



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Philipp Meyer

**Simulationsbasierte Analyse der Integration von TDMA
basierter Kommunikation in Ethernet AVB**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Philipp Meyer

**Simulationsbasierte Analyse der Integration von TDMA
basierter Kommunikation in Ethernet AVB**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

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

Betreuender Prüfer: Prof. Dr. Korf
Zweitgutachter: Prof. Dr. Heitmann

Eingereicht am: 1. Oktober 2013

Philipp Meyer

Thema der Arbeit

Simulationsbasierte Analyse der Integration von TDMA basierter Kommunikation in Ethernet AVB

Stichworte

Audio-Video Bridging, Time-Triggered Ethernet, Time Division Multiple Access, Fahrzeugnetze, OMNeT++, INET, TTE4INET

Kurzzusammenfassung

In modereren Fahrzeugen entstehen immer komplexere Kommunikationnetze. Im Fokus zukünftiger Netztechnologien steht darum eine ethernetbasierte Kommunikation. Deshalb wird nach geeigneten Echtzeitprotokollen gesucht. Das Audio-Video Bridging Protokoll ist eines dieser Kandidaten. In dieser Arbeit wird das Verhalten des Audio-Video Bridging Protokolls, mit einer TDMA Erweiterung, untersucht. Dies wird mit der eventbasierten Simulationsumgebung OMNeT++ realisiert. Das TTE4INET Framework wird um AVB Komponenten erweitert um das Time-Triggered Ethernet Protokoll als Repräsentant für den TDMA Anteil zu nutzen. Anhand von ausgewählten Fallbeispielen wird dann das Verhalten, des erweiterten Audio-Video Bridging Protokolls, untersucht.

Title of the paper

Simulation-based Analysis of the Integration of TDMA-based Communication in Ethernet AVB

Keywords

Audio-Video Bridging, Time-Triggered Ethernet, Time Division Multiple Access, In-Vehicle Networks, OMNeT++, INET, TTE4INET

Abstract

In modereren vehicles more complex communication networks are developed. The focus of future network technologies is therefore an Ethernet-based communication. So, there is a demand for suitable real-time protocols. The Audio Video Bridging protocol is one of those candidates. In this work, the behavior of the Audio Video Bridging protocol combined with a TDMA extension was investigated. This is done using the event-based simulation environment OMNeT++. The TTE4INET framework is extended to AVB components. Thus, the time-triggered Ethernet protocol can be used as a representative for the TDMA segment. The behavior of the advanced Audio-Video Bridging protocol is then analyzed by means of selected case studies.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	4
2.1	Die Simulationsumgebung OMNeT++	4
2.2	Audio/Video Bridging	5
2.2.1	Die AVB Wolke und das Link Layer Discovery Protokoll	5
2.2.2	Reservieren von Bandbreite mit dem Stream Reservation Protokoll	7
2.2.3	Queueing und Scheduling	9
2.2.4	Synchronisation	12
2.2.5	Maximale Latenz am Beispiel von Klasse A Nachrichten	13
2.3	Time-Triggered Ethernet	14
2.3.1	Nachrichtenklassen	14
2.3.2	Synchronisation	16
3	Verwandte Arbeiten	17
3.1	Analyse von AVB mit Hilfe von Network Calculus	17
3.2	Möglichkeiten zur Verringerung der AVB Klasse A Latenz	17
3.3	Leistungsanalyse des AVB Standards	18
3.4	Vergleich von AVB und Time-Triggered Ethernet	18
4	Analyse und Anforderung	20
4.1	Konzeptionelle Herausforderungen	20
4.2	Anforderungen	21
5	Konzept	22
5.1	Modifikationen am Queueing	22
5.2	Das Angepasste Credit Based Shaper Scheduling	23
5.3	Erwartete Auswirkungen auf AVB Garantien	25
6	Realisierung der Erweiterung von TTE4INET	28
6.1	Aufbau...	28
6.2	Die implementierten Netzwerkteilnehmer in OMNeT++	36
7	Tests und Fallbeispiele	39
7.1	Testszenarios	39
7.1.1	Smoke Test	39

7.1.2	Fingerprint Test	40
7.1.3	Funktionstest	40
7.2	Fallbeispiele	42
7.2.1	Nicht genügend Platz für AVB Pakete zwischen TT Paketen	43
7.2.2	Ausreichend Platz für AVB Pakete zwischen TT Paketen	46
7.2.3	Paketgrößen von TT und BE variieren	48
8	Fazit und Ausblick	50
8.1	Zusammenfassung und Ergebnisse	50
8.2	Ausblick	51

Abbildungsverzeichnis

1.1	Bordnetz eines VW Phaeton. (Quelle: NYFEGA Elektro-Garage AG)	1
2.1	Genutzte Frameworks.	5
2.2	AVB-Wolke. (Quelle: Weibel (2008))	6
2.3	SRP Kommunikation: Talker Advertise nicht erfolgreich.	7
2.4	SRP Kommunikation: Talker Advertise erfolgreich.	8
2.5	Reservierung eines AVB Streams.	9
2.6	AVB-Queueing.	10
2.7	Verhalten des CBS.	11
2.8	Verhalten des CBS bei Interference mit einer BE Nachricht.	12
2.9	Interaktion von TTE mit Standard Ethernet Protokollen. (Quelle: Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee (2011))	15
5.1	TTE/AVB-Queueing.	23
5.2	Vor einer TT Nachricht wird Platz gelassen um eine Überschneidung zu verhindern.	24
5.3	Neues Worst-Case Szenario.	26
6.1	Klassendiagramm TTEAVBHost.	29
6.2	Klassendiagramm TTEAVBSwitch.	30
6.3	Vererbungshierarchie von IOutputQueue und AVBIncoming.	33
6.4	Vererbungshierarchie von Buffer.	34
6.5	Vererbungshierarchie von IPort und BGTrafficHandle.	34
6.6	Vererbungshierarchie von EtherFrame.	35
6.7	Konzept der geräteinternen Nachrichtenrouten im TTE4INET Framework.	35
6.8	Beispiel eines TTE und AVB fähigen Hosts.	36
6.9	Beispiel eines TTE und AVB fähigen Switches.	37
7.1	Latenz von TT Paketen mit und ohne AVB	41
7.2	Netzwerk für die Fallbeispiele.	42
7.3	Nicht genügend Zeit zur Übertragung von AVB Nachrichten zwischen TT Nachrichten.	44
7.4	Anzahl der Nachrichten in den Puffern von „switch_2“ zu „switch_3“.	44
7.5	Histogramm der Ende zu Ende Latenz bei kompakter TT Konfiguration.	45
7.6	Genügend Zeit zur Übertragung von AVB Nachrichten zwischen TT Nachrichten.	47

7.7	Histogramm der Ende zu Ende Latenz bei gestreckter TT Konfiguration. . . .	47
7.8	Latenzen der AVB Klasse A Pakete bei unterschiedlichen maximalen Paketgrößen der anderen Nachrichtenklassen	49

1 Einleitung

Diese Arbeit ist in der „Communication over Realtime Ethernet“ Arbeitsgruppe (CoRE RG (a)) an der HAW Hamburg entstanden. Diese Arbeitsgruppe befasst sich mit der Realisierung von Echtzeit Ethernet Kommunikation im Auto und Flugzeug.

In modernen Fahrzeugen wird mittlerweile eine Vielzahl an Sensoren und Steuergeräten genutzt um Technologien zu ermöglichen, die die Sicherheit, die Leistung und den Komfort im Automobil der Neuzeit möglich machen. Daraus ergeben sich immer komplexere Kommunikationsstrukturen und erhöhte Datenmengen. Abbildung 1.1 zeigt ein komplexes Netz beispielhaft in einem VW Phaeton. In Zukunft könnte durch den Einsatz von Ethernet die Kommunikation dieser Teilnehmer überschaubarer und leistungsfähiger gestaltet werden.

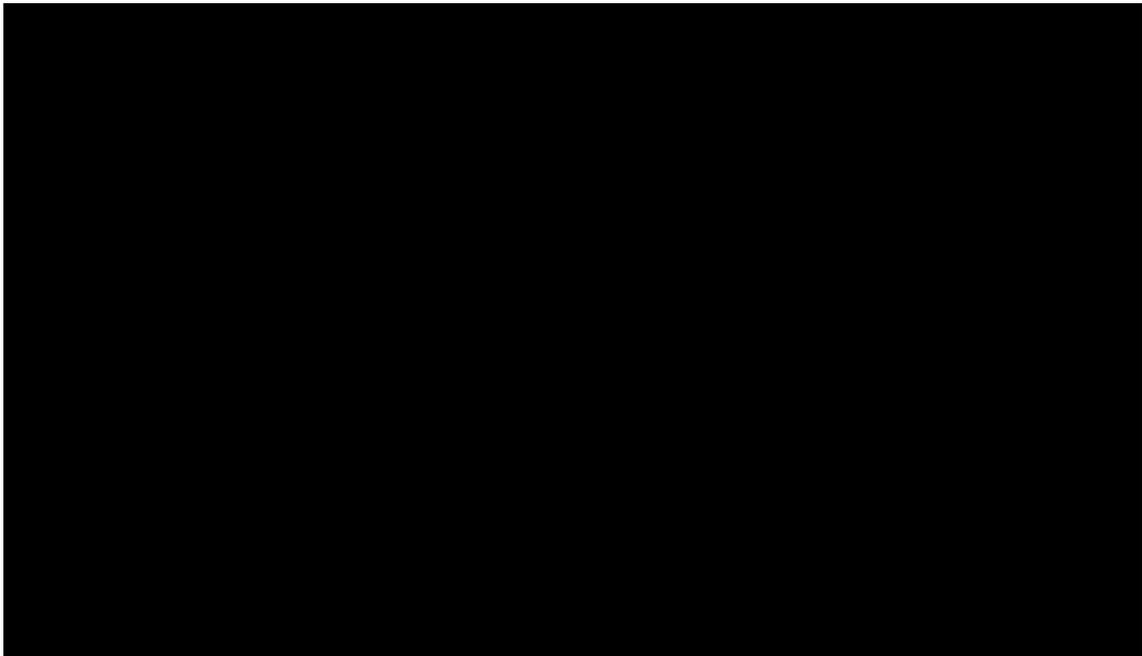


Abbildung 1.1: Bordnetz eines VW Phaeton. (Quelle: NYFEGA Elektro-Garage AG)

Die Echtzeit Anforderungen in diesem Kontext können mit Protokollen ermöglicht werden, die auf das Standard Ethernet IEEE 802.3 aufsetzen und mit diesen deren Funktionalitäten erweitern. Diese benötigen allerdings meistens einen hohen Konfigurationsaufwand. Protokolle wie das Audio/Video Bridging (im weiteren Verlauf auch AVB genannt) minimieren diesen Aufwand.

Die Erweiterung des AVB Standards um TDMA gestützte Kommunikation sind aktuelles Diskussionsthema des IEEE und vieler Automobilhersteller. Neue Ansätze werden in der Time Sensitive Networking (TSN) Task Group des IEEE diskutiert (vgl. [IEEE 802.1 TSN Task Group \(a\)](#) und [IEEE 802.1 TSN Task Group \(b\)](#)). Dabei geht es um die Verbesserungen der Genauigkeit und Latenz von AVB um auch höchst kritische Anwendungen mit Latenzen $< 1\text{ms}$, in der Kommunikation von Steuergeräten, zu ermöglichen.

Das Ziel dieser Arbeit setzt sich aus zwei Teilen zusammen. Im ersten Teil soll das TTE4INET Simulationsframework, welches Teile der Spezifikation von Time-Triggered Ethernet (vgl. [Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee \(2011\)](#)) implementiert, um AVB Komponenten erweitert werden. Im zweiten Schritt wird dann das Zusammenspiel des AVB Protokolls mit der TDMA basierten Kommunikation analysiert.

Auch ein Ansatz der TSN Arbeitsgruppe ist es, eine Time-Triggered Nachrichtenklasse in AVB zu integrieren um mit dieser höheren Anforderungen in der Latenz gerecht zu werden. Darum wird untersucht wie sich AVB mit TDMA basierter Kommunikation zusammenführen lässt und welche Auswirkungen die Kombination auf AVB hat. Außerdem soll eine mögliche Umsetzung implementiert werden. Die Umsetzung soll durch nachvollziehbare Testsszenarien überprüft werden um sicherzustellen, dass die Umsetzung wie geplant und erwartet funktioniert.

In der Analyse sollen Ergebnisse aus der Simulation des erweiterten AVB Protokolls betrachtet und ausgewertet werden. Bestimmte Punkte, wie die Latenz und die Konformität mit den zugrundeliegenden Standards, sind Zielsetzungen dieser Analyse.

Die Arbeit ist wie folgt gegliedert: In Kapitel 2 werden grundlegende Technologien erläutert, die für den weiteren Verlauf der Arbeit notwendig sind und die Basis für die Entwicklung eines Frameworks zur Simulation von AVB mit TDMA Kommunikation bilden. Das Kapitel 3 geht auf verwandte Arbeiten ein, die sich mit Themen beschäftigen, die im Bezug auf diese Arbeit interessante Ansätze liefern. In Kapitel 4 wird die Problemstellung analysiert. Basierend auf diesen Analysen werden Anforderungen an die Umsetzung erstellt. Kapitel 5 beschreibt mögliche Konzepte, die eine Kombination von AVB und Time-Triggered Nachrichten ermöglichen. In dem Kapitel 6 wird auf die Realisierung des Frameworks auf Grundlage von Kapitel 5 eingegangen.

Kapitel 7 beschreibt die Anforderungen und Tests an der fertiggestellten Umsetzung. Anhand von Fallbeispielen wird das Ergebnis der Simulation analysiert. Das Kapitel 8 enthält eine kurze Zusammenfassung der Arbeit sowie Anmerkungen zu möglichen Verbesserungen und Ideen zur Erweiterung des Frameworks. Es werden Schlussfolgerungen aus dem Projekt gezogen und Schwierigkeiten herausgestellt.

2 Grundlagen

In diesem Kapitel werden die Grundlagen für das weitere Verständnis der Arbeit vermittelt. Es wird das verwendete Simulationswerkzeug OMNeT++ und weitere Technologien erläutert die benutzt werden um die Zielsetzung zu erreichen. Dazu zählt die grundlegende Funktionsweise von Audio-Video Bridging und Time-Triggered Ethernet.

2.1 Die Simulationsumgebung OMNeT++

OMNeT++ ist eine Entwicklungsumgebung für Netzwerksimulationen (vgl. [OMNeT++ Community \(b\)](#)). Sie arbeitet eventbasiert und beinhaltet eine modulare, erweiterbare und komponentenbasierte C++ Bibliothek. Mit Hilfe dieser Bibliothek kann das Verhalten von beliebigen Modulen umgesetzt werden. Daraus lässt sich ein, auf die eigenen Anforderungen, zugeschnittenes Netzwerk erstellen und simulieren. So kann man jedes Netzwerkprotokoll umsetzen und untersuchen. Auch deswegen gibt es eine Menge Frameworks, die bereits verschiedene Netzwerkprotokolle und Teilnehmergeräte implementieren. Zwei dieser Frameworks werden in dieser Arbeit verwendet. Das erste ist das INET Framework, welches bereits viele Ethernetmodule implementiert. Auf dieses wird später genauer eingegangen. Das zweite ist das TTE4INET Framework, welches in der CoRE Arbeitsgruppe ([CoRE RG \(a\)](#)) entstanden ist und auf die INET und OMNeT++ Bibliothek aufsetzt. Auch auf dieses wird gleich genauer eingegangen. Des Weiteren ist die Nutzung von OMNeT++ durch die „Academic Public License“ kostenlos möglich, alternativ lässt sich eine kommerzielle Lizenz erwerben. Zusammengefasst sind das die Gründe für den Einsatz dieses Tools für das weitere Vorgehen.

INET Framework

Das INET Framework ist eine für OMNeT++ erstellte Open-Source Bibliothek (vgl. [OMNeT++ Community \(a\)](#)). Es sind bereits viele kabellose und kabelgebundene Switche und Netzwerkteilnehmer implementiert. Diese unterstützen Protokolle wie zum Beispiel: UDP, TCP, SCTP, IP, IPv6, Ethernet und viele weitere. Genutzt wird in dieser Arbeit die Ethernet Komponente. Diese

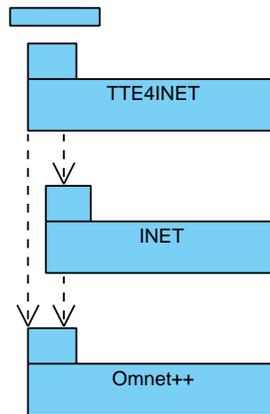


Abbildung 2.1: Genutzte Frameworks.

setzt bereits die MAC Layer, Server, Clients und Switches für die Ethernetkommunikation um. Der Funktionsumfang von OMNeT++ wird so auf Basis der OMNeT++ Bibliothek erweitert.

TTE4INET Framework

Diese Bibliothek (vgl. [CoRE RG \(b\)](#)) nutzt die Implementierung des Ethernet Protokolls vom INET Framework sowie OMNeT++ als Basis (Abb. 2.1). Sie ermöglicht die Simulation der speziellen TDMA basierten Kommunikation „Time-Triggered Ethernet“ (im weiteren Verlauf auch TTE genannt). Dieses Framework ist in der CoRE Arbeitsgruppe entwickelt worden und auch die Grundlage für die neuen AVB Komponenten.

2.2 Audio/Video Bridging

Audio/Video Bridging (AVB) ist der Name für eine Sammlung von Standards, die von der Audio Video Bridging Task Group des IEEE 802.1 entwickelt wurden und in dem IEEE 802.1BA Standard zusammengefasst sind (vgl. [Institute of Electrical and Electronics Engineers \(2011b\)](#)). Diese Spezifikationen erlauben Services über IEEE 802.3 oder Netzwerktechnologien mit ähnlichen Geschwindigkeiten und Fullduplex Betrieb. Diese Services können mit garantierter niedriger Latenz, synchron und priorisiert Daten über ein Netzwerk streamen.

2.2.1 Die AVB Wolke und das Link Layer Discovery Protokoll

AVB garantiert sein Verhalten innerhalb einer begrenzten Wolke aus AVB-fähigen Geräten (Abb. 2.2). Diese Geräte können Endknoten oder Bridges sein. Außerhalb dieser Wolke kann

nur über IEEE 802.3 Ethernet Nachrichten kommuniziert werden. Die minimalen Anforderungen an die physikalischen Ethernetlinks sind 100Mbit/s Übertragungsgeschwindigkeit und Fullduplexbetrieb. Ein AVB-Stream ist eine Datenübertragung in regelmäßigen Intervallen, welche immer unidirektional von einem Talker(Quelle) zu mindestens einem Listener(Senke) verläuft. Zwischen diesen beiden Endknoten dürfen maximal sieben weitere Knoten existieren damit die Garantien eingehalten werden können. Wenn ein AVB-Gerät gestartet wird, tauscht es als erstes Informationen mit seinen Nachbarn aus. Dies geschieht mit Hilfe des in IEEE 802.1AB (vgl. [Institute of Electrical and Electronics Engineers \(2009a\)](#)) entwickelten Link Layer Discovery Protocol (LLDP). Wenn ein Nachbar AVB fähig ist und die minimalen Anforderungen an das Netzwerk erfüllt sind, wird es Teil der AVB-Wolke. Der Vollständigkeit halber mit aufgeführt ist diese Spezifikation im weiteren Verlauf der Arbeit nicht mehr relevant.

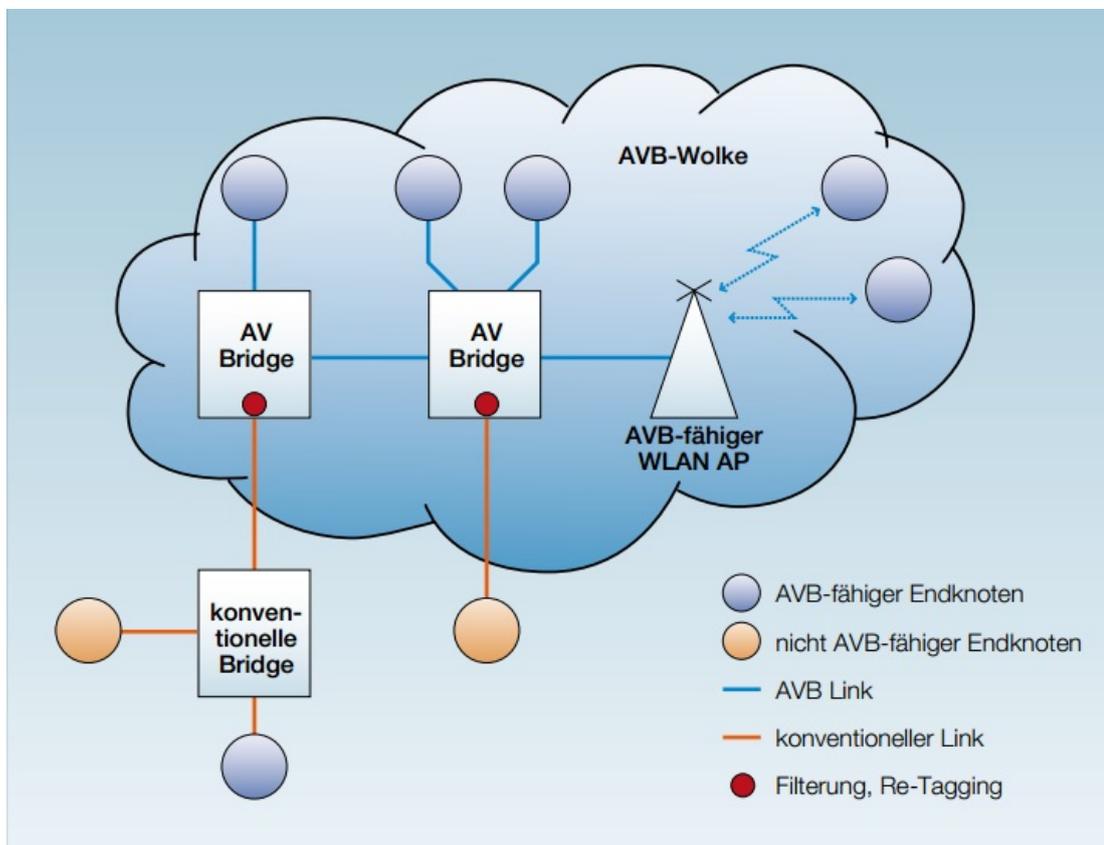


Abbildung 2.2: AVB-Wolke. (Quelle: [Weibel \(2008\)](#))

2.2.2 Reservieren von Bandbreite mit dem Stream Reservation Protokoll

AVB garantiert eine maximale Übertragungslatenz von 2ms. Diese Garantie kann nur dann eingehalten werden, wenn der Pfad zwischen Talker und Listener über die notwendigen Ressourcen verfügt. Dieses wird mit dem Stream Reservation Protokoll (SRP) ermöglicht, das in IEEE 802.1Qat (vgl. [Institute of Electrical and Electronics Engineers \(2010\)](#)) spezifiziert ist. Im ersten Schritt erzeugt der Talker ein „Talker Advertise“ Paket, in welchem die Übertragungsinformationen des eigenen Streams stehen. Diese Informationen beinhalten die maximale Länge der Pakete und wie häufig diese innerhalb eines spezifizierten Intervalls versendet wird (Class Measurement Interval). Daraus kann die maximal durch diesen Stream verbrauchte Bandbreite berechnet werden. Das „Talker Advertise“ Paket wird per Multicast ins Netzwerk AVB fähiger Geräte gesendet. Wenn ein Ausgangsport über genügend Ressourcen verfügt, wird die Nachricht ohne Änderungen auf diesem weiter gesendet. Sollte ein Port nicht über genug Ressourcen verfügen, wird die Nachricht zu einem „Talker Failed“ Paket gewandelt, das zusätzliche Fehlerinformationen enthält. Dieses wird ab diesem Knoten an die restlichen Netzwerkteilnehmer übertragen (Abb. 2.3). Auf diese Weise weiß jeder Listener, ob es möglich ist

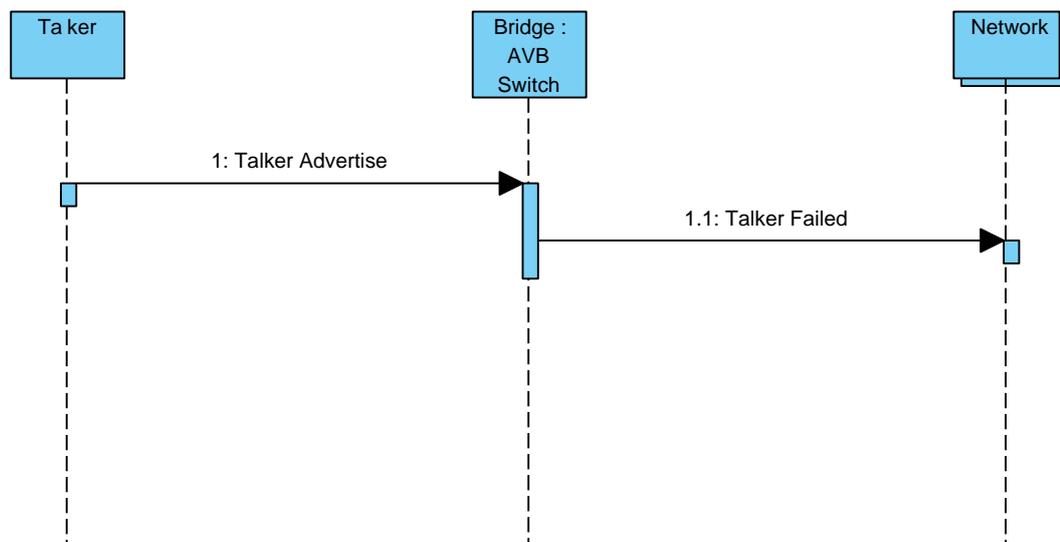


Abbildung 2.3: SRP Kommunikation: Talker Advertise nicht erfolgreich.

Im zweiten Schritt sendet ein Listener, der einen Stream empfangen will, ein „Listener Ready“ Paket an den zugehörigen Talker. Dieser durchwandert die Switches auf dem Weg

zum Talker. Die Switches reservieren in diesem Zug die benötigten Ressourcen. Gelingt dieses nicht, wird die Nachricht in ein „Listener Asking Failed“ gewandelt. Wenn vorher oder nachher von einem anderen Listener schon ein „Listener Ready“ von der momentanen Bridge an den Talker gesendet wurde, wird ein „Listener Ready Failed“ weitergeleitet. Ansonsten wird die „Listener Ready“ Nachricht versendet (Abb. 2.4).

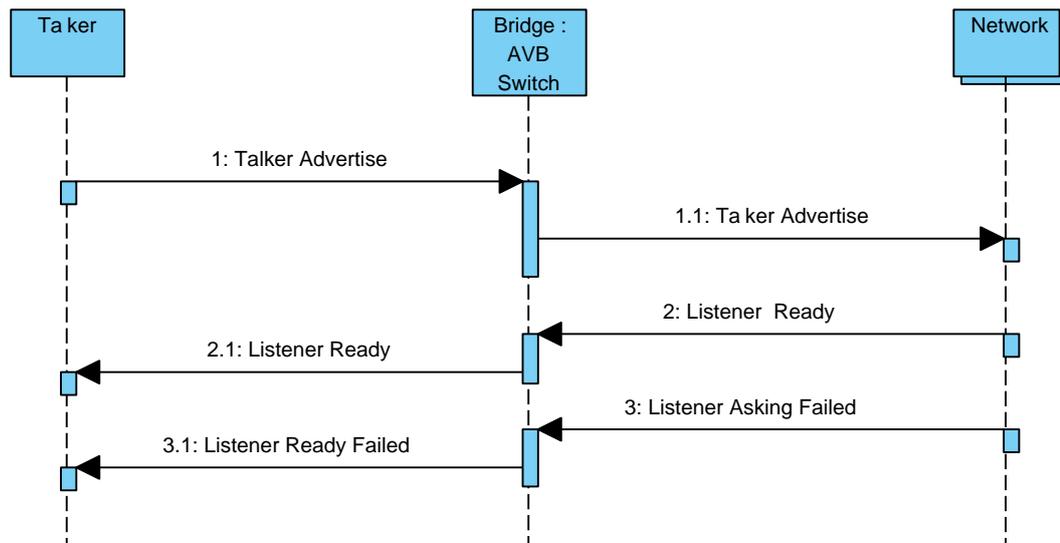


Abbildung 2.4: SRP Kommunikation: Talker Advertise erfolgreich.

Kommt beim Talker ein „Listener Ready“ Paket an, so beginnt dieser mit dem Senden der Streampakete. Die Reaktion des Talkers auf ein „Listener Ready Failed“ ist identisch. Es bedeutet für den Talker das mindestens ein Listener die Route nicht reserviert bekommt aber auch das mindestens ein Talker den Stream empfangen kann. Auf höheren Ebenen kann so nachvollzogen werden, dass nicht alle Listener die Pakete des angeforderten Streams des Talkers bekommen. Bei AVB sollte maximal 75% der zur Verfügung stehenden Ressourcen für AVB Verkehr reserviert werden. Bestehende Reservierungen müssen periodisch aufgefrischt werden, da sie sonst verfallen damit das System nicht unnötig zu blockiert wird, wenn eine sonst nötige Deregistrierung nicht erfolgt ist.

In Abbildung 2.5 ist das Verhalten des SR Protokolls nochmal beispielhaft visualisiert. Es zeigt die Situation nach dem „Talker Advertise“. In dieser wird die „Listener Ready“ Nachricht über die vorher, durch das bekanntgeben des Streams, festgelegte Route zum Talker gesendet.

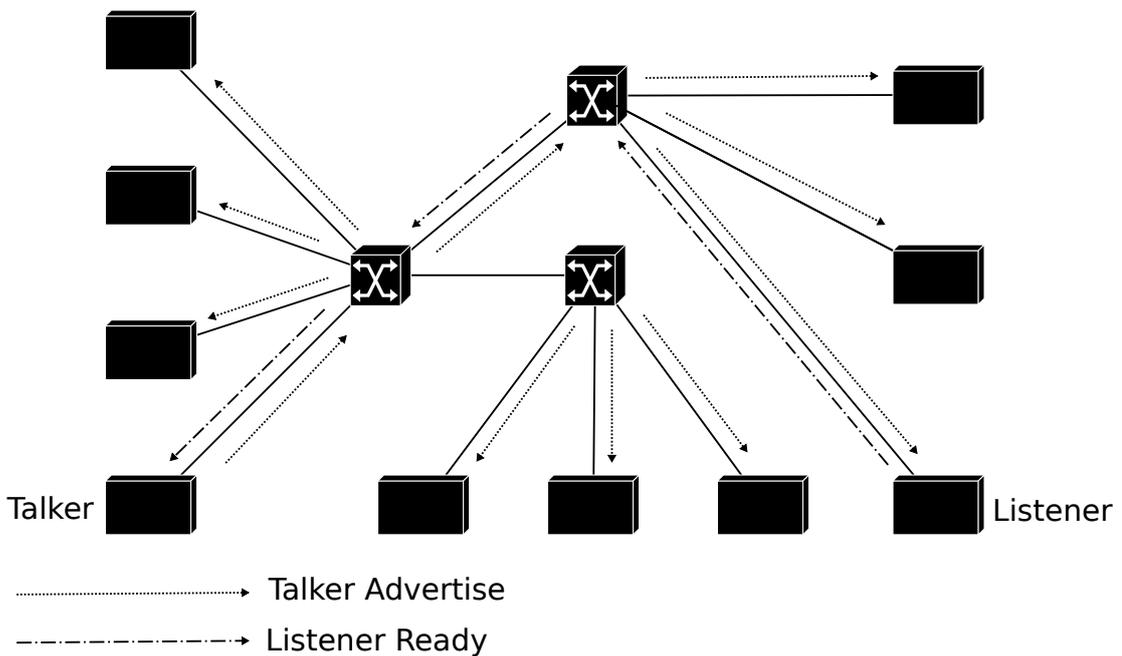


Abbildung 2.5: Reservierung eines AVB Streams.

2.2.3 Queueing und Scheduling

AVB spezifiziert zwei eigene Nachrichtenklassen: Klasse A und B. Klasse A Nachrichten haben die höchste Priorität und garantieren eine maximale Latenz von 2ms. Klasse B Nachrichten sind eine Stufe darunter und garantieren noch eine maximale Latenz von 50ms. Klasse A wird daher eher für kritische Anwendungen genutzt als Klasse B. Wie in IEEE 802.1Qav (vgl. [Institute of Electrical and Electronics Engineers \(2009b\)](#)) beschrieben existiert an jedem Port eine Queue pro Nachrichten Priorität. Es gibt also je einen Puffer für Klasse A, Klasse B und pro BE Priorität an einem Port. Die AVB Nachrichtenklassen A und B haben immer die höchste Priorität im Netzwerk. Die Queues der AVB-Frames werden mithilfe des „Credit Based Shaper“ (im weiteren Verlauf auch CBS genannt) Algorithmus Scheduled. Von allen Frames wird dann, durch einen prioritätenbasierten Algorithmus, nur die Nachricht an den Port geleitet, welcher die höchste Priorität hat (Abb. 2.6).

Das „Class Measurement“ Intervall (im weitem auch CMI genannt) ist das durch den Standard spezifizierte Intervall indem die reservierten Bandbreiten eingehalten werden müssen. Der CBS Algorithmus sorgt dafür, dass innerhalb des CMI diese vorherigen Bandbreitenreservierungen eingehalten werden. Dieses Intervall entspricht $125\mu s$ bei Klasse A und $250\mu s$

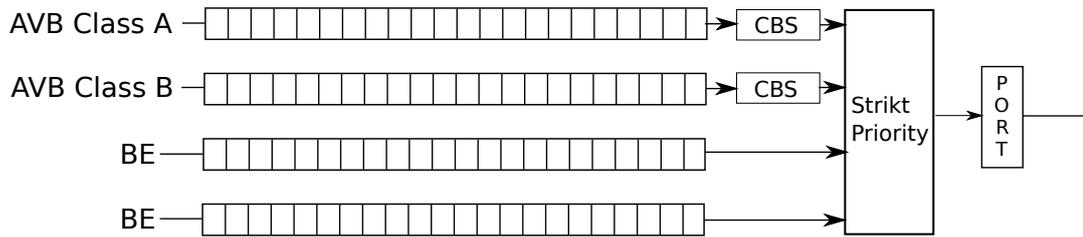


Abbildung 2.6: AVB-Queueing.

bei Klasse B. Dies bedeutet, dass die maximale Größe eines AVB Frames der Klasse A 75% der Bits entspricht, die in $125\mu s$ versendet werden können. Ansonsten würde mehr Bandbreite innerhalb des CMI genutzt als für die Nachrichtenklasse reserviert werden kann, da der AVB Standard wie vorher erwähnt eine maximale Reservierung von 75% empfiehlt. Die folgende Formel zeigt die Abhängigkeit der Paketgröße(S_A), einer AVB Klasse A Nachricht, von der reservierten Bandbreite(RB_A), des CMI(CMI_A) und der physikalisch verfügbaren Bandbreite(B):

$$B * CMI_A * RB_A = S_A \quad (2.1)$$

Der „Credit Based Shaper“ Algorithmus sorgt dafür, dass die vorher durch das „Stream Reservation Protokoll“ reservierten Bandbreiten eingehalten werden. Dies bedeutet, sowohl die nicht Überschreitung als auch die Zusicherung der Allokation am physikalischen Link. In Abbildung 2.7 ist zu erkennen wie der Credit beim Senden dekrementiert wird und danach wieder langsam zurück auf Null steigt.

Die negative Steigung des Credits trägt den Namen „sendSlope“ und die positive „idleSlope“. Ein AVB Paket darf nur weitergeleitet werden, wenn der Credit einen Wert größer oder gleich Null hat. Nach dem Senden wird der „sendSlope“ auf den Credit angewendet. Immer wenn der Credit kleiner Null ist und kein AVB Sendevorgang stattfindet, wird der Wert des Credits mit dem „idleSlope“ wieder bis auf Null gesteigert. Die beiden Steigungen hängen von der, für die jeweilige AVB Klasse, am Port reservierten Bandbreite(RB) ab:

$$sendSlope = RB - B \quad (2.2)$$

$$idleSlope = RB \quad (2.3)$$

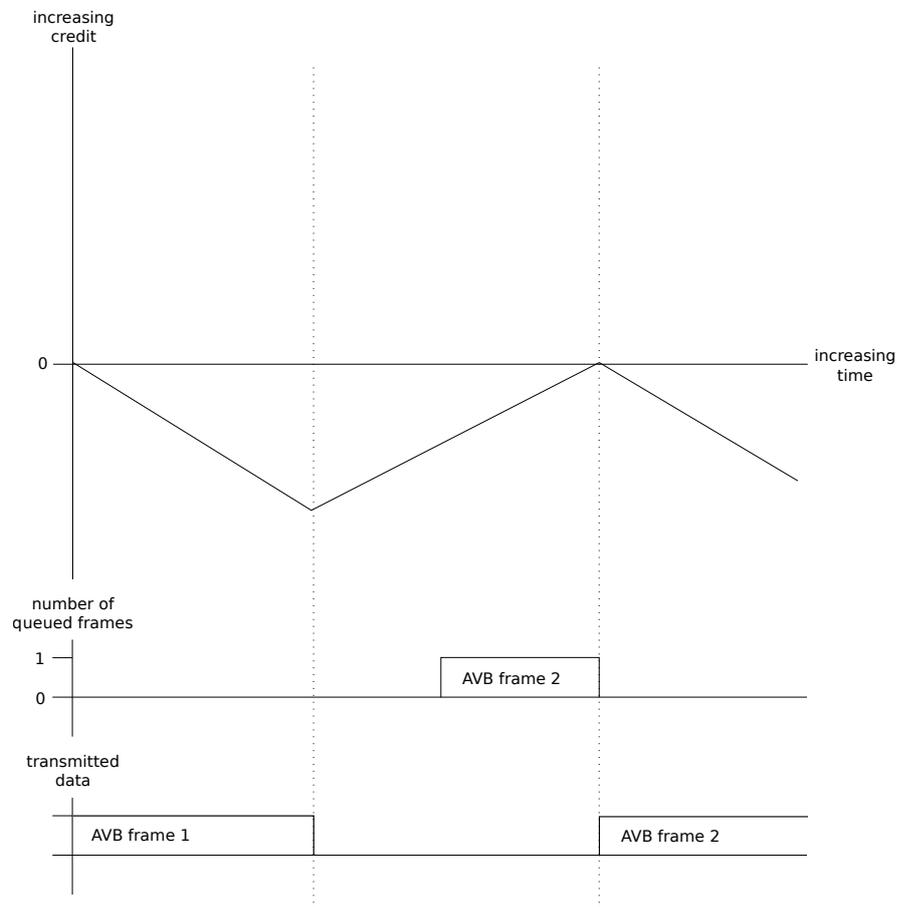


Abbildung 2.7: Verhalten des CBS.

Aufgrund der dahinterliegenden prioritätenbasierten Auswahl des nächsten Frames kommen Best-Effort Nachrichten nur an den Port, wenn keine AVB Nachricht anliegt. Es gibt aber die Situation, dass der Beginn einer BE Übertragung kurz vor dem Anstehen eines AVB Packets liegt. Dieses Szenario wird in der Abbildung 2.8 gezeigt. Um den aus dieser Interferenz resultierenden Rückstand für AVB Nachrichten wieder einholen zu können, steigt der Credit mit „idleSlope“ für die Dauer der Übertragung über Null hinaus. Nur in Szenarien dieser Art steigt der Credit im positiven Wertebereich. Nach Beendigung der Übertragung wird das anstehende AVB Paket versendet. Der Wert des Credits ist nun auch nach dem Versenden der

AVB Nachricht noch positiv. Wenn noch weitere Pakete anstehen würden, könnte das nächste direkt weitergeleitet werden. Da dieses in der Abbildung nicht der Fall ist, wird der Credit auf Null gesetzt.

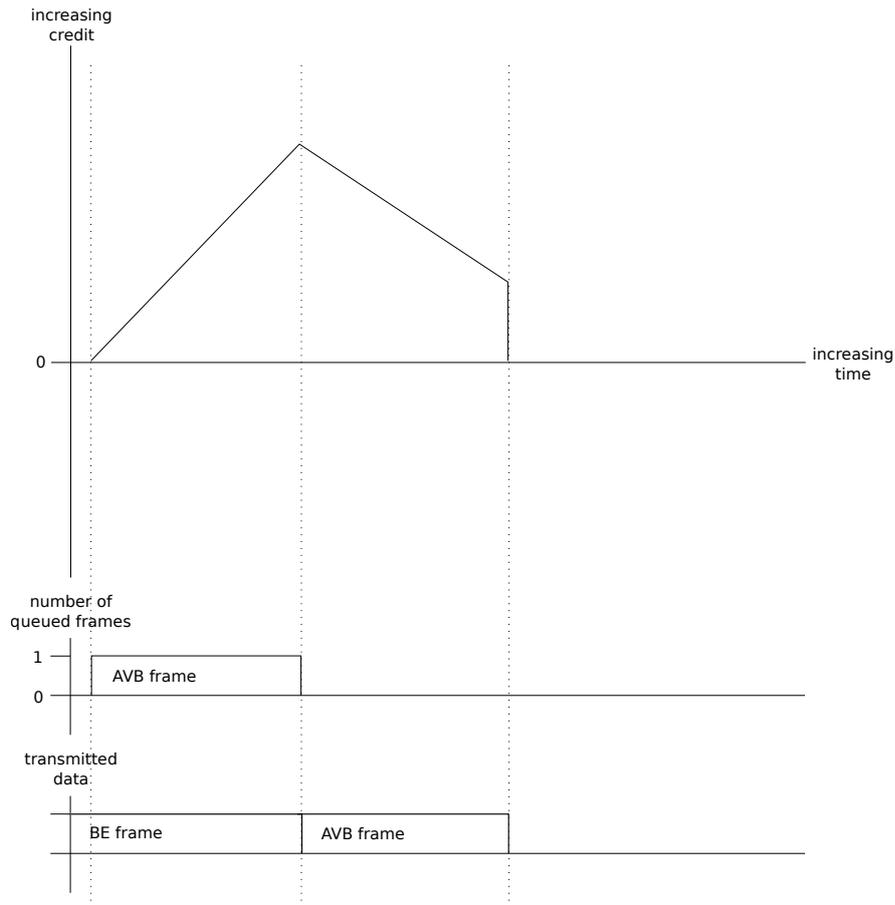


Abbildung 2.8: Verhalten des CBS bei Interferenz mit einer BE Nachricht.

2.2.4 Synchronisation

Die Zeitsynchronisation wird mit Hilfe eines Subsets vom in IEEE 1588 festgelegtem Precision Time Protocol (PTP) ermöglicht. Dieses Subset ist in IEEE802.1AS (vgl. [Institute of Electrical and Electronics Engineers \(2011a\)](#)) definiert. Die zeitliche Synchronisation wird nur auf

Anwendungsebene benötigt um zusammengehörige Streams, wie zum Beispiel dem Ton zu einem Video, an dem Empfänger anzugleichen. Um kostengünstige Hardware zu ermöglichen, wird mit 100 Synchronisationsmeldungen pro Sekunde gearbeitet. Die Synchronisation ist kein Bestandteil im weiteren Verlauf der Arbeit. Darum wird hier auch nicht näher darauf eingegangen.

2.2.5 Maximale Latenz am Beispiel von Klasse A Nachrichten

Die Latenz von AVB über sieben Bridges ergibt sich aus der maximalen Zeit, die ein Schedulingvorgang in einem AVB Gerät benötigt. Im schlechtesten Fall verzögert ein maximales Best Effort Ethernetpaket (M) mit einem Mindestabstand (Interframegap (T_{IFG})) von $0,96\mu s$ die AVB Übertragung auf einer 100Mbit/s Leitung (R) um:

$$T_i = \frac{M}{R} + T_{IFG} = \frac{1530\text{Byte}}{100\text{Mbits}} + 0,96\mu s = 118\mu s \quad (2.4)$$

Zur Gesamtlatenz pro Schedulingvorgang kommt nun noch, dass ein AVB Paket maximal für die Länge des „Class Measurement“ Intervalls durch andere AVB Pakete verzögert werden kann. Dieses Intervall (T_s) ist wie bereits erwähnt $125\mu s$ lang:

$$T_s = 125\mu s \quad (2.5)$$

Beides zusammen ergibt die maximale Latenz pro Schedulingvorgang (pro Netzwerkknoten über den der Stream verläuft):

$$T_{MaxLatenzDevice} = T_i + T_s = 118\mu s + 125\mu s = 243\mu s \approx 250\mu s \quad (2.6)$$

Daraus ergibt sich für acht Schedulingvorgänge (1 Talker + 7 Hops) die garantierte maximale Latenz von AVB:

$$T_{MaxLatenz} = (1 + N_{Hops}) * T_{MaxLatenzDevice} = 8 * 250\mu s = \underline{\underline{2ms}} \quad (2.7)$$

2.3 Time-Triggered Ethernet

Time-Triggered Ethernet (oder auch TTE) ist eine von der Firma TTTech Computertechnik AG entwickelte Netzwerkprotokoll für Echtzeitanwendungen. Diese ist im SAE AS6802 Standard (vgl. [Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee \(2011\)](#)) spezifiziert und unterstützt eine Kombination von synchroner und asynchroner Kommunikation über ein identisches physikalisches Medium das kompatibel zu IEEE 802.3 ist. So werden Anwendungen mit unterschiedlichen Kommunikationsanforderungen an Latenz, Jitter und Zuverlässigkeit, auf Layer 2 Ebene des OSI Modells, über das selbe Netzwerk ermöglicht. TTE folgt dem Prinzip des „Time Division Multiple Access“ (TDMA) um dieses zu erreichen.

2.3.1 Nachrichtenklassen

Time-Triggered Ethernet unterstützt drei unterschiedliche Nachrichtenklassen: Time-Triggered(TT), Rate-Constrained(RC) und Best-Effort(BE). Alle Nachrichtenklassen sind laut Spezifikation voll kompatibel mit Protokollen höherer OSI Schichten, wie zum Beispiel IP und TCP. Des Weiteren nutzen sie den Switched Ethernet Standard IEEE 802.3 zur Nachrichtenübertragung (siehe Abb.2.9).

Time-Triggered Nachrichten

Diese Nachrichtenklasse hat die höchste Priorität. Andere Nachrichtenklassen dürfen nicht gesendet werden, wenn sie sich mit Time-Triggered(TT) Nachrichten überschneiden würden. Darum können TT Nachrichten Pakete anderer aller anderen Klassen verdrängen. Das Grundgerüst für die Funktionalität von TT Nachrichten ist eine, über das gesamte Netzwerk synchronisierte, globale Zeit. Auf Basis dieser Zeit wird offline eine Konfiguration erstellt, die die Sendezeitpunkte und Empfangsfenster der TT Pakete in μs Genauigkeit festlegt. Diese werden dann zyklisch in den Geräten abgearbeitet. Das Absenden einer TT Nachricht wird so durch die Zeit bestimmt. Bei der Konfiguration muss darauf geachtet werden, dass zwei TT Pakete nicht zur selben Zeit über den selben physikalischen Link gesendet werden. Richtig konfiguriert bringt die TT Klasse nun eine deterministische, verlustfreie und synchrone Kommunikation, die durch sonstige asynchrone Kommunikation nicht beeinflusst wird.

Rate-Constrained Nachrichten

Rate-Constrained (RC) Nachrichten haben die zweithöchste Priorität, wobei diese noch in mehrere Prioritäten unterteilt werden kann. Die Übertragung ist unabhängig von der globa-

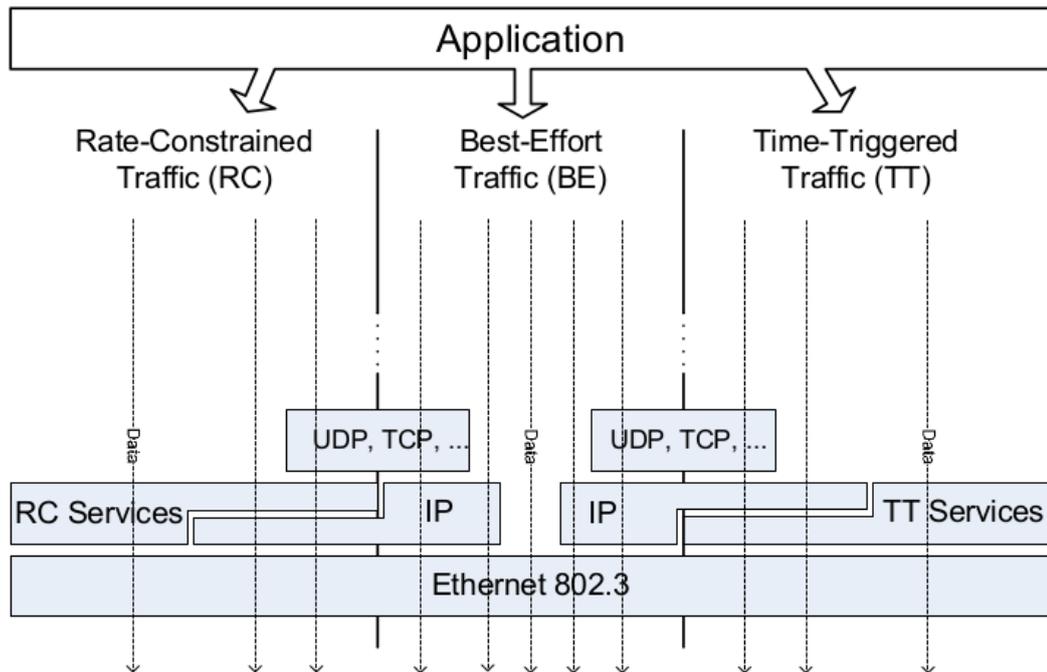


Abbildung 2.9: Interaktion von TTE mit Standard Ethernet Protokollen. (Quelle: [Society of Automotive Engineers - AS-2D Time Triggered Systems and Architecture Committee \(2011\)](#))

len Zeit. Für diese Nachrichtenklasse wird das Routing und die nutzbare Bandbreite offline konfiguriert. Je nach Priorität und Nachrichtenverkehr erhöht sich Jitter und Latenz von RC Nachrichten. Es wird garantiert, dass RC Pakete ihre konfigurierte Bandbreite zur Verfügung haben. Darum lässt sich mit dem Wissen von Übertragungsrate und Switchauslastungen ein verlustfreier Verkehr erzeugen.

Best-Effort Nachrichten

Best-Effort (BE) repräsentiert den Standard IEEE 802.3 Ethernetverkehr. Es gibt keine Garantien für Latenz, Jitter oder, ob ein Paket überhaupt sein Ziel erreicht. Der BE Verkehr nutzt die restliche, durch TT und RC Nachrichten nicht genutzte Bandbreite.

2.3.2 Synchronisation

Wie bereits erwähnt, spielt die Synchronisation bei TTE eine große Rolle, da die Funktion der TT Nachrichtenklasse eine einheitliche globale Zeit erfordert. Es gibt die drei Rollen „Synchronization Master“(SM), „Synchronization Client“(SC) und „Compression Master“(CM). Ein CM ist in der Regel ein Switch. Über ein zwei Wege Protokoll wird die Zeit synchronisiert. Im ersten Schritt bekommt der CM von allen SM's einen „Protocol Control Frame“(PCF) mit der eigenen Zeit. Diese PCF Frames werden in der RC Nachrichtenklasse mit der höchsten Priorität gekapselt. Der CM errechnet dann einen Mittelwert dieser Zeiten. Als zweites wird diese neue Uhrzeit wieder über PCF Pakete an die SM's und SC's gesendet. Diese stellen dann ihre Uhren auf die empfangene neue Uhrzeit.

3 Verwandte Arbeiten

In diesem Kapitel wird auf Arbeiten eingegangen, die sich bereits mit dem Verhalten von Audio/Video Bridging auseinandergesetzt haben. Der Inhalt und das Ergebnis dieser Arbeiten werden kurz zusammengefasst und von den Zielen dieser Arbeit abgegrenzt.

3.1 Analyse von AVB mit Hilfe von Network Calculus

[Queck \(2012\)](#) beschreibt in seinem Papier eine Worst-Case Berechnung von Ethernet AVB mithilfe des „Network Calculus“ Models. Es werden zwei Fallbeispiele aufgeführt in denen anhand eines Netzwerks exemplarisch das Worst-Case Verhalten errechnet wird. In diesem werden drei verschiedene Nachrichtentypen versendet. Diese sind AVB Klasse A, B und Best-Effort.

Das Ergebnis sind bis zu 5,32ms Latenz über 3 Hops für AVB Klasse A Nachrichten und sogar maximal 34,06ms Latenz bei Klasse B. Im Gegensatz dazu hat der BE Verkehr nur eine maximale Latenz von 7,66ms. Diese Ergebnisse würden den im AVB Standard garantierten 2ms über 7 Hops widersprechen.

Abweichend dazu werden die Ergebnisse in dieser Arbeit mit einer Simulation (siehe Kapitel [2](#)) generiert. Zusätzlich wird die Behandlung von TT Nachrichten mit aufgenommen um sehr niedrige Latenzen für hoch priorisierte Pakete zu garantieren.

3.2 Möglichkeiten zur Verringerung der AVB Klasse A Latenz

In der Arbeit von [Imtiaz u. a. \(2012\)](#) werden drei Ansätze zur Reduzierung der AVB Klasse A Latenz diskutiert. Der erste Ansatz handelt von der Idee Pakete mit niedrigerer Priorität unterbrechbar zu machen. Der Zweite versucht das Ziel mit der Verkleinerung von Paketen mit niedrigerer Priorität zu erreichen. Als letztes wird die Möglichkeit diskutiert den Konfliktfall von Klasse A mit anderen Nachrichten zu vermeiden. Für eine Falluntersuchung wird ein AVB ähnliches Shaping mit dem zweiten Ansatz simuliert.

Das Ergebnis zeigt, dass die Latenz der hoch priorisierten Pakete in Abhängigkeit von der Länge eines Paketes mit niedrigerer Priorität oft zu einer besseren Latenz führt als es im Standard AVB der Fall ist.

Im Vergleich ist der Ansatz im weiteren Verlauf dem dritten Ansatz in dem Papier am ähnlichsten. Da durch das TDMA Verhalten Konflikte mit Nachrichten höchster Priorität vermieden werden.

3.3 Leistungsanalyse des AVB Standards

Lim u. a. (2012) entwickeln im Zuge dieses Papiers eine AVB fähige Simulation für OMNeT++ auf Basis des INET Frameworks. Diese Implementation der kompletten AVB Spezifikation wird genutzt um den AVB Standard zu untersuchen. In dieser Fallstudie wird eine Netzwerktopologie genutzt, die einen Vergleich zwischen einer langen und einer kurzen Route ermöglicht. Es werden AVB Klasse A, B und BE Kontrolldatenpakete versendet.

Das Resultat bestätigt die Garantien von AVB. Alle Klasse A Übertragungen haben in jeder Situation eine Latenz unter 2ms. Trotzdem ist die Latenz abhängig von der maximalen Länge dieser Pakete. Interessant ist auch, dass selbst die über BE übertragenen Kontrolldaten gemessene maximale Latenz unter 1ms liegt.

Im Bezug darauf ist es interessant, ob auch in Kombination mit einer TT Nachrichtenklasse diese Latenzgarantie erhalten bleiben kann.

3.4 Vergleich von AVB und Time-Triggered Ethernet

Steinbach u. a. (2012) bewerten und vergleichen AVB und TTE im Kontext von Anforderungen im Netzwerk eines Autos. Dafür wird das selbe Netzwerk mit den zwei unterschiedlichen Netzwerkprotokollen in OMNeT++ simuliert. Einmal mit Hilfe des TTE4INET Frameworks. Als zweites mit dem AVB Modell, dass während der Arbeit, welche in Abschnitt 3.3 zusammengefasst ist, entstanden ist. Im ersten Szenario werden die TT und RC Nachrichtenklassen verwendet. Im zweiten AVB Klasse A und B Nachrichten.

Im Ergebniss erfüllen beide Protokolle die gestellten Anforderungen, wobei darauf verwiesen wird, dass kein Best-Effort Verkehr die Übertragungen gestört hat. In einem solchen Fall wären die Latenzen und der Jitter von AVB und TTE Nachrichten schlechter und bis auf die TT Klasse nicht so vorhersehbar. TT Verkehr schneidet in Hinblick auf Determinismus, Latenz und Jitter am Besten ab. Es wird jedoch eine netzwerkweite statische Konfiguration des komplexen Scheduling benötigt.

3 Verwandte Arbeiten

In dieser Arbeit wird im Gegensatz dazu kein Vergleich von TTE und AVB getätigt, sondern eine Untersuchung der Einbindung der Time-Triggered TTE Nachrichtenklasse in das AVB Protokoll.

4 Analyse und Anforderung

Für die Erweiterung von AVB um TDMA Kommunikation wird der Scheduler, die Synchronisation und die Konfiguration aus der TTE Spezifikation herangezogen. Eine Kombination dieser beiden Protokolle soll zu einem AVB Protokoll führen, das um eine Nachrichtenklasse erweitert wurde. Diese Position nehmen hier die TT Nachrichten ein. Diese Verschmelzung zieht zwangsläufig einige Probleme mit sich. In diesem Kapitel werden diese Probleme analysiert und Anforderungen an das Ergebnis herausgearbeitet.

4.1 Konzeptionelle Herausforderungen

1. Die erste Schwierigkeit ist, dass die Funktionalität von AVB in der Sicherungsschicht des OSI Modells nicht auf eine global synchronisierte Zeit abhebt. Um ein TDMA Verhalten mit TT Paketen zu erzeugen, muss es aber ein netzwerkweites, genaues Zeitverständnis geben. Ansonsten verfällt die hohe Zuverlässigkeit der TT Nachrichtenklasse komplett.
2. Zusätzlich muss die neue TT Nachrichtenklasse auch in der Kombination mit AVB als Klasse mit höchster Priorität interpretiert werden, da Pakete zu fest vordefinierten Zeiten verschickt werden müssen und nicht unterbrochen oder verhindert werden dürfen. Die AVB Spezifikation geht allerdings davon aus, dass die eigenen beiden Nachrichtenklassen A und B die höchste Priorität im Netzwerk haben. Daraus ergibt sich eine Überschneidung der Protokollanforderungen.
3. Anders also zuvor steht dem AVB Protokoll im initialen Netzwerkzustand nicht die gesamte Bandbreite zum Reservieren zur Verfügung. Diese Situation ergibt sich wenn TT Kommunikation konfiguriert ist. Bandbreite, die durch diese Kommunikation genutzt wird, ist für AVB Pakete nicht mehr abrufbar.
4. Darüber hinaus bietet das TTE Protokoll noch die RC Klasse zur Nachrichtenübertragung an, die wie AVB auf Bandbreitenallokierung setzt. In den Ergebnissen von [Steinbach u. a. \(2012\)](#) ist zu sehen, dass AVB Nachrichten bei Latenz und Jitter etwas besser abschneiden.

Zusätzlich ist bei RC Nachrichten statische Konfiguration notwendig. Dieses wird bei AVB durch ein dynamisches Protokoll (SRP) abgefangen.

4.2 Anforderungen

Die herausgearbeiteten Anforderungen beziehen sich auf die Punkte in Abschnitt 4.1.

- Zu 1. In der neuen Kombination muss eine synchronisierte globale Zeit existieren um das Versenden von Time-Triggered Paketen zu ermöglichen. Diese muss den Anforderungen von TTE entsprechen um dessen Garantien von Latenz und Jitter übernehmen zu können. Dafür eignet sich die Synchronisationstechnik der TTE Spezifikation besser, da diese genauer ist und für den Einsatz von TT Paketen konzipiert wurde.
- Zu 2. Des weiteren muss das Scheduling von Nachrichten neu überdacht werden um möglichst alle Vorteile und Garantien der beiden Netzwerkprotokolle zu erhalten. Das bedeutet, dass die TT Nachrichtenklasse weiterhin unabhängig von asynchronem Verkehr eine vorhersehbare Latenz und einen niedrigen Jitter behält. Trotzdem soll AVB Standardkonform bleiben und so wenig Kompromisse wie möglich erfahren. Es muss aber eine Möglichkeit geben AVB Klasse A und B sowie Best-Effort Pakete zu verzögern, wenn der Port für eine bevorstehende TT Übertragung freigehalten werden muss.
- Zu 3. Von der physikalisch verfügbaren Bandbreite sollen weiterhin, wie durch den AVB Standard empfohlen, maximal 75% durch Nachrichtenklassen mit höherer Priorität als Best-Effort belegt werden. Darum muss die durch TT Verkehr verursachte Bandbreite beim Reservieren von Bandbreiten durch das Stream Reservation Protokoll mit einbezogen werden.
- Zu 4. Die RC Nachrichtenklasse soll nicht mit in die neue Kombination übernommen werden, da sich durch die Ähnlichkeiten mit AVB Klasse A und B keine Vorteile ergeben. Rate-Constrained Nachrichten haben ein schlechteres Jitterverhalten, erfordern zusätzliche statische Konfiguration und bringen in diesem Fall keinen Mehrwert für die Zielkombination.

5 Konzept

In diesem Kapitel wird ein Konzept zur Umsetzung der Erweiterung von AVB mit Time-Triggered Nachrichten erarbeitet. Im ersten Teil werden Konzepte entwickelt, die die erschlossenen Anforderungen aus dem vorherigen Kapitel 4 erfüllen. Danach wird eine Auswahl getroffen, welche Elemente für die konkrete Umsetzung genutzt werden sollen.

Für das gesamte Konzept gelten bestimmte Regeln, die bei der Konfiguration eines Netzwerks zusätzlich zur TTE und AVB Spezifikation beachtet werden müssen. Diese sind Grundlage für die weiteren Überlegungen:

- Die Bandbreite, die TT Übertragungen auf dem Kabel verbrauchen, muss angegeben werden, damit AVB diese während der dynamischen Verteilung einbezieht. Dabei muss das Szenario beachtet werden, in dem der Port für eine anstehende TT Übertragung freigehalten werden muss.
- Synchronisationsnachrichten werden bei TTE über die RC Klasse mit der höchsten Priorität versendet. Da es keine RC Nachrichten gibt und diese Synchronisationspakete zu fest vordefinierten Zeitpunkten versendet werden, müssen diese wie TT Nachrichten interpretiert werden.

5.1 Modifikationen am Queueing

Der Aufbau für die Übermittlung von Paketen an den jeweiligen Port erfährt im Vergleich zum AVB Standard leichte Veränderungen und wird erweitert. Die Abbildung 5.1 zeigt den neuen Aufbau. Das Queueing von ausgehenden Paketen wird um die Virtuellen Puffer für die Links von Time-Triggered Nachrichten bestückt. Die Versandzeitpunkte werden durch einen Scheduler bestimmt, der diese durch die statische Konfiguration kennt. Aus diesem Grund übernehmen die TT Nachrichten die höchste Priorität. Zu so einem Versandzeitpunkt darf keine andere Nachricht auf dem Port verschickt werden. Die Zeitpunkte sind für den angepassten CBS Algorithmus transparent. Zusätzlich rückt der CBS in der Übermittlungsreihenfolge hinter die statische Auswahl des nächsten Frames. Dieser wirkt sich nun nunoch

auf die verschiedenen Prioritäten von BE Nachrichten aus. Auf diese Weise erhält der CBS die Möglichkeit, in Abhängigkeit von der TT Konfiguration, AVB und Best-Effort Pakete zu verzögern.

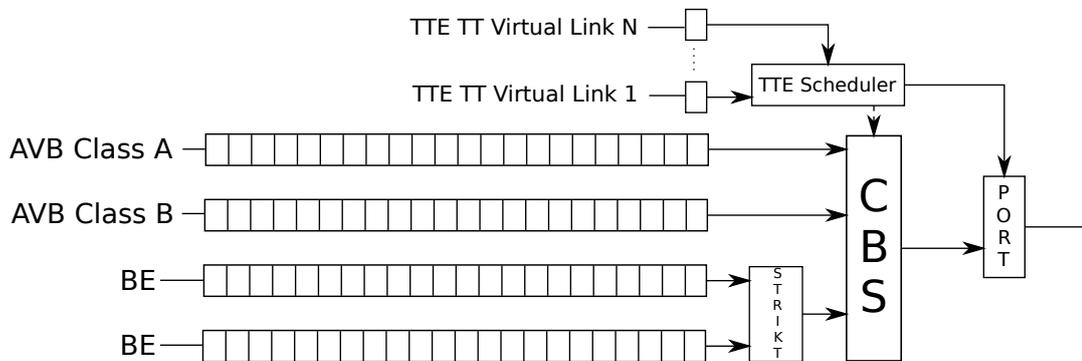


Abbildung 5.1: TTE/AVB-Queueing.

5.2 Das Angepasste Credit Based Shaper Scheduling

Der CBS Algorithmus wird um Eigenschaften erweitert, die das auf Bandbreiten basierte Konzept trotz der Time-Triggered Nachrichtenklasse erhält. Dazu zählt zum Beispiel das Aufzeichnen von Rückständen bei, durch TT Pakete, verursachten Wartezeiten.

Die Abbildung 5.2 zeigt beispielhaft das neue Verhalten des CBS Algorithmus. Der „TTE Send Window Start“ Zeitpunkt ist dem CBS bekannt. Es wird vor jedem Versenden eines Pakets berechnet, ob die Übertragung bis zu diesem Zeitpunkt möglich ist. Wenn dieses nicht der Fall ist und ein AVB Paket in der Queue ansteht, wird dieses verzögert um nicht mit der TT Nachricht zu interferieren. Auch Pakete niedrigerer Priorität, wie Klasse B oder Best-Effort Nachrichten, werden nicht versandt. Sobald diese Situation eintritt, wird der Credit um die Steigung „idleSlope“ inkrementiert bis die Übertragung des TT Pakets abgeschlossen ist. Wenn das Versenden des AVB Pakets daraufhin möglich ist, wird der „sendSlope“ beim Senden vom Credit abgezogen. Auf diesem Bild nicht zu sehen ist, dass weiterhin zusätzlich der Worst Case Fall mit BE Nachrichten des Standard AVB existiert.

Die Latenz und der Jitter des TT Verkehrs sind weiterhin unabhängig vom restlichen asynchronen Paketfluss. Alle niedrigeren Prioritäten überlassen TT Paketen das Sendefenster. In diesem Zusammenhang ergeben sich drei verschiedene Ansätze, die unterschiedliche Auswirkungen auf die Latenz von AVB Nachrichten haben:

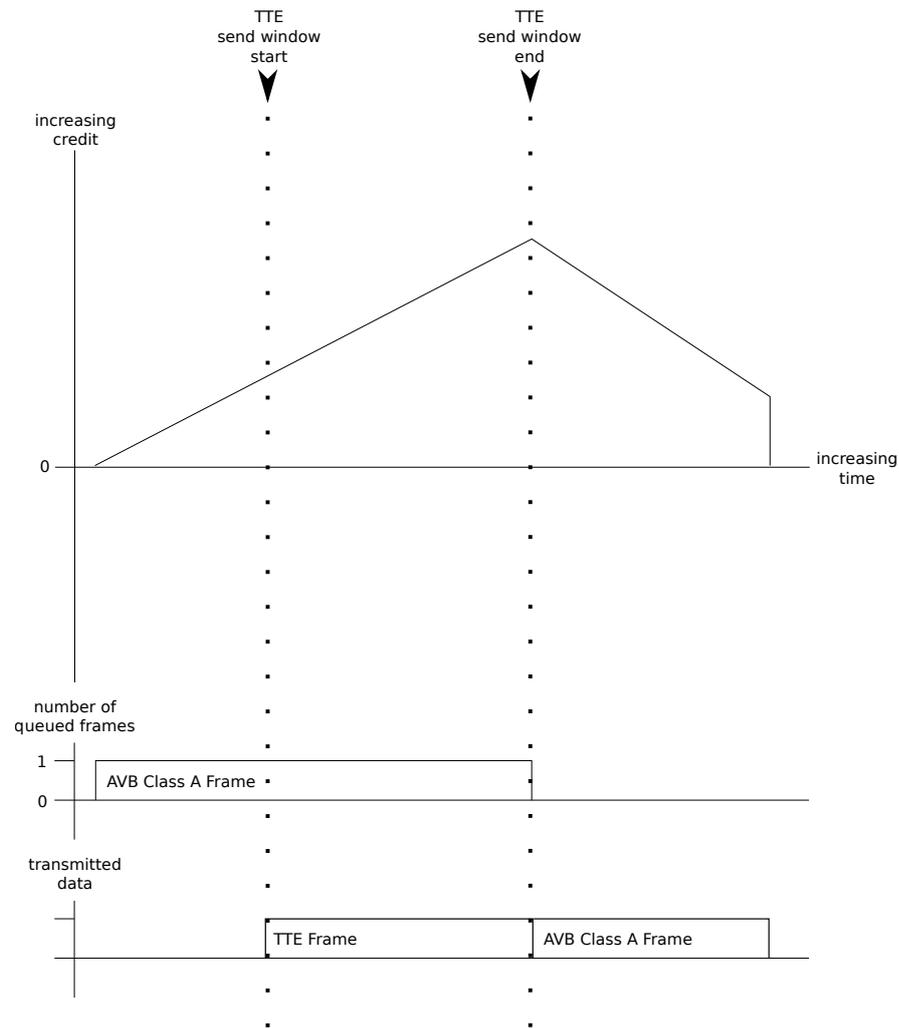


Abbildung 5.2: Vor einer TT Nachricht wird Platz gelassen um eine Überschneidung zu verhindern.

- **Auf Kosten von Latenz AVB Pakete verzögern:**

Eine Möglichkeit ist die Latenzgarantien von AVB in der Kombination mit TDMA Kommunikation herauf zu setzen. Diese Erhöhung wird durch das zusätzliche warten von AVB auf TT Pakete notwendig. Hierdurch wird die Latenzgarantie aus dem AVB Standard zwar überschritten, aber es ist weiterhin möglich ohne Anpassungen mit anderen Standard AVB Geräten zu kommunizieren.

- **Größe von Paketen begrenzen:**

Durch die Begrenzung der Länge von TT, AVB, und BE Paketen wird die Interferenz und Übertragungsdauer vermindert. Daraus resultieren bessere Latenzen und Jitter. Allerdings ist die Unterstützung von Standard AVB Geräten im selben Netzwerk nur noch eingeschränkt möglich.

- **BE Nachrichten für AVB Pakete unterbrechen:**

Präemption ist eine Möglichkeit um zumindest die Verzögerung von AVB Nachrichten durch Best-Effort Verkehr zu verhindern. Dadurch sinkt Latenz und Jitter von AVB Nachrichten. Allerdings nimmt die Belastung des Netzwerkes zu, da unterbrochene BE Pakete erneut verschickt werden müssen. Durch die Einführung einer Grenze, ab der sich ein solches Paket nicht mehr unterbrechen lässt, kann diese Belastung wieder fließend auf Kosten von Latenz und Jitter verringert werden.

Für den weiteren Verlauf wird der erste Ansatz gewählt. Er erfüllt den Standard am Besten und es lassen sich in der Netzwerkkonfiguration trotzdem die Paketgrößen so festlegen, dass sie die Bedürfnisse an das jeweilige Netzwerk erfüllen. Des Weiteren gibt es hier keinen unvorhergesehenen Overhead durch BE Nachrichten. So lässt sich die Auslastung des Netzwerkes präziser vorhersagen.

5.3 Erwartete Auswirkungen auf AVB Garantien

Wie bereits beschrieben verändern sich durch das angepasste Scheduling Latenz und Jitter von AVB Paketen. Dadurch werden auch die Garantien für Klasse A und B Nachrichten beeinträchtigt. Das neue Worst-Case Szenario, von Klasse A Paketen, im CBS wird in [Abbildung 5.3](#) gezeigt. Zunächst trifft eine BE Nachricht mit maximaler Länge ein und wird versandt weil kein AVB Paket ansteht und der Credit kleiner/gleich 0 ist. Kurz nach dem Beginn des Sendens der BE Nachricht kommt ein AVB Paket in der dazugehörigen Queue an und kann nicht versendet werden, da der Ausgangsport blockiert ist. Nach Beendigung der Übertragung reicht die Zeit bis zum „TTE send window start“ nicht mehr aus um das AVB Paket vollständig zu übertragen. Es wird deswegen weiter verzögert bis das Senden des TT Pakets abgeschlossen ist. Erst jetzt darf das AVB Paket versendet werden. Wenn noch andere AVB Pakete anstehen, wird der Bandbreitenrückstand über den Credit notiert, aber die Latenz dieses Pakets kann bis zum Maximum steigen.

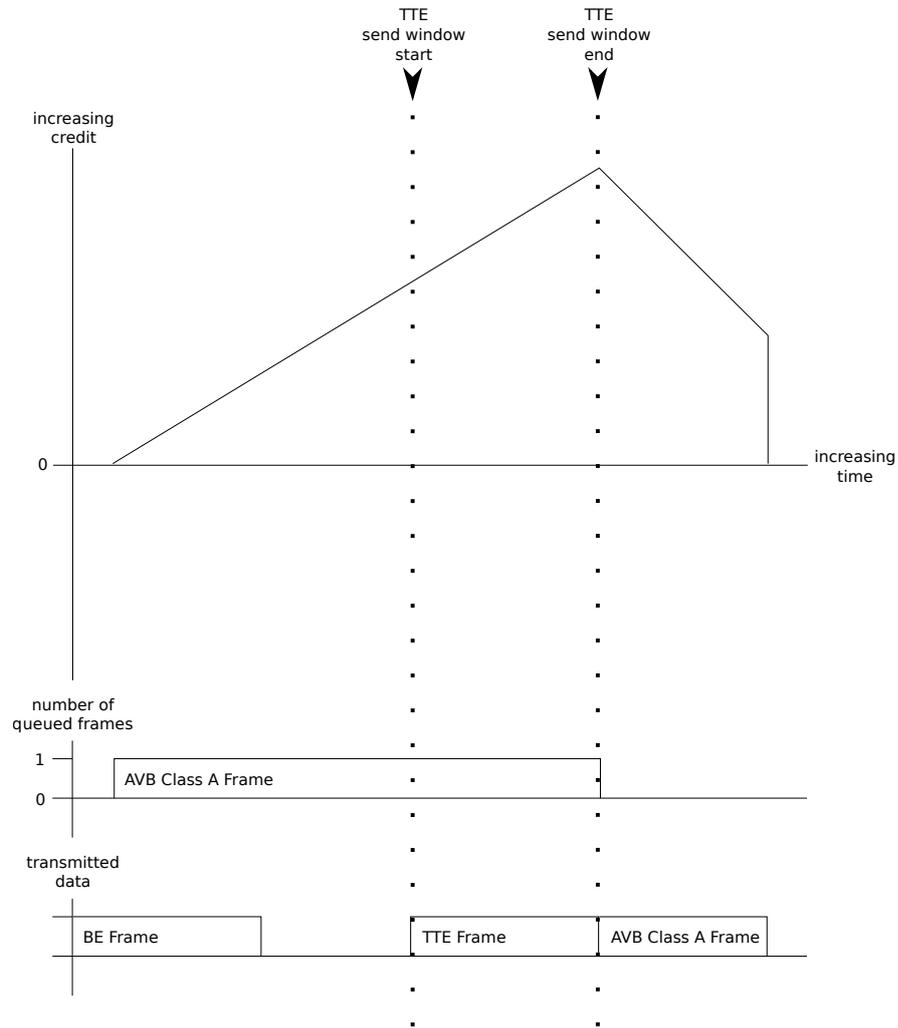


Abbildung 5.3: Neues Worst-Case Szenario.

In dieser Variante erhöht sich die maximale Dauer der Interferenz bei Klasse A Paketen um bis zu zwei maximale Ethernetpaketlängen(M). Bei einer Bandbreite(R) von 100Mbit/s mit einem Interframegap(T_{IFG}) von $0,96\mu s$ folgt daraus:

$$T_i = \frac{3 * M}{R} + 2 * T_{IFG} = \frac{3 * 1530Byte}{100Mbit/s} + 2 * 0,96\mu s = 352\mu s \quad (5.1)$$

Mit der Situation, dass das AVB Paket am Ende des CMI verschickt wird (T_s) folgt daraus die maximale Latenz pro Gerät:

$$T_{MaxLatenzDevice} = T_i + T_s = 352\mu s + 125\mu s = 477\mu s \approx 500\mu s \quad (5.2)$$

Die maximale Latenz hat sich folglich ungefähr verdoppelt. Darum können die 2ms maximale Latenz von AVB nur noch über 3 Hops garantiert werden, andernfalls verdoppelt sich diese über 7 Hops auf 4ms:

$$T_{MaxLatenz} = (1 + N_{Hops}) * T_{MaxLatenzDevice} = 4 * 500\mu s = \underline{\underline{2ms}} \quad (5.3)$$

$$T_{MaxLatenz} = (1 + N_{Hops}) * T_{MaxLatenzDevice} = 8 * 500\mu s = \underline{\underline{4ms}} \quad (5.4)$$

Es wird also erwartet, dass die Latenzen im schlechtesten Fall über den garantierten Latenzen im AVB Standard liegen. Das Reservieren und Einhalten von Bandbreiten ist weiterhin wie spezifiziert möglich, und auch die Einbindung in ein Netzwerk mit Standard AVB Geräten funktioniert weiterhin.

6 Realisierung der Erweiterung von TTE4INET

In diesem Kapitel wird die konkrete Erweiterung des TTE4INET Frameworks für OMNeT++ um AVB Komponenten erläutert. Im ersten Teil wird der Aufbau beschrieben. Danach folgt eine Vorstellung von ausgewählten Teilen der implementierten AVB Module. Diese Module sind in OMNeT++ in einer Kombination aus .h-, .cc- und .ned-Dateien umgesetzt. Die ersten Beiden werden genutzt um das Verhalten zu implementieren. Letzteres beschreibt die Architektur eines Moduls oder Netzwerks. Diese erfordern nicht zwingend eine C++ Klasse, da sie auch von anderen Modulen erben oder aus diesen zusammengesetzt sein können. Zur Vereinfachung der nachfolgenden Beschreibungen und Abbildungen werden Module und Klassen als gleichbedeutend interpretiert. Alle AVB Nachrichten sind vom Typ Klasse A. In den folgenden Abbildungen sind die in dieser Arbeit entstandenen Komponenten blau markiert. Die gelb markierten waren schon vorher im TTE4INET Framework enthalten oder stammen aus dem INET Framework.

6.1 Aufbau...

Um ein TTE-AVB-Netzwerk simulieren zu können, wird jeweils ein TTE-AVB-Switch und TTE-AVB-Host Modul entwickelt. Beide werden aus Modulen zusammengesetzt, die zum TTE4INET Framework hinzugefügt werden. Der Aufbau der Umsetzung wird hier unterteilt vorgestellt.

... der Host und Switch Komponenten

In Abbildung 6.1 ist das Klassendiagramm des Host Modules zu sehen. Auf der linken Seite befindet sich die Klasse „TTEAVBHost“, welche aus vielen Klassen besteht, die das spezifische Verhalten eines TTE-AVB-Hosts implementieren. Diese beginnen oben mit dem „TTEAVBPHY-Port“. Diese Klasse setzt einen Geräteport um. Sie besteht aus einer Input-, einer Output- und einer MAC Komponente. In- und Output sind mit der FullDuplex MAC Komponente aus dem

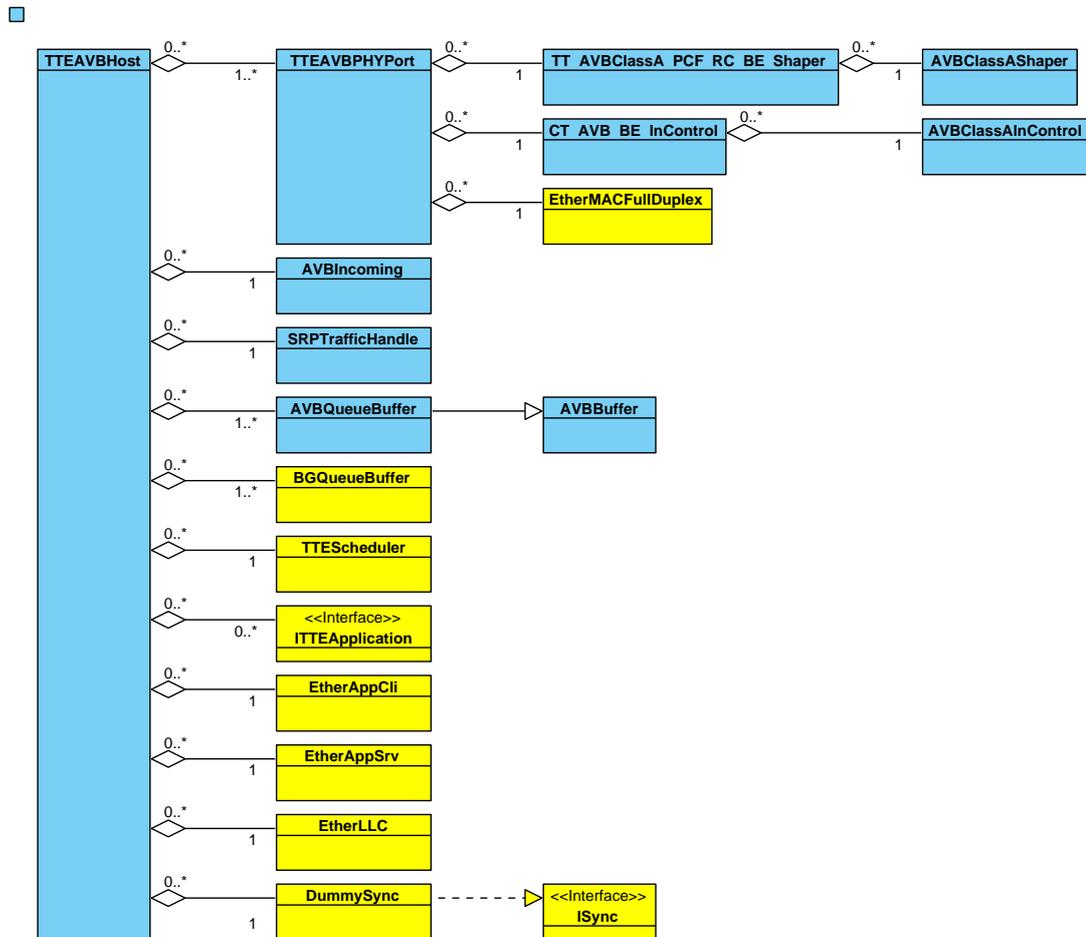


Abbildung 6.1: Klassendiagramm TTEAVBHost.

INET Framework verbunden um über diese Pakete zu senden oder zu empfangen. Der Output, hier der „TT_AVBClassA_PCF_RC_BE_Shaper“, implementiert zusammen mit dem AVB Puffer das Scheduling des „Credit Based Shaper“ Algorithmus. Die Besonderheiten dieser Klasse werden später genauer beschrieben. Zusammen ist dies die Umsetzung vom IEEE 802.1Qav Standard (vgl. [Institute of Electrical and Electronics Engineers \(2009b\)](#)).

Als nächstes folgt die „AVBIncoming“ Klasse. Diese setzt das Stream Reservation Protokoll um und sorgt für das Routing der AVB Pakete. Das ist der Teil aus dem IEEE 802.1Qat Standard (vgl. [Institute of Electrical and Electronics Engineers \(2010\)](#)). Die Identifizierung von SRP Nachrichten geschieht im „SRPTrafficHandle“.

Pro Port enthält der Host ein „AVBQueueBuffer“ Modul, das die vom „AVBIncoming“ ve-
teilten AVB Pakete puffert. Die „AVBBuffer“ Klasse von der diese erbt, enthält den Credit und
sendet in Abhängigkeit von diesem gepufferte Pakete an die Portkomponente.

Die Klasse „BGQueueBuffer“ aus dem TTE4INET Framework stellt den Puffer für Best-Effort
Pakete bereit.

Die „TTEScheduler“ Komponente steuert über Events das Versenden on Time-Triggered
Paketen.

Es gibt verschieden Klassen, die Stimuligeneratoren für die unterschiedlichen Nachrichten-
klassen umsetzen. Diese implementieren das „ITTEApplication“ Interface. „EtherAppCli“ und
„EtherAppSrv“ sind die Stimuligeneratoren für BE Verkehr aus dem INET Framework und
implementieren daher nicht das Interface aus TTE4INET. Sie setzen Client und Server um.
Das Modul „EtherLLC“ stammt ebenfalls aus diesem Framework und implementiert das Lo-
gical Link Control Protokoll auf Linklayer Ebene eines IEEE 802.3 Ethernet Netzwerks. Die
„DummySync“ ist ein Modul, dass die Ungenauigkeiten durch eine netzwerkweite Gerätesyn-
chronisation simuliert.

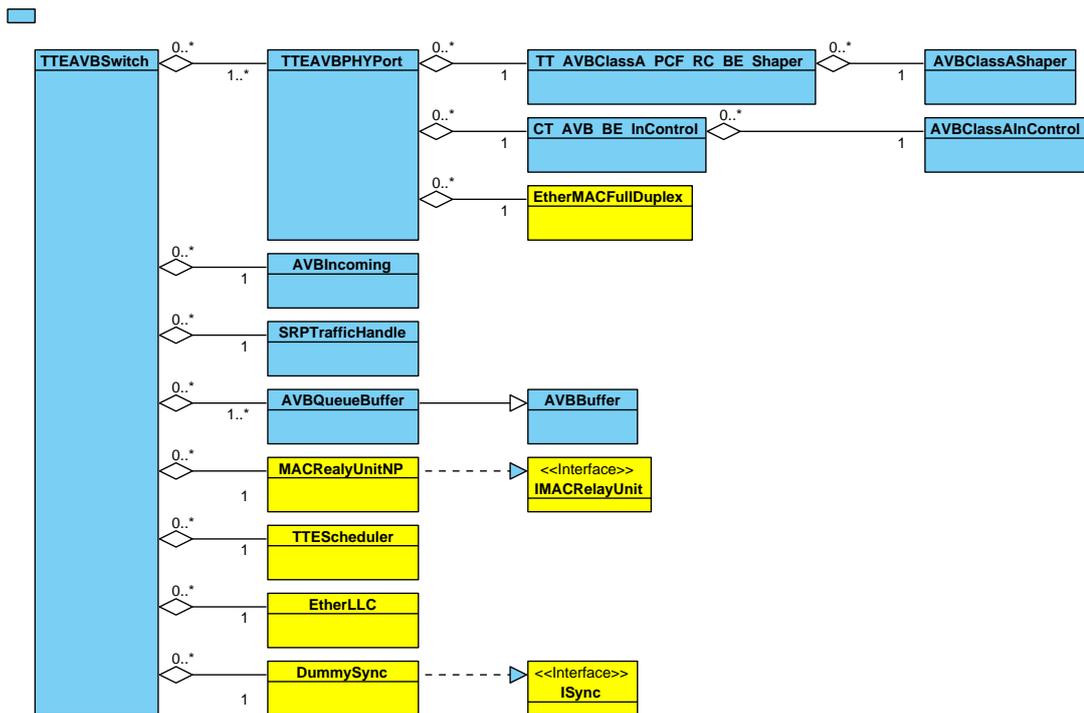


Abbildung 6.2: Klassendiagramm TTEAVBSwitch.

Die Umsetzung des Switches in „TTEAVBSwitch“ besteht zum großen Teil aus den gleichen Komponenten. Der erste Unterschied ist der Verzicht auf „BGQueueBuffer“. Statt diesen Best-Effort Puffern ist das Modul „MACRelayUnitNP“ aus dem INET Framework enthalten. Dieses implementiert das komplette Verhalten eines IEEE 802.3 Switches und wird hier für das Routing von BE Paketen genutzt. Der zweite Unterschied ist, dass ein Switch keine Stimuligeneratormodule benötigt. Der Aufbau des Switches ist in Abbildung 6.2 zu sehen.

... des „TT_AVBClassA_PCF_RC_BE_Shaper“ und „CT_AVB_BE_InControl“

Diese besonderen Klassen werden aus Templateklassen zusammengesetzt. Für jede Nachrichtenklasse gibt es eine Templateklasse, die das Verhalten des konkreten Shapers implementieren. Die Definition des AVB Klasse A Shapers sieht dann zum Beispiel wie folgt aus:

```
1 /**
2  * @brief A Shaper for AVB-Messages.
3  *
4  * The TTShaper only allows lower priorities to transmit when the
5  *   credit is <= 0.
6  */
7 template <class TC>
8 class AVBClassAShaper : public TC
9 {
10     ...
11 }
```

Ein zusammengesetzter Shaper, in diesem Beispiel der „TT_AVBClassA_PCF_RC_BE_Shaper“, wird dann schließlich wie folgend definiert:

```
1 /**
2  * @brief Class representing the TT_AVBClassA_PCF_RC_BE_Shaper
3  *   module
4  *
5  * This shaper handles traffic in the following order
6  * (highest to lowest priority):
7  * - time-triggered
8  * - AVB
9  * - PCF
10 * - rate-constrained
11 * - best-effort
12 */
```

```
12 * @see TTShaper , AVBClassAShaper , PCFShaper , RCShaper , BESHaper ,  
    BaseShaper  
13 */  
14 class TT_AVBClassA_PCF_RC_BE_Shaper : public TTShaper <  
    AVBClassAShaper <PCFShaper <RCShaper <BESHaper <BaseShaper > > >  
    >{};
```

Die Ineinanderschachtelung der Generics repräsentiert die Reihenfolge der Prioritäten. Wenn ein Paket durch das MAC Modul angefordert wird, geschieht nun Folgendes: Die Shaper Implementation mit der höchsten Priorität, in der alle anderen verschachtelt sind, wird aufgerufen ein Paket zu übergeben. Diese überprüft nun ob sie eines zum versenden bereit steht und ob sie es versenden darf. Sollte dieses nicht der Fall sein wird überprüft ob ein Paket der darunter liegenden Priorität versendet werden darf. Wenn dieses erlaubt ist, wird der gleiche Aufruf eine Ebene unter dem eigenen Objekt getätigt. Dieser Vorgang zieht sich weiter soweit durch bis ein Shaper ein Paket zum Versenden zur Verfügung stellt, keiner eines zur Verfügung stellen kann oder ein Shaper keine Übertragung einer niedrigeren Priorität erlaubt.

Dieses Verfahren ermöglicht ein beliebiges hinzufügen, entfernen oder vertauschen von Prioritäten, ohne das ein doppelter Code entsteht. Es lassen sich zudem bereits fertige Shaper Implementationen nutzen. Der eigene Code wird einfach, mit einer neuen Shaper Templateklasse, in eine weitere Shaper Klasse eingehängt. Das gleiche Konzept gilt für die InControl Klassen wie zum Beispiel die „CT_AVB_BE_InControl“ Klasse.

... der Module im erweiterten TTE4INET Framework

Alle Module sind in die Vererbungshierarchie des TTE4INET Frameworks eingehängt um bereits implementierte Funktionalität weitestgehend wiederzuverwenden. In Abbildung 6.3 ist eine Ausnahme zu sehen. Die Klasse „AVBIncoming“ erbt nicht, wie erwartet, von „Incoming“. Der Grund hierfür ist, dass es bisher nur eine 1:1 Beziehung zwischen Puffer und „Incoming“ gibt und dieses darum nur einen Ausgang für Pakete am Modul definiert. Da die neue AVB Komponente aber auch das Routing an die Puffer übernimmt braucht es ein Array von Ausgängen.

Jede Implementation von „IShaper“ und „InControl“ baut auf das bereits beschriebene Templateklassenkonzept auf. Die .ned Module dieser Klassen erben dennoch voneinander um auch die Architektur, die eine konkrete Shaper Implementation benötigt, mit zu übernehmen. Durch das Erben von „TT_PCF_RC_BE_Shaper“ kann die Architektur dieses Moduls mit übernommen werden. Nur die AVB Klasse A Shaper Architektur muss umgesetzt werden. Die „TT_AVBClassA_PCF_RC_BE_Shaper“ Klasse regelt den ausgehenden Verkehr und

die „CT_AVB_BE_InControl“ Klasse behandelt einkommende Pakete und verteilt sie an die zugehörigen Komponenten. Auch das „Buffer“ Modul implementiert auch das Interface „IOutputQueue“ und ist weiter unten aufgelöst.

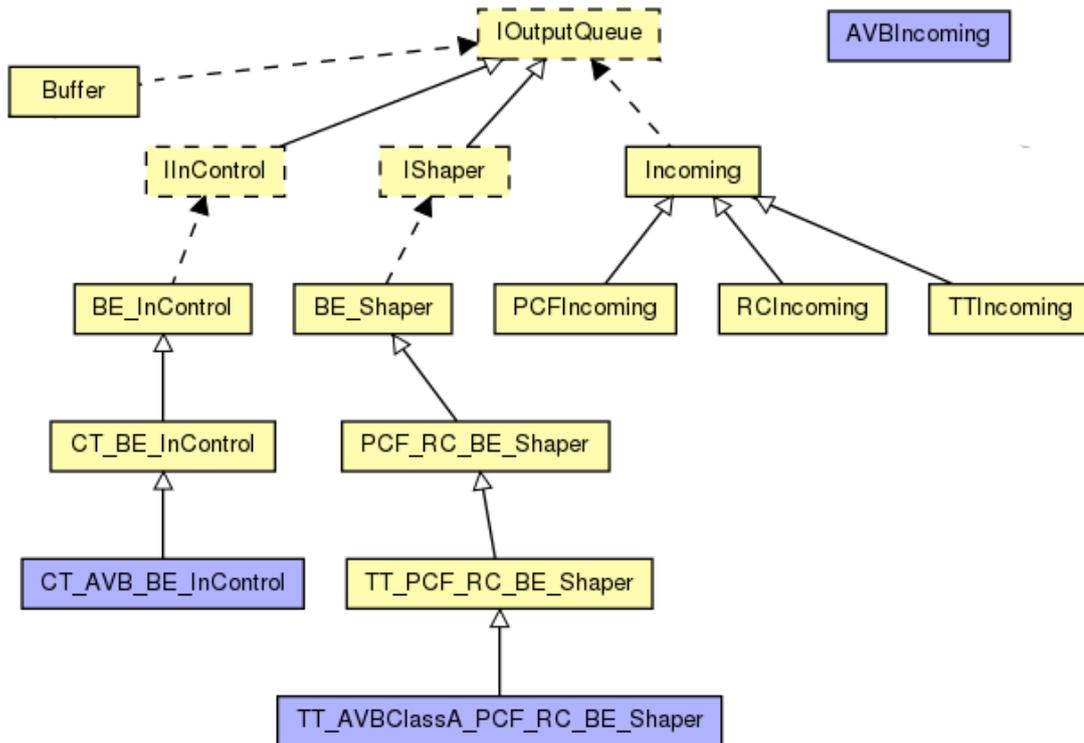


Abbildung 6.3: Vererbungshierarchie von IOutputQueue und AVBIncoming.

Von der „Buffer“ Klasse erben alle verschiedenen Ausprägungen der Puffer für die unterschiedlichen Nachrichtenklassen (siehe Abb. 6.4). Diese Implementierungen konzentrieren sich auf Besonderheiten der jeweiligen Spezifikationen. Das AVB Queueing unterscheidet sich in der Hinsicht, dass das Übertragen an den Port vom Zählerstand des Credits abhängig ist. Dies ist in „AVBBuffer“ implementiert und der Double und Queue Puffer erben dieses Verhalten. Die beiden anderen Pufferimplementationen („BGBuffer“ und „CTBuffer“) sind unabhängig von einem Credit und deswegen nicht für eine AVB Implementation geeignet.

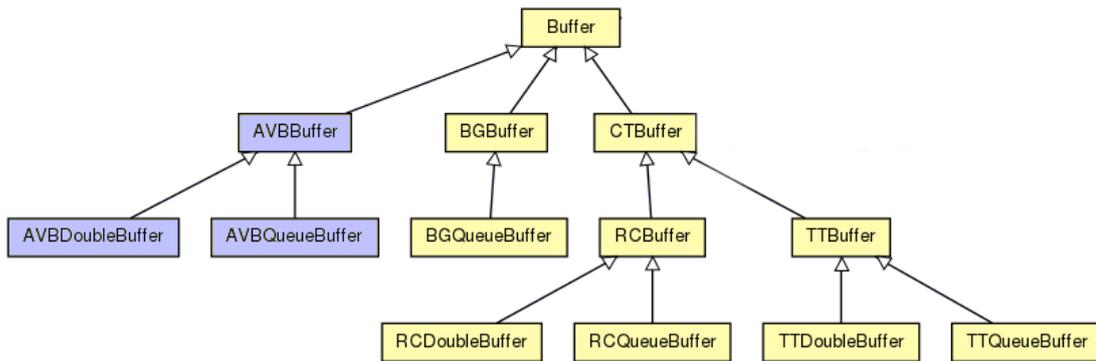


Abbildung 6.4: Vererbungshierarchie von Buffer.

In der Abbildung 6.5 sind die drei Klassen „TTEAVBPHYPort“, „SRPTrafficHandle“ und „AVBTrafficSourceApp“ zu sehen. Die ersten Beiden sind bereits beschrieben worden. Sie stellen den physischen Port und das Modul zur Identifikation der SRP Nachrichten dar. Die dritte Klasse in diesem Bild ist die Umsetzung eines AVB Talkers und Listeners. Sie realisiert folglich das Clientverhalten des SR-Protokolls und dient im Fall eines Talkers als Quelle der AVB Pakete und im Fall eines Listeners als Senke. Der Grund, dass der Listener trotzdem in der „TrafficSourceApp“ implementiert ist, ist die Tatsache, dass ein Listener SRP Pakete generiert.

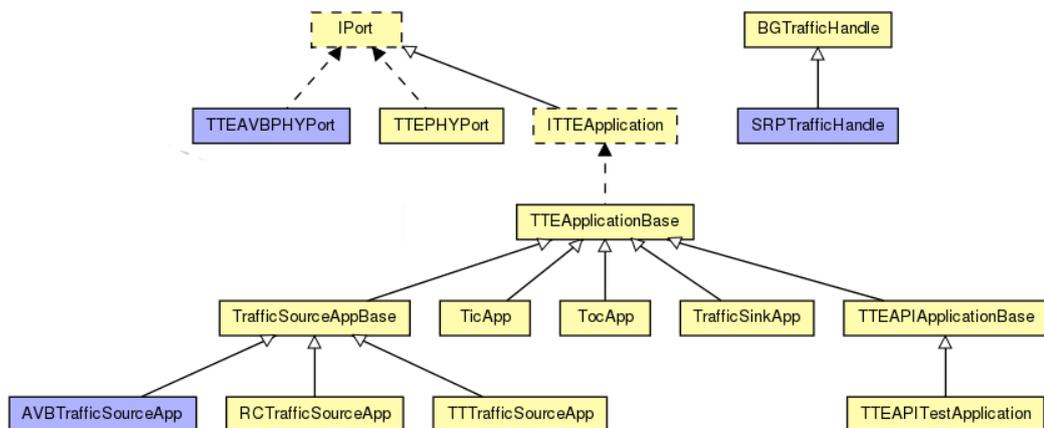


Abbildung 6.5: Vererbungshierarchie von IPort und BGTrafficHandle.

Die Pakete, die ein Stimulgenerator versendet oder empfängt sind „SRPFrame“ und „AVB-Frame“ Pakete (siehe Abb. 6.6). Das Erste repräsentiert ein Paket des Stream Reservation Protokolls und ist wie in IEEE 802.1Qat spezifiziert aufgebaut und zur Reservierung von

Bandbreiten in der Simulation genutzt. Die zweite Paketart wird von einem Talker generiert und als AVB Klasse A Paket in das Netzwerk übertragen. In zukünftigen Umsetzungen kann dieses auch für Klasse B Pakete verwendet werden, da beide den gleichen Aufbau haben.

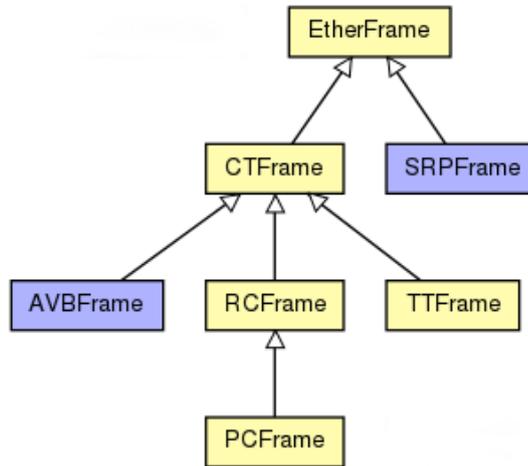


Abbildung 6.6: Vererbungshierarchie von EtherFrame.

... der Geräteinternen Nachrichtenrouten

Der Weg den Nachrichten innerhalb eines Gerätes nehmen, ist im TTE4INET Framework immer dem gleichen Konzept unterworfen. Dieser Weg ist in der Abbildung 6.7 vereinfacht dargestellt. Eine Nachricht kommt entweder über einen Stimuligenerator („APP“) oder einen Port im Gerät an. Sie wird dann an das zugehörige „Incoming“ Modul, das auch „Critical Traffic Control“ (CTC) genannt wird, weitergeleitet. Von dort aus geht die Nachricht, unter den durch das CTC gestellten Bedingungen, an einen Puffer und dieser sendet das Paket wieder an eine „APP“ oder einen Port. Hierüber verlässt die Nachricht dann wieder das Gerät.

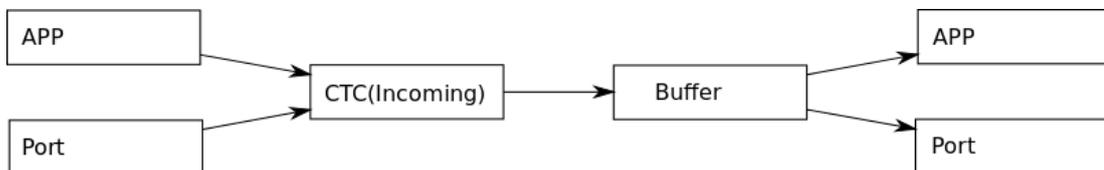


Abbildung 6.7: Konzept der geräteinternen Nachrichtenrouten im TTE4INET Framework.

6.2 Die implementierten Netzwerkteilnehmer in OMNeT++

Mit den Geräten lassen sich Netzwerke die eine Kommunikation von AVB und Time-Triggered Nachrichten unterstützen, sowohl modellieren als auch simulieren. Die Abbildung 6.8 zeigt beispielhaft die OMNeT++ Ansicht eines Hosts. Die Pfeile repräsentieren feste uni- oder bidirektionale Verbindungen über die die Nachrichten laufen.

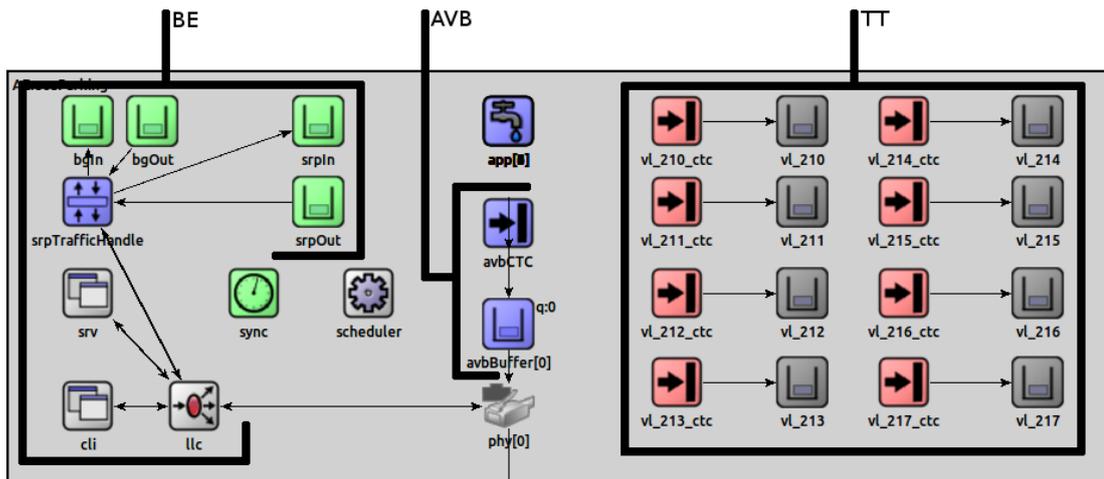


Abbildung 6.8: Beispiel eines TTE und AVB fähigen Hosts.

Es kann alternativ auch ohne eine solche Verbindung ein Paket an ein registriertes Objekt versendet werden. Dies ist zum Beispiel bei den Stimuligeneratoren notwendig, da erst nach der Konfiguration bekannt ist, ob und wieviele davon existieren. Die CTC und Puffer Module für die virtuellen TT Links sind auch erst nach der Konfiguration bekannt und vorhanden. In diesen Fällen kann daher zum Zeitpunkt der Implementation keine dieser Verbindungen festgelegt werden. In dem abgebildeten Host sind acht virtuelle Links für TT Nachrichten konfiguriert (vl_210 bis vl_217). Diese resultieren pro Link in ein „TTIncoming“ (CTC) und ein „TTBuffer“ Modul und sind beim Port registriert. Pakete mit der zugehörigen virtuellen Link ID werden an das jeweilige „Incoming“ übermittelt. Auch in dieser Ansicht sind die neuen AVB Komponenten blau gekennzeichnet. Als erstes ist das „app“, welches eine Menge von Stimuligeneratoren für verschiedene Nachrichtenklassen enthalten kann. Als Nächstes folgt eine Instanz des Modules „AVBIncoming“, das das Routing an die AVB Puffer „AVBQueue-Buffer“ regelt. In diesem Fall handelt es sich um einen Teilnehmer mit nur einem Port. Daher gibt es auch nur einen dieser AVB Puffer. Dann ist noch das „SRPTrafficHandle“ Modul zu

sehen. An dieses gehen eingehende **BE** Nachrichten und beinhaltende SRP Pakete werden identifiziert und gesondert gepuffert.

Ein Switch mit $n + 1$ Ports enthält jeweils auch $n + 1$ AVB Puffer, da im initialen Zustand eines Netzwerkes die Routen noch nicht bekannt sind. Dies ist anders als bei den statisch konfigurierten TT Routen. Der Switch enthält zwar mit acht Ports in Abbildung 6.9 auch acht Puffer für TT Nachrichten, diese Anzahlen sind aber nicht voneinander abhängig. Die TT Puffer sind an die Anzahl der virtuellen Links gekoppelt die über den Switch verlaufen.

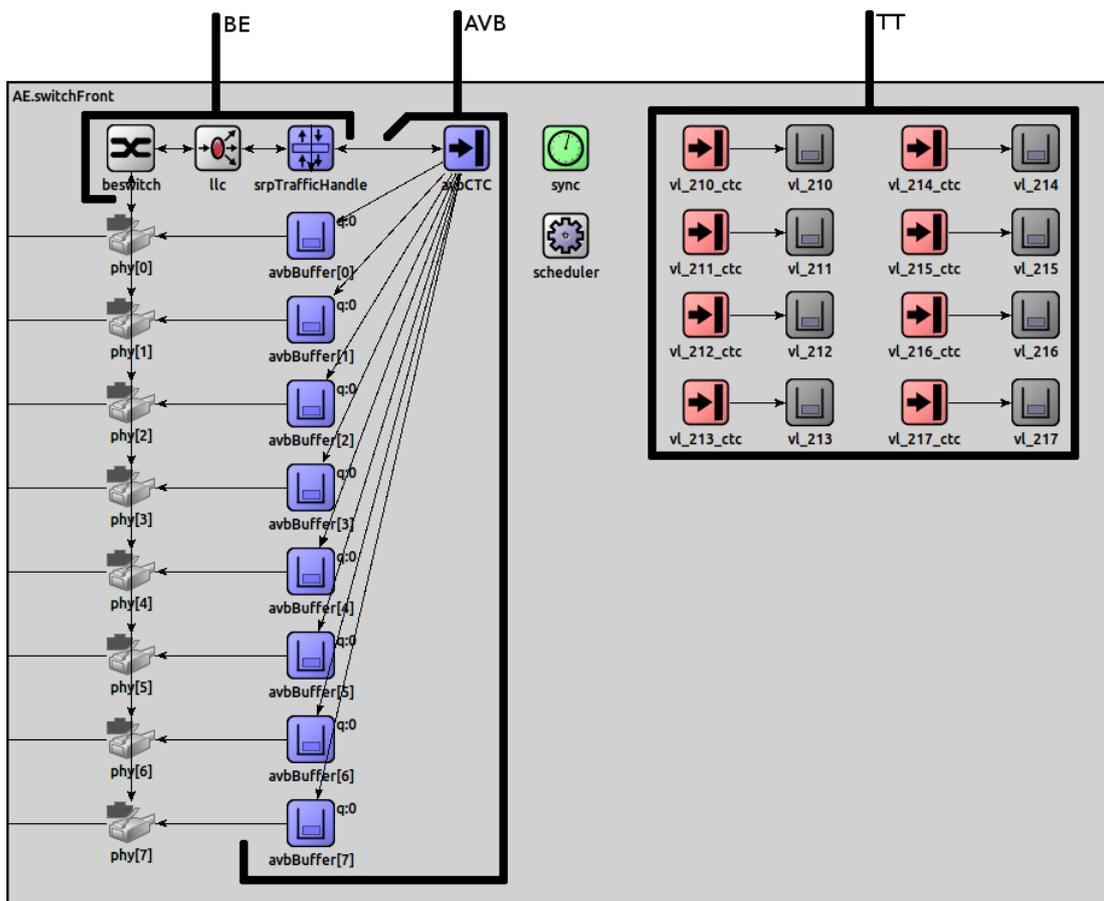


Abbildung 6.9: Beispiel eines TTE und AVB fähigen Switches.

Daher muss dieser Switch für das Versenden von AVB Paketen nicht konfiguriert werden. Nur die Teilnehmer, die als Talker oder Listener agieren, werden eingestellt. Die .ini Datei, die einen Talker konfiguriert enthält zum Beispiel folgenden Block für einen AVB Stream mit der ID 1:

```
1 **.cameraFront.numApps = 1
2
3 #App: Talker
4 **.cameraFront.app[0].typename = "AVBTrafficSourceApp"
5 **.cameraFront.app[0].talker = true
6 **.cameraFront.app[0].advertise_time = 50us
7 **.cameraFront.app[0].streamID = 1
8 **.cameraFront.app[0].intervalFrames = 1
9 **.cameraFront.app[0].payload = 390Byte
```

Diese Einstellungen resultieren in einen „Talker Advertise“ nach $50\mu s$ Simulationszeit, das einen Stream bekanntgibt, der pro Intervall (CMI) von $125\mu s$ ein Paket versendet. Die Nutzdaten dieser Pakete haben eine maximale Größe von 390 Byte.

Das zweite Beispiel zeigt den Ausschnitt einer Listener .ini Datei. Hierfür muss dem Stimuligenerator nur die ID des Streams mitgeteilt werden. Nach dem Empfangen eines „Talker Advertise“ Pakets, das die ID 1 enthält, wird ein „Listener Ready“ versendet um den Stream zu reservieren:

```
1 **.ecuParking.numApps = 1
2
3 #App: Listener
4 **.ecuParking.app[0].typename = "AVBTrafficSourceApp"
5 **.ecuParking.app[0].streamID = 1
```

Zusammen in einem Netzwerk resultiert diese Konfiguration folglich in einen Talker der einen Stream bereitstellt und bekanntgibt. Und in einen Listener, der versucht diesen Stream zu reservieren und bei Erfolg auch die zugehörigen Pakete empfängt.

7 Tests und Fallbeispiele

Dieses Kapitel ist in zwei Teile unterteilt und bezieht sich auf die Implementation der AVB Komponenten aus Kapitel 6. Im ersten Schritt werden Anforderungen aufgestellt, die das Gesamtsystem erfüllen müssen. Dazu werden die Testszenarien beschrieben, mit denen die Einhaltung der Anforderungen überprüft wird. Als zweites wird anhand von Fallbeispielen das Verhalten von AVB in Kombination mit TDMA Kommunikation untersucht.

7.1 Testszenarien

Zum Testen wird ein kleines und übersichtliches Netzwerk, das aus einem Switch mit drei angeschlossenen Endknoten besteht, herangezogen. Hier werden TT und AVB Nachrichten gleichzeitig versendet. Es gibt zwei unterschiedliche Konfigurationen. Der einzige Unterschied ist, dass die Eine mit, und die Andere ohne, nebenläufigen BE Verkehr simuliert.

7.1.1 Smoke Test

Die erste Anforderung an das erweiterte Framework ist, dass eine laufende Simulation nicht abstürzt. Dies wird durch den „Smoke“ Test sichergestellt. Hierfür gibt es ein Skript in der Testrubrik des Frameworks, das hintereinander verschiedene Netzwerke und Konfiguration startet. Konkret werden hier die oben beschriebenen Szenarien hinzugefügt. Die Ausgabe des Skriptes in der Konsole:

```
1 ~ $ ./smoketest
2 /small_network/ -f omnetpp.ini -c General -r 0 ... : PASS
3 /small_network/ -f omnetpp.ini -c With_Crosstraffic -r 0 ... : PASS
4 /rt_tictoc/ -f omnetpp.ini -c General -r 0 ... : PASS
5 /HighIntegrity_small/ -f omnetpp.ini -c General -r 0 ... : PASS
6 /StandardIntegrity_large/ -f omnetpp.ini -c General -r 0 ... : PASS
7 /StandardIntegrity_medium/ -f omnetpp.ini -c General -r 0 ... : PASS
8 /small_network_avb/ -f omnetpp.ini -c General -r 0 ... : PASS
9 /small_network_avb/ -f omnetpp.ini -c With_Crosstraffic -r 0 ... : PASS
10 -----
11 Ran 8 tests in 23.796s
```

12 OK

Alle Tests werden erfolgreich beendet. Damit kann nachgewiesen werden, dass die Netzwerkbeispiele erfolgreich starten und ohne Absturz laufen.

7.1.2 Fingerprint Test

Beim „Fingerprint“ Test wird ein Hashwert über die aufgezeichneten Ergebnisse gebildet. Ein Netzwerk mit der gleichen Konfiguration und den gleichen Eingabeparametern, muss immer zu den gleichen Ergebnissen führen. Daraus folgt, dass auch der Hashwert („Fingerprint“) der Gleiche ist. Es kann also überprüft werden, ob sich die Implementation deterministisch verhält. Beim ersten Ausführen des Tests wurde der Hashwert in eine Tabelle eingetragen mit der jeder weitere Test die Werte vergleicht. Hier ist die Konsolenausgabe eines „Fingerprint“ Tests:

```
1 ~ $ ./fingerprints
2 /small_network/ -f omnetpp.ini -c General -r 0 ... : PASS
3 /small_network/ -f omnetpp.ini -c With_Crosstraffic -r 0 ... : PASS
4 /rt_tictoc/ -f omnetpp.ini -c General -r 0 ... : PASS
5 /HighIntegrity_small/ -f omnetpp.ini -c General -r 0 ... : PASS
6 /StandardIntegrity_large/ -f omnetpp.ini -c General -r 0 ... : PASS
7 /StandardIntegrity_medium/ -f omnetpp.ini -c General -r 0 ... : PASS
8 /small_network_avb/ -f omnetpp.ini -c General -r 0 ... : PASS
9 /small_network_avb/ -f omnetpp.ini -c With_Crosstraffic -r 0 ... : PASS
10 -----
11 Ran 8 tests in 44.380s
12 OK
```

Mit dem Bestehen dieser Tests ist sichergestellt, dass sich alle Komponenten deterministisch verhalten. Das heißt, dass sie beim erneuten Ausführen unter den gleichen Bedingungen zum gleichen Fingerabdruck führen.

7.1.3 Funktionstest

Das Gesamtsystem muss folgende Anforderungen erfüllen:

- TT Pakete verpassen niemals ihr Empfangsfenster.
- Die maximale Latenz der Klasse A Pakete entspricht den Annahmen aus Abschnitt 5.3.
- BE Pakete werden durch AVB Pakete verzögert.
- Der Credit des CBS steigt oder sinkt nicht ins Unendliche.

Zur Sicherstellung dieser Anforderungen wird das Beispielnetzwerk für 30 Simulationssekunden in der Konfiguration mit nebenläufigem BE Verkehr simuliert. Für die Abbildung 7.1 wurde das Netzwerk zusätzlich ohne AVB über den gleichen Zeitraum simuliert. Die Geräte ohne AVB sind reine TTE Teilnehmer, die schon vor dieser Arbeit in der TTE4INET Bibliothek vorhanden waren. Auf dem Bild ist der Latenzvektor von TT Nachrichten einmal mit und einmal ohne AVB zu sehen. Die Latenzen sind in beiden Fällen identisch. Das zeigt, dass die Latenzen von TT Nachrichten weiterhin unabhängig vom asynchronen AVB und BE Verkehr sind. Es werden also keine TT Empfangsfenster verpasst.

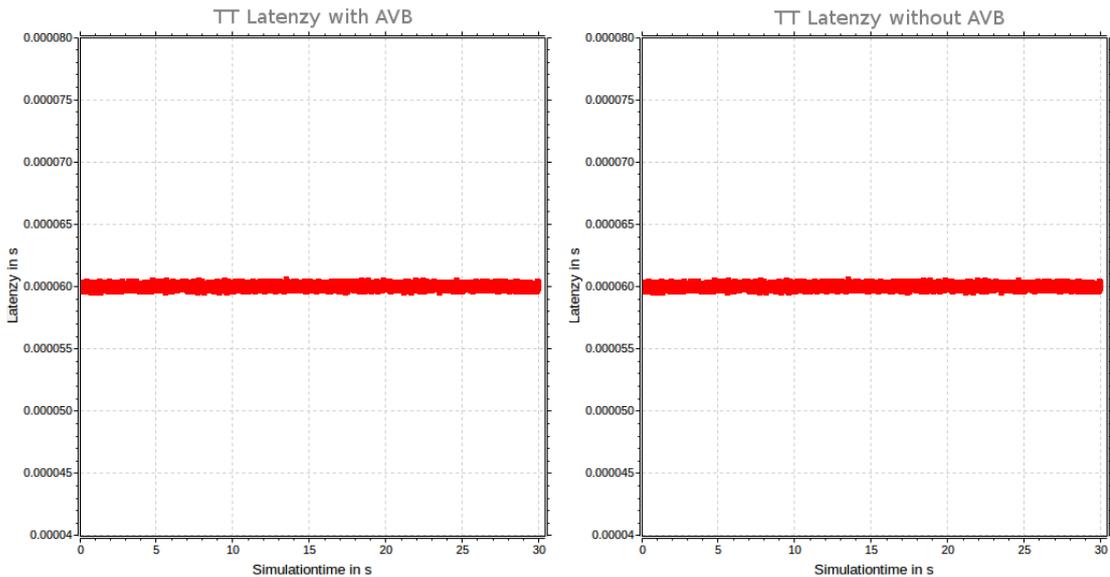


Abbildung 7.1: Latenz von TT Paketen mit und ohne AVB

Aus dem neuen Worst-Case Szenario für AVB Klasse A aus Abschnitt 5.3 folgt die maximale Latenz über 1 Hop:

$$T_{MaxLatenz} = (1 + N_{Hops}) * T_{MaxLatenzDevice} = 2 * 500\mu s = \underline{\underline{1ms}} \quad (7.1)$$

Vergleicht man diese mit den, in der Simulation gemessenen, Werten aus der Tabelle 7.1, wird deutlich, dass diese zu keinem Zeitpunkt überschritten wurde. Die sehr schlechte Latenz von BE Paketen resultiert daraus, dass die Endknoten mehr BE Pakete Versenden als Bandbreite zur Verfügung steht. Es gibt folglich einen Stau von BE Nachrichten. Das dieses nicht bei AVB Nachrichten geschieht zeigt, dass die reservierte Bandbreite Bestand hat. Die BE Pakete werden verzögert, um AVB den reservierten Platz zu lassen.

Tabelle 7.1: Latenz von AVB Klasse A und BE Paketen

	Minimum	Durchschnitt	Maximum
AVB Klasse A Latenz	69,61 μ s	238,38 μ s	433,52 μ s
BE Latenz	244,31 μ s	2,38s	5,95s

Die Tabelle 7.2 zeigt, dass der Credit des CBS je nach Auslastung eines Links unterschiedliche Extrema aufweist. Es treten aber keine Situationen auf, die den Algorithmus aushebeln könnten. Er steigt oder sinkt nie unaufhörlich.

Tabelle 7.2: Werte des CBS Credits

	Minimum	Durchschnitt	Maximum
Credit im Talkerausgang zum Switch	-2330	347,7	3885
Credit im Switchausgang zum Listener	-1606	1561,8	7507

7.2 Fallbeispiele

Nachdem nun die Funktion der TTE4INET Erweiterung erfolgreich getestet wurde, folgt die simulationsbasierte Untersuchung des erweiterten AVB Protokolls um TDMA basierte Kommunikation. Für diese Analyse wird beispielhaft eine ausgewählte Netzwerktopologie für verschiedene Konfigurationen genutzt um die Ergebnisse vergleichbar zu machen. Der Aufbau des Netzwerks ist in der Abbildung 7.2 zu sehen.

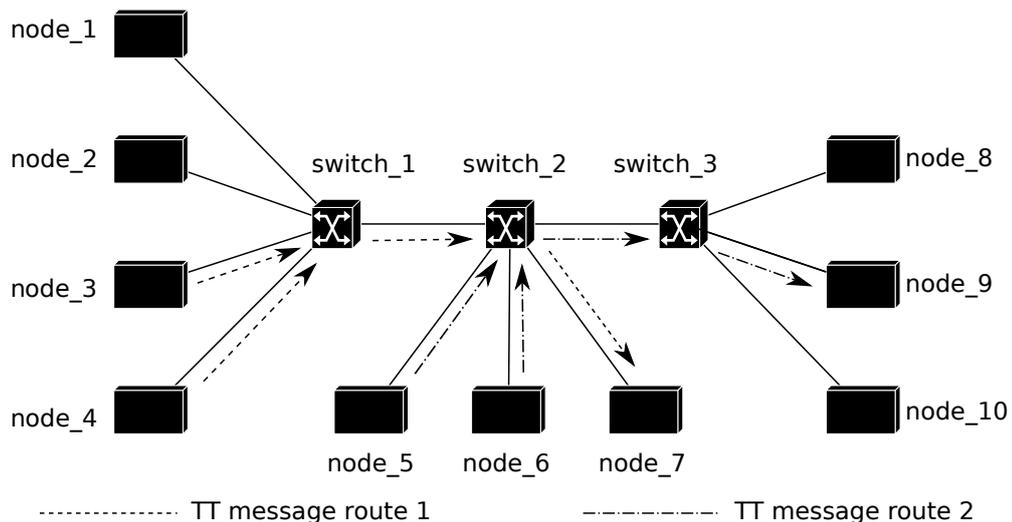


Abbildung 7.2: Netzwerk für die Fallbeispiele.

Die Basiseinstellungen für die Simulationen sind die Folgenden:

- Allgemein:
 - Alle physikalischen Links haben eine Bandbreite von 100Mbit/s.
 - Pro Fallbeispiel wird die Simulation für ca. 15s ausgeführt.
- Audio/Video Bridging:
 - „node_1“ und „node_2“ sind die AVB Talker.
 - Jeder AVB Stream versendet Pakete mit einer maximalen Nutzlast von 350Byte in dem „Class Measurement“ Intervall von $125\mu\text{s}$. Das entspricht einer Bandbreite von 50Mbit/s.
 - „node_8“ ist der AVB Listener für beide Streams.
- Time-Triggered Ethernet:
 - Der Zyklus für die TT Pakete ist 4ms lang.
 - Pro Zyklus und TT Nachrichten Route werden vier Pakete mit einem maximalen Payload von 1500Byte versendet. Das führt zu einer verbrauchten Bandbreite von 25Mbit/s.
- Best-Effort:
 - „node_10“ broadcastet ein maximales Best-Effort Paket in einem Intervall von 2 - 3ms.
 - Alle anderen Endknoten antworten mit einem weiteren maximalen Best-Effort Paket auf diesen Broadcast.

7.2.1 Nicht genügend Platz für AVB Pakete zwischen TT Paketen

In dieser Konfiguration wird versucht die Latenz von AVB Paketen in die Höhe zu treiben, indem zwischen zwei aufeinanderfolgenden TT Paketen gerade nicht genügend Zeit gelassen wird um ein AVB Paket zu übertragen. Genauer beträgt die Übertragungszeit der AVB Pakete $30\mu\text{s}$. Der Platz zwischen den vier TT Paketen beträgt $25\mu\text{s}$ Übertragungszeit. Abbildung 7.3 veranschaulicht die Situation auf dem Port von „switch_1“ zu „switch_2“ und von „switch_2“ zu „switch_3“. Auf diesen beiden kritischen Pfaden beträgt die Portauslastung nun 93%. 100% können nicht erreicht werden, da vor TT Paketen der Port freigehalten wird.

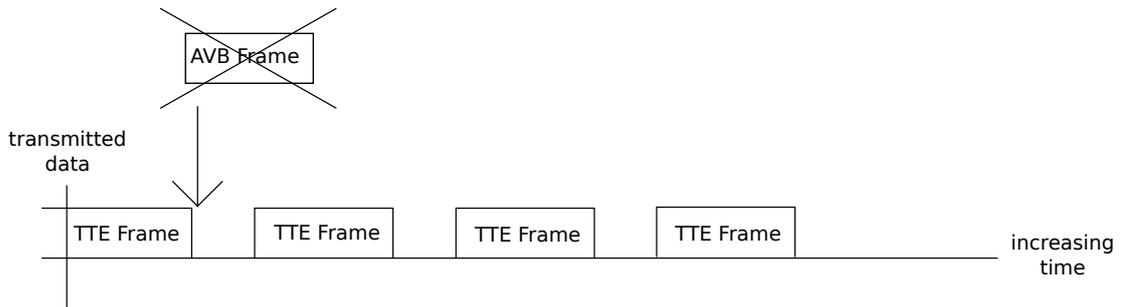


Abbildung 7.3: Nicht genügend Zeit zur Übertragung von AVB Nachrichten zwischen TT Nachrichten.

In dem Grafen in der Abbildung 7.4 ist ein Ausschnitt der Queuelänge von BE und AVB Paketen am Port von „switch_2“ in Richtung „switch_3“ zu sehen. Es wird gezeigt, dass die BE Queue in diesem Zeitraum immer Pakete enthält. Im Gegensatz dazu, schafft die AVB Queue sich innerhalb des gekennzeichneten Zykluses zu leeren. Am Anfang eines Zykluses werden die 4 TT Pakete versendet. Darum stauen sich die AVB Pakete in diesem Zeitraum auf. Danach wird der Rückstand durch Bursts wieder aufgeholt. Dies bedeutet, dass auch in einer so ungünstigen Situation für AVB, alle Pakete versendet werden können und die reservierte Bandbreite zur Not auch auf längere Sicht durch den CBS Algorithmus durchgesetzt wird.

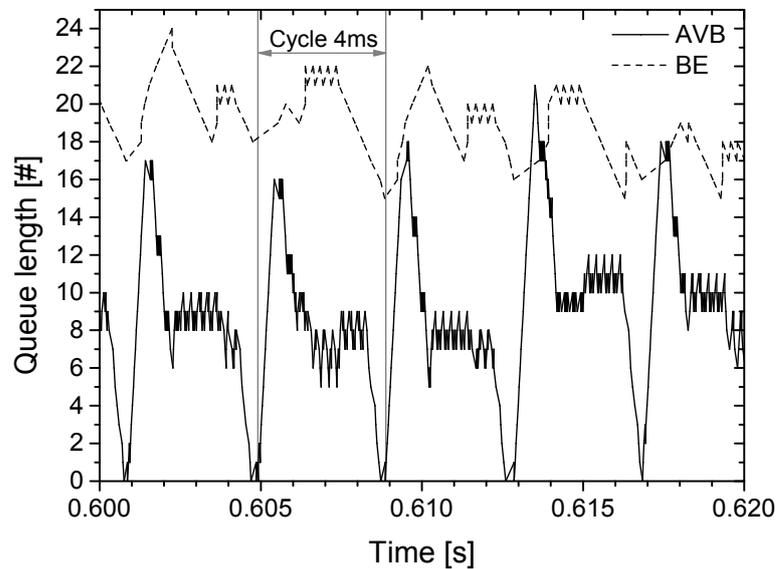


Abbildung 7.4: Anzahl der Nachrichten in den Puffern von „switch_2“ zu „switch_3“.

Das Histogramm über die Anzahl der Pakete pro Latenz für diese Simulation ist in der Abbildung 7.5 zu sehen. Die relative Anzahl der Pakete ist in Prozent angegeben. Dadurch kann die Verteilung der Verzögerungen der drei unterschiedlichen Nachrichtenklassen untersucht werden. Die BE Pakete haben den größten Jitter von $244,56\mu s$ bis $9268,56\mu s$ Latenz. Wie erwartet ist der Worst-Case dieser Übertragungsklasse der Höchste auch wenn knapp 20% der Pakete sogar einen besseren Wert erreichen als ihr TT Pendant. Die Time-Triggered Pakete haben einen Jitter im ns Bereich bei einer Latenz von $465\mu s$. Die Übertragung dieser, ist auch mit AVB bei einer niedrigen Latenz hochpräzise und vorherbestimmbar. Die niedrigste Latenz von AVB Klasse A Paketen beträgt $214,48\mu s$, die höchste aber $1748,83\mu s$.

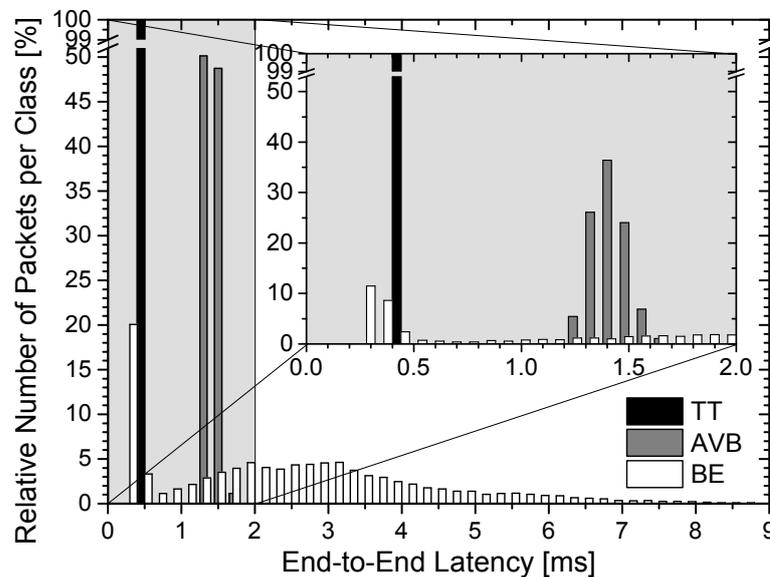


Abbildung 7.5: Histogramm der Ende zu Ende Latenz bei kompakter TT Konfiguration.

Laut Abschnitt 2.2.5 und 5.3 ergibt sich für dieses Netzwerk die maximale Latenz von Klasse A Paketen wie folgt:

$$\begin{aligned}
 T_{MaxLatenz} &= T_{MaxLatenzNode_{1/2}} + T_{MaxLatenzSwitch_1} \\
 &\quad + T_{MaxLatenzSwitch_2} + T_{MaxLatenzSwitch_3} \\
 &= 250\mu s + 500\mu s + 500\mu s + 250\mu s \\
 &= 1ms
 \end{aligned} \tag{7.2}$$

Dieses gilt, da von den Talkern „node_1“ und „node_2“, sowie von „switch_3“ zum Listener „node_8“ keine TT Pakete verschickt werden. Auf diesen Routen gilt die vom AVB Standard definierte maximale Latenz. Im Ergebnis wird diese hier übertroffen. Der Grund liegt darin, dass der Zeitraum zwischen den TT Paketen zu kurz ist. Mit der richtigen Konfiguration der TT Sendezeitpunkte und Paketgrößen, kann die Latenz der AVB Nachrichten also beliebig steigen.

In diesem Fall muss die maximale Latenz individuell betrachtet werden um eine Obergrenze definieren zu können. Im Folgenden wird dieses für das konkrete Szenario getan. Die Verzögerungszeit T_i erhöht sich auf die Sendedauer des Eigenen und fünf weiterer Pakete. Der Abstand T_{GAP} zwischen den TT Nachrichten beträgt wie oben beschrieben $25\mu s$:

$$T_i = \frac{6 * M}{R} + 2 * T_{IFG} + 3 * T_{GAP} = \frac{6 * 1530Byte}{100Mbit/s} + 2 * 0,96\mu s + 3 * 25\mu s = 811\mu s \quad (7.3)$$

Daraus folgt:

$$T_{MaxLatenzSwitch_{1/2}} = T_i + T_s = 811\mu s + 125\mu s = 936\mu s \approx 1000\mu s \quad (7.4)$$

$$\begin{aligned} T_{MaxLatenz} &= T_{MaxLatenzNode_{1/2}} + T_{MaxLatenzSwitch_1} \\ &\quad + T_{MaxLatenzSwitch_2} + T_{MaxLatenzSwitch_3} \\ &= 250\mu s + 1000\mu s + 1000\mu s + 250\mu s \\ &= 2,5ms \end{aligned} \quad (7.5)$$

Diese Obergrenze ist nun die AVB Latenzgarantie für das konkrete Netzwerk. Auch die beschriebenen Ergebnisse der Simulation zeigen, dass die neue Grenze in jeder Situation eingehalten wird. Das Szenario zeigt die Abhängigkeit der AVB Verzögerungen von der Konfiguration der TDMA Kommunikation. Für eine generelle Garantie muss die Zeit zwischen zwei TT Paketen lang genug sein um innerhalb dieser mindestens eine maximal große AVB Nachricht zu versenden.

7.2.2 Ausreichend Platz für AVB Pakete zwischen TT Paketen

In Simulation wird im Gegensatz zu Abschnitt 7.2.1 der Abstand zwischen den TT Paketen so vergrößert, dass mindestens ein AVB Paket versendet werden kann (Abb. 7.6). Der Abstand

zwischen den Time-Triggered Paketen ist auf $125\mu\text{s}$ gewachsen. Dadurch wird ein regelmäßiges verschicken von AVB Paketen gemäß des Ansatzes in Kapitel 5 möglich.

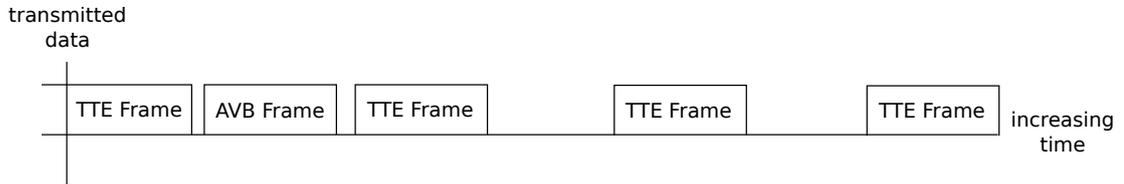


Abbildung 7.6: Genügend Zeit zur Übertragung von AVB Nachrichten zwischen TT Nachrichten.

Die AVB Pufferlänge an dem Port von „switch_2“ zu „switch_3“ ist in diesem Szenario maximal 7 Pakete lang. Im Gegensatz zum vorherigen Fallbeispiel ist das eine massive Verbesserung und zeigt an, dass die Nachrichten kürzeren Wartezeiten ausgesetzt sind. Dies ist auch auf dem Histogramm 7.7 zu sehen. Die TT Klasse hat die gleiche Latenz von $465\mu\text{s}$. Durch die angepassten Sendezeitpunkte der Pakete dieser Klasse, werden Latenz und Jitter der beiden Nachrichtenklassen AVB A und BE verbessert.

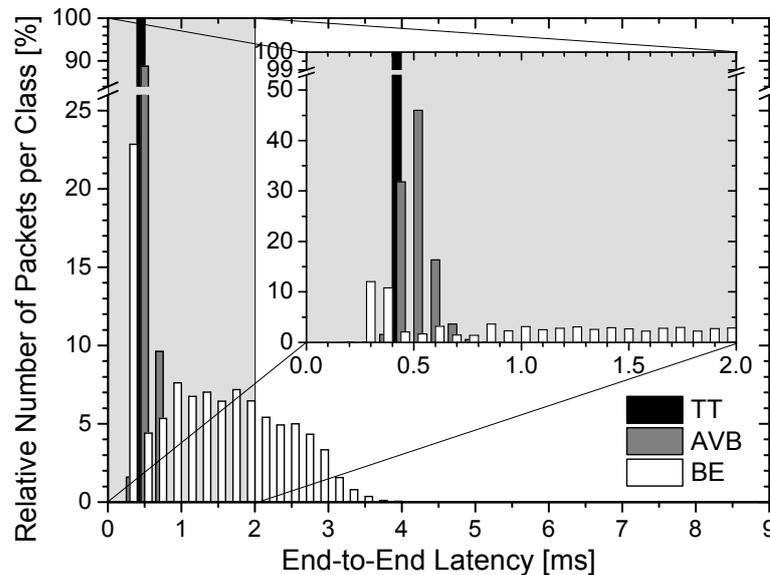


Abbildung 7.7: Histogramm der Ende zu Ende Latenz bei gestreckter TT Konfiguration.

Konkret bedeutet das eine maximale Latenz von $4388,67\mu\text{s}$ im Falle der BE Pakete und $874,52\mu\text{s}$ bei AVB Klasse A Paketen. Obwohl die genutzte Bandbreite die Gleiche wie im Fall

aus Abschnitt 7.2.1 ist, ist die ursprünglich erwartete Obergrenze der Latenz von 1ms bei AVB Klasse A unterschritten. Daraus folgt, dass in der Kombination von AVB mit TDMA basierter Kommunikation, nicht nur die Größe der Pakete anderer Nachrichtenklassen eine Auswirkung auf die AVB Latenz hat. Auch das konfigurierte Schedule der TT Nachrichten spielt eine große Rolle. Die Garantien von AVB Streams stehen und fallen mit dem statischen TT Schedule. Durch das Einhalten einer Regel, die einen Mindestabstand von $125\mu s$ zwischen TT Paketen vorschreibt, lässt sich die im Abschnitt 5.3 festgestellte neue Latenzgarantie als gültig erklären.

7.2.3 Paketgrößen von TT und BE variieren

Als Drittes wird die Paketgröße von TT und BE Paketen halbiert. Die anderen Basiskonfigurationen bleiben erhalten. Der Schedule der Time-Triggered Nachrichten ist wie im vorherigen Fall so angelegt, dass AVB Klasse A Pakete zwischen den TT Frames versendet werden können. Die Tabelle 7.3 zeigt, die aus den Simulationsergebnissen gewonnenen Ende zu Ende Latenzen. Die Latenz von TT Nachrichten hat sich durch die Veränderung der Länge dieses mal verkürzt. Der Jitter ist im Vergleich zu den beiden anderen Fällen unverändert niedrig. Die daraus resultierende kürzere Wartezeit beim Shaping von AVB Paketen, führt zu einer noch besseren Durchschnittslatenz als bei TT Frames. Auch die höchste Latenz von AVB und BE Paketen verbessert sich, wobei BE Pakete im Verhältnis am meisten profitieren. Dies ist auf die Entlastung der physikalischen Links, durch den geringeren Datendurchsatz, zurückzuführen.

Tabelle 7.3: Latenz der verschiedenen Nachrichtentypen bei halbiertem Paketlänge von TT und BE Nachrichten

	Minimum	Durchschnitt	Maximum
TT	$344,56\mu s$	$345,03\mu s$	$345,52\mu s$
AVB Klasse A	$145,12\mu s$	$324,65\mu s$	$555,31\mu s$
BE	$124,56\mu s$	$511,43\mu s$	$1530,54\mu s$

Die Latenzen von AVB Nachrichten sind wie erwartet von der Größe der TT und BE Pakete abhängig. Dieses wird durch die Abbildung 7.8 unterstrichen. Die Abbildung zeigt die Latenzen bei variierten maximalen Paketgrößen der beiden anderen Nachrichtentypen. Zur Erstellung wurde das Netzwerk 15 mal mit verschiedenen Paketgrößen (MTU) von 100 bis 1500 Byte simuliert. Das Szenario mit den kleinsten Paketen kommt der Praxis im Automobil am nächsten, da der Kontrollverkehr im Automobil, der über TT Nachrichten übertragen werden müsste, meist sehr kleine Daten aufweist. Die Größe der AVB Pakete bleibt dabei immer gleich.

Das die Punkte der maximalen Latenzen nicht genau auf der linearen Geraden liegen, liegt am Einfluss des TDMA Scheduling. Durch größere Pakete kann eine bessere Ausgangssitua-

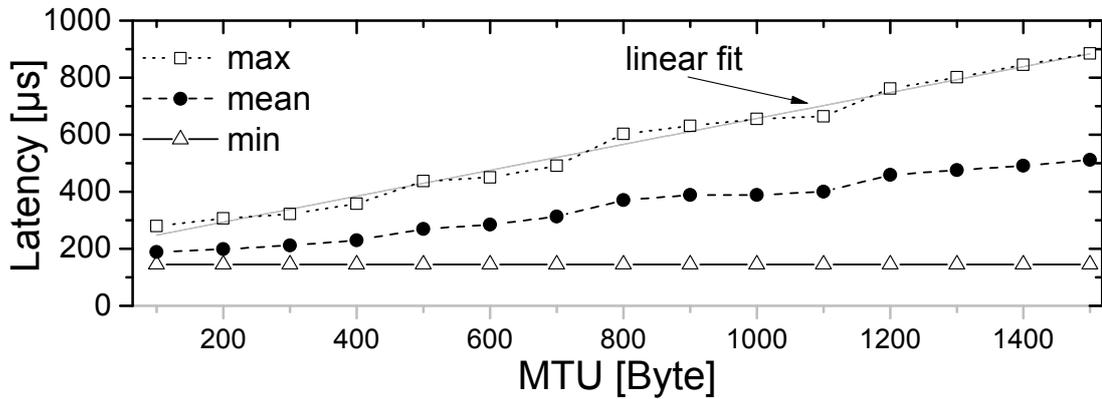


Abbildung 7.8: Latenzen der AVB Klasse A Pakete bei unterschiedlichen maximalen Paketgrößen der anderen Nachrichtenklassen

tion entstehen, wenn dadurch genau eine gerade Anzahl an AVB Paketen zwischen die TT Frames passt. Es passt in jeder dieser Simulationen mindestens eins in den Zwischenraum.

Zusammengefasst ist der größte Einfluss auf die Latenz von AVB Paketen die maximale Größe anderer Pakete im Netzwerk. Der Einfluss durch das Scheduling kann vernachlässigt werden solange immer mindestens ein maximales AVB Paket hinter einer TT Nachricht versendet werden kann. Ansonsten kann der Einfluss des Scheduling auch so weit reichen, dass im schlimmsten Fall kein einziges AVB Paket sein Ziel erreicht.

8 Fazit und Ausblick

Das abschließende Kapitel dieser Arbeit enthält eine Zusammenfassung, einen Überblick der Ergebnisse und einen Ausblick auf künftige Arbeiten und Projekte.

8.1 Zusammenfassung und Ergebnisse

Das erste Ziel dieser Arbeit war die Entwicklung von AVB Komponenten für das TTE4INET Framework, die sowohl TT als auch AVB Klasse Pakete behandeln können. Dazu wurde ein Konzept entwickelt, das eine mögliche Kombination von AVB und TDMA Kommunikation realisierbar macht. Darum wurden umfassende Kenntnisse über die beiden IEEE Standards 802.1Qat und 802.1Qav, die die Spezifikation vom Stream Reservation Protokoll und des Credit Based Shaper Algorithmus enthalten, gesammelt. Dieses Ziel konnte erreicht werden und es steht nun ein erweitertes TTE4INET Framework zur Verfügung, mit dem die Simulation von AVB, TTE oder einer Kombination aus beidem möglich ist.

Für das Erfüllen des zweiten Teilziels wurden mit diesem Framework verschiedene Fallbeispiele simuliert und der Einfluss der TDMA basierten Kommunikation auf AVB untersucht. Dafür wurde ein Netzwerk mit unterschiedlichen Auslastungen und TT Schedulings simuliert und interessante Ergebnisse dargestellt und ausgewertet.

Folgendes konnte festgestellt werden: Durch das Befolgen einiger Regeln bei der offline Konfiguration des TT Anteils in einem Netzwerk, lässt sich eine maximale Latenz von Klasse A Nachrichten garantieren. Diese beträgt das Doppelte (4ms über 7 Hops) der Garantie vom Standard AVB (2ms über 7 Hops). Nochmal zusammengefasst sind das diese Regeln:

- Das Stream Reservation Protokoll beachtet die, durch TT Nachrichten verursachte, Bandbreite.
- Frames für die Synchronisation der Zeit zwischen den Geräten werden vom Credit Based Shaper Algorithmus als TT Pakete interpretiert.

- Der Platz zwischen den TT Sendefenstern ist groß genug um mindestens ein AVB Paket mit maximaler Länge zu übertragen.

Bei Befolgung kann Ethernet AVB mit einer TDMA Erweiterung durch den TT Anteil höhere zeitliche Anforderungen erfüllen, als es das Standard AVB leisten kann. Trotz einer höheren Worst-Case Latenz von $500\mu\text{s}$ pro Sendevorgang ist weiterhin eine performante Klasse A Übertragung gewährleistet. Die Latenz Obergrenze dieser Nachrichtenklasse lässt sich durch kleinere TT und BE Pakete in einem Netzwerk weiter nach unten drücken.

Bei einer Anwendung im Auto mit sehr kleinen wichtigen Kontrolldaten, die die TT Klasse zur Übertragung nutzen würden, kann so für Klasse A Pakete eine sehr niedrige Latenz mit wenig Jitter Garantiert werden. Diese Latenz liegt dann nur leicht über der von Standard AVB Klasse A Nachrichten. Trotzdem gewinnt man die hohe Echtzeitfähigkeit der Time-Triggered Komponenten. Des Weiteren fällt nur für deren statische Konfiguration ein hoher offline Aufwand an. Klasse A Routen und Streamingfreigaben werden im Netzwerk dynamisch eingestellt.

Im Vergleich mit den Ergebnissen der Arbeiten aus Kapitel 3 ist zu sehen, dass die Latenz von AVB Nachrichten auch hier weit unter dem schlimmsten anzunehmenden Fall liegen. Nur die von [Queck \(2012\)](#) errechneten Latenzen, lassen sich durch die in dieser Arbeit durchgeführten Simulationen nicht nachvollziehen, weil sie weit über diesen liegen.

8.2 Ausblick

In Zukunft werden sich weitere Arbeiten in der CoRE Arbeitsgruppe mit der Kombination von AVB und TDMA basierter Kommunikation und mit AVB im Allgemeinen beschäftigen. Die Themen sind:

- **Erweiterung des TTE Stacks für Microcontroller um einen AVB Stack:**
Es wird der in der CoRE Arbeitsgruppe implementierte TTE Stack mit einem AVB Stack verbunden, so das jeweils eins oder beides zusammen genutzt werden kann. So kann das hier in der Simulation untersuchte Verhalten eines Endsystems in Hardware überprüft werden.
- **Hardwaresynthese eines TTE und AVB fähigen Switches:**
Auf einem FPGA wird ein Switch realisiert, der die Fähigkeiten von AVB und TTE vereint. In Verbindung mit dem Microcontroller kann so ein komplettes reales Netzwerk aufgebaut werden, dass mit diesen Nachrichtenklassen kommuniziert.

- **Architekturvergleich von AVB Netzwerken für das Automobil:**

Es werden verschiedene Netzwerktopologien für AVB Netzwerke auf ihre Auswirkungen auf die Leistung von AVB untersucht. Im weiteren Verlauf soll der fertige Entwicklungsstand von AVB Generation 2 der IEEE Time Sensitive Networking Group dafür genutzt werden.

- **Papier zur IEEE VNC:**

Im Zuge dieser Arbeit ist ein Paper entstanden (vgl. Meyer u. a. (2013)), das für die IEEE Vehicular Networking Conference eingereicht wurde.

Des Weiteren könnten die neuen AVB Komponenten im TTE4INET noch um die Unterstützung der Nachrichtenklasse B erweitert werden. Viele implementierte Komponenten wie der CBS können dafür genutzt werden. Der Aufwand beschränkt sich folglich auf den Einbau der Unterstützung in die Switche, Talker und Listener durch Klasse B Puffer, Identifikation am Port und dem Erweitern des bisher umgesetzten SRP Protokolls. So könnte in Zukunft auch das Verhalten von AVB Klasse B in diesem neuen Szenario simuliert und untersucht werden. Da es keine Garantie gibt, ob und wann BE Nachrichten ihr Ziel erreichen, muss die Zuverlässigkeit des SRP Protokolls in diesem Kontext weiter analysiert werden.

Die Entwicklung eines neuen AVB Standards durch die Time Sensitive Networking Group des IEEE wird aktuell vorangetrieben. Die beiden entstehenden Standards IEEE 802.1Qbu (vgl. IEEE 802.1 TSN Task Group (a)) und IEEE 802.1Qbv (vgl. IEEE 802.1 TSN Task Group (b)) werden neue Ansätze liefern um die Performanz von AVB zu steigern. Zu gegebener Zeit ist ein Vergleich der neuen Ansätze und deren Leistung, mit denen der Kombination aus TTE und AVB, sinnvoll.

Weitreichendere Schritte wären als nächstes die Untersuchung in echten Fahrzeugtopologien in der Simulation sowie in der Realität. Nur so kann am Ende abschließend geklärt werden, ob die Anforderungen die ein Auto an seine Kommunikation stellt, durch ein neues AVB mit TDMA Ansätzen erfüllt werden können.

Literaturverzeichnis

- [CoRE RG a] CoRE RG: *Communication over Real-time Ethernet*. – URL <http://core.informatik.haw-hamburg.de>
- [CoRE RG b] CoRE RG: *TTE for INET*. – URL <http://tte4inet.realmv6.org>
- [IEEE 802.1 TSN Task Group a] IEEE 802.1 TSN TASK GROUP: *IEEE 802.1Qbu - Frame Preemption*. – URL <http://www.ieee802.org/1/pages/802.1bu.html>
- [IEEE 802.1 TSN Task Group b] IEEE 802.1 TSN TASK GROUP: *IEEE 802.1Qbv - Enhancements for Scheduled Traffic*. – URL <http://www.ieee802.org/1/pages/802.1bv.html>
- [Imtiaz u. a. 2012] IMTIAZ, Jahanzaib ; JASPERNEITE, Jürgen ; WEBER, Karl: Approaches to reduce the latency for high priority traffic in IEEE 802.1 AVB networks. In: *9th IEEE International Workshop on Factory Communication Systems (WFCS 2012)*, 2012, S. 161–164. – ISBN 978-1-4673-0693-5
- [Institute of Electrical and Electronics Engineers 2009a] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: IEEE 802.1AB - IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery / IEEE. September 2009 (IEEE 802.1AB-2009). – Standard. – ISBN 978-0-7381-6038-2
- [Institute of Electrical and Electronics Engineers 2009b] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: IEEE 802.1Qav - IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks - Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams / IEEE. Dezember 2009 (IEEE 802.1Qav-2009). – Standard. – ISBN 978-0-7381-6143-3
- [Institute of Electrical and Electronics Engineers 2010] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: IEEE 802.1Qat - IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks - Amendment 14: Stream Reservation Protocol (SRP) / IEEE. September 2010 (IEEE 802.1Qat-2010). – Standard. – ISBN 978-0-7381-6501-1

- [Institute of Electrical and Electronics Engineers 2011a] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: IEEE 802.1AS - IEEE Standard for Local and metropolitan area networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks / IEEE. Februar 2011 (IEEE 802.1AS-2011). – Standard. – ISBN 978-0-7381-6536-3
- [Institute of Electrical and Electronics Engineers 2011b] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS: IEEE 802.1BA - IEEE Standard for Local and metropolitan area networks - Audio Video Bridging (AVB) Systems / IEEE. September 2011 (IEEE 802.1BA-2011). – Standard. – ISBN 987-0-7381-7639-8
- [Lim u. a. 2012] LIM, Hyung-Taek ; HERRSCHER, Daniel ; WATTL, Martin J. ; CHAARI, Firas: Performance analysis of the IEEE 802.1 ethernet audio/video bridging standard. In: *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*. New York : ACM-DL, März 2012, S. 27–36. – ISBN 978-1-4503-1510-4
- [Meyer u. a. 2013] MEYER, Philipp ; STEINBACH, Till ; KORF, Franz ; SCHMIDT, Thomas C.: *Extending IEEE 802.1 AVB with Time-triggered Shaping: Simulation based Assessment of the Effects of Additional Synchronous Traffic*. 2013
- [NYFEGA Elektro-Garage AG] NYFEGA ELEKTRO-GARAGE AG: *Pheaton Bordnetz*. – URL http://www.nyfega.ch/bilder/phaeton_bordnetz.jpg. – Zugriffsdatum: 2. September 2013
- [OMNeT++ Community a] OMNeT++ COMMUNITY: *INET Framework for OMNeT++ 4.0*. – URL <http://inet.omnetpp.org/>
- [OMNeT++ Community b] OMNeT++ COMMUNITY: *OMNeT++ 4.2.2*. – URL <http://www.omnetpp.org>
- [Queck 2012] QUECK, Rene: Analysis of Ethernet AVB for Automotive Networks using Network Calculus. In: *2012 IEEE International Conference on Vehicular Electronics and Safety*, 2012
- [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/>

- [Steinbach u. a. 2012] STEINBACH, Till ; LIM, Hyung-Taek ; KORE, Franz ; SCHMIDT, Thomas C. ; HERRSCHER, Daniel ; WOLISZ, Adam: Tomorrow's In-Car Interconnect? A Competitive Evaluation of IEEE 802.1 AVB and Time-Triggered Ethernet (AS6802). In: *2012 IEEE Vehicular Technology Conference (VTC Fall)*. Piscataway, New Jersey : IEEE Press, September 2012. – ISSN 1090-3038
- [Weibel 2008] WEIBEL, Hans: *Audio/Video Bridging das Realtime-Ethernetprotokoll des IEEE 802*. 2008

Glossar

- AVB Audio-Video Bridging, Seite 2
- BE Best-Effort, Seite 11
- CBS Credit Based Shaper, Seite 8
- CMI Class Measurement Interval, Seite 8
- CoRE Communication over Realtime Ethernet, Seite 1
- RC Rate-Constrained, Seite 13
- SRP Stream Reservation Protocol, Seite 6
- TDMA Time Division Multiple Access, Seite 2
- TT Time-Triggered, Seite 13
- TTE Time-Triggered Ethernet, Seite 5

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

Hamburg, 1. Oktober 2013 Philipp Meyer