

Követelményelemzés

A

SIP Protokoll

Projektvezető: Szabó Gyula

ELTE Eötvös Loránd Tudományegyetem

Követelményelemzés Gyakorlat

Budapest, Nov. 20, 2022

Contents

Revision History	3
1. Projekt bemutatása	4
2. Követelmények.....	5
2.1. Követelmények számozása és felépítése	5
2.2. Hardver- és Szoftver infrastrukturális követelmények.....	5
2.3. Kezelői paraméterek	7
2.4. Teszt paraméterek és adatok	9
2.5. Tesztfolyamat futtatása.....	10
2.6. Funkció hívási folyamatok	11
2.6.1. Between SIP User Agents	11
2.6.2. USER AGENT CLIENT	14
2.6.3. USER AGENT SERVER	15
2.6.4. Capability teszt	18
2.7. Teszteredmények és elemzések.....	18
3. Szójegyzék	20
3.1. SIP	20
3.2. Kommunikációs protokollok.....	20
3.3. Hardver és szoftver.....	21
3.4. Program komponensek	21

Revision History

Revision	Date	Author(s)	Description
1.0	2022.10.30	Szabó Gyula	Template létrehozása
1.2	2022.11.02	Szabó Gyula Bálint Zsolt Noll Péter	Fő pontok kialakítása az ügyfél leírás szerinti
2.0	2022.11.03	Csurilla Zsombor	Megállapodott funkciók frissítése
2.1	2022.11.07	Bálint Zsolt	Regisztrációs folyamat követelményeinek meghatározása v1
2.2	2022.11.08	Nátrán Bálint	Hardver- és szoftver követelmények elemzése v1
2.3	2022.11.12	Noll Péter	INVITE kliens oldali tesztelési követelményei és paraméterei
2.4	2022.11.13	Somogyi Ferenc	Adatbázisra vonatkozó követelmények elemzése v1
3.0	2022.11.13	Papp Péter Csurilla Zsombor	Jó és Rossz hívási funkció kommunikációs elvárásainak leírása
3.1	2022.11.13	Szabolcsi Ádám	UserAgent viselkedési esetek meghatározása magasszinten
3.2	2022.11.14	Nátrán Bálint	Hardver- és szoftver követelmények elemzése v2
3.3	2022.11.16	Kovács Péter	Capability teszt és SDP protokoll
3.4	2022.11.19	Papp Péter Szabó Gyula	Dokumentum formázása
3.5	2022.11.20	Somogyi Ferenc Nátrán Bálint	Dokumentum ellenőrzése

1. Projekt bemutatása

A projekt célja egy tesztelő program létrehozása, mely a HuTelCo vállalat hálózati mérnökei számára emulál nagy mennyiségű SIP alap funkcionalitást ellátó végesszközt, amik kielégítik a következőket:

- Többféle végpont eszköztípus emulálása, melyek száma konfigurálható
- Konfigurálható végponton generált hibás és helyes kérések aránya
- A programnak képesnek kell lennie a megfelelő és a nem megfelelő válasz detektálására
- A csomagvesztés és a (válasz nélküli üzenetek) nem megérkező üzenetek detektálására is képes
- A végpontnak az elképzelhető szoftveres hibákat tudnia kell generálni
- Konfigurálhatónak kell lennie a hívások mennyiségének és fajtájának
- Képesnek kell lennie 10 ezer végpont emulálására a megadott hardver keretein belül
- Képesnek kell lennie milliszekundum pontosságú válaszidők mérésére
- Két vagy több eszköz hang- és videóhívási összekötésének megvalósítását is tudnia kell tesztelni
- A tesztprogramnak időben változó terhelést is meg kell tudnia valósítani
- A teszt elfogadási paramétereinek változtathatóknak kell lenniük
- A tesztelfutás végén a teszt elfogadási paraméterek alapján végső konklúziót ad a tesztről

A projekt eredményessége a jól meghatározott és lefedett tesztesetektől függ.

A megrendelő elvárása, hogy a tesztprogram a lehető legtöbb hibát feltárja a megállapodott SIP funkcionalitásokban, és pontos válaszidőmérésekkel felmérje a terhelés miatt bekövetkező hibaesetek számát. A végső cél egy egyszerűen paraméterezhető tesztkörnyezet, mely a tesztelendő SIP példahálózatot a SIP protokoll által elfogadható és nem elfogadható kérésekkel, tervezett módon terheli és funkciók használatát emulálja, és az arra adott válaszokat elemzi. A helytelen vagy nemvárt működésű teszteseteket elmenti későbbi elemzésre, valamint a teszt konklúziójában több szempontból csoportosítja, átlagolja és megjeleníti a hibás eseteket. A felületre a teszt futtatása alatt folyamatosan adatszolgáltatást kell biztosítani a tesztek futási állapotról.

A program egy, a megrendelő által biztosított, szerver eszközön fog működni, melynek hardveres erőforrásai a következők:

- 8 darab, egyenként nyolcmagos, x86-64 architektúrájú processzor
- 256 GB memória (processzoronként 32 GB)
- 2 TB kapacitású háttértár
- 10 Gb/sec sebességre képes hálózati kártya

A hardver kapacitásának egy részét fenn kell tartani a program más részeinek és a grafikus felület meghajtására, mely ennek a projektnek nem része. A programnak 10 ezer eszközt kell emulálnia és 5 év időtartamra kell támogatnia az imént ismertetett funkciók halmazát.

A projekt jelentős háttérismeretet igényel a SIP (rfc3261) protokollról, melynek működési eseteit és viselkedési mechanizmusát körültekintően kell használni a tesztesetek tervezésénél cégünknek. Továbbá az SDP (rfc2327) és CALL FLOW (rfc3665) ismerete is szükséges.

2. Követelmények

2.1. Követelmények számozása és felépítése

A követelményeket tématerületekre bontjuk, melyeknek egyedi azonosítójuk van. Az egyes területeken belül a követelményeket tételesen soroljuk fel, és egyedi azonosítóval és leírással látjuk el. Az azonosítás az alábbi módon történik:

X_ID, ahol az ID követelmény azonosítója és az X a tématerületet jelöli

A tématerületek azonosítására használt jelölések:

- **HWSW** - Hardveres és szoftveres követelmények
- **KP** - Kezelői paramétereizhetőség követelményei
- **TP** - Teszt paraméterek és adatok
- **T** - Tesztfolyamat futtatása
- **P_I_UAC** - User Agent Client viselkedés leírása
- **P_I_UAS** - User Agent Service viselkedés leírása
- **P_CT** - Capability teszt viselkedés leírása
- **TR** - Teszt végeredmények követelménye

Más dokumentumokra való hivatkozások felépítése

docTitle/section/subSection/requirement_id , ahol:

- **docTitle** a dokumentum címe
- **section** a szekció száma (opcionális)
- **subSection**: al-szekció száma (opcionális)
- **requirement_id**: téma terület azonosítója (opcionális)

2.2. Hardver- és Szoftver infrastrukturális követelmények

Ebben a részben az emuláció futtatásához szükséges minimális és ajánlott szoftveres- és hardveres rendszerkövetelményeket ismerhetjük meg.

Azonosító	Kifejtés
HWSW_01	A szoftvert futtató processzor az alábbi tulajdonságokkal kell, hogy rendelkezzen: <ul style="list-style-type: none"> - 10 000 eszköz egyidejű SIP kommunikáció emulálásának számítási igényét fedeznie kell. - 1 000 teszt modul futtatását ki kell tudnia szolgálni, de az emulált eszközök kumulatív száma nem haladhatja meg egyidőben a 10 000 eszközt. - Valamint kompatibilisnek kell lennie a HWSW_05-ben felsorolt operációs rendszerekkel.
HWSW_02	Gyors író-olvasó háttértár szükséges, az alábbi paraméterekkel:

	<ul style="list-style-type: none"> - Legalább 4000 kommunikációs log kiírását el kell tudnia végezni másodpercenként (a logok átlagos mérete 1 KB). - Minimum 1TB tárhely és SSD technológiájú háttértár szükséges a gyors írás-olvasás megvalósításához. <p>Megjegyzés: a program alapvető működéséhez elegendő 50 GB tárhely, de ebben az esetben nem adott a 10 000 eszköz több órás tesztfuttatásának képessége. Javasolt a RAID biztonsági funkcióit kihasználni az értékes teszteredmények megtartásáért.</p>
HWSW_03	<p>Memória mérete:</p> <p>Eszközönként legalább 5 MB memória szükséges a funkciók teszteléséhez, valamint minden tesztmodul folyamatos működéséhez elengedhetetlen 30 MB memória. Az adatbázist működtető programrész és a tesztek futtatását összefogó programrész minimum. további 1500 MB memóriát igényel maximális kapacitás mellett.</p> <p>Típusokra lebontva:</p> <ul style="list-style-type: none"> - 10 000 eszköz esetén 50 GB memória - 1000 tesztmodul esetén 30 GB memória - Program további összetevői összesen 1,5 GB memória <p>Összesen legalább 81,5 GB szabad memória szükséges.</p>
HWSW_04	<p>Memória típusa:</p> <p>Virtuális memória nem megengedett, mivel a gyorsaság kritikusan fontos. Legalább 128GB, DDR5 vagy DDR4 gyorsaságú memória szükséges.</p>
HWSW_05	<p>Operációs rendszer:</p> <p>A Linux disztribúciók közül az Ubuntu utolsó két LTS változatát kell támogatnia.</p> <p>Ezek a következők:</p> <ul style="list-style-type: none"> - 20.04 LTS Focal Fossa (Támogatott: 2025. április) - 22.04 LTS Jammy Jellyfish (Támogatott: 2027. április)
HWSW_06	<p>Emulált adatmennyiség hálózati sávszélesség-követelménye:</p> <p>A tesztelő programnak a SIP kérések mellett a megvalósuló hang- és videóhívások adatforgalmát is kell emulálnia. Ez a forgalom a program átadása utáni 5 évben növekedni fog, ezért a mai forgalomhoz képest 30%-os növekedéssel kell számolni. Ezen terhelések a következők:</p> <p>Átlagos forgalom a különféle hívások során (le- és feltöltési sebességek):</p> <ul style="list-style-type: none"> - Hanghívások esetén: 35 kbps/35 kbps - Hang konferencia esetén: 245 kbps/35 kbps - Videóhívás esetén: 864 kbps/864 kbps - Videókonferencia esetén: 6048 kbps/864 kbps <p>A hálózati kártyának a következő adatforgalmi összetételt kell ellátnia kumulatív:</p> <ul style="list-style-type: none"> - Maximálisan 650 eszköz videóhívását - Maximum 70 eszköz konferencia videóhívását - Maximálisan 1500 eszköz hanghívását - Maximum 160 eszköz konferencia hanghívását

	- Ezen hívásokkal párhuzamosan a SIP funkció hívások megvalósítása maximum 10 000 eszközíg
HWSW_07	Hálózati kártya követelménye: Legalább 10 Gb/sec sebességű hálózati kapcsolattal kell rendelkeznie a futtató rendszernek. Továbbá kompatibilisnek kell lennie HWSW_05 követelményben meghatározott operációs rendszerekkel.
HWSW_08	A programnak rendelkeznie kell egy, a teszteredmények tárolásához megfelelő, konkurenciakezelésre képes adatbázissal.
HWSW_09	Az operációs rendszer jellege miatt az adatbázis használatához XFS vagy EXT4 típusú fájlrendszert támogató adatbázis megoldás szükséges.
HWSW_10	Az ügyfél által megadott elérhető maximális hardver erőforrások <ul style="list-style-type: none"> - CPU: <ul style="list-style-type: none"> o 8 processzor (x86-64) o 8 mag processzoronként - Memória: <ul style="list-style-type: none"> o 32 GB RAM processzoronként o Összesen 256 GB - Háttértár: <ul style="list-style-type: none"> o 2TB SSD tárterület - Hálózati kártya: <ul style="list-style-type: none"> o 10 Gb/s sebességű Ethernet

2.3. Kezelői paraméterek

A kezelő képes megalkotni egy tesztelési körülményt a grafikus interface-en (bővebben: Grafikus tesztelő felület/3/3.5) ezt ezentúl tesztelési blokknak hívjuk. Ezek a blokkok elkülönülnek az egész tesztre érvényes adatokra, valamint egy 60 perces felbontásban megadható, változtatható paraméterlistára. Ebből többet is szerkeszthet, melyeket következő lépésként a tesztelési modulban futtathat le, akár párhuzamosan 1000 tesztelő blokk is. A tesztelési modulon belül a futó blokkokban létrejönnek a funkció tesztelésére szolgáló funkció tesztelési blokkok. Ezen blokkok és funkciók paramétereit a kezelő mérnök a grafikus interface-en vagy a konfigurációs fájlokban állíthatja be. Az elindult tesztelő blokkon semmilyen paramétert nem lehet módosítani.

Azonosító	Kifejtés
KP_01 (regisztráció)	A kezelő képes megtervezni egy tesztet, melynek a következő adatait kell megadni a regisztrációs folyamatban: <ul style="list-style-type: none"> • User equipmentek száma (regisztrációs darabszám) • User equipmentek összetétele • Kívánt hibák generálásának aránya (százalékban) • Az elemzési elfogadási paraméterek <ul style="list-style-type: none"> o Megengedett válaszidő o Próbálkozások száma
KP_02	A kezelő képes megtervezni egy tesztet, melynek a következő adatait kell megadni a regisztrációs folyamatban:

	<ul style="list-style-type: none"> • Hívások száma • Hívások összetétele <ul style="list-style-type: none"> ○ Hang ○ Videó ○ Konferencia • Kívánt hibák generálásának aránya (százalékban) • Az elemzési elfogadási paraméterek <ul style="list-style-type: none"> ○ Megengedett válaszidő ○ Próbálkozások száma
KP_03	<p>A programnak biztosítania kell a kezelő számára, hogy az képes legyen megtervezni egy tesztelési blokkot (bővebben: Grafikus tesztelő felület/3/3.5) melynek a következő adatait kell megadni:</p> <ul style="list-style-type: none"> • Teszt neve • Teszt leírása • Teszt időtartama • 60 perces felbontásban: <ul style="list-style-type: none"> • User equipment-ek száma (regisztrációs darabszám) • User equipment-ek összetétele • Funkció tesztek • Kívánt hibák generálásának aránya • Az elemzés elfogadási paraméterek
KP_04	<p>Konfigurációs file-ok szerkesztésével megadható konfigurációk: Példa hálózat adatai</p> <ul style="list-style-type: none"> • Registrar szerver elérése RFC3261 - 10.2.6 <p>Eszközök adatai:</p> <ul style="list-style-type: none"> • Késleltetések eszközönként • Eszközök képességeinek megadása százalékos arányban • Szabványos üzenet és válasz adattárak • Hibás üzenet és válasz adattárak
KP_05	<p>User equipment-ek összetétele darabszámban:</p> <ul style="list-style-type: none"> • IP telefon • Okostelefon wifi • Okostelefon mobilhálózat • PC <p>Az eszközök emulált késleltetési adatai konfigurációs file-ban vannak elmentve és tesztblokkonként nem konfigurálhatóak, de az egész programra nézve igen.</p> <p>Funkció tesztekre értelmezett bemenetek:</p> <ul style="list-style-type: none"> • Regisztráció <ul style="list-style-type: none"> • Hibás kérések aránya • Hívások darabszáma <ul style="list-style-type: none"> • Hívás típusok aránya <ul style="list-style-type: none"> • Hanghívás aránya • Videóhívás aránya (kettő együtt 100%) • Konferencia hívások aránya <ul style="list-style-type: none"> • Hanghívás esetén

	<ul style="list-style-type: none"> • Videóhívás esetén • Hibás kérések aránya
KP_06	<p>Konferencia hívások:</p> <p>Mivel a konferencia hívások nagyon eltérő adatkommunikációs igényekkel rendelkeznek a normális hívásokhoz képest, ezért a darabszámuk befolyásolására lehetőséget kell adni. (videó és hang hívásra külön-külön). A kumulatív darabszámok erősen korlátozottak, így nem futtat több olyan tesztet, amelyek egyidőben ezen számokat meghaladják.</p>
KP_07	<p>Az elemzés elfogadási paraméterek:</p> <p>Az funkció tesztelési blokk lefutását akkor fogadja el helyesnek a funkció tesztelő modul, ha az alábbi paramétereket nem sérti meg a teszt futása. Mivel a forgalom ingadozó, ezért lehetőség van a forgalommal együtt 60 perces felbontásban változtatni az elemzési kritériumokat.</p> <p>Regisztráció esetén:</p> <ul style="list-style-type: none"> • Regisztrációs próbálkozások száma • Regisztrációs kérésre kapott válasz késleltetése <p>Hívás esetén:</p> <ul style="list-style-type: none"> • Hívási funkció próbálkozások száma • Hívási funkció megengedett késleltetése

2.4. Teszt paraméterek és adatok

A teszt paraméterek természetesen magukba foglalják a Kezelői Paraméterek követelmény pontokban felsorolt adatokat és adattípusokat, de ezen teszt paraméterekből a kezelő csak a fent említett pontokat tudja manipulálni közvetlenül. A teszt paraméterek egy részét a kezelő, másik részét pedig a funkcionális tesztelő blokk állítja elő. A funkció tesztelő a helyes és hibás kéréseket tároló adatállományok és a foglaltsági tábla alapján generálja, majd hajtja végre ezen paramétereket.

A teszt paraméterekből

Azonosító	Kifejtés
TP_01	<p>A funkció tesztek kéréseinek szerkezeti szinten konfigurálhatónak kell lenniük</p> <ul style="list-style-type: none"> - Kérés header-ök szintjén - Kérés body szintjén - Válasz header-ök szintjén - Válasz body szintjén
TP_02	<p>Az INVITE kérés header-jeinek szabadon konfigurálható paraméterei - RFC3261 - 13.2.1</p> <p>Kötelező paraméterek:</p> <ul style="list-style-type: none"> - Allow - Supported
TP_03	<p>A teszt paraméterek és adatok a modul szinteken</p> <p>Tesztelő modul paraméterei (ami futtatja a teszteket)</p>

	<ul style="list-style-type: none"> • A tesztblokkok listája • Mennyi teszt blokk fut adott időben (max. 1000) • Kumulatív maximum eszköz szám (max. 10 000) <p>Tesztelő blokk adatai</p> <ul style="list-style-type: none"> • User equipment-ek adatai • Eszköz foglaltsági tábla • Futó funkció tesztelő blokkok listája <p>Funkció tesztelő blokk adatai</p> <ul style="list-style-type: none"> • Elsődleges eszköz ID • Másodlagos eszközök ID listája • Funkció típusa és összetétele • Kommunikációs logok és teszteredmények
TP_04	<p>Funkció típusa és összetétele:</p> <p>Funkció típusai:</p> <ul style="list-style-type: none"> • REGISTER RFC 3261-10.1 • INVITE • ACK • BYE <p>Összetétele:</p> <p>A funkció tesztelő a helyes és hibás kéréseket tároló adatállományok és a foglaltsági tábla alapján generálja, majd hatja végre ezen paramétereket.</p>

2.5. Tesztfolyamat futtatása

A program a kezelő által korábban rögzített teszt blokkokat futtatja. Ezek a blokkok elkülönülnek egymástól. A kumulatív user equipment szám legfeljebb 10 000 lehet, valamint a futó tesztblokk szám nem lehet több mint 1000. A funkcionális tesztek kumulatív számára is van megkötés a HWSW_06 -ban. A tesztblokkokban futnak a funkció tesztek blokkjai, amik létrejönnek, ha egy funkciót akarunk tesztelni (regisztrációnál és hívásnál egyaránt). A 60 perces felbontásban változó user számot a regisztrációs kérelmek változtatják, de amíg tart egy hívás, (maximum 60 perc), addig nem lehet leregisztrálni az eszközt a SIP szerverekről.

Azonosító	Kifejtés
T_01	A különböző emulált hálózati műveletek végrehajtásának mérési idejét 1 milliszekundum pontossággal kell mérni
T_02	<p>A folyamatot fázisonként modellezni kell tudni és fázisonként, fázison belül lépésenként működést definiálni képesnek kell lenni</p> <p>Kliens oldal esetén:</p> <ul style="list-style-type: none"> - SIP regisztráció elvégzése az összes emulált végpontra - A kérés összeállítása - A kérés elküldése - Válasz kezelése

	<p>A teljes procedúrában jól definiáltan képesnek kell lennie a "happy path"-t, azaz hibamentes útvonalat reprodukálni kliens és szerver oldalon, mely esetén a kapcsolatnak fel kell állnia a két végpont között. A protokoll által definiált lépésekben lépésenként konfigurálhatónak kell lennie a helyes működéstől való eltérésnek és annak módjának. Képesnek kell lenni a kérés és válasz között eltelt idő, valamint két vagy több eszköz közötti kommunikációs késleltetés mérésére.</p>
T_03	<p>Az INVITE kérést a funkcionális tesztelő blokk tudja konfigurálni helyes vagy helytelen üzenetté rfc3261 - 13.2.1 Kötelező paraméterek: rfc3261 - 14.1</p> <ul style="list-style-type: none"> - Allow - Supported <p>Az INVITE kliens oldali kommunikáció tesztjében a szervertől érkező válasz típusai: Hibát jelző választípusok: rfc3261 - 14.2</p> <ul style="list-style-type: none"> - 1xx válasz - 3xx válasz - 4xx, 5xx, 6xx válasz <p>Helyes működést jelző válasz: <ul style="list-style-type: none"> - 2xx válasz </p> <p>Az INVITE szerver oldal válaszára küldött ACK válasz rfc3261 - 15</p>
T_04	<p>A tesztfolyamat futtatása közben a programnak képesnek kell lennie minden teszteset lefutása után a teszteredmények adatbázisba írására, valamint azok adatbázisból való kiolvasására is a grafikus felület számára.</p>
T_05	<p>Az adatbázisban tárolt teszteredményekről biztonsági mentésnek kell készülnie minden tesztalmanoz lefutása után</p>
T_06	<p>A teszteredmények folyamatos lekérése miatt az adatbázisba írás sebességének minimum 2 MB/s-nak kell lennie folyamatos 4 KB-os random adat írási metódus alatt is.</p>

2.6. Funkció hívási folyamatok

A programnak a következő hívási eseteket kell tudnia lekezelni, rögzíteni és elemezni a bemeneti paraméterek alapján. A funkció tesztelési blokk ezen követelmények mentén állítja össze a helyes és helytelen kéréseket és ennek alapján értékeli ki a kapott válaszokat.

Jelölések:

- UAC-A – User Agent Client A
- UAC-B – User Agent Client B

2.6.1. Between SIP User Agents

	Elvárt működés
	UA SIP REGISTER - SUCCESS

<p>P_I_01</p>	<p>Az adott UA inicializálja a regisztrációs folyamatot egy a Registrar felé küldött SIP REGISTER parancs küldésével.</p> <p>A kérés paraméterei a teszteseteket összeállító teszt-mérnökök által állíthatóak.</p> <p>Az alábbi kötelező mezőkkel kell felparaméterezni a SIP REGISTER kérést:</p> <ul style="list-style-type: none"> • Registrar IP címe / DNS neve • VIA Header és branch tag • From Header • To Header • Call-ID Header • Contact Header • CSeq Header • Content Length Header • Max-Forwards Header <p>Ekkor a Registrar a 401– Unauthorized választ adja és a válaszba tesz egy WWW-Authenticate Header-t.</p> <p>Az WWW-Authenticate Header-t felhasználva kigeneráljuk a megfelelő MD5 hash-t, és ezt a következő lépésben az Authorization Header-ben használjuk fel.</p>
<p>P_I_02</p>	<p>Az adott UA folytatja a regisztrációs folyamatot egy a Registrar felé küldött SIP REGISTER parancs küldésével.</p> <p>A kérés paraméterei a teszteseteket összeállító teszt-mérnökök által állíthatóak.</p> <p>Az alábbi kötelező mezőkkel kell felparaméterezni a SIP REGISTER kérést:</p> <ul style="list-style-type: none"> • Registrar IP címe / DNS neve • VIA Header és branch tag • From Header • To Header • Call-ID Header • Contact Header • CSeq Header • Content Length Header • Max-Forwards Header • Authorization Header <p>Ekkor, ha az Authorization Header értékét jól számoltuk ki, a Registrar a 100 – Trying választ adja.</p>

	Ezután a Registrar a 200 – OK választ adja, és ezzel a REGISTER kérés sikeresen teljesítettnek minősül.
	UA SIP REGISTER - FAILURE
P_I_03	A folyamat első lépése ugyanaz, mint a UA SIP REGISTER – SUCCESS első lépése.
P_I_04	<p>A folyamat második lépése hasonlóan zajlik az UA SIP REGISTER – SUCCESS második lépéséhez, de itt az alábbi lehetőségek közül választhatunk:</p> <ul style="list-style-type: none"> Rossz felhasználónevet adunk meg a REGISTER kérés paramétereiben, ekkor a Registrar a 403 – Wrong Password választ adja és meg kell ismételni a regisztrációs folyamatot Rossz Digest Authentication Response értékkel töltjük fel az Authorization Header struktúrát, ekkor a Registrar a 401 – nonce has changed választ adja és meg kell ismételni a regisztrációs folyamatot
	UAC DISCOVERY
P_I_05	<p>UAC-A küld egy INVITE üzenetet UAC-B-nek, akkor UAC-B RINGING 180 üzenetet küld vissza, ekkor UAC-A</p> <ul style="list-style-type: none"> Várákozik UAC-B 200 OK üzenetére UAC-B TEMPORARILY UNAVAILABLE 480 üzenetére CANCEL üzenetet küld UAC-B-nek <p>486 BUSY HERE üzenetet küld vissza</p>
P_I_06	Ha UAC-A és UAC-B közötti 2 irányú RTP Média csatorna felállása után UAC-B IP címet vált, akkor UAC-B egy (RE)INVITE üzenetet küld közvetlenül UAC-A-nak, amire UAC-A 200 OK-al válaszol, amire UAC-B ACK-al válaszol, így újra létrejön az RTP media stream.
P_I_07	Ha UAC-A és UAC-B közötti 2 irányú RTP Média csatorna felállása után UAC-B IP címet vált, akkor UAC-B egy (RE)INVITE üzenetet küld közvetlenül UAC-A-nak, amire UAC-A OK 200-zal válaszol, amire UAC-B ACK-val válaszol, így újra létrejön az RTP media stream kommunikáció rfc3665 - 3.2
	UAC -> PROXY
P_I_08	<p>UAC-A PROXY 1-nek küld egy INVITE Request-et, ekkor PROXY 1</p> <ul style="list-style-type: none"> továbbítja UAC-A INVITE Requestjét. AUTHORIZATION 407 REQUIRED üzenetet küld vissza UAC-A-nak. Elérhető, de nem válaszol, ilyenkor UAC-A a konfigurációjában következő PROXY-nak próbálja meg elküldeni az INVITE Request-et.
	PROXY -> PROXY
P_I_09	<p>Ha PROXY 1 PROXY 2-NEK küld egy INVITE Request-et akkor PROXY 2</p> <ul style="list-style-type: none"> továbbítja az INVITE Request-et UAC-B-nek, és TRYING 100 response-t küld vissza PROXY 1-nek.

	<ul style="list-style-type: none"> AUTHORIZATION 407 REQUIRED üzenetet küld vissza PROXY 1-nek.
	PROXY → UA
P_I_10	PROXY 2 küld egy INVITE üzenetet UAC-B-nek, akkor UAC-B <ul style="list-style-type: none"> 180 RINGING üzenetet küld. nem válaszol, ekkor PROXY 2 6-szor ismétli meg az INVITE üzenetet, majd TEMPORARILY UNAVAILABLE 480 üzenetet küld vissza annak akitől kapta az INVITE üzenetet
	UAC → RS
P_I_11	Ha UAC-A RS-nek INVITE üzenet küld UAC-B-nek címzett header-rel, akkor RS egy Moved Temporarily 307 választ küld UAC-A-nak, UAC-B SIP címével a header contact részében.

2.6.2. USER AGENT CLIENT

Azonosító	Kifejtés
P_I_UAC_01	UAC INVITE kérésnek az RFC 3665 - 8.1.1 -es formában leírt követelmények kell megfelelnie. Kezdeti INVITE kérés tartalmazza rfc3261 – 13.2 <ul style="list-style-type: none"> Allow fejléc mezőt: metódusok, amelyeket UAC fel tud dolgozni. rfc3261 – 20.6 Supported fejléc mezőt: kiegészítések, amelyeket UAC fel tud dolgozni. rfc3261 -20.37
P_I_UAC_02	Amennyiben az INVITE kérés tartalmazza a munkamenet leírását, speciális szabály van az üzenet törzsére: <ul style="list-style-type: none"> INVITE több dialógusból épül fel, melyben mindegyik egy offer/answer pár offer után mindig answer-nek kell következnie a kontextus dialógusban csakis INVITE kérésben, INVITE válaszbán, illetve ACK-ban jelenhet meg kezdeti offer: <ul style="list-style-type: none"> → = UAC által küldött INVITE: az answer egy UAS által UAC-nek INVITE-ra küldött 2xx válasza, UAC az első answer-t dolgozza fel, amely az kezdeti INVITE-ra érkezett → = UAS által küldött 2xx: az answer egy UAC által UAS-nek küldött ACK kezdeti offer után UAC több utólagos request offer-t is indíthat, de csak akkor, ha az előtte lévő offer-ekre kapott answer-t és nem küldött olyan offer-t amelyre nem kapott választ UAS nem generálhat utólagos offer-t, amíg az kezdeti tranzakció be nem fejeződött. összes UAC-nek tartalmaznia kell SDP-t

P_I_UAC_04_1xx	Több ideiglenes válasz is érkezhethet mielőtt a végleges beérkezik. Az ideiglenes üzenetek korai dialógust tudnak indítani. A korai dialógusokra akkor lehet szükség, mikor az UAC-nek kérést kell küldenie, mielőtt az kezdeti INVITE befejeződött volna. Ideiglenes válaszban lévő fejlécmezők mindaddig alkalmazhatóak, amíg a dialógus korai fázisban van.
P_I_UAC_04_2xx	Köszönhetően az elágazó proxynak egyetlen INVITE-ra több 2xx válasz üzenet is érkezhethet UAC-hoz. A To fejléc mezőben található címkeparaméterek különböztetik meg a küldőket, külön párbeszédpanellel és párbeszédpanel-azonosítóval.
P_I_UAC_04_3xx	Tartalmazhat: Contact fejléc mezőt: új címeteket, ahol a hívott fél elérhető lehet. 3xx válasz státusz kódjától függően UAC dönthet úgy, hogy az új címeteket próbálja meg elérni.
P_I_UAC_04_4xx_5xx_6xx	Tartalmazhat: Contact fejlécmezőt: Azt a helyet jelzi, ahol további információk találhatóak a hibáról. A későbbi végső válaszokat figyelmen kívül kell hagyni. Miután RFC3261- 13.2.2.3 válaszok valamelyikét kapja a UAC az INVITE-ot befejezettnek tekinti.

2.6.3. USER AGENT SERVER

Azonosító	Kifejtés
P_I_UAS_01	Amikor a UAS INVITE Requestet kap a hálózaton, akkor a request feldolgozását az RFC 3261 8.2 szekcióban meghatározott módon kezdi el (ami az RFC 3261 12. szekciójában leírja, hogyan jönnek létre a dialógusok). Az INVITE Request jellegétől függően több dolog történhet: <ol style="list-style-type: none"> 1. Ha a kapott INVITE Request fejlécében megtalálható az "Expires" mező, a UAS beállít egy visszaszámlálást a mezőben található időtartamra. <ul style="list-style-type: none"> • Ha ez az időzítő lejár, akkor a UAS egy Request Terminated 487 választ küld a Request feladójának. 2. Ha a Request egy dialógus közben érkező kérés, akkor a Request metódustól független feldolgozását az RFC 3261 12.2.2 szekció írja le részletesen. 3. Ha a Request fejlécének "To" mezőjében található tag által definiált dialógus nem létezik, akkor a szerver az RFC 3261 12.2.2 szekcióban leírt utasítások szerint jár el.
P_I_UAS_02	Ha a UAS által kapott INVITE Request tartalmaz Session Description-t, és a UAS egy adott UAC-nek címezve több ilyen request-et kap, és ha: <ol style="list-style-type: none"> 1. Az adott User már tagja a session-nek ahonnan az INVITE Request származik (origin mezője a Session Description Protocol-nak) 2. "Session" paraméterek nem változtak az 1-es opcióban említett session paramétereikhez képest

	Akkor a UAS küldhet 2xx üzenetet vissza az INVITE Request feladójának anélkül, hogy azt továbbítaná a címzett UAC-nek.
P_I_UAS_03	Ha a UAS által kapott INVITE Request nem tartalmaz Session Description-t, akkor a UAS-nek kínálnia kell egy új session-t a küldő UAC-nek az első non-failure, reliable üzenetében.
P_I_UAS_04	<p>A UAS a következő válaszokat küldheti az INVITE Request-ekre:</p> <ul style="list-style-type: none"> • Progress • Accept • Redirect • Reject <p>A listában felsorolt válaszok részletezését az RFC 3261 8.2.6 szekciója tartalmazza.</p>
P_I_UAS_05	Ha a UAS nem tud egyből válaszolni az INVITE Request-re, akkor valamilyen folyamatban levést jelezve "provisional" válaszokat tud küldeni az UAC-nek, amik 101 és 199 közötti kódszámmal vannak ellátva.
P_I_UAS_06	A Provisional válaszokhoz kapcsolódó procedúrák az RFC 3261 8.2.6-ban és a 12.1.1-ben vannak részletesen kifejtve.
P_I_UAS_07	<p>A UAS Tetszőleges Provisional választ küldhet ki egy adott INVITE Request-re válaszolva.</p> <p>Az ehhez az INVITE-hoz tartozó Provisional válaszok dialógus azonosítójának meg kell egyeznie.</p>
P_I_UAS_08	Ha a UAS meg szeretné hosszabbítani az INVITE Request válaszolására kapott időt, akkor hosszabbítást kell kérnie a Proxy UAS-ektól, hogy ne legyen lezárva az INVITE tranzakció.
P_I_UAS_07	A UAS Proxy érvénytelenítheti az INVITE tranzakciót, ha 3, vagy annál több perces szünetek vannak az adott INVITE tranzakcióhoz tartozó válaszokban.
P_I_UAS_08	Egy INVITE tranzakció tovább is élhet, ha az UAC on-hold-ra kerül, vagy ha a rendszer engedélyezi, hogy a kommunikáció a két végpont között úgy is létrejöjjön, hogy a címzett UAC nem válaszol az INVITE-ra.
P_I_UAS_09	<p>Ha a UAS átirányítja a hívást, akkor 3xx választ küld. rfc3261 -13.3.1.2</p> <p>A</p> <ol style="list-style-type: none"> 1. 300 (Multiple Choices) 2. 301 (Moved Permanently) 3. 302 (Moved Temporarily) <p>üzeneteknél a válasznak tartalmaznia kell a Contact fejlécben egy vagy több címet, ahová a hívó az INVITE Request-et küldheti.</p>
P_I_UAS_10	Ha az INVITE Request címében szereplő UAC elérhető, de nem képes, vagy hajlandó hívást fogadni, akkor ez az UAC Busy Here 486 választ fog küldeni.
P_I_UAS_11	Ha egy proxy UAS kap egy INVITE Requestet, és tudja, hogy a hívottak nem képesek hívást fogadni, akkor egy Busy Everywhere 600 választ küld a hívónak.
P_I_UAS_12	Egy UAS elutasíthat egy SDP Offert egy INVITE Request-ben, ekkor a UAS egy 488 Not Acceptable Here üzenetet küld, a fejlécben egy "warning" mezővel, ami az elutasítás okát tartalmazza.
P_I_UAS_13	Egy UAS elfogadhat egy INVITE Requestet egy 2xx válasszal.

	Ez a válasz egy dialógust állít fel, aminek a procedúrája az RFC 3261 8.2.6 -ban és 12.1.1 -ben van kifejtve.
P_I_UAS_14	Egy UAS által INVITE Requestre küldött 2xx válasz fejléce tartalmazza 3. az Allow mezőt 4. a Supported mezőt
P_I_UAS_15	Ha az INVITE Request egy SDP Offert tartalmazott, akkor a UAS első 2xx válaszában tartalmaznia kell egy SDP answer-t. Ha az UAC INVITE Request-je nem tartalmazott SDP Offert, akkor a UAS első 2xx válaszában tartalmaznia kell egy SDP Offer-t.
P_I_UAS_16	Egy UAS egy INVITE Request-re válaszolva addig kísérli meg a 2xx válasz kiküldését, amíg egy ACK választ nem kap rá, a SIP alatt működő (UDP / TCP) protokolltól függetlenül. A UAS első 2xx válasza és annak újraküldése között T1 idő telik el, ez után az összes következő újraküldések között eltelt idő duplázódik, amíg ez az időtartam el nem éri T2-t. T1 és T2-re vonatkozó leírások az RFC 3261 17 -es szekciójában találhatók. Ha az újraküldések közötti idő eléri a $64 * T1$ időt ACK UAS-hez érkezése nélkül, a UAS BYE üzenetet küld ki. (Lásd RFC3261- 13.3.1.4)
P_I_UAS_17	Ha a UAS akkor kap egy második INVITE Requestet, mielőtt az ugyanahhoz a dialógushoz tartozó, alacsonyabb CSeq szekvenciaszámmal rendelkező első INVITE-ra végleges választ küldene, akkor a UAS-nek 500 Internal Server Error választ kell küldenie a második INVITE-ra. Ennek a válasznak a fejlécének tartalmaznia kell a Retry-After mezőt, ami egy randomgenerált szám 0 és 10 között.
P_I_UAS_18	Ha egy UAS további INVITE Request-eket kap ugyanazon a dialóguson, amelyiken egy INVITE Request-et már továbbított és erre az első INVITE Request-re "in progress" választ kapott, akkor a minden további INVITE Request-re 491 Request Pending választ kell küldenie.
P_I_UAS_19	Ha a UAS egy re-INVITE kérést kap, akkor vizsgálnia kell, hogy a kérésben változtak-e: <ul style="list-style-type: none"> • a verzió azonosítói a session description-nek, vagy ha nincs ilyen, akkor • a session descripton tartalma Ha igen, akkor a UAS-nak a session paraméterek változásaihoz kell igazodnia, miután megerősítést kért és kapott az INVITE-ot küldő UAC-tól. Ha az új session description nem elfogadható a UAS által, akkor a UAS 488 Not Acceptable Here választ küld a re-INVITE-ra, fejlécében egy kitöltött warning mezővel.
P_I_UAS_20	Ha a UAS egy 2xx üzenetet küld, amire sosem kap ACK-t, akkor egy BYE üzenetet kell generálnia, és terminálnia kell a dialógust.
P_I_UAS_21	A UAS dönthet úgy, hogy nem küld 180 ringing választ egy re-INVITE-ra.

P_I_UAS_22	<p>Ha a UAS egy SDP offer-t küld egy 2xx üzenettel egy re-INVITE-ra, akkor az SDP offer-t úgy kell megterveznie, mintha egy új hívást hozna létre, figyelembe véve a szabályokat, amik egy meglévő session módosítására vonatkoznak. Ezeket az RFC 3261 13-as szekciója írja le.</p> <p>A UAS-nek biztosítani kell, hogy a generált session description média formátumai, transzportjai és egyéb, a résztvevők számára releváns paraméterek közös metszete minél nagyobb halmaz legyen.</p>
-------------------	---

2.6.4. Capability teszt

A SIP hívások indításának fontos lépése, hogy megbizonyosodjunk arról, hogy a két fél között létrehozni kívánt session képes olyan tartalmak átvitelére, amelyet mindkét fél értelmezni tud. Ebben segítségünkre van a Session Description Protocol, röviden SDP, aminek segítségével strukturáltan tudjuk elküldeni a hívott félnek azokat a paramétereket, amelyeket mi támogatunk, és így ő el tudja dönteni, hogy melyikkel szeretné azok közül indítani a hívást. Természetesen előfordulhat, hogy a hívott fél ezek közül egyiket sem támogatja, így azonban már az elején meg kell szakítani a sessiont, hiszen az nem lenne alkalmas média tartalmak, pl. hang vagy videó átvitelére.

Azonosító	Kifejtés
P_CT_01	<p>SIP request SDP-re vonatkozó részében lévő (RFC3261-6) session description-nek a megadott sorrendben</p> <ul style="list-style-type: none"> tartalmaznia kell: <ul style="list-style-type: none"> v – a protokoll verzióját leíró paraméter o – az originator, azaz a session kezdeményezőjének, valamint magának a sessionnek az azonosítója s – a session neve
P_CT_02	<p>SIP request SDP-re vonatkozó részében lévő time description-nek a megadott sorrendben</p> <ul style="list-style-type: none"> tartalmaznia kell: <ul style="list-style-type: none"> t – azok az időpillanatok milliszekundumban, amelyek között a session aktív
P_CT_03	<p>SIP request SDP-re vonatkozó részében lévő RFC2327 - 5.1 media description -nek a megadott sorrendben</p> <ul style="list-style-type: none"> tartalmaznia kell: <ul style="list-style-type: none"> m – a média neve c – kapcsolattal kapcsolatos információk, azonban, ha ez meg van adva a session description-ben, akkor nem kötelező megadni

2.7. Teszteredmények és elemzések

A program a lefutott funkció tesztek kommunikációs logját elemzi, és amennyiben a generált teszt az elvártak szerinti válaszokat adta és a megadott időkereten belüli, akkor "megfelelt" ítéletet ad rá.

A funkció tesztek végeredményének sokaságát a teszt blokk egy órás időszelétére is kiszámítja, majd a teszt legvégén az összes funkcionális teszt eredményét összegzi, és az elfogadási paraméterek alapján ítéletet hoz a teszt sikerességéről, valamint az összesített adatokat is elmenti.

Azonosító	Kifejtés
TR_01	A teszt futtatása során az összes kommunikációs logot meg kell címkézni, hogy azok a paraméterek megfelelően vagy nem megfelelően futottak le.
TR_02	Az összes funkcionális tesztelés során lefutott sikeres és sikertelen tesztelés eredményeit összegezni kell, valamint ezen eredményekhez hozzá kell csatolva lennie a teszt kommunikációs logjainak és a tesztelési blokk adatainak.
TR_03	A programnak képesnek kell lennie egy funkció teszteléséhez az összes érintett eszköz kommunikációs adatainak monitorozására, mentésére és kiértékelésére időbélyeggel ellátva.
TR_04	A teszteredmények elemzést adnak: <ul style="list-style-type: none"> • Egyéni funkció tesztek eredményéről • Egy órás funkció tesztjeinek csoportos eredményeiről • A teljes teszt blokk lefutásának végeredményéről

3. Szójegyzék

3.1. SIP

- **UA:** User Agent Kezeli a SÍP üzeneteket. Küldi és Fogadja, valamint feldolgozza az üzeneteket.
- **UAC:** User Agent Client A eszköz oldali UA
- **UAC-A:** User Agent Client A
- **UAC-B:** User Agent Client B
- **UAS:** User Agent Server A szerver oldali UA
- **Session Description:** Jól definiált forma pont elegendő információ átadásához, hogy egy User Agent multimédia session-öket találjon, és részt vegyen bennük. [Lásd RFC 2327.](#)
- **RTP Média:** Hálózati protokoll, amelyen keresztül audiót és videót lehet közvetíteni IP címen keresztül
- **Call-ID:** globális, egyedi azonosítója egy session-nek.
- **SIP URI:** Uniform Resource Identifier (URI) séma a SIP protokolnak. Egy SIP Cím hasonló módon épül fel, mint egy e-mail cím, ami egyedi azonosítóként identifikál egy UA-t.
- **Proxy server/ UAS Proxy:** User Agent-ektől érkező SIP üzenetek feldolgozását menedzselő eszköz egy hálózatban.
- **Tranzakció:** Egymást kiváltó, különböző üzenetek sorozata. Lásd [RFC 3261 - 17](#)
- **Registrar:** A SIP protokoll belső név szervere, az **AOR** <-> eszköz **IP & Port** összerendeléseket tárolja egy kereshető formában. Egy **AOR**-hoz több **IP & Port** bejegyzés is tartozhat.
- **AOR:** Adress of Record. Egy adott **UA**-hoz tartozó rekord bejegyzés a **Registrar**-on.
- **Originator:** (kezdeményező) a session kezdeményezőjének, valamint magának a sessionnek az azonosítója
- **Offer:** jelzi a kommunikáció típusát (audió, videó, játék), paramétereit, illetve az válaszadó által küldött média címét
- **Answer:** jelzi az elfogadott kommunikációs típust, a paramétereit, illetve az kérő által küldött média címét
- **SDP (Session Description Protocol):** szabálykészlet arra vonatkozóan, hogy hogyan állíthatóak be a multimédiás munkamenetek, hogy az összes végpont hatékonyan részt vehessen a munkamenetben
 - **Session Description:** az SDP request azon része, amely általános információkat tartalmaz a session-ről, mint például a session-ben résztvevő user-ek listája és a session neve
 - **Time Description:** az SDP request azon része, amely a session-re vonatkozó idővel kapcsolatos információkat tartalmaz, mint például milyen időpontok között aktív a session
 - **Media Description:** az SDP request azon része, amely a session-ben használt média konfigurációját írja le, azaz audió vagy videó, illetve azokon belül milyen codec

3.2. Kommunikációs protokollok

- **TCP:** Transmission Control Protocol A TCP egy számítógépen futó program és egy másik számítógépen futó másik program között egy adatfolyam megbízható, sorrendhelyes átvitelét hivatott biztosítani.

- **UDP:** User Datagram Protocol Egy protokoll gyors üzenet alapú szolgáltatás biztosítása, célja a gyorsaság.
- **Happy path :** Legoptimálisabb út két hálózati végpont között a hálózaton

3.3. Hardver és szoftver

- **User-CPU-Time:** az az időtartam (ezredmásodpercben mérve), amit a processzor az alkalmazás kódjának futtatásával tölt
- **SSD meghajtó:** (*Solid-state drive*) Nagy sávszélességű, rövid válasz idejű adattárolási technológia, mozgó alkatrészek nélkül, amivel megvalósítható a projekt célkitűzése
- **HDD:** (Hard Disk Drive): Egy tárolási egység melyet nagy tárhatalom jellemzi, de nem szekvenciális írás és olvasás közben az írási és olvasási kapacitása jelentősen csökken.
- **DDR5/DDR4 memória:** Legújabb családú memória típusok, melyek a leggyorsabbak és a legkisebb késleltetéssel operáló memóriák közé tartoznak, valamint a nagy tárhatalom is el lehet velük érni.
- **RAID:** (Redundant Array of Independent Disks) Tárolási technológia, melynek segítségével az adatok elosztása egy vagy több fizikailag független adattárolón, egy logikai lemez létrehozásával lehetséges. Használata növeli az adattárolás megbízhatóságát.
- **LTS:** (Long Term Support - hosszú ideig támogatott) legalább 5 évig támogatott szoftververzió az Ubuntu operációs rendszer készítőitől

3.4. Program komponensek

- **Tesztelési blokk:** egy teszteset csoport, amik definiálva vannak az eszközök típusai, az eszközök között elvégzendő funkció tesztek száma, hibaaránya és ezen tesztek elfogadási paraméterei szerint.
- **Tesztelő modul:** Tesztelési blokkokat futtat, egyidőben maximálisan 1000 darabot. Biztosítja, hogy ne indulhasson el olyan teszt, ami a maximális eszközszámot elérné.
- **Eszköz foglaltsági lista:** Tesztelési modul része, amiben a program tárolja az összes, már regisztrált eszközt, amin lehet további funkció tesztek futtatni. Biztosítja, hogy 1 eszközt csak 1 funkció tesztelő blokk teszteljen.
- **Funkció tesztelési blokk:** A tesztelő blokk futása alatt jön létre minden egyes funkció hívás esetében. Emulálja a *szabad eszközön a szabványos és nem szabványos funkció hívás kezdeményezését, és az összes kimeneti és bemeneti értéket rögzíti és elemzi. *Kivétel a regisztrációs funkció; ekkor még nincsen regisztrálva az eszköz a foglaltsági táblában.
- **Elsődleges eszköz:** a funkció tesztelésnél mindig van egy eszköz, ami kezdeményezi a funkciót
- **Másodlagos eszközök:** azon eszközök, akik csak az elsődleges eszköz hívására reagálva válaszolnak