

# 1 Teoria

## 1.1 Free-space path loss

Utrata w sile sygnału spowodowana przejściem fali elektromagnetycznej przez ośrodek (najczęściej powietrze). Wzór na obliczanie FSPL:

$$FSPL = P_{tx} + AG_{tx} + AG_{rx} - P_{rx} - FM - L \quad (1)$$

Gdzie symbole oznaczają:

- $P_{tx}$  - siła trasmitera, wyrażona w dBm
- $AG_{tx}$  - zysk energentyczny anteny trasmitera, wyrażony w dBi
- $AG_{rx}$  - zysk energentyczny anteny odbiorcy, wyrażony w dBi
- $P_{rx}$  - siła odbiornika, wyrażona w dBm
- $FM$  - margines zaniku sygnału (fade margin)
- $L$  - straty wynikające np z oddziaływania innych trasmiterów, przeszkód itp.

Dodatkowo, FSPL można obliczyć, używając następujący wzór:

$$FSPL = 20\log_{10}\left(\frac{d}{d_0}\right) + 20\log_{10}(f) + K \quad (2)$$

Gdzie symbole oznaczają:

- $d$  - dystans dzielący trasmiter od odbiorcy, wyrażony w metrach
- $d_0$  - dystans referencyjny - w tym wypadku 1 metr
- $f$  - częstotliwość trasmitera - wyrażona w MHz
- $K$  - stała, którą można określić wzorem:

$$K = 20\log_{10}\left(\frac{4\pi d_0}{C}\right) \quad (3)$$

gdzie  $d_0$  to dystans referencyjny (taki sam jak we wzorze wyżej), a  $C$  to długość fali emitowanej przez trasmiter

Po przekształceniu wzoru, użytkujemy:

$$d = 10^{\left(\frac{FSPL - K - 20\log_{10}(f)}{20}\right)} \quad (4)$$

A po połączeniu obu wzorów dostajemy:

$$d = 10^{\left(\frac{P_{tx} + AG_{tx} + AG_{rx} - P_{rx} - FM - L - K - 20\log_{10}(f)}{20}\right)} \quad (5)$$

## 1.2 Zysk energetyczny anteny

Zysk energetyczny anteny jest to stosunek mocy ateny wypromieniowanej w danym kierunku do mocy wypromieniowanej przez antenę wzorcową. Anteną wzorcową może być m.in. antena izotropowa, czyli antena bez fizycznych rozmiarów, która cały sygnał zasilany wysyła we wszystkich kierunkach. W takim wypadku, zysk energetyczny anteny wyrażany jest w  $dB_i$ .

Na zysk energetyczny mają również wpływ kierunkowość oraz materiał, z którego wykonana jest antena.

## 1.3 Received signal strength indication

Received signal strength indication (skrótom RSSI) jest to miara określająca moc sygnału odbieranego. Przyjmuje ona wartości niedodatnie (gdzie 0 oznacza sygnał najsilniejszy). Jednostką, w jakiej określa się siłę sygnału jest  $dBm$ , która jest logarytmiczną jednostką miary mocy odniesiona do mocy  $1mW$ .

System Android pozwala na odczytanie siły odbieranego sygnału. Można do tego wykorzystać API *WifiManager* (w przypadku odczytu sygnału WiFi) oraz *BroadcastReceiver* (w przypadku odczytu sygnału Bluetooth).

# 2 Bluetooth

# 3 Eksperymenty

## 3.1 Wykorzystane urządzenia

1. Smartphone Sony Xperia Z1 Compact (D5503) - odbiornik

Dane techniczne:

- Częstotliwość - 2,4GHz
- Przyrost siły sygnału z anteny - 2dBi

2. Router TP-Link TD-W8970 - nadajnik

Dane techniczne:

- Częstotliwość - 2,4GHz
- Dwie zewnętrzne anteny kierunkowe
- Przyrost siły sygnału z anteny - 4dBi
- Siła transmitera - 16.5dBm

3. Router TP-Link TL-WA701ND - nadajnik

Dane techniczne:

- Częstotliwość - 2,4GHz
- Jedna zewnętrzna antena kierunkowa
- Przyrost siły sygnału z anteny - 2dBi

- Siła transmitera - 15dBm
4. Smartphone Grand 2 (G7102) - nadajnik  
Dane techniczne:
- Częstotliwość - 2,4GHz
  - Jedna antena wbudowana
  - Przyrost siły sygnału z anteny - 0dBi
  - Siła transmitera - 10dBm

### 3.2 Warunki

Wszystkie pomiary wykonywane były w pomieszczeniu zamknięty, bez przeszkód na drodze sygnału, dlatego jako margines zaniku sygnału została przyjęta wartość 22 dBm. Inne straty (np interferencja sygnałów z routerów) zostały pominięte i ich wykrycie jest jednym z celów eksperymentu.

### 3.3 Cele

Celem eksperymentu jest ustalenie, jak zmierzona i obliczona, przy użyciu siły sygnałów, odległość między odbiornikiem i transponderami odnosi się do odległości rzeczywistej. Dodatkowo, będę się starał ustalić, jak duży wpływ na jakość sygnału mają przeszkody, kierunek, w jakim skierowane są względem siebie urządzenia oraz interferencja sygnałów.

### 3.4 Pomiar odległości

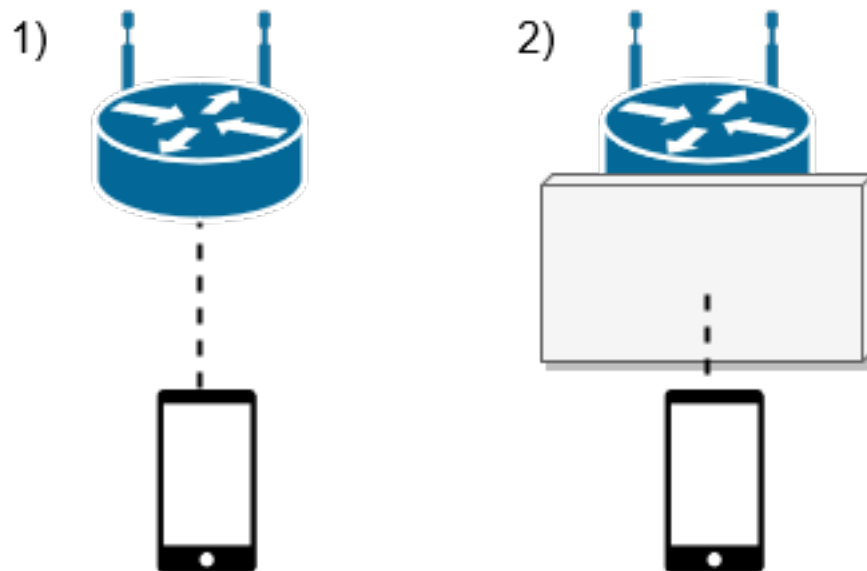
Eksperyment polegał na ustawieniu transpondera 1m od odbiornika na jednym poziomie, antenami do siebie. Następnie dodawana była przeszkoda (w tym wypadku książka) i pomiary zostały powtórzone. Eksperyment został wykonany dla wszystkich transponderów.

- Router TP-Link TD-W8970

Wersja bez przeszkody:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-41	1,16
2	-40	1,04
3	-37	0,73
4	-42	1,30
5	-37	0,73

Rysunek 1: Szkic eksperymentu nr 1



Wersja z przeszkodą:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-38	0,82
2	-39	0,92
3	-42	1,30
4	-46	2,07
7	-43	1,46

- Router TP-Link TL-WA701ND

Wersja bez przeszkody:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-44	0,98
2	-44	0,98
3	-45	1,10
4	-47	1,38
5	-45	1,10

Wersja z przeszkodą:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-49	1,74
2	-47	1,38
3	-46	1,23
4	-47	1,38
5	-47	1,38

- Samsung Grand 2

Wersja bez przeszkody:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-51	1,38
2	-50	1,23
3	-49	1,10
4	-48	0,98
5	-53	1,74

Wersja z przeszkodą:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-50	1,23
2	-54	1,95
3	-53	1,74
4	-55	2,19
5	-55	2,19

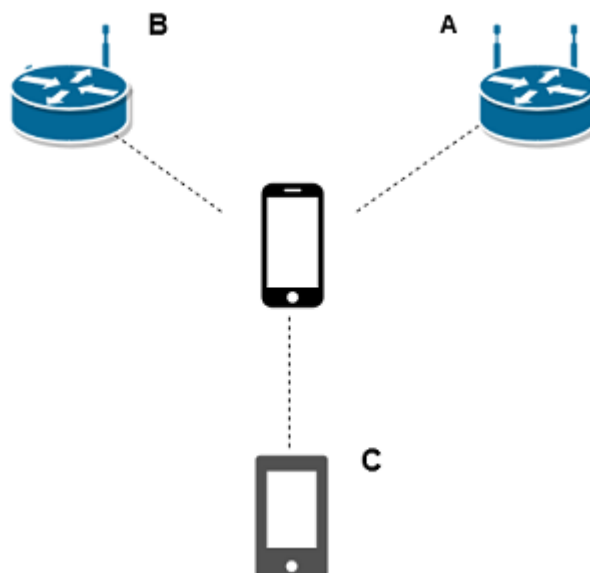
### 3.5 Pomiary zakłóceń

Eksperyment polegał na rozmieszczeniu trasmiterów na wierzchołkach trójkąta, w środku którego znajdował się odbiornik. Wszystkie urządzenia znajdowały się na tej samej wysokości. Mierzone były zmiany siły sygnału i obliczonej odległości w zależności od kąta położenia odbiornika w stosunku do trasmitera oraz ilości nakładających się na siebie sygnałów. Na początku, włączony był tylko trasmiter o indeksie A. Odbiornik znajdował się w stosunku do trasmitera pod kątem około 50 stopni. Następnie włączony został trasmiter B. Na końcu do modelu został dodany trasmiter C.

Informacje o urządzeniach:

- Transmitter A - TP-Link TD-W8970, współrzędne (1.80, 0)
- Transmitter B - TP-Link TL-WA701ND, współrzędne (0, 0)
- Transmitter C - Samsung Grand 2, współrzędne (1.07, 1.8)

Rysunek 2: Model systemu do pomiaru zakłóceń



- Odbiornik - współrzędne (1.2, 0.45)

Pomiar bez zakłóceń dla odległości 80cm przy kącie 50°:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-41	1,16
2	-43	1,46
3	-41	1,16
4	-42	1,30
5	-43	1,46

Pomiar z zakłóceniami z transmitera B dla odległości 80cm przy kącie 50°:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-44	1,64
2	-47	2,31
3	-45	1,84
4	-48	2,60
5	-47	2,31

Pomiar z zakłóceniami z obu transponderów dla odległości 80cm przy kącie 50°:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-48	2,60
2	-47	2,31
3	-44	1,64
4	-48	2,60
5	-46	2,06

### 3.6 Wyznaczanie lokalizacji użytkownika

Narazie mało do napisania. Z dwóch pomiarów dla modelu z góry, dostałem lokalizacje (-0,4; 1,2; -0,3) oraz (1,5; 1,67; -1,5).

## 4 Model wyznaczania lokalizacji

Stworzyłem model algorytmu w MatLabie. Nie wyobrażam sobie modelu w trzech wymiarach i z kolorem (według mnie wynikiem będzie prostopadłościan o granatowym dominującym kolorze ścian), dlatego stworzyłem model 2D, który jest tak naprawdę przekrojem modelu 3D (płaszczyzną XY). Algorytm zmieniłem według zastrzeżeń Pana Doktora. Prostopadłościan, który zawiera w sobie "sfery" ruterów, dzielony jest na max 20 kawałków. Wyznacza się najlepszą pozycję, dla niej brane są sąsiadujące pozycje i uzyskany sześciątka dzieli się na 9 kawałków i ponownie wyznacza najlepszą pozycję. Obliczenia kończą się, jak spełnione jest założenie:  $szerKawalka \leq okreslonaDokladnosc$ .

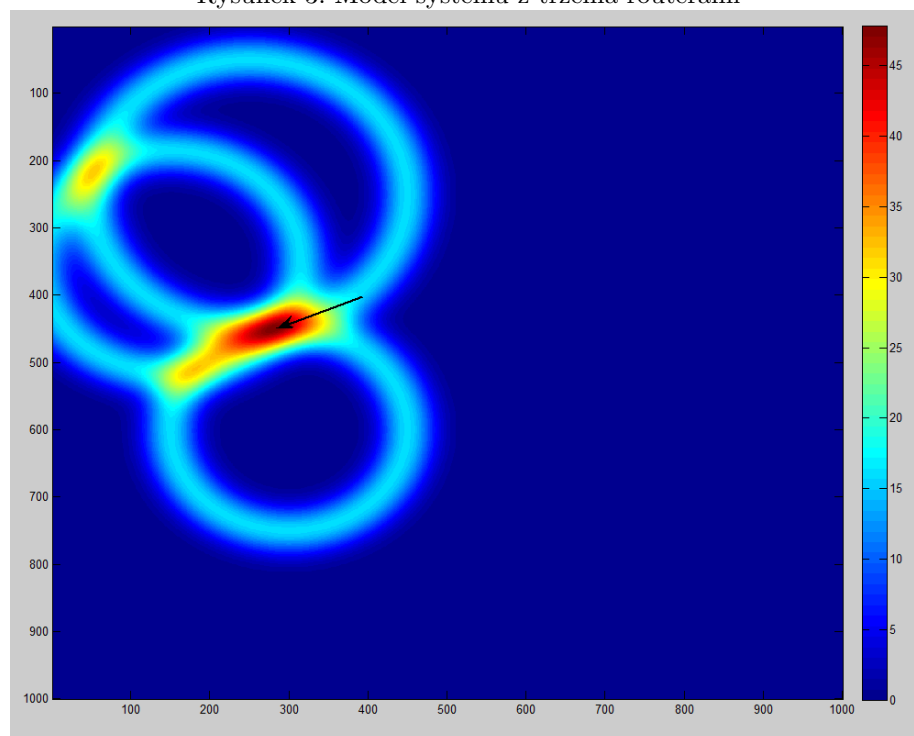
## 5 Implementacja

### 5.1 Serwer

Serwer jest aplikacją, która ma analizować zabrane dane, sterować urządzeniami zewnętrznymi oraz stanowić most pomiędzy klientem administracyjnym, a aplikacjami mobilnymi. Na zadania aplikacji serwerowej składają się:

- Zbieranie danych z aplikacji mobilnych oraz wyznaczanie lokalizacji użytkowników
- Zezwalanie użytkownikowi administracyjnemu na konfigurację systemu poprzez żądań aplikacji klienckiej
- Sterowanie i zarządzanie urządzeniami zewnętrznymi analizując dane odebrane z aplikacji mobilnych
- Dostarczanie, w formie skumulowanej lub w czasie rzeczywistym, danych na temat położenia użytkowników do analizy i monitoringu

Rysunek 3: Model systemu z trzema routerami





### 5.1.1 Lokalizacja użytkowników mobilnych

**Pobranie i analiza danych** Serwer pobiera dane od użytkowników w formie requestów HTTP. Każde żądanie wysłane do serwera musi posiadać adres MAC urządzenia Bluetooth wysyłającego oraz listę zarejestrowanych sygnałów. Każda pozycja na liście sygnałów musi posiadać nazwę urządzenia, którego sygnał został odebrany (w przypadku sygnału wysłanego przez router jest to SSID sieci, zaś w przypadku urządzeń Bluetooth jest to adres MAC), typ sygnału (WIFI albo Bluetooth) oraz zarejestrowaną siłę sygnału, określoną w dBm.

Następnie, dla każdego elementu z listy dociągane są stałe dane zarejestrowane w systemie - lokalizacja oraz waga sygnału. Dane na temat routerów oraz stałych urządzeń Bluetooth (np. Beaconów) pobierane są bazy danych. Jeżeli jakiś sygnał Bluetooth nie widnieje w bazie danych, sprawdzane są ostatnie żądania od urządzeń mobilnych, dla których udało się określić lokalizację, a czas od ostatniej aktualizacji nie jest większy niż 4 sekundy. Jeżeli sygnał pochodzi od jednego z tych urządzeń, potrzebne informacje pobierane są z dynamicznie budowanej, lokalnej bazy wiedzy. Jeżeli sygnał nie figuruje ani w bazie danych, ani w bazie dynamicznej, zostaje uznany jako sygnał przypadkowy i odrzucony. Waga sygnału przyjmuje wartości w skali od 1 do 4. Domyślnie, sygnałowi pochodzącemu od routera WiFi nadawana jest waga 3, sygnałowi ze stałego urządzenia Bluetooth waga 2, zaś sygnały pochodzące od innych użytkowników mobilnych wagę 1. Wagę stałych urządzeń WiFi i Bluetooth można edytować przy użyciu panelu konfiguracyjnego w aplikacji klienckiej. W ostatnim kroku, dla każdego zarejestrowanego sygnału obliczana jest odległość urządzenia od użytkownika. Wykorzystywana do tego jest lokalizacja użytkownika, wzór na Free-space path loss oraz dane statyczne (jak siła anten, siła trasmitera itp.).

**Algorytmiczne wyznaczenie lokalizacji** Celem algorytmu jest wyznaczenie punktu, dla którego suma prawdopodobieństw wynikających z odległości użytkownika od urządzenia, jest największa.

Dane pobrane od użytkownika, uzupełnione o statyczne dane przechowywane w systemie, przekazane są do sekcji napisanej w języku F#. Na wstępie, do każdego zarejestrowanego algorytmu zostaje przypisana probabilistyczna Guassa, określająca prawdopodobieństwo znalezienia się użytkownika w danym punkcie w przestrzeni, gdzie stała  $\mu$  przyjmuje wartość równą dystansowi obliczonemu na podstawie siły sygnału urządzenia. Dzięki takiemu podejściu, każdy sygnał można zwizualizować jako sferę, której powierzchnia zbliżona jest do chmury. Największe zagęszczenie prawdopodobieństwa występuje dla średnicy równej odległości obliczonej z siły sygnału, a która rzadnie zbliżając się i oddalając od środka sfery.

Pierwszym krokiem algorytmu jest wyznaczenie prostopadłościanu, dla którego wykonywane będą obliczenia. Wielkość bryły dobrana jest tak, aby wewnątrz niej znalazły się wszystkie sfery sygnałów (przy uwzględnieniu zapasu równego  $2\sigma$ ). Następnie prostopadłościan oraz sfery są normalizowane w taki sposób, aby początek układu zaczął się w punkcie (0,0,0), zaś wszystkie wartości współrzędnych

przyjmowały tylko wartości nieujemne. Celem takiej operacji jest uproszczenie algorytmu oraz wyzbycie się potrzeby skalowania iteratorów oraz odnośników do elementów w tablicach.

Kolejnym krokiem algorytmu jest podział prostopadłościanu na części, dla których liczona będzie suma prawdopodobieństw. Celem tego kroku jest podział pola działania na jednakowe sześciany w taki sposób, aby w żadnym wymiarze ilość sześciąt nie przekroczyła 100. Aby to uzyskać, najdłuższy bok prostopadłościanu zostaje podzielony na 100 części. Następnie boki w pozostałych 2 wymiarach zostają podzielone na sześciany o krawędziach równej długości. Dzięki takiemu zabiegowi dzieli się prostopadłościan na sześciany. Podział prostopadłościanu na sześciany pozwala na wyeliminowanie błędów obliczeniowych wynikających z nierealistycznego podziału pola oliczeń.

Kolejnym krokiem algorytmu jest wyliczenie sumy prawdopodobieństw ze wszystkich sfer sygnałów dla każdego sześcianu w prostopadłościanie. Każde prawdopodobieństwo, będące składową sumy, przemnażane jest przez wagę danego sygnału. Następnie wybierany jest sześcian, dla którego suma prawdopodobieństwa jest najwyższa. Jeżeli długość boku sześcianu jest równa lub mniejsza od naszego przybliżenia, współrzędna sześcianu staje się lokalizacją naszego użytkownika. Jeżeli długość boku sześcianu jest większa od naszego przybliżenia, dla wybranego sześcianu dobierane są jego sześciany sąsiednie. Następnie wybrane 27 sześciąt staje się nowym modelem obliczeniowym. Każdy wymiar nowego pola działane jest na 9 równych części, dzięki czemu uzyskuje się 729 sześciąt. Algorytm zostaje powtórzony, aż lokalizacja nie zostanie określona z interesującym nas przybliżeniem. Sześcian o największej sumie prawdopodobieństw staje się lokalizacją użytkownika.

Ostatnim krokiem algorytmu jest przeliczenie obliczonej lokalizacji przy użyciu danych uzyskanych podczas normalizacji, aby obliczona lokalizacja odpowiadała lokalizacji dla danych wejściowych.

**Zarządzanie lokalizacją użytkownika** Po pozytywnym obliczeniu lokalizacji, zostaje ona zapisana w bazie danych, aby potem mogła być użyta do wyświetlenia skumulowanej mapy przepływu użytkowników albo do sterowania urządzeniami. Następnie, lokalizacja zostaje asynchronicznie wysłana do wszystkich klientów administracyjnych, którzy zarejestrowali swoją chęć pobierania danych w trybie real-time (w czasie rzeczywistym). Następnie, lokalizacja wraz z adresem MAC użytkownika zostaje przekazana do wątków urządzeń zewnętrznych, które wykorzystują te dane do podjęcia decyzji o wywołaniu przypisanego urządzeniowi eventu. W ostatnim kroku, lokalizacja zostaje dodana (lub podmieniona, jeżeli wpis o danym użytkowniku już istnieje) w bazie dynamicznej, aby ta informacja mogła posłużyć przy wyznaczaniu lokalizacji innych użytkowników.

### 5.1.2 Sterowanie i zarządzanie urządzeniami

Serwer, poza pobieraniem i analizą danych, zajmuje się również sterowaniem przydzielonymi mu urządzeniami. Sterowanie urządzeniami odbywa się na dwa

sposoby:

- Przy użyciu eventów - urządzenia mogą mieć przypisaną klasę obsługującą zdarzenia specjalne. Mechanizm analizuje w czasie rzeczywistym ostatnie zarejestrowane lokalizacje użytkowników, a następnie na ich podstawie, oraz na podstawie wcześniej przypisanych sobie reguł decyduje, czy mają zostać podjęte jakieś działania. Jeżeli tak, komunikuje się z urządzeniem, aby ustawić mu wyznaczone parametry. Wywołana metoda zwraca zmienną typu Boolean, określającą, czy została podjęta decyzja o komunikacji z urządzeniem.
- Co określony interwał czasowy - mechanizm, który co jakiś określony czas oblicza parametry, jakie należy przekazać do urządzenia. Robi to na podstawie zarejestrowanych przez ten czas pozycji użytkownika. Do swoich obliczeń wykorzystuje wagi użytkowników, przypisane im w panelu administracyjnym. Mechanizm czasowy ma niższy priorytet niż mechanizm eventów, dlatego jeżeli klasa obsługująca zdarzenia przekazała urządzeniu parametry, system czasowy zostaje zawieszony aż do kolejnego przejścia pętli. Takie rozwiązanie zapobiega nadpisywaniu parametrów wysłanych do urządzenia, zanim wcześniejsze zostaną wykorzystane.

Informacje na temat urządzeń przechowywane są w bazie danych. Danymi, które są potrzebne do sterowania urządzeniem, niezależnie od jego typu są:

- Jego lokalizacja (określona przez 3 współrzędne)
- Ip urządzenia oraz port, na którym nasłuchuje
- Nazwę rozpoznawalną przez użytkownika (np żarówka na korytarzu)
- Sterownik określający sposób komunikacji, implementujący interfejs przypisany do konkretnego typu urządzenia
- Moduł określający, czy dla danego urządzenia przypisane jest jakieś sterowanie eventami (np natychmiastowa zmiana siły oświetlenia, spowodowana zbliżeniem się określonego użytkownika)
- Flaga określająca, czy dane urządzenie jest aktywne

**Dobór parametrów sterujących** Dla każdego aktywnego urządzenia zapisanego w systemie uruchomiony jest na serwerze osobny wątek sterujący. Taki sposób pracy został przyjęty, aby sterowanie i obsługa eventów odbywała się płynnie i w równy sposób dla wszystkich urządzeń, a błąd czy problemy komunikacyjne jednego z urządzeń nie miały wpływu na inne. Do każdego wątku sterującego przypisany jest obiekt klasy zawierającej wszystkie potrzebne informacje oraz metody, aby kontrolować dany typ urządzenia (dla oświetlenia jest to klasa `LightDeviceControllingThread`).

Praca wątku polega na wywołaniu metody `StartControll()`, na której ciało ślada się nieskończona pętla `while`. Niezależnie od typu urządzenia, którym zarządza

wątek, każda klasa posiada obiekt kolejki, do której kontroler przyjmujący dane od urządzeń mobilnych wstawia obliczone lokalizacje (pod warunkiem, że dla urządzenia sterowanego przypisano klasę obsługującą zdarzenia).

Kolejnym krokiem, jaki wykonywany jest w metodzie, jest pobranie z kolejki wszystkich oczekujących lokalizacji, a następnie przekazanie ich do klasy sterującej eventami. W systemie zaimplementowana jest przykładowa klasa sterująca światłem - ImportantUserFirstContr. Wybiera z listy użytkowników, którzy są w bliskiej odległości od źródła światła, a następnie wyszukuje wśród nich użytkowników o najwyższej wadze. Jeżeli taki użytkownik zostanie znaleziony, system wysyła do urządzenia rozkaz ustawienia światła o największej mocy. Taki poziom zostanie utrzymany aż ostatni użytkownik uprzywilejowany nie oddali się od źródła światła. System poczeka wtedy dodatkowe 4 sekundy, a następnie wyśle do urządzenia informację o ustawieniu mocy światła na poprzednią wartość. System pozwala na rozszerzanie klas sterujących oraz tworzenie swoich i przypisywanie ich urządzeniom (o ile odpowiedni wpis zostanie dodany do bazy danych).