Sip2Peer alapú gyűrű

P2P alkalmazások a gyakorlatba

# Bevezetés

Az első mérés alkalmával az S2P csomaggal ismerkedett meg, és felépített használatával egy egyszerű chat alkalmazást. Bár ez az alkalmazás a szó szoros értelmében P2P alkalmazás volt, a csomópontok egymással közvetlenül kommunikáltak, egy ilyen megoldás a gyakorlatban nem állná meg a helyét. A felhasználók nem szeretnének IP címeket fejben tartani ahhoz, hogy ismerőseiket elérjék. Az IP címek változnak, és a NAT-olt eszközöket nehéz megszólítani.

A második mérés során az első alkalmazás funkcionalitását fogja kibővíteni: a felhasználók ezentúl azonosítójuk alapján érhetik majd el egymást. Ehhez egy önszerveződő gyűrűt fog használni. A mérés során elkészített gyűrű sok lényeges problémára - skálázhatóság, robosztusság - nem ad választ, de kiindulási alapja lehet a további munkának.

A mérés célja a gyűrű szervezéséhez és a gyűrűn való üzenetküldéshez szükséges protokoll kidolgozása és megvalósítása.

# Minta alkalmazás felépítése és működése

A minta-alkalmazás tartalmaz minden szükséges osztályt. Az előző mérésen készített alkalmazáshoz képest kicsit át lett dolgozva hogy támogassa az új, kulcs alapú címzést.E mellett a ChatWindow menüsorában a Peer menü egy elemmel bővült: most már nem csak társalogást lehet indítani, hanem csatlakozás is kezdeményezhető más csomópontokhoz, itt szükséges az IP cím és port megadása. Az üzenetek címzése azonban kulcs alapú lett, így a kontaktokat kulcs alapján veheti fel a felhasználó, a dialógusról lekerült a cím és port beviteli mező, de ezektől eltekintve minden maradt a régi.

# A protokoll leírása

A hálózat csomópontjai rendelkeznek egy String típusú kulccsal (key), ezt a PeerDescriptor osztály kezeli. Ez az azonosító kölcsönösen egyértelműen hozzárendelhető a csomóponthoz – az azonosító megválasztása a felhasználókra van bízva, megegyezik a felhasználónévvel, a peer létrehozásakor lehet beállítani. Minden csomópont megegyező funkcionalitású, a csomópontok közt a protokoll szempontjából nincsen kitüntetett csomópontok.

#### Csatlakozás

A csomópontok egy egy elemű gyűrűt hoznak létre. Más csomópontok ehhez a gyűrűhöz csatlakozhatnak, vagy a csomópont csatlakozhat egy meglévő gyűrűhöz. Az újabb csomópontok mindig képesek kell hogy legyenek a gyűrűhöz csatlakozni, akármelyik csomópontnál is lépnek be a hálózatba. Csatlakozáshoz csak a gyűrű valamely csomópontjának címét kell ismerniük, ezt egyéb, nem specifikált csatornákon keresztül szerezhetik meg (pld. megkérdezhetik a mellettük ülőt szóban).

A csatlakozás folyamata:

– A csatlakozó csomópont elküldi a leíróját az általa ismert csomópontnak egy SignalingMessage-ben.

– A csatlakozási szándékot észlelő csomópont eldönti, hogy a csatlakozó csomópont az ő felső szomszédja lesz-e vagy sem. Ha közvetlen szomszédról van szó, akkor visszaküldi a leendő csomópont leendő közvetlen következő szomszédjának leíróját a csatlakozást kezdeményező csomópontnak, és felveszi azt új szomszédjának. Egyébként továbbadja a kérést a közvetlen szomszédnak.

– Amikor a csatlakozni szándékozó csomópont megkapja a leendő szomszédjának leíróját, felveszi azt következő szomszédjának, a leendő szomszédja pedig kicseréli saját leendő szomszédját az új csomópontra. A csatlakozás sikeres.

#### Üzenetküldés

A felületen lehetőség van szöveges üzenet küldésére. Az üzeneteket a csomópont azonosítójával kell megcímezni. Az üzenetet a csomópontok a kialakított gyűrűn továbbítják, mindig a szomszédjuknak küldve azt.

Amikor egy csomópont üzenetet kap, megnézi, hogy az ő azonosítójának van-e címezve. Amennyiben igen, úgy az üzenetet a felületen megjeleníti. Ellenkező esetben az eredeti feladót és címzettet megtartva továbbadja a szomszédjának.

Amikor egy csomópont saját maga által címzett üzenetet kap, az eldobja, és a felhasználónak jelzi a sikertelen kézbesítést

# Feladatok

A labor során a fent specifikált funkcionalitást kell megvalósítania a kapott projektből kiindulva. Hogy megoldása kompatibilis legyen más hallgatók megoldásaival, ne módosítsa az üzeneteket, a feladat megoldható kizárólag a RingPeer osztály kiegészítésével.

A feladat dekomponálását TODO üzenetek segítik, de ezek csak irányt mutatnak, figyelmen kívül is hagyhatóak. Egy lehetséges megoldást is letölthet a tárgy oldaláról.

# Szorgalmi feladatok

A mai labor anyaga rövidebb mint a megelőzőé: kizárólag a hálózati protolollt kell megvalósítania. Ha ezzel végzett, a következő területek közül választhat.

#### Redundancia

Egészítse ki a protokollt úgy, hogy a csomópontok k közvetlen szomszédjukat ismerjék. Sikertelen üzenetküldés esetén próbálják meg megjavítani a hálózatot.

#### Hatékonyság

Egészítse ki a protokollt úgy, hogy a csomópontok a String azonosítóik egy 16 bites hash értéke alapján rendeződjenek sorba. Az egyes csomópontok ismerjenek távolabbi csomópontokat is, így rövidítve le az üzenetküldés hop-számát.

A Chord hálózatban használt módszert alkalmazhatja: <http://en.wikipedia.org/wiki/Chord_%28peer-to-peer%29>.