

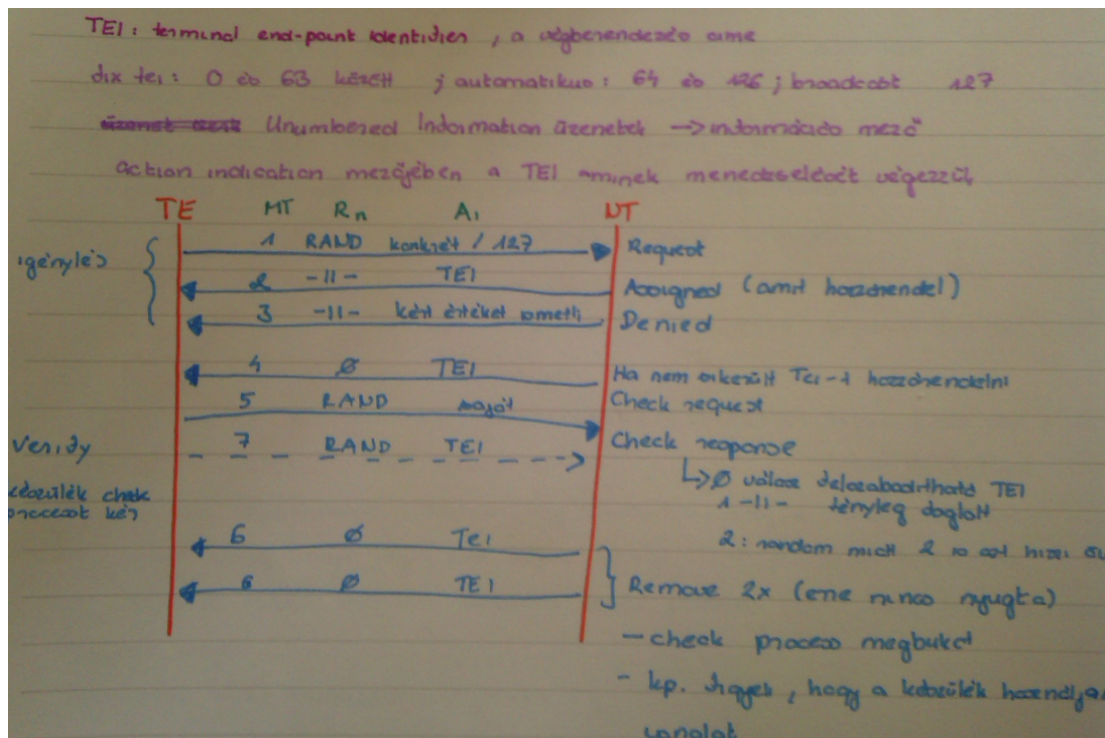
1. Rész ZH feladatai

Mivel általában a vizsga és a ZH ~70%-os átfedésben van egymással, ezeket is kell tudni.

Az összeállítás sok ember nevéhez fűződik, főleg Kuthi, Vincze, Valyon, Kiss, és a Wiki összes nagyörege a közreműködők.

Most vizsgán jelezte a Tanár úr, hogy ismeri a kidolgozásokat (wikiseket) szóval a jövő hetiek is készüljenek arra, hogy lesznek módosítások.

1. Ismertesse a LAPD TEI menedzselési eljárást!



Az automatikus TEI (Terminal Equipment Identifier) menedzselésének 3 eljárása van:

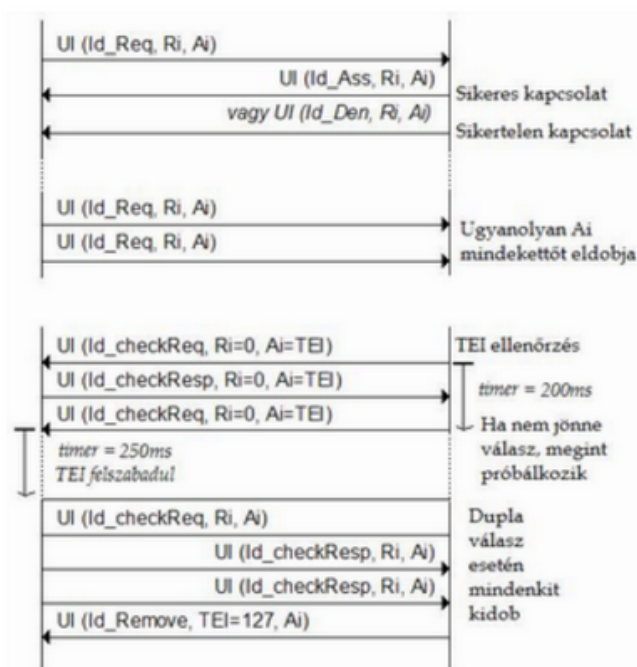
1. TEI hozzárendelés
2. TEI ellenőrzés
3. TEI visszavonás

A TEI menedzselés során UI (Unnumbered Information) kereteket használnak, melyek címmezőjében SAPI=63 és TEI=127 (broadcast) értékek vannak. A TEI menedzsmenet során 6 féle üzenettípust küldhetünk (Request, Assigned, Denied, Check Request/Response, Remove)

- A TEI hozzárendelést mindig a terminál oldal kezdeményezi egy **Request** üzenettel. Ebben a kérő oldal generál egy 16 bites véletlenszámot (RN), mely a kérés azonosítója lesz. A_i értéke vagy 127 vagy egy konkrét szám lesz. Ha a hozzárendelés teljesíthető, a hálózat egy **Assigned** üzenettel válaszol. Ha nem teljesíthető, a hálózat egy **Denied** üzenettel válaszol. → vagy nem adható ki mert már használt, vagy egyáltalán nincs szabad TEI

szám. Ha egyáltalán nincs szabad, akkor a hálózat ellenőrzi a TEI számokat → TEI Ellenőrzés

- TEI ellenőrzés során a hálózat ellenőrzi, hogy tényleg használatban van-e minden kiosztott TEI érték. A TEI ellenőrzést mindig a hálózati oldal kezdeményezi egy **Check Request** üzenettel. Erre a Terminál válaszolhat, illetve minden Terminal beküldheti az általa birtokolt értéket. Ez a válasz egy **Check Response** üzenetben megy. Ha az ellenőrzés során kiderül, hogy egy TEI érték több Terminal-nál is használatban van, akkor TEI visszavonás indul
- TEI visszavonás során a hálózat egy **Remove** üzenettel elveszi a Terminal-tól a TEI értékét.



Csatlakozáskor küld egy kérést a felhasználói oldal (pl. egy laptop). Ha ebben a kérésben

- $A_i = 127$, akkor bármilyen TEI-t elfogad
- $A_i \neq 127$ -tel, akkor a küldött értéket kéri TEI-nek.

A TEI-ket a központ tartja nyilván. Ha ugyanolyan R_i -vel többen is kérnek TEI-t, akkor ezt a kérést a központ visszautasítja, de nem válaszol, hanem DROP-olja (eldobja) a kéréseket.

Ha valaki kihúzza az eszközét, attól még a TEI hozzá van rendelve. Ezért szükségesek a Check és Remove lépések. Egy átlagos központnál éjjel 2-3 között vannak a TEI ellenőrzések.

TEI Check Sequence

Az azonosító ellenőrzés kérés után ha nem jön válasz, akkor még egyszer megpróbálja a központ. Ha akkor sincs válasz, akkor eldobja az adott TEI-t.

Ha több válasz is jön ugyanazzal az azonosítóval, akkor el kell venni mindegyiktől azt. (Remove) (TEI=127-tel küld ekkor a központ – broadcast üzenet, hogy mindenki vegye)

2. Adjon példát egy beágyazott TCAP tranzakcióra! Jelezze a példán a használt üzenet- és komponens típusokat, tranzakció és műveleti azonosítókat! Hogyan tudunk a TCAP-ben hibát jelezni?

Egy TCAP üzenet 3 részből áll

- Tranzakciós rész (ez mindig benne van, a másik 2 kimaradhat) – azonosítja a tranzakciókat és hogy azon belül hol járunk
- Dialógus rész
- Komponens rész

Hol járunk a tranzakción belül?

A legegyszerűbb tranzakció áll egy BEGINből és egy ENDBől. CONTINUE rész: további üzenetek, minden, ami nem Begin/End (lehet üres is, csak hogy kilegyen az oda-vissza szekvencia).

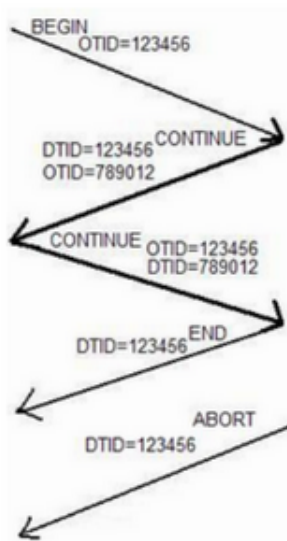
OTIP: Originating Transaction Identifier
(kezdeményezőoldali tr. azonosító)

DTID: Destination Transaction Identifier

ABORT-ot bármikor lehet küldeni, a tranzakció nem normális befejezésére utal.

A dialógus részt akkor használják, ha

- Nem egyértelmű milyen felhasználói üzenetek haladnak
 - Nem egyértelmű, milyen TCAP verziót használunk
- Egyedi 7byte-os azonosítót rendelnek minden TCAP verzióhoz, és az összes protokollhoz



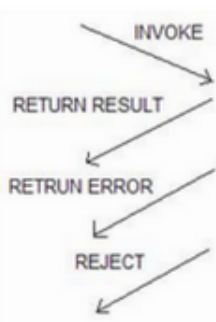
Komponens rész:

Azonosítja a műveletet, az ahhoz tartozó műveleti kódot és a paramétereket. Ezt a hármat együtt nevezzük komponensnek. Egy TCAPben lehet 0-1-több komponens is

INVOKE ID: tranzakción belül az üzenetek azonosítására; megkötés: egy tranzakción belül különböző számokat kell kapnia minden műveletnek.

Művelet

- kérés egy Invoke komponensben, azonosító és paraméterek
- műveleten belül 5 fázis:
 - Invoke
 - Return Result Not Last
 - Return Result Last
 - Return Error
 - Reject

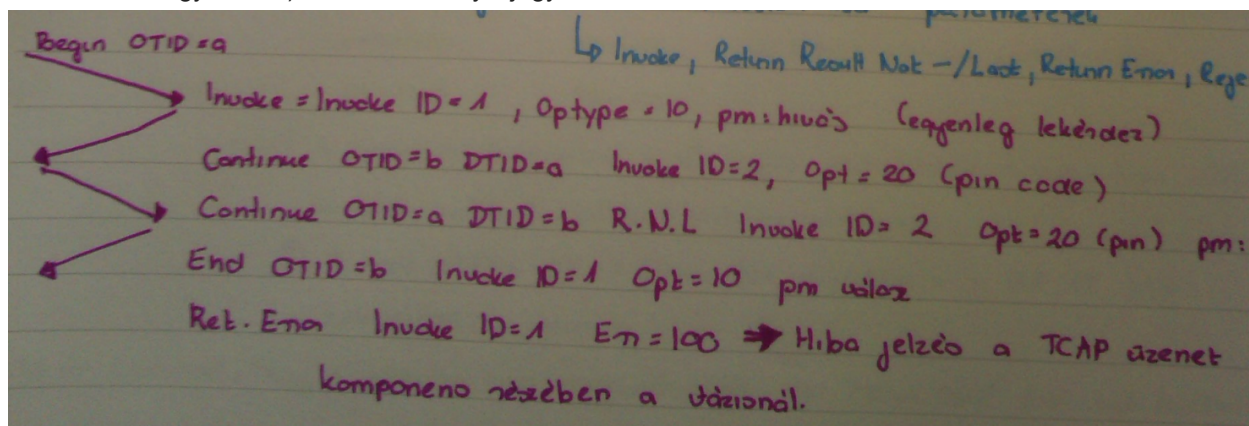


Egyszerű begin-endből álló, egy műveletet indító tranzakció esetében Begin üzenet Invocation komponens, míg End üzenet Return Result Last vagy return error komponens tartalmazhat. ARGUMENT kulcsszó után specifikált üzenetszerkezet használható Invoke komponensekben RESULT kulcsszó után specifikált üzenetszerkezet használható Return Result komponensekben ERROR kulcsszó után specifikált üzenetszerkezet használható Return Error komponensekben.

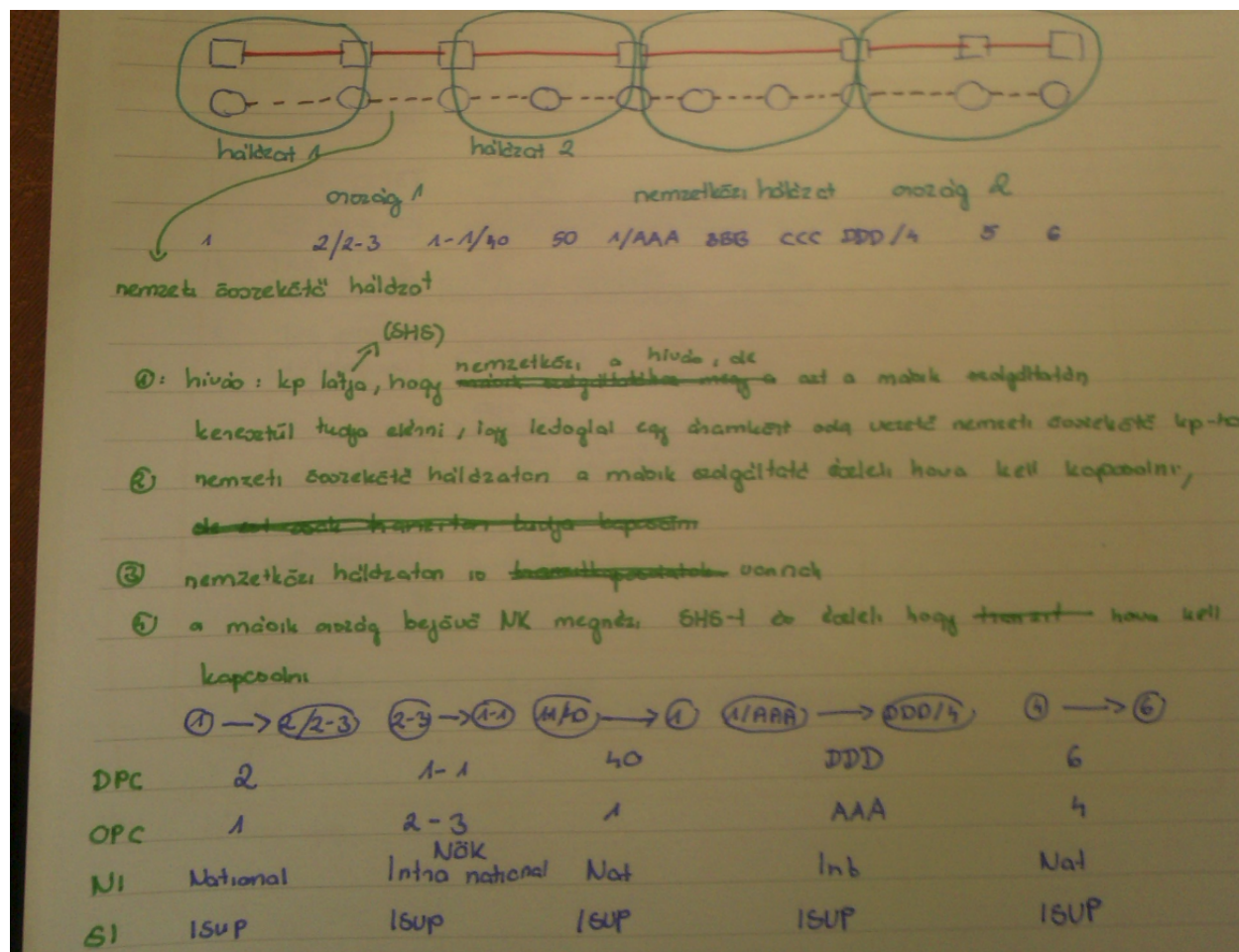
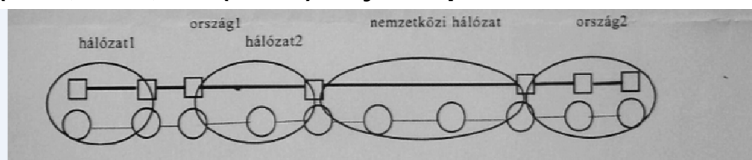
Hibajelzés:

1. TCAP Abort üzenet, ha pl. nem ismerjük fel magát a TCAP üzenetet, olyan típusú vagy verziójú beágyazott protokoll használatát kérték a dialógus részben, ami nincs implementálva, szoftverhiba, stb.
2. Reject komponens: azt jelzi, hogy a TCAP üzenetet felismertük, de a beágyazott protokoll üzenete értelmezhetetlen, például nem létező vagy rossz tranzakció vagy invokáció azonosító, nem létező művelet, értelmezhetetlen vagy hiányzó paraméter stb. van a kérdésben vagy a válaszban.
3. Return Error komponens: megértettük és végrehajtottuk a műveletet, de a művelet hibával végződött. Ez művelet függő, pl. nem létező előfizető adatait kértük le, vagy rossz PIN kódot adta meg, stb.

ht: "TCAP: kevés, érdemes odaírni, hogy adott tcap üzenetben mi a komponens azonosító (pl. Invoke ID 1), mi a kért művelet (pl. egyenleg lekérdezés, PIN kérés), és még valami paraméter (hogy ez most épp hívás/kérés, vagy válasz). Talán valamelyik jegyzet scanben benne van szebben."

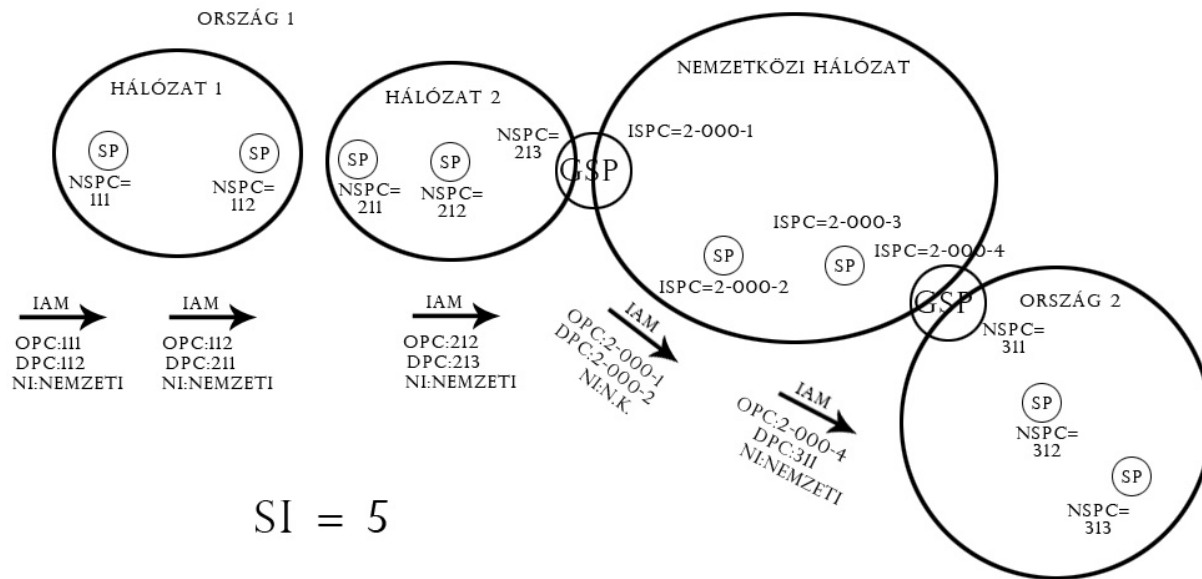


3. Egy nemzetközi hívás felépítése. pl.: meg kell adni a legfontosabb paramétereket (DPC, OPC, SIO(NI+SI) és jelzés-pont-kódokat kell az egyes szakaszokhoz rendelni.



Hibás a megoldás, a 4 és 6-os között van még egy megálló, a helyes megoldás 4->5; 5->6.
 "Ahány beszédáramkör, annyi jelzés"

egy másik megoldás példa:



Z

ht: "Nemzetközi hívásfelépítésnél érdemes arra figyelni, hogy látszódjon is, hogy a beszédút mindig egy-egy központ között épül fel, jelzéspontot csak érint. Tehát nem egy egyenes nyíl végig, hanem nyilak köpontok között, jelzéspontnál esetleg ilyen kis puklival jelezni, hogy ott azért történik valami (továbbítás)."

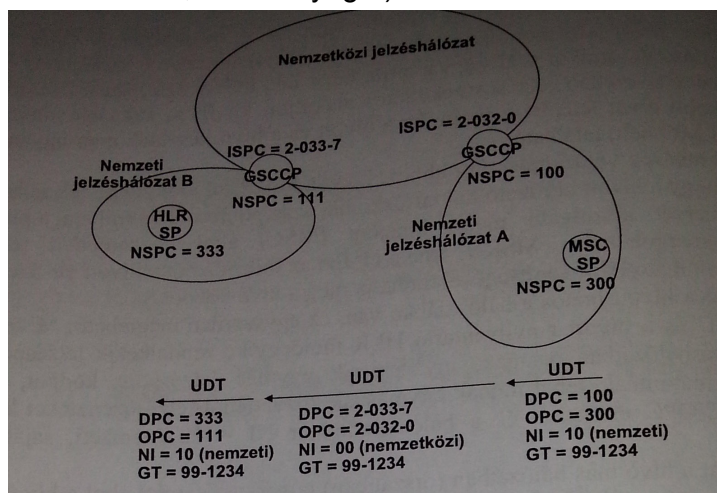
4. Mutassa be egy konkrét példa segítségével az SCCP globális címfordítását!

SCCP elméleti háttér (erre nem ad pontot, csak, hogy érthető legyen, könyv 105. oldal)

- Az ISUP az MTP-vel csak azonos jelzeshálózatokban tud jelzést átvinni, de sokszor különböző jelzeshálózatok között is át kell vinni, pl. mobil hívás ha a hívó és hívott nem ugyanabban a hálózatban tartózkodik (különböző jelzeshálózatbeli jelzésponthoz kell jelzésüzenetet átvinni)
- Bevezették ehhez a globális cím (GT - Global Title) fogalmát. A globális cím tetszés szerinti, nem az SS7-es jelzésrendszerben definiált, címként használt azonosító lehet
- SCCP (Signalling Connection Control Part) feladata a globális címen alapuló jelzésüzenet irányítást végezzen nem csak kapcsolatorientált, hanem kapcsolat nélküli szolgálatok számára is
- Meghatározza, hogy a globális címmel megcímezett egység felé merre kell jelzésüzeneteket irányítani → meghatározza a megcímezett egység jelzésponthoz kódját ha egy hálózatban vannak, ha pedig nem, akkor annak a Gateway SCCP-nek a kódját, amelyen keresztül lehet kijutni azon hálózat felé, amerre a globális címmel megcímezett egység található
- A jelzésponthoz elég azt ismerniük, hogy a globális cím analízisa után melyik kapu SCCP felé kell irányítaniuk a hálózaton belül az üzenetet, illetve a kapu SCCP-knek kell csak ismerniük a két szomszédos jelzeshálózat jellemzőit, de nekik sem kell tovább, más hálózatokba "előre látniuk"
- Az SCCP 2 féle szolgáltatást tud nyújtani, kapcsolat nélküli illetve kapcsolatorientált

Konkrét példán bemutatott működés (tankönyv 106. oldal, 73. ábra)

Amiből szerintem maga az ábra kell, és alatta az, hogy milyen üzenetek mennek. (zh-n is azt mondta, hogy ne írjunk sok rizsát, csak lényegét)



Ismertesse az SCCP protokoll helyét, szerepét, legfontosabb üzeneteit!

- lásd: fentebb

- *Legfontosabb üzenetek (szerintem): CR (Connection Request), CREF (Connection Refused), CC (Connection Confirmed), AK (Data Acknowledgement), RLSD (Released), RLC (Release Complete)*

5. PCM keretek és felépítéseik! Itt általában az vagy az EU vagy USA szabványt kérdezik.

Európai változat (A-LAW)

- Későbbi fejlesztés mint a mű karakterisztika.
- Jobb hangminőséget ad
- Főként az ipar védelme érdekében fejlesztették eltérőre
- 125us egy keret, ez 32 időrésre van felosztva
- keret
 - 0. időrés felváltva szinkron (SYN) és szervíz (SERV) csatorna
 - szinkron egy speciális bit sorozat (00110011)
 - szervíz 2. bitje (0-tól számozva) Alarm bit, amit ellenőrzésre használnak
 - 16. időrés jelzés csatorna (SIGNAL)
 - jelzi, hogy adott beszéd csatornákon mik változtak (beszéd indul, beszéd vége, stb.)
 - a többi időrés beszéd csatorna
 - 1,16 időrés szinkronizációra
- $32 * 64 \text{ kbps} = 2 \text{ mbps}$
- multikeret
 - 16 keretet fognak össze
 - a 0. csatornák felváltva szinkron és szervíz csatornák a fentiek alapján
 - a 16. csatornák jelzés csatornák a fentiek alapján
 - a 0. keret jelzés csatornája egy multikeret kódszó (1x10000)
 - az 1. keret jelzés csatornája az 1. és a 17. beszéd csatorna jelzés infója
 - a 2. keret jelzés csatornája a 2. és a 18. beszéd csatorna jelzés infója
 - ...
 - a 15. keret jelzés csatornája a 15. és a 31. beszéd csatorna jelzés infója
 - ha egy beszéd csatornán vége van a jelzésnek, akkor e miatt max egy keret idő múlik el, mire jelezni tudjuk, ami ~2ms (legrosszabb esetben a teljes multikeret, ami 16*125us) és ez még belefér a késleltetésbe)
- védelem
 - megj.: hiba javítás nem is igazán szükséges, mivel csak 10-20 keret felett hallja meg az emberi fül
 - az európai keret nem túl erős védelmi információval rendelkezik (hibajelzés)
 - paritás jellegű
 - 8 keretet fognak össze: SubMultiFrame
 - generálnak egy 4 bites CRC paritás információt
 - ezt a 4 bitet küldik a szinkron csatorna legelső bitjén (8 keretben 4 szinkron és 4 szervíz csatorna van)
 - egészen pontosan az előző 8 keretből generált 4 bitet a következő 8 keret 4 szinkron csatornáján küldik
- adatátvitel
 - unrestricted for digital signaling

Amerikai változat (u-LAW)

- 24 csatorna egymás mellett egy keret
- mindegyik csatorna beszédet visz át
- multikeret
 - 12 keretet fognak össze
 - mindegyik keretet egy bittel egészítik ki
 - FAW: Frame Alignment Word 101010 [A][B][C][0][1][0]
 - MAW: Multiframe Alignment Word 001110 [D][E][F]
 - ide jön, hogy mi hova jön:
 - 1 [A]
 - 2 [D]
 - 3 [B]
 - 4 [E]
 - 5 [C]
 - 6 [F]
 - ...
 - 12 []
 - 6. és 12. keret minden időrészéből 'lelopnak' egy bitet
 - az utolsó bit "agyon van ütve" egy jelzőbittel
 - emiatt minden hatodik hangminta rosszabb
- adatátvitel
 - restricted for digital signaling, mivel minden 6. bit elromlik
 - nem véletlen, hogy 7 bitesek a korai átviteli cuccok

Ismertesse az európai PCM struktúrát!

ISDN PRA (Primary Rate Access) az alközpontok illesztése a nyilvános hálózathoz

2 verzió terjedt el, Európai és az Amerikai . Az Európai jobb minőségű, de ára van.

Európai felépítése: 30+2 csatornás PCM (Pulse Code Modulation), ahol 30 adatcsatorna és 2 jelzés csatorna, minden időrés 8 bites. 1 keret 125 mikrosec hosszú, ezt felosztjuk 32 egyforma részre, melyet időrésnek hívunk (Channel Time Slot). Minden keretben a 0. és 16. időrésben jelzés információt viszünk át (signaling).

0	1	..	..	15	16	17	..	..	31
---	---	----	----	----	----	----	----	----	----

1 PCM csatorna 64 Kbps sávszélességű, ebből 32 db, így $32 \cdot 64 \text{ Kbps} = 2048 \text{ Kbps} = 2.048 \text{ Mbps}$

16 db ilyen keretet (Frame) összefogva egy MultiKeretet kapunk (MultiFrame), amelynek a hossza (időtartama) 2 ms, mivel $16 \cdot 125 \text{ us} = 2000 \text{ us} = 2 \text{ ms}$. Egy ilyen multikeret együttesen látja el a csatornák vezérlését, mivel keretenként 2 csatornára vonatkozó jelzést tudunk átvenni, így 16 kerettel 32 csatornára vonatkozó jelzést tudunk átvinni, és így az összes, 32 csatornához tartozó jelzést átvittük.

Az Európai szabvány még bithiba jelzésre is alkalmas egy 4 bites CRC-vel (Cyclic Redundancy Code), amelyet a Multikeret felére, azaz 8 db keretre generálunk (Sub-Multiframe → a multikeret $2 \cdot 8$ alkeretre van bontva). Ennek nem a beszédjel javítása a cél, hanem a vonal minőségének a jelzése, ellenőrzése, mivel ha sokat kell javítani, akkor rossz a vonal minősége, ha pedig hibátlanul átjön, akkor megfelelő.

Ismertesse az amerikai PCM struktúrát!

Az Európai szabványnál egyszerűbb. 24 csatornás felépítésű, mindegyik csatorna beszéd csatorna, így nincs dedikált jelzésátviteli csatorna. 1 keret itt is 125 us hosszú, ezt osztották fel 24 egyenlő részre. 1 Multikeret az 12 keretből áll, így $12 \cdot 125 \text{ us} = 1500 \text{ us} = 1.5 \text{ ms}$ hosszú. Sávszélessége $24 \cdot 64 \text{ Kbps} = 1536 \text{ Kbps} = 1.536 \text{ Mbps}$. 1 keret hossza 193 bit, amely 192 bit adat + 1 bit jelzés, amely szinkronkódzó.

A jelzésátvitel úgy történik, hogy minden 6. és 12. keretben "lecsípi" az utolsó hang bitet, és itt visznek át jelzéseket. Ezért minden 6. mintának a pontossága fele akkora mint a többié.

Beszélgetésnél ez nem probléma, mivel az emberi fül gyakorlatilag nem veszi észre, de az adatátvitel ilyen formában megvalósíthatatlan → Megoldás adatátvitelre: minden 6. csatornát ki kell hagyni, így megoldható, de nem hatékony, rossz kihasználtságú.

	EU	USA
1 keret hossza	125 mikrosec	125 mikrosec
Csatornák száma	30adat+2jelzés=32	24, nincs jelzés
Csatonra kódolása	PCM	PCM
Csatorna sávszélessége	64 kbps	64 kbps
Keret sávszélessége	$32 \cdot 64 = 2.048$ mbps	$24 \cdot 64 = 1.536$ mbps
Multikeret méret	16 keret	12 keret
Multikeret hossz	$16 \cdot 125 = 2$ ms	$12 \cdot 125 = 1.5$ ms
Hibajelzés	van, 4+4 bit, 16+16 csat.-ra	nincs
Jelzésinfó	0. és 16. csatorna	6. 12. keretek fele lecsípve

6. Mi az (országos szintű) nyílt és zárt számozási rendszer? hasonlítsa őket össze!

*NYÍLT: egy hívószám (HÁLÓZATON BELÜL) 2-ik alakja is van.
pl.: 20/346 853 és 0680 946833. RÖVID ÉS HOSZSÚ formátumok.
ZÁRT: egy hívószámmal EGYSEGISEN csak 1. formája van.*

Honnan tudja a központ, ha megérkezett az utolsó számjegy?

Ez a nyílt, vagy zár számozási rendszertől függ.

Zárt számozás: Ugyan annyi számjegyből áll, vagy legalábbis az első néhány számjegy beolvasása után lehet tudni hogy mennyit fog kapni. (ha egyes kezdődik akkor 5 hosszú, stb.) Lényeg: tudjuk hány szám lesz.

Nyílt számozás: nem tudjuk mennyi számjegyet fogunk kapni. (szálloda: 1-es a porta, 102- a 102-es szoba)

Azokon a pontokon, ahol vége is lehetne a számnak, el kell indítani egy timert. Ha lejárt, akkor az aszám, ha meg nyomunk még valamit, akkor nyilván várunk tovább.

Magyarországon ilyen értelemben **zárt**. (előtag illetve központ meghatározza az számot: budapeste elég 6, de 06-tal meg nyolc, stb.)

Fontos megjegyezni, hogy többértelmű ez a fogalom:

Ha egy országban nyílt a számozás, akkor ez azt jelenti, hogy sok számnak **több alakja van**. (előfizetői szám ha saját hálózaton belül van akkor nem kell a 0670 például)

Országkód: ha külföldre akarok tárcsázni	SHS: szolgáltatás, vagy hálózatkijelölés (körzetszám)	Előfizetői szám
--	---	-----------------

National prefix: 06. mindenhol magyarországon kívül "0". Csak nálunk nem...belföldi prefix.

A mobilszámok esetében zárták a számrendszert nálunk, így felemás a cucc itthon.

Változások a magyar számozásban

- 2010. jan. 15.: a mobilszámmező „lezárása”
 - Mobil számok csak a hálózatkijelölő számmal együtt hívhatók
 - Pl.: 0620-555-1234 vagy +3620-555-1234
 - azaz 555-1234 formában még Telenoros telefonról sem
 - Fő ok: a számhordozás után nem volt egyértelmű, hogy pusztán az előfizetői szám hívása melyik hálózatra vonatkozik: sok téves hívás volt
 - igazából az új hálózatra
 - pl.: 0670-555-6789 elmegy a Vodafone-tól a T-Mobile-hoz
 - a telefonjában elmentett bejegyzés: „Béla: 555-3333”
 - Béla száma 0670-555-3333
 - a számhordozás után véletlenül Béla helyett a 0630-555-3333-at hívta
 - Pozitív hozadék: kiadhatók a 0-val, 1-gyel kezdődő számok, 25%-os növekedés a számtartományban
 - pl: +3630-012-3456, 0670-123-4567
 - Ajánlott a +36205551234 formátumban tárolni a számokat
 - így külföldről is használhatóak

Számozás

□ Nyílt számozási rendszer:

- Nem kell mindig a belföldi rendeltetési számot tárcsázni, pl. Budapesten csak 7 jegy elég
- Sok esetben rövidebb a hívott szám
- De nem egyértelmű, máshonnan másképp kell (06 kell elé pl.)
- Ilyen a magyar hálózat

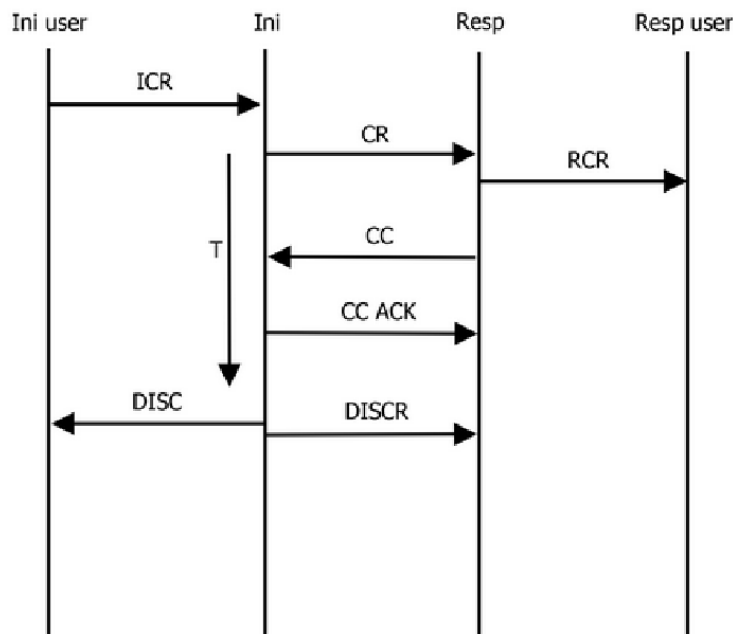
□ Zárt számozási rendszer:

- Mindig kell a belföldi rendeltetési szám
- Nem kell viszont a belföldi előtét (0, vagy Magyarországon 06)
- Egyszerű, egyértelmű
- Viszont nem lehet „rövidíteni” körzeten belül sem
 - De: tel. memóriájából tárcsázva nem gond
- Sok európai ország tért át erre (Norvégia, Franciaó., Olaszo., stb.)
 - Főleg az 1990-es évek közepétől, végétől

központon belüli számozási rendszer (just in case) _Az ITU-T E.164 telefonszám szabványa alapján létezik nyílt és zárt számozási rendszer.

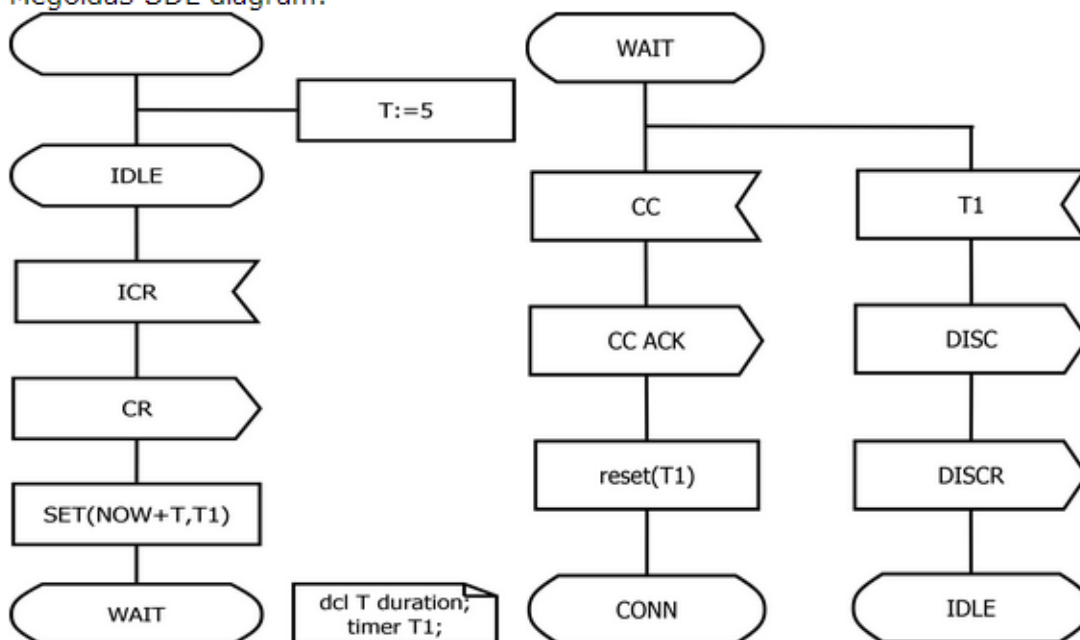
- Nyílt: a telefonszám hossza nem mondható meg azok elejéből, így timert használnak a vége ellenőrzésére. Körzeten belüli telefonáláshoz a rövidebb alak használata is elég pl. csak az előfizetői szám tárcsázása, de körzeten kívül szükséges az előtét- és körzetszám is. Bizonyos esetekben kevesebb számot is elég tárcsázni, így kényelmesebb, de a telefonközpont számára bonyolultabb a feldolgozás.
- Zárt: Előre vagy az első néhány számjegyből tudható, hogy milyen hosszú lesz a szám, mivel a számoknak csak egy előre rögzített alakja van, milyen típusú lesz az adott szám → számanalízist végez a telefonközpont. Mindig az egész, hosszabb számot tárcsázni kell, de a központ számára könnyebben és gyorsabban feldolgozható.

7. SDL feladat



Feladat értelmezése: Az ábrán szereplő folyamat Ini részét kell megvalósítanunk SDL nyelven. Az ICR jel hatására indul egy Timer (T1) és kiadjuk a CR jelet. Ha a timer lejártá előtt kapunk CC jelet, akkor kiadjuk a CC ACK-ot és átlépünk CONN állapotba. Ha a Timer lejár, akkor kiküldjük a DISC és DISCR üzeneteket.

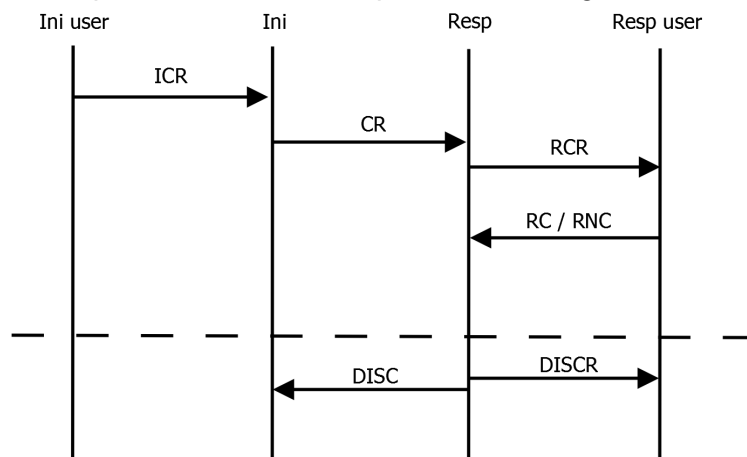
• Megoldás SDL diagram:



másik feladat

(wikiről: Az SDL diagramnál minden állapot kezdete után az első lépés egy bejövő üzenet kell, hogy legyen, ha ezt nem követed, de egyébként jó a logikád és amúgy működne is csak egyszerűsíteni akartál, akkor "ez nem SDL diagram" --> -7 pont a 10-ből.)

Valósítsa meg az ábrán látható "Resp" funkciót SDL diagrammal. A folyamatábrát egészítse ki szétkapcsolással és ezt is építse bele a diagrammba.

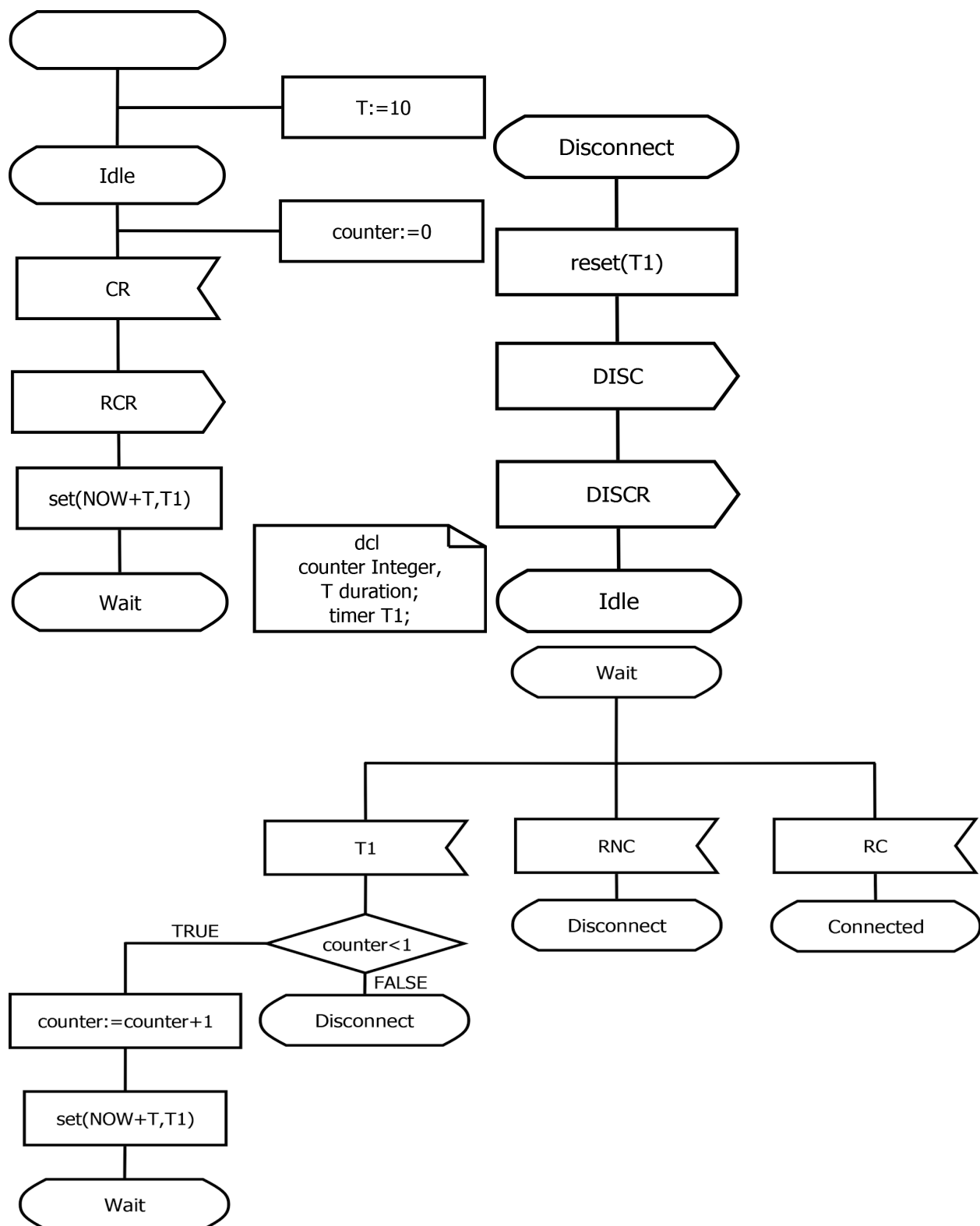


A feladat leírása: A vevő oldalon, egy CR üzenetre várunk, amely hatására kiadjuk az RCR üzenetet. Erre a válasz vagy egy RC vagy egy RNC üzenet lehet. Emellett a CR üzenet érkezésekor beállítunk egy timer-t, amely 10 másodperc múlva jár le. Az első time-out után tovább várunk, a másodiknál bontjuk a kapcsolatot.

Szintaktika:

- Start : üres krumplic az elején
- Állapot: ebben várakozik egy process amíg nem jön egy jel (pl. idle)
- "Jel": lehet in vagy out, téglalap, jobb szélén "nyíllal", input balra, output jobbra
- Időzítő: időzítő műveletek: időzítő indítása(SET) és leállítása (RESET). Indításkor a jelenhez (NOW) képest relatív időt lehet megadni

ht: "SDL-nél figyelni a kiinduló állapotra, első legyen egy bejövő jel, legyen változódeklaráció, jó helyen értékdás."



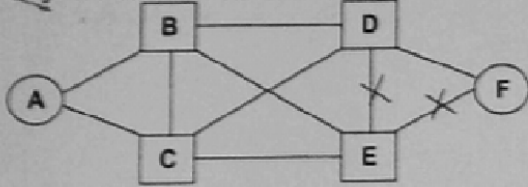
Erre kérdeztek rá most vizsgán megint, a disconnectet fejből kellett leírni.

Figyelem! Valszeg a disconnect állapot nem jó. Most vizsgán hasonló volt, 10ből 9 pontot kaptam arra, hogy a disconnect helyett leírtam a disconnectben kifejtett dolgokat, így nem szegtem meg az "inputtal kezdünk" szabályt. gondolom én

8. feladat

tk 86. oldal, érdemes átlapozni hozzá

2. Ismertesse, milyen lépésekben történik a jelzeshálózat átkonfigurálása, ha először az EF, majd az ED link hibásodik meg? Lásd az alábbi ábrát!



①

EF szakasz hibáját a 2. szint jelzése alapján vagy operátori parancsra jelölmen E. Majd átkapcsolási eljárást kezdeményez D-n keresztül az EPF útvonalra. F átkapcsolási eljárást D-n keresztül az FDE útvonalra utazva négyü kommunikáció esetében.

② E átviteli letiltást kezdeményez a D-be F irányába vonatkozóan. D ennek vétele után periodikusan jelzőútvonal kérelmet tesz üzeneteket küld E-nek F vonatkozásban.

③ DE normál esetben nem lenne át. E-ből D elérhetetlenné, D-ből E elérhetetlenné válik ezért átviteli letiltást küldenek az összes szomszédjuknak. Ezen üzenetek vétele a szomszédokban a megjelölt jelzőútvonal kérelmet tesz üzenetek küldéséből állja ki. E észreveszi, hogy számára F elérhetetlenné válik. Ekkor C-ben és B-ben kényszerített átirányítás hajtható végre. CEF útvonalról CDF-re, BEF útvonalról BDF-re vezet át. Az említett szakaszokról csak az F jelzőtábla útvonalat kell átirányítani.

1. szakasz:

1. E-F szakasz hibáját a 2. szint jelzése alapján vagy operátori parancsra felismeri F, és végrehajtja az átkapcsolási eljárást D keresztül küldött átkapcsolási eljárás segítségével az F-D-E útvonalra.
2. E is felismeri a hibát (vagy saját magától, vagy az F-től kapott átkapcsolás parancs vétele nyomán) és ő is végrehajtja az átkapcsolást a E-D-F útvonalra.
3. F az F-E forgalmát F-D-n keresztül, E az E-F fogalmát E-D-n keresztül újraindítja, ugyanakkor E átvitel letiltás üzenetet küld D-be F-re vonatkozóan.
4. D az átvitel letiltás vétele után periódikusan jelzésútvonal készlet teszt üzeneteket küld D E-nek F-re vonatkozóan.

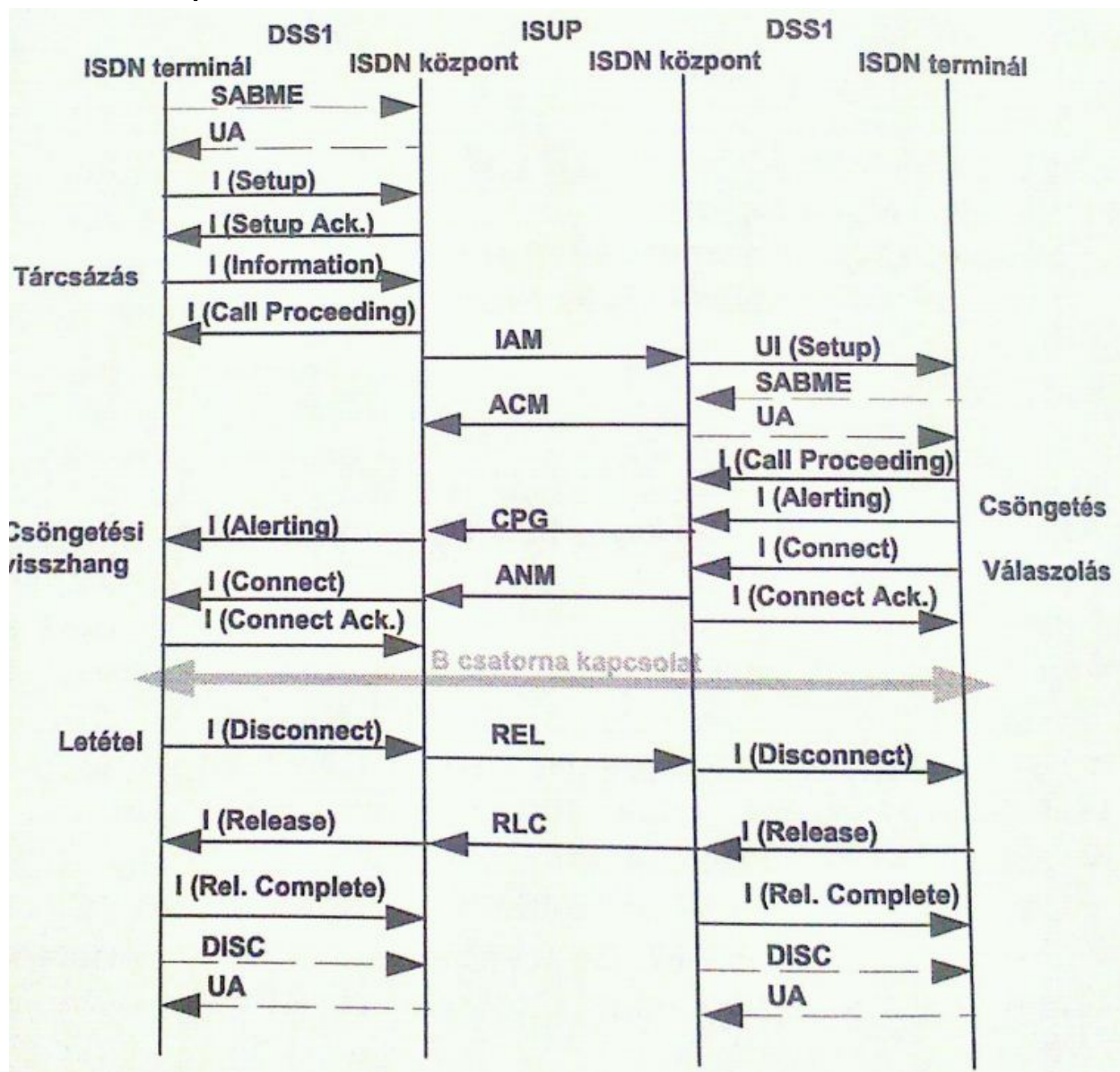
2. szakasz:

D-E normál esetben nem vinne át jelzést. E-ből D elérhetetlenné, D-E elérhetetlenné válik, ezért átvitel letiltást küldenek az összes szomszédjuknak. Ezen üzenetek vétele a szomszédokban a megfelelő jelzésútvonal készlet teszt üzenetek küldését váltja ki. E észreveszi, hogy számára F elérhetetlenné válik, ekkor C-ben, és D-ben kényszerített átírányítás hajtódik végre. C-E-F forgalmát C-D-F-re, B-E-F forgalmát B-D-F-re viszi át. Az említett szakaszokról csak az F felé tartó forgalmat kell áterelni.

eljárások:(TK)

- Átkapcsolás (changeover): egy szakaszból a teljes forgalom áterelése egy másik szakaszra. Elsőnek üzenet adás-vétel leáll a nem használhatóvá váló szakaszon. Meghatározzuk a kerülőutas szakaszt. Átkapcsolás parancsot küldünk a kerülőutas szakaszon a nem használhatóvá vált szakasz másik végpontjának. Ezután nyugta. meg az újraadási tárok feltöltése után megy a forgalom.
- Visszkapcsolás (changeback): Egy újra használhatóvá váló szakaszra a teljes forgalom visszaterelése. Megszűnik elsőnek a kerülőutas szakaszon az adás. Majd visszakapcs. parancs a kerülőúton. A kerülőúton nyugta küldése, majd sorrendhelyes üzenetfolyam folytatás.
- Kényszerített átírányítás (Forced Rerouting) egy nem használhatóvá váló útvonalról az adott útvonal teljes forgalmának áterelése egy használható útvonalra
 - Átvitel letiltás eljárás (Transfer Prohibited): az adott jelzőpont értesíti a szomszédokat, hogy egy hamradik jelzőpont felő őrajta keresztül ne próbáljanak meg forgalmat irányítani, mert elérhetetlen tőle.
- Vezérelt irányítás (Controlled Rerouting) egy útvonalra való visszaterelés.
- Jelzésútvonal készlet teszt eljárás: ő akkor van, ha egy jelzőpont átvitel letiltás vagy átvitel vezérlés üzenetet kap, akkor időközönként (ált. 30-60 sec) megkérdezi az üzenetet küldő jelzőpopntot, hogy a hiba oka még mindig fenn áll-e.

9. Ismertesse két digitális ISDN készülék közötti hívás felépülését, ha a két készülék különböző központban van!



IAM : Initial Address Message, áramkör lefoglalás, szállítja a hívott számot

SAM: Subsequent Address Message, hívás típusának meghatározásához további számok

ACM: Address CoMplete, minden szám megérkezett, erről egy nyugta

CPG: Call Process, egy paraméter kivételével megegyezik az ACM-mel

ANM: ANSwer Message, ha a hívott fél fogadja a hívást, ekkor indul a díjszámítás

REL: RELEase, ez az üzenet jelzi a hívásbontási szándékot

RLC: ReLEase Complete, a bontási nyugta üzenet

10. Milyen teszttípusokat ismer?

(Megtekintésen azt mondta, hogy 5 pontért 5 db-ot várt el, mindegyik csak 1 pontot ért, még akkor is ha egy oldalt írtál róla mélyen részletezve...

Tankönyv 37. oldalán van tesztelés, de előadáson sztem elmondott olyat, ami a könyvben nincs benne, és a könyvben is van, amit meg előadáson nem mondott el.)

Meg kell tervezni, hogy mit akarunk tesztelni, mivel minden eset tesztelése nem lehetséges a megközelítőleg végtelen számú teszt eset miatt. A tesztek lehetséges kimenetei

- Pass: ha minden megfelelő
- Fail: ha nem megfelelő értéket küld
- Inconclusive: ha nem az jött amit vártunk vagy egyáltalán nem jött semmi, pl. az eszköz jól működik de hálózati hiba miatt nem érkezett meg az üzenet

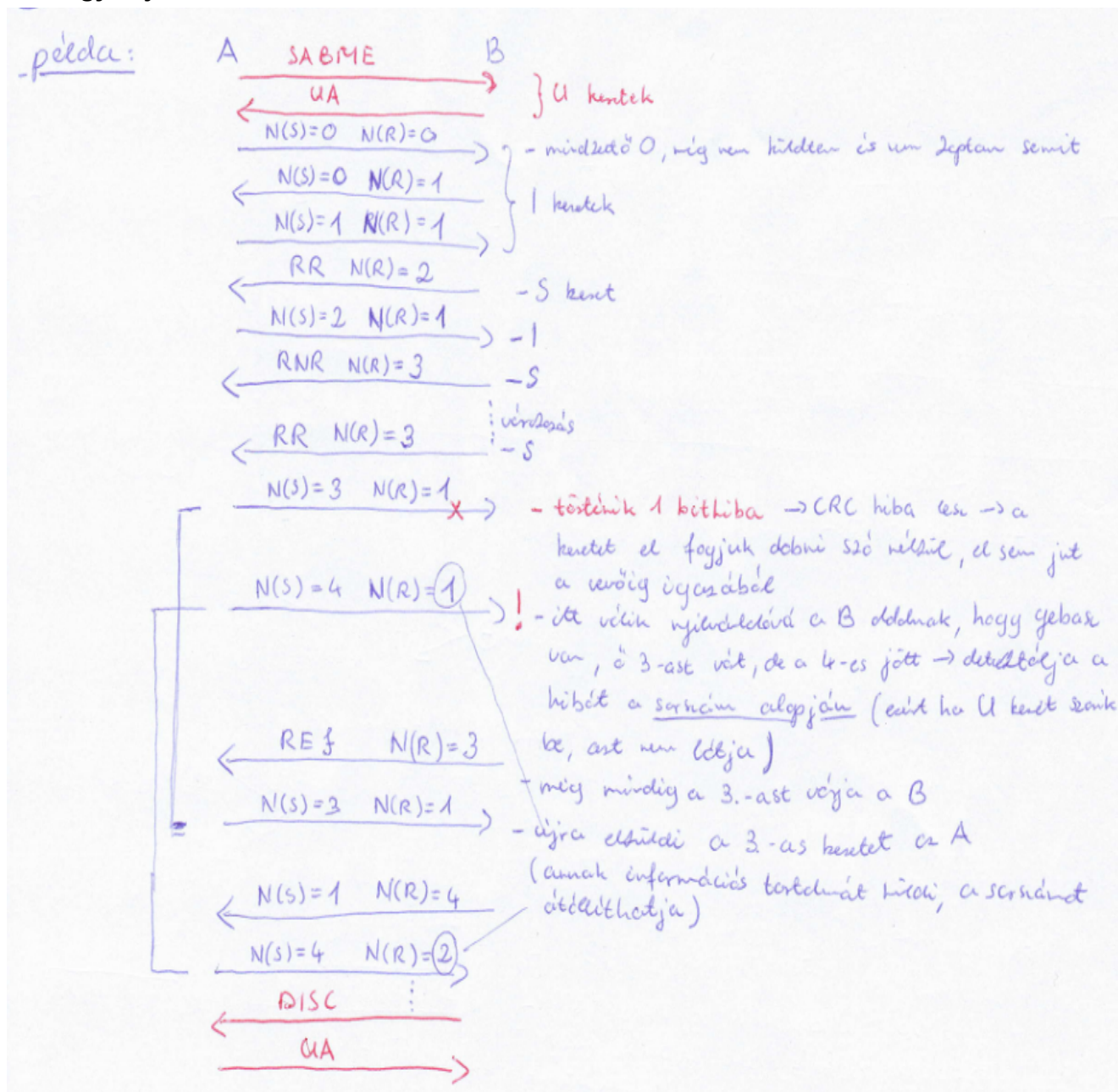
1. White-box testing: Főleg szoftvertesztelési módszer, amelynél a rendszer belső működését akarjuk tesztelni. Az adott tesztelendő komponenst egy "üvegdobozként" képzeljük el, tehát belelátunk a belső felépítésébe, így tudunk olyan tesztek írni, amely ezt figyelembe veszi. Ilyen tesztelési módszer a bemenetek tesztelése, helyes illetve helytelen valamint szélsőséges esetekben előforduló bemenetekkel futtatjuk, és megnézzük, hogy mit reagál, megfelelőek-e a bemenetek vizsgálata. Továbbá elágazások tesztelése, valóban minden IF ág le tud futni, és nincs benne olyan, amely olyan feltételt vár el, amely sohasem teljesülhet, így gyakorlatilag holtágak maradnak a kódban, amelyek sosem futhatnak le. Valamint a ciklusok tesztelése, nem kerül-e egyik sem végtelen ciklusba, mindegyik lefut-e. Ezzel a 3 módszerrel a kódok 90%-nak lefedettsége elérhető, amely egy jó arány.
 - a. Provokatív tesztelés: Azt ellenőrizzük, hogy a hibákat hogyan kezeli, pl. rossz input, rossz sorrend, szintaktikailag helytelen, rosszkor küldünk üzenetet stb..
 - b. Function test: egy funkció, modul, szolgáltatás tesztelése eszközön belül
 - c. System test: Több funkció együttes tesztelése, hogyan működnek együtt (gyakorlatilag együttműködési teszt más néven)
 - d. Regressziós teszt: Akkor kell lefuttatni, ha a belső megvalósítást megváltoztatjuk, ellenőrizni kell, hogy a módosítással nem rontottunk el valamit, így a korábban már helyesen lefutott tesztek a módosítás után ismét lefuttatjuk, hogy továbbra is megfelelően működik-e minden.
2. Black-box testing: Nem ismerjük az adott tesztelendő komponens belső felépítését, számunkra az egy "átláthatatlan" fekete doboz, csak az interfészt (kommunikációs pontjait) "látjuk" és a kimeneten megjelenő eredményt. A bemenet a Point of Control (PC) és a kimenet a Point of Observation (PO). A szoftverkomponenseken túl, a hardver eszközök tesztelése ilyen, mivel a hardver belső felépítése minket nem izgat, csak azt tudjuk, hogy milyen bemenetet vár el, és, hogy arra milyen kimenetet szeretnénk kapni.
3. Konformancia teszt: Ha egy szoftver már technikailag helyesen működik, akkor utána azt kell ellenőrizni, hogy valóban azt csinálja-e, amit a specifikáció előír (gyakorlatilag

verifikáció utáni validáció¹, tehát nem elég, hogy jól működik, valóban azt is csinálja amire eredetileg készíteni akarták). A konformancia teszt nem állítja, hogy a protokoll megvalósítás 100%-osan konform a specifikációval, a gyakorlatban ez egyfajta elfogadható kompromisszum a tesztesetek nagyon nagy száma miatt. *A konformancia teszt egyfajta black-box teszt, így nem biztos, hogy ez külön típusnak számít.*

- a. Basic Interconnection Test: a minimális követelményeket ellenőrzi, megnézi, hogy a megvalósítás egyáltalán alkalmas-e más megvalósított objektumokkal kapcsolatot felépíteni, kommunikálni. Érdemes-e egyáltalán további teszteket végezni → első szűrő.
 - b. Capability Test: ellenőrzi a megvalósított objektumban meglévő statikus tulajdonságokat, képességeket, pl. szabvány szerint előírt kötelező tulajdonságok megvannak-e.
 - c. Behavior Test:
- 4. Együttműködési teszt: A konformancia tesztelésen sikeresen átesett komponensek képesek-e egymással együttműködni.
 - 5. Teljesítőképeségi vizsgálat: Mekkora forgalmat generálnak, mekkora forgalmat képesek kezelni, forgalmi csúcsokat hogyan vezetnek le, egy adott forgalom mellett mekkora késleltetéssel működnek. Ennek teszteléséhez általában a szimuláció is sokat segít.
 - a. Load/Performance test: eltérő terhelések során hogyan viselkedik, mit bír el még az eszköz/rendszer

¹ <http://softwaretestingfundamentals.com/verification-vs-validation/>

11. Hogyan javít a LAPD hibát?

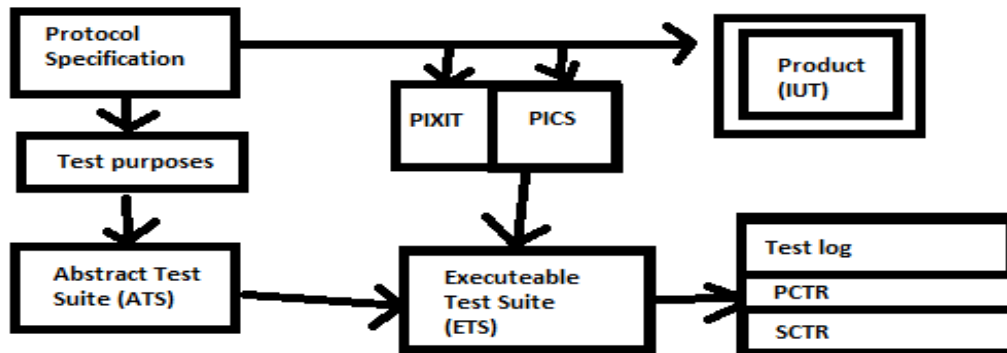


Lényeg, hogy nem javít bithibát, hanem újraküldet.

2. Vizsga + tuti közös rész

disclaimer: idén(2014 tavasz) kissé bővebb volt a ZH-s rész, szóval lehet, hogy pár példa (pl. DSL) már inkább vizsgában fordul elő csak. ASN szinte mindig tuti van.

1. Hogyan zajlik egy tesztlaborban egy protokoll megvalósítás tesztelése (hogyan választjuk ki a futtatandó teszteseteket, milyen dokumentumok használatosak? (ez gyakorlatilag a zh teszt témakör kérdés)



Wikiről:

IUT: (Implementation Under Test) tesztelés alatt álló megvalósítás

Input: PICS (Protocol Implementation Conformance Statement - protokoll megvalósítás alkalmassági nyilatkozat) és PIXIT (Protocol Implementation eXtra Information for Testing - protokoll megvalósítás vizsgálatára vonatkozó extra információ). Ezek alapján meghatározzák, hogy:

- Mely teszteseteket
- Milyen paraméterekkel
- Milyen elrendezésben

kell tesztelni. Az Abstract Test Suite-ból (absztrakt tesztkészlet) Executable TS-ot (végrehajtható tesztkészlet) készítenek. A futtatás során minden lényeges esemény naplózásra kerül.

Output:

PCTR (Protocol Conformance Test Report - protokoll alkalmassági teszt jelentés):

- Milyen megvalósításon
- Mely vizsgálatokat
- Milyen opciók mellett
- Milyen elrendezésben
- Milyen eredménnyel

hajtottak végre

SCTR (System Conformance Test Report - rendszer alkalmassági teszt jelentés): A teljes vizsgált rendszert jellemzi

Illetve tesztelés elmélet : lásd fentebb.

2. GSM hálózatban használt legfontosabb azonosítók

(Tankönyv 120. oldal)

- **IMSI(International Mobile Subscriber Identity):** a GSM hálózatban használt legfontosabb azonosító a SIM kártyát és azon keresztül az előfizetőt azonosítja. Ez 3 részből tevődik össze: **IMSI = MCC + MNC + MSIN** ahol
 - **MCC(Mobile Country Code):** az ún. mobil országkód, pl. Magyarországé 216
 - **MNC(Mobile Network Code):** az ún. mobil hálózati kód pl. 01,30,70
így az MCC+MNC együtt azonosítja azt a szolgáltatót, amely a SIM kártyát kiállította, tehát akinél előfizettünk.
- **MSIN(Mobile Subscriber Identification Number):** max. 10 jegyes mobil előfizető azonosító szám, amely az adott hálózaton belül egyértelműen azonosítja az előfizetőt.

Az IMSI egyértelműen szolgáltatóhoz köthető, tehát ha szolgáltatót váltunk és megtartjuk a hívószámunkat, akkor ezzel SIM kártyát és IMSI-t is cserélni kell. A GSM rendszerekben található adatbázisok (HLR, VLR, AUC..) mindig az IMSI alapján azonosítják a felhasználót, nem telefonszám alapján.

- **MSISDN(Mobile Station ISDN Number):** ez az a telefonszám, amelyen a készülék hívható. Ez a szám 3 részből áll az ITU-T E.164 ajánlás alapján:

MSISDN = CC + NDC + SN ahol

- **CC(Country Code):** országkód pl. Magyarország 36
 - **NDC(National Destination Code):** szolgáltatás- és hálózatkijelölő szám, amelyet a köznyelvben körzetszámnak neveznek pl. 20,30,70
 - **SN(Subscriber Number):** előfizetői szám, amely jelenleg 7 számjegyű
- **IMEI(International Mobile Equipment Identity):** a mobil készülék azonosítására szolgáló 15 jegyű azonosító. Ez is 3 részből áll: **IMEI = TAC + FAC + SN** ahol
 - **TAC(Type Approval Code):** a készülék típusazonosító kód
 - **FAC(Final Assembly Code):** a gyártó azonosító kód

E kettő együtt (TAC+FAC) együttesen, egyértelműen azonosít egy készüléktípust, míg a harmadik rész egy hatjegyű gyári szám **SN(Serial Number)**. Ez alapján lehet hálózati bejelentkezéskor ellenőrizni, hogy az adott készülék lopott-e.

- **MSRN(Mobile Station Roaming Number):** a földrajzi hálózatokkal ellentétben a mobil hálózatokban nem lehet telefonszám alapján megmondani, hogy egy készülék hol található, így emiatt bevezették az ún. barangoló számot. Ez alapján lehet hívást irányítani. Ha egy készülék egy adott MSC-hez bejelentkezik, akkor az MSC címe bejegyzésre kerül a HLR-be. Hívásvégződtetésre a HLR attól az MSC-től ahova az előfizető bejelentkezett, kérni fog egy ideiglenes telefonszámot amelyen az előfizető az adott hívás alkalmával elérhető lesz, ez lesz az MSRN. (ez megy az ISUP IAM hívásfelépítési üzenetében).
- **TMSI(Temporary Mobile Subscriber Identity):** az első bejelentkezésre a központ hozzárendel a készülékhez egy véletlenszám jellegű ideiglenes azonosítót (TMSI), amely alapján nem visszafejthető, hogy ki kommunikál. Erre a felhasználók kilétének elrejtésére

és a kommunikáció titkosítása miatt van szükség (ne lehessen lehallgatni, beazonosítani ki kommunikál). Ezt a TMSI-t a SIM kártyán tárolják. Következő bejelentkezéskor már ezt használja a készülék. Központ váltáskor az új központ egy újat rendel a készülékhez. Sokszor a gyakorlatban, akár minden beszélgetéskor újat generálnak a nagyobb biztonság érdekében.

- **LAI(Location Area Identifier)**: ún. körzet azonosító, amely szintén 3 részből áll **$LAI = MCC + MNC + LAC$** ahol az első két tag feljebb már ismertetett, a **LAC(Location Area Code)** pedig a hálózaton belül azonosít egy körzetet és azzal együtt a körzetet vezérlő központot.
- **GCI(Global Cell Identity)**: a körzeten belüli cella azonosítására a globális cellaazonosító, amely a körzet azonosító mellett egy cella azonosítót(CI) is tartalmaz.
 $GCI = LAI + CI$

ht:”GSM azonosítókhoz lehet felvenném a RAI-t és a HoN-t.”

- **RAI(Routing Area Identification)**:
 - LAC + RACból áll össze. (Routing Area Code). Paging és regisztrálási műveletekhez szükséges.
- **HoN(Handover Number)**: MSC-k közötti handover esetén van szükség a generálására.
 - (It is important to notice, however, that the target MSC and the source MSC are two telephone exchanges. The call can be transferred between two exchanges only if there is a telephone number identifying the target MSC. Such a situation makes it necessary to generate a new number, the Handover Number (HON).)

3. GSM Location Update NSS-ben (tankönyv 137. oldal)

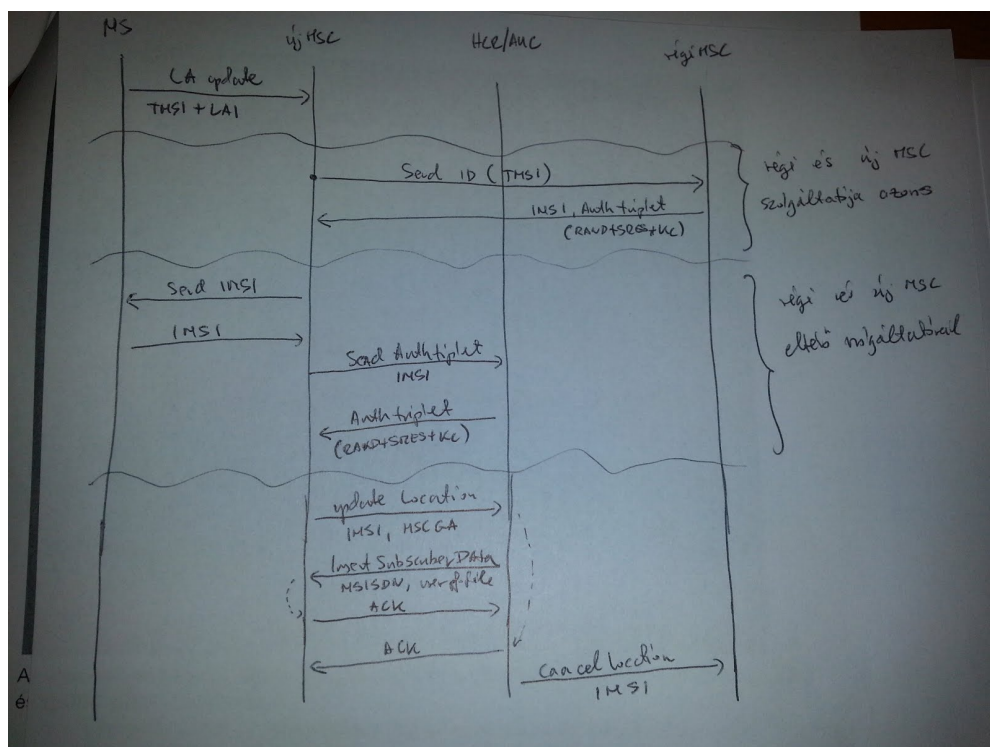
(wikiről: "A location update-nél mindkét verzió mindig kell, és a TCAP end hiánya miatt, ami mondjuk szerintem tökre nem a lényeghez tartozik szintén bőven lehet pontokat veszteni... Ja és az üzenetekhez kell, hogy mik a paraméterek, ez szintén pontlevonás. ")

Mivel effektíve a location update a kérdés, ez a cella váltásos rész nem biztos, hogy kell

- A készülék folyamatosan méri az adott cella jelerősségét amelyben tartózkodik, ha a jelszint egy megadott érték alá csökken és egy szomszéd celláé pedig egy érték fölé nő, akkor cellát vált. Erre azért van szükség, hogy ha cella szélén van, akkor kisebb változás miatt ne ugráljon ide-oda a cellákban a mobil.

Cellaváltáskor:

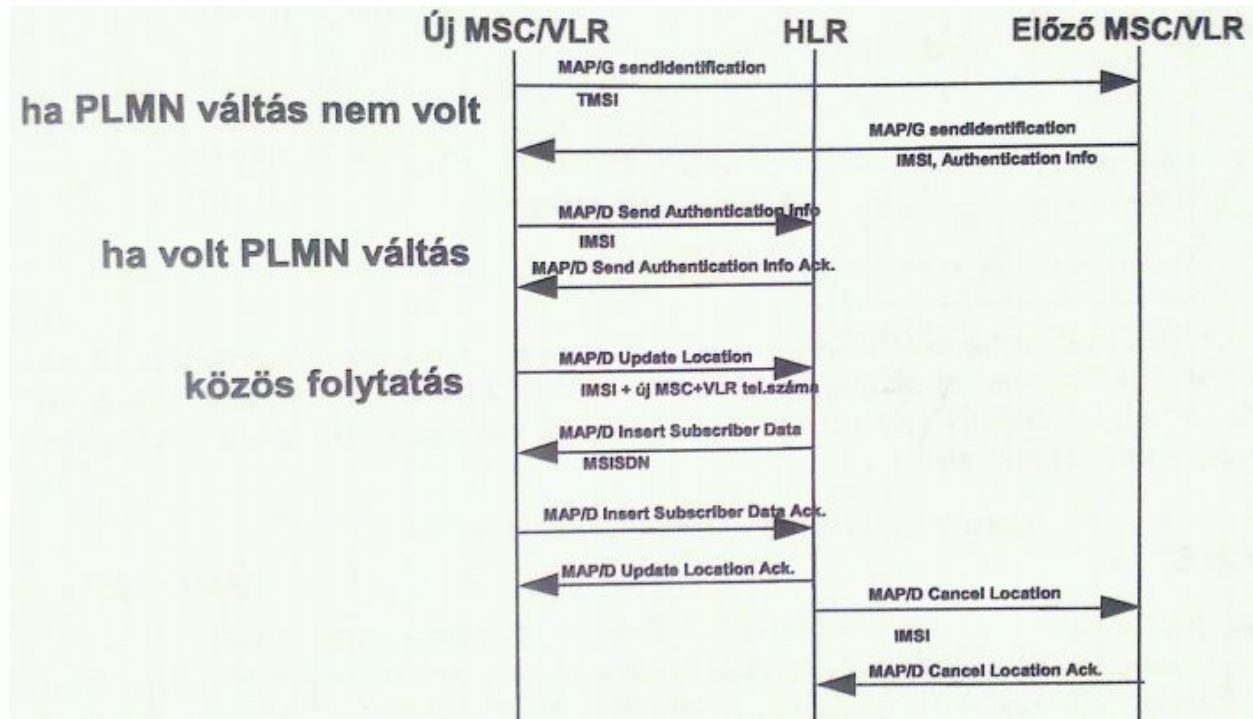
- Az adott cella ugyanahhoz a LA-hoz tartozik, nem kell tenni semmit
- Az adott cella egy másik LA-hoz tartozik
- Az adott cella egy másik LA-hoz tartozik, amely egy másik MSC-hez tartozik de ugyanahhoz a szolgáltatóhoz
- Az adott cella egy másik LA-hoz tartozik, amely egy másik MSC-hez tartozik és szolgáltató váltás is történt
- Ekkor ellenőrzi, hogy az adott cella még ugyanahhoz a Location Area-hoz tartozik, vagy LA-t is váltott-e.
 - Ha nem váltott LA-t, akkor ugyanahhoz az MSC-hez tartozik, akkor az MSC a saját VLR-jében átírja a LA-t (IntraLA váltás)
 - Ha váltott LA-t, akkor (InterMSC váltás) az új LA más MSC-hez tartozik mint az előző, az MSC-nek meg kell tudnia a készüléktől az IMSI-jét, kell AuthTriplet, kell készülék UserProfile (a mobil TMSI-vel jelentkezett be). HLR-hez el kell küldeni az új MSC azonosítót, értesíteni a régi MSC-t, hogy törölheti a VLR-jéből a készüléket



Az ábra első részében a régi és új központ azonos szolgáltatónál. Második esetben pedig a régi és új MSC eltérő szolgáltatónál.

1. A készülék elküldi a TMSI-jét és a régi LAI-t, itt a LAI-ból kiderül, hogy ugyanahhoz az MSC-hez tartozik vagy sem, illetve volt-e szolgáltató váltás vagy sem (mivel a LAI = MCC + MNC + LAC)
2. a TMSI alapján elkéri a régitől az IMSI-t és az autentikációs tripletet (RAND+SRES+Kc). Az IMSI-ben benne van az ország és szolgáltatókód (IMSI = MCC + MNC + MSIN),
3. Mivel az új MSC nem ismeri a régi MSC-t a szolgáltató váltás miatt, ezért elkéri a készüléktől az IMSI-t, amely alapján beazonosítja a honos szolgáltatót a HLR-t.
4. Az új MSC a HLR-től elkéri az autentikációs tripleteket, majd lezajlik az autentikációs folyamat (másik feladatnál ott van részletesen az ábra)
5. Az új MSC a HLR-nek elküldi a location update-et a készülék IMSI-jével és a saját globális azonosítójával, amelyet a HLR nyugtáz és az MSC is nyugtáz
6. Végül szól a HLR, hogy törölheti a régi MSC a VLR-ből a készüléket

Gyakorlatban, hogy spóroljanak a rádiós csatorna terhelésével, az eszközök LA update során mindig IMSI-t küld amúgy meg TMSI-t, így egy plusz üzenetküldés megspórolható. Ha nincs pozíció váltás, akkor is van periodikus LA update, hogy a HLR tudja frissíteni az adatait, abban az esetben, ha valamikor HLR leállás lett volna, de ezeket a HLR-ek eldobják általában, kivéve ha tényleg volt leállás.



(PLMN: public land mobile network, maga a szolgáltató)

4. GPRS-ben miért kell csökkenteni a paginget, és hogyan oldják meg?

http://books.google.hu/books?id=uxynrOg6BFcC&pg=PA92&lpg=PA92&dq=gprs+routing+area&source=bl&ots=jViWgQ7Yeb&sig=W5xKfmYd4alq7pvkKLyZATtdcdM&hl=hu&sa=X&ei=EiiCU6_GsLD7AaMIYGwBg&ved=0CFIQ6AEwAw#v=onepage&q=gprs%20routing%20area&f=false

Az adatforgalom jellemzése teljesen eltérő mint a beszédé. A beszéd rövid ideig tart, illetve 1 beszéd alatt kevesebb cellaváltás történik, de van rá handover, illetve flyamatosan tart, valamint nem gond, ha néhány csomag elveszik, más QoS van, pl ne legyen késleltetés ingadozás.

Az adatforgalom BURST jellegű és nem folyamatos, (nem veszhetnek el csomagok) így nagy valószínűséggel mozog el a cellából a következő burst-ig, hosszú szünetek vannak a burst-ök között. (pláne kis UMTS/LTE cellák esetén)

Nem célszerű végig követni a felhasználót, ahogy beszédnél sem volt, csak a burst elején keressük meg (sok burst, sok paging). A sok paging megspórolása miatt bevezették a **ROUTING AREA (RA)** azonosítót, amely kisebb területet fed le mint a LA. A beszédet 1 LA-ban építik fel, míg az adatkapcsolatot 1 RA-ban. tehát **RA < LA** ahol **RAI = LAI + RAC**

A mobilnak az RA váltást kell jeleznie, ezért elég csak az SGSN-hez bejelentkezni, mert pontosabban ismeri a helyzetét mint az MSC. Az SGSN-hez RA update során felküldi a **RAI+PTMSI**-t ahol PTMSI a Packet-TMSI.

Az RA update teljesen analóg a LA update-tel.

Beszédet tehát LA-ban építünk fel, adatkapcsolatot pedig RA-ban.

Definiálták a **NullRA-t**, amely olyan körzet, ahol nincs GPRS szolgáltatás

Továbbá a hálózat nyilvántartja, hogy a mobil milyen állapotban van:

- **Idle**: mobil nem kapcsolódik az adathálózathoz, így LA szinten tartjuk nyilván
- **Ready**: kereteket küld, cella szinten ismerjük a helyzetét
- **Standby**: ha befejeződik az adatforgalom, akkor indul egy timer, ha az lejár, menjen át standby-ba és cella helyett csak RA szinten tartjuk nyilván a pozícióját

Wiki megoldás:

Az adatforgalom börsztös, tehát kis-kis adagokban megy egymás után. Ennek következtében két adatbörst közötti cellavándorlás kikerülése végett, paging-et kell küldeni, hogy hol van a fogadó fél, hogy oda menjen az adat. Viszont ez rentgeteg paginget jelentene, ezért felosztották a location area-kat routing area-kra amik sokkal kisebbek, és elég csak egy ilyen kisebb környezetben jelezni, ami nem egy egész nagy location area-t terhel.

- **IDLE** állapot: a mobil állomás (MS) nem kapcsolódik a GPRS hálózathoz, ilyenkor sem adatforgalmazás, sem a felhasználó kiértékesítése (paging) nem lehetséges, a felhasználó elérhetetlen; a PDP környezet létrehozásához az MS-nek el kell végeznie a GPRS bejelentkezés folyamatát.
- **STANDBY** állapot: a mobil állomás (MS) már kapcsolódott a GPRS hálózathoz, így kiértékesítése lehetséges, adatforgalmazás azonban ilyenkor sem. MS helyzetét routing area szinten ismerjük.

- **READY** állapot: a mobil állomás (MS) képes PDP PDU-k (Packet Data Unit) küldésére és fogadására; az SGSN folyamatosan frissíti az útvonal- és cellaválasztási információkat. MS helyzetét cella szinten ismerjük.

Idle állapotból Ready állapotba úgy kerülhetünk ha bejelentkezünk a GPRS hálózatba. Ha kijelentkezünk akkor visszakerülünk Idle-ba. Ha Ready állapotban vagyunk és egy darabig nem küldünk adatot vagyis lejár a timer átkerülünk Standby állapotba. Standby állapotból Ready állapotba egy keret küldéssel léphetünk. Ha Standby állapotba vagyunk és itt is lejár a timer vagy kijelentkezünk a gprs hálózatból akkor visszakerülünk Idle állapotba.

Egyéb infó a megértéshez:

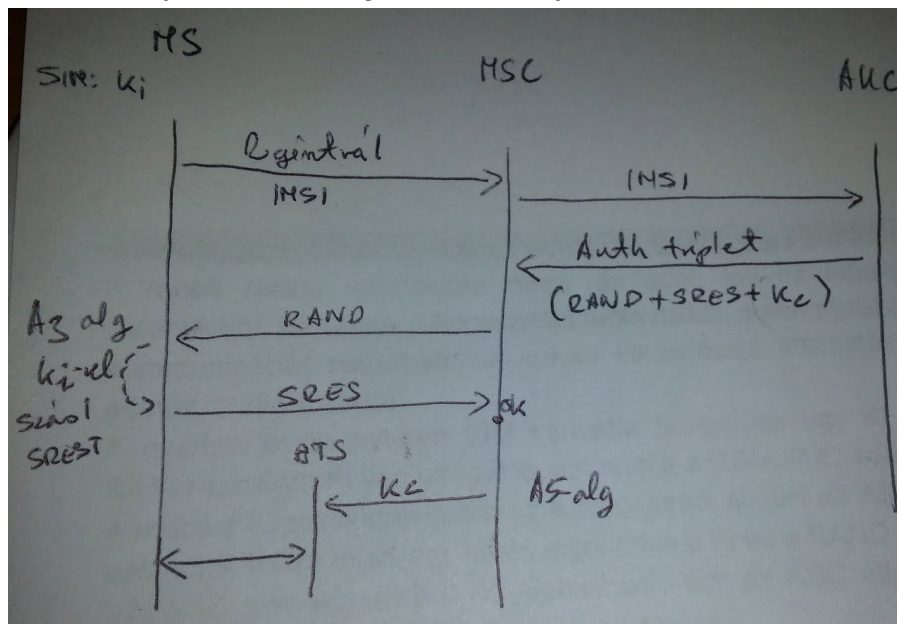
Paging: (GSM és telefónia esetében) Practically every cellular system has some kind of broadcast mechanism. This can be used directly for distributing information to multiple mobiles, commonly, for example in mobile telephony systems, the most important use of broadcast information is to set up channels for one to one communication between the mobile transceiver and the base station. This is called paging. The three different paging procedures generally adopted are sequential, parallel and selective paging.

Handover: In a cellular system, as the distributed mobile transceivers move from cell to cell during an ongoing continuous communication, switching from one cell frequency to a different cell frequency is done electronically without interruption and without a base station operator or manual switching. This is called the handover or handoff.

5. Mobil állomás autentikálása, miért van rá szükség és mikor, hogyan történik?

A mobil rádiós interfészen megy az adat, így ez lehallgatható lenne, ezért szükséges autentikálni, de ennek a felhasználó tudta nélkül, automatizáltan kell történnie. Továbbá az egész kommunikációt aszimmetrikus kulcsú titkosítással titkosítják is. (GSM esetén 2^{128} , UMTS esetén 2^{256} db kód)

A gyártás során minden SIM kártyába beégetnek egy K_i (Individual SubscriberKey) -t, adott SIM-hez milyen K_i van beégetve, ezt tárolja a GSM AUC adatbázisában.



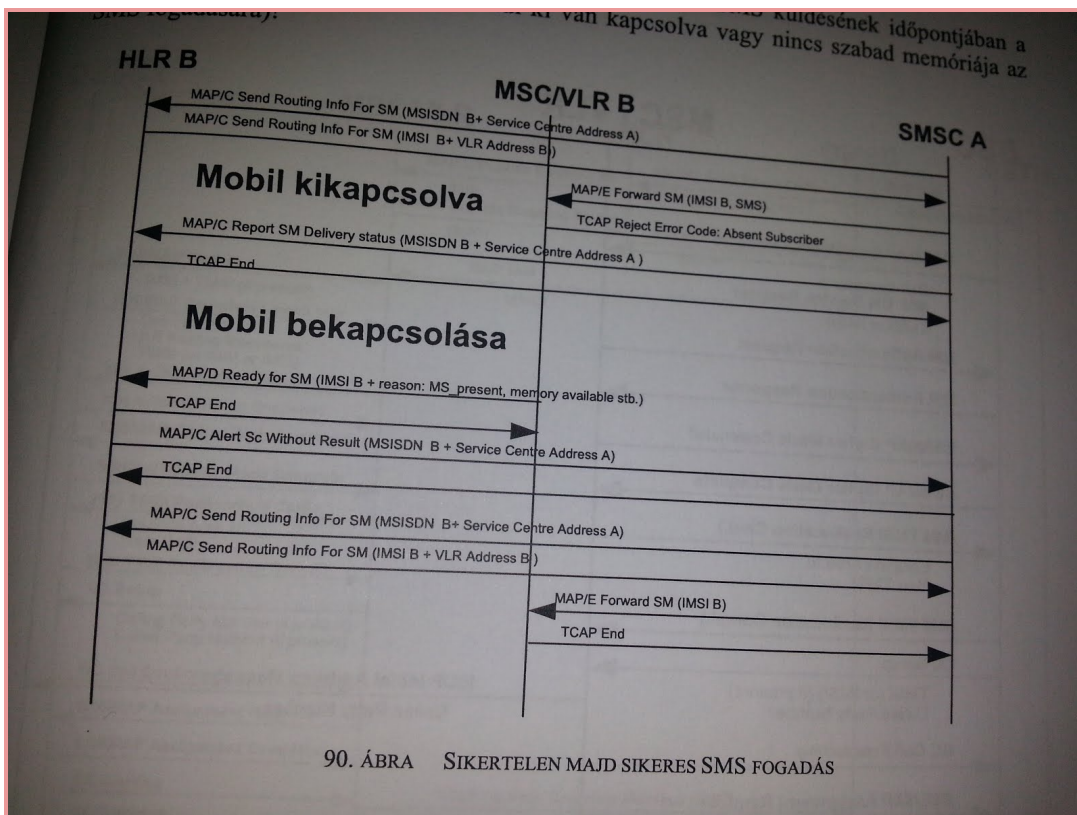
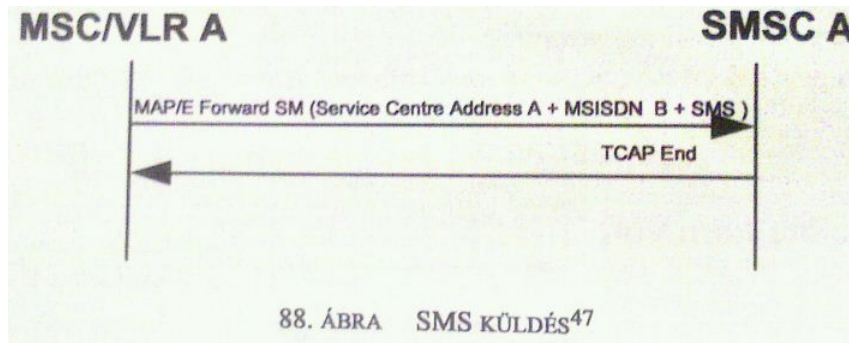
1. Először a mobil "regisztrál" LA update-tel az MSC-nél, az MSC elküldi az IMSI-t az AUC-nak és az alapján kér autentikációs adatokat, vagyis Autentikációs Tripletet (azért triplet, mert három paramétert kap vissza)
2. Az AUC az IMSI alapján elvégzi a számításokat és az autentikációs tripletet visszaküldi (RAND+SRES+ K_c).

Minden SIM kártyán található egy egyedi K_i kód (GSM esetén 2^{128} , UMTS esetén 2^{256} kódú), amelyet a szolgáltató az AUC-ban rögzít az IMSI-hez. Az autentikációs triplet előállításakor az AUC generál egy véletlenszámot, amely a RAND lesz, ekkor a RAND és az IMSI-hez tárolt K_i kód alapján kiszámít egy SRES értéket. Az MSC ezt megkapja, a RAND-ot elküldi a mobil készüléknek, akinek ugyanez a K_i értéke van a SIM kártyán, tehát ugyanazt az SRES értéket kell generálnia. Miután a mobil ezt kiszámolta, felküldi az MSC-nek, aki bit szintű egyezést vizsgál.

3. Az MSC elküldi a mobilnak a RAND-ot, hogy számítsa ki a saját SRES értékét
4. A mobil kiszámolja a választ (SRES **S**igned **R**ESPonse), amit a központ ellenőriz (bitenkénti összehasonlítás) és ha megfelelő, akkor engedélyezi (A3-alg. végzi a számítást)
5. Az MSC a BTS-nek elküldi a harmadik paramétert, a K_c kódot, amely a kommunikáció titkosításához szükséges. (A K_c (**c**iphering **k**ey) kód alapján az A5-algoritmus végzi a titkosítást)
6. A titkosított beszédátvitel csak a mobil és a BTS között van, a BTS végzi el

6. Ismertesse egy SMS útját a feladó központjától a címzett központjáig

SMS elmélet: jelzés szolgáltatás amely gyakorlatilag egy egyszerű jelzés üzenet, egyáltalán nem garantált a kézbesítése, nincs garancia a megérkezésére. Az SMS Datagram szolgáltatás, tehát nincs kapcsolat kiépítve a küldő és a címzett között. Ezért is van itt egy mobil kikapcsolt és bekapcsolt rész, mert a hálózat többször is újra próbálja periodikusan küldeni az üzenetet, ha a készülék ki volt kapcsolva, illetve ha bejelentkezik.¹



Az ábrán a nyílakon lévő MAP/C rész nem kell, csak utána az üzenet neve és a paraméterek.

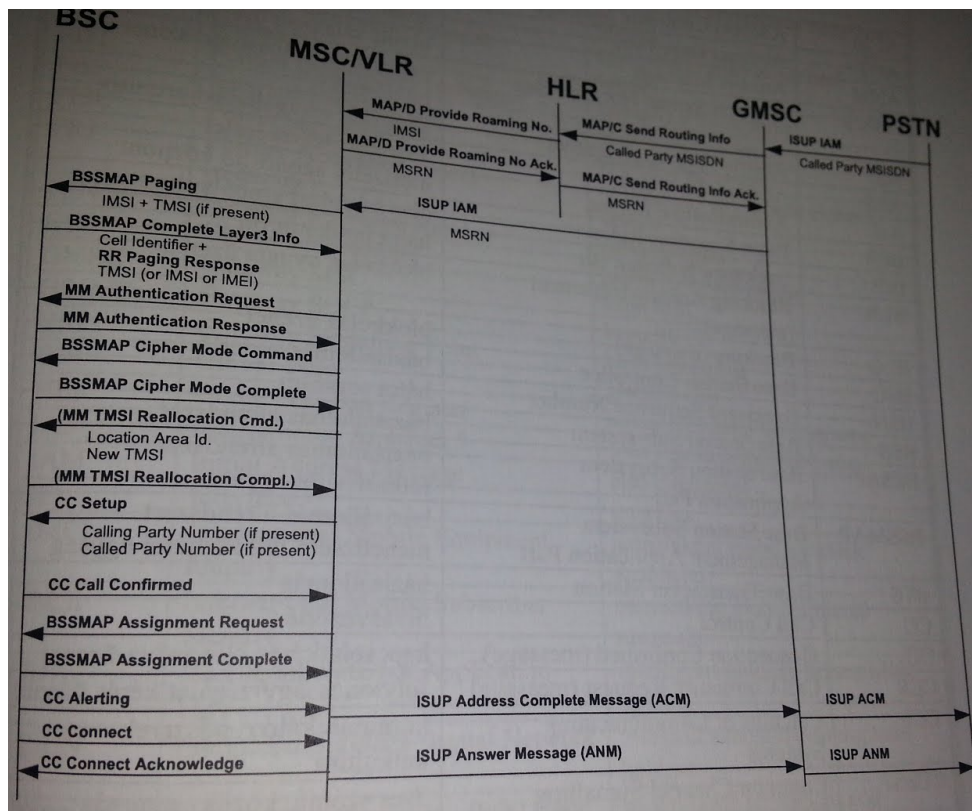
A: küldő adatai

B: címzett adatai

1. Az SMS feladójának MSC-je elküldi a kapott SMS-t a feladó SMS központjához, az SMSC-hez

2. A feladó SMSC-je a címzett HLR-jéhez fordul routing információért, amelyre válaszban a készülék IMSI-jét és az MSC MSISDN GT-jét kapja vissza
3. A feladó SMSC-je megpróbálja a címzett MSC-jének továbbítani az SMS-t, de az jelzi, hogy a mobil nem tudja fogadni, mert vagy ki van kapcsolva, vagy megtelt a memóriája
4. A feladó SMSC-je megkéri a címzett HLR-jét, hogy ha a mobil elérhetővé válik, akkor jelezze, ekkor majd újra megpróbálja elküldeni
5. A címzett MSC-je egy ReadyForSM üzenettel szól a HLR-nek, hogy a mobil elérhető
6. A HLR egy AlertInfoForSM üzenettel jelzi a feladó SMSC-jének, hogy próbálja meg ismét a küldést mert a mobil elérhető. Innétől ismét a fenti fázis fut le.

7. Ismertesse egy mobil készüléken végződő hívás (MT call) felépülését az A interfészen!



A lenti szöveg nem egészen nyerő, mert az A interface az a BSC és az MSC/VLR közti szakasz. Tehát pont nem az A if-ről ír.

A mobil telefonszáma (MSISDN) alapján csak a szolgáltató hálózatának Gateway-éig lehet eljutni, hálózaton belül már nem a telefonszám alapján történik az útirányítás, hanem az ún. MSRN alapján.

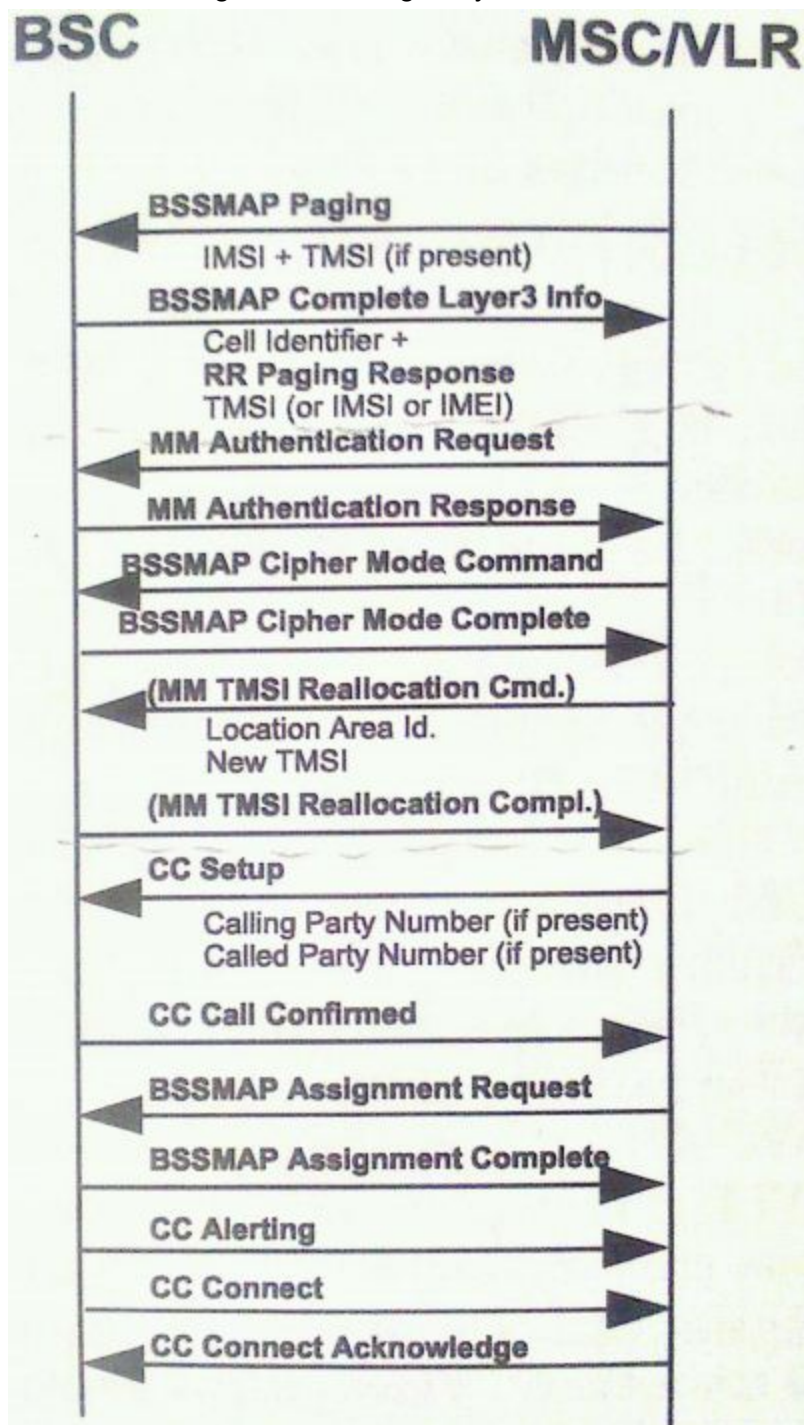
Beérkezik az MSISDN alapján a hívás a Gateway-hez, ekkor a GW a HLR-hez fordul, hogy az MSISDN alapján kérjen attól az MSC-től egy MSRN azonosítót, akinél jelenleg a mobil található. A HLR tárolja, hogy melyik MSISDN-hez melyik IMSI tartozik, és, hogy az adott IMSI-hez tartozó készülék melyik MSC-nél van (ez a VLR-ben van), így az adott MSC-hez fordul az IMSI-vel, hogy az IMSI-hez tartozó MSRN-t küldjön.

Az MSC elküldi a HLR-nek az MSRN-t, aki azt továbbítja a GMSC-nek, aki így már tudja irányítani a hívást, és felépíti a kommunikációs utat az MSC-vel.

MSC és a mobil között lezajlik a paging, megkell találni a készüléket és szólni neki, hogy bejövő hívása van. Miután a készülék vissza jelezte, lezajlik az autentikációs folyamat és utána a kommunikáció titkosítása a mobil és a BTS között.

Ezután kezdődik a hívás felépítése a SETUP üzenettel (ISUP üzenetek, ahogy korábban már tanultuk vonalas esetben a hívás felépítést).

Wiki szerinti megoldáshoz elég ennyi is:



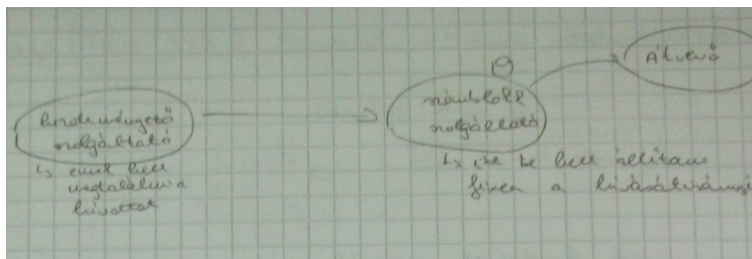
8. Milyen műszaki megoldások használhatóak Magyarországon a vezetékes számhordozás megvalósítására? Ismertesse őket röviden! Röviden magyarázza el, miért nem alkalmazhatóak a vezetékes hálózati megoldások a mobil hálózatokon!

Számhordozás típusai:

1. **szolgáltatók közötti** (operator portability) a szám marad, szolgáltató váltódik.
2. **Location portability** - szolgáltató igénybevételenek a helyét változtatni (pl. elköltözől)
 - a. csak vezetékes körzeten belül
 - b. jogi akadályok csak
 - c. technikailag megoldható
3. **Service portability** - szolgáltatásváltás, pl. VoIP jellegű megvalósítás
 - a. csak interfészek az analóg és digitális előfizetés berendezések, kapcs.közp.-ok

Technikai megvalósítási módszerek:

1. **Továbbirányítás** (onboard routing) feltétel nélküli hívásátírányítás, (1.-hez).
befogadó/átadó: akihez viszem a számom.

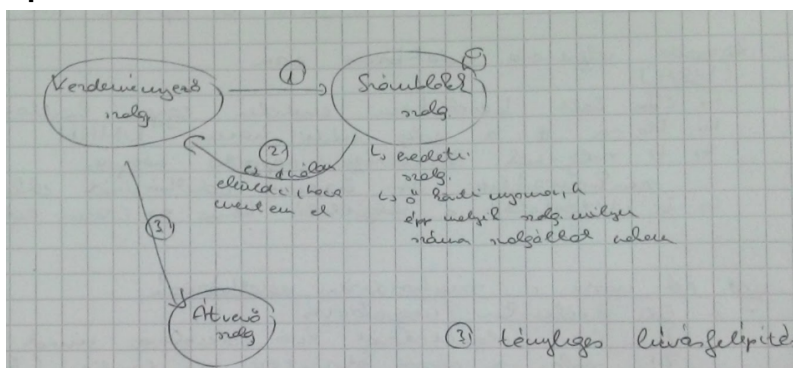


+mindenféle illesztés nélkül megoldható

- leterhelem a szolgáltatót, neki miért érné meg?

2. **Hívás visszahallás (drop-book)**

lépések:



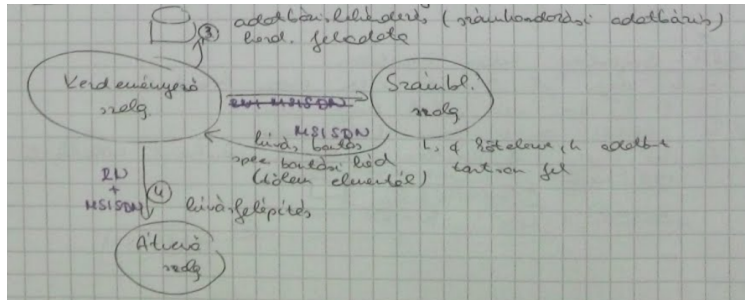
A válaszban (2.) elküldi hogy hova mentem el. Csak a 3. lépésben, tényleges hívásfelépítés történik.

+ a hívásútvonal a legrövidebb úton épül fel, nem megy keresztül a szolgáltatón

- a szolgáltatónak adatbázist kell fentartani, hogy kihez mentem

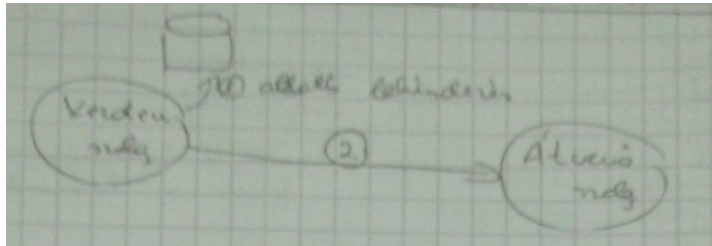
3. **Lekérdezés bontás (Query on-release)**

- a. Létezik egy adatbázis. Számhordozási adatbázis. Az ábra ugyan az. Az elv hasonló. Valami speckó hívás bontási eljárás van a számblokk és szolgáltató között, meg valahogy belejön az adatbázis is amit említettem, de képtelen vagyok értelmezni ezt az ábrát :D



- kevés hordozott számnál éri meg

4. Lekérdezés minden hívásról (All call query)



- a. Kezdeményező szolgáltatás --(1.)--> adatbázis
 b. Kezdeményező szolgáltatás --(2.)--> átvivő szolgáltatás
 - sok hordozott számnál éri meg

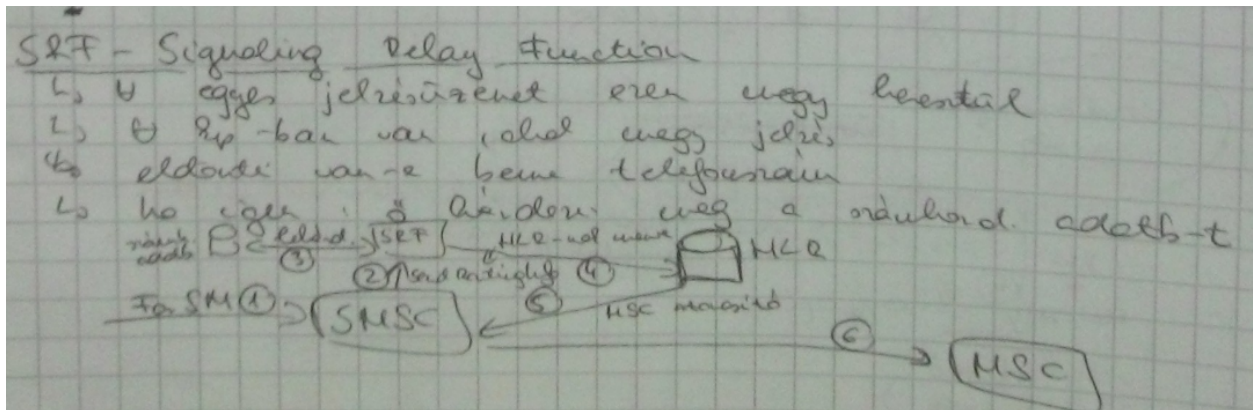
Központi referencia adatbázis (KRA)

- ebbe kell bejelentkeni minden egyes hordozást
- MO-on nem a szolgáltató vezérli hanem az NMHH (Nemzeti Média- és Hírközlési Hatóság)
- Minden szolgáltatónak kötelessége naponta négyszer szinkronizálni az üzemi adatbázisát ehhez
- Minden változtatást a szolgáltató beírja az üzemi adatbázisba

Mit ad vissza a számhordozási adatbázis?

- 2 féle technikai megvalósítás:
 - a. Új szolgáltatónál lévő ideiglenes számot tárolják
 - b. (Mo.-n) szolgáltatót, ahova átmentem, ezt tárolja
 - Minden szolgáltatónak van azonosítója: RN- routing number

RN: Szolgáltató Kód (SK) + Berendezés Kód (BK) (vezetékes szolgáltatók használják)



Miért nem lehet a vezetékes számhordozási technikákat alkalmazni mobil hálózatok esetében?

Azért mert vezetékesen BK-ra hordoznak, mobilon meg nem. Magyarul mert az egyik mobil a másik meg fix

-> tehát a vezetékes módszerek működnek, mert egy azonosító azonosítja a készüléket (BK), míg mobilon az IMSI, MSISDN, TMSI, stb. több különböző azonosító működik, ezért nem lehet kopipéztelni a módszereket.

9. TTCNv3 kód

```
module vizsgal {

type port PortType message { Inout integer, charstring;}

type component CompType {
    port PortType p1;
    port PortType p2;
    timer T := 10.0;
    var integer temp; }

template integer GoodNumber := ((1..20), 24, 50);

function f() runs on CompType {
    T.start; alt {
        [] p2. receive (GoodNumber) -> value temp ( p1. send (temp *2); setverdict (pass) ;
repeat }
        [] p2.receive { setverdict(fail); mtc.stop }
        [] T.timeout { setverdict(inconc); }
    }
}

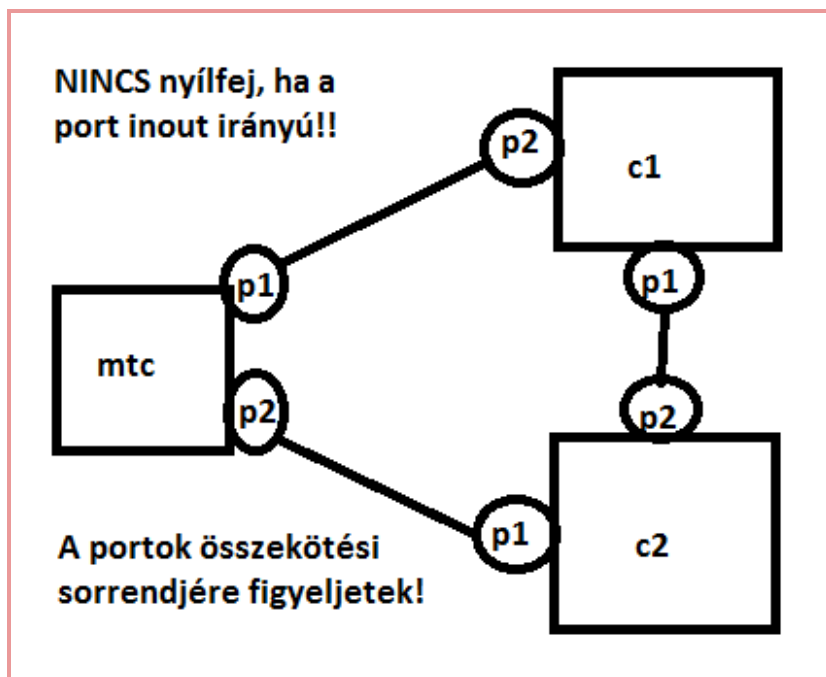
testcase tc() runs on CompType {
    var CompType c1 := CompType.create;
    var CompType c2 := CompType.create;

    connect(mtc:p1, c1:p2);
    connect(c1:p1, c2:p2);
    connect(c2:p1, mtc:p2);

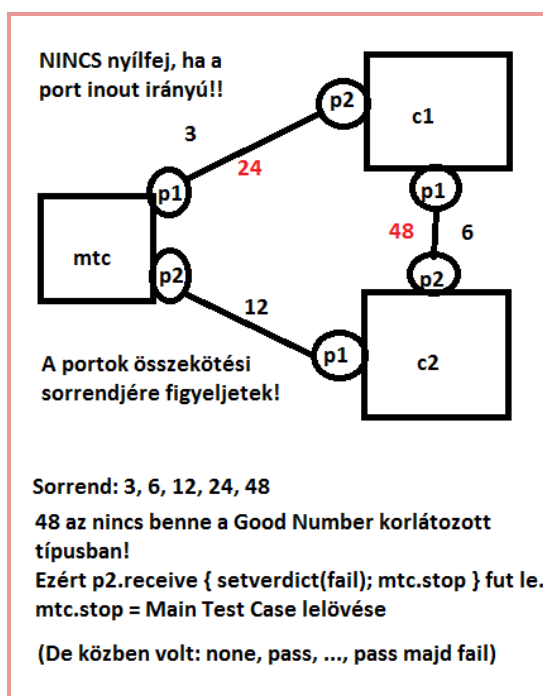
    c1.start(f());
    c2.start(f());
    p1.send(3);
    T.start;
    alt {
        [] p2. receive (GoodNumber) -> value temp { p1. send (temp *2); setverdict (pass) ;
repeat }
        [] p2.receive { setverdict(fail); self.stop }
        [] T.timeout { setverdict(inconc); }
    }
    all component.done;
}

control {
    execute(tc());
}
}
```

a.) Rajzolja fel a teszt komponenseket, portokat és azok kapcsolatait.



b.) Mutassa meg a teszt folyamatát! (Rajzoljon!)



c.) Mi a tesztesemény kimenete?

FAIL, mert a 48 nem egy elfogadott érték (nincs a Goodnumber elemei között)

Másik TTCNv3 feladat (bemásoltam a képet ide, így egyszerűbb)

másik feladat:

```

type port PortType message {
  inout integer, charstring;
} with { extension "internal" }

type component CompType {
  port PortType P;
  timer T := 10.0;
}

template integer GoodNumber := ((5..11), 13);

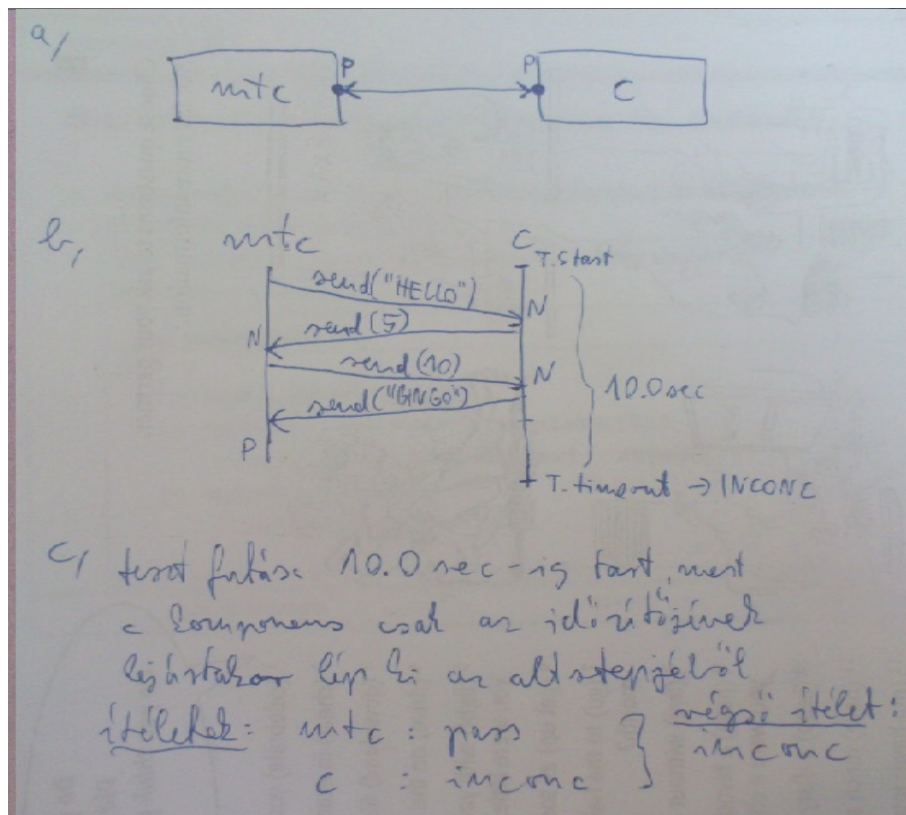
function f() runs on CompType
{
  T.start;
  alt {
    [] P.receive(GoodNumber) { P.send("BINGO"); repeat; }
    [] P.receive("HELLO") { P.send(5); repeat; }
    [] T.timeout { setverdict(inconc); }
  }
}

altstep MyAltstep(PortType X) runs on CompType
{
  var integer temp;
  [] X.receive(integer:?) -> value temp { X.send(temp * 2); repeat; }
  [] X.receive { setverdict(fail); self.stop; }
}

testcase tc() runs on CompType
{
  var default d := activate(MyAltstep(P));
  var CompType c := CompType.create;
  connect(mtc:P, c:P);
  c.start(f());
  P.send("HELLO");
  P.receive("BINGO");
  setverdict(pass);
  c.done;
}

control {
  execute(tc());
}

```



Tipus	Size	Sequence
a	OCTET	STRING;
b	[3]	IMPLICIT OCTET STRING;
c	[10]	OCTET STRING;
d	[5]	BOOLEAN,
e	[1]	IMPLICIT BIT STRING
f	[10]	BOOLEAN OPTIONAL

3



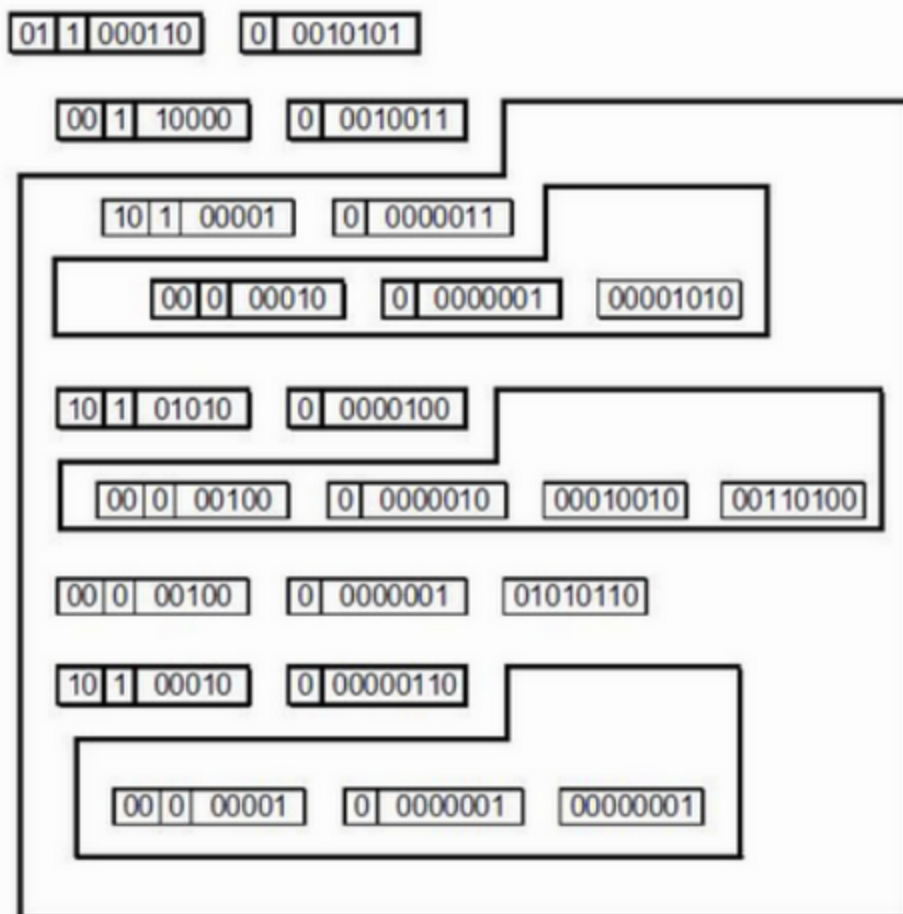
44

"benne" valami). Meg nem is kell: megnézed a konkrét adatot, és 8 biten = 2*4 biten 2db értéket tudsz elküldeni. Szóval ha a küldendő érték 56781234, akkor (5|6), (7|8), (1|2), (3|4) kell elkódolni, az ugye 4 byte, tehát a length 00000100 lesz, utána meg felveszel 4 byte mezőt, és elkódolod az értékeket."

Példa:

```
Tipus ::= [APPL 3] SEQUENCE {
    a [1] INTEGER , -- 10
    b [10] OCTET STRING, -- '1234'
    c OCTET STRING, -- '56'0
    d [2] BOOLEAN - TRUE
}
```

Megoldás:



4.) A következő ASN.1 leírás egy MAP üzenet (általam kissé módosított) definíciójának részlete:

```

sendRoutingInfoForSM OPERATION
  ARGUMENT
    routingInfoForSM-Arg [1] IMPLICIT SEQUENCE {
      msisdn [0] IMPLICIT OCTET STRING (SIZE (1..9)),
      sm-RP-PRI [1] BOOLEAN,
      serviceCentreAddress [2] OCTET STRING (SIZE (1..20)),
      teleservice [5] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
    }

  RESULT
  ...
  ERRORS
  ...
  ::= localValue 45

```

A vonalon a következő értékeket szeretnénk kiküldeni:

msisdn: '1234567890'
 sm-RP-PRI: TRUE (=1)
 serviceCentreAddress: '55667788'

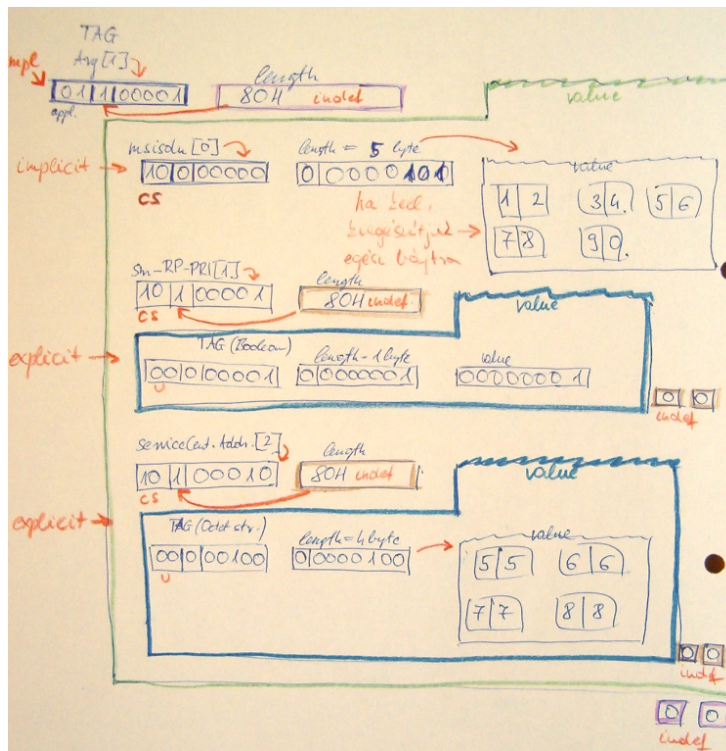
a. Milyen típusú TCAP keretben/komponensben küldhetjük a fenti üzenetet?

b. Készítse el a fenti üzenet MAP részének kódolását! Mindenhol, ahol lehet, indefinit hossz kódolást használjon! Segítségül:

a.): Argument -> Invoke komponens típus.

Ha argumentben van, akkor BEGIN vagy CONTINUE tranzakciós részű Invoke komponensű keretben, ha Result részben van, akkor CONTINUE vagy END tranzakciós rész Return Result Last vagy Return Result Not Last komponensben

- **ARGUMENT** kulcsszó után specifikált üzenetszerkezet használható **Invoke** komponensekben (ennél a feladatnál most ez van)
- **RESULT** kulcsszó után specifikált üzenetszerkezet használható **Return Result** komponensekben
- **ERROR** kulcsszó után specifikált üzenetszerkezet használható **Return Error** komponensekben



3. A következő ASN.1 leírás egy MAP üzenet (általánossá módosított) definíciójának részlete:
 sendRoutingInfoForSM OPERATION
 ARGUMENT

```

routingInfoForSM-Arg [12] IMPLICIT SEQUENCE {
  msisdn [41] IMPLICIT OCTET STRING (SIZE (1..9)),
  sm-RP-PRI [1] BOOLEAN,
  serviceCentreAddress [2] OCTET STRING (SIZE (1..20)),
  teleservice [5] OCTET STRING (SIZE (1)) OPTIONAL}
  
```

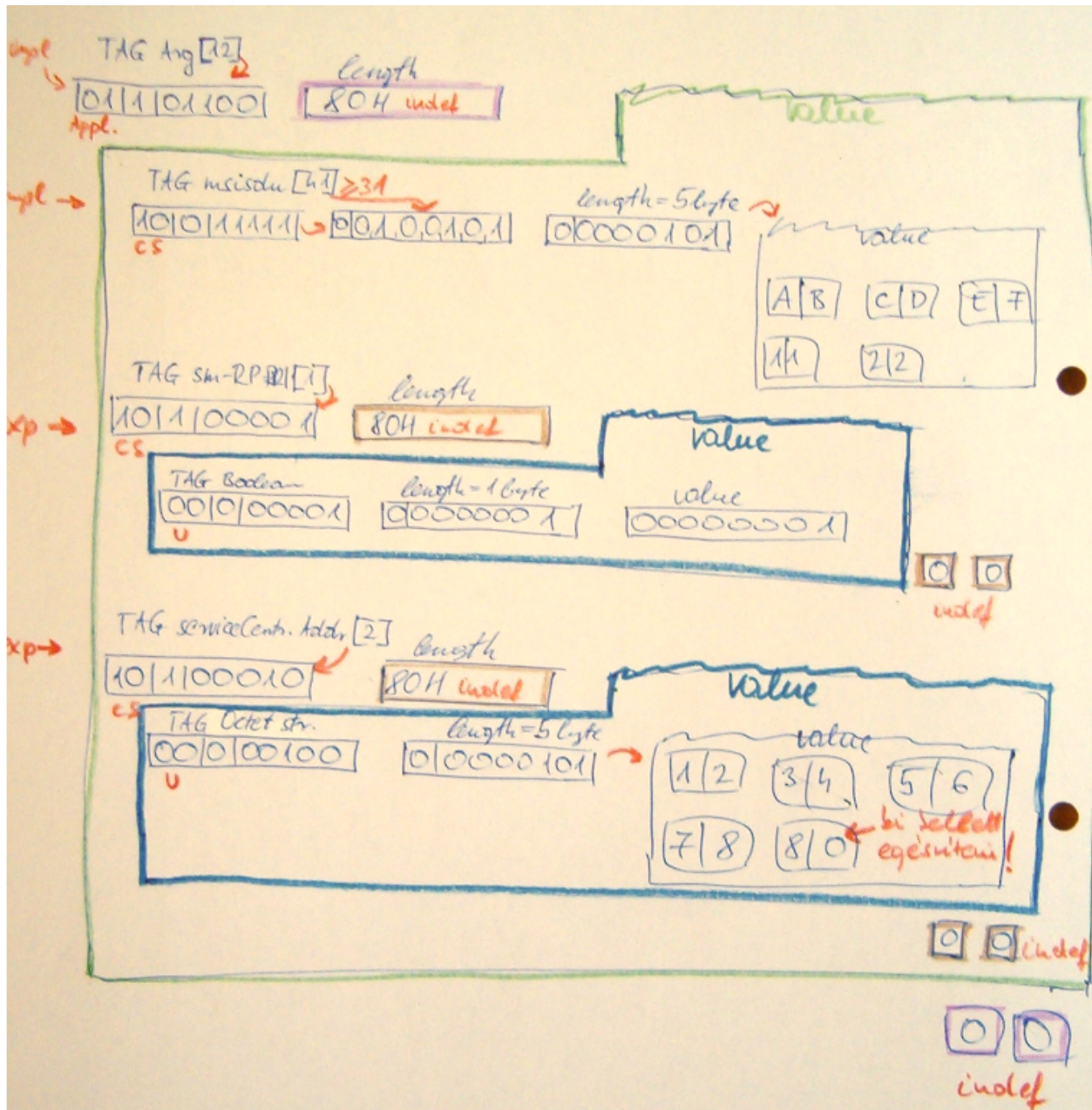
localValue 45

A vonalon a következő értékeket szeretnénk kiküldeni:

msisdn: 'ABCDEF1122' sm-RP-PRI: TRUE (=1) serviceCentreAddress: '123456788'0

a. Milyen típusú TCAP keretben/komponensben küldhetjük a fenti üzenetet?

b. Készítse el a fenti üzenet MAP részének kódolását! Mindenhol, ahol lehet, indefinit hossz kódolást használjon! Segítségül:




```
sendRoutingInfoForSM-Arg [1] SEQUENCE {
    routingInfoForSM-Arg [1] SEQUENCE {
        control [4] IMPLICIT BIT STRING,
        msisdn [0] OCTET STRING (SIZE (1..9)),
        sm-RP-PRI [31] IMPLICIT BOOLEAN,
        serviceCentreAddress [1] INTEGER,
        teleservice [5] OCTET STRING (SIZE (1)) OPTIONAL
    }
}

RESULT
...
ERRORS
...
 ::= localValue 45
A vonalon a következő értékeket szeretnénk kiküldeni:
    control: '011'B
    msisdn: '56781234'O
    sm-PR-PRI: TRUE (=1)
    serviceCentreAddress: 32
```

TAG [1]			LENGTH indefinite																										
11	1	00001	1	0000000																									
TAG Sequence[16]			LENGTH indefinite																										
00	1	10000	1	0000000																									
TAG IMPL [4]			LENGTH		hány db nulla	011'B																							
10	0	00100	0	0000001	0000110	01100000																							
TAG [0]			LENGTH indefinite																										
10	1	00000	1	0000000																									
TAG OCTET STR			LENGTH		VALUE '56781234'	indefinite	indefinite																						
00	0	00100	1	0000010	0000110	00000000	00000000																						
TAG IMPL [31]			[31] miatt	LENGTH		VALUE(true=1)																							
10	1	11111	0	00011111	0	0000001	0000001																						
TAG [1]			LENGTH indefinite																										
10	1	00001	1	0000000																									
TAG INT			LENGTH		VALUE(32)	indefinite	indefinite																						
00	1	00000	0	0000001	00100000	00000000	00000000																						
						indefinite	indefinite																						
						00000000	00000000																						
						indefinite	indefinite																						
						00000000	00000000																						
Decimális érték (egyes mezőkben lévő binárisok értéke)																													
225	128	48	128	132	1	6	96	160	128			0	0	191	130	31	1	1	161	128	32	1	32	0	0	0	0	0	0
E1	80	30	80	84	1	6	60	A0	80			0	0	BF	82	1F	1	1	A1	80	20	1	20	0	0	0	0	0	0
Hexa értékek a decimálisból átválta																													

A pirosas színnel jelölt OCTET STRING értékét nem tudtam, hogy kell elkódolni, így azt még ki kéne javítani, a többi szerintem kb jó-> 56h|78h|12h|34h| összesen 4db. okett
alatta a booleanam sem jó szerintem, tag flag-je 0.
felette meg a bit string sem stimmel, ha a 0-kat ki kell írni, amit szernintem nem, akkor is legalább a lenght-be beleszámít szerintem... de mondom, máshol nem láttam ilyen “hány db. nulla” dolgot.

Protokoll technológia vizsga 2014. máj. 28.

NÉV: _____

NEPTUN KÓD: _____

1. Milyen műszaki megoldások használhatók Magyarországon a vezetékes számhordozás megvalósítására? Ismertesse őket röviden! Röviden magyarázza el, miért nem alkalmazhatók a vezetékes hálózati megoldások a mobil hálózatokban! (4p)
2. Ismertesse egy Location Update lefolyását az NSS-ben! (4p)
3. Miért kell és hogyan lehet GPRS hálózatokban a Paging forgalmat csökkenteni? (4p)
4. A következő ASN.1 leírás egy MAP üzenet (általam kissé módosított) definíciójának részlete:


```

sendI OPERATION
  ARGUMENT
    routingInfoForSM-Arg [31] SEQUENCE {
      msisdn [1] IMPLICIT OCTET STRING (SIZE (1..9)),
      sm-RP-PRI [3] BOOLEAN,
      serviceCentreAddress [22] BIT STRING (SIZE (1..20)),
      teleservice [5] OCTET STRING (SIZE (1)) OPTIONAL;
    }
  ::= localValue 45
      
```

 A vonalon a következő értékeket szeretnénk kiküldeni:


```

msisdn: '212121'0
sm-RP-PRI: FALSE
serviceCentreAddress: '0101010101'B
      
```

 a.) Milyen típusú TCAP keretben/komponensben küldhetjük a fenti üzenetet? (2p)
 b.) Készítse el a fenti üzenet MAP részének kódolását! **Mindenhol, ahol lehet, indefinit hossz kódolást használjon!** (10p)
 Segítségül:

Class: - 00 = Universal - 01 = Application wide - 10 = Context-specific - 11 = Private Format: - 0 = Primitive - 1 = Structured	TAG kódok: * 1 = BOOLEAN * 2 = INTEGER * 3 = BIT STRING * 4 = OCTET STRING * 5 = NULL * 10 = ENUMERATED * 16 = SEQUENCE / SEQUENCE OF * 17 = SET / SET OF
---	--
5. Az ábrán bejelölt nemzetközi hívást akarjuk felépíteni. Az ábrán a telefonközpontokat négyzet, a hívás során felhasznált beszédáramköröket vastag vonal, míg az igénybevett jelzőpontokat / jelzéstovábbító pontokat kör, a használt jelzésszakaszokat pedig szaggatott vonal jelzi.

a.) Rendeljen jelzőpont kódokat az egyes jelzőpontokhoz az adott helyen használatos formátumban!
 b.) Jelölje be, hogy milyen jelzéskapcsolatok épülnek fel! Mi a jelzésüzenetek legfontosabb paramétereinek (DPC, OPC, SIO[NI+SI]) az értéke az egyes szakaszokon? (6p)

Új kérdések:

1. Mutassa be, hogy hogyan küldünk SMS-t hordozott számra!
2. Mi az a GTN? mire használják? 8asszem)