

Forgalomhoz alkalmazkodó védelmi módszerek

MITCSENKOV ATTILA, MESKÓ DIÁNA, CINKLER TIBOR

BME Távközlési és Médiainformatikai Tanszék, HSNLab
{mitcsenkov, mesko, cinkler}@tmit.bme.hu

Lektorált

Kulcsszavak: védelem, adaptív, dinamikus, átrendezés, újrakonfigurálás

A korszerű integrált adatátviteli- és távközlő hálózatok növekvő sávszélesség-igényével párhuzamosan mind fontosabbá válik hibátűrésük. A cikkben bemutatott hibavédelmi módszerek megosztott erőforrás-használat és bizonyos mértékű terhelés-ki egyenlítés, valamint a védelmi erőforrások átrendezése révén kívánnak nagyobb hatékonyságot elérni. Több referenciaként szolgáló algoritmus kerül felhasználásra, lehetővé téve a komplex módszereket alkotó összetevők hatásának elkülönült vizsgálatát. Ez elvezethet minket az összetettebb algoritmusok eredményeinek helyes értelmezéséhez.

1. Bevezetés

Az internet megjelenése óta az elérhető szolgáltatások, alkalmazások köre rohamosan bővült, a kezdetekben csak néhány kutatóközpont közti kommunikációra használt hálózat mindennapjaink részévé vált. Megjelent egy sor új szolgáltatás (például Peer-to-Peer, Grid Computing, Video on Demand, Voice over IP, banki szolgáltatások stb.). Napjainkra a hajdan elkülönült távközlő- és számítógép-hálózatok integrálódásának következtében már ugyanazt a digitális hálózatot használhatja a kábeltelevízió, a telefon és a számítógépünk is [1,2].

Az alkalmazások egyre szélesebb köre mellett természetesen a felhasználók száma is ugrásszerű növekedésnek indult [4]. E kettő együttesen pedig egyre nagyobb adatforgalmat eredményez (az internet teljes forgalma a számítások szerint hozzávetőlegesen évente megkétszereződik [5]). Az egyre komolyabb sávszélesség-igény új technológiák kialakulását segítette elő, például az optikai gerinchálózatok térnyerését is ennek köszönhetjük.

E szolgáltatások különböző speciális igényeket támasztanak a hálózattal szemben. Egy beszéd- vagy mozgókép-átviteli szolgáltatásnál nem engedhető meg az a nagyfokú késleltetés-ingadozás vagy csomagvesztés, ami egy böngésző-alkalmazásokat kiszolgáló csomagkapcsolt hálózat esetében még nem zavaró. Egyfelől tehát fontossá vált a szolgáltatás-minőség megfelelő szintű biztosítása (QoS – Quality of Service), másfelől foglalkozni kell a hibátűrési képességekkel is (QoR – Quality of Resilience) [3]. Így a hibákkal szembeni ellenálló-képesség biztosítása a minőség garantálása mellett az útvonalválasztó algoritmusok továbbfejlesztésének egyik fontos iránya lett – a cikkben bemutatott, vizsgált algoritmusok e célt szolgálják.

Nyilvánvaló, hogy az IP-nél megszokott „best effort” jellegű csomagtovábbítás jellegéből adódóan nem alkalmas végpontok közötti garantált minőségű szolgáltatás nyújtására, hiszen a csomagok útja időben folyton változik, nem határozható meg előre, és nincs biztosíték arra, hogy a hálózat erőforrásai egy kapcsolat minden csomagja számára egyformán hozzáférhetőek. Ehhez egyebek mellett a teljes útvonal rögzítésére, ismeretére, erőforrásainak lefoglalására van szükség – és így végpont-végpont útvonalválasztáshoz jutunk.

Minden védelmi megoldás alapvető szükséglete, hogy két pont között találjunk több, egymástól (legalább részben) független utat – különben a hálózat meghibásodott alkotóelemét nem tudjuk elkerülni. Ez tehát szövevényes, többszörös összekötött topológiát feltételez, ami tipikusan a gerinchálózatok sajátossága, hozzáférői hálózatokra kevésbé jellemző [3].

Gerinchálózatokon döntően optikai alapú technológiákat használnak a nagy sávszélesség-igényeknek való megfelelés érdekében. Ezeknek a vizsgált problémakör szempontjából fontos közös jellemzője a (virtuális) áramkörkapcsolt átvitel alkalmazása [2,6,7].

2. Hálózati védelem

A bevezetőben leírtaknak megfelelően szükség van a hálózatok meghibásodás elleni védelmére.

Az egyre megbízhatóbb hálózati eszközöknek köszönhetően lehetséges viszonylag egyszerű, hatékony algoritmusok alkalmazása. Egy akár országos méretű hálózatban egyidejűleg bekövetkező két meghibásodás esélye elenyészően csekély, de még európai méretű hálózatok esetén is indokolatlanná válhat a többszörös hibák ellen is garanciát nyújtó védelem alkalmazása, mert nem térül meg a szükséges erőforrás-többlet. Ezért a cikkben bemutatott algoritmusok fejlesztése során éltünk azzal a szakirodalomban elterjedt feltételezéssel, hogy a hálózatban csupán egyszeres hibák megjelenésével kell számolnunk. Különböző korlátozások bevezetésével e módszerek akár többszörös hibák ellen is felkészíthetőek [8]. A védelem célja: a bekövetkező hiba következményeinek elhárítása, minél gyorsabb helyreállítás, minél rövidebb kiesés, minél kisebb mértékű információ-vesztés [3,6,9].

2.1. Védelem alkalmazásának előnyei és hátrányai

Egy védelmi stratégia kialakításakor több különböző szempontot figyelembe kell vennünk. Nyilvánvaló, hogy amennyiben nem csak a hálózatba érkező igények számára foglalunk utat, hanem védelem céljára is, az többlet-számításokat fog igényelni, ezáltal megnő az egyes igényekre eső összeköttetés-felépítési idő.

A hiba következményeinek gyors felszámolása érdekében a használt protokollnak minél gyorsabban kell reagálnia: a lehető legrövidebb idő alatt kell a hibát felderíteni, és a sérültről az üzemképes, tartalék hálózat-részre átkapcsolni. E folyamat egyik legfontosabb összetevője az útvonalválasztó rutin, de teljesítményét befolyásolják a különböző menedzsment-funkciók (hiba-detektálás, kapcsolás stb.) megvalósításai is. Az átkapcsolás lehetőségének fenntartásához pedig szükség van erőforrások visszatartására: a hálózat egy részét nem használhatja a valós forgalom, azt a hiba esetén működésbe lépő tartalék útvonalak foglalják.

Az ideális védelmi stratégia kiválasztása a számítási idő, a reagálás gyorsasága és a takarékoság közötti kompromisszum eredménye lesz. Ha az erőforrás-igényt szeretnénk csökkenteni, a számítási idő nő meg, ha gyors reakciót és alacsony késleltetést tűzünk ki célul, az erőforrásigény fog megugrani [6,9].

Az itt bemutatott módszerek célja a védelmi erőforrások adaptív átrendezésében rejlő lehetőségek bemutatása. Ezért a számítási igény növekedése terén engedelményeket teszünk az erőforrásokkal való minél hatékonyabb gazdálkodás érdekében. Amikor megtehető, a teljesítmény érdekében erős számítási idő növekedést is kockáztatva, szükség esetén az idővel való takarékoság érdekében pedig a lehetőségeket némiképp korlátozva – éppen a fent említett kompromisszumot keresve.

2.2. Védelmi stratégiák csoportosítása

2.2.1. Védelem vagy helyreállítás?

A bekövetkező hibákra történő reagálás és az ezt megelőző felkészülés alapján beszélhetünk védelemről vagy helyreállításról.

Védelem alkalmazásakor még a meghibásodás előtt, az igény elvezetésekor keresünk és foglalunk számára védelmi utat, és a hiba bekövetkeztekor azonnal átkapcsolunk erre az előre számított védelmi útra.

Helyreállítás használatakor a meghibásodást követően kezdünk a hálózatban más utat keresni az érintett igények számára, nincsenek előre számított védelmi utak. Ezáltal az erőforrásigény csökken, és rövidebb lesz az igények elvezetéséhez szükséges számítási idő, de előfordulhat, hogy nem találunk alkalmas alternatív utakat a hiba bekövetkeztekor.

2.2.2. Hozzárendelt vagy megosztott védelem?

Nyilvánvaló, hogy a védelem többlet erőforrásigényt jelent. Újfent köthetünk egy kompromisszumot a számítási idő és az erőforrásigény között.

Egyszerű, de kevésbé takarékos megoldás a hozzárendelt védelem alkalmazása. Ekkor minden igény va-

lós forgalmat szállító „üzemi útja” mellé keresünk egy attól független, vele azonos kapacitású „védelmi utat”. Ez a hiba bekövetkezéséig tartalékként funkcionál, valós forgalmat nem szállít.

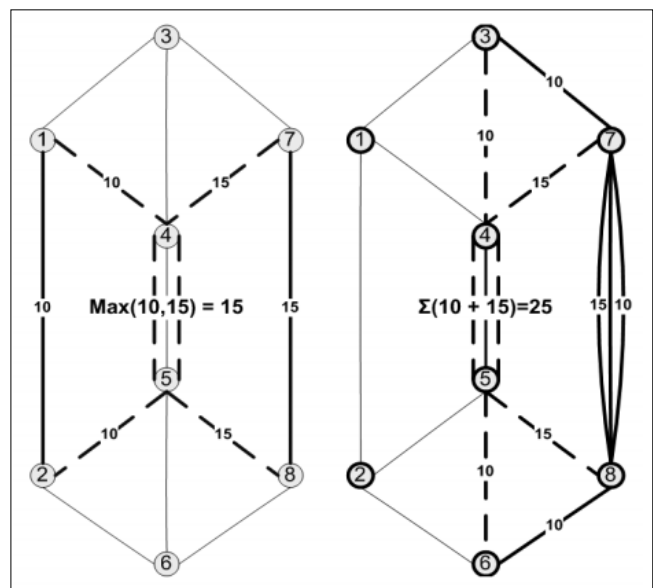
Bonyolultabb, de takarékosabb megoldás a megosztott védelem alkalmazása. Ennek alapja a bevezetőben említett, többszörös hibák kizárására vonatkozó feltételezés: két különböző igény üzemi útja, amely nem használ közös erőforrásokat, nem hibásodhat meg egyszerre, azaz nem válhatnak egyszerre mindkettő a védelmi útjukra. Ezért ezek védelme használhat közös erőforrásokat. Az „osztás” menete és feltételei nyomon követhetőek az 1. ábrán: adott két igényünk, 10 és 15 egységnyi kapacitással. Az üzemi utakat folytonos, a védelmi utakat szaggatott vonal jelzi. Az első esetben (bal oldali ábra) a két üzemi út egymástól független, így a védelmi útjaik közös szakaszán elegendő a sáv szélességük maximumát foglalni védelem céljából. Így bármelyikük üzemi útja hibásodik meg, a 4-5 élen foglalt 15 egységnyi kapacitás elegendő lesz az átkapcsolt forgalom számára – egyszerre mindkettő pedig a fentebb vázolt okok miatt nem sérülhet meg. A második esetben (jobb oldali ábra) viszont a két igény üzemi útjának van közös szakasza: a 7-8 él. Emiatt kettejük védelme nem használhat közös erőforrásokat, hiszen a 7-8 él meghibásodása esetén mindkettő forgalma a 4-5 éle fog átkerülni. Ez esetben tehát a két igény összegének megfelelő 25 egységnyi kapacitást kell védelem céljából a 4-5 élen lefoglalni.

A megosztott védelem alkalmazásának következménye egy takarékosabb algoritmus lesz, amely azonban a megoszthatóság felderítésének problémája miatt nagyobb számítási időt eredményez.

2.2.3. Szakasz, szegmens vagy útvédelem?

A szakaszvédelem a legkézenfekvőbb megoldás: minden szomszédos csomópontpár közti összeköttetéshez rendelhetünk egy alternatív útvonalat, amely a

1. ábra Megosztott védelem



szakasz két végpontját köti össze. A szakasz kiesése esetén teljes forgalmát átirányítjuk a hozzá rendelt védelmi útra.

Egy-egy igény elvezetése után kereshetünk a forrás és nyelő pontpár között egy, az üzemi úttól teljesen független, védelmi útvonalat, ekkor útvédelemről beszélünk. Az egy szakasznál hosszabb, de egy egész útnál rövidebb hálózatrészek védelmét pedig szegmensvédelemnek nevezzük.

2.2.4. Statikus, dinamikus vagy adaptív védelem?

Statikus védelem esetén minden pontpárra előre meghatározott üzemi és védelmi útvonalak közül választunk. Dinamikus a védelem, amennyiben az egyes pontpárok között meghatározott útvonalak időről időre változnak. Adaptív védelemről akkor beszélünk, ha az útvonalválasztó rutin képes a hálózat állapotának, terhelési viszonyainak megfelelően a védelmi utak rendszerének, a hálózat forgalmi viszonyainak megváltoztatására.

2.2.5. Hibafüggő vagy hibafüggetlen?

Az angol szakirodalomban failure-dependent (hibafüggő), illetve failure-independent (hibafüggetlen) néven szereplő fogalmak.

A hibafüggetlenség fogalma jelenti azt, hogy egy algoritmus az adott igény védelmét a hiba helyétől függetlenül határozza meg, tehát az üzemi útjának bármely alkotóeleme sérül is meg, a forgalom mindig ugyanarra a védelmi útra helyeződik át.

Hibafüggő ellenkező esetben: amikor különböző helyen bekövetkezett hibák esetén egyazon üzemi úthoz más és más védelmet rendel.

3. A hálózat modellje

Az egyes konkrét fizikai megvalósítások jellemzőitől független logikai modell megalkotása volt a cél, és erre a hálózatok legelterjedtebb matematikai reprezentációja, a gráf-modell [3] teljes mértékben megfelelő volt.

Ez biztosítja a minél szélesebb körű alkalmazhatóságot: az itt bemutatott algoritmusok bármely technológiával képesek együttműködni, amennyiben az adott technológia és a logikai gráf-reprezentáció közötti átjárást sikerül biztosítani – ez pedig a legtöbb esetben megtehető. Egyszerű irányítatlan gráfot alkalmazunk, ahol a szakaszok legfontosabb jellemzője a teljes kapacitásuk (sávszélességük) és az egységnyi kapacitás költsége. A bevezetőben említettekkel összhangban végpont-végpont útválasztást valósítunk meg.

A cél az egyes algoritmusok teljesítményének vizsgálata volt, így a könnyebb implementálhatóság és áttekinthetőség érdekében központosított útvonalválasztást alkalmazunk. Az elosztott rendszer felé történő továbblépés feltétele az egyes csomópontokban a hálózatról alkotott kép megfelelő részletessége és pontossága lenne – e kérdéssel bővebben foglalkozik pl. [10].

Az igények dinamikusan, a szimuláció szempontjából előre nem ismert minta szerint érkeznek. Ennek megvalósítása természetesen a reprodukálhatóság és összehasonlíthatóság érdekében egy előzetesen összeállított forgalmi állomány.

Az igények dinamikusan, a szimuláció szempontjából előre nem ismert minta szerint érkeznek. Ennek megvalósítása természetesen a reprodukálhatóság és összehasonlíthatóság érdekében egy előzetesen összeállított forgalmi állomány.

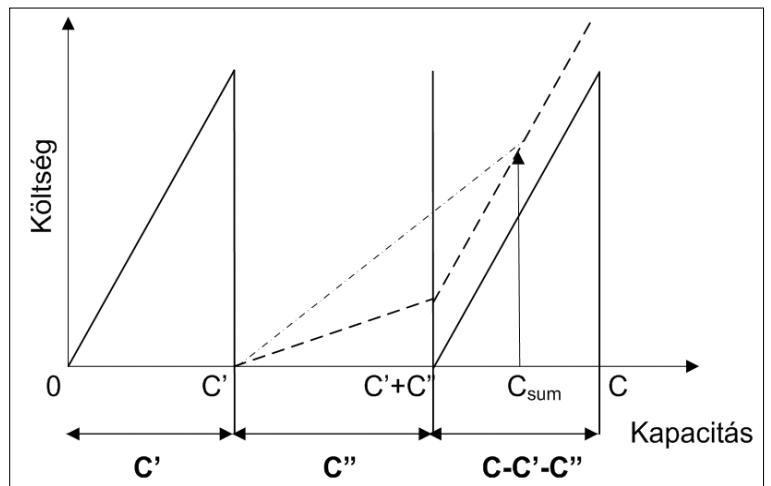
4. Dinamikus védelmi algoritmusok

A következőkben bemutatásra kerülnek a vizsgált algoritmusok, működésük, főbb jellemzőik. Referenciaként és első lépcsőként szolgál egy egyszerű, hibafüggetlen útvédelmi módszer. Ezt több lépésben továbbfejlesztve jutunk el a nagyobb tudású, komplexebb algoritmusokhoz. Az egyes lépcsőket jelentő változatok leírása a könnyebb megértést, eredményeik értelmezése a különbségek hatásának szemléltetését szolgálja.

Különböző szempontok alapján csoportosíthatjuk a bemutatott algoritmusokat. Egyikük hibafüggetlen, a többi hiba-függő (ld. 2.2.5). Vannak köztük út és szakasz-védelmet megvalósító módszerek is (ld. 2.2.3). Egy részük a korábban rögzített útvonalakhoz alkalmazkodó dinamikus, más részük azokat részben változtatni képes adaptív megoldás (ld. 2.2.4).

Az osztott védelem alkalmazásának következményként egy igény védelmi útja számára az egyes éleken elérhető kapacitás két részből tevődik össze. Egyrészt használhatja a tőle független forgalmak ott lefoglalt védelmi erőforrásait, másrészt az adott él még nem használt, szabad kapacitását (lásd 1. ábra). Valós többletfoglalást természetesen csak ez utóbbi jelent, ezért ezt nagyobb súllyal vesszük figyelembe a védelmi út költségének számításakor, míg előbbit csak ennek töredékével, ezáltal próbáljuk a védelmeket a megosztott erőforrások használatára kényszeríteni (2. ábra). E két súlyhányadosának értéke hatással van a védelmi utak kiépítésekor a felhasznált szabad és megosztott kapacitások arányára, valamint közvetve az üzemi, illetve védelmi utak hosszára.

2. ábra
Költségek számítása megosztott védelem esetén



Ha a hányados egyhez közeli, a védelmi utak közel egyenlő mértékben támaszkodnak a megosztható védelmi kapacitásra, és a még nem használt, szabad erőforrásokra. Így megközelítőleg egyforma hosszúságú üzemi és védelmi útvonalakhoz jutunk. Ez ugyan egyszerűbbé teheti a védelmi utak rendszerét, de éppen a megosztott védelem előnyeit veszítjük el vele: nem segítjük elő a közösen használható erőforrások alkalmazását. Ellenkező esetben, ha a megosztott kapacitásból foglalt rész súlya igen kicsi, a védelmi utak építésekor használható, igen olcsó megosztható kapacitás használata, és az ehhez képest igen drága kihasználatlan kapacitás elkerülése érdekében indokolatlanul hosszú elkerülő szakaszok alakulhatnak ki.

4.1. Hibafüggetlen megosztott útvédelem (Failure-independent SPP)

Failure Independent Shared Path Protection (SPP), avagy megosztott útvédelem, a hiba helyétől független védelmi út alkalmazásával. A legegyszerűbb módszer: hibafüggetlen, dinamikus útvédelem.

Működésének lépései:

1. lépés: Az érkező o igény számára Dijkstra algoritmusával üzemi út keresése. Ha sikerül, védelmi utat keresünk (\rightarrow 2. lépés), ha nem, az igényt blokkoljuk és vesszük a következőt (\rightarrow 1. lépés).

2. lépés: Az üzemi út éleit átmenetileg töröljük a hálózatból, ezeket ugyanis a védelmi út nem használhatja majd.

3. lépés: Kiszámítjuk a fent leírt költségfüggvényhez szükséges C' és C'' értékeket (lásd 2. ábra) az egyes élekre. Természetesen a cél az, hogy a védelmi út allokálásához minél kevesebb eddig nem használt sávszélességet foglaljunk, ezt fejezzük ki ennek magasabb költségével is. Ehhez szükség van tehát arra, hogy tudjuk: a hálózat egyes élein mennyi megosztható védelmi kapacitás található. Ez pedig kiszámítható az élen áthaladó azon védelmi utak kapacitásából, melyekhez tartozó üzemi utak függetlenek az új o igény üzemi útjától. Így tehát az üzemi út minden egyes l' élére, valamint a hálózat minden szóba jöhető (az üzemi úttól független) l'' élére kiszámítjuk a $C_{l',l''}$ értéket, melynek jelentése: az adott l' él meghibásodása esetén az l'' élen az igény összesen $C_{l',l''}$ védelem számára felhasználható, megosztható kapacitás található.

4. lépés: Ezen algoritmus hibafüggetlen módszer, ezért egy olyan értékre van szükségünk, amely bármely él meghibásodása esetén érvényes, és kifejezi, mennyi megosztható kapacitás található az adott élen bármely hiba esetén. Ezért végül vesszük a $\min_l \{C_{l',l''}\}$ értéket – ez lesz az l' élre adódó költség kiszámításakor a 2. ábrának megfelelő C'' érték. Ezek után képezhetjük a $C_{sum} - C_{igény}$ értéket, amely megmondja, hogy mennyi szabad kapacitás lefoglalására lesz szükség az adott élen és e kettő megfelelően súlyozott összege adja meg az l' él költségét.

5. lépés: Ismét alkalmazzuk a Dijkstra-algoritmust, és amennyiben ez talál védelmi utat az előzőekben módosított élköltségű hálózatban, az igényt elfogadjuk,

üzemi és védelmi útjait lefoglaljuk a hálózatban – ellenkező esetben blokkolás történik. Végül ugrás a következő igényre (\rightarrow 1. lépés)

4.2. Hibafüggő megosztott útvédelem (Failure-dependent SPP)

Failure Dependent Shared Path Protection (SPP), azaz megosztott útvédelem, a hiba helyétől függő védelmi út alkalmazásával. Hibafüggő, dinamikus útvédelem.

Működése nagyban hasonlít az előző módszerhez, az első három lépés lényegében megegyezik. A különbség a védelmi utak keresésében, pontosabban azok számában mutatkozik meg. Ezúttal az üzemi út minden egyes l' éléhez nyilvántartunk egy-egy külön védelmi utat, amelyre az l' él meghibásodása esetén a forgalom áthelyeződik. Ezáltal feleslegessé válik a negyedik pontban leírt minimum-számítás. Az ötödik pontban leírt Dijkstra-algoritmust lefuttatjuk az üzemi út minden egyes l' élének megfelelően módosított gráfban az ahhoz tartozó $C_{l',l''}$ értékekkel. Ha sikerült minden esetben az üzemi úttól független védelmi utat találni, az igény üzemi és védelmi útjait lefoglaljuk, ellenkező esetben blokkolás következik be.

Jól láthatóan ez annyiban jelent előrelépést az előzőhöz képest, hogy most nem kellett a kapott $C_{l',l''}$ értékek minimumát venni, lehetővé téve a megosztott védelmi erőforrások jobb kihasználását.

4.3. Megosztott védelem részben független utakkal (PDSP)

Partially Disjoint Shared Path Protection (PDSP), azaz megosztott védelem részben független utak alkalmazásával. Hibafüggő, dinamikus szegmensvédelem.

Amint nevéből is látszik, nem az üzemi utaktól teljesen független védelmi utakkal dolgozik. A korábban leírtakkal összhangban feltehetjük, hogy csak egyszeres hibák fordulnak elő a hálózatban. Az üzemi út minden éléhez egy külön védelmi utat rendelünk és így az egyes hibaesetekhez rendelt védelmi utak a hálózat bármely élét használhatják, a hiba esetén érintett él kivételével. Ennek eredményeként a védelmi utak nem feltétlenül lesznek függetlenek a hozzájuk tartozó üzemi utak egészétől. Ezáltal természetesen még nagyobb mozgásteret adunk az algoritmusnak, mint az előzőleg említett módszernél.

A hibafüggő megosztott útvédelemtől tehát mindössze az ideiglenesen törölt élek halmazának meghatározásában különbözik: ezúttal nem a teljes üzemi út kerül törlésre, a második lépésben megelégszünk az éppen vizsgált l' él átmeneti takarásával. Ezt követően a fent leírtak szerint a kiszámított $C_{l',l''}$ értékek alapján Dijkstra algoritmusával védelmi utat keresünk és ha ez minden l' él esetén sikeres volt, az üzemi- és védelmi utakat lefoglaljuk, ellenkező esetben blokkolás következik be.

Ezáltal a lehetséges védelmi út keresésekor a hálózat nagyobb részhalmaza áll rendelkezésre (csak egy élet törölünk egy egész útvonal helyett), növelve a védelmi útvonalválasztás mozgásterét.

5. Adaptív védelmi algoritmusok

Az eddig bemutatott algoritmusok a korábban a hálózatba bekerült igények üzemi és védelmi útjait a későbbiek folyamán már nem képesek módosítani. Ez jelentősen megnehezíti a később kialakuló szűk keresztmetszetek felszámolását, pedig a legtöbb hálózat esetén ezzel számolnunk kell.

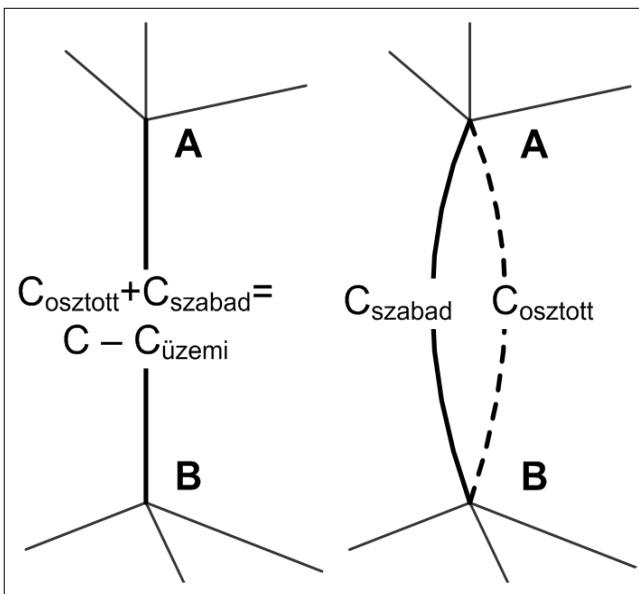
Az adaptív módszerek működésének lényege a védelmi utak átkonfigurálása a hálózat terheléséhez igazodva. Ezek megszakítása ugyanis a felhasználók számára nem észrevehető: hiba esetére fenntartott erőforrások valódi forgalmat nem bonyolítanak. Ezáltal a túltelített hálózatrészek, összeköttetések terhelése csökkenthető, ami kiegyenlítettebb terhelést eredményez és csökkenti a blokkolás esélyét.

5.1. Élkettozés (Link Duplication – LD)

A védelmi utak átrendezése felvet egy komoly problémát: eddig az elérhető megosztott kapacitások ($C_{i,l}$) számításakor erősen támaszkodtunk a már lefoglalt védelmi utakra. Jelen helyzetben viszont több védelmi utat szeretnénk egyidejűleg meghatározni vagy megváltoztatni, érintve a már korábban lefoglalt, de most át-helyezendő védelmeket – ezek előzetes ismeretére tehát nem támaszkodhatunk. A problémát az jelenti, hogy egy adott l él esetén most nem tudjuk meghatározni egy adott o_1 igény számára a felhasználható kapacitás értékét, az ugyanis függ a vele egyidejűleg elvezetésre váró o_2, o_3 stb. igényektől is – mivel azok adott esetben szintén használhatják az l élet.

Erre jelent megoldást a többtermékes folyam probléma (Minimal Cost Multicommodity Flow, MCMCF [11]) alkalmazása. A szakirodalomban található több különböző megközelítés is az MCMCF probléma kezelésére: heurisztikus módszerek, iteráció, vagy az egészértékű lineáris programozás (ILP) használata [12,14,17]. Az e cikkben bemutatott algoritmusok ILP-n alapulnak [13].

3. ábra Az élkettozés lépése



Az ILP alkalmazása azonban felvet egy problémát: csak lineáris költségfüggvényekkel működik. A védelmi utak keresésekor az egyes élek költségfüggvénye viszont nem lineáris, hanem a 2. ábránál leírtaknak megfelelően két, külön-külön lineáris függvény kombinációja. Ezt a problémát az okozza, hogy amíg a védelem talál elérhető megosztott kapacitást, addig azt használja – alacsony költségek mellett –, ezt követően viszont a jóval drágább szabad erőforrásokhoz kell nyúlnia. A szakaszonként lineáris költségfüggvények leírásához szükséges segédváltozók bevezetése hálózatok, gráfok esetén jól szemléltethető az élkettozés modellezési trükkjével.

A szomszédos A és B pontok között futó élen az összetett költség-függvény valójában két kapacitás-tartománnyal és az ezekhez tartozó lineáris költségfüggvényekkel számol. Válasszuk szét e kettőt: ahol eddig egyetlen él futott, oda húzzunk be kettőt, és osszuk meg közöttük a kapacitást – értelemszerűen a két különböző kapacitás-tartomány szerint. A megosztott erőforrásokhoz rendeljük a hozzá tartozó alacsonyabb lineáris költséget, a szabad erőforrásokhoz pedig a magasabbat (3. ábra). Így módon kaptunk két élet az eddigi egy helyett, viszont most már csupa lineáris költség-függvényünk van és alkalmazhatjuk az ILP eszközt [14] a többtermékes folyam probléma megoldására.

5.2. A probléma ILP megfogalmazása

Védelmi út keresésekor tehát nem csak az új igény védelmi útjának próbálunk helyet keresni, hanem a már lefoglalt igények egy csoportjának védelmét is változtatni fogjuk. Pontosabban egy l él kiesésének szimulálásakor az összes olyan igény vonatkozó védelmi útját töröljük, amely üzemi élként használta az l élet. Ezeknek és az új igénynek egyidejűleg próbálunk védelmet találni a kiesett él átmeneti törlése után maradó hálózatban. A következőkben leírt ILP probléma megoldása adja meg az új védelmi utak rendszerét.

Felmerülhet a kérdés, hogy miért csak az l él kiesése által érintett védelmi utakat rendezzük át és miért nem az összeset? Ennek magyarázata az adódó matematikai probléma komplexitása [15,16]. A hálózatban (gráfban) összesen jelen lehet az igények száma $|V^2|$ -tel arányos, míg az adott élen futó védelmi utak száma $|V^2|/|E|$ -vel. A lineáris programozási feladat változóinak száma tehát így kezelhető szintre csökken.

A védelmi utak előző pontban leírt halmazának együttes elvezetését adó többtermékes folyam probléma ILP megfogalmazása a következő:

Célfüggvény:
$$\min \sum_{o \in T_e} \left\{ \sum_{l \in E^{osztott}} x_l^o \omega_l + \sum_{l \in E^{szabad}} x_l^o \gamma_l \right\} \quad (1)$$

E^{szabad} az élkettozéskor a szabad kapacitásból keletkezett éleket, $E^{osztott}$ a megosztott kapacitásból keletkezett éleket jelöli, e az aktuális vizsgált él (melynek kiesését szimuláljuk), T_e az elvezetendő igények (védelmek) halmaza, az x_l^o változó értéke jelzi az o igény védelmi útjának l élre eső részét, ω_l a szabad, míg γ_l a megosztott erőforrások költségét.

A feltételek:

$$\sum_{o \in T_e} x_i^o \leq C_i^{szabadj} \quad \text{minden } \forall l \in E^{szabadj} \text{ esetén} \quad (2)$$

$$\sum_{o \in T_e} x_i^o \leq C_i^{osztott} \quad \text{minden } l \in E^{osztott} \text{ esetén} \quad (3)$$

$$\sum_{\forall j \in T_e, j \neq i} x_j^o - \sum_{\forall k \in T_e, k \neq i} x_k^o =$$

$$= \begin{cases} 0 & \text{ha } i \neq s^o \wedge i \neq d^o & \text{minden} \\ b^o & \text{ha } i = s^o & o \in T_e, i \in V \\ -b^o & \text{ha } i = d^o & \text{esetén} \end{cases} \quad (4)$$

$$0 \leq x_i^o \leq b^o \quad \text{minden } l \in E \text{ és } o \in T_e \text{ esetén} \quad (5)$$

$$\sum x_{ik}^o = b^o \cdot z_{ik}^o \quad \text{minden szomszédos} \quad (6)$$

$$i, k \in V \text{ és } o \in T_e \text{ esetén}$$

$$z_{ik}^o \in \{0, 1\} \quad (7)$$

(2) és (3) a kapacitáskorlátokat jelentő feltételek, (4) felelős a folyammegmaradásért, (5) az x változók értékére ad nyilvánvaló korlátot, (6) bevezet egy, az „osztatlanságot” biztosító bináris segédváltozót, (7) pedig a z változók bináris voltát rögzíti.

Az „osztatlanság” fogalma alatt az élkettőzés alkalmazásához szükséges feltételt értjük: ha egy védelmi út i pontba beérkező forgalma nem is egyetlen élen távozik onnan, csak az egyazon él kettőzésével keletkezett két élen osztható. A (6) feltétel másként megfogalmazva annyit jelent, hogy egy ilyen élpáron vagy átmegy az adott o igény teljes forgalma, vagy teljesen elkerüli azt.

5.3. Hibafüggő megosztott útvédelem élkettőzéssel (SPP-LD)

Failure Independent Shared Path Protection with Link Duplication (SPP-LD), avagy adaptív megosztott útvédelem. Adaptív módszer, mely képes a már lefoglalt védelmeket a megváltozott terhelésnek megfelelően átrendezni.

Működésének lépései:

1. lépés: Az érkező o igényre Dijkstra algoritmusával üzemi út keresése. Ha sikerül, védelmi utat keresünk (\rightarrow 2. lépés), ha nem, az igényt blokkoljuk, vesszük a következőt (\rightarrow 1. lépés)

2. lépés: Sorra vesszük az üzemi út l' éleit. A következő l' élre összegyűjtjük a rajta áthaladó üzemi úttal rendelkező igények ezen élhez tartozó védelmi útjait és ezeket kiegészítve az új o igény leendő védelmével megkapjuk az ILP megfogalmazásban látott T_e halmazt. Amennyiben nincs több l' él, az igény és annak védelmi útjai elvezethetőek a hálózatban, azokat lefoglaljuk, és vesszük a következőt (\rightarrow 1. lépés).

3. lépés: A T_e halmazban található védelmi utakat töröljük a hálózatból – ezek helyett a következő lépésben újakat keresünk.

4. lépés: Megoldjuk az ILP problémát, egy, az algoritmusra jellemző kiegészítéssel: minthogy útvédelemről beszélünk, garantálni kell, hogy a kialakuló védelmi utak függetlenek legyenek a hozzájuk tartozó teljes üzemi úttól. Ez megtehető, amennyiben az ILP-t kiegészítjük a következő korlátokkal (WP_o jelöli az o igény üzemi útját):

$$x_j^o = 0 \quad \text{bármely } o \in T_e \text{ és } l \in WP_o \text{ esetén}$$

Amennyiben az ILP-nek létezik megoldása, egyidőben elvezethetőek a hálózatban a korábbi igények, valamint az új o igény l' -höz tartozó védelmi útjai. Vesszük az üzemi út következő élét (\rightarrow 2. lépés). Ha az ILP nem megoldható, az újonnan jött igény számára védelem nem biztosítható, ezért az új igényt blokkoljuk, a korábban törölt védelmi utakat visszaállítjuk. Ugrás a következő igényre (\rightarrow 1. lépés).

Az algoritmus adaptív vá tétele lehetőséget ad bizonyos fokú terhelés-kiegyenlítésre, leginkább erősen összekötött hálózatok esetén. A védelmek keresésekor az átkonfigurálás lehetőségének köszönhetően nagyobb mozgástérrel rendelkezik mint a dinamikus verzió, ezért az e fázisban fellépő blokkolások arányának csökkenése várható. Ugyanakkor a nagyobb számításigény miatt megnövekedett futási idővel kell számolnunk, ami végző soron a beérkező igények kiszolgálásának késleltetését növeli.

5.4. Megosztott védelem részben független utakkal, élkettőzéssel (PDSP-LD)

Partially Disjoint Shared Path Protection with Link Duplication (PDSP-LD). Hibafüggő, adaptív szegmensvédelem. Mind dinamikus megfelelője, a PDSP, mind az előzőleg bemutatott SPP-LD továbbfejlesztésének is tekinthető, amint azt az 1. táblázat is mutatja, előbbihez képest az adaptív működés, a védelmek áttrendezése, utóbbihoz képest a szegmensvédelem révén.

Működése az előző algoritmus leírása után könnyen érthető, egyetlen változtatásra van szükség: ezúttal az egyes védelmi utaknak nem a hozzájuk tartozó üzemi út egészétől, hanem csak a vizsgált l éltől kell függetlennek lenniük. Így tehát a negyedik lépésben leírt feltétel e módszer esetén helyesen:

$$x_i^o = 0 \quad \text{bármely } o \in T_e \text{ esetén}$$

Ettől eltekintve a két algoritmus működése azonos.

1. táblázat Az algoritmusok csoportosítása

	Útvédelem	Szegmensvédelem
Dinamikus	Hibafüggő megosztott útvédelem (SPP)	Megosztott védelem részben független utakkal (PDSP)
Adaptív	Hibafüggő megosztott útvédelem élkettőzéssel (SPP-LD)	Megosztott védelem részben független utakkal, élkettőzéssel (PDSP-LD)

A fent említett mindkét módszernél nagyobb mozgástérrel rendelkezik, emiatt azt várhatjuk, hogy hatékonyságban mindkettőt felülmúlja. A dinamikus PDSP változattal összehasonlítva az adaptív működésnek köszönhetően kiegyenlítettebb terhelést és az ezzel járó előnyöket várhatjuk, az SPP-LD-hez képest pedig a védelmi utakra megfogalmazott enyhébb feltételeknek köszönhetően nagyobb hatékonyságot a védelmi utak kiépítésénél.

A legbonyolultabb, legösszetettebb működésű módszer a bemutatott algoritmusok közül, melyhez a kiindulásként alkalmazott hibafüggetlen, dinamikus útvédelemről három lépésben jutottunk el: először áttértünk a hibafüggő működésre, ezt követően lecseréltük az útvédelmet szegmensvédelemre, majd adaptívvá tettük a korábban már lefoglalt védelmek átrendezési képességével.

5.5. Részben adaptív megosztott útvédelem

Az adaptív módszerek alkalmazása során újabb problémákba ütközünk. Az egészértékű lineáris programozási feladat (ILP) számítási igénye, komplexitása meglehetősen magas. Egyes esetekben ez azt eredményezheti, hogy az egy igényre eső számítási idő megengedhetetlenül nagyra nő (a hálózat méretétől, összekötöttségétől, egyéb tulajdonságaitól függően pár másodperctől akár a több perces tartományig). A nyilvánvaló előnyei mellett így e módszerek csak korlátozottan alkalmazhatóak.

Érdemes tehát elgondolkodni a gyors és egyszerű dinamikus, valamint a komplex, de jobb teljesítményű adaptív módszerek ötvözésén. Ez természetesen kompromisszumos megoldást jelent, hiszen így ugyan alacsonyabb futási időt nyerünk, viszont veszítünk az adaptív megoldások jelentette előnyökből.

Ennek megvalósítására több lehetőségünk is van, talán a legkézenfekvőbb a következő: a folyamatosan érkező igények kiszolgálásakor elsőként alkalmazzuk a választott módszer (SPP vagy PDSP) dinamikus változatának útvonalválasztó rutinját. Amennyiben ez sikerrel járt, és mind az üzemi út, mind az egyes hibákhoz rendelt védelmi utak lefoglalásra kerültek, akkor folytatjuk a következő igényvel.

A védelmi utak keresésekor fellépő blokkolás során azonban van még egy további lehetőségünk: a hálózatban már lefoglalt védelmi utak átrendezése érdekében a módszer adaptív változatához továbbíthatjuk az igényt. Így tehát csak abban az esetben indítjuk el a jóval bonyolultabb és nagyobb számítási igényű újrendező mechanizmust, amikor arra ténylegesen szükség van.

A hálózatban egyidejűleg mozgatott védelmi utakra való tekintettel adminisztratív problémákat vetne fel egy hálózaton belül az út- illetve szegmens-védelem kevert alkalmazása. Ennek megfelelően a részben adaptív megosztott útvédelem az útvédelmet megvalósító dinamikus és adaptív algoritmusok kombinációját jelenti, melyre – rövidítésük alapján – SPP+SPP-LD-ként fogok hivatkozni.

5.6. Részben adaptív megosztott védelem, részben független utakkal

Az előző pontban leírt gondolat alapján a szegmensvédelmet megvalósító dinamikus és adaptív algoritmusok kombinációja, rövidítésük alapján PDSP+PDSP-LD-ként fogok rá hivatkozni.

6. Szimulációk

6.1. A szimulátor bemenete

6.1.1. Paraméterek

A szimulátor paramétereként értelemszerűen meg kell adnunk a használni kívánt hálózatot és az előre generált forgalmat (amely a szimuláció szempontjából nem lesz előzetesen ismert).

Céljaink között szerepelt, hogy különböző terhelések mellett is vizsgálni tudjuk az egyes módszereket – ehhez kézenfekvő megoldásnak tűnik az egyre erőteljesebb forgalmak alkalmazása, az igények nagyságának növelése. Az egyszerűség és a megegyező forgalomminták lehetőségének megőrzése érdekében azonban egy ezzel egyenértékű, más megoldást választottunk: a hálózat áteresztőképességét változtatjuk. Így paraméterként megadható az egyes élek kapacitása is.

6.1.2. Hálózatok és forgalmak

A szimulációk során összesen hat különböző méretű és összekötöttségű hálózatot alkalmaztunk. A cél ismert topológiák választása volt, ezek egyszerű gráfként kerültek alkalmazásra, egyenrangú (költségű és sávszélességű) élekkel, a hullámhossz-csatornák és egyéb, technológiára jellemző paraméterek elhagyásával. Így a bevezetőben leírtak szellemében egy igen általános modellhez juthatunk, lehetővé téve az algoritmusok teljesítményének technológia-független vizsgálatát, ugyanakkor nem elveszítve a konkrét megvalósításokhoz való illeszthetőség lehetőségét, tekintettel arra, hogy a legtöbb összetett hálózati modell gráfként leírható.

A felhasznált hálózatok csomópontszáma 13 és 28 között, élszáma 19 és 61 között változik. Különösen érdekes lehet az egyazon 28 csomóponton alapuló topológia különböző összekötöttségű 35, 41 és 61 élet tartalmazó változata (COST266 referencia-hálózatok). Ezek vizsgálata tanulságokkal szolgálhat a hálózat sűrűségének és az egyes algoritmusok teljesítményének összefüggéseire.

2. táblázat Adaptív módszerek számítási ideje

Hálózat	Csomópontok száma	Élek száma	Átlagos foksám	PDSP-LD futási idő
NSFNET	13	19	2,92	0,74
COST266_CT	16	23	2,88	2,68
o22	22	45	4,09	2,92
COST266_RT	28	35	2,50	6,68
COST266_BT	28	41	2,93	6,96
COST266_TT	28	61	4,36	7,73

Az egyes hálózatokhoz használt forgalmak paramétereit az adott hálózat megfelelő jellemzői határozzák meg. Ez lehetővé teszi a hálózati jellemzők hatásának vizsgálatát. Azonos valószínűséggel keletkeztek igények bármely csomópont-pár között, a szimuláció teljes időtartama alatt a csomópont-párok számával arányos számban. A hálózatban egyidejűleg aktív igények száma átlagosan a csomópont-párok számának felével egyezett meg, az egyes igények nagysága pedig 1 és 20 egység között egyenletes eloszlású volt. Ennek eredményeként a forgalom intenzitása (az azonos átlagos sávszélességnek és az egyidejűleg aktív igények számának köszönhetően) szintén a csomópont-párok számával, tehát a csomópontok számának négyzetével arányos. A forgalmak és hálózatok jellemzőit a 2. táblázat tartalmazza.

7. Numerikus eredmények

Bemutatásra került hét algoritmus, az egyes fejlődési lépések bemutatása és hatásuk vizsgálata céljából. Érdemes tehát összehasonlítani a hibafüggetlen és a hibafüggő megosztott útvédelmet, az útvédelmet és szakaszvédelmet (SPP vs. PDSP, SPP-LD vs. PDSP-LD), valamint egyazon módszerek dinamikus és adaptív változatait (SPP vs. SPP-LD, PDSP vs. PDSP-LD).

Minden hálózat esetén 10 különböző szimulációra támaszkodunk, elég hosszú forgalmi mintákkal ahhoz, hogy a kezdeti tranzienst hatása elhanyagolható legyen. A korábban leírtaknak megfelelően a hálózathoz képest változó mértékű terhelést a hálózat éleihez tartozó kapacitás-értékek biztosítják, 10 lépésben egy olyan tartományt felölve mely a meglehetősen magas blokkolási arányoktól eljut a 0% közeli arányokhoz. Az eredmények bemutatásánál egyes esetekben mind a 10 értékhez tartozó adatokat láthatjuk, ahol pedig teljes szimuláció-sorozatra átlagolt eredmények kerültek bemutatásra, ott a 10 érték átlagát találjuk.

Természetesen minden útvonalválasztó algoritmus fontos jellemzője a hatékonysága, az elérhető átbocsátóképesség. Ennek mérése, megfigyelése során azonban szembe kell néznünk a lineáris programozási feladat bonyolultságából eredő nehézségekkel: az ILP-feladat megoldása sok esetben egyszerűen, viszonylag gyorsan meghatározható.

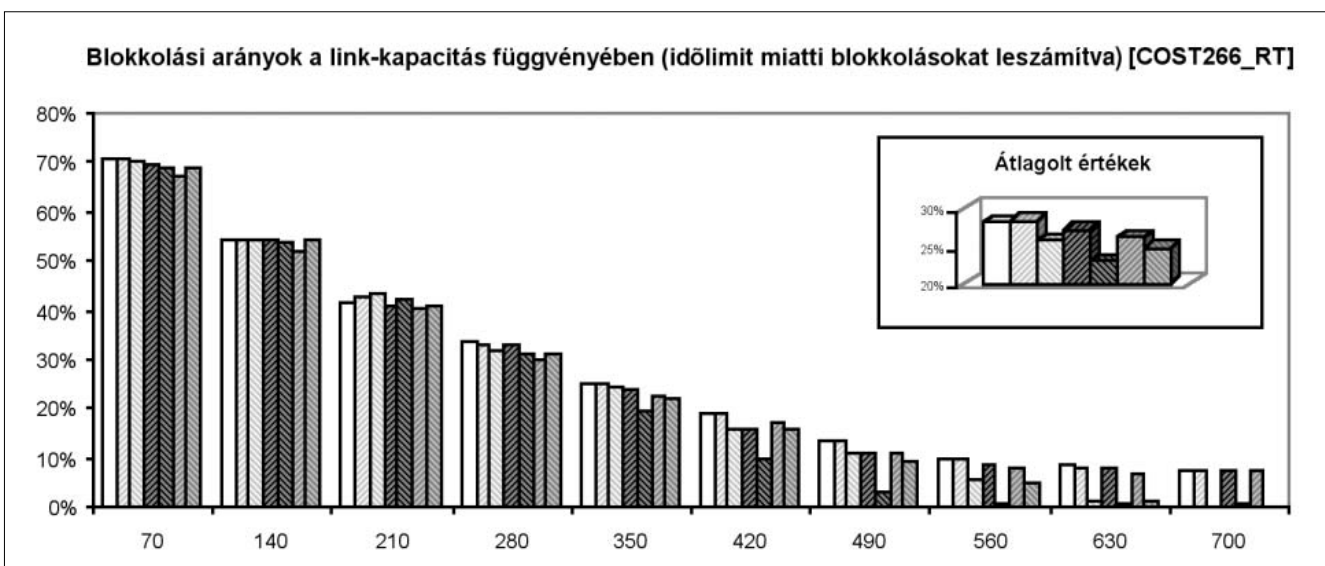
Néhány eset azonban, a hálózat speciális állapotának köszönhetően kiugróan hosszú számítási időt igényel. Annak érdekében, hogy ilyen esetben a folytonosan érkező igények kiszolgálása ne akadhasson el egy-egy igény miatt, a lineáris programozási feladat megoldásának kereséséhez szükséges egy időkorlátot rögzíteni.

Így persze egyes igények valójában azért kerülnek blokkolásra, mert nem lesz idő az érkezésük után megváltozó védelmi út rendszer kiszámítására, noha az ILP-feladat megoldható volna, tehát az igény számára elegendő szabad kapacitás rendelkezésre áll a hálózatban. Emiatt egyaránt hamis képet kapunk az algoritmus képességeiről, amennyiben ezen igényeket akár a blokkolt, akár a kielégített igények közé soroljuk, ezért a blokkolási arányok vizsgálatakor az adaptív algoritmusok esetében ezen forgalmi igényeket egyszerűen kihagytuk az eredmények öszszesítésekor – így nyílik leginkább lehetőségünk ezen módszerek elvi képességeinek vizsgálatára.

Valódi alkalmazások esetén természetesen a számítási idő korlát nem megkerülhető. A részben adaptív módszerek kifejlesztését éppen ez motiválta: amennyiben nincs lehetőség a védelem-átrendező rutin rendszeres futtatására, akkor az útválasztás gyorsítható a dinamikus és adaptív változatok kevert alkalmazásával, így az egy igényre eső számítási idő csökkenthető.

Az egyes módszerek teljesítménye közötti eltéréseket nagyobb hálózatok esetében szembetűnőbbek. Ha az egyik legnagyobb felhasznált hálózaton (COST266_BT) futtatott szimuláció eredményeit (4. ábra) vizsgáljuk, láthatóvá válnak a következők:

4. ábra Blokkolási arányok



- A hibafüggő útvédelem valóban alacsonyabb blokkolási szintet eredményez a hibafüggetlen változatnál.
(Hibafüggetlen SPP vs. Hibafüggő SPP)
- Az útvédelem helyett a szegmens-védelem alkalmazása valóban észrevehetően nagyobb mozgásteret adott az algoritmusnak, és ennek hatása a blokkolási arányokban is megjelent.
(SPP vs. PDSP, SPP-LD vs. PDSP-LD)
- Az adaptív módszerek egyértelműen felülmúlják a dinamikus változatokat.
(SPP vs. SLL-LD, PDSP vs. PDSP-LD)
- Amint várható volt, az adaptív védelmi út-kereső rutin részleges alkalmazása a dinamikus változatoknál jobb, az adaptív verzióknál rosszabb eredményt hozott (a blokkolási arányt és a későbbiekben bemutatott egyéb jellemzőket tekintve), rövidebb futási idővel.
(SPP+SPP-LD illetve PDSP+PDSP-LD)

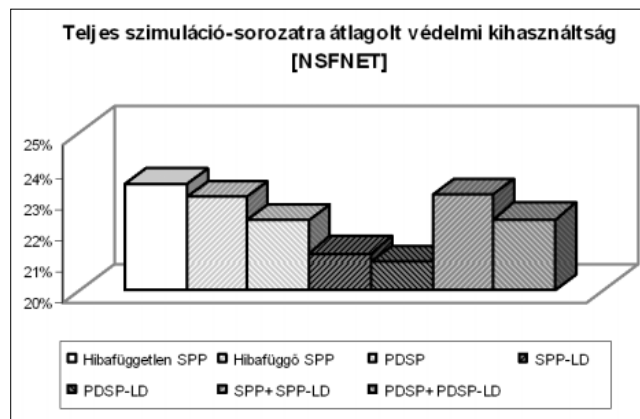
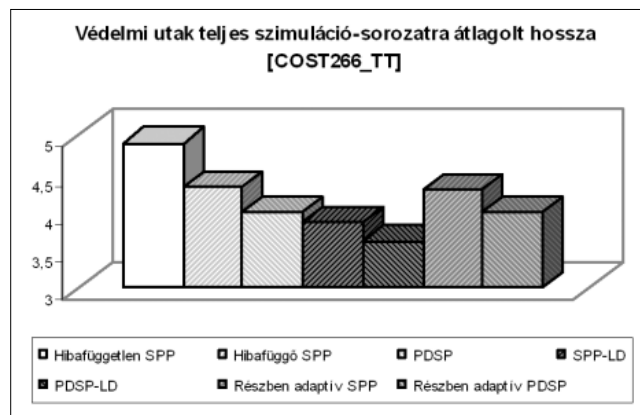
A COST266_RT hálózat felépítése (sok csúcs, kevés él) miatt kiválóan alkalmas egyben az útvédelem gyengeségeinek szemléltetésére. Ha ugyanis megnézzük, hogy a 4. ábra alsó diagramján az alacsonyabb blokkolást eredményező kapacitás-tartományokban melyik algoritmus hogyan teljesít, látni fogjuk, hogy a különböző adaptív és dinamikus SPP-verziók lényegesen gyengébben teljesítenek a leginkább szegmens-védelemként aposztrofálható PDSP változatoknál. Ennek oka a hálózat viszonylag alacsony fokú összekötöttsége – bizonyos esetekben gondot okozhat két pont között az üzemi és védelmi út számára két, egymástól élfüggetlen út kiépítése.

Ugyanakkor kiválóan megfigyelhető a dinamikus és adaptív módszerek kevert alkalmazásakor kötött kompromisszum eredménye: a részben adaptív változatok teljesítménye láthatóan felülmúlja a dinamikus változatokét, de alulmarad az adaptív módszerekkel szemben.

Az adaptív algoritmusok használatának valódi előnyei azonban nem is igazán a blokkolási arányokat nézve válnak nyilvánvalóvá. Érdeemes megvizsgálni a hálózat állapotát leíró más jellemzőket is.

Vizsgáljuk meg az előálló üzemi és védelmi utak hosszát. Amint az algoritmusok leírásánál láttuk, az üzemi utak keresésekor használt eljárás megegyezik az összes módszernél. Így tehát e téren nem is kell szignifikáns különbségekkel számolnunk. Ugyanakkor a védelmi utak hosszának tekintetében már észrevehető eltérésekkel találkozunk. Az 5. ábrán, amely a COST266_TT hálózatban mért értékeket jeleníti meg, ez jól látszik: a védelmi utak hosszának vonatkozásában jelentős különbségek tapasztalhatóak az adaptív módszerek javára. A tendencia ismételtén igazolja a fejlesztési lépéseket, mind a hibafüggő működés, mind a szegmens-védelem, mind az adaptív megközelítés előrelépést hozott.

A rövidebb védelmi utak kialakulásának magyarázata a védelem átrendezésének lehetőségében keresendő: egyidejűleg több védelmi út számára optimális konfigurációt építünk ki. Ezzel elkerülhetők azok az ese-



5. és 6. ábra
Védelmi utak hossza és védelmi kihasználtság

tek, amikor egy-egy korábbi védelmi út által foglalt erőforrások miatt az újabb igények hosszú kerültre kényszerülnek. Emellett az adaptív módszerek védelmi célokra alacsonyabb erőforrás-igénnyel lépnek fel a hálózattal szemben, mint a 6. ábrán megfigyelhető (NSFNet hálózat, védelmi célra foglalt / összes erőforrás aránya). Így tehát elmondható, hogy ez esetben a rövidebb védelmi utak nem a kisebb mértékű tartalék-megosztás, hanem az optimálishoz közelítő konfiguráció következményei.

Természetesen a több szempontból is jobb teljesítmény árát meg kell fizetni – ez esetben elsősorban a számítási idő területén. Ez korlátozhatja az adaptív algoritmusok használhatóságát és ez adja a részben adaptív algoritmusok létjogosultságát is. A bevezetőben leírtaknak megfelelően e módszerek jelenleg központosított működésűek, és ez jelentős mértékben befolyásolhatja skálázhatóságukat.

Választ kell tehát találnunk arra a kérdésre, mekkora hálózatok, milyen topológia mellett, milyen korlátozásokkal használhatóak az itt bemutatott algoritmusok. A szimulációk során csupán a hálózatok topológiai jellemzőit hasznosítottuk, és nem támasztottunk egyes hálózat-típusokra jellemző elvárásokat. Nagy, akár európai méretű hullámhossz-kapcsolt optikai gerinchálózatok esetén nem feltétlenül követelmény a másodperces, vagy annak csak törtrészét kitevő számítási idő, ami viszont kisebb méretű, más felhasználású hálózatok esetén elvárás lehet.

Kisebb hálózatok esetén még elfogadhatóan rövid számítási időket kapunk (2. táblázat), a nagyobb (28 csomópont/35-41-61 él) hálózatok esetén mért értékek bizonyos alkalmazások esetén már erősen korlátozzák a tisztán adaptív módszerek használhatóságát. Megfigyelhető, hogy döntően a hálózat csomópontjainak száma befolyásolja a számításigényt: a 13 csomópontból álló NSFNet hálózat esetén mért érték tízszeresével találkozunk az alig több, mint kétszer annyi pontból álló, hasonló sűrűségű COST266_BT hálózatnál. Ugyanakkor a három egyforma csomópontszámú, jelentősen különböző élszámú hálózathoz tartozó eredmények közötti eltérés lényegesen csekélyebb.

Nem kell azonban végleg lemondani az adaptív működés nyújtotta előnyökről nagyobb hálózatok vagy sebesség-kritikus alkalmazások esetén sem: ez a részben adaptív megoldások területe lehet. Ezek egy igényre eső számítási ideje még a legnagyobb hálózatok esetén sem lépi túl a 0,1 másodperces határt (8. ábra), kisebb gráfok esetén pedig 0,01 másodperc nagyságrendű (7. ábra). Emellett, mint korábban láttuk, bizonyos mértékig képesek ötvözni a bonyolultabb működés előnyeit, teljesítményét a dinamikus módszerekre jellemző alacsony számításigénnyel.

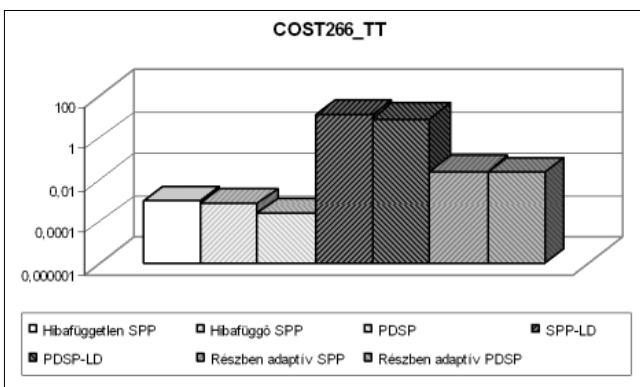
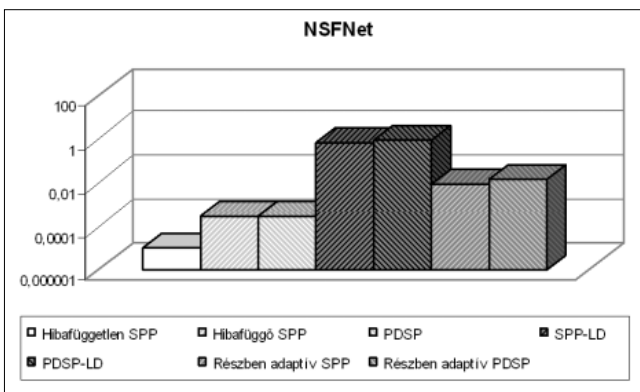
8. Összegzés

8.1. Következtetések

Bemutatásra került hét különböző, megosztott védelmet megvalósító algoritmus. Ezek egy hosszabb fej-

7. és 8. ábra

Egy igényre eső számítási idő a legkisebb (NSFNet) és a legnagyobb hálózatban (COST266_TT)



lesztési út egyes állomásai, illetőleg az egyszerűbb és bonyolultabb módszerek kevert használatán alapuló átmeneti megoldások voltak.

Elsőként a dinamikus módszerek talaján maradván láthattunk hibafüggetlen és hibafüggő, valamint út illetve szegmens-védelmet megvalósító módszereket. A szimulációk eredményeit vizsgálva igazolást nyert mindkét változtatás pozitív hatása: kisebb számításigény mellett alacsonyabb blokkolási szintet eredményeztek és a kiépített védelmi utak hosszát, valamint a védelmi célú erőforrás-használatot illetően is jobban teljesítettek.

Igazán nagy változást pedig az adaptív módszerek jelentenek. Teljesítményüket elemezve megállapítható, hogy némiképp alacsonyabb blokkolási arányok mellett jelentős előrelépést hoztak több fontos területen. Megfigyelhettük, hogy a kiépített védelmi utak hossza, és ezzel párhuzamosan a védelem céljára foglalt erőforrások aránya tekintetében is sikerült előrelépni a dinamikus módszerekhez képest. Az adaptív algoritmusok igazi előnye pedig azok rugalmasságában rejtezik: a nagymértékű átalakítási, átrendezési képesség következtében akár hibák, akár szűk keresztmetszetek megjelenése esetén megvan a lehetőségünk a terhelés ezt elkerülő elosztására, ezáltal csökkentve a kritikus terhelésű élek számát.

Emellett szembe kell néznünk a jelentős többletszámításokat igénylő adaptív működés negatív következményével, a megnövekedett futási idővel is. Láttuk, hogy a hálózat mérete erősen befolyásolja az egy igényre eső számítási időt, ezért nagyobb hálózatok vagy sebesség-kritikus alkalmazások esetén csak korlátozottan alkalmazhatóak ezek a módszerek. E problémára jelenthet megoldást a dinamikus és adaptív algoritmusok kevert használata, a kettő előnyeinek, az adaptív módszerek jobb teljesítményének és a dinamikus változatok rövidebb futási idejének ötvözésére. A megkövetelt kompromisszumot az előálló összetett algoritmusmal szembeni igényeink kell, hogy meghatározzák, a gyorsaság érdekében kevesebb, vagy a jobb teljesítmény érdekében több újrendezési lépést engedélyezve.

8.2. Kitekintés

A bemutatott módszerek számtalan továbblépési lehetőségét rejtenek magukban.

Az egyik legizgalmasabb kérdés az adaptív módszerek átrendezési lépésének időzítése és az ebbe bevont védelmek csoportjának kiválasztása. Jelenleg minden igény üzemi és védelmi útjának elvezetése között, utóbbival egyidejűleg az egy adott élet használó védelmeket rendezzük át. Megtehetjük azonban, hogy már az üzemi út elvezetése előtt, vagy azzal párhuzamosan teszünk így – ily módon feltehetőleg tovább csökkenthető az üzemi blokkolások száma.

Megpróbálkozhatunk valamilyen küszöbérték átlépése által kiváltott újirányítási, terhelés-kiegyenlítési rutinnal is, például: szűk keresztmetszetek kialakulása, egymást követő többszörös blokkolások stb. Ekkor nem kell minden igényt követően lefuttatni az időigényes

adaptációs lépést, így csökkenthető a futási idő. Különösen az utóbbi esetben igényel alapos átgondolást az átrendezni kívánt védelmek csoportjának kiválasztása. Csábító lehetőség például a szűk keresztmetszetű váltóéleket használó útvonalak felszabadítása, áthelyezése.

A mind alaposabb terhelés-kiegyenlítés további eszköze lehet aprólékosan kidolgozott (de az ILP kötöttsége miatt lineáris) költségfüggvények alkalmazása, például a szabadabb élek használatának elősegítése a terheltebbek helyett.

Köszönetnyilvánítás

Ez a munka az Európai 6. Kutatási Keretprogram IST IP NOBEL projektjének keretében készült.

Irodalom

- [1] Andrew S. Tanenbaum:
Számítógép-hálózatok, Panem, 2004.
- [2] T. Cinkler:
On Multi-Layer Multi-Domain Networks,
Winterschool Tutorial at NoE e-Pohoton/ONE
Winter School, Aveiro, Portugal, 2005.
- [3] Wayne D. Grover:
Mesh-Based Survivable Networks,
Prentice Hall, 2004.
pp.149–172, 173–268, 377–467.
- [4] Internet Software Consortium (ISC),
<http://www.isc.org>
- [5] London Internet Exchange (LINX),
<http://www.linx.net>
- [6] J.-P. Vasseur, M. Pickavet, Piet Demeester:
Network Recovery, Elsevier, 2004.
pp.39–131, 203–297, 297–423.
- [7] Harry G. Perros:
Connection-Oriented Networks,
John Wiley and Sons, 2005.
pp.19–46, 47–130, 131–180, 220–229.
- [8] Zs. Pándi, M. Tacca, A. Fumagalli:
A Threshold Based On-line RWA Algorithm with
Reliability Guarantees ONDM 2005 Conference,
February 2005, Milan, Italy
- [9] Ramesh Bhandari:
Survivable Networks – Algorithms for diverse Routing,
Kluwer, 1999.
- [10] Szigeti J., Ballók I., Cinkler T.:
Hatékony információfrissítési stratégiák
automatikusan kapcsolt optikai hálózatokban,
Híradástechnika, 2005/2; pp.45–50.
- [11] Ravindra K. Ahuja, Thomas L. Magnanti,
James B. Orlin:
Network Flows, Prentice Hall, 1993.
pp.108–113, 649–695.
- [12] Mokhtar S. Bazaraa, John J. Jarvis, Hanif D. Sherali:
Linear Programming and Network Flows,
John Wiley and Sons, 1977. pp.587–601.
- [13] Schrijver:
Theory of Linear and Integer Programming,
Wiley, 1998.
- [14] M. Pióro, D. Medhi:
Routing, Flow and Capacity Design in
Communication and Computer Networks,
Elsevier, 2004.
pp.105–148, 151–169, 353–400.
- [15] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann,
A. Marchetti-Spaccamela, M. Protasi:
Complexity and Approximation,
Springer-Verlag, 1999. pp.65–69.
- [16] B. M. E. Moret, H. D. Shapiro:
Algorithms from P to NP, Volume I,
Design and Efficiency,
The Benjamin/Cummings, 1991.
- [17] G. Nemhauser, L. Wolsey:
Integer and Combinatorial Optimization,
Wiley, 1988.
- [18] Felhasznált hálózatok rajza:
<http://opt.tmit.bme.hu/~mitcsenkov/networks>
- [19] T. Cinkler, D. Meskó, A. Mitcsenkov, G. Viola:
Adaptive Shared Protection Rearrangement,
Design of Reliable Communication Networks,
(DRCN) 2005.