

Multicast fák rendszeres újrakonfigurálása többrétegű optikai hálózatokban

Perényi Marcell, Soproni Péter, Cinkler Tibor
{perenyim, soproni, cinkler}@tmit.bme.hu

Kivonat. Cikkünkben dinamikusan változó multicast fákkal foglalkozunk kétrétegű optikai hálózatokban. A levél csomópontok állandó váltakozásával a fa egyre távolabb kerül az optimális topológiától. Ezért sok hálózati erőforrás és költség takarítható meg a fa rendszeres újrakonfigurálásával, melynek során az optimális topológiát állítjuk vissza. Vizsgáljuk az újrakonfigurálás eredményességét több dinamikus útvonalválasztó algoritmus és az újrakonfigurálási intervallum hosszának függvényében.

Kulcsszavak: optikai hálózat, dinamikus multicast, újrakonfigurálás, ILP, heurisztika

1. Bevezetés

Az elmúlt években a multicast hálózati alkalmazásoknak is köszönhetően a gerinchálózatok forgalma folyamatosan növekedett. Számos nagyon fontos, szélessávú alkalmazás sorolható a pont-multipont kategóriába, mint például a digitális média szétosztás (pl. IP-TV, IP-rádió, stb.), VoD (Video On Demand) médiafolyamok továbbítása, ill. távoktatási vagy virtuális magánhálózati szolgáltatások [1].

A rendkívül kedvező sávszélesség megtakarítás ellenére jelenleg a legtöbb kereskedelmi Internet-szolgáltató – különböző gyakorlati problémák miatt – a végfelhasználók számára nem teszi lehetővé a multicast szolgáltatások használatát. Ennek következtében rengeteg sávszélességet pazarolunk el a multipont forgalmak alkalmazásszintű multicast (Application Layer Multicast, ALM) módszerrel történő kiszolgálásával, ami gyakorlatilag unicast forgalom-szétosztást jelent. Az egyik nemrégiben megjelent, növekvő népszerűségű, jelentős sávszélesség-igényű alkalmazás, mely a multicast szolgáltatás beindítására sarkallhatja a szolgáltatókat, a peer-to-peer alapú műsorterjesztés (*TV peer-casting*). Ez az alkalmazás – szükségtelenül – nagy mértékű hálózati kapacitást emészt fel, hiszen ugyanaz a médiafolyam megy be és ki több ezer felhasználóhoz unicast átvitel segítségével.

Noha a végfelhasználók számára nem érzékelhető, a multicast szolgáltatás a gerinchálózatokban is fontos szerephez jut. Segítségével biztosítható televíziós csatornák hatékony és jól skálázható szétosztása, továbbítása a tartalom szerzőjétől (terjesztőtől) a helyi szolgáltatók felé. A végfelhasználókkal közvetlen kapcsolatban álló helyi szolgáltató a műsorfolyam szórásán (*broadcast*) túl annak cache-elését is végezheti. A televíziós műsorszolgáltatás a szolgáltatók által nyújtott *triple-play* csomag fontos részét képezi.

Általánosságban elmondható, hogy a multicast szolgáltatás megvalósítása a hálózati hierarchia lehető legalsóbb rétegében a leggazdaságosabb. Ugyanakkor, ha az alsó rétegben alkalmazott technológia kapcsolatorientált (mint például az optikai hálózatok esetében), akkor a kiépíthető kapcsolatok száma korlátozó tényezővé válik. A hullámhosszkapcsolt optikai hálózatok esetében ez a korlátozás több részből tevődik össze: a hullámhosszak, a kapcsolókban lévő (optikai jelet) elágaztató eszközök, valamint ezek kimeneteinek korlátozott száma, ill. a jel teljesítményének változása az elágaztatás következtében (fizikai hatások). Mindezen korlátozásokat tekintve a hullámhossz-fák kiépítése, karbantartása és optimalizálása igen lényeges és nagy kihívást jelentő probléma az újgenerációs, multicast képességgel rendelkező optikai hálózatokban.

Cikkünkben dinamikus multicast fákkal foglalkozunk, melyekben a levél csomópontok folyamatosan cserélődnek. Új levél csomópontok (célcsomópontok) léphetnek be a fába az érkező tartalom elérés céljából, míg más tagok kijelentkezhetnek, hogy esetleg visszatérjenek majd egy későbbi időpontban. Ez egy olyan forgatókönyvnek felel meg, ahol az IP multicast tagság szabja meg az optikai fa kiépítését. A gyakorlatban egy fát több kisebb multicast kapcsolat, vagy néhány igen nagy sávszélességű multicast igény kiszolgálása érdekében építünk ki.

Egy tipikus alkalmazás lehet egy digitális médiaterjesztő, szétosztó rendszer, ahol a közönség időben változik. Új ügyfelek jelenhetnek meg, akik előfizetnek a tartalomra, más felhasználók – akiknek esetleg lejárt az előfizetésük – elhagyják a hálózatot. A felhasználók ebben az esetben nem feltétlenül egyéni felhasználókat jelentenek, hanem inkább helyi szolgáltatókat (pl. helyi kábeltévé szolgáltató), akik végfelhasználók egy csoportját testesítik meg.

Egy másik alkalmazási példa lehet egy virtuális magánhálózati szolgáltatás (VLAN), ahol a LAN üzenetszórást el kell juttatni minden végponthoz. Az előző példával ellentétben ez a szolgáltatás kevésbé érzékeny az átvitelben fellépő kisebb megszakadásokra, melyek az újrakonfigurálás alatt jelentkezhetnek. Kivétel ez alól, ha a VLAN forgalom VoIP forgalmat is tartalmaz, mert ez késleltetés-érzékeny.

A multicast fa tagjainak állandó cserélődése a fa „elromlását” eredményezi hálózati- és erőforrásköltségek szempontjából. Ez a folyamatos „leromlás” a fa – bizonyos időközönkénti – rendszeres újrakonfigurálásával állítható meg. Az újrakonfigurálás, melynek során a fa optimális topológiája kerül visszaállításra, jelentős hálózati erőforrás – és költség – megtakarítást eredményezhet, mely a szolgáltató számára igen előnyös: a felszabadított erőforrások (ideértve a linkkapacitásokat is) hasznosíthatók más célokra.

Mindazonáltal van néhány hátulütője is az újrakonfigurálásnak:

- Az optimális topológia meghatározása igen számításgépes lehet, mivel a Steiner-fa kiszámítása NP-teljes probléma. Ugyanakkor jelentős időmegtakarítás érhető el gyors heurisztikus módszerek segítségével, melyek kompromisszumos megoldást jelentenek a sebesség és az optimalitás között.
- Az újrakonfigurálás rövid megszakadást okozhat az adatátvitelben, mely adatvesztést, kiesést eredményezhet egyes adatfolyamokban, vagy a csomagok sorrendjének megváltozását. Ez bizonyos alkalmazásokban elfogadhatatlan, ezért ki kell védeni valamilyen technikával.
- Az újrakonfigurálások többlet jelzési forgalmat generálnak.

1.1. Az újrakonfigurálás alatt fennálló szolgáltatás-kiesés

Noha cikkünknek nem célja e probléma megoldása, javasolunk néhány módszert, hogy megmutassuk, a probléma kiküszöbölhető.

Egy lehetséges megoldás a megszakadásra érzékeny alkalmazások (pl. média streaming) számára az ún. *soft switch-over (smooth reconfiguration)*. Az eljárás zökkenőmentes átállást biztosít a régi fáról az újra: az új multicast fa már ki van építve, amikor a régi lebontásra kerül. Van egy rövid időszak, amikor mindkét fa egyszerre létezik, és képes adatot továbbítani. Annak érdekében, hogy az adatfolyam folyamatosága ne törjön meg a fa megváltozásakor, a csomagok (keretek) átvitele felfüggeszthető egy rövid időre a forrás csomópontban, hogy biztosítsuk minden csomag „kiürülését” a régi fából. A másik megoldás, hogy az új fában érkező első néhány csomagot eltávolítjuk a kimeneti csomópont(ok)ban, amíg az átvitel végét jelentő jelzési csomag meg nem érkezik a régi fában.

Mindezekkel együtt a soft switch-over többlet erőforrásokat igényel a hálózatban. A mi egyszerű modellünkben, amennyiben minden linken rendelkezésre áll egyetlen szabad hullámhossz, akkor egy darab multicast fa újrakonfigurálása lehetséges – ILP optimalizálással. DWDM hálózatokban, ahol a hullámhosszak száma 30-nál is több, ez a plusz kapacitás elfogadható (különösen, ha összevetjük az optimalizálásból adódó jelentős kapacitás-nyereséggel). Ugyanakkor természetesen nem biztos, hogy ez a többlet kapacitás minden pillanatban rendelkezésre áll.

1.2. Kapcsolódó publikációk

Eddig viszonylag sok cikk jelent meg dinamikus multicast fák optimalizálásának témakörében optikai hálózatokban. Mivel az igények optimális elvezetése technikailag gyakran nem megoldható vagy időigényes, ezért sok heurisztikus megközelítés született. Ezek teljesítményét általában ILP-vel számolt optimális megoldáshoz is hozzámérték.

Statikus multicast igények elvezetését vizsgálták gyűrű és háló topológiájú, hullámhossz-kapcsolt optikai hálózatokban többek között a [2, 3]-ban. A [3] szerzői a kötegelés problémájára írtak fel egy analitikus modellt – nem lineáris programozási feladat formájában. Az eredményt heurisztikákkal hasonlították össze. További heurisztikus optimalizáló algoritmusokat mutat be a [4, 5, 6]. A [7] szerzői ILP formulák segítségével oldották meg a hullámhossz hozzárendelés és elvezetés problémáját. Megmutatták, hogy már viszonylag kevés hullámhosszosztó és konverter használatával is hatékonyan kezelhető a multicast probléma. Mustafa és szerzőtársai [8] szintén ILP, ill. heurisztika segítségével az elektromos rétegbeli eszközök és hullámhosszok számát minimalizálták.

Napjainkban a dinamikus változó multicast fák problémájával többen foglalkoznak. A dinamikus problémánál a cél általában a blokkolási ráta minimalizálása, nem pedig az összes igény elvezetése (különböző kényszerek betartásával), mint a statikus probléma esetében. Ez általánosságban még a statikus esetnél is erőforrás- és számításgépesebb. Azt tapasztaltuk ugyanakkor, hogy az útvonalválasztás néhány részproblémája (egyetlen fa vagy több fa külön-külön való optimalizálása) megoldható – elfogadható időn belül – ILP [9] segítségével optimális módon. Így lehetővé válik a dinamikus elvezetési algoritmusok eredményeinek összehasonlítása az optimális megoldásával, valamint a nyereség meghatározása.

Dinamikus fák elvezetésére és karbantartására (*routing and provisioning*) – kötegelésre képes optikai hálózatban – számos algoritmust mutat be [10, 11, 12].

[13]-ban dinamikus igények elvezetését vizsgálták forgalomtervezéssel (*traffic engineering*) unicast esetben, kötegelésre képes WDM hálózatokban.

A [14] szerzői egy dinamikus hullámhossz hozzárendelési algoritmust mutatnak be multicast esetben. A cél a blokkolási valószínűség minimalizálása azáltal, hogy minden lépésben maximalizálják a maradék szabad sávzsélességet a linkeken. Chowdhary és szerzőtársai hasonló problémát vizsgálnak [15]-ben, on-line multicast fák karbantartására dolgoztak ki algoritmust. A cél, hogy növeljék a használt eszközök kihasználtságát, és minimalizálják a később érkező igények blokkolási valószínűségét.

[16] szerzői bevezettek egy hullámhosszfa (*light-tree*) alapú védelmi sémát az egyszeres hibák kivédésére. Kidolgoztak egy ILP felírást a javasolt módszerhez szükséges minimális többlet sávzsélesség meghatározására.

Tudomásunk szerint még nem született olyan publikáció, mely dinamikus multicast fák rendszeres újrakonfigurálásának hatásait vizsgálná, elemzi a dinamikus útvonalválasztó algoritmusok eltérését az optimális topológiától összehasonlítva a dinamikusan változó költséget az optimummal.

2. A probléma megfogalmazása

Kétrétegű hálózatot tételeztünk fel: a felső, elektronikus réteg időosztásra, míg az alsó, optikai réteg hullámhossz-kapcsolásra képes. Az elektronikus réteg forgalomkötegelést is végezhet, tehát több, alacsony sávzélességű igény fogható össze (*multiplexálás*) egyetlen hullámhossz csatornába. A két réteg a *peer* (társ, vertikálisan összekapcsolt) *modell* [17] szerint működik együtt. Ennek megfelelően – az elvezetés során – a vezérlő sík számára mind az elektronikus, mind az optikai réteg állapotinformációi rendelkezésre állnak. Ez lehetővé teszi, hogy mindkét rétegre kiterjedő optimális megoldást találjunk.

A hálózati topológiát, az összeköttetéseket alkotó szálak számát és a forgalmi igények leírását előre ismertnek tekintjük. Emellett a használható hullámhosszak számát, azok kapacitását, ill. az forgalomelvezetés elemeinek költségeit (pl. térkapcsolás, O/E konverzió) is rögzítettnek vesszük.

Multicast igényeket tételezünk fel a hálózatban, amelyek dinamikusán változnak. A korábban leírtaknak megfelelően az igények egyetlen igen nagy sávzélességű IP multicast kapcsolatnak vagy több kisebb kapcsolat (amelyek a levél csomópontok többségében osztoznak) összefogásának felelnek meg. Nem foglalkozunk a cikkben azzal a problémával, hogy hogyan érdemes különböző kapcsolatokat összeszervezni egy multicast igényre. Hasonló megközelítést érvényes a multicast fák (hullámhossz fák) együttes optimalizálásával, ill. azok összevonásával kapcsolatban: az egyszerűség kedvéért a cikkben az egyes hullámhossz fák külön-külön kerülnek optimalizálásra. Természetesen az együttes optimalizálás nagyobb költségnyereséget tenne lehetővé, de jelentősen nagyobb számítási igényt igényel.

A multicast fa olyan ún. „részigények” összessége, amelyek képesek erőforrásokon osztozni a hálózatban, azaz sávzélességük nem adódik össze. Minden cél csomóponthoz egy-egy rész igény rendelhető (minden részigény forrása a multicast fa egyetlen csökér csomópontja). A fa célcsoomópontjainak halmaza dinamikusán változik: új csomópontok léphetnek be a fába, míg mások elhagyhatják azt. Az egyes új részigények útvonalát valós időben kell kiszámítani, míg a távozó csomópontokhoz kapcsolódó utakat a lehető legnagyobb odafigyeléssel kell lebontani, hogy a többi igényt a lehető legkisebb mértékben befolyásoljuk.

Minden célcsoomópontra a kapcsolat tartási ideje és a kapcsolatok közti idő exponenciális eloszlású. A forgalom intenzitása az eloszlások várható értékének (λ paraméter) megfelelő megválasztásával szabályozható. A cél, hogy a forráscsoomópontból minden időpillanatban, minden aktuális célcsoomópontba eljusson az információ.

3. Hálózati modell

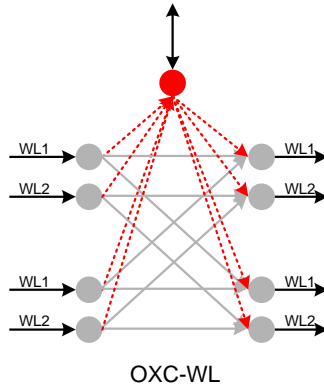
A kétrétegű optikai hálózatot hullámhossz gráffal modelleztük, mely lehetővé teszi az igények elvezetését a két réteg együttes kezelésével, különböző csomópont típusok és hálózati topológiák mellett. A hullámhossz gráf a fizikai hálózathoz származtatható a topológia és a különböző eszközök tulajdonságainak figyelembevételével. A modell egy egyszerű változata [18]-ban került publikálásra.

A RWA (Útvonalválasztás és hullámhossz-hozzárendelési) probléma ILP alapú, forgalomkötegelést tartalmazó megfogalmazása unicast igényekre a [19]-ben olvasható.

Számos eltérő típusú hálózati kapcsolót különböztethetünk meg: Optical Add-and-Drop Multiplexers (*OADM*), Optical Cross-Connect (*OXC*: optikai maggal), Opto-Electrical Cross-Connect (*OEXC*: elektronikus mag), melyek teljes vagy részleges, ill. tiszta optikai vagy elektronikus hullámhossz konverziót támogathatnak. Az eszközök egy része kötegelésre is alkalmas lehet. Ezek a képességek a hullámhossz-gráfokban egyszerűen figyelembe vehetők.

A hálózatot kapcsolóeszközök és összeköttetések (optikai szálak) alkotják. Az optikai szálak végei fizikai eszközök interfészeihez kapcsolódnak. Ez utóbbi határozza meg az adott fényszálban használható hullámhosszak számát. Minden csomópont interfészekből és egy belső kapcsológépből áll. Minden hullámhossz és kapcsológép reprezentálásra kerül a hullámhossz gráfban.

Egy fényszál pontosan annyi logikai éllel kerül megjelenítésre, ahány hullámhosszon terjedhet benne a jel. Egy eszköz logikai ábrázolása az eszköz fizikai képességeitől függ. A gráf minden éle rendelkezik használati költséggel és kapacitással. Az él kapacitása általában a hullámhossz-kapacitással egyenlő, ami a használt vivőtől függ (általában 2.5 vagy 10 Gb/s – a szimulációk során mi az előbbit feltételeztük). Egy él használatának költségét annak funkciója határozza meg (O/E konverzió, térkapcsolás, hullámhossz él).



1. ábra OXC-WL eszközt reprezentáló részgráf

Minden hálózati kapcsolót egy részgráf reprezentál, mellyel az eszköz összes interfészét és az eszköz belső kapcsoló képességét is modellezzük. A hullámhossz-gráf – kiegészítve a későbbiekben bemutatásra kerülő ILP megfogalmazásokkal – lehetőséget biztosít a különböző képességű fizikai eszközök leképezésére, még akkor is, ha azok egy adott hálózatban egyszerre vannak jelen. A modell könnyen kiterjeszhető, fejleszhető. Az egyes eszközök képességeinek változása könnyen követhető új részgráf típusok bevezetésével.

Egy sokoldalú eszköz részgráfiáját mutatja az 1. ábra. Az eszköz egyszerre rendelkezik egy OXC és egy OADM képességgel: lehetőség van igények indítására, végződtesítésére, ill. hullámhossz-konverzióra és kötegelésre. Hullámhossz-elágaztatásra csak az elektromos rétegen keresztül van mód. Az elektromos réteget egyetlen (a legfelső) csomópont reprezentálja. A többi csomópont interfészt reprezentál. Az 1. ábrán látható eszközök két bejövő és két kimenő interfésszel rendelkeznek, melyek mind két hullámhosszt támogatnak. A szimulációk során ezt a csomópont típust használtuk.

4. Útvonalválasztó algoritmusok

Több útvonal választó algoritmust is alkalmaztunk az igények elvezetésére. A cél ezek költségeinek és teljesítményének összehasonlítása volt. Az ILP alapú optimális útvonalválasztást tekintjük referenciának. A Dijkstra algoritmuson alapuló „legrövidebb utak láncolata” pedig egy igen egyszerű mohó módszer. Ezek mellett két feszítőfa módszeren alapuló heurisztikát is kipróbáltunk. A módszerek közötti különbséget szemlélteti a 2. ábra.

4.1. ILP útvonalválasztás és felírása

ILP segítségével meghatározható a hálózatban lévő igények optimális elvezetésének költsége. Ezért minden összehasonlítás alapjául szolgál. Természetesen az optimális költség nem azt jelenti, hogy az így kapott konfigurációban lévő erőforrások (pl. használt hullámhosszak, O/E, E/O átalakítók, stb.) száma mind minimális. Ugyanakkor az ILP megoldása kiszámítás nagyon sok időbe telhet. Szerencsére egyetlen multicast fa optimális elvezetésének meghatározása elfogadható idejű még meglehetősen nagy hálózatokban is. A megoldási idő – szimulációink szerint – 3 és 180 másodperc között változik a – 28 csomópontból és 41 linkből álló – COST266 [20] hálózatban egy 2.8 GHz-es Pentium D számítógépen. Ha több fát egyszerre vezetünk el, a költség megtakarítás még ennél is jelentősebb lehet, de a megoldás meghatározásához szükséges idő elfogadhatatlanul megnő. Ezért az egyetlen lehetőség, hogy a multicast fákat egymástól függetlenül vezessük el egyesével.

Az ILP egyik komoly hátránya, hogy az egymást követő megoldások nagymértékben eltérőek, így az igények útvonalának (beleértve az út során érintett kapcsolókat is) újrakonfigurálása elkerülhetetlen.

A következő ILP megfogalmazás több multicast fa együttes, optimális elvezetését teszi lehetővé a hálózatban:

$z_{ij}^{or} \in \{0, 1\}$: az r multicast fa o részigénye használja-e az (i, j) élt vagy sem.

$x_{ij}^r \in \{0, 1\}$: az r (unicast vagy multicast igény) használja-e (i, j) élet.

$y_{ij} \in \{0, 1\}$: az (i, j) élt használja-e bármelyik igény.

$$\sum_{\forall j \in V_i^+} z_{ji}^{or} - \sum_{\forall k \in V_i^-} z_{ik}^{or} = \begin{cases} -1 & \text{if } i = s^r \\ 0 & \text{if } i \notin \{s^r, t^{or}\} \\ +1 & \text{if } i = t^{or} \end{cases} \quad (0.1)$$

minden $i \in V$ (logikai) csomópontra, r igényre és o részigényre

V_i^+ jelenti azon csomópontok halmazát, amelyek i -ből elérhetők kimenő élen. V_i^- azon csomópontokat reprezentálja, melyekből i elérhető irányított élen át. A, V, V_E, O, R jelentése sorrendben a következő: élek, csomópontok, elektromos csomópontok, részigények, végül igények halmaza. Az r igény forrását s^r , míg nyelőjét t^{or} jelöli, ahol o a részigény azonosítója.

$$z_{ij}^{or} \leq x_{ij}^r, \forall (i, j) \in A, \forall o \in O, \forall r \in R \quad (0.2)$$

$$x_{ij}^r \leq \sum_{o \in O} z_{ij}^{or}, \forall (i, j) \in A, \forall r \in R \quad (0.3)$$

$$\sum_{j \in V_i^+} x_{ji}^r = \sum_{k \in V_i^-} x_{ik}^r \leq 1, \forall i \notin V_E, \forall r \in R \quad (0.4)$$

$$\sum_{j \in V_i^+} x_{ji}^r \leq \begin{cases} 0 & \text{if } i = s^r \\ 1 & \text{if } i \neq s^r \end{cases}, \forall i \in V_E, \forall r \in R \quad (0.5)$$

$$\sum_{r \in R} x_{ij}^r \cdot b^r \leq B_{ij}, \forall (i, j) \in A \quad (0.6)$$

$$x_{ij}^r \leq y_{ij}, \forall (i, j) \in A, \forall r \in R \quad (0.7)$$

$$y_{ij} \leq \sum_{r \in R} x_{ij}^r, \forall (i, j) \in A \quad (0.8)$$

$$\sum_{j \in V_i^+} y_{ji} = \sum_{k \in V_i^-} y_{ik} \leq 1, \forall i \notin V_E \quad (0.9)$$

Változók:

$$z_{ij}^{ro} \in \{0, 1\}, \forall (i, j) \in A, \forall o \in O, \forall r \in R \quad (0.10)$$

$$x_{ij}^r \in \{0, 1\}, \forall (i, j) \in A, \forall r \in R \quad (0.11)$$

$$y_{ij} \in \{0, 1\}, \forall (i, j) \in A \quad (0.12)$$

Célfüggvény:

$$\text{Minimalizálandó } \sum_{(i, j) \in A} c_{ij} y_{ij} \quad (0.13)$$

(0.1) a folyammegmaradás törvényét fejezi ki a részigényekre. (0.2) szerint egy multicast fa használ egy adott (i, j) élet, ha bármelyik részigénye áthalad rajta. (0.3) az előző fordítottja: egy (i, j) élet csak akkor használ egy fa, ha legalább egy részigénye áthalad rajta. Ez biztosítja, hogy fölöslegesen ne foglaljunk le kapacitást. (0.4) biztosítja, hogy igény ne tűnhessen el, illetve ne ágazhasson el olyan csomópontban, ahol ez nem engedélyezett. (0.6) szerint az adott (i, j) élen áthaladó igények sáv szélességeinek összege nem haladhatja meg az él (hullámhossz) kapacitását. (0.7) biztosítja, hogy egy élen csak akkor haladhasson át egy igény, ha az használatra le van foglalva. (0.8) ismét a fölösleges lefoglalást akadályozza meg: csak akkor kell lefoglalni egy élet, ha azon legalább egy igény áthalad. (0.8) elhagyható, mivel ezt a célfüggvény implicit tartalmazza. (0.9) nagyon hasonlít (0.4)-re, csak eggyel magasabb absztrakciós szinten. (0.9) elhagyható (mert redundáns kényszer), azonban gyorsíthatja a megoldást.

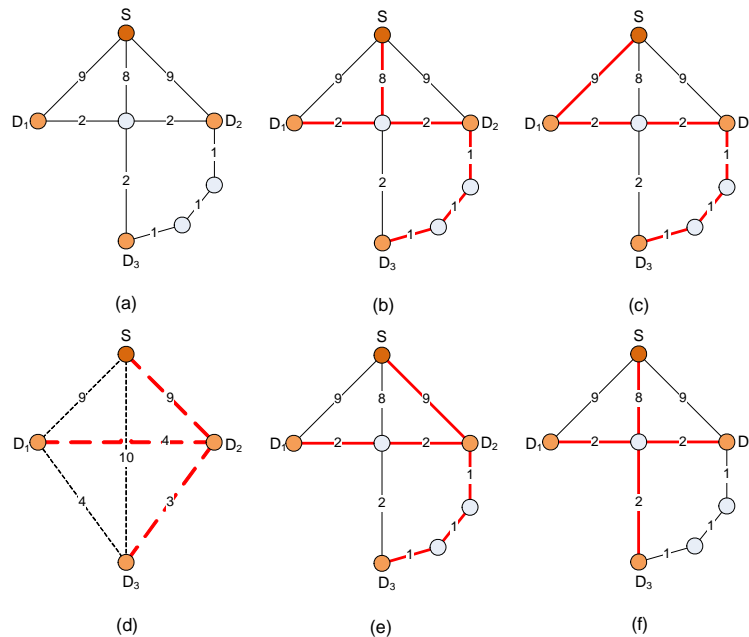
A célfüggvény (0.13) kifejezi, hogy a lefoglalt élek összköltségének minimumát keressük. Célunk tehát egy minimális költségű elvezetés megtalálása.

4.2. Legrövidebb utak láncolata (Dijkstra algoritmus)

Legrövidebb utak láncolatán alapuló algoritmus gyors és egyszerű. Anélkül használható új levélcsoomópontok fába való becsatlakoztatására, hogy ez hatással lenne a már bent lévő részigények útvonalára. Ugyanakkor a módszer költségpazarló.

Legrövidebb utak láncolata algoritmus a következőképpen működik: minden levélcsoomóponthoz egy „részigényt” rendelünk, a részigények útvonalai egymás után kerülnek kiszámításra a levélcsoomópontok és a forrás között. Az algoritmus közvetlenül a logikai hálózatot tekinti. A részigény forrás és cél csomópontja egyaránt a hullámhossz gráf egy-egy elektronikus csomópontja. A fa által éppen használt élek költsége nulla, tehát ezeket egy új részigény ingyen használhatja.

A fát elhagyó igények útvonalai törölődnek. Minden olyan él, melyet már nem használ a multicast fa (tehát egyetlen részigény sem használja már), felszabadításra kerül. A módszer sohasem változtatja meg a már elvezetett részigények útvonalát, ami végeredményben gyakran hosszabb útvonalakhoz, nem optimális megoldásokhoz vezet.



2. ábra Az eredeti topológia a forráscsomóponttal és három levélcsozópponttal (a), feszítőfa útvonalválasztás (b), legrövidebb utak láncolata (c), MPH virtuális topológia és útvonalválasztás (d), MPH útvonalválasztás (e), ILP optimális útvonalválasztás (f)

4.3. Legrövidebb út heurisztika (Minimal Path Heuristic, MPH)

Az MPH algoritmus az eredeti gráfot egy virtuális gráffá transzformálja, majd ezen alkalmazza Prim módszerét [21]. Így próbálja meghatározni a legjobb megoldását. A virtuális gráf egy teljes hálózat, ahol a forrást, ill. minden célt egy-egy pont reprezentál. A virtuális hálózat minden éle a legolcsóbb utat reprezentálja a valós gráfban az él kiinduló és végpontja között. Az élek költsége megegyezik a reprezentált út összköltségével. Ennek megfelelően a módszer alkalmazásához meg kell határozni a forrás és a nyelők, ill. a célok egymástól vett távolságát. Ez utóbbiakat két irányban is.

A Prim algoritmust a virtuális hálózaton kell alkalmazni. Miután a minimális költségű feszítőfa meghatározásra került, annak egyes éleit vissza kell vezetni az eredeti hálózatba, azaz a reprezentált utat le kell foglalni. Amennyiben új csomópont kerül hozzáadásra a hálózathoz, az új fa számítása során a már használt élek költségét nullára kell állítani. Ez garantálja, hogy a már elvezett részigények útvonala sose változzon. A részletek [22]-ben olvashatók.

4.4. Feszítőfa útvonalválasztás (Tree routing)

Ez az algoritmus nagymértékben hasonlít az MPH-hoz. Az eltérés annyi, hogy a Prim algoritmust ezúttal közvetlenül a hullámhossz gráfon alkalmazzuk, nem a virtuális gráfban. A kiszámított feszítőfának a multicast részfa által nem használt élei eltávolításra kerülnek. A fa frissítése és az élköltségek módosítása hasonlóan történik az előzőekhez.

Mind az MPH, mind a feszítőfa alapú útvonalválasztásnál problémát jelent, hogy elvezetéskor az újonnan felvett részigények olyan (nem elektromos rétegbeli) csomópontokban is szétághozhatnak, ahol ez nem engedélyezett. Az ilyen eseteket egy utómunkálati fázissal meg kell szüntetni. A probléma könnyen megoldható az elágazásnak az elektromos rétegbe való áthelyezésével.

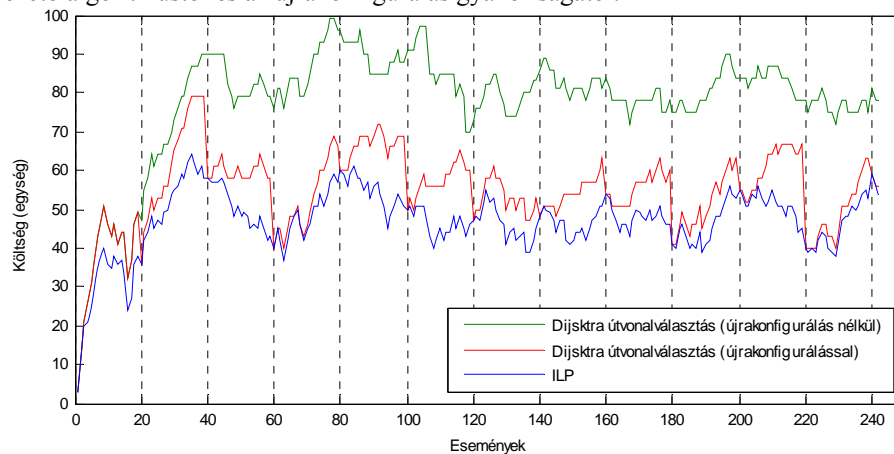
5. Eredmények

A szimulációkat a COST266 Európai referencia hálózatban [20] végeztük, minden csomópont csak elektronikus osztóképességgel rendelkezett. Minden futtatás során ugyanaz a dinamikus igényhalmaz került behelyettesítésre.

A 3. ábra a teljes elvezetés költségét szemlélteti a bekövetkezett események függvényében. Esemény alatt a célcsozóppontok halmazának megváltozását értjük. Az alsó (kék) görbe a mindenkor optimalis elvezetés költségét mutatja. A felső (zöld) a Dijkstra alapú elvezetést jelöli újrakonfigurálás nélkül. A középső (piros) 30 eseményenkénti újrakonfigurálás mellett mutatja a költségek alakulását.

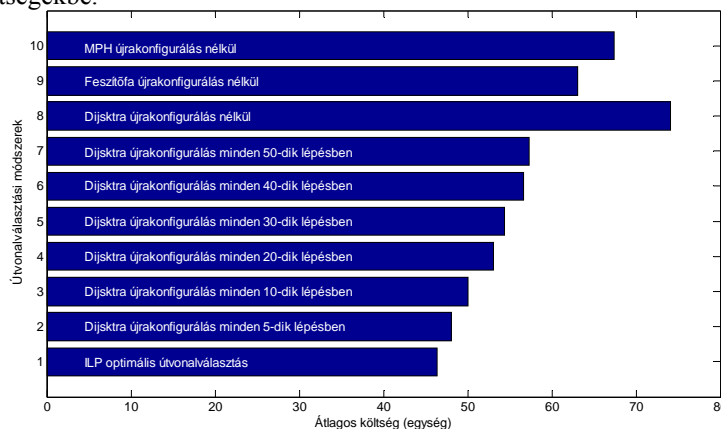
A szimuláció során Dijkstra algoritmusá átlagosan több mint 60 százalékkal felülmúlta költségben az optimalis megoldást. Az újrakonfigurálásos görbe általában gyorsan távolodik az optimalis megoldástól. A divergálás egészen addig folytatódik, amíg nem következik be a következő újrakonfigurálás. Habár az újrakonfigurálás

egyértelműen kedvező (3. ábra szerint), mégis jósága egyértelműen függ a hálózat topológiájától, a használt dinamikus elvezető algoritmustól és az újrakonfigurálás gyakoriságától.



3. ábra, Igények elvezetésének költsége a bekövetkezett események számának függvényében, Dijkstra algoritmusával, újrakonfigurálással és a nélkül, az ILP optimális útvonalválasztáshoz hasonlítva

Ezért megvizsgáltuk, hogy a különböző algoritmusok (részletek a 4. fejezetben) költségei hogyan viszonyulnak egymáshoz és a Dijkstra algoritmushoz különböző újrakonfigurálási időket (periódusidő) feltételezve. Az eredményeket a 4. ábra mutatja. Egyértelmű, hogy a különböző újrakonfigurálás nélküli algoritmusok, a jelen szimuláció szerint, az optimálistól igen távol esnek. Átlagosan 35-60% körüli mértékben múlják felül az optimális megoldást. Ugyanakkor jelentős megtakarítás érhető el ismétlődő újrakonfigurálással. A várakozásoknak megfelelően a kisebb újrakonfigurálási idő kedvezőbb átlagos költséget jelent. Természetesen az újrakonfigurálás számításgépes, és más hátrányokkal is rendelkezhet (1.1 fejezet). Ezeket a hátrányokat nem számítottuk bele a költségekbe.

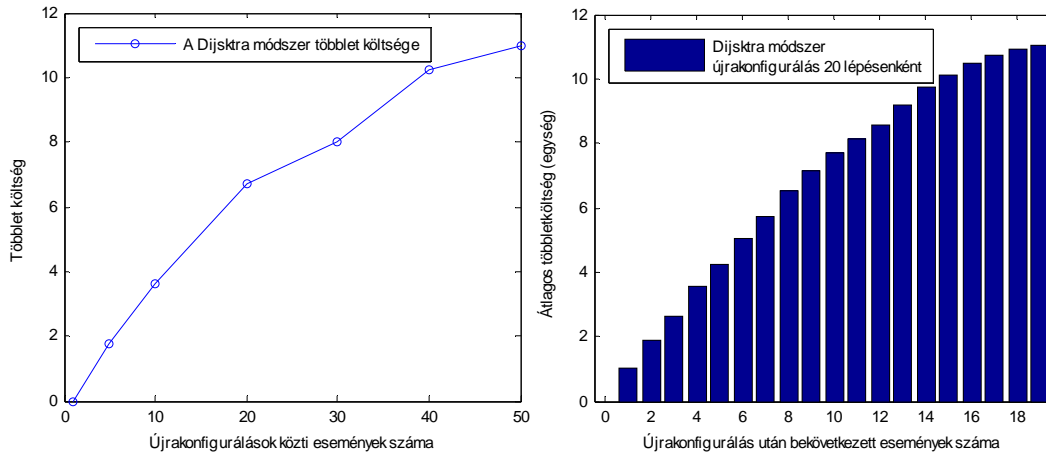


4. ábra, Az egyes algoritmusok átlagos útvonal-választási költsége, ill. a legrövidebb út módszerének (Dijkstra) költsége különböző újrakonfigurálási értékek mellett

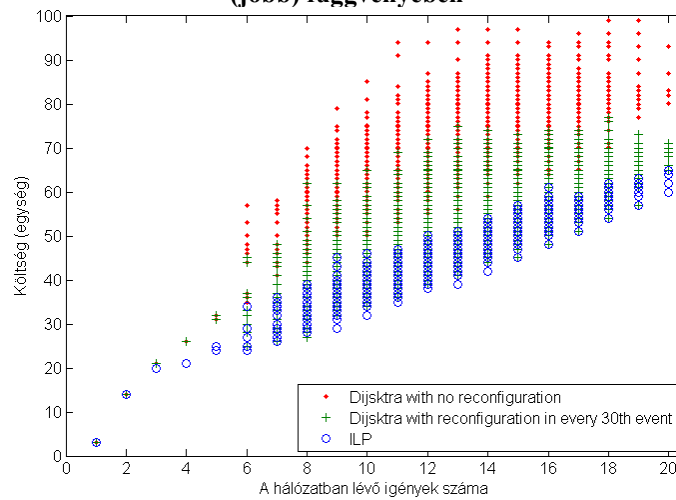
Szintén megvizsgáltuk, hogy az újrakonfigurálási periódus hossza, hogyan befolyásolja az átlagos többletköltséget (veszteséget). Az 5. ábra bal fele szemlélteti az átlagos plusz költséget az újrakonfigurálási gyakoriságának függvényében. Az ábra egy folyamatosan csökkenő meredekségű (telítődő, parabola-jellegű) görbét mutat. Tehát nagyobb nyereséghez gyakoribb újrakonfigurálás szükséges. Ritka újrakonfigurálás hozzáadott költségei között nincs jelentős eltérés.

Az 5. ábra jobb oldala azt szemlélteti, hogy az újrakonfigurálás óta eltelt eseményszám növekedésével milyen gyorsan távolodik a megoldás az optimálistól. Az előzőhöz hasonlóan ez is egy ellaposodó görbe, csökkenő meredekséggel. Ez arra utal, hogy a heurisztikus folytatás az első pár esemény során már nagymértékben távolodik az optimális megoldástól, majd a romlás üteme lassul.

A következő ábra, 6. ábra az elvezetés költségét mutatja az egyes algoritmusok esetében a cél csomópontok számának függvényében. Minden pont a szimuláció során adott pillanatbeli megoldást reprezentál. Ahogyan várható, az elvezetés költsége nő az igények számának növekedésével. Érdekes megfigyelés, hogy az újrakonfigurálás melletti legrövidebb utak módszerének jellemző pontjai általában az optimális megoldáshoz tartozó és az újrakonfigurálás nélküli pontok jellegzetes tartományai között helyezkednek el.



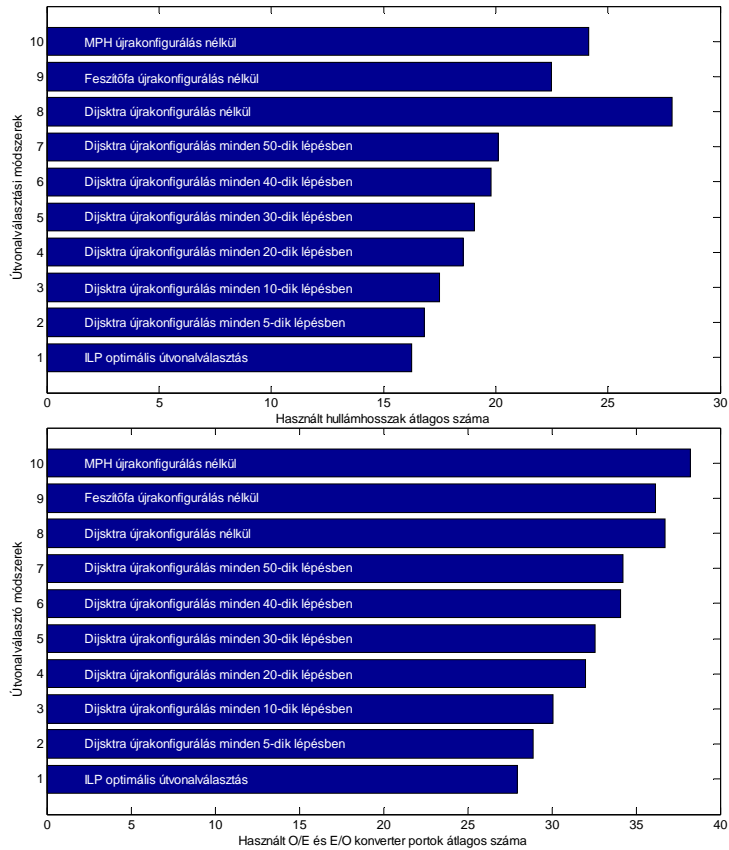
5. ábra, Átlagos plusz költség az újrakonfigurálási gyakoriság (bal), ill. az újrakonfigurálás óta eltelt idő (jobb) függvényében



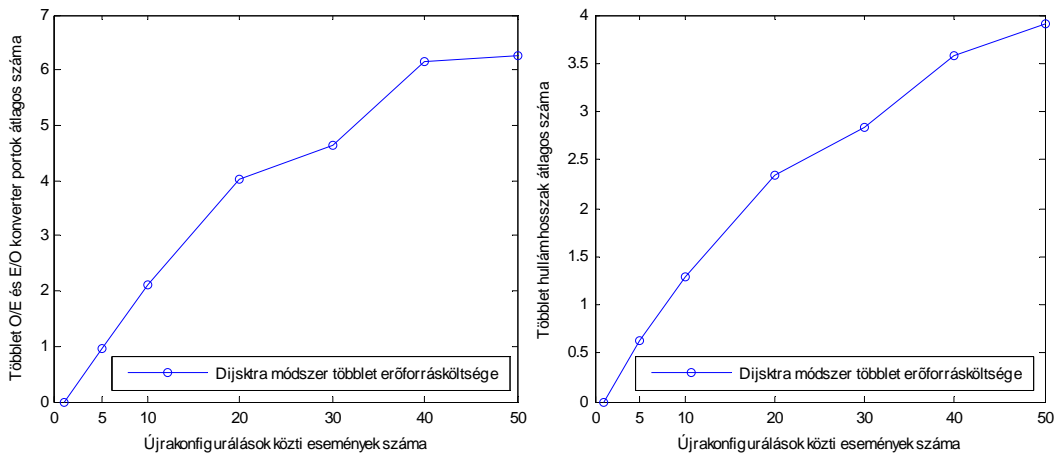
6. ábra, az elvezetés költsége a célok számának függvényében

A költségek mellett más, az elvezetés során lényeges, hálózati elemek kihasználtságát is vizsgáltuk (pl. használt O/E, E/O portok, hullámhosszak szám). A korábbiakhoz hasonló eredményekre jutottunk.

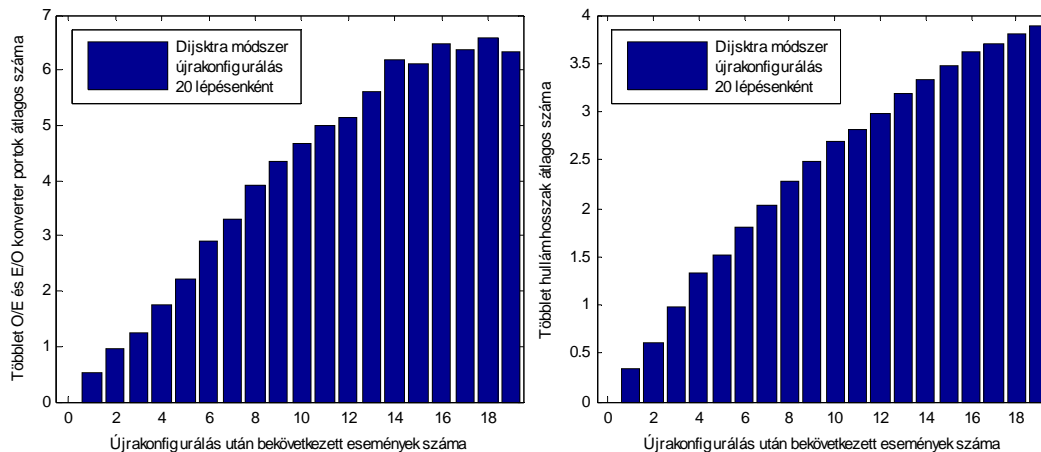
A 7. ábra a szükséges hullámhosszak, ill. O/E és E/O konverziós egységek számát mutatja a különböző algoritmusokra. Lényeges megfigyelés, hogy a Dijkstra módszer kimagaslóan nagy hullámhossz igényű, miközben O/E és E/O port használata kisebb, mint az MPH heurisztikáé. Mindkét erőforrás használtsága csökkenthető az újrakonfigurálási gyakoriságának növelésével.



7. ábra, a szükséges hullámhosszak átlagos száma (felső), ill. a szükséges O/E és E/O konverziós egységek száma (alsó) a különböző algoritmusokkal és újrakonfigurálási periódus hosszokkal



8. ábra, többlet O/E és E/O konverterek (bal), ill. hullámhosszak átlagos száma az újrakonfigurálási periódus hosszának függvényében



9. ábra, többlet O/E, E/O konverterek (bal), ill. hullámhosszak átlagos száma az újrakonfigurálás óta eltelt események számának függvényében

A 8. ábra azt mutatja, hogy hogyan változik a többlet konverziós portok és a hullámhosszak szám az újrakonfigurálások gyakoriságának csökkentésével. A szimulációs eredmények a korábbiakhoz hasonló, telítődő függvényt adnak. 9. ábra alapján a konverziós egységek száma közel lineáris, míg a használt hullámhosszak száma négyzetesen mértékben nő a szaturációs pontig.

6. Összegzés

Cikkünkben megmutattuk, hogy a dinamikusan változó multicast fák esetén az újrakonfigurálás előnyös a hálózati szolgáltató számára. A költségvesztés (beleértve a hálózati erőforrásokat, mint például a hullámhosszak vagy konverziós portok szám) csökkenthető az optimális elvezetéshez való visszatéréssel. Mivel a fa tulajdonságai az újrakonfigurálás után gyorsan romlanak, ezért azt gyakran meg kell ismételni. A cikkben megpróbáltuk megbecsülni a várható költség megtakarítást, ill. a periodikus újrakonfigurálás várható hatásait. Az eredmények arra utalnak, hogy az újrakonfigurálás költség-hatékony megoldás lehet, ha az egyes események közötti átlagos idő elégséges ahhoz, hogy a hullámhossz és egyéb erőforrás megtakarítások ellensúlyozzák az újrakonfigurálási művelet által okozott technikai problémákat (pl. a fennakadás nélküli átállás a régi fáról az újra). Ezeket azonban meg kell megoldani ahhoz, hogy az újrakonfigurálás jól használhatóvá váljon.

7. Irodalomjegyzék

1. B. Quinn and K. Almeroth, "IP multicast applications: Challenges and solutions", IETF RFC 3170, Sep. 2001
2. Madhyastha et al., "Grooming of multicast sessions in WDM ring networks", OptiComm 2003: Optical Networking and Communications, Nov. 2003
3. G. V. Chowdhary and C. S. R. Murthy, "Grooming of Multicast Sessions in WDM Mesh Networks", Workshop on Traffic Grooming, 2004
4. X. Zhang et al., "Constrained Multicast Routing in WDM Networks with Sparse Light Splitting", Journal of Lightwave Technology, vol. 18, issue 12, p. 1917, Dec. 2000
5. X. H. Jia et al., "Optimization of Wavelength Assignment for QoS Multicast in WDM Networks", IEEE Transactions on Communications, vol. 49, no. 2, Feb. 2001
6. Fatih Köksal and Cem Ersoy, "Multicasting for all-optical multifiber networks", Journal of Optical Networking, Vol. 6, Issue 2, pp. 219-238, Jan 2007
7. D. Yang and W. Liao, "Design of light-tree based logical topologies for multicast streams in wavelength routed optical networks," in Proc. IEEE Information Communications (INFOCOM), San Francisco, CA, Apr. 2003
8. R. Mustafa, A.E. Kamal, "Design and provisioning of WDM networks with multicast traffic grooming", IEEE Journal on Selected Areas in Communications, Volume 24, Issue 4, 2006
9. Alexander Schrijver, "Theory of Linear and Integer Programming", John Wiley and Sons, 1998.
10. X. Huang et al., "Multicast Traffic Grooming in Wavelength-Routed WDM Mesh Networks Using Dynamically Changing Light-Trees", Journal of Lightwave Technology, vol. 23, no. 10, Oct. 2005
11. Ahmed E. Kamat et al., "Algorithms for multicast traffic grooming in WDM mesh networks", IEEE Communications Magazine, Volume 44, Issue 11, November 2006
12. Ahmad Khalil et al., "Dynamic provisioning of low-speed unicast/multicast traffic demands in mesh-based WDM optical networks", Journal of Lightwave Technology, Volume 24, Issue 2, Feb. 2006
13. Keyao Zhu et al., "Traffic Engineering in Multi-granularity Heterogeneous Optical WDM Mesh Networks Through Dynamic Traffic Grooming", IEEE NETWORK vol. 17, no. 2, Mar/Apr. 2003, pp. 8-15.
14. Jianping Wang, Biao Chen, "Dynamic Wavelength Assignment for Multicast in All-Optical WDM Networks to Maximize the Network Capacity", IEEE Journal On Selected Areas in Communication", Vol. 21, No. 8, October 2003
15. G. Chowdhary, C. S. R. Murthy, "Dynamic multicast traffic engineering in WDM groomed mesh networks", Workshop

- on Traffic Grooming, 2004.
16. C. Boworntummarat et al., Light-tree based protection strategies for multicast traffic in transport WDM mesh networks with multifiber systems, IEEE International Conference on Communications, 2004 Volume 3, June 2004
 17. E. Dotaro, M. Vigoureux, D. Papadimitriou: "Multi-Region Networks: Generalized Multi-Protocol Label Switching (GMPLS) as Enabler for Vertical Integration", Alcatel Technology White Paper, February 2005
 18. T. Cinkler et al., "Configuration and Re-Configuration of WDM networks", NOC'98, European Conference on Networks and Optical Communications, Manchester, UK, 1998
 19. T. Cinkler, "ILP formulation of Grooming over Wavelength Routing with Protection", ONDM 2001, 5th Conference on Optical Network Design and Modeling, Wien, Feb. 2001
 20. A. Betker et al., "Reference transport network scenarios", Technical report, BMBF-Project MultiTeraNet, 2003. http://www.pt-it.pt-dlr.de/_media/MTN_Referenz_Netze.pdf
 21. Thomas H. Cormen et al., "Introduction to Algorithms", Second Edition, MIT Press and McGraw-Hill, 2001, Section 23.2: "The algorithms of Kruskal and Prim", pp.567–574.
 22. M. Ali, J.S. Deogun, "Cost-effective implementation of multicasting in wavelength-routed networks", Journal of Lightwave Technology, Vol. 18. Issue 12., 2000