Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Philipp Meyer

Informationssicherheit für Echtzeit-Ethernet-Fahrzeugnetzwerke

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Philipp Meyer

Informationssicherheit für Echtzeit-Ethernet-Fahrzeugnetzwerke

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Franz Korf
Zweitgutachter: Prof. Dr.-Ing. Martin Hübner

Eingereicht am: 7. Juni 2018

Philipp Meyer

Thema der Arbeit
Informationssicherheit für Echtzeit-Ethernet-Fahrzeugnetzwerke

Stichworte
Informationssicherheit, Echtzeit, Ethernet, Fahrzeug, Bordnetz, TTE, AVB, TSN

Kurzzusammenfassung
Heutige Fahrzeugnetzwerke, bestehend aus heterogenen Feldbussen, verbinden eine Vielzahl von Steuergeräten. Diese Steuergeräte sind heute angreifbar und können nach Kompromittierung genutzt werden, um die gesamte Kommunikation zu manipulieren. In Zukunft werden Ethernet-Netzwerke die Feldbusse Schritt für Schritt ersetzen. Echtzeit-Ethernet-Protokolle sollen die Funktionssicherheit des Kommunikationssystems gewährleisten. In dieser Arbeit werden die Echtzeit-Ethernet-Protokolle Time-Triggered Ethernet, Audio Video Bridging und Time-Sensitive Networking einer Sicherheitsanalyse unterzogen. Für ermittelte Schwachstellen werden Anforderungen entwickelt und bewertet. Im nächsten Schritt werden Schutzkonzepte erarbeitet, die diese Anforderungen erfüllen können. Ausgewählte Teile der Schutzkonzepte werden in einer Simulationsumgebung implementiert und abschließend in einer simulationsbasierten Fallstudie analysiert.
Philipp Meyer

Title of the paper
Security for Real-Time Ethernet In-Car Networks

Keywords
Security, Real-Time, Ethernet, Car, On-board Network, TTE, AVB, TSN

Abstract
Today's vehicle networks, consisting of heterogeneous fieldbuses, connect a multitude of control units. These ECUs are vulnerable and can be used to manipulate the entire communication. In the future, Ethernet networks will replace the fieldbuses step by step. Real-Time Ethernet protocols are designed to ensure the reliability of the communication system. In this work, the Real-Time Ethernet protocols Time-Triggered Ethernet, Audio Video Bridging and Time-Sensitive Networking are subject of a security analysis. For found vulnerabilities Requirements are developed and evaluated. In the next step protection concepts are developed that can meet these requirements. Selected parts of the concepts are implemented in a simulation environment and finally analysed in a simulation-based case study.

Inhaltsverzeichnis

1 Einleitung

In modernen Fahrzeugen arbeiten eine Vielzahl von Sensoren und Steuergeräten (Electronic Control Unit (ECU)). Ihre Funktionen verbessern die Leistungsfähigkeit, den Komfort und die Sicherheit. Des weiteren ermöglichen sie fortgeschrittene Fahrerassistenzsysteme und in Zukunft das autonome Fahren.

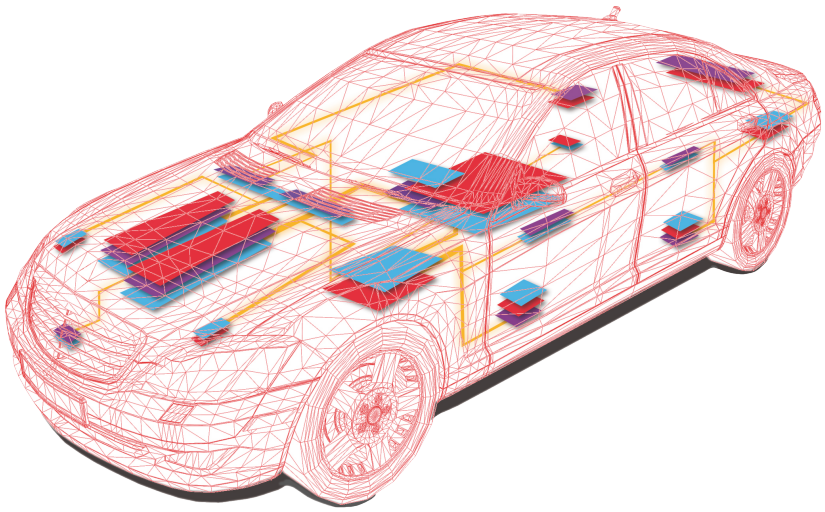

Abbildung 1.1: Verteilte ECUs in einem modernen Fahrzeug (Quelle: CoRE Working Group (a))

Diese ECUs und Sensoren sind über das gesamte Fahrzeug verteilt und miteinander verbunden (siehe Abbildung 1.1). Einzelne ECUs erfüllen jeweils spezifische Teilaufgaben und müssen miteinander kommunizieren, um Funktionen bereit zu stellen. Dies führt zu komplexen Kommunikationsarchitekturen. Aktuell werden hauptsächlich proprietäre Bustechnologien wie Controller Area Network (CAN), FlexRay, Media Oriented System Transport (MOST) und Local Interconnect Network (LIN) eingesetzt, um die unterschiedlichen Anforderungen der Kom-

munikationskanäle zu gewährleisten. Ethernet dagegen wird nur für einzelne Verbindungen eingesetzt (vgl. Matheus und Königseder (2015)).

Mit der Einführung von Ethernet im Backbone[1] (siehe. eeDesignIt) kann die Kommunikationsarchitektur in zukünftigen Fahrzeugen simpel und effizient gestaltet werden (vgl. Buckl u. a. (2012)). Dabei sind die Verfügbarkeit der Netzwerktechnologie, die hohen Bandbreiten und die Eigenschaften der Ethernet-Topologie die Hauptvorteile gegenüber aktuellen Bussystemen. Demgegenüber steht, dass Ethernet keinen harten Anforderungen von Echtzeitkommunikation standhält.

Echtzeit-Ethernet-Protokolle erweitern die Funktionalität von Ethernet, um harte Echtzeitanforderungen zu erfüllen. In der Automatisierungstechnik von Industrieanlagen sind solche schon heute erfolgreich im Einsatz. Beispiele hierfür sind EtherCat, PROFINET, SERCOS-III und Ethernet POWERLINK (vgl. International Electrotechnical Commission (2014)). Relevante Kandidaten für das Automobil sind zum Beispiel Time-Triggered Ethernet (TTE), Audio Video Bridging (AVB) und Time-Sensitive Networking (TSN).

Durch sie kann in Ethernet-Netzwerken die Ankunft von Nachrichten bei der Ziel-ECU in einem bestimmten Zeitfenster garantiert werden. Damit kann die funktionssichere Kommunikation zwischen den Steuergeräten umgesetzt werden.

Auch die Kommunikation mit externen Quellen wird das Auto der Zukunft ausmachen. Beispiele dafür sind zum einen der Informationsaustausch zwischen Fahrzeugen (Car-to-Car) und zum anderen zwischen Fahrzeug und Umgebung (Car-to-X). Des weiteren werden diese Fahrzeuge an Clouddienste angebunden.

Diese externen Informationen können im Fahrzeug verarbeitet werden, um weitere Funktionen zu ermöglichen. Ein mögliches Beispiel ist die Motorsteuerung oder die Schaltautomatik an die Verkehrslage anzupassen. Dadurch wird jedoch das interne Kommunikationsnetz des Automobils nach außen geöffnet und die Verletzbarkeit des Systems „Auto“ massiv erhöht.

Daher muss die Informationssicherheit ein grundlegendes Ziel bei der Entwicklung von zukünftigen Kommunikationstechnologien im Fahrzeug sein. Es ist wichtig, dass die Funktionssicherheit der Kommunikation zwischen den ECUs geschützt ist und nicht durch Angriffe von außen außer Kraft gesetzt werden kann.

In aktuellen Fahrzeugen sind kritische Teile der Fahrzeuginfrastruktur angreifbar. Je nachdem welches Subsystem angegriffen wird, kann dies sogar die Sicherheit der Personen im Fahrzeug gefährden.

Beispiele hierfür sind zum Beispiel der kabellose Hack eines Jeeps (siehe WIRED) und eines Teslas (siehe The Verge). Selbst wenn in diesen konkreten Fällen eine Verschlüsselung die

[1] Backbone (deutsch: Rückgrat, Basisnetz) bezeichnet den zentralen Kernbereich eines Kommunikationsnetzes.

gezielte Manipulation verhindert hätte, ist der Ausfall von Bremsen etc. durch das simple Fluten des Bordnetzes mit Nachrichten vorstellbar.

In dieser Arbeit wird die Informationssicherheit der Echtzeit-Ethernet-Protokolle TTE, AVB und TSN analysiert. Dafür werden die Protokolle einer Sicherheitsanalyse unterzogen. Für die gefundenen Schwachstellen werden die Sicherheitsanforderungen gesammelt und das Risiko bewertet.

Des weiteren werden Schutzkonzepte erarbeitet, die die gefundenen Schwachstellen schließen und Vorkehrungen treffen, um Informationssicherheit und Funktionssicherheit des Kommunikationsnetzes auch im Fall eines gezielten Angriffs zu erhalten.

Teile der Schutzkonzepte sollen in einer qualitativen Simulation getestet und analysiert werden. Mit Hilfe der Ergebnisse werden die Auswirkungen der Schutzkonzepte auf das Kommunikationsnetz diskutiert.

Die Arbeit ist wie folgt gegliedert: Kapitel 2 vermittelt die Grundlagen zum Verständnis der weiteren Arbeit. Dazu gehören die Funktion der ausgewählten Echtzeit-Ethernet-Protokolle, die Begrifflichkeiten der Informationssicherheit und die eingesetzte Simulationsumgebung. Darauf folgt Kapitel 3. In diesem werden Arbeiten vorgestellt, die die Informationssicherheit in aktuellen Fahrzeugen untersuchen. So wird ein Überblick über den Istzustand bei aktuellen Fahrzeugen gegeben. In Kapitel 4 wird dann die Sicherheitsanalyse der ausgewählten Echtzeit-Ethernet-Protokolle beschrieben. Hier werden Schwachstellen und Sicherheitsanforderungen identifiziert und deren Risiko bewertet. Kapitel 5 sammelt Konzepte zum Schutz und stellt neue Ideen vor, mit denen die zuvor erarbeiteten Sicherheitsanforderungen erfüllt werden. Im darauf folgendem Kapitel 6 wird der Entwurf, die Umsetzung und die Qualitätssicherung der erweiterten Simulationsumgebung vorgestellt. Danach beschreibt Kapitel 7 eine simulationsbasierte Fallstudie und die Analyse der Simulationsergebnisse. Hier werden die Auswirkungen einiger Schutzkonzepte auf das Kommunikationsnetzwerk im Fahrzeug diskutiert. Zum Ende folgt in Kapitel 8 Ausblick und Fazit.

2 Grundlagen

In diesem Kapitel werden die Grundlagen zum weiteren Verständnis der Arbeit vermittelt. Dazu gehört die Erläuterung der Funktion der ausgewählten Echtzeit-Ethernet-Protokolle TTE, AVB und TSN (s. Abschnitt 2.1). Des weiteren werden Begrifflichkeiten der Informationssicherheit erklärt (s. Abschnitt 2.2) und ein Überblick die eingesetzte Simulationsumgebung gegeben (s. Abschnitt 2.3).

2.1 Echtzeit-Ethernet-Protokolle

Die zeitlichen Anforderungen an die Kommunikation in einem Fahrzeug sind verschieden. Das Infotainment[1] hat zum Beispiel keine Echtzeitanforderungen an das Netzwerk, soll dieses aber trotzdem nutzen. Systeme wie Bremsen, Motorsteuerung und Rückfahrkamera dagegen haben Echtzeitanforderungen. Ein verzögertes Bremsen als Folge einer Verzögerung in der Kommunikation an die verantwortliche ECU ist nicht hinnehmbar.

Standard Ethernet nach IEEE 802.3 (vgl. Institute of Electrical and Electronics Engineers (2008b)) gibt keine Garantien auf die Qualität der Kommunikation (Quality of Service (QoS)) und erfüllt daher keine Echtzeitanforderungen. Echtzeit-Ethernet-Protokolle erweitern Ethernet mit Echtzeitkomponenten die QoS-Garantien mitbringen und so ein definiertes und vorhersagbares Kommunikationsverhalten umsetzen.

Bei den meisten Protokollen bedeutet dies, dass weitere Nachrichtenklassen eingeführt werden, die gesondert behandelt werden. Dabei wird der Standard Ethernet-Verkehr unter dem Namen Best-Effort[2] (BE) beibehalten. Auf diese Weise müssen Anwendungen ohne Echtzeitanforderungen keine Echtzeitressourcen belegen.

Für die Bewertung der Echtzeitkommunikation werden zwei zentrale Metriken verwendet. Die Erste ist die Latenz (T_{Latenz}). Diese beschreibt die zeitliche Verzögerung einer Nachricht von einem Sender- zu einem Empfängermesspunkt. Die Zweite ist der Jitter (T_{Jitter}). Durch

[1] Infotainment (zusammengesetzt aus dem englischen information und entertainment) fasst alle Anwendungen und Geräte zusammen, die zur Information oder Unterhaltung der Fahrgäste verbaut sind.

[2] Best-Effort (deutsch: größte Anstrengung) bezeichnet die geringste Übertragungspriorität in Kommunikationsnetzwerken. Es gibt keine Garantien für Erfolg und Dauer der Übertragung.

diesen wird die Differenz der maximalen und minimalen Latenz einer Menge von Nachrichten beschrieben.

Im folgendem werden drei für diese Arbeit relevante Echtzeit-Ethernet-Protokolle vorgestellt. Diese sind Time-Triggered Ethernet (2.1.1), Audio Video Bridging (2.1.2) und Time-Sensitive Networking (2.1.3).

2.1.1 Time-Triggered Ethernet

TTE ist ein auf TDMA[3]-Technologien basierendes Echtzeit-Ethernet-Protokoll. Es ist von der Society of Automotive Engineers standardisiert in AS6802 (vgl. Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee (2011)). In diesem wird synchroner und asynchroner Netzwerkverkehr in drei verschiedenen Klassen umgesetzt.

- **Time-Triggered (TT)**-Verkehr ist die am höchsten priorisierte Nachrichtenklasse. Pakete aus dieser Klasse werden auf offline definierten Routen synchron weitergeleitet. Die Zeitpunkte für Versand und Empfang sind statisch durch die Konfiguration festgelegt. Daher müssen alle Netzwerkteilnehmer eine globale Zeit synchronisieren. Pakete, die außerhalb eines Empfangsfensters eintreffen oder für die in einem Gerät keine Konfiguration existiert, werden verworfen. Die Latenz ist durch die Konfiguration definiert, der Jitter liegt im Bereich von Nanosekunden.

- **Rate-Constrained (RC)**-Verkehr ist der darauf folgenden Priorität zugeordnet. Auch hier werden die Routen der Pakete offline konfiguriert. Anstatt von konkreten Zeitpunkten wird jedoch nur der relative Mindestabstand zwischen zwei Paketen statisch festgelegt (Bandwidth Allocation Gap (BAG)). Innerhalb der Klasse von RC-Nachrichten sind zusätzlich acht numerische Prioritäten definiert.

- **Best-Effort (BE)**-Verkehr bildet die niedrigste Priorität ab.

Darauf aufsetzend können nun verbreitete Protokolle aus höheren Open Systems Interconnection (OSI)-Modellschichten eingesetzt werden.

Die Abbildung 2.1 zeigt beispielhaft die Interaktion der drei Nachrichtenklassen mit diesen. Im Fall von TT- und RC-Verkehr muss auf der Vermittlungsschicht ein Service existieren, der die Umwandlung auf die jeweilige Nachrichtenklasse umsetzt. Im Fall von BE-Verkehr ist dies nicht notwendig. Ansonsten können Protokolle wie zum Beispiel das Internet Protocol

[3] Time Division Multiple Access (TDMA) ist ein Zeitmultiplexverfahren bei dem Daten von unterschiedlichen Sendern zu jeweils bestimmten Zeitabschnitten versendet werden.

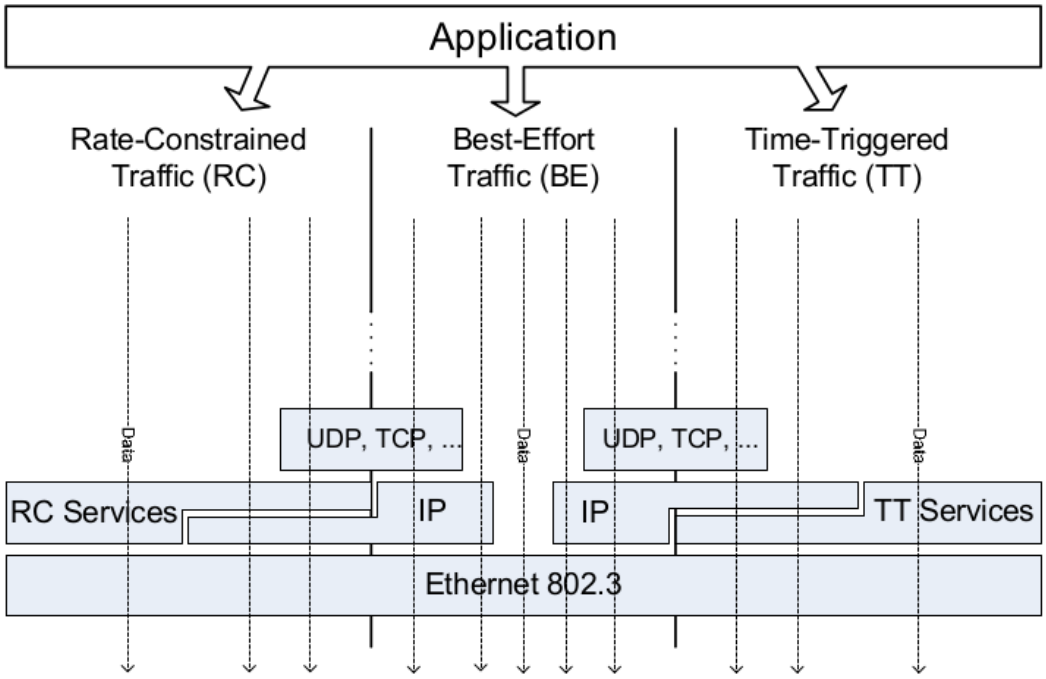

Abbildung 2.1: Interaktion von TTE mit verbreiteten Protokollen. (Quelle: Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee (2011))

(IP), User Datagram Protocol (UDP) und Transmission Control Protocol (TCP) wie allgemein bekannt eingesetzt werden.

2.1.2 Audio Video Bridging

Audio Video Bridging ist ein Protokoll, das vom Institute of Electrical and Electronics Engineers (IEEE) standardisiert wurde (vgl. Institute of Electrical and Electronics Engineers (2011)). Pakete werden asynchron über dynamische Routen innerhalb von definierten Nachrichtenströmen (Streams) versendet. Trotzdem kann garantiertes Zeitverhalten vorhergesagt werden. Das Protokoll basiert auf 802.1Q (vgl. Institute of Electrical and Electronics Engineers (2014)). Daher stehen acht Prioritäten für die Pakete zur Verfügung. Die beiden höchsten werden in AVB jedoch als gesonderte Klassen interpretiert. Diesen wird ein Class Measurement Interval (CMI) zugeordnet innerhalb dessen eine Bandbreite reserviert werden kann.

- **A**-Klasse-Verkehr ist Verkehr mit höchster Priorität. Das CMI entspricht 125 µs. Die maximale Latenz ist mit 2 ms über 7 Hops angegeben.
- **B**-Klasse-Verkehr beschreibt die nächste Prioritätsstufe. Das CMI entspricht 250 µs. Hier ist die maximale Latenz mit 50 ms über 7 Hops angegeben.
- **BE**-Verkehr bildet auch hier die niedrigste Priorität ab, wobei dieser über die restlichen sechs IEEE 802.1Q Prioritäten verteilt werden kann.

Die Routen für den Verkehr der Klassen A und B können dynamisch zur Laufzeit festgelegt werden. Dieses Verfahren nennt sich Stream Reservation Protocol (SRP) (vgl. Institute of Electrical and Electronics Engineers (2010)). Dafür macht ein Talker[4] die Eigenschaften seines Streams (Klasse, Paketperiode, Paketgröße) im gesamten Netzwerk bekannt. Ein Listener[5], der diesen Stream empfangen möchte gibt diesen Wunsch danach auch dem Netzwerk bekannt. Jeder Switch auf dem Weg zum Talker reserviert die erforderliche Bandbreite auf dem Port zurück zum Listener. Bei Ankunft einer Bekanntgabe am Talker stellt dieser den Stream bereit.

Damit an jedem Ausgangsport die reservierte Bandbreite für den Stream zur Verfügung steht, aber auch nicht überzogen werden kann, gibt es jeweils einen Credit Based Shaper (CBS) Algorithmus (vgl. Institute of Electrical and Electronics Engineers (2009)) pro Port für Klasse A und B. Dieser basiert auf dem Wert eines Credits. Wenn der Credit kleiner ist als 0, darf kein Paket der entsprechenden Klasse versendet werden, jedoch eines niederer Priorität. Ansonsten wird ein Paket der Klasse auf den Port gelegt. Die maximale Länge eines Pakets ist von der reservierten Bandbreite des zugehörigen Streams abhängig, sodass diese Bandbreite innerhalb des CMI eingehalten wird.

Es gibt zwei mögliche Gradienten mit denen der Credit manipuliert wird. Sie nennen sich $idleslope$ und $sendslope$. Zusammengesetzt werden diese aus der reservierten Bandbreite der Klasse auf dem Port (RB) und der maximalen Bandbreite des Ports (B). Die Definitionen sind in den Gleichungen 2.1 und 2.2 zu sehen.

$$idleslope = RB \tag{2.1}$$

$$sendslope = RB - B \tag{2.2}$$

Beim Versenden eines Paketes der zugehörigen Klasse wird der Credit für die Dauer des Sendens mit dem Gradienten $sendlope$ dekrementiert. Wenn der Wert des Credits negativ ist,

[4] Talker (deutsch: Sender, Sprecher) ist die Quelle eines AVB Nachrichtenstroms.
[5] Listener (deutsch: Empfänger, Zuhörer) ist die Senke eines AVB Nachrichtenstroms.

wird dieser mit dem Gradienten $idleslope$ inkrementiert bis er wieder 0 erreicht. In dem Fall, dass ein Paket der Klasse nicht versendet werden kann, weil bereits ein Paket mit niederer Priorität den Port blockiert, wird der Credit auch inkrementiert, wenn dieser größer oder gleich 0 ist. Wenn der Credit größer ist als 0 und kein Paket der Klasse im Puffer wartet, um versendet zu werden, wird der Credit auf 0 zurückgesetzt.

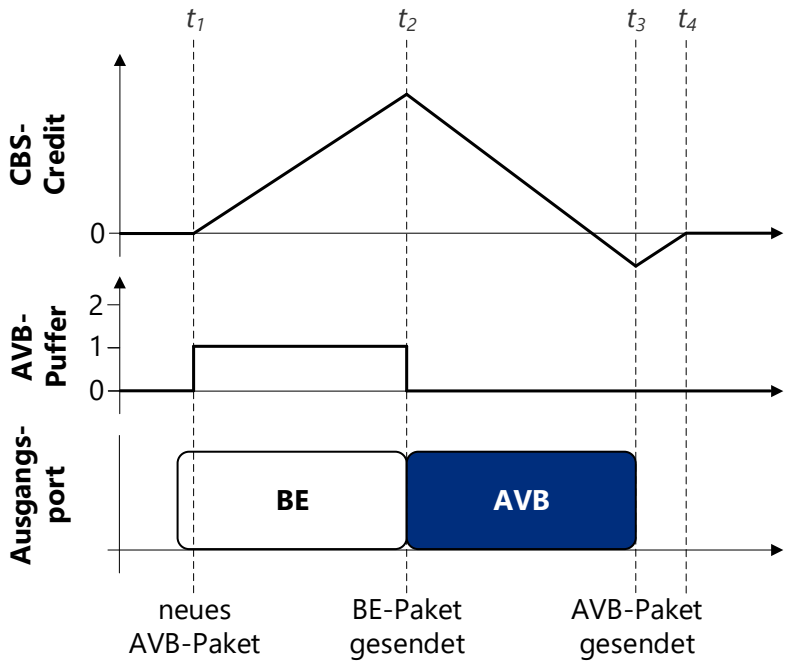

Abbildung 2.2: Credit Based Shaping Beispiel

Die Abbildung 2.2 zeigt ein Beispiel für die Funktionsweise eines CBS. Zu Beginn ist der Wert des Credits 0. Dann soll ein BE-Paket versendet werden. Da kein Paket der AVB-Klassen im Puffer ist, wird das BE-Paket auf die Leitung gelegt. Zum Zeitpunkt t_1 kommt nun ein AVB Paket im Puffer an. Da bereits das BE-Paket auf der Leitung liegt muss das AVB-Paket warten bis die Leitung frei ist. Um sich diesen Rückstand zu merken, wird der Credit für die Dauer der Blockade mit der Steigung $idleslope$ inkrementiert. Am Zeitpunkt t_2 ist die Leitung wieder frei und das anstehende AVB-Paket kann auf den Port gelegt werden. Für die Dauer des Sendevorgangs bis zum Zeitpunkt t_3 wird der Credit mit $sendslope$ dekrementiert. Da der Credit nun negativ ist, steigt er wieder bis auf 0 an. Zum Zeitpunkt t_4 erreicht der Credit 0 und das nächste AVB-Paket dürfte versendet werden.

Wenn nach der Übertragung eines AVB-Pakets der Credit größer oder gleich 0 ist, dürfen die nächsten AVB-Pakete sogar direkt aufeinanderfolgend versendet werden bis der Credit wieder kleiner 0 ist. Mit diesem Verhalten wird die durch die Blockade verlorene Bandbreite wieder aufgeholt.

2.1.3 Time-Sensitive Networking

Das TSN Echtzeit-Ethernet-Protokoll ist eine Sammlung von Standards, die speziell für den Einsatz in industriellen Kontrolleinrichtungen und Kommunikationsnetzwerken innerhalb von Fahrzeugen ausgelegt ist. Die Standardisierung wird in der IEEE 802.1 Time-Sensitive Networking Task Group (vgl. IEEE 802.1 TSN Task Group (a)) durchgeführt.

Im Protokoll werden die Paketauswahlmethoden von AVB und TDMA-Verkehr vergleichbar zu dem in TTE kombiniert. Dafür befindet sich an jedem Port eine Paketauswahlinstanz, die bestimmt wann welches Paket gesendet wird.

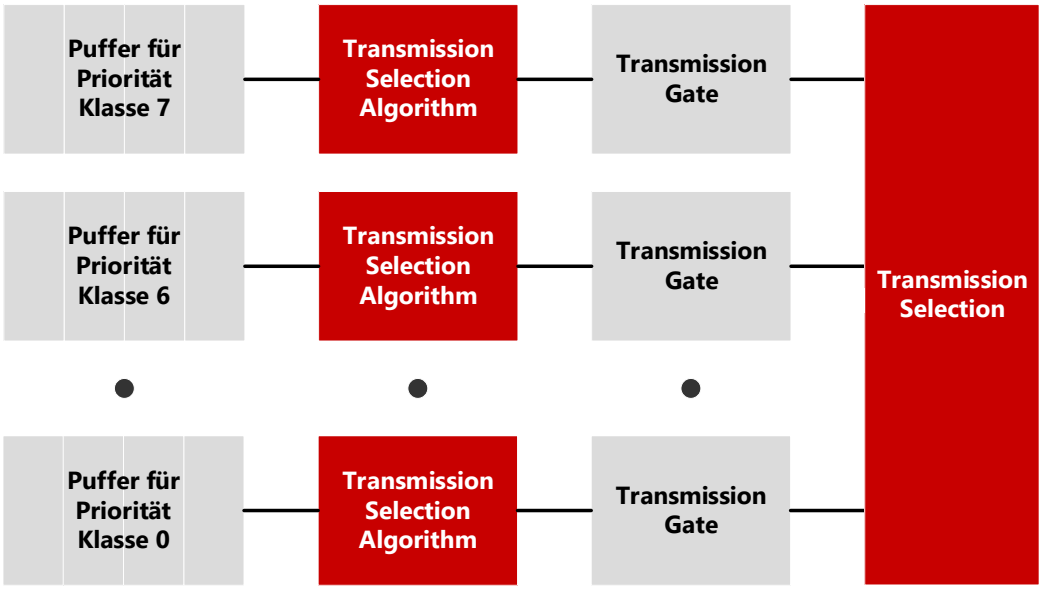

Abbildung 2.3: IEEE 802.1Qbv Paketauswahl (vgl. Institute of Electrical and Electronics Engineers (2016))

Die Abbildung 2.3 zeigt wie die Paketauswahl umgesetzt ist. Pakete können einer von acht Prioritäten zugeordnet werden. Diese reichen von 0 bis 7. Für jede Zuordnung enthält die Paketauswahl einen Puffer, einen „Transmission Selection Algorithm“ und ein „Transmission

Gate“. Wenn mehr als ein Pfad bis zur letzten „Transmission Selection“ offen ist, wird das Paket aus dem Pfad mit der höchsten Priorität entnommen und versendet.

Der „Transmission Selection Algorithm“ kann unterschiedliche Algorithmen zur Paketauswahl enthalten. Ein Beispiel dafür ist der CBS-Algorithmus, der im vorherigen Abschnitt 2.1.2 vorgestellt wurde. Weitere sind im Standard IEEE 802.1Qbv definiert (vgl. Institute of Electrical and Electronics Engineers (2016)) oder können im Spezialfall konstruiert werden.

Wenn der „Transmission Selection Algorithm“ das Versenden eines Pakets erlaubt, entscheidet als nächstes das „Transmission Gate“. Dieses hat einen der zwei möglichen Zuständen „OPEN “oder „CLOSED “. Ein Paket zu versenden ist nur erlaubt, wenn der Zustand „OPEN “aktiv ist. Der Zustandswechsel kann zeitgesteuert auf Basis einer vorherigen Offlinekonfiguration durchgeführt werden. So kann zum Beispiel TDMA-Verkehr umgesetzt werden, in dem man in einem bestimmten Zeitfenster nur einzelne Kanäle über das „Transmission Gate“ freischaltet.

Des weiteren kann in TSN eine entsprechende Eingangskontrolle an den Ports verwendet werden. Diese ist in dem Standard IEEE 802.1Qci definiert (vgl. Institute of Electrical and Electronics Engineers (2017)).

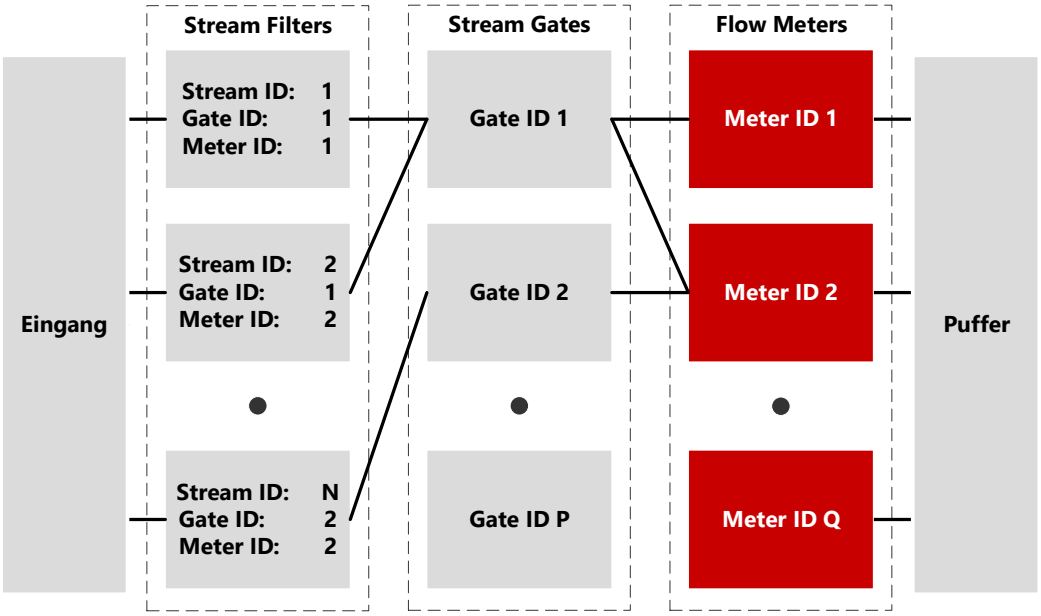

Abbildung 2.4: IEEE 802.1Qci Eingangskontrolle (vgl. Institute of Electrical and Electronics Engineers (2017))

Die Abbildung 2.4 skizziert die Eingangskontrolle. Eine eingehende Nachricht muss drei Ebenen durchlaufen bevor sie gepuffert wird. Dies sind die „Stream Filters“, „Stream Gates“ und „Flow Meters“. In der „Stream Filters“-Ebene wird der Nachricht, auf basis vorheriger Konfiguration, ein konkretes Gate und Meter in den darauffolgenden Ebenen zugewiesen. Die „Stream Gates“ haben, wie die „Transmission Gates“, einen der zwei Zustände „OPEN“ und „CLOSED“. Auch hier kann der Zustandswechsel zeitgesteuert auf Basis einer vorherigen Offlinekonfiguration durchgeführt werden. Eine Nachricht wird verworfen, wenn das zuständige Gate im Zustand „CLOSED “ist. Im anderen Fall ist die nächste Ebene für die Nachricht zuständig. Eine Meter-Instanz kann, ähnlich den „Transmission Selection“-Algorithmen, unterschiedliche Verfahren zum Filtern von Nachrichten umsetzen. Eine Nachricht, die vom zugewiesenen Meter akzeptiert ist, wird im Gerät für die Weiterverarbeitung gepuffert.

2.2 Informationssicherheit

In einem sicheren System beschreibt der Begriff „Sicherheit“ unterschiedliche Aspekte. Zum einen geht er auf die „Informationssicherheit“ (Englisch: security) ein. Diese beschreibt ein System, das keine unautorisierte Veränderung und Gewinnung von Informationen zulässt. Der andere Aspekt ist die „Funktionssicherheit“ (English: safety) und beschreibt die Übereinstimmung von Ist- und Sollzustand eines Systems. Bei einem informationssicheren System wird die Funktionssicherheit vorausgesetzt. Wobei sich Mängel in diesem Bereich auch auf die Funktionssicherheit des Systems auswirken können. Es gibt drei Basisanforderungen für die Informationssicherheit (vgl. Eckert (2014)):

- Informationsvertraulichkeit: Geheimhaltung der Datenobjekte. Informationen eines Systems sind nur mit entsprechender Autorisierung lesbar.
- Datenintegrität: Verhindern von unautorisierten Modifikationen an Datenobjekten. Aktive Angriffe, mit dem Ziel das Systemverhalten zu verändern, werden verhindert.
- Systemverfügbarkeit: Kein Performance-Verlust. Angriffe können die Leistungsfähigkeit des Systems nicht beeinflussen.

Datenobjekte sind zum Beispiel Dateien, Hauptspeicher, Cache oder auch Nachrichtenpakete einer Kommunikation. Der Schutz von Informationen ist durch den Schutz der Datenobjekte gewährleistet. Daher darf der Zugriff auf diese nur in Abhängigkeit von Zugriffsrechten erfolgen. Die Sicherheit des Systems wird nur erreicht, wenn alle Komponenten berücksichtigt werden.

2.3 Netzwerksimulation mit OMNeT++

Die in dieser Arbeit genutzte Simulationsumgebung basiert auf OMNeT++ (vgl. OpenSim Ltd. (b)). OMNeT++ ist eine IDE und C++ Simulationsbibliothek mit Fokus auf die eventbasierte Simulation. Sie ist besonders geeignet für die Simulation von Kommunikationsnetzwerken. Die Bibliothek Es ist modular und erweiterbar.

Erweitert wird diese Basis mit zwei weiteren Simulationsframeworks:

- **INET** (vgl. OpenSim Ltd. (a)) ist eine Framework für Omnet++, dass Standard Ethernet- und Internettechnologien umsetzt.
- **CoRE4INET** (vgl. CoRE Working Group (b)) ist ein Framework, dass in der Communication over Realtime Ethernet (CoRE) Arbeitsgruppe (vgl. CoRE Working Group (a)) entstanden ist. Es enthält Implementierungen von AVB und TTE mit deren Hilfe in der Vergangenheit schon TSN-Technologien simuliert wurden (vgl. Meyer u. a. (2013)).

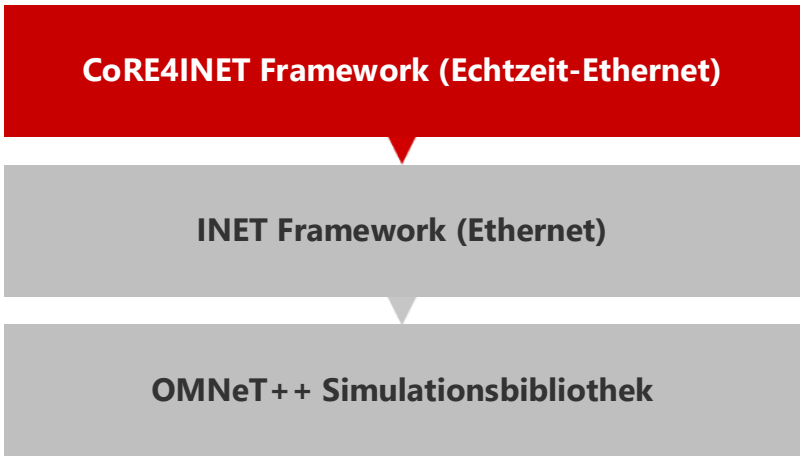

Abbildung 2.5: Aufbau der Simulationsumgebung

Die Abbildung 2.5 zeigt die Abhängigkeiten des Aufbaus der Simulationsumgebung. Das INET Framework ist von OMNeT++ abhängig. Das CoRE4INET nutzt vor allem die Standard Ethernet-Implementierung aus INET und erweitert diese, um Echtzeit-Ethernet-Protokolle umzusetzen.

Dieser Aufbau ist der gegebene Stand für die Simulationsbestrebungen in dieser Arbeit. In Kapitel 6 werden die in dieser Arbeit getätigten Erweiterungen vorgestellt.

3 Istanalyse

In diesem Kapitel werden Arbeiten vorgestellt, die den Stand der Informationssicherheit bei aktuellen Fahrzeugen untersuchen. Im Mittelpunkt steht die Frage, welchen Herausforderungen ein zukünftiger Ethernet-Backbone im Fahrzeug standhalten muss. Die Inhalte werden den Zielen dieser Arbeit gegenübergestellt.

Koscher et al. (vgl. Koscher u. a. (2010)) zeigen in wie weit sich ein heutiges Auto, das ohne Verschlüsselung betrieben wird, über die Diagnoseschnittstelle(ODB-II) angreifen lässt. Durch das Auslesen von CAN-Nachrichten der einzelnen Steuergeräte im Labor wurden genug Informationen extrahiert, um eine weitreichende Manipulation zu ermöglichen. Durch die physikalische Broadcast[1]-Eigenschaft von Bussystemen lassen sich solche Informationen jedoch auch direkt am Fahrzeug auslesen. Die Informationsvertraulichkeit ist an dieser Stelle nicht mehr gewährleistet. Im nächsten Schritt wird dann ein Laptop an den ODB-II Port angeschlossen. Von diesem lassen sich, vor und während der Fahrt, Angriffe auf das Fahrzeugnetz durchführen. Die Folge ist ein nahezu Vollzugriff auf die Systeme des Autos. Diese Kontrolle reicht von Radio über Tacho, einzelnen Bremsen und der Verriegelung bis hin zu Teilen der Motorsteuerung. In einem großen Bereich der Kommunikation ist die Datenintegrität verloren gegangen.

Auch wenn die Manipulation nicht möglich wäre kann mit Denial of Service (DoS)-Attacken die Kommunikation von CAN-Komponenten unterbunden werden. Auch das kann zu schwerwiegenden Funktionsausfällen führen. Spätestens hier ist auch die Systemverfügbarkeit nicht mehr gewährleistet.

Um diese detaillierte Analyse und Manipulation im Echtzeit-Ethernet-Bordnetz der Zukunft zu unterbinden, müssen Technologien zur Verschlüsselung der Kommunikation eingesetzt werden. Dies stellt die Informationsvertraulichkeit wieder her und macht eine gezielte Manipulation unmöglich. Hinzu kommt, dass Mechanismen zur Schließung von Schwachstellen für DoS-Angriffe vorhanden sein müssen, um die Systemverfügbarkeit zu gewährleisten.

[1] Broadcast (deutsch: Rundfunk, Rundruf, Rundspruch) bezeichnet das Senden eines Paketes von einem an alle Teilnehmer eines Netzwerks.

Technologien zur verschlüsselten Übertragung im Ethernet sind weit verbreitet. In einem Bordnetz auf Basis von Ethernet wird die Verschlüsselung in den meisten Fällen auf höheren OSI-Schichten stattfinden.

Um einen Angriff auf das Bordnetz durchzuführen, muss der Angreifer in das Bordnetz vordringen. Dafür kann eine Schnittstelle genutzt, ein Gerät hinzugefügt, ein vorhandenes Steuergerät stimuliert oder ein vorhandenes Steuergerät manipuliert werden.

Checkoway et al. (vgl. Checkoway u. a. (2011)) untersuchen in ihrer Arbeit vor allem über welche Schnittstellen ein Fahrzeugbordnetz angegriffen werden kann. Hierfür werden die unterschiedlichen I/Os (s. Abbildung 3.1) in drei Kategorien unterteilt:

- **Indirekter physikalischer Zugriff:** Hierzu gehören zum Beispiel die ODB-II Diagnoseschnittstelle und die Eingänge des Unterhaltungssystems (CD, USB, SD-Karte).
- **Kabelloser Zugriff über kurze Distanzen:** Beispiele hierfür sind Bluetooth, WiFi, Remote-Keyless-Entry und die kabellose Überwachung des Reifendrucks.
- **Kabelloser Zugriff über weite Distanzen:** GPS, Digital Radio und der Traffic Message Channel (TMC) sind in diesem Bereich Beispiele für Informationen, die an alle Fahrzeuge verteilt werden. Zusätzlich besitzen moderne Autos aber auch adressierbare Zugriffsmöglichkeiten über den Mobilfunk.

Unter Zuhilfenahme von Reverse-Engineering und Debugging gelingt es den Autoren in allen drei Bereichen, Zugriff auf das Bordnetz des Autos zu erhalten und über dieses zu kommunizieren. Dies wird durch spezielle WMA-Dateien auf einer CD, Lücken im Bluetooth-Stack der Freisprechanlage oder einer Mobilfunkverbindung möglich. Die Kontrolle über ein Fahrzeug lässt sich also, je nach verbauten Technologien, aus beliebiger Entfernung erlangen.

Durch die Einführung immer neuer Technologien zur Verbesserung von Komfort und Funktionssicherheit, wird die mögliche Anzahl an Schwachstellen an den Endgeräten immer weiter erhöht. Ein Echtzeit-Ethernet-Protokoll muss robust genug gesichert sein, um zu verhindern, dass die Kommunikation von Teilnehmern durch Angriffe, abseits der kompromittierten ECUs, unterbunden wird.

Aus diesen Arbeiten resultiert, dass der Zugriff auf die Kommunikationsinfrastruktur in aktuellen Fahrzeugen möglich ist und gefährliche Auswirkungen für Mensch und Maschine mit sich bringt. Die einfache Verschlüsselung reicht nicht aus um die Informationssicherheit herzustellen. Die Auswirkungen von DoS-Angriffen müssen unterbunden werden, um ein Fahrzeug und dessen Passagiere zu schützen.

Die nächste Arbeit (vgl. Henniger u. a. (2009)) setzt eine systematische Vorgehensweise zur Sicherheitsanalyse auf einer aktuellen Fahrzeuginfrastruktur ein. Das beobachtete System

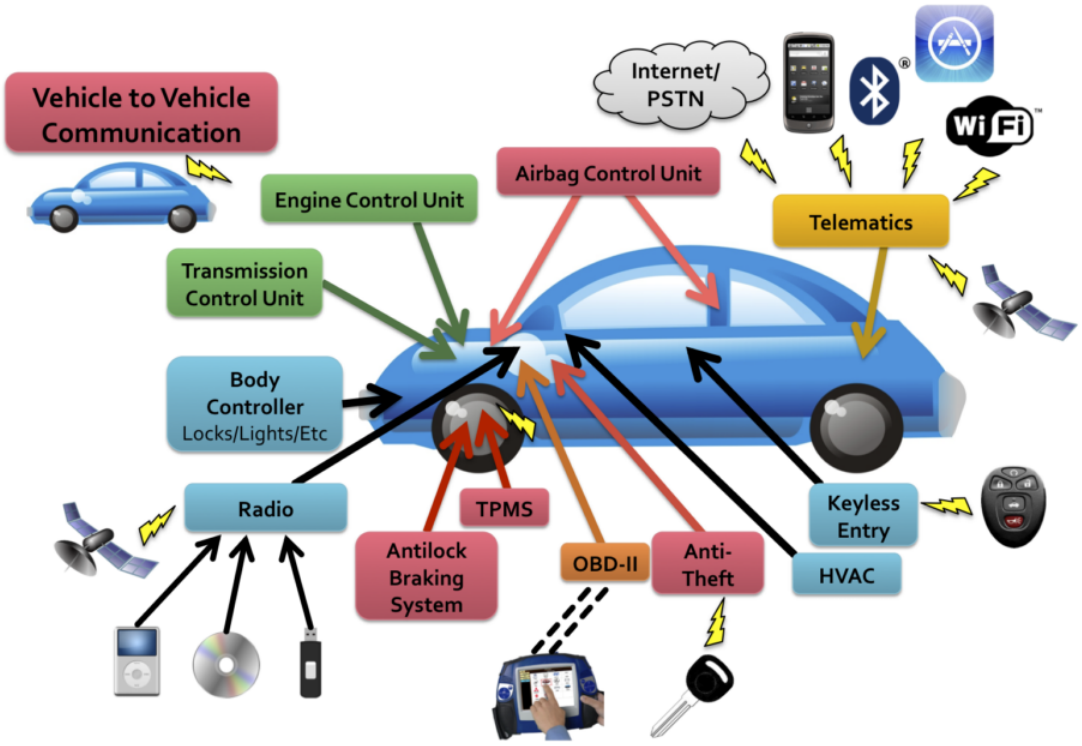

Abbildung 3.1: Angriffsfläche eines Autos (Quelle: Checkoway u. a. (2011))

von Henniger et al. ist ein Fahrzeug mit unterschiedlichsten ECUs und Schnittstellen. Diese sind über ein heterogenes Bussystem aus CAN- und FlexRay-Bussen miteinander verbunden. Die ECUs sind für Aufgaben, von Motorsteuerung bis hin zu Infotainment, verantwortlich. Die Schnittstellen sind zum Beispiel USB, Bluetooth, GPS und UMTS. Auf dieser Basis wird ein Prozess vorgestellt, der die Entwicklung von Sicherheitsanforderungen eines Bordnetzes ermöglicht. Dieser Prozess besteht aus drei Schritten. Zunächst werden Bedrohungen identifiziert. Um dies zu erreichen werden Bedrohungsbäume (vgl. Schneier (1999)) eingesetzt. Dabei ist die Wurzel das Ziel eines Angriffs und Blätter repräsentieren Subziele die das übergeordnete Ziel möglich machen können. Alle diese Ziele für Angreifer sind Bedrohungen für das System.

Die Abbildung 3.2 zeigt ein Beispiel eines Bedrohungsbaums. Das Ziel in diesem Beispiel ist der Diebstahl eines Fahrzeugs. Um dies zu bewerkstelligen muss entweder der Schlüssel gestohlen oder das Schloss geknackt werden. Um wiederum das Schloss zu knacken, braucht es Wissen über das Schloss und das richtige Werkzeug. Weitere Anforderungen, um den Schlüssel zu stehlen, das Werkzeug zu beschaffen und das Wissen zu erwerben sind im Beispiel nicht aufgeführt. Der Baum kann jedoch beliebig verfeinert werden.

Abbildung 3.2: Beispiel eines Bedrohungsbaums

Im zweiten Schritt werden in der Arbeit von Henniger et al. (vgl. Henniger u. a. (2009)) die Sicherheitsanforderungen, die die entdeckten Bedrohungen bekämpfen, identifiziert. Hier werden zum einen Anforderungen für die Sicherheit an den Systemgrenzen erarbeitet. Diese Systemgrenzen beschreiben hier die Kommunikation von Fahrzeugen untereinander. Zusammen mit den Anforderungen für die detaillierte Funktionalität der Onboard-Systeme (Sensoren, Steuergeräte) wird die Liste der Sicherheitsanforderungen vervollständigt. Um die wichtigsten Sicherheitsanforderungen zu erkennen werden diese im letzten Schritt priorisiert. Dazu werden die Bedrohungen mit Eintrittswahrscheinlichkeit, Kontrollierbarkeit und Schärfegrad klassifiziert. Der Schärfegrad wird mehrdimensional beschrieben. Auf diese Weise kann menschlicher und finanzieller Schaden eingeordnet werden ohne diese gegeneinander gewichten zu müssen. Aus diesen drei Faktoren wird das Risiko berechnet. Auf dessen Basis können dann die dazugehörigen Sicherheitsanforderungen priorisiert werden.

Diese Vorgehensweise soll für die Analyse des Echtzeit-Protokolle eingesetzt werden. Auch die Beschreibung eines mehrdimensionalen Schadens mit daraus resultierendem Risiko ist sinnvoll. Es ermöglicht die Sicherheitslösungen unter unterschiedlichen Gesichtspunkten zu betrachten.

4 Analyse der Informationssicherheit

Das Ziel dieses Kapitels ist die Sicherheitsanalyse der in Abschnitt 2.1 vorgestellten Echtzeit-Ethernet-Protokolle TTE, AVB und TSN. Das vorherige Kapitel 3 hat gezeigt, dass der Zugriff auf unterschiedlichste Komponenten im Fahrzeug möglich ist. Diese Komponenten sind im zukünftigen Fahrzeug auch Teilnehmer einer Echtzeit-Ethernet-Kommunikation und können im nächsten Schritt Schwachstellen in dieser ausnutzen.

Daher liegt der Fokus dieser Analyse auf der Systemverfügbarkeit der Echtzeitkommunikation auf der Sicherungsschicht[1]. Diese hat einen direkten Einfluss auf die Funktionssicherheit des Fahrzeugs. Es wird eine priorisierte Liste an Anforderungen zum Schließen dieser Schwachstellen ermittelt.

Dafür werden im ersten Schritt in Abschnitt 4.1 Schwachstellen identifiziert, die die Ethernet-Kommunikation durch einen Angriff gefährden können. Im nächsten Abschnitt 4.2 werden die Anforderungen ermittelt, die durch eine Lösung zum Schließen der Schwachstellen eingehalten werden müssen. Am Schluss wird in Abschnitt 4.3 das Risiko der Schwachstellen ermittelt und damit die Priorität der Anforderungen festgelegt.

Das Risiko wird als eine Kombination aus Schaden und Eintrittswahrscheinlichkeit definiert. Dies kann zum Beispiel eine Matrix oder die Multiplikation von Schaden und Eintrittswahrscheinlichkeit (vgl. Deutsches Institut für Normung (2010) und Bundesamt für Sicherheit in der Informationstechnik (2017)) sein. In dieser Arbeit wird das Risiko qualitativ bewertet. Somit werden Eintrittswahrscheinlichkeits- und Schadensklassen definiert und den Schwachstellen zugeordnet. Im letzten Schritt wird damit das Risiko ermittelt.

4.1 Identifikation der Schwachstellen

Ziel dieses Abschnitts ist die Auflistung entdeckter Schwachstellen in den vorgestellten Echtzeit-Ethernet-Protokollen.

Die entdeckten Schwachstellen werden in diesem Abschnitt aufgelistet und definierten Eintrittswahrscheinlichkeits- und Schadensklassen zugeordnet.

[1] Sicherungsschicht (englisch: Data Link Layer) beschreibt die 2. Schicht des OSI-Modells. Diese Schicht regelt den Zugriff auf das Übertragungsmedium.

Tabelle 4.1: Definition der Eintrittswahrscheinlichkeitsklassen (EK)

Eintrittswahrscheinlichkeitsklassen (EK)	**Umsetzung des Angriffs**
0	Nicht durchführbar
1	Nur mit geheimen Informationen durchführbar
2	Nur mit speziellem Fachwissen durchführbar
3	Nur mit zusätzlichen Mitteln durchführbar
4	Jederzeit durchführbar

Tabelle 4.2: Definition der Schadensklassen ($\vec{SK}$)

Schaden	**Auswirkungen der Schwachstelle**		
Schadensklasse (SK)	**Teilnehmer (SK_T)**	**Kommunikation (SK_K)**	**Fahrzeug (SK_F)**
0	Keine Auswirkungen	Keine Auswirkungen	Keine Auswirkungen
1	Erhöhte Belastung	Einzelne Links beeinträchtigt	Funktion fällt aus
2	Belastung höher als spezifiziert	Multiple Links beeinträchtigt	Multiple Funktionen fallen aus
3	Teilnahme teilweise unmöglich	Einzelner Link fällt aus	Sicherheitsrelevante Funktion fällt aus
4	Keine Teilnahme möglich	Multiple Links fallen aus	Multiple sicherheitsrelevante Funktionen fallen aus

Die Tabelle 4.1 zeigt die Definition der Klassen für die Eintrittswahrscheinlichkeit. Die Eintrittswahrscheinlichkeit wird hier über die Durchführbarkeit definiert. Wie vom BSI empfohlen werden 5 Klassen definiert (vgl. Bundesamt für Sicherheit in der Informationstechnik (2017)). Die erste „Nicht durchführbar“ entspricht einer Eintrittswahrscheinlichkeit von 0% und die letzte „Jederzeit durchführbar“ ist die maximale Eintrittswahrscheinlichkeit. Die weitere drei Klassen ordnen sich dazwischen ein. Damit lässt sich die Eintrittswahrscheinlichkeit einer Schwachstelle qualitativ einschätzen. Diese Wahrscheinlichkeitsklassen werden mit einem eindimensionalen Wert von 0 bis 4 quantifiziert (vgl. Henniger u. a. (2009).

Die Tabelle 4.2 zeigt die Definition der Klassen für den Schaden. Der Schaden wird hier mehrdimensional beschrieben. Dabei werden die Auswirkungen einer Schwachstelle hinsichtlich

des einzelnen Netzwerkteilnehmers, der Kommunikation und des Gesamtsystems Fahrzeug bewertet. Auch hier gibt es jeweils fünf Klassen, die über einen Wert von 0 bis 4 quantifiziert werden (vgl. Bundesamt für Sicherheit in der Informationstechnik (2017)). Im Ergebnis entsteht so für jede Schwachstelle ein dreidimensionaler Schadensvektor (vgl. Henniger u. a. (2009).

Die Bezeichnung der Schwachstellen erfolgt nach System. Der erste Teil vor dem Punkt entspricht der Protokollabkürzung und der zweite nach dem Punkt nummeriert die jeweilige Schwachstelle. Zum Beispiel trägt die erste entdeckte Schwachstelle in AVB die Bezeichnung *#AVB.1*.

4.1.1 *#TTE.1* - Löschen von TDMA Nachrichten

Bei TTE ist die Präzision der Nachrichten in der Klasse TT abhängig von einer global synchronisierten Zeit. Nachrichten die außerhalb eines definierten Zeitfensters an einem Switch oder Endgerät eintreffen werden verworfen (siehe Abschnitt 2.1.1). Durch die Manipulation der Synchronisation kann das Verwerfen von Nachrichten provoziert werden.

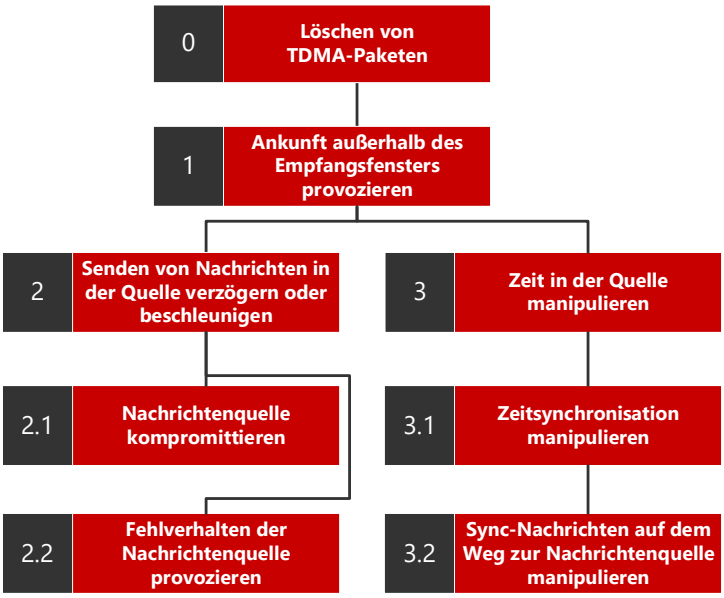

Abbildung 4.1: Bedrohungsbaum der Schwachstelle #TTE.1

Die Abbildung 4.1 zeigt den Bedrohungsbaum zu dieser Schwachstelle. Das Ziel des Angriffs ist das Löschen von TDMA-Paketen im Netzwerk. Dieses Ziel ist die Wurzel des Bedrohungsbaums (0).

Um dies zu erreichen muss die Ankunft außerhalb des definierten Zeitfensters provoziert werden (1). Hierfür kann das Senden verzögert oder beschleunigt werden (2). Ein weiterer Weg ist die Manipulation der Zeit (3).

Ersteres ist durch Kompromittierung der Nachrichtenquelle (2.1) oder durch das provozieren von Fehlverhalten (2.1) der Quelle möglich.

Über den zweiten Pfad (3) lässt ich das Löschen von Nachrichten jedoch auch ohne Zugriff auf die Quelle durchsetzen. Dafür muss der Zeitsynchronisationsmechanismus manipuliert werden (3.1). Dies ist durch die Manipulation der Synchronisationsnachrichten auf dem Weg zur Quelle umsetzbar (3.2).

Daraus folgt, dass der Angriff Kenntnis über das Synchronisationsprotokoll erfordert. Darum wird die Eintrittswahrscheinlichkeit dieser Schwachstelle mit „Nur mit speziellem Fachwissen durchführbar“ klassifiziert.
⇒ **Eintrittswahrscheinlichkeitsklasse: 2**

Nur die TT-Pakete sind von dem Verlust der Synchronisation betroffen. Dafür ist die Teilnahme am Netzwerkverkehr über die TT-Klasse nicht mehr möglich. Darum wird die Schadensklasse in der Dimension Teilnehmer mit „Teilnahme teilweise unmöglich“ klassifiziert.
⇒ **Schadensklasse Teilnehmer: 3**

Je nach Art des Angriffs kann die Synchronisation über ein oder mehrere Links ausfallen. Wieder ist dort jedoch nur der TT-Verkehr beeinflusst. Daraus folgt die Klassifizierung „Multiple Links beeinträchtigt“ für die Dimension Kommunikation.
⇒ **Schadensklasse Kommunikation: 2**

Da die TT-Klasse die besten QoS-Garantien bietet wird vor allem sehr zeitkritischer und sicherheitsrelevanter Verkehr der garantiert am Ziel ankommen muss über diese Klasse versendet. Dieser Verkehr kann zum Beispiel die Lenkung, Bremsen oder Motorsteuerung betreffen. Durch die Manipulation der Zeit wird das Prinzip der TT-Klasse-Nachrichten ausgehebelt. Mehrere Funktionen können davon betroffen sein. Darum wird der Schaden in der Dimension Fahrzeug mit „Multiple sicherheitsrelevante Funktionen fallen aus“ klassifiziert.
⇒ **Schadensklasse Fahrzeug: 4**

4.1.2 *#AVB.1* - Verzögern oder Löschen von Stream-Nachrichten

In AVB ist der CBS-Algorithmus am Ausgang für die Einhaltung der reservierten Bandbreite für Streams, die über die Nachrichtenklassen A und B transportiert werden, zuständig (s. Abschnitt 2.1.2). Durch die Kompromittierung einer Quelle kann in dieser der CBS übergangen werden

und beliebig viel Verkehr kann ins Netzwerk gelangen. Dieser DoS-Angriff führt dann zur Verdrängung von Verkehr anderer Quellen.

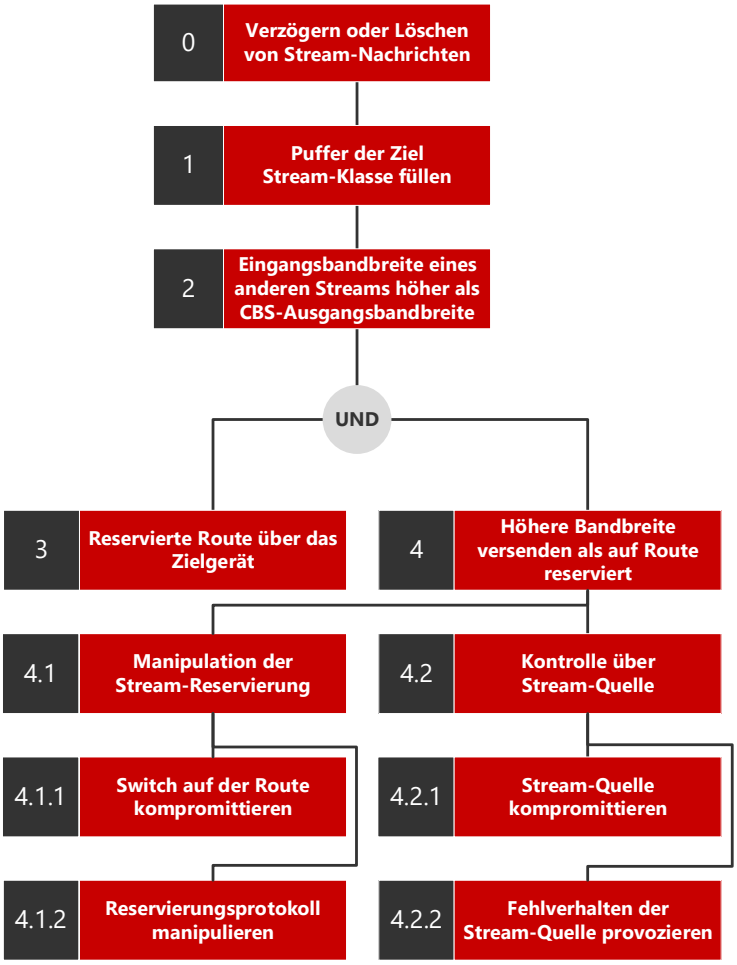

Abbildung 4.2: Bedrohungsbaum der Schwachstelle #AVB.1

Die Abbildung 4.2 zeigt den Bedrohungsbaum für diese Schwachstelle. Das Angriffsziel ist das Verzögern oder das Löschen von Stream-Nachrichten (0).

Dafür muss irgendein Puffer auf der Route soweit gefüllt werden, dass eingehende Nachrichten zum Verzögern möglichst weit hinten anstehen oder verworfen werden weil der Puffer voll ist (1). Um den Puffer zu füllen muss die eingehende Bandbreite eines anderen Streams der selben Klasse, der über den selben Port weitergeleitet wird, höher sein als die reservierte Bandbreite (2). Da sich der CBS an der reservierten Bandbreite orientiert, kann der Puffer nicht

in vorgegebener Zeit abgearbeitet werden. So werden auch Nachrichten die nicht von der kompromittierten Quelle stammen verzögert oder verworfen.

Um einen Stream mit erhöhter Bandbreite auf den Zielpuffer zu senden, müssen zwei Bedingungen erfüllt sein. Zum einen muss eine Route über das Gerät und dessen spezifischen Port reserviert sein (3). Zum anderen muss die Quelle mit einer höheren Bandbreite senden als über diese Route reserviert wurde (4).

Ersteres wird durch das Standard AVB-Protokoll SRP zum Reservieren von Routen durchgeführt.

Für letzteres kann entweder die Reservierung für den Stream manipuliert werden (4.1), um die reservierte Bandbreite zu verringern oder die Kontrolle über die Stream-Quelle muss vorhanden sein (4.2), damit diese mit mehr Bandbreite sendet als hierfür reserviert wurde.

Zur Manipulation der Stream-Reservierung kann entweder ein Switch auf der Route kompromittiert (4.1.1) oder das Reservierungsprotokoll SRP manipuliert werden (4.1.2).

Um die Kontrolle über die Stream-Quelle herzustellen, muss die Stream-Quelle entweder kompromittiert (4.2.1) oder zu Fehlverhalten provoziert werden (4.2.2). Kapitel 3 hat gezeigt, dass die Kompromittierung unterschiedlichster Netzwerkendgeräte möglich ist.

Auch hier muss Wissen über das Protokoll vorhanden sein, um einen gezielten Angriff durchzuführen. Ein Schaden kann jedoch schon entstehen, wenn eine Quelle ungezielt mehr sendet als reserviert ist. Es muss jedoch vorher die Kontrolle über eine Quelle bestehen. Daher wird die Eintrittswahrscheinlichkeit mit „Nur mit zusätzlichen Mitteln durchführbar“ klassifiziert.
⇒ **Eintrittswahrscheinlichkeitsklasse: 3**

Wenn Pakete wegen vollen Puffern verworfen werden fällt die Teilnahme für die Quelle teilweise aus. Von der Quelle gesendete Nachrichten der betroffenen Klasse kommen nicht mehr an. Daraus folgt die Klassifikation „Teilnahme teilweise unmöglich“ für den Schaden hinsichtlich des Teilnehmers.
⇒ **Schadensklasse Teilnehmer: 3**

Durch das Füllen des Puffers sind mindestens die Links vom kompromittierten Gerät zum Switch mit dem konkreten Puffer und die Verbindung von der Quelle des Angriffsziels zum Switch mit dem konkreten Puffer betroffen. Grundsätzlich sind aber alle Streams der Klasse betroffen die über diesen Port weitergeleitet werden und der selben Nachrichtenklasse entsprechen. Anderer Verkehr mit niedrigerer Priorität wie zum Beispiel BE-Verkehr kann zusätzlich verdrängt werden. Verkehr mit höherer Priorität ist nicht beeinflusst. Darum ist die Dimension Kommunikation mit „Multiple Links beeinträchtigt “ klassifiziert.
⇒ **Schadensklasse Kommunikation: 2**

Alle Funktionen die auf Informationen angewiesen sind, die über den Stream der entsprechenden Klasse transportiert werden, und deren Nachrichten über den angegriffen Pfad laufen, sind betroffen. Der Schaden in der Dimension Fahrzeug wird deshalb mit „Multiple sicherheitsrelevante Funktionen fallen aus“ klassifiziert.
⇒ **Schadensklasse Fahrzeug: 4**

4.1.3 *#AVB.2* - Blockieren von Streams

Wenn die reservierte Bandbreite eines Streams durch Manipulation erhöht wird, kann die erfolgreiche Reservierung von anderen Streams blockiert werden. Da die Quelle das Senden von Nachrichten nicht initiiert, wird der Stream blockiert.

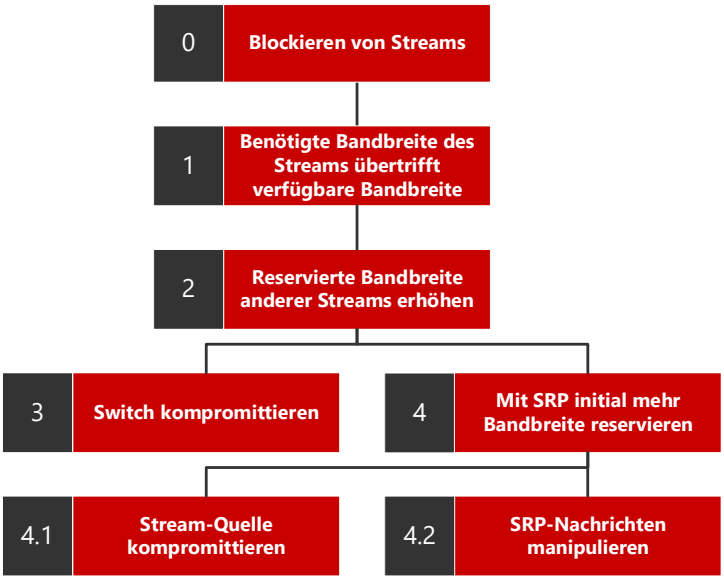

Abbildung 4.3: Bedrohungsbaum der Schwachstelle #AVB.2

Die Abbildung 4.3 zeigt den Bedrohungsbaum dieser Schwachstelle. Das Ziel ist das Blockieren von Streams (0).

Dafür muss die benötigte Bandbreite dieser Streams die verfügbare Bandbreite an einem Port auf der Route übertreffen (1). Dies kann mit der Erhöhung der reservierten Bandbreite anderer Streams auf dem Pfad ermöglicht werden (2).

Damit dieser Fall eintritt muss entweder der Switch kompromittiert (3) oder die initiale Beschreibung des Streams durch SRP manipuliert (4) werden.

Um die in den SRP-Nachrichten enthaltene Beschreibung des Streams zu manipulieren gibt es zwei Möglichkeiten. Die eine erfordert die Kompromittierung der Stream-Quelle, die die Beschreibung produziert (4.1). Die andere Möglichkeit ist die Manipulation der SRP-Pakete auf dem Weg zum Switch, der den Zielport enthält (4.2).

Für die Umsetzung dieses Angriffs muss spezielles Fachwissen über AVB-Streams und SRP vorliegen. Daher wird die Eintrittswahrscheinlichkeit mit „Nur mit speziellem Fachwissen durchführbar“ klassifiziert.
⇒ **Eintrittswahrscheinlichkeitsklasse: 2**

Betroffene Teilnehmer können die spezifischen Streams weder versenden noch empfangen. Anderer Nachrichtenverkehr ist nicht betroffen. Dies führt zu der Einstufung „Teilnahme teilweise unmöglich“ für die Dimension Teilnehmer des Schadens.
⇒ **Schadensklasse Teilnehmer: 3**

Durch das Blockieren der Reservierung an einem Zielport eines Switches können Streams von unterschiedlichen Quellen blockiert werden. Daraus folgt die Klassifizierung „Multiple Links beeinträchtigt“ für die Dimension Kommunikation des Schadens.
⇒ **Schadensklasse Kommunikation: 2**

Verkehr der über die Streams mit seinen QoS Garantien für Klasse A und B abgewickelt wird ist über multiple Links blockiert. Das kann zum Ausfall von multiplen sicherheitsrelevanten Funktionen führen. Daher wird die Dimension Fahrzeug mit „Multiple sicherheitsrelevante Funktionen fallen aus“ klassifiziert.
⇒ **Schadensklasse Fahrzeug: 4**

4.1.4 *#TSN.1* - Verzögern oder Löschen von BE Nachrichten

Für alle vorgestellten Protokolle gilt, dass durch DoS-Angriffe mit BE-Verkehr das Verwerfen oder Verzögern von BE-Paketen an den Switchen provoziert werden kann. Stellvertretend wird diese Schwachstelle hier daher TSN zugeordnet, welches die Funktionen von TTE und AVB vereint (s. Abschnitt 2.1.3).

Nach Kompromittierung einer BE-Nachrichtenquelle kann durch kontinuierliches Senden von Nachrichten sofort eine DoS-Attacke initiiert werden. Kapitel 3 hat gezeigt, dass Kompromittierung unterschiedlichster Netzwerkteilnehmer möglich ist. Daraus folgt die Eintrittswahrscheinlichkeitsklasse „Nur mit zusätzlichen Mitteln durchführbar “.
⇒ **Eintrittswahrscheinlichkeitsklasse: 3**

Die Teilnahme anderer Quellen mit BE-Verkehr ist eingeschränkt oder unmöglich. Die QoS-Garantien von höher priorisiertem Verkehr sind davon nicht beeinflusst. Da das SRP-Protokoll jedoch über BE-Verkehr transportiert wird, kann dadurch auch die Kommunikation über Streams ausfallen. Daraus folgt für die Schadensdimension Teilnehmer die Klassifizierung „Teilnahme teilweise unmöglich“.
⇒ **Schadensklasse Teilnehmer: 3**

Durch das Füllen der Puffer in dem an der kompromittierten Quelle angeschlossenen Switch, sind alle ausgehenden Ports von dem Angriff betroffen. Dies betrifft jedoch nur den BE-Verkehr. Darum wird der Schaden in der Dimension Kommunikation mit „Multiple Links beeinträchtigt“ klassifiziert.
⇒ **Schadensklasse Kommunikation: 2**

Nur die Funktionen ohne harte Echtzeitanforderungen werden über den BE-Verkehr abgewickelt. Jedoch kann die Kommunikation der Streams mit Bandbreitenreservierung durch das Blockieren des SRP-Verkehrs indirekt blockiert werden. Deshalb wird der Schaden in der Dimension Fahrzeug mit „Multiple sicherheitsrelevante Funktionen fallen aus“ klassifiziert.
⇒ **Schadensklasse Fahrzeug: 4**

4.2 Identifikation der Anforderungen

Das Ziel dieses Abschnitts ist zunächst die nicht priorisierte Auflistung von Anforderungen, die nötig sind um die entdeckten Schwachstellen aus Abschnitt 4.1 zu schließen.

Die Anforderungen werden den jeweiligen Protokollen und Schwachstellen zugeordnet und diskutiert.

Die Bezeichnung der Anforderungen setzt auf der Bezeichnung der Schwachstellen, aus dem vorherigen Abschnitt 4.1, auf. Der erste Teil trägt die Bezeichnung der jeweiligen Schwachstelle. Dieser wird nach einem Punkt die Nummerierung der Anforderungen angehängt. Zum Beispiel trägt die erste Anforderung zum Schließen der Schwachstelle *#AVB.1* die Bezeichnung *#AVB.1.1*.

4.2.1 *#TTE.1.1* - Sicherung des Synchronisationsprotokolls

Damit die Synchronisation nicht manipuliert werden kann, muss die Datenintegrität durch Authentifizierung der Synchronisationsteilnehmer hergestellt werden. Dadurch kann der Bedrohungsbaum (s. Abbildung 4.1) an der Stelle „Zeit in der Quelle manipulieren“ (3) beschnitten werden. Danach ist das Löschen von TDMA-Paketen nur noch über das Kompromittieren der Nachrichtenquelle oder durch die Provozierung von Fehlverhalten möglich. Um auch

die Informationsvertraulichkeit umzusetzen, müssen die Nachrichten zusätzlich verschlüsselt werden.

Der Fokus dieser Arbeit liegt vor allem bei speziellen Schutzmechanismen für Echtzeit-Ethernet-Protokolle. Aus diesem Grund wird im weiteren Verlauf nicht auf konkrete Umsetzungen von Authentifizierung und Verschlüsselung eingegangen.

4.2.2 *#AVB.1.1* - Begrenzung der eingehenden Stream-Nachrichten

Der Eingang von Nachrichten jedes Streams muss an jedem Eingangsport begrenzt werden, um ein Volllaufen der Puffer zu verhindern und damit die Systemverfügbarkeit sicherzustellen. Dadurch kann der Bedrohungsbaum (s. Abbildung 4.2) an der Stelle „Eingangsbandbreite eines anderen Streams höher als CBS Ausgangsbandbreite“ (2) beschnitten werden. Es ist dann nicht mehr möglich den Puffer von Klasse A und B weiter zu füllen als spezifiziert.

Diese Anforderung wird durch Mechanismen, die in dieser Arbeit vorgestellt werden, umgesetzt. Ziel ist ein spezieller Filter für die Streams der AVB-Klassen A und B.

4.2.3 *#AVB.1.2* - Aufholverhalten des CBS unbeeinflusst

In AVB erlaubt der CBS ein Aufholen von verpasster Bandbreitenkapazität (s. Abschnitt 2.1.2). Dadurch kann die eingehende Bandbreite kurzfristig höher sein, als für den Stream reserviert ist. Dieses valide Standardkonforme Verhalten muss weiterhin möglich sein, ohne das die Begrenzung am Eingang Nachrichten verwirft.

Diese Anforderung wird durch den selben Filter für AVB-Klassen A und B umgesetzt.

4.2.4 *#AVB.1.3* & *#AVB.2.1* - Sicherung des Reservierungsprotokolls

Wie bei der Sicherung des Synchronisationsprotokolls (s. Abschnitt 4.2.1) muss durch Authentifizierung die Datenintegrität der SRP-Nachirchten hergestellt werden. Zum einen kann dadurch der Bedrohungsbaum zur Schwachstelle *#AVB.1* (s. Abbildung 4.2) an der Stelle 4.1.2 beschnitten werden. Zum anderen kann der Bedrohungsbaum zur Schwachstelle *#AVB.2* (s. Abbildung 4.3) an der Stelle 4 beschnitten werden. Wenn auch die Informationsvertraulichkeit umgesetzt werden soll, müssen auch hier die Nachrichten verschlüsselt werden.

Wie zuvor in Abschnitt 4.2.1 beschrieben liegt der Fokus im weiteren Verlauf der Arbeit nicht auf der Vorstellung von konkreten Lösungen für die Authentifizierung und Verschlüsselung.

4.2.5 *#AVB.1.4 & #AVB.2.2* - Sicherung der Switche

Ein Switch ist eine kritische Komponente des Netzwerks und muss vor Kompromittierung geschützt sein. Dadurch kann die Stelle 4.1.1 im Bedrohungsbaum der Schwachstelle *#AVB.1* (s. Abbildung 4.2) und die Stelle 3 im Bedrohungsbaum der Schwachstelle *#AVB.2* (s. Abbildung 4.3) abgeschnitten werden.

Fokus dieser Arbeit sind Mechanismen innerhalb der Echtzeit-Ethernet-Protokolle. Darum werden hier keine konkreten Konzepte zum Schutz der Hardware vorgestellt.

4.2.6 *#TSN.0.1* - Anforderungen der Basisprotokolle übernehmen

Diese Anforderung ist keiner vorher definierten Schwachstelle zugeordnet. Da TSN jedoch die Funktionen von TTE und AVB vereint, gelten alle *#TTE* und *#AVB* Anforderungen auch für TSN.

Die in dieser Arbeit vorgestellten Schutzkonzepte sollen für alle drei Echtzeit-Ethernet-Protokolle umsetzbar sein und damit diese Anforderung erfüllen.

4.2.7 *#TSN.1.1* - Begrenzung der eingehenden BE Nachrichten

Der Eingang von BE-Nachrichten muss an jedem Eingangsport begrenzt werden, um das Volllaufen der zuständigen Puffer zu verhindern. Damit kann die Systemverfügbarkeit gesichert werden. Diese Anforderung gilt auch für TTE und AVB.

Das gezielte Filtern von BE-Paketen wird in dieser Arbeit diskutiert und es werden Konzepte vorgestellt.

4.3 Bewertung der Risiken

In diesem Abschnitt wird das Risiko der aufgeführten Schwachstellen ermittelt. Ziel ist daraus die Priorisierung der Anforderungen abzuleiten.

Die qualitative Definition des Risikos ist hier die Multiplikation der quantifizierten Eintrittswahrscheinlichkeitsklasse (EK) mit dem quantifizierten Schadensklassenvektor ($\vec{SK}$). Das ermittelte Risiko ($\vec{R}$) ist wieder ein dreidimensionaler Vektor. Die Gleichung 4.1 zeigt die definierte Formel für die Berechnung des Risikos in dieser Arbeit.

$$\vec{R} = EK * \vec{SK} = EK * \begin{pmatrix} SK_T \\ SK_K \\ SK_F \end{pmatrix} \tag{4.1}$$

Diese Formel wird nun angewendet, um aus der jeweiligen Eintrittswahrscheinlichkeitsklasse und Schadensklassen der Schwachstellen das jeweilige Risiko zu ermitteln.

Tabelle 4.3: Ermittlung des Risikos ($\vec{R}$)

Schwachstelle	$EK*$	$(SK_T,$	$SK_K,$	$SK_F) =$	**Risiko ($\vec{R} = (R_T, R_K, R_F)$)**
#TTE.1	2	3	2	4	$(6, 4, 8)$
#AVB.1	3	3	2	4	$(9, 6, 12)$
#AVB.2	2	3	2	4	$(6, 4, 8)$
#TSN.1	3	3	2	4	$(9, 6, 12)$

Die Tabelle 4.3 zeigt alle nötigen Parameter zum Wrmitteln des Risikos und das letztendlich ermittelte Risiko. Dabei entspricht ein höherer Wert einem höheren Risiko. Die Dimensionen des Risikos entsprechen dem Risiko der jeweiligen Dimensionen Teilnehmer (R_T), Kommunikation (R_K) und Fahrzeug (R_F).

Die eins-zu-eins Beziehung zwischen dem Risiko der Schwachstelle und Priorität der zugehörigen Anforderungen ist in der Gleichung 4.2 zu sehen.

$$\vec{P} = \vec{R} \tag{4.2}$$

Diese Gleichung wird nun angewendet, um den Anforderungen Prioritäten zuzuordnen. Ein Sonderfall ist die Anforderung *#TSN.0.1*. Diese ist keiner entsprechenden Schwachstelle zugeordnet. Für die Priorität dieser Anforderung wird das Maximum der jeweiligen Dimensionen des Risikos der Schwachstellen *#TTE.1* und *#AVB.1* herangezogen.

Tabelle 4.4: Einordnung der Anforderungen

Anforderung	**Priorität ($\vec{P} = (P_T, P_K, P_F)$)**
#TTE.1.1	$(6, 4, 8)$
#AVB.1.1 #AVB.1.2 #AVB.1.3 #AVB.1.4	$(9, 6, 12)$
#AVB.2.1 #AVB.2.2	$(6, 4, 8)$
#TSN.0.1	$(9, 6, 12)$
#TSN.1.1	$(9, 6, 12)$

Die Tabelle 4.4 zeigt die vorgestellten Anforderungen und die ermittelte Priorität. Ein höherer Wert in einer Dimension entspricht einer höheren Priorität aus der Perspektive dieser Dimension. Damit lassen sich die Anforderungen hinsichtlich der definierten Dimensionen priorisieren. Zum Beispiel aus der Perspektive „Fahrzeug“ ist die priorisierte Reihenfolge der Anforderungen: *#AVB.1.1* (12), *#AVB.1.2* (12), *#AVB.1.3* (12), *#AVB.1.4* (12), *#TSN.0.1* (12), *#TSN.1.1* (12), *#TTE.1.1* (8), *#AVB.2.1* (8), *#AVB.2.2* (8)

Es wird deutlich, dass alle entdeckten Schwachstellen einen hohen Schaden in der Dimension Fahrzeug haben. Dies spiegelt sich auch in den Prioritäten der Anforderungen wieder. Die Analyse hat somit gezeigt, dass alle Schwachstellen für die Sicherheit des Gesamtsystems „Fahrzeug“ relevant sind. Das gleiche gilt für die Perspektiven „Kommunikation“ und „Teilnehmer“ auch wenn die Zahlen hier niedriger sind.

5 Schutzkonzepte

Dieses Kapitel stellt Konzepte zum Schutz vor. Diese sollen die Schwachstellen schließen die in Abschnitt 4.1 ermittelt wurden indem die Anforderungen aus Abschnitt 4.2 umgesetzt werden.

In Abschnitt 5.1 werden die hier vorgestellten Konzepte abgegrenzt und es werden Basisannahmen für das System getroffen. Im darauf folgendem Abschnitt 5.2 werden Konzepte zum Filtern von eingehenden Nachrichten vorgestellt und ein Konzept für diese Arbeit entwickelt. Hier werden auch zwei in dieser Arbeit entwickelte Ideen zum zustandsbasierten Filtern dargestellt. Der Abschnitt 5.3 geht auf Konzepte zur einheitlichen Beschreibung der Filterkonfiguration und laufenden Aktualisierung ein. Abschließend wird in Abschnitt 5.4 der weitere Nutzen dieser Konzepte für eine Anomalieerkennung diskutiert.

5.1 Authentifizierung, Verschlüsselung und Schutz der Switche

Die in Abschnitt 4.2 vorgestellten Anforderungen *#TTE.1.1*, *#AVB.1.3* und *#AVB.2.1* erfordern die Authentifizierung von Protokollteilnehmern und die Verschlüsselung der Protokollnachrichten. In dieser Arbeit werden keine konkreten Konzepte für Authentifizierung und Verschlüsselung vorgestellt. Für alle weiteren Konzepte wird daher angenommen, dass diese Anforderungen erfüllt sind.

Es gibt Arbeiten, im Bereich der sicheren Uhrensynchronisation für TTE (vgl. Steiner (2013)) und TSN (vgl. Itkin und Wool (2017)), die sich mit diesem Problem auseinandersetzen.

Auch Steiner (vgl. Steiner (2013)) kommt nach eingehender Analyse der Stärken und Schwächen der Informationssicherheit von TTE zu dem Schluss, dass unter anderem Erweiterungen zur Authentifizierung der Teilnehmer notwendig sind.

Itkin und Wool (vgl. Itkin und Wool (2017)) führen in ihrer Arbeit eine detaillierte Sicherheitsanalyse des Precision Time Protocol (PTP) durch. Die Funktion von PTP ist im Standard IEEE 1588 (vgl. Institute of Electrical and Electronics Engineers (2008a)) festgelegt. In TSN wird eine spezifizierte Version unter der Bezeichnung IEEE 802.1AS (vgl. IEEE 802.1 TSN Task Group (b)) standardisiert. Danach werden von Itkin und Wool konkrete Lösungen vorgestellt, die die verbleibenden Schwachstellen schließen.

Auch für die Anforderungen *#AVB.1.4* und *#AVB.2.2* gilt, dass ihre Erfüllung für alle weiteren Konzepte angenommen wird. In dieser Arbeit werden keine konkreten Konzepte zum Schutz der Switche vorgestellt. Die Informationssicherheit der Switche wird vorausgesetzt.

5.2 Eingangsverkehr filtern

Im klassischen Netzwerkverkehr ist das Filtern des Eingangsverkehrs ein verbreitetes Instrument. Eingangsfilter verhindern das eintreffen von ungewollten Paketen im eigenen Netzwerk. Des weiteren können sie das Überschwemmen eines Netzwerkknotens und damit die Einstellung ihres Services (DoS) verhindern. Dies wird erreicht, indem ausgewählte Pakete am Netzwerkknoteneingang gelöscht werden. Dadurch werden sie vom folgendem Weiterleitungsprozess ausgeschlossen und können diesen nicht mehr beeinflussen. Die Strategie mit der diese Pakete ausgewählt werden wird durch ein Filterverfahren beschrieben.

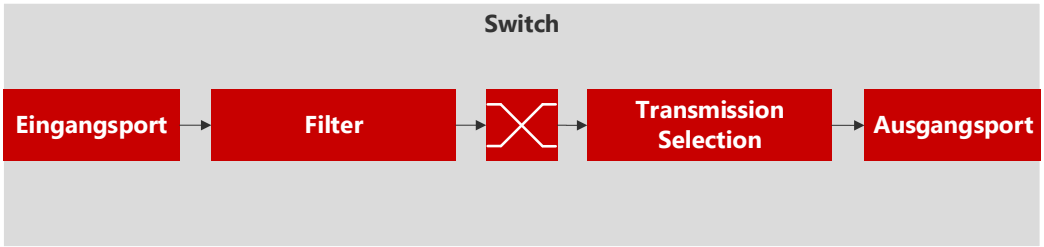

Abbildung 5.1: Pfad einer Nachricht innerhalb eines Switches

Im Gegensatz zu einer klassischen Firewall soll in diesem Fall das Filtern verteilt auf allen Netzwerkknoten durchgeführt werden, um eine Überlastung des Netzwerks aus jeder Richtung zu verhindern.

Die Abbildung 5.1 zeigt den Pfad einer Nachricht durch einen Switch. Der Filter ist das Gegenstück zur Transmission Selection und filtert Nachrichten direkt am Eingangsport. Auch andere Netzwerkteilnehmer setzen das Filtern am Eingang ein.

Das Filterkonzept für den Einsatz in Echtzeit-Ethernet-Protokollen orientiert sich an dem in Abschnitt 2.1.3 vorgestellten TSN Standard IEEE 802.1Qci (vgl. Institute of Electrical and Electronics Engineers (2017)). Es werden jedoch zwei Erweiterungen vorgenommen.

Zum einen können Pakete durch einen individuellen Identifier identifiziert werden. Es wird somit nicht nur die Stream ID zur Identifikation eingesetzt. Auf diese Weise können zum

Beispiel auch BE-Nachrichten auf Basis unterschiedlicher Header-Informationen identifiziert werden.

Des weiteren können unterschiedliche Meter-Instanzen innerhalb eines Meter-ID-Blocks existieren. Die Meter-Instanzen werden verkettet, so das ein Paket diese alle nacheinander passieren muss. Die Funktion einer Verkettung kann auch durch einen einzelnen Meter implementiert sein. Jedoch können mit der Verkettung komplexe Filterfunktionen aus einfachen Bausteinen komponiert werden. Diese Verkettung von Filtern wird so auch in aktuellen Firewalls eingesetzt.

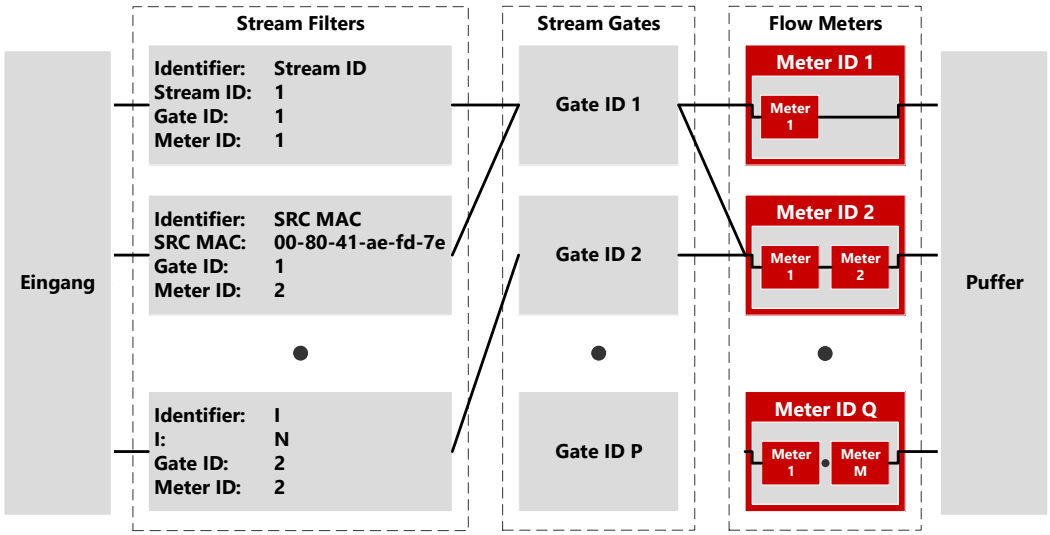

Abbildung 5.2: Echtzeit-Ethernet-Filter Konzept

Die Abbildung 5.2 zeigt den Aufbau eines solchen Echtzeit-Ethernet-Filters. Dieser wird pro Port eingesetzt und überwacht alle eingehenden Pakete. Weiterhin besteht der Filter immer aus drei Abschnitten. Diese sind „Stream Filters“, „Stream Gates“ und „Flow Meters“.

Im ersten Abschnitt werden die Pakete durch zustandslose Filter identifiziert. Ein Identifier-Feld legt fest, welche Eigenschaft der jeweilige „Stream Filter“zur Identifikation einsetzt. Dies kann zum Beispiel die Stream ID oder Quell-MAC-Adresse sein. Es kann eine standardmäßige Strategie ausgewählt werden. Diese entscheidet, ob nicht identifizierte Pakete verworfen oder direkt an die Puffer weitergeleitet werden.

Danach folgen die „Stream Gates“. Diese haben entweder den Zustand offen oder geschlossen. Der Zustandswechsel kann durch eine Offlinekonfiguration in Abhängigkeit von der

Zeit erfolgen. Des weiteren kann der Zustandswechsel auch dynamisch durch eine beliebige Kontrollinstanz oder einen Meter geregelt werden.

Im letzten Abschnitt befinden sich die Meter. Diese können entweder eine Meter-Instanz sein oder beliebig viele verkettete Meter-Instanzen enthalten. Pakete, die alle Abschnitte passieren werden gepuffert.

Ziel des Echtzeit-Ethernet-Filters ist, das Erfüllen der in Abschnitt 4.2 vorgestellten Anforderungen *#AVB.1.1*, *#AVB.1.2*, *#TSN.0.1* und *#TSN.1.1*

Im Folgenden werden unterschiedliche Arten von Filtern vorgestellt, die innerhalb des beschriebenen Echtzeit-Ethernet-Filters eingesetzt werden können. Diese sind entweder zustandslos (s. Abschnitt 5.2.1), zustandsbehaftet (s. Abschnitt 5.2.2) oder gehören zur Kategorie Deep Packet Inspection (DPI) (s. Abschnitt 5.2.3). Des weiteren werden neu erarbeitete Ideen für das Filtern von Echtzeit-Ethernet-Paketen vorgestellt.

5.2.1 Zustandslose Filter

Zustandslose Filter entscheiden statisch auf Basis konfigurierter Werte ob ein Paket angenommen oder verworfen wird. Dabei wird, im Fall von Ethernet, meist lediglich der Header der Sicherungsschicht eingehender Nachrichten überprüft.

Beispiele hierfür sind das Filtern nach MAC-Adresse oder das Filtern nach IDs in TTE (s. Abschnitt 2.1.1).

Bei ersterem kann zum Beispiel eine Black- oder Whitelist konfiguriert werden. Nachrichten deren Ziel-MAC-Adresse nicht mit der Whitelist übereinstimmen werden dann verworfen.

Zum Beispiel bei TTE werden Nachrichten verworfen, wenn sie einer ID entsprechen, die auf dem Gerät für den Port keine Konfiguration vorweisen kann.

In diesem Filterkonzept werden zustandslose Filter für die Identifikation und zur Zuordnung der Pakete eingesetzt. Des weiteren können einfache zustandslose Meter eingesetzt werden, um Nachrichten zu filtern.

Beispielsweise könnte eine VLAN-Grenze an dem Port im Netzwerk konfiguriert sein und bestimmte VLAN IDs werden von einem Meter erkannt und verworfen.

5.2.2 Zustandsbehaftete Filter

Zustandsbehaftete Filter entscheiden auf Basis des aktuellen Zustands, ob eine Nachricht angenommen oder verworfen wird. Dabei spielt nicht nur der Header der Sicherungsschicht

eine Rolle. Alle Header bis zur Anwendungsschicht[1] können teilweise von diesen Filtern analysiert werden.

Ein klassisches Beispiel aus der Welt der Firewalls hierfür ist das Filtern von Ethernet-Nachrichten auf Basis von aktiven TCP-Verbindungen. Dabei merkt sich der Filter den Zustand der TCP-Verbindung und verwirft Nachrichten, die im aktuellen Zustand nicht dem spezifizierten TCP-Protokollverhalten entsprechen.

Auch diese Art des Filterns wird vom Konzept des Echtzeit-Ethernet-Filters unterstützt. Diese werden als Meter im Echtzeit-Ethernet-Filter umgesetzt. Zwei in dieser Arbeit entwickelte Konzepte werden in den folgenden Abschnitten 5.2.2 und 5.2.2 vorgestellt. Diese sind nicht im IEEE 802.1Qci Standard spezifiziert und sollen konkret die Anforderungen *#AVB.1.1*, *#AVB.1.2* und *#TSN.1.1* erfüllen.

Credit Based Meter

Credit Based Meter (CBM) ist der Name für einen in dieser Arbeit entwickelten Filter, der als ein Meter in dem Echtzeit-Ethernet-Filterkonzept eingesetzt wird. Er agiert als Gegenstück zum in Abschnitt 2.1.2 vorgestellten CBS. Ziel ist es zu verhindern, dass die Bandbreite mit der Nachrichten eines Streams eintreffen, die reservierte Bandbreite dieses Streams übertrifft. Gleichzeitig wird der durch den CBS erlaubte Burst[2], nach Verzögerung des Streams, erlaubt.

Die Gradienten $idleslope$ und $sendslope$ werden auch hier eingesetzt. Diese sind hier abhängig von den Streams, die diesem Meter zugewiesen werden. Die Gleichungen 5.1 und 5.2 zeigen die Berechnung der Gardienten $idleslope_{cbm}$ und $sendslope_{cbm}$. Dabei ist RB_S die reservierte Bandbreite eines Streams und B die maximale Bandbreite des Ethernet-Links. N beschreibt die Anzahl der Streams die durch „Stream Filter“ an diesen Meter geleitet werden. $RB_S(n)$ ist die für die jeweiligen Streams reservierte Bandbreite.

$$idleslope_{cbm} = \sum_{n=1}^{N} RB_S(n) \tag{5.1}$$

$$sendslope_{cbm} = (\sum_{n=1}^{N} RB_S(n)) - B \tag{5.2}$$

Ein weiterer Parameter des CBM-Algorithmus ist $Credit_{max}$. Dieser Parameter ist abhängig von einem konfigurierbaren Parameter $Burst_{max}$. Dieser beschreibt die maximale Anzahl an

[1] Anwendungsschicht (englich: Application Layer) beschreibt die 7. Schicht des OSI-Modells. Diese Schicht stellt Funktionen für Anwendungen zur Verfügung.

[2] Burst (deutsch: Signalblock, Signalfolge) bezeichnet eine Anhäufung von direkt aufeinander folgenden Nachrichten auf einem Übertragungsmedium.

Paketen, die in einem Burst aufeinanderfolgend eintreffen dürfen. In Kombination mit der maximalen Übertragungsdauer eines Streampakets (T_S), die wiederum von der maximalen Paketgröße (FS_S) aller Streams, der Portbandbreite (B) und dem Ethernet IFG[3] (T_{ifg}) abhängig ist, wird der maximale Credit-Wert ($Credit_{max}$) bestimmt. Die Gleichungen 5.3 und 5.4 zeigen wie das Verhältnis von $Burst_{max}$ und $Credit_{max}$ konkret aussieht.

$$T_S = \frac{FS_S}{B} + T_{ifg} \tag{5.3}$$

$$Credit_{max} = |sendslope_{cbm}| * T_S * (Burst_{max} - 1) \tag{5.4}$$

Ein Burst mit der Länge von einem Paket ist erlaubt, wenn der Credit 0 ist. Daher wird 1 von $Burst_{max}$ subtrahiert. Wenn folglich der maximale Burst mit 1 initialisiert wird ($Burst_{max} = 1$), folgt daraus ein maximaler Credit von 0 ($Credit_{max} = 0$). 1 ist damit auch der kleinste konfigurierbare Wert. Bei 0 würde bereits jedes Paket verworfen werden, da ein Burst mit der länge 1 das Maximum bereits übertrifft.

Die Abbildung 5.3 zeigt ein UML-Zustandsdiagramm des CBM-Zustandsautomaten. Neben dem Initialzustand gibt es zwei weitere Zustände. Zu Beginn wird Credit mit 0 initialisiert.

Ersterer nennt sich „Running Receiving Allowed“ (R-RA). In diesem Zustand sind eingehende Pakete des Streams erlaubt und werden weitergeleitet. Dabei wird in Abhängigkeit zur Übertragungsdauer des Pakets der Credit mit $sendslope_{cbm}$ dekrementiert. Wenn daraufhin der Credit kleiner ist als 0, wechselt der Automat in den Zustand R-RF. Andernfalls bleibt er im Zustand R-RA. In diesem wird, immer wenn kein Stream-Paket eintrifft, der Credit in Abhängigkeit zur verstrichenen Zeit mit $idleslope_{cbm}$ inkrementiert. Dabei wird $Credit_{max}$ nie überschritten.

Der zweite Zustand heißt „Running Receiving Forbidden“ (R-RF). In diesem Zustand werden ankommende Pakete des Streams verworfen. Des weiteren wird der Credit in Abhängigkeit zur verstrichenen Zeit mit $idleslope_{cbm}$ inkrementiert. Wenn der Credit danach größer oder gleich 0 ist, wechselt der Automat in den Zustand R-RA. Andernfalls bleibt der Zustand R-RF erhalten.

Die Abbildung 5.4 zeigt wie der CBM auf ein beispielhaftes Szenario reagiert. Zu Beginn ist der Wert des Credits bei 0.

[3] Interframe Gap (IFG) bezeichnet den minimalen zeitlichen Abstand zwischen Paketen auf einem Übertragungsmedium.

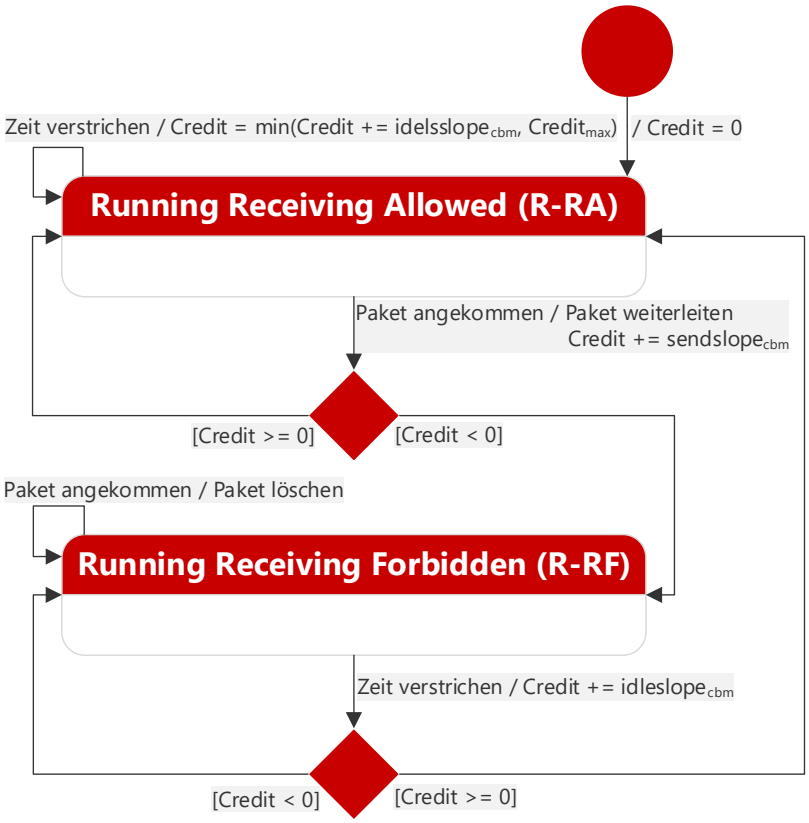

Abbildung 5.3: UML-Zustandsdiagramm des Credit Based Metering Algorithmus

Zum Zeitpunkt t_1 trifft ein Streampaket ein und der Credit wird für den Zeitraum der Übertragung dekrementiert. Danach ist der Credit zum Zeitpunkt t_2 negativ. Aus diesem Grund wechselt der Automat in den Zustand R-RF bis der Credit wieder 0 erreicht.

Als nächstes tritt im CBS auf der Gegenseite die Situation auf, dass ein Stream-Paket blockiert wird, da sich bereits ein BE-Paket auf dem Übertragungsmedium befindet. Das führt dazu, dass zum Zeitpunkt t_3 als nächstes das besagte BE-Paket am CBM eintrifft. Anders als beim CBS, macht das für den Credit keinen Unterschied. Dieser wird beim CBM immer inkrementiert bis ein Stream-Paket, für das die jeweilige Instanz verantwortlich ist, eintrifft.

Dies ist zum Zeitpunkt t_4 der Fall. Nach der Übertragung des zweiten Streampakets (t_5) ist der Wert des Credits noch positiv. Darum dürfte direkt darauffolgend ein weiteres Paket eintreffen. Dieses trifft in diesem Beispiel jedoch erst zum Zeitpunkt t_6 ein.

Das Verhalten des CBM ist abhängig von der $Burst_{max}$-Konfiguration. Das Ziel ist, den Parameter so niedrig wie möglich zu konfigurieren und dabei trotzdem das Worst-Case-Verhalten

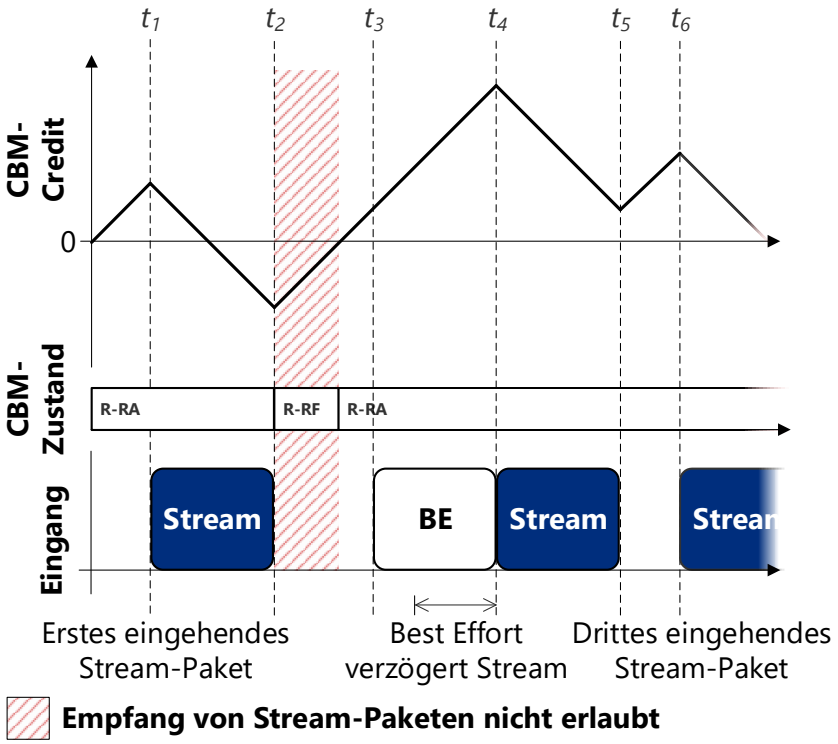

Abbildung 5.4: Credit Based Metering Beispiel

zu unterstützen. Zum einen liegt die Begründung dafür im Verhalten des CBS. Die maximale Anzahl von aufeinanderfolgenden Paketen ist abhängig vom jeweiligen Worst-Case. Des weiteren kann ein Angreifer mit dem Erzeugen des Worst-Case-Verhaltens keinen Schaden am Netzwerk anrichten, weil der Netzwerkentwurf diesen Fall einschließt.

Es gibt unterschiedliche Wege den passenden Wert für $Burst_{max}$ festzulegen. Ein Beispiel ist, den maximalen Burst an jedem Ausgang ($Burst_{out}$) zu errechnen. Wenn dieser Wert bekannt ist, muss dieser um 1 inkrementiert werden, um ein weiteres Paket zu erlauben, dass kurz nach dem Burst eintrifft. Die Formel hierfür ist in Gleichung 5.5 zu sehen.

$$Burst_{max} = Burst_{out} + 1 \tag{5.5}$$

Eine andere Möglichkeit ist die Ermittlung des $Burst_{max}$ Parameters durch explorative Simulation.

Das CBM-Konzept verhindert das Überziehen der reservierten Bandbreite. Es erlaubt aber auch das Aufholverhalten des CBS. Daher erfüllt es die Anforderungen *#AVB.1.1* (s. Abschnitt 4.2.2) und *#AVB.1.2* (s. Abschnitt 4.2.3).

Minimum Gap Meter

Der Minimum Gap Meter (MGM) ist ein möglicher Filter, der einen Mindestabstand zwischen Paketen pro Quelle überwacht. Zur Identifikation der Quelle wir die Quell-MAC-Adresse herangezogen. Der Mindestabstand ist ein konfigurierbarer Parameter.

Dieses Verhalten könnte mit angepasster Parametrisierung auch durch den CBM bewerkstelligt werden. Für die einfache Einhaltung eines Mindestabstands ist dies jedoch eine aufwendige Konfiguration.

Der MGM betrachtet die Quell-MAC-Addresse jedes eingehenden Pakets und zieht eine ablaufende Uhr auf, wenn diese Uhr für die Quelle nicht bereits existiert. Wenn diese Uhr bereits existiert wird das Paket verworfen. Wenn eine Uhr abläuft, wird sie entfernt.

Im Ergebnis wird damit, wie bei der RC-Klasse von TTE, eine maximale Bandbreite konfiguriert, die eine Quelle mit den durch die „Stream Filter“ zugeordneten Paketen, nicht überziehen darf.

Um die Anforderung *#TSN.1.1* zu erfüllen, wird ein „Stream Filter“ konfiguriert, der zur Identifikation die Nachrichtenklasse nutzt. Alle BE-Pakete werden zu einem MGM geleitet und dort differenziert nach der Quell-MAC-Adresse auf den Mindestabstand überprüft.

5.2.3 Deep Packet Inspection

DPI-Filter betrachten alle Bestandteile einer Nachricht, von den Headern bis hin zu den Daten. Dabei werden Signaturen gebildet. Diese werden mit bekannten Signaturen, von Angriffen oder anderen Paketen von Interesse, verglichen. Bei Übereinstimmung wird zum Beispiel das betreffende Paket gelöscht. Für das Bilden und Vergleichen der Signaturen werden unterschiedlichste Methoden herangezogen.

Beispiele für den Einsatz von DPI sind die Absicherung einer Netzwerkstruktur gegen bekannte Angriffsmuster, das automatische Erkennen und löschen von illegalen Inhalten (z.B.: Texte, Bilder, Videos) oder die Erkennung von Copyright Verstößen (vgl. El-Maghraby u. a. (2017)).

Technisch können auch DPI-Meter im Echtzeit-Ethernet-Filter eingesetzt werden. Die Meter operieren jedoch auf der Sicherungsschicht. Die Daten werden aus dieser Perspektive verschlüsselt vorliegen, was wiederum die Signatur verfälscht. Des weitern sind Daten in

Ethernet-Nachrichten meist fragmentiert und ergeben erst in höheren Protokollschichten ein zusammenhängendes Datenpaket.

Jedoch ist es trotzdem sinnvoll diesen Typ von Filtern in höheren Schichten eines Fahrzeugbordnetzes einzusetzen, um zum Beispiel eingehenden Verkehr aus einer Schnittstelle zum Internet zu filtern.

5.3 Systemkonfiguration

Das Fahrzeugbordnetz wird je nach Modell oder Ausstattung in unterschiedlichen Konfigurationen ausgeliefert. Des weiteren müssen defekte Geräte ausgetauscht werden. Auch nachträgliche Erweiterungen sind denkbar. Damit die vorgestellten Konzepte die Sicherheitsanforderungen erfüllen können, müssen diese an den aktuellen Netzaufbau und seine Teilnehmer angepasst sein.

Wenn in einem Fahrzeug, zum Beispiel durch höher auflösende Sensoren, mehr Daten anfallen, müssen die Filter entsprechend konfiguriert werden, um die effiziente und sichere Operation des Kommunikationsnetzes zu gewährleisten.

Damit die Konfiguration dieses Netzwerks so dynamisch wie möglich gestalten werden kann, müssen Möglichkeiten zur einheitlichen Beschreibung vorhanden sein. So dass zum Beispiel die Echtzeit-Ethernet-Filterkonfiguration des gesamten Netzwerks über eine Beschreibung gesammelt initialisiert und aktualisiert werden kann.

Ziel dieses Abschnitts ist die Vorstellung von Konzepten zur einheitlichen Beschreibung der Filterkonfiguration.

In Abschnitt 5.3.1 wird ein Konzept einer einheitlichen Beschreibung diskutiert. Im darauffolgenden Abschnitt 5.3.2 wird beschrieben, welche Auswirkungen eine laufende Aktualisierung haben kann.

5.3.1 Filterbeschreibung

Die Konfiguration der verteilten Echtzeit-Ethernet-Filter soll in einer Konfigurationsbeschreibung gesammelt werden. Diese kann dann genutzt werde, um die Konfiguration der Netzwerkteilnehmer einzeln oder über eine Verwaltungsebene gesammelt zu aktualisieren.

Ein Beispiel aus dem Software Defined Networking (SDN) ist OpenFlow (vgl. Open Networking Foundation (2015)). Es erlaubt die Konfiguration eines gesamten Netzwerks über Software, die verteilt auf den Routern läuft. Dafür müssen die Geräte im Netzwerk „OpenFlow “ unterstützen. In einem solchen Netzwerk lassen sich von zentraler Stelle die Weiterleitungsregeln (Flows) der einzelnen Switche bestimmen.

Abbildung 5.5: OpenFlow Flow-Eintrag (vgl. Open Networking Foundation (2015))

Die Abbildung 5.5 zeigt den Aufbau eines solchen Flow-Eintrags. Das erste Feld ist das „Match Field“. Dieses beschreibt die Pakete für die der Flow zuständig ist. Mit dem „Priority“-Feld kann eine Rangordnung der zuständigen Flows definiert werden. Daraufhin folgt das Feld „Counter“. Hier werden die passenden Pakte gezählt. Das Feld „Instructions“ beschreibt, was mit dem Pakete geschehen soll. Dies kann entweder eine Aktion wie „Paket löschen“ sein oder die Zuweisung zu einem bestimmten Meter enthalten. In „Timeouts“ wird bestimmt, wann der Flow ungültig wird. In „Cookie“ und „Flags“ werden weitere Verwaltungsinfomationen beschrieben.

Die Beschreibung der Konfiguration von Echtzeit-Ethernet-Filtern könnte sich an dem Aufbau der „OpenFlow“-Flow-Einträge orientieren.

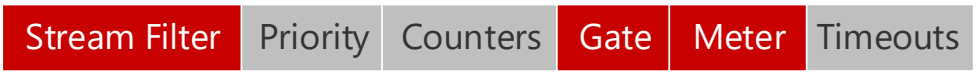

Abbildung 5.6: Eintrag einer Echtzeit-Ethernet-Filter Beschreibung

In Abbildung 5.6 wird der mögliche Aufbau eines Eintrags zur Echtzeit-Ethernet-Filter Beschreibung gezeigt. Das Feld „Stream Filter“ beschreibt die in Abschnitt 5.2 vorgestellten Parameter eines „Stream Filter“ zur Identifizierung der eingehenden Pakete.

In „Priority“ wird weiterhin die Priorität des Eintrags festgehalten. Damit kann die Reihenfolge mit der ein Paket anhand der „Stream Filter“identifiziert wird, konfiguriert werden.

Das Feld „Counters“ wird, wie bei „OpenFlow“ zum Zählen der für diesen Eintrag passenden Pakete genutzt.

Im Feld „Gate“ wird das zuständige „Gate“beschrieben. Dies kann auch zeitliche Abfolgen oder Zuständigkeiten enthalten.

Das „Meter“-Feld enthält die Konfiguration des Meters. Auch eine Verkettung mehrerer Meter wird hier festgelegt.

Zuletzt wird das „Timouts“-Feld genutzt, um die Gültigkeit von Einträgen auslaufen zu lassen. Dies führt zum Entfernen der jeweiligen Konfiguration im Echtzeit-Ethernet-Filter. Ohne Informationen in diesem Feld ist der Eintrag ewig gültig.

5.3.2 Laufende Aktualisierung

In Zukunft kann es erforderlich sein, dass diese Konfigurationen im laufenden Betrieb aktualisiert werden müssen. Also ohne das Fahrzeug zur Wartung in eine Werkstatt zu bewegen.

Ein Beispiel dafür wäre, wenn Systemanwendungen laufend aktualisiert werden und damit auch der Kommunikationsaufwand einer Funktion steigt oder sinkt.

Ein Angreifer, der diese Aktualisierung der Konfiguration manipuliert, kann alle vorgestellten Schutzkonzepte damit unterwandern.

Die Aktualisierung der Konfiguration muss deswegen wieder Techniken zur Authentifizierung und Verschlüsselung einsetzen, um diese Manipulation auszuschließen.

5.4 Nutzen für Anomalieerkennung

Anomalieerkennungssysteme, bekannt als Intrusion Detection System (IDS), sind Systeme welche Angriffe erkennen und melden. Es ist ein weitverbreitetes Element der Informationssicherheit im Internet. Ein Typ dieser Systeme nennt sich Network-based Intrusion Detection System (NIDS) und ist für die Erkennung von Angriffen anhand des Netzwerkverkehrs zuständig (vgl. Haas u. a. (2017)).

Solche Systeme wurden schon in Bezug auf CAN- (vgl. Cain (2015)) und Echtzeit-Ethernet-Kommunikation (vgl. Phieler (2017)) diskutiert.

Für die Funktion dieser IDS muss ein Normalverhalten oder Angriffsverhalten definiert werden, damit das System einen Angriff erkennen und melden kann.

Hier kommt der Nutzen des in Abschnitt 5.2 vorgestellten Echtzeit-Ethernet-Filters für die Anomalieerkennung ins Spiel. Durch die Filter und deren Konfiguration ist bereits ein Normalverhalten des Netzwerkverkehrs auf Sicherungsschicht definiert. Ein NIDS System sollte Informationen, wie die Anzahl der verworfenen Pakete durch die Filter und andere filterspezifische Metriken nutzen, um Angriffe zu erkennen.

6 Erweiterung des Simulationsmodells

Der in Kapitel 5 vorgestellte Echtzeit-Ethernet-Filter, wird in einer Simulationsumgebung implementiert, um dessen Auswirkungen auf das Bordnetz weiter zu analysieren. Dafür wird die in Abschnitt 2.3 vorgestellte Simulationsumgebung erweitert.

Ziel ist die Implementierung des Echtzeit-Ethernet-Filters im CoRE4INET Framework. Des weiteren wird für diesen Filter ein CBM-Algorithmus implementiert, um das Konzept im Einsatz zu untersuchen. Um Angriffsszenarien zu simulieren, werden zusätzlich kompromittierte Teilnehmer umgesetzt.

In Abschnitt 6.1 wird das Konzept für die Erweiterung von CoRE4INET vorgestellt. Daraufhin folgt in Abschnitt 6.2 die Beschreibung der Umsetzung des Konzepts. Am Schluss wird in Abschnitt 6.3 die Qualitätssicherung vorgestellt.

6.1 Konzept

Der Echtzeit-Ethernet-Filter soll erweiterbar und modular sein, damit zum Beispiel neue Meter oder auch komplett andere Filterkonzepte einfach und schnell umgesetzt werden können.

Des weiteren soll sich der Filter nahtlos in die vorhandene Struktur des CoRE4INET Frameworks integrieren. Wenn kein Filter aktiv ist, soll das Verhalten komplett unbeeinflusst sein. Für alle anderen beteiligten Module soll der Filter transparent sein und keine Modifikation dieser Module erfordern.

Der kompromittierte Teilnehmer muss volle Kontrolle über seinen Port haben, damit er Angriffsmuster fahren kann, die nicht protokollvalide sind. Diese Kontrolle soll für den Echtzeit-Ethernet-Protokoll-Stack transparent sein. Auf diese Weise kann eine Kompromittierung des Teilnehmers zu einem beliebigen Zeitpunkt nach dem Start der Simulation erfolgen.

Die Dynamik des Echtzeit-Ethernet-Filter Konzepts soll sich auch in der Konfiguration der Simulation widerspiegeln. Welche konkreten Implementierungen für „Stream Filter“, „Meter“ oder „Gates“ eingesetzt werden, wird konfigurierbar gemacht.

6.2 Umsetzung

Es wurden zwei Funktionen in CoRE4INET umgesetzt. Zum einen der Filter und zum anderen der kompromittierte Netzwerkteilnehmer.

In Abschnitt 6.2.2 wird die Implementierung des Filters beschrieben. Im darauffolgenden Abschnitt 6.2.2 die Umsetzung des kompromittierten Teilnehmers.

6.2.1 Filter

Für den Filter wird ein Module mit dem Namen „filtering“ in CoRE4INET eingeführt.

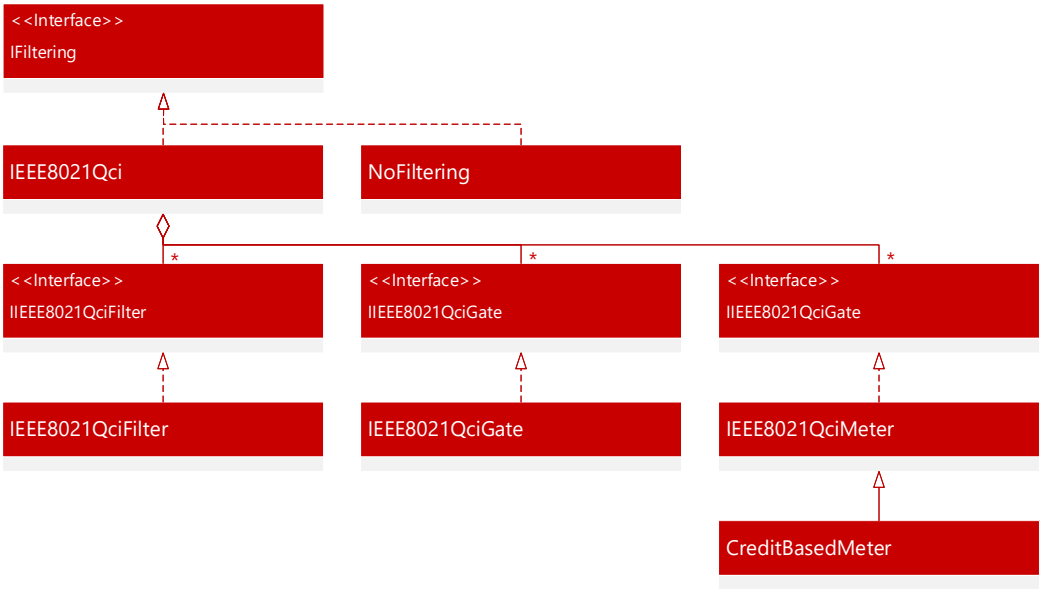

Abbildung 6.1: UML-Klassendiagramm des filtering Moduls

Die Abbildung 6.1 zeigt ein UML-Klassendiagramm des neuen „filtering“ Moduls. Dieses Modul muss das Interface „IFiltering“ implementieren. In diesem Fall gibt es zwei Implementierungen mit den Namen „NoFiltering“ und „IEEE8021Qci“. Welche Implementierung jeweils eingesetzt wird ist pro Port konfigurierbar.

„NoFiltering“ leitet jedes eingehende Paket direkt an den Ausgang des Filters weiter. Es findet also kein Filtern statt.

„IEEE8021Qci“ besteht wiederum aus beliebig vielen Modulen, die eines der Interfaces „IIEEE8021QciFilter“, „IIEEE8021QciGate“ oder „IIEEE8021QciMeter“ implementieren müssen. Im

Bild ist die Implementierung der „Stream Filter“ („IEEE8021QciFilter“), „Stream Gate“ („IEEE8021QciGate“) und „Flow Meter“ („IEEE802QciMeter“) des IEEE 802.1Qci Standards zu sehen (s. Abschnitt 2.1.3). Welche Implementierung jeweils eingesetzt wird ist konfigurierbar.

Des weitern ist auf der Abbildung die Implementierung des CBM eingezeichnet. Diese erbt von der IEEE 802.1Qci Meter Implementierung.

Im Folgendem wird gezeigt, wie sich das „filtering“ Modul in einem konkreten Switch in der Simulation einbettet.

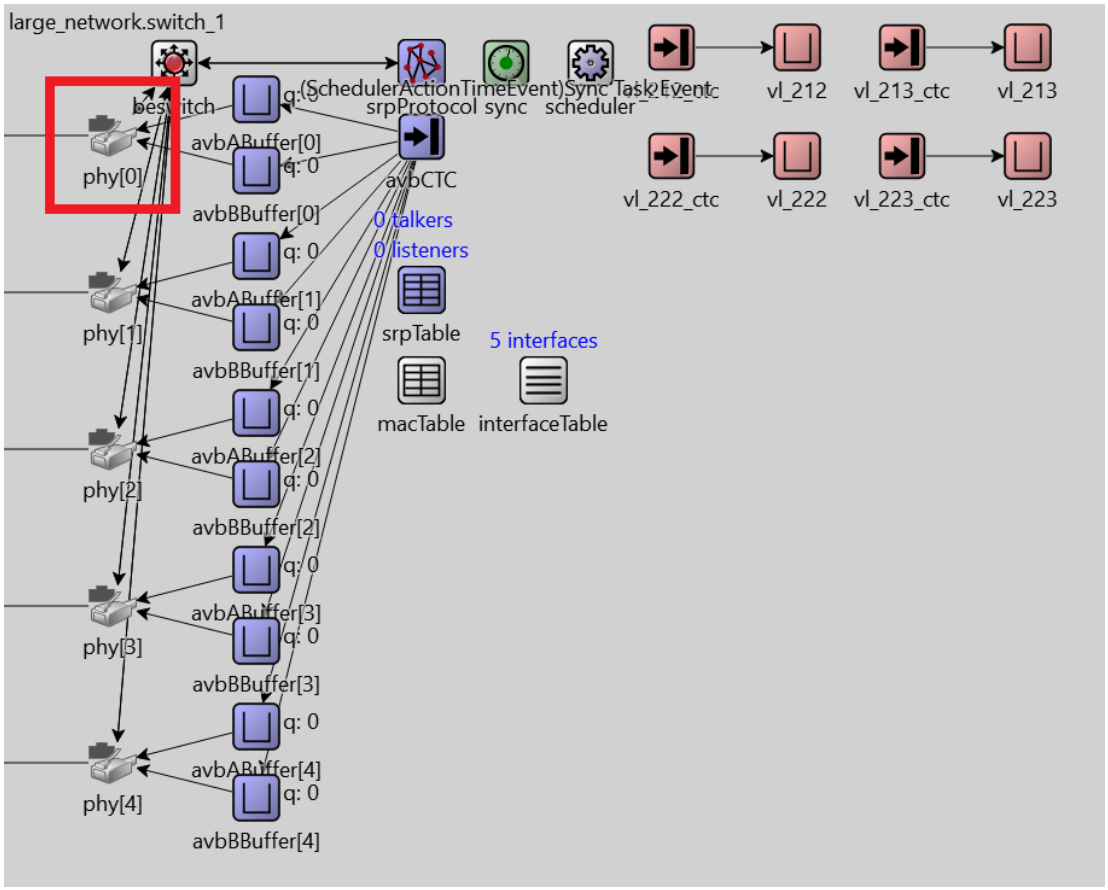

Abbildung 6.2: Beispiel des Aufbaus eines Switches in der Simulation

In der Abbildung 6.2 ist der innere Aufbau eines Switches in der Simulation zu sehen. Dieser Switch hat fünf Portmodule, die auf der linken Seite angeordnet sind. In jedem dieser Portmodule sind Filtermodule enthalten. Repräsentativ wird „phy[0]“ für die weitere Betrachtung ausgewählt.

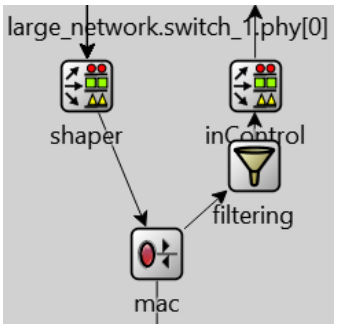

Abbildung 6.3: Beispiel des Aufbaus eines Ports in der Simulation

Die Abbildung 6.3 zeigt den inneren Aufbau des Ports „phy[0]“. Jedes ein- und ausgehende Paket muss dieses Modul passieren.

Ausgehender Verkehr kommt im „shaper“ Modul an und wird an das „mac“ Module weitergeleitet. Von dort aus wird das Paket auf das Übertragungsmedium gelegt.

Eingehender Verkehr kommt im „mac“ Modul an und wurde zuvor direkt an das „inControl“ Modul weitergeleitet. Nun ist das „filtering“ Modul direkt zwischengeschaltet. Pakete die hier verworfen werden, erreichen das „inControl“ Modul nicht.

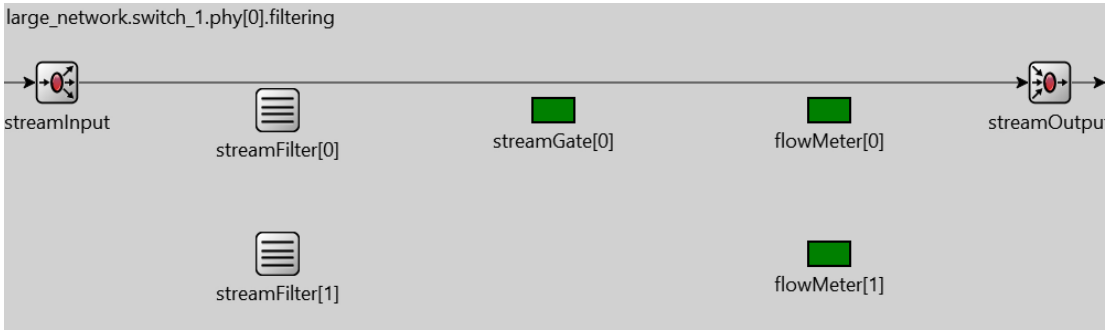

Abbildung 6.4: Beispiel des Aufbaus des Filters in der Simulation

Schließlich zeigt die Abbildung 6.4 den inneren Aufbau des „filtering“Moduls. In dieser Konfiguration gibt es zwei „streamFilter“, ein „streamGate“ und zwei „flowMeter“. Pakete, für die kein „streamFilter“zuständig ist, werden direkt an den Ausgang weitergeleitet. Ansonsten müssen sie die jeweilig konfigurierten „streamGate“ und „flowMeter“ Module passieren.

6.2.2 Kompromittierter Teilnehmer

Für den kompromittierten Teilnehmer wird ein Modul mit dem Namen „CorruptedTTEAVBEtherHost“ in CoRE4INET eingeführt. Dieses simuliert einen Netzwerkteilnehmer, der durch einen Angriff übernommen wurde und sich nicht mehr protokollkonform verhalten muss.

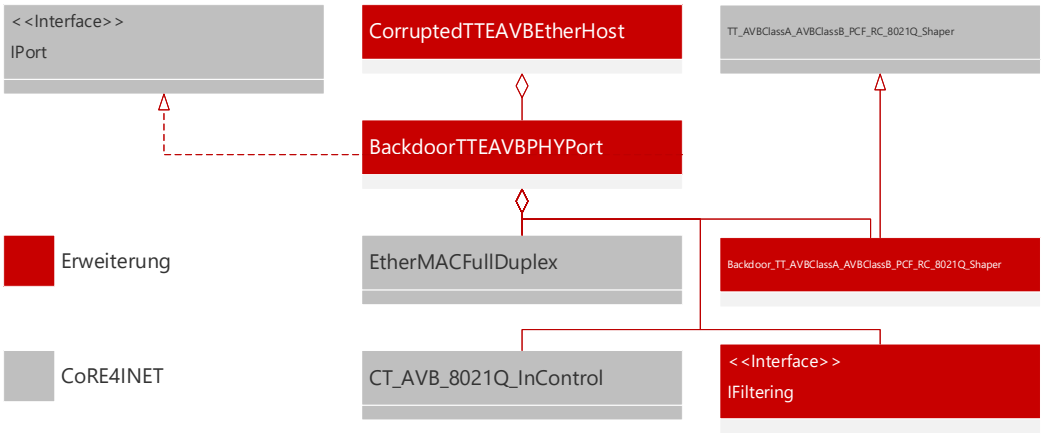

Abbildung 6.5: UML-Klassendiagramm des filtering Moduls

Die Abbildung 6.5 zeigt ein UML-Klassendiagramm des „CorruptedTTEAVBEtherHost“ Moduls. In diesem Diagramm sind neue Klassen rot und CoRE4INET Klassen grau markiert.

Für den Teilnehmer wurde ein neuer Port mit dem Namen „BackdoorTTEAVBPHYPort“ umgesetzt. Dieser implementiert das „IPort“Interface von CoRE4INET. Der Port hat einen zusätzlichen Eingang, der alle anderen Protokollprioritäten umgeht.

Dies ist konkret in „Backdoor_TT_AVBClassA_AVBClassB_PCF_RC_8021Q_Shaper“ implementiert. Hier werden die Nachrichten in Abhängigkeit zur Priorität ans MAC-Modul („EtherMacFullDuplex“) weitergeleitet.

Eine Anwendung kann nun eine Nachricht direkt an den „Backdoor“-Eingang des erweiterten Ports senden. Die Nachricht wird dann mit höchster Priorität auf das Übertragungsmedium gelegt.

6.3 Qualitätssicherung

Zur Qualitätssicherung werden zum einen OMNeT++ Funktionalitäten wie „Smoke“-, „Fingerprint“- und „Module“-Tests eingesetzt. Zum anderen wird in einem Vergleich von Simulationen mit

und ohne Erweiterungen die Funktion des Filters und der Einfluss dieser auf die gefilterten Pakete in einem protokollkonformen Szenario ohne Angriff untersucht.

Der „Smoke“-Test ist die einfachste Form des Tests. Hier wird durch kurzes Starten verschiedener Simulationen überprüft, ob diese fehlerfrei laufen.

Beim „Fingerprint“-Test wird der Determinismus ein und der selben Simulationskonfiguration überprüft. Dafür wird über die Simulation ein Fingerabdruck erzeugt. Dieser ist abhängig von allen erzeugten Events in der Simulation. Bei allen weiteren Läufen wird dann überprüft, ob der ursprüngliche Fingerabdruck dem aktuellen entspricht.

Beim „Module“-Test werden einzelne Module mit Eingaben versorgt und deren Ausgaben gegen einen Sollwert verglichen.

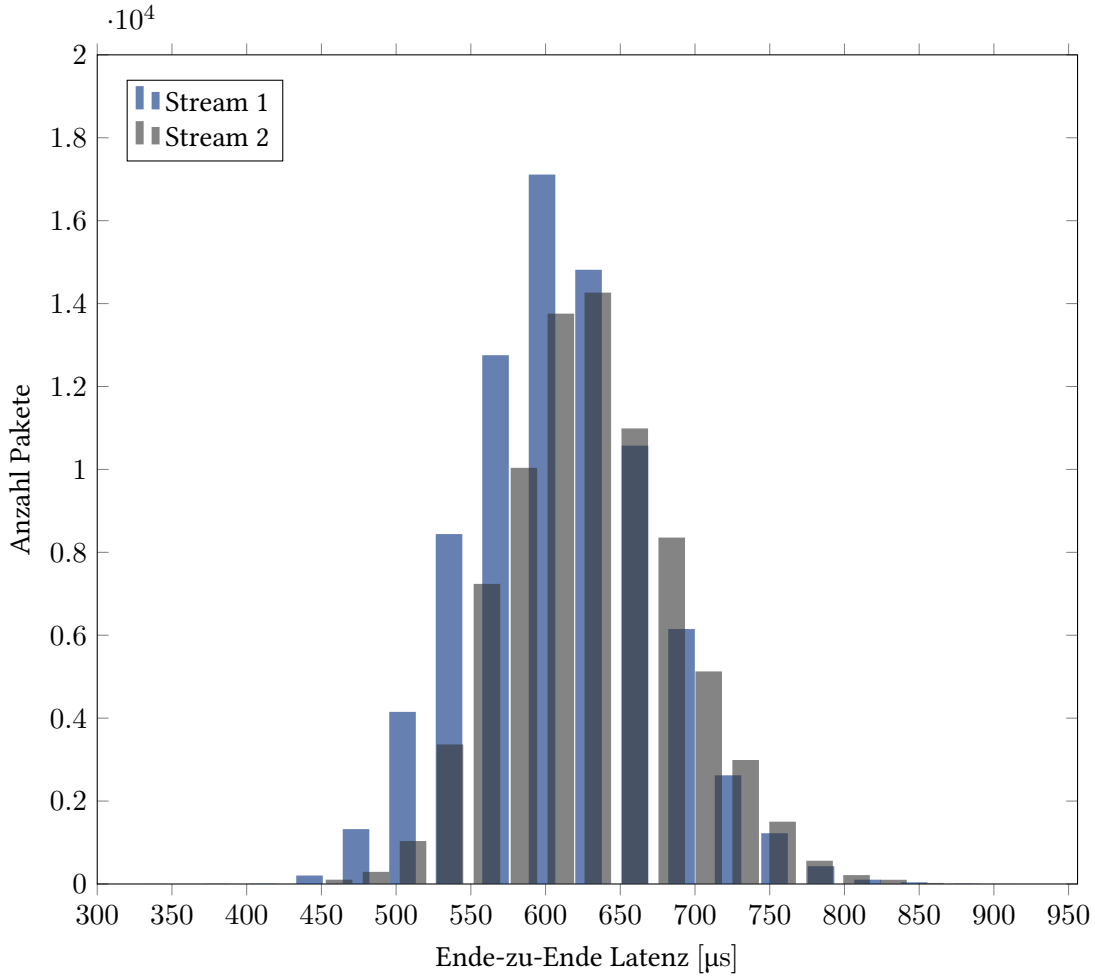

Abbildung 6.6: Ende-zu-Ende Latenz von AVB-Streams ohne Echtzeit-Ethernet-Filter

In dem Netzwerk für den Vergleichstest sind zwei AVB Streams der Klasse A konfiguriert. Diese werden in der Konfiguration mit Filter durch die Implementierung des Echtzeit-Ethernet-Filters und der CBM-Umsetzung überwacht. Des weiteren gibt es zusätzlichen TT- und BE-Verkehr auf den selben Ethernet-Links.

Im Folgendem werden die Ergebnisse des Vergleichstests genauer betrachtet. In den Uhren der Netzwerkteilnehmer sind Ungenauigkeiten konfiguriert, so dass die Ergebnisse unterschiedlicher Simulationskonfigurationen nicht komplett identisch sind.

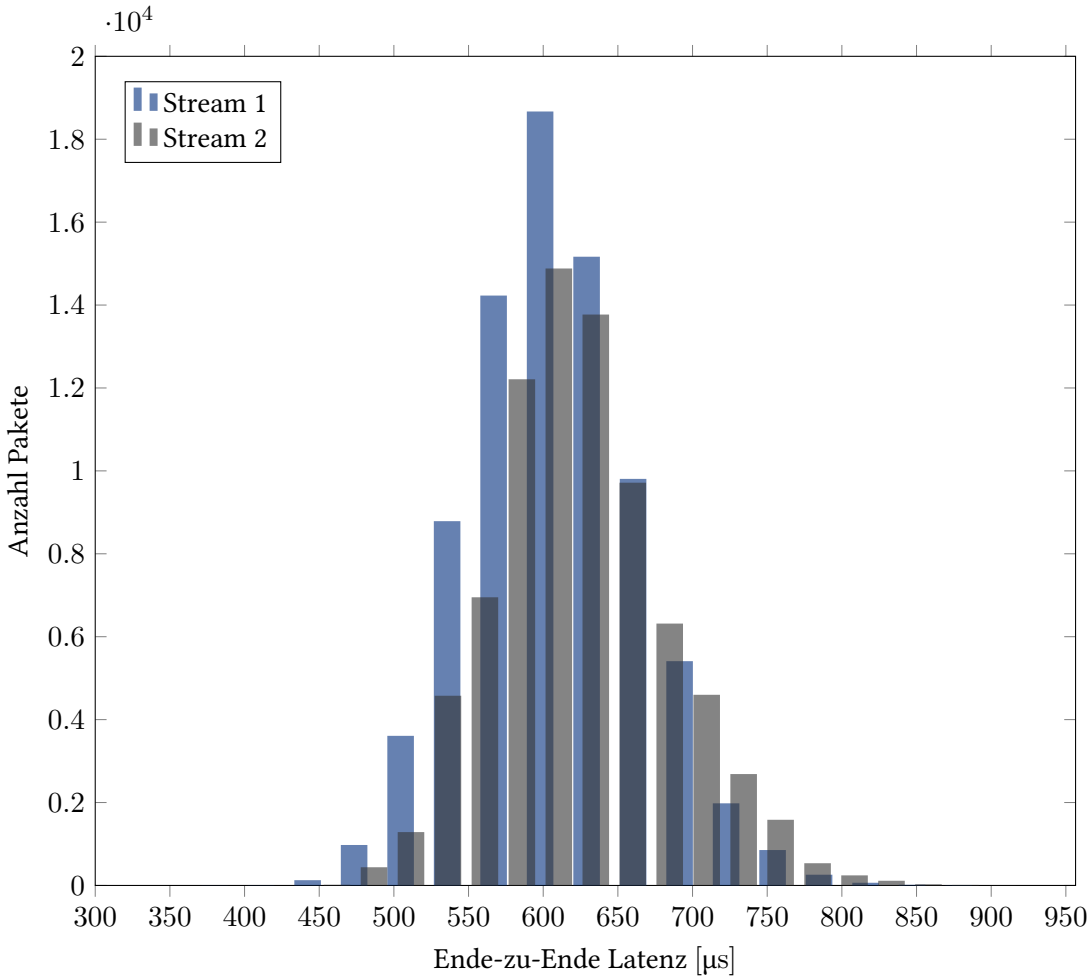

Abbildung 6.7: Ende-zu-Ende-Latenz von AVB-Streams mit Echtzeit-Ethernet-Filter

Die Abbildung 6.6 zeigt ein Histogramm der Ende-zu-Ende-Latenzen der beiden AVB Streams. Die maximale Ende-zu-Ende-Latenz von Stream 1 beträgt 925 µs. Die maximale Ende-zu-Ende-Latenz von Stream 2 beträgt 899 µs.

Im nächsten Schritt werden nun die Erweiterungen genutzt. Alle Netzwerkgeräte nutzen den Filter mit konfigurierten CBM für Stream 1 und Stream 2. Des weiteren wird eine Quelle durch die Umsetzung des kompromittierten Teilnehmers ersetzt. Sie verhält sich jedoch wie ein valider Teilnehmer und fährt keine Angriffe auf das Netzwerk.

In der Abbildung 6.7 ist das Histogramm der Ende-zu-Ende-Latenzen von Stream 1 und 2 mit aktiviertem Echtzeit-Ethernet-Filter abgebildet. Hier beträgt die maximale Ende-zu-Ende-Latenz von Stream 1 922 µs. Die maximale Ende-zu-Ende Latenz von Stream 2 beträgt 948 µs.

Die Differenz der maximalen Latenz beträgt folglich 3 µs für Stream 1 und 49 µs für Stream 2. Auch die Verteilung der anderen Latenz ist gleichwertig. Ein weiterer wichtiger Faktor ist die Tatsache, dass von allen Filter kein einziges Paket verworfen wird. Das protokollkonforme Verhalten wird nicht beeinflusst. Weitere Simulationsergebnisse werden in Kapitel 7 gezeigt.

7 Simulationsbasierte Fallstudie

In diesem Kapitel wird eine simulationsbasierte Fallstudie durchgeführt. In dieser versucht ein Angreifer die Schwachstelle *#AVB.1* auszunutzen. Durch das Fluten mit Nachrichten sollen Nachrichten eines anderen Streams verzögert oder sogar verworfen werden.

Um dies zu vermeiden und die Anforderungen *#AVB.1.1* und *#AVB.1.2* zu erfüllen, wird ein Echtzeit-Ethernet-Filter mit CBM eingesetzt. So soll die Bandbreite des angreifenden Streams begrenzt werden damit der zweite Stream unbeeinflusst am Ziel ankommt.

In Abschnitt 7.1 wird die Topologie des Netzwerks, das simuliert wird, beschrieben. Danach wird in Abschnitt 7.2 die Konfiguration der Filter erläutert und diskutiert. Am Ende folgen in Abschnitt 7.3 die Vorstellung und die Diskussion der Simulationsergebnisse.

7.1 Topologie

In dem simulierten Netzwerk werden TT-Nachrichten, AVB-Streams und BE Nachrichten eingesetzt. Diese passieren den selben kritischen Pfad. Auf diese Weise wird ein Beispiel einer TSN-Netzwerkkonfiguration simuliert.

In der Abbildung 7.1 ist der Netzwerkaufbau abgebildet. Die Teilnehmer, die über TT kommunizieren, sind rot markiert. Blau dargestellt sind die, die über AVB-Streams kommunizieren. Alle Teilnehmer kommunizieren zusätzlich über BE-Nachrichten.

Weitere Eigenschaft der Topologie ist, dass alle Ethernet-Links eine Bandbreite von $100\,\mathrm{Mbit/s}$ haben.

„Node 1“ und „Node 2“ sind die Talker für die jeweiligen Streams 1 und 2. „Node 8“ ist der Listener für beide Streams. Beide Streams sind Klasse-A-Streams mit einer individuellen reservierten Bandbreite von $23.8\,\mathrm{Mbit/s}$.

„Node 1“ ist ein kompromittierter Teilnehmer und kann daher mit seinem Verhalten vom validen Protokollstandard abweichen.

TT-Pakete mit maximaler Länge werden von „Node 3“,„Node 4“,„Node 5“ und „Node 6“ generiert. Das Ziel der ersten beiden Pakete ist „Node 7“. Das Ziel der letzten beiden „Node 9“. Die Switche versenden diese TT-Pakete mit einem Abstand von $123\,\mu\mathrm{s}$. Damit ist zwischen den TT-Paketen ausreichend Platz für die Pakete der Streams.

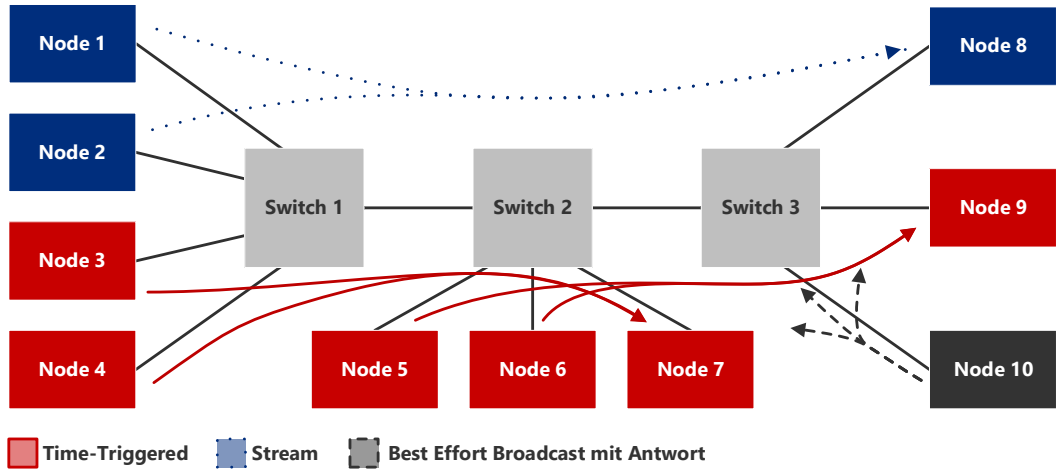

Abbildung 7.1: Netzwerktopologie der Simulation

Hinzu kommt Hintergrundverkehr in Form von BE-Paketen. „Node 10“ versendet über einen Broadcast Pakete mit maximaler Länge an alle anderen Teilnehmer. Diese antworten jeweils mit einem BE-Paket mit maximaler Länge, die wiederum von „Node 10“ empfangen werden.

Die Uhren in allen Geräten sind so konfiguriert, dass sie einer gewissen Ungenauigkeit unterworfen sind.

7.2 Filterkonfiguration

Für die Konfiguration des CBM innerhalb des Filters muss der $Burst_{max}$-Parameter festgelegt werden (s. Abschnitt 5.2.2).

In diesem Fall ist die maximale Anzahl der Pakete innerhalb eines Bursts ($Burst_{out}$) für diese Simulation für jeden Stream-Ausgangsport bekannt. Dieser Wert entspricht pro Stream 2 bei „Node 1“ und „Node 2“ und 4 bei „Switch 1“ und „Switch 2“.

$$Burst_{max_{switch_1 port_0}} = Burst_{out_{node_1}} \qquad +1 = 2 + 1 = 3 \tag{7.1}$$

$$Burst_{max_{switch_1 port_1}} = Burst_{out_{node_1}} \qquad +1 = 2 + 1 = 3 \tag{7.2}$$

$$Burst_{max_{switch_2}} = Burst_{out_{switch_1}} \qquad +1 = 4 + 1 = 5 \tag{7.3}$$

$$Burst_{max_{switch_3}} = Burst_{out_{switch_2}} \qquad +1 = 4 + 1 = 5 \tag{7.4}$$

Daraus folgt ein $Burst_{max}$ Wert von 3 für die CBMs in „Switch 1“ und 5 für die CBMs in „Switch 2“ und „Switch 3“ (s. Gleichung 7.1 bis 7.4).

```
1  [Config Timespan_between_TT_valid_with_Filtering]
2  extends = Timespan_between_TT_valid
3  switch_1.phy[*].filtering.typename = "IEEE8021Qci"
4  switch_1.phy[*].filtering.numStreamFilters = 2
5  switch_1.phy[*].filtering.numStreamGates = 1
6  switch_1.phy[*].filtering.numFlowMeters = 2
7  switch_1.phy[*].filtering.streamFilter[0].streamID = 1
8  switch_1.phy[*].filtering.streamFilter[0].gateID = 0
9  switch_1.phy[*].filtering.streamFilter[0].meterID = 0
10 switch_1.phy[*].filtering.streamFilter[1].streamID = 2
11 switch_1.phy[*].filtering.streamFilter[1].gateID = 0
12 switch_1.phy[*].filtering.streamFilter[1].meterID = 1
13 switch_1.phy[*].filtering.flowMeter[0].typename = "CreditBasedMeter"
14 switch_1.phy[*].filtering.flowMeter[0].maxBurst = 3
15 switch_1.phy[*].filtering.flowMeter[1].typename = "CreditBasedMeter"
16 switch_1.phy[*].filtering.flowMeter[1].maxBurst = 3
```

Listing 7.1: Filterkonfiguration von „Switch 1“ in der Simulation

Das Listing 7.1 zeigt die Konfiguration der Filter für „Switch 1“. Aus Gründen der Vereinfachung werden alle Ports gleich konfiguriert (phy[*]). Im ersten Schritt wird in Zeile 3 die konkrete Filterumsetzung festgelegt. Danach folgen die Parameter dieser Umsetzung.

Für den „IEEE8021Qci“ werden die Anzahl der „Stream Filter“, „Stream Gates“ und „Flow Meter“ benötigt. Für die „Stream Filter“ und „Stream Gates“ werden keine Typen konfiguriert, da der Standardwert bereits „IEEE8021QciFilter“ und „IEEE8021QciGate“ enthält. Daher folgt ab Zeile 7 direkt deren Konfiguration.

Für die „Flow Meter“ werden die Umsetzung des CBM konfiguriert. Der zugehörige $Burst_{max}$-Parameter wird auf den zuvor errechneten Wert 3 gesetzt.

$$\begin{aligned} Credit_{max_{switch_1}} &= |sendslope| * T_{duration} * (Burst_{max} - 1) \\ &\approx 75\,\mathrm{Mbit/s} * 31\,\mu\mathrm{s} * (3 - 1) = 4650 \end{aligned} \tag{7.5}$$

Daraus folgt für die CBMs die in „Switch 1“ operieren ein maximaler Wert für den Credit ($Credit_{max}$) von 4650 (s. Gleichung 7.5)

Listing 7.2 zeigt die gleiche Konfiguration für „Switch 2“. Auch hier wird für die zwei Streams jeweils ein CBM konfiguriert. In diesem Fall wird der $Burst_{max}$ Parameter jedoch auf 5 gesetzt.

```
[Config Timespan_between_TT_valid_with_Filtering]
extends = Timespan_between_TT_valid
switch_2.phy[*].filtering.typename = "IEEE8021Qci"
switch_2.phy[*].filtering.numStreamFilters = 2
switch_2.phy[*].filtering.numStreamGates = 1
switch_2.phy[*].filtering.numFlowMeters = 2
switch_2.phy[*].filtering.streamFilter[0].streamID = 1
switch_2.phy[*].filtering.streamFilter[0].gateID = 0
switch_2.phy[*].filtering.streamFilter[0].meterID = 0
switch_2.phy[*].filtering.streamFilter[1].streamID = 2
switch_2.phy[*].filtering.streamFilter[1].gateID = 0
switch_2.phy[*].filtering.streamFilter[1].meterID = 1
switch_2.phy[*].filtering.flowMeter[0].typename = "CreditBasedMeter"
switch_2.phy[*].filtering.flowMeter[0].maxBurst = 5
switch_2.phy[*].filtering.flowMeter[1].typename = "CreditBasedMeter"
switch_2.phy[*].filtering.flowMeter[1].maxBurst = 5
```

Listing 7.2: Filterkonfiguration von „Switch 2“ in der Simulation

$$\begin{aligned} Credit_{max_{switch_2 switch_3}} &= |sendslope| * T_{duration} * (Burst_{max} - 1) \\ &\approx 75\,\mathrm{Mbit/s} * 31\,\mu\mathrm{s} * (5 - 1) = 9300 \end{aligned} \tag{7.6}$$

Daraus folgt wiederum ein maximaler Credit ($Credit_{max}$) von 9300 (s. Gleichung 7.6) für die CBMs in „Switch 2“.

Gleiches gilt für „Switch 3“. Die Konfigurationen von „Switch 3“ und „Switch 2“ sind identisch.

7.3 Ergebnisse

Für die ersten Untersuchungen werden acht Simulationen durchgeführt. In jeder dieser Simulationen versendet der kompromittierte Teilnehmer „Node 1“ seinen „Stream 1“ mit einer höheren Bandbreite. Die Bandbreitenreservierung von $23.8\,\mathrm{Mbit/s}$ ist in jedem dieser Simulationsläufe identisch. Es ist folglich zu erwarten, dass der erste Filter in „Switch 1“ genau diese Bandbreite zulässt und alle anderen Pakete, die diese übertreffen würden, verworfen werden.

Die Abbildung 7.2 zeigt ausgewählte Ergebnisse dieser Simulationen. In blau ist das Verhältnis von Eingangs- und Ausgangsbandbreite des Filters, der für das Filtern von „Stream 1“ zuständig ist, dargestellt. In schwarz ist die Anzahl der in diesem Filter verworfenen Pakete pro Sekunde dargestellt. Dabei repräsentieren die eingezeichneten Punkte jeweils das Ergebnis eines Simulationslaufs. Die roten Linien markieren die für diesen Stream reservierte Bandbreite.

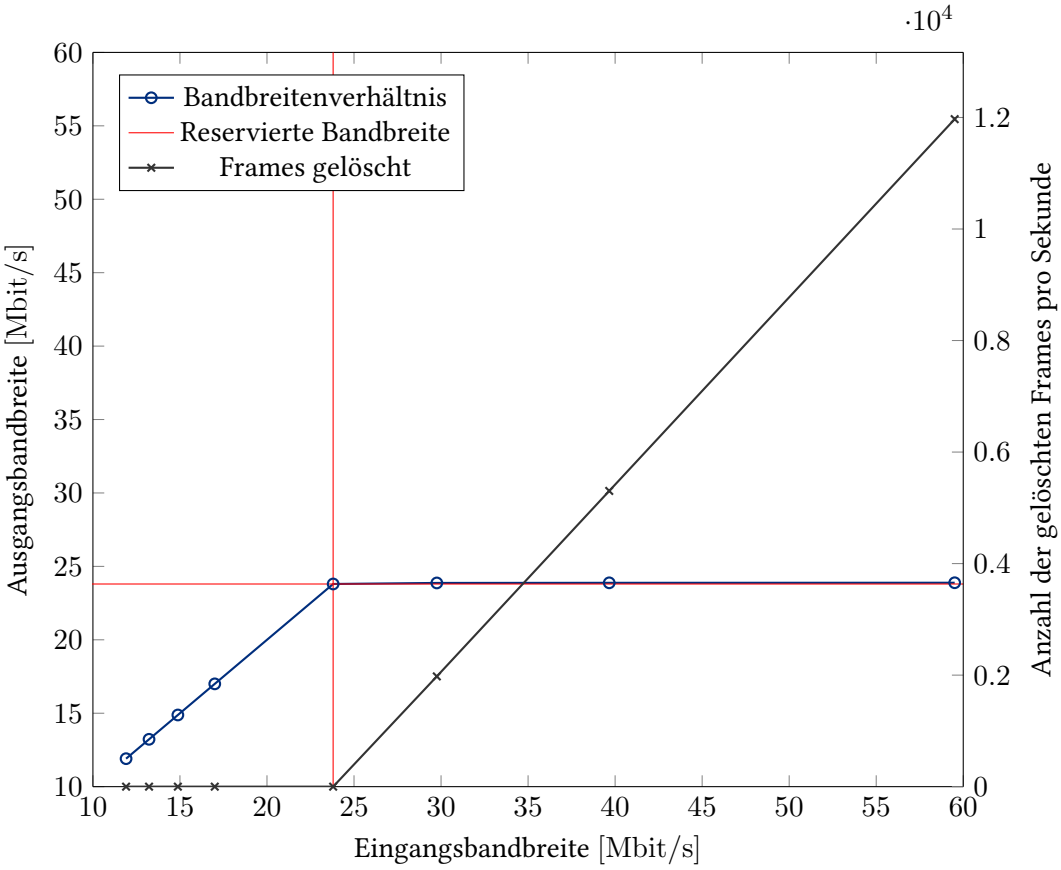

Abbildung 7.2: Einfluss des CBM auf Stream 1 in Switch 1

Es ist sichtbar, dass die Ausgangsbandbreite über den Zeitraum eines Simulationslaufs niemals die reservierte Bandbreite übertrifft. Des weiteren werden, wenn die Eingangsbandbreite der reservierten Bandbreite entspricht, noch keine Pakete verworfen. Bei Erhöhung der Eingangsbandbreite steigt die Anzahl der gelöschten Pakete pro Sekunde linear. Die Ausgangbandbreite bleibt konstant bei $23.8\,\mathrm{Mbit/s}$.

Damit wird deutlich, dass die Anforderung *#AVB.1.1* (s. Abschnitt 4.2.2) vom CBM erfüllt wird. Die Anzahl eingehender Pakete wird über die reservierte Bandbreite für den jeweiligen Stream beschränkt.

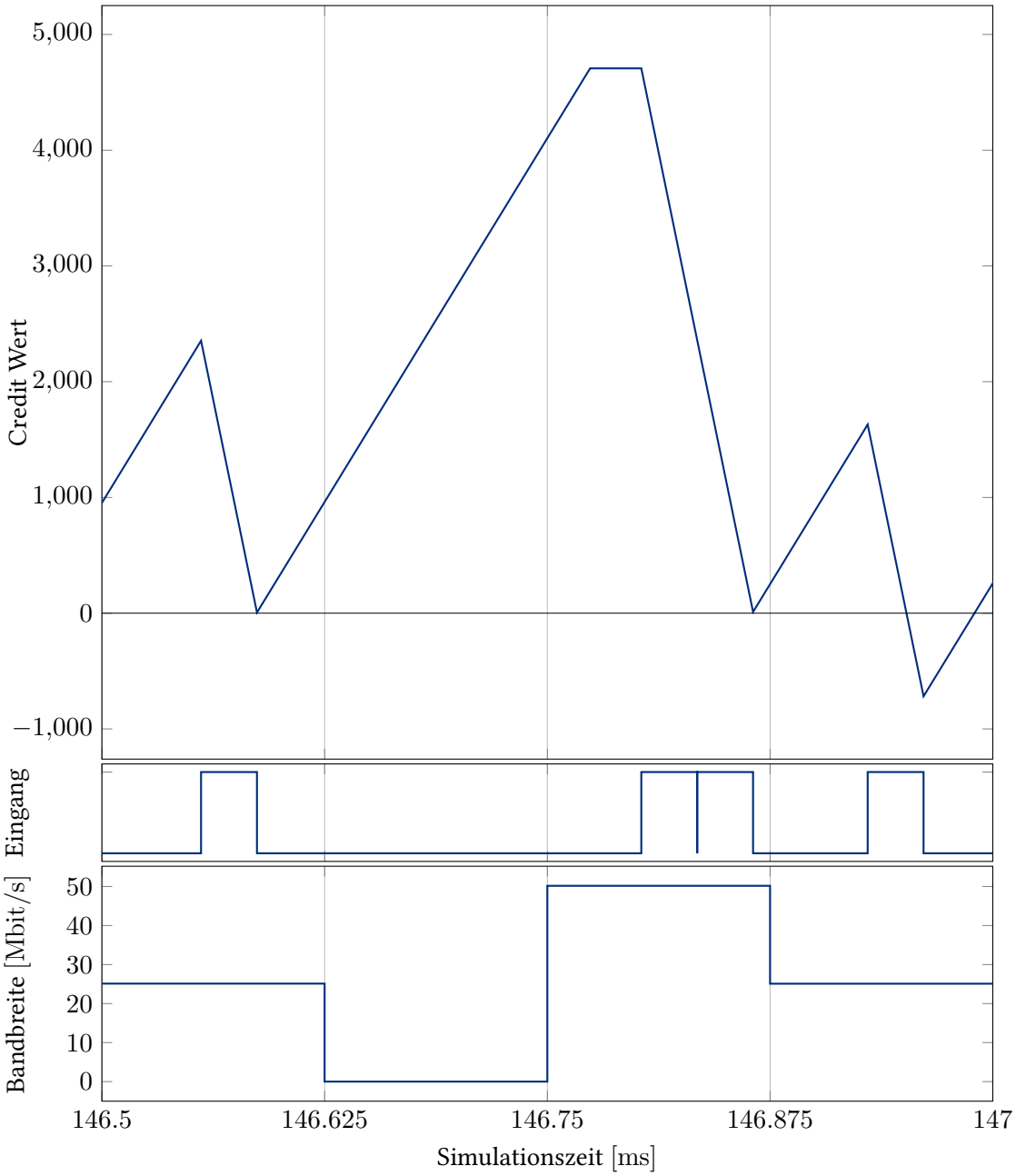

Abbildung 7.3: Ausschnitt der CBM Credit-, Paketeingangs-, und Bandbreitenvektoren

Im nächsten Schritt werden nun eine Ergebnisse einer Simulation betrachtet, die das konkrete Verhalten eines CBMs aufzeichnen. Der CBM-Algorithmus erlaubt das Aufholen von verlorener Bandbreite analog zum CBS. Es ist zu erwarten, dass die Bandbreite ausgehender Pakete über einen kürzeren Zeitraum die reservierte Bandbreite übertrifft.

In der Abbildung 7.3 sind der Vektor für den Wert des Credits, die eingehenden Pakete und die Bandbreite von einem CBM in „Switch 1“ über einen ausgewählten Zeitraum der Simulation zu sehen. Die Bandbreite wird dabei über die Länge eines CMIs der AVB-Klasse A von $125\,\mu s$ (s. Abschnitt 2.1.2) berechnet.

Der beobachtete Stream wird auf der Gegenstelle von einem validen CBS geformt. Die Ausgangsbandbreite an der Gegenstelle entspricht der reservierten Bandbreite von $23.8\,\mathrm{Mbit/s}$.

Zu Beginn am Simulationszeitpunkt $146.5\,\mathrm{ms}$ steigt der Credit an, weil kein Paket im CBM eintrifft. Erst wenn dieses im Eingang eintrifft, wird der Credit für die Übertragungsdauer des Pakets dekrementiert. In diesem CMI trifft kein weiteres Paket ein somit entspricht die Bandbreite der reservierten Bandbreite von $23.8\,\mathrm{Mbit/s}$.

Im nächsten CMI trifft kein Paket ein. Dies kann durch die Interferenz mit anderen Paketen im CBS entstehen (s. Abschnitt 2.1.2). Der Credit steigt darum für die gesamte Dauer des CMIs an.

Am Anfang des dritten CMIs trifft weiterhin kein Paket ein. Der Credit steigt weiter. Dieser erreicht jedoch das Maximum von 4650. Dieser Wert wird nun konstant beibehalten. Dann treffen zwei direkt aufeinanderfolgende Pakete ein. Der Credit wird für die Übertragungsdauer der beiden Pakete dekrementiert. Die Bandbreite für dieses CMI beträgt nun knapp $50\,\mathrm{Mbit/s}$.

Im letzten CMI steigt der Credit wieder an bis ein Paket eintrifft. Dieses reduziert den Credit auf unter 0. Ein weiteres eintreffendes Paket würde nun verworfen werden bis der Credit wieder mindestens 0 erreicht. Die Bandbreite des CMIs entspricht wieder der reservierten Bandbreite.

Dieses Ergebnis zeigt beispielhaft wie der CBM die Anforderung *#AVB.1.2* erfüllt. Je nach Konfiguration des $Burst_{max}$-Parameters sind Bursts erlaubt, die zwischenzeitlich die reservierte Bandbreite übertreffen. Dies jedoch in Abhängigkeit zum vorherigen Verkehr. Es muss eine im Verhältnis passende Pause vor eintreffenden Pakete gegeben haben.

Die nächsten Ergebnisse werfen nochmal einen Blick auf die Ende-Zu-Ende-Latenz der Stream-Pakete wie zuvor in Abschnitt 6.3. Dieses Mal ist „Stream 1“ jedoch kompromittiert. „Node 1“ legt so viele Pakete wie möglich auf den Ethernet-Link ($\approx 100\,\mathrm{Mbit/s}$).

Die Abbildung 7.4 zeigt die Histogramme der Ende-zu-Ende Latenz von „Stream 1“ in blau und „Stream 2“ in grau. Ohne Angriff lagen die maximalen Ende-zu-Ende Latenzen bei $925\,\mu s$ und $899\,\mu s$ (s. Abschnitt 6.3). Mit Angriff liegen sie nun bei $944\,\mu s$ für „Stream 1“ und $933\,\mu s$

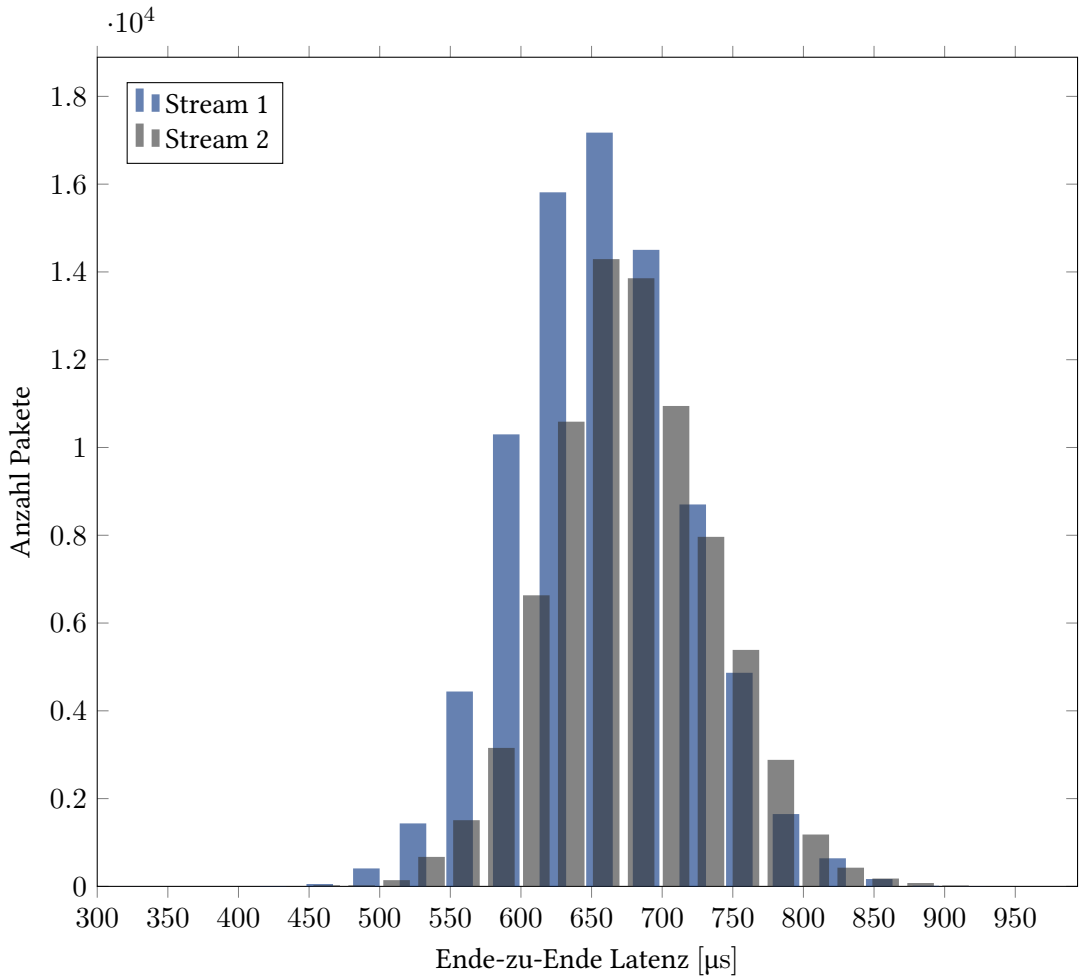

Abbildung 7.4: Ende-zu-Ende Latenz von AVB-Streams mit Echtzeit-Ethernet-Filter während eines DoS-Angriffs

für „Stream 2“. Die Differenzen sind auf Zufälle durch Ungenauigkeiten zurückzuführen. Auch die Verteilung der Latenzen in Vergleich mit den Histogrammen in der Abbildung 6.7 zeigt, dass der Angriff keinen gravierenden Einfluss auf die Ende-zu-Ende Latenz der Pakete hat.

Dafür wurden aufgrund der Beschränkung durch die Reservierung die meisten von „Node 1“ versendeten „Stream 1“ Pakete in „Switch 1“ verworfen. Dieser Stream ist dadurch auch weiterhin kompromittiert und wird mit hoher Wahrscheinlichkeit den Anforderungen des Listeners an anderer Stelle nicht genügen.

Jedoch wird kein einziges Paket von „Stream 2“ verworfen und auch hier sind die Latenzen innerhalb der gleichen Bereiche wie bei validen Verhalten durch „Node 1“. „Stream 2“ bleibt davon unbeeinflusst und kann weiterhin alle Anforderung wie spezifiziert erfüllen.

Die Fallstudie hat gezeigt, dass der CBM die Anforderungen *#AVB.1.1* und *#AVB.1.2* erfüllt. Die Erfüllung der Anforderungen ist sogar unnabhängig vom konfigurierten $Burst_{max}$-Wert. Aus diesem folgen nur die minimalen Puffergrößen und die Höhe von maximaler Ende-zu-Ende-Latenz und Jitter. Ein DoS-Angriff kann den maximalen Burst nur durchs Netzwerk schleusen wenn zuvor verhältnismäßig lange kein Paket gesendet wurde. Nach dem Burst gilt wieder die gleiche Voraussetzung.

8 Fazit und Ausblick

In diesem Kapitel wird die Arbeit mit Fazit und Ausblick abgeschlossen. Dafür wird diese zuerst in Abschnit 8.1 zusammengefasst. Daraufhin werden in Abschnitt 8.2 die Ergebnisse der Arbeit noch einmal präsentiert. Am Ende folgt der Ausblick in Abschnitt 8.3. Hier werden mögliche weiterführende Arbeiten vorgestellt.

8.1 Zusammenfassung

Das heutige System Fahrzeug besteht aus einer Vielzahl von ECUs, die erst im Zusammenspiel funktionssichere Features für den Komfort und die Leistungsfähigkeit bereitstellen. Das Bordnetz, dass diesen ECUs die Kommunikation ermöglicht, ist jedoch nicht informationssicher. Kompromittierte ECUs können die Funktionssicherheit des Systems gefährden.

In Zukunft wird das heutige Bordnetz, bestehend aus Feldbussen, durch Ethernet ersetzt. Um die Anforderungen eines Fahrzeugs an die Kommunikation zu gewährleisten, werden Echtzeit-Ethernet-Protokolle eingesetzt.

In dieser Arbeiten wurde die Informationssicherheit von drei ausgewählten Echtzeit-Ethernet-Protokollkandidaten analysiert. Dabei wurden Schwachstellen entdeckt. Es wurden Anforderungen zum Schließen dieser Schwachstellen erarbeitet und deren Risiko qualitativ bewertet.

Mit Fokus auf Anforderungen, die auf der Sicherungsschicht erfüllt werden können, wurden Konzepte für den Schutz erarbeitet. Das Hauptaugenmerk ist das Filtern von Paketen am Eingang jedes Netzwerkteilnehmers. Ein entwickeltes Konzept ist der CBM, der als Gegenstück zum für Stream-basierten Verkehr genutzten CBS konzipiert ist.

Die Konzepte wurden durch Erweiterung einer bestehenden Simulationsumgebung implementiert. In einer Fallstudie wurde dann diese Erweiterungen genutzt, um das Konzept des CBM-Filters in einer genaueren Betrachtung zu untersuchen.

8.2 Ergebnisse

- Es besteht Handlungsbedarf bei der Sicherheitsanalyse von Protokollen, die auf der Sicherungsschicht für die Kommunikation der Steuergeräte im Fahrzeug zuständig sind.
- Die Hardware der Switche muss vor Manipulation geschützt sein um eine sichere Kommunikation auf der Sicherungsschicht zu ermöglichen.
- Bei den Protokollen TTE, AVB und TSN gibt es Schwachstellen, die geschlossen werden müssen, um die Informationssicherheit zu gewährleisten. Die Auswirkungen der Manipulation von Sicherungsschicht-Protokollen, zum Beispiel für die Zeitsynchronisation oder auch SRP, müssen durch Authentifizierung und Verschlüsselung unterbunden werden.
- Der entwickelte CBM-Algorithmus verhindert wirkungsvoll die Auswirkungen von DoS-Angriffen über CBS-Streams auf das Bordnetz. Die Simulation hat in einem Versuch gezeigt, dass dieses Konzept das Soll erfüllt.
- Das entwickelte MGM-Konzept kann die Auswirkungen von DoS-Angriffe im Hintergrundverkehr (BE) verhindern.
- Für ein dynamisches Bordnetz muss die Konfiguration der vorgestellten Filter möglichst einfach sein. Weiterhin muss im Falle von entfernter Aktualisierung der Systemkomponenten auch die Konfiguration aus der Ferne möglich sein. In diesem Fall muss auch die Aktualisierung der Konfiguration geschützt sein, damit die vorgestellten Schutzkonzepte nicht unterwandert werden können.
- Die Filter definieren ein Normalverhalten der Kommunikation auf der Sicherungsschicht und überwachen dieses verteilt. Die Metriken dieser Filter können genutzt werden, um eine mögliche Anomalieerkennung mit Informationen zu versorgen.

8.3 Ausblick

Aus unterschiedlichen Aspekten dieser Arbeit ergeben sich weiterführende Tätigkeiten.

Im Bereich der Sicherheitsanalyse könnten zum einen weitere Echtzeit-Ethernet-Protokolle untersucht werden. Zum anderen könnte anhand eines realen Prototypen die Risikobewertung quantifiziert werden.

Passende Schutzkonzepte für die Authentifizierung und Verschlüsselung von Protokollen der Sicherungsschicht in Fahrzeugnetzen könnten in weiterführenden Arbeiten zusammengetragen werden. Gleiches gilt für das Thema der sicheren Switche.

Mit weiteren Fallstudien könnten die Auswirkungen von neuen Metern oder dem MGM in der Simulation untersucht werden. Ein weiteres Thema ist die Analyse der Interferenz von verschiedenen Schutzmechanismen auch über OSI-Schicht-Grenzen hinaus.

Die hier untersuchten Schutzmechanismen könnten weiterhin in einem realen Prototyp implementiert und untersucht werden, um zum einen die Algorithmen und zum anderen die entwickelten Simulationsmodelle zu bestätigen.

Literaturverzeichnis

[Buckl u. a. 2012] Buckl, C. ; Camek, A. ; Kainz, G. ; Simon, C. ; Mercep, L. ; Stähle, H. ; Knoll, A.: The software car: Building ICT architectures for future electric vehicles. In: *2012 IEEE International Electric Vehicle Conference*, March 2012, S. 1–8

[Bundesamt für Sicherheit in der Informationstechnik 2017] Bundesamt für Sicherheit in der Informationstechnik: Risikoanalyse auf Basis von IT-Grundschutz / BSI. 2017 (BSI-Standard 200-3). – Standard

[Cain 2015] Cain, Harel: Applying Machine Learning for Anomaly Detection in CAN bus networks. In: *Escar 2015* (2015), S. 1–3

[Checkoway u. a. 2011] Checkoway, Stephen ; Mccoy, Damon ; Kantor, Brian ; Anderson, Danny ; Shacham, Hovav ; Savage, Stefan ; Koscher, Karl ; Czeskis, Alexei ; Roesner, Franziska ; Kohno, Tadayoshi: Comprehensive Experimental Analyses of Automotive Attack Surfaces. In: *USENIX Security* (2011). – URL http://www.autosec.org/pubs/cars-usenixsec2011.pdf

[CoRE Working Group a] CoRE Working Group: *CoRE Researchgroup*. – URL https://core-researchgroup.de/

[CoRE Working Group b] CoRE Working Group: *CoRE Simulation Models for Real-time Networks*. – URL http://sim.core-rg.de

[Deutsches Institut für Normung 2010] Deutsches Institut für Normung: Sicherheit von Maschinen - Allgemeine Gestaltungsleitsätze - Risikobeurteilung und Risikominderung / DIN. 2010 (DIN EN ISO 12100:2010). – Standard

[Eckert 2014] Eckert, Claudia: *IT-Sicherheit Konzepte - Verfahren - Protokolle*. 9., aktualisierte und korr. Aufl. Muenchen : Oldenbourg, 2014

[eeDesignIt] eeDesignIt: *Molex to demo automotive Ethernet backbone solution at CES 2018*. – URL https://www.eedesignit.com/molex-to-demo-automotive-ethernet-backbone-solution-at-ces-2018/

[El-Maghraby u. a. 2017] El-Maghraby, R. T. ; Elazim, N. M. A. ; Bahaa-Eldin, A. M.: A survey on deep packet inspection. In: *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, Dec 2017, S. 188–197

[Haas u. a. 2017] Haas, R. E. ; Möller, D. P. F. ; Bansal, P. ; Ghosh, R. ; Bhat, S. S.: Intrusion detection in connected cars. In: *2017 IEEE International Conference on Electro Information Technology (EIT)*, May 2017, S. 516–519

[Henniger u. a. 2009] Henniger, Olaf ; Apvrille, Ludovic ; Fuchs, Andreas ; Roudier, Yves ; Ruddle, Alastair ; Weyl, Benjamin ; Paristech, Telecom ; Ltci, Cnrs ; Antipolis, Sophia: Security requirements for automotive on-board networks. (2009), S. 641–646. ISBN 9781424453474

[IEEE 802.1 TSN Task Group a] IEEE 802.1 TSN Task Group: *IEEE 802.1 Time-Sensitive Networking Task Group*. – URL https://1.ieee802.org/tsn/

[IEEE 802.1 TSN Task Group b] IEEE 802.1 TSN Task Group: *IEEE 802.1AS-Rev - Timing and Synchronization for Time-Sensitive Applications*. – URL https://1.ieee802.org/tsn/802-1as-rev/

[Institute of Electrical and Electronics Engineers 2008a] Institute of Electrical and Electronics Engineers: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. In: *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)* (2008), July, S. 1–300

[Institute of Electrical and Electronics Engineers 2008b] Institute of Electrical and Electronics Engineers: IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. In: *IEEE Std 802.3-2008 (Revision of IEEE Std 802.3-2005)* (2008), Dec, S. 1–2977

[Institute of Electrical and Electronics Engineers 2009] Institute of Electrical and Electronics Engineers: IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams. In: *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)* (2009), Jan, S. C1–72

[Institute of Electrical and Electronics Engineers 2010] Institute of Electrical and Electronics Engineers: IEEE Standard for Local and Metropolitan Area Networks—Virtual

Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP). In: *IEEE Std 802.1Qat-2010 (Revision of IEEE Std 802.1Q-2005)* (2010), Sept, S. 1–119

[Institute of Electrical and Electronics Engineers 2011] Institute of Electrical and Electronics Engineers: IEEE Standard for Local and metropolitan area networks–Audio Video Bridging (AVB) Systems. In: *IEEE Std 802.1BA-2011* (2011), Sept, S. 1–45

[Institute of Electrical and Electronics Engineers 2014] Institute of Electrical and Electronics Engineers: IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks. In: *IEEE Std 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011)* (2014), Dec, S. 1–1832

[Institute of Electrical and Electronics Engineers 2016] Institute of Electrical and Electronics Engineers: IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic. In: *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q— as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q—/Cor 1-2015)* (2016), March, S. 1–57

[Institute of Electrical and Electronics Engineers 2017] Institute of Electrical and Electronics Engineers: IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 28: Per-Stream Filtering and Policing. In: *IEEE Std 802.1Qci-2017 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv-2015, IEEE Std 802.1Qbu-2016, and IEEE Std 802.1Qbz-2016)* (2017), Sept, S. 1–65

[International Electrotechnical Commission 2014] International Electrotechnical Commission: Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series / IEC. 2014 (IEC 61158-1:2014). – Standard

[Itkin und Wool 2017] Itkin, E. ; Wool, A.: A Security Analysis and Revised Security Extension for the Precision Time Protocol. In: *IEEE Transactions on Dependable and Secure Computing* (2017), S. 1–1. – ISSN 1545-5971

[Koscher u. a. 2010] Koscher, K. ; Czeskis, A. ; Roesner, F. ; Patel, S. ; Kohno, T. ; Checkoway, S. ; McCoy, D. ; Kantor, B. ; Anderson, D. ; Shacham, H. ; Savage, S.: Experimental Security Analysis of a Modern Automobile. In: *2010 IEEE Symposium on Security and Privacy*, May 2010, S. 447–462. – ISSN 1081-6011

[Matheus und Königseder 2015] Matheus, Kirsten ; Königseder, Thomas: *Automotive Ethernet*. Cambridge, United Kingdom : Cambridge University Press, Januar 2015. – ISBN 9781107057289

[Meyer u. a. 2013] Meyer, P. ; Steinbach, T. ; Korf, F. ; Schmidt, T. C.: Extending IEEE 802.1 AVB with time-triggered scheduling: A simulation study of the coexistence of synchronous and asynchronous traffic. In: *2013 IEEE Vehicular Networking Conference*, Dec 2013, S. 47–54. – ISSN 2157-9857

[Open Networking Foundation 2015] Open Networking Foundation: OpenFlow Switch Specification / ONF. URL https://3vf60mmveq1g8vzn48q2o71a-wpengine.netdna-ssl.com/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf, 2015 (ONF TS-025). – Standard

[OpenSim Ltd. a] OpenSim Ltd.: *INET Framework*. – URL https://inet.omnetpp.org/

[OpenSim Ltd. b] OpenSim Ltd.: *OMNeT++ Discrete Event Simulator*. – URL https://omnetpp.org/

[Phieler 2017] Phieler, Stephan: *Misuse Detection für Echtzeit Ethernet basierte Kommunikationssysteme*. Berliner Tor 5, 20099 Hamburg, HAW Hamburg, Masterarbeit, 2017

[Schneier 1999] Schneier, Bruce: Attack trees. In: *Dr. Dobb's journal* 24 (1999), Nr. 12, S. 21–29

[Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee 2011] Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee: *Time-Triggered Ethernet AS6802*. SAE Aerospace. November 2011. – URL http://standards.sae.org/as6802/

[Steiner 2013] Steiner, Wilfried: Candidate security solutions for TTEthernet. In: *Digital Avionics Systems Conference (DASC), 2013 ...* (2013), S. 1–10. – URL http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6712579. ISBN 9781479915385

[The Verge] The Verge: *Car hackers demonstrate wireless attack on Tesla Model S*. – URL https://www.theverge.com/2016/9/19/12985120/tesla-model-s-hack-vulnerability-keen-labs

[WIRED] WIRED: *Hackers Remotely Kill a Jeep on the Highway—With Me in It.* – URL https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

Abbildungsverzeichnis

Tabellenverzeichnis

Symbolverzeichnis

$\vec{P}$	Priorität
$\vec{R}$	Risiko $\vec{R} = EK * \vec{SK}$
$\vec{SK}$	Schadensklassen
B	Maximale Bandbreite eines Ports
$Burst_{max}$	Maximale Anzahl an Paketen innerhalb eines Bursts
$Burst_{out}$	Maximale Anzahl an Paketen innerhalb eines ausgehenden Bursts
$Credit_{max}$	Der maximale Wert den der Credit annehmen darf $Credit_{max} = \lvert sendslope_{cbm} \rvert * T_S * (Burst_{max} - 1)$
EK	Eintrittswahrscheinlichkeitsklasse
FS_S	Maximale Paketgröße eines Pakets innerhalb eines Streams
$idleslope$	Aufwärtsgradient des Credit Based Shaper $idleslope = RB$
$idleslope_{cbm}$	Aufwärtsgradient des Credit Based Meter $idleslope_{cbm} = \sum_{n=1}^{N} RB_{S_n}$
P_F	Priorität Fahrzeug
P_K	Priorität Kommunikation
P_T	Priorität Teilnehmer
R_F	Risiko Fahrzeug
R_K	Risiko Kommunikation

R_T	Risiko Teilnehmer
RB	Reservierte Bandbreite
RB_S	Reservierte Bandbreite eines Streams
$sendslope$	Abwärtsgradient des Credit Based Shaper $sendslope = RB - B$
$sendslope_{cbm}$	Abwärtsgradient des Credit Based Meter $sendslope_{cbm} = (\sum_{n=1}^{N} RB_{S_n}) - B$
SK	Schadensklasse
SK_F	Schadensklasse Fahrzeug
SK_K	Schadensklasse Kommunikation
SK_T	Schadensklasse Teilnehmer
T_S	Maximale Übertragungsdauer eines Paket innerhalb eines Streams $T_S = \frac{FS_S}{B} + T_{ifg}$
T_{ifg}	Ethernet Interframe Gap
T_{Jitter}	Jitter $T_{Jitter} = T_{Latenz_{max}} - T_{Latenz_{min}}$
T_{Latenz}	Latenz

Abkürzungsverzeichnis

AVB Audio Video Bridging. 2–9, 12, 17, 19, 20, 22, 24, 26, 27, 47, 48, 50, 56, 57, 60, 68, 75

BAG Bandwidth Allocation Gap. 5
BE Best-Effort. 4, 5, 7, 8, 22, 24, 25, 27, 32, 36, 38, 48, 50, 51, 60

CAN Controller Area Network. 1, 13, 15, 41
CBM Credit Based Meter. 34–36, 38, 42, 44, 48–56, 58–60, 68
CBS Credit Based Shaper. 7, 8, 10, 20, 21, 26, 34, 36–38, 56, 59, 60
CMI Class Measurement Interval. 6, 7, 56
CoRE Communication over Realtime Ethernet. 12

DoS Denial of Service. 13, 14, 21, 24, 31, 57, 58, 60, 68
DPI Deep Packet Inspection. 33, 38

ECU Electronic Control Unit. 1, 2, 4, 15, 59

IDS Intrusion Detection System. 41
IEEE Institute of Electrical and Electronics Engineers. 6, 7
IFG Interframe Gap. 35, 74
IP Internet Protocol. 5

LIN Local Interconnect Network. 1

MGM Minimum Gap Meter. 38, 60, 61
MOST Media Oriented System Transport. 1

NIDS Network-based Intrusion Detection System. 41

OSI Open Systems Interconnection. 5, 14, 17, 34, 61, 74, 75

PTP Precision Time Protocol. 30

QoS Quality of Service. 4, 20, 24, 25

RC Rate-Constrained. 5, 38

SDN Software Defined Networking. 39
SRP Stream Reservation Protocol. 7, 22–26, 60

TCP Transmission Control Protocol. 6, 34
TDMA Time Division Multiple Access. 5, 19, 25, 75
TSN Time-Sensitive Networking. 2–5, 9, 10, 12, 17, 27, 30, 50, 60
TT Time-Triggered. 5, 19, 20, 48, 50
TTE Time-Triggered Ethernet. 2–5, 9, 12, 17, 19, 24, 27, 30, 33, 38, 60

UDP User Datagram Protocol. 6
UML Unified Modeling Language. 35, 36, 43, 46, 67, 68

Glossar

Anwendungsschicht

Anwendungsschicht (englich: Application Layer) beschreibt die 7. Schicht des OSI-Modells. Diese Schicht stellt Funktionen für Anwendungen zur Verfügung. 34

Backbone

Backbone (deutsch: Rückgrat, Basisnetz) bezeichnet den zentralen Kernbereich eines Kommunikationsnetzes. 2, 13

Best-Effort

Best-Effort (deutsch: größte Anstrengung) bezeichnet die geringste Übertragungspriorität in Kommunikationsnetzwerken. Es gibt keine Garantien für Erfolg und Dauer der Übertragung. 4

Broadcast

Broadcast (deutsch: Rundfunk, Rundruf, Rundspruch) bezeichnet das Senden eines Paketes von einem an alle Teilnehmer eines Netzwerks. 13, 51

Burst

Burst (deutsch: Signalblock, Signalfolge) bezeichnet eine Anhäufung von direkt aufeinander folgenden Nachrichten auf einem Übertragungsmedium. 34, 35, 37, 51

IFG

IFG bezeichnet den minimalen zeitlichen Abstand zwischen Paketen auf einem Übertragungsmedium. 35

Infotainment

Infotainment (zusammengesetzt aus dem englischen information und entertainment) fasst alle Anwendungen und Geräte zusammen, die zur Information oder Unterhaltung der Fahrgäste verbaut sind. 4, 15

Listener

Listener (deutsch: Empfänger, Zuhörer) ist die Senke eines AVB Nachrichtenstroms. 7, 50, 57

Sicherungsschicht

Sicherungsschicht (englisch: Data Link Layer) beschreibt die 2. Schicht des OSI-Modells. Diese Schicht regelt den Zugriff auf das Übertragungsmedium. 17, 33, 38, 41, 59, 60

Talker

Talker (deutsch: Sender, Sprecher) ist die Quelle eines AVB Nachrichtenstroms. 7, 50

TDMA

TDMA ist ein Zeitmultiplexverfahren bei dem Daten von unterschiedlichen Sendern zu jeweils bestimmten Zeitabschnitten versendet werden. 5, 9, 10

Index

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 7. Juni 2018 Philipp Meyer