

Zeitgesteuerte Kommunikationsschnittstellen in unterschiedlichen Anwendungskontexten

Raimund Kirner¹[0000–0003–3921–6813] and Peter Puschner²[0000–0002–2495–0778]

¹ Department of Computer Science, University of Hertfordshire, Hatfield, UK

`r.kirner@herts.ac.uk`

² TU Wien, Vienna, Austria

`peter@vmars.tuwien.ac.at`

Zusammenfassung Zeitgesteuerte Kommunikationsprotokolle bieten eine Vielzahl von Vorteilen für die Konstruktion von sicherheitskritischen Echtzeitapplikationen: der Nachrichtentransport ist zeitlich determiniert und die Realisierung von Fehlertoleranz durch die Replikation von Knoten bzw. des Netzwerks kann sehr einfach umgesetzt werden. Außerdem unterstützt die Spezifikation des statischen, über die Zeit vorhersagbaren Zugriffsverhaltens der Netzwerkschnittstelle die Composability von Netzwerkkomponenten im Zeitbereich. In dieser Arbeit zeigen wir, wie Echtzeitapplikationen in Abhängigkeit von ihren Anforderungen die Schnittstelle eines zeitgesteuerten Kommunikationssystems auf unterschiedliche Weise verwenden und von den Eigenschaften der zeitgesteuerten Kommunikation profitieren können.

1 Einleitung

Zeitgesteuerte Kommunikationssysteme sind besonders für die Kommunikation in verteilten, zuverlässigen Echtzeitsystemen geeignet [7]: In zeitgesteuerten Systemen werden Übertragungszeitpläne für alle Nachrichten vor der Laufzeit definiert, sodass die Übertragungszeiten und Bandbreiten für alle Nachrichten bereits vor der Laufzeit garantiert werden können. Die Inhalte der zeitgesteuerten Zustandsnachrichten werden an den Kommunikationsendpunkten wie Programmvariablen geschrieben und gelesen, die eine bekannte Gültigkeitsdauer und Aktualisierungsrate haben. Im Gegensatz zu ereignisgesteuerten Systemen sind die Knoten von zeitgesteuerten Systemen daher weder unbekanntem Empfangszeitpunkten von Nachrichten noch einer unbekanntem Last und damit einem unbekanntem Verarbeitungsdruck durch die Nachrichten aus dem Kommunikationssystem ausgesetzt [8]. Die Verwendung von Zustandsnachrichten unterstützt zudem die Bereitstellung von Fehlertoleranz durch räumliche und zeitliche Nachrichtenredundanz.

Das Buch von Kopetz [7] gibt einen guten und detaillierten Überblick über die Funktionsweise und Eigenschaften von zeitgesteuerten Kommunikationssystemen. Realisierungen zeitgesteuerter Kommunikationsprotokolle finden sich in TTP [5], TTEthernet [4] und FlexRay [2]. In der Literatur findet man für den Zugriff von Applikationen auf die zeitgesteuerten Nachrichten neben dem synchro-

nen, zeitgesteuerten Zugriff auch asynchrone Zugriffsmethoden, wie z.B. in [9] (NBW) oder [1].

Während wir in unseren früheren Arbeiten [11, 3] die Eigenschaften des asynchronen bzw. synchronen Zugriffs von Applikationen auf zeitgesteuerte Kommunikationsdaten systematisch analysieren, beleuchtet die vorliegende Arbeit, wie diese beiden Zugriffsmethoden von unterschiedlichen Applikationen genutzt werden können. Die Abschnitte 2 und 3 fassen die Funktionsweise von zeitgesteuerter Kommunikation und die Eigenschaften der beiden Zugriffsmethoden zusammen, während die verschiedenen Anwendungen der Zugriffsmethoden im Abschnitt 4 präsentiert werden.

2 Zeitgesteuerte Kommunikation

Ein zeitgesteuertes Kommunikationssystem ist ein autonomes Teilsystem eines verteilten Echtzeit-Computersystems, das Zustandsnachrichten zwischen den Knoten des Computersystems in einer zeitlich vorhersehbaren Weise transportiert [6]. Zeitgesteuerte Nachrichten haben eine bekannte Sendezeit und werden periodisch versendet. Das Kommunikationssystem transportiert die Nachrichten von ihrem Senderknoten zu einem oder mehreren Empfängerknoten, indem es einen statischen Nachrichtenübertragungsplan, der zur Entwurfszeit erstellt wird, interpretiert. Das Uhrensynchronisationsservice des Kommunikationssystems stellt dem verteilten System eine globale Zeit zur Verfügung. Die Kommunikationsschnittstellen, die das Kommunikationssystem mit den Knoten verbinden, verwenden diese global verfügbare Zeit, um ihre Nachrichtensende- und -empfangsoperationen entsprechend dem Nachrichtenübertragungsplan zu koordinieren. Darüberhinaus können die Applikationsrechner der Knoten die Schnittstelle so programmieren, dass sie (periodische) Zeitinterrupts erzeugt. Dies ermöglicht es den Applikationsrechnern, ihre Operationen mit der globalen Zeit zu synchronisieren (siehe z.B. [10]).

Abbildung 1 skizziert die Struktur eines zeitgesteuerten verteilten Computersystems. Die Knoten des verteilten Systems bestehen aus den Applikationsrechnern und Schnittstellenkomponenten. Auf der anderen Seite bilden die Schnittstellen aller Knoten und das Kommunikationsnetzwerk zusammen das zeitgesteuerte Kommunikationssystem. Die vertikalen, roten unterbrochenen Linien durch die Schnittstellen markieren die Trennlinie zwischen den Netzwerknoten und dem Kommunikationssystem.

Da zeitgesteuerte Kommunikationssysteme Zustandsnachrichten verwenden, muss, im Unterschied zur ereignisgesteuerten Kommunikation, nicht jede Nachricht genau einmal von jedem Empfänger interpretiert werden. Damit kann man in zeitgesteuerten Systemen den Empfängern von Nachrichten ein hohes Maß an Autonomie bei der Kontrolle der zeitlichen Abläufe ihrer lokalen Aktionen (z.B. durch die Wahl einer geeigneten Schedulingstrategie) einräumen. Auf der anderen Seite können sich die Applikationsrechner die vom zeitgesteuerten Kommunikationssystem aufgebaute Zeitbasis und das bekannte Übertragungsmuster der Nachrichten zunutze machen, um ihre eigenen lokalen Aktionen, wie auch

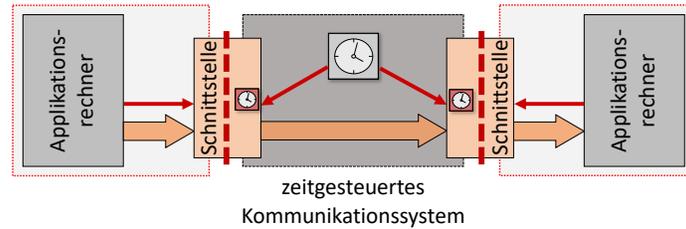


Abbildung 1: Systemmodell eines zeitgesteuerten Kommunikationssystems.

das Schreiben und Lesen von Nachrichten, an den Übertragungsplan des Kommunikationssystems anzupassen. Mit dieser Strategie können z.B. kurze Antwortzeiten von Transaktionen, an denen mehrere Rechner beteiligt sind, erreicht werden.

3 Zugriffsstrategien für zeitgesteuerte Netzwerkschnittstellen

Ein Knoten kann entsprechend der Erfordernisse einer Anwendung auf die Schnittstelle des zeitgesteuerten Kommunikationssystems auf eine von zwei verschiedenen Arten zugreifen: Der Zugriff kann zeitunabhängig, d.h. asynchron zur Arbeit des Kommunikationssystems, oder zeitsynchron mit der Nachrichtenübertragung durch das Kommunikationssystem erfolgen [11].

3.1 Asynchroner Schnittstellenzugriff

In zahlreichen Echtzeitsystemen werden die Tasks, die auf den Applikationsrechnern laufen, ereignisgesteuert, d.h. durch die Beobachtung von relevanten Ereignissen in der Umgebung aktiviert. Das Scheduling solcher Systeme erfolgt zur Laufzeit mittels dynamischer Schedulingverfahren und führt zum asynchronen Zugriff auf die Schnittstelle des Kommunikationssystems.

Das Kommunikationssystem stellt ereignisgesteuerten Systemen eine Shared-Memory Schnittstelle zur Verfügung. Es überträgt zu sendende Daten periodisch bzw. aktualisiert Daten, die von einem Knoten gelesen werden, in regelmäßigen Zeitabständen, wobei empfangene Daten jeweils eine bekannte, minimale Gültigkeitsdauer haben. Durch die vom Netzwerk definierte Aktualisierungsrate der Schnittstellendaten bildet die Schnittstelle eines zeitgesteuerten Kommunikationssystems eine Kontrollflussbarriere zwischen den Knoten des verteilten Systems. Das bedeutet, dass Knoten, die Nachrichten empfangen, nie einer unbekanntenen Last durch das Kommunikationssystem ausgesetzt sind. Damit kann es nicht zur Überlastung eines Knotens durch das Sendeverhalten anderer Knoten (Hochlast) kommen.

Das asynchrone Schreiben und Lesen der zeitgesteuerten Kommunikationsschnittstelle gibt den Knoten eines Echtzeitsystems einerseits ein hohes Maß an

Autonomie bezüglich der Ablaufsteuerung, was die Programmierung der Knoten vereinfacht. Andererseits erschwert asynchroner Schnittstellenzugriff aber die Realisierung replikadeterministischer Systeme: es kann nicht sichergestellt werden, dass redundante Knoten die gleichen Wertefolgen von der Netzwerkschnittstelle lesen. Weiters erfordert der asynchrone Schnittstellenzugriff spezielle Protokolle, um zu vermeiden, dass es aufgrund des wechselseitigen Ausschlusses zwischen Knoten und Kommunikationssystem zum Blockieren und somit zu zeitlichen Seiteneffekten beim Zugriff auf die Nachrichtendaten kommt [1, 11].

3.2 Synchroner Schnittstellenzugriff

Beim zeitsynchronisierten Zugriff werden die Zeitpunkte von Lese- und Schreiboperationen der Knoten auf die Schnittstelle mit der Nachrichtenübertragung durch das Kommunikationssystem synchronisiert. Damit werden einerseits Zugriffskonflikte auf die gemeinsamen Daten vermieden, andererseits kann die Nachrichtenlatenz zwischen Sender und Empfänger auf diese Weise kurz gehalten werden. Eine solche Synchronisation von Sende- und Empfangstasks stellt sicher, dass Nachrichtensender ihre Daten unmittelbar vor dem Transport durch das Netzwerk an die Schnittstelle übergeben, bzw. umgekehrt Empfängertasks Daten unmittelbar nach dem Nachrichtenempfang am Knoten aus der Schnittstelle auslesen. Ein zeitgesteuertes Kommunikationssystem kann die Synchronisation der Knoten mit Hilfe eines Uhrensynchronisationsinterrupts unterstützen [11].

Der synchrone Schnittstellenzugriff bringt eine Reihe von Vorteilen: Erstens wird durch die Synchronisierung der Schreib- und Lesevorgänge auf allen Knoten mit der globalen Zeit der Jitter der Nachrichtenverzögerung klein gehalten. Bei Verwendung eines synchronisierten, tabellengesteuerten Schedulers auf den Knoten kann dieser Jitter im Bereich der Synchronisationsgenauigkeit mit der globalen Uhr gehalten werden.

Mit tabellengesteuerten Schedules, die synchron zur Nachrichtenübertragung interpretiert werden, kann die Abarbeitung von Tasks so an die Nachrichtenkommunikation angepasst werden, dass die Antwortzeiten von Echtzeittransaktionen, die sich über mehrere Knoten erstrecken, kurz gehalten werden können.

Wenn sowohl die Datenverarbeitung auf den Knoten, als auch die Datenübertragungen durch das Kommunikationssystem einem globalen Zeitplan folgen, dann ist die Information über die zu übertragenden Datenelemente und deren Alter implizit in allen Knoten des verteilten Echtzeitsystems verfügbar. Diese Informationen müssen daher nicht in den Nachrichten gespeichert und transportiert werden.

Schlussendlich hilft ein synchronisierter und geplanter Schnittstellenzugriff, replikadeterministische Systeme zu bauen, was für die Konstruktion von fehlertoleranten Systemen entscheidend ist. Da ein zeitsynchronisierter Zugriff eine vorgegebene Reihenfolge und ein vorgegebenes Timing aller Lese- und Schreiboperationen auf die Kommunikationsschnittstelle erzwingt, können Schnittstellenzugriffe durch replizierte Komponenten so geplant werden, dass sie stets auf denselben Nachrichteninstanzen und Eingaben arbeiten.

3.3 Vergleich synchroner versus asynchroner Schnittstellenzugriff

Tabelle 1 vergleicht die Eigenschaften der beiden Zugriffsstrategien auf eine zeitgesteuerte Kommunikationsschnittstelle. Der asynchrone Zugriff hat den Vorteil, dass die Applikationsrechner des Systems vollkommen autonom agieren können. Die Komplexität beim Entwickeln der Anwendungssoftware kann niedrig gehalten werden, da das lokale Task-Scheduling nicht durch die Übertragungszeitpunkte der Nachrichten eingeschränkt wird. Auf der anderen Seite erlaubt es der synchrone Schnittstellenzugriff, die Aktivierungszeiten der Tasks so an die Übertragungszeitpunkte der Nachrichten anzupassen, dass kurze Antwortzeiten von Echtzeittransaktionen erzielt werden können. Die Synchronisation von Taskausführung und Nachrichtenübertragung führt zu einer großen Vereinfachung bei der Konstruktion replikadeterministischer, fehlertoleranter Echtzeitsysteme.

<i>Eigenschaft</i>	<i>Zugriff auf Netzwerkschnittstellen</i>	
	<i>asynchron</i>	<i>synchron</i>
Kontrolle im Rechner	vollständige Autonomie	Anpassung an Netzwerk
Schwankung Datenalter	2× Nachrichtenperiode	Präzision Netzwerkuhren
Phasenabstimmung Tasks	nein	ja
Kodierung von Zeit	Wert (explizit)	Kontrollsignal (implizit)
Replikadeterminismus	nein	ja
Konstruktionsaufwand	niedrig	mittel bis hoch

Tabelle 1: Vergleich asynchroner versus synchroner Zugriff auf eine zeitgesteuerte Netzwerkschnittstelle.

4 Fallstudien von Echtzeitanwendungen

Im folgenden beschreiben wir unterschiedliche Anwendungsdomänen von Echtzeitsystemen, um die Variabilität von Echtzeitanforderungen zu veranschaulichen.

4.1 Flugzeug-Autopilot

Ein *Flugzeug-Autopilot* ist eine Echtzeitanwendung, mit welcher die Flugbahn eines Flugzeuges automatisch gesteuert wird. Ein Flugzeug-Autopilot muss kurze Antwortzeiten haben, da Flugzeuge mit sehr hoher Geschwindigkeit fliegen. Zugleich hat ein Flugzeug-Autopilot eine hohe Zeitanforderung, da Variationen in der Antwortzeit sehr gering sein müssen. Ein Flugzeug-Autopilot benötigt eine hohe Ausfallsicherheit, da während des Fluges kein sicherer Fehlerzustand möglich ist. Allerdings kann im Falle eines Ausfalles je nach Flugzeugmodell der Pilot eingreifen und mit manueller Kontrolle fliegen. Bei Flugzeugen mit hoher

Manövrierfähigkeit ist allerdings eine Computer-Steuerung notwendig, da das Flugverhalten inhärent instabil ist.

4.2 Elektronische Lenkung

Das Charakteristikum einer *elektronischen Lenkung* ist die Ersetzung der mechanischen/hydraulischen Verbindung zwischen Lenkrad und Rädern durch computer-gesteuerte Aktuatoren. Elektronische Lenkung ist eine Anwendung mit hoher Sicherheitsanforderung. Zugleich hat eine elektronische Lenkung sehr strikte Echtzeitanforderungen wie etwa kurze Antwortzeiten und hohe Zeitanforderungen. Elektronische Lenkung benötigt auch eine hohe Ausfallsicherheit, da es Fehler auch bei hoher Fahrgeschwindigkeit tolerieren können muss. Derzeit ist die Anwendung einer elektronischen Lenkung allerdings in den meisten Gerichtsbarkeiten noch nicht zugelassen.

4.3 Fernchirurgie

Fernchirurgie ist eine Form der Telepräsenz mit sehr strikten Echtzeitanforderungen. Die Bedienelemente für den Chirurgen müssen mit dem Operationsroboter mittels Hochgeschwindigkeitsnetzwerken verbunden sein. Der Chirurg bekommt zur Kontrolle des Operationsvorganges visuelles Feedback mittels einer Videoübertragung mit typischerweise 24 Bildern pro Sekunde. Diese Videoübertragung muss durchgehend gefiltert werden zur Verbesserung der Bildqualität und damit zur Verringerung der kognitiven Belastung des Chirurgen. Fernchirurgie benötigt auch eine hohe Ausfallsicherheit, da auch bei technischen Fehlern die Sicherheit des Patienten garantiert werden muss. Diese Ausfallsicherheit kann durch geeignete Redundanzen erreicht werden. Alternativ könnte auch ein vor Ort anwesender Chirurg die Operation im Fehlerfall übernehmen.

4.4 Kabinendruckkontrolle

Kabinendruckkontrolle zur Regelung des Luftdruckes innerhalb des Flugzeuges ist in modernen Flugzeugen notwendig, um den Fluginsassen einen komfortablen Luftdruck in unterschiedlichen Flughöhen bieten zu können. Bei einer typischen Reiseflughöhe von 11km wird beispielsweise klimatisierte Luft in das Flugzeug gepumpt, um den Luftdruck einer Flughöhe von etwa 2.4km zu simulieren. Die Kabinendruckkontrolle ist eine sicherheitskritische Anwendung, da deren Ausfall in der Reiseflughöhe zum Tod der Insassen führen kann. Die Echtzeitanforderungen der Kabinendruckkontrolle kommen daher, dass man den Luftdruck möglichst gleichmäßig regeln möchte, um den Übergang zur virtuellen Höhe für die Flugzeuginsassen ohne Unannehmlichkeiten zu erreichen. Ein Flugzeug braucht nach dem Start etwa 15-20 Minuten bis die Reiseflughöhe erreicht wird, womit eine Anpassung des Luftdruckes alle paar Sekunden ausreicht. Die Zeitanforderung der Kabinendruckkontrolle ist daher relativ gering, da geringe Schwankungen im Zeitverhalten der Luftdruckanpassung keine ernsthaften

Konsequenzen haben. Die Kabinendruckkontrolle benötigt eine hohe Ausfallsicherheit im Falle von Fehlern. Sollte die Kabinendruckkontrolle komplett ausfallen, dann muss als letzte Rettung auf die Verwendung von Sauerstoffmasken zurückgegriffen werden.

4.5 Strukturelle Integritätsüberwachung

Unter *Struktureller Integritätsüberwachung* (SIÜ) versteht man die langzeitige Überwachung der strukturellen Integrität einer Brücke oder anderer Bauwerke. SIÜ benutzt in die Bauwerksstruktur eingebettete Sensoren um Informationen über etwaige strukturelle Veränderungen wie Belastung und Dehnung zu bekommen. Da SIÜ für Brücken und Flugzeuge angewendet wird, ist es definitiv eine sicherheitskritische Anwendung. Wenn auf Brücken oder Gebäude angewendet, sind die notwendigen Antwortzeiten relativ lang, da diese Überwachung über Jahrzehnte durchgeführt wird, und Veränderungen im Erdreich generell relativ langsam erfolgen. Für SIÜ von Brücken oder Gebäuden reichen relativ lange Antwortzeiten aus. Trotzdem müssen die zeitlichen Schwankungen der Antwortzeiten relativ gering sein, damit man vorhersagbare Updates von allen Sensoren bekommt. Während die strukturelle Integrität von Brücken und Gebäuden eine sicherheitskritische Eigenschaft ist, sind diese bei einem Ausfall von SIÜ immer noch kurzfristig benutzbar. Es wird daher bei SIÜ von Brücken und Gebäuden keine hohe Ausfallsicherheit benötigt, da eine kurzfristige Reparaturmöglichkeit ausreichend ist.

4.6 Motoreinspritzung

Die *Motoreinspritzung* steuert die Treibstoffzufuhr für interne Verbrennungsmaschinen. Eine Motoreinspritzung erfordert relativ kurze Antwortzeiten mit einer sehr geringen Zeitvariabilität. Der Motor eines Sportwagens kann beispielsweise 15,000 Umdrehungen pro Minute laufen, was etwa 4ms pro Umdrehung entspricht. Um eine präzise Treibstoffeinspritzung zu erreichen sind eine Einspritzung mit absoluter Genauigkeit und geringer Variabilität im Zeitverhalten notwendig. Motoreinspritzung hat eine mittlere Sicherheitsanforderung, da falsche Einspritzzeitpunkte den Motor beschädigen können, aber im Allgemeinen keine unmittelbare Gefährdung für menschliches Leben darstellen. Eine Motoreinspritzung benötigt keine Ausfallsicherheit.

4.7 Stahlwalzwerk, Nylondehnung

Ein *Stahlwalzwerk* ist eine Fabrik für die Herstellung von Stahl, und *Nylondehnung* dehnt Materialien wie Nylon zu einer dünnen Folie. Beide Anwendungen haben mittlere Sicherheitsanforderung, da sie keine unmittelbare Gefahr für menschliches Leben darstellen, aber hohe Folgekosten anfallen können, wenn ein Fehler auftritt, beispielsweise wenn ein Werkstück blockiert oder sich verfängt

aufgrund der Verletzung von Zeitanforderungen. Die Echtzeitanforderungen bei den Anwendungen ist definiert durch die Dynamik der zu transportierenden Materialien, typischerweise kurze Antwortzeiten mit mittlerer Zeitanforderung. Beide Anwendungen benötigen keine Ausfallsicherheit.

4.8 Multimedia-Unterhaltung: Videoübertragung

Videoübertragung kann in vielen sicherheitskritischen Anwendungen zum Einsatz kommen, wie etwa in der Fernchirurgie, welche in Abschnitt 4.3 beschrieben ist. Im Gegensatz dazu wollen wir in diesem Abschnitt Videoübertragung als ein Anwendungsbeispiel für Multimedia-Unterhaltung mit inkludierter Tonübertragung diskutieren, wo die Sicherheitsanforderung gering ist. Nichtsdestotrotz, um eine akzeptable Audio- und Bildqualität bei der Videoübertragung zu erhalten, können strikte Echtzeitanforderungen notwendig sein. Falls es sich dabei um eine Zweiwege-Kommunikation handelt, dann müssen definitiv die Antwortzeiten gering sein. Im Falle der Einweg-Kommunikation sind die Antwortzeiten und damit einhergehend die Verzögerungen nicht so kritisch. Trotzdem ist die Zeitanforderungen mindestens auf mittlerer Stufe, da Schwankungen in der Datenübertragung zu visuellen Artefakten und sehr störenden akustischen Störgeräuschen führen kann. Filmübertragung für Multimedia-Unterhaltung im Allgemeinen benötigt keine hohe Ausfallsicherheit. Trotzdem kann es Fälle geben, wo eine hohe Ausfallsicherheit aufgrund der involvierten Kosten sehr wohl notwendig ist, wie etwa im Falle von Massenkommunikation über Satelliten.

4.9 Diskussion

Im Folgenden untersuchen wir für die in Abschnitt 4 vorgestellten Echtzeitanwendungen, inwieweit für diese zeitgesteuerte Kommunikationssysteme empfehlenswert sind, beziehungsweise welche Art von zeitgesteuerten Netzwerkschnittstellen passend ist.

Anwendung	SA:	AZ:	ZA:	AS:	TT:	SST:
	Sicherheitsanforderung	Antwortzeiten	Zeitanforderung	Ausfallsicherheit benötigt	zeitgesteuerte Kommunikation	synchrone Schnittstelle
Flugzeug-Autopilot	hoch	kurz	hoch	ja	ja	ja
Elektronische Lenkung	hoch	kurz	hoch	ja	ja	ja
Fernchirurgie	hoch	kurz	hoch	ja	ja	ja
Kabinendruckkontrolle	hoch	mittel	mittel	ja	ja	nein
Strukturelle Integritätsüberwachung	hoch	lang	hoch	nein	eventuell	nein
Motoreinspritzung	mittel	kurz	hoch	nein	ja	ja
Stahlwalzwerk / Nylondehnung	mittel	kurz	hoch	nein	ja	ja
Multimedia-Unterhaltung	niedrig	kurz	mittel	nein	nein	n.a.

Tabelle 2: Eigenschaften von Echtzeit-Anwendungen

Tabelle 2 zeigt für die unterschiedlichen Echtzeitanwendungen Anforderungen zu Zuverlässigkeit und Echtzeitverhalten. Die Anwendungen in dieser Tabelle sind primär nach deren Sicherheitsanforderung geordnet, sowie sekundär nach deren Antwortzeiten. Die *Antwortzeit* (AZ) charakterisiert die geforderte Geschwindigkeit des Systems, während die *Zeitanforderung* (ZA) die erlaubten Schwankungen des Zeitverhaltens relativ zu der Größenordnung der Antwortzeiten beschreiben. Von der Perspektive der Systemzuverlässigkeit zeigen wir die *Sicherheitsanforderung* (SA) welche die Konsequenzen eines Systemausfalles klassifizieren, sowie die *Ausfallsicherheit* (AS) welche beschreibt ob das System eine sehr geringe Wahrscheinlichkeit eines Systemausfalles benötigt.

Die Spalte *zeitgesteuerte Kommunikation* (TT, von engl. Time-Triggered) in Tabelle 2 beschreibt, ob für die konkrete Applikation ein zeitgesteuertes Kommunikationssystem empfehlenswert ist.

Die Spalte *synchrone Schnittstelle* (STT) beschreibt, ob im Falle eines zeitgesteuerten Kommunikationssystems für die konkrete Anwendung eine synchrone Kommunikationsschnittstelle empfehlenswert ist.

Beispielsweise im Falle des *Flugzeug-Autopiloten* hat das System eine hohe Anforderung sowohl im Echtzeitbereich sowie auch in der Zuverlässigkeit, womit diese Anwendung ein natürlicher Kandidat für ein zeitgesteuertes Kommunikationssystem ist, am besten mit einem synchronen Kommunikationsinterface.

Auf der anderen Seite des Spektrums hat man die *Multimedia-Unterhaltung*, welche relativ geringe Anforderungen im Echtzeitbereich sowie auch in der Zuverlässigkeit hat, womit es schwierig ist, den extra Kostenaufwand für ein zeitgesteuertes Kommunikationssystem zu rechtfertigen.

Die anderen Applikationen liegen zwischen den beiden Extremfällen. Zum Beispiel bei der *Strukturelle Integritätsüberwachung* (SIÜ) wird ein regelmäßiges Überwachen von winzigen Bewegungen durchgeführt, um strukturelle Probleme frühzeitig erkennen zu können. Hier hat man für eine Echtzeitanwendung ungewöhnlich hohe Antwortzeiten, aber trotzdem eine hohe Zeitanforderung relativ zu den hohen Antwortzeiten. SIÜ ist ein Grenzfall und könnte durchaus mit einem zeitgesteuerten Kommunikationssystem ausgestattet werden, da man hier hohe Sicherheitsanforderungen sowie auch eine statische Systemkonfiguration hat, da beispielsweise die Anzahl der Sensoren sich im Normalbetrieb nicht ändert. Der extra Aufwand für einen synchronen Schnittstellenzugriff ist bei SIÜ allerdings nicht zu rechtfertigen.

Zusammengefasst zeigt Tabelle 2 einige brauchbare Faktoren um zu entscheiden ob ein zeitgesteuertes Kommunikationssystem empfehlenswert ist, und wenn ja, welche Form von Schnittstellenzugriff am besten geeignet ist.

5 Zusammenfassung

Zeitgesteuerte Kommunikationsprotokolle haben eine etablierte Nische im Bereich von sicherheitskritischen Echtzeitanwendungen. In diesem Artikel haben wir grundlegende Eigenschaften von zeitgesteuerten Kommunikationsprotokol-

len diskutiert, mit speziellem Fokus auf die unterschiedliche Zugriffsstrategien, nämlich asynchroner und synchroner Schnittstellenzugriff.

Weiters haben wir in diesem Artikel eine Analyse unterschiedlicher Echtzeitanwendungen gemacht im Hinblick auf deren Echtzeit- und Sicherheitsanforderungen. Basierend auf dieser Analyse haben wir aufgezeigt, ob für diese Anwendungen ein zeitgesteuertes Kommunikationsprotokoll empfehlenswert ist, und wenn ja, in welcher Weise (synchron oder asynchron) auf die Schnittstelle zugegriffen werden sollte. Wir haben bewusst eine größere Bandbreite von Anwendungsanforderungen gewählt, damit diese Empfehlungen auch leichter auf andere Echtzeitanwendungen übertragbar sind.

Literatur

1. Chen, J., Burns, A.: Loop-free asynchronous data sharing in multiprocessor real-time systems based on timing properties. In: Proc. 6th Int'l Conference on Real-Time Computing Systems and Applications (RTCSA'99). pp. 236–246 (1999). <https://doi.org/10.1109/RTCSA.1999.811236>
2. FlexRay Consortium: FlexRay Communications System - Protocol Specification, Version 2.0, 2.0 edn. (Jun 2004)
3. Kirner, R., Puschner, P.: A quantitative analysis of interfaces to time-triggered communication buses. *IEEE/ACM Transactions on Networking* **29**(4), 1786–1797 (August 2021). <https://doi.org/10.1109/TNET.2021.3073460>, <https://doi.org/10.1109/TNET.2021.3073460>
4. Kopetz, H., Ademaj, A., Grillinger, P., Steinhammer, K.: The time-triggered ethernet (tte) design. In: Eighth IEEE Int'l. Symposium on Object-Oriented Real-Time Distributed Computing. pp. 22–33 (2005). <https://doi.org/10.1109/ISORC.2005.56>
5. Kopetz, H., Grünsteidl, G.: TTP — A Protocol for Fault-Tolerant Real-Time Systems. *IEEE Computer* pp. 14–23 (January 1994)
6. Kopetz, H.: The time-triggered model of computation. In: Proc. IEEE Real-Time Systems Symposium. pp. 168–177. IEEE (1998)
7. Kopetz, H.: Real-Time Systems - Design Principles for Distributed Embedded Applications. Springer, London, UK, 2nd edn. (2011). <https://doi.org/10.1007/978-1-4419-8237-7>, ISBN: 978-1-4419-8236-0
8. Kopetz, H., Nossal, R.: Temporal firewalls in large distributed real-time systems. In: 6th IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems (1997)
9. Kopetz, H., Reisinger, J.: The non-blocking write protocol NBW: A solution to a real-time synchronization problem. In: Proc. IEEE Real-Time Systems Symposium. pp. 131–137 (Dec 1993)
10. Puschner, P., Kirner, R.: From time-triggered to time-deterministic real-time systems. In: Proc. 5th IFIP Working Conference on Distributed and Parallel Embedded Systems. pp. 115–124. Braga, Portugal (Oct 2006)
11. Puschner, P., Kirner, R.: Asynchronous vs. synchronous interfacing to time-triggered communication systems. *Journal of Systems Architecture - Embedded Software Design* **103** (2020). <https://doi.org/10.1016/j.sysarc.2019.101690>