



Budapesti Műszaki és Gazdaságtudományi Egyetem
Távközlési és Médiainformatikai Tanszék

Forgalommenedzsment algoritmusok MPLS hálózatokban

Ph.D. téziszfüzet

Józsa Balázs Gábor

*Nagysebességű Hálózatok Laboratóriuma
Távközlési és Médiainformatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem*

Tudományos vezetők:

Magyar Gábor, Ph.D.

*Forgalom Analízis és Hálózati Teljesítmény Laboratórium
Ericsson Magyarország Kft.*

Cinkler Tibor, Ph.D.

*Nagysebességű Hálózatok Laboratóriuma
Távközlési és Médiainformatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem*

Budapest, 2004. szeptember

1. Bevezetés

Disszertációmban bemutatom az eredményeimet számos *forgalommenedzsment* (TE, traffic engineering) problémára, amelyekkel a *többprotokollos címkekapcsolt* (MPLS, multiprotocol label switching) hálózatok területén találkoztam. Az elért eredmények nagy része azonban nem korlátozódik az MPLS környezetre, hanem más olyan hálózati technológiákon (pl. aszinkron átviteli mód) is alkalmazható, ahol lehetőség van sávszélességgel rendelkező útvonalakat explicit módon megadni.

Forgalommenedzsment

A forgalommenedzsment, amelyet az IETF (Internet Engineering Task Force) szervezet definiált [1]-ben, a működő IP (Internet Protocol) hálózatok teljesítményének kiértékelésével és optimalizálásával foglalkozik. A TE fő célja a hálózatok teljesítményének növelése, aminek két lényegesen különböző megközelítése létezik. A *forgalom-orientált* TE a minőség és rendelkezésre állás biztosítását célozza meg a hálózati felhasználók számára. A szolgáltatók részéről pedig a hálózati erőforrások gazdaságos és megbízható kihasználása a fő szempont, ami az *erőforrás-orientált* megközelítése a TE-nek.

A hálózatok fő célja az információ továbbítása a forrástól a nyelő felé, ezért a legfontosabb feladatuk a forgalom irányítása. Így a legjelentősebb TE funkciók az útvonalválasztás felügyelete és optimalizálása a lehető leghatékonyabb forgalom továbbítás érdekében. Ezen funkciók egy kifinomult, központosított útvonalválasztást tesznek lehetővé a hálózati állapot, politika, valamint különböző döntési változók alapján.

A TE nem egyszeri cél, hanem a hálózati teljesítmény javításának iteratív folyamata, ezáltal a TE funkciókat periodikusan kell használni a hosszú távon jó hálózati teljesítmény elérése érdekében. Mindazonáltal a TE optimalizálási céljai változhatnak az idő során, ahogy új követelmények, új technológiák és új problémák merülnek fel a hálózatban, amelyeket a TE funkcióknak figyelembe kell venniük.

MPLS

A többprotokollos címkekapcsolás (MPLS) [2] egy keretrendszer, amelyet IP csomagok gerinc-hálózatokban történő hatékony továbbítására dolgoztak ki. Az MPLS-ben két útvonalválasztó típust különböztetünk meg: (1) címke határ útvonalválasztó (LER, label edge router), amely az MPLS hálózatot a klasszikus IP hálózattal köti össze, és (2) címkekapcsoló útvonalválasztó (LSR, label switching router), amely csak MPLS képességű útvonalválasztókkal van összekötésben. A forgalom a LER-ek között előre kiépített címkekapcsolt útvonalakat (LSP-eket, label switched path-okat) használ, amelyek explicit módon is kijelölhetők az erőforrás-foglalási protokoll forgalommenedzsment kiterjesztésével (RSVP-TE, resource reservation protocol for traffic engineering) [3] vagy a kényszereket figyelembe vevő címke elosztási protokoll (CR-LDP, constraint based label distribution protocol) [4] segítségével. Továbbá sávszélesség foglalható az egyes LSP-khez a TE támogatása érdekében, ami a disszertáció egyik alapfeltevése.

A forgalommenedzsment MPLS-sel kombinálva egy megfelelő környezetet szolgáltat a megkülönböztetett szolgáltatások követelményeinek teljesítéséhez. Egyik nagy előnye ennek a koncepciónak, hogy nem igényel jelentős változtatásokat a meglévő IP hálózatok alapjaiban. To-

vábbá e koncepció összeillik az aszinkron átviteli módú (ATM, asynchronous transfer mode) kapcsolókkal, amelyek széles körben elterjedtek a szolgáltatók gerinchálózataiban. Összefoglalva, nagy esély van arra, hogy a következő generációs útvonalválasztási platform ötvözni fogja az MPLS és a forgalommenedzsment előnyeit.

2. Kutatási célkitűzések

A disszertációm motivációját az MPLS hálózatokban az operátor szempontjából felmerülő teljesítmény-optimalizálási problémák nyújtották, ezekre új forgalommenedzsment algoritmusok kidolgozását céloztam meg.

Az optimális erőforrás-kihasználás szükségessége az új hálózati alkalmazások rohamosan növekvő sávszélesség-igényére, valamint az egyre nagyobb számú felhasználóra vezethető vissza. A hálózati kapacitás igénye mellett a minőség és a rendelkezésre állás is egyre fontosabb a hálózat-üzemeltetők szempontjából annak érdekében, hogy nyereségesek maradjanak a szolgáltatók versenyében.

A forgalommenedzsment MPLS-sel lehetőséget nyújt a központosított útvonalválasztási döntések meghozatalában, ezzel lehetővé téve kifinomult útvonalválasztási algoritmusok alkalmazását, amelyek a működő hálózatok teljesítményének növelését célozzák meg. Jövőbeni MPLS hálózatok tervezése esetén pedig érdemes olyan tervezőalgoritmust használni, amely figyelembe veszi az útvonalak átkonfigurálhatóságának lehetőségét, ezáltal csökkentve a kiépítési költséget.

Fő célok

Disszertációm fő céljai a következők:

- egy robusztus globális útvonal-optimalizáló algoritmus kidolgozása, amely képes az MPLS sajátosságait figyelembe venni,
- a tartalék útvonalak számításának és a hibatűrésnek a tökéletesítése megosztott védelmi környezetben,
- megoldási javaslatok tétele az útvonal-optimalizálás után felmerülő útvonal átkonfigurálási problémára,
- egy a forgalomeloszlás változásait figyelembe vevő hálózattervező algoritmus megalkotása, amely képes a lépcsős költségfüggvények sajátosságainak kihasználására.

Mіндеzek mellett az új algoritmusok teljesítményvizsgálatát is megcéloztam annak érdekében, hogy bizonyítsam hatékonyságukat.

3. Módszertan

Kutatási munkám során *NP-nehéz* [5] problémákkal szembesültem, amelyekre a tudomány mai állása szerint optimális megoldás nem garantálható elfogadható időn belül a gyakorlatban előforduló méretű problémákra. Mivel a gyakorlati alkalmazhatóság elsődleges szempontom volt,

ezért a disszertációmban nem foglalkozom egzakt algoritmusokkal a futási idejük nagysága miatt. Néhány tíz csomópontból álló problémapéldányok egzakt megoldása több óráig is eltarthat [6], továbbá léteznek olyan gyakorlatban előforduló problémapéldányok, amelyek nem megoldhatók a hatalmas memóriaszükséglet miatt. Minden ismertetendő algoritmus *heurisztikus optimalizálás*on alapul, amely a szükséges hatékonyságot képes nyújtani és a futási ideje elfogadható.

A hálózati architektúrákra tervezett algoritmusok teljesítőképességének kiértékelésére három megközelítés létezik: analízis, szimuláció, valamint mérés mintahálózaton. Az analitikus módszereket széles körben alkalmazták évekkel ezelőtt a nyilvános kapcsolt telefonhálózatok esetén. Az adathálózatoknak azonban sok tulajdonsága eltér a telefonhálózatokétól, ami az analitikus eljárások alkalmatlanságát eredményezi a legtöbb esetben. Annak ellenére, hogy a mintahálózatokon történő mérések szolgáltatják a legpontosabb eredményeket, a kutatásaim számára szükségtelenek voltak, mert nem valósidejű, globális optimalizálási problémákra koncentráltam, amelyek központosított adatokon végzett számításokon alapulnak. A fentiek miatt a teljesítmény vizsgálatára *szimulációt* alkalmaztam.

A tézisekben a sok különböző hálózati topológia előállításának problémája hasonló formában merült fel, ezért az 1. tézisben javasoltam egy véletlenszerű topológia-generálási módszert. E generátor alapján a problémapéldányok előállítása automatizálható, amelynek segítségével egyszerűen hozható létre nagyszámú példány, ezzel növelve az eredmények megbízhatóságát. Másrészt pedig minden tézisben valós hálózati topológiákat is vizsgáltam és az eredményeiket összevetettem a véletlenszerű topológiáéval.

A hálózati topológiát irányított gráffal modelleztem, amelyek élei kapacitással rendelkeznek. Egy forgalmi igényt általában a forrás és nyelő csomópontja, valamint a szükséges sávszélessége határozott meg. Mindemellett néhány esetben ez a reprezentáció kiegészült kötelezően érintendő csomópontokkal/szakaszokkal, hosszkorláttal és tartalék útvonalakkal, amelyek az MPLS hálózatok forgalommenedzsment képességeiből adódtak. Az 1–3. tézisben az új algoritmusok kiértékelése elsősorban a *sikerességi valószínűség* mértékén alapult, amely a sikeresen megoldott problémapéldányok és az összes problémapéldány számának aránya. A 4. tézis egy költséghatékony hálózattervezéssel foglalkozik, ahol a *hálózat összköltsége* volt az elsődleges teljesítménymutató.

4. Új eredmények

Új tudományos eredményeimet négy fő tézisbe, ezeken belül altézisekbe rendeztem. A tézisek címénél megjelölöm a kapcsolódó publikációkat, az altéziseknél pedig hivatkozom a disszertáció megfelelő részeire, amely azokat részletesen kifejti (pl. [D3.3] a 3.3. fejezetre utal).

1. tézis. Globális útvonal-optimalizálás [J1]

A globális útvonal-optimalizálás a forgalommenedzsment egyik alapfeladata, és nagy a gyakorlati jelentősége, hiszen a változó útvonalak hosszútávon csökkent erőforrás-kihasználást eredményezhetnek a hálózatban. Ez az eset áll fenn akkor is, ha az RSVP-TE dinamikus forgalommenedzsmentjét használjuk, hiszen az RSVP-TE csak lokális optimalizálást végez. Ennek elkerülésére periodikus (nem túl gyakori, pl. napi vagy heti) globális útvonal-optimalizálást hajthat végre az operátor (ami akár automatizálható is) a dinamikus forgalommenedzsment

kikapcsolása mellett. E feladatban megbúvó optimalizálási probléma a következőképpen fordítható le a hálózati folyamatok elméletére: rendeljük *útvonalakat* a kapacitáskövetelményekkel rendelkező *forgalmi igények* halmazához egy kapacitáskorlátos gráfban. Ez egy többforrású, többtermékes osztatlan folyamprobléma [7, 8], NP-nehéz [5], és számos megoldása létezik az irodalomban. Egyik lehetséges megközelítési mód egészértékű lineáris programozási szoftvercsomagok használata, amelyekkel egzakt megoldás adható a feladatra, azonban csak nagyon kis problémaméretek esetén. Ezért az irodalomban csak közelítő megoldások [10, 11, 12, 13] és heurisztikák [14, 15, 16, 17, 18, 19, 20] találhatók. A gyakorlatban megjelenő problémaméretek olyan nagyok (ld. egy következő generációs mobil távközlési hálózat gerinchálózata), hogy a létező módszerek futási ideje az ilyen problémapéldányokon elfogadhatatlan a gyakorlati alkalmazásra. Ráadásul az új technológiák (pl. MPLS) új kényszerekkel egészítik ki a problémát (pl. kötelezően érintendő csomópontok/szakaszok, hosszkorlát és tartalék útvonalak), amelyeket az ismert módszerek nem kezelnek. Ezért alapvető fontosságú egy olyan algoritmus kidolgozása, amely elfogadható futási időn belül képes megoldani új kényszereket támaztó nagy problémapéldányokat.

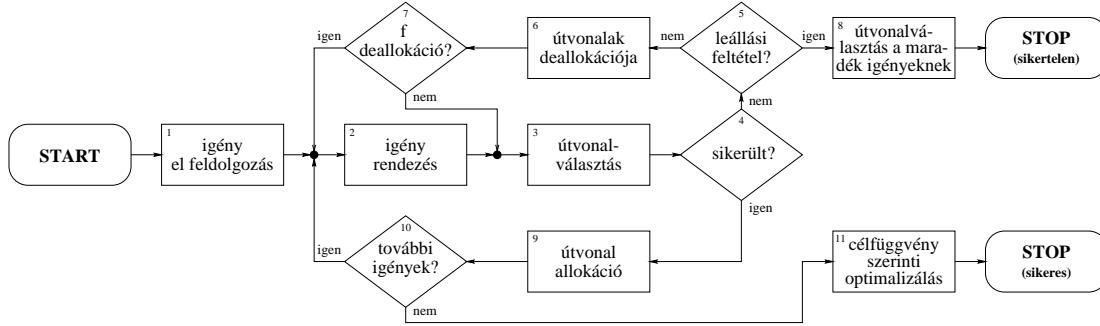
1.1. altézis. Egy hatékony algoritmus a globális útvonal-optimalizálásra [D3.3]

A globális útvonal-optimalizálásra javasolt algoritmus (lásd 1. ábra) egy iteráción alapul, amely emlékeztethet a *szimulált helyfoglalás* módszerére. Két halmazt tart fenn: (1) az aktuálisan kielégítetlen forgalmi igények halmazát és (2) a kiszámolt útvonalak halmazát. Minden iterációs lépésben egy igényből útvonal lesz (*allokáció*), vagy útvonalak bontódnak le és alakulnak vissza igényekké (*deallokáció*), hogy később új útvonalakká válhassanak. Az igényekhez tartozó útvonalak számolása szisztematikus módon, egyesével történik Dijkstra legrövidebb útvonalakat kereső algoritmusával [21]. Az iteratív útvonalkeresés során a választási lehetőségek idővel változnak. Egy igen fontos jellemzője az algoritmusnak az, hogy alkalmazkodik az aktuális problémapéldány tulajdonságaihoz és nehézségéhez azon élek felesleges terhelését elkerülve, amelyek szűk keresztmetszetet nyújtanak.

Az algoritmus magja a következő heurisztikákat tartalmazza (zárójelben jelölve a hozzájuk tartozó funkcionális blokkot) az aktuális problémapéldány nehézségeinek feltérképezésére és a problémapéldányok széles skálájának sikeres megoldása érdekében: igény előfeldolgozás (1), igény nehézségi mérték (2), kritikus igény beillesztése (2), adaptív élsúlyfüggvény (3), adaptív útvonalhossz korlát (4), leállási feltétel (5), deallokációra jelölt útvonalak (6), élsúly szorzó beállítás (6), a deallokált útvonalak számának változtatása (6), deallokációs verseny (6) és fő deallokáció (6). Az algoritmus lényege az előbbi heurisztikák kombinációja, amely hatékonyabbá teszi a többi módszerhez (többek között a szimulált helyfoglalás módszeréhez) képest. Ez a megközelítés az optimalizálási célok rugalmas megválaszthatóságát nyújtja (pl. az összes szabad kapacitás maximalizálása, a szűk keresztmetszet minimalizálása). Továbbá támogatja az MPLS hálózatokban megjelenő új technikai követelmények kezelését: kötelezően érintendő csomópontok/szakaszok, hosszkorlát és tartalék útvonalak.

1.2. altézis. A javasolt globális útvonal-optimalizáló algoritmus teljesítményvizsgálata [D3.4]

A javasolt globális útvonal-optimalizáló algoritmus teljesítmény-vizsgálatát szimuláció segítségével végeztem véletlenszerűen generált, valamint valós hálózati topológiákon. A véletlen-



1. ábra. A javasolt globális útvonal-optimalizáló algoritmus fő funkcionális blokkjai.

szerű topológiák előállítására kifejlesztettem egy módszert, amely a valós hálózati szituációkat utánozza és a topológia-generálás tudományos munkáin alapszik [22, 23, 24, 25]. A topológia generátornak két fő bemenő paramétere van: a csomópontok száma (n) és az átlagos csomóponti fokszám (g). A két paraméterből egyből adódik, hogy a szükséges élszám $\frac{n \cdot g}{2}$. A topológia generátor négy fő lépésből álló műveletsort végez. Először a csomópontokat egyenletes eloszlással elhelyezi egy A oldalhosszúságú négyzet alapú területre, ügyelve arra, hogy a csomópontok közötti távolság ne legyen az $\frac{A}{2 \cdot \sqrt{n}}$ kifejezésnél kisebb. A csomópontok helyének meghatározása után a gráfot kétszeresen összefüggővé teszi egy Hamilton kör éleinek hozzáadásával. A Hamilton kör a csomópontok konvex burkából indul ki, majd a maradék csomópontokat a kapcsolódási szög mohó maximalizálásával kapcsolja a körbe. Ez a kör két diszjunkt útvonalat eredményez minden csomópontpár között, ami tipikus a valós gerinchálózatokban. Ezután az átlagos csomóponti fokszámot növeli új él hozzáadásával. Minden lépésben az aktuálisan leghosszabb kört osztja két kisebb körre egy új él hozzáadásával. Végül a maradék éleket iteratívan adja a gráfhoz egy valószínűségi séma alapján (lásd [25]), ahol egy (u, v) él hozzáadásának valószínűsége $p(u, v) = \frac{1}{\mu} \cdot \exp(-\frac{D_{u,v}}{\mu})$, ahol $D_{u,v}$ a két csomópont távolsága és a $\mu = \frac{A}{\sqrt{n}}$ paraméter pedig az exponenciális eloszlásból következő várható élhosszúság. Az él hozzáadása akkor ér véget, amikor a szükséges átlagos csomóponti fokszámot elérte.

A szimulációk célja a következőkben felsoroltak vizsgálata volt: sikerességi valószínűség, futási idő, optimalizálási célfüggvény, nagy problémaméret, technikai követelmények kezelése, a funkcionális egységek szükségessége, stratégiai változók, valamint más módszerekkel történő összehasonlítás. Mivel más módszerek implementációja nem állt rendelkezésre, csak korlátozott számú összehasonlítást végeztem (a módszerek kifejlesztőinek segítségével). Ezen összehasonlítások alapján azt mondhatom, hogy a javasolt algoritmus jobb eredményeket ad a többi algoritmusnál, illetve ugyanolyan minőségű eredményt tud adni jelentősen kisebb futási idő alatt.

A legfontosabb szempont a sikeresen megoldott problémapéldányok és az összes problémapéldány aránya (azaz a sikerességi valószínűség), amelyet az 1. táblázat tartalmaz (minden érték 100 mérés átlagát reprezentálja). Azokban az esetekben, amikor a sikerességi ráta nem volt 100%, az elvezetett forgalom nagysága is fel van tüntetve (átlagolva), annak érdekében, hogy pontosabb képet kapjunk az algoritmus teljesítményéről. A táblázatban n a hálózat csomópontjainak száma és α jelképezi a többlet szabad hálózati erőforrások arányát az igények legkevesebb fizikai szakaszon történő elvezetése esetén. $\alpha = 0\%$ esetén, amikor nincs többlet

szabad erőforrás a hálózatban, az algoritmus n értékétől függetlenül nem tudott megoldani egyetlen problémapéldányt sem. Ez nem meglepő, hiszen a probléma NP-nehéz, így egy megfelelő megoldás megtalálása különösen nehéz feladat ilyen „szoros” esetekben. Másrésztől $\alpha > 3\%$ teljesülése esetén a sikerességi ráta minden esetben 100%. Továbbá megfigyelhető, hogy a relatíve kisebb problémapéldányok nehezen megoldhatók, $n \geq 100$ esetén viszont még 1%-os α értékre is kiváló eredményeket produkál az algoritmus. Ennek oka az lehet, hogy kisebb problémaméreteknél mind az alternatív útvonalak száma, mind az igények száma sokkal kisebb, következésképp a megfelelő megoldások száma is kisebb.

n / α	0%	1%	1,5%	2%	3%	5%	10%
25	0/98,21	0/98,91	1/99,16	6/99,53	73/99,94	100	100
50	0/98,79	0/99,60	46/99,89	99/100	100	100	100
75	0/99,27	25/99,85	99/100	100	100	100	100
100	0/99,38	92/100	100	100	100	100	100
200	0/99,50	100	100	100	100	100	100
400	0/99,78	100	100	100	100	100	100

1. táblázat.

A javasolt globális útvonal-optimalizáló algoritmus sikerességi valószínűség [%] és elvezetett forgalom [%] értékei második tizedesre kerekítve különböző problémaosztályokra.

A teljesítmény-vizsgálat fő eredményei a következőképpen foglalhatók össze. A javasolt algoritmus számos előnnyel rendelkezik a hasonló módszerekhez képest futási idő, flexibilitás és megoldási minőség terén is. Gyakorlati alkalmazása javasolt, mert nagyméretű hálózatokon is működik, kezelni tudja a valós hálózatok sajátosságait, futási ideje elfogadható, és eredményei kiválóak megfelelő megoldás keresésére és jók a különböző optimalizálási célfüggvények esetén is.

2. tézis. Védelmi útvonalak számítása megosztott védelmi környezetben [C3, C7]

A globális útvonal-optimalizálásnak ebben a változatában a cél egy *aktív* és egy *védelmi* útvonalhalmaz meghatározása a forgalmi igények alapján úgy, hogy a sávszélesség-foglalások összege megosztott védelem alkalmazása esetén a lehető legkisebb legyen. Az MPLS hálózatot egy irányított gráffal modellezem, ahol az élhez kapacitásértékek tartoznak (az e élhez $c(e)$), amelyek a hozzájuk tartozó fizikai szakaszok foglalható sávszélességét reprezentálják. Egy d_i forgalmi igényt az s_i forrás csomópontjával, a t_i nyelő csomópontjával, valamint a b_i igényelt sávszélességével adom meg. A végeredmény pedig a P_i^1 és P_i^2 diszjunkt útvonal pár d_i igényenként, azaz két rendezett élhalmaz közös elem nélkül. Mivel az erőforrás kihasználás eléggé pazarló abban az esetben, amikor a tartalék foglalások dedikáltak az egyes tartalék útvonalakhoz (*dedikált védelem*), ezért más védelmi stratégiák részesülnek előnyben.

A *megosztott védelem* az IETF által támogatott erőforrás-hatékony MPLS szintű védelmi módszer [26], amely a következő ötleten alapul. Feltételezve, hogy egyszerre csak egy meghibásodás történhet, két diszjunkt aktív útvonallal rendelkező LSP megoszthatja a tartalék útvonalaik közös szakaszára foglalt kapacitást, hiszen egyszerre csak az egyikük hibásodhat meg. A védelmi foglalások megosztásával a kapcsolódó irodalomban sokat foglalkoztak

[6, 20, 27, 28, 29, 30, 31], és a védelem és a sáv szélesség-foglalás jó kompromisszumának tűnik. Azonban a létező megoldások nem optimalizálják a sáv szélesség megosztásának mértékét, ezért szükséges új megközelítések után kutatni a sáv szélesség megosztás tökéletesítésére.

Az e_j élen foglalt $r(e_j)$ kapacitás két részből áll. Az egyik rész az aktív utak számára foglalt:

$$\sum_{x:e_j \in P_x^1} b_x. \quad (1)$$

A másik rész pedig a tartalék útvonalak számára van fenntartva, ami nem más, mint a lehetséges legnagyobb tartalék útvonalakon zajló forgalom az e_j élen bármely másik él meghibásodása esetén. Ez a foglalás-rész az úgynevezett *tartalék foglalási mátrix* (BRM, backup reservation matrix) segítségével számítható ki, ahol a sorok és az oszlopok a gráf éleinek felelnek meg. Feltételezve, hogy egy hiba egy fizikai szakasz mindkét irányára kihat, akkor a gyakorlatban elegendő $\frac{m}{2}$ sor a BRM-nek, ahol egy sor egy meghibásodott fizikai szakasz mindkét irányát reprezentáló két gráfélt jelöli. Tehát egy él helyett élpárral kellene számolni, de ettől a bonyolultabb formalizálástól most eltekintek, mert az eredmények az általánosság csorbítása nélkül átalakíthatók a kétirányú esetre. A BRM $B_{i,j}$ eleme az e_i él meghibásodása esetén az e_j élen szükséges sáv szélesség értékét jelöli a tartalék útvonalak számára:

$$B_{i,j} := \sum_{x:e_i \in P_x^1 \wedge e_j \in P_x^2} b_x. \quad (2)$$

Összegezve, a tartalék útvonalak számára e_j élen szükséges foglalás (megosztott védelem esetén) $\max_i B_{i,j}$ kifejezéssel egyezik meg, amely a lehetséges legnagyobb tartalék útvonalakon zajló forgalom az e_j élen egy szakasz meghibásodása esetén.

A feladat egy aktív és egy védelmi útvonalhalmaz meghatározása, amelyekre a foglalások összege a lehető legkisebb. Feltételezve, hogy az aktív útvonalak egyben legrövidebb utak, a probléma formálisan a következő:

$$\min_j \sum_i \max_i B_{i,j}. \quad (3)$$

2.1. altézis. Új módszerek a megosztott védelem hatékony kihasználására [D4.3]

Három különböző módszert mutatok be, amelyek célja a tartalék útvonalak sáv szélesség foglalásának csökkentése, vagyis a hálózat áteresztőképességének növelése. A módszereknek egy közös lépése egy aktív és egy tartalék útvonalat igénylő forgalmi igény kielégítése. Ezt a problémát az 1. tézis globális útvonal-optimalizáló algoritmusának olyan kiegészítésével oldom meg, amely képessé teszi a tartalék útvonalak foglalásainak megosztására. A három különböző megközelítés különböző kiterjesztéseket igényel az eredeti optimalizáló algoritmusban, amelyeket a következőkben részletezek.

Maximum csökkentés (CDM, Cut Down Maximums). Ez a megközelítés egy utólagos feldolgozás, mert a kiinduló állapot egy kész hálózati konfiguráció (pl. [J1] által meghatározott), ahol mind az aktív, mind a tartalék útvonalak meghatározottak, továbbá a tartalék

útvonalak foglalásai megosztottak. Lényeges különbség az új heurisztika és a hasonló publikált algoritmusok [27, 28] között, hogy az aktív útvonalak újraszámolása is engedélyezett.

Amint arra már kitértem, a konkrét kapacitásfoglalás a tartalék útvonalak számára egy adott e_j élen a tartalék foglalási mátrix (BRM) j -edik oszlopának maximumaként határozható meg. A CDM módszer ötlete a következő: próbáljuk meg csökkenteni egy adott oszlop maximumát úgy, hogy a többi maximum érték ne növekedjen jelentősen; azaz az oszlop-maximumok összege csökkenjen. Ezáltal minden sikeres csökkentés esetén a hálózat összterheltsége szintén csökken, ami az elsődleges célunk. Tehát a CDM alkalmazásával az útvonalrendszerek – amelyek eredetileg nincsenek sáv szélesség megosztásra optimalizálva – módosíthatók úgy, hogy az eredményként kapott konfigurációk kisebb terheltséggel bírjanak.

Adaptív módszer (AM, Adaptive Method). A második megközelítés nem igényel egy kész útvonalrendszert, hanem egy „üres” hálózatból indul ki kielégítetlen forgalmi igényekkel. Az AM módszer kulcsa egy speciális, adaptív súlyfüggvény. Az igények egymás utáni kielégítése közben a BRM folyamatosan változik, és az alkalmazott súlyfüggvény a BRM aktuális értékeit veszi alapul. Így minden egyes útvonalszámítás alkalmazkodik az adott hálózati szituációhoz, vagyis mindegyik forgalmi igény *mohó módon* kerül kielégítésre, ami az aktuális hálózati konfiguráció szerinti optimális elvezetést eredményezi.

Az adaptív költségfüggvény fő célja a BRM oszlop-maximumainak alacsony szinten tartása. Ez a költségfüggvény alapvetően különböző az adott igény aktív és tartalék útvonalára. A két ötlet a következő:

- aktív útvonal esetén az e_i él $w(e_i)$ súlya arányos az i -edik sorhoz tartozó átlagos relatív tartalék-foglalással: $w(e_i) \sim 1 + \frac{\sum_j \frac{B_{i,j}}{\max_x B_{x,j}} \cdot c(e_j)}{\sum_j c(e_j)}$,
- tartalék útvonal esetén az e_j él $w(e_j)$ súlya arányos a megfelelő aktív útvonalhoz tartozó átlagos relatív tartalék-foglalással: $w(e_j) \sim \frac{\sum_{i:e_i \in P_x^1} B_{i,j}}{|P_x^1| \cdot \max_i B_{i,j}}$, ahol d_x az aktuális igény.

Az arányosság az előbbi súlyfüggvényekben azt jelenti, hogy a globális útvonal-optimalizáló algoritmus eredeti súlyfüggvénye meg van szorozva a fenti kifejezések egyikével, attól függően, hogy a számolt útvonal aktív vagy tartalék. Az előbbi súlyfüggvények célja az útvonalak olyan élekre történő irányítása, amelyekhez tartozó BRM elemek értékei kicsik, így egy új, nagyobb maximum érték előállításának valószínűsége jelentősen csökken.

Iteratív módszer (IM, Iterative Method). Az IM alapötlete a hálózatban lévő tartalék útvonalak iteratív újraszámolása. A módszer két tartalék foglalási mátrixot (BRM-et) különböztet meg. A B^{actual} mindig az aktuális tartalék foglaláshoz tartozó $B_{i,j}$ értékeket tartalmazza, a B^{stored} pedig egy tárolt állapotot tükröz. A CDM-hez hasonlóan ez a módszer is egy kiindulási hálózati konfigurációt igényel, amelyhez első lépésként eltárolja az aktuális BRM-et ($B^{stored} := B^{actual}$). Ezután minden iterációs lépésben két műveletet hajt végre: (1) újraszámítja a tartalék útvonalakat az AM módszernél ismertett súlyfüggvény segítségével (amely a nagy foglalási értékű éleket próbálja elkerülni) B^{stored} alapján, és (2) az új BRM és tárolt BRM értékeit összekombinálja $B^{stored} := W \cdot B^{actual} + (1 - W) \cdot B^{stored}$ kifejezés szerint.

A két mátrix összekombinálása során az új BRM W relatív súlya minden iterációs lépésben csökken. Ez a módszer dinamikus indulását eredményezi, azaz gyorsan közelít az optimumhoz. Másrészt pedig a tárolt BRM növekvő aránya miatt a foglalási értékek oszcillációja elkerülhető.

A módszerek kombinációja. Mivel a három bemutatott módszer különböző megközelítésekben alapul, természetes gondolat kombinálni azokat egy hatékonyabb módszer kifejlesztése érdekében. Az AM egy üres hálózattól indul ki, ezért praktikus ezzel a módszerrel kezdeni. Utána az IM vagy a CDM alkalmazható, esetleg mindhárom módszer kombinációja is elképzelhető.

2.2. altézis. Többszörös hálózati hibák elleni megosztott védelem [D5.3]

Az egy tartalék útvonal alapú védelmi stratégiát – amellyel széles körben foglalkozik a szakirodalom – egyszerű hibákra tervezték, habár sok esetben többszörös hibák esetén is védelmet nyújt. Bár a tartalék foglalások megosztása jelentősen növeli a hálózati kapacitás gazdaságos kihasználását, hátránya, hogy a többszörös hibákat toleráló képességét csökkenti. A hálózatok méretének, integrációjának és bonyolultságának növekedésével a rendszer működése során bekövetkező többszörös hálózati hibák [32] egyre gyakoribbak lesznek. Emiatt a nagy fontosságú alkalmazások kiszolgálása során – főleg olyan hálózatokban, ahol nagy valószínűséggel következik be szakaszhiba – ezt az eseményt is figyelembe kell venni. A hálózatok kétszeres hibák elleni védelme nem ismeretlen a hálózatok világában (lásd pl. [33]), de gyakorlati megoldás nem található rá az irodalomban. Ezért bevezetek egy új megközelítést a hálózatok megóvására két egyszerre bekövetkező hibától, amelyben három teljesen független útvonalat határozunk meg minden egyes forgalmi igényhez. Továbbá egy új megosztott védelmi stratégiát javaslok, amely növeli a hibatoleráló képességet már abban az esetben is, amikor a forgalmi igények mindegyikéhez csak egy tartalék útvonalat rendelünk.

Amint azt már említettem, a tartalék foglalási mátrix (BRM) feladata a szükséges tartalék kapacitásértékek élenkénti tárolása. Egyszeres hibák elleni védelem esetén (amikor a forgalmi igényekhez egy tartalék útvonalat rendelünk) a BRM $\frac{m}{2}$ sort tartalmaz a lehetséges szakasz meghibásodásoknak megfelelően (feltételezve, hogy egy hiba egy fizikai szakasz mindkét irányára kihat). Erre az esetre SP1 jelöléssel utalok a következőkben.

A dupla hibákra való felkészülés érdekében a BRM-et úgy módosítottam, hogy annak egy sora két független szakasz egyidejű hibáját jelentse (SP2), ne csupán egy szakaszét. Ezért a módosított BRM-nek $\frac{(m/2) \cdot (m/2 - 1)}{2}$ sora lett, amely nem más, mint a két hibás esetek kombinációinak pontos száma. A különböző védelmi stratégiák összehasonlítása során referenciaként a védelem megosztása nélküli megközelítéseket (DP) is megvizsgáltam.

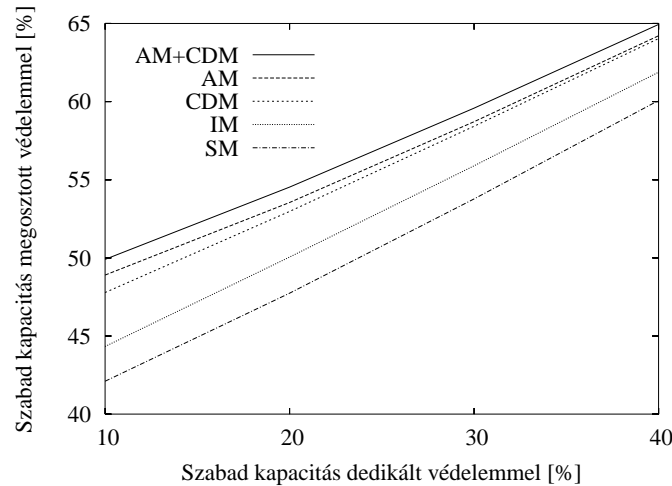
Ígéretes módszernek tűnik az SP2 foglalási stratégiát alkalmazni már az egy tartalék útvonalas esetben is, azaz a hálózatot megpróbáljuk dupla hibákra felkészíteni. Ez azt jelenti, hogy amennyiben két diszjunkt aktív útvonallal rendelkező LSP-nek van közös éle a tartalék útvonalukban, akkor az ezen az élen foglalt tartalék kapacitásnak elegendőnek kell lennie mindkét LSP számára. Mindazonáltal ha több, mint két diszjunkt aktív útnak van közös éle a hozzájuk tartozó tartalék útvonalakban, akkor a tartalék kapacitás csökkenthető (a nem megosztott védelmi esethez képest). Tehát csak azok az LSP-k szakadhatnak meg dupla hiba esetén, amelyeknél mindkét útvonal érintett a szakaszok hibájában.

Általánosítás. Általános esetben f számú ($f > 2$) egyszerre felmerülő hibára való felkészülésnél $f + 1$ diszjunkt útvonalat kell meghatározni az egyes LSP igények számára. Az előbbieken ismertetett megosztott védelmi módszer előnye, hogy kiterjeszhető tetszőleges számú tartalék útvonal esetére. Azonban a tartalék útvonalak számának növelése az útvonalszámítást és a védelem megosztását is sokkal nehezebbé teszi. Továbbá a valós hálózati topológiák vizsgálatánál kiderült, hogy négy (vagy több) diszjunkt útvonal meghatározása a csomópontok között az esetek többségében lehetetlen.

2.3. altézis. A javasolt megosztott védelmi módszerek teljesítményvizsgálata [D4.4, D5.4]

Új módszerek a megosztott védelem hatékony kihasználására. Egy hibára történő felkészülés esetén a tartalék útvonalak hatékony kiszámítására javasolt módszereimmel elérhető teljesítményjavulás vizsgálata érdekében számos szimulációt végeztem különböző példahálózatokon. A vizsgált hálózatokat az 1.2. altézisben ismertetett módszerrel generáltam.

A 2. ábra mutatja a különböző módszerek eredményeit valamint az eredeti globális útvonal-optimalizáló algoritmusét (SM) referenciaként. Az IM kombinációi azért nem szerepelnek az ábrán, mert azok nem tudták javítani a kapacitás kihasználását a kombinációban szereplő legjobb módszerhez képest. Az összes többi módszer jelentősen növelte a hálózati szabad kapacitást, továbbá azt tapasztaltam, hogy a módszerek kombinációjával a védelmi foglaltságok jobb megosztása érhető el. Az AM alkalmazását javaslom, amely majdnem a legnagyobb mértékben növelte a hálózat kihasználtságát, alacsony futási idő mellett. Amennyiben az idő nem jelentős tényező, akkor az AM és CDM kombinációjának (AM+CDM) alkalmazásával még jobb kihasználtság érhető el. Az AM és AM+CDM javasolt módszereket összevetettem más publikált algoritmusokkal és az SM referenciával is. Az elvégzett összehasonlítások alapján kiderült, hogy a javasolt algoritmusok jelentősen jobb megoldást tudnak biztosítani.



2. ábra.

A különböző tartalék útvonal számítási módszerek hatékonysága megosztott védelem esetén.

hálózat-típus	mérték	nincs védelem	1 tartalék útvonal			2 tartalék útvonal	
			DP	SP1	SP2	DP	SP2
20/ 4,5	+ kap.	0%	136%	53%	83%	306%	119%
	1 hiba	4,33/10,1	0,0/0,0	0,0/0,0	0,0/0,0	0,0/0,0	0,0/0,0
	2 hiba	8,73/18,7	0,14/6,06	0,31/7,82	0,14/6,08	0,0/0,0	0,0/0,0
	3 hiba	12,9/26,2	0,42/11,0	0,87/13,8	0,43/11,4	0,018/7,05	0,024/8,10
20/ 3,5	+ kap.	0%	147%	61%	94%	351%	141%
	1 hiba	6,83/15,1	0,0/0,0	0,0/0,0	0,0/0,0	0,0/0,0	0,0/0,0
	2 hiba	13,4/27,1	0,38/11,2	0,63/13,8	0,38/11,7	0,0/0,0	0,0/0,0
	3 hiba	19,6/37,1	1,09/20,5	1,73/24,5	1,10/21,1	0,086/13,3	0,100/15,4
50/ 4,5	+ kap.	0%	132%	45%	70%	296%	103%
	1 hiba	2,61/7,49	0,0/0,0	0,0/0,0	0,0/0,0	0,0/0,0	0,0/0,0
	2 hiba	5,18/14,1	0,05/3,62	0,09/4,97	0,05/3,74	0,0/0,0	0,0/0,0
	3 hiba	7,70/20,0	0,14/7,17	0,26/9,23	0,14/7,59	0,004/5,18	0,004/5,66
50/ 3,5	+ kap.	0%	130%	47%	76%	308%	102%
	1 hiba	4,44/12,2	0,0/0,0	0,0/0,0	0,0/0,0	0,0/0,0	0,0/0,0
	2 hiba	8,70/21,3	0,15/7,05	0,25/8,65	0,15/6,84	0,0/0,0	0,0/0,0
	3 hiba	12,7/29,5	0,43/13,8	0,70/16,3	0,43/14,0	0,022/6,42	0,024/8,56

2. táblázat.

A különböző megosztott védelmi módszerekhez szükséges többletkapacitás (+ kap.) és átlagos/maximális kapacitásvesztés értékek [%] 1, 2 és 3 hiba esetén.

Többszörös hálózati hibák elleni megosztott védelem. Annak érdekében, hogy megvizsgáljuk a különböző védelmi stratégiák viselkedését többszörös hibák esetén és összehasonlítsuk a hatékonyságukat, kiterjedt szimulációkat végeztem. A két védelmi útvonalon alapuló védelmi stratégiák vizsgálatához szükséges háromszorosan összefüggő topológiák biztosítását a generálás utáni ellenőrzéssel oldottam meg. Amennyiben nem létezett három független útvonal minden csomópontpár között, új topológia generálás és ellenőrzés következett.

A szimulációk legfőbb eredményei a 2. táblázatban találhatóak, amelyben minden érték 200 mérés átlagát reprezentálja. A táblázatban a többletfoglalás mértéke százalékban, valamint az átlagos és maximális kapacitásvesztés olvasható (forgalomkiesés a forgalmi igényekhez viszonyítva) 1, 2 és 3 hibára feltüntetve különböző hálózattípusok (csomópontszám/átlagos csomóponti fokszám) esetén. Kiderült, hogy a topológia típusa nagyban befolyásolja az eredmények minőségét. A nagyobb hálózatokban ugyanannyi hiba kevesebb kapacitásvesztést okoz. A második megfigyelés szerint a hálózat átlagos csomóponti fokszáma fordítottan arányos a kapacitásmegosztás hatékonyságával. Az SP2 új megosztott védelmi stratégia egy tartalék útvonalas esetben történő alkalmazásával elkerülhető a megosztott védelem okozta degradáció a hibatűrésben. Ennek a javulásnak az ára kb. 30% többletkapacitás az SP1 esethez képest, de a szükséges kapacitás még így is jóval a DP kapacitásszükséglete alatt van. A tartalék útvonalak számát egyről kettőre növelve a kapacitásvesztés kiküszöbölhető két egyidejű hiba esetén, sőt a három hiba elleni védelem is jelentősen javul, azaz az átlagos kapacitásvesztés általában több, mint egy nagyságrenddel csökken az egy tartalék útvonalas esethez viszonyítva. Másrészt a kapacitásfoglalás megosztott védelemmel az egy tartalék útvonalas dedikált védelmi stratégia kapacitásszükséglete alatt van.

3. tézis. Átkonfigurálási sorrend tervezése [C2, C4, J3]

Nagy terheltségű MPLS hálózatokban az LSP kiépítések és lebontások sorozata az LSP-k egy olyan halmazát eredményezheti, ahol az útvonalak nem a lehető legrövidebbek, ami rossz erőforrás-kihasználtságot jelent az optimális esethez képest. Ezért a szolgáltatóknak javasolt globális útvonal-optimalizálást végrehajtani bizonyos időközönként (pl. naponta, hetente) a hálózati teljesítmény fokozása érdekében pl. az 1. tézisben javasolt algoritmus segítségével. Annak érdekében, hogy az LSP-ken zajló forgalom megszakítását elkerüljük a régi utakról új utakra történő átkonfigurálások során, egy LSP átkonfigurálásának a *felépít-mielőtt-megszakít* koncepció szerint a következő lépésekből kell állnia. Először felépítjük az új útvonalat, miközben a forgalom a régi útvonalon zajlik, utána a forgalmat átkapcsoljuk az új útvonalra, végül pedig lebontjuk a régi útvonalat. E művelet sorozat közben (miközben a régi útvonalakat az újakra cseréljük egy bizonyos sorrendben) előfordulhat, hogy néhány LSP nem konfigurálható át az új útvonalakra, mert nincs elegendő foglалható sávszélesség ezen útvonalak valamelyik szakaszán, így ezen LSP-k átkonfigurálása *kapacitáskorlát sértést* okozna. Tehát az *átkonfigurálási sorrendet* az átkonfigurálás művelet sorozata előtt meg kell tervezni, hogy az LSP-k átkonfigurálása a kiszámított sorrendben kivitelezhető legyen, más szóval ne sértsen kapacitáskorlátokat. Ez az átkonfigurálási sorrend tervezési (RSP, reroute sequence planning) eljárás egy új funkciójaként képzelhető el annak a forgalommenedzsment eszközhöz, amely a globális útvonal-optimalizálást végzi. [34]-ben megmutatták, hogy 1000 LSP átkonfigurálása kevesebb, mint 10 másodpercet vesz igénybe, amíg a globális útvonal-optimalizálás után az átkonfigurálandó LSP-k maximális száma 100 körüli volt az előzetes tesztek során. Habár az átkonfigurálás művelete kevés időt vesz igénybe, a konzisztencia érdekében biztosítani kell, hogy az útvonalak konfigurációja ne változzon ez idő alatt. Mivel az RSVP-TE képes dinamikusan átkonfigurálni a felépített LSP-eket [3], ezért ezt az opciót ki kell kapcsolni, amennyiben az LSP-k felépítésére az RSVP-TE-t használjuk. Az átkonfigurálási sorrend tervezésének problémája újkeletű a hálózatok világában, ezért nincs irodalmi előzménye.

3.1. altézis. Az átkonfigurálási sorrend tervezés problémájának megfogalmazása [D6.2]

Az átkonfigurálási sorrend tervezés problémáját először én fogalmaztam meg [C2]-ben, amire most két ekvivalens definíciót adok. A probléma komplexitás szempontjából az NP-teljes kategóriába tartozik, ami a partíciós problémára való redukációjából következik.

Gráf alapú megfogalmazás. Az MPLS hálózatot a $G(V, E)$ irányított gráf reprezentálja, az n elemű V csúcshalmazzal és m elemű $E \subseteq \{(u, v) : u, v \in V, (u \neq v)\}$ irányított élhalmazzal, amelyek rendre az útvonalválasztóknak és fizikai szakaszoknak felelnek meg. Az élek halmazát kiegészítem a $c : E \rightarrow \mathbb{R}^+$ nem-negatív élkapacitás függvényvel, amely a szakaszok összes foglалható sávszélességének a jelképezésére szolgál. Egy P irányított útvonalat az s forráscsomópont és t nyelő csomópont között x számú él következő sorozataként definiálok: $\{e_1 = (s, u_1), e_2 = (u_1, u_2), \dots, e_x = (u_{x-1}, t)\}$. Egy l_i LSP struktúráját pedig az $(s_i, t_i, b_i, P_i, Q_i)$ ötös írja le, ahol P_i és Q_i a régi, illetve új útvonal, melyek mindegyikének forráscsomópontja s_i , nyelőcsomópontja t_i , és sávszélesség-igénye b_i . Megjegyzem, hogy a védelmi útvonalakkal rendelkező LSP-k jelenléte nem változtat a problémán, de ebben az esetben az

LSP-t leíró útvonalaknak (P és Q) az elsődleges és a védelmi útvonalak éleinek összességét kell tartalmazniuk.

A probléma formalizálásának érdekében az LSP-k k elemű $\mathcal{L} = \{l_i : 1 \leq i \leq k\}$ halmaza legyen adott, és bevezetem L_i -t, amely az éleken foglalt kapacitást jelöli az i -edik átkonfigurálás után. Feltételezem, hogy mind a régi, mind az új útvonalrendszer a hozzájuk tartozó kapacitással megfelelő, azaz a kiindulási és végállapotban sem sérül az élek kapacitáskorlátja: $L_0(e) = \sum_{i:e \in P_i} b_i \leq c(e)$ és $L_k(e) = \sum_{i:e \in Q_i} b_i \leq c(e), \forall e \in E$. További feltételezés, hogy $P_i \neq Q_i$ ($1 \leq i \leq k$), hiszen a változatlan útvonalak eliminálásával és a hozzájuk tartozó élkapacitások csökkentésével ekvivalens problémához jutunk, amelyben nincsenek változatlan útvonalak.

Az RSP probléma célja az LSP-k egy $\pi = \{\pi_1, \pi_2, \dots, \pi_k : \pi_i \in \{1, 2, \dots, k\}, i \neq j \Rightarrow \pi_i \neq \pi_j\}$ átkonfigurálási sorrendjének (permutációjának) meghatározása, ami lehetővé teszi az LSP-k átkonfigurálását a kapacitáskorlátok megsértése nélkül, formálisan $\sum_{i:e \in Q_{\pi_i}, i \leq j} b_{\pi_i} + \sum_{i:e \in P_{\pi_i}, i \geq j} b_{\pi_i} \leq c(e), \forall e \notin P_{\pi_j} \cap Q_{\pi_j}, 1 \leq j \leq k$ és $\sum_{i:e \in Q_{\pi_i}, i \leq j} b_{\pi_i} + \sum_{i:e \in P_{\pi_i}, i \geq j} b_{\pi_i} - b_{\pi_j} \leq c(e), \forall e \in P_{\pi_j} \cap Q_{\pi_j}, 1 \leq j \leq k$. Láthatjuk, hogy a kapacitáskorlát feltétel attól függ, hogy $e \in P_{\pi_j} \cap Q_{\pi_j}$ fennáll-e, mert egy LSP régi és új útvonalának közös élein nem kell a szükséges sávszélességet kétszer lefoglalni [35]. Végeredményképp, az első átkonfigurálandó LSP l_{π_1} , az i -edik l_{π_i} , az utolsó pedig l_{π_k} lesz.

Vektor alapú megfogalmazás. Az RSP probléma a *diszkrepancia-elmélet*hez [36] kapcsolódik a következő módon. Legyenek p_i és q_i rendre a P_i és Q_i útvonalakhoz tartozó (egyenként m elemű) incidencia vektorok: $p_i(e) = 1$, ha $e \in P_i$, különben $p_i(e) = 0$. Következésképpen az l_i LSP átkonfigurálását az $y_i = b_i \cdot q_i - b_i \cdot p_i$ vektor írja le, ahol a vektor komponensei a kapacitásfogalás nettó változását reprezentálják az egyes éleken. Így az $L_0(\leq c)$ kezdeti vektor definíciója, amely a kezdeti kapacitásfogalásoknak felel meg, és az $\{y_i : 1 \leq i \leq k\}$ vektorhalmaz meghatározása után a feladat a y_i vektorok egy $\pi = \{\pi_1, \pi_2, \dots, \pi_k\}$ permutációjának kiszámítása azzal a feltétellel, hogy egyik részösszegük se haladja meg c -t egyik komponensében sem. Formálisan, határozzuk meg az $1, 2, \dots, k$ egy $\pi = \{\pi_1, \pi_2, \dots, \pi_k\}$ permutációját úgy, hogy

$$L_0(e) + \sum_{i=1}^j y_{\pi_i}(e) \leq c(e) \quad \forall e \in E, 1 \leq j \leq k. \quad (4)$$

3.2. altézis. Algoritmusok az átkonfigurálási sorrend tervezés problémájának megoldására [D6.4]

Heurisztikus algoritmusokat javaslok az RSP probléma megoldására, amelyek alapja egy iteráció, amelyben egy kiválasztott LSP átkonfigurálása történik. Az algoritmusoknál előforduló átkonfigurálás kifejezés nem a valódi LSP átkonfigurálása az MPLS hálózatban, hanem csak a gráfon végzett módosításokat jelenti; a valódi átkonfigurálás az algoritmusok lefutása után végeredményként adott sorrend szerint történik. Az algoritmusok leállási feltétele az összes LSP kiválasztása, azaz az átkonfigurálási sorrend egyesével, mohó módon történő felépülése. Az algoritmusok legfontosabb eleme az átkonfigurálandó LSP kiválasztása, amely az algoritmusok különbözőségének alapja. Közös tulajdonsága a módszereknek, hogy a választás alapját a sértés

nélkül átkonfigurálható LSP-k adják, és amennyiben nincs ilyen LSP, akkor olyat választanak, amely sérti az élkapacitásokat, de a sértést alacsony szinten próbálják tartani. Az algoritmusok egy o_i *hasznossági értéket* rendelnek az összes átkonfigurálandó l_i LSP-hez minden iterációs lépésben, és a legnagyobb hasznossági értékkel rendelkező LSP-t választják ki. A különböző algoritmusok az o_i hasznossági értékeket következőképpen határozzák meg.

Véletlen rendezés (RS, Random Sorting). Az RS véletlenszerűen választja ki az LSP-
ket: o_i egy véletlenül generált valós szám a $[0;1)$ intervallumból. Ez egy nagyon egyszerű módszer, de gyors és jó kiindulási alapot szolgáltat az összehasonlításokhoz, továbbá az o_i értékek hozzárendelése konstans időben végrehajtható.

Legkisebb sértés (MV, Minimal Violation). Az MV a legegyszerűbb mohó módszer. Minden átkonfigurálandó l_i LSP-re kiszámolja, mekkora lenne a legnagyobb kapacitássértés az új élein (amelyek különböznek a régi élektől), ha átkonfigurálnánk: $o_i = -\max_{e \in Q_i \setminus P_i} \{b_i + r(e) - c(e)\}$, ahol $r(e)$ az e élen aktuálisan foglalt kapacitás. Amennyiben egy vagy több él kapacitása sérül az l_i LSP átkonfigurálása esetén $o_i < 0$, különben $o_i \geq 0$. Az ötlet e sorrend felállításá mögött, hogy a legkisebb szabad kapacitást a legkevésbé csökkentjük, illetve elkerülhetetlen sértés esetén a legkisebb kapacitássértést okozzuk.

Legnagyobb felszabadítás (MF, Maximal Freeing). Az MF segédváltozóként az $a(e)$ kapacitáértéket használja, amely megmutatja, hogy mennyi a kapacitásszükséglete az e élnek az elkövetkezendő LSP-k átkonfigurálásához. Formálisan $a(e) = \sum_{j: e \in Q_j \setminus P_j} b_j$ minden olyan j -re, amelyre l_j még nincs átkonfigurálva. Itt o_i az elkövetkezendő átkonfigurálásokhoz szükséges felszabadítandó kapacitásmennyiséget jelöli l_i régi útvonalának élein: $o_i = \sum_{e \in P_i \setminus Q_i} \{a(e) - c(e) + r(e)\}$. Ez a szabály előnyben részesíti azon LSP-
ket, amelyek olyan éleket tartalmaznak a régi útvonalaik mentén, amely sok átkonfigurálandó LSP új útvonalának része és relatíve kevés szabad kapacitás van rajta.

Legtöbb átkonfigurálható (MR, Most Reroutable). Az MR esetén o_i az l_i átkonfigurálása utáni kapacitássértés nélkül átkonfigurálható LSP-k számát reprezentálja. Ezt a megközelítést használva egy LSP kiválasztása után a következő lépésben a legtöbb átkonfigurálható LSP közül választhatunk. Ebben az esetben nagy valószínűséggel több, mint egy LSP-nek lehet ugyanaz a legnagyobb hasznossági értéke, ezért ezt az értéket következőképpen módosítjuk: l_i átkonfigurálása esetén fennmaradó minden egyes átkonfigurálandó LSP-re a nem átkonfigurálható élek és összes élek arányának összegét levonjuk o_i -ből. Ennek a megközelítésnek fontos előnye az, hogy figyelembe veszi a következő átkonfigurálási lépést. Másrészről pedig az előző algoritmusokhoz képest nagyobb számításigényű, ezért az alkalmazása időigényesebb a többiekénél.

Algoritmus változatok. Az előzőekben ismertetett mohó algoritmusok tökéletesítése a következő tényen alapul. A normál RSP probléma – az LSP-k átkonfigurálása a P utakról a Q utakra – ekvivalens a fordított problémával, ahol a feladat minden l_i LSP-t átkonfigurálni a Q_i útvonalról a P_i útvonalra, továbbá a kezdeti és végső kapacitásfoglalás fel van cserélve.

Ennek indoklása, hogy amennyiben létezik megfelelő sorrend a fordított problémára, a fordított sorrend egy jó megoldás a normál problémára, és ez visszafelé is igaz. Ezek alapján az algoritmusok három változatát mutatom be.

1. A legegyszerűbb változat megkísérelni mind a normál, mind a fordított probléma megoldását. Először a könnyebb problémával próbálkozunk, amelyik általában a kevesebb összkapacitású kiinduló állapotból történő átkonfigurálás – ez a tipikus valós körülmények között, hiszen az útvonal-optimalizálás célja a hálózati szabad kapacitás növelése.
2. Az algoritmusok második változatában a sorrendet a két végéről egyszerre építjük fel, vagyis az átkonfigurálandó LSP-k felváltva származnak a normál és fordított probléma megoldási lépéseiből.
3. A legbonyolultabb változat: visszalépés alkalmazása az algoritmusokban. Az algoritmus elkezdődik és amint elakad, azaz nincs több kapacitássértés nélkül átkonfigurálható LSP, kitorli egy részét a legutoljára választott LSP-nek a sorozatból (és visszakonfigurálja azokat a régi útvonalaikra), majd a fordított probléma megoldásával folytatja a működését. Amennyiben ismét elakad, megint visszalép és irányt vált. A végtelen ciklusok elkerülése érdekében minden visszalépési fázisnál kevesebb LSP-t töröl a sorból, mint amennyit kiválasztott az utolsó kiválasztási fázisban. A jelenlegi megvalósításban a törölendő LSP-k és az utolsó fázisban kiválasztott LSP-k számának aránya 0,95-ra van állítva (amely értéket finomhangolással határoztam meg).

Alternatív megoldások. Vizsgálataim során kiderült, hogy léteznek olyan szituációk, amelyekre az előbbieken ismertetett algoritmusok egyik változata sem talál megfelelő átkonfigurálási sorrendet. Ilyen esetekben két lehetőség kínálkozik a hálózati operátor számára: (1) nem konfigurálja át az LSP-eket a jobb útvonalakra, vagy (2) megpróbálja átkonfigurálni az LSP-eket a forgalom megszakítása, illetve degradációjának árán. Az utóbbi esetet követően először meghatározunk egy átkonfigurálási sorrendet, amely kapacitáskorlátokat sérthet, majd elimináljuk azokat a következő két módszer valamelyikével.

- A *megszakítás* módszere megengedi néhány LSP-nek a lebontását az átkonfigurálás idejére. Az átkonfigurálási sorrend meghatározása után kiválasztunk LSP-eket, amelyek sávszélességének összege a régi útvonalak mentén lefedi az összes kapacitássértést. Az átkonfigurálási műveletet ezek után a kiválasztott LSP-k lebontásával kell kezdeni, majd a meghatározott sorrendben végre kell hajtani az átkonfigurálási műveletsorozatot.
- Az *összenyomás* módszere az LSP-k foglalt sávszélességét csökkenti az átkonfigurálás idejére, de szerencsés esetben ez a csökkenés olyan kicsi, hogy a felhasználók számára észrevétlen. Ennél a módszernél is az átkonfigurálási sorrend meghatározása az első lépés. Ezután az LSP-k sávszélességének csökkentése következik. Ez legegyszerűbb esetben egyenlő arányban történik: az LSP-k foglalt sávszélesség értékeit megszorozzuk az $1/(1+x)$ kifejezéssel, ahol x a legnagyobb sértés (amelyet a teljesítményvizsgálatnál pontosan definiálunk). Ezután az átkonfigurálás következik a megtervezett sorrendben, végül az LSP-k sávszélességét visszaállítjuk az eredeti értékükre.

3.3. altézis. Az átkonfigurálási sorrend tervező algoritmusok teljesítményvizsgálata [D6.5]

Az átkonfigurálási sorrend tervező algoritmusok teljesítményét valós topológiákon véletlenszerűen előállított hálózati szituációkon szimuláció segítségével vizsgáltam. Törekedtem a minél nehezebb, de megoldható problémapéldányok előállítására, ezért egy kiindulási hálózati konfiguráció létrehozása után az élek kapacitását pontosan a lefoglalt kapacitásokra állítottam, azaz az élek szabad kapacitása nulla volt. Ezután előre meghatározott számú LSP-t konfiguráltam át egyesével új útvonalra úgy, hogy amennyiben az átkonfigurálás kapacitáskorlátot sértett volna, az élek kapacitását megnőveltem a minimálisan szükséges értékkel. Ezáltal „szoros” problémapéldányokhoz jutottam, amelyek megoldása is a rendelkezésemre állt. 1000 problémapéldányt vizsgáltam meg, amelyeknél a csomópontok száma, az átlagos csomóponti fokszám, valamint az átkonfigurálandó LSP-k száma rendre a [10,50], [3,6], valamint a [20,100] intervallumokból kerültek véletlenszerű kiválasztásra. Négy mértéket határoztam meg a módszerek összehasonlításához, amelyek kiszámításához az átkonfigurálási sorrend tervezése közbeni élsértések feljegyzése szükséges:

- *sikerességi valószínűség*: a kapacitássértés nélkül megoldott és az összes vizsgált teszt-példány számának aránya,
- *legnagyobb sértés*: az átkonfigurálás során fellépő legnagyobb, az él összesen foglalható kapacitásához viszonyított élkapacitás sértés,
- *sértett élek számának aránya*: a sértett élek és az összes él számának (m) aránya és
- *kapacitássértés arány*: az éleken előforduló legnagyobb sértések abszolút nagyságának összege a hálózati összkapacitáshoz ($\sum_{e \in E} c(e)$) viszonyítva.

A vizsgálatok során az algoritmusokat a teljesen véletlenszerű sorrenddel (RO, Random Order) is összehasonlítottam, amely az RS-től annyiban tér el, hogy a sorrend felépítése során akkor is választhat sértés nélkül át nem konfigurálható LSP-t, ha még vannak sértés nélkül átkonfigurálható LSP-k.

Az alapalgoritmusok négy vizsgált mérték szerinti eredményei a 3. táblázatban olvashatók. Meglepő eredmény, hogy az RO sikerességi valószínűsége szinte nulla, vagyis gyakorlatilag egy problémapéldányt sem tudott megoldani. Másrészt az RS – amely egy módosított változata az RO-nak – a példányok több, mint felét megoldotta. Az MR minden mérték szerint sokkal jobbnak bizonyult az összes többi algoritmusnál. Ennek oka az lehet, hogy figyelembe veszi a soron következő átkonfigurálási lépést. Az RS, MV és MF módszer közel azonos eredményt produkált, de érdekes módon az RS volt a legjobb közülük sikerességi valószínűségben és a sértett élek számának arányában. A MV volt a legjobb a három algoritmus közül legnagyobb sértésben, valamint kapacitássértés arányban, és az MF volt a legrosszabb minden tekintetben. Fontos megjegyezni azonban, hogy ezek az eredmények nagymértékben függenek a teszt-példány generálástól, így más környezetben az algoritmusok teljesítőképesége eltérhet.

A 4. táblázat az algoritmusok különböző változatainak sikerességi valószínűségét mutatja. A táblázat néhány változat-kombinációt is tartalmaz, amely azt jelenti, hogy egy teszt-példány megoldása több algoritmus változattal történt. Az első és harmadik változatát nem érdemes

algoritmus	sikerességi valószínűség	legnagyobb sértés	sértett élek számának aránya	kapacitás-sértés arány
RO	0,1	38,29	27,23	1,7224
RS	55,0	3,95	1,12	0,0419
MV	51,6	1,72	1,34	0,0239
MF	50,4	5,93	1,53	0,0572
MR	81,8	1,50	0,37	0,0044

3. táblázat.

Az átkonfigurálási sorrend tervező alapalgoritmusoknak a négy definiált mérték szerinti teljesítménye [%].

kombinálni, mert ha a harmadik nem tudja megoldani, akkor az első sem, mivel a harmadik változat az első általánosított változata. Az algoritmusok második változatai adták a legrosszabb eredményeket, mert a vizsgált „szoros” problémáknál jobb az egyszerűbb probléma megoldásával felépíteni a sorrendet, mint a könnyebb és nehezebb probléma közötti változtatással. A harmadik változat növelte a sikerességi valószínűséget a legnagyobb mértékben, és csak néhány tesztpéldány volt, amelyet a második változat meg tudott oldani, de a harmadik nem (vö. a 3-as és a 2+3-as oszlopokat).

algoritmus	változat				
	1	2	3	1+2	2+3
RS	55,0	22,2	76,4	59,0	77,3
MV	51,6	19,7	63,4	53,3	63,5
MF	50,4	26,2	71,1	55,0	71,4
MR	81,8	75,7	92,1	88,3	92,6

4. táblázat.

Az átkonfigurálási sorrend tervező algoritmusok különböző változatainak sikerességi valószínűsége [%].

4. tézis. Hálózattervezés a forgalomeloszlás változásának figyelembevételével [C8, J4]

Az előző téziseimben megmutattam, hogy az MPLS-TE koncepció alkalmazásával az LSP-k útvonalai explicit módon megadhatók és átkonfigurálhatók a hálózat működése közben. A napi és/vagy heti forgalomváltozásokhoz tartozó átkonfigurálásokat *kapacitásmenedzsmentnek* hívják, amely a Nemzetközi Távközlési Szövetség Távközlési Szabványosítás Szektorának (ITU-T, International Telecommunication Union – Telecommunication Standardization Sector) egyik ajánlása [37]. A kapacitásmenedzsmentnek egyik módszere a forgalomeloszlás változásának figyelembevételével történő hálózattervezés, amely figyelembe veszi a forgalmi útvonalak átkonfigurálhatóságát már a tervezés fázisában. Ily módon a forgalom térbeli és időbeli periodikus változása hatékonyan kezelhetővé válik. Tehát a hálózatot nem arra az esetre kell méretezni, amikor a forgalom nagysága a legnagyobb minden forrás–nyelő páros között, hanem

az időintervallumot több részre felosztva mindegyik intervallumhoz külön forgalmi igény halmaz rendelhető. Így kisebb a kapacitásszükséglet, azaz a hálózat kiépítési költsége kevesebb lehet.

Hasonló problémát már vizsgáltak az irodalomban ATM hálózati környezetben. Bauschert [38]-ben a problémát három alproblémára bontja és az iteratív megoldás közben kettőt mindig állandónak tekint közülük. Medhinek több módszere [39, 40, 41] van a problémára, továbbá a védelem kérdését is megvizsgálta. Azonban ezek a megközelítések nem lépcsős költségfüggvényeket használnak vagy a csomópontpárok közötti lehetséges útvonalakat előre meghatározott halmazokra szűkítik. Ezért hasznos egy új, a forgalomeloszlás változásait figyelembe vevő hálózattervező algoritmus kidolgozása, amely képes kihasználni a lépcsős költségfüggvények sajátosságait és tetszőleges útvonalat választhat a csomópontpárok között.

4.1. altézis. A forgalomeloszlás változásait figyelembe vevő új hálózattervező algoritmus [D7.3]

Egy új algoritmust javaslok a forgalomeloszlás változásait figyelembe vevő hálózattervezésre. A javasolt CND^{MH} (multi-hour core network design) algoritmus a forgalomeloszlás változásai-val nem foglalkozó hatékony CND (core network design) algoritmuson [C5] alapul és ugyanúgy három fázisból épül fel. A CND^{MH} fázisai összetettebbek a bonyolultabb probléma miatt, amelyet a több időintervallumnak megfelelő több forgalmi igény halmaznak köszönhető. Az első fázis a kezdeti kapacitásbecslés, amely megjósolja a tervezendő hálózat közelítőleges kapacitásszükségletét. Utána az iteratív útvonal-optimalizálás következik, amely egy megfelelő hálózati konfigurációt biztosít az előző fázis végeredményét felhasználva. Az utolsó fázis, az utólagos kapacitásfinomítás, az eredmények finomítására szolgál egy olcsóbb megoldás elérése érdekében. Megjegyzem, hogy a középső a fő fázis, amely képes érvényes megoldást biztosítani a problémára, míg az első és harmadik fázis használata opcionális, azonban érdemes azokat is alkalmazni, mert nagymértékben hozzájárulnak a jó megoldáshoz. Továbbá fontos még, hogy abban a speciális esetben, amikor az időintervallumok száma 1, a CND^{MH} mindegyik fázisa ugyanúgy működik, mint a forgalomeloszlás változásait figyelmen kívül hagyó tervező (CND) megfelelő fázisa.

Kezdeti kapacitásbecslés (ICE^{MH} , Initial Capacity Estimation). Az ICE^{MH} célja megjósolni a hálózati eszközök szükséges kapacitását a forgalmi igények halmazának vizsgálatával. Ez a fázis részlegesen méretezett hálózatot szolgáltat a következő fázisnak, így a fennmaradó feladatok az eszközök méretezésének befejezése, valamint az útvonalrendszerek meghatározása a különböző időintervallumokhoz tartozó forgalmi igények halmazához. Habár az ICE^{MH} nem ad megfelelő megoldást, egy jó kiinduló állapotot határoz meg a problémapéldányról alkotott globális kép alapján.

Az ICE^{MH} működése a forgalmi igényekhez az egymás után véletlen sorrendben történő útvonal hozzárendeléseken alapul. Minden forgalmi igény halmazhoz egymás után többször (egy előre meghatározott számszor) útvonalrendszereket rendelünk legrövidebb utat kereső algoritlussal. Feltételezem, hogy ha az összes véletlen útvonalrendszer igényelt egy bizonyos mennyiségű kapacitást egy adott eszközön, akkor nagy valószínűséggel az optimális megoldás is legalább ennyit igényel. Ezek alapján, az eszközök kapacitásértékeit úgy határozom meg, hogy azok a különböző útvonalrendszereknél felmerülő legkisebb értékek legyenek.

Iteratív útvonal-optimalizálás (IRO^{MH}, Iterative Routing Optimization). Az IRO^{MH} az algoritmus fő fázisa, ami egy teljes megoldást tud szolgáltatni az első és harmadik fázis nélkül is. Ennek a fázisnak az alapja egy olyan algoritmus, amely képes egy útvonalrendszert meghatározni egy adott forgalmi igény halmazhoz egy kapacitáskorlátokkal rendelkező gráfban. Erre a célra az 1. tézisben bemutatott globális útvonal-optimalizálási algoritmust használok, amely nagy valószínűséggel talál megoldást egy adott problémapéldányra, amennyiben annak létezik megoldása. Mindezek mellett fontos tulajdonsága az IRO^{MH}-nak, hogy ez az útvonal-optimalizáló helyettesíthető egy ugyanezt a feladatot megoldó másik algoritmussal.

Az IRO^{MH} egy kétlépéses iteráción alapul, amely első lépésben megpróbál útvonalrendszereket meghatározni a forgalmi igény halmazokhoz az adott globális útvonal-optimalizáló segítségével, figyelembe véve az aktuális kapacitáskorlátokat. Ha az algoritmus sikeresen fejeződik be az összes forgalmi igény halmazra, az IRO^{MH} leáll. Különben második lépésként egy bizonyos eszköz kapacitását egy „lépcsővel” növeli, majd az első lépésre ugrik. A bővítendő eszköz kiválasztása úgy történik, hogy a forgalmi igényekhez, amelyekhez nem sikerült az útvonal-optimalizálóknak útvonalakat rendelni a kapacitáskorlátokon belül, a korlátok figyelmen kívül hagyásával rendel útvonalakat, végül a legnagyobb kapacitáskorlát túllépéssel rendelkező eszköz lesz a bővítendő.

Utólagos kapacitásfinomítás (PCR^{MH}, Posterior Capacity Refinement). Habár az IRO^{MH} fázisa után rendelkezésre áll egy megfelelő megoldás, ez a megoldás a PCR^{MH} segítségével finomítható. Ez a fázis egy lokális keresési eljárás alapján, vagyis az eljárás egyszerre csak a hálózat egy részére koncentrál. A PCR^{MH} ötlete, hogy csökkentse a kis *relatív lépcső-kihasználtságú* eszközök kapacitását. A relatív lépcső-kihasználtság a szükséges és tényleges kapacitás aránya az aktuális kapacitáslépcsőn (amely az eszköz aktuális költségéhez tartozik).

A PCR^{MH} először az eszközöket átlagos relatív lépcső-kihasználtságuk szerint rendezi (ahol az átlagolás a különböző időintervallumokra vonatkozik), majd veszi az első eszközt. Az aktuális eszköz kapacitását egy lépcsővel csökkenti, és a forgalmi igényekhez megpróbál útvonalrendszert rendelni ezen szorosabb kapacitáskorlátok között az IRO^{MH} segítségével. Ha minden időintervallumra sikerült útvonalrendszert meghatározni költségnövekedés nélkül, akkor a PCR^{MH} újraindul, különben pedig az algoritmus az aktuális eszköz kapacitását visszaállítja és a soron következő eszközt próbálja „összenyomni”. Miután az utolsó eszközt sem sikerült kisebb kapacitására állítani, a PCR^{MH} befejeződik.

4.2. altézis. A javasolt hálózat-tervező algoritmus teljesítményvizsgálata [D7.4]

A javasolt hálózat-tervező algoritmus teljesítményét szimuláció segítségével vizsgáltam. Különböző méretű véletlenszerűen generált hálózati topológiákat és két valós (egy európai és egy USA-beli) topológiát vettem alapul. Egyik fő jellemzője a forgalomeloszlás változásait figyelembe vevő hálózat-tervezésnek, hogy a csomópontok közötti sáv szélesség-igény változik az időben. Több forgalmi igény halmaz létrehozásának érdekében egy új igény generálási eljárást vezettem be. A hálózati topológiát bizonyos számú régióra osztottam. Egy konkrét időintervallumban minden régiót párba állítottam pontosan egy másik régióval. A Δ paraméterrel megadott része az adott régió kimenő forgalmának a régió párja felé irányult, a forgalom maradék része pedig egyenletesen oszlott el az összes régió között (beleértve magát az adott régiót és a régió párját is). Így $\Delta = 0$ azt jelenti, hogy nincs extra forgalom a meghatározott régiópárok

között, és $\Delta = 100\%$ esetén pedig az összes forgalom a régiópárok között zajlik. Egy adott időintervallum forgalmi igényeinek generálásához a [C5]-ben részletezett forgalom generálási módszert használtam első lépésként, majd egy adott régióra vonatkozó sáv szélesség értékeket a Δ paraméter szerint skáláztam át úgy, hogy a sáv szélesség értékeinek összege ne változzon. Az időintervallumok közötti forgalmi igény különbségeket a régiópárok változtatásával értem el.

A megbízható eredmények elérése érdekében különböző forgalmi szituációkat teszteltem az egyenletesen elosztottól ($\Delta = 0\%$) egészen a teljesen polarizáltig ($\Delta = 100\%$). Két, a forgalomeloszlás változásait figyelmen kívül hagyó CND algoritmuson alapuló referenciaalgoritmust vezettem be. A TBM (traffic bandwidth maximizer) algoritmus kiszámítja az egyes csomópont párok között fellépő legnagyobb igényt, és erre tervezi meg a hálózatot a CND segítségével, míg az LCM (link capacity maximizer) algoritmus az összes időintervallumra megtervez egy hálózatot a CND-vel, majd az eredményül kapott kapacitásértékek közül a legnagyobbakat választja minden eszközre.

Az 5. táblázat tartalmazza a hálózati összköltség (a legjobb megoldás százalékában) és futási idő eredményeket különböző hálózatméretekre és -típusokra. Az új algoritmus általában jobban teljesített a referenciaalgoritmusoknál. A kivételes esetek az egyenletes eloszlású forgalom mellett a nagyobb hálózatméretekénél voltak, ahol a javasolt algoritmus a TBM referenciaalgoritmusnál kicsit gyengébben teljesített. Azonban ez nem mond ellent az elvárásaimnak, hiszen a forgalomeloszlás változásait figyelembe vevő hálózattervezésnek akkor van értelme, ha a különböző időintervallumokban bizonyos mértékű különbség figyelhető meg a forgalom eloszlásában. Az algoritmusok futási ideje még nagyobb hálózatok esetén is mérsékeltnek mondható, figyelembe véve, hogy egy nem valósidejű hálózattervezési feladatot oldanak meg. Összegzőként elmondható, hogy a javasolt forgalomeloszlás változásait figyelembe vevő hálózattervező algoritmus gazdaságos megoldást szolgáltat elfogadható futási időn belül a vizsgált hálózati szituációkra.

5. Az eredmények alkalmazása

Az összes kutatási munkámat az Ericsson Magyarország Kft. projektjeinek keretén belül végeztem. Az 1–3. tézisek algoritmusai egy komplex forgalommenedzsment eszköz egyes funkcióiként szolgálhatnak. Az 1. tézis globális útvonal-optimalizálója és a 3. tézis átkonfigurálási sorrend tervezője az Ericsson egy IP menedzsment eszközében kerültek megvalósításra. A 2. tézisben ismertetett új megosztott védelmi megközelítések a hálózati operátorok használatába kerülhetnek a közeljövőben, amint a hálózati technológiák lehetővé teszik azokat. A 4. tézis forgalomeloszlás változásait figyelembe vevő hálózattervező algoritmus a Magyar Oktatási Minisztérium IKTA-0092/2002 projektjében alapozódott meg, amely egy kutatási együttműködés az Ericsson Magyarország Kft., Budapesti Műszaki és Gazdaságtudományi Egyetem, valamint a Kovax 95 Kft. között. Valóság-hű költségmodelljének köszönhetően jól alkalmazható valós hálózattervezési problémákra.

Irodalomjegyzék

- [1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, X. Xiao: „Overview and Principles of Internet Traffic Engineering”, Internet Engineering Task Force, Request For Comments (Proposed

hálózat- méret	algoritmus	Δ					idő [perc] min-max (átlag)
		0%	25%	50%	75%	100%	
15 csomó- pont	CND ^{MH}	100,00	100,00	100,00	100,00	100,00	0–3 (< 1)
	TBM	100,17	107,32	123,10	145,80	152,23	0–1 (< 1)
	LCM	105,75	108,59	113,76	125,29	126,65	0–2 (< 1)
25 csomó- pont	CND ^{MH}	100,67	100,00	100,00	100,00	100,00	1–20 (7)
	TBM	100,00	104,11	119,12	139,09	144,20	1– 8 (2)
	LCM	106,57	106,94	108,61	114,87	118,20	1–37 (6)
35 csomó- pont	CND ^{MH}	104,83	100,00	100,00	100,00	100,00	1–61 (25)
	TBM	100,00	102,62	116,83	134,33	149,62	1–29 (10)
	LCM	108,25	106,80	107,79	112,34	121,88	1–58 (30)
Európa	CND ^{MH}	100,00	100,00	100,00	100,00	100,00	0–1 (< 1)
	TBM	111,42	106,86	120,06	129,85	128,74	0–1 (< 1)
	LCM	126,74	122,54	127,83	128,40	115,19	0–1 (< 1)
USA	CND ^{MH}	100,00	100,00	100,00	100,00	100,00	0–2 (< 1)
	TBM	108,08	109,14	115,12	126,67	134,96	0–1 (< 1)
	LCM	121,11	121,98	123,12	123,69	117,57	0–2 (< 1)

5. táblázat.

A javasolt hálózattervező algoritmus összköltség értékei (a legjobb talált megoldáshoz képest) [%] és futási idő [perc] eredményei.

Standard) 3272, May 2002.

- [2] E. Rosen, A. Viswanathan, R. Callon: „Multiprotocol Label Switching Architecture”, Internet Engineering Task Force, Request For Comments (Proposed Standard) 3031, January 2001.
- [3] D. O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow: „RSVP-TE: Extensions to RSVP for LSP Tunnels”, Internet Engineering Task Force, Request For Comments (Proposed Standard) 3209, December 2001.
- [4] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, and A. Malis: „Constraint-Based LSP Setup using LDP”, Internet Engineering Task Force, Request For Comments (Proposed Standard) 3212, January 2002.
- [5] M. R. Garey, D.S. Johnson: „Computers and Intractability: A Guide to the Theory of NP-Completeness”, Freeman, San Francisco, CA, 1979.
- [6] T. Cinkler, P. Laborczi, and Á. Horváth: „Protection through Thrifty Configuration”, Proc. of 16th International Teletraffic Congress (ITC 16), Edinburgh, UK, pp. 975–987, June 1999.
- [7] J. Kleinberg: „Single-source unsplittable flow”, Proc. of 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1996), Burlington, Vermont, USA, pp. 86–95, October 1996.

- [8] S. G. Kolliopoulos and C. Stein: „Improved approximation algorithms for unsplittable flow problems”, Proc. of 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1997), Miami Beach, Florida, USA, pp. 426–435, October 1997.
- [9] S. Van den Bosch, F. Poppe, H. De Neve, and G. H. Petit: „Multi-Objective Traffic Engineering of IP Networks using Label Switched Paths”, Proc. of International Telecommunications Network Strategy and Planning Symposium (Networks 2000), Canada, September 2000.
- [10] T. Leighton, F. Makedon, S. Plotkin, C. Stein, É. Tardos, and S. Tragoudas: „Fast approximation algorithms for multicommodity flow problems”, Proc. of 23th Annual ACM Symposium on Theory of Computing (STOC 1991), New Orleans, Louisiana, USA, pp. 101–111, May 1991.
- [11] S. Plotkin, D. Shmoys, and É Tardos: „Fast approximation algorithms for fractional packing and covering problems”, Mathematics of Operations Research, vol. 20, pp. 257–301, 1995.
- [12] N. Garg and J. Könemann: „Faster and simpler algorithms for multicommodity flow and other fractional packing problems”, Proc. 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1998), Palo Alto, California, USA, pp. 300–309, November 1998.
- [13] L. K. Fleischer: „Approximating fractional multicommodity flow independent of the number of commodities”, Proc. of 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1999), New York City, New York, USA, pp. 24–31, October 1999.
- [14] P. Gajowniczek, M. Pióro, Å. Arvidsson: „VP Reconfiguration through Simulated Allocation”, 13th Nordic Teletraffic Sem., Trondheim, Norway, pp. 251–260, 1996.
- [15] M. Pióro, P. Gajowniczek: „Simulated Allocation: a Suboptimal Solution to the Multi-commodity Flow Problems”, 11th UK Teletraffic Symposium, 1994.
- [16] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts: „On-line machine scheduling with applications to load balancing and virtual circuit routing”, Proc. of 25th Annual ACM Symposium on Theory of Computing (STOC 1993), San Diego, California, USA, pp. 623–631, May 1993.
- [17] P. Aukia, M. Kodialam, P. V. Koppol, T. V. Lakshman, H. Sarin, and B. Suter: „RATES: a server for MPLS traffic engineering”, IEEE Network Magazine, vol. 14, no. 2, pp. 34–41, March-April 2000.
- [18] R. A. Guerin, A. Orda, and D. Williams: „QoS Routing Mechanisms and OSPF Extensions”, Proc. of IEEE Global Telecommunications Conference (Globecom 1997), Phoenix, Arizona, November 1997.
- [19] S. Plotkin: „Competitive routing of virtual circuits in ATM networks”, IEEE Journal Selected Areas in Communications, vol. 13, no. 7, pp. 1128–1136, September 1995.

- [20] M. Kodialam, and T. V. Lakshman: „Minimum interference routing with applications to MPLS traffic engineering”, Proc. of Conference on Computer Communications (IEEE Infocom 2000), Piscataway, NJ, USA, pp. 884–893, March 2000.
- [21] R. K. Ahuja, T. L. Magnanti, J. B. Orlin: „Network Flows: Theory, Algorithms, and Applications”, Prentice-Hall, New Jersey, 1993.
- [22] E. W. Zegura, K. L. Calvert, M. J. Donahoo: „A Quantitative Comparison of Graph-Based Models for Internet Topology”, IEEE/ACM Transactions on Networking 5(6), pp. 770–783, 1997.
- [23] K. L. Calvert, E. W. Zegura, M. Doar: „Modeling Internet Topology”, IEEE Communications Magazine, June 1997.
- [24] M. Faloutsos, P. Faloutsos, C. Faloutsos: „On Power-Law Relationships of the Internet Topology”, Proc. of ACM SIGCOMM ’99, Cambridge, United States, pp. 251–262, 1999.
- [25] B. M. Waxman: „Routing of Multipoint Connections”, IEEE Journal on Selected Areas in Communications 6(9), pp. 1617–1622, December 1988.
- [26] W. D. Grover, J. Doucette, M. Clouqueur, D. Leung, D. Stamatelakis: „New Options and Insights for Survivable Transport Networks”, IEEE Communications Magazine, Vol. 40, No. 1, pp. 34–41, January 2002.
- [27] R. Kawamura, K. Sato, I. Tokizawa: „Self-Healing ATM Networks Based on Virtual Path Concept”, IEEE Journal on Selected Areas in Communications, Vol. 12, No. 1, pp. 120–127, 1994.
- [28] Y. Xiong, L. Mason: „Restoration Strategies and Spare Capacity Requirements in Self-Healing ATM Networks”, Proc. of Infocom 97, Kobe, Japan, April 1997.
- [29] Y. Zheng, W. D. Grover, and M. H. MacGregor: „Dependence of Network Capacity Requirements on the Allowable Flow Convergence Overloads in ATM Backup VP Restoration”, Electronics Letters, vol. 33, no. 5, pp. 362–363, 27th February 1997.
- [30] W. D. Grover and Y. Zheng: „VP-Based ATM Network Design with Controlled Over-Subscription of Restoration Capacity”, Proc. of 1st Int’l Workshop on Design of Reliable Communication Networks (DRCN 1998), Brugge, Belgium, paper O.33, May 1998.
- [31] S. Kini, M. Kodialam, T. V. Lakshman, and C. Villamizar: „Shared backup Label Switched Path restoration”, Internet Engineering Task Force, Internet Draft, November 2000. (Work in Progress)
- [32] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot: „Analysis of link failures over an IP backbone”, Proc. of 2nd ACM SIGCOMM Internet Measurement Workshop, Marseilles, France, November 2002.
- [33] D. A. Schupke, A. Autenrieth, and T. Fischer: „Survivability of Multiple Fiber Duct Failures”, Proc. of 3th IEEE International Workshop on Design of Reliable Communication Networks (DRCN 2001), Budapest, Hungary, pp. 213–219, October 2001.

- [34] C. Rossenhövel: „Alcatel Stars in Multiservice Switch Test”, Light Reading, Inc., Technical Report, Section 8, November 2002.
- [35] J. Ash, Y. Lee, P. Ashwood-Smith, B. Jamoussi, D. Fedyk, D. Skalecki, L. Li: „LSP Modification Using CR-LDP”, Internet Engineering Task Force, Request For Comments (Proposed Standard) 3214, January 2002.
- [36] J. Beck, V. T. Sós: „Discrepancy Theory”, Handbook of Combinatorics, R. Graham, M. Grötschel and L. Lovász eds., Elsevier, pp. 1405–1446, 1995.
- [37] ITU-T: „QoS Routing and Related Traffic Engineering Methods – Capacity Management Methods”, Recommendation E.360.6, CA, 2002.
- [38] T. Bauschert: „Multihour design of multi-hop virtual path based widearea ATM networks”, Proc. of the 15th International Teletraffic Congress (ITC-15), Washington DC, USA, pp. 1019–1029, 1997.
- [39] D. Medhi: „A Unified Approach to Network Survivability for Teletraffic Networks: Models, Algorithms and Analysis”, IEEE Transactions on Communications, vol. 42, pp. 534–548, 1994.
- [40] D. Medhi: „Multi-Hour, Multi-Traffic Class Network Design for VP-based Wide-Area Dynamically Reconfigurable ATM Networks”, Proc. of IEEE INFOCOM’95, Boston, MA, pp. 900–907, 1995.
- [41] D. Medhi: „Some Approaches to Solving a Multi-Hour Broadband Network Capacity Design Problem with Single-Path Routing”, Telecommunication Systems, vol. 13, pp. 269–291, 2000.

Publikációk

Disszertáció

- [D] **B. G. Józsa**: „Traffic Engineering Algorithms for MPLS Networks”, *Budapesti Műszaki és Gazdaságtudományi Egyetem*, 2004.

Folyóiratban megjelent cikkek

- [J1] **B. G. Józsa**, Z. Király, G. Magyar, Á. Szentesi: „An Efficient Algorithm for Global Path Optimization in MPLS Networks”, *Optimization and Engineering*, vol. 2, no. 3, ISSN: 1389–4420, pp. 321–347, September 2001.
- [J2] D. Orincsay, **B. G. Józsa**: „Távközlési hálózatok költségghatékony tervezése”, *Híradás-technika*, vol. 58, no. 4, pp. 39–45, April 2003.
- [J3] **B. G. Józsa**, M. Makai: „On the Solution of Reroute Sequence Planning Problem in MPLS Networks”, *Computer Networks*, vol. 42, no. 2, pp. 199–210, June 2003.

- [J4] L. Tamási, **B. G. Józsa**, D. Orincsay: „Távközlési hálózatok költségghatékony tervezése a forgalomeloszlás periodikus változásainak figyelembevételével”, *Híradástechnika*, vol. 59, no. 9, pp. 32–39, September 2004.
- [J5] L. Tamási, D. Orincsay, **B. G. Józsa**: „Cost-Optimal Design of VoIP Networks Using the VPN Concept”, submitted to *Computer Networks*, 2004.

Konferencia-kiadványban megjelent előadások

- [C1] **B. G. Józsa**, D. Orincsay, Á. Magi, B. Szviatovszki: „On the Use of Trunk Reservation in PNNI Routing”, *Proc. of the 3rd International Conference on ATM (ICATM 2000)*, Heidelberg, Germany, pp. 135–139, June 26–29 2000.
- [C2] **B. G. Józsa**, G. Magyar: „Reroute Sequence Planning for Label Switched Paths in Multiprotocol Label Switching Networks”, *Proc. of the 6th IEEE Symposium on Computers and Communications (ISCC 2001)*, Hammamet, Tunisia, pp. 319–325, July 3–5 2001.
- [C3] **B. G. Józsa**, D. Orincsay: „Shared Backup Path Optimization in Telecommunication Networks”, *The 3th IEEE International Workshop on Design of Reliable Communication Networks (DRCN 2001)*, Budapest, Hungary, pp. 251–257, October 7–10 2001.
- [C4] **B. G. Józsa**: „Reroute Sequence Planning for Protected Traffic Flows in GMPLS Networks”, *Proc. of the 2002 IEEE International Conference on Communications (ICC 2002)*, New York City, NY, USA, pp. 2702–2706, April 28 – May 2 2002.
- [C5] **B. G. Józsa**, D. Orincsay, A. Kern: „On the Use of Routing Optimization for Virtual Private Network Design”, *Proc. of the 7th IFIP Working Conference on Optical Network Design & Modelling (ONDM 2003)*, Budapest, Hungary, pp. 865–880, February 3–5 2003.
- [C6] D. Orincsay, **B. G. Józsa**, D. Kósa, L. Tamási: „Recovery Using Precalculated Paths Based Routing in ATM Networks”, *Proc. of the 10th Computer Networks Conference*, Szczyrk, Poland, pp. 119–128, June 10–13 2003.
- [C7] **B. G. Józsa**, D. Orincsay, A. Kern: „Surviving Multiple Network Failures Using Shared Backup Path Protection”, *Proc. of the 8th IEEE Symposium on Computers and Communications (ISCC 2003)*, Kemer-Antalya, Turkey, pp. 1333–1340, June 30 – July 3 2003.
- [C8] **B. G. Józsa**, D. Orincsay, L. Tamási: „Multi-Hour Design of Dynamically Reconfigurable MPLS Networks”, *Proc. of the 3rd International IFIP-TC6 Networking Conference (Networking 2004)*, Athens, Greece, pp. 502–513, May 9–14 2004.
- [C9] D. Orincsay, **B. G. Józsa**, L. Tamási: „Cost-optimal VPN based VoIP network design”, *Proc. of the 11th International Telecommunications Network Strategy and Planning Symposium (Networks 2004)*, Vienna, Austria, pp. 117–122, June 13–16 2004.
- [C10] D. Orincsay, **B. G. Józsa**, L. Tamási: „On the Use of EKB Routing Strategy in PNNI Networks”, *Proc. of the 2004 IEEE International Conference on Communications (ICC 2004)*, Paris, France, pp. 1003–1007, June 20–24 2004.