1 Teoria

1.1 Free-space path loss

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

$$FSPL = P_{tx} + AG_{tx} + AG_{rx} - P_{rx} - FM - L \tag{1}$$

Gdzie symbole oznaczają:

- P_{tx} - siła trasmitera, wyrażona w dBm
- AG_{tx} - zysk energentyczny anteny transmitera, 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 transmiterów, przeszkód itp.

Dodatkowo, FSPL można obliczyć, używając następujący wzór:

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

Gdzie symbole oznaczają:

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

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

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

Po przekształceniu wzoru, uzytkujemy:

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

A po połączeniu obu wzorów dostajemy:

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

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

1.3 Received signal strength indication

Received signal strength indication (skrótem 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 przyjęte 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 transmiterami 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

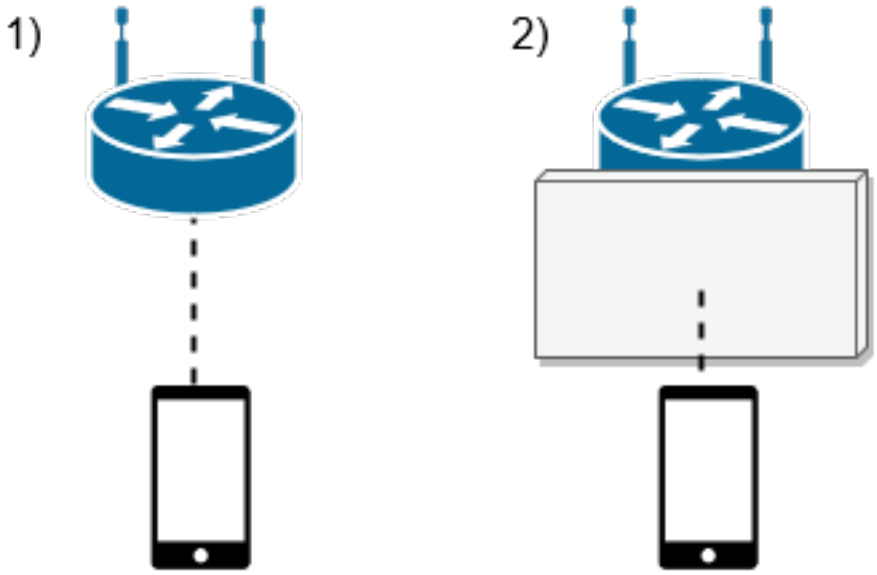
Eksperyment polegał na ustawieniu transmitera 1m od odbiornika na jednym poziomie, antenami do siebie. Następnie dodawana była przeszkoda (w tym wypadku książka) i pomiary zostały powtórzone. Eksperyment został wykonany dla wszystkich transmiterów.

- 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

//

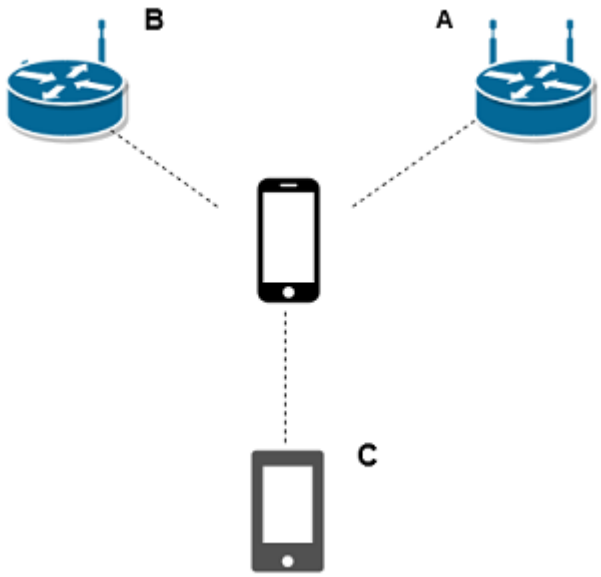
3.5 Pomiary zakłóceń

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 transmiter o indeksie A. Odbiornik znajdował się w stosunku do transmitera pod kątem około 50 stopni. Następnie włączony został transmiter B. Na końcu do modelu został dodany trasmiter C.

Informacje o urządzeniach:

- Transmiter A - TP-Link TD-W8970, współrzędne (1.80, 0)
- Transmiter B - TP-Link TL-WA701ND, współrzędne (0, 0)
- Transmiter 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 transmiterów 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 pozycję i uzyskany sześcian dzieli się na 9 kawałków i ponownie wyznacza najlepszą pozycję. Obliczenia kończą się, jak spełnione jest założenie: $szerKawalka \leqslant okreslonaDokladnosc$.

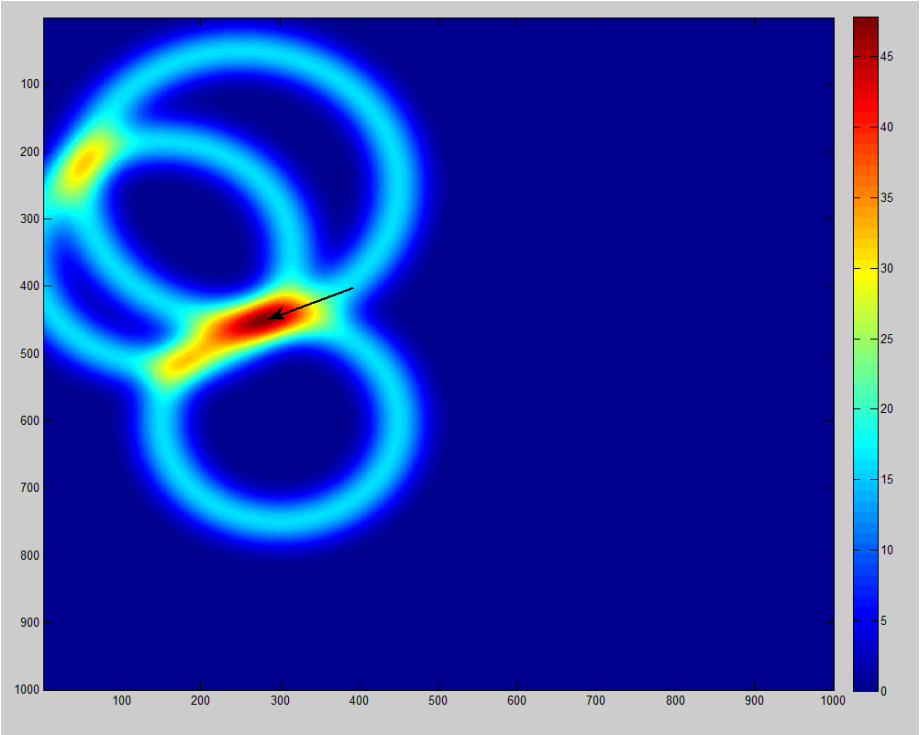
5 Implementacja

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 systemi - lokalizacja oraz waga sygnału. Dane na temat routerów oraz stałych urządzeń Bluetooth (np. Beconów) 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 lokalizacjęa, 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 edytowac 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 zarejestrownego algorytmu zostaje przypisana probabilistyczna Guassa, określająca prawdopodobieństwo znalezienia się użytkownika w danym punkcie w przestrzeni, gdzie stała μ 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 występuję dla średnicy równej odległości obliczonej z siły sygnału, a która rzednie zbliżając się i oddalając od środka sfery.
Pierwszym krokiem algorytmu jest wyznaczenie prostapadłościanu, dla którego wykonywane będą obliczenia. Wielkość bryły dobrana jest tak, aby wewnątrz niej znalazły się wszystkie sfery sygnałów (przy uwzględniu zapasu równego 2σ). Następnie prostopadłościan oraz sfery są normalizowane w taki sposób, aby początek układu zaczył 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 takie sposób, aby w żadnym wymiarze ilość sześcianów nie przekroczyła 100. Aby to uzyskać, najdłuższy bok prostopadłościanu zostaje podzielony na 100 częście. 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 algorymtu jest wyliczenie sumy prawdopodobieńst ze wszystkich sfer sygnałów dla każdego sześcianu w prostopadłościanie. Każde prawdopodieństo, 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 sześcianów staje się nowym modelem obliczeniowym. Każdy wymiar nowego pola dzialone jest na 9 równych części, dzięki czemu uzyskuje się 729 sześcianów. 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 podjętę 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 użytkowników. 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 słada 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 wsród 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 urządzania 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).
Jeżeli moduł zarządzania eventami zadecyduje o wysłaniu do urządzenia parametrów ustawiających, wykonuje się moduł zarządzania czasowego. Przed wykonaniem obliczeń, algorytm sprawdza, czy od ostatniej czasowej aktualizacji minął odpowiedni okres czasu (domyslnie algorytm ma się wykonywać co godzinę). Data poprzedniej aktualizacji przechowywana jest w polu LastUpdate, które jest aktualizowane obecną datą za każdym razem, jak algorytm pozytywnie wyliczy parametry dla urządzenia. Jeżeli odpowieni czas minął, program pobiera z bazy wszystkie pozycje użytkowników, które zostały zarejestrowane w okresie od ostatniej aktualizacji. Następnie, algorytm wybiera z pobranej listy użytkowników, którzy zostali zarejestrowani w odległości nie większej niż, domyślne, 5 metrów, a nestępnie sumuje ich wagi. Parametr, jaki ma zostać wysłany do urządzenia jest liczony na podstawie wzoru:

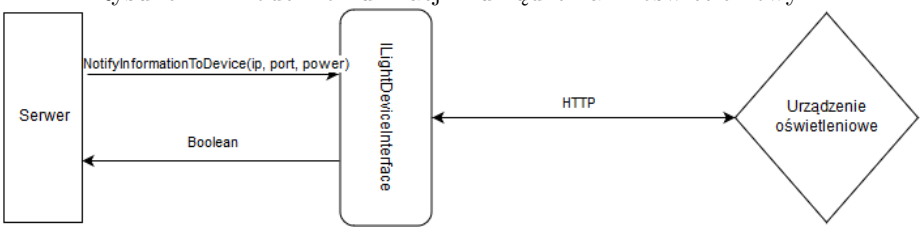
$$P_{ustalona} = (P_{max} - P_{min}) * \frac{N_{blisk}}{N_{og}} + P_{min} \tag{6}$$

gdzie zmienne w równaniu oznaczają:

- $P_{ustalona}$ - moc światła jaka ma być ustawiona dla urządzenia
- P_{min} - minimalna moc światła ustawiona dla urządzenia
- P_{max} - maksymalna moc światła ustawiona dla urządzenia
- N_{blisk} - suma wag użytkowników, którzy zostali zarejestrowani wystarczająco blisko urządzenia
- N_{og} - suma wag wszystkich użytkowników w danym okresie czasowym

Wysyłany parametr jest ustawiany dodatkowo w zmiennej PreviousStaticPowerLevel, aby mógł być potem, w ramach potrzeby, wykorzystany przez moduł sterujący eventami. Na końcu, do zmiennej przechowywującej ostatnią aktualizacje zostaje przypisana obecna data i godzina.

Rysunek 4: Model komunikacji z urządzeniami oświetleniowymi



Jeżeli parametry urządzenia zostały zaktualizone w obecnym przebiegu algorytmu, ponowne wykonywanie algorytmu rozpoczyna się od razu. Jeżeli nie, wątek zostaje uśpiony na 0,1 sekundy, aby nie zurzywać niepotrzebnie zasobów serwera.

Komunikacja z urządzeniami Sposób komunikacji serwera z urządzeniem nie jest zapisany bezpośrednio w kodzie serwera, ale jest przechowywany w bazie danych i przypisywany indywidualnie do konretnego urządzenia. Może zostać zmieniony przy użyciu panelu administracyjnego. Dla każdego typu urządzenia może być zdefiniowana pewna pula modułów komunikacji z urządzeniami. Każdy typ ma określony interfejs, po którym ma dziedziczyć klasa komunikująca się z tego typu urządzeniami. Wewnątrz modułu komunikacyjnego określony jest format wysyłanej wiadomości, kodowanie oraz kolejność parametrów. Ilość metod, które ma udostępniać klasa implementująca interfejs jest zależna od ilości parametrów, jakie mogą być ustawiane w danym typie urządzeń.
W systemie został stworzony intefejs ILightDeviceInterface. Powinien być implementowany przez moduły do komunikacyjni z urządzeniami odpowiedzialnymi za oświetlenie. Z racji tego, iż światło posiada tylko jeden sterowalny parametr - moc światła, interfejs narzuca na klasie zaimplementowanie metody NotifyInformationToDevice, która jako parametry przyjmuje ip urządzenia, jego port oraz wartość mocy światła, jaka ma być wysłana do urządzenia. Zadaniem implementacji tej metody jest sformatowanie wiadomości, zawierającej ustawianą moc światła, według standardu urządzenia oraz wysłanie jej protokołem obslugiwanym przez urządzenie. Metoda ma zwrócić zmienną typu Boolean, informującą o tym, czy komunikacja i wysłanie wiadomości się powiodło.
W ramach przykładu, stworzony został moduł do komunikacji z urządzeniami oświetleniowymi, LoggerLightDeviceInterface. Wysyła on protokołem HTTP na podany adres Ip wiadomość o formacie *"Należy ustawić wartość na: $P_{swiatla}$"*, gdzie $P_{swiatla}$ to moc światła, którą chcemy ustawić.

5.1.3 Konfiguracja systemu

Aplikacja serwerowa pozwala na konfigurację elementów systemu bez potrzeby jego restartowania. Serwer wykorzystuje do tego kontroler ConfController, przyjmujący żądania HTTP. Podstawowymi elementami, które mogą być konfigurowane w taki sposób są:

- Wielkość pola operacyjnego - wartość określająca, jak duży jest obszar, na którym zainstalowany jest system. Wewnątrz tego pola muszą znajdować się wszystkie obsługiwane routery i urządzenia Bluetooth, ktore nie zmieniają swojej pozycji (np Becony). Jednostką, w jakiej zapisana jest wielkość mapy, są metry. Wartość ta wykorzystywana jest podczas wyświetlania lokalizacji użytkowników w aplikacji administracyjnej - współrzędna lokalizacji zostaje przeskalowana przy użyciu wielkości pola w taki sposób, aby realistycznie odzorowywała położenie na mapie wyświetlanej w aplikacji administracyjnej.

- Dane o routerach Wifi i nieporuszających się urządzeniach Bluetooth - administrator systemu ma możliwość zmiany kluczowych danych na temat urządzeń, na podstawie których lokalizowani są potem użytkownicy. Poza danymi technicznymi, dotyczącymi specyfikacji urządzenia (siła trasmitera, siła anten, straty wynikające z kształtu obudowy - np antena wewnętrzna), zdefiniowane są również:

  - typ sygnału wysyłanego przez urządzenie - Wifi albo Bluetooth
  - nazwa, po której urządzenie jest rozpoznawalne przez aplikację mobilną - SSID w przypadku routera WiFi, Mac w przypadku urządzenia Bluetooth
  - lokalizacja urządzenia - wymagane jest podanie lokalizacji w trzech wymiarach. Współrzędne urządzenia, jak wszystkie współrzędne w systemie, są podawane stosunku punktu określonego jako początek pola działania systemu.
  - waga urządzenia, która uwzględniana jest podczas lokalizowania użytkownika. Wartość ta podczas dodawania nowego urządzenia, inicjalizowana jest wartościami domyślnymi, które potem można zmienić. Dopuszczalna wartość wagi jest liczbą całkowitą z zakresu $< 1 : 4 >$. Domyślne wartości to:
    - dla urządzeń Bluetooth, dynamicznie zmieniających swoją pozycję (użytkownicy aplikacji mobilnej) - 1
    - dla urządzeń Bluetooth, których lokalizacja jest określona - 2
    - dla urządzeń Wifi - 3

  Dane na temat routerów przechowywane są w bazie danych, skąd są pobierane podczas włączania się aplikacji serwerowej. Informacje te są również przechowywane lokalnie, w pamięci serwera. Ma to na celu zmiejszenie czasu wykonywania operacji, ponieważ pobieranie danych z bazy danych jest dużo wolniejsze niż korzystanie ze zmiennych zapisanych w pamięci. Dlatego, każdorazowa zmiana informacji o routerach powoduje zmianę danych przechowywanych lokalnie przez serwer. System zakłada, że dane o routerach dokonywane są tylko przy użyciu aplikacji serwerowej - zmiany dokonane na bazie przez zewnętrzne aplikacje (np GUI bazy danych) nie zostaną uwzględnione przez serwer aż do jego restartu.

- Dane na temat urządzeń sterowanych - podobnie jak w przypadku urządzeń do lokalizacji użytkowników, sterowane urządzenia mogą być dodawane, edytowane oraz usuwane przez administratorów systemu. Informacje różnią się w zależności od kategorii, do której należy dane urządzenie, jednak część z nich jest wspólna:
  - nazwa - nazwa urządzenia, która ma pozwolić na jego identyfikację przez użytkowników systemu
  - ip i port - podstawowe dane wykorzystywane do komunikacji z urządzeniem
  - lokalizacja - położenie urządzenia określone przez trzy współrzędne
  - moduł eventów - nazwa modułu, który ma obsługiwać wydarzenia związane z danym urządzeniem. Do urządzenia może nie być przypisany żaden moduł eventów
  - moduł komunikacyjny - nazwa modułu, który ma pozwolić na komunikację serwera z urządzeniem
  - czy moduł jest aktywny - informacja ta jest wykorzystywana podczas sterowania urządzeniem

  Informacje na temat sterowanych urządzeń, poza wpisami w bazie danych, przechowywane są również w pamięci serwera. W przeciwieństwie do routerów (których dane przechowywane są statycznie w klasie SystemDataKnowledge), dane na temat urządzeń znajdują się w przypisanych im obiektach sterujących. Jest to związane z tym, że wątki sterujące są jedynymi elementami aplikacji serwerowej, które często korzystają z tej informacji.
  Z racji tego, iż z każdym sterowanym urządzeniem, zarządzanym przez system, związany jest obiekt sterujący, kontroler konfigurujący odpowiedzialny jest również za edycję wątków sterujących. W przypadku dodawania do systemu nowego sterowanego urządzenia, kontroler inicjalizuje nowy obiekt sterujący oraz startuje związany z nim wątek. W przypadku usuwanie urządzenia, kontroler zatrzymuje wątek sterujący i usuwa jego obiekt z listy obiektów sterujących.

- Dane na temat użytkowników systemu - kontroler konfiguracyjny pozwala na edycję wagi użytkownika oraz jego nazwy. System rozpoznaje użytkowników po adresie MAC - zmiana nazwy użytkownika ma na celu jedynie pomóc osobie korzystającej z systemu łatwiejsze rozpoznawanie użytkowników. Waga używana jest przy sterowaniu urządzeniami - użytkownik o wadze 4 (przykładowo szef) jest ważniejsza z punktu widzenia modułu określający parametry urządzeń, niż osoba o wadze 1 (zwykły pracownik). Dodatkowo, waga może mieć wpływ na moduły obsługi zdarzeń, przypisany do urządzenia - światło urządzenia może być ustawiane wartością maksymalną w przypadku pojawienia się w okolicy użytkownika o wadze 4.

Dane o użytkownikach przechowywane są w bazie - nie ma potrzeby przechowywania lokalnej kopii.

Poza konfiguracją systemu, kontroler dostarcza aplikacji administracyjnej danych, potrzebnych do dokonywania zmian w systemie oraz poprawnego działania aplikacji klienckiej. Możliwe jest pobranie danych na temat:

- routerów
- typach sygnałów routerów
- sterowanych urządzeniach
- kategoriach sterowanych urządzeń
- dostępnych modułach sterujących eventami
- dostępnych modułach komunikacyjnych
- wielkości pola działania systemu
- użytkownikach

Dane, w zależności od rodzaju, pobierane są z bazy danych lub lokalnej pamięci serwera.

5.1.4 Zarządzanie pozostałymi funkcjami aplikacji administracyjnej

Do obsługi pozostałych, poza konfiguracją, żądań aplikacji klienckiej służy kontroler ClientController. Zawiera on metody pozwalające na:

- logowanie i wylogowywanie się użytkownika - użytkownika w momencie włączania aplikacji zmuszony jest do zalogowania się. W momencie logowania, na serwer zostaje wysłane żądanie zawierające jego login. Następnie wysłany login zostaje powiązany z adresem ip, z którego przyszło żądanie. Jest to robione w celu umożliwienia późniejszego wysyłania informacji do klienta bez potrzeby wysłania przez niego żądania - wykorzystywane w momencie, w którym użytkownik zadeklaruje chęć monitorowania użytkowników mobilnych.
- pobranie informacji o sterowanych urządzeniach zarejestrowanych w systemie. Ta metoda różni się od metody pobrania urządzeń z kontrolera konfigurującego - lista urządzeń jest pobierana w celu wyświetlenia na mapie. Dodatkowo, pobierane są wszystkie przyrządy, bez względu na ich kategorię. Jedynymi zwracanymi informacjami są lokalizacja urządzenia (w celu wyświetlenia w odpowiednim miejscu) oraz jego kategoria (aby aplikacja kliencka mogła dobrać odpowiednią ikonkę).

- zgłoszenie żądania zapisania użytkownika do subskrypcji lokalizacji użytkowników - system zostaje poinformowany, że dany administrator chce monitorować wszystkich pojawiających się użytkowników aplikacji mobilnej. W momencie określenia lokalizacji użytkownika, do wszystkich użytkowników, na adres ip pobrany podczas logowania, zostaje wysłana notyfikacja zawierająca współrzędne użytkownika oraz jego adres MAC.
- pobranie lokalizacji użytkowników z danego okresu czasowego - administrator zgłasza chęć pobrania wszystkich lokalizacji, które zostały zarejestrowane w podanym przez niego okresie czasu. W odpowiedzi, do osoby żądającej, wysłana zostaje lista zawierająca współrzędne lokalizacji użytkowników oraz ich adresy MAC.

5.2 Aplikacja administracyjna

Głównym celem aplikacji klienckiej jest konfiguracja systemu i wyświetlanie oraz monitorowanie lokalizacji użytkowników.