Sip2Peer bemutatkozás

P2P alkalmazások a gyakorlatba

# Bevezetés

Az első mérés alkalmával a Java és Android platformon is elérhet sip2peer könyvtárral fognak megismerkedni, a rendelkezésre álló minta-alkalmazást fogják kiegészíteni egy pont-pont üzenetküldést megvalósító alkalmazássá.

A minta-alkalmazás segítségéve egy gépen több csomópontot is létrehozhat. Az alkalmazás futtatásakor egy ablakot kap, melyen az új csomópontokat konfigurálhatja. Minden új csomópont saját felületet hoz létre. A felület kapcsolatok létrehozását és a kapcsolatokkal való üzenetváltást támogatja.

A mérés célja a minta alkalmazás felülete mögötti üzenetváltás megvalósítása a sip2peer könyvtár segítségével.

# Sip2Peer

Kevés p2p kommunikációt megvalósító csomag érhető el Java környezetben. A legelterjedtebb a komplex, több platform közti kommunikációt megvalósító Java JXTA implementáció, a JXSE. A JXTA protokoll egy teljes P2P stack, tartalmaz DHT megvalósítást, támogatja a NAT átjárást, saját szolgáltatás-leító nyelvet és szolgáltatások hírdetését, útválasztást, pont-pont és csoportos üzenetküldést... Sajnos a rendszer használata nehézkes, az Android port a PeerDroid rosszul dokumentált és a fejlesztését és támogatását leállították.

A skála másik oldalán a kutatási célokra fejlesztett megoldások szerepelnek, melyek jellemzően egyszerűbb felépítésűek, a fejlesztők hangsúlyt a p2p hálózati protokollra helyezik, ilyen többek közt az Open Chord és a Free Pastry. Ezeknek a megoldásoknak az előnye az egyszerű használhatóság akár valós hálózatokban akár szimulátorokban. Hátrányuk, hogy fejlesztésükkor nem a valódi hálózati környezet kihívásaira kerestek megoldást, hanem az elméleti eredményeket implementálták így olyan fontos szolgáltatások hiányoznak belől mint a NAT átjárás.

A sip2peer (s2p) a PeerDroid fejlesztőinek egy új projektje. Szakítottak a JXTA protokollcsaláddal, és egy új koncepció köré építették a könyvtárat. A s2p a SIP protokollt felhasználva ad megoldást a gyakorlati problémákra, a SIP-ből ismert címzést és üzenetváltási protokollt használja, lehetővé téve hagyományos SIP proxyk alkalmazását NAT lyukasztáshoz, címfeloldáshoz.

Az s2p könyvtár nem tartalmaz P2P protokollokat. Egy kényelmesen használható pont-pont SIP üzenetküldő csatornát biztosít a két csomópont közt, esetlegesen proxy-k közbenjárásával. A fejlesztők teljes szabadságot kapnak az alkalmazáshoz illeszkedő P2P overlay hálózat kialakítására.

# Minta alkalmazás felépítése és működése

A minta-alkalmazás négy osztályból áll, két ablakot és egy chat-felületet tartalmaz, valamint a csomópontot.

## MainWindow

Az alkalmazás belépési pontja a MainWindow, ez az ablak ad lehetőséget további csomópontok létrehozására. Minden csomópont az eszköz elsődleges interface-én fog kommunikálni, ez a laborban a belső LAN hálózathoz csatlakozik, így téve lehetővé a mérés végén az egymással való kommunikációt.

## ChatWindow

A csevegőprogramokból megismert felületet biztosít, bal oldalon a kapcsolatok listája látható, jobb oldalon tabulált felületen jelennek majd meg a beszélgetések. A felület egyelőre egyetlen menüelemet tartalmaz, ez egy új kontakt hozzáadását teszi lehetővé.

## ChatPanel

Ez a felület a ChatWindow tabulált nézetének egy eleme lesz, két felhasználó üzenetváltását prezentálja.

## ChatPeer

Placeholder osztály az s2p csomópont megvalósítására.

## ChatMessage

Egy chat üzenet megvalósítása.

# Feladatok

A labor során a minta-alkalmazást fogja kiegészíteni egy üzenet-osztállyal és az üzenetküldéssel.

* Töltse le a minta-alkalmazás kiindulási és végállapotát. Indítsa el az Eclipse környezetet és importálja a kiindulási állapotot. Az elérhető végállapot csak egy lehetséges, nem feltétlenül optimális megoldás, a labor célja nem a kész alkalmazás kódbázisának átmásolása hanem az egyéni problémamegoldás, a s2p könyvtár megismerése.
* Futtassa a mintaalkalmazást!
* Tekintse át a mintaalkalmazás kódját, a Tasks nézetben látni fogja az elvégzendő feladatokat és azok javasolt helyét.

Miután megismerkedett a feladatokkal nekikezdhet az implementálásnak.

* Biztosítsa, hogy a felhasználó cseveghessen a kiválasztott kontakttal: ha a kiválasztott kontakthoz nem tartozik tab, hozzon létre egyet!
* Származtassa le a ChatPeer-t a Peer osztályból. A Peer egy absztakt osztály mely lehetővé teszi üzenetek küldését és fogadását.
* Implementálja a hiányzó metódusokat melyek sikeres üzenet-továbbításkor vagy hiba esetén hívódnak meg!
* Implementálja az üzenetfogadási metódust. A labor során JSON alapú özeneteket fog használni, ezeket az onReceivedJSONMsg metódus fogadja. Az üzenetfogadás mechanizmusa a következő:
* A beérkező üzenetet meg kell vizsgálni, ha a típusa megfelelő, kinyerni belőle az üzenet törzsét.
* Ha a beérkező üzenet eddig ismeretlen felhasználótól érkezett, fel kell venni a felhasználót a kapcsolatok listájára és létre kell hozni egy új beszélgetés tabot. Implementálja ezt a folyamatot! Ha már ismert felhasználótól érkezett az üzenet, akkor a vonatkozó tabot ki kell választani!
* Implementálja az üzenetfogadási metódust. A labor során JSON alapú özeneteket fog használni, ezeket az onReceivedJSONMsg metódus fogadja. Az üzenetfogadás mechanizmusa a következő.
* Implementálja az üzenetküldést!