

POMIARY WARTOŚCI OPÓŹNIEŃ W TORZE AUDIO URZĄDZEŃ Z SYSTEMEM ANDROID

Maciej BLASZKE¹, Dawid WEBER², Szymon ZAPOROWSKI³

1. Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki
tel: 729-144-490, email: maciejb93@gmail.com
2. Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki
tel: 721-296-031, e-mail: dawweber92@gmail.com
3. Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki
tel: 58-348-6332, e-mail: smck@sound.eti.pg.gda.pl

Streszczenie: Poniższy artykuł opisuje metody pomiarów wartości opóźnienia w torze fonicznym urządzeń pracujących na różnych wersjach systemu Android. W pierwszej części artykułu podano krótką charakterystykę środowiska Android w kontekście opóźnień w torze fonicznym. Następnie przedstawiono sposób pomiaru opóźnień w torze fonicznym za pomocą aplikacji *SuperPowered Latency* oraz *Dr. Rick O'Rang Loopback*. W końcowej części pracy podano wyniki przeprowadzonych pomiarów i ich porównanie z wynikami dostępnymi na stronach producenta systemu Android. Zawarto również wnioski dotyczące możliwości implementacji aplikacji wykorzystujących tor foniczny i problemy powodowane przez opóźnienia sygnału.

Słowa kluczowe: urządzenia mobilne, system Android, tor foniczny, aplikacje multimedialne, przetwarzanie dźwięku.

1. WSTĘP

Jakość większości aplikacji audio przeznaczonych na urządzenia mobilne zależy w dużym stopniu od wartości opóźnień w torze fonicznym, między innymi w tej grupie aplikacji można znaleźć gry na urządzenia mobilne, stacje robocze dla producentów muzyki DAW (ang. *Digital Audio Workstation*), syntezatory i aplikacje wykorzystujące przetwarzanie dźwięku w czasie rzeczywistym. Głównym problemem dla twórców aplikacji na urządzenia z systemem Android jest opóźnienie w torze fonicznym, które skutkuje brakiem spełnienia wymagań stawianym przez użytkowników takich aplikacji. W praktyce jedynie opóźnienia sygnału fonicznego poniżej 10 ms są akceptowalne zarówno przez muzyków, słuchaczy, jak i konsumentów takich aplikacji. Większość aplikacji audio działających na platformie Android posiada opóźnienie rzędu powyżej 100 ms na wyjściu toru fonicznego. Jeżeli aplikacja obsługuje zarówno wejście, jak i wyjście, to opóźnienia są sumowane i ich wartość może być nawet większa niż 200 ms [1].

Poniżej podano przykłady aplikacji, które tracą swoją funkcjonalność, jeśli w torze fonicznym występują opóźnienia większe niż 10 ms:

- Aplikacje wirtualnych instrumentów - muzycy korzystający z takich aplikacji nie potrafią utrzymać odpowiedniego tempa utworu;

- DJ (dobór muzyki, odtwarzanie i miks muzyki) korzystający z aplikacji instrumentalnych nie potrafi utrzymać odpowiedniej rytmiki w utworze;
- Aplikacje korzystające z protokołu VOIP, takie jak np. Skype - latencja w torze fonicznym będzie wyższa niż opóźnienie w przesyłaniu pakietów w sieci, co powoduje wydłużoną formę komunikacji z powodu jeszcze wyższych i kumulowanych opóźnień przesyłania danych w sieci;
- Aplikacje bazujące na wirtualnej rzeczywistości mogą powodować desynchronizację pomiędzy obrazem wideo a ścieżką dźwiękową.

W niniejszym artykule przedstawiono pomiary wartości opóźnień w torze fonicznym kolejnych wersji systemu Android począwszy od wersji 5.0, aż do wersji 8.0). W pierwszej kolejności podano krótką charakterystykę toru fonicznego urządzeń pracujących w systemie Android. Następnie opisano metody i aplikacje wykorzystywane w pomiarach latencji toru fonicznego. W rozdziale 4 zawarto wyniki pomiarów dziesięciu urządzeń z różnymi wersjami systemu Android i porównano je z wynikami dostępnymi na stronie producenta systemu Android. W Podsumowaniu wskazano możliwe przyczyny rozbieżności uzyskanych wyników, a także możliwy kierunek kontynuacji badań poprzez stworzenie własnej aplikacji do pomiarów opóźnień w torze fonicznym urządzeń pracujących w systemie Android.

2. CHARAKTERYSTYKA TORU FONICZNEGO W SYSTEMIE ANDROID

2.1. Android 5.0

Pierwszym elementem w torze fonicznym w urządzeniu z systemem Android w wersji 5.0 jest wejście analogowe. Jest to układ złożony z przedwzmacniacza mikrofonu wbudowanego w urządzenie. Kolejny blok toru jest przetwornikiem ADC (ang. Analog-to-Digital Converter). Przetwornik ADC zawiera układ próbkujący w dziedzinie czasu i kwantujący w dziedzinie czasu, który pozwala na konwersję sygnału analogowego na postać cyfrową, następnie cyfrowy sygnał foniczny transportowany jest przez bufor. Transfer informacji z czipu audio urządzenia do sterownika odbywa się poprzez szynę danych w Androidzie 5.0 (Lollipop). Przekształcony do postaci

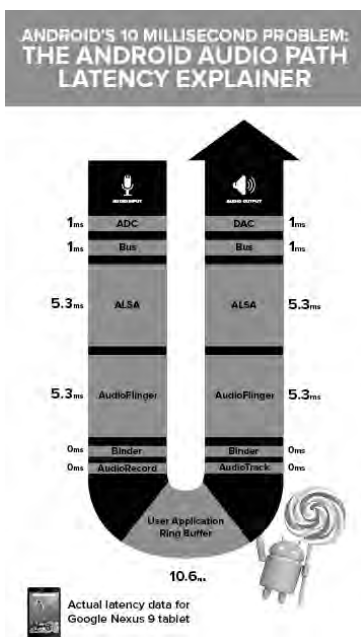
cyfrowej sygnał jest przekazywany do sterownika ALSA (ang. *Advanced Linux Sound Architecture*).

Architektura platformy Android jest strukturą warstwową. Elementem łączącym sterownik ALSA z serwerem mediów jest warstwa HAL (ang. *Hardware Abstraction Layer*). Serwer mediów systemu Android dostarcza buforowi HAL dane oraz tworzy ścieżki wejścia i wyjścia o zadanych parametrach takich jak: częstotliwość próbkowania czy rozmiar bufora przechowującego dźwięk. *AudioFlinger*, który tworzy wątek *RecordThread*, należy do warstwy natywnej środowiska Android i służy do komunikacji pomiędzy aplikacją i sterownikiem audio. Pozwala na zmianę rozmiaru bufora. Łącznik (ang. *Binder*) służy z kolei do transferu danych z bufora pomiędzy aplikacją użytkownika a *AudioFlingerem*. W torze istnieje też aplikacja, gdzie dane są przetwarzane, następnie docierają do elementu zwanego *AudioTrack* i są przesyłane na wyjście aplikacji, skąd trafiają ponownie do wszystkich elementów toru wyjściowego w odwrotnej kolejności [2].

2.2. Android 6.0

Wersja Android, nazwana *Marshmallow*, oznaczona numerem 6.0, używa mniejszych buforów, co powoduje, że sygnał foniczny jest analizowany w mniejszych pakietach. Powoduje to zmniejszenie latencji.

Kolejna dokonana zmiana w stosunku do wcześniejszej wersji systemu dotyczy poprawnej implementacji bufora HAL dla większości urządzeń sprzedawanych pod marką Nexus. Jednak największą zmianą w porównaniu do poprzedniej wersji platformy Android jest dodana obsługa komunikatów MIDI oraz znacznik *Professional Audio Flag*. Znacznik (flaga) może być używany na urządzeniach powyżej wersji 6.0, zapewnia latencję poniżej 20 milisekund w całym torze fonicznym, jak również kompatybilność urządzeń z USB oraz MIDI. Na rys. 1 przedstawiono wartości opóźnień w torze fonicznym dla środowiska Android w wersji 6.0.



Rys. 1. Tor foniczny w systemie Android 6.0 dla tabletu Google Nexus 9 [2]

Pomimo znaczącego zmniejszenia latencji w torze fonicznym tej wersji systemu Android, pozostaje problem zapewnienia niższego opóźnienia dla aplikacji mobilnych

obsługiwanych przez system Android [3]. Jest to możliwe poprzez implementację kodu w C++, który jest językiem natywnym środowiska Android. Aby opracować aplikację na urządzenia mobilne z systemem Android, należy stworzyć platformę łączącą za pomocą Java Native Interface [4], dla lepszej komunikacji pomiędzy częścią aplikacji napisanej w języku Java a częścią pisaną w języku programowania C++. Dodatkowo należy pamiętać o kilku zasadach [5]:

- Należy opracować jak najbardziej optymalny kod, który dopasuje się do dowolnego rozmiaru bufora oraz częstotliwości próbkowania;
- Nie należy stosować gotowych rozwiązań oferowanych przez Google do przygotowania aplikacji związanych z przetwarzaniem dźwięku na platformie Android;
- Należy wziąć pod uwagę różne typy sygnałów w ścieżce audio systemu Android i pamiętać o różnej wartości opóźnienia dla każdego z nich;
- Nie należy wykorzystywać modułu Bluetooth w aplikacji, pozwoli to uniknąć opóźnień w torze fonicznym rzędu powyżej 100 ms.

Google niestety nie dostarcza informacji na temat struktury toru audio w wersjach Androida oznaczonych numerem 4.0, 7.0 oraz 8.0.

3. POMIARY

3.1. Metody pomiarowe

Jedną z najprostszych metod pomiarowych jest test odpowiedzi toru fonicznego, zwany efektem Larsena. Efekt ten odnosi się do akustycznego sprzężenia zwrotnego. Pozwala to na pomiar wartości opóźnień na wejściu i na wyjściu toru. Jednakże wykorzystywane w niniejszych badaniach dwie aplikacje do pomiaru opóźnień toru fonicznego urządzeń mobilnych, tj. *SuperPowered Latency* i *Dr. Rick O'Rand Loopback* nie uwzględniają odpowiedzi toru w procesie pomiarowym. Wartość opóźnienia można pomierzyć, jeśli istnieje pętla pomiędzy wejściem i wyjściem. Sygnał otrzymany przez wejście z wyjścia toru jest wzmacniany i ponownie przetwarzany w pętli. W większości takich przypadków przybiera formę głośnego pisku bądź świergotu. Dzieje się tak ze względu na powstanie pętli sprzężenia zwrotnego składającego się kolejno z wzmacniacza, głośnika, pomieszczenia oraz mikrofonu. [6].

Istnieje możliwość pomiaru jedynie wartości opóźnień wejścia i wyjścia w torze fonicznym, korzystając z testu/efektu Larsena. Można zmierzyć wartość opóźnień w torze i podzielić tę wielkość przez dwa, otrzymując wartość opóźnień toru fonicznego w jednym kierunku. Typowo pomiar ten jest wykonywany poprzez podanie sygnału impulsowego w pętli wejścia-wyjścia, pozwalając na stworzenie pętli zwrotnej. Wynikiem pomiaru jest wartość opóźnienia całego toru fonicznego.

Pomiary mogą być wykonane na dwa sposoby, a mianowicie poprzez specjalny łącznik, który sparuje wejście i wyjście audio urządzenia. Jeżeli nie jest przewidziane użycie takiego urządzenia, to sygnał odtwarzany jest przez wyjście urządzenia (czyli głośnik) i następnie dociera do wejścia urządzenia, mikrofonu. W tym przypadku wartość opóźnienia może być większa niż przy pomiarach za pomocą urządzenia łączącego wejście i wyjście. Związane jest to z propagacją dźwięku oraz umiejscowieniem głośnika w telefonie.

Ważnym aspektem takich pomiarów jest to, że urządzenie łączące musi mieć zapewnioną odpowiednią

impedancję, aby widziane było jako zestaw słuchawkowy dla urządzeń mobilnych. W innym przypadku urządzenie mobilne używa mikrofonu wbudowanego i zewnętrznych układów elektronicznych. Impedancja dla zestawów słuchawkowych zawierających mikrofon w urządzeniach mobilnych powinna wynosić co najmniej 5kOhm [6].

3.2. Wykorzystane aplikacje

Jak wspomniano wcześniej, w badaniach wykorzystano dwie aplikacje. Pierwszą aplikacją, jaka została użyta do pomiarów wartości opóźnień w torze audio, była *SuperPowered Latency* [7]. Jak zauważają autorzy tej aplikacji, pozwala ona na pomiar najniższej możliwej wartości na danym urządzeniu. Korzysta ona z efektu Larsena do pomiaru latencji w torze fonicznym. Nie jest wymagane użycie urządzenia łączącego, jednakże jego użycie jest możliwe. Aplikacja *SuperPowered Latency* jest wieloplatformowa, w związku z tym może być również zainstalowana na urządzeniach mobilnych pracujących na systemie IOS. Dodatkowo istnieje możliwość współdzielenia danych poprzez Internet. Aplikacja działa z każdą wersją systemu Android powyżej wersji 3.0.

Wykorzystując dane dostarczone przez firmę, przeprowadzono dodatkowe testy na urządzeniach widniejących w bazie, aby porównać wyniki twórców *SuperPowered Latency* oraz z wynikami opracowanymi na potrzeby tego artykułu.

Drugą aplikacją użytą był *Dr. Rick O'Rand Loopback*. Jest to darmowa aplikacja mierząca wartości opóźnień, dostępna w oficjalnym sklepie Google Play. Podobnie, jak aplikacja *SuperPowered Latency* nie potrzebuje do pomiaru wartości opóźnień urządzenia łączącego, aczkolwiek autor aplikacji sugeruje użycie specjalnego czteropięściennego wejścia audio typu jack wraz z układem rezystorów oraz kondensatorów, aby móc zasymulować impedancję zestawu słuchawkowego dla urządzeń mobilnych. Aplikacja ta również korzysta z efektu Larsena do zmierzenia wartości opóźnień. Użytkownik ma możliwość określenia częstotliwości próbkowania oraz wybrać wielkość buforów. W trakcie badań zastosowano domyślne parametry aplikacji [8].

W pomiarach wykorzystano różne urządzenia. Połączono wejścia oraz wyjścia urządzeń przez interfejs I-Rig oraz kabel jack-jack audio. Następnie pomierzono wartość opóźnień. Wyniki pomiarów podane są w kolejnym rozdziale.

4. WYNIKI

W Tabeli 1, na podstawie pomiarów dziesięciu urządzeń z różnymi wersjami systemu Android, zaprezentowano wartości opóźnień dla każdego z urządzeń.

Do pomiarów wykorzystano pięć różnych wersji Android:

- KitKat (4.4.4);
- Lollipop (5.1, 5.1.1);
- Marshmallow (6.0.0, 6.0.1);
- Nugat (7.0, 7.0.1);
- Oreo (8.0.0).

Tabela 1. Wyniki pomiarów opóźnień w torze Audio dla różnych urządzeń i innych wersji systemu Android

	Android version	Dr. Rick O'Rand App [ms]	Super-Powered Latency - test [ms]	Super-Powered Latency - wyniki z bazy danych [ms]
Xiaomi Mi A1	7.0.1	45.65	43	36
Xiaomi Redmi Note 3 Pro Prime	6.0.1	46.19	41	41
Meizu m2 Note	5.1	278.76	346	-
OnePlus 5T	8.0.0	15.58	83	69
Xiaomi Mi5	7.0	32.76	36	27
Xiaomi Mi5S	7.0	8.06	24	32
Chuwi Hi8 Pro	4.4.4	9.67	320	301
Huawei P8 Lite	6.0.0	8.06	275	245
Samsung Galaxy Tab 2	5.1.1	427.54	307	251
Samsung Galaxy J3	5.1.1	261.57	306	306

W zależności od użytych do testów urządzeń oraz aplikacji, wyniki mogą znacząco różnić się od siebie. W pomiarze tabletu marki Chuwi aplikacją *SuperPowered Latency* wartość opóźnienia wyniosła 320 ms, podczas gdy aplikacja *Dr. Rick O'Rand* wskazała wartość 9,67 ms. Podobne wyniki zaobserwowano dla smartfonu Huawei P8 Lite, gdzie pierwsza aplikacja wskazała wartość opóźnienia 275 ms, a aplikacja *Dr. Rick O'Rand* wskazała 8,06 ms. Najniższą wartość opóźnienia w torze fonicznym uzyskał smartfon Xiaomi Mi5S z zainstalowanym systemem Android 7.0 (Tabela 1.).

Podczas porównania pomiędzy różnymi systemami Android nie może być jasno określone, co wpływa na zwiększenie wydajności toru fonicznego pod względem zmniejszenia wartości opóźnienia. Największą wartością opóźnień cechuje się system w wersji 6.0 Lollipop, występują w nim co najmniej dziesięciokrotnie większe opóźnienia w stosunku do wcześniejszych, jak i późniejszych wersji.

5. PODSUMOWANIE

Wyniki pomiarów pokazują znaczące różnice w wartości opóźnień w różnych urządzeniach bazujących na platformie Android. Powoduje to brak możliwości stworzenia uniwersalnych aplikacji, przetwarzających sygnały foniczne w czasie rzeczywistym, gdyż wymagają one wartości opóźnienia maksymalnie rzędu 10 ms. Testowane aplikacje do pomiarów wartości opóźnień posługują się różnymi metodami przetwarzania sygnału audio. Kolejnym problemem takich pomiarów jest dokładność pomiarów poprzez wybrane aplikacje. Zaobserwowana rozbieżność w wynikach może być skutkiem problemów z wymuszeniem przez aplikację zadanej wartości bufora audio. W skrajnych przypadkach może to prowadzić do ustawienia minimalnych wartości bufora, a co za tym idzie znacznego zmniejszenia opóźnień. W przypadku normalnego użytkowania toru fonicznego w systemie Android będzie to prowadziło do trzasków oraz niezadowolającej jakości przetwarzanego sygnału. Rozwiązaniem tego może być stworzenie aplikacji, która wykorzysta znaczniki czasu zapisane równolegle w sygnale fonicznym przetwarzanym w torze, od wejścia po wyjście urządzenia. Takie rozwiązanie pozwoliłoby nie tylko pomierzyć wartość opóźnienia, ale także jego zmienność.

Wraz z rozwojem nowych wersji systemu Android należy brać pod uwagę opóźnienie w standardzie MIDI oraz porównać je z przetwarzanym przez port USB sygnałem dźwiękowym. Badania takie zostały przeprowadzone, aczkolwiek nie dotyczą tego tematu [9],[10].

6. BIBLIOGRAFIA

1. Zigunovs, D., Smirnova, J., Vitols, G., and Stonys, G., "Solution for Sound Playback Delay on Android Devices," *Procedia Computer Science*, 104 (December 2016), pp. 413–420, 2016, ISSN 18770509, doi:10.1016/j.procs.2017.01.154.
2. SuperPowered, "Android Audio's 10 Millisecond Problem: The Android Audio Path Latency Explainer," data dostępu: 2018-03-10, <http://superpowered.com/androidaudiopathlatency>.
3. SuperPowered, "Rebooting Android's 10 Millisecond Problem: Audio Latency Improvements in Android 6.0 Marshmallow," data dostępu: 2018-03-10, <http://superpowered.com/android-marshmallowlatency>.
4. SuperPowered, "Android Audio Latency Problem Just Got Worse," data dostępu: 2018-03-10, <http://superpowered.com/android-audio-latencyproblem-just-got-worse>.
5. SuperPowered, "Audio Latency Measurements," data dostępu: 2018-03-10, https://source.android.com/devices/audio/latency_measurements.
6. SuperPowered, "Audio Loopback Dongle," data dostępu: 2018-03-10, <https://source.android.com/devices/audio/loopback>.
7. SuperPowered, "SuperPowered Latency," data dostępu: 2018-03-10, <https://github.com/superpoweredSDK/SuperpoweredLatency>.
8. Glenn, K., "Dr. Rick O'Rang Loopback - GooglePlay," data dostępu: 2018-03-10, <https://play.google.com/store/apps/details?id=org.drrickorang.loopback>.
9. Pan, X., Wilson, J., Balukoff, M., Liu, A., and Xu, W., "Musical Instruments Simulation on Mobile Platform," pp. 1–8, 2016, DOI: 10.2352/ISSN.2470-1173.2016.7MOBMU-300.
10. Michon, R., Smith, J.O, Wright, M., Chafe, C., Granzow, J., and Wang, G., "Mobile Music, Sensors, Physical Modeling, and Digital Fabrication: Articulating the Augmented Mobile Instrument", *Appl. Sci.* 2017, 7, 1311; doi:10.3390/app7121311.
11. Michon, R., Smith, J.O, Wright, M., Chafe, C., Granzow, J., and Wang, G., "Mobile Music, Sensors, Physical Modeling, and Digital Fabrication: Articulating the Augmented Mobile Instrument", *Appl. Sci.* 2017, 7, 1311; doi:10.3390/app7121311.

MEASUREMENT OF LATENCY IN THE ANDROID AUDIO PATH

This paper provides a description of experimental investigations concerning comparison between the audio path characteristics of various Android versions. First, information about the changes in each system version in the context of latency caused by them is presented. Then, a measurement procedure employing available applications to measure latency is described comparing to results contained in the Internet. Finally, a comparison between tested systems and results of tests are presented along with conclusions on possible audio processing implementations on the Android platform.

Keywords: android, latency, measurement, audio engineering.