

3. Kapcsolás, protokollok

A kommunikációs hálózatokban a magas színvonalon kiépített kapcsolatközpontoknak még hosszú ideig szerep jut a távközlési forgalom lebonyolításában. Mindazonáltal forradalmi változásnak vagyunk részesei a mobil kommunikáció, az Internet és a multimédia térhódítása terén. Mindezek az új technológiák, eljárások és módszerek nagy értékben építenek a távközlési protokollokra és távközlési szoftverekre.

Jelen fejezet összefoglaló képet ad a kapcsolóközpontok világáról retrospektív módon. (3.1, 3.2. alfejezetek.) A régi kapcsolóközpontok a következők funkcionálisokat tartalmazták:

- A kapcsolót,
- A kapcsoláshoz szükséges intelligenciát,
- A jelutakat, valamint
- A központ működését vezérlő jelzéseket.

Az Internet útvonalválasztói és a jelenlegi kapcsolóközpontok is alapvetően ugyanezen az elven működnek, bár a további funkciók, mint például a számlázás, a működési és forgalmi statisztikák készítése, valamint az automatikus önjavító képességeik a legtöbb esetben már jóval kifinomultabbak.

A jelen fejezetben belül külön alfejezet foglalkozik a digitális kapcsolástechnika hálózati jelzésrendszerével, áttekintő képet ad a mobil kommunikáció rendszertechnikai felépítéséről és protokolljairól. A modern kapcsolástechnika forgalmi kérdésköre újabb nagy kihívást jelent. A hálózatok forgalmi viszonyainak tervezési kérdéskörét tekinti át a **3.3** alfejezet.

A meglévő hálózatokon újabb és újabb megoldásokat és szolgáltatásokat a kommunikációs protokoll segítségével nyújthatunk. A **3.9** alfejezet áttekintő képet ad a protokollok tervezési folyamatáról, melyek a tervezési folyamatban kiemelten fontos szerepet játszanak a távközlési világ által elfogadott és széles körben használt formális leíró nyelvek és módszerek. A fejlett protokoll fejlesztési módszerek eredményeképpen nagyméretű szoftverek és protokollok állíthatók elő, azonban mindenkor ellenőrizni kell a megfelelőséget. A konformancia tesztelés folyamatát tárgyalja a **3.10** alfejezet.

A kommunikációs hálózatok világában kiemelt szerep jut az IP technológiának. Az IP és az általa nyújtott szolgáltatások minőségének témakörét tárgyalják a **3.6**, és a **3.8** alfejezetek. Hálózatainkban igen széles körben használatos az ATM technika, ennek kérdéskörét mutatja be a **3.7** alfejezet.

Jelentős az erőfeszítés napjainkban a hálózati megoldásokon nyújtandó újabb és minőségileg igényes szolgáltatásokra. A következő alfejezetek a kapcsolástechnika, jelzéstechnika, az IP, AM és GSM rendszertechnika tárgyalásán túl nagy teret szentelnek a kommunikációs protokollok világának.

Fejezet szerkesztő: dr. Csopaki Gyula, Dibuz Sarolta

3.1. Kapcsolástechnika áttekintése

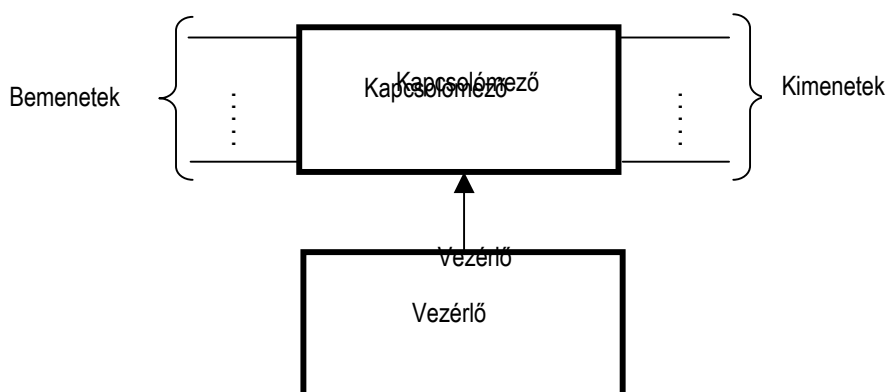
Szerzők: dr. Frajka Béla, dr. Csopaki Gyula

Lektor: dr. Lajtha György

Az áttekintés a kapcsolt közcélú távbeszélő hálózatok (PSTN) térosztásos előfizetői központjainak fejlődését tekinti át. A mai osztályozás szerint ezek a központok az áramkörkapcsolás elvén működnek, amely elve több mint 100 éven keresztül egyeduralkodó távközlő hálózati technika volt. Ezt alkalmazták a későbbi telex, a keskeny sávú ISDN, a mobil, sőt az első adat hálózatok is. Pl. az 1980-as évek elején Budapesten üzembe helyezett tárolt-program vezérlésű adat és telex központ. (NEDIX)

Kapcsolóközpontok fő funkciói

Kapcsolóközpont általános blokkdiagramját a 3.1.1 ábra szemlélteti. Az elvi ábrázolások a kapcsolómezőt általában két oldalasnak tüntetik fel: az egyik oldalon az összeköttetést igénylő áramkörök (csatornák), röviden bemenetek, a másik oldalon az igény kiszolgálására alkalmas áramkörök, kimenetek láthatók. A megvalósítások többsége követi ezt az elrendezést. Léteznek olyan megvalósítások is, ahol mindkét áramkörcsoport egy oldalon található s ezeket visszahajtott (folded back) kapcsolómezőnek hívják.



3.1.1 ábra. Kapcsolóközpont általános blokkdiagramja

A kapcsolómező feladata a bemenetek és a kimenetek időleges összekapcsolása. A kapcsolás lehet térosztásos, frekvenciaosztásos vagy

időosztásos, azaz az egyidejű összeköttetések térben, frekvenciákban, vagy időben vannak a kapcsolómezőben egymástól elkülönítve.

A vezérlő fő feladata az igényelt összeköttetés felépítése a kapcsolómezőben. Az összeköttetést igénylő (előfizetői végberendezés vagy egy másik központ) megadja a rendeltetési hely címét (hívószámát) a vezérlőnek, aminek alapján a vezérlő meghatározza a megfelelő kimenetet, majd a két végpont összekapcsolására rendelkezésre álló kapcsolási utak közül kiválaszt egy szabadot, lefoglalja azt és a megfelelő kapcsoló elemek működtetésével létrehozza a kapcsolatot. A vezérlő egyben értesíti a kimenet másik végét kezelő berendezést az összekapcsolásról. (Analog előfizetői vonal estén csengető jelet küld a vonalon, központok közötti áramkör esetén pedig átküldi a másik központnak mindazon információkat, melynek alapján ez utóbbi az összeköttetést a rendeltetési cím felé ki tudja terjeszteni.)

A kapcsolási funkciók mellett lényeges feladata a központnak az összeköttetés vezérlésével kapcsolatos információk fogadása, kiértékelése és küldése, amit a jelzésrendszerek valósítanak meg. A jelzésrendszerek két nagy csoportba oszthatók: úgymint előfizetői és központok közötti jelzésrendszerekre. Az előfizetői jelzésrendszerek egyszerűbbek, mint a központok közöttiek, részben mert a helyi központ számos jelzést jelzőhangokkal tudat a használókkal és a használók értékelik ki ezek jelentését, másrészt a központok között üzemeltetést támogató jelzések is vannak.

Térosztásos távbeszélőközpontok

Csaknem 2 évvel a telefon feltalálása után, 1878. január 28-án New Havenben (Connecticut, USA) helyezték üzembe az első távbeszélőközpontot, 21 előfizetővel. [3.1.1, 49. o.] Magyarország is igyekezett lépést tartani ezzel a fejlődéssel, hisz első távbeszélőközpontja Budapesten, 1881. május 1-én kezdett szolgáltatni, 50 előfizetővel, majd ezt rövidesen további budapesti és nagyobb vidéki városok központjai követték [3.1.3, 66. o.]

A térosztásos központok változatai kézikapcsolású, elektromechanikus és tárolt-program vezérlésű kategóriákba sorolhatók.

A **kézikapcsolású** (manuális) központokban az általános blokkdiagram két fő egysége a valóságban is elkülönült: a kapcsolószekrényre és a kiszolgáló kezelőre. A fejlett konstrukciókban az előfizetői vonalak a kapcsolószekrény függőleges

síkjában szerelt kapcsolóhűvelőkben végződtek (mellettük az előfizető hívását jelző eszközzel) és összekapcsolásukat a kezelő dugasz párban végződő zsinóráramkörökkel végezte. Kezdetben az előfizetői készülékek mikrofonjának táplálása helyi telepről (Local Battery) történt. A hívó előfizető hívási szándékát a készülékében lévő induktorral jelezte a központnak, s kezdetben a kezelő is induktorral csengette fel a hívottat.¹

Az 1890-es években a műszaki fejlődés eredményeként az LB táplálást fokozatosan a CB (Central Battery) táplálás váltja fel, s a tápáram egyúttal automatikusan közvetíti az előfizető jelzéseit (a kézibeszélő helyzetét) a központba. A kézibeszélő felemelésekor egy érintkező pár egyenáramúlag zárja az előfizetői hurkot s a központban az egyenáram megindulását egy jelfogó érzékeli, ami egy lámpát gyújt ki a kapcsolóhűvelő mellett.

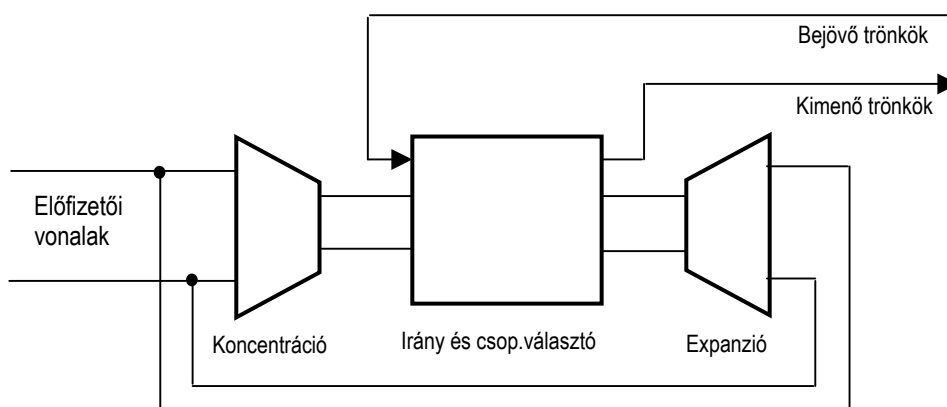
A CB rendszerű központok automata zsinóráramkörei a kezelők munkájának egy részét is átvették s ezáltal a kezelők lényegesen több hívást tudtak kezelni. A CB rendszer nagyobb előfizetői vonal kapacitású központokat, jobb szolgáltatást, kezelők hatékonyabb kihasználását, stb eredményezte.

Az **elektromechanikus** központok a kapcsolástechnika 2. generációját jelentik. A sok rendszer közül csak egy-két markáns központot és a hazai hálózatban is alkalmazott rendszert mutatunk be.

A központok megjelenése után rövidesen többen is (főleg az USA-ban) foglalkoztak a kezelők géppel történő kiváltásával. Az első, gyakorlatban is használható megoldást Almon B. Strowger szabadalmaztatta (1891). [3.1. 61.] A **STROWGER** kapcsológép egymás fölé helyezett 10 szinten (emeleten), szintenként félkörívben elrendezett 10, összesen 100 ívpontot és az ívpontokkal érintkezésbe hozható surlódó keféket tartalmazott. A keféket először a megfelelő szintre, majd a szinten belül a megfelelő pozícióba az előfizetők közvetlenül vezérelték. A kefék mozgását egy emelő és egy forgató mágnes végezte. A mágnesek egy-egy gerjesztésre egy-egy pozícióval (emelés, forgatás) változtatták meg a kefék helyzetét. Ezt a működési módot lépésenkénti (step-by-step, SxS) működésnek nevezik. A választó gép elé egy újabb választó fokozatot iktatva, a 10 emeletnek megfelelően 10 vonalválasztó gépcsoport közül lehet választani, s így a központ kapacitása 1000

¹ A mai központokban használt csengető jel paraméterei (feszültség, frekvencia) innen származnak.

vonalgig bővíthető. A Strowger rendszerben egy helyi központ kapacitásának felső határa 10 000 vonal, ami további csoportválasztó fokozat beiktatásával érhető el.



3.1.2 ábra. Helyi központok kapcsolómezőjének szerkezete

A csoportválasztó gépek számát nagy mértékben lecsökkentették, amikor koncentráció fokozatot iktattak be az előfizetői vonalak és az első csoportválasztó fokozat közé. Ezzel kialakult a léptető gépes helyi központok kapcsolómezőjének a 3.1.2 ábrán látható általánosan használt szerkezete.

Azokban a központokban, ahol a kapcsolási út kiválasztása és a kapcsoló elemek vezérlése teljesen elkülönül a kapcsoló elemtől, így a cross-bar és a tárolt-program vezérlésű központokban, a külön koncentrációs és expansziós fokozatot egyesítették. Az egyesített fokozatot előfizetői fokozatnak hívják.

A SxS rendszerek, különböző márka név (Strowger, Siemens, Angliában "director", stb.) alatt az összes kontinensen elterjedtek. Az angol director rendszertől eltekintve, valamennyi a **közvetlen vezérlést** alkalmazta. A közvetlen vezérlés előnye egyszerűségében rejlett, viszont az előfizetők hívószáma a központ kapacitásától és a központ hálózati hierarchia helyétől függően különböző hosszúságú (nyílt számozás) volt. A távhívás megjelenésekor ez kényelmetlenséget okozott az előfizetőknek. (Ugyanazt az előfizetőt más-más helyről esetleg más-más számmal lehetett elérni.)

A forgókefés kapcsológépes rendszerek másik nagy családja a **közvetett vezérlési** elvet alkalmazza és motor hajtású rendszereknek is szokták nevezni. A kapcsológépek keféinek mozgását közös tengely végzi.

A hívó előfizetőtől jövő választási információt a regiszter fogadja és tárolja, majd vezérli az összeköttetés felépítését. A közvetett vezérlés lehetővé teszi, hogy a kapcsológépek vezérlése a tízes számrendszertől eltérjen.

A közvetett vezérlés mellett műszaki (egyszerűbb kapcsológép), és telefon forgalmi érvek szóltak. Megfigyelték, amit később Erlang B formulája is igazolt, hogy a forgalmas órában a kiszolgáló árákörök kihasználtsága (teljesítménye) emelkedik, ha az egy csoportban lévő kiszolgálók száma növekszik. Pl.: $B = 0,005$ blokkolási valószínűség mellett egy tízes nyaláb áramköri használtsága 0,4 erlang, a huszasé 0.55 erlang, a harmincasé pedig 0.63 erlang. A teljesítmény növekedés lényegesen gazdaságosabb trónkhálózatot, a központon belül pedig kevesebb kapcsológépet eredményez.

A közvetett vezérlésű forgókefés kapcsológépes rendszerek között a legismertebb az ITT Rotary központ családja (a rendszer családot a "7" szám, család tagjait pedig a 7 mellett szereplő betűk jelölték. Ismertebbek: 7A (nagyvárosi), 7DU (kisvárosi) központ) és az LM Ericsson AGF központja. A 7A rendszer és az AGF rendszer a választó fokozatok vezérlését revertív (kapcsológépek mozgásuk közben a beszéd vezetékeken impulzusokat küldenek vissza a regiszterbe) impulzusokkal végezte.

A **ROTARY** 7A-1² rendszer továbbfejlesztett változatában, a 7A-2 rendszerben a választógép kapacitása 300 ívpont, amit 10 emeletre osztottak fel, egyenként 30 ívponttal. A választás itt is két-irányú, de az érintkező kefe emelését a kefekiváltás (aktiválás) helyettesíti. (A 10 emeletnek megfelelően 10 kefe készletet tartalmaz a kefeszán.)

A választógép vonalválasztó funkcióban 400 állomás (200 fő- és 200 ikerállomás) kapcsolását végzi. A középső 100-as ívcsúcs blokkra vannak kötve a közösívpontú ikerállomások vonalai. Az ikerállomások alkalmazásának elsődleges célja az előfizetői vonalak számának csökkentése. A magyar mérnökök által kifejlesztett megoldás egyúttal a központ kapacitásának bővülését is eredményezte. [3.1.3, 101. o] Két csoportválasztó fokozattal a központ kapacitása 40 000 előfizető. A koncentrációs fokozatban, valamint a regisztert kapcsoló fokozatban egy-mozgású, kereső típusú gépet alkalmaztak.

² A magyar távbeszélő hálózatot Rotary rendszerű központokkal automatizálták. Az első 7A központot 1928-ban helyezték üzembe a Krisztina épületében.

A Rotary család másik bevált tagja a 7DU központ. Ebben egy-mozgású, 100 ívpontos kereső típusú kapcsológépet (egyszerűbb gyártás) alkalmaztak minden kapcsolási fokozatban és a kapcsológépeket a regiszter előre irányú impulzusokkal vezérelte. A regiszterből küldött számjegyeket nem a kapcsológépek fogadták, hanem a vezérlő áramkörök, amelyek több (pl. 10) kapcsológép számára közősek voltak. A vezérlő áramkör a fogadott számjegy(ek) alapján elektromosan megjelölte azt az ívcsúcs (vonálválasztás) vagy ívcsúcs csoport (csoport választás) pozíciót, amelyre a gép keféjét vezérelni kellett, majd a gép megkereste a megjelölt pozíciót. Az egybefüggő ívcsúcs blokk lehetővé tette, hogy a CSV fokozatokban a tízes nyalábtól eltérő nyalábokat is lehetett képezni, ha ezt a forgalmi viszonyok igényelték.

Az **AGF rendszer** kapcsológépe 500 "ívpontos", 25x20-as csoportosításban. Az ívpontokat és a multiplikációt (egy csoportot alkotó gépek azonos ívpontjainak összekábelezése) kereteken kifeszített, kellően merev csupasz huzalok helyettesítik, ami jelentős anyag és munka megtakarítást eredményezett. A multiplikáció teljesen el van választva a kapcsológéptől, s húszas kereteket alkotnak. A keretek, félkörnél kisebb körív mentén, sugár irányban vannak függőlegesen felállítva.

A lapos kapcsológépen a kefeszerelvény egy forgó korongra van szerelve, amelynek elfordulásával lehet a keféket a kívánt sorra állítani, majd ezt követően sugár irányban mozgatni a keféket. A kapcsolás befejezése után a kefeszerelvény ellenkező irányú mozgást végez, amit az állandóan forgó tengelyhez történő kapcsolódás forgás irányának megfordításával értek el. A gépek dugaszolható kivitelben készültek, így az egy kereten lévő gépek száma rugalmasan változtatható (pl. bővítés), a meghibásodott gépek egyszerűen cserélhetők. A központban ugyanazt a gép típust használják a csoportválasztó, a vonálválasztó és a hívás koncentráló fokozatban. Ez utóbbi azt jelenti, hogy az előfizetői vonalak egyetlen multiplikációhoz csatlakoznak, és ugyanazon a kereten található a vonálválasztó és a híváskereső gépek. A rendszer robusztus konstrukcióját bizonyítja, hogy az 1930-as években Miskolcon felszerelt központ az 50-es években Egerbe történt áttelepítése után is jól működött.

A távolsági összeköttetések terjedése, valamint az előfizetők számának növekedése rávilágított a forgókefés kapcsológépek gyengéire: lassú működés, zajosság és nagy karbantartási munka igény (5 óra/év/előfizető). Ezek csökkentésére

az elektromechanikus technológiában a jelfogó a legalkalmasabb, viszont a jelfogós kapcsolómátrixok költségesek, a keresztpontonként szükséges mágnesek miatt

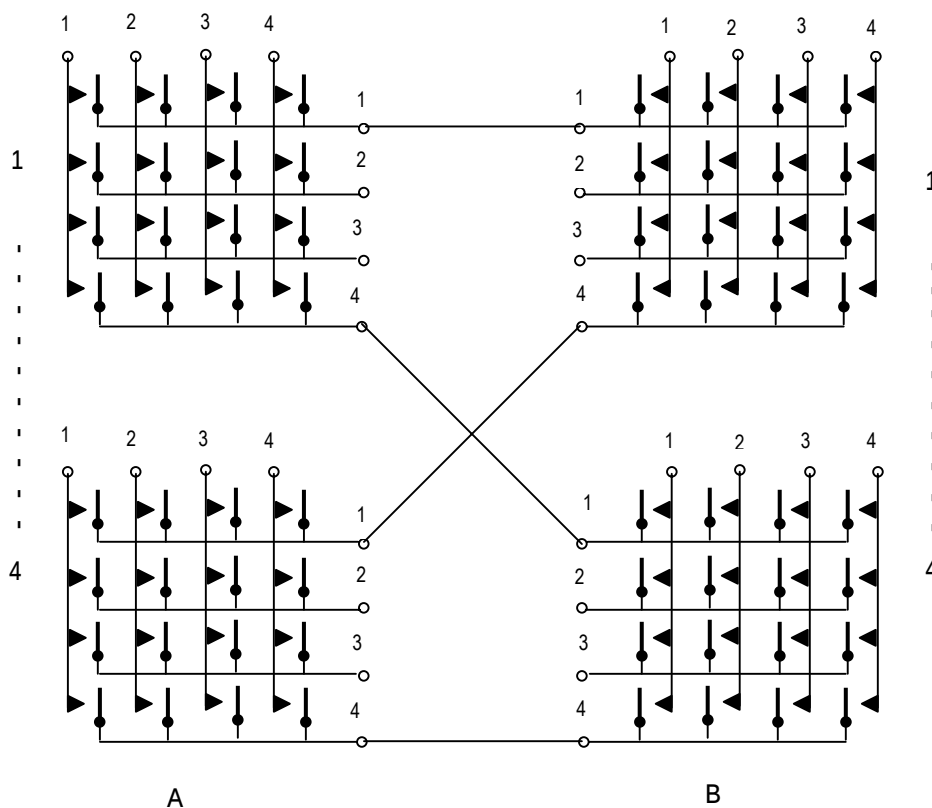
A 3.1.3 ábrán egy-vezetékes, jelfogó érintkező szimbólummal ábrázolt 4x4-es kapcsolómátrixok láthatók, az egyszerűség kedvéért a működtető mágnesek feltüntetése nélkül. (A koordináták metszéspontjában elhelyezett kapcsoló elemet az irodalom keresztpontnak nevezi.) Egy adott be- és kimenet összekapcsolása a metszéspontjukban található keresztpont működtetésével (zárásával) és az összeköttetés ideje alatti zárva tartásával (kapcsolási memória) történik. A függőlegesek számát n -el, a vízszintesek számát m -el jelölve, a keresztpontok egyedi vezérlése $n \times m$ mágnes igényel.

Koincidenciás szelekcióval a mágnesek száma $n+m$ értékre csökkenthető, ha a mágnesek hatását a keresztpontokra valamilyen alkalmas módszer továbbítja, például rudak kis mérvű elfordulása. A **crossbar** kapcsoló ezekről a rudakról kapta elnevezését. (A crossbar gépet a svéd Betulander már 1919-ben megalkotta.) Az oszlop mágnes a rúddal és a függőleges mentén elhelyezett érintkezőket egy közös mechanikai egységként gyártják, az álló érintkezőket multiplikálva lemez vagy huzal formájában. Az egységet hídnak nevezik, amit felfoghatunk, mint egy m rugócsomagos jelfogó. A vízszintes, vagy más néven jelölő mágnes feladata kijelölni, hogy a híd melyik keresztpontja záródjék. A jó minőségű érintkezés céljából az érintkezők nemesfémből készültek, ezért gazdaságossági szempontok kis keresztpont számú (10, vagy 20) hidak alkalmazását indokolták.³ Az egy közös mechanikai vázra szerelt hidakat a hozzátartozó jelölő mágnesekkel és rudakkal nevezik crossbar gépnek. A hidak vízszintes kimeneteinek multiplikálásával tetszőleges méretű mátrix képezhető, a közös mechanikai váz és jelölés a mátrix méretét nem befolyásolja.

Nagyobb kimenetű kapcsolók kis keresztpontszámú hidakból készített kapcsolómátrixok kaszkádba kapcsolásával építhetők. A 3.1.3 ábrán egy 4x4-es mátrixokkal képzett 16x16-os, kétfokozatú linkkapcsolás látható. A beépített keresztpontok száma 128. Ugyanilyen kapacitású kapcsolót megvalósító 16x16-os kapcsolómátrix 256 keresztpontot igényel. A linkkapcsolás csökkenti a kapcsolómezőben szükséges keresztpontok számát, azonban ennek ellentétetele

³ Pl. egy N ívpontos forgó kefések kapcsológéppel létesített kapcsolatnál 1 ívpont van hasznosan lefoglalva és $(N-1)$ fölöslegesen. Foszforbronzból készült ívpontnál ez még elviselhető volt.

van: szabad kimenet választása feltétel függő, azaz az oda vezető link is legyen szabad.

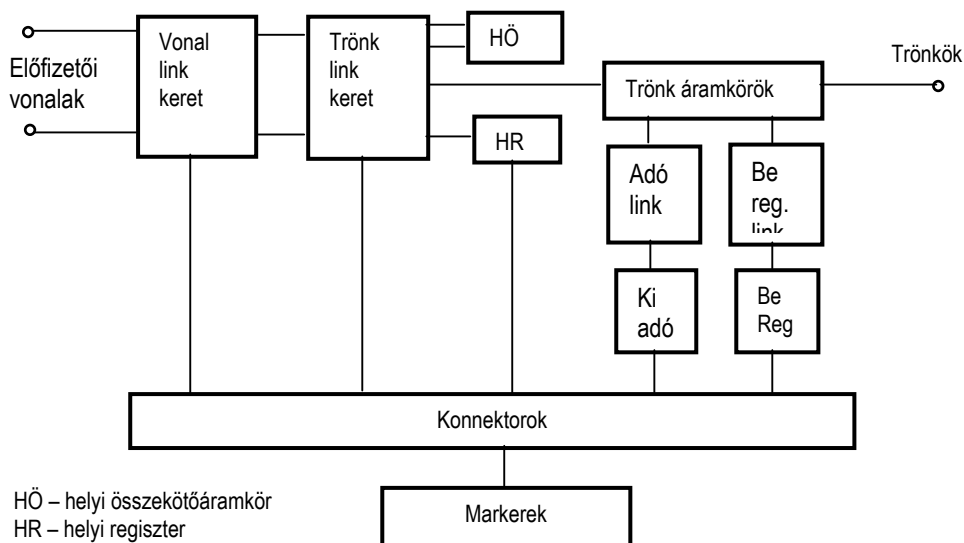


3.1.3 ábra. Kétfokozatú linkkapcsolás

Készíthetők 3, 4 vagy még több fokozatú kapcsolások is. (Ki- és bemenetek számának emelése, vagy az összekapcsolási lehetőségek – utak – növelése céljából.) Linkkapcsolást a crossbar rendszerekben vezették be s azóta minden nagykapacitású kapcsolómező (még a digitális is) ezen az elven készül.

A crossbar rendszerű központok az 1930-as évek végén jelentek meg az USA-ban. Az AT&T crossbar helyi központjának fejlett változata a **No 5 Crossbar rendszer** [3.1.4, 8. Fejezet] megnevezést kapta, s vázlatos blokkdiagramja a 3.1.4 ábrán látható.

A kapcsolómezőt a vonal link és a trónk link modul alkotja. Mindkét modul kétfokozatú linkkapcsolás, a kapcsológép 20 drb 10 keresztponthoz tartozik.



3.1.4 ábra. No. 5 Crossbar egyszerűsített blokkdiagramja

A rendszerben a vezérlési funkciókat tovább bontották. A viszonylag hosszú tartásidejű regiszterekből kiemelték a bonyolult összeköttetés vezérlési funkciókat és azokat a markerek valósítják meg. Ezek tartásideje rövidebb és így sokkal kevesebb kell belőlük, mint regiszterekből. Viszont nagyon bonyolultak. A nagy bonyolultságú marker áramkörök tervezése tudományos módszerek alkalmazását igényelte. Ekkor vette kezdetét a logikai áramkörök tervezésének elmélete.

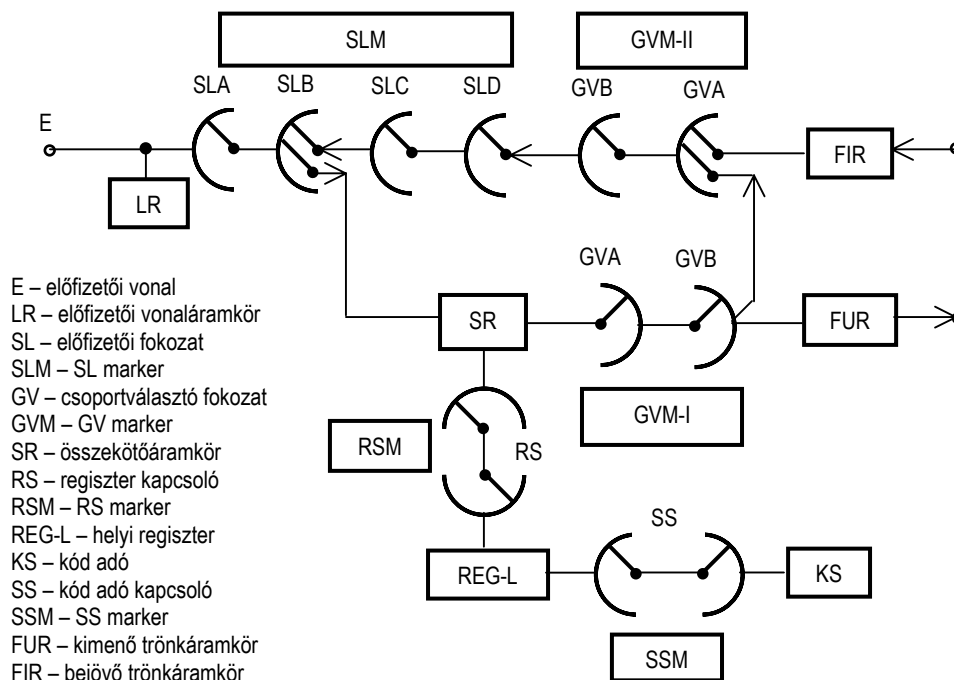
A kapcsolómező vezérlésnél a **közös vezérlési** elvet alkalmazták. Szabad kapcsolási út keresése majd felépítése a két modulban egyidejűleg történik. A közös vezérlésű kapcsolómezőben kevesebb kapcsolási út szükséges, mint a fokozatonként vezéreltben. A markerek a sok érintkezős konnektor áramkörökön keresztül kapcsolódnak a modulokhoz, a regiszterekhez és a jelzés adókhoz.

Európában az első crossbar központot az LM Ericsson helyezte üzembe az 1950-es évek elején. [3.1.1, VIII. Fejezet] Az AR⁴ crossbar család nagyvárosi központjának, az **ARF 102** központnak a blokkdiagramját mutatja a 3.1.5 ábra. Az Ericsson crossbar gép 10 drb 20 keresztpontos hidat és 10 jelölő és 2 választó mágneset tartalmaz. Az egyszerű áttekinthetőség kedvéért a diagramokon a hidat a forgókefés technikából ismert szimbólummal ábrázolták. A hídnak a kefe, a keresztpontoknak az ívpont szimbóluma felel meg, s így egyszerűen nyomon követhető a fokozatok közötti linkek bekötése.

⁴ Magyarország 1968-ban megvásárolta a rendszer licencét.

3.1.5 ábra. ARF 102 központ blokkdiagramja

Az ARF rendszer fokozatonként vezérelt, aminek eredményeként az egyes fokozatokban valamivel több kapcsolási út szükséges, viszont lényegesen



egyszerűsödnek a marker áramkörök. Az előfizetői fokozat a kimenő forgalmat két fokozattal (SLA-SLB) koncentrálja, a végződő forgalmat pedig 4 fokozaton át expandálja. Az alap kétfokozatú csoportválasztó modul kimeneteinek száma egy harmadik fokozat hozzáadásával tovább bővíthető, a kapacitás és a forgalmi igényeknek megfelelően.

Gazdasságossági megfontolásból és a különböző jelzésrendszerű központok együttműködési követelménye miatt ebben a rendszerben is kiemelésre került a regiszterből a választási információt továbbító funkció, amit a kódadók valósítanak meg. Az AR rendszer regiszterközi jelzésrendszere az R2 MFC jelzésrendszer. A kódadó kapcsoló fokozat segítségével a regiszter hívás felépítés közben is változtathatja a kódadót (jelzésrendszert). A kódadó különválasztása a regisztertől azért is előnyös, mert az üzemeltetés folyamán szükségessé váló változtatásokat kevesebb helyen kell elvégezni.

Crossbar rendszerű központot még számos vállalat dolgozott ki. Ezek rendszertechnikailag nem hoztak lényeges újdonságot az előzőekhez képest. Bár nem nyert tömeges alkalmazást, de említésre érdemes a magyar ipar által az 1960-

as évek első felében kifejlesztett elektronikusan vezérelt rural központ rendszer, az ECR család. [3.1.5]

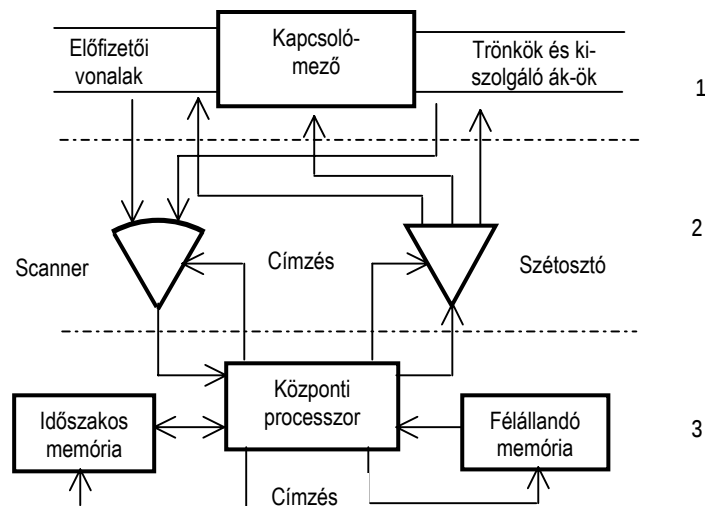
Az első **tárolt-program vezérlésű központ**, A Bell Laboratóriumokban kifejlesztett **No. 1 ESS** (Electronic Switching System) [3.1.5, 11. Fejezet] központot 1965. májusában helyezték üzembe világméretű intenzív kutató munka eredményeként. [3.1.2, II. és V. fejezetek] A kutató munka a telefonközpontok elektronizálására irányult. A huzalozott-program vezérlésű (mai terminológia szerint) crossbar központok bevált rendszertechnikájára alapozva, a nagyobb megbízhatóságú elektronika nagy sebességétől a hardver igény további csökkenését, azaz gazdaságosabban gyártható és üzemeltethető központot reméltek. A kor elektronikai alkatrész választékával (vákuum csövek, gázzal töltött hidegkatódos csövek, korai tranzisztorok és diódák) a teljes elektronizálás nem volt megvalósítható. Pl. az elektronikus keresztpontú kapcsolómezőhöz teljesen új készülékeket kellett kifejleszteni, ami a gazdaságosság ellen hatott, mert ebben az időben már hatalmas mennyiségű készüléket kellett volna lecserélni a meglévő központok kiváltásakor. Továbbá a nagy térosztásos elektronikus kapcsolómező nem teljesítette az átviteltechnikai követelményeket sem. (áthallás.)

A Bell Laboratóriumokban felismerték, hogy a gyártó és az üzemeltető követelményei (gazdaságosság, szolgáltatások bővítése és egyszerű bevezetése, stb) a tárolt-program vezérlésű (Stored Program Control) központokkal teljesíthetők. A hívások vezérlése egyszerű logikai (és aritmetikai) műveletek szekvenciájával leírható, amit memóriában lehet tárolni és előhívni. A processzor idejének jó kihasználása érdekében a keresési és választási műveletek ne a tényleges kapcsolómezőn, hanem annak a memóriában leképzett tükörképén történjen, (map-in-memory technika) elkerülendő a gyakori és lassú input/output műveleteket. A rendszer vázlatos blokkdiagramja (3.1.6 ábra) gyakorlatilag a 3.1.1 ábrán látható felosztást (1 és 3 rész) tükrözi vissza, kiegészítve a két rész közötti kommunikáció eszközeivel (2 rész).

A kapcsolómező a rövid áramimpulzussal vezérelhető, mágnes tartású, fémes érintkezőjű ferreed (ferrite + reed) keresztpontú mátrixokból készült.

A központi processzor diszkrét félvezetőkől készült s így nagyon drága volt. Ezért minden információ kiértékelési, feldolgozási és tárolási feladatot ebbe vontak össze, (centralizált vezérlés.), a kapcsolómezőhöz csatlakozó áramkörök csak

minimális hardvert tartalmaztak. A hívások feldolgozásával kapcsolatos adatokat ferrit memória, a programokat és a félállandó (központ függő) adatokat twistor (nem destruktív) memória tárolta.



3.1.6 ábra. No.1 ESS vázlatos blokkdiagramja

A központ biztonságos üzemét (max 2 óra össz kiesés 40 év alatt) a vezérlő rendszer (processzor és memóriák) duplikálásával biztosították. A vezérlők szinkron duplex üzemmódban működnek: mindkét processzor ugyanazt a hívást dolgozza fel, de csak az egyik, az aktív vezérli a telefonos perifériákat. Ha hiba, vagy egyéb (operátori parancsra) ok miatt át kell váltani a másik vezérlőre, ez hívás elvesztése nélkül megvalósítható.

A scanner periódikusan letapogatja a telefonos perifériák állapotát, ezeket a következő periódusig memóriájában megőrzi, hogy a szolgáltatási igényt jelölő állapotváltozásokat meg tudja állapítani. (Eseményt mindig állapotváltozás jelent.) A valós idejű működési követelmény teljesüléséhez hívásdetektálás céljából az előfizetői vonalakat átlagosan 100 ms-ként, a tárcsázási fázisban lévő vonalakat pedig szigorúan 10 ms-ként tapogatta le a processzor. Ez a gyakori, de egyszerű rutin feladat a processzor idejének jelentős részét kötötte le, ami miatt a központ vonalkapacitása jelentősen elmaradt a tervezettől. A hívás feldolgozási kapacitás növelése érdekében a továbbfejlesztett változatban a rutin feladatokat egy előfeldolgozó processzor, a Signal Processzor (szintén duplikált) végezte és adta át az eseményeket további feldolgozásra a központi processzornak.

A technológiai fejlődés eredményeit beépítve (1A processzor, remreed, stb) a rendszer méretei, fogyasztása, gyártási és üzemeltetési költségei jelentősen csökkentek. A 60-as évek második felében több központ gyártó cég fejlesztett ki tárolt-program vezérlésű központot. Ezek rendszertechnikában gyakorlatilag a No. 1 ESS-t követték. Kapcsolómezőben szintén fémes érintkezőt használtak, a reed jelfogó áramtartásos változatát, vagy crossbar elven működő, áram nélkül tartó (mágnesek méretének csökkentése céljából) speciális kapcsolókat. A francia CIT Alcatel cég a duplikált processzorok üzemmódjával a terhelés megosztásos (load sharing) üzemmódot alkalmazta. A mikroprocesszorok árának csökkenését követően a teljesen centralizált vezérlést felváltja a két szintű vezérlés. A hardver közeli funkciók, a rutin műveletek a telefonos periféria vezérlőkbe helyeződnek ki. Ezek a periféria vezérlők a No. 1 ESS szétosztott (distributed) SP processzorának tekinthetők. Ezen központok rendszertechnikáját úgy dolgozták ki, hogy a technológia fejlődés eredményei (pl időosztásos kapcsolás) rendszertechnikai változtatás nélkül beépíthetők legyenek. (l. 3.2 fejezetben ismertetésre kerülő központokat.)

Időosztásos kapcsoló központok.

A távbeszélő központok elektronizálása már a kezdetektől párhuzamosan folyt a térosztásos valamint az időosztásos kapcsolómezővel. Az időosztás esetén nincsenek áthallási problémák és az elemek többszörös kihasználása révén gazdaságosabb kapcsolómezővel kecsegtetett. A kísérleti rendszerek, egy kivétellel, a PAM technikát alkalmazták. Két rendszert érdemes megemlíteni ezek közül: az egyik az 1960-as évek elején kísérleti üzembe helyezett angol Highgate Wood-i központ, [3.1.2, II-5 fejezet] ami teljes kudarccal járt. (A PAM rendszer áthallása miatt.) Ez azért volt jelentős, mert az angol posta úgy döntött, hogy átugorja a crossbar rendszereket és a szükséges hálózati modernizálást időosztásos központokkal valósítja meg. A másik a Bell. Lab. No. 101 ESS nevű alközpontja, amely egyidőben készült el a No. 1 ESS központtal és nagy mértékben támaszkodott a No. 1 ESS technológiájára. [3.1.2, V-1 fejezet].

A PCM átviteli rendszerek elterjesztésével egyidőben a Bell Lab-ban kísérletet folytattak a digitális átviteli utak csatornáinak közvetlen összekapcsolására. A kísérleti rendszert (ESSEX) [3.1.2, VIII-2] az első generációs (koncentrálás térosztásos fokozat végezte) digitális központok előfutárának lehet tekinteni. A

kísérleti modell igazolta a digitális időosztásos kapcsolás megvalósíthatóságát, de a megfelelő alkatrészek hiánya miatt túl költséges volt.

Az olcsó félvezető memóriák tették gazdaságossá a digitális kapcsolást, először a tranzit központokban, ahol vagy már digitalizált átviteli utak, vagy csoportos kodekekkel gazdaságosan digitalizálható analóg trónknyalábok között kell kapcsolatot létesíteni. Az első digitális tranzit központot, a Bell Lab. No. 4 ESS központját 1976-ban helyezték üzembe. A VLSI technika látványos fejlődése a 80-as években, gazdaságosan megvalósíthatóvá tette az egyedi kodekeket, aminek eredményeként a digitális technika az előfizetői vonal csatlakozásáig elér, azaz létrejön az IDN (integrált kapcsolás és átvitel), majd az ISDN.

Irodalomjegyzék

[3.1.1] Robert J. Chapuis, 100 Years of Telephone Switching (1878-1978), Part 1: Manual and Electromechanical Switching (1878-1960's), North-Holland Publishing Company, 1982.

[3.1.2] Robert J. Chapuis, Amos E. Joel, Jr., Electronics, Computers and Telephone Switching. A book of technological history as Volume 2: 1960-1985 of "100 Years of Telephone Switching", North-Holland Publishing Company, 1990.

[3.1.3] Postamérnöki szolgálat 50 éve, 1887-1937, 2. kiadás az eredeti alapján, Budapest, 1990, Távközlési Könyvkiadó.

[3.1.4] Richard A. Thomson, Telephone Switching Systems, Artech House, 2000.

[3.1.5] A Magyar Híradástechnika Évszázada, MTESZ Házinyomda.

3.2. DIGITÁLIS KAPCSOLÓKÖZPONTOK

Szerzők: dr. Seres Péter

Lektor: dr.Frajka Béla

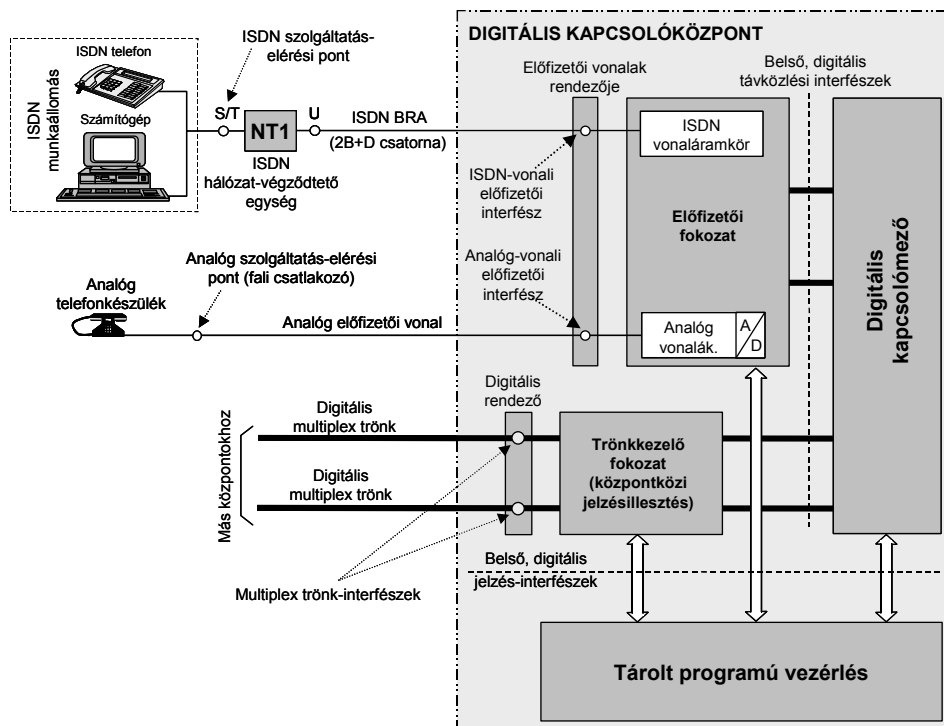
3.2.1. Bevezetés

A *digitális* kapcsolóközpontok – a hálózatban betöltött forgalmi feladataiktól függően – lehetnek előfizetői központok vagy különböző célú tranzit-központok (trónkközpontok), de számos esetben *kombinált központok*ként működnek, egyetlen központban egyesítve az eltérő forgalomtechnikai feladatokat. A kombinált alkalmazást elsősorban a digitális központok tárolt programú vezérlése teszi lehetővé. A tranzit-központi funkcióval kapcsolatban meg kell még jegyezni, hogy a hazai hálózatban ez a funkció (a magyar hálózat sturtúrájából adódóan) egyaránt jelenthet primer-, szekunder- vagy tandem-központi híváskezelést.

A fejezet további szakaszaiban a korszerű, digitális kapcsolóközpontok felépítési elvét (architektúráját) és legfőbb rendszertechnikai jellemzőit tekintjük át

3.2.2. Digitális kapcsolóközpont felépítése

Vizsgálatainkat a 3.2.1. ábrán látható központ-architektúra alapján végezzük el. Az ábrán látható funkcionális egységeknél nem tüntettük fel, hogy azok hardver-, szoftver vagy „vegyes” felépítésűek-e, mert ez erősen rendszerfüggő jellemző. Tárgyalásunk során kizárólag annak elemzésével foglalkozunk, hogy melyek azok a főbb feladatok (funkciók), amelyeket a rajzon árnyékolt vagy árnyékolás nélküli négyszöggént feltüntetett egységeknek végre kell hajtaniuk (az árnyékolás kizárólag azt a célt szolgálja, hogy a funkcionális egységeket egymástól jól látható módon lehessen elkülöníteni).



3.2.1. ábra. Digitális kapcsolóközpont funkcionális felépítése

A feltételezett, *kombinált funkciójú* digitális kapcsolóközpontban mind előfizetők, mind pedig más kapcsolóközpontok (előfizetői és tranzit-központok egyaránt) csatlakozhatnak. A központ ebből adódó feladatait ellátó főbb funkcionális egységek a következők:

- előfizetői fokozatok, amelyek az előfizetői vonalakkal kapcsolatos illesztési és forgalom koncentrálási feladatokat látják el,
- trónkkezelő fokozatok, amelyek a központközi jelzések kezelését végzik,
- digitális kapcsolómező, amely a kapcsolási funkciókat teljesíti, valamint
- tárolt programú vezérlés (TPV), amely a központ működését (a hívások komplex kezelését) irányítja.

A központban az alábbi, forgalmi esetek alakulhatnak ki, amelyek mindegyikét (bármilyen egyidejű kombinációban fordulnak azok elő) kezelnie kell a vezérlésnek:

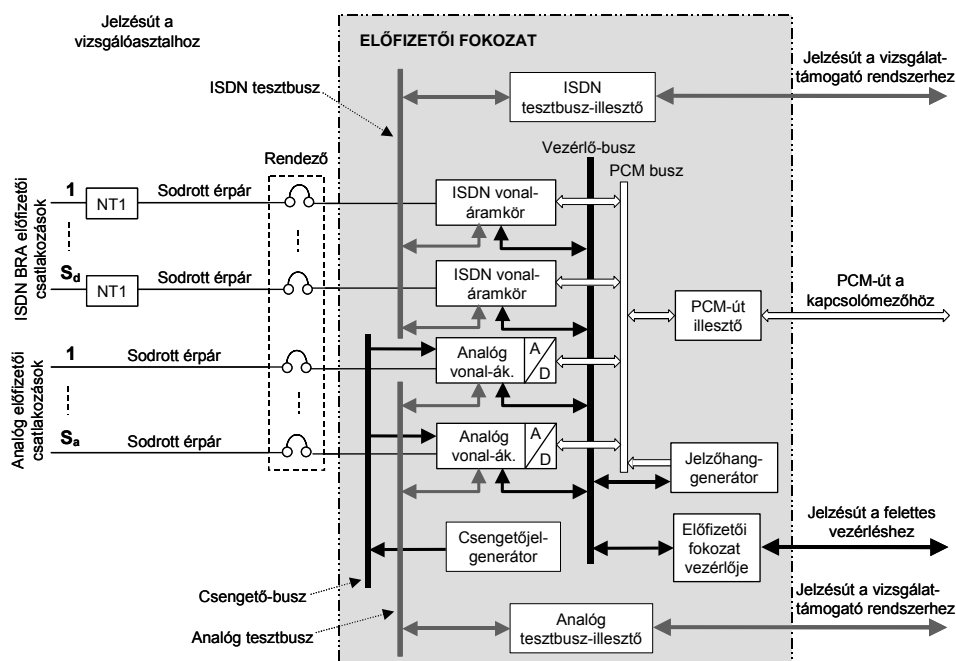
- helyben maradó és kimenő forgalom (kezelése előfizetői központ-funkció),
- bejövő, előfizetői vonalakon végződő forgalom (kezelése ugyancsak előfizetői központ-funkció) és
- bejövő tranzit forgalom (kezelése tranzit-központi funkció).

3.2.3. Főbb kapcsolástechnikai egységek

Ebben a szakaszban a digitális központnak azokat a főbb elemeit tárgyaljuk, amelyek egy-egy összeköttetés *felépítését* valósítják meg.

a) Előfizetői fokozat (digitális/ISDN és analóg előfizetői csatlakozás)

A fokozat funkcionális felépítését a 3.2.2. ábra mutatja be. Látható, hogy a fokozaton belül a kétféle előfizetői csatlakozási módot támogatva – kétféle (vö.: ISDN és analóg) előfizetői *vonáramkör* típus, ennek megfelelően kétféle *vonalkártya* típus különböztethető meg. A kártyákon lévő vonáramkörök száma a központ típusától, illetve ezen belül az alkalmazott gyártási technológiától függ. Jellemző a 4, 8 vagy 16 vonáramkört tartalmazó vonalkártyák alkalmazása.



3.2.2. ábra. Előfizetői fokozat funkcionális felépítése

A digitális szolgáltatás-elérési pontokhoz kapcsolódó ISDN végberendezések a csatlakozás *D-csatornáján* keresztül, a DSS1 előfizetői jelzésrendszer protokolljának megfelelő szabályok szerint tartanak fent adat-kapcsolatot az előfizetői fokozattal, illetve a központ vezérlőrendszerével.

Az *analóg* szolgáltatás-elérési pontokhoz csatlakozó végberendezések az előfizetői vonalon továbbított *egyenáramú* és *hangfrekvenciás* jelzésekkel kommunikálnak az előfizetői fokozattal, illetve azon keresztül a központ vezérlőrendszerével. Az ehhez szükséges környezeti feltételek megteremtése

céljából, az analóg vonaláramköröknek az ún. BORSCHT funkciókat kell megvalósítaniuk (az alábbi felsorolásban szereplő, zárójelbe tett nagybetűk az érintett funkciók angol elnevezésének kezdőbetűit jelentik):

- gondoskodnia kell az egyenáramú vonaltáplálásról (B),
- a vonal felől érkező nagyfeszültségű lökések elleni védelemről (O),
- a csengetőjel és a beszédjelek útjának szétválasztásáról (R),
- az előfizetői hurok állapotának figyeléséről (S),
- az analóg-digitális átalakítás mindkét átviteli irányban való megvalósításáról (C),
- a 2/4-huzalos átalakításról (H), valamint
- a tesztelés (T) lehetőségéről.

Mindenképpen megjegyzendő azonban, hogy a 3.2.2. ábra sodrott érpárú, *vezetékes* analóg csatlakozást említ, de a gyakorlatban – városi környezetben a gyors telepíthetőség megvalósítása, ritkán lakott területen (pl. tanyák esetében) gazdaságossági szempontok miatt – léteznek *vezeték nélküli* előfizetői csatlakozások is, mint pl. az RLL (Radio in the Local Loop) vagy WiLL (Wireless Local Loop) rendszereken át megvalósuló. A vezeték nélküli előfizetői csatlakozások részleteinek tárgyalásával, és így az ilyen típusú módszerekkel azonban itt nem kívánunk foglalkozni.

A 3.2.2. ábra szerinti fokozat valamennyi előfizetője számára (ISDN és analóg csatlakozásúak számára egyaránt) közös, digitális *jelzőhang-generátor* szolgál a különböző forgalmi esetekben szükséges tájékoztató hangjelzések kiadására.

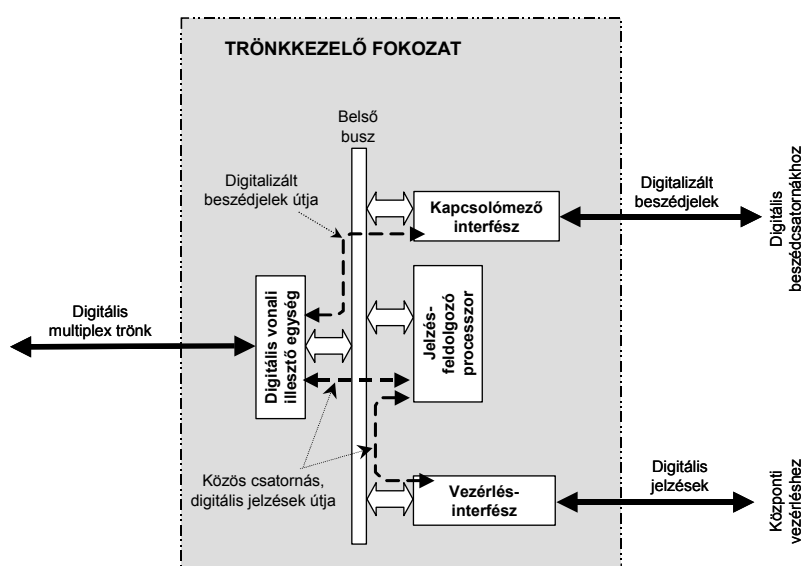
Az analóg vonalakkal csatlakozó hívott előfizetők felcsengetése céljából külön csengetőjel-generátort kell alkalmazni az előfizetői fokozatban. A csengetőjel-generátor jelfrekvenciája és váltófeszültsége olyan értékű, hogy egyaránt alkalmas mind a korszerű, elektroakusztikus *csengetésjelzők*, mind pedig a régi, elektromechanikus *csengők* megszólaltatására.

Az előfizetői fokozat a vonaláramkörökön kívül illesztő egységeket tartalmaz még a digitális kapcsolómező és a központ vezérlőrendszere irányában, valamint a fokozat működési ellenőrzését (tesztelését) támogató központi vizsgálórendszer felé is.

b) Digitális kapcsolómező

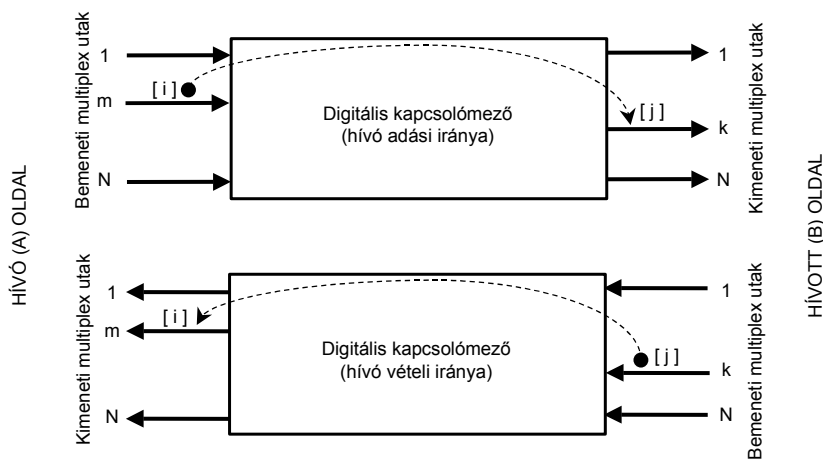
A digitális kapcsolómező funkcionális vázlatát a 3.2.3. ábra mutatja be. A kapcsolómező feladata: bármelyik *bemeneti* csatorna (időrés) *tartalmát* kívánság szerinti *kimeneti* csatornába (időrésbe) tegye át. Ebből következik, hogy a digitális kapcsolómezőnek *időbeli* és *térbeli* helyzeteket kell megváltoztatnia. Ezért a digitális kapcsolómező kétféle típusú *elemi kapcsoló* alkalmazásával építhető:

- időbeli helyzetet megváltoztató kapcsolókkal (T-kapcsolókkal), és
- térbeli helyzetet megváltoztató kapcsolókkal (S-kapcsolókkal).



3.2.3. ábra. Digitális kapcsolómező feladatainak vázlatos ábrázolása

Az elemi kapcsolók közül elsőnek a csatornák időbeli helyzetét megváltoztató *T-kapcsolót* (a név az angol *Time-switch* elnevezésből származik) tárgyaljuk. A T-kapcsoló egy bemeneti és egy kimeneti multiplex út időrései között hoz létre kapcsolatot, amely azt jelenti, hogy bármelyik *bemeneti* (pl. az *i*-edik) időrés tartalmát tetszőleges *kimeneti* időrésben (pl. az *x*-edikben) képes továbbítani. A példa szerinti *i*-edik bemeneti időrésben érkező információnak tehát addig kell *várnia* a T-kapcsolóban, amíg az *x*-edik kimeneti időrés időhelyzete be nem következik. Figyelembe véve, hogy a bemeneti multiplex úton az információ periodikusan (keretekbe szervezett formában) érkezik, ezért a T-kapcsolón belüli várakozás nem lehet hosszabb, mint a keretidő időtartama. Ellenkező esetben ugyanis a várakozó információt felülírja a következő keretben érkező új információ, és ez információvesztést jelent, amely megengedhetetlen. A *kimeneten vezérelt* T-kapcsoló főbb funkcionális elemeit a 3.2.4/a ábrán láthatjuk.



A digitális kapcsolómező adásiirányú feladata:

A hívó beszédjel-mintáit az m -edik bemeneti multiplex út i -edik időréséből át kell kapcsolnia a k -adik kimeneti multiplex út j -edik időrésébe: $m[i] \rightarrow k[j]$

A digitális kapcsolómező vételirányú feladata:

A hívott beszédjel-mintáit a k -adik bemeneti multiplex út j -edik időréséből át kell kapcsolnia az m -edik kimeneti multiplex út i -edik időrésébe: $k[j] \rightarrow m[i]$

A T-kapcsoló elvi működése két memóriaegység (IM információ-tároló, KM kapcsolótároló) és egy időrés-számláló (SZ) együttműködésén alapul.

3.2.4. ábra. Digitális kapcsolóelemek funkcionális felépítése: a) Kimeneten-vezérelt T-kapcsoló; b) Kimeneten-vezérelt S-kapcsoló

A továbbítandó (kapcsolandó) információ a *bemeneti* multiplex út időréseiből az IM tárolónak az SZ időrés-számláló által kijelölt rekeszeibe kerülnek. Az IM tárolóból – a kívánt összeköttetés által meghatározott időbeli késleltetés szerinti várakozás után – az SZ által címzett KM kapcsolómemória olvastatja ki a rekeszeket, és továbbítja tartalmukat a kimeneti multiplex út időréseiben. A KM rekeszébe ugyanis azon az IM-rekesz címe van beírva, amely tartalmát a KM-rekesz kiolvasásához rendelt *kimeneti* időrésben kell továbbítani, következésképpen a KM rekeszeibe egy-egy összeköttetés felépítésekor kell a szükséges adatokat beírni.

Az elemi kapcsolók másik típusa a csatornák (vagyis az időrések) *térbeli helyzetét* megváltoztató kapcsoló (S-kapcsoló – az angol *Space-switch* elnevezés alapján). Ennek feladata, hogy a kapcsolóhoz csatlakozó bármelyik bemeneti multiplex út időrésének tartalmát – kívánságra – tetszőleges kombináció szerint áttegye valamelyik kimeneti multiplex út ugyanazon helyzetű időrésébe. Ez gyakorlatilag azt jelenti, hogy a bemeneti x -edik időrésben érkező kódszó a kimeneten is az x -edik időrésben fog megjelenni, csak a bemenethez képest más térbeli helyzetben. Ezt az átkapcsolási képességet próbálja érzékeltetni a 3.2.5/b.

ábra vázlatos rajza, amelyen a *kimeneten vezérelt*, szimmetrikus szerkezetű S-kapcsoló felépítését és működését tanulmányozhatjuk.

A szimmetrikus szerkezetű S-kapcsoló bemenetéhez és kimenetéhez egyaránt N darab digitális multiplex út csatlakozik, amelyek lehetnek akár soros, akár párhuzamos működésűek. Az ábra szerinti példában – a működési elv egyszerűbb magyarázata érdekében – mind a bemenetek, mind pedig a kimenetek soros átviteli utak.

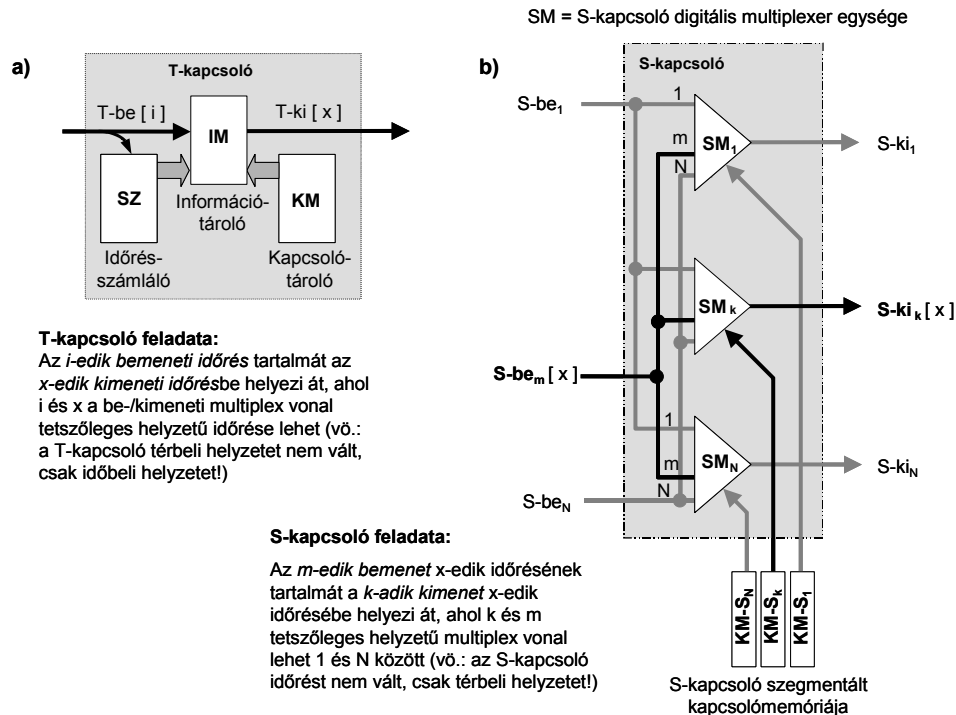
A kapcsolást logikai multiplexer áramkörök végzik (az ábrán SM jelzésűek). A logikai multiplexerek jellemzője, hogy több jelbemenetük, egy címző-bemenetük, és egyetlen jelkimenetük van. A jelkimenetet a címző-bemeneten megjelenő információ kapcsoltatja össze a címmel jellemzett jelbemenettel addig az időtartamig, amíg a címző-bemeneten jelen van a címinformáció. Mivel a rajz feltételezése szerint a kapcsoló kimeneteinek száma megegyezik a bemenetekével, a logikai multiplexerek száma N kell, hogy legyen. A multiplexerek azonos sorszámú bemenetei párhuzamosítva (multiplikálva) vannak.

A multiplexereket az S-kapcsoló szegmensekre osztott KM kapcsolótárolója a szegmensek rekeszeibe írt, majd onnan az SZ által kiolvasott bemeneti sorszámmal címezi meg, amelyeket egy-egy összeköttetés felépítésekor kell oda beírni. A KM szegmenseinek mindegyike egy-egy SM címző-bemenetéhez, míg a szegmens rekeszei (a T-kapcsoló KM-tárolójához hasonló módon) a kimeneti multiplex út egy-egy időréséhez vannak hozzárendelve.

A T- és S-kapcsolókból felépíthető kapcsolómezőket terjedelmi okokból itt nincs módunkban megtárgyalni. Közülük azonban az ún. T-S-T struktúrájú az elterjedtebb, ezért röviden ezt fogjuk a következőkben bemutatni.

A 3.2.3. ábrán már foglalkoztunk egy példa keretében a digitális kapcsolómező feladatával, és ezt a példát folytatjuk a T-S-T struktúrájú kapcsolómező bemutatásakor is. A T-S-T struktúrára az jellemző, hogy bemeneteinél és kimeneteinél T-kapcsolók (bemeneti, illetve kimeneti T-kapcsolók) helyezkednek el, míg a két T-kapcsoló fokozat között találjuk az S-kapcsolót. A *bemeneti T-kapcsolók* egy közbenső, belső időréshez kapcsolják a bemeneti multiplex út időrészeit, az *S-kapcsoló* a belső időrészekkel elvégzi az összeköttetéshez szükséges térbeli változtatást, majd a *kimeneti T-kapcsolók* a belső időrészeket kapcsolják a

kiválasztott kimeneti időréshez. A 3.2.5. ábrán bemutatott struktúra szerint a T-kapcsolók a négyhuzalos multiplex utak szerint vannak csoportosítva úgy, hogy az azonos sorszámú átviteli út bemeneti és kimeneti T-kapcsolója, valamint az említett



sorszámmal megegyező S-kapcsoló szegmens kapcsolótárolója képezik az azonos csoportba tartozó elemeket. Az ábrán külön van jelezve a példa szerint feltételezett összeköttetés útvonala a bemeneti időréstől egészen a kimeneti időrésig.

3.2.5. ábra. T-S-T struktúrájú kapcsolómező funkcionális elemei

3.2.4. A vezérlési funkció

A digitális kapcsolóközpontban valamennyi összeköttetés felépítését, tartását és bontását tárolt programú vezérlés végzi. Ez a feladat természetesen együtt jár az összeköttetések felépítéséhez szükséges jelzések kezelésének feladatával is.

Az alábbiakban azt fogjuk áttekinteni, hogyan alakulnak a digitális kapcsolóközpontok vezérlőrendszereinek *struktúrái*.

Az osztott hierarchikus vezérlőrendszerekben a feladatokat csoportosítják, és az így struktúrált feladatokat különböző, egymással kapcsolatot tartó processzorok között osztják szét.

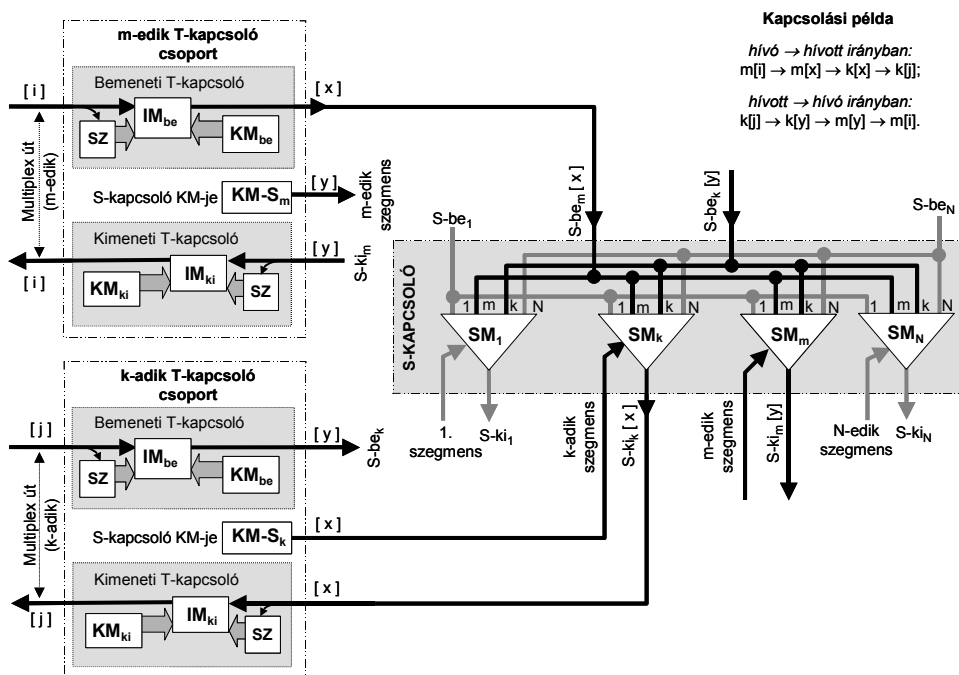
Többprocesszoros vezérlőrendszert elvileg különféle felépítési elv szerint lehet megvalósítani. A gyakorlat próbáját is kiállt struktúrák mindegyikének vannak előnyös, és kevésbé előnyös (hátrányos) műszaki illetve gazdaságossági tulajdonságai. *Mindig a gyártó* dönti el, hogy az általa fejlesztett rendszerben milyen struktúrát célszerű alkalmaznia, vagyis melyik struktúra előnyös tulajdonságait tudja a saját tervezési céljai szempontjából hasznosítani, miközben figyelmen kívül hagyhatja annak hátrányos jellemzőit. Ez egyúttal azt is jelenti, hogy nem lehet *egyértelműen* kijelenteni egy vezérlési struktúráról, hogy az jobb-, vagy rosszabb, mint a többi. Itt sem teszünk erre kísérletet, hanem minden értékelés mellőzésével bemutatjuk a magyarországi hálózatban megjelent digitális telefonközpont-rendszerekben alkalmazott vezérlési struktúrákat, és röviden foglalkozunk ezeknek a struktúráknak a működési elvével. Ennek megfelelően az alábbi három vezérlési struktúrával fogunk röviden foglalkozni:

- perifériabuszt alkalmazó vezérléssel,
- kapcsolt jelzéscsatornákat használó vezérléssel, és
- elosztott struktúrájú vezérléssel.

a) Perifériabuszt alkalmazó vezérlés

A perifériabuszt alkalmazó vezérlés funkcionális vázlatát a 3.2.6. ábrán tanulmányozhatjuk. A rendszer processzorai (vezérlőelemei) *hierarchikus felépítésben* osztják meg egymás között a feladatokat: van egy *központi processzor*, amelynek a működését kiegészítő (periféria-) processzorok támogatják. A központi processzor “előzetesen feldolgozva” kapja meg ezektől a kiegészítő processzoroktól az információt, és ellenkező irányba sem kell minden részletig bezáróan kidolgoznia a teendőket, mert a perifériaprocesszorok a végrehajtás részleteinek kidolgozásában is segítenek.

A struktúra egyik fő jellemzője a *perifériabusz* alkalmazása. A központi processzor és a perifériaprocesszorok között minden utasítás, adat és egyéb információ, amely a teljes vezérlőrendszer működéséhez szükséges, csak ezen a buszrendszeren keresztül továbbítható. Bekapcsoláskor vagy új szoftverváltozatok üzembe helyezésekor először a központi processzort töltik fel a szükséges programokkal és adatbázissal, a többi processzor feltöltését már a központi processzor vezérli.



3.2.6. ábra. Perifériabuszt alkalmazó vezérlés

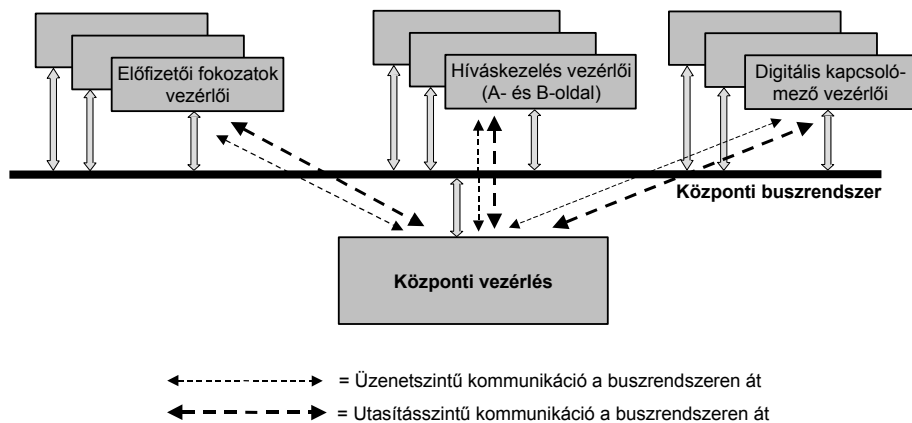
b) Kapcsolt jelzéscsatornájú vezérlés

A kapcsolt jelzéscsatornákon keresztül kommunikáló vezérlés vázlatos rajzát a 3.2.7. ábra mutatja be. Ez a struktúra is *hierarchikus felépítésű* processzorrendszert valósít meg, de a processzorok közötti információcsere kapcsolt jelzéscsatornákon keresztül, a kapcsolómező közreműködésével valósul meg. Bekapcsoláskor vagy új szoftverváltozatok üzembe helyezésekor a programok és az adatbázis betöltésének módja lényegében azonos a perifériabuszt alkalmazó, hierarchikus struktúrájú vezérlésnél említett megoldással.

c) Elosztott vezérlés

Az *elosztott struktúrájú* vezérlés vázlatos felépítését a 3.2.8. ábrán mutatjuk be. Ebben a struktúrában nincs olyan központi vezérlő, amely a hívások minden részletét felügyelné (vagyis nincs hierarchia). Minden kapcsolóegységnek saját, önálló feladatokat megoldani képes processzora van, amely a nála jelentkező használói igényeket a szomszédos processzorokkal együttműködve kezeli. A híváskezelést tehát ezek a processzorok együttműködéses formában hajtják végre. Ebben a vezérlési struktúrában is létezik azonban egy minden processzor által elérhető "közös" processzor, de ez csak *koordináló* feladatokat lát el. Ez azt jelenti,

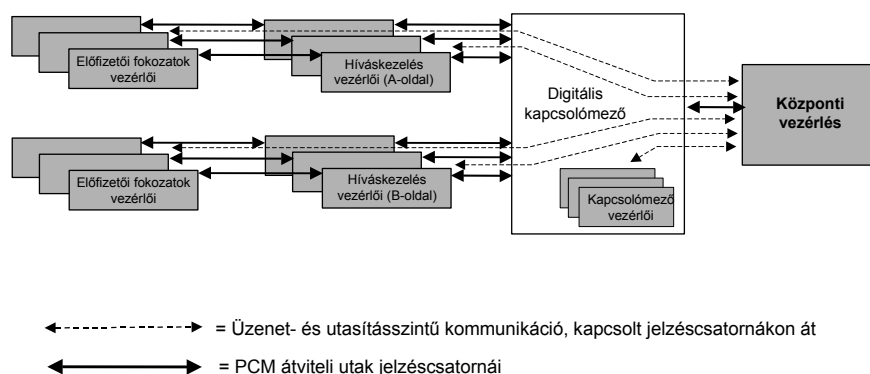
hogy a teljes rendszer adminisztrációs adatait, a rendszer valamennyi kapcsolóelemének pillanatnyi állapotát, és más ehhez hasonló jellegű információt kizárólag ez a processzor képes áttekinteni. Ez a koordinációs processzor “szervezi meg” azt, hogy a többi processzor közül melyek éppen azok, amelyek egy-egy hívásigény kezelése céljából egymással kapcsolatba kerülnek, a híváskezelés vagy egyéb használói igény lebonyolításánál azonban nincs semmiféle más feladata. Jellemző részlete az elosztott vezérlési struktúrának, hogy *nincs benne perifériabusz*.



3.2.7. ábra. Kapcsolt jelzéscsatornákkal működő centralizált vezérlés

Jellemzője még ennek a vezérlési struktúrának az is, hogy bekapcsoláskor-, vagy egy új szoftverváltozat üzembeállításakor bonyolultabb a programok illetve az adatok betöltése, mint a centralizált, hierarchikus struktúra esetében.

3.2.8. ábra. Elosztott felépítésű vezérlés



3.2.5. A megbízható működésről

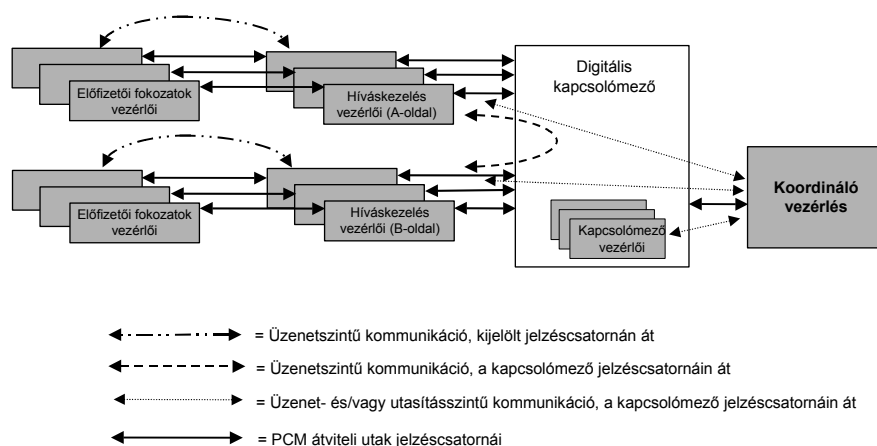
A digitális kapcsolóközpontok elektronikus alkatrészei kisebb-nagyobb valószínűséggel elromolhatnak, ennek következtében az általuk érintett környezet (így akár az egész központ) váratlanul működésképtelenné válhat.

A közcélú telefonközpontok 24-órás, folyamatos üzemben működnek, vagyis megengedhetetlen, hogy a kapcsolórendszer valamilyen hiba bekövetkezése miatt teljesen leálljon, és ezért az előfizetők számára lehetetlenné válják a telefonálás. E veszély ellen a legegyszerűbb megoldás a veszélyeztetett áramkörök valamilyen módon való azonnali (minimális, pl. ms-os nagyságrendű időn belüli) helyettesítése. Az ilyen célok megvalósítását a melegtartalékolásos módszer támogatja.

Melegtartalék alkalmazása azt jelenti, hogy a helyettesítésre szánt (tartalék) funkcionális egység *üzemszerű* állapotban van, és hiba bekövetkezése esetén, annak felderítése után automatikusan kiadott *távvezérlő utasításra* lép az elromlott egység helyébe.

A melegtartalékkal működtetett egységek üzembiztos (leállás nélküli) működtetése azonban csak abban az esetben valósul meg, ha soha nem fordul elő bennük *egynél több hiba*. Ez a feltétel azt jelenti, hogy egy hiba bekövetkezését minél kisebb reakcióidővel kell felfedni, és az ezt követő javítási folyamatnak is minél rövidebb idő alatt kell lebonyolódnia. Ezért – a nagyobb tűrőképesség és gazdaságos megoldások kialakítása érdekében – a telefonközpontokban *nem a teljes rendszerre* kiterjedőn, egyetlen “nagy” tartalék működtetése révén valósítják meg a melegtartalékolást, hanem a rendszer *részegységei* vannak külön-külön tartalékolva. A folyamatos működőképesség szempontjából tehát a tartalékolt részegységekben külön-külön csak egyetlen hiba bekövetkezése megengedett, a teljes rendszerben azonban már egyidejűleg több is létezhet anélkül, hogy az a kapcsolóközpont teljes leállítását eredményezné. A kevésbé bonyolult részegységek működésének figyelése egyszerűbb, a hibák bekövetkezésének észlelése gyorsabb, a hibák kijavítása pedig lényegesen rövidebb időt vesz igénybe, mintha a nagy kiterjedésű és bonyolultabb működésű teljes rendszert tartalékolnánk.

Digitális kapcsolóközpontok tervezésekor célszerűségi szempontok alapján határozzák meg a tartalékolandó részegységek kiterjedését, és az ott működtetett



melegtartalékolási módszert. A gyakorlatban kétféle melegtartalékolási módszert alkalmaznak:

- szinkron üzemmódú tartalékolást és
- terhelés-megosztásos tartalékolást

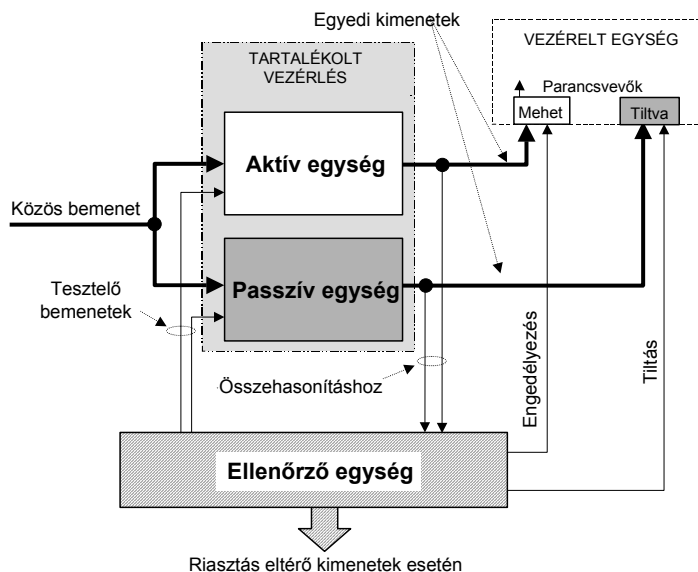
Szinkron üzemmódú tartalékolás esetén a részegységet két, egymással párhuzamosan működő funkcionális egység alkotja, amelyek közül az egyik *aktív*, a másik *passzív* (tartalék) állapotúnak van kijelölve. A két funkcionális egység a bemeneti eseményeket szinkronban dolgozza fel, de a kimenetekről származó adatokat, utasításokat stb. a vezérelt periféria mindig csak az aktív elemből fogadhatja el. A 3.2.9. ábrán egy szinkron üzemmódúan tartalékolt részegységnek és környezetének vázlatos rajzát láthatjuk.

3.2.9. ábra. Szinkron működésű tartalékolás elve

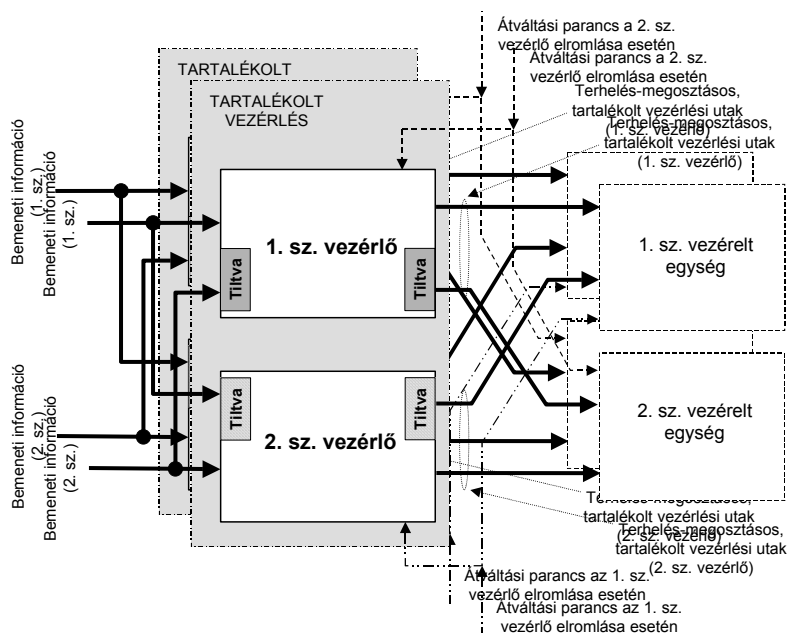
A tartalékolt részegység környezetéhez – a vezérelt perifériákon kívül - egy ellenőrző egység is tartozik, amelynek az a feladata, hogy egyrészt észlelje, ha a két funkcionális egység kimenetén nem azonos az információ (ez mindenképpen arra utal, hogy egyikük rosszul működik), másrészt automatikusan meg kell állapítania, hogy az adott pillanatban melyik funkcionális egység romlott el. Ez utóbbi körülményt természetesen nem képes *abszolút biztonsággal* megállapítani, de arra már képesnek kell lennie, hogy *nagy valószínűséggel* helyesen döntsön (a helytelen döntés ugyanis azzal járhat együtt, hogy a tartalékolt részegység, és ennek következtében a teljes rendszer is csődöt mond). Az említetteken túlmenően az ellenőrző rendszernek informálnia kell a vezérelt perifériákat arról, hogy melyik az aktív és melyik a passzív funkcionális egységhez tartozó kimenet, hiszen a periféria csak az aktívtól fogadhat el információt. Hiba észlelése esetén az ellenőrző egység leválasztja az elromlottnak feltételezett részt a vezérelt perifériáról, kizárólagos

jelleggel aktívvá teszi a jónak feltételezett részt, és – a javítási folyamat céljából – riasztja a karbantartó személyzetet.

A hibás funkcionális egység leválasztása és a hibátlan aktívvá nyilvánítása után a vezérlőrendszer a részegység megszakítás nélkül, változatlan kapacitással működhet tovább annak ellenére, hogy az egyik fele elromlott. A hiba helyének, vagyis az elromlott funkcionális egységen belüli hibás elem pozíciójának meghatározása alatt a részegység természetesen tartalékolás nélkül folytatja a működését. Az eredeti állapot csak akkor fog ismét helyreállni, miután a hibásnak nyilvánított elemet kicserélték, és ellenőrző vizsgálatok után a kijavított funkcionális egységet újra üzembe nem állították. A szinkron üzemmódú tartalékolás lényeges jellemzője tehát, hogy a rendszer hiba esetén is ugyanolyan sebességgel végzi a feladatát, mint hibamentes esetben.



a. kapcsolómező



b. vezérlés

3.2.10. ábra. Terhelés-megosztásos tartalékolás elve

Terhelés-megosztásos tartalékolás esetében legalább két, funkcionális egység (de esetenként lehet több is) egymás tartalékjaként működik párhuzamosan. Hibamentes esetben a funkcionális egységeknek különálló, de persze azonos jellegű funkciója van, és mindegyik végzi a maga feladatát. Ha valamelyikük elromlik, akkor a másik funkcionális egység veszi át az elromlott funkcióját, de ettől kezdve nagyobb terhelés mellett kell korábbi feladatait teljesítenie. Ez a körülmény azt eredményezheti, hogy a működésben maradt funkcionális egység csak lassabban lesz képes hatáskörét ellátni.

A 3.2.10. ábrán – nagyon leegyszerűsített formában és csak az elv bemutatására hagyatkozva – két, egymással terhelés-megosztásos tartalékolásban működő vezérlőegység vázlata látható (vö.: a perifériák nincsenek tartalékolva!). Az ábrázolás egyszerűsítése és a jobb áttekinthetőség érdekében a rajzon csak utalások találhatók arra a környezetre vonatkozóan, amely a vezérlőrendszert abban segíti, hogy zökkenőmentesen tudjon működni. Az egyidejűleg egyetlen hiba eltűrésének feltétele természetesen itt is fennáll, ezért a hiba bekövetkezésének pillanata és az eredeti állapot helyreállítása közötti időtartamnak ennél a megoldásnál is a lehető legrövidebbnek kell lennie.

3.3. Forgalmi modellek és forgalmi méretezés

Szerző: dr. Molnár Sándor

Lektor: dr. Frajka Béla

3.3.1. Bevezetés

A hálózati forgalom természetéről, valamint a forgalomelméleti alapokról az 1.7. fejezetben adtunk áttekintést. Ezeket az alapokat felhasználva ez a fejezet áttekintést ad a legfontosabb forgalmi modellekről amelyek a hálózati forgalom modellezésére alkalmasak lehetnek. Áttekintjük a forgalmi modell típusokat a legegyszerűbbektől a legbonyolultabbakig. A gyakorlatban a modelltípus kiválasztásánál mindig kompromisszumot kötünk, hogy a modell kellően komplex legyen ahhoz, hogy lehetőleg a legjobban leírja a valóságos forgalom természetét, de ugyanakkor a lehető legegyszerűbb legyen, hogy matematikailag is jól tudjuk kezelni. A kiválasztott forgalmi modellt általában valamilyen sorbanállási feladatban alkalmazzuk, amelynek segítségével teljesítményjellemzőket határozhatunk meg.

A fejezetben áttekintést adunk a telefonhálózatok és az adathálózatok forgalomméretezési eljárásairól. Az adathálózatoknál külön hangsúllyal tárgyaljuk az Internet forgalmi méretezését.

3.3.2. Forgalmi modellek

A forgalom általában egyedi vagy több diszkrét egységekből áll (csomagok, cellák, stb.). Matematikailag a forgalmat a *pontfolyamatok* elméletével írhatjuk le. A pontfolyamatoknak két lehetséges leírásmódja létezik: a *számlálófolyamatok* segítségével vagy az *érkezések közötti idők sorozatának* leírásával [3.3.1]. A számláló folyamat $\{N(t)\}_{t=0}^{\infty}$ egy folytonos idejű, nem negatív, egész értékű sztochasztikus folyamat, ahol $N(t) = \max\{n : T_n \leq t\}$ az érkezések száma a $(0, t]$ intervallumban. Az érkezések közötti idők sorozata egy valós értékű véletlen számsorozat $\{A_n\}_{n=1}^{\infty}$, ahol $A_n = T_n - T_{n-1}$ az n -ik és az azt megelőző érkezés közötti idő hossza. A forgalmat batch folyamatnak nevezzük kötegelt érkezések esetén. A batch

folyamatok leírására definiálunk batch érkezési folyamatot $\{B_n\}_{n=1}^{\infty}$ ahol B_n az érkezések száma az n -dik batchben. Hasonlóan hasznos leírófolyamat az ún. *munkahátralék folyamat* $\{W_n\}_{n=1}^{\infty}$. Ez leírja az n -dik érkezésben mennyi W_n munka érkezett a rendszerbe.

A következőkben forgalmi modelleket ismertetünk, melyek forgalom generálására használhatóak és jellemezhetőek a következő sorozatok valamelyikével: $\{N(t)\}_{t=0}^{\infty}$, $\{A_n\}_{n=1}^{\infty}$, $\{B_n\}_{n=1}^{\infty}$, $\{W_n\}_{n=1}^{\infty}$.

A felújítási folyamatban $\{A_n\}_{n=1}^{\infty}$ független és azonos eloszlású valószínűségi változók sorozata [3.3.1], [3.3.2]. Ez a modell egyszerű, de sok esetben nem valósághű, mert nem tudja modellezni a valódi forgalom erős korrelációs struktúráját.

A Poisson folyamat [3.3.1], [3.3.2] olyan felújítási folyamat, melynél az érkezések közötti idők $\{A_n\}_{n=1}^{\infty}$ exponenciális eloszlásúak λ paraméterrel. A Poisson folyamatot definiálhatjuk a számláló folyamat segítségével is, ahol $\{N(t)\}_{t=0}^{\infty}$ folyamatnak független és stacionárius növekményei vannak Poisson határeloszlással: $P\{N(t) = n\} = \exp(-\lambda t)(\lambda t)^n / n!$ A Poisson folyamatot egyszerűsége és néhány elegáns matematikai tulajdonsága miatt gyakran használják a forgalomelméletben. A telefonhívások érkezését a telefonhálózatokban általában Poisson folyamattal modellezzik.

A Bernoulli folyamatok [3.3.1], [3.3.2] a Poisson folyamatok diszkrét idejű analóg folyamatai. Ebben a modellben az érkezés valószínűsége minden időrésben p és független a többi érkezéstől. A k darab időrésben vizsgált érkezések száma binomiális eloszlású: $P\{N(t) = n\} = \binom{k}{n} p^n (1-p)^{k-n}$ és az érkezések közötti idő geometriai eloszlású: $P\{A_n = j\} = p(1-p)^j$

A fázis típusú felújítási folyamatok [3.3.1], [3.3.2] olyan speciális felújítási folyamatok, melyeknél az érkezések közötti idő eloszlása fázis típusú eloszlás. Ez egy fontos modellosztály, mert analitikusan jól kezelhető és ugyanakkor minden eloszlás tetszőleges pontossággal közelíthető fázis típusú eloszlásokkal.

A Markov-i forgalmi modellekben [3.3.1], [3.3.2] függőségi viszonyt modellezhetünk az A_n véletlen sorozatban. A modell konstrukciója a következő:

tekintsünk egy diszkrét állapotterű Markov folyamatot $M = \{M(t)\}_{t=0}^{\infty}$. M a következőképpen viselkedik: az i -dik állapotban marad λ_i paraméterű exponenciális eloszlású tartásideig. Vegyük észre, hogy az eloszlás paramétere egyedül az állapottól függ. Ezután j -dik állapotba ugrik p_{ij} valószínűséggel. A Markov folyamat minden ugrásához rendelünk egy érkezést, így az érkezések közötti idő exponenciális. Ez a legegyszerűbb *Markov-i forgalmi modell*.

A **Markov-i felújítási folyamat** [3.3.1], [3.3.2] sokkal általánosabb, mint az egyszerű Markov folyamat. Ennek ellenére ez a folyamat még mindig analitikusan kezelhető. A Markov felújítási folyamatot $R = \{(M_n, \tau_n)\}_{n=0}^{\infty}$ az $\{M_n\}$ Markov lánc és ennek az ugrálásai közötti idő $\{\tau_n\}$ segítségével definiálhatjuk. A következő szabályt vezetjük be: az (M_{n+1}, τ_{n+1}) pár eloszlása csak a jelenlegi állapottól M_n függ és nem függ a korábbi állapotoktól vagy a korábbi ugrálások közötti időktől. Ebben a modellben szintén az ugrálásokhoz rendelünk érkezéseket.

A **Markov-i érkezési folyamatok** (Markov arrival processes) (MAP) [3.3.1], [3.3.2], [3.3.3] a Markov felújítási folyamatok egy nagy osztályát képezik. A MAP folyamatokban az érkezések közötti idő fázis típusú és az érkezések a Markov folyamat abszorpciós állapotaiba ugrálásakor történnek. A folyamatot olyan eloszlásból indítjuk újra, ami az adott abszorpciós állapottól függ. A MAP folyamatok analitikusan kezelhetőek és egy nagyon jó modellezési tulajdonságokkal bírnak.

A **Markov modulált folyamatoknál** egy Markov lánc pillanatnyi állapota határozza meg az érkezések generálásának szabályát [3.3.1], [3.3.2]. Tekintsünk egy folytonos idejű Markov folyamatot $M = \{M(t)\}_{t=0}^{\infty}$ ahol az állapotter: $1, 2, \dots, m$. Amíg az M folyamat k állapotban van, az érkezések generálásának szabálya kizárólag a k állapottól függ. Amikor M egy másik állapotba kerül pl. j , akkor amíg ebben az állapotban van egy új, csak ezen állapot által meghatározott generálási szabály lesz érvényes. Másképp fogalmazva az érkezési szabályokat M modulálja. A moduláló folyamat természetesen a Markov folyamatnál jóval bonyolultabb is lehetne, de akkor a modell analitikusan kevésbé volna jól kezelhető.

A **Markov modulált Poisson folyamat** (Markov Modulated Poisson Process) (MMPP) [3.3.1], [3.3.2], [3.3.3] a leggyakrabban használt Markov modulált forgalmi modell. Ebben a modellben, amíg a moduláló Markov folyamat k állapotban van, az

érkezések Poisson folyamat szerint történnek, amelyeknek intenzitása λ_k . A legegyszerűbb Markov modell a kétállapotú MMPP, ahol az egyik állapotot "ON" bekapcsolt állapotnak értelmezzük valamilyen Poisson intenzitással, a másik állapotot "OFF" kikapcsolt állapotnak nulla intenzitással. Ezt a modellt megszakított Poisson folyamatnak is hívják. Az ON-OFF modellek gyakran használatosak beszédforgalom modellezésére, ahol az ON állapot a beszéd, az OFF állapot a szünet modellezésére szolgál.

A Markov-i állapotátmenet modulált folyamatokban [3.3.1], [3.3.2] a Markov folyamat $M = \{M(t)\}_{t=0}^{\infty}$ állapotátmenet ugrásai modulálják az érkezők generálásának szabályát. Az állapotátmeneteket egy állapotpár segítségével írhatjuk le: az ugrás előtti és az ugrás utáni állapottal. Az érkezők száma B_n az n időrésben teljesen szabályozva van a moduláló lánc ugrásaival: $P\{B_n = k | M_n = i, M_{n+1} = j\} = t_{ij}(k)$ és teljesen független a múlt bármely más állapotinformációjától.

Az általánosan modulált determinisztikus folyamatokban (Generally Modulated Deterministic Process) (GMDP) [3.3.3] a forrás bármely állapotban lehet a lehetséges N állapotból. Amikor j állapotban van, a forgalom állandó λ_j intenzitással generálódik. A j állapotban eltöltött időt tetszőleges eloszlás meghatározhatja, de a gyakorlatban legtöbbször geometriai eloszlást használnak, hogy a modell analitikusan könnyen kezelhető legyen. A kétállapotú GMDP, ahol az egyik állapotban nulla a generálási intenzitás, a diszkrét megfelelője a már tárgyalt ON-OFF modellnek.

A folyadék forgalmi modellezési technikában a forgalmat egy folyamatosan áramló folyadéknak modellezzük és eltekintünk a valódi forgalom diszkrét egységekből álló természetétől [3.3.1], [3.3.2], [3.3.3]. Ez a modellezés akkor elfogadható, ha a forgalomnak a vizsgált időskálán rengeteg diszkrét egységét, pl. csomagokat figyelhetünk meg. A modell előnye az, hogy sokkal egyszerűbb, mint azok a modellek, melyek a forgalom diszkrétizált egységei közötti struktúrát próbálják leírni. A legegyszerűbb folyadékmodellek két állapotot feltételeznek: egy ON állapotot, amikor a forgalom egy állandó λ rátával áramlik, és egy OFF állapotot, amikor nincs forgalom továbbítás. Az egyszerű analitikus kezelhetőség érdekében

sokszor az ON és OFF állapotokat független exponenciális eloszlású valószínűségi változókkal modellezzük. Ebben az esetben a modell egy alternáló felújítási folyamat.

Az **autoregresszív forgalmi modellben** a múltban bekövetkezett forgalmi változások explicit függvénye határozza meg a jelenben bekövetkező forgalmi változásokat [3.3.1], [3.3.2], [3.3.3]. Ezen modellcsalád tipikus példái a *lineáris autoregresszív (AR) folyamatok*, a *mozgó átlag (MA) folyamatok*, az *autoregresszív mozgó átlag (ARMA) folyamatok* és az *autoregresszív integrált mozgó átlag (ARIMA) folyamatok*. Ezek a modellek nagyon hasznosnak bizonyultak VBR videó forgalom modellezésére.

A **TES modellek** (*Transform-Expand-Sample*) [3.3.1], [3.3.2] olyan modellcsalád, mely a következő követelményeknek felel meg: a határeloszlást és az autokorrelációs függvényt illeszti az empirikus adatokra. A TES modellek jól használhatóak pl. MPEG videó modellezésére.

A **fraktális Gauss zaj** (Fractional Gaussian Noise, FGN) [3.3.1] egy másodrendben önhasnoló folyamat H hasonlósági paraméterrel, ahol $\frac{1}{2} < H < 1$. Ez egy stacioner Gauss folyamat $X = \{X_k\}_{k=1}^{\infty}$, a következő autokorrelációs függvénnyel:

$$\rho_X(k) = \frac{1}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}), k \geq 1.$$
 Az FGN egyúttal hosszú idejű összefüggő folyamat (LRD) H paraméterrel: $\rho_X(k) \approx H(2H-1)|k|^{2H-2}, k \rightarrow \infty$. Az FGN egy jó forgalmi modell aggregált LRD forgalom modellezésére.

A **fraktális ARIMA folyamatok** (fractional ARIMA, FARIMA) [3.3.1] a klasszikus $ARIMA(p,q,d)$ modellre épülnek, de a differencia operátor d parametere tört értékeket is felvehet. A FARIMA modellek sokkal rugalmasabbak, mint az FGN modellek, mert nem csak az LRD struktúrát, de egyúttal a rövid idejű összefüggőségi struktúra (short-range dependent, SRD) is jól modellezhető vele.

Az **M/Pareto modellben** λ intenzitású Poisson folyamattal érkeznek Pareto eloszlású börsztök [3.3.4]. A börszt ideje alatt az állandó intenzitású r rátával történnek érkezések. A börszt hosszát a Pareto eloszlás határozza meg, melynek

paraméterei: $1 < \gamma < 2, \delta > 0$:
$$P\{X > x\} = \begin{cases} \left(\frac{x}{\delta}\right)^{-\gamma}, & x \geq \delta \\ 1, & \text{egyébként} \end{cases}$$
 A modell LRD forgalmat

generál $H = (3 - \gamma)/2$ paraméterrel.

Az **alkalmazásorientált forgalmi modellek** azok, melyekkel közvetlenül valamilyen gyakorlati alkalmazás forgalmát modellezhetjük. Az eddig felsorolt forgalmi modellek (vagy ezek kombinációi) alkalmazhatóak számos alkalmazás forgalmának modellezésére. A következő táblázat tervezési irányelveket ad arra, hogy a legnépszerűbb alkalmazások modellezésére milyen típusú forgalmi modelleket használhatunk [3.3.5].

Alkalmazás	Modell	Eloszlás
TELNET	igény érkezések közötti idő	Exponenciális
	teljes hossz	Lognormális
	csomag érkezések közötti idő	Pareto
	csomag méret	többnyire 1 byte-os csomagok
FTP	igény érkezések közötti idő	Exponenciális
	elemek száma	Empirikus
	elemek mérete	Lognormális
CBR hang	igény érkezések közötti idő	Exponenciális
	teljes hossz	Exponenciális
	csomag érkezések közötti idő	Állandó
	csomag méret	Állandó
VBR videó telekonferencia	keret érkezések közötti idő	Állandó
	keret méret	Gamma
MPEG videó	keret érkezések közötti idő	Állandó
	jelenet hossz	Geometriai
	keret méret	Lognormális
WWW	igény érkezések közötti idő	Exponenciális
	dokumentum méret	Pareto

3.3.1. táblázat. Forgalmi modellek felhasználási területe és jellemzői

3.3.3. Klasszikus telefonhálózatok forgalomelméleti méretezése

A forgalomelmélet kulcsszerepet játszott a kezdetektől fogva a telefonhálózatok tervezésében és méretezésében. A telefonhívások modellezésére, a stacioner Poisson folyamatot feltételezve, a forgalom és a teljesítőképesség közötti

$$B = E(a, n) = \frac{a^n / n!}{\sum_{i=0}^n a^i / i!}$$

kapcsolatot a jól ismert Erlang veszteségi formula fejezi ki. Ez a formula megadja a hívásblokkolás valószínűségét B , ha a felajánlott forgalom a és a rendelkezésre álló vonalak száma n .

A formula kifejezi, hogy a blokkolási valószínűség a felajánlott forgalom egyszerű függvénye. Fontos megjegyeznünk, hogy a blokkolási valószínűség nem érzékeny a forgalom egyéb jellemzőire, pl. a hívás tartási idejének eloszlására. (A

formula minden $M/G/n/n$ sorbanállási rendszerre érvényes.) Ezt a formulát használták legtöbbször a forgalomelmélet történetében. A telefonhívásokat egymástól független módon, véletlenszerűen kezdeményezik, ezért a véletlen modellek, amelyek a forgalmas órákra stacioner forgalmat feltételeznek, alkalmasak voltak a mérnöki tervezésre. Mivel a telefonhívások pont-pont közötti állandó sáv szélességet lefoglaló kapcsolatok, az Erlang formula kiválóan alkalmas a telefonhálózatok tervezési feladataira.

Az Erlang formulának számos továbbfejlesztett változata van a legkülönbözőbb hálózati helyzetekre adaptálva, de az Erlang veszteségi formula (és a kapcsolódó Erlang késleltetési formula) mind a mai napig a mérnökök jól felhasználható eszköze. Kétség nélkül állíthatjuk, hogy az Erlang formula aratta eddig a legnagyobb sikert a forgalomelmélet történetében.

Az Erlang veszteségi és késleltetési formulákon kívül számos technikát fejlesztettek még ki a telefonhálózatok tervezési problémáinak megoldására. Ilyenek például az ekvivalens véletlen módszer, mely Wilkinson nevéhez fűződik, a forgalom borsztösségének különféle leírásai a csúcosság funkcionállal és diszperziós indexekkel, az Engset modell, stb. [3.3.11].

3.3.4. Az Internet forgalomelméleti méretezése

Jelenleg épp az Internet forgalomelméletének a születési pillanatait látjuk annak újdonságaival és szülési fájdalmaival együtt. Az Internet hálózatának tervezésében ma még a forgalomelmélet nem játszik jelentős szerepet és sokszor a hálózatok tervezői csak néhány tervezési ökölszabályt alkalmaznak. Amint azt a 1.7.2. fejezetben tárgyaltuk, az adatforgalom természete jelentősen eltér a beszédforgalom természetétől, és olyan hasonló univerzális szabályokat nem lehet találni, mint amilyenek a beszédforgalom modellezésénél nagy segítséget jelentettek. Új technikákat és modelleket kell kifejleszteni, hogy megbirkózzanak ezekkel a kihívásokkal. A következőkben áttekintjük az Internet forgalomelméleti tervezésének két valószínűsíthető alternatíváját. Az egyiket "nagy sáv szélesség filozófiájának", a másikat a "menedzselt sáv szélesség filozófiájának" fogjuk hívni [3.3.12].

3.3.4.1 A nagy sávszélesség filozófiája

Egy széleskörben elterjedt és jelentős álláspont manapság az, hogy nincs is igazából szükség komplikált forgalomelméletre az Internethez. Ennek az iskolának a képviselői azt állítják, hogy annak ellenére, hogy az Internet forgalma drasztikusan növekszik, a linkek kapacitása és a kapcsoló eszközök annyira olcsók lesznek a jövőben, hogy az erőforrások túlméretezése kivitelezhető lesz. Ez a *“nagy sávszélesség filozófiája”*. Érdeemes egy kicsit részletesebben megvizsgálnunk ennek az álláspontnak a realitását.

Az állítás képviselőinek a várakozása szerint az Internet átviteli és információs technológiája követni tudja az Internet forgalmának évenkénti duplázódási trendjét [3.3.10], továbbá ez a technológia olcsó megoldásokat tud majd kínálni. Technológiai szempontból ez a várakozás reálisnak mondható, legalább is a közeljövőt illetően. Gondoljunk bele, hogy ha a mai Internetben csak annyi változás történne, hogy a linkek kapacitása megnövekedne, akkor egy olyan csomagkapcsolt hálózat állna rendelkezésünkre, amely lehetővé tenné valós idejű kommunikációt mindenféle QoS támogatás nélkül! A jelenlegi “best effort” típusú Internet erre képes lenne!

Egy másik fontos tényező az adatok lokalitásának elve. Ez azt jelenti, hogy rengeteg adat helyileg lesz fontos és így a “caching” fontossága is megnő [3.3.10]. Amennyiben elképzeljük azt, hogy az összes bitet, amit ma hard drive-okon tárolnak, továbbítani szeretnénk, akkor ehhez 20 évre lenne szükség. Ez a lokalitási trend egy relatív csökkenést fog várhatóan okozni a teljes továbbítandó információ tömegben.

Egy további tényező az, hogy a “streaming” típusú forgalom, ami lényeges QoS támogatást igényel, várhatóan nem lesz domináns az Interneten [3.3.10]. Sokan várták azt, hogy ezen forgalom nőni fog, de mindez ideig ezek a várakozások nem teljesültek be és várhatóan nem is fognak. A statisztikák azt mutatják, hogy az igény ez iránt a forgalom iránt nem nő annyira, mint ahogy a linkek kapacitása nő. Vizsgáljunk meg egy egyszerű példát: legyen 1% “streaming” forgalmunk, ami QoS támogatást igényel. Két megoldási alternatíva kínálkozik. Kiépíthetünk egy QoS architektúrát, amivel megoldjuk a QoS támogatást vagy egyszerűen csak megnöveljük a linkek kapacitását 5%-al. A “nagy sávszélesség filozófia” hívei szerint a második megoldás olcsóbb. Továbbá azzal is érvelnek, hogy a multimédia alkalmazások “tárolj és továbbíts” technikát fogják használni a valós idejű “streaming”

helyett. Ezt az álláspontot támogatja az is, hogy a mágneses tárolók kapacitása ugyanolyan arányban növekszik, mint a linkek kapacitása. Mindezeken túl a különféle hálózati szűk keresztmetszetek miatt (pl. vezeték nélküli átvitel) is fontos lesz az adatok lokális tárolása.

Nagyon tanulságos az is, ha az elmúlt évek kapacitásnövekedési trendjeinek okait vizsgáljuk. Megfigyelhetjük azt, hogy az emberek általában nem azért fizetnek ADSL vagy modem hozzáférésért, mert annyi adatot akarnak továbbítani, hogy szükségük van a nagy kapacitásra, hanem azért, mert amikor ráklikkelnek egy hiperlinkre, akkor azt az oldalt azonnal akarják látni a képernyőjükön! Tehát a nagy kapacitás nem a sok adat továbbítása miatt kell, hanem a gyors hozzáférés (kis késleltetés) miatt. Ugyanez az oka annak, hogy az elmúlt tíz évben a LAN-ok átlagos kihasználtsága tizedére csökkent. Az emberek a nagy kapacitást a kis késleltetésű hozzáférés miatt fizetik meg!

Valóban az erőforrások túlméretezése lesz a megoldás a jövő Internetében? Ezt ma még senki sem tudja. Azért is meglehetősen nehéz jóslásokba bocsátkozni, mert ez a kérdés nem csak technológiai tényezőktől függ, hanem politikai és gazdasági tényezők is jelentősen befolyásolják. Azért, ha egy óvatos becslést e sorok írója megtehet, akkor az mondható el, hogy ha az erőforrások túlméretezése megoldás is lesz a gerinchálózatokban, az kevésbé valószínű, hogy a hozzáférési hálózatokban ugyanez megtörténik. Azokban az esetekben pedig, ahol a túlméretezés nem megoldás, a rendelkezésre álló erőforrásokkal kell ügyesen bánnunk. Ez vezet a második alternatívához, a *“menedzselt sávszélesség elvéhez”*.

3.3.4.2 A menedzselt sávszélesség elve

Amennyiben korlátozott hálózati erőforrás áll rendelkezésünkre, valamilyen forgalmi szabályozásra van szükség a megfelelő link kapacitás és router memória hozzárendeléséhez ahhoz, hogy a kívánt QoS követelményeket minden forgalom számára biztosítani tudjuk. A QoS követelményeknek három fő kategóriája van: transzparencia, hozzáférhetőség és throughput [3.3.7]. A *transzparencia* a továbbított adatok időbeli és szemantikai integritását fejezi ki. Például az adatátvitel szemantikai integritása általában követelmény, de a késleltetés nem annyira fontos. A *hozzáférhetőség* méri a hozzáféréskérés elutasítási valószínűségét és a kapcsolat felépítési késleltetést blokkolás esetén. Ebben a kategóriában a blokkolási

késleltetés a tipikus példa, ami egy gyakran használt jellemző telefonhálózatokban. A *throughput* a legfontosabb QoS mérték adathálózatokban. Például a mai Interneten egy 100Kbit/s throughput biztosítani tudja a legtöbb web lap kvázi azonnali továbbítását (egy másodperc alatt).

A forgalmi típusok természetük szerint két nagy csoportba oszthatóak: stream forgalom és elasztikus forgalom [3.3.7]. A *stream forgalom* olyan folyamokkal írható le, amelyeket időtartamuk és sebességük jellemez. A stream forgalom tipikus példái az audio és a valós idejű videó alkalmazások: telefon, interaktív videó szolgáltatások és videó konferencia. A stream forgalom idő integritását fontos megőrizni. A veszteség, a késleltetés és a dzsitter fontos QoS jellemzők ezen forgalomtípus esetén.

Az *elasztikus forgalom* digitális objektumokból (dokumentumok) áll, melyeket egyik helyről egy másikra szeretnénk továbbítani. A forgalom elasztikus, mert a folyam sebessége változhat külső hatások következményeképpen (pl. szabad kapacitás). Tipikusan elasztikus alkalmazások a web, az e-mail és a file transzfer. Elasztikus forgalom esetén a szemantikus integritást fontos megőrizni. Az elasztikus forgalom leírható az igények érkezési folyamatával és az objektumok méretének eloszlásával. A throughput és a válaszadási idő a tipikus QoS mértékei ennek a forgalomtípusnak.

A következő két alfejezetben áttekintjük a stream és az elasztikus forgalom menedzselésének alapelveit.

3.3.4.3 A streaming forgalom nyílt hurkú szabályozása

A streaming típusú forgalmat általában forgalmi szerződésre alapuló *nyílt hurkú, megelőző forgalmi szabályozás* módszereivel szabályozzák [3.3.7]. A forgalmi szerződés egy olyan sikeres megállapodás a felhasználó és a hálózat között, melyben a felhasználó forgalmi igényeit leíró paraméterek, és az igényelt QoS paraméterek szerepelnek. Ezen információk segítségével a hálózat végrehajt egy belépés szabályozási algoritmust, amely a kapcsolatot csak akkor fogadja el, ha az igényelt QoS biztosítható és a meglévő kapcsolatok QoS paraméterei sem sérülnek.

A szabályozás hatékonysága nagymértékben függ attól, hogy a teljesítményjellemzők milyen pontosan becsülhetőek a *forgalom leíró paraméterek*

segítségével [3.3.6]. A gyakorlat azt mutatja, hogy nem könnyű jó forgalmi jellemzőket találni. A forgalmi jellemzőknek *egyszerűnek* (érthetőek legyenek a felhasználó számára), *hasznosnak* (erőforrás allokációs szempontból) és *szabályozhatónak* (a hálózat számára ellenőrizhetőnek) kell lenniük. A gyakorlat azt mutatja, hogy mindhárom szempontból ideális forgalomleíró lehetetlen találni. Például az ATM és az Internet szabványosítási törekvéseiben az elfogadott *token bucket forgalomleírók* jól szabályozható és ellenőrizhető forgalomleírók, de kevésbé hasznosak erőforráshozzárendelés szempontjából. A felhasználók használhatnak eljárásokat (pl. forgalomsimítás), amikkel biztosíthatják a deklarált forgalmi jellemzők betartását. A hálózat pedig a belépési pontokon alkalmazhat eljárásokat (forgalom felügyelet), amik ellenőrzik a deklarált forgalmi jellemzőket. Ezek az eljárások legtöbbször a már említett token bucket módszerekre épülnek.

A legfontosabb típusai a nyílt hurkú forgalomszabályozásnak jelentősen eltérnek attól függően, hogy statisztikus multiplexálási nyereséget akarunk-e elérni vagy nem [3.3.7]. A következő tábla mutatja a fő kategóriákat:

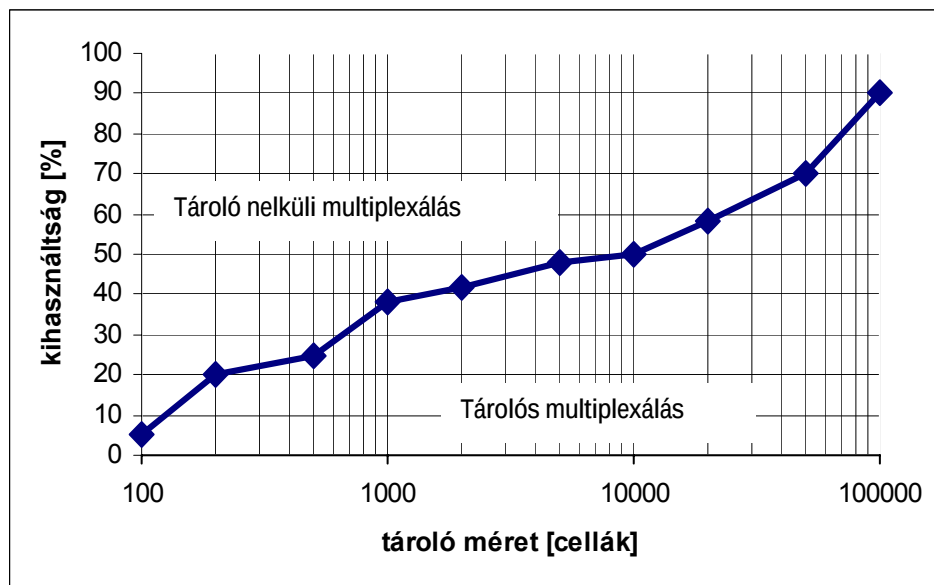
Eljárás	Tároló megosztás	Sávszélesség megosztás
cúcssebességű kapacitás hozzárendelés	NEM	NEM
sebesség burkoló multiplexálás	NEM	IGEN
kapacitás megosztás	IGEN	IGEN

Amennyiben multiplexálási nyereséget nem kívánunk elérni, a legegyszerűbb eljárást, a csúcssebességű kapacitás kiutalást alkalmazhatjuk. Ez a módszer egyszerűen minden kapcsolat számára annak maximális sebességének megfelelő sávszélességet foglal le. A módszer előnye, hogy az egyetlen forgalmi jellemző a csúcssebesség. A beléptetési szabályozás nagyon egyszerű, csak azt kell ellenőrizni, hogy az igényelt csúcssebességek összege meghaladja-e a teljes kapacitást. A fő hátránya ennek a módszernek, hogy nagyon pazarló a kapacitással, mert statisztikailag csak az idő igen kis hányadában fordul az elő, hogy az összes kapcsolat csúcssebességgel továbbít.

Ha a sávszélességet statisztikailag megosztjuk a kapcsolatok között, de a tárolókapacitást még nem, akkor a *rate envelope multiplexing* (sebesség-burkoló multiplexálás) esetéhez jutunk [3.3.6], [3.3.7], [3.3.8]. Az eljárást *tárolónélküli multiplexálásnak* is hívják, mert a folyadékmodelles analízisben nincs tárolóra

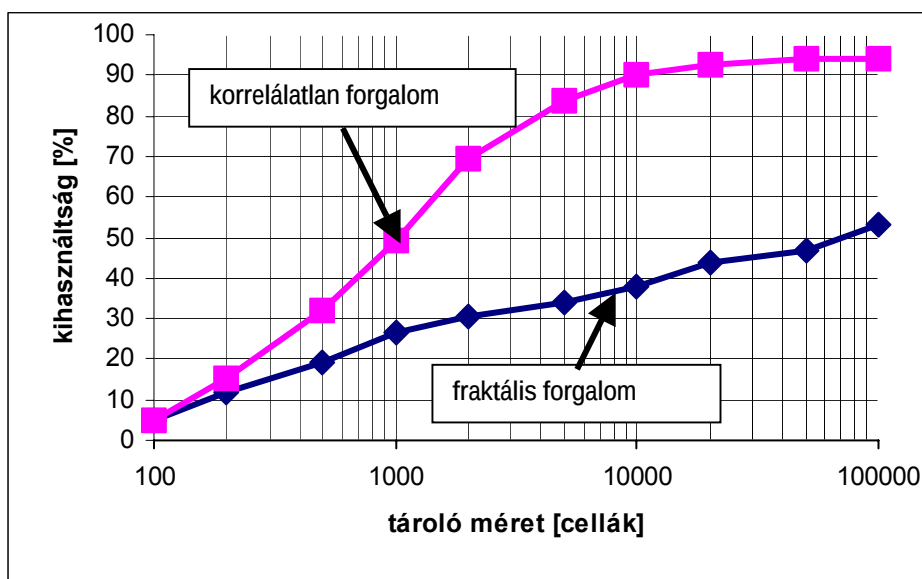
szükség. A rate envelope multiplexing módszer esetében a cél az, hogy az aggregált folyam sebessége a teljes kapacitás alatt maradjon. Annak az eseménynek a valószínűsége, hogy az aggregált sebesség meghaladja a kapacitást, adott érték alatt kell maradjon, $P(\Lambda_t > c) < \varepsilon$, ahol Λ_t az aggregált forgalom sebessége, c a link kapacitása és ε a megengedett túlcsoordulási valószínűség. A valóságban tárolók mindig kellenek, hogy az egyszerre érkező csomagok ne vesszenek el (cella szintű torlódás). A túlcsoorduló forgalom elveszik és az átlagos veszteségi ráta: $E[(\Lambda_t - c)^+ / E(\Lambda_t)]$. A veszteségi ráta csak Λ_t stacioner eloszlásától függ, és nem függ annak időbeli függőségi struktúrájától. Ez egy fontos tényező, mert ez azt jelenti, hogy a korrelációs struktúrának nincs hatása a veszteségre. Ennek pedig az a fontos következménye, hogy a forgalommodellezés azon nagyon nehéz feladatára, hogy a korrelációs összefüggéseket pontosan leírjuk, nincs szükség. A forgalmi struktúrának valójában van hatása a teljesítményjellemzőkre, de ezek elhanyagolhatóak, ha a veszteség kicsi. Például LRD forgalom eredményezhet hosszabb veszteségi periódusokat mint SRD forgalom, de ez a hatás elhanyagolható kis veszteségnél. A legfőbb hátránya ennek a módszernek, hogy az elérhető kihasználtság még mindig nem olyan jó.

Amennyiben a link kihasználtságot tovább akarjuk növelni a tárolót is statisztikusan meg kell osztanunk, lásd a 3.3.1. ábrát. Ez a *rate sharing* (kapacitás megosztás) módszere [3.3.6], [3.3.7], [3.3.8] amit *tárolós multiplexelésnek* is hívnak. A módszer ötlete az, hogy a tároló segítségével is eliminálhatunk néhány túlcsoordulási veszteségi periódust. A cél a tárolóban levő sorhossz adott alacsony valószínűség alatt tartása, $P(Q > q) < \varepsilon$, ahol q a megengedett sorhossz is, Q az aktuális sorhossz és ε a megengedett valószínűsége annak, hogy a sorhossz meghaladja a megengedett sorhosszt. Ezzel a módszerrel sokkal nagyobb multiplexálási nyereség és kihasználtság érhető el.



3.3.1. ábra. Statisztikus multiplexálási alternatívák

A rate sharing (kapacitás megosztás) legfőbb hátránya az, hogy a veszteség adott tárolóméretnél és link kapacitásnál elég bonyolult módon függ a forgalmi jellemzőktől, beleértve a korrelációs struktúrát is. Például a veszteségi és késleltetési jellemzők nagyon bonyolultan számíthatóak, ha a forgalom LRD. Ez az oka annak, hogy a hívás belépést engedélyező eljárások sokkal bonyolultabbak, mint a rate envelope multiplexing esetében [3.3.8]. További probléma, hogy a komplex forgalomszabályozás dacára az elérhető kihasználtság még ennél az esetenél sem olyan nagy, ha a forgalom fraktális természetű, erős SRD és LRD tulajdonságokkal, lásd a 3.3.2. ábrát.



3.3.2. ábra. A korrelációs struktúra hatása

Sokféle hívés belépést engedélyező eljárást dolgoztak ki mind a rate envelope multiplexing, mind a rate sharing esetére [3.3.8]. A tapasztalat azt mutatja, hogy a legeredményesebb eljárások a mérés alapú módszerek, ahol az egyetlen forgalomleíró a csúcssebesség és a pillanatnyi elérhető sebességet valós időben becsüljük.

3.3.4.4 Az elasztikus forgalom zárt hurkú szabályozása

Az elasztikus forgalmat általában *reaktív zárt hurkú forgalomszabályozó módszerekkel* menedzselik [3.3.6], [3.3.7]. Ez az elve a TCP-nek az Interneten és az ABR-nek az ATM-ben. Ezek a protokollok a maximális szabad sávszélesség kihasználtságra törekszenek úgy, hogy a fennálló kapcsolatok között a sávszélességet fair módon osszák szét. Most a TCP-t vizsgáljuk meg, ami az Internet általánosan használt átviteli protokollja. A TCP-ben egy additív növelési és multiplikatív csökkentési algoritmust implementáltak. Amíg nincs csomagvesztés, a sebesség lineárisan nő, de csomagvesztés esetén a csomagtovábbítási sebességet az algoritmus felezi. Az algoritmus megpróbál egy olyan átlagsebességre beállni, ami a kapacitás és a pillanatnyi forgalom jellemzőitől függ. Az elérhető sávszélesség közelítőleg fair módon oszlik meg a TCP folyamatok között.

A TCP-nek az egyszerű modellje [3.3.9], ami a legfontosabb viselkedését leírja, a jól ismert négyzetgyökös összefüggés a throughput (B) és a csomagvesztés (p) között:

$$B(p) = \frac{c}{RTT\sqrt{p}},$$

ahol RTT a TCP folyam körülfordulási ideje, és c egy konstans. Fontos megjegyeznünk, hogy ez az egyszerű formula csak számos feltétel mellett igaz: RTT állandó, p kicsi (1% alatt van) és a TCP forrásnak mindig van adata továbbításra. A TCP eljárásról feltételezzük, hogy a fast retransmit és recovery eljárások működnek (nincs timeout) és a slow-start fázist nem modellezzük. Ennél sokkal bonyolultabb TCP modellek is ismertek, de a négyzetgyökös összefüggés egy elég általános szabálya a TCP-nek.

Hívásbelépési eljárások kifejlesztése elasztikus forgalmakra egy nem lezárt kutatási terület [3.3.6], [3.3.7]. Ezekben az eljárásokban a szabályozásoknak úgy kell működniük, hogy biztosítsák a megfelelő throughput-ot túlterheléses állapotokban is, de másrésztől elkerüljék a folyamatok visszautasítását normális terhelési viszonyok között.

3.3.5. Záró gondolatok a forgalmi méretezésről

A megfelelő forgalmi modell választásának az a tétje, hogy milyen pontosan tudjuk megragadni a forgalom legfontosabb jellemzőit. Az így kiválasztott a forgalmi modellt alkalmazzuk a legtöbb forgalomelméleti rendszerben, ami a leggyakrabban egy sorbanállási rendszer. A legfontosabb kérdés az alapvető összefüggés a forgalmi jellemzők, a hálózati erőforrások és a teljesítményjellemzők között. Több sorbanállási rendszer analitikusan kezelhető (pl. Poisson, MMPP, MAP, stb.), de van több olyan is, amik nagyon nehezen analizálhatóak analitikusan (pl. ARIMA, TES, FGN, stb.). Jelenleg is fontos kutatási feladat olyan új technikák és modellek kifejlesztése, amelyek jól kezelhetőek és ugyanakkor jól megragadják a valóságos rendszer jellemzőit.

A fenti áttekintésünk mutatja, hogy az Internet forgalmi méretezése még nem megoldott feladat és számos nyitott probléma vár megoldásra. Ellentétben a telefonhálózatok forgalomelméletével, ami egy kiforrott és jól megértett terület, az

Internet forgalomelmélete és méretezési eljárásai a jövő kutatási feladatai közé tartoznak.

Irodalomjegyzék

- [3.3.1] D. L. Jagerman, B. Melamed, W. Willinger: Stochastic modeling of traffic processes, In J. Dshalalow, ed., *Frontiers in Queueing: Models, Methods and Problems*. CRC Press, 1997. pp. 271-320.
- [3.3.2] V. S. Frost, B. Melamed: Traffic Models for Telecommunications Networks, *IEEE Communications Magazine*, March 1994. pp 70-81.
- [3.3.3] G. D. Stamoulis, M. E. Anagnostou, A. D. Georganas: traffic sources models for ATM networks: a survey, *Computer Communications*, vol. 17, no. 6, June, 1994. pp. 428-438.
- [3.3.4] R. G. Addie, M. Zukerman, T. D. Neame: Broadband Traffic Modeling: Simple Solutions to Hard Problems, *IEEE Communications Magazine*, August 1998. pp. 88-95.
- [3.3.5] B. O. Lee, V. S. Frost, R. Jonkman: NetSpec 3.0 source Models for telnet, ftp, voice, video and WWW traffic, 1997.
- [3.3.6] J. Roberts, Traffic Theory and the Internet, *IEEE Communications Magazine*, January 2000.
- [3.3.7] J. Roberts, Engineering for Quality of Service, in the book of Self-Similar Network Traffic and Performance Evaluation, (eds. K. Park, W. Willinger), Wiley, 2000.
- [3.3.8] J. Roberts, U. Mocci, J. Virtamo (eds.), *Broadband Network teletraffic*, Springer-Verlag, 1996.
- [3.3.9] J. Padhye et al. Modeling TCP Throughput: A Simple Model and Its Empirical validation, *Proc. SIGCOMM'88*, ACM, 1998.
- [3.3.10] A. Odlyzko: The history of communications and its applications for the Internet, available at <http://www.research.att.com/~amo/doc/complete.html>, 2000.
- [3.3.11] H. Akimaru. K. Kawashima: *Teletraffic, Theory and Applications*, Springer-Verlag, 1999.
- [3.3.12] S. Molnár, IP hálózatok forgalmi méretezése, *Magyar Távközlés*, September 2000.

3.4. A közös csatornás jelzésrendszeri koncepció

Szerző: dr. Adamis Gusztáv

Lektor: dr. Csopaki Gyula

A hagyományos, csatornához rendelt jelzésrendszerek, vagyis, amikor a beszédutat felépítő jelzések a későbbi beszédútvonalon haladnak, nem megfelelőek az ISDN hálózatok számára.

A jelzések és a beszélgetések (adatok) továbbítása eltérő követelményeket támaszt az átvivő közeggel szemben.

A beszélgetés során viszonylag hosszú időn keresztül nagyon pontosan ütemezve (125 μ s-onként) kell továbbítani egyenként 1 bájtos, de összességében jelentős mennyiségű információt.

A jelzések viszont nem igénylik a szigorú ütemezést, információtartalmuk viszonylag csekély és az egyes jelzések között akár hosszú idő is eltelhet.

Ez a felismerés adta az ötletet, hogy válasszuk szét egymástól a beszéd- (ISDN) hálózatot és a jelzeshálózatot.

A jelzeshálózaton megoldható, hogy amíg egy beszédkapcsolat következő jelzését nem tudjuk elküldeni, addig ebben az "üres" időben más beszédkapcsolat(ok) jelzéseit tudjuk továbbítani ugyanazon az útvonalon.

Ha a jelzéseket digitális formátumban továbbítjuk, akkor a jelzések gazdagon paraméterezhetőek, viszonylag komplikált szerkezetűek lehetnek, amely lehetővé teszi, hogy ugyanaz a jelzésrendszer többféle szolgálat jelzési igényeit ki tudja elégíteni.

A beszéd- illetve a jelzeshálózatoknak eltérő biztonsági követelményeknek kell megfelelniük. Hiszen - sarkítva fogalmazva - amíg egy hiba a beszédhálózatban csak egy (néhány) beszélgetés megszakadásával jár, addig egy elvesztett jelzésüzenet akár azt is okozhatja, hogy például egy beszédösszeköttetés "soha" nem bontódik le, sokkal nagyobb veszteséget okozva. A beszéd- és a jelzeshálózat szétválasztásával megteremtődik annak a lehetősége, hogy ki tudjuk elégíteni a jelzeshálózat szigorú biztonsági követelményeit anélkül, hogy feleslegesen ugyanilyen követelményeknek

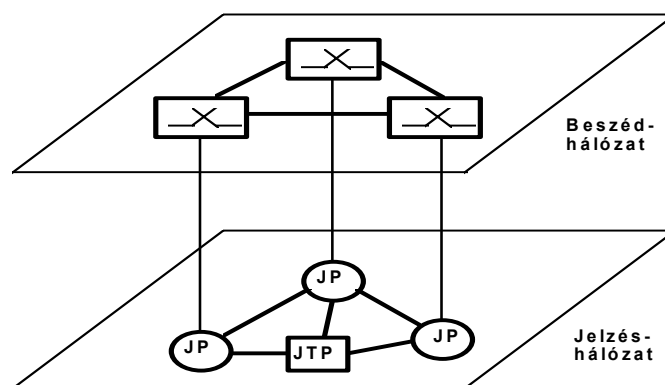
megfelelően építenénk ki a beszédhálózatot is. Az elkülönült jelzeshálózat továbbá alkalmas lehet egyéb, belső hálózati információ (fenntartás, teljesítmény monitoring, hálózatmenedzselés stb.) továbbítására is.

Ugyanakkor meg kell említenünk néhány jelentős hátrányt is:

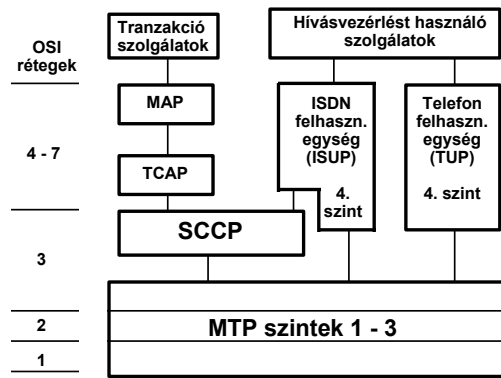
Az elkülönült jelzeshálózat kiépítése, fenntartása plusz költséggel jár. A jelzések szerkezete bonyolultabb lesz, hiszen azonosítani kell tudni, hogy melyik jelzés melyik kapcsolatra vonatkozik, valamint, többé nem biztos, hogy a jelzeshálózaton "megbeszél" beszédútvonal tényleg felépült, azt esetleg külön ellenőrizni kell. (A csatornához rendelt jelzések esetén ilyen probléma nem volt, hiszen, ha egy jelzés megérkezett, akkor ugyanazon az úton később a beszélgetés is lebonyolódhat majd.).

Mindezen hátrányok azonban eltörpülnek az előnyök mellett, így a közös csatornás jelzésrendszer egyre nagyobb jelentőséggel bír.

Az 3.4.1. ábrán a közös csatornás jelzés koncepciót láthatjuk. A telefonközpontok mindegyike tartalmaz egy elkülönült funkcionális egységet, amely a jelzéseket kezeli. Ezek az (ábrán körrel jelölt) jelzőpontok (JP). Mint az ábrából látható, a jelzésösszeköttetések független útvonalon mehetnek a beszédútvonalakhoz képest, sőt a hálózat optimalizálása, biztonságának növelése céljából további, (ábrán négyzettel jelölt) jelzéstovábbító pontokat (JTP) is tartalmazhatnak.



3.4.1. ábra. A jelzeshálózati koncepció



3.4.2. ábra. A CCSS7 architektúra

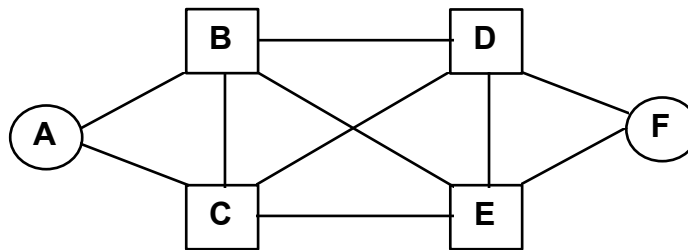
A CCITT/ITU-T kidolgozott és elfogadott egy nemzetközileg szabványos közös csatornás jelzésrendszert, ez a CCSS7 (Common Channel Signalling System 7).

A CCSS7-es leegyszerűsített architektúrális felépítése a 3.4.2. ábrán látható. Felépítése - az OSI modellhez hasonlóan - hierarchikus, de az egyes hierarchia szintek nem pontosan egyeznek meg az OSI modell rétegeivel. Ezért itt a réteg (layer) helyett a szint (level) kifejezést használják.

Az alsó három szint - az úgynevezett üzenettovábbító egység (Message Transfer Part - MTP), - a jelzésüzeneteknek ugyanazon jelzeshálózat tetszőleges pontjától tetszőleges más pontjára való eljuttatásáért felelős. A 4. szinten találhatók a felhasználói egységek (User Part - UP), amelyek a jelzéseket generálják a tranzakciós szolgálatok (pl. GSM), illetve a hívásvezérlést használó szolgálatok (telefon, ISDN) számára. Az SCCP feladata logikai jelzés összeköttetések kialakítása, elsősorban a tranzakciós szolgálatok számára.

Mint már említettük, a jelzeshálózatnak szigorú megbízhatósági követelményeknek kell eleget tennie. Ezért nagy fontossággal bír az, hogy tetszőleges két jelzőpont között több lehetséges útvonal is létesülhessen. A 3.4.3 ábrán bemutatjuk azt a hálózati topológiát, amelyet a nemzetközi hálózatban ki kell alakítani két tetszőleges (az ábrán A és F) jelzőpont között. A hálózat lényege, hogy minden pontban - legalább - két lehetséges továbbvivő útvonalnak kell lennie (például A-ból az AB és az AC, B-ből a BD és a BE stb.) és ezeken az útvonalakon - normális körülmények között - 50%-os terhelésmegosztást kell alkalmazni, tehát például a A-ból F felé menő forgalom egyik felét az AB, a másik felét az AC szakaszra kell irányítani, stb. A biztonság fokozása érdekében a jelzéstovábbító pontok között haránt összeköttetéseket is kialakítanak (BC és DE szakaszok),

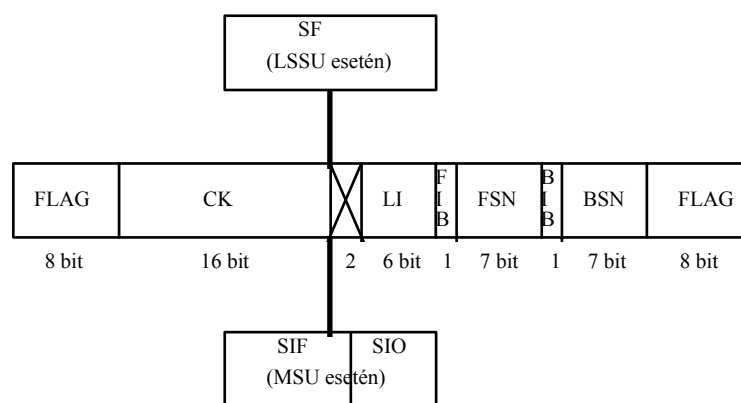
amelyeken - normális körülmények között - az adott irányban (jelen példánkban A-F között) nem halad forgalom.



3.4.3. ábra. A jelzeshálózati szabványhálózat

MTP 1. szint: Jelzéskapcsolat (Signalling Data Link level) szint a jelzésszakaszok fizikai, elektromos jellemzőit, jelalakjait, hozzáférési módjait, csatlakozóit stb. specifikálja. A CCSS7 a duplex 64 kb/sec korlátozás nélküli digitális transzparens csatornára optimalizált, de lehetőség van más - akár analóg - átvivő közegek használatára is.

MTP 2. szint: Jelzésszakasz (Signalling Link level) szintjének feladata, hogy két szomszédos jelzéspontot összekötő jelzésszakaszokon biztosítsa a jelzésinformáció hibamentes átvitelét. Ezt úgy oldja meg, hogy a magasabb szintek felől érkező jelzésinformációt kiegészíti jelzésszakasz vezérlő résszel, mely a következő részekből áll (3.4.4. ábra):



3.4.4. ábra. MTP 2. szintű jelzéselem formátum

- F: flag, értéke: 01111110. Ennek feladata az egyes jelzésüzenetek egymástól való elválasztása. Természetesen meg kell oldani, hogy a jelzésüzenetekben máshol ne alakulhassanak ki ilyen alakú oktettek, ezt a jól ismert mechanizmussal, a minden öt egymást követő 1-es után történő 0 beszúrással érik el.
- BSN: hátra sorszám. Az utoljára vett jelzésüzenet sorszáma (modulo 128), egyben (pozitív) nyugtát is jelent az adott és az azt megelőző üzenetekre.

- BIB: hátra indikátor bit. Ha a jelzőpont hibás jelzésüzenetet vesz, akkor ennek a bitnek az invertálásával jelezheti ezt az ellenoldalnak ("negatív nyugta").
- FSN: előre sorszám. A küldött jelzésüzenet sorszáma (modulo 128).
- FIB: előre indikátor bit. Ennek invertálásával jelezheti a jelzőpont egy korábban már elküldött jelzésüzenet megismétlését.
- LI: hosszindikátor, mely az üzenet információs részének hosszát jelzi oktettekben mérve. Ennek alapján lehet különbséget tenni a háromféle üzenettípus között.

A kitöltő jelzésem (FISU - Fill-In Signal Unit) nem tartalmaz információs mezőt; akkor használják, ha pillanatnyilag nincs küldeni való, "hasznos" üzenet.

A periodikusan küldött szakaszállapot jelzésem (LSSU - Link Status Signal Unit) esetén az 1 vagy 2 oktettes információs mezőt szakaszállapot mezőnek hívják, ez az adott szakasz állapotát jelzi (pl. használható, torlódott stb.).

A harmadik típus a felhasználói jelzésinformációt hordozó üzenet jelzésem (MSU - Message Signal Unit). Ilyenkor az információs mező két részre oszlik: a szolgálat információs oktett (SIO) jelzi, hogy az adott jelzésüzenet milyen felhasználói egységnek szól, míg a másik, a jelzésinformációs mező (SIF) tartalmazza magát a jelzésinformációt + a címet.

- CK: 16 bites ellenőrző összeg.

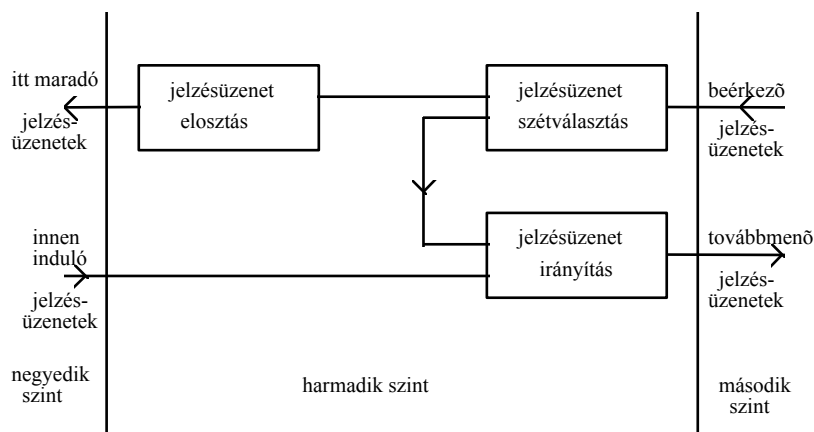
Mint a fentiekből látható, a második szinten a hagyományos forgóablakos protokollt használják (FSN, BSN) 127-es ablakmérettel, explicit negatív nyugtázási lehetőséggel (BIB invertálás).

MTP 3. szint: Jelzeshálózat (Signalling Network level) szint feladata kettős.

Az egyik feladata az, hogy a jelzeshálózatban belül el tudja juttatni a jelzésüzeneteket tetszőleges jelzőponttól tetszőleges másik jelzőpontig. Ezt a feladatot végzi el a jelzésüzenet kezelési (Signalling Message Handling) funkció. A másik funkciója pedig az, hogy a jelzeshálózat struktúrájában (pl. hiba, javítás, új elemek betétele stb. miatt) fellépő változások nyomán átkonfigurálja az irányítási rendszert. Ezt a feladatot a jelzeshálózat menedzselési (Signalling Network Management) funkció látja el.

Jelzésüzenet kezelés blokkvázlata az 3.4.5. ábrán látható. A beérkező üzenetekről az üzenet szétválasztási funkció dönti el, hogy az adott jelzőpontnak szól-e vagy pedig továbbítani kell más jelzőpont felé. Ha az adott jelzőpontnak szól az üzenet, akkor azt, hogy az melyik felhasználói egységnek szól (ISUP stb.), az

üzenet elosztási funkció dönti el. Az adott jelzőpont által küldött, illetve az üzenetszétválasztás által továbbítandónak talált üzenetek elküldéséért az üzenet irányítási funkció felelős.



3.4.5. ábra. A jelzésüzenet kezelés funkcionális blokkvázlata

Az üzenetek irányítására a 3.4.6. ábrán látható címzési rendszert használják. A már említett szolgálat információs oktett (mely azt a felhasználói egységet választja ki, amelynek az üzenet szól) követi az úgynevezett iránycímke. Az iránycímke három részre osztható. A rendeltetési (DPC - Destination Point Code) és a kezdeményező pont (OPC - Originating Point Code) kódja egyaránt 14 bites. Az iránycímke harmadik eleme a jelzésszakasz kiválasztó (SLS - Signalling Link Selection) kód, mely 4 bites. Ez azt a célt szolgálja, hogy két jelzőpontot összekötő (a biztonság növelése illetve a terhelésmegosztás céljából kialakított) legfeljebb 16 különböző útvonal közül kiválasszon egyet.

SLS	Kezdeményező pont kód (OPC)	Rendeltetési pont kód (DPC)	Szolgálat információs oktett (SIO)
4 bit	14 bit	14 bit	8 bit

3.4.6. ábra. Az MTP harmadik szint címzési formátuma

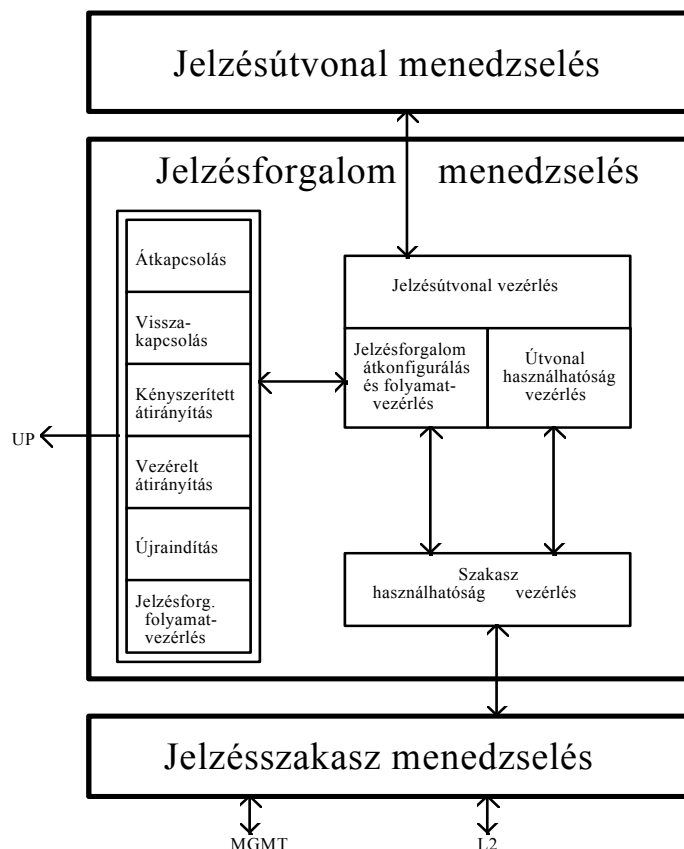
Jelzeshálózat menedzselés

A jelzeshálózat menedzselés blokkvázlata a 3.4.7. ábrán látható.

3.4.7. ábra. A jelzeshálózat menedzselés blokkvázlata

A jelzésszakasz menedzselés (Signalling Link Management - SLM) feladata, hogy az operátoroktól ("menedzsment" - MGMT) illetve a 2. szint (L2) bithiba-arány figyelő funkciója felől érkező jelzések alapján detektálja a szakaszok használhatóságában bekövetkező változásokat és ezeket jelentse a jelzésforgalom menedzselés számára.

A jelzésforgalom menedzselés (Signalling Traffic Management - STM) feladata egyrészt az, hogy nyilvántartsa az adott jelzéspontból kiinduló illetve oda érkező jelzésszakaszok és jelzésútvonalak állapotát, másrészt az, hogy a



jelzésszakasz menedzseléstől kapott információ alapján módosítsa saját irányítási tábláit.

A jelzésútvonal menedzselés (Signalling Route Management - SRM) feladata pedig az, hogy a jelzésforgalom menedzselés által végrehajtott változásokat a hálózatban a változtatásban közvetlenül nem érintett jelzéspontok felé is "elterjessze".

Összetettsége miatt nézzük meg a jelzésforgalom menedzselés felépítését részletesebben!

A szakasz használhatóság vezérlés (Link Availability Control) tartja nyilván az egyes szakaszok használhatósági állapotait táblázatok segítségével. Ez az eljárás, amely a jelzésszakasz menedzseléstől kapott információ alapján módosítja a táblázati bejegyzéseket, illetve kezdeményezi a szükséges szakasz irányítás módosítási eljárásokat.

Hasonló szerepe van az útvonal használhatóság vezérlés (Route Availability Control) eljárásnak az útvonalakat tekintve.

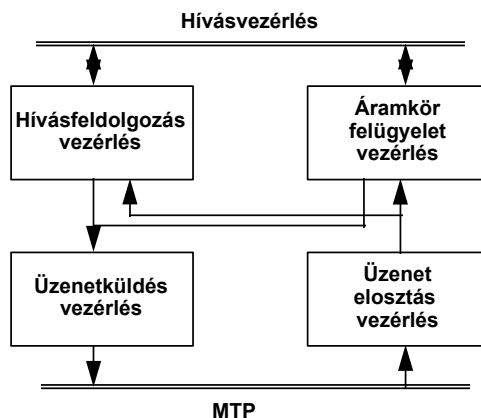
A jelzésútvonal vezérlés eljárás szerepe az, hogy ha az állapotváltozások - a változásokban közvetlenül érintetteken kívül - más jelzéspontokat is érintenek/érinthetnek, akkor kezdeményezze a jelzésútvonal menedzselésnél a megfelelő (átvitel letiltás illetve átvitel engedélyezés) eljárást.

A jelzésforgalom átkonfigurálás és folyamat vezérlés eljárás feladata az igények alapján az irányításmódosítási eljárások tényleges indítása. Ezek az eljárások a következők:

- Átkapcsolás (Changeover). Egy nem használhatóvá váló szakaszból a forgalom áterelése egy használhatóra.
- Visszkapcsolás (Changeback). Egy használhatóvá váló szakaszra a forgalom visszaterelése.
- Kényszerített átirányítás (Forced Rerouting). Egy nem használhatóvá váló útvonalról az adott útvonal forgalmának áterelése egy használhatóra.
- Vezérelt átirányítás (Controlled Rerouting). Egy használhatóvá váló útvonalra a forgalom visszaterelése.
- Újraindítás (Restart). A jelzéspont újraindítása.
- Jelzésforgalom folyamatvezérlés (Signalling Traffic Flow Control). Célja, hogy jelzésponthoz tartozó felhasználói egységeket értesítse a jelzésálózatban fellépő torlódásról és felkérje őket az általuk generált forgalom mérséklésére.

4. szint: ISDN felhasználói egység (ISUP)

Eddig azt láttuk, hogy a CCSS7 hogyan tudja eljuttatni az üzeneteket a hálózat egyik pontjától a másikig, most nézzünk meg egy felhasználói egységet. Ez legyen az ISUP (ISDN User Part), amely generálja az ISDN (és azon belül a telefon) jelzéseket a központ hívásvezérlési funkciója utasításainak alapján. (3.4.8. ábra)



3.4.8. ábra. Az ISUP felépítése

A hívásfeldolgozással kapcsolatos tevékenységek és üzenetek mellett szükség van az áramkörfelügyeleti (értsd beszédáramkör-felügyeleti) tevékenységekre és üzenetekre is azért, hogy jelzések segítségével a beszédáramköröket is menedzselhessük. A másik két blokk (üzenetküldés vezérlés és üzenet elosztás vezérlés) önmagáért beszél.

Az ISUP jelzések felépítése

Az ISUP jelzései funkcionálisan sokban hasonlítanak a hagyományos telefonos, központok között használt jelzésrendszerek jelzéseire. A fő különbség a digitális formátum, és az egyes jelzések gazdag paraméterezhetősége. Ezt illusztrálандó, nézzük meg egy általános ISUP üzenet felépítését (3.4.9. ábra)!

Iránycímke
Áramkör azonosító kód
Üzenettípus kód
Az 1. kötelező fix hosszú pm. értéke
Az utolsó kötelező fix hosszú pm. értéke
Pointer az 1. köt. változó hosszú pm-re
Pointer az utolsó köt. változó hosszú pm-re
Pointer az opcionális rész elejére
→ Az 1. köt. vált. hosszú pm. hossza
Az 1. köt. vált. hosszú pm. értéke
→ Az utolsó köt. vált. hosszú pm. hossza
Az utolsó köt. vált. hosszú pm. értéke
→ Az 1. opcionális pm. neve
Az 1. opcionális pm. hossza
Az 1. opcionális pm. értéke
Opcionális paraméterek vége mező

3.4.9. ábra. ISUP üzenet formátum

Minden egyes ISUP üzenet tartalmazza a már említett iránycímkét, illetve egy (beszéd)áramkör azonosító kódot, amely segítségével azonosítani lehet azt a beszédáramkört, amelyre az adott ISDN jelzés vonatkozik.

Ezt a mezőt követi az üzenet típusát azonosító kód.

Minden egyes üzenet több paramétert tartalmazhat. Ezek közül lehetnek olyan paraméterek, melyeknek kötelezően szerepelniük kell az adott üzenetben. (Például egy címüzenetben a tárcsázott szám kötelező paraméter.) A kötelező paraméterek között lehetnek olyanok, melyek hossza mindig azonos és előre tudható, illetve olyanok, amelyek hossza különböző lehet (például a már említett tárcsázott szám is ilyen, hiszen különböző helyekre - Budapest, vidék, nemzetközi - szóló telefonszámok hossza különböző). Ezek figyelembe vételével beszélhetünk kötelező fix hosszúságú és kötelező változó hosszúságú paraméterekről.

A kötelező paramétereket követ(het)ik az opcionális paraméterek. Az opcionális paraméterek olyan paraméterek, amelyek nem kötelezően tartozékaik egy-egy üzenetnek, hanem például attól függően vannak az üzenetben, hogy egy bizonyos szolgáltatást igénybe vettünk-e vagy sem. (Például a már említett címüzenet opcionális paramétere a hívó fél száma.)

Az egész jelzésüzenetet egy, az opcionális paraméterek végét jelző kód zárja.

Az ISUP jelzések osztályozása

Az ISUP jelzések funkciójuk szerint négy csoportba oszthatók:

1. Hívásfelépítési, -felügyeleti és -elbontási üzenetek
2. Felépített hívás módosítási üzenetek
3. Áramkör és áramkörcsoport felügyeleti üzenetek
4. Vég-vég üzenetek (jelenleg nem használják).

A legfontosabb hívásfelépítési, -felügyeleti és -elbontási üzenetek

IAM - első címüzenet (impliciten a kimenő beszédáramkör lefoglalása is) - előre

SAM: - további címüzenet (ha a hívószámot több részletben továbbítjuk, akkor a hívószám második stb. részletét viszi).

ACM - cím teljes - előre

CPG - hívás folyamatban (például hívás átirányításnál használjuk) - hátra

ANM - a hívott válaszolt (ennek vételekor kezdődik a díjazás) - hátra

REL - bontás (bármely irányban)

RLC - bontás nyugtázása (bármely irányban)

A legfontosabb felépített hívás módosítási üzenetek

FAR - szolgáltatás kérés

FAA - szolgáltatás kérés nyugtázás (teljesítés)

FRJ - szolgáltatás kérés elutasítása

A legfontosabb áramkörfelügyeleti üzenetek

CCR - folytonosság vizsgálat kérés (annak az ellenőrzését kéri, hogy a felépített beszédút valóban folytonos-e).

LPA - visszahurkolás (nyugta a CCR-re)

OLM - túlterhelés

SUS/RES - felfüggesztés / visszavétel

BLO/BLA - blokkolás kérés / nyugta

UBL/UBA - blokkolásból felszabadítás kérés / nyugta

CCITT Specifications of Signalling System No. 7. Recommendations Q.704. - Q.705.

ETSI Integrated Services Digital Network: Application of the ISDN user part of CCITT Signalling System No. 7. for international ISDN interconnections - CCITT Recommendation Q.767.

3.5. TCP/IP hálózatok és IP telefonia

Szerző: Szabó Róbert

Lektor: dr. Réthy György

3.5.1. Definíció

„Az Internet telefonia (IPT) jellemezhető az Interneten történő hívás megvalósítással függetlenül attól, hogy a hívásban szereplő eszközök közt hagyományos telefonhálózati eszközök is szerepelnek-e; valamint hogy a kommunikációs út csak egy részében vagy teljes egészében az Interneten keresztül húzódik.” – követve Jiri Kuthant.

3.5.2. Áttekintés

Az IP telefonia mögötti hajtóerőt leginkább a *költség- takarékoság*, illetőleg az újszerű szolgáltatások könnyű megvalósíthatósága jelenti. A költséghatékonyság nyilvánvaló, ha az integrált szolgáltatású hálózati trendeket követjük, ahol ugyanazt az egyszeresen kiépített hálózati infrastruktúrát – esetünkben az IP alapú gerinchálózatot – használjuk többszörös céllal: mint Internet és telefon szolgáltatás nyújtásának alapját. A felmerülő probléma azonban esetünkben két hagyományosan eltérő technológiának az összevonásában rejlik, hiszen a hagyományos áramkörkapcsolt technológia verseng a csomagkapcsolt hálózattal. Természetesen a jövő csomagkapcsolt hálózatának nagyon sok áramkörkapcsolt tulajdonsággal kell rendelkezni, ha versenyképes alternatívát kíván adni; ilyenként említhetjük a garantált minőségű szolgáltatások nyújtását, melyben az Internet igencsak elmaradt. Az is nyilvánvaló, hogy csakis csomagkapcsolt hálózatok képesek hatékonyan támogatni mind a karakterisztikailag erősen ingadozó sávszélességű adatfolyam igényeket, mind pedig a beszédátvitelt. Ezen az irányvonalon el is juthatunk a költséghatékonysághoz, hiszen a csomagkapcsolt hálózatok kiaknázzák az egyes folyamatok multiplexálásával elérhető nyereséget, mely révén megszüntethető a hagyományos POTS hálózatok többszörös túlméretezettsége. Remélhetőleg a költséghatékonyságból nemcsak a szolgáltatók, hanem az előfizetők is profitálnak a szolgáltatások árszínvonalának csökkenésével.

A mai Internet másik nyilvánvaló előnye az árpolitikájából adódik. Jelenleg az Internet szolgáltatók úgynevezett egységes (flat) árrendszert használnak, melynek alapja nem a kommunikációs távolság, hanem az átalánydíj, esetleg forgalom alapú számlázás. Ezzel szemben a telefonhálózaton hierarchikus számlázási struktúrát használnak, melyben a távolsági hívások jelentősen drágábbak a helyi hívásokhoz viszonyítva. Ebből az anomáliából adódóan a távolsági hívások Interneten keresztüli lebonyolítása jelentős költségmegtakarítást eredményez, persze csak ha a minőség elfogadható. Pontosan ez az a tényező, amiért a mai Internetes beszédátvitel még sokkal olcsóbb, mint a hagyományos áramkörkapcsolt beszédkommunikáció. Az előrejelzések előre vetítik, hogy a minőségi paraméterek javulásával (nyilván az Internetet érinti) az árak is kiegyenlítődnének [3.5.5].

Másrészről a költségtakarékosság az Internetes telefonia szoftver alapú fejlesztéséből is profitál, hiszen így a beruházási költségek relatív olcsón tarthatóak. Továbbmenve, a szoftver alapú fejlesztések további előnyeként értelmezhető, hogy egyszerűen integrálhatóak más szoftveres alkalmazásokkal, illetőleg viszonylag egyszerűen tovább bővíthetők. Ilyen kibővített alkalmazásokra mutatnak példát a hangos levelek, Web alapú hívasközpontok, a különböző felületmegosztó programok stb...

Azonban a szembetűnő előnyök ellenére is az IP telefonia széleskörű elterjedését mindeztáig hátráltatja az a tény, hogy az IP alapú hálózatokban nincsenek szolgáltatás-minőséget biztosító hordozó szolgálatok; vagyis hiányzik az úgynevezett minőség vagy QoS biztosítás. A minőségbiztosítási problémákon felül még olyan technikai kérdésekkel is megoldásra várnak, mint a biztonság, számlázás vagy felhasználói mobilitás mint extra tényező. A betárcsázásos hálózati hozzáférés sem ideális az IP-s telefoniának, hiszen nem biztosít állandó elérhetőséget. Ezen probléma azonban a kábeltévés vagy a különböző digitális előfizetői hurkokkal (xDSL) megoldódni látszik, hiszen ezek állandó Internet kapcsolatot biztosítanak.

3.5.3. Szabványok és architektúrák

Ha IP fölötti telefoniáról beszélünk, akkor napjaink egyik domináns szereplője az ITU⁵ által elfogadott H.323 keret-ajánlás, amely különböző más ITU ajánlások

⁵ International Telecommunication Union

összefogásával határozza meg, hogy milyen rendszer alkotóelemekből, milyen vezérlési üzenetekkel és milyen folyamatokkal építhető ki multimédia kommunikáció nem garantált minőségű csomagkapcsolt hálózatokon – mint pl. az Internet. Másrészt persze az IETF⁶, az Internet specifikáló szervezete is kidolgozta ajánlását multimédia kommunikáció kiépítésére és lebontására. Ezt az ajánlást kapcsolat kezdeményező protokollnak hívhatjuk, eredeti nevén pedig Session Initiation Protocol vagy SIP. A SIP specifikációhoz szorosan kapcsolódva jelenik meg a hordozóvezérlési információkat kezelő kapcsolat leíró protokoll (Session Description Protocol) [3.5.9], de többnyire SIP-ként globálisan hivatkozzák az IETF-es specifikációt. SIP is mint a H.323 helymeghatározási és hívás menedzsment szolgáltatást nyújt a végfelhasználók részére. Mindkét ajánlás elsődlegesen az IETF specifikálta valós-idejű átviteli protokollt (real-time protocol – RTP) [3.5.6] használja a multimédia adatfolyamok továbbítására. Az Internet szabad szelleméből adódóan egyik ajánlás sem köti magát az RTP-hez, jóllehet használata javasolt, hanem támogatnak bármely szállító rétegbeli protokollt mint TCP vagy UDP. [3.5.1], [3.5.2]

H.323 protokollok

Az áttekintés kedvéért az alábbi felsorolás tartalmazza az ITU H.32x ajánlóssorozat elemeit. Az egyes ajánlások a különböző hálózati infrastruktúra feletti multimédia szolgáltatás nyújtást definiálják:

Ajánlás	IP	Rövid leírás
H.320	-	Keskenysávú, digitális ISDN hálózatokra
H.321	-	Szélessávú, digitális ISDN és ATM hálózatokra
H.322	-	Sávszélesség biztosítással rendelkező csomagkapcsolt hálózatokra
H.323	X	Nem minőségbiztosított csomagkapcsolt hálózatokra
H.324	-	Analóg telefonhálózatokra (POTS)

A fenti ajánlóssorozatból is látszik, hogy a H.323-at az Internetes best-effort jellegű, vagyis minőségbiztosítás nélküli, szolgálta tervezték eredetileg. [3.5.4]

A H.323, mint a bevezetőben is említettük több szabványosított protokoll együttműködését definiálja, ezért keret (esernyő) ajánlásnak is nevezik (lásd: 3.5.1 ábra). Jelentősebb protokoll ajánlások a H.323 általános funkcionális leíráson belül a következők:

H.225.0 CC – hívás- és erőforrás-vezérlési protokoll.

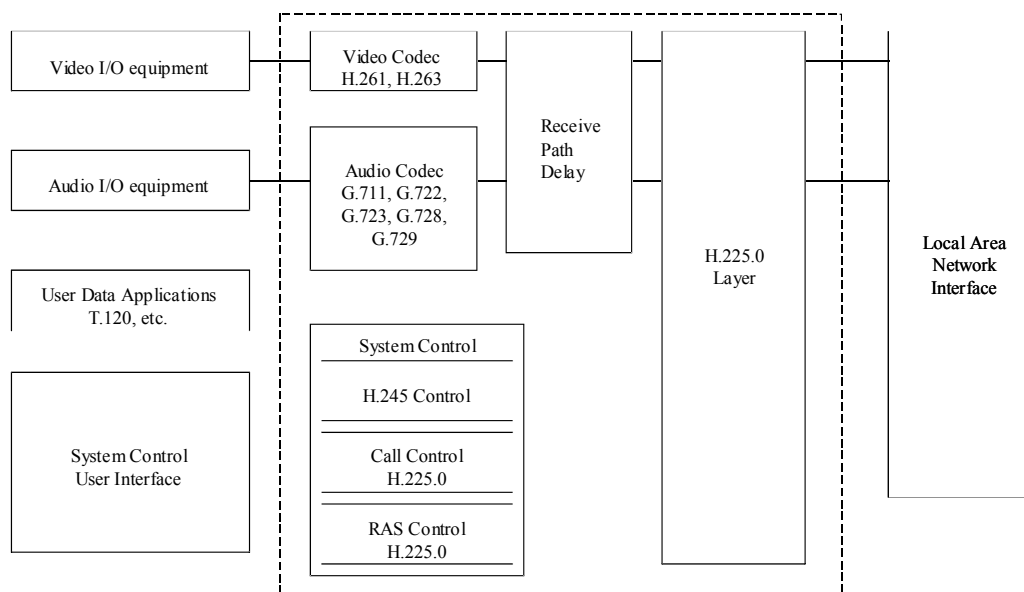
⁶ Internet Engineering Task Force

H.225.0 RAS – Ez a protokoll definiálja a regisztrációs, hívásengedélyezési és státusz jelzések formátumát. (RAS - registration, admission and status)

H.245 – hordozóvezérlési funkciókat tartalmaz; a H.323-as terminálok ezen jelzés segítségével nyitják meg a hang és videó kommunikációs csatornáikat.

T.120 – rögzíti az egyéb adatfolyamok csomagolási és küldési mechanizmusához tartozó formátumokat (pl.: adatkonferencia).

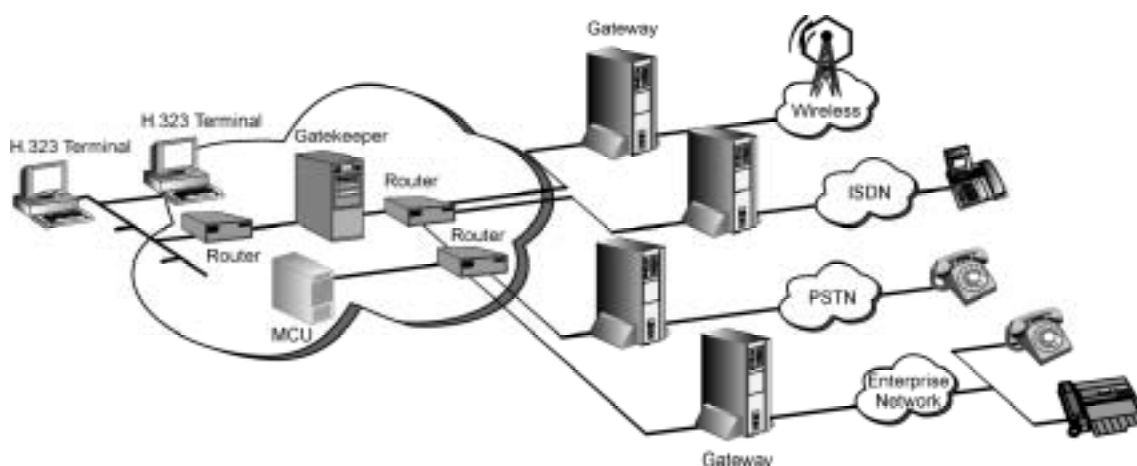
H.450.x – többletszolgáltatások meghatározása



3.5.1 ábra. H.323 protokollcsalád

H.323 összetevők

A H.323-as megvalósítások tipikusan négy logikai összetevőt különböztetnek meg, ezek név szerint: végberendezések (terminals), átjárók (gateways), zónavezérlők (gatekeepers) és többpont vezérlőegységek (multipoint control units – MCU) (lásd: 3.5.2 ábra). A **végberendezések** jelentik a H.323-as rendszerek végberendezéseit; beleértve mind az adat, mind pedig a jelzési kommunikációt. A beszéd (hang) kommunikációs képességet kötelezően támogatniuk kell, míg videó vagy más multimédia támogatás opcionális. A H.323-as rendszerben az **átjárók** biztosítják a különböző hálózatok közötti adat és jelzési információk átalakítását; a különböző kódolási rendszerek közötti átalakítást. Az átjárók nem kötelező elemei a rendszernek. A **zónavezérlők**- szintén opcionális elemként- irányíthatják a H.323-as zóna működését. Feladatuk a megbízható kommunikáció biztosítása. A



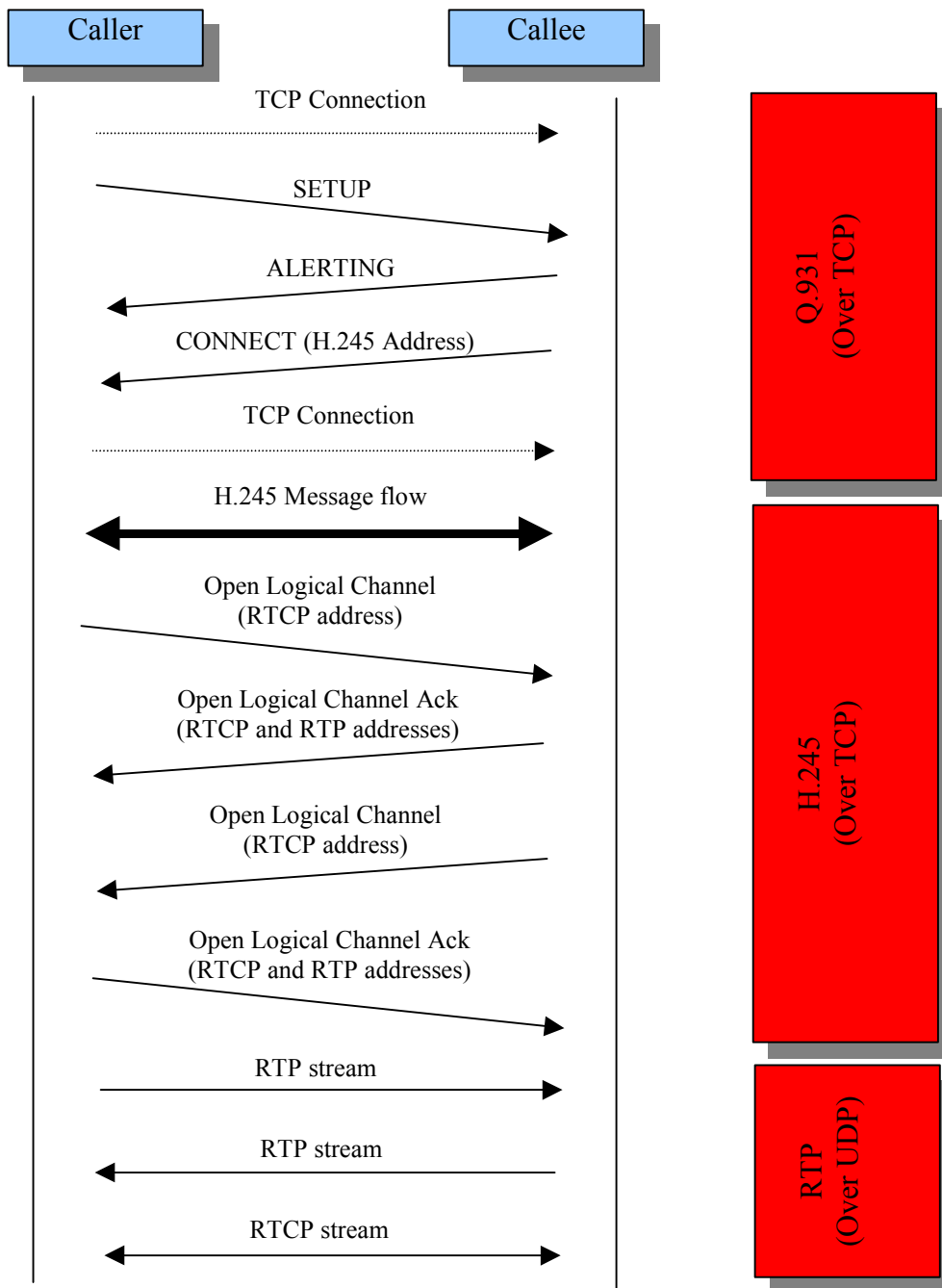
zónavezérlőkre úgy tekinthetünk, mint a H.323-as rendszer központosított fő logikai egységeire. Ha jelen vannak a rendszerben, akkor a zónához tartozó összes H.323-as eszköznek regisztrálnia kell magát a vezérlőben. Ezért cserébe kiterjedt cím translációt nyújt; hívás engedélyezést és hozzáférés vezérlését végez a végberendezések számára; sávszélesség biztosítást és útvonal-választási támogatást nyújt valamint felhasználói mobilitást és egyéb felügyeleti funkciók ellátását is végrehajtja [3.5.3]. Jegyezzük meg, hogy annak ellenére hogy a zónavezérlő nem kötelező eleme a H.323-as hálózatnak nem csak „kényelmi” szolgáltatásokat nyújt hanem általános működési irányelvek meghatározását végzi, ezért a legkisebb hálózatoktól eltekintve szinte kötelező hálózati elemnek is tekinthetjük. A *konferencia egységek* (MCU) feladata a terminálok közötti konferenciák támogatása. Ezen egység is nélkülözhető a hálózathoz egyszerű konferenciahívásoknál, azonban a később részletezendő többletfunkciókhoz használata nélkülözhetetlen. Az MCU-k két alapvető építőelemből állnak, i) a kötelező vezérlőből (multipoint controller – MC) és a ii) tetszőlegesen választott számú konferencia processzorból (multipoint processors – MP). A vezérlő feladata a konferencia hívások felépítése, míg a konferencia processzorok az adatfolyamokon hajtják végre a szükséges összeadást, keverést, kapcsolást és egyéb feldolgozást igénylő műveleteket. Az MCU használata szükség a hatékonyság növelése érdekében, ha különböző adatformátumok szerinti konverziók kell végrehajtani, különböző konferenciaszolgálatok igénybevételéhez mint például elnöki vezérlés vagy egy képen több helyszín megjelenítése.

3.5.2 ábra. H.323 összetevők

H.323 hívásfelépítés

A H.323-as protokoll legegyszerűbb hívás-felépítési módusa is megmutatja, hogy milyen erős kapcsolódás van a protokollcsalád különböző, akár önállóan is tekinthető protokolljai között. Ez a H.323 protokollcsaládot meglehetősen komplexé és monolitikussá teszi. A hívásfelépítés illusztrációja a 3.5.3-as ábrán látható; mely két H.323-as terminál kapcsolatfelépítését mutatja.

Megjegyzendő, hogy a kapcsolat-felépítési idő, amely a hasznos adatfolyam megjelenéséig tart (RTP stream), számottevő; esetünkben a körbefordulási idő (round-trip time) többszöröse. Pontosan a fenti hátrány kiküszöbölésére, a H.323-as szabvány 2-es verziójában vezették be az ún. gyors kapcsolatkiépítést (Fast Connect), mely esetében a hasznos kommunikáció rögtön a CONNECT üzenet után megindulhat. Természetesen ezt csak úgy valósíthatták meg, hogy a képességek kicserélése, valamint a logikai csatorna megnyitásának javaslata is belekerült a SETUP ill. a CONNECT üzenetekbe. A H.323 2-es verziója a fent említett javításon túl további javításokat is tartalmazott, valamint többlet szolgáltatásokat definiált. A H.323 protokollszabvány jelenleg a 4-es verziónál tart, melyet 2000. novemberében fogadtak el.



3.5.3 ábra. H.323-as direkt kapcsolatfelépítés

Session Initiation Protocol (SIP)

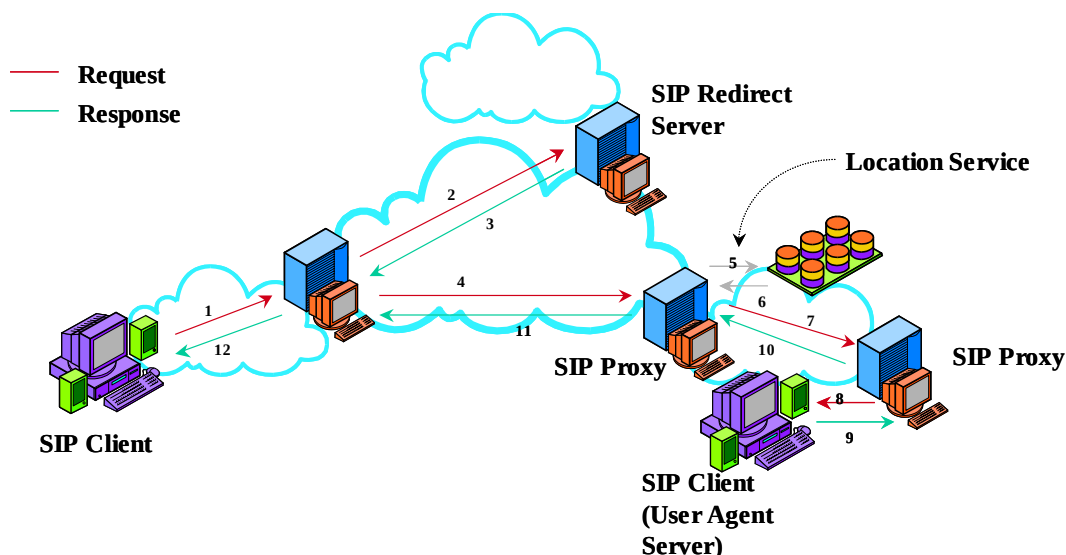
A SIP, melyet az IETF specifikált, egy egyszerű, szöveg alapú kérés-felelet protokoll. A SIP esetében a ügyfelek hívás kezdeményezést a kiszolgálók pedig a hívások fogadását végzik (ügyfél-kiszolgáló architektúra). A leggyakoribb megvalósítások mind az ügyfél, mind pedig a kiszolgáló funkciókat tartalmazzák. Maga a kliens-szerver kommunikáció tetszőleges alsó rétegbeli protokoll felett létrejöhet, ezáltal használva akár TCP, STCP vagy UDP protokollokat. Maga a

kapcsolatfelépítés közbelső elemeken keresztül történik (lásd alább), azonban a végpont-végpont kapcsolat kiépülte után az adatfolyamok közvetlenül a végpontok közvetlenül kommunikálnak egymással. A SIP protokoll nem tekinti feladatának, hogy akár erőforrás menedzseléssel vagy adatküldéssel és fogadással foglalkozzon; netán pont-többpont kapcsolatok adatfolyamát vezérelje. Ezeket az moduláris felépítése miatt más IETF specifikációra bízta, míg a SIP kizárólag multimédia kapcsolat vezérléssel foglalkozik. Ezek közé tartozik a hívástovábbítás (call forwarding), hívó és hívott címeinek kezelése, felhasználói mobilitás támogatása, végpont képességek egyeztetése, hitelesség ellenőrzés és dinamikus konferencia vezérlés. A konferencia vezérlés csak hívásszintű funkciókat foglal magába mint csatlakozás kiépítése, képességek egyeztetése, kilépés és bontás de nem érinti a konferencia adatküldést. [3.5.7]

SIP összetevők:

Hasonlóan a H.323-as hálózati felépítéshez, a SIP hálózat is különböző alap építőelemekből tevődik össze. A felhasználói ügyfél (user agent client – UAC) az a logikai egység, amely SIP kéréseket állít össze és küld. Ez a szerep egyetlen tranzakcióra vonatkozik. A felhasználói kiszolgáló ügynök (user agent server – UAS) az a logikai egység, mely fogadja és válaszol a SIP kérésekre. A SIP kérésekre tipikusan három válasz érkezik: elfogadás, visszautasítás vagy átirányítás. Ez a szerep szintén egy tranzakció idejére korlátozódik. Egy felhasználói ügynök (user agent – UA) tipikusan magába foglalja mind az UAC-t, mind pedig az UAS-t. A SIP végpont mindig tartalmaz egy felhasználói ügynököt (UA), és ezenfelül támogatnia kell valós idejű kétirányú kommunikációt más SIP terminállal. SIP proxy-k tipikus közbelső elemei a SIP hálózatnak, melyek feladata, hogy kéréseket és válaszokat továbbítsanak. Két fő típusát különböztethetjük meg: állapotinformációt tároló és nem tároló egységeket. Az állapotinformációt nem tároló proxy-k kizárólag üzenetek továbbítását végzik, míg a másik típus az üzenetek feldolgozását is elvégezheti. Egy másik elem a SIP hálózatban az átirányító kiszolgáló (redirect server), mely SIP kérések célcíme alapján meghatároz néhány új (lehet nulla is) lehetséges címet és ezeket visszaküldi az ügyfél számára. Ezek az átirányító kiszolgálók a proxy kiszolgálókkal ellentétben kizárólag SIP kérésekre képesek válaszolni, önmaguk nem kezdeményeznek kéréseket. A SIP hálózatokban helymeghatározás támogatására alkalmazhatnak helymeghatározó szolgáltatást (location service) is. Ez a logikai

egység nyújt információt mind a proxy, mind pedig az átirányító kiszolgálóknak a végfelhasználó helyzetét illetően. Tipikus megvalósításában egybeépítik valamely más SIP szolgálattal (proxy vagy átirányító kiszolgáló). Egy lehetséges SIP hálózati struktúrát mutat be a 3.5.4-es ábra. [3.5.7]



3.5.4-ábra. SIP összetevők és kliens felderítés

SIP hívásfelépítés

SIP kliensek hat fajta kérdéstípust küldhetnek a szerverek felé [3.5.7]:

INVITE	Híváskezdeményezési üzenet, melyben a hívó fél közli preferenciáit: kommunikációs médium, port számok stb...
ACK	Híváskezdeményezés megerősítése (elfogadása); szintén tartalmazza a fogadó fél preferenciáit (lásd INVITE)
OPTIONS	Kiszolgáló által nyújtott szolgáltatások lekérdezése
REGISTER	A felhasználói elérhetőség bejegyzése szerverekbe
BYE	Hívásbontás
CANCEL	Egy sikertelen felhasználó felderítés lezárása

A fenti kérésekre szintén hat fajta válaszüzenet érkezik [3.5.7]:

1xx	Információ közlésre
2xx	Kérés elfogadva (successful)
3xx	Átirányítás (redirection)
4xx	Kérés elutasítva (request failure)
5xx	Szerver hiba (server failure)
6xx	Globális hiba (global failure)

A fenti ábra (3.5.4) tartalmaz egy hívás-felépítési folyamatot is. A folyamatban lejátszódó protokollüzenetek a következők:

Legelőször a SIP kliens az (1-INVITE) üzenettel megkezdí a kapcsolatfelépítést. Az első közbenső elem egy SIP proxy szerver, mely a kezdeményező SIP ügyfél szerepében helymeghatározási szerepet végez. Ez a proxy kiszolgáló egy új (2-INVITE) üzenetet küld az átirányító kiszolgálóhoz. Az átirányító kiszolgáló válaszában tudatja a kérdező proxyval a felhasználó új helyzetét (3-3xx). Az átirányításnak megfelelően az első proxy kiszolgáló egy új (4-INVITE) üzenettel próbálkozik a frissen kapott cél felé. Vegyük észre, hogy az átirányítást a proxy kiszolgáló kezelte le és nem a kezdeményező SIP ügyfél. A címzésben következő proxy kiszolgáló egy helymeghatározó kiszolgálóhoz fordul (5-ös és 6-os üzenet), majd az adatbázisból nyert információ alapján folytatja a hívásfelépítést (7-INVITE). Mielőtt a cél klienst elérnénk még egy hasonló proxy szerveren keresztül megy üzenetünk (8-INVITE), majd elérkezünk a kívánt klienshez. A hívott fél ezután egy pozitív nyugtát (ACK) küld vissza a proxy útvonalon a kezdeményező klienshez (9, 10, 11 és 12-es üzenetek). Az ACK üzenet megérkezésével tekinthetjük a hívást kiépültnek.

Valós idejű protokoll (Real-Time Protocol - RTP)

Az RTP protokoll célja, hogy valós idejű információt szállítson mint pl. beszéd- és képtávitel azáltal, hogy kiegészíti az UDP/IP fejlécet. Példaként az RTP hasznos adat (payload) azonosítást nyújt, hogy az alkalmazások meghatározhassák a vett média típusát. Az RTP fejléc tartalmaz sorszámot és időbélyeget, hogy a vevő oldal visszaállíthassa a csomagok sorrendjét vagy észlelhesse a csomagvesztést; illetőleg a kép-hang szinkront vagy egyéb multimédia folyamatok szinkronizálását elvégezhesse. Az RTP továbbá támogat pont-többpont kapcsolati kommunikációt is. [3.5.6]

3.5.4. A H.323 és a SIP összehasonlítása

Általánosságban elmondható, hogy mindkét protokoll gyorsan alkalmazkodik a felmerülő igényekhez, és hogy a két protokoll közötti kezdeti teljesítménybeli különbségek mára már elenyészők. Jelenleg is különböző azonban az üzenetcsomagok kódolásának megvalósítása: míg a H.323 az ASN.1-es és

táblázatos bináris üzenatkódolást alkalmaz, addig a SIP szöveges HTTP alapú üzenatkódolást használ. Természetesen mindkét protokoll profitálni próbál az őt támogató és körülvevő környezetből. A H.323 erőssége a távközlési világ szabványosítási múltjában rejlik: következetes és precíz ITU-T szabványok, együttműködési tesztek és körültekintően kidolgozott átjárhatóság a hagyományos telefonhálózat felé (POTS és PSTN). Másrészt a SIP előnyére válhat az a flexibilis és gyors specifikációs lehetőség, amely az Internetet magát jellemzi; valamint az Internetes társadalom bizonyos mértékű „ellenállása” külső szabványokkal szemben.

3.5.5. Az IP telefonia jövője

Alapvetően elmondható, hogy az Internet telefonia bármilyen széleskörű kereskedelmi elterjedésének egyelőre gátat szabnak a felmerülő együttműködési problémák (azonos protokollok különböző gyártói között is); a hiányzó minőségi, számlázási, biztonsági, jelzés átalakítási (SS7), stb... támogatások. Az is elmondható, hogy a jelenlegi IP-s multimédia piac beszédátvitelre korlátozott, hiszen sok a modemes, alacsony sebességű és betárcsázásos hozzáférés. Másrészt a technológia aránylag új; időnek kell eltelnie, amíg megbízható, együttműködő és minőségi termékek jelennek meg a piacon. A jövő szempontjából talán az ár a legfontosabb: IP fölött jelenleg olcsóbban lehet telefon szolgáltatást nyújtani, miáltal még a hagyományos – tipikusan nagy – szolgáltatókat is újszerű versenyre lehet készíteni. Természetesen az átállás a PSTN rendszerről az IP-s telefoniára időt vesz igénybe; abban azonban biztosak lehetünk, hogy a jövő technikai fejlesztéseitől nemcsak beszéd, hanem videó és egyéb multimédia kommunikációt várnak. [3.5.8]

3.5.6. Összefoglaló

Összefoglalásként azt mondhatjuk, hogy az IP-s telefonia jelen van, bár széleskörű elterjedését még néhány hiányossága gátolja. Két jelentős, versengő szabvány küzd a piacért: az egyik az ITU-T által szabványosított H.323 protokoll míg a másik az IETF-en szabványosított SIP protokoll. A piaci helyzetet tekintve a H.323 protokoll előnyösebb pozíciókkal bír, azonban a verseny még korántsem eldőlt. A 3. generációs mobil rendszerek bevezetése még elősegítheti a SIP szélesebb körű elterjedését is.

Mindazonáltal, talán az egyik legfontosabb kérdése az IP-s telefoniának a minőségi szolgáltatás nyújtásának kérdése. Hiszen míg a hagyományos telefonrendszerekben ezen kérdés hatékonyan kezelt, addig a mai Interneten ezek a problémák még korántsem megoldottak. Erőfeszítéseket tesz az Internetes társadalom ezen irányban, melynek eredményei remélhetőleg a közeljövőben felvirágozhatják az IP-s telefoniát.

A másik számottevő probléma az együttműködési kérdések megoldatlansága. Ez a kérdés túl azon, hogy heterogén IP-s telefonia hálózatokat és a hagyományos hálózattal való együttműködést érinti, még azonos szabványok különböző gyártói között sem teljesen megoldott. Különböző szervezetek léteznek, melyek a fenti problémát próbálják orvosolni, ezek közül a teljesség igénye nélkül néhány megemlítve: ETSI's TIPHON, iNOW!, IMTC stb...

A biztonság mint harmadik megoldandó problémakör helyet követel magának. Mindkét protokoll ajánl megoldásokat, azonban ezek még korántsem tekinthetők kielégítőnek. A biztonság problémakörét a jövőben mindkét protokollnak figyelembe kell vennie.

Összességében elmondhatjuk, hogy az IP-s telefoniát tekinthetjük akár az első lépésként a valós idejű, integrált multimédia szolgáltatások felé.

Irodalomjegyzék

- [3.5.1] Bo Li, Mounir Hamdi, Dongyi Jiang, Xi-Ren Cao, Y. Thomas Hou "QoS-Enabled Voice Support in the Next-Generation Internet: Issues, Existing Approaches and Challenges" IEEE Communications Magazine, April 2000
- [3.5.2] D. Bergmark, S. Keshav "Building Blocks for IP Telephony" IEEE Communications Magazine, April 2000
- [3.5.3] Trillium, "H.323 Tutorial," 1999, 20 pages, <http://www.webproforum.com/h323/>
- [3.5.4] Lasse Huovinen, Shuanghong Niu, "IP Telephony", <http://www.tml.hut.fi/Opinnot/Tik-110.551/1999/papers/04IPTelephony/voip.html>
- [3.5.5] Communications Industry Researchers, Inc, <http://www.cir-inc.com/>
- [3.5.6] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson "RTP: A Transport Protocol for Real-Time Applications", Internet Engineering Task Force, Internet Standard, RFC 1889, January, 1996
- [3.5.7] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg "SIP: Session Initiation Protocol", Internet Engineering Task Force, Internet Standard, RFC 2543, March, 1999
- [3.5.8] H.323 forum at <http://www.h323forum.org>

[3.5.9] M. Handley, V. Jacobson “SDP: Session Description Protocol”, Internet Engineering Task Force, Internet Standard, RFC 2327, April, 1998

3.6. ATM, IP ATM felett, ATM-LAN és MPLS

Szerző: dr. Cinkler Tibor

Lektor: dr. Henk Tamás

Az ATM kifejtése: Asynchronous Transfer Mode, vagy magyarul: aszinkron átviteli mód.

Az ISDN technika kidolgozásánál az volt a cél, hogy a hagyományos távbeszélő hálózatok legdrágább és legkevésbé kihasznált részét, az előfizetői vonalt újrahasznosítsák. Ebből adódott az ISDN hátránya is - keskenysávú lett, ezért sokszor N-ISDN azaz keskenysávú (Narrow-band) néven emlegetjük. (A határt a keskeny és szélessávú vezetékes hozzáférésnél 2Mbit/s-ra teszik.) Az ISDN előnye viszont az volt, hogy viszonylag olcsón a távbeszélő szolgálat mellett adatátvitelre, videokonferenciára, jobb minőségű és gyorsabb fax küldésre illetve értéknövelt szolgáltatások nyújtására is lehetőség nyílt.

Ahhoz, hogy az egyre növekvő igényeket kiszolgálhassák, az ITU-T kezdett dolgozni a B-ISDN azaz szélessávú (Broad-band) ISDN ajánlásokon. Az elvárás az volt, hogy olyan szélessávú integrált szolgáltatású digitális hálózatot alakítsanak ki, ahol a pont-pont kapcsolatok mellett pont - több-pont illetve több-pont - több-pont összeköttetéseket is lehet létrehozni, illetve támogatja mind a kapcsolt mind az állandó összeköttetéseket, úgy mint az áramkör és csomagkapcsolt szolgáltatásokat is, illetve mind az egyirányú mind a kétirányú (szimmetrikus vagy asszimmetrikus) összeköttetéseket is.

A B-ISDN-hez az ITU-T definiált referenciamodellt, az ISDN-éhez hasonló jelzésrendszert, és az ATM-et találta legmegfelelőbbnek a B-ISDN követelményrendszerének kielégítésére.

3.6.1. Érvék az ATM mellett

A forgalom az adatátviteli és távközlő hálózatokban mind térben mind időben eloszlik. Nappal a munkahelyeken nagy a forgalom, munka után és hétvégén a lakóközrzetekben. Ha a különféle forgalmakat összefogjuk és egy hálózaton szállítjuk, akkor azok hatékonyabban kezelhetők. Így nem kell az adat, távbeszélő és

műsorszétoztó hálózatainkat mind csúcsforgalomra méretezni, hanem a térbeli és időbeli eloszlás miatt az erőforrásokat megosztjuk.

A korábbiakban alapvetően két hálózattípussal találkoztunk. Az áramkörkapcsolt hálózatok a távbeszélő hálózatokban bevált, elterjedt megoldás, míg a csomagkapcsolás a számítógép illetve adatátviteli hálózatokban használatos.

Áramkörkapcsolás esetén igény szerint létrehozunk egy adott kapacitású összeköttetést (áramkört) melyet csak az adott felek használnak. Miután véget ért az információátvitel az áramkört bontjuk, és az addig foglalt erőforrásokat átengedjük más összeköttetések számára. Ez lehetővé teszi a minőségi garanciák biztosítását (QoS: Quality of Service, pl. kis késleltetés, kis adatvesztés), feltéve., hogy az összeköttetés létrehozása nem ütközött erőforráskapacitás korlátba. Másrészt viszont, ha a rögzített sávszélességű lefoglalt csatornán csomós ("bőrsztös") adatot küldünk az az idő jelentős hányadában kihasználatlan lesz, sőt lehet, hogy az átvitel kapacitás csak az egyik irányban lesz kihasználva.

Csomagkapcsolás esetén a helyzet a fordított, csak akkor küldünk csomagot, ha összegyűlt "elegendő mondanivalónk". Ezt várakozás nélkül tehetjük, de hogy biztosítsuk csomagunk célbajutását azt "megcímezzük", azaz hozzáteszünk egy fejlécszt. Így jobb erőforráskihasználást tudunk elérni ritkán de sokat küldő források esetén, ám minőségi garanciákat nem biztosít a hálózat.

Minőséget csak akkor tudunk biztosítani, ha lefoglaljuk a szükséges erőforrásokat ÉS ellenőrizzük, hogy forgalomforrásaink betartják-e e foglalást. Ezért az ATM ötvözi a fenti technikákat, és úgynevezett virtuális áramkörkapcsolást használ míg az információt kis, rögzített méretű csomagokba, cellákba tördeli.

Így lehetőség nyílik a determinisztikus multiplexelés kiváltására statisztikus multiplexeléssel. Ma az ATM az egyetlen technika mely QoS garanciák mellett jó erőforráskihasználást biztosít, mind a hozzáférői, mind a gerinchálózatban a 2 Mbit/s - 2.5 Gbit/s bitsebesség tartományban.

Az ATM legfontosabb előnyét, az együttes QoS biztosítást és hatékony erőforráskihasználást a rendszer bonyolultságával fizettük meg (lásd 4.2 fejezet). Az IP hálózatokban jelenleg az egyik legfontosabb törekvés a minőség biztosítás, elsősorban a valós idejű alkalmazások miatt, pl. beszédátvitel IP hálózat felett (VoIP:

Voice over Internet Protocol). Ezt csak úgy érhetik el, ha növelik a rendszer bonyolultságát a DiffServ illetve IntServ megoldások bevezetésével.

3.6.2. Az ATM jellemzői

Az ATM tömören az alábbiakkal jellemezhető:

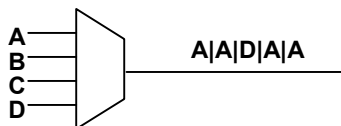
- Az ATM a B-ISDN (szélessávú ISDN) ajánlott átviteli technológiája.
- Cellakapcsolt, az adatok rögzített méretű, kis csomagokban továbbítódnak.
- Virtuális áramkör kapcsolást használ, ezért útvonal-választást csak egyszer kell végezni, utána csak a cellák egyszerű továbbítása szükséges.
- A kapcsolat hardverből történik, sem folyamatszabályozás, sem javítás nincs a csomópontokban, csak a hálózat peremén.
- Statisztikus multiplexelés révén tetszőleges sáv szélesség foglalható illetve használható az adott hozzáférői sebesség tartományban.
- Tetszőleges minőséget tud biztosítani jó erőforrás kihasználtsága mellett.

Tekintsük át az ATM-et megalapozó műszaki fogalmakat és megoldásokat, hogy fény derüljön a fenti jellemzőkre!

3.6.3. Műszaki alapok

Mitől *aszinkron* átviteli mód az ATM?

Multiplexeléskor különböző sebességű csatornák is összefoghatók, és a kimenten tetszőleges szabad időrést használhatnak az adataegységek, azaz nincsenek egy keretszervezéshez szinkronizálva. De ahhoz, hogy vételi oldalon



szét tudjuk válogatni ezeket az egységeket mindegyiket felcímkézzük.

3.6.1. ábra. aszinkron nyalábolás

Hogyan épül fel az adatátvitel egysége a cella?

VPI		
VPI	VCI	
VCI		
VCI	PT	CLP
HEC		

48 oktet rakomány

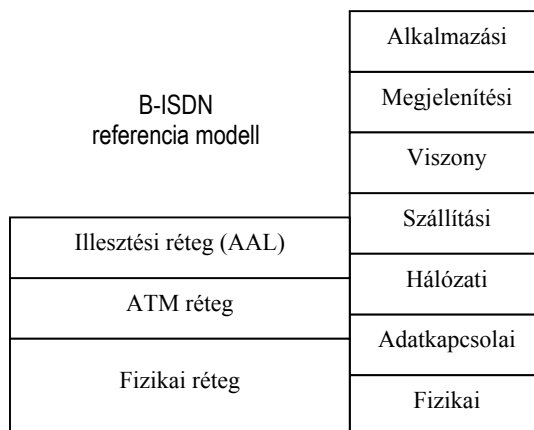
3.6.2. ábra. Az ATM cella felépítése

Az ábrán látható egy cella felépítése egy hálózati interfészen (NNI: Network to Network Interface). Az egyes sorok okteteket jelentenek, balról jobbra, majd fentről soronként lefelé haladva olvassuk ki tartalmát. Az első 12 bit a VPI (Virtual Path Identifier: virtuális út azonosító), mely azt jelzi, hogy az adott cella mely csoporthoz (VP-hez: Virtual Path) tartozik. A következő 16 bit a VCI mező (Virtual Channel Identifier: virtuális csatorna azonosító), mely a VPI értékkel jelölt csoporton belül azonosítja az összeköttetést. A három bites PT (Payload Type) mező a rakomány azonosításában illetve torlódásvezérlésben játszik szerepet, míg a CLP (Cell Loss Priority) mező jelöli, hogy az adott cella eldobható-e, azaz magas vagy alacsony prioritással dobható el ha a hálózatban torlódás keletkezik. A HEC (Header Error Control) mező funkciója kettős: egyrészt fejrészhiba észlelésére és javítására, másrészt a cellahatárok észlelésére szolgál. Az 5 oktetes fejrészt 48 oktet hasznos adat követi, így a cella teljes mérete 53 oktet.

Amennyiben nem hálózati interfészt, hanem felhasználó és hálózat közti interfészt (UNI) vizsgálunk akkor a cella első 4 bitje a GFC (Generic Flow Control) mező lesz melyet forgalomszabályozási funkciók megvalósítására terveztek.

Hogyan illeszkedik az ATM az ISO OSI referenciamodelljéhez?

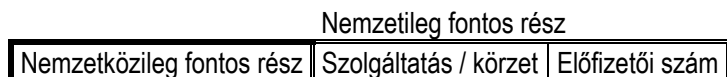
Az ATM az ITU-T által definiált B-ISDN referenciamodellnek felel meg, mely egy bonyolultabb háromdimenziós modell. Az ábrán csak egy dimenziót emeltünk ki. Az ATM-nél az ITU-T három réteget definiált csak. Látjuk, hogy az OSI referenciamodellhez képest az ATM fizikai rétege többet tud, és ez az elcsúszás a többi rétegnél is érzékelhető. Illesztési rétegből több féle van, legelterjedtebb az AAL5 (ATM Adaptation Layer 5) mely igen egyszerű, kevés funkciót valósít meg, elsődleges feladata a magasabbszintű adategységek tördelése az adásnál, illetve ezek újraegyesítése a vételnél. Az IP, FR illetve primér PDH fogalmata az illesztési réteg segítségével helyezzük az ATM hálózatra.



3.6.3 ábra. Az OSI és B-ISDN referenciamodellek összevetése

Milyen címezést használnak ATM hálózatokban?

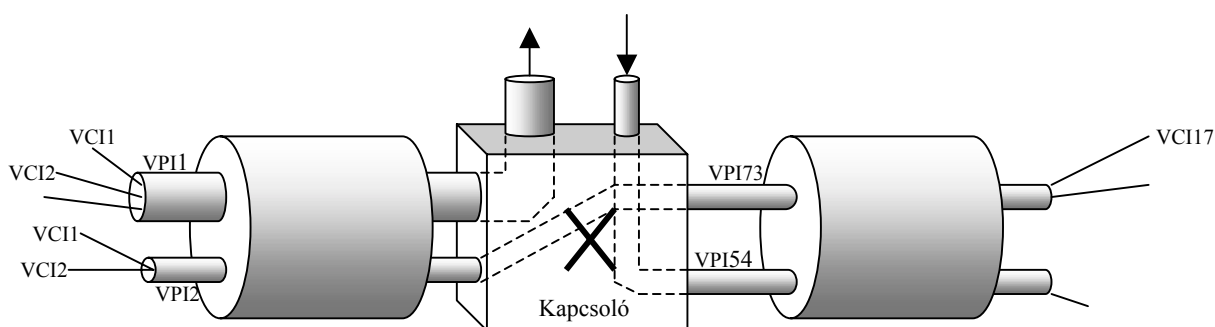
Nyilvános ATM hálózatokban a távbeszélő hálózatokban is használatos ITU-T E.164 ajánlást használják. A magánhálózatokban használatos címezés ettől eltér.



3.6.4 ábra. ATM címezés

Milyen "összeköttetések" vannak ATM hálózatokban?

Egy adatforgalmat szállító összeköttetés két végpont között egy "látszólagos áramkör" vagy tükörfordításban egy "virtuális csatorna összeköttetés" (VCC). Ez a VCC több szakaszon keresztül több VP-t (látszólagos utat) használva valósul meg.



3.6.5 ábra. VP és VC kapcsolás, illetve rendezés

Egy virtuális csatorna összeköttetés lehet állandó (PVC) vagy kapcsolt (SVC). Míg a PVC konfigurálással hálózat- vagy hálózati elem menedzsment rendszer révén hozható létre, az SVC-t az előfizető kezdeményezésére jelzésrendszer révén hozzák létre.

A pont - pont összeköttetések mellett az ATM támogatja a pont - több-pont összeköttetéseket is, sőt ATM-beli összeköttetések felett összeköttetés mentes alkalmazások is megvalósíthatók.

Forgalommenedzsment

Az ATM hálózatoknak kiváló forgalommenedzsmentje van. Mint említettük, az ATM egyetlen olyan megoldás, mely tetszőleges minőségi garanciák biztosítása mellett jó erőforráskihasználást valósít meg.

Az ATM forgalommenedzsment specifikációk *forgalmi* és *minőségi* igényeik alapján több csoportba sorolja a forrásokat. Ezek közül a 4.1 fejezet többet részletez, itt csak felsoroljuk:

- CBR: Constant Bit Rate:
 - o Állandó bitsebesség jellemzi, példaként a bérelt vonalat említhetjük.
- rt-VBR: Variable Bit Rate, valós idejű változó bitsebességű forrás.
- nrt-VBR: nem valós idejű VBR
- ABR: Available Bit Rate: rendelkezésre álló bitsebességű forrás, ahol az adási sebesség a hálózat forgalmi terheltségének függvényében visszacsatolással szabályozható
- UBR: Unspecified Bit Rate. Forrás, ahol nincsenek minőségi paraméterek megadva.
- GFR: Guaranteed Frame Rate: Olyan forrás, mely nagyobb méretű keretek továbbítását igényli. Itt a cellákra tördelt keretek összefüggését nyilvántartják.

A forgalmi és minőségi paraméterek alapján történik az erőforrásfoglalás és betartásukat a felhasználó, illetve a hálózat részéről felügyelik az összeköttetés teljes időtartama alatt.

3.6.4. Hogyan lehet ATM fölött számítógép-hálózatot kialakítani?

Ennek több módja van:

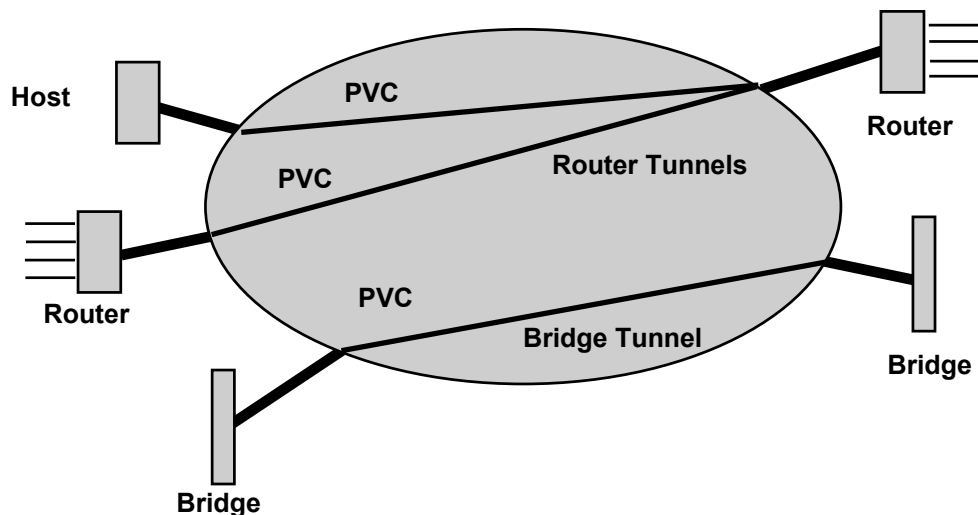
1. Több-protokollós beágyazás (multi-protokoll encapsulation) (IETF RFC 1483)
2. Hagyományos IP ATM felett (Classical IP over ATM, CLIP) (IETF RFC 1577)

3. LAN emuláció (LAN Emulation over ATM Version 1.0) (ATM Forum LANE 1.0)
4. ATM alkalmazásprogramozó interfész (ATM Application Programming Interface) (ATM API)
5. Hagyományos IP és címfeloldás ATM felett (Classical IP and ARP over ATM) (IETF RFC 2225)
6. Több protokoll ATM felett (Multi-Protocoll over ATM) (ATM Forum MPOA)
7. LAN emuláció 2.változata (ATM Forum LANE 2.0)
8. Többprotokollós címkekapcsolás (MultiProtocol Label Switching) (IETF MPLS)

E technikák csoportosíthatók az alapján, hogy mire van kiélezve. Míg a 3., 6. és 7. módszert elsősorban LAN keretek szállítására dolgozták ki, addig a 2, 5, és 8 módszereket elsősorban az IP csomagok közvetlen szállítására. Az 1. módszer mind LAN mind IP keretek szállítására alkalmas. Ennek megfelelően a LAN jellegű megoldások a közeghozzáférés függvényében kisebb távolságok áthidalására, míg a közvetlen IP csomagok szállítását célzó megoldások nagyobb távolságok áthidalására szolgálnak.

3.6.5. Több-protokollós beágyazás (IETF RFC 1483)

Ez az első megoldás, mely lehetővé tette az IP alapú kommunikációt ATM hálózatok felett. A megoldást az IETF dolgozta ki, alapjait az 1993 júliusában elfogadott RFC 1483 tartalmazza. A megnevezésben azért szerepel a *többprotokollós*, mert több különböző LAN keret és protokoll oszt egy PVC-t. Az egyetlen használt beágyazás az LLC (Link Layer Control) beágyazás. Az RFC 1483-ban rögzített megoldás csak PVC-eket, azaz állandó virtuális csatornákat használ. Így minden, a kommunikációban résztvenni kívánó félnek állandó csatornát alakítunk ki, függetlenül, hogy az szállít-e adatot vagy sem. Ezeket a PVC-eket alagutaknak (tunnel) nevezzük, hiszen ez csak egy biteket szállító cső. Egy hálózatban 2 munkaállomás noha tud közvetlenül kommunikálni, a teljes PVC-szövevény elkerülése érdekében jellemzően bridge vagy router segítségével kommunikál egymással. Egy alagút vagy hidak (bridgek), vagy útválasztók (routerek) összekötését szolgálja, de mindkettőt egyidőben nem.



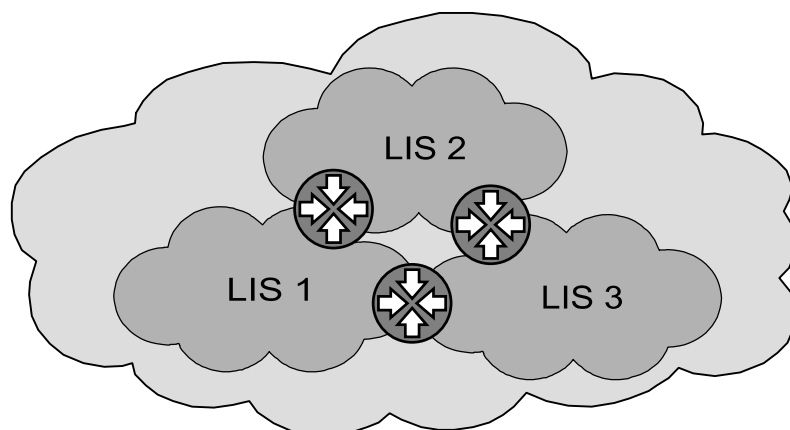
3.6.6 ábra. "Alagutak" a többprotokollós beágyazásnál.

E megoldás egyik hátránya, hogy az állandó VC-k (PVC-k) állandóan lefoglalják az erőforrásokat, beleértve a VPI/VCI címet is. Másrészt figyelmen kívül hagyja az ATM VC-kapcsoló (SVC) és útvonalválasztási képességeit.

3.6.6. Hagyományos IP ATM felett (IETF RFC 1577 és RFC 2225)

Ez már egy bonyolultabb, de egyben rugalmasabb megoldás is, mely a kapcsolt VC-k (SVC-k) előnyeit is képes kihasználni. Hátránya viszont, hogy csak az IP-t támogatja, egyéb hálózati protokollokat nem. Mint a nevéből is kiderül a TCP/IP csomagok ATM hálózaton való átküldését és a címmegfeleltetést tartalmazza. Alapjait az IETF (Internet Engineering Task Force) által specifikált RFC 1577 illetve 2225 rögzítik.

Az IP csomagok tördelését az adási oldalon illetve ezek újraegyesítését a vételi oldalon az AAL5 illesztési réteg végzi. E technika mind PVC mind SVC alapú összeköttetéseket támogat.



3.6.7 ábra. Logikai IP alhálózatok és kapcsolatok.

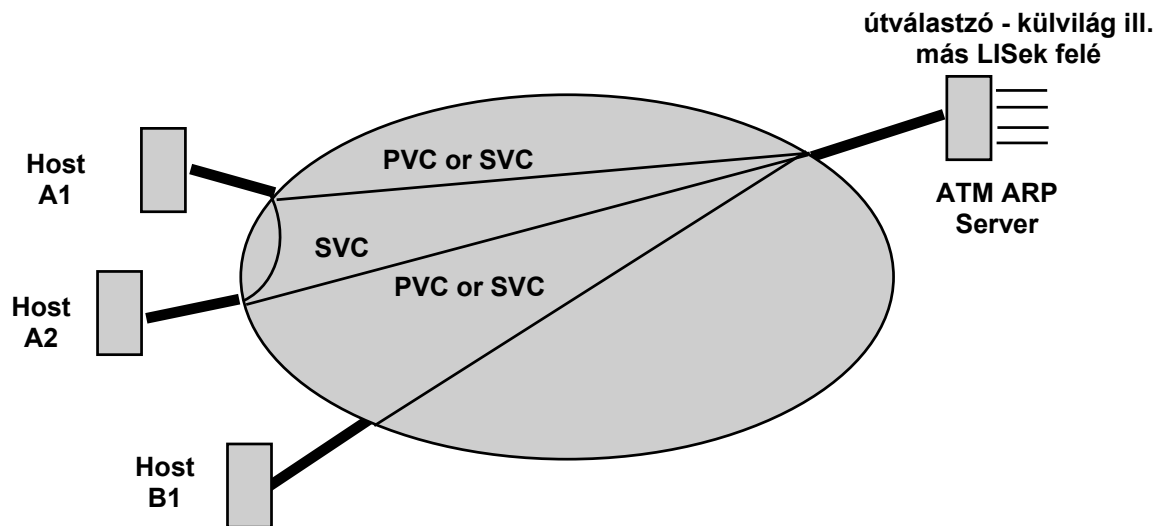
Azon gépeket melyek egymással közvetlenül kell kapcsolatot létesítsenek logikai IP alhálózatokba soroljuk (LIS: Logical IP subnet). Három ilyen LIS-t szemléltet a fenti ábra. Míg a LIS-en belüli munkaállomások képesek közvetlen ATM összeköttetést (SVC) kialakítani egymás között, addig a különböző alhálózatba tartozók csak több SVC-n keresztül útválasztók (routerek) segítségével érik el egymást.

A továbbiakban az egyes LIS-eken belüli kapcsolat kiépítését vizsgáljuk. Minden IP munkaállomás melyet szeretnénk hálózatba kötni ATM hálózat felett, kell rendelkezzen egy ATM címmel. Amennyiben PVC-t használunk (például ha a hálózat nem támogat jelzésrendszert), minden munkaállomásnak van saját címfeloldási táblázata, azaz minden IP címnek megfelel egy adott VPI/VCI címke. Amennyiben SVC-t használunk minden LIS-en belül kell legyen egy címfeloldó szerver (ARP: Address resolution Protocol).

Nézzük, hogyan épül ki egy IP folyam.

Először is a kliens kiépít egy összeköttetést az ARP szerver felé, mely vagy egy külön munkaállomáson, vagy gyakrabban egy ATM kapcsolón "fut". Az ARP szerver címe konfigurálható, vagy dinamikusan lekérhető. Ezt követően az ARP szerver egy inverz ARP kéréssel lekéri a csatlakozó munkallomás ATM címének megfelelő IP címet is. Így tartja karban az ARP szerver a címmegfeleltetési táblázatot. Ezt követően, ha például az A1 munkaállomás az A2 munkaállomást szeretné elérni annak IP címe alapján, akkor megnézi a saját ideiglenes címmegfeleltetési tárában, ha ott nem találja az ARP szervertől lekéri A2 ATM címet, majd egy ATM SVC-t épít ki A2 felé. Ezt adatátvitel követi, majd hosszabb

átvitelmentes időszak után az SVC bontódik. Ha rövidesen A1 ismét szeretné elérni A2-t, helyi címmegfeleltetési tárában ezúttal megtalálja annak ATM címét, ezzel is gyorsítva a kapcsolatkiépítést, illetve csökkentve az ARP szerver terhelését.



3.6.8 ábra. Hagyományos IP ATM felett: egy LIS.

Amennyiben A1 egy olyan munkaállomást (pl. B1) szeretne elérni mely nem ugyanabban a LIS-ben van, akkor ezt routeren keresztül teheti csak meg. Ábránkon az ARP szerver és útválasztó egybeesik, de ez a gyakorlatban nem feltétlenül valósul meg így.

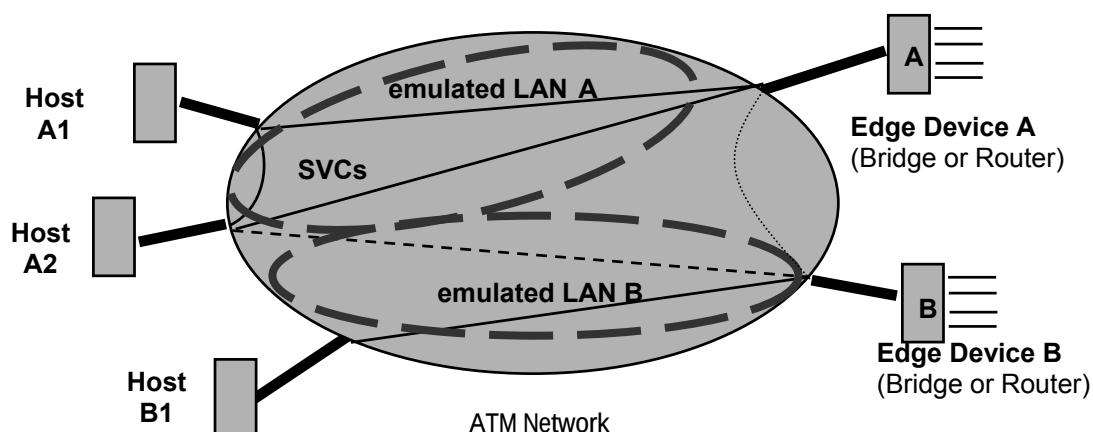
A pont - több-pont összeköttetéseket a hagyományos IP ATM felett nem támogatta. E funkciót úgy valósították meg, hogy minden csomópont kialakított egy pont-többpont összeköttetést az összes többi felé. Ennél erőforráskímélőbb megoldás a szétosztó-szerver (MCS: MultiCast Server) használata, ahol a szerver bármely csomóponttól kapott üzenetet szétküldi minden végpontnak. Ehhez egy IP multicast csoport címhez több ATM címet feleltet meg.

Az RFC 2225 több, adatbázisaikat egymással állandóan szinkronizáló ARP szerver használatát javasolja, mely meghibásodások ellen védettebbé teszi a hálózatot.

A hagyományos IP ATM felett megoldás előnye, hogy az IP hálózat számára is lehetőség nyílik az ATM biztosította minőségi garanciák kihasználására.

3.6.7. LAN emuláció (ATM Forum LANE 1.0 és LANE 2.0)

A LANE emuláció, szemben a CLIP megoldással, nem csak az IP forgalom átvitelére szolgál, hanem, mint neve is mondja LAN-t (pl.Ethernet-et vagy Token Ring-et) emulál, így minden 3. rétegbeli protokoll változtatás nélkül használható, például IPX, DecNet, Novell Netware Client, NetBEUI, stb.



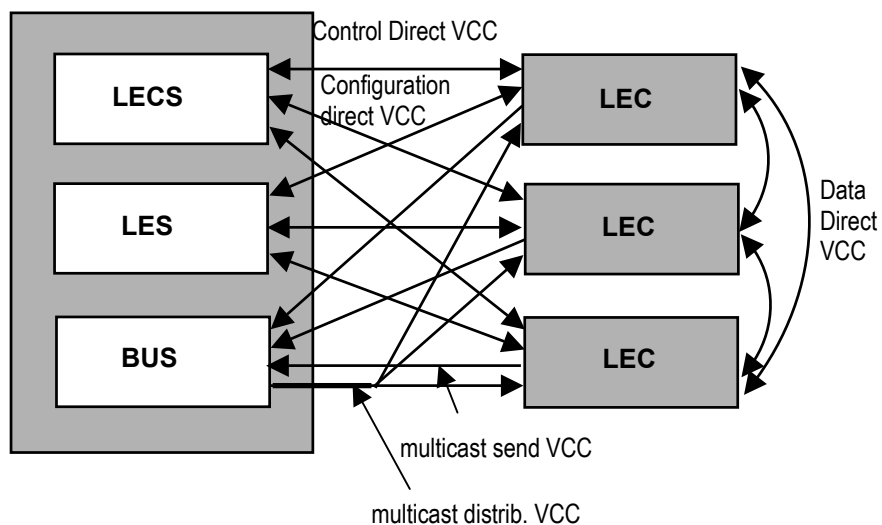
3.6.9 ábra. Emulált LAN-ok és viszonyuk LANE-ban.

Egy ATM hálózat felett létrehozhatunk emulált vagy virtuális LAN-okat (ELAN-okat), melyekbe ATM interfészkártyával rendelkező munkaállomásokat vagy ATM interfésszel rendelkező bridgek révén egész alhálózatokat csatlakoztathatunk. Itt is, egy adott ELAN-ba tartozó munkaállomások közvetlenül, SVC-ken keresztül kommunikálnak egymással, míg külön ELAN-okba tartozó állomások csak routereken keresztül képesek erre, ahol a routerek közti összeköttetés is egy ATM kapcsolókon/rendezőkön áthaladó ATM összeköttetés. E megoldás másik hátránya, viszont, hogy nincs lehetőség minőségi garanciák biztosítására, hiszen olyan LAN-t emulálunk mely maga sem képes erre.

A LAN-emuláció megvalósításhoz kliensekre (LEC: LAN Emulation Client) és 3 modulból álló kiszolgálóra van szükség. E három modul a:

- **LECS:** LANE konfiguráció szerver (LANE Configuration Server)
- **LES:** LANE szerver (LANE Server); fő funkciói a tagok nyilvántartása (azonosítás, vezérlési funkciók) illetve a címfeloldás.
- **BUS:** Ismeretlen cél című és szórt küldésre szolgáló szerver (Broadcast and Unknown Server); Ez olyasmi mint az MCS a CLIP-nél. Fő funkciója a szórás megvalósítása (pont-többpontos). Amennyiben ismeretlen cél című csomagot küldünk az osztott közeget emulálja, és míg nem jön létre közvetlen VCC a két végállomás között a kommunikáció is erre folyik.

A LEC és a LANE szolgálatot megvalósító szerver közti interfész a LUNI (LAN Emulation UNI).



3.6.10 ábra. A LANE kliensek és a LANE-t megvalósító szerver elemei.

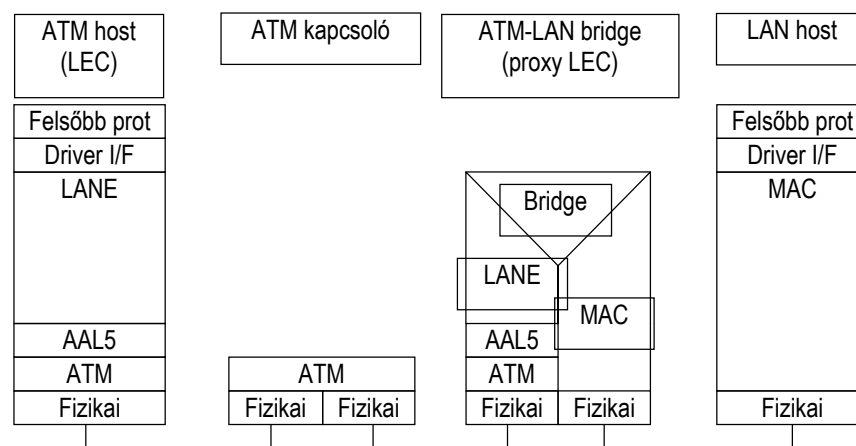
Nézzük, hogyan működik a LANE. Az adatátvitelt üzembehelyezés, illetve az adatátvitelhez szükséges összeköttetések kiépítése előzi meg jelzésüzenetek révén.

A LEC először a LECS-hez fordul konfigurációs információért. Ehhez egy ideiglenes "Configuration Direct VCC"-t hoz létre, majd egy "Control Direct VCC"-t hoz létre a címbejelentés, illetve címmegfeleltetés céljából a LES felé. Ezt követően egy egyirányú "Multicast Send VCC"-t hoz létre a BUS felé, a BUS viszont létrehoz egy multicat forward vagy distribute egyirányú pont-többpont összeköttetést a LEC-ek felé.

Ezt követően, ha egy LEC küld egy üzenetet/csomagot a LUNIn keresztül először egy táblázatban megnézik, hogy van-e már VCC az adott MAC (Media Access Control) cím felé. Ha igen arra küld, ha nem, akkor az adott MAC cím alapján kér a LES-től egy ATM címet, majd kiépít egy "Data Direct" VCC-t. Míg kiépül e VCC a csomagok küldhetők a BUS-on keresztül is. A használatlan összeköttetések "lejárnak" és megszűnnek. A használatlan MAC cím szintén lejár. Ekkor ellenőrzik, és ha nem találják meg a megfelelő LEC-et akkor meg is szűnik. Ez teszi lehetővé a LEC-ek mozgását egy ATM hálózatban.

Vegyük észre, hogy itt kettős címmegfeleltetésről van szó: a hagyományos IP cím - MAC cím, illetve a MAC cím - ATM cím megfeleltetésről, ahol csak az utóbbi a LANE feladata.

Az alábbi protokollverem ábrán látni, hogy hogyan csatlakoztathatunk egy ATM hálózati kártyával rendelkező munkaállomást (bal oldal), illetve egy LAN (pl. Ethernet) hálózatot vagy LAN kártyával ellátott munkaállomást (jobb oldal) egy ELAN-hoz.



3.6.11 ábra. A LANE protokollverme proxy-LEC-cel

A LANE szintén az AAL5 illesztési réteget használja. A proxy LEC teszi lehetővé LAN munkaállomások illetve teljes LAN alhálózatok csatlakoztatását ELAN-hoz.

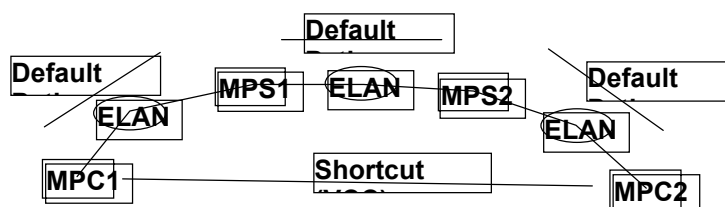
A LANE 2.0 változata néhány továbbfejlesztést tartalmaz: Lehetőség van több szerver használatára egy ELAN-on belül, a magasabb rétegek hozzáférnek az ATM nyújtotta QoS lehetőségekhez, jobb a multicast kommunikáció, több magasabb rétegbeli folyam osztható egy VCC-n, és támogatja az MPOA megoldást is.

3.6.8. ATM API

ATM alkalmazásprogramozó interfész használata olykor indokolt, ha szeretnénk saját ATM feletti alkalmazást megvalósítani, vagy fejleszteni a meglévőket. Az ATM hálózati interfész gyártók kártyáikhoz függvénykönyvtárakat adnak, melyek segítik a fejlesztést. Ez egy tudás- és időigényes megoldás. Pl. a Linux-ATM projektet meg kell említeni, mely a Linux operációs rendszert használó munkaállomások ATM alapú hálózatba kötését fejleszti.

3.6.9. Több protokoll ATM felett (ATM Forum MPOA)

Míg LANE esetén a nem azonos ELAN-ba tartozó kliensek nem képesek közvetlen összeköttetés kialakítására, erre az MPOA lehetőséget nyújt. Az MPOA eredetileg a Newbridge kezdeményezésére indult, de részben beépült a LANE 2.0-ba.



3.6.12 ábra. Összeköttetés kiépítés MPOA-ban.

Mint az ábrán is látható, ha különböző ELAN-ba kapcsolt munkaállomások szeretnének egymással közvetlen kapcsolatba lépni, előbb több MPOA serveren (MPS) keresztül veszik fel a kapcsolatot, majd az IP címnek megfelelő ATM címét visszakapván egy áthidaló VCC (shortcut) épül ki. E feladat megvalósításához ki kell terjeszteni a címfeloldást, az NHRP (Next-Hop Resolution Protocol) útvonalválasztási protokoll segítségével.

A LANE-hoz hasonlóan az MPOA is kliens-szerver architektúrán alapszik. A rendszer lefelé kompatibilis a LANE-val, így az MPC (MPOA kliens) tartalmazza a LEC funkcióit is, míg az MPS (MPOA server) a LANE szolgálatot biztosító funkciókat.

3.6.10. Többprotokollos címkekapcsolás (IETF MPLS)

Mint láttuk, ATM hálózat felett több módon lehet IP hálózatot kialakítani, azonban az egyszerűbb megoldások lehetőségei korlátozottak, míg a bonyolultabb rendszerek túl bonyolultak. Az ATM szerepe is döntően az IP forgalom szállítására korlátozódott. Így felmerült a kérdés, hogy szükséges-e két útvonalválasztási technikát, két címezési módszert, különböző jelzésrendszereket bonyodalmak árán megfeleltetni, vagy inkább ötvözzük e technikákat? A döntés az "ötvözés" mellett szólt. A mai MPLS implementációk jellemzően ATM hardvert használnak, megőrizték a címkézett cella alapú információtovábbítást virtuális áramkörök mentén (ezt LSP-nek nevezik: Label Switched Path), ám az útvonalválasztás és címezés az IP hálózatokban használatos megoldásra támaszkodik.

Az MPLS előnye a hagyományos IP hálózatokkal szemben, hogy az LSP-k révén egyenletesebben osztja el a forgalmat a hálózatban ami jobb kihasználtsághoz vezet. Dinamikus forgalommenedzsmenttel lehet terelgetni a forgalmat, lehet erőforrást foglalni és minőséget biztosítani, illetve nem kell minden egyes csomagnál útvonalválasztási döntést hozni, hanem folyamanként csak egyszer, ez után csak címkealapú csomagtovábbítást végzünk.

Előzményképpen említhető a Toshiba 1994-ben javasolt CSR: Cell Switching Routere vagy az Y cég 1996-ban javasolt IP switching megoldása. Az IBM szintén 1996-ban javasolta az ARIS (Aggregated Route Based IP Switching) megoldást. Végül a mai MPLS-hez illegközelebbi megoldás az 1996-ban a CISCO által javasolt Tag Switching. Valamennyi megoldás a fenti szempontokra alapozott.

Az MPLS-nek több fejlődési stádiuma van. Kezdetben az LSP-k rögzítettek, míg a fejlődés során igény szerint dinamikusan létrehozhatókká és bonthatókká válnak, teljes, ATM-éhez hasonló forgalommenedzsmenttel. Az MPLS nem csak ATM-re képes építeni, hanem egyéb, pl. Ethernet, Frame Relay, Token Ring és egyéb technikákra is, de az ATM alapú megoldása a gyakorlatban a legelterjedtebb. Többprotokollosnak viszont azért nevezzük, mert felette nem csak IP hanem általánosan beszéd, video, multimédia és adatátviteli alkalmazások egyaránt megvalósíthatók.

Az MPLS elvet általánosították hullámhosszirányítású hullámhosszosztásos (WR-DWDM) hálózatokra ahol a címke helyett a hullámhosszinformációra támaszkodnak. Ez az MPLambdaS, ahol a lambda a hullámhossz információra utal. Így MPLS/MPLambdaS kombinációval hatékonyabb kétrétegű hálózatokat lehet kialakítani, vagy tovább általánosítani GMPLS-re ahol 4-5 hálózati technikát (réteget) helyeznek egymásra. A GMPLS a generalised, azaz általánosított MPLS rövidítése, ami arra utal, hogy az üvegszálkapcsolt rétegre hullámsávkapcsolt, vagy közvetlen hullámhosszkapcsolt réteget helyeznek. Ezt követi az időosztásos nyálábolt réteg nagyobb időkeretekkel (pl. SDH-szerű keretzéssel) majd erre épül a csomag- vagy cella-kapcsolt felső réteg.

3.7. Minőségi szolgáltatások IP hálózatokban

Szerző: Szabó Róbert

Lektor: dr. Réthy György

3.7.1. A szolgáltatás minőség meghatározása

A *szolgáltatás minőség* (Quality of Service – QoS) nyújtás meghatározása nem egyszerű feladat, legtöbb esetben azonban tekinthetjük a szolgáltatást nyújtó (réteg, hálózati elem stb...) azon *képességének* melyek által meghatározott *szolgáltatásokat* nyújtanak a felhasználóknak (felsőbb rétegek, vég-felhasználók stb...) úgy, hogy az elvárt *menyiségi* és *minőségi* követelmények (valószínűségi vagy determinisztikus alapon) találkozzanak, amennyiben a felhasználó a forgalmi szerződés keretein belül marad. Az előbbi meghatározás célja, hogy magába foglalja mind a különböző hálózati rétegek közötti szolgáltatás minőséget mind pedig a tartományi határelemek vagy végpont-végpont közötti szolgáltatás minőséget, ha ezeket úgy tekintjük mint hálózati rétegek egyenrangú (peer-to-peer) kommunikációját. Természetesen, maga az elvárt QoS megfogalmazása alkalmazásonként nagyon eltérő lehet.

3.7.2. QoS támogatás fejlődése

A hagyományos Internet kizárólag egyfajta szolgáltatás minőséget biztosított a felhasználóknak amelyet *best-effort* szolgáltatásnak neveztek el. Maga a best-effort találoan kifejezi, hogy a hálózat nem biztosít semmilyen minőségi szolgáltatást, azonban mindent megtesz a lehető legjobb szolgáltatás nyújtásáért. Látható, hogy az előző egy gumi definíció, amit nem lehet számon kérni és nem lehet rá alapozni. A fenti működésből adódóan a felhasználók közötti megkülönböztetés tipikusan a hozzáférési technológiában jelent meg, mely által különböző szolgáltatás minőséget lehetett észlelni (pl. analóg, modem és ISDN hozzáférés, betárcsázás vagy állandó bérelt vonali hozzáférés [3.7.2]). Az elmúlt években azonban az egyre újabb és újabb alkalmazások és az általuk támasztott másfajta szolgáltatási igények megjelenésével az Internet szolgáltatók (Internet Service Providers – ISPs) részéről felmerült az igény az általuk nyújtott szolgáltatások differenciálásának technikai

alapjainak megteremtésére. A fenti folyamatot tovább erősítette, hogy a hajdani szlogen miszerint „IP minden felett” napjainkra úgy változott, hogy „minden IP felett”, ami ha lehet még jobban kikezdte a 60-as évekhez visszanyúló IP-s alapokat.

Éppen a fenti folyamatok miatt lehet mondani, hogy a felmerült igényekkel teljes összhangban az elmúlt évtizedben az Internettel kapcsolatos kutatások jelentős része olyan metódusok, protokollok és architektúrák kutatásával foglalkozott, melyek kielégíthetik a jelenlegi illetve a jövőben várható felhasználói és alkalmazási igényeket.

A fejezet továbbiakban az alkalmazások által támasztott QoS osztályozásával foglalkozik majd pedig a QoS bevezetésével foglalkozó kutatások eredményeit összegezi röviden.

3.7.3. QoS az alkalmazások szempontjából

Teljesen nyilvánvaló, hogy az örökké megújuló alkalmazások adják a hajtóerőt a hálózati szolgáltatások fejlesztéséhez, hiszen a hálózat van az alkalmazásért és nem fordítva. Az alkalmazásokat általában kétféle szempont szerint osztályozzák: i) az előre becsülhető átviteli sebességük alapján ill. ii) késleltetési vagy késleltetés ingadozási (delay jitter) toleranciájuk alapján [3.7.1].

Az átviteli sebességnek (i) megfelelően a következő kategóriákat használják:

Egyenletes (Stream)	Előre becsülhető és közel konstans sebességű átvitel (constant bit rate – CBR). Az alkalmazások ebben a kategóriában fix csúcssebesség (peak rate) korláttal rendelkeznek. Tipikus példaként említhetjük a videó alkalmazásokat vagy az IP feletti telefóniát.
Ingadozó (Burst)	Az átviteli sebesség előre nem becsülhető és erősen ingadozik időben. Tipikusan adatblokkok kerülnek átvitelre, melynél sebességi felső korlátot csak a fizikai átviteli kapacitások adnak. Jellemző példaként a fájl átvitelt említhetjük.

Az alkalmazások késleltetési toleranciáját (ii) a következő képen osztályozhatjuk:

Aszinkron (asynchronous)	Elasztikus alkalmazás bármilyen késleltetési elvárás nélkül. (Pl.: elektronikus levelezés, fájl átvitel...)
--------------------------	---

Szinkron (synchronous)	Flexibilis alkalmazások minimális időzírási kritériumokkal. (Pl.: web-szörfözés)
Interaktív (interactive)	Olyan alkalmazások, melyeknél a felhasználhatóság és vagy a funkcionalitás nem függ, azonban a felhasználói elégedettség valamennyire függ az észlelt késleltetési karakterisztikától. (Pl.: telnet, web-szörfözés)
Egyidejű (isochronous)	Felhasználhatóságot erősen befolyásolja az észlelt késleltetési karakterisztika. (Pl.: IP telefonia)
Feladat-kritikus (Mission-critical)	Használhatatlan, ha a késleltetési elvárások nincsenek kielégítve. (Pl.: PSTN minőségű IP telefonia)

3.7.4. QoS Megközelítések

A hálózatok a fent részletezett alkalmazási igényekhez igazodóan fejlődtek és fejlődnek ma is. A technológia legelső fázisában kizárólag sávszélesség biztosításának lehetősége nyújtotta a beavatkozás lehetőségét. Az átviteli kapacitások állandó növelésével minden alkalmazási igény kielégíthető. Ezt a megközelítésmódot nevezzük *túlméretezésnek*. Természetesen vannak olyan nézetek, melyek szerint a hálózat teljes egészében soha sem képes elegendő átviteli kapacitást biztosítani, ezért az alkalmazási elvárások változásával olyan intelligens módszerek bevezetésére és kutatására van szükség, mellyel biztosítható az alkalmazások megkülönböztetett kezelése. Ezen törekvések első zászlóshajóját az *integrált szolgáltatású* (integrated services – IS) hálózatok megjelenése jelentette. Ezen megközelítésmód erős hasonlóságot mutatott az ATM-ben használt erőforrás menedzselési módszerekkel. Az IS azonban- ahogy az ATM is- megbukott a széleskörű elterjedésben a nagyfokú komplexitása és az ebből adódó skálázhatósági problémái miatt. Tanulva a hibákból, napjainkban azon irányvonalak erősödnek, melyek az egyszerűséget, a skálázhatóságot és ezáltal az erőforrások prioritizálását szemben az erőforrások individuális foglalásaival részesítik előnyben. Ezen megközelítési módot differenciált szolgáltatásoknak (differentiated services –DS) nevezzük.

A QoS teljességéhez tartoznak még olyan nem elhanyagolható faktorok mint az együttműködési kérdések, megfelelő menedzsment, hitelesítés és számlázási keretek kidolgozása stb... Ezen funkciók azonban túlmutatnak jelen összefoglaló keretein; az integrált szolgáltatású és a differenciált szolgáltatású hálózati architektúra a következőben részletesen ismertetésre kerül.

3.7.5. Integrált szolgáltatások

Hordozó Szolgáltatások (Bearer Services)

A már létező ún. best-effort szolgáltatáson fölül az IETF meghatározott további két hordozót a *Guaranteed Quality of Service* [3.7.11] és a *Controlled Load Quality of Service* [3.7.12] szolgáltatásokat. Mindkét fent említett szolgáltatás erőforrás-foglalást és kontrollált késleltetést ill. csomagvesztést biztosít és mindkettő feltételez egy háttérben lévő hívásengedélyező folyamatot (admission control) valamint lyukas vödör (leaky-bucket/token bucket) forgalomleírást [3.7.13].

A két hordozó azonban számottevően különbözik is egymástól:

A *Guaranteed QoS* hálózati elem *QoS* szempontból ZÉRÓ puffereelési csomagvesztést biztosít és határozott felső korlátot a puffereelési késleltetésre mindaddig amíg az adatmennyiség az előre egyeztetett határokon belül van [3.7.11]. Ez azt is jelenti, hogy ezen szolgáltatás NEM kontrollálja sem a minimális sem az átlagos késleltetést sőt a késleltetés ingadozását sem; kizárólag csak a maximális késleltetés értékét biztosítja. Mivel azonban ez egy laza felső korlát – legrosszabb esetben is teljesül – különböző tanulmányokban kimutatták [3.7.11], hogy az adatfolyam nagy többsége sokkal kisebb késleltetést szenved mint a felső korlát. A szolgáltatás megvalósítása hívásengedélyezést (Call Admission Control - CAC) feltételez, melyhez *token bucket* (lásd alább) forgalomleíró paraméterek használatát szabványosították. A teljes körű szolgáltatás nyújtásához valamennyi közbenső elemnek támogatnia kell a szolgáltatást; mely hiányában a fenti erős garanciák nem biztosíthatóak. Azonban részleges támogatás esetén is jelentős szolgáltatás minőségi javulás érhető el (pl.: ha a torlódott hálózati elemek támogatják a szolgáltatást, míg a túlméretezettek nem).

A *Controlled Load Network Element szolgáltatás* egy olyan *QoS* minőséget nyújt megközelítőlegesen, mint amelyet ugyanezen szolgáltatás kapna terheletlen hálózat esetén. Mint látható, ez a meghatározás nem tartalmaz konkrétumokat, viszont nagyon egyszerű. És tulajdonképpen pont ez volt a célja az IETF-nek ezen szolgáltatási osztály meghatározásánál [3.7.12].

A szolgáltatás meghatározását értelmezve, egy alkalmazás, amely a *Controlled Load* szolgáltatást veszi igénybe a következőket várhatja el: i) nagyon

nagy valószínűséggel az átvendő csomagok sikeresen megérkeznek, ii) a késleltetés a csomagok túlnyomó részénél nem haladja meg jelentősen azt a minimális késleltetést amelyet bármely sikeresen átvitt csomag elszenvedett. Tehát virtuálisan veszteségmentes és késleltetés ingadozás szempontjából pedig terheletlen hálózatnak megfelelő kiszolgálást kap ezen szolgáltatás. Hasonlóan a Guaranteed QoS szolgáltatáshoz, hívásengedélyezés szükséges (CAC) a szolgáltatás biztosításához. Forgalomleíróként a már előzőekben említett token-bucket forgalomleírót használják.

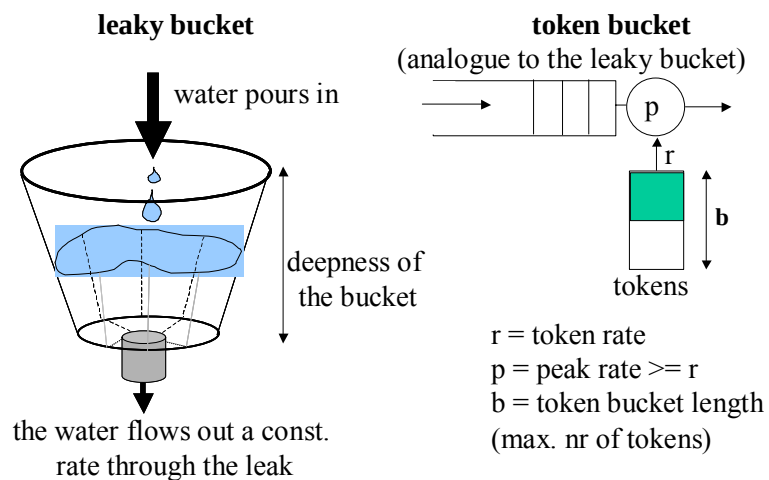
Forgalom szabályozás (Traffic Policing)

Az erőforrás-foglalás következményeként a hálózat garantálja az igényelt hordozó szolgáltatást a lerögzített QoS paraméterekkel, feltéve ha a forrás adatmennyisége a forgalmi szerződés keretein belül marad. Az előbbi szerződés ellenőrzését a határcsomópontok végzik.

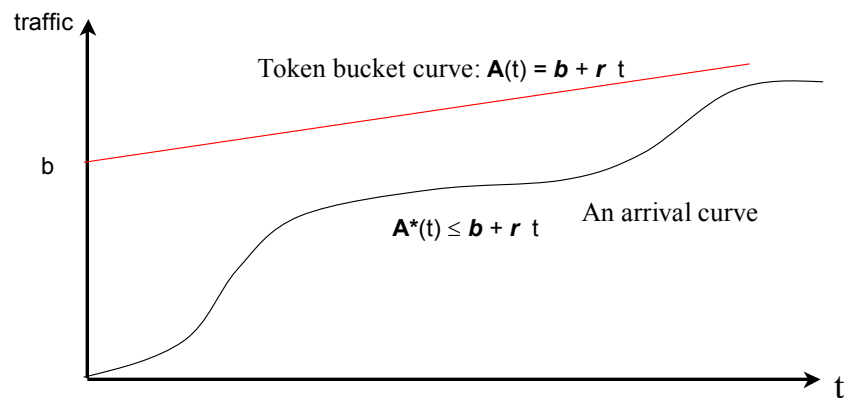
A felhasználó jogosultsága az erőforrás-foglalás igénybevitelére szintén vizsgálat tárgya.

Általánosságban elmondhatjuk, hogy a forgalom szabályozás legfontosabb célkitűzése az átviteli sebesség korlátozása ill. ellenőrzése. Ez leggyakrabban a hálózati határcsomópontokban kerül vizsgálatra, ahol azon forgalmi rész mely a megállapodott határokon belül marad változatlanul kerül továbbításra míg a többlet forgalom vagy eldobásra kerül vagy pedig a prioritási szint (dobás vagy késleltetés) megváltoztatásával kerül továbbításra.

A leggyakrabban használt forgalom szabályzó és ellenőrző a jelzés vagy lyukas vödör (token, leaky bucket) (lásd 1. ábra). Ezen jelzés vödrök tipikusan három paraméterrel kerülnek jellemzésre, melyek a jelzés generálás sebessége (r), a jelzés vödör mélysége (b) és a kimeneti csúcssebesség (p). A kimeneti csúcssebességet gyakran elhagyják amennyiben $p \gg r$, mely esetben végtelennek feltételezzük. A jelzés vödöröt használhatjuk mind a forgalmunk újraformázására mind pedig szabályzásra mely szerint eldobjuk vagy prioritásilag hátrább soroljuk a többlet forgalmat. A.2. ábra egy jelzés vödör paramétereknek megfelelő forgalmat mutat be.



3.7.1. ábra. Lyukas és jelzés vödör

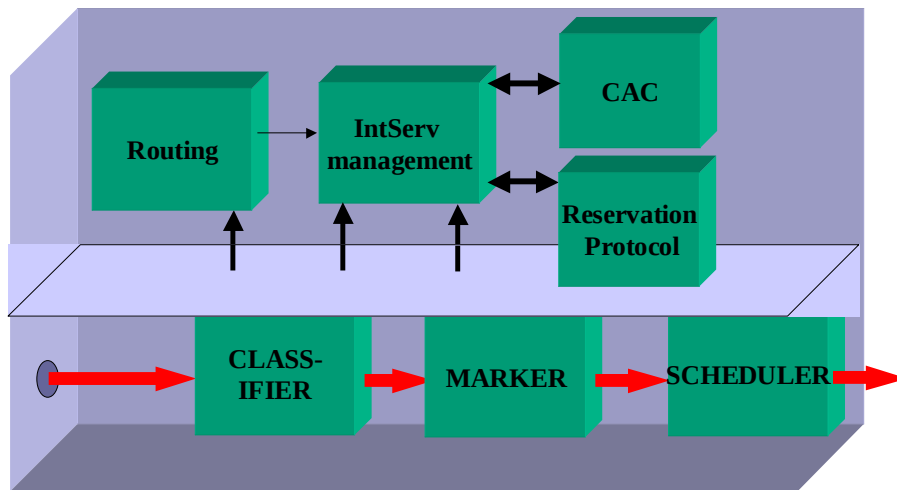


3.7.2. ábra. Jelzés vödör formázott forgalom

Forgalom vezérlés (Traffic Control)

Forgalom vezérlésnek hívjuk mindazokat a funkciókat, melyek által a hálózati eszközök (útvonalválasztók) megkülönböztetett (különböző) minőségű szolgáltatást tudnak nyújtani.

A megvalósítandó vezérlést forgalom vezérlésnek (traffic control) nevezik és alapvetően a következő funkciókat kell ellátnia [3.7.14]: csomagok ütemezését (packet scheduler), csomagok osztályozását és jelölését (packet classifier and marker) és hívásengedélyezési funkciókat (call admission control) (lásd. 3 ábra).

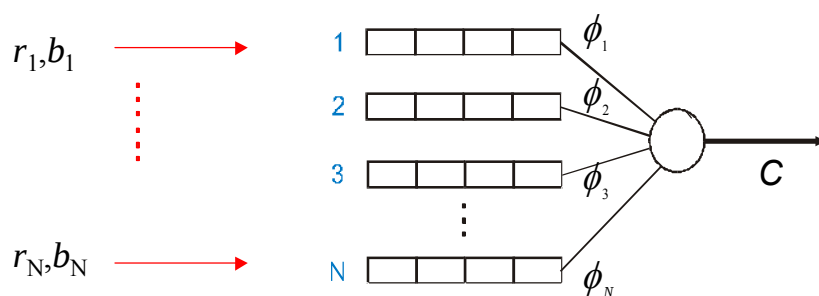


3.7.3. ábra. egy IntServ csomópont

Csomagütemezés

Csomagütemezés az egyik legfontosabb komponens a minőségi szolgáltatás biztosításában, hiszen ezáltal biztosított a kimenetre várakozó csomagok közötti megkülönböztetés lehetősége. Az integrált szolgáltatású hálózatok egyik ilyen ütemezője a súlyozott igazságos sorbaállítás vagy közismertebb nevén a *weighted fair queueing* (WFQ). WFQ ütemezők jellemző tulajdonsága, hogy a különböző kapcsolatokat egymástól elkülönülten kezeli és számukra sáv szélességen alapuló erőforrás allokációt biztosít úgy, hogy a rendszerben fennmaradó többlet sáv szélességet a kapcsolatokhoz rendelt súlyok alapján igazságosan szétosztja. További előnye, hogy tetszőleges jelzés vödörök által meghatározott forgalmi együttes ($\{r_i, b_i\}; i=1, \dots, N$) számára kapcsolatonként biztosítani tudja a maximálisan elszenvedett késleltetés felső korlátját ($D_i = b_i * \sum r_j / C * r_i; i=1, \dots, N$), ahol C a szerver kapacitás, valamint csomagvesztés mentes továbbítást amennyiben az egyes kapcsolatokhoz rendelt pufferek (B) nagyobbak mint a hozzájuk tartozó jelzés vödör mélység ($B_i > b_i; i=1, \dots, N$). A fenti megállapítás természetesen csak stabil rendszerben értelmezhető ($\sum r_j < C; i=1, \dots, N$) (lásd .4 ábra). Általánosabb esetben az egyes kapcsolatokhoz a fentiektől eltérő súlyok (ϕ) is rendelhetők.

Fontos megjegyeznünk, hogy a fenti példából is láthatóan az integrált szolgáltatású struktúra folyamonti puffereket kíván a hálózatban a nagyon szigorú (garantált QoS szolgáltatási elem) minőségi elvárások teljesítésének érdekében (lásd .4 ábra). Ezen tulajdonsága miatt, rossz skálázhatóság, gerinchálózati megvalósításoknál nemigen kerülhet szóba.



3.7.4. ábra. Weighted fair queueing (WFQ)

Erőforrás-foglalás RSVP-vel

Az RSVP (resource ReSerVation Protocol) az egy végpont-végpont közötti erőforrás menedzselő protokoll [3.7.15]. Az RSVP támogat egyedi erőforrás foglalásokat vagy filter által meghatározott folyamatok közös erőforrás menedzselését; egyirányú foglalást valósít meg kizárólag a nyelőből indulva; támogatja a pont-többpont kommunikációt a kérések hatékony összevonásával. Mint ahogy említettük, az RSVP egyirányú foglalásokat kezel, azonban bármelyik fél párhuzamos kezdeményezheti ezen foglalásokat. Ha az ISO-OSI rétegbeli elhelyezkedését tekintjük, akkor tipikusan az átviteli (transport) rétegben található, annak ellenére, hogy valós adatfolyamot ez a protokoll nem szállít, hanem azok erőforrás szükségleteit menedzseli. Néha feltételezik, hogy az RSVP a kommunikációs útvonalakat is menedzseli, ez azonban nincs így. Az RSVP teljesen független az útvonal-választástól, amivel egy jól meghatározott interfészen keresztül kommunikál biztosítva a különböző útvonalválasztókkal az együttműködést.

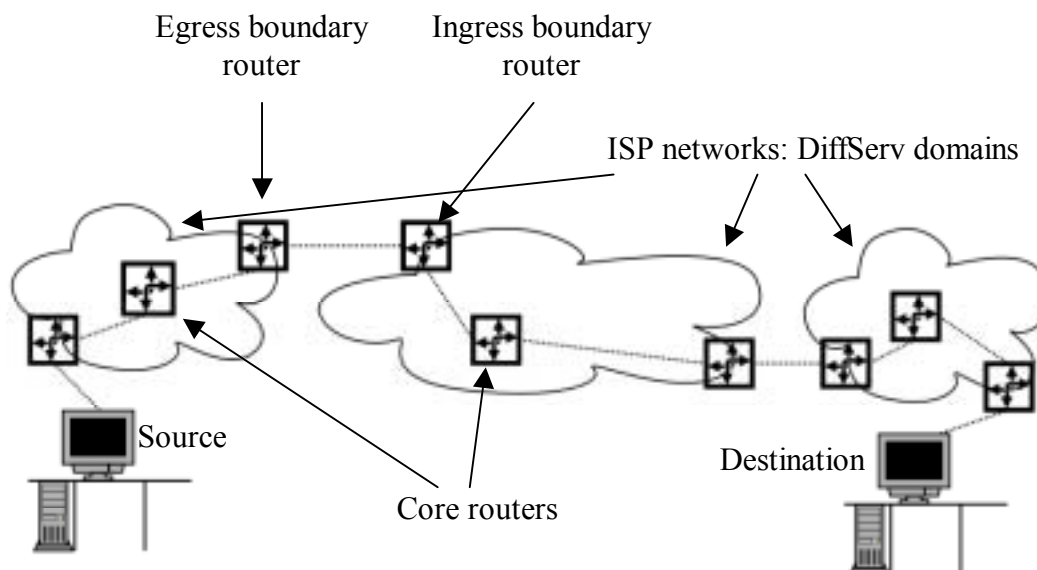
Az RSVP az erőforrás-foglalásokat ún. „soft” állapotokként kezeli, mely szerint az RSVP protokollnak periodikusan frissíteni (megerősíteni) kell az állapot információkat különben azok lebontódnak [3.7.15].

3.7.6. Differenciált szolgáltatások

A differenciált szolgáltatású modell (DS) azon az egyszerű alapelven működik, hogy minden a hálózat határán belépő forgalom egy osztályozón és kondicionálón halad keresztül, mely által minden csomaghoz hozzárendelődik egy *globális tulajdonság* (behavior aggregate – BA) és ezen keresztül egy DS kód bejegyzés (DiffServ Code-Point – DSCP) az IP fejlécben. A tartomány/hálózat belsejében ezután a csomagok a csomópontokhoz hozzárendelt *csomóponti tulajdonság* (per-hop behavior – PHB) alapján továbbítódnak amelyek a már fent említett DSCP alapján működnek. Egy DS hálózat legkisebb önálló egysége a DS tartomány (domain),

amelyen belül a szolgáltatások teljes szintje egyeztetett. Egy ilyen tartomány tipikusan kétfajta csomópontot tartalmazhat: határcsomópontot és hálózat belső csomópontot (core). A hálózat belső csomópontok jellemzően nem vesznek részt a kapcsolódó jelzésekben, kondicionálásban és egyéb komplex funkciókban hanem kizárólag nagy sebességgel továbbítják a csomagokat. Az így létrehozott architektúra jelentősen jobb skálázhatósági paraméterekkel rendelkezik mint az előbbieken említett IS struktúra, hiszen a komplex funkciókat kizárólag a tartomány határán elhelyezkedő eszközök végzik. Természetesen a DS mező az IP fejlécben tartalmazza mindazon szükséges információkat melyek a megfelelő pufferek, kiszolgálási és csomageldobási mechanizmus meghatározásához kellenek; az előbbi funkciókat hívjuk PHB-nek. A szolgáltatás meghatározása szempontjából pont ezek a PHB-k a meghatározóak. Ezzel szemben a felhasználókat tipikusan nem a belső szolgáltatási definíciók érdeklik, hanem az általuk elérhető szolgáltatások. Ezen gondolatmenet alapján a felhasználóknak a szolgáltatás igénybevétele előtt meg kell egyezniük a szolgáltatóval az igényelt szolgáltatás paramétereiről. Ezek a megegyezések a *szolgáltatási szintű megállapodások* vagy *service level agreements* (SLAs). Az SLA a technikai meghatározáson kívül – melyet *szolgáltatási szint specifikációnak* vagy *service level specifications* (SLS) hívnak – tartalmaz még fizetési feltételeket, szolgáltatási időket stb... Technikai szempontból az SLS legfontosabb része a *forgalom kondicionálási specifikáció* (traffic conditioning specification – TCS) mely magába foglalja a részletes szolgáltatási paramétereket mint: igényelt átviteli sebesség, csomagvesztési arány és vagy késleltetés, az a forgalom profil amire a szolgáltatás megkötött valamit, hogy a profilon kívül eső plusz forgalom hogyan kezelődjön.

Természetesen ahhoz, hogy végponttól-végpontig működő szolgáltatást nyújthassunk az egyes tartományoknak együtt kell működniük. Ez annyit jelent, hogy azon tartományokban melyhez a felhasználó közvetlenül nem kapcsolódik a szomszédos tartománynak kell megkötnie a megfelelő megállapodásokat (SLA) az ügyfelet képviselve. Az ilyen, együttműködő tartományok halmazát hívják DS régióknak. Egy ilyen régiót mutat az .5 ábra. [3.7.17]



3.7.5 ábra. Differenciált szolgáltatású architektúra

A szolgáltatás teljesítése alatt vagy a végpont vagy pedig az első határcsomópont beállítja a csomag DS mezejét az előre egyeztetett megállapodásnak megfelelően.

Az IETF differenciált szolgáltatású munkacsoportja meghatározta mind az IPv4-es (.6 ábra) mind pedig az IPv6-os (.7 ábra) DS mezőt [3.7.3]. Az IPv4-ben a régi *type of service* (ToS) mező került átdefiniálásra, míg az IPv6-ban a *traffic class* nyolcas (octet/byte) hordozza ezen információt. Ezen 8 bites DS mezőből jelenleg 6 bit van használatban míg a maradék kettő fönntartott későbbi használatra és a csomópontok a működés szempontjából figyelmen kívül hagyják. [3.7.17]

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time To Live		Protocol	Header Checksum	
Source Address				
Destination Address				

3.7.6 ábra. IPv4 fejléc formátum

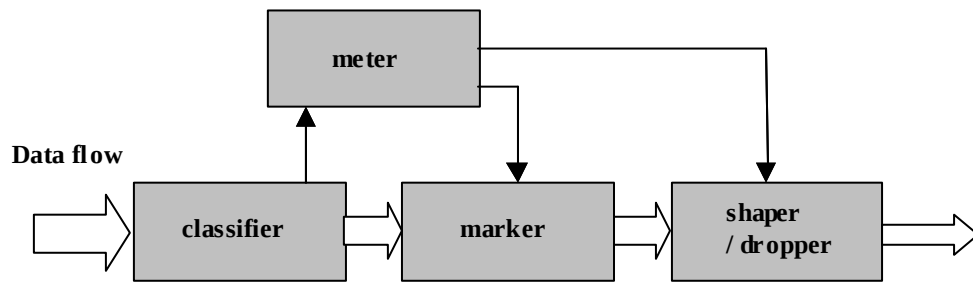
Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination Address			

3.7.7 ábra. IPv6-os fejléc formátum

A DS kódpont kiosztásoknál természetesen ügyeltek az IPv4-es korábbi ToS mezőkkel való kompatibilitásra is [3.7.3].

Forgalom osztályzás és kondicionálás

Amennyiben az SLS megegyezés megszületett, a szolgáltatónak feladata, hogy a határ csomóponti forgalom kondicionálóját megfelelően beállítsa. A forgalom kondicionálása szükséges ahhoz, hogy a szolgáltató megfelelően menedzselhesse a hálózatot ill. hogy biztosítani tudja az előre megalkudott szolgáltatási minőséget. Ezzel együtt jár a nem megegyezés szerinti forgalom kiszűrése esetleges eldobása is [3.7.4]. A forgalom osztályozási elvek határozzák meg a forgalom azon részét, amely a differenciált szolgáltatásban részesül azáltal, hogy egy vagy több kódpontnak megfelelően beállítódik a fejléc. A forgalom osztályozók az IP fejlécben hordozott információ alapján működnek. Két alaptípusuk létezik, az egyik a BA osztályozó mely kizárólag a DS kódpont alapján működik, míg a többmezős (multi-field) osztályozók a fejléc összetettebb részeit is használják mint a forrás és cél cím, DS mező, protokoll azonosító, forrás- és célport stb... [3.7.4]. Az osztályozók döntései alapján különböző feldolgozási útvonalra kerülnek a csomagok. A forgalom kondicionálás a fentiek felül *forgalom mérést* (metering), *forgalom formázást* (shaping), és/vagy *újrakezelést* (re-marking) végez, hogy a hálózatba belépő forgalom a forgalom profilnak megfelelő legyen ami a TCS-ben került meghatározásra (lásd .8 ábra). A fenti kondicionálás alapján a forgalom vagy profilon belüli vagy profilon kívüli (in-profile/out-profile) lesz. A profilon kívüli forgalom vagy várakozik amíg profilon belüli lesz (formázás), vagy eldobásra kerül, vagy új kódpont beállítást kap vagy akár változatlanul továbbításra is kerülhet plusz díj ellenében. Amennyiben új kódpont hozzárendelés történik, úgy az új kódpont szolgáltatás minőségileg alacsonyabb osztályt kell hogy meghatározzon. [3.7.17]



3.7.8 ábra. Forgalom kondicionálás DS csomópontban

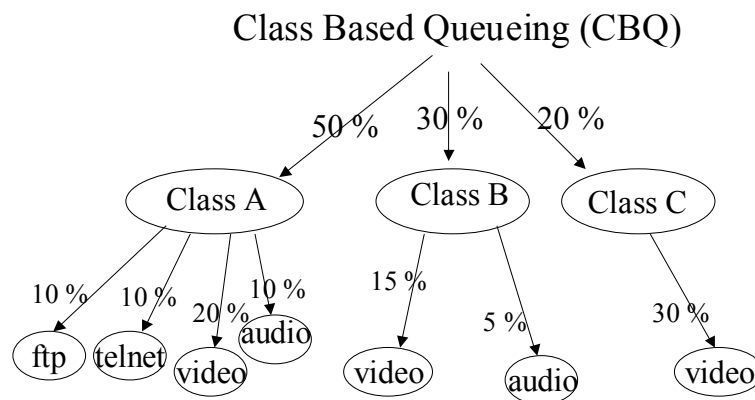
DiffServ ütemezés és puffer menedzsment

Ellentétben az integrált szolgáltatású megközelítéssel a differenciált szolgálatú hálózatok meghatározásánál az ütemező struktúra nem került lerögzítésre hanem egyedüli követelményként a különböző forgalmi osztályok megkülönböztetett kezelésének biztosítása szükséges. Ezáltal a megvalósítási lehetőségek a gyártók kezébe kerülnek nagyobb szabadsági fokot adva az újításoknak. Mindezek ellenére mondhatjuk, hogy van egy olyan ütemezési eljárás amely sikeresen pályázhat differenciált szolgáltatású hálózatokban az ütemező eljárás megvalósítására, mégpedig az osztály alapú sorbaállítás vagy közismertebb nevén a *class based queueing* (CBQ). CBQ ütemezőt úgy képzelhetjük el mint egy hierarchikus WFQ vagy WRR (weighted round robin) amely nem folyamokon hanem forgalmi osztályokon működik. Fontos, hogy a súly szerinti megkülönböztetésen felül a CBQ támogatja az egyes szinteken belüli prioritásos megelőzőséget is. Továbbá minden leszármazott osztály korlátozható erőforrás hozzáférésben vagy számára másoktól szabadon maradt erőforrás leosztható. A .9 ábra egy ilyen lehetséges leosztást mutat a hozzárendelt súlyokkal. Megfigyelhető, hogy ugyanazon forgalmi típusok megjelenhetnek különböző helyein a hierarchikus fának, természetesen különböző minőségi paramétereket biztosítva.

DS szolgáltatások osztályozása

A DS tartomány által felajánlott szolgáltatásokat *minőségi* vagy *mennyiségi* szolgáltatásként osztályozhatjuk. A mennyiségi szolgáltatás fix garanciákat nyújt amely megfelelő mérési eljárással ellenőrizhető más párhuzamos szolgáltatásoktól függetlenül. A minőségi szolgáltatások a fentiekkel szemben nem biztosítanak semminemű garanciát hanem relatív kiszolgálási sorrendet biztosít más szolgáltatási osztályokhoz mérten. Ez utóbbi szolgáltatás nehezen ellenőrizhető explicit

módszerekkel, kizárólag más szolgáltatási osztályok összehasonlításával számszerűsíthető.



3.7.9 ábra. class based queueing (CBQ)

A fentiekén túl egy másíkfajta szolgáltatási megközelítést ismertet az [3.7.5,6] cikk, melyben az osztályozás alapját a felajánlott QoS adja. [3.7.17]

Szolgáltatási szabványok

Jelenleg 4 szolgáltatás van szabványosítva a DS-en belül, ezek: i) az alapértelmezett best-effort PHB, ii) a kompatibilitási osztály-választó csoport, a iii) biztosított továbbítási (assured forwarding) PHB osztály valamint a iv) gyorsított továbbítási (expedited forwarding) PHB csoport. Az alapértelmezett osztálynak jelen kell lennie minden DS hálózatban [3.7.3] és működését [3.7.7]-ben rögzítették. A [3.7.3]-ban a '000000' kódpontot ajánlják az alapértelmezett osztály számára és minden az adott tartomány számára értelmezhetetlen kódpontot szintén erre a kódpontra kell átállítani. A kompatibilitási osztály-választó csoport valósítja meg a visszamenőleges támogatást az eredeti ToS definícióknak megfelelően. Ennek megfelelően minden kódpont mely a 'xxx000' értéket tartalmaz az eredeti ToS meghatározáshoz legközelebb eső támogatott DS kódpont szerint kezelendő. A biztosított továbbítási osztály két további paramétert használ: (x) amely meghatározza az elsőbbségi sorrendet (prioritást) és (y) mely a csomageldobási elsőbbséget határozza meg (AFxy). A előzőeknek megfelelően AF11 és AF12 valószínűleg ugyan abba a kiszolgálási sorba kerül, de AF12-nek magasabb csomageldobási arányt kell tapasztalnia torlódás esetén. (A megvalósítás használhat különböző sorokat de a sorrendhelyességet meg kell őriznie.) A rendelkezésre álló

12 AF kódpont 4 AF osztályra van felbontva és mindegyiken belül 3 csomageldobási sorrendre [3.7.8]. Végezetül, a gyorsított továbbítási osztály biztosítja a fix mennyiségi garanciákat, vagyis alacsony csomagvesztést, alacsony késleltetést és késleltetés ingadozást valamint átviteli sebességet a megalkudott mértékig [3.7.9]. Ez a szolgáltatás egy virtuális bérelt vonalként jelenik meg a felhasználók felé. [3.7.9]-nek megfelelően, ha abszolút prioritással rendelkezik ezen osztály akkor az összes erőforrás-felhasználását korlátozni kell azért, hogy a többi szolgáltatási osztály „kiéhezését” megakadályozzuk. [3.7.17]

Sávszélesség ügynökök (Bandwidth Broker-BB)

Sávszélesség ügynökök (BB) végzik a dinamikus menedzselését a DS hálózatok erőforrásainak. Ezen funkciók magukba foglalják a hívásengedélyezést, az útvonalválasztók beállítását, az erőforrás foglalások jelzését és felhasználói hitelesítést. Továbbmenve, a BB-k felelősek az egyes tartományok közötti erőforrás menedzselésért is. Ezen megközelítés szerint a BB-k olyan ügynökök, amelyek eldöntik, hogy a felhasználó kérés kielégíthető-e, és amennyiben igen, úgy a felmerülő erőforrás menedzselési funkciókat ellátják. Tipikusan DS tartományonként legalább egy BB található.

Irodalomjegyzék

- [3.7.1] Stardust Forums, Inc.: "The Need for QoS", White paper, <http://www.qosforum.com/>, 1999.
- [3.7.2] K. Nichols, S. Blake, F. Baker and D. Black: "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", Internet RFC 2474, December 1998
- [3.7.3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss: "Architecture for Differentiated Services", Internet RFC 2475, December 1999
- [3.7.4] H. Saito, Cs. Lukovszki, I. Moldován: "Local Optimal Proportional Differentiation Scheduler for Relative Differentiated Services", IEEE International Conference on Communication and Computer Networks, 2000, Las Vegas, Nevada, USA
- [3.7.5] C. Dovrolis, P. Ramanathan: "A case for relative differentiated services and the proportional differentiation", IEEE Network, September/October 1999, pp. 26-34
- [3.7.6] F. Baker: "Requirements for IP Version 4 Routers", RFC1812, June 1995
- [3.7.7] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski: "Assured Forwarding PHB Group", Internet RFC2597, June 1999
- [3.7.8] V. Jacobson, K. Nichols and K. Poduri: "An Expedited Forwarding PHB", Internet RFC2598, June 1999
- [3.7.9] Weibin Zhao, David Olshefski and Henning Schulzrinne, "Internet Quality of Service: an Overview," Columbia University, New York, New York, Technical Report CU-CS-003-00, Feb. 2000.

- [3.7.10] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997., <ftp://ftp.isi.edu/in-notes/rfc2212.txt>
- [3.7.11] J. Wroclawski, "Specification of the Controlled-Load Network Element Service", RFC 2211, September 1997., <ftp://ftp.isi.edu/in-notes/rfc2211.txt>
- [3.7.12] S. S. Sathaye, "Traffic Management Specification Version 4.0", The ATM Forum Technical Committee, ATM Forum/95-0013R10, February 1996
- [3.7.13] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC-1633, June 1994
- [3.7.14] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC-2205, 1997
- [3.7.15] R. Neilson, J. Wheeler, F. Reichmeyer, S. Hares, "A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment, Version 0.7", Internet2 Qbone BB Advisory Council, 1999. August
- [3.7.16] Császár András, "Differentiated Services for Voice Communication", Master's Thesis, Budapest University of Technology and Economics, 2001

3.8. A mobil hálózatok kapcsolástechnikája

Szerző: dr. Adamis Gusztáv

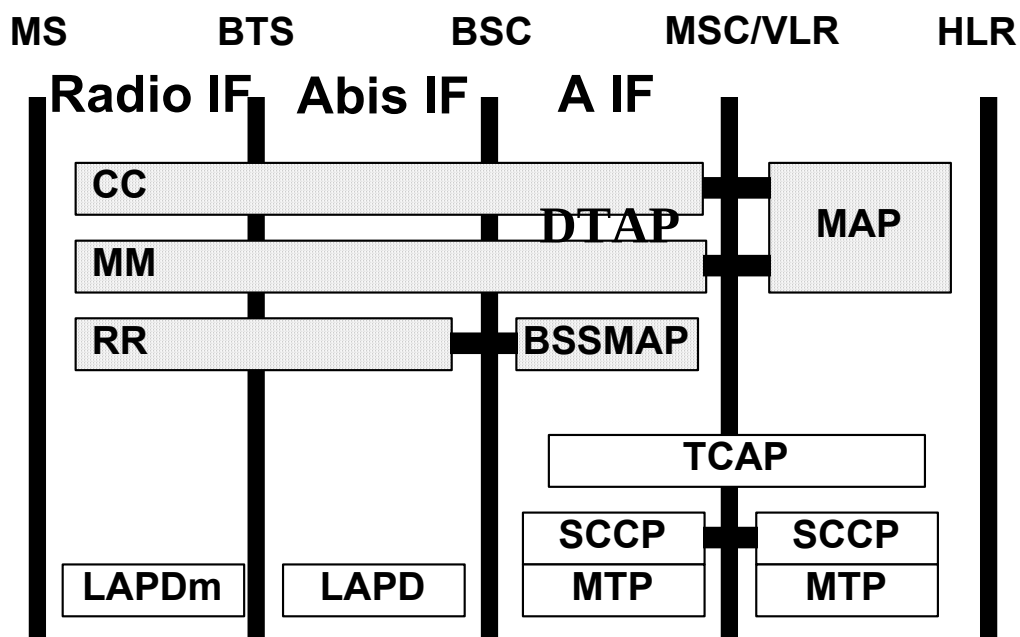
Lektor: dr. Tarnay Katalin

A mobil hálózatokban alkalmazott kapcsolástechnika a lényeges eltérése a fix hálózatban alkalmazottól az, hogy itt kezelni kell tudni az előfizető mozgásából fakadó problémákat. Erre különböző funkcionális egységeket és speciális protokollokat használnak. A mobilitás vezérlésével kapcsolatos legfontosabb egységek a HLR (Home Location Register) és a VLR (Visitor Location Register). HLR-ből minden hálózatban egy van és ez tartalmazza a szolgáltató minden előfizetőjének összes adatát, beleértve azt is, hogy pillanatnyilag hol tartózkodik. Minden mobil központhoz (MSC) csatlakozik egy-egy VLR, ahol azoknak az előfizetőknek az adatai találhatók meg, akik az adott MSC által kiszolgált környezetben tartózkodnak. Amikor egy előfizető bejelentkezik egy MSC körzetbe (Location Update), akkor az MSC értesíti erről az előfizető HLR-jét, és a készülékhez egy temporális hívószámot (MSRN - Mobile Station Routing Number) rendel. A mobilitás menedzselése az MSC-k és a többi hálózati szintű egység (HLR, VLR, stb.) között a MAP protokoll segítségével történik, amely a CCSS7-es jelzésarchitektúra része, és amely használja a CCSS7 alsóbb szintű protokolljainak szolgáltatásait. Azonban magának a hívásnak a felépítésére azután már az ISUP protokollt használják.

Az MSC és a mobil készülék között többféle funkcionális egység, többféle átvivő közeg és többféle protokoll használatos.

A központ (MSC) és az adóállomás (BSC) közötti vezérlésre szolgál a BSSMAP protokoll, amely egyik legfontosabb feladata a hívásokhoz rádiócsatorna rendelése. A BSC és az MSC közötti összeköttetésen (A interfész) még transzparens módon áthaladnak azok az üzenetek is, amelyek a központ és a készülék (MS) közötti, a hívások felépítésével (CC) és a mobilitás vezérlésével (MM) tevékenységeket vezérlik. Ezen üzenetek összessége alkotja a DTAP protokollt.

Az adóállomás és az adó (BTS) között található az Abis interfész. Itt a jelzések már nem a CCSS7 hálózat, hanem - hasonlóan az ISDN előfizetői protokolljához, a DSS1-hez - a LAPD protokoll segítségével jutnak el a megfelelő helyre, míg a rádió interfészen ennek egy változatát, a rádiótovábbításhoz optimalizált (elsősorban a sokkal kisebb üzenethosszakban eltérő) változatát használják. Itt, az előfizetői oldalon, azaz a készülék és az adó között, a rádióerőforrások menedzselése az RR protokoll segítségével történik. A további fejezetekben áttekintjük a legfontosabb protokollokat, végül pedig egy példán bemutatjuk használatukat.



3.8.1. ábra. Rádiós kapcsolatok CCSS7 jelzés struktúrája

3.8.1. SCCP

Míg az MTP csak arra alkalmas, hogy egy adott hálózaton belüli két jelzőpont között bonyolítson le jelzőforgalmat, addig az SCCP (Signalling Connection Control Part - Jelzőkapcsolat vezérlő egység) tetszőleges, akár két különböző hálózaton lévő jelzőpont között képes kapcsolatot teremteni. Ennek alapját az képezi, hogy az SCCP nem csak a CCSS7 jelzőrendszerben használatos (pontkódokon alapuló) címzési képességekkel rendelkezik, hanem az irányításhoz a CCSS7-en kívüli azonosítókat, ún. globális címek (Global Title, pl. telefonszám, IMSI stb.) is felhasználhatók. Mivel a CCSS7 jelzőpontkódok csak 14 bitesek, ezért ezek csak mintegy 15 ezer jelzőpont megkülönböztetésére alkalmasak világszerte. Ez

azonban kevés. Ezért minden szolgáltató a saját hálózatában saját jelzéspontkódjai alapján irányít, és csak néhány jelzéspontja kapcsolódik a nemzetközi jelzés-hálózathoz. Ezek az ún. híd (Gateway) jelzéspontok két jelzéspontkóddal rendelkeznek, egy a nemzetközi hálózatban használt, egy pedig a saját hálózatban. Hogy mikor melyik formátumot kell használni, ez az MTP által használt SIO oktett jelzi. Ha egy olyan jelzésösszeköttetés felépítésére van szükség, amelyben szereplő jelzéspontok ugyanabban a hálózatban vannak, akkor az SCCP a hagyományos pontkódok alapján irányít. Ellenben, ha a jelzéspontok különböző hálózatokban vannak, (ez pl. akkor fordulhat elő, ha egy másik szolgáltató HLR-jét kell elérni) akkor az SCCP a példabeli HLR telefonszámát, mint globális címet használja irányításra. Először az alapján meghatározza, hogy a célhálózat felé melyik kimenő gateway-t kell elérnie a saját hálózaton belül. Oda az irányítást a saját pontkódok alapján kéri az MTP-től. A Gateway SCCP meghatározza - a globális cím segítségével -, hogy a célhálózat belépő jelzéspontja melyik, és a nemzetközi hálózatban használatos pontkódok alapján odairányítja az üzenetet, míg a célhálózat Gateway SCCP-je - szintén a globális cím alapján - eldönti, hogy a saját hálózatán belül melyik jelzéspont felé kell irányítani az üzenetet, a saját pontkódjait használva.

Az SCCP arra is képes, hogy részhálózatokat (subsystem) is meg tudjon különböztetni egy jelzésponton belül. Ezek pl. HLR, VLR, MSC stb. lehetnek.

Az SCCP kétféle kapcsolat-orientált és kétféle kapcsolat nélküli szolgáltatást tud nyújtani (az utóbbiak közül azonban a GSM csak az egyiket használja).

A kapcsolat nélküli szolgáltatást a tranzakciók (pl. HLR lekérdezés) jelzésigényének kielégítésére használjuk. Ilyenkor az SCCP tulajdonképpen csak az arra kell, hogy interfészt biztosítson a CCSS7 hozzáféréshez, illetve az irányításra. A tranzakciók menedzselésére (pl. kérdés/válasz összerendelése) egy másik protokoll, a TCAP szolgál.

A kapcsolat-orientált szolgáltatásait az SCCP-nek a BSSAP/DTAP protokoll használja ki, amikor kapcsolatot kell felépíteni az MSC és a mobil készülék között, például Location Update során.. A kapcsolatfelépítés menete nagyon hasonló az ISUP-éhoz, csak az üzenetek mások (CR - Connection Request - kapcsolatkérés, CC - Connection Confirmed - kapcsolat elfogadva, DT1 - Data Form - adat, RLSD - Released - bontás, RLC - Release Complete - bontás kész), illetve a jelzések és a vezérelt kapcsolat összerendelése nem egy fizikai jellemző (az áramkör-azonosító

kód), hanem a két végpont által véletlenszerűen kiosztott logikai kapcsolatazonosítókkal (SLR/DLR - Source/Destination Local Reference - Forrás/cél oldali referencia) történik.

3.8.2. TCAP

A GSM hívások vezérlése során ún. tranzakciókat használunk (pl. adatok lekérdezése a HLR-ből). A tranzakciók során nem kell kapcsolatot felépíteni, de a tranzakciókat valahogy azonosítani kell, illetve ha a tranzakció kérdés-válasz alakú (ez így van a legtöbb esetben), akkor jelezni kell tudni, hogy melyek az összetartozó üzenetpárok. Többek között ez a feladata a TCAP (Transaction Capabilities Application Part - Tranzakciós képességek felhasználói egység) protokollnak, amely az SCCP kapcsolat nélküli osztályát, azon belül is az UDT (Unit Data - adategység) üzenetet használja. Az SCCP-re itt csak azért van szükség, hogy interfészül szolgáljon a CCSS7 hálózathoz való hozzáféréshez (SCCP címezés, MTP használat), a tranzakciók menedzseléséhez nem nyújt egyéb segítséget, az kizárólag a TCAP feladata.

A TCAP ún. dialógusokat (tranzakciókat) képes kezelni, amelyek egy vagy több művelet (operation) végrehajtásából állhatnak. Ennek megfelelően egy TCAP üzenet két fő részt tartalmazhat: egy dialógust vezérlő és egy vagy több komponenst (műveletet) vezérlő részt.

A TCAP üzenetek BEGIN ([dialógus] kezdés), END (vége), CONTINUE (folytatás) ill. ABORT (megszakadás) típusúak lehetnek. Egy dialógus titpkusan egy BEGIN/END párosból áll, de ha a kérdés/válasz nem fér el egy-egy üzenetben illetve a tranzakció során több műveletet kell végeznünk, bármely oldal tetszőleges számú CONTINUE-t iktathat be. A dialógusnak a normálistól eltérő módon történő megszakadását is lehet jelezni, azzal együtt, hogy ez a TCAP ill. az alsóbb szintek hibája (P-ABORT), illetve a TCAP-t használó szint (pl. MAP) hibája (U-ABORT) miatt következett-e be. Itt található meg a tranzakció azonosítását szolgáló OTID/DTID (Originating/Destination Transaction Identifier, kezdeményező/végződő tranzakció azonosító) értékek, melyek szerepe hasonló itt a tranzakciók azonosításában, mint az SLR/DLR szerepe volt az SCCP kapcsolatok azonosításában.

Az üzenettípus meghatározása és a tranzakció azonosítása után következhet a dialógust vezérlő rész. Ez tartalmazza a TCAP verzió azonosítását, a TCAP üzenet által szállított protokoll és annak verziója azonosítását, illetve itt van lehetőség tetszés szerinti, a CCSS7-től független felhasználói információ továbbítására is. Ha ilyet nem akarunk küldeni, illetve az adott szakaszon egyértelműen csak adott verziójú TCAP és csak egy, szintén egyértelmű típusú azt használó protokoll üzenetei fordulhatnak elő, ez a rész kimaradhat.

Ezután következik az az egy vagy több ún. komponens, amely(ek) a TCAP-t használó protokollok üzeneteit szállítják(k). Ez GSM esetén általában a MAP vagy az INAP.

Ezek a komponensek INVOKE, RETURN RESULT LAST/NOT LAST, RETURN ERROR illetve REJECT típusúak lehetnek. Az INVOKE komponens egyművelet elvégzését kéri, mely eredményét a RETURN RESULT LAST, illetve, ha az nem elég, ezt megelőző szükséges számú RETURN RESULT NOT LAST komponens hordozza. A RETURN ERROR komponenst akkor küldik, ha a kérés értelmes volt, de valamilyen ok miatt nem teljesíthető (például a kért adat nincs meg egy adatbázisban), míg a REJECT komponens jelzi azt, ha a kérés valamilyen hibaok miatt értelmezhetetlen.

3.8.3. MAP

A MAP (Mobile Application Part - mobil felhasználói egység) protokoll szolgál a GSM hívások felépítéséhez szükséges adatok megszerzésére, és a mobilitás menedzselésére. A MAP tranzakciókat használ, azaz a TCAP szolgáltatásaira támaszkodik.

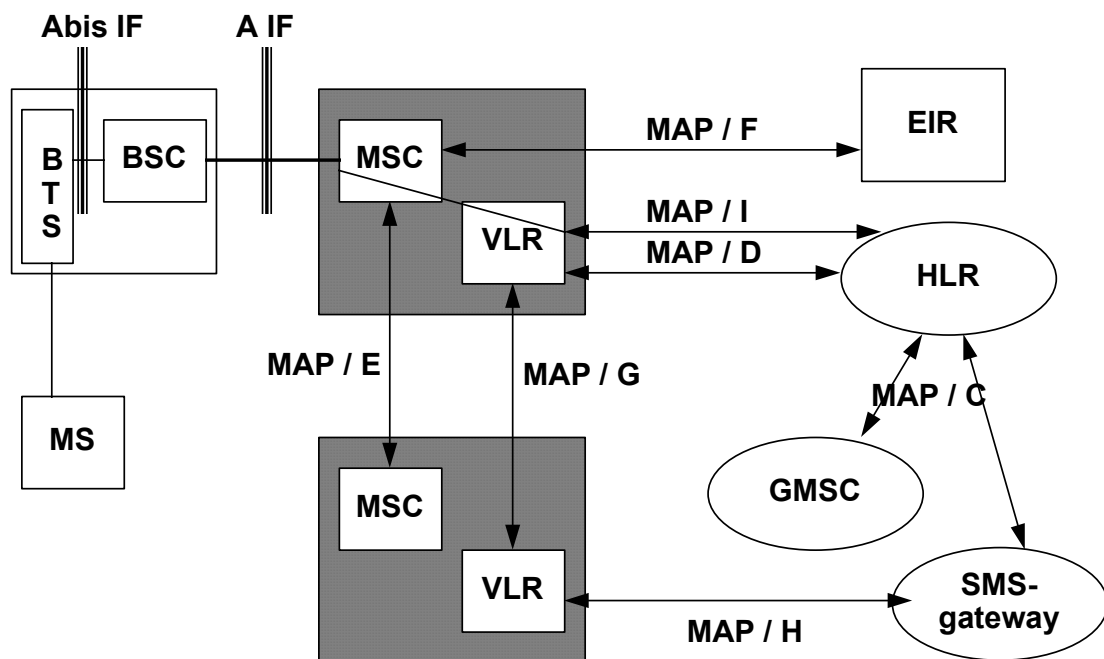
A MAP protokoll funkcionálisan több alprotokollra osztható. A B interfész az MSC és a VLR közötti üzenetváltást írja le, de mivel a gyakorlatban ezeket egy funkcionális egységként valósítják meg, ennek használatától eltekinthetünk.

A C interfész a HLR és a Gateway MSC illetve SMS Gateway közötti üzeneteket specifikálja. Ezek arra szolgálnak, hogy a HLR-ből lekérdezzék az irányítási információt (MSRN).

A D interfész a VLR és a HLR közötti üzeneteket tartalmazza. Ezek főképp azt a célt szolgálják, hogy ha egy mobile készülék egy új VLR körzetbe jelentkezett be,

akkor erről az ő HLR-jét értesítse az új VLR, illetve a HLR a régi VLR-t értesítse arról, hogy onnan az előfizető adatait törölni kell, illetve itt vannak specifikálva azok az üzenetek, amelyekkel a HLR megtudhatja az aktuális VLR-től az irányítási információt (MSRN), amit majd továbbít a C interfészen keresztül a gateway felé.

A D interfész másik feladata a hívás során igénybe vehető kiegészítő szolgáltatások menedzselése (bekapcsolása, kikapcsolása, jelszó küldése stb.). Ezeket az ábrán mint I interfészt tüntettük fel, bár ez nem szabványosított jelölés.



3.8.2. ábra. GSM hálózatok jelzései a szükséges adattovábbítási feladatokhoz

Az E interfész az MSC-k közötti üzenetváltásokat specifikálja. Ezek elsősorban az MSC váltással járó handoverek kezelését, illetve az SMS-ek elküldését szolgálják. Ez utóbbit az ábrán a - nem szabványos - H interfész elnevezéssel jelöltük.

Az F interfész az MSC és az EIR közötti egyetlen műveletet, az IMEI ellenőrzést tartalmazza, míg a G interfész egyetlen művelete arra szolgál, hogy MSC váltás során az új VLR a régitől megkaphassa a mobil készülék azonosító kulcsait.

A MAP protokoll segítségével szerezhethetjük meg azt az információt, hogy a hívott mobil készülék hol található a hálózatban és milyen temporális telefonszámon (MSRN) irányíthatjuk a hívást felé. Magának a hívásnak a felépítése azonban nem a MAP feladata, az az ISUP segítségével történik ugyanolyan lépésekben mint egy fix

hívásé, csak a tárcsázott szám helyett a MAP segítségével megszerzett MSRN-t használjuk.

Itt kell megjegyeznünk, hogy mind a TCAP mind a MAP üzenetek kódolása az ASN.1-ben definiált kódolási szabályok segítségével történik. Sőt, maguknak a MAP műveletek üzeneteinek a specifikálása is az ASN.1 (OPERATION illetve ERROR makró lehetősége) igénybevételével történik.

3.8.4. Az A interfész protokolljai

Az A interfész a mobil hálózatok telefonközpontjainak funkcióit betöltő MSC és az adótornyokat vezérlő BSC között található. Az itt futó protokollokat összefoglaló néven BSSAP-nek (Base Station Subsystem Application Part) hívják. A BSSAP két protokollra oszlik. Az egyik az ún. BSSMAP (Base Station Subsystem Management Application Part), amely a BSC rádióerőforrásainak menedzselésére szolgál. A másik protokoll a DTAP (Direct Transfer Application Part), amely a mobil készülékek és a hívások vezérlésére szolgál, szerepe - a mobilitásból fakadó plusz követelményeket leszámítva - hasonló, mint az ISDN hálózatok DSS1 előfizetői oldali protokolljáé.

A két protokoll a CCSS7 MTP és SCCP protokolljait használja. A DTAP - hiszen egyik feladata a hívások vezérlése - elsősorban az SCCP kapcsolat-orientált, míg a BSSMAP inkább a kapcsolat nélküli szolgáltatásaira támaszkodik.

A két protokoll üzenetei között az SCCP üzenet adatmezőjének az elején lévő diszkriminátor oktett segítségével lehet különbséget tenni.

A BSSMAP kapcsolat nélküli üzenetei döntő részt az ISUP áramkörfelügyeleti üzeneteire hasonlítanak, csak itt rádiócsatornák be ill. kikapcsolására szolgálnak. Egy másik tipikus kapcsolat nélküli BSSMAP üzenet az ún. Paging üzenet, amelyet mobil végződő hívás esetén használnak. A GSM-ben ugyanis a hívásvégződtesítés teljesen eltérő módon történik, mint a fix hálózatokban. Itt a hívott kap egy felszólítást (ez a Paging), hogy hívása érkezett és ezért ő kapcsolódjon a hálózathoz, hasonló módon, mintha ő kezdeményezne hívást (persze ennek során azt jelezni kell, hogy a híváskezdeményezés egy ilyen felszólításra válasz vagy tényleges hívásindítás).

A BSSMAP másik feladata a handoverek menedzselése.

További BSSMAP kapcsolatorientált üzenetek használatosak a hívásfelépítés során. Ilyen minden hívás első üzenete, amely felépíti a jelzéskapcsolatot, valamint a híváshoz rádiócsatorna rendelése és a titkosítás bekapcsolása a fontosabbak.

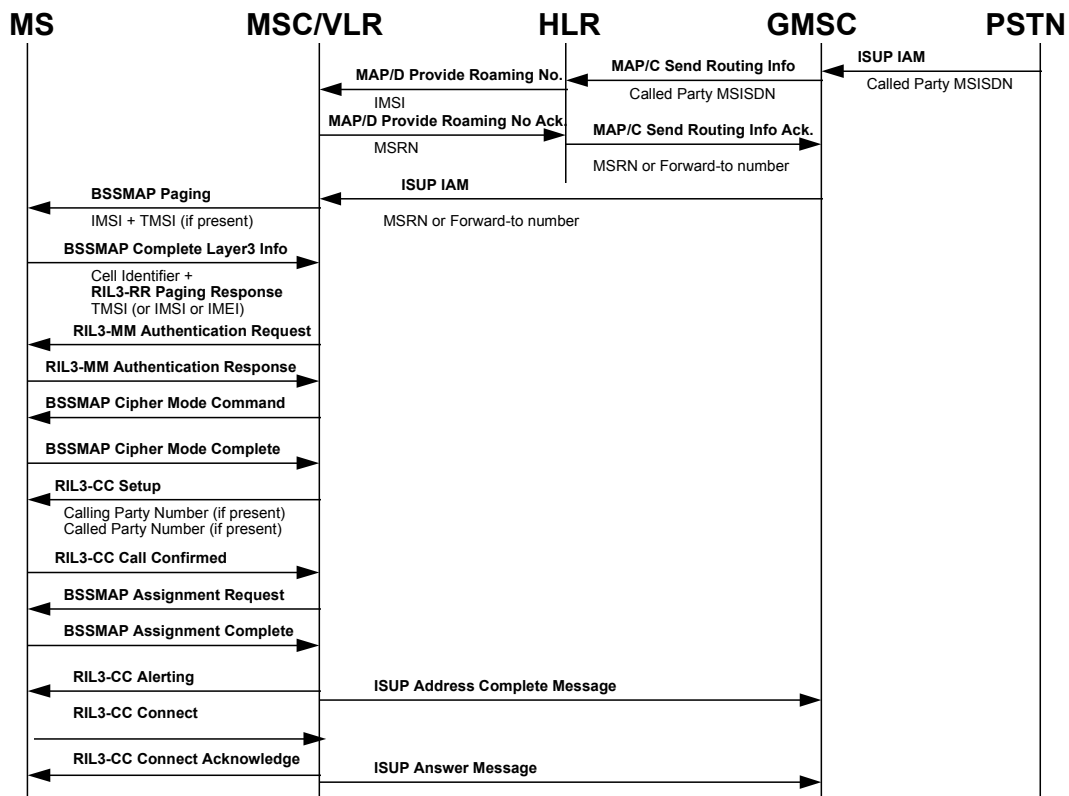
A DTAP üzenetek elsősorban a hívások felépítésében és bontásában játszanak szerepet, a mobil készülék és az MSC közötti parancsváltás lebonyolításával. A DTAP protokoll három részre osztható funkcionálisan. A DSS1 analógiájának tekinthető hívásvezérlés (CC - Call Control) az egyik rész. Ide tartoznak a hívást felépítő Alerting, Setup, Call Confirmed és a bontást vezérlő Disconnect, Release, Release Complete üzenetek. A másik rész a mobilitás vezérlést (MM - Mobility Management) szolgálja. Fontosabb feladatai a Location Update kezdeményezése, a mobil készülék azonosítása, a temporális hívószám (TMSI) hozzárendelése és a készülék kikapcsolásának a jelzése. A harmadik rész az SMS-küldéssel kapcsolatos üzeneteket tartalmazza.

3.8.5. A mobil hívások felépítése

A GSM protokollok áttekintése után nézzük meg, hogy történik egy hívás felépítése. A példában egy fix hálózathoz (PSTN) mobil készülékre irányuló hívás lépéseit követhetjük nyomon.

Első lépésként a hívó hálózathoz egy ISUP IAM üzenet érkezik, - amely a hívott mobil készülék telefonszámát (MSISDN) tartalmazza -, a mobil szolgáltató megfelelő Gateway központjához (GMSC). A Gateway feladata lesz, hogy a hívást ahhoz a mobil központhoz (MSC) irányítsa, ahol az előfizető pillanatnyilag tartózkodik. Ehhez azonban meg kell szerezni azt a temporális telefonszámot (MSRN), amin a készülék elérhető. Ezt a számot az adott MSC rendelte a készülékhez akkor, amikor az bejelentkezett hozzá. Ezért a GMSC a HLR-hez fordul, amely tudja, hogy az előfizető pillanatnyilag melyik MSC körzetben tartózkodik, így meg tudja kérdezni tőle (pontosabban a VLR-jétől) a kívánt MSRN-t, amit majd továbbít a GMSC felé. Ezután a GMSC már fel tudja építeni a hívást a kívánt MSC felé, ISUP szinten a megkapott MSRN-t használva.

A híváskezdeményezés észlelésekor a központ Paging üzenettel felszólítja a mobil készüléket (MS), hogy az kezdeményezzen egy hívást. Erre válaszol a készülék a Complete Layer 3 Info üzenetbe ágyazott Paging Response üzenettel.



3.8.3. ISUP rendszerű ielzésváltás folyamata

(Innen tudja a központ, hogy mi az oka a híváskezdeményezésnek.) Ezt követi a készülék azonosítása, majd átkapcsolnak titkosított üzemmódra. Ezután a hívás felépítése a DSS1 analógiájára megtörténik, a lényeges különbség a megfelelő rádiócsatorna hozzárendelése. Amikor a hívást “felvették” (Connect), a központ értesíti a GMSC-t erről az ISUP ANM üzenet segítségével, ami továbbítódik a hívó központja felé.

Irodalomjegyzék

[3.8.1]. Mouly - Pautet: The GSM System for Mobile Communications ISBN 2-9507190-0-7 1992.

[3.8.2.] ETSI GSM 09.02 Mobile Application Part Specification

[3.8.3] ITU-T Recommendation Q.771 - Q 774 TCAP

[3.8.4] ETSI GSM 04.08 Mobile Radio Interface Layer 3 Specification

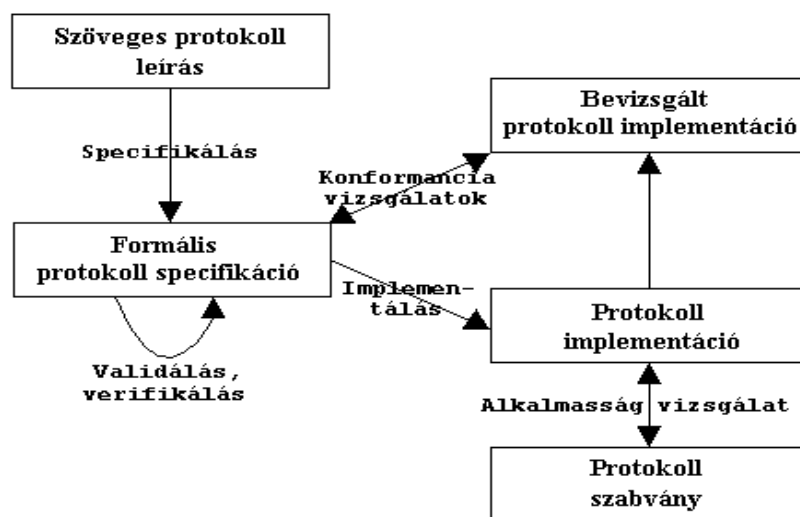
3.9. Távközlési szoftverek és protokollok fejlesztése

Szerző: dr. Adamis Gusztáv

Lektor: Kesselyák Péter

A kommunikációs protokollok előállításának kérdéseivel egy új tudományág, a protokoll technológia (protocol engineering) foglalkozik. Eszerint a protokollok előállításának főbb lépései a következők:

A protokollal szemben támasztott követelmények alapján először elkészül a protokoll szöveges, informális leírása. Azonban ez a leírás többnyire nem kellő részletességű, nem teljesen pontos, illetve vizsgálata nem automatizálható. Ezért szükség van a protokollok formális eszközökkel (*FDT* - Formal Description Technique) történő leírására, *specifikálására*. Ezután ellenőrizni kell, hogy a formális specifikáció nem tartalmaz-e önmagában ellentmondásokat, hibákat (*validálás*), illetve, hogy összhangban van-e a felhasználói igényekkel (*verifikálás*). E tevékenységekben segítségre lehet a *szimuláció* is. Természetesen a feltárt hibák alapján módosítani kell a specifikációt.



3.9.1. ábra. Protokollok előállításának lépései

Az ellenőrzött specifikáció ezek után megvalósítható, *implementálható*. Majd a kész implementációt tesztelni kell, hogy az megfelel-e a specifikációnak

(*konformancia tesztelés*). Egy szabványban sok választható paraméter, mechanizmus van. Az *alkalmasság vizsgálat* (angolul *conformance testing*) egy konkrét paraméterérték- és mechanizmus- választáson alapuló megvalósítást vizsgál. Természetesen a felderített hibák alapján módosítani kell az implementációt. A tesztelésről részletesebben a 3.10. fejezetben olvashatunk. Ebben a fejezetben a specifikálásról és az implementálásról szólnak részletesen.

3.9.1. Protokollok specifikálása

A protokollok specifikációja alapvető jelentőségű, hiszen a protokollkészítési folyamat további lépései mind ezen alapulnak. Ezért az elkészült protokoll minőségét és az előállítás költségét nagyban befolyásolja a specifikálás módja.

A specifikációnak:

- (1) egyértelműnek kell lennie, vagyis biztosítania kell, hogy mindenki ugyanazt értse alatta;
- (2) teljesnek kell lennie, vagyis az összes lehetséges és fontos követelményt tartalmaznia kell;
- (3) lehetőleg támogatnia kell a specifikáció könnyű (automatikusan végezhető) ellenőrizhetőségét;
- (4) megvalósíthatónak, implementálhatónak kell lennie, lehetőleg minél egyszerűbben.

A specifikációs módszerek közül a fenti követelmények teljesülését csak a *formális leírótechnikák* biztosítják. A formális leírótechnika olyan módszer, mely jól definiált szintaktikával és formális módon definiált szemantikával rendelkezik.

A protokollok legáltalánosabban használt modellje a kommunikáló véges automatákon alapul. Minden folyamatnak egy-egy véges automata felel meg, melyek egymás között (csatornákon keresztül) üzeneteket váltanak. A csatornákat hibamentes FIFO sorokként reprezentálják.

Valós protokollok megadására azonban ez a modell a gyakorlatban nem alkalmas az állapotok túlzottan nagy száma miatt. Ezen a problémán segítenek a kiterjesztett véges automaták (EFSM - Extended Finite State Machines), amelyeket a következőképpen jellemezhetünk:

$EFSM = \langle S, I, O, V, P, A, s_0, f(s,i,p) \rangle$, ahol

S : az állapotok véges halmaza

I, O : a bemenő/kimenő üzenetek véges halmaza

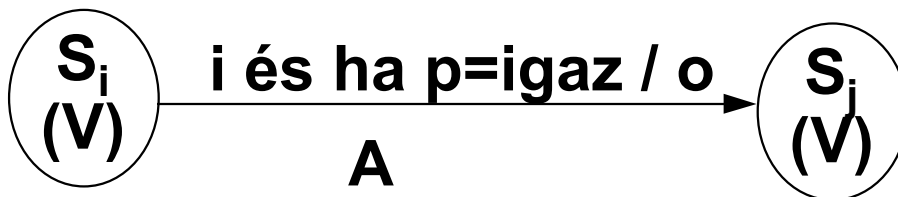
V : a változók véges halmaza

P : a predikátumok (feltételek) véges halmaza

A : az akciók (pl. értékadás egy változónak) véges halmaza

$s_0 \in S$: a kezdőállapot

$f(s,i,p)$: az állapotátmenet függvény, amely azt mondja meg, hogy ha az EFSM az $s \in S$ állapotban van, $i \in I$ üzenetet kap és $p \in P$ predikátum teljesül, akkor melyik $a \in A$ akciót hajtja végre, melyik $o \in O$ kimenő üzenetet generálja és melyik $s' \in S$ állapotba megy át.



3.9.2. ábra. Kiterjesztett véges automata illusztrációja

Ez a modell már megfelelő a - valós méretű - távközlési protokollok leírásához is. A modell elég szemléletes, ezért könnyű a megértése és használata.

A kiterjesztett véges automatákon számtalan FDT eszköz alapul. Ezek közül azonban a távközlésben kiemelkedő jelentőségű a CCITT/ITU-T által kifejlesztett SDL (Specification and Description Language) nyelv.

3.9.1.1 Az SDL nyelv

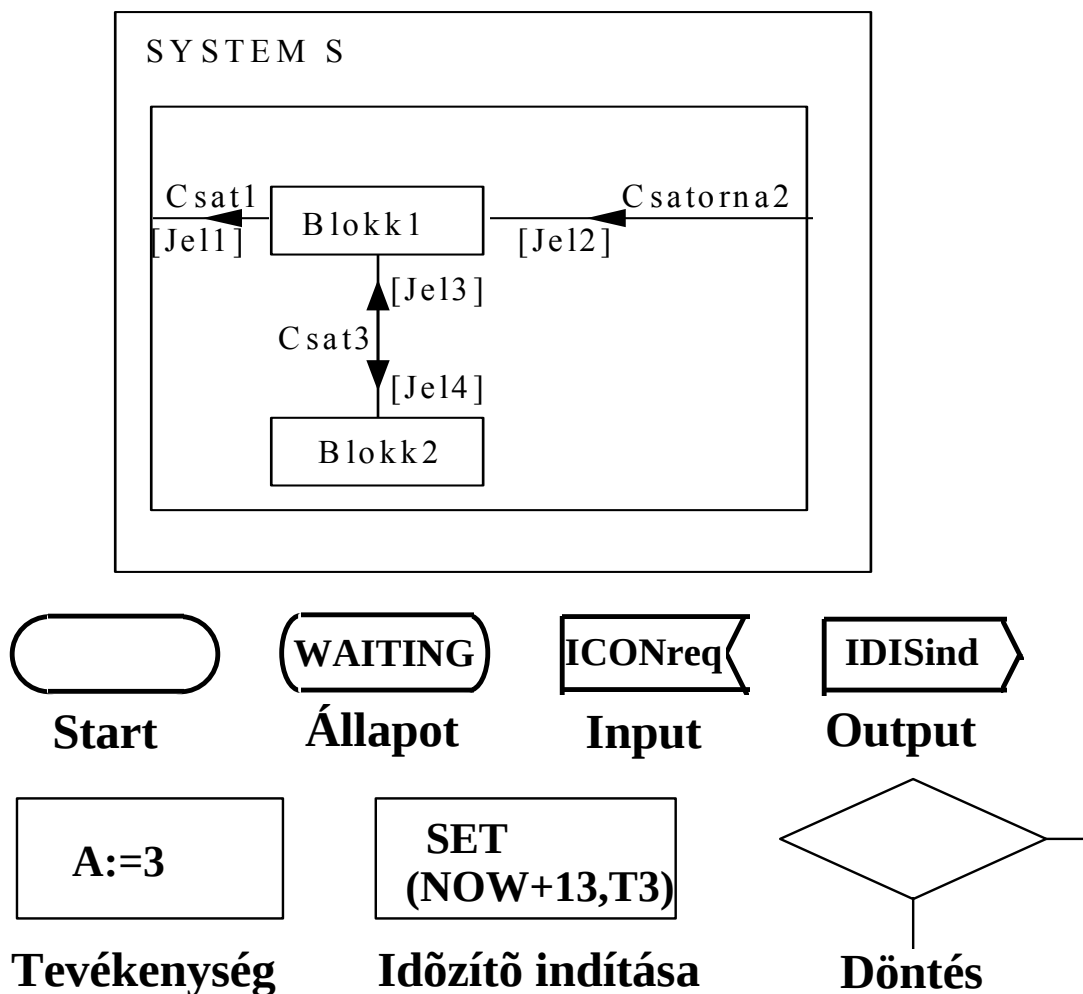
Az SDL kiterjesztett véges automatákon alapuló specifikációs nyelv.

Az SDL-ben a környezettel kommunikáló *rendszer blokkokra* oszlik, melyeket - késleltetéssel rendelkező - *csatornák* kötnek össze, ezeken *jelek* haladhatnak. A blokkok tovább finomíthatóak, végül eljutunk a blokkok működését leíró folyamatokig (*processzekig*), melyek egy-egy kiterjesztett véges automatának felelnek meg. Az egy blokkon belüli processzeket, illetve a processzeket a csatornákkal - késleltetés

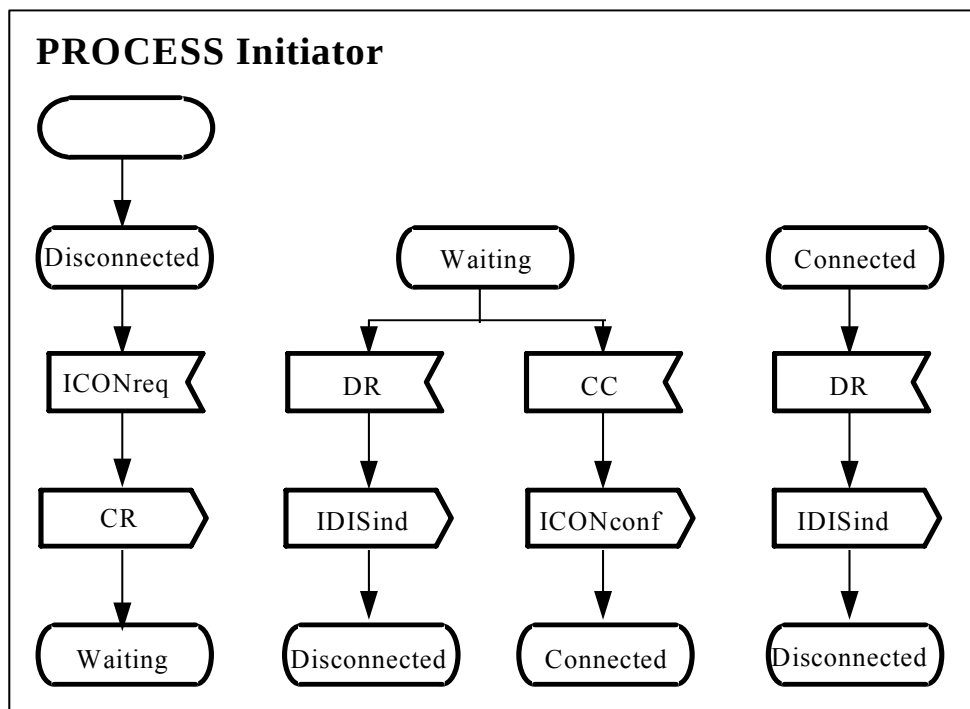
nélküli - *jelutak* kötik össze; ezeken illetve a csatornákon történik a kommunikáció *jelek* segítségével. Minden *processz*hez *tartozik* egy - elvileg végtelen hosszú - FIFO sor, amelyben a beérkezett jelek gyűlnek a feldolgozásukig. Az SDL lehetővé teszi *időzítések* definiálását. Az SDL adatdefiníciós része *absztrakt adattípus* (ADT) alapú. Az SDL újabb verzióiban sok más kiegészítés mellett több, az objektum-orientált programozási nyelvekben használatos lehetőséget is bevezettek.

Az SDL az egyetlen a szabványos FDT nyelvek közül, amelynek van grafikus változata is, sőt tulajdonképpen *először* volt grafikus változata (SDL/GR) és csak annak nagy sikere után kezdték el a programnyelvi verzió (SDL/PR) kifejlesztését.

Az SDL a gyakorlati élethez nagyon közel áll, különösen a grafikus változata nagyon szemléletes. A nyelv nagy előnye, hogy a programtervezés teljes spektrumát átfogja az egész magas blokkszintű specifikációtól az egész implementáció közeli leírásig.



3.9.3. ábra. Néhány fontos SDL szimbólum



3.9.4. ábra. Egy SDL processz leírása (illusztráció)

3.9.1.2 Adatszerkezetek specifikálása

Az SDL az adatszerkezetek definiálására egy absztrakt adattípus koncepción alapuló leírásmódot használ. Ezzel a módszerrel nagyon pontosan definiálhatók az adattípusok, a konkrét megvalósításuktól jórészt független módon. Ez jelenti a módszer előnyét, de egyben hátrányát is, ugyanis ez az absztrakt leírásmód nagyon megnehezíti az automatikus úton történő implementálást, illetve más nyelvektől nagyrészt eltérő gondolkodásmódot követel.

Ezért lehetőség van SDL-ben adatszerkezetek leírására más módon is, az ASN.1 (Abstract Syntax Notation One) nyelv használatával. Ez a nyelv sokban hasonlít a magasszintű programozási nyelvek adatdefiníciós képességeire. Használatának másik előnye, hogy az ASN.1 a TTCN tesztspecifikációs nyelvben is használható, azaz teszt sorozatok készítésekor nem kell az adatszerkezeteket kétszer, két különböző módon elkészíteni.

Az ASN.1 nyelvet eredetileg applikációs rétegbeli protokollok adatkommunikációjának egyértelmű és megvalósítás-független leírására találták ki. Az applikációs rétegbeli protokollok jellemzője általában az, hogy bár maga a protokoll automata egyszerű (néhány állapotú), de a küldhető üzenetek rendkívül

összetettek, sok, sokféle típusú, sokszor opcionális vagy alapértelmezett értékű mezőkből állnak.

Az adatkommunikáció leírása két fő dolgot jelent:

- Legyen lehetőség a (sokszor nagyon bonyolult) üzenetformátum egyértelmű leírására.
- Legyen lehetőség olyan kódolási szabály definiálására, amelynek segítségével egyértelműen eldönthető, hogy mely üzenetelemek mely értékekkel kerültek továbbításra (például, hogy az opcionális mezők közül melyek jöttek ténylegesen).

Ennek megfelelően az ASN.1 is két fő részből áll. Egyrészt egy adattípus-definíciót lehetővé tévő, másrészt egy kódolási szabályt definiáló részből.

Az ASN.1 adattípus definiáló része hasonlít a legtöbb programozási nyelvéhez. Egyrészt rendelkezik beépített típusokkal, másrészt lehetőséget biztosít arra, hogy egyszerűbb elemekből összetett típusokat hozzunk létre. Ez az ún. konstrukciós képesség azonban a legtöbb programozási nyelvéhez képest sokkal flexibilisebb. Ugyanakkor az ASN.1 nem programozási nyelv, tehát végrehajtható utasításokkal (ciklus, feltételes, stb.) nem rendelkezik. Az ASN. 1 nem definiálja a beépített típusok megvalósítását, tehát például az INTEGER értéktartománya nincs korlátozva. Ezek a kérdések a kódolás során dőlnek el.

Az egyszerű típusokból "átnevezéssel" hozhatunk létre új (egyszerű) típusokat, illetve strukturált típusok létrehozására a következő konstrukciós szabályok állnak rendelkezésre: SEQUENCE, SET, SEQUENCE OF, SET OF, CHOICE.

A SEQUENCE különböző típusú elemekből álló rekordot, míg a SET különböző típusú elemekből álló halmazt hoz létre. A kettő között a fő különbség az, hogy az első esetben az elemek sorrendje rögzített, míg a másodikban tetszőleges. A SEQUENCE és a SEQUENCE OF illetve a SET és a SET OF között az a különbség, hogy az utóbbiak azonos típusú, sorbarendezett elemekből álló rekordok (azaz tömbök) illetve halmazok. A CHOICE konstrukciós szabály segítségével a C-beli unionokhoz hasonló egységet lehet létrehozni, azaz olyan "rekordot", amely a definiált mezői közül egyszerre csak egyet tartalmaz.

A típusokból korlátozásokkal altípusokat lehet létrehozni. Néhány példa:

- `Telefon_szam:= IA5String(FROM('1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'|'0'|'*'|'#'))`

- Tiznel_kisebb_szamok:= INTEGER(0..9)
- Tomb_1_64_elemmel:= SEQUENCE (SIZE (1..64)) OF INTEGER
- Tomb_pontosan_64_elemmel:= SEQUENCE (SIZE (64..64)) OF INTEGER
- Kis_primek:= (2|3|5|7)

Az adatok specifikálásakor sokszor a típusuk megadása nem elegendő ahhoz, hogy továbbításuk során egyértelműen azonosíthatók legyenek. Gondoljunk például arra, hogy van egy rekordunk, két opcionális, egyforma típusú mezővel, amelyek közül az adott szituációban csak az egyik mezőt kell küldeni. Azért, hogy az, hogy melyiket küldtük a vételi oldalon is egyértelműen azonosítható legyen, lehetőség van egyedi típusazonosítók (ún. toldalékok, TAG-ek) definiálására. Ezek segítségével lehet az azonos típusú mezőkből különböző típusú mezőket készíteni.

Koordinatak:= SEQUENCE{

- x [0] INTEGER OPTIONAL,
- y [1] INTEGER OPTIONAL}

A példában két opcionális, INTEGER típusú mezőnk van, és úgy lehet azonosítani őket, hogy az x "[0]-ás típusú INTEGER", míg az y "[1]-es típusú INTEGER". A [] zárójelben szereplő szám a toldalék - TAG.

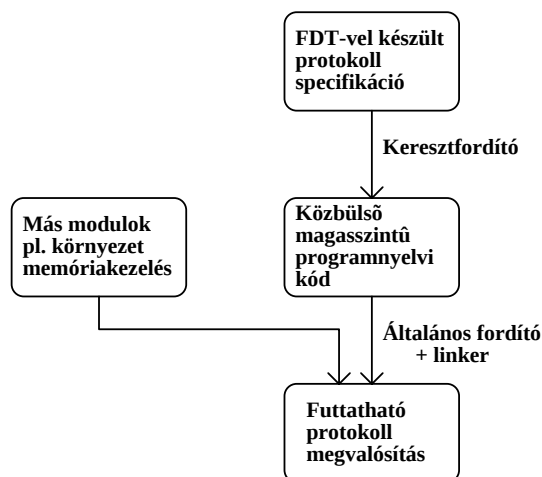
Az ASN.1-ben egyértelmű kódolási szabályokat is definiáltak. Ezek lényege, hogy minden adatküldés során az adat típusát, hosszát és értékét is elküldjük.

3.9.2. Protokollok implementálása

A protokoll technológia egyik kulcskérdése a protokollok implementálása, hiszen a fejlesztés célja, végterméke a megtervezett, specifikált, validált protokoll működő formátumra hozatala, előállítása. Ezt a folyamatot nevezzük implementációnak, vagyis amikor létrehozuk azt a szoftver (vagy hardver) megvalósítást, mely a specifikációban megfogalmazott összes követelményt teljesíti.

A protokollok implementálása a specifikálással kezdődik. A kész specifikáció azonban - bár döntő jelentőségű, de - nem elég az implementálás szempontjából.

A protokoll specifikáció ugyanis - többé-kevésbé - elvonatkoztat a megvalósítás konkrét körülményeitől, a protokoll konkrét fizikai környezetétől. (Gondoljunk csak arra, hogy ugyanazt a protokollt két teljesen más fizikai felépítésű és operációs rendszerű számítógépen akarjuk megvalósítani.)



Ennek megfelelően a protokollok megvalósítása során kapott eljárások két többé-kevésbé diszjunkt halmazt alkotnak.

Az első halmaz tartalmazza azokat a tulajdonságokat, amelyek *közvetlenül* a



protokoll specifikációból adódnak.

A második halmazban találhatók azok az eljárások, melyek a kapcsolódást biztosítják a protokoll és a célgép operációs rendszere között. Ezek között az eljárások között elsősorban a következők találhatók: memória kezelés, sorok kezelése, valós idejű (real-time) támogatás, taskok közötti kommunikáció vezérlése, ütemezési eljárások, esemény feldolgozó eljárások, hibakezelő eljárások. Ezt a részt döntő mértékben a célgép lehetőségei befolyásolják. Természetesen ezeket az eljárásokat a protokoll-rutinok használják, vagyis a kettő között egy interfésznek kell lennie.

3.9.5. ábra. A protokollok megvalósítása során kapott eljárások felosztása

3.9.6. ábra. Az automatikus protokoll implementálás általánosan alkalmazott módszerének elve

A protokollok implementálásának két fő módja van.

	AD-HOC	FÉLAUTOMATIKUS
	<i>megvalósítással készült protokoll program esetén</i>	
Futási sebesség:	magas	alacsony
Kódméret:	kicsi	nagyobb (2x vagy több)
Program szerkezete:	bonyolult	tagolt, egyszerű
Modularitás:	ált. csekély	magas fokú
Hordozhatóság:	csekély	magas fokú
Fenntartás, bővítés:	nehéz	egyszerű
Hibakeresés, -javítás:	bonyolult, időigényes	egyszerű
Fejlesztési idő:	hosszú	rövid
Konformancia a specifikációval:	nem biztosított	biztosított

3.9.1. táblázat. Az ad-hoc és a félautomatikus eljárással készült protokollok összehasonlítása

A hagyományos az ún. *ad-hoc* módszer, azaz minden egyes protokollnak, mint önálló szoftvernek a kifejlesztése. Ilyenkor a protokoll specifikációt teljesítő és a környezettel való kapcsolatot biztosító eljárások egy egységet képeznek, az elkészült realizáció messzemenően figyelembe veszi az adott protokoll, illetve az adott környezet igényeit. Ez jelenti ennek a módszernek a legfőbb előnyét, vagyis a *kis méretet* és a *nagy sebességet*. A módszer hátránya viszont, hogy nagyon *munkaigényes*, az elkészült program *kevésbé hibátűrő, nehezen javítható és nem hordozható*.

Ha a protokoll specifikáció FDT-ben készül el, akkor lehetőség van az implementálás során számítógép segítségét igénybe venni, azaz a specifikációs nyelven leírt protokollt átfordítani egy programozási nyelvre. Ez - a célrendszerrel függő részek miatt - nem lehet teljesen automatikus, hiszen ezeket a részeket is hozzá kell szerkeszteni a protokollspecifikációból automatikusan generált rutinokhoz. Ezt megkönnyítendő általában a kétlépéses implementációt használják, ahogy az a 4. ábrán látható.

A két módszer összehasonlítása az 1. táblázatban látható.

Irodalomjegyzék

[3.9.1] ITU-T Recommendation Z.100: Formal description techniques (FDT) - Specification and Description Language (SDL), 1999.

[3.9.2] ITU-T Recommendation Z.105: SDL combined with ASN.1 (SDL/ASN.1), 1994.

[3.9.3] ITU-T Recommendation X.680 Abstract Syntax Notation One, 1994.

[3.9.4] ETSI EG 202 106 (1999): Methods for Testing and Specification (MTS): „Guidelines for the use of formal SDL as a descriptive tool”.

[3.9.5] ETSI EG 202 107 (1999): Methods for Testing and Specification (MTS): „Planning for validation and testing in the standards-making process”.

[3.9.6] ETSI TR 101 680 (1999): Methods for Testing and Specification (MTS): „A harmonized integration of ASN.1, TTCN and SDL”.

3.10. Protokollok és szoftverek tesztelése

Szerző: dr. Dibuz Sarolta

Lektor: dr. Tarnay Katalin

3.10.1. A szoftver tesztelés alapjai

A távközlési szoftverek fejlesztési költségeinek több mint 50 százalékát a tesztelési költségek teszik ki. A tesztelés célja hogy minél több hibáját megtaláljuk a szoftvernek. Azt azonban soha nem lehet bebizonyítani, hogy egy szoftver teljesen hibamentes. Tesztelés során bemenetekkel gerjesztjük a szoftvert és figyeljük, hogy a szoftver válaszai helyesek-e. A helyes válaszok a szoftver specifikációja alapján határozhatók meg, vagy ennek hiányában a felhasználó az elvárt viselkedés alapján dönti el hogy jó volt-e a válasz. A tesztelési követelmények jelentősen eltérhetnek a használt fejlesztési módszerek és technológiák esetében.

A tesztelés három lépésből áll:

- a teszt tervezése,
- a tesztelés pontos definíciója és
- a teszt végrehajtása.

A teszt terv részletesen meghatározza a tesztelés célját, a használt módszert, az elvégzendő feladatokat, az erőforrásokat, az összefüggéseket és a lehetséges veszélyforrásokat.

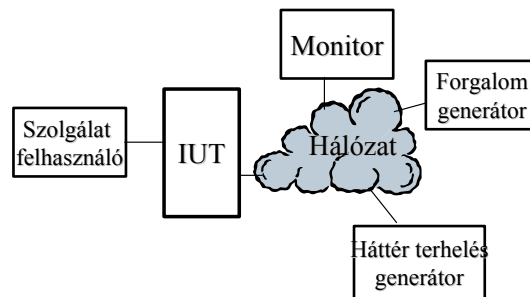
A teszt definíciója azt határozza meg, hogy milyen tesztekkel kell elvégezni és milyen céllal. A teszt definíció elkészítése a teszt terv végrehajtásának első lépése, ezután következik a teszt végrehajtása vagy automatikus tesztelés esetén a teszt sorozat elkészítése. A teszt definíció egy megismételhető, pontos meghatározását tartalmazza a tesztelés folyamatának.

Az automatikusan végrehajtható tesztsorozat elkészítését követi a tesztelés, ami lehet fekete-doboz tesztelés: installálás, kompatibilitás, funkcionalitás és megbízhatóság ellenőrzésére, vagy fehér-doboz tesztelés: amikor a szoftver kódjába is bele lehet nézni.

Vannak olyan eszközök, amik támogatják az automatikusan végrehajtható teszt sorozatok elkészítését a teszt terv és teszt definíció alapján. Egy szoftvert számos szempontból lehet tesztelni attól függően, hogy a szoftver mely képességeit szeretnénk ellenőrizni.

Funkcionális tesztelés: Annak ellenőrzése, hogy egy alkalmazás megfelel-e a saját specifikációjának, és hogy helyesen végzi minden funkcióját. Ez olyan tesztek sorozatából áll, melyek a szoftvert tulajdonságról tulajdonságra végig ellenőrzi úgy, hogy számos helyes és helytelen input adattal gerjesztik. Ide tartozik a termékek felhasználói interfészének ellenőrzése, adatbázis menedzsment, biztonságosság, installáció, hálózati viselkedés. A funkcionális tesztelés mind kézzel, mind automatikusan végrehajtható.

Teljesítmény tesztelés: Annak vizsgálata, hogy mennyire könnyen bővíthető, skálázható egy szoftver, vagy hogy más termékekkel (szerverek, middleware-ek) egy környezetbe helyezve milyen teljesítmény tulajdonságai vannak. Elsősorban a szoftverek teljesítmény szempontjából szűk keresztmetszetű részeinek meghatározására szolgál. Általában automatikusan végrehajtható teszt sorozatot használnak erre a célra, hiszen így lehet szimulálni normál, kiugró és kivételesen nagy terheléseket a tesztelés során. Fontos, hogy teljesítmény tesztelés során megcélzott értékeket, metrikát határozzanak meg, hiszen így azonosíthatók a szűk keresztmetszetek. A teljesítmény tesztelés egy lehetséges elrendezésére mutat példát a 3.10.1 ábra.



3.10.1. ábra. Egy elrendezés teljesítmény tesztelésre

Stressz tesztek: Ekkor olyan körülmények között tesztelik a szoftvert, ahol a várhatónál jóval nagyobb terhelés éri, és figyelik, hogy hogyan viselkedik ilyen esetben.

Regressziós tesztek: Célja hasonló a funkcionális teszteléshez, arra szolgál, hogy megismételjük a tesztet egy szoftver új verzióin. Ezzel ellenőrzik, hogy a bejelentett szoftver hibákat kijavították-e, és hogy ennek során nem hoztak-e be újabb hibákat. Bár regressziós tesztelés kézzel is végezhető, automatikus teszttel jelentősen csökkenthető a befektetett idő és energia.

Konformancia tesztelés: Azt ellenőrzi, hogy egy termék megfelel-e a rá vonatkozó szabványok előírásainak, azaz hogy a termék megvalósítja-e azt a portabilitást, együttműködő képességet és kompatibilitást, amit egy szabvány előír.

Együttműködés tesztelés: Egy adott specifikáció két különböző megvalósításának együttműködési képességeit ellenőrzi. Mind a szabvány specifikációk nem konzisztens részeit, mind a megvalósítások kompatibilitási problémáit felfedi. Általában konformancia tesztelés után végzik, mert a konformancia teszteléssel csökkenthető a fel nem tárt együttműködés problémák száma.

3.10.2. Protokollok tesztelése

A protokollok speciális szoftverek, melyek a távközlési szotverek kommunikációt végző funkcióját valósítják meg. A protokollok a hálózat két elemének a kommunikációját határozzák meg. Számos protokoll létezik, mind más és más hálózati kommunikációs funkciók egvalósítására.

A protokollok definiálják:

- azon üzenetek szintaxisát, melyet a kommunikáció során váltanak az elemek,
- a kommunikáció szemantikáját, azaz hogy milyen választ kell küldeni egy beérkező üzenetre,
- valamint az időbeli viselkedést, például időzítők lejáratí idejét.

Láttuk a 3.9-es fejezetben hogy a protokollok előállításiának megvan az életciklusa, a konformancia tesztelés fontos szerepet játszik ebben. Ugyanazok a protokoll modellek használhatók a protokoll tesztelés alapául, amelyeket a protokollok specifikálására is használnak. Protokoll megvalósítások ellenőrzésére a következő módszereket használják leggyakrabban:

- funkcionális tesztelés
- konformancia tesztelés
- együttműködés tesztelés, és
- teljesítmény tesztelés.

A protokoll tesztelési módszerek általában fekete-doboz technikák, azaz a megvalósítást egy fekete dobozként kezelik, amelynek üzeneteket küldünk és figyeljük az ezekre érkező válaszait. A protokoll megvalósítás állapotai és belső megvalósítása nem látható.

A protokollok a referencia modell egy-egy rétegében helyezkednek el. Általában feltételezzük, hogy tesztelés során hozzáférünk a protokoll megvalósítás legalább egyik réteghatárához tesztelés közben. Előfordulhat olyan helyzet, hogy a protokoll egyik réteghatára sem hozzáférhető, ekkor beszélünk beágyazott tesztelésről. Ilyenkor a tesztelés nehezebb feladat, hiszen minden, amit meg tudunk figyelni függ egy másik protokoll viselkedésétől.

3.10.3. A konformancia tesztelés és teszt szabványosítás

A protokollokat szabványként definiálják. Szabványok definiálnak ezenkívül protokoll tesztelési módszert és teszt sorozatokat, amelyekkel elsősorban lehet tesztelni. Soha nem lehet 100 %-osan tesztelni, mindig egy megfelelő egyensúlyt kell találni a teszt költsége és a teszt lefedés között. A teszt szabványok azt definiálják, hogy hogyan kell ellenőrizni azt, hogy a protokoll interfészek mennyire valósítják meg a szabványokban előírtakat. A teszt követelmények, teszt eljárások valamint tesztsorozatok szabványosítása lehetővé teszi, hogy összehasonlítható és megismételhető teszt eredményeket kapjunk, hiszen minden gyártó ugyanazt a teszt módszert és tesztsorozatot használja termékei ellenőrzésére. Ennek nagyon fontos szerepe van akkor, amikor különböző gyártók termékeinek együttműködését kell biztosítani.

Az ISO 9646-os szabványa definiálja a konformancia tesztelés módszertanát, amit az ITU-T átvett az X.290-296 szabványsorozatban. Ezek a szabványok meghatározzák:

- A teszt architektúráját, azaz a konformancia tesztelés absztrakt teszt módszereit.
- A konformancia tesztelés folyamán készülő dokumentumok formátumát.
- A TTCN (Tree and Tabular Combined Notation) jelölésrendszert, mellyel a teszt készletek írhatók le.
- A teszt eszközökkel szemben támasztott követelményeket.
- A teszt végrehajtásának menetét, a teszt kampányt.

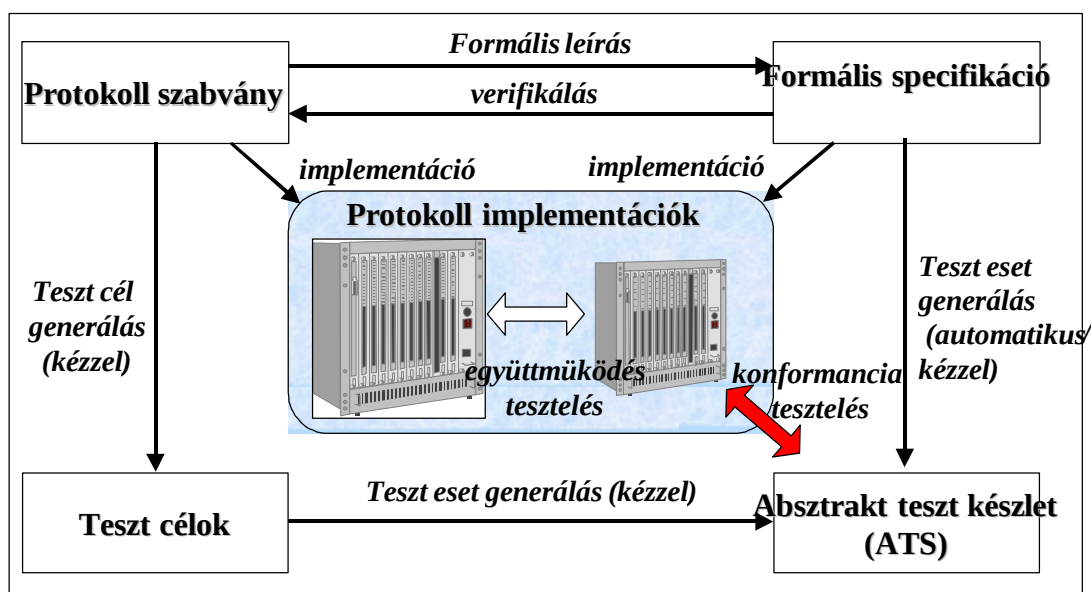
A legtöbb tesztkészletet és teszt dokumentumot az ETSI (European Telecommunication Standardization Institute) szabványosítja. A tesztelés leíró nyelvét azért szabványosították, hogy különböző teszt eszközökkel lehessen a tesztet végrehajtani. Az ETSI a protokoll szabványok mellett ezek tesztkészleteit is szabványosítja TTCN-ben.

A konformancia tesztelés azt ellenőrzi, hogy egy protokoll megvalósítás megfelel-e a szabványnak (3.10.1 ábra). A protokoll szabvány meghatározza a protokoll tulajdonságait és képességeit, ezek a konformancia követelmények, melyek

megvalósulását ellenőrizzük a tesztelés során. Mivel a különböző gyártók azonos szabványt megvalósító termékei ugyanannak a protokoll szabványnak kell

megfeleljenek, így a konformancia tesztelés növeli annak a valószínűségét, hogy különböző termékek együtt tudjanak működni.

Nemcsak a konformancia tesztelés módszerét szabványosították, hanem a tesztkészleteket is. Miért olyan fontos szabványosítani a konformancia tesztelést? Mert így lehetséges, hogy a különböző gyártók ugyanazt a tesztmódszert és teszt sorozatot használják. Így annak ellenére, hogy általában más tesztkörnyezetet használnak (hogy ne legyen túlzottan meg legyen kötve a gyártó keze) és más laboratóriumokban hajtják végre a tesztek a teszt eredmények mégis összehasonlíthatóak lesznek az azonos protokollokat megvalósító termékekre.

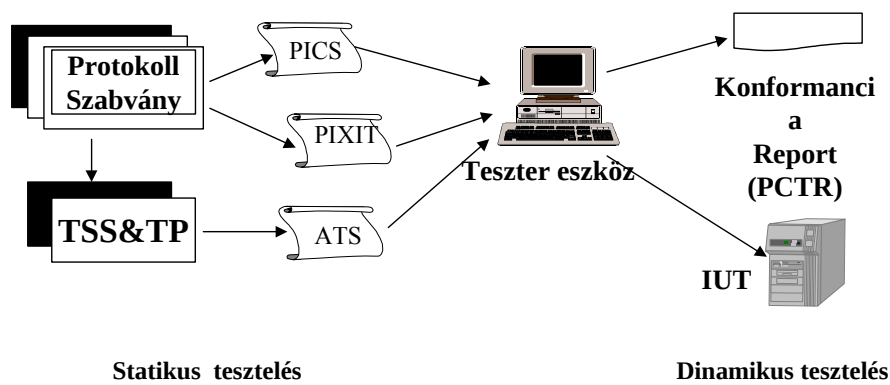


3.10.2. ábra. A protokoll fejlesztés életciklusa és a tesztelés

A tesztkészlet megírása előtt teszt célokat határoznak meg a protokoll szabvány alapján (3.10.2. ábra). Ezek jól definiált szöveges megfogalmazásai egy-egy tesztelési lépésnek, melyek néhány konformancia követelményt fednek le. Az ETSI-ben a TSS&TP (Test Purpose and Test Suite Structure) szabványos dokumentumban gyűjtik össze egy protokollra vonatkozó teszt célokat. Ezeket MSC (Message Sequence Chart) formátumban is meg lehet adni.

A teszt eseteket a teszt célok alapján írják meg, minden teszt célra egy teszt eset készül. A teszt eset valójában a teszt célnak egy speciális program nyelven, a teszt jelölésrendszerben megírt változata. A konformancia tesztelés szabványa tartalmazza a TTCN (Tree and Tabular Combined Notation) teszt leíró nyelv (jelölésrendszer) definícióját. A tesztek származtatása kézzel és automatikusan is történhet. A tesztek automatikus származtatása tesztsorozat generáló módszerekkel történik. A tesztesetek valósítják meg a teszt célokat, ezek összessége adja az absztrakt teszt készletet (Abstract Test Suite, ATS). Az ATS is szabványosított azért, hogy összehasonlítható és megismételhető teszt eredményeket kaphassunk, ezeket verdiktnek nevezik. Különböző gyártók használhatják ugyanazt az absztrakt teszt készletet az általuk gyártott protokoll megvalósítás tesztelésére. Ha a protokoll megvalósítás minden teszt esetre helyesen működik, (azaz kielégít minden konformancia követelményt), akkor megfelel a protokoll szabvány előírásainak. A teszt készletet azért hívják absztraktnak, mert nem függ a protokoll megvalósításától, ezért is használható bármely gyártó által. Ez azt is jelenti, hogy az ATS-t nem lehet végrehajtani mielőtt paramétereinek értéket nem adtunk és le nem fordítottuk egy végrehajtható programra. Ideális esetben az ATS teljes, azaz egy implementáció tesztelése akkor és csak akkor ad helyes eredményt az adott tesztkészlettel, ha az implementáció valóban megfelelően működik. Általában nem lehet teljes tesztkészletet írni véges számú testesetből, így azt várjuk el a tesztkészlettől, hogy megbízható legyen, azaz minden olyan implementáció amely nem ad helyes eredményt teszteléskor valóban hibás legyen. Egy megbízható tesztkészlettől azt is elvárjuk, hogy minden implementáció melynek teszteléskor helyes eredményt kaptunk valóban megfeleljen a protokoll szabványnak.

További dokumentumok segítik, hogy különböző implementációk teszteléskor összehasonlítható és megismételhető eredményeket kapjunk, ezek a PICS, PIXIT, PCTR proformák (3.10.3. ábra). Ezeket a proformákat töltik ki adatokkal tesztelés előtt és közben, így kapjuk a PICS, PIXIT és PCTR dokumentumokat.



3.10.3. ábra. A konformancia tesztlés dokumentumai

A PICS (Protocol Implementation Conformance Statement) összefoglalja egy adott protokoll implementációban megvalósított protokoll tulajdonságokat és paraméter értékeket, pl. időzítők lejáratí értékeit). A protokoll szabványok, melyeket az ETSI, ITU-T készíít általában tartalmazzák a PICS proformát, ami egy üres PICS dokumentum. A proforma a protokoll tulajdonságainak táblázatos formában megadott összefoglalása. A PICS minden tulajdonságra egy referenciát tartalmaz a protokoll szabvány megfelelő fejezetére, ahol az adott tulajdonságot részletesen leírják. Azt is tartalmazza a táblázat, hogy az adott tulajdonságot kötelező-e támogatni, vagy csak opcionális. A protokollt megvalósító feladata, hogy kitöltse a PICS-et "igen" "és nem" állításokkal attól függően, hogy az adott funkciót megvalósította-e vagy sem. A konformancia tesztlés folyamán kiválasztják azokat a teszt sorozatokat a PICS alapján, amiket végre fognak hajtani, hiszen a PICS mutatja meg, hogy mely tulajdonságokat valósítottak meg.

Az absztrakt tesztkészlet paramétereinek konkrét értékekkel való feltöltését a PIXIT (Protocol Implementation eXtra Information for Testing) dokumentum alapján végzik tesztlés előtt. Ebben foglalják össze azon paraméterek értékét, melyek egy konkrét implementációtól függenek. Ez a dokumentum tartalmaz olyan információkat a tesztlésről, amelyek ahhoz szükségesek, hogy a tesztlés megismétlélhető legyen később.

A tesztlés eredményét a PCTR (Protocol Conformance Test Report) nevű dokumentum tartalmazza. Az absztrakt tesztkészlet minden tesztetere megadja,

hogyan elvégezték-e a konkrét tesztelés során, és ha igen, akkor azt is, hogy mi volt az eredmény: “pass, fail vagy inconclusive”, azaz “ sikeres, sikertelen vagy nem eldönthető”. Ha egy rendszerben levő több protokoll megvalósítást is teszteltek, akkor egy SCTR (System Conformance Test Report) nevű dokumentum is készül. Ezen dokumentumok elkészítése az utolsó lépése a teszt laboratóriumban végzett tesztelésnek. A konformanciáról, szabványnak való megfelelésről szóló bizonyítvány kiállítása nem a teszt laboratórium feladata, hanem egy hivatalos szerv végzi természetesen a teszt laboratórium tesztelési eredményei alapján. A hivatalos szerv rendszeresen ellenőrzi, hogy a teszt laboratóriumban megfelelő módon végzik-e a tesztelést azért, hogy megfelelő teszteredmények alapján adhassák ki a bizonyítványokat.

Egy konkrét megvalósítás konformancia teszt laborban végzett tesztelését teszt kampánynak nevezzük, mely a következő lépésekből áll:

- Felkészülés a tesztelésre,
- Teszt végrehajtás, és
- Teszt eredmények elemzése.

A teszt felkészülés a PICS és PIXIT dokumentumok kitöltését jelenti elsősorban. A megvalósító tölti ki a PICS-et, a teszt labor a PIXIT-et, bár ehhez is kap adatokat a megvalósítótól. A teszt végrehajtása során először végignézik a PICS-et, hogy valamennyi kötelezőnek tekintett protokoll funkciót megvalósítottak-e a protokollban, és hogy az opcionális és feltételes tulajdonságok is konzisztensen lettek-e megvalósítva. Ez a folyamat a statikus konformancia vizsgálat (static conformance review). Ezután hajtják csak végre a tesztkészletet. A tesztelés ezen részét nevezik dinamikus tesztelésnek, hiszen ekkor történik a tényleges üzenetküldés a teszter és a tesztelt objektum (Implementation Under Test, IUT) között (3.10.3.ábra). A teszt reportok alapján látják el a tesztelt terméket hivatalos bizonyítvánnyal.

3.10.4. Teszt készleteket leíró nyelvek

TTCN-2

A TTCN (Tree and Tabular Combined Notation) szabványát az ISO (International Organization for Standardization) adta ki először 1992-ben. Azóta

egyre több távközlési szoftver gyártó cég, operátor és teszt laboratórium használja ezt a nyelvet. Számos teszt eszköz támogatja ezen a nyelven írt teszt készletek végrehajtását: editotok, compiler-ek, szintaktikus ellenőrzők és szimulátorok. A TTCN-t elsősorban konformancia tesztelésre használják, bár készült benne funkcionális tesztelést végző tesztkészlet is. A TTCN-2 nevet 2000 óta viseli, amikor a nyelv új verziója a TTCN-3 megjelent. A TTCN-nek két megjelenési formája van: az egyik egy emberek számára is olvasható TTCN.GR a másik a számítógépek által végrehajtható TTCN.MP formátum. A TTCN.GR tesztkészlet számos táblázatból áll. A táblázatok közötti eligazodáshoz feltétlenül szükséges egy editor.

A TTCN-ben írt tesztkészlet öt részből áll:

- Az áttekintő rész tartalmazza a tesztkészlet hierarchikus csoport struktúráját, valamint a teszt esetek, teszt lépések, comment-ek és default viselkedések listáját.
- Az Imports rész sorolja fel az újrafelhasználásra importálható objektumokat. A tesztkészlet modulokból épül fel, és az objektumok más modulokban is felhasználhatók, mint amelyben definiálva lettek.
- Deklarációs rész, melyben adattípusok és más a tesztkészletben használt adatobjektumok kerülnek definiálásra. Az adatobjektumok ASN.1- ben is megadhatók.
- Korlátok rész, itt határozzák meg a küldésben és vételben szereplő adatobjektumok konkrét értékeit.
- Dinamikus rész, mely a teszt célokat megvalósító teszt esetek leírását tartalmazza. Ebben a részben található a teszt lépések, melyek a teszt esetekben több helyek előforduló közös részeit tartalmazzák. A dinamikus részhez tartozik a default könyvtár, mely a teszt esetekben előforduló közös, alapértelmezés szerinti viselkedést adja meg.

A 3.10.4. ábrán látható teszteset dinamikus viselkedését definiáló táblázatnak van egy fejléce, mint minden TTCN-2-ben használt táblázatnak. Ez tartalmazza a teszteset nevét, a teszt célt, melyet a teszteset megvalósít, valamint megadható egy kiválasztási feltétel is. Ez alapján történik a teszteset kiválasztása a teszt végrehajtás során. A táblázat törzs része tartalmazza a viselkedést leíró oszlopot ahol a protokoll és szolgálati üzenetek található (PDUs, Protocol data Units, and ASPs Abstract Service Primitives), amelyeket küldeni kell a teszteset végrehajtása során a teszternek, illetve amelyeket válaszként várunk. A sorok nem egyforma beugrással kezdődnek, ezt használják az események sorrendjének megadására. Minél beljebb kezdődő sorban helyezkedik el egy üzenet annál később kerül sorra. Az azonos beugrással kezdődő sorok alternatívákat tartalmaznak, így ezekben csak vett

üzenetek szerepelhetnek. A felkiáltó jel ("!") olyan üzenetek előtt áll, melyeket a teszter kiküld, a kérdő jel ("?") pedig olyanok előtt, melyet a teszter kap.

A korlátok referenciáit tartalmazó oszlopban olyan táblázatok nevei állnak, melyek az üzenetekben előforduló paraméter értékeket definiálják, azaz a korlátok részhez tartozó táblázatokra való hivatkozást. Az üzenetek adattípusai TTCN-ben megadhatók mind táblázatokban mind pedig ASN.1-ben. Az ASN.1 definíciókat általában olyan protokollok esetén használják, ahol a protokoll szabványban is szerepelnek az ASN.1 adatdefiníciók. Az ASN.1 használata egyre jobban terjed a protokoll tervezésben és a konformancia tesztelésben. A verdikt oszlop tartalmazza az alternatív bejövő üzenetekhez tartozó végső teszt eredményeket, azaz hogy egy adott üzenet vétele esetén helyes vagy hibás a teszteredmény. Egy tesztesethez csak egy helyes teszt eredmény tartozik.

Test Case Dynamic Behaviour					
Test Case Name:TC_2_Task_not_understood_with_first_trial					
Group :					
Purpose : Test that IUT does not accept rubbish as work instructions					
Configuration : MTC_2_PTCs					
Default :					
Comments : This case shows some basic principles of non concurrent TTCN test cases.					
Selection Ref :					
Description : ENV: SYS REQ: FS New order form hadler, Section A Formats and codes.					
Ind	Label	Behaviour Description	Constraints Ref	Verdict	Comments
0		+preamble			
1		MTC_PCO ! Task_envelope START T_course_30s	high_work (work_unclear_instruction!)		
2		MTC_PCO ? Task_envelope CANCEL T_course_30s	high_work (dont_understand)		
3		MTC_PCO ! Task_envelope START T_course_30s	high_work (work_harder)		
4		MTC_PCO ? Task_envelope CANCEL T_course_30s	high_work (ok)	(F)	
5		+postamble			
4		? TIMEOUT T_course_30s		(F)	
5		+postamble			
4		MTC_PCO ? OTHERWISE CANCEL T_course_30s		(F)	
5		+postamble			
2		? TIMEOUT T_course_30s		(F)	
3		+postamble			
2		MTC_PCO ? OTHERWISE CANCEL T_course_30s		(F)	
3		+postamble			
Detailed Comments: The sequence of events is:					
		MTC	IUT		
		I--TASK_DESCRIPTION----	>I	rubbish	
		I<--TASK_CONFIRMATION--I		don't understand	
		I--TASK_DESCRIPTION----	>I	reasonable	
		I<--TASK_CONFIRMATION--I		OK	

3.10.4 ábra. Példa tesztesetre

A teszt esetek végrehajtási sorrendjét nem határozza meg a tesztkészlet. A teszt eseteket logikai csoportokba rendezik a tesztkészleten belül, amely segíti a

tesztkészlet olvasását és írását. A tesztkészlet legfontosabb építőköve a teszt, eset, és ezek legkisebb építőkövei a teszt események. A teszt események legtöbbször üzenetek (Protocol Data Units) és ASP-k (Abstract Service Primitives) cseréjét írják le a tesztrendszer teszt megfigyelési pontjain (Point of Control and Observation, PCO). A tesztesetek közös részei teszt lépésként adhatók meg, melyek további teszt lépéseket foglalhatnak magukba.

TTCN-3

Egyre növekvő szükség volt egy olyan teszt nyelvre, melyen a konformancia tesztelés mellett további tesztelési típusok is leírhatók. A TTCN új verziója a TTCN-3 2000-ben jelent meg.

Szintaktikai szempontból a TTCN-3 eltér a nyelv régebbi verzióitól, egy modern programozási nyelvhez hasonlít. Számos új képessége mellett megőrizte a TTCN-2 hasznosnak bizonyult tulajdonságait is. A teszt írók és teszt végrehajtók számára egyszerűbb lesz megtanulni és használni ezt az általános célú, flexibilis és felhasználó-barát nyelvet. Az új szintaxis és az új tulajdonságok révén olyan területeken is használható lesz, ahol eddig nem alkalmazták a TTCN-t. Miután ez a nyelv közelebb áll a termékek implementációs nyelvéhez így várhatóan a gyártók szoftver fejlesztési folyamatába is könnyebb lesz beépíteni.

TTCN-3 használható nemcsak a mobil és Internet protokollok tesztelésére, hanem CORBA-alapú platformok és API-k (Application Programming Interfaces) tesztelésére is. A konformancia tesztelésen túl alkalmazható együttműködés, robusztusság, regressziós, rendszer és összekapcsolási tesztekhez is.

A TTCN-3 szintaxisa új, de megőrizte és továbbfejlesztette elődje hasznos tulajdonságait:

- Dinamikus és párhuzamosan futó teszt konfigurációk alakíthatók ki vele,
- Szinkron és aszinkron kommunikációt tud megvalósítani,
- Megadhatók kódolási szabályok az üzenetekhez, a felhasználók is definiálhatnak ilyeneket,
- Hatékony illesztési algoritmust használ adat formátumokhoz,
- Típusok és értékek paraméterezhetők,
- Teszt végeredmények kezelése verdikttel történik,
- Teszt készlet paraméterezés és teszt este kiválasztási feltételek adhatók meg,

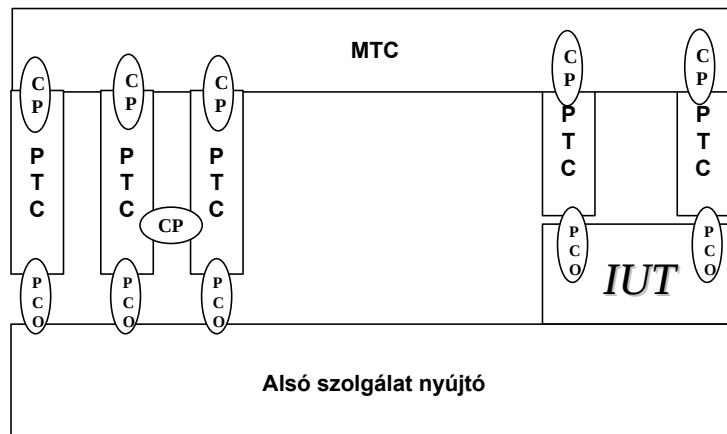
- Együtt használható az ASN.1-el, esetleg más nyelvvel mint pl. az IDL.

A TTCN-3-nak jól definiált szintakszisa és statikus szemantikája van. Opcionálisan különböző megjelenítési formátumok (táblázatos, grafikus stb.) használhatók vele hogy a felhasználók könnyebben olvashassák a TTCN-3 teszteket. Miután a TTCN-2-t már számos helyen használják, ezen tesztelőknek ismerős lesz majd a táblázatos megjelenítési formátum.

Akik a hagyományos programozási nyelvek használatához szoktak hozzá, azoknak a Core nyelv lesz a legegyszerűbben használható eszköz tesztek írásához. Akik MSC-ben szokták olvasni a teszteket, azok számára készült a grafikus megjelenítési formátum.

3.10.5. Teszt konfigurációk és teszt végrehajtás

A teszt konfiguráció vagy elrendezés a teszt rendszer felépítését adja meg és kapcsolatát a tesztelt protokoll megvalósítással (Implementation Under Test, IUT). A teszt konfiguráció nagyon függ attól, hogy milyen típusú tesztet hajtunk végre. Konformancia tesztelés esetén csak a teszter és az IUT része a teszt konfigurációnak. A teszt eszköz számos interfészen keresztül tud kapcsolódni az IUT-hez a párhuzamos teszt komponensein keresztül (3.10.5. ábra). Az egyik teszt komponens koordinálja a teszt végrehajtást (main test component) és további számos párhuzamosan futó teszt komponens tud küldeni és fogadni üzeneteket az IUT-től. Ezek az üzenetek a PCO-n (Point of Control and Observation) keresztül jutnak el az IUT-hez, mely egy absztrakt interfész a teszter és a tesztelt protokoll között. A teszter az IUT-t az alsó réteghatáron tudja elérni az alsó réteg szolgáltatását felhasználva. Előfordul, hogy egyes IUT-k felső réteghatárja is elérhető a tesztelés során. A teszt komponensek egymásnak is tudnak üzenetet küldeni, ezek a koordináló üzenetek és a koordinációs pontokon (Coordination Point, CPs) keresztül kerülnek átadásra. Ez a teszt konfiguráció mind TTCN-2-ben mind TTCN-3-ban alkalmazható. Mivel a teszt esetek egy adott teszt elrendezésre vonatkoznak, így ezt is meghatározzák a teszt esetben.



3.10.5. ábra. Konformancia teszt elrendezés

A konformancia tesztelés szigorúan szabványosított és a legjobban kidolgozott tesztelési eljárás. Az egyéb teszt típusok bár fontos részét képezik a gyártók fejlesztési tevékenységének, mégsem szabványosították őket, hiszen nem jelentősek a különböző implementációk együttműködése szempontjából. A konformancia tesztelést elsősorban a távközlési szoftverek és protokollok fejlesztésénél alkalmazzák. Az Internet protokoll megvalósításokat együttműködés teszteknek szokták alávetni. A konformancia tesztelés az Internet protokoll megvalósításokra is hasznos teszt eredményeket ad, így ezen a területen is egyre jobban terjed, bár nem kapcsolódik hozzá bizonyítvány kiadási eljárás.