Memory size bounds of prefix DAGs

János Tapolcai, Gábor Rétvári, Attila Kőrösi

MTA-BME Future Internet Research Group, High-Speed Networks Laboratory (HSNLab),
Dept. of Telecommunications and Media Informatics, Budapest University of Technology,
Email: \{tapolcai, retvari, korosi\}@tmit.bme.hu

Abstract—In this report an entropy bound on the memory size is given for a compression method of leaf-labeled trees. The compression converts the tree into a Directed Acyclic Graph (DAG) by merging isomorphic subtrees.

I. COUPON COLLECTOR’S PROBLEM WITH ARBITRARY COUPON PROBABILITIES

Given a set of $C$ coupons, where $\delta = |C|$ denotes the number of coupons. At each draw $p_o$ denotes the probability for getting coupon $o$ for $o \in C$. We draw $m$ coupons, and let $E$ denote the expected number of different coupons we have obtained. The task is to give an upper bound on $E$.

Let $V$ denote the set of coupons we have after $m$ draw. The probability of having coupon $o$ in $V$ is

$$P(o \in V) = 1 - (1 - p_o)^m$$

(1)

Thus the expected cardinality of $V$ is

$$E(|V|) = \sum_{o \in C} E(I(o \in V)) = \sum_{o \in C} P(o \in V) = \sum_{o \in C} (1 - (1 - p_o)^m)$$

(2)

Let $H_C$ denote the entropy of the coupon distribution

$$H_C = \sum_{o \in C} p_o \log_2 \frac{1}{p_o}$$

(3)

Lemma 1.

$$E(V) \leq \min \left\{ \frac{m}{\log_2(m)} \cdot H_C + 3, m, n \right\}$$

for $m \geq 3$.

Proof: Trivially holds that $E \leq m$ and $E \leq n$. Next, let us expand

$$\sum_{o \in C} (1 - (1 - p_o)^m) \leq \frac{m}{\log_2(m)} \sum_{o \in C} p_o \log_2 \frac{1}{p_o} + 3$$

(4)

The above inequality holds if the inequality holds for each $o \in C$. Thus next we prove that

$$1 - (1 - p_o)^m \leq \frac{m}{\log_2(m)} p_o \log_2 \frac{1}{p_o}$$

(5)

holds for $p_o \leq \frac{1}{e}$. Let us assume $m \geq \frac{1}{e}$. Note that the right hand side is a monotone increasing function of $m$, when $m \geq e$. Thus we can substitute $m = \frac{1}{e}$ if $\frac{1}{p} > e$ in the right hand side and we get

$$1 - (1 - p_o)^m \leq \frac{1/p}{\log_2(1/p_o)} p_o \log_2 \frac{1}{p_o} = 1.$$  (6)

In the rest of the proof we focus on the other case, which is $m < \frac{1}{p_o}$. Let us define $1 > x > 0$ as

$$x := \log_{p_o} m.$$  (7)

After substituting $m = \frac{1}{x}$ we have

$$1 - (1 - p_o)^{\frac{1}{x}} \leq \frac{1}{x \log_2 (\frac{1}{p_o})} p_o \log_2 \left( \frac{1}{p_o} \right) = \frac{1}{x} p_o = \frac{1}{x p_o - 1},$$

(8)

which can be reordered as

$$(1 - p_o)^{\frac{1}{x}} \geq 1 - \frac{1}{x p_o - 1}.$$  (9)

Taking the $p_o^{x-1} > 0$ exponent of both sides we get

$$(1 - p_o)^{\frac{1}{x}} \geq \left( 1 - \frac{1}{x p_o - 1} \right)^{p_o^{x-1}}.$$  (10)

Note that $x < 1$, thus $\frac{1}{x} > 1$, and we can prove

$$(1 - p_o)^{\frac{1}{x}} \geq \left( 1 - \frac{1}{p_o^{x-1}} \right)^{p_o^{x-1}}.$$  (11)

Bernoulli discovered that $(1 - p_o)^{x}$ is monotone decreasing function and equals to $\frac{1}{e}$ for $p_o \rightarrow 0$. Thus the inequality holds if

$$p_o \leq \frac{1}{p_0 - 1}.$$  (12)

which holds because

$$p_0 \leq 1.$$  (13)

This proves (5) with the assumption of $p_o < \frac{1}{e}$. There are at most $3 > \frac{1}{e}$ coupons for which (5) cannot be applied, but the expected number of these coupons is still at most 3. 

II. TRIE-FOLDING

For IP address lookup a binary trie is used, where each leaf has a label called next hop. To compress the trie we will use trie-folding, which merges the sub-tries with exactly the same structure and next hops labels at each leaf instead of repeating it in the binary trie. After the process the trie is transformed into a DAG. See an example below.
We evaluate the efficiency of the trie-folding methods on a randomly generated trie, where the next hops follow a given distribution. The randomly generated trie is denoted by $T = (V_T, E_T)$ and has the following properties:

- $h$ is the height bound of the trie, typically 24 in IPv4.
- $\delta$ is the set of next hops.
- $p_i$ is the probability that an IP address is forwarded to next hop $i \in \delta$.

Let $V^j_T$ denote the set of nodes in $T$ at the $j$-th level for $1 \leq j \leq h$. The level of a node is $h$ minus the hop count of the path to the root node. Thus the root node has level $h$. At the $j$-th level there are $2^{h-j}$ nodes, formally $|V^j_T| = 2^{h-j}$. Each node at the $j$-th level has $2^{j+1}$ child nodes, and eventually $2^j$ leaves each of which is assigned with a next hop.

The DAG resulted by the trie-folding method is denoted by $D = (V_D, E_D)$, and $V^j_D$ denotes the set of nodes in $D$ at the $j$-th level for $1 \leq j \leq h$.

**Lemma 2.** The expected number of nodes at the $j$-th level in a DAG resulted by trie-folding of a randomly generated trie with height $h$ and next hop distribution $p_1, \ldots, p_N$ is at most

$$E(|V^j_D|) \leq \min \left\{ \frac{H_O}{h-j} 2^h + 3, 2^{h-j}, \delta^j \right\} \quad (14)$$

where $H_O$ denotes the entropy of the next hops

$$H_O = \sum_{o \in \delta} p_o \log_2 \frac{1}{p_o} \quad (15)$$

Proof: We treat the problem as a coupon collection problem, where each coupon is a subtree with $j$ height and $2^j$ next hops on leaves. In other words each coupon is a string with length $2^j$ on alphabet $\delta$, and we draw $m = 2^{h-j}$ coupons. Note that there are $C = \delta^{2^j}$ different coupons. Lemma 1 gives an upper bound on the number of different coupons, which are the subtrees in this case. Thus we have $|V^j_D| \leq 2^{h-j}$, $|V^j_D| \leq \delta^j$ and

$$E(|V^j_D|) \leq \frac{2^{h-j}}{\log_2 (2^{h-j})} H_C + 3 = \frac{2^{h-j}}{\log_2 (2^{h-j})} H_O 2^j + 3 = \frac{2^h}{h-j} H_O + 3 \quad (16)$$

where $H_C = H_O 2^j$ is the entropy of a $2^j$ long string made of next hops. We need to find a reference or add a lemma proving it.

Let $k^*$ be the row where the bounds take the maximum value for all $j = 1, \ldots, h$. See also Figure 1 as an illustration of the bounds on the width of the DAG given by the above lemma. Such $k^*$ clearly exists, because the bounds by Lemma 2 are decreasing function of $j$ until $2^h - j$ holds, while both $\frac{H_O}{h-k} 2^h + 3$ and $\delta^j$ are monotone increasing functions of $j$.

We store each pointer for a node in $h-k^*$ bits. Since each node has two child nodes, it can be stored in $2h - 2k^*$ bits. At level $k^*$ the bound is

$$E(|V^j_D|) \leq E(|V^{k^*}_D|) \leq \frac{H_O}{h-k^*} 2^h + 3 \leq 2^{h-k^*} \quad j = 1, \ldots, h \quad (17)$$

As each node is stored in $2h - 2k^*$ bits we have the following corollary on the width of the DAG.

**Corollary 1.** The expected number of bits to store the nodes at any level $j = 1, \ldots, h$ in the DAG resulted by trie-folding of a randomly generated trie with height $h$ and next hop distribution $p_1, \ldots, p_N$ is at most

$$M = 2H_O 2^h + 6h.$$ 

Based on this we have the following theorem on the size of the DAG.

**Theorem 1.** The expected number of bits to store the nodes in the DAG resulted by trie-folding of a randomly generated trie with height $h$ and next hop distribution $p_1, \ldots, p_N$ is at most

$$2H_O 2^h + 6h^2.$$ 

The lower bound above theorem can be further improved if $H_O \geq \frac{h}{2}$. Let $k$ be the smallest level where $\frac{H_O}{h-k} 2^h + 3$ is larger than $2^{h-k}$. Note that, $k^* < k$. The value of $k$ is

$$k > \lceil \log_2 \left( \frac{h}{H_O} \right) \rceil \quad (18)$$

because

$$2^{h-k} < 2^{h-\lceil \log_2 (\frac{h}{H_O}) \rceil} \leq 2^{h-\log_2 (\frac{h}{H_O})} = 2^h \frac{H_O}{h} \frac{h}{h-k} 2^h + 3 \quad (19)$$

Note that, $\log_2 \left( \frac{h}{H_O} \right) \leq h$ when $H_O \geq \frac{h}{2}$. 

![Diagram of DAG](image_url)
To count the total space needed to store the DAG we divide it into two parts (see also Figure 1).

head for levels $h, \ldots, k$,
body for levels $k-1, \ldots, 1$.

First we estimate the size of head and use the bound $2^{h-j}$ from [14]. The expected number of bits needed for the DAG at level $j = k, \ldots, h$ is

$$
\sum_{j=k}^{h} E(|V_{D_j}|) \leq \sum_{j=k}^{h} 2^{h-j} = \sum_{j=0}^{h-k} 2^{i} = 2^{h-k+1} < 2\frac{H_O}{h-k} 2^h + 6
$$

where the last inequality comes from (19). After multiplying with $2h - 2k^*$ bits for each node we have

$$
(2h - 2k^*) 2^{\left(\frac{H_O}{h-k} 2^h + 6\right)} < (4(h-k) - 2) 2^h + 12h = 4H_O 2^h + 12h = 2M
$$

(20)

For the size of body we use Corollary 1

$$
\sum_{j=1}^{k-1} (2h - 2k - 2) E(|V_{D_j}|) \leq (k-1)M = \left[ \log_2 \left( \frac{h}{H_O} \right) \right] - 1 \leq \log_2 \left( \frac{h}{H_O} \right) - 1
$$

(21)

Finally, summing up with (20) we get the following bound.

**Theorem 2.** The expected number of bits to store the nodes in the DAG resulted by trie-folding of a randomly generated trie with height $h$ and next hop distribution $p_1, \ldots, p_N$ is at most

$$(2 + \log_2(h) - \log_2 H_O) (2H_O 2^h + 6h),$$

when $H_O \geq \frac{h}{2^h}$.

Finally we further improve the lower bound above theorem when $\delta$ is a finite number. Let $l$ be the largest level where $\delta^{2^j}$ is smaller than $\frac{H_O}{h-l} 2^h + 3$. The value of $l$ is

$$l < \left[ \log_2 \left( \frac{h - \log_2 \left( \frac{h}{H_O} \right)}{\log_2(\delta)} \right) \right]$$

(22)

because

$$
\delta^{2^l} < \delta^{2^{l+1}} \leq \delta^{2^j} = \frac{h - \log_2 \left( \frac{h}{H_O} \right)}{\log_2(\delta)}
$$

(23)

Note that $k \geq l+1$, because of the floor an ceiling function and

$$
\log_2 \left( \frac{h}{H_O} \right) > \log_2 \left( \frac{h - \log_2 \left( \frac{h}{H_O} \right)}{\log_2(\delta)} \right)
$$

and taking both side on power 2 we have

$$
\frac{h}{H_O} > \frac{h - \log_2 \left( \frac{h}{H_O} \right)}{\log_2(\delta)}
$$

Note that $H_O \leq \log_2(\delta)$, thus

$$
h > h - \log_2 \left( \frac{h}{H_O} \right),
$$

which always holds.

To count the total space needed to store the DAG we divide it into three parts (see also Figure 1)

head for levels $h, \ldots, k$,
body for levels $k-1, \ldots, l+1$,
tail for levels $l, \ldots, 1$.

To estimate the size of head we use (20). For the size of the tail we use the bound $\delta^{2^j}$ from (13). We have

$$
\sum_{j=1}^{l} E(|V_{D_j}|) \leq \sum_{j=1}^{l} \delta^{2^j} < \sum_{i=1}^{2^l} \delta^i = N \sum_{i=0}^{2^l-1} \delta^i = \frac{N \delta^l - 1}{\delta - 1} < \frac{\delta^2 - 1}{\delta - 1} = \left( \frac{H_O}{h-l} 2^h + 3 \right)
$$

(24)

where the last inequality comes from (23). After multiplying with $2h - 2k^*$ bits for each node we have

$$
(2h - 2k^*) \delta \frac{H_O}{h-l} 2^h + 3 < \left( \frac{\delta^2 - 1}{\delta - 1} \right) \left( \frac{H_O}{h-l} 2^h + 3 \right) = \frac{\delta}{\delta - 1} M
$$

(25)

For the size of body we use Corollary 1

$$
\sum_{j=l+1}^{k-1} (2h - 2k - 2) E(|V_{D_j}|) \leq (k-1-l)M = \left( \log_2 \left( \frac{h}{H_O} \right) - \log_2 \left( \frac{h - \log_2 \left( \frac{h}{H_O} \right)}{\log_2(\delta)} \right) - 1 \right) M = \left( \log_2 \left( \frac{h}{H_O} \right) - \log_2 \left( \frac{h - \log_2 \left( \frac{h}{H_O} \right)}{\log_2(\delta)} \right) - 1 \right) M
$$

(26)

Finally, summing up with (25) and (20) we get the following bound.

**Theorem 3.** The expected number of bits to store the nodes in the DAG resulted by trie-folding of a randomly generated trie with height $h$ and next hop distribution $p_1, \ldots, p_N$ is at most

$$
\left( 1 + \log_2 \left( \frac{h}{h - \log_2 \left( \frac{h}{H_O} \right) + \log_2(\delta)} \right) \log_2(\delta) + \frac{\delta}{\delta - 1} \right) \left( \frac{2H_O 2^h + 6h}{2} \right)
$$

(27)

when $H_O \geq \frac{h}{2^h}$ and $\delta$ is a finite number.

Note that the above bound asymptotically leads to

$$
\left( 1 - \log_2(\delta) + \frac{\delta}{\delta - 1} \right) \frac{2H_O}{(6 - 2 \log_2 H_O)} H_O
$$

bits for each leaf when $h \rightarrow \infty$. For $\delta = 2$ it is $H_O < 1$ and

$$
(6 - 2 \log_2 H_O) H_O.
$$