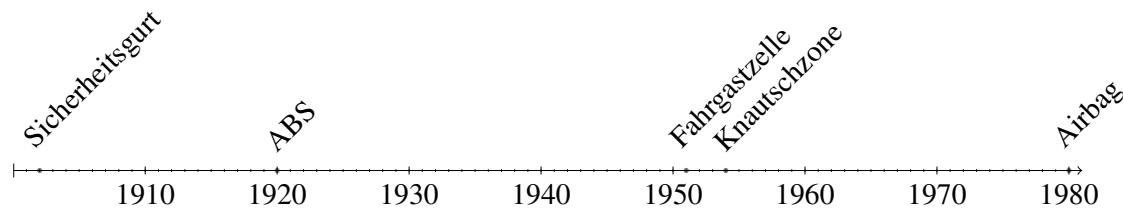
“ Fang nie an aufzuhören, hör nie auf anzufangen.

”

---

Marcus Tullius Cicero, 106-43 v. Chr.

Die Autoindustrie ist seit je her bemüht, das Fahren angenehmer und sicherer zu gestalten. Erfindungen wie der Airbag, Sicherheitsgurte, Deformationselemente und Einscheiben-Sicherheits-/Verbundglas sind physische Komponenten, die zum Schutz des Fahrgastes und auch zur Sicherheit anderer Verkehrsteilnehmer beitragen. Auf elektronischer Seite stehen Entwicklungen wie elektronische Stabilitätskontrolle (ESP), Antiblockiersystem (ABS), Antriebsschlupfregelung (ASR) (Traktionskontrolle (TCS)), Abstandstempomat (ACC), Notbremsassistent, Fahrbahn- und Umfelderkennung. Unterschieden werden diese zwei Konzepte in *passive* und *aktive* Sicherheitssysteme. Der Ausdruck *passiv* geht auf die Tatsache zurück, dass diese Systeme helfen sollen, wenn es zu einer Kollision gekommen ist. Die *aktiven* Systeme sollen dabei helfen, eine solche Situation zu vermeiden.



Anhand der Entwicklung lässt sich erkennen, dass der Trend zu immer mehr elektronischen Unterstützern im Fahrzeug führt. Ein Umstand der auch darin begründet ist, dass der Standard für die passiven Systeme soweit fortgeschritten ist, dass kaum noch Potential für Verbesserung vorhanden ist, sodass mehr Augenmerk auf die aktiven Systeme gelegt wird. Im Fachjargon laufen diese aktiven Hilfsmittel unter dem Titel *Fahrerassistenz-Systeme*, die in bestimmten Verkehrssituationen den Fahrer unterstützen sollen, die Kontrolle über das Fahrzeug zu behalten und Gefahrensituationen rechtzeitig zu erkennen. Aktive Systeme bringen gleichfalls ökonomische Aspekte mit sich.

So wird beispielsweise die Regelung für die Kraftstoffeinspritzung so optimiert, dass der Kraftstoffverbrauch bei einer hohen Leistung minimiert wird.



Abbildung 1.1: Fahrerassistenzsysteme (Quelle: Audi)

In der Abbildung 1.1 sind die gebräuchlichsten Assistenzsysteme, am Beispiel des Automobilherstellers Audi, illustriert.

Hierdurch steigert sich natürlich auch die Komplexität, der im Fahrzeug verbauten Eletronik und der Anforderung an die Bedienung. Durch die Technologisierung befindet sich heutzutage schon mehr Technik in einem Auto als in der Apollo 11 [1]<sup>1</sup>. Mitunter kann dies zu einer Überreizung der Autofahrer führen. Aus diesen Gründen kommt es zu einer neuen Generation von Automobilen.

## 1.1 Autonome Fahrzeuge

Aktuell wird an Fahrzeugen geforscht, die sich selbstständig durch den Straßenverkehr bewegen können. Hierzu greift die Fahrzeugelektronik auf eine Vielzahl von Sensoren zurück.

Prominentestes Beispiel hierfür sind die Autos von Google. Aktuell (Stand: November 2015) [25] unterhält der Konzern 53 Fahrzeuge, darunter 23 Lexus RX450h SUVs und 30 Prototypen, die auf amerikanischen Straßen in Mountain View (Kalifornien) und Austin (Texas) unterwegs sind. Dabei wurden von den Fahrzeugen, seit Beginn des Projektes, etwa 2.125.549 autonome Kilometer zurückgelegt. Durchschnittlich fahren sie ca. 16.093 - 24.140 Kilometer pro Woche auf öffentlichen Straßen.

<sup>1</sup>Erste Raumfahrtmission mit dem Flug und der Landung auf dem Mond, angestrebt durch John F. Kennedy im Jahre 1961.

Nicht nur Google experimentiert mit autonomen Kraftfahrzeugen. Automobilhersteller und Universitäten betätigen sich ebenfalls in diesem Bereich.

An der Freie Universität Berlin wurde das „Autonomos Labs“ gegründet, mit dem Ziel der Erforschung und Entwicklung autonomer Systeme. Die Grundlage bildet ein umgebauter VW Passat, der selbstständig durch Berlin fahren kann.

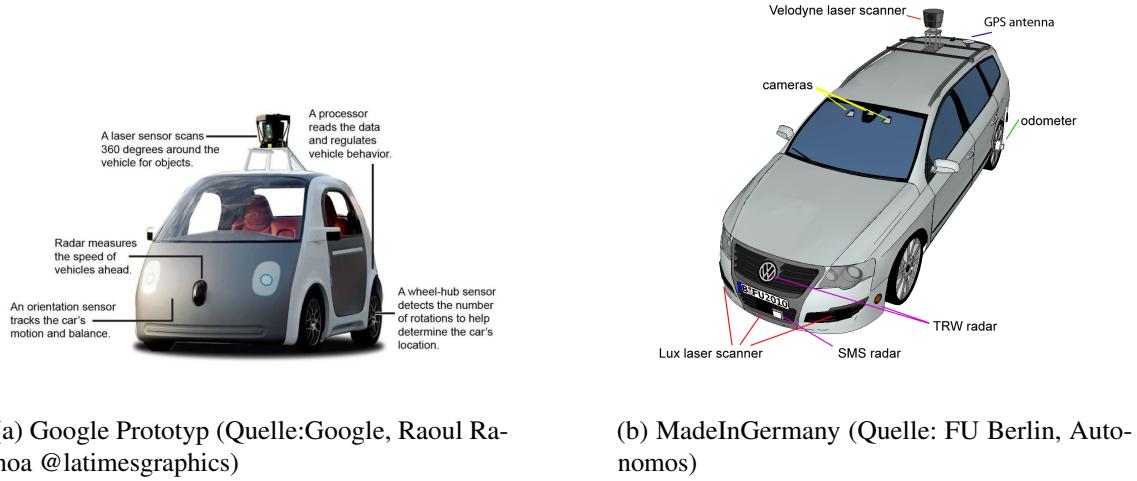


Abbildung 1.2: Selbst fahrende Fahrzeuge

Wie in der Abbildung 1.2 zu sehen ist, verfügen die Autos über eine Spannweite an Sensorik zur Wahrnehmung der Umwelt.

Am offensichtlichsten ist dabei der 360° Laserscanner (auch Lidar genannt), mit dem Laserstrahlen die Umgebung abtasten. Aus den gewonnenen Daten lässt sich dann später ein 3d-Modell konstruieren. Im vorderen Bereich befinden sich Kameras zur Erkennung von Fußgängern, Ampeln, Verkehrszeichen oder anderen, vor dem Fahrzeug liegenden Objekten. Am vorderen und hinteren Teil des Fahrzeugs sind Radar-Sensoren verbaut, mit denen sich Objekte und dessen Abstände detektieren lassen. Eine Antenne im hinteren Bereich des Fahrzeugs fängt Global Positioning System (GPS)-Signale auf, mit denen die eigene Position bestimmt wird. An den Rädern sind weitere Sensoren angebracht, um die Bewegung des Fahrzeugs zu messen. Zur Unterstützung und genaueren Abschätzung der Position befinden sich im Auto Gyroskope, Höhen- und Beschleunigungsmesser.

Im europäischen Raum existieren klare Vorstellungen ab, welchem Zeitpunkt die Einführung von autonomen Fahrzeugen voranschreiten soll. Nach [26] wird bis 2020 das teilweise autonome Fahren in wenig komplexen Umgebungen mit geringer Geschwindigkeit beispielsweise in Parkhäusern oder in einem Verkehrsstaub auf einer Autobahn zur Verfügung stehen. Bis spätestens 2025 soll der Grad der Autonomie erhöht werden, sodass Fahrzeuge selbstständig von einem Startpunkt zu einem Ziel (auf Bundes- oder Landstraßen) fahren. Zusätzlich wird Wert auf die Sensibilisierung der Umgebung gelegt, um Wildunfälle oder dergleichen zu vermeiden, damit sich der Fahrer auf andere Aktivitäten konzentrieren kann. Bis 2030 ist das vollständig autonome Fahren in einer städtischen Umgebung geplant, in

denen die Fahrzeuge auf Fußgänger, andere Verkehrsteilnehmer, Ampeln und vieles mehr achten müssen.

## 1.2 Der Weg zur Kommunikation

Wenn sich Fahrzeuge ohne menschliches Eingreifen auf Straßen bewegen und eine Fülle an Sensorik mit sich führen, dann liegt es nahe, dass sie zum Austausch der jeweiligen Sensordaten untereinander kommunizieren müssen, um sich gegenseitig vor Gefahren oder Hindernissen zu warnen. Überlegungen zur Fahrzeugkommunikation wurden bereits 1936 durch die Funkschau [27] in dem Artikel „Können Ultrakurzwellen den Kraftverkehr sicherer gestalten?“ angestellt. Ein skizzierter Anwendungsfall war das Signalisieren eines Überholvorgangs eines Kleinkraftwagens hinter einem LKW. Kommunikationen zwischen den Autos über längere Entfernung fanden zunächst keine Berücksichtigung, da die Reichweite der Signalübertragung zur damaligen Zeit gerade einmal ca. 50 Metern betraf.

Die Entwicklung der Fahrzeugkommunikation ist dabei unabhängig von der Entwicklung von autonom agierenden Kraftfahrzeugen. Beide Systeme können getrennt voneinander agieren. Der einzige Unterschied ist die Komponente, welche aus den gewonnenen Daten eine schlüssige Beziehung herstellt und dementsprechend angemessen reagiert. Bei nicht autonomen Fahrzeugen heißt die Komponente Mensch, auf der anderen Seite ist es ein leistungsfähiger Computer.





## 2. Zur Arbeit

“ Mancher lehnt eine gute Idee bloß deshalb ab, weil sie nicht von ihm ist. ”

---

Luis Bunuel, 1900-1983

## 2.1 Motivation

Die Automobilbranche steckt derzeit viel Energie in die Entwicklung von intelligenten Systemen, die Unfälle erkennen sollen, bevor sie passieren. Ein Teil davon ist die Kommunikation zwischen Fahrzeugen, aber auch zwischen Fahrzeugen und der Umgebung.

Bisherige Forschungsarbeiten benötigen zum Testen von Algorithmen und konkreten Anwendungsszenarien eine Simulationsumgebung sowie ein größeres Testfeld, in dem simulierte Ergebnisse unter realen Bedingungen getestet werden können.

Für Universitäten, die unabhängige Forschungen von den großen Verbänden betreiben wollen, ist der finanzielle Aufwand für die Anmietung einer Teststrecke, des Equipments und einer Vielzahl von Fahrzeugen zu groß.

## 2.2 Ziele

In dieser Arbeit soll ein Fahrzeug-Kommunikationssystem für autonome Modelfahrzeuge auf Basis des *Berlin United*-Frameworks entwickelt werden. In Anlehnung an die Verfügbarkeit und die weite Verbreitung von 2,4 GHz Wireless Local Area Network (WLAN), soll diese als Kommunikationskanal fungieren.

Somit soll eine Plattform geschaffen werden, auf der Anwendungen im Bereich Fahrzeugkommunikation im Maßstab 1:10 realisiert und getestet werden können.

Zusätzlich soll an ausgewählten Beispielen die Funktionsweise der entwickelten Architektur gezeigt werden.

## 2.3 Aufbau

Die Arbeit wird eingeleitet mit einer Übersicht über bisherige Sicherheitskonzepte in Fahrzeugen. Weiterführend werden neue Ansätze diskutiert, in denen die Wahrnehmung mittels einer Vielzahl an Sensorik innerhalb des Fahrzeugs eine große Rolle spielt.

Der darauf folgende Abschnitt beschäftigt sich mit den grundlegenden Details zur Car to X (C2X)-Technologie. Es werden Vergleiche zwischen den unterschiedlichen Ansätzen in verschiedenen Staaten getroffen und ein Einblick in die Entwicklungsphasen auf europäischer Seite erläutert. Dies umfasst vor allem den Prozess von der Forschung bis zur Entwicklung und Einführung dieser neuen Technologie.

Das anschließende Kapitel beschreibt die europäischen Konzepte zum Aufbau eines Intelligent Transport System (ITS), die als Grundlage für die Kommunikation zwischen Fahrzeugen und der Infrastruktur dienen. Allem voran wird das Referenz-Modell vorgestellt, das die Grundlage für jeden ist, der sich innerhalb des ITS bewegen und von dessen Informationsgehalt profitieren möchte.

Hierzu werden die einzelnen Elemente des Referenz-Modells untersucht und veranschaulicht. Ebenso werden die Netzwerkstrukturen und die Datenformate für die Kommunikation zwischen den einzelnen Elementen eines ITS betrachtet.

Der Hauptteil der Arbeit befaßt sich mit den Kommunikationsprotokollen für die sogenannte „Cooperative Awareness“. Dessen Bestandteile dienen dem Austausch von Informationen zum aktuellen Zustand eines Fahrzeugs oder der Warnung vor Gefahrensituationen.

Die Informationen sind an Zeiten geknüpft, zu denen ein Ereignis aufgetreten ist. In der Regel wird bei Fahrzeugen davon ausgegangen, dass diese mit Hilfe von GPS immer ein exaktes Zeitsignal erhalten. Da diese Technologie bei Modell-Fahrzeugen nicht zur Verfügung steht, werden alternative Möglichkeiten ermittelt, mit denen eine Zeitsynchronisierung zwischen den einzelnen Teilnehmern eines ITS vonstatten gehen kann.

Das anschließende Kapitel beschäftigt sich mit den Herausforderungen der Nachrichtenweiterleitung innerhalb von Ad-Hoc Netzwerken. Es werden verschiedene Algorithmen, deren Einsatzmöglichkeit und die daraus resultierenden Vor- und Nachteile erörtert.

Nachfolgend wird ein Überblick über die Projektstruktur gegeben, dessen Rahmen die Grundlage und Motivation für diese Arbeit ist. Hierzu wird das FU-RacingTeam mit dessen Ausstattung, das Berlin-United Framework sowie der Carolocup-Wettbewerb vorgestellt.

In den abschließenden Kapiteln wird die Umsetzung der Softwarekomponenten zur Realisierung eines C2X-Systems erläutert und eine Bewertung der somit realisierten Fahrzeug-Kommunikations-Plattform für Modelfahrzeuge.



# Car2X Kommunikation

<b>3</b>	<b>Hintergrund .....</b>	<b>25</b>
3.1	Car2X-Kommunikation und ITS	
3.2	Anwendungsbereiche	
3.3	Europäische Projekte	
3.4	Der Weg zum Standard	
<b>4</b>	<b>ETSI Grundlagen .....</b>	<b>31</b>
4.1	ETSI Standards	
4.2	Kommunikations-Architektur	
4.3	Die Referenz Architektur	
4.4	Elemente eines ITS	
4.5	Kommunikation	
4.6	ASN.1	
<b>5</b>	<b>Cooperative Awareness .....</b>	<b>45</b>
5.1	Cooperative Awareness Message	
5.2	Decentralized Environment Notification Messeage	
5.3	Local Dynamic Map	
<b>6</b>	<b>Zeitpräzision und Synchronisierung ...</b>	<b>57</b>
6.1	Global Positioning System	
6.2	Synchronisation in der Welt der Verteilten Systeme	
6.3	Network Time Protocol	
6.4	Precision Time Protocol	
<b>7</b>	<b>Routing-Algorithmen .....</b>	<b>67</b>
7.1	Flooding-basierte Verfahren	
7.2	Positions-basierte Verfahren	
7.3	Karten-basierte Verfahren	





### 3. Hintergrund

“ Das Ganze ist mehr als die Summe seiner Teile. ”

---

Aristoteles, *Metaphysik VII 17*, 1041b

Eingeleitet wird dieser Bereich mit einer Einführung, wobei es sich bei der Fahrzeugkommunikation handelt. Es geht auf die verschiedenen regionalen Besonderheiten bei der Technik ein und zeigt einige Anwendungsbeispiele. Abschließend wird die Entwicklungsgeschichte im europäischen Raum näher erörtert.

### 3.1 Car2X-Kommunikation und ITS

Das Themengebiet Car to X (Car2X), umfasst die Kommunikation und den Datenaustausch zwischen einem Fahrzeug und der Verkehrsinfrastruktur (Car to Infrastructure (C2I) oder Vehicle to Infrastructure (V2I)) oder einem anderen Fahrzeug (Car to Car (C2C) oder Vehicle to Vehicle (V2V)). Die Elemente, die miteinander kommunizieren befinden sich dabei in einem Netzwerk, dem „Vehicular Ad-Hoc Network (VANET)“. Abstrahiert handelt es sich hierbei um ein „Mobile Ad-Hoc Network (MANET)“, in dem die einzelnen Elemente (Knoten), Fahrzeuge oder Teile der Verkehrsinfrastruktur (Road Side Unit (RSU)) (Ampeln, Verkehrszeichen, etc.) darstellen. Im Ganzen wird dies unter dem Oberbegriff ITS eingegliedert.

**Definition 3.1.1** Ein ITS zeichnet sich durch die Verwendung von Informations- und Kommunikationstechnologien für den Transport von Personen und Waren Güter aus. Dies geschieht unter der sorgsamen und effizienten Nutzung der gebotenen Verkehrsinfrastruktur und dessen Teilnehmern (wie Autos, Züge, Flugzeuge und Schiffe).

Mittels Car2X soll die Erhöhung der Sicherheit (Unfallvermeidung, etc.) von Verkehrsteilnehmern erreicht werden. Gleichfalls spielen ökologische Aspekte, wie die Reduzierung von Abgasen, eine wichtige Rolle. Dabei werden wirtschaftliche Aspekte ebenso betrachtet, wie beispielsweise kürzere Wege vom Startpunkt bis zum Ziel oder das Umfahren von Stausituationen. Zusätzlich besteht die Hoffnung, dem weiter ansteigenden Verkehrsaufkommen durch eine effizientere Fahrweise zu begegnen. Dies zielt in Richtung Verkehrsmanagement und Optimierung des Verkehrs.

Für den Fahrzeughbereich existieren verschiedene Stadien, in denen der Autofahrer über Gefahren informiert wird. Darüber hinaus helfen Fahrerassistenzsysteme kritische Situationen zu meistern. Ist ein Unfall nicht mehr zu vermeiden, setzen Mechanismen ein, welche die Fahrzeuginsassen schützen sollen. Nach einem Unfall kommen weitere Verfahren hinzu, mit denen das Überleben der Insassen sichergestellt werden soll.

In Abbildung 3.1 ist der Informationsfluß und die dazugehörigen Anwendungen zu sehen, die zur Sicherheit beitragen.

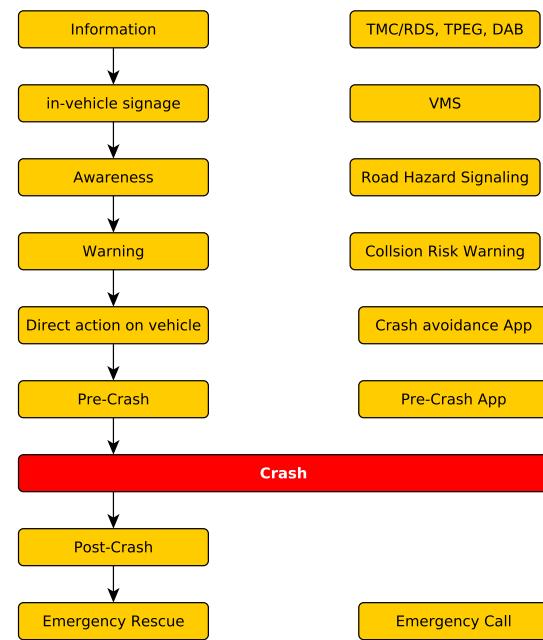


Abbildung 3.1: Informationsablauf

Die Fahrzeugkommunikation findet sich dabei in vielen Bereichen vor und nach einem Unfall wieder. Zur Abwehr von Gefahrensituationen sind diesbezüglich drei wesentliche Bereiche definiert, die Informations-, Warn- und Manöverzone, in denen die Fahrzeugkommunikation aktiv ist.

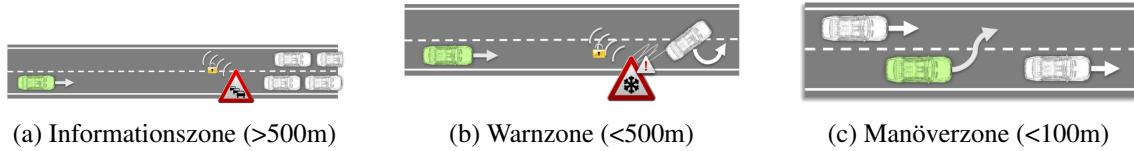


Abbildung 3.2: Typische Bereiche für Car2X

Der Austausch von Informationen zwischen den einzelnen Teilnehmern eines ITS ist ein Grundbaustein für die Realisierung eines solchen Systems.

Im europäischen Raum findet hierzu der Nahbereichskommunikations-Standard, basierend auf IEEE 802.11p beziehungsweise ITS-G5A seine Anwendung. Dieser funktioniert im Frequenzbereich von 5,855 GHz bis 5,925 GHz und lässt es zu, Daten bei einer Geschwindigkeit von 200 km/h auf eine Entfernung von 1000 Metern zu senden. Eine generelle Kommunikationsreichweite wird mit 300m beziffert. Verwendet werden jeweils zwei 10 MHz Kanäle, einer für den Nachrichtentransport von Service-Mitteilungen und der andere für die Kommunikation mit RSUs.

Weltweit finden Forschungen und Entwicklungen im Bereich der Fahrzeugkommunikation statt. Derzeit gibt es national gesehen noch einige Unterschiede.

	Japan	USA	Europa
<b>Standard/Komitee</b>	ITS-Forum	IEEE802.11p/1609.x	CEN/ETSI EN302-663
<b>Frequenzbereich</b>	755 - 765 MHz	5850 - 5925 MHz	5855 - 5925 MHz
<b>Anzahl der Kanäle</b>	ein 10 MHz Kanal	sieben 10 MHz Kanäle <sup>1</sup>	sieben 10 MHz Kanäle
<b>Modulation</b>	OFDM	OFDM	OFDM
<b>Datenrate pro Kanal</b>	3-18 Mbit/s	3-27 MBit/s	3-27 MBit/s
<b>Leistung</b>	20 dBm (Antennen-eingang)	23-33 dBm (EIRP)	23-33 dBm (EIRP)
<b>Kommunikation</b>	unidirektonaler multicasting service (Broadcast ohne ACK)	unidirektonaler multicasting service, ein Kanal für multi communication, simplex communication (Broadcast ohne ACK, Multicast, Unicast, mit ACK)	
<b>Protokolle</b>	ARIB STD-T109	WAVE (IEEE 1609) / TCP/IP	ETSI EN 302 665 (inkl. GeoNetworking) TCP/UDP/IP

Tabelle 3.1: Weltweite Fahrzeugkommunikation (Quelle: [28])

Wünschenswert ist hier eine Vereinheitlichung der Technik um einen globalen Standard.

### 3.2 Anwendungsbereiche

In den Vorstellungen der Entwickler von Fahrzeugkommunikationssystemen sollte die C2X-Technik vordergründig eingesetzt werden, um andere Verkehrsteilnehmer, die sich in einem bestimmten Umkreis befinden, über gewissen Gefahren oder Ereignisse aufmerksam zu machen.

Die möglichen Einsatzszenarien lassen sich in die Kategorien Sicherheit, Effizienz, Unterhaltung und Geschäftlich gliedern.

Unter **Sicherheit** werden Aspekte wie die Signalisierung von Einsatzfahrzeugen betrachtet, gleich ob es sich um einen Krankenwagen, Feuerwehr- oder Polizeifahrzeug handelt. Besondere Aufmerksamkeit liegt hierbei im Straßenverkehr bei Spurwechseln, Überholvorgängen oder uneinsichtigen Kreuzungssituationen, die häufig zu Unfällen führen. Speziell für Autobahnen existieren Konzepte zur Wahrnehmung von Falschfahrern (den sogenannten Geisterfahrern), Baustellen, Personen auf der Fahrbahn, liegen gebliebenen Fahrzeugen, voraus befindlichen Unfällen oder Staus. Innerstädtisch sollen andere Autofahrer aufmerksam gemacht werden, wenn Straßenteilnehmer Stopp-Schilder oder rote Ampeln missachten.

Nicht nur die Sicherheit steht als alleiniges Merkmal für die Fahrzeugkommunikation, gleichfalls ist ein **effizienterer Verkehr gewünscht**. Die zugrunde liegende Ideen sind der Verkehrs-Situation angepaßte Geschwindigkeitsregularien oder das optimale Umschalten von Ampeln, um den Effekt einer „Grünen Welle“ zu optimieren. Das schließt insbesondere die Bereitstellung von Daten zur jeweiligen Verkehrssituation mit ein. In besonderen Bereichen wie Parkhäusern oder generell Parkplätzen, sollen Fahrzeuge informiert werden, ob am Zielort eine Stellfläche frei ist oder eine Alternative existiert. Dadurch soll die Suche nach einem Parkplatz reduziert und somit Zeit und Treibstoff eingespart werden. Auf Autobahnen besteht die Überlegung, dass Fahrzeuge mit dem gleichen Fahrziel in einer Art Kollone fahren, wodurch mittels des Windschattens ebenfalls Kraftstoff eingespart werden kann.

Zum **Unterhaltungswert** trägt die Fahrzeugkommunikation bei, indem Fahrzeuginsassen in bestimmten Gebieten vorab über interessante Sehenswürdigkeiten informiert werden. Genauso können mediale Inhalte (Filme, Musik, etc.) angeboten werden. (Auch wenn dies nicht direkt mit der Fahrzeugkommunikation zusammenhängt, sondern eher Daten aus einer Cloud kommen). Ebenso kann es für die alltägliche Kommunikation (SMS, IM, VoIP) genutzt werden.

Im Bereich des **geschäftlichen Betätigungsgebietes** existieren Konzepte zur Vermietung oder zum Teilen von Fahrzeugen, wie auch der automatischen Gebühren-Abwicklung an Mautstellen. Aber auch die Aktualisierung der Fahrzeugsoftware und Ferndiagnosen über den Zustand des Fahrzeugs mit eventuellen Reparatur-Meldungen sind nicht unüblich.

Weitere wesentliche Punkte sind die Alarmierung über ein gestohlenes Fahrzeug oder eine Hilfe-Ruf-Funktion in Notsituationen.

### 3.3 Europäische Projekte

Die Aufgabe zur Planung, Konzeptionierung und Errichtung einer Infrastruktur für die Fahrzeugkommunikation kann nicht von einer Institution allein getragen werden. Es beteiligen sich Fachleute aus verschiedenen Bereichen mit diesem Thema. In erster Linie engagieren sich hierzulande die Automobilhersteller, Zuliefererfirmen, Forschungseinrichtungen sowie Universitäten in diesem Bereich.



(a) Entwicklungspartner

(b) Automobilhersteller

(c) Zulieferer

Abbildung 3.3: Entwicklungsmitglieder (Quelle: C2C-CC)

Ein Dreh- und Angelpunkt für die Unterstützung zur Umsetzung und Erforschung von Nutzungsmöglichkeiten länderübergreifender Vernetzung von Fahrzeugen und Infrastruktur ist das CAR 2 CAR Communication Consortium (C2C-CC).

Das Konsortium [59] wurde 2003 von mehreren europäischen Automobilherstellern als eine nicht kommerzielle Organisation gegründet. In Zusammenarbeit mit Standardisierungs-Organisationen, unterstützt es das Vorantreiben der Erschaffung europäischer Standards in Bezug auf die Fahrzeugkommunikation.



Neben der reinen Forschung mit Modellen, die in Simulationsumgebungen getestet werden, existieren verschiedene Testfelder in Europa. Ihre Daseinsberechtigung besteht in der Erprobung der Wirksamkeit, Alltagstauglichkeit und Funktionalität der Fahrzeugkommunikation unter realen Aspekten.

In Deutschland besteht hierzu das Forschungsprojekt: Sichere Intelligente Mobilität - Testfeld Deutschland (SIM-TD).

Es kommen die Ergebnisse vorangegangener Forschungsprojekte zur Anwendung. Dazu werden realitätsnahe Szenarien in einer großflächigen Testfeld-Infrastruktur rund um die hessische Metropole Frankfurt adressiert und die politischen, wirtschaftlichen und technologischen Rahmenbedingungen für



eine erfolgreiche Einführung der Fahrzeug-zu-Fahrzeug und Fahrzeug-zu-Infrastruktur-Vernetzung vorbereitet.

### 3.4 Der Weg zum Standard

Die Umsetzung von der Idee bis zum anerkannten Standard durchläuft mehrere Stufen.

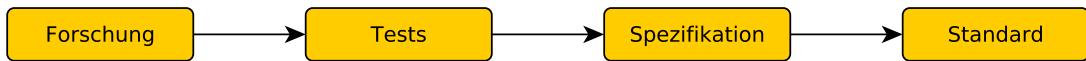


Abbildung 3.4: Weg zum Standard

Zunächst definieren und formen einzelne Forschungsgruppen die grundlegenden Strukturen der Protokolle und Anwendungen. Die Technologien werden in größeren Testfeldern (wie simTD) auf Anwendbarkeit und Reife getestet. Die Ergebnisse und Spezifikationen werden vom C2C-CC gesammelt und bei einer Standardisierungsstelle vorgelegt, welche bevollmächtigt ist einen Standard zu entwickeln der europaweit oder weltweit anerkannt wird. Für den europäischen Raum ist hier die European Telecommunications Standards Institute (ETSI) ein Ansprechpartner.



## 4. ETSI Grundlagen

“ Alle Menschen streben von Natur nach Wissen. ”

Aristoteles, *Metaphysik I* 21, 980a

Die European Telecommunications Standards Institute [60] ist verantwortlich für die Entwicklung und Verbreitung von weltweit anwendbaren Standards für den Sektor von Informations- und Kommunikationstechnologien. Die Standards ermöglichen Technologien im gesellschaftlichen sowie geschäftlichen Bereich. Es handelt sich um eine von der Europäischen Union anerkannte Organisation zur Veröffentlichung von europäischen Standards, sie handelt gemeinnützige ohne Einbehalt von Gewinnen und zählt derzeit um die 800 Mitglieder.

In diesem Abschnitt werden die von der ETSI definierten Spezifikationen zum Aufbau eines ITS erläutert.

## 4.1 ETSI Standards

Das ETSI produziert eine Reihe von Standards, Reports und Spezifikationen, die zu unterschiedlichen Zwecken angefertigt werden. Die verschiedenen Arten der Standards werden aufgegliedert in

- EN - European Standard
- ES - ETSI Standard
- EG - ETSI Guide
- TS - ETSI Technical Specification
- TR - ETSI Technical Report
- SR - ETSI Special Report
- GS - ETSI Group Specification

Die unterschiedlichen Arten von Standards werden je nach Zweck und Zeit angefertigt. Sie werden unter übereinstimmender Meinung aller Beteiligten hergestellt.

## 4.2 Kommunikations-Architektur

In Bezug auf das Thema Fahrzeugkommunikation, hält die ETSI als Einstiegspunkt die in [29] beschriebene Spezifikation parat. In ihr wird die grundlegende Kommunikationsarchitektur für eine ITS beschrieben.

Sie ist nicht nur für die Kommunikation von Fahrzeugen ausgelegt, sondern bildet vielmehr ein Anwendungsbereich ab, mit den verschiedene Elemente (ITS-Station (ITS-S)) innerhalb eines Transport-Systems kommunizieren können. Diese Art der Kommunikation wird auch allgemein als ITS Communications (ITSC) bezeichnet. Innerhalb einer ITSC existieren zwei Domänen. Die „ITS Domäne“ beinhaltet

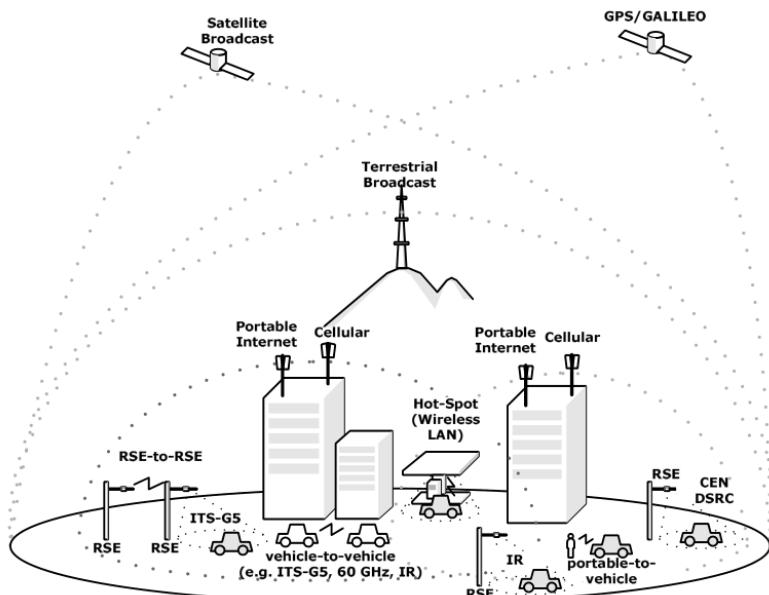


Abbildung 4.1: ITSC Szenario (Quelle: ETSI)

Elemente, die in dem ITS/ITSC Standard definiert sind, und eine „generische Domäne“, dessen Elemente nicht spezifiziert sind, dennoch benutzt werden für ITS/ ITSC.

### 4.3 Die Referenz Architektur

Mit der Referenz-Architektur werden die einzelnen funktionalen Bestandteile einer jeden ITS-S spezifiziert. Sie folgt den Prinzipien des Open System Interconnection (OSI) Schichtenmodells, um die einzelnen Protokolle voneinander zu trennen. Der Austausch von Informationen wird über Schnittstellen oder sogenannter Service Access Point (SAP) geregelt.

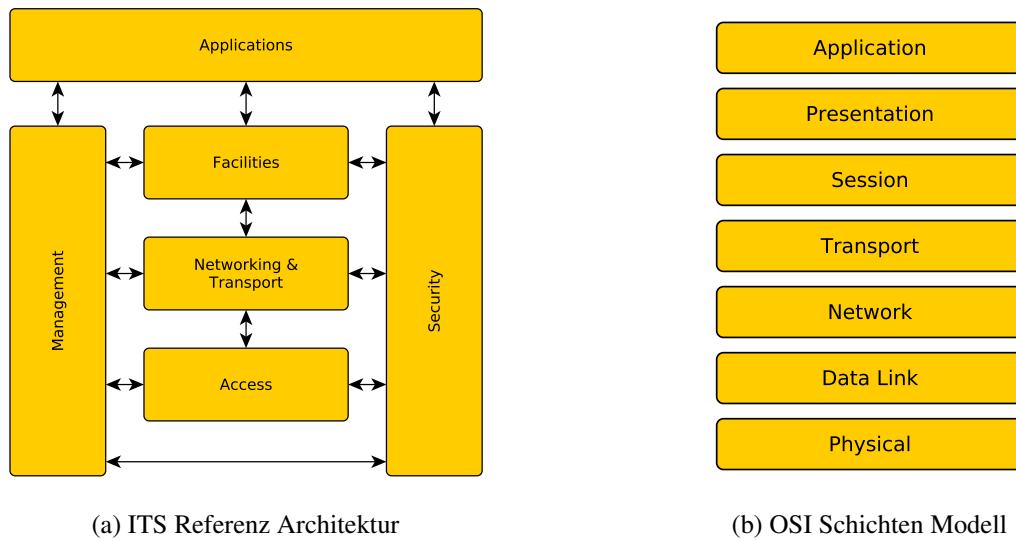


Abbildung 4.2: Architektur-Modelle

Der funktionale Vergleich mit dem OSI Schichtenmodell äußert sich vor allem bei der ITS Referenz Architektur in den Schichten **Access** (OSI 1 und 2), **Networking & Transport** (OSI 3 und 4) und **Facilities** (OSI 5, 6 und 7).

Die einzelnen Schichten werden im Weiteren näher erläutert.

#### 4.3.1 Application-Schicht

Die „Applications (Anwendungs)-Schicht“ ist ursprünglich unterteilt in die Bereiche „Road Safety“, „Traffic Efficiency“ und „Other Applications“. Eine Anwendung besteht in der Regel aus zwei oder mehr, sich ergänzenden Anwendungen. Häufig wird dies durch eine Server- und Client-Rolle repräsentiert. Die Anwendungen sollten so konzipiert sein, dass sie zuverlässig, sicher und ohne große Latenzen funktionieren. Manche Anwendungen werden vorrangig behandelt, um die Betriebsfähigkeit und Funktionalität einer ITS Anwendung zu gewährleisten.

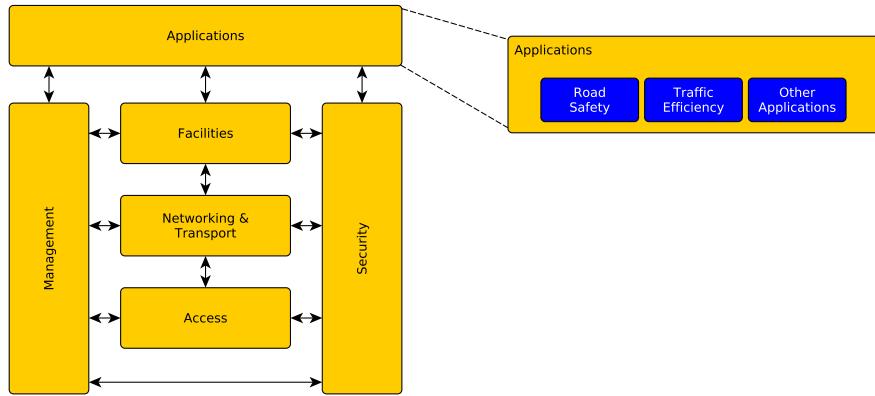


Abbildung 4.3: ITS Referenz Architektur - Anwendungs-Schicht

#### 4.3.2 Access Layer

Die Zugangs-Schicht (Access Layer) wird differenziert in eine physikalische Schicht, die physikalisch mit dem Kommunikations-Medium verbunden ist, und einer „Data-Link“-Schicht, die den Zugriff auf das Kommunikationsmedium verwaltet. Nach dem ITSC-Standard werden eine Vielzahl verschiedener Zugangs-Technologien unterstützt. Darunter fallen drahtgebundene Technologien wie Ethernet oder drahtlose wie WLAN oder Bluetooth.

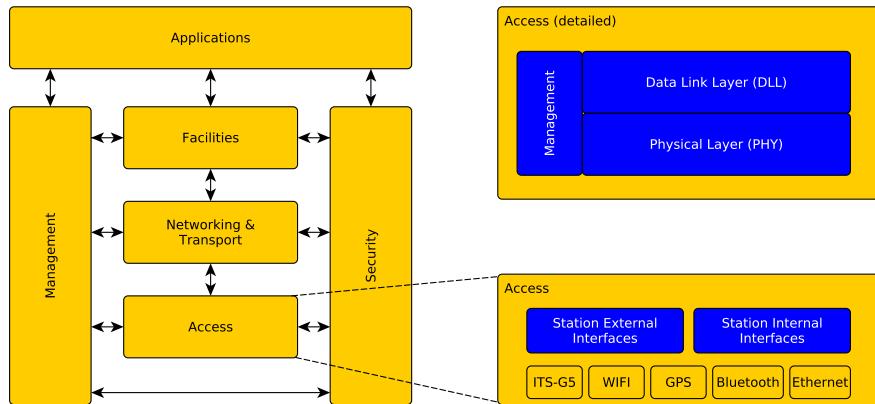


Abbildung 4.4: ITS Referenz Architektur - Access-Schicht

#### 4.3.3 Networking & Transport Layer

Diese Schicht beinhaltet im wesentlichen die bekannten Funktionalitäten der Netzwerk und Transport Schicht des OSI-Schichtenmodells. Es wurde für die Verwendung mit ITSC um weitere Netzwerk- und Transportprotokolle erweitert. Unterstützung finden auf der Netzwerk-Ebene die Protokolle: GeoNetworking [30, 31, 32], IPv6 mit Mobilitätsunterstützung [33], IPv6 mittels GeoNetworking [30] und weitere.

Die einzelnen Netzwerkprotokolle sind wiederum verbunden mit den Transportprotokollen,

entweder mit einem zugehörigen ITSC Transportprotokoll oder einem bereits existierenden, wie User Datagram Protocol (UDP), Transmission Control Protocol (TCP) oder IPv6. Zusätzlich umfasst das IPv6 Netzwerk Methoden zur Unterstützung von älteren IPv4 Systemen.

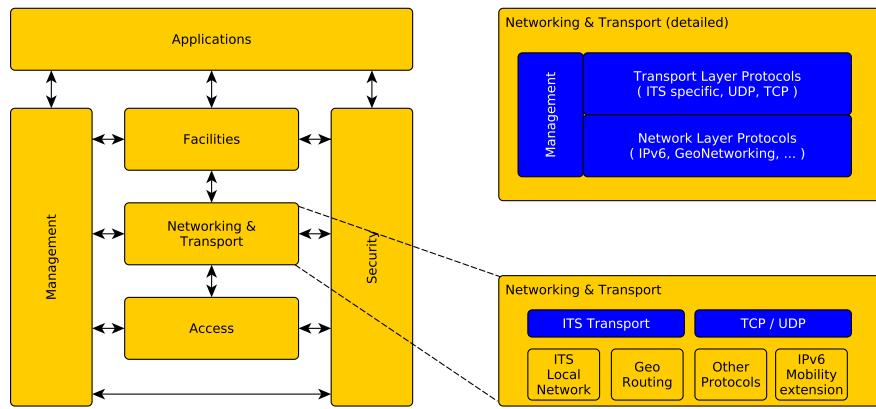


Abbildung 4.5: ITS Referenz Architektur - Networking&Transport-Schicht

#### 4.3.4 Facilities Layer

Der Facilities-Layer ist funktional gleichzusetzen mit der OSI Anwendungs-, Präsentations- und Session-Schicht. Der Tätigkeitsbereich wurde zur Unterstützung für die ITSC erweitert.

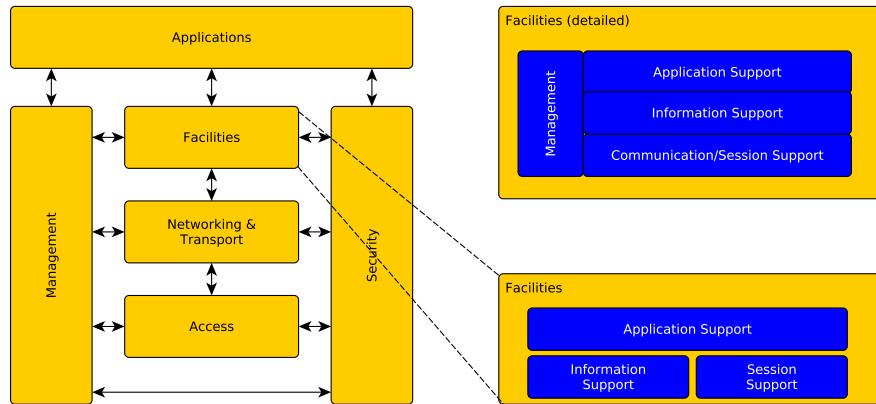


Abbildung 4.6: ITS Referenz Architektur - Facilities-Schicht

In dieser Ebene werden Informationen vom System für den Benutzer bereitgestellt, welche dann über ein Human Machine Interface (HMI) abgerufen werden können.

Weiterhin werden Informationen zur geographischen Position (Longitude, Latitude, Altitude) sowie die aktuelle Uhrzeit bereitgestellt und verwaltet. Dies dient der Unterstützung, um ausgehende Daten mit einem Zeitstempel und einem Ortsverweis zu versehen. Das Verfahren wird als „Location Referencing“ bezeichnet.

Das Gegenstück dazu ist das „Relevance Checking“. Eingehende Nachrichten werden auf ihre Relevanz im derzeitigen Kontext geprüft. Dies geschieht im Zusammenspiel mit der Local Dynamic Map (LDM), einer Datenbank, welche die Positionen und Ereignisse von dynamischen Objekten beinhaltet. Die LDM befindet sich ebenfalls in der „Facility“-Schicht und ist mit den enthaltenden Daten eine Art Karte der Umgebung mit den aktuellen Vorkommnissen.

Einer der wichtigsten Aspekte dieser Ebene ist die Verwaltung für den Nachrichtenaustausch. Dabei gibt es drei grundlegende Typen von Nachrichten.

**Ereignis-basierte**, die ausgelöst werden sobald ein Ereignis auftritt. Diese Nachrichten werden solange wiederholt, solange das Ereignis besteht. Bekannte Vertreter für diese Art von Nachrichten sind die Decentralized Environmental Notification Message (DENM) und die Transport Protocol Expert Group - Road Traffic Message (TPEG-RTM). **Periodisch-basierte** werden mit einer bestimmten Frequenz wiederholt ausgesendet. Hier ist vor allen die Cooperative Awareness Message (CAM) zu nennen, welche die Position, Richtung und Geschwindigkeit einer ITS-S beinhaltet. Die Frequenz, mit der die Nachricht versendet wird, unterliegt der Kontrolle der Facility-Schicht. In der Regel gibt es eine feste Frequenz, die durch Umwelteinflüsse, wie der Geschwindigkeit mit der sich eine ITS-S fortbewegt, variieren kann. Als letzte Art von Nachrichten existieren **Service-basierte**, die Sessions verwalten. Beispiele hierzu finden sich in [34].

Für die Kommunikation werden zudem geeignete Kommunikationskanäle ermittelt und ausgewählt. Mit dem Konzept der Service Oriented Architecture (SOA) werden netzwerk basierte Dienste unterstützt und angeboten. Wodurch Software-Aktualisierungen eingespielt und vorhandene aktualisiert werden kann.

Nebenbei werden in der Schicht die Eigenschaften und Fähigkeiten einer ITS-S verwaltet. Das entscheidet darüber, um welche Art von Station es sich beispielsweise handelt (Fahrzeug oder Infrastruktur).

#### 4.3.5 Management Layer

Die „Management“-Schicht stellt die Verwaltungsstruktur für die Kommunikation in einer ITS-S zur Verfügung und gibt Zugriff auf die Management Information Base (MIB) [35], die es erlaubt einzelne Parameter, die das Verhalten oder die Eigenschaften einer ITS-S bestimmen, zu ändern.

Die Management-Schicht übernimmt die Aufgaben der Verwaltung verschiedenster Anwendungen. Darunter fällt die Auslastungssteuerung, mit welcher die Frequenz des Versands von Datenpaketen, die über eine bestimmte Schnittstelle geschickt werden soll, gesteuert wird. Dies ist insbesondere wichtig, da die physische Bandbreite der Kommunikationskanäle limitiert ist.

Des Weiteren werden die Kommunikations-Dienstleistungen mit denen beispielsweise der

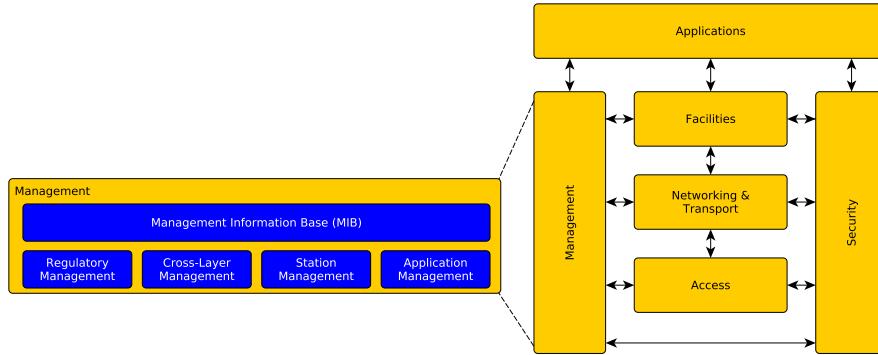


Abbildung 4.7: ITS Referenz Architektur - Management-Schicht

Aktualisierung-Prozess neuer ITS Anwendungen geregelt wird, überwacht und gesteuert.

Ein weiterer Punkt ist die Stations-Verwaltung, welche Informationen zu anderen, in der Nähe befindlichen, ITS-S in einer Local Node Map (LNM) speichert. In dieser Karte werden die Kommunikations Parameter wie MAC- und Netzwerkadressen, zusammen mit den kinematischen Zustandsvektoren wie Position, Ausrichtung und Geschwindigkeit hinterlegt.

In [29] werden noch weitere Funktionen der Management-Schicht dargestellt.

#### 4.3.6 Security Layer

Mit der „Security“-Schicht werden sicherheitsrelevante Dienste für den Kommunikationsaustausch bereitgestellt.

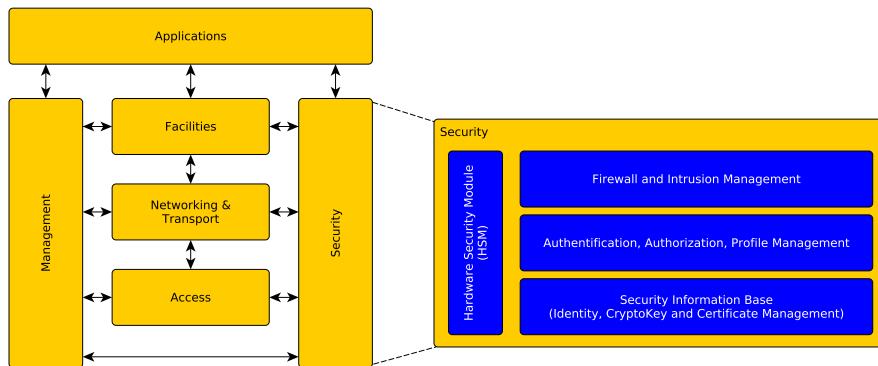


Abbildung 4.8: ITS Referenz Architektur - Security-Schicht

Die Aufgaben bestehen in der Verwaltung der Firewall und der Erkennung und Vermeidung von unerlaubten Zugriffen auf das System. Hierzu werden Authentifizierungs- und Autorisierungsmechanismen genutzt. Dies wird umgesetzt durch die Verwendung von Zertifikaten und kryptographischen Schlüsselmechanismen zur Identifikation einer berechtigten Identität.

#### 4.4 Elemente eines ITS

In der Spezifikation [29] [36] werden vier funktionale Systeme/Stationen erwähnt. Jedes dieser Systeme besteht aus einer ITS-S, dessen Aufbau sich nach dem ITS-S Referenz-Design richtet. Diese Systeme können weitere Unterelemente beinhalten, wobei die Funktionalität sich entweder in einer oder mehrerer physikalischen Einheiten befinden kann.

##### 4.4.1 Personal Station

Von einer „Personal-Station“ ist die Rede, wenn die Funktionalität der ITSC sich in tragbaren Geräten, wie einem Smartphone, Personal Digital Assistant (PDA) oder dergleichen, befindet. Sie dienen als mobile Geräte die als eigenständige ITS-S fungieren oder als HMI zur Erweiterung einer vorhandenen ITS-S.

##### 4.4.2 Central Station

Eine „Central Station“ ist beispielsweise ein Gebäude und fungiert entweder als ein Gateway oder ein Router. Sie arbeitet als eine Art Übersetzer für verschiedene Kommunikationsprotokolle zwischen einem Kommunikationspartner und der ITS-S. Die „Central Station“ bewerkstelligt verschiedene Aufgaben, wie die Koordination des Straßenverkehrs oder die Bereitstellung von Informationen und Dienstleistungen.

##### 4.4.3 Vehicle Station

Bei der „Vehicle Station“ befindet sich die Funktionalität einer ITS-S innerhalb eines Fahrzeugs. Neben der reinen ITS Referenz-Architektur beinhalten Fahrzeuge zudem ebenso ein Router-Modul zur Anbindung an die Kommunikationshardware. Ebenso ist ein Gateway für den Datenaustausch mit der proprietären Fahrzeugelektronik vorgesehen.

##### 4.4.4 Roadside Station

Die „Roadside Stations“ entsprechen Verkehrsschildern, Ampeln oder einfachen Kommunikationsgeräten an Fahrbahnen. Sie verwenden ebenfalls Gateway- und Routing-Mechanismen für den Datenaustausch mit einem proprietären Elementen eines Verkehrsnetzes, wie Anzeigetafeln oder Geschwindigkeitsmessungen.

#### 4.5 Kommunikation

Die Kommunikation zwischen den einzelnen ITS-S wird durch ein „Ad-Hoc“ Netzwerk realisiert. Als Protokoll wird von europäischer Seite das „GeoNetworking“-Protokoll [30] verwendet. Es ist in der Netzwerk-Schicht des „ITS“ Referenz Modells beheimatet und ist geschaffen für mobile „Ad-Hoc“-Netzwerke, basierend auf drahtloser Kommunikationstechnologie (beispielsweise ITS-G5).

GeoNetworking selbst verwendet geographische Positionsdaten für das Versenden von Informationen und die Weitergabe von Datenpaketen. Es ermöglicht die Kommunikation über mehrere drahtlose Knoten, wodurch die Kommunikationsreichweite erhöht wird.

Ursprünglich wurde GeoNetworking für die Anwendung von mobilen Ad-Hoc Netzwerken konzipiert. Mittlerweile existieren weitere Varianten für andere Netzwerkarten, darunter fallen VANETs, Mesh-Netzwerke und drahtlose Sensor Netzwerke. Somit lässt sich GeoNetworking auch als eine Art Familie von Netzwerk-Protokollen ansehen, basierend auf der Benutzung von geographischen Positionsdaten für die Adressierung und den Transport von Datenpaketen in verschiedenen Arten von Netzwerken.

In VANETs stellt GeoNetworking die drahtlose Kommunikation zwischen Fahrzeugen und zwischen Fahrzeugen und fest verbauten Stationen entlang der Straße her. Dabei arbeitet GeoNetworking verbindungslos und vollkommen verteilt, basierend auf den Konzepten eines Ad-Hoc Netzwerkes. Die Kommunikation kann durch die ITS Infrastruktur unterstützt werden, sie ist aber nicht zwingend erforderlich.

Die Prinzipien des GeoNetworkings werden dabei den Anforderungen der Umgebung von Fahrzeugen gerecht. Es ist sehr geeignet für hoch mobile Netzwerknoten und schnellen Änderungen in der Netzwerktopologie. Durch die Flexibilität sind heterogene Anwendungen wie die Verkehrssicherheit, Verkehrseffizienz und Infotainment möglich. Um ins Detail zu gehen, ermöglicht es das periodische Aussenden von sicherheitsrelevanten Status-Meldungen, eine schnelle Multi-Hopverbreitung von Datenpaketen in geographische Regionen für Notfallwarnungen und einen Unicast-Pakettransport für Internetanwendungen.

Das GeoNetworking bietet die zwei wesentlichen Funktionen „geographische Adressierung“ und „geographische Weiterleitung“. Anders als bei konventionellen Netzwerken, bei dem ein Kommunikationspartner durch seine Identität (z.B. IP-Adresse) adressiert wird, sendet GeoNetworking Datenpakete an einen Knoten, bestimmt durch seine Position, oder an mehrere Knoten, die sich innerhalb eines bestimmten geographischen Gebietes befinden. Für eine Weiterleitung von Datenpaketen nimmt GeoNetworking an, dass jeder Knoten einen Teilbereich seiner Nachbarschaft sieht, und dass jedes Datenpaket eine geographische Adresse (geographische Position / Bereich) als Ziel enthält. Wenn ein Knoten ein Datenpaket erhält, vergleicht es die enthaltene geographische Adresse mit der ihm bekannten Netzwerktopologie und trifft automatisch die Entscheidung darüber, ob das Paket weitergeleitet werden soll. Das entscheidende dabei ist, dass das Paket ohne jeglichen Eingriff in die Routing-Tabelle des jeweiligen Knotens weitergeleitet wird.

Der effektivste Weg zu Verbreitung von Informationen mittels geographischem Routing, besteht in der gezielten Übertragung einer Nachricht zu einem geographischen Bereich. In der Praxis wählen Fahrzeuge einen geeigneten Bereich aus, zu dem eine Nachricht verschickt wird. Fahrzeuge, die sich zwischen dem aussendenden Fahrzeug und dem Zielbereich befinden, haben eine weitergebende Funktion, sie leiten eine Nachricht lediglich

zum Zielbereich weiter. Nur die Fahrzeuge innerhalb des Zielbereichs werten die Informationen in der Nachricht aus. Somit werden nur Fahrzeuge über eine mögliche Gefahr unterrichtet, die es auch betrifft, alle anderen bleiben unbehelligt.

#### 4.5.1 Kommunikations-Szenarien

Die dargestellten Kommunikationsszenarien sind in [30] und [31] spezifiziert und erläutert. Grundlegend umfasst das geographische Routen die Weiterleitungs-Szenarien „GeoUnicast“, „GeoBroadcast“ und „Topology-scoped broadcast“.

Beim **GeoUnicast** geht es darum, dass ein Paket zwischen zwei Knoten mit Hilfe mehrerer Hops ausgetauscht wird. Das geschieht in der Regel indem ein Knoten, der ein Unicast-Paket senden möchte, zunächst einmal die Zielposition bestimmt. Daraufhin sendet es das Paket an einen Knoten, der sich in Richtung des Zielpunktes befindet. Dieser wiederum leitet das Paket weiter in Richtung des Ziels. Dies geschieht so lange, bis das Paket beim Ziel eingetroffen ist.

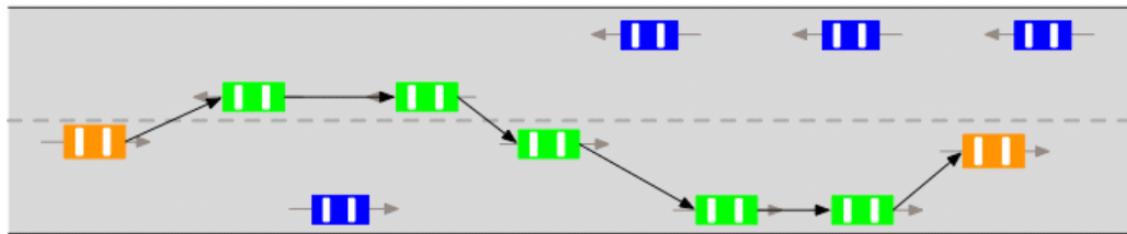


Abbildung 4.9: Unicast (Quelle: ETSI)

Das **GeoBroadcast** funktioniert so, dass ein Paket über mehrere Stationen weitergeleitet wird, bis es zum Zielbereich gelangt. Knoten innerhalb des Zielbereichs verbreiten die Nachricht an andere Knoten innerhalb des Zielbereichs.

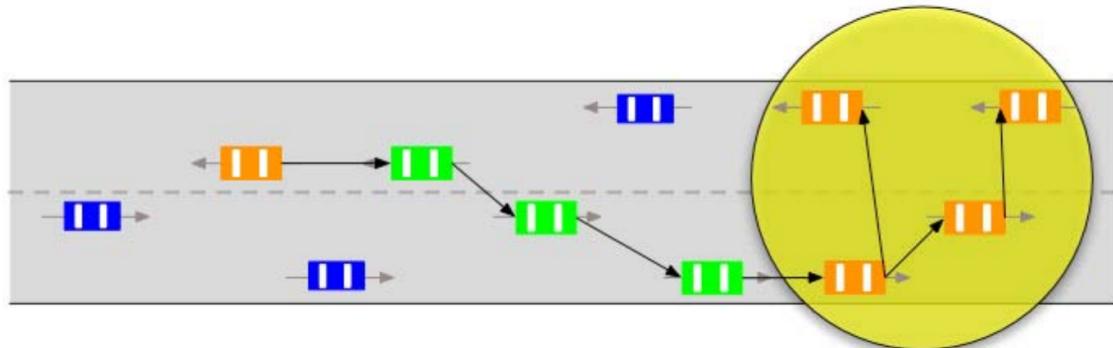


Abbildung 4.10: Geo-Broadcast (Quelle: ETSI)

Mit **Topologie-scoped broadcast** wird ein Datenpaket von einer Quelle zu allen seinen in n-hop Reichweite befindlichen Nachbarn verschickt. Das Single-Hop Broadcasting ist ein Sonderfall vom Topologie-scoped broadcasting, welcher verwendet wird, um ein Paket mit nur einem Hop an seine Nachbarschaft zu verteilen.

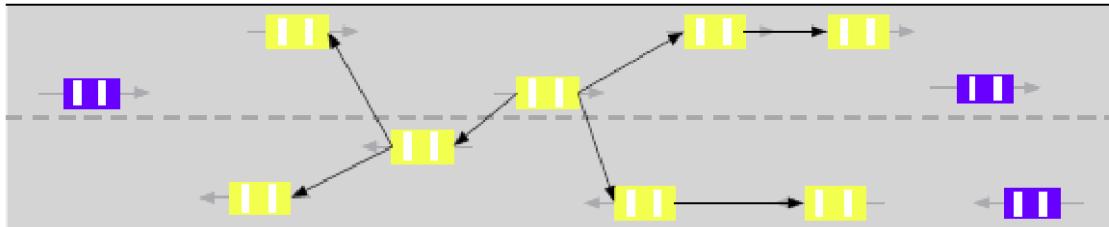


Abbildung 4.11: Topologie Broadcast (Quelle: ETSI)

## 4.6 ASN.1

Alle ITSC Standards beinhalten aussagekräftige, funktionelle Spezifikationen, die für ein klares Verständnis in einer abstrahierten Notation beschrieben werden. Der hierfür eingesetzte Standard nennt sich Abstract Syntax Notation One (ASN.1) mit der speziellen platzsparenden Packed Encoding Rules (PER) Kodierung, die auch im Mobilfunkbereich seine Anwendung findet.

Mit ihr lassen sich abstrakte Beschreibungen von Datentypen definieren, um Elemente einer Nachricht von Protokollen eindeutig festlegen zu können. Dabei ist ASN.1 international standardisiert [2], und auch Plattform- sowie Sprachen-unabhängig.

Es existieren weitgehend für die meisten Systeme Werkzeuge, mit denen sich aus den in ASN.1 beschriebenen Data-Strukturen die jeweiligen Definitionen einer Programmiersprache übersetzen lassen.

In der Literatur [3] wird auf die einzelnen Strukturen von ASN.1 näher eingegangen. Dies betrifft vorrangig den Aufbau und die Funktionsweise. Weiterführend werden die einzelnen Datenelemente erläutert und in Verbindung mit diversen Programmiersprachen in Einklang gebracht.

### 4.6.1 Beispiel

```

1 Record ::= SEQUENCE {
2   name PrintableString (SIZE(1..30)),
3   age INTEGER,
4   gender ENUMERATE { unknown(0),
5                      male(1),
6                      female(2)
7 } }
```

codes asn.1/example.asn

Hier sei beispielhaft eine Struktur definiert, in der Personen mit ihrem Namen, Alter und Geschlecht hinterlegt werden können. Es handelt sich dabei um eine Abstraktion, die in verschiedene Programmiersprachen konkretisiert werden kann.

Eine Implementierung in C würde zu folgendem Resultat führen.

```

1 typedef struct Record {
2   char name[31];
3   int age;
4   enum { unknown=0, male=1, female=2 } gender;
5 } Record;
```

codes asn.1/example.c

### 4.6.2 Datentypen

Dieser Abschnitt zeigt einige der gebräuchlichsten Datentypen, die in ASN.1 zum Einsatz kommen.

Typ	Beschreibung
BOOLEAN	Datentyp mit den Werten TRUE oder FALSE
BIT STRING	Wird verwendet für Arrays die kein Vielfaches von 8 sind
CHOICE	Auswahl eines Elements, wenn mehrere zur Verfügung stehen
INTEGER	Darstellung von ganzzahligen Werten jeglicher Größenordnung
OCTET STRING	Findet Verwendung bei Daten die einen Teiler von 8 besitzen.
DATE	Darstellung von Datum-Informationen
DATE-TIME	Darstellung von Zeit- und Datumsinformationen
REAL	Darstellung von Gleitkommawerten
ENUMERATED	Darstellung von Aufzählungen
NULL	Findet Verwendung als Platzhalter, wenn es keine Werte gibt
SEQUENCE	Gruppieren von mehreren Elementen

Tabelle 4.1: Datentypen von ASN.1

Es existieren noch weitere Datentypen, darunter fallen auch Elemente, die man nicht

verwenden sollte. Eine vollständige Übersicht bietet die Einführung zu ASN.1 von ITU [61].

#### 4.6.3 Kodierregeln

Neben der bereits oben angedeuteten Kodierung PER, unterstützt ASN.1 eine Reihe von Kodierregeln. Sie dienen dem Zweck, Nachrichten für die Kommunikation mit anderen Geräten zu spezifizieren. Dabei unterstützen sie verschiedene Merkmale, die sich in der Kodiergeschwindigkeit oder der Kompaktheit einer Nachricht auszeichnen.

Bei der am längsten, bekanntesten Kodierungsregel handelt es sich um **Basic Encoding Rules (BER)**. Sie verwendet ein Format zur Kodierung, das mit einem „TAG“ beginnt, mit dem gekennzeichnet wird, welche Daten als nächstes folgen. Nach dem TAG werden eine Längenangabe und die eigentlichen Daten übermittelt. Die Längenangabe bezieht sich dabei auf die Länge der folgenden Daten. Bekannt ist das Format unter dem Namen Tag-Length-Value (TLV). Abgeleitet vom BER existieren die Regeln Distinguished Encoding Rules (DER) und Canonical Encoding Rules (CER), die nicht ganz so flexibel sind, aber wie das DER in speziellen sicherheitsrelevanten Protokollen wie dem digitalen Zertifikat X.509 eingesetzt werden.

Die **PER** liefern ein kompaktes Datenformat. Es wird durch den Wegfall der TAG Information, wie sie beim BER gebräuchlich ist, erreicht. Dies ist möglich, da die Reihenfolge der eintreffenden Daten bekannt ist. Ebenso wird keine Längeninformation mitgesendet, weil die Daten eine feste Länge besitzen. Eine weitere Strategie zur Reduzierung der Datenmenge, liegt in der Eliminierung redundanter Daten. Die Kodierung erfolgt entweder *aligned* (ausgerichtet) mit einer auf 8-Bit festgelegten Datengröße oder *unaligned* (unausgerichtet). Letzteres findet auch Anwendung in der 3GPP Technologie wie sie bei den Kommunikationsprotokollen Universal Mobile Telecommunications System (UMTS) oder Long Term Evolution (LTE) eingesetzt wird.

Zur Beschleunigung der Kodierung wurden die **Octet Encoding Rules (OER)** eingeführt. Es findet eine Datenreduktion wie bei PER statt, mit einer festen Breite der Datenfelder. Hierdurch wird eine höhere Geschwindigkeit beim Kodieren und Dekodieren erreicht.





## 5. Cooperative Awareness

“ Aber obwohl der Sinn gemeinsam ist, leben die Vielen, als hätten sie eine eigene Einsicht. ”

---

Heraklit, 520 - 460 v.Chr.

Dieser Abschnitt präsentiert einen Überblick zu Cooperative Awareness. Anschließend werden die zwei wesentlichsten Nachrichten-Typen in diesem Bereich erläutert. Zusätzlich wird ein Mechanismus zum Archivieren und Verwalten der Nachrichten vorgestellt.

Man spricht von einem „Kollegialen Bewusstsein (Cooperative Awareness)“ innerhalb eines Verkehrsnetzes, wenn die Verkehrsteilnehmer als auch die Verkehrsinfrastruktur Daten austauschen.

Dabei informieren sich die einzelnen Parteien gegenseitig über ihre aktuelle Position, Dynamik und die Attribute. Dabei spielt es keine Rolle, ob es sich um Autos, Fahrräder, Fußgänger, Ampeln oder Verkehrsschilder handelt. Das Bewusstsein und die Rücksicht jedes einzelnen ist von fundamentaler Bedeutung für die Sicherheit und Effizienz im Straßenverkehr.

Die Informationen, die von jedem ITS-Teilnehmer ausgesendet werden, um ein *kollektives Bewusstsein* zu schaffen, werden entweder periodisch (im Falle von CAM) oder ereignisbasiert (DENM) übertragen.

Anwendungen, die sich aus der Kommunikation ergeben, finden sich in [37]. Ein Hauptaugenmerk liegt hierbei in der aktiven Sicherheit auf der Straße. Hierunter fallen Assistenzsysteme, die vor liegen gebliebenen oder zu langsam fahrenden Fahrzeugen warnen. Ebenfalls wird vor Kollisionsgefahren an Kreuzungen informiert. Weitergehend wird ein effizienterer Verkehrsfluss, durch das Setzen von angemessenen Geschwindigkeitsregularien oder dem intelligenten Schalten von Verkehrsampeln angestrebt.

Für den Tourismus oder ortsunkundige Personen sind lokale Dienstleistungen vorgesehen. Diese können interessante Orte beim Vorbeifahren durch Einblendung von Informationen kenntlich machen. Gleichfalls werden Maut-Gebühren elektronisch erfasst, ohne ein Beschränkungssystem einsetzen zu müssen. Genauso lassen sich automatisch freie Parkplätze identifizieren und Parkgebühren abrechnen.

## 5.1 Cooperative Awareness Message

Wie eingangs schon angedeutet, handelt es sich bei einer CAM (beschrieben in [38] als europäischer Standard) um eine periodisch versandte Mitteilung. Sie wird von allen ITS-S verschickt, die sich im ITS-Netzwerk befinden. Dies sorgt dafür, dass jeder Teilnehmer über den Anderen informiert ist, der sich im Funkbereich befindet.

Das Nachrichtenformat ist flexibel gehalten um zukünftige Erweiterungen für weitere ITS-S oder Anwendungen einfach adaptieren zu können.

Eine CAM beinhaltet unterschiedliche Informationen, die je nach Art der ITS-S (siehe 4.4) variieren. Bei Fahrzeugen finden sich Statusinformationen wie Zeit, Position und Bewegungszustand. Hinzu kommen Attribute wie der Fahrzeug-Typ, die Fahrzeug-Dimension und die Rolle (Auto, Krankenwagen, etc.), die das Fahrzeug im Straßenverkehr genauer definieren.

Empfängt ein ITS-Teilnehmer eine CAM wird er mit den oben genannten Informationen versorgt und kann sich so ein Bild seiner Umgebung machen. Gleicht der Empfänger

die eingehenden Informationen mit seinen eigenen Daten ab, könnte er beispielsweise abschätzen, ob es in näherer Zukunft zu einer Kollision kommen könnte. Sollte dies der Fall sein, kann die Boardelektronik den Fahrer warnen oder auch im autonomen Einsatz das Fahrzeug gegensteuern.

### 5.1.1 CA Basic Service

Der Aufbau, die Verwaltung und die Verarbeitung der CAMs wird durch den „Cooperative Awareness Basic Service (CA basic Service)“ geregelt. Dieser ist Teil der „Facility“ Schicht in der ITS-Kommunikations-Architektur [29] (siehe Abschnitt 4.3.4 und Abbildung 4.6) und ist verpflichtend, für alle ITS-S, die am Straßenverkehr teilnehmen.

Der „CA Basic Service“ ist verbunden mit anderen Instanzen der Facility- und Anwendungs-Schicht, um relevante Informationen für das Erstellen einer CAM zu sammeln. Aber auch um empfangene CAM Informationen zur weiteren Verarbeitung aufzubereiten.

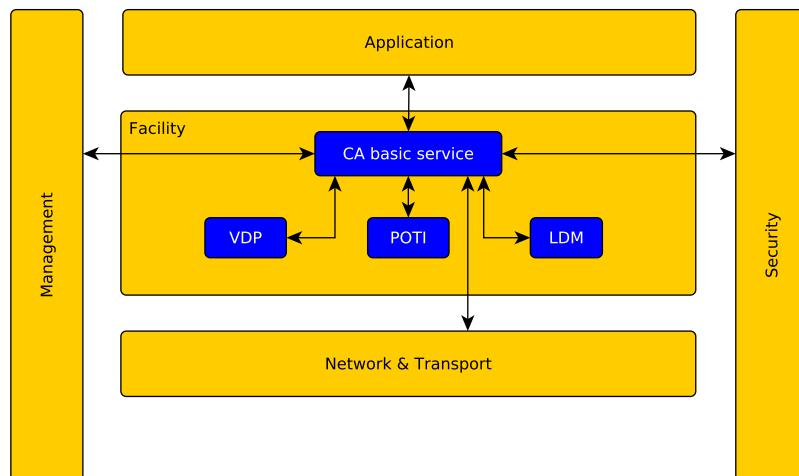


Abbildung 5.1: Kommunikationskanäle CA basic Service

Die Spezifikation des „CA basic service“ wurde ursprünglich von dem europäischen „Car-to-Car Communication Consortium“ entwickelt. Die Anforderungen bezüglich der Leistungsfähigkeit die an den „CA Basic Service“ gestellt werden, als auch der Inhalt und die Qualität der Datenelemente einer CAM, sind abgeleitet vom „Basic Set of Applications (BSA)“ [37] und im Einzelnen von den in [39] [40] [41] definierten Verkehrssicherheits-Anwendungen.

Unter Heranziehung der ITS-Referenz-Architektur (siehe Abbildung 4.2a), befindet sich der „CA Basic Service“ in der *Facility-Schicht* und sorgt für die Umsetzung des CAM-Protokolls. Dies hat die Bereitstellungen zweier grundlegender Funktionen zu folge, dem Senden und Empfangen von CAMs. Hierzu verwendet der „CA Basic Service“ die bereitgestellten Funktionalitäten der „Netzwerk- und Transport-Schicht“, um eine CAM zu verbreiten.

Es existieren verschiedene Szenarien, wann der „CA Basic Service“ aktiviert wird. Dies entscheidet sich mit der Art der ITS-S. Bei Fahrzeugen wird er mit dem Einschalten der Boardelektronik gestartet und beendet seine Aktivität beim Ausschalten. Solange der „CA Basic Service“ aktiv ist, wird das Erstellen der CAMs durch ihn verwaltet.

### 5.1.2 Aufbau einer CAM

Die Zusammensetzung einer CAM wird beschrieben in [38, S.20]. Der Grundaufbau einer CAM besteht aus einem *Header* und mehreren *Containern*.

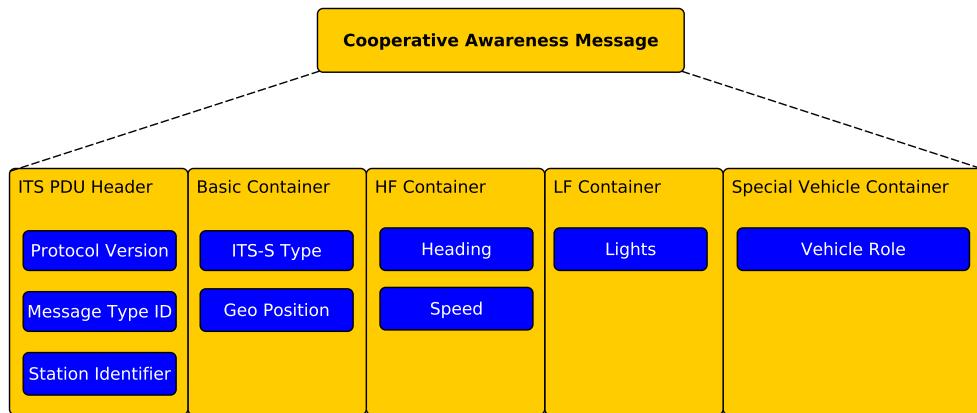


Abbildung 5.2: Struktur einer CAM

Der *Header* (auch *ITS PDU* genannt), spezifiziert in [42], beinhaltet dabei Informationen wie die Version des Protokolls, das verwendet wird, den Nachrichtentyp (in diesem Fall CAM) und eine Identifikationsnummer der ITS-S.

Eine ITS-S vom Typ *Fahrzeug* beinhaltet neben den oben aufgeführten, zusätzlich einen Basic- sowie High Frequency (HF)-Container in der CAM. Abhängig von den Bedingungen sind obendrein ein Low Frequency- und ein oder mehrere Special-Container involviert. Bei einer RSU ist nur der Basic- und ein HF-Container verpflichtend, alle weiteren sind optional.

Der „Basic-Container“ enthält Informationen über die sendende ITS-S. Das kann beispielsweise der Typ und die letzte bekannte geographische Position sein. Dieser Container ist verpflichtend und sollte von jeder ITS-S in eine CAM verpackt werden.

Im „High-Frequency-Container (HF-Container)“ werden Informationen hinterlegt, die eine hohe Dynamik aufweisen. Darunter fallen Daten wie die Ausrichtung des Fahrzeugs oder die Änderung der Geschwindigkeit.

Als Gegensatz steht der „Low Frequency Container (LF-Container)“ parat, welcher Daten beheimatet, die keine große Dynamik aufweisen. Hier könnte es sich beispielsweise um den Status der Außenbeleuchtung handeln. Sie werden erst nach einer gewissen Zeitspanne ( $\geq 500\text{ms}$ ) aktualisiert.

Zu guter Letzt beinhaltet der „Special Vehicle Container“ Informationen über die Rolle eines Fahrzeugs. Eine Rolle legt dabei die Fahrzeugcharakteristik fest. Einige der wichtigsten Rollen sind dabei :

- **Public Transport**, öffentlicher Transportverkehr
- **Special Transport**, spezielle Transportfahrzeuge (bsp.: Schwerlast)
- **Dangerous Goods**, Gefahrentransporte
- **Road Work**, Baustellenfahrzeuge
- **Rescue**, Bergungsfahrzeug ohne Privilegien (Abschleppwagen)
- **Emergency**, Rettungsfahrzeug im Einsatz mit Sonderprivilegien
- **Safety Car**, Fahrzeug zur Vermeidung von Unfällen

### 5.1.3 Versand und Empfang von CAM

Zum Prozess des Versendens einer Nachricht, gehören das Erstellen und Übermitteln einer CAM. Der „CA basic service“ regelt dabei die Häufigkeit, mit der die periodischen CAMs versendet werden. Die Frequenz ist abhängig von der Änderung des Zustands der jeweiligen ITS-S. Ausgehend von einem Fahrzeug, wird die Frequenz beispielsweise durch das Ändern der Position oder der Geschwindigkeit beeinflusst.

Zum Erstellen einer CAM organisiert der „CA Basic Service“ die erforderlichen Daten von der Fahrzeug-Peripherie, die dann in die einzelnen „Container“, wie sie oben beschrieben sind, überführt werden.

Jede Nachricht wird dabei mit einem Zeitstempel versehen, damit wird sichergestellt, dass die CAM beim Empfänger chronologisch richtig interpretiert wird. Dies setzt eine akzeptable Zeit-Synchronisierung zwischen den verschiedenen ITS-Ss voraus. In der Regel werden Zeitsignale von Navigationssatelliten (zum Beispiel GPS) verwendet.

Der Zeitstempel selbst bezieht sich bei Fahrzeugen auf die Referenz-Position, zu der eine CAM ausgesendet wurde. Er ist in [42] als Ganzzahliger Wert (Integer) in Millisekunden definiert. Der Startzeitpunkt wurde auf den *2004-01-01 T 00:00:00Z* festgelegt.

Bei einer RSU ist der Wert des Zeitstempels, der Zeitpunkt zu dem die CAM generiert wurde. Zu beachten ist, dass die Differenz zwischen dem Zeitpunkt, zu dem die CAM generiert wurde und dem Zeitpunkt des Zeitstempels, weniger als *32.767 ms* betragen sollte.

Des Weiteren sind für den Erstellungsprozess einer CAM zeitliche Rahmenbedingungen [39] [41] gegeben, die eingehalten werden müssen. Das Intervall zum Erzeugen einer Nachricht sollte dabei nicht unter 100ms (entspricht einer Rate von 10 Hz) und nicht über 1000ms (entspricht einer Rate von 1 Hz) liegen. Innerhalb dieser Begrenzungen ist die Erzeugung einer CAM abhängig von der Dynamik der ITS-S und dem Auslastungszustand des Übertragungskanals. Auch ist zu beachten, dass ein zeitlicher Abstand zwischen der Generierung von aufeinander folgenden CAMs liegt. Damit wird der Überflutung

des Übertragungskanals vorgebeugt und es wird die Anforderungen des Decentralized Congestion Control (DCC) [43] Rechnung getragen.

Die soeben beschriebenen zeitlichen Anforderungen beziehen sich größtenteils auf eine Fahrzeug ITS-S. Bei RSU ITS-Ss verhält es sich etwas anders. Das Zeitintervall zwischen zwei aufeinander folgenden Nachrichten ist so gewählt, dass mindestens eine Nachricht versendet wird, solange sich ein Fahrzeug in Kommunikationsreichweite befindet. Das Zeitintervall sollte dabei größer oder gleich 1000ms sein, was einer maximalen CAM Generations-Rate von 1 Hz entspricht.

Nach dem Erstellen einer Nachricht wird sie an die „Netzwerk- und Transport-Schicht“ weitergeleitet. Die Verbreitung der CAM ist dann abhängig vom verwendeten Kommunikationssystem. In einem ITS-G5A Netzwerk [44] wird die Nachricht, ausgehend von der Sendestation, an alle ITS-Stationen innerhalb der eigenen Kommunikations-Reichweite übertragen. Die Reichweite kann je nach Sendeleistung variieren.

Für den Versand einer Nachricht wird eine „Point-to-Multipoint“ Kommunikation verwendet, wie sie in [32] spezifiziert ist. Unter Anwendung des ITS-G5<sup>1</sup> wird der Kontroll-Kanal (G5-CCH) [44] verwendet, um eine CAM zu verbreiten. Die Nachricht selbst wird mittels eines „single-hops“ von der Sendestation an Empfangstationen in der Reichweite versendet.

Generell sollte der Prozess vom Erstellen bis zum Versand einer CAM nicht länger als 50ms betragen. Dabei bezieht sich die Zeitspanne (von 50ms), auf die zeitliche Differenz, die von der Auslösung der CAM-Generierung bis zu dem Zeitpunkt, zu dem die CAM an die „Networking & Transport“-Schicht ausgeliefert wurde, vergeht.

Der Empfang einer CAM hat zur Folge, dass der „CA Basic Service“ den Inhalt der Nachricht extrahiert und abhängig vom Zeitstempel die Daten in einer Datenbank (der LDM, auf die später noch eingegangen wird) hinterlegt werden. Zu berücksichtigen ist, dass eine eintreffende CAM nicht an andere ITS-Stationen weitergeleitet wird.

---

<sup>1</sup>ITS-G5 bezeichnet die europäische Version der Nahbereichskommunikationstechnologie (Dedicated Short Range Communication (DSRC)), welche der US-amerikanischen Verkehrsbehörde verwendet wird.

## 5.2 Decentralized Environment Notification Messeage

Eine DENM enthält Informationen, die relevant sind für ein Ereignis, das die Verkehrssicherheit oder den Verkehrszustand beeinträchtigt. Vordergründig durch das von Sensoren entdeckte Gefahrenpotenzial, werden sie dazu verwendet, Verkehrsteilnehmer vor bestimmten Situationen zu warnen. Charakterisiert wird ein Ereignis grundlegend durch seine Art (was für eine Gefahr ist aufgetreten), eine Position (wo besteht die Gefahr), einer Auftrittszeit (seit wann besteht die Situation) und einer abgeschätzten Zeitspanne, in der das Ereignis anhält.

In [39] [41] werden eine Reihe DENM betreffende Anwendungen und Ereignisse definiert. Beispielsweise sendet eine ITS-S eine DENM aus, um zu signalisieren, dass ein Bremsvorgang eingeleitet wurde. Dieselbe ITS-S ist dann für das Aussenden und Beenden der Transmission zuständig. Es kann aber auch vorkommen, dass mehrere ITS-Ss eine Nachricht aussenden, um beispielsweise an einer Stelle eine Warnung für Glatteis zu signalisieren. In diesen Fällen bleibt die Meldung bestehen, auch wenn sich das Fahrzeug nicht mehr an der Position befindet.

Es gibt vier verschiedene DENM Arten, die wie folgt definiert sind

- **New DENM**

Diese DENM-Variante wird erstellt, wenn zum ersten Mal ein Ereignis auftritt. In der Nachricht befindet sich eine Identifikationsnummer und Attribute (wie einer Position, der Art, Ereignis Auftritts Zeit), die das aufgetretene Ereignis beschreiben.

- **Update DENM**

Mit dieser Variante werden die Informationen zu einem Ereignis einer bereits bestehende DENM aktualisiert. Die aktualisierte Nachricht wird von der selben ITS-S versendet, die auch schon die ursprüngliche Nachricht versandt hat.

- **Cancellation DENM**

Diese Nachricht informiert darüber, dass das ursprüngliche Ereignis, das von der selben Station ausgesandt wurde, beendet ist.

- **Negation DENM**

Diese Nachricht informiert über die Beendigung eines Ereignisses. Sie wird in der Regel von einer anderen ITS-S versendet, als derjenige, welche die ursprüngliche Nachricht versandt hat, um ein Ereignis für beendet zu erklären (beispielsweise bei Glatteis).

Der „DEN Basic Service“ (auf dem im Anschluss näher eingegangen wird), sollte in der Lage sein, die oben genannten vier Arten von Nachrichten zu erstellen. Eine Anwendung stellt hierfür eine Anfrage an den „DEN Basic Service“, der daraufhin das Erstellen der DENM veranlasst. Die Art einer DENM, hängt somit auch vom Typ der Aufforderung der Anwendung ab.

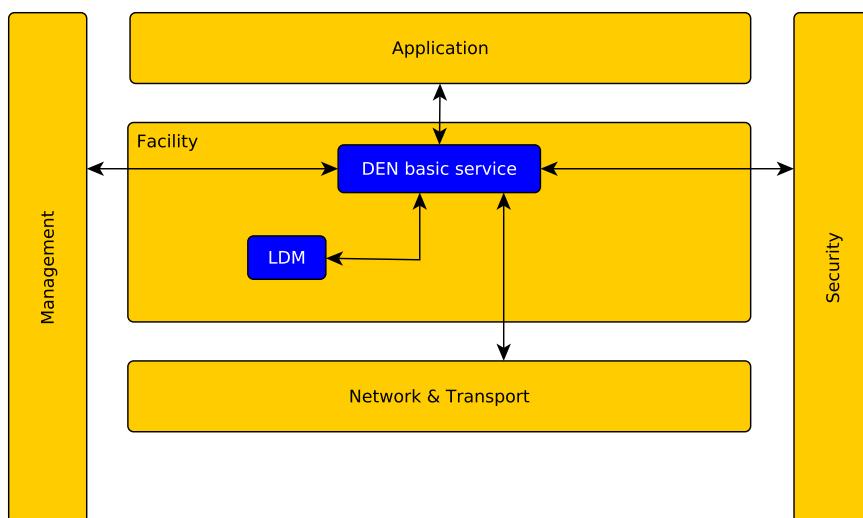
### 5.2.1 DEN Basic Service

Der „Decentralized Environmental Notification (DEN) Basic Service“ dient zur Unterstützung von Anwendungen und befindet sich im „Facilities Layer“. Er erstellt, verwaltet und verarbeitet die DENM. Hierzu sind verschiedene Schnittstellen vorgesehen, die von anderen Entitäten der unterschiedlichen Schichten genutzt werden können.

Zusätzlich interagiert er mit der LDM [45], die als Datenbank für die einzelnen Informationen fungiert.

Bei eintreffenden DENMs werden die Informationen innerhalb der LDM aktualisiert und stehen so der Anwendungsschicht für eine weitere Verarbeitung zur Verfügung.

Abbildung 5.3: Schnittstellen des DEN basic service



Die Abbildung 5.3 zeigt den Zusammenhang des „DEN Basic Service“ innerhalb der ITS-Architektur und dessen Schnittstellen wie sie in [29] definiert sind.

Grundlegend bietet der „DEN Basic Service“ die Funktionalität eine DENM in einem bestimmten Format [38] zu erstellen und ebenso empfangene Nachrichten zu dekodieren. Eine weitere Aufgabe stellt die Verwaltung für die Übertragung einer Nachricht dar. Dies beinhaltet das Erstellen einer Neuen, die Aktualisierung und Beendigung einer bereits existierender Nachrichten, als auch das in einem festgelegten Intervall wiederholende Versenden einer Nachricht.

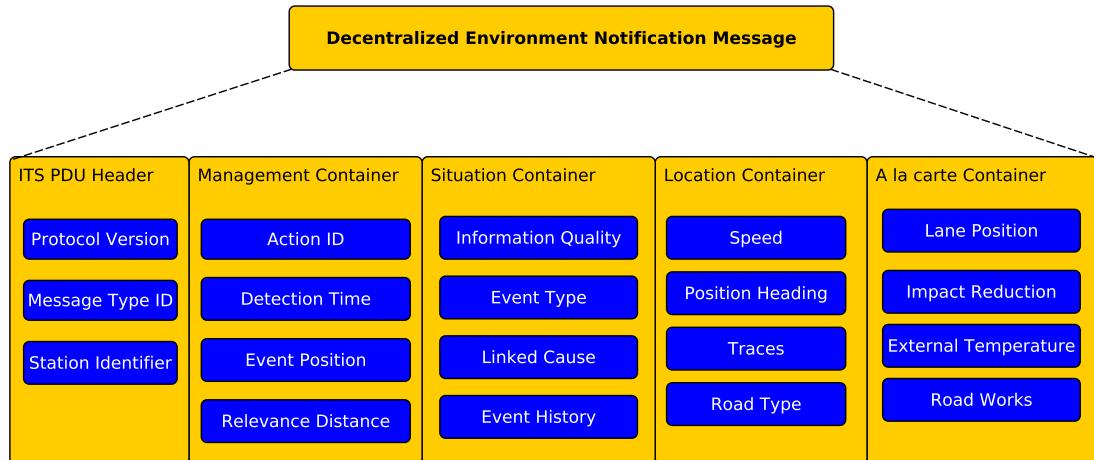
Beim Empfang einer Nachricht sorgt der „DEN Basic Service“ für die Aktualisierung der Datenbank (LDM), verwirft ungültige Nachrichten und versorgt die Anwendungs-Schicht mit den neuen Informationen der DENM.

Als optionale Besonderheit existiert das Keep Alive Forwarding (KAF), das das Weiterleiten von Nachrichten ermöglicht. Dabei handelt es sich um ein Protokoll, mit dem DENMs gespeichert und an andere ITS-S weitergeleitet werden, solange es sich um eine gültige Nachricht handelt.

### 5.2.2 Aufbau einer DENM

Eine DENM beinhaltet wie eine CAM einen „ITS Protocol Data Unit (PDU) Header“ und mehrere Container, welche die Nutzlast einer DENM darstellen.

Abbildung 5.4: Struktur einer DENM



Die generelle Struktur einer DENM ist in der Abbildung 5.4 dargestellt.

Der Header enthält Informationen zur Protokoll Version, dem Nachrichtentyp ( in diesem Fall DENM ) und die ID der ITS-S, welche die Nachricht versendet hat.

Des Weiteren besteht eine Nachricht aus vier festen Informationscontainern. Dem „Management Container“, welcher relevante Informationen für das DENM Management bereithält.

Der „Situation Container“ enthält Informationen des aufgetretenen Ereignisses. Wenn der Situation Container verwendet wird, sollte auch der Location Container mit in eine DENM verpackt werden.

Der „Location Container“ enthält Informationen über den Ort des aufgetretenen Ereignisses. Man unterscheidet hierbei zwischen zwei Kategorien von Arealen. Der **Relevanzbereich** ist die Position, auf den sich eine Begebenheit bezieht. Dieses Segment wird durch die Anwendung bestimmt, welche die DENM auslöst. Zur Abgrenzung der Region sind in [46] die geometrischen Formen kreisförmig, elliptisch und rechteckig definiert. Je nach Art und Position variiert die Form des Relevanzbereiches. Neben dem Relevanzbereich kann ein **Ziel (Destination) - Bereich** festgelegt werden. Das definiert ein Gebiet, das den Bestimmungsort einer DENM entspricht. Dieser Bereich lässt sich ebenfalls in den drei geometrischen Formen ausdrücken.

Die **Strecke (Traces)** zwischen der aktuellen Position bis zum Relevanzbereich, werden als Wegpunkte hinterlegt. Die Länge oder die Dichte der Wegpunkte in einem Trace variiert mit den Anforderungen. Als Untergrenze sollte eine DENM mindestens einen Trace enthalten.

Es besteht aber auch die Möglichkeit mehrere Traces in einer DENM unterzubringen, wenn es mehr als einen Weg zu einem erkannten Ereignis gibt (beispielsweise durch eine Kreuzung).

Zu guter Letzt, befinden sich im „A la carte Container“ die Informationen zu einem bestimmten Anwendungsfall und weitere Details, die nicht in die übrigen Container untergebracht werden konnten.

Für alle DENMs wird der „Header“ und der „Management Container“ vorausgesetzt, die Übrigen gelten als optional. Um eine DENM zu widerrufen, sind die drei letztgenannten Container unwichtig und müssen nicht in der DENM eingebettet werden.

### 5.2.3 Versand und Empfang einer DENM

Der Vorgang des DEN - Nachrichtenaustauschs zwischen den einzelnen ITS-S wird durch das DENM-Protokoll beschrieben.

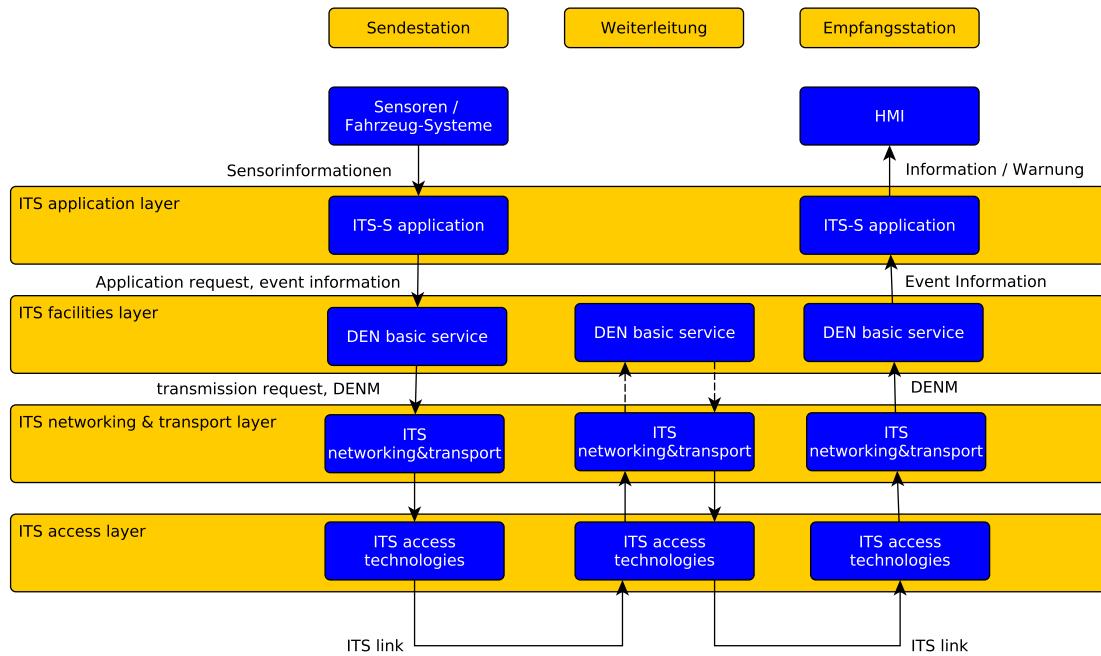


Abbildung 5.5: Datenfluss einer DENM

Die Abbildung 5.5 zeigt den prinzipiellen Ablauf der Verbreitung einer DENM.

Fordert eine Anwendung die Erzeugung einer neuen DENM beim „DEN Basic Service“ an, wird eine neue Identifikationsnummer (Action ID) dem Ereignis zugewiesen. Diese Identifikationsnummer ist eine Kombination aus der ITS-S ID und einer fortlaufenden Nummer. Unter Umständen kommt es vor, dass mehrere ITS-S das gleiche Ereignis zur selben Zeit wahrnehmen. Da es sich um unterschiedliche Stationen handelt, resultieren hieraus unterschiedliche Identifikations-Nummern für das selbe Ereignis. Der „DEN Basic Service“ ist dafür zuständig, dass jedes Mal eine neue Action ID erzeugt wird.

Aus Sicherheitsgründen ändert sich die Stations ID von Zeit zu Zeit. Alle bisher erstellten und gespeicherten Nachrichten mit einer älteren Stations ID werden daraufhin aktualisiert.

Erkennt die ITS-S, dass sich ein Ereignis ändert, nachdem eine DENM für dieses Ereignis erzeugt wurde, wird anstelle einer weiteren DENM lediglich die Informationen der bestehenden aktualisiert. Als weitere Maßnahme wird die „Referenz Zeit“ einer DENM, die ursprünglich die Zeit repräsentiert, zu der sie erstellt wurde, ebenfalls aktualisiert. Zwischen zwei aufeinander folgenden Aktualisierungsschritten der DENM, ist laut Spezifikation das wiederholte Versenden einer DENM, innerhalb eines vordefinierten Wiederholungsintervalls vorgesehen.

Ist ein Ereignis beendet, wird dieses durch den Versand einer entsprechenden DENM signalisiert. Dabei gibt es zwei Möglichkeiten, wie das Ende eines Ereignis kommuniziert werden kann. Zum Einen durch eine „Cancellation“ DENM, die von der ursprünglichen ITS-S stammt, zum Anderen durch eine „Negation“ DENM, die von jeder anderen ITS-S vermittelt werden kann.

Eine DENM ist dafür vorgesehen, an so viele ITS-Ss wie möglich geschickt zu werden, die sich im Einzugsbereich des Ereignisses befinden. Das bedeutet, dass die Nachricht auch an ITS-S geschickt werden müssen, die sich außerhalb des Funkbereichs einer ITS-S befinden. Dies wird mittels Multi-Hopping erreicht. Die zentrale Rolle der zentralen Weiterleitung einer Nachricht übernimmt die „Networking & Transport“-Schicht.

In der „Facilities“-Schicht kann der „DEN Basic Service“ auf eine optionale Funktion mit dem Namen KAF zurückgreifen. Hierbei speichert der „DEN Basic Service“ eine empfangene DENM, und leitet sie bei Bedarf an andere ITS-S weiter. Im Gedächtnis ist dabei der Umstand zu behalten, dass die Nachricht zum Zeitpunkt der Speicherung gültig und der Empfänger sich innerhalb des Ziel- oder Relevanzbereiches befinden muss.

Die zentrale Weiterleitung als auch die KAF ergänzen sich beide in ihrer Funktionalität.

Für den Empfang einer Nachricht wird mittels des Relevanzbereiches der Nachricht und der Position der empfangenden ITS-S herausgefiltert, ob die Informationen von Belang sind.

Beim Empfang einer Nachricht, wird ebenfalls die Reiseroute mit den in der DENM abgebildeten „Traces“ verglichen, um zu entscheiden, ob die Informationen für die empfangende ITS-S von Interesse sind, und, ob ein Ereignis auf der gewählten Route aufgetreten ist.

### 5.3 Local Dynamic Map

Ein kooperatives System für kritische Verkehrssicherheits-Anwendungen profitiert von der Benutzung von digitalen Karten. Solche Karten innerhalb eines ITS beinhalten fahrspurbezogene Daten wie Bordsteine, laufende Fußgänger, Fahrradwege, Ampeln oder Verkehrsschilder.

Weiterhin werden alle dynamischen Objekte, welche direkt durch die ITS mittels Sensoren oder indirekt durch andere ITS (mit Hilfe von CAMs) in eine Karte, der LDM, eingetragen werden. Mit den Informationen in der LDM können Beziehungen zwischen den einzelnen Einträgen hergestellt und Abhängigkeiten ermittelt werden. Zu diesem Zweck ist es erforderlich, dass die Einträge mit einem Zeitstempel versehen sind.

Die LDM ist ein wesentlicher Bestandteil eines ITS. Sie stellt Informationen, beeinflusst durch die Straßenverkehrssituation, bereit. Dabei kann es sich beispielsweise um die Information von bewegten (andere Verkehrsteilnehmer) oder starren (Ampeln, Verkehrszeichen) Objekten handeln, die sich in der Nähe der ITS-S befinden.

Somit fungiert eine LDM als eine Art Datenbank, in der sich Informationen aus verschiedenen Quellen befinden. Ihre Spezifikationen sind in [45] hinterlegt. Ein wesentlicher Aspekt stellt die Authentifizierung bei der Abfrage der Daten dar. Somit können Daten zu umgebenden Fahrzeugen oder RSUs nur von autorisierten Anwendungen abgefragt werden.

Die eigentlichen Informationen werden in Form von Objekten, den sogenannten „LDM Data Objects“, gespeichert oder gelesen. Die Objekte können aus Teil-Objekten zusammengesetzt sein, ähnlich wie bei einer hierarchischen Struktur. Sie enthalten Attribute, welche die Datenelemente, wie sie in [42] definiert sind, repräsentieren. Daten, die in der LDM gespeichert werden, werden nicht durch die LDM modifiziert.

Einrichtungen wie der „DEN- oder CA- Basic Service“ können ihre Nachrichten (Informationen) in der LDM speichern. Anwendungen können Informationen in die LDM sowohl speichern als auch aus ihr lesen. Hierzu ist es notwendig, dass sich die jeweilige Anwendung bei der LDM registriert, um darüber informiert zu werden, wenn ein neuer Datensatz vorhanden ist.



## 6. Zeitpräzision und Synchronisierung

“ Die Zeit ist aus den Fugen. ”

William Shakespeare, *Hamlet*, 1601-1602

Die Zeit in voneinander unabhängigen Systemen kann in erster Linie vernachlässigt werden. Interagieren diese Systeme jedoch miteinander, ist eine präzise Uhrzeit von besonderer Bedeutung. Mit ihr können koordinierte Abläufe bestimmt werden.

Diese Kapitel führt in die Technik von verschiedenen Methoden zur Zeiterhebung und Synchronisierung ein. Dazu wird zunächst auf das Verfahren mittels GPS eingegangen, das in ITS Systemen bevorzugt wird. Für modelbasierte Fahrzeuge, die sich innerhalb von Gebäuden bewegen, ist diese Technik durch dessen physikalische Begrenzungen ungenügend. Hierzu wird auf Alternativen wie das NTP oder bekannte Verfahren aus dem Bereich der Verteilten Systeme eingegangen.

## 6.1 Global Positioning System

Die Teilnehmer eines ITSs verwenden vornehmlich GPS-Empfänger, um einerseits eine geographische Verortung vorzunehmen und andererseits ein global verfügbares Zeit-Signal nutzen zu können, mit dem sie synchron zu anderen Teilnehmern sind. Für den Nachrichtenaustausch wird nach BSA die auch von den GPS-Satelliten favorisierte Temps Atomique International (TAI) [62] als Referenzzeit verwendet.

Grundlegend handelt es sich beim GPS um ein Navigationssystem, gestützt durch eine Vielzahl von Satelliten. Es wurde vom amerikanischen Verteidigungsministerium entwickelt und wird derzeit vom Militär als auch privaten Anwendern genutzt.

Die Position (geographische Länge, Breite und Höhe ) wird mit Hilfe von drei Satelliten bestimmt. Hierzu werden die Signallaufzeiten zwischen den Satelliten und dem Empfänger berechnet und so die Entfernung ermittelt. Das dahinter stehende Verfahren nennt sich Triangulation. Die Genauigkeit wird von [4] mit einem Bereich zwischen 20m und bis zu 1mm angegeben. Ein vierter Satellit kümmert sich um die Synchronisierung der Uhren des Empfängers und der Satelliten. Die Zeitangabe entspricht der Weltzeit (Universal Time Coordinate (UTC)) und unterliegt einer Genauigkeit im Bereich von 60ns bis 5ns.

Neben dem amerikanischen GPS existieren ähnliche Ansätze für Navigationssysteme in weiteren Ländern. Die Europäische Union entwickelt ein unter ziviler Kontrolle stehendes System namens „Galileo“. In Russland wird ebenfalls vom Verteidigungsministerium ein eigenständiges Satellitennavigationssystem mit dem Namen „GLONASS“ eingesetzt. Mit „Beidou“ unterhält die chinesischen Regierung eine Varität, um die Abhängigkeit vom amerikanischen System zu verringern. Die grundsätzlichen Verfahren bleiben bei allen Alternativen dennoch gleich.

Auf der Anwenderseite übermitteln GPS-Geräte die empfangenen GPS-Daten gemäß des National Marine Electronics Association (NMEA)-0183 [63] Protokolls. Hinter dem Protokoll verbirgt sich die gleichnamige Organisation NMEA [64]. Üblicherweise werden die Daten dabei im American Standard Code for Information Interchange (ASCII)-Format über eine serielle Verbindung mit einer Geschwindigkeit von 4800 Baud übertragen. In einem Datensatz werden die Information wie Geschwindigkeit, Position oder Zeit übermittelt.

## 6.2 Synchronisation in der Welt der Verteilten Systeme

Was für Fahrzeuge in einem menschengerechten Maßstab mit GPS funktioniert, ist nur bedingt oder nicht anwendbar auf Modelfahrzeuge, die sich größtenteils in geschlossenen Gebäuden befinden. Hier müssen andere Wege gefunden werden, um die Uhren der einzelnen Teilnehmer eines Netzwerkes zu synchronisieren. Die Ursachen voneinander abweichender Uhren, liegt in der Regel in unterschiedlich schnell laufenden Uhren begründet. Unter anderem liegt dies an unterschiedlichen Taktungen, bedingt durch die verwendeten

Quarze als Taktgeneratoren. Eine weitere Facette, die berücksichtigt werden muss, sind unterschiedliche Signallaufzeiten, die in einem Netzwerk entstehen können.

Ein Modellfahrzeug verhält sich in einem Netzwerk nicht anders als voneinander unabhängige Prozesse, die in einem verteilten System agieren. Die Prozesse tauschen dabei Informationen über Nachrichten aus, die in einem zeitlichen Kontext gesehen werden müssen, um so Beziehungen zwischen auftretenden Ereignissen herleiten zu können. In [9] wird auf die gebräuchlichsten Verfahren in Bezug auf verteilte Systeme eingegangen. Der Artikel [10] beschäftigt sich mit der Analyse von Schwachstellen und der Bereitstellung von Sicherheitskonzepten für verschiedene Synchronisierungsprotokolle.

### 6.2.1 Happened-Before

Es liegt in der Natur des Menschen, Ereignisse in eine gewisse zeitliche Ordnung zu bringen, um so Abläufe beschreiben zu können.

Der Mathematiker und Informatiker Leslie Lamport postulierte in [11] das „Happened Before“-Modell zur Ordnung von kausal zusammenhängenden Ereignissen innerhalb eines asynchron verteilten Systems.

**Definition 6.2.1** Wenn ein Ereignis  $a$  vor einem anderen Ereignis  $b$  auftritt, dann sollte das Ereignis  $a$  zeitlich vor dem Ereignis  $b$  stattgefunden haben (happened before).

$$\text{Für jedes Ereignis } a, b: \quad \text{Wenn } a \rightarrow b, \quad \text{dann } C\langle a \rangle < C\langle b \rangle \quad (6.1)$$

Handelt es sich bei den Ereignissen  $a, b$  um das Senden ( $a$ ) und Empfangen ( $b$ ) einer Nachricht, so muss nach der Beziehung  $a \rightarrow b$  das Senden vor dem Empfangen eingetreten sein. Die Relation zwischen den einzelnen Ereignissen ist transitiv. Das bedeutet, wenn ein weiteres Ereignis ( $c$ ) hinzukommt, das nach dem Ereignis ( $b$ ) stattfindet, lässt sich schlussfolgern, dass  $a \rightarrow c$ .

$$a \rightarrow b; \quad b \rightarrow c \quad \Rightarrow \quad a \rightarrow c \quad (6.2)$$

Das Modell liefert dabei nur eine partielle Ordnung und ist nicht ausreichend bei der Verwendung von Nebenläufigkeiten. Dies lässt sich zeigen durch  $a \rightarrow b$ , also das Ereignis  $a$  trat zeitlich gesehen vor  $b$  auf, somit sollte  $\text{Uhr}(a) < \text{Uhr}(b)$  sein. Der daraus resultierende Umkehrschluss, unter der zeitlichen Betrachtung zweier Ereignisse  $\text{Uhr}(c) < \text{Uhr}(d)$ , ist nicht gleichbedeutend mit  $c \rightarrow d$  (also das C vor D stattgefunden hat).

### 6.2.2 Lamport-Uhr

In der Lamport-Uhr findet die „Happened-Before“-Relation ihre Anwendung. Dabei abstrahiert *Lamport* die reale Hardware-Uhr eines Systems auf einen einfachen Zähler, der beim Auftreten eines Ereignisses hochgezählt wird. Somit zählt sie zu den logischen Uhren, die im Gegensatz zu einer Hardware-Uhr nicht die physikalische Zeit misst, sondern nur

monoton steigende Werte (Zeitstempel) verwendet. Mit dem Zeitstempel lassen sich zeitliche Geschehnisse in eine partielle Ordnung bringen, die einen kausalen Zusammenhang besitzen.

Der Ablauf der Lamport-Uhr lässt sich wie folgt beschreiben.

- Jeder Prozess  $P_i$  in einem verteilten System besitzt eine Uhr  $C_i$  deren Wert inkrementiert wird, sobald ein Sende- oder Empfangsereignis  $e$  auftritt.
- Sendet ein Prozess eine Nachricht, schickt er den aktuellen Wert seiner Uhr als Zeitstempel mit. Daraufhin erhöht er den Wert seiner Uhr um eins.

$$C_i = C_i + 1$$

- Empfängt ein Prozess eine Nachricht, wird zunächst dessen Uhr erhöht. Sollte die Uhrzeit kleiner sein als der Zeitstempel in der empfangenden Nachricht, so wird die Uhrzeit mit dem Wert des Zeitstempels um eins addiert angepasst.

$$C_i = \max(C_i, \text{Zeitstempel}) + 1$$

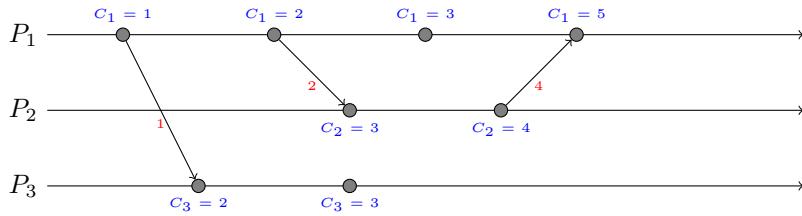


Abbildung 6.1: Ablauf der Lamport-Uhr

In der Abbildung 6.1 ist ein exemplarisches Verlauf skizziert, der auch eine plausible kausale Sortierung zwischen den beteiligten Ereignissen aufzeigt.

Der Ansatz der Lamport-Uhr bietet nicht die Möglichkeit einer globalen (totalen) Sortierung von Ereignissen. Dies spiegelt sich auch in der Tatsache wieder, dass sie keine starke Konsistenzbindung erfüllt. Hierdurch ist es ausgeschlossen, eine Aussage über kausal unabhängige (nebenläufige) Ereignisse zu treffen.

### 6.2.3 Vektor-Uhr

Das Konzept der Vektor-Uhr stellt die Erweiterung zur Lamport-Uhr dar, die in der Lage ist, auch Nebenläufigkeiten von Ereignissen zu registrieren. Beschrieben wird das Verfahren von *Collin J. Fridge* in [47] und *Friedemann Mattern* in [48].

Prinzipiell verfügt jeder Prozess über einen Zähler, der im Gegensatz zur Lamport-Uhr nicht nur aus einer Zahl besteht, sondern vielmehr aus einem Vektor der Länge  $m$ , wobei  $m$  die Anzahl der Prozesse im gesamten System darstellt. Die Indizierung des Vektors

erfolgt in der Regel über die ID des Prozesses, einer IP-Adresse oder einer Kombination aus beiden.

Im Folgenden sei der Ablauf zur Aktualisierung der Vektor-Uhr dargestellt.

- Beim Start des System erzeugt jeder Prozess  $P_i$  einen Vektor der Länge  $m$  und befüllt diesen mit dem Startwert Null für jedes Element.
- Jeder Prozess  $P_i$  verwaltet seinen Eintrag in der Vektoruhr  $C[i]$ , deren Wert inkrementiert wird, sobald ein Sende- oder Empfangsereignis  $e$  auftritt.
- Sendet ein Prozess eine Nachricht, wird der Eintrag  $C[i]$  des Prozesses  $P_i$  um eins erhöht und der komplette Vektor als Zeitstempel an den Empfänger geschickt.

$$C[i] = C[i] + 1$$

$$\text{Zeitstempel} = C$$

- Empfängt ein Prozess  $P_i$  eine Nachricht, wird zunächst dessen lokal gehaltene Vektor-Uhr  $C[i]$  erhöht. Im nächsten Schritt wird mit Hilfe des empfangenen Zeitstempels das Maximum der beiden Vektoren gebildet.

$$C[i] = C[i] + 1$$

$$\forall_j : C[j] = \max(C[i], \text{Zeitstempel}[j])$$

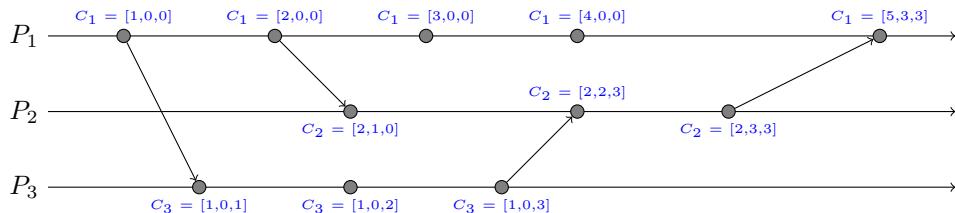


Abbildung 6.2: Ablauf der Vektor-Uhr

Mit den Vektor-Uhren lässt sich wie bei der Lamport-Uhr die Beziehung von mindestens zwei Ereignissen bestimmen. Es gilt  $a \rightarrow b$  und somit  $\text{Uhr}(a) < \text{Uhr}(b)$ . Ergänzend lässt sich nun auch der umgekehrte Fall beschreiben. Unter dem Bewußtsein, dass  $\text{Uhr}(c) < \text{Uhr}(d)$ , lässt sich daraus entnehmen, dass  $c \leftarrow d$ . Diese Verbindung konnte mit der Lamport-Uhr nicht hergestellt werden.

## 6.3 Network Time Protocol

Mit dem Network Time Protocol (NTP)[49] erschuf David Mills [5] ein Konzept zur Synchronisierung von Uhren. Bestehend aus einem Software Programm, einem Protokoll zum Verteilen eines Zeitsignals zwischen einem Server und einem Client, sowie einer Ansammlung von Algorithmen zur Anpassung der Systemzeit, dienen diese drei Komponenten der zeitlichen Abstimmung von Rechnern in einem Netzwerk.

Das Zeitsignal mit dem die lokale Uhr synchronisiert werden soll, stammt in der Regel von einer Atomuhr, einer Funkuhr oder einem NTP-Server Verbund. Der Aufbau von NTP

ist hierbei hierarchisch organisiert und wird in verschiedene Schichten (Straten) unterteilt. Das oberste Glied (auch Stratum 0 genannt) stellt einen präzisen Zeitgeber (beispielsweise eine Atomuhr) dar. In den darunter liegenden Schichten befinden sich NTP Server- und Client-Systeme, die ihre Uhren mit den Zeitsignal eines über ihnen liegenden Systems angleichen. Ebenso tauschen die einzelnen Systeme eines Stratums untereinander die Daten aus. Es wird vorausgesetzt, dass es sich dabei um zuverlässige und erreichbare Systeme handelt, auch wenn es zu Stabilitätsproblemen innerhalb des Netzwerkes kommt. Die Anpassung der Zeit wird auch bei einer eingangs großen Abweichung nur in kleinen Intervallen geändert.

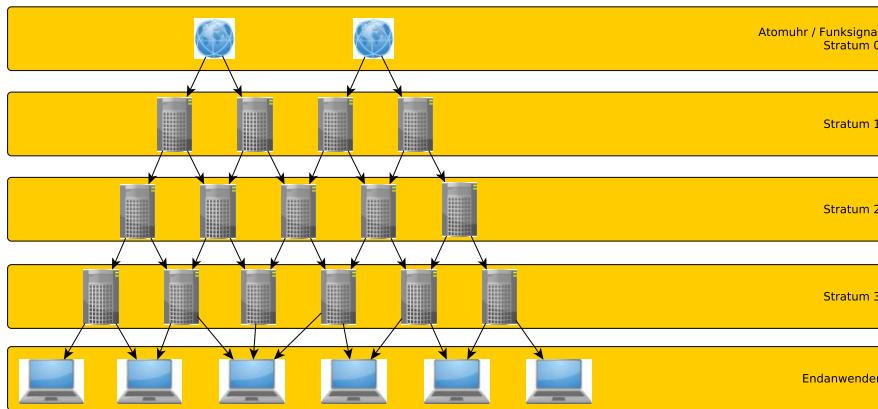


Abbildung 6.3: NTP Hierarchie

Die dargestellte Zeit in einem Computersystem wird im UTC Format angegeben. Für den Austausch des Zeitsignals verwendet NTP zwei unterschiedlich lange Formate (64bit und 128bit), die als Zeitstempel vom Server zu einem Client versendet werden.

Bei beiden Formaten stellen die höherwertigen Bits die Zeit seit dem 1. Januar 1900 in Sekunden dar. Die niederwertigen Bits geben den Anteil der Sekunden an. Mit dem 64bit Format lässt sich die Zeit bis ins Jahr 2036 darstellen, wohingegen mit dem 128bit Format die Lebenszeit des Universums abgedeckt werden könnte.

In Europa verweist [65] auf zwei Zeitgeber:

Zum einen der DCF77-Sender [12], [66] der von der Physikalisch-Technische Bundesanstalt (PTB) [67] betrieben wird. Die Zeitreferenz kommt von einer Caesium-Atomuhr, die sich in Braunschweig befindet.

Zum anderen das TZIZ Distribution Francaise (TDF), dessen Zeitsignal ebenfalls von einer Caesium-Atomuhr stammt und von einem Sender in Allouis, mittels des Sendemasts France-Inter, über das auch ein Radioprogramm, vom Betreiber Radio France ausgesandt wird.

### 6.3.1 Simple Network Time Protocol

Als eine Vereinfachung zum NTP wurde ebenfalls von David Mills das Simple Network Time Protocol (SNTP) [50] entwickelt. Mit ihm ist es möglich die Synchronisierung mit nur einem Zeitserver vorzunehmen, wohingegen beim NTP mehrere Server involviert sind. Das Protokoll von SNTP ist mit dem vom NTP nahezu identisch, sodass auch NTP Server als Zeitserver in Frage kommen. SNTP verwendet einfachere, sowie Ressourcen schonende Algorithmen, kann aber nicht den selben Grad der Genauigkeit wie NTP liefern.

Beide Varianten wurden durch den aktuellen Standard im RFC 5905 [51] aufbereitet. Hierdurch existiert im Grunde nur noch der NTPv4 Standard, in dem SNTPv4 eine Untergruppe darstellt.

## 6.4 Precision Time Protocol

Das Precision Time Protocol (PTP) dient wie NTP und SNTP der Synchronisierung von Uhren innerhalb eines Computer Netzwerkes.

Das PTP wurde erstmal im *Standard IEEE 1588-2002* veröffentlicht. Aktuell ist derzeit der *Standard IEEE 1588-2008* unter dem Namen PTP Version 2, welche aber nicht abwärtskompatibel ist. Dafür bringt sie eine verbesserte Genauigkeit und Robustheit mit sich.

Die Struktur der einzelnen Komponenten ist hierarchisch gegliedert, mit einem oder mehreren Netzwerk-Segmenten (Communication Media) und einer oder mehrerer Uhren. Eine einfache Uhr ist ein Gerät, mit einer einfachen Netzwerkverbindung und ist entweder die Quelle von (Master) oder das Ziel für (Slave) eine Synchronisations-Referenz.

Als Verbindungsstück zwischen einem Master und mehreren Slaves, existiert die Idee einer Grenz-Uhr (boundary-clock). Sie besitzt zahlreiche Netzwerkverbindungen und kann akkurat die einzelnen Netzwerksegmente synchronisieren.

Für jedes Netzwerk-Segment wird ein Master gewählt. Der oberste Zeitgeber der Hierarchie wird Grandmaster genannt. Der Grandmaster sendet seine Zeitinformationen an die Mitglieder seines Netzwerksegments. Wenn sich Grenz-Uhren in dem Segment befinden, leiten diese präzise die Zeit an ihre jeweiligen Segmente weiter. Ein einfacher Aufbau des PTP findet in einem einfachen Netzwerk statt, das keine Grenz-Uhren beinhaltet. Hier wird nur der Grandmaster gewählt und alle anderen Uhren synchronisieren sich mit ihm. Typischerweise benutzt PTP denselben Zeitraum wie die Unix-Zeit, beginnend vom 1. Januar 1970 um 0 Uhr. Während Unix-Systeme die koordinierte Weltzeit verwenden (UTC), ist PTP auf die internationale Atomzeit (TAI) ausgelegt. Der Grandmaster ist in der Lage den entstehenden Offset zwischen UTC und TAI zu kompensieren.

In [52] wird der Synchronisationsprozess in zwei Schritten zusammengefasst. Anfangs wird die Master-Uhr, unter Verwendung des im *IEEE 1588 Standard* definierten „Best Master“

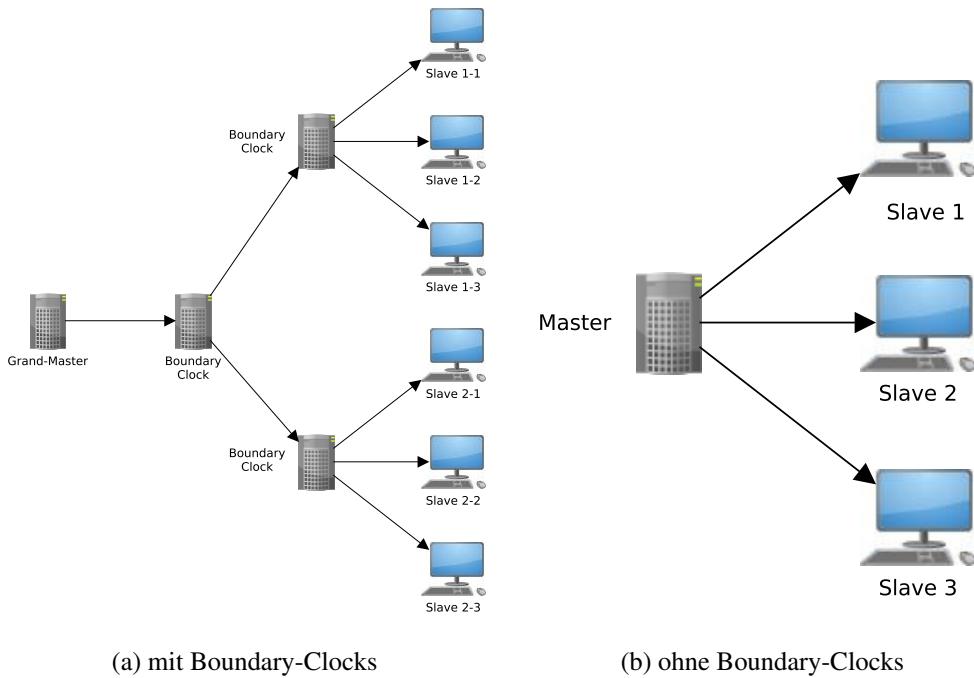


Abbildung 6.4: Darstellung eines PTP Netzwerkes

Clock“-Algorithmus bestimmt. Der zweite Schritt besteht im Messen und korrigieren von Zeitunterschieden, die durch Uhrenabweichungen und Netzwerkverzögerungen entstanden sind. Die Master-Uhr wird diejenige, welche die präziseste Zeit vorhält. Alle anderen werden sogenannte Slaves und synchronisieren ihre Uhren mit dem Master.

Die Synchronisierung erfolgt über den Austausch von Nachrichten. So gibt es „Event (Ereignis)“- und „General (Allgemeine)“-Nachrichten. Prägnant für den Ablauf der Synchronisierung sind die Ereignis-Nachrichten. Insbesondere die Paarung „Sync“ und „Follow Up“ sowie „Delay Request“ und „Delay Response“ auf die später noch eingegangen wird. In der Abbildung 6.5 ist exemplarisch ein vereinfachter Ablauf zur Anpassung einer „Slave“-Uhr dargestellt. Nach [53] und [52] lässt sich die Funktionsweise wie folgt veranschaulichen.

Für die Korrektur der Uhrzeit ist es notwendig den Zeitunterschied zwischen Master und Slave, sowie die Signallaufzeiten der Nachrichten zu berücksichtigen.

Im ersten Schritt wird der **Versatz (Offset)** zwischen den zwei Uhren berichtet. Hierzu sendet der „Master“ eine „Sync“- und „Follow-Up“-Nachricht an den „Slave“. Registriert der „Slave“ eine eingehende „Sync“-Nachricht wird diese mit einem Zeitstempel der lokalen Uhr versehen und mit dem Wert der sich in der später eintreffenden Follow-Up-Nachricht befindet, verglichen. Die Differenz zwischen den beiden Zeitstempeln stellt den Versatz (Offset) der Slave-Uhr zum Master und der benötigten Zeit zur Nachrichtenübertragung dar. Der Slave aktualisiert seine Uhr mit dem Versatz (Offset). Um Schwankungen während der Übertragung zu kompensieren, wird ein weiteres Mal ein „Sync“ und „Follow-Up“



Abbildung 6.5: Ablauf einer Synchronisierung mittels PTP

Nachricht-Paar versendet und die Uhren des „Slave“ bei Bedarf angepasst.

Der zweite Schritt betrachtet die **Verzögerungszeit**, die während des Versands der Nachrichten entsteht.

Der Slave sendet eine „Delay Request“-Nachricht an den Master und sichert sich die Sendezzeit intern ab. Im Gegenzug, sendet der Master eine „Delay Response“-Nachricht, die die Ankunftszeit der „Delay-Request“-Nachricht enthält, an den Slave zurück. Mit der lokalen Sendezzeit und dem Zeitstempel der sich in der „Delay Response“ Nachricht befindet, errechnet der Slave die Verzögerungszeit zum Master.





## 7. Routing-Algorithmen

“ Obwohl das Land von Strassen zerschnitten ist, Wege von Haus zu Haus führen, Zufahrten den letzten Hof erreichen, ist es manchmal schwer, den Weg zueinander zu finden. ”

---

Kurt Haberstich, Schweizer Buchautor und Aphoristiker, \*1948

Das Verteilen von Nachrichten ist ein wichtiger Aspekt bei der Kommunikation zwischen verschiedenen Teilnehmern eines ITS.

Dieser Abschnitt beschreibt, wie Nachrichten, zwecks der Verteilung zu verschiedenen ITS Teilnehmern adressiert werden. Insbesondere wird der Fragestellung nachgegangen, wer eine Nachricht über ein gewisses Ereignis erhalten soll.

Im europäischen Raum setzt auf die GeoNetworking Technologie, wie sie schon im Kapitel 4.5 erwähnt wurde. Sie baut auf dem in [54] beschriebenen Basic Transport Protocol (BTP) auf. Das Hauptziel in diesem Abschnitt ist es, die Ziel-Position herauszufinden. Normalerweise werden solche Informationen in den Nachrichten-Paketen vom Sender mitgeliefert. Der einfachste Fall ist eine feste Position wie sie bei RSUs der Fall ist. Bei ungünstigen Konstellationen werden einfache Flooding Techniken eingesetzt, um die Position von allen, im Netzwerk befindlichen Knoten, zu erhalten.

In der Literatur [6] [7] [8] [13] werden verschiedene Möglichkeiten und Strategien zur Umsetzung von Routing-Algorithmen innerhalb eines drahtlosen Netzwerkes erläutert.

## 7.1 Flooding-basierte Verfahren

Hierbei besitzt ein Knoten eine Tabelle, in der alle erreichbaren Knoten in seiner Umgebung mit ihren Positionsdaten hinterlegt sind. Es sind einfache Verfahren, die allerdings den Nachteil haben, dass sie nicht sehr gut skalieren und eine Menge Datenverkehr verursachen. Ein Vorteil ist jedoch, dass sie in kleinen Netzwerken (beispielsweise auf Landstraßen) recht robust funktionieren.

### 7.1.1 Einfaches Flooding

Beim „Flooding“ (Fluten) handelt es sich um einen sehr einfachen Algorithmus, beschrieben in [14]. Bei dieser Methode werden Nachrichten (Pakete) von einer Quelle zu einem oder mehreren Zielen über ein Netzwerk gesendet.

Die Vorteile liegen darin, dass er immer den kürzesten Weg zu allen Empfängern findet und keine globalen Adressen benötigt. Der Nachteil besteht darin, dass eine Menge Netzwerk-Verkehr verursacht wird, was zu einer hohen Wahrscheinlichkeit von Paketkollisionen führt. Die Problematik ist auch unter den Namen **Broadcast Storm Problem** [55] bekannt. Der Umstand, dass die einzelnen Knoten nach dem Eintreffen einer Nachricht, diese sofort weiterleiten, verursacht eine Menge Datenverkehr. Schlussendlich kann dies dazu beitragen, dass Verbindungen nicht mehr aufgebaut und bestehende unterbrochen werden können.

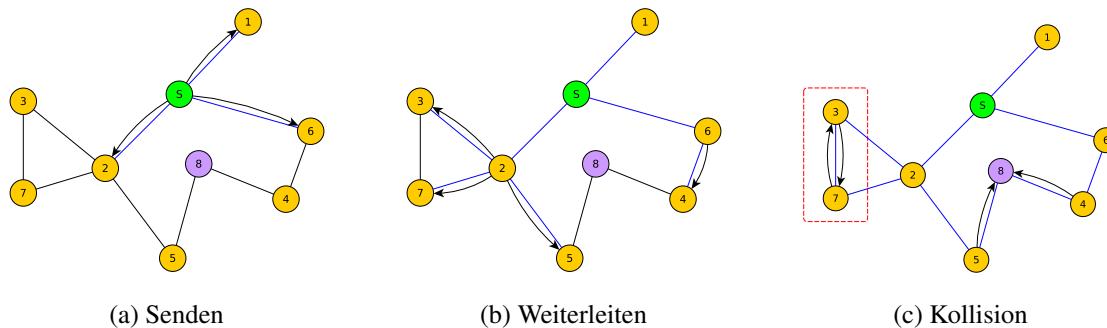


Abbildung 7.1: Ablauf von Flooding

In der Abbildung 7.1 ist der generelle Ablauf des Algorithmus dargestellt.

Die Quelle 8 sendet zunächst die Nachricht an die von ihm in der Nähe befindlichen Knoten (Abbildung 7.1a). Diese leiten wiederum die Nachricht weiter, bis sie beim Zielknoten 3 angelangt ist (Abbildung 7.1b). Bei diesem Prozess kann es dazu kommen, dass zwei Knoten sich dieselbe Nachricht zuschicken, je nachdem, woher sie die Nachricht bekommen haben (Abbildung 7.1c).

In dem Ablauf wird auch deutlich, dass es keine Limitierung in Bezug auf eine bestimmte Position oder Bereich vorgesehen ist. Alternativen wie „Geography Aware Flooding (GAF)“[15]- oder der „Optimised Link State Routing (OLSR)“-Algorithmus [16] [17] versuchen dem entgegenzuwirken.

### 7.1.2 Ad-Hoc On Demand Vector Routing

Beim Ad-Hoc On-Demand Distance Vector (AODV) Routing [56][18] handelt es sich um ein reaktives Protokoll<sup>1</sup> mit einer Routing-Tabelle. Das Ziel, an das Daten gesendet werden soll, wird durch seine Position (Standort) bestimmt. Dazwischen liegende Knoten benötigen nur eine lokale Adresse.

Die Vorteile dieses Verfahrens bestehen darin, dass nur eine lokale Adressierung notwendig ist. Die Routen zu den einzelnen Knoten werden bei Bedarf festgelegt. Angewendet werden kann es auf halb-statischen Netzwerktopologien worunter im Verkehrswesen Autobahnen fallen.

Nachteilig hingegen ist der anfangs langsame Ausbreitungsprozess, woraus sich schließen lässt, dass es in sich schnell ändernden Netzwerktopologien nicht einsetzbar ist.

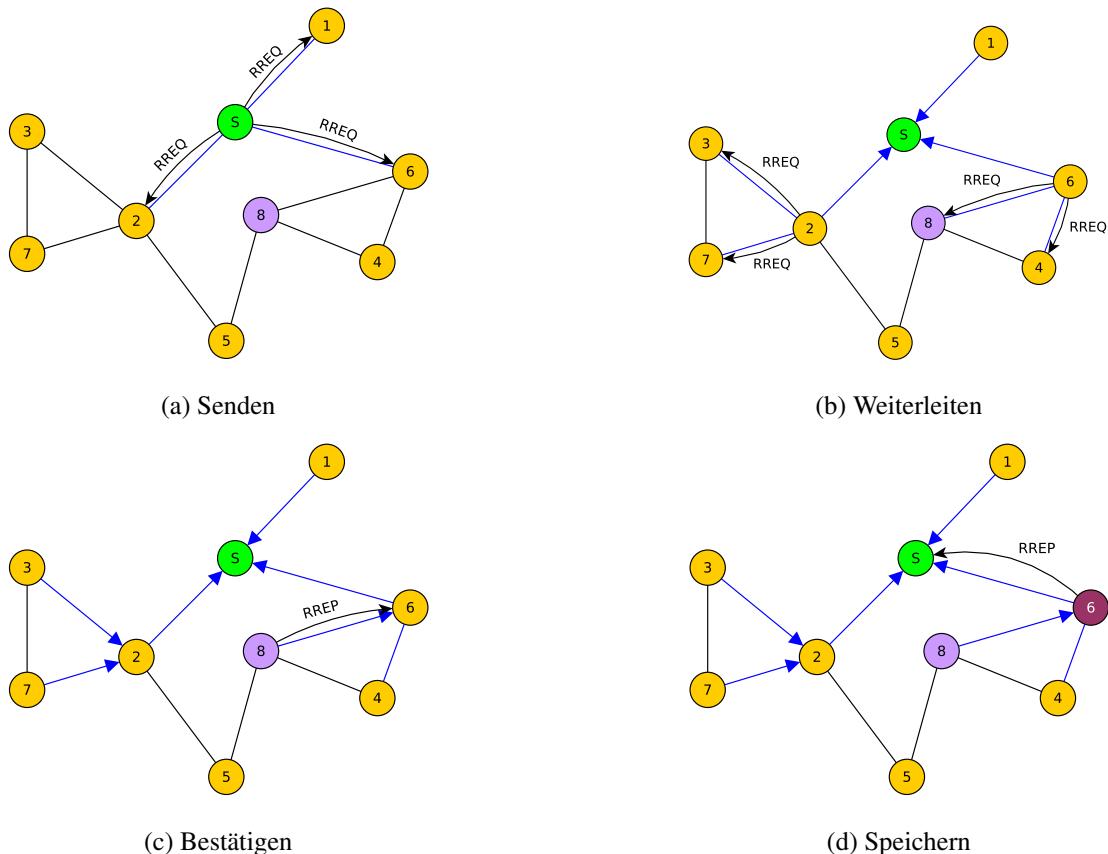


Abbildung 7.2: Ablauf beim AODV Routing

Die Abbildung 7.2 zeigt einen beispielhaften Ablauf des AODV Routing Algorithmus.

Zunächst wird (wie in Abbildung 7.2a skizziert) von dem anfragenden Knoten eine Route Request (RREQ)-Nachricht an alle in seiner Umgebung befindlichen Knotenpunkte verschickt (Broadcast). Dieses Verhalten setzt sich fort (Abbildung 7.2b) bis die RREQ-

<sup>1</sup>Die Pfade zwischen zwei Knoten werden erst bestimmt, wenn ein Datenpaket übertragen werden soll.

Nachricht den Zielknoten erreicht hat oder ein Zwischenknoten, eine schon bekannte Route zum Zielknoten vorhält. In beiden Fällen wird als Antwort auf die RREQ-Nachricht eine Route Replies (RREP)-Nachricht an den ursprünglichen Knoten zurückgesendet.

Die Pfad-Informationen zu einem bestimmten Ziel werden in den Routing-Tabellen der durchlaufenden Knoten gespeichert. Kommt es zu einer Unterbrechung der Verbindung wird eine Route Errors (RERR)-Nachricht emittiert, in der Informationen zu nicht mehr verfügbaren Knoten hinterlegt sind.

## 7.2 Positions-basierte Verfahren

Positionsbasierte Verfahren benötigen in der Regel mindestens zwei Positionsdaten, mit denen sie arbeiten können. Zum einen die eigene Position, zum anderen die Zielposition und als Hilfsmittel die Positionsdaten der umliegenden Fahrzeuge oder RSUs.

### 7.2.1 Geographic Unicast Routing

Das Geographic Unicast Routing (GUR) basiert auf der Adressierung einer geographischen Position und verwendet keine Routing-Tabellen. Die Weiterleitung erfolgt bei Bedarf mittels einer gierigen (greedy) Kostenfunktion, mit welcher der minimale Abstand zu einem bestimmten Zielbereich berechnet wird. Man spricht auch vom **Greedy Forwarding**, wenn Datenpakete zum jeweils nächsten Nachbarn, der sich in Richtung der Zielposition befindet, gesendet wird. Berücksichtigt wird hierbei auch der Abstand zwischen Sende-, Weiterleitungs- und Zielknoten. Ein optimaler Ausbreitungsweg besteht, wenn Knoten für die Weiterleitung ausgewählt werden, die sich in der Nähe des Zielknotens befinden.

Illustriert sei ein Beispiel in Abbildung 7.3 zum „Greedy Forwarding“, bei dem der Knoten ① eine Nachricht an den Knoten ⑧ senden möchte. Durch den beschränkten Kommunikationsradius von ① wird das Paket zunächst an ⑦ geschickt, da dieser den kürzesten Abstand zum Ziel ⑧ besitzt, und sich außerdem noch in dessen Empfangsreichweite befindet. Dieser Prozess wird so lange fortgesetzt, bis das Paket beim Zielknoten eingegangen ist.

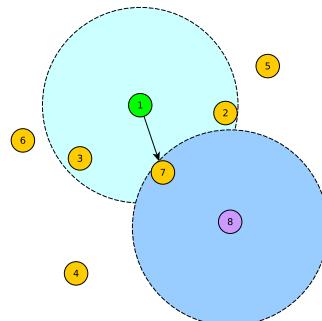


Abbildung 7.3: Greedy Forwarding

Die Nachbarn eines Sende-Knotens werden durch das sogenannte „Beacon“-Forwarding identifiziert. Dabei werden die Positionen aller Nachbarn bestimmt, die sich *1-Hop* vom Sender entfernt befinden. Ein „Beacon“ selbst ist ein kleiner Sender (Transmitter), der sich an einer bekannten Position befindet. Er sendet in bestimmten Abständen eine Information darüber, wer er ist, und, wo er sich befindet.

Die Auswahl, welcher umliegende Knoten vom Sender benutzt wird, um eine Nachricht weiter zu leiten, wird vom Sender selbst bestimmt.

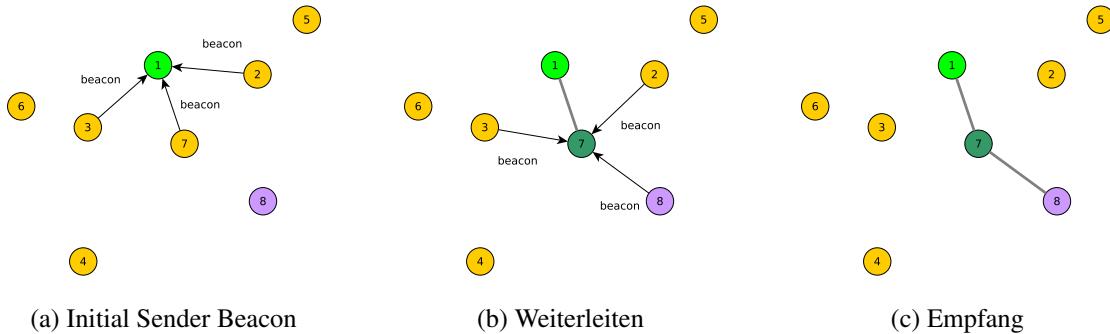


Abbildung 7.4: Ablauf von Geographischen Unicast Routing

Der Vorteil liegt in der Einfachheit des Algorithmus, zusätzlich ist er schneller als AODV. Gegen den Algorithmus spricht, dass er in vereinzelten Situation dazu führen kann, dass eine Nachricht in einer Art Kreis gefangen ist, oder auch komplett Fehl schlägt, wenn der Empfängerknoten sich nicht in der Sendereichweite eines Senders befindet.

Anfangs empfängt der Sende-Knoten die Positionsdaten der umliegenden Knoten durch die übermittelten *Beacons*. Unter Abschätzung der Position des Zielknotens wird der naheliegendste Knoten ausgewählt, der sich in Richtung des Zielknotens befindet (siehe Abbildung 7.4). Dieser Prozess wird fortgesetzt, bis der Zielknoten erreicht wurde. Es zeigt sich auch das Problem, wenn sich der Zielknoten nicht in der Sendereichweite eines der Knoten befindet. Das führt dazu, dass entweder eine Schleife entsteht, bei der immer der Knoten in der näheren Umgebung ausgewählt wird, oder eine Zustellung gänzlich nicht möglich ist.

### 7.2.2 Greedy Perimeter Stateless Routing

Der in [19] publizierte Greedy Perimeter Stateless Routing (GPSR) Algorithmus verwendet „Greedy Forwarding“, solange es die Situation zulässt. Unter Umständen kann es passieren, dass ein Nachrichtenpaket nicht weitergeleitet wird, auch wenn Pfade existieren, die dies zulassen würden. Hierzu müsste sich ein Sende-Knoten dazu entscheiden, die Daten zu einem weiter vom Ziel entfernten Knoten zu senden. Dies wird durch das „Greedy Forwarding“ jedoch nicht unterstützt.

In der Abbildung 7.5 ist die oben angesprochene Problematik versinnbildlicht. Der Knoten ④ befindet sich am nächsten zum Zielpunkt ① (lokales Maximum), dessen Kommunikationsradius der Entfernung zwischen ① und ④ entspricht. Obwohl es zwei Pfade ( $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$  und  $4 \rightarrow 5 \rightarrow 6 \rightarrow 1$ ) zum Ziel gibt, werden diese durch den „Greedy Forwarding“-Algorithmus nicht verwendet. Hierdurch wird das Datenpaket nicht weitergeleitet.

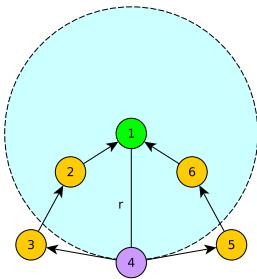
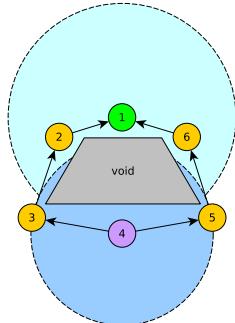
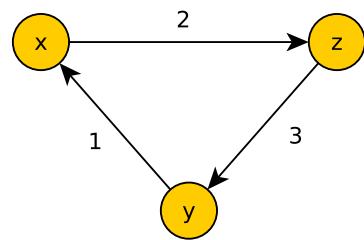


Abbildung 7.5: Problem beim Greedy Forwarding

Zur Umgehung solcher Situation verwendet GPSR einen weiteren Ansatz, das „Perimeter Routing“ unter Anwendung der „Rechten-Hand-Regel“ und einer Methode zum Durchlaufen eines „planaren Graphen“.



(a) leere Zone (void)



(b) Rechte Hand Regel

Abbildung 7.6: Perimeter Forwarding

Angenommen, die Überlappung der beiden Funkradien vom Sender ④ und Empfänger ① in Augenschein, so fällt auf, dass sich in diesem Bereich keine anderen Knotenpunkte befinden. Aus der Sicht vom Sender ④ kann keine Nachricht weitergeleitet werden, umgangssprachlich wird der Ausdruck „void“-Bereich (siehe Abbildung 7.6a) verwendet. Es muss nun also eine Möglichkeit gefunden werden, das Datenpaket um diesen leeren Bereich umzuleiten.

Dies geschieht in diesem Fall mit der **Rechten-Hand-Regel (Perimeter-Modus)** (dargestellt in Abbildung 7.6b), bei der die Datenpakete gegen den Uhrzeigersinn an den nächsten Nachbarn gesendet werden. Der GPSR-Algorithmus merkt sich, bei welchem Knoten der „Perimeter-Modus“ zum Einsatz kam, um Daten-Schleifen zu vermeiden. Diese Vorgehensweise lässt sich allerdings nur auf einen planaren Graphen<sup>2</sup> anwenden. Die gebräuchlichsten Varianten und Algorithmen sind der Relative Neighborhood Graph (RNG) oder Gabriel Graph (GG).

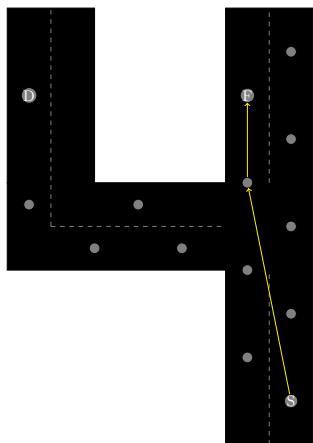
<sup>2</sup>Ein Graph der keine sich überschneidenden Kanten beinhaltet.

### 7.2.3 Greedy Perimeter Coordinator Routing

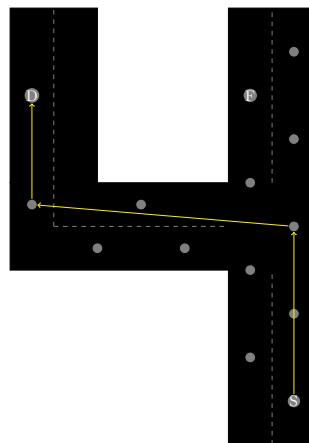
Erfahrungen zeigten, dass das GPSR sich sehr gut eignet für Situationen in denen die Umgebung einer hohen Fluktuation unterzogen ist. Problematisch wird dieser Ansatz in städtischen Gebieten, die eine Vielzahl an Funk-Hindernissen wie Gebäude und dergleichen aufweisen.

Mit dem in [20] gezeigten Greedy Perimeter Coordinator Routing (GPCR) Verfahren, wird eine Verbesserung von GPSR erreicht. Vor allem die intensive Berechnung des planaren Graphen entfällt. Hinzu kommt eine Strategie zur Vermeidung von lokalen Maxima<sup>3</sup>.

Der Ansatz besteht darin, ein Datenpaket entlang von Straßen immer zu einer Kreuzung zu schicken. Knotenpunkte an den Kreuzungen dienen als Koordinatoren, die darüber entscheiden, in welche Richtung das Paket weitergesendet werden soll. Der Datentransfer entlang von Straßen wird ebenfalls mit einer „Greedy“ Funktion durchgeführt, welche die Daten zum nächsten Nachbarn in Richtung der Zielkoordinaten schicken. Der Greedy-Ansatz ist hierbei eingeschränkt, sodass die Daten vorzugsweise an Kreuzungselemente weitergereicht werden. Die Koordinatoren an den Kreuzungen entscheiden dann über den weiteren Verlauf, dies geschieht mit der „Rechten Hand Regel“, wie sie schon bei GPSR eingesetzt wird.



(a) Problem mit lokalen Maximum



(b) Restricted Greedy

Abbildung 7.7: Ablauf des GPCR

In Abbildung 7.7a ist das Problem des lokalen Maximums, das durch das Greedy Forwarding entsteht, gezeigt. Dem gegenüber steht in Abbildung 7.7b die Anwendung mittels GPCR zur Vermeidung des Problems mit dem lokalen Maximums.

---

<sup>3</sup>Nachrichten werden nur in die Nähe des Ziels versendet, es kann aber zu einer Abweichung führen, beispielsweise einer Parallelstraße, etc

### 7.3 Karten-basierte Verfahren

Kartenbasierte Verfahren nutzen anders als die zuvor erwähnten keine Weiterleitung und Hopping Techniken im eigentlichen Sinne zur Verbreitung der Daten. Hier besteht der Ansatz in der Nutzung der Verkehrsinfrastruktur und die Einbeziehung von Kartenmaterial zur Berechnung der bestmöglichen Strecke zur Verbreitung eines Datenpaketes. Diese Verfahren haben ihre Berechtigung dahingehend, dass mittlerweile in Fahrzeugen ein Kartendienst vorhanden ist.

#### 7.3.1 Global State Routing

In dem Artikel [21] wird ein kartenbasiertes Verfahren Global State Routing (GSR) vorgestellt, das zu dem Zeitpunkt mittels eines Simulationsmodell evaluiert wurde. Es wird ein Reactive Location Service (RLS) zur Bestimmung von gewünschten Kommunikationspartnern verwendet. Hierbei schickt der anfragende Knoten mittels *Flooding* eine „Positionsanfrage (position request)“ an Knoten mit einer bestimmten Kennung. Erhält ein Knoten mit dieser Kennung die Anfrage, sendet dieser eine „Positionantwort (position reply)“ an den anfragenden Knoten zurück. Optimierungen sollen verhindern, dass ähnliche Netzwerkauslastungen wie beim „Broadcast Storm Problem“ entstehen.

Nach dem Austausch der Sender- und Empfangsinformationen berechnet der Sender-Knoten mit Hilfe des vorliegenden Kartenmaterials einen Pfad (der Kreuzungspunkte bis zum Zielknoten enthält), dem die Daten folgen müssen. Es ist hierbei dem Anwender überlassen, ob die Daten der Kreuzungspunkte mit in den Paket-Header geschrieben oder von jedem Knoten, der das Paket weiterleitet, neu berechnet werden. Zwischen zwei Kreuzungspunkten werden die Pakete mittels „Greedy Forwarding“ weitergeleitet, da sich hier keine Hindernisse befinden, die das Funksignal stören könnten.

In der bisherige Umsetzung wird der Pfad zwischen Sender und Empfänger mittels des *Dijkstra Algorithmus*<sup>4</sup> bestimmt.

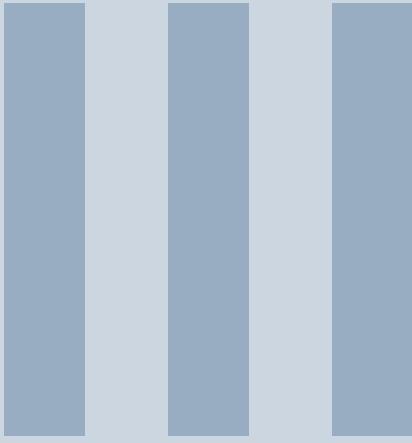
#### 7.3.2 Erweiterungen zum GSR

Eine Erweiterung zum GSR ist Security-Aware Ad-Hoc Routing (SAR) [22], der zusätzlich eine Liste von Kreuzungen im Nachrichten-Header mitsendet. Zusätzlich werden Pakete in einen Nachrichten-Puffer gespeichert und für eine Zeit vom Knoten mit befördert.

Eine andere Erweiterung zum GSR, das nur mit statischen Kartenmaterial arbeitet, ist Anchor-based street and traffic aware routing (A-STAR) [23], nicht zu verwechseln mit dem bekannten A\*-Suchalgorithmus, bei dem weitere dynamische Kartenschichten mit aktuellen Informationen wie Bus-Routen mit einbezogen werden.

---

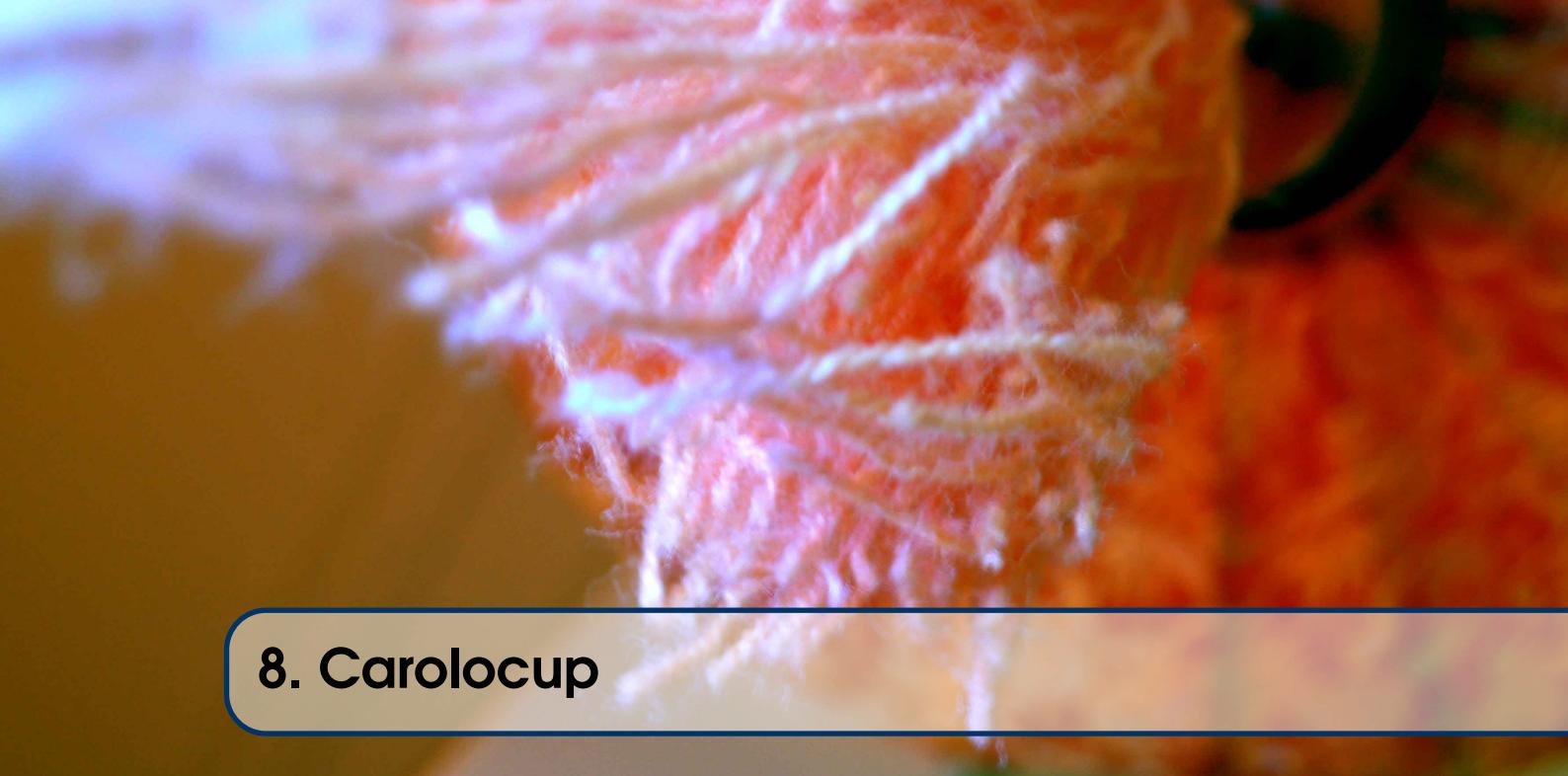
<sup>4</sup>Der Dijkstra Algorithmus berechnet ausgehend von einem Startknoten immer den kürzesten Weg zu einem Zielknoten innerhalb eines kantengewichteten Graphen.



# Projektumgebung

<b>8</b>	<b>Carolocup .....</b>	<b>77</b>
8.1	Der Wettbewerb	
8.2	Das FU-RacingTeam	
8.3	Fuhrpark	
8.4	Rennstrecken	
<b>9</b>	<b>Berlin-United .....</b>	<b>85</b>
9.1	Framework	
9.2	Software-Tools	





## 8. Carolocup

“ Champions werden nicht in Trainingshallen gemacht. Champions werden durch etwas gemacht, das sie in sich tragen: ein Verlangen, einen Traum, eine Vision. Sie brauchen außergewöhnliche Ausdauer, sie müssen ein wenig schneller sein, sie brauchen die Fähigkeiten und den Willen. Aber der Siegeswillen muss stärker sein als die Fähigkeiten. ”

---

Mohammad Ali, \*1942

Diese Kapitel erläutert das Konzept des Carolocups und welche Herausforderungen für die einzelnen Wettbewerbsteilnehmer bestehen. Im Anschluss kommt es zu einer Vorstellung des FU-RacingTeams, das sich mit der Entwicklung von autonomen Modellfahrzeugen an der Freien Universität Berlin beschäftigt und regelmäßig an dem Carolocup und anderen Veranstaltung teilnimmt.

## 8.1 Der Wettbewerb

Der *Carolo-Cup* [68] ist ein jährlich stattfindender Wettbewerb, der sich an Studierende richtet, die ein Interesse an der Entwicklung und Umsetzung eines autonom fahrenden Modellfahrzeugs zeigen. Veranstaltet wird der Wettbewerb in Braunschweig unter der Leitung der TU-Braunschweig.



Abbildung 8.1: Logo des Carolo-Cups

Als Vorgabe und gleichzeitig auch als Reglementierung für die Fahrzeuge, gilt ein Größenverhältnis von 1:10 zur realen Umgebung. Weiterhin müssen die Autos mit einem Elektromotor ausgestattet sein, andere Varianten wie Motoren mit Kraftstoffeinspritzung sind nicht erlaubt. Des Weiteren sind auf jedem Fahrzeug Signalleuchten zur Richtungsanzeige (Blinker), zum Anzeigen eines Bremsvorgangs und zur Signalisierung der Nutzung einer Fernsteuerung vorgeschrieben. Seit dem Jahr 2015 müssen Taster am Fahrzeug angebracht sein, mit denen ein Jury-Mitglied die einzelnen Wettbewerbsszenarien starten kann.

Um die Einstiegshürden für den Wettbewerb niedrig zu halten, bieten die Organisatoren zwei Arten an. Zum einen den *Junior-Cup* [57] und zum anderen den *Hauptwettbewerb* [58]. Die Regeln für den Junior-Cup sind identisch mit denen des Hauptwettbewerbs, jedoch in abgeschwächter Form. Dies äußert sich beispielsweise darin, dass keine Hindernisse und auch keine Fehlstellen im Parkour existieren.

In einer realitätsnahen Fahrumgebung müssen Disziplinen absolviert werden. Diese sind an Bedingungen aus dem alltäglichen Straßenverkehr angelehnt. Die Hauptpunkte sind dabei :

- **Das Fahren auf einem unbekannten Parkour**

Zunächst besteht hier die Herausforderung sich innerhalb einer Spur zu bewegen und diese möglichst fehlerfrei, sowie schnell zu absolvieren. Fehlerfrei bedeutet in diesem Zusammenhang, dass sich mindestens drei Räder innerhalb der Fahrspur befinden müssen.

Der Schwierigkeitsgrad wird in einem zweiten Durchgang erhöht. Zur Spurhaltung, muss das Fahrzeug auf Hindernisse achten. Dazu zählen Ausweichmanöver und das beachten der Vorfahrt an Kreuzungssituationen. Für die Bewältigung der Aufgaben steht eine begrenzte Zeit zur Verfügung.

- **Finden und manövrieren in eine Parklücke**

Das Fahrzeug fährt in diesem Szenario entlang eines Parkstreifens und muss potentielle Parklücken erkennen. Nach dem Regelwerk [58] gibt es drei unterschiedlich große Flächen, die zum Parken genutzt werden können. Beim Einparken, spielt es keine Rolle welches Manöver (vorwärts oder rückwärts einparken) gewählt wird. Wichtig ist hierbei, dass kein schon geparktes Auto (symbolisiert durch weiße Pappkisten) berührt wird. Ebenfalls sollte sich das Fahrzeug nach dem Vorrang innerhalb der Parkmarkierung befinden.

In den einzelnen Disziplinen können je nach Fahrweise Plus- oder Minuspunkte erworben werden. Am Ende des Wettbewerbs werden diese als Endresultat aufsummiert und mit den anderen Teams verglichen.

Neben diesen Tätigkeiten, die hauptsächlich das Ingenieurs-Geschick und die programmiererischen Fähigkeiten der Studierenden zeigen sollen, wird an jedes Team die Aufgabe gestellt, das Konzept der Hard- und Software Entwicklung vorzustellen. Ebenso fließen die Punkte Energiebilanz und Kostenfaktor mit ein, die von jedem Team berücksichtigt werden müssen.

Ein Gremium von Vertretern aus der Wissenschaft und Wirtschaft bilden die Jury, welche mit der Bewertung der einzelnen Teams beauftragt ist. Darunter fallen Experten von Bosch, IAV, VW, BMW und viele weitere.

## 8.2 Das FU-RacingTeam

Das FU-RacingTeam entstand aus der Idee heraus, am Carolocup-Wettbewerb teilzunehmen. Geboren wurde die Idee bei einem weihnachtlichen Umtrunk im Jahre 2011. Ein Jahr zuvor experimentierten Studierende im Rahmen eines Softwareprojektes mit zwei autonom fahrenden Modellautos. Ziel war es, diese gegeneinander in einem Wettrennen antreten zu lassen. Die Ergebnisse blieben allerdings aus. Aber aus einigen motivierten Teilnehmern dieses Kurses entwuchs das erste *RacingTeam*, unter den Gründern Steffen Heinrich und Naja von Schmude.



Abbildung 8.2: RacingTeam Logo

Mittlerweile besteht das Team aus Studierenden unterschiedlicher Fachrichtungen und Nationalitäten. So arbeiten nicht nur Bachelor- oder Master-Studenten der Informatik, sondern auch Mathematiker und Studierende aus Indien mit an den jeweiligen Aufgaben.

Die Schwerpunkte bei der Entwicklung eines Modellfahrzeug bestehen indes in der Konstruktion eines funktionstüchtigen Fahrwerks, der Programmierung von Mikrokontrollern zur Ansteuerung der Motoren und Auslesen von Sensorwerten als auch der Bildverarbeitung, zum Extrahieren von Spurmodellen und der Hinderniserkennung.

Zur Motivation der Studierenden lassen sich Abschlussarbeiten zu gewählten Themen verfassen. In jüngster Zeit entstanden so Arbeiten über das Erkennen von Hindernissen mittels einer Stereokamera [73], die Realisierung einer Fahrspurerkennung [74] [75], der Vergleich verschiedener Algorithmen zur Umsetzung eines Einparkverhaltens [76], das Erkennen von sowie Reagieren auf Kreuzungssituationen [77] und ein Kontroller zur Fahrregelung [78].

### 8.3 Fuhrpark

Das FU-RacingTeam besaß zur Zeit der Drucklegung drei Generationen an Modellfahrzeuge. Ein Hauptunterschied zu anderen Teams besteht in der Spezialisierung der Auswertung von Kameradaten zur Umfelderkennung. Anderen Teams bemühen hierzu eine Menge Sensoren zur Abstands- oder Hinderniserkennung durch Infrarot-, Ultraschall-Sensoren oder Laserscanner. Durch das Vermeiden unterschiedlicher Sensoren erhält man so einen robusteren Hardwareaufbau. Diese Herangehensweise ist bis zur jüngsten Fahrzeuggeneration wiederzufinden.

Zur Unterscheidung der einzelnen Fahrzeuge wurden sie mit einem Namen getauft. Ab der zweiten Fahrzeuggeneration folgt die Namensgebung den ersten Affen, die in den Weltraum entsandt wurden.

#### 8.3.1 Gorilla-Klaus

Bei *Gorilla-Klaus* handelt es sich um das erste Fahrzeug, das innerhalb von anderthalb Monaten entwickelt wurde und am Carolo-Cup 2012 teilnahm.



Vision	SingleVision
Hauptrechner	IGEPv2
MikroController	Arduino Nano V3
Camera-System	NET GmbH iCube USB Kamera Model NS1500CU
Fahrtregler	AI Pro Reverse Digital
Odometrie	Megatron Impulsgeber

Die erste autonome Fahrt unternahm Gorilla-Klaus kurz vor dem Carolocup-Wettbewerb. Herausfordernd gestaltete sich das langsame Fahren beim Einsatz der Fernbedienung<sup>1</sup>, das Aufgrund des verbauten Motors nur unbefriedigend möglich war.

Er ist heute nicht mehr im Einsatz aufgrund einer gebrochenen Antriebsachse und der veralteten Hardware.

<sup>1</sup>Im Regelwerk des Carolocup-Wettbewerbes ist eine maximale Geschwindigkeit von 0,3 m/s während der Benutzung der Fernbedienung vorgeschrieben

### 8.3.2 Able

Zum Carolo-Cup 2013 wurde ein völlig neues Auto konzipiert, mit dem Augenmerk auf Stabilität und Modularität.



Vision	OmniVision
Hauptrechner	Odroid-X2
MikroController	Otto-V2
Sensoren	IMU, Tickgeber
Fahrtregler	Faulhaber

Dank des robusten Konzepts der Fahrzeugmechanik wurde Able bei den Folgewettbewerben in den Jahren 2014 und 2015 sowie auf wissenschaftlichen Veranstaltungen (wie der Langen Nacht der Wissenschaften) eingesetzt.

### 8.3.3 Miss Baker

Vom Fahrwerksaufbau identisch zu *Able*, ist *Miss Baker* eine Weiterentwicklung in Sachen Computertechnik und Kamerasystem. Höhepunkt ist hier das verwendete Stereokamerasystem, mit dem ein weiteres Sichtfeld als auch die räumliche Erkennung von Hindernissen ermöglicht wird.



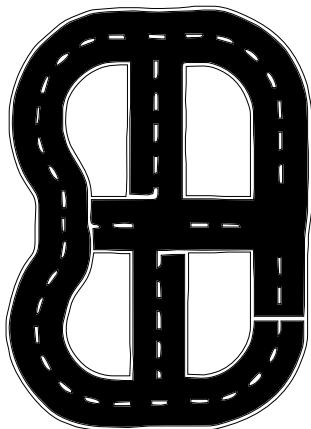
Vision	OmniVision + StereoVision
Hauptrechner	Odroid-XU
MikroController	Otto-V2
Sensoren	IMU, Tickgeber
Fahrtregler	Faulhaber

Derzeitige Problematiken mit der Spannungsversorgung ließen es leider noch nicht zu, dass Miss Baker zum produktiven Einsatz kam.

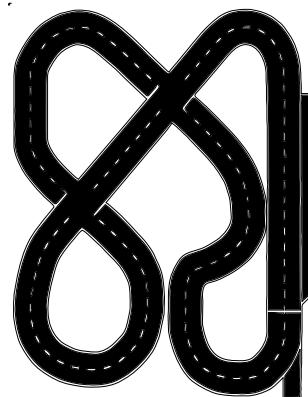
## 8.4 Rennstrecken

Neben den Fahrzeugen existieren zwei Teststrecken, die zur Erprobung der entwickelten Algorithmen Verwendung finden. Mit der kleinen Laborteststrecke lassen sich schnell Evaluierungen erreichen. Sie hat eine Abmessung von 6x4 Metern und bietet einen kleinen Rundkurs mit einem in der Mitte liegenden Kreuzungsszenario.

Die größere Teststrecke besitzt die Maße 8x10 Metern und ist einer möglichen Strecke wie sie auch beim Carolocup angewendet wird nachempfunden. Hier lassen sich auch Einparkmanöver neben den schon auf der kleinen Teststrecke erwähnten Disziplinen testen.



(a) Kleine Laborteststrecke



(b) Große Teststrecke

Abbildung 8.3: Teststrecken des FU Racing-Teams





## 9. Berlin-United

“ Wer Türme bauen will, muss lange beim Fundament verweilen. ”

---

Anton Bruckner, österreichischer Komponist, 1824-1896

Dieser Abschnitt geht auf die Grundlagen der Software-Architektur ein, die in dieser Arbeit zur Umsetzung der Fahrzeugkommunikation verwendet wurden. Zunächst wird das Framework und dessen Funktionsweise erläutert. Anschließend werden Tools vorgestellt zur Unterstützung des Entwicklungs-Prozess dienen.

## 9.1 Framework

In dem Projekt des FU-RacingTeams wird das Berlin-United Framework eingesetzt. Es basiert auf dem „NaOTH Software Architecture for an Autonomous Agent“ [69] und wurde an den eigenen Bedürfnissen angepasst und stellenweise komplett neu entwickelt.

Das Framework ist an das Konzept einer „Blackboard Architektur“ angelehnt.

Die Idee dahinter ist die Zusammenarbeit von einzelnen Teilprozessen die ihre Daten in einer hierarchischen Form organisieren und an einer zentralen Stellen (dem Blackboard) ablegen. Die einzelnen Prozesse kommunizieren niemals direkt miteinander sondern nur über das Blackboard, indem sie die Daten anfordern (require) oder sie auf das Blackboard schreiben (provide).

Veranschaulichen lässt sich dieses Konzept mit einer Expertengruppe in einem Raum mit einer Tafel, auf der eine Problemstellung umrissen ist. Wenn ein Experte etwas zur Lösung an der Tafel beiträgt, können es alle anderen sehen, ihre Meinung dazu äußern und dies ebenfalls auf der Tafel festhalten. Dieser Prozess wird solange fortgeführt bis das ursprüngliche Problem gelöst ist.

Unterteilt wird die Architektur des Berlin-United Frameworks in drei wesentliche Komponenten:

- **Repräsentation**

Hierbei handelt es sich um eine Struktur zur Zwischenspeicherung und Veröffentlichung von Daten. Module nutzen Repräsentationen für den Informationsaustausch zwischen den einzelnen Modulen.

- **Modul**

Ein Modul stellt eine Ausführungseinheit dar, in welcher die eigentliche Logik zur Bearbeitung einer Aufgabe steckt. Ergebnisse von Berechnungsschritten werden in Repräsentationen gespeichert. Die Anzahl der Repräsentationen ist dabei nicht beschränkt. Fordert ein Modul Informationen aus Repräsentationen an, die nicht zum eigenen Modul gehören, sind diese nur lesbar verfügbar.

- **Modul Manager**

Ein Modul-Manager organisiert die Ausführung einzelner Module. Das Framework bietet die Möglichkeit Ereignis- oder Zeit- basierte Manager zu verwenden.

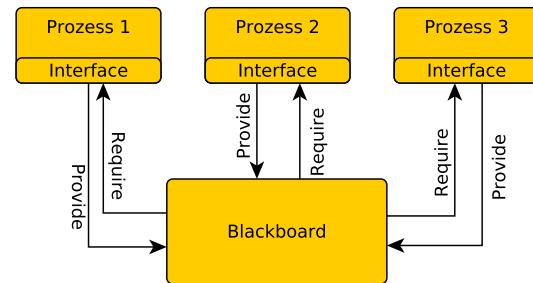


Abbildung 9.1: Blackboard-Struktur

Der Datenverarbeitungsprozess folgt den Regeln der „Sense-Think-Act“ Software-Architektur, die typischerweise im Robotik-Umfeld anzutreffen ist.

Es beginnt mit der **Wahrnehmung (Sense)** bei der Daten von Sensoren beschafft und im Vorfeld bei Bedarf gefiltert werden. Im Bezug auf die Modell-Fahrzeuge sind dies das Kamerasytem, die Inertial Measurement Unit (IMU) und der Tickgeber für die zurückgelegte Strecke.

Daraufhin folgt die **Verarbeitung (Think)** der zuvor gewonnenen Daten, die zur Entscheidungsfindung beitragen. Es werden beispielsweise Berechnungen zur Erkennung von Fahrbahn und Hindernissen für die Navigation oder die Lokalisierung auf einer Karte durchgeführt.

Zum Schluss folgt die **Umsetzung (Act)** bei der die Aktuatoren angesteuert werden, um die Geschwindigkeit und Richtung von einem Fahrzeug zu ändern.

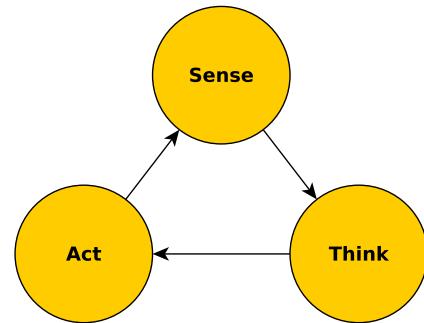


Abbildung 9.2: Sense-Think-Act

### 9.1.1 Protobuf

Das Framework bietet den Austausch von Nachrichten über ein Netzwerk mit Hilfe von Google's „Protocol Buffers“ [70] an. Mit dieser Technologie lassen sich Plattform und Programmiersprachen unabhängige Datenstrukturen beschreiben, die später zur Serialisierung oder zur Speicherung verwendet werden können.

Einzelne Datensätze werden als sogenannte Nachricht (Message) abgebildet. Sie enthalten eindeutige Einträge die einen Datentyp und einen Namen spezifizieren, die als vorausgesetzt, optional oder mehrmalig markiert werden können. Bei den Datentypen kann es sich um Zahlen (ganzzahlig oder Gleitkomma), Wahrheitswerte (Wahr oder Falsch), Zeichenketten oder selbst definierte Nachrichten handeln.

Der Entwicklerleitfaden von Protobuf gibt hierzu ein kleines Beispiel.

```

1 message Person {
2     required string name = 1;
3     required int32 id = 2;
4     optional string email = 3;
5
6     enum PhoneType {
7         MOBILE = 0;
8         HOME = 1;
9         WORK = 2;
10    }
11
12    message PhoneNumber {
13        required string number = 1;
14        optional string type = 2 [default = "MOBILE"];
15    }
16
17    repeated PhoneNumber phones = 4;
18}
  
```

```

13     required string number = 1;
14     optional PhoneType type = 2 [ default = HOME];
15   }
16
17   repeated PhoneNumber phone = 4;
18 }
```

codes/protobuf/person\_example.proto

Man erkennt die Struktur mit der eine Person abgebildet werden kann, die einen Namen und eine Identifikationsnummer besitzt. Optional lässt sich eine E-Mail Adresse angeben. Weiterhin lassen sich ein oder mehrere Telefonnummern mit unterschiedlichen Ausprägungen (Mobilfunknummer, Festnetz oder Geschäftlich) festhalten.

## 9.2 Software-Tools

Neben dem Framework stellt der Berlin-United Softwarebestand Werkzeuge zur Verfügung mit denen der Zustand der Software überwacht werden kann. Ein wichtiger Vertreter ist das Programm „FU-Remote“.

Die in Java geschriebene Software ist modular aufgebaut und kann durch sogenannte Plugins in ihrer Funktionalität erweitert werden.

Hauptsächlich wird das Tool dazu benutzt um Darstellungen von Berechnungen anzusehen. Ebenfalls lassen sich Status-Informationen darstellen oder Einstellungen bezüglich der Hardware, wie dem Kamerasystem eines Fahrzeugs vornehmen.

Zur Softwarequalitätskontrolle werden im BerlinUnited-Framework sogenannte „Unit- oder auch Modul-Tests“ verwendet.

Mit diesem Testprozess lassen sich einzelne Module auf ihre korrekte Funktionsweise hin überprüfen.

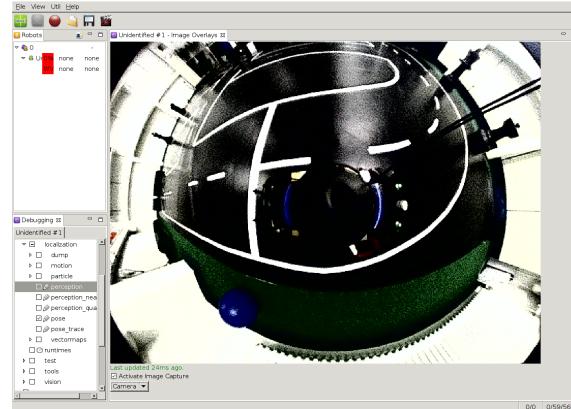


Abbildung 9.3: Darstellung FU-Remote

# IV

## Implementierung

<b>10</b>	<b>Architektur .....</b>	<b>91</b>
10.1	Aufbau	
10.2	Lokalisierung	
10.3	Modulübersicht	
10.4	Datenaustausch	
10.5	Datenverwaltung	
10.6	Zeitsynchronisierung	
<b>11</b>	<b>Anwendungsfälle .....</b>	<b>97</b>
11.1	Hinderniserkennung	
11.2	Verkehrsschild Geschwindigkeitsbegrenzung	
11.3	Ampelszenario	





## 10. Architektur

“ Man glaubt an den Weltuntergang, weil man die Architektur Gottes für so baufällig hält wie jene des Menschen. ”

---

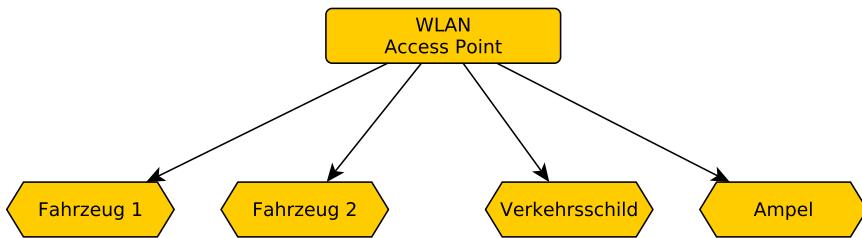
Germund Fitzthum, österreichischer Aphoristiker, 1938

In diesem Abschnitt wird die Umsetzung der zuvor erläuterten theoretische Grundlagen dargestellt. Dies beinhaltet den Aufbau der Kommunikationsbausteine sowie die Implementierung der ETSI Referenz-Architektur in das BerlinUnited Framework.

## 10.1 Aufbau

Die in dieser Arbeit entwickelte Fahrzeug-Kommunikations-Architektur für autonome Modellfahrzeuge basiert auf einem drahtlosen 2.4 GHz Netzwerk. Hintergrund ist die verbreitete Verfügbarkeit und die große Akzeptanz dieser Funktechnologie.

Den Rahmen bilden ein Wireless-LAN Access Point als Basisstation und die einzelnen Teilnehmer (Autos, RSUs), die sich mit dem Netzwerk verbinden. Jede Einheit im Netzwerk erhält eine eindeutige Identifikationsnummer (IP-Adresse), über die sie ansprechbar ist.



## 10.2 Lokalisierung

Ein Grundelement der Fahrzeugkommunikation liegt in der Bestimmung der eigenen sowie der Position der anderen Verkehrsteilnehmer. Die Positionsbestimmung in geschlossenen Gebäuden und Räumen ist mittels GPS nicht möglich. Aus diesem Grund müssen andere Wege gefunden werden, die eine Lokalisierung der Modell-Fahrzeuge ermöglicht. In der Arbeit [79] wird auf ein Konzept der Lokalisierung mit Hilfe einer Karte eingegangen. Dieses Konzept wurde im Rahmen des Carocolcup Wettbewerbes für das FU-RacingTeam entwickelt. Es werden über die am Fahrzeug befindliche Kamera Bilder aufgenommen. Von den Bildern werden Straßenzüge extrahiert und mit der im Auto befindlichen Karte abgeglichen. Mit diesem Verfahren lässt sich die Position des Fahrzeugs innerhalb der Karte abschätzen. Genauere Ergebnisse werden durch die zusätzliche Verwendung von Odometrie-Daten erreicht.

Die mit dem Verfahren gewonnenen Lokalisierungsdaten werden in dieser Arbeit für die Positionsbestimmung verwendet. Sie werden durch eine Repräsentation im BerlinUnited Framework bereitgestellt.

## 10.3 Modulübersicht

In Hinsicht auf die Umsetzung der ETSI Referenz Architektur, wurden diverse Module entwickelt mit denen die Fahrzeugkommunikation und deren Anwendung realisiert wurde. Grundlegend wurde die Design-Struktur so erstellt, dass sie mit der Referenz-Architektur im Einklang ist.

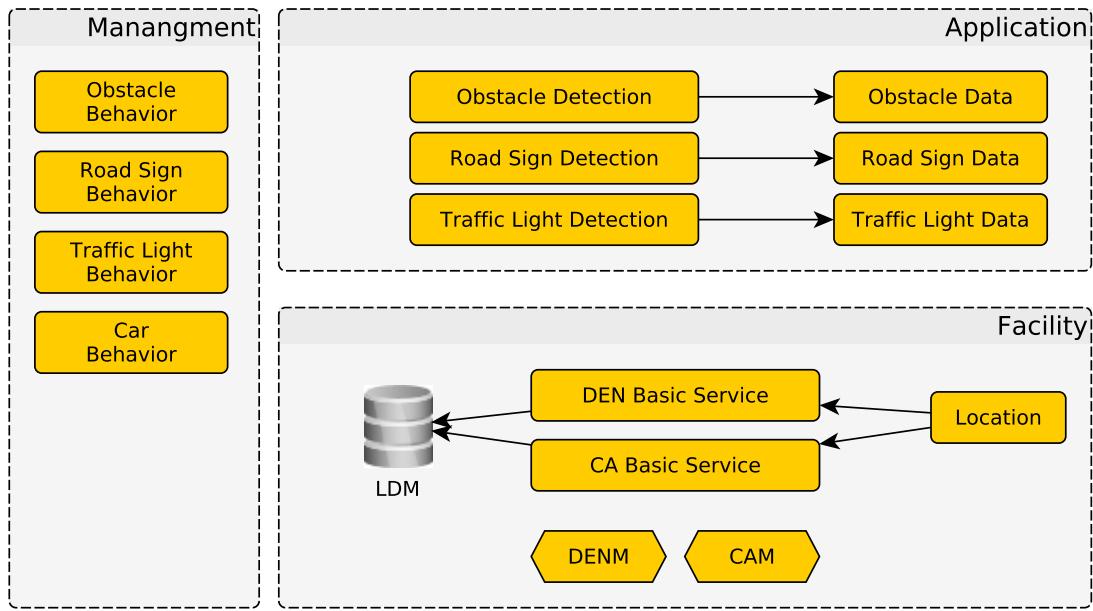


Abbildung 10.1: Modulübersicht

Für die „Facility“-Schicht wurden eine Umsetzung des „DEN Basic Service“ und des „CA Basic Service“ verwirklicht. Beide nutzen das „Location“-Modul, was mit den oben angesprochenen Algorithmus die Positionsdaten für die jeweilige ITS-S bereithält. Die Implementierung einer einfachen Datenbank für die Funktionalität der LDM rundet diesen Bereich ab.

In der Anwendungsebene (Application) befindet sich die Logik zur Erkennung von Hinderissen, Verkehrsschildern und Ampeln. Diese werden im nächsten Kapitel ausführlicher behandelt. Die gewonnenen Daten werden in Repräsentation des BerlinUnited-Frameworks für eine spätere Verwendung gespeichert.

Unter der Management-Schicht wurden die einzelnen Verhaltensmuster realisiert, die das Handeln der ITS-S bestimmen.

## 10.4 Datenaustausch

Für den Nachrichtenaustausch wird das von Google entwickelte Protobuf-Protocol verwendet. Mit ihm lassen sich wie bereits erwähnt ähnlich zu ASN.1 Datenstrukturen beschreiben. Die Entscheidung zu Gunsten von Protobuf liegt in der bereits vorhandenen Integration im BerlinUnited-Framework.

In dieser Arbeit lag die Konzentration speziell auf den Nachrichtenformaten für das „Cooperative Awareness“, da diese einen Hauptteil der Fahrzeugkommunikation ausmachen.

Entwickelt wurden für die Nachrichten CAM und DENM jeweils eine eigenständige Beschreibung der Protokolle im Protobuf-Stil. Für die Anforderungen zur Ausführung

im Innenbereich von Gebäuden fanden einige Modifikationen zum gängigen ETSI Standard statt.

Es wurden weitere Datenfelder hinzugefügt, die es erlauben einen Kommunikationsbereich zu definieren. Die Ursachen hierzu werden weiter unten angeführt. Ebenso sind Felder für die Abmessungen des Fahrzeugs vorgesehen.

Das Versenden der einzelnen Nachrichtenpaket erfolgt mittels Broadcasting über das UDP. Die Daten werden zunächst zentral zum Access Point und somit an alle im Netzwerk befindlichen Teilnehmer gesendet. Hierdurch ist es nicht möglich eine direkte Kommunikation zu bestimmten Fahrzeugen aufzubauen. Aus diesem Grund wurde das Konzept des Kommunikationsradius eingeführt, womit das Versenden von Nachrichten beschränkt werden kann. Durch dieses Verhalten lassen sich für die Netzwerk-Teilnehmer unterschiedliche Sendeleistungen simulieren.

In der Arbeit wurde eine kreisförmige Ausbreitungsfläche für Nachrichten implementiert. Das Zentrum befindet sich im Mittelpunkt einer ITS-S. Der Radius kann dabei beliebig gewählt werden.

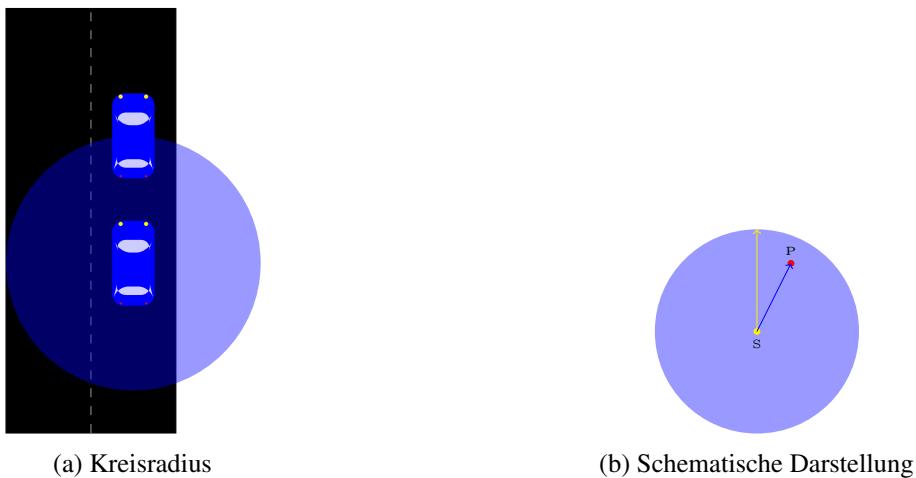


Abbildung 10.2: Kommunikationsausdehnungsschema (Kreis)

Wenn eine ITS-S ein Datenpaket durch das Broadcasting erhält, wird über die Lokalisierung bestimmt, ob das Paket auch empfangen werden konnte. Die Berechnungsfunktion nutzt hierfür den euklidischen Abstand ausgehend vom Sender, dessen zuvor festgelegten Kommunikationsbereich und den Zielpunkt, wo die Nachricht empfangen wird. Es werden dabei die aktuelle Position der empfangenden Station ( $S_x, S_y$ ) und die Position der Station, die das Paket ( $P_x, P_y$ ) sendet, verarbeitet.

$$d = \sqrt{(P_x - S_x)^2 + (P_y - S_y)^2} \quad (10.1)$$

Wenn der ermittelte Skalar (d), kleiner oder gleich dem vorgegebenen Wert für den Kommunikationsreichweite entspricht, wird das Datenpaket als empfangen angesehen und

weiter verarbeitet.

Andere Formen für die virtuelle Sendeleistung sind denkbar. Eine in Fahrtrichtung gerichtete Ellipse stellt bei Fahrzeugen einen besseren Ansatz für innerstädtische Szenarien dar. Eine zentrische Funkausbreitung ist hier in der Regel aufgrund von Hindernissen (z.B. Häuser), welche den Sendebereich einschränken, nicht möglich.

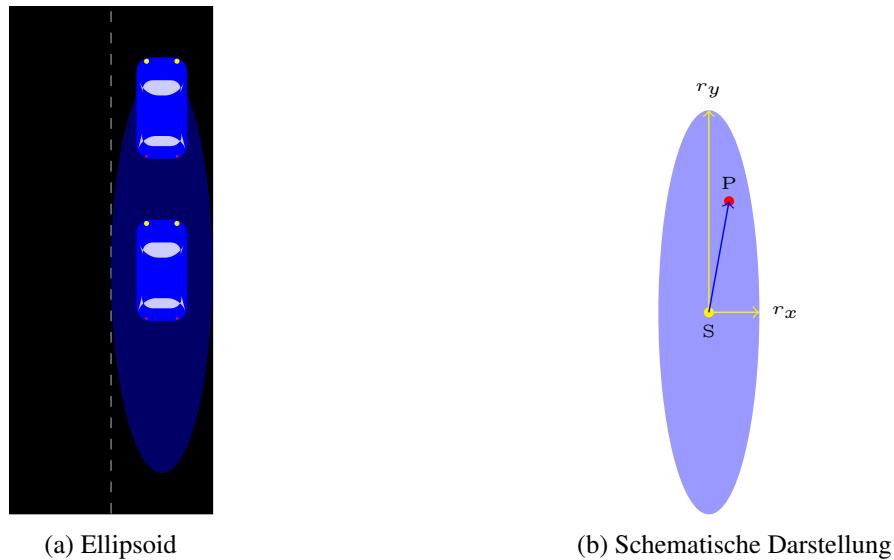


Abbildung 10.3: Kommunikationsausdehnungsschema (Ellipse)

Die Berechnung kann mit der nachfolgenden Formel durchgeführt werden. Mit ihr lässt sich bestimmen, ob ein Punkt sich innerhalb einer Ellipse befindet.

$$d = \frac{(P_x - S_x)^2}{r_x^2 + (P_y - S_y)^2} \quad (10.2)$$

Liegt der berechnete Wert (d) im Bereich zwischen 0 .. 1, befindet sich der andere Kommunikationspartner in Reichweite und es können Daten gesendet und empfangen werden.

## 10.5 Datenverwaltung

Die Verwaltung der einzelnen Nachrichten-Typen (DENM und CAM) wird gesteuert über die zwei „Basic Services (DEN und CA)“.

Der „CA Basic Service“ ist so implementiert, dass er für jede ITS-S in einem vorgegebenen Zeitintervall „Cooperative Awareness“-Nachrichten mit den aktuellen Daten der Station, wie Geschwindigkeit, Position und dergleichen erstellt und diese über die Netzwerk-Schicht versendet. Empfangene Nachrichten werden in der ebenfalls implementierten LDM hinterlegt und bei Bedarf aktualisiert. Die hinterlegten Informationen können dann

von den einzelnen Anwendungen, wie beispielsweise im folgenden Anwendungsfall der Hinderniserkennung genutzt werden.

In ähnlicher Weise verfährt der entwickelte „DEN Basic Service“. Eingehende Daten werden in der LDM hinterlegt und ausgehende versendet. Der Unterschied liegt hier in der Erstellung der ausgehenden Daten. Sie werden nicht selbst vom Service generiert, sondern stammen von den Anwendungen. Die einzelnen Anwendungen stellen einen Umstand fest, der für andere Teilnehmer informativ sein könnte. Daraufhin erstellen sie eine Aufforderung zum Versenden einer DENM. Hierzu werden alle benötigten Informationen in der Nachricht hinterlegt und anschließend an den „DEN Basic Service“ übergeben. Dieser speichert die Nachricht in der LDM ab. Zum Versenden wird durch die Datenbank traversiert und die Nachrichten, welche von der eigenen Station stammen werden in einem festgelegten Intervall übertragen.

## 10.6 Zeitsynchronisierung

Für eine Zeitsynchronisierung aller Fahrzeuge und Elemente der Infrastruktur wurde das NTP verwendet. Hierzu fungiert ein Rechner im Netzwerk als Zeitserver, der über eine stetig aktuelle Uhrzeit verfügt. Alle weiteren Elemente, die sich im Netzwerk anmelden und per C2X kommunizieren wollen, müssen zuvor ein Uhrenabgleich vornehmen.

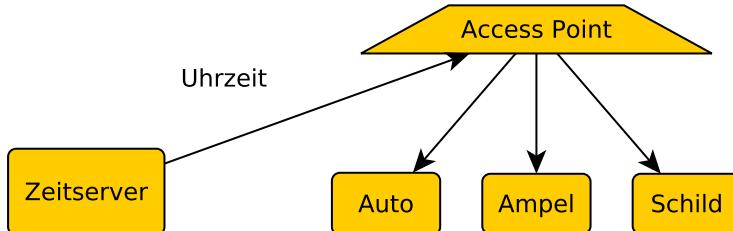


Abbildung 10.4: Aufbau Zeitserver

Für Unix-Betriebssysteme existieren Werkzeuge, mit denen ein Zeitserver oder eine Client-Anwendung erstellt werden können.

Die Verwendung von logischen Uhren wurde nicht praktiziert, da sie zwar verwendet werden können, um von Ereignissen einen kausalen Ablauf darzustellen. Der Standard für Fahrzeugkommunikation diesbezüglich die Verwendung eines Zeitstempels im UTC Format vorschreibt. Ebenfalls bedarf die Organisation der logischen Uhren einen sehr hohen Speicher- und Verwaltungsaufwand, je nach dem wie viel Elemente sich in dem ITS-Netzwerk befinden. Für jeden Netzwerk-Teilnehmer muss ein Eintrag in einer Tabelle vorgesehen werden. Zudem würde sich mit jeder neuer ITS-S der Inhalt der Datenpakete vergrößern. Unter Erwägung, dass sich die Identifikationsnummern der ITS-Ss aus Sicherheitsgründen ändern können, müssten alle darüber informiert werden und ihre Tabelleneinträge diesbezüglich aktualisieren.



## 11. Anwendungsfälle

“ Wenn die anderen glauben, man ist am Ende, dann muß man erst richtig anfangen! ”

---

Konrad Adenauer, *deutscher Politiker*, 1876-1967

In diesem Abschnitt werden ausgesuchte Anwendungsfälle vorgestellt, mit der die Implementierung der Architektur für die Fahrzeugkommunikation getestet wurde. Bei der Auswahl der Anwendungsfälle fanden Szenarien Berücksichtigung, die auch zum Carocolcup-Wettbewerb eingesetzt werden können. Gleichfalls geht es um Einsatzgebiete die grundlegend die Kommunikation zwischen Fahrzeugen und Verkehrsinfrastruktur demonstrieren.

## 11.1 Hinderniserkennung

Zu Beginn soll eine Hinderniserkennung auf Basis der Fahrzeugkommunikation vorgestellt werden. In dem Kapitel 5 über „Cooperative Awareness“ wurde der Nachrichtentyp CAM vorgestellt. Mit dieser Nachricht senden Fahrzeuge mitunter ihre aktuelle Position und Geschwindigkeit. Dieser Nachrichtentyp ist Hauptbestandteil für diesen Anwendungsfall, bei dem ein Fahrzeug über die Eigenschaften von voraus fahrenden Fahrzeugen informiert wird. Nähert sich das Fahrzeug einem vor ihm langsamer fahrenden oder stehenden Fahrzeug, wird dieses erkannt und ein Überholmanöver oder eine Vollbremsung eingeleitet.

Dieses Szenario stellt, wie bereits in Kapitel 8.1 zum CaroloCup erläutert, eines der elementaren Disziplinen im Wettbewerb dar.

In der Ausgangssituation definiert jedes Fahrzeug einen Erkennungsbereich für die Detektion von Hindernissen. In dieser Arbeit wurde ein rechteckiger Bereich, der sich vor dem Fahrzeug befindet vorgesehen. Empfängt ein Fahrzeug eine CAM, werden die Positionsdaten mit dem Erkennungsbereich abgeglichen. Befindet sich ein Fahrzeug in diesem Bereich, werden die Geschwindigkeitsvektoren beider Fahrzeuge miteinander verglichen. Der Erkennungsbereich ist zur Entscheidungsunterstützung in drei Untersektionen aufgeteilt. Im ersten Bereich wird das voraus befindliche Fahrzeug registriert. Der zweite Sektor löst einen Überholvorgang aus und im dritten Bereich ist ein geordnetes Überholen nicht mehr möglich, was zu einer (Not)-Bremsung führt, um einen Auffahruntfall zu vermeiden.

Zur Berechnung des Bremsweges sind in der Literatur [71] zwei Varianten angegeben. Die erste beschreibt den normalen Bremsweg (Anhalteweg), die zweite die Gefahrenbremsung.

**Definition 11.1.1** Der Bremsweg sei der Weg, der benötigt wird, von Beginn der Bremsung bis zum Stillstand des Fahrzeugs ohne die Reaktionszeit zu berücksichtigen.

$$\text{Bremsweg (m)} = \left( \frac{\text{Geschwindigkeit (km/h)}}{10} \right)^2 \quad (11.1)$$

Zur Bestimmung des Weges für die Gefahrenbremsung wird der Bremsweg mit der obigen Formel durch den Faktor zwei geteilt.

Der Detektionsbereich wird je nach Geschwindigkeit variable beeinflusst. Somit werden bei höheren Geschwindigkeiten ein größerer und bei niedrigen Geschwindigkeiten, ein kleinerer Bereich abgedeckt.

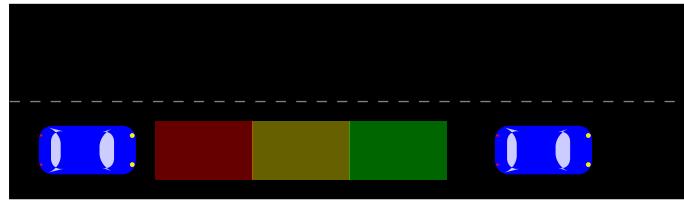


Abbildung 11.1: Zonen der Hindernis Erkennung

Die Berechnung nach der sich ermitteln lässt, ob sich ein Hindernis im Detektionsbereich befindet, wird über die Kanten der rechteckigen Fläche und den Maßen des Fahrzeugs bestimmt.

$$\text{Objekt.X} \leq (\text{Bereich.X} + \text{Weite}) \quad (11.2)$$

$$\text{Objekt.X} \geq (\text{Bereich.X}) \quad (11.3)$$

$$\text{Objekt.Y} \leq (\text{Bereich.Y} + \text{Länge}) \quad (11.4)$$

$$\text{Objekt.Y} \geq (\text{Bereich.Y}) \quad (11.5)$$

Erst wenn diese vier Bedingung zutreffen befindet sich ein Objekt im Erkennungs-Bereich. Die Positionierung des Bereichs ist so gewählt, dass er sich vor dem Fahrzeug befindet.

Im Falle eines Überholmanövers wird der Erkennungsbereich rund um das Fahrzeug ausgedehnt. Dies soll beim Einleiten des Überholmanövers sicher stellen, dass sich kein Fahrzeug auf der Überholspur befindet. Ebenfalls dient es der Kontrolle, ob das Fahrzeug (Hindernis) bereits überholt worden ist, um anschließend einen Spurwechsel auf die rechte Fahrbahn vorzunehmen.



(a) Überholmanöver einleiten

(b) Prüfen ob Hindernis überholt

Abbildung 11.2: Szenerie eines Überholmanövers

Der Prozess zum Empfang und Versenden von CAMs wird durch den „CA Basic Service“ unterstützt. Diesbezüglich wurde ein „Message Callback Handler“ vorgesehen, der beim eintreffen der Nachrichten aufgerufen wird. Diese Funktion vermittelt die Nachrichten an die LDM.

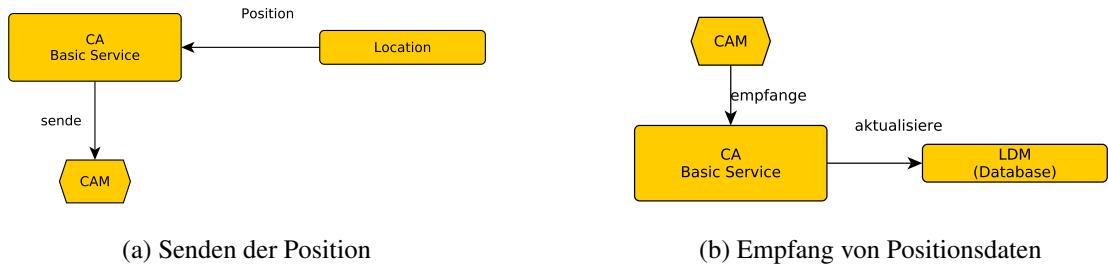


Abbildung 11.3: Senden und Empfangen von CAMs zur Positionsbestimmung

Die notwendige Erkennung von Hindernissen, wird durch das Modul „ObstacleDetector“ realisiert. Die Ergebnisse der Berechnungen werden in eine Repräsentation geschrieben. Diese Informationen bestimmen das weiterführende Verhalten des Fahrzeugs.

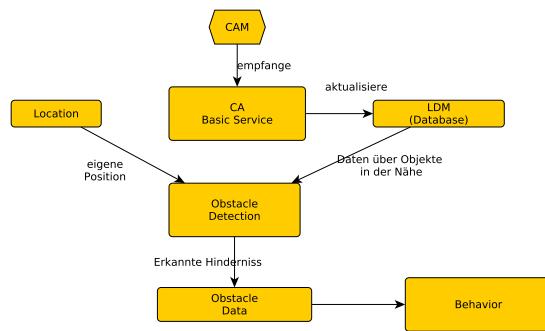


Abbildung 11.4: Module für die Hinderniserkennung

In der Arbeit wurde eine Anpassung der CAM vorgenommen, aus dem Grund, dass in der standardisierten Beschreibung keine Fahrzeugabmessungen mitgesendet werden und sich die Position des Fahrzeugs lediglich auf einen nicht näher spezifizierten Punkt beschränkt. Für einen Überholvorgang sind solche Information unabdingbar, um das Tranchieren von Fahrzeugen beim Überholen zu verhindern.

## 11.2 Verkehrsschild Geschwindigkeitsbegrenzung

Bei diesem Anwendungsfall sendet ein Teil der Verkehrsinfrastruktur, um genauer zu sein, ein Verkehrszeichen mittels einer DENM die für den kommenden Streckenabschnitt erhobene Höchstgeschwindigkeit.

Als Informationen werden die Art des Verkehrszeichen (in diesem Fall eine Geschwindigkeitsbegrenzung), welche zulässige Höchstgeschwindigkeit erlaubt ist, die Position des Verkehrszeichens und ein Relevanz Bereich (Aktivierung Bereich), ab dem das Verkehrsschild seine Gültigkeit hat, gesendet.

Der Relevanzbereich, in dem ein Fahrzeug auf die Geschwindigkeitsbegrenzung aufmerksam gemacht wird, überdeckt dabei die komplette Straßenbreite. Er geht von der Position des Straßenschildes aus und deckt sowohl den linken als auch rechten Teil des Schildes ab. Somit ist es irrelevant auf welcher Seite einer Straße das Verkehrsschild aufgestellt wird.

Das Verkehrsschild enthält einen Richtungsvektor mit dem festgelegt wird, für welche Autos das Schild von Interesse ist. Entgegenkommende Fahrzeuge ignorieren also das Schild, auch wenn sie in den Relevanzbereich eintreten.

Wenn ein Fahrzeug in den Relevanzbereich fährt, überprüft die Boardelektronik die momentane Geschwindigkeit des Fahrzeugs und gleicht diese, sollte es notwendig sein, der aktuellen Verkehrssituation an.

Die vom FU-RacingTeam verwendeten Fahrzeuge können mit einer maximalen Geschwindigkeit von ca. 4 Metern pro Sekunde fahren. Dies entspricht einem Wert von 14,4 km/h. Unter zur Kenntnisnahme, dass es sich dabei um Modelfahrzeuge im Maßstab 1:10 handelt, lässt sich annehmen das die übertragenen Geschwindigkeit auf ein normales Fahrzeug dem Faktor 10 entspricht. Somit können realistische Verkehrsschilder für die Geschwindigkeitsbegrenzung eingesetzt werden.

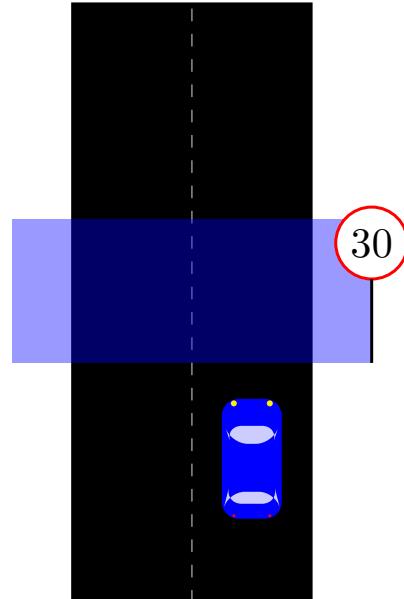


Abbildung 11.5: Szenario: Verkehrszeichen

Die Umrechnung der Modellfahrzeug in die reale Fahrzeug Geschwindigkeit erfolgt über die unten gegebene Formel.

$$V \text{ in km/h} = V \text{ in m/s} \cdot 3600 \cdot 10 \quad (11.6)$$

Die zwei festen Werte bestimmen dabei zunächst die Umrechnung von m/s in km/h (3600) und die Adaptierung vom Modell in die reale Fahrzeuggröße (10).

In Deutschland sind Geschwindigkeitsbegrenzung von 50 km/h innerhalb von Ortschaften, 80 km/h, 100 km/h und 130 km/h auf Autobahnen üblich. Die folgende Tabelle zeigt welche Geschwindigkeiten vom Modellauto erreicht werden können und welche Verkehrsschilder somit sinnvoll einsetzbar sind.

Geschwindigkeit Modellauto	reale Geschwindigkeit	mögliche Begrenzung
1,0 m/s	36 km/h	30
1,5 m/s	54 km/h	50
2,0 m/s	72 km/h	70
2,5 m/s	90 km/h	80
3,0 m/s	108 km/h	100
3,5 m/s	126 km/h	120
4,0 m/s	144 km/h	130

Tabelle 11.1: Geschwindigkeiten und Verkehrszeichen

Das Verhalten des Verkehrszeichen ist so konzipiert, dass es einmalig selbst eine DENM Nachricht in der Anwendungsschicht erzeugt und an den „DEN Basic Service“ übermittelt (triggert). Aufgrund der Besonderheit des Berlin United Frameworks mit seiner Modul und Repräsentationsebenen weicht die Vorgehensweise hier in einigen Zügen von der ETSI Spezifikation ab. In der Anwendungsschicht wird demnach eine Ereignis-Anforderung mit der DENM in einer Repräsentation gespeichert. Werden im Zuge der Ausführung die Management-Module vom Framework abgearbeitet, kommt es im ersten Schritt zur Speicherung der DENM in die LDM. Daraufhin werden die gespeicherten Nachrichten in der LDM, die von der eigenen ITS-Station (Verkehrszeichen) stammen vom „DEN Basic Service“ in einer bestimmten Frequenz im Netzwerk verteilt.

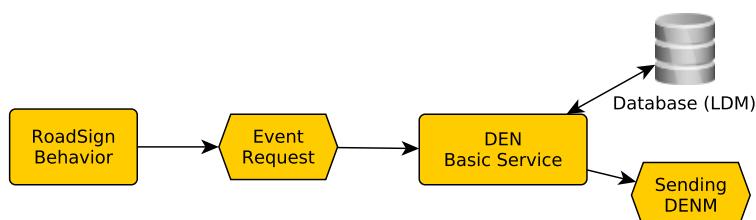


Abbildung 11.6: Senden der DENM

Die Strategie zum Versenden der DENM wird mittels einfachen Flooding und Broadcast über das Netzwerk realisiert. Andere ITS-Stationen (Fahrzeuge) entscheiden anhand ihrer Position, ob die empfangene Nachricht für sie relevant sind, andernfalls werden sie ignoriert.

Der Empfang der DENM Nachricht wird durch den „DEN Basic Service“ organisiert, der je nach Relevanz die eingehende Nachricht in einer lokalen Datenbank speichert. Damit stehen die Nachrichten für eine weitere Auswertung in der Anwendungsschicht zur Verfügung.

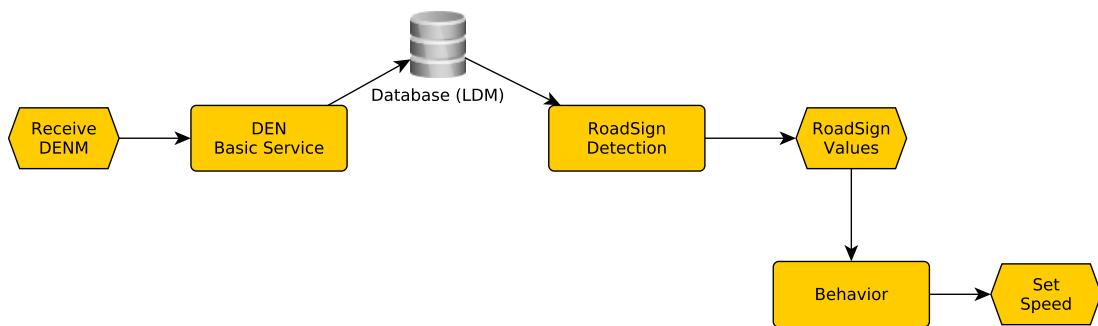


Abbildung 11.7: Empfang der Nachricht und Verhalten beim Fahrzeug

Für das Szenario mit dem Verkehrszeichen, wird über das Modul „RoadSign-Detection“ geprüft, ob sich das Fahrzeug im Relevanzbereich für die Geschwindigkeitsbeschränkung befindet. Im weiteren Verlauf werden die Fahrtrichtung bestimmt und mit dem Verkehrszeichen abgeglichen. Stimmen Position und Fahrtrichtung überein, aktualisiert das Modul die ihm zugehörige Repräsentation „Road-Sign Values“. Befinden sich in der Repräsentation für den weiteren Fahrverlauf relevante Daten, passt das Verhaltensmodul des Fahrzeugs die Geschwindigkeit entsprechend des erkannten Verkehrszeichens an.

### 11.3 Ampelszenario

Der letzte in dieser Arbeit behandelte Anwendungsfall besteht aus einer Ampel und einem Fahrzeug. Die Ampel besitzt dabei definierte Zustände, eine Position an der sie sich befindet und einen Richtungsvektor für den der Ampelzustand gültig ist. Der Nachrichtenaustausch wird über das DENM-Protokoll abgehandelt. In der Nachricht werden Position, Richtung, Signalzustand der Ampel und die Umschaltphasen versendet.

Für die einzelnen Zustände einer Ampel sind folgende Punkte vorgesehen:

- **AUS**

Die Signalanlage ist nicht in Betrieb

- **Blinkend**

Die Signalanlage sendet ein Achtungssignal in Form eines gelben in geregelten Intervallen emittierten Lichts

- **Normalbetrieb**

Die Signalanlage befindet sich im Betriebsmodus mit den Ampelsequenzen rot, rot-gelb, grün und gelb.

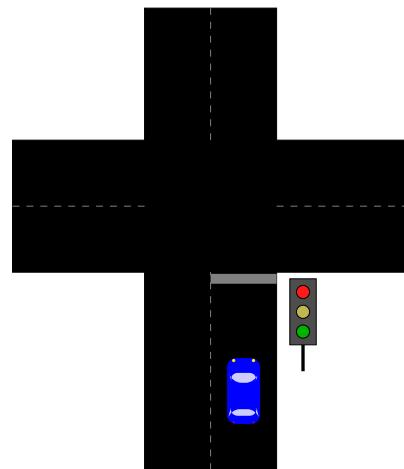
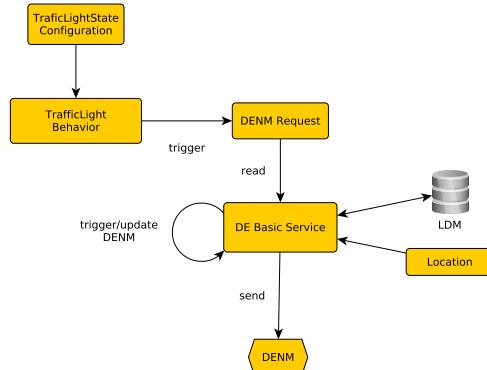
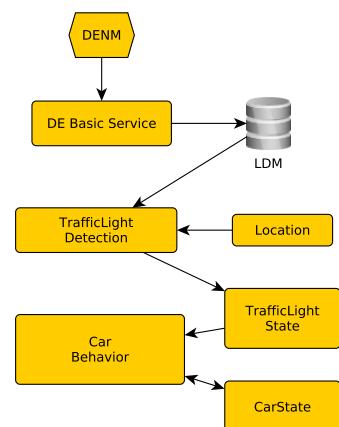


Abbildung 11.8: Szenario: Ampel

Die Ampel triggert zur Aussendung der einzelnen Zustände und Sequenzen mit Hilfe des „DEN Basic Service“ eine Nachricht. Diese wird durch das Verhaltensmodell der Ampel stetig aktualisiert.



(a) Modulaufbau der Ampel



(b) Modulaufbau des Fahrzeugs

Abbildung 11.9: Modulübersicht der Ampel und des Fahrzeugs

Ein ankommendes Fahrzeug empfängt die DENM der Ampel, wenn es sich im Funkbereich befindet. Der „DEN Basic Service“ verarbeitet die Nachricht und aktualisiert die interne Datenbank (LDM) in der alle Vorkommnisse hinterlegt werden.

Das Modul „TrafficLightDetection“ verarbeitet den aktuellen Zustand der Ampel und berechnet die Entfernung. Die prozessierten Daten werden in eine Repräsentation abgelegt und vom Verhaltensmodul interpretiert.

Das Verhaltensmuster des Fahrzeugs ist so entwickelt, dass beim ausgeschalteten Zustand der Ampel das Fahrzeug normal weiterfährt. Der Ansatz dahinter geht davon aus, dass sich das Fahrzeug auf einer Hauptstraße befindet.

Im Normalzustand der Ampel reagiert das Auto auf die bestimmten Ampelphasen. Bei rot, bleibt das Fahrzeug vor der Ampel stehen und setzt seine Fahrt weiter fort, sobald die Ampel umschaltet. Sind die Ampelsequenzen gelb oder grün und steuert das Auto gerade auf die Ampel zu, werden die Abstände zwischen Fahrzeug und Ampel errechnet. Dies dient der Feststellung, wann die Ampel erreicht wurde. Die Implementierung sieht es vor, dass das Auto je nach Zeit der Ampelumschaltung und Entfernung die Geschwindigkeit drosselt oder erhöht.

Befindet sich die Ampel im blinkenden Zustand initiiert das Verhalten ein langsames anfahren an die Ampel und beschleunigt, nachdem der Ortspunkt der Ampel erreicht und überschritten wurde.

### 11.3.1 Aufbau der Signalanlage

Da im Repertoire vom FU-RacingTeam bisher keine Ampelanlage existierte, wurde auf Grundlage eines Odroids, der auch als Hauptrechner für die Fahrzeuge dient, als Steuereinheit für eine selbst konstruierte LED-Matrix verwendet. Der Odroid besitzt General-purpose input/outputs (GPIOs), dessen Kontakte (Pins) unabhängig für Ein- und Ausgaben definiert werden können. Bei dem verwendeten Odroid-XU können pro Pin jeweils 1.8V abgeführt werden. Dies ist für das Betreiben von typischen Light-Emitting Diodes (LEDs) zu gering. Die nachfolgende Tabelle listet typische Flussspannungswerte für LEDs auf.

LED-Farbe	LED-Flussspannung $U_{LED}$
infrarot	1,2V ... 1,8V
rot	1,6V ... 2,2V
gelb	1,9V ... 2,5V
grün	1,9V ... 2,5V
blau	3,0V ... 4,0V
weiß	3,0V ... 4,0V

Tabelle 11.2: LED Betriebsspannungen

Idealerweise lässt sich an den GPIOs auch eine Spannungsquelle von 5V abnehmen. Diese wird als Quellspannung zum Betreiben der LEDs mit einem Vorwiderstand verwendet. Drei der GPIOs mit 1.8V Ausgangsspannung werden zur Schaltung eines Transistors benutzt, der die jeweilige LED an- oder ausschaltet.

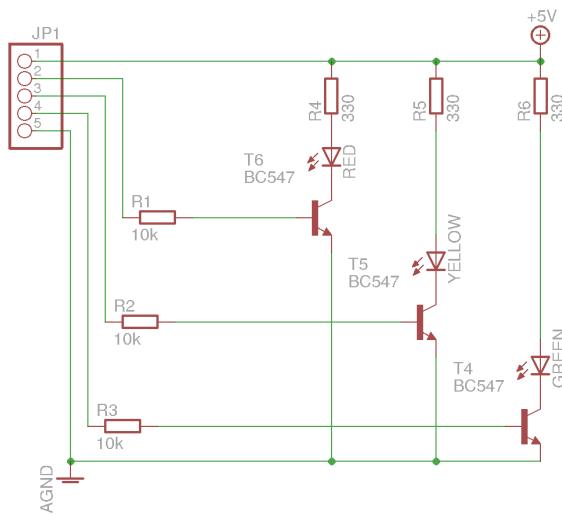


Abbildung 11.10: Schematische Darstellung der Ampel-Elektronik

Die Transistoren dienen hierbei als Schalter, die an ihrem Basis-Eingang einen geringen Strom entgegennehmen und die am Kollektor angeschlossene LED mit der Masse zu verbinden, um einen geschlossenen Stromkreis herzustellen.

# Auswertung



<b>12</b>	<b>Reflektion .....</b>	<b>109</b>
12.1	Implementierung	
12.2	Bewertung	
12.3	Ausblick	





## 12. Reflektion

“ Aufhören können, das ist nicht eine Schwäche, das ist eine Stärke. ”

---

Ingeborg Bachmann, 1926-1973

Dieser Abschnitt debattiert die Implementierung und beschäftigt sich mit den Herausforderungen, die während der Arbeit entstanden sind. Neben einer Bewertung, werden Erweiterungsmöglichkeiten behandelt. Abschließend wird ein Ausblick gegeben und alternative Ansätze diskutiert.

## 12.1 Implementierung

Der gezeigte Anwendungsfall für die Hinderniserkennung funktioniert weitestgehend robust. Als Schwachstelle bei der bisherigen Umsetzung ist anzumerken, dass in Situationen in denen ein Fahrzeug sich in einer Kurve befindet, die Erkennungsrate durch den Grad des Kurvenradius und der aktuellen Geschwindigkeit beeinträchtigt ist. Prinzipiell wurden keine Situationen mit entgegenkommenden Verkehr getestet, So dass das Haupteinsatzszenario eher auf Autobahnen als auf städtische Umgebungen angesiedelt ist. Eine weitere Einschränkung besteht in der Anzahl der Fahrspuren. Geschuldet durch die Anlehnung an den Carocolup berücksichtigt die Implementierung derzeit nur zwei Fahrspuren. Genauso zeigt sich, dass die Erweiterung der CAM um die Fahrzeugdimension sich als hilfreich für das Ausweichen und wieder Einscheren erweist.

Die Umsetzung für den Anwendungsfall mit dem Verkehrszeichen zur Geschwindigkeitsregulierung zeigt sich akkurat. Passiert ein Fahrzeug einen solchen Abschnitt, regelt es selbstständig die Geschwindigkeit. Aufgrund der engen Kapazität an Hardware-Geräten ließ sich derzeit nur mit einem Verkehrszeichen arbeiten. Softwareseitig besteht diesbezüglich keine Einschränkung.

Beim Anwendungsfall für die Ampel kommt es zu einer Problematik mit der Beachtung der Vorfahrtssituation. Dies tritt durch die Annahme auf, dass ein Fahrzeug bei einer ausgeschalteten Signalanlage davon ausgehen kann, dass es sich um eine Hauptstraße handelt. Besonders deutlich zeigt sich das Phänomen bei einer Kreuzung. An dieser Stelle kann es keine zwei Hauptstraßen geben. Umgehen ließe sich die Problematik mit einer Erweiterung der Schild-Erkennung, die auch eine Erkennung für Vorfahrtszeichen zulässt. Diese Verkehrsschilder müßten zusätzlich zu den Ampeln mit aufgestellt werden. Ein anderer Ansatz wäre die Einbeziehung der Hinderniserkennung für Kreuzungssituation mit einer Ampel. Demnach ließe sich nach der Straßen-Verkehrs-Ordnung (StVO) ein rechts vor links Gebot mit integrieren. Des Weiteren ist zu sagen, dass die Umsetzung der Signalanlage mit dem DENM-Protokoll verwirklicht wurde. Nach dem ETSI-Standard wird für Signalanlagen ein spezielles Nachrichtenformat vorgeschlagen. Mit dem Signal Phase and Time (SPAT)-Protokoll [43] [36] werden momentane und zukünftige Signalzustände und Umschaltzeiten ausgesendet. Die Daten werden periodisch an andere Verkehrsteilnehmer mittels single-hop versendet.

Bewußt wurde in dieser Arbeit auf die Integration von Sicherheitsrelevanten Mechanismen verzichtet, da diese selbst noch nicht im vollen Umfang standardisiert sind und sich somit noch in der Entwicklung befinden. Weiterführend ist es für den Ansatz mit Modelfahrzeugen derzeit noch unkritisch zu sehen, da sich der Einsatz im Laborumfeld befindet und hierdurch keine gewollten Angriffe auf die Protokolle und Mechanismen zu erwarten sind. Es schließt natürlich nicht aus, dass Bestrebungen in Richtung Sicherheitskonzepte auch für den Modell-basierten Ansatz stattfinden können.

Die Synchronisierung der einzelnen Elemente im Netzwerk birgt mit der, in der Arbeit angewendeten Praxis noch die Gefahr, dass nach einiger Zeit die Uhren auf den einzelnen Rechner ihre Synchronizität verlieren und ein erneuter Abgleich vorgenommen werden muss. Zur Umgehung dieser Problematik müsste ein Ableichmechanismus in das Berlin-United Framework integriert werden, dass in einer bestimmten Periode den Zeitversatz korrigiert. Eine andere Option wäre durch einen Hintergrundprozess im Linux System gegeben.

Der verwendete Kommunikationsradius berücksichtigt derzeit nicht das Abschwächen der Sendeleistung beispielsweise durch Häuser, Brücken oder ähnliches. Es stellt einen idealisierten Sendebereich dar. Zum Versenden der Pakete wurde in den gezeigten Anwendungsfällen das „Single-Hopping“ verwendet. Somit übertragen die einzelnen ITS-S ihre Daten direkt an die nächsten Nachbarn. Einsatzgebiete für „Multi-Hopping“ wurden Aufgrund der beschränkten Hardwareressourcen in dieser Arbeit nicht betrachtet, sie werden dennoch von der softwareseitigen Implementierung unterstützt.

## 12.2 Bewertung

In dieser Arbeit sind die grundlegenden Bestandteile und Funktionsweisen der Fahrzeugkommunikation dargestellt worden. Insbesondere deren Anwendung auf den Bereich von Modellfahrzeugen. Somit erlaubt sie die Implementierung, Entwicklung und Evaluierung von Algorithmen nicht nur in einem Simulationsumfeld.

Generell kann die Fahrzeugkommunikation zu einer rücksichtsvolleren Fahrweise und somit zur Vermeidung von Unfällen beitragen. Zu bedenken ist, dass die Technik weiterhin kritisch betrachtet werden sollte. Vor allem im Bezug auf die Sicherheit der Systemkomponenten und den daraus resultierenden Angriffspunkten. Außerdem wird die Technik nicht gleichzeitig in allen Fahrzeugen integriert sein, sodass nur ein Teil von dieser Technologie profitieren.

Der in der Arbeit verwendete 2.4 GHz WLAN Standard dient hier lediglich als kostengünstige Alternative zum bisher noch nicht weit verbreiteten und damit stark vermarkteten 5.9 GHz Standard, mit den Frequenzen wie sie beim 802.11p Standard benutzt werden. Die Netzwerktechnologie kann in einem späteren Rahmen ausgetauscht werden ohne das die darüber liegende „Facility oder Application“ Schicht davon betroffen wäre, dessen Hauptbestandteil diese Arbeit darstellt.

Beschränkungen unterlag diese Arbeit im Bezug auf die Verfügbarkeit von Hardwaremodulen. Jede ITS-S, sei es ein Fahrzeug oder eine RSU (Ampel, Verkehrsschild), wurde als eigenständiges Gerät verwendet ohne auf Virtualisierung, Simulationen oder dergleichen zurückzugreifen. Es sollte gezeigt werden, das es möglich ist, auch im kleinen Rahmen Anwendungen für die Fahrzeugkommunikation zu verwirklichen.

### 12.3 Ausblick

Trotz erheblicher Anstrengungen bei dieser Arbeit, so bleiben doch immer noch Punkte offen, die zu einer Verbesserung beitragen können.

Eine zukünftige Verbesserung läge in der Änderung des Algorithmus zur Berechnung des Bereichs, in dem Fahrzeuge für die Hinderniserkennung erkannt werden. Eine mögliche Alternative wäre durch einen polynomiellen Ansatz, der sich der Straßenlage anpasst, gegeben.

Ein weiteres interessantes Szenario, dass in dieser Arbeit nicht untersucht wurde, ist die Staumeldung mit der ankommenden Fahrzeuge gewarnt werden und nach Ausweichrouten suchen. Diese Anwendung ließe sich ebenfalls gut umsetzen, da die Modellautos einem vorgegebenen Pfad folgen, der mit Kenntnis der Umgebung und dem vorhandenen Kartenmaterial manipuliert werden müsste.

Die Firma „u-blox“ arbeitet derzeit an Modulen zur Fahrzeugkommunikation. Im ersten Quartal 2016 soll es hierzu erste Entwicklerversionen geben und eine Serienfertigung sei für Ende 2016 geplant. Mit diesen Modulen, die im 802.11p Standard funken, könnte eine realitätsnahe und unabhängig vom BerlinUnited-Framework operierende Kommunikation zwischen Modelfahrzeugen entwickelt werden. Bisherige auf dem Markt erhältliche und sogenannte „OnBoards-Units“ waren für den Einsatz in Modelfahrzeugen bisher zu überdimensioniert in ihren Abmessungen. Schon jetzt bietet die Firma in ihrem Produktkatalog mit der THEO-P1 Serie ein kombiniertes Sende- und Empfängermodul an, das für den Einsatz von Vehicle to X (V2X) Anwendungen gedacht ist.

Ein weiterer interessanter Ansatz wäre die Verwendung des IEEE 802.11p Linux Treibers, der in [24] beschrieben und auf Github [72] veröffentlicht wurde.

Genauso ließe sich über eine Integration eines ANS.1 Compiler und Parser in das „Berlin United“-Framework nachdenken.

# Referenzen

# VI

<b>Literatur . . . . .</b>	<b>115</b>
Bücher	
Artikel	
Technische Berichte	
Online	
Abschlussarbeiten	
<b>Abkürzungsverzeichnis . . . . .</b>	<b>121</b>
<b>Abbildungsverzeichnis . . . . .</b>	<b>124</b>
<b>Tabellenverzeichnis . . . . .</b>	<b>126</b>



# Literatur

## Bücher

- [1] Frank O'Brien. *The Apollo Guidance Computer: Architecture and Operation*. 1. Auflage. Springer, 2010 (siehe Seite 15).
- [2] Prof. John Larmouth. *ASN.1 - Complete*. 1. Auflage. <http://www.oss.com/>: Open System Solution, 1999 (siehe Seite 41).
- [3] Olivier Dubuisson. *ASN.1 - Communication Between Heterogeneous Systems*. 1. Auflage. <http://www.oss.com/>: OSS Nokalva, 2000 (siehe Seite 41).
- [4] Jean-Marie Zogg. *GPS Grundlagen - User's Guide*. 1. Auflage. Zürcherstrasse 68, 8800 Thalwil, Switzerland: u-blox AG, 2003 (siehe Seite 58).
- [5] David L. Mills. *Computer Network Time Synchronization - The Network Time Protocol*. 1. Auflage. 6000 Broken Sound Parkway NW, Suite 300: CRC Press, 2006 (siehe Seite 61).
- [6] Mohammad Ilyas. *The Handbook of Ad-Hoc Wireless Networks*. 1. Auflage. CRC Press, 2000 N.W. Coporate Blvd, Boca Raton, Florida 3341: CRC Press, 2003 (siehe Seite 67).
- [7] Huaqun Guo. *Automotive Informatics and Communicative Systems (Priniciples in Vehicular Networkd and Data Exchange)*. 1. Auflage. 3 Henrietta Street, Covent Garden, London WC2E 8LU: Information Science Reference, 2009 (siehe Seite 67).
- [8] P.Nicopolidis u. a. *Wireless Networks*. 1. Auflage. The Atrium, Southern Gate, Chichester, West Sussex P019 8SQ, England: John Wiley & Sons Ltd., 2003 (siehe Seite 67).

---

## Artikel

- [9] Paul Krzyzanowski. „Lectures on distributed systems - Clock Synchronization“. In: (2009), Seiten 1–14 (siehe Seite 59).
- [10] Saurabh Ganeriwal, Srdjan Capkun und Mani B. Srivastava. „Secure Time Synchronization in Sensor Networks“. In: V.N (März 2006), Seiten 1–31 (siehe Seite 59).
- [11] Leslie Lamport. „Time, Clocks, and the Ordering of Events in a Distributed System“. In: 21.7 (März 1978), Seiten 558–565 (siehe Seite 59).
- [12] Dirk Piester, Peter Hetzel und Andreas Bauch. „Zeit- und Normalfrequenzverbreitung mit DCF77“. In: 4 (2004), Seiten 345–368 (siehe Seite 62).
- [13] Abdul Wahid, Hongseok Yoo und Dongkyun Kim. „Unicast Geographic Routing Protocols for Inter-vehicle Communications: A Survey“. In: PM2HW2N '10 (2010), Seiten 17–24. DOI: 10.1145/1868612.1868616. URL: <http://doi.acm.org/10.1145/1868612.1868616> (siehe Seite 67).
- [14] P. Muhlethaler, A. Laouiti und Y. Toor. „Comparison of Flooding Techniques for Safety Applications in VANETs“. In: (Juni 2007), Seiten 1–6. DOI: 10.1109/ITST.2007.4295910 (siehe Seite 68).
- [15] S. Biswas, R. Tatchikou und F. Dion. „Vehicle-to-vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety“. In: *Comm. Mag.* 44.1 (Jan. 2006), Seiten 74–82. ISSN: 0163-6804. DOI: 10.1109/MCOM.2006.1580935. URL: <http://dx.doi.org/10.1109/MCOM.2006.1580935> (siehe Seite 68).
- [16] S.K. Dhurandher, M.S. Obaidat und M. Gupta. „A reactive Optimized Link State Routing protocol for Mobile ad hoc networks“. In: (Dez. 2010), Seiten 367–370. DOI: 10.1109/ICECS.2010.5724529 (siehe Seite 68).
- [17] P. Jacquet u. a. „Optimized link state routing protocol for ad hoc networks“. In: (2001), Seiten 62–68. DOI: 10.1109/INMIC.2001.995315 (siehe Seite 68).
- [18] Charles E. Perkins und Elizabeth M. Royer. „Ad-Hoc On-Demand Distance Vector Routing“. In: () (siehe Seite 69).
- [19] Brad Karp und H.T.Kung. „GPSR: Greedy Parameter Stateless Routing for Wireless Networks“. In: (2000) (siehe Seite 71).
- [20] Christian Lochert u. a. „Geographic Routing in City Scenarios“. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 9.1 (Jan. 2005), Seiten 69–72. ISSN: 1559-1662. DOI: 10.1145/1055959.1055970. URL: <http://doi.acm.org/10.1145/1055959.1055970> (siehe Seite 73).
- [21] Christian Lochert u. a. „A Routing Strategy for Vehicular Ad Hoc Networks in City Environments“. In: (2003), Seiten 156–161 (siehe Seite 74).
- [22] Seung Yi, Prasad Naldurg und Robin Kravets. „Security-aware Ad Hoc Routing for Wireless Networks“. In: MobiHoc '01 (2001), Seiten 299–302. DOI: 10.1145/501449.501464. URL: <http://doi.acm.org/10.1145/501449.501464> (siehe Seite 74).

- 
- [23] Boon-Chong Seet u. a. „A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications“. English. In: Lecture Notes in Computer Science 3042 (2004). Herausgegeben von Nikolas Mitrou u. a., Seiten 989–999. DOI: 10.1007/978-3-540-24693-0\_81. URL: [http://dx.doi.org/10.1007/978-3-540-24693-0\\_81](http://dx.doi.org/10.1007/978-3-540-24693-0_81) (siehe Seite 74).
  - [24] R. Lisovy, M. Sojka und Z. Hanzalek. „IEEE 802.11p Linux Kernel Implementation“. In: (Dez. 2014) (siehe Seite 112).

## Technische Berichte

- [25] Google. *Google Self-Driving Car Project: Monthly Report*. Technischer Bericht. Google, Nov. 2015 (siehe Seite 15).
- [26] Dr. Jadranka Dokic, Dr. Beate Müller und Dr. Gereon Meyer. *European Roadmap - Smart Systems for Automated Driving*. Technischer Bericht. European Technology Platform on Smart Systems Integration (EPoSS), Apr. 2015 (siehe Seite 16).
- [27] Funkschau. *Können Ultrakurzwellen den Kraftverkehr sicherer gestalten?* Technischer Bericht. Funkschau, Jan. 1936 (siehe Seite 17).
- [28] L. Ward und Dr. M. Simon. *Intelligent Transportation Systems Using IEEE 802.11p*. Technischer Bericht. Rohde & Schwarz, 2014 (siehe Seite 27).
- [29] ETSI. *Intelligent Transport Systems (ITS); Communication Architecture*. Technischer Bericht. ETSI, Sep. 2010 (siehe Seiten 32, 37, 38, 47, 52).
- [30] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communicattions; GeoNetworking Part 1: Requirements*. Technischer Bericht. ETSI, Nov. 2013 (siehe Seiten 34, 38, 40).
- [31] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communicattions; GeoNetworking Part 2: Scenarios*. Technischer Bericht. ETSI, Nov. 2013 (siehe Seiten 34, 40).
- [32] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communicattions; GeoNetworking Part 3: Network Architecture*. Technischer Bericht. ETSI, Dez. 2014 (siehe Seiten 34, 50).
- [33] ISO. *Intelligent transport systems – Communications access for land mobiles (CALM) – IPv6 Networking*. Technischer Bericht. ISO, 2012 (siehe Seite 34).
- [34] ISO. *Intelligent transport systems – Communications access for land mobiles (CALM) – Management*. Technischer Bericht. ISO, 2010 (siehe Seite 36).
- [35] ETSI. *Intelligent Transport Systems (ITS); OSI cross-layer topics; Part1: Architecture and addressing schemes*. Technischer Bericht. ETSI, Nov. 2012 (siehe Seite 36).
- [36] ETSI. *Intelligent Transport Systems (ITS); User and Application Requirements; Part 1: Facility Layer Structure, functional requirements and specification*. Technischer Bericht. ETSI, Sep. 2013 (siehe Seiten 38, 110).
- [37] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definition*. Technischer Bericht. ETSI, Juni 2009 (siehe Seiten 46, 47).

- [38] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Application; Part2: Specification of Cooperative Awareness Basic Service*. Technischer Bericht. ETSI, Nov. 2014 (siehe Seiten 46, 48, 52).
- [39] ETSI. *Intelligent Transport Systems (ITS); V2X Applications; Part1: Road Hazard Signaling (RHS) application requirements specification*. Technischer Bericht. ETSI, Aug. 2013 (siehe Seiten 47, 49, 51).
- [40] ETSI. *Intelligent Transport Systems (ITS); V2X Applications; Part2: ??? application requirements specification*. Technischer Bericht. ETSI, Okt. 2013 (siehe Seite 47).
- [41] ETSI. *Intelligent Transport Systems (ITS); V2X Applications; Part3: Longitudinal Collision Risk Warning (LCRW) application requirements specification*. Technischer Bericht. ETSI, Nov. 2013 (siehe Seiten 47, 49, 51).
- [42] ETSI. *Intelligent Transport Systems (ITS); User and Application Requirements; Part 2: Applications and facility layer common data dictionary*. Technischer Bericht. ETSI, Sep. 2014 (siehe Seiten 48, 49, 56).
- [43] ETSI. *Intelligent Transport Systems (ITS); Harmonized Channel Specification for Intelligent Transport Systems operating in the 5 GHz frequency band*. Technischer Bericht. ETSI, Okt. 2012 (siehe Seiten 50, 110).
- [44] ETSI. *Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band*. Technischer Bericht. ETSI, Nov. 2012 (siehe Seite 50).
- [45] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)*. Technischer Bericht. ETSI, Sep. 2014 (siehe Seiten 52, 56).
- [46] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communications; Geographical Area Definition*. Technischer Bericht. ETSI, Juli 2011 (siehe Seite 53).
- [47] Colin J. Fridge. *Timestamps in Message-Passing Systems That Preserve the Partial Order*. Technischer Bericht. Department of Computer Science, Australian National University, Canberra, ACT, Feb. 1988 (siehe Seite 60).
- [48] Friedemann Mattern und Reinhard Schwarz. *Detecting Causal Relationships in Distributed Computations: In Search of the Holy Grail*. Technischer Bericht. Department of Computer Science, University of Saarland, 1994 (siehe Seite 60).
- [49] David Mills. *RFC958: Network Time Protocol (NTP)*. Technischer Bericht. Sep. 1985 (siehe Seite 61).
- [50] David Mills. *RFC4340: Simple Network Time Protocol (SNTP)*. Technischer Bericht. Jan. 2006 (siehe Seite 63).
- [51] David Mills. *RFC5905: Network Time Protocol (NTP) Version 4: Protocol and Algorithms Specification*. Technischer Bericht. Juni 2010 (siehe Seite 63).

- 
- [52] National Instruments. *Understanding the IEEE 1588 Precision Time Protocol*. Technischer Bericht. Nov. 2015. URL: <http://www.ni.com/newsletter/50130/en/> (siehe Seiten 63, 64).
- [53] Andreas Dreher und Dirk Mohl. *Precision Clock Synchronization - The Standard IEEE 1588*. Technischer Bericht (siehe Seite 64).
- [54] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communicattions; GeoNetworking Part 5: Network Architecture; Sub-part 1: Basic Transport Protocol*. Technischer Bericht. ETSI, Dez. 2014 (siehe Seite 67).
- [55] Sze-Yao Ni u. a. *The broadcast storm problem in a mobile ad hoc nework*. Technischer Bericht. Department of Computer Science und Information Engineering, National Central University, 1999 (siehe Seite 68).
- [56] C. Perkins. *RFC3561: Ad hoc On-Demand Distance Vector Routing*. Technischer Bericht. Juli 2003 (siehe Seite 69).
- [57] TU Braunschweig. *Junior-Cup Regelwerk 2015*. Technischer Bericht. Juni 2014 (siehe Seite 78).
- [58] TU Braunschweig. *Carolocup-Regelwerk 2015*. Technischer Bericht. Juni 2014 (siehe Seiten 78, 79).

## Online

- [59] C2C-CC. *Car 2 Car Communication Consortium Webauftritt*. 28. Nov. 2015. URL: <https://www.car-2-car.org> (siehe Seite 29).
- [60] ETSI. *ETSI*. 24. Juli 2015. URL: <http://www.etsi.org/> (siehe Seite 31).
- [61] ITU. *Introduction to ASN.1*. 30. Okt. 2015. URL: <http://www.itu.int/en/ITU-T/asn1/Pages/introduction.aspx> (siehe Seite 43).
- [62] TimeAndDate. *Internationale Atomzeit*. 24. Nov. 2015. URL: <http://www.timeanddate.de/zeitzonen/internationale-atomzeit> (siehe Seite 58).
- [63] NMEA. *NMEA-0183 Standard*. 23. Nov. 2015. URL: [http://www.nmea.org/content/nmea\\_standards/nmea\\_0183\\_v\\_410.asp](http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp) (siehe Seite 58).
- [64] NMEA. *Internetauftritt NMEA*. 23. Nov. 2015. URL: <http://www.nmea.org/> (siehe Seite 58).
- [65] Jörg Reitter. *Zeitsynchronisation im LAN*. 4. Nov. 2015. URL: <http://www.linux-magazin.de/Ausgaben/2004/04/Immer-im-Takt-bleiben> (siehe Seite 62).
- [66] Physikalisch-Technische Bundesanstalt Bundesministerium für Wirtschaft und Energie. *DCF77*. 7. Nov. 2015. URL: <http://www.ptb.de/cms/de/ptb/fachabteilungen/abt4/fb-44/ag-442/verbreitung-der-gesetzlichen-zeit/dcf77.html> (siehe Seite 62).

- [67] Physikalisch-Technische Bundesanstalt Bundesministerium für Wirtschaft und Energie. *Internetauftritt PTB*. 7. Nov. 2015. URL: <https://www.ptb.de/cms/> (siehe Seite 62).
- [68] TU Braunschweig. *Carolo-Cup*. 8. Sep. 2015. URL: <https://wiki.ifr.ing.tu-bs.de/carolocup/> (siehe Seite 78).
- [69] Nao Team Humboldt. *NaoTH Software Architecture*. 20. Okt. 2015. URL: <http://www.naoteamhumboldt.de/en/projects/multi-platform-robot-controller/> (siehe Seite 86).
- [70] Google. *Protocol Buffers*. 20. Okt. 2015. URL: <https://developers.google.com/protocol-buffers/> (siehe Seite 87).
- [71] Fahrschule 123. *Bremsweg berechnen*. 29. Nov. 2015. URL: <http://www.fahrschule-123.de/formeln/bremsweg/> (siehe Seite 98).
- [72] Lisovy. *IEEE 802.11p Linux Treiber Source Code*. 30. Dez. 2015. URL: <https://github.com/CTU-IIG/802.11p-linux> (siehe Seite 112).

## Abschlussarbeiten

- [73] Jannis Ihrig. „Obstacle Detection for autonomous model car using stereo vision“. Bachelor’s Thesis. Freie Universität Berlin, Jan. 2015 (siehe Seite 80).
- [74] Till-Julius Krüger. „Fahrspurerkennung mit Punktvalidierung bei autonomen Modellfahrzeugen“. Bachelor’s Thesis. Freie Universität Berlin, Sep. 2015 (siehe Seite 80).
- [75] Conrad Läßig. „Fahrspurerkennung mit RANSAC bei autonomen Modellfahrzeugen“. Bachelor’s Thesis. Freie Universität Berlin, Mai 2015 (siehe Seite 80).
- [76] Tom Bullmann. „Parkingbehavior of RC-Cars - Comparison of A\* and RRT“. Bachelor’s Thesis. Freie Universität Berlin, Sep. 2015 (siehe Seite 80).
- [77] Benjamin Zengin. „Kreuzungserkennung und -verhalten bei autonomen Modellfahrzeugen“. Bachelor’s Thesis. Freie Universität Berlin, Aug. 2014 (siehe Seite 80).
- [78] Daniel Krakowczyk. „A path following control architecture for autonomous vehicles“. Bachelor’s Thesis. Freie Universität Berlin, Okt. 2014 (siehe Seite 80).
- [79] Lukas Maischak. „Lane Localization for Autonomous Model Cars“. Master’s Thesis. Freie Universität Berlin, Juli 2014 (siehe Seite 92).



## Abkürzungsverzeichnis

<b>ABS</b>	Antiblockiersystem .....	14
<b>ACC</b>	Abstandstempomat .....	14
<b>AODV</b>	Ad-Hoc On-Demand Distance Vector .....	69
<b>ASCII</b>	American Standard Code for Information Interchange .....	58
<b>ASD</b>	Automatische Sperrdifferenzial .....	14
<b>ASN.1</b>	Abstract Syntax Notation One .....	41
<b>ASR</b>	Antriebsschlupfregelung .....	14
<b>A-STAR</b>	Anchor-based street and traffic aware routing .....	74
<b>BAS</b>	Bremsassistent .....	14
<b>BER</b>	Basic Encoding Rules .....	43
<b>BSA</b>	Basic Set of Applications .....	47
<b>BTP</b>	Basic Transport Protocol .....	67
<b>C2C</b>	Car to Car .....	26
<b>C2C-CC</b>	CAR 2 CAR Communication Consortium .....	29
<b>C2I</b>	Car to Infrastructure .....	26
<b>C2X</b>	Car to X .....	20
<b>Car2X</b>	Car to X .....	26
<b>CAM</b>	Cooperative Awareness Message .....	36

---

<b>CER</b>	Canonical Encoding Rules .....	43
<b>DCC</b>	Decentralized Congestion Control .....	50
<b>DEN</b>	Decentralized Environmental Notification .....	52
<b>DENM</b>	Decentralized Environmental Notification Message .....	36
<b>DER</b>	Distinguished Encoding Rules .....	43
<b>DSRC</b>	Dedicated Short Range Communication .....	50
<b>ESP</b>	elektronische Stabilitätskontrolle .....	14
<b>ETSI</b>	European Telecommunications Standards Institute .....	30
<b>GAF</b>	Geography Aware Flooding .....	68
<b>GG</b>	Gabriel Graph .....	72
<b>GPSR</b>	Greedy Perimeter Stateless Routing .....	71
<b>GPCR</b>	Greedy Perimeter Coordinator Routing .....	73
<b>GUR</b>	Geographic Unicast Routing .....	70
<b>GPS</b>	Global Positioning System .....	16
<b>GPIO</b>	General-purpose input/output .....	106
<b>GSR</b>	Global State Routing .....	74
<b>HMI</b>	Human Machine Interface .....	35
<b>IMU</b>	Inertial Measurement Unit .....	87
<b>ITS</b>	Intelligent Transport System .....	20
<b>ITSC</b>	ITS Communications .....	32
<b>ITS-S</b>	ITS-Station .....	32
<b>KAF</b>	Keep Alive Forwarding .....	52
<b>LDM</b>	Local Dynamic Map .....	36
<b>LDW</b>	Lane Departure Warning (Spurverlassenswarnung) .....	14
<b>LED</b>	Light-Emitting Diode .....	106
<b>LKA</b>	Lane Keeping Assistent (Spurhalteassistent) .....	14
<b>LNM</b>	Local Node Map .....	37
<b>LTE</b>	Long Term Evolution .....	43
<b>MANET</b>	Mobile Ad-Hoc Network .....	26
<b>MIB</b>	Management Information Base .....	36
<b>NMEA</b>	National Marine Electronics Association .....	58
<b>NTP</b>	Network Time Protocol .....	61

---

<b>OER</b>	Octet Encoding Rules .....	43
<b>OLSR</b>	Optimised Linked State Routing .....	68
<b>OSI</b>	Open System Interconnection .....	33
<b>PDA</b>	Personal Digital Assistant .....	38
<b>PDU</b>	Protocol Data Unit .....	53
<b>PER</b>	Packed Encoding Rules .....	41
<b>PTB</b>	Physikalisch-Technische Bundesanstalt .....	62
<b>PTP</b>	Precision Time Protocol .....	63
<b>RERR</b>	Route Errors .....	70
<b>RLS</b>	Reactive Location Service .....	74
<b>RNG</b>	Relative Neighborhood Graph .....	72
<b>RREQ</b>	Route Request .....	69
<b>RREP</b>	Route Replies .....	70
<b>RSU</b>	Road Side Unit .....	26
<b>SAP</b>	Service Access Point .....	33
<b>SAR</b>	Security-Aware Ad-Hoc Routing .....	74
<b>SIM-TD</b>	Sichere Intelligente Mobilität - Testfeld Deutschland .....	29
<b>SNTP</b>	Simple Network Time Protocol .....	63
<b>SOA</b>	Service Oriented Architecture .....	36
<b>SPAT</b>	Signal Phase and Time .....	110
<b>StVO</b>	Straßen-Verkehrs-Ordnung .....	110
<b>TAI</b>	Temps Atomique International .....	58
<b>TCS</b>	Traktionskontrolle .....	14
<b>TCP</b>	Transmission Control Protocol .....	35
<b>TDF</b>	TZIZ Distribution Francaise .....	62
<b>TLV</b>	Tag-Length-Value .....	43
<b>TPEG-RTM</b>	Transport Protocol Expert Group - Road Traffic Message .....	36
<b>TPMS</b>	Tire Pressure Monitoring System .....	14
<b>UDP</b>	User Datagram Protocol .....	35
<b>UTC</b>	Universal Time Coordinate .....	58
<b>UMTS</b>	Universal Mobile Telecommunications System .....	43
<b>VANET</b>	Vehicular Ad-Hoc Network .....	26

<b>V2V</b>	Vehicle to Vehicle .....	26
<b>V2I</b>	Vehicle to Infrastructure .....	26
<b>V2X</b>	Vehicle to X .....	112
<b>WLAN</b>	Wireless Local Area Network.....	20

# Abbildungsverzeichnis

1.1 Fahrerassistenssysteme (Quelle: Audi) .....	15
1.2 Selbst fahrende Fahrzeuge .....	16
3.1 Informationsablauf .....	26
3.2 Typische Bereiche für Car2X .....	27
3.3 Entwicklungsmitglieder (Quelle: C2C-CC) .....	29
3.4 Weg zum Standard .....	30
4.1 ITSC Szenario (Quelle: ETSI) .....	32
4.2 Architektur-Modelle .....	33
4.3 ITS Referenz Architektur - Anwendungs-Schicht .....	34
4.4 ITS Referenz Architektur - Access-Schicht .....	34
4.5 ITS Referenz Architektur - Networking&Transport-Schicht .....	35
4.6 ITS Referenz Architektur - Facilities-Schicht .....	35
4.7 ITS Referenz Architektur - Management-Schicht .....	37
4.8 ITS Referenz Architektur - Security-Schicht .....	37
4.9 Unicast (Quelle: ETSI) .....	40
4.10 Geo-Broadcast (Quelle: ETSI) .....	40
4.11 Topologie Broadcast (Quelle: ETSI) .....	41
5.1 Kommunikationskanäle CA basic Service .....	47
5.2 Struktur einer CAM .....	48
5.3 Schnittstellen des DEN basic service .....	52
5.4 Struktur einer DENM .....	53
5.5 Datenfluss einer DENM .....	54
6.1 Ablauf der Lamport-Uhr .....	60
6.2 Ablauf der Vektor-Uhr .....	61
6.3 NTP Hierarchie .....	62
6.4 Darstellung eines PTP Netzwerkes .....	64
6.5 Ablauf einer Synchronisierung mittels PTP .....	65
7.1 Ablauf von Flooding .....	68

7.2 Ablauf beim AODV Routing .....	69
7.3 Greedy Forwarding .....	70
7.4 Ablauf von Geographischen Unicast Routing .....	71
7.5 Problem beim Greedy Forwarding .....	72
7.6 Perimeter Forwarding .....	72
7.7 Ablauf des GPCR .....	73
8.1 Logo des Carolo-Cups .....	78
8.2 RacingTeam Logo .....	80
8.3 Teststrecken des FU Racing-Teams .....	83
9.1 Blackboard-Struktur .....	86
9.2 Sense-Think-Act .....	87
9.3 Darstellung FU-Remote .....	88
10.1 Modulübersicht .....	93
10.2 Kommunikationsausdehnungsschema (Kreis) .....	94
10.3 Kommunikationsausdehnungsschema (Ellipse) .....	95
10.4 Aufbau Zeitserver .....	96
11.1 Zonen der Hindernis Erkennung .....	99
11.2 Szenerie eines Überholmanövers .....	99
11.3 Senden und Empfangen von CAMs zur Positionsbestimmung .....	100
11.4 Module für die Hinderniserkennung .....	100
11.5 Szenario: Verkehrszeichen .....	101
11.6 Senden der DENM .....	102
11.7 Empfang der Nachricht und Verhalten beim Fahrzeug .....	103
11.8 Szenario: Ampel .....	104
11.9 Modulübersicht der Ampel und des Fahrzeugs .....	104
11.10 Schematische Darstellung der Ampel-Elektronik .....	106



## Tabellenverzeichnis

3.1 Weltweite Fahrzeugkommunikation (Quelle: (28)) .....	27
4.1 Datentypen von ASN.1 .....	42
11.1 Geschwindigkeiten und Verkehrszeichen .....	102
11.2 LED Betriebsspannungen .....	106



# Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben. Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

**Datum**

Berlin, 22. Januar 2016

**Unterschrift**

.....