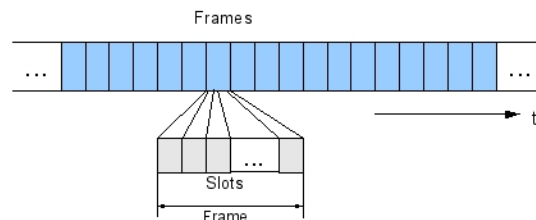


Vorbemerkungen

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. Unabhängig davon sind alle Stationen ständig empfangsbereit.

Die Zeitachse wird dazu in gleichgrosse **Frames** (Rahmen) unterteilt, die wiederum jeweils in n **Slots** (Zeitschlitz) unterteilt werden. Jeder Slot ist dann ein Sendeplatz für eine der Stationen.

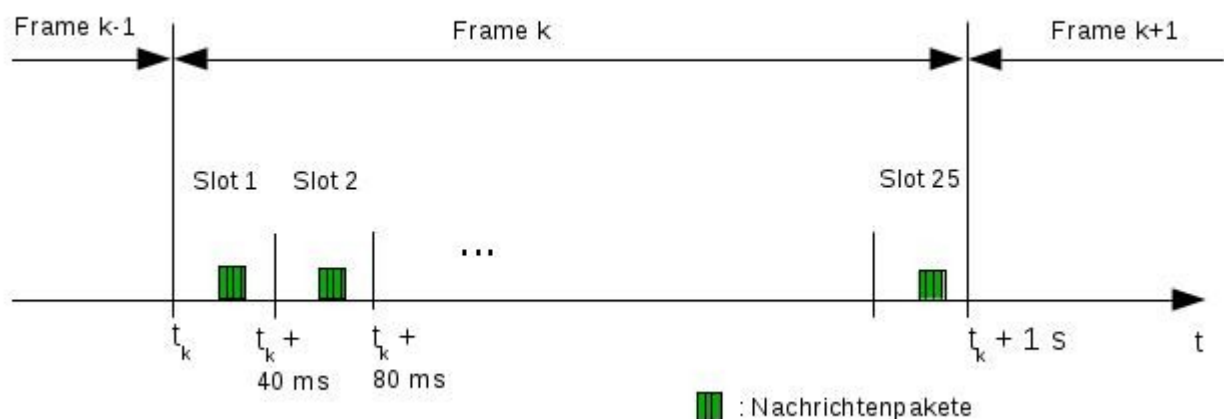


In dieser Aufgabe soll eine Sende-/Empfangstation implementiert werden, die Daten mithilfe eines solchen Verfahrens über einen Kanal sendet und empfängt. Anstelle eines Funkkanals soll hier *IP-Multicast* mit einer vereinbarten Adresse und Empfangsportnummer verwendet werden.

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

Für die Nachrichtenpakete ist einheitlich nachfolgendes Format zu verwenden:

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

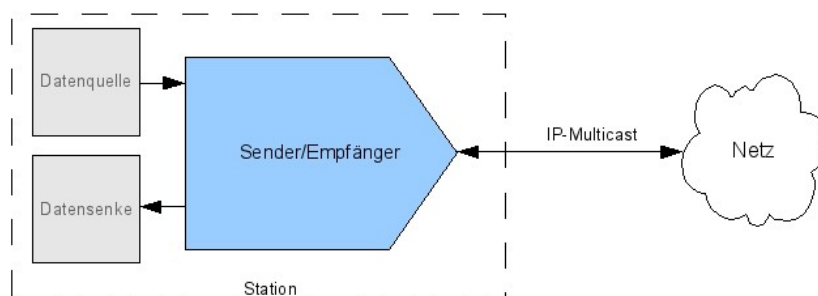
Multicast-Adresse, Empfangsport sowie das zu verwendende Netzwerkinterface müssen per Parameter beim Programmstart einstellbar sein.

Um Interferenzen zwischen den Teams im Praktikum zu vermeiden, sollen in der Entwicklungsphase nachfolgende Werte verwendet werden:

Adresse:	225.10.1.2
Empfangsport:	15000 + <Ihre Teamnummer>

Datenquelle und Datensenke

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



Als Datenquelle ist in jedem Fall eines der Programme zu benutzen, die [hier heruntergeladen](#) werden können.

Die Datenquelle liefert 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`.

Logging

Erkannte Kollisionen sollen von der Station protokolliert werden. Hat die Station dabei auch selbst in diesem Slot gesendet, soll dies ebenfalls vermerkt werden.

Ferne soll ferner die Anzahl der Frames in denen gesendet wurde und die Anzahl der Frames in denen nicht gesendet wurde geloggt werden.

Das Logging kann auch über die Datensenke erfolgen.

Hinweise

- Zum Testen steht ein [Sniffer](#) zur Verfügung.
- Für den Entwicklungs- und Testbetrieb im Labor wird die Benutzung von Linux empfohlen. (U.a. weil nur hier alle Netze verfügbar sind. Netzwerkplan [hier](#)). Die Vorführung und die Prüfung der Abgabe werden unter Linux erfolgen.
- Bei Kollisionen gelten die beteiligten Pakete als nicht empfangen bzw. nicht auswertbar.
- Kollisionen, die a priori zu erkennen sind, sind zu verhindern. Der kollisionsfreie Betrieb muss erreicht und darf bei unveränderter Stationsanzahl nicht wieder verlassen werden. Ist die Anzahl der Stationen kleiner als die Zahl der Slots im Frame, darf eine neu hinzukommende Station keinerlei Kollisionen verursachen.
- Bei kleinen Änderungsraten der Referenzzeit (bis $\frac{1}{4}$ Slotlänge 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, sodass sie auch in einer `ssh`-Sitzung auf einem anderen Rechner ferngestartet werden können. Es empfiehlt sich hier, das für die Abgabe ohnehin vorgeschrieben [Startskript](#) zu benutzen.
- **Die TTL der Multicast-Pakete ist auf 1 zu setzen, um unnötige Netzlasten und -störungen zu vermeiden. Das gilt auch in den freien Übungszeiten.**

Vorbereitung: Bis zum **Sonntagabend 20:00 Uhr** vor dem Praktikumstermin ist ein Konzeptpapier als **PDF**-Dokument per E-Mail mit **Teamnummer**, **Namen** und **Mailadressen** beider Teammitglieder über den Verteiler [Abgabeverteiler](#) abzugeben. Darin sind die wesentlichen Abläufe mit geeigneten UML-Diagrammen zu beschreiben. Hierbei ist zu erklären, welche Kollisionen mit den verwendeten Strategien verhindert werden können und welche nicht (Begründung!). Ferner muss der ausgefüllte [Dokumentationskopf](#) enthalten sein.. Bis zum Praktikumstermin ist der Entwurf ggf. soweit zu verfeinern, dass die Implementierung im Praktikum abgeschlossen werden kann.

Vorführung: Die Lösung muss auf den Laborrechnern unter Linux laufen. Um **15:00** beginnt eine gemeinsame Vorführung mit den Lösungen aller Teams.

Abgabe als ZIP-Archiv **am Ende des Praktikums**. Wird die Lösung beanstandet, ist genau eine Überarbeitung möglich. Diese wird bis spätestens Donnerstag 20:00 Uhr nach dem Praktikum akzeptiert. Gibt es auch dann noch Beanstandungen, gilt die Aufgabe als nicht erfolgreich

bearbeitet.

Der Inhalt des Archivs muss wie folgt strukturiert sein:

- für Ihre Stationen angepasstes Startskript (auf der obersten Verzeichnisebenen),
- Binaries, so, dass sie unmittelbar nach Entpacken mit dem Startskript gestartet werden können,
- Quellcode,
- aktualisierter Dokumentationskopf.

Abgaben, die diese Form nicht erfüllen oder unvollständig sind, werden nicht akzeptiert und als nicht erfolgreicher Abgabeversuch gewertet.

Voraussetzung für die erfolgreiche Bearbeitung ist eine fristgerechte, ausreichende Vorbereitung. Die Befragung und die Vorführung müssen erfolgreich absolviert werden. Aus organisatorischen Gründen sind alle gesetzten Termine verbindlich und unbedingt einzuhalten. Im Übrigen gelten die aus den vorangegangenen Aufgaben bekannten Regelungen.