SZÉCHENYI ISTVÁN EGYETEM
GYŐR
TÁVKÖZLÉSI TANSZÉK

Mérési utasítás Wireshark megismerésének folytatása, TCP működésének vizsgálata

Az előző mérésen részben már megismert Wireshark programot fogjuk mai is használni. Ha valakinek szüksége van rá, akkor használhatja az előző mérési utasítást is. Nagyon fontos, hogy a cél NEM a feladatok gépies elvégzése, hanem az, hogy a hallgatók készség szintjén megtanulják a Wireshark, mint munkaeszköz használatát, amely a továbbiakban nélkülözhetetlen lesz. Ezenkívül természetesen a TCP működésének mélyebb megértése is a mai mérés fontos célja.

1. **feladat.**

Indítson egy csomagelkapást az eth0 interfészen, úgy hogy a leállítás feltétele legyen 1 perc, valamint a képernyő automatikusan gördüljön a csomagokkal. (A 2. gomb segítségével nyissa meg *Capture Options* dialógus ablakot, és ott a „Stop Capture Automatically After…” felirat alatt helyezzen el egy pipát az „1 minutes” elé, valamint ellenőrizze, hogy van-e pipa a „Display Options” felirat alatt található „Automaticall scroll during live capture” előtt.) Majd a böngészőt elindítva kérje le a *www.hirado.hu* honlapot!

A Wireshark az elkapott csomagok sorszámát, a forrás és cél IP-címet, a protokoll nevét valamint a csomag részletét jeleníti meg első látásra. Alul látható, hogy a Wireshark a különböző protokollokat sorrendbe helyezi. Először a csomag méretét adja meg, majd az Ethernet opciókat. Itt található a forrás és cél MAC cím. Alant az IP protokoll adatai láthatók, mint a forrás és cél IP-cím. Majd végezetül a TCP tulajdonságokat nézhetjük meg. Mint például a forrás és cél port, valamint a különböző TCP bitek értékét (SYN, ACK, FIN stb.).

Jól megfigyelhető, hogy először a mi gépünk lekéri a DNS bejegyzést a névkiszolgálótól, majd megkezdi IP cím alapján a *www.hirado.hu* oldalt letölteni.

Ezenkívül valószínűleg sok egyéb, a feladat szempontjából érdektelen forgalmat is megfigyelhet. Ezek kiszűrésére használhatók a csomagszűrők.

2. **feladat**

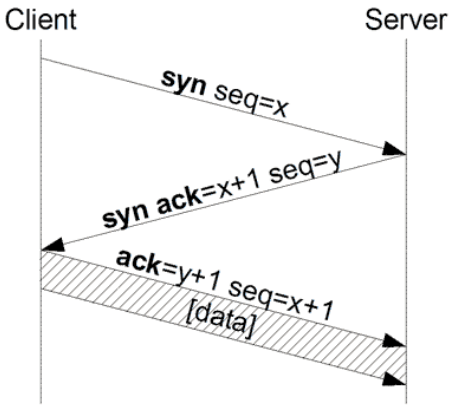
Hajtsa végre az előző feladatot úgy, hogy most a Wireshark csak a 80-as TCP portot érintő csomagokat kapja el. (A 2. gomb segítségével nyissa meg a *Capture Options* dialógus ablakot, és ott *Capture Filter*ként állítsa be, hogy: *tcp port 80*. Majd indítsa el a csomagelkapást, és töltse le az oldalt).

Mit tapasztalt?

3. **feladat**

Hajtsuk végre az előző feladatot úgy, hogy a csomagelkapás leállításának feltétele 3 csomag elkapása legyen. (Szűrjünk továbbra is a 80-as TCP portot érintő csomagokra.) Ezzel az előző feladatból csak a „three way handshake”-et vagyis a 3 utas kézfogást kaptuk meg.

Ez a TCP protokoll kapcsolat felépítési fázisa.



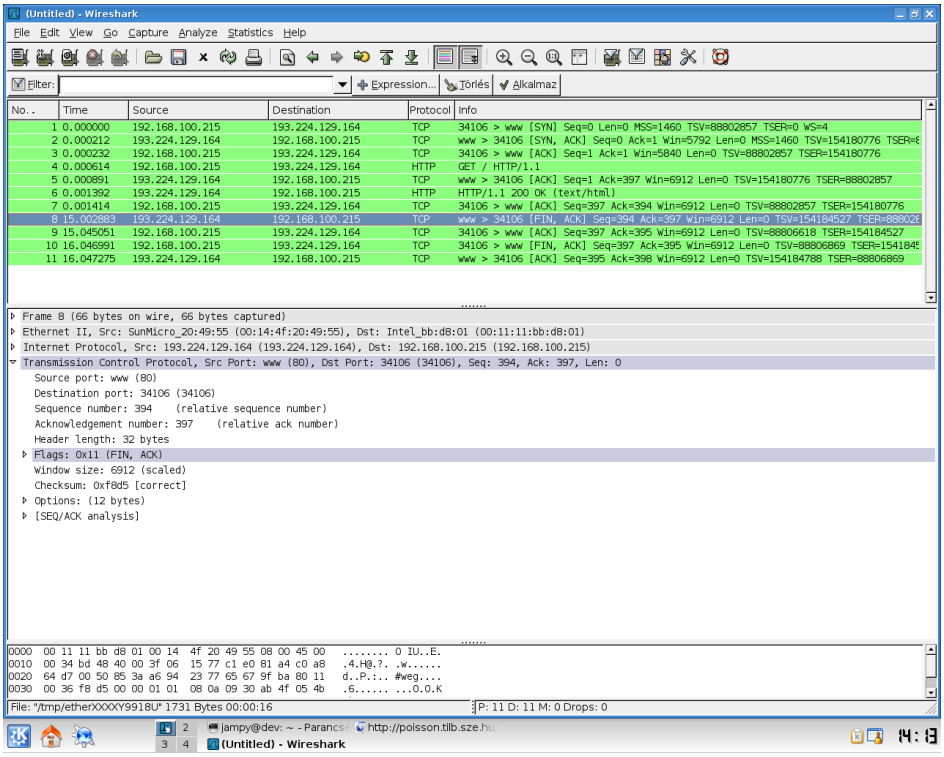
A csomagokat „kibontva” látható, hogy a 3 utas kézfogás a klienstől a szerver fele küldött TCP SYN szegmenssel, benne Sequence number=0-val kezdődik, majd a szerver válasza a TCP SYN,ACK szegmens, benne Sequence number=0 és Acknowledge number=1, majd ismét a kliens válaszol egy TCP ACK bittel, ahol mind a Sequence number mind az Acknowledge number 1-re van állítva. Figyelje meg a Sequence Number és az Acknowledge number értékét az ablak alsó részén is, ahol a Wireshark a csomag valódi tartalmát hexadecimális formában jeleníti meg: ott ezek értéke ott NEM 0! Magyarázat: a Wireshark az ott látható kezdőértékekhez képest vett relatív értéket jelenít meg azért, hogy a felhasználó számára az értelmezést segítse.

Ezzel létrejött a TCP kapcsolat.

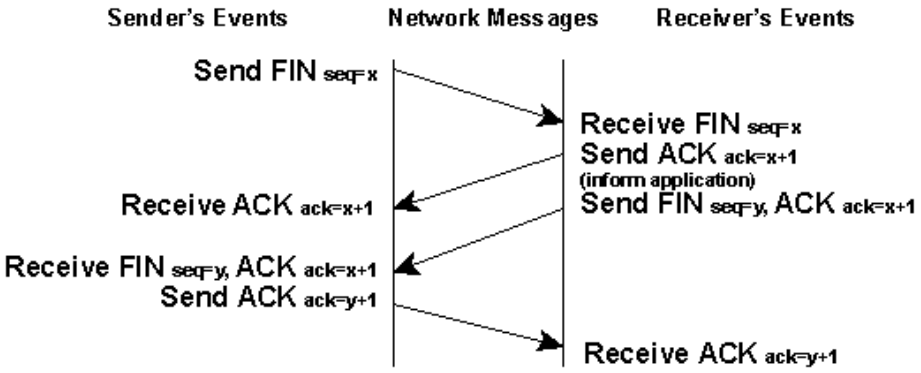
Figyeljük meg a TCP Options mezőjét azokban a csomagokban, melyekben a SYN bit be van állítva. Ezekben a csomagokban a két kommunikáló fél engedélyezi egymás között a SACK (Selective Acknowledgment) opciót. Mire jó ez? Amennyiben csak egy csomag veszik el, nem tudjuk közöli a küldővel, hogy csak azt az egyet küldje újra hiszem az ACK egy előre meghatározott pontig nyugtáz. Erre való az SACK, mely segítségével megadhatjuk, hogy melyek azok a csomagok melyek megjöttek a ténylegesen várt nyugta pontjáig, és melyik/melyek azok melyek nem érkeztek meg.

4. feladat

Hajtsuk végre az előző feladatot úgy, hogy vegyük ki a csomagelkapás leállítási feltételt, és most a http://dev.tilb.sze.hu lapot kérjük le. (Az egyszerűség kedvéért.)



Itt az utolsó négy csomagban megfigyelhető a 4 utas kézfogás, mely a TCP kapcsolat lebontását jelenti. Először a szerver küld egy FIN bitet amelyre mi ACK bittel válaszolunk. Majd mi is küldünk egy FIN bitet, amelyre a szerver válaszol ACK-kal.



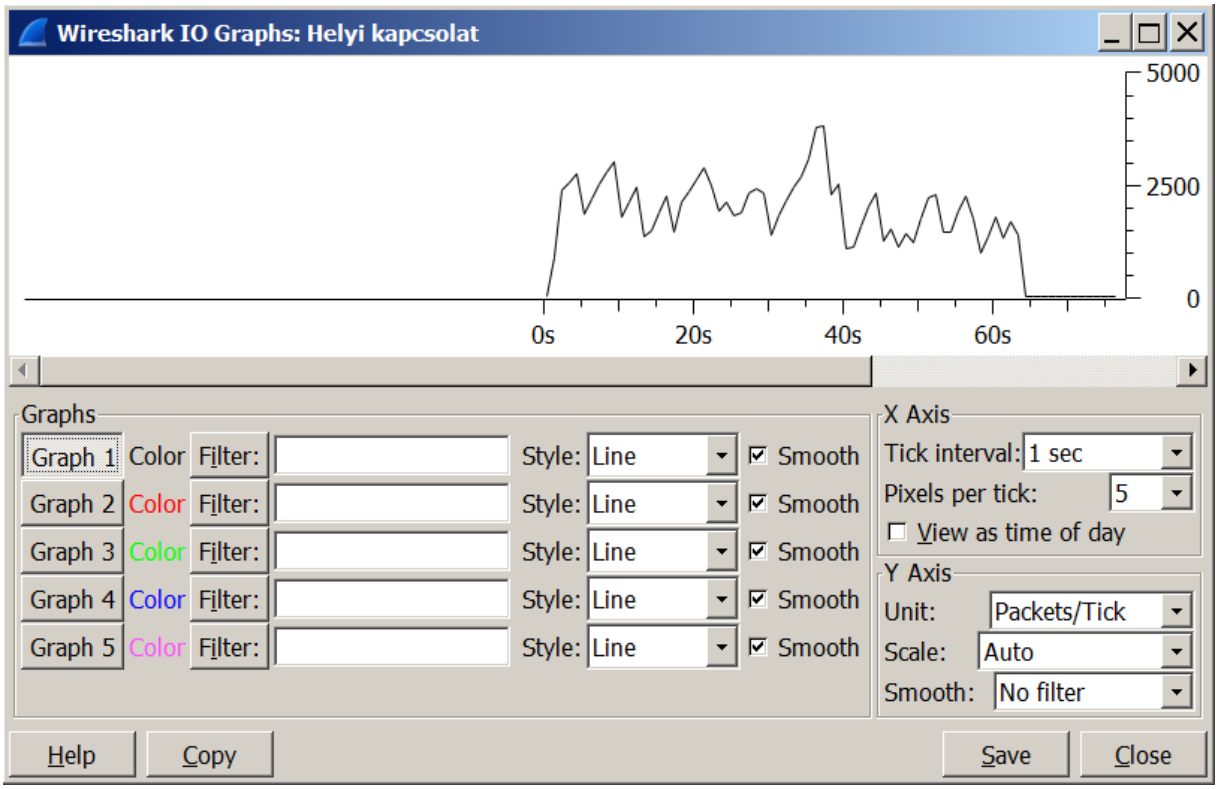
Mivel a hálózaton (feltehetőleg) semmilyen torlódásra utaló jelet nem tapasztaltunk (és a TCP protokoll sem) így a TCP FLAGS opciók között a CWR, vagyis az a bit mely jelzi, hogy torlódás miatt csökkentettük az átküldhető adatok mennyiségét, nem aktív.

5. feladat

A TCP torlódásvezérlés (congestion control) működése közvetlenül nem figyelhető meg Wireshark segítségével, mivel a torlódási ablak értékét mindkét fél helyben kezeli (és nem küldi át a másik félnek). Az algoritmus működésének a hatása azonban megfigyelhető az átviteli sebesség változásában. Annak érdekében, hogy érdemi megfigyelést végezhessünk, egy távoli szerverről fogunk nagyméretű fájlt letölteni.

Először nyissuk meg egy böngészőben a http://whale.hit.bme.hu/TCP/ oldalt, aztán indítsunk egy Wireshark csomagelkapást, majd a böngészőben kattintsunk rá a 100MB nevű fájlra. Várjuk meg, amíg a fájl teljesen letöltődik, majd állítsuk le a csomagelkapást. (Amennyiben a letöltés 1 percnél tovább tartana, akkor kb. 1 perc után leállíthatjuk.)

Jelenítsük meg a Wireshark *Satistics* menüjéből az *IO Graph*-ot. Ezen a másodpercentként átvitt csomagok száma látható az idő függvényében:



Figyeljük meg, hogy a grafikon egyáltalán NEM olyan, amit vártunk. Az AIMD (Adaptive Increase / Multiplicative Decrease) algoritmus működésének megfelelően lineáris növekedést, majd hirtelen visszaesést vártunk, de a fent látható „fűrészfog” mintázat emelkedési és esési meredeksége meglehetősen hasonló. Mi lehet ennek az oka? Tanulmányozza önállóan a megjelenítési dialógusablak beállításait, és próbálkozzon azok megváltozatásával!

A következő oldalra csak akkor menjen tovább, ha 3 perc alatt sem sikerült rájönnie a megoldásra. (Súgás: milyen időállandóval kell működnie a torlódás elkerülésének? – És a megjelenítésnek?)

Állítsa be a következőket:

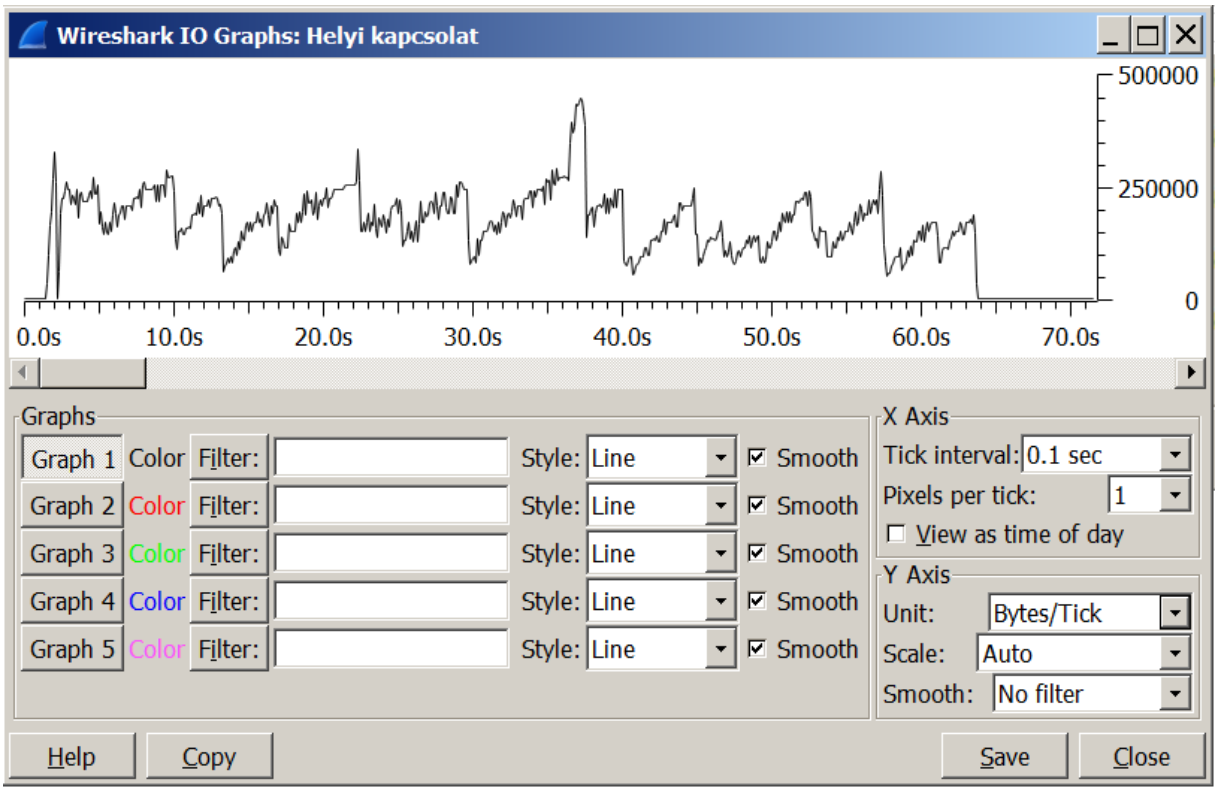
X Axis:

- Tick interval: 0.1 sec (Ez lényeg: az átlagolási időablak ne legyen túl nagy!)
- Pixels per tick: 1 (Csak a fenti beállítás kompenzálására, hogy elférjen a grafikon)

Y Axis:

- Unit: Bytes/tick (Ez érdekel bennünket. – Bár a szegmensméret valószínűleg állandó.)

A beállítások hatására a grafikon már egészen jól tükrözi az AIMD működését:



Jegyezze meg ennek a feladatnak a tanulságát, és szükség esetén alkalmazza a későbbiekben tanulmányai/munkája során a mérési eredményei kiértékelésekor!

Ha maradt még ideje, akkor végezze el a torlódásvezérlés vizsgálatát egy a Távközlés-informatika Laborban található szerver használatával is: http://dev.tilb.sze.hu/TCP/

Mit tapasztalt?