

Network Monitor projekt – Felhasználói fejlesztési kérések összefoglalója

A beszélgetési napló alapján az alábbiakban összefoglaljuk Németh Gyula (PLBHOJ) felhasználó által a Network Monitor projekt során megfogalmazott főbb fejlesztési kéréseket. **Csak** az új funkciókra, működési logikára vagy dokumentációra irányuló kérések szerepelnek – a kizárólag korábbi hibás kód javítását célzó utasítások **nem** kerültek bele a listába.

Új funkciókra és fejlesztési logikára vonatkozó kérések

- **Letöltési és feltöltési sebesség mérése:** A felhasználó kérte egy funkció implementálását a hálózati letöltési és feltöltési sebesség mérésére[1]. Ennek megvalósítása egy tesztfájl le- és feltöltésével számítja ki a sebességeket, és megjeleníti az eredményeket a felületen (Download/Upload Mbps értékek).
- **Többszöri ping statisztikák (átlag, jitter, csomagvesztés):** Igény merült fel a ping alapú késleltetés részletes vizsgálatára. A felhasználó kérte, hogy a program több ping küldésével számolja ki a válaszidők átlagát, a jittert (ingadozást) és a csomagvesztési arányt[2]. Ezzel a Network Monitor átfogó képet ad a hálózati kapcsolat stabilitásáról és megbízhatóságáról.
- **HTTP válaszidő mérése:** A felhasználó kérte egy funkció hozzáadását a HTTP válaszidő mérésére is[3]. Ennek keretében a program egy megadott URL-re HTTP kérést indít, és méri, hogy mennyi idő alatt érkezik válasz (ez az **HTTP válaszidő**, ms-ban).
- **Valós idejű grafikon a sebesség- és ping-értékekről:** Felmerült az igény az adatok vizuális követésére, ezért a felhasználó egy valós idejű grafikon beépítését kérte[4]. A grafikon folyamatosan frissül a mért letöltési/feltöltési sebesség és ping értékekkel, így a felhasználó időben követheti ezen paraméterek változását a mérés során.
- **Traceroute funkció integrálása:** A hálózati útvonal elemzéséhez a felhasználó kérte egy traceroute eszköz beépítését[5]. A program a rendszer tracert parancsát futtatva megállapítja, milyen útvonalon jut el a csomag a célhostig, és lépésenként (hopról hopra) megjeleníti az eredményt a felhasználó számára.
- **Netstat adatok megjelenítése:** A felhasználó igényelte, hogy a program jelenítse meg az aktuális hálózati kapcsolatok listáját a netstat -ano parancs kimenetéhez hasonló módon[6]. Ezzel a Network Monitor felületén egy külön lapon láthatóak a folyamatban lévő hálózati kapcsolatok (portok, címek, folyamat azonosítók stb.), melyeket a program a rendszer **netstat** parancsának meghívásával gyűjt össze és jelenít meg.

- **LAN hálózat pásztázása (/24-es tartományban):** Felmerült az igény a helyi hálózat feltérképezésére, így a felhasználó kérte egy LAN scan funkció implementálását[7]. A megvalósított funkció a kiválasztott hálózati interfész IP-címe alapján végigpingeli annak /24 tartományát (általában az 1–254 címtartományt), és listázza, mely hálózati eszközök érhetők el (mely IP-címek válaszolnak).
- **UPnP port forwarding (porttovábbítás) kezelése:** A felhasználó szeretne volna, ha a program képes UPnP segítségével porttovábbítási szabályokat létrehozni és törölni az útválasztón[8]. Ennek megfelelően implementálásra került két művelet: **Port forward hozzáadása** és **Port forward törlése**, amelyek a szabványos UPnP AddPortMapping és DeletePortMapping hívásokat használják. A program automatikusan felderíti az UPnP-képes routert a hálózaton, majd SOAP üzenetekkel végrehajtja a kért portnyitást vagy -zárást.
- **Unicast, Broadcast, Multicast, Anycast tesztek külön grafikonon:** A felhasználó kérésére a Network Monitor kiegészült különféle csomagküldési tesztekkel is[9]. A program kis UDP csomagokat küld **unicast** (egy konkrét hostnak), **broadcast** (a helyi hálózat minden hostjának), **multicast** (egy megadott multicast csoportcímre) és **anycast** (egy globális anycast címre, pl. DNS szerver) módon, majd méri a válaszidőt. Az eredményeket egy külön grafikonon, eltérő színekkel jeleníti meg a program, így összehasonlíthatóak a különböző címzési módok válaszidejei.
- **Mérési eredmények exportálása JSON formátumba:** A felhasználó kérte, hogy a program támogassa a mért adatok exportálását is, hogy az eredmények későbbi elemzésre elmenthetők legyenek. Ennek nyomán elkészült egy **JSON export** funkció: a felhasználó egy gombnyomással ki tudja menteni az összes eddig gyűjtött mérési eredményt egy JSON fájlba[10] (a program egy listában gyűjti a méréseket, majd ezt a listát JSON formátumú fájlba írja ki).
- **Sötét mód (Dark Mode) bekapcsolási lehetőség:** Felhasználói kérés volt egy sötét téma támogatása a felülethez. A programba került egy „**Sötét téma**” jelölőnégyzet, mellyel át lehet váltani a felület színvilágát világosról sötétre[11]. Sötét mód bekapcsolásakor a hátterek sötétszürkére, a szövegek világos színűre változnak a kényelmesebb éjszakai használat érdekében.

Dokumentációs igény a programhoz

- **A program funkcióinak összefoglaló dokumentálása:** Végezetül a felhasználó kérést intézett a generatív MI-hez egy rövid dokumentáció vagy összegzés elkészítésére is, amely bemutatja a Network Monitor program összes funkcióját és használatának lényeges lépéseit. Ennek eredményeként készült el egy áttekintő leírás, amely pontokba szedve sorolja fel a megvalósított funkciókat[12], így bemutatva, hogy a ChatGPT hogyan segítette a fejlesztés során az új ötletek implementálását és azok magyarázatát.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] NetworkMonitor.java

file:///file_00000000c34471f5a67ad41ac71fba59