



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Referat

Ralf von der Reith

STDMA - Zeitmultiplexing

Ralf von der Reith

STDMA - Zeitmultiplexing

Referat eingereicht im Rahmen der Vorlesung Verteilte Systeme

im Studiengang Angewandte Informatik (AI)
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. C. Klauck

Abgegeben am 31.05.2017

Inhaltsverzeichnis

1	Aufgabenstellung	4
2	Systementwurf	5
2.1	Komponenten.....	5
2.1.1	Slotmanager	5
2.1.2	Timemanager	7
2.1.3	Receiver	8
2.1.4	Sender	9
2.1.5	Station	10
2.2	Gesamtsystem.....	10
3	Zeit-Synchronisation	13
4	Kollisionsvermeidung	14
5	Quellen	15

1 Aufgabenstellung

Angelehnt an die Übertragung von Daten per Funk mit nur einem Kanal soll eine Sende-/Empfangsstation zur Datenübertragung auf Basis von Zeitmultiplexing entworfen und implementiert werden. Statt Funk wird IP-Multicast mit vorgegebener IP und Ports genutzt.

Um die gleichberechtigte Teilnahme mehrerer Stationen an der Kommunikation zu gewährleisten, wird die Zeit in Frames fester Größe (hier eine Sekunde pro Frame) seit 1.1.1970 00:00 geteilt. Diese Frames wiederum sind in 25 Slots á 40ms unterteilt. Jede teilnehmende Station darf genau einen Slot pro Frame zur Datenübertragung nutzen.

Desweiteren gibt es zwei Typen von Stationen, A und B. Jede teilnehmende Station gehört einem Typen an. Die Stationen des Typs A sind zeitgebend. Alle Teilnehmer synchronisieren ihre lokale Zeit anhand der Zeitstempel dieser Stationen. Da auch die A-Stationen untereinander zeitliche Abweichungen haben können synchronisieren sich diese auch gegenseitig.

Die zu sendenden Nachrichten-Pakete müssen folgender Vorgabe entsprechen:

Byte 0 Stationsklasse ('A' oder 'B')

Byte 1-24 Nutzdaten

Byte 1- 10 Name der Sendestation

Byte 25 Nummer des Slots, in dem im nächsten Frame gesendet wird

Byte 26-33 Zeitpunkt des Sendens dieses Pakets: Millisekunden seit 1.1.1970 00:00

Ziel ist es, eine möglichst Kollisionsfreie (d.h. nur genau eine Nachricht pro Slot) Kommunikation zu gewährleisten.

2 Systementwurf

2.1 Komponenten

Die Station besteht aus mehreren Komponenten:

- Dem Empfänger: Dieser wartet auf Nachrichten, nimmt sie entgegen und prüft, ob es Kollisionen gegeben hat.
- Der Slot-Verwaltung: Sie bekommt sämtliche Informationen über getätigte Slot-Reservierungen und gibt auf Anfrage einen freien Slot zurück.
- Dem Leser: Er nimmt Input entgegen/liest die zu sendenden Daten aus einer Datei aus und bereitet daraus eine Nachricht ohne Zeit und Slotreservierung vor.
- Dem Schreiber: Dieser nimmt erhaltene Nachrichten entgegen und loggt sie.
- Dem Sender: Er beschafft sich die nächste zu sendende Nachricht vom Leser, ergänzt sie um eine Slotreservierung von der Slot-Verwaltung sowie um einen Zeitstempel von der Zeit-Verwaltung und versendet diese Daten in Form eines UDP-Pakets.
- Der Zeitverwaltung: Sie stellt Zeitstempel bereit und übernimmt die Uhrensynchronisation.
- Der Station: Die Station steuert den zeitlichen Ablauf. Nach Beginn eines Slots beschafft sie sich die Nachrichten vom Empfänger, delegiert die einzelnen Informationen an die entsprechenden Komponenten und leitet den Sendeprozess ein.

2.1.1 Slotmanager

Der Slotmanager sammelt Informationen über den Zustand der einzelnen Slots innerhalb eines Frames. Er nimmt Reservierungen von Slots entgegen und stellt auf Anfrage eine noch verfügbare Slotnummer zur Verfügung.

Datenstrukturen

Liste der freien Slots: Liste aller noch nicht reservierten Slots. Jeder Slot wird durch eine eindeutige Zahl entsprechend ihrer Reihenfolge im Frame repräsentiert.

Der Slotmanager hat folgende (internen) Schnittstellen:

get_slot

Format: {get_slot, PID}

PID: Sender der Anfrage.

Reserviert einen Slot für den nächsten Frame und gibt die Nummer des reservierten Slots zurück.

Format der Antwort: {slot, SlotNr}

SlotNr: Ganze Zahl, die den Slot eindeutig repräsentiert.

reserve

Format: {reserve, Slot}

Slot: Nummer des reservierten Slot.

reserviert einen Slot für den nächsten Frame.

new_frame

Format: new_frame

Signalisiert dem Slotmanager, dass ein neuer Frame begonnen hat und setzt die Slotliste zurück.

Information: Wird nur am Ende eines Frames aufgerufen.

2.1.2 Timemanager

Die Zeitverwaltung dient der Synchronisation und der Bereitstellung der lokalen Zeit. Ihr werden nach Erhalt einer Nachricht die enthaltenen Zeitstempel bereitgestellt, sodass nach einem Frame der Mittelwert über die Zeitstempel aller A-Stationen gebildet und dieser als neuer Offset weiterverwendet werden kann.

Datenstrukturen:

Offsets: Liste aller Offsets, die während eines Frames erhalten wurden. Die Offsets werden als ganze Zahlen in Millisekunden hinterlegt.

Der Timemanager hat folgende (internen) Schnittstellen:

get_time

Format: {get_time, PID}

PID: Sender der Anfrage.

Fragt die aktuelle, lokale Zeit an. Sendet einen entsprechenden Zeitstempel zurück

Format der Antwort: {time, LocalTime}

LocalTime: Zeitstempel in Millisekunden

remote_time

Format: {remote_time, RemoteTimeStamp, ReceiveTime}

RemoteTimeStamp: Zeitstempel des Senders in Millisekunden

ReceiveTime: Zeitstempel des Empfängers in Millisekunden

Berechnet die Differenz aus den beiden Zeitstempeln und fügt sie der Liste der erhaltenen Zeitstempel hinzu

sync_time

Format: {sync_time}

Berechnet den durchschnittlichen Offset aus allen bisher erhaltenen Offsets seit dem letzten Aufruf von sync_time. Die Lokale Zeit wird um diesen Durchschnitts-Offset angepasst. Anschließend wird die Liste der erhaltenen Offsets geleert.

Information: Dieser Aufruf findet immer zum Ende eines Frames statt.

2.1.3 Receiver

Der Receiver nimmt alle UDP-Pakete entgegen und extrahiert die Informationen aus diesen. Auf Anfrage stellt der Receiver eine erhaltene Nachricht bereit, bzw meldet eine Kollision, wenn seit der letzten Anfrage mehr als eine Nachricht erhalten wurde.

Datenstrukturen:

Messages: Liste aller erhaltenen Nachrichten inklusive Empfangs-Zeitstempel seit der letzten Anfrage (`get_message`)

Eine Nachricht hat folgendes Format:

{Station, Inhalt, Slot, Zeitstempel Sender, Zeitstempel Empfang}

Station: "A" oder "B"

Inhalt: Die eigentlichen Informationen der Nachricht

Slot: Der für den nächsten Frame reservierte Slot

Zeitstempel Sender: Zeitstempel des Senders in Millisekunden

Zeitstempel Empfang: Zeitstempel des Empfängers in Millisekunden

Der Receiver hat folgende (internen) Schnittstelle:

get_message

Format: {`get_message`, PID}

PID: Sender der Anfrage.

Der Receiver schaut, wie viele UDP-Nachrichten seit der letzten Anfrage von `get_message` erhalten wurden und gibt entsprechend folgender Bedingungen verschiedene Nachrichten zurück:

Es wurde seit der letzten Anfrage keine Nachricht erhalten:

Format der Antwort: {msg, none}

Es wurde seit der letzten Anfrage genau eine Nachricht erhalten:

Format der Antwort: {msg, Message}

Message: siehe Format einer Nachricht in Datenstruktur

Es wurden mehrere Nachrichten erhalten (entspricht einer Kollision):

Forma der Antwort: {collision}

2.1.4 Sender

Der Sender nimmt die zu sendenden Daten und eine Slotreservierung entgegen und sendet diese an die ihm zugewiesene IP-Adresse. Beim Senden wird ein aktueller Zeitstempel, sowie der Stationstyp angefügt.

Datenstruktur:

Eine Nachricht hat folgendes Format:

{Station, Inhalt, Slot, Zeitstempel Sender}

Station: "A" oder "B"

Inhalt: Die eigentlichen Informationen der Nachricht

Slot: Der für den nächsten Frame reservierte Slot

Zeitstempel Sender: Zeitstempel des Senders in Millisekunden

Der Sender hat folgende (interne) Schnittstelle:

send

Format: {send, PID, TimeSlotStart, TimeSlotEnd}

PID: Sender der Anfrage.

TimeSlotStart: lokale UTC-Zeit Slot-Beginn.

TimeSlotEnd: lokale UTC-Zeit Slot-Ende.

Der Sender holt sich den Inhalt, eine neue Reservierung und den aktuellen Zeitstempel und sendet die Nachricht, sofern der Zeitstempel im Zeitfenster TimeSlotStart – TimeSlotEnd liegt. Anschließend meldet der Sender den reservierten Slot oder das Verfehlen des Zeitfensters zurück.

Format der Antwort:

{ok, NewSlot}

NewSlot: reservierter Slot für den nächsten Frame.

{missed}

Zeitfenster verfehlt.

2.1.5 Station

Die Station startet alle anderen Komponenten. Anschließend übernimmt sie die Koordination, indem sie zu Beginn jedes Slots die empfangene Nachricht des letzten Slots einliest, die Daten an die anderen Komponenten zur Auswertung weitergibt und bei eigener Reservierung den Sender aktiviert. Sie meldet den Komponenten auch den Beginn eines neuen Frames und kümmert sich um den sauberen (wieder-)Einstieg in die Kommunikation.

2.2 Gesamtsystem

Bei Start der Station werden alle Komponenten initialisiert. Anschließend wird vom Timemanager die aktuelle Zeit abgefragt und ein Timer bis zum Beginn des nächsten Slots gestartet.

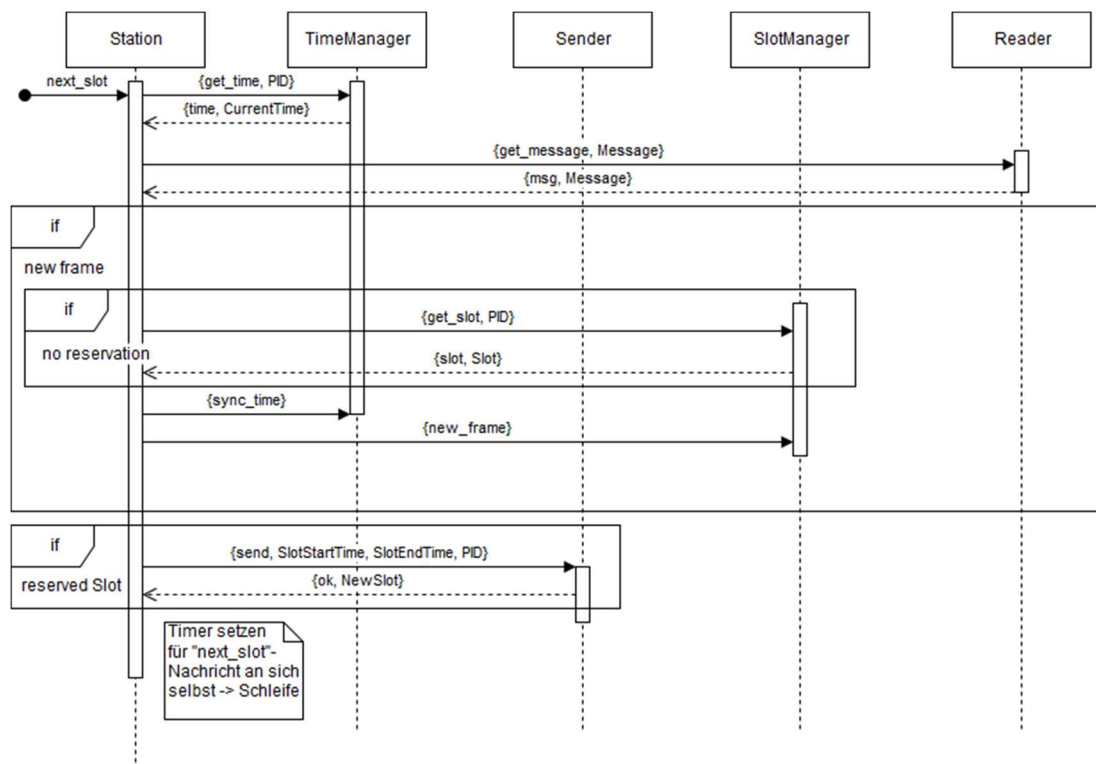


Bild 1: Ablauf während eines Slots

Bei Ablauf des Timers zum nächsten Slot wird zunächst die lokale Zeit abgefragt und anschließend überprüft, in welchem Slot sich das System derzeit befindet. Anschließend werden die Nachrichten des letzten Slots abgefragt.

Der Receiver gibt die erhaltene Nachricht zurück, sofern genau eine vorliegt oder meldet eine Kollision bzw. keine erhaltene Nachricht. Im Falle einer erkannten Kollision wird zusätzlich geprüft, ob die Kollision mit einer eigenen Nachricht stattgefunden hat. Falls ja, wird die eigene Reservierung aufgehoben, da sie aufgrund der Kollision für andere Kommunikations-Teilnehmer nicht wahrnehmbar ist.

Die Bestandteile der Nachricht werden nun an die einzelnen Komponenten weitergeleitet:

- Die enthaltene Reservierung wird an den Slotmanager übergeben.
- Der mitgegebene Zeitstempel wird an die Zeitverwaltung übergeben, sofern die Nachricht von einer Station des Typs A kommt.
- Die Daten werden an die Senke weitergeleitet, welche die Daten in diesem Fall lediglich in einer Datei loggt.

Nach der Verarbeitung der Nachricht wird geprüft, ob ein neuer Frame begonnen hat. Bei Beginn eines neuen Frames werden sowohl Zeitmanager als auch Slotmanager darüber informiert, sowie der Einstieg in die Kommunikation vorbereitet. Dazu wird ein während des letzten Slots nicht reservierter Slot ausgewählt.

Im letzten Schritt wird überprüft, ob der aktuelle Slot von der Station reserviert wurde. Wenn dies der Fall ist, werden die zu sendenden Daten vom Reader abgeholt, ein Slot für den nächsten Frame reserviert und diese Informationen an den Sender übergeben.

Zuletzt berechnet die Station anhand der erneut erfragten lokalen Zeit die Dauer bis zum Beginn des nächsten Frames und startet einen neuen Timer. Dieser Prozess wird kontinuierlich bis zur Terminierung der Station durch den Nutzer fortgesetzt.

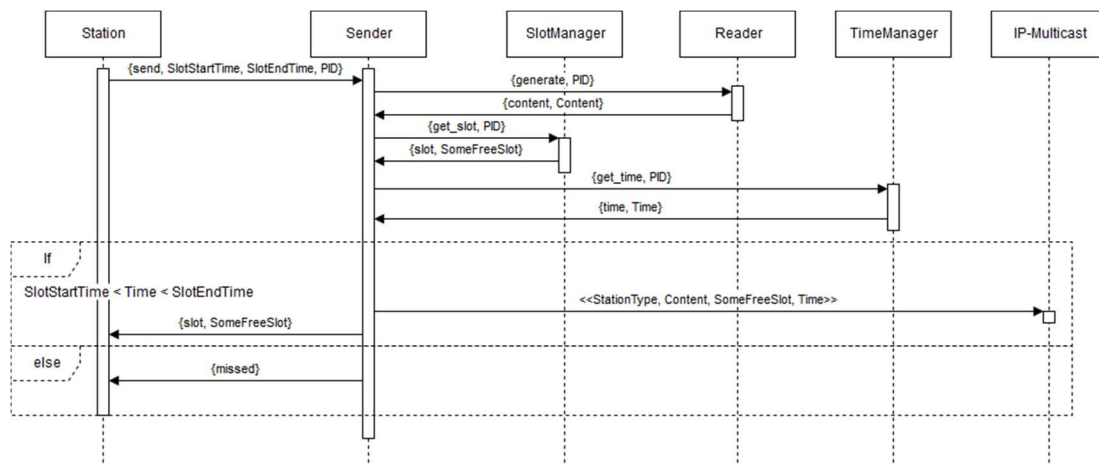


Bild 2: Ablauf beim Senden

Der Sender beschafft sich die zu sendenden Daten, eine Slotreservierung und den aktuellen Zeitstempel, bildet daraus das Datenpaket und sendet es anschließend an die ihm bei Initialisierung übergebene IP-Adresse und Port.

Die Synchronisation der lokalen Zeit geschieht immer zu Beginn eines neuen Frames. Dabei wird der Durchschnitt aller von Stationen des Typs A erhaltenen Zeitstempel gebildet und als Offset auf die lokale Zeit addiert.

3 Zeit-Synchronisation

Die Stationen besitzen keine physikalischen Zeitgeber. Dies führt dazu, dass die lokalen Uhren der Stationen z.T. ungenau sind und sie so über die Zeit asynchron laufen. Außerdem kann eine Station eine falsche lokale Zeit eingestellt haben, welche ebenfalls von der Zeit des Kommunikationskanals abweicht.

Abweichungen der lokalen Zeit führen dazu, dass Reservierungen den falschen Frames zugeordnet und Zeitfenster für das eigene Senden verpasst werden. Daraus resultierend kann es zu Kollisionen kommen, welche zusätzlich auch den Datentransfer aller anderen Teilnehmer einschränken.

Um dieses Problem zu lösen, ist es notwendig, die Zeit des lokalen Systems mit der Zeit der anderen Teilnehmer zu synchronisieren. Zunächst einmal gibt es Stationen deren lokale Zeitgeber hinreichend genau arbeiten (Typ A) und Stationen, deren Zeit nicht ausreichend vertraut wird (Typ B). Durch den Abgleich der lokalen Zeit mit dem empfangenen Absende-Zeitstempeln der A-Stationen können Differenzen in der lokalen Zeit berechnet und ausgeglichen werden.

Im folgenden gehe ich auf zwei mögliche Strategien beim Synchronisieren der Zeit mit den Zeitstempeln der A-Stationen ein.

Variante A:

Bei Empfang eines Zeitstempels wird direkt die Differenz zwischen dem erhaltenen Zeitstempel und der lokalen Zeit berechnet. Dieser Offset wird nun auf die lokale Zeit addiert, um so die Differenz auszugleichen.

Variante B:

Es werden alle relevanten Zeitstempel über einen bestimmten Zeitraum (z.b. innerhalb eines Frames) gesammelt. Über der Menge der erhaltenen Zeitstempel wird anschließend der Mittelwert gebildet und dieser als Offset auf die lokale Zeit addiert.

Variante A hat den Vorteil, dass es sehr schnell die lokale Zeit mit einer zeitgebenden Station abgleicht. Ein Problem dabei ist jedoch, dass auch die Stationen

des Typ A Abweichungen untereinander haben können. Dies führt dazu, dass die lokale Zeit immer wieder hin und her springt. Insbesondere wenn eine der A-Stationen wider erwarten größere Abweichungen hat, kann dies schnell zum Verfehlen von Slots beim Senden und somit zu Kollisionen führen.

Variante B arbeitet in dieser Hinsicht stabiler, da extreme Sprünge und das "Zittern" durch häufige Anpassungen dadurch reduziert werden, dass die lokale Zeit seltener synchronisiert wird. Hier ist der Nachteil, dass etwas länger auf die Synchronisation der lokalen Zeit mit dem System gewartet werden muss.

Ich habe mich für Variante B entschieden, da mir die stabilere Zeit wichtiger erschien, insbesondere da es in dieser Arbeit weniger um die schnelle, sondern um die Fehler-/Kollisionsfreie Übertragung der Daten geht.

4 Kollisionsvermeidung

Kollisionen bei der Datenübertragung sind in dem hier beschriebenen Anwendungsfall äußerst problematisch, da Kollisionen zu Verlust der Datenpakete führt und so die komplette Kommunikation des Systems gestört wird. Deshalb ist es wichtig, Daten möglichst kollisionsfrei zu übertragen.

Die Grundlage zur Vermeidung von Kollisionen besteht in der Aufteilung der Zeit in Frames und Slots und der Reservierung dieser. Bevor selbst gesendet wird, muss allerdings sichergestellt sein, dass auch die lokale Zeit im Einklang mit dem System ist, da sonst das Einhalten des Zeitfensters eines Slots nicht gewährleistet werden kann.

Bei dem Einstieg in die Kommunikation wird daher zuerst die lokale Zeit mit dem System abgeglichen und anschließend ein kompletter Frame lang auf sämtliche

Reservierungen gehorcht. Im ersten Schritt ist wird dadurch sichergestellt, dass die lokale Zeit hinreichend korrekt ist und im zweiten Schritt werden die Reservierungsinformationen gesammelt, die als Grundlage dienen, einen eigenen Slot für die Kommunikation auszuwählen.

Um beim gleichzeitigen Einstieg mehrerer Stationen Kollisionen zu vermeiden, wird der Slot für den Einstieg per Zufall ausgewählt, um dort Überschneidungen zu minimieren.

5 Quellen

<http://s3.amazonaws.com/publicationslist.org/data/xavier.gelabert/ref-2/Synchronisation%20Strategies%20in%20TDMA%20Tactical%20Radio%20Area%20Networks%20-%20MSc%20Thesis.pdf>

Ebene 1

6

Erklärung zur schriftlichen Ausarbeitung des Referates

Hiermit erkläre ich, dass ich diese schriftliche Ausarbeitung meines Referates selbstständig und ohne fremde Hilfe verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe sowie die aus fremden Quellen (dazu zählen auch Internetquellen) direkt oder indirekt übernommenen Gedanken oder Wortlaute als solche kenntlich gemacht habe. Zudem erkläre ich, dass der zugehörige Programmcode von mir selbstständig implementiert wurde ohne diesen oder Teile davon von Dritten im Wortlaut oder dem Sinn nach übernommen zu haben. Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher nicht veröffentlicht.

Hamburg, den 31.05.17

