BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
TÁVKÖZLÉSI ÉS MÉDIAINFORMATIKAI TANSZÉK
VILLAMOSMÉRNÖKI TUDOMÁNYOK DOKTORI ISKOLA

AZONNALI HIBA HELYREÁLLÍTÁS ÉS LOKALIZÁCIÓ AZ ÁTVITELI HÁLÓZATOKBAN

Pašić Alija
okleveles villamosmérnök

Tézisfüzet

Tudományos vezető

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

Budapest
2018

1. Bevezetés

Napjainkban a gerinchálózati szolgáltatók az új alkalmazásoknak (távsebészet, tőzsde, VoIP, multimédia) köszönhetően egyre szigorúbb minőségi (QoS - Quality of service) követelményekkel szembesülnek, mind a rendelkezésre állás, mind a késleltetés terén. Továbbá, a jelentősen megnövekedett (Tbyte/sec-es nagyságrendű) adatforgalom miatt az erőforrások (pl. átviteli kapacitás) is egyre szűkebb keresztmetszetet képeznek. Így korántsem meglepő, hogy az átviteli hálózatok késleltetés tűrése és megfelelő hibavédelme igencsak aktuális kérdés.

Mára már az átviteli hálózatok átalakultak. Egyrészt a teljesen optikai alapú átvitel került előtérbe, kiküszöbölve az időigényes és költséges optikai/elektromos/optikai (O/E/O) jelátalakítást [16] [17]. Másrészt megjelent egy új technológia, a Software-Defined Networking (SDN), amely teljesen új alapokra helyezi a hálózatmenedzsmentet. Az SDN esetén a hálózatvezérlő és adattovábbító funkciók teljesen szétválasztódnak, így téve a hálózati vezérlést közvetlenül programozhatóvá. Nem meglepő tehát, hogy az új módszereknek meg kell felelniük az új technológiai (teljesen átlátszó optikai hálózatok, SDN) igényeknek és követelményeknek.

Viszont a magas QoS szint, vagyis a megbízható hálózatok tervezése és üzemeltetése továbbra is a szolgáltató érdekében áll, mivel a jól működő, megbízható, QoS garanciákat vállaló hálózatokért az üzleti felhasználók lényegesen nagyobb összeget is hajlandóak fizetni [12]. A felek ezeket a feltételeket a szolgáltatásminőségi (SLA - Service Level Agreement) szerződésben specifikálják. Abban az esetben ha nem képes a szerződésben előírt minőségű szolgáltatást nyújtani, a szolgáltató komoly összegű kötbért fizet a felhasználónak. Ennek elkerülése érdekében a szolgáltató törekszik a hálózat megfelelő (QoS szerinti) modellezésére, valamint a megfelelő hibamenedzsmenti technológiák kiválasztására, természetesen az alacsony létesítési és üzemeltetési költségek mellett. Ezekben az új hálózatokban a Tbyte/sec-os nagyságrendű [18] [19] átviteli sebességek miatt a legrövidebb idejű kiesés sem megengedett, mivel ez hatalmas adatmennyiségek elvesztésével és a QoS szint drasztikus csökkenésével jár. Így minden kétséget kizáróan mondhatjuk azt, hogy az *azonnali helyreállítás* kulcskérdés a megfelelő QoS biztosításában.

1.1. Definíció. Azonnali helyreállításról *akkor beszélünk, ha a teljes helyreállítási*

idő kevesebb mint 50ms ($t_R \leq 50ms$) [16].

Ahhoz, hogy a helyreállítási időt érdemben vizsgálni tudjuk, ismernünk kell a helyreállítási ciklust (GMPLS [14]). Ennek fő lépései a következők:

1. hibaészlelési idő (t_l): vagyis a meghibásodás és annak észlelése között eltelt idő,
2. hibaértesítési idő (t_n): a hibát észlelő hálózati entitások szétterjesztik a hiba tényét a vezérlő síkon (control plane, CP) küldött jelzések segítségével, hogy a megszakadt összeköttetések helyreállításáért felelős csomópontok fel tudjanak készülni a helyreállításra,
3. korrelációs idő (t_c): az az idő, ami alatt az összes hibaüzenet beérkezik,
4. döntéshozatali idő (t_d): az az idő, ami alatt döntés születik arról, hogy milyen védelmi kapcsolás szükséges,
5. kapcsolási idő (t_s): a csomópontok konfigurációjához, kapcsolók beállításához szükséges idő (az optikai rétegben ez a leginkább időigényes feladat, több tíz milliszekundumot vehet igénybe [36, 37]):

$$t_R = t_l + t_n + t_c + t_d + t_s. \tag{1}$$

Célunk tehát a teljes helyreállítási idő 50ms alatt tartása, így biztosítva azonnali helyreállítást. Azonban az azonnali helyreállítás nem az egyetlen fontos jellemzője egy védelmi megoldásnak, hanem a robusztusság kérdése is igen jelentős.

1.2. Definíció. Robusztus *védelmi megoldásról akkor beszélnünk, ha a helyreállítás során a vezérlő síkban nincs szükség üzenetek küldésére, valamint az optikai kapcsolók (OXC - Optical Cross-Connect) újrakonfigurálására [J1].*

Disszertációm első felében olyan új módszereket javaslok, amelyek *robusztus módon azonnali helyreállítást* biztosítanak, valamint hiba után is eleget tesznek a késleltetési kényszereknek alacsony erőforrásigény mellett. Ezen új módszereket összehasonlítom a jelenleg leggyakrabban használatos technológiákkal.

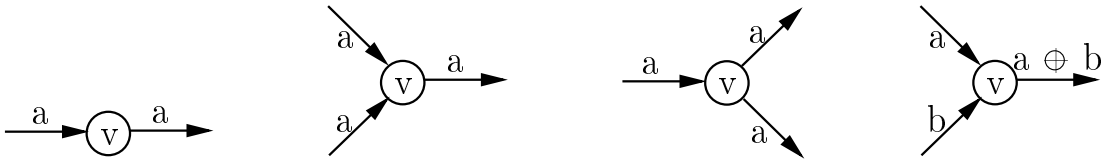
Disszertációm második felében a hibalokalizáció kérdésével foglalkozom, vagyis olyan új lokális hibalokalizációs módszert javaslok, amely lehetővé teszi az igen gyors

helyreállítást megosztott védelmi módszerek esetén is (t_R számottevően csökken). A megosztott védelmek természetéből adódóan ekkor is szükség van az optikai kapcsolók (OXC) újrakonfigurálására, vagyis ezen módszerek *semmilyen esetben sem lesznek robusztusak*. Kiemelendő, hogy mivel az optikai kapcsolók (OXC) újrakonfigurálásának ideje a technológiától függően akár több 10ms lehet [36, 37], így annak ellenére, hogy a hibaértesítési idő és hibalokalizációs idő is csak néhány milliszekundum, mégsem garantálható minden esetben az azonnali helyreállítás.

2. Kutatási célkitűzések

Az értekezésem célja, hogy az átviteli hálózatokban olyan új módszereket javasoljak amelyek amellett, hogy azonnali helyreállítást biztosítanak, még a hiba bekövetkezése után is képesek eleget tenni késleltetési kényszereknek, mégpedig alacsony erőforrás igény mellett.

Tehát az új technológiai trendeknek (teljesen átlátszó optikai hálózatok, SDN) megfelelő hálózati képességekkel rendelkező hálózatokban javaslok olyan új módszereket, amelyek erőforrás takarékos módon képesek robusztus azonnali helyreállítást biztosítani, valamint megfelelni különböző késleltetési kényszereknek még a hiba bekövetkezése után is. A technológiából adódó kényszerek a következőek:



(a) Továbbítás (b) Összevonás (c) Osztás (d) Hálózati kódolás

1. ábra. Alapvető csomóponti szerepek, melyekbe az összeköttetés által használt valamennyi v csomópont besorolható

– A hálózatok a folyamok osztását csak korlátozottan tudják támogatni- ez várhatóan a jövőben sem fog változni. Vagyis a felhasználói adat nem bontható tetszőleges számú részre [J1].

– A hálózat belsejében csak egyszerű műveletek végrehajtására van lehetőség [J1]. Ilyen egyszerű művelet a XOR (Exclusive OR), amely már tisztán optikai módon is megvalósítható [20]. Az 1. ábrán láthatóak a csomóponti képességek, vagyis a klasszikus csomóponti képességek mellett az osztás, összevonás és az egyszerű kombináció (hálózati kódolás) művelete is megengedett.

Az értekezés első részében egy új, rendkívül erőforrás-hatékony és a gyakorlatban megvalósítható egyszeres linkhiba ellenálló módszert mutatok be, az úgynezevzett Survivable Routing with Diversity Coding-ot (SRDC). A módszer könnyen telepíthető, mivel a *a kódolás a forrás és a cél csomópontokban elvégezhető, azaz nincs szükség a csomóponti képességek frissítésére*. Az SRDC több alproblémáját vizsgáltam, a hálózat kapacitása és a csomóponti képességek függvényében. A többszörös hibák védelmére a Survivable Routing with Network Coding (SRNC) került bevezetésre. Számos kutatás foglalkozik a QoS és a differenciális késleltetést (DD) figyelembe vevő megbízható útvonalválasztás kérdésével a transzport hálózatokban, azonban mindegyik munka a diszjunkt utak kérdését veszi szemügyre. Disszertációmban ezen eredményeket általánosítom az irányított körmentes gráf (DAG - Directed Acyclic Graph) struktúrát használó diverzitás kódolás alapú megbízható útvonalválasztási módszerekre. Ehhez meghatározom a DAG-ok meghibásodás előtti és utáni késleltetését, ezután pedig megvizsgálom a QoS és a DD tudatos útvonalválasztás késleltetési kényszereinek hatását az azonnali helyreállítást biztosító optimális megbízható útvonalakra. Disszertációm második felében a hibalokalizáció kérdésével foglalkozom. Javaslok egy olyan új lokális hiba monitorozó keretrendszert, amely lehetővé teszi az igen gyors helyreállítást megosztott védelmi módszerek esetén is. Illetve bevezetek egy új koncepciót is, a tiltott linkpárok fogalmát, amelynek segítségével általánosítok több monitorozó út alapú problémát.

Kiemelendő, hogy a megosztott védelmi módszerek természetéből adódóan ekkor is szükség van az optikai kapcsolók (OXC) újrakonfigurálására, vagyis ezen módszerek *semmilyen esetben sem lesznek robusztusak*, viszont igencsak erőforrástakarékosnak bizonyulnak.

3. Módszertan

A tézisekben javasolt módszerek jellemzőit és helyességét bonyolultságelméleti és gráfelméleti eredményeket felhasználva igazoltam, vagyis analitikus módszerekkel.

Az általam javasolt eljárásokat a LEMON [3] hálózati optimalizációs $C++$ osztálykönyvtár segítségével implementáltam. Ezáltal szimulációkkal igazoltam a módszerek helyességét és hatékonyságát, összehasonlítva azokat az irodalomban korábban javasolt módszerekkel. Az NP-teljes komplexitású problémák optimális megoldásához - az irodalomban általánosan elfogadottak alapján -, egészértékű lineáris programokat (ILP - Integer Linear Programming) fogalmaztam meg. Ehhez a GUROBI [2] és CPLEX [1] solverek segítségét vettem igénybe.

4. Új eredmények

A telekommunikációs hálózatokban a megbízhatóság - vagyis a hálózat magas rendelkezésre állása - az alacsony késleltetés mellett az egyik legfontosabb kérdés. Természetesen ahhoz, hogy a kérdéssel érdemben foglalkozni tudjunk, modelleznünk kell a valóságot. Ennek érdekében, az adott hálózatot egy $G = (V, E)$ irányított gráf segítségével reprezentáljuk, ahol a csomópontok a hálózati csomópontokat jelölik, míg az élek a kommunikációra alkalmas csatornát. A hibák modellezésére számos módszer létezik. Értekezésemben az irodalomban legelterjedtebb, az úgynevezett közös kockázatú csoportok (Shared Risk Link Group, SRLG) [34] módszerét használom. Ez a módszer az egymástól nem független meghibásodásokat is kitűnően kezeli. Egy SRLG képes kifejezni a statisztikai összefüggéseket a benne foglalt linkek között, tehát képes figyelembe venni a fizikai és a logikai topológia közötti összefüggéseket. Egy SRLG tetszőleges számú linket tartalmazhat, illetve minden link tetszőleges számú SRLG-ben szerepelhet [23]. Maguk az SRLG-k, amelyek lényegében azt fejezik ki, hogy az egyes események milyen linkek kiesésével járnak, statikus információnak számítanak. A továbbiakban az egy adott szcenárióhoz (például egy adott QoS szinthez) tartozó hibalistát $\mathcal{F}$-el jelölöm. Mivel az átviteli hálózatok elemeinek igen magas minőségi követelményeknek kell megfelelniük, ezért ezekben a hálózatokban igen ritkán fordulnak elő egyszerre független hibák (más néven többszörös hibák). Vagyis gyakorlati

szempontból az átviteli hálózatokban az egyszeres hibák védelme kiemelkedően fontos [13]. Ezért értekezésemben is főként ezen meghibásodások azonnali helyreállítását és lokalizációját vizsgálom.

4.1. Azonnali helyreállítás átviteli hálózatokban

Az átviteli hálózatoknál a szállított (Tbyte/sec) adatmennyiségek miatt a kiesések a legritkább esetben megengedettek. A nagy fizikai távolságok miatt viszont ezen hálózatok igenis hajlamosak a különféle meghibásodásokra, link hibákra. Ezért ezekben a hálózatokban már régóta használunk különböző védelmi megoldásokat [16]. A gerinchálózatok védelme esetén három fontos aspektust kell figyelembe vennünk, ezeknek az épp megfelelő vagy nem megfelelő kombinációja határozza meg egy védelmi megoldás hatékonyságát. A három aspektus a következő: a helyreállítási idő, a komplexitás (mind a számítási komplexitás, mind a védelem által védett hibák komplexitása - tetszőlegessége - szerepet játszik), és a védelmi megoldás miatt lefoglalt extra kapacitás, vagyis az erőforrás menedzsment.

Joggal merül fel a kérdés, hogy a három szempont közül melyik is a legfontosabb? Ehhez érdemes a gyakorlatot megvizsgálnunk.

Jelenleg továbbra is a hozzárendelt védelmek családjába tartozó úgynevezett $1 + 1$ védelem a legelterjedtebb, melynek lényege, hogy minden összeköttetés igény adatát két diszjunkt útvonalon egyszerre szállítjuk, vagyis egyrészt az üzemin, míg másrészt a védelmi útvonalon is továbbítjuk ugyanazon felhasználói adatokat. Így az üzemi út meghibásodása esetén a nyelő átkapcsol a védelmi útvonalra, azonnali helyreállítást biztosítva (vagyis a helyreállítási idő kevesebb, mint 50ms). Természetesen a kapacitásfoglalás az elsődleges igénynek legalább a kétszerese, ami az $1 + 1$ Achilles sarka. Valamint az $1 + 1$ csak egyszeres hibák védelmére alkalmas és csak egyetlenegy költség paramétert vesz figyelembe. Vagyis az újonnan felmerülő komplex QoS (Quality of Service) igényeknek *nem képes* eleget tenni. Viszont igazi előnye, hogy *egyszerű robusztus módon azonnali helyreállítást biztosít*, ez pedig máig a vezető védelmi módszerré teszi az $1 + 1$ védelmet, pazarló erőforrás igénye ellenére [7].

Természetesen léteznek olyan védelmi módszerek is, amelyek a hálózati erőforrásokkal igen gazdaságosan bánnak, ilyenek a megosztott védelmi módszerek [9, 10].

Lényegük, hogy egy egységnyi védelmi kapacitáson több üzemi útvonal osztozik, mégpedig olyan üzemi útvonalak, amelyek a hiba szempontjából egymástól függetlenek. Ilyenkor tehát a hiba esetén egyszerre legfeljebb egy felhasználó (vagyis érintett üzemi útvonal) szeretné használni a megosztott védelmi erőforrásokat. Ezen módszerek hátránya, hogy (a megosztás mértékétől függően) az útvonalak meghatározása komplex feladat. Továbbá a hiba bekövetkezése után jelzésüzenetek küldésére van szükség, mivel a meghibásodott üzemi útvonaltól függően más és más csomópontoknak kell a védelmi útvonalakat, kapacitásokat használnia. Ezen üzenetek küldése természetesen időigényes, így az azonnali helyreállítás semmiképpen sem garantált, ami gerinchálózatok esetén követelmény. Így ezek a módszerek bonyolultságuk és lassúságuk miatt még nem jelentenek konkurenciát az $1+1$ számára.

Természetesen az $1+1$ nem az egyetlen hozzárendelt védelmi megoldás. Sok kutatás foglalkozott azzal, hogy a (hozzárendelt) védelmet hogyan lehet hatékonyan kiterjeszteni több link- vagy csomópont hibára is. Ez különösen abban az esetben nehéz kérdés, amikor a hálózat ritka [C2]. Ezen nehéz feladat megoldására javasolták a GDP-t [J1] (General Dedicated Protection) is, amely hasonlóan az $1+1$-hez a felhasználói adatokat több útvonalon szállítja. De az $1+1$-gyel ellentétben ezek nem végponttól végpontig terjedő útvonalak, hanem útvonal szegmensek. Ezen módszer képes bármely tetszőleges védhető $\mathcal{F}$ hiba (SRLG) lista védelmére, azonnali helyreállítást garantálva. Védhető alatt itt azokat a hibákat értjük, amelyek esetében a hálózat összefüggő marad és az igények elvezetésére (üzemi és védelmi útvonal létrehozására) egyaránt elegendő kapacitás áll rendelkezésre.

Ha a folyamok osztása nem megengedett, akkor az úgynevezett GDP-R (General Dedicated Protection with Routing) problémáról beszélünk, amelynek megoldása erőforrás-igényes és *NP-teljes* [J1]. Abban az esetben, ha megengedjük a folyamok osztását és hálózati kódolást [J1] is alkalmazunk, egy olyan módszert kapunk, amelynek a kapacitásfoglalása az összes hozzárendelt védelem közül a legalacsonyabb, ráadásul ezen probléma *polinom* időben megoldható.

4.1. Definíció. Hálózati kódolásról *akkor beszélünk, amikor a hálózat közbülső csomópontjaiban is megengedett az adatok kódolása (kombinálása) [24] [25].*

Ekkor a hálózati csomópontok nemcsak az adatok továbbítására, átirányítására képesek, hanem bizonyos műveleteket is képesek végrehajtani azokon. Így a kimenet

adatfolyama a csomópontba korábban beérkezett bemenetek valamilyen matematikai függvényeként képzelhető el. A célcsomópontnak természetesen vissza kell tudnia kódolni az adatfolyamokat ahhoz, hogy visszanyerje az eredeti információkat. A *gyakorlatban ez a módszer nem implementálható*, mivel az *adatok tetszőleges osztásának engedélyezése nem oldható* meg, valamint azok a bonyolult aritmetikai műveletek, amelyek szükségesek a hálózati kódolás kivitelezéséhez, *még nem valósíthatók meg az optikai rétegben*. De mivel ez a módszer *elméleti alsó korlát* kapacitásfoglalás szempontjából, így igen jól használható a módszerek hatékonyságának kiértékelésénél. Fontos kiemelni, hogy a hálózati kódolást alkalmazó módszerek esetén a kimenet része *egy megbízható routing* $R = (V^R, E^R, f)$, amely tartalmazza a megoldásban szeplő, csomópontokat, éleket, illetve az ahhoz rendelt folyamértékeket, valamint egy *robusztus konfiguráció* $\mathcal{C}$, amely megadja az optikai csomópontok (OXC) beállításait. Az ilyen beállítási információk közé tartoznak a kapcsoló mátrixok, jelek összevonása, esetleg a hálózati kódoláshoz szükséges információk.

A disszertációmban a megbízható routing feladat problémájával foglalkozom. Fontos, hogy az általam bemutatott módszerekhez a megbízható routing alapján kinyerhető a *robusztus konfiguráció* $\mathcal{C}$ már ismert módszerek [27, 28] segítségével.

1. Tézis. *[C1, C2, J1] Javasoltam egy új, hálózati kódolás alapú azonnali helyreállítást biztosító, robusztus módszert, vagyis az SRNC-t (Survivable Routing with Network Coding). Megmutattam, hogy általános hibalista ($\mathcal{F}$) estén a feladat NP-teljes komplexitású, bármely adott számú osztás esetén. Ezek alapján egészértékű lineáris programot javasoltam ezen probléma megoldására. Az egyszeres linkhibák esetére javasoltam egy új, robusztus azonnali helyreállítást biztosító módszert, az SRDC-t (Survivable Routing with Diversity Coding). Az SRDC-t kapacitás és csomóponti kényszerek mellett is vizsgáltam (lásd 2. táblázat). A problémák megoldására javasoltam ILP-t és több heurisztikát is. Valamint beláttam, hogy kapacitáskorlát nélküli esetben (ICAN és ICCN) ezen problémának az* $1+1$ *a 4/3-os approximációja. Továbbá beláttam, hogy a kapacitáskorlátos esetet (ICAN és ICCN) az* $1+1$ *nem approximálja. A csomóponti kényszeres esetre (ICCN) egy approximációs algoritmust adtam (SRDC-IC).*

A feladat bemenetét képezi a dinamikus összeköttetés igény érkezésének pillanatában a hálózat aktuális állapotát leíró irányított gráf: $G = (V, E)$, ahol minden

$e \in E$ élhez adott: egy nemnegatív költségfüggvény ($c : E \to R^+$), az aktuális szabad kapacitás ($k : E \to R^+$), valamint az adott linkhez tartozó késleltetés ($d : E \to R^+$). Az összeköttetés igény $C = (s, t, b, D)$, amely tartalmazza: a forrást (s) és nyelőt (t), az igényelt sávszélességet ($b \in \mathbb{N}$) tetszőleges egységben megadva. Valamint abban az esetben, ha az összeköttetésnek valamilyen késleltetés kényszert is teljesítenie kell, a késleltetési küszöböt (D). Továbbá a probléma bemenetét képezi a védeni szükséges hibák listája (SRLG-k: $\mathcal{F}$) is, amely segítségével általános esetben minden hibához (SRLG-hez - hibamintához) tartozik egy $\forall f \in \mathcal{F} : G_f = (V, E_f)$ segédgráf (E_f élhalmazt az f hibamintában megadott meghibásodott élek törlésével nyerjük E-ből). Feltételezzük, hogy valamennyi SRLG az $\mathcal{F}$ listában *védhető*, azaz minden G_f gráf $s - d$ összefüggő. Fontos, hogy ezen segédgráfok létrehozása általános hibalista esetén szükséges, míg abban az esetben, ha egyszeres hibákat feltételezünk egyéb segédgáfok használata a célszerű. Itt fontos leszögezni, hogy míg általános hibamodell esetén a hálózat belsejében is szükség lehet hálózati kódolásra (természetesen ekkor $\mathcal{C}$ tartalmazza a megfelelő információkat), addig egyszeres hiba esetén **kizárólag a hálózat szélén** [27, 28], vagyis a forrás és nyelő csomópontban kell hálózati kódolást használni. Vagyis a két probléma merőben különbözik egymástól, ezért is van szükség a két probléma különálló kezelésére.

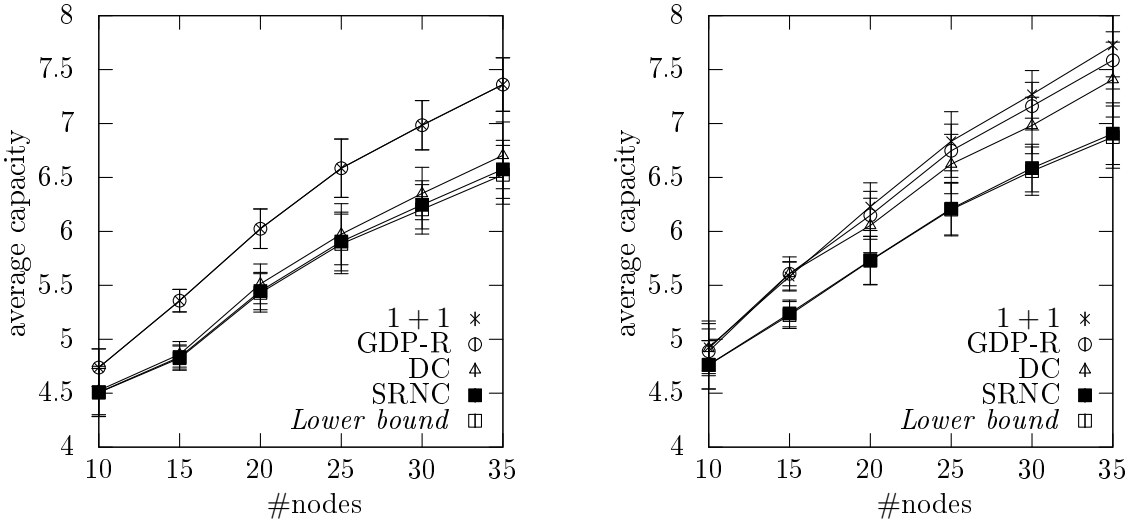
4.1.1. Átviteli hálózatok megbízhatósága általános hibamodell esetén

1.1. Tézis (SRNC komplexitás). *[C2, J1] Bebizonyítottam, hogy az SRNC NP-teljes komplexitású. Ezen probléma megoldására egészértékű lineáris programot javasoltam.*

A bizonyításhoz az osztatlan folyamos GDP, vagyis GDP-R probléma SRNC-re való visszavezetését mutattam meg [C2]. A bizonyítás egy gráfkonstrukción alapszik. Így, hogy beláttam, hogy az SRNC probléma NP-teljes, az optimális megoldás kinyerésére a szakirodalomban általánosan elfogadott és használt módszert használtam, vagyis felírtam a megfelelő egészértékű lineáris programot. A program végigiterál mindegyik $\forall f \in \mathcal{F} : G_f = (V, E_f)$ segédgráfon, így figyelembe véve az adott tetszőleges hibalistát, közben egy segédváltozó segítségével beállítva a végleges megoldás költségeit az éleken.

1. táblázat. SRNC és SRDC védelem esetén alkalmazott jelölések

Jelölés	Leírás
$G = (V, E, k, c, d)$	a hálózat irányított gráfmodellje V csomópont és E élhalmaz esetén c az éleken definiált költségfüggvény $e \in E$ k az éleken rendelkezésre álló szabad kapacitás $e \in E$ d az adott link késleltetése $e \in E$
$C = (s, t, b, D)$	a dinamikusan érkező igény forrás- és célcsomópontja, sávszélességigénye, valamint késleltetési kényszere
$\mathcal{F}$	a védendő közös kockázatú csoportok listája
$G_f = (V, E_f)$	az adott $f \in \mathcal{F}$ csoportban meghibásodott élek törlésével nyert SRLG gráf
$R = (V^R, E^R, f)$	az összeköttetés megbízható routingja $V^R \subseteq V$ csomópont, és $E^R \subseteq E$ élhalmazzal, és $\forall e \in E^R : f(e) \leq k(e)$ folyamértékekkel
$\mathcal{C}$	a csomópontok konfigurációja
A, B	a két adatrészlet, amire a felhasználói folyamot osztjuk
$A \oplus B$	XOR művelettel az s ben létrehozott redundáns adatrészlet
$E_A, E_B, E_{A \oplus B}$	routing DAGok A, B és $A \oplus B$ számára



(a) Egyszeres linkhibák (fokszám 3.2)

(b) Többszörös linkhibák (fokszám 3.2)

2. ábra. Átlagos kapacitás foglalás a különböző hibalisták és hálózatok esetén. [J1]

Természetesen az ILP minimalizál egy adott költségfüggvényt, vagyis egy olyan megoldást keresünk, ahol minimalizáljuk az összeköttetés számára lefoglalt erőforrásokat:

$$\min \sum_{\forall e \in E} c(e) \cdot \frac{b(e)}{b \cdot 2}. \tag{2}$$

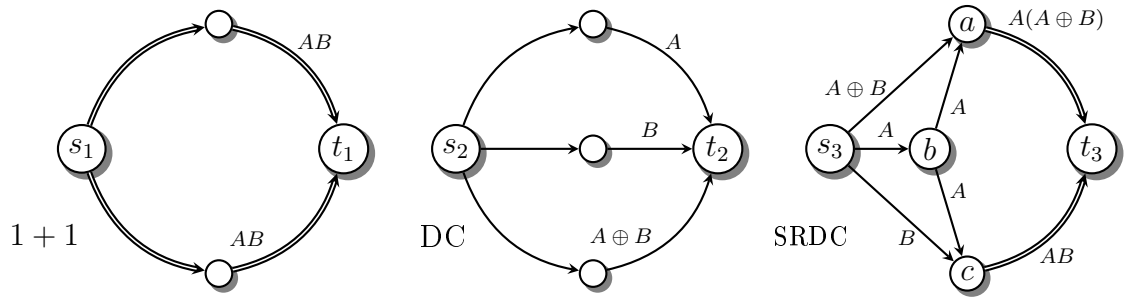
ahol $b(e)/2 \leq k(e)$ a megoldásban használt sávszélesség az $e \in E$ linken. A kettővel való osztásra azért van szükség, mivel az adatokat két részre (A és B-re) bontottuk és ezeket (külön) vezetjük el.

Az alkalmazott jelöléseket az 1. táblázat tartalmazza.

Kiemelendő, hogy szimulációs eredmények segítségével megmutattam, hogy az adatok két részre való osztása esetén is nagyon jól meg lehet közelíteni az elméleti alsó korlátot (az ábrán *Lower bound*), vagyis a tetszőleges osztást engedélyező és hálózati kódolást alkalmazó GDP-t (2 ábra).

4.1.2. Átviteli hálózatok megbízhatósága egyszeres hibák esetén

A hibák túlnyomó része (több, mint 70% [13]) egyszeres hiba, ezért a jelenlegi hálózatok jellemzően vagy védelem nélkül, vagy egyszeres hiba elleni védelem mellett



3. ábra. A különböző azonnali helyreállítást nyújtó védelmi módszerek illusztrációja. A dupla élek a teljes igényt továbbítják, míg a sima élek annak felét [J3].

üzemelnek. Az egyszeres hibák védelmére az $1+1$ mellett az úgynevezett Diversity Coding (DC) [26] a legjobb jelölt. A DC lényege, hogy 3 független útvonalon küldi a felhasználói adatokat, mégpedig oly módon, hogy az egyik útvonalon az adat egyik felét (jelöljük ezt továbbra is A-val), a másikon a másik felét (jelöljük ezt B-vel), míg a harmadikon a két adatrész kizáró vagy-ját ($A \oplus B$). A módszer az $1+1$ erőforrás pazarlását képes csökkenteni, amellett, hogy továbbra is robusztus módon azonnali helyreállítást biztosít [7]. A DC hátránya, hogy nagy összefüggőséget követel meg (3-szorosan összefüggő hálózatokban hatékony), amire a valós hálózatok esetén ritkán van példa. A különböző módszerek struktúrájának szemléltetése a 3. ábrán látható.

A [27] [28] is az egyszeres hibák védelmének problémáját járta körül. Bemutatásra került, hogy abban az esetben, ha a felhasználói adatfolyam két részre való osztását engedjük meg és hálózati kódolást alkalmazunk, az *optimális (költséghatékony) robusztus* kódolás három, úgynevezett kódoló DAG (Directed Acyclic Graphs) megtalálására vezethető vissza [27]. Viszont a routing gráf (megbízható routing) megtalálása nyitott kérdés maradt. Ezzel a kérdéssel a [C1] cikkben foglalkoztunk.

Több lehetséges feladat létezik, összesen négy, annak függvényében, hogy a hálózatban vannak-e kapacitás korlátok, illetve, hogy mely csomópontok képesek az osztás és összevonás műveletére (2. táblázat). Bebizonyítottuk, hogy abban az esetben, ha a linkeken nincsen kapacitáskorlát, valamint az osztó és összevonó csomópontok tetszőlegesek lehetnek (ICAN), a probléma polinom időben megoldható (SRDC-IA), míg kapacitáskorlátos esetben (CCAN) ez még nyitott kérdés. Valamint az is bizonyítást nyert, hogy ha az osztásra és összevonásra csak előre meghatározott csomópontok képesek (CCCN), akkor a probléma NP-teljes. A különböző feladatokat, az azokra

javasolt megoldásokat és azok komplexitását a 2 táblázat foglalja össze, ahol $\mathcal{P}$ a lehetéses osztó, míg $\mathcal{M}$ a lehetséges összevonó csomópontok halmazát jelöli. A feladatokról bővebben:

- ICAN (Infinte Capacity and All Node capabilities): Ebben az esetben nincs sem kapacitás-, sem csomóponti korlátozás. A kapacitás végtelensége annyit tesz, hogy minden linken rendelkezésre áll az igénynek megfelelő szabad kapacitás. Valamint minden csomópont képes a folyam összevonás és osztás műveletére. Ez egy olyan esetnek felel meg amikor a hálózat nem túlterhelt és minden csomópontban a csomóponti képességek frissítésre kerültek, vagyis az összes csomópontban megtörtént a megfelelő eszközcsere, illetve/vagy szoftverfrissítés.

- ICCN (Infinte Capacity and Constrained Node capabilities): Ebben az esetben továbbra sincsenek kapacitás korlátok, de csomóponti korlátozás már van, vagyis nem minden csomópont képes a folyam összevonás és osztás műveletére. Ez egy olyan esetnek felel meg, amikor a hálózat nem túlterhelt de nem minden csomópontban történt meg a csomóponti képességek frissítése, vagyis nem az összes csomópontban történt meg a megfelelő eszközcsere, illetve/vagy szoftverfrissítés. Ilyen eset akkor fordulhat elő, ha a szolgáltató fokozatosan tér át az új technológiára (incremental deployment).

- CCAN (Constrained Capacity and All Node capabilities): Ebben az esetben kapacitás korlátok vannak a hálózatban, de csomóponti korlátozások nincsenek, vagyis minden csomópont képes a folyam összevonás és osztás műveletére. Ekkor a hálózat túlterhelt, de minden csomópontban a csomóponti képességek frissítésre kerültek.

- CCCN (Constrained Capacity and Constrained Node capabilities): Ebben az esetben mind kapacitás-, mind csomóponti korlátok vannak a hálózatban. Tehát a hálózat túlterhelt és nem minden csomópontban történt meg a csomóponti képességek frissítése.

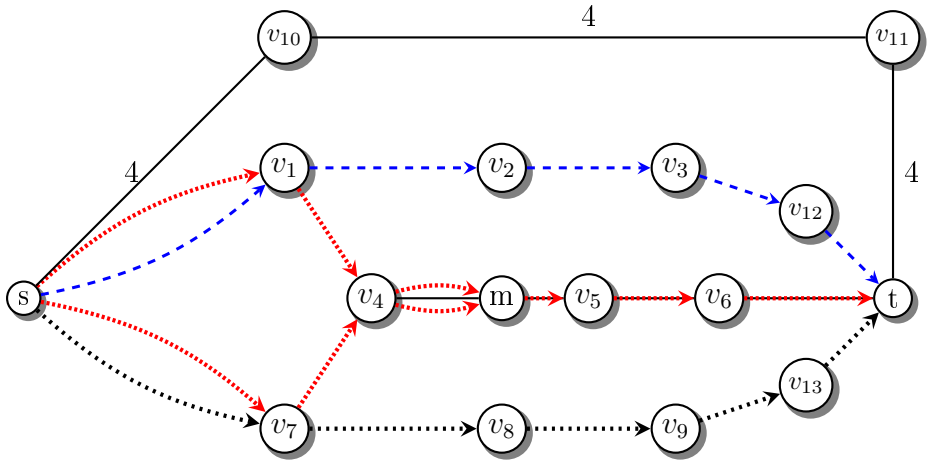
Érdemes kiemelni, hogy a kapacitás korlát és csomópont kényszer nélküli esetre (ICAN) adott polinom idejű algoritmus kapacitás korlátos esetben (CCAN) nem

2. táblázat. SRDC összefoglaló táblázat. Az osztók halmazát $\mathcal{P}$-vel, míg az összevonók halmazát $\mathcal{M}$ jelöltük. A szürke hátterű eredmények már a szakirodalomban létező eredmények [C1], míg a fehér hátterű eredmények az új, a disszertációban tárgyalt eredmények.

	Kapacitás korlát nélkül	Kapacitás korlátos
	ICAN	*CCAN*
$\mathcal{P} = V$,	SRDC-IA	ILP + SRDC-CA
$\mathcal{M} = V$	Polinom időben	-
	ICCN	*CCCN*
$\mathcal{P} \subset V$,	ILP + SRDC-IC (Approximációs algoritmus)	ILP + SRDC-CC
$\mathcal{M} \subset V$	-	NP-teljes

használható, mivel a virtuális élek, vagyis a diszjunkt útpárokat helyettesítő élek (úgynevezett szigetek) visszamappelése az eredeti élekre nem egyértelműen kivitelezhető. Abban az esetben, ha nincsenek kapacitás korlátok, de nem mindegyik csomópont lehet osztó és összevonó (ICCN), akkor sem ad optimális megoldást az SRDC-IA.

Az ICCN feladatra egy approximációs algoritmust adok, vagyis a SRDC-IC-t. Az algorimus igen hasonló a [C1]-ben bemutatott SRDC-IA-hoz, de nem minden csomópont közé rakunk virtuális éleket, hanem csak azon a csomópontok közé, amelyek lehetnek osztók, illetve összevonók. Az SRDC-IC viszont nem ad optimális eredményt, erre látható egy példa a 4. ábrán: m-mel jelöljük azt a csomópontot amely lehet összevonó (a forrás és nyelő csomóponton kívül). Ebben a példában, ha DC-t használunk, amely három független utat keres, akkor a megoldás összköltsége 22 egység (szaggatott (5), pontozott (5) út és a $s-v_{10}-v_{11}-t$ (12)). Abban az esetben, ha az SRDC-IC algoritmust használjuk az $1+1$-gyel azonos megoldást kapunk, vagyis 20 (szaggatott $(2 \cdot 5)$ és pontozott $(2 \cdot 5)$) utakon két egység átvitele). Az optimális megoldás pedig 19 (szaggatott (5), pontozott (5) út sűrűn pontozott (9)), ahogy ez az ábrán is látható. Érdekesség, hogy ekkor a v_4 és az m csomópontok között érdemes két egységnyi kapacitást foglalni annak érdekében, hogy eljussunk az összevonó csomópontig. Tehát látszik, hogy az SRDC-IC ebben az esetben nem ad optimális megoldást, viszont approximálja azt, ahogyan az $1+1$ is.



4. ábra. SRDC-IA és DC korlátainak illusztrációja ICCN esetén. A forrás és nyelő csomóponton kívül m lehet a lehetséges osztó vagy összevonó. Az élköltségek egységnyiek, kivéve ha az élen más szerepel. A három optimális routing DAG összköltsége 19.

1.2. Tézis (SRDC Egészértékű programok). *[C1] A CCAN problémájára javasoltam egy gyors futást biztosító ILP-t. Valamint kiterjesztettem ezen ILP-t a csomóponti kényszeres esetre (CCCN).*

Természetesen a CCCN ILP futási ideje lényegesen hosszabb, mivel a csomóponti kényszereket is figyelembe kell venni.

1.3. Tézis (SRDC heurisztikák). *[C1] Javasoltam két heurisztikát, az SRDC-CA-t amely a CCAN problémára minden esetben megengedett megoldást szolgáltat (ha létezik). Továbbá, a SRDC-CC-t amely gyorsan és igen hatékony módon szolgáltat megoldást a legtöbb gyakorlati problémára.*

Tehát az SRDC-CA heurisztika minden esetben megoldást ad (amennyiben létezik megoldás), de abban az esetben viszont, ha csomóponti kényszerek is vannak a hálózatban (CCCN), ezen heurisztika már nem elégséges, hanem az SRDC-CC-t kell alkalmazni. Az SRDC-CC nem garantál minden esetben megoldást (mivel ennek eldöntése már magában NP-teljes feladat), de igen nagy valószínűséggel kiváló

eredményt produkál (lásd 5. ábra). A heurisztikáknál felhasználásra került a [28]-ben bizonyított állítás, miszerint az úgynevezett redukált kapacitású gráfban a három értékű folyam megtalálása egy, az egyszeres hibáknak *ellenálló* gráfot fog alkotni. Tehát bármelyik link hibája esetén a gráfban két értékű folyam átvihető (ebben az esetben az egy értékű folyam a felhasználói adat felének felel meg).

Az SRDC-CA herusztika futása során egy minimális költségű három értékű folyamot keresünk, egy speciális redukált kapacitású gráfban $G_{rs} = (V, E, \bar{c}, c_s)$. $G_{rs} = = (V, E, \bar{c}_s, c_s)$ pedig egy olyan gráf, ahol a redukált kapacitású gráfot [28] alakítjuk multigráffá. Vagyis azon élek mellé, amelyeken a redukált kapacitás 1.5 ($\forall e \in E$ ahol $\bar{c}(e) == 1.5$), kihúzunk még egy párhuzamos élt, mégpedig olyan módon, hogy az eredeti él e_r kapacitását egyre csökkentjük ($\bar{c}(e_r) = 1$), az új élnek e_n pedig a kapacitását fél egységre állítjuk ($\bar{c}(e_n) = 0.5$). Mindeközben az eredeti él, vagyis e_r költsége az eredeti marad vagyis $c(e_r)$, míg az új él költsége az eredet költség konstansszorosa lesz: $c(e_n) \cdot \alpha$.

Fontos, hogy α a hálózat sűrűségétől függő költség paraméter. Vagyis a hálózati topológia azon tulajdonságát veszi figyelembe, hogy sűrű hálózatok esetén nagyobb a valószínűsége annak, hogy 3 független út van a hálózatban (DC-hez hasonló optimális megoldás), míg ritka hálózatok esetén ennek valószínűsége alacsonyabb ($1+1$-hez hasonló megoldás). Természetesen a megoldás néha a DC vagy az $1+1$ megoldását adja vissza (vagyis ezek tekinthetők a két végletnek), éppen annak függvényében, hogy melyik a költség vagyis erőforrás hatékonyabb. De az igazán értékes, újat nyújtó és kapacitás-hatékony megoldások azok, amelyek esetén a DC tulajdonságait ki tudjuk bővíteni a kétszeresen összefüggő gráfokra is. A SRDC-CA megoldása tehát *egy, a három DAG-ot tartalmazó optimális routing gráf*, és nem két vagy három független útpár, mint a DC és $1 + 1$ esetén. A SRDC-CA heurisztika skálázható és gyors lefutási idejű, így nagy hálózatok esetén használata igen előnyös lehet. A heurisztika hatékonyságát szimulációkkal igazoltam [C1] (lásd 5. ábra). Ahogy már azt említettem korábban is, abban az esetben, ha az osztó- és összevonó csomópontok nem lehetnek tetszőlegesek, a probléma tovább nehezedik. Az ICCN probléma komplexitása még nyitott kérdés, de a CCCN bizonyítottan NP-teljes [C1]. Ezen SRDC-CC heurisztika is a [28]-ban definiált redukált kapacitású gráfra épül. Viszont nem három folyamot keresünk a hálózatban, hanem három utat, mégpedig 2 lépésben. A heurisztika

skálázható és gyors lefutási idejű, így nagy hálózatok esetén használata igen előnyös lehet. A heurisztika hatékonyságát szimulációkkal igazoltam, amelyek során az igen jó eredményeket produkált (lásd 5. ábra).

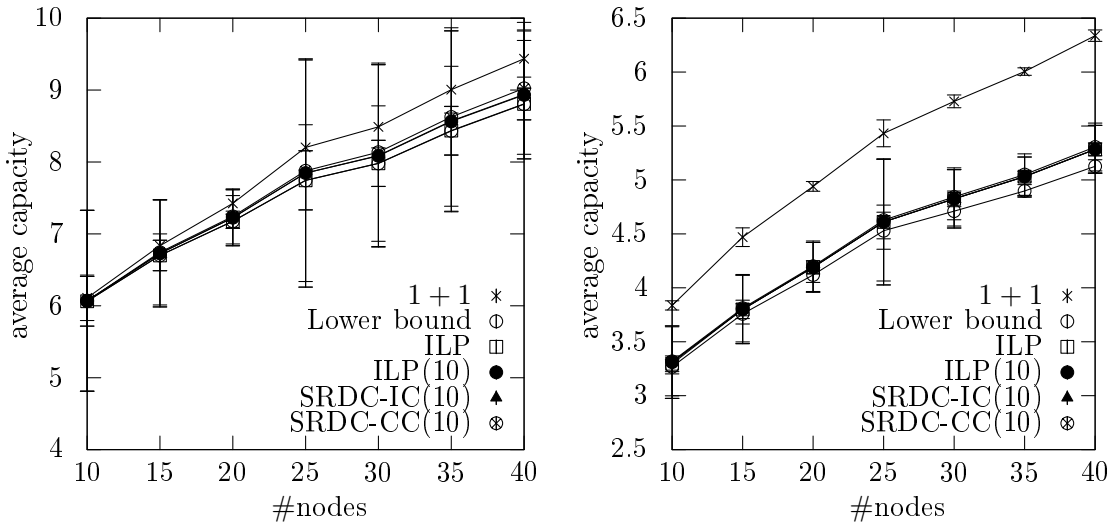
A megbízható routingok kinyerése mellett továbbá foglalkoztam annak approximálhatoságával. Tudjuk, hogy az adatok két részre való osztása esetén az $1+1$ 2-approximációja a megbízható routing problémának [6]. Beláttam, hogy a kapacitás korlát nélküli esetnek az $1+1$ a 4/3-os approximációja.

1.4. Tézis (SRDC approximáció). *Bebizonyítottam, hogy az* 1+1 *a kapacitás korlát nélküli eseteknek a 4/3-os approximációja. Valamint megmutattam egy gráfkonstrukció segítségével, hogy kapacitás korlátos esetben nem approximálja az* $1+1$*-et, vagyis a kapacitás foglalása közötti különbség tetszőlegesen nagy lehet. Az ICCN esetre javasoltam egy* 4/3*-os approximációs algoritmust.*

A bizonyítás arra épül, hogy az ICAN polinom idejű algoritmusa három út megtalálására vezethető vissza egy segédgráfban. Ennek segítségével az $1+1$ (2 útja) és a SRDC-IA (3 útja) között fennálló egyenlőtlenségek segítségével algebrai úton láttam be, hogy az $1+1$ az SRDC-ICAN problémának a 4/3-os approximációja. A kapacitás korlátos esetnél (CCAN) egy gráfkonstrukcióval szemléltettem, hogy nem approximálja azt, vagyis, abban az esetben, ha van kapacitás korlát a hálózatban, akkor konstruálható olyan gráf, amely esetén az SRDC tetszőlegesen jobban teljesít (lásd 6. ábra). Figyeljük meg, hogy az $1+1$ csak a dupla éleket használhatja, mivel ott van elegendő kapacitás, míg a SRDC-CCAN használhatja az egységnyi kapacitású éleket is. Így egy olyan gráfot kapunk, amelynél az n élszámtól függően az $1+1$ költsége 4+n, míg az SRDC-CCAN költsége 6 egység.

4.2. Azonnali helyreállítás biztosítása késleltetési kényszerek mellett egyszeres linkhibák esetén

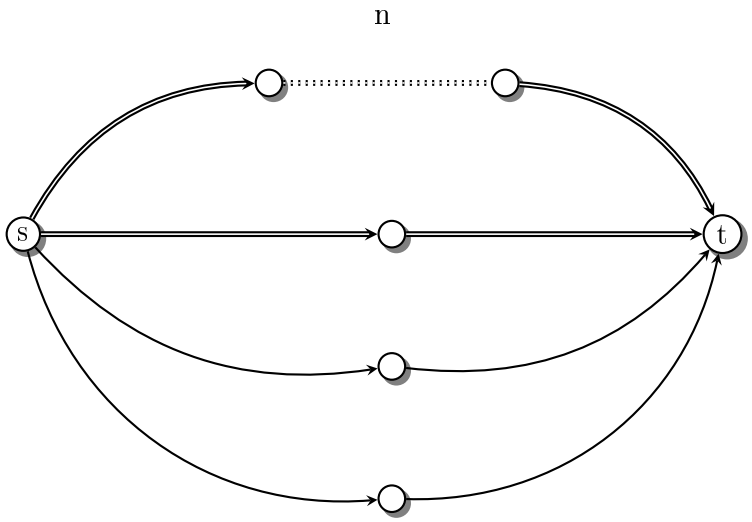
Az utóbbi időben az új alkalmazásoknak (pl. távsebészet, tőzsde, VoIP, stb) köszönhetően egyre jobban reflektorfénybe került a késleltetés kérdése az átviteli hálózatokban, mivel ezen új alkalmazások nem csak nagyfokú rendelkezésre állást és rugalmasságot követelnek meg, de igencsak késleltetés érzékenyek is. Ezen új követelmények a szolgáltatókat igen nagy kihívások elé állítják.



(a) Ritka hálózatok kapacitás foglalása csomóponti kényszerek mellett

(b) Sűrű hálózatok kapacitás foglalása csomóponti kényszerek mellett

5. ábra. Szimulációs eredmények végtelen kapacitás és csomóponti kényszerek mellett. A csomópontok 10% képes az osztás és összevonás műveletére.



6. ábra. CCAN approximálhatóságának ellenpéldája. Az $1+1$ csak a dupla éleket használhatja, míg az SRDC-IA bármely élt. Így az n (élek számának) növelésével a két megoldás közötti költség különbség tetszőlegesen növelhető.

Ezért én egy olyan módszert dolgoztam ki, amely nem csak *robusztus módon képes azonnali helyreállítást biztosítani, hanem egyben különböző késleltetési kényszereknek is képes megfelelni a* hiba bekövetkezése után is. Ehhez a *kapacitás korlát és csomópont kényszerek nélküli* SRDC (ICAN) késleltetés kényszerekkel való kibővítését vizsgáltam. Ezen új probléma a Delay Aware Routing with Coding vagyis a DARC.

Ezen módszer hasonlóan az $SRDC$-hez és DC-hez a felhasználói adatokat több útvonalon (vagy DAG-on) keresztül juttatja el a nyelőhöz. Ezért az első lépés a többutas forgalomirányítás, vagyis a multipath routinggal kapcsolatos késleltetési problémák vizsgálta volt. Ezen problémát már több hálózati rétegben igen alaposan vizsgálták, mivel a többutas forgalomirányítás igazi lehetőséget nyújt a megfelelő szolgáltatási szint, vagyis Quality-of-Service (QoS) megteremtéséhez. A több diszjunkt út által nagyobb átviteli kapacitás és könnyebb forgalom menedzsment valósítható meg, a megbízhatóság biztosítása mellett.

A mulitpath routing esetén két fő problémakör kapcsolódik a késleltetéshez [15]:

- Az úgynevezett QoS (Quality of Service) routing: ekkor a cél az utak, vagy azok összegének egy bizonyos késleltetési küszöb alatt való tartása, miközben az utak költségét minimalizáljuk. Ezen probléma már egyetlen egy út esetén is NP-teljes [8]. Eme probléma egy bizonyos QoS szint tartásához köthető, vagyis például, hogy a távsebészeti beavatkozás során ne legyen nagy a késleletetés az orvos és a gép mozgása között.

- Differenciál késleltetés kényszeres routing (Differential Delay Aware Routing): itt a különböző utak közötti késleltetés különbségére van megadva egy korlát, és emelett kerül a költségfüggvény minimalizálásra. Ezen probléma is NP-teljes, sőt egyetlenegy minimális költségű út megtalálása, ami kielégít egy maximum és minimum késleltetés küszöböt is, már NP-teljes [29]. A probléma a szükséges buffer méret problémájára vezethető vissza, mivel az adatok visszanyeréséig a gyorsabban beérkezett adatokat (adat részleteket) tárolni kell.

Mivel eddig DAGok esetén a késleltetés nem lett definiálva, ezért ez volt az első akadály, amelyet le kellett küzdeni. Ezután kezdhettem el érdemben foglalkozni a késleltetési kényszeres problémák bevezetésével és azok megoldásával. A routing DAG-ok késleltetésének definiálásánál felhasználtam, hogy a routing DAG-ok felbonthatók

3. táblázat. Végponttól-végpontig terjedő késleltetés különbsége a két alacsonyabb késleltetésű DAG között, tetszőleges egyszeres linkhiba esetén (wlog $\delta_{E_A} \leq \delta_{E_B} \leq \leq \delta_{E_{A\oplus B}}$).

disrupted \ inc. delay	$\emptyset$	E_A	E_B	$E_{A\oplus B}$
$\emptyset$	$\lvert\delta_{E_A} - \delta_{E_B}\rvert$	$\lvert\delta_{E_B} - \min\{\Delta_{E_A}, \delta_{E_{A\oplus B}}\}\rvert$	$\lvert\delta_{E_A} - \min\{\Delta_{E_B}, \delta_{E_{A\oplus B}}\}\rvert$	$\lvert\delta_{E_A} - \delta_{E_B}\rvert$
E_A	$\lvert\delta_{E_B} - \delta_{E_{A\oplus B}}\rvert$	–	$\lvert\Delta_{E_B} - \delta_{E_{A\oplus B}}\rvert$	$\lvert\delta_{E_B} - \Delta_{E_{A\oplus B}}\rvert$
E_B	$\lvert\delta_{E_A} - \delta_{E_{A\oplus B}}\rvert$	$\lvert\Delta_{E_A} - \delta_{E_{A\oplus B}}\rvert$	–	$\lvert\delta_{E_A} - \Delta_{E_{A\oplus B}}\rvert$
$E_{A\oplus B}$	$\lvert\delta_{E_A} - \delta_{E_B}\rvert$	$\lvert\Delta_{E_A} - \delta_{E_B}\rvert$	$\lvert\delta_{E_A} - \Delta_{E_B}\rvert$	–

utak és szigetek láncolatává, mégpedig olyan módon, hogy minden egyes sziget csak egyetlenegy kódoló DAG-nak lehet a része (lásd *SRDC*). Fontos, hogy a *DARC*-nál is elégséges a *kódolást, illetve dekódolást a forrás- és nyelő csomópontban elvégeznünk.* Tehát a hálózat belsejében nincsen szükség komplex műveletek elvégzésére. Így a módszer nem csak robusztus, de egyszerű is.

2. Tézis. *[C3, J2] Definiáltam a routing DAG-ok késleltetését. Bevezettem a DARC-QoS (Delay Aware Routing with Coding - Quality of Service) és DARC-DD (Delay Aware Routing with Coding - Differential Delay Aware Routing) problémákat. Beláttam, hogy mindkét probléma, vagyis a DARC-QoS és a DARC-DD is NP-teljes. Továbbá bebizonyítottam, hogy a DARC-DD probléma nem approximálható n^ϵ-on belül bármilyen $\epsilon < 1$ esetén, Hamilton gráfokban. Felírtam a feladatok megoldásához szükséges ILP-ket és a DARC- DD megoldására adtam két hatékony heurisztikát.*

Annak érdekében, hogy az ilyen változásokat kezelni tudjam, minden egyes szigethez két értéket rendeltem:

- hibamentes állapotban mennyi a legkisebb késleltetés az adott routing DAG-ban (d^I_{min}) és
- meghibásodás esetén menyivel nő a késleltetés (Δ^I).

Mivel minden egyes routing DAG utak és szigetek sorozatából tevődnek össze, így a teljes késleltetést felírhatjuk a következő formában, hibamentes állapotban:

$$\delta_{E_j} = \sum_{P \in \mathcal{P}_{E_j}} \sum_{e \in P} d(e) + \sum_{I \in \mathcal{I}_{E_j}} d^I_{min} \tag{3}$$

Hiba esetén pedig:

$$\Delta_{E_j} = \delta_{E_j} + \max_{I \in \mathcal{I}_{E_j}} \Delta^I \tag{4}$$

Vagyis hibamentes esetben a minimum késleltetések (utak és szigetek) összegének felel meg az adott E_j routing DAG-hoz tartozó késleltetés, míg hiba esetén azt feltételezzük, hogy a legnagyobb késleltetésnövelést okozó sziget esik ki. Fontos, hogy mind az ILP, mind a heurisztikák egy olyan transzformált gráfban futnak, amelyben a szigeteket virtuális élek reprezentálják. Vagyis kibővítjük a [C1]-ben bemutatott gráf transzformációt, tehát létrehozunk egy $\widehat{G_d} = (V, \widehat{E}, \widehat{c})$ irányított multi-gráfot $\widehat{E}$ élhalmazzal, ahol $\widehat{E}$-ben szerepel minden $e \in E$, valamint mindegyik $\forall(u, v)$ csomópontpár között hozzáadunk egy úgynevezett virtuálist élt, mégpedig oly módon, hogy ezen él költsége az u és v közötti minimális költségű diszjunkt útpár összegének költségével egyenlő ($\widehat{c}(e_n) = \text{cost}(u, v)$). Továbbá, minden egyes élhez hozzárendeljük a $d^I_{min}(e)$ és $\Delta^I(e)$ értékeket. Természetesen, ha nem virtuális élről van szó, akkor $d^I_{min}(e) = d(e)$ és $\Delta^I(e) = 0$. Egy ilyen transzformált gráfban a QoS és DD probléma visszavezethető utak keresésére. Kiemelendő, hogy mivel ezen utakat egy transzformált gráfban kell megtalálni, az eredeti feladatok NP-teljessége nem öröklődik. Így ezen problémák komplexitása nyitott kérdés volt.

2.1. Tézis (QoS komplexitás). *[C3,J2] Bebizonyítottam, hogy a DARC-QoS probléma NP-teljes.*

Az NP-teljesség bizonyítás a háromfelé osztás problémájára (Three-Way Partition [8]) vezethető vissza, amely egy ismert NP-teljes probléma.

2.2. Tézis (DD komplexitás). *[C3,J2] Bebizonyítottam, hogy a DARC-DD probléma NP-teljes.*

Ezen probléma esetén is beláttam, hogy NP-teljes, a bizonyítás egy gráf konstrukció segítségével a leghosszabb út problémára vezethető vissza [8].

Fontos különbség a QoS és DD probléma között, hogy a DD esetén igen szigorú *késleltetés különbségi* kényszereknek kell megfelelni. Így a megoldás költsége másodlagossá válhat, vagyis egy nem megfelelően felírt ILP esetén kialakulhatnak független komponensek, illetve hurkok annak érdekében, hogy ezen késleltetési különbségi

kényszerek ki legyenek elégítve. Természetesen ezek nem valódi megoldások, mivel, bár szigorúan nézve, a kényszereket kielégíthetik, de a feladat értelmét veszti. Ezért a független komponensek kiküszöbölésére az ILP-ben a feszültség analízis módszerét használtam, míg a hurkok létrejöttét különböző folyamkényszerekkel zártam ki. A DARC-DD esetén a két gyorsabb routing DAG közötti késleltetés különbséget akarjuk egy bizonyos késleltetési küszöb alatt tartani. Ekkor az összes, a 3. táblázatban megadott kényszert meg kell vizsgálnunk.

2.3. Tézis (Egészértékű programok). *[C3, J2] Az új, a DAG-okban bevezetett késleltetés definíciója segítségével, felírtam a feladatok (QoS és DD) megoldásához szükséges ILP-ket.*

2.4. Tézis (Heurisztikák). *[C3, J2] Az ILP lassabb futási ideje miatt javasoltam két (költség és késleltetés alapú) gyors heurisztikát, amelyek a SPLIT [11] módszert képessé tették a probléma megoldására.*

Az ILP lassú futási ideje miatt javasoltam két gyors heurisztikát. A módszer első lépése a k (először $k = 3$) éldiszjunkt út megtalálása. Második lépésként az algoritmus identifikálja azt a közbülső csomópontot, amelyen a legtöbb út áthalad, jelöljük ezt a csomópontot v_x-szel. Majd azon utakat, amelyek áthaladnak v_x-en, feldaraboljuk szegmensekké ($s - v_x$, valamint $v_x - t$ szegmensekké, ahol s a forrást míg t a nyelőt jelöli). Ezeket a szegmenseket úgy ragasztjuk össze, hogy a legnagyobb késleltetésű $s - v_x$ szegmenshez a legkisebb késleltetésű $v_x - t$ szegmenst rendeljük hozzá. A felhasznált éleket töröljük, majd visszatérünk a második lépésre mindaddig, amíg a gráf el nem fogy. Az így kialakult utak halmazából tripleteket hozunk létre és megvizsgáljuk, hogy azok kielégítik-e a késleltetési kényszereket. Ha igen, akkor ezt elmentjük a lehetséges megoldások közé.

A k értékét iteratívan növeljük, amíg lehet, vagyis van k független útpár. Az algoritmus végén a lehetséges megoldások közül kiválasztjuk a legkisebb költséggel rendelkező tripletet. Ne feledjük, hogy természetesen ez a módszer is a transzformált gráfon fut.

2.5. Tézis (DD approximálhatóság). *[C3, J2] Bebizonyítottam, hogy a DARC-DD probléma nem approximálható n^ϵ-n belül bármilyen $\epsilon < 1$ esetén Hamilton grafokban.*

A bizonyítás az adott konstansnál hosszabb utak megtalálásának approximálhatóságára vezethető vissza Hamilton gráfokban. Ezt a technikát Srivastava is használja a [33] 4-C alfejezetben, annak érdekében hogy belássa, hogy a DDR (Differential Delay Routing) nem approximálható n^{ϵ}-n belül bármilyen $\epsilon < 1$ esetén Hamilton gráfokban.

4.3. Védelmi kapcsoláson alapuló hibalokalizáció

Disszertációm első részében olyan védelmi módszereket javasoltam, amelyek a hálózati kódolás segítségével robusztus módon képesek garantálni az azonnali helyreállítást, egyszeres és többszörös hibák esetén is. Ezen módszerek egyrészt megtartották az $1+$ $+1$ jó tulajdonságait (robusztus, egyszerű és azonnali helyreállítást biztosít), másrészt lényegesen csökkentették az ehhez szükséges erőforrásigényt (persze hozzárendelt védelmekről lévén szó, ezeknek a módszereknek is jelentős az erőforrásigényük). Annak érdekében, hogy ezen erőforrásigényt tovább csökkenthessük, a megosztott védelmek felé kell fordulnunk. Eddig a megosztott védelmeket az alacsony erőforrásigény és lassú helyreállítási idő jellemezte (vagyis azonnali helyreállítást nem biztosítottak), de a helyreállítási idő csökkentése már régóta kutatási téma. Vegyük észre, hogy míg a robusztus védelmi módszereknél ($1+1$, SRDC) $t_n = t_c = t_p = 0$, ami miatt a helyreállítási idő $t_R < 20-30$ ms, addig megosztott, vagyis hibafüggő védelmek esetén ezen idők (főleg a hibaterjesztési és kapcsolási idők) igen jelentősen hozzájárulnak a teljes helyreállítási ciklushoz. Annak érdekében, hogy a helyreállítási idő csökkenjen (t_R), a hibadetektálás nem hagyatkozhat az elektromos réteg lassú üzenetváltásaira. Ennek kiküszöbölésére bevezetésre kerültek az úgynevezett felügyeleti fényutak, az optikai *monitorozó utak*, vagyis *m-trailek* [5]. A felügyeleti fényutakkal történő hibalokalizációt már régóta kutatják. Először a centralizált, majd a lokális hibalokalizáció került előtérbe. A módszerek lényege, hogy minden fényút végén található egy monitorozó eszköz (monitor), mely a fényút állapotát ellenőrzi, és hibajelet generál, ha annak megszakadását érzékeli. Ezek alapján pedig vagy a hibamenedzser központilag [22,35], vagy minden egyes csomópont lokálisan [4,21,30,31] *egyértelműen* lokalizálja a hibát a begyűjtött jelzésekből. Persze a helyreállítási idő csökkentése szempontjából a lokális hibalokalizáció sokkal izgalmasabb téma.

A lokális hibalokalizáció első lépése az volt, amikor *egy adott csomópont* képessé vált a lokálisan elérhető m-trailek státuszaiból egyértelműen megállapítani az összes

egyszeres linkhibát. Ezen módszer az úgynevezett lokális egyértelmű hibalokalizáció *(Local Unambigous Failure Localization* röviden *L-UFL)*. Természetesen a kapcsolást végző csomópontokat továbbra is az elektromos rétegen keresztül kellett értesíteni az elvégezendő kapcsolásról, ami megint csak lassú helyreállítást eredményezett. A következő lépés az volt, amikor a hálózat minden egyes csomópontja L-UFL képessé vált, vagyis *minden egyes csomópont* képessé vált az egyértelmű lokális hibalokalizációra (*Network-Wide Local - UFL* röviden *NWL-UFL*) [30]. Ez a módszer már teljesen optikai, vagyis nem szükséges elektromos rétegbeli üzenetváltás sem a hibalokalizációs, sem a hibaértesítési fázisához. Sőt, mivel mindegyik csomópont képes minden egyszeres hibát észlelni, így a szükséges kapcsolás azonnal megtörténhet.

A módszer nagy hátránya, hogy az m-trailek igen hosszúak (feszítőfák [30]), ami implementálási gondokat eredményezett, valamint növelte a hibamonitorozási késleltetést. Továbbá vegyük észre, hogy minden egyes csomópont minden egyes hibát képes észlelni annak ellenére, hogy rengeteg linkhiba helyreállításában nem vesz részt.

Ezeknek a hátrányoknak a kiküszöbölését vette célba a Global Neighborhood Failure Localization (G-NFL) [21, 31]. A módszer lényegi ötlete, hogy minden egyes csomópontban csak egy kis élhalmaz hibáit lokalizáljuk egyértelműen, rövid m-trailek segítségével, vagyis azoknak a linkeknek a halmazát amelyek esetén az adott csomópontnak védelmi kapcsolást kell végeznie. Ezen linkek halmaza pedig az adott csomópont *szomszédsága*. Az új módszer komoly áttörésnek bizonyult, mivel csökkenti a lokálisan lokalizálandó linkek számát, ezzel drasztikusan és automatikusan csökkentve a módszer erőforrásigényét. Viszont fontos kiemelni, hogy a G-NFL-nél továbbra is *egyértelműen kell lokalizálni* a szomszédsághoz tartozó linkhibákat.

Szakítva az egyértelmű hibalokalizáció paradigmájával, disszertációmban egy teljesen új keretrendszert javasoltam, amely lokálisan lokalizálja a linkhibákat, de nem mindegyiket egyértelműen, hanem a *védelmi kapcsolásnak megfelelő* módon. Ez azt jelenti, hogy, ha például két különböző linkhibához ugyanaz a védelmi kapcsolás tartozik, akkor a módszer nem feltétlenül lokalizálja a két hibát egyértelműen (különkülön), hanem elég tudnunk, hogy egy adott csoporthoz tartozó valamelyik link sérült meg, annak érdekében, hogy a megfelelő védelmi kapcsolást végre tudjuk hajtani. Ezen új keretrendszer az Advanced Global Neighborhood Failure Localization (AG-NFL). Ezzel az új módszerrel nagyságrendileg sikerült tovább javítani az erőforrásigényt.

A 7. ábrán látható egy illusztratív példa, amely szemlélteti a paradigmaváltás hatásait. A 7(a). mutatja az üzemi (W_1) útvonalhoz tartozó két védelmi szegmenst $P_1^{(v_1 \to v_3)}$-t és $P_1^{(v_3 \to v_4)}$-t. Abban az esetben, ha (v_1, v_2) vagy (v_2, v_3) link kiesik, $P_1^{(v_1 \to v_3)} - - t$ kell aktiválni, míg (v_3, v_4) kiesése esetén $P_1^{(v_3 \to v_4)}$-t. A 7(b). ábrán az AG-NFL megoldás látható. Míg ebben az esetben elegendő két m-trail a megfelelő hibalokalizációhoz, vagyis a megfelelő kapcsolás elvégzéséhez, addig a G-NFL megoldás esetén a szükséges m-trailek száma 5 (7(c). ábra). Láthatjuk, hogy az AG-NFL a v_6-os csomópontban nem különbözteti meg (v_1, v_2) és (v_2, v_3) meghibásodását, mert mindenképpen ugyanazt a védelmi kapcsolást kell elvégezni, vagyis $P_1^{(v_1 \to v_3)}$-t. Az is megfigyelhető, hogy még (v_1, v_5) és (v_5, v_6) hibáját sem különbözteti meg (v_1, v_2) és (v_2, v_3)-től. Ez azért lehetséges, mivel az AG-NFL megenged úgynevezett *„nem diszruptív"* téves kapcsolást is, vagyis abban az esetben, ha (v_1, v_5) vagy (v_5, v_6) kiesik, elvégezhetjük a $P_1^{(v_1 \to v_3)}$ kapcsolást a v_6 os csomópontban, de mivel úgyis csak előre lefoglalt védelmi kapacitást kapcsolunk, ezért semmi nem történik csak kapcsolási energia vész kárba. Természetesen, a kapcsolás során üzemi útvonalat semmiképp nem szakíthatunk meg, így csak bizonyos csomópontok esetén engedélyezett az ilyen jellegű összevonás. Összefoglalva, az AG-NFL esetén 5 helyett 2 m-trail elégséges a megfelelő védelem megvalósításához, vagyis az erőforrás nyereség több, mint 50%.

Tehát elmondható, hogy a módszer két elvre épül:

(i) Abban az esetben, ha pontosan ugyanaz a védelmi kapcsolás tartozik két vagy több linkhez (az adott szomszédságban), akkor ezeket a hibákat nem kell megkülönböztetni, vagyis egyértelműen lokalizálni, hanem elég összevonva. Ezek a linkek *üzemi szegmenseket* alkotnak, erre példa 7(a). ábrán a v_6 szempontjából (v_1, v_2) és (v_2, v_3) linkek.

(ii) Az olyan link hibákat, amelyek az adott csomópont szomszédságban nem járnak üzemi útvonal megszakítással, vagyis amely linkekhez csak *„nem diszruptív" kapcsolások tartoznak* (a védelmi kapcsolás aktiválása nem okozza egy üzemi útvonal megszakítását sem) nem kell megkülönbözetni a szomszédságon kívül eső linkektől. Erre jó példa a (v_1, v_5) és (v_5, v_6), amelyeket nem kell megkülönbözetni $(v_1, v_2), (v_2, v_3)$ linkektől a v_6 os csomópontban (7(a). ábra).

Ezen elvek alapján a szomszédság minden egyes csomópontban két részre oszlik,

4. táblázat. Az m-trail újrakonfigurálásának összefoglalása az adatréteg megváltozásának függvényében [J4] [31]

	SOD-IO	SOD-O	LOD-IO	LOD-O
W-LP felépítés	X	X		
W-LP bontás	X		X	

vagyis az egzakt és approximált identifikációt igénylő linkek csoportjára. Ezzel az új, még finomabb osztályozással szignifikánsan csökkenteni tudjuk az erőforrásigényt bármely megosztott védelem esetén. De ami még fontosabb, lehetővé tehetjük tetszőleges megosztott védelem számára az erőforrástakarékos azonnali helyreállítást (gyors OXC konfigurációt feltételezve [36,37]). Természetesen a védelem robusztussága semmiképpen sem valósulhat meg.

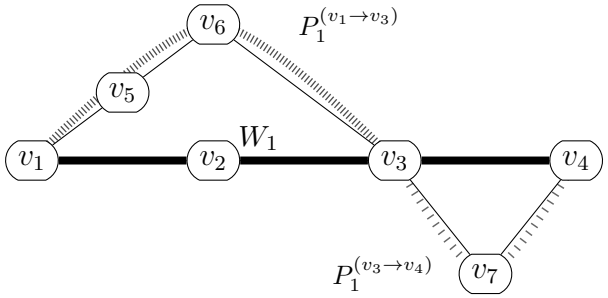
Az AG-NFL adatréteg függőségét több szempontból is megvizsgáltam [31]. Egyrészt azt az esetet vizsgáltam, amikor az üzemi utak nem használhatók monitorozásra, ezt -O-val jelöltem. Azt az esetet, amikor felhasználhatóak az üzemi utak a hibamonitorozásra, -IO-val jelöltem. Másrészről attól függően, hogy éppen az aktuális forgalmi mátrixra (Strictly On-Demand (SOD)), vagy az összes, a jövőben előforduló forgalomra optimalizálunk (Loosely On-Demand (LOD)), másik faladatot kapunk [31].

Így tehát összesen négy különböző feladatot vizsgáltam (lásd 4. táblázat).

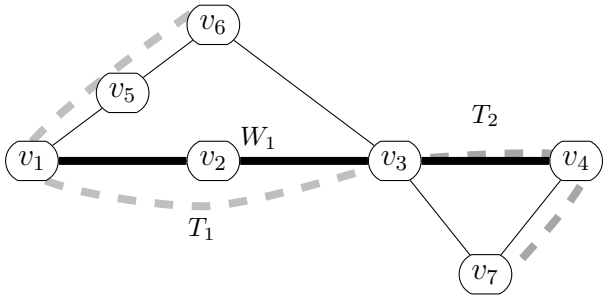
3. Tézis. *[C4, J4] Javasoltam egy új hibalokalizációs keretrendszert, az úgynevezett Advanced Global Neighborhood Failure Localization-t, amely lehetővé teszi a rendkívül gyors helyreállítást még a megosztott védelmi módszerek esetén is. Bevezettem az úgynevezett tiltott linkpárok fogalmát, amelynek segítségével több m-trail feladatot általásítottam, javítva a megoldásuk komplexitásán. A probléma megoldására egy új heurisztikát javasoltam. Beláttam, hogy az AG-NFL SOD-IO és LOD-IO NP-teljes.*

3.1. Tézis (Általánosított koncepció). *[C3, J4] Bevezettem az úgynevezett tiltott linkpárok fogalmát, amelynek segítségével több m-trail feladatot általásítottam, javítva a megoldásuk komplexitásán. A probléma megoldására egy új heurisztikát javasoltam.*

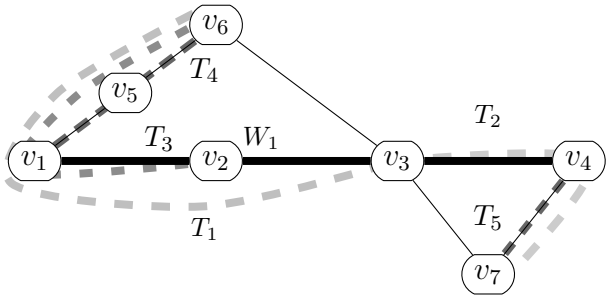
3.2. Tézis (AG-NFL komplexitás). *[C3, J4] Beláttam, hogy az AG-NFL SOD-IO és LOD-IO NP-teljes. A probléma megoldására egy új heurisztikát javasoltam.*



(a) W-LP W_1 és a hozzá tartozó P-LPs.



(b) S-LPs T_1 és T_2.



(c) G-NFL megoldás (hárommal több S-LPs szükséges).

7. ábra. A W_1 üzemi út és a T_1, T_2 S-LPs elég információt szolgáltatnak ahhoz, hogy a megfelelő védelmi kapcsolás megtörténjen, míg a G-NFL esetén még három m-trailre (T_3, T_4, T_5) van szükség az egyértelmű hibalokalizációhoz. A köztes csomópontok is képesek a fényutak monitorozására.

Az NP-teljesség bizonyítása a Hamilton $s-t$ út [8, Problem GT39] problémájára támaszkodik,vagyis egy gráf konstrukció segítségével sikerült belátnom, hogy az AG-NFL NP-teljes. A probléma megoldására adtam egy gyors és hatékony heurisztikát, aminek az alapja a Dijkstra algoritmus.

Az 5. táblázatban bemutatott szimulációs eredményekből is látható, hogy ezen új keretrendszer segítségével az erőforrásigény jelentősen, akár 50%-kal is csökkenthető. Természetesen az elérhető nyereség függ a probléma mivoltától (SOD vagy LOD, -IO vagy -O).

5. Összefoglalás

Disszertációm első felében hálózati kódolás alapú új módszereket javasoltam, amelyek megfelelnek a új hálózati trendeknek és képesek megfelelni az új komplex QoS (Quality of Service) követelményeknek. Ez azt jelenti, hogy képesek *robusztus módon azonnali helyreállítást* biztosítani, valamint hiba után is eleget tesznek késleltetési kényszereknek alacsony erőforrásigény mellett. Ezen új módszereket összehasonlítottam a jelenleg leggyakrabban használatos technológiákkal. **A módszerek valós *hálózatban is bizonyítottak*, vagyis implementálásra kerültek egy európai méretű SDN hálózatban, ahol a gyakorlataban igazolták hatékonyságukat.**

Disszertációm második felében a hibalokalizáció kérdésével foglalkoztam. Vagyis, olyan új lokális hibalokalizációs módszert javasoltam, amely lehetővé teszi az igen gyors helyreállítást megosztott védelmi módszerek esetén is (t_R számottevően csökken), valamint jelentősen csökkenti a lokalizációhoz szükséges erőforrás igényt.

5. táblázat. Valós hálózatok szimulációs eredményei (Internet Topology Zoo [38]). SOD esetén a csomópont párak $s-t$ 30%-a között van (aktív) forgalom. LOD esetén az üzemi utaknak ugyanaz a 30%-a aktív, míg a megoldást a forgalom 100%-ra számoltuk [J4].

Network topology			$\frac{\|\mathcal{T}\|}{\|E\|}$								b							
			SOD-IO		SOD-O		LOD-IO		LOD-O		SOD-IO		SOD-O		LOD-IO		LOD-O	
Name	$\|V\|$	$\|E\|$	G-NFL	AG-NFL	G-NFL	AG-NFL	G-NFL	AG-NFL	G-NFL	AG-NFL	G-NFL	AG-NFL	G-NFL	AG-NFL	G-NFL	AG-NFL	G-NFL	AG-NFL
Abilane	11	14	2.14	1.14	3.71	2.92	2.14	1.57	3.71	3.00	13	8	26	23	13	10	26	22
Germany	17	25	2.20	1.32	3.48	2.60	2.52	1.48	3.96	3.12	26	19	45	39	29	19	49	43
BtEurope	17	30	1.50	0.86	2.90	2.20	2.03	1.63	3.36	2.90	25	13	52	39	31	24	56	48
AS6461	17	37	2.16	0.97	3.56	2.37	2.59	1.75	3.97	3.02	44	19	75	50	53	34	83	63
Inter.MCI	18	32	2.53	1.03	4.12	2.75	3.09	1.81	4.59	3.68	38	16	70	50	45	27	74	63
AS1755	18	33	1.60	0.81	2.72	2.15	2.54	1.90	3.78	3.24	30	17	54	41	42	32	68	59
ChinaTelc	20	44	1.56	0.77	3.68	3.20	5.15	2.56	6.93	4.75	37	18	86	73	105	52	145	101
AS3967	21	36	3.19	1.94	5.02	4.02	3.41	2.16	5.38	4.52	52	33	89	71	55	36	94	79
BellSouth	21	36	0.75	0.36	2.27	2.02	0.75	0.55	2.27	2.11	16	7	54	48	16	10	54	48
AT&T	22	38	1.76	0.86	4.05	3.34	2.50	1.73	4.71	4.23	33	17	76	60	48	35	90	80
NSF	26	43	3.00	2.00	5.60	4.76	3.65	2.97	6.32	5.79	55	38	106	91	65	54	116	107
BICS	27	42	2.88	1.07	5.42	4.40	3.42	2.02	5.90	4.85	50	22	105	83	58	37	110	89
AS3257	27	64	2.34	1.39	4.46	3.50	2.68	1.85	4.79	3.98	78	44	150	115	88	56	157	125
AS1239	30	69	2.39	1.01	5.49	3.98	3.94	2.14	7.18	5.37	80	35	181	130	121	65	226	164
Arnes	31	47	2.17	1.12	5.08	3.93	2.53	1.70	5.29	4.48	47	24	112	86	52	36	112	95
Geant	31	49	2.65	1.69	5.32	4.20	3.18	2.28	5.73	5.02	56	39	120	97	64	48	125	111
Italy	33	56	1.69	0.96	3.96	3.37	1.76	1.17	4.00	3.71	47	30	116	99	47	35	116	105
BtNAmer.	36	76	3.21	2.01	6.93	5.48	4.03	2.69	7.63	6.22	109	68	239	184	134	87	257	204
BellCan.	39	55	3.25	2.12	7.76	5.80	3.56	2.45	8.16	6.20	62	45	156	121	68	53	163	129

Hivatkozások

[1] CPLEX. http://www.ilog.com/products/cplex/.

[2] GUROBI. Gurobi Optimization, Inc http://www.gurobi.com/products/gurobi-optimizer/gurobi-overview/.

[3] Lemon: A c++ library for efficient modeling and optimization in networks. Technical report, http://lemon.cs.elte.hu.

[4] S. Ahuja, S. Ramasubramanian, and M. Krunz. Single-link failure detection in all-optical networks using monitoring cycles and paths. *IEEE/ACM Transactions on Networking (TON)*, 17(4):1080–1093, 2009.

[5] P. Babarczi, J. Tapolcai, and P.-H. Ho. Adjacent link failure localization with monitoring trails in all-optical mesh networks. *IEEE/ACM Transactions on Networking*, 19(3):907–920, 2011.

[6] G. Brightwell, G. Oriolo, and F. B. Shepherd. Reserving resilient capacity in a network. *SIAM journal on discrete mathematics*, 14(4):524–539, 2001.

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

[8] Garey, Michael R and Johnson, David S *Computers and intractability: a guide to NP-completeness*, WH Freeman New York, 1979.

[9] W. Grover, J. Doucette, M. Clouqueur, D. Leung, and D. Stamatelakis. New options and insights for survivable transport networks. *IEEE Commun. Mag.*, 40(1):34–41, Jan. 2002.

[10] 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.

[11] S. Huang, C. Martel, and B. Mukherjee. Survivable multipath provisioning with differential delay constraint in telecom mesh networks. *IEEE/ACM Transactions on Networking*, 19(3):644–656, 2011.

[12] L. Huawei Technologies Co. White paper on technological developments of optical networks. *white paper,Huawei Technologies Co., Ltd*, 2016.

[13] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot. Characterization of failures in an ip backbone. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2307–2317. IEEE, 2004.

[14] D. Papadimitriou and E. Mannie. Analysis of generalized multi-protocol label switching (gmpls)-based recovery mechanisms (including protection and restoration). Technical report, 2006.

[15] H. F. Salama, D. S. Reeves, and Y. Viniotis. A distributed algorithm for delay-constrained unicast routing. In *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, volume 1, pages 84–91. IEEE, 1997.

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

[17] Musumeci, F. and Tornatore, M. and Pattavina, A. A Power Consumption Analysis for IP-Over-WDM Core Network Architectures. *Optical Society of America, Journal of Optical Communications and Networkin*, 4(2):108–117, 2012

[18] Y. Yano, T. Ono, K. Fukuchi, T. Ito, H. Yamazaki, M. Yamaguchi, and K. Emura. 2.6 Terabit/s WDM transmission experiment using optical duobinary coding. In *Optical Communication, 1996. ECOC'96. 22nd European Conference on*, volume 5, 1996.

[19] Konstantinos N. Georgakilas, Kostas Katrinis, Anna Tzanakaki, and Ole B. Madsen. Impact of Dual-Link Failures on Impairment-Aware Routed Networks. In *Transparent Optical Networks, 2010. Proceedings of 2010 12th International Conference on*. IEEE, 2010.

[20] A. Kotb and K. E. Zoiros. Performance analysis of all-optical xor gate with photonic crystal semiconductor optical amplifier-assisted mach–zehnder interferometer at 160 gb/s. *Optics Communications*, 402:511–517, 2017.

[21] J. Tapolcai, P.-H. Ho, P. Babarczi, and L. Rónyai. On achieving all-optical failure restoration via monitoring trails. In *Proc. 32nd IEEE INFOCOM*, pages 380–384, April 2013.

[22] J. Tapolcai, P.-H. Ho, L. Rónyai, P. Babarczi, and B. Wu. Failure localization for shared risk link groups in all-optical mesh networks using monitoring trails. *IEEE/OSA Journal of Lightwave Technology*, 29(10):1597–1606,

[23] J. Strand, AL Chiu, and R. Tkach. Issues for routing in the optical layer. *IEEE Communications Magazine*, 39(2):81–87, 2001.

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

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

[26] Ayanoglu, Ender and Chih-Lin, I and Gitlin, Richard D and Mazo, James E. Diversity coding for transparent self-healing and fault-tolerant communication networks *IEEE Transactions on Communications*, 41(11):1677–1686, 1993.

[27] Babarczi, Peter and Tapolcai, Janos and Rónyai, Lajos and Médard, Muriel Resilient Flow Decomposition of Unicast Connections with Network Coding *Proc. IEEE Intl. Symp. on Information Theory (ISIT)*,:116-120, 2014.

[28] Rouayheb, S. and Sprintson, A. and Georghiades, C. Robust Network Codes for Unicast Connections: A Case Study *IEEE/ACM Transactions on Networking*, 19(3):644–656, 2011.

[29] Ahuja, Satyajeet and Krunz, Marwan and Korkmaz, Turgay Optimal path selection for minimizing the differential delay in Ethernet-over-SONET, *Elsevier, Computer Networks*, 50(13):2349–2363, 2006.

[30] J. Tapolcai, P.-H. Ho, P. Babarczi, and L. Rónyai, „On signaling-free failure dependent restoration in all-optical mesh networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1067–1078, Aug 2014.

[31] J. Tapolcai, P. H. Ho, P. Babarczi, and L. Rónyai, „Neighborhood failure localization in all-optical networks via monitoring trails," *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1719–1728, Dec 2015.

[32] A. Orda and A. Sprintson, „Efficient algorithms for computing disjoint QoS paths," in *23th IEEE INFOCOM*, vol. 1, 2004.

[33] A. Srivastava, S. Acharya, M. Alicherry, B. Gupta, and P. Risbood, „Differential delay aware routing for Ethernet over SONET/SDH," in *24th IEEE INFOCOM*, vol. 2, 2005, pp. 1117–1127.

[34] Georgios Ellinas, Eric Bouillet, Ramu Ramamurthy, Jean-Francois Labourdette, Sid Chaudhuri, and Krishna Bala. Routing and restoration architectures in mesh optical networks. *Optical Networks Magazine*, pages 91–106, January/February 2003.

[35] Tapolcai, János and Wu, Bin and Ho, Pin-Han and Rónyai, Lajos A novel approach for failure localization in all-optical mesh networks. *IEEE/ACM Transactions on Networking*, vol. 19, 2011 pages 275–285,

[36] Yadav, Ravinder and Aggarwal, Rinkle Rani Survey and Comparison of Optical Switch Fabrication Techniques and Architectures, arXiv preprint arXiv:1004.4481, 2010

[37] Ma, Xiaohua and Kuo, Geng-Sheng Optical switching technology comparison: optical MEMS vs. other technologies, *IEEE communications magazine*, vol. 41, 2003 pages S16–S23,

[38] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The Internet Topology Zoo. http://www.topology-zoo.org.

Hivatkozások

Folyóirat publikációk

[J1] P. Babarczi, A. Pašić, J. Tapolcai, F. Németh, and B. Ladóczki. Instantaneous recovery of unicast connections in transport networks: Routing versus coding. *Elsevier Computer Networks*, 82(1):68–80, 2015. (6/4 = 1,5)

[J2] A. Pašić, P. Babarczi, and A. Kőrösi. Diversity coding-based survivable routing with QoS and differential delay bounds. *Optical Switching and Networking*, 23(2):118–128, 2017. (6/2 =3)

[J3] P. Babarczi, J. Tapolcai, A. Pašić, L. Rónyai, E. R. Bérczi-Kovács, and M. Médard. Diversity coding in two-connected networks. *IEEE/ACM Transactions on Networking*, 25(4):2308–2319, 2017. (6/5 =1.2)

[J4] A. Pašić, P. Babarczi, and J. Tapolcai. Unambiguous switching link group failure localization in all-optical networks. *Networks*, 70(4):327–341, 2017. (6/2 =3)

Konferencia publikációk

[C1] A. Pašić, J. Tapolcai, P. Babarczi, E. Bérczi-Kovács, Z. Király, and L. Rónyai. Survivable routing meets diversity coding. In *IFIP Networking*, pages 1–9, 2015. (3/5 = 0,6)

[C2] P. Babarczi, J. Tapolcai, A. Pašić, S. R. Darehchi, and P.-H. Ho. New addressing scheme to increase reliability in mpls with network coding. In *Design of Reliable Communication Networks (DRCN), 2013 9th International Conference on the*, pages 36–43. IEEE, 2013. (3/4 = 0,75)

[C3] A. Pašić and P. Babarczi. Delay aware survivable routing with network coding in software defined networks. In *Reliable Networks Design and Modeling (RNDM), 2015 7th International Workshop on*, pages 41–47. IEEE, 2015. (3/1 = 3)

[C4] A. Pašić and P. Babarczi. Switching link group failure localization via monitoring trails in all-optical networks. In *Resilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on*, pages 92–99. IEEE, 2016. (3/1 = 3)

Egyéb publikációk

[C5] Tornatore, Massimo and André, Joao and Babarczi, Péter and Braun, Torsten and Følstad, Eirik and Heegaard, Poul and Hmaity, Ali and Furdek, Marija and Jorge, Luisa and Kmiecik, Wojciech and Mas Machucaand, Carmen and Martins, Lucia and Medeiros, Carmo and Musumeci, Francesco and Pašić, Alija and Rak, Jacek and Simpson, Steven and Travanca, Rui and Voyiatzis, Artemios. A survey on network resiliency methodologies against weather-based disruptions. In *Resilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on*, pages 23–34. IEEE, 2016. (3/18 = 0.16)

[C6] Gomes, Teresa and Tapolcai, János and Esposito, Christian and Hutchison, David and Kuipers, Fernando and Rak, Jacek and de Sousa, Amaro and Iossifides, Athanasios and Travanca, Rui and André, Joao and Jorge, Luísa and Martins, Lúcia and Ortiz Ugalde, Patricia and Pašić, Alija and Pezaros, Dimitrios and Jouet, Simon and Secci, Stefano and Tornatore, Massimo. A survey of strategies for communication networks to protect against large-scale natural disasters. In *Resilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on*, pages 11–22. IEEE, 2016. (3/15 = 0.2)

[C7] Pašic, Alija and Girao-Silva, Rita and Vass, Balázs and Gomes, Teresa and Babarczi, Péter FRADIR: A Novel Framework for Disaster Resilience. In *Resilient Networks Design and Modeling (RNDM), 2018 10th International Workshop on*, pages 1–7. IEEE, 2018. (3/4 = 0.75)