Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

Transzport hálózatok regionális hibák elleni védelme

TDK dolgozat

Készítette:

Mogyorósi Ferenc

Konzulens:

Dr. Pašić Alija

2019

Tartalomjegyzék

Kivonat i

Abstract ii

1. Bevezetés 1
1.1. A dolgozat struktúrája . . . 3

2. Transzport hálózatok 4
2.1. A transzport hálózatok fejlődése . . . 4
2.2. A hálózat és a forgalom leírása . . . 5
2.3. Hálózati hibák modellezése és a rendelkezésre állás . . . 6
2.3.1. Hálózati hibák modellezése . . . 6
2.3.2. Megbízhtósági értékek (Availability, Reliability) . . . 7
2.4. Védelmi megoldások transzport hálózatokban . . . 8
2.4.1. Dedikált védelmi módszerek . . . 8
2.4.1.1. Útvonalválasztás alapú módszerek . . . 9
2.4.1.2. Hálózati kódoláson alapú módszerek . . . 9
2.4.2. Megosztott védelmű módszerek (Shared Protection Approaches) . . . 9

3. A FRADIR keretrendszer 10
3.1. Hálózatfejlesztés . . . 10
3.1.1. Spine . . . 10
3.2. Hibamodellezés . . . 11
3.3. General Dedicated Protection . . . 11
3.3.1. General Dedicated Protection with Network Coding . . . 11
3.3.2. General Dedicated Protection with Routing . . . 12
3.4. FRADIR . . . 13

4. Idősor analízis és predikció 15
4.1. Idősor előrejelzési stratégiák . . . 15
4.1.1. Rekurzív előrejelzési stratégia . . . 15
4.1.2. Közvetlen előrejelzési stratégia . . . 16
4.2. Idősor előrejelzési megoldások . . . 16
4.2.1. Neurális hálózatok . . . 17
4.2.1.1. Visszacsatolt hálózatok . . . 17
4.2.1.2. LSTM hálózatok . . . 18
4.2.1.3. Konvolúciós neurális hálózatok (Convolutional Neural Networks - CNNs) . . . 19

5. Hálózatfejlesztés az összekapcsoltság megőrzésére 21
5.1. Hálózatfejlesztési módszer a vágást okozó hibák kivédésére . . . 21

6. A hálózati forgalom becslése **24**
6.1. A rendelkezésre állás és a szabad kapacitás kapcsolata 24
6.2. Energy Sciences Network (ESnet) 25
6.2.1. A forgalmat alakító tényezők 25
6.2.2. A forgalom letöltése és előfeldolgozása 26
6.2.3. A forgalom előrejelezhetősége 26
6.3. A neurális hálózat felépítése 26
6.3.1. A hibafüggvény 27
6.3.2. A hálózatok finomhangolása 27
6.3.3. Tanítási módszerek 28

7. Szimulációs eredmények értékelése **29**
7.1. A hálózatfejlesztés hatása 29
7.2. A forgalom előrejelzés hatása 33

8. Összefoglalás **36**

Irodalomjegyzék **37**

Kivonat

Az Internet, mint a legnagyobb mesterséges hálózat elkerülhetetlen része életünknek. Rengeteg különböző alkalmazás használja ki a kis késleltetésű, nagy adatsebességű kommunikációt, ilyen például a távgyógyászat, a tőzsde, de említhetnénk bármilyen valós idejű irányítást igénylő alkalmazást. Az ilyen alkalmazások esetén bármilyen kiesés súlyos következményekkel jár, ezért szükséges ezen kapcsolatok kiemelt védelme. A hálózatot többféle meghibásodás is sújthatja, a linkhibáktól kezdve egészen a regionális kiesésekig. A hálózatok megbízhatóságának növeléséhez 3 fő területet hívhatunk segítségül:

- *Hibamodellezés*: A különböző hibalehetőségek modellezésével felfedezhetjük a hálózat gyenge pontjait és tesztelhetjük a védekezési lehetőségeket.
- *Hálózattervezés*: A hálózat megfelelő tervezése és célirányos fejlesztése nagyban hozzájárul a magas megbízhatóság eléréséhez.
- *Megbízható útvonalválasztás*: A megfelelő útvonalválasztás (routing) kiválasztásával sok hibát kivédhetünk, viszont ennek az ára a magas erőforrásigény lehet.

A dolgozatban egy olyan rendszert fejlesztettem, az úgynevezett FRADIR-t (FRAmework for DIsaster Resiliance), amely ezt a három területet egyesíti és regionális hibák esetén is garantálja a megszakítatlan információáramlást. Dolgozatomban javasoltam egy hálózatfejlesztési metódust, amely biztosítja a hálózat összefüggőségét nagy kiterjedésű hibák esetén is. A hibamodellezésnek megfelelően többféle fejlesztési lehetőség áll rendelkezésünkre, amik közül pénzügyi szempontok alapján kell választani. A megbízható útvonalválasztás az alacsonyabb rendelezésre állású hálózaton is képes a szükséges megbízhatósággal átvinni az információt, viszont ez magas erőforrás-használattal járhat.

Mivel az útvonalválasztási algoritmusok alkalmazhatósága a hálózatban elérhető szabad kapacitástól is függ, így az alkalmazható megbízható útvonalválasztás (QoS szint) megfelelő kiválasztásának érdekében meg kell tudni becsülni, hogy a kapcsolat fennállásának ideje alatt milyen forgalom várható az egyes linkeken. A forgalombecslést neurális hálózatok segítségével végeztem el. Többféle hálózat típus kipróbálása után egy hibrid megoldás mellett döntöttem, ami egy konvolúciós hálózat és egy visszacsatolt LSTM (Long Short-Term Memory) hálózat összekapcsolásából állt. Mivel a forgalom alulbecslése azt jelentené, hogy a megbízható útvonalválasztási algoritmus olyan szabad kapacitásokat használna fel, amelyek nem is állnak rendelkezésre, ezért a hálózatot egy saját költségfüggvény segítségével tanítottam be, ami a forgalom alulbecslését nagyobb mértékben bünteti. A neurális hálózatot az ESnet tudományos laboratóriumok közötti nagysebességű hálózatának forgalmi adatai alapján tanítottam be.

Tehát a FRADIR egy olyan keretrendszer, amely egyszerre képes kihasználni a hibamodellezés, a hálózattervezés és a megbízható útvonalválasztás előnyeit mégpedig költséghatékony módon. A FRADIR hatékonyságának növelésére új hálózattervezési módszereket javasoltam és a megbízható útvonalválasztás finomhangolásához betanítottam egy forgalombecslésre képes neurális hálót. A forgalom becslése előrevetíti a közeljövőben elérhető szabad kapacitást, amit a kapcsolatok megbízhatóságának növelésére lehet fordítani.

Abstract

Internet as the largest artificial network is an unavoidable part of our life. Many different applications utilize this low latency and high throughput communication form. Some of these applications like remote surgery, stock exchange and other management systems require a very high QoS (Quality of Service). Any kind of failure can cause huge damages to these services so even increased delay is impermissible. Hence these applications require sophisticated protection mechanisms. The protection must cover link failures and regional failures, too. To increase the reliability of the network we can benefit from 3 fields of network science:

- *Failure Modelling*: With the modelling of different failure scenarios we can discover the weak points of the network and test the protection approaches
- *Network Planning*: The right network planning and deployment is indispensable to achieve high reliability
- *Survivable Routing*: Choosing the right routing approach can help us protect patterns, however it can require more resources.

In this study I developed the framework called FRADIR (FRAmework for DIsaster Resiliance) which combines the achievements from these 3 fields and guarantees that even in the case of regional failures the information flow is uninterrupted. I proposed a network planning method that ensures the connectivity of the network even in case of regional failures. Based on the failure modelling several strategies are available, nonetheless the decision has to take into account the financial aspects, too. Survivable routing can provide the required reliability even for poor circumstances, however it can require significantly more resources. The traffic in the network has a large influence on the survivable routing method. In order to select the proper routing method (QoS level) at every moment an estimated traffic on each link has to be known for the time period of the connection. For this task I created a neural network traffic prediction tool. After testing several types of neural networks I selected a hybrid neural network. It is a convolutional neural network connected to a LSTM ((Long Short-Term Memory)) neural network. Since the underestimation of the traffic would mean that the survivable routing would use unavailable capacity I trained the network with a custom loss function which punishes the underestimation more. I trained the neural network with the traffic data of the ESnet high-speed scientific network.

In summary the FRADIR is a framework which can utilize the advantages of failure modelling, network planning and survivable rooting at the same time and in a cost-efficient manner. To improve the efficiency of FRADIR I proposed new network planning methods and I fine-tuned the survivable routing selection with the help of a neural network that can estimate the traffic for a required time period. The estimated traffic defines the spare capacity which can be used to improve the availability of the connections.

1. fejezet

Bevezetés

A megbízhatóság az élet minden területén az egyik legfontosabb kérdés. A bizalom alapvető szükségletünk, mert enélkül társadalmunk működőképessége nagyban lecsökkenne. Elég csak a közlekedésre vagy az egészségügyre gondolni.
Hasonló a helyzet a telekommunikációs hálózatok esetében is, mivel egyre nagyobb az igényünk ezen hálózatok mindennapos használatára, és ezzel együtt életünk egyre nagyobb részét befolyásolja, a bevásárlástól kezdve az ügyintézésig. A fiatalabb generáció mára már magától értetődőnek tartja, hogy "bárki bárkit bárhol és bármikor" el tud érni a telekommunikációs hálózatoknak köszönhetően. Vagyis az igények és követelmények igen magassá váltak, hiszen minden szolgáltatásnak azonnal elérhetőnek kell lennie. Mivel a szolgáltatók bevételét a felhasználók jelentik, így a versenyelőny megszerzésének érdekében szeretnék ezen igényeket maradéktalanul kielégíteni. A jobb hálózatok új felhasználókat és ezzel együtt több használatot jelentenek, ami jelentősen hozzájárul a bevétel növekedéséhez.
Ezek miatt a megbízhatóság és QoS (Quality of Service) vált a két legfontosabb mérőszámmá a telekommunikációs hálózatok operátorainak szemében [32]. Miután a kommunikációs rendszerek felelnek az információáramlásért és sok irányítási rendszerért, így nem meglepő hogy sokan ezt tartják a legkritikusabb infrastruktúrának és így a jövőben a szerepük még inkább nőni fog.
Napjainkban egyre több alkalmazás igényel különösen magas megbízhatóságú és kis késleltetésű összeköttetést, ilyen például a tőzsde és a telesurgery (távoli operáció), ami még inkább növeli a megbízhatósági követelményeket és természetes igényként jeleníti meg a megbízható útvonalválasztást. Ezeknél az alkalmazásoknál bármilyen rövid idejű kiesés óriási kárt tud okozni, így meghibásodások esetén is azonnali helyreállítást kell biztosítani. Ezeknek a magas QoS elvárásoknak a kielégítése szükségszerű ahhoz, hogy nagy megbízhatóságú kommunikációs és irányítási szolgáltatásokat tudjunk kiszolgálni, amit egyre több közigazgatási szerv, vállalat és magánszemély szeretne használni [12].
Eddig a hibamodelleket gyakran egyszerűsítették le egyszerű linkhibákra, vagyis feltételezték, hogy egyszerre csak egy link hibásodhat meg, de a kommunikációs rendszerek kritikus infrastruktúrává és célponttá válásával ez már nem elegendő. Ezen okok miatt a kommunikációs hálózatok katasztrófák (regionális kiesések) elleni védelme reflektorfénybe került az utóbbi években. Már a problémakör első lépése is kihívás, vagyis hogy hogyan modellezzünk egy regionális kiesést, hiszen a különböző típusú katasztrófák sokféle alakban, hatókörben és erősségben fordulnak elő, mely nagyon megnehezíti a hálózati védelem megtervezését.
Például egy természeti csapás (pl. cunami) hatalmas részét érinti egy hálózatnak és akár az összes eszközt használhatatlanná teszi az érintett területen [22]. Nem is olyan régen a Sandy hurrikán, mely 2012-ben söpört végig az USA-n, New York áramhálózatának a felét érintette [13]. Sajnálatos módon nem csak efajta természeti katasztrófák képesek ilyen

mértékű pusztításra, hanem emberalkotta fegyverek is, például egy atombomba vagy célzott támadások. Ezeket a hibákat összefoglalóan regionális hibáknak nevezzük és definíció szerint olyan együttes kiesése csomópontoknak vagy linkeknek, melyek területileg egy érintett helyszínen vannak [22]. A kommunikációs hálózatok efajta kiesésektől való megóvása általában a hálózat fejlesztésén alapszik, vagyis az elemek megbízhatóságának növelésén, de ezen kívül létezik többféle más megoldás is. A jó eredmény érdekében a következő 3 területet hívhatjuk segítségül [27]:

Hibamodellezés: A terület célja a lehetséges hibák minél pontosabb modellezése, amely elengedhetetlen a hálózat megfelelő fejlesztéséhez, hiszen segít megtalálni annak gyenge pontjait.

Hálózattervezés: Ennek segítségével modellezhetjük, hogy a hálózatban eszközölt változtatásaink mennyiben segítették elő az általunk elvárt igények kielégítését. Meglathatjuk, hogy mely linkeket érdemes fejleszetni a magasabb megbízhatósági követelmények kielégítése érdekében. Mivel a linkek fejlesztése költséges dolog, így nagyon kell ügyelni arra, hogy minél költséghatékonyabban érjük el a kívánt eredményt.

Védelmi mechanizmusok: Ez a terület a kapcsolathoz tartozó megbízható útvonalválasztással foglalkozik, mely kielégíti az alkalmazás QoS igényeit. Itt is fontos figyelni az erőforrások hatékony felhasználására, de méginkább a meghatározott hibák megfelelő védelmére.

Általában a három terület egymástól függetlenül fejlődik és az egyes területeken elért új eredmények külön-külön viszik előre a hálózatok megbízhatóságát. Ez leginkább annak köszönhető, hogy a hálózatok mindenhol jelen vannak a világunkban és az egyes szektorokat (pl. energiaszektor) más más cél vezérli modellezési, tervezési és védelmi szempontból. Még a kommunikációs hálózatokon belül is teljesen más védelmi mechanizmusokra van szükségük a különböző QoS osztályokba tartozó kapcsolatoknak. A dolgozatomban egy olyan rendszert fejlesztettem, mely ezt a három területet egyesítve erősíti a kritikus alkalmazások regionális hibák elleni védelmét.

A FRADIR (FRAmework for DIsaster Resilience) egy olyan keretrendszer, amely egyesíti a hibamodellezést, a hálózattervezést és a megbízható útvonalválasztást, hogy erősítse a kritikus alkalmazások regionális hibák elleni védelmét. A hálózattervezés segítségével kialakít egy feszítőfát (Spine), melyet nagy megbízhatóságú linkek alkotnak, így növelve a kritikus alkalmazások összeköttetését. A regionális hibamodellezés segítségével SRLG-ket (Shared Risk Link Group) alakít ki, melyeket felhasználva dedikált védelmi struktúrákat hoz létre a kommunikáció megbízhatóságának javítása érdekében. Dolgozatomban javasoltam egy hálózatfejlesztési metódust, amely biztosítja a hálózat összefüggőségét nagy kiterjedésű hibák esetén is. A hibamodellezésnek megfelelően többféle fejlesztési lehetőség áll rendelkezésünkre, amik közül pénzügyi szempontok alapján kell választani.

Mivel a hálózatok megbízhatóságát nem csak a külső tényezők befolyásolják, így ha van rá lehetőség mindenképpen érdemes a hálózati adatforgalmat is megvizsgálni. Az adatforgalom kiértékelése, a trendek megértése és a jövőbeli forgalom becslése lehetővé teszi a hálózattervezési módszereink javítását, illetve az útvonalválasztási algoritmusok fejlesztését. Ezen okok miatt a forgalmi analízis kifejezetten fontos lett a hálózatüzemeltetők számára. Egy pontos forgalom-előrejelző rendszer elengedhetetlen a helyes hálózatmenedzsmenthez, segít az erőforrásfoglalás megtervezésében, hosszútávú kapacitás tervezésben, hálózatfejlesztésben és anomália detekcióban. Már a 90-es évektől kezdve jelentek meg cikkek melyek a hálózati forgalom előrejelzésére adtak különböző megoldásokat, valamint javaslatot tettek az előrejelzések felhasználására is. A forgalmi trendek megértése akkor is segítséget jelent, ha a rejtett információkat szeretnénk felfedezni benne. Az anomália

detekció rendkívül fontos napjainkban, amikor a támadók nem fizikailag szeretnék tönkretenni a hálózatot, hanem inkább el akarják lehetetleníteni a működését. Mivel a hálózati forgalom idősor, így az idősoranalízis eredményei felhasználhatóak a forgalmi trendek keresésére, előrejezésére. Ennek megfelelően az első megoldások, az idősoranalízis már meglévő módszereit alkalmazták a forgalom előrejelzésére. Többen értek el jó ereményeket a Holt-Winters [36], az ARIMA [10] és a neurális hálózatok [8, 19] segítségével. Mivel a *megfelelő útvonalválasztási algoritmus kiválasztását befolyásolja a hálózatban elérhető szabad kapacitás, ezért az helyes választás érdekében meg kell tudnunk becsülni, hogy a kapcsolat élettartama alatt milyen forgalom* várható az egyes linkeken. A forgalombecslést neurális hálózatok segítségével végeztem el. Többféle hálózattípus kipróbálása után egy hibrid megoldás mellett döntöttem, ami egy konvolúciós hálózat és egy visszacsatolt LSTM hálózat összekapcsolásából állt. Mivel a forgalom alulbecslése azt jelentené, hogy a megbízható útvonalválasztási algoritmus olyan szabad kapacitásokat használna fel, amelyek nem is állnak rendelkezésre, ezért a hálózatot egy saját költségfüggvény segítségével tanítottam be, ami a forgalom alulbecslését nagyobb mértékben bünteti. A neurális hálózatot az ESnet tudományos laboratóriumok közötti nagysebességű hálózatának forgalmi adatai alapján tanítottam be.

A dolgozatomban a transzport hálózatok regionális hibák elleni védelmére fogok koncentrálni, bemutatom a state-of-the-art megoldásokat, de a munkám fókuszában a FRADIR fog állni. Dolgozatomban javaslatot teszek egy új hálózattervezési eljárásra, amely regionális hibák esetén is képes biztosítani a hálózat összefüggőségét. A rendszert kiegészítettem egy neurális hálózat alapú forgalombecslő eljárással, ami a linkek korábbi forgalmi adatai alapján képes becslést adni az egyes linkeken várható szabad kapacitásra. Ez a becslés lehetővé teszi az útvonalválasztás továbbfejlesztését, hiszen a linkek szabad kapacitása meghatározza az alkalmazható megbízható útvonalválasztási algoritmusokat.

1.1. A dolgozat struktúrája

A dogozatom 8 fejezetből áll, melyek bevezetnek a problémakörbe és bemutatják a jelenlegi megoldásokat. Ezen felül részletezik a munkám során létrejött fejlesztéseket és azok eredményeit. A fejezetek a következő sorrendben követik egymást:
Az 1. fejezetben a munkám motivációi és a problémakör alapjai találhatók meg.
A 2. fejezetben a transzport hálózatokról és a transzport hálózatokban alkalmazott védelmi mechanizmusokról adok áttekintést.
A 3. fejezetben bemutatom a FRADIR keretrendszert, az eddigi megoldásokat és az új fejlesztési irányokat, melyeken én is dolgoztam.
A 4. fejezetben az idősor analízis és forgalombecslés témakörökről, valamint az általam használt módszerekről adok áttekintést.
A 5. fejezetben az általam készített hálózatfejlesztési metódusokat mutatom be, melyek különböző módszerekkel segítik a hálózat összefüggőségének megtartását.
A 6. fejezetben a forgalombecslő megoldásomat mutatom be és az alkalmazását az összeköttetések megbízhatóságának növelésére.
A 7. fejezetben kiértékelem a szimulációs eredményeket.
8. fejezetben összefoglalom az elért eredményeket és bemutatom a további fejlesztési irányokat.

2. fejezet

Transzport hálózatok

A transzport hálózatok optikai kapcsolókból (OXC) és a köztük menő kommunikációs csatornákból állnak. Ezek a csatornák – linkek – kötik össze egy hálózatban a kapcsolókat. A kapcsolók továbbítják az adatot a megfelelő irányba. Az útvonalakat már előre kiszámolják, így a kapcsolóknak már csak a továbbítás a feladata. Nyilván több lehetséges útvonal létezik két csomópont között, de nem mindegyik elégíti ki az alkalmazás követelményeit. Az útvonalakat optimalizálhatjuk megbízhatóságra, késleltetésre vagy akár sávszélességre, költségre is.
A mi célunk olyan útvonalak megtalálása, melyek még akkor is kielégítik a követelményeket, ha hibák lépnek fel a hálózatban (ember vagy természet okozta). Ehhez ismernünk kell a hálózatok tulajdonságait és a metódusokat, melyek segítenek kihasználni az erősségeiket és elfedni/védeni gyengeségeiket. Ebben a fejezetben a következő témákat fogom érinteni:

- A transzport hálózatok fejlődése
- A hálózatok leírása és a forgalom modellezése
- Gyakori hálózati topológiák
- Hibamodellezés és a hibák leírása SRLG-k segítségével
- Helyreállítási folyamatok és idők hibák esetén

2.1. A transzport hálózatok fejlődése

Ahhoz, hogy megértsük a transzport hálózatok működését először ismernünk kell a kialakulásuk körülményeit. A transzport hálózatok régebben különböző hálózatok összekötésére szolgáltak, a köztük való adatcserére hoztak létre utakat. Mivel óriási adatmennyiséget (Tbit/s) kell továbbítaniuk, így tipikusan optikai hálózatokról van szó, ahol az adatot optikai jelként (modulált fényként) továbbítják [30].
Az úgynevezett Pont-Pont optikai hálózatok esetén a kapcsolat forrásának és céljának direkt összeköttetésben kell lenniük egy link segítségével. A kezdetekben így funkcionált a hálózatok optikai rétege, de ahogy a forgalom exponenciálisan nőtt az idővel (az Internet megjelenésének köszönhetően) ez a megoldás túlságosan redundánsá vált. Így természetesen megjelent az igény óriási adatmennyiségek hatékony mozgatására ezekben a hálózatokban [40].
Sok megoldást kipróbáltak, melyek nem váltották be a hozzájuk fűzött reményeket. Ezek legtöbbször skálázhatatlannak bizonyultak, mint például a bitsebesség növelése vagy a kábelekben használt optikai szálak számának növelése. Mivel ezek költség és megvalósíthatóság szempontjából is lehetetlen megoldások voltak, így megbuktak [40].

Az első nagy lépés előre a különböző hullámhosszú fényutak multiplexálása volt (WDM – Wavelength Divison Multiplexing). Ennek lényege, hogy több jelet is átküldhetünk egy optikai szálon, ha különböző hullámhosszú fénysugarakat használunk. Így már egy optikai szál annyi adat átvitelére volt képes, amelyhez korábban több is szükséges volt. Legnagyobb előnye, hogy a meglévő infrastruktúra kapacitását növelte. Több protokoll is ezt a koncepciót alkalmazza, például a Synchronous Optical NETworking (SONET) és a Synchronous Digital Hierarchy (SDH), ezek az egész világon alkalmazott megoldások[40].
A nagyszerű ötlet ellenére ennek is vannak hátrányai, amivel a SONET/SDH gyorsan találkozott. Mivel előre meghatározott méretű csomagokat továbbítottak, így nagy volt a kihasználatlan sávszélesség miközben adat- vagy hangforgalmat továbbítottak.
Az Optikai Transzport Hálózatok (OTN – Optical Transport networks) a SONET/SDH techológiákból fejlődtek ki, amikor létrejöttek az első optikai csomópontok. Ezek már nem hajtották végre az optikai-digitális-optikai átalakítást, ezért láthatatlan kapcsolóknak (transparent switch) is hívták őket. Ez lehetővé tette, hogy a forgalom egészen a céljáig az optikai síkon maradjon. Így a hálózat már nevezhető tisztán optikai hálózatnak [27].
Az Optikai Transzport Hálózatok a következő részekből épülnek fel [40]:

- Optikai Továbbító Réteg (OTS – Optical Transmission Section): Ez a legalsó réteg, mely biztosítja, hogy az optikai jelek továbbítása minden optikai szálon működjön.
- Optikai Multiplexáló Réteg (OMS – Optical Multiplex Section): Ez a következő réteg, ennek feladata, hogy a több hullámhosszt tartalmazó jelek is helyesen menjenek át a hálózaton.
- Optikai Csatorna (OCh – Optical Channel): Ez a legfelső réteg, mely továbbítja az alkalmazások különféle jeleit a hálózatban.

Az Internet elterjedéséig az optikai hálózatok fő forgalmát a telefonhívások adták, de az ezredfordulóra (az Internetnek köszönhetően) ez teljesen eltolódott az adatforgalom irányába. Ennek a váltásnak köszönhetően jelentkezett az igény az optikai rétegek összekapcsoltságának dinamikus megváltoztatására. Erre vezették be az ASON-okat (Automatically Switched Optical Network), ahol a vezérlő réteg lehetőséget ad automatikus védelemre és újraküldésre, kapcsolat-felépítésre és lebontásra valamint topológia és erőforrás felderítésre. Manapság a fejlesztések fő célja a flexibilitás és megbízhatóság növelése [40].
Az ilyen típusú dinamikus hálózatokban nehéz megfelelő védelmi megoldásokat találni, hiszen összhangot kell teremteni a helyreállítási idő, a komplexitás és a késleltetési értékek között.

2.2. A hálózat és a forgalom leírása

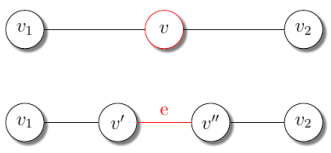
A kommunikációs hálózatokat általában gráfokkal reprezentálják ($G = (V, E)$), ahol a gráf csúcsai az optikai kapcsolókat (melynek halmazát V-vel jelöljük), élei pedig a köztük futó kommunikációs csatornákat (halmaza E) reprezentálják. Tehát, ha két optikai kapcsoló össze van kötve egy optikai szállal, akkor a kapcsolókat reprezentáló két csúcs szomszédos a gráfban, vagyis fut köztük él. Amennyiben a kommunikáció csak egy irányban lehetséges – pl. irányított a kommunikációs csatorna – akkor irányított éleket használunk a gráfban. Ha a kommunikáció szokásos kétirányú, akkor vagy két irányított vagy egy irányítatlant élt használunk. Mivel a munkám során kétirányú kommunikációs csatornákkal foglalkoztam, így a továbbiakban él alatt irányítatlan élt értek.
Az éleknek ($e \in E$) többféle tulajdonága lehet, mint például a használat költsége ($c(e)$), a rendelkezésre állás ($A(e)$) vagy a szabad kapacitás ($k(e)$).

A költségfüggvény ($c(e)$) megadja, hogy az adott élen mennyibe kerül egységnyi erőforrás (pl. sávszélesség) lefoglalása. A rendelkezésre állás ($A(e)$) a link működési időtartamához kapcsolódik (lásd. 2.3.2). A hálózat leírása után térjünk át a kapcsolatok leírására. Egy kapcsolat kérést a $C = (s, t, b)$ hármassal reprezentálunk, ahol $s \in V$ a forrás, $t \in V$ a cél és b az igényelt sávszélesség. Mivel a hálózat irányítatlan, így feltételezhetjük, hogy a kommunikáció két iránya ugyanazt az útvonalat használja.
A munkám során dinamikus útvonalválasztást használtam, vagys feltételeztem, hogy a forgalmi igények egymás után érkeznek és a jövőbeli igényekről nincs információnk, így egymástól függetlenül választunk nekik útvonalakat. Fontos megjegyezni, hogy a célunk minden kapcsolat esetén egy olyan útvonal biztosítása, hogy bármelyik lehetséges hiba esetén (egy link vagy akár regionális) az adat megérkezzen a céljához.

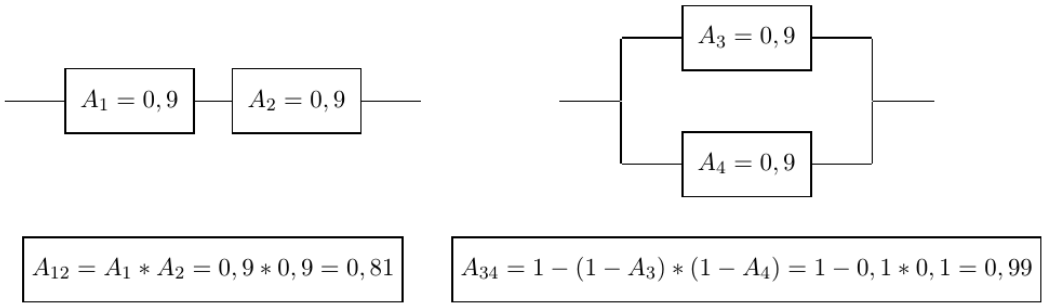
2.3. Hálózati hibák modellezése és a rendelkezésre állás

2.3.1. Hálózati hibák modellezése

Ahogy már korábban említettem, a hálózat gráf reprezentációja egy V csúcshalmazból és egy E élhalmazból áll, melyek az optikai kapcsolókat és a kétirányú optikai szálakat reprezentálják. Mivel a valóságban semmi sem tökéletes, így ezeknek az elemeknek a meghibásodásával is számolnunk kell.
Hibaeseménynek azt nevezzük, ha a hálózat valamely része nem működik megfelelően. Ezt okozhatják külső hatások, például természeti katasztrófák, de gyakran emberi okok állnak a háttérben, például szoftverhibák vagy akár eltervezett támadások. Hibák nem csak külső hatások révén keletkezhetnek, hanem elhasználódásból fakadólag is [40].
A hálózat megfelelő üzemeltetéséhez, a hiba okától függetlenül képesnek kell lennünk a hálózat védelmére vagy legalább arra, hogy megakadályozzunk egy komoly kiesést. A megelőzéshez nyilvánvalóan tudnunk kell megbízható módon modellezni a hálózatot és az esetleges meghibásodásokat.
A hálózati hibamodellezés legelterjedtebb módszere az SRLG (Shared Risk Link Group) listák használata. Ez a módszer lehetőséget ad arra, hogy egymással összefüggő hibákat együtt írjunk le, mivel képes a hibák közti függőségek kihasználására. Példaképp tekintsünk két látszólag független élt, ezek egy SRLG-ben lehetnek, ha a valóságban a kábelük ugyanazon az alagúton halad végig, mivel ha egy katasztrófa (rosszul sikerült robbantás) esetén az alagút beomlik, akkor mindkét link megszakadhat. Ez egy nagyszerűen használható eszköz a regionális hibák modellezésére is, mivel a linkek meghibásodása főleg a geometriai elhelyezkedésüktők függ. Nyilvánvaló, hogy minden lehetséges élhalmaz leírása regionális hibamodellezés céljából nem lehetséges, mivel ez exponenciálisan sok SRLG-t jelentene. A megfelelő modellezéshez nagyon gondosan kell kiválogatni az SRLG-ket, hogy minden fontos hibát lefedjenek, mégis elfogadható mennyiségű SRLG-vel dolgozzunk.
Mivel a fellépő hibákat linkhibával modellezzük, így a csomópontok nem hibásodhatnak meg. Ezt úgy tehetjük meg, hogy egy csúcspont meghibásodását egy linkhibára vezetjük vissza, úgy hogy a meghibásodott v csúcspontot felbontjuk v' és v'' csúcspontokra, melyeket egy e éllel kötünk össze (lásd. 2.1 ábra).



2.1. ábra. Egy csúcspont hibamodellezése egy új link bevezetésével.



2.2. ábra. Példa a 2.3.2 és 2.3.2 egyenletek használatára.

Egy másik módszer a csomóponti hibák modellezésére azok linkhibákkal való helyettesítése, vagyis egy node kiesését reprezentáló SRLG lista minden olyan élt tartalmazni fog, melynek egyik végpontja a meghibásodott node volt.

2.3.2. Megbízhtósági értékek (Availability, Reliability)

A megbízhatósági értékekkel a hálózat minőségét írjuk le. Ahhoz, hogy elemezni tudjuk a transzport hálózatok megbízhatóságát pontosan értenünk kell a kifejezések definícióját.
A megbízhatósági (reliability) érték annak a valószínűsége, hogy a hálózat egy adott időintervallumban ($T = t_2 - t_1$) megfelelően működik [40].
A rendelkezésre állás (availability) annak a valószínűsége, hogy egy hálózati elem működik egy adott időpontban (t_1) [40].
Ahhoz, hogy egy elem rendelkezésre állását pontosan leírjuk az MTBF (Mean Time Between Failures) és MTTR (Mean Time To Repair) értékeket használjuk. Az MTBF a hibák közt eltelt átlagos időtartamot adja meg egy elemre, míg az MTTR a hiba kijavításához szükséges átlagos időt adja meg. Ezeket az értékeket használva egy él rendelkezésre állását ($A(e)$) a következőképpen írhatjuk le:

$$A(e) = \frac{MTBF}{MTBF + MTTR}$$

Az egész hálózat rendelkezésre állását az egyes linkek rendelkezésre állásának átlagaként szokás meghatározni [40]. Ezen kívül más metódusok is vannak a hálózat rendelkezésre állásának meghatározására, például egy adott időintervallumban azoknak a felhasználói perceknek a száma, amelyek valamilyen hálózati hiba alatt voltak.
Linkek soros kapcsolásából létrehozott út rendelkezésre állását (A_S) az egyes linkek rendelkezésre állásának szorzataként határozhatjuk meg:

$$A_S = \prod_{l=1}^{n} A_l$$

A párhuzamos linkek együttes rendelkezésre állását (A_P) a következő kifejezéssel kaphatjuk meg:

$$A_P = 1 - \prod_{l=1}^{n} (1 - A_l)$$

A 2.2 ábrán látható, hogy az együttes rendelkezésre állások, hogyan számolhatók ki sorosan (bal) és párhuzamosan (jobb) összekapcsolt linkek esetén.
Nyilvánvaló, hogy nem minden hálózat bontható le csak soros és párhuzamos linkekre,

vannak olyan struktúrák, melyek túl összetettek ennek a metódusnak az alkalmazására, így nagyon sok más elterjedt módszer is van rendelkezésre állás számolására.

A csúcspontok rendelkezésre állását általában 1-nek vesszük, mert legtöbbször a csúcsponti hibákat is linkhibákkal írjuk le.
Egy másik gyakran alkalmazott érték, mely jó képet ad a komponensekről az unavailability (elérhetetlenség). Ez annak a valószínűsége, hogy egy hálózati elem nem működik [39]. Ez a következő módon számítható:

$$U(e) = \frac{(downtime)}{(downtime) + (availabletime)}$$

Az elérhetetlenség és rendelkezésre állás közötti kacsolat könnyen megfogalmazható:

$$U(e) = 1 - A(e)$$

2.4. Védelmi megoldások transzport hálózatokban

Eddig megismerkedtünk a hálózatokkal, a hálózati hibák modellezésével, definiáltuk az availability és reliablility értékeket melyekkel a kapcsolatokat jellemezni tudjuk. A következő lépés az egyes hibaesemények kivédése vagyis annak a biztosítása, hogy a felhasználók/alkalmazások adatai akkor is célba érjenek, ha valamilyen kiesés történik. Hogyan biztosíthatunk az alkalmazások számára egy adott megbízhatósági szintet?
Ahhoz, hogy kielégítsük a QoS igényeket nem elegendő egyetlen útvonalat használni, mivel egyetlen link kiesése megszakíthatja a kapcsolatot. Ezért van szükségünk védelmi mechanizmusokra [31]. Ezeknek hálózati hibák esetén biztosítaniuk kell az azonnali helyreállítást ($t_R < 50$ ms), hogy megakadályozzuk az adatvesztést.
A használt útvonalak 3 védelmi szintbe sorolhatók: védelem nélküli, egy link kiesése ellen védett és több link kiesése ellen védett. A hálózatok többsége egy link kiesése ellen ad védelmet, de ha több link kiesik vagy regionális hiba lép fel akkor a kapcsolatok védtelenek. A jelenlegi trendek a hálózatok védelmének megerősítése felé mutatnak. Ez nem meglepő hiszen a kommunikációs hálózatok egyre hangsúlyosabbak a mindennapi élet során. Például [4]-ban egy védelmi keretrendszert mutattak be mely képes meghatározott hibákat (SRLG listák) kivédeni, amennyiben a hálózatban a kiesés után is lesz út a forrás és a cél között. Természetesen a különböző védelmi mechanizmusok különböző előnyökkel rendelkeznek (pl. helyreállítási idő, számítási komplexitás, erőforráshasználat).
Ebben a fejezetben különböző napjainkban alkalmazott védelmi módszereket és jelenlegi kutatási irányokat fogok bemutatni.

2.4.1. Dedikált védelmi módszerek

A [27] szerint a transzport hálózatok szigorú QoS követelményeinek teljesítésének legjobb módja a dedikált védelmi mechanizmusok használata.
Ennek fő oka az, hogy a dedikált védelemi megoldások az adatfolyam több másolatát küldik el egyszerre, de különböző útvonalakat használva. Ez az eljárás magas rendelkezésre állást eredményez még előre nem látható kiesések esetén is [27]. Nagy előnye még, hogy ez garantálja az azonnali helyreállást hiba esetén. Ez azt jelenti, hogy a helyreállítási idő kevesebb mint 50 ms, így nincs szükség adatfolyam-átirányításra, mert ugyanazok az adatok már el lettek küldve a tartalékutakon [27].

Mivel a munkám során szövevényes típusú hálózatokkal foglalkoztam, így csak az ezekben a hálózatokban alkalmazott védelmi megoldásokkal fogok foglalkozni.

2.4.1.1. Útvonalválasztás alapú módszerek

Az egyik legígéretesebb dedikált védelmi módszer szövevényes hálózatokban az ún. *Dedicated Backup Path Protection* (DBPP), melyet gyakran dedikált 1+1-es védelemnek vagy csak egyszerűen 1+1 védelemnek hívnak. Ez egy olyan védelmi megközelítés, amely mind a üzemi, mind a tartalék útvonalon párhuzamosan továbbítja a forgalmat. Így a üzemi útvonalon történő hiba esetén a vevő egyszerűen átkapcsol a tartalék útvonalra. Fő előnye az egyszerűsége és hogy azonnali helyreállítást biztosít hiba esetén. A hibalokalizációs időt szinte 0-nak tekinthetjük és az átkapcsolási idő az útvonalak késleltetéskülönbségével egyezik meg. Az előnyei mellett sajnos hátrányai is vannak, például, hogy főleg egyszeres linkhibák véldelmére alkalmas, egy költségparaméterre optimalizálható és nagyon sávszélességigényes [6]. Ezek miatt a közeljövőben valószínűleg kiváltja egy jobb algoritmus, de az egyszerűségével egyenlőre egy sem veheti fel a versenyt. Természtesen léteznek más dedikált védelmi módszerek is, ilyen a General Dedicated Protection (GDP) családjába tartozó *General Dedicated Protection with Routing* (GDP-R) [5], amely nem két utat, hanem egy tetszőlges védelmi struktúrát használ a különböző SRLG listában szereplő hibák védelmére. Erről a módszerről részletesebben szót ejtek a 3 fejezetben.

2.4.1.2. Hálózati kódoláson alapú módszerek

Hálózati kódolás esetén a csomópontok nem egyszerű továbbítóként működnek, hanem képesek algebrai műveleteket végrehajtani a beérkező csomagokon, hogy kódolt kimenő csomagokat állítsanak elő, vagyis hogy hálózati kódolást hajtsanak végre a felhasználói adatokon [27].
A hálózati kódolás használható statikus (inter-session) és dinamikus (intra-session) forgalom esetén is [27]. A hálózati kódolás előnyeit a transzport hálózatokban először egy forgalom-minimalizálási kísérletben ismerték fel, ahol adott forgalmi igényeket kellett a lehető legkevesebb kapacitás felhasználásával kielégíteni. Az első alkalmazása védelmi módszerben az ún. $1 + N$ védelemben volt [17]. A módszer azóta sokat fejlődött, de az alapprobléma ugyanaz maradt. A forgalom ismerete csak ritkán lehetséges a mai dinamikus hálózatokban, így a különböző kapcsolatok közötti hálózati kódolás került a figyelem középpontjába [27].
A leghíresebb hálózati kódoláson alapuló módszer a *Diversity Coding* (DC). Ebben az adatot először kétfelé osztjuk és két különböző útvonalon továbbítjuk a hálózatban, majd a redundancia növelése érdekében egy kódolt csomagot ($A \oplus B$) is elküldünk az előzőektől éldiszjunkt úton.Már elsőre sejthető ennek a módszernek a nehézsége, mivel a hálózattól nagymértékű összekötöttséget vár el.A hálózati kódolást alkalmazza még a GDP családjába tarozó GDP with Network Coding (GDP-NC) is.

2.4.2. Megosztott védelmű módszerek (Shared Protection Approaches)

Ezen módszerek esetén a tartalék útvonal csak a hiba fellépése után aktiválódik, vagyis nem küldjük folyamatosan az adatot a tartalék útvonalon [6]. Ez azzal jár, hogy hiba esetén szükség van az adat újraküldésére, amely a kapcsolat gyors helyreállítását lehetetlenné teszi, hiszen a hiba lokalizációjával és az értesítési időkkel már annyira megnövekszik ez az idő, hogy az azonnali helyreállítása lehetetlenné válik [27].
A nevéből adódóan fontos tulajdonsága, hogy egy tartalék út több üzemi út védelmét látja el. Ez előnyös erőforráshasználat szempontjából, de miután az azonnali helyreállítás egy kötelező elvárás, így nem nagyon használható ebben a problémakörben.

3. fejezet

A FRADIR keretrendszer

Ebben a fejezetben bemutatom a FRADIR keretrendszert komponenseire lebontva. Elsőként a hálózatfejlesztési módszereket, melyek célja a hálózat megbízhatóságának és robosztusságának növelése. Ezután a hibamodellezést mutatom melyet a hálózatfejlesztés és útvonalválasztási módszer is használ. Utoljára pedig az útvonalválasztási metódust részletezem, ami a *General Dedicated Protection* (GDP). Végül a teljes keretrenszert mint egységet mutatom be.

3.1. Hálózatfejlesztés

A FRADIR hálózatfejlesztése két részből áll: a Spine módszerből és a hálózat összekötöttségét megerősítő linkfejlesztésből. A munkám során az utóbbit dolgoztam ki, így azt a későbbiekben mutatom be. A két módszer külön-külön vagy együtt is alkalmazható, így biztosítva a még nagyobb megbízhatóságot.

3.1.1. Spine

A hálózat megbízhatóságának növelésének fontos eszköze a hálózati linkek fejlesztése. A Spine egy speciális hálózatfejlesztési módszer amely a megfelelő linkek fejlesztésével növeli a hálózat általános megbízhatóságát.
A Spine ahogy a neve is utal rá egy gerinchálózat/feszítőfa fejlesztésével növeli a hálózat megbízhatóságát. Ehhez a *Diameter-Constrained Reliability* (DCR) módszert használja, melyet röviden ismertetek. Legyen adott egy $G = (V, E)$ gráf, $v \in V$ csúcsokkal és $e \in E$ élekkel. Adott még $K \subset V$ terminálhalmaz, egy vektor $p = (p_1, \ldots, p_n) \in [0, 1]$ és egy d pozitív szám, melyet átmérőnek nevezünk. Feltételezzük, hogy a node-ok nem hibásodnak meg, viszont a linkek egymástól függetlenül meghibásodhatnak (pl. az i-edik link esetén $q_i = 1 - p_i$ valószínűséggel). A diameter-constrained reliability annak a valószínűsége, hogy a terminálok a belőlük képzett részgráfban legfeljebb d link segítségével kapcsolatban maradnak [7].
A Spine célja tehát egy olyan nagymegbízhatóságú "gerinchálózat" kialakítása amely hatékonyan képes támogatni a védelmi módszereket és útvonalválasztást, így megnövekedett végpont-végpont megbízhatóságot biztosítva [2]. A Spine által kiválasztott linkek fejlesztése általában egy elvárt megbízhatósági szintre törekszik, amit bármely két végpont között biztosítani szeretnénk. A fejlesztés a valóságban történhet redundancia hozzáadásával (tartalék eszközök) vagy pl. szabadvezetékeken menő kábelekről földalatti kábelekre váltással.

3.2. Hibamodellezés

A FRADIR keretrendszer középpontjában egy új valószínűségi hibamodellező algoritmus áll, amely képes regionális hibák modellezésére és az SRLG listák létrehozására [25]. Az algoritmus alapötlete, hogy a fellépő katasztrófákat egy körlemezzel modellezzük. Első lépésként a hálózat köré egy befoglaló téglalapot rajzolunk. A befoglaló téglalap oldalait kívánt finomsággal feloszjuk, majd négyzethálószerűen kis téglalapokra osztjuk fel. A kis téglalapok középpontjai lesznek a pontenciális katasztrófa-centrumok. A modellezés során végighaladunk ezeken a pontenciális centrumokon és felmérjük a katasztrófa lehetséges hatását. Az egyes linkek megbízhatóságát úgy vesszük számításba, hogy a link megbízhatósága a katasztrófa centrumától vett távolságot változtatja meg. Vagyis a nagyobb megbízhatóságú linkeket távolabb helyezzük a centrumtól, míg a kisebb megbízhatóságúakat közelebb. Ezután felmérjük az egyes linkek, linkpárok, linkhármasok, stb. meghibásodásának a valószínűségét. Ez n link esetén 2^n linkhalmazt jelent. Mint már említettem nagyon fontos, hogy az SRLG lista megfelelő hosszúságú legyen (a túl rövid nem tartalmaz minden fontos hibát, a túl hosszú sok megkötést jelent, így nem tudunk minden hibát kivédeni). A lista hosszának kézben tartására egy T küszöbérték használható, azokat az SRLG-ket hagyjuk fenn az SRLG listán melyek meghibásodásának a valószínűsége nagyobb, mint T. Ezzel biztosítani tudjuk, hogy a modell pontos és egyszerű legyen egyszerre [25]. A modell paraméterei:

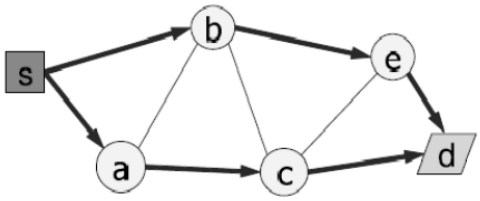
- Radius: A hiba kiterjedése (a regionális hiba egy kör alakú területen fejti ki hatását)
- Küszöbérték: Alsó határ a kiesések valószínűségére
- Felbontás: Milyen finomsággal helyezünk el a katasztrófa-centrumokat

3.3. General Dedicated Protection

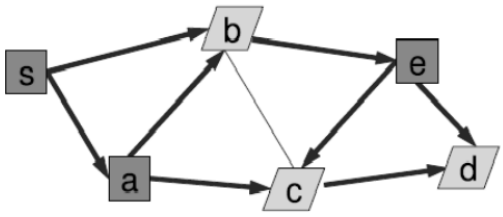
General Dedicated Protection (GDP) egy komplex keretrendszer megbízható útvonalválasztásra, amely többféle védelmi módszert tartalmaz [5]. Ezek közül néhány hálózati kódolást használ, néhány pedig egyszerű útvonalválasztással juttatja el az információt biztonságosan a céljába. Ebben a részben mindkét fajta megoldást röviden ismertetem.

3.3.1. General Dedicated Protection with Network Coding

A GDP azon fajtája mely hálózati kódolást használ (a csomagokon algebrai műveleteket hajt végre) elméletileg azonnali helyreállítását biztosít a kapcsolatnak és nagyon hatékony az erőforrás kihasználása. Az azonnali helyreállítást biztosító megoldások közül ez használja átlagosan a legkevesebb kapacitást. Egyetlen nagy hátránya az implementálhatósági nehézség, mivel a jelenlegi transzport hálózatokban nem biztosított a hálózati kódolásra képes eszközpark.
[28]-ben egy új módszer került bemutatásra, amely kijavítja ezeket a hiányosságokat. A neve *Survivable Routing with Network Coding* (SRNC). Robosztus és azonnali újraépülést biztosít meghatározott hibaesetekre, emellett az erőforráshasználata is az elméleti minimum körül van. Az előző módszerrel ellentétben a felhasználói csomag csak két részre van osztva, ami egyszerű és megvalósítható. Sajnos ennek is problémája, hogy a hálózati elemeknek képesnek kell lenniük a hálózati kódolásra. Ez később a Diversity Coding (Survivable Routing with Diversity Coding) kiegészítéssel már megoldásra került az egyszeres hibák esetén [29]. A hálózati kódolás kivitelezése az optikai rétegen továbbra is nehézkes ezért munkámban azt feltételezem, hogy a hálózati kódolás a hálózat közbülső csomópont-



(a) 1+1 dedicated protection



(b) Generalized Dedicated Protection

3.1. ábra. Csúcspont típusok 1+1 és GDP védelem esetén

jaiban nem lehetséges, így ezen algoritmusokat nem fogom felhasználni a keretrendszer fejlesztése során.

3.3.2. General Dedicated Protection with Routing

A *General Dedicated Protection with Routing* (GDP-R) a GDP azon esete, amikor a megbízható adatátvitelt csak útvonalválasztás segítségével érjük el. A GDP-R bizonyítottan NP-teljes probléma, amennyiben sávszélesség költségre szeretnénk optimalizálni az útvonalválasztást [5]. A problémát egy $P = G, D, F$ hármassal írhatjuk le, ahol

$G = (V, E)$ a hálózat gráf leírása. Minden $e \in E$ élre adott a nemnegatív költségfüggvény ($c : E \to R^+$) és a szabad kapacitás ($k : E \to R^+$).

$D = (s, d, b)$ a forgalmi igények listája, ahol s a forrás, d a nyelő és $b \in \mathbb{N}$ a sávszélesség igény.

F a lehetséges hibák listája. Szükséges, hogy minden hiba esetén s-d összekötött marad a hálózat vagyis minden kiesés esetén vezet út a forrás és a nyelő között.

A GDP-R azonnali helyreállítást biztosít minden hiba esetén, melyet előzetesen megadtunk az SRLG lista formájában. A kivételesen magas megbízhatóságot csak nagyon magas számítási komplexitás árán kaphatunk. A nagy számításigényt az okozza, hogy a GDP-R minden SRLG esetén eltávolítja a hálózatból a hibás linkeket és kiszámolja a legjobb útvonalat. A csúcspontokat a GDP-R modellben a következő szerepekbe osztjuk be:

- Továbbítás (Forwarding): Egyszerű csomagtovábbítás
- Osztás (Splitting): Csomagok duplikálása és két linken való továbbküldése
- Összevonás (Merging): Két megegyező adatfolyam egyesítése

Az 1+1 utas dedikált védelem esetén az egyetlen Splitting csúcspont a forrás és az egyetlen Merging csúcspont a nyelő. A GDP-R esetében bármelyik csúcspont kerülhet bármelyik szerepbe. A GDP-R célja egy olyan minimális költségű útvonal létrehozása, mely bármelyik az SRLG listában szereplő hiba esetén védett. Az útvonal megtalálása egy Egészértékű Programozási Feladat (Integer Linear Program, röviden ILP) megoldásával történik, mely egy minimális költségű részgráfot ad vissza megoldásként. Amennyiben a hálózat minden SRLG kiesése esetén összefüggő marad, a GDP-R biztosan talál megbízható útvonalat a forrás és cél között.

3.4. FRADIR

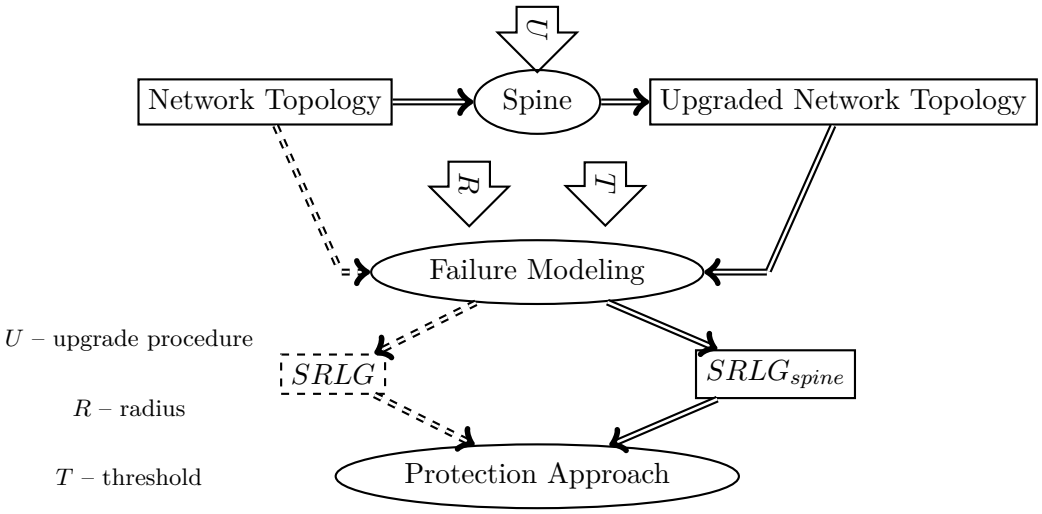
A FRADIR az eddig bemutatott módszereket egyesíti, így több terület eredményeit felhasználva próbál meg egy megbízható, de egyben költséghatékony módszert kialakítani a hálózatok regionális hibák elleni védelmére [25].
A FRADIR-ban használt hálózatot ugyanúgy egy $G = (V, E, c, A)$ gráffal reprezentáljuk, ami egy kétdimenziós térben helyezkedik el. A csúcsok halmaza V, az irányítatlan élek halmaza E. Ezek a valóságban az optikai kapcsolók és az őket összekötő kétirányú optikai kábelek. A gráf minden csúcsa rendelkezik egy pozícióval amit az (x, y) koordinátákkal adunk meg, az élek pedig a csúcsokat összekötő egyenes szakaszokként jelennek meg.
Az élek kiinduló availability értékét ($A(e) \in [0, 1]$) a következőképp számítjuk ki:

$$A(e) = 1 - \frac{MTTR}{MTBF(e)}$$

ahol $MTTR = 24$ h a hibajavítás átlagos ideje, míg a hibák közötti átlagos idő:

$$MTBF(e) = \frac{CC * 365*}{l(e)}$$

ahol CC a Cable Cut metrika és 450 km-nek feltételezzük, $l(e)$ pedig a kábel hossza. Az unavailability értéket a következőképp számoljuk: $U(e) = 1 - A(e)$.
A metódus első lépése a Spine [38] kiszámolása és a hálózat fejlesztése. Ehhez jelenleg a [37]-ban bevezetett hibamodellezést alkalmazza. A Probabilistic SRLG modell – PSRLG – minden linkhez és linkhalmazhoz rendel egy valószínűséget, aszerint, hogy milyen valószínűséggel esnek ki együtt [25]. Végül a General Dedicated Protection with Routing (GDP-R) végzi az útvonalválasztást a PSRLG listának megfelelően [5].



3.2. ábra. A FRADIR koncepciója. A szaggatott vonalak a Spine hálózatfejlesztés nélküli esetet, a folytonos vonalak a Spine hálózatfejlesztést tartalmazó esetet reprezentálják.

A 3.2 ábrán látható a FRADIR koncepciója. A hálózati topológiából kiindulva először a Spine módszert alkalmazzuk egy kívánt megbízhatóság elérése érdekében, majd ezután a hibamodellezés segítségével továbbfejlesztjük a hálózatot. A végső hálózatnak bármely a PSRLG listában szereplő kiesés esetén összefüggőnek kell maradnia. Azután erre a fejlesztett hálózatra végzünk egy hibamodellezést, melynek eredménye egy SRLG lista. A végső lépés a megbízható útvonalválasztás megtervezése, erre a GDP-R módszert használjuk, ami a fejlesztett hálózatban keres minél költséghatékonyabb útvonalat az SRLG listában szereplő hibák kivédésére.

4. fejezet

Idősor analízis és predikció

Az utóbbi időben sok figyelmet kaptak a komplex hálózatok, amelyek olyan természetes és mesterséges rendszerek leírását teszik lehetővé, mint az internet, a légiközlekedési rendszerek, az elektromos hálózatok infrastruktúrája vagy a világháló [16, 3]. A forgalom modellezése valóban alapvető fontosságú a hálózat teljesítményének értékelése és a hálózati vezérlési rendszerének tervezésében, amely kulcsfontosságú a nagysebességű hálózatok sikeres működéséhez [21]. A hálózati forgalom kiértékelése, a trendek megértése és a jövőbeli forgalom becslése lehetővé teszi a hálózattervezési módszereink javítását illetve az útvonalválasztási algoritmusok fejlesztését. A hálózatüzemeltetőknek így kiemelten fontos a forgalom minél jobb megértése. Egy pontos forgalom-előrejelző rendszer elengedhetetlen a helyes hálózatmenedzsmenthez, segít az erőforrásfoglalás megtervezésében, hosszútávú kapacitástervezésben, hálózatfejlesztésben és anomália detekcióban.

A halózatok forgalmát ugyanúgy idősorokkal írjuk le, vagyis időben egymás után következő megfigyelések sorozatával, mint például a villamosenergia-hálózatban az egyes napokon keletkező fogyasztást. Ebben a fejezetben bemutatom a legelterjedtebb idősor előrejelzési stratégiákat és elterjedt megoldásokat. A megoldások közül az általam is alkalmazott neurális hálózatokra térek ki részletesebben.

4.1. Idősor előrejelzési stratégiák

Az idősor-előrejelzési probléma a jövőbeni értékek előrejelzése az idősorok korábbi értékei és jelenlegi értéke alapján. Az idősorok korábbi értékeit és aktuális értékét az előrejelzési modell bemeneteiként használjuk. Általában szükség van egy egylépéses előrejelzésre, erre rövid távú előrejelzésként hivatkozunk. De amikor már többlépéses előrejelzésekre van szükség, akkor ezt hosszútávú előrejelzési problémának nevezzük.

A rövid távú idősor-előrejelzéssel ellentétben a hosszú távú előrejelzés általában különféle forrásokból származó növekvő bizonytalanságokkal néz szembe. Például a hibák halmozódása és az információhiány megnehezíti az előrejelzést. A hosszú távú előrejelzés során, többlépéses előrejelzést használva számos alternatíva áll rendelkezésünkre a modellek felépítéséhez. A következő szakaszokban az előrejelzési stratégiák két változatát mutatom be és hasonlítom össze: a közvetlen és a rekurzív előrejelzési stratégiákat.

4.1.1. Rekurzív előrejelzési stratégia

Az idősor értékeinek több lépéses előrejelzésére a rekurzív stratégia tűnik a leginkább intuitív és egyszerű módszernek. A becsült értékeket ismert adatokként használja a követ-

kező értékek előrejelzéséhez. Részletesebben, a modell úgy állítható elő, hogy először egy lépéssel előrejelzi az idősort:

$$\hat{y}_{t+1} = f_1(y_t, y_{t-1}, \ldots, y_{t-M+1}) \tag{4.1}$$

, ahol M a bemenetek számát jelöli. Nyilvánvaló, hogy a bemenet nem csak szabályos időközönkénti megfigyelésekből álló vektor lehet, de a jelölés egyszerűsége és egységessége érdekében erre nem térek ki. A (t+2)-beli érték megjóslásához ugyanezt a modellt használjuk:

$$\hat{y}_{t+2} = f_1(\hat{y}_{t+1}, y_t, \ldots, y_{t-M+2}) \tag{4.2}$$

A 4.2 egyenletben az $\hat{y}_{t+1}$-et használjuk az ismeretlen valós érték helyett. Az N-lépéses előrejelzéshez az $\hat{y}_{t+2}$-től $\hat{y}_{t+N}$-ig iteratívan jelezzük előre az értékeket. Tehát, amikor a bemenet hossza (M) nagyobb, mint N, akkor M-N valós adat van a bemenetben, hogy előrejelezze a N-edik lépést. De amikor N nagyobb mint M, akkor már az összes bemenet becsült érték. Minél több becsült érték jelenik meg a bemenetben annál kisebb lesz az előrejelzés pontossága.

4.1.2. Közvetlen előrejelzési stratégia

A hosszú távú előrejelzés másik stratégiája a közvetlen előrejelzési stratégia. Az N-lépéses előrejelzéshez a modell:

$$\hat{y}_{t+n} = f_n(y_t, y_{t-1}, \ldots, y_{t-M+1}), \text{ ahol } \quad 1 \leq n \leq N \tag{4.3}$$

Itt az előrejelzést csak mért értékek alapján végezzük el minden n-re, a $t + n$-edik időpillanat becslését ugyanabból a bemenetből számoljuk minden n-re. A becsült értékek hibái így nem halmozódnak fel az egyre távolabbi becslésekben, viszont megnöveli az előrejelzés bonyolultságát, mivel N különböző modellt kell felépíteni. De [34] szerint sokkal pontosabb előrejelzés érhető el vele.

4.2. Idősor előrejelzési megoldások

A legismertebb hagyományos idősor előrejelzési modellek közül a legismertebbek mind alkalmazva voltak hálózati forgalom előrejelzésre. A Holt-Winters módszer rövid és hosszútávú szabályosságok alapján modellezi az idősort és [11]-ban sikeresen alkalmazták forgalomelőrejelzésre. Az ARIMA egy sokkal komplexebb módszer mely stacionárius idősorok előrejelzését teszi lehetővé. Három paramétere van, így a modelre ARIMA(p,d,q) néven hivatkozhatunk, ahol p a becsléshez használt előző értékek száma, d a stacionaritás eléréséhez elvégzett differenciálások száma és q az előzőleg elkövetett hibavektor hossza. Van néhány speciális eset, amely széles körben ismert: az ARIMA(1,0,0) az elsőrendű előrejelző modell, az ARIMA(0,1,0) a véletlen séta, az ARIMA(0,1,1) pedig az exponenciális simítás. [24]-ben a szerzők különböző időskálákon keresnek trendeket a forgalomban, majd a trendeket az ARIMA (AutoRegressive Integrated Moving Average) segítségével modellezik. A forgalom előrejelzés alapján javaslatokat tesznek, hogy az IP gerinchálózatban mikor és hol kell linkeket fejleszteni vagy új linkeket létrehozni. [15]-ban egy hálózati támadásfelismerő rendszert mutattak be. A rendszer a forgalom statisztikai tulajdonságaira koncentrál és dinamikus küszöbértékek segitségével ismeri fel a támadásokat.

Az idősor előrejelzés legújabb eredményeit már főleg neurális hálózatokkal érik el. Népszerűségüket annak köszönhetik, hogy viszonylag egyszerű modellek már nagyon jó

eredményeket produkáltak előrejelzési feladatokra. A mély neurális hálók elterjedése és a tanítási technikák fejlődése sok területnek adott lökést, köztük az idősoranalízisnek is. Ennek megfelelően a mérnöki tudományok idősorainak előrejelzésére is szívesen használják. A következőkben egy rövid áttekintést adok a neurális hálózatokról, különös tekintettel az idősoranalízisban alkalmazott típusokról.

4.2.1. Neurális hálózatok

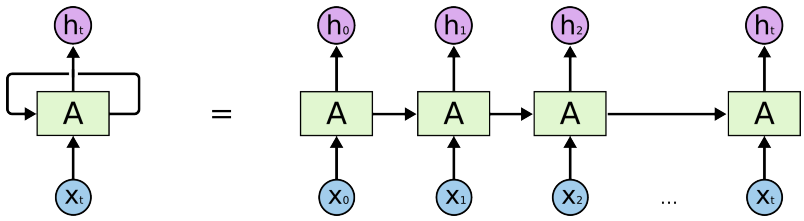
A mesterséges neurális hálók a biológiai neurális rendszerek elvére felépített számítógépes rendszerek. A hálók több, egymáshoz kapcsolódó és párhuzamosan dolgozó processzorból (neuronból) állnak, és ily módon próbálják utánozni a biológiai idegrendszer információfelvételének és -feldolgozásának módját. A mesterséges neurális hálók a hagyományos algoritmikus eljárások helyett más módon, tanulással nyerik el azt a képességüket, hogy bizonyos feladatokat meg tudjanak oldani. Ez nagy előnyt jelent akkor, ha olyan bemenetre kell választ adni, amivel eddig nem találkoztak, mivel a tanulás által valamekkora általánosító képességre is szert tesznek.

A neurális hálók alapeleme az elemi neuron. Az elemi neuron egy több-bemenetű, egykimenetű eszköz, ahol a kimenet a bemenetek lineáris kombinációjaként előálló közbenső érték függvénye. Ez a függvény az aktivációs függvény. A neurális hálózat rétegekből áll, melyeket neuronokból épitünk fel. Általában kapcsolat csak az egyes rétegek között van, rétegeken belül nincs, így az adott réteg minden neuronja ugyanazt a bemenetet kapja meg. A neurális hálózat paraméterei a lineáris kombináció képzéséhez használt súlyok és eltolás (bias) értékek.

A hálózat tanítása tekinthető egy függvény approximációs feladatnak, melyben a hálózat paramétereit úgy szeretnénk megváltoztatni, hogy a hálózat kimenete és az elvárt kimenet között meghatározott hibafüggvényt minimálisra csökkentsük. Ez egy sokdimenziós optimalizálási feladat, melyre csak közelítő megoldást lehet adni. A leggyakrabban alkalmazott tanító eljárás a hibavisszaterjesztés. Ez a gradiensszámítás segítségével kiszámolja a hibafüggvény gradiensét az egyes súlyok szerint, majd ennek a gradiensnek megfelelően változtatja meg a súlyokat a hibafüggvény csökkentéséhez. A hálózatok tanítása egy nagyon komplex témakör, de ennél bővebben sajnos nem térhetek ki rá.

4.2.1.1. Visszacsatolt hálózatok

A neurális hálózatok egy speciális fajtája a visszacsatolt neurális hálózatok (Recurrent Neural Networks - RNNs), melyekben a neuronok előző bemenetre adott válasza visszacsatolódik, megjelenik egy plusz bemenetként. Az RNN-re úgy is gondolhatunk, mint egy hálózat időbeli fejlődésére.



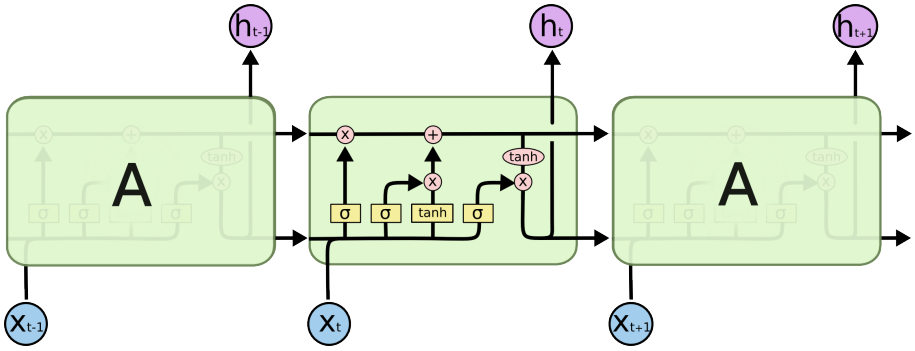
4.1. ábra. Egy visszacsatolt neuron időbeli kiterítése [23]

Ez a láncstruktúra is mutatja, hogy az RNN nagyon közeli kapcsolatban áll a sorozatokkal, ennek megfelelően az utóbbi évtizedekben nagy sikereket értek el velük beszédfelismerésben, fordításban és más sorozat-sorozat leképezést igénylő feladatokban. A hagyományos RNN egyetlen gyengéje a hosszútávú kapcsolatok kezelése, ennek a problémának a

megoldására fejlesztették ki az LSTM (Long Short Term Memory) hálózatokat [14], amivel az előzőleg említett eredmények nagy részét elérték.

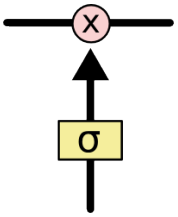
4.2.1.2. LSTM hálózatok

A hosszú időbeli kapcsolatok megtanulása érdekében az LSTM neuronjai sokkal komplexebbek a sima RNN neuronjaihoz képest, ahogy az a 4.2 ábrán is látható.



4.2. ábra. Egy LSTM neuron időbeli kiterítése [23]

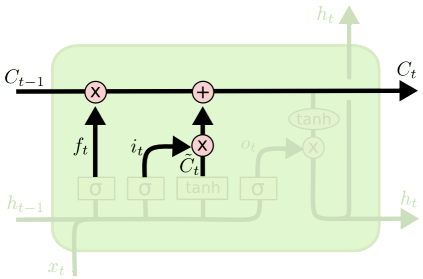
Miután a sárga dobozok elemi neuronokat reprezentálnak, így rögtön látható, hogy 1 LSTM neuron nagyon sok változtatható paraméterrel rendelkezik. A legfontosabb azonban a neuron tetején végigfutó, a neuron állapotát reprezentáló vonal. A neuronnak megvan a lehetősége az állapot változtatására, viszont ez nagyon szigorúan szabályozva van az úgynevezett kapuk által. A kapuk határozzák meg az információ átjutását elemenkénti szorzat segítségével. A kapuk vezérlését szigmoid függvények látják el, ahogy az a 4.3 ábrán látható. A szigmoid függvény 0 és 1 közötti számokat ad eredményül, ami így szorzás esetén meghatározza, hogy mely komponensek jutnak át.



4.3. ábra. Az LSTM-ben használt kapu elemi modellje [23]

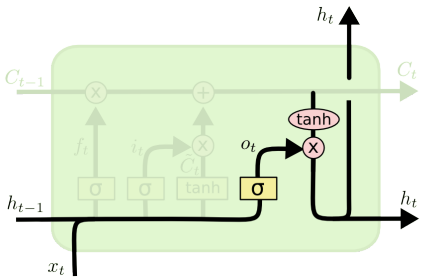
Az első kapu határozza meg, hogy a cellaállapotból mit dobunk el, ezért felejtő kapunak is nevezik. A második kapu meghatározza, hogy milyen információkat szeretnénk tárolni a cellaállapotban. Először egy szigmoid réteg meghatározza a frissítendő értékeket, majd egy tanh réteg létrehozza az új állapotvektort, ezután az elemenkénti szorzatukat hozzáadva az állapotvektorhoz frissítjük a neuron állapotát. A két kapu által meghatározott állapotfrissítés a 4.4 ábrán látható.

A kimenet képzése hasonló az állapotfrissítéshez. A neuron kimenetét a cella állapota fogja meghatározni, viszont ezt a bemenet alapján szűrjük. A cella állapotvektorának értékeit először egy tanh függvény segítségével a (-1,1) tartományba képezzük le, majd egy a bemenet által vezérelt sigmoid függvénnyel megszűrjük, ahogy azt a 4.5 ábrán láthatjuk.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

4.4. ábra. Az LSTM neuron állapotfrissítése két lépésben [23]

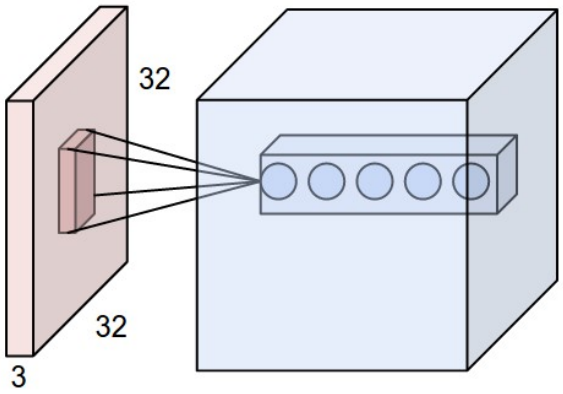


$$o_t = \sigma\left(W_o\ [\,h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

4.5. ábra. Az LSTM neuron kimeneténekképzése [23]

4.2.1.3. Konvolúciós neurális hálózatok (Convolutional Neural Networks - CNNs)

A CNN-ek eredetileg képfelismerésre lettek kifejlesztje, mivel a hagyományos hálózatok nagyon rosszul skálázódnak a bemenet növekedésével. A hagyományos felismerési eljárások először a képet valamilyen sokdimenziós jellemzőtérbe képzik le, majd ebben a jellemzőtérben próbálják meg az egyes osztályok (pl. autó, kutya, stb.) különbözőségeit megkeresni. Ennél a módszernél viszont a jellemzőket kézzel kell meghatározni, amit csak tapasztalati úton lehet, így sok hibát rejt magában. A konvolúciós hálók ennek a problémának a megoldására jöttek létre. Az egyes neuronok egyszerre csak a bemenet egy kis részéhez kapcsolódnak és végzik el rajta a konvolúciót. Az egész képen egy ablak tologatásával haladunk végig, így a neuronok mindig csak egy meghatározott részt látnak belőle amire elvégzik a műveletet (4.6. ábra). Ha például 10 neuront teszünk a konvolúciós rétegbe és a bemenetből egyszerre mindegyik egy 3×3-as részt lát, akkor ha ezt az ablakot pixelenként végigmozgatjuk egy 32×32-es képen akkor kimenetként egy $30 \times 30 \times 10$ méretű mátrixot kapunk.



4.6. ábra. A konvolúciós művelet elvégzése egy lokális régióban [18]

A tanítás során az egyes neuronok súlymátrixát változtatjuk, tehát filtereket tanítunk meg nekik, így a hálózat a tanítás alapján nyeri ki a képek legjobb jellemzőit. A konvolúciós rétegek egymás mögé helyezésével a későbbi réteg neuronjai a bemenet egyre nagyobb részét kapják meg valamilyen formában bemenetként ami így egyre nagyobb alakzatok felismerését teszi lehetővé. A konvolúciós rétegek elvégzik a kép jellemzőinek a kinyerését, viszont az osztályozáshoz általában egy teljesen előrecsatolt hálózatot használnak. Itt viszont a neuronok minden bemenetet megkapnak, így a konvolúciós rétegek kimenetének dimenzióit mindenképpen csökkenteni kell. A képekben jelenlévő nagy redundancia miatt ez nem jár nagy információvesztéssel. A konvolúciós réteghez hasonlóan végigcsúsztatunk egy ablakot a "képen" és csak az ablakban lévő maximális értéket tartjuk meg, ez a Max Pooling réteg feladata, ami egy egyszerű mintavételezés. Több ilyen rétegtípus van, a különbözőségüket a megtartani kívánt érték kiválasztása adja. Az idősor analízisben is egyre gyakoribb a CNN használata, ami annak köszönhető, hogy a konvolúciós rétegek automatikusan képesek a bemenet komplex jellemzőit megtanulni. Mivel az idősorok 1 dimenzió szerint változnak, ezért a konvolúziós ablak eltolását is csak 1 dimenzióban kell megtennünk a képek esetén megszokott 2 dimenziós eltoláshoz képest. A másik különbség még a hagyományos CNN-ekhez képest a bemenet mérete, az idősorokban sokkal kisebb redundancia van jelen, mint a képekben, így kevesebb dimenziócsökkentő rétegre van szükség.

5. fejezet

Hálózatfejlesztés az összekapcsoltság megőrzésére

A regionális hibák kivédésének egyik legnagyobb nehézsége, hogy egyszerre több link is meghibásodhat. Ez gyakran a hálózat szétszakadásához (több komponensre való szétesésé-hez) vezet, ami viszont megengedhetetlen transzport hálózatok esetében. Ennek elkerülése érdekében célzottan fejlesztenünk kell a hálózatot. Ez már a FRADIR első verziójában [25] is kiderült, hiszen a szerzők kiemelték, hogy a Spine féle hálózati fejlesztése ellenére is sok nem védhető hibát maradt a hálózatban, így azokat a GDP-R sem volt képes védeni. A feladatunk tehát az, hogy a hálózatot olyan mértékben fejlesszük, hogy a hibamodellezéskor meghatározott PSRLG lista egyik eleme se okozhassa a hálózat szétszakadását. Mivel a hálózat minden hiba esetén fennálló összefüggősége biztosíték arra, hogy a GDP-R képes egy megfelelően védett útvonal kialakítására. Ebben a fejezetben bemutatom a hálózat összefüggőségét biztositó fejlesztési metódusaimat, melyek a FRADIR-t egészítik ki, ami ezzel együtt már képes minden regionális hiba esetén biztosítani a kapcsolatok folytonosságát.

5.1. Hálózatfejlesztési módszer a vágást okozó hibák kivédésére

A 1. algoritmus írja le az általam kitalált iteratív hálózatfejlesztésnek a működését. Az algoritmus alapja a vágást okozó SRLG-k megkeresése és eltávolítása a linkek fejlesztése által. A költség a legalapvetőbb dolog amit minimalizáni szeretnénk, ehhez pedig az kell, hogy a lehető legkevesebb számú élt kelljen fejlesztenünk.
Az alapötlet az, hogy keressünk egy olyan minimális élhalmazt, melynél minden vágást okozó SRLG tartalmaz legalább egy élt a halmazból és fejlesszük ezeket. Ennek a problémának a neve *set cover problem*, melynek alapfeladata az, hogy részhalmazaival akarunk lefedni egy halmazt és a legkevesebb olyan részhalmazt keressük amivel ez megtehető. A mi esetünkben az SRLG-k halmazát szeretnénk lefedni egy élhalmazzal, ahol az élekhez az SRLG-k azon részhalmaza tartozik, melynek része az él. A feladat megoldására egy mohó (*Greedy*) algoritmust implementáltam, ami minden lépésben azt az élt veszi be, amely a legtöbb még nem lefedett SRLG-ben van benne. *Ez a megoldás polinomiális becslést ad nekünk a problémára.* A megoldásunk $H(n)$ tehát egy becslés, $H(n) = \sum_1^n 1/\eta < \ln(n)+1$, ahol n a vágást okozó SRLG-k száma.
Egy másik ötlet lehet a feszítőfa tulajdonságainak kihasználása. Ismert, hogy a legkevesebb élből álló összefüggő gráf a fa és a feszítőfa a gráf összes csúcsát tartalmazza. Így ha a vágást okozó SRLG-k egyesített élhalmazának és egy feszítőfa élhalmazának a metszetét képezzük, akkor nyilvánvalóan minden ilyen SRLG-ből lesz benne él.

Algorithm 1: Hálózatfejlesztési módszer a vágást okozó hibák kivédésére

Input: $G = (V, E, a)$, $\mathcal{F}$ – list of SRLGs, $\mathcal{S}$ – set of edges on the spine
Result: $G' = (V, E, a')$ graph with improved availability values

```
1  begin
2    repeat
       // Calculate SRLGs of G
3      G ⟶ F
       // Find cut sets
4      Define edge set E* = ∅
       // Iterate over the SRLG list
5      for f ∈ F do
6        Define graph G* = G \ f
7        if G* is not connected then
           // Add edges of the SRLG
8          E* = E* ∪ f;
       // Remove edges off the spine
9      E* = E* ∩ S;
       // Minimal cover of SRLGs with edges
10     E* ⟶ E*_min
       // Upgrade edges
11     for e ∈ E*_min do
12       a_{k-1}(e) ⟶ a_k(e)
13   until E* ≠ ∅;
```

A két módszer együtt is alkalmazható először a vágást okozó SRLG-k élhalmazának és egy feszítőfának a metszetét képezzük, majd ezen éleken még elvégzünk egy minimális fedés keresést.
A két ötlet kombinálásával 4 különböző módszert kapunk a fejlesztendő élhalmaz meghatározására:

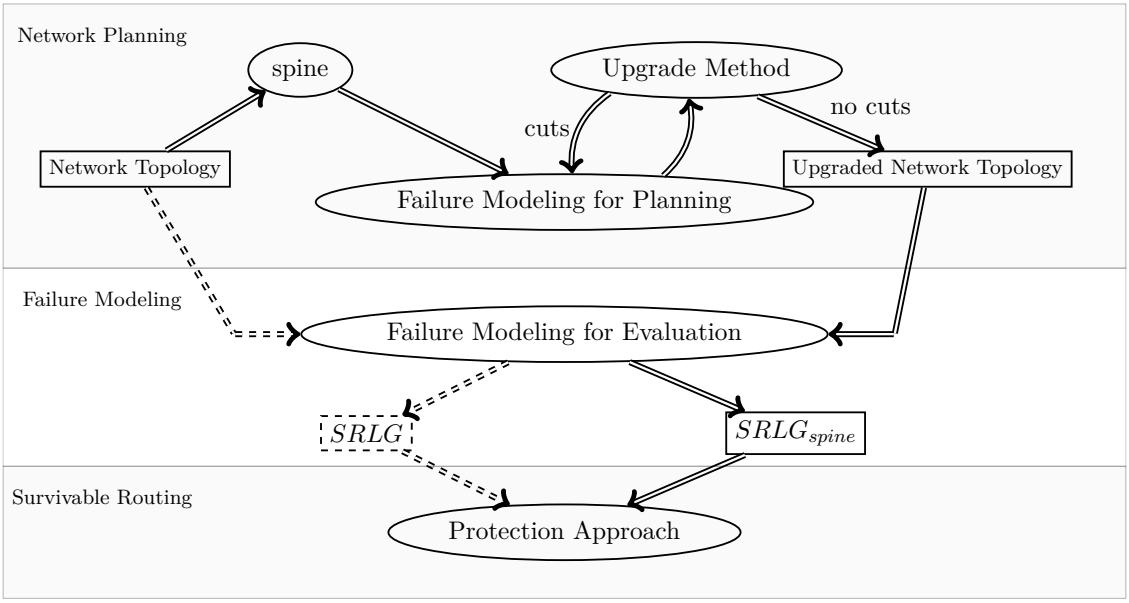
1. Minden olyan él fejlesztése, amely legalább egy vágás-SRLG-ben benne van (1. módszer).

2. Legkisebb olyan élhalmaz fejlesztése, amelyből minden vágás-SRLG legalább egy élt tartalmaz (2. módszer).

3. Egy feszítőfa minden olyan élének fejlesztése, amely legalább egy vágás-SRLG-ben benne van(3. módszer).

4. Legkisebb olyan élhalmaz fejlesztése egy feszítőfának, amelyből minden vágás-SRLG legalább egy élt tartalmaz (4. módszer).

Az élek fejlesztése meghatározott rendelkezésre állású állapotokra lehetséges. Ezek az értékek a következőek: $a_1 = 0.999$, $a_2 = 0.9995$, $a_3 = 0.9999$ $a_4 = 0.99995$ $a_5 = 0.99999$ and $a_6 = 0.999995$ ($a_0(e)$ a link eredeti rendelkezésre állása, még minden fejlesztés előtt). A fejlesztések alkalmával a linkeket mindig a következő szintre fejlesztjük. Vagyis az a_{k-1} rendelkezésre állású élt a_k rendelkezésre állásúra fejlesztjük, ahol $k = 1, \ldots, K'$ és $K' = 6$. Az algoritmus bemeneti paramétere a hálózatot reprezentáló G gráf és a feszítőfa (vagyis a Spine). Az első lépés a regionális hibák modellezése, amelynek eredménye egy SRLG lista. Ezután létrehozunk egy E^* üres élhalmazt mely a fejlesztendő éleket fogja tartalmazni. Ezután végigiterálunk az SRLG listán, és minden SRLG kiesése esetén ellenőrizzük a hálózat összefüggőségét. Amennyiben az SRLG kiesése esetén a hálózat nem lesz összefüggő, az SRLG éleit hozzáadjuk E^*-hoz. Ezután a fent felsoroltak szerint négyféleképp képezhetjük



5.1. ábra. A FRADIR koncepciója. A szaggatott vonalak a hálózatfejlesztés nélküli, a folytonos vonalak a hálózatfejlesztést alkalmazó változatot jelentik [26].

a fejlesztendő élek végső listáját. Vagy vesszük E^* metszetét a feszítőfa élhalmazával, vagy nem. Vagy megoldjuk a minimális fedési feladatot az E^* élhalmazra és a vágást-SRLG-k halmazára vagy nem.
Ezután az E^*-ban maradt éleken végigiterálva felfejlesztjük azokat egy rendelkezésre állási szinttel magasabbra. Amennyiben E^* üres halmaz volt, mivel nem volt egyetlen vágást okozó SRLG sem, akkor végeztünk a hálózatfejlesztéssel és a jelenlegi hálózat megfelel az elvárásainknak.
Ha végeztünk élfejlesztést ($E^* \neq \emptyset$), akkor újrakezdjük az algoritmust ezzel a fejlesztett hálózattal és egészen addig folytatjuk a fejlesztést amíg a vágást okozó SRLG-k el nem tűnnek az SRLG listából vagyis eléggé lecsökken a valószínűségük.

A FRADIR hálózatfejlesztéssel való kiegészítését a 5.1. ábra mutatja be. Első lépésként a Spine hálózatfejlesztési módszerrel kialakítunk egy nagy megbízhatóságú feszítőfát, ezután a regionális hibamodellezés segítségével meghatározunk egy védendő SRLG-ket tartalmazó listát. Az SRLG listából kinyerjük a vágást okozó SRLG-ket, majd a 4 bemutatott eljárás közül az egyikkel meghatározzuk a fejlesztendő élhalmazt. Ezután elvégezzük a fejlesztést és újra elvégezzük a hibamodellezést. Ha még mindig vannak vágást okozó SRLG-k, akkor újra meghatározzuk a fejlesztendő éleket és fejlesztjük őket. Ezt addig folytatjuk, amíg olyan hálózatot nem kapunk, amely regionális hibák esetén is összefüggő marad. Ebben a hálózatban ezután megbízható útvonalválasztás segítségével védjük a kapcsolatokat.

Ebben a fejezetben megmutattam, hogy lehet megőrizni a hálózatok összefüggőségét regionális hibák esetén és ez hogyan integrálható a FRADIR rendszerbe. Ezzel a kiegészítéssel a FRADIR már minden vizsgált regionális hiba esetén képes védeni a kapcsolatot. A következő fejezetben bemutatom, hogy növelhető a kapcsolatok rendelkezésre állása akkor ha pontos becslést tudunk adni a hálózat egyes linkjein várható szabadkapacitásra.

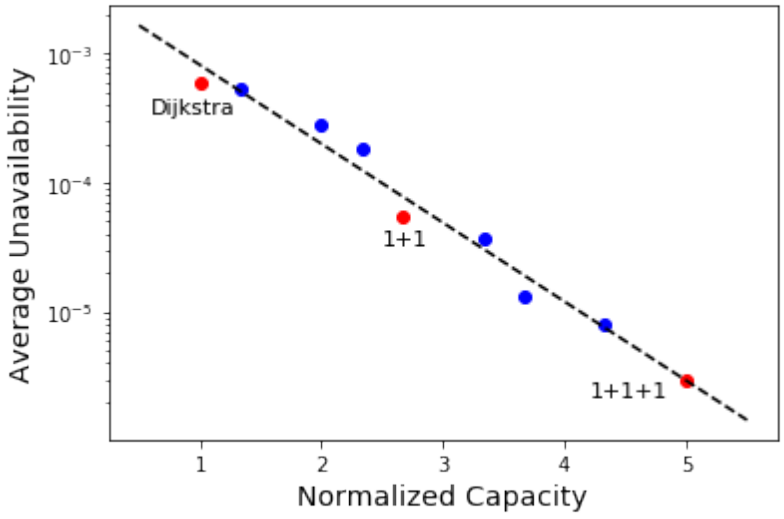
6. fejezet

A hálózati forgalom becslése

Az előzőekben megmutattam, hogy lehet megőrizni a hálózatok összefüggőségét regionális hibák esetén, most pedig bemutatom, hogy lehet növelni a kapcsolatok rendelkezésre állását akkor ha pontos becslést tudunk adni a hálózat egyes linkjein várható szabad kapacitásra. Az esettanulmányhoz egy olyan hálózat forgalmi adatait használtam, mely világméretű és forgalmi adatai szabadon elérhetőek.

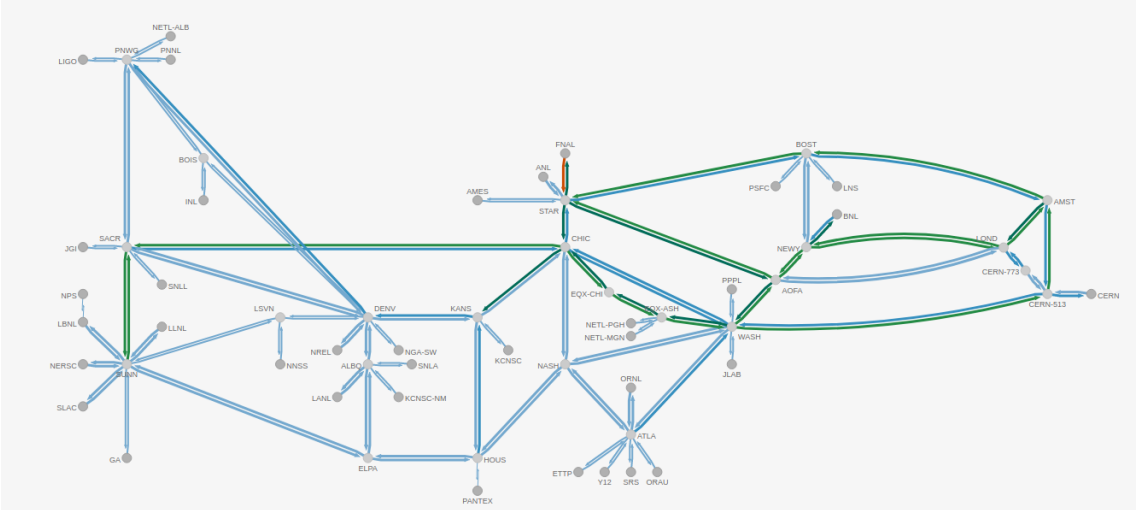
6.1. A rendelkezésre állás és a szabad kapacitás kapcsolata

A bevezetőben már leírtam, hogy a megbízható útvonalválasztáshoz használható algoritmusokat meghatározza a hálózatban elérhető szabad kapacitás. Egy adott összeköttetésre az algoritmusok más rendelkezésre állású útvonalakat javasolnak, melyek felhasznált kapacitása is különböző. Általánosságban elmondható, hogy minél több kapacitás elérhető egy adott összeköttetés számára, annál nagyobb rendelkezésre állású kapcsolatot lehet kiépíteni.



6.1. ábra. A különböző megbízható útvonalválasztási algoritmusok normalizált átlagos kapacitáshasználata és az ezzel elért átlagos rendelkezésre állás

A 6.1 ábrán 3 útvonalválasztási módszert emeltem ki. A Dijkstra a legrövidebb útvonalat keresi meg, így a kapacitáshasználata minimális. Az 1+1, már 2 útvonalon juttatja el az információt a célba így sokkal kisebb a rendelkezésre nem állása, de ez több mint



6.2. ábra. Az ESnet felépítése [1]

kétszer akkora kapacitáshasználattal jár. A legnagyobb rendelkezésre állású útvonalat az 1+1+1 metódus képes elérni, de itt a hálózat architektúrája már nagy korlátozást jelent, mivel ez megköveteli a 3 éldiszjunkt út meglétét a forrás és nyelő között. A köztes pontok különböző szabad kapacitáshoz tartozó GDP-R számolta útvonalak. Az ábra mutatja, hogy ha jól meg tudjuk becsülni az egyes linkek szabad kapacitását, akkor tudunk becslést adni a kapcsolatnak biztosítható rendelkezésre állásról (QoS szintről). Amennyiben a biztosítható QoS szint kielégíti a kapcsolat igényeit, akkor létrehozzuk a kapcsolatot.

6.2. Energy Sciences Network (ESnet)

Az ESnet a világ leggyorsabb tudományos hálózata, mely összeköttetést biztosít a világ vezető tudományos laboratóriumai között. A Lawrence Berkeley Nemzetközi Laboratórium által működtetett hálózat 100 Gbit/s sebességű optikai linkeket használ az USA-ban és az EU-ban, amiket összesen 450 Gbit/s transzatlantic kapacitás köt össze. A hálózat havonta több mint 100 PB tudományos adat szállítását végzi el. Az ESnet-hez hasonló tudományos hálózatok karakterisztikája eltér a hagyományos Internet karakterisztikájától, melynek okait a következő alfejezetben fejtem ki.

6.2.1. A forgalmat alakító tényezők

A tudományos hálózatok, mint például az LHC Open Network Environment (LHCONE) hálózat, jól szervezett közösségek. A hálózathoz csatlakozás csak ismert szereplőknek lehetséges, akik előtte elfogadták az üzemeltetők által meghatározott speciális szabályokat. Amíg a hagyományos internetszolgáltatók emberi felhasználók forgalmát továbbítják, addig a tudományos hálózatok főleg gép-gép közti forgalmat továbbítanak. Ennek köszönhetően az emberek általi forgalomban megfigyelhető trendek (pl. esti csúcsforgalom) nem figyelhetők meg az ilyen hálózatokban. Ennek egy másik oka lehet a hálózat hatalmas mérete, hiszen a hálózat több időzónán ível át (a két legtávolabbi pont között 9 óra időkülönbség van). Tehát míg az Atlanti-óceánon átívelő linken körülbelül 7000 km-t utazik az információ, addig a hagyományos forgalom kb. 280 km-t tesz meg a legközelebbi CDN-ig.

Az ESnet forgalma exponenciálisan évente 36.6 %-kal nő, ez nagyobb növekedés, mint a hagyományos hálózatokon előrejelzett 26 % [9]. Ez a növekedés nagy kihívás elé állítja a forgalombecslő és kapacitástervező rendszereket.

A forgalom típusa is jelentősen eltér a hagyományos internetforgalomtól, melynek nagy részét a webböngészés és a multimédia stream-elés teszi ki. A tudományos hálózatokban az adatokat általában valamilyen fájlátviteli protokoll segítségével mozgatják, mint amilyen a GridFTP vagy az XRootD. A nagy adatmennyiségek miatt a csomagok mérete is nagyobb (átlagosan 1500 byte csomagonként).

6.2.2. A forgalom letöltése és előfeldolgozása

A hálózat architektúrája a linkek kapacitása és aktuális forgalma szabadon elérhető az ESnet honlapján[1], ahol lehetőség van áttekinteni a forgalom alakulását akár 1 hónapra visszamenőleg. A korábbi forgalmi adatok két időpont között lekérhetőek egy adatbázisból a link és interface nevének valamint a két időpontnak a megadásával. A forgalom fél perces felbontásban érhető el, ami ugyancsak megnehezíti a forgalom előrejelzését. Az adatbázis egy linkre egy időbélyeggel két értéket tartalmaz, a két irányban áthaladt adatmennyiséget az előző időponthoz képest.

Mivel a hálózat folyamatosan fejlődik, így az egyes linkek forgalmi adatai más kezdőidőponttól kezdve elérhetőek, az új linkek üzembehelyezése miatt. A hálózat fejlesztése és a hibák fellépése ezért elég sok érvénytelen forgalmi adatot eredményez. Ezek kiküszöböléséhez különböző adattisztító eljárásokat kellett alkalmaznom, illetve olyan formába kellett átalakítanom amivel egy neurális hálózat betanítható.

A linken menő kétirányú forgalmat teljesen külön kezeltem. Az önmagában álló hibás értékeket a két szomszédjának az átlagaként határoztam meg. Amennyiben nagyobb kiesés történt a kiesést kivágtam az adathalmazból és az így kapott két részt külön kezeltem a folytatásban. A forgalmat 0 és 1 közé skáláztam be, mivel a bemenet normalizálása segíti a neurális hálózat tanítását [41].

A normalizálás után kapott idősort bemenet-kimenet párokká kellett alakítani, ehhez meg kellett határoznom, hogy a hálózat bemenetként milyen hosszú idősort lásson. Mivel viszonylag hosszabb távú forgalmat szeretnék előrejelezni 10-15 perc, ami félperces felbontásnál 20-30 lépés, ezért több időskálán is kipróbáltam a forgalmat, hogy megnézzem melyiken milyen eredményt tudok elérni. A bemenet és a kimenet hosszát ezért a különböző időskáláknak megfelelően választottam meg. Például ha a lépésköz 5 perc, akkor a kimenetet 4 hosszúra választottam, míg a bemenetet 64-re.

6.2.3. A forgalom előrejelezhetősége

Egy idősor előrejelezhetőségének meghatározása nagyon fontos a megfelelő rendszer kialakításához, hiszen előrevetíti, hogy milyen eredményt várhatunk el a rendszertől. Az olyan entrópia mértékek, mint például a Minta Entrópia (Sample Entropy [33]), becslést adnak az idősorban fellelhető regularitásról vagy hasonlóságról. A magasabb Minta Entrópia nagyobb rendezetlenséget, véletlenszerűséget és rendszerkomplexitást jelent. Egyszerűbben előrejelezhető idősoroknál ez az érték 0.4 alatt van, a nagyon nehezen előrejelezhetőknél pedig 1.0 felett. Az ESnet különböző linkjeinek forgalmán mért Minta Entrópia 0.6 és 0.9 között mozgott, ami azt jelenti hogy ezen idősorok előrejelzése nehéz, de nem lehetetlen vállalkozás.

6.3. A neurális hálózat felépítése

Mivel idősoranalízisre a legtöbb jó eredményt LSTM neurális hálózatokkal érték el, ezért úgy gondoltam, hogy ezt a típust mindenképpen érdemes kipróbálni forgalmi predikcióra. Egy másik új megközelítés a CNN hálózatok használata, amelyek képesek magasszintű

[1] https://my.es.net

jellemzőket kinyerni a konvolúciós rétegek segítségével. Ezekkel a hálózatokkal is sokan értek el jó eredményeket idősor előrejelzésre. Mivel a konvolúciós neurális hálók általában klasszifikációs problémák megoldására vannak kifejlesztve így a jellemzők kinyerése utáni döntést egy előrecsatolt hálózat végzi el. Mivel itt egy idő jellemzői alapján kell előrejelzést adnunk, ezért kipróbáltam egy olyan összeállítást, amelyben a konvolúciós rétegek mögé az egyszerű előrecsatolt hálózat helyett egy LSTM-et kapcsolok. Ezt az architektúrát már mások is sikeresen alkalmazták például anomáliadetekcióra [20]. A konvolúciós hálózatok használatának másik előnye a gyorsabb taníthatóság. Mivel nem állt rendelkezésemre végtelen számítási kapacitás, ezért összhangot próbáltam keresni a pontosság és tanítási idő között.

6.3.1. A hibafüggvény

Mivel a becsült szabad kapacitás nagyban befolyásolja, milyen QoS szintet biztosítunk egy adott kapcsolatnak, így a becslésnek nagyon pontosnak kell lennie. Ha viszont hibás az előrejelzés, akkor nagyon nem mindegy hogy alul vagy vagy felülbecsüljük a forgalmat. Ha alulbecsüljük a forgalmat akkor olyan szabad kapacitásokat használunk fel egyes kapcsolatok megbízhatóbbá tételére amik nem léteznek, a túl nagy forgalom pedig csomagvesztésekhez vezethet. A csomagvesztés legrosszabb esetben a kapcsolat megszakadásához vezethet, amit kritikus alkalmazások esetén nem engedhetünk meg.

Ennek a kivédésére többféle megközelítést is alkalmazhatunk. Félretehetjük a teljes linkkapacitás 5-10 %-át az új kapcsolatoknak, amit elérhetetlennek tekintünk a megbízható útvonalválasztás számára. Tehetünk egy +10 %-os biztosítékot a becsült forgalomra, hogy semmiképpen se becsüljük alá a forgalmat. Megtehető, hogy a neurális hálózatot a valós forgalmi adatokra betanítjuk, de a teljesítménye alapján megszorozzuk egy tényezővel, hogy az esetek $< x\%$-ban becsülje alá a forgalmat. Én egy olyan megoldást választottam, amely már a hálózat tanításakor figyelembe veszi, hogy az alulbecslés nagyobb probléma a felülbecslésnél.

Létrehoztam egy saját költségfüggvényt, aminek minimalizálása a tanítás célja, a függvény az alulbecslést jobban bünteti, ezzel ösztönözve a hálózatot, hogy inkább felülbecsüljön mint alul.

$$\boldsymbol{\Delta}(\mathbf{y}, \hat{\mathbf{y}}) = \hat{\mathbf{y}} - \mathbf{y} \tag{6.1}$$

$$\boldsymbol{\Delta}_{-} = \frac{\boldsymbol{\Delta} - |\boldsymbol{\Delta}|}{2} \tag{6.2}$$

$$c(\mathbf{y}, \hat{\mathbf{y}}) = \overline{\boldsymbol{\Delta} + \boldsymbol{\Delta}_{-}^{\mathbf{2}}} \tag{6.3}$$

A költségfüggvényt az 6.3 egyenletek határozzák meg. $\mathbf{y}$ az elvárt kimenet vektora, $\hat{\mathbf{y}}$ a hálózat által adott kimenet. $\boldsymbol{\Delta}(\mathbf{y}, \hat{\mathbf{y}})$ ezen vektorok elemenkénti különbsége, vagyis a hibavektor. $\boldsymbol{\Delta}_{-}$-ban már csak a negatív hibák vannak jelen. A költségfüggvényben ($c(\mathbf{y}, \hat{\mathbf{y}})$) a negatív hibák kapnak egy extra négyzetes büntetést.

6.3.2. A hálózatok finomhangolása

Mindegyik hálózattípus esetén próbáltam megkeresni azokat a hiperparaméter értékeket, amelyekkel képes a legnagyobb pontosságot elérni, ennek érdekében mindegyik típusra több mint 30 hiperparaméter kombinációt próbáltam ki és vizsgáltam a teljesítményüket a neuronszám és rétegszám változtatásától függően. A komplexitás növelése nem mindig járt az eredmények javulásával és az időskála megválasztása is befolyásolta az egyes hálóza-

tok teljesítményét. A tanítási időben nagyon nagy különbség volt az egyes hálózattípusok között, aminek architektúrális okai vannak. A CNN-ben a számítások nagyon jól párhuzamosíthatók, viszont az LSTM komplex neuronjaiban végbemenő műveletek megkívánnak egy bizonyos sorrendet, ami ad egy alsó korlátot a futási időtartamra. A rétegszámtól és neuronszámtól függően a CNN akár 10-szer gyorsabb is lehet, mint egy hasonló paraméterszámú LSTM hálózat. A legjobb LSTM hálózat a következőképp épült fel:

- 3 LSTM réteget tartalmazott mindegyikben 64 neuronnal. Ennél kisebb komplexitású hálózatok már rosszabb eredményt értek el, míg a nagyobb komplexitásúak hasonló eredményre vezettek, de a tanításuk túl sok időt vett igénybe.
- A kimeneti réteget a kimeneti lépésszámnak megfelelő hagyományos neuron alkotta
- A neuronok aktivációs függvényeként tanh helyett ReLU-t (Rectified Linear Unit) használtam, mivel gyorsítja a tanulást és jobb eredményeket értek el vele.
- A túltanulás elkerülésére Dropout-ot (Tanulás során egy réteg neuronjainak adott részét "inaktiváljuk" [35]) és a súlyvektor hosszának korlátozását használtam

A legjobb hibrid hálózat konvolúciós, maxpooling és LSTM rétegeket tartalmazott. Mivel a bemenet dimenziószáma nem volt kifejezetten nagy ezért nem felváltva voltak konvolúciós és maxpooling rétegek, hanem két konvolúciós után következett egy maxpooling réteg.

- A hálózat a 2 konvolúciós és egy maxpooling réteg alkotta egységből kettőt tartalmazott. Minden konvolúciós réteg 100 neuront tartalmazott, melyek "látószélessége" 7 egység volt.
- Az konvolúciós rétegek után következett egy LSTM réteg, mely 100 neuront tartalmazott
- A kimeneti réteget a sima LSTM hálózathoz hasonlóan a kimeneti lépésszámnak megfelelő hagyományos neuron alkotta
- A túltanulás elkerülésére itt is ugyanazokat a módszereket alkalmaztam

6.3.3. Tanítási módszerek

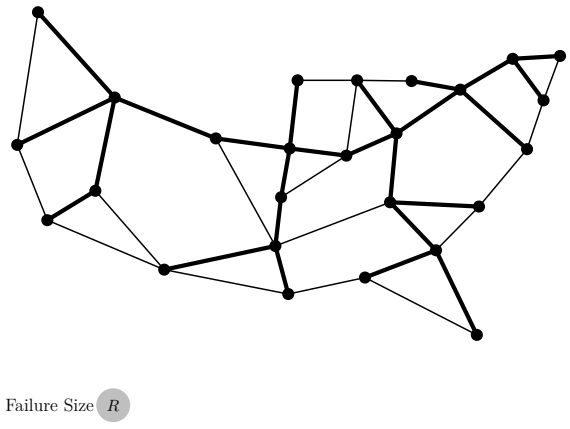
A hálózat tanításánál az adathalmaz felhasználására háromféle ötlet is az eszembe jutott: Az első szerint az adathalmazt 3 részre osztjuk, egy tanító, egy tesztelő és egy validáló részre. A tanítóhalmazon tanítjuk a hálózatot, melynek fejlődését a teszthalmazon figyeljük meg. Ha a hálózat teszthalmazon elért eredménye már nem javul megállítjuk a tanulást. A második szerint az idő múlásával folyamatosan növelnénk a tanítóhalmaz méretét. Ez úgy valósulna meg, hogy először a tanítóhalmaz az első hét adatait tartalmazná a teszthalmaz pedig a következő hét adataiból állna. Ha a teszthalmazon elért eredmény nem javul tovább akkor továbblépünk egy héttel. Ekkor a tanítóhalmazt az első két hét alkotná, míg a teszthalmazt a harmadik hét. Validációs halmaznak megtartanánk az utolsó hét adatait, amit nem használnánk fel tanításra és tesztelésre sem. A harmadik megközelítés hasonló a másodikhoz azzal a különbséggel, hogy a tanítóhalmaz méretét nem növelnénk, hanem egyszerűen léptetnénk. Ez abból a szempontból jobb, hogy nem lennének hatalmas különbségek abban, hogy melyik hét adatai hányszor szerepelnek tanítóadatként, viszont a tanítás valamennyire hektikus lesz a folyamatos tanítóhalmaz változása miatt.

7. fejezet

Szimulációs eredmények értékelése

7.1. A hálózatfejlesztés hatása

A hálózatfejlesztési algoritmusaimat két hálózaton teszteltem. Az egyik egy európai optikai hálózat modellje, melynek 16 csúcsa és 22 éle van. A másik pedig egy amerikai optikai hálózaté, melynek 26 csúcsa és 42 éle van. A teszteléshez azért nem az ESnet hálózatot használtam, mivel ott nem állt rendelkezésemre információ a hálózat linkjeinek megbízhatóságáról, ami viszont szükséges a hibamodellezéshez.



7.1. ábra. Egy amerikai optikai hálózat

A 7.1 ábrán az amerikai hálózat látható. Az eredmények összehasonlíthatóságának érdekében először hálózatfejlesztés nélkül vizsgáltam meg az útvonalválasztási algoritmusok eredményét, hogy lássam mekkora a blokkolás valószínűsége (hogy nincs minden hiba esetén megfelelő védelmi út), illetve mekkora az egyes módszerek kapacitáshasználata. Kétféle útvonalválasztási algoritmusra teszteltem, a GDP-R módszerre és egy dedikált 1+1 utas védelemre, amit én implementáltam ILP-ként.

Jól látható 7.1 táblázatban, hogy a hálózatfejlesztés nélkül bármilyen a hálózat átmérőjéhez képest kis sugarú kiesés esetén jelentős blokkolás jelentkezik. Fontos megjegyezni

R	GDP-R		1+1	
	Avg. cap.	Blocking	Avg. cap.	Blocking
2	5.990	0.125	6.143	0.125
4	6.714	0.125	6.724	0.125
6	6.901	0.242	6.901	0.242
8	7.455	0.450	7.164	0.542
10	8.309	0.542	7.861	0.700
12	9.833	0.700	8.000	0.917
14	10.619	0.825	6.667	0.975
16	-	1.000	-	1.000
18	-	1.000	-	1.000
20	-	1.000	-	1.000

7.1. táblázat. Az európai hálózat útvonalválasztási eredményei hálózatfejlesztés nélkül

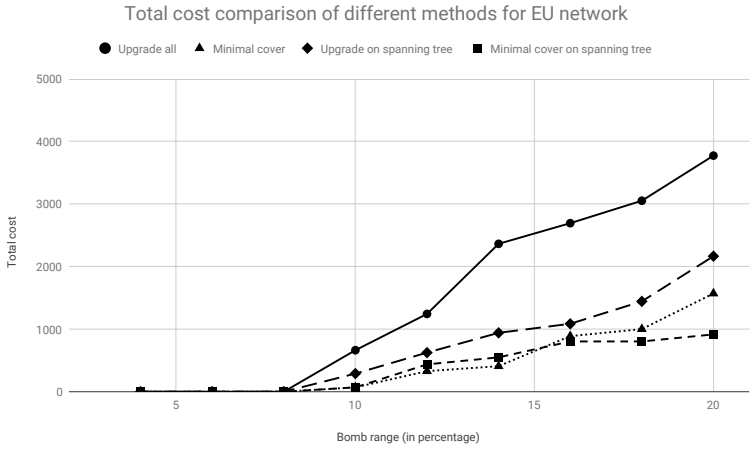
hogy a két útvonalválasztási módszer közül azonos katasztrófa hatósugár mellett az 1+1 blokkolása nagyobb.

R	GDP-R		1+1	
	Avg. cap.	Blocking	Avg. cap.	Blocking
2	4.417	0.000	4.875	0.000
4	5.767	0.000	5.942	0.000
6	6.143	0.125	6.248	0.125
8	6.286	0.125	6.390	0.125
10	6.619	0.125	6.341	0.242
12	6.714	0.242	6.423	0.350
14	6.848	0.450	6.924	0.450
16	6.848	0.450	6.924	0.450
18	7.470	0.450	6.982	0.525
20	7.470	0.450	6.982	0.525

7.2. táblázat. Az európai hálózat útvonalválasztási eredményei Spine fejlesztés után

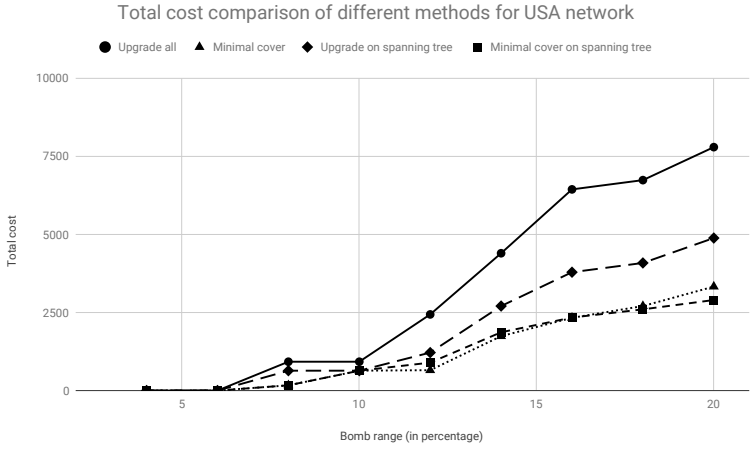
A Spine fejlesztési módszer alkalmazása már látható előnyökkel jár. A blokkolás sokkal kisebb mértékben növekszik, de szinte végig jelen van mindkét módszer esetén. Ez is mutatja mennyire fontos a hálózatot védeni kifejezetten a szétesés ellen. A kapacitáshasználat mindkét esetben kisebb, de ez a megbízhatóbb hálózatnak tudható be, hiszen egy jobb hálózat kevesebb SRLG-t eredményez, így kevesebb hibát kell védeni.

Tisztán látható, hogy a hálózat szétesése még a Spine metódus alkalmazása esetén is megjelenik, amit semmiképpen sem lehet megengedni kritikus alkalmazások esetén. Az általam javasolt 4 módszer hatósugár-költség függvényét a 7.2 és 7.3 ábrákon láthatjuk. Az európai hálózaton a várakozásoknak megfelelően a legdrágább a minden linket fejlesztő módszer lesz (●), ami akár 2-3-szoros költséggel éri el az összekötöttséget. Jól látható, hogy az egyes ötletek külön-külön is jól működnek. A feszítőfa éleinek fejlesztése (⋄) és az SRLG-ket lefedő minimális élhalmaz (△) fejlesztése önmagában jól működik, viszont a legjobb eredményt az együttes alkalmazásuk adja (■). A legkisebb fedőhalmaz (Minimal cover) egyes esetekben még jobb is, mint a legkisebb fedőhalmaz a feszítőfán (Minimal cover on spanning tree), de a katasztrófa hatótávjának növelésével már jól láthatóan fölé megy.



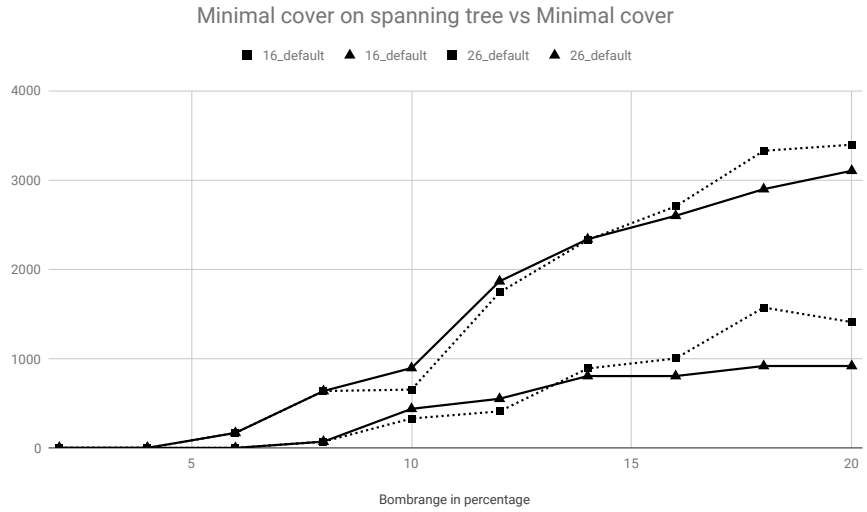
7.2. ábra. Az európai hálózat költségfüggvénye

Az amerikai hálózat esetében már nem ilyen egyértelmű a helyzet (lásd 7.3 ábra), ott a két legjobb módszer szinte végig együtt halad. Ez valószínűleg annak tudható be, hogy a feszítőfa nem a legoptimálisabban van kiválasztva. A szimulációk során a Spine által is használt feszítőfát használtam én is feszítőfaként, viszont ez nem minden esetben optimális erre a célra. Az optimális feszítőfa kiválasztása egy ILP segítségével lenne megoldható.



7.3. ábra. Az amerikai hálózat költségfüggvénye

A 7.4 ábrán a két legjobb módszert emeltem ki mindkét hálózat esetén. Összességében elmondható, hogy a legolcsóbb megoldást a feszítőfa éleiből létrehozott legkisebb fedőhal-



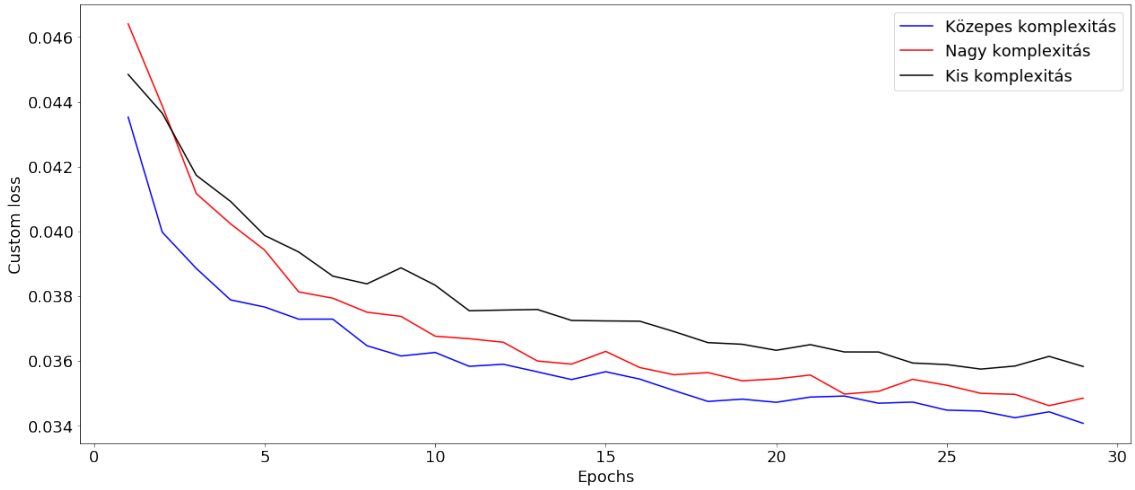
7.4. ábra. A két hálózat esetén a legjobb módszerek

7.3. táblázat. GDP-R útvonalválasztási események az USA hálózaton mind a 4 hálózatfejlesztési módszer után

R	1. módszer		2. módszer		Csak a feszítőfán fejlesztve 3. módszer		4.módszer	
	SRLG	Avg. cap.	SRLG	Avg. cap.	SRLG	Avg. cap.	SRLG	Avg. cap.
2	21	5.871	21	5.871	21	5.871	21	5.871
4	35	7.932	35	7.932	35	7.932	35	7.932
6	43	8.378	44	8.477	42	8.406	44	8.477
8	41	9.415	42	9.508	42	9.508	42	9.508
10	57	10.212	68	11.234	58	10.597	61	10.923
12	51	9.554	71	11.554	56	10.191	62	10.708
14	35	8.182	62	11.735	48	9.702	59	11.622
16	39	8.366	67	11.492	49	9.852	63	11.659
18	29	7.548	67	11.382	53	9.434	85	11.634
20	30	7.538	73	11.508	48	8.662	109	12.588

maz adja. Jól megfigyelhető a nagyobb hálózat magasabb költségigénye az összefüggőség megtartásához.

A 7.3 táblázatban az amerikai hálózat SRLG szám, átlagos kapacitás értékeit láthatjuk a hatósugár függvényében a 4 különböző módon fejlesztett hálózatra. A módszerek a 5.1 fejezetben bemutatott módon vannak jelölve. A 7.3 ábrával összevetve látható, hogy a magasabb költség alacsonyabb SRLG számmal és alacsonyabb átlagos felhasznált kapacitással jár. Ez nem meglepő hiszen egy jobban felfejlesztett hálózatban kisebb a hiba valószínűsége, így az SRLG listák is rövidebbek.



7.5. ábra. Egy kis, közepes és nagy komplexitású LSTM hálózat tanítása

7.2. A forgalom előrejelzés hatása

Ha a hálózat minden általunk vizsgált hiba esetén összefüggő marad, akkor a GDP-R megbízható útvonalválasztó algoritmus képes létrehozni egy kapcsolatot a forrás és a nyelő között. Amennyiben a hálózat linkjein elérhető szabad kapacitásokat is meg tudjuk adni, az útvonalválasztás biztosan megbízható lesz. A következőkben több neurális hálózat eredményeit mutatom be a forgalom előrejelzésére, melyeket a saját költségfüggvényem segítségével tanítottam be az ESnet legforgalmasabb linkjének forgalmi adatain.

Elsőként a hálózat által igényelt komplexitást határoztam meg, hogy ellenőrizzem, milyen hiperparaméterek esetén képes valóban megragadni az adathalmaz lényegét és jó becslést adni. Egy ilyen összehasonlítás a 7.5 ábrán látható, ahol a hibafüggvény változása látható a tanítás során. Az alacsony komplexitású hálózat érezhetően nem képes komolyabb összefüggések megtanulására, ellenben a közepes és nagy komplexitású hálózatokkal. Az alacsony komplexitású LSTM hálózat csak 2 réteget tartalmazott rétegenként 32 neuronnal, a közepes 3 réteget rétegenként 64 neuronnal, míg a nagy komplexitású 4 réteget rétegenként 128 neuronnal.

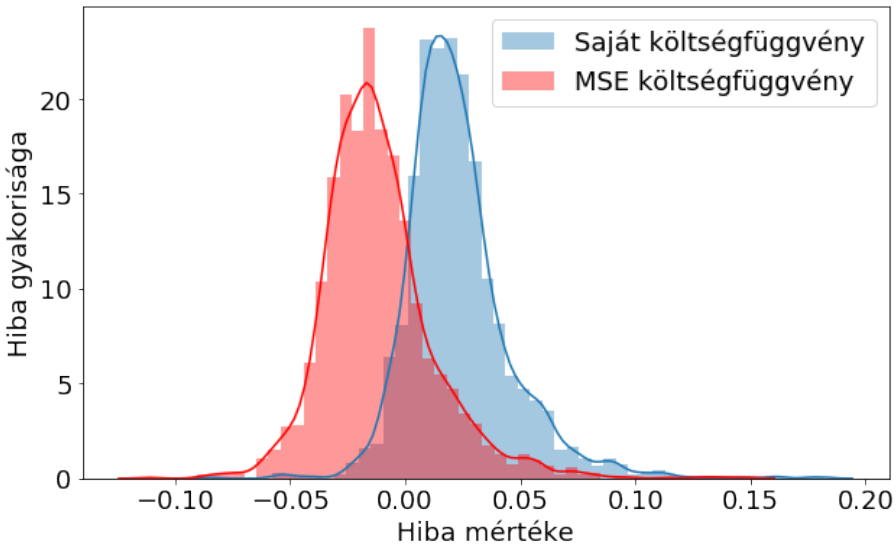
Ezután a különböző tanítási módszereket hasonlítottam össze, amiket a 6.3.3 fejezetben mutattam be. Ezek a tanítóhalmaz szerint normál (1 nagy tanítóhalmaz), növekvő (a tanítóhalmaz folyamatosan nő) és lépegető (a tanítóhalmaz az idővel folyamatosan változik). Mivel a tanítóhalmaz 2 esetben is folyamatosan változik, ezért nincs sok értelme összehasonlítani az azon kapott eredményeket, ezért egységesen a validációs halmazon elért eredményeket vetettem össze, amely egyik esetben sem volt tanításra használva.

	Normál	Növekvő	Lépegető
Eredmény a validációs halmazon (custom loss)	0.0326	0.0351	0.0318

7.4. táblázat. A három tanítási módszer összehasonlítása

Az eredmény igen érdekes lett. Azt mutatja, hogy a legjobb eredményt akkor kapjuk, ha folyamatos továbbtanítással fejlesztjük a hálózatot, úgy, hogy a régebbi adatokkal már nem tanítjuk újra. Ez azt is jelentheti, hogy az adathalmaz folyamatos nagymértékű változása miatt hátrányos folyamatosan a régi adatokkal is tanítani.

Az eddigiekben már a saját költségfüggnyényemet alkalmaztam a hálózatok összehasonlítására, viszont fontos, hogy a költségfüggvény teljesítményét is összehasonlítsuk, hogy



7.6. ábra. A becslés hibájának eloszlása a két költségfüggvény esetén

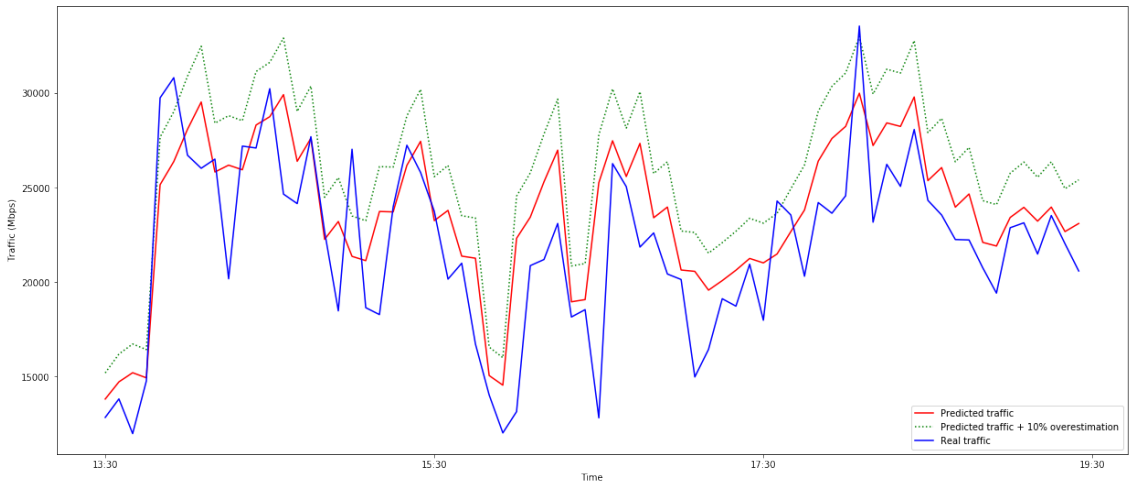
látszódjon működik-e a célzott büntetés. Elsőként megvizsgáltam, hogy a büntetés miatt tényleg inkább felülbecsüli-e a forgalmat a hálózat. A leggyakoribb költségfüggvények szimmetrikusak, mint például az átlagos abszolút hiba (Mean Absolute Error - MAE) vagy az átlagos négyzetes hiba (Mean Squared Error - MSE). Én az utóbbival hasonlítottam össze a költségfüggvényemet, hogy megvizsgáljam a hiba természetében történő változást.

Ahogy az a 7.6 ábrán látható a hiba hisztogrammja ténylegesen a kívánt irányba változott. Míg az MSE költségfüggvény esetén a hiba inkább negatív, addig az én költségfüggvényem már inkább pozitív irányban hibázik. Ha teljesen biztosra akarunk menni az alulbecslés megszüntetésére akkor a már említett módszereket használhatjuk fel.

A különböző neurális háló típusok nagyon nagy mértékben különböznek egymástól komplexitás, hiperparaméterszám és tanítási idő szempontjából is, így teljesen objektív összehasonlítást nagyon nehéz adni. Mivel az LSTM neuronjai nagyon komplexek, a konvolúciós réteg neuronjai pedig nagyon egyszerűek, így egy sokkal nagyobb paraméterszámú CNN-LSTM hálózat is sokkal gyorsabban tanítható, mint egy tisztán LSTM rétegekből álló hálózat. A legjobb tisztán LSTM hálózat szinte nagyon hasonló eredményt (körülbelül 1 %-al rosszabbat) produkált mint a legjobb CNN-LSTM hibrid hálózat, viszont a tanítási időben óriási volt a különbség. Hiperparaméterektől függően akár 10-szeres sebességbeli különbség is volt a két hálózat között, ezért az LSTM hálózatot elvetettem. A továbbiakban tehát a CNN-LSTM hibrid hálózatot használtam forgalmi predikcióra, amely a legjobb pontosság mellett a leggyorsabban tanítható is.

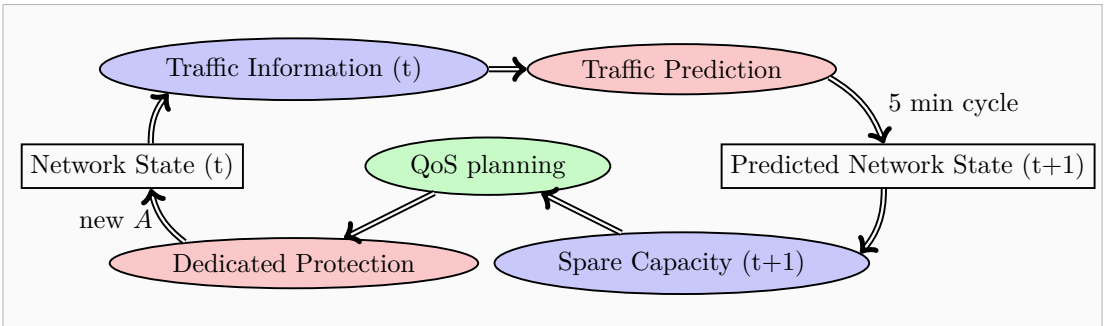
A 7.7. ábrán a legjobb neurális hálózat előrejelzése illetve egy +10%-os biztonsági sávval való kiegészítése látható a Sacramento és Chicago linkeken (2019.08.18 - 13:30 és 19:60 között). Az előrejelzés 2 lépéses és mivel 5 perces lépésközzel állítottam be a becslést, így a következő 10 percre kapunk előrejelzést.

Maga a forgalmi előrejelzés felhasználása a 6.1. ábra segítségével történik. Ha a hálózat minden linkjén megbecsültük a szabad kapacitást a kapcsolat idejére, akkor meghatározhatunk egy átlagos szabad kapacitást, amelyet a kapcsolat felhasználhat. A kapcsolat kapacitásigényéből kiindulva meghatározhatjuk, hogy hányszoros a legnagyobb kapacitásnövekedés, amit még ráterhelhetünk a hálózatra. Ehhez a kapacitásnövekedéshez tartozik egy becsült rendelkezésre állási érték, amit biztosítani tudunk a kapcsolat számára. Például, ha a kapcsolat a védtelen útvonal kapacitásának háromszorosát is felhasználhatja,



7.7. ábra. Forgalmi előrejelzés a CNN-LSTM hibrid hálózattal (SACR-CHIC link (2019.08.18. 13:30 - 19:30))

akkor 0.999-es rendelkezésre állás helyett, 0.9999-es rendelkezésre állást tudunk biztosítani. Ha még több szabad kapacitás érhető el a hálózatban, akkor még jobb rendelkezésre állás érhető el. Ötször akkora kapacitáshasználattal akár 0.99999-es rednelkezésre állást is biztosíthatunk a kapcsolatnak.



7.8. ábra. A QoS fejlesztése forgalom előrejelzés segítségével

Természetesen annak érdekében, hogy ténylegesen csak a rendelkezésre álló szabad kapacitást használjuk fel és ne okozzunk blokkolást túlzott erőforrásfoglalással érdemes valamekkora biztonsági sávot használni a szabad kapacitás tekintetében. Emellett érdemes figyelembe venni a összeköttetés (kapcsolat) QoS igényét is az erőforrásfoglalás során. A további munkám során célom, hogy a különböző QoS osztályoknak megfelelő QoS plannert (tervezőt) tervezzek és annak teljesítményét vizsgáljam annak érdekében, hogy az összeköttetés igényeknek (és azok QoS osztályának) megfelelő optimális biztonsági sávot megtaláljam. A 7.8. ábrán látható a hálózati predikció felhasználása a QoS növelése érdekében. A forgalmi információkat felhasználva megbecsüljük a hálózat állapotát a következő 5 percre. Az így meghatározott szabad kapacitást felhasználhatjuk az elérhető QoS osztályok meghatározására. A kapcsolatokat azután a kívánt QoS szintnek megfelelő dedikált védelemmel látjuk el. A folyamat legfontosabb komponenense maga a forgalom előrejelzés (Traffic Predicition). A dolgozatom ezen fejezetben megmutattam, hogy egy kifejezetten nehezen predikálható forgalmú hálózatra (lásd 6.2.3. fejezet) is képesek vagyunk megfelelő neurális hálózatos predikciót adni, így ezáltal a megfelelő hálózati forgalomi becslést felhasználhatjuk a rendelkezésre állás növelése érdekében.

8. fejezet

Összefoglalás

Modern társadalmunk egyre jobban hagyatkozik az Internetre, így egyre több kritikus alkalmazás jelenik meg.

A dolgozatomban olyan magas rendelkezésre állású hálózat kifejlesztésén munkálkodtam amely lehetővé teszi a kritikus alkalmazások elterjedését, mivel a magas (az adott QoS osztálynak megfelelő) rendelkezésre állás biztositása mellett még a regionális hibáknak is ellenáll.

Munkám során a FRADIR (FRAmework for DIsaster Resiliance) rendszert fejlesztettem tovább, amely egyszerre képes kihasználni a hibamodellezés, a hálózattervezés és a megbízható útvonalválasztás előnyeit mégpedig költség-hatékony módon. Regionális hibák esetén komoly probléma a hálózat szétszakadása (több komponensre esése), ami a kapcsolatok megszakadásához vezet. Az új QoS kritikus alkalmazások, mint például a tőzsde és távgyógyászat ez megengedhetetlen. Ennek megoldására a dolgozatomban több hálózatfejlesztési módszert is javasoltam, melyek költséghatékony módon képesek megerősíteni a hálózatot, úgy hogy a regionális kiesések se szakítsák szét. Költséghatékonyság szempontjából a módszer egy nagy megbízhatóságú feszítőfán történő fejlesztést kombinálja az SRLG-ket minimálisan lefedő élhalmaz megkeresésével.

A hálózatokban alkalmazható megbízható útvonalválasztási algoritmusok alkalmazhatóságát nagyban befolyásolja a linkeken elérhető szabad kapacitás. Ennek az előrejelzése lehetővé tenné, hogy a kritikus alkalmazásoknak mindig a várható forgalomnak legmegfelelőbb megbízható útvonalat javasoljuk. A különböző szabad kapacitási szintekhez hozzárendelhető az elérhető legjobb rendelkezésre állású összeköttetés (dedikált védelmi megoldás), ennek köszönhetően javítható a kapcsolat rendelkezésre állása. A forgalom előrejelzésére egy hibrid neurális hálózatot tanítottam be, amely kombinálja a konvolúciós és LSTM hálózatok előnyeit. Mivel a forgalom alulbecslése olyan kapacitások felhasználását tenné lehetővé a megbízható útvonalválasztás számára, amelyek nem lesznek elérhetőek a jövőben, ezért létrehoztam egy saját költségfüggvényt (loss function-t) a neurális hálózat betanítására, amely egy plusz büntetést ad alulbecslés esetén.

A további munkám során a halózati forgalom becslését végző neurális hálózatot fogom felhasználni egy QoS planner (QoS tervező) létrehozására, amely a becsült szabad kapacitást képes dinamikusan felhasználni dedikált védelmi utak formájában a kritikus alkalmazások regionális hibák elleni védelmére.

Irodalomjegyzék

[1] 2016. URL `http://my.es.net`. [Online; accessed 28-October-2019].

[2] Abdulaziz Alashaikh – Teresa Gomes – David Tipper: The spine concept for improving network availability. *Computer Networks*, 82. évf. (2015. 05), 4–19. p.

[3] Réka Albert – Albert-László Barabási: Statistical mechanics of complex networks. *Reviews of modern physics*, 74. évf. (2002) 1. sz., 47. p.

[4] P. Babarczi – J. Tapolcai – P. Ho – M. Médard: Optimal dedicated protection approach to shared risk link group failures using network coding. In *2012 IEEE International Conference on Communications (ICC)* (konferenciaanyag). 2012. June, 3051–3055. p. ISSN 1938-1883.

[5] Péter Babarczi – Alija Pašić – János Tapolcai – Felicián Németh – Bence Ladóczki: Instantaneous recovery of unicast connections in transport networks: Routing versus coding. *Computer Networks*, 82. évf. (2015), 68–80. p.

[6] Eric Bouillet – Georgios Ellinas – J.-F Labourdette – Ramu Ramamurthy: *Path Routing in Mesh Optical Networks*. 2007. 10, 265. p. ISBN 9780470015650.

[7] Eduardo Canale – Pablo Romero – Gerardo Rubino: Irrelevant components and exact computation of the diameter constrained reliability. 2014. 09.

[8] Samira Chabaa – Abdelouhab Zeroual – Jilali Antari: Identification and prediction of internet traffic using artificial neural networks. *Journal of Intelligent Learning Systems and Applications*, 2. évf. (2010) 03. sz., 147. p.

[9] VNI Cisco: Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 2018.

[10] Paulo Cortez – Miguel Rio – Miguel Rocha – Pedro Sousa: Multi-scale internet traffic forecasting using neural networks and time series methods. *Expert Systems*, 29. évf. (2012) 2. sz., 143–155. p.

[11] Robert Fildes – Michèle Hibon – Spyros Makridakis – Nigel Meade: Generalising about univariate forecasting methods: further empirical evidence. *International Journal of Forecasting*, 14. évf. (1998) 3. sz., 339–358. p.

[12] Teresa Gomes – János Tapolcai – Christian Esposito – David Hutchison – Fernando Kuipers – Jacek Rak – Amaro De Sousa – Athanasios Iossifides – Rui Travanca – Joao André és mások: A survey of strategies for communication networks to protect against large-scale natural disasters. In *2016 8th international workshop on resilient networks design and modeling (RNDM)* (konferenciaanyag). 2016, IEEE, 11–22. p.

[13] John Heidemann – Lin Quan – Yuri Pradkin: *A preliminary analysis of network outages during hurricane sandy.* 2012, University of Southern California, Information Sciences Institute.

[14] Sepp Hochreiter – Jürgen Schmidhuber: Long short-term memory. *Neural computation*, 9. évf. (1997) 8. sz., 1735–1780. p.

[15] Jun Jiang – Symeon Papavassiliou: Detecting network attacks in the internet via statistical network traffic normality prediction. *Journal of Network and Systems Management*, 12. évf. (2004) 1. sz., 51–72. p.

[16] WU Jianjun – GAO Ziyou – SUN Huijun: Statistical properties of individual choice behaviors on urban traffic networks. *Journal of Transportation Systems Engineering and Information Technology*, 8. évf. (2008) 2. sz., 69–74. p.

[17] Ahmed E. Kamal – Mirzad Mohandespour: Network coding-based protection. *Opt. Switch. Netw.*, 11. évf. (2014. január), 189–201. p. ISSN 1573-4277.
URL `http://dx.doi.org/10.1016/j.osn.2013.06.006`. 13 p.

[18] Andrej Karpathy: Convolutional neural networks, 2013.
URL `http://cs231n.github.io/convolutional-networks/`. [Online; accessed 23-September-2019].

[19] Alireza Khotanzad – Nayyara Sadek: Multi-scale high-speed network traffic prediction using combination of neural networks. In *Proceedings of the International Joint Conference on Neural Networks, 2003.* (konferenciaanyag), 2. köt. 2003, IEEE, 1071–1075. p.

[20] Tae-Young Kim – Sung-Bae Cho: Web traffic anomaly detection using c-lstm neural networks. *Expert Systems with Applications*, 106. évf. (2018), 66–76. p.

[21] Will E Leland – Murad S Taqqu – Walter Willinger – Daniel V Wilson: On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on networking*, 2. évf. (1994) 1. sz., 1–15. p.

[22] Sebastian Neumayer – Gil Zussman – Reuven Cohen – Eytan Modiano: Assessing the vulnerability of the fiber infrastructure to disasters. *IEEE/ACM Transactions on Networking*, 19. évf. (2011) 6. sz., 1610–1623. p.

[23] Christopher Olah: Understanding lstm networks, 2015.
URL `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`. [Online; accessed 17-September-2019].

[24] Konstantina Papagiannaki – Nina Taft – Z-L Zhang – Christophe Diot: Long-term forecasting of internet backbone traffic: Observations and initial models. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)* (konferenciaanyag), 2. köt. 2003, IEEE, 1178–1188. p.

[25] Alija Pasic – Rita Girão-Silva – Balazs Vass – Teresa Gomes – Péter Babarczi: Fradir: A novel framework for disaster resilience. *2018 10th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2018., 1–7. p.

[26] Alija Pašic – Rita Girao-Silva – Balázs Vass – Teresa Gomes – Ferenc Mogyorósi – Péter Babarczi – János Tapolcai: Fradir-ii: An improved framework for disaster resilience. *2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2019., 1–7. p.

[27] Alija Pašić: Instantaneous failure recovery and localization in transport networks. *Villamosmérnöki Tudományok Doktori Iskola, BME*, 2018.

[28] Alija Pašić – Péter Babarczi: Delay aware survivable routing with network coding in software defined networks. In *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)* (konferenciaanyag). 2015, IEEE, 41–47. p.

[29] Alija Pašić – János Tapolcai – Péter Babarczi – Erika R Bérczi-Kovács – Zoltán Király – Lajos Rónyai: Survivable routing meets diversity coding. In *2015 IFIP Networking Conference (IFIP Networking)* (konferenciaanyag). 2015, IEEE, 1–9. p.

[30] Burjiz Pithawala – Faiyaz Shahpurwala – Jacob Hartinger – Narayanan Thyagarajan: Network availability monitor, 2004. 8. US Patent 6,747,957.

[31] Lin Quan – John Heidemann – Yuri Pradkin: Trinocular: Understanding internet reliability through adaptive probing. 43. köt. 2013. 08, 255–266. p.

[32] Jacek Rak – David Hutchison – Eusebi Calle – Teresa Gomes – Matthias Gunkel – Paul Smith – Janos Tapolcai – Sofie Verbrugge – Lena Wosinska: Recodis: Resilient communication services protecting end-user applications from disaster-based failures. In *2016 18th International Conference on Transparent Optical Networks (ICTON)* (konferenciaanyag). 2016, IEEE, 1–4. p.

[33] Joshua S Richman – Douglas E Lake – J Randall Moorman: Sample entropy. In *Methods in enzymology*. 384. köt. 2004, Elsevier, 172–184. p.

[34] Antti Sorjamaa – Jin Hao – Nima Reyhani – Yongnan Ji – Amaury Lendasse: Methodology for long-term prediction of time series. *Neurocomputing*, 70. évf. (2007) 16-18. sz., 2861–2869. p.

[35] Nitish Srivastava – Geoffrey Hinton – Alex Krizhevsky – Ilya Sutskever – Ruslan Salakhutdinov: Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15. évf. (2014) 1. sz., 1929–1958. p.

[36] Maciej Szmit – Anna Szmit – Sławomir Adamus – Sebastian Bugała: Usage of holt-winters model and multilayer perceptron in network traffic modelling and anomaly detection. *Informatica*, 36. évf. (2012) 4. sz.

[37] János Tapolcai – Balazs Vass – Zalán Heszberger – József Bíró – David Hay – Fernando A. Kuipers – Lajos Rónyai: A tractable stochastic model of correlated link failures caused by disasters. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018., 2105–2113. p.

[38] David Tipper: Resilient network design: challenges and future directions. *Telecommunication Systems*, 56. évf. (2014), 5–16. p.

[39] M. To – P. Neusy: Unavailability analysis of long-haul networks. *IEEE Journal on Selected Areas in Communications*, 12. évf. (1994. Jan) 1. sz., 100–109. p. ISSN 0733-8716.

[40] Jean-Philippe Vasseur – Mario Pickavet – Piet Demeester: *Network recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. 2004, Elsevier.

[41] Liu Xiao-Tong: Study on data normalization in bp neural network [j]. *Mechanical Engineering & Automation*, 3. évf. (2010), 122–123. p.