



22. DEZEMBER 2014

ENTWURF

VERTEILTE SYSTEME PRAKTIKUM 4

STEFFEN GIERSCHE & MARIA LÜDEMANN
BAI 5 VERTEILTE SYSTEME BEI HERRN PROFESSOR KLAUCK
HAW Hamburg

Team: 2 Steffen Giersch & Maria Lüdemann

Aufgabenaufteilung:

Aufgabe	Teammitglied
Entwurfsplanung	Steffen Giersch
Entwurfsplanung	Maria Lüdemann
Implementation	Steffen Giersch
Implementation	Maria Lüdemann

Bearbeitungszeitraum:

Datum	Dauer	Teammitglied
15.12.2014	2 Stunden	Steffen & Maria
22.12.2014	5,5 Stunden	Steffen & Maria
21.11.2014	2 Stunden	Maria
25.11.2014	6,5 Stunden	Steffen & Maria
26.11.2014	7 Stunden	Steffen
27.11.2014	2 Stunden	Steffen & Maria
28.11.2014	1 Stunde	Maria
29.11.2014	4,5 Stunden	Steffen & Maria

Gesamt: 29 Stunden

Quellen: Beim Entwurf haben wir uns am Entwurf der Gruppe von Max Zender und Sven Freiberg orientiert.

Aktueller Stand:

- ❖ Entwurfsdokument ist in Arbeit
- ❖ Implementation noch nicht begonnen

1 INHALTSVERZEICHNIS

1	Inhaltsverzeichnis.....	2
2	Funktionalität.....	3
3	Komponenten.....	3
3.1	Station - station	3
3.2	Zeit Synchronisation time_Sync_Manager.....	5
3.3	Datenquelle – data_source	5
3.4	Datensenke – data_sink.....	5
4	Nachrichten.....	5

2 FUNKTIONALITÄT

Es soll ein System implementiert werden dass ein Zeitmultiplexverfahren(TDMA) simuliert. Also die Möglichkeiten bietet mit mehreren Sendestationen über nur eine einzige Leitung Nachrichten zu senden. Dabei haben wir uns entschieden das System in Erlang zu implementieren da Erlang eine sehr gute Grundlage für Zeitkritische Systeme bietet.

3 KOMPONENTEN

3.1 STATION – STATION

Die Station ist das eigentliche Herzstück des Systems, es beinhaltet alle Relevanten Komponenten für die jeweiligen Sende Stationen und kümmert sich um das Senden und Empfangen von Nachrichten, das Nachrichten zusammen bauen und die Stationsinterne Zeit. Wobei es zwei Arten von Stationen geben soll. Die Stationen des Typs A mit hinreichend genauer Zeit die ihre Zeiten mit denen der anderen Stationen vom Typ A synchronisieren und denen vom Typ B die eine nicht hinreichend genaue Zeit haben. Diese Stationen synchronisieren ihre Zeit mit denen der Stationen vom Typ A. Diese Komponente kümmert sich auch um die Konfiguration und das Starten der beinhalteten Module

Beinhaltet:

- Sender
- Reciever
- Slot_manager

3.1.1 Sender

Der Sender kümmert sich um das Versenden der Nachrichten. Er kümmert sich nicht darum wann er senden muss, also wann sein Slot innerhalb des Frames ist sondern wird vom Stationseigenen slot_manager angewiesen wenn sein Sendeslot kurz bevor steht. Er prüft allerdings, ob er bei Benachrichtigung vom slot_manager noch in der richtigen Zeit liegt und noch senden darf. Es soll so implementiert werden, dass der Sender überprüft ob er seinen Sendeslot verpasst hat, dies kann durch ungenaue Zeiten passieren, wenn er es verpasst hat dann sendet er in diesem Slot nicht.

3.1.2 Reciever

Der Reciever erhält und verwertet die Nachrichten die über das System eingehen. Dabei stellt er auch Kollisionen fest und entscheidet wie damit zu verfahren ist. Dies macht er indem er für jeden

Slot die Nachrichten sammelt und am Ende eines Slots überprüft ob mehr als eine Nachricht empfangen worden ist. Ist tatsächlich mehr als eine Nachricht eingegangen, wird davon ausgegangen dass es sich somit um eine Kollision handelt. Wenn nicht, liegt auch keine Kollision vor und es kann ganz normal fortgefahren werden.

Verhalten bei Kollision:

Wenn mehr als eine fremde Nachricht entdeckt wird (nicht aus der eigenen Station), werden sie verworfen.

Wenn mehrere Nachrichten entdeckt werden von denen mindestens eine, eine eigene ist, werden alle Nachrichten verworfen und eine Nachricht gesendet um die geplante Slot Reservierung vom Slot_Manager zurück setzen zu lassen. Dies wird getan weil mit der eigenen Nachricht ein neuer Slot hätte reserviert werden sollen, dies durch die Kollision allerdings nichtig wird. So muss beim nächsten Senden ein zufälliger Slot genutzt werden.

Verhalten ohne Kollision:

Der Slot_Manager bekommt eine Nachricht durch die eine Reservierung gesetzt wird. Die Zeit aus der empfangenen Nachricht wird an die Time_Master Komponente weiter gegeben sofern sie aus einer Station vom Typ A kommt und die Data_Sink bekommt den Payload der Nachricht übermittelt.

3.1.3 Slot_manager

Der Slot_Manager kümmert sich um die Definition, Handhabung und Vergabe der Slots innerhalb des Frames. Er ist die einzige Komponente mit einem Überblick über die einzelnen Slots und kümmert sich darum, wann ein Slot beginnt und endet sowie wann ein Frame beginnt und endet, um die Reservierung der Slots und welche Slots noch frei sind. Dementsprechend hält er sich eine Liste der freien Slots.

Der Slot_Manager setzt immer wieder von neuem einen Slot_Timer, immer wenn dieser abgelaufen ist wird der Time_Master nach der aktuellen Zeit gefragt und der Reciever darüber benachrichtigt, dass ein Slot beendet wurde und setzt dann den Timer neu.

Immer wenn ein Frame vorbei ist wird der Time_Master darüber benachrichtigt, danach erfragt er vom Time_Master wie nach jedem Slot die aktuelle Zeit.

Für den Fall dass ein Slot reserviert wurde wird dem Sender dieser Slot übergeben. Wurde kein Slot reserviert wird ein beliebiger freier Slot aus dem letzten Frame genommen, dann aber im nächsten Frame, übergeben dabei wird darauf geachtet, dass wirklich ein zufälliger Slot gewählt wird und nicht etwa ein fest gescripteter z.B der erste freie genommen wird. Um zu vermeiden dass man gescriptete Kollisionen erzeugt.

Danach wird die Liste der freien Slots zurückgesetzt.

3.2 ZEIT SYNCHRONISATION TIME_TIME_MASTER

Der Time_Manager wird immer von Komponenten nach der aktuellen Zeit gefragt. Und gibt diese auf Anfrage zurück.

Er erhält außerdem vom Reciever die Zeiten aus den Nachrichten der Stationen vom Typ A.

Die dritte Nachricht die der Time_Master erhält ist die Information darüber, dass das Frame abgelaufen ist. Beim erhalten dieser Nachricht, wird das arithmetische Mittel aus den erhaltenen Zeiten ermittelt und mit der Zeit verrechnet.

3.3 DATENQUELLE – DATA_SOURCE

Die Datenquelle liest durchgehend auf dem InputStream die 24 Bytes Nutzdaten ein und versendet die jeweils aktuellsten 24 Bytes auf Anfrage.

3.4 DATENSENKE – DATA_SINK

Die Datensenke nimmt die 24 Bytes Nutzdaten vom Reciever entgegen und schreibt diese in eine Log-Datei weg.

4 NACHRICHTEN

Hier sollen einmal alle Nachrichten aufgezählt und definiert werden die eine Komponente erhalten kann.

4.1 STATION

4.1.1 Sender

Nachricht: {new_timer, TimeToWait}

Absender: Slot_Manager

Parameter: TimeToWait -> Zeit die bis zum Slot gewartet werden muss

Nachricht: {pay_load, Data}

Absender: Data_source
Parameter: data -> 24 Bytes Nutzdaten

4.1.2 Reciever

Nachricht: {slot_passed}
Absender: Slot_Manager
Beschreibung: Information darüber wenn der Slot abgelaufen ist
Antwort: {no_messages}
Beschreibung: Keine Nachrichten erhalten
Antwort: {slot_reservation, Slot}
Parameter: Slot -> Slot der reserviert wurde
Beschreibung: Slot erfolgreich reserviert
Antwort: {collision_detectet}
Beschreibung: Eine Kollision wurde entdeckt, Reservierung verworfen

4.1.3 Slot_Manager

Nachricht: {get_reservable_slot}
Absender: Sender
Beschreibung: Ist die Frage nach einem reservierbarem Slot
Antwort: {reservable_slot, Slot}
Parameter: Slot -> Slot der reserviert wurde

4.2 TIME_MASTER

Nachricht: {new_message, StationTime, RecievedTime }
Parameter: StationTime -> Zeit der Station vom Typ A
 RecievedTime -> Zeitpunkt an dem die Nachricht eingegangen ist
Absender: Reciever

Nachricht: {get_current_time, PID}

Parameter: PID -> Prozess ID der abfragenden Komponente

Absender: Reciever, Slot_Manager, Sender

Antwort: {current_time, Time}

Parameter: Time -> aktuelle Zeit

Nachricht: {sync}

Absender: Slot_Manager

Beschreibung: Triggert, dass der Time_Master sich selbst, anhand den erhaltenen Zeiten aus dem letzten Frame mit den anderen Stationen synchronisiert

4.3 DATA_SINK

Nachricht: {data, Data}

Parameter: Data -> 24 Bytes Nutzdaten die weggeschrieben werde

Absender: Reciever