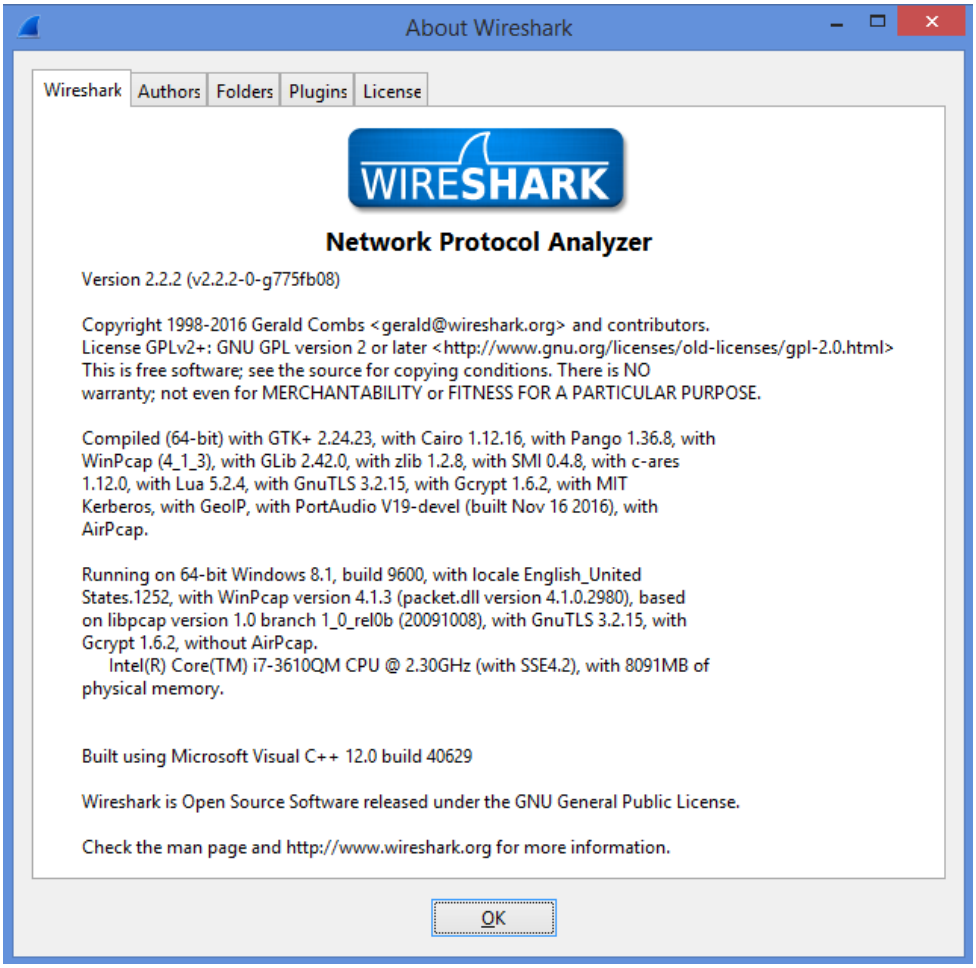
Instrukcja 8 - Filtry w analizatorach protokołów

8.1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z wybranymi funkcjami zaawansowanych analizatorów protokołów z możliwością zastosowania filtrów.

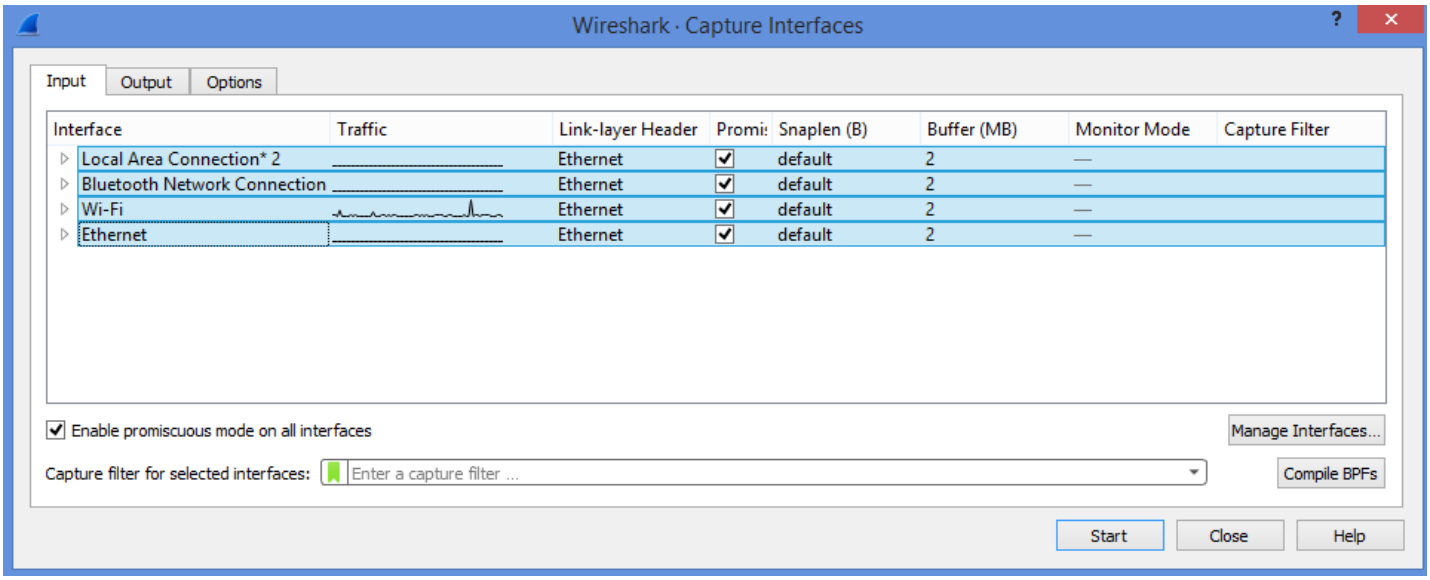
8.2 Analizator Wireshark

Analizator Wireshark jest programowym analizatorem protokołów. Dzięki olbrzymim i stale rosnącym możliwością i dostępności na licencji GNU jest najpopularniejszym tego typu programem na świecie. Jest on kontynuacją projektu Ethereal.



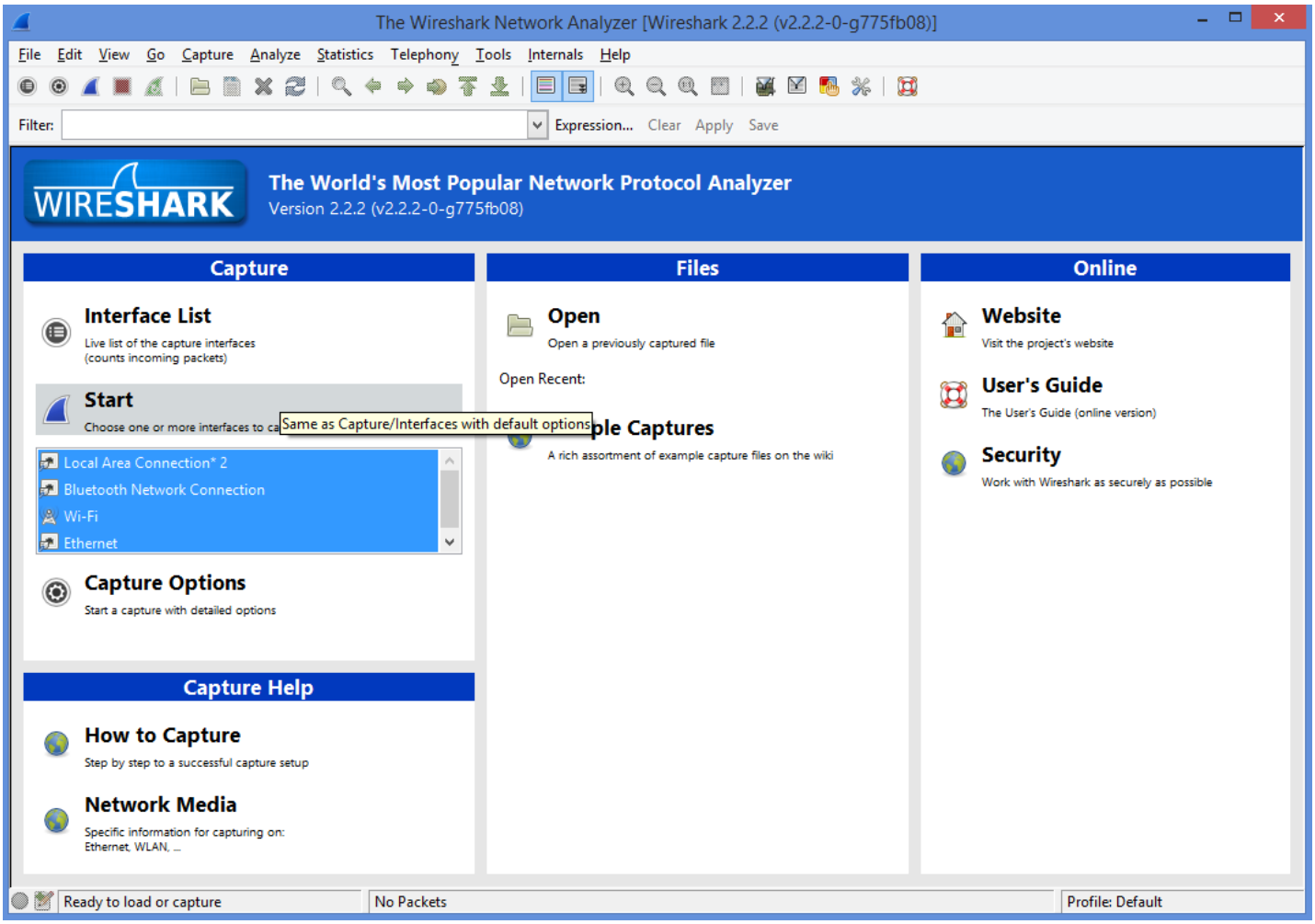
Rysunek 8.1: Program Wireshark

Po włączeniu programu **Wireshark** należy wybrać opcję Go → Options, zaznaczyć interesujące interfejsy (w ramach ćwiczenia badane będą wszystkie interfejsy) i wybrać przycisk **Start**.



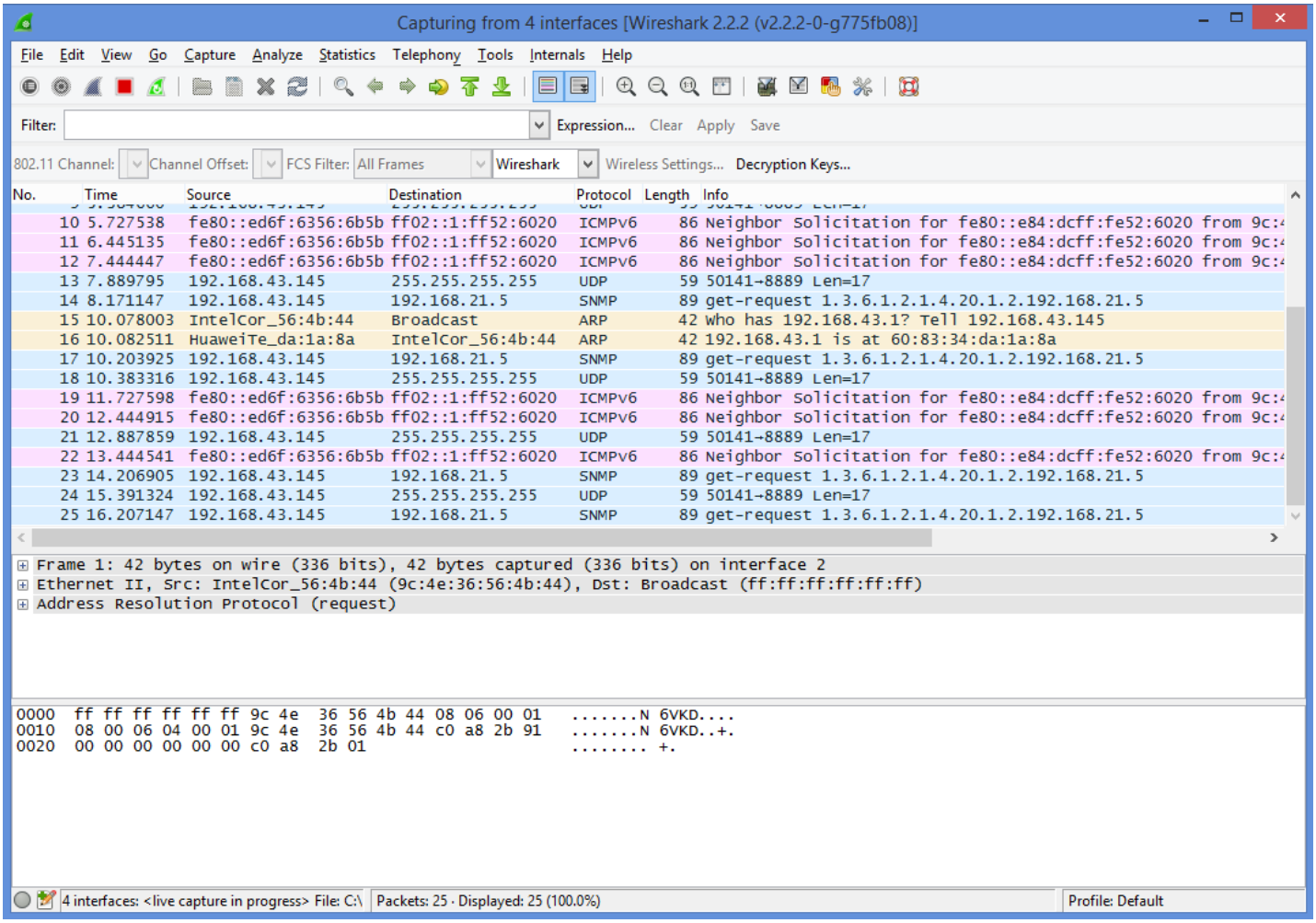
Rysunek 8.2: Rozpoczęcie działania programu

Możliwe jest także uruchomienie programu w wersji Legacy. W takim wypadku wybór interfejsu odbywa się bezpośrednio z okna głównego programu:



Rysunek 8.3: Wireshark Legacy

Okno programu składa się z kilku elementów ulokowanych pod sobą. Są to: menu, pasek narzędzi, pole filtru wyświetlania, lista przechwyconych ramek, szczegóły wskazanej ramki, treść bajtów ramki, pasek stanu. Widoczność wymienionych elementów można włączać i wyłączać.



Rysunek 8.4: Okno programu Wireshark w trybie przechwytywania ramek

Bardzo ważnym elementem pracy z analizatorem protokołu jest korzystanie z **filtrów** przechwytywania i wyświetlania. W programie Wireshark filtry tworzy się w oparciu o poniższe zasady:

[pole operator_relacji wartość] operator_logiczny [pole operator_relacji wartość] ...

Przy czym wartość w nawiasie [] może odpowiadać także **protokołowi**. W takim wypadku w nawiasie [] umieszczana jest tylko nazwa protokołu.

Operatory relacji

- == - równe,
- != - różne,
- > - większe,
- < - mniejsze,
- >= - większe lub równe,
- <= - mniejsze lub równe.

Operatory logiczne

- and, && - logiczne AND
- or, || - logiczne OR
- not, ! - logiczne NOT

Nazwy protokołów
arp, dns, tcp, udp, ip, ipv6, irc, idp, ipx, http, pop, smtp, ftp, gnutella, kerberos, l2tp, netlogon, smb, ...

Przykładowe pola - Ethernet

- eth.addr ==
- eth.dst ==
- eth.len ==
- eth.src ==
- eth.trailer ==
- eth.type ==

Przykładowe pola - IP

- ip.dst eq www.mit.edu
- ip.src == 192.168.1.1
- ip.addr == 129.111.0.0/16
- ip.fragment ==
- ip.id ==
- ip.len ==
- ip.ttl ==

Przykładowe pola - TCP

- tcp.port == 80
- tcp.dstport ==
- tcp.srcport ==
- tcp.ack == numer potwierdzenia
- tcp.flags == flaga 8-bit
- tcp.flags.reset ack, syn, fin,
- tcp.len == ???
- tcp.window_size ==

Przykładowe pola - UDP

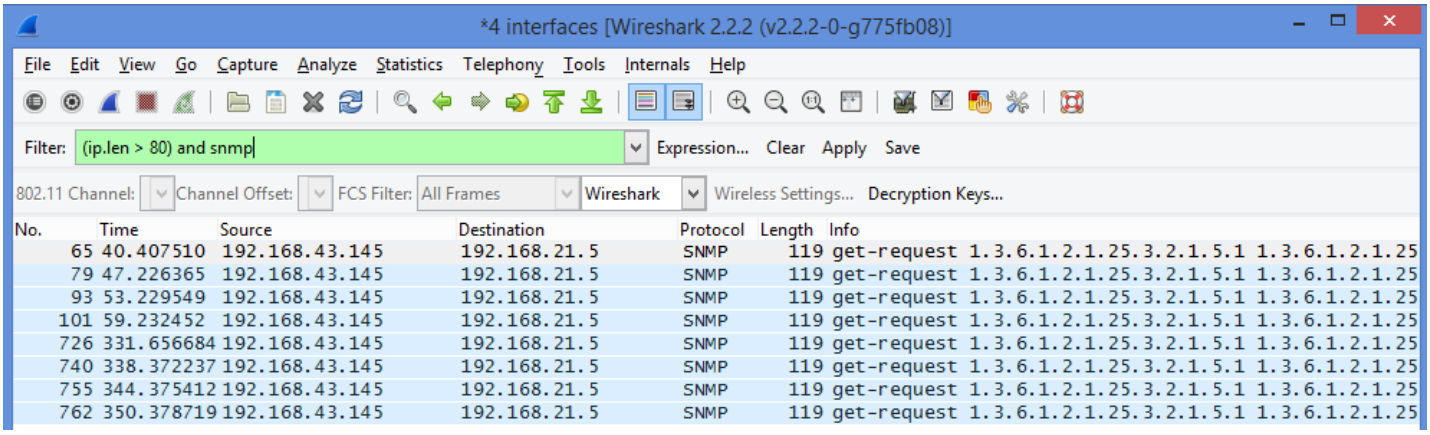
- udp.checksum ==
- udp.checksum_bad ==
- udp.dstport ==
- udp.length ==
- udp.port ==
- udp.srcport ==

Przykładowe pola - HTTP

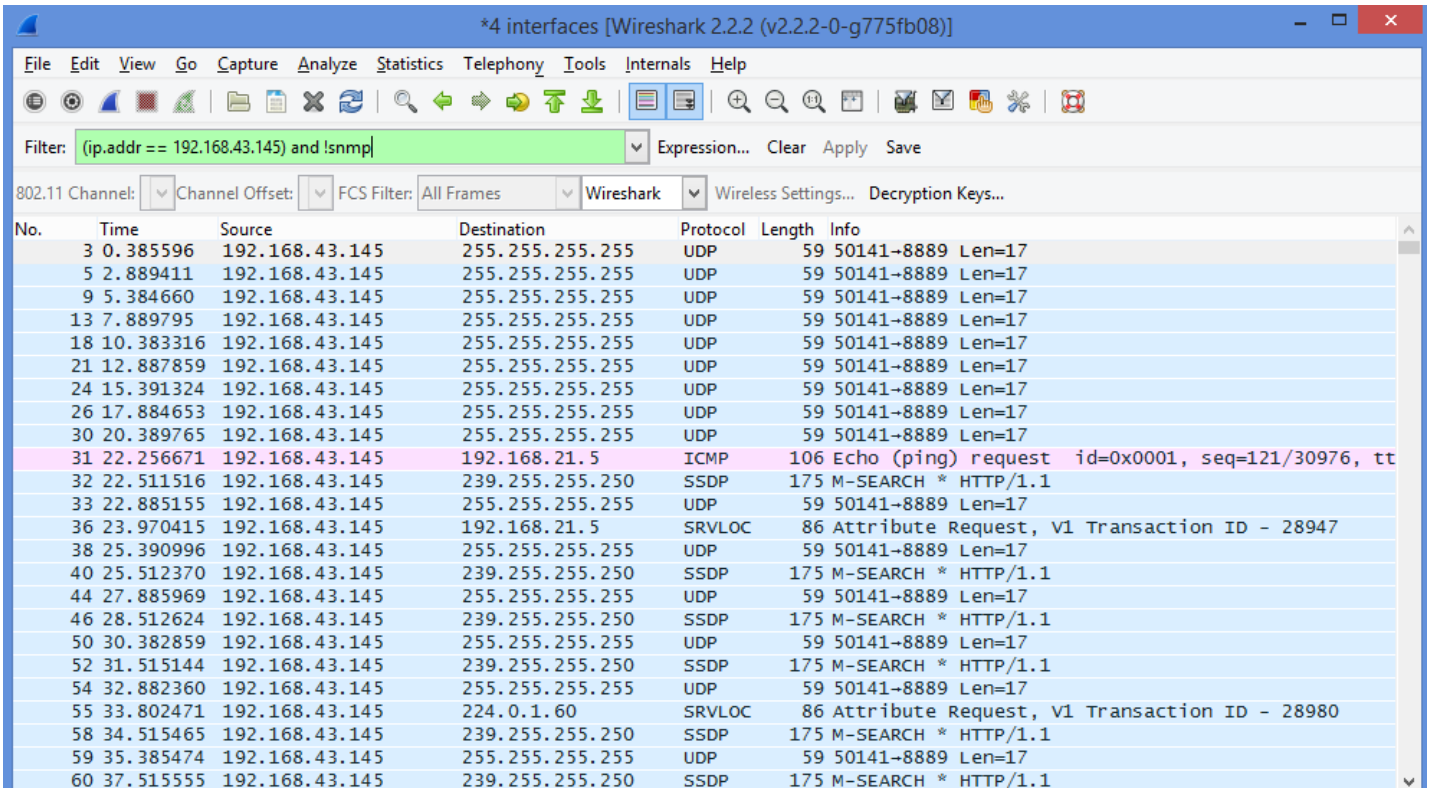
- http.cookie ==
- http.host ==

Przykładowe pola - Echo

- echo.data ==
- echo.request ==
- echo.response ==

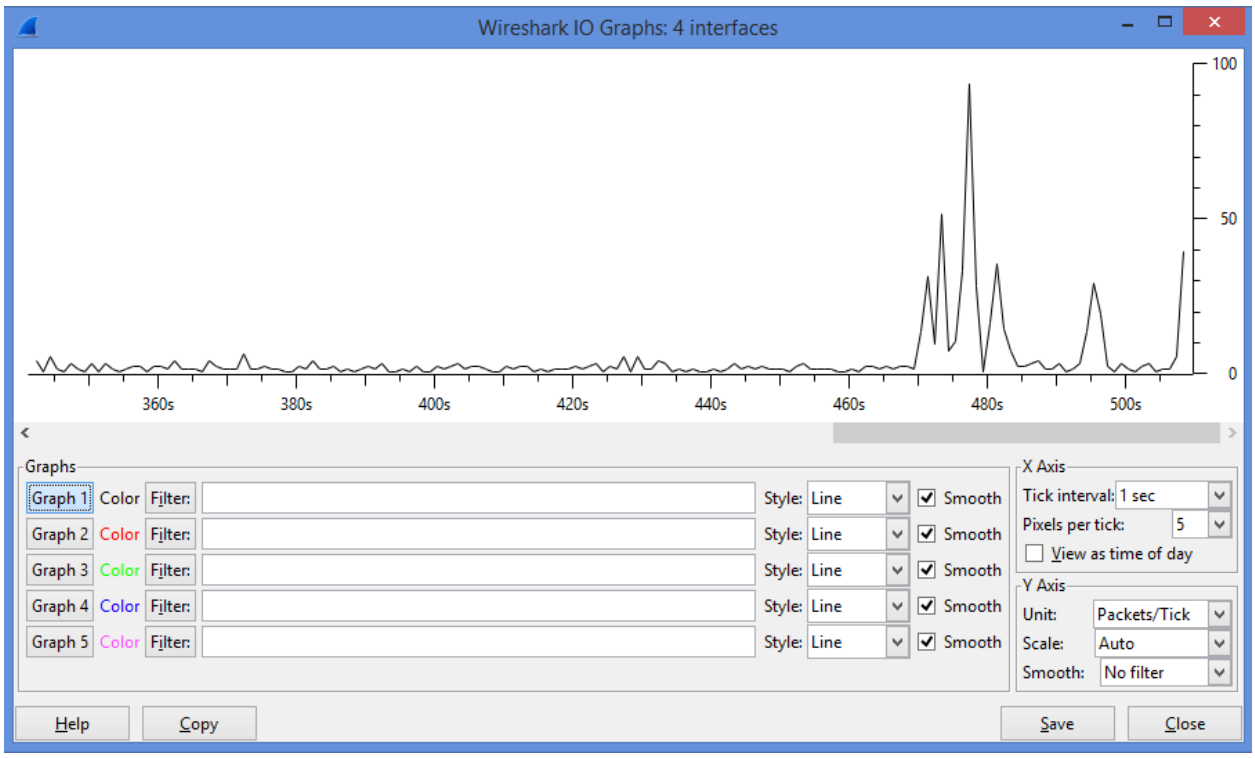


Rysunek 8.5: Przykład filtra: *(ip.len > 80) and snmp*



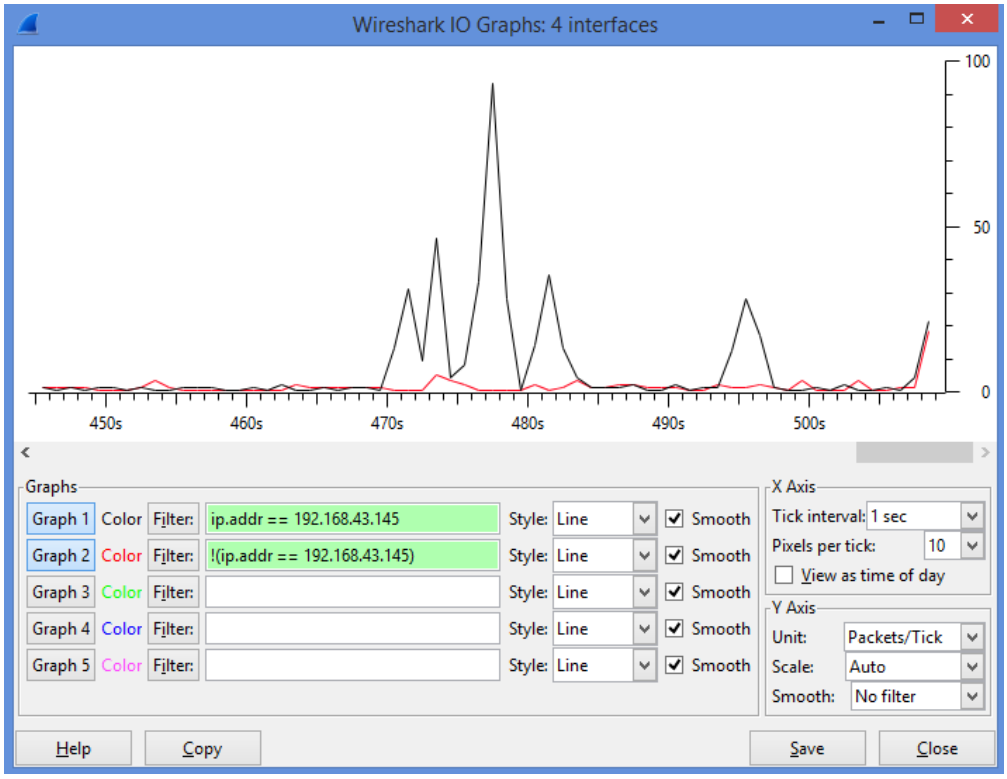
Rysunek 8.6: Przykład filtra: *(ip.addr == 192.168.43.145) and !snmp*

Program Wireshark umożliwia także podgląd statystyk w trybie graficznym (**Statistics** → **IO Graph**).



Rysunek 8.7: Wyświetlenie statystyk graficznych

Na wykresie możliwe jest przedstawienie do 5 wykresów, z których każdy dotyczy innego, dowolnie przypisanego zestawu filtrów:



Rysunek 8.8: Wyświetlenie statystyk graficznych z uwzględnieniem filtrów

8.3 Przebieg ćwiczeń

Przechwycić możliwie największą ilość ramek. W tym celu należy uruchomić przechwytywanie ruchu, a następnie uruchomić różne programy korzystające z zasobów sieciowych i programów diagnostycznych, np. przeglądarkę internetową, klienta poczty e-mail, ftp, ping, nslookup, net, przeszukiwanie zasobów sieciowych explorerem windows.

Wśród przechwyconych pakietów należy korzystając z filtru wyświetlania wybrać te, które:

- kierowane są tylko do naszego komputera,
- wysyłane są z naszego komputera,
- wykorzystują protokół arp,
- wykorzystują protokół icmp,
- wykorzystują protokół ftp,
- wykorzystują protokół pop,
- wykorzystują protokół smtp,
- wykorzystują protokół arp i zostały wysłane z naszego komputera,
- inne dla dowolnie zaprojektowanych filtrów.

1. Wykreślić statystyki ruchu w sieci z podziałem na wybrane protokoły (http, udp, tcp, smb, icmp, arp itp.).

2. Porównać statystyki dla ip.addr ustawionego na adres naszego komputera oraz na wartość różną od adresu.

3. Wykonać **ping** serwera pcz.pl. Odnaleźć wszystkie ramki (protokół ICMP). Ile ramek zostało nadanych?

4. Sprawdzić ile % zostało nadanych ramek **TCP** o długości (frame.len): a) mniejszej niż 100, b) większej lub równej 100 i mniejszej niż 1000, c) większej lub równej niż 1000. Informacje o wyświetlanej liczbie ramek dostępne są w dolnym pasku statusu programu.

5. Jaki protokół wykorzystuje ramki o rozmiarze 42?

6. Wyłączając wszelkie filtry przejrzeć statystyki hierarchiczne ramek (**Statistics → Protocol Hierarchy**). Użyć dowolnej z wyświetlanych statystyk jako filtra (prawy przycisk myszy → Apply as Filter).

7. Podejrzeć zawartość ramek HTTP (dwukrotne kliknięcie). Czy zawartość jest przejrzysta i można z niej odczytać jakieś informacje?

8.4 Sprawozdanie

Studenci pracują i przygotowują sprawozdania w parach. W sprawozdaniu należy przedstawić przebieg przeprowadzonych eksperymentów, ich wyniki oraz wnioski.