

POLITECHNIKA OPOLSKA

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI I INFORMATYKI
INSTYTUT AUTOMATYKI I INFORMATYKI
KIERUNEK: INFORMATYKA



MARIUSZ HELFAJER

**Praca dyplomowa
magisterska**

ANALIZA AKTYWNOŚCI PRACY UŻYTKOWNIKÓW W SIECIACH LOKALNYCH

**Promotor:
dr inż. Krzysztof Zatwarnicki**

Spis treści

1. WPROWADZENIE	4
1.1. Cel pracy	5
1.2. Zakres pracy	5
2. APLIKACJE UMOŻLIWIAJĄCE MONITOROWANIE SIECI	7
3. PROJEKT APLIKACJI LANANALYZER	11
3.1. Diagram przypadków użycia	12
3.2. Diagram klas	32
3.3. Diagram przebiegu (sekwencji)	37
4. IMPLEMENTACJA APLIKACJI	44
4.1. Środowisko	44
4.2. Narzędzia	45
4.2.1. Aplikacje z Qt SDK	46
4.2.2. Narzędzia graficzne	48
4.2.3. Instalacja	49
4.3. Kod źródłowy	51
5. OPIS INTERFEJSU UŻYTKOWNIKA	65
5.1. Główne okno aplikacji	65
5.1.1. Pasek menu	66
5.1.2. Paski narzędziowe	73
5.1.3. Pasek stanu	73
5.1.4. Zakładki	74
5.2. Zakładka <i>Users</i>	74
5.3. Zakładka <i>Packets</i>	75
5.4. Zakładka <i>Transfers</i>	76
5.4.1. Lista <i>Top active users (uploaded)</i>	77
5.4.2. Lista <i>Top active users (downloaded)</i>	79
5.4.3. Lista <i>Upload/Download speed graph</i>	79
5.5. Zakładka <i>Applications</i>	83
5.5.1. Lista <i>Top active ports (uploaded)</i>	84
5.5.2. Lista <i>Top active ports (downloaded)</i>	85
5.5.3. Lista <i>Top active users on selected port (uploaded)</i>	86

5.5.4. Lista <i>Top active users on selected port (downloaded)</i>	87
5.6. Zakładka <i>Hosts</i>	88
5.6.1. Lista <i>Top active hosts (uploaded)</i>	90
5.6.2. Lista <i>Top active hosts (downloaded)</i>	91
5.7. Wybór urządzenia	92
5.8. Wybór filtra pakietów	95
5.9. Start przechwytywania	97
5.10. Eksport danych	100
5.11. Edycja listy aplikacji	103
6. PODSUMOWANIE	105
LITERATURA	106
SPIS RYSUNKÓW	107
DODATEK A. MATERIAŁY ZAŁĄCZONE DO PRACY	109
DODATEK B. PODRĘCZNIK UŻYTKOWNIKA APLIKACJI	110

1. Wprowadzenie

Za najważniejszą przyczynę powstania sieci komputerowych niewątpliwie można uznać potrzebę swobodnego komunikowania się oraz dzielenia danych, ale także możliwość dzielenia dostępu do programów, ułatwiona administracja systemem, centralne gromadzenia danych, a więc i ich ułatwiona archiwizacja, możliwość dzielenia sprzętu itp.

Obecnie, w XXI wieku, człowiek posługuje się codziennie mnóstwem urządzeń elektronicznych. Coraz więcej z nich daje możliwość korzystania z Internetu (np. nowe modele telefonów komórkowych czy nowe generacje telewizorów), który nie jest już luksusem, a dostęp do niego ma praktycznie każdy człowiek. Do tego w domach często znajduje się więcej niż jeden komputer. Wszystko to sprawia, że oprócz przedsiębiorstw, również w domach coraz częściej tworzy się sieci komputerowe. Przeważnie podłączone są one do Internetu. Czasami w którejś z tych sieci zachodzi potrzeba monitorowania jej ruchu czy wykorzystania jej przez użytkowników. Powody mogą być różne, czasami celem jest diagnostyka, usunięcie usterki, a innym razem sprawdzenie, ze względów bezpieczeństwa, co robią dzieci w sieci domowej czy pracownicy w sieci firmowej. Do tego celu używa się różnego rodzaju oprogramowania. Większość aplikacji jest trudna w obsłudze dla przeciętnego użytkownika, gdyż przeważnie przedstawia wyniki działania w sposób bardzo techniczny, a więc zrozumiały tylko dla specjalistów lub zaawansowanych użytkowników. O ile w dużych firmach tego typu działaniem może się zająć osoba odpowiedzialna za działanie sieci, czyli jej administrator, a następnie odpowiedni raport przedstawić szefowi, o tyle w małych firmach czy domach, gdzie takich ludzi nie ma, zająć się tym musi osoba często bez jakiegokolwiek wiedzy z tej dziedziny. Jak więc widać istnieje potrzeba tworzenia oprogramowania do monitorowania pracy użytkowników, którymi posługiwać mogliby się zwykli użytkownicy komputerów.

1.1. Cel pracy

Celem pracy magisterskiej jest opracowanie systemu umożliwiającego monitoring stanu łącza internetowego oraz pracy użytkowników w sieci lokalnej małej firmy.

Do celów szczegółowych należy opracowanie aplikacji przechwytyjącej ruch w sieci LAN firmy oraz przedstawienie wyników analizy tego ruchu sieciowego w sposób prosty i zrozumiały dla osób nieposiadających specjalistycznej wiedzy z dziedziny sieci komputerowych.

1.2. Zakres pracy

Praca podzielona została na rozdziały. W rozdziale drugim zaprezentowano obecną sytuację wśród systemów do monitorowania sieci oraz użytkowników. Przedstawiono przykładowe aplikacje służące do tego celu. Opisano ich działanie oraz najważniejsze funkcje, w podsumowaniu prezentując ich wady.

Rozdział trzeci zawiera projekt aplikacji. Przedstawiono jej zastosowanie, wymagania oraz podstawowe funkcje. Następnie, wykorzystując język UML (ang. *Unified Modeling Language* – zunifikowany język modelowania), opisano szczegółowo wymagania funkcjonalne (wykorzystując do tego celu diagram przypadków użycia), strukturę aplikacji (przedstawiając klasy wraz z atrybutami i powiązaniem między nimi) oraz interakcje między klasami aplikacji (przedstawiając za pomocą diagramu przebiegu kolejność w czasie wysyłania komunikatów pomiędzy różnymi obiektami w systemie).

W rozdziale czwartym przedstawiono implementację aplikacji. Opisano wykorzystane środowisko: WinPcapa do przechwytywania pakietów w sieci oraz Qt SDK do stworzenia aplikacji. Zrobiono również przegląd wykorzystanych narzędzi, które posłużyły do napisania kodu, przetłumaczenia tekstów aplikacji, stworzenia potrzebnych ikon i innych obrazów. Opisano także narzędzie oraz sposób utworzenia pakietu instalacyjnego aplikacji oraz omówiono najważniejsze fragmenty kodu źródłowego.

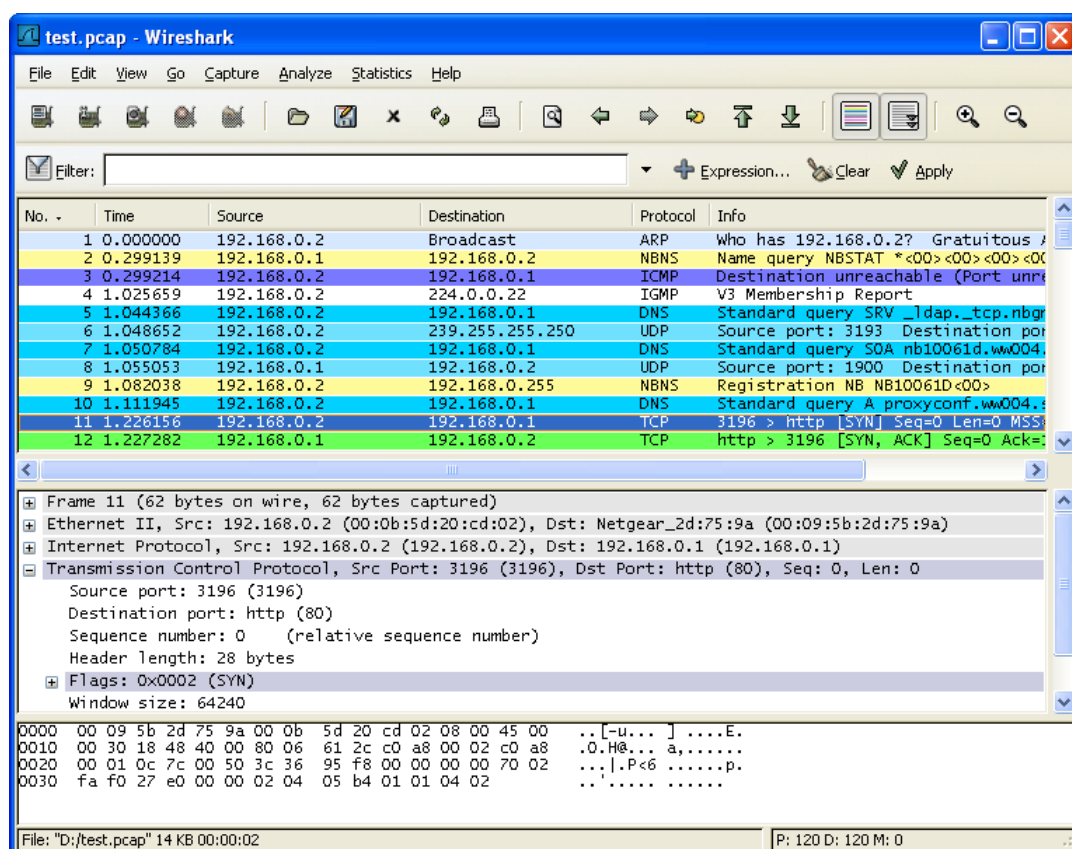
W rozdziale piątym opisano interfejs użytkownika. Przedstawiono wszystkie dostępne opcje w głównym oknie aplikacji. Pokazano jak wybrać urządzenie do przechwytywania oraz jak je uruchomić. Przedstawiono eksport danych do plików oraz jak odczytywać prezentowane przez aplikację dane o użytkownikach w sieci.

2. Aplikacje umożliwiające monitorowanie sieci

Aplikacji służących do monitorowania sieci powstało bardzo wiele. Są zarówno komercyjne, np. CommView, jak i darmowe, np. Wireshark, Karen's LAN Monitor, Axence NetTools, Network Monitor, Look@LAN. Poniżej opisano wybrane, przykładowe programy.

Wireshark

Wireshark to jedno z najbardziej popularnych narzędzi typu analizator pakietów, inaczej nazywanych analizator sieci, analizator protokołów, sniffer. Jest on dostępny na licencji GNU GPL na wiele systemów operacyjnych. Do przechwytywania pakietów wykorzystuje biblioteki pcap (dla systemu Windows jest to WinPcap). Wygląd aplikacji widoczny jest na rys. 2.1.



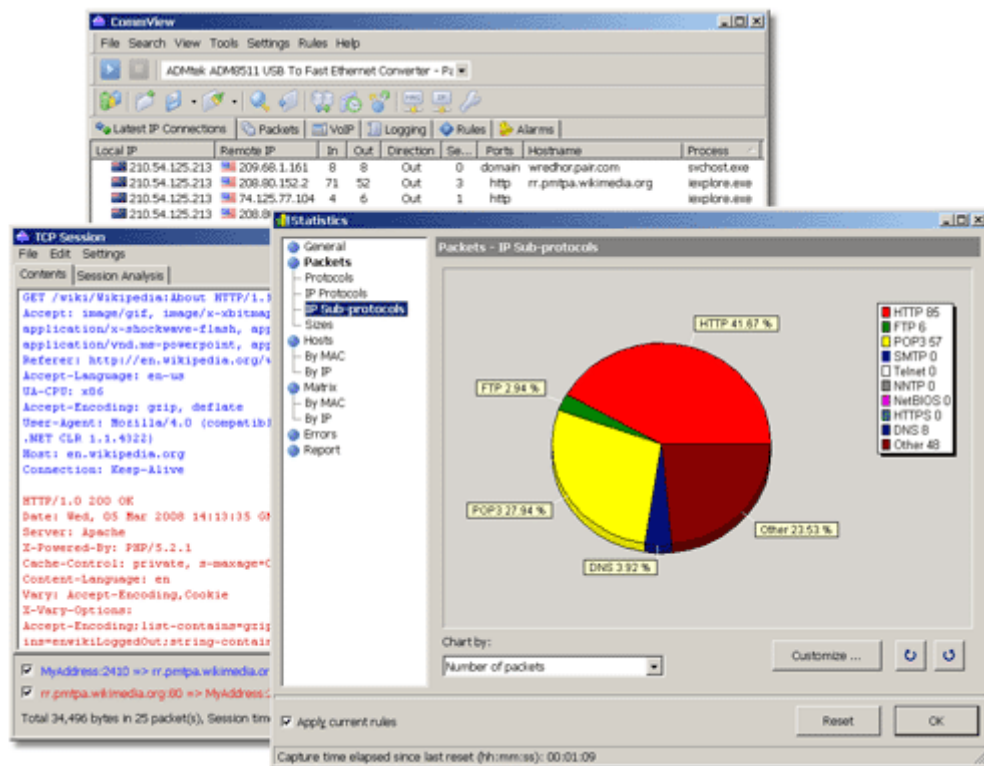
Rys. 2.1. Zrzut ekranu aplikacji Wireshark

Źródło: Wireshark User's Guide – podręcznik użytkownika aplikacji Wireshark

Najważniejsze cechy aplikacji to: wieloplatformowość, przechwytywanie pakietów z sieci w czasie rzeczywistym i wyświetlanie o nich bardzo szczegółowych danych, zapisywanie oraz otwieranie przechwyconych danych, zaawansowane filtrowanie pakietów, zaawansowane wyszukiwanie pakietów, kolorowanie wyświetlanych pakietów na podstawie filtrów, generowanie różnych statystyk, np.: hierarchia pakietów, „konwersacje” pomiędzy określonymi adresami IP, monitorowanie ruchu do i z określonego adresu IP, wykres liczby przesyłanych pakietów.

CommView

CommView (<http://www.tamos.com/products/commview/>) jest aplikacją komercyjną przeznaczoną jedynie dla systemów Windows. Wizualnie sprawia lepsze wrażenie niż Wireshark, z tego powodu może być chętniej wykorzystywana przez użytkowników. W odróżnieniu od niego potrafi jednak za pomocą dodatku CommView Remote Agent monitorować dowolny komputer na świecie podłączony do Internetu, na którym ten dodatek jest zainstalowany. Wygląd aplikacji zaprezentowano na rys. 2.2.



Rys. 2.2. Zrzut ekranu aplikacji CommView

Źródło: CommView Tutorial, <http://www.ethernet-analyzer.com/>

CommView obsługuje mniej protokołów niż Wireshark jednak posiada bardzo rozbudowane statystyki, generowanie raportów, ustawianie alarmów dla różnego rodzaju zdarzeń (np. zbyt duże obciążenie sieci).

workAgent

workAgent to produkt polski. Aplikacja dostępna jest w dwóch wersjach: komercyjnej oraz darmowej do użytku domowego. Działa jedynie na systemach Windows. Składa się z dwóch modułów. Moduł workAgent Monitor wymaga zainstalowania na każdym z komputerów, które chcemy monitorować. Program ten jest całkowicie niewidoczny dla użytkownika, a jego praca polega na zbieraniu informacji o jego działaniach, między innymi o odwiedzanych stronach WWW. Drugi z modułów instalujemy na komputerze, z którego będziemy łączyć się z komputerami monitorowanymi. Daje nam to możliwość podglądu działań na tych komputerach w czasie rzeczywistym, a także analizowanie danych zebranych przez moduły monitorujące. Wygląd przykładowego raportu pokazano na rys. 2.3.

Zestawienia

Zest.: Raport z odwiedzanych stron WWW z: Wczoraj

na: Domowy dla: Wszystkich użytkowników

Raport z odwiedzonych stron WWW

Domena	Czas aktywności		Tytuł
	Strona WWW	Użytkownik	
www.haftowanke.fora.pl	01:05:42	00:50:06	Prywatne Wiadomości - Haft Krzyżkowy - Mozilla Firefox
www.4shared.com	00:04:34	00:09:16	4shared.com - free file sharing and storage - Mozilla Firefox
index.jsp	00:00:02	00:00:02	Mozilla Firefox
account/home.jsp?sessionId=77838876CAF3DF10562F7...	00:02:18	00:01:00	4shared - free file sharing and storage - share folder - My 4shared - Mozilla Firefox
account/home.jsp	00:01:25	00:01:25	4shared - free file sharing and storage - share folder - My 4shared - Mozilla Firefox
account/file.jsp?id=25543274&id=DQ8dUhmQedxCLZVW	00:00:08	00:00:08	File Properties - Mozilla Firefox
www.fotosik.pl	00:02:01	00:01:40	picture.jpg - Rozmiar oryginalny - Fotosik.pl - Mozilla Firefox
show.7ya.ru	00:01:33	00:01:11	7я.ру Фотоальбомы участников. / - Mozilla Firefox
mampka2.multiply.com	00:00:59	00:00:29	Marty's Bell - Photos - Mozilla Firefox
multiply.com	00:00:51	00:00:25	Multiply - Mozilla Firefox
streamphoto.ru	00:00:31	00:00:15	СТРИМ.Фото Понзосателн Drk_ness Животные_2 Bears of Duckpont.jpg - Mozilla Firefox
www.streamphoto.ru	00:00:24	00:00:12	СТРИМ.Фото Понзосателн Olester Teddy Bear Times - Stoney Creek - Mozilla Firefox
www.onet.pl	00:00:23	00:00:23	Onet.pl - Polski Portal Internetowy - Mozilla Firefox
strimfoto.ru	00:00:14	00:00:12	Nie znaleziono obiektu! - Mozilla Firefox
marqaios.multiply.com	00:00:12	00:00:06	marina's Site - marina's Photos - Mozilla Firefox
W sumie:			01:17:30 00:58:20

Rys. 2.3. Zrzut ekranu raportu odwiedzanych stron WWW z aplikacji workAgent

Źródło: strona WWW aplikacji, <http://www.workagent.info/>

Na podstawie zaprezentowanych aplikacji można zauważyć, że istnieją ich dwa rodzaje. Niektóre wymagają instalacji dodatkowego oprogramowania na komputerach, które mają być monitorowane. Czasami zdarza się, że to oprogramowanie jest rozpoznawane, przez programy antywirusowe, jako szkodliwe dla użytkownika. Komplikuje więc to korzystanie z takich aplikacji. Inne natomiast potrafią monitorować ruch w sieci z jednego miejsca, jednak w tym przypadku są one mniej czytelne (statystyki są bardziej techniczne) i mniej wygodne dla zwykłego użytkownika.

W statystykach opisanych aplikacji brakuje jednak ważnych danych, brak porównania aktywności (np. liczby i rodzaju odwiedzanych stron, liczby przesyłanych danych) monitorowanych użytkowników oraz ich wpływu na całość sieci.

Z powodu niedostatków aplikacji istniejących na rynku postanowiono zająć się tym tematem i stworzyć własną aplikację, która łącząc wyżej wymienione cechy, pozwoli na łatwe i wygodne monitorowanie działań użytkowników z jednego miejsca w sieci, bez instalowania dodatkowych programów na ich komputerach.

3. Projekt aplikacji LANAnalyzer

Aplikacja LANAnalyzer to program służący do monitorowania pracy użytkowników w sieciach lokalnych. Aplikacje tego typu znajdują zastosowanie głównie w firmach, gdzie pracodawcy z różnych powodów chcą mieć kontrolę nad tym, co robią ich pracownicy. Innym zastosowaniem może być wspomaganie pracy administratorów sieci w celu diagnozowania oraz usuwania problemów.

LANAnalyzer wymaga instalacji jedynie na jednym z komputerów w sieci. Istotne jest, aby program został uruchomiony na komputerze, do którego dochodzą wszystkie pakiety z danej sieci. Ważne jest to zwłaszcza w nowych sieciach komputerowych opartych na przełącznikach (ang. *switch*), które przekazują pakiety tylko do adresata pakietu, odwrotnie niż robią to koncentratory (ang. *hub*), gdzie pakiety od nadawcy trafiają do wszystkich osób w sieci. W przypadku pierwszym, czyli sieci zbudowanej na przełącznikach, ważne jest, aby był to przełącznik zarządzalny, gdzie istnieje możliwość przekierowania całego ruchu w sieci na wybrany port, do którego można by podłączyć komputer z tym programem.

Aplikacja ma za zadanie przechwytywanie całego ruchu sieciowego (lub tylko konkretnych, wcześniej zdefiniowanych pakietów) oraz jego analizę, aby w sposób czytelny przedstawić to, co się dzieje w sieci oraz w jaki sposób wykorzystują ją użytkownicy. Ważną kwestią jest możliwość eksportu danych. Dane są zapisywane w postaci typowych i bardzo popularnych plików tekstowych CSV (ang. *Comma Separated Values* – wartości rozdzielone przecinkiem). Wykresy natomiast można zapisywać w plikach graficznych w wielu dostępnych i popularnych formatach.

Po pierwszym uruchomieniu aplikacji jej użytkownik najpierw powinien wybrać odpowiednie urządzenie sieciowe do przechwytywania pakietów. Następnie może zdefiniować filtr pakietów w celu przechwytywania tylko konkretnych ich rodzajów. Uruchomienie przechwytywania może nastąpić na jeden z dwóch sposobów. Pierwszy natychmiast uruchamia bezterminowe przechwytywanie z domyślnymi opcjami sterownika. Drugi wyświetla okno dialogowe z opcjami startu, zatrzymania przechwytywania jak również umożliwia zmianę opcji sterownika. W trakcie przechwytywania oraz po jego zakończeniu można przeglądać wiele z dostępnych informacji zarówno o użytkownikach jak i o całej sieci.

Do stworzenia projektu aplikacji został użyty język UML (ang. *Unified Modeling Language*), czyli zunifikowany język modelowania. Służy on do modelowania głównie systemów informatycznych i stanowi metodologię projektowania oraz jego zapisu. Jest ustandaryzowanym procesem ułatwiającym tworzenie rozbudowanych aplikacji oraz ich implementację. Poniżej przedstawiono jedynie elementy projektu UML, a do stworzenia projektu zostały wykorzystane jedne z najczęściej używanych rodzajów diagramów:

- **Diagram przypadków użycia** – spojrzenie na system z punktu widzenia użytkownika, w celu określenia wymagań i funkcjonalności.
- **Diagram klas** – jest widokiem implementacyjnym systemu określającym budowę klas (właściwości i metody) i ich powiązania.
- **Diagram przebiegu (sekwencji)** – opisuje kolejność w czasie wysyłania komunikatów pomiędzy różnymi obiektami w systemie w celu zapewnienia określonej funkcjonalności.

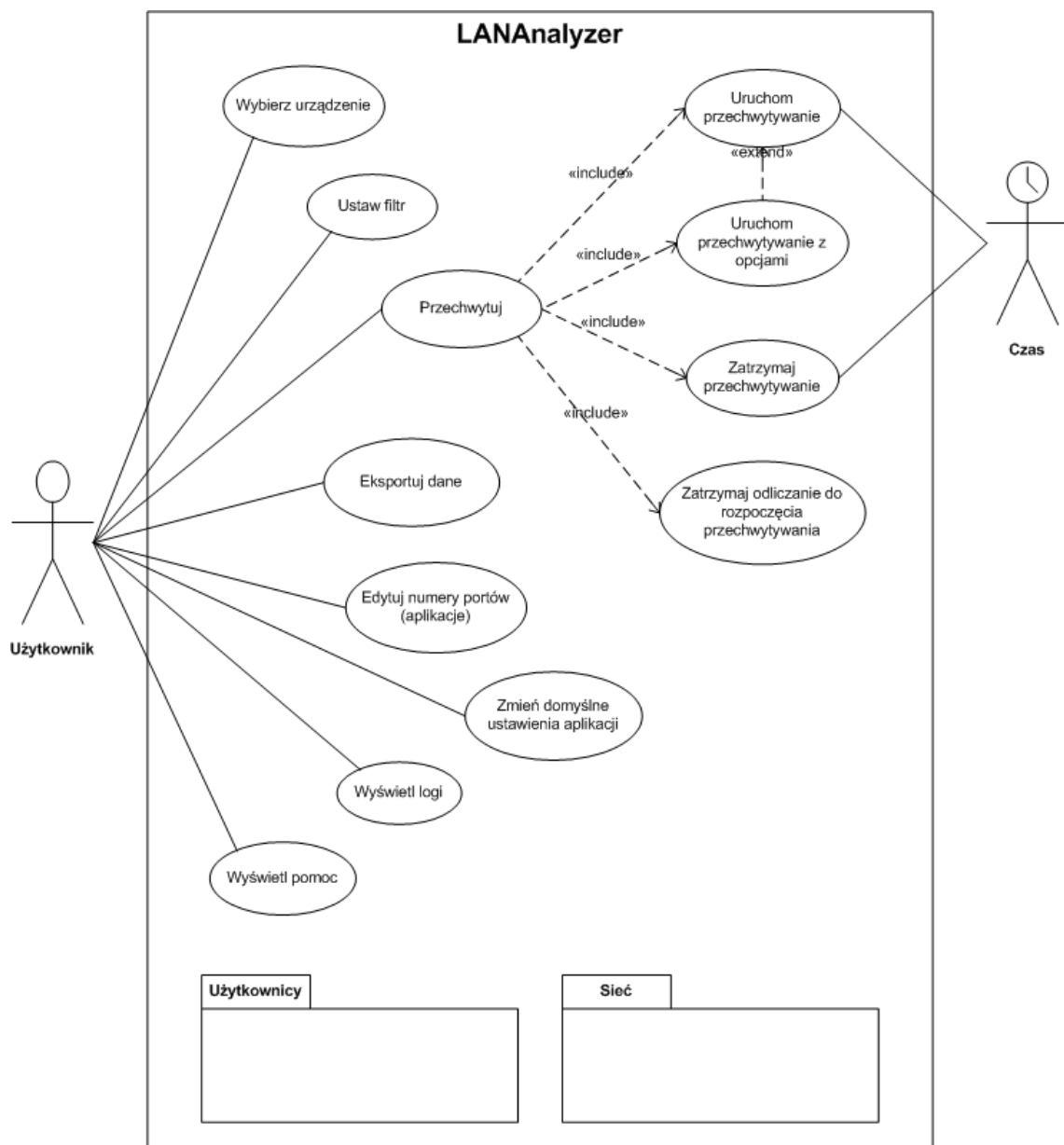
Przedstawione diagramy UML zostały stworzone w Microsoft Visio Professional 2007, w którym wykorzystano *Visio Stencil and Template for UML 2.2* – wzornik (kolekcja kształtów) wraz z szablonem obsługującym UML, aż do aktualnej wersji 2.2 włącznie. Dodatek pochodzi ze strony <http://softwarestencils.com/> i zgodnie z licencją można go dowolnie wykorzystywać w celach prywatnych.

[2, 4, 5, 8]

3.1. Diagram przypadków użycia

Diagram przypadków użycia z rys. 3.1. oprócz przypadków użycia zawiera również dwa pakiety, które grupują inne przypadki użycia w celu zwiększenia czytelności diagramu. Pakiety przedstawione zostały na kolejnych rysunkach. Na poniższych diagramach przypadków użycia istnieją dwa typy aktorów:

- **Użytkownik** – to aktor osobowy, użytkownik aplikacji, osoba, która będzie ją obsługiwała, np.: administrator sieci, szef w danej firmie.
- **Czas** – to aktor nieosobowy, *timer* w aplikacji, który może uruchomić lub zatrzymać przechwytywanie.



Rys. 3.1. Diagram przypadków użycia aplikacji

Przypadek użycia „Uruchom przechwytywanie”:

1. Uczestniczący aktorzy
 - Użytkownik
 - Czas
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym lub aktor Czas uruchamia odpowiednią funkcję.

- Aplikacja uruchamia przechwytywanie oraz informuje o stanie na pasku stanu.
3. Alternatywny ciąg zdarzeń
 - Brak
 4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Informacja o stanie na pasku stanu.
 - Informacje o aktywności użytkowników w sieci wyświetlane na zakładkach.

Opis przypadku użycia „Uruchom przechwytywanie z opcjami”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno dialogowe z opcjami uruchomienia przechwytywania.
 - Użytkownik wybiera moment startu, czas przechwytywania oraz zaawansowane opcje sterownika.
 - Użytkownik potwierdza uruchomienie przechwytywania klikając na **Start**.
 - Aplikacja realizuje zadanie oraz informuje o stanie na pasku stanu.
3. Alternatywny ciąg zdarzeń
 - Użytkownik rezygnuje z uruchomienia przechwytywania klikając na **Cancel**.
 - Aplikacja zamyka okno dialogowe.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Informacja o stanie na pasku stanu.
 - Informacje o aktywności użytkowników w sieci wyświetlane na zakładkach.

Przypadek użycia „Zatrzymaj przechwytywanie”:

1. Uczestniczący aktorzy
 - Użytkownik
 - Czas
2. Podstawowy ciąg zdarzeń

- Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym lub aktor Czas uruchamia odpowiednią funkcję.
 - Aplikacja zatrzymuje przechwytywanie oraz informuje o stanie na pasku stanu.
3. Alternatywny ciąg zdarzeń
 - Brak
 4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Informacja o stanie na pasku stanu.

Przypadek użycia „Zatrzymaj odliczanie do rozpoczęcia przechwytywania”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja zatrzymuje odliczanie do rozpoczęcia przechwytywania oraz informuje o stanie na pasku stanu.
3. Alternatywny ciąg zdarzeń
 - Brak
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Informacja o stanie na pasku stanu.

Przypadek użycia „Wybierz urządzenie”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno dialogowe z lista urządzeń sieciowych mogących służyć do przechwytywania danych.
 - Użytkownik zaznacza wybrane urządzenie.
 - Użytkownik potwierdza wybór klikając na **Ok**.
 - Aplikacja zmienia urządzenie na nowe oraz informuje o tym na pasku stanu.

3. Alternatywny ciąg zdarzeń

- Użytkownik rezygnuje ze zmiany urządzenia klikając na **Cancel**.
 - Aplikacja zamyka okno dialogowe.
- Użytkownik wybiera aktualnie wybrane urządzenie.
 - Użytkownik klika na **Ok**.
 - Aplikacja zamyka okno dialogowe.
- Użytkownik wybiera inne urządzenie w trakcie przechwytywania.
 - Użytkownik klika na **Ok**.
 - Aplikacja wyświetla okno dialogowe z informacją o braku możliwości zmiany w trakcie działania przechwytywania.
 - Aplikacja zamyka okno dialogowe.

4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia

- Informacja o wybranym urządzeniu na pasku stanu.

Przypadek użycia „Ustaw filtr”:

1. Uczestniczący aktorzy

- Użytkownik

2. Podstawowy ciąg zdarzeń

- Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
- Aplikacja wyświetla okno dialogowe z listą przykładowych filtrów z możliwością ich edycji.
- Użytkownik wybiera filtr.
- Użytkownik klika na **OK**.
- Aplikacja ustawia nowy filtr i wyświetla jego nazwę na pasku stanu.

3. Alternatywny ciąg zdarzeń

- Użytkownik rezygnuje ze zmiany filtra klikając na **Cancel**.
 - Aplikacja zamyka okno dialogowe.
- Użytkownik wybiera aktualnie wybrany filtr.
 - Użytkownik klika na **Ok**.
 - Aplikacja zamyka okno dialogowe.
- Użytkownik wybiera inny filtr w trakcie przechwytywania.
 - Użytkownik klika na **Ok**.

- Aplikacja wyświetla okno dialogowe z informacją o braku możliwości zmiany w trakcie działania przechwytywania.
 - Aplikacja zamyka okno dialogowe.
- 4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Nazwa filtra jest wyświetlana na pasku stanu.

Przypadek użycia „Eksportuj dane”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno dialogowe opcjami eksportu.
 - Użytkownik zmienia odpowiednie opcje, wybiera folder docelowy.
 - Użytkownik klika na **Export**.
 - Aplikacja zapisuje wybrane dane we wskazanym miejscu w odpowiednich plikach.
 - Aplikacja wyświetla okno podsumowania eksportu.
3. Alternatywny ciąg zdarzeń
 - Użytkownik rezygnuje z eksportu klikając na **Cancel**.
 - Aplikacja zamyka okno dialogowe.
 - Użytkownik wybiera nieprawidłowy folder.
 - Użytkownik klika na **Export**.
 - Aplikacja wyświetla informację o nieprawidłowym folderze i wraca do okna dialogowego (w celu umożliwienia zmiany foldera).
 - Użytkownik nie zaznacza żadnej zakładki do eksportu.
 - Użytkownik klika na **Export**.
 - Aplikacja wyświetla informację o braku danych do eksportu i wraca do okna dialogowego (w celu umożliwienia zaznaczenia przynajmniej jednej zakładki).
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Pliki z danymi z wybranych zakładek we wskazanym folderze.

Przypadek użycia „Edytuj numery portów (aplikacje)”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno dialogowe z listą portów i odpowiadających im aplikacjami z możliwością ich edycji.
 - Użytkownik wybiera odpowiednie pole i edytuje je.
 - Użytkownik klika na **OK**.
 - Aplikacja zapisuje zmiany i zamyka okno.
3. Alternatywny ciąg zdarzeń
 - Użytkownik rezygnuje z modyfikacji klikając na **Cancel**.
 - Aplikacja zamyka okno dialogowe.
 - Użytkownik klika na **Add new** w celu dodania nowej pozycji.
 - Aplikacja dodaje nową pozycję.
 - Użytkownik klika na **Ok**.
 - Aplikacja zapisuje zmiany i zamyka okno.
 - Użytkownik wybiera pozycję z listy i klika na **Delete**.
 - Aplikacja usuwa wybraną pozycję.
 - Użytkownik klika na **Ok**.
 - Aplikacja zapisuje zmiany i zamyka okno.
 - Użytkownik w trakcie przechwytywania wprowadza zmiany na liście.
 - Użytkownik klika na **Ok**.
 - Aplikacja zapisuje zmiany.
 - Aplikacja wyświetla okno dialogowe z informacją, że zmiany zostaną zastosowane dopiero po ponownym uruchomieniu przechwytywania.
 - Aplikacja zamyka okno dialogowe.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Zmodyfikowana lista portów (aplikacji).

Przypadek użycia „Zmień domyślne ustawienia aplikacji”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno dialogowe z ustawieniami aplikacji.
 - Użytkownik wybiera kategorię.
 - Użytkownik zmienia domyślne ustawienia.
 - Użytkownik klika na **Ok**.
 - Aplikacja zapisuje zmiany i zamyka okno.
3. Alternatywny ciąg zdarzeń
 - Użytkownik rezygnuje ze zmian klikając na **Cancel**.
 - Aplikacja zamyka okno dialogowe.
 - Użytkownik wybiera nieprawidłowy folder.
 - Użytkownik klika na **Ok**.
 - Aplikacja wyświetla okno dialogowe z informacją o nieprawidłowym folderze i wraca do kategorii, w której wystąpił błąd.
 - Użytkownik wprowadza błędną wartość.
 - Użytkownik klika na **Ok**.
 - Aplikacja wyświetla okno dialogowe z informacją o błędnej wartości i wraca do kategorii, w której znajduje się pole z błędną wartością.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Zmiany domyślnych ustawień aplikacji.

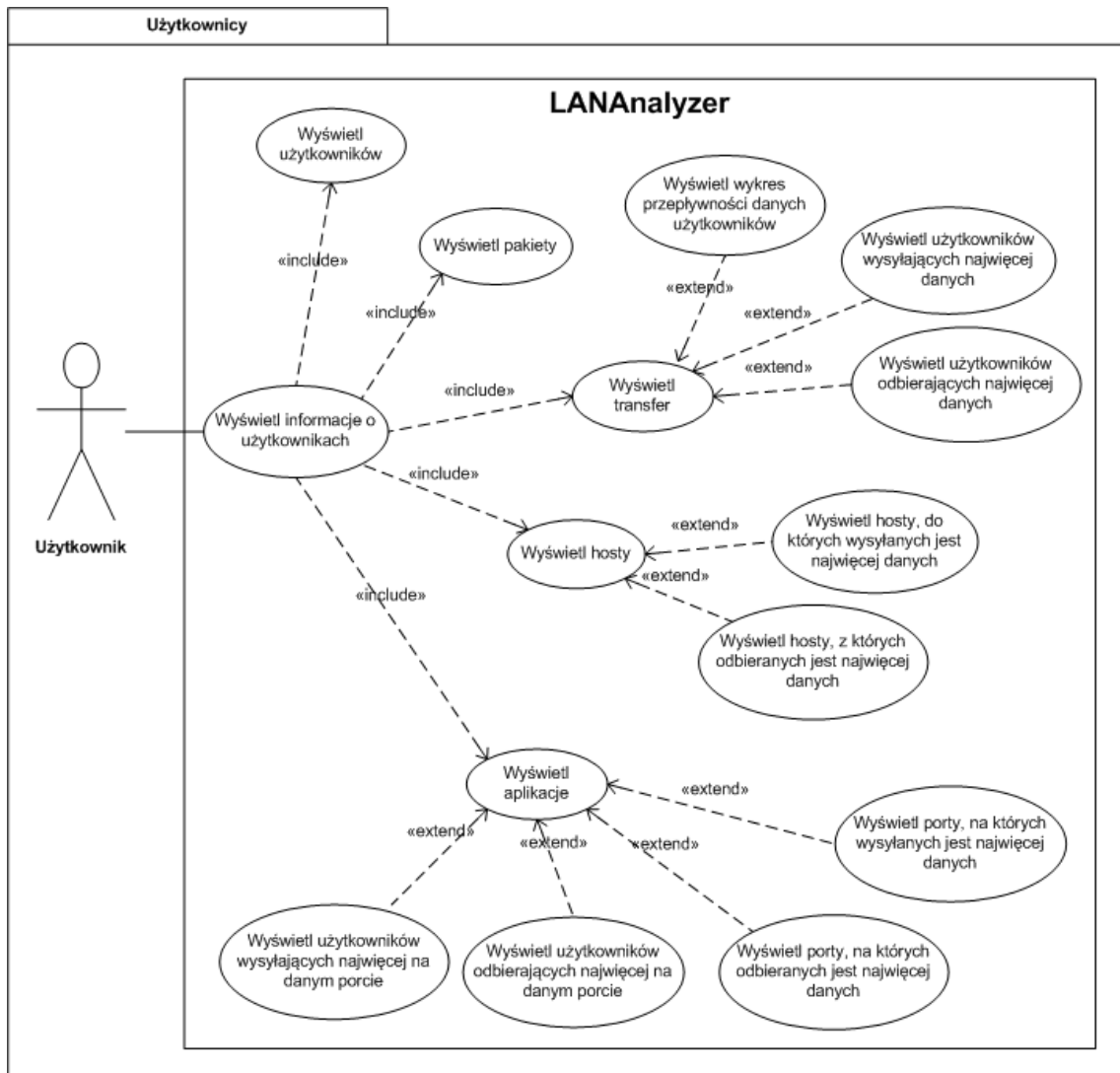
Przypadek użycia „Wyświetl logi”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla nowe okno z listą zdarzeń, które wystąpiły w trakcie działania aplikacji.

3. Alternatywny ciąg zdarzeń
 - Brak
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl pomoc”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja otwiera plik pomocy w systemie.
3. Alternatywny ciąg zdarzeń
 - Brak
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak



Rys. 3.2. Diagram przypadków użycia z pakietu *Użytkownicy*

Przypadek użycia „Wyświetl użytkowników”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią zakładkę.
 - Użytkownik przegląda tabelę z danymi o użytkownikach.
3. Alternatywny ciąg zdarzeń
 - Brak
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl pakiety”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią zakładkę.
 - Użytkownik przegląda tabelę z danymi o użytkownikach i ilości i rodzaju przesłanych przez nich pakietów.
3. Alternatywny ciąg zdarzeń
 - Użytkownik klika na **In**.
 - Aplikacja wyświetla jedynie pakiety przychodzące.
 - Użytkownik klika na **Out**.
 - Aplikacja wyświetla jedynie pakiety wychodzące.
 - Użytkownik klika na **Total**.
 - Aplikacja wyświetla sumę pakietów wychodzących i przychodzących.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl transfer”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią zakładkę.
 - Użytkownik przegląda tabelę z danymi o użytkownikach i ilości przesłanych oraz aktualnej prędkości przesyłanych przez nich danych.
3. Alternatywny ciąg zdarzeń
 - Brak
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl wykres przepływności danych użytkowników”:

1. Uczestniczący aktorzy
 - Użytkownik

2. Podstawowy ciąg zdarzeń

- Użytkownik klika na **Upload/Download speed graph**.
- Aplikacja wyświetla okno z wykresem prędkości przesyłanych danych przez użytkowników.

3. Alternatywny ciąg zdarzeń

- Użytkownik wybiera nowego użytkownika z listy.
 - Aplikacja prezentuje wykres prędkości dla nowego użytkownika.
- Użytkownik zmienia ramy czasowe wykresu.
 - Aplikacja prezentuje wykres dla nowych ram czasowych.
- Użytkownik zmienia opcje wyświetlania wykresu.
 - Aplikacja wprowadza zmiany w wyglądzie wykresu.
- Użytkownik klika na **Save image**.
 - Aplikacja wyświetla okno z prośbą o podanie nazwy, typu oraz lokalizacji pliku graficznego z obrazem wykresu.
 - Użytkownik klika na **Save**.
 - Aplikacja zapisuje plik.

4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia

- Brak

Przypadek użycia „Wyświetl użytkowników wysyłających najwięcej danych”:

1. Uczestniczący aktorzy

- Użytkownik

2. Podstawowy ciąg zdarzeń

- Użytkownik klika na **Top active users [uploaded]**.
- Aplikacja wyświetla okno dialogowe z prośbą o podanie liczby użytkowników do wyświetlenia.
- Użytkownik podaje liczbę użytkowników i klika na **Ok**.
- Aplikacja zamyka okno i wyświetla nowe okno dialogowe z listą osób wysyłających najwięcej danych.
- Użytkownik klika na **Ok**.
- Aplikacja zamyka okno.

3. Alternatywny ciąg zdarzeń

- Użytkownik klika na **Cancel** w oknie z prośbą o podanie liczby użytkowników.
 - Aplikacja zamyka okno dialogowe.
- 4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl użytkowników odbierających najwięcej danych”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik klika na **Top active users [downloaded]**.
 - Aplikacja wyświetla okno dialogowe z prośbą o podanie liczby użytkowników do wyświetlenia.
 - Użytkownik podaje liczbę użytkowników i klika na **Ok**.
 - Aplikacja zamyka okno i wyświetla nowe okno dialogowe z listą osób odbierających najwięcej danych.
 - Użytkownik klika na **Ok**.
 - Aplikacja zamyka okno.
3. Alternatywny ciąg zdarzeń
 - Użytkownik klika na **Cancel** w oknie z prośbą o podanie liczby użytkowników.
 - Aplikacja zamyka okno dialogowe.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl hosty”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią zakładkę.
 - Użytkownik przegląda tabelę z danymi o hostach, z jakimi się kontaktował użytkownik (czas połączenia, rodzaj aplikacji, transfery itp.).
3. Alternatywny ciąg zdarzeń

- Brak
- 4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl hosty, do których wysyłanych jest najwięcej danych”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik klika na **Top active hosts [uploaded]**.
 - Aplikacja wyświetla okno dialogowe z prośbą o podanie liczby hostów do wyświetlenia.
 - Użytkownik podaje liczbę hostów i klika na **Ok**.
 - Aplikacja zamyka okno i wyświetla nowe okno dialogowe z listą hostów, do których użytkownik wysłał najwięcej danych.
 - Użytkownik klika na **Ok**.
 - Aplikacja zamyka okno.
3. Alternatywny ciąg zdarzeń
 - Użytkownik klika na **Cancel** w oknie z prośbą o podanie liczby hostów.
 - Aplikacja zamyka okno dialogowe.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl hosty, z których odbieranych jest najwięcej danych”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik klika na **Top active hosts [downloaded]**.
 - Aplikacja wyświetla okno dialogowe z prośbą o podanie liczby hostów do wyświetlenia.
 - Użytkownik podaje liczbę hostów i klika na **Ok**.
 - Aplikacja zamyka okno i wyświetla nowe okno dialogowe z listą hostów, od których użytkownik odebrał najwięcej danych.
 - Użytkownik klika na **Ok**.

- Aplikacja zamyka okno.
- 3. Alternatywny ciąg zdarzeń
 - Użytkownik klika na **Cancel** w oknie z prośbą o podanie liczby hostów.
 - Aplikacja zamyka okno dialogowe.
- 4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl aplikacje”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią zakładkę.
 - Użytkownik przegląda tabelę z danymi o aplikacjach, z jakich korzysta użytkownik.
3. Alternatywny ciąg zdarzeń
 - Brak
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl porty, na których wysyłanych jest najwięcej danych”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik klika na **Top active ports [uploaded]**.
 - Aplikacja wyświetla okno dialogowe z prośbą o podanie liczby aplikacji do wyświetlenia.
 - Użytkownik podaje liczbę aplikacji i klika na **Ok**.
 - Aplikacja zamyka okno i wyświetla nowe okno dialogowe z listą aplikacji użytkownika, które wysłały najwięcej danych.
 - Użytkownik klika na **Ok**.
 - Aplikacja zamyka okno.
3. Alternatywny ciąg zdarzeń

- Użytkownik klika na **Cancel** w oknie z prośbą o podanie liczby aplikacji.
 - Aplikacja zamyka okno dialogowe.
- 4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl porty, na których odbieranych jest najwięcej danych”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik klika na **Top active ports [downloaded]**.
 - Aplikacja wyświetla okno dialogowe z prośbą o podanie liczby aplikacji do wyświetlenia.
 - Użytkownik podaje liczbę aplikacji i klika na **Ok**.
 - Aplikacja zamyka okno i wyświetla nowe okno dialogowe z listą aplikacji użytkownika, które odebrały najwięcej danych.
 - Użytkownik klika na **Ok**.
 - Aplikacja zamyka okno.
3. Alternatywny ciąg zdarzeń
 - Użytkownik klika na **Cancel** w oknie z prośbą o podanie liczby aplikacji.
 - Aplikacja zamyka okno dialogowe.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

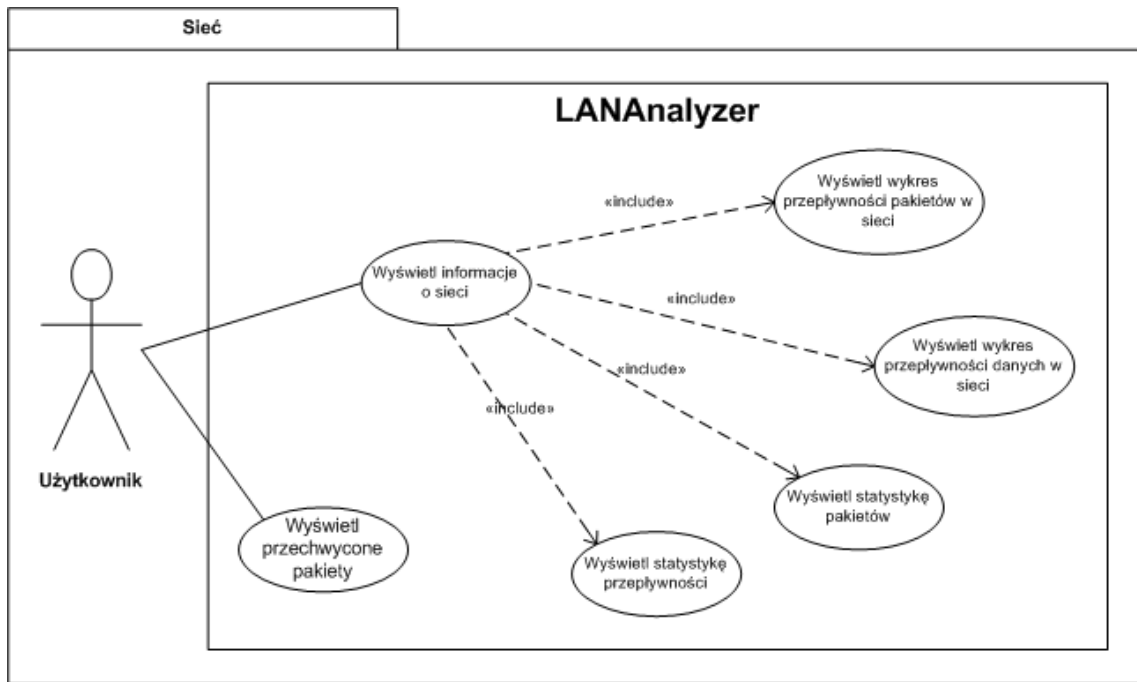
Przypadek użycia „Wyświetl użytkowników wysyłających najwięcej na danym porcie”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik klika na **Top active users on selected port [uploaded]**.
 - Aplikacja wyświetla okno dialogowe z prośbą o podanie liczby użytkowników do wyświetlenia.
 - Użytkownik podaje liczbę użytkowników i klika na **Ok**.

- Aplikacja zamyka okno i wyświetla nowe okno dialogowe z listą użytkowników wysyłających najwięcej danych przez aktualnie wybraną aplikację.
 - Użytkownik klika na **Ok**.
 - Aplikacja zamyka okno.
3. Alternatywny ciąg zdarzeń
- Użytkownik klika na **Cancel** w oknie z prośbą o podanie liczby użytkowników.
 - Aplikacja zamyka okno dialogowe.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
- Brak

Przypadek użycia „Wyświetl użytkowników odbierających najwięcej na danym porcie”:

1. Uczestniczący aktorzy
- Użytkownik
2. Podstawowy ciąg zdarzeń
- Użytkownik klika na **Top active users on selected port [downloaded]**.
 - Aplikacja wyświetla okno dialogowe z prośbą o podanie liczby użytkowników do wyświetlenia.
 - Użytkownik podaje liczbę użytkowników i klika na **Ok**.
 - Aplikacja zamyka okno i wyświetla nowe okno dialogowe z listą użytkowników odbierających najwięcej danych przez aktualnie wybraną aplikację.
 - Użytkownik klika na **Ok**.
 - Aplikacja zamyka okno.
3. Alternatywny ciąg zdarzeń
- Użytkownik klika na **Cancel** w oknie z prośbą o podanie liczby użytkowników.
 - Aplikacja zamyka okno dialogowe.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
- Brak



Rys. 3.3. Diagram przypadków użycia z pakietu *Sieć*

Przypadek użycia „Wyświetl wykres przepływności pakietów w sieci”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno z wykresem prędkości wszystkich przesyłanych pakietów w sieci.
3. Alternatywny ciąg zdarzeń
 - Użytkownik zmienia ramy czasowe wykresu.
 - Aplikacja prezentuje wykres dla nowych ram czasowych.
 - Użytkownik zmienia opcje wyświetlania wykresu.
 - Aplikacja wprowadza zmiany w wyglądzie wykresu.
 - Użytkownik klika na **Save image**.
 - Aplikacja wyświetla okno z prośbą o podanie nazwy, typu oraz lokalizacji pliku graficznego z obrazem wykresu.
 - Użytkownik klika na **Save**.
 - Aplikacja zapisuje plik.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia

- Brak

Przypadek użycia „Wyświetl wykres przepływności danych w sieci”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno z wykresem prędkości wysyłanych i odbieranych danych w sieci.
3. Alternatywny ciąg zdarzeń
 - Użytkownik zmienia ramy czasowe wykresu.
 - Aplikacja prezentuje wykres dla nowych ram czasowych.
 - Użytkownik zmienia opcje wyświetlania wykresu.
 - Aplikacja wprowadza zmiany w wyglądzie wykresu.
 - Użytkownik klika na **Save image**.
 - Aplikacja wyświetla okno z prośbą o podanie nazwy, typu oraz lokalizacji pliku graficznego z obrazem wykresu.
 - Użytkownik klika na **Save**.
 - Aplikacja zapisuje plik.
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl statystykę pakietów”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno z informacją o typie i liczbie przesyłanych pakietów w sieci.
3. Alternatywny ciąg zdarzeń
 - Brak

4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl statystykę przepływności”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla okno z liczbą wysłanych oraz odebranych danych oraz aktualną prędkością ich przesyłania w całej sieci.
3. Alternatywny ciąg zdarzeń
 - Brak
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

Przypadek użycia „Wyświetl przechwycone pakiety”:

1. Uczestniczący aktorzy
 - Użytkownik
2. Podstawowy ciąg zdarzeń
 - Użytkownik wybiera odpowiednią opcję z menu lub na pasku narzędziowym.
 - Aplikacja wyświetla nowe okno z listą przechwytywanych pakietów wraz z informacjami o nich.
3. Alternatywny ciąg zdarzeń
 - Brak
4. Wartości uzyskiwane przez aktorów po zakończeniu przypadku użycia
 - Brak

3.2. Diagram klas

Na rys. 3.4. przedstawiono uproszczony diagram klas. Pominięto na nim klasy niektórych mniej istotnych funkcji przedstawionych na diagramach przypadków użycia. Pominięto również klasę (i jej powiązania) odpowiedzialną za ustawienia aplikacji (ich przechowywanie, zapis, odczyt itd.). Przedstawione klasy zawierają jedynie najważniejsze, pod względem zrozumienia działania aplikacji, atrybuty oraz metody.

MainWindow

Główna klasa aplikacji. Wyświetla główne okno programu. Tworzy większość obiektów, jak również dodatkowe wątki: do odbierania danych (CaptureThread) oraz drugi do analizowania odebranych danych (ReceiverCore zostaje przeniesiony do wątku ReceiverThread).

CaptureThread

Jedna z najważniejszych klas aplikacji. Uruchamiana w osobnym wątku. Odbiera dane przechwytywane przez WinPcapa oraz po wstępnej analizie przekazuje je klasie ReceiverCore.

ReceiverCore

Klasa ta jest uruchamiana w osobnym wątku o nazwie ReceiverThread. Odpowiada za interpretację danych otrzymywanych od CaptureThread. Sprawdza kierunek przepływu pakietów, ich pochodzenie, aktualizuje odpowiednie zmienne oraz wysyła co 1 sekundę dane do powiązanych z nią klas.

TopActiveDialog

Klasa wyświetlająca okno z listą najbardziej aktywnych użytkowników, hostów, aplikacji oraz najbardziej aktywnych użytkowników na wybranej aplikacji. Oblicza oraz wyświetla procentowy udział „jeden” w „całości”.

PortNumbersDialog

Klasa odpowiedzialna za wyświetlanie, edycję oraz wyszukiwanie aplikacji wraz z powiązanymi z nimi portami.

FiltersDialog

Wyświetla listę domyślnych filtrów dla przechwytywanych pakietów oraz umożliwia ich edycję.

EditorDialog

Prosty edytor tekstowy dla długich i skomplikowanych nazw oraz kodów filtrów.

DevicesDialog

Wyświetla okno dialogowe z listą dostępnych urządzeń sieciowych w danym systemie, które mogą zostać użyte do przechwytywania danych.

ExportDataDialog

Klasa odpowiedzialna jest za eksport danych. Wyświetla okno dialogowe z opcjami eksportu.

SummaryDialog

Proste okno dialogowe z informacją o wyniku eksportu danych. Zawiera pole, którego odznaczenie spowoduje, że więcej nie będzie domyślnie wyświetlane.

StartCaptureDialog

Klasa wyświetlająca okno dialogowe z opcjami uruchomienia przechwytywania (moment startu, czas działania oraz zaawansowane opcje WinPcapa).

DoubleGraphWidget

Klasa jest bezpośrednio odpowiedzialna za rysowanie podwójnego wykresu, danych wysyłanych oraz odbieranych.

UserTransfersGraphDialog

Okno dialogowe wyświetlające wykres przepływności danych użytkowników w sieci. Posiada opcje zmiany wyglądu wykresu oraz wyboru użytkownika i typu prezentowanych danych (wysyłane, odbierane czy oba).

NetTransferGraphDialog

Okno dialogowe wyświetlające wykres przepływności danych w sieci (do i z Internetu). Posiada opcje zmiany wyglądu wykresu.

NetTransferDialog

Okno dialogowe wyświetlające całkowity transfer w sieci, aktualną prędkość wysyłania i odbierania danych oraz maksymalną wartość, jaka była od momentu uruchomienia przechwytywania.

NetPacketsDialog

Okno dialogowe prezentujące typy oraz liczbę przechwytywanych pakietów w sieci.

GraphWidget

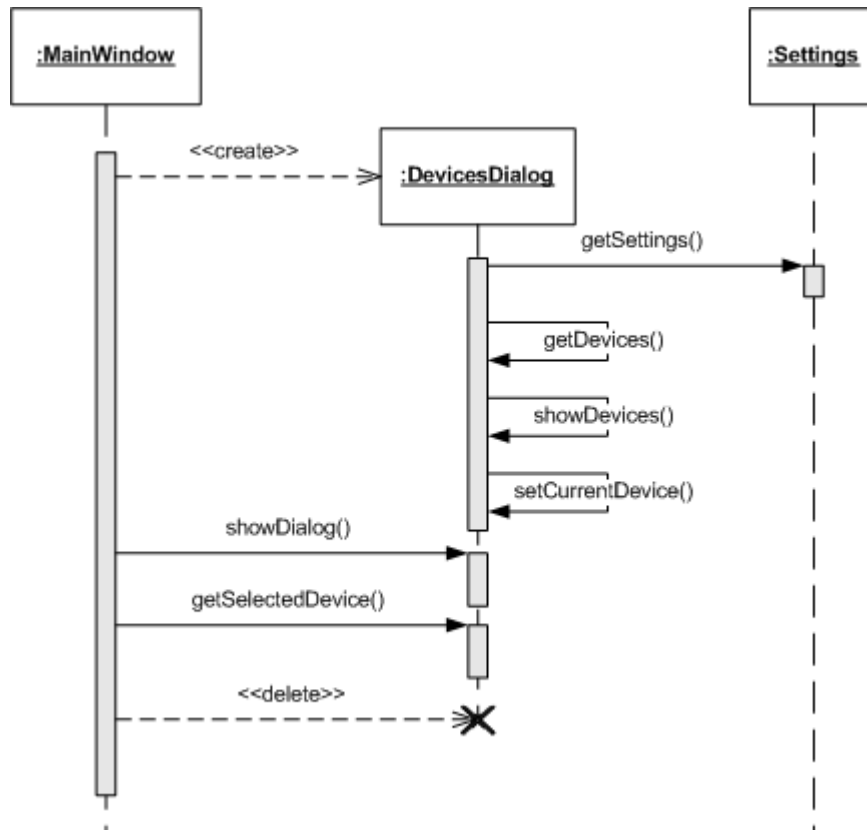
Klasa jest bezpośrednio odpowiedzialna za rysowanie pojedynczego wykresu.

NetPacketsGraphDialog

Okno dialogowe wyświetlające wykres przepływności pakietów w sieci. Posiada opcje zmiany wyglądu wykresu.

3.3. Diagram przebiegu (sekwencji)

W tym rozdziale przedstawione zostały diagramy sekwencji najważniejsze z punktu widzenia przeznaczenia i zrozumienia działania aplikacji. Pod każdym z diagramów przedstawiono obszerny opis, aby ułatwić ich zrozumienie oraz rozwiązać mogące się pojawić wątpliwości.

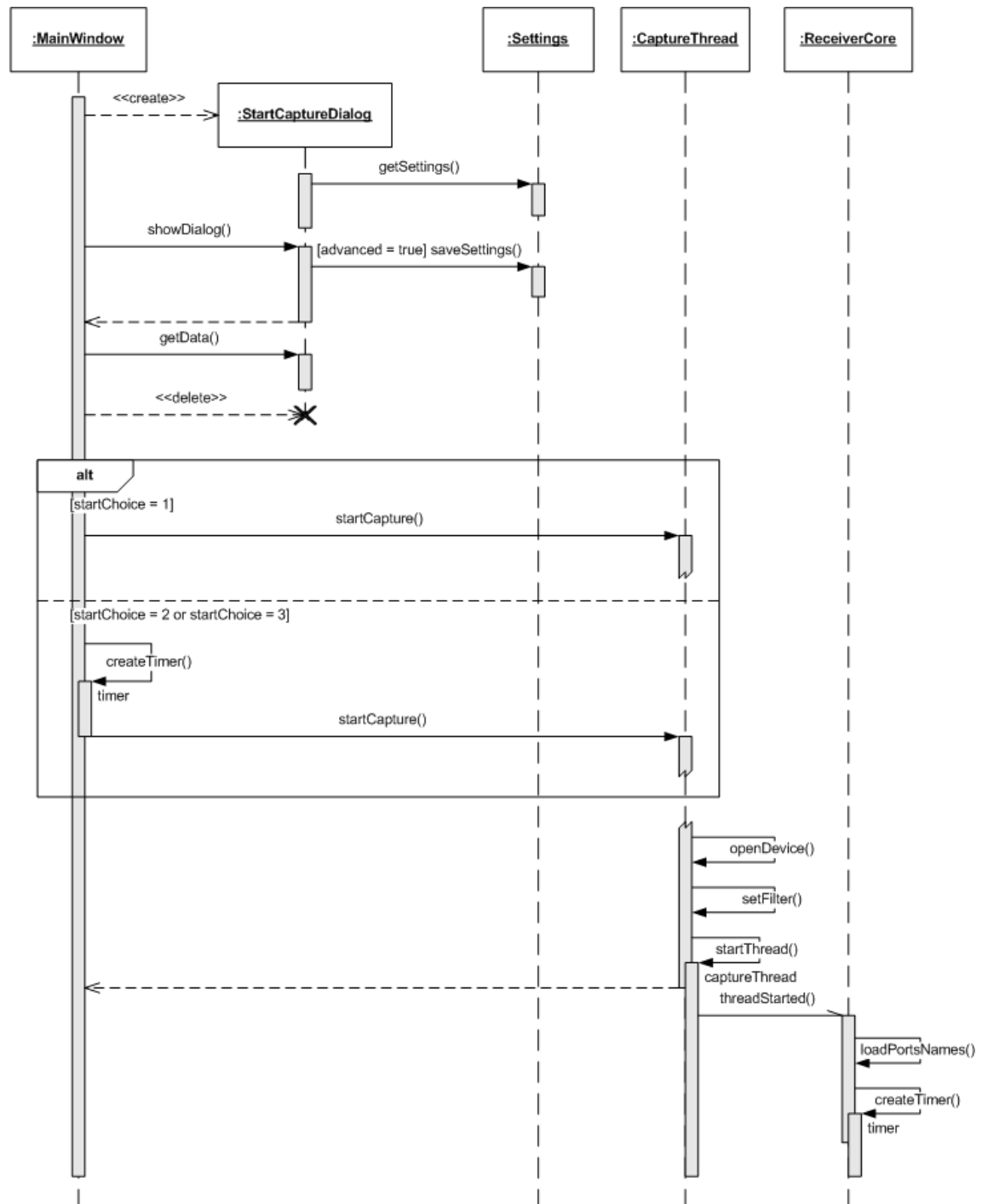


Rys. 3.5. Diagram przebiegu przypadku użycia „Wybierz urządzenie”

Przebieg wyboru urządzenia sieciowego służącego do przechwytywania pakietów przedstawiony jest na rys. 3.5.

Główne okno aplikacji **MainWindow** po wybraniu przez użytkownika odpowiedniej opcji w menu lub na pasku narzędziowym tworzy nowe okno dialogowe dające możliwość wyboru urządzenia służącego do przechwytywania pakietów w sieci. W konstruktorze nowo tworzonego okna w pierwszej kolejności następuje wczytanie ustawień (czy mają być pokazywane adresy IPv4, IPv6 oraz dodatkowe informacje o urządzeniu). Następnie tworzona jest lista dostępnych w systemie urządzeń oraz przygotowywana jest ona do pokazania również dodatkowych informacji. Jeśli istnieją to

ustawiane jest także aktualnie wybrane urządzenie. Zostaje wyświetlane okno. Po jego zamknięciu przez użytkownika pobierane jest, z jeszcze istniejącego obiektu, urządzenie wybrane przez użytkownika, a następnie obiekt jest usuwany.



Rys. 3.6. Diagram przebiegu przypadku użycia „Uruchom przechwytywanie z opcjami”

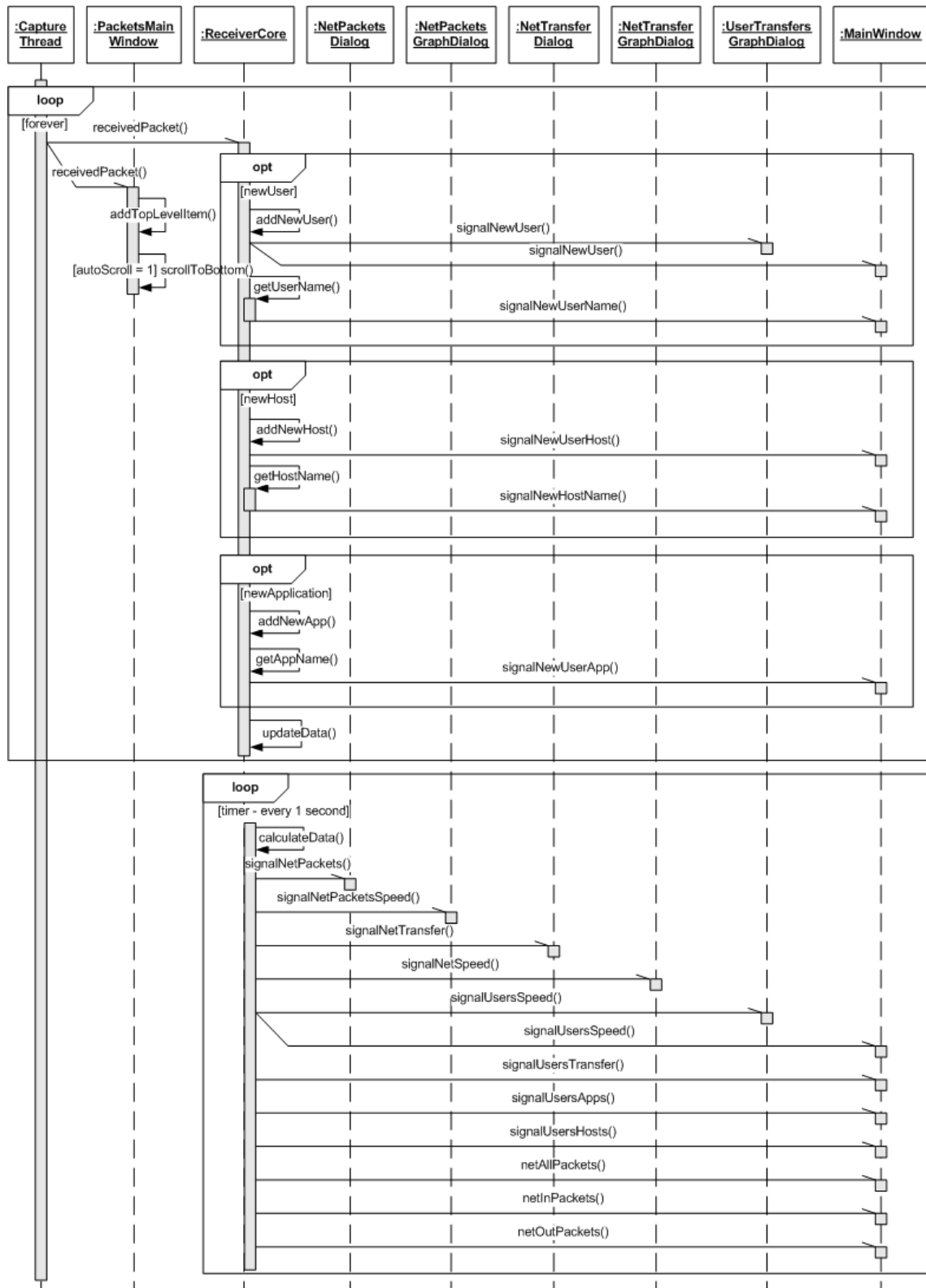
Uruchomienie przechwytywania pakietów może nastąpić na dwa sposoby – natychmiast lub poprzez okno dialogowe dające możliwość ustawienia momentu

uruchomienia, czasu trwania przechwytywania oraz dodatkowych opcji sterownika przechwytyującego. Diagram przebiegu przedstawiony na rys. 3.6. przedstawia drugi z tych przypadków, gdyż pierwszy jest końcowym jego etapem, jedynie uruchamianym poprzez kliknięcie na odpowiednią pozycję w menu lub na pasku narzędziowym.

Główne okno aplikacji **MainWindow** po wybraniu przez użytkownika odpowiedniej opcji w menu lub na pasku narzędziowym tworzy nowe okno dialogowe. W konstruktorze wczytuje aktualne ustawienia sterownika przechwytywania. Zostaje wyświetlone okno. Po wybraniu uruchomienia przechwytywania sprawdzane jest czy zaznaczona jest grupa opcji dotyczących działania sterownika przechwytywania. Jeśli tak to wprowadzone zmiany są zapisywane w ustawieniach aplikacji. Następnie wczytywane są opcje uruchomienia, czasu działania przechwytywania oraz usuwany jest obiekt okna dialogowego **StartCaptureDialog**.

Widoczna ramka **alt** to alternatywa. Wykonywana jest ta część, dla której zostaje spełniony warunek. Jeśli wybrano natychmiastowe uruchomienie przechwytywania (**startChoice = 1**) to jest ono uruchamiane (jako parametr podawany jest czas działania). Jeśli wybrano uruchomienie po określonym czasie lub o konkretnej godzinie (**startChoice = 2** lub **startChoice = 3**) to tworzony jest **timer**, który w odpowiednim momencie uruchomi przechwytywanie.

Uruchomienie przechwytywania rozpoczyna się od otwarcia w odpowiednim trybie urządzenia sieciowego oraz ustawienia filtra pakietów (jeśli został jakiś ustawiony). Następnie tworzony jest nowy wątek, który będzie odpowiedzialny za przechwytywanie oraz wysyłany jest sygnał o jego uruchomieniu do innego wątku (obiekt klasy **ReceiverCore**), który to będzie odbierał oraz odpowiednio przetwarzał przechwytywane pakiety. Po otrzymaniu sygnału wczytuje on aktualną listę aplikacji oraz uruchamia swój własny **timer**, który co jedną sekundę będzie obliczał niektóre wartości i wyniki przesyłał dalej do innych obiektów aplikacji.



Rys. 3.7. Diagram przebiegu przypadku użycia „Przechwytyj”

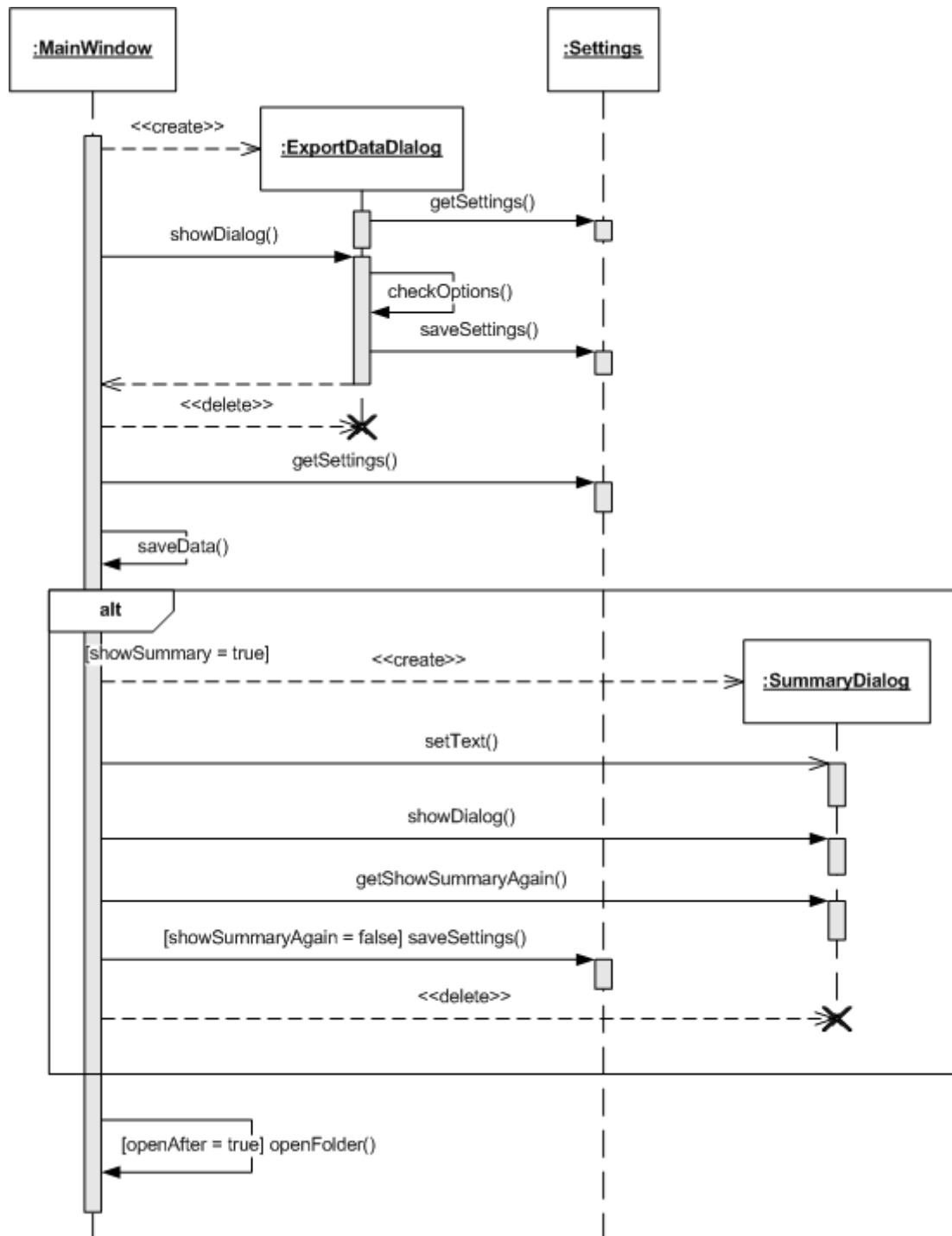
Komunikacja pomiędzy obiektami podczas przechwytywania oparta jest całkowicie na mechanizmie sygnałów i slotów z biblioteki Qt. Sygnały na diagramie przebiegu przechwytywania pakietów (rys. 3.7.) zostały przedstawione za pomocą

ciągłych linii z połową pustej i otwartej strzałki. Zgodnie z notacją stosowaną w UML są to komunikaty asynchroniczne, po wysłaniu których obiekt nie oczekuje na odpowiedź.

Użytkownicy oraz hosty to zawarte w pakiecie adresy IPv4. W oddzielnych wątkach otrzymujemy ich nazwy (nazwę komputera użytkownika oraz nazwę domeny przypisaną danemu adresowi). Aplikacje skojarzone są z numerami portów, na których występuje komunikacja. Lista jest zapisana w pliku dostarczonym razem z aplikacją (można ją edytować wybierając odpowiednią opcję w aplikacji).

Przechwytywanie pakietów, a tym samym przesyłanie ich za pomocą sygnału **receivedPacket()** trwa cały czas (w wątku – obiekcie klasy **CaptureThread**), stąd ramka **loop** definiująca pętlę typu **while (forever)**. Przerwanie następuje na żądanie użytkownika lub w wyniku błędu. Odebrane pakiety po wstępnej analizie przekazywane są za pomocą sygnału **receivedPacket()** do dwóch obiektów. Pierwszy z nich to obiekt klasy **PacketsMainWindow**, czyli okno pokazujące wszystkie przechwycone pakiety. Po odebraniu pakietu następuje dodanie go do listy. W przypadku wybrania przez użytkownika przewijania listy do aktualnego pakietu – lista zostaje przewinięta. Drugi z obiektów to obiekt klasy **ReceiverCore**, czyli innego wątku odpowiedzialnego za szczegółową analizę pakietów oraz dalsze przekazanie przetworzonych już danych do odpowiednich obiektów. W tym obiekcie w pierwszej kolejności następuje sprawdzenie czy pakiet pochodzi od lub jest przeznaczony dla nowego użytkownika. Jeśli użytkownika nie ma na liście to jest on dodawany oraz wysyłany jest odpowiedni sygnał do obiektów: wykresu przepływności danych użytkowników (**UserTransfersGraphDialog**) oraz głównego okna aplikacji (gdzie wyświetlana jest lista wszystkich użytkowników sieci). Następnie sprawdzana jest nazwa użytkownika (dokonywane jest to w osobnym wątku). Po jej otrzymaniu przekazywana jest do głównego okna aplikacji. Teraz sprawdzane jest czy pakiet pochodzi od lub jest przeznaczony dla znanego hosta. Jeśli nie to nowy host jest dodawany do listy oraz wysyłany jest odpowiedni sygnał do głównego okna. Następnie w oddzielnym wątku sprawdzana jest jego nazwa. Po jej otrzymaniu wysyłana jest do głównego okna aplikacji. Kolejny krok to sprawdzenie czy komunikacja odbywa się na znanym porcie, czyli przez znaną aplikację. Jeśli nie to dodawany jest nowy port oraz sprawdzana jest nazwa przypisanej jemu aplikacji. Informacja o nowej pozycji przesyłana jest do głównego okna. Teraz aktualizowane są odpowiednie zmienne (z listy użytkowników) na podstawie danych z pakietu.

W obiekcie klasy **ReceiverCore** istnieje od momentu uruchomienia przechwytywania **timer** (na diagramie umieszczony tuż obok linii życia). Jego zadaniem jest co 1 sekundę (stąd ramka **loop**) dokonywanie obliczeń takich jak np. przepływność danych (KB/s) czy liczba przesyłanych pakietów na sekundę oraz wysłanie ich do odpowiednich obiektów aplikacji.



Rys. 3.8. Diagram przebiegu przypadku użycia „Eksportuj dane”

Pokazany na rys. 3.8. diagram przebiegu eksportu danych przedstawia kroki wykonywane w celu zapisu danych z zakładki aplikacji w plikach.

Po wybraniu odpowiedniej opcji w menu lub na pasku narzędziowym główne okno **MainWindow** tworzy nowy obiekt – okno dialogowe dla eksportu danych (**ExportDataDialog**). W konstruktorze następuje wczytanie poprzednich opcji eksportu (folderu docelowego, format plików itp.). Wyświetlane jest okno. Przed zamknięciem okna, w przypadku, gdy użytkownik zdecydował się na eksport, sprawdzana jest poprawność ustawień (czy folder docelowy istnieje, czy została zaznaczona przynajmniej jedna zakładka do zapisu itp.) oraz zapisywane są w ustawieniach aplikacji wprowadzone zmiany. Obiekt zostaje usunięty. Przez główne okno, gdzie znajdują się zakładki, wczytywane są ustawienia eksportu oraz przeprowadzany jest zapis danych w plikach. Następnie, jeśli wybrano wyświetlenie okna podsumowującego, tworzony jest nowy obiekt – okno dialogowe, któremu przekazywany jest tekst do wyświetlenia. Okno podsumowania zostaje wyświetlone. Po zamknięciu sprawdzane jest czy została odznaczona opcja pokazywania podsumowania kolejnym razem. Jeśli tak to zmiana zostaje zapisana w ustawieniach. Ostatnim krokiem, jeśli użytkownik tego zażądał (**openAfter = true**), jest wyświetlenie w systemie folderu, gdzie eksportowane zostały pliki z danymi.

4. Implementacja aplikacji

Aplikacja powstała dla systemu Microsoft Windows XP, Vista oraz 7, zarówno w wersjach 32 jak i 64 bitowych. Program jest 32 bitowy. Dla tej rodziny systemów operacyjnych dostępne jest bardzo popularne, darmowe narzędzie umożliwiające łatwe przechwytywanie ruchu sieciowego – WinPcap. Ponieważ istnieje również jego odpowiednik dla systemów Linuks (LibPcap) w programie wykorzystano jedynie funkcje, które występują w obu wersjach. Ułatwić to ma w przyszłości przeniesienie aplikacji do systemów Linuks.

Oryginalny WinPcap oraz LibPcap napisane są dla języków C oraz C++ (istnieją również tzw. wrappery dla innych języków, ale ich stosowanie może wiązać się ze spadkiem wydajności programu). Z tego powodu aplikację utworzono wykorzystując wieloplatformową, napisaną całkowicie obiektowo w języku C++, bibliotekę Qt, która wraz z odpowiednimi narzędziami programistycznymi wchodzi w skład Qt SDK.

Kod programu kompilowany jest za pomocą darmowego MinGW (kompilator GCC), dedykowanego dla systemów Windows. Kompilator jest dostarczany oraz instalowany wraz z Qt SDK. [1, 3, 10, 11, 12, 13]

4.1. Środowisko

WinPcap (wersja 4.1.1)

WinPcap to narzędzie umożliwiające dostęp do sieci w warstwie łącza danych modelu ISO OSI RM (ang. *ISO Open System Interconnection Reference Model* – model odniesienia łączenia systemów otwartych), czyli w warstwie dostępu do sieci w modelu TCP/IP, w różnych systemach operacyjnych Windows. Pozwala na przechwytywanie (lokalnie, a także zdalnie) oraz wysyłanie pakietów, jak również na ich filtrowanie czy generowanie statystyk. WinPcap składa się z instalowanego w systemie sterownika, umożliwiającego aplikacjom wspomniany niskopoziomowy dostęp do sieci, oraz z API (ang. *Application Programming Interface* – interfejs programowania aplikacji) potrzebnego do napisania programów go wykorzystujących. WinPcap jest

wykorzystywany jako silnik filtrujący i przechwytyjący pakiety przez wiele aplikacji monitorujących ruch sieciowy, wykrywających próby włamań do sieci, sniffery, generatory ruchu w celach testowych itd. [13]

Qt SDK (wersja 4.5.3)

Qt SDK (ang. *Software Development Kit*) to zestaw przenośnych (wieloplatformowych) bibliotek i narzędzi programistycznych służących do tworzenia aplikacji, głównie opartych na graficznym interfejsie użytkownika. Biblioteki Qt charakteryzują się w pełni obiektową architekturą napisaną w języku C++. Wykorzystują one mechanizm sygnałów i slotów służący do rozsyłania zdarzeń z kontrolek, które w tym przypadku nazywane są widżetami. Z kolei widżety są podstawą w tworzeniu GUI (ang. *Graphical User Interface* – graficzny interfejs użytkownika). Oprócz obsługi interfejsu użytkownika biblioteki Qt zawierają także, niezależne od platformy systemowej, moduły do obsługi plików (tekstowych, graficznych), sieci, lokalizacji, wielowątkowości, zaawansowanej obsługi napisów. W skład Qt wchodzi również narzędzia programistyczne. Są to programy potrzebne do generowania pliku wynikowego (w tym kompilator MinGW), jak i dodatkowe aplikacje wspomagające tworzenie kodu, a są to np.: Qt Creator (zintegrowane środowisko programistyczne, które między innymi automatyzuje wykorzystanie wspomnianych programów do tworzenia plików wynikowych), Qt Designer (tworzenie graficznego interfejsu użytkownika), Qt Linguist (wspomaganie tłumaczenie programu na różne języki), Qt Assistant (rozbudowany system pomocy dla programistów). [11, 12]

4.2. Narzędzia

Narzędzia typowo programistyczne potrzebne do stworzenia kodu aplikacji, jej kompilacji oraz późniejszego przetłumaczenia pochodzą z Qt SDK.

Ponieważ aplikacja powinna być przyjazna dla użytkownika, konieczne jest również wykorzystanie między innymi ikon ułatwiających posługiwanie się nią. Ikony oraz zrzuty ekranów, wykorzystane w pliku pomocy aplikacji, utworzono za pomocą odpowiednich narzędzi graficznych.

Plik instalacyjny typu MSI, najpopularniejszy, oraz zalecany dla systemów Windows, został utworzony za pomocą pakietu Windows Installer XML (WiX).

Plik pomocy, dostępny poprzez menu w aplikacji, został napisany w Microsoft Office 2007, a konwersja, do wynikowego pliku PDF, została wykonana za pomocą OpenOffice.org 3.2.

4.2.1. Aplikacje z Qt SDK

Qt Creator

Kod aplikacji został napisany w wieloplatformowym zintegrowanym środowisku programistycznym (ang. *Integrated Development Environment*, IDE) jakim jest Qt Creator, dostarczany wraz z Qt SDK. Za pomocą tego narzędzia można stworzyć, modyfikować, testować oraz konserwować aplikacje. W skład Qt Creatora wchodzi: edytor kodu C++ (posiadający między innymi autouzupełnianie kodu, podświetlanie aktualnego bloku kodu oraz jego składni), zintegrowany edytor interfejsu użytkownika (ang. *Graphical User Interface*, GUI) – Qt Designer, narzędzie do zarządzania projektem oraz opcjami jego kompilowania, zintegrowany system pomocy (łącznie z pomocą kontekstową), graficzny debugger. [11, 12]

Qt Designer

Qt Designer to aplikacja, która umożliwia łatwe, graficzne tworzenie i edytowanie GUI tworzonej aplikacji. Pozwala, za pomocą techniki „przeciągnij i upuść”, na wygodne umieszczanie dużej ilości różnych gotowych widżetów (kontrolerek) na tworzonym formularzu, którym może być nowe okno dialogowe lub główne okno z menu i paskami narzędziowymi oraz stanu. Qt Designer jest oddzielną aplikacją, jednak została ona również zintegrowana z Qt Creatorem, dzięki czemu jest łatwiej i szybciej tworzyć i edytować okna tworzonej aplikacji. [11, 12]

Qt Assistant

Pod względem działania Qt Assistant jest aplikacją bardzo podobną do przeglądarek internetowych. Jej przeznaczeniem jest umożliwienie łatwego przeglądania (w tym również wyszukiwania konkretnych słów) dokumentacji technicznej dostępnej w Qt SDK. [11, 12]

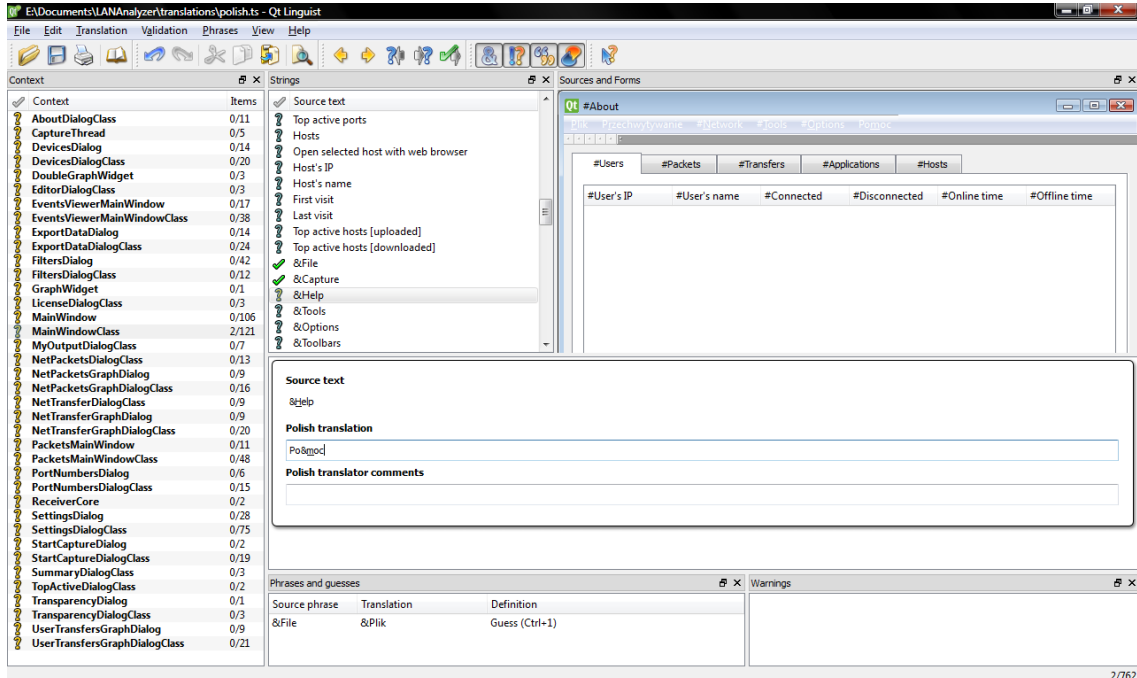
Qt Linguist

Domyślnym językiem interfejsu użytkownika aplikacji jest angielski. Jednak po uruchomieniu programu w jego ustawieniach można go zmienić. Tłumaczenie zrealizowano za pomocą odpowiednich funkcji oraz narzędzi dostarczonych wraz z Qt SDK według następujących kroków:

- Wszelkie teksty w kodzie aplikacji, które miały zostać przetłumaczone zostały zawarte w funkcji `QObject::tr()`. O możliwości tłumaczenia elementów GUI zdecydowano poprzez Qt Designera, który pozwala na zmianę wielu właściwości wykorzystywanych widżetów.
- Wykorzystano odpowiednie funkcje Qt do ładowania tłumaczeń z plików zewnętrznych (z rozszerzeniem `.qm`) i przeprowadzenia tłumaczenia aplikacji po jej uruchomieniu na język inny niż domyślny.
- Po odpowiednim zmodyfikowaniu głównego pliku projektu (plik `.pro`) za pomocą programu `lupdate`, który z plików źródłowych wyciąga oznaczony wcześniej tekst, utworzono pliki `.ts` (po jednym pliku dla każdego z wybranych języków).
- Za pomocą aplikacji Qt Linguist przeprowadzono tłumaczenie tekstów z plików `.ts`, a następnie eksportowano je do plików `.qm`, które są wykorzystywane przez aplikacje do zmiany jej języka.

Z powyższego opisu widać, że Qt Linguist jest wykorzystywany w jednym z etapów w trakcie realizacji tłumaczenia aplikacji. Program ten przyspiesza tłumaczenie tekstów. Na rys. 4.1. przedstawiono zrzut ekranu w trakcie tworzenia pliku z tłumaczeniem na język polski. Jak widać aplikacja składa się z kilku okienek. Wyświetla między innymi listę klas i okien, które wymagają tłumaczenia, konkretne

teksty znajdujące się w nich, podgląd fragmentu, z którego pochodzą, podpowiedzi tłumaczonych słów, ostrzeżenia o błędach oraz informacje o zatwierdzonych już tłumaczeniach. [11, 12]



Rys. 4.1. Zrzut ekranu aplikacji Qt Linguist w trakcie tłumaczenia

4.2.2. Narzędzia graficzne

Inkscape (wersja 0.46)

Inkscape jest wolnym programem wydany na licencji GNU GPL służącym do edycji grafiki wektorowej. Aplikacją posłużono się w celu utworzenia logo programu w jednym z popularnych formatów grafiki wektorowej – SVG (ang. *Scalable Vector Graphics*). Wybrano ten rodzaj grafiki, ponieważ w przeciwieństwie do grafiki rastrowej nie traci się na jakości obrazów podczas zmiany jego rozmiarów, a utworzone logo zostało wielokrotnie wykorzystane w różnych rozmiarach. Logo aplikacji zostało utworzone z połączenia oraz modyfikacji innych obrazów svg dostępnych w domenie publicznej.

IcoFX (wersja 1.6.4)

IcoFX jest aplikacją służącą do tworzenia oraz edycji ikon dla wielu systemów operacyjnych. Wydana jest na licencji Freeware. Program ten został wykorzystany do utworzenia specjalnej ikony aplikacji, tak aby można było ją prawidłowo wyświetlać również w systemie Windows Vista oraz nowszych, gdzie użytkownik może zmieniać rozmiar ikon. Plik ikony w formacie ICO zawiera aż 12 różnych obrazów (w rozdzielczościach 16x16, 32x32, 48x48, 256x256 pikseli, każdy w 4-, 8- oraz 32-bitach kolorów), jest zapisany w skompresowanym png z zachowaniem przeźroczystości.

Gimp (wersja 2.6.6)

Gimp jest jednym z najpopularniejszych programów do obróbki grafiki rastrowej, udostępnianym na licencji GNU GPL. Posłużył on do przygotowania rysunków wykorzystanych w pliku pomocy aplikacji LANAnalyzer oraz w rozdziale przedstawiającym interfejs użytkownika. Również w Gimpie utworzona została animowana ikonka programu, wyświetlana na żądanie użytkownika w zasobniku systemowym, informująca o włączonym przechwytywaniu pakietów. Ikonka jest w formacie gif w rozdzielczości 32x32 pikseli i 8-bitach kolorów.

4.2.3. Instalacja

Bardzo ważnym aspektem każdej tworzonej aplikacji, często zaniedbywanym przez programistów, jest jej instalacja. Proces instalacji ułatwia użytkownikowi wstępne zapoznanie się z aplikacją, np. jej licencją czy składnikami oraz wyręcza go w tworzeniu skrótów w menu start czy na pulpicie oraz kopiowaniu aplikacji itp.

Istnieje wiele narzędzi pomagających w tworzeniu instalatorów, zarówno płatnych jak i darmowych. Podzielić je można między innymi na podstawie typów tworzonych plików. Dla systemu Windows podstawowym typem są pliki .msi, które są pakietami instalacyjnymi obsługiwanymi przez Instalatora Windows (ang. *Windows Installer*). W tworzeniu tego typu instalatorów pomaga między innymi narzędzie wydane na licencji Common Public License przez Microsoft – **Windows Installer XML (WiX)**.

WiX pozwala na łatwe tworzenie wspomnianych pakietów instalacyjnych na podstawie pliku tekstowego o strukturze XML. Do prawidłowej pracy wymaga .NET Framework 2.0. Wersja użyta do stworzenia pakietu instalacyjnego aplikacji LANAnalyzer to 3.0.5419.

Posiadając wcześniej napisany plik tekstowy o strukturze XML o rozszerzeniu wxs, który jest głównym plikiem źródłowym, kompilujemy za pomocą programu *candle.exe*, a następnie wykorzystujemy linkera *light.exe* do połączenia, na podstawie powstałego pliku, plików aplikacji i utworzenia pliku końcowego. Wywołanie obu poleceń może wyglądać tak jak przedstawiono poniżej.

```
candle.exe -out lananalyzer.wixobj lananalyzer.wxs -dcodepage=1250

light -out setup_pl-pl.msi lananalyzer.wixobj -ext WixUIExtension
      -cultures:pl-pl -loc pl-pl.wxl

candle.exe -out lananalyzer.wixobj lananalyzer.wxs -dcodepage=1252

light -out setup_en-us.msi lananalyzer.wixobj -ext WixUIExtension
      -cultures:en-us -loc en-us.wxl

light -out setup_de-de.msi lananalyzer.wixobj -ext WixUIExtension
      -cultures:de-de -loc de-de.wxl
```

Polecenia te są wywoływane kilkakrotnie z różnymi parametrami, aby uzyskać trzy pliki .msi – pakiety instalacyjne w trzech językach: polskim, angielskim oraz niemieckim. W pliku źródłowym .wxs zastosowano odpowiednie zmienne:

- `"!(loc.nazwa)"` – do tłumaczeń komunikatów znajdujących się w osobnych plikach typu XML o rozszerzeniu .wxl. Są to tłumaczenia komunikatów wyświetlanych w czasie instalacji oraz nazwy skrótów utworzonych przez instalatora w wyniku instalacji.
- `"$(var.codepage)"` – do podania w czasie kompilacji strony kodowej dla danego języka.

Ponieważ przechowywanie oraz rozprowadzanie aplikacji wśród użytkowników w kilku plikach językowych (w tym przypadku wspomnianych trzech) jest niezbyt wygodne utworzono wielojęzyczny pakiet instalacyjny **LANAnalyzer.msi**. W zależności od ustawień regionalnych w systemie plik zostanie uruchomiony w jednym ze wspomnianych języków.

Aby utworzyć taki plik potrzebne są dodatkowe narzędzia: program *MsiTran.exe* oraz skrypty *WiSubStg.vbs* i *WiLangId.vbs*. Pliki te dostępne są w pakiecie *Windows Software Development Kit (SDK)*, który można pobrać ze strony Microsoftu. Poniżej przedstawiono wywołanie tych plików. Jako plik bazowy posłuży plik *.msi* w wersji angielskiej, gdyż ten język jest domyślnym w aplikacji.

```
MsiTran.exe -g setup_en-us.msi setup_de-de.msi de-de.mst
MsiTran.exe -g setup_en-us.msi setup_pl-pl.msi pl-pl.mst

WiSubStg.vbs setup_en-us.msi de-de.mst 1031
WiSubStg.vbs setup_en-us.msi pl-pl.mst 1045

WiLangId.vbs setup_en-us.msi Package 1033,1031,1045
```

Program *MsiTran.exe* daje nam w wyniku porównania dwóch plików *.msi* tak zwany plik transformacji *.mst*. Otrzymane pliki *.mst* za pomocą skryptu *WiSubStg.vbs* łączymy z oryginalnym plikiem *.msi*. Ostatnim etapem jest odpowiednie zmodyfikowanie już wielojęzycznego pliku wynikowego *.msi*, za pomocą skryptu *WiLangId.vbs*, tak aby instalator Windowsa wiedział, że ma on do czynienia z wielojęzycznym pakietem instalacyjnym (że odpowiednie pliki transformacji są zawarte w pliku *.msi*) i mógł wybrać odpowiedni język w zależności od ustawień w systemie użytkownika. [9, 14]

4.3. Kod źródłowy

Jak już wspomniano, w rozdziale o implementacji aplikacji, wykorzystanym językiem jest C++. W tym języku najważniejszą oraz jedyną wymaganą funkcją jest *main()*. To od niej rozpoczyna się wykonywanie programu. W projekcie znajduje się ona w pliku *main.cpp*. Funkcja ta zwraca wartość całkowitą typu *int*. Jej argumenty to liczba argumentów wywołania programu ($argc \geq 1$), oraz tablica wskaźników do tekstów argumentów (*argv*) wywołania.

```
#include <QtGui/QApplication>
#include "mainwindow.h"

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
```

```
Q_INIT_RESOURCE(images);

QApplication::setQuitOnLastWindowClosed(false);

QTextCodec::setCodecForTr(QTextCodec::codecForName("UTF-8"));
QTextCodec::setCodecForCStrings(QTextCodec::codecForName("UTF-8"));

QCoreApplication::setApplicationName("LANAnalyzer");
QCoreApplication::setOrganizationName("Helfajer");
QCoreApplication::setOrganizationDomain("helfajer.info");
QCoreApplication::setApplicationVersion("0.1.0");

MainWindow w;
w.show();

return a.exec();
}
```

W pierwszej kolejności tworzymy obiekt o nazwie *a* klasy *QApplication*, wykorzystując jeden z jej dostępnych konstruktorów, podając jako parametry argumenty wywołania funkcji *main()*. Klasa ta jest podstawą w tworzeniu programów z GUI z Qt. Następnie inicjalizujemy zasoby (w tej aplikacji są to wszystkie używane pliki graficzne), wyłączamy możliwość zakończenia aplikacji po zamknięciu jej ostatniego okna (jest to wykorzystywane podczas zamykania aplikacji w trybie minimalizowania do ikony w zasobniku systemowym), ustawiamy system kodowania dla używanych łańcuchów znaków (wyświetlanych tekstów) oraz ustawiamy podstawowe informacje o aplikacji (wykorzystywane przez niektóre funkcje). Kolejnym krokiem jest deklaracja obiektu naszej aplikacji i wywołanie funkcji ją wyświetlającej. W ten sposób wyświetlone zostaje główne okno programu LANAnalyzer. Po jego zamknięciu funkcja *main* zwraca wartość jaką otrzymała z funkcji *exec()*. [1, 3, 10]

Uruchomienie przechwytywania

W klasie *CaptureThread* znajdują się metody do uruchomienia, zatrzymania oraz działania przechwytywania pakietów.

Metoda *startCapture()* wywoływana jest z klasy *MainWindow* w momencie, gdy rozpoczynamy przechwytywanie pakietów. Metoda *pcap_open_live()* otwiera w WinPcapie wybrane urządzenie do przechwytywania z opcjami, jakie zostały podane przez użytkownika w ustawieniach aplikacji. Następnie odczytujemy maskę sieci, kompilujemy oraz ustawiamy filtr pakietów. Na końcu, aby nie blokować dostępu do aplikacji, uruchamiamy przechwytywanie w osobnym wątku.

```
bool CaptureThread::startCapture(pcap_if_t *d, quint8 mode, quint16 bytes,
                                quint16 timeout, const QString &filterCode,
                                qint32 packetsLimit)
{
    abort = false;
    packets = 0;
    this->packetsLimit = packetsLimit; // -1 if no limit

    // open the device
    char errbuf[PCAP_ERRBUF_SIZE];
    if ((adhandle = pcap_open_live(d->name, bytes, mode, timeout, errbuf)) == NULL)
    {
        // 3 - critical
        emit infoMessage(3, tr("Capture thread"), tr("Unable to open the network device.
                                                                    Probably the selected device is
                                                                    not supported by WinPcap."));

        // free the device list
        //pcap_freealldevs(alldevs);

        return false;
    }

    // retrieve the mask
    u_int netmask;

    if (d->addresses != NULL)
    {
        // retrieve the mask of the first address of the interface
        Netmask = ((struct sockaddr_in*) (d->addresses->netmask))->sin_addr.S_un.S_addr;
    }
    else
    {
        // if the interface is without addresses we suppose to be in a C class network
        netmask = 0xffffffff;
    }

    // compile the filter
    struct bpf_program fcode;

    if (pcap_compile(adhandle, &fcode, filterCode.toLocal8Bit().data(), 1, netmask) < 0)
    {
        // 3 - critical
        emit infoMessage(3, tr("Capture thread"), tr("Unable to compile the packet
                                                                    filter. Check the syntax."));

        // free the device list
        //pcap_freealldevs(alldevs);

        return false;
    }

    // set the filter
    if (pcap_setfilter(adhandle, &fcode) < 0)
    {
        // 3 - critical
        emit infoMessage(3, tr("Capture thread"), tr("Error setting the filter."));

        // free the device list
        //pcap_freealldevs(alldevs);
    }
}
```

```
        return false;
    }

    if (!isRunning())
        start(NormalPriority);

    if (isRunning())
    {
        emit threadStarted();

        return true;
    }
    else
    {
        return false;
    }
}
```

Zatrzymanie przechwytywania

Metoda *stopCapture()*, tak jak poprzednia, również jest wywoływana z klasy *MainWindow*. Zatrzymuje ona wątek przechwytywania oraz wysyła odpowiedni sygnał, gdy to nastąpiło.

```
bool CaptureThread::stopCapture()
{
    abort = true;
    wait();

    emit threadStopped();

    return true;
}
```

Przechwytywanie

Metoda *run()* to reimplementacja chronionej metody z Qt. Jest ona uruchamiana w oddzielnym wątku w momencie jego uruchomienia. Poniżej opisano jej fragmenty, gdyż pozostałe typy pakietów (protokoły) są obsługiwane w podobny sposób.

W pierwszej kolejności tworzymy wskaźniki do struktur opisujących wykorzystywane typy pakietów. Struktury te znajdują się w pliku *protocols.h*. Następnie tworzymy kolejne zmienne.

```
void CaptureThread::run()
{
    eth_header *ethHeader;
    arp_header *arpHeader;
```

```

rarp_header *rarpHeader;
ip_header *ipHeader;
tcp_header *tcpHeader;
udp_header *udpHeader;
icmp_header *icmpHeader;
igmp_header *igmpHeader;

char tcpFlag[8][5] = {"FIN ", "SYN ", "RST ", "PSH ", "ACK ", "URG ", "ECE ", "CWR
                    "};

int icmp_mesglen = 16;
icmp_mesg icmpMesg[] = { {0, "Echo Reply"},
                          {3, "Destination Unreachable"},
                          {4, "Source Quench"},
                          {5, "Redirect Message"},
                          {6, "Alternate Host Address"},
                          {8, "Echo Request"},
                          {9, "Router Advertisement"},
                          {10, "Router Selection"},
                          {11, "Time Exceeded"},
                          {12, "Parameter Problem"},
                          {13, "Timestamp Request"},
                          {14, "Timestamp Reply"},
                          {15, "Information Request"},
                          {16, "Information Reply"},
                          {17, "Address Mask Request"},
                          {18, "Address Mask Reply"}
                        };

int igmp_mesglen = 8;
igmp_mesg igmpMesg[] = { {0x11, "Membership Query"},
                          {0x12, "IGMPv1 Membership Report"},
                          {0x16, "IGMPv2 Membership Report"},
                          {0x17, "Leave Group"},
                          {0x22, "IGMPv3 Membership Report"},
                          {0x24, "Multicast Router Advertisement"},
                          {0x25, "Multicast Router Solicitation"},
                          {0x26, "Multicast Router Termination"}
                        };

struct pcap_pkthdr *header;
const u_char *pkt_data;
time_t local_tv_sec;
struct tm *ltime;
char timeStr[16];
QString info;
char sMac[18], dMac[18], source[20], dest[20];
u_int ip_hlen;
int i, res;

```

Przechwytywanie pakietów odbywa się w pętli *forever()*, która wykonuje się póki nie zażądamy jej przerwania lub wystąpi jakiś błąd w funkcji WinPcapa przechwytyjącej pakiety – *pcap_next_ex()*.

```

// retrieve the packets
forever
{

```

```
if (abort)
    return;

res = pcap_next_ex(adhandle, &header, &pkt_data);

// The return value can be:
// 1 if the packet has been read without problems
// 0 if the timeout set with pcap_open_live() has elapsed, in this case
//   pkt_header and pkt_data don't point to a valid packet
// -1 if an error occurred
// -2 if EOF was reached reading from an offline capture

if (res == -1)
{
    // 3 - critical
    emit infoMessage(3, tr("Capture thread"), QString(tr("Error while reading
        packet: \"%1\"")).arg(pcap_geterr(adhandle)));

    emit breakThread();
    return;
}

if (res == 0)
    continue;

if (packets == packetsLimit)
{
    emit breakThread();
    return;
}
++packets;
```

Z przechwyconego pakietu wyciągamy czas, a następnie, wiedząc, że odbieramy tylko pakiety Ethernetowi, odczytujemy adresy MAC źródła oraz celu.

```
// convert the timestamp to readable format
local_tv_sec = header->ts.tv_sec;
ltime = localtime(&local_tv_sec);
strftime(timeStr, sizeof timeStr, "%H:%M:%S", ltime);

// ETH
ethHeader = (eth_header*)pkt_data;

sprintf(sMac, "%.2X:%.2X:%.2X:%.2X:%.2X:%.2X", ethHeader->smac[0],
    ethHeader->smac[1], ethHeader->smac[2], ethHeader->smac[3],
    ethHeader->smac[4], ethHeader->smac[5]);
sprintf(dMac, "%.2X:%.2X:%.2X:%.2X:%.2X:%.2X", ethHeader->dmac[0],
    ethHeader->dmac[1], ethHeader->dmac[2], ethHeader->dmac[3],
    ethHeader->dmac[4], ethHeader->dmac[5]);
```

Następnie sprawdzamy typ pakietu, odczytujemy z niego odpowiednie dane i za pomocą sygnału *receivedPacket()* wysyłamy je do klasy *ReceiverCore*, która działając w osobnym wątku dokonuje ich dalszej analizy. W ten sam sposób postępujemy z RARP.


```
// ARP
// 0x0806 Address Resolution Protocol (ARP)
if (ntohs(ethHeader->type) == 0x0806)
{
    arpHeader = (arp_header*)(pkt_data + ETHERNET_LENGTH);

    sprintf(source, "%d.%d.%d.%d", arpHeader->spa.byte1, arpHeader->spa.byte2,
        arpHeader->spa.byte3, arpHeader->spa.byte4);
    sprintf(dest, "%d.%d.%d.%d", arpHeader->tpa.byte1, arpHeader->tpa.byte2,
        arpHeader->tpa.byte3, arpHeader->tpa.byte4);

    // 1 ARP request
    // 2 ARP response
    // 3 RARP request
    // 4 RARP response
    // 5 Dynamic RARP request
    // 6 Dynamic RARP reply
    // 7 Dynamic RARP error
    // 8 InARP request
    // 9 InARP reply

    if (ntohs(arpHeader->oper) == 0x0001)
        info = "ARP request";

    if (ntohs(arpHeader->oper) == 0x0002)
        info = "ARP response";

    emit receivedPacket(timeStr + QString("%.1").arg(header->ts.tv_usec),
        header->len,
        sMac,
        dMac,
        0x0806,
        inet_addr(source),
        inet_addr(dest),
        65536,
        65536,
        info);

    continue;
}
```

Jeśli odebraliśmy pakiet typu IP to sprawdzamy (w instrukcji *switch*) typ pakietu wyższego poziomu, z którego następnie odczytujemy potrzebne dane (np. typ komunikatu ICMP).

```
// IP // 0x0800 Internet Protocol, Version 4 (IPv4)
// 0x86DD Internet Protocol, Version 6 (IPv6)
if (ntohs(ethHeader->type) == 0x0800)
{
    ipHeader = (ip_header*)(pkt_data + ETHERNET_LENGTH);

    // Internet Header Length is the length of the internet header in 32
    // bit words, and thus points to the beginning of the data.
    // Note that the minimum value for a correct header is 5 (5×32 = 160 bits).
    // Being a 4-bit value, the maximum length is 15 words (15×32 bits) or 480
    // bits.

    ip_hlen = (ipHeader->ver_ihl & 0xf) << 2;
```

```
switch (ipHeader->proto)
{
// IP TCP
    case 6: tcpHeader = (tcp_header*)((u_char*)ipHeader + ip_hlen);

        for (i = 0, info = ""; i < 8; ++i)
        {
            if (tcpHeader->flag & 1<<i)
                info.append(tcpFlag[i]);
        }

        emit receivedPacket(timeStr
                            + QString("%.1").arg(header->ts.tv_usec),
                            header->len,
                            sMac,
                            dMac,
                            6,
                            ipHeader->saddr,
                            ipHeader->daddr,
                            ntohs(tcpHeader->sport),
                            ntohs(tcpHeader->dport),
                            info);

        break;
}
```

Poniżej wspomniany odczyt danych z komunikatu ICMP, w pętli *for* sprawdzany jest typ wiadomości.

```
// IP ICMP
    case 1: icmpHeader = (icmp_header*)((u_char*)ipHeader + ip_hlen);

        for (i = 0; i < icmp_mesglen; ++i)
        {
            if (icmpHeader->type == icmpMesg[i].type)
            {
                info = icmpMesg[i].mesg;
                break;
            }
        }

        if (i == icmp_mesglen)
            info = "unknown ICMP message type";

        emit receivedPacket(timeStr
                            + QString("%.1").arg(header->ts.tv_usec),
                            header->len,
                            sMac,
                            dMac,
                            1,
                            ipHeader->saddr,
                            ipHeader->daddr,
                            65536,
                            65536,
                            info);

        break;
}
```

W przypadku, gdy pojawił się nieobsługiwany protokół wysyłamy sygnał z odpowiednią informacją.

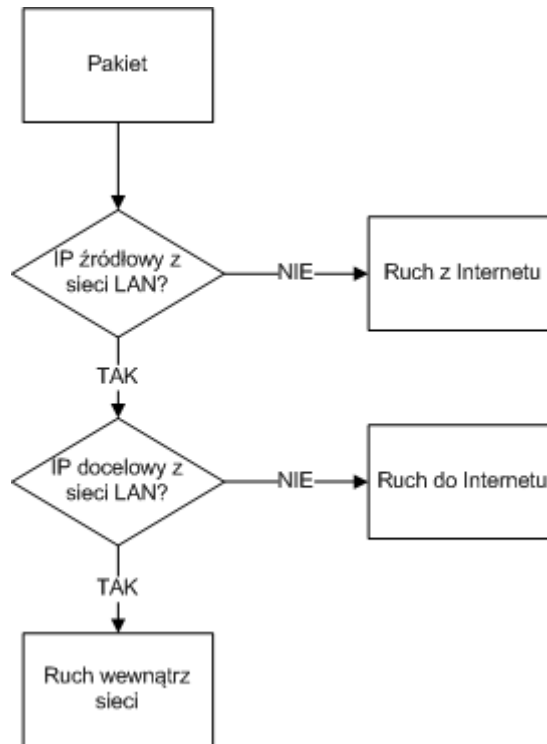
```
// other
    default: info = "protocol not supported";

        emit receivedPacket(timeStr
                            + QString("%.1").arg(header->ts.tv_usec),
                            header->len,
                            sMac,
                            dMac,
                            0,
                            ipHeader->saddr,
                            ipHeader->daddr,
                            65536,
                            65536,
                            info);

        break;
    }
    continue;
}
```

Analiza pakietów

Analiza pakietów dokonywana jest według bardzo prostego schematu pokazanego na rys. 4.2. Na jego podstawie widać, że istnieją trzy kierunki przesyłania pakietów. Ruch wewnątrz sieci nie jest brany pod uwagę. Aplikacja analizuje jedynie ruch użytkowników do i z Internetu.



Rys. 4.2. Schemat blokowy algorytmu analizy pakietów

Klasa *ReceiverCore*, jak już wspomniano wyżej, działa w osobnym wątku. Metoda *ReceivedPacket()* jest wywoływana dla każdego przechwyconego pakietu. W pierwszej kolejności wywoływana jest w niej metoda odpowiedzialna za zmianę liczników pakietów przesyłanych w sieci. Następnie sprawdzany jest kierunek przesyłanego pakietu. Funkcja *checkIP()* zwraca *true*, jeśli podany adres IP należy do sieci, w przeciwnym przypadku zwraca *false*. Ruch wewnątrz sieci jest odrzucany. Ruch do oraz z Internetu interpretowany jest tak samo, jedynie odpowiednio inne zmienne są aktualizowane, dlatego w dalszej części opisany zostanie przypadek ruchu do Internetu.

```

void ReceiverCore::receivedPacket(const QString &time, quint32 length,
                                   const QString &sMac, const QString &dMac,
                                   quint16 type, quint32 sIP, quint32 dIP, quint32 sPort,
                                   quint32 dPort, const QString &info)
{
    incrementNetCounters(type);

    // IP from our network?
    if (checkIP(sIP))
    {
        if (checkIP(dIP))
            // network traffic
            {
            }
        // to Internet
    }
}
    
```

```
else  
{
```

W pierwszej kolejności odrzucane są pakiety związane z *multicastem*. Następnie sprawdzane jest czy pakiet pochodzi od użytkownika już znanego. Jeśli nie to dodajemy nowe pozycje do odpowiednich list oraz wysyłamy sygnał z informacją o nowym użytkowniku oraz sprawdzamy jego nazwę wywołując *QHostInfo::lookupHost()* – metodę dostępną w Qt.

```
if (multicastIP(dIP))  
{  
    // 0 download  
    emit signalMulticast(dIP, sIP, length, 0);  
    return;  
}  
  
if (!usersList.contains(sIP))  
{  
    usersList.append(sIP);  
  
    Hosts host;  
    usersHosts.append(host);  
  
    Apps app;  
    usersApps.append(app);  
  
    usersUpSpeed.append(0.0);  
    usersDownSpeed.append(0.0);  
  
    usersUp.append(0);  
    usersUpPrev.append(0);  
    usersDown.append(0);  
    usersDownPrev.append(0);  
  
    listsAppend();  
  
    addr.S_un.S_addr = sIP;  
    QString user = inet_ntoa(addr);  
  
    emit signalNewUser(user,  
        QDateTime::currentDateTime().toString("yyyy-MM-dd hh:mm:ss"));  
  
    QHostInfo::lookupHost(user, this, SLOT(userLookedUp(QHostInfo)));  
}
```

Aktualizujemy ilość danych wysłanych przez użytkownika oraz wysłanych z całej sieci.

```
int user = usersList.indexOf(sIP, 0);  
usersUp[user]+=length;  
  
netUpTotal+=length;
```

Sprawdzamy czy użytkownik wcześniej już się kontaktował z docelowym hostem. Jeśli nie to dodajemy hosta do listy, a następnie aktualizujemy odpowiednie dane (godzina odwiedzin, numer portu, ilość przesłanych danych itp.).

```
if (!usersHosts.at(user).hostIp.contains(dIP))
{
    usersHosts[user].hostIp.append(dIP);
    usersHosts[user].hostName.append("");

    addr.S_un.S_addr = dIP;
    QHostInfo::lookupHost(inet_ntoa(addr), this,
                          SLOT(hostLookedUp(QHostInfo)));

    usersHosts[user].dPort.append(QString::number(dPort));
    usersHosts[user].dApp.append(portToName(dPort));
    usersHosts[user].downBytes.append(0);
    usersHosts[user].upBytes.append(0);
    usersHosts[user].firstVisit.append(
        QDateTime::currentDateTime().toString("yyyy-MM-dd hh:mm:ss"));
    usersHosts[user].lastVisit.append(
        QDateTime::currentDateTime().toString("yyyy-MM-dd hh:mm:ss"));

    emit signalNewUserHost(user, usersHosts.at(user));
}

int index = usersHosts.at(user).hostIp.indexOf(dIP, 0);
usersHosts[user].upBytes[index]+=length;
usersHosts[user].lastVisit[index] =
    QDateTime::currentDateTime().toString("yyyy-MM-dd hh:mm:ss");
```

Ostatnia czynność to sprawdzenie czy aplikacja (port), z której korzysta użytkownik, znajduje się już na jego liście. Jeśli nie to dodajemy, a następnie aktualizujemy liczbę przesłanych danych (dodając aktualną liczbę wysłanych bajtów).

Aktualizujemy również listę z liczbą przesłanych w sieci typów pakietów.

```
if (dPort != 0)
{
    if (!usersApps.at(user).hostPort.contains(QString::number(dPort)))
    {
        usersApps[user].hostPort.append(QString::number(dPort));
        usersApps[user].hostPortName.append(portToName(dPort));
        usersApps[user].upBytes.append(0);
        usersApps[user].downBytes.append(0);

        emit signalNewUserApp(user, usersApps.at(user));
    }
    usersApps[user].upBytes[usersApps.at(user).hostPort.indexOf(
        QString::number(dPort), 0)]+=length;
}

incrementOutLists(type, user);
}
```

Wyświetlanie listy najbardziej aktywnych użytkowników

Po kliknięciu przez użytkownika na odpowiedni przycisk w głównym oknie aplikacji wywołana zostaje metoda *showTopActiveUpDlg()*. Odpowiedzialna jest ona za wyświetlenie okna dialogowego z listą najbardziej aktywnych użytkowników, osób wysyłających najwięcej danych z sieci.

W pierwszej kolejności sprawdzamy ile użytkowników znajduje się na liście, gdyż tylko tyle możemy wyświetlić (jednak nie więcej niż 25). Następnie wyświetlamy okno dialogowe z prośbą o podanie liczby użytkowników do wyświetlenia, ponieważ użytkownik aplikacji może chcieć wyświetlić ich mniej.

```
void MainWindow::showTopActiveUpDlg()
{
    bool ok;
    int max;
    ui.treeWidgetTransfer->topLevelItemCount() > 25 ? max = 25 :
                                                    max = ui.treeWidgetTransfer->topLevelItemCount();
    if (max == 0) return;
    int maxTop = QInputDialog::getInteger(this, tr("Select number of top active users"),
                                          tr("Users:"), 10, 1, max, 1, &ok);
    if (!ok) return;
```

Odłączamy sloty od sygnałów z *ReceiverCore* na czas obliczeń. Generujemy listę najbardziej aktywnych użytkowników.

```
disconnect(receiverCore, SIGNAL(signalUsersTransfer(QList<quint64>,QList<quint64>)),
            this, SLOT(usersTransfer(QList<quint64>,QList<quint64>)));
disconnect(receiverCore, SIGNAL(signalNetTransfer(quint64,quint64)), this,
            SLOT(netTransfer(quint64,quint64)));

quint64 maxValue;
QList<quint32> topUsersList;

for (int i = 0; i < maxTop; ++i)
    topUsersList.append(-1);

// find top users
for (int i = 0; i < maxTop; ++i)
{
    maxValue = 0;
    for (int j = 0; j < ui.treeWidgetTransfer->topLevelItemCount(); ++j)
    {
        if (usersUp.at(j) >= maxValue)
        {
            if (!topUsersList.contains(j))
            {
                maxValue = usersUp.at(j);
                topUsersList[i] = j;
            }
        }
    }
}
```

```
    }  
  }  
}
```

Tworzymy okno dialogowe, obiekt klasy *TopActiveDialog*. Ustawiamy jego nazwę, ikonę oraz pierwszy wiersz z sumą wszystkich wysłanych danych. Następnie w pętli podajemy kolejne wartości najbardziej aktywnych osób. Tak przygotowane okno wyświetlamy jako modalne wywołując metodę *exec()*.

```
TopActiveDialog dlg(this);  
dlg.setWindowIcon(QIcon(":/images/oBarsUp.png"));  
dlg.setWindowTitle(tr("Top %1 active users [uploaded]").arg(maxTop));  
dlg.setFirstItem(tr("Total uploaded on network"), bytesToStr(netUpTotal));  
  
for (int i = 0; i < maxTop; ++i)  
{  
    dlg.insertItem(i,  
        (ui.treeWidgetTransfer->topLevelItem(topUsersList.at(i))->text(0) + " " +  
        ui.treeWidgetTransfer->topLevelItem(topUsersList.at(i))->text(1)),  
        ui.treeWidgetTransfer->topLevelItem(topUsersList.at(i))->text(2),  
        usersUp.at(topUsersList.at(i)), netUpTotal);  
}  
  
dlg.exec();
```

Po zamknięciu okna przez użytkownika przywracamy połączenie slotów z sygnałami klasy *ReceiverCore*.

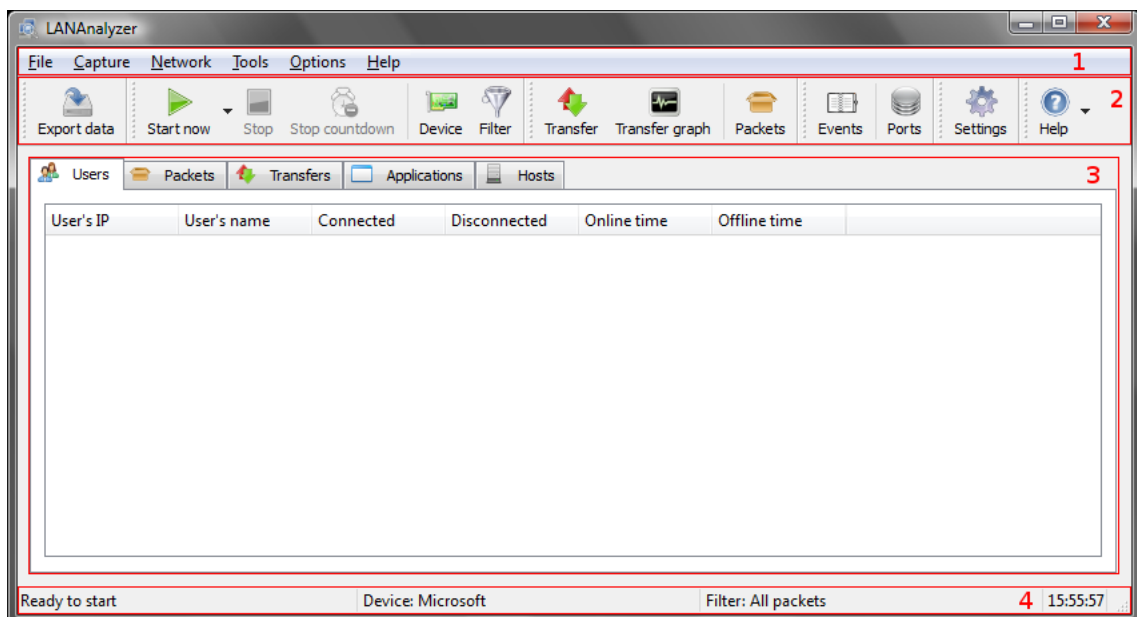
```
connect(receiverCore, SIGNAL(signalUsersTransfer(QList<quint64>,QList<quint64>)),  
        this, SLOT(usersTransfer(QList<quint64>,QList<quint64>)));  
connect(receiverCore, SIGNAL(signalNetTransfer(quint64,quint64)), this,  
        SLOT(netTransfer(quint64,quint64)));  
}
```


5. Opis interfejsu użytkownika

W tym rozdziale przedstawiono opis głównego okna aplikacji oraz najważniejszych funkcji. Szczegółowy opis wszystkich okien oraz wszystkich funkcji zawarty jest w pliku pomocy programu oraz w dodatku B tej pracy. Plik dostępny jest przez przycisk **Help** w menu oraz na paskach narzędziowych aplikacji. Interfejs programu (wraz ze wszystkimi komunikatami) domyślnie jest w języku angielski. Jednak poprzez ustawienia (**Settings** w menu **Options**) można zmienić domyślny język na polski (aktualne tłumaczenie w celach demonstracyjnych zawiera tylko przetłumaczone nazwy menu oraz informacje o tłumaczu).

5.1. Główne okno aplikacji

Na rys. 5.1. pokazano wygląd głównego okna aplikacji tuż po jej pierwszym uruchomieniu. W zależności od istniejących w systemie urządzeń sieciowych na pasku stanu, w polu **Device** (pol. *urządzenie*), może być inna nazwa urządzenia lub napis **not selected**, który oznacza, że nie wybrano żadnego urządzenia lub ich brak.



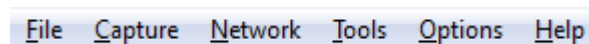
Rys. 5.1. Główne okno aplikacji po uruchomieniu

Główne okno aplikacji składa się z wielu dobrze znanych z innych programów elementów interfejsu użytkownika:

1. **Pasek menu** – zawiera większość opcji programu.
2. **Paski narzędziowe** – zawierają przyciski umożliwiające szybki dostęp do najczęściej używanych opcji z menu.
3. **Zakładki** – główny element okna, prezentują informacje o użytkownikach w sieci.
4. **Pasek stanu** – wyświetla informacje o stanie aplikacji, wybranym urządzeniu sieciowym, wybranym filtrze pakietów oraz aktualną godzinę.

5.1.1. Pasek menu

Pasek menu (rys. 5.2.) znajduje się zawsze u góry okna (rys. 5.1.). Pewne pozycje w menu mogą być nieaktywne. Oznacza to, że dana opcja jest niedostępna w danej chwili (np. **Start now** w menu **Capture**, gdy przechwytywanie właśnie jest uruchomione) lub w ogóle (np. **Minimize to tray**, gdy zasobnik systemowy jest niedostępny w systemie).



Rys. 5.2. Pasek menu

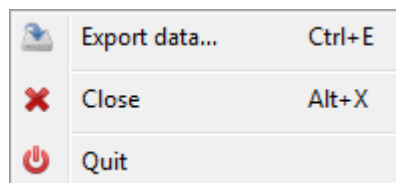
Pasek menu składa się z następujących menu:

1. **File** – zawiera przyciski do eksportu danych, zamknięcia okna aplikacji oraz zakończenia działania aplikacji.
2. **Capture** – tu uruchamiamy, zatrzymujemy przechwytywanie pakietów z sieci oraz wybieramy urządzenie sieciowe do tego celu jak również możemy tu ustawić filtr dla przechwytywanych pakietów.
3. **Network** – w tym menu uruchamiamy różne okna z informacjami na temat całej sieci (np. typy przechwyconych pakietów, przepływność danych w sieci) oraz okno do podglądu przechwytywanych pakietów.

4. **Tools** – zawiera przydatne narzędzia: podgląd zdarzeń aplikacji oraz edytor nazw aplikacji działających na poszczególnych portach w sieci.
5. **Options** – umożliwia zmianę w wyglądzie głównego okna (np. włączenie/wyłączenie paska stanu, zmianę wyglądu pasków narzędziowych), zmianę jego stanu (np. włączenie aplikacji na pełnym ekranie) oraz uruchomienie nowego okna dialogowego z ustawieniami aplikacji.
6. **Help** – umożliwia wyświetlenie pomocy aplikacji oraz informacji o niej.

Menu *File*

Wygląd menu **File** przedstawia rys. 5.3.



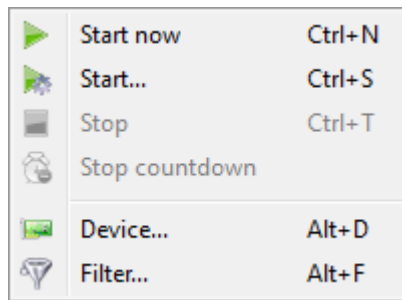
Rys. 5.3. Menu *File*

Poszczególne pozycje mają przypisane następujące funkcje:

- **Export data...** – wyświetla nowe okno dialogowe z opcjami eksportu danych widocznych w zakładkach.
- **Close** – zamyka główne okno i aplikację. Jeśli w opcjach zaznaczono minimalizowanie aplikacji do ikony w zasobniku systemowym podczas zamykania to okno zostaje zamknięte oraz, jeśli nie ma, zostaje utworzona ikona w zasobniku. Jeśli nie można utworzyć ikony to okno nie zostaje zamknięte i nie się nie zmienia.
- **Quit** – natychmiastowe i bezwzględne zamknięcie, zakończenie działania aplikacji.

Menu *Capture*

Wygląd menu **Capture** przedstawia rys. 5.4.



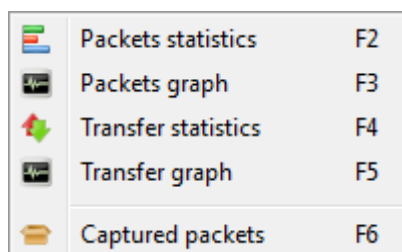
Rys. 5.4. Menu *Capture*

Poszczególne pozycje mają przypisane następujące funkcje:

- **Start now** – uruchamia przechwytywanie pakietów w sieci za pomocą wybranego urządzenia oraz ustawionego filtra pakietów.
- **Start...** – wyświetla okno dialogowe z opcjami przechwytywania, takimi jak czas startu, czas zatrzymania oraz innymi bardziej zaawansowanymi opcjami.
- **Stop** – zatrzymuje przechwytywanie pakietów.
- **Stop countdown** – zatrzymuje odliczanie do rozpoczęcia przechwytywania w przypadku, gdy wybrano odpowiednią opcję w oknie startu przechwytywania.
- **Device...** – wybór urządzenia do przechwytywania. Wyświetla okno dialogowe z urządzeniami sieciowymi dostępnymi na danym komputerze, które mogą zostać użyte do przechwytywania pakietów oraz informacje o nich.
- **Filter...** – wybór filtra do użycia podczas przechwytywania. Wyświetla okno dialogowe z listą filtrów oraz umożliwia ich edycję, dodawanie, usuwanie.

Menu *Network*

Wygląd menu **Network** przedstawia rys. 5.5.



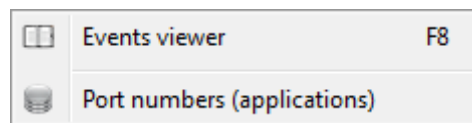
Rys. 5.5. Menu *Network*

Poszczególne pozycje mają przypisane następujące funkcje:

- **Packets statistics** – wyświetla okno dialogowe z informacją o liczbie przechwyconych pakietów i ich typie.
- **Packets graph** – wyświetla okno dialogowe z wykresem liczby przechwyconych pakietów na sekundę.
- **Transfer statistics** – wyświetla okno dialogowe z informacją o liczbie wysłanych i odebranych bajtów oraz aktualną szybkość transmisji w sieci.
- **Transfer graph** – wyświetla okno dialogowe z wykresem wysłanych oraz odebranych bajtów na sekundę w sieci.
- **Captured packets** – wyświetla nowe okno, w którym można podejrzeć wszystkie przechwycone oraz aktualnie przechwytywane pakiety.

Menu *Tools*

Wygląd menu **Tools** przedstawia rys. 5.6.



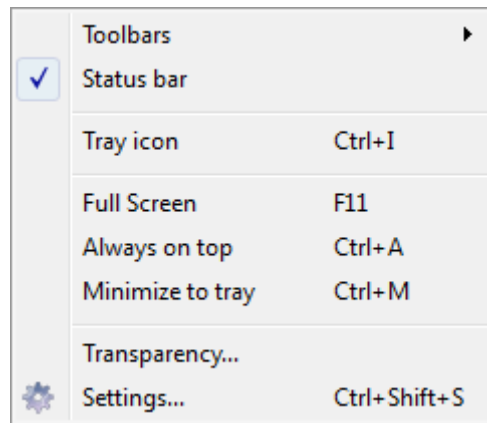
Rys. 5.6. Menu *Tools*

Poszczególne pozycje mają przypisane następujące funkcje:

- **Events viewer** – wyświetla nowe okno z podglądem zdarzeń, jakie miały miejsce w aplikacji.
- **Port numbers (applications)** – wyświetla okno dialogowe z listą portów oraz odpowiadających im aplikacji. Umożliwia ich edycję (dodawanie, usuwanie, kasowanie) oraz wyszukiwanieżądanego portu czy aplikacji.

Menu *Options*

Wygląd menu **Options** przedstawia rys. 5.7.



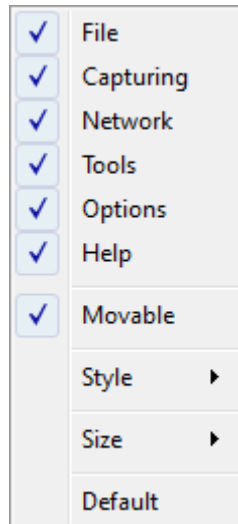
Rys. 5.7. Menu *Options*

Poszczególne pozycje mają przypisane następujące funkcje:

- **Toolbars** – podmenu. Zawiera opcje pasków narzędziowych.
- **Status bar** – pokazuje/ukrywa pasek stanu.
- **Tray icon** – pokazuje/ukrywa ikonę w zasobniku systemowym (jeśli jest dostępny).
- **Full Screen** – włącza/wyłącza tryb pełnego ekranu aplikacji.
- **Always on top** – ustawia widoczność aplikacji zawsze na wierzchu.
- **Minimize to tray** – minimalizuje aplikację do ikony w zasobniku systemowym (jeśli jest dostępny).
- **Transparency...** – wyświetla okno dialogowe umożliwiające zmianę przezroczystości głównego okna aplikacji.
- **Settings...** – wyświetla okno dialogowe z ustawieniami aplikacji.

Podmenu *Toolbars*

Wygląd podmenu **Toolbars** przedstawia rys. 5.8.



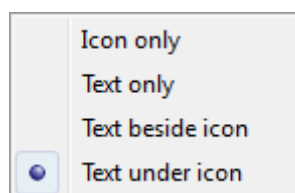
Rys. 5.8. Podmenu *Toolbars*

Poszczególne pozycje mają przypisane następujące funkcje:

- **File** – włącza/wyłącza pasek narzędziowy **File**.
- **Capturing** – włącza/wyłącza pasek narzędziowy **Capturing**.
- **Network** – włącza/wyłącza pasek narzędziowy **Network**.
- **Tools** – włącza/wyłącza pasek narzędziowy **Tools**.
- **Options** – włącza/wyłącza pasek narzędziowy **Options**.
- **Help** – włącza/wyłącza pasek narzędziowy **Help**.
- **Movable** – włącza/wyłącza możliwość przemieszczania pasków narzędziowych.
- **Style** – podmenu. Zawiera opcje zmiany stylu wyświetlania pasków narzędziowych.
- **Size** – podmenu. Zawiera opcje zmiany rozmiaru ikon.
- **Default** – resetuje wszystkie paski narzędziowe do domyślnego wyglądu.

Podmenu *Style*

Wygląd podmenu **Style** przedstawia rys. 5.9.



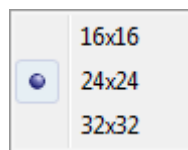
Rys. 5.9. Podmenu *Style*

Poszczególne pozycje mają przypisane następujące funkcje:

- **Icon only** – paski narzędziowe będą posiadały tylko ikony.
- **Text only** – paski narzędziowe będą posiadały tylko tekst.
- **Text beside icon** – paski narzędziowe będą posiadały tekst obok ikon.
- **Text under icon** – paski narzędziowe będą posiadały tekst pod ikonami.

Podmenu *Size*

Wygląd podmenu **Size** przedstawia rys. 5.10.



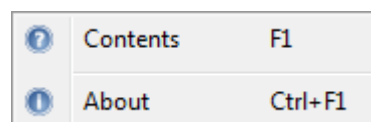
Rys. 5.10. Podmenu *Size*

Poszczególne pozycje mają przypisane następujące funkcje:

- **16x16** – zmienia rozmiar ikon wszystkich pasków narzędziowych na 16 x 16 pikseli.
- **24x24** – zmienia rozmiar ikon wszystkich pasków narzędziowych na 24 x 24 pikseli.
- **32x32** – zmienia rozmiar ikon wszystkich pasków narzędziowych na 32 x 32 pikseli.

Menu *Help*

Wygląd menu **Help** przedstawia rys. 5.11.



Rys. 5.11. Menu *Help*

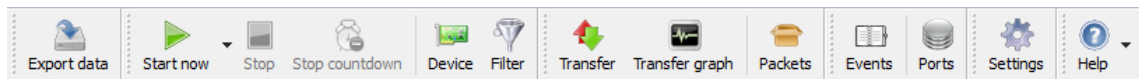
Poszczególne pozycje mają przypisane następujące funkcje:

- **Contents** – wyświetla pomoc programu.
- **About** – wyświetla okno dialogowe z informacjami o aplikacji.

5.1.2. Paski narzędziowe

Wygląd wszystkich pasków narzędziowych pokazano na rys. 5.12. Jak już wcześniej wspomniano zawierają one niektóre najczęściej używane polecenia dostępne w menu. Nazwy pasków odpowiadają nazwom menu, których opcje zawierają. Poprzez podmenu **Toolbars** w menu **Options** można dostosować wygląd pasków. Dostępne są następujące paski narzędziowe:

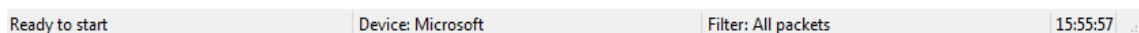
- **File**
- **Capture**
- **Network**
- **Tools**
- **Options**
- **Help**



Rys. 5.12. Paski narzędziowe

5.1.3. Pasek stanu

Pasek stanu wyświetlany jest na samym dole głównego okna aplikacji. Można go włączyć/wyłączyć poprzez opcję **Status bar** w menu **Options**. Przykładowy wygląd paska pokazuje rys. 5.13.



Rys. 5.13. Pasek stanu

Pasek stanu składa się z następujących pól (zaczynając od lewej strony):

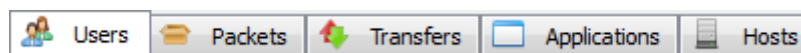
- **Stan programu** – wyświetla informacje o aktualnym stanie programu.
- **Urządzenie** – wyświetla nazwę aktualnie wybranego urządzenia sieciowego.

- **Filtr** – wyświetla nazwę aktualnie wybranego filtra pakietów.
- **Godzina** – wyświetla aktualną godzinę.

5.1.4. Zakładki

Zakładki to centralna, główna część aplikacji. Za pomocą zakładek wybieramy aktualnie prezentowane dane dotyczące użytkowników w sieci. Na rys. 5.14. zostały pokazane wszystkie dostępne zakładki:

- **Users** – użytkownicy sieci.
- **Packets** – rodzaje oraz liczba pakietów przesłanych przez użytkowników.
- **Transfers** – ilość wysłanych i odebranych danych oraz szybkość transmisji.
- **Applications** – ilość danych przesłanych przez aplikacje użytkowników.
- **Hosts** – ilość danych przesłanych przez użytkowników dla poszczególnych serwerów.



Rys. 5.14. Zakładki

5.2. Zakładka *Users*

Zakładka **Users** (pol. *użytkownicy*) została pokazana na rys. 5.15. Na tej zakładce można zobaczyć użytkowników sieci, a dokładnie ich adresy IP oraz nazwy przypisane komputerom. Dodatkowo wyświetlane są dane o aktywności użytkowników.

User's IP	User's name	Connected	Disconnected	Online time	Offline time
192.168.1.100	CHRYSLER	2010-05-26 11:38:54			
192.168.1.103	Lailoken	2010-05-26 11:39:02			
192.168.1.104	SIRRAS	2010-05-26 11:39:28			
192.168.1.1		2010-05-26 11:42:13			
192.168.1.106		2010-05-26 11:42:14			
192.168.1.107	MORRISONKA	2010-05-26 11:46:02			
192.168.1.108		2010-05-26 12:28:28			

Rys. 5.15. Zakładka *Users*

Znaczenie poszczególnych kolumn:

- **User's IP** – adres IP użytkownika.
- **User's name** – nazwa komputera użytkownika.
- **Connected** – data oraz godzina podłączenia się użytkownika.
- **Disconnected** – data oraz godzina odłączenia się użytkownika.
- **Online time** – czas, jaki użytkownik był ostatnio podłączony do sieci.
- **Offline time** – czas, jaki upłynął od ostatniego odłączenia się użytkownika.

5.3. Zakładka *Packets*

Zakładka **Packets** (pol. *pakiety*) zawiera informacje o liczbie pakietów przesyłanych przez użytkowników. Zakładka **Packets** została pokazana na rys. 5.16.

User's IP	User's name	ARP [total]	RARP [total]	ICMP [total]	IGMP [total]	TCP [total]	UDP [total]	OTHER [total]	ALL [total]
192.168.1.100	CHRYSLER	0	0	205	0	32374	2430	0	35009
192.168.1.103	Lailoken	0	0	0	0	124855	27	0	124882
192.168.1.104	SIRRAS	0	0	719	0	7489	24	0	8232
192.168.1.1		0	0	0	0	0	8	0	8
192.168.1.106		3	0	596	0	12646	31	0	13276
192.168.1.107	MORRISONKA	3	0	0	0	1742	25	0	1770
192.168.1.108		0	0	686	0	26060	19786	0	46532

Rys. 5.16. Zakładka *Packets*

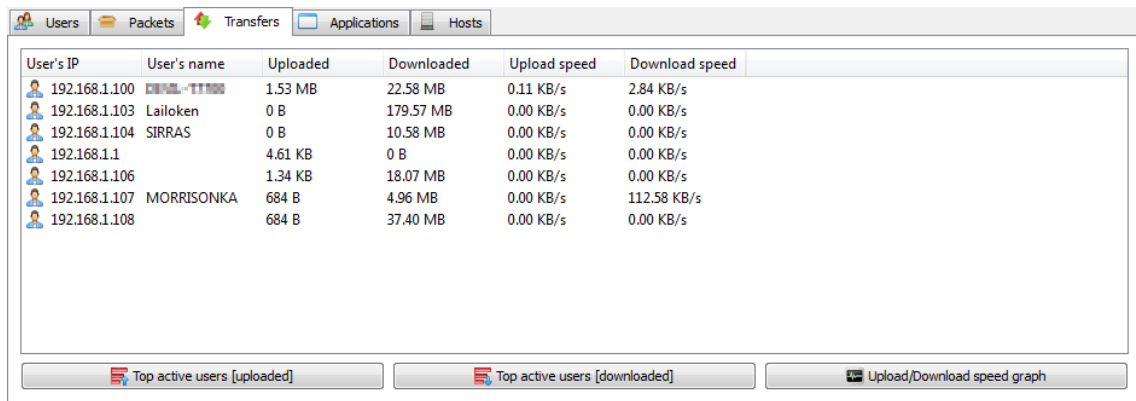
Znaczenie poszczególnych kolumn:

- **User's IP** – adres IP użytkownika.
- **User's name** – nazwa komputera użytkownika.
- **ARP [total], RARP [total], ICMP [total], IGMP [total], TCP [total], UDP [total], OTHER [total], ALL [total]** – suma liczby wysłanych oraz odebranych pakietów danego typu, pozostałych oraz ich suma.
- **ARP [in], RARP [in], ICMP [in], IGMP [in], TCP [in], UDP [in], OTHER [in], ALL [in]** – liczba odebranych pakietów danego typu, pozostałych oraz ich suma.
- **ARP [out], RARP [out], ICMP [out], IGMP [out], TCP [out], UDP [out], OTHER [out], ALL [out]** – liczba wysłanych pakietów danego typu, pozostałych oraz ich suma.

Przyciski **In**, **Out**, **Total** pozwalają na wybór prezentowanych danych. **In** to pakiety odebrane przez danego użytkownika. **Out** to pakiety wysłane przez danego użytkownika. **Total** to suma pakietów odebranych oraz wysłanych przez danego użytkownika.

5.4. Zakładka *Transfers*

Zakładka **Transfers** (pol. *transfery*) została pokazana na rys. 5.17. Na tej zakładce wyświetlane są dane o liczbie wysłanych i odebranych bajtów oraz aktualnej prędkości wysyłania i odbierania danych przez poszczególnych użytkowników.



User's IP	User's name	Uploaded	Downloaded	Upload speed	Download speed
192.168.1.100	192.168.1.100	1.53 MB	22.58 MB	0.11 KB/s	2.84 KB/s
192.168.1.103	Lailoken	0 B	179.57 MB	0.00 KB/s	0.00 KB/s
192.168.1.104	SIRRAS	0 B	10.58 MB	0.00 KB/s	0.00 KB/s
192.168.1.1		4.61 KB	0 B	0.00 KB/s	0.00 KB/s
192.168.1.106		1.34 KB	18.07 MB	0.00 KB/s	0.00 KB/s
192.168.1.107	MORRISONKA	684 B	4.96 MB	0.00 KB/s	112.58 KB/s
192.168.1.108		684 B	37.40 MB	0.00 KB/s	0.00 KB/s

Rys. 5.17. Zakładka *Transfers*

Znaczenie poszczególnych kolumn:

- **User's IP** – adres IP użytkownika.
- **User's name** – nazwa komputera użytkownika.
- **Uploaded** – liczba wysłanych bajtów.
- **Downloaded** – liczba odebranych bajtów.
- **Upload speed** – prędkość wysyłania danych (bajtów na sekundę).
- **Download speed** – prędkość odbierania danych (bajtów na sekundę).

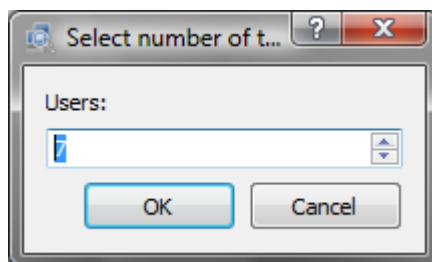
Dodatkowe informacje dostarczają okna znajdujące się pod widocznymi przyciskami:

- **Top active users [uploaded]** – lista osób, które najwięcej wysłały danych.
- **Top active users [downloaded]** – lista osób, które najwięcej odebrały danych.
- **Upload/Download speed graph** – wykres prędkości wysyłania/odbierania danych przez użytkowników.

5.4.1. Lista *Top active users (uploaded)*

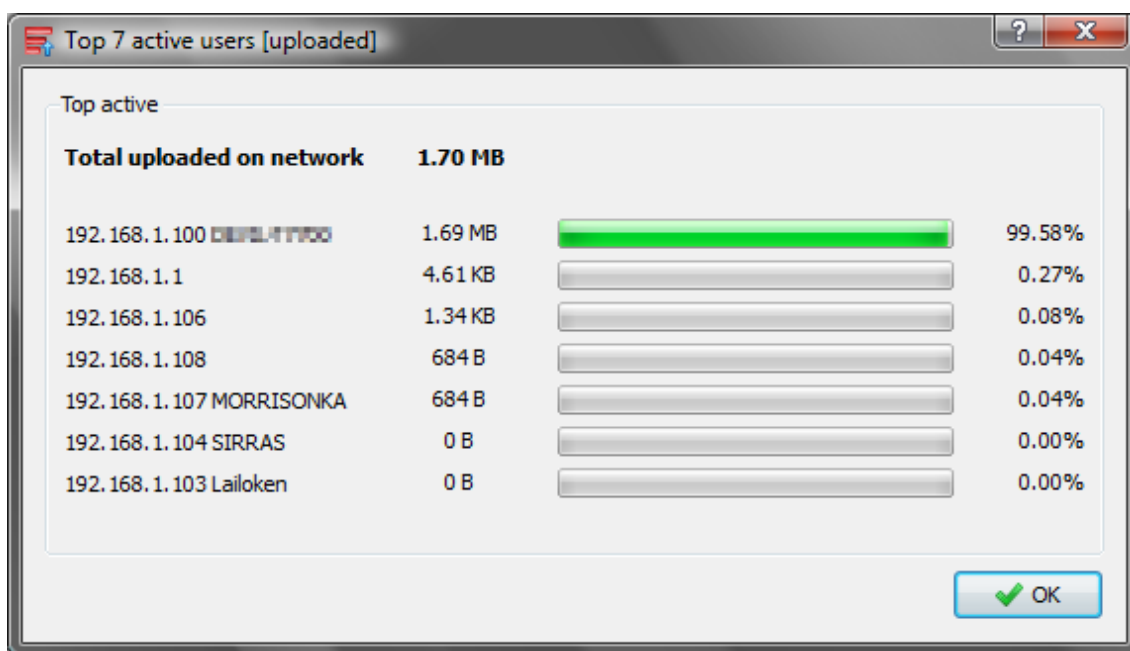
Kliknięcie na przycisk **Top active users [uploaded]** w pierwszej kolejności powoduje wyświetlenie okna dialogowego z pytaniem o maksymalną liczbę użytkowników do wyświetlenia (rys. 5.18.). Maksymalna liczba użytkowników na liście

nie może przekroczyć 25, a domyślna ich liczba to 10. Jednak, jeśli ich jest mniej to zostaje wyświetlona aktualna liczba wszystkich użytkowników.



Rys. 5.18. Okno dialogowe z prośbą o podanie liczby użytkowników

Po podaniu maksymalnej liczby użytkowników do wyświetlenia i kliknięciu na **OK** wyświetlone zostanie okno z listą najbardziej aktywnych użytkowników, osób wysyłających najwięcej danych (rys. 5.19.).

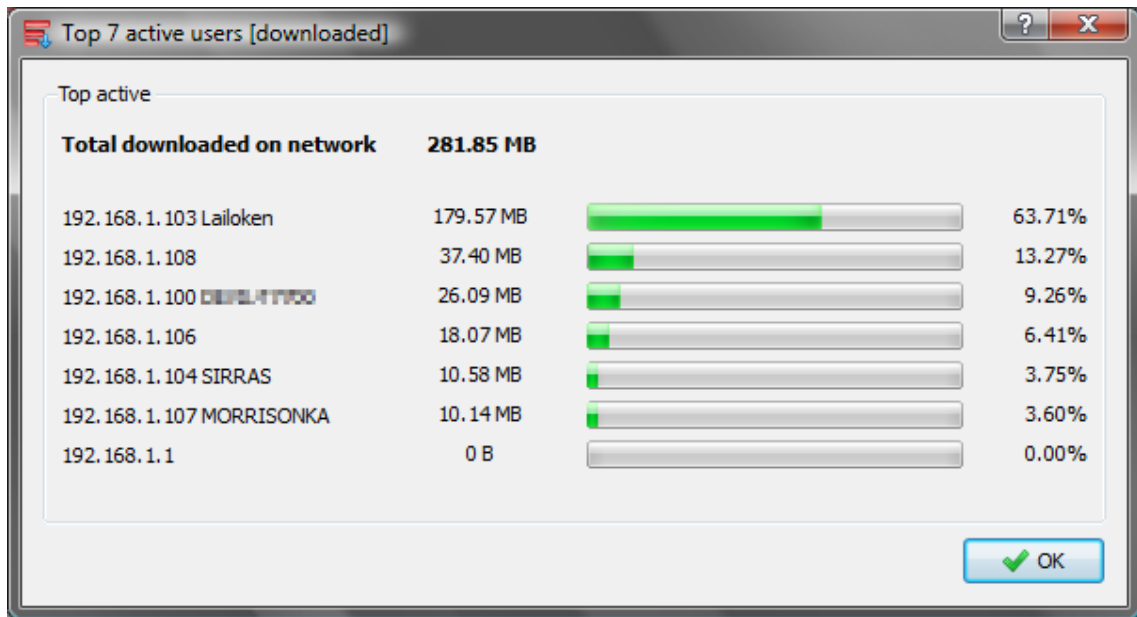


Rys. 5.19. Lista użytkowników wysyłających najwięcej danych

Pierwszy wiersz to liczba wysłanych bajtów z całej sieci (ang. *Total uploaded on network*). Kolejne wiersze to poszczególni użytkownicy, którzy wysłali najwięcej bajtów, oraz ich procentowy udział w liczbie wysłanych bajtów z całej sieci.

5.4.2. Lista *Top active users (downloaded)*

Kliknięcie na **Top active users [downloaded]** spowoduje analogiczne zachowanie jak w poprzednim przypadku (**Top active users [uploaded]**), z tą różnicą, że zostanie wyświetlona lista użytkowników odbierających najwięcej danych (rys. 5.20.).

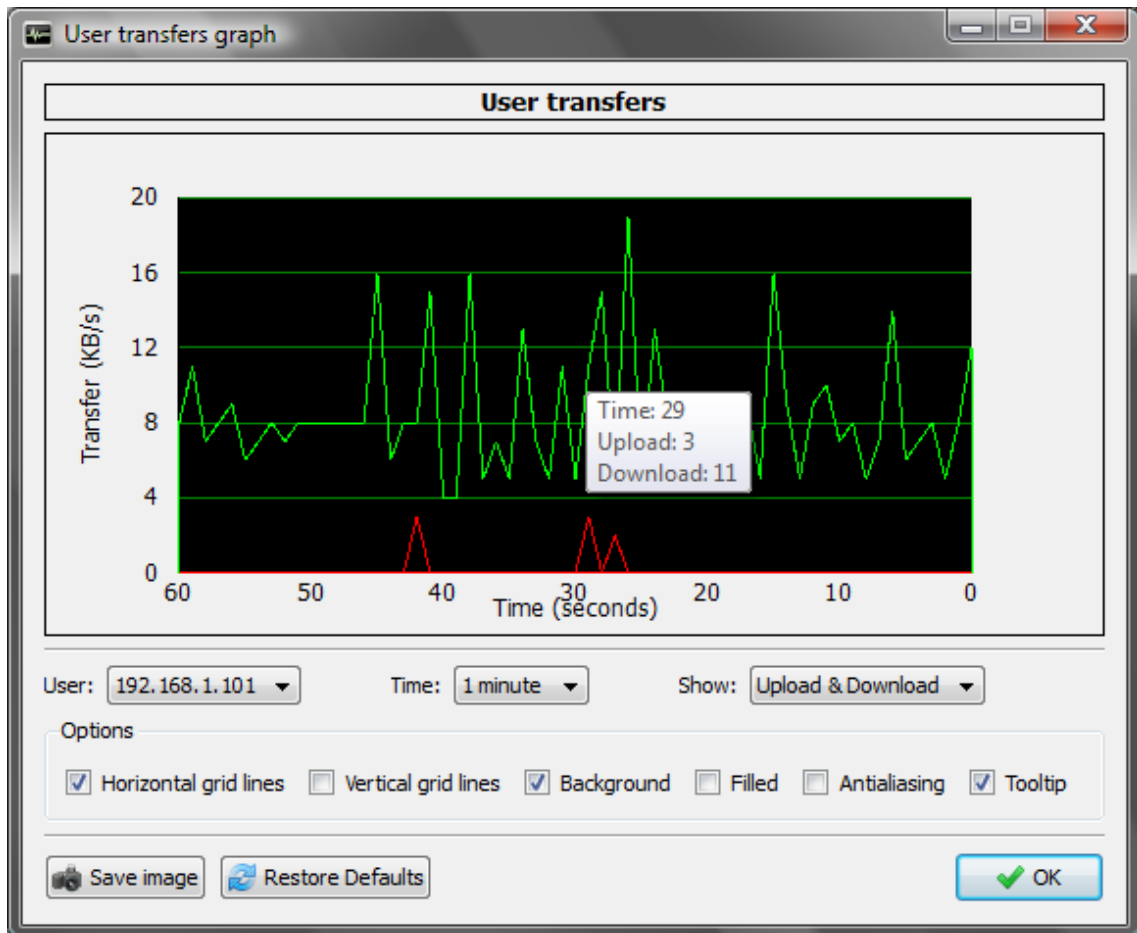


Rys. 5.20. Lista użytkowników odbierających najwięcej danych

Pierwszy wiersz to liczba odebranych bajtów w całej sieci (ang. *Total downloaded on network*). Kolejne wiersze to poszczególni użytkownicy, którzy odebrali najwięcej bajtów, oraz ich procentowy udział w liczbie odebranych bajtów w całej sieci.

5.4.3. Lista *Upload/Download speed graph*

Przykładowy wykres prędkości wysyłania/odbierania danych (przepływności danych) przez użytkowników został pokazany na rys. 5.21. Wykres ten pozwala sprawdzić jak bardzo poszczególni użytkownicy obciążają sieć, czyli np. czy obciążenie ma charakter chwilowy czy długotrwały (np. gdy pobierane są duże ilości danych), czy jest to duże obciążenie czy małe.

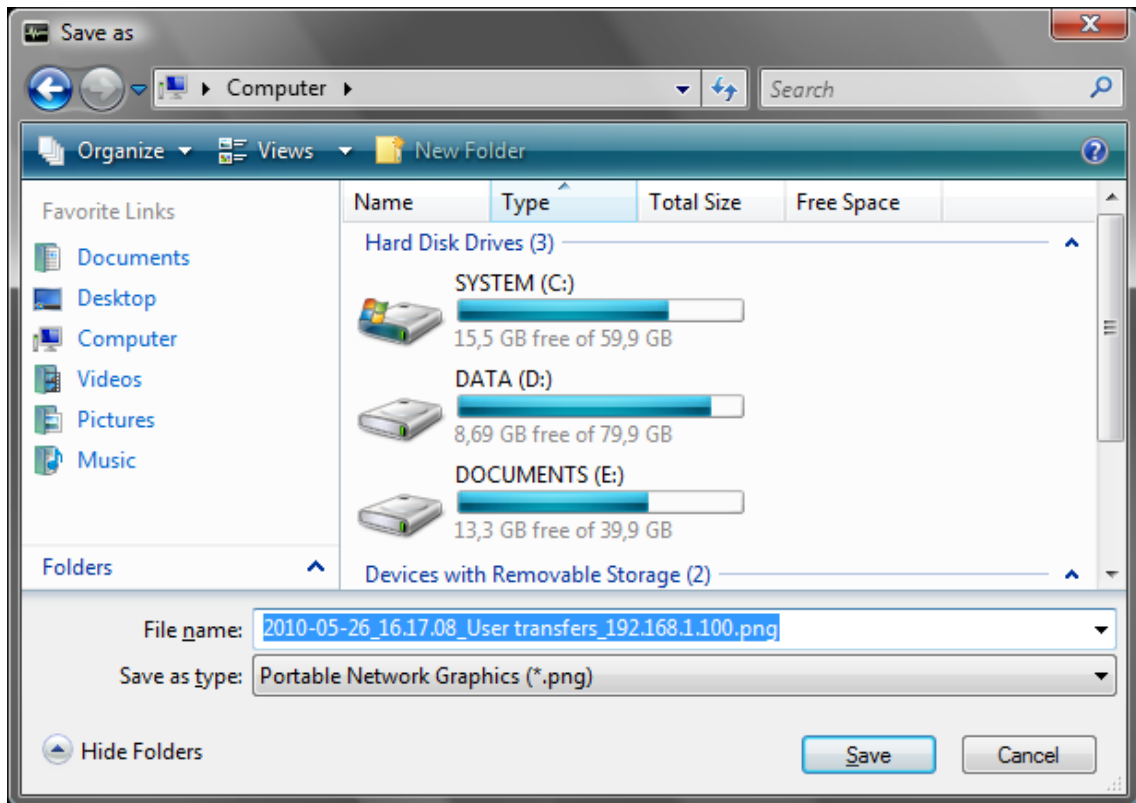


Rys. 5.21. Okno wykresu przepływności danych użytkowników

Znaczenie poszczególnych pól jest następujące:

- **User** – umożliwia wybór użytkownika, dla którego zostanie przedstawiony wykres przepływności.
- **Time** – określa ramy czasowe prezentowane na wykresie. Dostępne pozycje:
 - **1 minute** – na osi X przedstawione zostaną ostatnie 60 sekund, a na osi Y liczba przesłanych KB (kilobajtów) na sekundę.
 - **1 hour** – na osi X przedstawione zostaną ostatnie 60 minut, a na osi Y średnia liczba przesłanych KB (kilobajtów) na sekundę w danej minucie.
- **Show** – typ wyświetlanych danych. Dostępne opcje to:
 - **Upload & Download** – zostaną przedstawione równocześnie dane wysyłane oraz odbierane.
 - **Upload** – zostaną przedstawione tylko dane wysyłane – kolor czerwony.

- **Download** – zostaną przedstawione tylko dane odbierane – kolor zielony.
- Grupa **Options** umożliwia dostosowanie wyglądu wykresu do upodobań użytkownika. Dostępne opcje to:
 - **Horizontal grid lines** – wyświetla poziome linie ułatwiające odczyt wartości na wykresie. Ich liczba jest zmienna, zależna od maksymalnej wartości prezentowanej na wykresie. Wykres posiada autoskalowanie. Prezentowana jest zawsze najwyższa wartość liczby KB/s w danych ramach czasowych, które są wyświetlane (a nie maksymalna liczba, jaka była od momentu rozpoczęcia przechwytywania).
 - **Vertical grid lines** – wyświetla pionowe linie ułatwiające odczyt czasu na wykresie.
 - **Background** – wyświetla czarne tło pod wykresem.
 - **Filled** – wyświetla wykres z wypełnieniem.
 - **Antialiasing** – wygładza krawędzie linii wykresu.
 - **Tooltip** – wyświetla „dymek” z dokładną informacją o czasie oraz liczbie danych wysłanych oraz odebranych w danym miejscu na wykresie wskazanym przez kursor myszki.
- **Save image** – zapisuje zrzut wykresu w pliku graficznym. W zależności od wybranych opcji (w ustawieniach aplikacji) może zostać wyświetlone okno dialogowe jak np. na rys. 5.22. (ustawienie domyślne) lub plik zostanie zapisany bez jakichkolwiek pytań czy informacji (zgodnie z podanymi wcześniej ustawieniami).
- **Restore Defaults** – przywraca domyślne ustawienia wykresu. Nie wpływa na opcje zapisywania zrzutów.



Rys. 5.22. Zapisywanie wykresu

Wspomniane okno zapisywania wykresu (rys. 5.22.) otwiera się z opcjami (folder, nazwa pliku wraz z rozszerzeniem, typ pliku) ustawionymi w ustawieniach aplikacji, gdzie można też sprawdzić, jakie są domyślnie.

Dostępne formaty plików graficznych to:

- **Joint Photographic Experts Group (*.jpeg *.jpg)**
- **Portable Network Graphics (*.png)**
- **Scalable Vector Graphics (*.svg)**
- **Tagged Image File Format (*.tiff *.tif)**
- **Windows Bitmap (*.bmp)**

Jakość zapisywanych plików w formatach kompresji stratnej (**Joint Photographic Experts Group** oraz **Portable Network Graphics**) domyślnie jest wybierana automatycznie przez program.

5.5. Zakładka *Applications*

Zakładka **Application** (pol. *aplikacje*) została pokazana na rys. 5.23. Składa się ona z dwóch części. Po lewej stronie można zobaczyć użytkowników sieci, a dokładnie ich adresy IP oraz nazwy przypisane komputerom. Natomiast po prawej stronie wyświetlana jest lista aplikacji używanych przez aktualnie wybranego (w lewej części) użytkownika oraz ilość danych wysłanych i odebranych przez te aplikacje.

User's IP	User's name	Host's port	Application	Uploaded	Downloaded
192.168.1.100	DEVAL-TT00	80	World Wide Web HTTP	1.42 MB	23.05 MB
192.168.1.103	Lailoken	53	Domain Name Server	71.56 KB	121.59 KB
192.168.1.104	SIRRAS	137	NETBIOS Name Service	73.58 KB	0 B
192.168.1.1		8074	Gadu-Gadu	1.97 KB	2.37 KB
192.168.1.106		65536	Shirt Pocket netTunes	0 B	22.27 KB
192.168.1.107	MORRISONKA				
192.168.1.108					

Rys. 5.23. Zakładka *Applications*

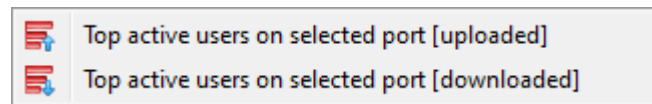
Znaczenie poszczególnych kolumn części lewej:

- **User's IP** – adres IP użytkownika.
- **User's name** – nazwa komputera użytkownika.

Znaczenie poszczególnych kolumn części prawej:

- **Host's port** – port, na jakim działa aplikacja.
- **Application** – nazwa aplikacji korzystającej z danego portu.
- **Uploaded** – ilość danych wysłanych przez daną aplikację użytkownika.
- **Downloaded** – ilość danych odebranych przez daną aplikację użytkownika.

W prawej części zakładki dostępne jest menu kontekstowe, widoczne na rys. 5.24. Jego pozycje odpowiadają funkcjom przycisków o tej samej nazwie.



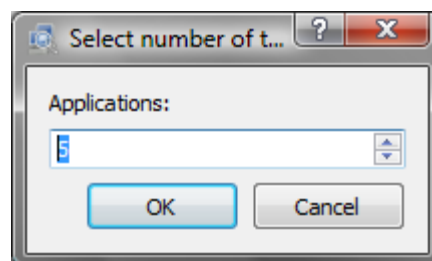
Rys. 5.24. Menu kontekstowe zakładki *Applications*

Dodatkowych informacji na tej zakładce dostarczają okna znajdujące się pod widocznymi przyciskami:

- **Top active ports (uploaded)** – lista aplikacji (portów) danego użytkownika, które najwięcej wysłały danych.
- **Top active ports (downloaded)** – lista aplikacji (portów) danego użytkownika, które najwięcej odebrały danych.
- **Top active users on selected port (uploaded)** – lista osób, które najwięcej wysłały danych na aktualnie zaznaczonym porcie (aplikacji).
- **Top active users on selected port (downloaded)** – lista osób, które najwięcej odebrały danych na aktualnie zaznaczonym porcie (aplikacji).

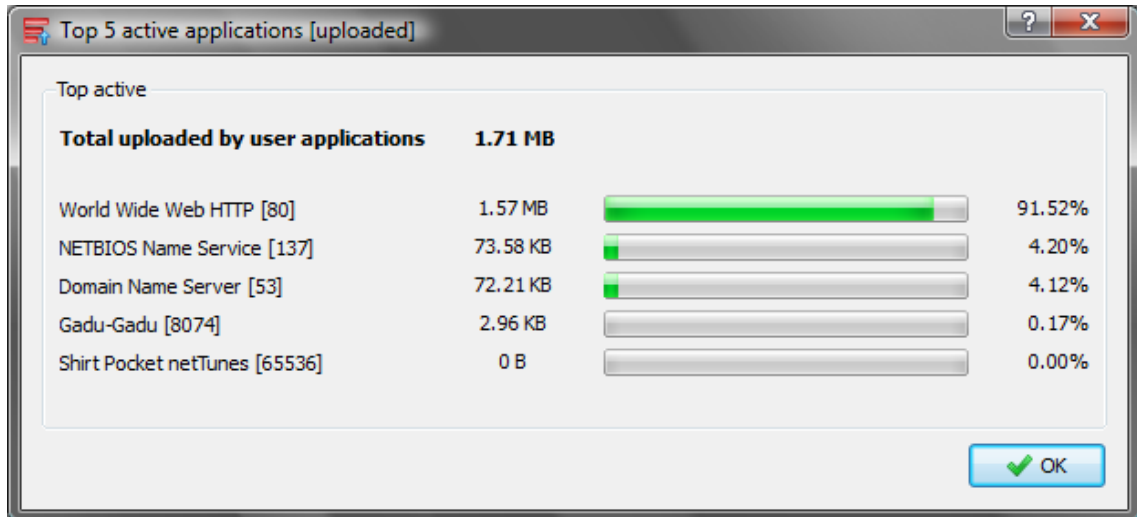
5.5.1. Lista *Top active ports (uploaded)*

Kliknięcie na przycisk **Top active ports (uploaded)** w pierwszej kolejności powoduje wyświetlenie okna dialogowego z pytaniem o maksymalną liczbę aplikacji danego użytkownika do wyświetlenia, co widać na rysunku 5.25. Maksymalna liczba aplikacji na liście nie może przekroczyć 25, a domyślna ich liczba to 10. Jednak, jeśli ich jest mniej to zostaje wyświetlona aktualna liczba wszystkich aplikacji.



Rys. 5.25. Okno dialogowe z prośbą o podanie liczby aplikacji

Po podaniu maksymalnej liczby aplikacji do wyświetlenia i kliknięciu na **OK** wyświetlone zostanie okno z listą aplikacji, wybranego użytkownika, wysyłających najwięcej danych (rys. 5.26.).

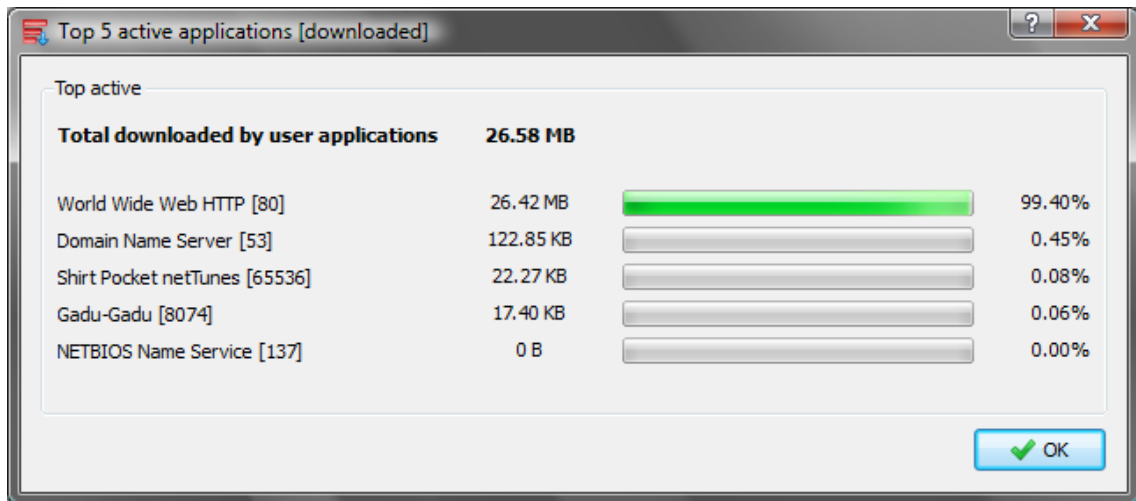


Rys. 5.26. Lista aplikacji użytkownika wysyłających najwięcej danych

Pierwszy wiersz to liczba wysłanych bajtów przez wszystkie aplikacje użytkownika. Kolejne wiersze to poszczególne aplikacje, które wysłały najwięcej bajtów, oraz ich procentowy udział w liczbie wysłanych bajtów przez wszystkie aplikacje.

5.5.2. Lista *Top active ports (downloaded)*

Kliknięcie na **Top active ports (downloaded)** spowoduje analogiczne zachowanie jak w poprzednim przypadku (**Top active ports (uploaded)**), z tą różnicą, że zostanie wyświetlona lista aplikacji, wybranego użytkownika, odbierających najwięcej danych (rys. 5.27.).

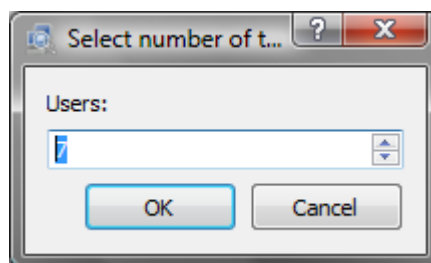


Rys. 5.27. Lista aplikacji użytkownika odbierających najwięcej danych

Pierwszy wiersz to liczba odebranych bajtów przez wszystkie aplikacje użytkownika. Kolejne wiersze to poszczególne aplikacje, które odebrały najwięcej bajtów, oraz ich procentowy udział w liczbie odebranych bajtów przez wszystkie aplikacje.

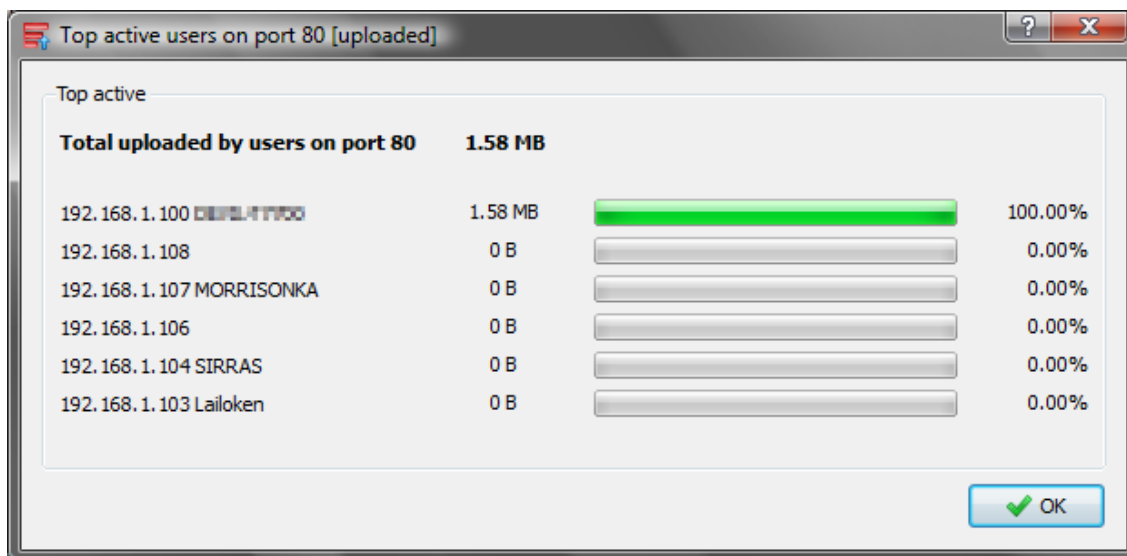
5.5.3. Lista *Top active users on selected port (uploaded)*

Klikając na przycisk **Top active users on selected port (uploaded)** w pierwszej kolejności powoduje wyświetlenie okna dialogowego z pytaniem o maksymalną liczbę użytkowników do wyświetlenia, co widać na rysunku 5.28. Maksymalna liczba użytkowników na liście nie może przekroczyć 25, a domyślna ich liczba to 10. Jednak, jeśli ich jest mniej to zostaje wyświetlona aktualna liczba wszystkich użytkowników.



Rys. 5.28. Okno dialogowe z prośbą o podanie liczby użytkowników

Po podaniu maksymalnej liczby użytkowników do wyświetlenia i kliknięciu na **OK** wyświetlone zostanie okno z listą osób, które najwięcej wysłały danych na aktualnie zaznaczonym porcie (aplikacji, rys. 5.29.).

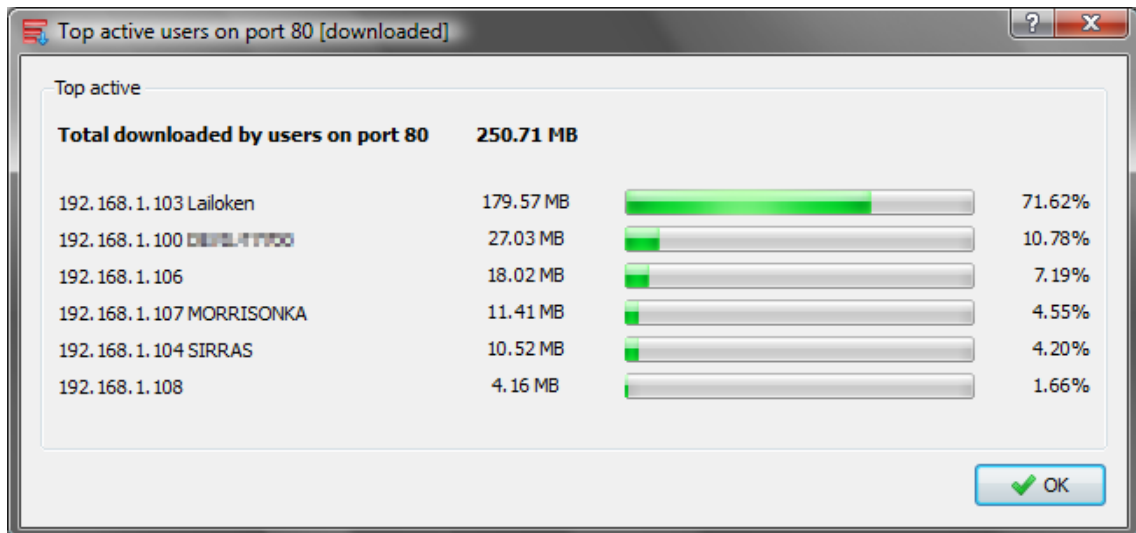


Rys. 5.29. Lista użytkowników, którzy najwięcej wysłali danych przez wybraną aplikację

Pierwszy wiersz to liczba wysłanych bajtów przez wszystkich użytkowników na wybranym porcie. Kolejne wiersze to poszczególni użytkownicy, którzy wysłali najwięcej bajtów, oraz ich procentowy udział w liczbie wysłanych bajtów przez wszystkich użytkowników.

5.5.4. Lista *Top active users on selected port (downloaded)*

Kliknięcie na **Top active users on selected port (downloaded)** spowoduje analogiczne zachowanie jak w poprzednim przypadku (**Top active users on selected port (uploaded)**), z tą różnicą, że zostanie wyświetlona lista osób, które najwięcej odebrały danych na aktualnie zaznaczonym porcie (rys. 5.30.).

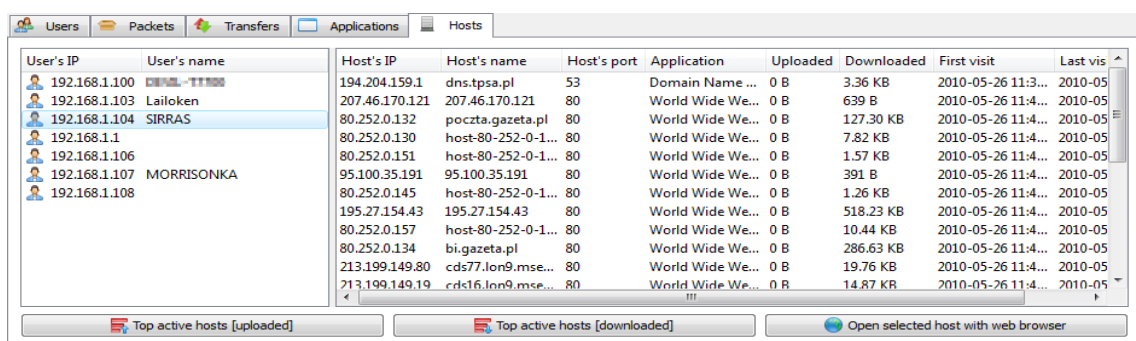


Rys. 5.30. Lista użytkowników, którzy najwięcej odebrali danych przez wybraną aplikację

Pierwszy wiersz to liczba odebranych bajtów przez wszystkich użytkowników na wybranym porcie. Kolejne wiersze to poszczególni użytkownicy, którzy odebrali najwięcej bajtów, oraz ich procentowy udział w liczbie odebranych bajtów przez wszystkich użytkowników.

5.6. Zakładka *Hosts*

Zakładka **Hosts** (pol. *hosty*) została pokazana na rys. 5.31. Tak jak poprzednia zakładka ta również składa się z dwóch części. Po lewej stronie można zobaczyć użytkowników sieci, a dokładnie ich adresy IP oraz nazwy przypisane komputerom. Natomiast po prawej stronie wyświetlana jest lista serwerów wraz z dodatkowymi informacjami, z którymi kontaktował się aktualnie wybrany (w lewej części) użytkownik.



Rys. 5.31. Zakładka *Hosts*

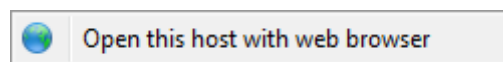
Znaczenie poszczególnych kolumn części lewej:

- **User's IP** – adres IP użytkownika.
- **User's name** – nazwa komputera użytkownika.

Znaczenie poszczególnych kolumn części prawej:

- **Host's IP** – adres IP hosta (serwera).
- **Host's name** – nazwa hosta.
- **Host's port** – port, na jakim nawiązano połączenie.
- **Application** – nazwa aplikacji korzystającej z danego portu.
- **Uploaded** – ilość danych wysłanych przez użytkownika.
- **Downloaded** – ilość danych odebranych przez użytkownika.
- **First visit** – data oraz godzina pierwszego połączenia z hostem.
- **Last visit** – data oraz godzina ostatniego połączenia z hostem.

W prawej części zakładki dostępne jest menu kontekstowe (rys. 5.32.). Posiada ono jedynie jedno polecenie, którego działanie odpowiada dokładnie funkcji przycisku o tej samej nazwie.



Rys. 5.32. Menu kontekstowe zakładki *Hosts*

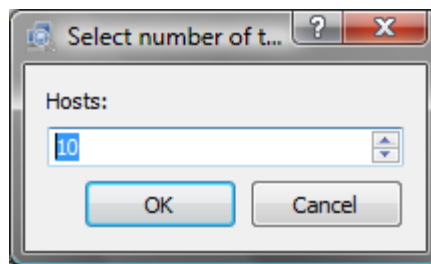
Dodatkowe informacje dostarczają okna znajdujące się pod widocznymi przyciskami:

- **Top active hosts [uploaded]** – lista hostów, z którymi kontaktował się aktualnie wybrany (w lewej części) użytkownik, do których wysłał najwięcej danych.
- **Top active hosts [downloaded]** – lista hostów, z którymi kontaktował się aktualnie wybrany (w lewej części) użytkownik, od których odebrał najwięcej danych.

- **Open selected host with web browser** – za pomocą domyślnej przeglądarki internetowej (w systemie użytkownika) otwiera stronę aktualnie zaznaczonego hosta.

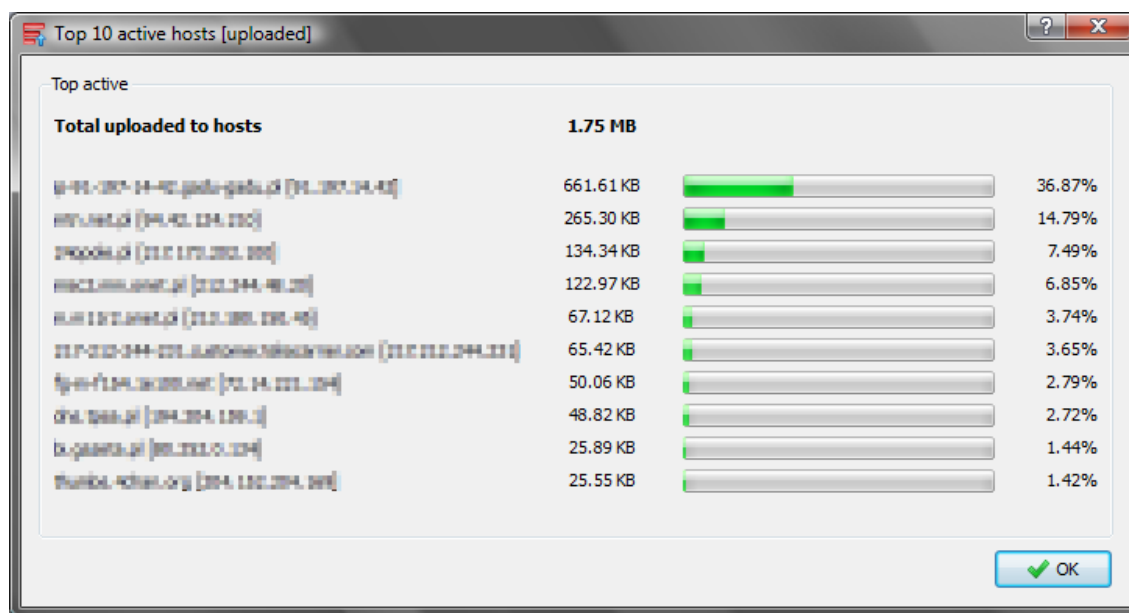
5.6.1. Lista *Top active hosts (uploaded)*

Kliknięcie na przycisk **Top active hosts [uploaded]** w pierwszej kolejności powoduje wyświetlenie okna dialogowego z pytaniem o maksymalną liczbę hostów danego użytkownika do wyświetlenia, co widać na rysunku 5.33. Maksymalna liczba hostów na liście nie może przekroczyć 25, a domyślna ich liczba to 10. Jednak, jeśli ich jest mniej to zostaje wyświetlona aktualna liczba wszystkich hostów.



Rys. 5.33. Okno dialogowe z prośbą o podanie liczby hostów

Po podaniu maksymalnej liczby hostów do wyświetlenia i kliknięciu na **OK** wyświetlone zostanie okno z listą hostów, z którymi kontaktował się aktualnie wybrany (w lewej części) użytkownik, do których wysłał najwięcej danych (rys. 5.34.).

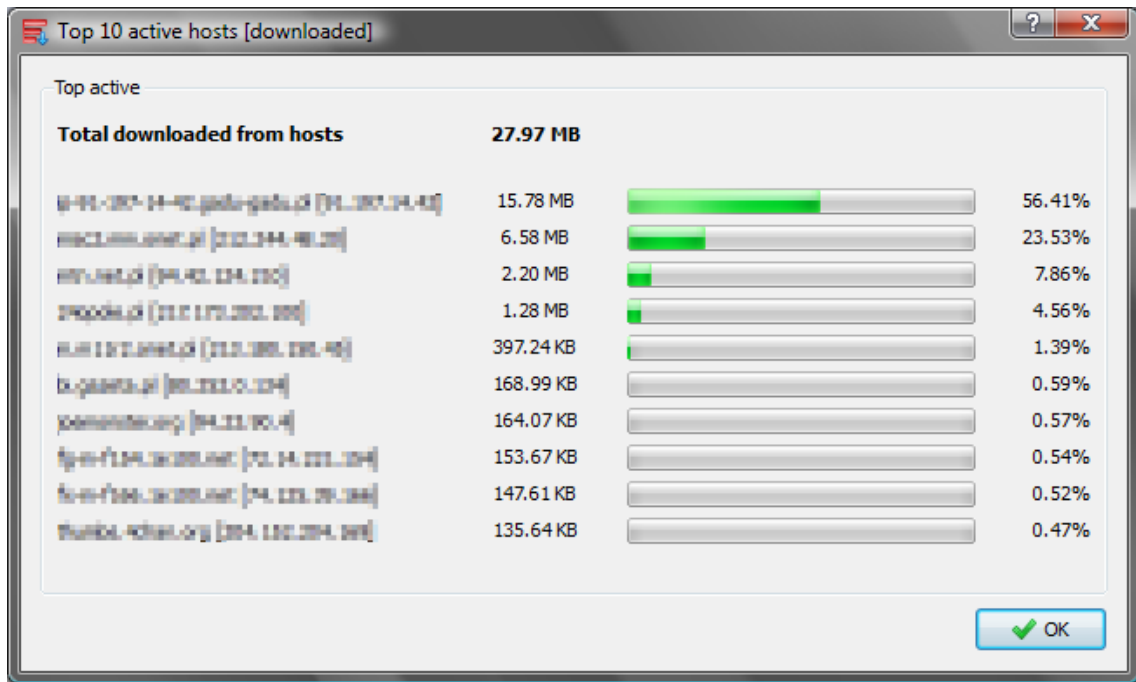


Rys. 5.34. Lista hostów użytkownika, do których wysłał najwięcej danych

Pierwszy wiersz to liczba wysłanych bajtów do wszystkich hostów przez wybranego użytkownika. Kolejne wiersze to poszczególne hosty, do których zostało wysłanych najwięcej bajtów, oraz ich procentowy udział w liczbie wysłanych bajtów do wszystkich hostów (przez wybranego użytkownika).

5.6.2. Lista *Top active hosts (downloaded)*

Kliknięcie na **Top active hosts [downloaded]** spowoduje analogiczne zachowanie jak w poprzednim przypadku (**Top active hosts [uploaded]**), z tą różnicą, że zostanie wyświetlona lista hostów, z którymi kontaktował się aktualnie wybrany (w lewej części) użytkownik, od których odebrał najwięcej danych (rys. 5.35.).

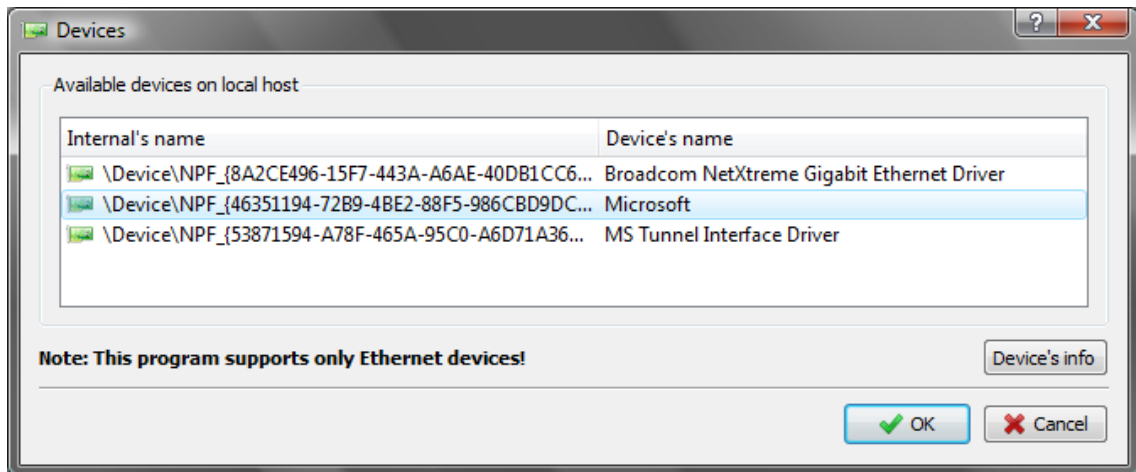


Rys. 5.35. Lista hostów użytkownika, od których odebrał najwięcej danych

Pierwszy wiersz to liczba odebranych bajtów od wszystkich hostów przez wybranego użytkownika. Kolejne wiersze to poszczególne hosty, od których zostało odebranych najwięcej bajtów, oraz ich procentowy udział w liczbie odebranych bajtów od wszystkich hostów (przez wybranego użytkownika).

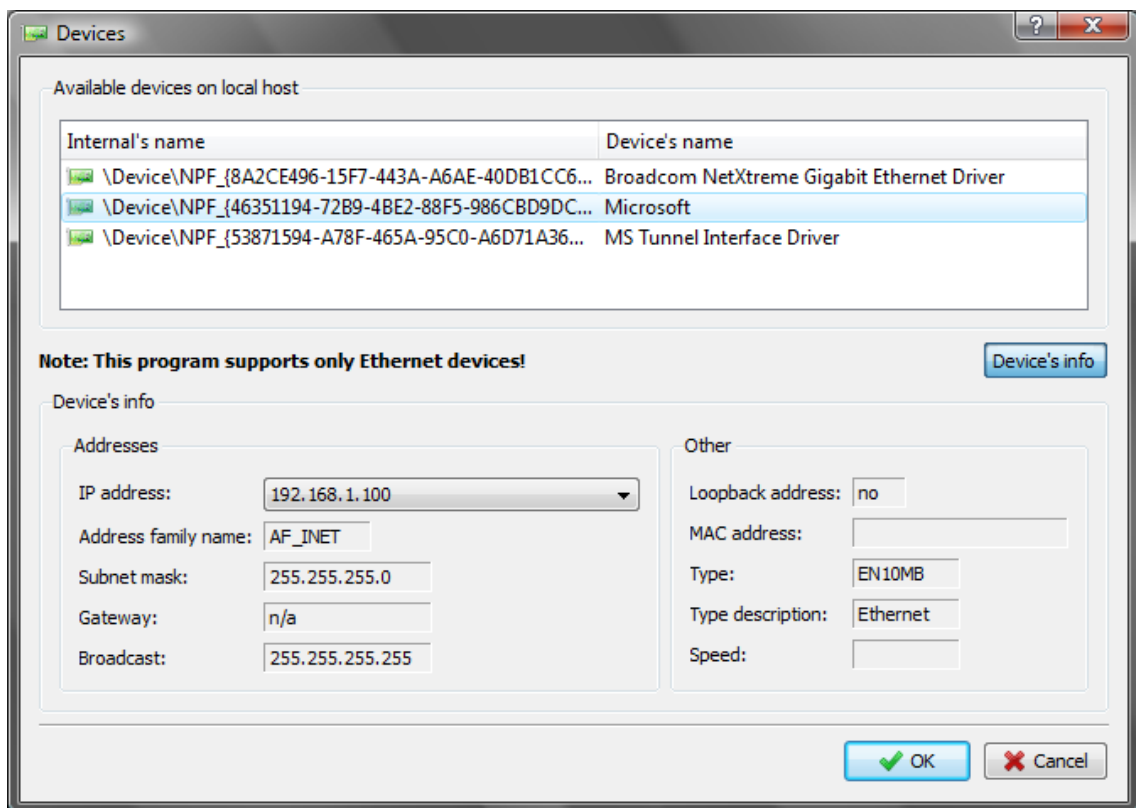
5.7. Wybór urządzenia

Wybór urządzenia sieciowego do przechwytywania ruchu sieciowego, a tym samym do analizowania pracy użytkowników, dokonujemy wybierając pozycję **Devie...** w menu **Capture** lub **Device** na pasku narzędziowym **Capture** lub z pomocą skrótu **Alt+D**. Okno wyboru urządzenia sieciowego domyślnie wygląda tak jak na rys. 5.36.



Rys. 5.36. Okno wyboru urządzenia sieciowego

Podstawowe okno wyboru urządzenia sieciowego (rys. 5.36.) wyświetla listę urządzeń dostępnych na lokalnym komputerze. Kolumna **Internal's name** to wewnętrzna nazwa urządzenia sieciowego. Jego bardziej przyjazna nazwa widoczna jest w kolumnie **Device's name** (o ile taka nazwa istnieje). Natomiast przyciskiem **Device's info** można powiększyć okno o dodatkowe, szczegółowe informacje o wybranym urządzeniu (rys. 5.37.).



Rys. 5.37. Okno wyboru urządzenia sieciowego wraz z informacjami o nim

Na rys. 5.37. widoczne jest rozszerzone okno wyboru urządzenia sieciowego. Oprócz listy urządzeń można zapoznać się również ze szczegółowymi informacjami o nim.

Grupa **Addresses** zawiera informacje o adresach IP przyznanych aktualnie zaznaczonemu urządzeniu:

- **IP address** – adres IP urządzenia. W zależności od wybranych opcji w ustawieniach aplikacji mogą tu być wyświetlane adresy IPv4 lub IPv6 lub obydwa lub żadne.
- **Address family name** – uogólniając jest to informacja o typie adresu IP: AF_INET to IPv4, a AF_INET6 to IPv6.
- **Subnet mask** – maska podsieci. Jest to liczba umożliwiająca wyodrębnienie z adresu IP części sieciowej od części hosta.
- **Gateway** – brama sieciowa, czyli adres IP urządzenia, za pomocą którego urządzenia sieci lokalnej komunikują się z innymi sieciami.
- **Broadcast** – adres rozgłoszeniowy sieci.

Grupa **Other** zawiera pozostałe informacje o aktualnie zaznaczonym urządzeniu:

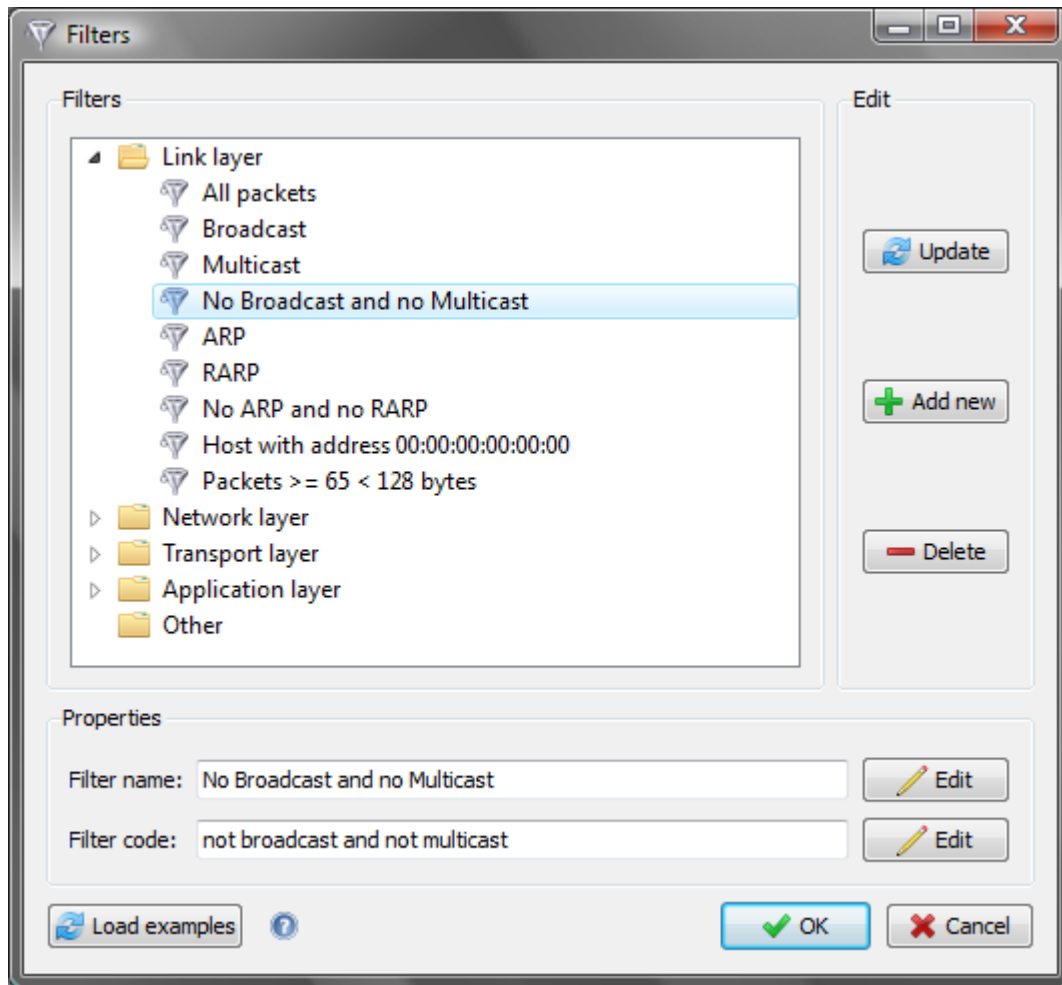
- **Loopback address** – informacja czy wybrany adres jest tzw. adresem pętli zwrotnej. Możliwe wartości w tym polu to **yes** lub **no**.
- **MAC address** – adres MAC, czyli adres sprzętowy urządzenia sieciowego nadawany przez producenta, unikatowy w skali światowej.
- **Type** – typ urządzenia. **EN10MB** oznacza, że jest to urządzenie zgodne z Ethernetem (10Mb, 100Mb, 1000Mb i szybszym) i można na nim przechwytywać ruch tym programem.
- **Type description** – bardziej przyjazna informacja o typie urządzenia (np. Ethernet dla EN10MB z pola **Type**).
- **Speed** – aktualna prędkość pracy urządzenia.

5.8. Wybór filtra pakietów

Okno wyboru filtra pakietów otwieramy wybierając **Filter...** w menu **Capture** lub **Filter** na pasku narzędziowym **Capture** lub za pomocą skrótu **Alt+F**. Wygląd okna z domyślną (przykładową) listą filtrów przedstawia rys. 5.38. Filtry dla ułatwienia zostały pogrupowane w pięć kategorii, które wynikają z warstwowej budowy modelu TCP/IP:

- **Link layer** – warstwa dostępu do sieci.
- **Network layer** – warstwa sieci.
- **Transport layer** – warstwa transportowa.
- **Application layer** – warstwa aplikacji.
- **Other** – pozostałe typy filtrów, np. mogące łączyć kilka warstw.

Kategorii nie można usunąć, ani dodać nowej. Nie można również zmienić ich nazwy. Każdy filtr składa się z nazwy oraz kodu. Nazwy widoczne są w drzewie filtrów, natomiast kod można zobaczyć w polu **Filter code** po wybraniu któregoś z filtrów. Szczegółowy opis składni oraz budowy filtrów znajduje się pod adresem: http://www.winpcap.org/docs/docs_411/html/group__language.html.



Rys. 5.38. Okno wyboru filtra

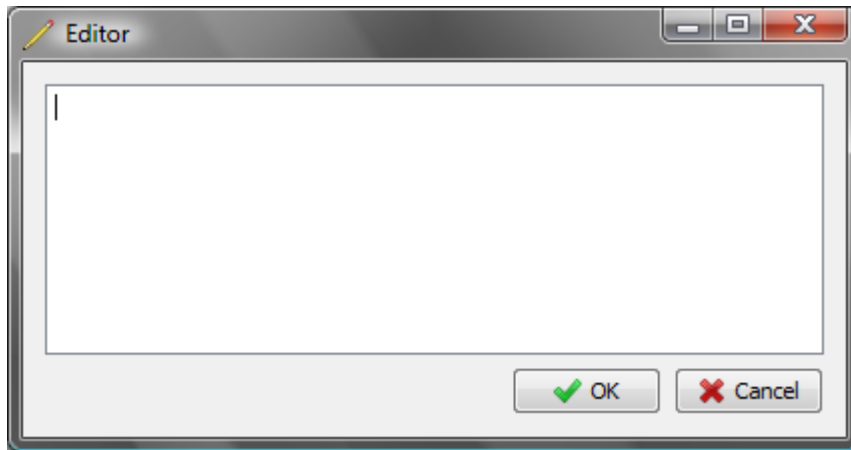
Przyciski w grupie **Edit** służą do edycji filtrów:

- **Update** – zapisuje zmiany wprowadzone w polach **Filter name** oraz **Filter code** dla wybranego filtra.
- **Add New** – dodaje do listy nową pozycję. Nowy filtr posiada nazwę oraz kod aktualnie zaznaczonego filtra, chyba że kategoria jest pusta lub po prostu jest zaznaczona jakaś kategoria. Wówczas nowy filtr posiada nazwę **All packets**, a pole kodu pozostaje puste.
- **Delete** – usuwa aktualnie zaznaczony filtr. Nie jest możliwe skasowanie kategorii.


Grupa **Properties** służy do zmian w składni oraz nazwie filtrów:

- **Filter name** – nazwa filtra taka sama jak w drzewie filtrów.

- **Filter code** – kod filtra.
- **Edit** – otwiera okno edytora tekstowego odpowiednio z nazwą lub kodem wybranego filtra. Przydatne dla bardzo długich nazw lub skomplikowanych kodów. Wygląd edytora przedstawia rys. 5.38.



Rys. 5.39. Okno edytora

Dodatkowy przycisk **Load examples** kasuje wszystkie aktualne filtry i ładuje domyślne. Operacja ta jest nieodwracalna. Natomiast przycisk  wyświetla listę dostępnych skrótów klawiaturowych przyspieszających oraz ułatwiających edycję filtrów. Dostępne skróty klawiaturowe to:

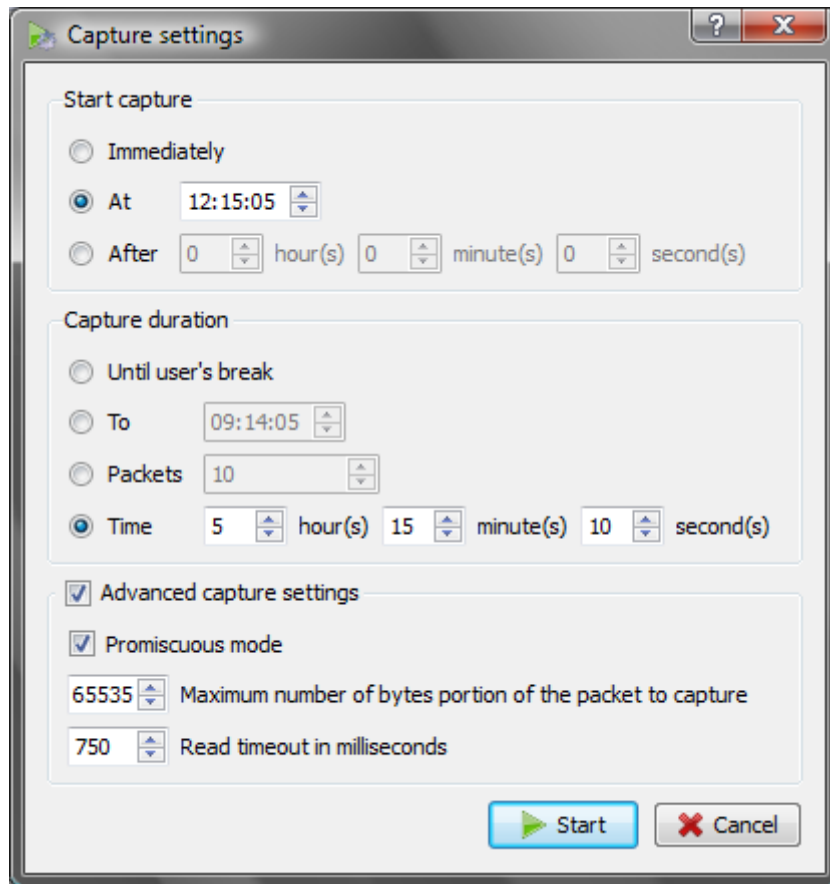
- **F2 – Edit Item** – staje się aktywne pole edycji nazwy filtra.
- **Insert – Add new Item** – odpowiada działaniu przyciskowi **Add New**.
- **Delete – Delete Item** – odpowiada działaniu przyciskowi **Delete**.

5.9. Start przechwytywania

Start przechwytywania pakietów możliwy jest na dwa sposoby (oba dostępne przez menu lub pasek narzędziowy **Capture**):

- **Start now – Ctrl+N** – uruchamia przechwytywanie pakietów w sieci za pomocą wcześniej wybranego urządzenia oraz ustawionego filtra pakietów.

- **Start...** – **Ctrl+S** – wyświetla okno dialogowe z opcjami przechwytywania (rys. 5.40.) takimi jak czas startu, czas zatrzymania oraz innymi bardziej zaawansowanymi opcjami.



Rys. 5.40. Okno dialogowe startu przechwytywania

Grupa **Start capture** określa, kiedy przechwytywanie powinno się rozpocząć. Dostępne opcje to:

- **Immediately** – przechwytywanie rozpocznie się natychmiast po kliknięciu na przycisk **Start**.
- **At** – przechwytywanie rozpocznie się o określonej godzinie (system 24-godzinny).
- **After** – przechwytywanie rozpocznie się po upływie podanej liczby godzin, minut, sekund.

Grupa **Capture duration** określa jak długo program ma przechwytywać pakiety:

- **Until user's break** – przechwytywanie będzie trwało do momentu ręcznego przerwania przez użytkownika.
- **To** – przechwytywanie będzie trwało do podanej godziny (system 24-godzinny).
- **Packets** – przechwytywanie będzie trwało do momentu przechwycenia podanej liczby pakietów. Maksymalna wartość to 2147483647 pakietów. Liczba przechwyconych pakietów określa w pewnym stopniu obciążenie sieci.
- **Time** – przechwytywanie będzie trwało podaną liczbę godzin, minut, sekund.

Zmiany wprowadzone w powyższych grupach będą zapamiętane jedynie po naciśnięciu na **Start** oraz tylko dla danej sesji programu. Po ponownym uruchomieniu aplikacji grupy te będą znów zresetowane do ustawień domyślnych.

Domyślnie parametry przechwytywania w grupie **Advanced capture settings** są optymalne i nie powinny być zmieniane, zwłaszcza przez osoby, które nie posiadają odpowiedniej wiedzy o ich przeznaczeniu. Z tego powodu jest ona domyślnie wyłączona. Po jej zaznaczeniu będzie można zmienić zaawansowane ustawienia przechwytywania pakietów:

- **Promiscuous mode** – tryb promiscuous (ang. *promiscuous mode* – tryb nasłuchiwania) to tryb pracy urządzenia sieciowego polegający na odbieraniu całego ruchu docierającego do tego urządzenia, nie tylko skierowanego na jego adres MAC. Odznaczenie tej opcji nie zawsze oznacza, że dane urządzenie sieciowe nie będzie działało w tym trybie, gdyż może się zdarzyć, że wcześniej inna aplikacja lub system przełączyły to urządzenie sieciowe na pracę w trybie promiscuous. Tryb ten jest konieczny dla analizowania pracy wszystkich użytkowników w lokalnej sieci.
- **Maximum number of bytes portion of the packet to capture** – maksymalna liczba bajtów przechwytywanych pakietów przekazywanych przez sterownik do aplikacji. 65535 bajtów zapewnia przechwytywanie pełnych, nawet największych pakietów.

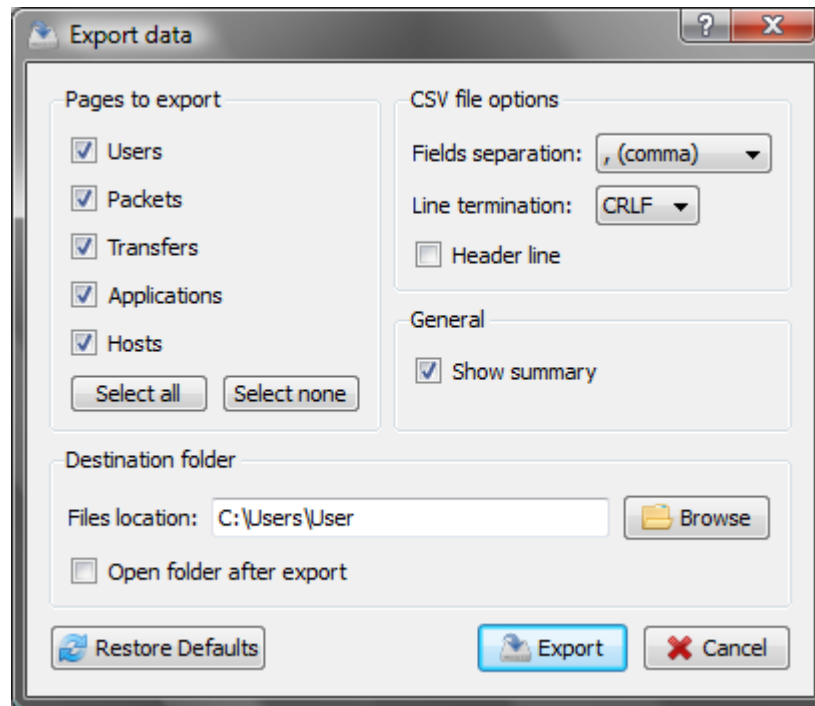
- **Read timeout in milliseconds** – czas przechwytywania pakietów przez sterownik przed przekazaniem ich aplikacji. 1000 ms oznacza, że sterownik będzie przez 1 sekundę przechwytywał i buforował pakiety, a po tym czasie przekaże je aplikacji do interpretacji. Ten parametr może być ignorowany przez niektóre systemy operacyjne.

Zmiany wprowadzone w grupie **Advanced capture settings** są zapisywane jedynie, gdy grupa ta jest zaznaczona oraz kliknięto na **Start**. Ustawienia są przechowywane po zamknięciu aplikacji.

5.10. Eksport danych

Eksportu danych znajdujących się w tabelach na poszczególnych zakładkach aplikacji dokonamy wybierając **Export data...** w menu **File** lub **Export data** na pasku narzędziowym **File** lub za pomocą skrótu **Ctrl+E**. Wygląd okna z domyślnymi parametrami przedstawia rys. 5.41.

Dane eksportowane są do plików tekstowych typu **CSV** (ang. *Comma Separated Values* – wartości rozdzielone przecinkiem). Format ten jest obsługiwany przez wiele aplikacji, jak choćby przez bardzo popularne Microsoft Excel czy Calc z pakietu OpenOffice.org. Nazwy zapisywanych plików to **data_godzina_nazwa karty.csv**, np.: **2009-11-20_20.46.23_users.csv**. Wszelkie zmiany ustawień eksportu są zapisywane jedynie po kliknięciu na **Export**.



Rys. 5.41. Okno eksportu danych

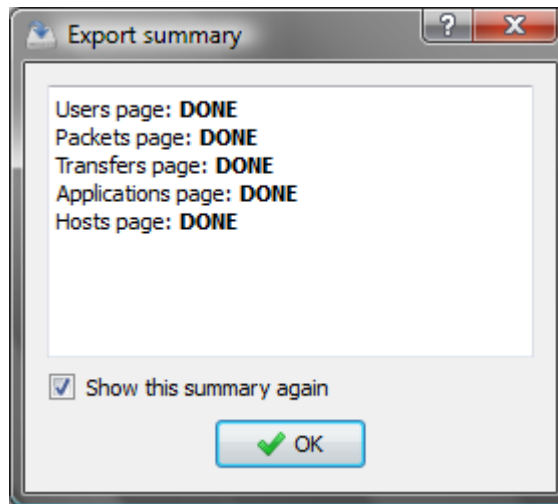
Grupa **Pages to export** pozwala na wybór, z których zakładek dane zostaną eksportowane. W tej grupie przynajmniej jedna pozycja musi być zaznaczona, aby można było dokonać eksportu. Nazwy tu występujące są identyczne jak nazwy zakładek. Przyciski w tej grupie to:

- **Select all** – zaznacza wszystkie pozycje.
- **Select none** – odznacza wszystkie pozycje.

Grupa **CSV file options** umożliwia zmianę podstawowych parametrów zapisywanych plików CSV, aby można było je prawidłowo odczytać w różnych aplikacjach:

- **Fields separation** – separator pól. Domyślnym separatorem pól jest przecinek (ang. *comma*), jednak można zastosować także średnik (ang. *semicolon*).
- **Line termination** – znak końca linii. Możliwe są do wyboru **CRLF**, **CR**, **LF**. CR to ang. *carriage return* (wartość ASCII równa 13, '\r') czyli powrót karetki, a LF to ang. *line feed* (wartość ASCII 10, '\n') czyli nowa linia.
- **Header line** – nagłówek. Pierwszy wiersz będzie nagłówkiem zawierającym nazwy pól rekordów.

Grupa **General** zawiera tylko jedną pozycję – **Show summary**. Zaznaczenie tej opcji oznacza, że po eksporcie danych zostanie wyświetlone okno z podsumowaniem eksportu. Przykład takiego okna przedstawia rys. 5.42. Jak widać w przykładzie eksport wszystkich zakładek się powiódł (ang. *Done* – zrobione). W oknie tym mamy możliwość wyłączenia pokazywania go następnym razem poprzez odznaczenie pola **Show this summary again** (pol. *pokaż to podsumowanie ponownie*).



Rys. 5.42. Okno podsumowania eksportu danych

Grupa **Destination folder** umożliwia zmianę foldera docelowego, gdzie zostaną zapisane pliki:

- **File location** – aktualny folder docelowy.
- **Browse** – umożliwia zmianę foldera docelowego poprzez wybranie już istniejącego w systemie lub stworzenie nowego.
- **Open folder after export** – zaznaczenie tej opcji spowoduje automatyczne otwarcie foldera docelowego po dokonaniu eksportu (po oknie podsumowania, jeśli ma być też wyświetlone).

Funkcje pozostałych przycisków są następujące:

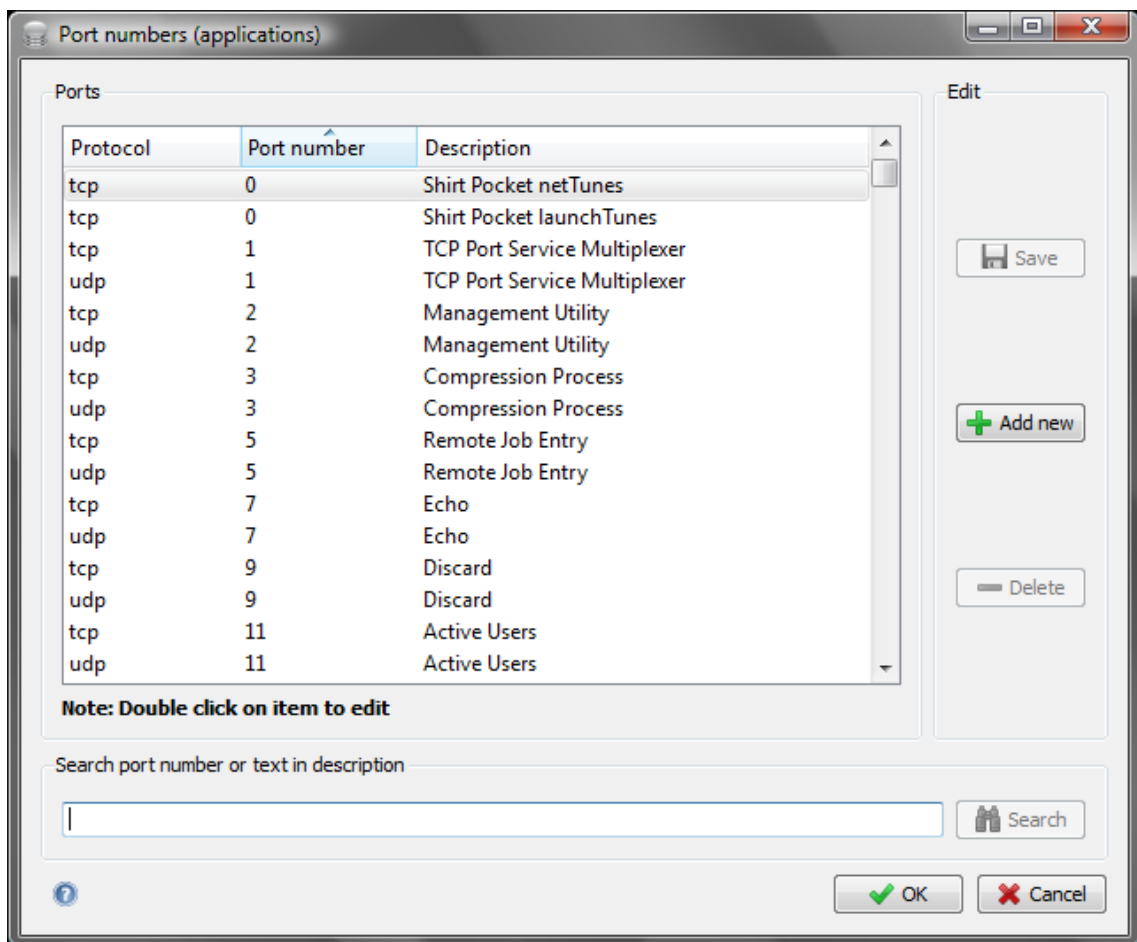
- **Restore Defaults** – przycisk umożliwia przywrócenie domyślnych opcji eksportu danych.
- **Export** – dokonuje eksportu danych oraz zapisuje aktualne ustawienia eksportu.

- **Cancel** – zamyka okno, nie zapisuje żadnych zmian w opcjach.

5.11. Edycja listy aplikacji

Edycji listy portów oraz aplikacji z nimi powiązanych dokonujemy otwierając prosty edytor (rys. 5.43.) wybierając **Port numbers (applications)** w menu **Tools** lub **Ports** na pasku narzędziowym **Tools**.

Aplikacje używane przez użytkowników są rozpoznawane na podstawie portów, na jakich one działają. Zawsze aktualna lista portów oraz powiązanych z nimi aplikacji dostępna jest pod adresem: <http://www.iana.org/assignments/port-numbers>.



Rys. 5.43. Okno edycji i wyszukiwania portów (aplikacji)

Aby edytować daną pozycję należy na nią podwójnie kliknąć lub po zaznaczeniu nacisnąć **F2**.

Znaczenie poszczególnych kolumn jest następujące:


- **Protocol** – typ protokołu: **tcp** lub **udp**.
- **Port number** – numer portu, na jakim działa aplikacja (numer musi być z zakresu **0** do **65535**).
- **Description** – nazwa aplikacji/usługi działającej na danym porcie.

Przyciski w grupie **Edit**:

- **Save** – zapisuje zmiany w pliku (przycisk jest aktywny, gdy dokonano jakichś zmian).
- **Add new** – dodaje nową pozycję do listy identyczną jak aktualnie zaznaczona.
- **Delete** – usuwa aktualnie zaznaczoną pozycję.

Przycisk **Search** jest aktywny, gdy wpisujemy jakiś znak w polu obok. Identyczne działanie posiada naciśnięcie klawisza **Enter** na klawiaturze.

Wyszukiwanie, w zależności od tego, co wpisujemy, odnajduje wpisany tekst w kolumnie **Description** lub numer portu w kolumnie **Port number** (jeśli wpisana została prawidłowa liczba z zakresu **0** do **65535**). Wyszukiwanie rozpoczyna się od aktualnie zaznaczonej pozycji. Każde kolejne naciśnięcie na **Search** szuka kolejnego wystąpienia podanego tekstu lub podanej liczby. Jeśli jest to ostatnie wystąpienie to wyszukiwanie rozpoczyna się od pierwszej pozycji.

Przycisk  wyświetla listę dostępnych skrótów klawiaturowych przyspieszających oraz ułatwiających edycję listy portów (aplikacji). Dostępne skróty klawiaturowe to:

- **F3 – Search Item** – przenosi użytkownika do pola **Search**.
- **Insert – Add new Item** – odpowiada działaniu przyciskowi **Add New**.
- **Delete – Delete Item** – odpowiada działaniu przyciskowi **Delete**.

6. Podsumowanie

Celem pracy magisterskiej było opracowanie systemu umożliwiającego monitoring stanu łącza internetowego oraz pracy użytkowników w sieci lokalnej małej firmy. W ramach pracy została zaprojektowana oraz utworzona aplikacja, która z powodzeniem realizuje wyznaczony cel.

W rozdziale trzecim, wykorzystując przeznaczony do tego celu język UML, opisany został projekt, przedstawiono założenia oraz sposób ich realizacji poprzez opis struktury klas i komunikacji pomiędzy nimi.

W rozdziale czwartym przedstawiono wykorzystane środowisko oraz narzędzia. Opisano, w jaki sposób utworzono logo i ikonkę aplikacji oraz sposób wykonania wielojęzycznego pakietu instalacyjnego. Omówiono również fragmenty kodu źródłowego najważniejszych funkcji programu.

Rozdział piąty to opis interfejsu użytkownika programu. Pokazano, w jaki sposób zrealizowano prezentowanie danych o użytkownikach sieci (oraz stanie łącza internetowego) osobie korzystającej z aplikacji.

Aplikacja LANAnalyzer będzie dalej rozwijana poprzez testowanie i usunięcie istniejących błędów oraz dodanie nowych funkcjonalności, takich jak: analiza pakietów w kolejnych warstwach modelu ISO/OSI, a co się z tym wiąże to zwiększenie liczby obsługiwanych protokołów, dodanie nowych sposobów prezentacji kolejnych danych (np. wykresy słupkowe czy kołowe), całkowite przetłumaczenie aplikacji na język polski, rozwijanie pliku pomocy (podręcznika użytkownika).

Literatura

Literatura

- [1] Blanchette Jasmin, Summerfield Mark: *C++ GUI Programming with Qt 4*, Prentice Hall, 2006
- [2] Fowler Martin: *UML w kropelce, wersja 2.0*, LTP Oficyna Wydawnicza, 2005
- [3] Ganczarski Janusz, Owczarek Mariusz: *C++. Wykorzystaj potęgę aplikacji graficznych*, Helion, 2008
- [4] Jaskiewicz Andrzej: *Inżynieria oprogramowania*, Helion, 1997
- [5] Miles Russ, Hamilton Kim: *UML 2.0. Wprowadzenie*, Helion, 2007
- [6] Siyan Karanjit S., Tim Parker: *TCP/IP. Księga eksperta. Wydanie II*, Helion, 2002
- [7] Tanenbaum Andrew S.: *Sieci komputerowe*, Helion, 2004

Materiały dostępne w Internecie

- [8] Frączek Beata: *PSK - projektowanie systemów komputerowych*
<http://brasil.cel.agh.edu.pl/~09sbfraczek/home,1,1.html> – maj 2010
- [9] Gábor Deák Jahn: *WiX tutorial*
<http://www.tramontana.co.hu/wix/> – czerwiec 2010
- [10] Małecki Michał: *C++ bez cholesterolu*
<http://www.intercon.pl/~sektor/cbx/> – maj 2010
- [11] Opis narzędzi i biblioteki Qt
<http://qt.nokia.com/> – czerwiec 2010
- [12] Portal o Qt
<http://www.qtcentre.org/> – czerwiec 2010
- [13] Opis narzędzia WinPcap
<http://www.winpcap.org/> – czerwiec 2010
- [14] Opis narzędzia WiX
<http://www.wixwiki.com/> – czerwiec 2010

Spis rysunków

Rys. 2.1. Zrzut ekranu aplikacji Wireshark	7
Rys. 2.2. Zrzut ekranu aplikacji CommView.....	8
Rys. 2.3. Zrzut ekranu raportu odwiedzanych stron WWW z aplikacji workAgent.....	9
Rys. 3.1. Diagram przypadków użycia aplikacji.....	13
Rys. 3.2. Diagram przypadków użycia z pakietu <i>Użytkownicy</i>	21
Rys. 3.3. Diagram przypadków użycia z pakietu <i>Sieć</i>	29
Rys. 3.4. Diagram klas	33
Rys. 3.5. Diagram przebiegu przypadku użycia „Wybierz urządzenie”	37
Rys. 3.6. Diagram przebiegu przypadku użycia „Uruchom przechwytywanie z opcjami”	38
Rys. 3.7. Diagram przebiegu przypadku użycia „Przechwytuj”	40
Rys. 3.8. Diagram przebiegu przypadku użycia „Eksportuj dane”	42
Rys. 4.1. Zrzut ekranu aplikacji Qt Linguist w trakcie tłumaczenia.....	48
Rys. 4.2. Schemat blokowy algorytmu analizy pakietów	60
Rys. 5.1. Główne okno aplikacji po uruchomieniu	65
Rys. 5.2. Pasek menu	66
Rys. 5.3. Menu <i>File</i>	67
Rys. 5.4. Menu <i>Capture</i>	68
Rys. 5.5. Menu <i>Network</i>	68
Rys. 5.6. Menu <i>Tools</i>	69
Rys. 5.7. Menu <i>Options</i>	70
Rys. 5.8. Podmenu <i>Toolbars</i>	71
Rys. 5.9. Podmenu <i>Style</i>	71
Rys. 5.10. Podmenu <i>Size</i>	72
Rys. 5.11. Menu <i>Help</i>	72
Rys. 5.12. Paski narzędziowe	73
Rys. 5.13. Pasek stanu.....	73
Rys. 5.14. Zakładki	74
Rys. 5.15. Zakładka <i>Users</i>	75
Rys. 5.16. Zakładka <i>Packets</i>	75
Rys. 5.17. Zakładka <i>Transfers</i>	77
Rys. 5.18. Okno dialogowe z prośbą o podanie liczby użytkowników	78
Rys. 5.19. Lista użytkowników wysyłających najwięcej danych.....	78
Rys. 5.20. Lista użytkowników odbierających najwięcej danych	79
Rys. 5.21. Okno wykresu przepływności danych użytkowników	80
Rys. 5.22. Zapisywanie wykresu	82
Rys. 5.23. Zakładka <i>Applications</i>	83
Rys. 5.24. Menu kontekstowe zakładki <i>Applications</i>	84

Rys. 5.25. Okno dialogowe z prośbą o podanie liczby aplikacji	84
Rys. 5.26. Lista aplikacji użytkownika wysyłających najwięcej danych.....	85
Rys. 5.27. Lista aplikacji użytkownika odbierających najwięcej danych	86
Rys. 5.28. Okno dialogowe z prośbą o podanie liczby użytkowników	86
Rys. 5.29. Lista użytkowników, którzy najwięcej wysłali danych przez wybraną aplikację	87
Rys. 5.30. Lista użytkowników, którzy najwięcej odebrali danych przez wybraną aplikację.....	88
Rys. 5.31. Zakładka <i>Hosts</i>	88
Rys. 5.32. Menu kontekstowe zakładki <i>Hosts</i>	89
Rys. 5.33. Okno dialogowe z prośbą o podanie liczby hostów	90
Rys. 5.34. Lista hostów użytkownika, do których wysłał najwięcej danych	91
Rys. 5.35. Lista hostów użytkownika, od których odebrał najwięcej danych.....	92
Rys. 5.36. Okno wyboru urządzenia sieciowego	93
Rys. 5.37. Okno wyboru urządzenia sieciowego wraz z informacjami o nim	93
Rys. 5.38. Okno wyboru filtra	96
Rys. 5.39. Okno edytora.....	97
Rys. 5.40. Okno dialogowe startu przechwytywania	98
Rys. 5.41. Okno eksportu danych	101
Rys. 5.42. Okno podsumowania eksportu danych	102
Rys. 5.43. Okno edycji i wyszukiwania portów (aplikacji)	103

Dodatek A. Materiały załączone do pracy

Do pracy została dołączona płyta CD, na której umieszczono utworzoną aplikację oraz inne przydatne pliki:

- *LANAnalyzer.msi* – pakiet instalacyjny aplikacji.
- *LANAnalyzer.zip* – skompresowane archiwum z kompletną aplikacją (instalacja nie jest wymagana).
- *help.pdf* – plik pomocy (dostępny również poprzez menu Help aplikacji).
- *WinPcap_4_1_1.exe* – plik instalacyjny sterownika WinPcap, wymaganego do prawidłowej pracy aplikacji.
- Folder *Źródła* – zawiera kod źródłowy aplikacji, wykorzystane ikony, plik źródłowy pliku pomocy, kod źródłowy oraz skrypty do stworzenia pakietu instalacyjnego.

Dodatek B. Podręcznik użytkownika aplikacji

Na płycie CD został dołączony plik pomocy aplikacji LANAnalyze – **help.pdf**. W programie jest on dostępny poprzez menu Help. LANAnalyze otwiera go za pomocą domyślnego, zainstalowanego w systemie, czytnika plików typu pdf (ang. *Portable Document Format* – przenośny format dokumentu). Plik pomocy został utworzony w formie podręcznika użytkownika, po którym można się poruszać za pomocą zakładek, domyślnie wyświetlanych przez przeglądarkę. Ich liczba oraz nazwy odpowiadają pozycjom spisu treści.

Mariusz Helfajer

Opole, dnia 1.07.2010 r.

Wydział Elektrotechniki, Automatyki i Informatyki

Kierunek: Informatyka

OŚWIADCZENIE

Oświadczam, że złożona praca magisterska pt. „Analiza aktywności pracy użytkowników w sieciach lokalnych” została napisana przeze mnie samodzielnie, pod kierunkiem promotora dra inż. Krzysztofa Zatwarnickiego, a jej wersja drukowana (forma papierowa) jest w swej treści zgodna z przedłożoną wersją elektroniczną pracy.

Równocześnie informuję, że praca nie narusza praw autorskich osób trzecich w rozumieniu ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (tekst jednolity: Dz. U. Nr 80 z 2000 r., poz. 904 z późniejszymi zm.).

Praca nie zawiera informacji i danych uzyskanych w sposób nielegalny i nie była wcześniej przedmiotem innych procedur związanych z uzyskaniem dyplomów lub tytułów zawodowych wyższych uczelni.