

SPRAWOZDANIE Z LABORATORIUM

Sieci komputerowych

Łukasz Sujka Karol Kozłowski Karol Nikšcin Mirosław Rogotowicz	Grupa lab.: 4	Termin: WTOREK/N 13¹⁵	Data: 19-12-2006
Ćwiczenie nr 2 Badanie własności modelu warstwowego TCP/IP			Ocena

1. Celem ćwiczenia

Celem ćwiczenia jest zapoznanie się z programem Wireshark (dawniej Ethereal) służącym do analizy ruchu sieciowego oraz poznanie i analiza wybranych protokołów sieciowych.



Ilustracja 1: Logo programu Wireshark

2. Przebieg ćwiczenia.

2.1. Analiza protokołów diagnostycznych – PING

a) host lokalny

Podczas analizy przeprowadzono badanie odpowiedzi na polecenie ping hosta w sieci lokalnej. Wykonano 10 zapytań typu „echo” do hosta określonego nazwą netbios. W pierwszej kolejności komputer wysłał zapytanie do serwera DNS o zwrócenie adresu sieciowego komputera o podanej nazwie lecz go nie otrzymał. Następnie po nieudanej próbie uzyskania adresu IP z serwera nazw został wysłany broadcast typu NBNS (system nazw netbios) w wyniku którego został uzyskany adres IP poszukiwanego komputera. Kolejnym krokiem jaki wykonuje komputer jest wysłanie standardowego broadcastu ARP w celu uzyskania adresu sprzętowego badanego komputera. Po uzyskaniu tych informacji komputer mógł przystąpić do badania komputera wysyłając zapytania „echo” w ramach protokołu ICMP po otrzymaniu poprawnej odpowiedzi (bądź po przekroczeniu timeout'u) komputer wysłał kolejne zapytanie. W międzyczasie badany komputer wysłał zapytanie ARP odpytując nasz komputer o jego adres sprzętowy a także z badającego komputera zostało wysłany pakiet utrzymujący sesję protokołu NBNS.

```
C:\Documents and Settings\Karol>ping -n 10 waski
```

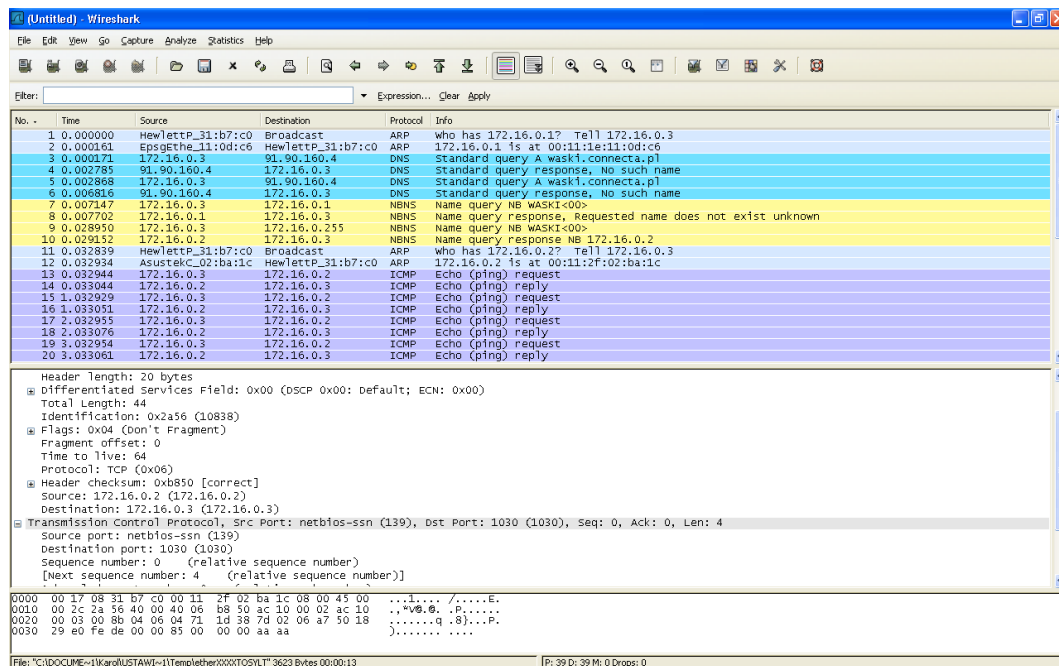
Badanie waski [172.16.0.2] z użyciem 32 bajtów danych:

```
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 172.16.0.2: bajtów=32 czas<1 ms TTL=64
```

Statystyka badania ping dla 172.16.0.2:

Pakiety: Wysłane = 10, Odebrane = 10, Utracone = 0 (0% straty),
Szacunkowy czas błędzenia pakietów w millisekundach:
Minimum = 0 ms, Maksimum = 0 ms, Czas średni = 0 ms

Kod 1: Wynik polecenia ping



Ilustracja 2: Okno programu Wireshark

a) Host zdalny

Operacja od poprzedniej różni się tylko tym, że adres sieciowy uzyskiwany jest z serwera DNS (pozostałe protokoły nie są wykorzystywane). Po uzyskaniu adresu IP następuje seria zapytań typu „echo”.

2.2. Analiza protokołów diagnostycznych – TRACE

a) host lokalny

W wyniku wywołania polecenia wyznaczania trasy pakietu w sieci lokalnej (gdzie nie ma żadnych przeskoków) spowodowała wysłanie 3ch zapytań do hosta docelowego.

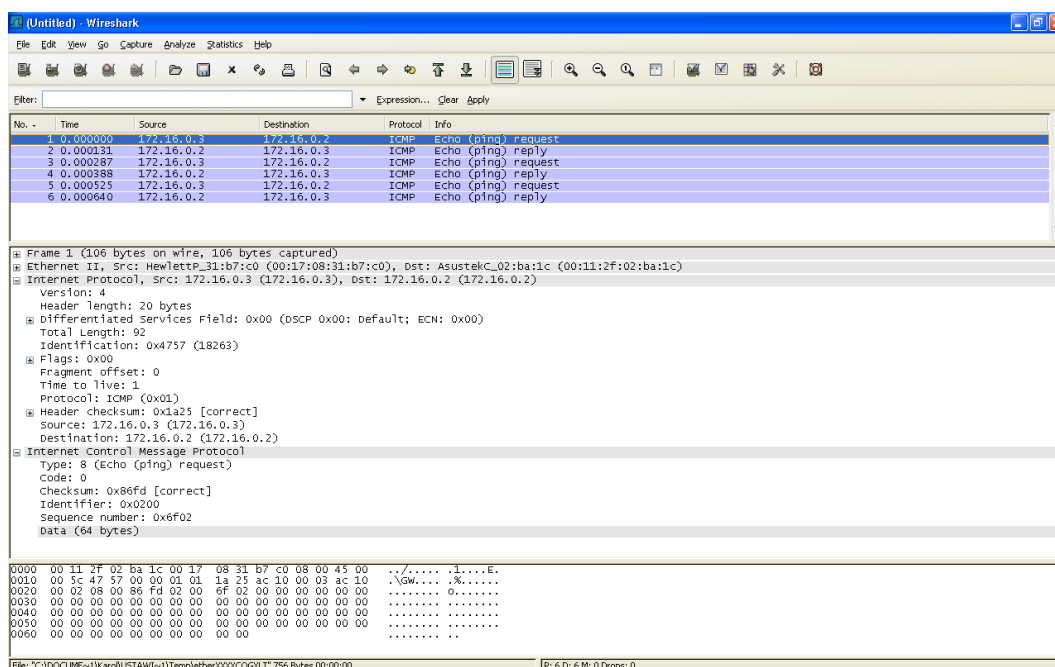
```
C:\Documents and Settings\Karol>tracert waski
```

```
Trasa śledzenia do waski [172.16.0.2]  
przewyższa maksymalną liczbę przeskoków 30
```

```
1      <1 ms      <1 ms      <1 ms      172.16.0.2
```

Śledzenie zakończone.

Kod 2: Wynik polecenia tracert



Ilustracja 3: Okno analizatora po wykonaniu śledzenia trasy

b) host zdalny

W tym przypadku sytuacja jest bardziej złożona ponieważ na początku następuje odpytanie o adres sieciowy a po jego uzyskaniu komputer zaczyna rozpoznawać długość trasy poprzez wysyłanie serii zapytań typu „echo” do hosta docelowego zwiększając co 3 zapytania wielkość TTL, która na początku wynosi 1. Kiedy komputer przestanie otrzymywać od serwerów pośredniczących odpowiedź informującą o przekroczeniu czasu życia pakietu uznaje że pakiet dotarł do komputera docelowego i wobec tego zna już długość trasy. Natomiast adresy poszczególnych hop'ów uzyskuje odczytując adres źródłowy pakietów informujących o przekroczeniu czasu życia pakietu. Dodatkowo program wykonuje rewersyjne zapytanie DNS wysyłając odpowiednie zapytanie do serwera nazw, z którego może otrzymać nazwę domenową komputera.

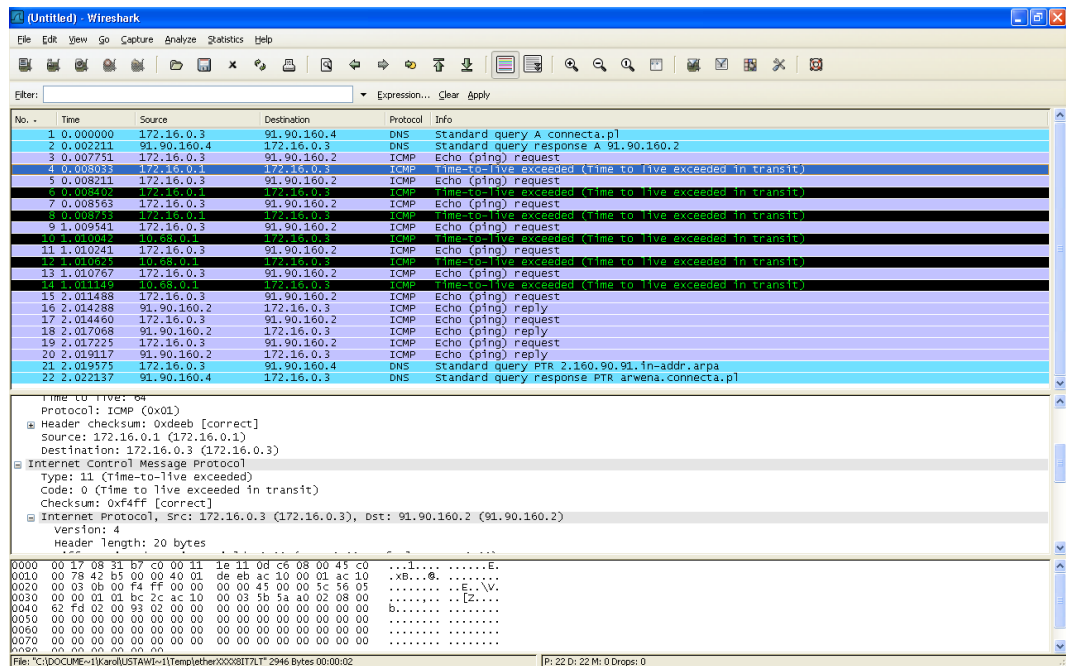
C:\Documents and Settings\Karol>tracert connecta.pl

Trasa śledzenia do connecta.pl [91.90.160.2]
przewyższająca maksymalną liczbę przeskoków 30

1	<1 ms	<1 ms	<1 ms	172.16.0.1
2	<1 ms	<1 ms	<1 ms	10.68.0.1
3	2 ms	2 ms	1 ms	arwena.connecta.pl [91.90.160.2]

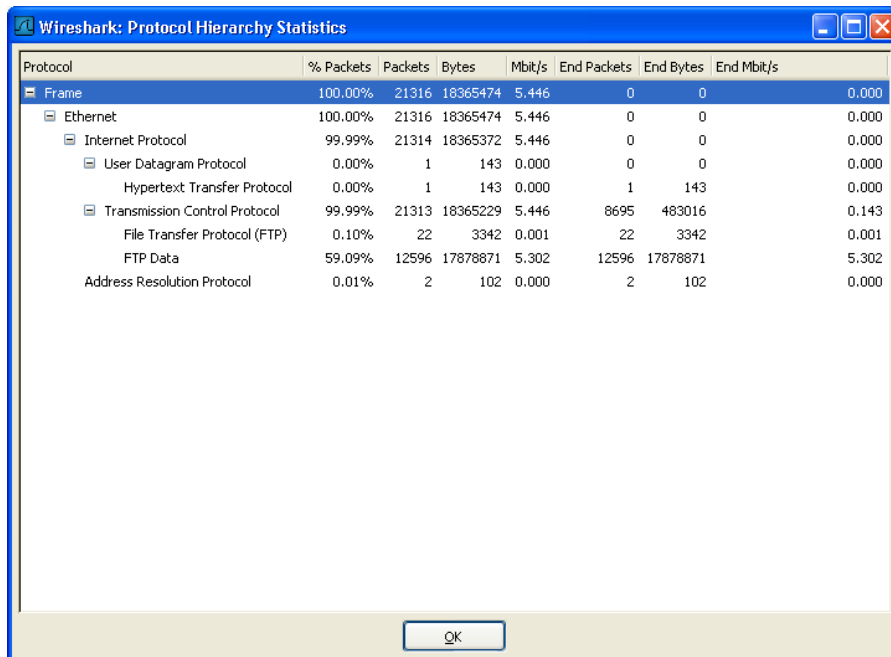
Śledzenie zakończone.

Kod 3: Wynik polecenia tracert



Ilustracja 4: Okno analizatora po wykonaniu śledzenia trasy

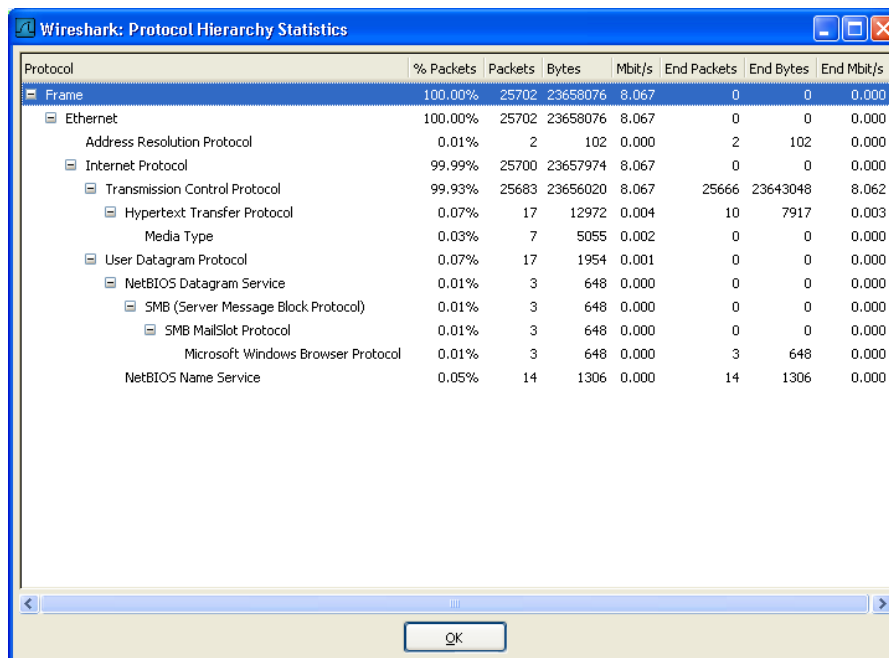
2.3. Analiza ruchu sieciowego – protokół ftp



Ilustracja 5: Hierarchia protokołów

W tej części dokonaliśmy analizy ruchu sieciowego podczas pobierania pliku o wielkości 17 183 831 bajtów poprzez protokół ftp. Statystyki pokazują jednak, że w ramach protokołu ftp pobranych zostało 18 365 229 co stanowi przyrost rzędu 6,9%.

2.4. Analiza ruchu sieciowego – protokół http

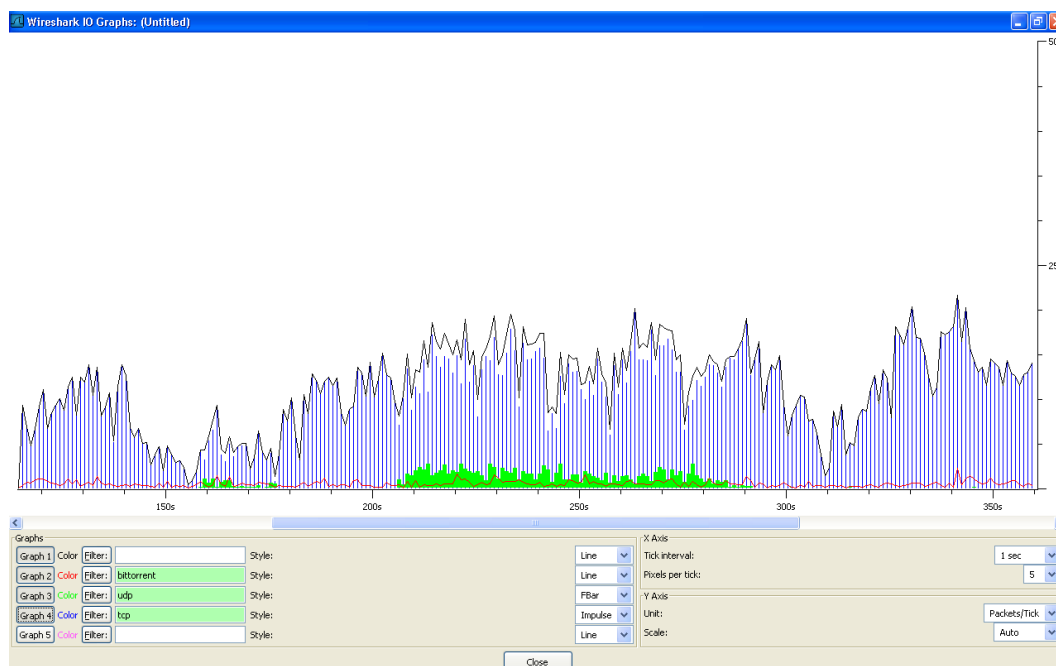


Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00%	25702	23658076	8.067	0	0	0.000
Ethernet	100.00%	25702	23658076	8.067	0	0	0.000
Address Resolution Protocol	0.01%	2	102	0.000	2	102	0.000
Internet Protocol	99.99%	25700	23657974	8.067	0	0	0.000
Transmission Control Protocol	99.93%	25683	23656020	8.067	25666	23643048	8.062
Hypertext Transfer Protocol	0.07%	17	12972	0.004	10	7917	0.003
Media Type	0.03%	7	5055	0.002	0	0	0.000
User Datagram Protocol	0.07%	17	1954	0.001	0	0	0.000
NetBIOS Datagram Service	0.01%	3	648	0.000	0	0	0.000
SMB (Server Message Block Protocol)	0.01%	3	648	0.000	0	0	0.000
SMB MailSlot Protocol	0.01%	3	648	0.000	0	0	0.000
Microsoft Windows Browser Protocol	0.01%	3	648	0.000	3	648	0.000
NetBIOS Name Service	0.05%	14	1306	0.000	14	1306	0.000

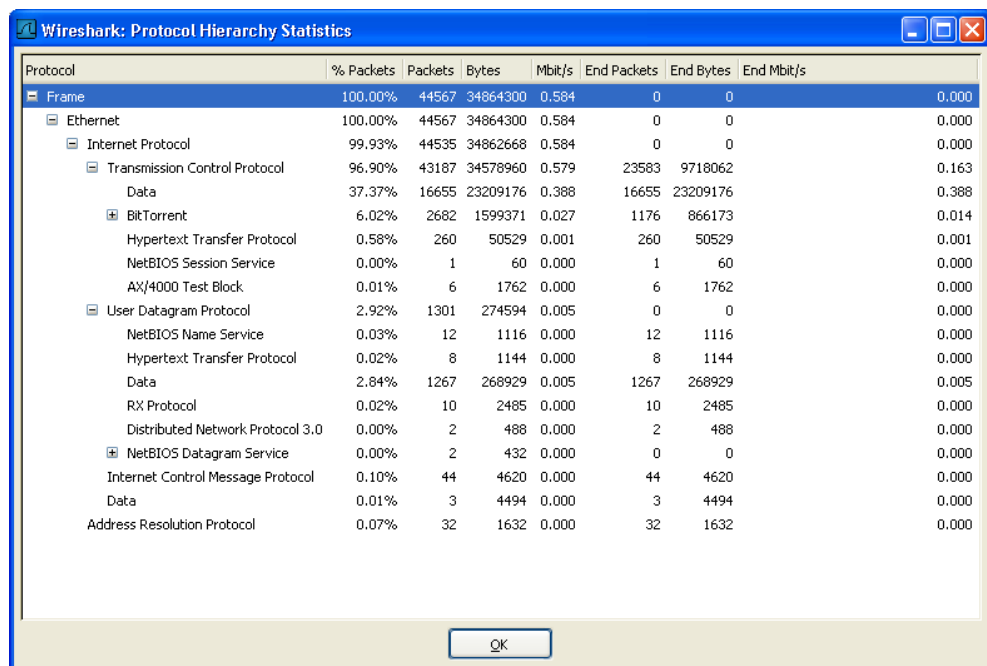
Ilustracja 6: Hierarchia protokołów

W przypadku protokołu http narzut protokołu wyniósł więcej niż w przypadku protokołu ftp bo ponad 8,4%

2.5. Analiza ruchu sieciowego – protokół rozproszony torrent



Ilustracja 7: Wykres czasowy przepływu pakietów.



Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00%	44567	34864300	0.584	0	0	0.000
Ethernet	100.00%	44567	34864300	0.584	0	0	0.000
Internet Protocol	99.93%	44535	34862668	0.584	0	0	0.000
Transmission Control Protocol	96.90%	43187	34578960	0.579	23583	9718062	0.163
Data	37.37%	16655	23209176	0.388	16655	23209176	0.388
BitTorrent	6.02%	2682	1599371	0.027	1176	866173	0.014
Hypertext Transfer Protocol	0.58%	260	50529	0.001	260	50529	0.001
NetBIOS Session Service	0.00%	1	60	0.000	1	60	0.000
AX/4000 Test Block	0.01%	6	1762	0.000	6	1762	0.000
User Datagram Protocol	2.92%	1301	274594	0.005	0	0	0.000
NetBIOS Name Service	0.03%	12	1116	0.000	12	1116	0.000
Hypertext Transfer Protocol	0.02%	8	1144	0.000	8	1144	0.000
Data	2.84%	1267	268929	0.005	1267	268929	0.005
RX Protocol	0.02%	10	2485	0.000	10	2485	0.000
Distributed Network Protocol 3.0	0.00%	2	488	0.000	2	488	0.000
NetBIOS Datagram Service	0.00%	2	432	0.000	0	0	0.000
Internet Control Message Protocol	0.10%	44	4620	0.000	44	4620	0.000
Data	0.01%	3	4494	0.000	3	4494	0.000
Address Resolution Protocol	0.07%	32	1632	0.000	32	1632	0.000

Ilustracja 8: Hierarchia protokołów

W tym punkcie postanowiliśmy przeanalizować zachowanie się protokołu internetowego „torrent”. Jest on powszechnie wykorzystywany do wymiany plików w sieci internet. Jak łatwo zaobserwować w trakcie trwania transmisji wykorzystywany jest głównie protokół TCP niemniej jednak w pewnym momencie można zaobserwować duże wykorzystanie transmisji UDP – związane jest to z charakterystyką protokołu torrent – wykorzystuje on pakiety UDP do omijania maskarady.

3. Wnioski

Wireshark (Ethereal) to potężne narzędzie do analizy ruchu sieciowego. Pozwala ono na obserwację wykorzystania połączenia internetowego zarówno przez dostawców internetowych jak i w domu. Za jego pomocą można przeanalizować działanie różnych protokołów sieciowych lub mechanizmów sieciowych – diagnostyki, routingu. Pomocny jest również w odwrotnej inżynierii zamkniętych protokołów sieciowych które nie mają otwartej specyfikacji.