AKUSTISCHES UND MOBILES FRONT-END FÜR EIN PUBLIC TERMINAL

Thomas Fehér, Sören Wittenberg und Rüdiger Hoffmann

Institut für Akustik und Sprachkommunikation, TU Dresden
thomas.feher@ias.et.tu-dresden.de

Kurzfassung: Public Terminals finden ein breites Anwendungsgebiet, vom Geldautomaten über den Fahrkartenautomat bis zum Informationsterminal für Touristen. Eine gute Bedienbarkeit ist jedoch nicht immer gewährleistet. In einem Verbund-Projekt der TU Dresden und lokalen mittelständigen Betrieben wurde ein Prototyp eines Public Terminals mit Sprachsteuerung und Sprachausgabe entwickelt. In diesem Beitrag werden die beiden Teilaufgaben des akustischen Front-Ends für die Sprachein- und ausgabe und des mobilen Front-Ends für die Ein- und Ausgabe mittels tragbarem Gerät, wie zum Beispiel einem Mobiltelefon oder PDA, vorgestellt. Für das akustische Front-End wurden Mikrofon- und Lautsprecherarrays in das Terminal integriert und mit Beamformingalgorithmen angesteuert. Das mobile Front-End stellt ein reduziertes Abbild des Public Terminals dar und ist mittels Bluetooth an diesen angebunden. Die Einbeziehung des geräteinternen Mikrofons bzw. Lautsprechers ermöglicht auch dem mobilen Front-End die Bedienung per Sprache.

1 Einleitung

Im Rahmen des Projektes "Multimodales, personalisiertes Bedienkonzept für Public Terminals"[1] wurde der Prototyp eines Public Terminals entworfen und gebaut, mit dem eine multimodale Bedienung sowie eine Bedienung durch mehrere Benutzer gleichzeitig möglich ist.

Multimodal heißt in diesem Fall, die Bedienung kann mit Hilfe des Touch-Screens und über Sprachein- und ausgabe erfolgen. Die Nutzung des Terminals durch mehrere Personen erfolgt mit Hilfe des mobilen Frontends (siehe Abschnitt 3). Dieses erlaubt es alle Funktionen mit einem mobilen Geräte (Handy, PDA, etc.) über eine Bluetooth Verbindung zu nutzen. Dadurch können in der Praxis lange Warteschlangen an Verkaufs- oder Informationsterminals verhindert werden, denn meist ist nicht die Leistungsfähigkeit des Terminals der begrenzende Faktor, sondern das Interface.

Im folgenden Beitrag werden die Teilgebiete des akustischen und mobilen Frontends näher erläutert.

Abbildung 1 - Public Terminal.

[1]Förderprojekt des AiF, Förderkennzeichen: KF0033704LF8

2 Akustisches Front-End

2.1 Eingabe

2.1.1 Mikrofonarray

Zur robusten Eingabe der Sprachsignale an den Erkenner kommt ein Mikrofonarray zum Einsatz. In Abbildung 2 ist die genutzte Anordnung zu sehen. Es wurde versucht einen Kompromiss aus Mikrofonanzahl, Ausdehnung des gesamten Arrays und Bündelung zu finden, da bei dem Prototyp auch spätere ökonomische Randbedingungen beachten werden mussten. Gewählt wurde ein Mikrofonarray mit 4 nicht äquidistanten Elektretkapseln.

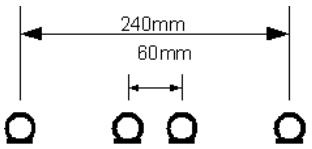

Abbildung 2 - Geometrie des Mikrophonarrays.

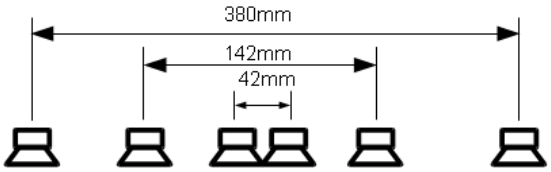

Abbildung 3 - Geometrie des Lautsprecherarrays.

2.1.2 Beamforming-Algorithmus

In [1] konnten sehr gute Erkennungsraten mit einem Minimum Variance Distortionles Response (MVDR) Beamformer erzielt werden werden. Dieser wurde statisch mit auf diffusen Schalleinfall optimierter Richtwirkung betrieben (superdirektiver Beamformer). Es kamen hochwertige Studiomikrofone zum Einsatz. Diese Ergebnisse konnten aber bei der Messung am Prototypen nicht reproduziert werden. Je nach Einfallswinkel und Signal-Stör-Abstand (SNR) bewegen sich die Erkennungsraten des MVDR Algorithmus unter denen des einfacheren Delay-And-Sum (DSB) Algorithmus. Der Grund dafür liegt in der bekannten Empfindlichkeit des MVDR gegenüber Ungenauigkeiten bei der Arraygeometrie sowie Phasen- und Frequenzgang der verwendetetn Mikrofone. Die in diesem Fall verwendeten handelsüblichen Elektretkapseln sind dabei wesentlich höheren Toleranzen unterworfen als die vorher verwendeten Studiomikrofone.

Daher wurde im Folgenen nach eher datengetriebenen Algorithmen gesucht, welche ein geringes Vorwissen über die vorhandene Array-Architektur benötigen, und adaptiv auf die anfallenden Audiodaten reagieren.

Im Rahmen des Projektes wurden 2 Varianten eines Postfilters (PF) auf Basis des Wiener-Filters getestet und mit Einzelmikrofonen, Delay and Sum Beamformer (DSB) und dem superdirektiven MVDR Beamformer, wie in [1], verglichen.

Die beiden Varianten unterscheiden sich in der Schätzung von Stör- und Nutzleistungsdichtespektrum, da diese nicht bekannt sind. Die erste Variante des Wiener-Filters ist der Algorithmus nach [2], bei dem das LDS des Nutzsignals durch

$$\hat{\phi}_s(k) = \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \mathrm{Re}\{\hat{\phi}_{X_i X_j}(k)\} \tag{1}$$

und das LDS der Summe aus Nutz- und Störsignal durch

$$\hat{\phi}_{s,v}(k) = \frac{1}{M}\sum_{i=1}^{M}\hat{\phi}_{X_iX_i}(k) \tag{2}$$

geschätzt werden. M ist die Anzahl der verwendeten Mikrofone und $\hat{\phi}$ das geschätzte LDS. X_i ist das fourier-transformierte Signal des i-ten Mikrofons. Genutzt wird bei diesem Verfahren, dass sich bei dem Kreuzleistungsdichtespektrum die unkorrelierten Signalanteile zu Null ergeben und dadurch nur die korrelierten Anteile abgebildet werden. Die Mittelung der Autoleistungsdichtespektren aller Einzelmikrofone wie in Gleichung (2) ergibt das LDS der Summe aus Nutz- und Störsignal. Daraus kann das Störsignal durch spektrale Subtraktion extrahiert werden.

Die Zweite Variante unterscheidet sich in der Schätzung des Störsignals. Dabei werden die Mikrofonsignale paarweise von einander abgezogen und dann das LDS berechnet. Das subtrahieren der Signale ergibt einen Schalldruckgradienten in Richtung der Verbindungslinie zwischen den beiden Mikrofonen. Dieser Gradient ist proportional zur Schallschnelle in die gleiche Richtung, welche für ebene Wellen wiederum proportional zum Schalldruck ist. Dadurch entsteht die Richtcharakteristik "Acht" quer zur bevorzugten Schalleinfallsrichtung des Arrays. Das ist vergleichbar mit der Blockingmatrix bei dem Generalized Sidelobe Canceler (GSC).

$$Y_l = X_i - X_j \qquad \forall i \neq j \tag{3}$$

Für das LDS des Störsignals ergibt sich folgende Formel

$$\hat{\phi}_v(k) = \frac{1}{M}\sum_{l=1}^{M}\hat{\phi}_{Y_lY_l}(k) \tag{4}$$

Die Schätzung des Nutzsignals erfolgt nach Formel (1).

2.1.3 Messungen

Mit dem Prototypen des Terminals wurden Impulsantworten in einem halligen Büroraum (Nachhallzeit $T_{60} \approx 1s$) aufgenommen. Die Aufnahme der Impulsantworten erfolgte für Winkel von 0 bis 90 Grad in 15 Grad Schritten und einer Entfernung von 0,8 m. Dadurch können in späteren Versuchen Stör- und Nutzschallquelle in beliebigem Winkel zum Array angeordnet werden.

Mit diesen Impulsantworten wurden Erkennerexperimente mit dem UASR-Erkenner des IAS vorgenommen. Zur Verwendung kam ein Wortschatz von 1020 Kommandos aus dem Apollo Korpus. Die Erkennungsraten der ungestörten Kommandos liegt bei ca. 97%. Die Ergebnisse für verschiedene Einfallswinkel des Störsignals und verschiedene Signal-Rausch-Abstände sind in Abbildungen 4 und 5 zu sehen. Da der Erkenner mit ungestörten Signalen trainiert wurde, ist durch ein spezielles Training mit eingeschaltetem Postfiltering ein zusätzliche Verbeserung der Erkennungsraten zu erwarten.

Das Postfiltering konnte eine deutliche Verbesserung der Erkennungsraten erzielen, wobei beide Algorithmen in etwa gleich gut arbeiten. Interessant ist, dass auch ohne Störsignal, eine Verbesserung erfolgt. Die Nachfilterung arbeitet demnach auch als Enthallung.

2.2 Ausgabe

2.2.1 Lautsprecherarray

Durch die Reziprozität der untersuchten Algorithmen bzw. des Beamformings im Allgmeinen, können die gewonnenen Erkenntnisse ebenfalls für die gerichtete Beschallung genutzt werden.

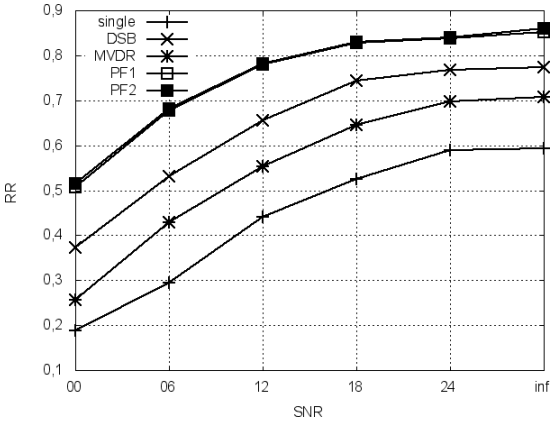

Abbildung 4 - Abhängigkeit der Erkennungsrate vom SNR, Einfallswinkel der Störschallquelle: 90 Grad

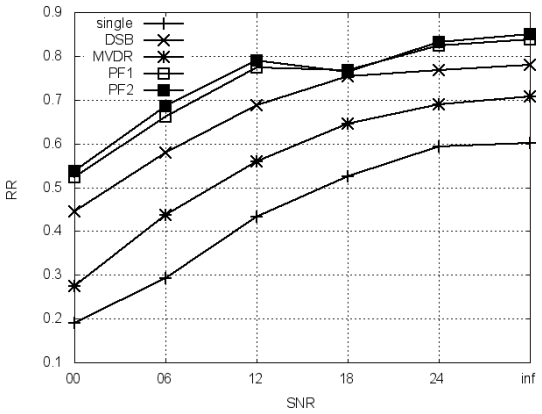

Abbildung 5 - Abhängigkeit der Erkennungsrate vom SNR, Einfallswinkel der Störschallquelle: 45 Grad

Dabei erweißt sich die Anhebung bei tiefen Frequenzen durch den MVDRB-Algorithmus jedoch als problematisch. Deswegen wird für die Beschallung lediglich der DSB-Algorithmus zum Einsatz kommen. Verwendet wurde die Anordnung aus Abbildung 3.

2.2.2 Messung

Bei den Messungen mit dem superdirektiven MVDR Beamformer erwiesen sich dieser als schlechter als der einfachere DSB, was, ähnlich wie bei den Mikrofonen, durch die geringe Qualität der Lautsprecher zuerklären ist. Die Ergebnisse der Messungen sind in den Abbildungen 6 und 7 zu sehen.

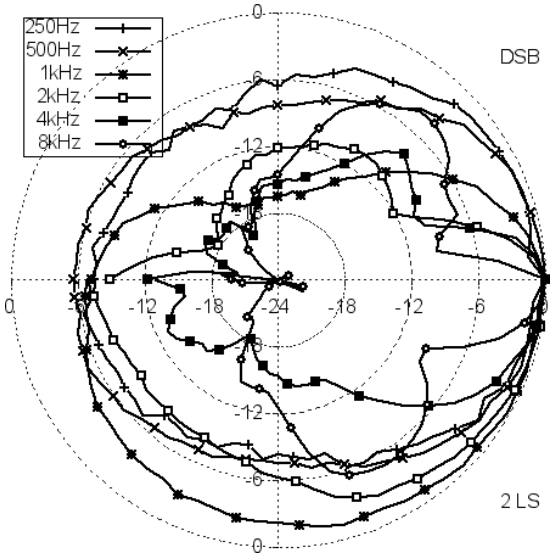

Abbildung 6 - Vergleich der Richtwirkung des DSB mit den 2 inneren Lautsprechern

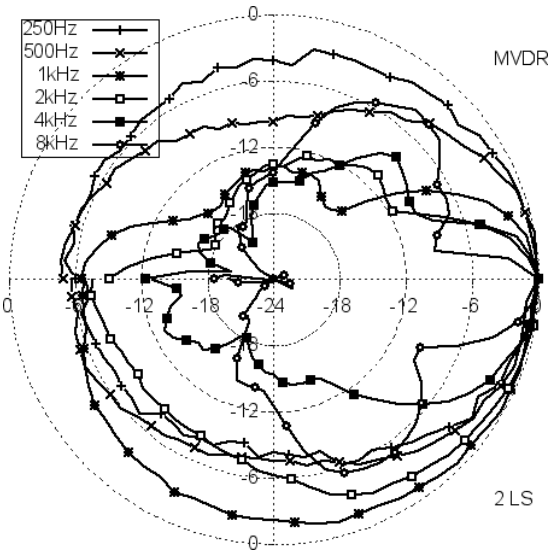

Abbildung 7 - Vergleich der Richtwirkung des MVDR mit den 2 inneren Lautsprechern

3 Mobiles Front-End

Die Integration des mobilen Front-End in einen Public Terminal (Abb. 8) erfolgt ohne spezielle Audiohardware, erlaubt aber dennoch die drahtlose Übertragung von Breitbandsprache. Als Übertragungstechnik wurde der Industriestandard *IEEE 802.15.1* bzw. Bluetooth gewählt, bei dem sich bis zu acht aktive Geräten eigenständig in einem sogenannten *Piconetz* organisieren.

Abbildung 8 - Grobstruktur des Systems

Bluetooth stellt zwei Verbindungsvarianten bereit. Anwendungen die geringe Latenzen benötigen, verwenden den synchronen, verbindungsorientierten Datentransfer (*synchronous connection oriented* (*SCO*)). Die garantierte Nettodatenübertragungsrate (bidirektional) beträgt 64 Kbits/s. Die Daten werden dabei nur ein einziges Mal übertragen und es findet keine Verifikation der Datenintegrität statt. Konzipiert ist dieser Verbindungstyp für Telefonsprache. Diese Qualität kann in der Spracherkennung zu ungenügenden Erkennungsergebnissen und die Ausgabe von synthetisierter Sprache zu erhöhter Höranstrengung führen.

Die asynchrone verbindungslose Variante (engl. *asynchronous connectionless link* (*ACL*)) wird für paketorientierten Datentransfer verwendet und erlaubt mit *EDR* (enhanced data rates) eine Bruttodatenübertragungsrate von 3 Mbit/s, dies allerdings mit höherer Latenz. Zur Übertragung hochqualitativen Audio kann das auf *ACL* basierende *Advanced Audio Distribution Profile* (*A2DP*) verwendet werden, vorausgesetzt beide Bluetooth-Geräte besitzen dieses Profil und den gleichen Übertragungscodec (z.B. der verlustbehafteter *Subband Codec* (*SBC*)). Durch fehlende Programmierschnittstellen sind diese von den Geräteherstellern vorgegeben. Gegenwärtig bietet kein Bluetooth-Headset die Möglichkeit verlustlos unkomprimiert *PCM* Breitbandsprache bidirektional zu übertragen. Deshalb bietet die vorgestellte Alternative die Möglichkeiten den eingebautem Lautsprecher und das Mikrofon von Mobiltelefonen mit Bluetooth-Einheit und Unterstützung von *JavaME* (Java Platform Micro Edition) zu nutzen. Da das System nicht für die Mensch-Mensch-Kommunikation konzipiert ist, sind Latenzen wenig kritisch.

Die drahtlose Anbindung von bis zu sieben Mobiltelefonen an einen Public Terminal (Abb. 9) erfolgt mit einer *Middleware*. Diese in der Programmiersprache Java formulierte Software und die in *JavaME* für Mobiltelefon verarbeiten die über die Bluetoothschnittstelle ein- und ausgehenden Daten. Da weder *SCO* noch *A2DP* möglich ist, wurde ein symmetrischer Datentransfer implementiert, der auf *Serial Port Profile* basiert, welches mittels *RFCOMM* über *ACL* läuft. [3] benennt eine ähnliche Technologie *voice over ACL*

Um die akustischen Eigenschaften des Mobiltelefons ohne die Wirkung dessen verlustbehafteten (*AMR*- Encoder aufzunehmen, werden die Audiodaten unkomprimiert mit mindestens 16 kHz bei 16 bit pro Abtastwert (*PCM*-Format) bidirektional übertragen. Die minimal notwendige Datenübertragungsrate beträgt mindestens 256 kbit/s pro Kommunikationsrichtung.

3.1 Mobiltelefon

Die verwendeten Mobiltelefone waren das *K550i* und *K610i* der Firma Sony Ericsson. Das *K550i* arbeitet mit einem Prozessor der *ARM*-Familie und einer Taktfrequenz von 200 MHz. Die Bluetootheinheit unterstüetzt Version 2.0 mit *EDR*. Die integrierte Sony Ericsson Java Platform 7 (*JP-7*) unterstützt Connected Limited Device Configuration 1.1, Mobile Information Device Profile 2.0, Mobile Media Application Programing Interface und *API for Bluetooth*. Werden

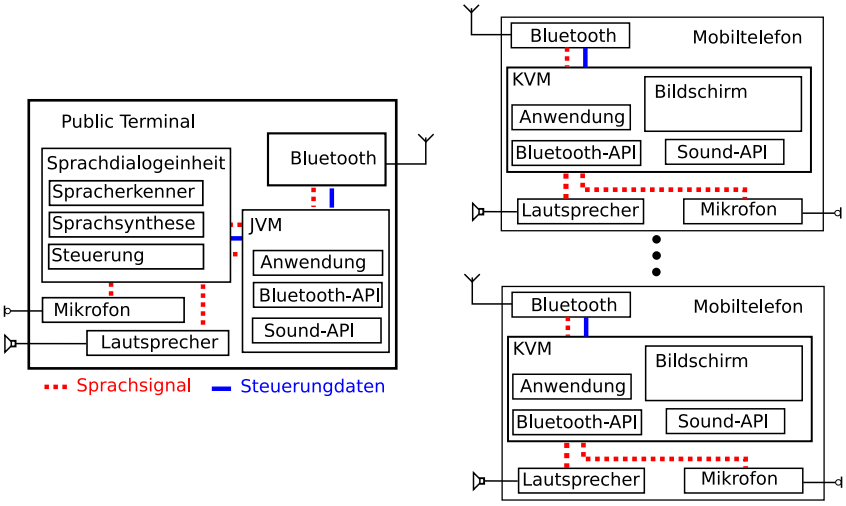

Abbildung 9 - Blockschaltbild der Systemstruktur

die genannten Spezifikationen unterstützt, erlaubt dies der Java *MIDP* Anwendung (*MIDlet*) in einer wohldefinierten Umgebung abzulaufen.

Das *MIDlet* besteht aus drei Threads. Der erste Thread koordiniert die Interaktionen mit dem Benutzer. Der zweite Thread analysiert den ankommenden Datenstrom und wertet den Kopfteil jedes Datenpakets aus. Anhand dieser Information leitet er die Daten zur Audioausgaberoutine, zum Bildschirm oder den Steuerungsalgorithmen. Der dritte Thread behandelt die Mikrofonaufnahmen, indem er die aufgezeichneten Audiodaten paketiert, mit einem Paketkopf versieht und an die Bluetoothschnittstelle übergibt. Es können so auch Steuerbefehle und visuelle Informationen ausgetauscht werden. Zur Vermeidung akustischer Rückkopplungen wird dieser Thread während der Wiedergabe angehalten.

3.2 Public Terminal

Der Public Terminal wurde durch einen Personal Computer mit dem Betriebssystem *Microsoft Windows XP SP2* und einem Bluetooth 2.0 USB Dongle mit *EDR* simuliert. Es wird der native Bluetooth-Support des Betriebssystem (*winsock*) verwendet. Der Bluetoothzugriff erfolgt durch die OpenSource Implementierung des Bluetooth Stack für Java *BlueCove*. Die erstellte Software befähigt den Public Terminal mittels Bluetooth Object Push Profile (*OBEXpush*) das *MIDlet* an mobile Endgeräte zu verteilen und als Dienst mittels speziellen Adresse (*btspp://*) den Zugang zum Public Terminal bereitzustellen. Die Adresse identifiziert den Protokolltyp *Serial Port Profile*. Es sind bis zu sieben Sitzungen quasi parallel möglich, wobei jede Sitzung innerhalb eines eigenen Thread abläuft.

3.3 Experimente und Ergebnisse

Die wichtigste Größe ist die erzielbare Langzeitdatenübertragungsrate zwischen Mobiltelefon und Public Terminal. Kurzzeitiges Unterschreiten ist nicht kritisch, da Ausgangs- und Eingangspuffer Daten zwischenspeichern. Die durchgeführten Experimente untersuchten die frei beeinflussbaren Parameter: Entfernung, Paketgröße zur zwischen der Software und Bluetooth-Einheit, weitere aktive Endgeräte. Bei allen Experimenten bestand zwischen den beteiligten Geräte Sichtkontakt. Die Bestimmung der Datenübertragungsraten erfolgte unter der Voraussetzung einer symmetrischen Verbindung, wodurch es legitim ist nur eine Kommunikationsrichtung zu untersuchen. Dies ist notwendig, da vom Public Terminal mehr Daten bereitgestellt werden können, als es die Echtzeitaufnahme des Mobiltelefons zuläßt.

Tabelle 1 - *Paketanzahl bei unterschiedlichen Paketgrößen (a) 256 ms and b) 32 ms bei 16 kS/s)*

Länge [s]	Länge [Byte]	a)	b)
1	32,000	4	32
2	64,000	8	63
...	...	...	...
64	2,048,000	251	2000
128	4,096,000	501	4000

Ein Testlauf übertrug fünf Signaltypen (MFV-Töne, weisses Rauschen, Sprachsignal, 100 Hz Sinuston and 1000 Hz Sinuston) mit 20 verschiedenen Längen (Tabelle 1) vom Public Terminal zum Mobiltelefon. Die Datentransferraten wurden für jede Signallänge aus allem fünf Signalen bestimmt. Da das System paketorientiert arbeitet, enthält die Tabelle die Anzahl der zu übertragenden Pakete (jedes Paket enthält zusätzlich 5 Byte für den Paketkopf). Die Bluetooth-Übertragung hingegen ist in Zeitschlitzen organisiert, wobei üblicherweise ein Datenpaket küerzer als ein Zeitschlitz ist. Um den Datendurchsatz zu steigern, werden die Pakete größer als die zur Verfügung stehende Zeitschlitzlänge gemacht [4] und es entstehen *Multi Slot Packets* die mehrere Zeitschlitze belegen aber nur einen Header senden. Dieses Vorgehen blockiert Zeitschlitze, die für andere aktive Geräte im Piconetz gedacht sein könnten.

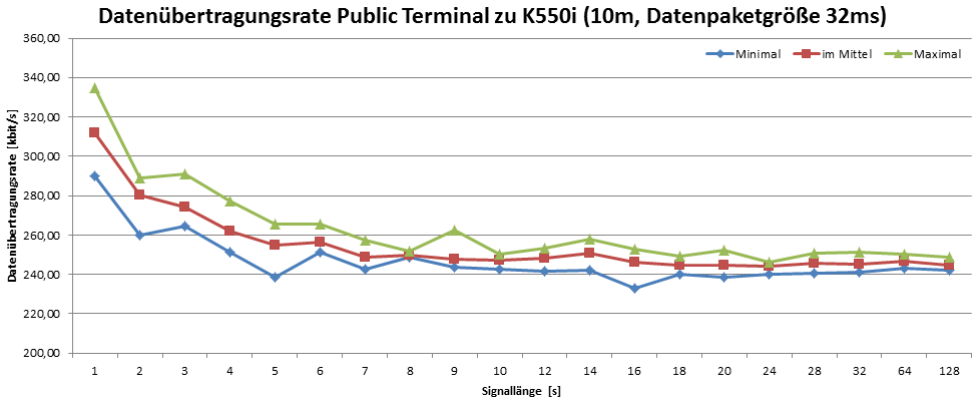

Abbildung 10 - Datenübertragungsrate Public Terminal zu K550i bei 10 m und Paketgröße 32 ms.

3.4 Abhängigkeit von der Paketgröße

Abbildung 10 zeigt die minimale, mittlere und maximal erzielte Datenübertragungsrate bei einer Paketgröße von 32 ms. 32 ms ist dabei die entstehende Latenz die durch das Befüllen der Audiobuffer entsteht. Die durch das Betriebssystem und Bluetooth entstehenden Latenzen sind zu addieren. Da beide Systeme autonom arbeiten, ist eine exakte Messung nicht möglich. Aus diesem Grund wurde die Zeit vom ersten Aufruf der entsprechenden Sende-Methode bis zur Rückkehr aus dieser Methode nach deren letzen Aufruf gemessen. Durch dieses Vorgehen wurde die Datenübertragungsrate kurzer Signale und grossen Paketgrößen zu stark positiv beeinflusst. Am aussagekräftigsten sind die kleinsten erzielten Übertragungsraten, welche bei langen Signalen anzutreffen sind. Der kleinste erzielte Werte ist etwa 240 kbit/s bei einer Paketgröße von 32 ms. Diese entspricht nicht den notwendigen 256 kbit/s, weshalb zu einer größeren Paketgröße übergegangen wurde. Die jeweils kleinsten Datentransferraten der Versuche mit anderen Paketgrößen sind in der Tabelle 2 zusammengetragen.

Tabelle 2 - *Minimale Datenübertragungsrate bei verschieden Paketgrößen (Public Terminal - K550i)*

Latenz [ms]	32	64	128	192	256	320
min. Datenübertragungsrate [kbit/s]	240,05	361,65	476,99	537,51	575,39	597,50

3.5 Abhängigkeit von der Entfernung und Performanz bei zwei mobilen Endgeräten

Eine Abhängigkeit von der Entfernung konnte in der räumlichen Begrenzung der Testumgebung nicht festgestellt werden. Die Datenübertragungsraten bei 1 m und 10 m zeigten vernachlässigbare Unterschiede. Dies ist hauptsächlich auf den direkten Sichtkontakt zwischen beiden Geräte zurückzuführen. Größere Enfernungen konnten nicht untersucht werden.
Das dritte Experiment stellte die Leistungsfähigkeit bei der gleichzeitigen Verwendung zweier Mobiltelefone fest. Abbildung 11 stellt das Ergebnis des parallelen Datentransfer vom Public Terminal zum K550i und dem K610i dar. Anders als erwartet sinkt die Datentransferrate von 570 kbit/s (ein aktives Gerät) nur auf 470 kbit/s pro Gerät. Es ist zu beachten, dass jedes Gerät auf dem Public Terminal durch einen eignen Thread repräsentiert wird und seine eigenen Daten bekommt. Die Daten wurden bei diesem Test nicht mittels Broadcast verteilt.

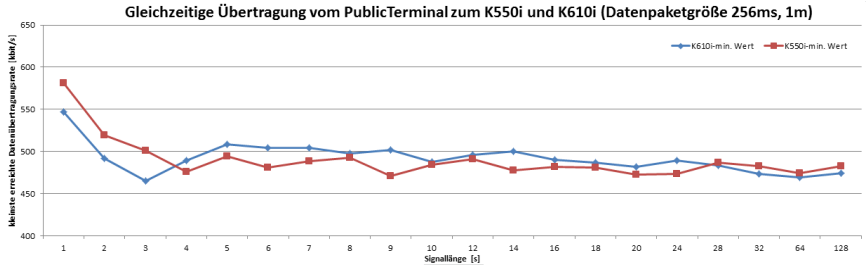

Abbildung 11 - Datentransferrate Public Terminal zu K550i und K610i bei 10 m und Paketgröße 256 ms

Literatur

[1] Fehér, T.; Petrick, R.; Hoffmann, R., “Mehrkanaliges akustisches Front-End für Spracherkennungssysteme”, Elektronische Sprachsignalverarbeitung 2009, 2009.

[2] Zelinski, R., “A microphone array with adaptive post-filtering for noise reduction in reverberant rooms”, in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), p.2578–2581, 1988.

[3] Russo, M.; Begusic, D.; Rozic, N. and Stella M.,“Speech Recognition over Bluetooth ACL and SCO Links: A Comparison”, Second IEEE Consumer Communications and Networking Conference, p.493-497, 2005.

[4] Ziirbes, S.; Stahl, W.; Matheus, K. and Haartsen, J.,“Radio Network Performance of Bluetooth”, IEEE International Conference on Communications, Vol. 3, p.1563-1567, 2000.