**Zadanie 3 - QOS**

**Vladimír Jančich, Roman Kasala**

# Topológia

Dvojica smerovačov prepojená sériovým káblom na rozhraní s0/0/0. Koncové počítače sú k obom smerovačom pripojené na rozhraní fa0/0. IP adresa počítača pripojeného k R1 bola 10.0.1.10/24. Rozhranie fa0/0 smerovača R1 malo IP adresu 10.0.1.1/24. Rozhranie s0/0/0 smerovača R1 malo IP adresu 10.0.12.1/24 a rozhranie s0/0/0 smerovača R2 dostalo IP adresu 10.0.12.2/24. Rozhranie fa0/0 smerovača R2 malo IP adresu 10.0.2.1/24 a počítač pripojený k tomuto rozhraniu používal IP adresu 10.0.2.10/24. Medzi smerovačmi fungovalo jednoduché statické smerovanie.

**R1**

**R2**

**10.0.1.0 / 24**

**10.0.2.0 / 24**

**10.0.12.0 / 24**

**128 k**

**clock rate**

**bandwidth**

**Sender**

**Reciever, WS**

**S0**

**S0**

**f0**

**f0**

# 1. Cisco AutoQOS

**a. otestovať na danej verzii IOS**

V našej topológii používal smerovač R1 verziu IOS 15.5 (2) XB a smerovač R2 mal verziu 12.4 (13r) T5. Oba smerovače podporujú Cisco AutoQOS.

**b. Zistiť ako a v akom režime sa spúšťa, zistiť čo všetko daná funkcionalita nastaví a kde všade (aké QoS pravidlá aktivuje)**

Cisco AutoQOS sa spúšťa príkazom *auto qos voip* a to v režime konfigurácie rozhrania. My sme túto konfigurácii zapli na výstupnom rozhraní smerovača R1 teda na rozhraní s0/0/0. Výsledkom bola automatická konfigurácia týchto class-map:

class-map match-any AutoQoS-VoIP-Remark

match ip dscp ef

match ip dscp cs3

match ip dscp af31

class-map match-any AutoQoS-VoIP-Control-UnTrust

match access-group name AutoQoS-VoIP-Control

class-map match-any AutoQoS-VoIP-RTP-UnTrust

match protocol rtp audio

match access-group name AutoQoS-VoIP-RTCP

AutoQOS sa tiež postaral o automatickú konfiguráciu týchto policy-map:

policy-map AutoQoS-Policy-UnTrust

class AutoQoS-VoIP-RTP-UnTrust

priority percent 70

set dscp ef

class AutoQoS-VoIP-Control-UnTrust

bandwidth percent 5

set dscp af31

class AutoQoS-VoIP-Remark

set dscp default

class class-default

fair-queue

A došlo k vytvoreniu takýchto access-listov:

ip access-list extended AutoQoS-VoIP-Control

permit tcp any any eq 1720

permit tcp any any range 11000 11999

permit udp any any eq 2427

permit tcp any any eq 2428

permit tcp any any range 2000 2002

permit udp any any eq 1719

permit udp any any eq 5060

ip access-list extended AutoQoS-VoIP-RTCP

permit udp any any range 16384 32767

Poznámka: v našom prípade po zadaní príkazu cisco qos voip na rozhraní s0/0/0 smerovača R1 došlo k automatickému vytvoreniu rozhrania Multilink 5, pričom systém tomuto rozhraniu pridelil IP adresu, ktorá bola dovtedy pridelená fyzickému rozhraniu s0/0/0 a bola prerušená konektivita. Ani odobranie tejto IP adresy z rozhrania Multilink 5 a spätné pridelenie na s0/0/0 však problém nevyriešilo a sériové rozhranie hlásilo protokol v stave down. Po hlbšom skúmaní sme zistili, že príkaz cisco qos voip automaticky zmení typ enkapsulácie z HDLC na PPP pričom na sériovom rozhraní smerovača R2 bola aj naďalej HDLC enkapsulácia. Tieto rozhrania sa tak medzi sebou nevedeli dorozumieť a problém vyriešila zmena enkapsulácie rozhrania s0/0/0 na smerovači R1 na HDLC.

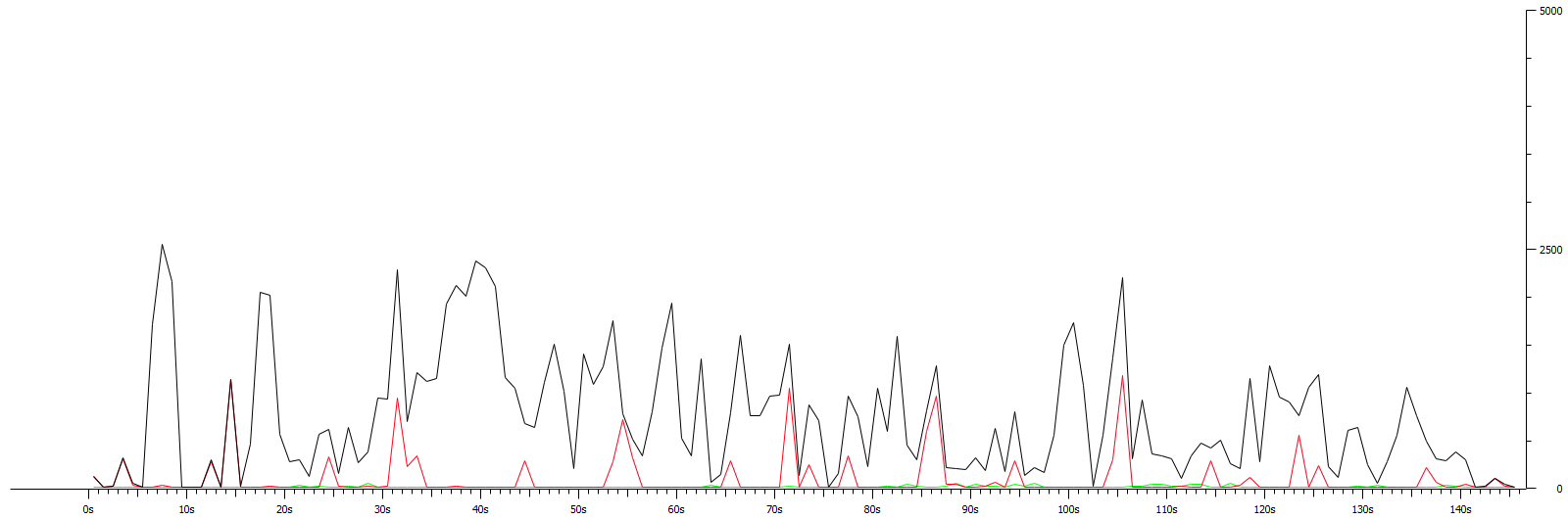
# 2. Značkovanie paketov na vstupnom rozhraní smerovača

**a. Cez D-ITG vygeneruj 3 rôzne typy prevádzky (intenzita tokov tak, aby linka stíhala): FTP, HTTP, HTTPS.**

My sme tieto typy prevádzky generovali na živej sieti. Načítali sme video z FTP servera, cez YouTube, ktorý reprezentoval HTTPS spojenie sme pozerali druhé video a HTTP prevádzku sme generovali jednoduchých obnovovaním nezabezpečenej webovej stránky SHMÚ. Neskôr sme však generovali túto prevádzku vrátane sFTP cez D-ITG.

**b. Odchyť dáta Wiresharkom na cieľovej stanici a analyzuj či D-ITG generuje vhodné dáta, pozri aj IO graphs**

Generovali sme správnu prevádzku. Čierna krivka je FTP, červená HTTPS a nevýrazná zelená http. Že občas zelená krivka vyskočí je vidieť iba po výraznom priblížení nakoľko obnovenie stránky malo malý podiel na celkovej prevádzke a videá zabrali väčšinu z priepustnosti.



**c. klasifikácia, značkovanie**

Pre klasifikovanie FTP, HTTP a HTTPS prevádzky sme si vytvorili 3 class-mapy pomocou nasledujúcich príkazov:

class-map match-any HTTP

match protocol http

class-map match-any FTP

match protocol secure-ftp

match protocol ftp

class-map match-any HTTPS

match protocol secure-http

Následne sme im v policy-map pridali značky. FTP prevádzka dostala DSCP značku 1, HTTP dostala DSCP 2 a HTTPS dostala DSCP 3:

policy-map znackuj

class FTP

set dscp 1

class HTTP

set dscp 2

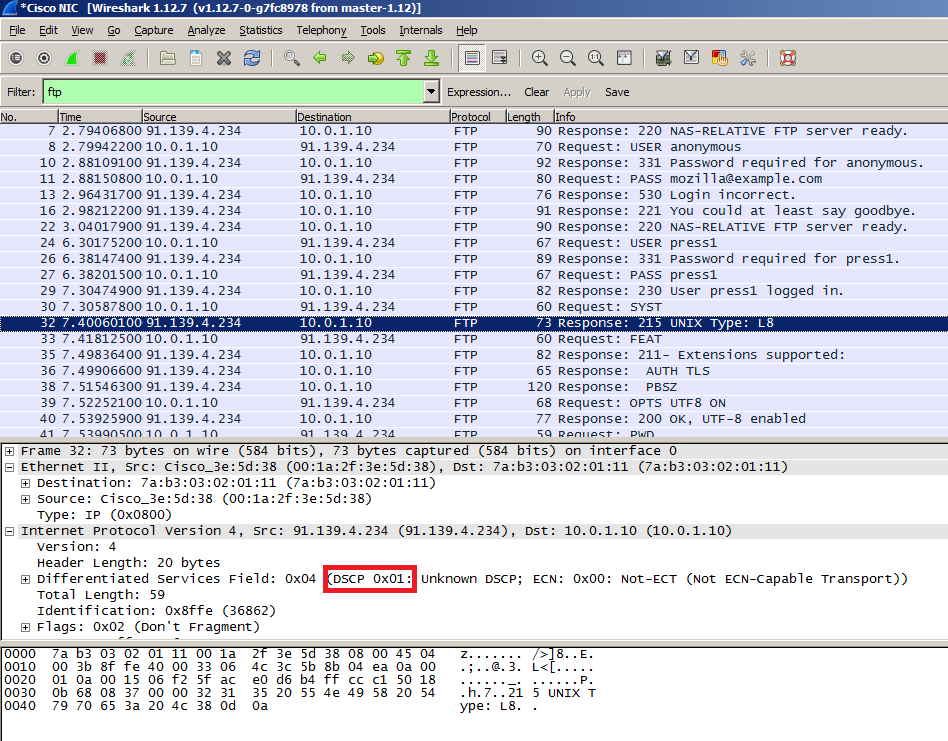
class HTTPS

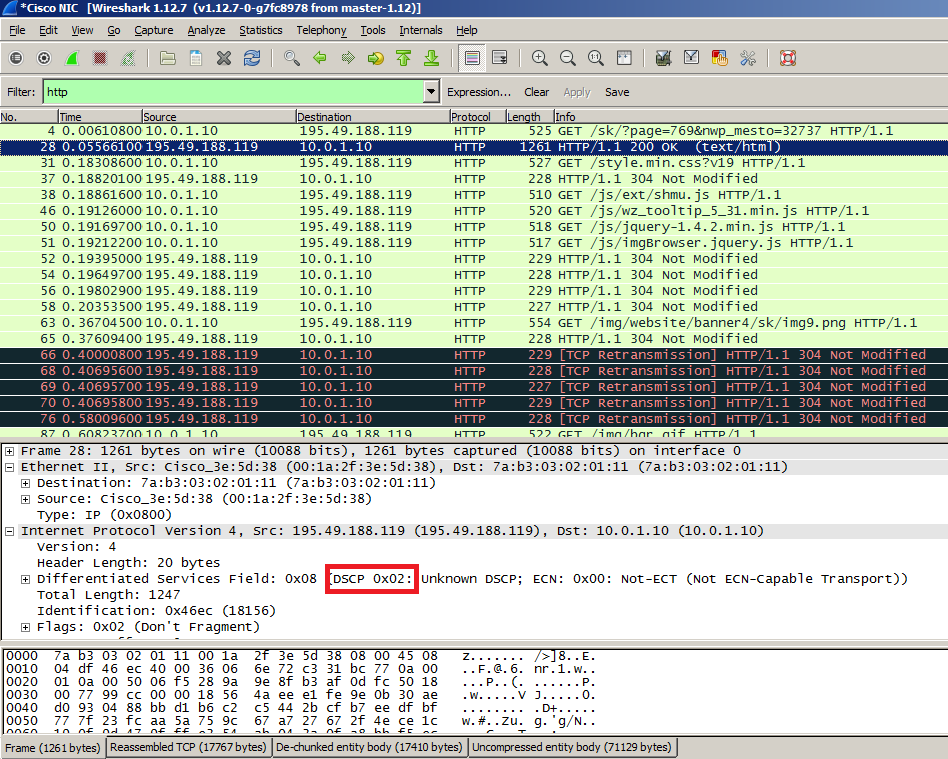
set dscp 3

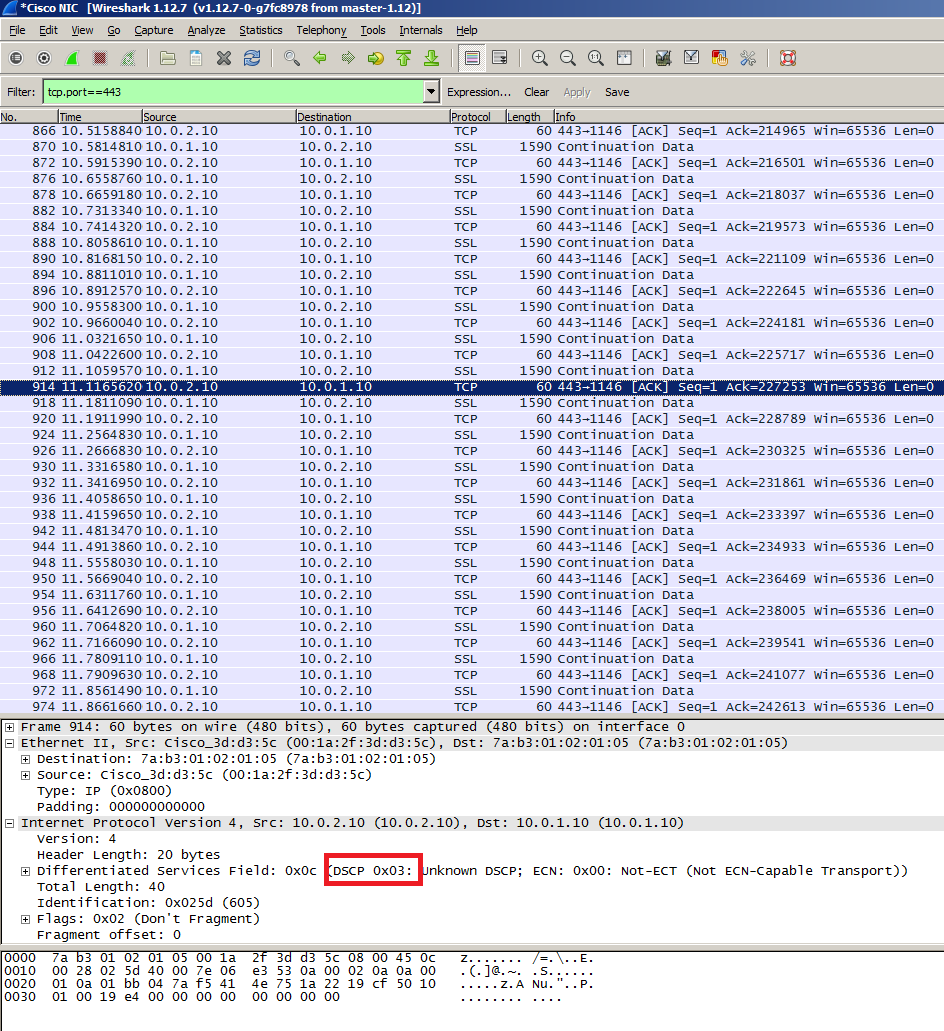
Nakoniec bolo potrebné aplikovať policy-map na rozhranie f0/0 smerovača R1 a zapnúť NBAR pre rozpoznávanie protokolov:

service-policy output znackuj

ip nbar protocol-discovery



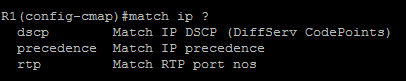




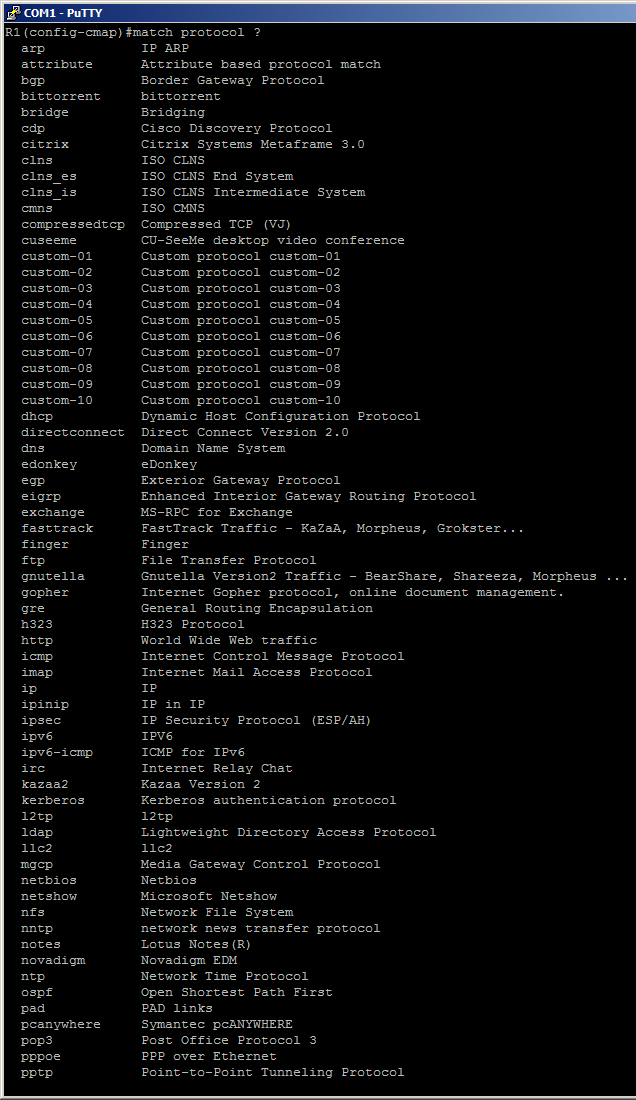
Poznámka: značkovanie fungovalo korektne u protokolov FTP a HTTP v prípade, že bol smerovač R1 zapojený do internetu a generovali sme reálnu prevádzku z kroku 2b. HTTPS značkovanie z neznámych príčin nefungovalo. Pri generovaní umelej prevádzky cez D-ITG zas naopak fungovalo FTP a HTTPS, no HTTP z neznámych príčin nie.

**d. Preskúmaj parametre príkazov**

Príkaz match-ip:

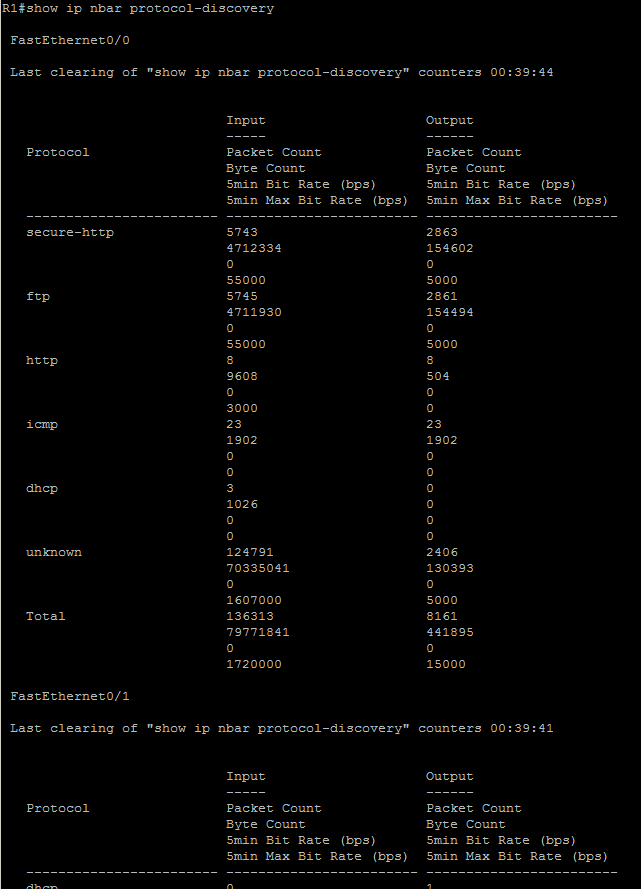


Príkaz match protocol:

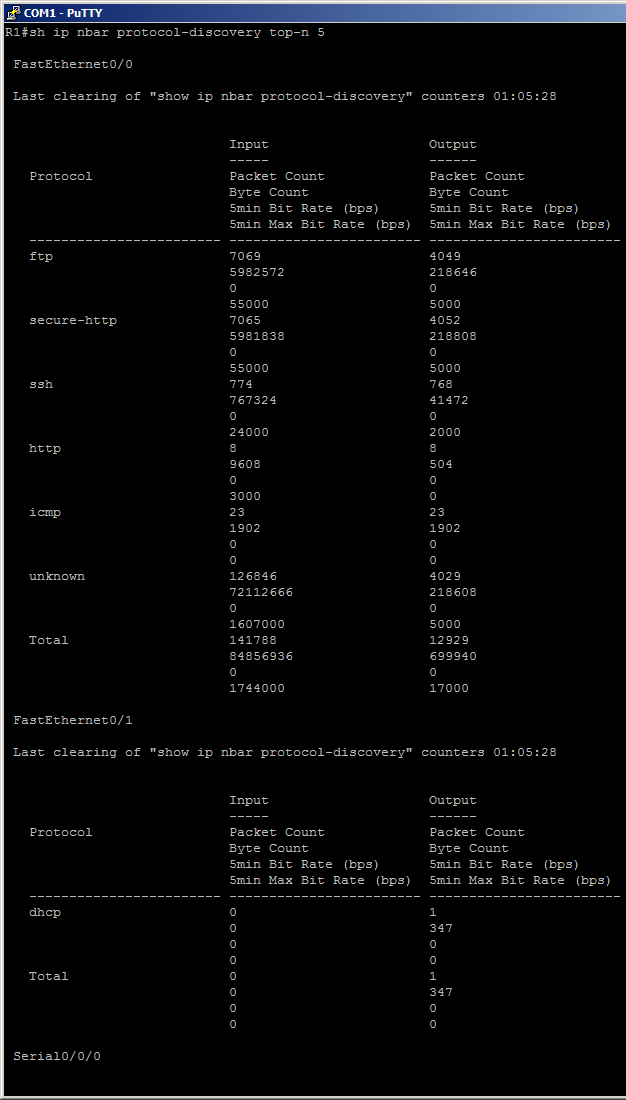


# 3. Sledovanie štatistík NBAR

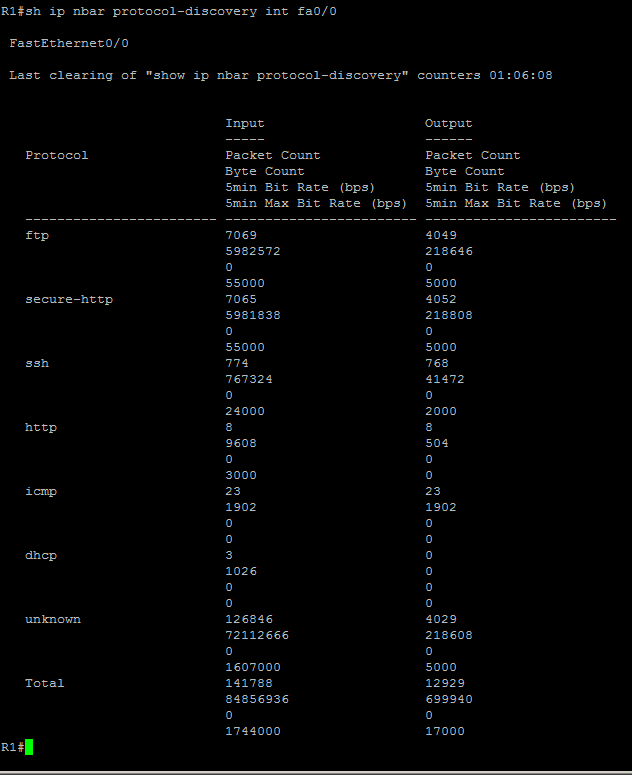
**a. show ip nbar discovery-protocol**



**b. show ip nbar protocol-discovery top-n 5**

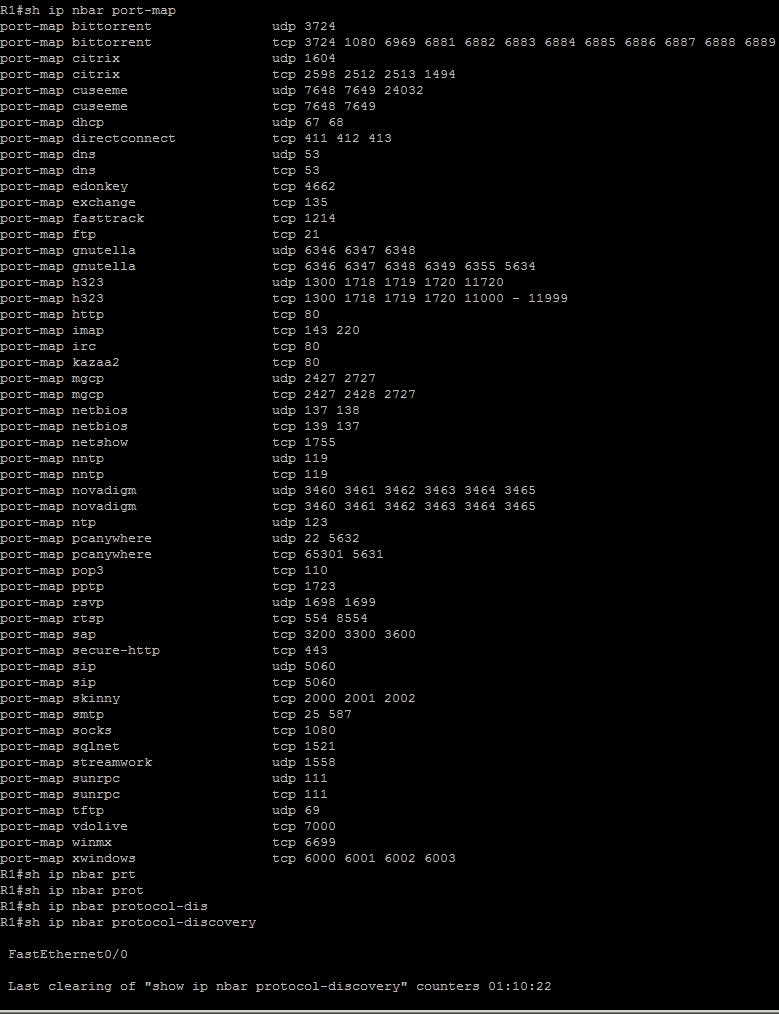
****

**c. show ip nbar protocol-discovery int f0/x**

****

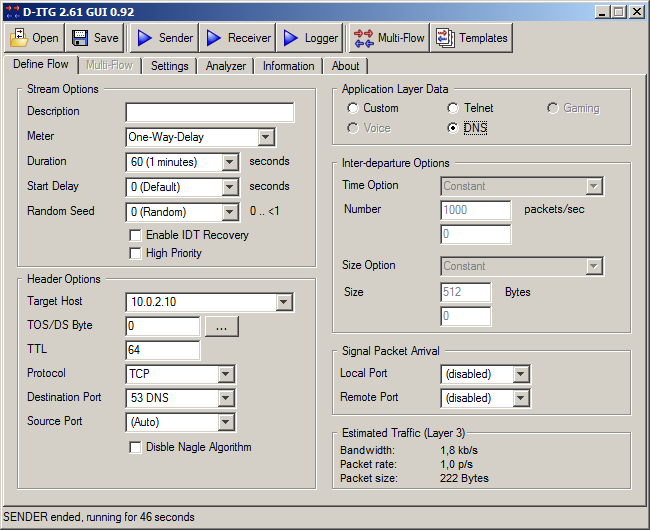
# 5. bonusová úloha

**a. Zistite, ktoré z protokolov v D-ITG pozná NBAR (show ip nbar port-map) a aj ich korektne rozpozná.**

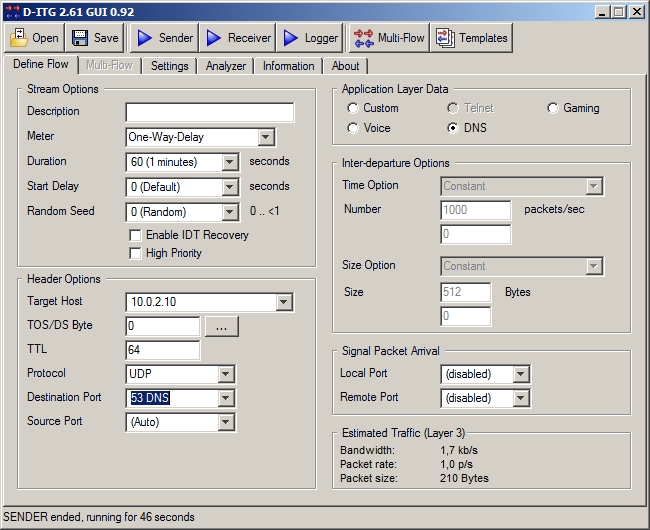


NBAR korektne rozpoznal z prednastavených profilov DNS. Telnet nerozpoznal.

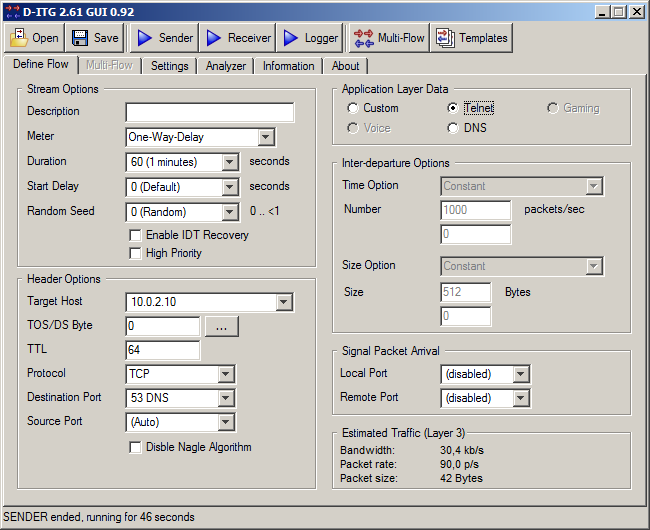
* Parametre toku v D-ITG pre TCP DNS



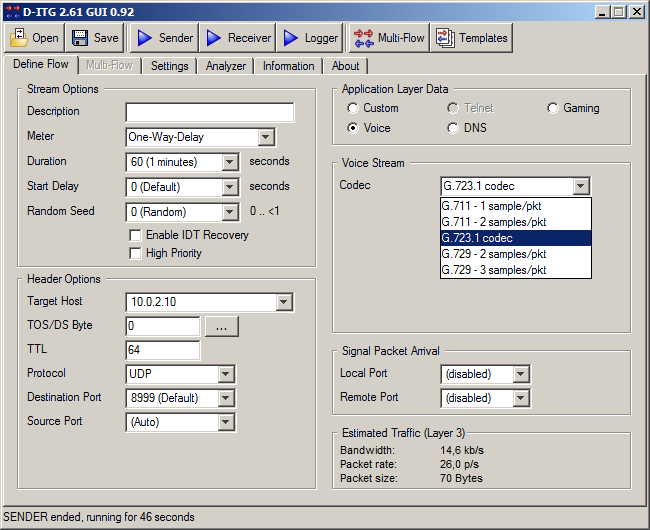
* Parametre toku v D-ITG pre UDP DNS



* Parametre toku v D-ITG pre telnet



* Parametre toku v D-ITG pre Voice



* Parametre toku v D-ITG pre Gaming

