Tűzfal és eljárás hálózati adatcsomagok belső hálózatok és külső hálózatok közötti forgalmának vezérlésére

K I V O N A T

A találmány tárgya tűzfal hálózati adatcsomagok belső hálózatok és külső hálózatok közötti forgalmának vezérlésére,

amely tűzfal a hálózatok között továbbítandó adatcsomag adatmezőinek tartalmától függően az adatcsomagra alkalmazható szabályt - szabályok teljes készletéből - kiválasztó, az adatcsomagot megállító vagy a tűzfalon keresztül továbbító szűrővel rendelkezik. Jellegzetessége, hogy

a forráscímmel és a rendeltetési címmel társított előtag ábrázolás utáni megkeresésére az adatcsomag forráscímét és rendeltetési címét címek előtagjainak halmazában két dimenzióban kereső kétdimenziós címkeresőt (8) tartalmaz,

a szabályok teljes készletéből mindegyik előtaghoz részkészlet tartozik,

az adatcsomagra alkalmazható szabály megkeresésére pedig az adatmező tartalma alapján szabály egyeztetésére alkalmas szabályegyeztetővel (10) van ellátva.

A találmány tárgyához tartozik továbbá az eljárás hálózati adatcsomagok belső hálózatok és külső hálózatok közötti, tűzfalon keresztülmenő forgalmának vezérlésére, melynek során

a hálózatok között továbbítandó adatcsomag adatmezőinek tartalmától függően az adatcsomagra alkalmazható szabályt szabályok teljes készletéből kiválasztjuk,

a szabályt az adatcsomaggal egyeztetjük,

az adatcsomagot pedig a szabálytól függően szűrő segítségével megállítjuk, vagy a tűzfalon keresztül továbbítjuk. Jellegzetessége, hogy a szűrés során

a forráscímmel és a rendeltetési címmel társított előtag ábrázolás utáni megkeresésére az adatcsomag forráscímét és rendeltetési címét címek előtagjainak halmazában kereső kétdimenziós keresést hajtunk végre, a szabályok teljes készletéből mindegyik előtaghoz részhalmazt rendelünk,

az adatcsomagra alkalmazható szabály megkeresésére pedig az adatcsomag adatmezőinek tartalma alapján a szabályok részkészletén szabályegyeztetést hajtunk végre.

12. ábra

P 01 03 814

A2

(2001.III.)

Képviselő: ADVOPATENT Szabadalmi és Védjegy Iroda

Tűzfal és eljárás hálózati adatcsomagok belső hálózatok és külső hálózatok közötti forgalmának vezérlésére

A találmány tárgya tűzfal hálózati adatcsomagok belső hálózatok és külső hálózatok közötti forgalmának vezérlésére, amely tűzfal a hálózatok között továbbítandó adatcsomag adatmezőinek tartalmától függően az adatcsomagra alkalmazható szabályt - szabályok teljes készletéből - kiválasztó, az adatcsomagot megállító vagy a tűzfalon keresztül továbbító szűrővel rendelkezik.

A találmány tárgya továbbá eljárás hálózati adatcsomagok belső hálózatok és külső hálózatok közötti, tűzfalon keresztülmenő forgalmának vezérlésére, melynek során a hálózatok között továbbítandó adatcsomag adatmezőinek tartalmától függően az adatcsomagra alkalmazható szabályt - szabályok teljes készletéből - kiválasztjuk, a szabályt az adatcsomaggal egyeztetjük, az adatcsomagot pedig a szabálytól függően szűrő segítségével megállítjuk, vagy a tűzfalon keresztül továbbítjuk.

15014

A legtöbb Internettel összekapcsolt szervezet számára igen fontos a biztonság, következésképpen a legtöbb szervezet számítógép és hálózati stratégiájában egyre fontosabb szerepet játszanak a tűzfalak. A web kiszolgálóhoz vagy a szervezet más nyilvános szolgáltatásaihoz hozzáférő felhasználóknak nem szabad elérniük a belső szolgáltatásokat, pl. a számfejtést, az Internet információs kiszolgálót és más kényes céginformációkat.

A rendszerkiszolgálókat nem szabad megszakítani, tehát a kiszolgálókat és a munkaállomásokat védeni kell az Internet felhasználók DOS (denial-of-service) támadásai ellen.

A tűzfal vagy szűrő útvonalválasztó olyan eszköz, amely lényegében az útvonalválasztóhoz hasonlóan működik. Tehát a bejövő csatoló egységen keresztül csomagokat kap, a csomagok rendeltetési címét megvizsgálja, majd a csomagot a rendeltetési cím figyelembevételével a megfelelő kimenő csatoló egységre irányítja.

A tűzfal azonban minden egyes csomagot sokkal alaposabban megvizsgál. A forrás és rendeltetési címeket, forrás és rendeltetési portokat, protokoll mezőt, állapotjelzőket és opciókat mind meg kell vizsgálni és a tűzfal szabályok jegyzékével össze kell hasonlítani. Attól függően, hogy melyik szabály illik a csomagra, a tűzfal nem továbbítja a csomagot, ha például megállító szabály illik rá.

A jogosulatlan elérésen kívül az Internethez kapcsolódó szervezet további veszélyeknek is ki van téve. A lényeg az, hogy az ismeretlen eredetű adatokban nem lehet megbízni. A technika állása szerinti tűzfalak feladata közé tartozik a vírusok és trójai falovak felkutatása villámpostákban és hálóoldalakon.

Később, ahogy a hálózat sávszélessége egyre nő, úgy válik a tűzfal szerepe mind fontosabbá.

A tűzfalak sok különböző szinten dolgozhatnak és a rajtuk áthaladó adatok átvizsgálását különféle módokon végezhetik el. Az összes tűzfal alapvető működéséhez tartozik azonban a szűrés, amely a hálózati (IP = Internet Protocol) és transzport (TCP = Transport Control Protocol) és ICMP (Internet Control Message Protocol) rétegek fejlécén alapul.

Ilyen IP szűrés hiányában minden más tevékenység, pl.adatvizsgálat haszontalan, hiszen a belső hálózatok felhasználói így hálózati alkalmazásaikat kialakíthatják úgy is, hogy azok a távoli kiszolgálóhoz ne a vizsgálón áthaladva kapcsolódjanak, így pedig megkerülik az összes biztonsági intézkedést.

Cégek és szervezetek különböző okokból kapcsolódnak az Internethez, ilyen lehet pl. az, hogy a cégről, termékeikről és szolgáltatásaikról tájékoztatást tegyenek közzé a hálón, elérjék az Interneten rendelkezésre álló adatokat, valamint levelezésüket villámpostán bonyolítsák le

A cég azonban rendszerint olyan belső adatokkal is rendelkezik, amelyeket az Internet felhasználói nem érhetnek el, ilyenek pl. az Internet közlemények kiszolgálói, a fájlkiszolgálók, stb. A legáltalánosabb kialakítás szerint megengedik a kapcsolatot az Internetről a kiszolgálók bizonyos csoportjával (háló, villámposta és más nyilvános szolgáltatások), más gazdák (pl. intranet kiszolgálók) elérését azonban tiltják.

Ennek elérésére "demilitarizált zónát" (DMZ) létesítenek. A DMZ-ben a számítógépekkel mind az Internet, mind pedig az intranet kapcsolatot teremthet, ámde az intranet hozzáférés az Internetről korlátozva van. A technika állása szerinti belső hálózat, pl. intranet a demilitarizált zónához tűzfal, míg a DMZ az Internethez útvonalválasztó útján kapcsolódik.

Következésképpen a hálózati forgalom szabadon bonyolódhat az Internet és a DMZ között, amely az intranet felhasználói elől egyáltalán nincs védve. Ennek az az oka, hogy a technika állása szerinti tűzfalaknál nincs lehetőség arra, hogy kettőnél több hálózatot, vagyis egy belső és egy külső hálózatot kapcsoljanak össze.

Ismeretesek három hálózati csatoló egységgel rendelkező tűzfalak is. Az Internet és a DMZ vagy az intranet közötti forgalmat ezeknél ugyanúgy korlátozni lehet. Bizonyos mértékig korlátozzák a forgalmat a DMZ-ben a gazdákhoz és gazdáktól, pl. a web kiszolgáló csak a HTTP (Hypertext Transfer Protocol) porton érhető el. Az Internet felhasználóknak a többi szolgáltatáshoz nem szabad hozzáférni.

Az intranet felhasználók azonban többféleképpen hozzá akarnak férni a web kiszolgálóhoz, mint az Internet felhasználók, pl. adminisztratív célból, ezért eközött a két hálózat között több elérési lehetőséget kell nyújtani. Hasonló szabályokra van szüksége a villámposta kiszolgálójának, SMTP (Simple Mail Transfer Protocol) kapcsolatokat kell engedélyezni az Internet felől, de a villámpostát csak az intranet bizonyos kijelölt gazdái olvashatják, és esetleg néhány gazda az Interneten.

Tűzfal környezetben a gépek száma a DMZ-ben pl. 30. A szabály a gépekhez a DMZ-ben minden gépre más-más lehet, de a szabályok száma gépenként viszonylag alacsony, pl. 10-15. Több szabály vonatkozhat az intranetről a DMZ-re irányuló forgalomra, de ezek valószínűleg általánosabbak. Tehát a DMZ-ben az összes gépre viszonylag kevés szabály vonatkozik.

A legtöbb esetben az Internet és az intranet(ek) közötti forgalomra is kevés szabály vonatkozik, ha egyáltalán vannak ilyenek. A forgalom legnagyobb részét le kell állítani. Mindazonáltal az intranetről kezdeményezett forgalom megengedhető.

Ahogy az Internet felhasználók száma növekszik, a nyilvános kiszolgálókat mind gyakrabban látogatják, és ezzel növelik a forgalmat. A forgalom az intranetre és az intranetről szintén nő, mivel az intranet felhasználók is részesülni akarnak az Interneten rendelkezésre álló növekvő mennyiségű információból. Ezáltal a sávszélesség iránti követelmény is egyre nagyobb lesz, ami pedig nagyobb követelményeket támaszt a tűzfal működésével szemben.

A tűzfal fő feladatává így a csomagszűrés vált, vagyis adva van az IP csomag és a szabályok készlete, kérdés, hogy melyik szabályt alkalmazzák az adott csomagra. Ha ugyanarra a csomagra több szabály is vonatkozik, a kiválasztandó szabály meghatározásához irányelvekre van szükség. A kérdés kezelésére a technika állásában két módszer ismeretes.

Az egyik módszer szerint kiválasztják azt a szabályt, amely a csomag legtöbb mezőjére ráillik, és ha ugyanannyi mezőre két különböző szabály illik rá, akkor meg kell határozni a sorrendjüket. Ezt használja Borg és Flodin a csomagosztályozó algoritmusban (N.Borg, Flodin, Malin: Packet

Classification, 1997. június, N.Borg: A Packet Classifier for IP Networks, Masters. Lic., Luleå University of Technology, 1998. február).

Egy másik módszer szerint meghatározzák a szabályok közötti sorrendet, és a kiválasztandó szabály meghatározásához ezt a sorrendet használják. A második módszer előnye, hogy a szűrő szabályok meghatározásához nagyobb rugalmasságot nyújt, és a NetBSD tűzfal kód ezt a módszert használja.

A szűrő szabály egy sor kielégítendő feltételt, majd a feltételek kielégítése után végrehajtandó műveletet foglal magában. A feltételek az alábbiakon alapulnak: IP forrás és rendeltetési címek (32 bites előtagok), IP protokoll mező (8 bites egész szám), redelkezik-e a csomag IP lehetőséggel vagy nem, és az IP/TCP forrás és rendeltetési portok számának (két 16 bites egész szám) megfelelően melyek ezek a lehetőségek (egész szám), TCP fejléc jelzők (3 bit), ICMP fejléc típus és kódmező (két 8 bites egész szám), melyik csatoló egységről olvasták be a csomagot (8+8 bit), és melyik csatoló egységre kell a csomagot továbbítani (8+8 bit).

A jelenlegi tűzfalak nem célozzák meg külön a szabályegyeztetés kérdését, hanem általában szabályok kapcsolt jegyzékét vagy tömbjét tartalmazzák, amelyekkel a csomagot addig hasonlítják össze, amíg nem találnak egyezést. A módszer nem eléggé hatékony. Egy másik módszer szerint a szabályokat feldarabolják. Ha a szabályok között kétértelműség adódik, azaz ugyanarra a csomagra két szabály is ráillik, akkor a legtöbbször úgy oldják meg a kérdést, hogy követendőként az első vagy az utolsó ráillő szabályt veszik figyelembe.

A Cisco Systems technika állása szerinti PIX tűzfala kapcsolatorientált biztonsági berendezés, amely a belső hálózatot külső hálózat ellen védi. A PIX tűzfal rendkívül költséges berendezés, egyidejű kapcsolatainak felső korlátja pedig 16000. A PIX tűzfal legfontosabb része az alkalmazott biztonsági algoritmusra (ASA = Adaptive Security Algorithm) épülő védelmi terv, amely megbízható, kapcsolatorientált biztonságot kínál.

Az ASA nyilvántartja a forrás és rendeltetési címet, a TCP sorozatszámokat, portszámokat, valamint valamennyi csomag kiegészítő TCP jelzőit. Ezeket az adatokat táblázatban tárolják, és a táblázati bejegyzésekkel valamennyi kimenő és bejövő csomagot összehasonlítják.

A létrejött kapcsolatok mindegyikének adatait meg kell őrizni, amíg csak tart a kapcsolat, a lehetséges kapcsolatok számát ezért a rendelkezésre álló memória befogadóképessége szabja meg. A teljesen leterhelt Cisco PIX tűzfal körülbelül 90 Mbájt/s sebességgel képes működni. A Cisco PIX tűzfal azonban a portcím fordítást (PAT = Port Address Translation) is támogatja, ami által egyetlen külső IP cím több, mint 64000 belső gazdát tud kiszolgálni.

A technika állásában ismert ipf (IP filter) elnevezésű csomagszűrőt a NetBSD 1.3. disztribúciója magában foglalja.

A szabálykészletek az ipf-ben fel vannak darabolva azokon a csatoló egységeken, amelyekre érvényesek. A szabályokat kétszer ellenőrzik, először amikor a csomag a gazdához beérkezik, másodszor pedig amikor a gazdát elhagyja. A kimenő portnál ellenőrzendő szabályok között a

csak a bemenő csomagokra érvényes szabályok nem szerepelnek, és fordítva. Az adatszerkezet lényegében optimalizált csatolt lista.

Az Exokernel, D. Engler, M.F. Kaashoek, J.O'Tool Jr., Exokernel: An operating system architecture..., Proceedings of the 15th ACM symposium on Operating Systems principles (1995. december) a csomagok demultiplexálására (DPF) (D.Angler, M.F.Kaashoek, DPF: Fast, flexible message demultiplexing..., D.Engler, M.F.Kaashoek, Computer Communication Revue, 26. kötet, 4. szám, 1996. október) másfajta megközelítést javasol. A szabályok különleges programnyelven vannak írva, majd ezután fordítva. A fordító valamennyi felsorolt szabályt ismeri, a létrehozott programot pedig a várható forgalmi minták szerint lehet optimalizálni.

A találmány célja, hogy a belső és külső hálózatok közötti hálózati forgalom vezérlésére olyan javított tűzfalat és eljárást fejlesszen ki, amely hatékony címkereső és szabályegyeztető módszert nyújt, és amelynek segítségével hatékony és gyors IP csomagszűrést lehet végrehajtani, a kapcsolatok száma a tűzfalon keresztül pedig nincs korlátozva.

A kitűzött célt a találmány szerinti tűzfal és eljárás valósítja meg, amely szerint a lineárisan keresendő szabálykészletet a szabálykészlet felosztása útján csökkentettük. A találmány szerinti tűzfal kétdimenziós címkeresőt tartalmaz, amely kétlépéses keresést hajt végre, elsőként a csomag forráscímének és rendeltetési címének keresését a címek előtagjainak halmazában. Valamennyi előtaghoz a szabályok teljes készletének szabály-részkészlete tartozik. Az eredményül kapott szabály-részkészletben lineáris keresést hajtunk végre, hogy a szóbanforgó adatcsomagra vonatkozó szabályt megtaláljuk.

A találmány további célja, hogy olyan tördelőt fejlesszen ki, amely alkalmas a felaprózott csomag összes töredékének szűrésére.

Az is célja a találmánynak, hogy olyan hálózati címfordítót fejlesszen ki, amely a tűzfaltól továbbított csomag belső forráscímét külső forráscímre, míg a tűzfalba továbbított csomag külső forráscímét belső forráscímre fordítja.

Az is célja a találmánynak, hogy olyan hálózati címfordítót fejlesszen ki, amely belső hálózatról külső hálózatra továbbított csomag belső forráscímét külső forráscímre, míg külső hálózatról belső hálózatra továbbított csomag külső forráscímét belső forráscímre fordítja.

Az is célja a találmánynak, hogy belső hálózat által kezdeményezett kapcsolathoz az adatcsomag ideiglenes kivételét a kívülről-befelé megállító szabály alól az előtagba foglalt adatok alapján elrendelő lyukasztó berendezéssel lássa el, a külső hálózatról a belső hálózatra továbbított adatcsomagok számára pedig a kapcsolat tartamára a tűzfalon visszatérő csatornát nyújtson.

A találmány olyan tűzfal kifejlesztését tűzi ki célul, amely legalább 1000 különféle szabályt képes kezelni.

A találmány szerinti tűzfal és eljárás előnye, hogy a lehetséges egyidejű kapcsolatok száma nincs korlátozva, az IP szűrés gyors, a lehetséges szabályok száma pedig igen nagy.

A találmány további célja, hogy olyan tűzfalat nyújtson, amely útvonalválasztóval rendelkezik.

A találmányt kiviteli pédák kapcsán, a mellékelt ábrák alapján ismertetjük részletesebben. Az

1. ábra a találmány szerinti tűzfalat tartalmazó közös hálózat topológiája, a

2. ábra a találmány szerinti tűzfal blokkdiagramja, a

3. ábra a kétdimenziós sűrű szakasz felosztásának képe, a

4. ábra a találmány szerinti adatszerkezet képe, az

5. ábra a (0,0) osztályú mező képe, a

6. ábra a (1,1) osztályú mező képe, a

7. ábra a (1,2) osztályú mező képe, a

8. ábra a (2,1) osztályú mező képe, a

9. ábra a (1,3+) osztályú mező képe, a

10. ábra a (3+,1) osztályú mező képe, a

11. ábra a (2+,2+) osztályú mező képe, a

12. ábra egy bizonyos lekérdezett kulcs sikertelen keresése a hat kulcsot tartalmazó Patricia fán, a

13. ábra a Patricia fa a lekérdezett kulcs a 12. ábra szerinti sikertelen keresést követő beillesztése után.

Az 1. ábrán egy cég vagy szervezet nézőpontjából korszerű hálózat topológiájára mutatunk be példát. Az 1 belső hálózat, mint pl. intranet, néhány hálózati 2 csomópontot, pl. PC-t, munkaállomást, fájlkiszolgálót, stb. tartalmaz, amelyek a 3 tűzfalhoz hozzá vannak kapcsolva. A 4 külső hálózathoz, pl. Internethez hozzákapcsolt cégek vagy szervezetek a társasággal kapcsolatos információkat, pl. termékeket vagy szolgáltatásokat kívánnak közzétenni a hálón, továbbá

szeretnének hozzáférni más cégek vagy szervezetek Interneten közzétett információihoz, valamint villámposta útján levelezni.

Lehetnek azonban a cégnek olyan adatai, amelyekhez az Internet felhasználóinak nem szabad hozzáférniük, pl. az intranet információ kiszolgálói, fájlkiszolgálók, stb. útján rendelkezésre álló adatok. Ezért ahhoz, hogy az Internet felhasználók hozzáférhessenek a nyilvános adatokhoz, hozzákapcsolódhatnak a kiszolgálók korlátozott csoportjához, pl. a hálóhoz, az e-mail-hez, de más gazdák, pl. az intranet kiszolgálók adatait nem szabad elérniük.

A nyilvános kiszolgálók az 5 demilitarizált zónában (DMZ) érhetők el, amely a 3 tűzfalhoz csatlakozik. A 3 tűzfal a 6 útvonalválasztó útján az Internettel is összeköttetésben áll, és ebből eredően az 5 demilitarizált zóna csomópontjaival mind a 4 külső hálózatról, pl. az Internetről, mind pedig az 1 belső hálózatról, pl. intranetről összeköttetést lehet teremteni, míg az 1 belső hálózat elérése a 4 külső hálózatról korlátozva van.

A következő leírásban számos részletet ismertetünk, hogy a találmányt alaposabban leírjuk. A szakember számára nyilvánvaló, hogy a találmány a részletes leírás nélkül is felhasználható. Néhány jólismert jellegzetességet nem részletezünk a találmány könnyebb áttekinthetősége kedvéért.

A 2. ábrán a 3 tűzfal egyik kiviteli alakját mutatjuk be a különböző egységekkel, valamint hogy a szűrt csomagok hogyan folynak rajta keresztül a találmány szerint.

Egyszerű esetben a csomag az 1,4,5 hálózatról a 3 tűzfal bemeneti 7 csatlakozójához érkezik, és a kétdimenziós 8 címkereső bemenetére vagy a 8 címkereső kétdimenziós SFT tömbjébe kerül. A közbenső 9 kapcsolat összeköti a 8 címkereső kétdimenziós SFT tömbjét a 10 szabályegyeztető tömbjével, aholis a csomag vagy továbbjut (lefelé), vagy a b5 kimenetnél megállítjuk. A megfelelő működéshez a találmány szerinti 3 tűzfal számos kiegészítő egységgel van ellátva.

Ebben a kiviteli alakban a forráscímek és rendeltetési címek keresése a 8 címkereső kétdimenziós SFT tömbjében megy végbe, eredménye pedig egy szabály vagy inkább szabályok rövid jegyzéke. A szabályjegyzék a 10 szabályegyeztető tömbjében marad mindaddig, amíg a jegyzékben folyik a keresés és az odaillő szabályt meg nem találjuk.

Azt az információt, hogy a csomagot más egységeknek is fel kell-e dolgozni vagy nem, a kétdimenziós SFT kereső hozza létre. Ezeknek a döntéseknek némelyike a szabályegyeztetés alatt keletkezik, ami annyit jelent, hogy a szabályegyeztetés gyakorlatilag a tömbbe lépés előtt megkezdődik, mint ahogyan az a 2. ábrán látható. A 8 címkereső kétdimenziós SFT tömbjét a későbbiekben részletesen ismertetjük.

Ha a csomag túlságosan nagy ahhoz, hogy egy kapcsolat alatt átmenjen, akkor tördeljük, és minden egyes töredéket saját IP fejléccel látunk el. Ez annyit jelent, hogy minden, ami az IP fejlécet követi, darabokra (töredékekre) lesz aprítva. A kiegészítő töredékjelző és a töredékeltolás minden töredékben be van állítva, hogy jelezze hogy az az utolsó töredék, vagy nem, és hogy nyilvántartsa, hogy az eredeti (tördeletlen) csomagban a töredék adatai hová illenek be.

Ha a csomag tördelve van, csak az első töredék, a töredékfejléc tartalmazza a transzport fejlécet (TCP, UDP, vagy ICMP fejlécet). Ez azt jelenti, hogy a következő töredékeket nem lehet olyan szabállyal egyeztetni, amely pl. portokra vonatkozik.

A találmány szerint a 11 tördelő minden egyes tördelt csomagból összegyűjti a töredékeket, amíg a töredékfejléc meg nem érkezik (a töredékek nem szükségszerűen érkeznek sorrendben). Ezután a csak a fejlécben meglévő információ a hozzárendelt töredékcsomaghoz tartozó bejegyzésbe kerül, az összegyűjtött töredékek pedig a 7 csatlakozóval összekötött o1 kimenetre kerülnek, élükön a töredékfejléccel.

Valamennyi töredék, amely a 11 tördelőből továbbjut, el van látva a töredékfejléc információval, tehát a szűrő úgy tudja feldolgozni, mintha csak felaprózatlan csomag volna. A kiegészítő töredékjelzőt és a töredékeltolást ellenőrizzük annak eldöntésére, hogy a csomag a 11 tördelő 7 csatlakozóval összekapcsolt i1 bemenetére kerüljön-e, vagy sem.

Ha a 11 tördelőbe a felaprózott csomag összes töredéke megérkezett, a csomaghoz tartozó bejegyzést el lehet távolítani. Néhány helyen előfordulhat az is, hogy a 11 tördelő a töredékek megállítását határozza el. Ez történik olyankor, ha sérült tördelt csomagok érkeznek (valószínűleg támadás következtében), ha az összegyűjtött töredékek száma bizonyos határt meghalad, vagy egyszerűen hulladékgyűjtés következtében (a régi bejegyzéseket eltávolítjuk, hogy helyet adjunk az újaknak).

Ha egy cégnek sok belső IP címet és csak kevés külső (valódi) IP címet tartalmazó hálózata van, akkor általában a hálózati címfordítót (NAT = Network Address Translation) használja. Az IP címtartomány bizonyos része belső címek számára van fenntartva, pl. 10.*.*.*, 192.168.*.*, és 172.16.*.*. Ezeket a címeket a belső/magán hálózatokon korlátlanul lehet használni. Kifelé azonban soha nem szabad látszaniuk.

Emiatt a tűzfal úgy van kialakítva, hogy a belső forráscímeket külső forráscímekké alakítsa, amint a csomag a belső hálózatról a külső hálózatra megy. Az ellenkező irányba haladó csomagoknál a külső rendeltetési címet belső rendeltetési címre fordítjuk, ahogy a csomag keresztülmegy a tűzfalon. A sok belső cím néhány külső címmé leképezésére a portok is használhatók.

Előfordulhat például, hogy a tűzfal úgy van kialakítva, hogy leképezi a belső címeket 10.1.0.0-tól 10.1.255.255-ig (2**16 cím) külső címekké 194.22.187.0-tól 194.22.187.255-ig (2**8 cím), miközben a 20000-től 20255-ig terjedő portokat (2**8 port) használja.

Ha kapcsolatot kezdeményeznek a 10.1.1.1. cím 4000 portjáról a 130.240.64.46.cím 6000 portjára, akkor a cím és port tartományból úgy kell kiválasztani egy *a* címet és egy *p* portot, hogy (*a*,*p*) egyetlen más NAT kapcsolattal se ütközzék.

Ezután a kapcsolat minden kimenő, belülről kifelé (I2X) csomagjának 10.1.1.1 forráscímét és 4000 portját a vonatkozó *a* és *p* fogja.helyettesíteni. Minden bejövő, kívülről befelé (X2I) csomag *a* rendeltetési címét és *p* portját a megfelelő 10.1.1.1 és 4000 fogja helyettesíteni.

Ilyen módon a 256 külső cím a 256 porttal együtt 65536 címet tud ábrázolni a belső hálózaton.

A kétdimenziós SFT keresés során tájékoztatást kapunk arról is, hogy a csomag alá van-e vetve külsőről belsőre címfordításnak, és hogy a csomag a 12 hálózati címfordító, az X2I-NAT tömb i2 bemenetére kerül-e a külsőről belsőre címfordítás végrehajtására. Ezért a X2I-NAT keresést megtakarítjuk minden olyan csomagnál, amelyik nem igényel fordítást.

Azokat a csomagokat, amelyeknél elvégezzük az X2I-NAT keresést, az s2 lassú útvonal kimeneten keresztül a 13 lassú útvonalra küldjük sikertelenség esetén, minthogy a NAT adatszerkezet frissítése ott következik be. Ha Az X2I-NAT keresés sikerült, a cím és a portok kicserélődnek, és az új forrás-rendeltetés pár szabályegyeztetését keressük vissza, mielőtt a csomagot az o2 kimeneten keresztül a következő szűrésre küldjük.

Tehát a kétdimenziós SFT keresés vagy az X2I-NAT keresés eredményeképpen az is világos, hogy kell-e a csomagot belsőről külsőre (I2X) címfordítani. Ez lényegében az X2I-NAT-tal azonos módon történik, de a legutolsó szűrő lépésként következik be. Az a csomag, amely a 10 szabályegyeztető tömbjének kimeneti 15 kapcsolatáról érkezik és belsőről külsőre (I2X) kell fordítani, a 14 hálózati címfordító, az I2X-NAT tömb i5 bemenetére jut a belsőről külsőre történő címfordítás végrehajtása céljából.

Azok a csomagok, amelyeknél I2X-NAT keresésre van szükség, a 13 lassú útvonalra mennek az s5 lassú útvonal kimeneten keresztül sikertelenség esetén, minthogy a NAT adatszerkezet frissítését

ott lehet elvégezni. Ha az I2X-NAT keresés sikerült, a cím és a portok kicserélődnek, a csomagot pedig a megfelelő hálózatra továbbítjuk az o2 kimeneten és a kimeneti 15 kapcsolaton keresztül.

Annak, hogy a kétdimenziós SFT keresés utáni első lépésként az X2I-NAT következik, míg az I2X-NAT az utolsó lépés, az az oka, hogy a szűrés szabályai az állandó belső címeknek, nem pedig a dinamikus kijelölésű NAT címeknek megfelelően vannak megadva.

A 4 külső hálózatról az 1 belső hálózatra irányuló forgalom legnagyobb részét rendszerint megállítjuk az 1 belső hálózat védelmére. Az 1 belső hálózaton lévő gazdák azonban általában elérhetik a 4 külső hálózaton lévő gazdákat.

Ahhoz, hogy a 4 külső hálózatról visszatérő forgalmat fogadhassuk, a kívülről-befelé megállító szabály alól ideiglenes kivételt kell tennünk, hogy az 1 belső hálózat által kezdeményzett kapcsolatok létrejöhessenek. Ezt lyukasztásnak (HP = hole punching) nevezzük, és azt jelenti, hogy a 3 tűzfalon keresztül a visszatérő csomagok számára lyukat képezünk. A lyuk csak a kapcsolat időtartamára létezik, és kizárólag a kapcsolatból érkező csomagok számára.

A lyukasztás a TCP sorszámokat is nyilvántartja, hogy a lyukasztott kapcsolatokat az eltérítéstől megvédje. Ezért a HP keresést nemcsak a kifelé menő (I2X) csomagokon a 16 I2X-HP tömb által, hanem a befelé jövő (X2I) csomagokon a 17 X2I-HP tömb által is el kell végezni.

A kétdimenziós SFT vagy az X2I-NAT keresés eredményeként tudjuk, hogy kell-e a csomaghoz belülről kifelé (I2X) vagy kívülről befelé (X2I) lyukasztást végezni. Ez annyit jelent, hogy

elkerülhetjük a HP keresés felesleges elvégzését azoknál a csomagoknál, amelyeknél lyukasztás nem merülhet fel. A kimenő csomag, amelyhez lyukasztás szükséges, a 16 I2X-HP tömb i3 bemenetére jut, ahol a forrás és rendeltetési címeket és portokat, továbbá a protokollt végignézzük, hogy megállapítsuk a meglévő állapotot.

Ha ilyen állapot nincs, akkor a csomagot az s3 kimeneten át a 13 lassú útvonalra küldjük, ahol a HP adatszerkezetet frissítjük, és létrehozzuk az állapotot. Ha hozzáillő állapotot találunk, akkor a TCP sorozatszámokat, stb. frissítjük, mielőtt a csomagot továbbküldjük a másik o3 kimeneten a következő szűrési lépéshez.

A X2I-HP-t ugyanilyen módon hajtjuk végre. A bejövő csomag, amelyhez lyukasztani kell, a 17 X2I-HP tömb i4 bemenetére kerül, ahol a forrás és rendeltetési címeket és portokat, továbbá a protokollt végignézzük, hogy meglévő állapotot találjunk. Ha ilyen állapot nem létezik, akkor a csomagot nemlétező lyukon próbáltuk meg átjuttatni, és a megállító szabály révén a csomag a b4 kimenetnél megáll. Ha viszont megfelelő állapotot találunk, akkor frissítjük, mielőtt a csomagot a másik o4 kimeneten keresztül a következő szűrési lépésre küldenénk.

Visszatérve a 8 címkereső kétdimenziós SFT tömbjére, a hálózatok között továbbítandó adatcsomag adatmezőinek tartalmától függően az adatcsomagra alkalmazható szabályt szabályok teljes készletéből kiválasztjuk, ezáltal az adatcsomagot megállítjuk, vagy a tűzfalon keresztül továbbítjuk. A szabálykészlet lineáris keresésésnek csökkentésére a szabálykészletet felosztjuk.

A találmány szerint ezt a csomag kétdimenziós forráscím és rendeltetési címkeresése útján, előtagok halmazában hajtjuk végre, ahol a forráscímmel és a rendeltetési címmel társított előtag megkeresésére a szabályok teljes készletéből minden előtagnak részkészlete van. Ezután az adatcsomagra alkalmazható szabály megkeresésére az adatcsomag adatmezőinek tartalma alapján a 10 szabályegyeztető útján egyeztetjük a szabályokat.

A címek kétdimenziós keresése során minden szabályt úgy tekintünk, hogy kétdimenziós sík téglalap alakú területét fedi le, ahol a téglalap helyét és méretét a cím előtagok és cím hosszúságok határozzák meg. A keresés így azzal a kérdéssel egyenlő, hogy hogyan találhatjuk meg a sík egy pontját körülvevő téglalapot. A keresés egyszerűsítésére még azzal a szigorítással élünk, hogy a sík minden pontját egy és csakis egy téglalap fedheti le, ami által a keresés folyamata egyszerűsödik.

A kétdimenziós keresés végrehajtása után a keresés a megtalált előtaghoz rendelt szabály-részkészlet keresésével folytatódik. A végleges szabályegyeztetéshez azonban a címmezőket nem használjuk. Ha egy szabály az aktuális csomag címeire nem vonatkozik, akkor a a címkeresés eredményeként kapott szabályjegyzéken nem szerepel.

Minthogy minden egyes szabály a teljes címtartomány egy részét lefedő téglalapnak felel meg, egyazon címmel pedig több szabály is egyeztethető, a téglalapok átfedhetik egymást. Ahhoz azonban, hogy a találmány szerinti eljárás megfelelőképpen működjék, az átfedést nem szabad megengednünk. A nem-átfedés követelményének teljesítésére az alábbi lépéseket kell végrehajtanunk:

1. Minden szabályhoz létrehozzuk a téglalapot a címtartományban.

2. Az újonnan kialakított téglalapokból halmazt képezünk. Ezt a halmazt összehasonlító halmaznak nevezzük.

3. A már a síkon lévő téglalapok mindegyikét összehasonlítjuk az összehasonlító halmaz minden egyes téglalapjával.

4. Ha átfedést találunk, a nem-átfedő részeket kivágjuk. Az átfedett részek szabályjegyzékét kijelöljük, az új téglalap szerinti szabályt ennek végére illesztjük.

5. Valamenyi részt, ha eredetileg a síkon lévő téglalap része volt, visszahelyezzük a síkra. Ha nem, hozzáadjuk az összehasonlítandó téglalapok halmazához.

6. Ha az összehasonlító halmaz nem üres, akkor visszatérünk a 3. lépésre. A már korábban a síkon lévő téglalapokat és a már összehasonlított téglalapokat kihagyhatjuk.

7. Ebben az állapotban az összehasonlító halmaz üres. Ha volt olyan téglalap, amelyik egy újat átfedett, azt szükség esetén kisebb részekre osztjuk, a közös részeknek pedig az új szabályt is tartalmazó jegyzékük lesz.

Az átfedés elkerülésének egy másik eljárási módjánál a síkon nem egyszerűen egy téglalaphalmaz van, hanem minden téglalap, koordinátáitól és szabályjegyzék indexétől függetlenül további téglalapok vagy altéglalapok halmazát tartalmazza. Az altéglalapok még további altéglalap-halmazokat tartalmaznak. Néha azonban mégis ugyanarra az altéglalapra kell hivatkozni, és át kell járni a téglalapok mélységének irányított körmentes gráfját (DAG = directed acyclic graph).

Mindig létezik egy olyan "gyökér-téglalap", amely a teljes síkot lefedi. Ez reprezentálja az alapértelmezést, amelyet akkor kell követni, ha minden összehasonlítás negatív. A szabály hatása a konfigurációtól függően a csomag megállítása vagy továbbengedése.

A gyökérnek nevezett téglalap az a gyökér-téglalap, amelyhez új téglalapot hozzáadjuk.

Ha a gyökér és az új téglalapok azonos méretűek, az új téglalapok szabályai hozzáadódnak a gyökér-téglalaphoz rendelt szabályok jegyzékéhez.

Iteráljunk a gyökér-téglalapok valamennyi altéglalapjára. Ha az új téglalapot ezek bármelyike teljesen lefedheti, térjünk vissza az előző lépéshez az altéglalappal, mint gyökértéglalappal, majd térjünk vissza.

Ezután mégegyszer hajtsunk végre iterációt a gyökér-téglalap valamennyi altéglalapjára.

Ha egy altéglalap teljes mértékben bennefoglaltatik az új téglalapban, akkor azt a gyökér-téglalapból áthelyezzük az új téglalapba. Az altéglalap és az összes alatta lévő téglalap szabályjegyzékét úgy kell módosítani, hogy az új téglalap szabályát is magában foglalja.

Ha az altéglalap az új téglalapot metszi, új téglalap keletkezik, amelyik mindkettő közös részeit tartalmazza. Az átmetsző téglalap szabályjegyzéke az eredetiek kombinációja lesz. Ezután az új téglalapot mind az eredeti altéglalaphoz, mind pedig az új téglalaphoz hozzáadjuk.

Ha az összes téglalapot hozzáadtuk a DAG-hoz, a gráfot átjárhatjuk, és létrehozható az előtagok által meghatározott téglalapok jegyzéke, amelyre a kétdimenziós keresést felépítő programnak szüksége van. Az átmetsző téglalap szabályos, előtaggal meghatározott téglalap lesz, de a

környező téglalap maradékát az altéglalapok kivágása után már nem lehet előtagokkal meghatározni.

Ha az adatszerkezetet a fentiek szerint szűrő keresésre használjuk fel, a keresést két lépésben végezzük el. Először kétdimenziós címkeresést hajtunk végre, amelynek eredménye egy egész szám. Az egész szám a szabályok tömbjébe mutató index, ahol minden szabály meghatározza, hogy melyik mezőt kell összehasonlítani, és milyen tevékenység következik, ha egyezést találunk.

Minden szabályhoz tartozik egy következő mező, amely azt jelzi, hogy melyik szabállyal kell folytatni, ha nincs egyezés. A szabályjegyzék átvizsgálását addig folytatjuk, amíg egyezést találunk, amikoris a csomag megállítására vagy továbbítására intézkedünk.

A kétdimenziós előtag feladatot az alábbiak szerint oldjuk meg.

A címtartomány vagy az **U** tér kétdimenziós tér, amely a

$$0 \le s < 2^{32} \text{ és } 0 \le d < 2^{32}$$

feltételt kielégítő (s,d) egész számpárokból áll.

U-nak az $(s,d) \in R$, ha $s_o \le s < s_1$, $d_o \le d < d_1$ feltételt kielégítő R részhalmazát, ahol $(s_o,d_o),(s_1,d_1) \in \mathbf{U}$, téglalapnak nevezzük. Az $[(s_o,d_o),(s_1,d_1)]$ pontpár az R téglalapot egyértelműen meghatározza.

Az $[(s_o,d_o),(s_1,d_1)]$-gyel meghatározott R logikai téglalapot, ahol

$s_1 - s_o = s_1 - 2^{is}*k_s = 2^{is}$ és $d_1 - d_o = d_1 - 2^{id}*k_d = 2^{id}$

valamely i_s, i_d, k_s és k_d nem-negatív egész számra, előtagnak nevezzük.

Ha adott az $(s,d) \in$ **U** pont és az előtagok halmaza **P** = $\{P_1,P_2,...P_n\}$, ahol **P az U**-nak egy partíciója, akkor a kétdimenziós előtag egyeztetési feladat azonos **i** kiszámításának feladatával, ahol $(s,d) \in P_i$.

A tűzfalszűrési kérdés forrás-rendeltetés része kétdimenziós előtag-egyeztetési kérdésnek fogható fel, ahol **P** halmazt megkapjuk, ha az útvonalválasztó táblázatot és a szűrési szabályokat előtagok partíciójává konvertáljuk. Minthogy minden szűrendő csomag előtagját egyeztetni kell, P-nek olyan ábrázolást kell találni, amellyel az előtag-egyeztetés hatékonyan számítható.

A 32 x 32 bites kisméretű teret felosztó előtagok sokaságát tüntettük fel a 3. ábrán. A 18 fekete négyzetek a beállított biteket jelentik (ábrázolások), a 19 fehér négyzetek a be nem állított biteket jelentik. Megjegyezzük, hogy a 0,0 pont a 3. ábrán a bal felső sarokban van.

Valamennyi előtagra P = $[(s_o,d_o),(s_1,d_1)] \in$ **P** és a $p_o = (s_o,d_o)$ pontot P ábrázolásaként választjuk ki. Jelöljük az előtagok ábrázolásainak halmazát **P**-ben a következő módon :

p = $\{p_1,p_2,...p_n\} = \{(s_1,d_1),(s_2,d_2),...(s_n,d_n)\}$.

Ha adott az $(s_d,d_d) \in$ **U** pont, minden $(s,d) \in$ **U**-hoz úgy, hogy $s_d \geq s$ és $d_d \geq d$, akkor (s_d,d_d) dominálo pontja (s,d)-nek, vagy másképpen, (s,d) (s_d,d_d)-vel van dominálva.

Ha adott az $(s_1,d_1),(s_2,d_2) \in \mathbf{U}$ pontpár, akkor a közöttük lévő távolság az L_∞ norma szerint

$$\lim k \to \infty \sqrt[k]{|s_1 - s_2|^k + |d_1 - d_2|^k} = \max(|s_1 - s_2|, |d_1 - d_2|)$$

Ha most adva van a $p = (s,d)$ pont, akkor a hozzátartozó előtagot megkeresni **P**-ben annyi, mint megkeresni **p**-ben az L_∞ norma szerinti legközelebbi p dominálo pontot, vagyis azt a p dominálo pontot $p_i \in$ **p**-ben, amely a p_i és p közötti L_∞ távolságot minimalizálja. Ennélfogva az előtagok helyett elegendő, ha csak a dominálo pontokat ábrázoljuk.

Mint a 4. ábrán látható, a **p** halmaz fogalmilag úgy van ábrázolva, mint egy 2^{32} x 2^{32} pontos bitmátrix, ahol a p bit be van állítva, ha $p \in$ **p**. Az ábrázoláshoz szükséges hely csökkentésére **p**-t aktuálisan úgy ábrázoljuk, mint egy négyszintű 2^{8+8} rendű fát. Minden szintet (ismét) fogalmilag 2^8 x 2^8 bitmátrixként ábrázolunk, ahol az (s,d) bit be van állítva, ha az alatta lévő ágban van dominálo pont. Tehát az 1 szinten (a legfelső szinten) az (s,d) bit ábrázolja **U** $[(2^{24}*s,2^{24}*d),(2^{24}*(s+1),2^{24}*(d+1))]$ téglalapjában a dominálo pont meglétét vagy hiányát.

Egy szint aktuális ábrázolása kétdimenziós sűrű szakasz vagy egyszerűen 2d-szakasz. Hogy egy szintet hogyan és mikor ábrázolhatunk 1d-szakasszal, azt később fogjuk tárgyalni. A 2d-szakasz 32 x 32 mezőből áll, ahol minden mező 8 x 8 bitet jelent. Minthogy a mezőt meghatározó pontok az előtagok dominálo pontjai, nem mind a 2^{64} fajta mező lehetséges. Azzal a szigorítással élünk, hogy csak 677 különböző fajta lehetséges.

Ha a T mezőben létezik egy olyan pont, (egy pont az al-terek valamelyikében, amelyet a mezőben lévő bitek egyike ábrázol), amelynek a legközelebbi dominál pontja egy másik, T_d mezőben van, akkor T valamennyi pontjának a legközelebbi dominál pontjai T_d-ben vannak. A dominál pont meghatározása kiterjeszthető a dominál mezőre. Eszerint a T_d mezőt a T dominál mezőjének nevezzük, vagy a T_d mező dominálja T-t.

Az előző feltétel teljesítéséhez a következő lemmára van szükség.

Ha $P = [(s_o,d_o),(s_1,d_1)]$ az $s_1 - s_o > 1$ feltételt kielégítő előtag, akkor $[(s_o,d_o),(s_o+2^i,d_1)]$ és $[(s_o+2^i,d_o),(s_1,d_1)]$, ahol $s_1 - s_o = 2^i$ valamely nem-negatív i egész számra ugyancsak előtagok. A lemma a másik dimenzióra szimmetrikus.

A fenti lemma segítségével az előtagot két részre vághatjuk, ha kívánjuk. Ebből adódik, hogy a $\mathbf{P}_d$ előtagok halmazát a T_d mezőben lévő ábrázolásukkal annyiszor vághatjuk ketté, amíg minden előtag végpontja ugyanabban a mezőben lesz mindkét dimenzióban, és ezzel teljesíti a fenti feltételt. Ezt a műveletet meződarabolásnak nevezzük, és a sűrű 2d-szakaszok felépítésének a döntő mozzanata.

A mezők különböző fajtáit 7 osztályba soroljuk, amelyeket az 5-11. ábrákon szemléltettünk. Minden mezőosztály bitmátrix alakjában látható (*-gal jelöltük azt a bitet, amely akár 1, akár 0 lehet.) Minden meghatározott értékű (nem *) bithez és mezőosztályhoz vonalak tartoznak, amelyek a bit (pont) által biztosan dominált részhalmaz határait jelzik.

Megfigyelhető, hogy egy mező beállított bitje más mezők pontjait jellemzően jobbra és/vagy lefelé dominálhatja. Osztályonként megadtuk a különböző mezőfajták számát is, és megkülönböztettük a természetes és a korlátozott mezőosztályokat. Végül ismertetjük a mezők ábrázolási/kódolási módját a sűrű 2d-szakaszban.

Az 5. ábrán a (0,0) osztályú mező látható. Egy bit sincs beállítva: természetes, 1 fajta, amelyet mindig dominál a T_d mező az (1,1), (1,2), (2,1), (1,3+) vagy (3+,1) osztályból. Ha egy pont dominálópontját megtaláljuk a (0,0) osztályú mező (s_b, d_b) bitjében, az pontosan olyan, mintha a T_d dominálómező (s_b, d_b) bitjében a megfelelő pont dominálópontját találnánk meg. Tehát a (0,0) osztályú mezőt mindig ugyanúgy lehet és kell kódolni, mint T_d dominálómezőjét.

Az (1,1) osztályú mezőt a 6. ábrán mutatjuk be. Egy bit be van állítva: természetes, 1 fajta, és valószínűleg dominálja a jobbra és lefelé lévő (0,0) osztályú mezőket. Mivel ebben a mezőben minden pontnak ugyanaz a legközelebbi dominálópontja, egyszerűen a mezőn belüli pontra utalva kódolunk.

A 7. ábrán az (1,2) osztályú mezőt tüntettük fel. Az első sorban (D dimenzió) két bit van beállítva: természetes, 1 fajta, és valószínűleg dominálja az alatta lévő (0,0) osztályt. A jobbra lévő (0,0) osztályú mezőket nem tudja dominálni.

Ennek a mezőnek a pontjai között két legközelebbi dominálópont van, az egyik a baloldali félen lévő pontok, a másik a jobboldali félen lévő pontok számára. Mindkét dominálópontra, mint egy 2

hosszúságú tömbre utalva kódolunk, ezután pedig a lekérdezett pontok bal/jobboldalát indexekként használhatjuk.

A 8. ábrán a (2,1) osztályú mezőt szemléltettük. Az első oszlopban (S dimenzió) két bit van beállítva: természetes, 1 fajta, és valószínűleg dominálja a jobbra lévő (0,0) osztályú mezőket. Nem dominálhatja az alatta lévő (0,0) osztályú mezőket. Ebben a mezőben a pontok között két legközelebbi dominál pont van, egy a felső fél pontjai, egy pedig az alsó fél pontjai számára. Az erre a két dominál pontra való hivatkozást 2 hosszúságú tömbként kódoljuk, majd indexül a lekérdezett pont felső/alsó felét használjuk.

Az (1,3+) osztályú mezőt ismertetjük a 9. ábra kapcsán. Három vagy több bit van beállítva az első sorban: természetes, 24 fajta, és valószínűleg dominálja az alatta lévő (0,0) osztályú mezőket. A jobbra lévő (0,0) osztályú mezőket nem tudja dominálni. Ebben a mezőosztályban a pontok között sok dominál pont lehet. A mezőfajta kódolása szükséges, mert itt 24 külöböző mezőfajta található.

Az első sorban minden egyes beállított bithez az alatta lévő dominál ponthoz (ha csak egy ilyen létezik), vagy a következő szinti szakaszhoz (ha több dominál pont van) mutatót kell kódolni. Végül kódoljuk a hivatkozást az első mutatóra (alapmutató).

Ilymódon a lekérdezett pont (s,d) dominál pontja (vagy az utalás a következő szinti szakaszra) úgy található meg, hogy egyszerűen megnézzük, melyik oszlopban van d, és a szakasz fajtáját figyelembevéve táblázatkeresést hajtunk végre, hogy megkapjuk az x mutatóeltolást, végül pedig

megkapjuk az alapmutatótól x mutatóra mutatót. Megfigyelhető, hogy bármelyik következő szinti szakasznak csak egy (D-) dimenziósnak kell lennie, mert a mezőben az összes ábra egyazon S-koordinátán fekszik.

A 10. ábra a (3+,1) osztályú mező. Három vagy több bit van beállítva az első oszlopban: természetes, 24 fajta, és valószínűleg dominálja a jobbra lévő (0,0) osztályú mezőket. Az alatt lévő (0,0) osztályú mezőket nem dominálhatja. Ebben a mezőosztályban a pontok között igen sok dominálo pont fordulhat elő. Ezt a fajta mezőt kódolni kell, mert 24 különböző fajta van.

Az első oszlop minden beállított bitjéhez mutatót kódolunk az alatta lévő dominálo pontra (ha csak egy van), vagy a következő szinti szakaszra (ha több dominálo pont létezik). Végül kódoljuk a hivatkozást az első mutatóra (alapmutató).

A lekérdezett (s,d) pont dominálo pontját (vagy a következő szinti szakasz hivatkozását) megtalálhatjuk úgy, hogy egyszerűen átnézzük, melyik sorban található meg s, és a szakasz fajtájával együtt táblázatkeresést végzünk, hogy megtaláljuk az x mutató helyét, végül pedig megtaláljuk az alapmutatótól x mutatónyira lévő mutatót. Bármelyik következő szinti szakasz csak egy (S-) dimenziós lehet, mert a mező összes ábrázolása azonos D-koordinátán fekszik.

A 11. ábra a (2+,2+) osztályú mező. Két vagy több bit van beállítva az első sorban és az első oszlopban: korlátozott, 625 fajta, más mezőt nem dominálhat és más mező nem dominálhatja. Ebben a mezőosztályban jellemzően sok dominálo pont található. Pontosan úgy kódoljuk, mint az (1,3+) és (3+,1) osztályú mezőket.

Mielőtt azonban az aktuális kódolást elkezdenénk, korlátozást vezetünk be, hogy a különböző fajták számát csökkentsük. Az első feladat a 8 Definíció szerintihez hasonló mezőkorlátozás bevezetése minden biten. Ezután kiszámítjuk a 8 hosszúságú S_v és D_v bitvektorpárokat, ahol

S_i = 1, ha az i-edik sorban van beállított bit, és

0, ha nincs

D_i = 1, ha az i-edik oszlopban van beállított bit, és

0, ha nincs.

Végül S_v és D_v^T szorzatából mátrixszorzás útján új mezőt hozunk létre.

Mint az (1,3+) és (3+,1) osztályok esetében, ebben az esetben is egydimenziós alszintek adódhatnak. Ellenőrizzük, hogy az egy ábrázolásnál többet tartalmazó bitek összes ábrázolásai **U** egy sorában vannak-e, ami azt jelenti, hogy a S-dimenzió összeomlik, vagy **U** azonos oszlopában, amikor a D-dimenzió omlik össze.

A következőkben a tűzfalaknál a NAT és HP bejegyzések ábrázolására használt adatszerkezeteket ismertetjük részletesebben.

A keresés kulcsaként mindkét esetben az *saddr* és *daddr* IP címeket, *sport* és *dport* portpárokat, és a feldolgozott csomag *proto* protokollját használjuk fel. A keresés első lépéseként kiszámítjuk a feldarabolási értéket. Ezt nagyon egyszerű és gyors utasításokkal, pl. biteltolással és bitenkénti

logikai műveletekkel hajtjuk végre. A feldarabolás értékét indexként használva 16 bites mutatót nyerhetünk egy nagy tömbből (a feldaraboló táblázatból).

A mutató vagy 0, ami azt jelenti, hogy a keresés sikertelen (üres), vagy a Patricia fa gyökerére mutat, amely a kulcsok kis halmazánál igen hatásos adatszerkezet. Ha a mutató Patricia fára utal, akkor az *saddr*, *daddr*, *sport*, *dport* és *proto* bitminták sorbaállításával kulcsot képezünk. A Patricia fa keresése során a kulcsot a következő bekezdésben leírtak szerint használjuk.

A Patricia fa bináris fa, amely a lekérdezett kulcsokat bittömbökként kezeli, és az elágazások irányítására minden belső csomópontban bitindexet használ. A keresés során gyökerétől a levélig végighaladunk a fán. Amikor elérünk egy *f* bitindexszel rendelkező csomóponthoz, megvizsgáljuk a lekérdezett kulcs *i* bitjét, hogy a keresést balra (ha a bit 0), vagy jobbra (ha a bit 1) lévő ágon folytassuk-e.

A haladás végetér, ha levélhez érkezünk. Ezután annak eldöntésére, hogy a lekérdezett kulcs jelen van-e a táblázatban, vagy nem, a lekérdezett kulcsot összehasonlítjuk az abban a levélben tárolt kulccsal. Ha a két kulcs egyenlő, a keresés sikeres volt.

A 12. ábra szerinti példa a 001111 lekérdezett kulcs sikertelen keresése a hat kulcsot tartalmazó Patricia fán. A végighaladás során a 0,2 és 3 számú biteket vizsgáljuk, a vizsgálat a 011101 kulcsot tartalmazó levélnél fejeződik be. A lekérdezett kulcsot és a levélkulcsot összehasonlítva azt találjuk, hogy az 1 számú bit nem egyezik.

A belső csomópontokban tárolt bitindexek tekintetében a Patricia fa rendezett. Vagyis a gyökér kivételével minden belső csomópontnak nagyobb bitindexe van, mint az elődeinek. Ebből következik, hogy minden *i* bitindexszel ellátott csomópontból kiinduló ágon tárolt kulcs az *i*-1 bitig, beleértve az *i*-1-et is, azonos.

A beillesztést úgy végezzük el, hogy először végrehajtjuk a sikertelen keresést, majd a lekérdezett kulcs és a levélkulcs összehasonlítása során először a nem egyező bit *i* indexét feljegyezzük. Ekkor két új csomópontot hozunk létre, az *i* indexszel ellátott belső csomópontot és a lekérdezett kulcshoz a levélcsomópontot.

Attól függően, hogy a lekérdezett kulcs *i*-edik bitje 0 vagy 1, a levelet a baloldali vagy jobboldali ágban tároljuk a belső csomóponthoz lépest. Ha hivatkozási mezőként a másik ágmezőt használjuk, akkor a belső csomópontot közvetlenül a gyökértől a levélig megtett út *i*-nél nagyobb legkisebb bitindexszel ellátott csomópontja fölé illesztjük be.

A 13. ábra Patricia fája a 12. ábra szerinti korábbi sikertelen keresés után a kulcs beillesztésével keletkezik. Új, 1 bitindexszel ellátott belső csomópont keletkezett, amelyet a gyökértől kiinduló úton a 0 és 2 bitindexű csomópontok közé illesztettünk be.

Az a Patricia feldarabolás, amelyet a lyukasztásnál használunk, pontosan ugyanígy működik, tehát egy egyszerű feldarabolási táblázat keresést követő Patricia fa keresésből áll. A levelet legtöbbször közvetlenül elérjük, ami azt jelenti, hogy a paraméterekből nem kell bittömböt felépíteni, hanem

ezek a Patricia leveleket tartalmazó/jelentő szerkezet megfelelő mezőivel közvetlenül is összehasonlíthatók.

Megadunk egy olyan *hp_lookup*(*iaddr*, *xaddr*, *iport*, *xport*, *proto*) keresőfüggvényt, amelyet mind az I2X-HP, mind pedig az X2I-HP kereséshez használunk. Az egyetlen különbség közöttük a paraméterek megadásának sorrendje. Az I2X-HP-hez a függvény behívása *hp_lookup*(*saddr*, *daddr*, *sport*, *dport*, *proto*), míg az X2I-HP-hez a hívás *hp_lookup*(*daddr*, *saddr*, *sport*, *dport*, *proto*).

A keresőfüggvény hivatkozást ad vissza arra a szerkezetre, amely a Patricia levélkulcsot tartalmazza, azaz *iaddr*, *xaddr*, *iport*, *xport*, *proto*, és még további mezőket, amelyek a kapcsolat állapotát, pl. a TCP sorozatszámokat jelzik.

A Patricia feldarabolás a NAT-hoz kissé bonyolultabb, mint a HP-hez. Ennek az az oka, hogy három különböző cím és port van: *iaddr, naddr, xaddr, iport, nport, xport* a HP-vel szemben, ahol csak két cím és port fordult elő. Ezt annyit jelent, hogy az I2X és X2I közötti különbség is valamivel bonyolultabb annál, minthogy a kereséskor a címeket és portokat egyszerűen felcserélnénk.

A feladatot úgy oldjuk meg, hogy engedjük a feldarabolási érték legalacsonyabb helyiértékű bitjét visszatükröződni, ha a keresés I2X vagy X2I (ez lényegében olyan, mintha két feldarabolási táblázatot használnánk). A NAT kapcsolat Patricia levélkulcsokat tartalmazó szerkezete ugyanaz az I2X-nél és az X2I-nél, de mindahárom címet és portot tartalmazza.

Két keresőfüggvény van, *nat_i2x_lookup(saddr, daddr,sport, dport, proto)* és *nat_x2i_lookup(saddr, daddr,sport, dport, proto)*. A feldarabolási érték kiszámításához az argumentumot mindkét függvény használja, a legalacsonyabb helyiértékű bitet pedig ennek megfelelően beállítja. Ha az eredményül kiadódó mutató Patricia csomópontra (belső csomópontra) utal, a címeket, portokat és protokollt sorbaállítjuk, hogy a Patricia fa végigjárásához szükséges bittömböt kapjunk. Ha elértük a levélszerkezetet, a címeket, portokat és protokollt összehasonlítjuk a levél megfelelő mezőivel.

Ha a csomagot I2X-NAT-tal fordítjuk:

saddr-et (a csomagban) *iaddr*-rel (a levélszerkezetben)

daddr-et *xaddr*-rel

sport-ot *iport*-tal

dport-ot *xport*-tal

proto-t *proto*-val hasonlítjuk össze.

Ha ezek mindegyike egyezik, a keresés sikeres,és a csomag *saddr* és *sport* forráscímét és portját a levélszerkezet *naddr* és *nport* értékeivel helyettesítjük, mielőtt a csomagot továbbítanánk.

Ha a csomagot X2I-NAT-tal fordítjuk:

saddr-et (a csomagban) *xaddr*-rel (a levélszerkezetben)

daddr-et *naddr*-rel

sport-ot *xport*-tal

dport-ot *nport*-tal

proto-t *proto*-val hasonlítjuk össze.

Ha ezek mindegyike egyezik, a keresés sikeres, és a csomag *daddr* és *dport* rendeltetési címét és portját a levélszerkezet *iaddr* és *iport* értékeivel helyettesítjük, mielőtt a csomagot a következő feldolgozó lépésre továbbítanánk.

A HP és NAT adatszerkezetek frissítését az EffNIX (korábban NetBSD) kernelje hajtja végre, amely a BSP-n (1 processzor) fut, de a keresések legtöbbjét az AP-n (2 processzor) futó továbbító kernel végzi.

A HP adatszerkezetnek csak egy előfordulása van, és a NAT adatszerkezetnek is csak egy előfordulása van. Ezek a megosztott memóriában vannak, és a két processzor egyidejűleg érheti el őket. Ez igen érdekes szinkronozási problémát vet fel : egy írásit és egy olvasásit.

A szinkronozást úgy oldjuk meg, hogy mielőtt bármit megváltoztatnánk (felülírnánk), a frissítő rutinokkal érvényteleníttetjük a levélszerkezeteket és csomópontokat. A kereső rutinok ellenőrzik, hogy az elért levelek és csomópontok érvényesek-e az elérés előtt és után, és azt is, hogy az elérés során nem változtak.

Versenyhelyzetben, ha észleljük (valamennyi veszélyes versenyhelyzet feltételt nyomonkövetjük), a keresés sikertelen, és a csomagot további kezelésre a BSP-re küldjük (a feldolgozást követő sikeres keresés vagy az adatszerkezet frissítése céljából).

Nyilvánvaló, hogy a találmány szerinti tűzfal hálózati adatcsomagok belső hálózatok és külső hálózatok közötti forgalmának vezérlésére maradéktalanul megfelel a kitűzött célnak, és a fent kimutatott előnyökkel jár.

Noha a találmányt meghatározott kiviteli alak kapcsán ismertettük, a találmány további kiviteli alakok megvalósítására is alkalmas, a jelen ismertetés pedig a találmány elveinek példaszerinti ismertetésének tekintendő, a terjedelmet pedig semmiképpen nem korlátozza a bemutatott példára.

SZABADALMI IGÉNYPONTOK

1. Tűzfal hálózati adatcsomagok belső hálózatok és külső hálózatok közötti forgalmának vezérlésére,

amely tűzfal a hálózatok között továbbítandó adatcsomag adatmezőinek tartalmától függően az adatcsomagra alkalmazható szabályt - szabályok teljes készletéből - kiválasztó, az adatcsomagot megállító vagy a tűzfalon keresztül továbbító szűrővel rendelkezik, azzal j e l l e - m e z v e, hogy

a forráscímmel és a rendeltetési címmel társított előtag ábrázolás utáni megkeresésére az adatcsomag forráscímét és rendeltetési címét címek előtagjainak halmazában két dimenzióban kereső kétdimenziós címkeresőt (8) tartalmaz,

a szabályok teljes készletéből mindegyik előtaghoz részkészlet tartozik,

az adatcsomagra alkalmazható szabály megkeresésére pedig az adatmező tartalma alapján szabály egyeztetésére alkalmas szabályegyeztetővel (10) van ellátva.

2. Az 1. igénypont szerinti tűzfal azzal j e l l e m e z v e, hogy a kétdimenziós címkereső (8) a forráscímmel és a rendeltetési címmel társított előtagot **p**-ben az $L\infty$ norma szerinti legközelebbi p dominált pont, azaz $pi \in$ **p**-ben a pi és p közötti $L\infty$ távolságot minimalizáló p dominált pont meghatározása útján megkereső eszközt tartalmaz.

3. A 2. igénypont szerinti tűzfal azzal j e l l e m e z v e, hogy

a forráscímet és a rendeltetési címet az $(s,d) \in$ **U** pont ábrázolja, ahol **U** a

$0 \le s < 2^{32}$ és $0 \le d < 2^{32}$

feltételt kielégítő (s,d) egész számpárokkal ábrázolt kétdimenziós címtartomány, és ahol,

az előtagok halmaza **P** = $\{P_1,P_2,..P_n\}$az **U** címtartomány partíciója,

mindegyik Pi előtag az **U** címtartományban $[(s_o,d_o),(s_i,d_i)]$-vel meghatározott R logikai téglalap, ahol

$s_1 - s_o = s_1 - 2^{is}*k_s = 2^{is}$ és $d_1 - d_o = d_1 - 2^{id}*k_d = 2^{id}$

valamely i_s,i_d,k_s és k_d nem-negatív egész számra,

az R logikai téglalap az **U** címtartomány részhalmaza, amelyre teljesül, hogy

$(s,d) \in R$, ha

$s_o \le s < s_1$, *és* $d_o \le d < d_1$,

ahol $(s_o,d_o),(s_1,d_1) \in$ **U**, és az $[(s_o,d_o),(s_1,d_1)]$ pontpár az R téglalapot egyértelműen meghatározza.

4. A 2. vagy 3. igénypont szerinti tűzfal azzal j e l l e m e z v e, hogy mindegyik előtag

$P = [(s_o,d_o),(s_1,d_1)] \in$ **P**, a $p_o = (s_o,d_o)$ pont P ábrázolása, és

p = $\{p_1,p_2,...p_n\} = \{(s_1,d_1),(s_2,d_2),...(s_n,d_n)\}$

az előtagok ábrázolásainak halmaza **P**-ben, ahol minden $(s,d) \in$ **U**-hoz adva van az $(s_d,d_d) \in$ **U** pont, ahol

$s_d \ge s$ és $d_d \ge d$, (s,d)-t pedig (s_d,d_d) dominálja..

5. A 3. igénypont szerinti tűzfal azzal j e l l e m e z v e, hogy az

$(s_1,d_1),(s_2,d_2) \in$ **U** adott pontpár közötti távolság az L_∞ norma szerint

$\lim k \to \infty \sqrt[k]{|s1 - s2|^k + |d1 - d2|^k} = \max(|s_1 - s_2|, |d_1 - d_2|)$

által van adva.

6. Az 1-5. igénypontok bármelyike szerinti tűzfal azzal j e l l e m e z v e, hogy felaprózott adatcsomag töredékeket az adatcsomag töredékfejlécének beérkeztéig összegyűjtő tördelőt (11), az adatcsomag töredékfejléc mezőjébe foglalt adatokat bemeneti eszközben tároló töredékfejléc tárolót, az adatcsomag töredékfejléccel kezdődő, töredékfejléc-adatokkal ellátott töredékeit továbbküldő töredéktovábbítót, valamint mindegyik töredéket szabályos felaprózatlan adatcsomagként feldolgozó szűrőt tartalmaz.

7. Az 1-6. igénypontok bármelyike szerinti tűzfal azzal j e l l e m e z v e, hogy az előtagba foglalt adatoktól függően a tűzfalon (3) keresztül kifelé továbbított adatcsomag belső forráscímeit külső forráscímekké vagy a tűzfalon (3) keresztül befelé továbbított adatcsomag külső forráscímeit belső forráscímekké alakító hálózati címfordítóval (12,14) rendelkezik.

8. Az 1-6. igénypontok bármelyike szerinti tűzfal azzal j e l l e m e z v e, hogy az előtagba foglalt adatoktól függően a belső hálózatról (1) a külső hálózatra (4) továbbított adatcsomag belső forráscímeit külső forráscímekké vagy a külső hálózatról (4) a belső hálózatra (1) továbbított adatcsomag külső forráscímeit belső forráscímekké alakító hálózati címfordítóval (12,14) rendelkezik.

9. Az 1-8. igénypontok bármelyike szerinti tűzfal azzal j e l l e m e z v e, hogy a belső hálózat által kezdeményezett kapcsolathoz az adatcsomag ideiglenes kivételét a kívülről-befelé megállító szabály alól az előtagba foglalt adatok alapján elrendelő lyukasztó berendezéssel (16,17) van

ellátva, a külső hálózatról (4) a belső hálózatra (1) továbbított adatcsomagok számára pedig a kapcsolat tartamára a tűzfalon (3) visszatérő csatorna vezet át.

10. Az 1-9. igénypontok bármelyike szerinti tűzfal azzal j e l l e m e z v e, hogy a felaprózott adatcsomag töredékeit az adatcsomag töredékfejlécének beérkezte előtt összegyűjtő tördelőt (11), az adatcsomag töredékfejléc mezőjébe foglalt adatokat bemeneti eszközben tároló töredékfejléc tárolót, az adatcsomag töredékfejléccel kezdődő, töredékfejléc-adatokkal ellátott töredékeit továbbküldő töredéktovábbítót, valamint mindegyik töredéket szabályos felaprózatlan adatcsomagként feldolgozó szűrőt tartalmaz.

11. Eljárás hálózati adatcsomagok belső hálózatok és külső hálózatok közötti, tűzfalon keresztülmenő forgalmának vezérlésére, melynek során

a hálózatok között továbbítandó adatcsomag adatmezőinek tartalmától függően az adatcsomagra alkalmazható szabályt szabályok teljes készletéből kiválasztjuk,

a szabályt az adatcsomaggal egyeztetjük,

az adatcsomagot pedig a szabálytól függően szűrő segítségével megállítjuk, vagy a tűzfalon keresztül továbbítjuk, azzal j e l l e m e z v e, hogy a szűrés során

a forráscímmel és a rendeltetési címmel társított előtag ábrázolás utáni megkeresésére az adatcsomag forráscímét és rendeltetési címét címek előtagjainak halmazában kereső kétdimenziós keresést hajtunk végre, a szabályok teljes készletéből mindegyik előtaghoz részhalmazt rendelünk,

az adatcsomagra alkalmazható szabály megkeresésére pedig az adatcsomag adatmezőinek tartalma alapján a szabályok részkészletén szabályegyeztetést hajtunk végre.

12. A 11. igénypont szerinti eljárás azzal j e l l e m e z v e, hogy az adatcsomagra alkalmazható szabály kiválasztása előtt

felaprózott adatcsomag töredékeket az adatcsomag töredékfejlécének beérkeztéig összegyűjtünk,

az adatcsomag töredékfejléc mezőjébe foglalt adatokat bemeneti eszközben tároljuk,

az adatcsomag töredékfejléccel kezdődő töredékfejléc-adatokkal ellátott töredékeit továbbküldjük,

a töredékeket pedig szűrő segítségével szabályos felaprózatlan adatcsomagként dolgozzuk fel.

13. A 11. vagy 12. igénypont szerinti eljárás azzal j e l l e m e z v e, hogy a szabályegyeztetés végrehajtása előtt az előtagba foglalt adatoktól függően a tűzfalon (3) keresztül befelé továbbítandó adatcsomag külső forráscímét belső forráscímmé alakítjuk.

14. A 11-13. igénypontok bármelyike szerinti eljárás azzal j e l l e m e z v e, hogy a szabályegyeztetés végrehajtása előtt az előtagba foglalt adatoktól függően a külső hálózatról (4) a belső hálózatra (1) továbbítandó adatcsomag külső forráscímét belső forráscímmé alakítjuk.

15. A 11-14. igénypontok bármelyike szerinti eljárás azzal j e l l e m e z v e, hogy

az előtagba foglalt adatoktól függően a tűzfalon (3) keresztül kifelé továbbítandó adatcsomag belső forráscímét külső forráscímmé alakítjuk.

16. A 11-15. igénypontok bármelyike szerinti eljárás azzal j e l l e m e z v e, hogy

az előtagba foglalt adatoktól függően a belső hálózatról (1) a külső hálózatra (4) továbbítandó adatcsomag belső forráscímét külső forráscímmé alakítjuk.

17. A 11-16. igénypontok bármelyike szerinti eljárás azzal j e l l e m e z v e, hogy a szabályegyeztetés végrehajtása előtt a belső hálózat (1) által kezdeményezett kapcsolathoz az adatcsomag ideiglenes kivételét a kívülről-befelé megállító szabály alól az előtagba foglalt adatok alapján elrendeljük, és ebben az esetben a külső hálózatról (4) a belső hálózatra (1) továbbított adatcsomagok számára a kapcsolat tartamára a tűzfalon (3) át visszatérő csatornát létesítünk.

18. A 11-17. igénypontok bármelyike szerinti eljárás azzal j e l l e m e z v e, hogy az adatcsomagra alkalmazható szabály kiválasztása előtt

az adatcsomag töredékfejlécének beérkeztéig felaprózott adatcsomag töredékeket gyűjtünk össze,

az adatcsomag töredékfejléc mezőjébe foglalt adatokat bemeneti eszközben tároljuk,

az adatcsomagnak a töredékfejléccel kezdődő, töredékfejléc-adatokkal ellátott töredékeit továbbküldjük,

a töredékeket pedig szűrő segítségével szabályos felaprózatlan adatcsomagként dolgozzuk fel.

19. A 11-18. igénypontok bármelyike szerinti eljárás azzal j e l l e m e z v e, hogy az adatcsomag forráscímének és rendeltetési címének kétdimenziós keresése során **p**-ben az L_∞ norma szerinti legközelebbi p dominálo pontot, azaz $p_i \in$ **p**-ben a p_i és p közötti L_∞ távolságot minimalizáló p dominálo pontot meghatározzuk.

20. A 19. igénypont szerinti eljárás azzal j e l l e m e z v e, hogy

a forráscímet és a rendeltetési címet az $(s,d) \in \mathbf{U}$ ponttal ábrázoljuk, ahol **U** a

$0 \le s < 2^{32}$ és $0 \le d < 2^{32}$

feltételt kielégítő (s,d) egész számpárokkal ábrázolt kétdimenziós címtartomány, amelynél

az előtagok halmaza $\mathbf{P} = \{P_1,P_2,..P_n\}$ az **U** címtartomány partíciója,

mindegyik P_i előtag az **U** címtartományban $[(s_o,d_o),(s_i,d_i)]$-vel meghatározott R logikai téglalap, ahol

$s_1 - s_o = s_1 - 2^{is} * k_s = 2^{is}$ és $d_1 - d_o = d_1 - 2^{id} * k_d = 2^{id}$

valamely i_s, i_d, k_s és k_d nem-negatív egész számra,

az R logikai téglalap az **U** címtartomány részhalmaza, és

$(s,d) \in R$, ha $s_o \le s < s_1$, $d_o \le d < d_1$, ahol $(s_o,d_o),(s_1,d_1) \in \mathbf{U}$, és az $[(s_o,d_o),(s_1,d_1)]$ pontpár az R téglalapot egyértelműen meghatározza, mindegyik előtag

$P = [(s_o,d_o),(s_1,d_1)] \in \mathbf{P}$, az (s_o,d_o) pont P ábrázolása, és

$\mathbf{p} = \{p_1,p_2,...p_n\} = \{(s_1,d_1),(s_2,d_2),...(s_n,d_n)\}$

az előtagok ábrázolásainak halmaza **P**-ben, ahol minden $(s,d) \in$ **U**-hoz adva van az $(s_d,d_d) \in \mathbf{U}$ pont, és ahol

$s_d \ge s$ és $d_d \ge d$, (s,d)-t (s_d,d_d) dominálja,

az $(s_1,d_1),(s_2,d_2) \in \mathbf{U}$ adott pontpár közötti távolságot pedig az L_∞ norma szerint

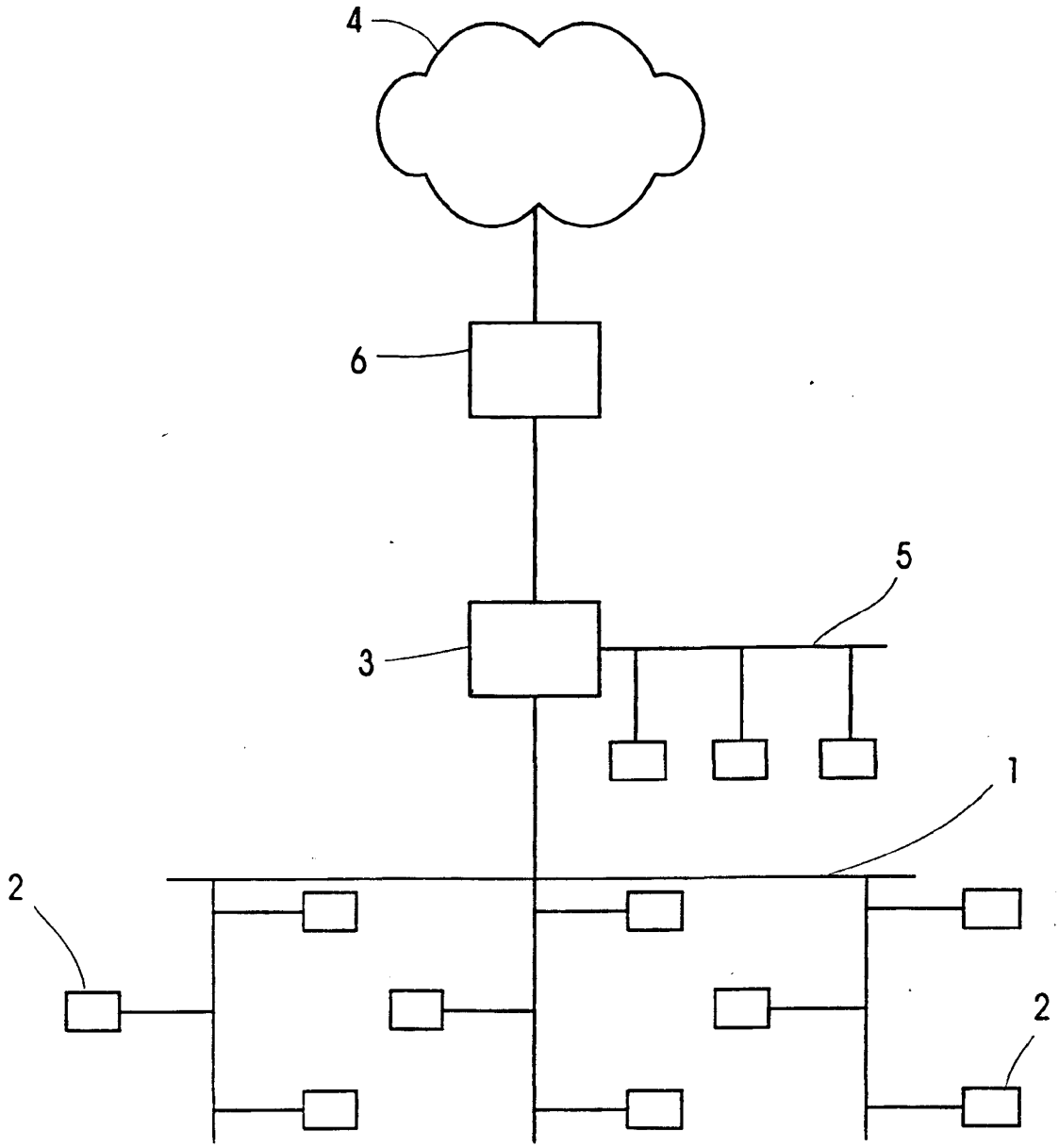
$\lim k \to \infty \sqrt[k]{|s_1 - s_2|^k + |d_1 - d_2|^k} = \max(|s_1 - s_2|, |d_1 - d_2|)$

által adjuk meg.

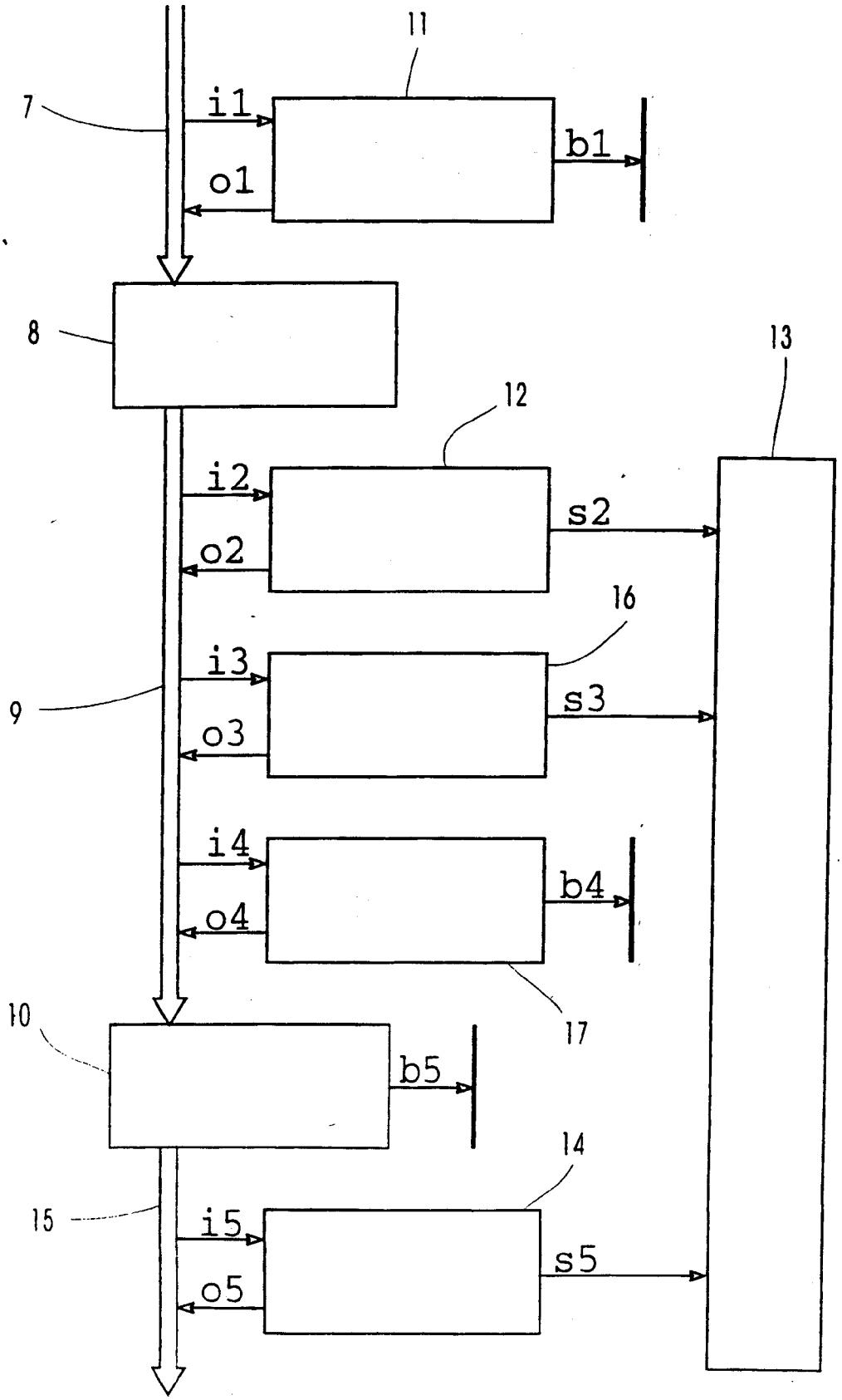
A meghatalmazott:

7 rajz, 13 ábra

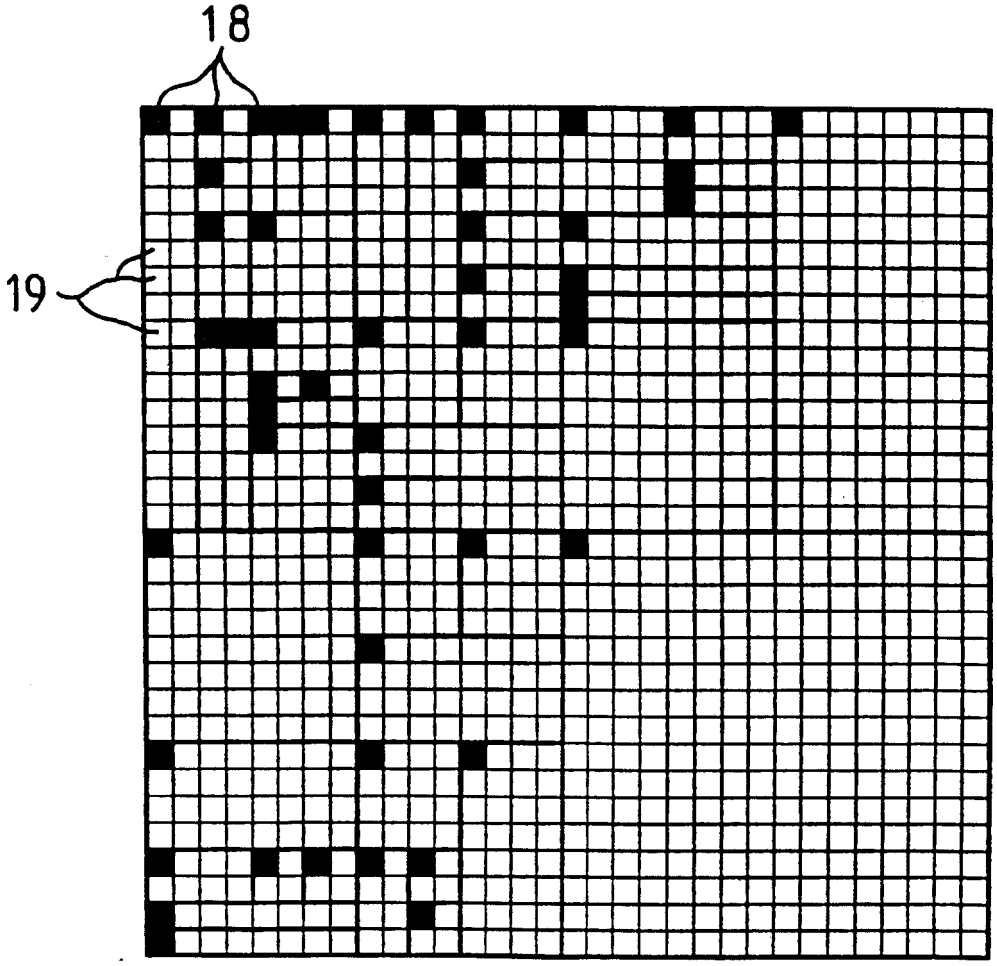
KÖZZÉTÉTELI PÉLDÁNY



1. ábra



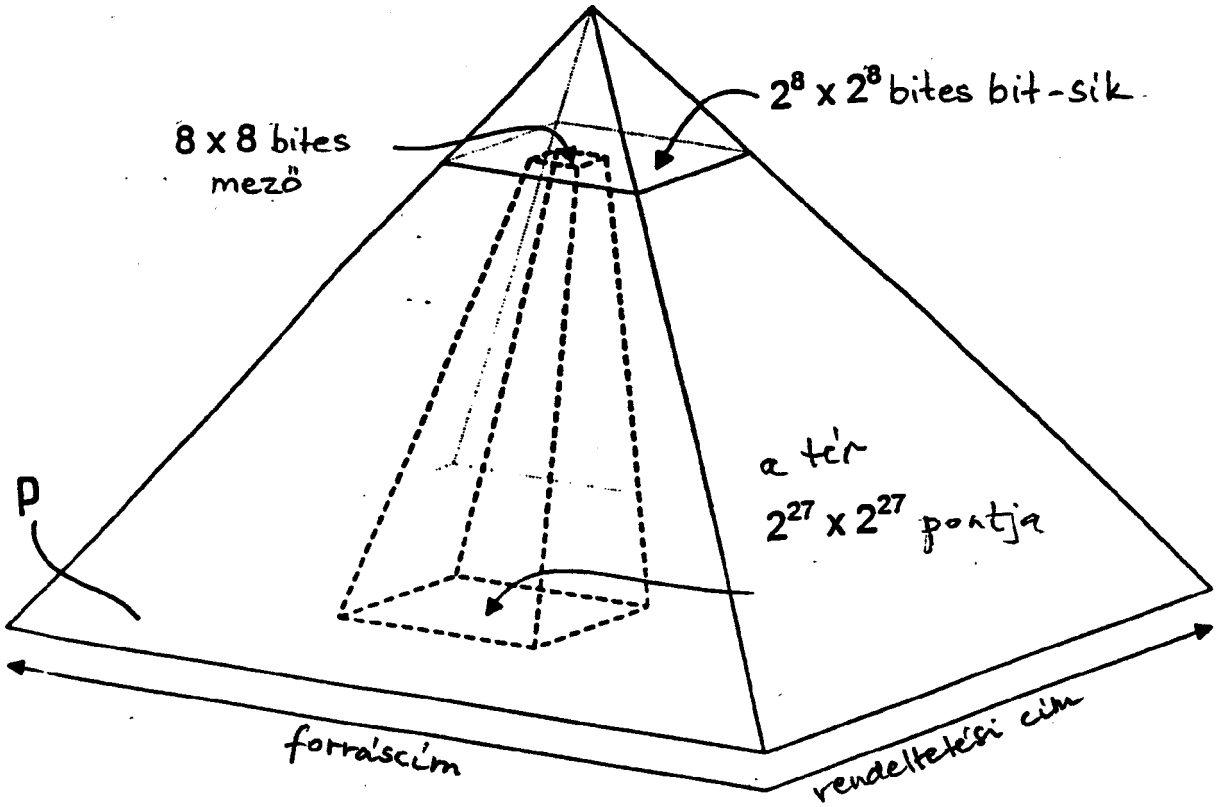
2. ábra



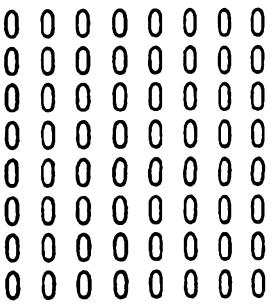
3. ábra

7/4

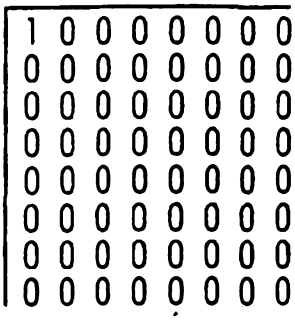
KÖZZÉTÉTELI
PÉLDÁNY



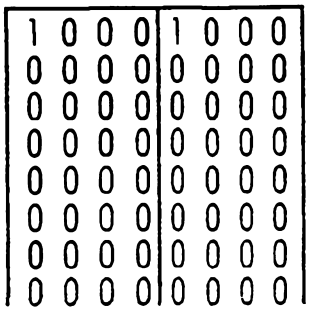
4. ábra



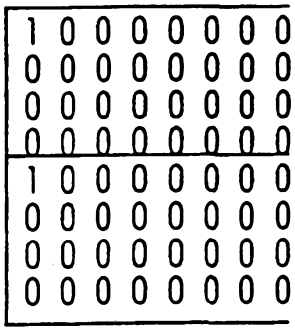
5. ábra



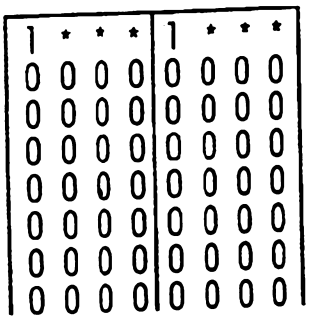
6. ábra



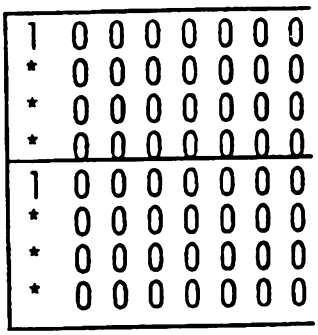
7. ábra



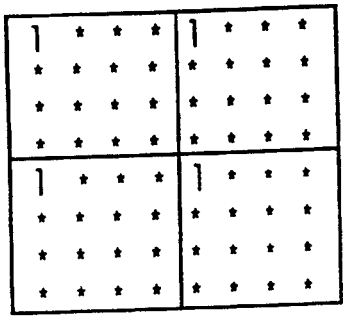
8. ábra



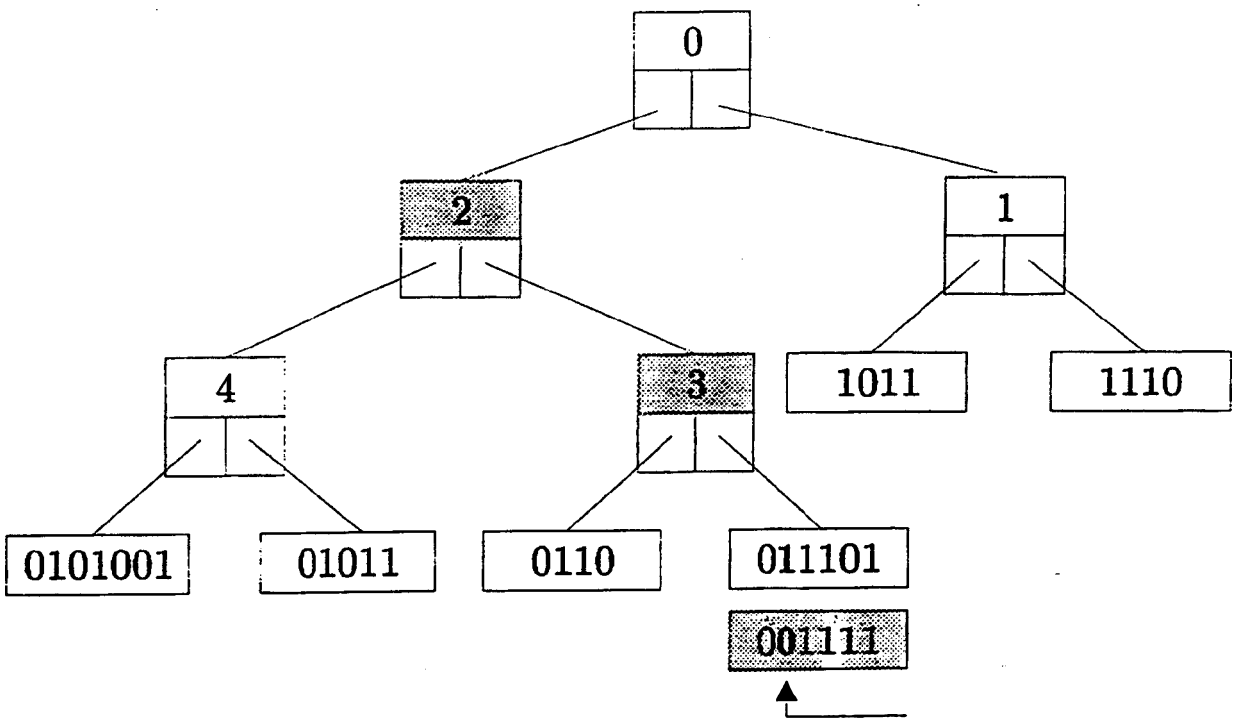
9. ábra



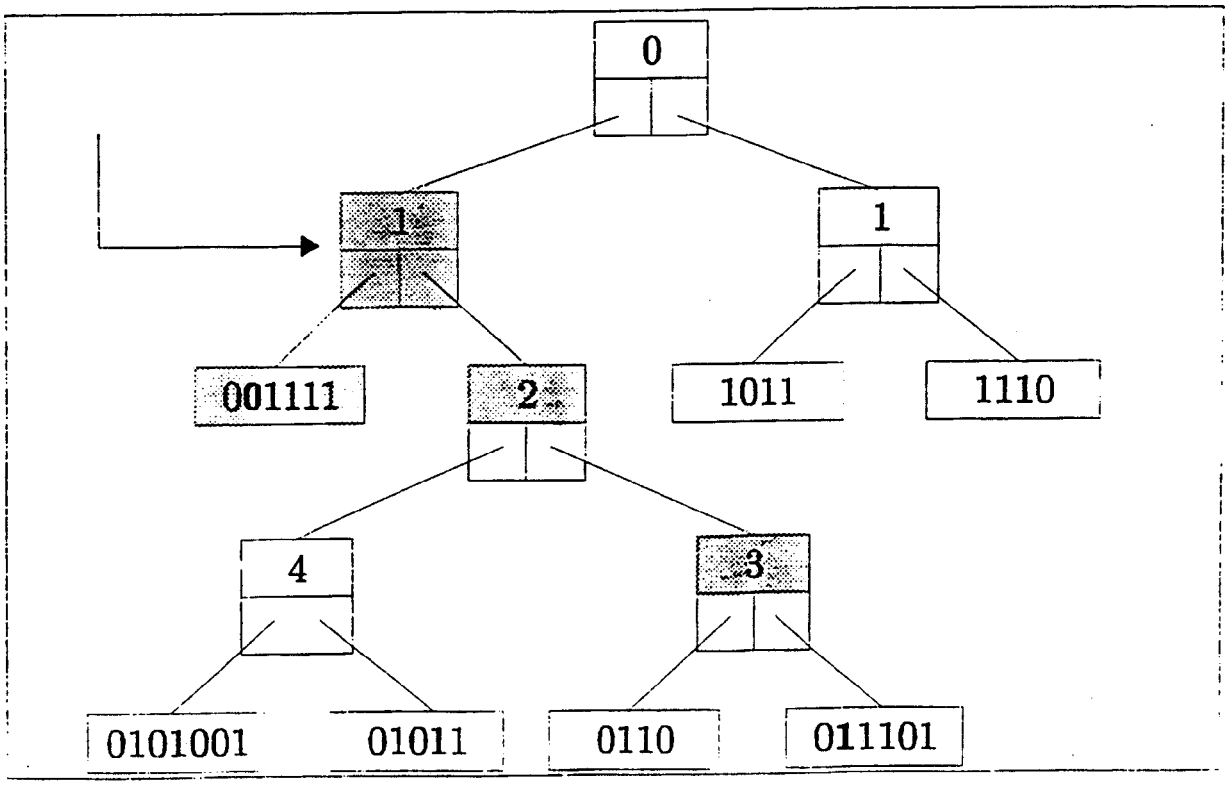
10. ábra



11. ábra



12. ábra



13. ábra