

Skálázható útválasztás mobil környezetben

BICZÓK GERGELY, ÉGI NORBERT, FODOR PÉTER, KOVÁCS BALÁZS, VIDA ROLLAND

Budapesti Műszaki és Gazdaságtudományi Egyetem, Távközlési és Médiainformatikai Tanszék
{biczok, egi, fodorp, kovacsb, vida}@tmit.bme.hu

Reviewed

Kulcsszavak: hierarchikus irányítás, többszolgáltatós kapcsolat, hálózati struktúrák, proaktív tervezés

A jövő hálózatait az eszközök számának robbanásszerű növekedése, és ezeknek jelentős mobilitása jellemzi majd. Mivel a jelenlegi IP alapú megoldások előreláthatóan nem fognak megfelelni az új hálózati struktúráknak, olyan alternatív címzési és útválasztási megoldások válnak szükségessé, melyek képesek lesznek ezen nagyméretű és mobilitású dinamikus rendszerek hatékony kezelésére. E cikk átfogó képet szeretne nyújtani azokról a már létező algoritmusokról, javaslatokról, melyek hasznos kiindulópontot jelenthetnek egy hasonló követelményrendszerre épülő jövőbeli hálózati architektúra kiépítéséhez.*

1. Bevezetés

A napjainkban elterjedőben lévő elképzelés a jövő hálózatairól, hogy annak minden és mindenki között egy helytől és időtől független távközlési csatornát kell fenntartania. Érzékelhető, hogy egy ilyen jellegű elképzelés megvalósulásához számos különböző eszköz és technológia összehangolt együttműködésére van szükség. A hálózati trendek alapján az egyik legszembevetőbb változás, mely a jövőbeli hálózati architektúrát jellemzi majd, az a résztvevő fix és mobil eszközök arányának eltolódása.

Ma már léteznek technológiák, melyek képesek a számítógépes hálózatok periferiájában megjelenő mobilitást megfelelően kezelni, azonban az imént említett arányok radikális változásához már nem képesek kielégítő és hatékony mobilitási és címzési támogatást nyújtani.

Másfelől, ha a mobilitás nem csak a perifériára lesz jellemző, az útválasztási mechanizmusok újragondolása válik szükségessé. Gondoljunk csak a vezetékes hálózatokban használt IGP (*Interior Gateway Protocol*) és EGP (*Exterior Gateway Protocol*) protokollokra, melyek a különböző összeköttetés-meghibásodásokra is nagyon lassan reagálnak, hosszú konvergenciaidővel. Egy dinamikus hálózatban, melyben a mobil csomópontok más és más, esetleg szintén mobil csomópontokon keresztül érik el a hálózatot, nehezen elképzelhető eme útválasztó protokollok helyes működése.

A másik alapvető tulajdonság, mely meghatározza majd a jövő hálózatait a kommunikálni kívánó csomópontok számának növekedése. Elsősorban itt nem a hagyományos fix telepítésű számítógépekre gondolunk, hanem mobiltelefonokra, laptopokra, PDA-kra, személyazonosító kártyákra, szenzorokra stb. Ez a változás leginkább a skálázhatóság szempontjából kérdőjelezi meg a jelenlegi távközlési megoldásokat.

Ezen tendenciák alapján tehát olyan új, skálázható útválasztási és címzési mechanizmusok kidolgozása válik szükségessé, melyek képesek lesznek folyamatos és konzisztens kapcsolattartást biztosítani különféle technológiákat használó eszközökből álló hálózat számára. Cikkünkben szeretnénk eme területeket megcélzó, a hagyományos IP-alapú címzési és útválasztási architektúra alternatíváját kínáló algoritmusokról egy tömör összefoglalót adni.

2. Egysíkú (flat) útválasztó eljárások

A különböző ad-hoc útválasztási protokollok több szempontból is érdekes kiindulópontot nyújtanak. Bár általában kisméretű hálózatokra voltak tervezve, a skálázhatóság viszonylagos hiányának ellenére hatékonyak a dinamikus felépítésű, nagy mobilitással rendelkező hálózatok kezelésénél. Egysíkú algoritmusok esetén minden útválasztó ismeri az összes lehetséges célcsomópontot. Ez nagy hálózatoknál természetesen nagy útválasztó táblákat eredményez. Az egysíkú eljárások alapvetően két csoportba sorolhatóak: *proaktív* és *reaktív* (kérés alapú) algoritmusok. A proaktív algoritmusok folyamatosan gyűjtik a hálózati topológiára és az útválasztásra vonatkozó információkat, ezáltal többletforgalmat generálnak a hálózaton, míg a kérés alapú eljárások csak akkor keresnek utat, ha a hálózati forgalom azt indokolja.

A legegyszerűbb proaktív eljárás a *Fisheye State Routing (FSR)* [1], amely *összeköttetés-állapot* (*Link State, LS*) algoritmust használ. Az *LS* algoritmusokban az útválasztók topológia-információkat terjesztenek, és minden eszköz ismeri a hálózat teljes topológiáját. Az *FSR* útválasztók a közeli szomszédokról pontosabb, a távolabbiakról kevésbé pontos információkat ismernek. Az *Optimized Link State Routing (OLSR)* [2] megoldás

* A szerkesztő megjegyzése:

A Nemzetközi Távközlési Unió (ITU) is keresi az egységes, minden szolgáltatást elérő címzést, melyet ENUM betűszóval jellemez.

ezzel szemben csökkenti a szórt (broadcast) üzenetek számát, közvetítő pontokat kijelölve a hálózatban. Minden útválasztó ezeknek küldi el LS frissítő üzeneteit, és utána ezek a csomópontok cserélgetik egymás között az információkat. Ez a megoldás csak sűrű hálózatokon hatékony, egyébként hagyományos LS-ként működik.

A *Topology Dissemination Based on Reverse Path Forwarding (TBRPF)* [3] egy feszítő fát készít a topológiából, és a frissítő üzenetek elárasztás helyett ezen a fán, az üzenet forrásával ellentétes irányba terjednek. Az üzenetek vonatkozhatnak teljes vagy részleges topológia információkra. A részleges esetben a csomópontok a forrás fa egy részébe csak a változásokat küldik tovább. Ez a megoldás is leginkább sűrű topológia esetén hatékony.

Ezzel szemben a kérés alapú (reaktív) útválasztó eljárások csak akkor keresnek útvonalat, ha van átvendő adat. Az *Ad-Hoc On Demand Distance Vector Routing (AODV)* [4] algoritmusban például a célcsompont felfedezése a hálózat elárasztásával történik. Amint megvan a cél, a visszairányú, illetve a további forgalom már egyértelmű, mivel a köztes csomópontok megtanulják a továbbküldési irányokat. Az AODV-hez hasonló távolság alapú (*Distance Vector, DV*) eljárások nem ismerik a hálózati topológiát; az útválasztók az egyes csomópontokhoz tartozó távolságaikat és a következő csomópont címét hirdetik.

A kérés alapú megoldás egy másik válfaja forrás útválasztást alkalmaz [5], melyben a küldő a csomag fejlécébe teszi a köztes csomópontok címeit. Ez a megoldás okozza a legkisebb többletforgalmat és a küldő több útvonalat is használhat. Egy hálózati hiba esetén azonban egy új útvonal választása késleltetést okozhat. A kérés alapú algoritmusok nagyméretű hálózatokban is használhatók, bár nagyobb többletforgalmat termelnek, de igazi hátrányuk, hogy a mobilitást csak késleltetve képesek kezelni a folyamatos útvonalkeresés miatt.

3. Hierarchikus útválasztás

Hierarchikus szervezési struktúrákat már a távbeszélő hálózatok építésénél is alkalmaztak [6], az úgynevezett *multipoláris* rendszerek formájában. A multipoláris rendszerek több csillaghálózatból állnak, melyek csúcspontjai szövevényes hálózatot alkotnak. Mindemellett, a távbeszélő hálózatok forgalomirányítási technikáit, a topológián kívül, a használt harántnyalábok típusa is meghatározza. A multipoláris és csillag alaphálózatot haránt kötőélek egészíthetik ki, melyek lerövidíthetik az összeköttetések hosszát. Egy harántnyaláb lehet túlcsondulásos, azaz a nyaláb csatornáinak foglaltsága esetén a hívásokat megkíséreljük más nyalábon felépíteni, avagy lehet veszteséges, azaz a nyaláb foglaltsága esetén a forgalom elvész.

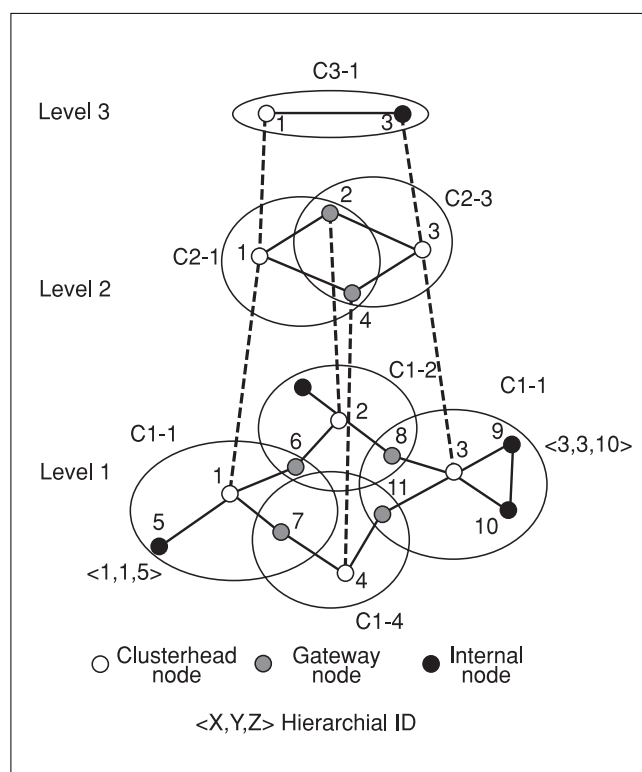
A fentiek alapján a szövevényes hálózatokban tulajdonképpen nincs szükség forgalomirányításra (hisz

minden csúcspont között közvetlen kötőél létezik), míg a veszteséges harántnyalábok esetén direkt-utas rendszert, a túlcsondulásos harántnyaláb esetén pedig alternatív (több utas) forgalomirányítási rendszert lehet alkalmazni. Az alternatív forgalomirányítási technikák a szükséges csatornaszám kiszámításához figyelembe veszik a hálózati forgalom átlag és szórásértékeit, a felkínált forgalom méretét, az érkezési intenzitást, valamint a szabad alternatív utak által felajánlott kapacitást is [7].

A telefonhálózatok építésében használt hierarchikus szervezési megoldásokat fellelhetjük napjaink csomagkapcsolt útválasztási struktúráiban is. Egy többszáz csomópontból álló hálózat esetén az egysíkú megoldások már nem jelentenek skálázható alternatívát. Célszerű tehát egy ilyen hálózatot hierarchikusan felépíteni, a csomópontokat csoportokba sorolni, és a csomópontokhoz különböző funkciókat rendelni. Így elég, ha a csomópontoknak csak a hálózat egy részéről van információjuk. Minden csoportnak van egy vezetője, mely a többi csoporttal tartja a kapcsolatot átjárókon (több csoporthoz csatlakozó csomópontok) keresztül.

A legegyszerűbb ilyen eljárás, a *Clusterhead-Gateway Switch Routing (CGSR)* [8], mely DV algoritmust használ. Minden csomópont tárol egy útválasztási táblát (csoportonként egy bejegyzés) és egy csoport táblát (ki melyik csoporthoz tartozik). Ha valaki üzenetet akar küldeni, a csoport táblában megnézi, hogy melyik csoporthoz tartozik a célcsompont, majd az útválasztási tábla alapján továbbküldi a csomagot. *Hierarchical State Routing (HSR)* [9] esetén a csoportvezetőket is csoportokba szervezik, többszintű hierarchiát kialakítva (1. ábra).

1. ábra Hierarchical State Routing



A csoportvezetőt minden hierarchia szinten több lehetséges szempont szerint (pl. QoS paraméterek) jelölik ki. Minden csomópontnak van egy címe, ez alapján történik az útválasztás. A cím a csomópont egyedi azonosítójából, valamint az egyes hierarchia szinten levő csoportvezetők azonosítóiból képezik.

Csoporton belül és azon kívül lehet különböző útválasztási algoritmusokat is használni, mint teszi azt a *Zone Routing Protocol (ZRP)* [10]. Ez a hibrid eljárás zónán (csoporton) belül proaktív (*Intrazone Routing*), míg zónán kívül kérés alapú (*Interzone Routing*) útválasztást használ.

4. Földrajzi információkon alapuló útválasztási eljárások

Egy csomópont fizikai pozíciójának ismerete megkönnyíti a csomag célcsomópont felé való irányítását. Ehhez természetesen szükséges, hogy az egyes csomópontok képesek legyenek helyzetük meghatározására. *Geographical Addressing and Routing (GeoCast)* [11] esetén a forrás először egy helyi központi csomópont-hoz (GeoNode) továbbítja a csomagot. Ha a cél nincs annak környezetében, akkor a csomag egy magasabb hierarchia szintű csomópont-hoz (GeoRouter) kerül, mely a többi GeoNode környezetében próbálja megkeresni a célcsomópontot. A GeoRouterek hierarchikus szervezése által az útválasztási táblák mérete alaposan lecsökken.

A csomópont utolsó ismert koordinátái alapján egy részleges elárasztást alkalmaz a kérés alapú *Location-Aided Routing (LAR)* [12] eljárás. A cél pozíciója és mozgása, valamint forrás helyzete meghatároz egy területet, amelyen belül a csomópontok elárasztással továbbítják a csomagot (2. ábra). Egy másik megoldás az, hogy egy csomópont akkor árasztja tovább a csomagot, ha önmaga közelebb van a célhoz, mint ahonnan azt kapta.

Egy proaktív megoldás a *Distance Routing Effect Algorithm for Mobility (DREAM)* [13], melyben a csomópontok a linkek helyzetét tartják nyilván. A többletforga-

lom azáltal csökken, hogy a periodikus frissítési üzenetekben a csomópont a helyzetét két szempont szerint terjeszti: minél távolabbi egy csomópont, annál ritkább a frissítés, valamint a gyorsan mozgó csomópontok gyakrabban küldenek frissítéseket. A csomagok részleges árasztása miatt a többszörös kézbesítés redundanciát okoz, így a rendszer a mobilitásra kevésbé érzékeny.

Míg a *DREAM* és *LAR* algoritmusok helymeghatározásra elárasztást használnak, addig a *Greedy Perimeter Stateless Routing (GPSR)* [14] eljárás forrás útválasztást alkalmaz. A csomópont szomszédainak pozíciói periodikusan terjesztődnek minden csomópontban, így a továbbítás mindig a célhoz földrajzilag legközelebbi csomópont felé történik, melyet a szomszédossági gráfból számolunk ki. Mivel a szomszédossági útválasztási táblák kicsik, ezért skálázhatósági problémák itt nincsenek.

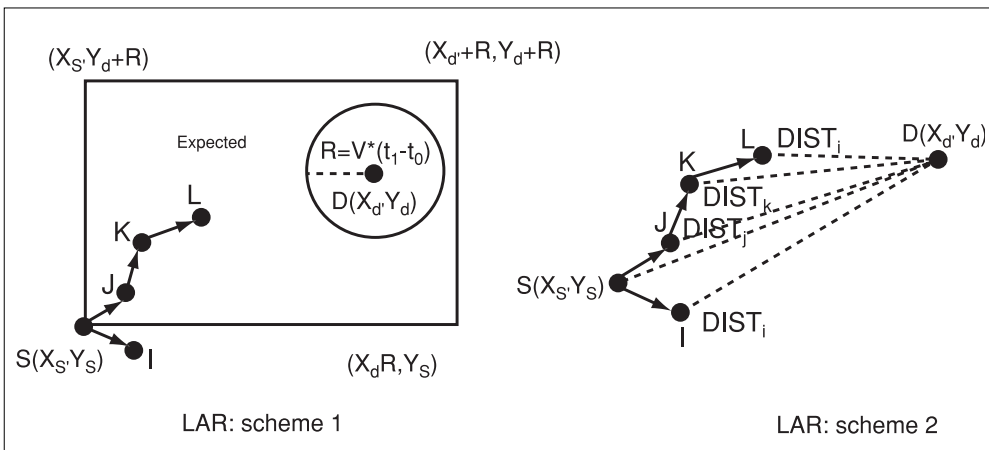
A *Terminode* [15] eljárás proaktív DV-t használ helyi útválasztásra. Az útválasztó táblákban a helyi csomópontok azonosítói és pozíciójuk van tárolva. Egy távoli cél felé a hálózatban található fix pontok mentén továbbítja a csomagokat. Amennyiben fix ponton alapuló utat nem talál a forrás, akkor a csomag a célhoz földrajzilag legközelebbi csomópont felé továbbítódik. Ezen fix pontok, útvonalak mentén történő továbbítás egy jól skálázható megoldást biztosít, mivel a forrás és a cél közti hálózat nagyságától szinte független.

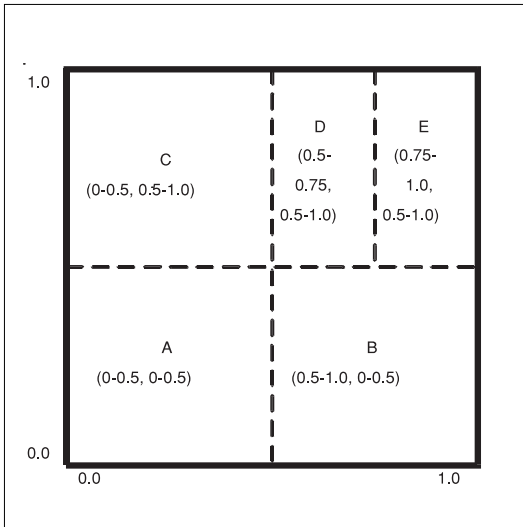
5. Elosztott hash tábla alapú megoldások

Az *elosztott hash tábla (Distributed Hash Table, DHT)* alapú módszerek leginkább az új generációs peer-to-peer rendszerekben terjedtek el, de alkalmazhatóságuk nem korlátozott. A DHT megoldások hasznos kiindulópontnak bizonyultak jónéhány nagyméretű rendszer kifejlesztése során. Számos kutatási eredmény ajánlja használatukat, mint egy lehetséges réteget, amelyre akár több millió csomópontból álló hálózati megoldásokat építhetünk, legyen az egy elosztott fájlrendszer, alkalmazás szintű multicast megoldás, eseményközlő vagy chat szolgáltatás stb.

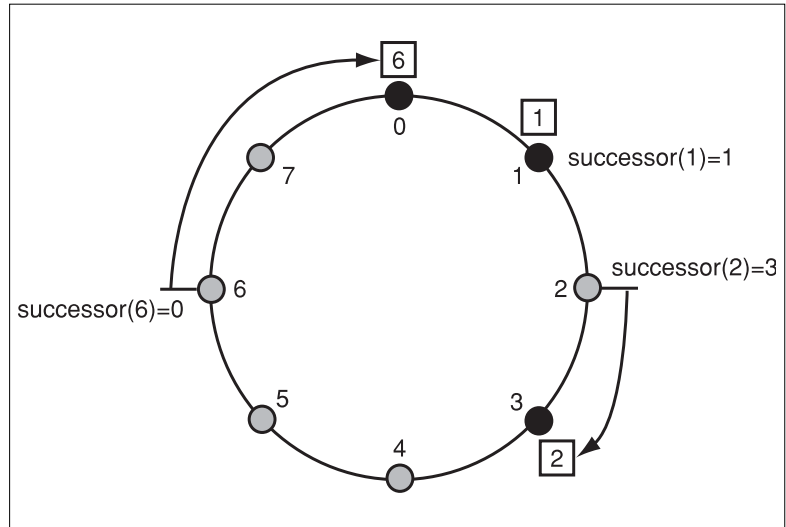
Az alábbiakban bemutatunk néhány olyan ismertebb címzési és útválasztó megoldást, mely elosztott hash tábla használatára épül. Ha egy elosztott fájlrendszer esetét elemezzük, a cél az, hogy megtaláljuk az utat ahhoz a résztvevőhöz, amely a keresett fájl tárolja. A fájlhoz kulcsok vannak rendelve, melyeket például a fájl nevének végrehajtott hash függ-

2. ábra Location-Aided Routing





3. ábra Content Addressable Network



4. ábra Keresés a Chord rendszerben

vénnyel állítunk elő. A hálózat minden csomópontja felelős a kulcsok egy bizonyos tartományának tárolásáért. Ezen rendszerek alapvető művelete a kulcs lekép-zés (*lookup(key)*), amely megadja azon csomópont azonosítóját (például IP címét), amely tárolja a kulcs által meghatározott objektumot. A csomópontok egy hálózati fedőréteget (*overlay*) alkotnak, melyben a szomszédossági viszonyok különböznek a fizikai rétegbeli szomszédosságtól.

A DHT alapú rendszerek magja a címezési struktúrájuk és az ehhez tartozó útválasztó algoritmus. A *tartalom szerint címezhető hálózat (Content Addressable Network, CAN)* [16] egy *d* dimenziós Descartes-féle koordináta rendszert használ a címtér ábrázolására, mely koordináta rendszer minden időpillanatban dinamikusan felosztott a hálózat összes csomópontja közt. Minden csomópont-hoz egy-egy tartomány tartozik (3. ábra).

Ebben a virtuális koordináta rendszerben van tárolva minden (*K*, *V*) kulcs-érték páros, az alábbi módon: a *K* kulcsot leképezve egy egyenletes hash függvénnyel, megkapunk egy *P* pontot a koordináta rendszerben. A kulcs-érték párost az a csomópont fogja tárolni, mely jelenleg felelős a *P* pontot magába foglaló tartományért.

Egy CAN csomópont útválasztó táblája a virtuális koordináta rendszerben szomszédos csomópontok IP címét és a koordináta rendszerbeli címét tartalmazza. Útválasztás közben mindig a cél koordinátához legközelebb fekvő tartomány csomópontjához lépünk tovább.

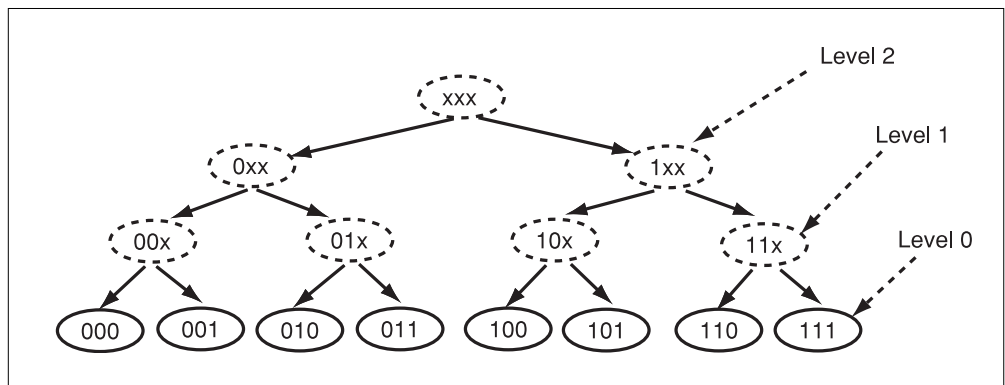
A *Chord* [17] rendszer az érték-azonosító lekép-zés során szintén egy hash függvényt használ, amelynek kimenete egy *m* bit hosszúságú bitsorozat. A hash függvény kimenete nagy valószínűséggel egyenletes eloszlású, és a használt

címtér egy modulo 2^m egyenletes méretű részre osztott kör. Egy *K* kulcs a körön azon csomópont-hoz van rendelve, melynek azonosítója azonos, vagy legelsőként követi a *K* kulcs azonosítóját. Ezt a csomópontot a *K* kulcs *successor* csomópontjának nevezik (4. ábra). A címtér minden csomópontjában mindössze a *successor* címét kell tárolni, így egy lekérdezés során a *successor*-okon körbe haladva, a kört bejárva megtalálhatjuk a keresett értéket.

A *Pastry* [18] rendszerben minden csomópont-hoz egy 128 bites azonosító van rendelve, mely megadja egy szintén kör alakú címtérben a csomópont pozícióját (0 és $2^{128}-1$ között). Itt is feltételezhető, hogy a generált azonosítók egyenletes eloszlásúak. Az útválasztáshoz egy csomópont nyilvántartja a címtérben szomszédos csomópontok címét, a valós hálózati szomszédok címtérbeli címét és egy útválasztó táblát. Az útválasztás ezen adatok felhasználásával történik, az előbbi megoldásoknál dinamikusabban és költségkímélőbben.

A *PeerNet* [19] hálózati megoldásban a címtér egy bináris fa, ahol minden csomópont címe egy-egy levél a fában (5. ábra). A csomagtovábbítás során a küldő csomópontnak mindössze a célcsomópont azonosítóját kell ismernie, mely egy cím lekép-zéssel tudható meg. A csomópontok útválasztó táblája azon csomó-

5. ábra PeerNet hierarchia



pontok címét tartalmazza, amelyek az egyes helyiértékek által azonosított – az érintett csomópont szempontjából a fa ellenkező oldalán lévő – részfák csúcsai. Mivel a címleképzés és a cím kiosztás mechanizmusa ebben az esetben is összefügg, garantált, hogy bármely a hálózatban jelenlévő csomópont elérhető bármely más csomópontból.

Végül megemlítjük röviden a *Tribe* [20] protokollt, mely egy vezeték nélküli önszerveződő hálózatokban használható közvetett útválasztó stratégiának felel meg. A *Tribe* egy egyenrangú elemekből álló elosztott rendszer, ahol a csomópontok egy globálisan egyedi, valamint egy elhelyezkedéstől függő ideiglenes azonosítóval rendelkeznek. A csomópontok által alkotott infrastruktúra leírja az adott csomópont környezete szerinti relatív elhelyezkedését. Az útválasztás egyedi módon történik, a home-agent koncepciót követve, amely teljesen független bármely hálózati szintű útválasztó protokoll által biztosított globális összeköttetéstől. A csomópontok pozíciójából meghatározható azok relatív elhelyezkedése a hálózatban, így nincs szükség semmilyen fix csomópontokkal rendelkező infrastruktúrára, földrajzi helymeghatározásra, vagy távolságmérésre.

6. Valószínűségeen alapuló módszerek

Nagyméretű mobil hálózatoknál fontos, hogy a rendszer hamar észlelje az esetleges topológia változásokat, és képes legyen dinamikusan adaptálni útválasztását ezekhez. Léteznek különböző, valószínűségeen alapuló útválasztó megoldások, melyek lehetővé teszik mindezt. A *Hangya* (Ant) útválasztó algoritmus [21] alapötletét a természettől kölcsönzi: azt utánozza, ahogy a hangyakolonniák megtanulják az élelmiszer-lelőhelyekhez vezető legrövidebb utakat.

Kétféle változat létezik. Az egyik a *szabályos hangya algoritmus* (*Regular Ant Algorithm, RAA*), amely egy legrövidebb utat keres, és csak szimmetrikus linkköltségű hálózatokban használható. A másik az *egyenletes hangya algoritmus* (*Uniform Ant Algorithm, UAA*), mely több alternatív utat kezel és aszimmetrikus linkköltségek esetén is működik.

Minden csomópont (hd) periodikusan generál egy rövid üzenetet (hangyát), melyet elküld egy véletlenszerűen választott másik csomópontnak (hs). A (hd , hs , c) hangyában c a két csomópont közti út költsége (kezdetben $c = 0$). A hs felé vezető úton található összes útválasztó növeli a c mező értékét annak a linknek a költségével, amelyen az üzenetet kapta. A hangyák kis méretűek (<10 bájtt), és ellenirányban (a céltól a forrásig) térképezik fel a hálózatot. Minden r útválasztó minden x célcímhöz egy $(x, (p_1, y_1), \dots, (p_n, y_n))$ bejegyzést tárol, ahol y_i az r útválasztó egy szomszédja, és p_i annak a valószínűsége, hogy a cél cím felé vezető út következő állomása y_i . Az útválasztó táblák tehát valószínűségi alapon működnek, és frissítésük a megerősítéses tanulás módszerével történik.

RAA esetén a p_i értékek különbözőek, de összegük 1. Az UAA megoldásnál minden $p_i = 1/n$, és a hangyák nem tartalmazzák a cél cím (hs) mezőt, így szükség van egy TTL (time-to-live) mezőre, mely garantálja az üzenetek véges érvényességét. RAA esetén egy idő után beáll egy állandósult állapot, amikor a továbbítási valószínűségek 0-hoz vagy 1-hez konvergálnak (a következő linkek kapnak 1-hez tartó valószínűséget). Az UAA eljárás viszont alkalmazkodni tud a hálózatban bekövetkezett változásokhoz, így a két módszert vegyítve egy flexibilis és ugyanakkor hatékony útválasztási algoritmust kapunk.

Valószínűségeen alapuló útválasztást akkor is lehet alkalmazni, ha nem csak egy adott célállomás, hanem a hálózat egy tartományában lévő összes állomás felé szeretnénk továbbítani egy adott csomagot. A hagyományos elárasztás alapú megoldások hátránya a generált forgalom mérete, hiszen a legtöbb eljárásban egy állomás többször is megkapja ugyanazt a csomagot más és más szomszédjától. Ezzel szemben a *pletyka alapú útválasztás* (*Gossip Based Routing*) [22] valószínűségeen alapuló elárasztást használ, ezzel csökkentve a routing forgalmat.

Az alapötlet egyszerű: egy csomópont bizonyos valószínűséggel szór tovább egy kapott üzenetet. Négy pletyka algoritmus ismeretes, ezek egymás továbbfejlesztett változatai. A $GOSSIP_1(p)$ az alap algoritmus, mely p valószínűséggel elárasztást alkalmaz, $(1-p)$ -vel pedig eldobja a csomagot. Ezzel szemben a $GOSSIP_1(p,k)$ a forrástól számított k linken keresztül $GOSSIP_1(1)$ -ként viselkedik, megelőzve az üzenet korai kihalását. Kihalás akkor történhet, ha egy csomópontnak nagyon kevés szomszédja van. Ezért hasznos a $GOSSIP_2(p_1, k, p_2, n)$ változat: ha egy csomópont fokszáma kisebb mint n , akkor a p_2 ($>p_1$) valószínűség lép érvénybe. A $GOSSIP_3(p,k,m)$ szerint, ha egy csomópont eredetileg nem szórt tovább egy kapott üzenetet, de adott időn belül nem kapta meg ugyanazt az üzenetet legalább m másik csomóponttól, akkor azonnal tovább szórja azt. A negyedik megoldás zónák bevezetésével ér el további javulást.

7. Összefoglalás

Ez a cikk rövid áttekintést nyújt a ma létező, alternatív címzési és útválasztási javaslatokról. Az itt bemutatott eljárások önmagukban nem képesek megoldani napjaink útválasztási kérdéseit, azonban olyan ötleteket tartalmaznak, melyek felhasználhatók egy új architektúra kidolgozásához. A hierarchikus megoldások a csomópontok struktúrába rendezésével, a földrajzi adatokra épülő protokollok a számítógépek tényleges tartózkodási helyének figyelembevételével, az elosztott hash táblás algoritmusok új címzési és továbbítási ötleteikkel, míg a valószínűségi alapokon nyugvó mechanizmusok hasznos redundancia-képző tulajdonságaikkal járulhatnak hozzá egy valóban skálázható és hatékony útválasztási megoldás megszületéséhez.

Irodalom

- [1] T.-W. Chen,
„Fisheye State Routing:
A Routing Scheme for Ad Hoc Wireless Networks”,
Proc. of ICC 2000,
New Orleans, LA, June 2000.
- [2] T. Clausen, P. Jacquet,
„Optimized Link State Routing Protocol”,
RFC 3626, Oct. 2003.
- [3] R. G. Ogier, F. L. Templin, and M. G. Lewis,
„Topology Dissemination based on
Reverse-Path Forwarding (TBRPF)”,
RFC 3684, Feb. 2004.
- [4] C. E. Perkins and E. M. Royer,
„Ad-Hoc On-Demand Distance Vector Routing”,
Proc. of IEEE WMCSA '99,
New Orleans, LA, Feb. 1999, pp.90–100.
- [5] D. B. Johnson and D. A. Maltz,
„Dynamic Source Routing in
Ad Hoc Wireless Networks”,
Mobile Computing, T. Imielinski and H. Korth,
Eds., Ch. 5, Kluwer, 1996, pp.153–181.
- [6] Dely Z., Ecsedi Gné., Huszty G., Madarász E.,
Oprics Gy., Plank Gy., Sallai Gy.,
„Távközlő hálózatok forgalmi tervezése”,
PKI, Budapest, 1980.
- [7] Wilkinson, R.I.,
„Theories for Toll Traffic Engineering in the USA”,
Bell System Technical Journal,
Vol. 35, no.2, pp.421–514., March 1956.
- [8] C.-C. Chiang and M. Gerla,
„Routing and Multicast in Multihop,
Mobile Wireless Networks”,
Proc. of IEEE ICUPC '97,
San Diego, CA, Oct. 1997.
- [9] G. Pei, M. Gerla, X. Hong and C.-C. Chiang,
„A Wireless Hierarchical Routing Protocol with
Group Mobility”,
Proc. of IEEE WCNC '99,
New Orleans, LA, Sept. 1999.
- [10] Z. J. Haas and M. R. Pearlman,
„The Performance of Query Control Schemes for the
Zone Routing Protocol”,
ACM/IEEE Trans. Net.,
Vol. 9, no.4, Aug. 2001, pp.427.
- [11] J. C. Navas and T. Imielinski,
„Geographic Addressing and Routing”,
Proc. of 3rd ACM/IEEE Int'l. Conf. Mobile Comp. Net.,
Budapest, Sept. 26-30, 1997.
- [12] Y.-B. Ko and N. H. Vaidya,
„Location-aided Routing (LAR) in
Mobile Ad Hoc Networks”,
ACM/IEEE Int'l. Conf. Mobile Comp. Net.,
1998, pp.66–75.
- [13] S. Basagni, I. Chlamtac,
V.R. Syrotiuk and B.A. Woodward,
„A Distance Routing Effect Algorithm
for Mobility (DREAM)”,
Proc. of ACM/IEEE Int'l. Conf. Mobile Comp. Net.,
1998, pp.76–84.
- [14] B. Karp and H. T. Kung,
„GPSR: Greedy Perimeter Stateless Routing for
Wireless Networks”,
Proc. of Mobicom 2000,
Boston, MA, USA, 2000, pp.243–254.
- [15] L. Blazevic, S. Giordano and J.-Y. Le Boudec,
„Self Organized Terminode Routing”,
Technical Report, DSC/2001/024,
Swiss Federal Institute of Technology, Lausanne.
- [16] S. Ratnasamy et al.,
„A Scalable Content-Addressable Network”,
Proc. of ACM SIGCOMM,
San Diego, CA, Aug. 2001, pp.161–172.
- [17] I. Stoica et al.,
„Chord: A Scalable Peer-to-Peer Lookup Service for
Internet Applications”,
Proc. of ACM SIGCOMM,
San Diego, CA, Aug. 2001.
- [18] P. Druschel, A. Rowston,
„Pastry: Scalable, Distributed Object Location and
Routing for Large-scale Peer-to-Peer Systems”,
Proc. of Middleware 2001,
18th IFIP/ACM International Conference on
Distributed Systems Platforms, Nov. 2001.
- [19] J. Eriksson, M. Faloutsos, S. Krishnamurthy,
„PeerNet: Pushing Peer-to-Peer Down the Stack”,
Proc of IPTPS'03,
2nd International Workshop on
Peer-to-Peer Systems,
Berkeley, CA, USA, Feb. 2003.
- [20] A. Viana, M. Dias de Amorim, S. Fdida,
and J.F. de Rezende,
„Indirect Routing Using Distributed
Location Information”, in Proc of PerCom'03,
IEEE International Conference on Pervasive
Computing and Communications,
Fort Worth, TX, USA, March 2003.
- [21] D. Subramanian, P. Druschel, J. Chen,
„Ants and Reinforcement Learning:
A Case Study in Routing in Dynamic Networks”,
Proc. of IJCAI-97, Nagoya, Japan, Aug. 1997.
- [22] Z.J. Haas, J.Y. Halpern, and L. Li,
„Gossip-based Ad Hoc Routing”,
Proc. of INFOCOM'02,
New York, USA, June 2002, pp.1707-1716.
- [23] Gódor Balázs,
„Technológia-független, univerzális azonosítók”,
PKI Közlemények,
2004/48. szám, pp.127–140.