



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung - Seminar - WS 12/13

Jan Kamieth

Scheduling in einem Time-Triggered Ethernet System

Jan Kamieth

Thema der Ausarbeitung - Seminar - WS 12/13

Scheduling in einem Time-Triggered Ethernet System

Stichworte

Echtzeit Ethernet, TTEthernet, Bussysteme, Automotive Anwendungen, Scheduling

Betreuende Prüfer

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

Betreuer

Prof. Dr. Franz Korf

Title of the paper

Scheduling of a Time-Triggered Ethernet System

Keywords

Real-time Ethernet, TTEthernet, Bussystems, Automotive Applications, Scheduling

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	1
1.3	Inhaltlicher Aufbau der Arbeit	2
2	Rückblick	3
2.1	TT-Ethernet	3
2.1.1	Time-Triggered Nachrichten	3
2.1.2	Perioden	4
2.2	Anforderungen	5
2.3	Schedulingverfahren	6
2.3.1	Heuristiken	6
2.3.2	Model-Checking	6
3	Umsetzung im Projekt	7
3.1	Integration	7
3.1.1	FIBEX	7
3.2	Scheduling	8
4	Zusammenfassung	12
4.1	Ausblick	12
4.2	Risiken	12
	Literaturverzeichnis	14

1 Einleitung

1.1 Motivation

Die Notwendigkeit für neue Kommunikationssysteme im Automobilbereich ist durch den rasanten Anstieg von elektronischen Komponenten ohne Zweifel gegeben. Die kommenden Neuerungen, wie Steer-By-Wire und Break-By-Wire Systeme, stellen besondere Anforderungen an die Bussysteme, wie Fehlertoleranz and Determinismus. In Anwendungen 1 und Anwendungen 2 wurde dieser Umstand deutlich gemacht und auch, dass das Time-Trigged Ethernet System eine Lösung für diese Herausforderungen darstellt, so dass hier nicht noch einmal darauf eingegangen werden wird.

Stattdessen soll hier hervorgehoben werden, warum es sich lohnt, sich mit dem Thema des Scheduling innerhalb von TT-Ethernet Systemen zu beschäftigen. Denn während es für etablierte Bussysteme, wie Flexray, bereits Tools gibt, um ein komplettes System zu designen, sieht es im TT-Ethernet Bereich ganz anders aus. Die Berechnungen, die erforderlich sind, um ein Automobil mit seinen dutzenden Steuergeräten und hunderten von Nachrichten (vgl. Nolte u. a. (2005)) zu einem funktionierendem Ganzen zu formen, lassen sich nicht ohne die Hilfe der Mathematik und Informatik durchführen. Daher ist es erforderlich, ein automatisches und algorithmengestütztes Verfahren zu entwickeln, dass einem beim Design solcher Systeme unterstützt.

1.2 Zielsetzung

Im Zuge von Anwendungen 1 wurde das Core Projekt näher vorgestellt und das in diesem Zusammenhang auftretende Scheduling Problem bei Time-Triggered Ethernet Netzwerken beschrieben. Im Seminar Anwendungen 2 wurde der Blick dann auf die aktuelle Forschung geworfen, um ein Bild dafür zu bekommen, wie andere Gruppen das Schedulingproblem bei ähnlichen Problemstellungen zu lösen versuchen.

In dieser Arbeit geht es nun darum, die Erkenntnisse und Entscheidungen, die aus dieser Vorarbeit gezogen werden konnten, darzulegen. Des weiteren wird der aktuelle Stand der eigenen Implementation erläutert und ein Ausblick darauf gegeben, in welche Richtung die Entwicklung in Hinsicht auf der Masterarbeit weitergehen soll.

1.3 Inhaltlicher Aufbau der Arbeit

Im zweiten Kapitel dieser Arbeit wird ein Rückblick auf die bisher getätigte Arbeit geworfen. Dabei werden auf der einen Seite noch einmal die wichtigsten Aspekte des TT-Ethernet Systems beschrieben, die für das Scheduling von Bedeutung sind und auf der anderen Seite eine kurze Zusammenfassung der Ergebnisse aus Anwendungen 1 und Anwendungen 2 und welche Entscheidungen anhand dieser getroffen wurden.

Das dritte Kapitel beschäftigt sich mit dem aktuellen Stand des Projektes. Wie wird das Scheduling in die Toolchain von Core integriert und in welchem Umfang ist das derzeitig implementierte Scheduling umgesetzt. Dafür wird grob beschrieben, wie das Scheduling aufgebaut wird und welche Algorithmen zum Einsatz kommen.

Im letzten Abschnitt wird ein Fazit gezogen in Hinblick auf die aufgetretenen Schwierigkeiten und erkannten Risiken. Zuallerletzt wird noch ein Ausblick auf die folgenden Arbeiten in Hinsicht auf die Masterarbeit gegeben.

2 Rückblick

In diesem Kapitel werden zunächst die für das Scheduling wichtigsten Eigenschaften des TT-Ethernet Systems vorgestellt. Danach wird näher auf die Anforderungen eingegangen, die sich als sinnvoll er ergeben haben, um dann zum Schluss noch einmal kurz auf die beiden Schedulingverfahren einzugehen, deren Umsetzung in Betracht gezogen wurde.

2.1 TT-Ethernet

Innerhalb der TT-Ethernet Spezifikation sind primär die Definitionen der Time-Triggered Nachrichten und der Perioden für das Scheduling von Interesse.

2.1.1 Time-Triggered Nachrichten

Das Kernstück der TT-Ethernet Spezifikation bilden die Time-Triggered Nachrichten (vgl. Kopetz u. a. (2005)). Sie besitzen innerhalb des Netzwerks die höchste Priorität und verdrängen alle anderen auftretenden Nachrichten. Um die Übertragung dieser Nachrichten sicher zu stellen, besitzt jede TT-Nachricht an jedem beteiligten Sender, Empfänger und Switch ein fest definiertes Sende-, bzw. Empfangsfenster. Innerhalb dieses Zeitraumes werden alle anderen Nachrichten auf dem entsprechenden Gerät blockiert.

Versendet werden diese Nachrichten auf sogenannten virtuellen Links, die aus mehreren physikalischen Links bestehen und einer Nachricht auch mehrere Empfänger zuordnen können. Die Abbildung 2.1 zeigt ein kleines Netzwerk in dem die ECU1 über einen virtuellen Link eine TT-Nachricht an die Empfänger ECU2 und ECU3 sendet.

Im Rahmen des Scheduling, dass für dieses Nachrichten statisch zur Designzeit berechnet wird, müssen auch die Tasks, die auf den ECUs ausgeführt werden, betrachtet werden. Da die TT-Nachrichten erst als Ergebnis einer ausgeführten Task zum Versenden bereit liegt, haben wir es nicht nur mit einem Nachrichten Scheduling zu tun, sondern müssen auch gegebenenfalls beachten, in welcher Reihenfolge die Tasks auf den ECUs ausgeführt werden.

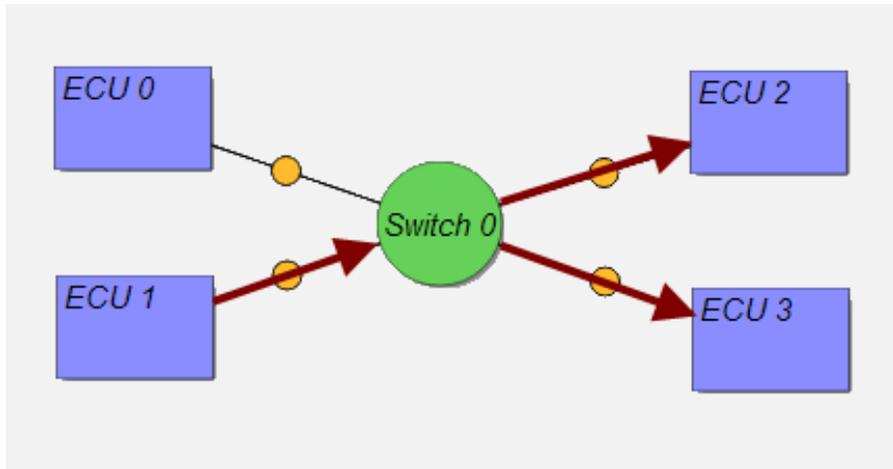


Abbildung 2.1: Netzwerk mit 1 virtuellen Link

2.1.2 Perioden

Die zweite Komponente, die Auswirkungen auf das Scheduling hat, sind die Perioden (vgl. Kopetz u. a. (2005)). Jeder Netzwerk Cluster besitzt eine Master-Periode. Diese Periode legt fest, in welchem Zeitraum jede TT-Nachricht mindestens einmal versendet wird. Zusätzlich können beliebig viele Sub-Perioden existieren. Diese Sub-Perioden sind kürzer als die Master Periode und sind immer ganze Teiler der Master-Periode. Jeder virtuelle Link wird einer Periode zugeordnet, so dass auf diesem Weg festgelegt wird, wie oft eine TT-Nachricht innerhalb der Master-Periode versendet wird. Zusätzlich kann jede der Sub-Perioden phasenverschoben sein, so dass ein Offset zu der Master-Periode definiert werden kann.

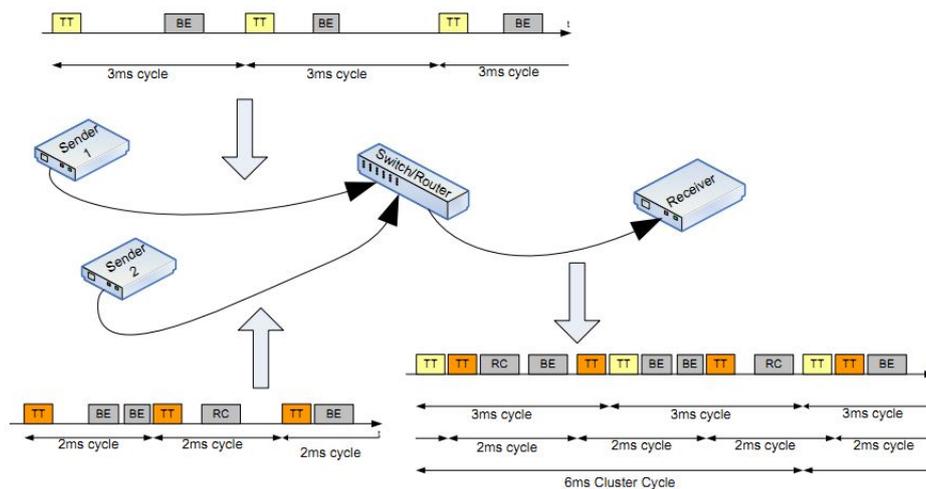


Abbildung 2.2: Beispiel für einen Cluster Cycle

Die Abbildung 2.2 demonstriert ein Netzwerk Cluster mit zwei Sendern und einem Empfänger (vgl. Bartols (2010)). Der Sender 1 hat eine Sub-Periode von 3 ms, in der zu Beginn der Periode eine TT-Nachricht gesendet wird. Sender 2 hat eine Sub-Periode von 2 ms, so dass alle 2 ms eine TT-Nachricht versendet wird. Zusätzlich werden noch Rate-Constrained (RC) und Best-Effort (BE) Nachrichten versendet, deren Betrachtung aber zunächst für das Scheduling außer Acht gelassen wird. Diese Sub-Perioden ergeben nun die Master-Periode von 6 ms, in der die erste TT-Nachricht zweimal und die zweite TT-Nachricht dreimal versendet wird. Beim Empfänger sieht man, dass die Periode des Sender 2 einen Offset hat. Diese Phasenverschiebung wird benötigt, da die beiden gesendeten TT-Nachrichten sonst beim Sendeversuch des Switches kollidieren würden. Diese Offsets zu berechnen ist Aufgabe des Scheduling.

2.2 Anforderungen

In diesem Abschnitt wird festgehalten, welche Anforderungen gewählt wurden, die das Scheduling erfüllen muss. Die beiden Elemente, für die Zeiten berechnet werden müssen, sind der Task, der auf einer ECU ausgeführt wird und die TT-Nachricht, die durch den Task generiert wird.

Für den Task wird nur eine Deadline definiert, also der Zeitpunkt, an dem die Ausführung des Tasks spätestens beendet sein muss (vgl. Kenfack (2012) S. 48) und die TT-Nachricht zum Versenden bereit liegen muss. Die Deadline des Tasks ist somit auch die Arrival Time der TT-Nachricht, also der Zeitpunkt, an dem die Nachricht zum Versenden bereit ist. Hinzu kommen als Anforderung für die TT-Nachricht eine maximale Latenz, die Zeitspanne, die zum Versenden der TT-Nachricht verwendet werden darf und die Deadline, der Zeitpunkt, an dem die TT-Nachricht spätestens beim Empfänger angekommen sein muss (vgl. Kenfack (2012) S. 48).

Diese Anforderungen lassen sich auf 2 Elemente reduzieren, nämlich auf die maximale Latenz und die Deadline der TT-Nachricht. Anhand dieser beiden Werte lässt sich die Arrival Time der TT-Nachricht berechnen, die, wie weiter oben beschrieben, gleichbedeutend ist mit der Deadline des entsprechenden Tasks.

Eine Besonderheit ist, dass die Anforderungen für jeden einzelnen Empfänger definiert werden können. Das heißt, dass die Deadline nicht für den virtuellen Link, also die TT-Nachricht, definiert wird, sondern für jeden Empfänger des virtuellen Links getrennt, was eine zu beachtende Komplexitätssteigerung beim Scheduling bedeutet.

2.3 Schedulingverfahren

Im Rahmen des Anwendungen 2 Vortrages wurden zwei Schedulingverfahren vorgestellt, die hier noch einmal kurz vorgestellt werden sollen.

2.3.1 Heuristiken

Die Berechnung durch Heuristiken wurde unter anderem anhand einer Arbeit vorgestellt, die sich mit einem Schedulingverfahren beschäftigt, um den Buszugriff innerhalb eines verteilten Systems zu optimieren (vgl. Eles u. a. (2000)). Das Kernstück war ein List-Scheduling Algorithmus, der Anhand eines Taskgraphen die Zeitpunkte für die Ausführung der Tasks bestimmt hat. Heuristiken sind ein analytischer Lösungsversuch, die leicht anzupassen sind und beim Berechnen relativ kostengünstig sind. Der Nachteil ist, dass sie in der Regel nicht das optimale Ergebnis finden.

2.3.2 Model-Checking

Model-Checking steht für ein Verfahren, bei dem ein System gegen eine Spezifikation geprüft wird. Vorgestellt wurden der SPIN (vgl. Gu u. a. (2007b)) und der UPPAAL (vgl. Gu u. a. (2007a)) Model-Checker mit denen das System modelliert werden kann. Die zu erfüllenden Eigenschaften können dann mit einer Temporal Logic beschrieben werden, anhand derer dann eine Lösung berechnet werden kann. Die Lösung mittels Model-Checking ist komplexer zu implementieren, benötigt mehr Ressourcen bei der Berechnung und skaliert schlecht bei größeren Systemen. Der Vorteil ist, dass der gesamte Lösungsraum abgesucht wird und somit bessere Ergebnisse gefunden werden können, als mit der Heuristik.

3 Umsetzung im Projekt

In diesem Kapitel wird erläutert, wie das Scheduling im Umfeld der Core Projektgruppe eingesetzt wird und wie der aktuelle Stand der Implementation aussieht.

3.1 Integration

Das Scheduling wird derzeit benötigt, um die Simulation von TT-Ethernet Netzwerken mit Daten zu versorgen. Um dies zu erreichen, wird von Drittpersonen ein Netzwerk definiert, welches dann einem Scheduling unterzogen wird, um letztendlich mittels Omnet++ (vgl. omnet-web) simuliert zu werden. Als Austauschformat zwischen den einzelnen Werkzeugen kommt das FIBEX Format zum Einsatz, welches im folgenden Abschnitt vorgestellt wird.

3.1.1 FIBEX

FIBEX steht für *Field Bus Exchange Format* (vgl. fibex-web) ist ein XML-basierter Standard zur Beschreibung von Steuergeräte-Netzwerken. Entwickelt wurde es von der *Association for Standardisation of Automation and Measuring Systems (ASAM e. V.)* und wird von der Automobilindustrie hauptsächlich dazu verwendet, um Flexray-Netzwerke zu beschreiben. Florian Bartols hat, im Zuge seiner Arbeit in der Core Gruppe, diesen Standard erweitert, um damit auch TT-Ethernet Netzwerke beschreiben zu können.

Listing 3.1: Virtual Link in Fibex

```
1 <ttethernet:VIRTUAL-LINK xsi:type="ttethernet:VIRTUAL-LINK-TT-TYPE" ID="VL105">
2   <ho:SHORT-NAME>TTVirtualLink3</ho:SHORT-NAME>
3   <ttethernet:VLINK-SENDER>
4     <fx:ECU-REF ID-REF="ECU3-CL"/>
5   </ttethernet:VLINK-SENDER>
6   <ttethernet:VLINK-RECEIVER>
7     <fx:ECU-REF ID-REF="ECU1-MA"/>
8     <fx:ECU-REF ID-REF="ECU5-CL"/>
9   </ttethernet:VLINK-RECEIVER>
10  <ttethernet:VL-IDENTIFIER>105</ttethernet:VL-IDENTIFIER>
11  <ttethernet:VIRTUAL-LINK-PERIOD-REF ID-REF="VL-PERIOD3"/>
12 </ttethernet:VIRTUAL-LINK>
```

Für das Scheduling wird die physikalische Beschreibung des Netzwerkes, die virtuellen Links und die Anforderungen benötigt. Das Listing 3.1 zeigt die Definition eines virtuellen Links im FIBEX Format. Die TT-Nachricht mit der ID 105 hat zwei Empfänger und referenziert auf die Sub-Periode 3, die in der gleichen FIBEX Datei konfiguriert ist.

Durch das Scheduling wird die FIBEX Datei um die Port Timings ergänzt, also die Empfangsfenster, an denen die TT-Nachricht an den physikalischen Ports auftritt. Das nächste Listing demonstriert dies beispielhaft für einen Port für den eben definierten virtuellen Link.

Listing 3.2: Virtual Link in Fibex

```
1 <ttethernet:OUTPUT-PORT ID="OP-SW1-ECU5-VL105" xsi:type="ttethernet:ECU-PORT-TYPE">
2   <fx:FRAME-TRIGGERING-REF ID-REF="FT-VL105"/>
3   <ttethernet:TIMINGS>
4     <ttethernet:ABSOLUTELY-SCHEDULED-TIMING>
5       <ttethernet:WINDOW-START>2100000</ttethernet:WINDOW-START>
6       <ttethernet:WINDOW-END>2300000</ttethernet:WINDOW-END>
7     </ttethernet:ABSOLUTELY-SCHEDULED-TIMING>
8   </ttethernet:TIMINGS>
9 </ttethernet:OUTPUT-PORT>
```

3.2 Scheduling

Dieser Abschnitt beschäftigt sich mit dem eigentlichen Prozess des Scheduling. Die Entscheidung ist zugunsten einer Heuristik gefallen, um den Aufwand gering zu halten, unterschiedliche Algorithmen zu testen.

Der erste implementierte Ansatz basiert auf einem List-Scheduling-Algorithmus, der sowohl für das Task-Scheduling auf den ECUs, als auch für das Nachrichten Scheduling innerhalb des Netzwerkes einsetzbar ist. Um mit dem Scheduling zu beginnen, müssen die Daten aus dem FIBEX Format weiter aufbereitet werden.

Zuerst müssen alle Daten extrahiert werden, die das TT-Ethernet System auf logischer Ebene beschreiben. Dazu gehören hauptsächlich die Perioden und die virtuellen Links. Danach wird, anhand der Informationen zu ECUs und Switches, das physikalische Netzwerk aufgebaut und für jeden virtuellen Link ein Pathfinding durchgeführt, um festzulegen über welche physikalischen Links die TT-Nachricht versendet werden muss. Zuletzt wird für jede TT-Nachricht anhand der definierten Anforderungen ein Dummy-Task erzeugt und der Taskgraph aufgebaut, der die Abhängigkeiten innerhalb des Systems abbildet. Anhand dieses Taskgraphen können nun alle Tasks, bzw. Nachrichten ausgewählt werden, die bereit sind ausgeführt, bzw. versendet zu werden. Sollten auf einer ECU mehrere Tasks bereit sein, bzw. auf einem Link mehrere Nachrichten, wird das letztendlich gewählte Element durch zwei verschiedene Kriterien bestimmt, dem *Earliest Deadline First (EDF)* und dem *Longest Path First (LPF)* Verfahren.

Der Einsatz dieser Verfahren soll mit der Hilfe eines Beispiels näher erläutert, welches sich in der Abbildung 3.1 widerspiegelt.

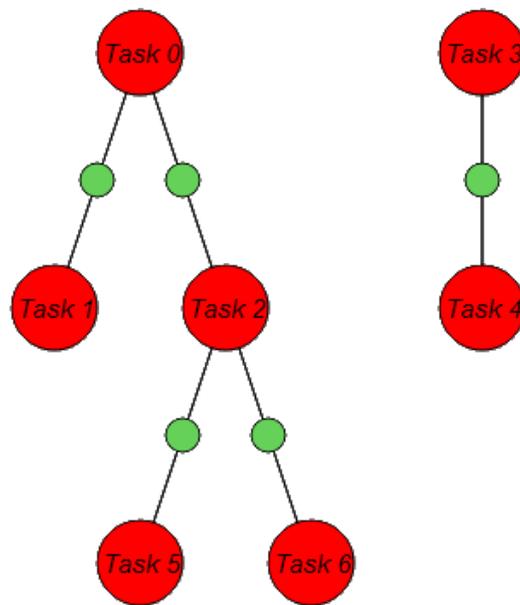


Abbildung 3.1: Taskgraph

Das Beispiel beinhaltet einen Taskgraphen mit 6 Tasks und 2 unabhängigen Zweigen. Task2 hängt von Task0 ab und so weiter. Wir gehen davon aus, dass Task0 und Task3 auf der selben ECU ausgeführt werden und bereit sind zur Ausführung, also in der *Ready List* sind. Da die beiden Tasks keine Abhängigkeit untereinander haben, muss ausgewählt werden, welcher Task zuerst zum Zuge kommt. Zuerst wird mittels *EDF* geprüft, welcher der beiden Tasks die kritischere Deadline hat. Sollte Task3 seine Deadline nicht mehr einhalten können, wenn zuerst Task0 ausgeführt wird, wird Task3 bevorzugt. Sollte bei beiden Tasks kein Problem mit der Deadline auftreten, wird die Entscheidung mit der Hilfe des *LPF* Verfahrens getroffen. In dem vorliegenden Beispiel folgt nach Task0 ein längerer Pfad als nach Task3. Der Pfad summiert sich aus den Zeiten, die für die Ausführung der Tasks, bzw für das versenden von Nachrichten benötigt werden, die nach dem entsprechendem Task folgen. Da der Pfad nach Task0 länger ist, also an dessen Ausführung die meisten Elemente abhängen, wird der Task0 bevorzugt. Nach der Ausführung von Task0 werden nun die nachfolgenden Nachrichten und Tasks in die *Ready List* übernommen. So wird gewährleistet, dass die abhängigen Tasks, bzw. Nachrichten so frühzeitig wie möglich in das Scheduling mit einbezogen werden.

Während sich die Tasks auf diese Weise gut priorisieren lassen, kommt beim Festlegen der Sendefenster für die TT-Nachrichten ein weiterer Komplexitätsgrad hinzu. Nehmen wir das Netzwerk aus Abbildung 3.2 mit dem dazu gehörendem Taskgraphen.

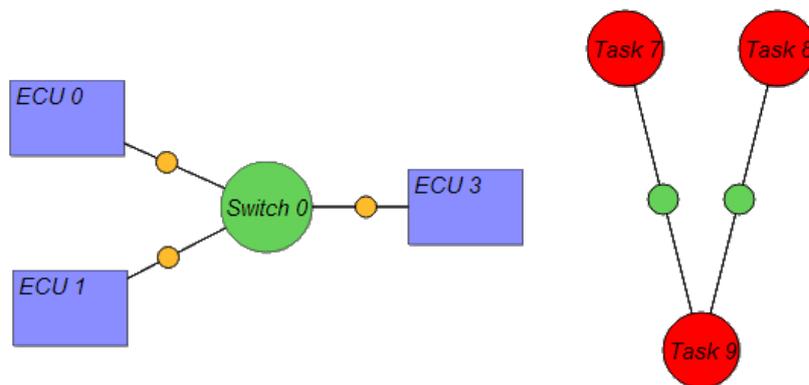


Abbildung 3.2: Netzwerk mit Taskgraph

Das Netzwerk besteht aus drei ECUs und einem Switch. Task7 läuft auf ECU1, Task8 auf ECU0 und Task9 auf ECU3. Task7 und Task8 haben keinen Vorgänger, sind also zur Ausführung bereit und haben die gleiche Laufzeit. Task9 hat Abhängigkeiten zu Task7 und Task8 und kann erst ausgeführt werden, wenn beide Nachrichten empfangen wurden. Weiterhin wird festgelegt, dass die Nachrichten beide mit einer Periode von 2 ms versendet werden.

Werden nun Task7 und Task8 parallel ausgeführt, liegen die Nachrichten zur gleichen Zeit zum Versenden bereit und kommen demnach gleichzeitig beim Switch an, der die Nachrichten nun an die ECU3 senden will. Es würde sich folgendes Bild 3.3 ergeben.

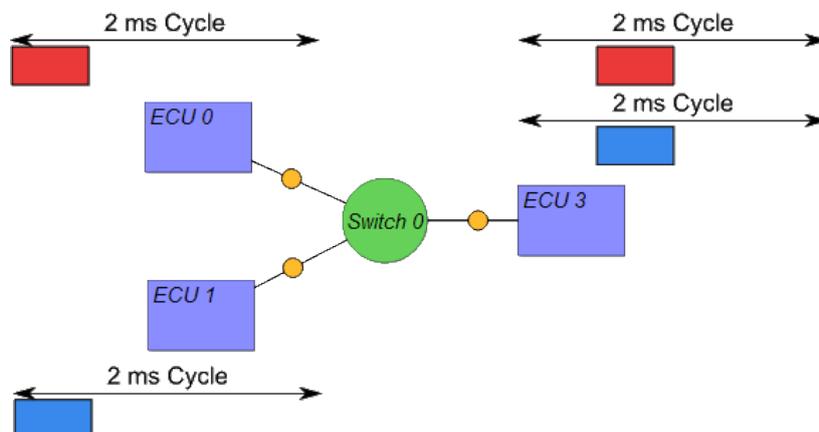


Abbildung 3.3: Schedule 1

Es ist sofort zu sehen, dass die beiden TT-Nachrichten von ECU0 und ECU1 auf dem Link vom Switch zur ECU3 im gleichen Sendefenster liegen. Dies führt zu einem Konflikt, einem ungültigem Schedule. Es muss daher nach einem freiem Fenster gesucht werden. Dafür wird

ein Backstepping eingesetzt. Eine früher getroffene Entscheidungen wird revidiert, so dass sich die Ausgangslage ändert und gegebenenfalls zu einem gültigem Ergebnis führt. Schedules wir jetzt zum Beispiel die Nachricht von ECU0 nicht direkt am Anfang der Periode, sondern verzögern das Senden um die Zeit, die die andere Nachricht zum Senden benötigt, kommen die beiden Nachrichten nicht gleichzeitig am Switch an, so dass es zu keiner Überschneidung kommt. Es würde sich der in Abbildung 3.4 gezeigte Schedule ergeben.

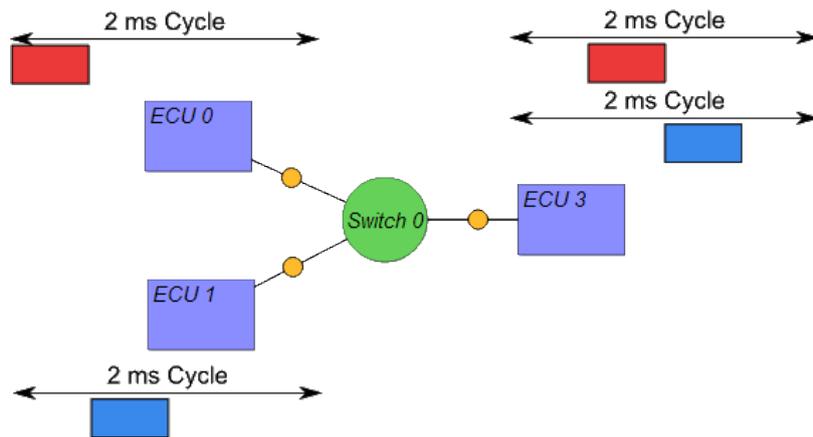


Abbildung 3.4: Schedule 2

Ein ähnliches Problem ergibt sich, wenn zwei virtuelle Links mit unterschiedlichen Periodenlängen auf einem physikalischen Link überlagern. Auch in diesem Fall wird mittels Backstepping so lange nach passenden freien Sendefenstern gesucht, bis ein gültiger Schedule gefunden wurde.

4 Zusammenfassung

Die Entwicklung in der Automobilbranche schreitet schnell voran. Es wurde dargelegt, dass neue flexible Bussysteme gebraucht werden, um die neuen Anforderungen erfüllen zu können. Das Time-Triggered Ethernet System ist ein potenzieller Kandidat, um diese Herausforderungen zu bestehen. Ein wesentlicher Teil dieses Systems hängt dabei vom Scheduling ab. Im Zuge dieses Master Themas wurde sich näher mit diesem Problem beschäftigt und ein Blick auf ähnliche Forschungsprojekte geworfen, um einen Überblick über mögliche Lösungen zu bekommen.

Auf der Basis dieses Wissens wurde ein eigener Lösungsentwurf implementiert, der direkt auf die Eigenschaften des Time-Triggered Ethernet Systems zugeschnitten ist. Dieses erste rudimentäre Schedulingverfahren wird bereits im Umfeld der Core Gruppe eingesetzt und unterstützt die Weiterentwicklung der Simulation, indem es nun ermöglicht wird, auch größere und komplexere Netzwerke mit einem gültigen Scheduling zu designen.

4.1 Ausblick

Für die noch folgende Projekt- und Masterarbeit stehen die eigentlichen Schedulingalgorithmen im Vordergrund. Während bisher nur ein Basisalgorithmus implementiert wurde, um erste Ergebnisse testen zu können, wird es nun darum gehen verschiedene Algorithmen umzusetzen und miteinander zu vergleichen. Dabei geht es nicht nur darum, sich an klassischen Lösungen zu orientieren, wie sie vor allen Dingen beim Task-Scheduling in verteilten Systemen eingesetzt werden, sondern auch andere Verfahren, wie zum Beispiel Ameisenalgorithmen auf ihre Tauglichkeit zu überprüfen. Dazu gehört auch, dass untersucht wird, welchen Einfluss die speziellen Eigenschaften von TT-Ethernet auf die Komplexität des Algorithmen haben, wie zum Beispiel die unterschiedlichen Perioden der virtuellen Links.

4.2 Risiken

Zu den bestehenden Risiken gehört an erster Stelle natürlich die knappe Zeit. Die Lektüre der unterschiedlichen Lösungsansätze ist dabei nur der erste Schritt. Die Besonderheiten vom

Time-Triggered Ethernet machen es in jedem Fall erforderlich, bestehende Algorithmen anzupassen oder komplett neu zu überdenken. Erschwert wird dies durch die hohe Komplexität des Systems. Mehrere Nachrichtenklassen, virtuelle Links mit mehreren Empfängern, die unterschiedliche Anforderungen an die Nachricht stellen. Hinzu kommen Perioden für jeden virtuellen Link, die sich in Länge und Phasenverschiebung unterscheiden. Ein weiteres Problem ist, dass bisher noch kein reales Modell eines Automobils erstellt wurde, an dem das Scheduling getestet werden kann. Die Spezifikationen und Anforderungen der Systeme, an denen bisher entwickelt wird, sind daher noch von sehr theoretischer Natur, so dass die Gefahr besteht, an der Realität vorbei zu entwickeln.

Literaturverzeichnis

- [fibex-web] : *ASAM MCD-2 NET (market name: FIBEX)*. – URL http://www.asam.net/nc/home/standards/standard-detail.html?tx_rbwmbmasamstandards_pi1%5BshowUid%5D=561
- [omnet-web] : *OMNeT++ Network Simulation Framework*. – URL <http://http://www.omnetpp.org/>
- [Bartols 2010] BARTOLS, Florian: *Leistungsmessung von Time-Triggered Ethernet Komponenten unter harten Echtzeitbedingungen mithilfe modifizierter Linux-Treiber*. 2010
- [Eles u. a. 2000] ELES, Petru ; DOBOLI, Alex ; POP, Paul ; PENG, Zebo: Scheduling with bus access optimization for distributed embedded systems. In: *Very Large Scale Integration (VLSI) Systems, IEEE Transactions* 8 (2000), Nr. 5, S. 472–491
- [Gu u. a. 2007a] GU, Zonghua ; HE, Xiuqiang ; YUAN, Mingxuan: Optimal Static Task Scheduling on Reconfigurable Hardware Devices Using Model-Checking. In: *Real Time and Embedded Technology and Applications Symposium (2007)*, S. 32 – 44
- [Gu u. a. 2007b] GU, Zonghua ; HE, Xiuqiang ; YUAN, Mingxuan: Optimization of Static Task and Bus Access Schedules for Time-Triggered Distributed Embedded Systems with Model-Checking. In: *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE (2007)*, S. 294–299
- [Kenfack 2012] KENFACK, H. D.: *Designmigrationsstrategien von FlexRay nach Time-Triggered Ethernet*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2012
- [Kopetz u. a. 2005] KOPETZ, H. ; ADEMAJ, A. ; GRILLINGER, P. ; STEINHAMMER, K.: The time-triggered Ethernet (TTE) design. In: *Object-Oriented Real-Time Distributed Computing, 2005. ISORC 2005. Eighth IEEE International Symposium on*, May 2005, S. 22–33
- [Nolte u. a. 2005] NOLTE, Thomas ; HANSSON, Hans ; LO BELLO, Lucia: Automotive communications-past, current and future. In: *Emerging Technologies and Factory Automation (2005)*, S. 985 – 992