

# MAPacket

Budowanie topologii sieci na podstawie plików PCAP

Skład zespołu:

Łukasz Knop  
Adam Matuszak  
Szymon Kaszuba

# Spis treści

<b>Wstęp</b>	<b>3</b>
<b>Dlaczego wybraliśmy ten temat?</b>	<b>3</b>
<b>MAPacket</b>	<b>3</b>
Wymagania	3
Instalacja	3
Funkcjonalności	3
<b>Instrukcja obsługi</b>	<b>5</b>
Interfejs aplikacji	5
Menu główne	6
Zakładki	7
Okno pliku .pcap	8
Wczytanie pliku pcap	9
Zapisanie plik pcap	10
Zamknięcie karty	11
Zamknięcie wielu kart	13
Zamknięcie aplikacji	16
Filtrowanie pakietów	17
Łączenie plików	17
Generowanie grafu	17
Nawigowanie grafu	18
<b>Podsumowanie</b>	<b>20</b>
Podobne aplikacje	20
Możliwe kierunki rozwoju	20
<b>Sprawozdanie zespołu</b>	<b>21</b>
Organizacja zespołu	21
Środki implementacji	22
Platformy programowania	22
Środowisko	22
Narzędzia	22
Ocena projektu i realizacji	22
Ukończenie projektu	22
Praca zespołowa	22
Metodyka pracy	22
Napotkane problemy	23

# 1. Wstęp

## 1.1. O projekcie

Aplikacja MAPacket powstała w ramach zajęć z przedmiotu Podstawy Teleinformatyki. Projekt wykonany został całkowicie przez trzyosobowy zespół studentów Politechniki Poznańskiej i został poddany ocenie przez osobę prowadzącą zajęcia.

Z pomocą tego programu można wygenerować graf reprezentujący topologię sieci na podstawie pakietów zawartych w pliku o rozszerzeniu .pcap. Program oferuje także możliwość łączenia plików .pcap. Funkcja ta wspiera dodatkowo możliwość generowania grafu poprzez zgromadzenie w pojedynczym pliku jak największej porcji danych o sieci. Docelowo połączone powinny być pliki uzyskane w wyniku nagrywania ruchu (na przykład za pomocą Wireshark) z różnych stacji w tej samej sieci. W efekcie pozwoli to na dokładniejsze odwzorowanie istniejącej struktury połączeń pomiędzy poszczególnymi elementami sieci.

## 1.2. Dlaczego wybraliśmy taki temat?

Nasz zespół zdecydował się wybrać ten temat, ponieważ nie znaliśmy żadnego programu o podobnym funkcjach i uznaliśmy ten temat za ciekawy. W ramach tych zajęć chcieliśmy zrealizować aplikację konsolową napisaną w języku C#. Jako, że nasz projekt zakłada, że dane wejściowe będą pochodzić z nagranych sesji podsłuchiwanie sieci lokalnej, nagranych przez inne aplikacje, wybór aplikacji okienkowej był wskazany.

# 2. MAPacket

## 2.1. Wymagania

Program do działania wymaga zainstalowanego Frameworka .NET w wersji 4.5, jak i również biblioteki WinCap, która instalowana jest chociażby wraz z programem Wireshark.

## 2.2. Instalacja

Dostęp do aplikacji i całego projektu można znaleźć na publicznie dostępnym repozytorium GitHub. Aby uruchomić aplikację należy znaleźć folder Release. W tym folderze znajduje się plik wykonawczy o nazwie MAPacket.exe.

## 2.3. Funkcjonalności

Opisany w tym dokumencie program o nazwie MAPacket służy budowaniu grafu reprezentującego strukturę sieci na podstawie analizowanego pliku .pcap. Po otwarciu pliku .pcap program umożliwia uruchomienie procesu analizy, którego wynikiem jest graf przedstawiający unikalne urządzenia oraz kierunek przesyłania danych pomiędzy wyróżnionymi interfejsami sieciowymi.

Program zapewnia możliwość otwierania i przeglądania plików o rozszerzeniu .pcap, które są wykorzystywane przy generowaniu grafu. W celu umożliwienia analizy i przeglądu dostępnych danych z wielu plików .pcap zdecydowano się na reprezentację danych za pomocą zakładek i tabel, oraz udostępniono możliwość filtrowania danych za pomocą różnych parametrów.

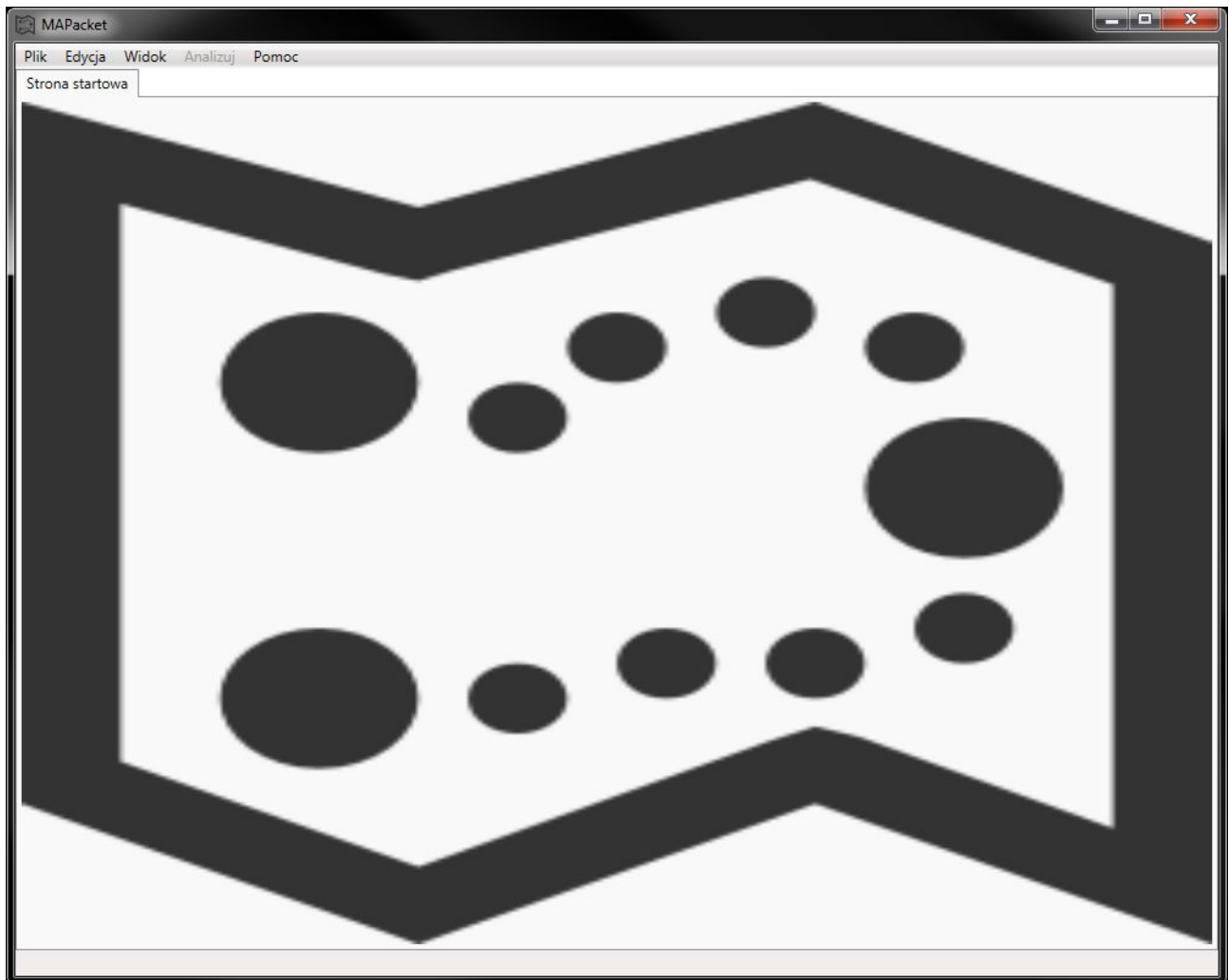
Oprócz możliwości otwierania i łączenia plików o rozszerzeniu .pcap program umożliwia zapisywanie wygenerowanych poprzez łączenie nowych zbiorów pakietów do formatu .pcap.

Dodatkowo program wspiera wyżej opisane funkcjonalności poprzez umożliwienie łączenia wielu plików o rozszerzeniu .pcap z automatycznym usunięciem redundancji nasłuchiowanych pakietów. Połączone w ten sposób pliki mogą zostać przeanalizowane w celu stworzenia dokładniejszego obrazu sieci. Docelowo powinno się łączyć pliki, które nagrano w tej samej sieci. W przypadku łączenia nagrań z niezależnych sieci dojdzie do powstania grafu o dwóch niepowiązanych grupach wierzchołków.

W trakcie generowania grafu dane podlegają dodatkowej analizie, która między innymi liczy różnicę czasu pomiędzy pierwszym i ostatnim odebrany pakietem w danym połączeniu pomiędzy unikalnymi interfejsami sieciowymi, zlicza także ilość przesyłanych bitów. Dane te są wykorzystywane przy szacowaniu prędkości połączenia pomiędzy urządzeniami. Analizie podlega także liczba interfejsów przypisana do pojedynczego urządzenia.

## 3. Instrukcja obsługi

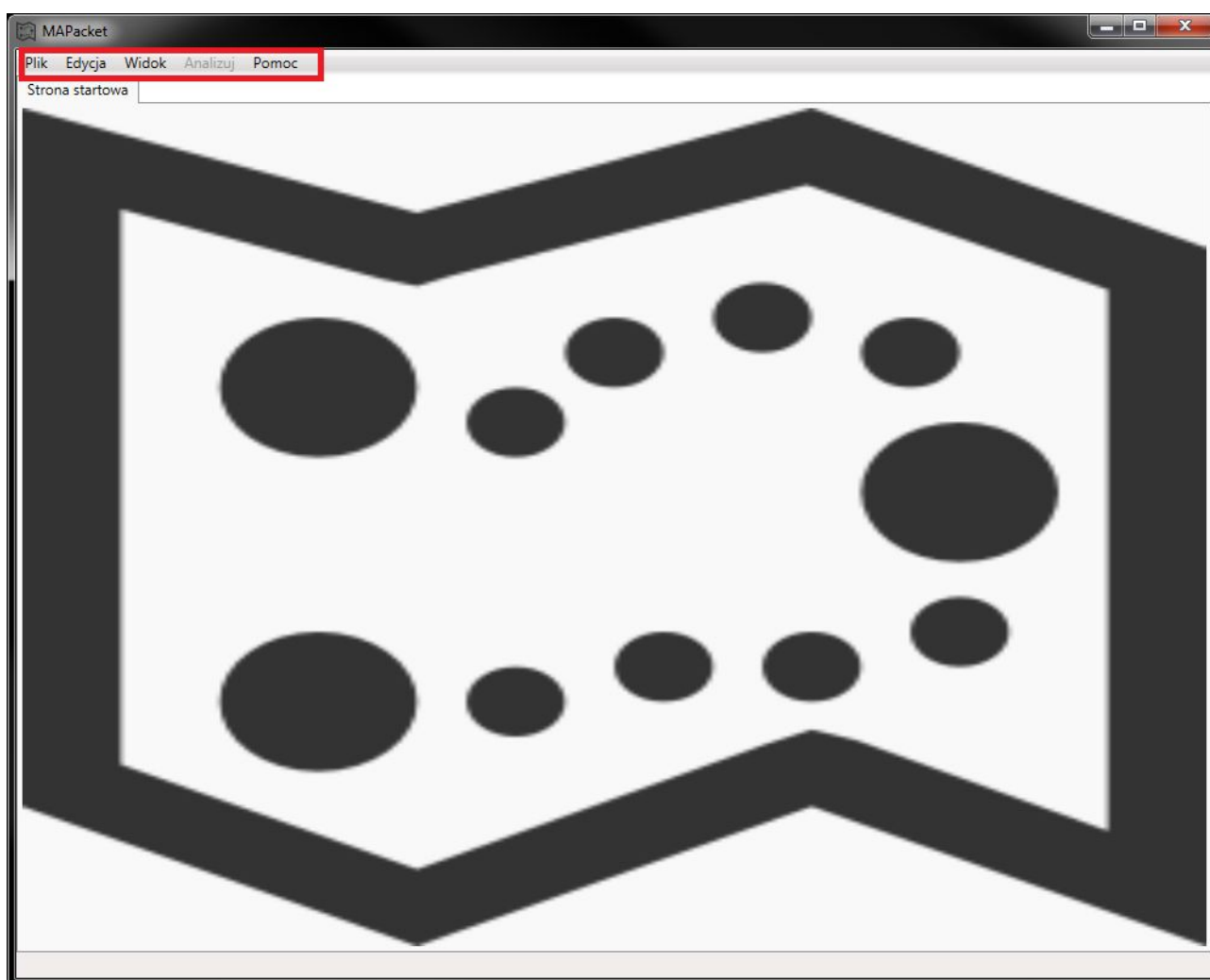
### 3.1. Interfejs aplikacji



*Ilustracja 1 Główne okno aplikacji*

Na ilustracji 1 przedstawiony jest widok obserwowalny po uruchomieniu aplikacji. Zawiera on stronę startową umieszczoną w zakładce pod głównym menu nawigacyjnym programu. Z tego okna mamy dostęp do najważniejszych funkcji programu pogrupowanych pod intuicyjnymi przyciskami w menu głównym.

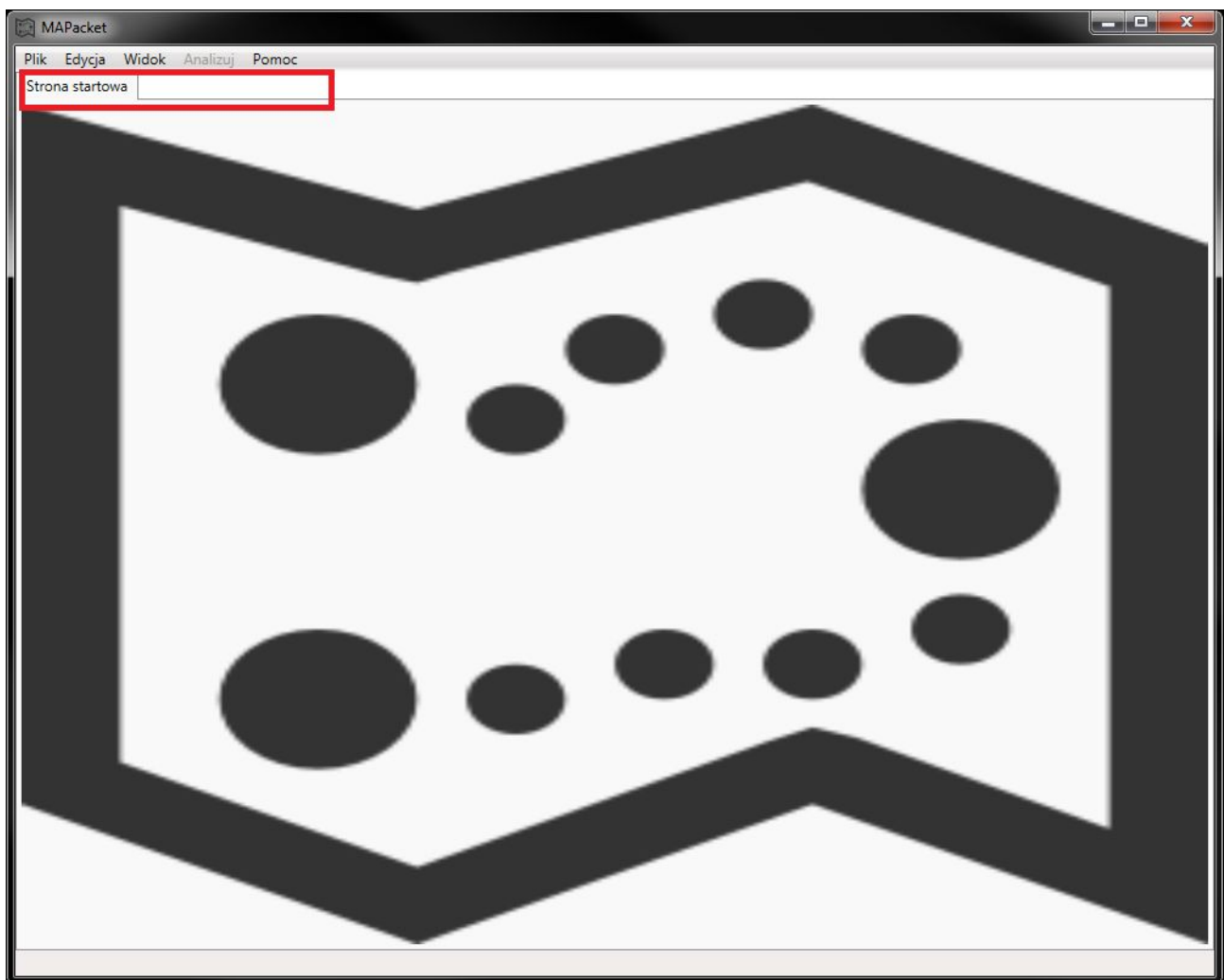
### 3.1.1. Menu główne



Ilustracja 2 Główne okno aplikacji z zaznaczonym menu

Na ilustracji 2 przedstawione zostało menu główne aplikacji służące nawigacji w programie. Umieszczone zostało bezpośrednio w górnej części aplikacji. Jest to związane z intuicyjnością i standardem stosowanym w aplikacjach okienkowych. Menu posiada rozwijaną listę menu kontekstowego, gdzie każda z opcji odpowiada poszczególnym funkcjonalnością programu. Na liście menu można wyróżnić takie obiekty nawigacyjne jak *Plik*, *Edycja*, *Widok*, *Analizuj*, *Pomoc*. Kontrolka *Plik* pozwala na zarządzanie wczytywaniem nowych danych do aplikacji, zapisywaniem i zamykaniem otwartych plików .pcap oraz zamknięcie aplikacji. *Edycja* służy inicjowaniu łączenia plików .pcap. *Widok* służy wyświetlaniu karty powitalnej programu. *Analizuj* zapewnia dostęp do generowania grafów. Pod kontrolką *Pomoc* umieszczone zostały informacje takie jak autorzy projektu, czy link odsyłający do dokumentacji projektu.

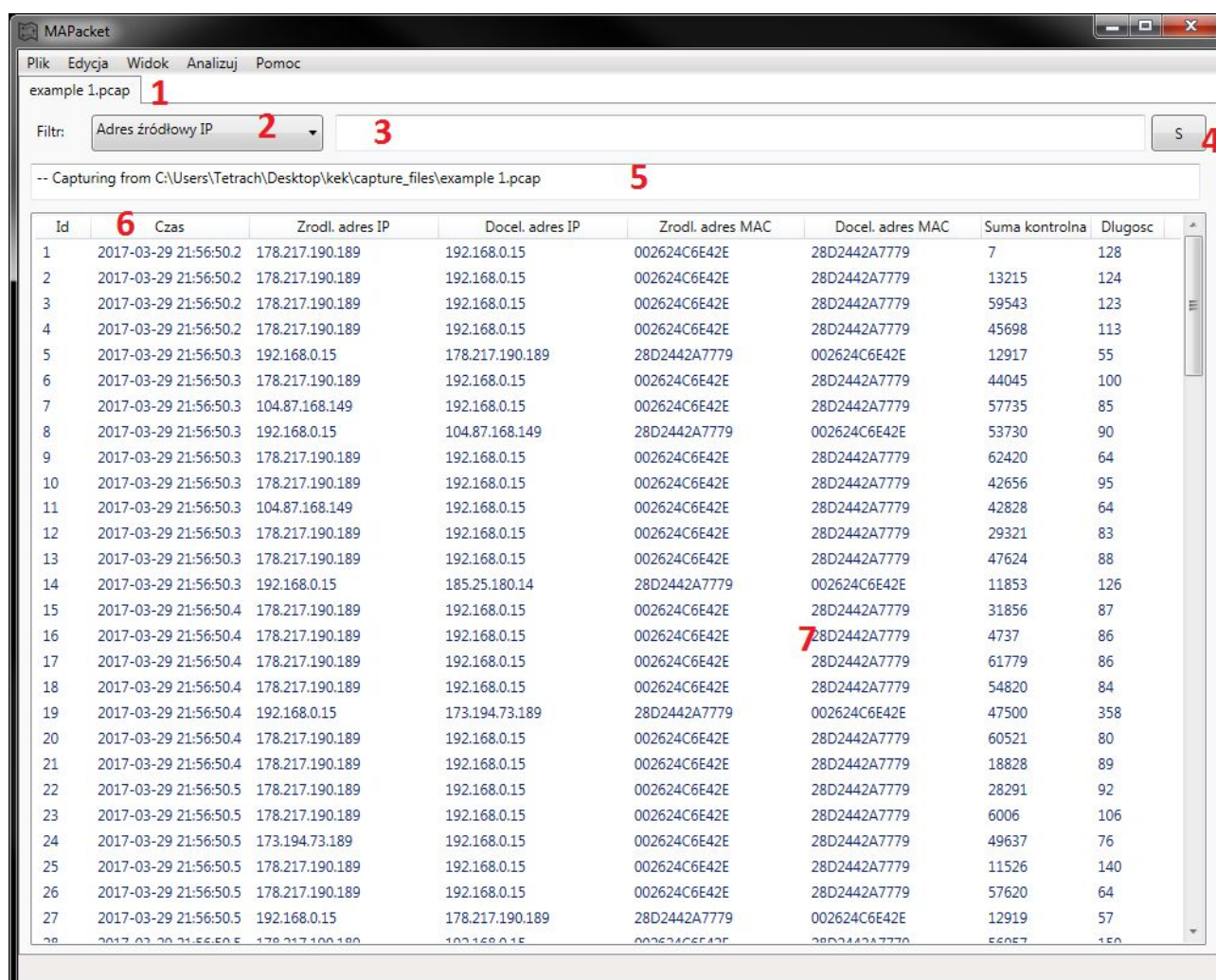
### 3.1.2. Zakładki



*Ilustracja 3 Główne okno aplikacji - zakładki*

Na ilustracji 3 przedstawiony został pasek zakładek. Każdy z obiektów reprezentuje jeden z obiektów: zawartość pliku PCAP, Graf wygenerowany na podstawie pliku PCAP, Strona startowa. Nagłówek zakładki tworzony jest na podstawie nazwy pliku.

### 3.1.3. Okno pliku .pcap



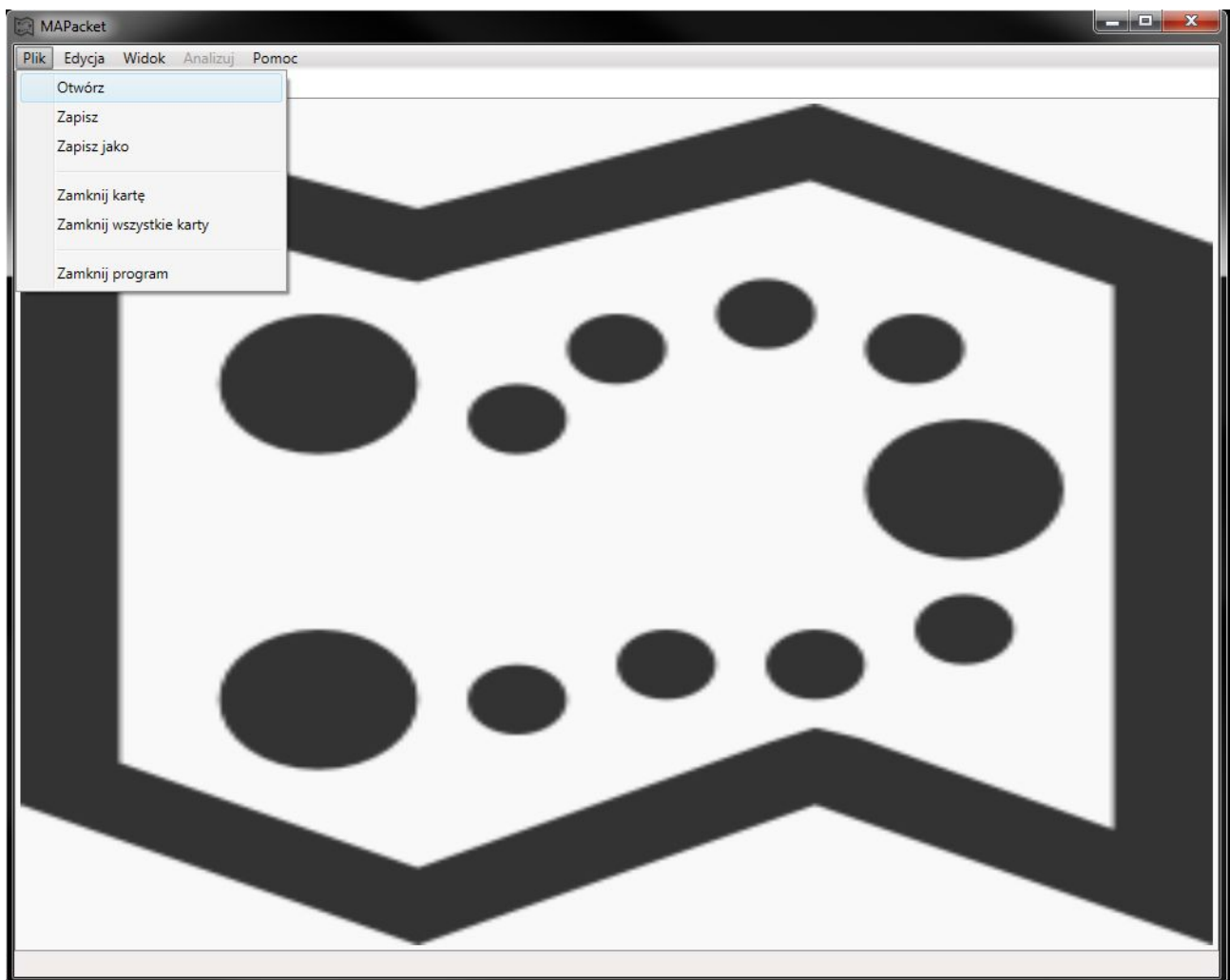
Ilustracja 4 Okno pliku .pcap

Zakładka składa się z kilku elementów opisanych numerami na ilustracji numer 4.

1. To nagłówek zakładki zgodny z nazwą wczytanego pliku.
2. Lista rozwijana zawierająca listę dostępnych filtrów.
3. Reprezentuje pole tekstowe, gdzie możliwe jest podanie parametru filtra np. szukanego adresu IP.
4. Przycisk wywołujący funkcję filtrującą
5. Pole tekstowe prezentujące pełną ścieżkę wczytanego pliku.
6. Nagłówki informacji odczytanych z każdego wiersza pliku pcap.
7. To zbiór skategoryzowanych informacji prezentujący wybraną zawartość pliku pcap

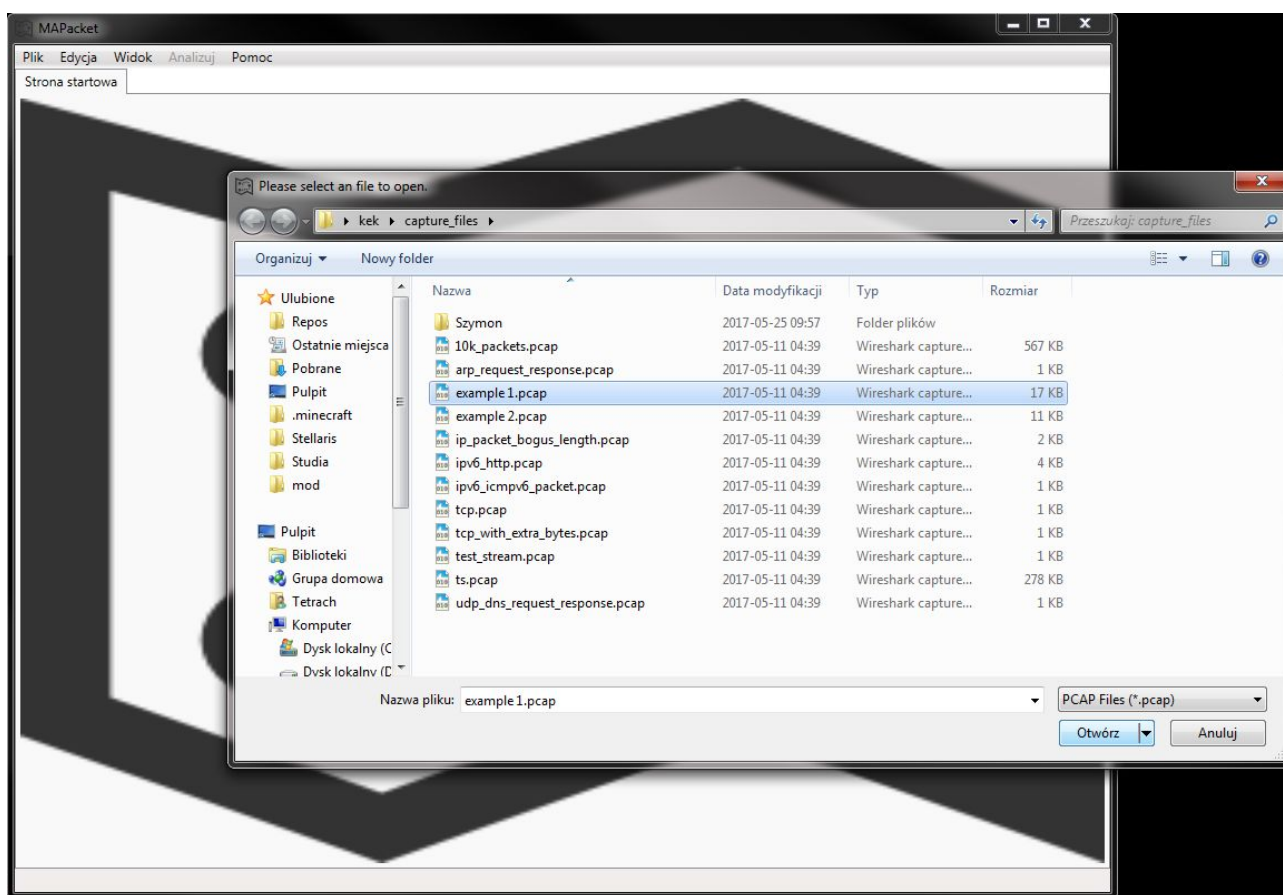


## 3.2. Wczytanie pliku pcap



*Ilustracja 5 Wybieranie opcji z menu - Otwórz*

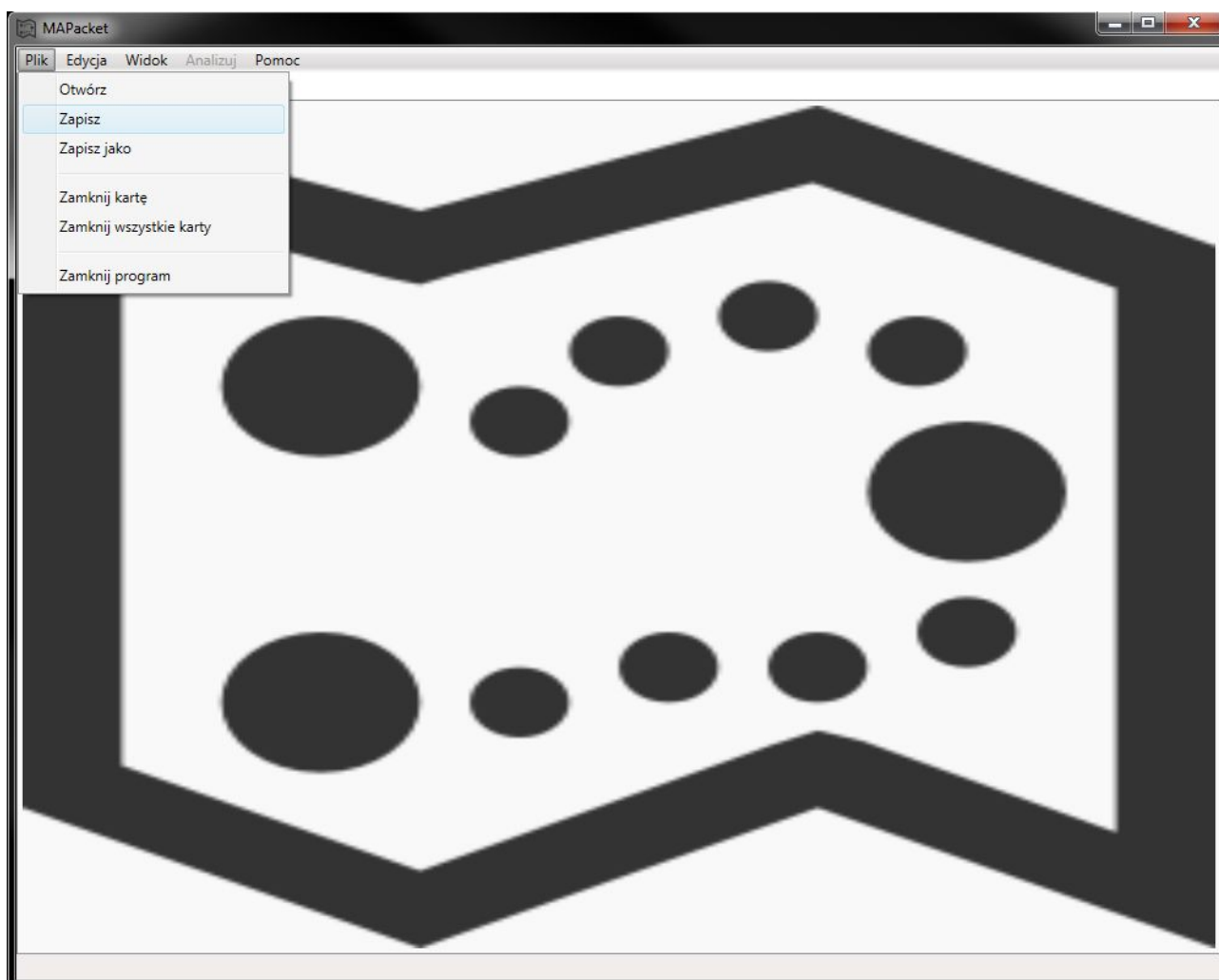
Program umożliwia wczytywanie plików pcap za pomocą opcji Otwórz. Opcja ta znajduje się w zakładce Plik.



*Ilustracja 6 Wybieranie opcji z menu - Otwórz, okno dialogowe*

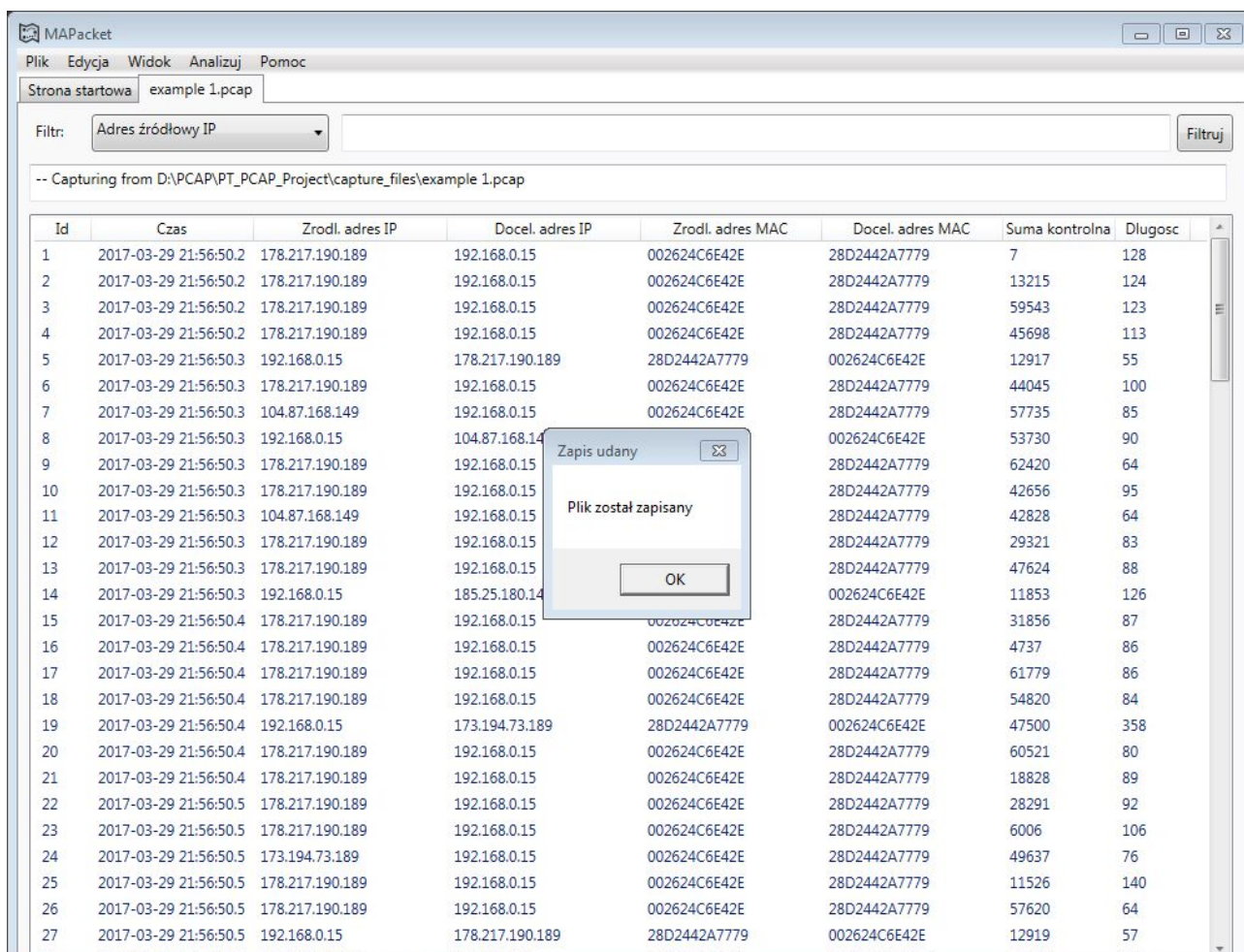
Po kliknięciu zobaczymy okno dialogowe, dzięki któremu możemy wybrać plik który zostanie wczytany. Filtr zapobiega wczytywaniu nieodpowiednich plików oraz ułatwia odnalezienie plików z odpowiednim rozszerzeniem, gdy folder zawiera dużo plików z różnymi rozszerzeniami.

### 3.3. Zapisanie plik pcap



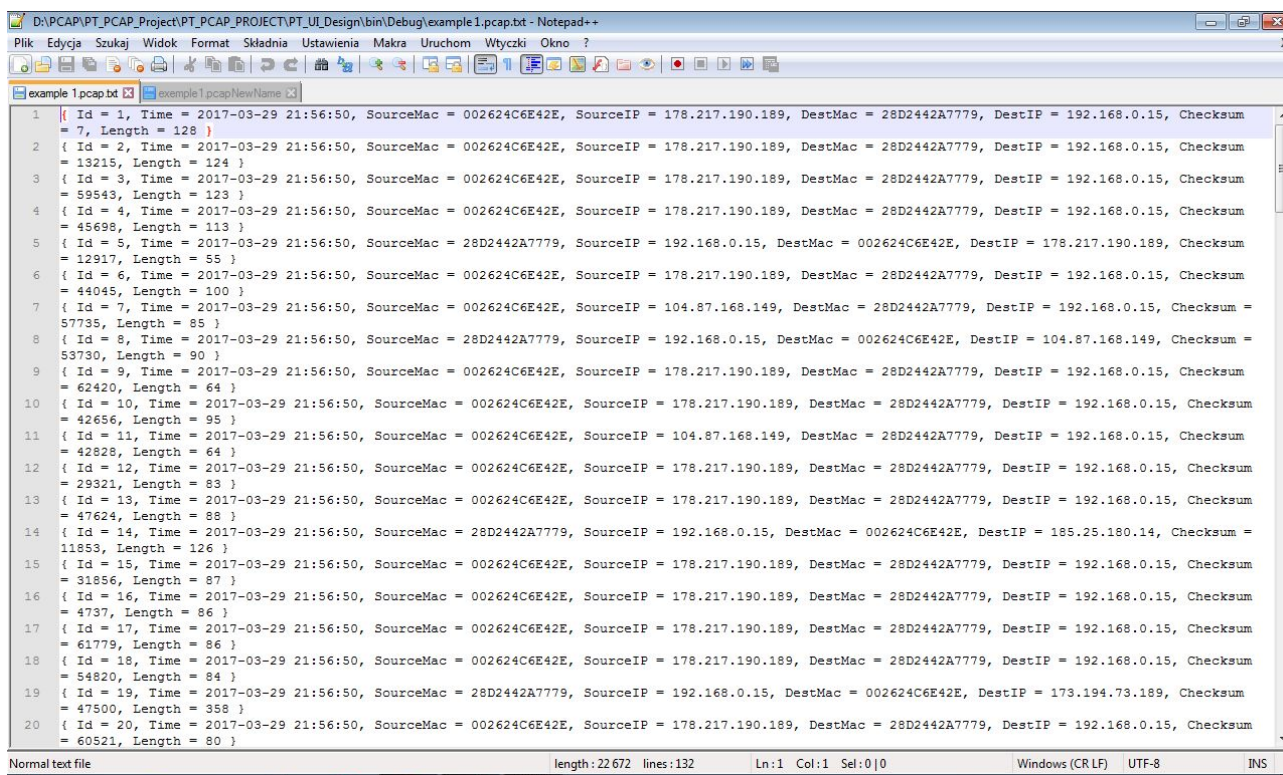
*Ilustracja 7 Wybieranie opcji z menu - Zapisz*

Po kliknięciu plik zostanie zapisany, z nazwą jaką posiada zakładka, w formacie txt. Jeżeli obiektu nie da się zapisać np. Strony Startowej, to zostaniemy poinformowani o braku możliwości zapisu. Jeżeli plik został zapisany poprawnie komunikat potwierdzi użytkownikowi wykonanie akcji.



*Ilustracja 9 Wybieranie opcji z menu - Zapisz, komunikat pozytywny*

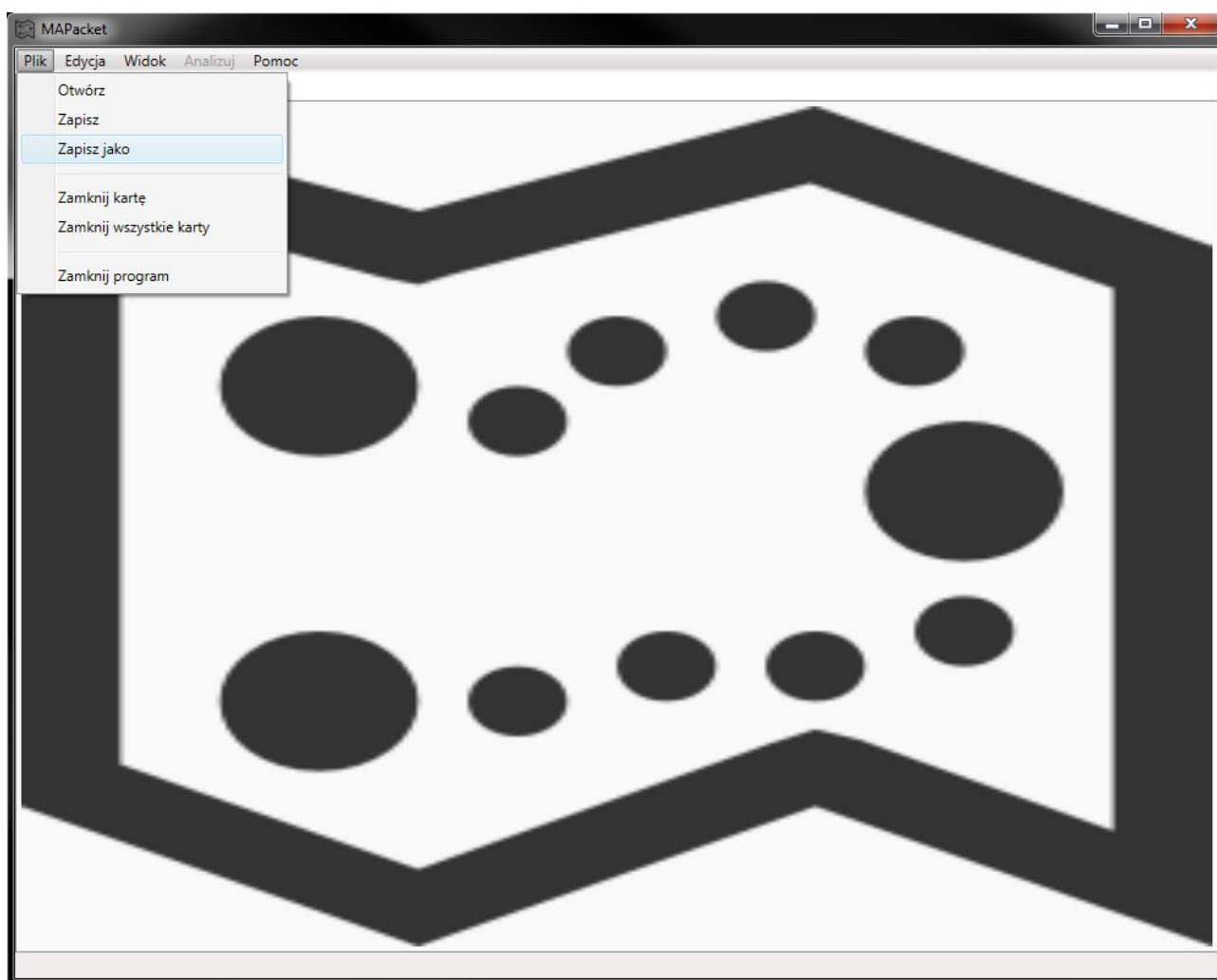
Plik example1.pcap został pozytywnie zapisany w miejscu, gdzie została uruchomiona aplikacja.



```
1 { Id = 1, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 7, Length = 128 }
2 { Id = 2, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 13215, Length = 124 }
3 { Id = 3, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 59543, Length = 123 }
4 { Id = 4, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 45698, Length = 113 }
5 { Id = 5, Time = 2017-03-29 21:56:50, SourceMac = 28D2442A7779, SourceIP = 192.168.0.15, DestMac = 002624C6E42E, DestIP = 178.217.190.189, Checksum = 12917, Length = 55 }
6 { Id = 6, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 44045, Length = 100 }
7 { Id = 7, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 104.87.168.149, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 57735, Length = 85 }
8 { Id = 8, Time = 2017-03-29 21:56:50, SourceMac = 28D2442A7779, SourceIP = 192.168.0.15, DestMac = 002624C6E42E, DestIP = 104.87.168.149, Checksum = 53730, Length = 90 }
9 { Id = 9, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 62420, Length = 64 }
10 { Id = 10, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 42656, Length = 95 }
11 { Id = 11, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 104.87.168.149, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 42828, Length = 64 }
12 { Id = 12, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 29321, Length = 83 }
13 { Id = 13, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 47624, Length = 88 }
14 { Id = 14, Time = 2017-03-29 21:56:50, SourceMac = 28D2442A7779, SourceIP = 192.168.0.15, DestMac = 002624C6E42E, DestIP = 185.25.180.14, Checksum = 11853, Length = 126 }
15 { Id = 15, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 31856, Length = 87 }
16 { Id = 16, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 4737, Length = 86 }
17 { Id = 17, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 61779, Length = 86 }
18 { Id = 18, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 54820, Length = 84 }
19 { Id = 19, Time = 2017-03-29 21:56:50, SourceMac = 28D2442A7779, SourceIP = 192.168.0.15, DestMac = 002624C6E42E, DestIP = 173.194.73.189, Checksum = 47500, Length = 358 }
20 { Id = 20, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 60521, Length = 80 }
```

Ilustracja 10 Podgląd zapisanego pliku

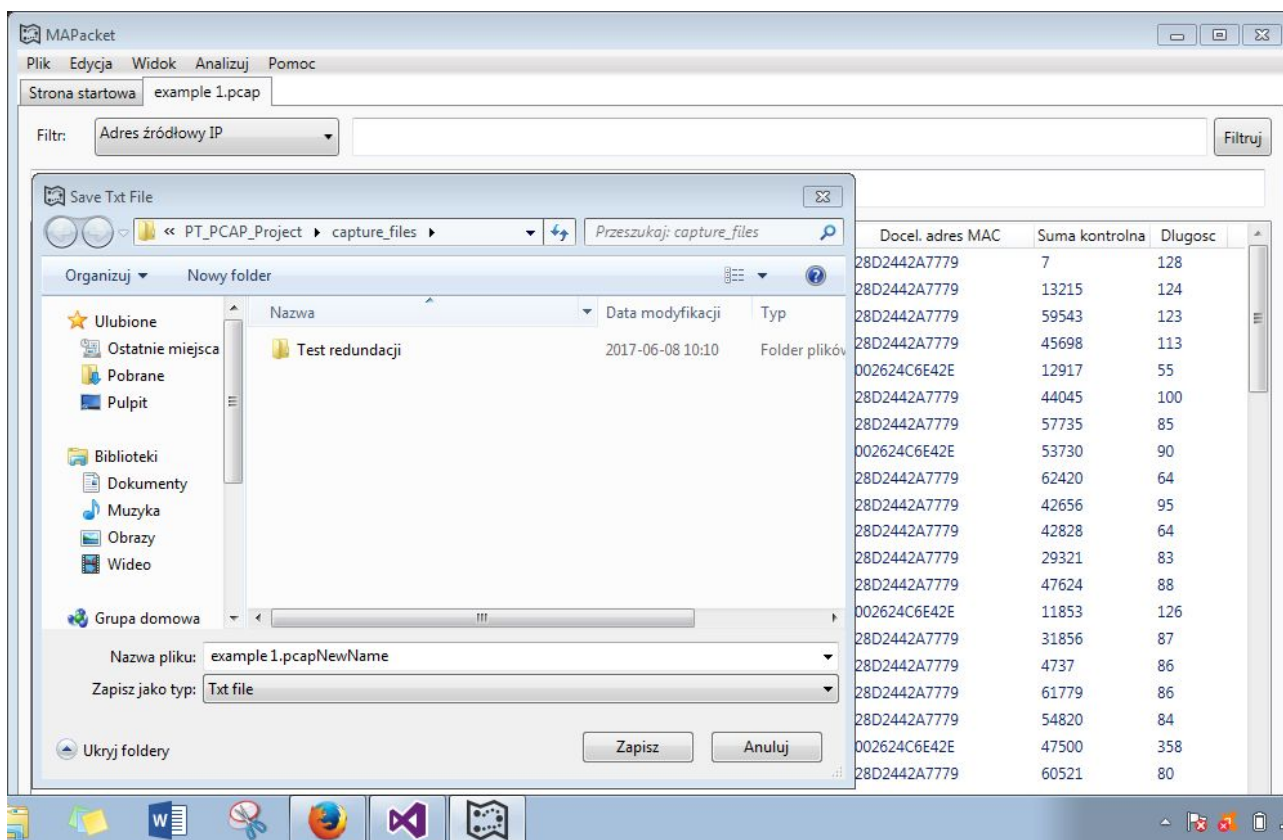
Plik example1.pcap.txt zawiera informacje o pakietach. Zdecydowaliśmy się na plik tekstowy przetwarzanie ze względu na złożoną strukturę i format PCAP.



*Ilustracja 11 Wybieranie opcji z menu - Zapisz jako*

Opcja Zapisz jako znajduje się w kategorii Plik. Pozwala ona na customizację nazwy pliku oraz jego lokalizacji.





*Ilustracja 12 Wybieranie opcji z menu - Zapisz jako, okno dialogowe*

Po otwarciu okna dialogowego możemy wybrać lokalizację pliku oraz jego nazwę. O udanym zapisie zostaniemy poinformowani przedstawionym wcześniej komunikatem.

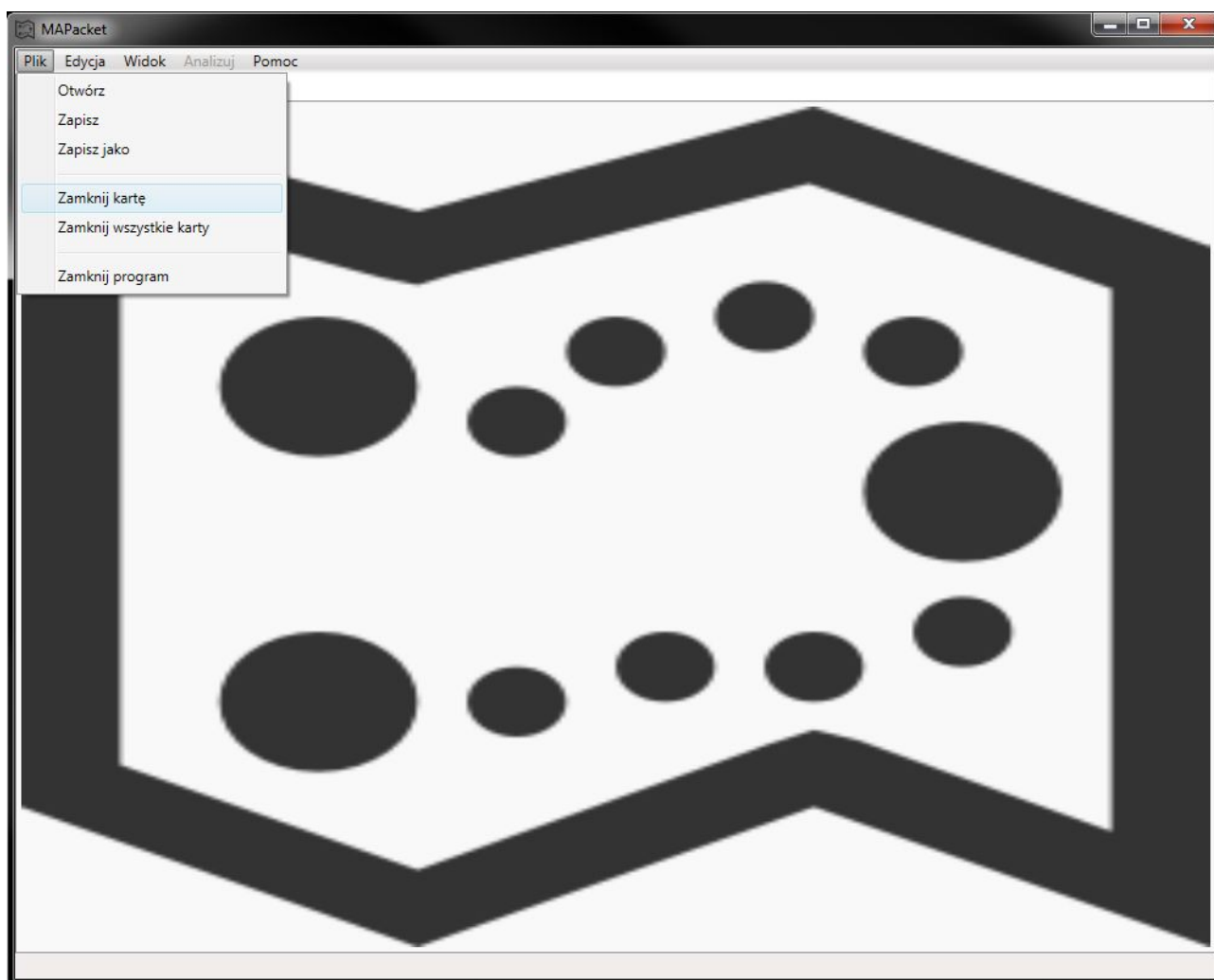
```
1 { Id = 1, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 7, Length = 128 }
2 { Id = 2, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 13215, Length = 124 }
3 { Id = 3, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 59543, Length = 123 }
4 { Id = 4, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 45698, Length = 113 }
5 { Id = 5, Time = 2017-03-29 21:56:50, SourceMac = 28D2442A7779, SourceIP = 192.168.0.15, DestMac = 002624C6E42E, DestIP = 178.217.190.189, Checksum = 12917, Length = 55 }
6 { Id = 6, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 44045, Length = 100 }
7 { Id = 7, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 104.87.168.149, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 57735, Length = 85 }
8 { Id = 8, Time = 2017-03-29 21:56:50, SourceMac = 28D2442A7779, SourceIP = 192.168.0.15, DestMac = 002624C6E42E, DestIP = 104.87.168.149, Checksum = 53730, Length = 90 }
9 { Id = 9, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 62420, Length = 64 }
10 { Id = 10, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 42656, Length = 95 }
11 { Id = 11, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 104.87.168.149, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 42828, Length = 64 }
12 { Id = 12, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 29321, Length = 83 }
13 { Id = 13, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 47624, Length = 88 }
14 { Id = 14, Time = 2017-03-29 21:56:50, SourceMac = 28D2442A7779, SourceIP = 192.168.0.15, DestMac = 002624C6E42E, DestIP = 185.25.180.14, Checksum = 11853, Length = 126 }
15 { Id = 15, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 31856, Length = 87 }
16 { Id = 16, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 4737, Length = 86 }
17 { Id = 17, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 61779, Length = 86 }
18 { Id = 18, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 54820, Length = 84 }
19 { Id = 19, Time = 2017-03-29 21:56:50, SourceMac = 28D2442A7779, SourceIP = 192.168.0.15, DestMac = 002624C6E42E, DestIP = 173.194.73.189, Checksum = 47500, Length = 358 }
20 { Id = 20, Time = 2017-03-29 21:56:50, SourceMac = 002624C6E42E, SourceIP = 178.217.190.189, DestMac = 28D2442A7779, DestIP = 192.168.0.15, Checksum = 60521, Length = 80 }
```

Ilustracja 13 Podgląd pliku

Plik example1.pcapNewName.txt posiada tą samą strukturę co podczas zapisu bez personalizacji nazwy i lokalizacji.



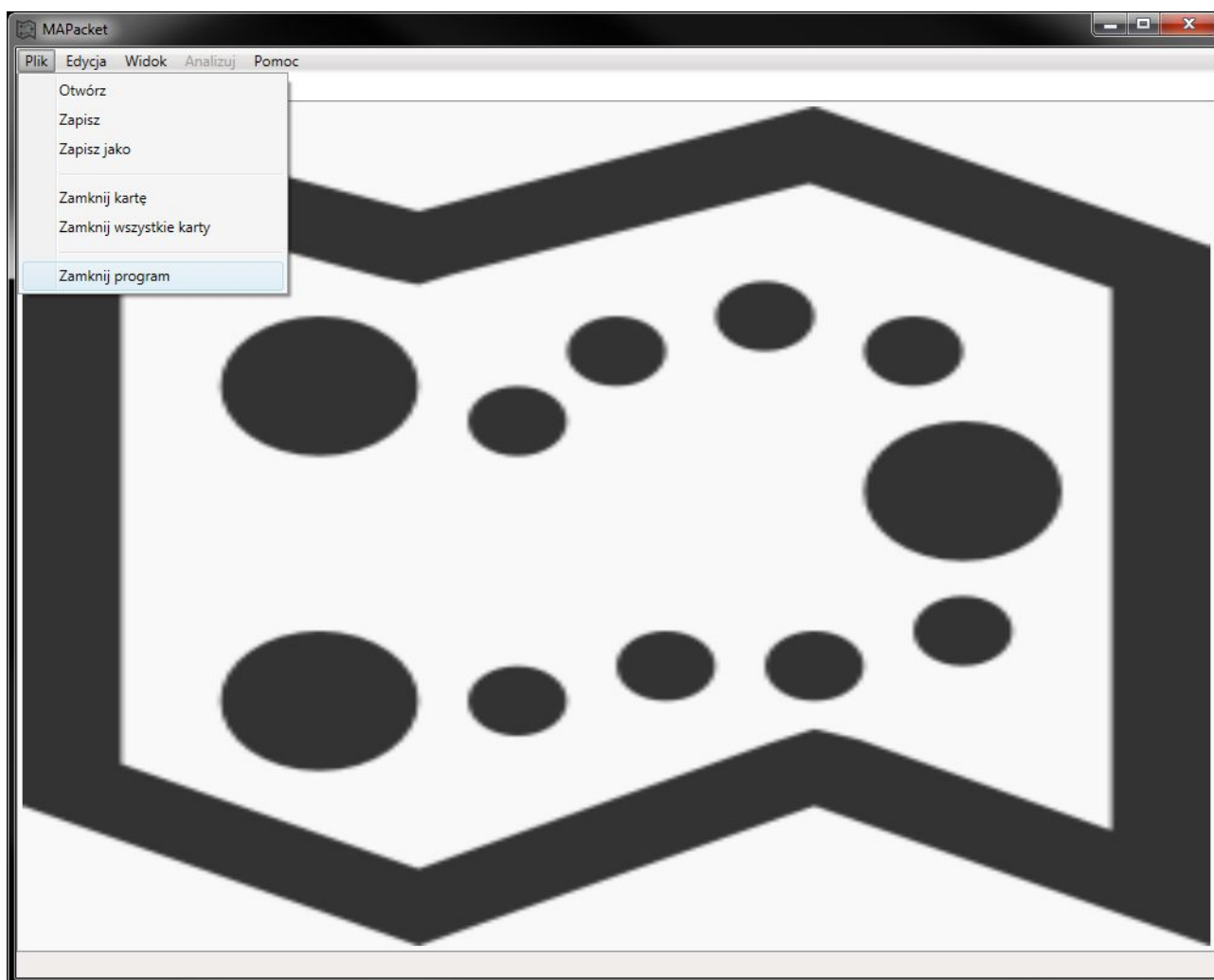
### 3.4. Zamknięcie karty i zamknięcie wszystkich kart



*Ilustracja 14 Wybieranie opcji z menu - Zamknij kartę*

Po kliknięciu w opcję Plik->Zamknij kartę zostanie zamknięta aktywna karta i system przejdzie do kolejnej dostępnej karty lub będzie przedstawiał tło programu w przypadku gdy zamknięta karta była jedyną otwartą w programie. Gdy zamykamy wszystkie karty program zamknie wszystkie karty i wyświetli tło programu

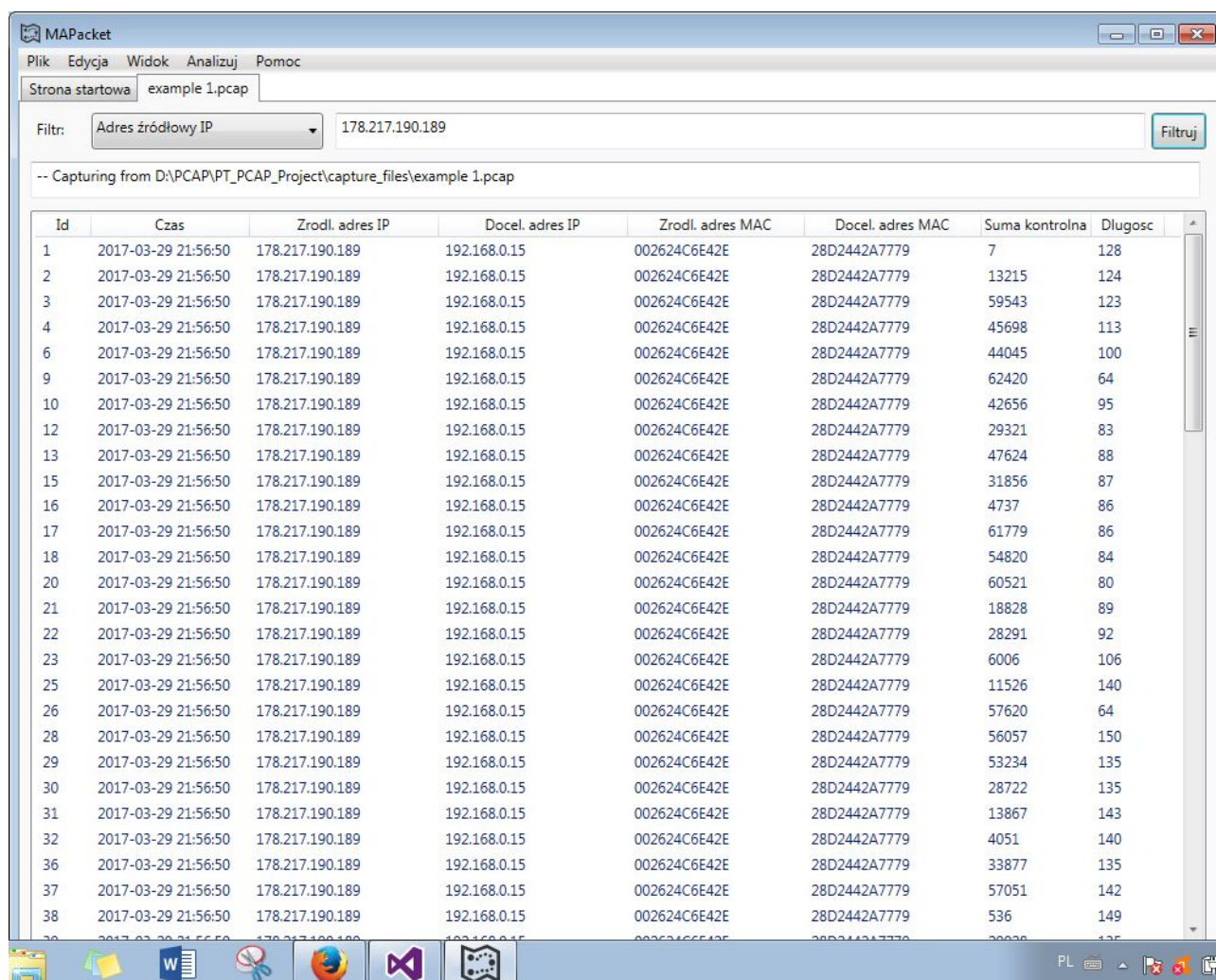
### 3.5. Zamknięcie aplikacji



*Ilustracja 19 Wybieranie opcji z menu - Zamknij program*

Opcja zamykania programu dostępna jest w kategorii Plik. Po kliknięciu program zostaje zamknięty w sposób bezpieczny a wszelkie przyznane mu zasoby zostają zwolnione.

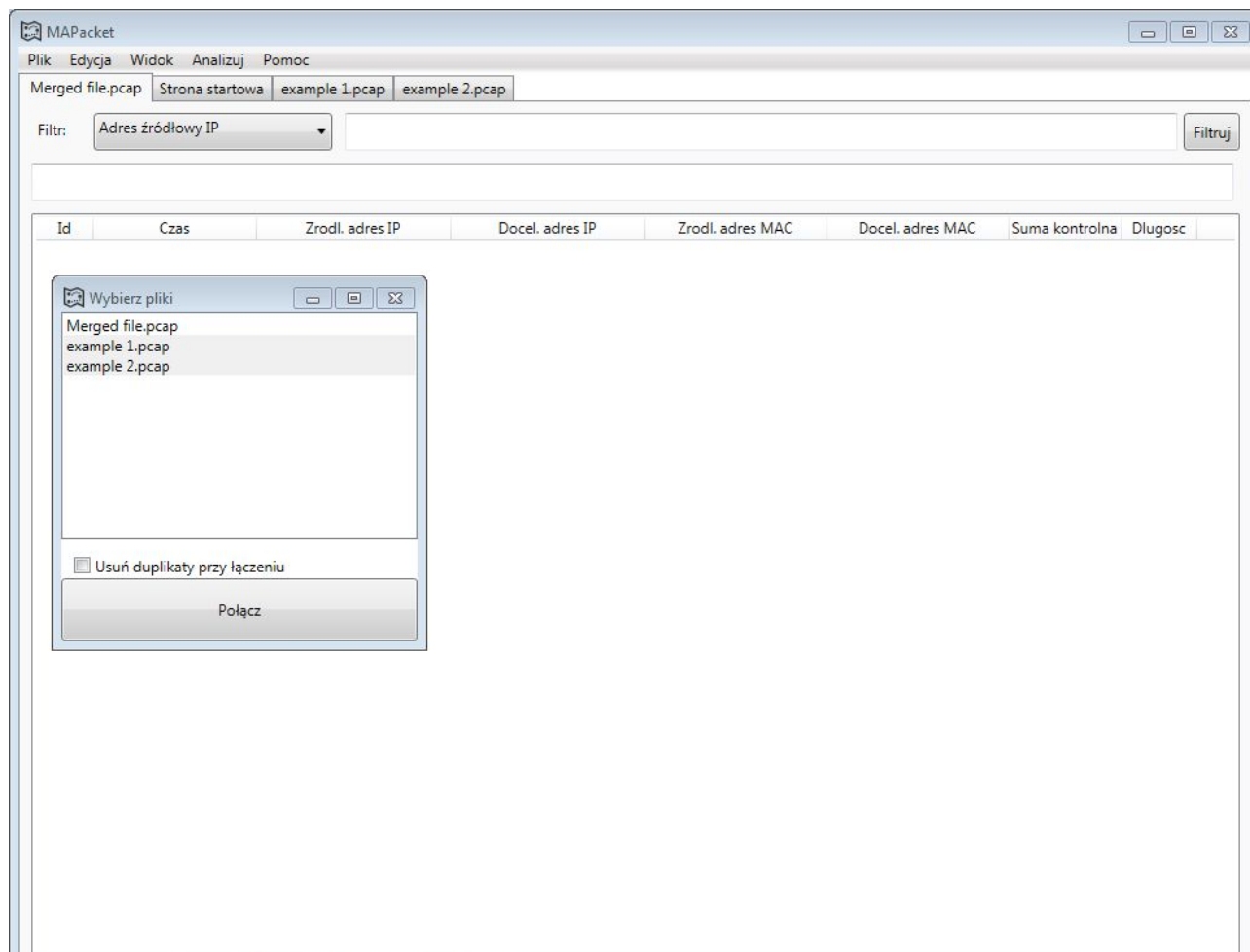
## 3.6. Filtrowanie pakietów



Ilustracja 20 Wybieranie opcji z menu - Filtrowanie

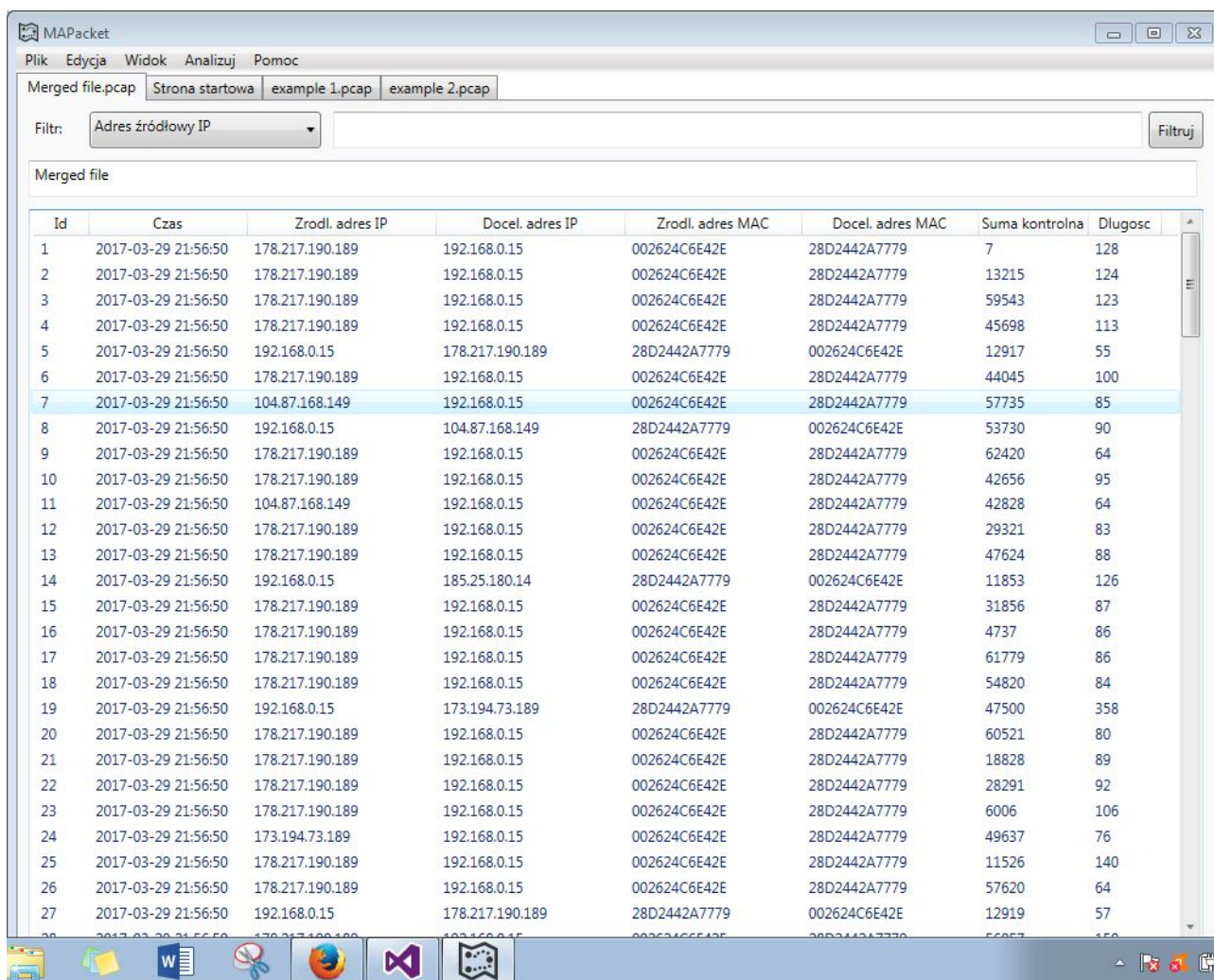
Wybieramy element po którym chcemy filtrować za pomocą listy rozwijanej. Następnie wpisujemy odpowiedni parametr np. adres IP. Po kliknięciu przycisku filtruj pakiety zostały przefiltrowane na podstawie wskazanych parametrów a wyniki zostały zaprezentowane powyżej. Usunięcie wartości filtra i zatwierdzenie Enterem przywróci stan tabeli sprzed filtracji.

### 3.7. Łączenie plików



*Ilustracja 21 Wybieranie opcji z menu - Łączenie*

Po kliknięciu przycisku opcji Edycja -> Połącz pliki pokaże nam się nowe okno z dostępnymi plikami do łączenia. Mamy możliwość wyboru czy chcemy usuwać duplikaty pakietów. Wybieramy interesujące nas pliki oraz tryb łączenia i klikamy przycisk Połącz. Jeżeli łączenie powiodło się komunikat poinformuje nas o tym a jego zamknięcie spowoduje również zamknięcie okna łączenia plików.

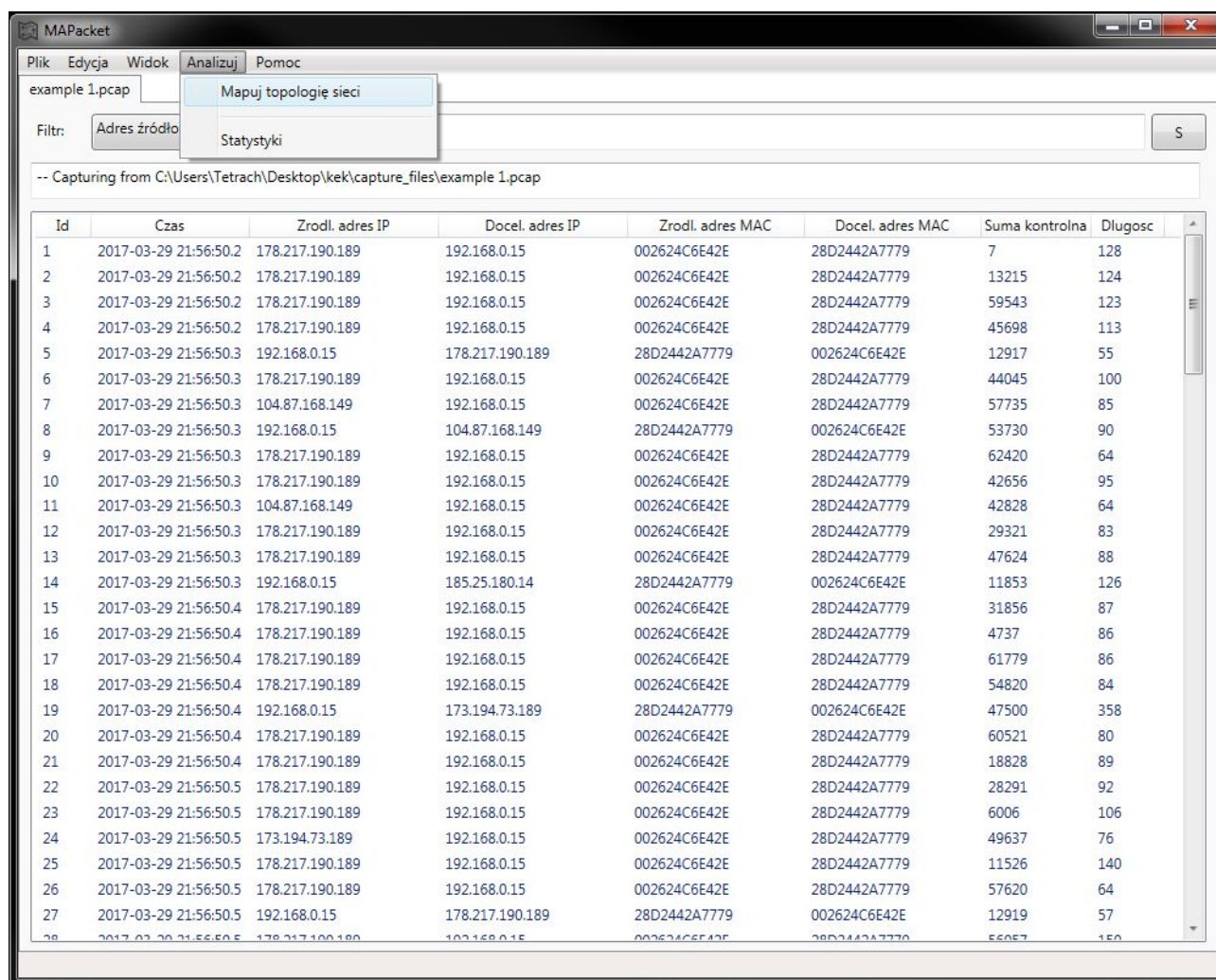


Ilustracja 22 Wybieranie opcji z menu - Łączenie

Aktywną obecnie zakładką będzie zawartość z połączonych plików pcap.



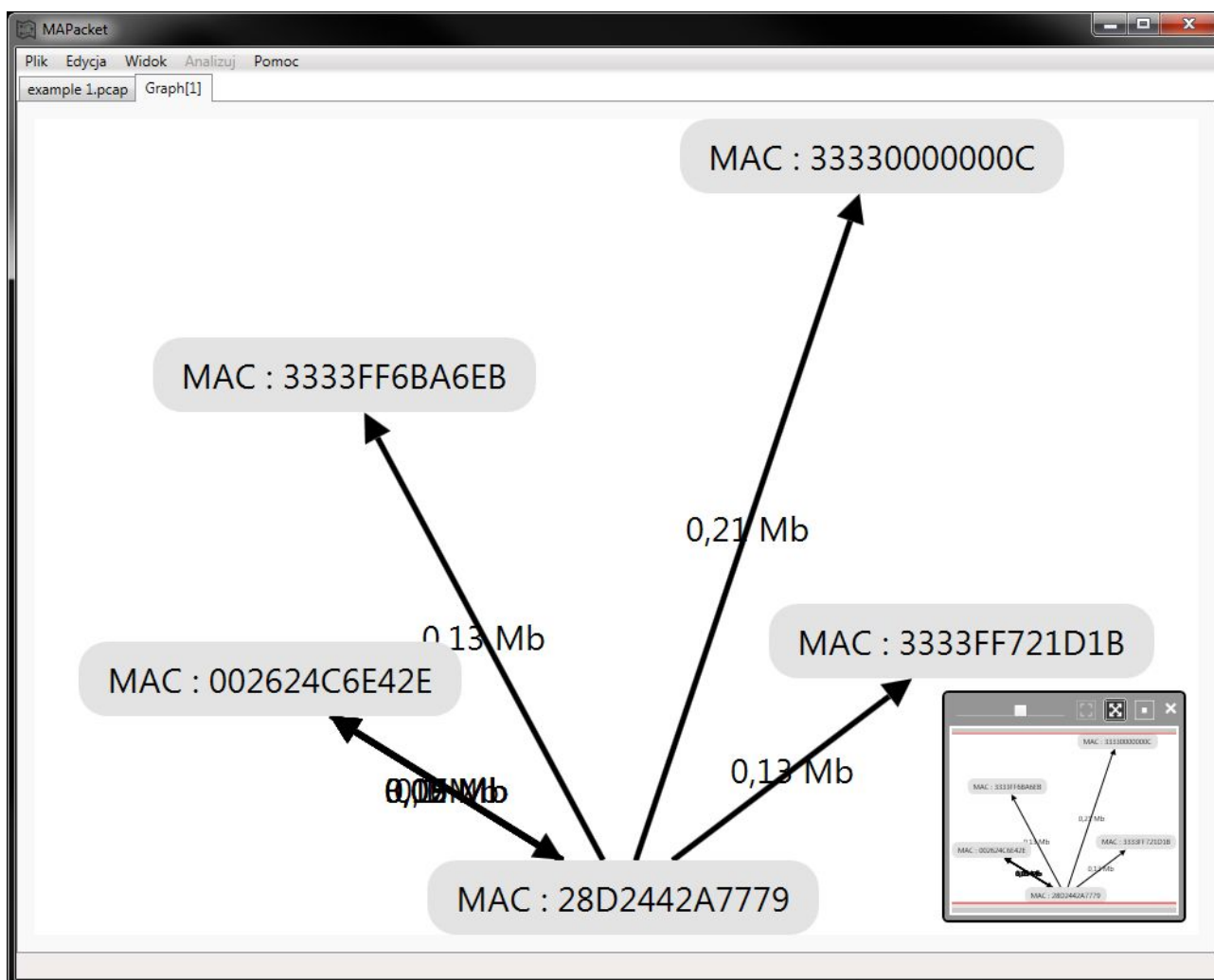
### 3.8. Generowanie grafu



Ilustracja 23 Generowanie grafu

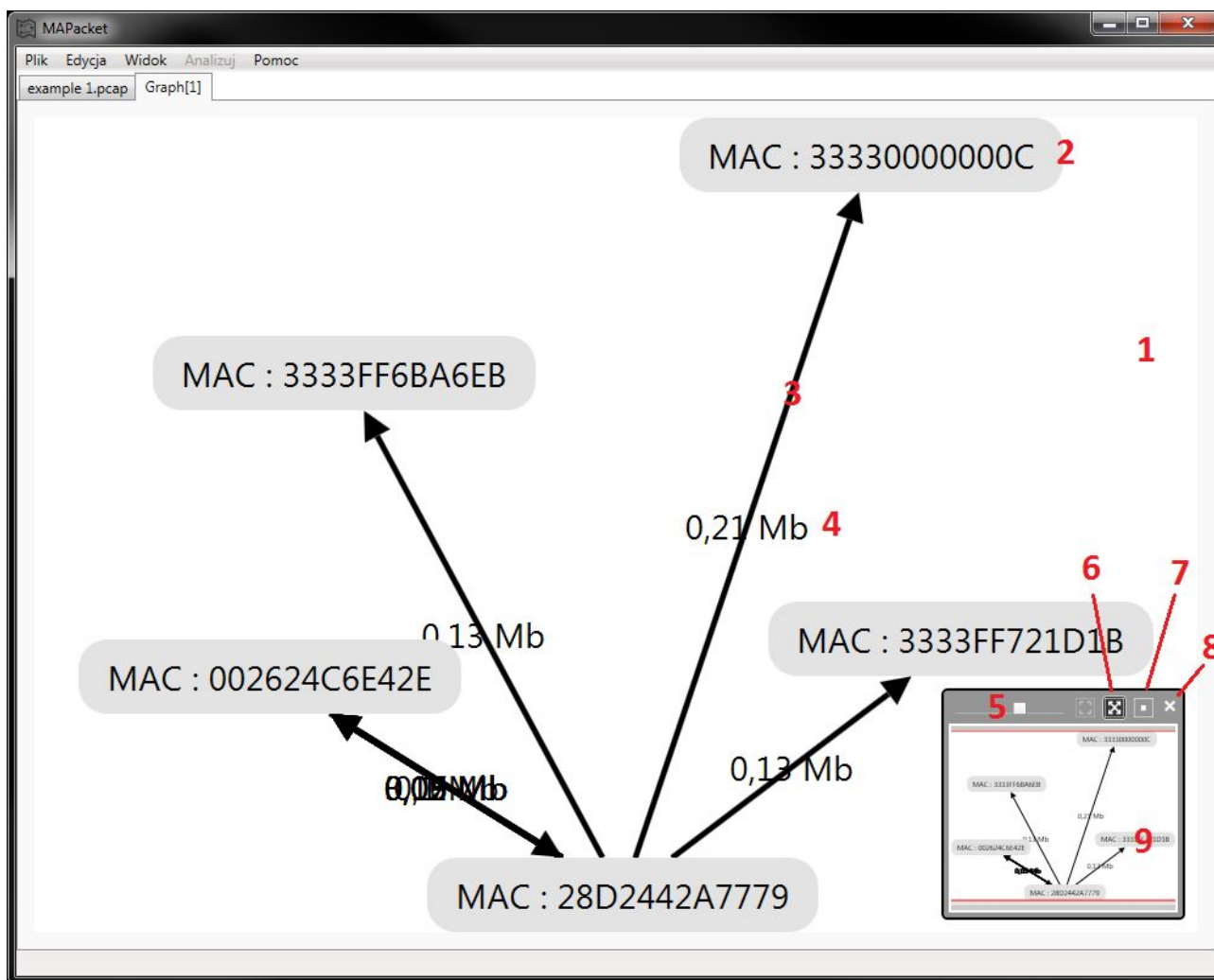
Aby wygenerować graf mapujący sieć na podstawie pakietów z pliku .pcap należy mieć otwarty plik .pcap. W przeciwnym przypadku przycisk nawigacyjny *Analizuj* będzie zablokowany. Po Wczytaniu odpowiedniego pliku w programie funkcja zostanie odblokowana, a w menu, zgodnie z ilustracją 23, znajdzie się opcja *Mapuj topologię sieci*. Po kliknięciu przycisku tej funkcji rozpocznie się generowanie grafu, który zostanie wyświetlony w nowej karcie okna głównego aplikacji.

### 3.9. Nawigowanie grafu



Ilustracja 24 Okno grafu

Po zakończeniu procesu analizy pakietów i generowania grafu zostanie on wyświetlony w formie nowej zakładki o nazwie *Graph[i]*, gdzie 'i' oznacza liczbę wygenerowanych grafów od uruchomienia programu. Efekt generowania grafu z przykładowego pliku można zaobserwować na ilustracji 24.



Ilustracja 25 Okno grafu z zaznaczonymi elementami nawigacyjnymi karty grafu

Na karcie grafu wyróżnić możemy następujące elementy interfejsu i nawigacji:

1. Płótno grafu. Pozwala przesuwając się po powierzchni grafu za pomocą strzałek lub lewego przycisku myszy, oraz skalować graf za pomocą kombinacji klawiszy CTRL+strzałki lub środkowego przycisku myszy.
2. Unikalne urządzenie działające w sieci rozpoznane za pomocą adresu MAC, które jest wierzchołkiem grafu.
3. Strzałka symbolizująca ukierunkowaną komunikację pomiędzy rozpoznanymi urządzeniami.
4. Prędkość przesyłanych danych wyliczona na podstawie liczby przesłanych bitów w czasie pomiędzy pierwszym i ostatnim komunikatem pomiędzy dwoma konkretnymi urządzeniami.
5. Suwak do skalowania grafu.
6. Maksymalne wypełnienie okna karty wygenerowanym grafem.
7. Wycentrowanie okna grafu na środku grafu.
8. Zamknięcie okna nawigacyjnego
9. Miniaturowy podgląd płótna okna grafu.



## 4. Podsumowanie

### 4.1. Podobne aplikacje

Przy projektowaniu aplikacji zespół wzorował się na różnych aplikacjach, których omówione później fragmenty stanowiły inspirację do wyglądu oraz funkcjonalności projektu. W projekcie wykorzystano takie aspekty jak wygląd interfejsu, czy funkcjonalności.

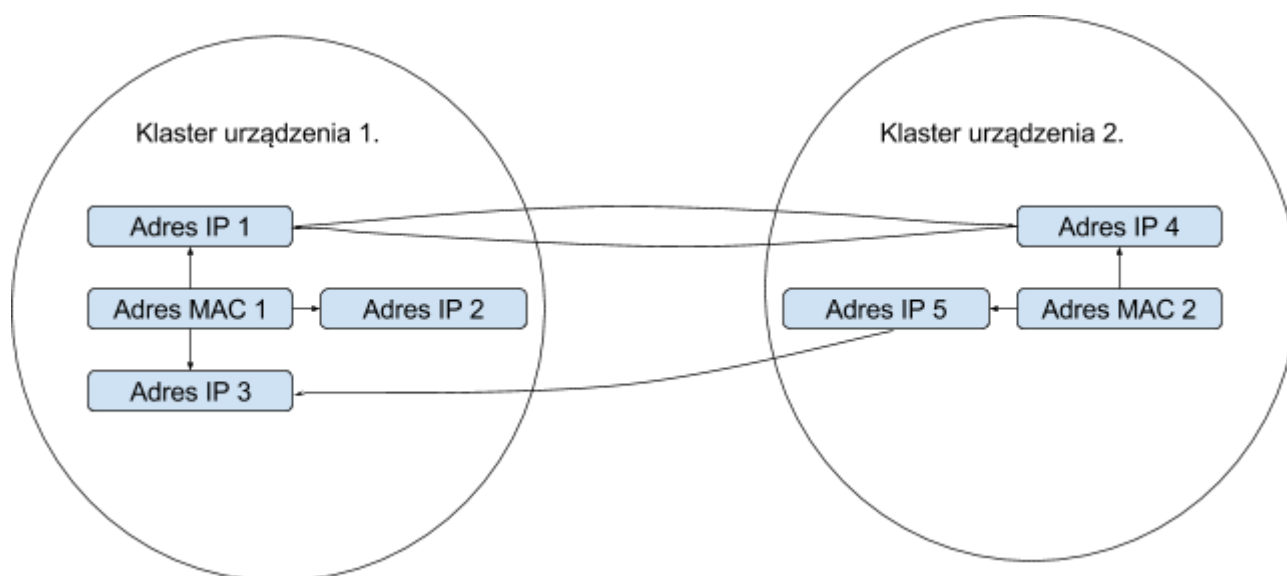
Program Wireshark, będący analizatorem protokołów sieciowych, posłużył nam jako wzór w kwestii sposobu wyświetlania danych z pakietów pochodzący z plików .pcap. Z tego programu zaczerpnęliśmy także ideę paska, w postaci pola tekstowego, służącego do filtrowania wyświetlanych danych.

Pomysł na podział otwartych plików na zakładki zaczerpnięty został z programu Visual Studio. Nie jest to program, który robi to jako pierwszy, lub jedyne, jednak stanowił nieodłączną część tego projektu oraz doświadczenia członków zespołu. Z tych względów grupa uznała to rozwiązanie za proste i eleganckie.

### 4.2. Możliwe kierunki rozwoju

Celem zespołu było zwizualizowanie sieci na podstawie dostarczonych plików pcap. Grupa skupiła się na wyświetleniu podstawowych danych, czyli adresów warstwy fizycznej i sieciowej, oraz innych danych przydatnych z punktu widzenia działania programu (długości pakietu, suma kontrolna). Program można rozwinąć o wyświetlanie innych informacji jaką można odczytać z przechwyconego pakietu przechowywanych w pliku o rozszerzeniu .pcap. Poszerzenie zasobu gromadzonych informacji zapewniłoby możliwość rozbudowy wyświetlanych i analizowanych statystyk połączeń.

Zupełnie innym aspektem naszego projektu jest wyświetlanie grafu. W przyszłości zespół mógłby zmienić bibliotekę służącą do wyświetlania grafu. W późnym etapie projektu powstała idea gromadzenia węzłów grafu w klastry, które identyfikował by unikalne urządzenia rozpoznawane po adresie fizycznym. Poniżej na schemacie 1 przedstawiono pomysł widoku na przykładową sieć. Ze względu na problemy powiązane z GraphX implementacja tego rozwiązania okazała się utrudniona i odłożona.



*Schemat 1 Proponowana wizja rozwoju grafu*

## 5. Sprawozdanie zespołu

### 5.1. Organizacja zespołu

W celu realizacji projektu zespół wykorzystał do organizacji pracy GitHub'a i Trello. Serwisy te wspierały grupę w pracy i komunikacji pomiędzy poszczególnymi członkami grupy.

GitHub jest to serwis do hostingu projektów wykorzystujący system kontroli wersji Git. pozwolił on, zgodnie z jego założeniami, na śledzenie zmian w kodzie źródłowym i łączenie zmian w edytowanych jednocześnie plikach.

Trello jest to aplikacja webowa wspierająca zarządzanie projektami. Pozwala on na tworzenie tablic, które zawierają listy kart i przypisywanie poszczególnych użytkowników do zarówno tablic, jak i kart. W celu realizacji projektu zespół spotykał się personalnie oraz częściej używał różnych komunikatorów do komunikowania się i organizacji pracy.

### 5.2. Środki implementacji

#### 5.2.1. Platformy programowania

C# WPF jest technologią stworzoną i rozwijaną przez Microsoft. Wyboru tego języka dokonaliśmy głównie z tego powodu, że każdy z członków zespołu zna ten język i posiada już pewne doświadczenie dzięki wcześniejszym projektom. Natomiast czynnikiem, który skłonił nas do skorzystania z technologii WPF był fakt, że zewnętrzna biblioteka służąca generowaniu grafów oparta jest o tą technologię i udostępnia wiele przydatnych

metod i funkcji, z których przyszło nam skorzystać. Tym samym skłoniło nas to do poszerzenia naszych umiejętności w tej platformie przy wykorzystaniu środowiska Visual Studio 2015.

### 5.2.2. Środowisko

Środowiskiem, w którym zaimplementowaliśmy aplikację było Visual Studio. Jest on dostarczany przez Microsoft i wybraliśmy je ze względu na wybór technologii firmy Microsoft. Poza tym środowisko jest bardzo przyjazne dla użytkownika i posiada wbudowany system kontroli wersji. Co znacznie ułatwiało wspólną pracę nad projektem.

### 5.2.3. Narzędzia

Używany przez nas systemem kontroli wersji był GitHub. Każdy z członków zespołu posiadał już konto w tym serwisie, System zapewnia dostęp do kodu na wielu urządzeniach, umożliwia sprawdzenie dokonanych zmian przy każdej nowej wersji kodu i tym samym łatwy wgląd do historii edycji kodu.

## 5.3. Architektura projektu

Podczas pisania projektu zespół nie realizował żadnych znanych wzorców projektów, np. MVVM. Nasz program składa się z kilku okien, z których jedno z nich, okno startowe, jest oknem głównym naszej aplikacji. Zawiera ono kontrolkę typu `TabControl`, do której dodajemy zakładki, a następnie zakładki wypełniamy nowymi oknami z naszymi funkcjonalnościami. Każda funkcjonalność naszej aplikacji znajduje się w osobnej zakładce.

### 5.3.1. Otwieranie pliku

Otwierając plik `pcap` z dysku odczytywane są z niego wszystkie ramki Ethernet i tym samym warstwa sieciowa i transportowa powiązane z tą ramką. Przeglądając plik `pcap`, z pojedynczego pakietu pobierane są z niego adres źródłowy i docelowy MAC oraz IP, znacznik czasowy, suma kontrolna i rozmiar pakietu. Pobrane informacje zapisywane są w obiektach modelu pakietu, a następnie dodawane są na listę wewnątrz kontrolki.

## 5.4. Ocena projektu i realizacji

### 5.4.1. Ukończenie projektu

Program dysponuje najważniejszymi zakładanymi przed realizacją funkcjami. Za jego pomocą można odczytywać i zapisywać pliki o rozszerzeniu `.pcap`, łączyć te pliki i usuwać redundancję pakietów, oraz mapować sieć połączeń na podstawie zgromadzonych danych. Praca nad aplikacją wymagała od zespołu nauki nowych zagadnień i zapoznania się z nowymi technologiami i bibliotekami. Jak zauważono w punkcie 4, aplikacja posiada wiele ciekawych kierunków rozwoju, które można by kontynuować. Jednak powstały projekt można uznać za solidny fundament, który można wykorzystać w tym, lub innym projekcie.

### 5.4.2. Praca zespołowa

Zadanie	Osoba
Graficzny interfejs użytkownika, filtrowanie pakietów	Szymon Kaszuba
Obsługa plików pcap, łączenie plików	Łukasz Knop
Obsługa grafu i implementacja związanych z tym modeli	Adam Matuszak

### 5.4.3. Metodyka pracy

Przyjętą metodyką pracy zespołu był Scrum. Jest to metodyka zwinna, która przewiduje, że rozwój produktu będzie podzielony na mniejsze iteracje zwane sprintami. W przypadku tego projektu sprint określony był jako okres 2 tygodnie trwających pomiędzy zajęciami. Same zajęcia laboratoryjne i prezentowane w ramach tych zajęć postępy prac na projektem były końcem jednego sprintu i początkiem kolejnego. Poszczególne sprinty były okresem na dodawanie nowych funkcjonalności aplikacji, a celem sprintu było zaprezentowanie działającej wersji programu. Praca i oczekujące zadania były rozdzielane pomiędzy poszczególnych członków zespołu. Były one najczęściej wybierane samodzielnie przez zespół bez nadzoru lidera.

## 5.5. Napotkane problemy

Zawodem okazała się biblioteka GraphX, która nie sprostała wymaganiom i oczekiwaniom zespołu. W czasie pisania projektu forum dyskusyjne biblioteki okazało się być niedostępne, a obecne w sieci materiały niewystarczające do zapewnienia odpowiedniego wsparcia dla projektu. Zapewnione algorytmy rozłożenia węzłów, oraz wykrywania krawędzi, które miały zapobiegać nakrywaniu się obiektów nie działają jak należy. Brak dostępu do dokumentacji uniemożliwia rozpoznanie sposobu na samodzielne rozkładanie węzłów według własnego algorytmu. Z tego powodu w przyszłości warto byłoby zwrócić uwagę na dokumentację biblioteki, o ile będzie dostępna, lub możliwość zamiany na inną bibliotekę graficzną.

Zespół porzucił ideę zapisywania danych, na których operuje program jako plik pcap i zdecydowaliśmy się na plik tekstowy. Trudnością okazała się złożona struktura i format pliku pcap. Dodatkowo zespół zdał sobie sprawę, że w naszym programie korzystamy z obiektów naszego modelu pakietu, który zawiera tylko wybrane przez nas pola.