Instrukcja programu Wireshark (wersja 1.8.3) w zakresie TCP/IP

I. Na początek

Czym jest analizator sieciowy jakim jest Wireshark?

Analizator sieciowy pozwala na przechwytywanie i analizę danych, które są przesyłane przez sieć. W łatwy sposób możemy "obejrzeć" cały ruch w naszej sieci. Możemy sprawdzić jakie porty wykorzystuje dane oprogramowanie oraz wysyłany/odbierany ruch przez nasz komputer.

Wireshark jest najbardziej popularnym analizatorem sieciowym. To silne narzędzie dostarcza sieciowe i wyższych protokołów informacyjnych dotyczące zrzuconych danych w sieci.

Do roku 2006 Wireshark nosił nazwę Ethereal.

Ważna uwaga:

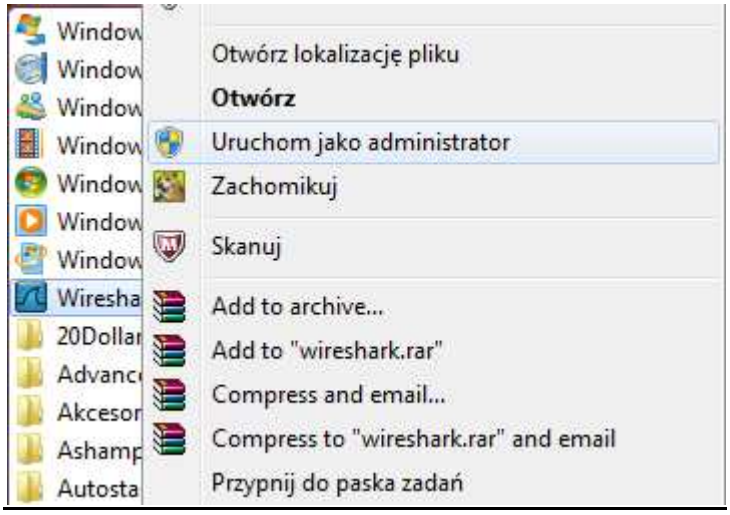
Jeśli szukasz programu do włamywania się, podrzucania wirusów, wykradania haseł czy czegoś w tym stylu - to NIE TO. Wireshark tylko wyświetla w czytelny sposób to co juz i tak dociera do twojego komputera (zwykle oznacza to tylko ruch własny). Nie spowoduje, że nagle zobaczysz hasło do banku swojego szefa, albo kody do odpalenia amerykańskich, czy rosyjskich rakiet balistycznych.

Do czego może służyć Wireshark?

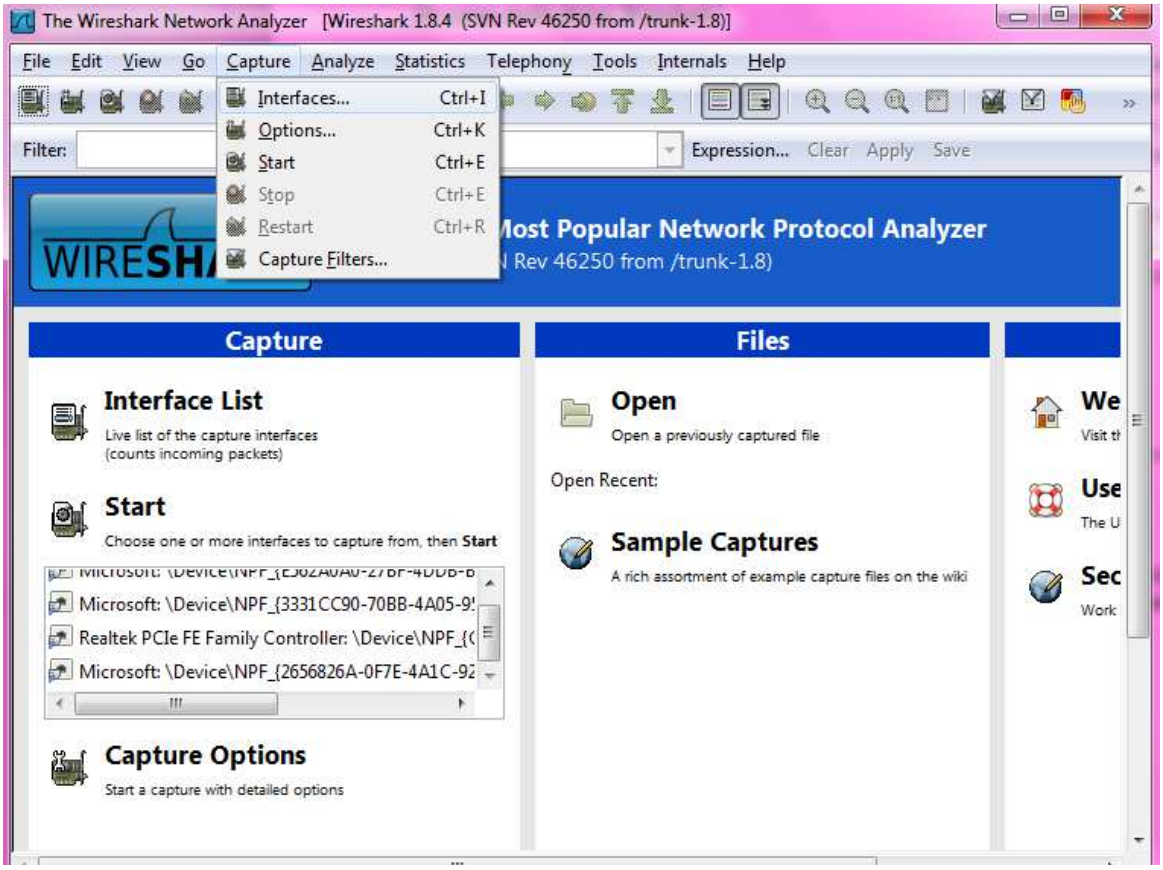
- Głównie do diagnostyki: np. nie można połączyć się z jakimś serwerem - on nie działa, czy mój komputer w ogóle nie próbuje, a może DNS podał zły adres? To oznacza, że raczej trzeba już wiedzieć czego się szuka, a wtedy instrukcja obsługi nie jest potrzebna ;)
- Jest doskonałym narzędziem do nauki - żaden wykład teoretyczny nie zastąpi popatrzenia sobie w praktyce jak te pakiety faktycznie wyglądają.
- No dobrze, można też podsłuchiwać. Jest to możliwe, kiedy komputer "słyszy" nie swoje pakiety. Dzieje się tak np. gdy mamy sieć zbudowaną na HUBach (a nie switchach), bezprzewodową (chociaż wtedy mogą być problemy z driverami i i tak się nie uda) lub nasz komputer robi za router ("udostępnia sieć" innym).

II. Przechwytywanie

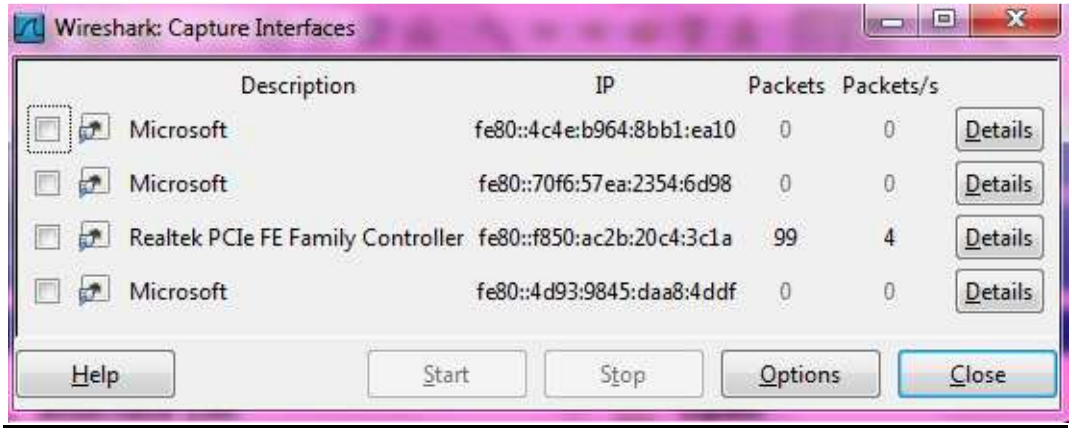
- Uruchom **Wiresharka** z prawami Administratora

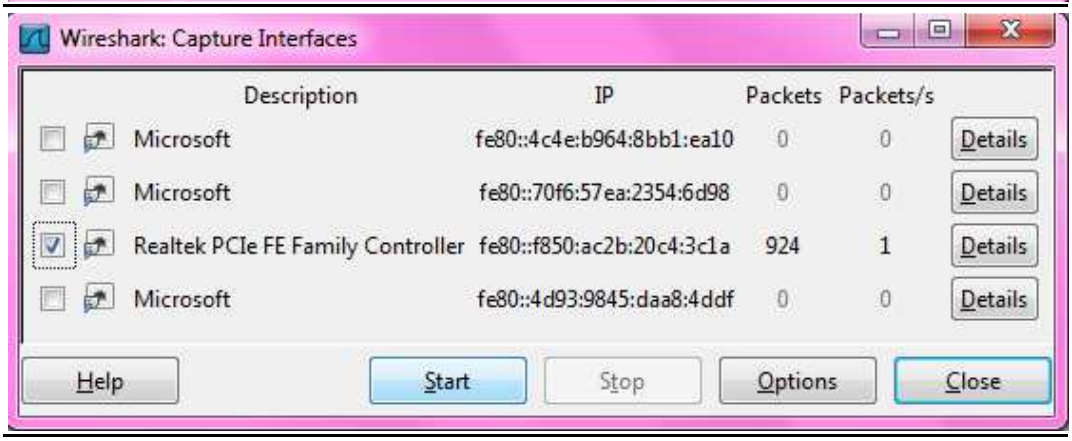


- Następnie wybierz **Capture - Interfaces**

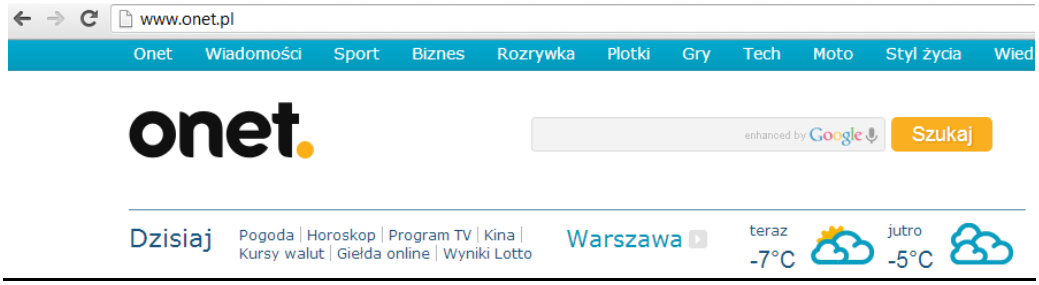


- Wybierz interfejs którym twój komputer łączy się z siecią (zwykle **eth0**) i naciśnij start.

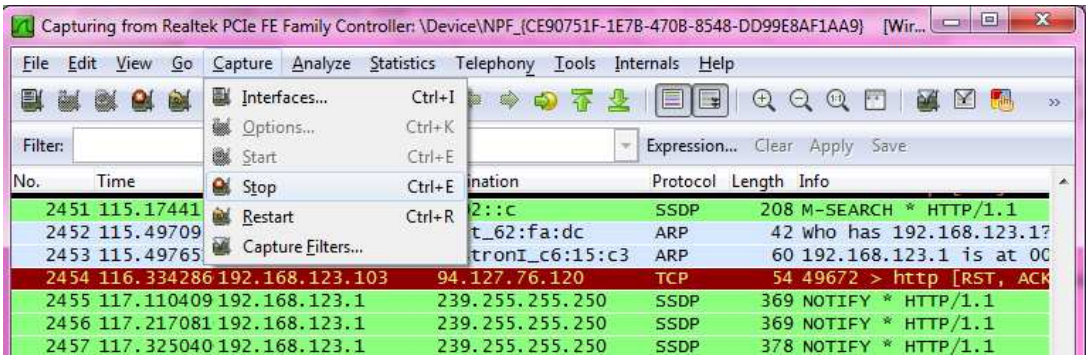




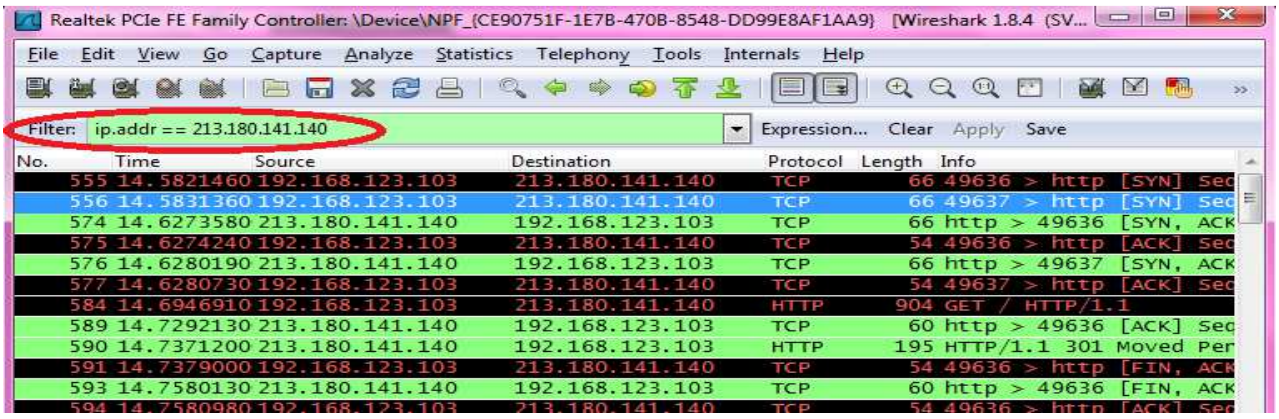
- Przeglądarką połącz się z jakąś stroną internetową (**np. onet.pl**).



- Skończ chwytanie pakietów (**Capture - Stop)**

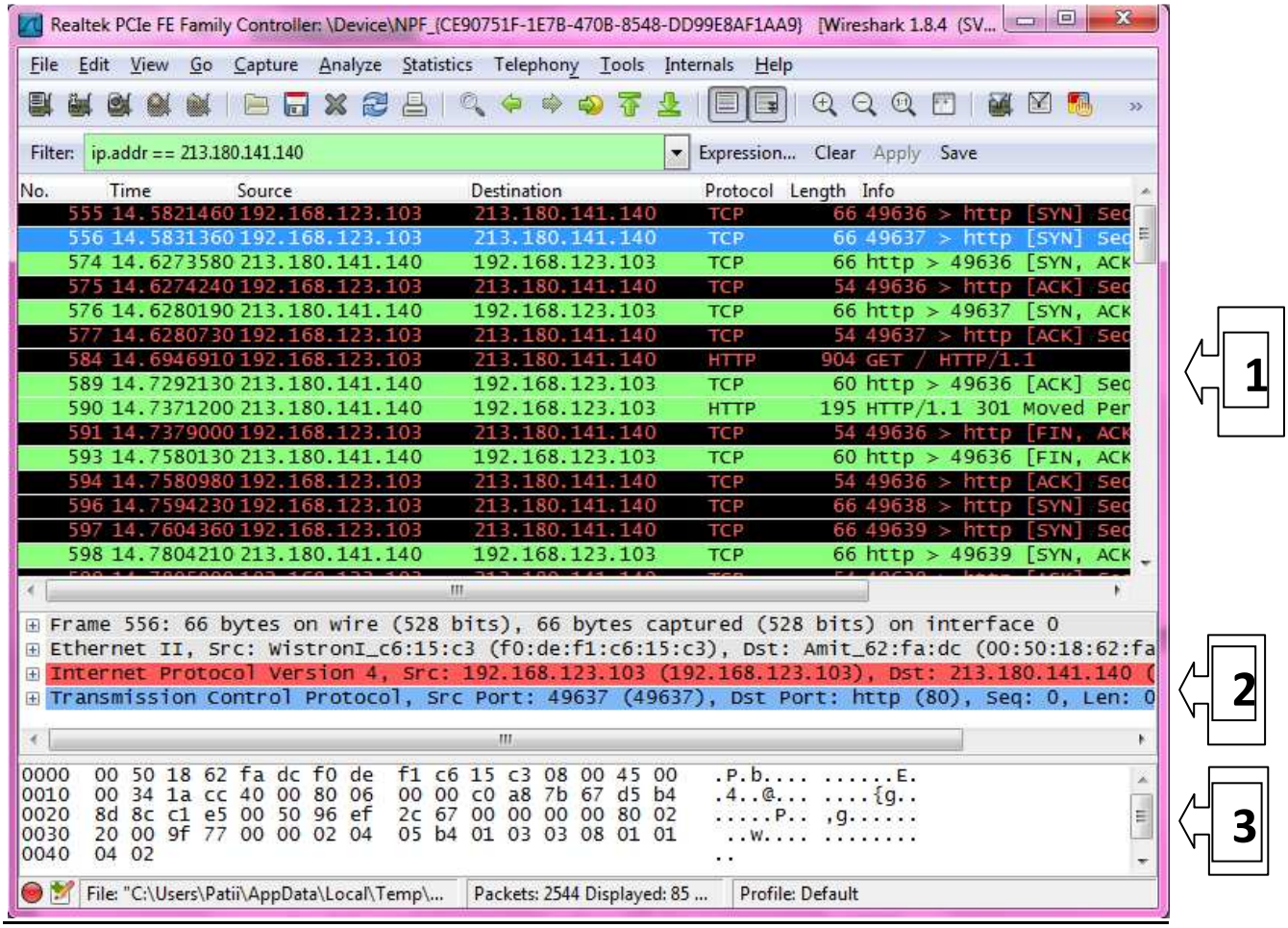


- Przefiltruj listę pakietów by zostawić tylko te opisujące komunikację z hostem www.onet.pl (IP: 213.180.141.140): w pasku **„Filter”** u góry okna programu wpisz: ip.addr == 213.180.141.140.

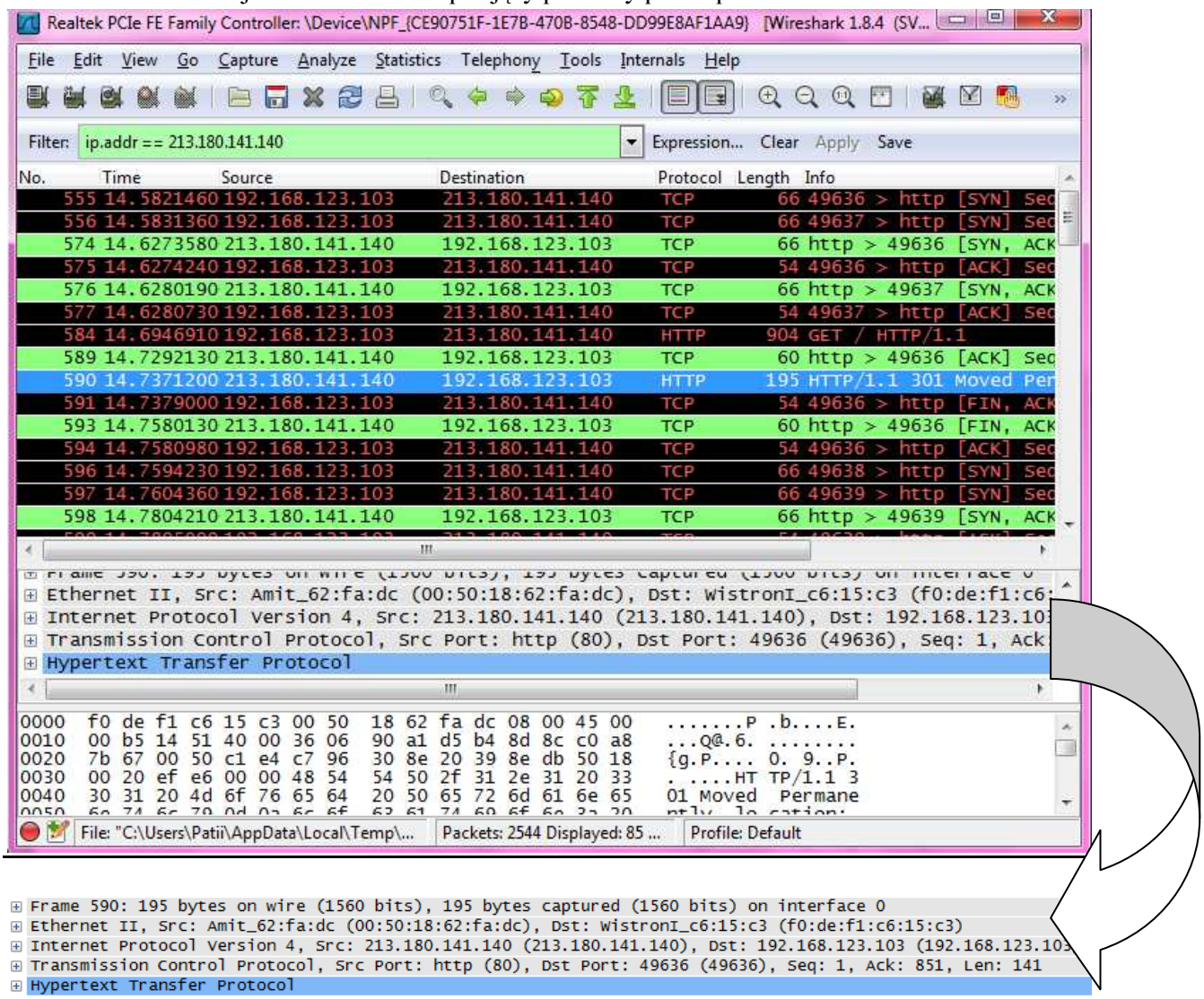


III. Analiza pakietów

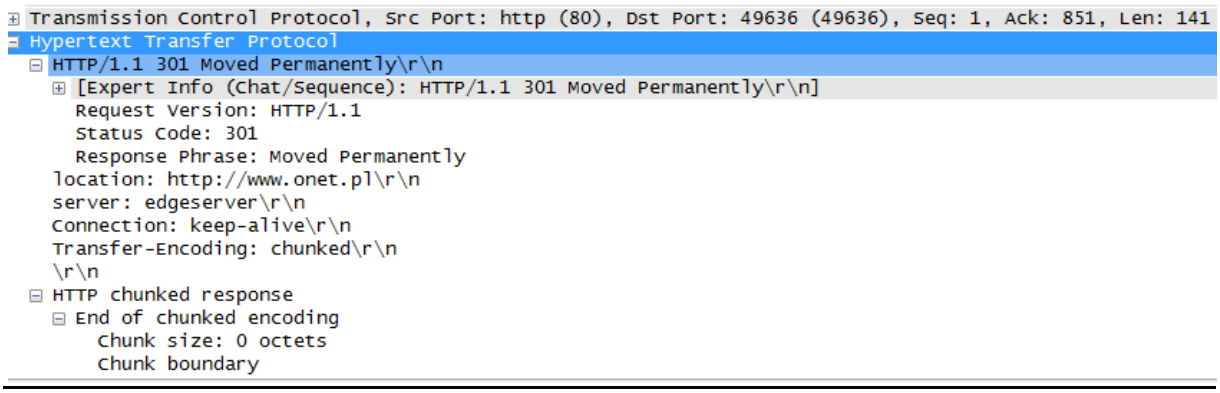
- Okno Wireshark podzielone jest na trzy części:
  - górna część zawiera listę złapanych pakietów (przefiltrowanych przez filtr) - **1**
  - środkowa część to analiza wybranego pakietu - **2**
  - dolna część to zawartość pakietu (binarna i reprezentacja ASCII) - **3**



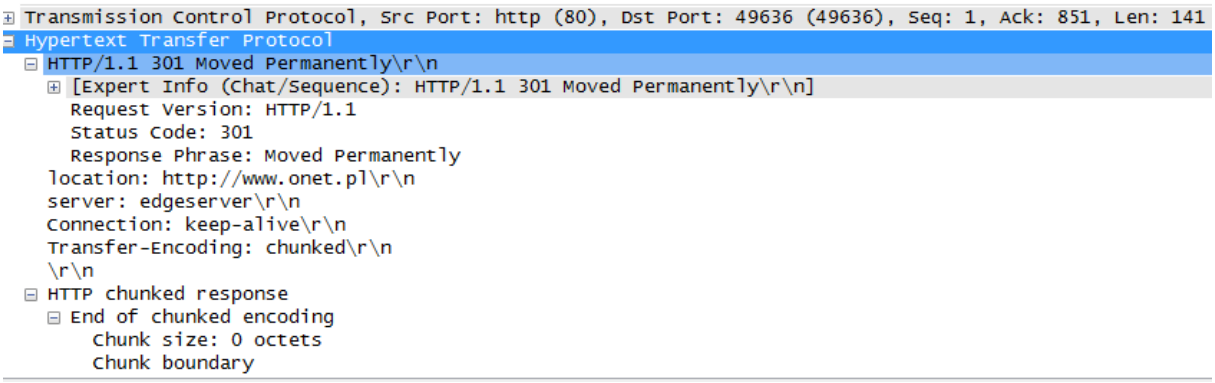
- Zadania:
  - Kliknij w wiersz tabelki opisujący pierwszy pakiet protokołu HTTP.



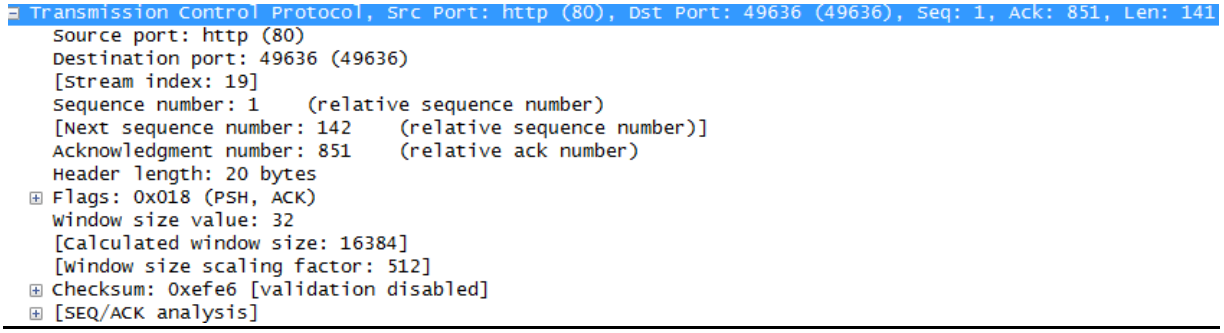
  - Obejrzyj szczegółowe informacje w środkowej części okna.

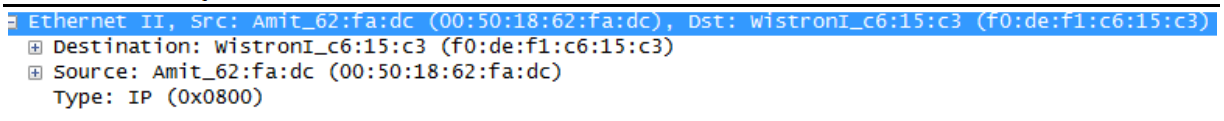


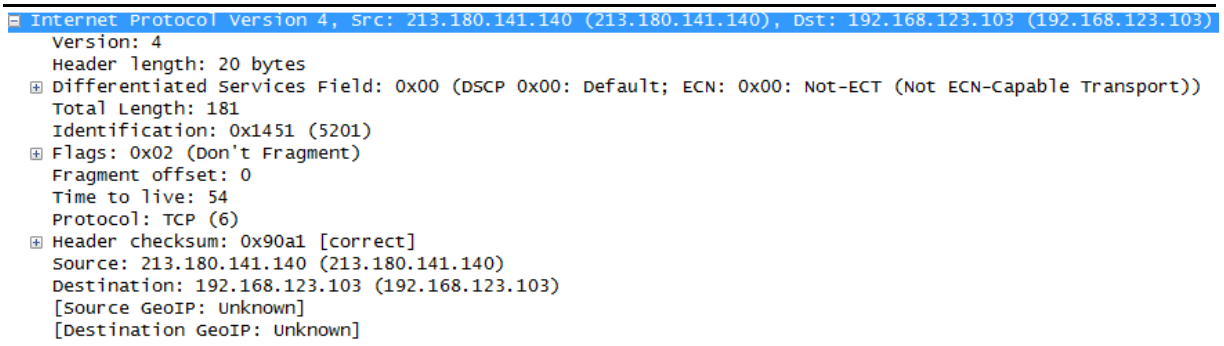
- Kliknij w Hypertext Transfer Protocol w środkowej części okna.



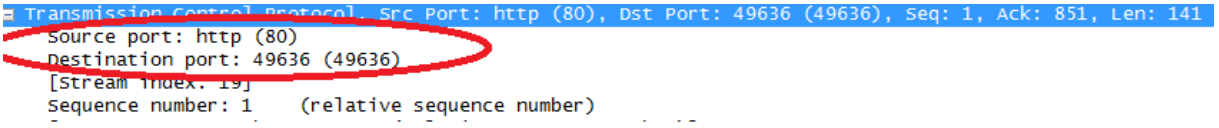
- Obejrzyj które bajty w pakiecie odpowiadają za zawartość HTTP, a które za nagłówki (Ethernet, IP, TCP).







- Jaki jest lokalny numer portu? Jaki jest numer portu serwera?



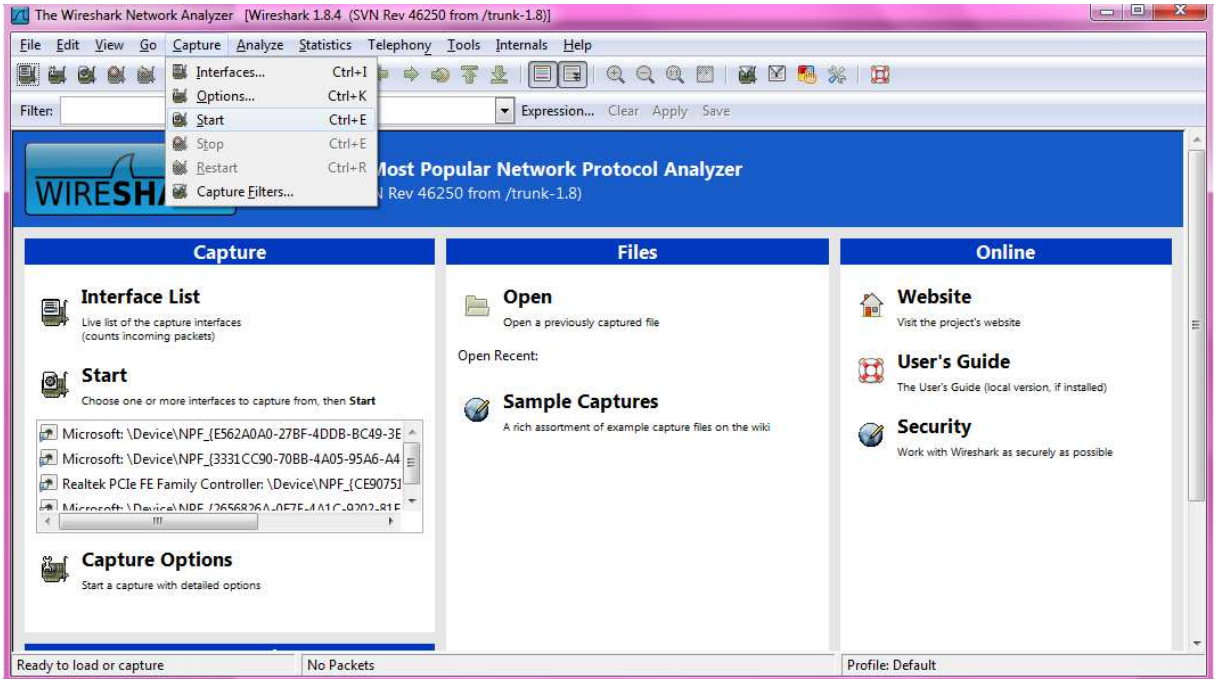
- Znajdź pakiet z odpowiedzią serwera.

```
Frame 590: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface 0
    Interface id: 0
    WTAP_ENCAP: 1
    Arrival Time: Dec  8, 2012 18:16:38.678479000 Środkowoeuropejski czas stand.
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1354986998.678479000 seconds
    [Time delta from previous captured frame: 0.007907000 seconds]
    [Time delta from previous displayed frame: 0.007907000 seconds]
    [Time since reference or first frame: 14.737120000 seconds]
    Frame Number: 590
    Frame Length: 195 bytes (1560 bits)
    Capture Length: 195 bytes (1560 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:tcp:http:data]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80]
```

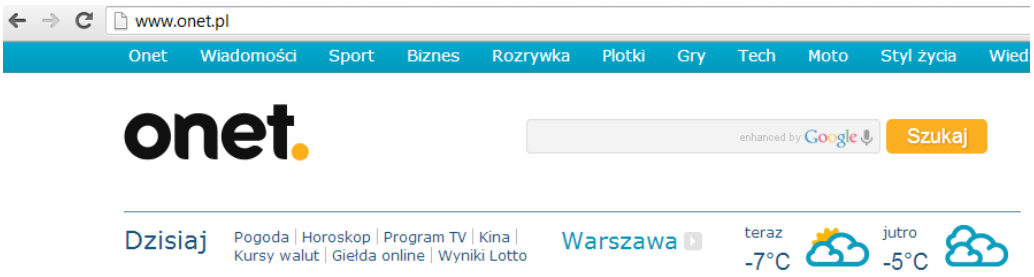
- Z jakiego portu została wysłana odpowiedź? Na jaki port w naszym hoście?
- Jaka jest zawartość odpowiedzi serwera?

IV. Cache'owanie w HTTP

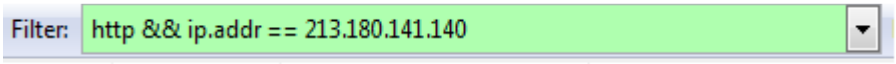
- Tą część trzeba wykonać z włączonym łapaniem pakietów [po wstępnym wybraniu interfejsu, łapanie pakietów można włączać przez Capture – Start]



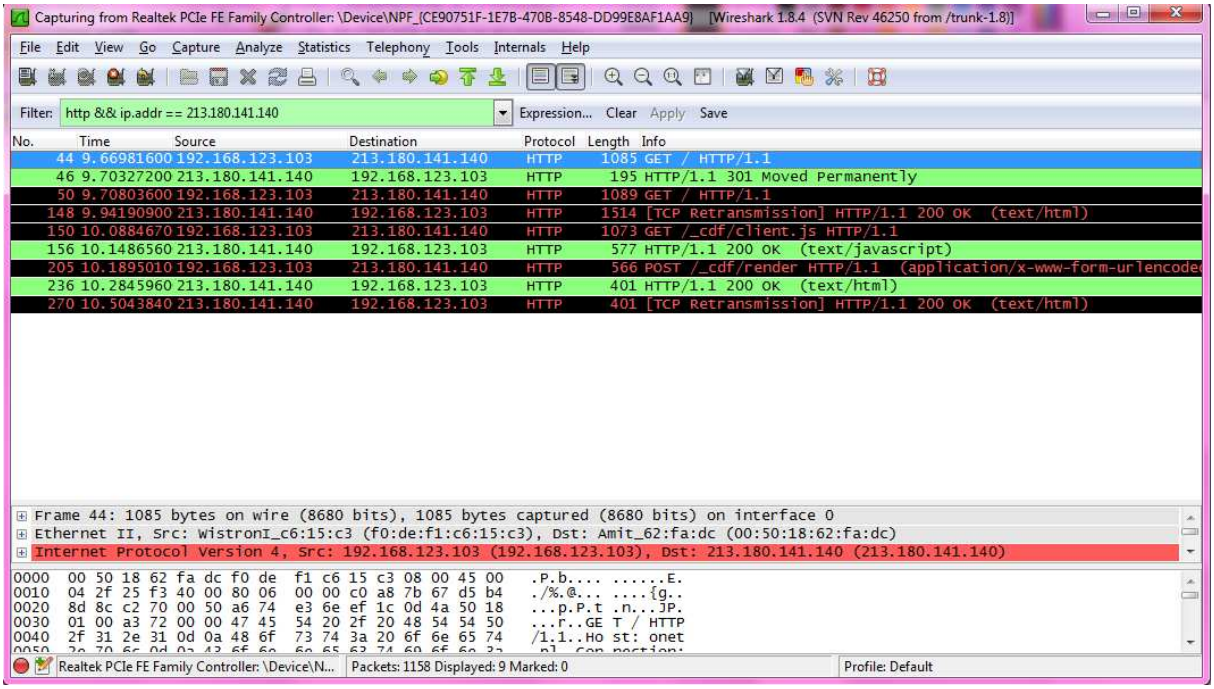
- Połącz się jeszcze raz z Onet.pl (lub inną wybraną stroną)



- Ogranicz pakiety do http (filtr: http && ip.addr == 213.180.141.140)

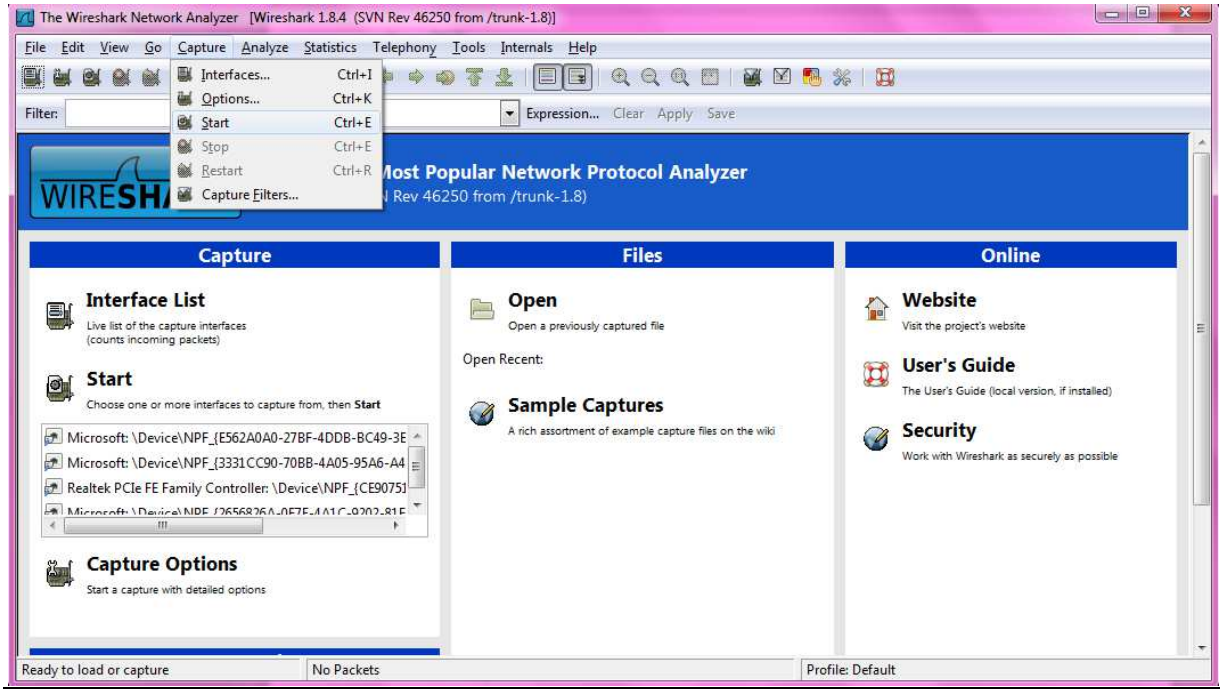


- Co zmieniło się w odpowiedzi serwera? Dlaczego serwer http mógł tak odpowiedzieć?

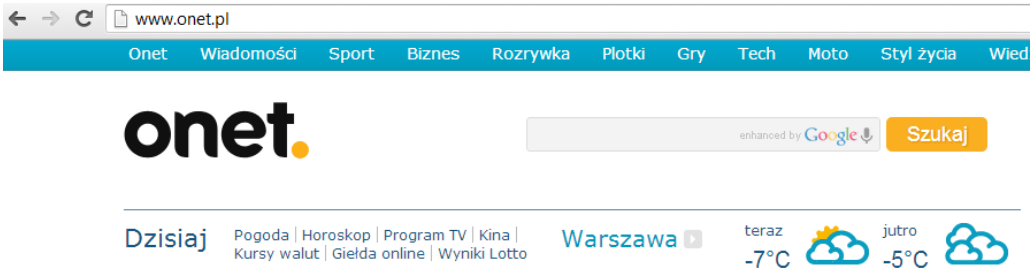


V. TCP

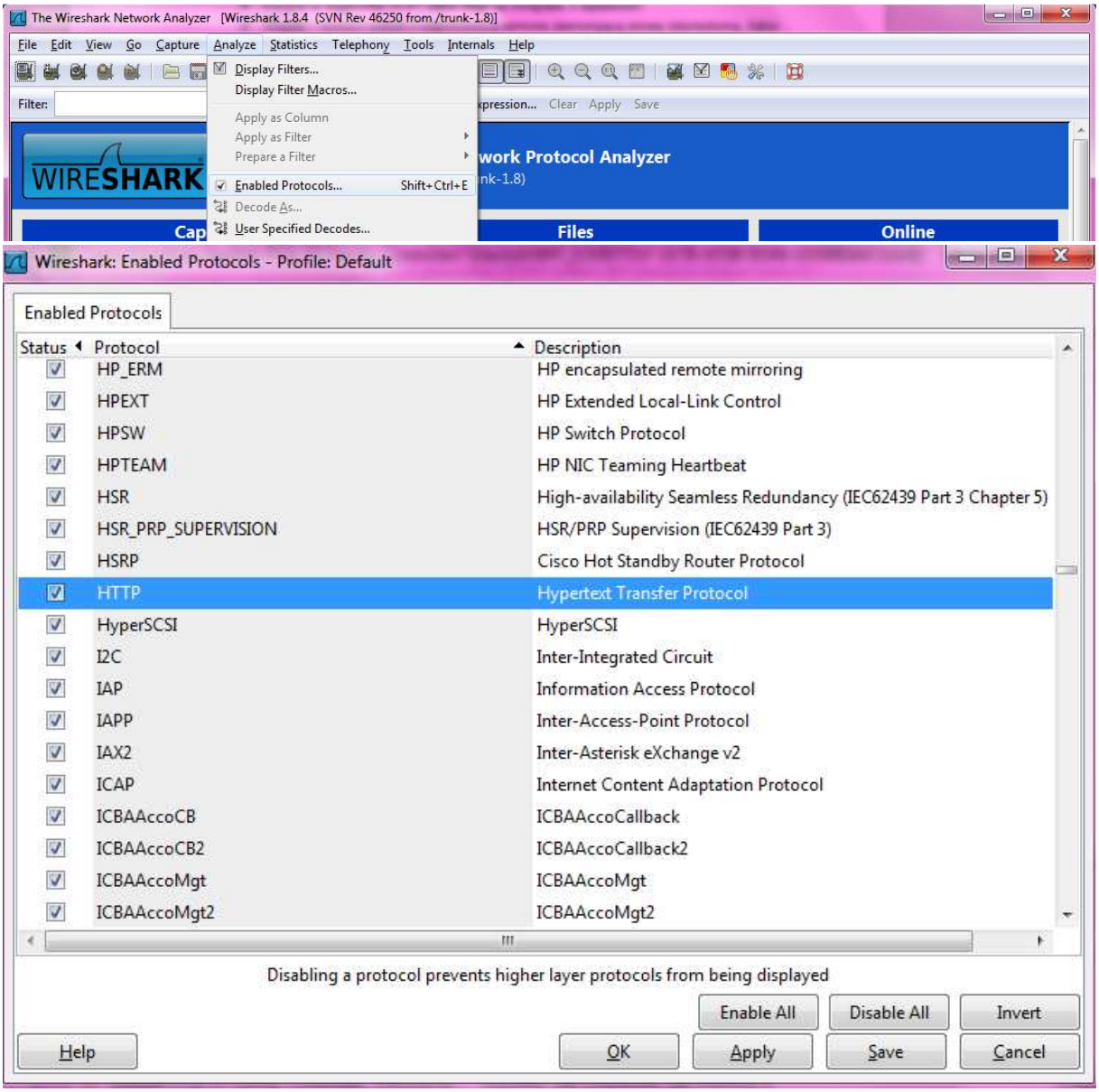
- Nagłówek TCP (tą część wykonujemy z włączonym łapaniem pakietów)



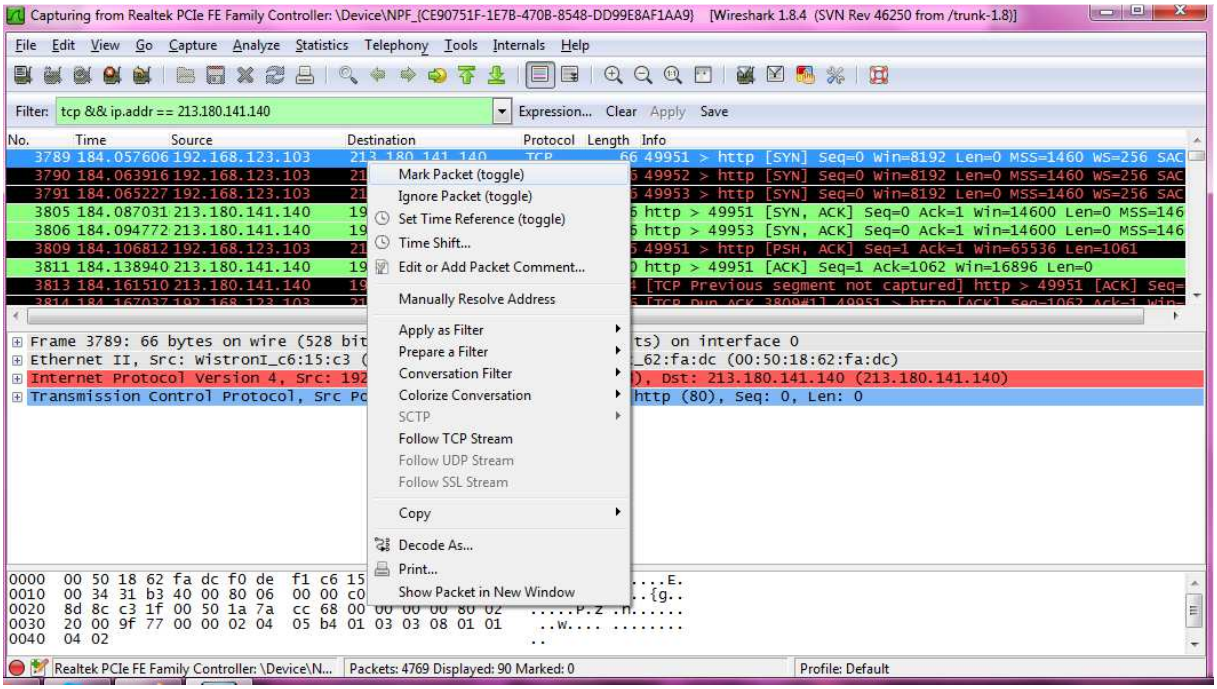
- użyj przeglądarki i połącz się z jakąś stroną (np. onet.pl)



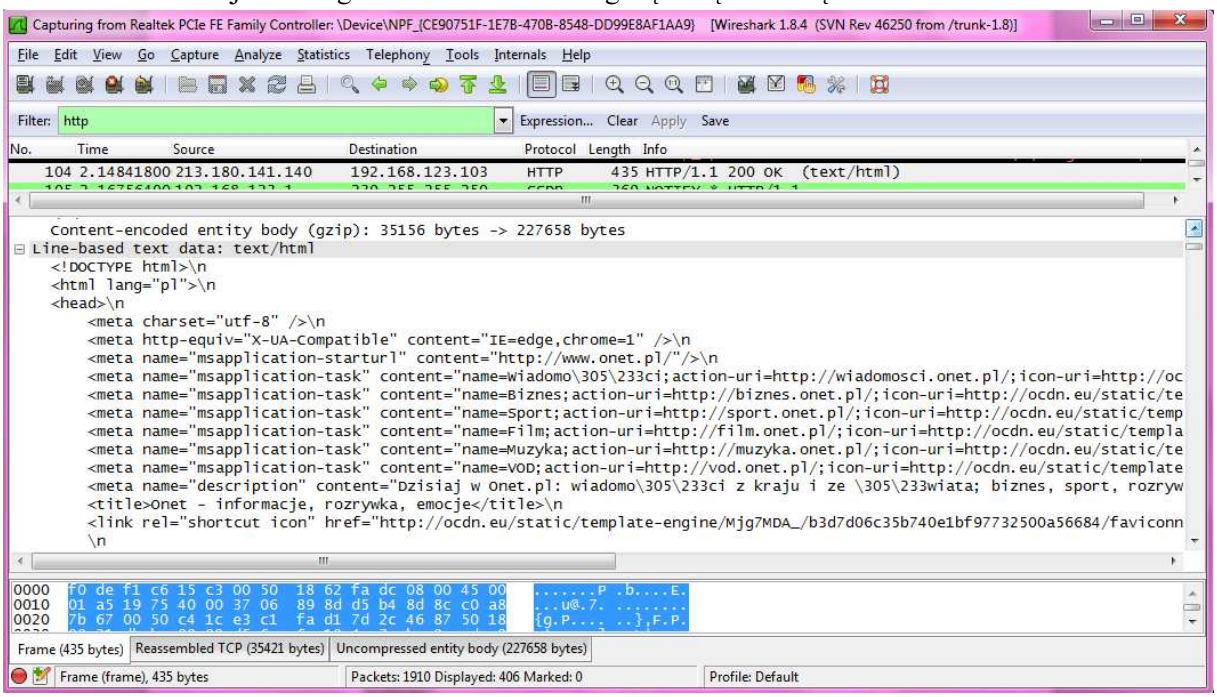
- Wyłącz analizę protokołu HTTP (Analyze - Enabled Protocols). Ustaw filtr na protokół TCP i adres IP serwera www.onet.pl



- Znajdź i zaznacz (Edit - Mark Packet) pakiet w którym przeglądarka przesyła żądanie do serwera.



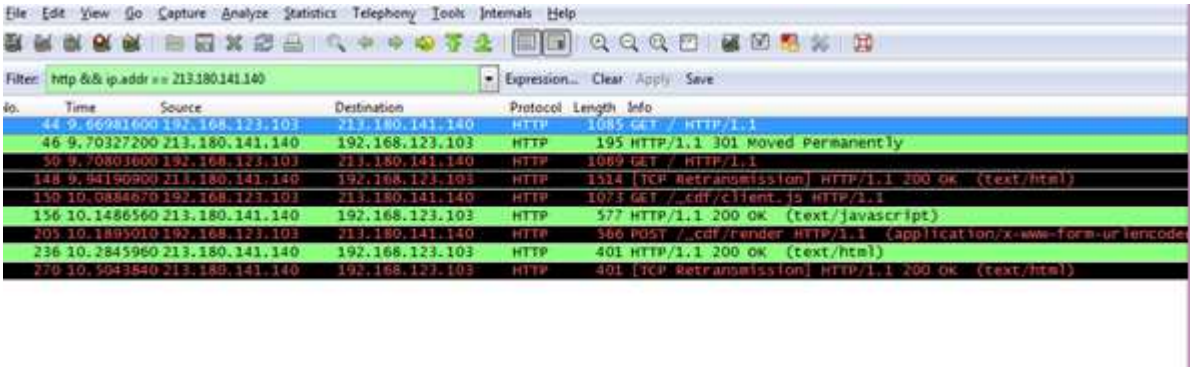
- Co jest w nagłówku TCP? Jakie flagi są związane z żądaniem?



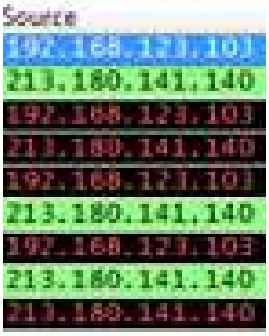
- Znajdź i zaznacz pakiet z odpowiedzią serwera zawierającą stronę internetową. Jakie flagi są ustawione? Czy jest to jedyna odpowiedź TCP serwera? Jeśli nie - dlaczego jest ich więcej? Czym różnią się pozostałe odpowiedzi?

```
Flags: 0x02 (Don't Fragment)
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
```

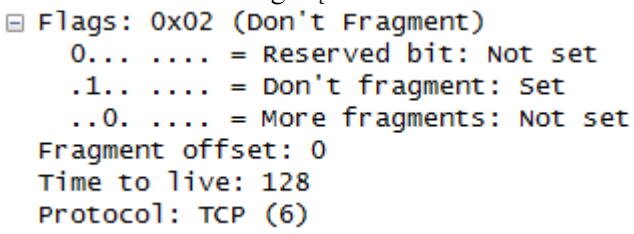
- Zestawianie i zamykanie połączenia TCP
  - Przeanalizuj pakiety tcp poprzedzające pakiet z żądaniem http.



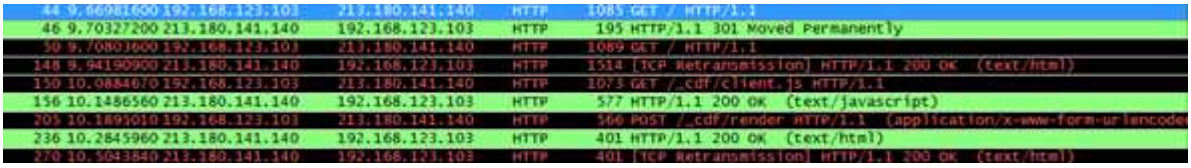
  - Ile jest takich pakietów?
  - Kto inicjuje połączenie?



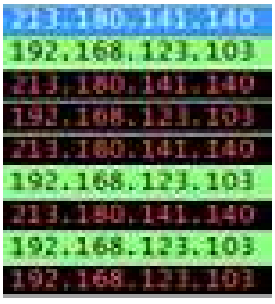
  - Jakie flagi są ustawiane?



  - Przeanalizuj pakiety tcp po pakiecie z odpowiedzią http.



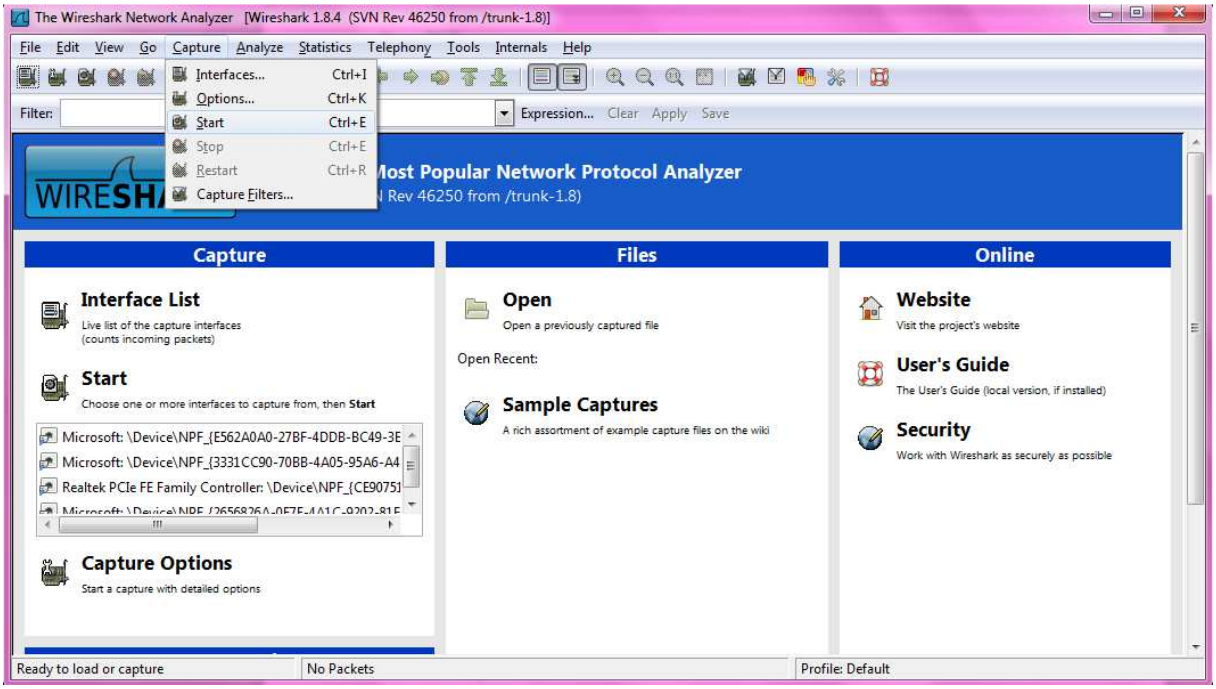
  - Ile jest takich pakietów?
  - Kto zaczyna proces zamykania połączenie?



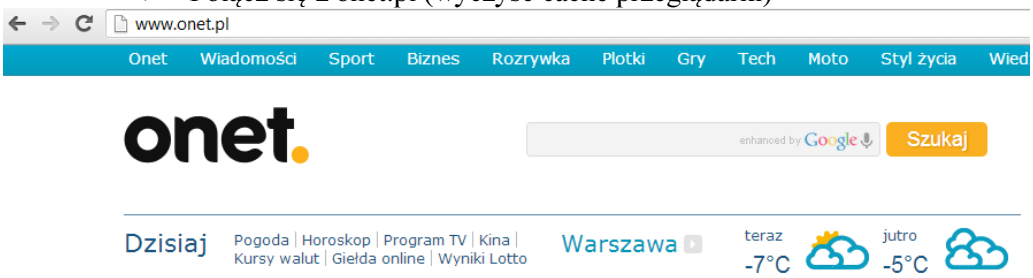
- Jakie flagi są ustawiane?

```
Flags: 0x02 (Don't Fragment)
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
```

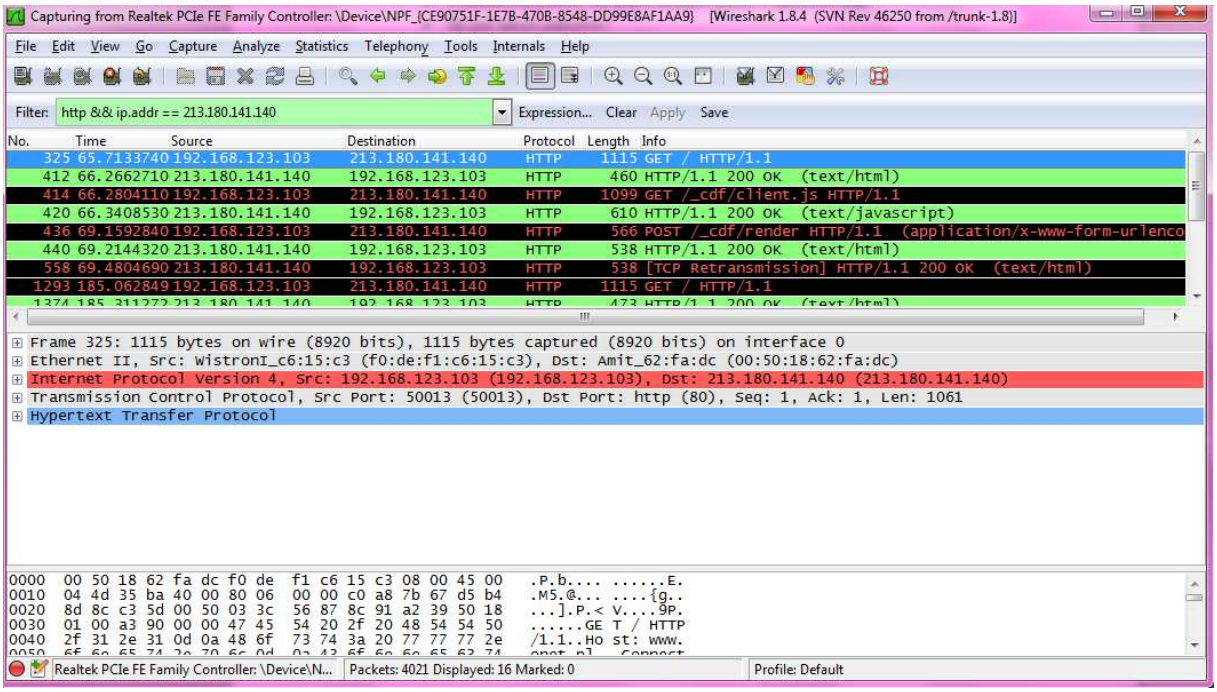
- Połączenia keep-alive w http - zobaczymy jak przeglądarka i serwer obsługują strony wymagające więcej niż jednego żądania http. (Tą część wykonujemy z włączonym łapaniem pakietów)



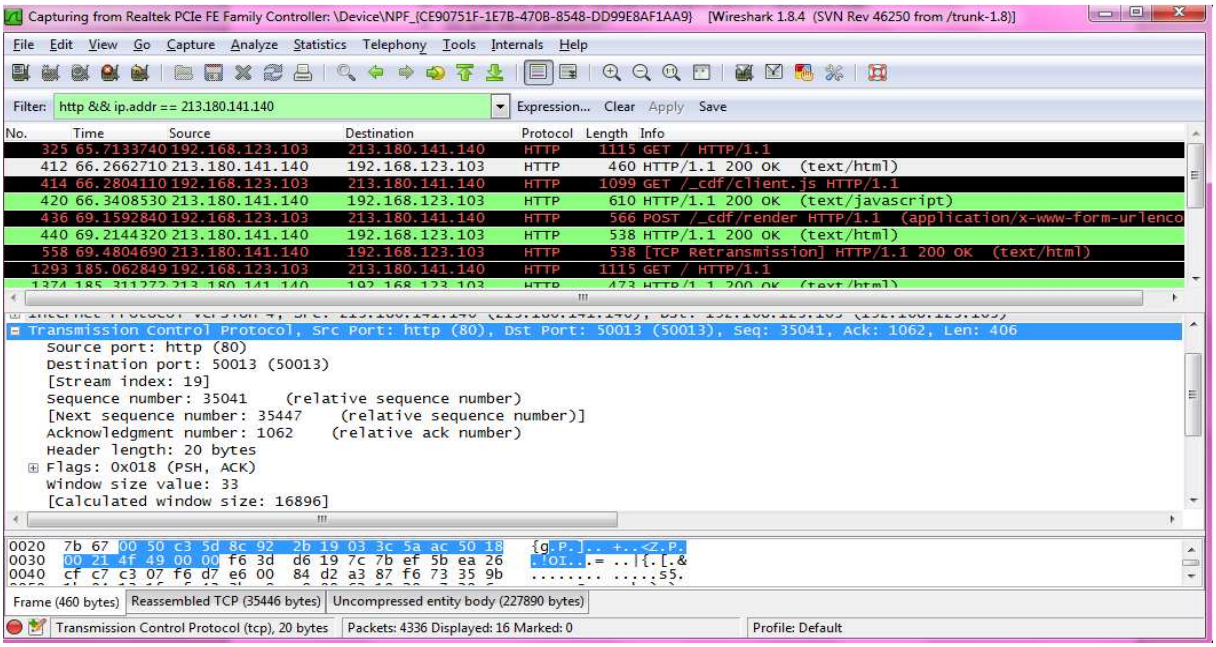
- Połącz się z onet.pl (wyczyść cache przeglądarki)

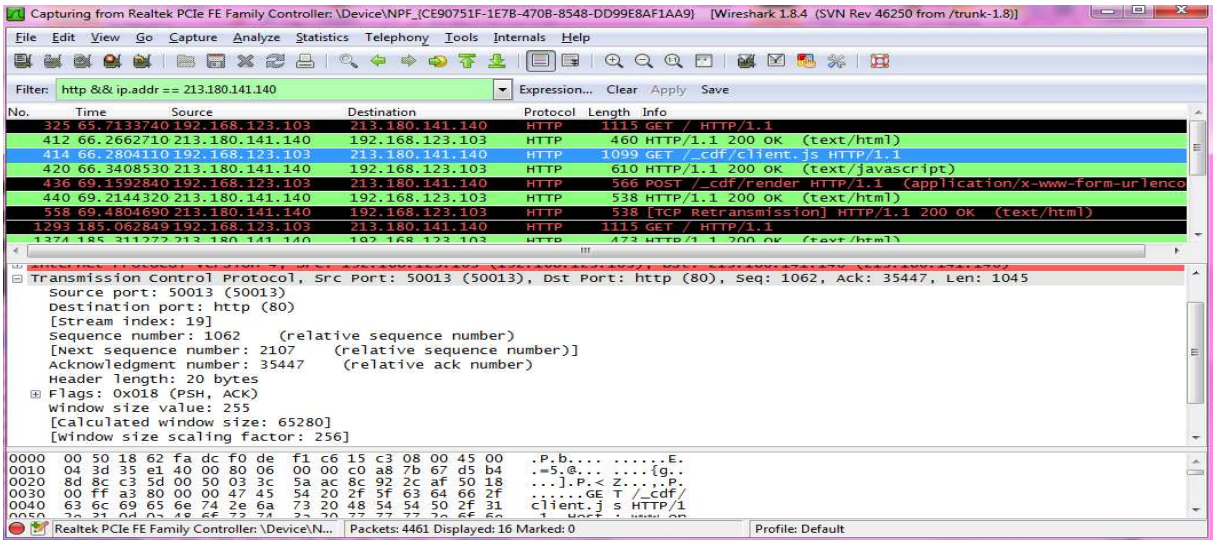


- Znajdź i zaznacz pakiety zawierające dwa żądania http



- Czy dla każdego żądania nawiązywana była oddzielna sesja TCP, czy też oba żądania były obsłużone w jednej sesji?

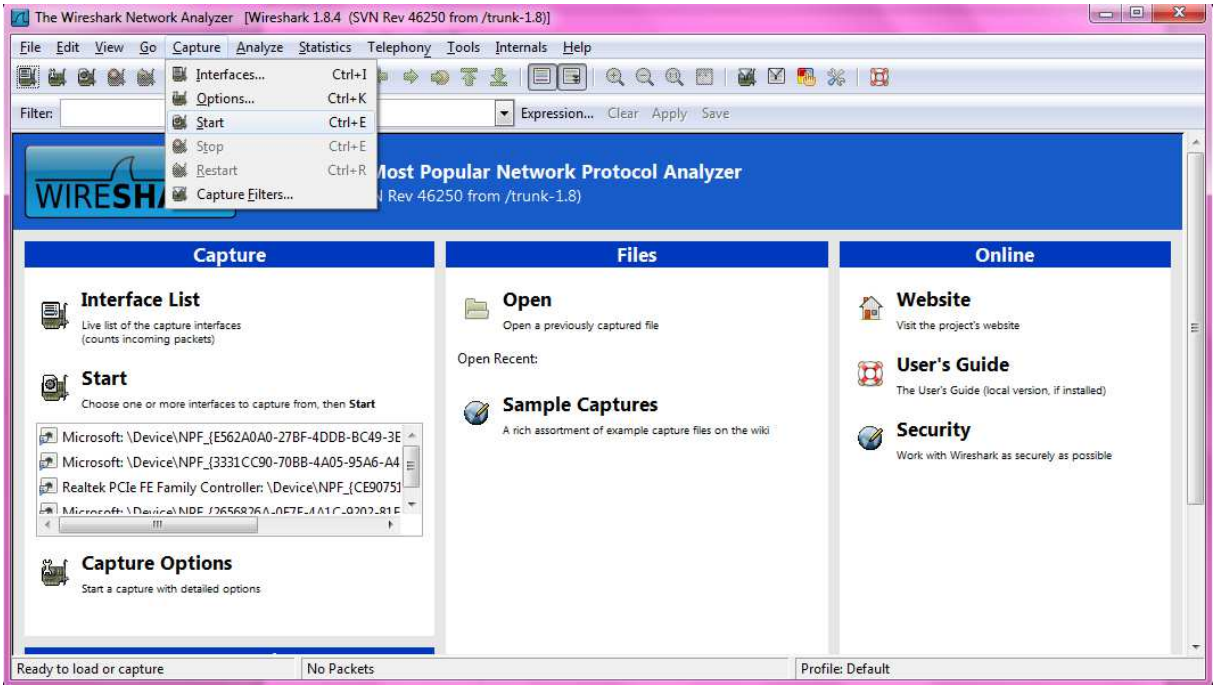




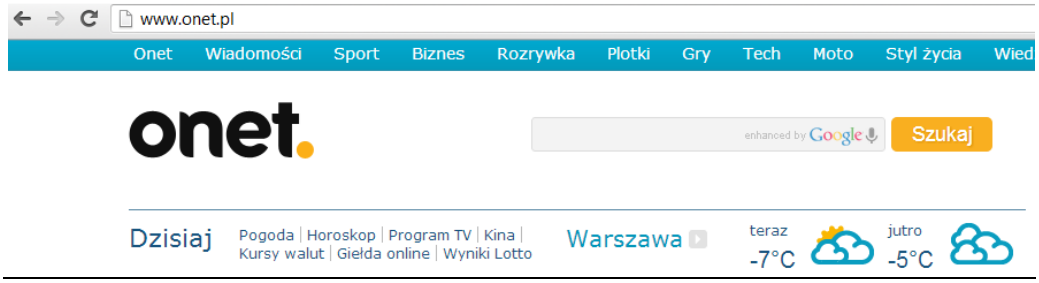
- Co daje niezamykanie połączeń (czyli http w trybie keep-alive)? Jakie są możliwe problemy?

VI. IP

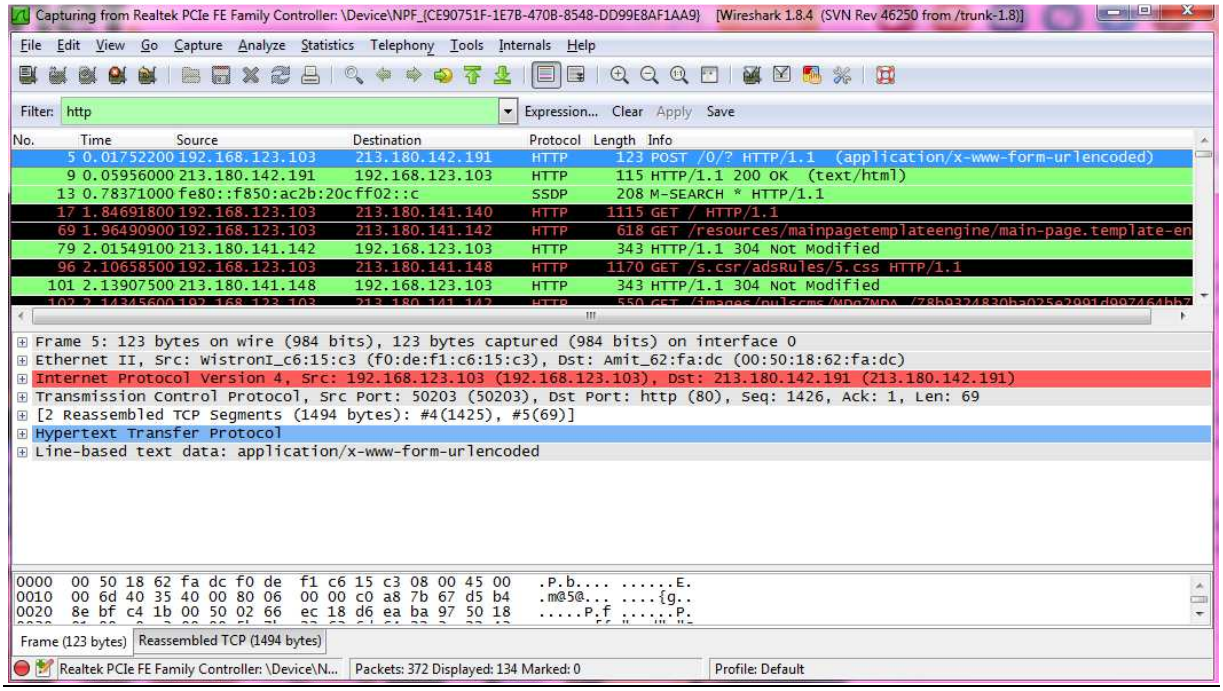
- Nagłówek IPv4 - tą część wykonujemy z włączonym łapaniem pakietów



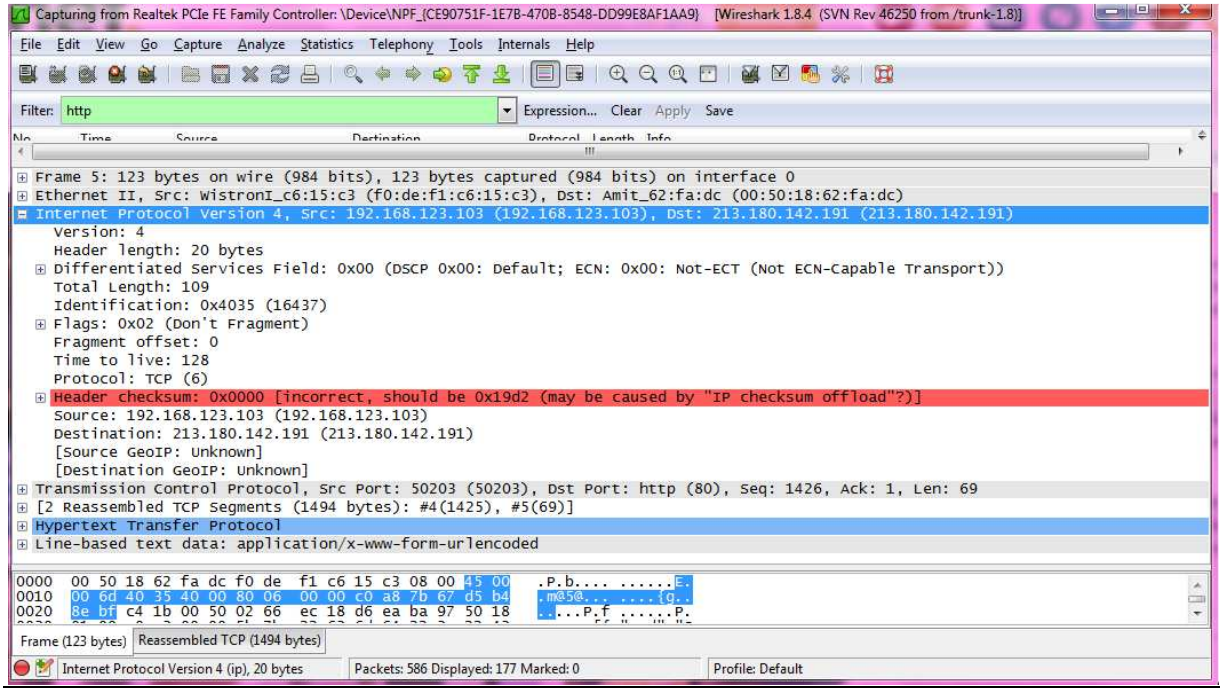
- Połącz się z wybraną stroną (np. onet.pl)



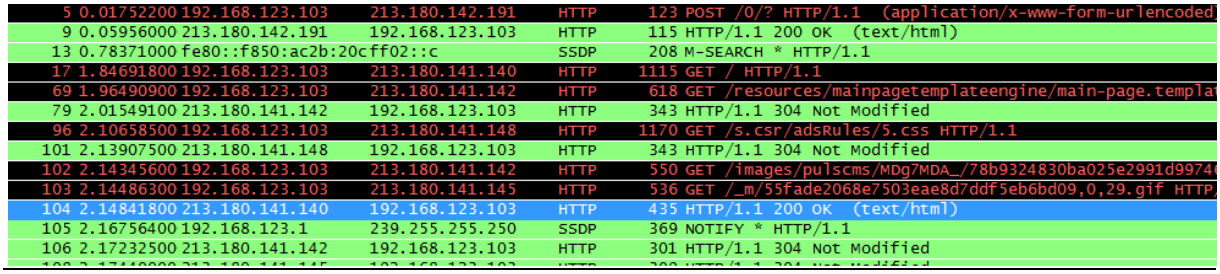
- Wybierz pakiet zawierający żądanie http



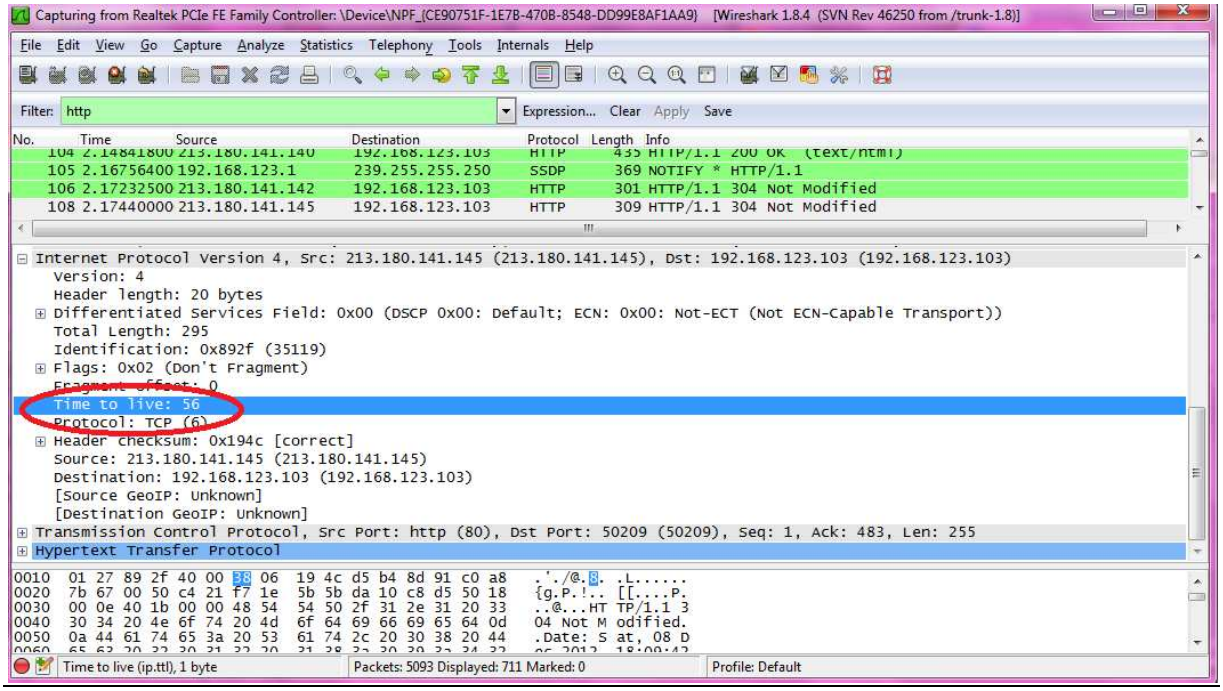
- Przeanalizuj zawartość nagłówka IP



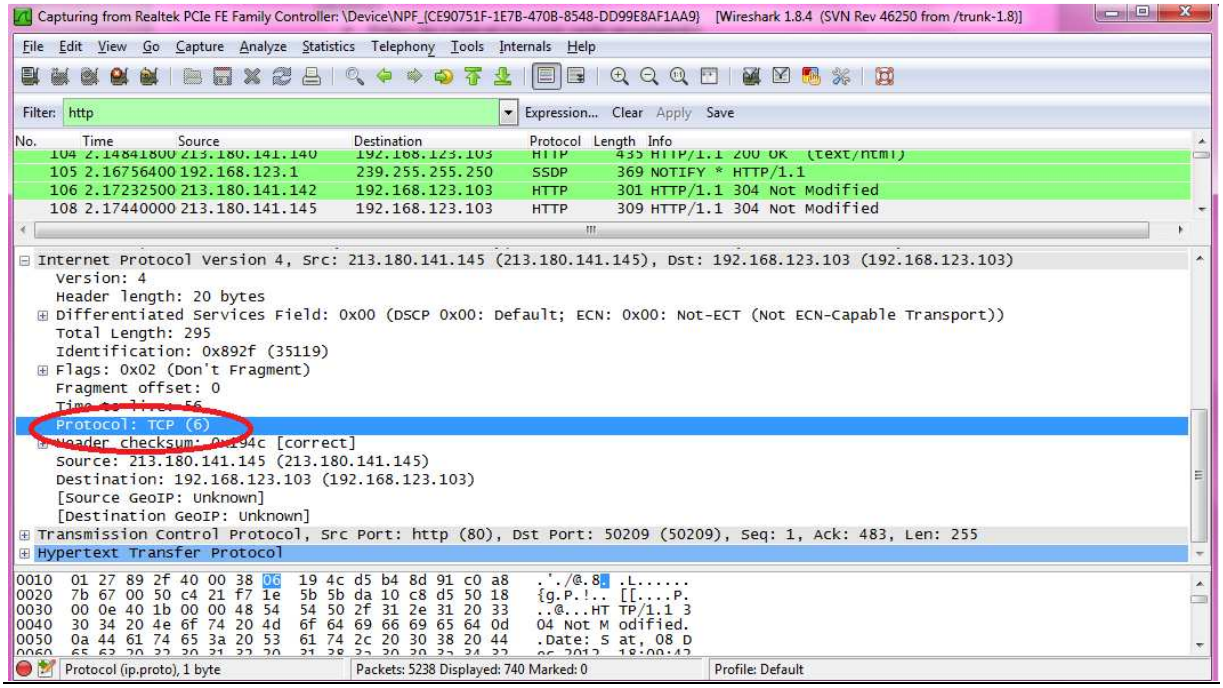
- Wybierz pakiet zawierający odpowiedź http



- Jakie jest TTL?



- W jakim celu wprowadzono pole Protocol?



- Co sugeruje obecność pola "Fragment Offset"?