Gyors hibajavítás IP hálózatokban

Enyedi Gábor, Rétvári Gábor[1]

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék

{enyedi, retvari}@tmit.bme.hu

Kulcsszavak: IP, hibajavítás, útválasztás, IPFRR, Not-via

Bár az IP komoly fejlődésen ment át, továbbra is hiányzik belőle a hálózatban fellépő hibák gyors elkerülésének módszere. Ezt a rést napjainkban az IP Fast Reroute (IPFRR) megoldások igyekeznek betölteni. Cikkünkben áttekintjük a gyors hibajavítás alapelveit, majd ezen elveket a „Not-via addresses"-en, a legnagyobb támogatással rendelkező módszeren keresztül szemléltetjük. Egy teljes értékű Not-via tesztrendszer tervezése és vizsgálata során szerezett tapasztalataink alapján bemutatjuk a módszer erényeit és hibáit, valamint megoldási javaslatunkat ezen hibák javítására.

1. Bevezetés

Napjainkban az Internet előretörésével a kommunikáció egyre jelentősebb része ezen a médiumon zajlik. A fejlődésnek köszönhetően azonban az Interneten használt alkalmazások köre is jelentősen kiterjedt. A hagyományosnak mondható szolgáltatások mellett – e-mail, web –, a felhasználók újabban (videó)telefonálásra, on-line játékokra, sőt televíziózásra is használni kezdik.

Ezen alkalmazások azonban az Interneten megszokott és jól kezelt elasztikus forgalommal szemben valós idejű követelményeket támasztanak, amellyel egyre nehezebb megbirkózni. A probléma oka az Internet Protocol (IP) alapvető működési elvéből, az úgynevezett *best effort* (legjobb szándék) hozzáállásból fakad. Mivel a tervezés idején nem merülhetett fel a valós idejű forgalom elvezetésének szükségessége, az IP hálózatok mind a mai napig kihívásokkal küzdenek a szolgáltatás minőségének biztosítása (QoS) terén.

Az egyik legfontosabb még ma is megoldatlan feladat a valós idejű alkalmazások számára az esetlegesen felmerülő meghibásások gyors kezelése. Ezt a jelenlegi IP hálózatok tipikusan valamilyen útvonalválasztó protokoll (OSPF, ISIS) segítségével reaktív módon végzik, a hiba létrejötte *után* felderítik a hálózat új topológiáját és átkonfigurálják az útválasztókat. Természetesen ez a megoldás jelentős időt vehet igénybe, a valós idejű alkalmazások által megkövetelt maximálisan 50-100ms-mal szemben a rendszer helyreállása könnyen elérheti a másodperces, de speciális esetben akár a perces nagyságrendet is [1].

A helyzet azonban sokszor még ennél is rosszabb lehet. A jelenlegi hálózatokban ugyanis az IP szinten látható, esetlegesen meghibásodó linkek/csomópontok az esetek többségében rövid idő elteltével újra elérhetővé válnak [2] – ezt nevezzük tranziens hibának –, ezzel a hálózat újabb átkonfigurálását kiváltva, holott lehet, hogy még az előző, a meghibásodás által kiváltott átkonfigurálás sem fejeződött be.

A következőkben először a megoldást jelentő *IP alapú gyors hibajavítás* [3] (IPFRR – IP Fast ReRoute) alapvető tulajdonságait ismertetjük, majd a harmadik fejezetben rátérünk az ezen

[1] A szerzőt a Magyar Tudományos Akadémia Bólyai János ösztöndíjjal támogatta.

technikák közül legígéretesebbnek tartott *Not-Via addresses* [4] módszerre. A Budapesti Műszaki és Gazdaságtudományi Egyetemen készített Not-Via tesztrendszerrel szerzett tapasztalatok alapján [5] ismertetjük ezen módszer előnyeit és hibáit, valamint az ezekre adott válaszunkat. A negyedik fejezetben az általunk végzett mérések eredményeit ismertetjük, majd az utolsó részben összefoglaljuk munkánkat.

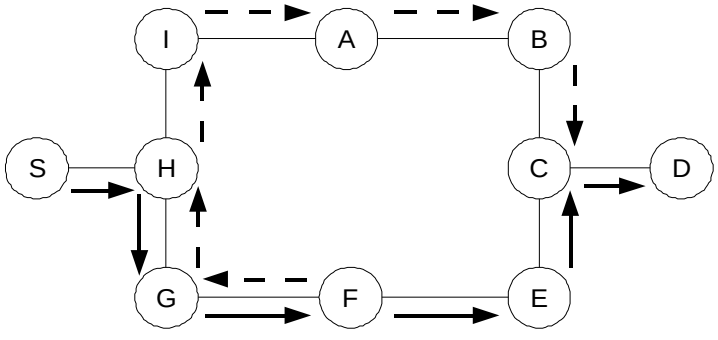
2. A hibajavítás gyorsítása

Az előzőekben áttekintettük a fő okokat, amelyek az IP alapú hálózatok gyors hibajavítását szükségessé teszik. Ebben a fejezetben a gyors hibajavítás megvalósításának alapelveit ismertetjük, melyek megvalósítására példát a következő fejezetben a Not-via addresses-en keresztül mutatunk.

Az IP hálózatának a meghibásodásokra történő lassú válasza az útvonalválasztásért felelős módszerek két alapvető tulajdonságára vezethetőek vissza: a jelenlegi rendszerek a hibákat reaktívan kezelik és rájuk globális választ adnak.

A reaktív hibajavítás során a rendszer csak akkor kezd el foglalkozni a hiba elkerülésének módjával, az útválasztók csak az után kezdik az elkerülő utat kiszámolni, hogy a hiba létrejött. Az ezzel járó késleltetést azonban mindenképpen el kell kerülni, ennek érdekében egy gyors hibajavító eljárás csak proaktív megközelítésű lehet. Proaktív módon persze korlátlan mennyiségű, egyidejű meghibásodásra előre felkészülni lehetetlen, ezért az IPFRR módszerek tipikusan egyszerre csak egy meghibásodó erőforrás esetén működnek, több egyidejű meghibásodás esetén pedig hagyományos megoldásokhoz fordulnak. Ha egy erőforrás kiesése állandónak bizonyul, akkor szintén a hagyományos megoldások kerülnek előtérbe és a hálózat az új topológia felderítése után átkonfigurálja magát, felkészülve egy újabb meghibásodás gyors javítására. Fontos megjegyezni, hogy az IPFRR módszereknek ekkor is nagy hasznuk van, hiszen segítségükkel biztosítható, hogy a szolgáltatás egy pillanatra se essen ki, hogy az átkonfigurálás alatt a csomagok továbbra is zavartalanul továbbíthatóak legyenek.

Globális válasz esetén a teljes hálózat, vagy annak egy jelentős része részt vesz a hiba kijavításában. Mivel azonban ez számottevő kommunikációt jelent, amihez időre van szükség, a globális válasz egy gyors hibajavító módszer számára elfogadhatatlan. Az IPFRR módszerek használata esetén tehát anélkül kell megoldani, hogy a meghibásodott erőforrást a csomagok elkerüljék, hogy a meghibásodás közvetlen szomszédjain kívül más útválasztó ismerné a meghibásodás tényét. Ezt nevezzük lokális válasznak.



1. ábra: Gyors hibajavítás szemléltetése. Az egyszerű nyilak a normál továbbítás, a szaggatottak a gyors hibajavítás irányait jelölik.

Az 1. ábra a fenti elveket szemlélteti egy meghibásodott csomópont elkerülése során. Tételezzük fel, hogy a linkek költségei olyanok, hogy a legrövidebb út *S*-ből *D* csomópontba az *S-H-G-F-E-C-D* útvonal, a csomagok ezen kerülnek továbbításra. Ha az *E* csomópont elérhetetlenné válik *F* csomópont (a meghibásodott erőforrás szomszédja) gyors hibajavításba

kezd és visszaküldi a csomagot G csomópontnak. Mivel a hálózat gyors hibajavítást alkalmaz, így *G* nem továbbítja ismét a csomagot *F*-nek, hanem *H*-nak küldi, aki szintén nem *G*, hanem *I* irányába, a szaggatott nyilak által jelzett úton továbbítja azt. Miután a csomag elkerülte a meghibásodást *C* csomóponttól folytathatja útját az eredeti útvonalon.

Nagyon fontos kiemelni, hogy sem *G*, sem pedig a javító úton található, a hibával nem szomszédos többi csomópont „nem tud" a meghibásodásról, ezen útválasztók állapota nem változik meg. Csupán az *F* és a *C* csomópontokban módosul a csomagtovábbítás – ezek az erőforrás közvetlen szomszédjai. Ha a meghibásodás állandónak bizonyul (bizonyos ideig nem áll helyre a kapcsolat), *F* és *C* elkezdi hirdetni a meghibásodás tényét, ám addig – akár másodperceken át – az összes csomag útja *S-H-G-F-G-H-I-A-B-C-D* lesz.

A fentiek alapján kitűnik, hogy az IP alapú gyors hibajavító módszerek talán legfontosabb kérdése, hogy hogyan lehet ezt a fajta lokális átirányítást megvalósítani tisztán IP hálózat felett. Erre a kérdésre alapvetően kétféle válasz adható: a megoldások az elkerülő úton haladó csomagot vagy explicite megjelölik, vagy implicite valahogyan következtetnek arra, hogy a csomag elkerülő úton van.

A csomag megjelölése többféleképpen történhet. Lehetséges a már meglévő fejlécbe elhelyezni a jelölést, amihez tipikusan a TOS (Type of Service) mező bitjei jöhetnek számításba [8]. A gyakorlatban azonban a bitek az IP fejlécben túl értékesek, így más módszerek elterjedése valószínűbb. Könnyebben megvalósíthatóak azok a csomagokat megjelölő megoldások, amelyek nem a már meglévő bitek közül igyekeznek lefoglalni, hanem tulajdonképpen új biteket „vezetnek be" [4]. Bár először úgy tűnhet, hogy az IP fejléc kiterjesztése nem lehetséges, a valóságban IP-in-IP alagutazással és az ezzel járó új fejléccel ez a feladat megvalósítható. A külső IP fejléc cél címe – az a mező, amit a csomagtovábbításkor az útválasztók amúgy is figyelembe vesznek – az egyik legalkalmasabb hely a meghibásodások jelölésére. A gyakorlatban egy, a többi címtől jól elkülönülő javító címteret definiálnak, és az ebbe eső célcímmel rendelkező csomagokra speciális helyzetüknek megfelelő továbbítási tábla bejegyzések vonatkoznak. Miután pedig a csomag sikeresen elkerülte a meghibásodást, a csomagot az alagútból egy az eredeti továbbítási úton a meghibásodott eszköz utáni útválasztó veszi ki.

Természetesen felmerül, hogy az IP címek napjainkban egyre értékesebbek, egyre nehezebb szabad címhez jutni, így komoly problémának tűnik a javításhoz szükséges elkülönülő címtér biztosítása. Valójában azonban – mivel az IPFRR módszerek tipikusan adott autonóm rendszeren (Autonomous System) belül működnek – erre a célra privát címtartományba eső (192.168.0.0/16, 10.0.0.0/8) címek is felhasználhatóak.

Lehetőség van teljesen elkerülni a csomagok explicit megjelölését [1]. Ezen megoldások lényege, hogy a nem megszokott irányból – interfészen – érkező csomag érkezési irányából is következtetni lehet a meghibásodás helyére, azaz itt a jelölést a bejövő interfész adja. Ezen módszerek azonban bizonyos korlátokkal rendelkeznek; az ilyen megoldást használó rendszerekben többszörös hibák esetén szükségszerűen továbbítási hurkok alakulhatnak ki, ha a csomagok a meghibásodás nélküli hálózatban a legrövidebb utakat követik [7].

3. Not-Via addresses

Az előzőekben ismertetésre kerültek az IP alapú gyors hibajavító módszerek alapvető tulajdonságai. Ebben a fejezetben a Not-Via addresses nevű IPFRR megoldáson keresztül bemutatjuk ezen elvek egy lehetséges gyakorlati megvalósítását. Választásunk azért esett épp erre a módszerre, mivel jelenleg kétségtelenül a legnagyobb ipari valamint IETF támogatás a Not-via addresses mögött áll. A Not-via addressesről részletesebb leírás [4]-ben található.

A Not-Via addresses a lokálisan átirányított csomagokat alagutazás segítségével megjelelő módszerek közé tartozik. Elkerüléskor a speciális címtér egy eleme, mely a külső IP fejléc célcíme, egyszerre jelöli az alagút végpontját és az elkerülendő csomópontot – innen a megoldás neve. Az 1. ábra hálózatában az *F* által átirányított csomag célcímének jelentése „*C* not via *E*”, azaz „juttasd el *C*-be, de *E*-t nem érintve”. Ezen a példán megfigyelhető a Not-Via másik alapvető tulajdonsága: mivel nem lehet eldönteni, hogy a kapcsolat a link vagy a csomópont meghibásodása miatt szűnt meg, a módszer, ha teheti, mindig csomóponthibát feltételez, amivel persze akkor is célt ér, ha csak a link szakadt meg.

Az alagút végpontja Not-Via-ban az úgynevezett next-next hop (NNH), azaz a meghibásodott elemet követő csomópont, hiszen a NNH várhatóan ép, és közelebb van a célhoz, mint az alagút kezdőpontja. Abban az esetben, ha a next-next hop mégsem elérhető, vagy ha az alagútban lévő csomag továbbítása egy újabb meghibásodás miatt valahol nem lehetséges, a Not-Via addresses nem végez ismételt hibajavítást, hanem eldobja a csomagot, így akadályozva meg egy esetleges továbbítási hurok kialakulását.

A hibára való proaktív felkészülés módszere igen egyszerű: az ép hálózaton érvényes útvonalak mellett az egyes csomópontok elhagyásával adódó gráfokon is rendre kiszámítjuk a legrövidebb utakat.

Bár fenti alapelvek igen jól alkalmazhatóak, tesztrendszerünk elkészítése során mégis számos komoly problémával találkoztunk. Ezek közül a legfontosabb, hogy a Not-Via addresses számos extra IP címet követel, melyek menedzselése, terjesztése a hálózatban nem megoldott. Az IP címek nagy száma arra vezethető vissza, hogy ezek a címek nem csak egy cél csomópontot jelölnek, hanem egyben magukban hordozzák azt az információt is, hogy melyik csomópont hibásodott meg. Ez az információ pedig azt jelenti, hogy a szükséges címek száma már pont-pont hálózatok esetén is igen nagyra nőhet. Ha a hálózatban LAN is található, akkor a szükséges hibajavító címek száma a LAN-ban szereplő csomópontok számával négyzetesen skálázódik.

A másik fontos hiányosság, hogy számos legrövidebb út számításra van szükség. Bár bizonyos heurisztikák használatával ezek száma pár tucatra csökkenthető, azonban ez még mindig jelentős sebességcsökkenést jelent a jelenlegi egyetlen legrövidebb út számításhoz képest.

Ezen kívül a gyakorlati implementáció során sajnos kiderült, hogy bár az alapelvek igen egyszerűnek tűnnek, a valóságban számos speciális esetet kell figyelembe venni [6], melyek mind a fejlesztést, mind a már elkészült rendszerben esetleg szükséges hibakeresést jelentősen megnehezítik.

Mivel úgy véljük, hogy ezek a hátrányok jelentős szerepet játszanak abban, hogy a Not-Via addresses és vele együtt az IP alapú gyors hibajavítás elterjedése továbbra is várat magára, a Budapesti Műszaki és Gazdaságtudományi Egyetemen elkészítettük az eredeti megoldás egy módosítását is. Lightweight Not-via [6] algoritmusunk ahelyett, hogy számos legrövidebb út számítást végezne, ún. *maximálisan redundáns fákat* [9] keres a hálózat minden csomópontjához mint célhoz, és meghibásodás esetén ezeket használja. Ezzel a módszerrel egyrészt a számítási komplexitást lecsökkentettük annyira, hogy megoldásunk számítási komplexitását a legrövidebb út számításhoz használt Dijksta-algoritmus dominálja, ám ennél fontosabb, hogy megoldásunknak legfeljebb csak három IP címre van szüksége csomópontonként. Továbbá, bár már csomópontonként három cím is lineáris skálázódást tesz lehetővé, a legtöbb mai IP hálózatban egyáltalán nincs szükség Lightweight Not-via alkalmazása esetén plusz IP címre. Megoldásunk részletes ismertetésére terjedelmi korlátok miatt nincs lehetőség, ez [6]-ban található meg.

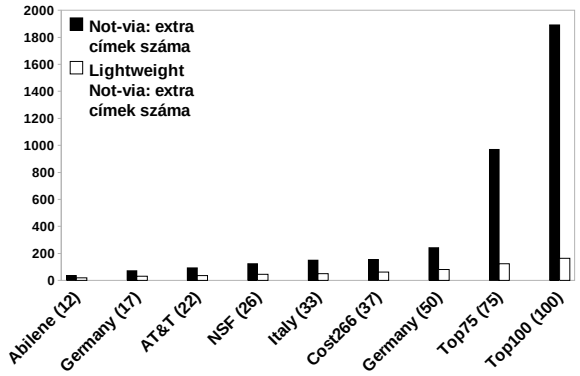
4. Teljesítményvizsgálat

Úgy véljük, hogy az IPFRR módszerek elterjedésének manapság nem technológiai, hanem bizonyos praktikus megfontolások szabnak gátat. Ennek bizonyítására elkészítettünk egy tesztrendszert, amely a Not-via addresses, illetve Lightweight Not-via addresses IPFRR módszer segítségével a gyakorlatban is működő gyors hibajavításra képes.

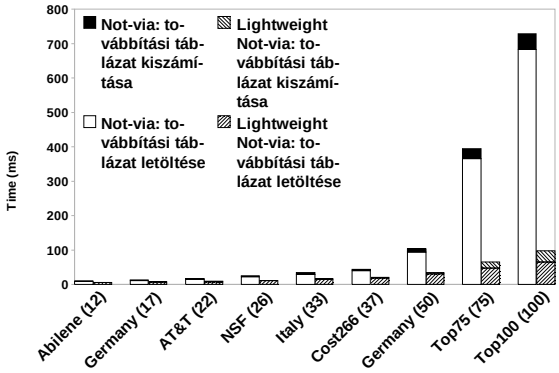
A prototípus rendszerben GNU/Linux-ot futtató PC alapú útválasztókat használtunk. Az útválasztók az IP hálózatokban megszokott Open Shortest Path First (OSPF) protokol segítségével mérik fel a topológiát, a szomszédok között, a kapcsolat esetleges megszakadásának jelzését pedig Bidirectional Forwarding Detection (BFD) valósítja meg. Részletes leírás a tesztrendszer felépítéséről [5]-ben olvasható.

Tapasztalataink megerősítették, hogy az IP alapú gyors hibajavítás valóban igen gyors hibaelkerülésre képes. Tesztrendszerünkben a hibajavítás ideje soha nem volt magasabb, mint 18,5ms, míg hagyományos OSPF alkalmazásakor ez az érték a körülményektől függően 120ms és pár másodperc között változik.

Igen fontos kérdés Not-via használata esetén a címmenedzsment és a szükséges számítási komplexitás. Azért, hogy ezeket vizsgálhassuk teszthálózatunkba mesterségesen néhány jól ismert topológia alapján Link State Advertistment (LSA) csomagokat juttattunk, ezzel szimulálva a kérdéses hálózatot. Hogy a LAN-ok hatását figyelembe vehessük a mérések során feltételeztük, hogy a topológia minden 5. csomópontja egy LAN-t jelképez.



2. ábra: Hibajavításhoz szükséges IP címek száma (zárójelben a topológia csomópontjainak száma)



3. ábra: Az útvonalak kiszámításának és a továbbítási táblák feltöltésének ideje (zárójelben a topológia csomóponjainak száma)

A 2. illetve 3. ábrán az egyes módszerek által nyújtott teljesítmények láthatóak. Megfigyelhető, hogy a Lightweight Not-via mind a szükséges IP címek, mint a komplexitás terén jelentős előnnyel rendelkezik. Észrevehető továbbá, hogy módszerünk a számítási komplexitása a jelenlegi, elkerülő útvonalat nem számító routerek teljesítményéhez mérhető körülbelül 100ms-os nagyságrendű.

5. Összefoglalás

Az IP alapú gyors hibajavítás az egyik utolsó hiányzó technológia ahhoz, hogy az IP protokoll teljes mértékben képessé váljon megfelelni modern korunk kihívásainak. Ebben a cikkben bemutattuk a gyors hibajavítás szükségességének okait, valamint a lehetséges megoldások alapelveit. A gyors hibajavítás elveinek szemléltetését a jelenleg legígéretesebbnek tartott Not-via addresses módszeren keresztül szemléltettük, majd ismertettük a megoldást megvalósító prototípus rendszer tervezésekor, valamint a rendszerrel végzett mérésekkor tapasztalt tulajdonságait. Javasoltunk egy módosított megoldást, amely az eredeti algoritmus számos hiányosságát képes kiküszöbölni és mely, úgy hisszük, képes lehet arra, hogy a hálózati operátorokat az IPFRR módszerek használatára rábírja.

Irodalom

[1] Nelakuditi, S., Lee, S., Yu, Y., and Zhang, Z.-L., “Failure Insensitive Routing for Ensuring Service Availability”, 11th International Workshop on Quality of Service (IWQoS), Monterey, California, USA, June 2003

[2] Innaccone, G., Chuah, C.-N., Mortier, R., Bhattacharyya, S., and Diot, C., “Analysis of Link Failures in an IP Backbone”, ACM SIGCOMM Internet Measurement Workshop 2002, Marseille, France, Nov. 2002

[3] Shand, M., and Bryant, S., “IP Fast Reroute framework”, Internet Draft, online elérhető: http://tools.ietf.org/html/draft-ietf-rtgwg-ipfrr-framework-08, feb. 2008

[4] Bryant, S., Shand, M., and Previdi, S., “IP fast reroute using Notvia addresses”, Internet Draft, online elérhető: http://tools.ietf.org/html/draft-ietf-rtgwg-ipfrr-notvia-addresses-03, Oct. 2008

[5] Szilágyi, P., and Tóth, Z., “Design, implementation and evaluation of an IP Fast ReRoute prototype”, BME, Technical Report, 2008, első díj BME Tudományos Diákköri Konferencia’08, online elérhető: http://opti.tmit.bme.hu/~enyedi/papers/

[6] Enyedi, G., Rétvári, G., Szilágyi, P., and Császár, A., “IP Fast ReRoute: Lightweight Not-Via without Additional Addresses”, elfogadva INFOCOM Mini-Conference’09, Rio de Janeiro, Brazil, April 2009, online elérhető: http://opti.tmit.bme.hu/~enyedi/papers/

[7] Enyedi, G., Rétvári, G., and Cinkler, T., “A Loop-Free Interface-Based Fast Reroute Technique”, 4th EURO-NGI Conference on Next Generation Internet Networks, Kraków, Poland, April 2008

[8] Cicic, T., Hansen, A. F., and Apeland, O. K., “Redundant trees for fast IP recovery”, 4th International Conference on Broadband Communications, Networks, and Systems (Broadnets), Raleigh, North Carolina, USA, Sept. 2007

[9] Enyedi, G., Rétvári, G., and Császár, A., “On finding maximally redundant trees in strictly linear time”, beküldve: IEEE Symposium on Computers and Commmunications (ISCC’09), July 2009, online elérhető: http://opti.tmit.bme.hu/~enyedi/papers/