Prof. Dr. Thomas Schmidt
HAW Hamburg, Dept. Informatik,
Raum 480a, Tel.: 42875 - 8452
Email: t.schmidt@haw-hamburg.de
Web: http://inet.haw-hamburg.de/teaching

Verteilte Systeme

Aufgabe 3: Gruppenkoordination im Slotted TDMA

Ziele:

1. Koordination in einem verteilten Zeitsystem erproben
2. Dezentralen wechselseitigen Ausschluss kennenlernen
3. Gruppenkommunikationssystem implementieren

Vorbemerkungen:

Zur Datenübertragung über Funk werden häufig sogenannte Zeitmultiplexverfahren (TDMA – *Time Division Multiple Access*) eingesetzt. Vorteilhaft ist, dass dazu nur ein einziger Kanal benötigt wird, der von mehreren teilnehmenden Stationen zeitlich gestaffelt zum Senden benutzt werden kann. Die Herausforderung von TDMA-Verfahren besteht in der Koordination und Synchronisation, da zu einer Zeit natürlich immer nur ein Sender sprechen darf. Unabhängig davon sind alle Stationen ständig empfangsbereit.

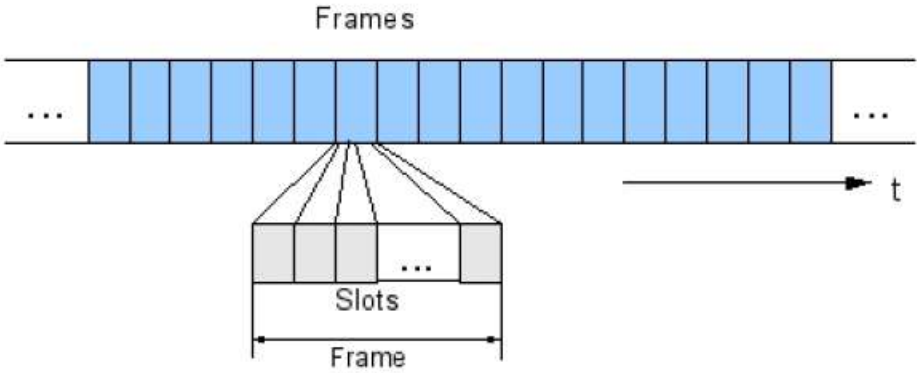


Figure 1: Anordnung von Slots in Frames

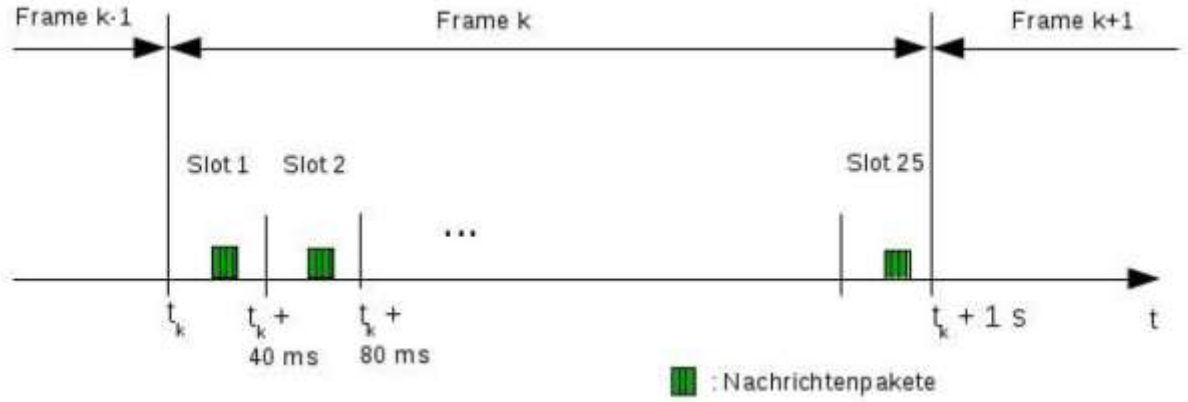
Beim Slotted TDMA-Verfahren wird die Zeitachse in gleich große **Frames** (Rahmen) unterteilt, die wiederum jeweils in *n* **Slots** (Zeitschlitze) unterteilt werden. Jeder Slot ist dann ein Sendeplatz für eine der Stationen.

Aufgabenstellung:

In dieser Aufgabe soll eine Sende-/Empfangsstation gemäß nachfolgenden Vorgaben implementiert werden, die Daten mithilfe eines solchen Verfahrens über einen Kanal sendet und empfängt. Anstelle eines Funkkanals soll hier *IP Multicast* mit einer vereinbarten Adresse und Empfangsportnummer verwendet werden.

Ablauf des Übertragungsverfahrens:

Jede Station soll genau einmal pro Frame senden. Die Vergabe der Slots soll ohne zentrale Instanz erfolgen (sog. STDMA-Verfahren). Jede Station muss selbstständig einen freien Slot zum Senden finden. Hierzu gibt es in den Nachrichtenpaketen ein Datenfeld, in das eine sendende Station die Nummer des Slots einträgt, in dem sie im nächsten Frame senden wird. Generell ist so zu verfahren, dass Kollisionen vermieden werden, andererseits aber die Ressourcen des Kanals ausgenutzt werden.
Damit das Verfahren mit allen Stationen aller Lösungen funktionieren kann, gelten nachfolgende Definitionen:



Ein Frame dauert 1 Sekunde. Jeder Frame besteht aus 25 Slots. Eine Station soll in der Mitte ihres gewählten Slots senden. Der Betrieb soll mit bis zu 25 Stationen möglich sein. Die Frames sind in Sekunden seit dem 1.1.1970, 00:00 Uhr nummeriert.

Synchronisation der Uhren:

Die Stationen sollen in die Klassen **A** und **B** unterteilt werden. Die Zeiten sollen grundsätzlich auf UTC basieren. Die Uhr einer Station der Klasse **A** wird als hinreichend genau betrachtet. Die Uhren dieser Stationen sollen sich im Rahmen der Möglichkeiten untereinander synchronisieren. Beim Start soll eine anfängliche Abweichung von der UTC einstellbar sein.
Die Uhren der Stationen der Klasse **B** gelten als nicht hinreichend genau. Sie müssen sich mit den Uhren von Stationen der Klasse A synchronisieren.

Format und Semantik der Nachrichtenpakete:

Die Nachrichtenpakete haben einheitlich nachfolgendes Format:

Byte 0:	Stationsklasse ('A' oder 'B')
Byte 1 – 24:	Nutzdaten. (Darin Byte 1 – 10: Name der sendenden Station.)
Byte 25:	Nummer des Slots, in dem die Station im nächsten Frame senden wird.
Byte 26 – 33:	Zeitpunkt, zu dem dieses Paket gesendet wurde. Einheit: Millisekunden seit dem 1.1.1970 als 8-Byte Integer, Big Endian.

Gesamtlänge: 34 Bytes

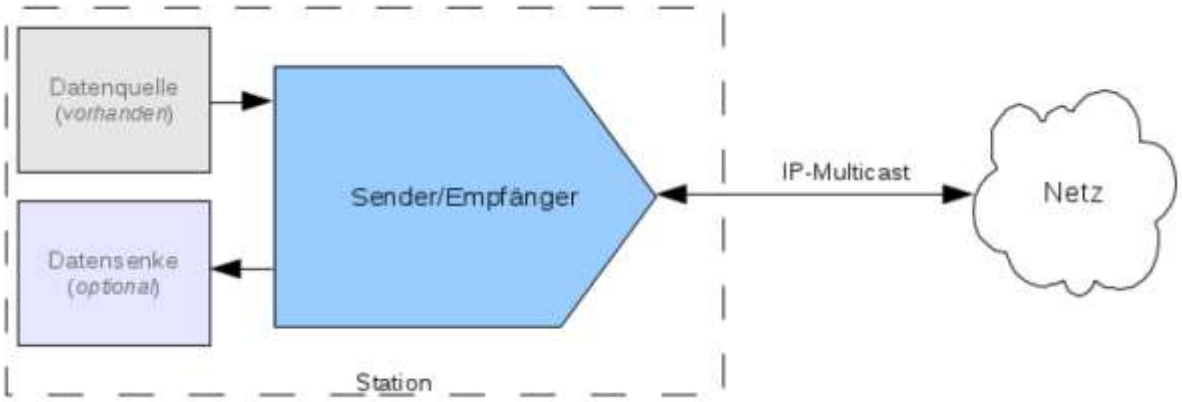
Multicast-Adressen und Ports für den simulierten Funkkanal:

Multicast-Adresse, Empfangsport sowie das zu verwendende Netzwerkinterface müssen per Parameter beim Programmstart einstellbar sein. Um Interferenzen zwischen den Teams im Praktikum zu vermeiden, sollen in der Entwicklungsphase nachfolgende Werte verwendet werden:

Adresse: 225.10.1.2
Empfangsport: 15200 + <*Ihre Teamnummer*>

Datenquelle und Datensenke:

Die Bereitstellung und Entgegennahme der Nutzdaten sollte von gesonderten Komponenten übernommen werden.



Als Datenquelle wird eines der herunterladbaren Programme[1] benutzt. Die Programme liefern periodisch 24-Byte-Nutzdatenpakete, die in den Bytes 0-9 einen identifizierenden Stationsnamen als Text im Format team <*Team-Nnr.*>-<*Stationsnr.*> enthalten, also z.B. team 02-01.

Hinweise:

- Zum Testen steht ein Sniffer[2] zur Verfügung.
- Für den Entwicklungs- und Testbetrieb im Labor wird die Benutzung von Linux empfohlen. Die Vorführung und die Prüfung der Abgabe erfolgen auf den Laborrechnern unter Linux.
- Auf den Rechnern in Raum 765 laufen unter Linux je fünf Stationen zum Testen. Sie werden beim Systemstart automatisch gestartet. Näheres hier[3].
- Kollisionen:
  - Bei Kollisionen gelten die beteiligen Pakete als nicht auswertbar
  - Der kollisionsfreie Betrieb muss nach kurzer Zeit erreicht werden und darf nicht wieder verlassen werden. (Voraussetzung: Anzahl der Stationen nicht größer als die Zahl der Slots im Frame.)
  - Bei kleinen Änderungsraten der Referenzzeit (bis ¼ Slot-Länge pro Frame) dürfen keine Kollisionen auftreten.
- Die Datenquelle ist als Messwertaufnehmer anzusehen, d.h. der Sender nimmt immer die aktuell verfügbaren Daten. Die Aktualität der gesendeten Daten ist der Genauigkeit der Zeiten untergeordnet.
- Die Programme sind von der Kommandozeile zu starten, so dass sie auch in einer *ssh*-Sitzung auf einem anderen Rechner ferngestartet werden können: Startskript[4]

Quellenhinweis: Diese Aufgabe wurde ursprünglich von Herrn Hartmut Schulz entworfen.

[1] https://www.inet.haw-hamburg.de/teaching/materials/vs/datasource.zip
[2] https://www.inet.haw-hamburg.de/teaching/materials/vs/sniffer.zip
[3] https://www.inet.haw-hamburg.de/teaching/materials/vs/readme-teststations.pdf
[4] https://www.inet.haw-hamburg.de/teaching/materials/vs/shellscripts.zip