Budapesti Műszaki és Gazdaságtudományi Egyetem
Távközlési és Médiainformatikai Tanszék

Hozzárendelt védelem gyakorlati megvalósíthatóságának vizsgálata hálózati kódolás alkalmazásával

Pašić Alija

TDK dolgozat

Konzulensek:

Dr. Babarczi Péter és Dr. Tapolcai János

MTA-BME Jövő Internet Kutatócsoport
Nagysebességű Hálózatok Laboratóriuma, Távközlési és Médiainformatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Budapest, Magyarország
2012.

Abstract

Within the next 10–15 years Internet traffic is expected to grow up to 50 times its present value, and all the traffic will be carried IP based in the upper layer [22]. The current IP based solutions are not suitable for providing reliable connections, because of the slow recovery mechanisms. Therefore, the protection in the lower, optical layer is more and more important, because already a short-therm outage leeds to the loss of huge amount of data (in order of terabytes). Furthermore, the customores require higher QoS from the service providers, by using different real time applications.

Nowadays the most widespread technology in the optical backbone is the 1+1 dedicated protection, which sends the data simultaneously on two edge and node disjoint paths, ensuring immediate restoration of the connection, if one of the paths falls out. On the other hand, because of the compexity of the network and the protection of the mulitiple link failures, requiered by high QoS, this technology will be applied only in a limited way, not to mention its high capacity demand. The Generalized Dedicated Protection has been proposed, in order to eliminate these disadvantages [4]. The GDP suits felxible the demands of customers and provides reliable connectivity.

Based on the optical equipments available at the network nodes, different GDP problems can be formulated. The solution can be in the form of non-bifurcated or bifurcated flows. The non-bifurcated method is simple, but has a bad resource utilization. On the other hand, by the bifurcated solution, in exchange for low bandwidth reservation, we have to use network coding with high complexity to ensure resilient and robust protection.

In my work I present a new GDP solution, that combines the advantages of the aforementioned ones. The method I suggest, allows data splitting (bifurcation), but pays regard to the practical feasibility, too. Considering the most recent network coding results, I place a special emphasis on the case, when the flow can be divided in two parts. Using simulations, I investigate the case, when the data is splitted in more than two parts, increasing the complexity. I try to figure out how close we can come to the optimal solution, that can be computed in polinomial time, but it’s difficult to implement in practice and requires complex coding and decoding for immediate recovery [26].

Kivonat

A jelenlegi trendeket figyelembe véve tíz-tizenöt éven belül az internet forgalom az ötvenszeresére növekedhet, valamint a szélessávú kommunikációs hálózatok a felső rétegben IP alapon fogják szállítani mind az adat-, mind a hangforgalmat [22]. A jelenlegi IP megoldások nem alkalmasak megbízható összeköttetések kialakítására, mivel a meghibásodások utáni (lassú) helyreállításon alapszanak, ezért egyre fontosabbá válik az összeköttetések védelme az alsó, optikai rétegben, mivel a rövid ideig tartó kiesések is hatalmas (akár terabyte nagyságrendű) adatmennyiségek elvesztéséhez vezetnek. Továbbá a felhasználók is egyre szigorúbb megbízhatósági követelményeket támasztanak a szolgáltatók felé a különböző valós idejű alkalmazások használatával.

A jelenleg leggyakrabban használatos védelmi megoldás az optikai rétegben az 1+1 hozzárendelt védelem, amely két független útvonalon párhuzamosan küldi a felhasználói adatot, ezáltal biztosítva az azonnali helyreállítást az egyik útvonal kiesése esetén. Viszont a hálózat és a magas megbízhatóságú összeköttetések kialakításához védendő többszörös link hibák komplexitása miatt a jövőben már csak korlátozottan alkalmazható, nem beszélve a nagy kapacitás igényéről. Ezen hátrányok kiküszöbölésére javasolták az általános hozzárendelt védelmet (GDP- Generalized Dedicated Protection) [4], amely a felhasználó igényeihez rugalmasan illeszkedő, megbízható összeköttetések kiépítését támogató védelemi megoldás. A hálózatban rendelkezésre álló optikai eszközök, illetve az alkalmazott technológiai megoldásoknak megfelelően a GDP védelem vagy osztatlan formában küldte a felhasználói adatot minden független útvonalon, vagy az adat tetszőleges osztását megengedte. Az előbbi esetben a módszer az egyszerűsége ellenére az 1+1 védelemhez hasonlóan magas erőforrás-foglalást eredményezett, míg az utóbbi esetben alacsony sávszélesség foglalásért cserébe magas komplexitású hálózati kódolást (network coding) kellet alkalmazni az azonnali helyreállítás biztosítása érdekében.

A dolgozatomban egy olyan új GDP megoldást javasolok, mely az előző két módszer előnyeit ötvözi. Az általam javasolt módszer felhasználói adatok osztását megengedi, de mindvégig szem előtt tartva annak gyakorlati megvalósíthatóságát. A legújabb hálózati kódolás eredményeket figyelembe véve külön hangsúlyt fektetek annak az esetnek a vizsgálatára, amikor a felhasználói adat pontosan két részre osztható. Szimulációk segítségével megvizsgálom, hogy az adat további osztásával (a komplexitás növelésével) mennyire közelítjük meg az optimális erőforrás használatot, amely ugyan polinom időben kiszámolható, viszont gyakorlatban nehezen megvalósítható, egyrészt mivel a felhasználói adat tetszőleges osztása nem lehetséges, másrészt igen komplex kódolás szükséges az azonnali helyreállításhoz [26].

Tartalomjegyzék

Ábrák jegyzéke

1. fejezet

Bevezetés

Manapság egyre nyilvánvalóbb, hogy modern társadalmunk egyre nagyobb mértékben támaszkodik a kommunikációs hálózatra, így annak kifogástalan, hibamentes működése egyre fontosabb témává válik. Ennek megfelelően az utóbbi időben a kommunikációs hálózatokon szállított adatmennyiség jelentősen megnövekedett. Ez a növekedés leginkább az internet és a hozzá kapcsolódó szolgáltatások népszerűségének köszönhető. Gondoljunk csak a nagy felbontású video műsorok szétosztására, vagyis a különböző Video on Demand, illetve IP tévé szolgáltatásokra. Ez a tendencia az elkövetkezendő időben minden bizonnyal folytatódni fog, ahogy a szolgáltatások, illetve hálózatok konvergenciája is. De a gyorsan növekvő adatmennyiség, a részben emiatt bevezetésre kerülő új technológiák ellenére (vagy talán éppen amiatt is) és a nagy verseny miatt a szolgáltatókkal szembeni minőségi elvárások, illetve követelmények egyre nőnek [34].

A hatalmas adatmennyiség és a rendkívül nagy felhasználószám (akár több tíz millió) miatt a gerinchálózatok minőségi követelményei a legmagasabbak. Itt a rövid ideig tartó kiesések esetén is hatalmas adatmennyiségek vesznek el, amit a szolgáltatók a szolgáltatás-minőség (QoS) szinten tartása miatt nem igen engedhetnek meg maguknak.

Szolgáltatói szempontból is egyre fontosabbá válik a hálózati rendelkezésre állás növelése mivel sok üzleti felhasználó hajlandó lényegesen nagyobb összeget fizetni a megbízható szolgáltatásért cserébe, de természetesen ezért garanciát kér a szolgáltatótól, amely megszegése esetén annak komoly anyagi vonzattal kell számolnia. Ilyen felhasználók közé tartozhatnak a hálózatban valós időben játszható játékokat üzemeltető cégek, mivel azok profitjuk igen nagyban függ az adott hálózati szolgáltatás minőségétől. Másik jellemző példa a banki szektor résztvevői, akik szintúgy nem engedhetik meg maguknak a nagy hálózati kieséseket. Természetesen, hogy a szolgáltató fel tudjon vállalni egy ilyen megállapodást, ismernie kell a saját hálózatának paramétereit, illetve a felhasználók igényeit.

Összességében elmondható, hogy a bevétel növelése érdekében a szolgáltatóknak érdekükben áll minél megbízhatóbb hálózatok tervezése. Persze az egyes alkalmazások lényegesen eltérő QoS követelményei miatt gazdaságtalan valamennyi összeköttetést a magas QoS követelményeknek megfelelően

kiépíteni, ezért a felhasználói igényekhez legjobban illeszkedő szolgáltatás nyújtása a cél. Ennek megfelelően a szolgáltatás megkezdése előtt a két fél aláír egy úgynevezett szolgáltatásminőségi szerződést (SLA), amelyben szerepelnek a minimális QoS követelményei és az ennek ellenében a szolgáltatásért fizetendő összeg, illetve annak elmaradása esetén a kártérítés mértéke, amit a szolgáltató köteles fizetni a felhasználónak.

A vállalható biztonság szintje természetesen függ a hálózatban alkalmazott technológiáktól is. A mai gerinchálózatokban már optikai vezetőket alkalmaznak, amelyekben a hagyományos áramkörkapcsolás nem kivitelezhető. A hálózatok evolúciója mindenképpen a teljesen átlátszó gerinchálózat felé tendál, de maga a megvalósítás, illetve átállás biztosan eltart még egy darabig. Addig is több technológiájú hálózat fog működni egymás mellett, ezekről a 2. fejezetben részletesebben is szó esik.

Magára a hálózat megbízhatóságára jellemző szám a rendelkezésre állás, ami annak a mérőszáma, hogy egy adott időszakban milyen valószínűséggel működik egy hálózati elem, illetve ezen elemeken megvalósított összeköttetés. A nem rendelkezésre állás pedig ennek ellenkezője; annak a valószínűsége, hogy egy időszakban valamely hálózati elem hibája miatt az összeköttetés használhatatlan állapotban van, azaz megszakad. A rendelkezésre állás meghatározásához ismerni kell az adott eszközök – például optikai kábelek, routerek – meghibásodásának okait, valószínűségeit [3]. Alapvetően tervezett, illetve nem tervezett kiesésekről beszélhetünk; a lényegi különbség, hogy a nem tervezett hibák esetén azok semmiképpen sem szándékosan következnek be, míg a tervezettnél egy előre ismert időszakban és az előre megadott eszközök nem állnak rendelkezésre. Ezért a tervezett kiesések számunkra nem érdekesek, mivel könnyű tervezni velük és védeni őket. A nem tervezett kieséseknek viszont rengeteg kiváltó okuk lehet, és csak statisztikai módszerek segítségével becsülhetjük azok előfordulásának valószínűségét.

A nem tervezett meghibásodások hatékony és rugalmas védelméről szól a dolgozatom, pontosabban a hálózati kódolást is használó, adott osztású általános hozzárendelt védelem (GDP – Generalized Dedicated Protection) vizsgálatáról. A 2. fejezetben a jelenleg használható gerinchálózati technológiákat foglaltam össze, majd részletesen tárgyalásra kerülnek a különböző hibák lehetséges okai (szó esik magáról a hálózati modellről, illetve matematikai leírásról is). A 3. fejezetben a hozzárendelt védelmi mechanizmust mutatom be, majd a 4. fejezetben egy új védelmi eljárásról, az úgynevezett GDP-ről lesz szó. Az 5. fejezetben ezen védelmi módszerek megvalósíthatóságát tárgyalom, míg azok szimulációs eredményét a 6. fejezet tartalmazza. Végül a 7. fejezetben összefoglalom a dolgozatomban ismertetett eredményeket.

2. fejezet

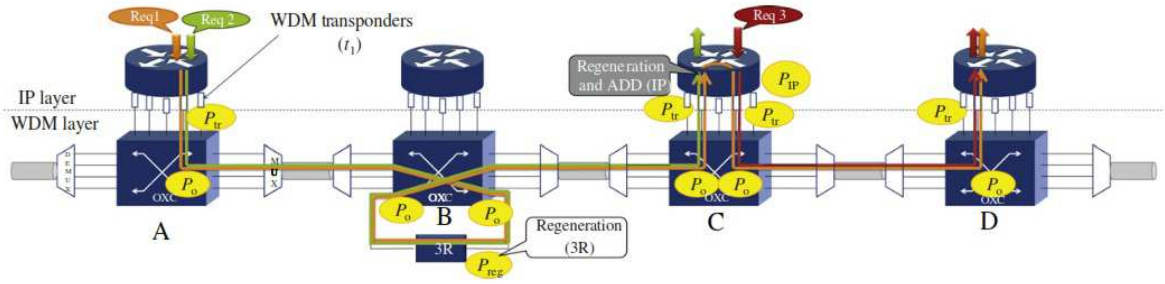
Hálózati technológiák és a reprezentációs modell

Ebben a fejezetben összefoglalom a jelenlegi gerinchálózati technológiákat és azok fő jellemzőit, aztán pedig bemutatásra kerül a hálózat gráf reprezentációs modellje, valamint a hálózati elemek képességeiről is szót ejtek, amelyek igen fontos szerepet játszanak a védelmek gyakorlati megvalósíthatóságának szempontjából.

2.1. A hálózati technológiák

Az előrejelzések szerint tíz-tizenöt éven belül az internet forgalom az ötvenszeresére növekedhet [22]. Emiatt a robbanásszerű növekedés miatt a jövőben a szélessávú kommunikációs hálózatok IP alapon fogják szállítani mind az adat-, mind a hangforgalmat. Ahhoz, hogy a gerinchálózat képes legyen ekkora adatforgalom továbbítására, optikai hullámhosszosztásos WDM (Wavelength Division Multiplexing), illetve DWDM (Dense Wavelength Division Multiplexing) hálózatra van szükség. A WDM alapú hálózatok lényege, hogy a költséges, új optikai kábelek kiépítése helyett a hálózat topológiájának megváltoztatása nélkül megnövelik az egy optikai kábelen belül használatos csatornák számát, így lényegesen növelve az átviteli sávszélességet. Az optikai WDM hálózatok képesek a forrás és cél közötti fényutak különböző hullámhosszokon való szállítására, ráadásul a legtöbb esetben a köztes csomópontoknál nincs is szükség semmilyenfajta adatfeldolgozásra, így csökkentve a szükséges O/E/O (optical/electrical/optical) átalakítások számát. Természetesen ez a képesség nagyban függ a hálózati elemektől, de attól is, hogy milyen protokollt használunk a WDM fölött.

Jelenleg a legelterjedtebbek az SDH/SONET (SDH – Synchronous Digital Hierarchy, SONET – Synchronous Optical NETworking) hálózatok, de a forgalom folyamatos növekedése miatt egy lassú evolúciót figyelhetünk meg az OTN (Optical Transport Network) hálózatok felé [34]. Fontos különbség, hogy az OTN hálózatok teljes hullámhossz csatornákat egységes egészként kapcsolnak, míg az SDH/SONET hálózatok a hullámhossznál kisebb granularitási egységekkel teszik ugyanezt (időosztás



2.1. ábra. A Tl-IPoWDM hálózati illusztrációja [22]

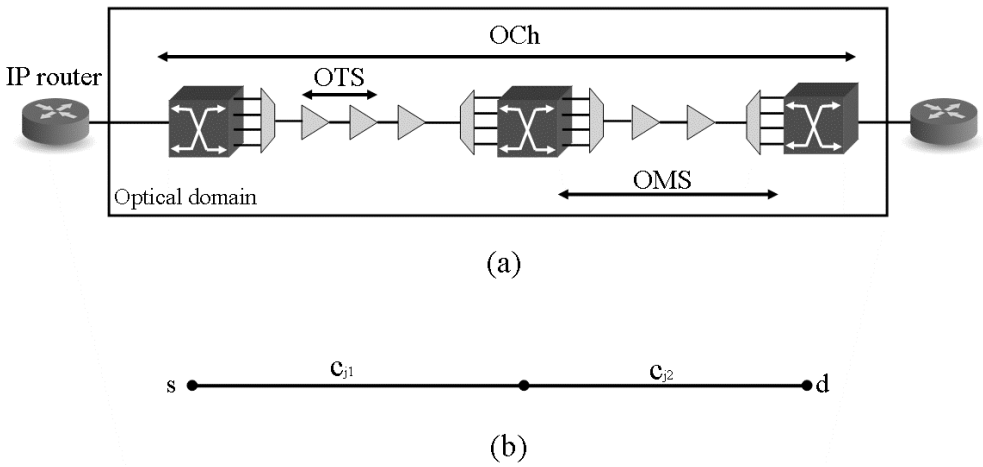
alkalmazásával). Jelenleg négy technológia van még versenyben egymással: a basic IP over WDM (B-IPoWDM), az IP over SDH/SONET over WDM (IPoSDH), a Transparaent IP over WDM (Tp-IPoWDM) és a Translucent IP over WDM (Tl-IPoWDM) [22].

Az alap B-IPoWDM estén az IP routerek pont-pont összeköttetésben állnak egymással, így mind a forgalom kapcsolása, mind a grooming (forgalomszövés) – amely valójában a kisebb sávszélességű adatfolyamok, nagyobb sávszélességű adatfolyamokká való egyesítését jelenti – az elektromos tartományban történik. Tehát minden egyes csomópontban megvalósul az O/E/O átalakítás, amelynek egyrészről magas az energia igénye, másrészről pedig szűk keresztmetszetet eredményez a hálózat sebességében.

IPoSDH-nál az IP forgalom SDH konténerekbe helyeződik, majd ezután az elektromos jelet átalakítják optikai jellé egy transponder segítségével, ezután pedig a forgalom a WDM csatornán keresztül továbbítódik. Itt is minden csomópontnál szükség van O/E/O átalakításra; annyi különbség van a B-IPoWDM-hez képest, hogy a hálózatban található digitális mátrix kapcsolók (DXC – Digital Cross-Connect) képesek az adatfolyam kapcsolására anélkül, hogy forgalomszövést vagy szétbontást végeznének a folyamon.

Az áttetsző Tp-IPoWDM hálózatokban már találhatóak optikai mátrix kapcsolók (OXC – Optical Cross-Conect) és rekonfigurálható optikai add drop multiplexerek (ROADM – Reconfigurable Optical Add-Drop Multiplexer), így maga a grooming, azaz forgalomszövés nagy része megoldható az optikai rétegben, de a routerek továbbra is kapcsolhatnak. Viszont a jel regenerálása itt még nincs megoldva az optikai tartományban, erre csak a routerek képesek. Így, ha arra is szükség van, kénytelenek vagyunk az O/E/O átalakítás végrehajtására.

Az átlátszó Tl-IPoWDM esetén az optikai mátrix kapcsolók (OXC) és a rekonfigurálható optikai add drop multiplexerek (ROADM) mellett a hálózatban már megtalálhatóak az optikai 3R (erősítés – reamplification, jelformázás – reshaping, és időzítés – retiming) regenerátorok, amelyek az újraerősítést, újraformázást és újraidőzítést hajtják végre. Így ebben a rendszerben ténylegesen csak a forrásnál és a nyelőnél van szükség E/O, illetve O/E átalakításra, ahogy ez az a 2.1. ábrán is megfigyelhető.



2.2. ábra. Optikai csatorna (Optical Channel OCh), optikai multiplex szakasz (Optical Multiplex Section, OMS) és az optikai továbbító szakaszok (Optical Transmission Sections, OTS) valamint a hozzá tartozó gráf reprezentáció [15]

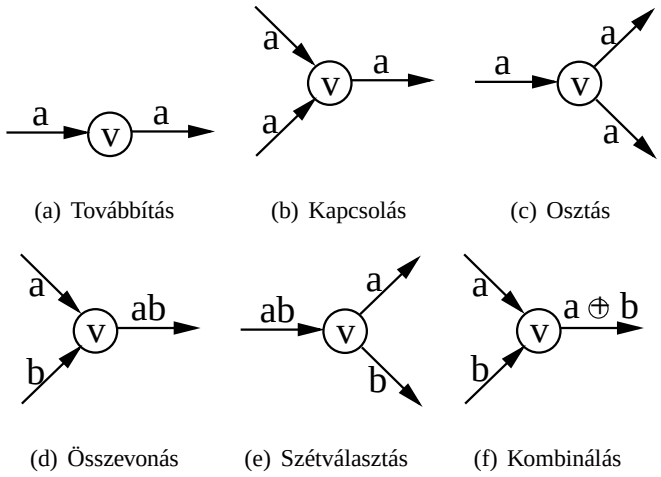
A hálózat helyreállítás tekintetében elmondható, hogy az SDH/SONET alapú technológia általánosan elfogadott, mint egy olyan technológia, amely már bebizonyította, hogy képes nagyon gyors védelmi kapcsolásra (nagyságrendileg 50-60 milliszekundum alatt). Egyrészt ez kifinomult felügyeleti folyamatok révén valósul meg, amelyek kiterjednek a hibajelzésre, értesítésre és a terjedési folyamatra [34]. Másrészről az automatikus védelmi kapcsolás (Automatic Protection Switching, APS) protokoll felelős a gyors kapcsolásért az érintett (hibás) erőforrások felől az ideiglenes védelmi / biztonsági erőforrásokra. A teljesen áttetsző hálózatok esetén viszont a hiba lokalizálása problémákat vet fel, mivel a hibaüzenet terjedését az O/E/O átalakítások már nem szűrik ki. Ezáltal a hibaüzenetek sokkal tovább terjednek egy SDH/SONET hálózathoz viszonyítva, így egyes esetekben a helyreállítás hosszabb időt vehet igénybe, ami természetesen befolyásolja a rendszer rendelkezésre állását is.

2.2. Az optikai hálózat matematikai modellje

Ennek az alfejezetnek a célja, hogy bemutassa az optikai hálózatnak a $G = (V, E)$ gráf reprezentációját, amely bemenetként szolgál a különböző algoritmusok számára, legyen útvonalválasztó vagy védelmi funkció megvalósítását vizsgáló algoritmusról szó.

Az optikai hálózat két rétegből áll: a *fizikaiból*, amely vezetékekből és optikai mátrix kapcsolókból áll (WDM réteg), illetve egy *logikaiból*, amely optikai linkekből (vagyis fényutakból) és az ezek a végződtetéseként értelmezett csomópontokból tevődik össze (IP réteg) [32]. Manapság a logikai réteg már dinamikusan is konfigurálható (milliszekundumok alatt), gondoljunk csak a GMPLS-re [17] (Generalized Multi-Protocol Label Switching), ezért munkámban a fizikai réteg reprezentációját használtam, mivel az statikusnak tekinthető. Ahogy a 2.2. ábrán is látszik, a linkek az optikai multiplex szakaszoknak (OMS – Optical Multiplex Section) felelnek meg, továbbá mindegyik j linkekhez tartozik egy c_j

jelölésű költség, amely az egységnyi igény elvezetésének költségét jelenti, amely függhet a link hosszától, az optikai erősítők számától, vagy éppen a link kihasználtságától is. Mindegyik élhez tartozik egy kapacitás érték is, amely megadja, hogy maximálisan menyi igényt képes elszállítani az adott link. Az *összeköttetés igényeket* pedig a $\mathcal{D} = (s, d, b)$ vagy néha a $\mathcal{D} = (s, d, b, t_a, t_d)$-vel szokás megadni, ahol s a forrás, d pedig a cél csomópont, b pedig az összeköttetés kapacitás igénye (t_a az érkezés t_b pedig az igény megszűnésének az idejét jelöli).



(a) Továbbítás (b) Kapcsolás (c) Osztás

(d) Összevonás (e) Szétválasztás (f) Kombinálás

2.3. ábra. Alapvető csomóponti szerepek, melyekbe az összeköttetés által használt valamennyi v csomópont besorolható [3].

A gráfban általában irányítatlan éleket használunk, amelyek kétirányú kapcsolatot jelentenek, de egyes alkalmazások modellezéséhez szükség lehet az irányított élekre, amelyek az egyirányú kapcsolatot jelképezik. A hálózati csomópontok is különböző képességekkel rendelkezhetnek, ezek befolyásolják a hálózat intelligenciáját. Hat alapvető hálózati szerepe lehet egy csomópontnak: továbbítás, kapcsolás, osztás, összevonás, szétválasztás és kombinálás (2.3. ábra). Ezek a különböző védelmi megoldások gyakorlati megvalósításánál játszanak fontos szerepet, és az egyes módszerek bemutatásánál ismertetem részletesebben a szerepüket (a 4. fejezetben).

2.3. A hálózati kiesések és annak okai

Ebben a fejezetben bemutatásra kerülnek a hálózati kieséseket okozó hibák. Alapvetően két fajta kiesésről beszélhetünk: a tervezett, illetve a nem tervezett kiesésről. A tervezett kiesésekhez tartoznak a karbantartási, illetve felújítási munkák miatti kimaradások; ezeknek az aránya némely esetben elérheti akár az összes kimaradás húsz százalékát is [20]. Ezek azonban sokkal kevésbé érdekesek számunkra, mivel könnyű számolni velük és védeni őket. Ennél sokkal érdekesebbek a nem tervezett kiesések. Ezeknek rengeteg oka lehet, de általában öt csoportba szokás osztani azokat:

– **Hardver hibák**, amelyek általában az elöregedés miatt lépnek fel. Hibaforrás lehet szinte bármi a

processzortól kezdve, a memórián át, a táp meghibásodásáig.

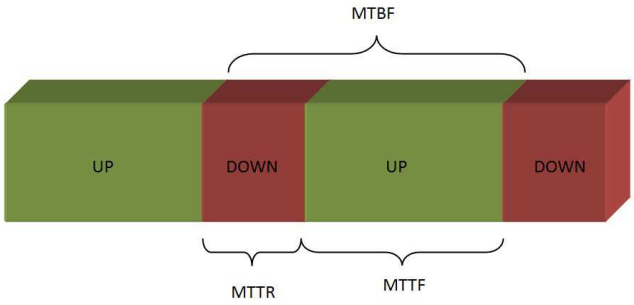
- **Szoftver hibák**, amelyek közé tartoznak a különböző rossz implementációból, vagy például memória-olvasási, illetve írási hibából fakadó kiesések.
- **A emberi hanyagság, illetve figyelmetlenség** által okozott (operátor miatti) meghibásodások. Példaként elég megemlíteni a téves konfigurálást, vagy a helytelen biztonsági beállításokat.
- **A felhasználónak köszönhető hibák**, mint például router lopás, csomópont elleni támadások, vagy éppen a DoS (denial-of-service) támadás. Ide tartoznak még az akaratlanul okozott kiesések is, mint amikor azért bénul le egy hálózat, mert túl sokan akarják egyszerre használni, például egy földrengés után.
- **Környezeti hatások miatti meghibásodások**: ez a kategória sok különböző hibát felölelő csoport, ide tartoznak a munkagépek által okozott kábel átvágások, vagy éppen a rágcsálók okozta kábelhibák, az energiaellátás kimaradása miatti kiesések és a különböző természeti katasztrófák, mint a tűzvész, földrengés és terrorista támadás miatti hibák.

A leggyakoribb meghibásodások az optikai kábelek meghajlása, elszakadása, az eszköz meghibásodások, az emberi mulasztások, illetve a hálózat elleni kifinomult támadások. Érdemes megemlíteni, hogy ideális esetben a fizikai meghibásodásokat detektálják, lokalizálják és helyreállítják még az optikai rétegben azelőtt, hogy a felsőbb rétegbeli protokollok ezt érzékelnék és hozzálátnának a saját időigényes helyreállítási folyamatukhoz, mely az alkalmazások szempontjából megengedhetetlenül hosszú kiesést eredményezne. A következő alfejezetben ismertetem az imént felsorolt meghibásodások modellezésére leggyakrabban használt módszert (SRLG – Shared Risk Link Groups), valamit a rendelkezésre állás matematikai modelljéről is szó esik.

2.4. Rendelkezésre állás és a hibák modellezése

A hálózat rendelkezésre állása a megbízhatóság egyik legfontosabb mérőszáma, amely annak a valószínűsége, hogy egy adott időszakban mennyi ideig működik minden hálózati elem, azaz a hálózat rendelkezésre áll, vagyis a felhasználó igénybe tudja venni a szolgáltatást. Ennél a legtöbb esetben kevesebb is elég, hiszen egy összeköttetés nem használja a hálózat valamennyi erőforrását. Viszont egy megbízható elemekből felépített megbízható hálózat tipikusan magasabb rendelkezésre állású összeköttetések kiépítésére alkalmas, ezért a következőekben ismertetem egy megbízható hálózat tervezéséhez szükséges modelleket.

A hálózati eszközök rendelkezésre állásának a meghatározására több módszer létezik. A gyakorlatban is használatos egyik módszer, hogy katonai kézikönyvet használunk [21]. A kézikönyv megfigyelésen alapszik, azaz a különböző eszközök meghibásodásait megfigyelték, és ezután erre próbáltak



2.4. ábra. Rendelkezésre állás

görbéket illeszteni. Ma már általánosan elfogadott, hogy a meghibásodás között eltelt időt exponenciális eloszlással lehet közelíteni, amelynek λ paramétere az adott eszköz jellemzője, a javítási idő pedig szintúgy exponenciális eloszlásúnak tekinthető, amelynek paramétere μ . Ez a rendszer egy kétállapotú Markov láncnak tekinthető, amelynek a stacionáris eloszlása a rendelkezésre állást adja meg, vagyis:

$$\mathbf{A}\text{vailability} = \frac{\mu}{\mu + \lambda}. \tag{2.1}$$

Egy könnyebben érthető, de ekvivalens definíciója egy hálózati elem rendelkezésre állásának az, hogy bevezetjük az alábbi időket (lásd a 2.4. ábrát):

Mean Time To Failure, MTTF. – két meghibásodás között eltelt várható idő (MTTF $= \frac{1}{\lambda}$),

Mean Time To Repair, MTTR. – az átlagos ideje az adott meghibásodás megjavításának, vagy a meghibásodott hálózati elem cseréjének (MTTR $= \frac{1}{\mu}$),

Mean Time Between Failures, MTBF. – az átlagos eltelt idő a következő hibáig, az előző két idő összegeként számolható.

Ezen értékek ismeretében pedig megvizsgáljuk az adott intervallumok hosszának egymáshoz való viszonyát, így megkapjuk a j elem a_j rendelkezésre állását (2.2), és u_j nem rendelkezésre állását (2.3), azaz:

$$a_j = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}, \tag{2.2}$$

$$u_j = \frac{\text{MTTR}}{\text{MTTF} + \text{MTTR}}. \tag{2.3}$$

Természetesen a kettő összege egy j eszközre egy ($a_j + u_j = 1$), mivel az eszköz vagy működik, vagy pedig nem. A hálózat rendelkezésre állása akkor magas, ha az egyes eszközöké is az. Ezt pedig kétféleképpen lehet elérni: egyrészt az MTTF értékét kell magasan taratani, másrészt pedig a MTTR-t alacsonyan. Jobb minőségű (és drágább) eszközök használatával, valamint a karbantartás rendszerességével, illetve magas színvonalával a MTTF értékét tudjuk magasan tartani. A gyors javítással, amelyet

például a huszonnégy órás ügyelet bevezetésének segítségével érhetünk el, a MTTR értéke csökkenthető. Természetesen gerinchálózatok esetében mindig a lehető legmagasabb rendelkezésre állásra törekszünk.

Amikor egy hálózatrésznek vagy (egész) hálózatnak a rendelkezésre állását szeretnénk meghatározni, akkor figyelembe kell vennünk a benne foglalt összes hálózati elemet. Miután meghatároztuk az összes elem rendelkezésre állását, az s forrásból a d nyelőbe menő egyetlen útvonal (*üzemi útvonal*, working path, $\mathcal{W}$) rendelkezésre állását a soros szabály alkalmazásával tudjuk számolni, hiszen bármelyik útvonal menti elem kiesése az összeköttetés megszakadásához vezet, azaz:

$$A_{s-d} = \prod_{j \in \mathcal{W}} a_j. \tag{2.4}$$

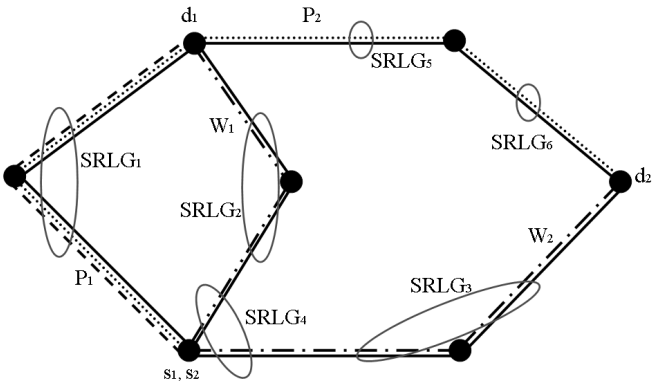
Hasonlóan, ha egy üzemi út mellé egy tőle csomópont-diszjunkt (*védelmi út*, protection path $\mathcal{P}$) útvonalat is foglalunk (azaz például $1+1$ védelemet alkalmazunk), akkor a párhuzamos szabály alkalmazható az összeköttetés rendelkezésre állásának kiértékelésére:

$$A_{s-d} = 1 - [(1 - \prod_{j \in \mathcal{W}} a_j) \cdot (1 - \prod_{j \in \mathcal{P}} a_j)]. \tag{2.5}$$

Érdemes megjegyezni, hogy ha az összeköttetésünk számára foglalt részgráf nem soros-párhuzamos, akkor a rendelkezésre állásának kiértékelése igen nehéz feladattá válik [28], a probléma #P-teljes, azaz leegyszerűsítve már a megoldások leszámlálása is NP-teljes feladat. Ekkor több pontos és közelítő módszer is létezik az összeköttetés rendelkezésre állásának meghatározására [33]. A 4. fejezetben bemutatott, illetve az általam javasolt általános hozzárendelt védelem esetében is az általánosságban ugyan exponenciális pivotal dekompozíció [25] hatékonyan alkalmazható az összeköttetés pontos rendelkezésre állásának meghatározására.

Az eddig tárgyalt modellek a hibák egymástól való függetlenségét feltételezik, ami sajnos a gyakorlatban koránt sem igaz [30]. Így ahhoz, hogy megfelelően tudjuk modellezni a valóságot, figyelembe kell vennünk az eszközök, illetve meghibásodások egymásra gyakorolt hatását. Erre több módszer is lehetőséget nyújt; a leggyakrabban használt megoldás úgynevezett közös kockázatú csoportok (SRLG-k) definiálása a hálózatban. Ennek lényege, hogy csoportosítjuk azokat a hálózati elemeket (vagy akár a logikai IP linkeket), amelyek valamilyen esemény, hálózati hibahatás következtében egyszerre hibásodnak meg, így képessé válunk látszólag független linkek fizikai és földrajzi összefüggőségeit is figyelembe venni.

A közös kockázatú csoportokat a szolgáltatók a megadott szolgáltatási minőségi (QoS) osztályhoz definiálják, és a cél az összeköttetéseket olyan módon kiépíteni, hogy az adott esemény bekövetkezésekor kieső (azaz a hibához tartozó SRLG-ben megadott) linkek eltávolítása esetén is létezzen aktív útvonal az összeköttetés számára, biztosítva ezzel az összeköttetés azonnali helyreállítását. Mivel a legmagasabb nem rendelkezésre állási valószínűséggel a linkek rendelkeznek ($\approx u_i = 10^{-2}$) [35], így annak a valószínűsége, hogy egyszerre több elem is meghibásodik a legrosszabb esetben is 10^{-4} nagyságrendű, de tipikusan kisebb, mint 10^{-6}, ami még igen magas QoS igények esetén is elhanyagolható értéknek



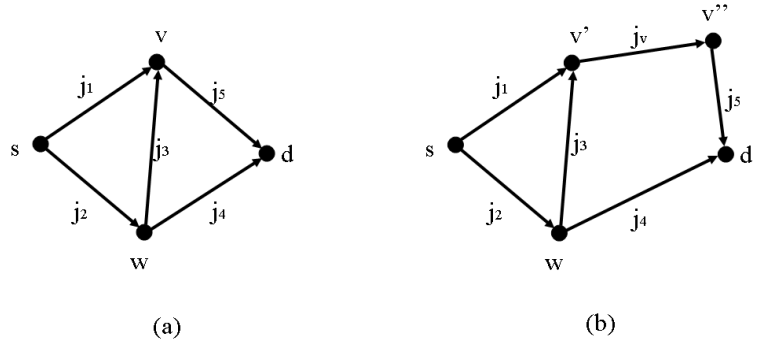
2.5. ábra. A hálózaton meghatározott SRLG-k; a két üzemi út ($\mathcal{W}_1$ az s_1 forrás és d_1 nyelő között és a $\mathcal{W}_2$ az s_2 forrás és d_2 nyelő között) él-diszjunktak, de áthaladnak egy közös SRLG-n (az SRLG_4-en) [3]

számít [3]. Ezért a gyakorlatban a szolgáltatók minden egyszeres link és csomópont hibához rendelnek egy SRLG-t, illetve tipikusan az adott hálózatban statisztikailag leggyakoribb többszörös meghibásodásokhoz. Ilyen SRLG-re való példaként lehet említeni az egymáshoz közel- vagy egy ideig közösen futó kábeleket/linkeket, melyek egy esetleges kábelátvágásnál, rágcsálókárnál vagy földrengésnél nagy valószínűséggel egyszerre esnek ki.

Fontos, hogy egy SRLG tetszőleges számú linket tartalmazhat, és egy link tetszőleges számú SRLG-ben lehet benne, tehát egy nagyon rugalmas modellről van szó. A 2.5. ábrán látható egy példa az SRLG definiálásra: megfigyelhetőek egyszeres (pl: SRLG_5) és kétszeres (pl: SRLG_1) hibák, valamint az SRLG-k átlapolódása is (pl: SRLG_4 és SRLG_3). Továbbá, az is látszik, hogy két útvonal lehet diszjunkt (s_1 és d_1 között a $\mathcal{W}_1$, s_2 és d_2 között a $\mathcal{W}_2$), de attól még tartozhatnak egy SRLG-be (SRLG_4).

Természetesen léteznek olyan hibák, amelyek ellen nem lehet védekezni, például ha a topológiát reprezentáló gráf egy teljes vágása kiesik. Nézzük meg a 2.6. ábrán látható hálózatot, abban az esetben, ha s-ből küldünk adatot d-be, és egyszerre kiesik a j_1-es és j_2 link, akkor már semmiképpen sem tudjuk fenntartani az összeköttetést. Az ilyen meghibásodásokkal védelmi szempontból nem tudunk mit tenni, hiszen semmilyen védelmi módszerrel nem tudunk ellenük védekezni, ezért feltesszük, hogy egyetlen SRLG sem tartalmazza a gráf egy tetszőleges $s - d$ vágását. De természetesen a rendelkezésre állás szempontjából igenis fontosak, és számolni kell velük.

A csomópontokat a legtöbb esetben hibátlannak tekintjük, vagyis rendelkezésre állásuk egy. Ez azonban bizonyos esetekben modellezési szempontból nem megfelelő; ekkor egy segédgráfot hozunk létre a [23] leírtak alapján (lásd a 2.6. ábrát). Vagyis az összes irányítatlan élt kicseréljük kétirányú irányított élekre, utána pedig az összes v csomópontot két csomóponttá bontjuk, v'-re és v''-re, majd behúzunk közéjük egy $v' \rightarrow v''$ élt. Minden v-be bemenő élt a v'-be irányítunk, míg minden a v-ből kimenő és v''-ből is kimenő éllel reprezentálunk, ahogy azt a 2.6(b) ábra is mutatja. Az így kapott segédgráfban már a csomóponti hibákat is tudjuk modellezni SRLG-k segítségével, a v' és v'' csomópontok közötti



2.6. ábra. Példahálózat $c_{j_1} = c_{j_4} = 3; c_{j_2} = c_{j_3} = c_{j_5} = 1$ élköltségekkel, valamint a v csomópont szétválasztás eredményeképpen kapott segédgráf [15]

link kiesésével.

Meg kell említeni, hogy az optikai hálózatokban használt eszközök rendelkezésre állása igen magas, ezért annak a valószínűsége, hogy kettő vagy háromnál több független eszköz hibásodik meg egyszerre, igen kicsi, így ezeket a gyakorlatban elhanyagolják. Természetesen lehet tudomásunk olyan tényezőkről vagy hatásokról, amelyek indokolttá teszik, hogy több eszköz egyszeri kiesését is vizsgáljuk, ekkor ezeket érdemes felvenni az $\mathcal{F}$ SRLG listába. Munkámban az egyszeres és szomszédos kétszeres hibák védelmét vizsgáltam [20]. Ezt a modellt az indokolja, hogy a statisztikailag leggyakrabban bekövetkező egyszeres linkhibákon felül a közös kábelbe fektetett optikai fényszálak egyszerre történő meghibásodása tekinthető a leggyakrabban előforduló meghibásodás típusnak optikai gerinchálózatokban [5].

2.5. Azonnali helyreállítás biztosítása az SRLG modellben

Az optikai hálózatokban használatos technológiák mindegyike a virtuális áramkörkapcsolás elvén működik, így egy-egy SRLG kiesése az áramkör megszakadását eredményezheti. Ennek az újbóli kiépítése időigényes lehet, ami ellentétben állhat a QoS követelményekkel. Ezért fontos, hogy a szolgáltatók egy adott QoS osztályhoz tartó összes hibára egyaránt fel legyenek készülve és kezelni tudják azokat. Ennek modellezésére bevezetésre kerül az $\mathcal{F}$ lista, amely egy adott szolgáltatásminőségi szinthez védendő SRLG-k listáját tartalmazza. Ezekkel a hibákkal szemben kell egy azonnali helyreállítást biztosító védelemi módszernek ellenállónak és robusztusnak (E&R) bizonyulnia [3]. Az *ellenálló védelem* azt jelenti, hogy a $\mathcal{F}$-ben felsorolt összes SRLG kiesését képes az összeköttetés túlélni megszakadás nélkül, a *robusztusság* pedig azt, hogy nem szükséges sem jelzések küldése a vezérlő síkon, sem pedig a hálózati kapcsolók (OXC-k) újrakonfigurálása a meghibásodást követően; elegendő csupán a nyelő intelligenciája, illetve lokális döntéseken alapuló kapcsolás ahhoz, hogy az adat továbbra is kinyerhető legyen bejövő folyamokból a nyelőnél. A dolgozatomban kizárólag ellenálló és robusztus védelmi módszerekkel foglalkoztam, hiszen ezek azok, amelyek biztosan garantálni tudják az azonnali helyreállítást a jövő optikai hálózataiban. Mivel az E&R módszerekben valamennyi linken folyamatosan küldeni kell az adatot, ezért

a gyakorlati megvalósíthatóságon kívül ezen módszerek erőforrás-foglalása (elsősorban sávszélesség) lesz a legfontosabb vizsgálandó tulajdonság.

3. fejezet

Hozzárendelt védelem azonnali helyreállítással

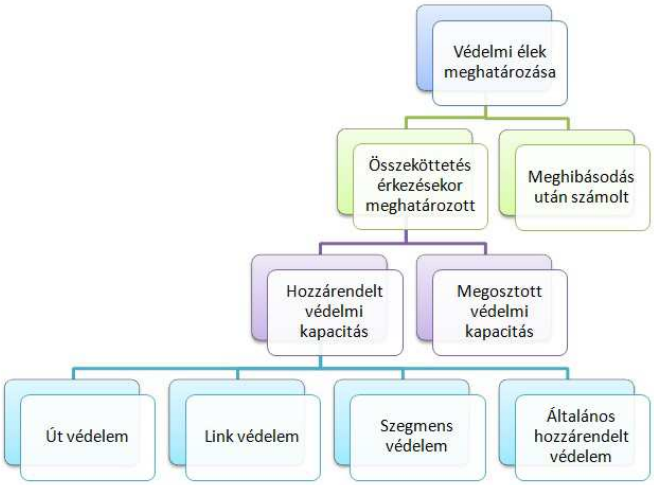
Az előző fejezetben bemutattam a hálózat gráf reprezentációját, bevezettem a rendelkezésre állás fogalmát, valamint a hibák modellezésére leggyakrabban használt módszert, vagyis a közös kockázatú csoportok módszerét (SRLG). Ebben a fejezetben a különböző hozzárendelt védelmi módokat mutatom be, amelyek az SRLG modellre épülnek, és amelyek célja az összeköttetések QoS igényének biztosítása és az azonnali helyreállítás garantálása.

3.1. A rendelkezésre állás növelése

A hálózat rendelkezésre állását kétféleképpen lehet növelni: vagy magasabb rendelkezésre állású, de egyben drágább hálózati elemeket vásárolunk, és azokat bizonyos szinten redundánssá is tesszük, vagy valamelyik védelmi mechanizmust használatával. Az első módszer inkább csak elméleti, mivel nagy anyagi ráfordítást követel meg, és ezt egyik szolgáltató sem vállalja fel. A második esetben a már meglévő hálózati infrastruktúrát használjuk fel, és az üzemi útvonalak mellé védelmi útvonalakat foglalunk. Így magasabb rendelkezésre állást érünk el, de több hálózati erőforrás felhasználása mellett.

Egy védelmi útvonal lehet előre tervezett vagy meghibásodás után (real-time) számolt, ahogy ez a 3.1. ábrán is látható. Utóbbi hosszabb helyreállítási időt igényel, de jobb erőforrás kihasználtságot eredményez, mivel csak meghibásodás esetén számolunk és foglaljuk le védelmi utat, míg az előre tervezett esetében már az igény beérkezésénél mindkét útvonalat lefoglaljuk. Optikai hálózatokban a nagy sávszélességű linkek miatt nem engedhetjük meg magunknak a lassú védelmet, ezért szinte kizárólag előre tervezett védelmi utakat használnak.

Az előre tervezett útvonal esetén megkülönböztetünk megosztott, illetve hozzárendelt védelmet. A megosztott védelem esetén egységnyi védelmi kapacitást (pl.: hullámhosszt) megoszthatunk olyan felhasználók között, akik üzemi erőforrásai függetlenek egymástól [12]. Megosztott védelemmel ugyan



3.1. ábra. Módszerek a rendelkezésre állás növelésére [15]

jobb erőforrás kihasználtságot érünk el, mint hozzárendelt védelem esetén, de ha esetleg több üzemi útvonal egyszerre esne ki, akkor nem tudjuk védeni az összeset. Például a 2.5. ábrán látható összeköttetések védelmi erőforrásai nem lennének megoszthatóak egymással, mert az $SRLG_4$ kiesése esetén mindkét összeköttetés forgalmát a védelmi élekre kellene terelni, mely versenyhelyzet kialakulásához, és az egyik összeköttetés megszakadásához vezetne. Ráadásul, a megosztás mértékétől függően az útvonalak meghatározása komplex feladat. Mindemellett a gyakorlati alkalmazása is bonyolult, mivel meghibásodás esetén újra kell konfigurálni a kapcsolókat (OXC), és ez természetesen növeli a helyreállítás időtartamát is, elrontva az azonnali helyreállítás lehetőségét. Manapság a hálózatok egyre nagyobb komplexitása miatt egyre fontosabbá válik, hogy egy védelem ellenálló és robusztus legyen. Az előbbi tulajdonság miatt a megosztott védelmek nem számítanak robusztusnak, részben ezért nem is alkalmazzák őket a gyakorlatban optikai gerinchálózatokban. A hozzárendelt védelmek viszont képesek garantálni az azonnali helyreállítást. Mivel dolgozatomban egy hozzárendelt védelmi megoldást javasoltam, ezért a következő alfejezetben részletesen ismertetem ezen eljárások tulajdonságait.

3.2. Hozzárendelt $1 + 1$ védelem lehetséges megvalósításai

A gyakorlatban a hozzárendelt védelmet implementáljak a gerinchálózatokban, mivel a szolgáltatók a védelmek terén inkább előnyben részesítik az egyszerű megoldásokat, továbbá jelenleg a kapacitás még nem számít szűk keresztmetszetnek gerinchálózatok esetén. De hamarosan ez megváltozhat, így új megoldások kerülhetnek előtérbe.

A hozzárendelt védelem az erőforrás és a felhasználó között egy-egy hozzárendelést valósít meg. Ez azt jelenti, hogy az egységnyi védelmi erőforrást kizárólag egy felhasználó használhat, így a felhasználói adatfolyam küldhető akár folyamatosan, mind az üzemi, mind a védelmi útvonalon ($1 + 1$ védelem). Az

üzemi útvonal kiesése esetén pedig elég a nyelő csomópontnak egy másik bejövő interfészre kapcsolnia. Ez a védelemi megoldás egyszerű, ellenálló és robusztus, viszont nagy a kapacitás igénye.

Az $1 + 1$ védelemnek több megvalósítása is létezik [24] (például hálózati kódolással vagy a jelek redundáns osztásával); az üzemi és védelmi út lehet él- vagy SRLG-független, azaz egy él vagy SRLG kiesése esetén legfeljebb egy útvonal esik ki. Valamint nem csak végponttól végpontig terjedő védelmet lehet alkalmazni. Ahogy ez már a [3] tanulmányban meg lett fogalmazva, ha az üzemi útvonalat linkekre, vagy átlapolódó vagy nem átlapolódó szegmensekre osztjuk, és a szegmens két végpontja között számolunk egy független útvonalat *hozzárendelt szegmens (vagy link) védelemnek* [11], vagy *részleges útvédelemnek* [36] [37] nevezzük. A részleges útvédelem esetén jobban érvényesül a lokális hibajavítás elve, és jobban összeegyeztethető az SRLG modellel is, viszont a kapacitás igénye nagyobb, mint a teljes útvédelemé.

Az előzőleg ismertetett hozzárendel védelmi módszerek esetében felmerülhet a kérdés, hogy mennyire nehéz megtalálni két csomópont között az él- vagy éppen SRLG-független optimális útvonal párt. A helyzet korántsem triviális. Nem használhatunk mohó algoritmusokat, mivel azok által könnyen csapda szituációba kerülhetünk, azaz ha az üzemi útvonalat a lehetséges legrövidebbre választjuk (mondjuk, Dijkstra algoritmus [8] használatával), olyan szituációba kerülhetünk, hogy már képtelenek vagyunk védelmi utat találni, vagyis az összeköttetésünk védelem nélkül marad. Így tehát egy olyan algoritmusra van szükség, ami már eleve útvonal párokat keres, ráadásul, ha egy mód van rá, polinom időben. A hetvenes években publikálták minimális költségű diszjunkt útpárok keresésére a Suurballe algoritmust [31], amely már képes volt megbirkózni ezzel a feladattal. További előnye, hogy ha létezik ilyen útpár, akkor azt az algoritmus garantáltan megtalálja. Fontos megjegyezni, hogy ugyan egyszeres hibák védelmére Suurballe algoritmusa polinom időben ad megoldást, a helyzet korántsem ilyen egyszerű tetszőleges többszörös hibákat is tartalmazó SRLG független utak esetén, ugyanis ekkor a feladat NP-teljessé válik [9].

Annak ellenére, hogy jelenleg a leggyakrabban használatos védelmi megoldás az optikai rétegben az $1 + 1$ hozzárendelt védelem, a hálózat és a magas megbízhatóságú összeköttetések kialakításához védendő többszörös link hibákat tartalmazó SRLG-k esetén komplexitása miatt a jövőben már csak korlátozottan alkalmazható. Ezen hátrányok kiküszöbölésére javasolták az általános hozzárendelt védelmet (GDP) [3], amely a felhasználó igényeihez rugalmasan illeszkedő, megbízható összeköttetések kiépítését támogató védelemi megoldás, továbbá megőrzi az $1 + 1$ védelem azonnali helyreállítás tulajdonságát. Erről az új védelmi megoldásról szól a következő fejezet, és ez képezi munkám vizsgálati tárgyát is.

4. fejezet

Általános hozzárendelt védelem (GDP)

Az eddigiekben szó esett a hálózati védelmekről, és megismerhettük a leggyakrabban alkalmazott $1+1$ hozzárendelt védelmi módszert is. Ebben a fejezetben az általános hozzárendelt védelmet (GDP – Generalized Dedicated Protection) mutatom be. A GDP módszer egy általános matematikai modell, amely általánosan fogalmazza meg a hozzárendelt útvonalválasztási feladatot, és annak kiválasztását egy optimalizálási feladatnak tekinti. Munkámban ennek a védelemnek a gyakorlati megvalósíthatóságát vizsgálom.

4.1. A GDP védelmi feladat megfogalmazása [3]

A hálózatban rendelkezésre álló optikai eszközök, illetve az alkalmazott technológiai megoldásoknak megfelelően a GDP védelemnek több különböző megoldása lehet, de maga a probléma egységesen definiálható. Ahogy ezt később látni is fogjuk, a különbség csak a keresendő megoldás kényszereiben rejlik. A 4.1. táblázatban az általános hozzárendelt védelem esetén alkalmazott jelölések összefoglalója látható.

Jelölje $\mathcal{I} = \{G, \mathcal{D}, \mathcal{F}\}$ a GDP feladat egy tetszőleges példányát, ahol G a hálózat gráf modelljét, $\mathcal{D}$ az összeköttetés igényeket, míg $\mathcal{F}$ a védendő SRLG-k listáját adja meg, ahogy ezt a 2. fejezetben ismertettem. Továbbá:

- Az útvonalválasztás dinamikus változata esetén (amit a dolgozatban vizsgálok) minden összeköttetés igényt, annak a beérkezési t pillanatában azonnal elvezetünk, és mindaddig fenntartjuk, amíg az összeköttetési igény fennáll.
- A hálózatot leíró irányítatlan $G = (V, E)$ gráf minden $e \in E$ éléhez tartozik egy nem negatív költség függvény ($c : E \to R^+$) , és egy a szabad kapacitást jelölő k változó ($k : E \to R^+ +$), amely a mindenkori t időnek a függvénye, mivel a beérkező igényektől és azok kiszolgálási módjától is függ a pillanatnyi szabad kapacitás mennyisége.
- A $\mathcal{D} = (s, d, b)$ összeköttetés igény tartalmazza a s forrás, illetve d nyelő csomópontokat, valamint a hozzájuk tartozó b ($b \in \mathbb{N}$)) sávszélesség igényt.

– Illetve minden hibamintához (vagyis SRLG-hez) amely a $\mathcal{F}$-ben található, konstruálunk egy segéd gráfot: $\forall f \in \mathcal{F} : G_f = (V, E_f)$, ahol az E_f úgy kapjuk meg, hogy az adott SRLG-hez, vagyis f-hez tartozó éleket eltávolítjuk az E halmazból. Továbbá feltételezzük, hogy az összes SRLG, amely $\mathcal{F}$-ben található, valóban védhető is, vagyis, hogy minden G_f gráf $s - d$ összefüggő.

Jelölje $\mathcal{X}_\mathcal{I}$ az $\mathcal{I}$ példányt kielégítő $y_\mathcal{I} = \{H, \mathcal{R}\}$ megoldások halmazát. Egy kielégítő megoldás két részből tevődik össze, a $H = (V, E)$ részgráfból és $\mathcal{R}$ konfigurációs beállításokból:

– A $H = (V, E)$ részgráf tartalmazza a megoldásban szereplő folyam értékeket, vagyis $\forall e \in E$: : $b_e \leq k_e$, ahol b_e az adott élen foglalt kapacitást, míg k_e az adott élhez tartozó szabad kapacitást jelöli. A megoldásnak az összes $f \in \mathcal{F}$ hibának ellen kell állnia, azaz mind a $H = (V, E)$ gráfban mind a $\forall H_f = (V, E_f)$ SRLG gráfban a maximális folyam értéke s és d között legalább $\geq b$.

– A GDP megoldás másik fele a robusztus konfiguráció $\mathcal{R}$. Ez megadja a különböző hálózati elemek beállításait kapcsolat-felépítéskor, mint például a kapcsoló mátrixok beállításai, vagy a jelek összevonásához, illetve a forgalomszövéshez szükséges információk.

Fontos megemlíteni, hogy a robusztus konfiguráció feltétele a kielégítő megoldásnak, tehát ha egy folyamgráfhoz nem találunk robusztus konfigurációt, akkor azt nem tekintjük megfelelő megoldásnak. Ez abból adódik, hogy a GDP célja, hogy megőrizze a hozzárendelt $1 + 1$ védelem pozitív tulajdonságait: az egyszerűséget, gyorsaságot és az azonnali helyreállíthatóságot , amely az ellenálló és robusztus megoldások ismérve.

Továbbá, minden megengedett megoldáshoz ($y_\mathcal{I} \in \mathcal{X}_\mathcal{I}$) megadjuk annak költségét ($g(y_\mathcal{I})$), amely a lefoglalt kapacitás függvénye:

$$g(y_\mathcal{I}) = \sum_{\forall e \in E} c_e \cdot \frac{b_e}{b}, \tag{4.1}$$

Ahol b_e a $H = (V, E)$ gráfban az e élhez tartozó a megoldásban használt sávszélesség, amelynek természetesen kisebbnek kell lennie, mint az adott linken lévő szabad kapacitás, azaz teljesül a $b_e \leq$ $\leq k_e$ összefüggés. A célfüggvény a költség b sávszélességére normált értéket adja meg, de a függvény a normálás nélkül is használható lenne a megoldások összehasonlítására. A kielégítő megoldást kereső GDP algoritmus, ezen költségfüggvényt minimalizálja.

A hálózatban rendelkezésre álló optikai eszközök képességei (lásd a 2.3. ábrát), illetve alkalmazott technológiai megoldásoknak megfelelően jelenleg két eltérő GDP feladatról beszélhetünk:

– Abban az esetben, ha a GDP megoldást osztatlan folyamok formájában keressük, IGDP feladatról beszélhetünk (IGDP – Integer (or non-bifurcated) GDP). Ekkor minden, a megoldásban használatos linken b kapacitást foglalunk, a hálózatban pedig a 2.3. ábrán látható (a)-(c) csomópont szerepek a megengedettek [29].

– Ha osztott folyamok formájában keressük, akkor a megoldásban használt linkeken nem kizárólag b kapacitást foglalhatunk, hanem bármilyen értéket 0 és b között. Ez a feladat önmagában nem

4.1. táblázat. Általános hozzárendelt védelem esetén alkalmazott jelölések [3]

Jelölés	Leírás
$\mathcal{I} = \{G, \mathcal{D}, \mathcal{F}\}$	az útvonalválasztási feladat egy példánya
$G = (V, E)$	a hálózat irányított vagy irányítatlan gráfmodellje V csomópont és E élhalmaz esetén
c_e	az éleken definiált költségfüggvény $e \in E$
k_e	az éleken rendelkezésre álló szabad kapacitás $e \in E$
$\mathcal{D} = (s, d, b)$	a dinamikusan érkező igény forrás- és célcsomópontja, valamint sávszélesség igénye
$\mathcal{F}$	a szolgáltató által meghatározott közös kockázatú csoportok listája, amely ellen az összeköttetést védeni kell
$G_f = (V, E_f)$	az adott $f \in \mathcal{F}$ csoportban meghibásodott élek törlésével nyert SRLG gráf
$\mathcal{X}_\mathcal{I}$	az $\mathcal{I}$ bemenetet kielégítő $y_\mathcal{I}$ megoldások halmaza
$y_\mathcal{I} = \{H, \mathcal{R}\}$	egy kielégítő megoldás tartalmazza a megfelelő részgráfot H és a csomópontok konfigurációját $\mathcal{R}$
b_e	az $e \in E$ élen foglalt kapacitás a megoldásban

biztosítja az azonnali helyreállítást, csak abban az esetben, ha a folyamok osztása mellet a hálózati kódolás is megengedett, akkor GDP-NC (GDP-NC – GDP with Network Coding) feladatról beszélhetünk [4], ekkora az összes csomóponti szerep alkalmazható, vagyis (a)-(f).

Míg az IGDP feladat megvalósításához szükséges (a)-(c) eszközök rendelkezésre állnak a jelenlegi optikai gerinchálózatokban, a lefoglalt sávszélesség igen magas lehet, annak ellenére, hogy ha létezik $1+1$ védelem az adott összeköttetésre, az IGDP módszer legalább ilyen, vagy alacsonyabb sávszélességhasználatot garantál. A GDP-NC módszer ugyan biztosítja az optimális erőforrás használatot valamennyi azonnali helyreállítást garantáló hozzárendelt (és nem csak a GDP módszerek) között, viszont a gyakorlatban nem állnak rendelkezésre a megvalósításához szükséges (d)-(f) eszközök. A dolgozatomban ezért egy új hálózati kódolást alkalmazó módszert dolgozok ki, amely közel optimális az erőforrás használatban, de ehhez olyan eszközöket használ, amelyek akár jelenleg akár a közeljövőben jelen lesznek a legtöbb optikai gerinchálózatban. De mi is a hálózati kódolás? Illetve milyen előnyei illetve hátrányai vannak? Ennek a bemutatása szánom a következő alfejezetet.

4.2. A hálózati kódolás

A hálózati kódolás – azaz network coding – egy viszonylag új kutatási terület [2], amely a forrás- és csatorna-kódolással ellentétben nem csak a forrás, hanem a hálózat közbülső csomópontjaiban is lehetővé teszi az adatok kombinálását (kódolását). Egy hálózati csomópont tehát ebben az esetben nemcsak az adatok továbbítására, átirányítására képes, hanem bizonyos műveleteket is képes végrehajtani azokon. Így a kimeneti link adatfolyama a csomópontba korábban beérkező bemenetek valamilyen matematikai függvénye lehet. Természetesen a célcsomópontnak tudnia kell visszakódolni az adatfolyamokat, hogy így visszanyerje az eredeti információkat. A hálózati kódolás az útvonalválasztáshoz képest több komoly előnnyel is rendelkezik, a teljesség igénye nélkül:

– nagyobb megvalósítható sávszélesség például többesadás esetén,

– alacsony energiafogyasztás, vagyis alacsony energia/bit arány,

– alacsony késleltetési idők [6],

– valamint a hálózat robusztusságát is komoly mértékben növelni képes [10].

Természetesen a hálózati kódolásnak nem csak előnyei vannak, egyik hátránya, hogy bizonyos komplexitást visz a rendszerbe, így nehézkesebbé válik a hálózat menedzselése, valamint a hibalokalizáció is. A munkám szempontjából viszont leginkább a robusztusság, illetve a nagyobb átviteli sávszélesség bír jelentőséggel, így csak azokat tárgyalom részletesebben.

4.2.1. A hálózati kódolás típusai

A hálózati kódolás elméletében az igazi áttörést [16] jelentette, amely egy alapvetően gráfelméleti problémát algebrai alapokra helyezett, így lehetőséget adva a feladat megoldásához az igen széles algebrai eszköztár alkalmazására. Ezek után a hálózati kódolás lényege úgy fogalmazható meg, hogy a kimeneti adatfolyam a bemenetekre érkező adatok valamilyen matematikai függvénye. A hozzárendelés alapján a hálózati kódolás lehet lineáris, illetve nem lineáris. A legtöbb esetben az optimális eredmény eléréséhez elegendő a lineáris operációk használata, de ez nem mindig igaz, így a nem lineáris hálózati kódolással is érdemes foglalkozni. Viszont annak komplexitása miatt, ahol lehet, próbálják kerülni. A lineáris kódolás egyszerűsége, átláthatósága és igen jó használhatósága miatt népszerű kutatási területnek számít. Az általam vizsgált általánosan hozzárendelt védelem estén is elegendő a lineáris hálózati kódolás használata, ennek matematikai háttere pedig röviden a következő:

Legyen $G = (V, E)$ gráf irányított körmentes, azaz DAG (ez a GDP feladat esetén a $H = (V, E')$ összeköttetés részgráfra teljesül), illetve jelölje $y(e)$ a v csomópontból kilépő, az e éleken továbbított szimbólumokat (amelyek elemei egy véges T testnek – azaz Galois Field (GF)), míg $y(e')$ a v-be az e' éleken beérkező szimbólumokat; ekkor az $y(e)$-t az alábbi módon kapjuk meg:

$$y(e) = \sum_{e'} \beta'_e(e) \cdot y(e'), \tag{4.2}$$

ahol $\beta'_e(e)$ az e élhez tartozó lokális kódoló vektor, amelynek hossza megegyezik a v csomópontba belépő e' élek számával.

Tovább általánosítva a helyzetet, legyenek az s forrás csomópontban a forrás szimbólumok a $x_1, \ldots, x_h$, akkor $\forall e \in E$-hez tartozó $y(e)$ megkapjuk kizárólag a bemeneti forrás szimbólumok lineáris kombinációjaként:

$$y(e) = \sum_{i=1}^{h} g_i(e) \cdot x_i. \tag{4.3}$$

Ekkor az e élhez tartozó lineáris kombináció vektort, azaz $g(e) = [g_1(e), \ldots, g_h(e)]$-t *globális kódoló vektornak* is szokás nevezni.

A helyreállíthatóság érdekében a d csomópont beérkező független szimbólum vektort számának legalább h-nak kell lennie. A továbbiakban tegyük fel az egyszerűség kedvéért, hogy a d nyelő csomópontba pontosan h élen érkeznek be a adatok és hogy azok lineárisan függetlenek. A $e_1, \ldots, e_h$ éleken beérkező adatokat pedig jelöljük $y(e_1), \ldots, y(e_h)$-vel , ekkor:

$$\begin{pmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{pmatrix} = \begin{pmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_h \end{pmatrix} = G_t \begin{pmatrix} x_1 \\ \vdots \\ x_h \end{pmatrix} \tag{4.4}$$

Ezek után már könnyen belátható, ha G_t *dekódolási mátrix* inverzét vesszük, és megszorozzuk a beérkező szimbólum vektort, akkor visszakapjuk az eredeti üzenet vektort. Felmerül természetesen a kérdés, hogy G_t mikor invertálható, illetve hogyan lehet elérni, hogy G_t adott valószínűséggel invertálható legyen. Ezzel el is jutunk a hálózati kódolás egy újabb osztályozásához, mivel létezik polinom időben futó determinisztikus algoritmus, amely végigmenve a hálózati csomópontokon, azokhoz hozzárendeli a lokális kódolási együttható vektorokat [14]. Ebben az esetben nem szükséges a kódolási vektorok küldése, mivel ezek előre meghatározottak, így a célcsomópont is előre ismeri azokat. Sajnos a hálózatok komplexitása és nagysága miatt, valamint a nehéz gyakorlati megvalósíthatósága miatt ez a fajta hálózati kódolás inkább csak elméleti jelentőségű.

A másik lehetőség, amely a gyakorlatban is könnyebben implementálható, az úgynevezett random network coding, azaz a random hálózati kódolás. Ennek a lényege, hogy a lokális kódolási vektorokat decentralizált módon, véletlenszerűen választjuk ki az adott csomópontokban. Természetesen nem teljesen véletlenszerűen, hanem egy adott véges testből (jelöljünk azt T-vel), amelynek arányában lényegesen nagyobbnak kell lenni a hálózat nagyságánál, ahogy ezt Ho [13] és Sanders [27] egymástól függetlenül cikkeikben már bemutatták és részletesen tárgyalták. Ekkor természetesen egy bizonyos valószínűséggel lineárisan függők lesznek a vektorok, így azok a Gauss elimináció során kiesnek. Ez a valószínűség függ

$|T|$ illetve $|E|$ arányától, ha például $|T| = 2^{16}$ illetve $|E| = 2^8$, akkor ez a valószínűség $1 - |E|/|T| \approx$ ≈ 0.996, ami már elég jó valószínűséggel biztosítja a dekódolhatóságot [6]. Ezért megállapítható, hogy a random hálózati kódolás igen jól működik annak ellenére, hogy nem 1 valószínűséggel dekódolható a küldött adat a nyelő csomópontban. Viszont ekkor a kódolási vektorokat az üzenettel együtt el kell küldeni (akár a fejlécben, akár a payloadban), tehát ez egy bizonyos overheadet képez. Másrészről bizonyos szempontból ez előnyösnek is számit, mivel a célcsomópontnak nem kell birtokában lennie semmilyen extra információnak, vagyis a kódolási vektort az üzenettel együtt kapja meg. Továbbá, ez a hálózat rendelkezésre állásának szempontjából is kedvező, mivel ekkor a dekódolási mátrix is dinamikusnak számít, így a link-, illetve csomóponti hibák nem okoznak problémát a dekódolás során.

Dolgozatom, és a gyakorlati megvalósíthatóság szempontjából a T testméret az, amely meghatározza a módszer komplexitását. Vagyis minél kisebb a T testméret, annál könnyebb eszközökben megvalósítani, mivel ekkor elegendő egyszerűbb algebrai műveletek elvégzése. Kisebb test esetén az adatok visszanyerése is gyorsabb, mely a gyakorlati megvalósíthatóság szempontjából igen fontos. Ezért munkámban különös figyelmet fordítottam arra az estre, amikor a felhasználói adatok osztását csak két részre engedjük meg, mivel ekkor a [26]-ben ismertetett kódolás segítségével a robosztus hálózati kódolás megoldható $|T| = 2$ mellett, azaz a bejövő adatokon egyszerűen XOR műveletet kell végrehajtanunk, mely eszközök akár már a közeljövőben is rendelkezésre állhatnak az optikai gerinchálózatokban [19].

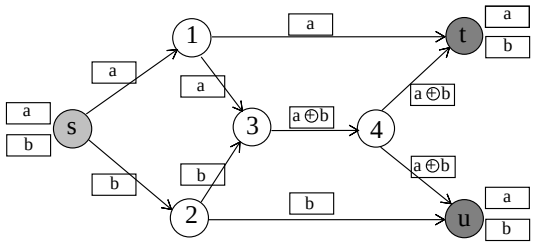
4.2.2. A hálozati kodolás előnyei

A hálózati kódolás előnyei közül itt kettőt veszünk jobban szemügyre. Először egy egyszerű példán keresztül mutatom be, hogy hogyan növelhető az átviteli sávszélesség a hálózati kódolás segítségével. Azután pedig a hálózat robusztusságának növelését is bemutatom egy példán keresztül.

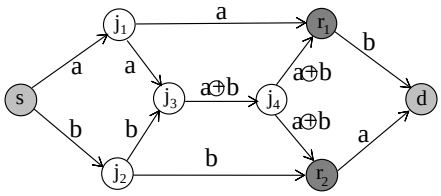
Átviteli sávszélesség növelés

A hálózati kódolás legnagyobb előnyeként az átviteli sávszélesség növelését szokták említeni, amely leginkább többesadás esetben szembeötlő. Vizsgáljuk meg a 4.1. ábrán látható esetet, ahol mindegyik hálózati link kapacitása egy, és az s forráscsomópontból szeretnénk az a és b forrás szimbólumokat mind t-be, mind pedig u-ba elküldeni. Láthatjuk, hogy mind t-be, mind u-ba létezik két független útpár, azaz $s-1-t$ illetve $s-2-3-4-t$, valamint $s-2-u$ és $s-1-3-4-t$. Viszont hálózati kódolás nélkül nem tudjuk eljuttatni mindkét nyelő csomópontba egyszerre a-t és b-t, mivel ekkor a 3 és 4 csomópont között egyszerre kellene áthaladnia mind a-nak, mind b-nek, és mivel ennek a linknek a kapacitása egy, ez nem lehetséges.

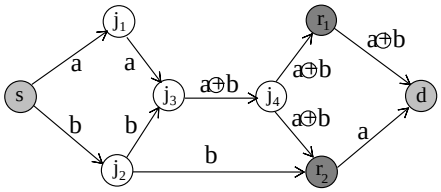
Hálózati kódolás használata esetén viszont ez nem jelent problémát, mivel – ahogy az ábrán is látszik – az $s-1-t$ útvonalon küldhetjük az a-t, a $s-2-u$ útvonalon pedig a b-t, a 3-as csomópontba viszont mind a-t, mind b-t el kell juttatnunk, ahol a csomópont veszi a bejövő szimbólumok kizáró VAGY (azaz XOR) értékét, majd így küldi tovább. A 4-es csomópont szétosztja az adatfolyamot, így mind u, mind t



4.1. ábra. Az úgynevezett „pillangó hálózat” illusztrációja



(a) Az összeköttetés hibátlan állapotában a jel mindkét, a és b fele is eléri a célcsomópontot.



(b) A GDP megoldás ellenálló és robosztus valamennyi, pl.: $f = (j_1, r_1)$ SRLG hiba esetén.

4.2. ábra. Egy lehetséges LP megoldás $\{H = (V, E), \mathcal{R}\} \in \mathcal{X}_{\mathcal{I}}$ az $\mathcal{I} = \{G = (V, E), \mathcal{D} = \{s, d, 2\}, \mathcal{F} = \{(j_1, r_1), (j_2, r_2), (j_3, j_4)\}\}$ példánynak, mely tartalmazza a hálózati kódolásból ismert pillangó gráfot r_1 és r_2 vevővel. Minden linken $b_e = 1$, a jel két felét a és b jelöli [4].

csomópontok egy egyszerű XOR művelet után rendelkeznek mindkét forrás szimbólummal, azaz a-val és b-vel. A 4.2.1. fejezetben ismertetett jelölésekkel a jelen példában egy $|T| = 2$ véges test elegendő volt a dekódoláshoz valamennyi nyelő csomópontban. Már ez az egyszerű példa jól mutatja a hálózati kódolásban rejlő lehetőségeket.

Robusztusság és adaptivitás

Dolgozatom szempontjából másik fontos előnye a hálózati kódolásnak a védelem robusztusságának biztosítása osztott folyamok esetén. A 4.2. ábrán látható egy ellenálló megoldás a GDP-NC feladatra, azaz egy olyan megoldás, amely a $\mathcal{F}$ szereplő összes hiba esetén képes védeni a hálózatot, vagyis elvezetni az adott igényeket. De hálózati kódolás nélkül ez a megoldás nem robusztus, mivel ha például a jel b felét konfigurálnánk a (j_3, j_4) linkre, akkor az összeköttetés a (j_1, r_1) link hibája esetén nem tartható fent köztes csomópontban történő kapcsolás nélkül. Vagyis szükség van vezérlő síkon küldött jelzésre és kapcsolásra a j_3 csomópontban ahhoz, hogy az összeköttetés helyreálljon. Másrészt, ha a jel a felét konfigurálnánk a (j_3, j_4) linkre, akkor pedig (j_2, r_2) link kiesése esetén merülne fel ugyanaz a probléma. Viszont hálózati kódolás alkalmazásával egy robusztus védelmi megoldás is kivitelezhető ugyanebben a hálózatban, ahogy ez a 4.2. ábrán látható. Ennek a megoldásnak a polinom időben történő megtalálására

például a [13] ismertetett random kódoló algoritmus segítségével lehetünk képesek. Pontosabban ez az algoritmus egy ellenálló, osztott folyamú megoldáshoz rendel polinom idő alatt egy robusztus hálózat-kódolási megoldást, ezzel biztosítva a GDP-NC védelmi módszernek az azonnali helyreállítást.

5. fejezet

A GDP gyakorlati megvalósíthatósága

Eddig bemutattam a hálózati védelemhez kapcsolódó legfontosabb alapfogalmakat, a hálózatban használatos technológiákat, illetve azok reprezentációs modelljét, a hibák kialakulásának legfontosabb okait, valamint az azok egymástól való függésének modellezésére leggyakrabban használt módszert, az úgynevezett közös kockázatú csoportokat (SRLG). A 3. fejezet fejezetben összefoglaltam a különböző védelmi mechanizmusokat, részletesebben kitértem a gyakorlatban leggyakrabban alkalmazott $1+1$ hozzárendelt védelmi megoldásra, tárgyaltam annak előnyeit, illetve hátrányait, megindokoltam a jövőben miért jelenthet gondot annak implementálása. Azután pedig ismertettem a [3] által javasolt általános hozzárendelt védelmet (GDP): egy olyan új matematikai modellt, amely általánosan fogalmazza meg a hozzárendelt útvonalválasztási feladatot. Ebben a fejezetben az eddig bemutatott különböző általános hozzárendelt védelmi mechanizmusok megvalósításával foglalkozom, vagyis azok algoritmusait írom le és vizsgálom meg azokat gyakorlati megvalósítás szempontjából, azután pedig javaslatot teszek egy új GDP módszerre, amely jobban szem előtt tartja a gyakorlati megvalósíthatóságot.

5.1. Technológiai és matematikai háttér

Ebben az alfejezetben röviden összefoglalom azon elveket és módszereket, melyek az optikai gerinchálózat technológiájából, illetve működési elveiből adódnak, illetve hatékonynak bizonyultak a 4. fejezetben bemutatott GDP matematikai modell megoldására.

Dolgozatomban feltételeztem, hogy mindegyik csomópont képes az úgynevezett teljes hullámhossz konverzióra (bármely bejövő hullámhosszt képes átkapcsolni tetszőleges link tetszőleges hullámhosszára) és az adatszövésre, vagyis groomingra. A két módszer egyidejű jelenlétének köszönhetően az összeköttetéseket folyamokként lehet kezelni az egyes GDP algoritmusokban [3]. Megjegyzendő, hogy a létező és az általam javasolt GDP algoritmus is képes kezelni a hullámhossz folytonosságot (az összeköttetés által használt valamennyi linken azonos hullámhosszt kell használni), egy lényegesen bonyolultabb segédgráf segítségével. Az egyszerűség kedvéért csak a folyam alapú megoldásokat ismertetem, de hasonló módon végiggondolhatóak a módszerek kevésbé kifinomult alkalmazási környezetekben is.

Az iménti modellben a védelmi feladatok megoldására kiválóan használható a lineáris programozás. A lineáris programozási (LP – Linear Programming) feladatot már 1939-ben Kantorovics szovjet matematikus tárgyalta, de akkor még nem ismerték fel annak fontosságát. Ezután 1947-ben Dantzig amerikai matematikus vette újra elő a témát és dolgozta ki az úgynevezett szimplex módszert. Ezek utána az operációkutatás és a matematikai programozás rohamos fejlődésnek indult. Azóta is rengeteg probléma megoldására használják a lineáris programozást, amely matematikai módszerrel segíti a legjobb kimenet megtalálását [7]. A lineáris programozás egy lineáris célfüggvényt maximalizál/minimalizál, figyelembe véve adott lineáris egyenlőtlenségi relációkat, illetve határfeltételeket. A feladat kanonikus alakja a következő:

- maximalizálandó vagy minimalizálandó lineáris függvény: $\underline{c}^T \underline{x}$
- lineáris egyenlőtlenségeket: $\underline{\underline{A}}\underline{x} \leq \underline{b}$
- korátok: $\underline{x} \geq 0$
- Ahol $\underline{x}$ a meghatározandó változókat tartalmazza, $\underline{c}$ és $\underline{b}$ ismert együttható vektorok ($\underline{c}^T$ a $\underline{c}$ vektor transzponáltja), $\underline{\underline{A}}$ pedig egy ismert együttható mátrix.

Amennyiben a GDP védelmi feladatot a fent említett módon tudjuk megfogalmazni, akkor az LP felírás megoldásával egy E&R megengedett megoldást kapunk, mely biztosítja az összeköttetés azonnali helyreállítását. Ráadásul a megoldást polinom időben kapjuk meg, amennyiben nem alkalmazunk egészértékűségi kényszereket, azaz nem követeljük meg az x változóktól, hogy csupán egész értékeket vehetnek fel. Ha alkalmazunk ilyeneket, akkor már ILP (ILP – Interger Linear Programming) feladatról beszélünk, amelynek futási ideje már nem polinomiális.

5.2. Osztatlan IGDP megoldás tulajdonságai

Az általános GDP feladatot a 4.1. fejezetben részletesen bemutattam. Ahogy már láthattuk, mindkét GDP feladatnak három bemeneti paramétere van: $G = (V, E)$ a hálózat gráf reprezentációja, $\mathcal{D}$, amely tartalmazza a különböző összeköttetés igényeket, valamint $\mathcal{F}$ a védendő SRLG-k listáját. A (4.1) egyenlet pedig a minimalizálandó célfüggvény, azaz a cél egy minimális sávszélességhasználattal rendelkező, megengedett, vagyis azonnali helyreállítást garantáló $y_{\mathcal{I}} = \{H, \mathcal{R}\}$ megoldás találása. $y_{\mathcal{I}}$ két részből tevődik össze: egy $H = (V, E)$ részgráfból, amely tartalmazza a lefoglalt kapacitásokat, vagyis folyam értékeket, és az $\mathcal{R}$ konfigurációs beállításból. Megjegyzendő, hogy osztatlan folyamok esetén $H = (V, E)$-ből az $\mathcal{R}$ konfiguráció azonnal következik, azaz IGDP esetében elegendő egy ellenálló részgráf találására koncentrálni. Ilyen részgráf megtalálásához pedig az osztatlan IGDP feladat estén $G = (V, E)$ gráfból kitörölhetjük a kevés szabad kapacitással rendelkező éleket, vagyis ahol $k_e < b$, hiszen ezek nem alkalmasak az összeköttetés továbbítására. Így az általánosság elvesztése nélkül elég a maradék éleken $\mathcal{D} = (s, d, 1)$ igény megoldását vizsgálnunk. Ha mindegyik élre igaz, hogy $k_e \geq b$, akkor, ha létezik az

$\mathcal{D} = (s, d,1)$ igénynek megfelelő, s és d csomópontok közötti megengedett megoldása, akkor biztosan el tudjuk vezetni a teljes b felhasználói adatot osztás nélkül (egyszerűen 1 helyett b sávszélességet foglalva a megoldásban szereplő éleken).

Mivel az IGDP Karp redukcióval visszavezethető a Steiner-erdő problémára, amely bizonyítottan NP-teljes, így az IGDP is biztosan az. Azaz nem remélhetünk az IGDP feladatra egy általánosságban gyors algoritmust, amely a dinamikus útvonalválasztás szempontjából elengedhetetlen lenne. A Steiner-erdő problémában adott egy $G = (V, E)$ irányítatlan gráf, amelynek mindegyik éléhez tartozik egy $c : E \to \mathbb{R}^+$ költségfüggvény, valamint egy diszjunkt r részhalmaz: $S_i \subseteq V$. A kérdés az, hogy létezik-e olyan $F \leq k$ költségű fa, amelyre teljesül, hogy mindegyik i és $u, v \in S_i$ esetén, F tartalmaz útvonalat u és v között. Az IGDP feladat NP-teljességét ezek alapján a következőképpen láthatjuk be:

1. Tétel. *Annak az eldöntése, hogy létezik-e az IGDP feladatra $\leq k$ költségű megoldás, NP-teljes.*

Az állítás bizonyítása [4]-ben található. Röviden a bizonyítás arra épül, hogy a Steiner-erdő feladat egy tetszőleges példányát visszavezetjük az IGDP feladat egy $\mathcal{D} = (s, d,1)$ összeköttetés igénnyel rendelkező példányára. A polinom idejű transzformáció pedig a következőképpen néz ki: a már meglévő $G = (V, E)$ gráfhoz hozzáadjuk s és t csomópontot a következő élekkel:l $E^+ = \{(s, s_i), (t_i, t)\}$, $i = 1,2, \dots r$. Ezen élek költsége legyen nulla, vagyis $\forall e \in E^+ : c_e = 0$, továbbá minden él kapacitása legyen egységnyi ($\forall e \in E : k_e = 1$). Mindegyik forrás-cél csomópont számára definiálunk egy $f_j \in \mathcal{F} : i = = 1,2, \dots, r, i \neq j : \{(s, s_i), (t_i, t)\}$ SRLG-t, amely eleme $\mathcal{F}$-nek. Így már meg is van a IGDP probléma bemenete, vagyis $\mathcal{I} = \{G = (V, E \cup E^+), \mathcal{D} = \{s, t,1\}, \mathcal{F}\}$.

A két probléma költsége megegyezik, mivel a Steiner fában akkor létezik az összes s_i és t_i párt összekötő $\leq k$ költségű megoldás, ha létezik $\leq k$ költségű IGDP megoldás s és d között. Így tehát az IGDP is NP-teljes. A fenti tétel következtében alkalmazhatjuk az ilyenkor általánosan bevetett módszert, azaz a védelmi problémát ILP feladatként írjuk fel, és ily módon keressük az optimális megengedett megoldást [4]. Mivel $b = 1$, a minimalizálandó függvény a következőre módosul:

$$\min \sum_{e \in E} c_e \cdot b_e.$$

A következő feltételek mellett (a felírásban az irányítatlan éleket kétirányú irányított élekkel helyettesítjük):

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } 0 \leq b_{e,f} \leq 1, \tag{5.1}$$

$$\forall f \in \mathcal{F}, \forall i \in V\text{: } \sum_{\forall (i,j) \in E_f} b_{(i,j),f} - \sum_{\forall (j,i) \in E_f} b_{(j,i),f} = \begin{cases} -1 & , \quad \text{ha } i = s \\ 1 & , \quad \text{ha } i = d \\ 0 & , \quad \text{egyébként} \end{cases}, \tag{5.2}$$

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } b_{e,f} \leq b_e, \tag{5.3}$$

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } b_{e,f} \text{ egészértékű változók.} \tag{5.4}$$

Az (5.1) egyenlet segítségével megadjuk, hogy a $b_{e,f}$ változó nulla és egy közötti értéket vehessen fel. Az (5.2) összefüggés a folyam megmaradást írja le az összes SRLG segéd gráfra vonatkozóan, amely szerint a forrás és nyelő csomópont kivételével minden csomópont esetén igaz, hogy a bejövő folyam mennyisége egyenlő a kimenő folyam mennyiségével. Az (5.3) kényszer a megoldást jelentő b_e értékét állítja be. Az (5.4) egyelnet megadja, hogy minden $b_{e,f}$ értéknek egész értékűnek kell lennie, vagyis a (5.1)-mal kiegészülve ez azt jelenti, hogy $b_{e,f}$ vagy nulla, vagy egy értéket vehet fel. Tehát vagy használunk egy élt a megoldásban, vagy sem. Így megkapjuk az adott igény IGDP elvezetésének egy megengedett osztatlan megoldását.

Az ILP felírás elkerüli a korábban említett csapda szituációt még abban az estben is, ha egy előre adott üzemi úthoz kell védelmi megoldást számolnunk. Ugyanis könnyen belátható, hogy az üzemi út megválasztása nem befolyásolja $\mathcal{I}$ feladat esetleges megoldhatóságát, vagyis ha a $\mathcal{W}_1$ útvonallal létezett kielégítő megoldás, akkor $\mathcal{W}_2$-vel is léteznie kell.

Összegezve tehát, a $\mathcal{D} = (s, d, b)$ által megadott igények elvezetésének problémája visszavezethető mindegyik igény $\mathcal{D} = (s, d,1)$ példányának a megoldására. Az aktuális megoldás pedig az SRLG részgráfokban talált folyamok összevonásaként tevődik össze. Továbbá, mivel a feladat maga az $s - d$ útvonal megtalálása, ezért az IGDP megoldás mindig ellenálló és robusztus is egyben.

5.3. Osztott GDP-NC megoldás tulajdonságai

Ahogy arról már korábban is szó volt, ha a hálózatban a folyamok osztását megengedjük, akkor GDP-NC-ről beszélhetünk. Ebben az alfejezetben a probléma komplexitásáról, illetve azok megvalósítási algoritmusáról lesz szó.

Ahogy a GDP-NC feladat ismertetésekor a 4. fejezetben is megjegyeztem, osztott folyamokkal megvalósított GDP feladat csak abban az esetben ad ellenálló megoldást, ha hálózati kódolást is alkalmazhatunk. Ellenkező esetben a feladat NP-teljes marad, és csak egy igen bonyolult algoritmussal tudnánk a megoldást megtalálni [3]. A GDP-NC ezzel szemben egy olyan védelem, amelyben nemcsak a folyamok osztása lehetséges, hanem hálózati kódolás használata is megengedett. A GDP-NC az IGDP-hez képest több előnnyel rendelkezik. Legfontosabb előnye a polinom idejű futás, ami alkalmassá teszi alkalmazhatóságát dinamikus útvonalválasztási feladatok elvégzésre, valamint lényegesen kisebb lefoglalt kapacitás mellett képes az azonnali helyreállítást garantálni az IGDP-hez képest. Ehhez azonban a hálózat csomóponti képességeket bővíteni kell kódolási képességgel (az (f) szerep a 2.3. ábrán), valamint ez a megoldás egy bizonyos extra komplexitással jár. Az IGDP feladattal ellentétben a GDP-NC feladat esetén az ellenálló $H = (V, E)$ részgráf nem garantálja a robusztusságot, ezért második lépésben egy robosztus konfigurációra is szükség van, melyhez hálózati kódolást alkalmazhatunk [3]:

1. először találnunk kell egy olyan $H = (V, E)$ gráfot, amely ellenálló az $\mathcal{F}$-ben található összes hibára,

2. azután pedig erre a $H = (V, E)$ gráfra illeszkedő algebrai operációk összességét $\mathcal{R}$-t kell megtalálni, amelyeknek a csomópontokban való elvégzése esetén az összeköttetés robusztus az $\mathcal{F}$-ben található összes hibára.

Az IGDP feladattal ellentétben – ahol csak ILP volt használható – az $H = (V, E)$ ellenálló gráf lineáris program megoldásával található meg, melynek bemenete az előzőekhez hasonlóan $I = \{G = (V, E), \mathcal{D} = (s, d, b), \mathcal{F}\}$, illetve az irányítatlan éleket ebben az esetben is mindkét irányban irányított élekre cseréljük. A minimalizálandó függvény továbbra is

$$g(y_{\mathcal{I}}) = \sum_{\forall e \in E} c_e \cdot \frac{b_e}{b},$$

a következő feltételek mellett:

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } 0 \leq b_{e,f} \leq \min\{b, k_e\}, \tag{5.5}$$

$$\forall f \in \mathcal{F}, \forall i \in V\text{: } \sum_{\forall (i,j) \in E_f} b_{(i,j),f} - \sum_{\forall (j,i) \in E_f} b_{(j,i),f} = \begin{cases} -b & , \quad \text{ha } i = s \\ b & , \quad \text{ha } i = d \\ 0 & , \quad \text{egyébként} \end{cases}, \tag{5.6}$$

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } b_{e,f} \leq b_e. \tag{5.7}$$

Ellentétben az IGDP-vel, itt az egész b igényt próbáljuk elvezetni, de természetesen az adott élen lefoglalt kapacitás nem lehet nagyobb az azon lévő szabad k_e szabad kapacitásnál. Amikor megvan a hibákkal szemben ellenálló $H = (V, E)$ gráf, akkor következik a második részfeladat, azaz egy robusztus konfiguráció ($\mathcal{R}$) megkeresése. Erre a [13] cikkben megfogalmazott algoritmus segítségével polinomiális időben képesek vagyunk. Ennek lényege, hogy a $\mathcal{C} = \{(s, d)\}$ összeköttetés sávszélessége semelyik $f \in \in \mathcal{F}$ SRLG hiba esetén nem csökken b alá, ezért teljesíti a robusztus kód létezésének feltételét.

A második részfeladat futási ideje lényegesen rövidebb, így elhanyagolható az ellenálló gráf megtalálásának idejéhez képest. Az egész algoritmus futási idejéről pedig elmondható, hogy polinom idejű, hiszen mindkét feladat polinom időben megvalósítható.

5.4. A GDP feladatok összehasonlítása

Ebben az alfejezetben elemzem az IGDP és GDP-NC feladatokat, azok megoldásainak előnyeit, hátrányait, illetve gyakorlati megvalósíthatóságát. Mindkét védelem megoldási algoritmusa – az $1 + 1$ védelemmel ellentétben – képes az összes $f \in \mathcal{F}$ SRLG hiba védelmére, ami igen fontos szolgáltatói szempontból, hiszen így garantálható egy adott QoS szint teljesítése, ezáltal elkerülhetőek a felhasználóknak fizetendő kártérítések. Továbbá, mivel magasabb szolgáltatás-minőség nyújtására képesek, ezáltal akár nagyobb profitra is szert tehetnek.

5.4.1. Az IGDP megvalósíthatósági problémái

– Csak osztatlan folyamok esetére ad optimális megoldást.

– Nem szükséges a csomóponti szerepek felülvizsgálata, vagyis csak az (a)-(c) szerepek a megengedettek, így a szolgáltatónak nem kell a hálózati csomópontok upgradelése vagy cseréje miatt extra kiadással számolni.

– A jelenlegi technológia szint mellett is ténylegesen kivitelezhető, vagyis minden adott a gyakorlati megvalósításhoz.

– Az algoritmus futási ideje nem polinomiális.

5.4.2. A GDP-NC megvalósíthatósági problémái

– Optimális megoldás osztott folyamok estén: ugyanazt a védelmi feladatot jelentősen kisebb kapacitásfoglalással oldja meg, mint az IGDP.

– A hálózati csomópontoknak képesnek kell lenniük a 2.3. ábrán szereplő összes csomóponti szerep betöltésére, vagyis (a)-(f) szerepek a megengedettek. Ezen képességek megléte befektetést követel meg a szolgáltató részéről.

– Az algoritmus futási ideje az eddigiekkel ellentétben polinom idejű, így akár dinamikus útvonalválasztási feladatok megoldására is alkalmazható.

– Viszont ez a megoldás a gyakorlatban nem valósítható meg, mivel a tetszőleges folyamosztás nem kivitelezhető a mai gerinchálózatokban. Egyrészt erre hardveresen nincs lehetőség, másrészt pedig a hálózat komplexitása olyan mértékben nőne, hogy az már gondot jelenthetne a zökkenőmentes működésben.

5.5. Javasolt megvalósítható védelemi módszer: GDP-NC^{ILP}

A dolgozatomban egy olyan új GDP megoldást javasolok, mely az IGDP és GDP-NC módszer előnyeit ötvözi, vagyis az általam javasolt módszer megengedi a felhasználói adatok osztását, de mindvégig szem előtt tartva annak gyakorlati megvalósíthatóságát. A legújabb hálózati kódolás eredményeket felhasználva [26] külön hangsúlyt fektetek annak az esetnek a vizsgálatára, amikor a felhasználói adat pontosan két részre osztható. A javasolt algoritmusnak (ILP) a futási ideje ugyan nem polinom idejű, de a kapacitásfoglalása jól közelíti a GDP-NC kapacitásfoglalását, és – ami a dolgozatom célja – a gyakorlatban is megvalósítható.

Az általam javasolt algoritmus esetén a GDP-NC módszerhez hasonlóan $\mathcal{D} = (s, d, b)$ igényeket vizsgáltam. Az eddigi bemeneti paraméterek mellett szükséges a maximális megengedett osztásszám

megadása, melyet div_{max}-szal jelöltem. Ebben a paraméterben tudjuk beállítani, hogy a hálózati eszközök maximálisan hány részre tudják osztani a felhasználói adatfolyamot, azaz mi az a megoldás, amely még a gyakorlatban kivitelezhető. A szimulációkban maximálisan 4 felé osztást engedtem ($div_{max} = = 4$), ekkor a minimális költségű megengedett GDP-NC^{ILP} megoldást kereső algoritmus a folyamot 1, 2, 3, vagy 4 részre oszthatja. Azaz a lenti ILP felírásban az elvezetendő folyam (div belső változó) ezeket az értékeket veszi fel. A javasolt algoritmus lényege, hogy iteratívan minden $1 \leq div \leq div_{max}$ osztáshoz meghatározzuk az optimális, ellenálló $H_{div} = (V, E)$ gráfot, és ezek közül kiválasztja azt a megoldást és osztás számot, amely a legkisebb erőforrás-foglalást eredményezi. Az egyes div értékek esetén a $H_{div} = (V, E)$ megengedett részgráf találására az alábbi módszert alkalmaztam. A bemenet $I = \{G = (V, E), \mathcal{D} = (s, d, div \cdot b), \mathcal{F}\}$, a kimenet $H_{div} = (V, E)$. Az irányítatlan éleket ebben az esetben is mindkét irányba irányított élekre cseréljük az ILP felírásban. A minimalizálandó célfüggvény

$$g(y_{\mathcal{I}}) = \sum_{\forall e \in E} c_e \cdot \frac{b_e}{b \cdot div}, \tag{5.8}$$

a következő feltételek mellett:

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } 0 \leq b_{e,f} \leq div \cdot \min\{b, k_e\}, \tag{5.9}$$

$$\forall f \in \mathcal{F}, \forall i \in V\text{: } \sum_{\forall (i,j) \in E_f} b_{(i,j),f} - \sum_{\forall (j,i) \in E_f} b_{(j,i),f} = \begin{cases} -div & , \text{ ha } i = s \\ div & , \text{ ha } i = d \\ 0 & , \text{ egyébként} \end{cases}, \tag{5.10}$$

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } b_{e,f} \leq b_e, \tag{5.11}$$

$$\forall f \in \mathcal{F},\ \forall e \in E\text{: } b_{e,f} \text{ egészértékű változók.} \tag{5.12}$$

Tehát a folyam helyén a div változó értéke szerepel, amely megadja az osztás értéket, továbbá a (5.12) megadja, hogy a lefoglalt kapacitásértékek csak egész értékűek lehetnek, így kimentként ténylegesen a div változóhoz tartozó optimális ellenálló $H_{div} = (V, E)$ gráfot kapjuk. Megjegyzendő, hogy az eredeti $\mathcal{D} = (s, d, b)$ igény megoldását a $H_{div} = (V, E)$ megoldásból a linkeken foglalandó kapacitás div-vel való osztásával kaphatjuk meg ($\forall e \in E : b_e := b_e/div$). A (5.8) egyenlet szerinti minimális értéket biztosító osztást jelölje div_{OPT} ($1 \leq div_{OPT} \leq div_{max}$).

Ahogy azt láthattuk már a GDP-NC estén is, amikor már rendelkezésünkre áll az ellenálló $H_{div_{OPT}} = = (V, E)$ gráf, ekkor második lépésként szükséges a robusztus kódolás hozzárendelése. Ez a GDP-NC^{ILP} esetén sincs másképp, hiszen az osztott folyamok miatt általánosságban itt sem garantálható az azonnali helyreállítást biztosító E&R megoldás létezése. Erre pedig a 4.2.1. fejezetben már említett kódoló módszerek tökéletesen megfelelnek, illetve a $div_{max} = 2$ speciális esetre a $GF(2)$ kódokat alkalmazó, azaz egyszerű XOR műveletekkel megvalósítható kódok is alkalmasak [26].

Fontos megjegyezni, hogy a fenti ILP felírást nem elegendő div_{max}-ra megoldani, hiszen ekkor egy olyan megoldást kapunk, ami a folyamot pontosan div_{max} részre osztja, nekünk viszont egy olyan azonnali helyreállítást garantáló módszerre volt szükségünk, mely a folyamot legfeljebb div_{max} részre oszthatja.

6. fejezet

Szimulációs eredmények

Ebben a fejezetben szimulációk segítségével összehasonlítom az általam javasolt módszert az irodalomban korábban mutatott azonnali helyreállítást biztosító eljárásokkal. A módszeremet C++ nyelven implementáltam a LEMON [1] osztálykönyvtár felhasználásával. A futási időket Linux alatt egy 1 GHz-es CPU-val és egy 2 MB memóriával rendelkező gépen mértem.

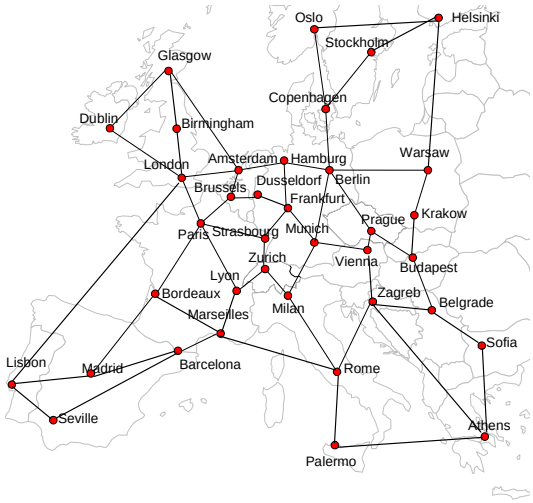
6.1. Bemeneti paraméterek

A szimulációk során elsősorban a módszerek átlagos lefoglalt kapacitását vizsgáltam a hálózat sűrűségének, méretének és az SRLG listában található hibák számának függvényében. Ehhez három fajta hálózattípust vizsgáltam, random generált ritka és sűrű hálózatokat, valamint egy létező optikai topológiát, a 37 csomópontú európai optikai gerinchálózatot (lásd a 6.1. ábrát).

A ritka és sűrű hálózatok estében 10, 15, 20, 25, 30 és 35 csomópontos, a LEMON beépített random gráfgenerátorával készített gráfokra futtattam a szimulációkat. A módszerek korrekt összehasonlítása érdekében az összes élköltséget egynek választottam, valamint minden egyes beérkező igény sávszélesség igénye (b) szintén egy. Továbbá az élek kapacitását kellően nagyra választottam annak érdekében, hogy elkerüljem az igények blokkolását, mivel a különböző módszerek lefoglalt kapacitását vizsgáltam.

Az alkalmazott forgalom generátorban különböző tartási idejű bejövő igények Poisson-folyamat szerint érkeznek, melyhez az igények közötti átlagosan eltelt időt 5 időegységnek választottam. A szimulációk során több, egyenként 150 dinamikusan beérkező igényt tartalmazó forgalmi fájlt vizsgáltam, a forgalom intenzitását 20 Erlang állítottam.

Végül a hibák modellezéséhez egy SRLG sűrűséget is definiálok, mégpedig a következőképpen: Mindegyik SRLG tartalmazza az összes egyszeres hibát, valamint a szomszédos kétszeres hibák valamennyi százalékát, jelöljük ezt p-vel. Tehát ha $p = 0$, akkor az SRLG csak egyszeres hibákat tartalmaz. Ezen kívül még három SRLG esetet vizsgáltam, mégpedig, amikor p a tíz, ötven és kilencven értékeket veszi fel. Ezekre a továbbiakban alacsony ($p = 10$), közepes ($p = 50$) és magas ($p = 90$) SRLG sűrűségként fogok hivatkozni.

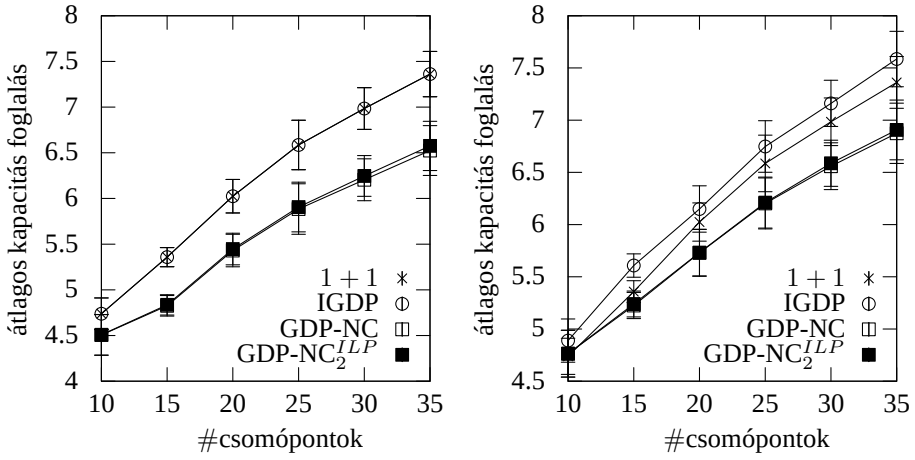


6.1. ábra. 37 pontos európai gerinchálózat [18]

Munkámban az algoritmusokat több szempont alapján is értékelem: a sávszélesség igény mellett mérem azok futási idejét is, illetve az optimális erőforglalás-foglaláshoz szükséges osztás számot is vizsgálom. Elsősorban a GDP-NC$_2^{ILP}$ teljesítményére koncentrálok, vagyis arra az esetre, amikor a felhasználói adatfolyam maximum két részre osztható és ezáltal gyakorlati szempontból a legfontosabb módszer. A teljesítményét összehasonlítom az IGDP-vel, a GDP-NC-vel és a $1 + 1$ védelemmel, mind pedig a további osztást megengedő GDP-NC$_3^{ILP}$ és GDP-NC$_4^{ILP}$ módszerekkel. Mindegyik szimulációt több azonos méretű és sűrűségű hálózaton futtattam, különböző forgalmi igények mellett, az ábrákon pedig ezen futtatások átlaga látható, kiegészítve az eredmények 95%-os konfidencia intervallumával.

6.2. Referencia algoritmus

Referencia algoritmusként a hozzárendelt útvonal védelmet (DPP – Dedicated-Path Protection) választottam, amelyre $1 + 1$ védelemként hivatkozom, mivel jelenleg ez a gyakorlatban leginkább elterjedt védelmi mechanizmus. Ennek megvalósítására a Suurbelle algoritmust használtam, amely egy olyan algoritmus, mely irányított nem negatív élű gráfban megkeresi a legkisebb költségű él-független útpárt. Ennek részletei a [31]-ben megtalálhatók. Ebben az esetben a többszörös hibákat tartalmazó SRLG-ket nem vettük figyelembe, tehát ekkor az útpár nem feltétlenül SRLG független. Így ez a megoldás magában **nem nyújt megfelelő védelmet a különböző $\mathcal{F}$-ben felsorolt hibák esetén**, de referenciának ez a módszer volt a legszerencsésebb. Fontolóra vettem egy kétlépcsős algoritmus használatát is, amely SRLG-független útpárokat keresett volna, a Dijkstra [8] algoritmus segítségével. Mivel magas p értékek mellett legtöbbször nem létezne SRLG-független útpár, ezért sok igényt blokkolnánk, így összehasonlítási alapként kevésbé felelne meg ez az algoritmus.



(a) Egyszeres hibák esetén (b) Alacsony SRLG sűrűség mellett

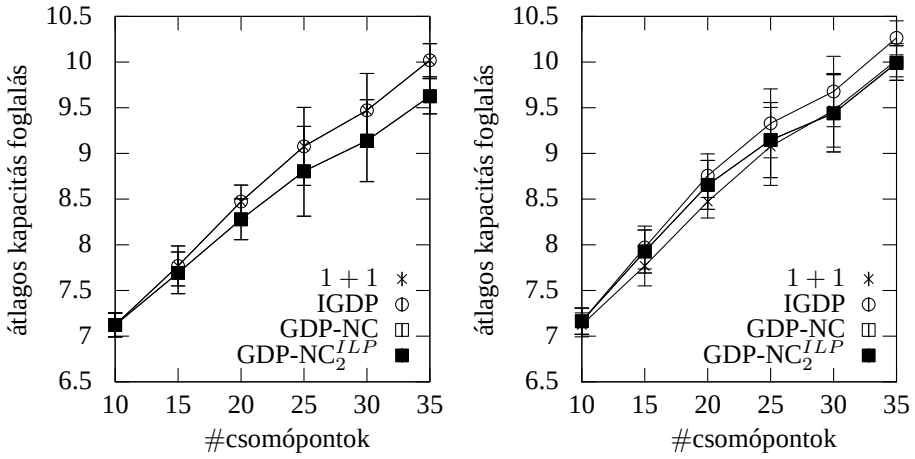
6.2. ábra. Átlagos kapacitás foglalás sűrű hálózat esetén (átlagos csomóponti fokszám 3.2 körüli).

6.3. Az algoritmusok sávszélesség foglalásának összehasonlítása

A 6.2(a). ábrán látható a sűrű hálózatokon egyszeres hibák mellett futtatott szimulációk eredménye. Megfigyelhető, hogy az IGDP, illetve a referencia $1 + 1$ algoritmusnak ebben az esetben ugyanaz a kapacitás-foglalása (például egy 10 csomópontos hálózatban átlagosan 5 élet kell használni az összeköttetések azonnali helyreállításának biztosításához). Ez azzal magyarázható, hogy az egyszeres hibákat az $1+1$ is tökéletesen védi, így a két algoritmus eredménye egybeesik. Azonban már a 6.2(b). ábrán látható alacsony SRLG sűrűség mellett az IGDP megoldásnak már nagyobb a kapacitás-foglalása. Természetesen ez nem jelenti azt, hogy az $1 + 1$ jobb eredményt produkálna, mivel az $1 + 1$ **nem képes az összes hiba védelmére**, míg az IGDP igen. Az ábrákon az is látszik, hogy minél nagyobb hálózatot vizsgálunk, annál nagyobb az átlagos kapacitásigény. Ez egyszerűen abból adódik, hogy nagyobb hálózatok estén a forrás és cél csomópontok távolabb eshetnek egymástól, így a két csomópont közötti útvonal több élen keresztül halad át, ami miatt nagyobb mennyiségű kapacitás-foglalására kényszerülünk.

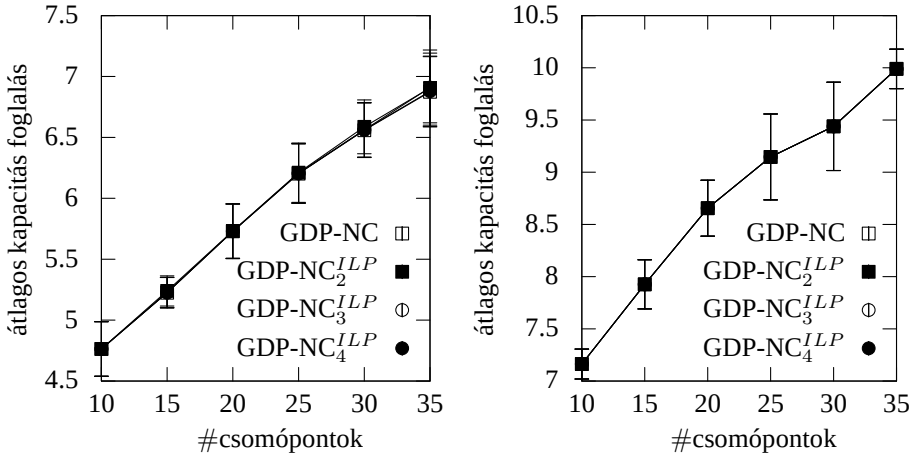
Továbbá az is leolvasható, hogy a GDP-NC, illetve a GDP-NC$_2^{ILP}$ mindkét esetben lényegesen jobb eredményt produkál, mint az IGDP és a referencia algoritmus: ez a folyamok osztásának, illetve a hálózat kódolás alkalmazásának köszönhető. Érdekes viszont, hogy a GDP-NC, illetve a GDP-NC$_2^{ILP}$ között nincsen lényegi különbség, vagyis, ha csak két részre osztható a felhasználói adatfolyam, már abban az esetben is igen jól közelítjük a hozzárendelt védelmi módszerek között optimális GDP-NC megoldást. Ezt a többi szimulációs eredmény is igazolta, sőt ritka hálózatok esetén a különbség még elenyészőbb, ahogy ez a 6.3. ábrán is megfigyelhető. Ez a forrás és cél csomópont közötti lehetséges utak számával függ össze, mivel ha két csomópont között kevés útvonal található, akkor a GDP-NC esetén is korlátozódik az adat oszthatósága.

Ha megengedjük az adatok három vagy négy részre osztását, akkor várhatóan jobban megközelítjük az optimális megoldást, ahogy ez a 6.4. ábrá igazolást is nyert. A GDP-NC$_2^{ILP}$ optimálishoz közeli



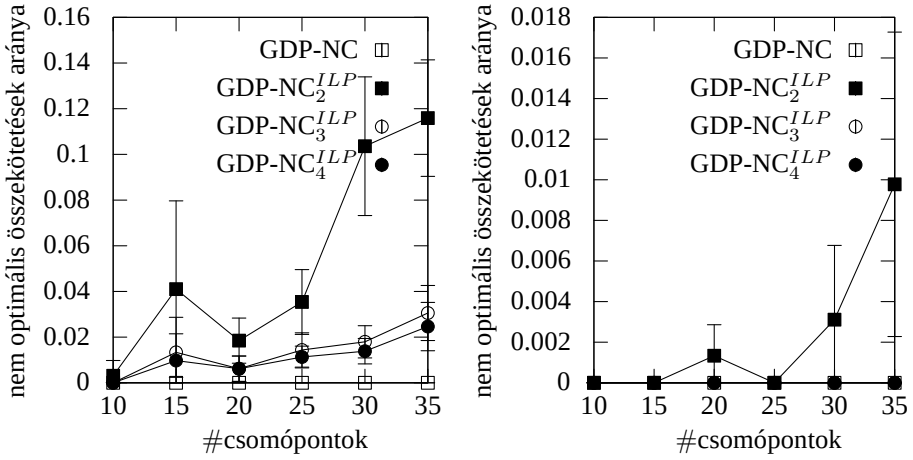
(a) Egyszeres hibák esetén (b) Alacsony SRLG sűrűség mellett

6.3. ábra. Átlagos kapacitás foglalás ritka hálózatok esetén (átlagos csomóponti fokszám 2.5 körüli).



(a) Sűrű hálózatok esetén (b) Ritka hálózatok esetén

6.4. ábra. Átlagos kapacitás foglalás, ritka és sűrű hálózatok esetén, az osztásszám függvényében, alacsony SRLG sűrűség mellett



(a) Sűrű hálozatok esetén (b) Ritka hálozatok esetén

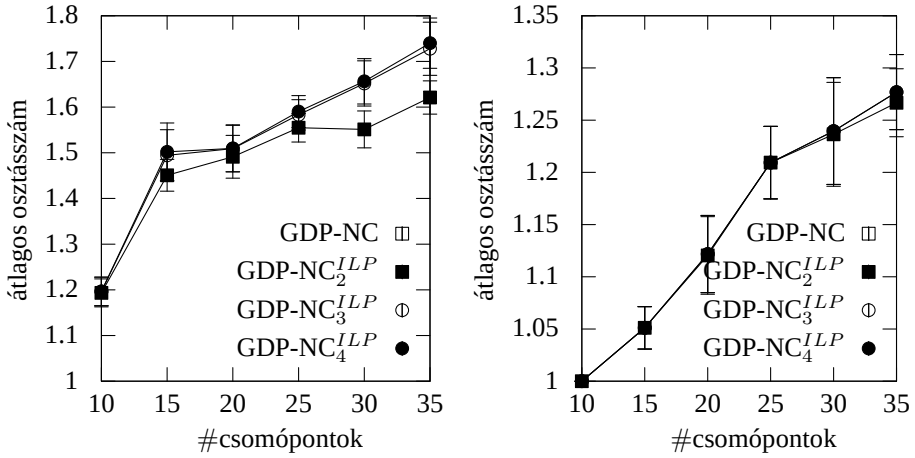
6.5. ábra. Nem optimális útvonalak százaléka, ritka és sűrű hálózatok esetén, az osztásszám függvényében, alacsony SRLG sűrűség mellett.

sávszélesség használata azonban kevésbé indokolja az alig kivehető javulásért cserébe fizetett komplexitás növekedést. Ezen módszerek teljesítményének értékeléséhez jobb összehasonlítási alapot ábrázoltam a 6.5. ábrán, ahol megmértem, hogy a százötven igény hány százalékában nem találtuk meg az optimális megoldást div_{max} osztással. Az ábrákon szerepel referenciaként a GDP-NC is, amely esetén az algoritmus jellegéből kifolyólag mindig megtaláljuk az optimális osztást, így annak értéke mindig nulla. Az ábrák alapján megállapítható, hogy minél több részre oszthatjuk a folyamot (sűrű és több csomóponttal rendelkező hálózatok), annál több esetben nem találjuk meg az optimális megoldást div_{max}-nál kevesebb osztással. Ami viszont gyakorlati szempontból lényeges, hogy a legrosszabb szcenárió estén is, a GDP-NC$_2^{ILP}$ az estetek mintegy 90%-ában az optimális megoldást adta. Ez azért fontos, mivel az adatok két részre bontása még viszonylag könnyen kezelhető a hálózatban [26], míg a további bontás egyre nagyobb komplexitást visz a rendszerbe, így annak szükségességét mérlegelni kell.

Egy további érdekes tendenciára enged következtetni a 6.6. ábra, amely az átlagos osztásszámot mutatja a csomópontok számának függvényében. Látható, hogy a hálózat méretének növekedésével nő az átlagos osztás is: ez szintén a csomópontok közötti utak számára vezethető vissza. Ezt támasztja alá az is, hogy a ritkább hálózat esetén az osztásszám elmarad a sűrűbb hálózatban lévőtől. Továbbá ez figyelhető meg egyre sűrűbb SRLG esetén is. Tehát minél több védendő hiba van, annál kevesebb az egyszerre használható lehetséges útvonal, így átlagban az átlagos osztásszám is csökken.

Összegezve, a szimulációs eredmények arra engednek következtetni, hogy a GDP-NC$_2^{ILP}$ minden esetben már igen jó megoldást ad. Viszont, ahogy növekszik a hálózat, illetve annak sűrűsége, annál nagyobb nyereséget hoz az adatok további osztása, mely a csomópontok közötti utak számával magyarázható.

A javasolt algoritmust más megközelítésből is megvizsgáltam: a harminchét csomópontú európai



(a) Sűrű hálozatok esetén (b) Ritka hálozatok esetén

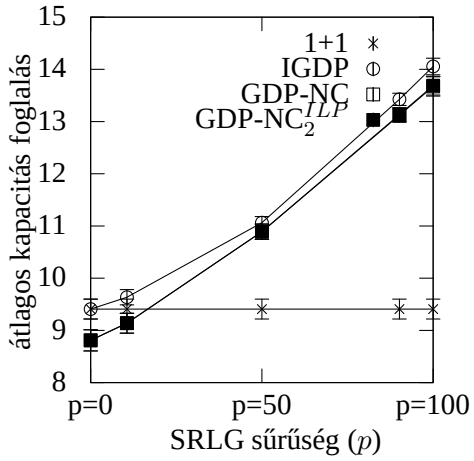
6.6. ábra. Átlagos osztásszám, ritka és sűrű hálózatok esetén, az osztásszám függvényében, alacsony SRLG sűrű mellett.

gerinchálózaton futtattam olyan szimulációt, amelyben az SRLG sűrűséget változtattam, vagyis az egyszeres hibák mellett megvizsgáltam az alacsony, közepes és magas SRLG sűrűségű esetet is, valamint a teljesség kedvéért belevettem a $p = 100$, vagyis az összes szomszédos kétszeres hiba esetét is. Ahogy ez a 6.7. ábráról leolvasható, a GDP-NC$_2^{ILP}$ ebben az esetben is kiválóan teljesített: a kapacitás-foglalásban szinte nem is látható különbség a GDP-NC-hez képest. Az eredményeket tovább erősíti, hogy a GDP-NC$_2^{ILP}$ a legrosszabb esetben is csak az igények mintegy 3%-át nem volt képes elvezetni az optimális módon.

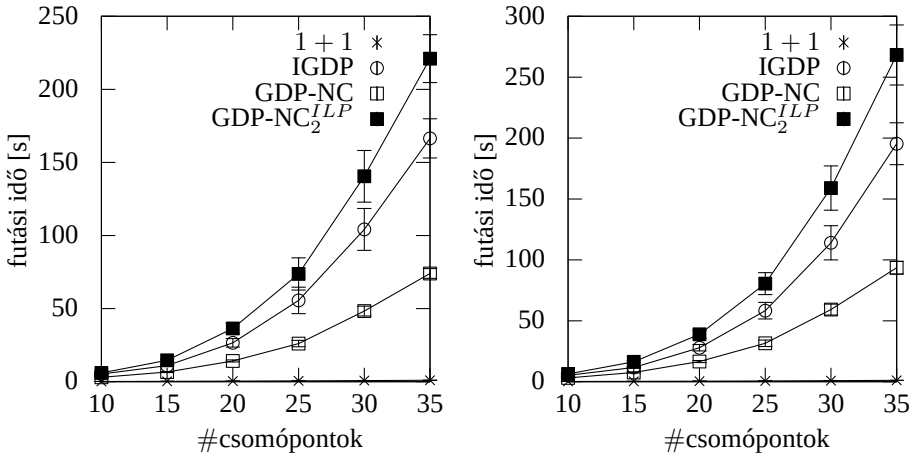
6.4. Az algoritmusok futási idejének összehasonlítása

Az algoritmusok futási ideje természetesen függ a hálózat nagyságától, illetve a megoldandó feladat komplexitásától is. A 6.8. ábrán megfigyelhető, hogy minél több csomópontot tartalmaz a hálózat, vagy minél nagyobb az SRLG sűrűség, annál hosszabb ideig tart az algoritmusok futása. Összességében elmondható, hogy az algoritmusok futási ideje a vártak szerint alakult: a referencia $1 + 1$ algoritmus fut le a leggyorsabban, aztán pedig a GDP-NC következik, amelyet LP-ként tudtunk megfogalmazni, azaz létezik polinomidejű algoritmus a megoldására. A két ILP-ként felírt feladat rendelkezik a leghosszabb futási idővel. A hálózat méretének növekedésével a futási idő meredeken növekszik, mind a GDP-NC^{ILP}, mind a többi módszer esetén is.

A különböző GDP-NC^{ILP} megoldások futási ideje viszont nem különbözik nagyban egymástól, mivel az esetek nagy többségében a felhasználói adatok kétfelé osztása már elégséges, így az algoritmus a GDP-NC$_3^{ILP}$ és a GDP-NC$_4^{ILP}$ esetén se fut tovább.



6.7. ábra. Európai harminchét csomópontos hálózat átlagos kapacitás foglalása változó SRLG sűrűség mellett ($p = 0$ esetén $|\mathcal{F}| = 57$, $p = 100$-nál pedig $|\mathcal{F}| = 189$)



(a) Egyszeres hibák esetén (b) Alacsony SRLG sűrűség mellett

6.8. ábra. Százötven igény teljes futási ideje ritka hálózatok esetén.

7. fejezet

Összefoglalás

Modern társadalmunk gyorsan növekvő igényei új megoldandó problémák elé állítják a szolgáltatókat. Egyre fontosabbá válik a telekommunikációs hálózatok megbízható és zavartalan működése, ennek kulcsa pedig az optikai gerinchálózatok magas rendelkezésre állása. Az optikai rétegben azonnali helyreállításra jelenleg leggyakrabban használatos védelmi megoldás, az $1+1$ hozzárendelt védelem a magas megbízhatóságú összeköttetések kialakításához védendő többszörös link hibák mennyisége miatt a jövőben már csak korlátozottan alkalmazható. Ezen probléma megoldására javasolták az általános hozzárendelt védelmet (GDP), amely a felhasználó igényeihez rugalmasan illeszkedő, megbízható összeköttetések kiépítését támogató védelemi megoldás. A GDP egy matematikai modell, amely általánosan fogalmazza meg a hozzárendelt útvonal-választási feladatot, és annak kiválasztását egy optimalizálási feladatként tekinti. A hálózatban rendelkezésre álló optikai eszközök, illetve alkalmazott technológiai megoldásoknak megfelelően korábban két GDP változatot javasoltak: az IGDP-t (osztatlan folyamok estén) és a GDP-NC-t (osztható folyamok és hálózat kódolás estén), ezeket az 5.2. fejezetben és az 5.3. fejezetben részletesen bemutattam. A GDP-NC optimális megoldást ad az azonnali helyreállítást garantáló módszerek sávszélesség foglalására, de a gyakorlatban nehezen kivitelezhető. Dolgozatomban egy olyan megoldást javasoltam, amely nagyon jól közelíti a GDP-NC kapacitás-foglalását, viszont a gyakorlati implementálása könnyebben kivitelezhető, mivel már létező, vagy egyszerűbben megvalósítható eszközöket használ. A javasolt védelmi megoldást GDP-NC^{ILP}-nek neveztem, melynek lényege, hogy a GDP-NC-hez hasonlóan a felhasználói adatfolyam továbbra is osztható, de nem tetszőlegesen, hanem előre adott számú részre, majd hálózati kódolás alkalmazásával ellenálló és robusztus védelem valósítható meg.

Alaposabban a gyakorlati megvalósíthatóság szempontjából legfontosabb esetet vizsgáltam, azaz amikor az adatfolyam maximum két részre osztható (GDP-NC$_2^{ILP}$ faladat). Ennek fontossága egyrészt abban rejlik, hogy az adatfolyam két felé osztását lehetővé tevő optikai eszközök jelen vannak az optikai hálózatban, másrészt a [26] cikkben mutatott kódolás eseten a hálózati eszközökben az egyszerű XOR művelet elvégzése elegendő (vagyis a test $GF(2)$) a robusztus kódolás megvalósításához. A kettő együtt pedig – a közel optimális erőforrás használat mellett – vonzóvá teheti a gyakorlati megvalósítását olyan

összeköttetések esetén, ahol az azonnali helyreállítás elengedhetetlen a magas QoS biztosításához. Szimulációkkal megmutattam, hogy a módszer már két részre történő osztás esetén is kiváló eredményt ad: legrosszabb estben is csak az igények 10%-a esetén nem találta meg az optimális elvezetését, a kapacitásfoglalás terén pedig még ennél is kisebb volt a különbség a GDP-NC$_2^{ILP}$ és az optimális GDP-NC között. Megfigyelhető volt az a tendencia, hogy ha növekszik a hálózat, illetve annak sűrűsége, akkor egyre nagyobb nyereséget hoz az adatok további osztása. Ez vélhetően a csomópontok közötti utak számával magyarázható, de ezen összefüggés pontos megállapításához további vizsgálatok szükségesek.

Összegezve, az általam javasolt módszer egy olyan, a gyakorlatban is megvalósítható megoldás, mely képes a bemenetben felsorolt SRLG-k ellenálló és robusztus védelmére, így biztosítva egy adott szolgáltatás-minőségi szintet. Ennek a módszernek a kapacitásfoglalása már kétszeres osztás mellett, tehát GDP-NC$_2^{ILP}$ esetén közelíti az optimumot, vagyis a GDP-NC-t. A kutatást abban az irányban tervezem folytatni, hogy mi történik abban az esetben, ha csak előre megadott számú csomópont rendelkezik kódoló képességgel. Ez annak az esetnek felel meg, hogy szolgáltatók egyszerre csak megadott számú csomópontot szerelnek fel a XOR kódolást megvalósító berendezésekkel. Ebben az átmeneti időszakban lényeges kérdés, hogy ez hogyan befolyásolja a hálózat költségeit, illetve az összeköttetések által lefoglalt sávszélességet.

Irodalomjegyzék

[1] LEMON: A C++ library for efficient modeling and optimization in networks. Technical report, http://lemon.cs.elte.hu.

[2] R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[3] P. Babarczi. *Survivable Optical Network Design with Unambiguous Shared Risk Link Group Failure Localization*. PhD thesis, Budapest University of Technology and Economics, 2012.

[4] P. Babarczi, J. Tapolcai, P.-H. Ho, and M. Médard. Optimal Dedicated Protection Approach to Shared Risk Link Group Failures using Network Coding. In *Proc. IEEE International Conference on Communications (ICC)*, pages 3084–3088, 2012.

[5] H. Choi, S. Subramaniam, and H. Choi. Loopback recovery from neighboring double-link failures in WDM mesh networks. *Information Sciences*, 149(1-3):197–209, 2003.

[6] P. Chou and Y. Wu. Network coding for the internet and wireless networks. *Signal Processing Magazine, IEEE*, 24(5):77–85, 2007.

[7] M. Davaadorzsín. *Valós lineáris algebra és lineáris programozás*. Műszaki könyvkiadó, 2001.

[8] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[9] G. Ellinas, E. Bouillet, R. Ramamurthy, J. Labourdette, S. Chaudhuri, and K. Bala. Routing and restoration architectures in mesh optical networks. *Optical Networks Magazine*, 4(1):91–106, 2003.

[10] C. Fragouli, J. Le Boudec, and J. Widmer. Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review*, 36(1):63–68, 2006.

[11] A. Fumagalli and L. Valcarenghi. IP restoration vs. WDM protection: is there an optimal choice? *Network, IEEE*, 14(6):34–41, 2000.

[12] P.-H. Ho, J. Tapolcai, and T. Cinkler. Segment shared protection in mesh communication networks with bandwidth guaranteed tunnels. *IEEE/ACM Transactions on Networking*, 12(6):1105–1118, December 2004.

[13] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *Information Theory, IEEE Transactions on*, 52(10):4413–4430, 2006.

[14] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *Information Theory, IEEE Transactions on*, 51(6):1973–1982, 2005.

[15] Y. S. Kavian and M. S. Leeson. *Resilient Optical Network Design: Advances in Fault-Tolerant Methodologies*. IGI Global, 2012.

[16] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM transactions on networking*, 11(5):782–795, 2003.

[17] K. Kompella and Y. Rekhter. Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS), Internet Draft. Technical report, http://tools.ietf.org/html/rfc4202, 2005.

[18] S. Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz. Pan-European optical transport networks: an availability-based comparison. *Photonic Network Communications*, 5(3):203–225, 2003.

[19] E. Manley, J. Deogun, L. Xu, and D. Alexander. All-optical network coding. *Optical Communications and Networking, IEEE/OSA Journal of*, 2(4):175–191, 2010.

[20] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot. Characterization of failures in an IP backbone. In *Proc. IEEE Infocom*, volume 4, pages 2307–2317. Citeseer, 2004.

[21] S. Morris and J. Reilly. Mil-hdbk-217-a favorite target. In *Reliability and Maintainability Symposium, 1993. Proceedings., Annual*, pages 503–509. IEEE, 1993.

[22] F. Musumeci, M. Tornatore, and A. Pattavina. A power consumption analysis for ip-over-wdm core network architectures. *Journal of Optical Communications and Networking*, 4(2):108–117, 2012.

[23] S. Orlowski and M. Pioro. On the complexity of column generation in survivable network design with path-based survivability mechanisms. In *International Network Optimization Conference (INOC)*, 2009.

[24] H. Overby, G. Biczók, P. Babarczi, and J. Tapolcai. Cost Comparison of 1+1 Path Protection Schemes: A Case for Coding. In *Proc. IEEE International Conference on Communications (ICC)*, pages 3100–3105, 2012.

[25] L. Page and J. Perry. A model for system reliability with common-cause failures. *Reliability, IEEE Transactions on*, 38(4):406–410, 1989.

[26] S. Rouayheb, A. Sprintson, and C. Georghiades. Robust network codes for unicast connections: A case study. *IEEE/ACM Transactions on Networking*, 19(3):644–656, 2011.

[27] P. Sanders, S. Egner, and L. Tolhuizen. Polynomial time algorithms for network information flow. In *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, pages 286–294. ACM, 2003.

[28] D. Shier. *Network reliability and algebraic structures*. Clarendon Press New York, NY, USA, 1991.

[29] P. Soproni, P. Babarczi, J. Tapolcai, T. Cinkler, and P.-H. Ho. A meta-heuristic approach for non-bifurcated dedicated protection in wdm optical networks. In *Design of Reliable Communication Networks (DRCN), 2011 8th International Workshop on the*, pages 110 –117, oct. 2011.

[30] J. Spragins. Dependent Failures in Data Communication Systems. *IEEE Transactions on Communications, [legacy, pre-1988]*, 25(12):1494–1499, 1977.

[31] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.

[32] K. Thulasiraman, M. Javed, and G. Xue. Circuits/cutsets duality and a unified algorithmic framework for survivable logical topology design in ip-over-wdm optical networks. In *INFOCOM 2009, IEEE*, pages 1026 –1034, april 2009.

[33] M. Tornatore, M. Carcagni, and A. Pattavina. Availability formulations for segment protection. *Communications, IEEE Transactions on*, 58(4):1031 –1035, April 2010.

[34] J. Vasseur, M. Pickavet, and P. Demeester. *Network recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann Publishers, 2004.

[35] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger. General availability model for multilayer transport networks. In *Proceedings. 5th International Workshop on Design of Reliable Communication Networks, 2005.(DRCN 2005).*, page 8, 2005.

[36] H. Wang, E. Modiano, and M. Médard. Partial path protection for WDM networks: End-to-end recovery using local failure information. In *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, pages 719–725. IEEE, 2002.

[37] G. Xue, W. Zhang, T. Wang, and K. Thulasiraman. On the partial path protection scheme for WDM optical networks and polynomial time computability of primary and secondary paths. *MANAGEMENT*, 3(4):625–643, 2007.

Tárgymutató