



**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE**

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Advertising vs Extended Advertising in BLE

Autorzy: Bartłomiej Kręgielewski, Szymon Lepianka, Małgorzata Pinior
Kierunek studiów: Informatyka
Opiekun projektu: dr Robert Lubaś

Kraków, 2020r.

Spis Treści

Spis Treści	2
Wstęp	4
Opis pracy	4
Potencjalne korzyści	4
Opis literatury	5
Czym jest Bluetooth?	5
Bluetooth Advertising	5
Bluetooth Low Energy - BLE	5
Advertising vs Extended Advertising	6
Technologie i narzędzia	7
Technologie	7
Narzędzia	7
Zastosowane rozwiązania	8
Diagram	8
Metody poszczególnych klas	8
Device	8
LegacyDevice	8
ExtendedDevice	8
World	9
Channel	9
Parametry obiektów	9
Device	9
World	9
Channel	9
Opis algorytmu	10
Legacy Advertising	10
Funkcja główna symulacji:	10
Wątek urządzenia rozgłaszającego:	10
Wątek urządzenia słuchającego:	10
Extended Advertising	11
Funkcja główna symulacji:	11
Wątek urządzenia rozgłaszającego:	11
Wątek urządzenia słuchającego:	11
Obsługa konfliktów “w powietrzu”	12
Wyniki symulacji	13

Scenariusz nr 1 - 2 urządzenia Legacy (1 rozgłaszające, 1 słuchające)	13
Scenariusz nr 2 - 2 urządzenia Extended (1 rozgłaszające, 1 słuchające)	14
Scenariusz nr 3 - 4 urządzenia Legacy (2 rozgłaszające, 2 słuchające, z losowymi opóźnieniami)	15
Scenariusz nr 4 - 4 urządzenia Legacy (2 rozgłaszające, 2 słuchające, bez losowych opóźnień)	16
Scenariusz nr 5 - 4 urządzenia Extended (2 rozgłaszające, 2 słuchające, z losowymi opóźnieniami)	18
Scenariusz nr 6 - 4 urządzenia Extended (2 rozgłaszające, 2 słuchające, bez losowych opóźnień)	19
Scenariusz nr 7 - 10 urządzeń Legacy (5 rozgłaszające, 5 słuchające, z losowymi opóźnieniami)	21
Scenariusz nr 8 - 10 urządzeń Extended (5 rozgłaszające, 5 słuchające, z losowymi opóźnieniami)	22
Podsumowanie	24
Legacy vs Legacy	24
Extended vs Extended	25
Legacy vs Extended	26
Wnioski	28

Wstęp

Opis pracy

Celem projektu jest zamodelowanie rozgłaszania się urządzeń Bluetooth Low Energy, symulacja oraz pomiar wydajności rozgłaszania tzw. 'legacy' z extended advertising. Jest to zagadnienie bardzo praktyczne, sumiennie wykonanie pozwoli na zdobycie dużej i przydatnej wiedzy o technologii Bluetooth.

Potencjalne korzyści

Zmniejszenie problemów spowodowanych dużą ilością urządzeń nadających na raz, poprzez zwiększenia ilości kanałów advertising i sprytne nimi zarządzanie.
Zwiększenie prędkości, poprzez zwiększenie długości pojedynczego pakietu do 255 bajtów, jak i zmniejszenie ilości liczby konfliktów w powietrzu powstających w wyniku advertisingu.

Opis literatury

Przystępując do realizacji projektu, postanowiliśmy najpierw zgłębić wiedzę o tematyce technologii Bluetooth, aby odpowiednio merytorycznie podejść do rozwiązania problemu. Użyliśmy w tym celu specyfikacji Bluetooth Core Specification v5.2.

Czym jest Bluetooth?

Bluetooth to standard w jakim urządzenia mogą "porozumiewać" (wymieniać informacje) bezprzewodowo, używając fali krótkich (short-length) czyli od 2.400 do 2.485 GHz ($\lambda \rightarrow 124.91 - 120.64$ mm), czyli w [ISM Band](#). Prace rozpoczęto w 1989 roku, oraz zaprezentowano publicznie po raz pierwszy w 1999 roku. Za rozwój technologii odpowiada w głównej mierze [Bluetooth Special Interest Group \(SIG\)](#), których standardy muszą zostać spełnione aby producent mógł powiedzieć, że jego urządzenie obsługuje Bluetooth. Przeciętny zasięg w Bluetooth class 2 to od 15 do 40 metrów.

Bluetooth Advertising

Jest to metoda porozumiewania się między urządzeniami (wysyłania i odbierania informacji) używając technologii Bluetooth. Metoda bazuje na zgodach, czyli dany odbiorca ma wybór czy chce przyjąć daną wiadomość czy nie (odbiorca musi zasygnalizować, że chce odbierać dany advertising).

Advertising process:

- a) Urządzenie nadające nadaje mały pakiet z informacjami dla urządzenia skanującego
- b) Urządzenie skanujące odbiera dany pakiet i decyduje co z nim dalej zrobić

Urządzenia mogą prowadzić proces reklamowania się na jednym z 3 dostępnych kanałów, podczas kiedy pakiety z danymi mogą zostać nadane dynamicznie na jednym z 37 innych kanałów.

Jak często następuje advertising? Dopuszczalne wartości to od 20 milisekund do 10 sekund (dodawane są do ustawionych wartości losowe opóźnienia, aby uniknąć kolizji).

Bluetooth Low Energy - BLE

BLE (kiedyś nazywane Bluetooth Smart) to technologia mająca na celu zminimalizowanie zużycia energii przez urządzenie korzystające z Bluetooth, przy zachowaniu zbliżonego zasięgu. Większość dzisiejszych urządzeń mobilnych (telefony z Androidem, iOS, laptopy itp.) domyślnie wspierają tą właśnie technologię.

BLE zostało wprowadzone w Bluetooth 4.0 co za tym idzie nie wszystkie urządzenia je wspierają. Kompatybilność wsteczna (backward compatibility) występuje w standardzie Bluetooth jednak należy pamiętać o tym, że urządzenia starsze mogą używać jedynie funkcji, które znają. Wyjątkiem od tego jest połączenie między urządzeniem z BLE, które nie implementuje Classic. Dlatego para nowych słuchawek nie połączy się z starym telefonem posiadającym Bluetooth 3.0. Sytuacja odwrotna, to jest nowy telefon implementujący Bluetooth 4.0 chcący się połączyć z starymi słuchawkami 3.0 powinien bez problemu

działać, jednak wtedy nie będzie korzystał z Low Energy. (Grafika oraz powyższy wywód na podstawie: [Link](#))

Classic vs Low Energy Bluetooth



Specifications	Bluetooth 1.1	Bluetooth 1.2	Bluetooth 2.0	Bluetooth 2.1 plus EDR (Enhanced Data Rate)	Bluetooth 3.0	Bluetooth 4.0
Voice dialing	Yes	Yes	Yes	Yes	Yes	Yes
Call mute	Yes	Yes	Yes	Yes	Yes	Yes
Last-number redial	Yes	Yes	Yes	Yes	Yes	Yes
Improved resistance to radio frequency interference		Yes	Yes	Yes	Yes	Yes
10-meter range	Yes	Yes	Yes	Yes	Yes	Yes
100-meter range			Yes	Yes	Yes	Yes

Advertising vs Extended Advertising

Bluetooth 5 wprowadził funkcję, która pozwala urządzeniom transmitować znacznie więcej danych niż poprzednie wersje Bluetooth (nawet 8-krotnie). Ta nowa funkcja nosi nazwę “Advertising Extensions” lub “Extended Advertisements”. We wcześniejszych wersjach dane były ograniczone do 31 bajtów, a dzięki nowym rozszerzonym pojemność jest zwiększana do 255 bajtów. Extended Advertising działa w ten sposób, że urządzenie wysyła “primary advertisements”, które zawierają informacje o tym, jak zlokalizować w czasie i częstotliwości “the secondary advertisements”, zwane “extended advertisements”. One zaś wysyłane są za pośrednictwem wspomnianych wcześniej kanałów “the secondary advertisement”, które są takie same jak kanały danych używane podczas połączenia.

Technologie i narzędzia

Technologie

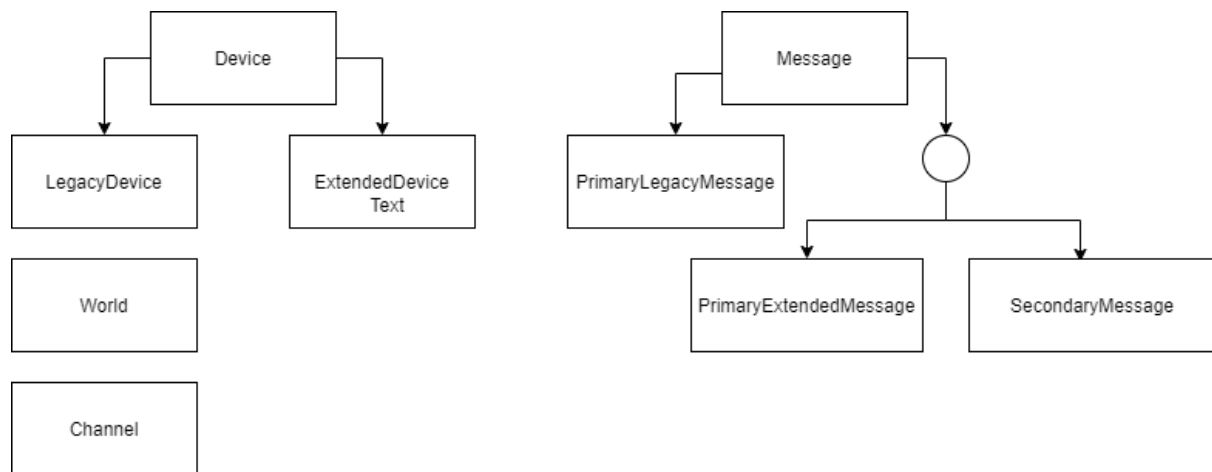
Wykorzystaliśmy do symulacji język programowania java, który dostarcza wiele przydatnych bibliotek (Instant, ByteBuffer). Taki dobór technologii pozwala na bezproblemowe zrozumienie kodu symulacji w celu ewentualnych, przyszłych samodzielnych modyfikacji. Kod symulacji jest upubliczniony i gotowy do uruchomienia. Naszą ramą jest świat, w którym mamy zbiór kanałów. Na nie nadawane są wiadomości przez urządzenia. Każde urządzenie ma swój wątek, ile wątków tyle urządzeń nadających. Urządzenia posiadają możliwe stany (enum Mode {WAIT, SCAN, LISTEN, SECONDARY, ADVERTISE, FINISHED}).

Narzędzia

Stworzenie symulacji wymagany był dobór odpowiednich narzędzi. Jako zespół podjęliśmy decyzję, żeby użyć bezpłatną wersję oprogramowania IntelliJ IDEA 2020.1.1 dostarczanego przez firmę JetBrains. Kod symulacji był bezpośrednio publikowany w serwisie GitHub. Jest on jednym z najbardziej popularnych systemów kontroli wersji. Do komunikacji między członkami zespołu użyliśmy komunikatora Discord, który zapewnia łączność głosową, tekstową i wideo.

Zastosowane rozwiązania

Diagram



Metody poszczególnych klas

Device

- void generateContent() - generuje losowy kontent do przesłania
- boolean isMessageNew(Message receivedMessage) - sprawdza czy komunikat nie jest duplikatem już odebranego komunikatu
- void scanningCheck() - sprawdza czy urządzenie nie nasłuchuje zbyt długo nie dostając żadnego komunikatu

LegacyDevice

- void scan() - urządzenie skanuje kanały primary w poszukiwaniu komunikatów
- void advertise() - urządzenie nadaje na kanałach primary swoje komunikaty

ExtendedDevice

- void scan() - urządzenie skanuje kanały primary w poszukiwaniu komunikatów
- void secondaryListen() - urządzenie czeka na kanale secondary na komunikaty
- void advertise() - urządzenie nadaje na kanałach primary swoje komunikaty (dotyczące tego, kiedy i gdzie oczekiwać na secondary)
- void secondaryListen() - urządzenie nadaje komunikat secondary

World

- `World getInstance()` - świat jest singletonem, metoda pozwala dostać referencję do obiektu

Channel

- `void setPayload(Message payload, long TTLinMS)` - pozwala na wstawienie komunikatu na kanał, oraz jego czasu życia, sprawdza czy wystąpił konflikt w powietrzu, jeżeli tak to ustawia odpowiednią flagę
- `boolean isEmpty()` - sprawdza czy kanał jest pusty
- `Message getPayload()` - zwraca zawartość umieszczoną na kanale
- `void clearPayload()` - czyści kanał

Parametry obiektów

Device

- `int deviceId;`
- `byte[] data;`
- `boolean randomDelay = true;`
- `ArrayList<Pair<Message, Instant>> receivedMessages;`
- `int advertiseCounter = 0;`
- `int advertiseFor = 3;`
- `long advertiseBreak;`
- `Integer deviceToListenTo;`
- `Instant scanningSince;`

World

- `int numberOfChannels;`
- `Channel[] channels;`

Channel

- `int id;`
- `World world = World.getInstance();`
- `private boolean empty = true;`
- `Message payload;`
- `Instant payloadArrivalTime;`
- `Instant fullUntil = Instant.now();`
- `boolean inAirConflict = false;`

Opis algorytmu

Legacy Advertising

Urządzenia posiadają możliwe stany: WAIT, SCAN, ADVERTISE, FINISHED.

Funkcja główna symulacji:

1. Tworzymy urządzenia rozgłaszające (nadajemy stan ADVERTISE) i słuchające (stan SCAN).
2. Dla każdego urządzenia rozgłaszającego generujemy całą wiadomość (kontent) jaką będziemy chcieli przesłać.
3. Generujemy pierwszą paczkę. Paczka ma 31 bajtów i składa się z messageID (4 bajty), senderID (4 bajty), content (22 bajty). Ostatnia paczka posiada znacznik lastMessage (1 bajt), który mówi że dana paczka jest już ostatnia.
4. Startujemy wątki.

Wątek urządzenia rozgłaszającego:

5. Urządzenie rozgłaszające „wstawia” paczkę na kanał 37, 38 lub 39 (losowo). Na tym kanale paczka utrzyma się przez obliczoną wartość: $\frac{\text{rozmiar paczki}}{1\text{Mb/s}}$ (średnio około 30 nanosekund). Na tyle też czasu urządzenie zatrzymujemy urządzenie funkcją sleep().
6. Procedura wstawiania dla danej paczki powtarzana jest jeszcze 2 razy. Za każdym razem wybór kanału jest losowy spośród tych, na których dana paczka jeszcze nie była rozgłaszana. Przykładowa kolejność: 38, 37, 39.
7. Urządzenie jest zatrzymywane na czas podany przez użytkownika (z zakresu 20ms - 10s). Tą wartością można sterować - częstość advertisingu. Jeśli jeszcze pozostała wiadomość do wysłania, generujemy następną paczkę i wracamy do punktu 1.
8. Po wysłaniu ostatniej paczki urządzenie zmienia stan na FINISHED i wątek kończy działanie.

Wątek urządzenia słuchającego:

9. Urządzenie skanujące od momentu wystartowania wątku skanuje kanały 37, 38 i 39 oraz sprawdza, czy coś się na nim znajduje.
10. Jeśli trafi na niepusty kanał porównuje messageID i senderID z dotychczas zapamiętanymi wiadomościami, żeby nie zapamiętywał kolejny raz tej samej wiadomości.
11. Jeśli wiadomość jest „nowa” zapamiętuje ją i wraca do skanowania kanałów.
12. Jeśli zapamiętana wiadomość posiada znacznik lastMessage = true, urządzenia przechodzi w stan FINISHED i wątek kończy działanie.

Extended Advertising

Urządzenia posiadają dodatkowo stany: LISTEN, SECONDARY.

Funkcja główna symulacji:

1. Tworzymy urządzenia rozgłaszające (nadajemy stan ADVERTISE) i słuchające (stan SCAN).
2. Dla każdego urządzenia rozgłaszającego generujemy całą wiadomość (kontent) jaką będziemy chcieli przesłać.
3. Generujemy pierwszą paczkę (Primary). Paczka w obecnym stanie ma 21 bajtów i składa się z messageID (4 bajty), senderID (4 bajty), channel (kanał na którym będzie się odbywał Secondary Adv., 4 bajty), time (czas - obiekt Instant, kiedy będzie się odbywał Secondary Adv., 8 bajtów). Startujemy wątki.

Wątek urządzenia rozgłaszającego:

1. Urządzenie rozgłaszające wykonuje Primary Advertising w dokładnie ten sam sposób, jak w przypadku Legacy (również 3 razy).
2. Po rozgłoszeniu Primary generujemy paczkę typu Secondary (rozmiar to 255 bajtów). Zawiera: messageID (4 bajty), senderID (4 bajty), content (247 bajtów). Ostatnia paczka posiada znacznik lastMessage (1 bajt), który mówi że dana paczka jest już ostatnia.
3. Urządzenie zmienia stan na SECONDARY.
4. Urządzenie „wstawia” paczkę secondary na kanał wskazany w paczce primary. Na tym kanale paczka utrzyma się przez obliczoną wartość: (średnio około 243 nanosekund). Na tyle też czasu urządzenie zatrzymujemy urządzenie funkcją sleep().
5. Urządzenie jest zatrzymywane na czas podany przez użytkownika (z zakresu 20ms - 10s). Tą wartością można sterować - częstość advertisingu. Jeśli jeszcze pozostała wiadomość do wysłania, generujemy paczkę typu Primary i wracamy do punktu 1.
6. Po wysłaniu ostatniej paczki urządzenie zmienia stan na FINISHED i wątek kończy działanie.

Wątek urządzenia słuchającego:

1. Urządzenie skanujące początkowo zachowuje się jak w przypadku Legacy.
2. Jeśli wiadomość jest „nowa” zapamiętuje ją i zmienia stan na LISTEN.
3. W stanie LISTEN urządzenie nasłuchuje na wskazanym kanale we wskazanym czasie.
4. Jeśli odebrana wiadomość nie jest ostatnia urządzenie zmienia stan na SCAN. W przeciwnym przypadku urządzenie przechodzi w stan FINISHED i wątek kończy działanie.

Obsługa konfliktów “w powietrzu”

Przypadek konfliktu w powietrzu pojawia się przypadku, w którym dwa lub więcej urządzeń próbują rozgłosić swoją paczkę na tym samym kanale (paczka urządzenia pierwszego jeszcze się znajduje na danym kanale w momencie, w którym urządzenie drugie zaczyna rozgłaszać). W rzeczywistości paczki nawzajem się zagłuszają i urządzenia słuchające otrzymują uszkodzone wiadomości. W zastosowanym algorytmie, przy takiej sytuacji, kanał, na którym wystąpi konflikt w powietrzu przyjmuje flagę `inAirConflict = true` do czasu, w którym jedno z urządzeń “zwolni” kanał.

```
synchronized void setPayload(Message payload, long TTLinNS) {
    if(this.fullUntil.isBefore(Instant.now())) {
        this.payload = payload;
        this.inAirConflict = false;
    } else {
        if(this.id >= world.numberOfChannels-3)Result.getInstance().increasePrimary();
        else Result.getInstance().increaseSecondary();
        this.inAirConflict = true;
        this.payload = null;
    }
    this.empty = false;
    this.payloadArrivalTime = Instant.now();
    this.fullUntil = this.payloadArrivalTime.plusNanos(TTLinNS);
}
```

W przypadku, w którym zostanie zagłuszona paczka posiadająca `lastMessage = true` (ostatnia), urządzenie słuchające pozostaje w stanie SCAN/LISTEN przez 1 sekundę po czym zmienia stan na FINISH.

```
void scanningCheck() {
    if(this.scanningSince == null)this.scanningSince = Instant.now();
    else if(this.scanningSince.plusMillis(1000).isBefore(Instant.now())) {
        this.mode = Mode.FINISHED;
        //System.out.println("Scanning for nothing");
    }
}
```

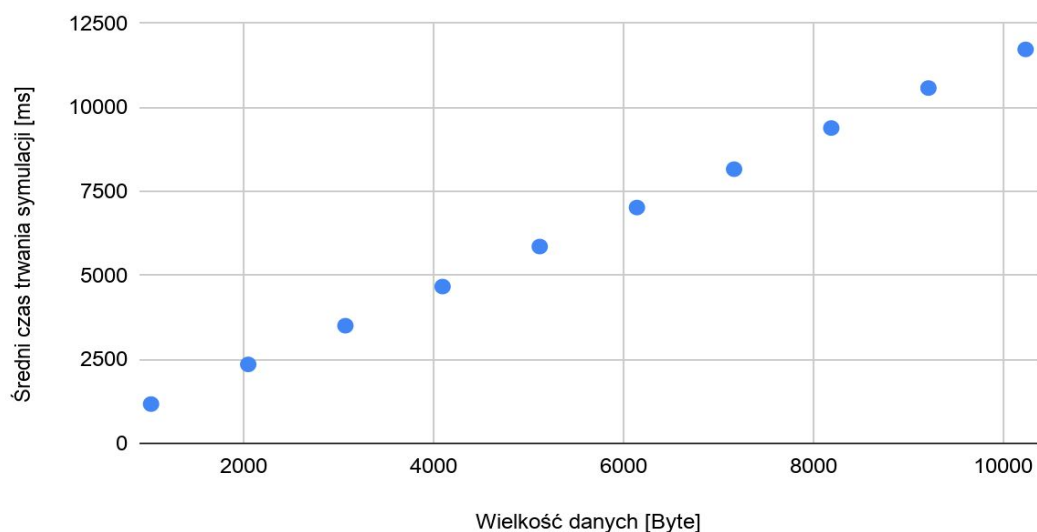
Wyniki symulacji

Scenariusz nr 1 - 2 urządzenia Legacy (1 rozgłaszające, 1 słuchające)

Tabela przedstawia dane wysłane i odebrane dla urządzeń Legacy. Zostało to przedstawione na wykresie niżej. Widać tutaj zależność liniową między wielkością danych a czasem trwania transmisji.

Wielkość danych na urządzenie [Byte]	Średnia ilość wszystkich odebranych danych [Byte]	Średni czas trwania symulacji [ms]	Średnia ilość konfliktów primary	Średnia ilość konfliktów secondary	Średni % odebranych danych
1024	1024	1170.8	0	0	100.00%
2048	2048	2351.6	0	0	100.00%
3072	3072	3503.6	0	0	100.00%
4096	4096	4668.9	0	0	100.00%
5120	5120	5859.7	0	0	100.00%
6144	6144	7020.7	0	0	100.00%
7168	7168	8162.5	0	0	100.00%
8192	8192	9388	0	0	100.00%
9216	9216	10577.8	0	0	100.00%
10240	10240	11728.3	0	0	100.00%

Wykres zależności czasu trwania symulacji od wielkości danych

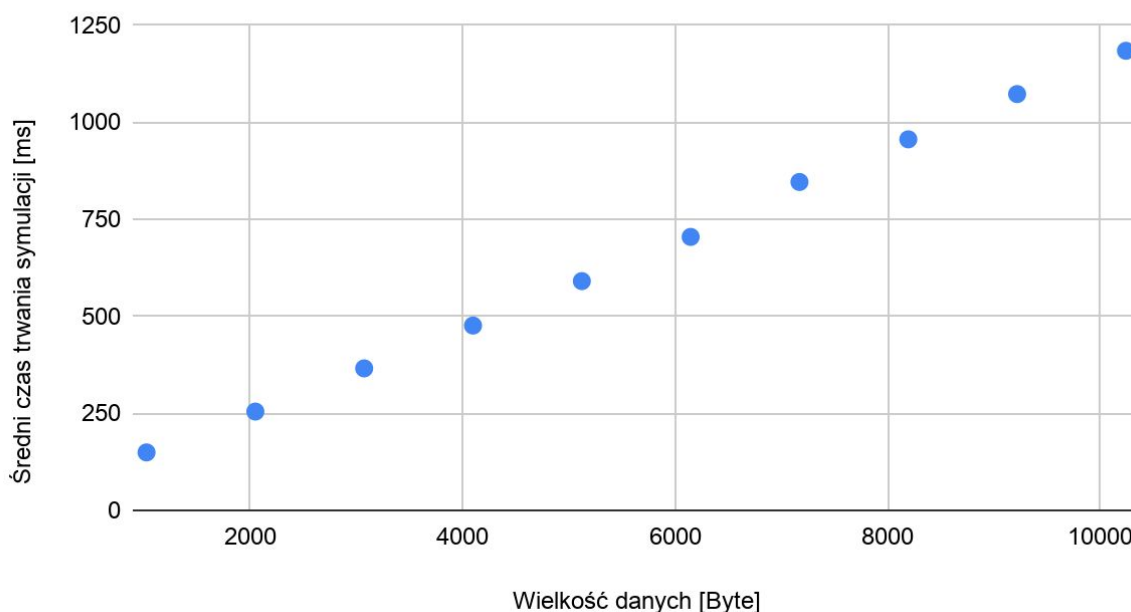


Scenariusz nr 2 - 2 urządzenia Extended (1 rozgłaszające, 1 słuchające)

Urządzenia Extended również mają zależność liniową między wielkością danych a czasem trwania transmisji, jednak w tym wypadku dla tego samego rozmiaru danych, transmisja trwała prawie 10 razy krócej.

Wielkość danych na urządzenie [Byte]	Średnia ilość odebranych danych [Byte]	Średni czas trwania symulacji [ms]	Średnia ilość konfliktów primary	Średnia ilość konfliktów secondary	Średni % odebranych danych
1024	1024	149.4	0	0	100.00%
2048	2048	254.7	0	0	100.00%
3072	3072	366	0	0	100.00%
4096	4096	476.4	0	0	100.00%
5120	5120	590.9	0	0	100.00%
6144	6144	704.8	0	0	100.00%
7168	7168	846.4	0	0	100.00%
8192	8192	956.3	0	0	100.00%
9216	9216	1072.9	0	0	100.00%
10240	10240	1184.5	0	0	100.00%

Wykres zależności czasu trwania symulacji od wielkości danych

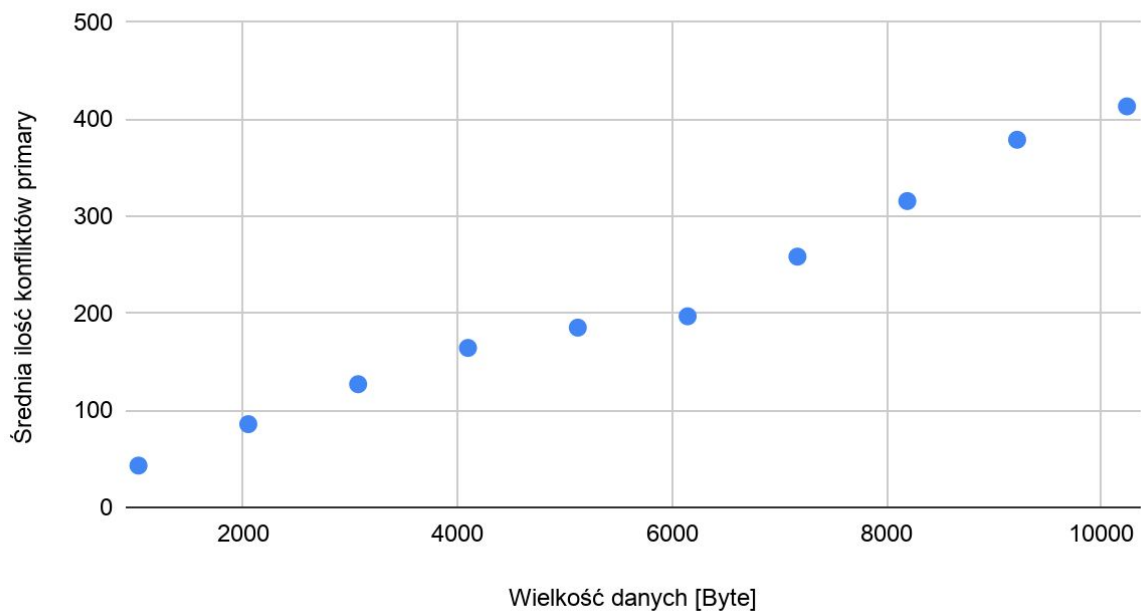


Scenariusz nr 3 - 4 urządzenia Legacy (2 rozgłaszające, 2 słuchające, z losowymi opóźnieniami)

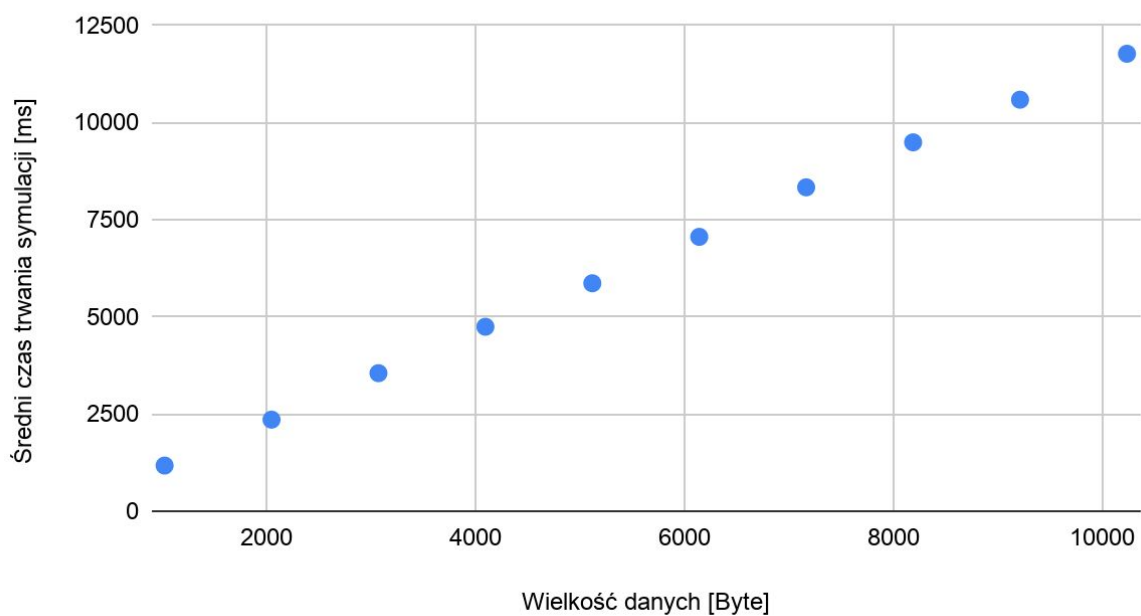
W tym scenariuszu dodano losowe opóźnienie. Można zauważyć, że czym większy rozmiar danych, tym większa liczba konfliktów, które zagłuszały sygnały. Skutkowało to brakami w odebranych danych.

Wielkość danych na urządzenie [Byte]	Średnia ilość wszystkich odebranych danych [Byte]	Średni czas trwania symulacji [ms]	Średnia ilość konfliktów primary	Średnia ilość konfliktów secondary	Średni % odebranych danych
1024	1962.9	1174.7	43.1	0	95.84%
2048	3941.9	2356.4	85.8	0	96.24%
3072	5923.2	3552.3	127	0	96.41%
4096	7904.5	4747	164.4	0	96.49%
5120	9883.5	5866.4	185.3	0	96.52%
6144	11917.7	7061	197	0	96.99%
7168	13771	8335.7	258.5	0	96.06%
8192	15668.7	9493.5	315.8	0	95.63%
9216	17514.3	10591	379	0	95.02%
10240	19417.4	11773.4	413.4	0	94.81%

Wykres zależności ilości konfliktów primary od wielkości danych



Wykres zależności czasu trwania symulacji od wielkości danych

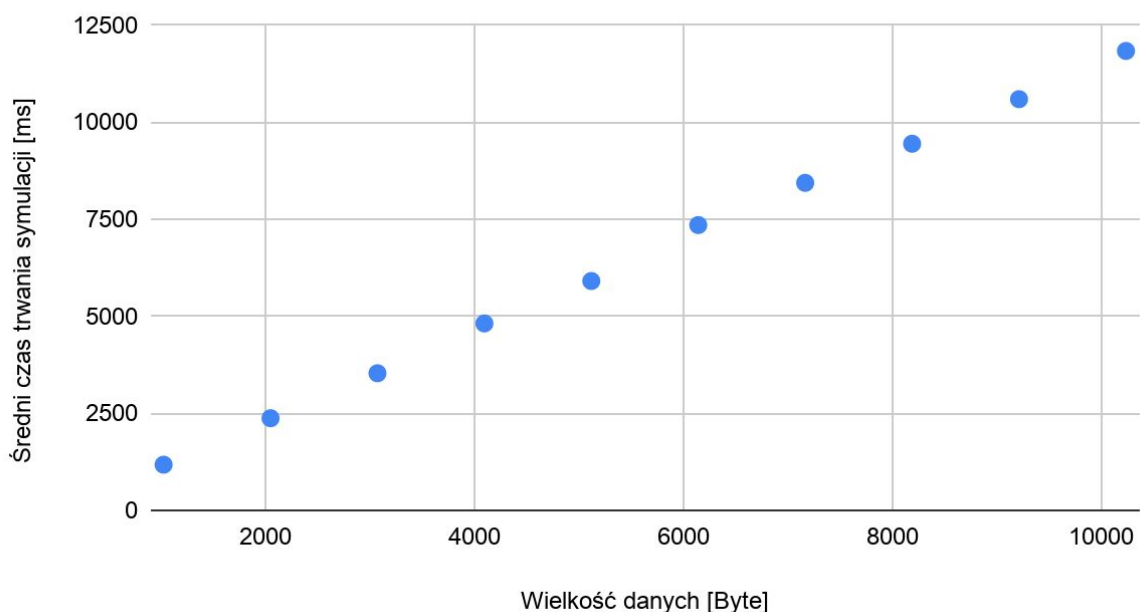


Scenariusz nr 4 - 4 urządzenia Legacy (2 rozgłaszające, 2 słuchające, bez losowych opóźnień)

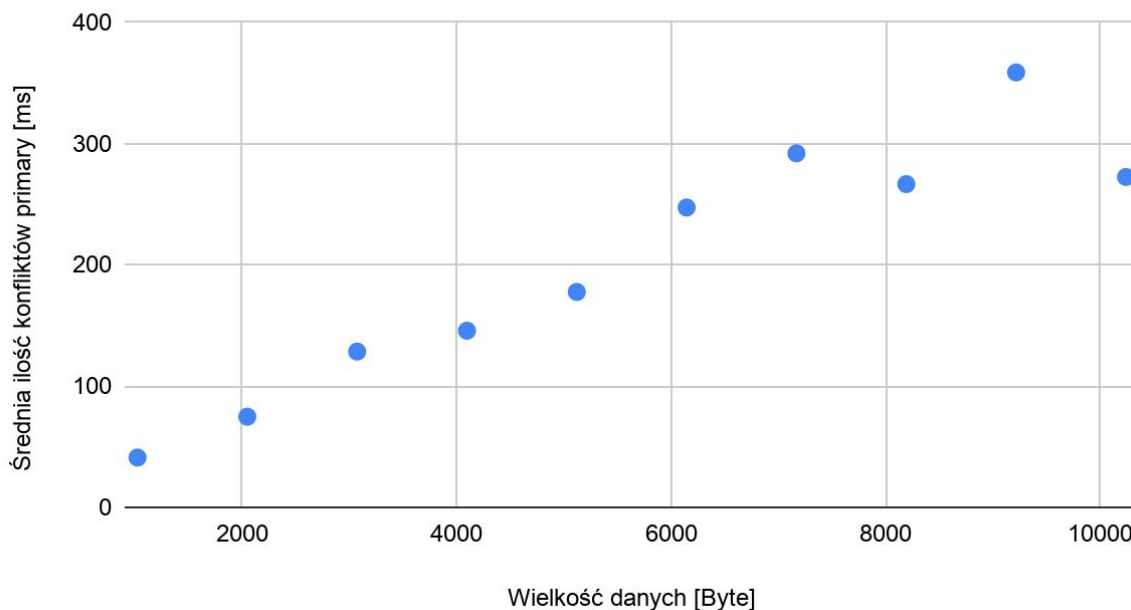
Scenariusz nie obejmował losowych opóźnień, tutaj również zachowała się liniowa zależność czasu od danych. Dodanie losowych opóźnień w niewielkim stopniu wpłynęło na ilość konfliktów, za to można zaobserwować, że pomimo rosnącej liczby konfliktów, procent odebranych danych wciąż pozostał taki sam (zagłuszenia na poziomie 4%).

Wielkość danych na urządzenie [Byte]	Średnia ilość wszystkich odebranych danych [Byte]	Średni czas trwania symulacji [ms]	Średnia ilość konfliktów primary	Średnia ilość konfliktów secondary	Średni % odebranych danych
1024	1979	1177.4	41.1	0	96.63%
2048	3960.3	2375.3	74.8	0	96.69%
3072	5925.5	3534.1	128.5	0	96.44%
4096	7931.9	4818.2	145.7	0	96.82%
5120	9867.4	5911.3	177.7	0	96.36%
6144	11797.2	7353.4	247.3	0	96.01%
7168	13684.4	8441.2	292	0	95.45%
8192	15772.2	9449.2	266.6	0	96.27%
9216	17643.1	10597.9	358.7	0	95.72%
10240	19863.6	11840.6	272.5	0	96.99%

Wykres zależności czasu trwania symulacji od wielkości danych



Wykres zależności ilości konfliktów primary od wielkości danych

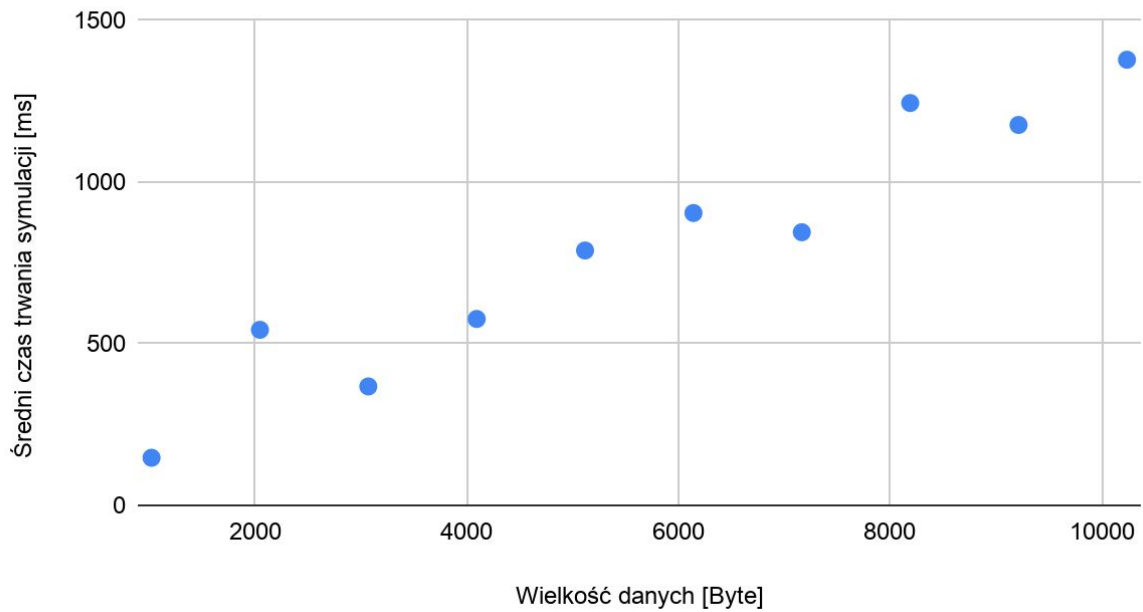


Scenariusz nr 5 - 4 urządzenia Extended (2 rozgłaszające, 2 słuchające, z losowymi opóźnieniami)

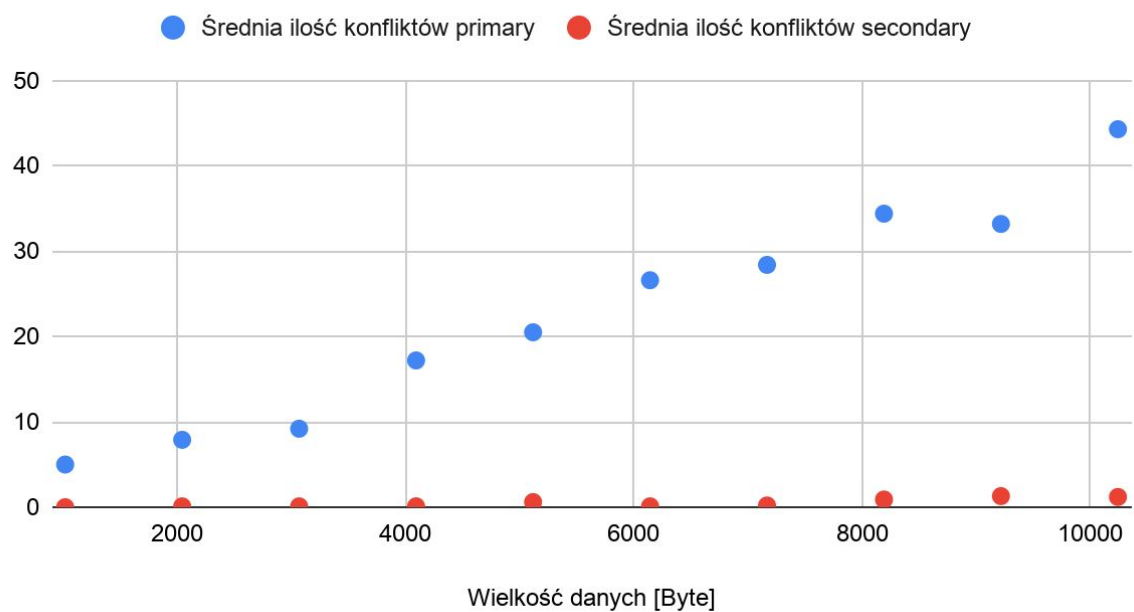
W wypadku urządzeń Extended możemy zauważyć pewne odstępstwa od liniowej zależności czasu od wielkości danych (najprawdopodobniej spowodowane losowością - okres jednej sekundy oczekiwania w przypadku zagłuszenia ostatniego komunikatu - co ma większą szansę zdarzyć się w przypadku Extended Advertising). Można natomiast zauważyć, że średnio nastąpiło mniej konfliktów na kanale primary w porównaniu do Legacy.

Wielkość danych na urządzenie [Byte]	Średnia ilość wszystkich odebranych danych [Byte]	Średni czas trwania symulacji [ms]	Średnia ilość konfliktów primary	Średnia ilość konfliktów secondary	Średni % odebranych danych
1024	1924.5	147.3	5	0	93.97%
2048	3950.9	543	7.9	0.1	96.46%
3072	5921.7	367.9	9.2	0.1	96.38%
4096	7930.6	576.6	17.2	0.1	96.81%
5120	9660.6	788.3	20.5	0.6	94.34%
6144	11775.5	904.2	26.6	0.1	95.83%
7168	13718.5	844.8	28.4	0.2	95.69%
8192	15556.6	1244.4	34.4	0.9	94.95%
9216	17700.3	1176.6	33.2	1.3	96.03%
10240	19345.9	1378.3	44.3	1.2	94.46%

Wykres zależności czasu trwania symulacji od wielkości danych



Wykres zależności ilości konfliktów od wielkości danych

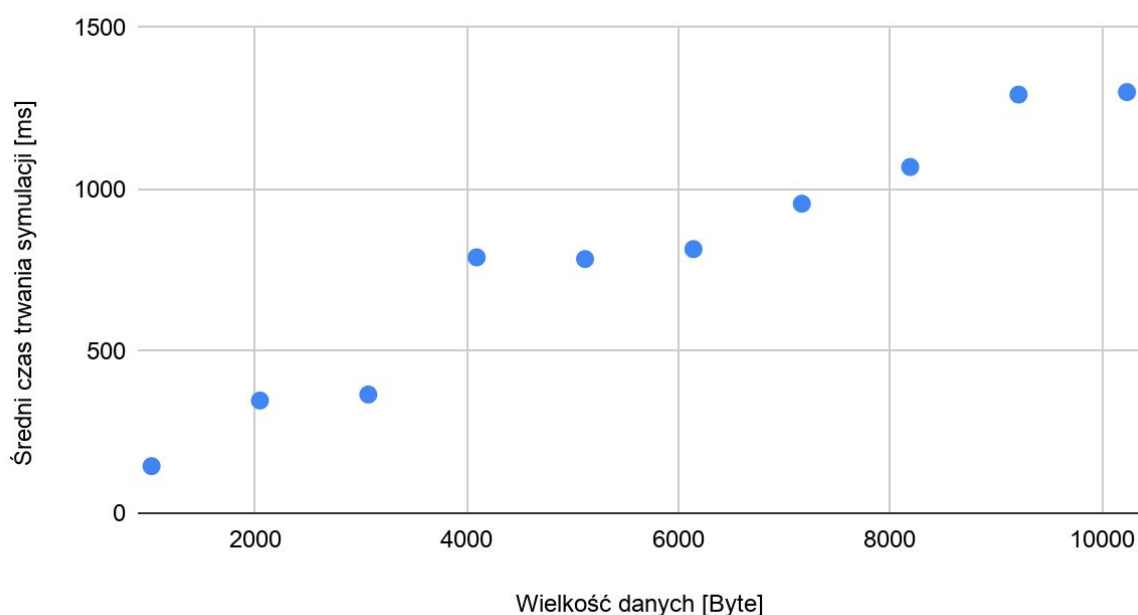


Scenariusz nr 6 - 4 urządzenia Extended (2 rozgłaszające, 2 słuchające, bez losowych opóźnień)

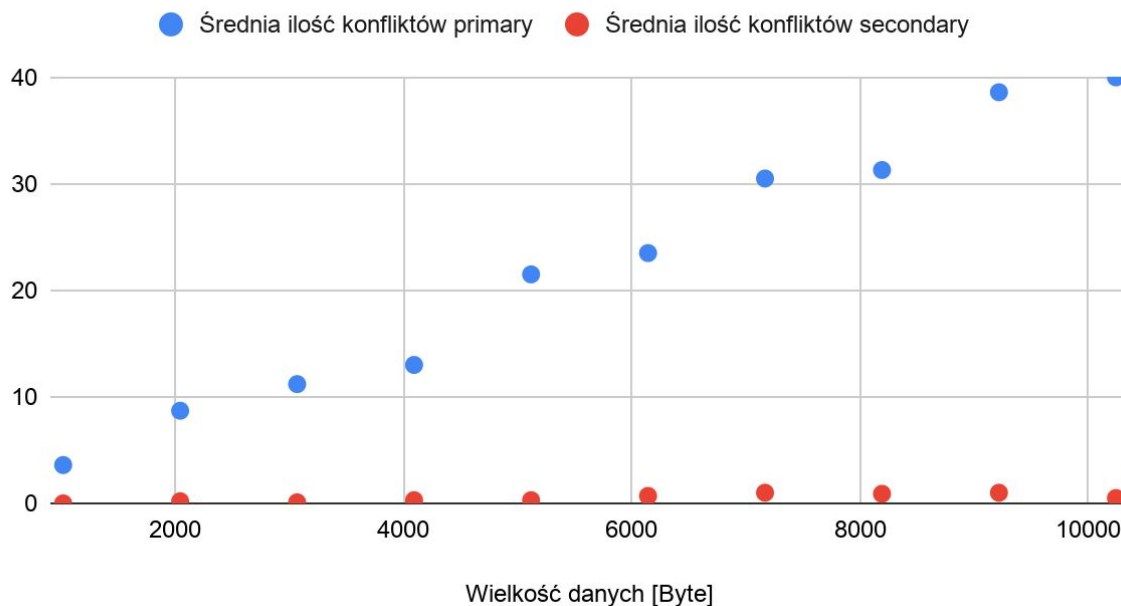
W tym scenariuszu również utraciła się liniowa zależność czasu od wielkości danych, jednakże widać spadek ilości odebranych danych. Liczba konfliktów na kanałach primary jest nieznaczaco różniaca się od wersji z losowymi opóźnieniami.

Wielkość danych na urządzenie [Byte]	Średnia ilość wszystkich odebranych danych [Byte]	Średni czas trwania symulacji [ms]	Średnia ilość konfliktów primary	Średnia ilość konfliktów secondary	Średni % odebranych danych
1024	1973.9	145.3	3.6	0	96.38%
2048	3841.8	348	8.7	0.2	93.79%
3072	5921.7	366.5	11.2	0.1	96.38%
4096	7827.7	790.2	13	0.3	95.55%
5120	9685.3	785	21.5	0.3	94.58%
6144	11673.6	815.6	23.5	0.7	95.00%
7168	13471	955.8	30.5	1	93.97%
8192	15515.4	1069.3	31.3	0.9	94.70%
9216	17453.3	1292.9	38.6	1	94.69%
10240	19456	1300.6	40	0.5	95.00%

Wykres zależności czasu trwania symulacji od wielkości danych



Wykres zależności ilości konfliktów od wielkości danych

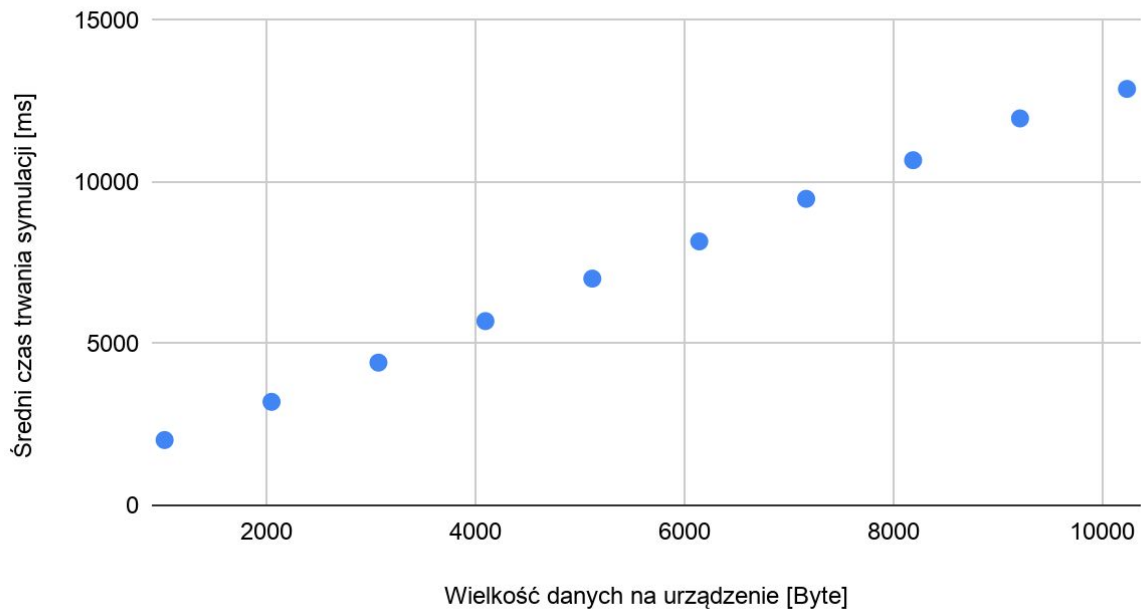


Scenariusz nr 7 - 10 urządzeń Legacy (5 rozgłaszające, 5 słuchające, z losowymi opóźnieniami)

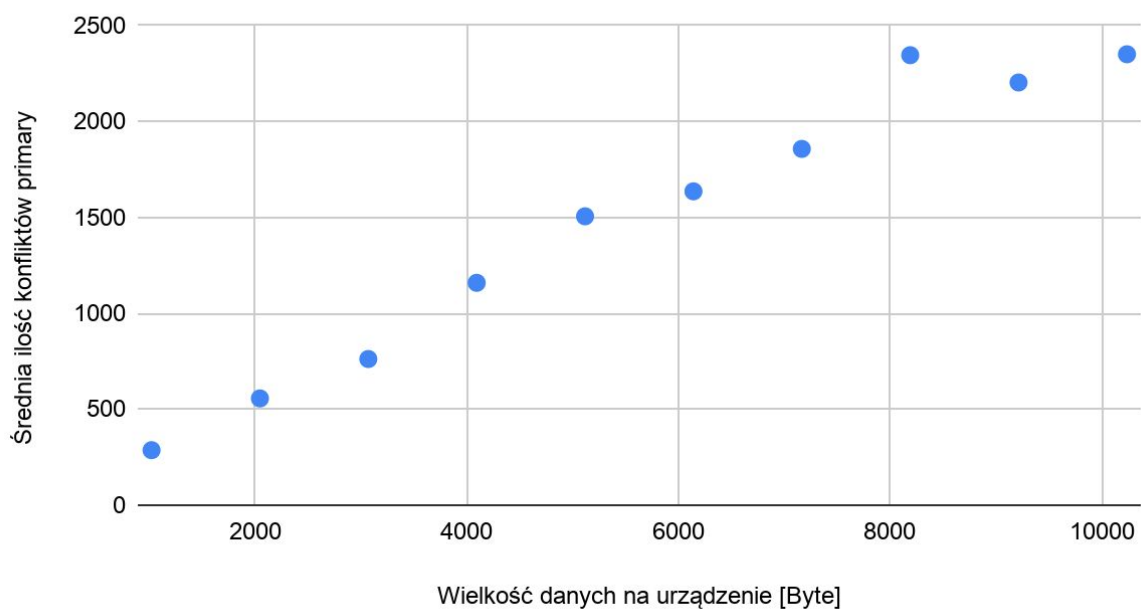
Przy 10 urządzeniach Legacy zachowała się liniowa zależność czasu od wielkości danych, jednakże procent odebranych danych spadł o około 14%. Wzrosła również liczba konfliktów w porównaniu z 4 urządzeniami (zdecydowanie więcej urządzeń będzie chciało nadawać na tych samych kanałach jednocześnie).

Wielkość danych na urządzenie [Byte]	Średnia ilość wszystkich odebranych danych [Byte]	Średni czas trwania symulacji [ms]	Średnia ilość konfliktów primary	Średnia ilość konfliktów secondary	Średni % odebranych danych
1024	4054	2021.2	286.9	0	79.18%
2048	8211	3202.8	556.6	0	80.19%
3072	12677.2	4415.9	761.9	0	82.53%
4096	16198.4	5699.8	1159.1	0	79.09%
5120	19671.8	7013.6	1505.8	0	76.84%
6144	24467.6	8164	1635.9	0	79.65%
7168	28662	9480.6	1857.1	0	79.97%
8192	31359.8	10675.3	2345.6	0	76.56%
9216	37341.8	11968	2203.3	0	81.04%
10240	41913.2	12880.3	2350.4	0	81.86%

Wykres zależności czasu trwania symulacji od wielkości danych



Wykres zależności ilości konfliktów od wielkości danych

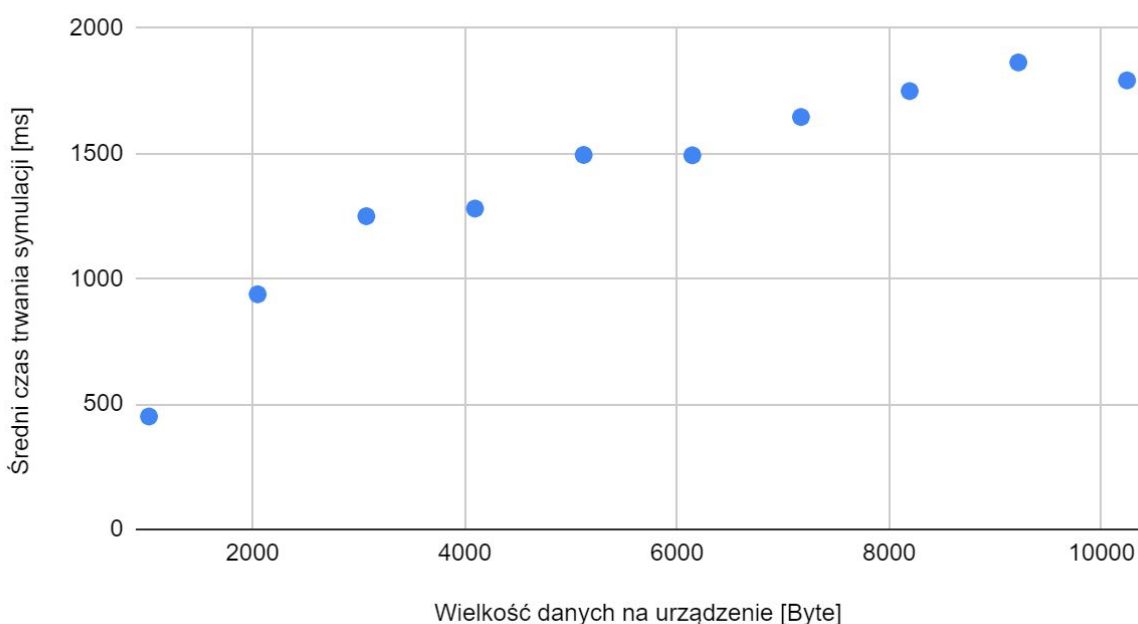


Scenariusz nr 8 - 10 urządzeń Extended (5 rozgłaszające, 5 słuchające, z losowymi opóźnieniami)

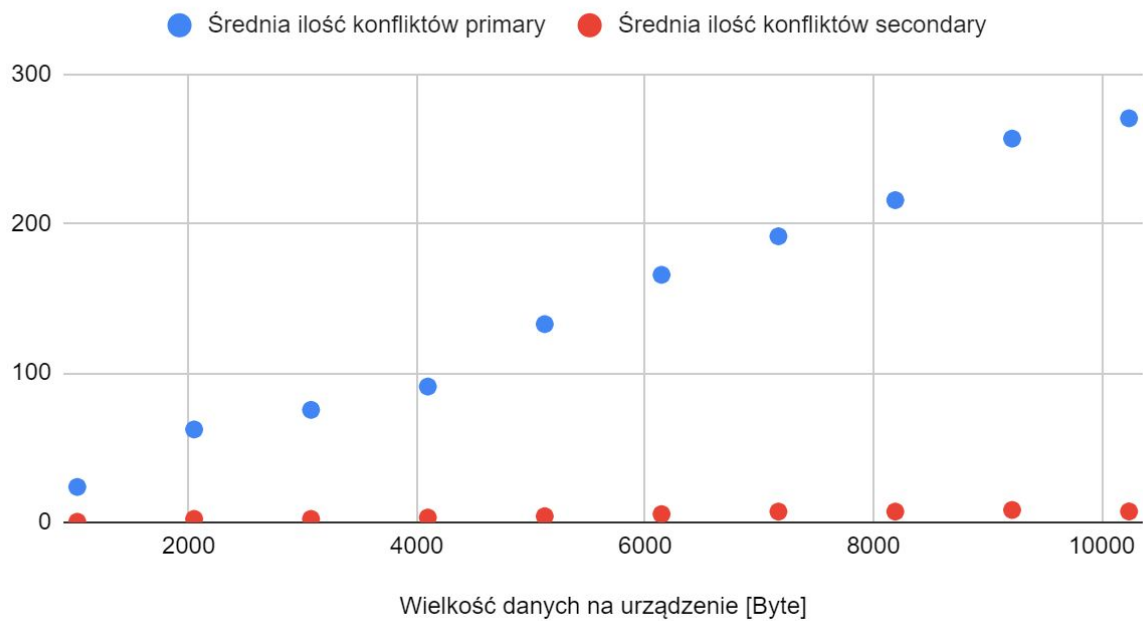
Tak jak w scenariuszu z czterema urządzeniami Extended została utracona zależność liniowa czasu od wielkości danych. Podobnie jak w poprzednich podobnych przypadkach widać tendencję liniową, ale średnie dane są obciążone błędami. Natomiast widocznie spadł procent odebranych danych (o około 16%) wraz ze wzrostem ilości urządzeń. Widać również, że liczba konfliktów rośnie wraz z wielkością danych.

Wielkość danych na urządzenie [Byte]	Średnia ilość wszystkich odebranych danych [Byte]	Średni czas trwania symulacji [ms]	Średnia ilość konfliktów primary	Średnia ilość konfliktów secondary	Średni % odebranych danych
1024	3804.2	450.5	23.7	0.4	74.30%
2048	6791	938.3	62.2	2.3	66.32%
3072	11539.8	1249.9	75.3	2.3	75.13%
4096	16178.6	1280.4	90.9	3.2	79.00%
5120	18839.4	1494.4	132.7	4.1	73.59%
6144	22487.4	1493	165.7	5.5	73.20%
7168	26493.6	1645.9	191.5	7.2	73.92%
8192	30416.2	1749.3	215.7	7.2	74.26%
9216	33697.8	1863.6	256.9	8.3	73.13%
10240	37940.4	1791.9	270.5	7.3	74.10%

Wykres zależności czasu trwania symulacji od wielkości danych



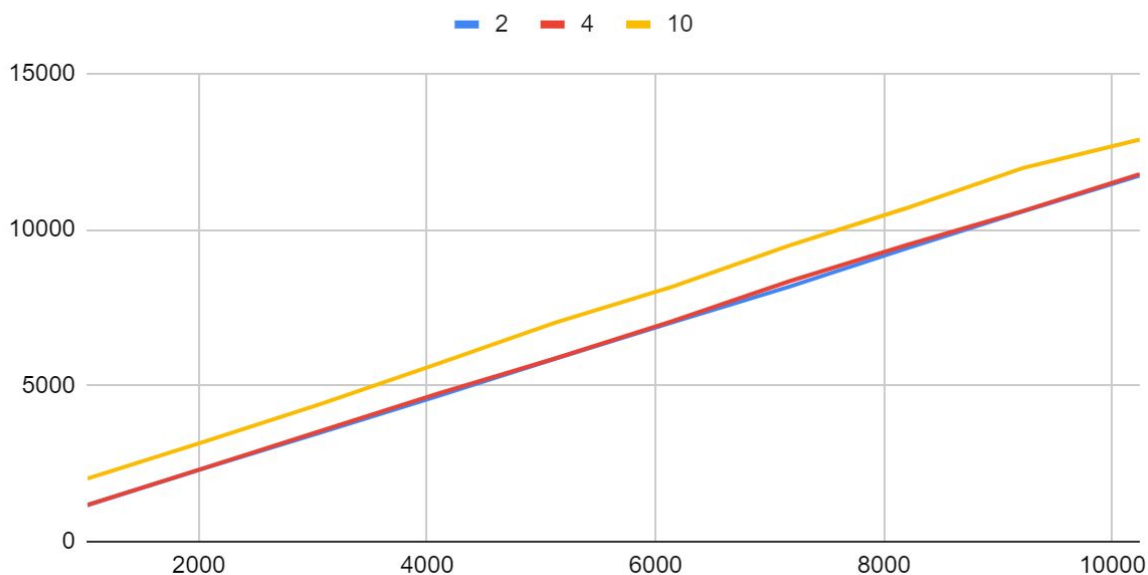
Wykres zależności ilości konfliktów od wielkości danych



Podsumowanie

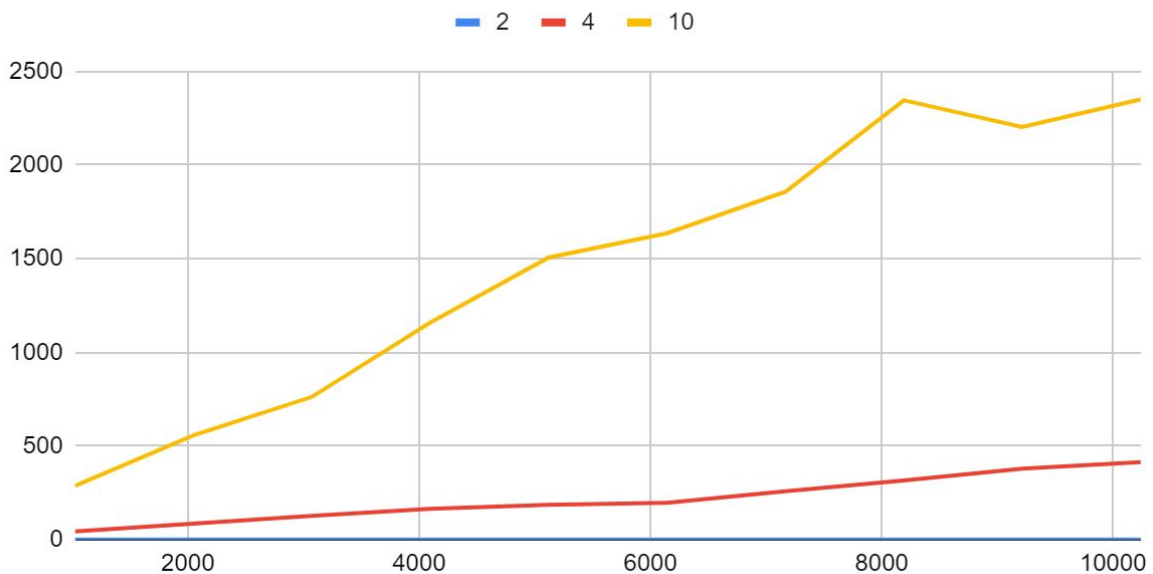
Legacy vs Legacy

Wykres zależności czasu symulacji [ms] od wielkości danych [Byte] (dla różnej liczby urządzeń)



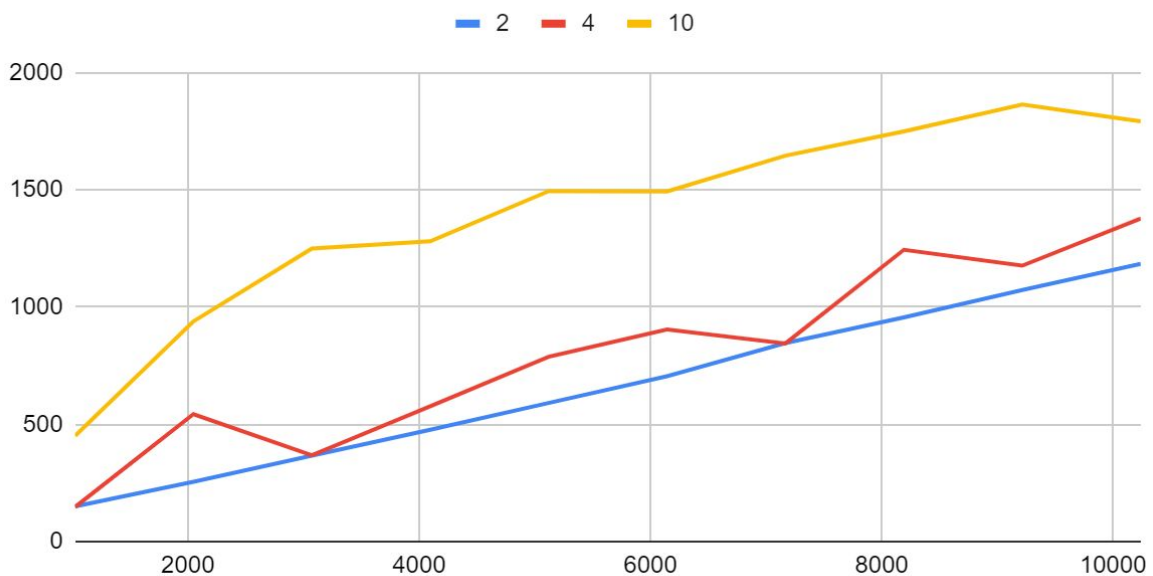
Jak widzimy na powyższym wykresie czas symulacji dla 10 urządzeń nieznacznie się wydłużył. Analizując algorytm dochodzimy do wniosków, że taka rzecz nie powinna mieć miejsca, jako że jedynym elementem mogącym wywołać taki efekt jest jedno sekundowy czas oczekiwania w przypadku zagłuszenia ostatniego komunikatu (co może się zdarzyć częściej gdy mamy więcej urządzeń) lub może to być wpływ maszyny, na której symulacja jest uruchomiona (podczas próby uruchomienia jej na innym (wolniejszym) komputerze czas dla 10 urządzeń był zauważalnie większy).

Wykres zależności ilości konfliktów primary w zależności od wielkości danych [Byte] (dla różnej liczby urządzeń)



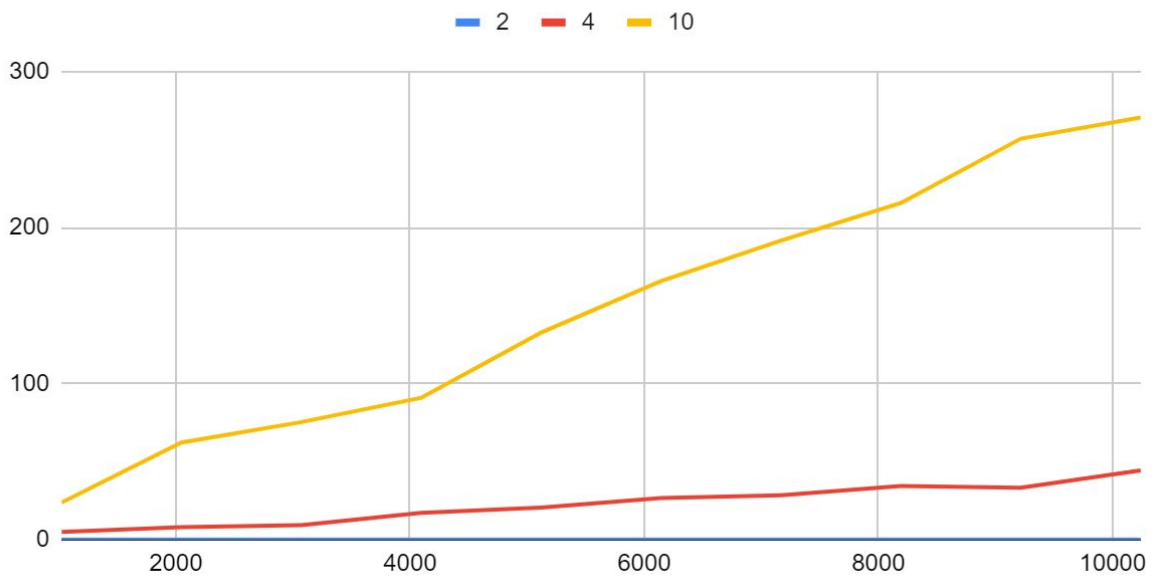
Extended vs Extended

Wykres zależności czasu symulacji [ms] od wielkości danych [Byte] (dla różnej liczby urządzeń)

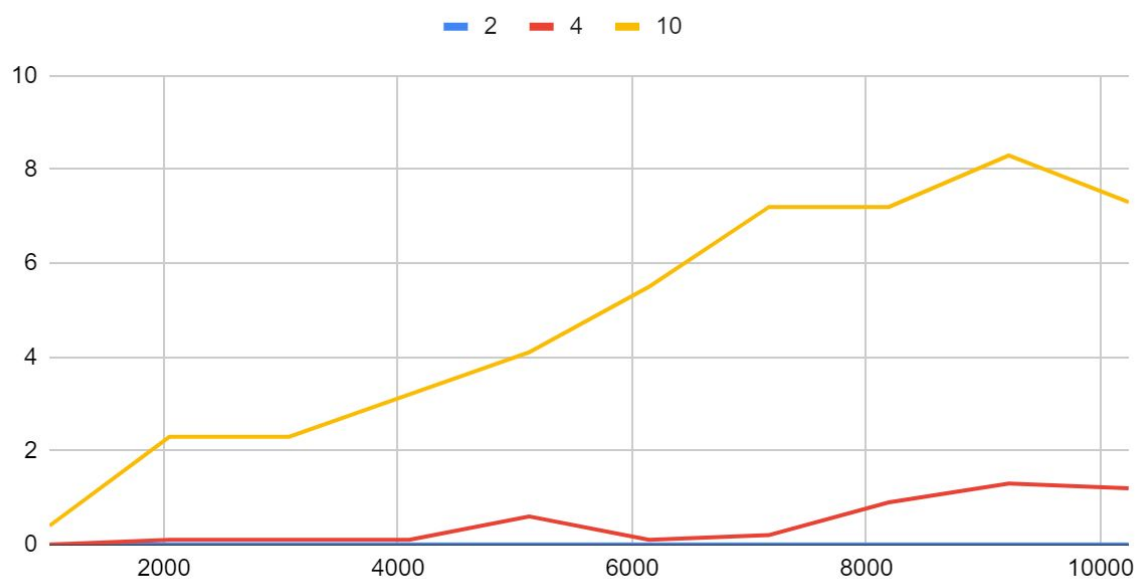


Tutaj również możemy zaobserwować wzrost czasu symulacji wraz ze wzrostem ilości urządzeń, co zapewne jest spowodowane powodami opisanymi w [Legacy vs Legacy](#).

Wykres zależności ilości konfliktów primary od wielkości danych [Byte] (dla różnej liczby urządzeń)



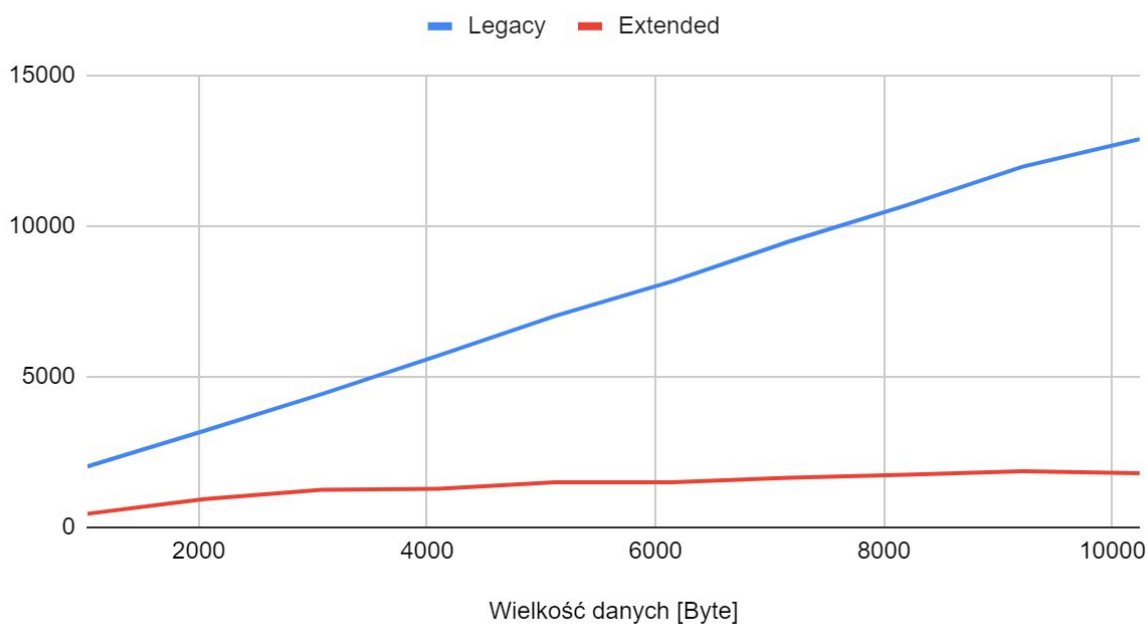
Wykres zależności ilości konfliktów secondary od wielkości danych [Byte] (dla różnej liczby urządzeń)



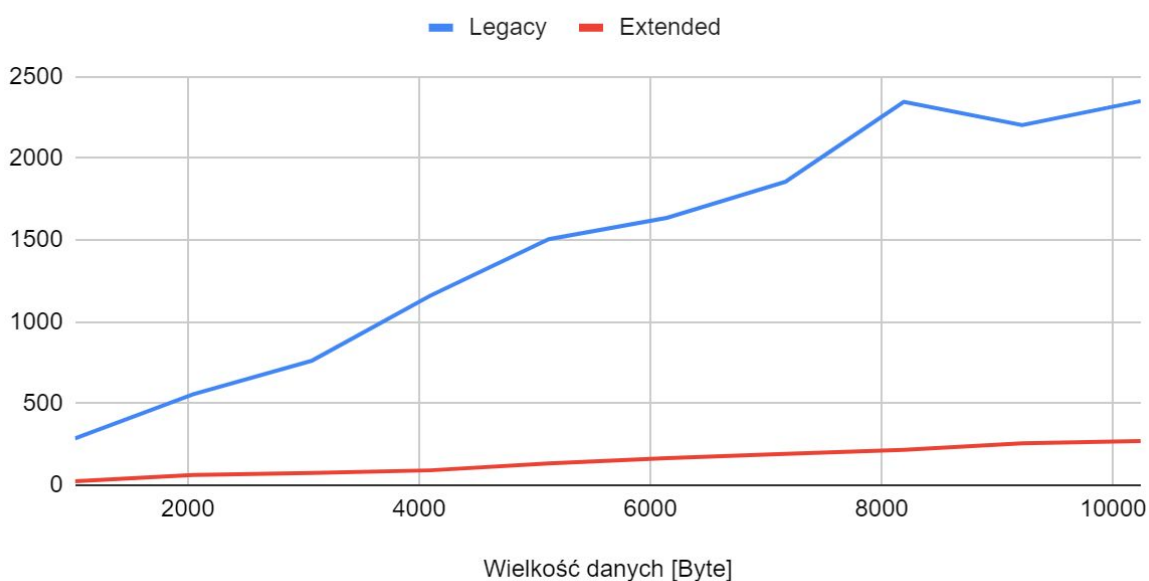
Legacy vs Extended

- oba z opóźnieniem
- ilość urządzeń: 10

Wykres zależności czasu symulacji [ms] od wielkości danych [Byte]



Wykres zależności ilości konfliktów primary od wielkości danych [Byte]



Wnioski

Porównując czasy symulacji od razu rzuca się w oczy zdecydowana przewaga Extended Advertising. Było to do przewidzenia ze względu przede wszystkim na to, że jest to następnik Legacy. Różnica w czasie zwiększa się wraz ze wzrostem przesyłanych danych, jak i ze wzrostem urządzeń biorących udział w symulacji. Ilość konfliktów przy tych samych parametrach jest mniejsza w symulacji typu Extended - jest ich około 10 razy mniej. Dodatkowo ilość tych konfliktów rośnie liniowo w zależności od wielkości danych, z dużo mniejszym współczynnikiem niż w przypadku Legacy.

Spowodowane jest między innymi znacznym odciążeniem kanałów primary (37, 38, 39) i wysyłaniem paczek na kanałach secondary. Aczkolwiek procent danych odebranych w przypadku symulacji z większą ilością urządzeń jest widocznie większy dla Legacy (okolice 80%+) niż dla Extended (70%+). Mimo to, dużą wadą jest ogromna różnica w czasie. Dla największej symulowanej ilości danych wysłanych wynosi około 13 sekund.