

Asterisk tapasztalatok Sopronban, a Nyugat-magyarországi Egyetemen

2012 végén vezettük be az Asterisk rendszert Sopronban. A korábban is meglévő telefonközpontunk nagyszámú analóg (POTS) és zárt szabványú IP (Unistim) melléket valamint néhány digitális melléket kezel. Az Asterisk központ célja nem a régi központ teljes kiváltása, hanem párhuzamos működés, az új épületek ellátása nyílt szabványú (SIP) IP mellékekkel. A régebbi telefonközpont E1 felületen keresztül kapcsolódott a Telekom és az NIIF VoIP rendszeréhez, és saját maga döntötte el, melyik irányba küldi a kimenő hívásokat. Az új felállásban az alapvető forgalomirányítási feladatokat egy Cisco AS5350 gateway végzi. Ez a gateway a Telekom, az NIIF VoIP és a régi telefonközpont felé összesen öt E1 interfészen keresztül kapcsolódik, az Asteriskkel azonban SIP kommunikációt folytat. Ezen elrendezés lehetővé tette, hogy az Asterisk szerver E1 vagy bármilyen más vonali kártya nélkül, virtuális szerveren fusson. A rendszer összetettsége okán és tapasztalatok hiányában rendszerintegrátorral végeztettük el a hibrid telefonrendszer kiépítését.

Időközben átvettük a rendszer menedzselését, jelentősen optimalizáltuk a működését, és célul tűztük ki, hogy az Asterisket képessé tesszük a régi központ használatba nem vett vagy időközben felszabaduló zárt szabványú IP telefonjainak hibátlan felhasználó élményt nyújtó működtetésére, valamint egy softphone rendszer kiszolgálására. A softphone rendszert ingyenes multiplatform szoftverekkel valósítottuk meg oly módon, hogy megfelelő hangminőséget, felhasználóbarát kezelőfelületet, megbízható működést és elfogadható biztonságot nyújt akár a legszigorúbb tűzfal és NAT környezetben is.

10 A softphone rendszer kialakítása

Az Asterisk szerverre alapozva egy softphone rendszer kiépítését határoztam el. Az elképzelésem az volt, hogy a külföldi úton levő munkatársaink az okostelefonjaikra vagy számítógépeikre telepített alkalmazás(ok)at használva ingyenes WiFi kapcsolaton – jellemzően az Eduroam-on – keresztül csatlakozhassanak az Asteriskhez. Az ilyen módon egyetemi telefonmellékké változtatott készülékek értelemszerűen magyar vezetékes telefonszámmal rendelkeznek, és ingyen, korlátlanul beszélhetnek egymással, az összes többi egyetemi mellékkel, és az ún. flat rate konstrukciójú távközlési szerződésünkből kifolyólag minden magyar vezetékes és mobil állomással.

A softphone rendszerben felhasználandó VoIP alkalmazásokkal kapcsolatos követelmények a következők voltak:

- Multiplatform (Android, iOS, Windows, Linux, stb.)
- Könnyen kezelhető, felhasználóbarát
- Ingyenes
- Lassú, rossz minőségű hálózatban is használható
- NAT mögül is működik
- Nagyon szigorú tűzfalakon is átjut
- Titkosított jelzést és médiát biztosít

A legnyilvánvalóbb nehézséget az okozza, hogy majdnem minden Asterisk által támogatott VoIP protokoll RTP-t használ a média küldésére. Ez tűzfal és NAT környezetben is rendkívül problémás, mivel egy UDP kapcsolat csak egyetlen egyirányú hangcsatornát hordoz. Az RTP adatfolyamok kiindulási- és végpontjainak IP címei eltérhetnek a jelzési információkat hordozó kommunikációt lefolytató címeiktől, az UDP portszámok nem meghatározottak és hívásonként véletlenszerűen változhatnak, ráadásul az RTP médiacsatorna menet közbeni diagnosztikáját és finomhangolását segítő RTCP protokoll minden médiacsatornához egy-egy további UDP kapcsolatot is elhasznál, így egyetlen (videocsatorna nélküli) beszélgetés akár négy UDP kapcsolatot is elfoglalhat, amelyek közül kettőt ráadásul a VoIP szerver kísérel megnyitni az esetleg NAT mögött levő kliens felé.

Az RTP protokoll véletlenszerű, 1024 feletti portokon működő, illetve visszafelé irányuló UDP kapcsolatai önmagukban is sok nehézséget okoznak NAT és tűzfal környezetben, azonban a H.323-hoz képest egyszerűnek nevezhető, internetbarát SIP protokoll további komplikációkat okoz, ugyanis előfordulhat (különösen NAT környezetben), hogy a SIP parancsok számára az egyetlen – a kliens egy 1024 feletti portja felől a szerver 5060-es portja felé irányuló – kezdeti UDP kapcsolat mellé további kapcsolatok nyílnak, akár a szerver felől a kliens irányába.

Az RTP-re épülő VoIP protokollokról – köztük a softphone projekt egyik legfontosabb jelöltjéről, a SIP protokollról – elmondhatjuk tehát, hogy alapesetben csak teljesen szüretlen, NAT mentes hálózati környezetben működnek tökéletesen.

A SIP vezérlést hordozó adatfolyam NAT-baráttá tételére alkották meg a SIP üzenetek „via” fejlécébe beszúrandó „rport” opciót (RFC 3581). Ha egy kliens ezzel az opcióval küldi a SIP kéréseit a szervernek, akkor a szerver a válaszokba nem csak azt írja be, hogy az ő szemszögéből nézve mi a kliens IP címe, hanem azt is, hogy a kliens mely UDP portjáról látja megérkezni a SIP üzeneteket. Ezen felül a szerver minden további SIP üzenetet arról az IP címről és portról küld a kliensnek ahova a SIP kéréseket kapja, illetve arra az IP-címre és portra küldi őket, ahonnan érkeztek. Az „rport” opció bekapcsolásával tehát legalább a SIP jelzéseket egyetlen UDP (vagy TCP) csatornára korlátozhatjuk, ezáltal biztosítva, hogy NAT- és tűzfalállók legyenek. Ha a kliens esetleg nem támogatná az rport opciót, a „force_rport” konfigurációs lehetőség arra kényszeríti az Asterisk szervert, hogy pontosan úgy járjon el, mintha „rport” jelöléssel kapta volna a klienstől a SIP kéréseket.

Az RTCP protokoll használata nem kötelező, és nem is teljesen elterjedt, így már csak az RTP adatfolyamok NAT-baráttá tételén kell dolgoznunk. Ezen a téren segítséget jelent, hogy a VoIP kliens alkalmazások általában ugyanarról az UDP portról küldik el a kimenő hang audiofolyamát, mint ahová a beérkező adatfolyamot várják. Az Asterisk SIP csatornájának „comedia” opciója hasonló viselkedést vált ki: Az Asterisk szerver előbb megvárja a bejövő audiofolyam első csomagját, és feltétel nélkül ugyanezen az UDP kapcsolaton keresztül küldi el a kimenő hangcsatornát.

A szerveroldali „force_rport” és „comedia” opciók mellett kliensoldali NAT-kezelő megoldások is léteznek, pl. a STUN (RFC 5389) és TURN (RFC 5766) protokollok, valamint ezek kombinációja az ICE protokoll (RFC 5245), ezeket azonban feleslegessé teszik az említett Asterisk megoldások. Az RTP protokoll véletlenszerű UDP portjai szigorú tűzfalkörnyezetben működésképtelenné teszik a fentiek szerint NAT-baráttá tett SIP protokollt, ezért mérlegeltem az IAX (Inter-Asterisk eXchange) protokoll bevezetését is, mivel ez a szabvány minden jelzést és hangot egyetlen UDP kapcsolaton (a 4569-es porton) visz át, de elvettem, mert a gyakorlatban nem titkosítható, és a SIP protokollal ellentétben rossz hangminőséget produkált a kiválasztott iLBC kodekkel.

A SIP kommunikáció titkosítására a SIP/TLS (SIPS), illetve SRTP/ZRTP megoldásokkal kísérleteztem, de csak egyetlen VoIP klienset találtam, ami maradéktelenul támogatta ezeket a szabványokat. Ez a kliens a CsipSimple alkalmazás, és sajnos csak Androidra létezik. A titkosított SIP kommunikáció ráadásul semmivel sem enyhíti a tűzfalakon való átjutás nehézségeit, úgyhogy célszerűbbnek tűnt egy VPN megoldás használata. Először a Cisco IPSec megoldásával kísérleteztem, de mivel az Android beépített VPN kliense hónapok (ha nem évek) óta hibás, és nincs ingyenes alternatívája, a nyílt, multiplatform OpenVPN technológiára váltottam. Az OpenVPN az IPSec-kel ellentétben egyetlen UDP porton (1194) kommunikál, amelyet a rendkívül elterjedt IKE protokoll 500-as portjára állítottam be, ezáltal leküzdve minden NAT és tűzfal akadályt. Az OpenVPN klienset úgy konfiguráltam be, hogy csak a VoIP forgalmat route-olja a VPN engine bejáratául szolgáló virtuális interfész (pl. tun0) felé. A SIP kliensek az SDP paraméterek segítségével adják meg a szervernek, hogy mely interfészük IP címére küldje az RTP adatfolyamot. Általában az alapértelmezett átjáróra néző interfész címét adják meg (ami értelemszerűen nem a tun0), ezért a kliens felé bejövő adatfolyam a VPN csatornán kívül haladna, ha találna magának utat a kliens és a szerver között levő NAT és tűzfal eszközökön keresztül, de mivel ez általában nem sikerül, a SIP és a VPN kliensek kombinációja alapesetben egyirányú hangot eredményez. Ezt úgy lehet kiküszöbölni, ha a VoIP kliens rávesszük, hogy az SDP paraméterekben a tun0 interfész IP címét adják meg a bejövő RTP adatfolyam céljaként. Ez mind a CsipSimple alkalmazásban, mind a későbbiekben használt Zoiper kliensekben megoldható.

A hálózati problémák leküzdése után a kezdetben rendkívül rossz hangminőség javítását tűztem ki célul. A hangminőségre a legnagyobb hatással a megfelelő kodek kiválasztása volt, mivel a különböző kodek sávszélességigénye rendkívül eltérő, és vannak köztük csomagvesztésre és jitterre (internet) optimalizált megoldások, illetve csomagkorruptióra (GSM hálózat) optimalizált megoldások is. Az Asterisk SIP csatornájának jitterbuffer beállításai is sikerült elérni némi javulást, azonban a bufferek által okozott késleltetés értelmezhetetlenné tette a résztvevők számára a beszélgetést, ráadásul rájöttem, hogy a megfelelő algoritmus kiválasztása sokkal nagyobb jelentőséggel bír. A széles körben támogatott, ingyenes, internetre optimalizált iLBC kodekre esett a választásom, mert tapasztalataim szerint a papírforma szerint hasonló teljesítményt nyújtó kodekeknel jobb felhasználói élményt biztosít a gyakorlatban.

20 UNISstim szabványú IP telefonok beüzemelése Asterisk alatt

A zárt szabványú UNISstim IP telefonokat az Asteriskben a reverse engineering módszerrel készült nyílt chan_unistim driver kezeli. A chan_unistim csatorna nem igazán támogatott, kísérleti drivernek számít. Problémái:

- Az Asterisk 10.x.x verzióig bezárólag nem működik a gyors és felhasználóbarát dialplan alapú tárcsázás. A „call” soft key használata kötelező minden hívás indításakor.
- 11.0.0-től új, felhasználóbarát képességek jelennek meg, de sajnos hibákat is hoznak magukkal: Ha egy UNISstim telefon felé hívás irányul, akkor az Asterisk azonnal egy folyamatos RTP adatfolyam küldésére veszi rá, amelynek csak a beszélgetés megszakításával lesz vége. Ha a telefon nem fogadja a hívást, az RTP adatfolyam nem áll le, és használhatatlanná teszi a telefont, mert az nem tud rendes audiofolyamot küldeni mindaddig, amíg a „korai RTP adatfolyam” meg nem szakad egy fogadott (de néma) hívás eredményeképpen.
- 11.7.0-től a korai RTP konszolidálódik, legalábbis a nem fogadott hívások kicsengésének vége is megszakítja a szükségtelen RTP adatfolyamot, de UNISstim → UNISstim irányú hívás esetén a hívó nem hallja a kicsengést (ún. ringback tone).
- A hívások átadásakor a telefon kijelzőjén nem a megfelelő információ jelenik meg, és a teljes Asterisk processz összeomlik.

Kapcsolatba léptem a chan_unistim driver fejlesztőjével, akit pontos diagnosztikai információkkal láttam el, teszkészülékeket küldtem neki, és egy fejlesztői környezetet is létrehoztam a számára. A hibajavítás sikerrel járt, az Asterisk szerver 11.9.0 és 12.2.0 verziói előreláthatóan mentesek lesznek a fentiekben felsorolt hibáktól.