

Verteilte Systeme Aufgabe 4 Konzept

Marian Triebe, Moritz Heindorf
{marian.triebe, moritz.heindorf}@haw-hamburg.de

1 Analyse der Aufgabenstellung

Es soll ein Programm zur Zeitsynchronisation erstellt werden, dazu wird ein Zeitschlitzverfahren verwendet welche oft in echtzeitfähigen Systemen anzufinden sind. Zeitschlitz werden hierbei statisch bei Programmstart zugewiesen. Die Kommunikation zwischen den einzelnen Rechnern/Instanzen geschieht über UDP (Multicast). Es existiert eine Master Instanz welche ein so genanntes *Beacon* sendet, darauf hin beginnt eine neue *Superframe*. In einem *Superframe* senden alle Instanzen in Ihrem zugeteilten Zeitschlitz eine *Slot-Nachricht*, das versenden soll genau in der Mitte des zugeteilten Zeitschlitzes geschehen. Die Anwendung soll später mit Lösungen von anderen Teams arbeiten können, deshalb sind die definierten Datenstrukturen aus der Aufgabenstellung zu verwenden.

2 Aufbau der Programme

Die Anwendung wird in C Programmiert werden und nur Support für Linux besitzen, da zum sicherstellen der korrekten Timings der *HRTimer* (*High Resolution Timer*) des Linux Kernels eingesetzt werden soll. UDP ist zwingend notwendig um Nachrichten an alle Instanzen per Multicast ausliefern zu können. Der *HRTimer* löst das Signal *SIGALRM* aus, woraufhin eine neue Nachricht versendet wird. Beim Empfangen einer neuen Nachricht wird vom Socket *SIGIO* ausgelöst. Es müssen Signalhandler für diese Signale implementiert werden.

2.1 Superframe

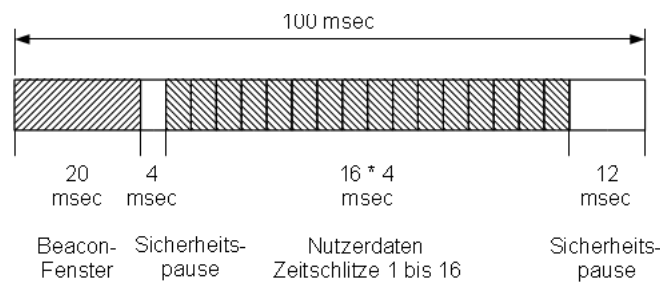


Abbildung 1

Das Format der Superframe ist definiert und Abbildung 1 zu entnehmen. Die Superframe ist 100msec lang und setzt sich aus dem *Beacon-Fenster* (20msec) gefolgt von einer Sicherheitspause von 4msec, 16*4msec Zeitschlitz und einer 12msec Sicherheitspause zusammen.

2.2 Beacon

```
#define HOST_LEN 32
struct beacon {
    uint32_t frameNr;
    uint64_t timeShift;
    char host[HOST_LEN];
};
```

Listing 1

In Listing 1 ist die interne definition einer *Beacon-Nachricht* dargestellt. Diese Nachricht soll dann als ASCII formatierte Nachricht in das Netzwerk gesendet werden. Der Aufbau dieses ASCII Strings ist der Aufgabenstellung entnommen und ist in Listing 2 definiert.

```
/*
    B<frameNr>:<timeShift>:<hostname>

    frameNr:    Fortlaufende Nummerierung des Frames.
    timeShift:  Tatsaechlicher Sendezeitpunkt des Beacons in
                nsec, relativ zum Startzeitpunkt des Frames.
    hostname:   ASCII-String, maximale Laenge: 32 Zeichen

    Beispiel:
        B218:2888685:Lab22
*/
```

Listing 2

2.3 Slot-Nachricht

Die *Slot-Nachricht* soll von der jeweiligen Instanz in der Mitte des zugeteilten Zeitschlitzes versendet werden. Die Nachricht soll als ASCII String versendet werden, die Zusammensetzung ist in Listing 3 definiert.

```
/*  
D<slotnummer>:<hostname>  
  
    slotnummer: Zahl 1 bis 16  
    hostname:   ASCII-String, maximale Laenge: 32 Zeichen  
  
    Beispiel:  
    D9:Lab29  
*/
```

Listing 3

Quelle: <http://tiserver02.cpt.haw-hamburg.de/htm/vs.php?page=aufgabe4>

3 Mögliche Probleme

- Nicht korrekte Verwendung der Datenstrukturen einzelner Teams
- Senden zu falschen Zeiten (nicht im eigenen Slot)
- Delay durch Signallaufzeiten (Im Labor irrelevant)
- Ausbleiben des Beacons

4 Grober Ablauf eines Beacons

Das hier dargestellte Sequenzdiagramm hat einige Fehler, soll jedoch das versenden der Beacon Nachricht visualisieren, sowie das versenden eines Frames innerhalb des Zeitschlitzes.

