

Csomagkapcsolt hálózatok kommunikációs protokolljainak optimalizálása neurális algoritmusokkal



Karlócai Balázs

Pázmány Péter Katolikus Egyetem
Információs Technológiai Kar

Multidiszciplináris Műszaki Tudományok Doktori Iskola

Ph.D. disszertáció

2013. tavasz

Témavezető: Dr. Levendovszky János MTA Dr.

Szüleimnek, drága feleségemnek, gyermekeimnek...

Köszönet

Elsőként szeretném megköszönni témavezetőmnek, Dr. Levendovszky János professzor úrnak a sok segítséget, támogatást, tudományos szemléletmódot, amelyet folyamatosan kaptam tőle. Szeretném megköszönni Dr. Roska Tamásnak a folyamatos emberközeli figyelmét, amivel doktori munkámat kísérte. Köszönettel tartozom továbbá Nyékyné Dr. Gaizler Judit dékánasszonynak, aki emberileg a legtöbbet támogató tanulóim során. Köszönetet mondok továbbá dr. Oláh Andrásnak és Bojársky Andrásnak is, akik a mindennapok kérdéseiben segítséget nyújtottak, valamint Balogh Ádámnak, Treplán Gergelynek, Tornai Kálmánnak, akik tanulmányaimat, kutatásaimat végigkövették és segítettek. Szintén köszönöm Karlócai Miklósnak, aki az egész disszertációt átolvasva hasznos tanácsokkal, javaslatokkal látott el. Külön köszönetet mondok feleségemnek, Orsinak, aki mindvégig támogató és segített, ahol tudott. Köszönetet mondok édesapámnak és édesanyámnak szeretetükért és támogatásukért, valamint köszönet illeti további családtagjaimat, barátaimat, ismerőseimet.

Összefoglaló

A jelenlegi kommunikációs hálózatokban egyre növekvő igény van újabb (az eddiginél jóval komplexebb) szolgáltatásokra. Ez az igény egyrészt a fizikai rétegbe integrálható új technológiákkal (kiloprocesszoros rendszerek), másrészt párhuzamos architektúrákkal szolgálható ki. Azonban vannak olyan kihívások, amelyeknél a jelenleg használatos számítási platformok alkalmazása és a technológiai fejlődés eredményei mellett is világosan megjelennek a fizikai korlátok, amelyeket csak algoritmusokkal lehet áthidalni. Ez főleg a nagysebességű hálózati kommunikációban (lásd router-technológia), illetve a vezeték nélküli kommunikációban jelenik meg (átviteli közeg és energia-korlátok). Ezért a disszertáció fő célkitűzése az ezen kérdésekre adandó válasz, pontosabban új algoritmusok kifejlesztése:

- hatékony csomagklasszifikáció routerekben
- előírt quality of service és megbízhatóság elérése energiában korlátozott vezeték nélküli szenzoriális hálózatok esetén
- hálózatdimenzionálás nagy mennyiségű forgalmi folyamatok megbízható továbbítására.

A dolgozat ezekre a kérdésekre összpontosít, a kidolgozott új algoritmusok formális leírása mellett részletes teljesítőképesség-analízissel bizonyítva hatékonyságukat.

Tartalomjegyzék

Tartalomjegyzék	4
Ábrák jegyzéke	7
1. Bevezetés	11
2. Tézisek fellelhetősége	13
3. Csomagklasszifikáció IP-hálózatokban	14
3.1. IPv6 és a csomagklasszifikáció	14
3.1.1. Csomagklasszifikáció a gyakorlatban	15
3.1.2. Class-based Internet Addressing	16
3.1.3. Classless Internet-domain Routing	16
3.2. Routerek	18
3.2.1. A routerek feladatai, funkciói	18
3.2.2. A routerek felépítése	18
3.2.3. A routerek képességei	20
3.3. A csomagosztályozás formális modellje – packet classification táblák	22
3.3.1. A csomagosztályozás feladata	22
3.3.2. Szabálytáblák	25
3.4. Tradicionális PC-algoritmusok rövid összefoglalása	26
3.4.1. Kimerítő keresés	26
3.4.2. Lineáris keresés	27
3.4.3. A címek gyorsítótárba másolása	28
3.4.4. Radix Trie	29
3.4.5. Tree Bitmap	31
3.4.6. Area based quad tree	32
3.4.7. Klasszikus módszerek teljesítmőképesség-analízise	36
3.5. A csomagklasszifikáció mint képfeldolgozási feladat – csomagklasszifikáció Cellular Neural Network (CNN) segítségével	37
3.5.1. A CNN bemutatása	37
3.5.2. Új csomagklasszifikációs megoldás CNN-architektúrával	38

3.5.3.	Nagyobb szabályrendszerek esetén a szabályok diszjunkt területté való leképzése	42
3.5.4.	Megvalósítás	43
3.6.	Numerikus eredmények	44
3.6.1.	A CNN tranzienst ideje és az igényelt adatátviteli sebesség közötti kapcsolat	47
4.	Vezetéknélküli szenzorhálózatok	50
4.1.	Megbízható csomagtovábbítás vezetéknélküli hálózatokban	50
4.1.1.	Vezetéknélküli szenzorhálózat	51
4.1.2.	Szenzorhálózat alkalmazása	55
4.1.3.	Vezetéknélküli szenzorhálózat szabványa	56
4.1.4.	Szenzorok energiafelhasználása	57
4.1.5.	Jelenlegi energiatudatos protokollok bemutatása	58
4.2.	Modell és probléma felvetés	60
4.3.	Új algoritmusok	61
4.3.1.	Minimális energiát garantáló algoritmus	62
4.3.2.	Módosított algoritmus	63
4.3.3.	Kevés energiájú csomópontok kizárása	64
4.4.	Teljesítőképesség-analízis	64
4.5.	Az elért eredmények jelentősége	67
5.	Hálózatdimenzionálás csomagkapcsolt hálózatokban	68
5.1.	Hálózat-dimenzionálás csomagkapcsolt hálózatokban	68
5.2.	QoS a csomagkapcsolt hálózatokban	69
5.2.1.	QoS teljesítőképességi mértékei	70
5.3.	A hálózat-dimenzionálás fontossága	73
5.4.	A modell	75
5.5.	Új dimenzionáló algoritmusok bemutatása	79
5.6.	Új többnode-os dimenzionálási algoritmus	81
5.7.	Numerikus eredmények	83
6.	Konklúziók	87
6.1.	Csomagklasszifikáció CNN-alapú technológiával	87

TARTALOMJEGYZÉK

6.2. Vezetéknélküli csomagtovábbítás útvonalválasztása	88
6.3. Hálózatdimenzionálás csomagkapcsolt hálózatokban	88
6.4. Összefoglalás	89
A szerző publikációi	90
Hivatkozások	91

Ábrák jegyzéke

1.	4 byte-os IP-cím felépítése	15
2.	Router típusok: Core és Edge routerek: A Core routerek az Internet gerinchálózataért felelősek, míg az Edge routerek a végfelhasználókat kötik az Internetre. Forrás: http://www.lightwaveonline.com/articles/print/volume-18/issue-5/special-report	19
3.	Egy router architektúrája	20
4.	A PC-táblák működése	23
5.	Az osztályozó geometriai megjelenítése	24
6.	Egy egyszerű többinterfészes számítógép routing architektúrája. Forrás: http://www.nat32.com/image5.jpg	25
7.	A kimerítő keresés működési diagrammja	27
8.	A lineáris keresés működési diagrammja	28
9.	Radix trie működése	29
10.	Radix trie működési diagrammja	30
11.	Area based quad tree felépítése. [1]	32
12.	Area based quad tree dekompozíciója [2]	33
13.	A filter és a negyed viszonya az AQT-ben, fehérrel ábrázolva az állapottér, szürkével a filter [2]	34
14.	AQT működési diagrammja	35
15.	Egy AQT-alapú GPU-megoldás rendszerterve [3]	36
16.	A kapcsolati mintázata a CNN-nek $r = 1$ esetén	37
17.	Egy CNN-ben használható nemlinearitás	38
18.	A hullámterjedés egy egyszerű szabályhalmazon ($\tau = 0,3,6,12,20$)	39
19.	A referenciapontok elhelyezése a CNN területén	41
20.	Az architektúra logikai felépítése	43
21.	A CNN alapú csomagklasszifikáció megvalósításának rendszerdiagramja	44
22.	Különböző szabályszerű teljesítőképesség (klasszifikációs idő - ms) azonos csomagforrás-paraméterek mellett. Látható, hogy a CNN-es megoldás, több szabály esetén a kis területek miatt hamarabb tüzel	45

ÁBRÁK JEGYZÉKE

23.	A klasszifikációs idő (ms) a csomagforrás hosszát változtatva (leg-hosszabb prefix - W) azonos szabálysám és csomagszám mellett . . .	46
24.	A klasszifikációs idő (ms) az egyes algoritmusokkal az Ohio-i egyetem szerveréről szedett csomagfolyammal	47
25.	Processzorok órajelének fejlődése az elmúlt 40 év során. Forrás: http://smoothspan.wordpress.com/2007/09/06/a-picture-of-the-multicore-crisis/	48
26.	A CPU és a GPU fejlődésének összehasonlítása elvégzett műveletek alapján (GFLOP) Forrás: Paul E. McKenney 2011-ben kiadott "Is Parallel Programming Hard, And, If So, What Can You Do About It?" .	49
27.	Egy szenzornode sematikus felépítése Forrás: https://www.assembla.com/spaces/32093-project	52
28.	A Mica2 szenzornode felépítése [4]	53
29.	Közvetlen (single-hop) kommunikáció	54
30.	Közvetett (multi-hop) kommunikáció	54
31.	Egy ténylegesen használt szenzor rendszer felépítése. Forrás: http://www.altenergymag.com/emagazine.php?issue_number=08.12.01&article=links	56
32.	Az IEEE 802.15.4 szabvány által leírt készülék architektúrája [5] . .	57
33.	Vezetéknélküli szenzornálzat energiahatékonyságát javító módszerek	59
34.	A továbbított csomagok száma (x1000) a hálózat lemerüléséig a különböző algoritmusok használatával	65
35.	A lemerülést követően a hálózatban maradt átlagenergia	65
36.	Az átlag energiaértékek hálózatonként	66
37.	IP hálózati forgalom továbbítása ATM Cloudon keresztül [6]	70
38.	Késleltetés megoszlása egy 1500byteos csomag transzatlanti továbbítása esetén, különböző sávszélességekkel [6, 13. o]	73
39.	A hálózati eszközök energiafogyasztása a sávszélesség függvényében.	74
40.	A méretező-algoritmus kimeneteként kapott struktúra	76
41.	A többnode-os optimaizációs algoritmus folyamatábrája	83
42.	Átlagkapacitások a különböző forgalomkonfigurációkhoz	85
43.	Kiegyensúlyozott terheléses algoritmus segítségével készített optimális hálózati topológia	86

Rövidítések jegyzéke

AAA Authentication, Authorization and Accounting

AQT Area based quadtree

AS Admission Set

CFS Crossing Filter Set

CIDR Classless Internet-domain Routing

CLP Cell Loss Probability

CNN Cellular Neural Network

DMZ Demilitarized Zone

DRAM Dynamic RAM

DSP Digital Signal Processor

HAA Hierarchical Admission Architect

HiCuts Hierarchical Intelligent Cuttings

IP Internet Protokoll

IPCP IP Capacity Planning

IPv6 IP version 6

LEACH Low Energy Adaptive Clustering Hierarchy

NAC Network Admission Control

NCAP Node and Capacity Arrangement Problem

PAMAS Power Aware Multiaccess with Signaling

PC Packet Classification

QoS Quality of Service

ÁBRÁK JEGYZÉKE

RCF Recursive Flow Classification

SDN Self-Defending Network

TCP Transmission Control Protocol

VoIP Voice over IP

WBAN Wireless Body Area Network

WSN Wireless Sensor Network

1. Bevezetés

Napjainkban a nagy adatátviteli sebesség igénye, ugyanakkor a limitált erőforrások újabb és újabb kihívások elé állítják a kommunikációs technológiákat. Az elmúlt tíz év hálózati fejlődését vizsgálva az tapasztalható, hogy a legnagyobb kihívás a minőségi kommunikáció, a Quality of Service (QoS) biztosítása véges erőforrások (pl. energia, processzási képesség, sávszélesség) mellett. Ez általánosan a hálózat tervezését és optimális működtetését, mint kényeseres optimalizálási feladatot veti fel: pl. vezeték nélküli szenzorhálózatok esetén keressük azt az előírt minőségű útvonalat, amely minimális energiafogyasztást igényel a csomag továbbadásában résztvevő node-októl. Hasonló feladatok megfogalmazhatóak mind az Internet Protokoll (IP), mind a vezeték nélküli szenzorhálózatok (Wireless Sensor Network - WSN)[7], mind a testközeli vezeték nélküli hálózatok (Wireless Body Area Network - WBAN)[8, 9] terén. Ezek közül a fő problémák a következők:

- Az elmúlt évek alatt a multimédiás kommunikáció (VoIP, Video streaming) nagy teret hódított a teljes internetes forgalomban, azonban ezek a szolgáltatások előírt minőséget igényelnek. Ez hangsúlyozza a routereken működő csomagosztályozás feladatának fontosságát, mert különböző szolgáltatási minőséghez tartozó csomagokat különböző módokon kell kiszolgálni, és a kiszolgáláshoz tartozó akciókat real-time módon kell elvégezni. Így az IP esetében nagyon fontos kérdéskörre vált a csomagosztályozás (packet classification - PC) [10], mely alapja többek között a QoS szolgáltatásnak is [11]. Tekintettel arra, hogy az Internet központi routereiben kiemelkedően fontos a csomagosztályozást elvégezni, ez könnyen szűk keresztmetszetet képezhet a kommunikációban. A csomagok prioritizálására az IPv6 (Internet Protocol version 6) már specifikációjában is lehetőséget biztosít, itt a címinformáció mellett a csomagfejlécben QoS-jellegű, valamint a származási helyre jellemző adatok találhatóak. Így lehetővé válik fire-wall és QoS alkalmazások megvalósítása is. Ez a probléma arra egyszerűsíthető, hogy hogyan lehet a csomagok fejlécben felismert információ alapján gyors akciót vagy osztályba sorolást elvégezni. Ez algoritmikusan is mélyebb feladatokhoz vezet, amelyek tradicionálisan az ún. "computational geometry" tárgykörébe esnek (lásd [12]).

- A vezeték nélküli szenzorhálózatok esetében kiemelkedően fontos az energiafelhasználás optimalizálása. Ismert az, hogy a Rayleigh fading-modell [13] alapján milyen valószínűséggel történik a csomag sikeres vétele adott energiájú küldés és adott távolság esetén. Mivel adott megbízhatóságú direkt (egy hop-os) csomagtovábbítás a bázisállomástól távolra nagy energiákat igényelne [14], ezért Multi-Hop kommunikációs modellt kell használnunk [15], amelyben a küldő node és a bázisállomás között több "relay" (közvetítő) node helyezkedik el. Itt viszont felmerül a kérdés, hogy hogyan lehet minimalizálni azokat a küldési energiákat a Multi-Hop láncon belül, amelyek az aztán adott megbízhatóságot (a csomagnak a bázisállomásra történő adott valószínűségű megérkezését) eredményezik. A minimális energia egyértelműen az élettartam meghosszabbítását jelenti [16, 17].
- A nagy sávszélességű, különböző QoS paraméterű hozzáférési hálózatok megjelenésével ezeknek a hálózatoknak a kiszolgálását végző eszközök, node-ok új kihívásokat támasztanak a méretezés és teljesítőképesség terén [18, 19]. Ezeknek az rendszereknek közös tulajdonsága, hogy fa-szerű topológiába szerveződnek, ahol a belső node-ok további kapacitási korlátokkal rendelkeznek, valamint hogy a shared-bus architektúrának köszönhetően a rendelkezésre álló kapacitás az aktuális feltöltési és letöltési forgalmon múlik [20]. Ezért a legfontosabb cél egy olyan új, többnode-os méretező módszer kifejlesztése, amely képes kezelni további kapacitási korlátokat, valamint a kétirányú forgalom hatását. Ezen módszer segítségével képesek vagyunk megállapítani, hogy ténylegesen hány node-ra és milyen link kapacitásokra van szükség ahhoz, hogy egy adott számú felhasználót megadott cellavesztési valószínűség mellett kiszolgáljunk.

A fenti, látszólag különálló csomagkapcsolt hálózati tématerületeket a kombinatorikus optimalizálás igénye kapcsolja össze.

2. Tézisek fellelhetősége

Ebben a fejezetben az olvasás megkönnyítése érdekében a tézisekhez kapcsolódó fejezet- ill. alfejezetcímeket sorolom fel.

Tézisszám	Tézis leírása	Fejezet
Tézis 1.1	Bizonyítottam, hogy a csomagklasszifikáció CNN-architektúrával megoldható	3.5.2
Tézis 2.1	Megmutattam, hogy minimális energiájú, adott megbízhatóságú algoritmus megoldható Bellmann-Ford-algoritmussal, így a feladat polinomiális időben kivitelezhető	4.3.1
Tézis 2.2	A hálózati topológiát figyelembe véve egy módosított algoritmust adtam a teljesítőképesség további javítására	4.3.2
Tézis 3.1	Az optimális HAA eléréséhez kifejlesztettem egy egynode-os optimalizációs algoritmust	5.5
Tézis 3.2	Az optimális HAA eléréséhez kifejlesztettem egy többnode-os optimalizációs algoritmust	5.6

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

3. Csomagklasszifikáció IP-hálózatokban

Ahhoz, hogy a felhasználók csomagjai a lehető leggyorsabban és legkisebb hibával célba érjenek, a routereken a csomagok hibamentes feldolgozására és továbbításra alkalmas algoritmusokat kell telepíteni.

Ezek az algoritmusok a csomagok fejlécének analízise alapján eldöntik, hogy a router melyik output portjára kell továbbítani (address lookup), illetve hogy milyen egyéb processzálás szükséges (pl. csomagok szűrése tűzfal alkalmazása esetén, vagy prioritásos csomagkezelés adott QoS kritériumoknak eleget téve)[11].

3.1. IPv6 és a csomagklasszifikáció

Napjainkban a routerek működésének nagy része pusztán egy továbbító funkció, melynek irányát a beérkező adatcsomag fejléce határozza meg, ugyanakkor az IPv6-ban ennél jóval bonyolultabb csomagosztályozásra is szükség van. A feladat algoritmikus kihívása nemcsak a megvalósítandó csomagosztályozási feladatok sokrétűségében, hanem ennek a sebességében is rejlik, hiszen szélessávú adatfolyamokon kell végrehajtani.

Az elmúlt években az Internethez hozzáférő felhasználók száma, valamint az ezen felhasználók által igénybevett sávszélesség drámai mértékben megnőtt [21]. A küldött és fogadott adatok mennyisége és minősége számtalan szolgáltatás biztosításának lehetőségét igényli a routerektől. Hiszen nem mindegy, hogy egy videókonferencia adatait, egy mp3-letöltést, vagy egy email-továbbítást milyen sorrendben és prioritással kezel a router; a végrehajtás sorrendje eltérő lehet a beérkezett igények sorrendjétől, holott a router feladata elsődlegesen annyi, hogy egyre közelebb és közelebb kerüljön a csomag a címzetthez.

Ahhoz, hogy a routerek el tudják dönteni egy csomagról vagy beérkezett igényről, hogy milyen speciális módon kell kezelni, szükség van a célállomás címén kívül többletinformációra. Ez jelentősen lelassíthatja a továbbítást, hiszen nagyobb mennyiségű adatot kell a routernek feldolgoznia. Egy másik nehézség abból fakad, hogy a felmérések szerint az Interneten áthaladó csomagok 75%-ának mérete kisebb, mint egy átlagos TCP- (Transmission Control Protocol)- csomag (522 byte), és döntő részük csupán 40-50 byte nagyságú[22]. Tekintve, hogy a használt sávszélesség több gigabi-

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

tes léptékűre nőtt, és a router által használt memória elérési ideje is szab egy időkeretet, jól látható, hogy a routerben dolgozó algoritmusnak szűk időrés alatt kell precíz döntést hoznia az áthaladó csomag jövőjét illetően. Új megoldás szükségességét a következő példával illusztrálnám: vegyünk IP-csomagokat 40 byte méretben, a router portjait vegyük 10Gbit/s sebességűnek, és számoljunk 10 porttal. Ezzel a felállással a legrosszabb esetben 3.2 nsec alatt kell döntést hoznunk a csomag jövőjét illetően. A DRAM átlagos elérési sebessége is ebben a tartományban található, 3-5 nsec. Ez jelzi, hogy szükséges gyors csomagklasszifikációs algoritmusokat használni a megfelelő átvitel biztosításának érdekében, mivel gyorsabb hardverre egyre kevésbé lehet számítani.

3.1.1. Csomagklasszifikáció a gyakorlatban

Az Interneten haladó csomagok fejléce napjainkban egy 32 bites, vagyis 4 byte-os címet tartalmaz, mely négy, egyenlően felosztott részre bomlik, az első kettő a netid, a második kettő a hostid. A netid mutatja meg a célhálózatot, a hostid ezen belül megjelöl egy specifikus hostot. Ez az IP version 4 (IPv4) címzési architektúra, melyben jól láthatóan 256 a negyediken darab különböző cím van.

Netid	Hostid
255.255.255.255	

1. ábra. 4 byte-os IP-cím felépítése

Ez a szám az IP protokoll tervezésekor bőven elégnek tűnt, de ma már láthatóan kevés, ezért folyamatos bevezetés alatt áll az IPv6 [23, 24]. Ez 128 bites címmezőt jelent, ami minden szárazföldi négyzetméterre 1500 címet jelent a Földön. Ezzel hatalmas lehetőségek tárulnak fel, hiszen így lehet külön IP-címe egy lakás összes elektromos berendezésének a hűtőtől a riasztóig, ami új szolgáltatások sokaságát biztosítja a felhasználóknak. Az újonnan bevezetendő eszközöknek pedig már az új ajánlások szerint kell támogatniuk az IPv6-os rendszereket [25]. Viszont nagyon megnehezíti a routerek gyors működését, mert nem 32 bites, hanem 128 bites címtartományban kell a döntést meghozni, ezért a csomagok továbbítása lassulás nélkül a régi algoritmusokkal elképzelhetetlen [26]. Ennek a nehézségnek érzékeltetésére röviden összefoglaljuk a használatos címzési módokat.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

3.1.2. Class-based Internet Addressing

1993-ig a class-based architecture [27] volt érvényben. Ez felosztotta az IP-cím területet öt részre, ebből az első három (A, B és C) unicast adatfolyamra volt fenntartva, egy (D) multicast és egy további (E) későbbi felhasználásra volt félretéve. A különböző részeket a netid segítségével azonosították. Ez a címzés a routereket egy viszonylag egyszerű keresési algoritmusnak rendelte alá, a továbbító táblának három része volt a három különböző unicast számára. A beérkező igényhez tartozó keresési eljárás a következő volt:

- A csoport meghatározása a beérkező csomag célcímének szignifikáns részeinek vizsgálatával. Ez meghatározta, hogy melyik táblát kell használni.
- A keresett netid és a használt táblázatban szereplő célok között egy pontosan egyezőt találni.

A pontosan egyező cél megkeresésére sok ismert algoritmust használhatunk, például a bináris fa-algortmust. Ez a címzési-keresési mód nagyon jól működött az Internet korai éveiben, de később két jelentős problémával kellett szembenézni:

- Az IP cím tartomány kimerülése.
- A routing tábla méretének exponenciális növekedése.

3.1.3. Classless Internet-domain Routing

Annak érdekében, hogy lecsökkentsék a routing táblák méretének növekedési ütemét, és az IP-címtartományt minél eredményesebben ki lehessen használni, egy alternatív címkeresési metódust vezettek be Classless Internet-domain Routing (CIDR)[28] néven 1993-ban. Ez az eljárás nem korlátozza a netid méretét, mely így szabadon bármilyen hosszúságot felvehet. Így a változó hosszúságú előtagok követik az Internet fizikai felépítését. Ebben az esetben a routereknek mindig a legspecifikusabb szabály szerint kell továbbítaniuk a csomagot. Ezt a szabályt a leghosszabb egyező előtag adja meg. A CIDR-el a router párosítja a célcím előtagját a következő ugrás címével, ami routerről routerre haladva egyre nagyobb egyezést kell, hogy mutasson (hisz közeledik a céljához a csomag). A routernek a keresés folyamán mindig a leginkább egyező utat kell megtalálnia, ezt hívjuk „longest prefix matching problem”-nek [29]. Ez egy

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

komplexebb probléma, mint az egyszerű továbbítás, hiszen az előtag hossza nem meghatározott. Ezért a keresés során az eddigi legjobb eredményt is el kell menteni, valamint nemcsak az egyezés pusztá tényéről kell dönteni, hanem ez után is folytatni kell a keresést (sokszor hiába). A csomagosztályozást végző algoritmusokat a következő tulajdonságaik alapján szoktuk minősíteni-rangsorolni:

- Tárigény (Tároló kapacitás): A tárigény mutatja, hogy az algoritmus futása közben mennyi adatot tárol, mennyi adattal dolgozik. Azt gondolhatnánk, hogy jó, ha egy algoritmus sok adattal dolgozik egyszerre, de az ilyen típusú problémák esetében, mint a csomagklasszifikáció, az adathoz való hozzáférés ideje összemérhető az algoritmus futási idejével, ezért sokat nyerhetünk időben azzal is, ha kevés adattal dolgozunk, azokat viszont gyorsabban érjük el.
- Végrehajtási idő: Végrehajtási idő alatt az algoritmus futási idejét értjük. Elsődleges célunk ennek csökkentése.
- Algoritmus-komplexitás: Az algoritmus komplexitásával azt tudjuk kifejezni, hogy az algoritmus hányféle és mennyire elemi műveletekből áll. Itt törekednünk kell arra, hogy minél kevesebb műveletet alkalmazzunk, és azok közül is a legegyszerűbbet. (Pl.: egy összeadás kis komplexitású, míg egymásba ágyazott 'for' ciklusok nagyon nagy komplexitást eredményeznek.)
- Frissítés gyorsasága: A szabálytáblák dinamikus kezelése esetén igen gyakran kell módosítanunk a megadott táblázatot. Ennek elérési ideje is meghatározó, hiszen ha egy router valamiért tiltó listára helyez egy címet (vírus, spam miatt), akkor a szabálytábla módosítására van szükség.

A fenti jellemzőket nagyban módosíthatják a feladat külső paraméterei. Ez esetünkben is így van, ezek a meghatározó paraméterek:

- A leghosszabb előtag hossza: Ez a beérkező IP cím hosszát jelöli, ami IPv4 alatt 32 bit, IPv6 esetében 128 bit, jelölése W .
- A rendelkezésünkre álló szabályok száma: Ez az érték routerenként változó lehet, de általában 2 és 50 között mozog, jelölése N .
- Dimenziószám: A végrehajtási tér dimenzióját jelöli, jelölése d .

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

3.2. Routerek

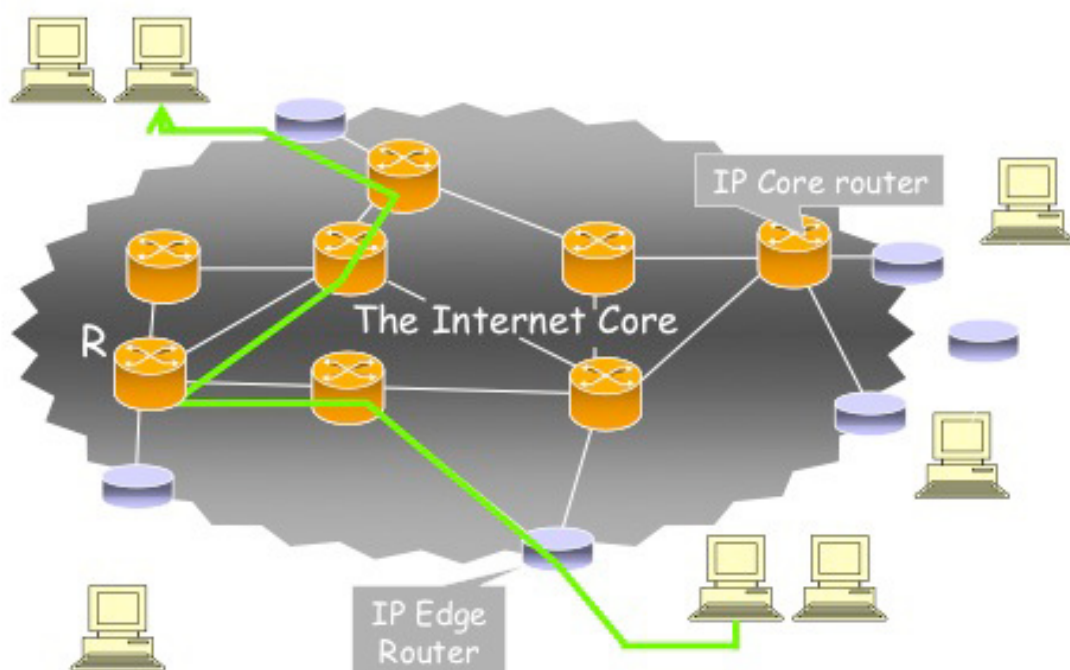
3.2.1. A routerek feladatai, funkciói

A routerek egy beérkező csomagot a célcímre, vagy afelé továbbítva látják el legfontosabb feladatukat, ehhez szükséges a szabálytábla, melyet a routerek memóriájukban tárolnak. A szabálytábla dinamikus módosítására is lehetséges, a routerek egymás közötti kommunikációjában lehetőség van a szabálytábla-változások továbbküldésére is. A szabálytábla tartalmazza a célcímekhez tartozó akciókat, a router a csomagklasszifikáció során a beérkező csomag fejlécét megvizsgálva a célcím alapján a célcímhez tartozó akciót végrehajtja a csomagon. Az Internet működésében megkülönböztetünk 'core' routereket, melyek igen gyors sávszélességgel vannak egymással összekapcsolva, valamint 'edge' routereket, melyek a csomag pontos célbaéréséért felelnek. Mivel a cél a csomagok minél gyorsabb célba juttatása, így előfordul, hogy minden csomag más utat jár be, ha a hálózat terheltsége ezt indokolja. A routerek számos funkciót ellátnak, automatikusan szűrnék, prioritást adnak kijelölt protokolloknak, illetve képesek minden olyan probléma elhárítására, melyek a hálózat működőképességét veszélyeztetik.

3.2.2. A routerek felépítése

A routerek működését alapvetően a következő egységek határozzák meg: a processzálást végző chip (és az azon futó algoritmus), a flash memória mérete és fajtája, a „sima” memória mérete és fajtája, a rendelkezésre álló tárhely – ez a sorbanállási hossz méretét határozza meg –, és a hálózati csatoló sebessége. A routerek alapvető felépítése első ránézésre ugyanolyan, mint egy hagyományos személyi számítógépé. A legfontosabb különbség azonban az, hogy a routerek lemezmentesek. Az útvonalválasztási feladat mellett a legtöbb esetben semmiféle információt nem kell, hogy tároljanak, azokban a komolyabb routerekben, ahol mégis naplózni kell eseményeket vagy hibákat, másik gépre vagy a router egy flash memóriájába írják ezeket. A másik fontos különbség a processzorban van: a routerekben nincsen egy olyan szabványos processzortípus, mint a személyi számítógépeknél az x86, vagy az Apple gépekben használt Motorola 68000, vagy akár a SUN által gyártott Sparc processzorcsalád. A routerekben teljesen egyéni, feladatorientált processzorok találhatóak. A Cisco SOHO 70 router családban például – az egyébként teljesen ehhez a routerhez kialakított

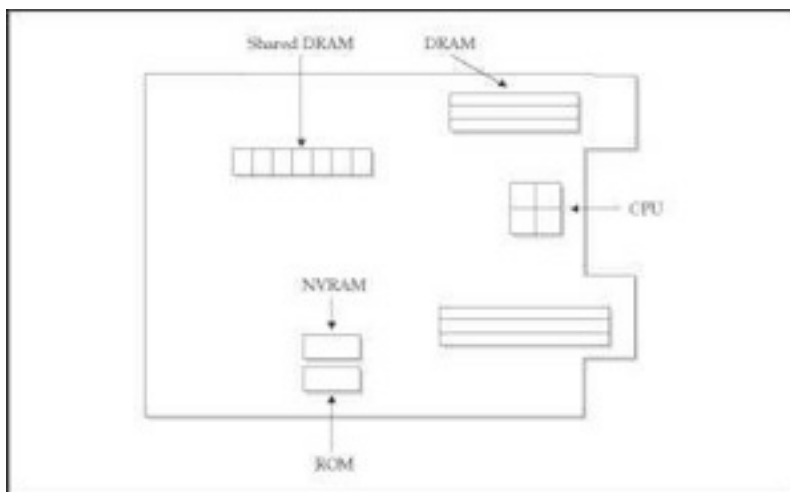
3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN



2. ábra. Router típusok: Core és Edge routerek: A Core routerek az Internet gerinchálózataért felelősek, míg az Edge routerek a végfelhasználókat kötik az Internetre. Forrás: <http://www.lightwaveonline.com/articles/print/volume-18/issue-5/special-report>

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

– Motorola MPC 855T RISC processzor 50Mhz-en működik.



3. ábra. Egy router architektúrája

A részletes hardverfelépítést minden gyártó szigorúan titkosan kezeli, hasonlóan a routeren futó csomagklasszifikációs algoritmushoz és annak képességeihez. A memória fajtáját és méretét megadják, de ez amúgy is könnyen cserélhető. Ha pusztán a memória mérete akadályozná a routerek működését, akkor nyilván többet tennének bele a gyártók, így viszont sejtethetjük, hogy a felhasznált memóriaméret körül van a routerek teljesítőképességének határa. A routerek célhardverek, tehát kialakításuk és felépítésük egyértelműen alá van rendelve a hatékony működésnek.

A nagy forgalmat bonyolító routerekben külön kártyák találhatóak, melyek a gyakran használt hálózati protokollok kiszolgálásáért felelnek. A maximális sebességük slotonként 2,5 Gb/s-től 10Gb/s-ig terjed. A nagyobb sebességet pusztán a slotok számának növelésével érik el. Amint az látható, a routerek felépítéséről keveset tudunk, de az biztos, hogy a piaci verseny hangsúlya inkább a routerekbe ágyazott szolgáltatások széles spektrumában van. Ezek elsősorban a könnyebb kezelhetőséget, a jobb konfigurálhatóságot és magasabb biztonságot jelentik.

3.2.3. A routerek képességei

Mára a csomagklasszifikáció a routerek egyik funkciójává zsugorodott, emellett széles spektrumú személyre szabott alkalmazásokat tartalmaz minden router, me-

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

lyek segítségével a rendszergazda nagyon könnyen és viszonylag tág határok között igazíthatja a router működését a felmerülő igényekhez. A routerek tekintetében mindenképpen a legnagyobb piaci erőt képviselő Cisco-t kell megnéznünk. A Cisco routereknél a belépő termékektől kezdve a professzionális routerekig mindegyik jelen van a piacon, mindegyik mélyén ugyanaz az IOS-kernel van.

Általános tulajdonságok (Cisco IOS): [30] A Cisco routerek az adatátviteli hatékonyság- és terhelés-optimalizáció érdekében nagyon intenzív és operatív kommunikációt folytatnak egymással. A Cisco fejlesztői a rendszer megfelelő menedzselése – hálózati stratégiák beállítása – érdekében egy kifinomult szoftveres környezetet hoztak létre, amelynek segítségével a megfelelő optimalizáció a rendszertervező által offline módon tervezhető része elvégezhető. A Cisco által kidolgozott IOS operációs mag minden komolyabb Cisco routerben megtalálható, ez a dinamikus szoftver megfelelő szakértelem segítségével könnyedén erős és jól működő rendszer kialakítását teszi lehetővé.

A Cisco IOS rendszer segítségével VoIP, V3PN, DMVPN-szolgáltatásokat automatikusan és dinamikusan tudunk létrehozni, ezzel nagyobb vállalatoknál az IP-alapú telefonálással, videokonferencia-szolgáltatással komoly megtakarításokat eredményezve. Ehhez tartozik a kifinomult QoS (quality of service)-szolgáltatás, melynek segítségével hálózatunkon belül forrás-IP, cél-IP, vagy szolgáltatás (pl. HTTP) alapján az adattovábbítás során elsőbbséget élvező csomagokat (csomagfolyamokat) tudunk kijelölni. Hasonlóan a fenti tulajdonságok alapján tudunk egy adott sávszélességet garantálni vagy korlátozni. A Cisco routerek egész spektruma elérhető a piac részére. A kisvállalatok béreltvonali végponti útvonalválasztási feladatokat ellátó kisebb eszközeiktől kezdve a komoly sokszálas optikát is kezelni tudó interfészekkel felszerelt 320Gbit/sec továbbítási sebességet is elérő legnagyobb csomópontokig a Cisco a termékek egész skáláját forgalmazza. A kisebb routerekben, az igényekhez mérten, dsl és isdn kártya is van, melyek a sávszélesség esetleges csökkenése (torlódás) esetén automatikusan aktivizálódnak. A torlódás kiküszöbölése érdekében a nagyobb routerek operációs magja is fel van készítve arra, hogy azonosítsa a régóta a hálózatban keringő csomagokat, és eldobja azokat.

Az IOS része a NAT (Network Address Translation) is, melyet az IPv4 címek sajnálatos hiányának következményeként, és lehetséges megoldásaként találtak ki. Ennek lényege, hogy a privát hálózatunkban fiktív IP címeket használunk, és eze-

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

ket a router a rendelkezésre álló információk alapján (pl. port számok) valódi IP címekké fordítja, így szolgáltatva zökkenőmentes kommunikációt. A NAT egyúttal a kisebb cégek részére egy elsődleges hálózati biztonságot is növeli, hiszen kívülről láthatatlanná teszi a belső hálózat gépeit.

Biztonság:

A Cisco által fejlesztett Self-Defending Network (SDN) [31] szintén minden routerben megtalálható. Ennek segítségével a routerek folyamatosan frissítik egymást a legújabb tűzfal-beállításokat, tiltólistákat, figyelendő címeket és portokat cserélve. Emellett a Network Admission Control (NAC) segítségével azonosítják a hálózaton lévő, sebezhető klienseket, majd korlátozzák ezek Internet-elérését, amíg a megfelelő biztonsági frissítésekkel (vírusölő, tűzfal) fel nem patchelik a klienst. A kívülről jövő támadások (flooding) ellen vagy a külső forgalom teljes blokkolásával, vagy IP, protokoll, vagy adatmennyiség alapján szűréssel védekeznek a routerek (az egymástól érkező csomagfolyamokat egyértelműen azonosítják az Authentication, Authorization and Accounting –AAA – protokoll segítségével).

A nagyobb rendszerekben, illetve azokon a hálózati helyeken, ahol a belső biztonság mellett az Internet felől publikus szolgáltatásokat is meg akarunk valósítani, a Cisco routerek rendszerszinten támogatják az úgynevezett Demilitarized Zone (DMZ) létrehozását, ahol tetszés szerint szűrhetünk minden kifelé illetve befelé irányuló forgalmat úgy, hogy a biztonságos belső hálózatot minél kevésbé veszélyeztessük.

Minden Enterprise kategóriájú Cisco routerben önálló hardveres titkosító chip van, mely AES, DES, 3DES –ben kódolja a küldött adatot, és tehermentesíti a központi processzort a titkosítással járó erőforrás-elvonástól.

3.3. A csomagosztályozás formális modellje – packet classification táblák

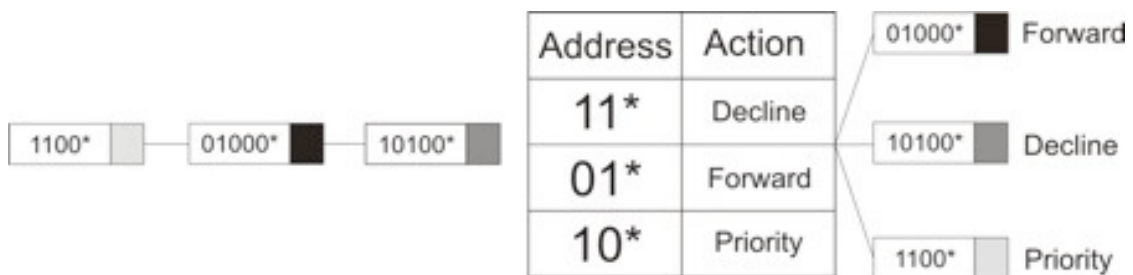
3.3.1. A csomagosztályozás feladata

A fentiek alapján a csomagosztályozás feladata az, hogy a beérkező fejlécmezők alapján a csomaghoz kapcsolódó akciókat (tevékenységeket) jelöljön ki, pl. tűzfal esetén tiltott címről érkező csomag eldobása, VoIP esetén prioritást biztosítson. Ez a feladat a következőképpen formalizálható:

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

- q_1, \dots, q_D jelöli a csomag fejlécmezőiben megfigyelhető bináris stringeket;
- adott a fejléceken értelmezett logikai függvényeknek egy f_1, \dots, f_L halmaza, valamint minden egyes függvényhez egy A_1, \dots, A_L akció;
- találjuk meg a lehető leggyorsabban azokat a logikai függvényeket i_1, \dots, i_l amelyekre $f_{i_j}(q_1, \dots, q_D) = TRUE, j = 1, \dots, l$
- ezek közül kiválasztva a legnagyobb prioritású $v \in \{i_1, \dots, i_l\}$ -t hajtsuk végre A_v -t.

Az előző definíció alapján a Packet Classification (PC) működését a 4. ábra szemlélteti:



4. ábra. A PC-táblák működése

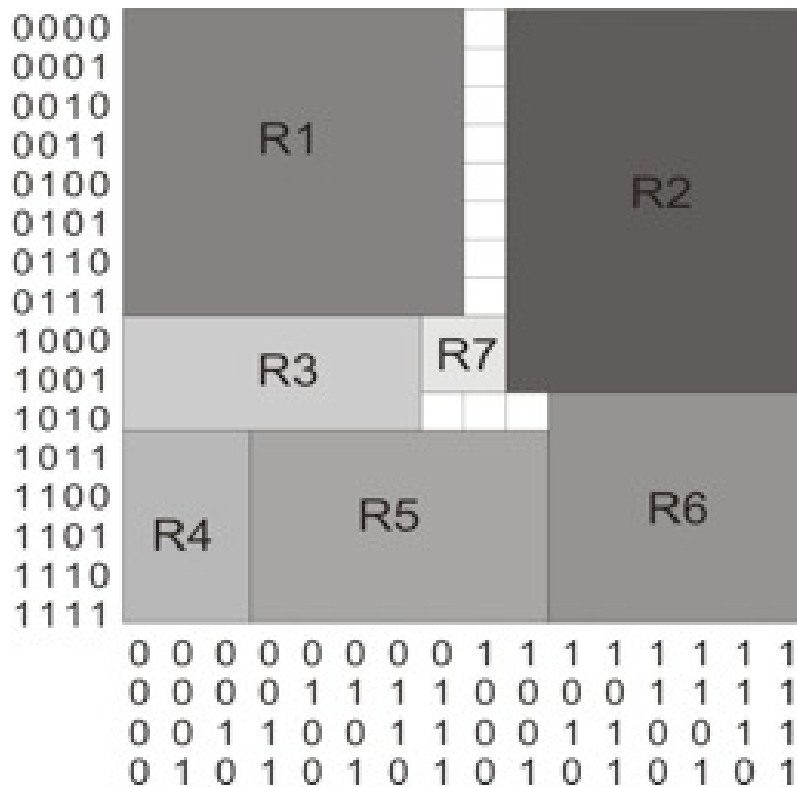
A gyakorlatban a logikai függvények általában maszkolást jelentenek, ezért teljesülésük igazából illeszkedést („matching”) jelent. A legnagyobb prioritása a leghosszabb illeszkedésnek van. Egy gyakorlati PC-táblát a következő táblázat szemléltet:

Rule	F1	F2
R_1	00*	00
R_2	0*	01
R_3	1*	0*
R_4	00*	0*
R_5	0*	1*
R_6	*	1*

1. táblázat. PC-tábla felépítése

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

Az előzőeknek megfelelően a PC mint geometriai feladat is interpretálható (lásd [32, 33, 1]). Jelölje Q azt a halmazt, amely az A_i akcióhoz tartozik, nevezetesen $Q_i = \{q_1, \dots, q_D : f_i(q_1, \dots, q_D) = TRUE\}$. Megfigyelve a csomagfejléct q_1, \dots, q_D ez meghatároz egy D dimenziós pontot $r = (q_1, \dots, q_D)$, ahol minden koordináta a megfelelő bináris formában adott. A feladat a lehető leggyorsabban megtalálni azt a Q_i halmazt, amely tartalmazza r -et, azaz $r \in Q_i$. Ez alapján az A_i akció elvégezhető.



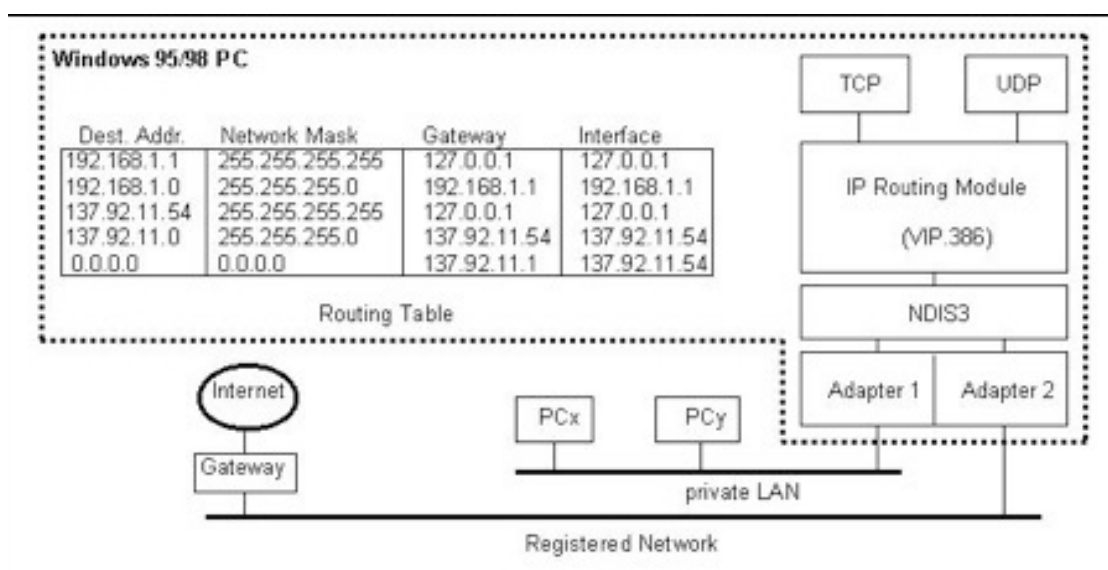
5. ábra. Az osztályozó geometriai megjelenítése

A 5. ábrán látható egy byte geometriai reprezentációja. Jól látható, hogy egy szabály egy területet definiál, egy beérkező IP csomag pedig egy pontot. Feladatunk a pont helyzetének meghatározása a síkon. Ennek a problémának kiterjedt irodalma van (lásd [34, 35, 36]).

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

3.3.2. Szabálytáblák

A szabálytábla tartalmazza a célcímhez tartozó akciót. Az itt tárolt akciókat fogja a router végrehajtani a beérkező csomagokon. A táblák mérete a router internet kapcsolatától, a hálózatban elfoglalt helyétől függ. Az egyszerűbb routerek pár tíz szabályt ismernek, a komolyabb, széles kapcsolattal rendelkezők több mint ezret. (A routerek mindössze 0,7%-a tartalmaz ezernél több szabályt.) A szabályok 8%-a redundáns, ezeket a router csomagklasszifikációs tulajdonságainak megváltoztatása nélkül ki lehet törölni. A szabályok száma exponenciálisan nő, napjainkban már közelít az egymillióhoz. Az IPv6 bevezetésével ez a szám a sokszorosára fog nőni pár éven belül. Így a csomagklasszifikációs algoritmusoknak készen kell állniuk a nagyobb szabálytáblák kezelésére, a klasszifikációs idő csökkentése mellett, ami szinte lehetetlen feladatnak tűnik.



6. ábra. Egy egyszerű többinterfészes számítógép routing architektúrája. Forrás: <http://www.nat32.com/image5.jpg>

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

3.4. Tradicionális PC-algoritmusok rövid összefoglalása

A következő fejezetben egy IEEE Network-ben megjelent összefoglaló [2] alapján gyűjtöm össze a csomagklasszifikációs algoritmusokat, kiemelve azokat a részeket, amelyek jelenleg is használatosak, kiegészítve kurrens felhasználási- és performancia-adatokkal. Az algoritmusok között megemlítem azokat is, amelyeket bár nem használnak, de a mai algoritmusok alapját képezik.

Az Interneten történő címzés változásával megnőtt a célkeresés összetettsége, ezért a régebben használt keresőalgoritmusoknak is változniuk kellett, vagy újakat kellett (és kell) bevezetni helyettük. A dolgozatban most röviden bemutatom az elmúlt egy évtized címzési változásait, és a használt jelentősebb keresési algoritmusok működését, előnyeit és hátrányait (lásd [37]). Az algoritmusokat a 2. táblázat foglalja össze kategóriák szerint:

Kategória	Algoritmus
Adatstruktúrán alapuló	Lineáris keresés, Hierarchikus keresés, Radix trie
Geometriai alapú	Grid of trie, AQT, FIS tree
Heurisztikus	RFC, High cuts
Hardware megoldás	Ternary CAM

2. táblázat. Jelentősebb PC-algoritmusok típusonként

Jelen dolgozat nem teszi lehetővé az összes algoritmus részletes bemutatását, de a fontosabbakat röviden bemutatja.

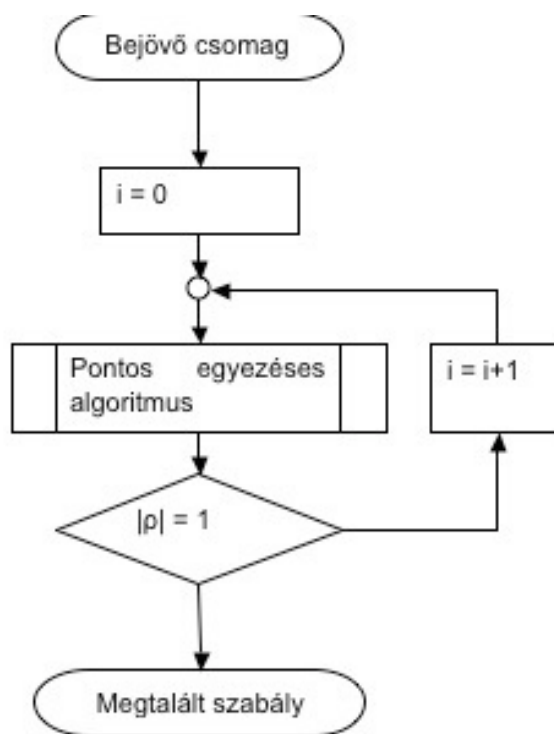
3.4.1. Kimerítő keresés

Az algoritmus lépései a következők:

- Legyen $i = 0$, ahol i az előtag hosszát jelöli.
- Végrehajtunk egy pontos egyezései algoritmust.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

- Eltároljuk a szabályokat, amik a beérkező csomagok fejlécével egyeznek, ez legyen ρ .
- Növeljük $i := i + 1$ -el, és végrehajtjuk a második lépést újra. ($0 \leq i < 32$)
- Ha $|\rho| = 1$, akkor az adott szabályt kiválasztjuk, mert ez a legspecifikusabb szabály.



7. ábra. A kimerítő keresés működési diagrammja

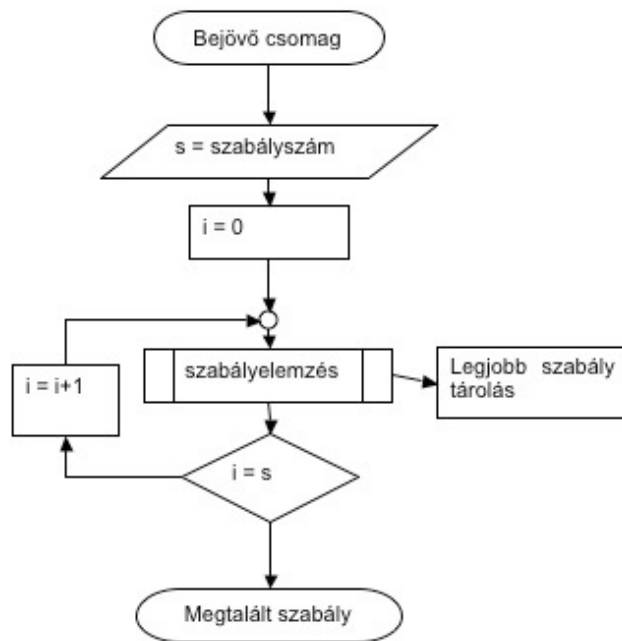
Természetesen az algoritmus megkíván pontosan egyező szabályokat, valamint a lefutási sebessége is hagy kivetnivalót maga után.

3.4.2. Lineáris keresés

Az egyik legegyszerűbb adatstruktúra az előtagok listájában a továbbítótábla. A keresőalgoritmus áthalad az előtagok listáján, és jelenti a leghosszabb egyezőt. Az

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

algoritmus tárolókapacitása $O(N)$, N számú előtag esetén. A keresési idő $O(N)$, és emiatt nagy N esetén túl lassúnak bizonyul, bár rövidíthető, ha a szabályokat méret szerint csökkenő sorba rendezzük.



8. ábra. A lineáris keresés működési diagrammja

3.4.3. A címek gyorsítótárba másolása

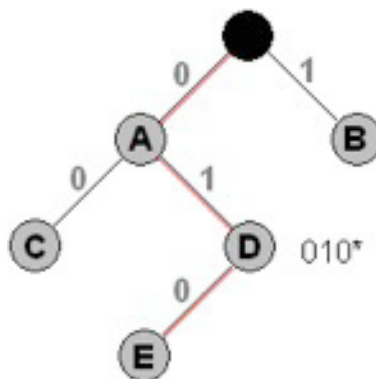
A gyorsítótár a processzorok teljesítményének fokozására találták ki, de jól használható útvonalkeresésre is úgy, hogy a gyakran keresett címek elérési útját egy route-cache-be másoljuk. A teljes keresés csak azoknál a címeknél szükséges, amelyek nincsenek a gyorsítótárban. Hogy gyorsítsuk a keresést, a gyorsítótárak találati arányának magasnak kell lennie. Ha például a teljes keresés húszszor annyi időt vesz igénybe, mint a gyorsítótáras keresés, akkor ennek a találati aránynak 95%-nak kell lennie ahhoz, hogy az algoritmus sebességét tízszeresére növeljük. Figyelembe kell vennünk azonban azt is, hogy az internetes adatforgalom növekedésével, vagy a hostok számának növekedésével együtt kell növelnünk a gyorsítótár méretét, ami

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

az előbbieket lineáris növekedésének esetén exponenciális gyorsítótár-méretnövekedést eredményez. Ezen felül az Interneten történő adatforgalom növekedésével együtt csökken a találati arány. Emiatt a gyorsítótár egyre kevésbé hasznosítható napjainkban.

3.4.4. Radix Trie

A Radix Trie (másnéven Patricia Trie) egy bináris fa, címkézett ágakkal (lásd [38, 39]). Ez a keresés keresztülmegy a bináris fán, a csomag fejlécének egymás utáni bitjei szerint haladva. A fa minden pontjából két ág van lefelé, a baloldali a '0'-nak, a jobboldali az '1'-nek. Aszerint, hogy a kezdőponttól milyen úton jutottunk el az adott szint adott pontjára, ez egyértelműen megad egy, a szint számával megegyező hosszúságú bináris számot. Például, ha definiálva van egy 1^* vonatkozó szabály, akkor ez vonatkozik a kezdőponttól jobbra elhelyezkedő összes gyerekre. A fa elfogadó-állapotai azok a pontok, amelyekre van érvényes szabályunk (pl. 010^* szabály).



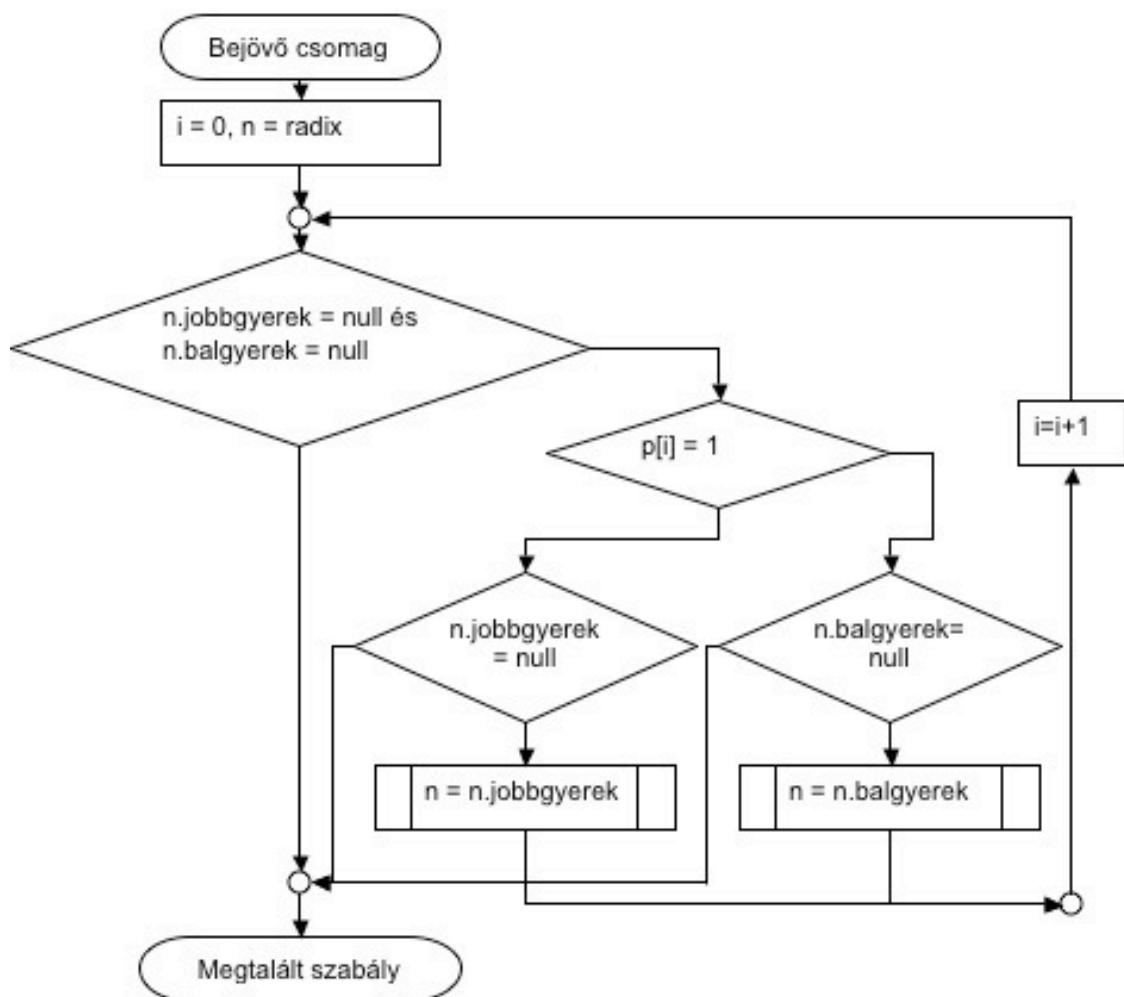
9. ábra. Radix trie működése

Az algoritmus a következő lépésekből épül fel:

- A kezdőpont lesz a pont, amit vizsgálunk, és $j=1$
- Ha a beérkező cím első bitje 1, akkor jobbra lépünk egyet lefelé, ha 0, akkor balra.
- $j = j + 1$ a beérkező cím következő bitjét vizsgáljuk.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

- Keresés közben feljegyezzük a legspecifikusabb szabályt, amely e szerint vonatkozik a beérkezett csomagra.
- Mikor vége a keresésnek, a feljegyzett szabály lesz a legspecifikusabb, ami a beérkezett csomagra vonatkozik.



10. ábra. Radix trie működési diagrammja

A Radix Trie-jal való keresés W nagyságú memória-hozzáférést igényel, a végrehajtási idő nagyságrendje $O(W)$. W bites fa tárolókapacitása N szabállyal

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

$O(NW)$. Ez a keresőalgorithmus 32 bites memória-hozzáférést igényel a legrosszabb esetben, és a sebessége sem optimális.

A Radix Trie továbbfejlesztéseként született meg az **LC Trie** (Level Compressed Trie), amely a Radix Trie azon hiányosságát igyekszik pótolni, hogy túl sok node van a fában, és így feleslegesen nagy a memóriaigény (lásd [40]). Sajnos az LC Trie egy komoly hibája, hogy az inkrementális frissítéseket nagyon lassan lehet rajta elvégezni, ezért ez elsősorban statikus táblák esetében használható módszer.

Ennek a módszernek a továbbfejlesztéseként lehet tekinteni a **Lulea** algoritmusra, amelynek a legfontosabb célkitűzése az, hogy egész nagy szabálytábla is elférjen a processzor cache-ében, így biztosítva a gyors hozzáférést az adatokhoz (lásd. [41]) Ugyanakkor két memóriahozzáférésre van szüksége minden node eléréséhez, nem is beszélve arról, hogy a szűkös helykihasználás érdekében olyan kompaktan vannak tárolva az adatok, hogy gyakorlatilag lehetetlen inkrementális frissítést elvégezni rajta, a legtöbb esetben a teljes táblát újra kell építeni frissítéskor. Az algoritmus másik hátránya, hogy a dedikált memóriarendezése nem skálázható sem nagy szabályhalmazokra, sem IPv6-ra.

3.4.5. Tree Bitmap

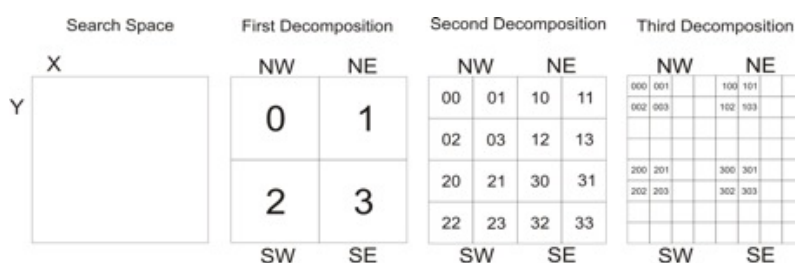
A Tree Bitmap a Lulea hibáit igyekszik kijavítani, és egy skálázható és IPv6ra is működő algoritmust hoz létre (lásd. [42]). Az algoritmus nagy előnye, hogy hagyományos alkatrészekkel (SRAM, RAMBUS) dolgozik, gyors keresést tesz lehetővé, ugyanakkor gyorsabban frissít, mint a Lulea. A jelenleg használatos Trie-algoritmusok közül ezt használják a leggyorsabb routerekben. A módszer lényegében memóriaelérési minták segítségével olyan módszert hozott létre, amely teljesíti a következőket:

- Egy node-ot egy memóriaelérési ciklus alatt ér el.
- Optimalizált helykihasználtságának köszönhetően kis helyen elfér az adatbázis.
- Hagományos architektúrákra épít.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

3.4.6. Area based quad tree

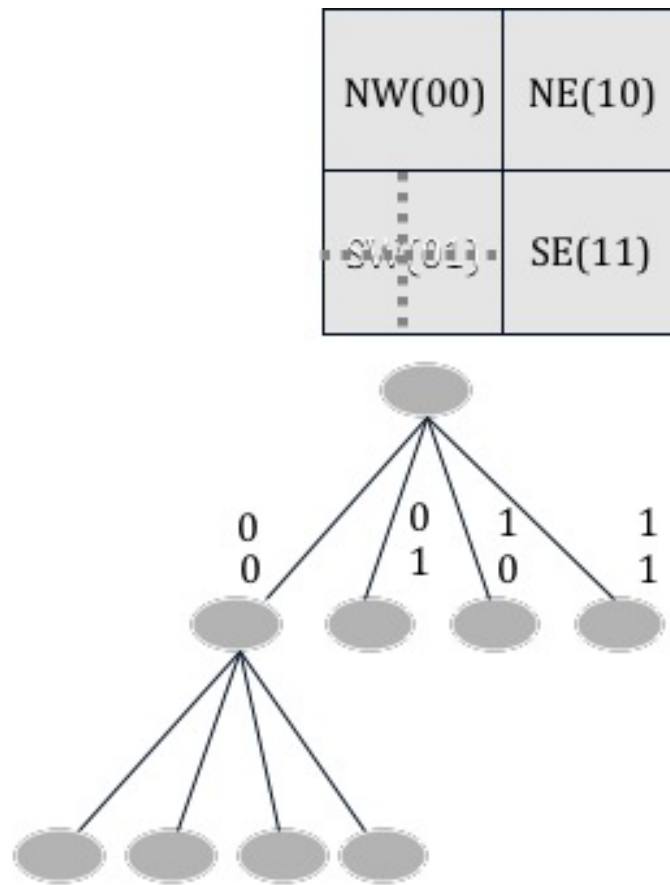
Kilépve a hagyományos architektúrára építő algoritmusok közül, a területalapú algoritmusok sorából elsőként az AQT-t ismertetem. Az Area based quad tree (AQT) része a PACARS (a kereső-osztályozó algoritmusok rekurzív területfelosztáson alapul) algoritmus-osztálynak (lásd [43]). Az AQT-t eredetileg két dimenzióra fejlesztették ki, de kiterjeszhető multi-dimenzióra. Az algoritmus alkalmazza a csomagosztályozás geometriai reprezentációját. Mielőtt részletesen bemutatnánk az algoritmust, szót kell ejteni a területfelosztásról és a quad tree-ről.



11. ábra. Area based quad tree felépítése. [1]

A quad tree a címek egy olyan geometriai ábrázolása, ahol a téglalapok felosztását addig növeljük, amíg egy téglalapban csak konstans mennyiségű információ van. Az alap quad tree-nek négy része marad, ezeket negyedeknek hívjuk (angol égtájak szerint elnevezve *NW*, *NE*, *SW*, *SE*). A felosztás rekurzív módon történik, minden negyednek van négy alnegyede, és így tovább.

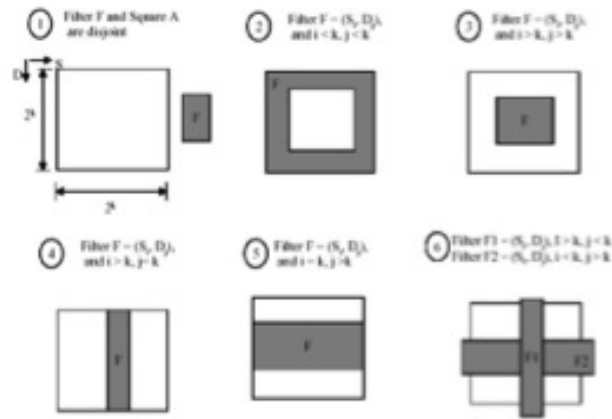
3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN



12. ábra. Area based quad tree dekompozíciója [2]

Az AQT bemutatásának részeként meg kell értenünk a CFS (crossing filter set) mibenlétét. A következő ábra mutatja, hogy egy negyed és egy filter hogy viszonyulhat egymáshoz.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

Crossing Filter Set (CFS)

13. ábra. A filter és a negyed viszonya az AQT-ben, fehérrel ábrázolva az állapottér, szürkével a filter [2]

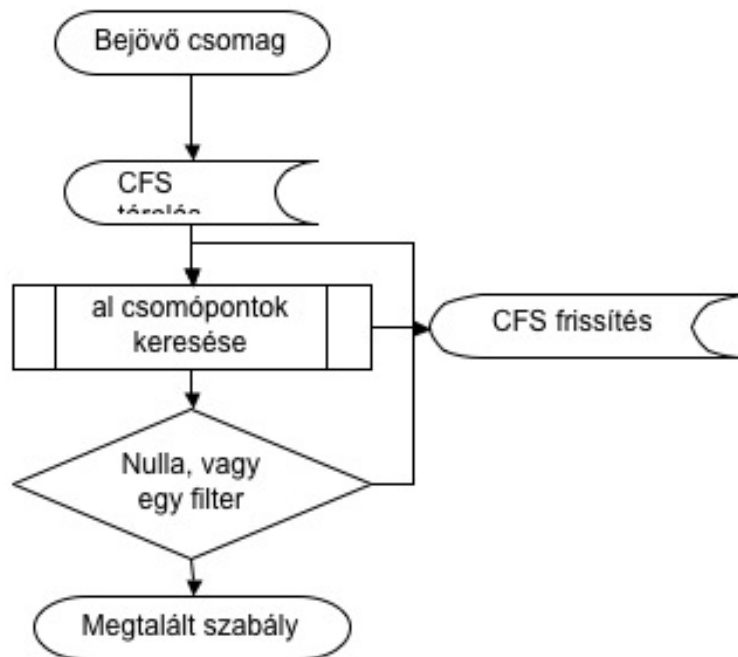
- A filter és a negyed között nincs összefüggés. A filtert nem kell számításba vennünk.
- A filter tartalmazza a negyed összes pontját. Ezt továbbra is nyomon kell követnünk.
- A filter összes pontját tartalmazza a negyed. Ez további vizsgálatot, felosztást igényel.
- Az utolsó eset, amikor a filter metszi a negyedet. Ez történhet mindkét dimenzióban, több filter keresztezheti is egymást.

Az utolsó a legérdekesebb. Azt mondjuk, hogy R filter keresztezi a Q negyedet. Lehet több R is, melyek nem csak a Q -t, de egymást is keresztezhetik. Ezeket a szabályokat a CFS-ben tároljuk. A CFS-t felhasználhatjuk a vizsgált tér feletti quadtree létrehozására. Ennek lépései:

- Adott a p szabálycsomag.
- A keresőfa gyökere megegyezik az egész vizsgált térrel.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

- A CFS-t kiszámoljuk és eltároljuk egy adatbázisban.
- A CFS-ben található szabályokat kivonjuk p -ből, a vizsgált területet pedig az alábbiak szerint osztjuk fel:
 - Meg kell vizsgálni azokat a szabályokat, melyeket az alcsomópontok tartalmaznak, és ki kell számítani a megfelelő CFS-t.
 - Ezt rekurzív módon folytatni kell addig, amíg csak egy, vagy nulla filter van hátra.



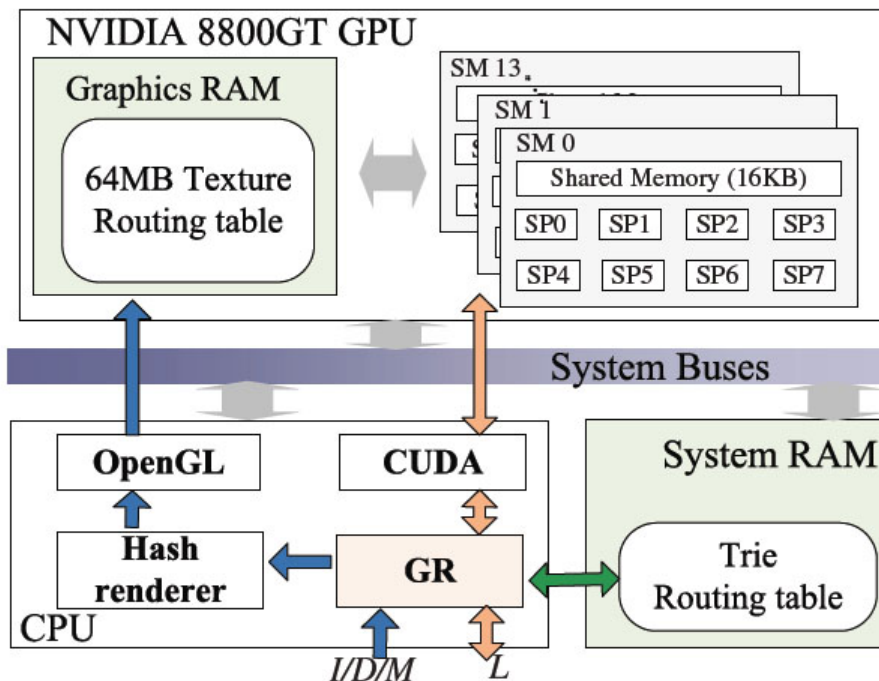
14. ábra. AQT működési diagrammja

A CFS használata garantálja, hogy a memória-igény $O(N)$ lesz, mert minden szabályt csak egyszer tárolunk el. Minden pontnak megvan a saját Crossing Filter Data Structure-ja (CFSDS). A CFSDS-eket eloszthatjuk két részre X és Y összetevőik szerint. Ez meghatározza azt is, hogy X , vagy Y irányban van-e párhuzamos filter. A filtert a CFS-ben minden szabályra kivetíthetjük egy tengelyre. Ha a filtert előtagok határozzák meg, akkor ezek a kivetítések is prefixek. Ez a kivetítés a problémát

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

egy „legjobban egyező filter”- (best matching filter) problémává egyszerűsíti, melynek széles irodalma van ([44, 43, 45]).

Jelenleg a GPU-alapú csomagosztályozási algoritmusok kidolgozásánál találhatunk AQT-alapú megoldásokat [3] (lásd 15). ábra).



15. ábra. Egy AQT-alapú GPU-megoldás rendszerterve [3]

3.4.7. Klasszikus módszerek teljesítőképesség-analízise

Számos tradicionális algoritmus létezik még, melyek a fent tárgyaltaknak módosításai. Ezek részletes tárgyalásától most eltekintünk, itt csak a legfontosabbak tulajdonságait soroljuk fel, hogy az új algoritmus bemutatása után az összehasonlításnál ezeket is figyelembe vegyük (lásd [46]).

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

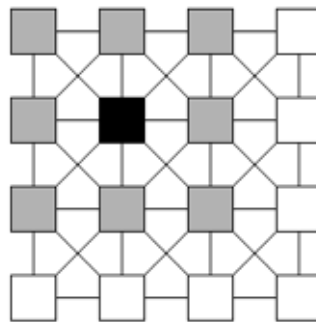
Algoritmus	Végrehajtási idő	Tárolókapacitás
Lineáris keresés	$O(N)$	$O(N)$
Radix Tree	$O(W^2)$	$O(N)$
LC Tree	$O(W)$	$O(N)$
Tree bitmap	$O(W)$	$O(N)$
Binary Search on Intervals	$O(\log_2(N))$	$O(N)$
Grid of Tries	$O(W)$	$O(NdW)$
Area-based Quad Tree	$O(W)$	$O(NW)$

3. táblázat. Tradicionális algoritmusok teljesítőképesség-analízise, ahol W a leghosszabb előtag hossza, N a szabályok száma, d a dimenzió.

3.5. A csomagklasszifikáció mint képfeldolgozási feladat – csomagklasszifikáció Cellular Neural Network (CNN) segítségével

3.5.1. A CNN bemutatása

A következőkben a CNN azon tulajdonságait mutatom be, amelyek szükségesek ahhoz, hogy csomagklasszifikációs feladatra használjuk (részletes leírás itt található [47, 48, 49]). CNN tekinthető egy analóg processzorok 2 dimenziós rácsának. A processzáló egységek a rácspontokban vannak, és összeköttetésük van a szomszédjaikkal. A 16. ábrán $C(i, j) = C(2, 2)$ cella valamint a hozzá tartozó $r = 1$ szomszédságot láthatjuk.



16. ábra. A kapcsolati mintázata a CNN-nek $r = 1$ esetén

Az i, j paraméterek határozzák meg a cella helyzetét a rácson. $S_r(i, j)$ a $C(i, j)$

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

cella r sugarú szomszédságát jelöli, amelyet a következőképpen definiálunk:

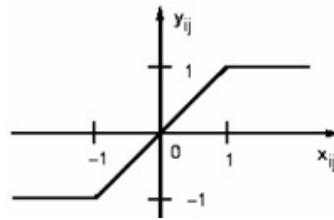
$$S_r(i, j) = \{C(k, l) \mid \max_{l \leq k \leq M, 1 \leq l \leq N} (|k - i|, |l - j|) \leq r\} \quad (1)$$

A CNN-ben kétfajta cella van: (i) belső cella és (ii) határoló cella. Egy cella belső cella, ha minden szomszédos cella eleme $S_r(i, j)$ -nek. Ha nem felel meg ennek a kritériumnak, akkor határoló cella.

A cellák állapota a következő differenciálegyenlet által írható le:

$$\dot{x}_{ij} = -x_{ij} + \sum_{C(k,l) \in S_r(i,j)} A(i, j; k, l) y_{kl} + \sum_{C(k,l) \in S_r(i,j)} B(i, j; k, l) u_{kl} + z_{ij} \quad (2)$$

ahol $A(i, j; k, l)$ és $B(i, j; k, l)$ a súlymátrixai a visszacsatolásnak és a bemenetnek. A kimenet a következők szerint írható le: $y_{ij} = f(x_{ij}) = 1/2|x_{ij} + 1| - 1/2|x_{ij} - 1|$ ahol $f()$ a nemlinearitás.



17. ábra. Egy CNN-ben használható nemlinearitás

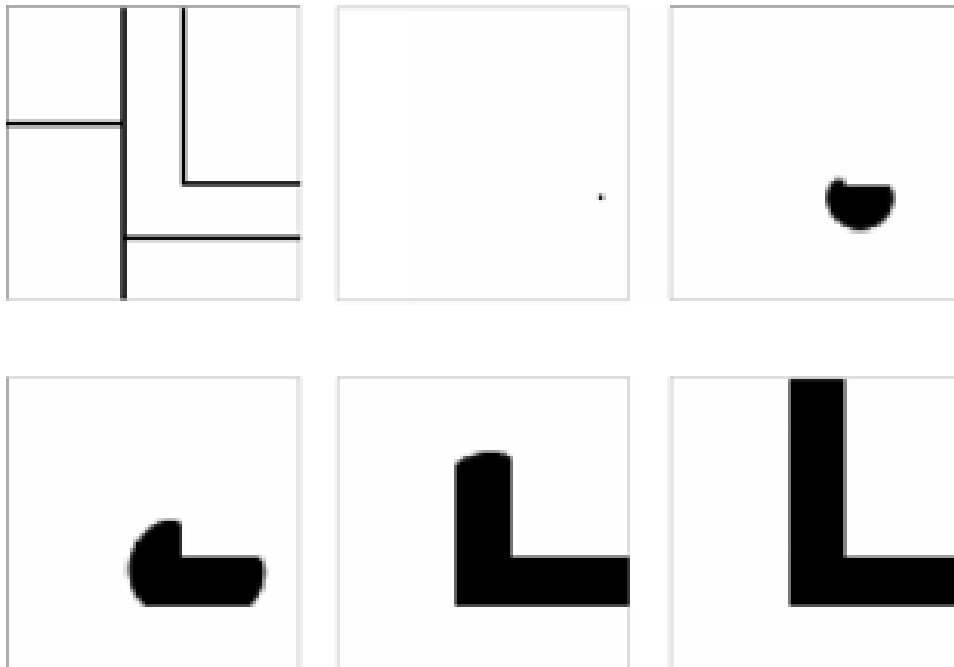
3.5.2. Új csomagklasszifikációs megoldás CNN-architektúrával

Mint ahogy az előzőekben már szó esett róla, a csomagklasszifikáció felfogható egy geometriai feladatként, ahol a beérkező csomag fejlécmezeje által meghatározott pontnak egy adott szabállyal fémjelzett halmazba való tartozását kell polinomiális komplexitásban kimutatni. Azaz, ha a szabálycsomagokat a fentiek szerint egy geometriai ábrába rendezzük, ahol minden terület egyértelműen meghatároz egy szabályt, akkor

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

csak azt kell kiolvasnunk, hogy a beérkező cím melyik területre esett. Így a probléma egy képfeldolgozási feladatként is interpretálható: ki kell "színezni" azt a halmazt, amelybe a kurrens csomag fejléce által meghatározott pont esik. Ez az interpretáció azért lényeges, mert amennyiben valós idejű képfeldolgozásra nyílik lehetőség (pl. egy CNN-hálózattal), a csomagklasszifikáció processzási ideje jelentősen felgyorsítható (ami az egyik szűk keresztmetszete a jelenlegi routing technológiának, és így az IPv6 alapú hálózati kommunikációnak is).

A fenti probléma megoldható egy CNN-architektúrával [50, 51, 52, 53], ahol az egyes képpontok (fekete/fehér) állapotának egy neuron bináris kimenete felel meg, valamint a beérkező csomag fejlécének megfelelő pontból hullámokat triggerelünk, amik az adott tartomány határáig "kiszínezik" a régiót (azaz a régióban fekvő összes neuron kimenete aktívvá válik a stacionér állapotban). Mivel a hullámterjedésnek megfelelő tranzien ideje a mikroszekundumok tartományába esik, a csomagklasszifikáció műveleti ideje is ebben a tartományban mozog, ami lehetővé tenné a Mbit/sec-os sebességű csomagfeldolgozást a routerekben. A megoldás intuitív képét a 18. ábra mutatja.



18. ábra. A hullámterjedés egy egyszerű szabályhalmazon ($\tau = 0,3,6,12,20$)

Ahhoz, hogy a csomagklasszifikációt CNN-re lehessen implementálni, a következő

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

feladatokat kell megoldani:

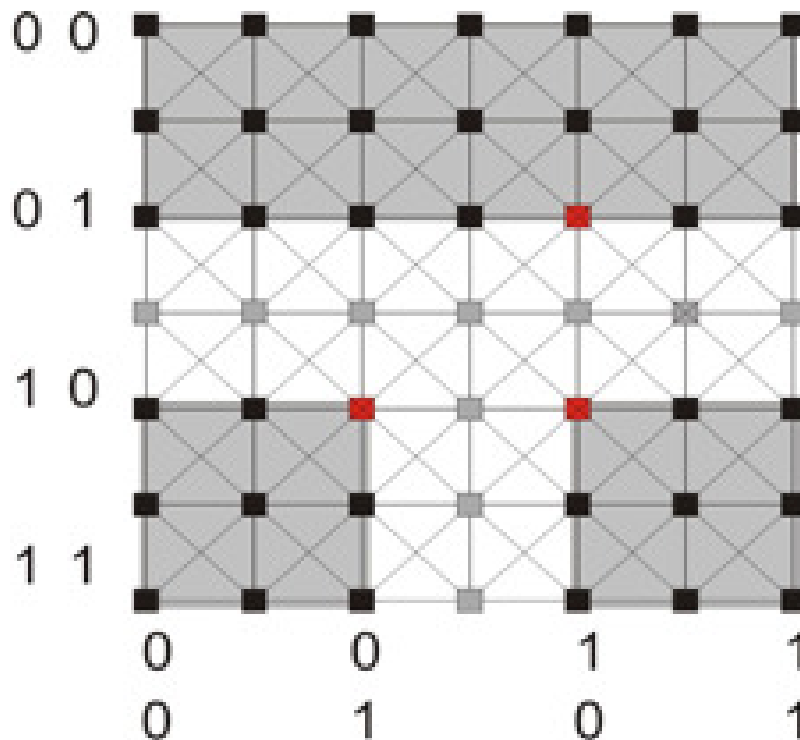
- a szabályok egyértelmű leképezése egy 2D-s térbe (még akkor is, ha több mint kettő fejlécmezőt kell vizsgálni);
- a szabályoknak megfelelő halmazkontúrok elhelyezése a CNN-en;
- a csomagfejlécnek mint pontnak "elhelyezése" a raszteren;
- hullámok generálása adott template szerint;
- a "beszínezett" rész kiolvasása.

A probléma CNN-es megvalósítása a következő geometriai leképezésen alapul: A jobb átláthatóság érdekében celláink számozását $C(i, j) = C(0, 0)$ -tól kezdjük. A kép pontjait két halmazra szeparáljuk, minden $u_{ij} : (i \wedge j = odd)$ pixel tartalmazza a bejövő IP-csomagot, míg a többi $y_{ij} : (i \vee j = even)$ pixel tartalmazza a szabályokat. A bejövő IP-cím egyértelműen meghatároz egy darab cellát a leképezésen, míg egy szabályhalmaz egy területet definiál. Feladatunk annak a területnek a kiolvasása, mely tartalmazza azt a cellát, amelyet az IP-cím meghatároz. Ennek a kézenfekvő módja – figyelembe véve a CNN előnyeit – az, hogy az IP-cím által meghatározott cella hullámokat indukál, amelyek átszínezik a szomszédos cellákat, egészen addig, amíg a szabályok által meghatározott határfelületet el nem érik [54]. Minden körbehatarolt területen definiálunk egy $r_t \in u_{ij} (i \wedge j = odd)$ referenciapontot. A referenciapont átszíneződéséből tudjuk egyértelműen kiolvasni, hogy melyik terület aktív.

Ennek a megoldásnak az alkalmazásánál a következő problémába ütközünk: ahhoz, hogy a csomagklasszifikációt egy chipen végezzük, szükségünk van a chipen belül az összes IPv6-os cím kétszeresének megfelelő számú cellára. A legnagyobb CNN-chip jelenleg 128X128 méretű (ACE16K, QEye), ugyanakkor a megoldási javaslat a CNN méretére vonatkozó igénye $2^{\lceil W/2 \rceil} \times 2^{\lceil W/2 \rceil}$. Tehát kénytelenek vagyunk a címfelismerést szétarabolni, például byte-okra. Ehhez két dolgot kell tennünk:

- A bejövő IP-címet byte-okra daraboljuk, így egy chip minden byte-ja egyértelműen meghatároz egy cellát. Ezek fogják indukálni a színhullámokat.
- A meglévő szabályainkat is feldaraboljuk. Egy szabály egy területet fed le a chip felületén, kivéve, ha a szabály egy byte-nál hosszabb (pl.:10101010111*);

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN



19. ábra. A referenciapontok elhelyezése a CNN területén

ez esetben ugyanis az első chipen ez a szabály is csupán egy cellát fog jelölni, a következőn pedig egy területet.

A szabályok átvitele geometriai síkra a következő módon lehetséges:

Az n -ik bytehoz tartozó CNN-chipen kialakítjuk a következő struktúrát minden olyan R_k -val, amely értelmezve van azon a byte-on:

$y_{i,j} = 1$, ha \exists olyan $p, q \in \{-1, 0, 1\}$ hogy $i + p$ és $j + q$ páros és $IP_{i+p,j+q} \in R_k[n]$, és $IP_{i-p,j-q} \notin R_k[n]$, egyébként $y_{i,j} = -1$.

Ahol $IP_{i,k}$ tartalmazza az értelmezett bejövő IP-csomagok bináris értékét a 32×32 cella méretű CNN felületén, a következő módon: $IP_{i,k} = [i(1) j(1) i(2) j(2) i(3) j(3) i(4) j(4)]$ nyolcbites értéket, ahol az $i(x)$ bináris értékének vett x -ik bitjét jelenti.

Egy olyan CNN-template-re van tehát szükség, amely ezt a feladatot valósítja meg [47, 48, 49]. A szükséges A,B templateket keresési algoritmusokkal sikerült elérni, amelyeket megfelelőségi vizsgálattal ellenőriztem, így ha a szabályokat a bemenetre rakjuk ($y_{i,j}$), és a beérkező pixelt a kezdeti állapotba ($u_{i,j}$), valamint a CNN szélén lévő pixeleket fehérre állítjuk (Boundary=-1), akkor a következő változókkal pont a

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

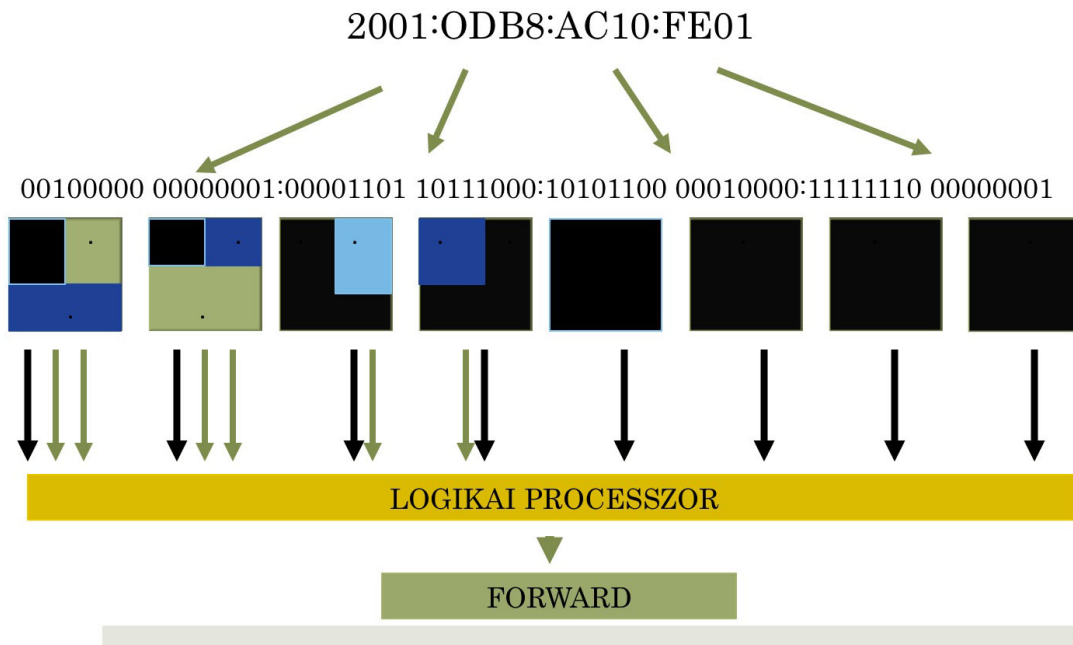
feladatot valósítja meg:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -8 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad Z_0 = -2 \quad (3)$$

3.5.3. Nagyobb szabályrendszerek esetén a szabályok diszjunkt területté való leképezése

Az architektúra tervezésekor fontos szempont volt, hogy minden esetet lefedjen a módszer. A klasszifikációs szabályok nagy része átfedő területeket hoz létre a CNN chip felületén. Ennek kiküszöbölésére az átfedő területeket önálló diszjunkt halmazokká bontottuk. A transzformáció után a logikai processzor feladata lesz egyszerű szabálytáblák alapján megtalálni a megfelelő sorszámú szabályt. A CNN futási feltételeként bármely referenciapont 1-re váltását (aktiválását) adjuk meg. A végső döntést végzi el a logikai processzor, aminek a bemenetére kötjük a 6 CNN-t (20. ábra). A logikai processzor mind a 6 CNN-től vár egy eredményt, majd a rendelkezésre álló szabályhalmaz szerint logikai úton eldönti, hogy melyik szabály teljesült.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN



20. ábra. Az architektúra logikai felépítése

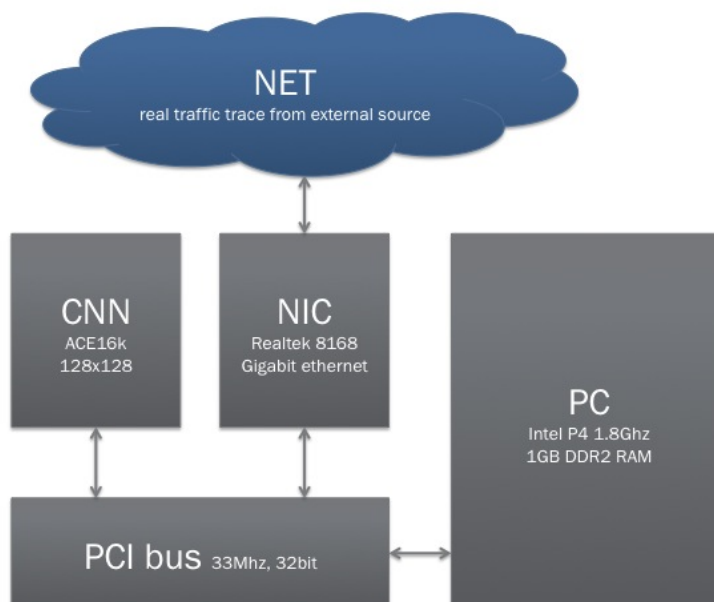
3.5.4. Megvalósítás

A technológiai megvalósítás során a következő célok elérése volt a feladat:

- a módszertan működésének bizonyítása
- a módszer teljesítőképességének vizsgálata
- valós adatokon végzett terheléses tesztek

A hálózati kommunikációt a hoszt számítógép végezte, ugyancsak itt helyezkedtek el a logikai kapuk is. A kapcsolat a CNN architektúrával PCI-sloton keresztül valósult meg. Az eredményeket szintén a PC értékelte ki. A megvalósítást a 21. ábra szemlélteti.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN



21. ábra. A CNN alapú csomagklasszifikáció megvalósításának rendszerdiagramja

3.6. Numerikus eredmények

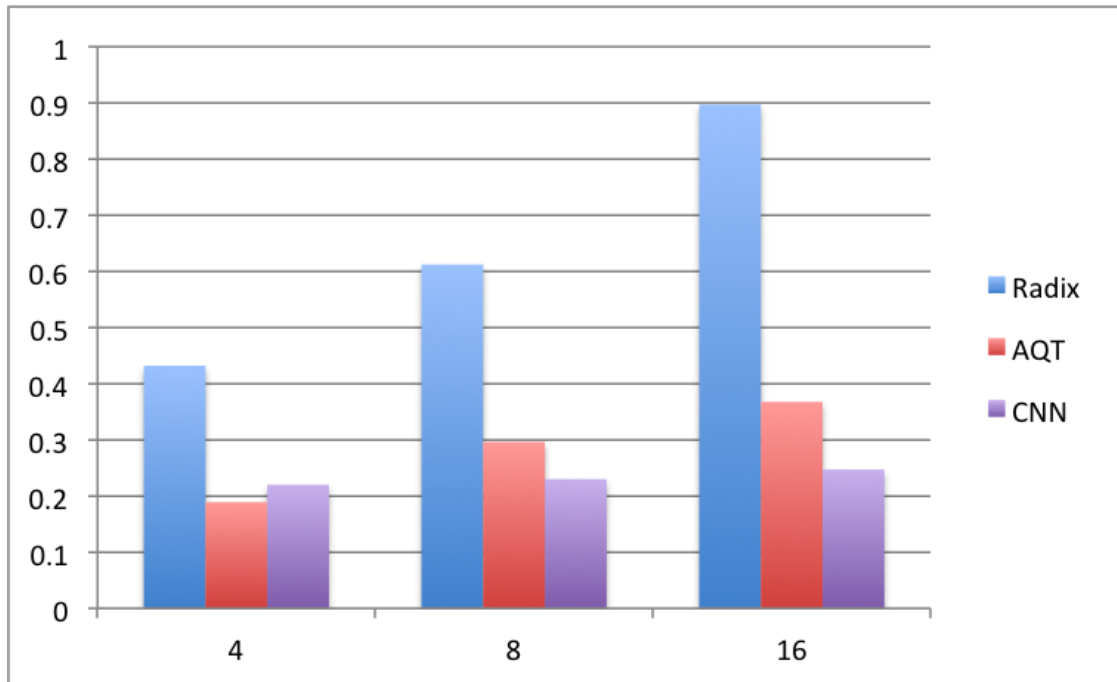
A numerikus eredmények, az összehasonlíthatóság és a modell vizsgálhatósága kedvéért, szimulációkkal készültek. Ezeket a Martin Haenggi, Java nyelven íródott CNN toolkitjéne segítségével sikerült megalkotni. A teljesség kedvéért, valamint a működőképesség igazolása miatt az algoritmust egy valós CNN-en (egy ACE16K rendszeren) is megvalósítottam.

A szimulációhoz a környezetet a következő módon alkottam meg: 80479 csomagot vettem át az ohioi Wilbreforce University Argus nevű szerveréről, amely 2 percnyi kommunikációnak felelt meg. A szerveren 1219 darab szabály volt aktív ebben a pillanatban. Az egyes algoritmusok közül a Radix Trie-t, valamint az AQT-t szimulációs programként megvalósítottam, majd ugyanazon a számítógépen, ugyanazon a hálózati topológián lefuttattam őket.

A következő eredmények egyrészt a szimulált környezetben változó paraméterű szimulált csomagfolyamok hatását, másrészt a leírt valós csomagfolyam tel-

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

jesítőkéességét mutatja be.

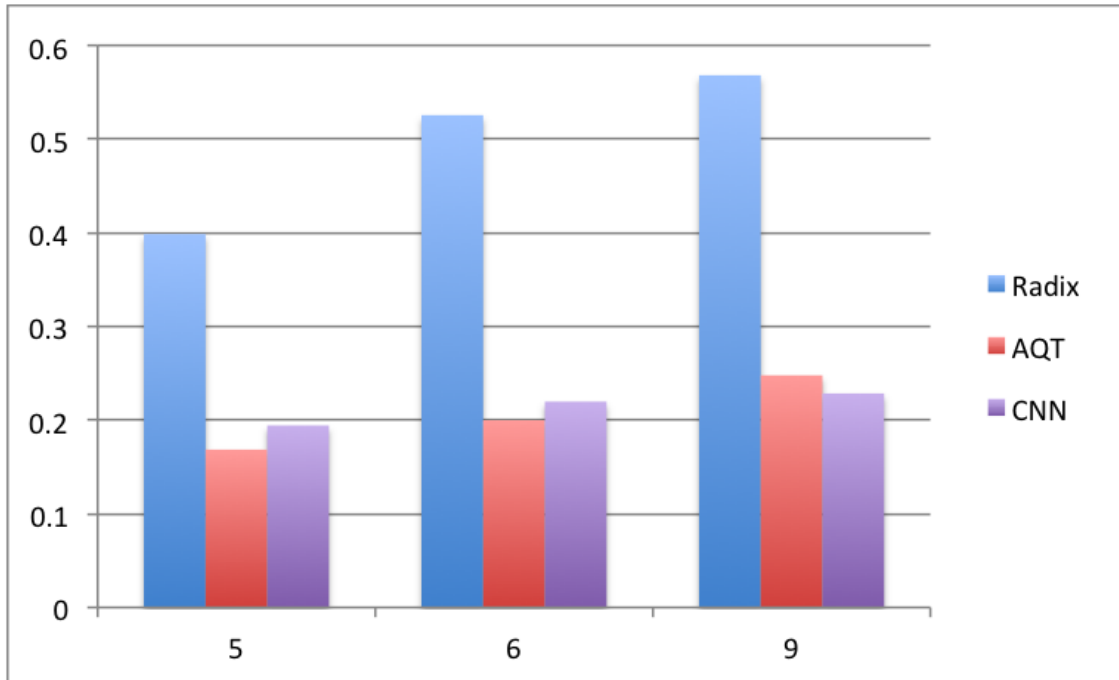


22. ábra. Különböző szabályszerűségű teljesítőkéesség (klasszifikációs idő - ms) azonos csomagforrás-paraméterek mellett. Látható, hogy a CNN-es megoldás, több szabály esetén a kis területek miatt hamarabb tüzel

Látható, hogy a Radix érzékeny a szabályszerűságra. Ugyanakkor elmondható, hogy az AQT abban az esetben, ha DSP-n (Digital Signal Processor) implementáljuk, a gyors szorzásnak és a párhuzamosságának köszönhetően a szabályszerűságra sokkal kevésbé érzékeny, mint a Radix. A pontos hatást az algoritmus esetében egyéb környezeti körülmények befolyásolják.

A szimuláció során az architektúrákat megfelelően modellezve építettem be a szimulációs környezetbe. Az AQT esetén egy TTT (Token Triggered Threading)[55, 56] alapú multithread DSP implementációt vizsgáltam meg, és ez alapján alakítottam ki a szimulációs környezetet, amelyet megvizsgáltam abból a szempontból, hogy milyen műveleteket végez el párhuzamosan, és ezeket a szimulációban is párhuzamosan végezték tekintetem. Ez az oka annak, hogy a DSP erős párhuzamossági képessége a szimulációban is megjelenik.

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

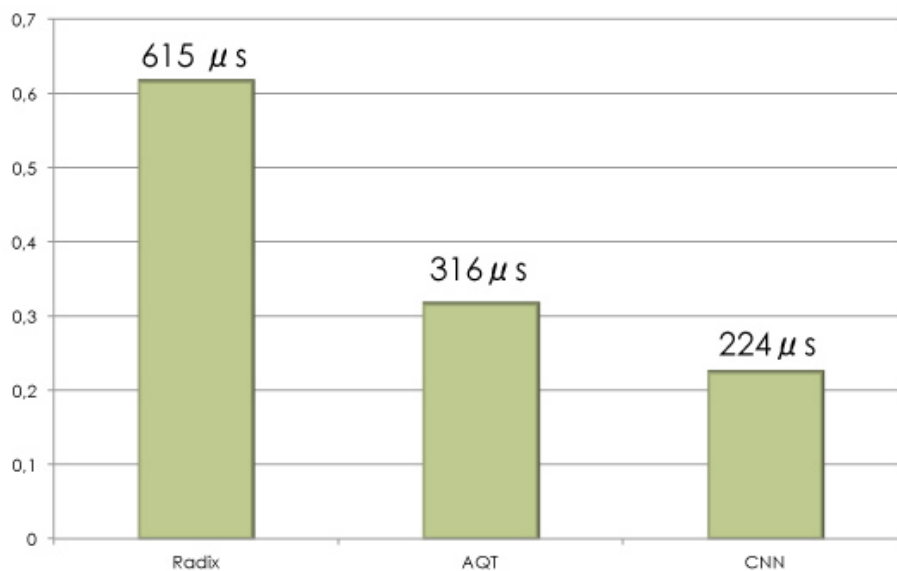


23. ábra. A klasszifikációs idő (ms) a csomagforrás hosszát változtatva (leghosszabb prefix - W) azonos szabálysám és csomagszám mellett

Jól látszik, hogy a hagyományosnak mondható Radix-módszer miért nem alkalmas IPv6-os hálózatok számára. A Radix a csomagforrás-hossz függősége miatt komoly hátrányt jelent a hosszabb fejléc kiválasztásakor. Ebben az esetben láthatjuk, hogy a CNN mutatja a legjobb eredményeket.

Végül a valós csomagfolyamon elért eredmények:

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

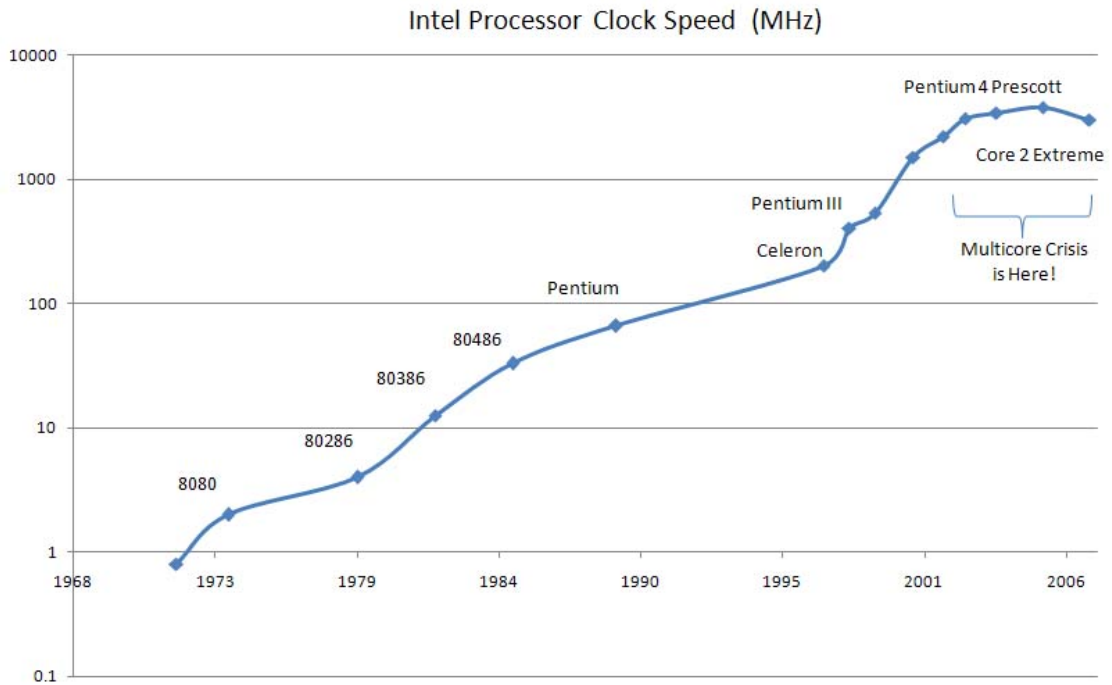


24. ábra. A klasszifikációs idő (ms) az egyes algoritmusokkal az Ohio-i egyetem szerveréről szedett csomagfolyammal

3.6.1. A CNN tranziens ideje és az igényelt adatátviteli sebesség közötti kapcsolat

A jelenlegi processzorok fejlődését (25. ábra) figyelemmel kísérve látható, hogy technológiai okoknál fogva bizonyos sebességnél gyorsabb CPU-t nem tudnak készíteni, ezért a fejlődés a többprocesszoros világ irányába mutat (26. ábra).

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN

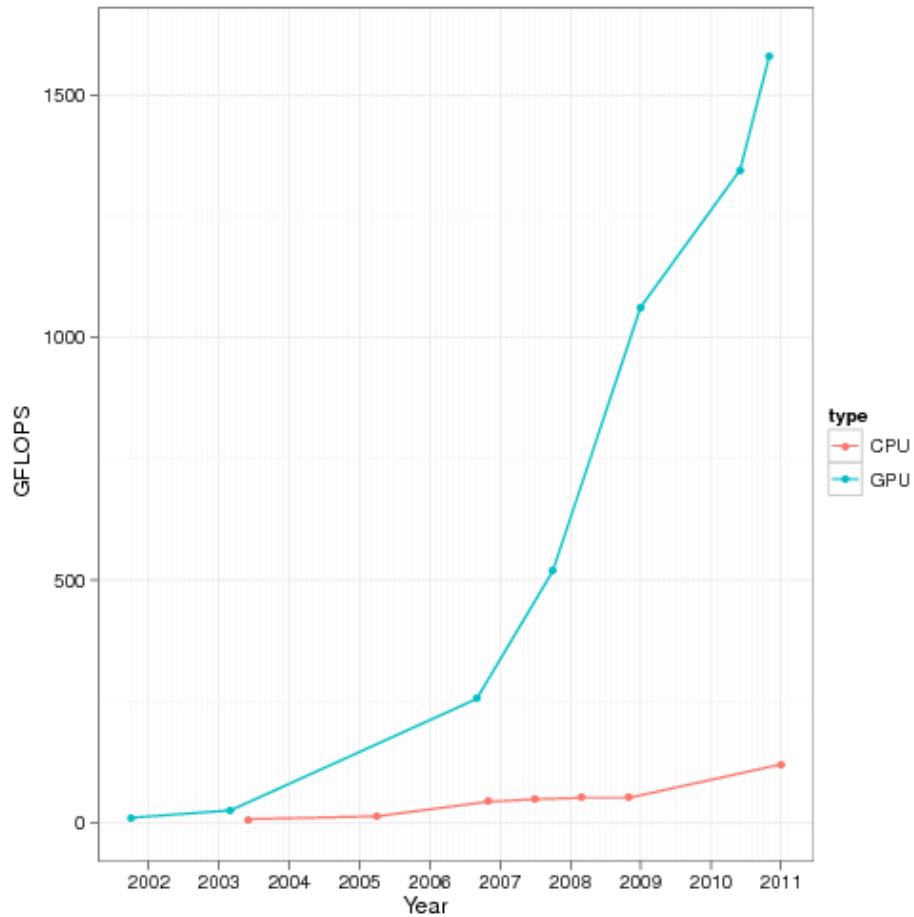


25. ábra. Processzorok órajelének fejlődése az elmúlt 40 év során. Forrás: <http://smoothspan.wordpress.com/2007/09/06/a-picture-of-the-multicore-crisis/>

Mindez a CNN implementálását is több lehetséges kontextusba helyezi:

- egyrészt a CNN chipben (CNN-UM) integrált node-ok számának növelésére és a működés gyorsításra,
- a CNN architektúrán futó számítások párhuzamos architektúrákon való gyors szimulációja (pl. GP-GPU)

3. CSOMAGKLASSZIFIKÁCIÓ IP-HÁLÓZATOKBAN



26. ábra. A CPU és a GPU fejlődésének összehasonlítása elvégzett műveletek alapján (GFLOP) Forrás: Paul E. McKenney 2011-ben kiadott "Is Parallel Programming Hard, And, If So, What Can You Do About It?"

Így a disszertációban bemutatott technológia, a sokprocesszoros rendszerek lehetőségeit kihasználva lehetőséget ad a Gbit/s-os sávszélesség igény kielégítésére. Ezt GPU[57, 58], VLSI vagy FPGA[59] implementációval lehet elérni.

4. VEZETÉKNÉLKÜLI SENZORHÁLÓZATOK

4. Vezetéknélküli szenzorhálózatok

A vezetéknélküli szenzorhálózatok élettartamának maximalizálása az energiafelhasználás optimalizálásával jelenleg is kutatások tárgya. Ezen munkák [60, 61, 62, 63, 64, 65] közös jellemzője, hogy általában az optimális energiaszint beállítását egy folytonos skálán keresik. Azonban a fizikai megvalósítás szempontjából a szenzorok adóenergiája egy diszkrét halmazból választható. Ennek megfelelően az optimális küldési energia meghatározását pusztán a node által használható diszkrét értékeken kell elvégezni. Ez viszont egy új típusú kombinatorikus optimalizálási feladathoz vezet.

Ebben a fejezetben három új, egymáshoz hasonló módszert dolgoztam ki, melyek mindegyike alacsony komplexitású műveletekkel képes olyan új útvonalakat választani, amelyek energiája minimális a node-ok diszkrét energiakészletéből választva. Ezek a megoldások szuboptimálisak ugyan, de a Leach-protokollhoz [16, 66] képest energiatakarékosabb útvonalválasztást tesznek lehetővé. A javulást két módon értelmezve

- egyrészt előírt megbízhatóságot garantálva (a csomag előírt valószínűséggel érkezik meg a bázisállomásra)
- másrészt a megbízhatósági kényszer mellett kisebb energiával juttatjuk a csomagokat a bázisállomásra.

4.1. Megbízható csomagtovábbítás vezetéknélküli hálózatokban

Az elmúlt években a vezetéknélküli szenzorhálózatok alkalmazása széles körben elterjedt, ezért, tekintettel a szenzorok limitált energiakészletére, szükségessé vált olyan kommunikációs protokollok kifejlesztése, amelyek energiafelhasználás-érzékenyek [14]. Ez a kérdéskör különösen előtérbe került az olyan helyzetekben, ahol az elemek feltöltése nem lehetséges. Erre jó példa a testbe implantált szenzor [67], amelynél a legfontosabb az, hogy a meglévő energiakészletet minél tovább használhassuk. Tekintettel arra, hogy a szenzorban a legnagyobb energiafelhasználása a rádióadónak van, az új kommunikációs protokollok leginkább a küldési energiára optimalizálnak [68]. Ez a feltétel különösen igaz az ún. multihop-kommunikációban, ahol az egyes csomópontok, annak érdekében, hogy a bázisállomásra juttassák az információjukat, igénybe veszik a többi szenzort is. A sikeres csomagtovábbítás valószínűsége

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

egy adott fading modell segítségével írható le (pl. Rayleigh fading) [13], így a multihop-kommunikáció során csomagvesztés léphet fel [15]. A feladat tehát olyan útvonalválasztó algoritmus kifejlesztése, amely garantálja, hogy az útvonalon a csomagvesztés valószínűsége egy előírt értéknél kisebb, ugyanakkor a csomag-továbbításban résztvevő node-ok energiafogyasztása minimális (a hálózat élettartamát maximalizáljuk)[16, 17].

Jelenleg az energia-tudatosságot a következő protokollok biztosítják:

- Energy Conserving Routing[69];
- LEACH [16];
- PAMAS [70].

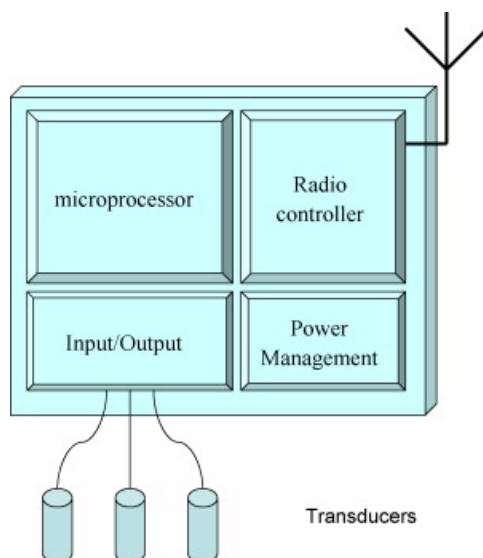
A LEACH eredetileg valóban 2000-ben lett publikálva, de azóta (2005-2009) több továbbfejlesztést publikáltak hozzá [71, 72]. A PAMAS algoritmus a legkorábbi (1998) algoritmus az említettek közül, ennek csak történetiség szempontjából van jelentősége. A ECR a LEACH eredeti verziójával egy évben jelent meg, ebben az irányban kevesebb továbblépés történt.

Az említett algoritmusok az energia-optimalitás mellett nem garantálnak előírt megbízhatóságot, ezért kihívás olyan új protokollok tervezése, amely az energiatudatosság mellett az előírt megbízhatóságot is garantálják.

4.1.1. Vezetéknélküli szenzorhálózat

A vezetéknélküli szenzorhálózatok alapvető építőeleme maga a szenzor. Ezek a szenzorok kis energiafogyasztású elektronikai eszközök, amelyek rendelkeznek legalább egy processzorral, memóriával, akkumulátorral, rádióegységgel. Ehhez az alapkiépítéshez kapcsolódik egy vagy több érzékelő szenzor (lásd 27. ábra).

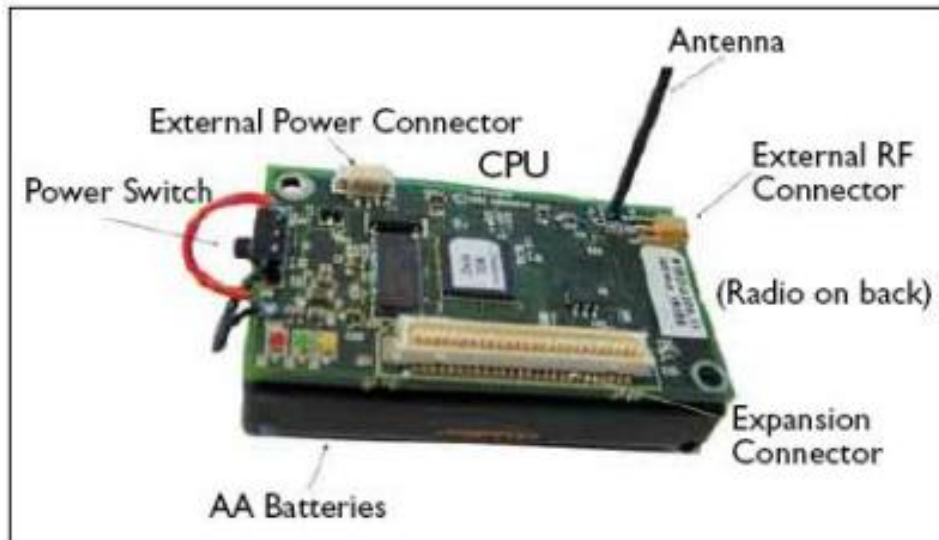
4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK



27. ábra. Egy szenzornode sematikus felépítése Forrás: <https://www.assembla.com/spaces/32093-project>

Ezek a környezetet érzékelő szenzorok lehetnek mechanikusak, termikusak, biológiaiak, kémiaiak, optikaiak vagy akár mágnesesek. [7] Több szenzorgyártó cég modulárisan építi fel ezeket az érzékelőket, hogy könnyen lehessen hozzá fejleszteni újabb érzékelő paneleket. A 28. ábrán egy Mica2 node látható, megjelölve a strukturális komponenseket[4].

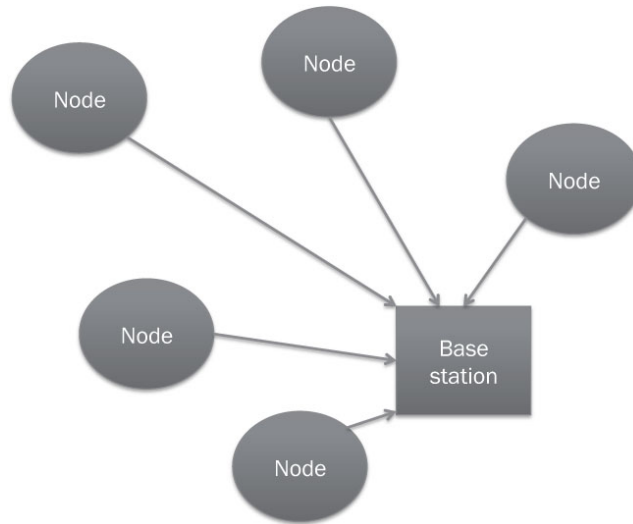
4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK



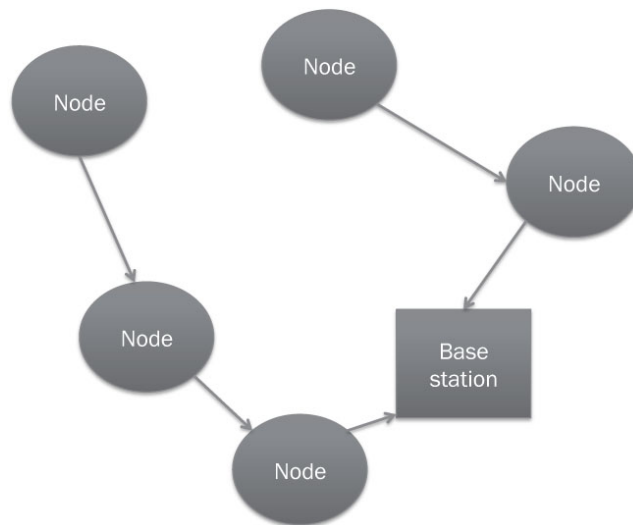
28. ábra. A Mica2 szenzornode felépítése [4]

Jól látható, hogy bővítő-panelek helyezhetőek a szenzorra. Általában ezek a szenzoregységek kevés memóriával rendelkeznek, valamint az érzékelt információra általában kis késleltetéssel van szükség, ezért szükséges, hogy az adatokat minél hamarabb a bázisállomásra továbbítsuk. A csomagtovábbítás történhet közvetlen módon (single-hop, 29. ábra) és közvetett módon (multi-hop, 30. ábra).

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK



29. ábra. Közvetlen (single-hop) kommunikáció



30. ábra. Közvetett (multi-hop) kommunikáció

Ezen túl az egyes csomagtovábbításokról a protokollok tervezésekor eldönthetjük, hogy kérünk-e visszaigazolást (acknowledgement) vagy sem. A csomagtovábbítás lehetőségeit a 4. táblázat szemlélteti.

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

Single-Hop Visszaigazolás nélkül	Single-Hop Visszaiga- zolással
Multi-Hop Visszaigazolás nélkül	Multi-Hop Visszaiga- zolással

4. táblázat. Vezetéknélküli csomagtovábbítás típusai

Jelen fejezet a Multi-Hop Visszaigazolás nélküli csomagküldési protokollok energiatudatosságával, valamint előírt csomagküldési valószínűség teljesítésével foglalkozik.

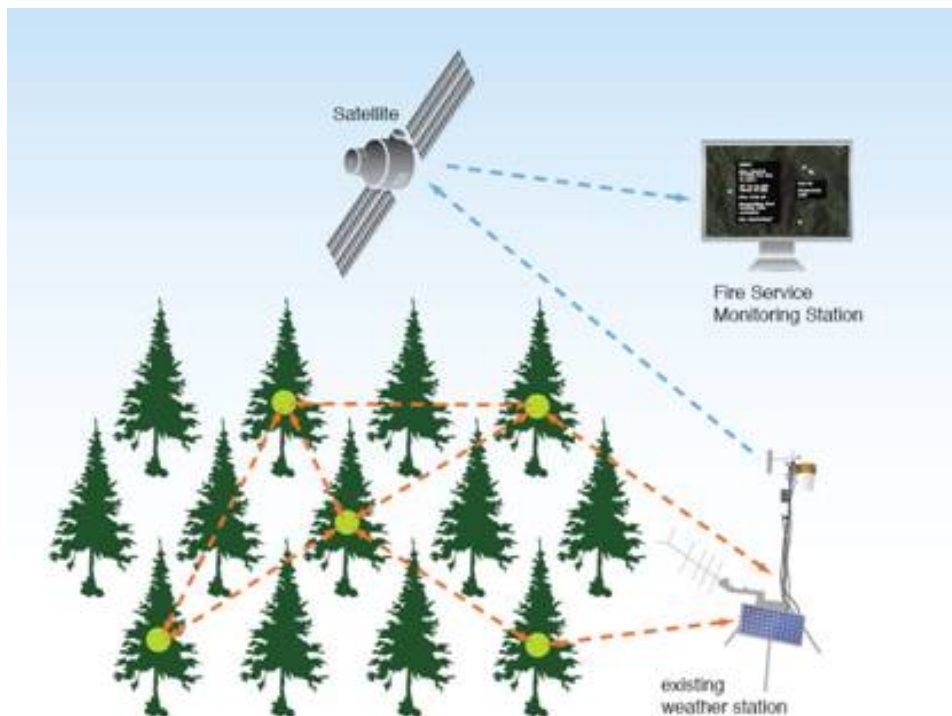
4.1.2. Szenzorhálózat alkalmazása

A vezetéknélküli szenzorhálózat elsődleges célja, hogy alacsony kiépítési és karbantartási költségekkel megfigyelhessünk valamit. A szenzorok pozíciója általában nem mérnöki megoldásokkal van kiszámítva, hanem véletlen- vagy közel véletlenszerűen helyezik el őket. A szenzorhálózat néhány meghatározó tulajdonsága [73]:

- nagyon sok node-ot tud kezelni;
- a node-okat sűrűn elhelyezhetjük;
- a node-ok könnyen elromolhatnak;
- a topológia sűrűn változhat;
- a node-ok processzorban, memóriában és energiában erősen limitáltak;
- a node-ok hálózati kommunikáció szempontjából nincsenek egyesével azonosítva.

Egy tipikus vezetéknélküli szenzorhálózati topológiát a 31. ábrán figyelhetünk meg, egy erdészeti tűzvédelmi alkalmazásban [74].

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK



31. ábra. Egy ténylegesen használt szenzor rendszer felépítése. Forrás: http://www.altenergymag.com/emagazine.php?issue_number=08.12.01&article=links

A vezeték nélküli szenzorhálózatok nagyon széles felhasználási területen alkalmazhatóak [73], mint például katonai, környezet-monitorozási, egészségügyi vagy akár háztartási felhasználás.

4.1.3. Vezeték nélküli szenzorhálózat szabványa

Az IEEE a helyi hálózatokkal és a városi hálózatokkal foglalkozó szabványainak egy csoportja (IEEE 802) 2006-ban kiadta [75] az alacsony átviteli vezeték nélküli személyi hálózatokra vonatkozó jelenleg érvényben lévő szabványát, amely ezen hálózatoknak a fizikai és MAC rétegre vonatkozik.

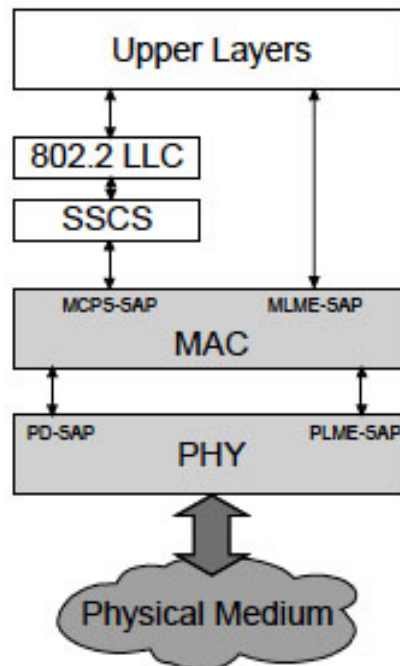
A szabvány a következőképpen karakterizálja ezeket a hálózatokat:

- alacsony átviteli sebességek (250 kb/s, 100kb/s, 40 kb/s és 20 kb/s)

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

- nagyon hosszú akkumulátor-élettartam (hónapok, akár évek)
- alacsony komplexitású rendszerek
- skálázhatóság a következő szempontok szerint:
 - átviteli sebesség
 - hatótávolság
 - energiafelhasználás

A szabvány által definiált rendszerarchitektúrát a 32. ábra illusztrálja.



32. ábra. Az IEEE 802.15.4 szabvány által leírt készülék architektúrája [5]

4.1.4. Szenzorok energiafelhasználása

Tekintettel arra, hogy a vezeték nélküli szenzorhálózat egyik legfontosabb tulajdonsága pont az, hogy a node-okban kicsi akkumulátor van, viszont keveset is fogyasztanak,

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

kiemelkedően fontos feladat az energiafelhasználásukat optimalizálni. Egy kutatás alapján [76] a node-ok átlagosan a következő arányban fogyasztják az energiájukat:

- 1% CPU - helyi feldolgozás
- 74% Szenzoreszköz - adatgyűjtés
- 25% Hálózati kommunikáció

Ebből is látszik, hogy nagyon komoly mennyiségű energiát lehet spórolni energiatudatos protokollok fejlesztésével.

4.1.5. Jelenlegi energiatudatos protokollok bemutatása

A következőkben jelenlegi, a vezeték nélküli szenzorhálózatokhoz használható, energiatudatos protokollokat [77] mutatunk be. Ezek közös jellemzője, hogy mindegyik célfüggvénye az, hogy a node-ok energiafelhasználását minimalizálja, és így az élettartamát maximalizálja.

Nagy általánosságban három kategóriát definiálunk az energiafelhasználás minimalizálására:

- Terheléselosztás (Duty Cycling)
- Adat-alapú megközelítés (Data-driven)
- Mobilis megoldások (Mobility)

A **Terheléselosztásos** megközelítés alapvetően két problémakört foglal magában: Topológia kialakítását és az Energia-felügyeletet. A Topológia esetében két fontos szempont van: A Node-ok pozíciója, valamint a Node-ok közötti kapcsolati rendszer. Az Energia-felügyelet esetében pedig a "Sleep/Wakeup" problematikájáról, valamint a hálózati kommunikációs réteg optimalizálásáról beszélünk.

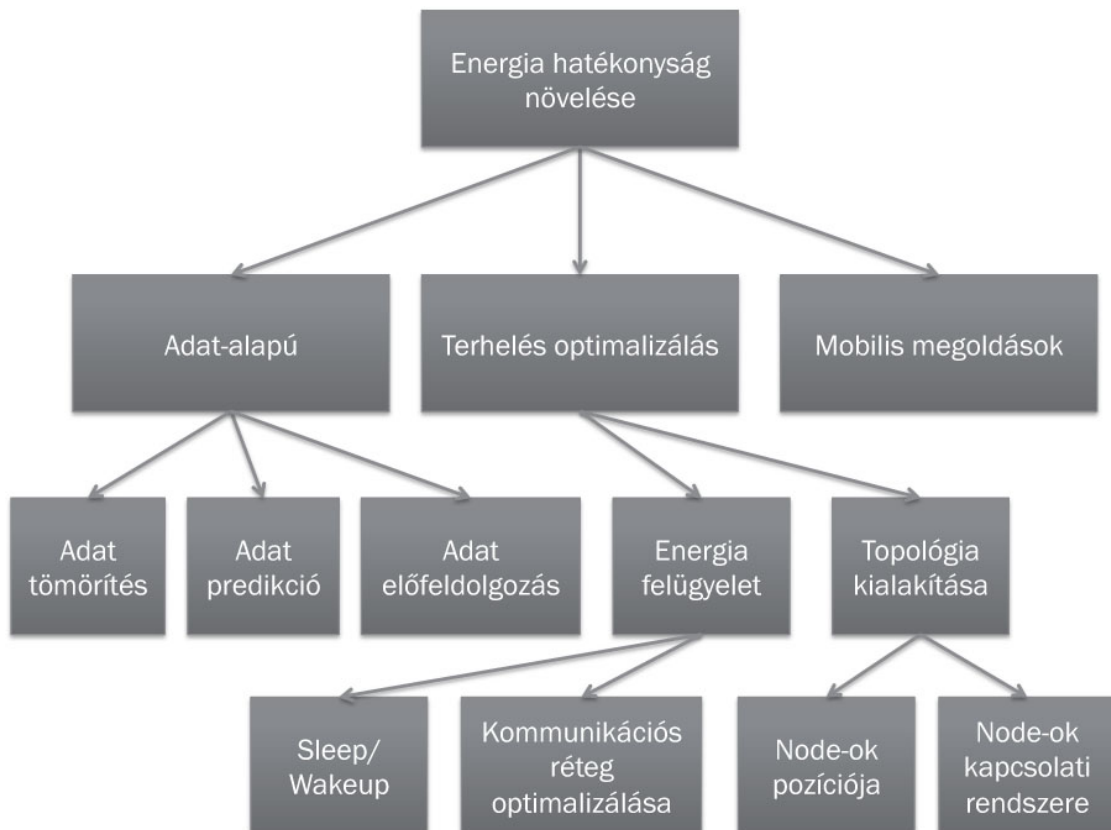
Az **Adat-alapú megközelítés** esetében megkülönböztethetjük az Energiahatékony adatgyűjtést valamint az Adatmennyiség-csökkentést - ez utóbbi leginkább az Adattömörítést, az Adatpredikációs algoritmusokat, valamint a Hálózaton belüli

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

adat-előfeldolgozást jelenti.

A **Mobilis megoldások** egy olyan megoldást jelentenek, amikor kihasználjuk, hogy a környezetben vannak amúgy is mozgó eszközök (pl. városban tömegközlekedési járművek), amelyek felszerelhetők egy Mobil bázisállomással, vagy Mobil továbbítóállomással.

A vezeték nélküli szenzorhálózatok energiahatékonyság-javításának megoldási irányait a 33. ábra szemlélteti.



33. ábra. Vezeték nélküli szenzorhálózat energiahatékonyságát javító módszerek

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

4.2. Modell és probléma felvetés

Célom egy olyan útvonalat találni, amely az energia szempontjából minimális, ugyanakkor a csomag megérkezési valószínűsége a bázisállomásra egy meghatározott érték felett marad. A hálózatot egy kétdimenziós gráf modellezi, melynek éleit v jelöli, ezek az élek a node-ok közötti rádiós csomagátvitelt reprezentálják. Minden élhez tartozik egy valószínűség, amely a küldő és fogadó node közötti sikeres csomagátvitel valószínűsége, amely a két node közötti távolság (d) és az adóteljesítmény (G) függvénye (Ψ). Ezt Rayleigh-fading modell alapján (4) írja le,

$$P_{i_j, i_{j+1}} = \Psi(d_{i_j, i_{j+1}}, G_{i_j, i_{j+1}}) = e^{-\theta * \sigma^2 * d_{i_j, i_{j+1}}^\alpha / G_{i_j, i_{j+1}} - G_0} \quad (4)$$

ahol

- θ az érzékenységi küszöb,
- σ a zaj energiája,
- P_r annak a valószínűségét definiálja, hogy az adott csomag a csomagtovábbítás során nem veszik el,
- d a távolság,
- G az adóenergia,
- G_0 a készülék belső konstans energiaigénye.

Látható, hogy ez a valószínűség az adó energiafüggvénye, amely egy diszkrét halmazból veszi fel az értékeit. Ha a csomag egy adott útvonalon halad végig, akkor a linkátvitel függetlenségét feltételezve a sikeres bázisállomásba-érkezés valószínűsége a következő:

Az egyes R útvonalakat úgy írom le, hogy az tartalmazza azt a V vektort, amelynek tartalma az útvonalban szereplő v -k, azt az E vektort, amely az egyes élekhez tartozó energiaértékeket adja, valamint a megfelelő élekhez tartozó távolságot.

$$R(V, E, d_{\in V}) \quad (5)$$

Így tulajdonképpen célom megtalálni az optimális R_{opt} -ot.

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

Modellemben olyan környezetet választottam, ahol a csomagok megérkezési valószínűsége vezeték nélküli csomagtovábbítás esetében leírható egy olyan függvénnyel (Ψ), amelynek paraméterei a távolság (d) és a küldési energia (g). Későbbiekben lehetőségem lesz bármilyen terjedési modellt alkalmazni a Ψ függvényre, például a legtöbbet alkalmazott Rayleigh Fading-modellt.

$$P_{u,v} = \Psi(d_{u,v}, g) \quad (6)$$

A probléma alapfeltétele az, hogy az útvonalon végighaladó csomag teljes megérkezési valószínűsége egy általunk, illetve a hálózat tervezője által definiált kritikus érték (ϵ) felett maradjon, így egy produktummal kifejezhetjük ezt a feltételt a következőképpen:

$$\prod_{u,v} \Psi(d_{u,v}, g) \geq 1 - \epsilon \quad (7)$$

Ezt a feltételt átvisszük összeges formára, így a következő logaritmikus skálázást fogjuk kapni:

$$\sum_{(u,v) \in R} -\lg(\Psi(d_{u,v}, g)) \leq -\lg(1 - \epsilon) \quad (8)$$

Tekintettel arra, hogy a jelenlegi alapfeltevésünk szerint az energiaértékek diszkrét értékeket (G_1, G_2, \dots, G_N) vehetnek fel, a probléma célfüggvénye az, hogy kényszeres optimalizálási feladatként minimalizáljuk a küldési energiát, ahol a kényszert a 8 jelenti.

$$\min(g), g \in \{G_1, G_2, \dots, G_N\} \quad (9)$$

Ezzel a célfüggvényt additív optimalizálási feltétellé alakítottam, úgy hogy az u értékhez negatív logaritmikus értékeket rendeltem.

4.3. Új algoritmusok

A következőkben saját fejlesztésű új algoritmusokat mutatok be az előbbieken leírt probléma megoldására.

4. VEZETÉKNÉLKÜLI SENZORHÁLÓZATOK

4.3.1. Minimális energiát garantáló algoritmus

A megoldás során kihasználom azt az alapfeltételt, hogy a csomópontok csak diszkrét energiaértékeket vehetnek fel, így a folytonos energiaszint-optimalizálás helyett 8-10 diszkrét energiaszintet kell vizsgálni. Ezzel a feltétellel gyorsan találhatunk az energia szempontjából optimális útvonalat. A feladatot a következő iterációs algoritmus oldja meg:

1. Lépés

Első fázisban az energiát a beállítható minimumértéknek veszem, és így keresek egy optimális útvonalat; a csomagtovábbítás teljes valószínűségére (9. képlet) vonatkozó kritériumot figyelmen kívül hagyom. Így a feladat a következőképpen alakul:

$$g_0 \rightarrow R_0 : \min \sum_{(u,v) \in R} -\lg(\Psi(d_{u,v}, g)) \quad (10)$$

Ez a minimum-keresés egy Dijkstra- vagy egy Bellmann-Ford-algoritmus segítségével polinomiális időn belül véghez vihető.

2. Lépés

Az iterációs lépés második fázisa az, hogy megvizsgálom, hogy a kapott érték vajon teljesíti-e a fent leírt feltételt, és ennek alapján közelítem a megoldást.

$$\sum_{(u,v) \in R} -\lg(\Psi(d_{u,v}, g)) \leq -\lg(1 - \epsilon) \quad (11)$$

Ha teljesíti, akkor

$$g_1 := g_0 \quad (12)$$

Ha nem, akkor

$$g_1 := g_0 + 1 \quad (13)$$

Ez a módszer addig emeli az energiaszinteket, amíg a feltételnek szabott kritériumnak eleget nem tesz. Így tulajdonképpen egy költségghatókony és gyors módszerhez jutok.

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

4.3.2. Módosított algoritmus

Az előbb bemutatott módszer egy egyenletesen elosztott hálózat esetén nagyon jó eredményt hoz, ugyanakkor nem veszi figyelembe azt, hogy a hálózat csomópontjai nem egyenletesen vannak elszórva a síkon, csomókat alkothatnak.

A másik probléma, hogy azt sem veszi figyelembe, hogy a bázisállomáshoz közeli node-ok szűk keresztmetszetté, azaz olyan csomóponttá alakulnak, amelyen minden forgalom keresztülhalad, így gyorsan elfogy az energiája. Ezért módosítjuk a módszert úgy, hogy figyelembe vesszük az energiaszinteket is. Mindig a legjobban töltött csomópont energiaszintjét növeljük, egészen addig, amíg vagy a feltételt nem teljesíti, vagy az energiaszintje nem különbözik túlságosan az alapszinttől.

Ezért definiálunk egy n egész számot, amely azt határozza meg, hogy mennyire növelhetjük az egyes node-ok energiaszintjét. Az alapenergiaszintet akkor emelem, ha már nem tudok emelni a node-ok energiáin. Figyeljük meg, hogy $n = 0$ esetében ez az algoritmus azonos az előzővel.

Az algoritmus a következő:

Algorithm 1 G - iterációs algoritmus

```

GlobalG := 0
loop
  G1..N := GlobalG
  repeat
    if Required Probability satisfied then
      return G
    end if
    Gx := Gx + 1 where Gx < GlobalG
  until exists G1..N < GlobalG
  GlobalG := GlobalG + 1
end loop

```

A módszert a szimuláció során G néven említem.

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

4.3.3. Kevés energiájú csomópontok kizárása

A második módosítás, egy másik megközelítése ugyanennek a problémának, hogy az energiaszinteket is figyelembe vesszük. A módosítás lényege, hogy kizárjuk a csomag-továbbítás folyamatából azokat a csomópontokat, amelyek átlag alatti energiaszinttel rendelkeznek. Ezek a csomópontok csak akkor küldenek csomagot, ha azok rajtuk generálódtak.

Algorithm 2 MinG - Kevés energiájú csomópontok kizárása

```

loop
   $T := \text{average value of } A_{1..N}$ 
   $I = \emptyset$ 
  for all Nodes do
    if  $A(\text{Node}) > T$  then
       $I+ = \text{Node}$ 
    end if
  end for
  RUN  $G(I)$ 
end loop

```

Ezzel a módszerrel nem terheljük az erőseket, hanem védjük a gyengéket.

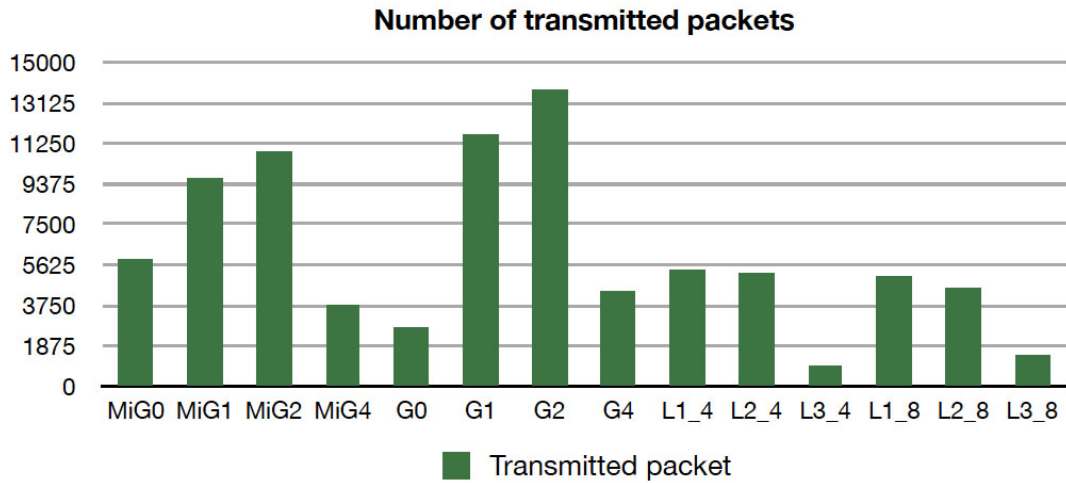
A módszert a szimuláció során MinG néven említtem.

4.4. Teljesítőképesség-analízis

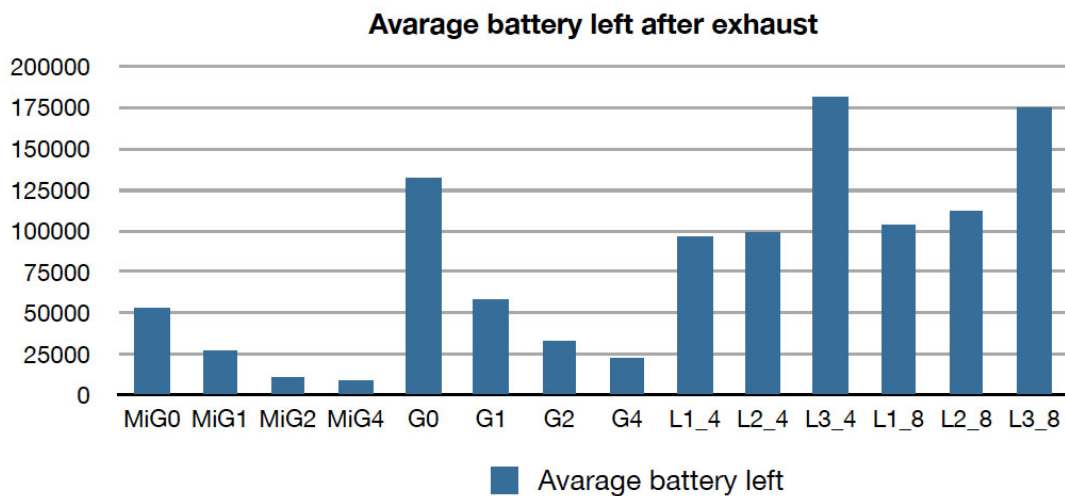
A következő ábrán láthatjuk azt, hogy hány ezer csomagot tudtunk továbbítani a hálózatokban, míg azok elérték a lemerülési kritériumot.

Jól látható, hogy a fix-G algoritmusaink többszörös eredményt mutatnak. Az eredmények azonban azt is megmutatják, hogy ha túl nagyra választjuk a fix-G algoritmusunkhoz az n paramétert, akkor a performancia visszaesik.

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK



34. ábra. A továbbított csomagok száma (x1000) a hálózat lemerüléséig a különböző algoritmusok használatával



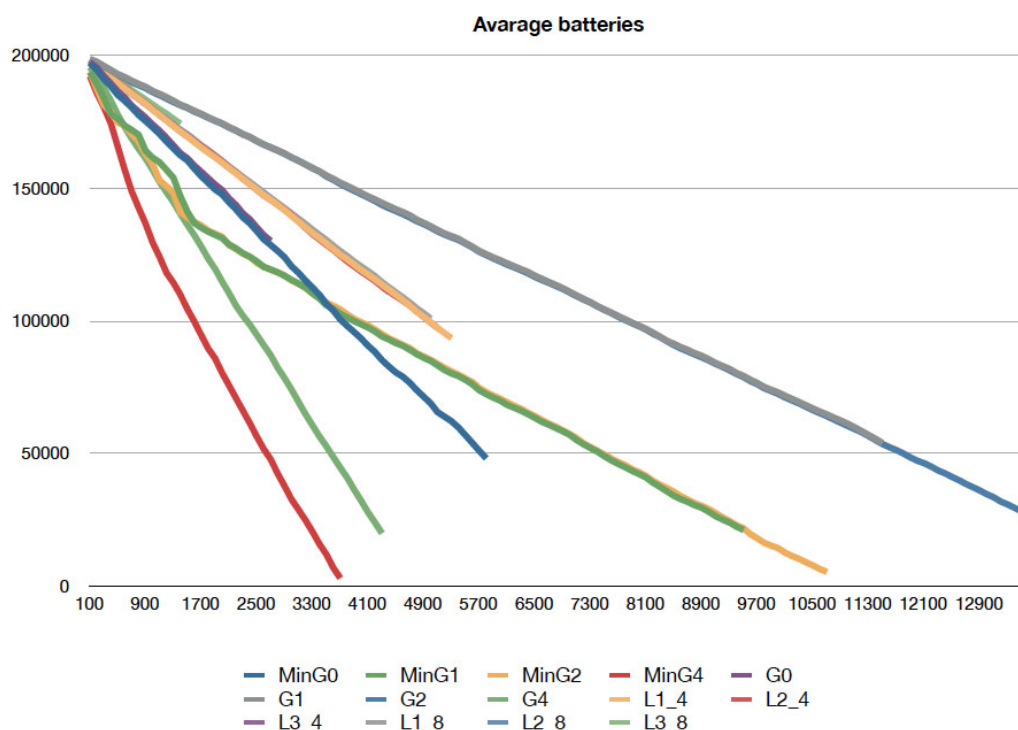
35. ábra. A lemerülést követően a hálózatban maradt átlagenergia

Ezen az ábrán az látható, hogy a MinG algoritmusok valóban arra törekednek, hogy teljesen egyenletesen osszák el az energia-terhelést, hiszen ezzel az algoritmussal maradt a legkevesebb felesleges energia a hálózatban. A 34. ábrával összevetve azonban

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

látszik, hogy a fő célt – a csomagok továbbítását – mégsem ez a módszer érte el a legjobban.

Ha megfigyeljük, látható, hogy nem feltétlenül az a hálózat teljesít jobban, amelyikben kevesebb energia marad a végére, hiszen a G2-algoritmussal használt hálózatban majdnem háromszor annyi energia maradt, mint a MinG4-ben, amely viszont már alig hárommillió csomag elküldése után lemerült, ellentétben a másikkal, amelyek több mint 13 millió csomagot tudott továbbítani.



36. ábra. Az átlag energiaértékek hálózatonként

A 36. ábrán látható, hogy hogyan alakul a legtöbb energiával, illetve a legkevessebbel bíró node sorsa. Látszik, hogy a MinG algoritmusok, miután észrevették, hogy vészesen fogy az egyik node energiája, nem hagyták, hogy azok a csomópontok részt vegyenek a kommunikációban, és ezért a minimum energia kisebb szögben csökkent egy idő után. Ezen kívül az is megállapítható, hogy azon hálózatok, amelyekben a max node energia elég magas, még rendelkeznek tartalékokkal.

4. VEZETÉKNÉLKÜLI SZENZORHÁLÓZATOK

4.5. Az elért eredmények jelentősége

Összességében elmondható, hogy az algoritmus segítségével elérhető, hogy

- a hálózat csomópontjai később merüljenek le
- a hálózat csomópontjai nagyjából egyszerre merüljenek le
- a hálózat kommunikációja egészen a lemerülésig az elvárt minőséget éri el

Az egyszerű, sima G verzió a gyors konvergencia érdekében egyszerre emeli a linkek energiaértékeit, így az optimálistól távoli kezdőállapotból is gyorsan halad az optimum felé, de az elért megoldás minősége csak közelíti a globális optimumot. A második algoritmus nem egyszerre emeli a linken energiaértékeit, így lassabb a konvergenciája, de jobb minőségű eredményt kapunk.

Így az elért eredmények például egy épületmonitorozási feladat esetén nagyban csökkentik a szenzorhálózat fenntartási költségeit, de például egészségügyi monitorozás esetén (idősek otthona, kórház) a kritikusnak számító adattovábbítási minőséget is garantálja.

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

5. Hálózatdimenzionálás csomagkapcsolt hálózatokban

A csomagkapcsolt kommunikációs technológiák másik fontos határa az adott forgalom kiszolgálására alkalmas access module hardvereinek száma és komplexitása. A gazdaságos telekommunikációs szolgáltatások nyújtásához minimális költséggel kell adott mennyiségű felhasználói forgalmat kiszolgálni. Ez előtérbe helyezi a hálózatdimenzionálás problémakörét, ahol a tervezéshez jó hatékonyságú optimalizálási módszerek szükségesek. Ebben a fejezetben a hierarchikus hozzáférési modul (Hierarchical Admission Module - HAM) tervezésére szolgáló eljárásokat vezetek be, amellyekkel a cellavesztési valószínűség előírt érték alatt tartható. Az eljárások bemenete a forgalom mennyisége és a szolgáltatás minősége (cellavesztési valószínűség), míg a kimenete a forgalom kiszolgálásához szükséges hardver komplexitása. A feladat a statisztikai elemzés után egy nagy állapotterű optimalizáláshoz vezet, amelynek részleteit az 5.5. illetve az 5.6. fejezet mutatja be.

Ebben a fejezetben célok a HAM optimális topológiájának kialakítása, ami előírt QoS biztosítása mellett minimális hardverkomplexitást eredményez. Az optimális architektúra megtalálásához a lehetséges topológiák és linkkapacitások teljes állapotterét kell végignézni, hogy megtaláljuk az optimális architektúrát. Ez a méretezést egy optimalizációs problémává alakítja, amelyre mostantól NCAP-ként (Node and Capacity Arrangement Problem) hivatkozunk. A tervezést két eszköz segítségével végezhethetjük el: a Nagy Eltérések Elméletének alkalmazásával megbecsüljük a forgalom farokeloszlását, valamint kombinatorikus optimalizációs algoritmusokat használunk.

5.1. Hálózat-dimenzionálás csomagkapcsolt hálózatokban

Csomagkapcsolt kommunikációs protokollok esetében a beengedett forgalom maximalizálása mellett nagy fontossággal bír az előírt szolgáltatási minőség biztosítása. Ez konkrét hálózati technológiák (pl. ATM) esetén a hálózattervezést (dimenzionálást) egy kényszeres optimalizálási feladatra képezi le, ahol a cél minimális hardverigényű hozzáférési architektúrák tervezése adott cellavesztési valószínűség mellett. Ezért a tézis eredményei a csomagkapcsolt hálózatok hozzáférési architektúrájának komplexitását optimalizálja a minimális költségű hardver megvalósítása érdekében. Ugyanakkor az architektúrának az előírt cellavesztési valószínűséget kell biztosítania. A fel-

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

adat tehát olyan dimenzionálási eljárás kidolgozása, amely képes az előírt felhasználói igények hozzáféréseinek hatékony menedzselésére.

5.2. QoS a csomagkapcsolt hálózatokban

Hagyományosan az Interneten működő routerek és LAN switch-ek "best effort" módon működnek. Ez azt jelenti, hogy a szolgáltatások minőségére (átviteli sebesség, válaszidő) semmiféle garanciát nem kaphatunk, ezek az éppen aktuális terheléstől függenek. [6, 9. o.]

Jelenleg széleskörben elterjedt multimédiás szolgáltatások, mint például az IPTV, vagy VOIP, vagy akár WEB-en keresztüli videók streamelése ugyanakkor megkövetelik azt, hogy az átviteli szolgáltatás minőségét garantálni lehessen. [78]

Több Layer 2-es szolgáltatás alkalmas arra, hogy ezt a szolgáltatás minőség garanciát (Quality of Service - QoS) kiszolgálja, ilyen például az ATM, vagy a Frame-Relay, ugyanakkor a legszélesebb körben elterjedt módja a QoS biztosításának az Ethernet protokoll IEEE 802.1p vagy IEEE 802.1Q módozata. [6, 16. o.]

A hálózati forgalmak, illetve szolgáltatások három kategóriába sorolhatóak:

- "A" Típus: Real-Time szolgáltatás: szigorú késleltetési kritériummal (pl. VoIP)
- "B" Típus: Interaktív és Streamelt adatszolgáltatások: elviselnek a szolgáltatások némi késleltetést, de az átviteli sebesség kiemelkedően fontos a felhasználói élmény biztosítása miatt (pl. Web-böngészés).
- "C" Típus: Késleltetéstűrő szolgáltatások: Nagyobb késleltetést is elvisel anélkül, hogy ez a Quality of Service (QoS) kárára menne (pl. fájl- vagy email-továbbítás).

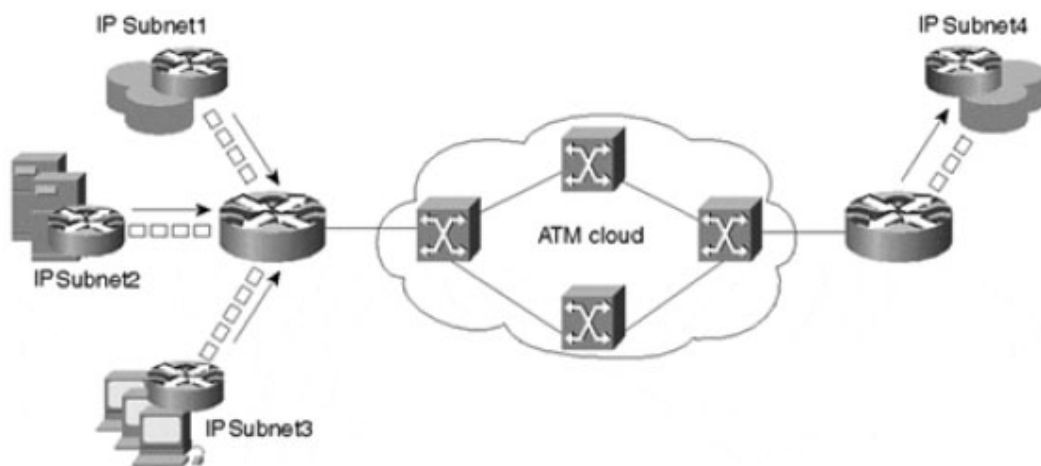
Ugyanakkor a hálózattervezés egy kiemelkedően fontos szempontja a hálózat minőségi paramétereinek az elvárt szinten tartása. A QoS-t csomagkapcsolt hálózatok esetében elsődlegesen a következő három alapján definiáljuk:

- Csomagvesztés: annak a valószínűsége, hogy a csomag ütközés miatt nem ér célba, eldobódik

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

- Blokkolás valószínűsége: annak a valószínűsége, hogy az engedélyezési réteg visszautasítja a bejövő kéréseket
- Átviteli sebesség: egy adott idő alatt átvitt adat mennyisége (pl. kb/s)

A különböző Layer 2-es technológiák különböző módokon oldják meg, hogy a minőséget garantálni tudják. Külön technológiák léteznek arra, hogy olyan hálózatokban, ahol többféle megoldás található, mint például IP és ATM (lásd 37. ábra), hogyan kezeljék a szolgáltatások minőségi előírásainak teljesítését [6, 135.o.].



37. ábra. IP hálózati forgalom továbbítása ATM Cloudon keresztül [6]

5.2.1. QoS teljesítőképességi mértékei

A QoS alkalmazása biztosítani kíván előírt teljesítőképességi paramétereket a hálózatban. A számítógépes hálózatokban a legelterjedtebb kapcsolati mértékek a következők [6, 12.o.]:

- Adatátviteli sebesség (Bandwidth)
- Csomagvesztés (Packet Loss)
- Átviteli késleltetés (Packet Delay)

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

- Késleltetés-ingadozás (Packet Jitter)

Adatátviteli sebesség:

Adatátviteli sebességnek hívjuk a médium, protokoll vagy kapcsolat névleges átviteli sebességét. Hatékonyan írja le azt, hogy egy alkalmazásnak a hálózaton keresztüli kommunikációhoz mekkora méretű "csőre" van szüksége. Általánosságban minden olyan alkalmazásnak, amelyiknek garantált szolgáltatást kell nyújtania, van egy adatátviteli sebesség igénye, és szeretné, hogy a hálózat egy meghatározott minimális sávszélességet folyamatosan biztosítson csak neki. Például egy 64-kbps adatfolyamú digitális hang közvetítő alkalmazás gyakorlatilag használhatatlan abban az esetben, ha a hálózati útvonal bármelyik részén tartósan kevesebb mint 64 kbps adatátvitel áll rendelkezésre. Az ilyen alkalmazás a hálózattól adatátviteli sebesség garanciát kérhet.

Csomagvesztés:

Csomagvesztés határozza meg azt a számot, hogy hány csomagot veszünk el a hálózati kommunikáció során. Csomagvesztés történhet torlódás vagy hibás továbbítás miatt. Tipikusan akkor történik csomagvesztés, amikor a bejövő csomagok száma messze meghaladja a kimenő sor puffert, vagy eleve a bejövő pufferméret alacsony. A csomagvesztést általában meghatározott mennyiségű csomag egységnyi idő alatti továbbítása során elveszett csomagok számaként definiáljuk. Vannak olyan alkalmazások, amelyek egyáltalán nem, vagy nagyon hatékonytalanul működnek, ha csomag elveszik. Az ilyen csomagvesztést nem tűrő alkalmazások a hálózattól csomagvesztési garanciát kérhetnek.

$$p = \frac{N_{\text{lost}}}{N_{\text{lost}} + N_{\text{received}}} \quad (14)$$

Átviteli késleltetés:

A csomagtovábbítás során minden egyes hopnál keletkezik továbbítási (transmission, serialization), terjedési (propagation), kapcsolási (switching) késleltetés. A továbbítási késleltetés (D_t) az az idő, amíg csomagot az első bittől az utolsóig az eszköz megkapja. Ez nyilvánvalóan függ az adott eszköz sávszélességétől. Egy 64

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

byte-os csomagot például 3Mbps-s eszközön 0,171ms alatt lehet továbbítani, ugyanezt a csomagot egy 19.2 kbps-s kapcsolaton keresztül 26ms időbe telik.

$$D_t = N/R \quad (15)$$

ahol N a bitek száma, és R az átviteli sebesség.

A terjedési késleltetés (D_p) az az idő, amíg a küldő oldalról a fogadó oldalig elér az információ. Ez a továbbító médium és a távolságon múlik. Ez komoly időt is igénybe vehet, tekintettel arra, hogy a fénysebesség felső korlátot szab a megoldásoknak. Egy transzatlanti terjedési késleltetés 30ms-os nagyságrendben van.

$$D_p = d/s \quad (16)$$

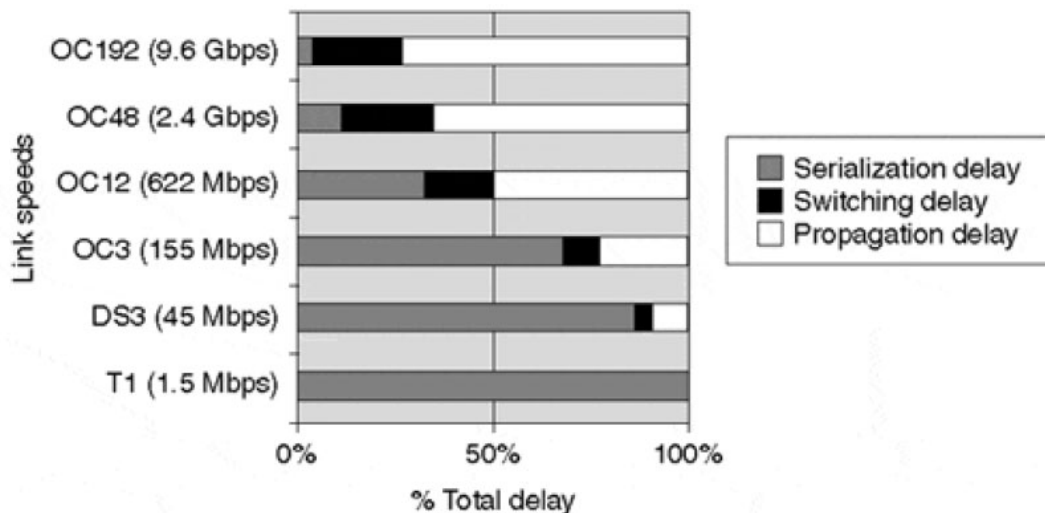
ahol d a távolság, és s a hullám terjedési sebessége.

A kapcsolási késleltetés (D_s) az az idő, amíg az eszköz továbbítja a csomagot azután, hogy megérkezett. A nagy sávzélességű interfészeknél a továbbítási késleltetés elhanyagolhatóan kicsivé válik a kapcsolási és a terjedési késleltetésekhez képest. (lásd 38. ábra)

A teljes késleltetést (D) a fent leírt három késleltetés összege adja meg.

$$D = D_t + D_p + D_s \quad (17)$$

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN



38. ábra. Késleltetés megoszlása egy 1500byteos csomag transzatlanti továbbítása esetén, különböző sávszélességekkel [6, 13. o]

Késleltetés-ingadozás:

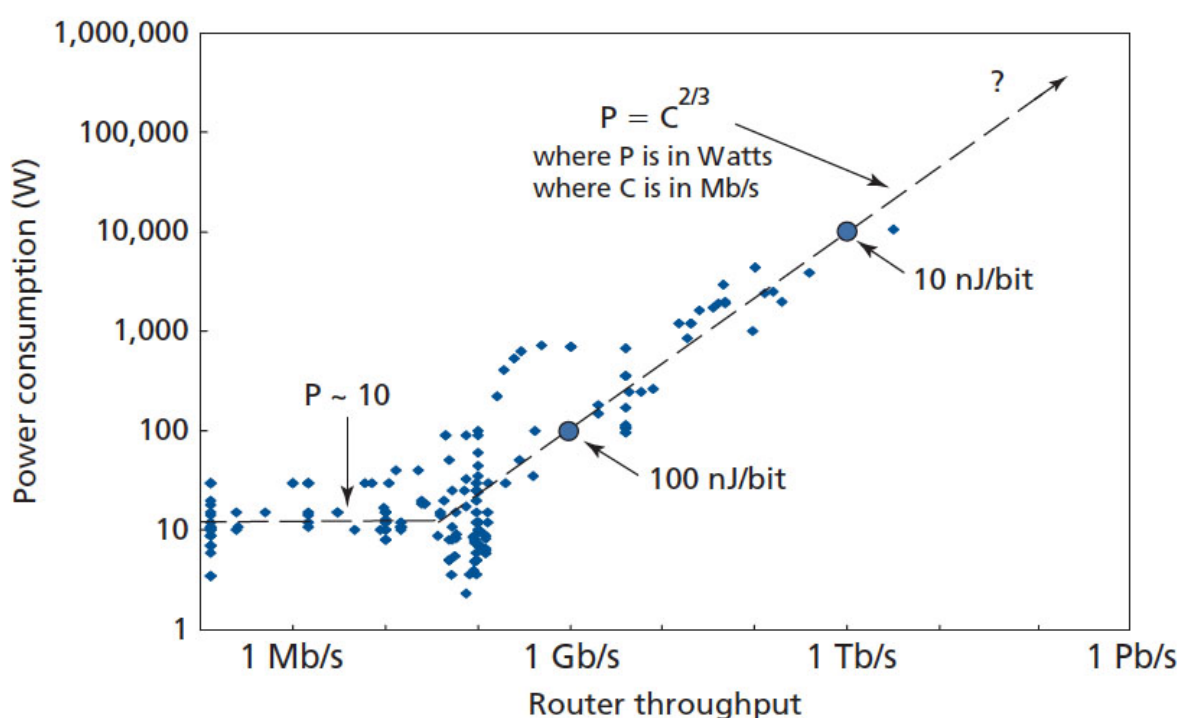
Ha a hálózaton torlódás alakul ki, a sorbanállási késleltetés a teljes csomag-továbbítás idejében növekedést okoz és hozzájárul ahhoz, hogy egy kapcsolathoz tartozó csomagok különböző idő alatt érkeznek meg. Ezt az eltérést nevezzük késleltetés-ingadozásnak. Ez azért fontos, mert meg lehet becsülni vele a maximális időközöket, amik eltelhetnek az egyes csomagok beérkezése között. A fogadó fél, alkalmazástól függően, beállíthat egy késleltetés-ingadozási puffert, hogy kiegyenlítse ezeket a lyukakat. Ilyen alkalmazások, amik folyamatos adatfolyamot igényelnek, például a VoIP, a videókonferencia vagy a videó megosztó alkalmazások. A késleltetés ingadozást a véletlen késleltetés szórása adja meg.

5.3. A hálózat-dimenzionálás fontossága

Már a legelső csomagkapcsolt hálózatok esetében felmerültek a dimenzionálási problémák. Az egyes eszközök áteresztőképességei és a felhasználók igénye gyakorta nem találtak, ezért már a korai hálózatok esetében kiemelkedően fontos volt a méretezés [18].

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

A jelenlegi hálózatokon, köszönhetően a nagy számú felhasználói igénynek, valamint a munkák, feladatok elektronikussá válásának, a csomagkapcsolt hálózatok megfelelő méretezése egyre fontosabb. A kiemelkedően magas igények miatt egyre több routert, eszközt helyeznek el például az Interneten. A sávszélesség növekedésével a routerek energiafogyasztása is drasztikusan emelkedik (lásd 39. ábra [19]).



Source data: METI, 2006, Nordman, 2008

39. ábra. A hálózati eszközök energiafogyasztása a sávszélesség függvényében.

Egy Japán kutatás szerint [79] 2015-re teljes Japán energiafogyasztásának 10%-át routerek energiafogyasztása fogja kitenni, és ez a szám 2020-ra 50%-ra nő. Ennek a folyamatnak az eredményeként, azon túl, hogy egy komoly költségtenyezővel kell számolnunk a következő évek informatikájában, komoly ellentét mutatkozik az energiatudatos és fenntartható fejlődés irányelveivel.

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

A leírtak alapján egyértelműen látszik, hogy paradigmaváltásra van szükség a routerek jelenlegi kialakítási módszereiben, nagyobb figyelmet kell szentelni a hálózatok méretezésére. Ugyanakkor nincsen egységes, kialakult módszer arra, hogy milyen keretek között, milyen mérőszámokkal, milyen procedúrán keresztül ajánlott a méretezést elvégezni. Jól mutatja ezt, hogy többek között a Cambridge Egyetemen Frank Kelly az IBM-mel karöltve igyekszik megalkotni a módszertant IPCP (IP Capacity Planning) néven, amely számos csomagkapcsolt hálózat (3G, ATM, MPLS, IP) tervezésénél használható keretrendszer alkot[80].

Számos kutatás foglalkozik ezzel a témával [81, 82], statisztikai alapon, valós forgalom felhasználásával, ugyanakkor nincsen egy olyan módszertan, amely szerint a méretezést elvégezhetnénk úgy, hogy közben energiatudatosságot is figyelembe vehetnénk.

A routerek közül az access típusú, azaz a hozzáférést biztosító eszközök azok, amelyek a processzorigényes elsődleges QoS-t biztosítják a szélessávú internet előfizetők részére, így a routerek energiafogyasztása tekintetében kiemelkedően fontos ezek hardware komplexitásának optimalizálása.

A dimenzionálási feladat tehát az elvárt szolgáltatás biztosítása a felhasználóknak minimális eszköz és erőforrás használatával úgy, hogy az elvárt minőségi paramétereknek a szolgáltatás megfeleljen.

5.4. A modell

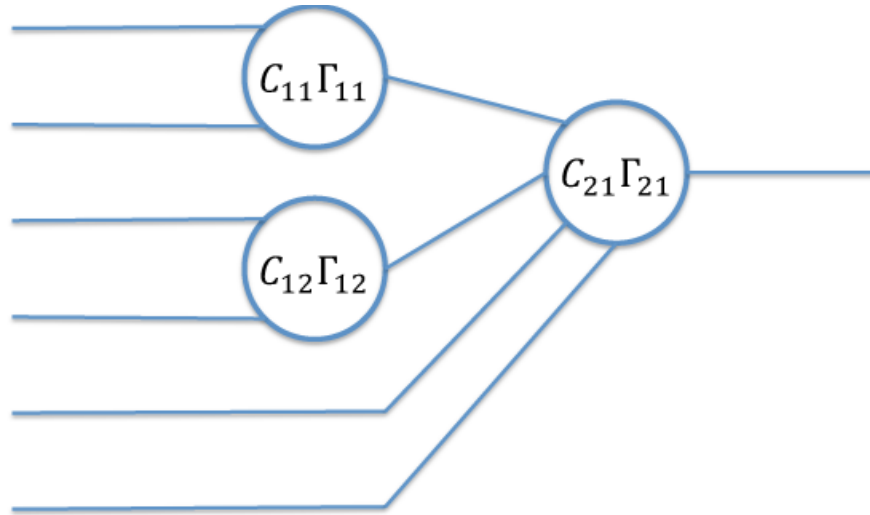
Tipikus Internet-elérési esetben a szolgáltató a felhasználókat a következő 3 forgalomosztályba sorolja:

- Internet Access1;
- Internet Access2;
- Voice over ADSL.

A forgalmi osztályokhoz a felhasználók tekinthetők ON/OFF forrásnak, rendre m_1, m_2, m_3 átlaggal és h_1, h_2, h_3 csúcserővel. A $\gamma_1, \gamma_2, \gamma_3$ QoS-paraméter azt határozza meg, hogy mekkora cellavesztést (Cell Loss Probability - CLP) engedünk meg.

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

A fentieknek megfelelően a méretezés várt kimenetelét az alábbi ábra mutatja:



40. ábra. A méretező-algoritmus kimeneteként kapott struktúra

Az ábra alapján a HAM node-ok egy halmazának tekinthető, amelyek fa-topológiában vannak elhelyezve. A méretező-algoritmus meghatározza a topológiát, valamint minden node-hoz linkkapacitásokat és QoS-paramétereket rendel. Ez alapján a HAM formálisan a következő módon írható le:

$$HAM = \{V, E, C, \Gamma\}, \quad (18)$$

ahol a V a csúcsokat, E az éleket jelenti, amíg C és Γ a kapacitások és QoS-paraméterek mátrixait jelöli a következő módon:

$$C_{kj} = C_j(k) \quad (19)$$

a kapacitása a j . node-nak a k -ik rétegben.

$$\Gamma_{kj} = \gamma_j(k) \quad (20)$$

az elvárt QoS-érték a j . node-nak a k -ik rétegben.

Megjegyzendő, hogy amikor a HAM sok node-ból áll össze, a cellák bármelyik eszközön elveszhetnek, ezért ilyenkor szigorúbb CLP-követelményt kell előírni az

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

egy node-okon. Ennek eredményeként valóban kihívás az aggregált CLP-érték node-okra vonatkozó dekompozíciója.

A következő jelöléseket fogjuk használni a HAM leírásához:

- forgalomosztályok: $i = 1, \dots, M$;
- rétegek a fatopológiában: $k = 1, \dots, K$;
- node-ok a k -ik rétegben: $l = 1, \dots, L_k$;
- engedélyezési vektor a j -ik node-hoz a k -ik rétegben: $\mathbf{n}^j(k)$, ahol az $n_i^j(k)$ komponens az i osztálybeli források számát jelöli;
- az engedélyezhető vektorok egy halmazát Admission Set (AS)-nek nevezzük, amely a forgalomvektorokat tartalmazza az adott node-hoz a fatopológiában, a következők szerint:

$$\text{AS} = \{ \mathbf{n}^l(k) \mid \forall l = 1, \dots, L_k \forall k = 1, \dots, K \}; \quad (21)$$

- a bemeneti forgalom-állapotvektort a következőképpen írjuk le:

$$\mathbf{v}(1) = (\mathbf{n}^1(1), \mathbf{n}^2(1), \dots, \mathbf{n}^{L_1}(1)) \quad (22)$$

Megjegyzendő, hogy az AS a bemeneti állapotvektorral összefüggésben áll, mégpedig olyanformán, hogy mindegyik bemeneti állapotvektor felbontható egy AS-re, még hozzá a következő formában:

A bemeneti állapotvektor dekompozíciója:

A bemeneti állapotvektor $\mathbf{v}(1) = (\mathbf{n}^1(1), \mathbf{n}^2(1), \dots, \mathbf{n}^{L_1}(1))$ dekompozícióját egy AS-ra, a folyamatábra alapján, a következőképpen definiáljuk:

$$n_i^l(k) = \sum_{j \in A_l} n_i^j(k-1), \quad (23)$$

ahol A_l azon node-ok halmaza $k-1$. rétegben, amelyek kapcsolatban állnak l -ik node-dal a k . rétegben.

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

Egyértelmű, hogy a fenti definíció tekinthető egy $V \rightarrow AS$ -leképezésnek, ahol a bemenet a bemeneti állapotvektor $\mathbf{v}(1)$, és a kimenetet nevezzük $AS(\mathbf{v}(1))$ -nek.

A HAM-t reprezentáló adatstruktúra a következőképpen néz ki: A HAM mint egy gráf $G\{V, E, C, \Gamma\}$ teljes mértékben leírható a C és a Γ mátrixokkal. Ezen mátrixok segítségével mind a topológia és hozzákapcsolódó kapacitási beállítások $\{C_l(k), l = 1, \dots, L_k, k = 1, \dots, K\}$, mind a QoS-paraméterek $\{\gamma_l(k), l = 1, \dots, L_k, k = 1, \dots, K\}$ visszaállíthatók.

A topológiát egy G mátrix írja le:

$$G_{kl} = \begin{cases} 1 & \text{ha van node a } k\text{-ik réteg } l\text{-ik pozíciójában} \\ 0 & \text{egyébként} \end{cases} \quad (24)$$

A HAM QoS-beállításait egy adott G topológia esetében Γ^G mátrixszal jelöljük, ahol a kl elem a k -ik rétegben található l -ik node QoS-paraméterét jelenti. Ha a k -ik réteg l -ik pozíciójában nincsen node, akkor a $\Gamma_{kl}^G = 0$, ami azt jelenti, hogy ha $G_{kl} = 0$, akkor $\Gamma_{kl}^G = 0$. Továbbá feltételezzük, hogy a QoS lehetséges értékei egy diszkrét $\gamma_1, \dots, \gamma_V$ halmazt alkotnak. Ezért a $— = \{\Gamma_{min}, \dots, \Gamma_{max}\}$ halmaz tartalmazza a lehetséges QoS-mátrixokat. A Γ_{min}^G mátrixok $\Gamma_{kl} = \text{Min}$ -nek, míg a Γ_{max}^G mátrixot $\Gamma_{kl} = \text{Max}$ -nak definiáljuk, ahol Min és Max korábban meghatározott értékek. Ebben a formában a megfelelő QoS-séma kiválasztásához a méretező algoritmusnak minden egyes node-hoz ($l = 1, \dots, L_k$ and $k = 1, \dots, K$) végig kell mennie a $G_{kl} \in (\text{Min}, \text{Max})$ intervallumon.

A HAM kapacitás-hozzárendelését a C mátrixszal fejezzük ki. Megjegyzendő, hogy ha $G_{kl} = 0$, akkor $C_{kl} = 0$, ami azt jelenti, hogy a topológiában kapacitás csak létező node-hoz rendelhető. A G topológiához tartozó lehetséges kapacitás-mátrixot C^G -vel jelöljük (ahol $C_{ij}^G \in \{C_1, \dots, C_R\}$). Ezek a mátrixok $\mathcal{C}^G = \{C_{min}^G, C_2^G, \dots, C_{max}^G\}$ diszkrét teret alkotnak, ahol $C_{min}^G : C_{ij} = C_1 G_{ij} \forall i, j$ a minimális kapacitású hálózati topológiát jelenti, míg $C_{max}^G : C_{ij} = C_R G_{ij} \forall i, j$ ugyan ezen topológia maximális kapacitású node-jait. Tekintettel arra, hogy a lehetséges kapacitások száma véges, a programozó a \mathcal{C}^G halmazt bármilyen szabály alapján rendezni tudja. (Jelen adaptációban a rendezési séma az elemek összegétől és a megfelelő

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

mátrixon belüli indexek rangjától függ.)

Ezen adatstruktúrák segítségével a méretező-algoritmus teljes egészében leírható.

5.5. Új dimenzionáló algoritmusok bemutatása

A következőkben új algoritmusokat mutatok be az előbbieken bemutatott probléma megoldására. Annak érdekében, hogy a méretezési feladatot megfoghassuk, először egy egynode-os méretezési módszertant írok le. Ebben az esetben a feladat az, hogy minimális kapacitású eszközzel ki tudjam szolgálni a forgalmi igényeket az előre megadott CLP mellett. A megoldás a Chernoff határt és a logaritmikus momentum generáló függvényeket használja fel.

Chernoff határ

Chernoff határ az egyik alapeszköz a nagy eltérések elméletében. [11, 83] Ezzel a határral farokeloszlást tudunk becsülni, azzal a kitételrel, hogy Y valószínűségi változó csak pozitív értékeket vehet fel [11, 84, 85]:

$$P(Y > C) \leq e^{\mu_Y(s) - sC}. \quad (25)$$

Ebben az esetben Y logaritmikus momentum generáló függvénye $\mu_Y(s) = \log E(e^{sY})$ [83, 11], amit szokás hatásos sávszélességnek is hívni (elosztva s -sel), úgy hogy s kötelezően pozitív érték. A legélesebb határt úgy kapjuk, hogy

$$s_{opt} : \inf_s \mu_Y(s) - sC. \quad (26)$$

választjuk. Ahhoz, hogy használni tudjuk a Chernoff határt feltételezzük, hogy $Y = \sum_{i=1}^M \sum_{j=1}^{n_i} X_j$ az aggregált terhelés, ahol $X_j^{(i)}$ az i . osztályhoz tartozó j . forrás véletlenszerű forgalom terhelése. Így a Chernoff határt a következőképpen tudjuk felírni:

$$P\left(\sum_{i=1}^M \sum_{j=1}^{n_i} X_j^{(i)} > C\right) \leq e^{\sum_{i=1}^M n_i \mu_i(s) - sC}, \quad (27)$$

ahol az i osztályhoz tartozó On/Off forrás logaritmikus momentum generáló függvénye a következő:

$$\mu_i(s) = \log\left(1 - \frac{m_i}{h_i} + \frac{m_i}{h_i} e^{sh_i}\right) \quad (28)$$

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

Ennek eredményeképpen a QoS γ szintjét úgy biztosíthatjuk, hogy garantáljuk, hogy

$$e^{\sum_{i=1}^M n_i \mu_i(s_{opt}) - s_{opt} C} < e^{-\gamma} \quad (29)$$

vagy logikusan

$$\sum_{i=1}^M n_i \mu_i(s_{opt}) < s_{opt} C - \gamma. \quad (30)$$

teljesüljön, ahol

$$s^* : \min_s \sum_{i=1}^M n_i \mu_i(s) - sC$$

Először egy egynodeos tervező algoritmust mutatok be a koncepció megvilágítása érdekében.

Egynode-os optimalizációs algoritmus

Rendelkezésre áll egy $\mathcal{C} = \{C_1, \dots, C_R\}$ $C_1 < C_2 < \dots < C_R$ diszkrét kapacitáshalmaz, egy bemeneti forgalom, amit $\mathbf{n} = (n_1, \dots, n_M)$ forgalmi konfigurációs vektor fejez ki, valamint egy γ CLP-szint, mint QoS-paraméter.

Legyen $C := C_1$ és $r := 1$.

Számoljuk ki a logaritmikusan generáló $\mu_i(s)$ $i = 1, \dots, M$ függvényeket.

1. Határozzuk meg $s_{opt} : \inf_s \sum_{i=1}^M n_i \mu_i(s) - sC$
2. Nézzük meg, hogy $\sum_{i=1}^M n_i \mu_i(s_{opt}) < s_{opt} C - \gamma$ teljesül-e.
3. Ha IGEN, térjünk vissza C -vel, ha NEM, legyen $r := r + 1$ és térjünk vissza az 1. lépéshez.

Ezzel az algoritmussal megtalálhatjuk a minimális kapacitású C_{min} -t, ami elég hatékony ahhoz, hogy biztosítson az \mathbf{n} terhelési vektorhoz megfelelő kapacitást γ QoS-szinthez.

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

5.6. Új többnode-os dimenzionálási algoritmus

A dimenzionálási feladat célja, hogy találjunk egy olyan topológiát, megfelelő kapacitásokkal, amely egy az egész rendszerre vonatkozó QoS feltételt teljesít a megfelelő terhelés-vektorra. Ezért megadunk egy leképezést (Ψ) bemeneti terhelés-vektor $v(1)$, a QoS γ paraméter és a $G_{opt}(V, E, C, \Gamma)$ között. Így az optimalizálási probléma formálisan a következőképpen írható le:

$$G_{opt} \{V, E, C, \Gamma\} = \Psi(v(1), \gamma); \quad (31)$$

ahol

$$G_{opt}(V, E, C, \Gamma) : \min_{G(V, E, C, \Gamma)} \sum_{k=1}^K L_k. \quad (32)$$

Észrevehető, hogy számos Γ mátrix kielégíti a QoS feltételeket, így nem csak a kapacitásokat, hanem a QoS paramétereket is érdemes jól beállítani, hogy a node-ok között el legyen osztva. Ez a gondolat adja a rekurziós algoritmus alapötletét: Kezdjünk egy minimális konfigurációval, $G\{V, E, C, \Gamma\}$, és nézzük meg, hogy teljesíti-e a QoS kritériumokat. Ha nem, akkor bővítjük a konfigurációt egészen addig, amíg a $v(1)$ bemeneti vektorra a QoS ill. a CLP követelmények teljesülnek. Tekintettel arra, hogy a minimális konfigurációval indítjuk az iterációt, az optimális megoldást fogjuk megkapni.

Többnode-os optimalizációs algoritmus

Az első réteg minden node-jának bemeneténél rendelkezésre áll egy

$$v(1) = (n^1(1), n^2(1), \dots, n^{L_1}(1)) \quad (33)$$

forgalomkonfiguráció, és egy \mathbf{T} logikai változókat tartalmazó mátrix, melynek minden T_{kl} eleme azt jelzi, hogy a k -ik réteg l -ik node-ján levő helyi QoS-kritériumnak az adott node megfelel-e.

Az U változó azt jelöli, hogy az egészében vett QoS-kritériumnak megfelel-e.

1. Legyen $\mathbf{G} = \mathbf{G}_{min}$;

2. Dekomponáljuk a $v(1)$ -et $AS(v(1)) = \{n^l(k), l = 1, \dots, L_k, k = 1, \dots, K\}$ AS-

 5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

sé;

3. Legyen $\mathbf{C}^G := \mathbf{C}_{max}^G$;

4. Legyen $\mathbf{\Gamma}^G := \mathbf{\Gamma}_{max}^G$;

5. Számoljuk ki a $s_{l\ opt}(k)$ -t megoldva a

$$s_{l\ opt}(k) : \sum_{i=1}^M n_i^l(k) \frac{d\mu_i(s)}{ds} = C_l(k) \quad \forall l = 1, \dots, L_k \quad k = 1, \dots, K$$

egyenletet.

6. Ha

$$T_{lk} = \begin{cases} \text{IGAZ} & \text{if } \sum_{i=1}^M n_i^l(k) \mu_i(s_{l\ opt}(k)) < s_{l\ opt}(k) C_l(k) - \gamma_l(k) \\ \text{HAMIS} & \text{if } \sum_{i=1}^M n_i^l(k) \mu_i(s_{l\ opt}(k)) > s_{l\ opt}(k) C_l(k) - \gamma_l(k) \end{cases} \quad (34)$$

7. Számoljuk ki az $U := \bigcap_{k=1}^K \bigcap_{l=1}^{L_k} T_{kl}$ értéket.

8. Ha $U = \text{HAMIS}$, növeljük a topológiát azáltal, hogy beállítjuk $\mathbf{G} := \mathbf{G}_2$ -t, visszatérünk a 3. lépéshez és addig ismételjük ezt, amíg $U = \text{IGAZ}$

9. Ha $U = \text{IGAZ}$, akkor találtunk topológiát, de azzal lazítunk a QoS-követelményeken, hogy másik $\mathbf{\Gamma}^G \in -^G$ -t választunk a $\mathbf{\Gamma}^G := \mathbf{\Gamma}_2$ beállításával, visszatérünk az 5. lépéshez és addig ismételjük ezt, amíg $U = \text{HAMIS}$ vagy $\sum_{k=1}^K \sum_{l=1}^{L_k} e^{-\gamma_l(k)} > e^{-\gamma}$ majd visszatérünk $\mathbf{\Gamma}^G$ előző értékéhez.

10. Ha $U = \text{IGAZ}$, akkor találtunk topológiát, de azzal csökkentjük a kapacitást, hogy másik $\mathbf{C}^G \in \mathcal{C}^G$ -t választunk a $\mathbf{C}^G := \mathbf{C}_2$ beállításával, visszatérünk a 4. lépéshez és addig ismételjük ezt, amíg $U = \text{HAMIS}$, majd visszatérünk \mathbf{C}^G előző értékéhez.

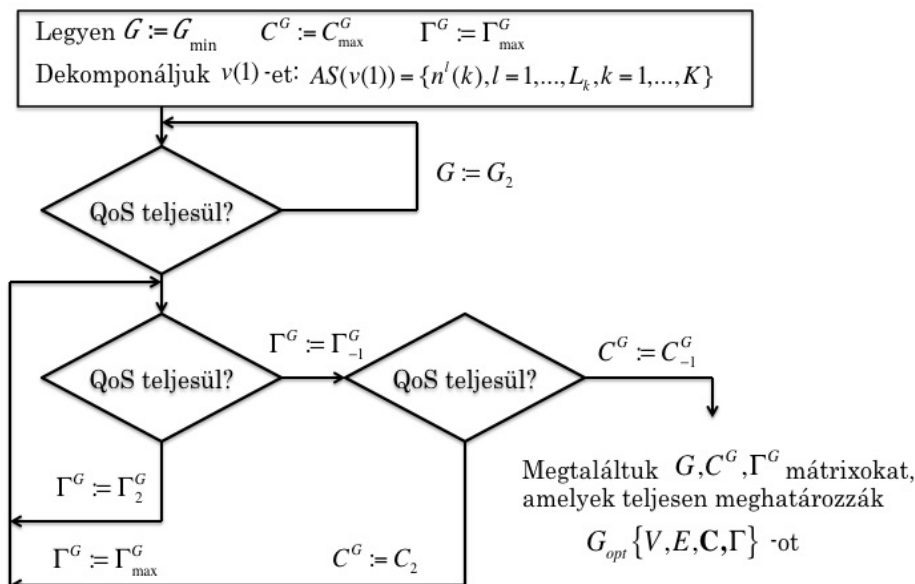
11. Visszatérünk a $\mathbf{G}, \mathbf{C}^G, \mathbf{\Gamma}^G$ mátrixokkal, amelyek teljesen meghatározzák $G_{opt} \{V, E, \mathbf{C}, \mathbf{\Gamma}\}$ -ot.

Látható, hogy az algoritmus az optimális HAM-t úgy adja vissza, hogy nem csak a node-ok száma minimális (megtalálva a legkisebb topológiát), hanem ezzel együtt a

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

hozzá tartozó kapacitás és QoS-beállítások is. Így megtalálhatjuk a legkevésbé szigorú feltételeket, melyekkel egy minimális topológiájú hálózat kiegyensúlyozott hálózati forgalmat tud kezelni egy meghatározott teljes QoS mellett.

A leírt algoritmus folyamatábrája a következő ábrán látható:



41. ábra. A többnode-os optimalizációs algoritmus folyamatábrája

5.7. Numerikus eredmények

A következő táblázatok egy adott tipikus forgalmi-terhelési konfiguráció (a forgalmi-terhelési konfiguráció az adott forgalmi osztályban lévő felhasználók számával fejezhető ki) mellett megtalált optimális topológiákat mutatja be:

1. 100% Internet Access1 users: $\mathbf{n} = (3000, 3000, 0, 0, 0, 0)$;

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN

2. 70% Internet Access1 users + 30% Voice over DSL users :

$$\mathbf{n} = (2100, 2100, 0, 0, 900, 900);$$

3. 50% Internet Access1 users + 20% Internet Access2 users + 30% Voice over DSL users: $\mathbf{n} = (1500, 1500, 600, 600, 900, 900)$

4. 70% Internet Access2 users + 30% Voice over DSL users :

$$\mathbf{n} = (0, 0, 2100, 2100, 900, 900)$$

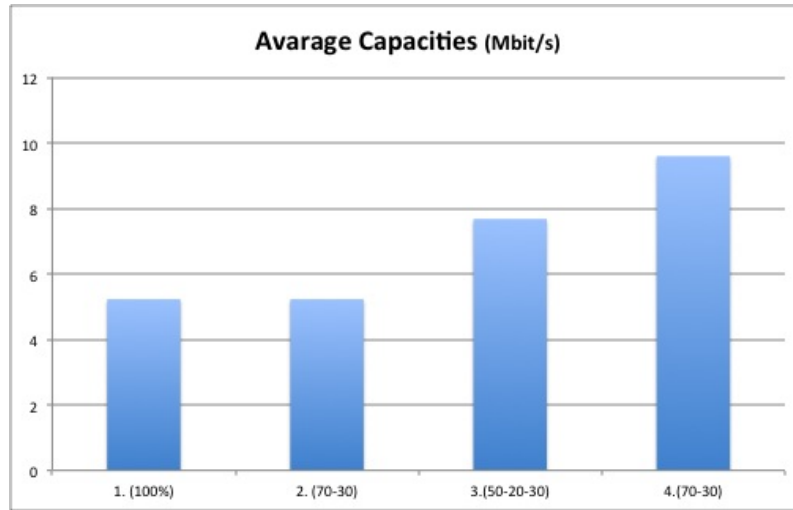
Az 5. táblázat a többnode-os kiegyensúlyozott terheléses optimalizáló algoritmus segítségével elkészített eredményeket mutatja be.

	eszközsám	QoS- paraméter $((\gamma_1, \gamma_2))$	kapacitás (C_1, C_2)
1.(100%)	1. réteg 12 2. réteg 1	1. réteg 7.895 2. réteg 9.215	5.23 Mbit/s 12.21 Mbit/s
2.(70-30)	1. réteg 12 2. réteg 1	2. réteg 7.895 2. réteg 9.215	5.23 Mbit/s 12.21 Mbit/s
3.(50-20-30)	1. réteg 12 2. réteg 1	1. réteg 7.895 2. réteg 9.215	6.98 Mbits/s 19.2 Mbit/s
4.(70-30)	1. réteg 12 2. réteg 1	1. réteg 7.895 2. réteg 9.215	8.72 Mbit/s 22.69 Mbit/s

5. táblázat. Forgalom-terhelési konfiguráció mellett megtalált optimális topológiák kiegyensúlyozott terheléses algoritmussal (1-es módszer)

Látható, hogy a különböző kimeneti vektorok esetében csak a kapacitás és a QoS beállítások különböznek.

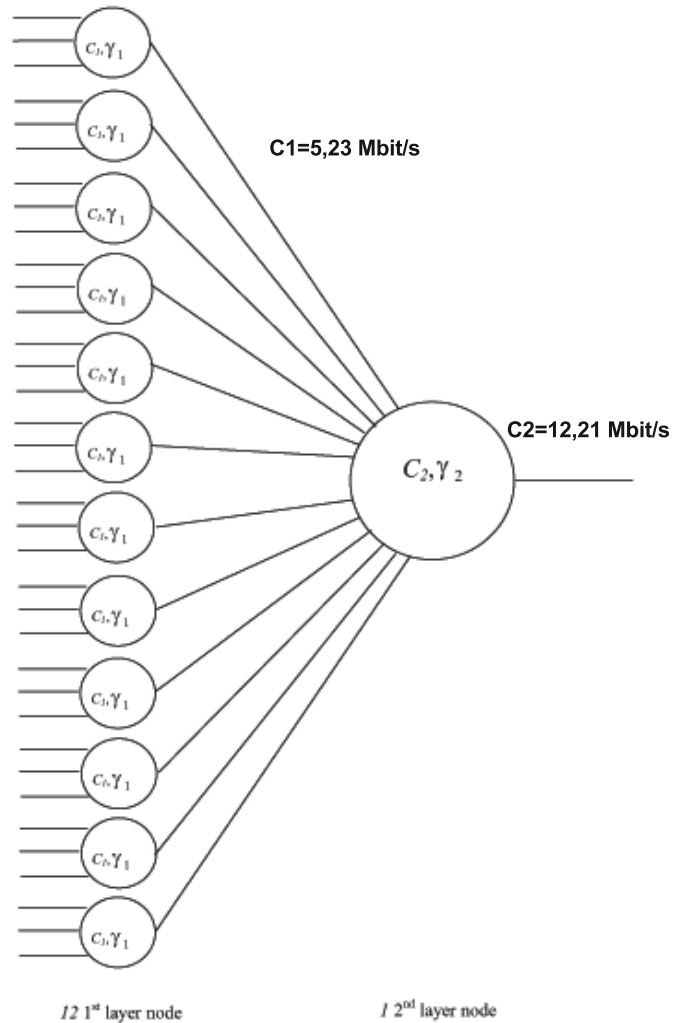
5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN



42. ábra. Átlagkapacitások a különböző forgalomkonfigurációkhoz

A 42. ábrán látható az átlagos kapacitás igény $(\frac{1}{\sum_{k=1}^K L_k} \sum_{k=1}^K \sum_{l=1}^{L_k} C_l(k))$ a négy különböző esetben. Észrevehető, hogy a negyedik esetben a legnagyobb az átlagos kapacitás. Az oszlopdiagramból egyértelműen leolvasható, hogy a 4. konfiguráció jelenti a legszigorúbb kapacitás követelményeket.

5. HÁLÓZATDIMENZIONÁLÁS CSOMAGKAPCSOLT HÁLÓZATOKBAN



43. ábra. Kiegyensúlyozott terheléses algoritmus segítségével készített optimális hálózati topológia

A fenti eredmények megadják az adott forgalmi konstellációhoz tartozó minimális HAM architektúrát (topológia, kapacitás és belső QoS értékek). Az új algoritmus minden más forgalmi konfigurációra is képes elvégezni a dimenzionálást.

6. Konklúziók

Ez a fejezet az elért eredményeket a jelenlegi technológiák szűk keresztmetszetei alapján taglalja. Az alábbi táblázat ezek legyőzésére vonatkozó módszerek dolgozat-beli helyét adja meg.

	Csomag- klasszifikáció	Vezetéknélküli csomag- továbbítás	Hálózat- dimenzionálás
Csomagtovábbítás- minőségi feltételek biztosítása	3.5 fejezet	4.3.1 fejezet	5.5 fejezet 5.6 fejezet
Energiafelhasználás		4.3.2 fejezet 4.3.3 fejezet	
Forgalom- optimalizálás	3.5 fejezet		5.6 fejezet

Összességében elmondható, hogy a csomagkapcsolt hálózatok szűk keresztmetszeteire vonatkozóan a következő, az alábbi alfejezetekben részletezett, akadályokat sikerült áttörni.

6.1. Csomagklasszifikáció CNN-alapú technológiával

A jelenleg használatos IPv6-os routing-technológiát sikerült az általam készült módszerrel javítani, ennek segítségével az Interneten nyújtott - főleg multimédiás - szolgáltatások minősége javítható.

A dolgozat alapján az új algoritmus működőképes, elvégzi a csomagklasszifikáció feladatát. Az algoritmust szimulációval teszteltem, és vizsgáltam a teljesítőképességét.

A CNN-nel történő klasszifikáció technológiai nyitás egy teljesen új rendszer felé. A tesztelés és az algoritmus megírása bebizonyította, hogy egy működőképes és grafikai és párhuzamossági tulajdonságának köszönhetően egy gyors módszert jelent a CNN-alapú megoldás .

A teljesítőképesség-analízis során rangsort állítottam fel az algoritmusok között. A neurális és statisztikai alapú algoritmusok jól teljesítenek, és megfelelő hardveres

támogatás mellett mindegyik egy használható módszer. A konstrukció egy mért routing táblában valós adatokkal 0,224 ms alatt végezte el a klasszifikációt, míg az AQT 0,316 ms és Radix 0,615 ms alatt teljesítette ezt a feladatot. A CNN-es megoldás így majdnem 3x-os sebességnövekedést tud elérni.

Az eredmények segítségével új felépítésű csomagtovábbítási eszközök kialakítása valósulhat meg, amelyek megfelelő csomagtovábbítási paramétereket tudnak teljesíteni.

6.2. Vezetéknélküli csomagtovábbítás útvonalválasztása

Új algoritmust hoztam létre, amely az útvonalkeresést egy energiaoptimalizálással ötvözi, úgy, hogy figyelembe veszi a valóságos csomópontok diszkrét energiaküldési lehetőségeit is. A szimuláció során belátható, hogy az algoritmus alkalmas arra, hogy csomagokat továbbítsunk, valamint képes arra, hogy a csomópontokra a küldési információt továbbítsa. A szimulációs eredményeket tekintve azt láttuk, hogy az algoritmus a hagyományos Leach-protokollal szemben átlagosan 1.8x olyan jól teljesít. Azt is megfigyelhetjük, hogy az esetek 90%-ban teljesít jobban, és azon esetekben, amikor a csomópontok "cluster"-esednek, tud a Leach előnyt szerezni.

Az eredmények segítségével biológiai területen életminőség-javítás érhető el, ipari folyamatok monitorozásában költséghatékonyság és jobb szervezettség érhető el, javíthatja a minőséget; lakás vagy irodaház monitorozása esetében pedig energiahatékonyságot, jobb környezetet tudunk kialakítani.

6.3. Hálózatdimenzionálás csomagkapcsolt hálózatokban

Csomagkapcsolt hálózatok tervezése esetén kiemelkedően fontos, hogy adott szolgáltatás-minőséget minimális hardvereszközök segítségével valósítsunk meg. Ezt a hálózattervezési feladatot egy kényszeres optimalizálási feladatként lehet értelmezni, ahol a cellavesztési valószínűség meghatározott.

A téziscsoportban olyan iterációs algoritmusokat mutattam be, amelyek az egyes felhasználói forgalom-terhelési konfigurációs kezdeti feltételek mellett megoldják a feladatot úgy, hogy a cellavesztési valószínűségi kritériumot a teljesítik.

Sikerült az adott felhasználói forgalmat minimális kapacitású linkekkel megfelelő minőségben kiszolgálni, ahol a link kapacitás 12.21 MBit/sec.

6.4. Összefoglalás

A fenti eredmények átfogják a csomagkapcsolt hálózati kommunikáció legfontosabb korlátait, és ezen korlátok feloldására szolgáló algoritmusokat mutatnak be:

- a router-technológiában fontos real-time csomagklasszifikációs feladatok CNN-alapú gyors megoldása;
- energiahatékony útvonalválasztási protokollok kidolgozása energiában limitált vezeték nélküli szenzoriális hálózatok számára;
- minimális komplexitású hardver tervezése adott forgalom kiszolgálására internet-hozzáférési modulokban.

A felsorolás alapján a disszertáció új eredményekkel járult hozzá a csomagkapcsolt hálózatok teljesítőképességének növeléséhez (ezeknek számszerű értékei a [6.1](#), [6.2](#) és a [6.3](#) fejezetben találhatóak).

A szerző publikációi

- [1] J. Levendovszky and B. Karlócai, „Dimensioning hierarchical admission architect,” *Periodica Polytechnica*, 2012 - in progress.
- [2] B. Karlócai, A. Bojársky, and J. Levendovszky, „Energy aware routing protocols for wireless sensor networks using discrete transmission energies,” in *2011 International Joint Conference of IEEE TrustCom-11/IEEE ICES-11/FCST-11*. IEEE, 2011, pp. 1704–1707.
- [3] J. Levendovszky, B. Karlócai, and A. Bojársky, „Novel cnn based packet classification algorithms for networking,” *Periodica Polytechnica*, 2011 - in progress.
- [4] J. Levendovszky, A. Bojársky, B. Karlócai, and A. Oláh, „Energy balancing by combinatorial optimization for wireless sensor networks,” *WSEAS Transactions on Communications*, vol. 7, no. 2, pp. 27–32, 2008.
- [5] A. Bojársky and B. Karlocai, „Új neurális alapú csomagklasszifikációs algoritmusok fejlesztése és tesztelése ipv6 hálózatok számára,” in *Országos Tudományos Diákköri Konferencia, Informatika szekció, Konferenciakiadvány*, vol. 1, 2005, p. 76.

Hivatkozások

- [1] H. Lim, M. Kang, and C. Yim, „Two-dimensional packet classification algorithm using a quad-tree,” *Computer communications*, vol. 30, no. 6, pp. 1396–1405, 2007.
- [2] P. Gupta and N. McKeown, „Algorithms for packet classification,” *Network, IEEE*, vol. 15, no. 2, pp. 24–32, 2001.
- [3] J. Zhao, X. Zhang, X. Wang, and X. Xue, „Achieving o(1) ip lookup on gpu-based software routers,” in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 429–430.
- [4] S. Summers, „Wireless sensor networks for firefighting and fire investigation,” *UCCS, CS526 Project, Spring*, 2006.
- [5] S. Ergen, „Zigbee/ieee 802.15. 4 summary,” 2004.
- [6] S. Vegesna, *IP quality of service*. Cisco Systems, 2001.
- [7] J. Yick, B. Mukherjee, and D. Ghosal, „Wireless sensor network survey,” *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [8] E. Jovanov, A. Milenkovic, C. Otto, P. De Groen, B. Johnson, S. Warren, and G. Taibi, „A wban system for ambulatory monitoring of physical activity and health status: applications and challenges,” in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the IEEE*, 2006, pp. 3810–3813.
- [9] B. Latre, B. Braem, I. Moerman, C. Blondia, E. Reusens, W. Joseph, and P. Demeester, „A low-delay protocol for multihop wireless body area networks,” in *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on*. IEEE, 2007, pp. 1–8.
- [10] D. Taylor, „Survey and taxonomy of packet classification techniques,” *ACM Computing Surveys (CSUR)*, vol. 37, no. 3, pp. 238–275, 2005.

- [11] J. y. N. Hui, *Switching and traffic theory for integrated broadband networks*. Boston: Kluwer Academic Publishers, 1990.
- [12] M. De Berg, O. Cheong, and M. Van Kreveld, *Computational geometry: algorithms and applications*. Springer-Verlag New York Inc, 2008.
- [13] M. Hasna and M. Alouini, „End-to-end performance of transmission systems with relays over rayleigh-fading channels,” *Wireless Communications, IEEE Transactions on*, vol. 2, no. 6, pp. 1126–1131, 2003.
- [14] S. Kumar, „Sensor networks: evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [15] J. Levendovszky and B. Hegyi, „Optimal statistical energy balancing protocols for wireless sensor networks,” *WSEAS Transactions on Communications*, pp. 689–694, 2007.
- [16] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, „Energy-efficient communication protocol for wireless microsensor networks,” in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. IEEE, 2002, pp. 10–pp.
- [17] D. Puccinelli and M. Haenggi, „Wireless sensor networks: applications and challenges of ubiquitous sensing,” *Circuits and Systems Magazine, IEEE*, vol. 5, no. 3, pp. 19–31, 2005.
- [18] A. Bache, L. BUILLOU, H. Layec, B. LORIG, and Y. Matras, „Rcp, the experimental packet-switched data transmission service of the french ptt: History, connections, control,” in *Proceedings of the Third International Conference on Computer Communication (ICCC)*, 1976, pp. 3–6.
- [19] R. Tucker, J. Baliga, R. Ayre, K. Hinton, and W. Sorin, „Energy consumption in ip networks,” in *Optical Communication, 2008. ECOC 2008. 34th European Conference on*. Ieee, 2008, pp. 1–1.
- [20] A. Pras, L. Nieuwenhuis, R. van de Meent, and M. Mandjes, „Dimensioning network links: a new look at equivalent bandwidth,” *Network, IEEE*, vol. 23, no. 2, pp. 5–10, 2009.

- [21] A. Lööf and H. Seybert, „Internet usage in 2009-households and individuals,” *Eurostat Data in focus*, vol. 46, p. 2009, 2009.
- [22] R. Sinha, C. Papadopoulos, and J. Heidemann, „Internet packet size distributions: Some observations,” *USC/Information Sciences Inst.[Online]*. Available: <http://netweb.usc.edu/~rsinha/pkt-sizes>, 2007.
- [23] Z. Meng, „The overview of the strategy of transition from ipv4 to ipv6,” *Science*, p. 12, 2008.
- [24] M. Sabir, M. Fahiem, and M. Mian, „An overview of ipv4 to ipv6 transition and security issues,” in *Communications and Mobile Computing, 2009. CMC'09. WRI International Conference on*, vol. 3. IEEE, 2009, pp. 636–639.
- [25] C. Liljenstolpe, W. George, C. Donley, and L. Howard, „Ipv6 support required for all ip-capable nodes,” *Internet Draft*, 2011.
- [26] X. Zhao, D. Pacella, and J. Schiller, „Routing scalability: an operator’s view,” *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 8, pp. 1262–1270, 2010.
- [27] J. Postel, „Rfc 791: Internet protocol,” 1981.
- [28] Y. Rekhter and T. Li, „An architecture for ip address allocation with cidr,” *RFC*, 1993.
- [29] W. Doeringer, G. Karjoth, and M. Nassehi, „Routing on longest-matching prefixes,” *IEEE/ACM Transactions on Networking (TON)*, vol. 4, no. 1, pp. 86–97, 1996.
- [30] V. Bollapragada, C. Murphy, and R. White, *Inside cisco ios software architecture*. Cisco Systems, 2000.
- [31] D. Capite, *Self-defending networks: The next generation of network security*. Cisco Press, 2006.
- [32] R. Barnhill, „Representation and approximation of surfaces,” *Mathematical Software*, vol. 3, pp. 69–120, 1977.

- [33] H. Baumgarten, H. Jung, and K. Mehlhorn, „Dynamic point location in general subdivisions,” in *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1992, pp. 250–258.
- [34] M. De Berg, M. Van Kreveld, and J. Snoeyink, „Two-and three-dimensional point location in rectangular subdivisions,” *Algorithm Theory SWAT92*, pp. 352–363, 1992.
- [35] J. Bentley, „Solutions to klee’s rectangle problems,” *Unpublished manuscript, Dept of Comp Sci, Carnegie-Mellon University, Pittsburgh PA*, 1977.
- [36] —, „Decomposable searching problems, inform,” *Proc. Lett*, vol. 8, pp. 244–251, 1979.
- [37] R. Karlsson and U. of Waterloo. Faculty of Mathematics, *Algorithms in a restricted universe*. Faculty of Mathematics, University of Waterloo, 1984.
- [38] S. Sahnı and K. Kim, „Efficient construction of multibit tries for ip lookup,” *IEE/ACM Transactions on Networking (TON)*, vol. 11, no. 4, pp. 650–662, 2003.
- [39] K. Sklower, „A tree-based packet routing table for berkeley unix,” in *Proceedings of the Winter 1991 USENIX Conference*, 1991, pp. 93–104.
- [40] S. Nilsson and G. Karlsson, „Ip-address lookup using lc-tries,” *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 6, pp. 1083–1092, 1999.
- [41] M. Degermark, A. Brodnik, S. Carlsson, and S. Pink, *Small forwarding tables for fast routing lookups*. ACM, 1997, vol. 27, no. 4.
- [42] W. Eatherton, G. Varghese, and Z. Dittia, „Tree bitmap: hardware/software ip lookups with incremental updates,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 97–122, 2004.
- [43] M. Buddhikot, S. Suri, and M. Waldvogel, „Space decomposition techniques for fast layer-4 switching,” in *Protocols for High Speed Networks*, vol. 6. Citeseer, 1999, pp. 25–42.

- [44] F. Baboescu, P. Warkhede, S. Suri, and G. Varghese, „Fast packet classification for two-dimensional conflict-free filters,” *Computer Networks*, vol. 50, no. 11, pp. 1831–1842, 2006.
- [45] H. Lu and S. Sahni, „Conflict detection and resolution in two-dimensional prefix router tables,” *Networking, IEEE/ACM Transactions on*, vol. 13, no. 6, pp. 1353–1363, 2005.
- [46] P. Gupta and N. McKeown, „Packet classification on multiple fields,” in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 1999, pp. 147–160.
- [47] L. Chua and T. Roska, „The cnn paradigm,” *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 40, no. 3, pp. 147–156, 1993.
- [48] T. Roska and L. Chua, „The cnn universal machine: An analogic array computer,” *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 40, no. 3, pp. 163–173, 1993.
- [49] T. Roska, „Cellular wave computer architectures in a new era of computing 15 years later,” *International Journal of Circuit Theory and Applications*, vol. 36, no. 5-6, pp. 523–524, 2008.
- [50] L. Chua and L. Yang, „Cellular neural networks: Theory,” *Circuits and Systems, IEEE Transactions on*, vol. 35, no. 10, pp. 1257–1272, 1988.
- [51] —, „Cellular neural networks: Applications,” *Circuits and Systems, IEEE Transactions on*, vol. 35, no. 10, pp. 1273–1290, 1988.
- [52] T. Roska, G. Paziienza, and C. Wu, „Cellular wave computing via nanoscale chip architectures,” *International Journal of Circuit Theory and Applications*, 2010.
- [53] T. Roska, L. Belady, and M. Ercsey-Ravasz, „Cellular wave computing in nanoscale via million processor chips,” *Cellular Nanoscale Sensory Wave Computing*, p. 5, 2009.

- [54] T. Roska, C. Wu, and L. Chua, „Stability of cellular neural networks with dominant nonlinear and delay-type templates,” *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 40, no. 4, pp. 270–272, 1993.
- [55] A. Abnous, K. Seno, Y. Ichikawa, M. Wan, and J. Rabaey, „Evaluation of a low-power reconfigurable dsp architecture,” *Parallel and Distributed Processing*, pp. 55–60, 1998.
- [56] J. Glossner, K. Chirca, M. Schulte, H. Wang, N. Nasimzada, D. Har, S. Wang, A. Hoane Jr, G. Nacer, M. Moudgill *et al.*, „Sandblaster low power dsp [parallel dsp arithmetic microarchitecture],” in *Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004*. IEEE, 2004, pp. 575–581.
- [57] T. Ho, P. Lam, and C. Leung, „Parallelization of cellular neural networks on gpu,” *Pattern Recognition*, vol. 41, no. 8, pp. 2684–2692, 2008.
- [58] B. Soos, A. Rak, J. Veres, and G. Cserey, „Gpu powered cnn simulator (simcnn) with graphical flow based programmability,” in *Cellular Neural Networks and Their Applications, 2008. CNNA 2008. 11th International Workshop on*. IEEE, 2008, pp. 163–168.
- [59] A. Kiss, „An optimal mapping of numerical simulations of partial differential equations to emulated digital cnn-um architectures,” 2011.
- [60] P. Pereira, A. Grilo, F. Rocha, M. Nunes, A. Casaca, C. Chaudet, P. Almström, and M. Johansson, „End-to-end reliability in wireless sensor networks: survey and research challenges,” in *EuroFGI Workshop on IP QoS and Traffic Control*. Citeseer, 2007, pp. 67–74.
- [61] C. Wan, A. Campbell, and L. Krishnamurthy, „Psfq: a reliable transport protocol for wireless sensor networks,” in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 1–11.
- [62] F. Zhao, J. Shin, and J. Reich, „Information-driven dynamic sensor collaboration for tracking applications,” *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 61–72, 2002.

- [63] J. Levendovszky, G. Kiss, and L. Tran-Thanh, „Energy balancing by combinatorial optimization for wireless sensor networks,” *Performance Modelling and Analysis of Heterogeneous Networks*. River Publishers, Aalborg, Denmark, pp. 169–182, 2009.
- [64] J. Levendovszky, A. Bojársky, B. Karlócai, and A. Oláh, „Energy balancing by combinatorial optimization for wireless sensor networks,” *WSEAS Transactions on Communications*, vol. 7, no. 2, pp. 27–32, 2008.
- [65] J. Levendovszky, A. Olah, G. Treplan, and L. Tran-Thanh, „Reliability-based routing algorithms for energy-aware communication in wireless sensor networks,” *Performance Models and Risk Management in Communications Systems*, pp. 93–126, 2011.
- [66] G. Ran, H. Zhang, and S. Gong, „Improving on leach protocol of wireless sensor networks using fuzzy logic,” *Journal of Information & Computational Science*, vol. 7, no. 3, pp. 767–775, 2010.
- [67] D. Gough, L. Kumosa, T. Routh, J. Lin, and J. Lucisano, „Function of an implanted tissue glucose sensor for more than 1 year in animals,” *Science Translational Medicine*, vol. 2, no. 42, pp. 42ra53–42ra53, 2010.
- [68] A. Goldsmith and S. Wicker, „Design challenges for energy-constrained ad hoc wireless networks,” *Wireless Communications, IEEE*, vol. 9, no. 4, pp. 8–27, 2002.
- [69] J. Chang and L. Tassiulas, „Energy conserving routing in wireless ad-hoc networks,” in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 2000, pp. 22–31.
- [70] S. Singhand and C. Raghavendra, „Pamas: Power aware multi-access protocol with signalling for ad hoc networks,” *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, 1998.
- [71] F. Xiangning and S. Yulin, „Improvement on leach protocol of wireless sensor network,” in *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*. IEEE, 2007, pp. 260–264.

- [72] S. Muruganathan, D. Ma, R. Bhasin, and A. Fapojuwo, „A centralized energy-efficient routing protocol for wireless sensor networks,” *Communications Magazine, IEEE*, vol. 43, no. 3, pp. S8–13, 2005.
- [73] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, „Wireless sensor networks: a survey,” *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [74] Y. Li, Z. Wang, and Y. Song, „Wireless sensor network design for wildfire monitoring,” in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 1. IEEE, 2006, pp. 109–113.
- [75] IEEE, „Ieee 802.15.4 wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans),” 09 2006.
- [76] H. Sumi, „Sentience abstractions in pervasive spaces,” Mobile and Pervasive Computing Laboratory, University of Florida, Tech. Rep., 2011.
- [77] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, „Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [78] Z. Wang and J. Crowcroft, „Quality-of-service routing for supporting multimedia applications,” *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 7, pp. 1228–1234, 1996.
- [79] H. Imaizumi and H. Morikawa, „Directions towards future green internet,” *Towards Green Ict*, vol. 9, p. 37, 2010.
- [80] G. Davies, M. Hardt, and F. Kelly, „Come the revolution—network dimensioning, service costing and pricing in a packet switched environment,” *Telecommunications Policy*, vol. 28, no. 5-6, pp. 391–412, 2004.
- [81] J. Soldatos, E. Vayias, P. Stathopoulos, and N. Mitrou, „Enforcing effective rates for packet-level qos control in ip networks: theory and validation based on real traffic data,” *Telecommunication Systems*, vol. 27, no. 1, pp. 9–31, 2004.

- [82] S. Sharafeddine and Z. Dawy, „Robust network dimensioning for realtime services over ip networks with traffic deviation,” *Computer Communications*, vol. 33, no. 8, pp. 976–983, 2010.
- [83] F. Kelly, „Notes on effective bandwidths,” *Stochastic networks: theory and applications*, pp. 141–168, 1996.
- [84] J. Levendovszky and S. Imre, „Comparative analysis of cac algorithms for atm networks,” *Scientific Progress Report, COP579-SPR-Î, EU*, pp. 303–314, 1994.
- [85] J. Levendovszky, S. Imre, E. van der Meulen, and L. Pap, „Tail distribution estimation for call admission in atm networks,” in *Participants Proceedings of 3th IFIP Workshop and Performance Evaluation and Modelling of ATM Networks, Ilkley, West Yorkshire, UK*, 1995.