



M Ű E G Y E T E M 1 7 8 2

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

Anda Géza Pál

10 GBIT/S-OS OPTIKAI GERINC HÁLÓZAT KONSZOLIDÁCIÓJA

TDK dolgozat

KONZULENS

Dr. Cinkler Tibor,
Dr. Szigeti János

BUDAPEST, 2014

Tartalomjegyzék

1. Abstract - Bevezető	3
2 Elméleti áttekintő	4
2.1 Többszörös hozzáférésű közegek	4
2.2 Optikai átviteli közeg.....	4
2.3 DWDM	5
2.4 Hálózatok felépítése.....	7
2.5 Fogalmak	10
2.5 Teszt hálózat	11
3 Kezdeti állapot.....	12
3.1 Hálózat szerkezete	12
3.2 Előzetes eredmények	14
3.3 Konkrét feladat	14
4 Kidolgozott algoritmusok és eredményeik	16
4.1 Hullámhosszok szerinti átrendezés	16
4.1.1 Alulról töltő mohó módszer.....	16
4.1.2 Felülről töltő mohó módszer.....	18
4.1.3 Heurisztikus módszer.....	18
4.2 Eredmények	20
4.3 Újabb futtatások más tartományokon	22
4.3 Útvonalmódosítással történő átrendezés.....	24
4.5 Dokumentálás	25
5 Összefoglalás, jövőbeni lehetőségek	26
5.1 Specifikáció teljesülése.....	26
5.2 Tapasztalatok összefoglalása	26
5.3 Jövőbeni lehetőségek	27
Irodalomjegyzék.....	29

1. Abstract - Bevezető

A társadalom információfogyasztása napról napra növekszik. Ahogy a technológia próbálja tartani a tempót, sokszor nem az infrastruktúra bővítése a legösszegebb megoldás. Optikai hálózatok régóta léteznek, optikai átvitelrel kommunikálnak egymással ügyfelek, szolgáltatók, országok, földrészek. Az, hogy új kábel lefektetése nélkül ériünk el kapacitásnövekedést, szintén nem idegen dolog. A végponti eszközök fejlesztése egyre inkább lehetővé teszi, hogy egyre nagyobb részletességgel, mélységgel használjuk ki a fizikai lehetőségeket. Munkám során egy ilyen jellegű hálózatbővítést követő, további bővítést elősegítő konszolidálást módszereit kutattam. A konszolidálás lényege, hogy az eddigi igénybevételt kisebb és összefüggő kapacitáson csoportosítsuk, helyet teremtve a jövő igényeinek, miközben folyamatosan kiszolgálja a hálózat az aktuális igényeket.

Jelenleg az optikai átviteli közegben előre definiált hullámhosszokon történik az átvitel. A bővítéssel új hullámhossz-tartományok érhetőek el. A rádiós közegtől eltérően itt nem új tartomány kerül bevezetésre, hanem a már meglévő tartományban történik a hullámhossz sávok kisebbre osztása. A cél a jelenlegi dedikált átviteli útvonalak áthelyezésével elérni, hogy kevesebb hullámhosszt használjunk fel. Az eredményül felszabadított egybefüggő hullámhossztartomány felhasználó jövőbeni igények ésszerű elvezetésére.

Külön problémát jelent a hálózat összetettsége. Bizonyos eszközök képesek a teljes spektrum értelmezésére, de bizonyosak nem. A módosítások során figyelni kell az összeköttetés képességeit.

Elsődlegesen a cél egy megfelelő szint elérése volt, tehát hány hullámhosszon tudjuk elvezetni a jelenlegi igényeket. A tervezés során arra is figyelni kell, hogy a meghatározott módosítások lépésről lépésre legyenek dokumentálva, illetve azok minél kevesebb fizikai beavatkozást igényeljenek.

Dr. Szigeti János munkája során ugyanezen hálózaton az elvi optimális megoldást kereste matematikai szemszögből több szempont alapján [2]. Eredményei alapján belátható, hogy létezik optimális megoldás, munkám során az e felé vezető gyakorlatban is megvalósítható utat kerestem két megközelítésből.

2 Elméleti áttekintő

2.1 Többszörös hozzáférésű közegek

Többszörös hozzáférésű (Divided Multiple Access) közegekben az átvitel felosztása jelen ismereteink szerint három módon történhet:

1. TDMA: Időalapú osztás, azaz az átviteli közeg időablakokra van osztva, amin a felhasználók osztoznak. Egy időablak egy felhasználóhoz tartozik.
2. CDMA: Kódosztás, azaz az átvitelre szánt információt speciális módon kódolják. Így a felhasználók azonos fizikai paraméterekkel osztozhatnak ugyanazon a közegen. Az ilyen hálózatokban lévő végeseleközök a több, kevert jelből dekódolni tudják az átvitt információt.
3. WDMA: Frekvenciaosztás, azaz az átviteli közeget diszkrét frekvenciatartományokra van osztva. Egy frekvenciatartományhoz egy felhasználó tartozik.

A továbbiakban mivel nem a felhasználó, hanem a hálózat oldaláról közelítem meg a témát, így többszörös hozzáférés helyett annak megvalósítási (multiplexálási) stratégiájáról fogok írni. (Divided Multiple Access -> Division Multiplexing)

2.2 Optikai átviteli közeg

Az optikai közegen történő adattovábbítás már a XX. század derekán felmerült, de a csillapítás használható mértékűre csökkentése csak az elmúlt néhány évtizedben történt meg. Manapság gyakorlatilag az adatátvitelben a vezető technológia. Ennek okai [3]:

- Hatalmas adatátviteli sebesség (jelenlegi sebességrekord 273 Gbps/hullámhosszcsatorna [12])
- Nagy sáv szélesség (közel 60 THz, viszonyítási alapként a koaxiális kábel 12.4 GHz [13])

- Alacsony csillapítás (0.2dB/km, viszonyításként 13 km hosszú optikai kábel csillapítása megegyezik 1 m koaxiális kábel csillapításával [13])
- Magas ár/kapacitás arány

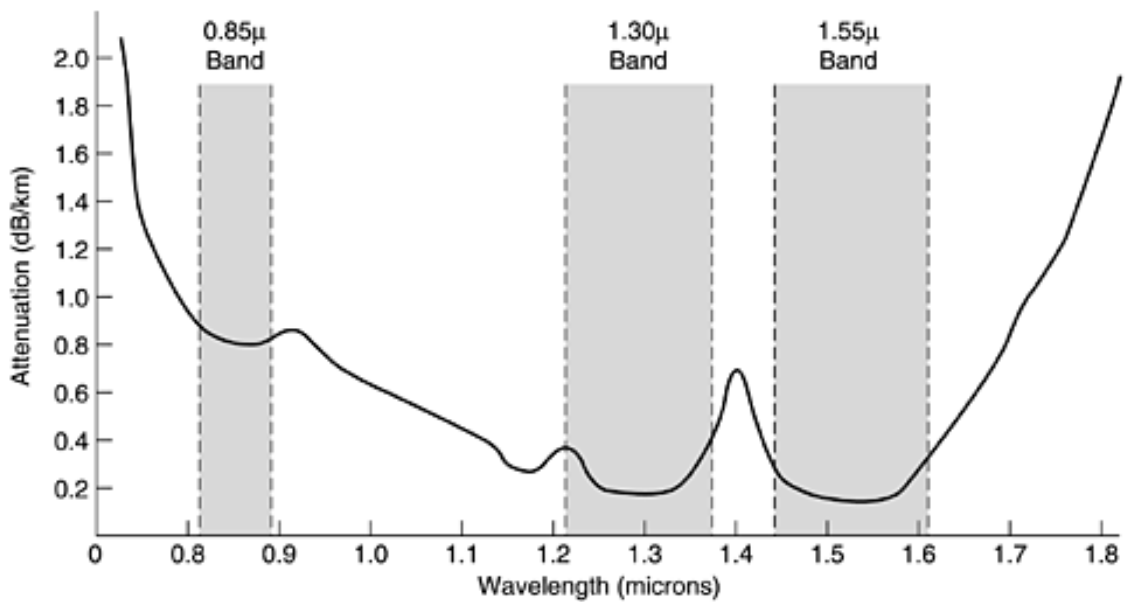
Optikai átviteli technológiákkal a ITU-T külön csoportja foglalkozik [6].

Fizikailag az átvitelre legalkalmasabb az egymódosú szál. Lényegesen kisebb magátmérője van, mint a multimódosú kábelnek, ellenben sokkal jobban tűri a csillapítást. Ezek a szálak a közönséges műanyag borítás helyett Nátrium-szilikát (vízüveg) borítással rendelkeznek. A fény által használt belső rész vastagsága 9 μm , szemben a multimódosú kábellel, amiben 50 μm .

Optikai közegen az időalapú multiplexálás nehezen megvalósítható. Ennek feltétele, hogy az összeköttetés végén lévő két feldolgozó eszköz azonos órajellel működjön és tökéletesen legyen szinkronizálva. Belegondolni is nehéz, hiszen egy hullámhossz alig több, mint 5 fs ($5,178 \cdot 10^{-15}$ másodperc), ami nagyságrendileg egy szinten van a céziumos atomóra pontosságával (10^{-16}). Arról nem is szólva, hogy ha ötvözzük hullámhosszosztással, akkor a különböző hullámhosszokon különböző időben érnek be a jelek, ami szintén problémát okoz. A kódosztásos multiplexálás szintén nehezen megvalósítható, hiszen a kódolás feldolgozásához használatos mai processzorok lényegesen lassabban működnek, mint egy ilyen átviteli közegben a jel chip-rátája [3]. Ezek indokolják azt, hogy ma az optikai átviteli technológiák alapvetően hullámhosszosztásra építenek.

2.3 DWDM

A DWDM (dense wavelength-division multiplexing – sűrű hullámhossz-osztásos multiplexálás) technológia alapjaként az ITU telekommunikációs szabványtestület mérések alapján három hullámhossztartományt határoz meg ajánlásként használatra (1. ábra: Csillapítás optikai átviteli közegben hullámhossz függvényében).



1. ábra: Csillapítás optikai átviteli közegben hullámhossz függvényében

Legjobb átviteli jellemzőkkel az 1260-1675nm-es tartomány rendelkezik. Ezt további hat különböző tartományra osztották fel [5], melyen belül a C sáv 1530–1565nm között található (ebben a sávban gyakorlatilag 0 a szálon belüli diszperzió [8]). A C (common) sávon belül a 193.1 THz-hez kötve frekvenciatáblázatot adott ki [7] (1552.5244nm középponti hullámhossz). A táblázatban nominális középponti hullámhosszok vannak megadva, illetve a hozzájuk tartozó nominális átviteli frekvenciák. E mellé 4 sáv szélességet jelölnek meg: 12.5, 25, 5, és 100 GHz (vagy a fölötti). Erre példa a 2. ábrán alább látható. Az, hogy milyen sáv szélességet használ egy-egy szolgáltató, határozza meg, hogy az ajánlott tartományt hány részre osztják fel.

Nominal central frequencies (THz) for spacings of:				Approximate nominal central wavelengths (nm) (Note)
12.5 GHz	25 GHz	50 GHz	100 GHz and above	
193.1000	193.100	193.10	193.1	1552.5244
193.0875	–	–	–	1552.6249
193.0750	193.075	–	–	1552.7254
193.0625	–	–	–	1552.8259
193.0500	193.050	193.05	–	1552.9265
193.0375	–	–	–	1553.0270
193.0250	193.025	–	–	1553.1276
193.0125	–	–	–	1553.2282
193.0000	193.000	193.00	193.0	1553.3288
•	•	•	•	•
•	•	•	•	•

2.

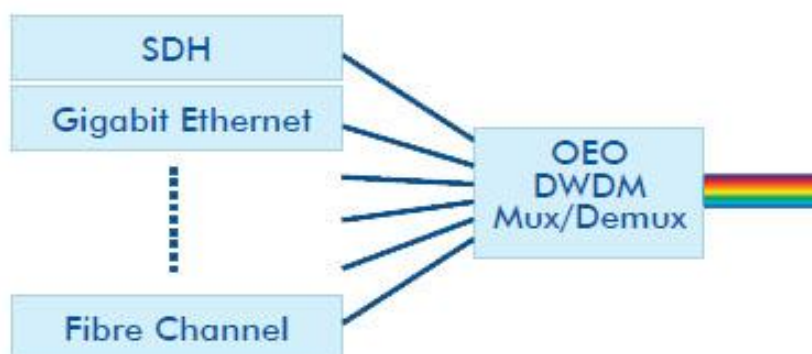
ábra: ITU-T G694.1 ajánlás nominális hullámhosszokra és frekvenciákra -részlet -

Ezen ajánlás alapján a C sávot 100 GHz sávszélességgel fel lehet osztani 72 különböző hullámhosszcsatornára. Ez a felosztás „atomi” része, egy ilyen hullámhosszcsatornához egy átvitel tartozik.

2006. szeptemberében a Japán Nippon Telegraph and Telephone vállalata olyan berendezéseket tesztelt, mellyel 14 Tbps átviteli sebességet értek el egy egymódusú optikai szálon. A kutatás idevágó tanulsága, hogy nem feltétlenül szükséges új kábeleket lefektetni a hálózati kapacitás bővítéséhez, elegendő a végponti eszközöket fejleszteni. A legtöbb rekordot ugyanazon a típusú optikai szálon érik el, pusztán a végberendezéseket, vagy azok szoftverét fejlesztik. Az NNT az eredményét például úgy érte el, hogy a addig nem, vagy nem erre a célra használt hullámhosszcsávokat is felhasznált az átvitel során (C-common mellé L – long és S – short sávokat), illetve új jelkódolást alkalmaztak (CSRZ-DQPSK - Carrier Suppressed Return to Zero Differential Quadrature Phase Shift Keying [12])

2.4 Hálózatok felépítése

Optikai hálózatokat jellemzően olyan helyeken használnak ahol két egymástól távol álló hálózatot kell összekötni vagy két hálózat között nagy forgalmat kell lebonyolítani. Mindkét esetben szükséges, hogy a lokális, elektronikus átvitelen alapuló hálózatból optikai hálózatba lépéskor átalakítás történjen. Ezt az átalakítást úgynevezett E/O illetve O/E (electrical to optical illetve optical to electrical) jelátalakítók végzik. Az előbbi a bejövő elektromos jelet optikai jellé alakítja, míg az utóbbi az optikai jelet alakítja elektromos jellé (3. ábra). A gyakorlatban a folyamatot végeztetésnek, a folyamatot végző eszközt pedig transzponder hívják. A transzponder komplexitását az adja meg, hogy hány különböző hullámhossz érzékelésére és kezelésére alkalmas.



3. ábra: Elektronikus és optikai hálózat kapcsolata (transzponder) [5]

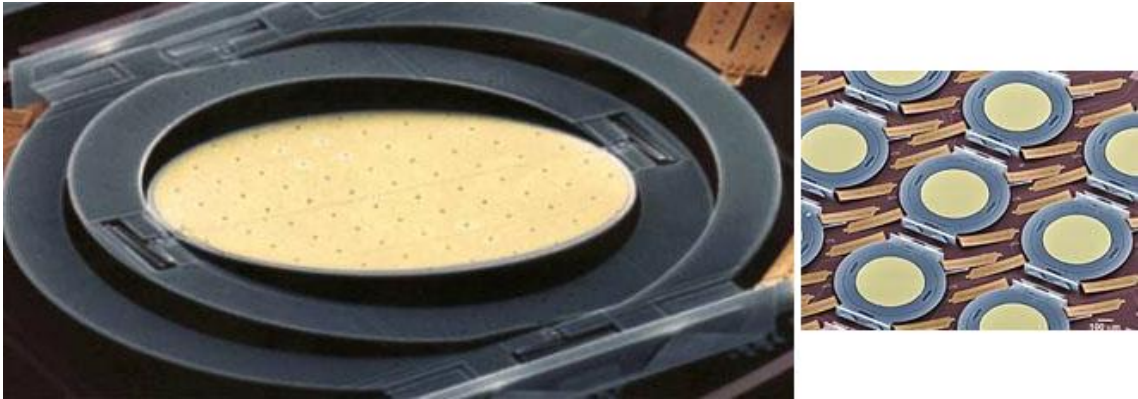
Az átalakítás mellett szükség van az adatok hullámhosszcsatornákra történő rendezésére. A már fényként közlekedő (E/O által) adatok egy optikai multiplexer által kerülnek egy fényszálra. A szál másik végén a folyamatot ellentétes irányban egy optikai demultiplexer végzi, szétválasztja a hullámhosszokat. A külön álló jeleket adott hullámhosszokra szintezett O/E transzponder alakítja vissza elektromos jellé, ahogy az a 4. ábraán is látható.



4. ábra: Optikai multiplexer és demultiplexer [5]

Ha a bekezdés elején megfogalmazott gondolatot megfordítjuk és az elvet nem pont-pont összeköttetésekre, hanem gerinchálózatokra alkalmazzuk, akkor újabb feladatok merülnek fel. Ha egy hálózati csomópont több optikai kimenettel is rendelkezik, akkor szükséges lehet a fénynyalábok továbbítása egyik ágból a másikba (kapcsolás - transzfer). Előfordulnak olyan esetek, hogy a bejövő és a kimenő ágon eltérnek a hullámhosszok (hullámhossz-konverzió). Komplexebb csomópont esetén (2-nél több ágat kezel) szükséges lehet az irányok közül választani és ezek között kapcsolni (útvonalválasztás).

A kapcsolás és útválasztás feladatát szintén optikai eszközök látják el, hiszen az elektronikus elven működő eszközök nem tudnák ellátni ezt a feladatot. Fizikailag a kapcsolást apró tükrökkel oldják meg. A tükör a névleges hullámhosszból kiindulva igen kicsi és még mozognia is kell. Ezeket az eszközöket mikroelektronikus-mechanikus rendszernek hívják (MEMS).



5. ábra: MEMS tükör és tükrökből álló kapcsolótábla [14]

Egy-egy ilyen tükör átmérője akár 0,8 mm is lehet, irányzópontossága pedig eléri a $0.5\mu\text{m}$ -t (viszonyítás képpen egy emberi hajszál 0,05mm). Ezekből a tükrökből készítenek lapokat (amik közel $0,5\text{cm}^2$ alapterületűek, vastagságuk kevesebb, mint $1\mu\text{m}$) [14]. A tényleges modult, ami a kapcsolást végzi, OXC-nek (Optical Cross Connect – Optikai keresztkapcsoló) hívják. Az olcsóbb OXC modulok hullámhossz-konverzióra nem képesek, tehát az adott nyalábot csak továbbítják, hullámhosszán módosítani nem tudnak. A gyakorlatban ez azt eredményezi, hogy az adatok két végpont (végeztetés) között végig ugyanazon a hullámhosszon haladnak.

Végeredményben tehát egy DWDM alapú optikai gerinchálózat olyan csomópontokból áll, melyek hullámhossz konverzió nélkül más útvonalra tudják kapcsolni az egyes fény-nyalábokat, míg igény esetén végeztetni is tudják azokat az adott csomóponton.

2.5 Fogalmak

Egy hálózat működtetése során felmerülő költségeket OPEX-nek (operational expense), míg a fejlesztésekhez kapcsolódó költségeket CAPEX-nek (capital expenditure) hívja a szaknyelv. Működő hálózatok esetén értelemszerű cél, hogy a működéshez szükséges költségeket minimalizáljuk, ehhez pedig minél kevesebb fejlesztési költséget hozunk létre.

Egy konkrét optikai hálózat általánosan modellezhető gráfokkal, így a következő fogalmakat vezetem be, melyek a továbbiakban a megértéshez szükségesek:

- csomópont (node): A gyakorlatban egy olyan központ, ahol transzponderek és oxc-k vannak.
- él (link): Két csomópontot összekötő optikai kábel.
- port (iface): Az optikai kábelek konkrét végpontja egy csomópontban. Egy csomópontnak több (legalább kettő) portja is van. Minden porthoz tartozik egy kapcsoló, ami az adott hullámhossz jelét egy másik portra kapcsolja (transzfer igény) vagy végezteti a transzponderen.
- igény (demand): két csomópont között teljes hullámhosszcsatornát lefoglaló elvezetés
- útvonal (route): szakaszlista, ami megadja, hogy az igény a két végeztető csomópont között melyik linkeken halad keresztül.
- hullámhosszcsatorna (röviden hullámhossz, vagy csatorna): az a hullámhossz, amin az igény elvezetésre kerül.
- hullámhossz gráf: egyértelműen azonosítja az igény elvezetését, azaz megadja a relációt útvonal és hullámhossz között [1].

2.5 Teszt hálózat

Kutatásom során egy nagy magyar telekommunikációs vállalat optikai gerinchálózatát vettem alapul. Tekintve hogy a hálózat a vállalat ipari titkát képezi, így én is csak a kutatáshoz szükséges legfontosabb részletekhez jutottam hozzá.

Az említett hálózat létrehozásakor 40 hullámhosszcsatorna került meghatározásra (ezeket a továbbiakban párosnak fogom hívni). A legutóbbi kapacitás növelés során a hálózati csomópontok nagy részére olyan eszközök kerültek, melyek kétszer annyi hullámhosszcsatornát képesek kezelni. Az ITU-T ajánlásában szereplő táblázat alapján logikusan következik, hogy a bővítés eredményeként a hálózatban már használt hullámhosszok közé (és nem mellé) kerültek az új, üres hullámhosszok (az új hullámhosszok a párosak közé kerültek, így a továbbiakban páratlannak fogom hívni őket). Ezáltal létrejött, 80 hullámhosszt kezelő hálózatban a kihasználtság meglehetősen egyenlőtlené vált. A terv az, hogy a rendelkezésre álló tartomány egyik végébe rendezzük a jelenlegi (10 Gbps) összeköttetéseket, helyet teremtve a tartomány másik végében későbbi (100 Gbps) összeköttetéseknek. A hálózat jelenlegi formájában is tovább kell üzemeljen, tehát nem történhet olyan módosítás, ami egy aktuálisan használatban lévő hullámhosszt érint. Kutatásom célja az, hogy különböző módszerekkel optimális megoldást találjak erre.

3 Kezdeti állapot

3.1 Hálózat szerkezete

A hálózatot leíró adathalmaz négy, vesszővel tagolt fájlban álltak rendelkezésre. Ezek az alábbiak:

```
nodes (node_id)
```

Csomópontokat leíró fájl, csak az azonosítókat tartalmazza.

```
ifaces (node_id, port_id, xconn, oddwl)
```

Egy adott csomópontba befutó optikai kábel csatlakozási pontja. Paraméterei hogy melyik csomópontban található, saját sorszáma, képes-e kapcsolásra (van-e benne oxc), valamint képes-e páratlan hullámhosszok végeztetésére (transzponder képessége). Fontos megjegyezni, hogy a transzponder és az oxc egymástól független eszközök. Előfordulhat olyan csomópont ami képes ugyan a páratlan hullámhosszok kapcsolására, de végeztetni nem tudja azt.

```
links(link_id, snode_id, sport_id, dnode_id, dport_id, length)
```

Konkrét optikai kábeles kapcsolat két csomópont között. Paraméterei az azonosítója, az általa összekötött két csomópont (source és designation) és az ott kapcsolódó interfész azonosítója, valamint a saját hossza.

```
demands(demand_id, snode_id, dnode_id)
```

Két tetszőleges csomópont közötti adatkapcsolat. Paraméterei az azonosítója, valamint a két csomópont azonosítója, amik között működik.

```
routes(demand_id, seq, link_id, wl)
```

Adott igény konkrét útvonala. Paraméterei az igény, az útvonalon belül a kiinduló csomóponttól hányadik él, magának az érintett élnek az azonosítója, valamint hogy az igény melyik hullámhosszon halad.

A fentebbi adatokból racionalizált sémákat hoztam létre, hogy azok az optimalizálás során könnyen felhasználhatóak legyenek. Ezt többek között az indokolja, hogy a jelenlegi struktúra egy bővebb adatbázis részhalmaza, melyből attribútumok törlésre kerültek. Emiatt a sémák normalizálási foka csökkent.

A két különböző feladat külön struktúrát igényel. Az első esetben érdemes észrevenni, hogy sok minden nem változik. Az igény ugyanazokat a csomópontokat fogja érinteni, tehát ahol eddig áthaladt, ott hullámhossz módosítása után is át fog tudni. Amit ellenőrizni kell, hogy a végpont alkalmas-e a fogadására (kezel-e páratlan hullámhosszokat). Ehhez az igényeket tartalmazó relációba számítás alapján felvettem egy új attribútumot `may_odd_wl` néven, ami 1 ha az igény végpontjai alapján elvezethető páratlan igényen és 0. Példaként az igény egyik végére a számítás:

```
firststep:=routes(demand).first_link
where(demand.snode_id=firststep.snode_id)
      firstiface:=firststep.sport_id

if(firstiface.oddw1)
  return 1
else
  return 0
```

Látható hogy a számítás négy relációt is érintett, ami nagymértékben növeli a számítási időt. A kódban egyszerűsítésre is került, de azért érdemes megfigyelni, hogy egy linknek van forrása és célja, de ez nem feltétlenül egyezik meg a rajta áthaladó igény forrásával és céljával, hiszen lehet, hogy az éppen fordítva lett felvéve.

További attribútumként felvételre került a demand relációba egy `changed` érték, ami 1 ha már változtattuk a hullámhosszát és 0 ha még nem. A további számítások megkönnyítése érdekében szintén hozzászámításra kerül az igény által igénybevett hullámhossz értéke is, így a reláció végleges formája:

```
demands (demand_id, snode_id, dnode_id, wl, changed, may_odd_wl)
```

3.2 Előzetes eredmények

Dr. Szigeti János hasonló alapokról kereste a hálózatban az optimumot. Ő különböző engedményekkel (befektetett erőforrásokat nem számolva, módosítás lépéseit nem követve) kereste a hálózat végső kapacitását. Kidolgozott olyan algoritmusokat, melyek többek között képesek voltak a rendelkezésre álló igényeket a lehető legkevesebb hullámhosszon elvezetni vagy éppen egyenletesen egy megadott tartományban. Eredményei alapján arra jutott, hogy a számomra is rendelkezésre álló információk alapján az igények mindössze 23 hullámhosszcsatornán elvezethetők [2]

Természetesen ezek az információk a gyakorlatban dimenzionálás céljából hasznosak, de a konkrét eredmények valóságba átültetése ebben a formában megvalósíthatatlan (tulajdonképpen a hálózat teljes leállításával járna).

3.3 Konkrét feladat

A Magyar Telekom részéről felmerült további konkrétumok, melyek a feladatot specializálják:

- A változásokat minél kevesebb lépésben kell alkalmazni. Számukra egy igény hullámhosszának a módosítása egységnyi munkával jár, így törekedni kell arra, hogy egy igény áthelyezésekor az már rögtön a megfelelő helyre kerüljön. Ez azt is jelenti, hogy az optimalizáló algoritmus bonyolultsága tetszőleges lehet, az egyetlen optimalizálási paraméter a módosítások száma.
- Működésben ne okozzon fennakadást a hálózat átalakítása. Ez azt jelenti, hogy egy hullámhosszt nem szabadíthatunk fel, amíg a rajta lévő igény nem került teljesen áthelyezésre (smooth transition).
- A meglévő csomópontoknak csak egy része képes 80 hullámhosszcsatorna végeztetésére, így figyelni kell arra, hogy bizonyos csomópontok csak minden második (páros) hullámhossz kezelésére alkalmasak.
- A rendelkezésre álló nyolcvan hullámhosszcsatorna három részre legyen osztva. Egyik felére a 100Gbit/s sávszélességű igények fognak kerülni a jövőben, másik

felére az aktuális 10Gbit/s sávszélességű igények, míg köztük 6 hullámhosszcatornányi védősávot kell hagyni. Hogy a határ hova kerüljön, az az optimalizálás függvényében dől el.

Ezen kritériumok mellett a hullámhosszokat 1-től 80-ig számoztam, a határvonalat pedig változóként kezeltem és ennek a minimumát kerestem, az igényeket pedig a határpont fölé próbáltam elhelyezni.

A fentebb említett szempontok alapján az alapvető stratégiára három lépést dolgoztam ki:

- Első lépésben a kijelölt védősávban és az alatt található igényeket a felső tartomány páratlan hullámhosszaira próbálom meg áthelyezni, amíg van számukra szabad hely (mivel eddig a páratlan hullámhosszok kihasználatlanok voltak, cél először ezek feltöltése, törekedve a hálózat egyenletes kihasználtságára).
- Ha az első lépésben maradt elvezetetlen igény, akkor második lépésben megpróbálom a felső tartomány páros vagy páratlan hullámhosszaira áthelyezni őket.
- Ha ezek után még mindig marad áthelyezendő igény, akkor a felső tartományban lévő igényeket helyezem át valamilyen módon úgy, hogy így teremtek helyet. Végül ismét megpróbálkozok az első vagy második lépéssel.

Annak ismeretében, hogy optimális eredmény elérhető, azaz az igények elvezethetők 23 hullámhosszcatornán is, előre tudható, hogy a feladat kivitelezhető, csak megfelelő algoritmust kell találni rá.

4 Kidolgozott algoritmusok és eredményeik

4.1 Hullámhosszok szerinti átrendezés

Három különböző algoritmust készítettem a feladat végrehajtására. Mindegyik a megadott adatstruktúrán dolgozik, bemenő paramétere a tartomány határa, kimenete az új elrendezés. A határ megválasztásában eredetileg nem volt kikötés, így az első futtatásokat a 44-80 tartományra végeztem (kijelöltem a középén a 6 hullámhossznyi védősávot és a rendezésre a felső tartományt választottam).

A kiinduláskor rendelkezésre álló adatok alapján 48 csomópont van, közöttük összesen 67 link fut, melyeken 219 igény van elvezetve. Amennyiben a 44-80-as tartományt tekintjük ideális célnak, így az igények 42.46%-a van (96 darab) megfelelő helyen. Ez azt jelenti, hogy 126 igényt kell áthelyezni, amiből következik, hogy a feladat 126 lépésből megoldása az ideális megoldás, ennél jobb nincsen.

Első lépésként kézenfekvőnek tűnt, hogy az optimalizálásra valamilyen mohó algoritmus adaptációt alkalmazzak. Végül két megoldás mellett döntöttem. Az egyik a kijelölt tartomány minimumától kezdi el inkrementálisan a feltöltést, míg a másik a maximumától dekrementálisan. Ezen felül további attribútumok számítása után egy ezekből heurisztikusan optimalizáló algoritmust is írtam. Ennek nyilván hosszabb lesz a futásideje, de kevesebb lépésből oldja meg a problémát.

4.1.1 Alulról töltő mohó módszer

Az eredetileg meghatározott 3 lépés ebben az esetben a következőképp alakul:

Első lépés:

```
route_is_free:=true
border:=43

for(∀demand|demand->w1<border)
  for(w1:=border to 80 step by 2)
    for(demand->route)
      if(!is_free(route->link))
        route_is_free:=false
      if(route_is_free)
```



```
demand->w1:=w1
break
```

Minden igényt, ami a határon kívül halad, sorra veszünk. A második ciklus a jó hullámhosszokon halad végig kettesével (páratlanról indul, így végig páratlanra próbáljuk áthelyezni). Minden hullámhossz esetén megvizsgáljuk az igény teljes útvonalát és hogy ott már használatban van-e az adott hullámhossz. Ha szabad, áthelyezzük oda.

Az algoritmus nem mérlegel, egyszerűen az első elérhető szabad helyre áthelyezi az igényt és halad tovább. Elképzelhető, hogy ez kevés és nem kerül áthelyezésre az összes igény, ekkor jön a második lépés:

```
route_is_free:=true
border:=44

for(∀demand|demand->w1<border)
  for(w1:=border+1 to 80 step by 2)
    for(demand->route)
      if(!is_free(route->link))
        route_is_free:=false
    if(route_is_free)
      demand->w1:=w1
      break
```

Ez csak annyiban tér el az előző lépéstől, hogy itt most páros hullámhosszról indul a keresés és így páros hullámhosszra is fog átkerülni az igény. Ha ez sem jelentene megoldást, van egy harmadik lépés:

```
route_is_free:=true
border:=43

for(∀demand|demand->w1>=border& demand->w1%2=0)
  for(w1:=border to 80 step by 2)
    for(demand->route)
      if(!is_free(route->link))
        route_is_free:=false
    if(route_is_free)
      demand->w1:=w1
      break
```

illetve

```
route_is_free:=true
border:=43

for(∀demand|demand->w1>=border& demand->w1%2=0)
```

```

for(wl:=border to 80 step by 1)
  for(demand->route)
    if(!is_free(route->link))
      route_is_free:=false
    if(route_is_free)
      demand->wl:=wl
      break

```

A harmadik lépésben a már jó helyen lévő igényeket rendezzük át, hogy ezzel helyet teremtsünk, melyhez két különböző megoldás is elképzelhető. Az első csak azokat az igényeket rendezi át, amik páratlan hullámhosszra tudnak kerülni, ezzel páros hullámhosszt szabadítva fel. A másik változat nem nézi a célt, csak a legközelebbi alkalmas helyre helyezi át az igényt. Ennek a célja az aktuális helyzet megbolygatása, a már jó helyen lévő igények kevesebb hullámhosszra történő tömörítése. Ebben lépésben nem lesz mérhetően jobb a helyzet, így ezután ismét le kell futtatni a 2. lépést.

4.1.2 Felülről töltő mohó módszer

A felülről töltő módszer lényegében azonos az előzővel, csak a hullámhosszok bejárásának iránya változik:

```

route_is_free:=true
border:=43

for(demand|demand->wl<border)
  for(wl:=80 to border step by -2)
    for(demand->route)
      if(!is_free(route->link))
        route_is_free:=false
    if(route_is_free)
      demand->wl:=wl
      break

```

Különbség az eredmények terén mégis várható, hiszen a hálózat eredeti állapotba véletlenszerűnek mondható.

4.1.3 Heurisztikus módszer

Ahhoz, hogy a módosítást ésszerűbben el tudjuk végezni, további mellékszámításokra van szükség. Két további relációt vetettem fel számítás alapján:

```

demand_wl_option(demand_id, sum_of_wl, odd_wl)

```

azaz egy igény számára hány különböző hullámhossz érhető el, illetve az igény áthelyezhető-e páratlan hullámhosszra. Erre azért van szükség, mert elképzelhető, hogy lesznek olyan igények, amiket szinte mindegyik hullámhosszra áthelyezhetünk és lesz olyan, amit csak néhányra. Ebből az adathalmazból rendezés útján helyezzük át azt az igényt, amelyiknek a legkevesebb választási lehetősége van.

Számítása a következőképpen történik:

```
sum_of_wl [number_of_demands]
for(demand)
    for(links|links=demand->links)
        for(wl)
            if(has_no_demand(wl))
                sum_of_wl[demand]++
```

Illetve:

```
wl_demand_option(wl, sum_of_demand)
```

azaz adott hullámhosszra hány különböző igény helyezhető át. Erre azért van szükség, mert elképzelhető, hogy lesznek telített és kevésbé telített hullámhosszok. Az előzőleg kiválasztott, csekély lehetőségekkel rendelkező igényt megpróbáljuk olyan hullámhosszra áthelyezni, amire minél kevesebb igénynek van szüksége.

Számítása a következőképpen történik:

```
sum_of_demand [80]
for(wl)
    for(links|links=demand->links)
        for(demand)
            if(demand->wl<border)
                sum_of_demand[wl]++
```

Ennek a számításnak két célja is van. Az egyik hogy az áthelyezéseket minél kevesebb lépésben oldjuk meg. Előfordulhat olyan eset, mikor egy igényt feleslegesen részesítünk előnyben és úgy helyezzük át, hogy egy másik igény lehetőségeit szűkítjük. A másik cél hogy az áthelyezés után a hullámhosszak szerinti elosztásban egyenletesebb képet kapjunk, mint a mohó algoritmusoknál, ahol várhatóan elég egyenetlen lesz az eloszlás.

Ha az optimalizálandó tartomány túl kicsi lenne előfordulhat, hogy olyan igényekhez jutunk, amikhez tartozó `sum_of_wl` érték 0, azaz nincs olyan hullámhossz, ahova át lehetne helyezni, pedig erre szükség lenne (őket a továbbiakban lehetetlen igénynek fogom hívni). Ilyen esetben megnézzük, hogy melyik igények azok, amik egy lépésben áthelyezhetőek, ezzel helyet teremtve a lehetetlen igényünknek.

```

if(demand->sum_of_wl=0)
  impossible_demand := demand

for(demand|demand->link=impossible_demand->link)
  if(can_be_optimized(demand))
    optimized(demand)

```

Így egy rekurzívan hívható függvényt kaptunk, melynél már csak a mélységi szinteket kell ellenőrizni. Ennek a lépésnek a számítási igénye is igen magas, de a folyamat szempontjából szükséges.

Jelenleg már minden részlet rendelkezésre áll, így a teljes algoritmus így néz ki:

```

poor_demand = minimum(sum_of_wl)->demand

if(poor_demand->sum_of_wl=0)
  impossible_demand_r(poor_demand)

wl := order_ascending(sum_of_demand)

for(wl)
  if(is_free(wl_list) on poor_demand->link)
    poor_demand->wl = wl

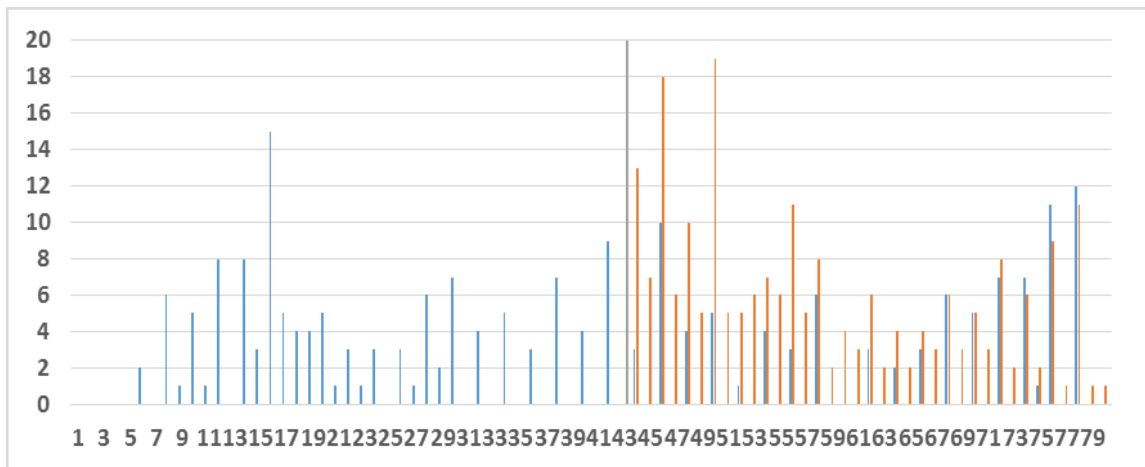
```

4.2 Eredmények

A különböző algoritmusok eredményei az alábbi táblázatban láthatóak, ha a cél a 43-80-as tartomány.

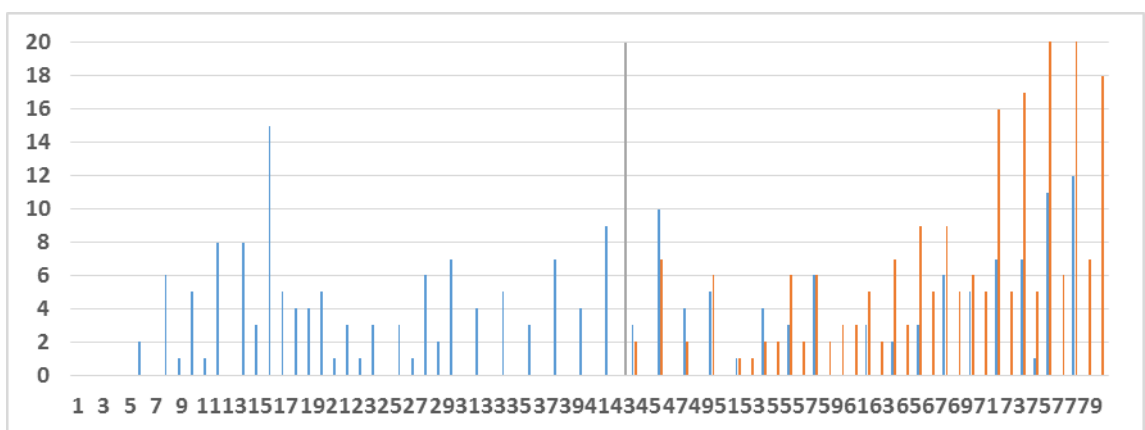
Algoritmus	1. lépés	2. lépés	3. lépés	eredmény
Alulról töltő mohó	41 lépés (61.18%)	84 lépés (99.54%)	23 lépés (100%)	148 lépésben siker
Felülről töltő mohó	41 lépés (61.18%)	84 lépés (99.54%)	23 lépés (100%)	148 lépésben siker
Heurisztikus	126 lépés (100%)	-	1 lépés (99.54%)	127 lépésben siker

Meglepő módon a két mohó algoritmus ugyanannyi lépésben jutott el a sikeres megoldáshoz, ám nyilván különböző eredményekkel. Mindhárom esetben 1-1 igény volt, amit nem sikerült egy lépésben áthelyezni. Utóbbi esetben heurisztikusan megtalálta az algoritmus azt az egy igényt, aminek áthelyezésével helyet tud teremteni, míg a mohó algoritmusok erre nem voltak képesek. A gyorsaság ára volt, hogy 23 igényt helyezett át a harmadik lépés, nyilvánvalóan ebből 22 felesleges volt.



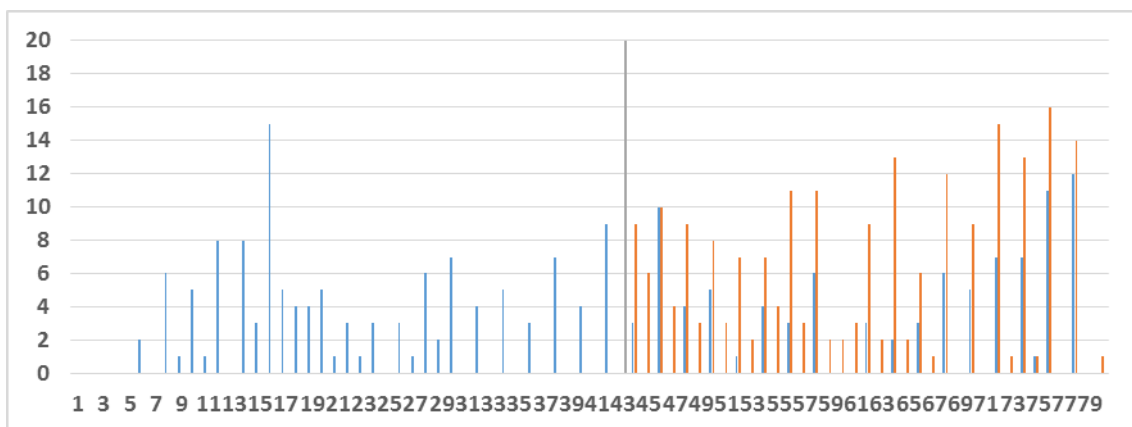
6. ábra: alulról töltő mohó módszer eredményei 44-80 tartományra

Az igények hullámhosszávonkénti eloszlása viszont várhatóan látszódik. Az 6. ábrán látszik, hogy a 44-80 tartomány alsó részén tömörödnek az eredmények, míg a 7. ábrán viszont ennek ellenkezője, hogy a tartomány felső részén sűrűsödtek össze az igények.



7. ábra: felülről töltő mohó módszer eredményei 44-80 tartományra

Ezen a heurisztikus módszer sokat javít. A 8. ábrán látszik, hogy lényegesen egyenletesebb elosztással sikerült az igények áthelyezése.

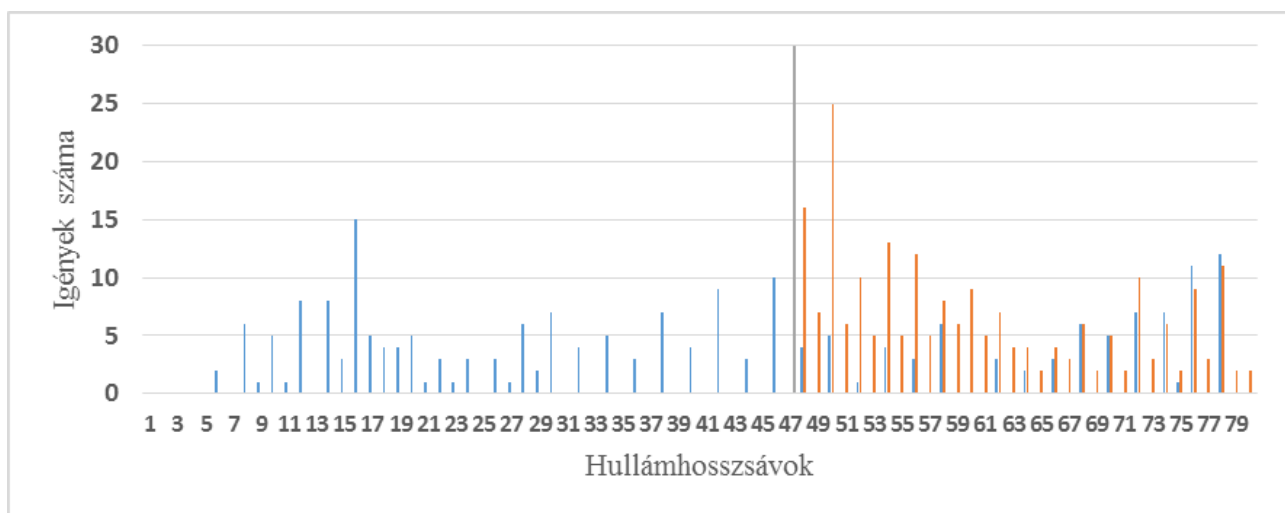


8. ábra: heurisztikus módszer eredményei 44-80 tartományra

A grafikonokon észrevehető, hogy a páros hullámhosszú sávokon még mindig sokkal több igény van elvezetve, mint a páratlanokon. Ez azzal magyarázható, hogy egyrészt az optimalizálásnak nem volt célja a kiegyenlítés (bár részben törekedtem rá), másrészt a csomópontok nagy része nem tudja kezelni a páratlan hullámhosszokat még.

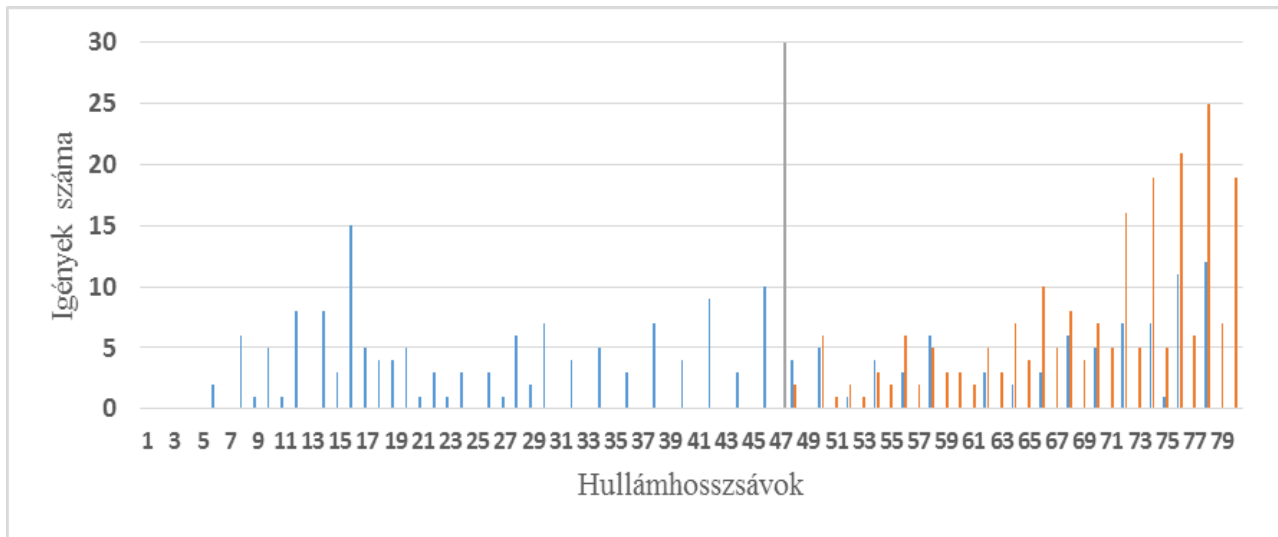
4.3 Újabb futtatások más tartományokon

A fentebbi eredmények alapján gyorsan (kevés lépésben) sikerült elérni a célt, még a mohó algoritmusok is jól teljesítettek. Felmerült így a gondolat, hogy a rendelkezésre álló tartományt szűkítsem és így is végezzek futtatásokat. Vegyük például a 48-80 tartományt.



9. ábra: alulról töltő mohó módszer eredményei 48-80 tartományra

Az 50-es hullámhossz sávba már 25 igény került (9. ábra), ezen érezhető hogy sokat tömörítettünk az igényeken, de a teljes optimalizálás még így is 154 lépésben kivitelezhető volt. Ugyanezen tartományom a felülről töltő algoritmusnak 156 lépés volt szükséges a teljes optimalizáláshoz, melynek eredménye a 10. ábraán látható.



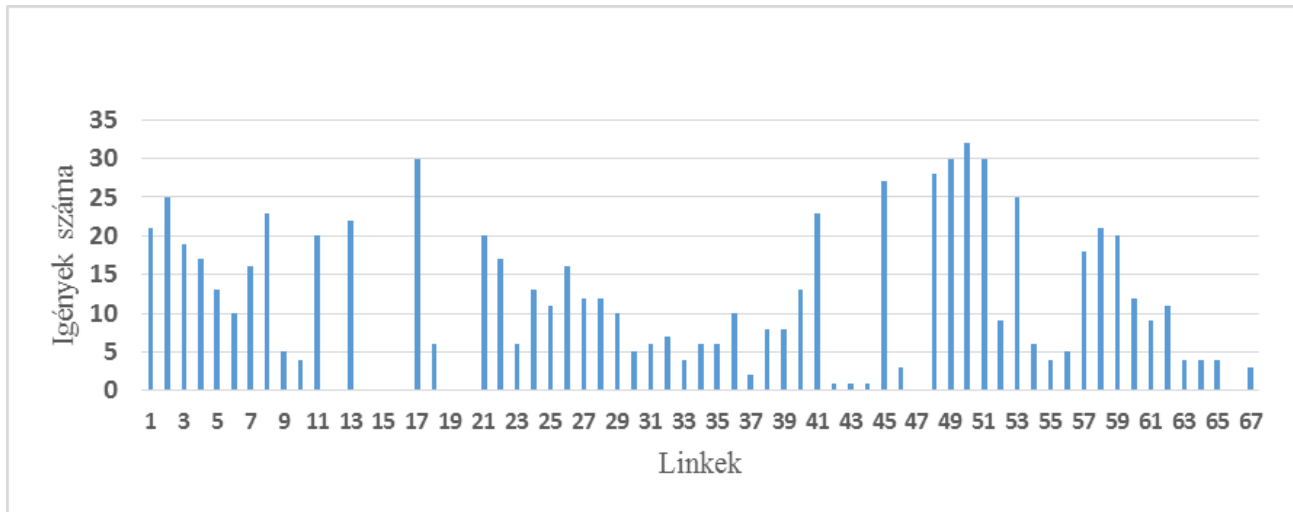
10. ábra: felülről töltő mohó módszer eredményei 48-80 tartományon

Újabb mérést végeztem 52-80 tartományra, ahol 29 hullámhosszcsatornán kell elvezetni a 219 igényt. Alulról töltő mohó módszerrel 157 lépés után 99,54%-ig jut el. E felett már nincsen olyan igény, aminek az áthelyezésével helyet tudnánk teremteni, az algoritmus futna a végtelenségig. A felülről töltő algoritmus még kevésbé tud sikert elérni ezen a tartományon, 163 lépésben jut el a 99,08%-os állapotig, 412 lépés után jut el a 99,54%-ra, de a teljes optimalizálást nem tudja elérni.

Az eredmények ellenőrzésekor derült ki, hogy kézzel sem optimalizálható tovább a hálózat, mert bizonyos linkek olyan terheltek, hogy egyszerűen nincs szabad hullámhossz sáv. Például az 51-es linken 32 igény van átvezetve ami egyrészt alsó korlátot ad az optimalizálás ezen formájának, másrésztől tovább szűkíti a lehetőségeket hogy ezen igényen nagy több mint fele olyan csomópontokat köt össze, melyek nem alkalmasak páratlan hullámhossz kezelésére. Ezen eredmények számos fejlesztés forrását teremtik meg.

Eddig az adatokat pusztán a hullámhossz sávok szempontjából vizsgáltam, de érdemes megnézni másik szemszögből is. Ha megnézzük, hogy az egyes linkeken hány igény

kerül elvezetésre, akkor egyértelműen láthatóvá válik, hogy az igényen zöme a linkek kis százalékán halad keresztül (11. ábra). Ez azért is probléma, mert néhány link kiiktatásával a hálózati forgalom nagy része kiesik. A hullámhosszokon való igényáthelyezés ezeken az eredményeken nyilvánvalóan nem változtatott, így kézenfekvő optimalizálási lehetőség lenne, hogy az igényeket más útvonalakon vezessük el.



11. ábra: Igények száma linkenként

4.3 Útvonalmódosítással történő átrendezés

Építsük fel a hálózat gráfját új módon. Vegyük először a csomópontokat illetve az összekötő éleket. Az így kapott 48 csúcsú gráfot, mint egy réteget, másoljuk fel önmaga fölé 79-szer, tehát kapunk 80 szintet. Minden szinten csak azokat az igényeket vegyük fel, amelyek az adott szint indexe szerinti hullámhosszon működnek. Ekkor az adott szinten az egyes igények részére lehetőségünk van hullámhosszát módosítása nélkül új útvonalakat találni például Dijkstra algoritmussal.

A cél elsősorban nem is biztosan az útvonal rövidítése, hiszen nanoszekundum nagyságrendű változásokról van szó, a jel két végpont közötti továbbításában nem a hálózaton keresztüljutás igényli a legtöbb időt. A nagyobb lehetőség abban rejlik, hogy bizonyos linkeket leilthatunk (például olyan linkeket, amik túlterheltek), így találva biztonságosabb, jobb utat az igény számára.

4.5 Dokumentálás

Minden egyes futtatás során a módosításokat dokumentálni kellett a jövőbeni, lépésről lépésre történő módosítás okán. Ehhez a következő relációt alkalmaztam a hullámhosszok módosítása esetében:

```
change_log_wl(change_id, demand_id, old_wl, new_wl)
```

ahol a módosításokat azonosítóval láttam el, feljegyeztem az érintett igényt, valamint hogy melyik hullámhosszról melyikre lett áthelyezve.

Útvonalak módosításakor a következő relációt alkalmaztam:

```
change_log_wl(change_id, demand_id, link_id, type)
```

ahol a módosítások azonosítója mellett az érintett igényt jegyzem fel, valamint a módosításban érintett útvonalat. Előfordulhat, hogy az új útvonal nem egyezik az előzővel lépésszámban, így ezeket nem párban tároltam el, hanem egy type jelzéssel láttam annak megfelelően hogy az adott módosítás milyen formában érinti a linket (az igény távozik a linkről vagy az igény újonnan került rá a linkre).

A dokumentálást az eredeti formátumnak megfelelően vesszővel tagolt, fejléccel rendelkező fájlokba végeztem.

5 Összefoglalás, jövőbeni lehetőségek

5.1 Specifikáció teljesülése

A módosítások száma jól közelíti az elvi optimális korlátot már a mohó algoritmusok esetében is. Elvileg egy igény által használt hullámhossz módosítása néhány nap alatt kivitelezhető, így a teljes optimalizálás kevesebb, mint egy év alatt kivitelezhető. Ha azzal is számolunk hogy a módosítások nagy része párhuzamosítható, az eredmények igen sikeresnek mondhatóak.

A puha átállás maradéktalanul megvalósul, hiszen csak úgy módosítható egy igény hullámhosszhasználata, ha az előzőleg már szabad volt.

A többi feltételt az algoritmusok kidolgozásakor feltételként szintén beépítettem, így folyamatosan ügyeltem arra, hogy egy páratlan hullámhosszsávon közlekedő igény ne köthessen össze olyan csomópontokat, amik nem képesek azt kezelni. Az eredetileg számolt felosztást is sikerült csökkenteni, azaz az eredetileg 40, később 80 csatornát igénybe vevő hálózat végül 30 hullámhosszcsatornán elhelyezhetővé vált.

Minden lépés a 4.5 Dokumentálás pont alattiak szerint részletesen feljegyzésre került.

5.2 Tapasztalatok összefoglalása

A feladat rávilágított arra, hogy az optimalizálásnak számos módja van és meglepő módon a mohó algoritmusok is lehetnek jók. Ahogy Cinkler Tibor és Szigeti János is munkájukban azt vették észre, hogy a véletlenszerűen meghatározott módosítások többször hatékonyabbak voltak a többszörösen átgondolt és irányított módosításokkal szemben.

Az algoritmusok futásidejét tekintve a két mohó algoritmus az igénylista szisztematikus, lineáris bejárását igényli, így lépésszáma az igények számával lineárisan becsülhető. A harmadik algoritmusról ez nem mondható el. Minden ciklusban kiszámolt függőségek az igények számával szintén lineárisan becsülhetőek felülről (olyannyira, hogy a futás előrehaladtával fogynak az áthelyezendő igények, a számítás gyorsasága is növekszik).

Emellett a kialakult függőségek rendezése, igények ismételt bejárása és egyenkénti áthelyezése erősen lassítja az algoritmust. Elméletben ez még mindig felülről becsülhető az igények skalárszorosaival, ám lényegesen lassabb, mint az első két változat. Az algoritmus hatékonyabbá tételén lehet még dolgozni.

Az optimalizálás során kiderült, hogy a hálózat több szempontból egyenlőtlen kialakítású, számos szempontból fejlesztésre szorul. Többek között sok útvonal kihasználatlan, aminek az oka az, hogy olyan csomópontokon halad keresztül, melyek nem tudnak kapcsolni, csak a csomópontban végeztetni az igényeket. Egy oxc beépítésével új, szinte üres útvonallal lehetne a többi terheltségét csökkenteni. A hullámhossz szerinti optimalizálásnak további korlátot ad, hogy több olyan útvonal is van, melynek végpontjai nem képesek páratlan hullámhosszok kezelésére. Vagy a csomópontok fejlesztésére volna szükség (precízebb transzponderek beépítése) vagy az adott útvonalon elvezetett igények összefogására és 10 Gbps elvezetés helyett 100 Gbps elvezetni, ami szintén a csomópont fejlesztését igényelné.

5.3 Jövőbeni lehetőségek

A kidolgozott módszerek hasznos eredményeket hoztak és a jövőben is rugalmasan felhasználhatóak a hálózat további optimalizálásakor. Érdeemes megjegyezni, hogy az alapvetően alkalmazott hullámhosszsávokat módosító optimalizálást ötvözni lehetne az útvonalakat módosító módszerrel, ezek együttesen számos új eredményt hozhatnak. Kevésbé ésszerű megoldás lehet a két elvet egymást után alkalmazni, de érdekesebb algoritmust kidolgozni arra, amely talál súlyozható attribútumot és ez alapján ki lehet egyensúlyozni a számítást a két elv alkalmazása között.

A vizsgált hálózatban lévő eszközök egyike sem képes hullámhosszkonverzióra, azaz a csomópontban az adott igény hullámhosszát módosítani. Ez a funkció nagy mértékben növeli a hálózat komplexitását és összetettségét, ám további optimalizálási lehetőségek rejlenek benne. A fejlesztés tulajdonképpen a kapcsolásban történne, a csomópontok ettől függetlenül ugyanúgy képesek a jelet feldolgozni (O/E/O átalakítással).

További lehetőségek rejlenek a párhuzamosításban. Több olyan igény is van, amely ugyanazokat a csomópontokat, ugyanazon az útvonalon köti össze. Ennek inkább

biztonságtechnikai, mintsem optimumkeresési céljai vannak. Hiszen ha két csomópont kapcsolata egy link állapotán múlik, akkor az összeköttetés egyáltalán nem redundáns. Tekintve hogy egy sérült optikai kábel javítása akár napokig is elhúzódhat, ezen szinten való gondolkodás egy szolgáltató részére igen fontos lehet, hiszen a szolgáltatás minőség garantálása köti.

Már nagy előrelépést nyújthat, ha az egyes elvezetéseket link-diszjunktá tesszük, de a teljes függetlenségük elérése lenne az ideális állapot. Köztes megoldásként szolgálhat ha védelmi utakat határozunk meg az egyes igények részére, azaz szabad hullámhosszavokat-útvonalakat biztosítanánk kiesés esetére. Ilyenkor a vonal áthangolása után a szolgáltatás zökkenőmentesen folyhatna tovább, melyet a javítás immár nem akadályozna.

Irodalomjegyzék

- [1] Dr. Szigeti János. „Network state advertisement and p-cycle protection for reliable connections” PhD thesis, BME TMIT 2010
- [2] Dr. Cinkler Tibor, Dr. Szigeti János. Hullámhossz kiosztási stratégia kidolgozása vegyes 10G/100G rendszerekre, BME-TMIT 2014
- [3] Biswanath Mukherjee (1997): *Optical Communication Networks*. McGraw-Hill, Inc. USA (ISBN 0-07-044435-8)
- [4] Michael Pioro, Deepankhar Medhi (2004): *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers (ISBN 0-12-557189-5)
- [5] ITU-T kiadvány DWDM alapú optikai átviteli közegű rendszerek megvalósítására http://www.itu.int/dms_pub/itu-t/oth/1D/01/T1D010000090001PDFE.pdf (2014.10.21)
- [6] ITU-T Study Group 15 – Távközlő média és optikai rendszerek karakterisztikája G.600–G.699 ajánlások család - <http://www.itu.int/rec/T-REC-G/en>
- [7] ITU-T G.694.1 ajánlás a spektrum felosztásáról WDM alkalmazásokhoz, DWDM frekvencia táblázat <http://www.itu.int/rec/T-REC-G.694.1-201202-I/en> (2014.10.21)
- [8] ITU-T G.652 Egymódosú optikai szálak karakterisztikája. <http://www.itu.int/rec/T-REC-G.652-200911-I/en> (2014.10.21)
- [9] Yasin, Zafar. *Optical Fiber Communication*. előadás : Stanford University. <https://www.slac.stanford.edu/slac/sass/talks/opticalfiber.pdf> (2014.10.21)
- [10] Jeff Hecht (2011): „Ultrafast fibre optics set new speed record”, *NewScientist*, 2809. kiadás <http://www.newscientist.com/article/mg21028095.500-ultrafast-fibre-optics-set-new-speed-record.html> (2014.10.21)
- [11] Fiberdyne Labs weboldala <http://www.fiberdyne.com/products/itu-grid.html> (2014.10.21)
- [12] Nippon Telegraph and Telephone kiadvány 2006. szeptember 29. <http://www.ntt.co.jp/news/news06e/0609/060929a.html> (2014.10.21)
- [13] Koaxiális kábelek paramétereit http://www.madaboutcable.com/cables/coaxial_cables/products/rg400_coaxial_cable.html (2014.10.21)
- [14] Fiberguide Industries weboldala, *2D Fiber Optic Arrays* <http://www.fiberguide.com/product/2d-fiber-optic-arrays/> (2014.10.21)