

14. fejezet – Többszörös hozzáférésű protokollok 2

Időszeletelt, vagy réselt ALOHA

Az egyszerű ALOHA kihasználtsággal kapcsolatos problémái leginkább arra vezethetők vissza, hogy az adók bármely időpillanatban kezdeményezhettek adást. Nyilvánvaló, hogy ha csak 1db adó van, akkor objektíve nincs kivel ütköznie. Az egyszerű ALOHA indulásakor ugyan nem volt nagy az adatátviteli igény, de a hatékonyság növelése, mint igény gyorsan megjelent, és szerencsére szintén gyorsan, pár éven belül megoldódni is látszott. Lawrence G. Roberts (aki az ARPANET-et kifejlesztő csoport egyik tagja volt) 1972-ben publikálta azt a módszert, amivel a csatorna áteresztő képessége duplájára, azaz 36%-ra növekedhetett.

Az új módszer abból állt, hogy az idő diszkrét szeletekre lett bontva, és az időrészek kezdetét a központi gép által kibocsájtott szinkronjel jelzi. Egy időrésben csak egy üzenet lehet, tehát az időrés egyben a keretek hosszát is meghatározza.

CSMA

(Carrier Sense Multiple Access / Vivőjel érzékeléses többszörös hozzáférés)

A cél természetesen a csatorna áteresztő képességének további növelése, amit a csatorna figyelésével érhetünk el. Ez a módszer leginkább vezetékes összeköttetések esetében használatos, ahol az állomások egyszerűbben érzékelhetik egymás tevékenységét, és könnyebben képesek egymáshoz alkalmazkodni, azaz képesek ellenőrizni a foglaltságot, és foglaltság esetén nem kezdeményeznek adást.

A CSMA protokollnak több módozata is ismert:

[A perzisztens jelentése hosszútávon tartós, ellenálló. Az informatikában jellemzően olyan adatra használjuk, mely túléli az őt létrehozó folyamatot.]

- 1-perzisztens CSMA

Az állomás, amelyik adni készül, először belehallgat a csatornába. Amennyiben a csatorna nem foglalt, elkezd az adást. Amennyiben a csatorna foglalt akkor várakozni kezd, amíg a csatorna szabad nem lesz. Amennyiben a küldéskor ütközés lép fel (pl. „A” állomás éppen ad, „B” és „C” állomások egymás után belehallgatnak, majd „A” adásának végén egyszerre elkezdenek adni.), az elküldött keretet az adó egy véletlen hosszúságú ideig vár, majd újra kezdi a küldési procedúrát.

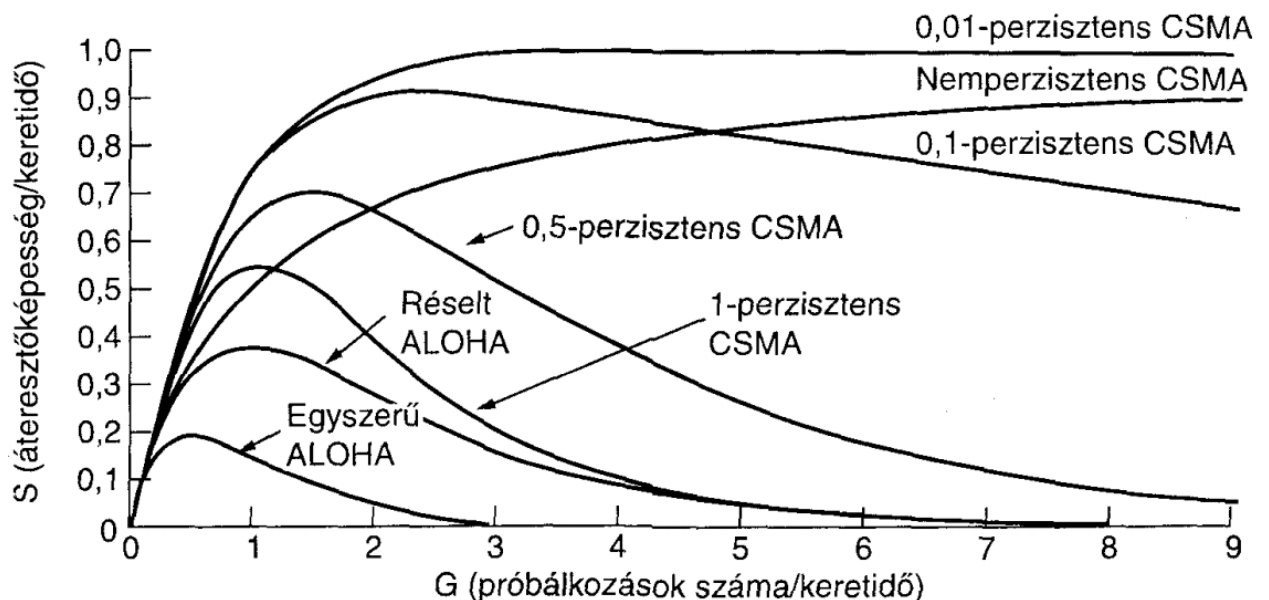
A protokoll elnevezésében az „1” azt jelzi, hogy az adás megkezdésének a valószínűsége „1” (azaz biztos), ha a csatornát szabadnak érzékeli.

- nem perzisztens CSMA

Ez a megoldás az előző protokollnál „türelmesebb” megoldás, mégis – bizonyos esetekben – hatékonyabb a csatorna áteresztő képessége szempontjából, viszont a jeltovábbítás késleltetése szempontjából kevésbé hatékony. Ez esetben az adni szándékozó állomás szintén azzal kezd, hogy megvizsgálja a csatorna foglaltságát. Ha szabad a csatorna, akkor elkezdi az adást. Ha nem szabad a csatorna, akkor „t” ideig vár és újra ellenőriz. Ez a protokoll így nagyobb valószínűséggel kerüli el azt az állapotot, hogy egy keret befejezése után több adó is egyszerre, azaz ütközve kezdjen el adni.

- p-perzisztens CSMA

Ez a protokoll tulajdonképpen az 1-perzisztens CSMA „p”-vel paraméterezett megoldása. Ez esetben az a szabad csatorna észlelése utáni adáskezdés valószínűsége nem „1”, hanem „p”. Nyilvánvaló, hogy a „p” értékét tekintve a $0 < p < 1$ értékek jöhetnek szóba. Minél kisebb a „p”, azaz a perzisztencia értéke, annál jobb a csatornakihasználtság, de egyben annál hosszabb az átlagos késleltetési idő. Például a „p”=0,01 esetében egy állomás szabad csatorna észlelésekor is csak 0,01-es valószínűséggel fog az adásba belekezdeni, azaz az esetek egy részében feleslegesen várakozik.



Mielőtt az ütközésmentes protokollokkal elkezdénénk foglalkozni, elmélkedjünk egy kicsit néhány elemi összefüggésen. A hálózatokon fellépő késleltetések miatt az üresnek érzékelt csatorna közel sem biztos, hogy valóban üres, mivel egy távoli állomás már elkezdhetette az adást, csak még lehet, hogy nem ért el hozzánk. Az még nulla késleltetés esetén is előfordulhat, hogy két állomás pont egyszerre kezd adásba, és a keretek ütköznek.

Az ütközés érzékelése valamint kezelése precíz megoldást igényel. Maga az ütközés néhány bit adat átviteli ideje alatt megállapítható. (Többek közt ez is az oka hogy a fizikai rétegben a 0V feszültség szint a legtöbb esetben nem azonos a „0” bittel, hiszen így – a feszültség hiányában – két ütköző „0” tartalmú bitet lehetetlen lenne érzékelni.) Az ütközés érzékeléséhez a keret időbeli hosszának valamint a hálózat fizikai hosszának és a jel terjedési sebességének megfelelő viszonyban kell lennie egymáshoz. (A jel terjedési sebessége megközelítőleg réz- és üvegekábel esetén is $200\text{m}/\mu\text{s}$.) A jel terjedési ideje két versengő állomás között legyen „T”. Ez esetben a keret hossza „2T”-nél hosszabb kell hogy legyen, különben az ütközés nem érzékelhető biztonsággal.

A fentiek szerint tehát meghatározható, hogy az ütközés biztos érzékeléséhez mekkora lehet a maximális kábelhossz egy adott ütközési szegmensben. Amennyiben a keret mérete túl kicsi, akkor azzal a hálózatunk (szegmensünk) méretét korlátozzuk, az sem lehet így bármilyen nagy. Viszont, ha a keret méretét megnöveljük – amivel a hálózat (szegmens) mérete is megnőhet – akkor pedig a rövid, vagyis a keret méreténél rövidebb üzenetek küldése lerontja az átvitel határfokát.

- CSMA/CD (CSMA Collision Detection / CSMA ütközésérzékeléssel)

A legfontosabb eltérés az eddig tárgyalt CSMA protokollokhoz képest az, ami adás megkezdése után történik. Az előzményekben nincs eltérés, nyilván csak üres csatorna esetén kezdődik meg az adás. Az adás megkezdése után az adó tovább figyeli a csatornát, és amennyiben azt érzékeli, hogy egy másik adó is megkezdte adását, azaz keretütközés történt, azonnal megszakítja adását, hiszen az ütközés miatt az már úgysem érhet pontosan célba. Ilyen módon időt, erőforrást és sávszélességet takarít meg. Az újraküldés előtti véletlen hosszú várakozási idő ez esetben is megmarad, de az összes várakozással töltött idő biztos – legalább a keret eldobott részével – rövidebb lesz.

Az ütközés érzékelése a teljesítmény mérésével oldható meg, azaz ha saját adási teljesítményénél nagyobb teljesítményt érzékel, akkor ütközés történt. Ehhez nyilván arra van szükség, hogy a hálózat (szegmens) két legtávolabbi pontja között is képes legyen a jel olyan kicsi csillapítással és késleltetéssel terjedni, hogy az ütközés detektálható legyen.

Ütközésmentes protokollok

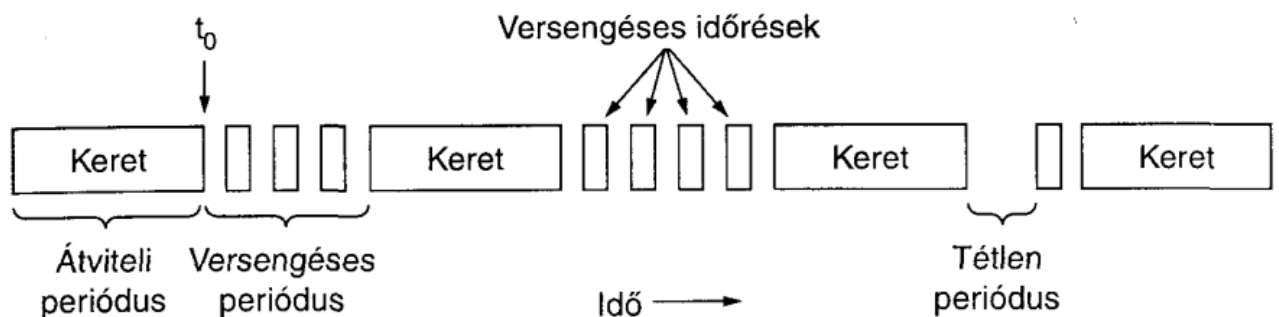
A sávszélesség minél hatékonyabb kihasználása érdekében az eddig megismert újabb és újabb eljárások komoly javulást mutattak elődeikhez képest, de magát az ütközést nem sikerült kiküszöbölniük. A nagytávolságú üvegszál hálózatokon ez eddig ismert protokollok egyike sem hatékony, hiszen a késleltetési idő nagy, ezért túlságosan hosszúra kellene választani a legrövidebb keret méretét. Túl hosszú keret azért nem lesz hatékony, mert akár egyetlen karakter elküldéséhez is egy teljes keretet fogunk a hálózatra tenni. A valós idejű átvitel (például a VoIP) esetében is szükséges követelmény az ütközésmentesség.

A megoldást az ütközésmentes és a korlátozott versenyes protokollok jelentik. Először vegyük sorra az ütközésmentes protokollokat.

- Helyfoglalásos protokoll (Reservation Protocol)

Azokat a protokollokat, melyek a tényleges forgalmazás kezdete előtt adatszórással (Broadcast) jelzik az adási igényüket nevezzük helyfoglalásos protokollnak. A modellben induljunk ki abból, hogy véges számú „N” darab állomás létezik, és mindegyiket lássuk is el 0-tól N-1-ig terjedő címmel.

Az egyik alapvető eljárás bit-térkép (Basic Bit-Map) protokoll. A versengési periódus pontosan „N” darab időrésből áll. Ha az 0-ik (azaz sorban az első) állomás adni szeretne, az 0-ik (azaz sorban az első) időrésbe elhelyez egy „1” értékű bitet. Minden állomás akkor helyezhet bitet a csatornába, amikor a neki megfelelő számú időrés következik. A többi – nem az állomás számára fenntartott – időrés alatt nincs adatforgalmazás. Az állomások a versengési periódusban bejelentett foglalásuk sorrendjében kezdik meg az adást, mivel minden állomás ismeri az adni kívánó állomások azonosítóját, és így az adási sorrendet. Az utolsó keret vége könnyen érzékelhető, és ettől kezdődően újra indul a csatorna-foglalási eljárás.



A protokoll felvet egy hatékonysági kérdést. Amennyiben az adatkeretek száma is korlátozott, azaz kevesebb, mint az időszeletek száma, akkor az alacsonyabb sorszámú állomások nagyobb valószínűséggel tudnak adni, mint a magasabb sorszámúak. Főleg, ha azt is figyelembe vesszük, hogy egy alacsony sorszámú állomás az adás után valószínűleg már adásra kész lesz a következő keretével, amíg a magasabb sorszámú állomások adnak, így a következő foglalási periódusban be fogja jelenteni foglalási igényét. A magasabb sorszámú adó viszont valószínűleg nem lesz kész a következő kerettel az adattovábbítás után a versengési periódus megkezdéséig, így a következő foglalási periódusból valószínűleg kimarad, és csak az azt követően tud újra foglalni és adni. Más szempontból nézve viszont, ha az állomások az új kereteket az átviteli időnél sokkal gyorsabban állítják elő, akkor meg az alacsonyabb sorszámú állomások is kénytelenek több időt tölteni várakozással, hiszen hiába van adásra kész állapotban a keretük, meg kell várniuk, amíg a magasabb sorszámú állomások is befejezik adásaikat. Az állomásszámok permutálásával az adási valószínűségek kiegyenlíthetők. Minden állomás egy versengési időrés idejével növeli az adatátviteltől elvett időt. Alacsony terhelésnél a versengési időrések jelentősen rontják a hatásfokot. Ha egyetlen állomás akar csak adni, akkor is végig kell várunk az összes foglalási időrést, ugyanúgy mintha az összes állomás adni akarna.

- Vezérjeles gyűrű (Token Ring) és Vezérjeles sín (Token Bus)
Legfontosabb jellemző a gyűrű illetve a sín topológia. A topológia egyben egy állomás sorrendet is jelent. A módszer lényege, hogy az állomások topológiai sorrendjüknek megfelelően egy vezérlőjelet (Token) adnak egymásnak tovább. Mindig az az állomás adhat, aki a tokent birtokolja. Amennyiben a tokent birtokló állomásnak éppen nincs adásra szánt kerete, akkor csak a tokent adja tovább; ha van adásra szánt kerete, akkor az adás és a token átadása is megtörténik. Nyilvánvaló, hogy a token a gyűrűben folyamatosan körbejár, a sín esetében hasonlóan, hiszen a sánt logikailag gyűrűként kezeljük. A keret nem tesz meg több kört, hiszen legkésőbb a kör végén meg kell hogy érkezzon a megcímzett állomáshoz. Minden állomás csak a következő állomásig juttatja el a keretet, tehát amit az adó adott, azt a többi állomás – egészen a címzett állomásig – megismétli. Ütközés így ebben az esetben biztosan nem fordulhat elő, az adás prioritása szempontjából pedig minden állomás azonos szintű.

- Bináris visszaszámlálás protokoll (Binary Countdown)

Az állomások számának növekedésével túl sok versengési időrés kerül a sorba, állomásonként az időrés egy bittel nő. Több ezer állomás esetében ez már jelentős idővesztést okoz. Ha az állomások sorszámát, azaz címét viszont binárisan adjuk a csatornába, akkor a versengési idő jelentősen csökkenthető. Pl.: 63 állomás, ami 63 időrést igényelne, leképezhető 6 biten (000001₂-től 111111₂-ig), tehát 6 darab időrés így elegendő; 15 állomás esetén pedig 4 darab. Az, hogy a csupa 0-ból álló címet mért nem használjuk, hamarosan kiderül.

Minden adni kívánó állomás a versengési időrésekben kiteszi saját címét a csatornára addig, amíg a szabályok szerint vissza nem lép, vagy adásba kezd. Azt azonban fontos észrevenni, hogy a (fenti példában szereplő) 6 bit csak egy számot, azaz egyetlen állomást képes azonosítani. Ezért szükség van egy jól felépített döntéshozó szabályrendszerre, ami eldönti, hogy melyik állomás adhat.

Például, ha (a 4 bitnél maradva) a 0010₂, 0100₂, 1001₂ és a 1010₂ számú adó kívánja bejelenteni adási szándékát, a folyamat a következő képpen zajlik. Mind a négy állomás elkezd feltölteni az a legmagasabb helyiértékű, vagyis balról az első bitjét, azaz a fenti sorrendben 0, 0, 1, 1, de természetesen csak egy érték kerülhet be. A bekerülő érték az összes beküldésre szánt érték VAGY logikai kapcsolatából származó érték, vagyis az 1-es, amit ne felejtsük el, ketten is küldtek. Aki 0-át küldött, az kiesett, mert volt 1-es küldés is. A második lépésben a maradék két állomás (sorban a harmadik és a negyedik) egyaránt 0-át küld, azaz a versenyben lévők közül senki sem küld 1-est, így mindketten továbbra is versenyben maradnak. A harmadik lépésben ugyanez a két állomás küldi következő bitjét, ami a harmadik állomás esetében 0, a negyedik állomás esetében 1, tehát a negyedik állomás kezdheti meg az adást.

A protokoll nyilvánvalóan a magasabb számú állomásoknak kedvez. Azért, hogy az ne fordulhasson elő, hogy egy alacsony sorszámú állomás csak nagyon sokára, vagy esetleg soha ne kerüljön sorra, vezették be az úgynevezett virtuális állomásszámokat. Ez azt jelenti, hogy az állomások száma minden sikeres átvitelt követően permutálódik.

A protokoll a csatorna 100%-os kihasználását is lehetővé tudja tenni a keretek optimális felépítése esetén. Amennyiben a keret eleve a küldő címével kezdődik, akkor egyetlen bit sem ment veszendőbe, mert mindig van egy olyan keret, ami adásba kerülhet.

Korlátozott versenyes protokollok

A versenyhelyzetes protokollok jól teljesítenek kis terhelésnél, de nagy terhelésnél nagyon romlik a hatásfokuk. Az ütközésmentes protokollok kis terhelésnél is képesek jelentős késleltetéseket produkálni, de a terhelés növekedésével javul a hatásfokuk. Kívánatos lenne a kedvező tulajdonságok ötvözése. Ezt a célt valósítják meg a korlátozott versenyes protokollok.

Az alapötlet az, hogy az állomásokat csoportokba rendezzük, és hozzárendeljük egy-egy időréshez. Egy időrásben csak az adott csoportba tartozó állomások versengenek az időszelért. Elképzelhető, hogy egy időréshez tartozó csoportban egyetlen állomás van. A másik szélső esetben az összes állomást egy csoportba soroljuk (ez gyakorlatilag a TDM illetve a réselt ALOHA protokoll).

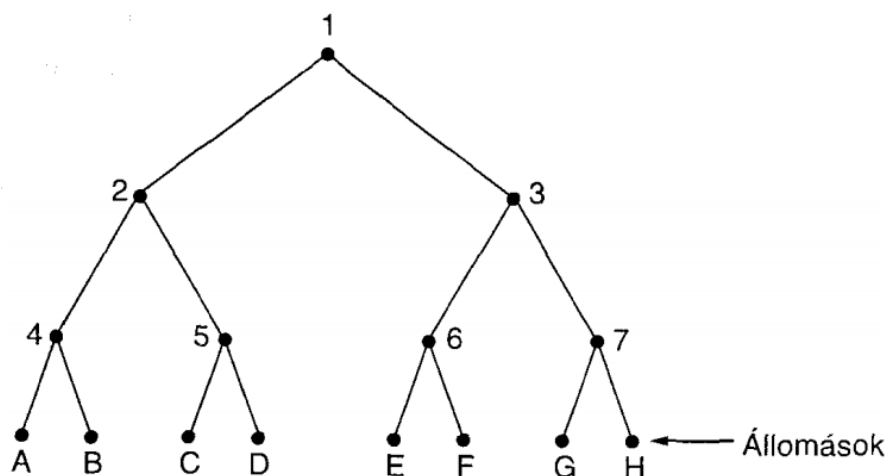
- Adaptív fabejárési protokoll

Érdekes történelmi előzménye van a protokoll modelljének. A II. világháború alatt az amerikai hadseregben a következő módszerrel szűrték ki a szifilisz katonákat (Robert Dorfman, 1943). Egy adott csoportból („n” darab katona) vért vettek, majd a vett vérminták egy kis részét egy gyűjtőbe öntötték. Első lépésként ezt a kevert mintát vizsgálták meg. Ha nem találtak benne fertőzést, akkor az összes katona egészséges volt. Ha fertőzést találtak, akkor az „n” létszámú csoportot 2 darab „n/2” csoportra osztották. Az így kialakított csoportokban újra gyűjtött mintát képeztek, és a mintákat megvizsgálták. Amelyik mintában fertőzést találtak azt az „n/2”-es csoportot hasonló módon tovább bontották. Így végül el lehetett jutni a fertőzött katoná(k)hoz.

A fenti eljárás számítógépre adaptált modellje a bináris fa struktúra, illetve az ebből levezetett adaptív fabejárési protokoll, melyet egymástól függetlenül dolgozott ki Boris Solomonovich. Tsybakov és Viktor Alexandrovich Mikhailov (1978) illetve John I. Cepetakis (1979) – a nyugati források jellemzően csak az utóbbi nevet említik...

A protokoll felépítése és működése a következő. Az állomásokat egy bináris fa levelei szimbolizálják.

- A 0. (azaz fizikailag az első) időrésben minden állomás indíthat küldést. Amennyiben csak egy állomás kezd el adni, akkor minden rendben van, de ha több állomás kezd el adni, akkor ütközés történik, és megkezdődik a fa mélységi bejárása, valamint egyetlen állomás sem fogad újabb adatcsomagot a hálózati rétegtől.
- A következő, az 1. (fizikailag második) időrésben csak a sorszám szerint következő, vagyis csak a 2-es csomópont alatti részfa állomásai vesznek részt. Ezt követően az elágaztatás algoritmus a következő lesz.
- Amennyiben a 2-es csomópontban nincs ütközés, elküldésre kerül a keret, és a következő, a 2. (fizikailag harmadik) időszelét a 3-as csomópontból kezdi a vizsgálatot.
- Amennyiben a 2-es csomópontban is még ütközés van, akkor a következő, a 2. (fizikailag harmadik) időszelét a 4-es csomópontból kezdi a vizsgálatot.
- Amennyiben a 4-es csomópontban nincs ütközés, elküldésre kerül a keret, és a következő, a 3. (fizikailag negyedik) időszelét az 5-ös csomópontból kezdi a vizsgálatot.
- Amennyiben az 5-ös csomópontban is még ütközés van, akkor „C” és „D” állomás is adni akar. Ez esetben a következő, a 4. (fizikailag ötödik) időszelétben a „C” adhat, majd a következő, az 5. (fizikailag hatodik) időszelétben a „D” adhat.
- A folyamat a következő, a 6. (fizikailag hetedik) időszelétben a 3-as csomópontban folytatódik a fentiek szerint.



Amennyiben a rendszer terhelése nagy, célszerű az ütközés kereső algoritmust nem a struktúra csúcsán, hanem (nagyobb struktúrák esetén) jópár szinttel lejjebb elkezdni. Matematikailag megválaszolható a „Na de melyik szinten kezdjük?” kérdés is.

Felülről lefelé számozva az ábrán szereplő fa struktúra esetében az 1. csomópont a 0. szint, a 2-es és 3-as az 1. szint, a 4-es, 5-ös, 6-os, 7-es pedig a 2. szint. Könnyen belátható, hogy az „x”-edik szint minden csomópontjához az alattuk lévő állomások 2^{-x} része tartozik. (Azaz az 1-es szint csomópontjai – a 2-es és 3-as – alatt az összes csomópont 2^{-1} része, azaz 0,5-e, azaz a fele tartozik. A 2-es szint csomópontjai – a 4-es, 5-ös, 6-os, 7-es – alatt az összes csomópont 2^{-2} része, azaz 0,25-e, azaz negyede tartozik.) Ha „y” darab adásra kész állomás – valószínűsíthetően egyenletesen elosztva – található a fában, akkor azon a szinten optimális a keresést elkezdni, ahol a versengő állomások száma 1, tehát $y \cdot 2^{-x} = 1$. Ha ezt tovább rendezzük „x”-re, akkor az „ $x = \log_2 y$ ” összefüggést kapjuk.