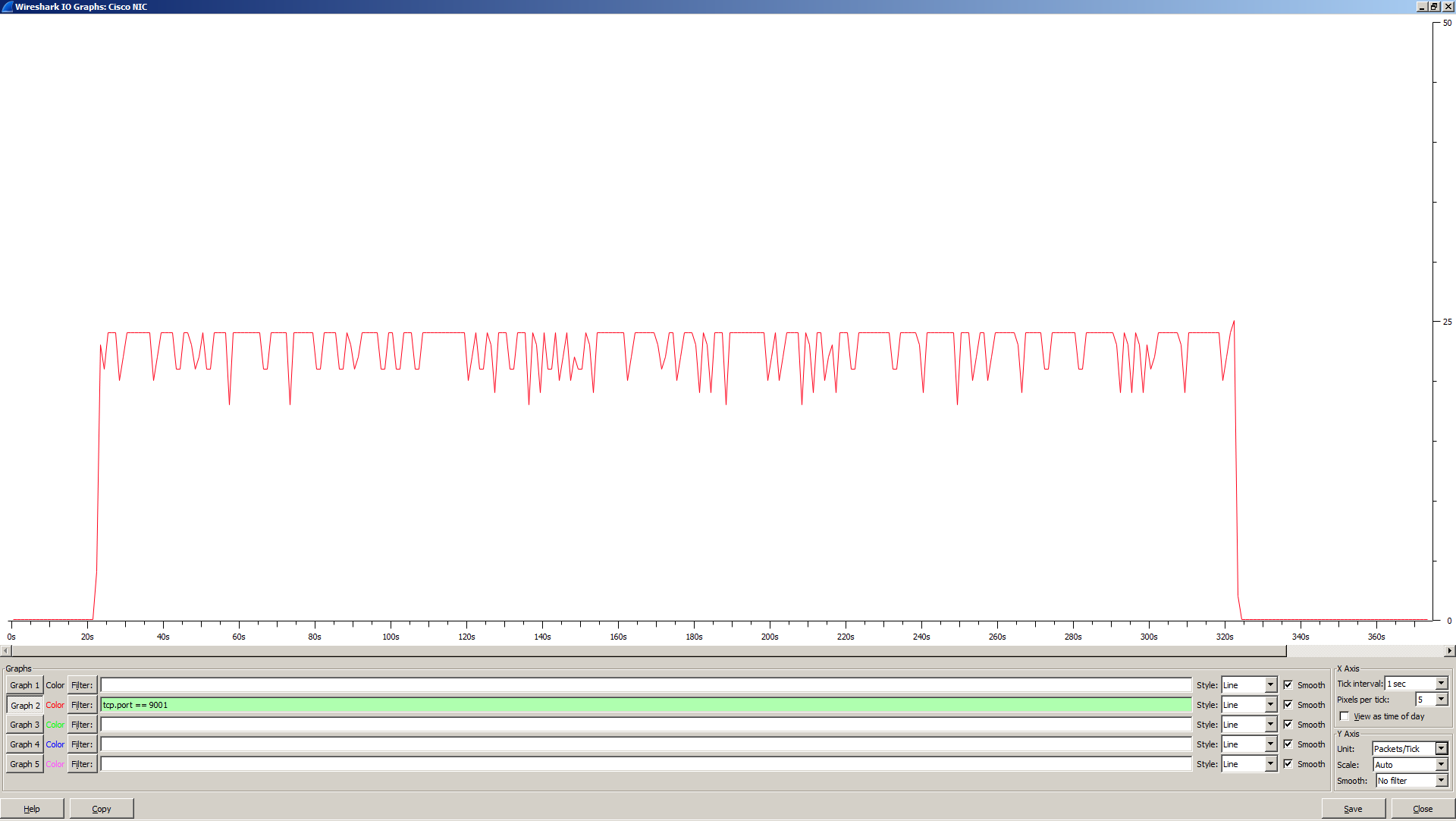
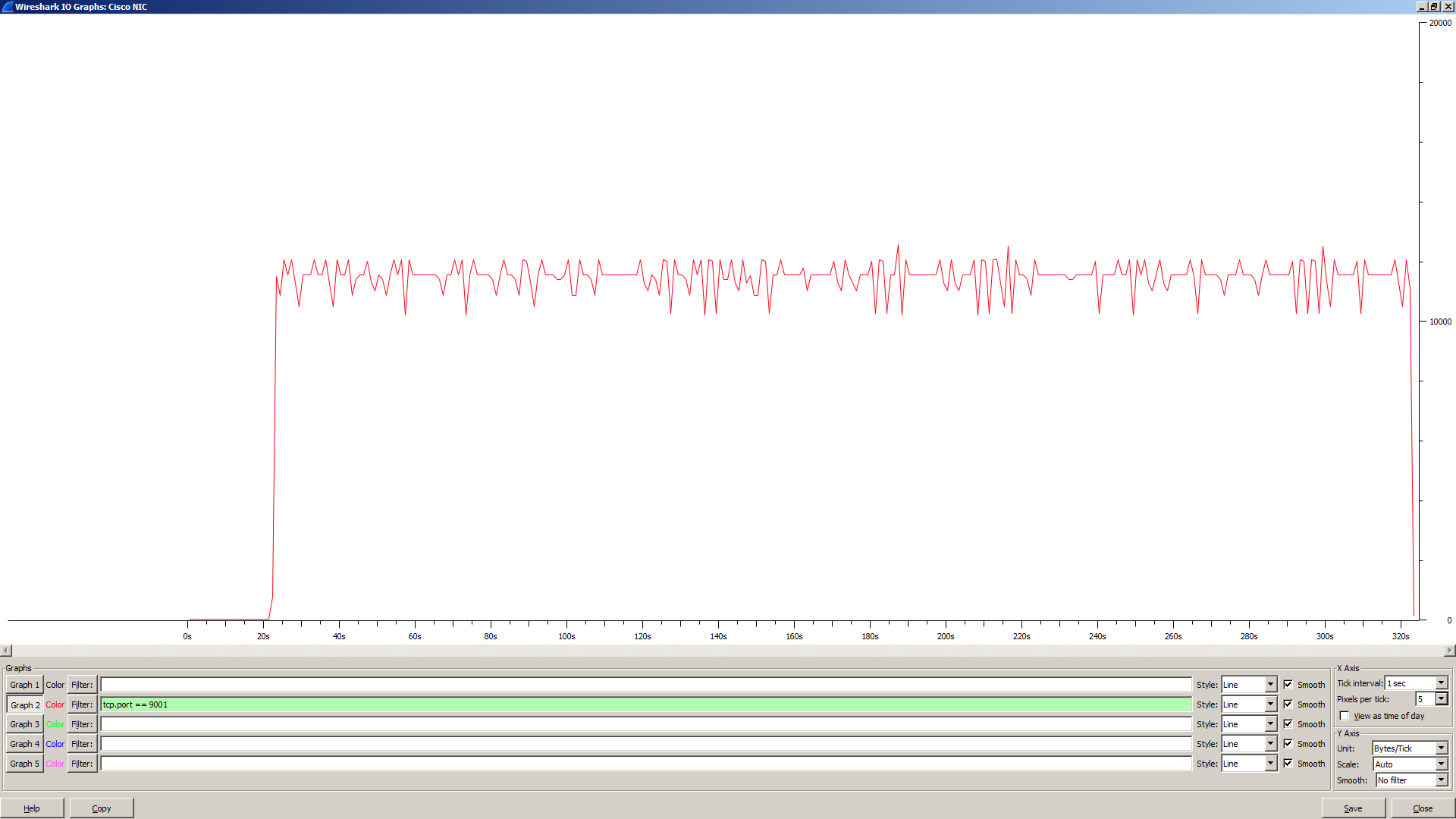
**Cvičenie 02 – Drozdík, Dočár**

Úlohou cvičenia bolo generovať rôzne typy tokov a sledovať ich správanie. Linku sme mali obmedzenú na 128 kbit/s a generovali sme pakety o veľkosti 512B, čo je 4096 bitov za sekundu. Cez linku nám teda prejde maximálne 128000/4096=31,25 paketov za sekundu.

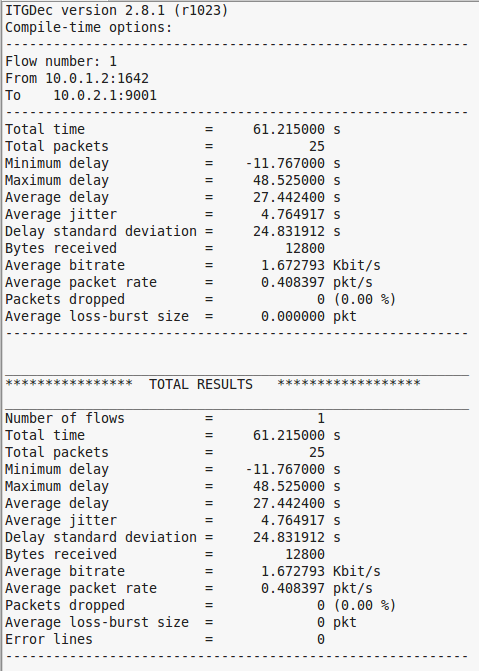
**Úloha 1 - Generovanie TCP toku (bez zahltenia)**

* generovali sme tok dlhý 5 minút a posielali sme 20paketov za sekundu o veľkosti 512B.
* výsledkom sú dva grafy (p/s a kbit/s) a logy.

*Pakety za sekundu*

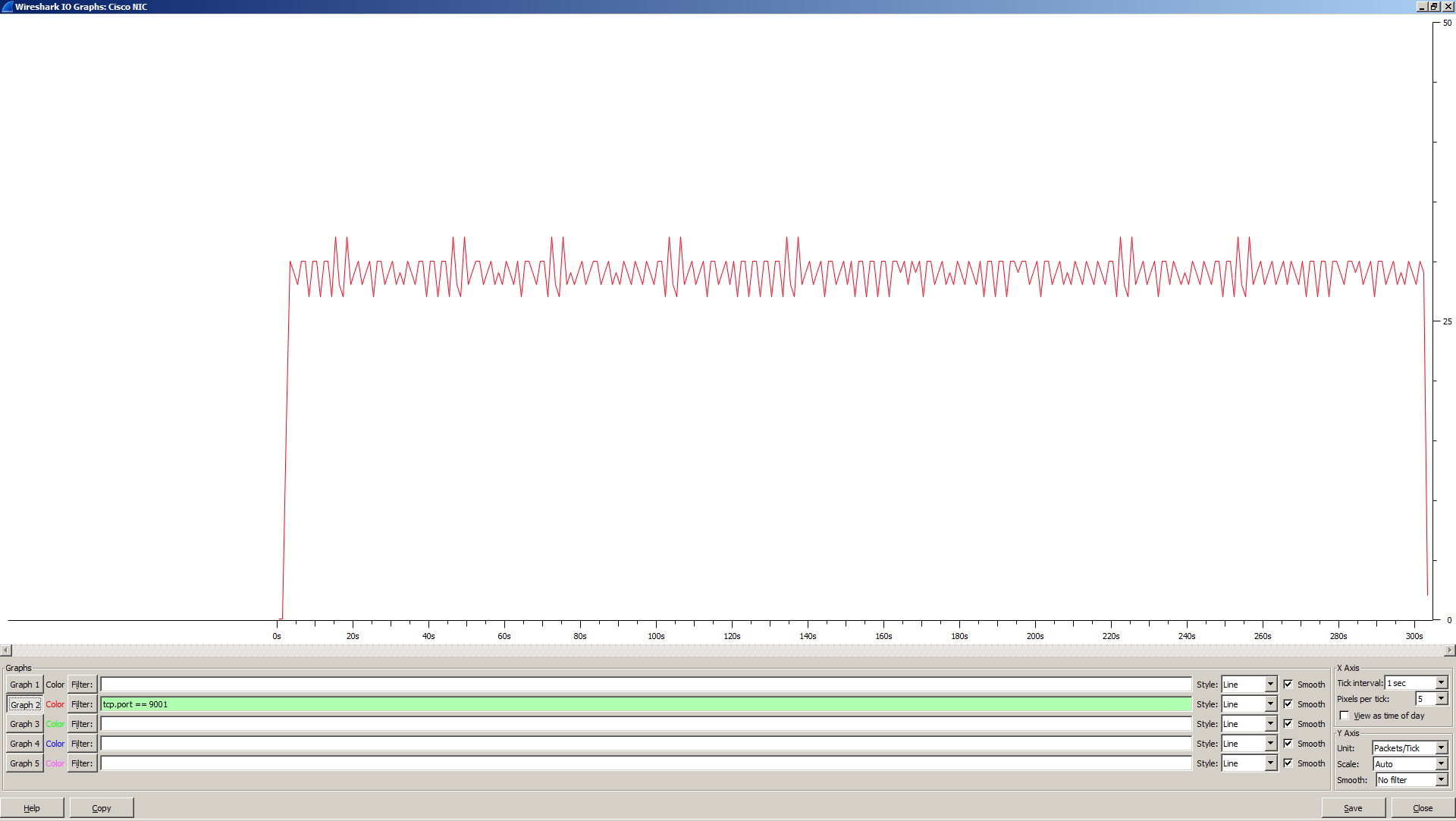
*Bity za sekundu*

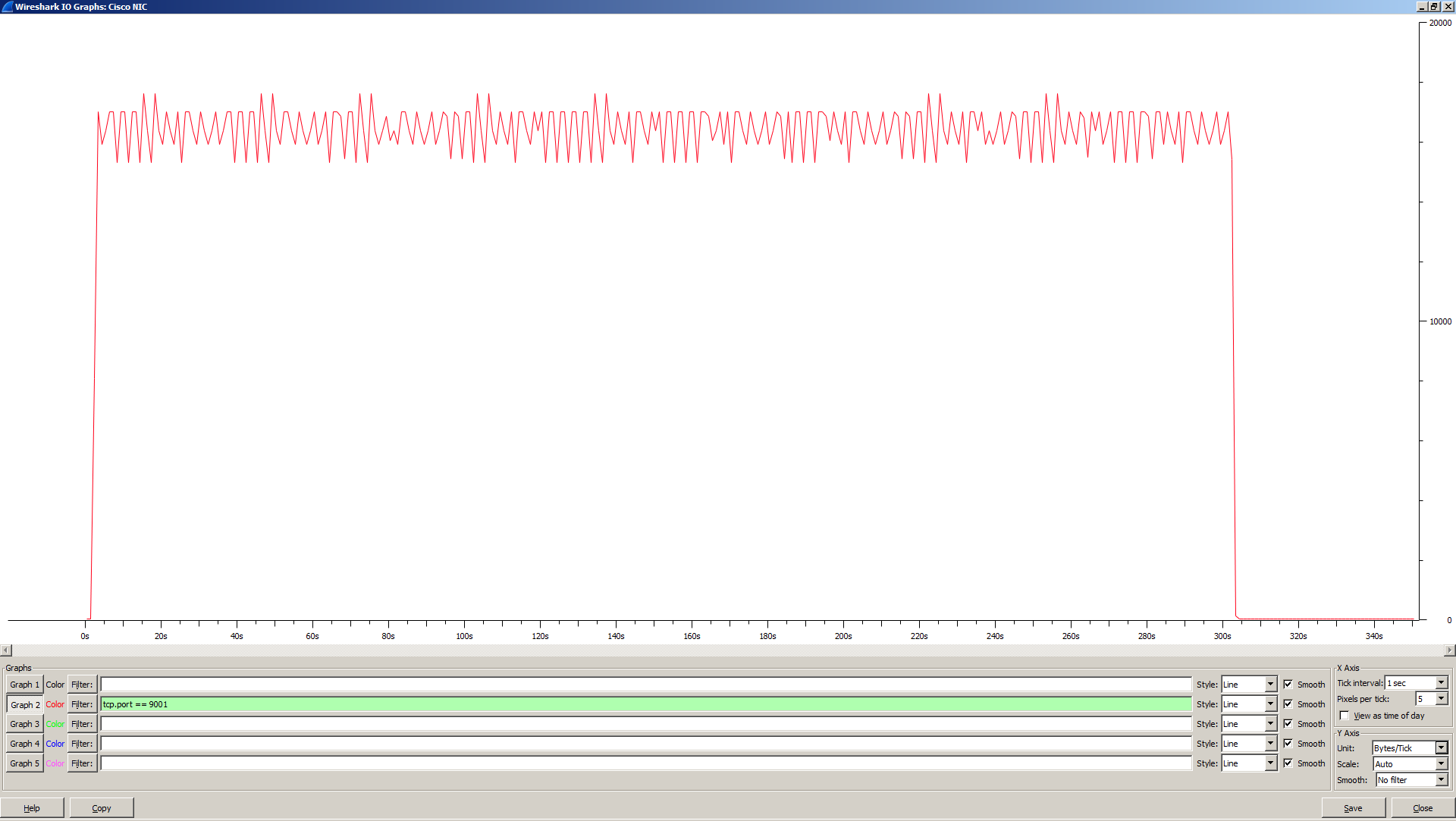
V logoch vidíme z časových dôvodov len tok dlhý 1 minútu:



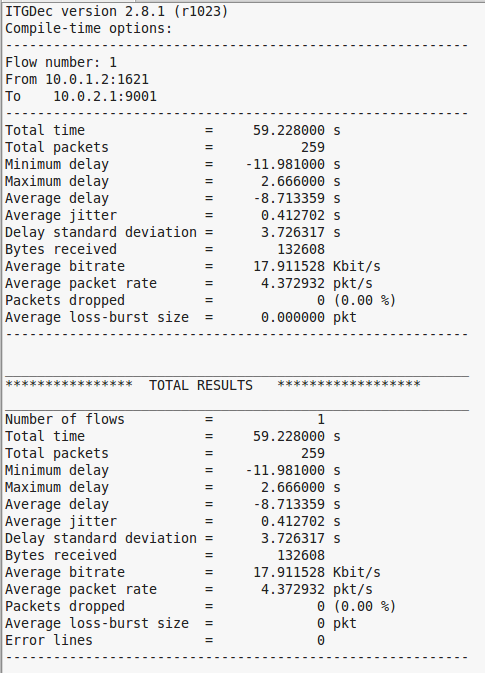
**Generovanie TCP toku (so zahltením)**

* generovali sme tok dlhý 5 minút a posielali sme 43 paketov za sekundu o veľkosti 512B.

*Pakety za sekundu*

*Bity za sekundu*

*Logy*



Vidíme že v druhom prípade, pri generovaní 43 paketov za sekundu sme dosiahli maximálnu kapacitu linky čo je vypočítaných 31 paketov. V logoch vidíme, ale aj pri testovaní pripojenia telnetom sme zistili, že výpadky nenastali, pretože TCP je spoľahlivý protokol.

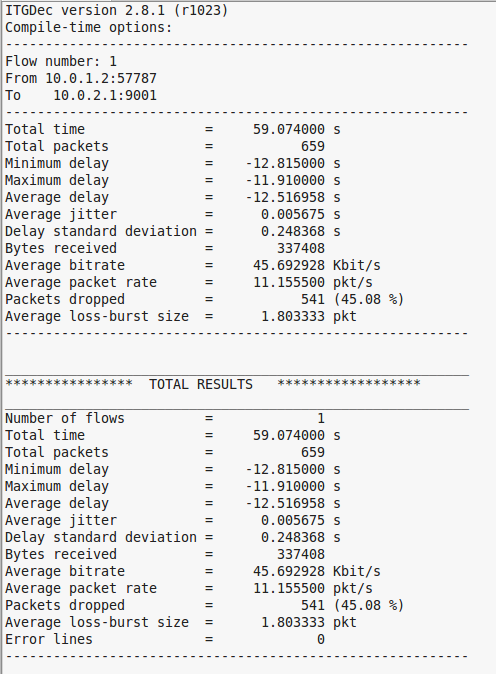
**Úloha 2 - Generovanie UDP toku (bez zahltenia)**

* generovali sme tok dlhý 5 minút a posielali sme 20paketov za sekundu o veľkosti 512B.

*Paketov za sekundu*

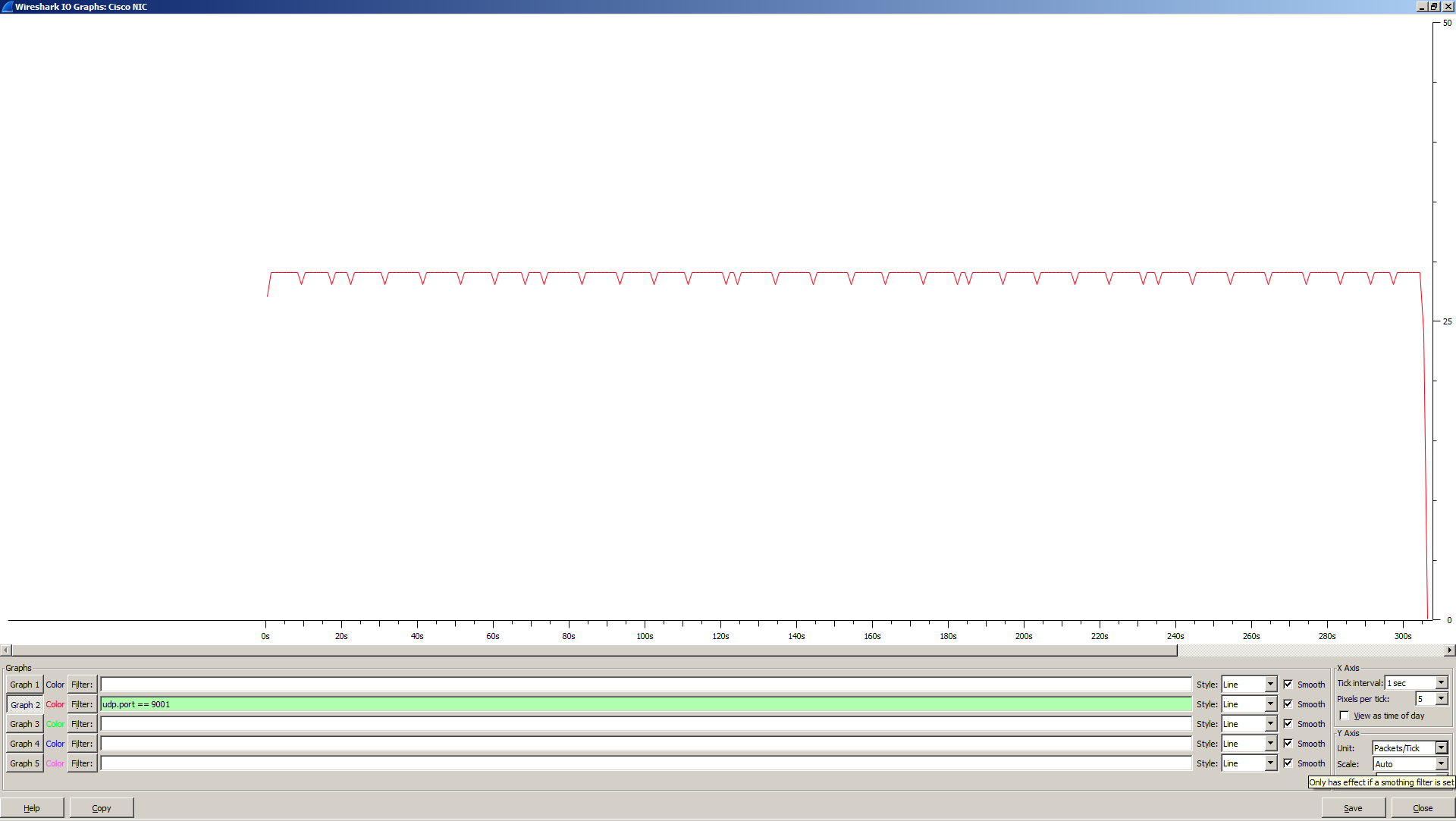
*Bitov za sekundu*

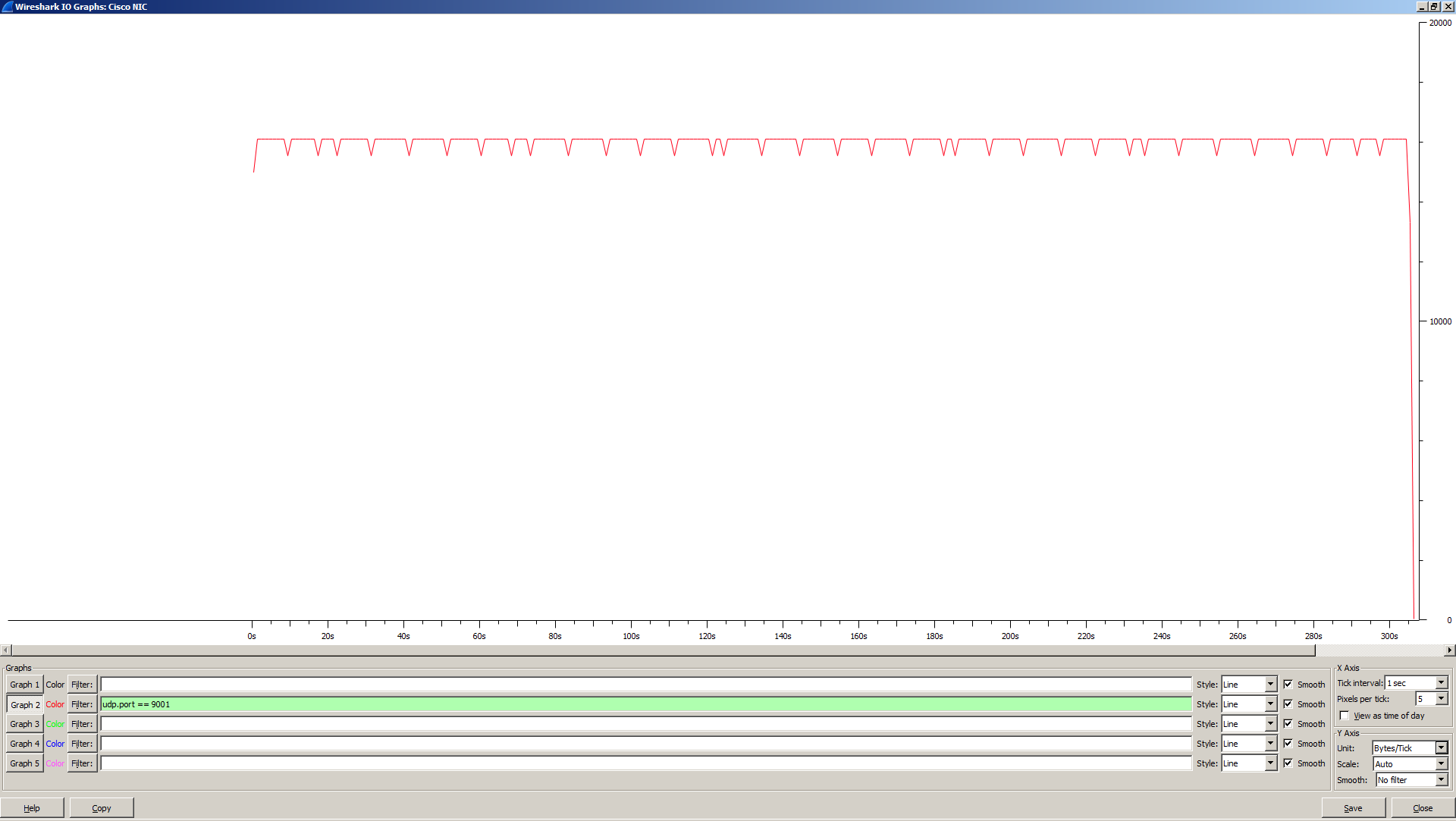
*Logy*



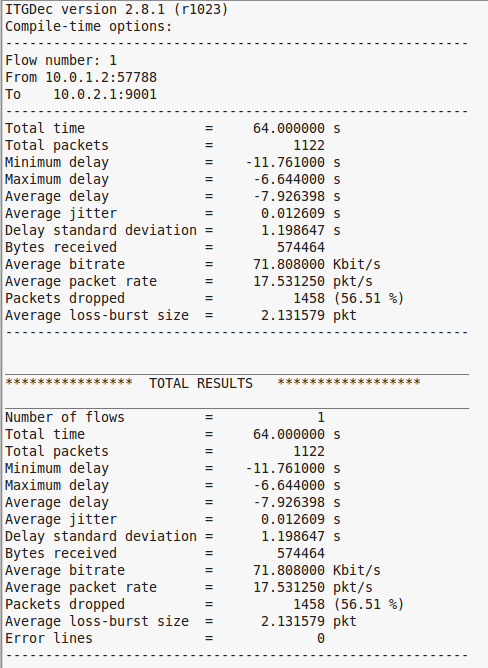
**Generovanie UDP toku (so zahltením)**

* generovali sme tok dlhý 5 minút a posielali sme 43paketov za sekundu o veľkosti 512B.

*Paketov za sekundu*

*Bitov za sekundu*

*Logy*



Vidíme že pri UDP už dochádza k strate paketov, a takisto aj telnet bol výrazne pomalý pri menšom zaťažení, pri väčšom nešiel vôbec.

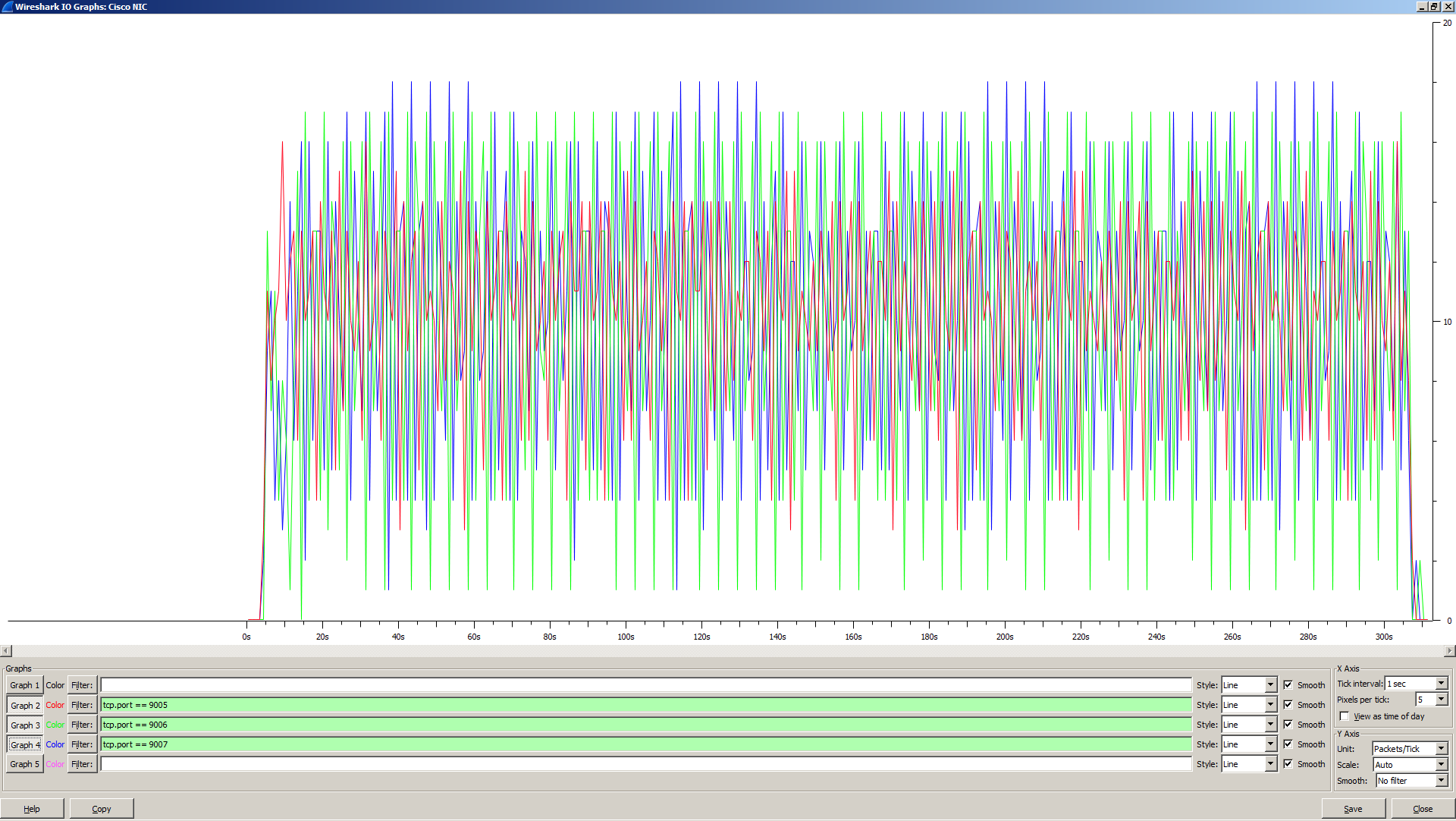
**Úloha 3**

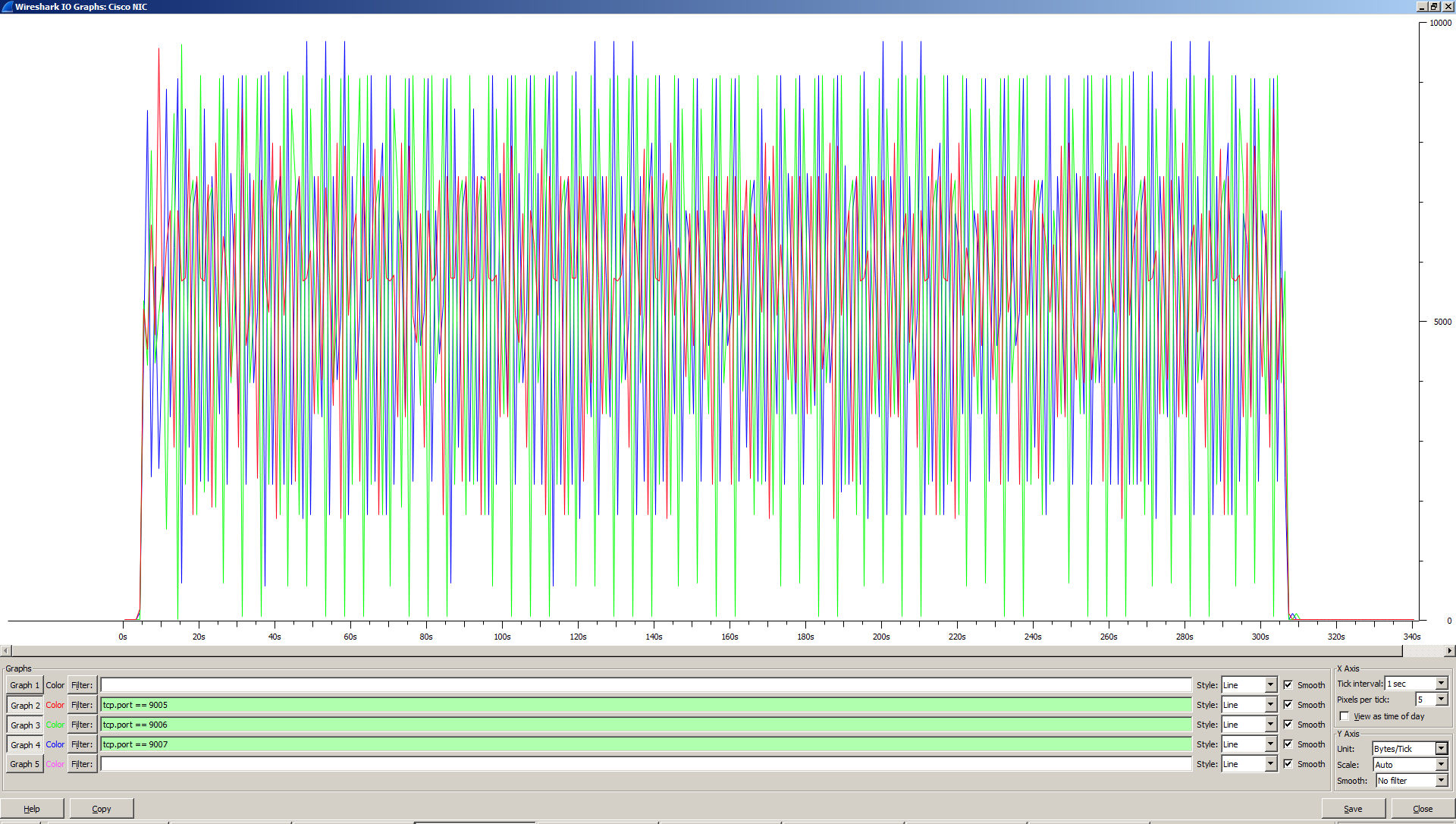
Ďalšou úlohou bolo otestovať správanie pri multiflow

* 3 TCP toky
* 2 toky, UDP a TCP
* 2 toky, UDP a TCP s fair-queue

**3 TCP toky**

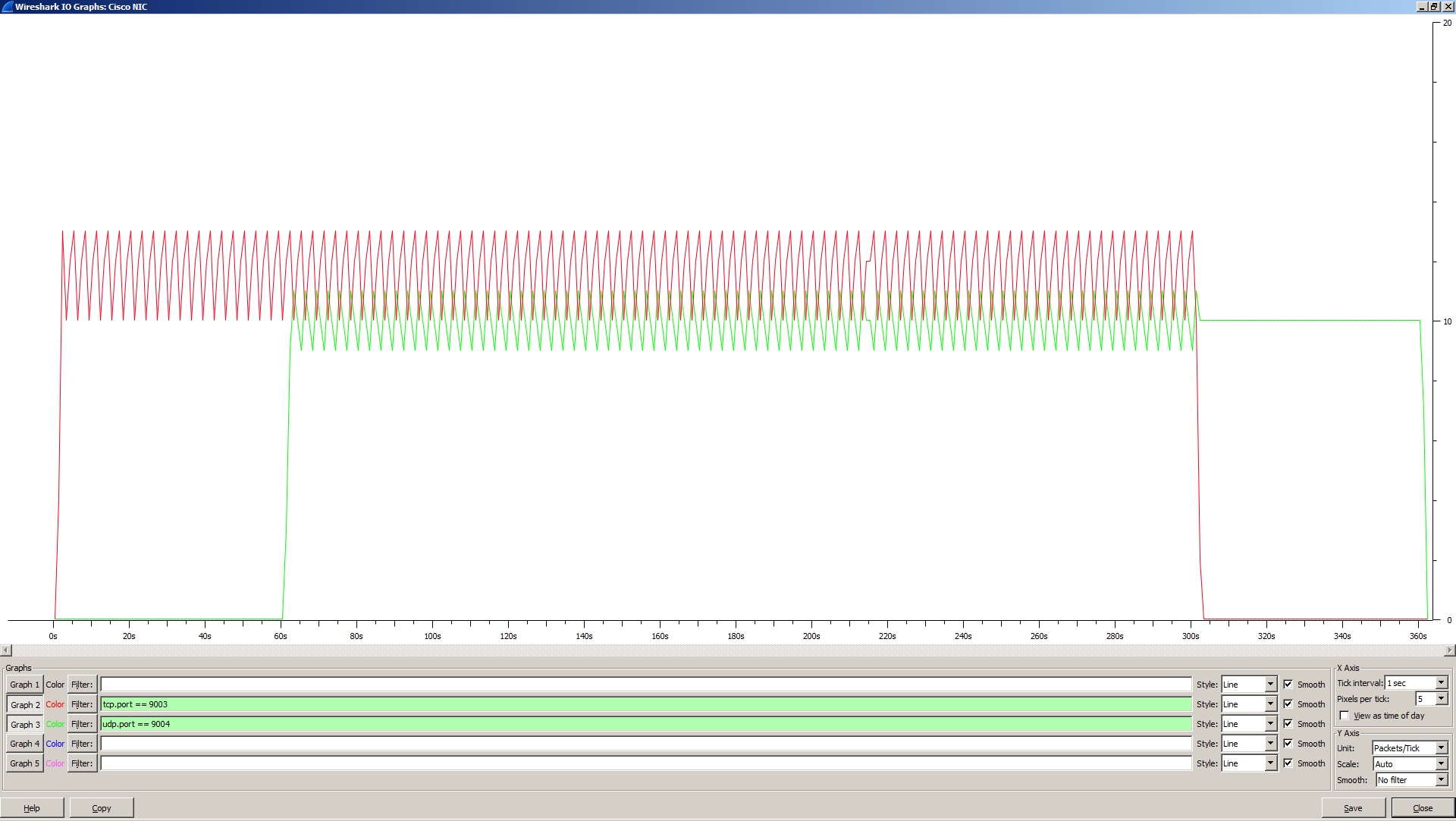
* generovali sme tok dlhý 5 minút a posielali sme 20 paketov za sekundu o veľkosti 512B z každého toku.

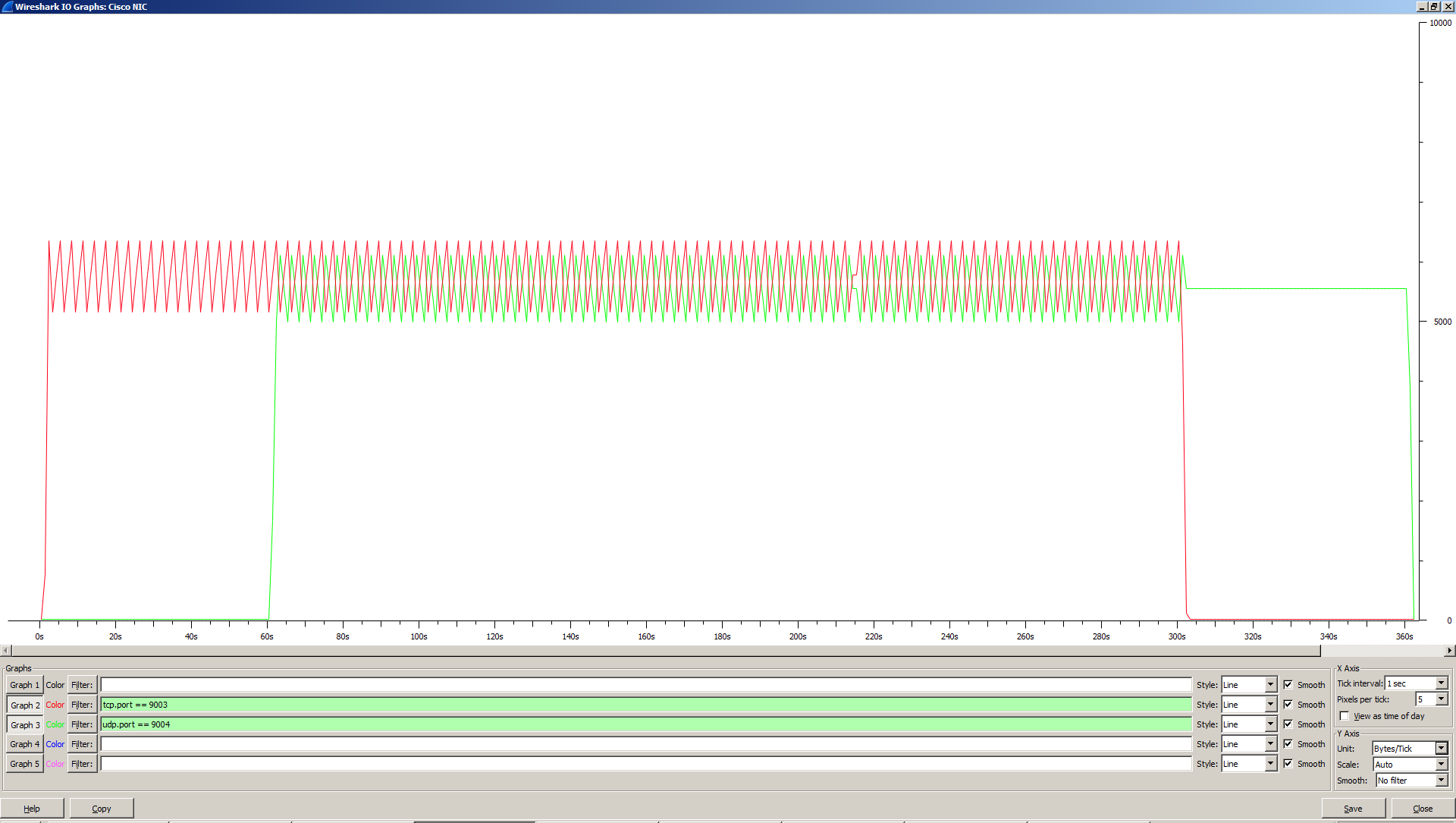
*Pakety za sekundu*

*Bity za sekundu*

**2 toky, UDP a TCP (bez zahltenia)**

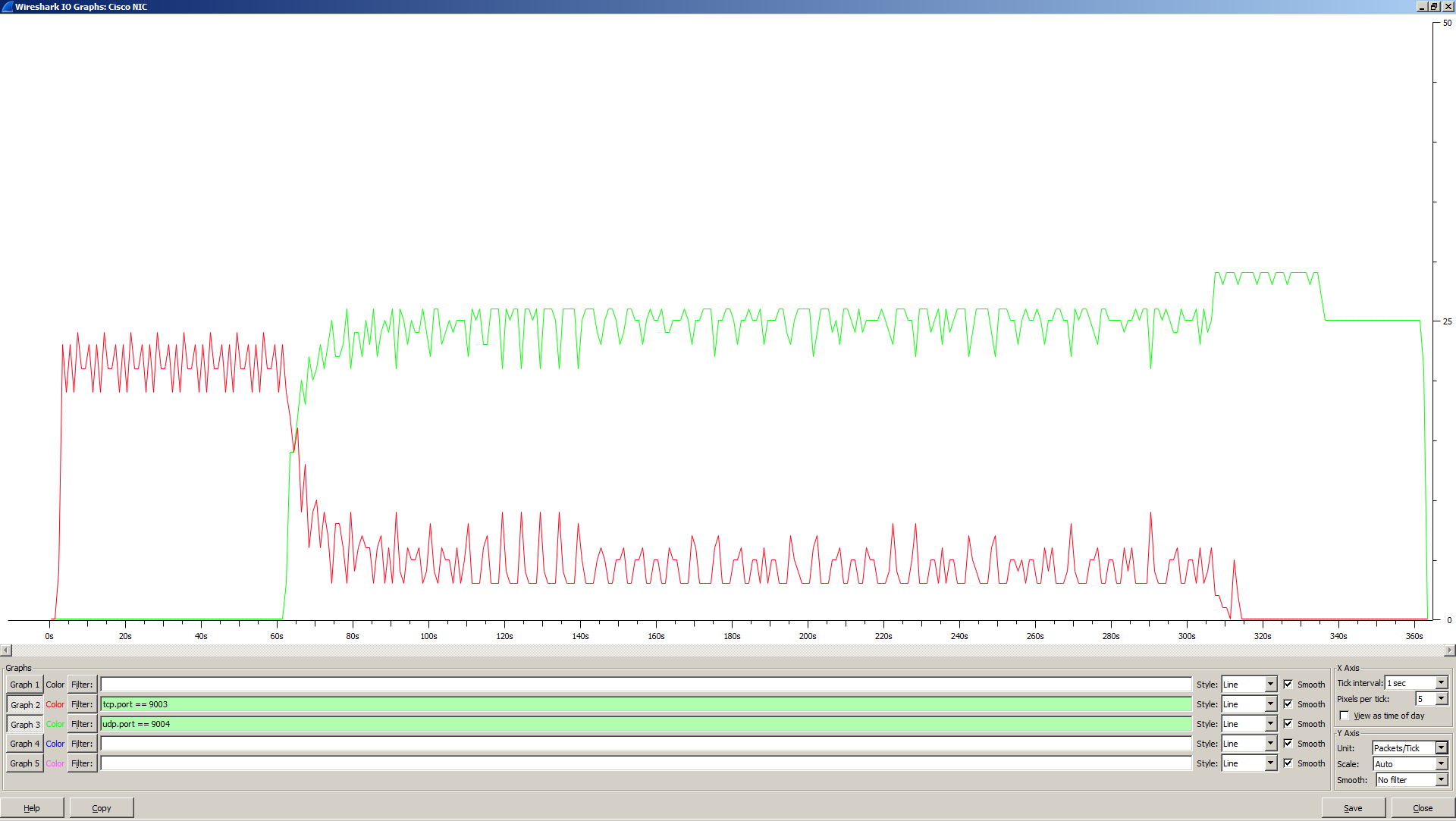
* generovali sme tok dlhý 5 minút a posielali sme 10 paketov za sekundu o veľkosti 512B z každého toku.

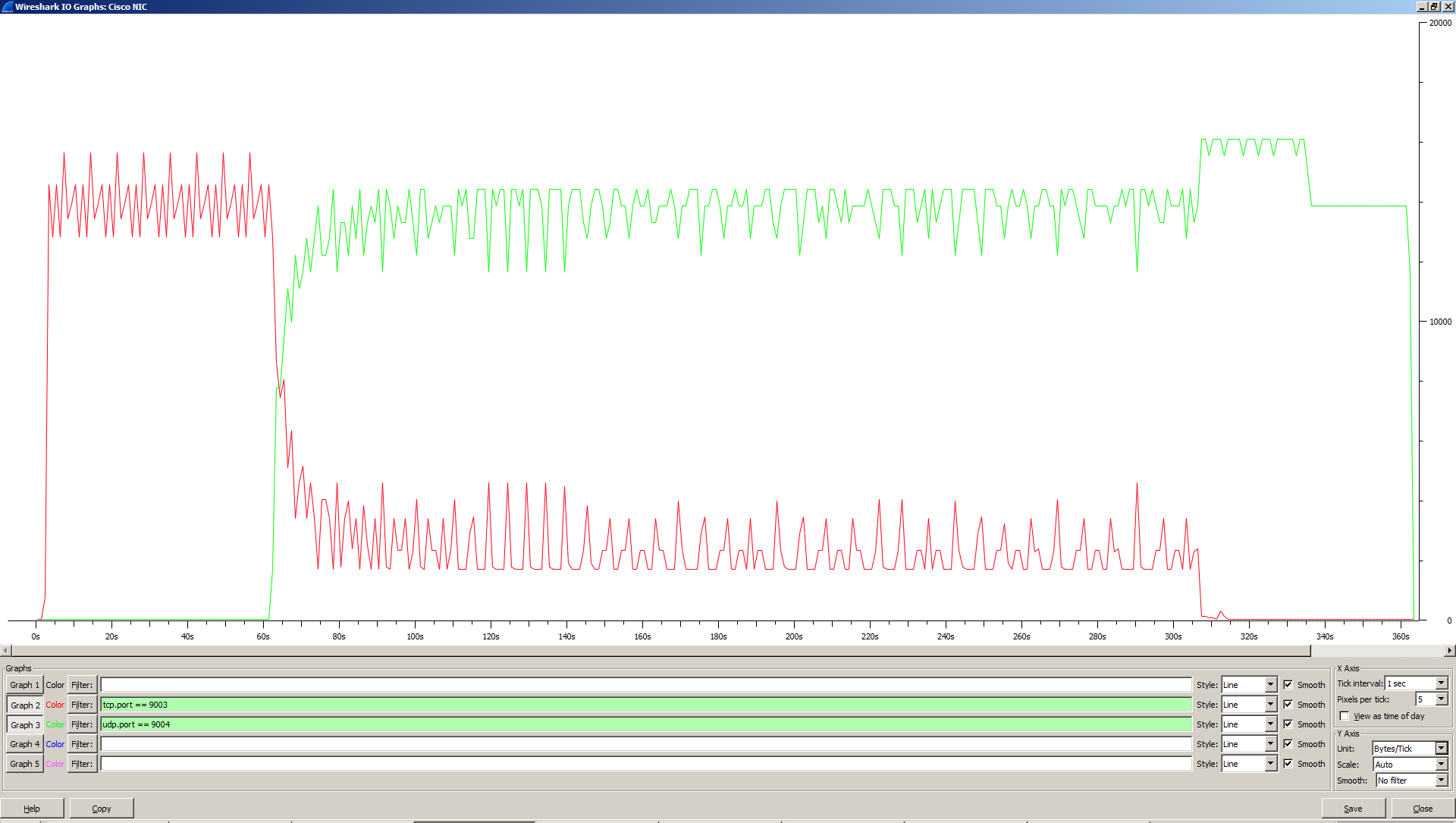
*Pakety za sekundu*

*Bity za sekundu*

**2 toky, UDP a TCP (so zahltením)**

* generovali sme tok dlhý 5 minút a posielali sme 25 paketov za sekundu o veľkosti 512B z každého toku.

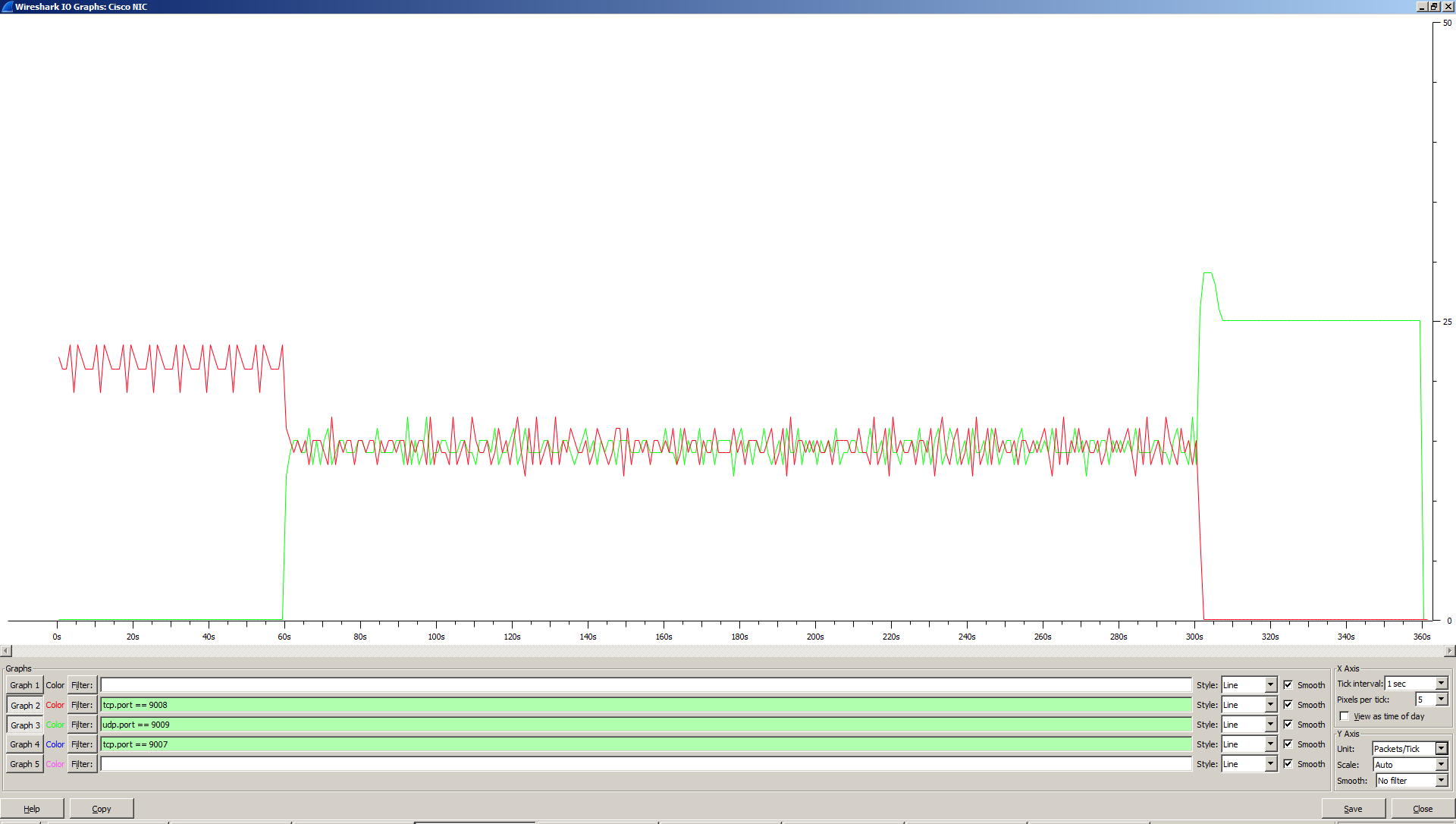
*Pakety za sekundu*

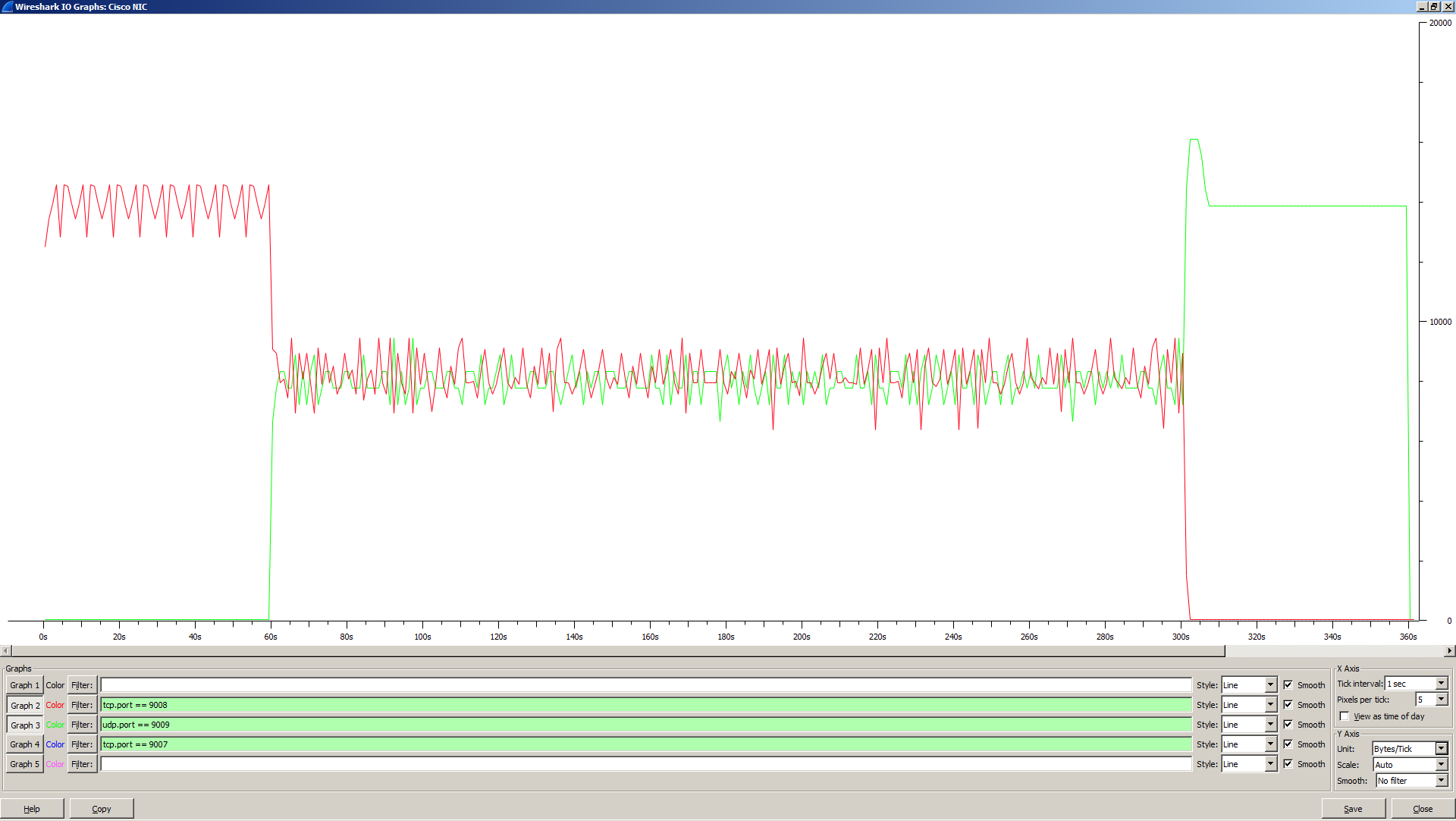
*Bity za sekundu*

Vidíme, že keď posielame dva toky, aj UDP aj TCP, tak pri väčšej kapacite ako je kapacita linky dochádza k zahlteniu takým spôsobom, že ako prvý odíde TCP protokol. Zatiaľ čo UDP posiela pakety bez kontroly, TCP sa ich snaží overovať.

Riešením je nasadenie fair-queue politiky na výstupné rozhranie pri generovaní paketov.

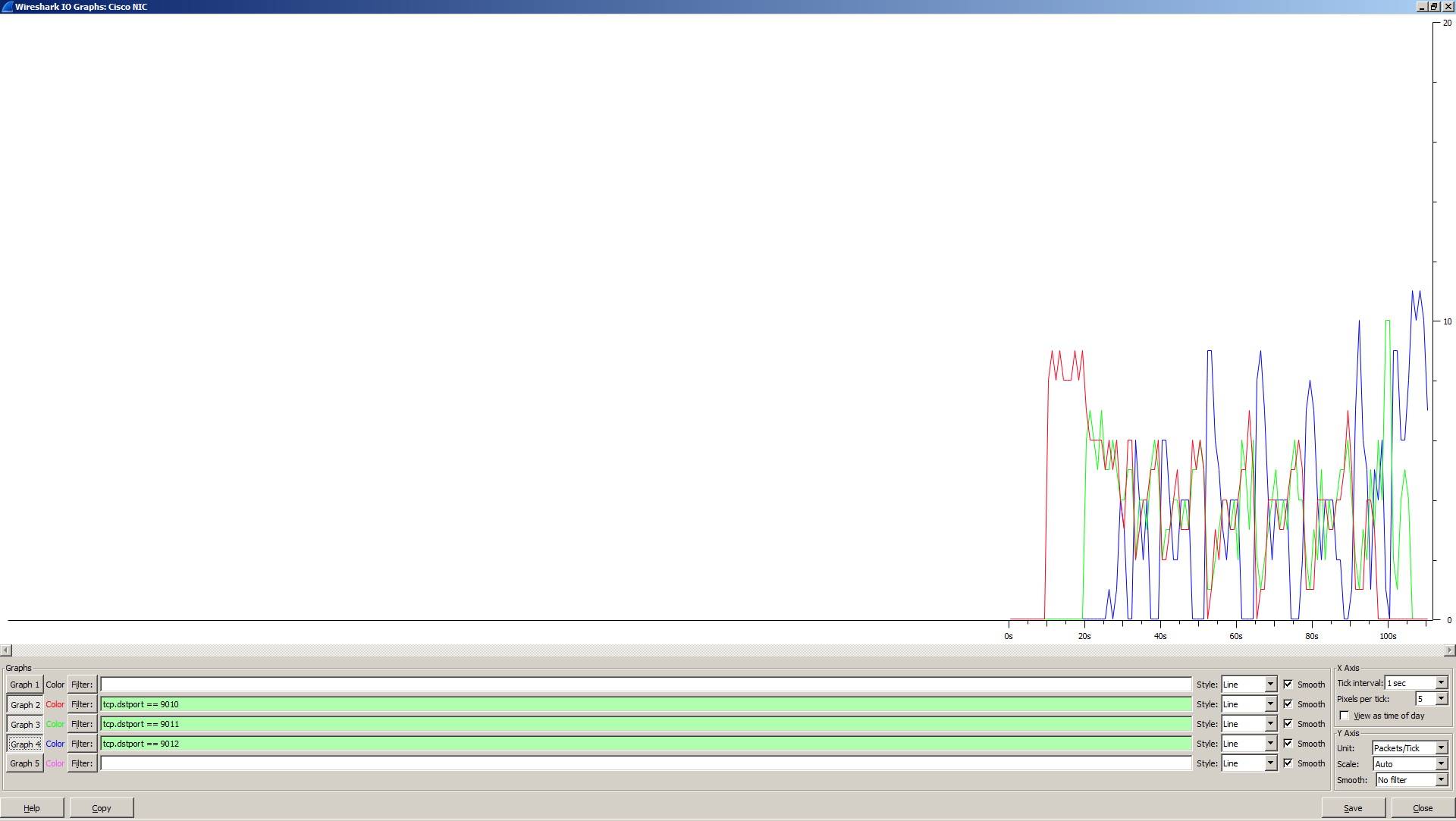
Po aplikovaní politiky môžeme vidieť nasledovné zmeny:

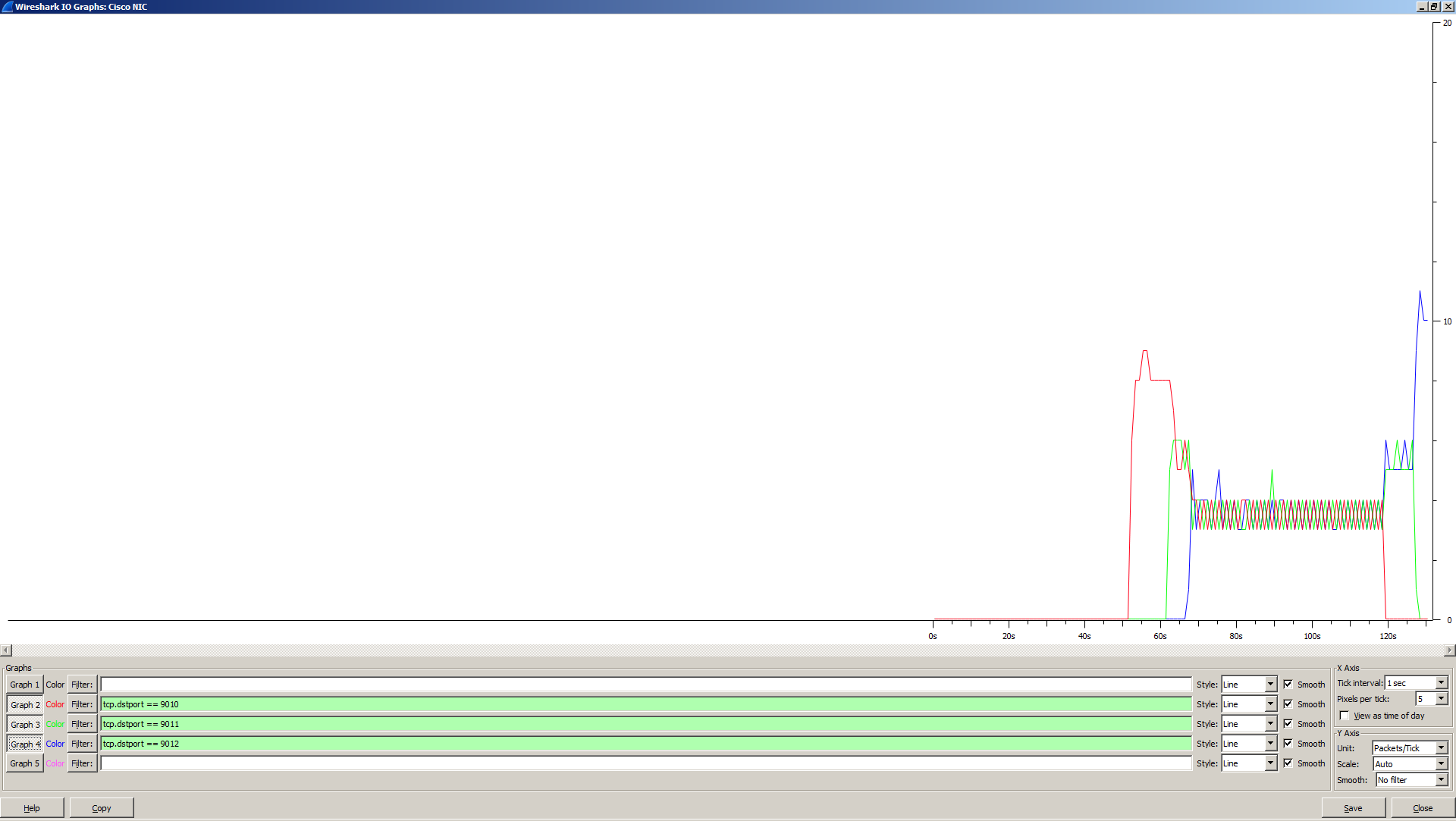
*Paketov za sekundu*

*Bitov za sekundu*

**Úloha 4**

* Cieľom úlohy 4 bolo značkovanie odchádzajúcich paketov v DSCP poli

*3 toky s rôznymi značkami*

*3 toky s rôznymi značkami a s politikou fair-queue*

*Ukážka tagu vo Wiresharku*