Hálózati kommunikáció monitorozása

Informatika 2- Hallgatói segédlet

1. Szükséges előismeretek

A gyakorlat elvégzéséhez az alábbi hálózati kommunikációs protokollok ismerete szükséges:

- ARP
- IP
- UDP, TCP
- DNS
- HTTP

2. A Wireshark program használata

A Wireshark (régi nevén Ethereal) protokoll analizátor program, amelyet a hálózat adminisztrátorok a hálózati hibák behatárolására, a forgalom analizálására használnak. A labor során ennek a programnak a segítségével megvizsgáljuk néhány hálózati protokoll pontos működését. Lehetőség nyílik arra, hogy az elméleti órákon tanultakat a gyakorlatban ellenőrizhessük.

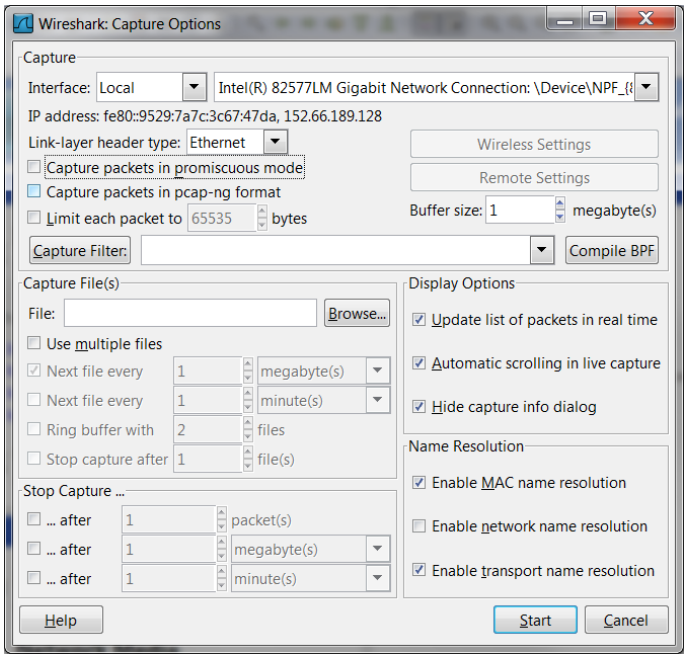
A labor során a Wireshark program funkcióinak csak egy részét használjuk ki, így az ismertetés során is csak ezekre a funkciókra koncentrálunk.

A program elindítása után az alkalmazás fő ablakát láthatjuk magunk előtt. Ezen ablak menüsorából elsősorban a **Capture** és az **Analyze** lenyíló menük elemeit használjuk a labor során. A **Capture** menü tartalmazza a hálózati csomagok monitorozásának beállításait és a monitorozás elindítását. Az **Analyze** menü a monitorozás során vett csomagok értelmezésének beállításait tartalmazza.

2.1. Egyszerű monitorozás

Amikor a számítógépünkön nem futattunk semmilyen hálózaton forgalmazó programot, akkor is kapunk folyamatosan hálózati csomagokat. Ezek broadcast csomagok, például ARP címlekérések, UPnP csomagok, stb. A monitorozás kipróbálása során ezeket a csomagokat fogjuk megvizsgálni.

A monitorozás elindításához válasszuk ki a **Capture** menü **Options** elemét. Ekkor a következő ablakot láthatjuk:



Wireshark monitorozási beállítások

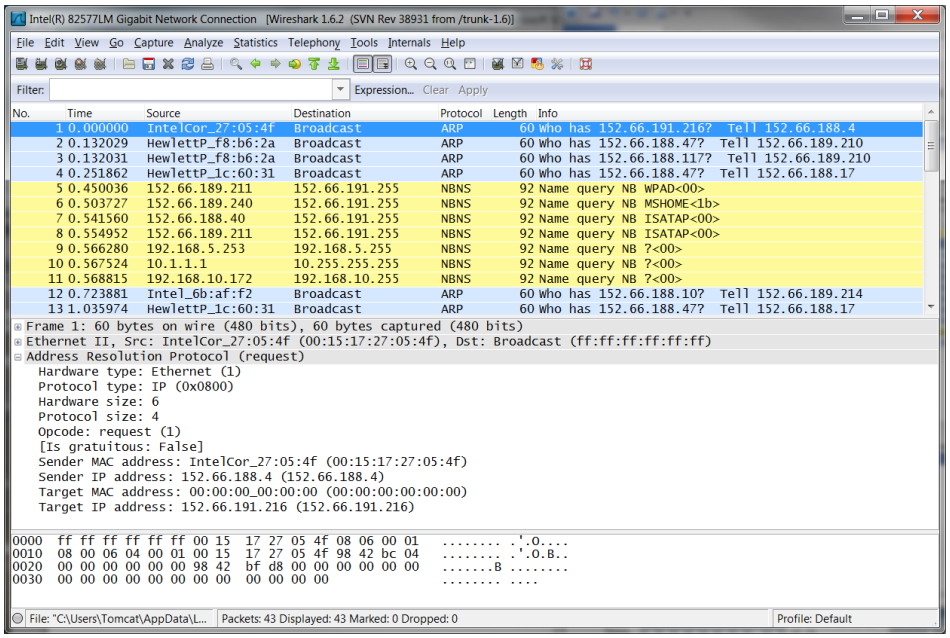
A dialógusablak **Interface** mezőjében megadhatjuk a vizsgálandó hálózati interfészt. Linux rendszerek esetén az interfészek elnevezése tipikusan ethN, ahol N az interfész indexe. Az MS Windows rendszerekben az interfészek elnevezése a hálózati vezérlő típusától függ.

A **Capture Filter** mezőben szűrési feltételeket állíthatunk be, amellyel a hálózati interfészen átvitt számos csomag közül kiválaszthatjuk azt a halmazt, amely számunkra érdekes. Helyes megválasztásával nagyban megkönnyíthetjük a munkánkat. Míg helytelen megválasztása jó pár hálózati csomag átböngészését eredményezheti.

Az esetek többségében csak ezt a két paramétert állítjuk, azonban érdekesek lehetnek a **Stop Capture** beállítások, amellyel a monitorozás végének feltételét adhatjuk meg, vagy a **Display Options** csoporton belüli **Update list of packets in real time** opció, amely jelentős terhelést jelenthet, azonban lehetővé teszi a beérkező csomagok folyamatos követését. Ezen kívül a **Name Resolution** csoportban található opciókkal megadhatjuk, hogy mely címeket és portokat jelenítsen meg szövegesen a program.

A paraméterek beállítása után a **Start** gombbal indíthatjuk a monitorozást. Amennyiben nem kapcsoltuk be a **Hide capture info dialog** opciót, akkor a megjelenő dialógus ablakon szám szerint követhetjük az elkapott csomagok számát. Amennyiben szeretnénk a monitorozást leállítani a **Stop** gombbal tehetjük meg. Ha kikapcsoltuk a dialógus ablak megjelenítését, akkor a **Capture** menü **Stop** elemével állíthatjuk le a monitorozást.

A monitorozás leállítása után a program főablaka a következő képen néz ki:



A wireshark program főablaka monitorozás után

A felület 3 részre tagolódik. A felső részben egy listát találhatunk, amely az elkapott csomagokat tartalmazza. Ha ebből a listából kiválasztunk egy csomagot, akkor az alsó részben láthatjuk a bináris tartalmát, míg a középső rész értelmezett formában mutatja. A vizsgálódásaink során elsősorban a középső részt használjuk.

A középső mező a kiválasztott csomag által tartalmazott keretek listáját mutatja. Ahogy a csomag az egyes hálózati rétegeken keresztül haladva felépül, minden réteg egy új keretet alkot belőle. Minden keret rendelkezik néhány fejléc mezővel és egy adat résszel. A lista elemeit lenyitva a fejléc mezők tartalmát tekinthetjük meg. Példaként a képen egy ARP kérés csomag felépítését láthatjuk. A csomagban található egy Ethernet keret, amely tartalmazza a forrás MAC címét, míg a cél MAC cím helyén egy broadcast cím található. Az Ethernet keret tartalmaz egy ARP keretet, amelynek mezői megadják a forrás MAC címet, IP címet, illetve a cél IP címét. A cél MAC címe a kérdés, amelyet a csomag próbál lekérni. Így annak értéke nem definiált.

Az előzőekben ismertetett módon bármikor megismételhetjük a monitorozást. Ilyenkor a program a monitorozás elindításakor felteszi a kérdést, hogy a korábban vett csomagokat lementse-e, vagy törölje. Döntsünk belátásunk szerint!

3. A TCP kommunikáció követése

A TCP protokoll egy adatfolyam jellegű kommunikációt tesz lehetővé. Ez a kommunikáció a két folyamat között, mint egy csővezeték funkcionál, amelyen mindkét irányba közlekedhetnek protokoll üzenetek vagy adatok. Ugyanakkor a hálózati modell alsóbb rétegeiben ez a kommunikáció is csomagokra tördelve jelenik meg, az üzenetek csomagok láncolatában továbbítódnak. Ha az előzőekben ismertetett módon egy TCP kommunikáció csomagjait monitorozzuk le, akkor azt az ablak felső részében, mint csomagok listája láthatjuk. Ezeket a csomagokat egyenként megvizsgálhatjuk, azonban ha az alkalmazási réteg protokoll üzeneteire vagyunk kíváncsiak, akkor célszerű a csomagokat összefűzni.

A TCP kommunikáció összefűzésére az **Analyze** menü **Follow TCP Stream** menüpontját használhatjuk. Ehhez ki kell választanunk a TCP kapcsolat egy csomagját, majd aktiválnunk a menüpontot. Ezt a menüpontot úgy is elérhetjük, ha az adott csomagot kiválasztva az egér jobb gombjával kattintunk.

A megjelenő ablakban olvashatjuk a két oldal üzeneteit. Az egyes irányokat színek különböztetik meg.

Megfigyelhetjük, hogy a menüpont kiválasztása módosítja az ablak tetején a **Filter** opció értékét. Ha újra az összes vett csomagot szeretnénk látni, akkor a **Clear** gombbal törölhetjük a szűrést.

4. Tájékoztató ellenőrző kérdések

Az alábbi kérdések megválaszolásával ellenőrizheti tudását. A gyakorlat beugrójában ilyen vagy ehhez hasonló kérdések várhatóak.

1. Hogyan rétegződnek a következő protokollok: UDP, IP, ICMP, TCP, Ethernet ?
2. Mi a hardware, más néven MAC cím, és mi a feladata?
3. Mi az ARP protokoll feladata?
4. Melyek az UDP és a TCP protokollok jellemzői?
5. Mi a DNS protokoll célja?
6. Hogyan működik a HTTP protokoll?
7. Hogyan valósul meg az IP kommunikáció két azonos alhálózatban lévő számítógép között?
8. Hogyan valósul meg az IP kommunikáció két különböző alhálózatban lévő számítógép között?
9. Mi a wireshark program feladata?
10. Hogyan vizsgálhatjuk meg egy hálózati csomag felépítését?
11. A TCP alapú kommunikáció esetén miért csomagokat látunk, és miért van szükség az összefűzésre?