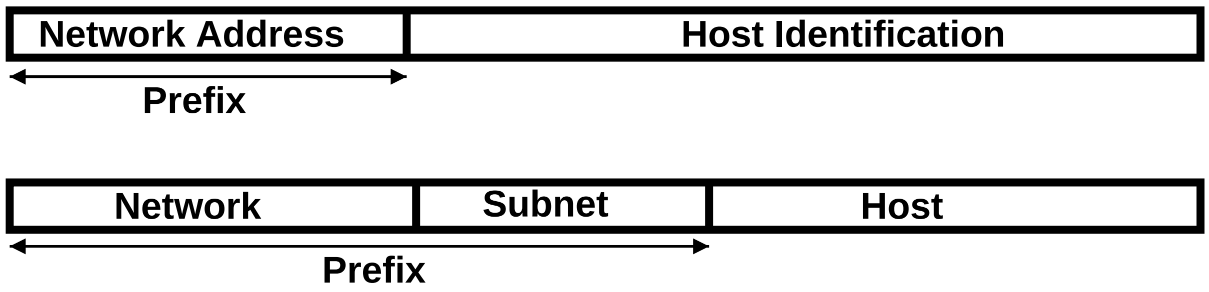
Hálózatok II
2006

6: Az IP Prefix Lookup Probléma IV.

Internet Protocol IP

- Az adatok a küldőtől a cél állomásig IP-csomagokban kerülnek átvitelre
- A csomagok fejléce tartalmazza a cél IP-címét



- Minden csomópont (router) a cél címe alapján dönt, hogy melyik szomszédos csomópontnak kell továbbítania a csomagot
  - Az ehhez szükséges információt minden csomópontban egy forwarding-tábla tárolja

IP Prefix Lookup Probléma

- A címek *W* hosszúságú bináris sztringek.

Egy router forwarding-táblája *R*

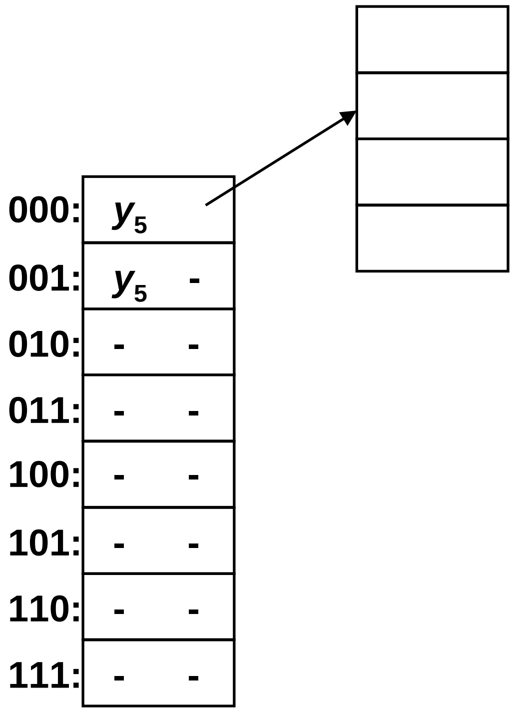
- bejegyzéseket tartalmaz (*x,y*) formában
  - *x*: legfeljebb *W* hosszú bináris sztring, melynek a neve prefix. *x* lehet az üres sztring is, amit ε-nal jelölünk,
  - *y*: egy a router-ből kivezető link.
- Minden *x* bináris sztringhez *R* legfeljebb egy (*x,y*) bejegyzést tartalmaz.
- A bejegyzések száma a forwarding-táblában *N*.

Feladat:

- A routerhez érkező csomaghoz, melynek cél címe *k*,
- találjuk meg az (*x,y*) bejegyzést *R*-ben a leghosszabb egyező prefixszel (BMP).

Multibit-Trie és prefix-expanzió

- A BMP keresése Trie adatstruktúrával $O(W)$ időt igényel (worst-case).
- Egy lehetőség a keresési idő megjavítására:
  - Minden Trie-csomópontban a célcímnek egyszerre több bitetjét kezelni
    - Ha pl. egy Trie-csomópontban 3 bitet egyszerre kezelünk, akkor a Trie-csomópont tartalmaz egy 2^3 elemű arrayt, melyet a kezelt 3 bit indexel
    - Az array elemei tárolnak
      - egy mutatót egy masik Trie-csomópontra (lehet „null”) és
      - egy linket, ha a feldolgozott bitek a egy R beli bejegyzés prefixét határozzák meg.



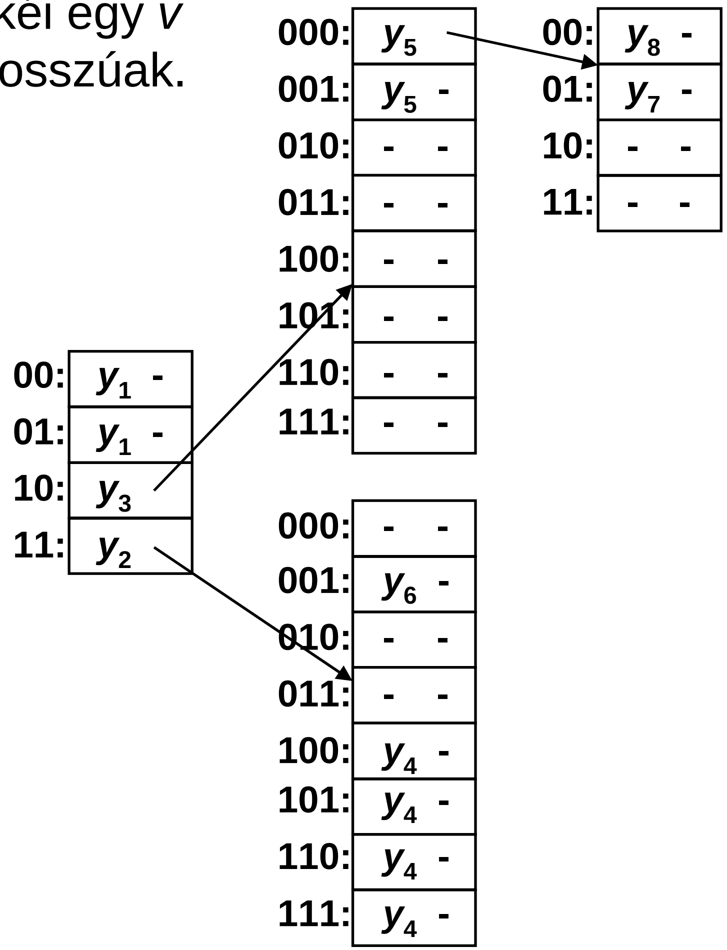
Multibit-Trie és prefix-expanzió

- Egy Multibit Trie egy olyan Trie, amiben az élek cimkéi egy v csomópontból a gyermekeihez mind egyformán k_v hosszúak.
  - A cimkék tárolása helyett egy 2^{k_v} elemű arrayt használunk.
  - Az array i. elemében, $0 \le i < 2^{k_v}$, az áll, hogy
    - létezik-e egy él a Trie-ban i cimkével, és ha igen, hová vezet;
    - ahhoz a prefixhez, ami a gyökérből a v-hez vezető úton lévő élek cimkéinek és az i sztring konkatenációjaként áll elő, tárolunk-e egy linket.
- Példa:

  L_2 = { (00,y_1), (01,y_1), (10,y_3), (11,y_2) }

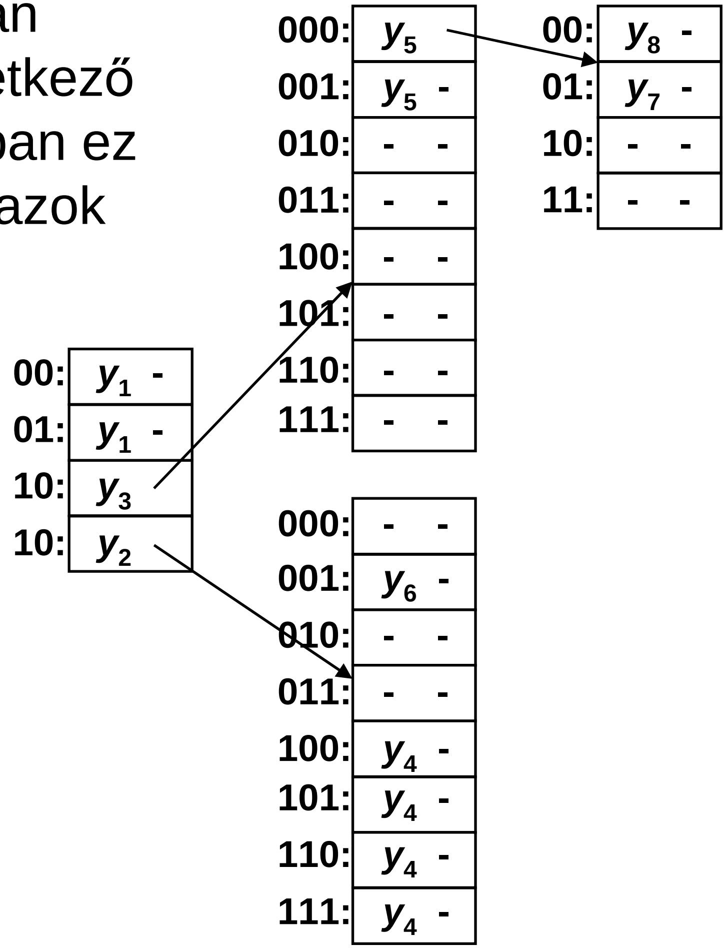
  L_5 = { (11100,y_4), (11101,y_4), (11110,y_4), (11111,y_4), (10000,y_5), (10001,y_5), (11001,y_6) }

  L_7 = { (1000000,y_8), (1000001,y_7) }



Multibit-Trie és prefix-expanzió

- A bitek száma b_v, amit egy v Trie-csomópontban kezelünk, egy nemüres hosszosztály és a következő nemüres osztály hosszkülönbsége. A példánkban ez a gyökérben 2, annak gyermekeinél 5-2=3, és azok gyermekeinél 7-5=2.
- Egy k célcím keresése:
  A w gyökérben tekintjük k első b_w bitjét $k[1..b_w]$-t. A keresést annál a gyermeknél folytatjuk, amelyre w-ben az array $k[1..b_w]$-edik eleme mutat, ha ilyen mutató létezik. Amíg ilyen mutató létezik, követjük a megfelelő utat. Az utolsó link y, amit ezen az úton találtunk, az tartozik a BMP-hez.



Multibit-Trie és prefix-expanzió

- A worst-case keresési idő egy *T* Multibit-Trie-ban lineáris *T* mélységéhez, azaz $O(s)$, ahol s a nem üres hosszosztályok száma.

- Egy bejegyzés befűzése vagy törlése hatákonyan elvégezhető.

A tárigény minimalizálása a Multibit-Trie-ban

- A Trie minden *v* csomópontjának tárigénye 2^{b_v} (az array elemeinek száma *v*-ben).
- A *T* Trie teljes tárigénye $\Sigma_{v\in T}\, 2^{b_v}$.

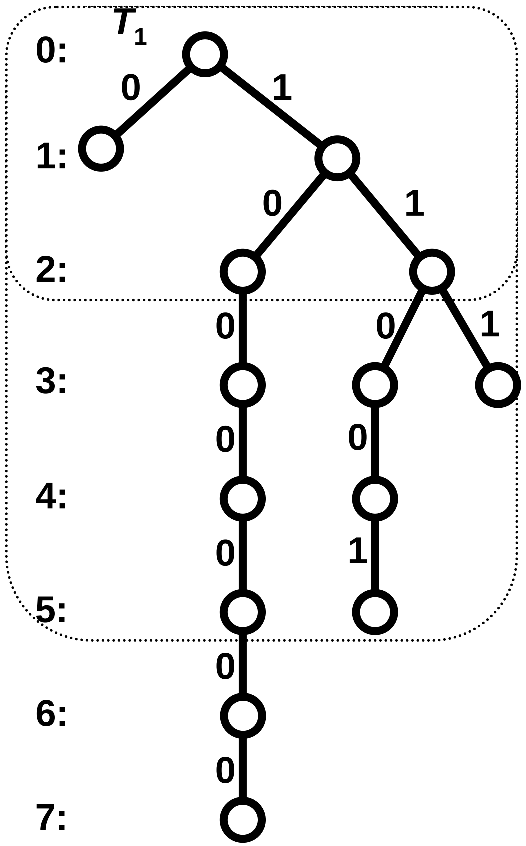
Optimalizálási feladat: Hogyan kell egy megadott s esetén az s hosszosztályt kiválasztani, hogy a Multibit-Trie tárigénye minimális legyen?

Feltétel: A hosszosztályok és a Trie rétegei között egy-az-egyhez megfelelés van. (Multibit-Trie esetén enélkül a feltétel nélkül is optimalizálhatunk. [Srinivasan, Varghese 99])

- A Trie egy rétege azon Trie-csomópontok halmazát jelenti, melyek a gyökértől ugyanolyan távolságra vannak. Az 1. réteg a gyökérből áll. A 2. réteg a gyökér gyermekeiből, és így tovább…
- Az *i.* réteg megfelel az L_i hosszosztálynak.
- Következésképpen egy réteg minden csomópontjánál ugyanannyi bitet tekintünk.
- Ezért egy réteg minden csomópontja ugyanakkora tárigényű.

A tárigény minimalizálása a Multibit-Trie-ban

- Jelölje $A[j,l]$, $0 \le j < l \le m$, a nemüres L_l hosszosztály minimális tárigényét, ha a tőle rövidebb leghosszabb nemüres hosszosztály L_j, $j < l$.
- $A[j,l]$, $0 \le j < l \le m$ kiszámítása:
  - Felépítünk egy 1-Bit-Trie-t T_1-t az eredeti prefixekhez úttömörítés nélkül.
  - Legyen $n(i)$ azon csomópontok száma T_1-ben, amelyek távolsága a gyökértől i és legalább egy gyermekük van.
  - Teljesül: $A[j,l] = 2^{l-j}\, n(j)$.
    A Multibit-Trie L_j -nek megfelelő rétegének minden csomópontjában az array minden eleme, amely T_1-ben egy olyan csomópontnak felel meg, aminek legalább egy gyermeke van, L_l-nek megfelelő réteg csomópontjára mutat. Tehát az L_l -nek megfelelő rétegnek pontosan $n(j)$ csomópontot kell tartalmaznia, amelyek mindegyike 2^{l-j} elemet tartalmaz.
    Minden $0 \le j < l \le m$ –hez $A[j,l]$ kiszámítható T_1 segítségével $O(m^2) = O(W^2)$ idő alatt.



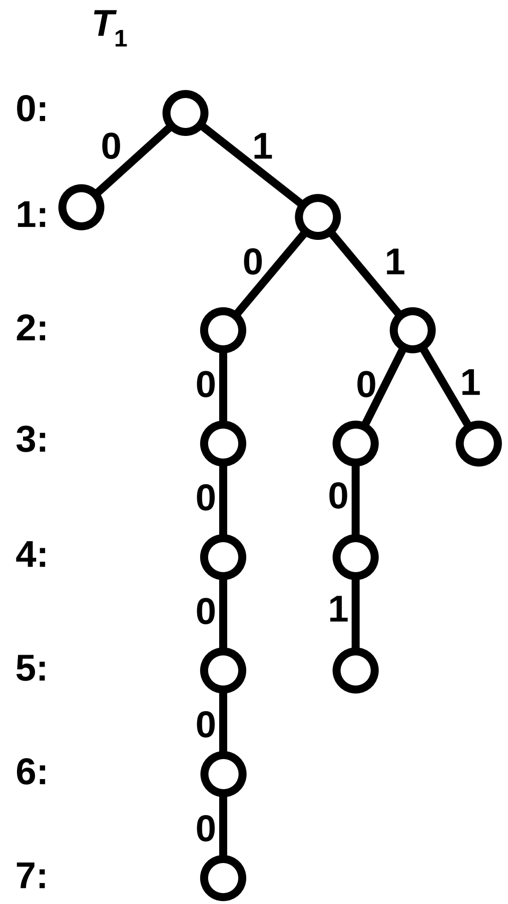
A tárigény minimalizálása a Multibit-Trie-ban

Példa:

- $L_1 = \{(0,y_1), (1,y_2)\}$
- $L_2 = \{(10,y_3)\}$
- $L_3 = \{(111,y_4)\}$
- $L_4 = \{(1000,y_5)\}$
- $L_5 = \{(11001),y_6\}$
- $L_6 = \{(100000),y_7\}$
- $L_7 = \{(1000000),y_8\}$

- $A[0,2] = 2^{2-0}\, n(0) = 4 \cdot 1 = 4$,
- $A[2,5] = 2^{5-2}\, n(2) = 8 \cdot 2 = 16$,
- $A[5,7] = 2^{7-5}\, n(5) = 4 \cdot 1 = 4$.

- Ez megadja egy 3 szintű Trie társzükségletét L_2, L_5 és L_7. hosszosztályokkal.



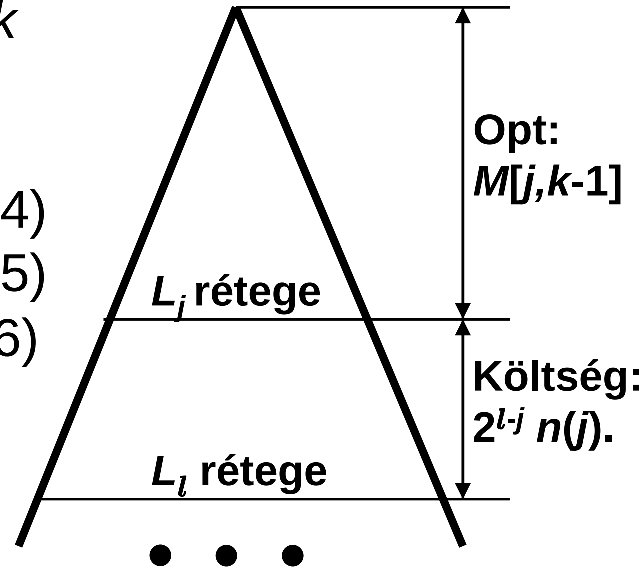
A tárigény minimalizálása a Multibit-Trie-ban

- Tekintsük azt a Multibit-Triet, ami akkor keletkezik, ha az eredeti $L_1,\ldots,L_l$ hosszosztályokat úgy alakítjuk át prefix expanzióval, hogy
  - legfeljebb k nem üres hosszosztály marad és
  - L_l a legnagyobb hosszt tartalmazó hosszosztály.
- Legyen $M[l,k]$ a minimális tárigénye a Trie rétegeinek a gyökértől addig a rétegig, ami L_l –nek felel meg, ha k hosszosztályt engedünk meg. Ekkor érvényes:

$$M[l,1] = A[0,l] \qquad \forall\, l \geq 1 \qquad (4)$$

$$M[0,k] = 0 \qquad \forall\, k \geq 1 \qquad (5)$$

$$M[l,k] = \min_{0\leq j<l}\{\, M[j,k-1] + A[j,l] \,\} \quad \text{ha } l > 0,\ k > 1 \qquad (6)$$



A tárigény minimalizálása a Multibit-Trie-ban

(4) ($M[l,1] = A[0,l]$, $\forall\ l \geq 1$) igaz, mert a gyökér társzükséglete pontosan $A[0,l] = 2^l n(0) = 2^l$, ha L_l az a nemüres hosszosztály, ami a legrövidebb sztringeket tartalmazza.

(5) ($M[0,k] = 0$, $\forall\ k \geq 1$) csak annyit mond, hogy egy üres Multibit-Trie nem igényel tárt.

(6) ($M[l,k] = \min_{0 \leq j < l}\{ M[j,k-1] + A[j,l] \}$ für $l > 0$, $k > 1$) igaz, mert az L_l-t megelőző leghosszabb hosszosztályra, L_j-re minden lehetőséget megvizsgálunk és ekkor a tárigény a következő két érték összege:

- Az optimális tárigény a Multibit-Trie rétegekhez, amelyek a hosszosztályoknak L_j-ig felelnek meg és legfeljebb k-1 nem üres hosszosztályt tartalmaznak (ezt $M[j,k-1]$ írja le) és
- Az L_l hosszosztály rétegének a tárigénye (ezt $A[j,l]$ írja le).

A tárigény minimalizálása a Multibit-Trie-ban

- Dinamikus programozással hatákonyan ki tudjuk számítani $M[\ell,k]$ minden értékét.
- $M[m,s]$ megadja a minimális tárigényt a Multibit-Trie-hoz, ami s réteget tartalmaz.
- Legyen $P[\ell,k]$ az a j érték, amire $M[\ell,k] = \min_{0\leq j<\ell}\{ M[j,k-1] + A[j,\ell] \}$ a minimumot veszi fel. A számítás végén

  $$m,\ P[m,s],\ P[P[m,s],s\text{-}1],\ \ldots$$

  adja meg a prefix-hosszakat az s hosszosztályban.

A tárigény minimalizálása a Multibit-Trie-ban

<u>Algoritmus</u> Optimalis_Multibit_Trie

<u>Input</u>: $A[j,\ell]$, $\forall\ 0 \le j < \ell \le m$; A hosszosztályok száma s

<u>Output</u>: $M[\ell,k]$ és $P[\ell,k]$, $\forall\ 0 \le \ell \le m$, $1 \le k \le s$

1. <u>for</u> $\ell = 0$ <u>to</u> m <u>do</u>
2.   <u>for</u> $k = 1$ <u>to</u> s <u>do</u>
3.   <u>if</u> $\ell = 0$ <u>then</u> $M[\ell,k] := 0$; $P[\ell,k] := 0$;
4.   <u>else</u> <u>if</u> $k = 1$ <u>then</u> $M[\ell,k] := A[0,\ell]$; $P[\ell,k] := 0$;
5.   else
6. $M[\ell,k] := \min_{0 \le j < \ell}\{ M[j,k-1] + A[j,\ell] \}$;
7. $P[\ell,k] :=$ az a j, amire az előző sor a minimumot adja;
8.   <u>fi</u>;
9.   <u>od;</u>
10. <u>od;</u>

Az optimális Multibit-Trie kiszámításának időanalízise

- Az 1-Bit-Trie felépítése és az $n(i)$ und $A[j,\ell]$ értékek kiszámítása $O(N \cdot W + W^2)$ időt igényel.
- $M[\ell,k]$ értékeinek kiszámítása az Optimalis-Multibit-Trie algoritmussal $O(s \cdot W^2)$ időt igényel.
- Összesen: $O(N \cdot W + s \cdot W^2)$ idő.

<u>Tétel 2</u>: Legyenek egy forwarding-tábla bejegyzései és a kívánt nem üres hosszosztályok száma s adva. Az s nemüres hosszosztály kiválasztása, amely a forwarding-táblához egy s rétegű Multibit-Trie tárigényét minimalizálja, $O(N \cdot W + s \cdot W^2)$ idő alatt kiszámítható. □

Irodalom

- V. Srinivasan, G. Varghese: **Fast Address Lookups using Controlled Prefix Expansion**. *ACM Transactions on Computer Systems*, Vol. 17(1), 1-40, 1999.