Exact Worst-Case TCAM Rule Expansion

Ori Rottenstreich, Rami Cohen, Danny Raz and Isaac Keslassy

***Abstract*—In recent years, hardware-based packet classification has became an essential component in many networking devices. It often relies on ternary content-addressable memories (TCAMs), which can compare in parallel the packet header against a large set of rules. Designers of TCAMs often have to deal with unpredictable sets of rules. These result in highly variable rule expansions, and can only rely on heuristic encoding algorithms with no reasonable guarantees.**

In this paper, given several types of rules, we provide new upper bounds on the TCAM worst-case rule expansions. In particular, we prove that a W-bit range can be encoded in W TCAM entries, improving upon the previously-known bound of $2W-5$. We further prove the optimality of this bound of W for prefix encoding, using new analytical tools based on independent sets and alternating paths. Next, we generalize these lower bounds to a new class of codes called hierarchical codes that includes both binary codes and Gray codes. Last, we propose a modified TCAM architecture that can use additional logic to significantly reduce the rule expansions, both in the worst case and using real-life classification databases.

***Index Terms*—TCAM, Packet Classification, Range Encoding.**

I. Introduction

A. Background

Packet classification is the key function behind many network applications, such as routing, filtering, security, accounting, monitoring, load-balancing, policy enforcement, differentiated services, virtual routers, and virtual private networks [4]–[7]. For each incoming packet, a packet classifier compares the packet header fields against a list of rules, e.g. from access control lists (ACLs), then returns the first rule that matches the header fields, and applies a corresponding action on the packet.

Today, hardware-based ternary content-addressable memories (TCAMs) are the standard devices for high-speed packet classification [8], [9]. TCAMs are associative-memory devices that match packet headers using fixed-width ternary arrays composed of 0s, 1s, and $*$s (don't care). For each packet, TCAM devices can check all rules in parallel, and therefore can typically reach higher line rates than software-based classification algorithms [4]–[6], [10]–[13]. For instance, the 55 nm CMOS-based NL9000 TCAM device can run over 1 billion searches per second on headers of up to 320 bits [8].

However, *power consumption* constitutes a bottleneck for TCAM scaling [14]. Given the same access rate, a TCAM chip can consume 30 times more power than an equivalent SRAM chip with a software-based solution [15]. As a consequence, in the Cisco CRS-1 core router, classification and forwarding constitute a third of all power consumption, the highest usage of power together with the power management devices such as fans, which constitute another third [16].

TCAM devices run each search in parallel on all their entries, therefore their power consumption is proportional to their number of searched entries. Unfortunately, this number of entries is often larger than the number of classification rules. This is because there are two types of rules: simple rules (exact- and prefix-matches), which need a single entry per rule; and range rules, which can need many entries per rule, thus causing range expansion.

Today, *TCAM power consumption is mostly and increasingly due to range expansion*. Typically, while range rules constitute a minority of the rules, they also cause the majority of the entries, and therefore the majority of the TCAM power consumption [17]. In addition, there is evidence that the percentage of range-based rules is increasing. For instance, a comparison of two typical classification databases from 1998 and 2004 shows that the total percentage of range-based rules has increased from 1.3% to 13.3%, including an emergence of rules with two range-fields from 0% to 1.5% and an increase in the number of diverse ranges [18]. Unfortunately, as the number of range-based rules increases in an unpredictable way, it is unclear whether it is possible to provide any reasonable guarantee on the worst-case number of TCAM entries needed to encode them.

The goal of this paper is to gain a more fundamental understanding of the worst-case number of TCAM entries needed to encode a rule. Our objective is to provide upper bounds on the worst-case rule expansion. We also try to develop lower bounds on the rule expansion and examine the tightness of the upper bounds. These bounds would characterize the theoretical capacity of TCAM devices depending on the complexity of the rules: e.g., single-field or multiple-field range rules, using simple or complex ranges, either alone or in interaction with other rules. We also ask whether these lower bounds on the worst-case expansion can be improved using more general codes such as Gray codes. In a sense, we want to help define the *TCAM capacity region*.

B. Related Work

It is well-known that each range defined over a W-bit field can be encoded in $2W-2$ entries for $W \geq 2$ with an *internal* expansion, i.e. an expansion that only uses entries from within the range [19]. More generally, the product of d ranges defined on d different fields of size W each can be internally encoded in up to $(2W-2)^d$ entries, which amounts to 900 TCAM entries for $d = 2$ port range-fields of 16 bits each [4]. For instance, assume that $W = 3$, and that we want to internally

O. Rottenstreich and I. Keslassy are with the Department of Electrical Engineering, Technion, Israel (e-mails: or@tx.technion.ac.il, isaac@ee.technion.ac.il).

R. Cohen is with Haifa Research Lab, IBM, Israel (e-mail: ramic@cs.technion.ac.il).

D. Raz is with the Department of Computer Science, Technion, Israel (e-mail: danny@cs.technion.ac.il).

This paper is an extended version of [1]–[3].

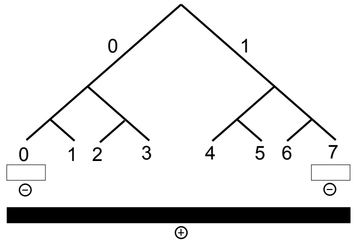


Fig. 1. External encoding of $R = [1, 6]$.

encode the single-field range $R = [1, 6] \subseteq \left[0, 2^W - 1\right]$ so that packets in that range get accepted, while others get denied (default action). Then we get the following $2W - 2 = 4$ TCAM entries, not counting the last default entry:

$$\begin{pmatrix} 001 & \rightarrow & \text{accept} \\ 01* & \rightarrow & \text{accept} \\ 10* & \rightarrow & \text{accept} \\ 110 & \rightarrow & \text{accept} \\ (*** & \rightarrow & \textit{deny}) \end{pmatrix}$$

A first improvement of the $2W - 2$ result has relied on non-prefix *internal* TCAM encoding and a connection to Boolean DNF (disjunctive normal form) to show a $2W - 4$ upper-bound [**?**]. A second improvement has kept prefix encoding but relied on Gray codes instead of binary codes to reduce the worst-case *internal* range expansion from $2W-2$ to $2W-4$ for W sufficiently large [17]. This result has since been improved to $2W - 5$ using a more complex encoding [21].

These results, however, do not consider the full potentiality of TCAM encoding, and in particular the *order of the entries*. For instance, Fig. 1 shows how the example above could be encoded in only 3 TCAM entries using an *external* encoding that exploits a different entry order.

$$\begin{pmatrix} 000 & \rightarrow & \text{deny} \\ 111 & \rightarrow & \text{deny} \\ *** & \rightarrow & \text{accept} \end{pmatrix}$$

We can see that the range exterior (complementary) is encoded first, and then the range itself is encoded indirectly later. This encoding does not require any changes in the packet header nor in the TCAM architecture. It makes use of the inherent property of TCAMs of returning the action of the first matching entry. First, a header from the range complementary matches one of the first two entries as well as the last entry, and therefore is denied. Likewise, a header within the range matches only the last entry and is thus accepted. Likewise, in this paper, we consider all possible TCAM entry orders when providing worst-case bounds.

Besides the papers above, there is extensive literature on providing *efficient heuristics* for TCAM rule expansion. These rely, for example, on redundancy removal, truth table equivalency, additional bits, additional TCAM hardware, dynamic programming, and topological transformation [4]–[6], [15], [22]–[29]. However, while these heuristics are often efficient, they often focus on *average-case* instead of *worst-case* performance. In addition, papers that are interested in worst-case performance do not provide new worst-case bounds [30].

Lower bounds on encoding length have more rarely been considered. If encoding is constrained to be *internal*, the worst-case code length is known to be at least W [17]. Also, an independent set of minterms in sum-of-products expressions is presented in [31]. However, none of these consider *external* encoding, and therefore they do not fully exploit TCAM properties.

C. Contributions

This paper investigates worst-case rule expansions in TCAMs.

In the first part, we consider single-field ranges of W-bit elements and attempt to encode them using efficient guaranteed upper bounds. We first consider W-bit extremal ranges of the form $[0, x]$, and prove that they can be encoded in $g(W) \leq \left\lceil \frac{W+1}{2} \right\rceil$ TCAM entries, nearly half the best-known bound of W entries [17].

Later, we consider regular ranges of the form $[x_1, x_2]$, and prove that they can always be encoded in $f(W) \leq W$ TCAM entries. Therefore, for large W, this is nearly half the size of the best-known binary bound for prefix TCAM encoding of $2W - 2$ and best-known overall bound of $2W - 5$ [19], [21].

We then introduce new analytical tools that are suited for TCAM analysis. We first define the hull $H(a^1, \ldots, a^n)$ of n binary strings $a^1, \ldots, a^n$, and show that these strings match a TCAM entry iff all the strings in their hull $H(a^1, \ldots, a^n)$ match this TCAM entry. We use this property to define an independent set of n points using some specific hull-based alternating path, and demonstrate that *an independent set of n points cannot be encoded in less than n TCAM entries*, given any arbitrary TCAM entries, in any order, and with any corresponding actions.

Next, we use this strong property to prove that the upper-bound on the expansion $g(W)$ of extremal ranges is tight. Since our encoding only uses TCAM prefix entries, it is therefore *optimal* both among prefix-based encodings and general encodings.

Then, we also prove that the upper bound on the range expansion $f_p(W)$ is tight as well among prefix-based encodings, hence proving *optimality* in this encoding class (but not among non-prefix encodings).

Next, we show that our lower bounds on the general binary encoding still hold for a new and more general class of codes, including Gray codes [17].

Later, we prove that any union of k ranges of W-bit elements can be encoded in at most $k \cdot (W + 1)$ TCAM entries and then improve this bound. Further, we show that our encoding bound is asymptotically optimal as $k \to \infty$.

Next, we consider multidimensional ranges and present an upper bound on the expansion of a single multidimensional range. The upper bound is linear instead of exponential in the number of fields.

We would like to emphasize that all these bounds for the encoding of a single one-dimensional range, a single multidimensional range and a union of one-dimensional ranges are satisfied in any conventional TCAM architecture and do not require any additional logic or changes to the architecture.

Finally, we conclude by illustrating the results of our suggested schemes in the worst case as well as in the average case.

In the Appendix, we also propose a modified TCAM architecture that can use additional logic to significantly reduce the rule expansions for the encoding of a set of multidimensional ranges. We also present additional simulation results based on large synthetic classifiers generated by the ClassBench benchmark tool [**?**] and using real-life classification databases.

We would like to stress that this paper does not attempt to provide the best encoding scheme for any possible classification database. Instead, it simply provides some insights of the worst-case TCAM efficiency given simple classification rules.

Furthermore, note that our proposed encoding for a general classification database (and not a single range) requires additional logic inside the TCAM. However, we are not the first to suggest such changes. For instance, [23] and [24] suggest additional logic for range matching and [33] for a parallel error detection.

Paper Organization: We start with preliminary definitions in Section II. Then, in Section III we prove upper bounds on the range expansions of extremal ranges and general ranges later. In Sections IV, V we present general analytical tools in order to show that those upper bounds are tight. Next, in Section VI we provide lower bounds given hierarchical codes. Later, in Section VII we deal with the encoding of a union of ranges and with multidimensional ranges in Section VIII. We evaluate the suggested encoding schemes in Section IX. Last, in the Appendix we suggest several TCAM architectures that enable us to implement efficient encoding of a set of multidimensional ranges and present additional simulation results.

II. Model and Notations

A. Terminology

We first formally define the terminology used in this paper. We initially assume a binary code expansion, and will later revisit this assumption. For simplicity, whenever there will be no confusion, we also do not distinguish between a W-bit binary string (in $\{0,1\}^W$) and its value (in $[0, 2^W - 1]$).

Definition 1 (Header): A packet *header* $x = (x_1, \ldots, x_d) \in \{0,1\}^W$ is a W-bit string defined on the d fields $(F_1, \ldots, F_d)$. Each sub-string x_i of length W_i represents field F_i, with $\sum W_i = W$.

Example 1: A header could typically consist of the following $d = 5$ fields: $(F_1, \ldots, F_5) =$ (source IP address, destination IP address, source port number, destination port number, and protocol type), of respective lengths $(W_1, \ldots, W_5) = (32, 32, 16, 16, 8)$ bits.

Definition 2 (Range Rule): The *range rule* R_i in field F_i represents a set of allowed strings over $\{0,1\}^{W_i}$. It is defined as an integer range $[r_1, r_2]$, where r_1 and r_2 are W_i-bit integers and $r_1 \leq r_2$. A packet header sub-string $x_i \in \{0,1\}^{W_i}$ is said to match R_i whenever $x_i \in [r_1, r_2]$.

In particular, the range rule R_i could be a *prefix rule*, with a prefix $r' \in \{0,1\}^k$ of size $k \in [0, W_i]$, $r_1 = \{r'\}\{0\}^{W_i - k}$, and $r_2 = \{r'\}\{1\}^{W_i - k}$. It is an *exact match* with $k = W_i$ and r_1=r_2.

Typically, a rule defined on headers from Example 1 includes range rules in two fields: the source port field and the destination port field.

Definition 3 (Rule): A classification *rule* $R = ((R_1, \ldots, R_d) \to a)$ is defined as the union of a set of range rules (predicates) $(R_1, \ldots, R_d)$ defined over fields $(F_1, \ldots, F_d)$, and an action (decision) $a \in \mathcal{A}$, where $\mathcal{A}$ is a set of legal actions (e.g. $\mathcal{A} = \{\text{'accept'}, \text{'deny'}, \text{'accept with logging'}\}$). A packet header $x = (x_1, \ldots, x_d)$ matches a rule R iff each x_i matches R_i.

Definition 4 (Classifier): A *classifier* $C = (R^1, ..., R^{n(C)})$ is an ordered set of $n(C)$ classification rules. For each header $x \in \{0,1\}^W$, let $R^j = \left((R_1^j, \ldots, R_d^j) \to a^j\right)$ be the first rule matched by x. Then the classifier effectively defines a *classifier function* $\alpha : \{0,1\}^W \to \mathcal{A}$ that returns an action for each header so that $\alpha(x) = a^j$. We assume that the last rule $R^{n(C)}$ is matched by all headers and returns a default action $a_d \in \mathcal{A}$, and therefore the classifier is complete and α is always defined.

Definition 5 (TCAM entry): A *TCAM entry* $S \to a$ is defined as the union of a TCAM rule $S = (s_1, \ldots, s_W) \in \{0,1,*\}^W$, where $\{0,1\}$ are bit values and $*$ stands for *don't-care*, and an action $a \in \mathcal{A}$. A W-bit string $b = (b_1, \ldots, b_W)$ matches S, denoted as $b \in S$, iff for all $i \in [1, W]$, $s_i \in \{b_i, *\}$.

Definition 6 (TCAM Encoding Scheme): A *TCAM encoding scheme* ϕ is said to map a function α to an ordered set of $n_\phi(\alpha)$ TCAM entries $\left(S_1 \to a_1, \ldots, S_n \to a_{n_\phi(\alpha)}\right)$ using a default action $a_d \in \mathcal{A}$ iff for any header $x \in \{0,1\}^W$, either the first TCAM entry $S^j \to a_j$ matching x satisfies $\alpha(x) = a_j$, or no TCAM entry matches x and $\alpha(x) = a_d$. The number $n_\phi(\alpha)$ of non-default TCAM entries is called the *expansion* of encoding scheme ϕ for the classifier function α.

In the Introduction, we saw an example of TCAM encoding of a single-field range classifier function α, with $\alpha([1,6]) =$ 'accept' and $\alpha(\{0\} \cup \{7\}) =$ 'deny'. In the remainder of the paper, we will always assume for simplicity that the default action is $a_d = 0$. Each single-field *range* R is uniquely characterized by its range indicator function α_R, which takes a value of 1 on R and 0 outside R. We will use *range* to indicate either R or its indicator function α_R.

Definition 7 (Prefix Encoding Scheme): A TCAM *prefix encoding scheme* ϕ is a TCAM encoding scheme such that for any TCAM entry $S \to a$ with $S = (s_1, \ldots, s_W) \in \{0,1,*\}^W$, if $s_j = \{*\}$ for some $j \in [0, W]$, then $s_{j'} = \{*\}$ for any $j' \in [j, W]$.

We will denote as Φ_p the set of all prefix encoding schemes, and the general set of encoding schemes including non-prefix schemes as Φ, so that $\Phi_p \subset \Phi$.

B. Optimal Range Expansion Problem

We want to find a TCAM prefix encoding scheme $\phi \in \Phi_p$ that minimizes the worst-case TCAM prefix expansion $n_\phi(\alpha_R)$ over all possible range functions α_R. We first focus on prefix encoding schemes, and later consider non-prefix schemes. To do so, we will first define extremal ranges, then define the TCAM-expansion minimization problem over all extremal ranges, before defining the TCAM-expansion minimization problem over all possible ranges

Definition 8 (Extremal Ranges): Let us define two types of *extremal ranges* over $[0, 2^W - 1]$.
(i) A *left-extremal* range R^{LE} denotes a range of the form $R^{LE} = [0, y]$ for some arbitrary value of y.
(ii) Likewise, a *right-extremal* range R^{RE} denotes a range of the form $R^{RE} = [y, 2^W - 1]$ for some arbitrary value of y.

A *non-extremal* range $R = [y_1, y_2]$ is a range such that $0 < y_1 \leq y_2 < 2^W - 1$. Therefore, a range is either left-extremal, right-extremal, or non-extremal. We now want to define our optimization problem, first over all range functions, then over extremal ranges.

Definition 9 (Range Expansion): For any positive integer W and any TCAM prefix encoding scheme $\phi \in \Phi_p$, the *range expansion of* ϕ, denoted $f_\phi(W)$, is the worst-case TCAM expansion $n_\phi(\alpha_R)$ over all possible range functions α_R, i.e.

$$f_\phi(W) = \max_{R \subseteq [0, 2^W - 1]} n_\phi(\alpha_R), \tag{1}$$

We now want to optimize the range expansion over all possible encoding schemes $\phi \in \Phi$. Then the *range expansion* $f(W)$ is defined as the best-achievable range expansion for W-bit ranges given all encoding schemes, i.e.

$$f(W) = \min_{\phi \in \Phi} \left(\max_{R \subseteq [0, 2^W - 1]} n_\phi(\alpha_R) \right). \tag{2}$$

Likewise, we define $f_p(W)$ as the best-achievable range expansion given all *prefix* encoding schemes $\phi \in \Phi_p$.

Definition 10 (Extremal Range Expansion): Define the *left-extremal range expansion* $g(W)$ and *right-extremal range expansion* $g'(W)$ as the best-achievable range expansion given all encoding schemes $\phi \in \Phi$ for left-extremal and right-extremal ranges, respectively. Then,

$$g(W) = \min_{\phi} \max_{y:\, 0 \leq y \leq 2^W - 1} n_\phi(\alpha_{[0,y]}), \tag{3}$$

$$g'(W) = \min_{\phi} \max_{y:\, 0 \leq y \leq 2^W - 1} n_\phi(\alpha_{[y, 2^W - 1]}). \tag{4}$$

Likewise, define $g_p(W)$ and ${g_p}'(W)$ over all prefix encoding schemes $\phi \in \Phi_p$.

III. Range Expansion Guarantees

A. Upper-Bound on the Extremal Range Expansion

We now want to provide range expansion guarantees by proving upper bounds on the range expansions of extremal ranges first, and general ranges later. To do so, we first prove that left-extremal and right-extremal ranges have the same range expansion.

Lemma 1: The left-extremal and right-extremal range expansions are the same, i.e. for all $W \in \mathbb{N}^*$, $g(W) = g'(W)$.

Proof: For any $y \in [0, 2^W - 1]$, the value obtained when inverting the bits in the binary representation of y is $y' = (2^W - 1) - y$. In particular, for $y = 0$ we get $y' = 2^W - 1$. Therefore, a left-extremal range $R^{LE} = [0, y]$ is transformed into a right-extremal range $R^{RE} = [(2^W - 1) - y, 2^W - 1]$, and vice-versa. Consequently, given W-bit binary strings, the bit inversion defines a bijection between the set of left-extremal ranges and the set of right-extremal ranges.

Let $(S_1 \rightarrow a_1, \ldots, S_n \rightarrow a_n)$ denote the n TCAM entries encoding a left-extremal range. Then, by inverting the $\{0, 1\}$ symbols in each S_i, we get n TCAM entries encoding the corresponding right-extremal range. Therefore, we have $g'(W) \leq g(W)$, and likewise $g(W) \leq g'(W)$, hence the result. ∎

We now want to find $g(W)$. To do so, we will first prove the following lemma on range shifting. The lemma shows that if we shift a range $R \subseteq [0, 2^W - 1]$ by a positive multiple of 2^W, then the range expansion of the shifted range does not need more TCAM entries, because we only need to add a prefix to the TCAM expansion of R.

Lemma 2: Consider a W-bit range $R = [y_1, y_2] \subseteq [0, 2^W - 1]$, a w-bit value $x \in [0, 2^w - 1]$, and a shifted range $R' = [x \cdot 2^W + y_1, x \cdot 2^W + y_2] \subseteq [0, 2^{W+w} - 1]$. Then the range expansion of the shifted range R' is no more that that of R.

Proof: Let $(S_1 \rightarrow a_1, \ldots, S_n \rightarrow a_n)$ denote the TCAM entries encoding R, where each S_i is of length W. For each $i \in [1, n]$, let $S'_i = \{x\} \cdot S_i$ denote the $(w+W)$-bit concatenation of x and S_i. Then $(S'_1 \rightarrow a_1, \ldots, S'_n \rightarrow a_n)$ has the same number of TCAM entries and encodes R' (Definition 6). ∎

Example 2: For $W = 3$, as shown in the Introduction, the range $R^1 = [1, 6]$ can be encoded with the three TCAM entries $(000 \rightarrow 0, 111 \rightarrow 0, *** \rightarrow 1)$ using default action 0. Likewise, the range $R' = [17, 22] = [2 \cdot 2^3 + 1, 2 \cdot 2^3 + 6]$ can be encoded by simply adding the prefix 10 to all three TCAM entries: $(\mathit{10}000 \rightarrow 0, \mathit{10}111 \rightarrow 0, \mathit{10}*** \rightarrow 1)$.

We are now ready to characterize $g(W)$. We first find an upper-bound on $g(W)$ by constructing an encoding scheme, and then later show that this upper-bound is actually tight. The following result improves by a factor of nearly two the best-known bound of W [17].

Theorem 1: For all $W \in \mathbb{N}^*$, the extremal range expansion satisfies the following upper-bound:

$$g(W) \leq \left\lceil \frac{W+1}{2} \right\rceil. \tag{5}$$

Proof: By Definition 10 of $g(W)$, we only need to exhibit an encoding scheme ϕ that manages to encode each left-extremal range $R^{LE} = [0, y] \subseteq [0, 2^W - 1]$ using at most $\left\lceil \frac{W+1}{2} \right\rceil$ non-default TCAM entries. Let's do it *by induction* on $W \in \mathbb{N}^*$.

Induction basis: For $W = 1$, the only left-extremal ranges are $R_1^{LE} = [0, 0]$ and $R_2^{LE} = [0, 1]$, which are respectively encoded by $(0 \rightarrow 1)$ and $(* \rightarrow 1)$, i.e. in at most $\left\lceil \frac{1+1}{2} \right\rceil = 1$ TCAM entry each.

For $W = 2$, there are four left-extremal ranges: $R_1^{LE} = [0, 0]$ is encoded as $(00 \rightarrow 1)$, $R_2^{LE} = [0, 1]$ is encoded as $(0* \rightarrow 1)$, $R_3^{LE} = [0, 2]$ is encoded as $(0* \rightarrow 1, 10 \rightarrow 1)$, and $R_4^{LE} = [0, 3]$ is encoded as $(** \rightarrow 1)$, i.e. in at most $\left\lceil \frac{2+1}{2} \right\rceil = 2$ TCAM entries each.

Induction step: Let's now assume that the result is correct until $W - 1$, and prove it for W. We will show that

$$g(W) \leq 1 + g(W - 2), \tag{6}$$

which suffices to prove the result, since it would imply that

$$g(W) \leq 1 + \left\lceil \frac{(W-2)+1}{2} \right\rceil = \left\lceil \frac{W+1}{2} \right\rceil.$$

Consider the left-extremal range $R^{LE} = [0, y] \subseteq [0, 2^W - 1]$. We will cut the W-bit range $[0, 2^W - 1]$ into four equal sub-ranges of size 2^{W-2}, and show that no matter the sub-range to which y belongs, R^{LE} can be encoded in $1 + g(W-2)$ TCAM entries, thus proving Equation (6).

(i) If $y \in [0, 2^{W-2} - 1]$, then R^{LE} can be seen as a $(W-2)$-bit left-extremal range, which can be encoded in $g(W-2)$ entries.

(ii) If $y \in [2^{W-2}, 2^{W-1} - 1]$, then we first encode the sub-range $[0, 2^{W-2} - 1]$ using a single TCAM entry $(\{00\}\{*\}^{W-2} \to 1)$, and then by Lemma 2, we can encode the remaining sub-range $[2^{W-2}, y]$ by adding at most $g(W-2)$ TCAM entries (using the $\{01\}$ prefix for all entries). Thus, we use a total of at most $1 + g(W-2)$ TCAM entries.

(iii) Likewise, if $y \in [2^{W-1}, 2^{W-1} + 2^{W-2} - 1]$, then we first encode the sub-range $[0, 2^{W-1} - 1]$ using a single TCAM entry $(\{0\}\{*\}^{W-1} \to 1)$, and then by Lemma 2 we encode the remaining sub-range $[2^{W-1}, y]$ by adding at most $g(W-2)$ TCAM entries, thus using at most $1 + g(W-2)$ TCAM entries.

(iv) Last, if $y \in [2^{W-1} + 2^{W-2}, 2^W - 1]$, we actually first encode the complementary range $[y+1, 2^W - 1]$, which by Lemma 1 can be done in up to $g'(W-2) = g(W-2)$ TCAM entries. To do so, for each TCAM entry we use its complementary action. Then, we add the TCAM entry $(\{*\}^W \to 1)$ to encode the range, thus using again at most $1 + g(W-2)$ TCAM entries. Since the four cases imply Equation (6), we finally get the result by induction. ■

We can actually obtain a stronger result by showing that the worst-case extremal range expansion $g_p(W)$ over all *prefix* encoding schemes $\phi \in \Phi_p$ satisfies the same upper bound.

Theorem 2: For all $W \in \mathbb{N}^*$, $g_p(W)$ satisfies

$$g(W) \leq g_p(W) \leq \left\lceil \frac{W+1}{2} \right\rceil. \qquad (7)$$

Proof: We note that we only used TCAM prefix entries in the proof of the previous theorem, and therefore the encoding scheme ϕ used in the proof satisfies $\phi \in \Phi_p$. All other arguments stay the same, and in particular Lemma 1 and Lemma 2 are still valid within Φ_p, hence $g_p(W) \leq \lceil \frac{W+1}{2} \rceil$. Last, since $\Phi_p \subset \Phi$, $g(W) \leq g_p(W)$ by definition. ■

B. Upper-Bound on the Range Expansion

We now want to find an upper-bound on the range expansion $f(W)$ by constructing an efficient encoding scheme. We will later show that this upper-bound is actually tight for prefix encoding schemes.

Theorem 3: For all $W \in \mathbb{N}^*$, the worst-case range expansion satisfies the following upper-bound:

$$f(W) \leq W. \qquad (8)$$

Proof: Let's prove this by induction on $W \geq 1$.

Induction basis: For $W = 1$, all non-empty ranges are extremal, therefore the result follows by Theorem 1. In addition, for $W = 2$, all non-empty and non-extremal ranges are either single points, or $[1, 2]$, which can be encoded in two entries.

Induction step: Now let $W \geq 3$, and assume the claim is true until $W - 1$. Consider any range $R \subseteq [0, 2^W - 1]$, and cut it into four possibly-empty sub-ranges, that correspond to its intersection with the four consecutive sub-spaces of size 2^{W-2} of the space $[0, 2^W - 1]$ of size 2^W: $R = R^1 \cup R^2 \cup R^3 \cup R^4$, with $R^1 = R \cap [0, 2^{W-2} - 1]$, $R^2 = R \cap [2^{W-2}, 2^{W-1} - 1]$, $R^3 = R \cap [2^{W-1}, 2^{W-1} + 2^{W-2} - 1]$, and $R^4 = R \cap [2^{W-1} + 2^{W-2}, 2^W - 1]$. We want to show that R can be encoded in at most W TCAM entries. Distinguish between several cases:

(i) If $R = R^1 \cup R^2 \subseteq [0, 2^{W-1} - 1]$, i.e. $R^3 = R^4 = \emptyset$, then by induction R can be encoded in at most $W - 1$ entries.

(ii) Else if $R = R^3 \cup R^4 \subseteq [2^{W-1}, 2^W - 1]$, i.e. $R^1 = R^2 = \emptyset$, then this is just a shifted version of the previous case and, by Lemma 2, R can be encoded in at most $W - 1$ entries.

(iii) Else $|R^2| > 0$ and $|R^3| > 0$, because R is a range. Let's distinguish between two similar sub-cases. *(a)* If $R^4 = \emptyset$, then $R = (R^1 \cup R^2) \cup R^3$. $(R^1 \cup R^2)$ is a right-extremal range on $[0, 2^{W-1} - 1]$, and by Lemma 1, can be encoded in $g(W-1)$ TCAM entries. Further, R^3 is just a shifted version of a left-extremal range, and by Lemma 2, can be encoded in $g(W-2)$ TCAM entries. *(b)* Likewise, if $R^1 = \emptyset$, then $R = R^2 \cup (R^3 \cup R^4)$, R^2 can be encoded in $g(W-2)$ TCAM entries, and $(R^3 \cup R^4)$ in $g(W-1)$ TCAM entries. Therefore in both sub-cases, by Theorem 1, R can be encoded in up to

$$g(W-1) + g(W-2) \leq \left\lceil \frac{W}{2} \right\rceil + \left\lceil \frac{W-1}{2} \right\rceil = W$$

TCAM entries. Note that in both sub-cases, the TCAM entries can be merged because the construction in the proof of Theorem 1 is prefix-based and limited to the range sub-space, therefore there are no conflicts between the entries corresponding to two distinct range sub-spaces.

(iv) Last, if all $|R^i| > 0$ for $i \in [1, 4]$, then we use 2 different techniques according to the parity of W.

If W is even, we encode R in a very similar way to what we did in the previous case. $(R^1 \cup R^2)$ is a right-extremal range on $[0, 2^{W-1} - 1]$ and, by Lemma 1, can be encoded in $g(W-1)$ TCAM entries. Further, $(R^3 \cup R^4)$ is just a shifted version of a left-extremal range and, by Lemma 2, can be encoded in $g(W-1)$ TCAM entries. Again there are no conflicts and by Theorem 1, R can be encoded in up to

$$g(W-1) + g(W-1) \leq 2 \cdot \left\lceil \frac{W}{2} \right\rceil = W$$

TCAM entries.

If W is odd, we first encode the range complementary, and then use an additional TCAM entry with action 1 $(\{*\}^W \to 1)$ to encode the remaining range. To encode the range complementary, we encode the left-extremal range $\left([0, 2^{W-2} - 1] \setminus R^1\right)$ in $g(W-2)$ entries, and the shifted right-extremal range $\left([2^{W-1} + 2^{W-2}, 2^W - 1] \setminus R^4\right)$ in $g(W-2)$ entries as well (using Lemma 2 on shifts and Lemma 1 on right-extremal ranges). Again there are no conflicts between the entries. Therefore R can be encoded in up to

$$2g(W-2) + 1 \leq 2 \left\lceil \frac{W-1}{2} \right\rceil + 1 \leq W$$

entries, and considering all cases, $f(W) \leq W$. ■

As for extremal ranges, we also get the corresponding stronger result on prefix encoding schemes.

Theorem 4: For all $W \in \mathbb{N}^*$, $f_p(W)$ satisfies

$$f(W) \le f_p(W) \le W. \tag{9}$$

Proof: We note that we only used TCAM prefix entries in the proof of the previous theorem, and therefore the encoding scheme ϕ used in the proof is also in Φ_p. All other arguments stay the same and are valid within Φ_p. Last, since $\Phi_p \subset \Phi$, $f(W) \le f_p(W)$ by definition. ■

IV. Hull, Independence, and Alternating Paths

We now want to introduce new general analytical tools that will help us analyze the minimum number of TCAM entries needed to encode a classifier function. Intuitively, given any range that we need to encode, we will want to exhibit n points that are *independent* in some sense, and prove that they cannot be encoded in less than n TCAM entries.

First, we define the *hull* of a set of W-bit strings in the W-dimensional string space (this hull is also known as the *isothetic rectangle hull*, *minimum bounding rectangle*, or *minimum axis-aligned bounding box* in different contexts).

Definition 11 (Hull): Let $(n, W) \in \mathbb{N}^{*2}$, and consider n strings $a^1, \ldots, a^n$ of W bits each, with $a^i = (a^i_1, \ldots, a^i_W)$ for each $i \in [1, n]$. Then the hull of $\{a^1, \ldots, a^n\}$, denoted $H(a^1, \ldots, a^n)$, is the smallest cuboid containing $a^1, \ldots, a^n$ in the W-dimensional string space, and is defined as

$$\begin{aligned} H(a^1, \ldots, a^n) = \quad & \{x = (x_1, \ldots, x_W) \in \{0,1\}^W \mid \\ & \forall j \in [1, W], x_j \in \{a^1_j, \ldots, a^n_j\}\}. \end{aligned} \tag{10}$$

We can now relate the hull of a set of points to the TCAM entries that they jointly match.

Proposition 1: Let $(n, W) \in \mathbb{N}^{*2}$, and consider n strings $a^1, \ldots, a^n$ of W bits each. Then $a^1, \ldots, a^n$ match the same TCAM entry iff all the strings in the hull $H(a^1, \ldots, a^n)$ match this TCAM entry.

Proof: On the one hand, by Equation (10) defining the hull, we always have $\{a^1, \ldots, a^n\} \subseteq H(a^1, \ldots, a^n)$. Therefore, if all strings in $H(a^1, \ldots, a^n)$ match a TCAM entry, so does any a^i.

On the other hand, assume that $a^1, \ldots, a^n$ match a TCAM entry $S \to a$, with $S = (s_1, \ldots, s_W) \in \{0, 1, *\}^W$. Then by Definition 5 of TCAM entry matching, for all $i \in [1, n]$ and for all $j \in [1, W]$, $s_j \in \{a^i_j, *\}$. Now consider $x = (x_1, \ldots, x_W) \in H(a^1, \ldots, a^n)$. Then by Equation (10), for all $j \in [1, W]$, $x_j \in \{a^1_j, \ldots, a^n_j\}$. Therefore, for each bit j, either all a^i_j are equal, and x_j obviously matches s_j like all a^i_j, or some of them are distinct, and then $s_j = *$, so x_j matches s_j again. ■

Using the definition of the hull, we now define *independent sets* of points, and then show that an independent set of n points cannot be encoded in less than n TCAM entries. Therefore, this result enables us to simply exhibit an appropriate independent set of points whenever we want to prove a lower bound on the expansion of a classifier function.

Definition 12 (Alternating Path and Independent Set): Let n and W be positive integers, and let $\alpha : \{0,1\}^W \to \{0,1\}$ be a classifier function. Then an *alternating path* A_n of size n is defined as an ordered set of $2n-1$ W-bit strings $A_n = (a^1, \ldots, a^{2n-1})$ that satisfies the following two conditions:
(i) Alternation: For $i \in [1, 2n-1]$,

$$\begin{aligned} &\alpha(a^1) = \alpha(a^3) = \cdots = \alpha(a^{2n-1}) = 1, \text{ and} \\ &\alpha(a^2) = \alpha(a^4) = \cdots = \alpha(a^{2n-2}) = 0. \end{aligned} \tag{11}$$

(ii) Hull: For any i_1, i_2, i_3 such that $1 \le i_1 < i_2 < i_3 \le 2n-1$,

$$a^{i_2} \in H(a^{i_1}, a^{i_3}). \tag{12}$$

In such an alternating path, $(a^1, a^3, a^5, \ldots, a^{2n-1})$ is an *independent set* of size n.

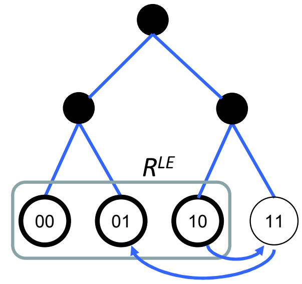


Fig. 2. Alternating path: R^{LE} requires at least two TCAM entries using any encoding scheme.

Example 3: As shown in Fig. 2, let $W = 2$, $n = 2$, and consider the left-extremal range $R^{LE} = [0, 2] = \{\{00\}, \{01\}, \{10\}\}$. Let $a^1 = 2 = \{10\}$, $a^2 = 3 = \{11\}$, and $a^3 = 1 = \{01\}$. Then $A_2 = (a^1, a^2, a^3)$ is an *alternating path of size* 2 and (a^1, a^3) is an independent set, because they satisfy the two conditions:
(i) Alternation: $a^1 \in R^{LE}$, $a^2 \notin R^{LE}$, $a^3 \in R^{LE}$.
(ii) Hull: $a^2 \in H(a^1, a^3)$, i.e. $\{11\} \in H(\{10\}, \{01\})$, because it shares its first bit with a^1 and its second bit with a^3.

Lemma 3: Let n be a positive integer, and $(a^1, \ldots, a^{2n+1})$ be an alternating path of size $n+1$. Then removing any two successive elements in the alternating path yields an alternating path of size n.

Proof: Removing elements a^i and a^{i+1} yields $(a^1, \ldots, a^{i-1}, a^{i+2}, \ldots, a^{2n+1})$ for any $i \in [1, 2n]$. Then the two conditions defined above for the alternating path still hold. First, odd elements should still yield action 1, and even elements action 0. Second, for any three elements in the list, the middle element is still in the hull of the other two, since it was already there before the removal of the two elements. ■

Theorem 5: A classifier function with an alternating path of size n cannot be encoded in less than n TCAM entries.

Proof: The proof is by induction on n.

For the induction basis, we can see that for $n = 1$, we need to encode at least one element with a non-default action of 1, therefore we need at least one TCAM entry.

For the induction step, we assume that we cannot encode a classifier function with an alternating path of size n in less than n TCAM entries, and want to show it for $n+1$ as well. To do so, we assume, by contradiction, that we can encode a classifier function with an alternating path $A_{n+1} = (a^1, \ldots, a^{2n+1})$ of size $n+1$ in less than $n+1$ TCAM entries. Then we consider the first TCAM entry $S \to a$ (as defined in Definition 6). In the full proof, presented in [34] for space reasons, we distinguish several cases according to the number of elements from A_{n+1} that are matched by this TCAM entry. We show that each of them leads to a contradiction according to Lemma 3. ■

V. Range Expansion Optimality

A. Extremal Range Expansion Optimality

Thanks to the tools developed above, we can now prove the following theorem, which shows that the upper-bound $g(W) \leq \lceil \frac{W+1}{2} \rceil$ proved in Theorem 1 is tight, and therefore that our iterative encoding scheme reaches the optimal extremal range expansion.

Theorem 6: The bound in Theorem 1 is tight, and therefore for all $W \in \mathbb{N}^*$, the extremal range expansion is exactly

$$g(W) = \left\lceil \frac{W+1}{2} \right\rceil . \tag{13}$$

Proof: We have to show that $g(W) \geq \lceil \frac{W+1}{2} \rceil$.

The case of $W = 1$ is trivial. To distinguish between the two left-extremal ranges $R_1^{LE} = [0, 0]$ and $R_2^{LE} = [0, 1]$, it is clear that we need at least one TCAM entry.

Assume $W \geq 2$. First, notice that for each even value of $W \in \mathbb{N}^*$, the upper-bound is the same for $g(W)$ and $g(W + 1)$, and is equal to $\left(\frac{W}{2} + 1\right)$, i.e. $\lceil \frac{W+1}{2} \rceil = \left\lceil \frac{(W+1)+1}{2} \right\rceil = \left(\frac{W}{2} + 1\right)$. Therefore, to prove the tightness of the upper-bound, it is sufficient to do it for the positive even values of W.

More specifically, for each positive even value of W, we simply need to exhibit a left-extremal range $R^{LE}(W) \subseteq [0, 2^W - 1]$ that cannot be encoded in less than $\left(\frac{W}{2} + 1\right)$ TCAM entries. As a consequence, this left-extremal range $R^{LE}(W)$ would also suffice to prove the tightness of the upper-bound for $W + 1$, because $R^{LE}(W) \subseteq [0, 2^W - 1] \subseteq [0, 2^{W+1} - 1]$, and $\left\lceil \frac{(W+1)+1}{2} \right\rceil = \frac{W}{2} + 1$.

Therefore, we assume that $W \geq 2$ is even. Define W-bit string $c(W) = 1010...10 = \{10\}^{\frac{W}{2}}$ with the binary value of

$$c(W) = \sum_{k=0}^{\frac{W}{2}-1} 2 \cdot 2^{2k} = \frac{2}{3}\left(2^W - 1\right). \tag{14}$$

Consider the left-extremal range $R^{LE}(W) = \left[0, \frac{2}{3}\left(2^W - 1\right)\right] = \left\{\{0\}^W, \ldots, c(W)\right\} \subseteq [0, 2^W - 1]$. In the full proof, presented in [34] for space reasons, we show that given $R^{LE}(W)$ we can obtain an alternating path of size $\frac{W}{2} + 1$. To do so, we define $a^1 = \{01\}^{\frac{W}{2}}$, and construct the alternating path $(a^1, \ldots, a^{W+1})$ by flipping each time the i^{th} bit of a^i to obtain a^{i+1}. We show that the odd-indexed elements are in $R^{LE}(W)$ while the even-indexed are not, proving the alternation property. We also show that $1 \leq i_1 < i_2 < i_3 \leq W + 1$, $a^{i_2} \in H(a^{i_1}, a^{i_3})$, and the hull property is satisfied as well. Then, the result follows by Theorem 5. ■

B. Range Expansion Optimality

The next theorem shows that the upper bound on the range expansion $f_p(W)$ from Theorem 4 is actually tight among all TCAM prefix encoding schemes, and therefore their prefix encoding scheme is optimal among all prefix encoding schemes for the worst-case range expansion.

Theorem 7: For all $W \in \mathbb{N}^*$, the optimal range expansion among all prefix encoding schemes is exactly

$$f_p(W) = W. \tag{15}$$

Proof: We have proved earlier, using an alternating path, that the expansion of extremal ranges on spaces of size 2^{W-1} is $g(W - 1) = \lceil \frac{W}{2} \rceil$.

We first assume that W is odd. We define $R^1 = \left[\frac{1}{3}\left(2^{W-1} - 1\right), 2^{W-1} - 1\right]$, $R^2 = \left[2^{W-1}, 2^{W-1} + \frac{2}{3}\left(2^{W-3} - 1\right)\right]$. We then build a hard-to-encode W-bit range $R = R^1 \bigcup R^2 = \left[\frac{1}{3}\left(2^{W-1} - 1\right), 2^{W-1} + \frac{1}{3} \cdot 2^{W-2} - \frac{2}{3}\right] \subseteq [0, 2^W - 1]$ composed of a shifted right-extremal range R^1 of size $c(W - 1)$ and a shifted left-extremal range R^2 of size $c(W - 3)$.

The first range R^1 is included in the sub-space $[0, 2^{W-1} - 1]$ of size 2^{W-1} and the second range R^2 in the sub-space $[2^{W-1}, 2^{W-1} + 2^{W-3} - 1]$ of size 2^{W-3}. Therefore in R^1 we can build an alternating path of size $g(W - 1)$, and in R^2 another one of size $g(W - 3)$.

There are two approaches we can use in order to encode R. We can either encode the range itself or encode the complimentary range first and then add the entry $(*^W \to 1)$. In the full proof, presented in [34] for space reasons, we show that using prefix encoding, no matter which way is chosen, we would then need a total number of (at least) W TCAM entries.

If W is even, similar considerations can show that the range $R = \left[\frac{2}{3}2^{W-1} - \frac{1}{3}, \frac{4}{3}2^{W-1} - \frac{2}{3}\right] \subseteq [0, 2^W - 1]$ cannot be encoded in less than W prefix TCAM entries. ■

Similarly to the equality $g(W) = g_p(W)$, we conjecture that we have the same equality here, i.e. that non-prefix expansions cannot obtain a better range expansion than prefix expansions.

Conjecture 1: For all $W \in \mathbb{N}^*$, $f(W) = f_p(W) = W$.

VI. Range Expansion with Hierarchical Codes

We saw in the Introduction that encoding internally using binary prefixes can be done in $2W-2$ entries per rule, but can be improved using Gray codes and similar codes to $2W-4$ and $2W-5$ entries per rule, respectively [17], [21]. It is natural to ask whether our lower bounds on the general binary encoding still hold with different codes, such as a Gray code.

We show that counter-intuitively, *Gray codes do not reduce the worst-case expansion*. We first define a general class of *hierarchical codes* that includes both binary codes and Gray codes, and then prove that they satisfy the exact same results on extremal range expansion and range expansion, respectively.

Let a code $\sigma : \{0,1\}^W \to \{0,1\}^W$ be a bijection that transforms a binary W-bit string representation into another W-bit string, and let Σ denote the set of all such codes. We first provide some useful definitions, and then prove that hierarchical codes satisfy several equivalent properties.

Definition 13 (Suffix Distance): The *suffix distance* $d_S(a, b)$ between two W-bit strings a and b is

$$d_S(a, b) = W - \max\{j \in [0, W] | \, (a_1, \ldots, a_j) = (b_1, \ldots, b_j)\}.$$

Definition 14 (Prefix Set): A *prefix set* $S \subseteq \{0,1\}^W$ is a set of all elements that share the same prefix, i.e. a W-bit string $a \in \{0,1\}^W$ and an index $j \in [0, W]$ exist such that

$$S = \{a_1, \ldots, a_j\} \{0,1\}^{W-j}.$$

Theorem 8 (Hierarchical Codes): For any $\sigma \in \Sigma$, the following three properties are equivalent:
(i) σ is a graph automorphism on the tree representation, i.e. it preserves all subtrees in the tree structure;
(ii) σ preserves prefix sets, i.e. if S is a prefix set, then $\sigma(S)$ is a prefix set as well;
(iii) σ preserves the suffix distance, i.e. $d_S(\sigma(a), \sigma(b)) = d_S(a, b)$.
We will denote by Σ^H the set of all codes satisfying these properties, and call them *hierarchical codes*.

Proof: We start by proving $(i) \Rightarrow (ii)$. Consider the subtree corresponding to the prefix set S. Since σ is a graph automorphism, its image is a subtree with the same size. This subtree corresponds to another prefix set S'.

Next, we show that $(ii) \Rightarrow (iii)$. If $d_S(a,b) = d$, then the minimal size of a prefix set that contains both a and b is 2^d. Let denote by S such a set. By property (ii), we have that $\sigma(S)$ is a prefix set of the same size (2^d) that contains both $\sigma(a)$ and $\sigma(b)$. Therefore, $d_S(\sigma(a), \sigma(b)) \leq d = d_S(a,b)$. In order to see that $d_S(\sigma(a), \sigma(b)) = d_S(a,b)$, we show that assuming that $d_S(\sigma(a), \sigma(b)) < d_S(a,b)$ leads towards a contradiction. We first observe that by property (ii) we must also have that if $\sigma(S)$ is a prefix set, then S is a prefix set as well, since the total number of prefix sets is S and $\sigma(S)$ is equal. Therefore, assuming property (ii), we can deduce that σ^{-1} preserves prefix sets as well. If $d_S(\sigma(a), \sigma(b)) < d = d_S(a,b)$, there exists a prefix set S' of size smaller than 2^d that contains both $\sigma(a)$ and $\sigma(b)$. From the corollary above we have that $\sigma^{-1}(S')$ is a prefix set smaller than 2^d that contains both a and b. Contradiction.

Last, we prove that $(iii) \Rightarrow (i)$. We use the equality between the suffix distance and half the distance in the tree between the corresponding leaves. Therefore, by property (iii) we have that the distance in the tree between any two nodes is also preserved under σ. Thus, again using the connection between the distance in the tree and the minimal size of a tree that contains two points, we must have that σ preserves all subtrees in the tree structure and is a graph automorphism. ■

Example 4: We want to show that both the binary code and the Gray code have these three properties. For the binary code, σ is the identity function and therefore preserves all subtrees in the tree structure, the prefix sets and the suffix distance. Thus, it satisfies these three properties.

For the Gray code we prove that it has property (i) by induction. For $W = 1$ the Gray code is the same as the binary code, and thus has the same properties. By the induction hypothesis, we assume that all the subtrees of size smaller than 2^{W-1} are preserved under the code. For a general W, from the reflection property of the Gray code, the values of $[0, 2^W - 1]$ are assigned first to the left subtree of size 2^{W-1} and later to the right one. Thus, the two subtrees of size 2^{W-1} are also preserved, and the Gray code satisfies property (i) and the two others by their equivalence.

TABLE I
EXAMPLE OF 2 ADDITIONAL CODES. ϕ IS HIERARCHICAL, ψ IS NOT.

	000	001	010	011	100	101	110	111
ϕ	011	010	001	000	100	101	110	111
ψ	000	010	100	110	111	101	011	001

Example 5: For $W = 3$, we present two additional codes, $\phi, \psi \in \Sigma$, defined in Table I. We show, by checking that the properties are satisfied, that $\phi \in \Sigma^H$ while $\psi \notin \Sigma^H$.

We start with ψ. For $a = 000, b = 111$, we have $d_S(a,b) = d_S(000, 111) = 3$. However, $d_S(\psi(a), \psi(b)) = d_S(\psi(000), \psi(111)) = d_S(000, 001) = 1$ Therefore, ψ does not satisfy property (iii) and ψ is not an hierarchical code.

Next, we examine the code ϕ. To show that it preserves prefix sets and satisfies property (ii), we consider all the possible prefix sets that contain more than one element. There are four prefix sets of size two: $S^1 = \{00\}\{0,1\}, S^2 = \{01\}\{0,1\}, S^3 = \{10\}\{0,1\}, S^4 = \{11\}\{0,1\}$ and two prefix sets of size four: $S^5 = \{0\}\{0,1\}^2, S^6 = \{1\}\{0,1\}^2$. We can see that ϕ maps S^1 to S^2, S^2 to S^1 and S^3, S^4, S^5, S^6 to themselves. Finally, the only prefix set of size 2^W, ($\{0,1\}^W$) is mapped, of course, to itself. Thus, ϕ satisfies property (ii) and is a hierarchical code, i.e. $\phi \in \Sigma^H$.

Given a hierarchical code $\sigma^H \in \Sigma^H$, define $g^H(W)$ and $f_p^H(W)$ for this code as $g(W)$ and $f_p(W)$ were defined for the binary code, respectively. Then we obtain:

Theorem 9: In any hierarchical code $\sigma^H \in \Sigma^H$, the extremal range expansion and prefix-based general range expansion have the same lower bounds as with a binary code:

$$g^H(W) \geq \left\lceil \frac{W+1}{2} \right\rceil = g(W), \tag{16}$$

$$f_p^H(W) \geq W = f_p(W). \tag{17}$$

Proof: We start by proving the first part of the theorem. As explained earlier in this paper, it is enough to prove the theorem when W is even. The proof is by induction on W and follows the proof of Theorem 6. For each W, we exhibit an extremal range R^{LE} and an alternating path of size $\left\lceil \frac{W+1}{2} \right\rceil$ and show that given any hierarchical code $\sigma^H \in \Sigma^H$, R^{LE} cannot be encoded in less than $\left\lceil \frac{W+1}{2} \right\rceil$ TCAM entries. To do so, we use the notation $c(W) = \frac{2}{3}\left(2^W - 1\right)$ from Theorem 6 and consider the extremal range $R^{LE} = [0, c(W)]$. Let α be the indicator function of R^{LE}, and for a bit value b let b' be the bit value $1 - b$.

Induction basis: We start with the case of $W = 2$. Here $R = [0, c(2)] = [0, 2]$. Without loss of generality, the code σ^H is of the form: $\sigma(00) = (b_1, b_2), \sigma(01) = (b_1, b_2'), \sigma(10) = (b_1', b_3), \sigma(11) = (b_1', b_3')$. Here $\alpha((b_1, b_2)) = \alpha((b_1, b_2')) = \alpha((b_1', b_3)) = 1, \alpha((b_1', b_3')) = 0$. There are two possible cases: If $b_2 = b_3$, we look at $\sigma(01), \sigma(10)$ and $\sigma(11)$. We define $A_2 = (a^1, a^2, a^3)$, for $a^1 = \sigma(01) = (b_1, b_2')$, $a^2 = \sigma(11) = (b_1', b_3')$ and $a^3 = \sigma(10) = (b_1', b_3)$. Then $A_2 = (a^1, a^2, a^3)$ is an *alternating path of size* 2 and $\{a^1, a^3\}$ is an independent set, because it satisfies the two needed conditions:
(i) Alternation: $a^1 \in R^{LE}$, $a^2 \notin R^{LE}$, $a^3 \in R^{LE}$.
(ii) Hull: $a^2 = (b_1', b_3') = (b_1', b_2') \in H((b_1, b_2'), (b_1', b_3)) = H(a^1, a^3)$, because a^2 shares its first bit with a^3 and its second bit with a^1.

If $b_2 \neq b_3$, we look at $\sigma(00), \sigma(10)$ and $\sigma(11)$. We define $A_2 = (a^1, a^2, a^3)$, for $a^1 = \sigma(00) = (b_1, b_2)$, $a^2 = \sigma(11) = (b_1', b_3')$ and $a^3 = \sigma(10) = (b_1', b_3)$. Then, $A_2 = (a^1, a^2, a^3)$ is again an *alternating path of size* 2.

Induction step: For a general even value of W, we have $R^{LE}(W) = [0, c(W)] = [0, 2^{W-1} + c(W-2)]$. We can see that $R^{LE}(W) = [0, 2^{W-1} - 1] \cup [2^{W-1}, 2^{W-1} + c(W-2)] = R^1 \cup R^2$, where R^2 is a shifted version of $R^{LE}(W-2)$, and observe that $2^{W-1} < c(W) < (2^{W-1} + 2^{W-2})$. By property (ii) of the hierarchical code σ^H, we must have that the first two bits in the W-bit string of the code of the points in $[2^{W-1}, 2^{W-1} + c(W-2)] \subseteq [2^{W-1}, 2^{W-1} + 2^{W-2} - 1]$ are equal. Further, if we denote them by (b_1, b_2), then all the points with code that starts with the first two bits (b_1, b_2') belong to $[2^{W-1} + 2^{W-2}, 2^W - 1] \subseteq (R^{LE}(W))^c$. Further, all the points with code that starts with the first bit $\{b_1'\}$ belong to $[0, 2^{W-1} - 1] \subseteq R^{LE}(W)$. Let $(a^1, \ldots, a^{W-1})$ with $a^i = (a_1^i, a_2^i, \ldots, a_W^i)$ be the alternating path for $R^{LE}(W-2)$. We build the alternating path $(b^1, \ldots, b^{W+1})$ as follows: We first define $b^i = (b_1, b_2)a^i$ for $i \in [1, W-1]$. We get the next point by flipping the second bit of b^{W-1} to have $b^W = (b_1, b_2', a_1^{W-1}, a_2^{W-1}, \ldots, a_W^{W-1})$. To get the last point we flip in addition the first bit, $b^{W+1} = (b_1', b_2', a_1^{W-1}, a_2^{W-1}, \ldots, a_W^{W-1})$.

By the last observations we can see that $(b^1, \ldots, b^{W+1})$ is an *alternating path of size* $\frac{W}{2} + 1$. In the full proof, presented in [34] for space reasons, it satisfies the *Alternation* condition as well as the *Hull* condition.

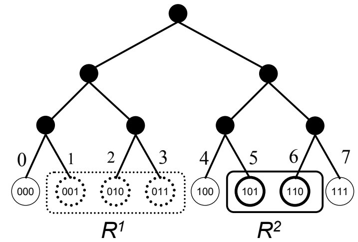
We can now deduce that $(b^1, \ldots, b^{W+1})$ is an alternating path of size $\frac{W}{2} + 1$ and by Theorem 5 we have the result.

The proof of the second part of the theorem is similar to the proof of Theorem 7. For instance, if W is odd we consider again the W-bit range $R = \left[\frac{1}{3}\left(2^{W-1} - 1\right), 2^{W-1} + \frac{1}{3} \cdot 2^{W-2} - \frac{2}{3}\right] \subseteq [0, 2^W - 1]$. In the full proof, we show that by the result of the first part, using only prefix encoding, a total number of W TCAM entries are required for the encoding of the range R, also in this general hierarchical code. ■

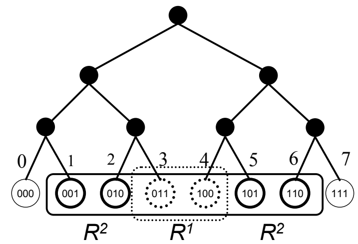
VII. Union of Ranges

We have shown that any range can be encoded using $f_p(W) = W$ entries. However, it is not straightforward that encoding k ranges would also be possible in roughly kW ranges. For instance, if we encode some range R^1 using external encoding, i.e. by first encoding its complementary $(R^1)^c$, we might encompass another range $R^2 \subseteq (R^1)^c$, and therefore yield a wrong encoding. A simple apparent solution is to encode R^2 first, but then we might need to encode it using its complementary $(R^2)^c$ first. This is again a problem because $R^1 \subseteq (R^2)^c$. Here is a simple example of such a phenomenon.

Example 6: Assume we want to encode $k = 2$ ranges of $W = 4$-bit strings. Let $R^1 = [0, 11] = \{\{0000\}, \ldots, \{1011\}\}$ and $R^2 = [15, 15] = \{1111\}$. We want to encode $R^1 \cup R^2$. Then R^1 can be encoded as $(11{*}{*} \to 0, {*}{*}{*}{*} \to 1)$, neglecting the last default entry. Likewise, R^2 can be encoded as $(1111 \to 1)$. However, directly combining the entries would yield $(11{*}{*} \to 0, {*}{*}{*}{*} \to 1, 1111 \to 1)$, which actually encodes R^1 and not $R^1 \cup R^2$. Instead, a correct encoding would have been $(1111 \to 1, 11{*}{*} \to 0, {*}{*}{*}{*} \to 1)$.



(a)



(b)

Fig. 3. Union of two ranges. Fig. 3(a) presents a union of two non-adjacent ranges that yields two non-default intervals. Fig. 3(b) presents a union of two adjacent ranges that yields three non-default intervals, each of them with the same resulting action.

The example shows there might be a problem when the encoding of a range defines treatment for values that appear outside it but inside other ranges. Note that this problem does not occur when the ranges are in different halves of the W-bit range, since we can rely on prefix-based encoding. Further, it does not occur if one of the two ranges is included in a prefix sub-space and the second does not intersect it. In such a case, the prefix of that sub-space may be used to avoid a detrimental effect of the first encoding on the second, so we can first encode the first range and later the second.

We want to generalize Theorem 4, which states that any single range can be encoded in W entries, not including the default one. We will consider a set of k distinct ranges, defined in the same way as [30]. Namely, by a range, we mean a non-default interval with the same resulting action. Therefore, although two non-adjacent ranges can cut the space of all 2^W elements into five intervals (successively corresponding to default, then first range, then default, then second range, and default again), we consider these as two ranges only. On the other hand, if a first rule is strictly contained within a second rule but has priority, then the two rules create three ranges (successively corresponding to the second, first, and again second rule).

Example 7: For $W = 3$, we consider the case of two ranges R^1, R^2 defined with corresponding actions 'accept' and 'log'. Let 'deny' be the default action. As shown in Fig. 3(a), the range $R = R^1 \cup R^2 = [1, 3] \cup [5, 6]$ is considered as two ranges only since R^1, R^2 are non-adjacent ranges. However, as shown in Fig. 3(b), if $R = R^1 \cup R^2 = [3, 4] \cup [1, 6]$. R^1 has priority over R^2 and R^1 is strictly contained within R^2, there are three ranges.

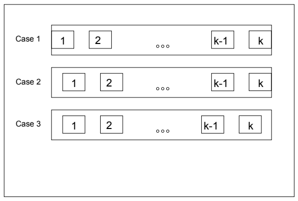


Fig. 4. Three types of unions of k ranges defined in Definition 15.

The next theorem follows directly from a tighter result presented below in Theorem 11. To prove Theorem 11, we will first establish several lemmas based on the different types of range unions.

Theorem 10: For all $W \in \mathbb{N}^*$ and $k \in \mathbb{N}$, any k ranges of W-bit elements can be encoded using prefix encoding in at most $k(W+1)$ TCAM entries.

For any range R^i, we define its *bit size* W^i as its number of meaningful bits, i.e the number of bits that can vary in the string representation of the elements of R^i, corresponding to the maximum possible suffix distance within R^i. As usual, we refer to W as the total number of bits in the definition in each of the ranges, i.e. they are all defined over a sub-space of size 2^W. From now on, we assume that each range R^i is defined with a corresponding action a^i.

In the following lemma, we give an upper bound on the expansion of the union of two distinct ranges.

Lemma 4: Let R^1 and R^2 denote two ranges of respective bit sizes W^1 and W^2. We consider their union $R^1 \cup R^2$.
1. If R^1, R^2 are both extremal ranges, their union can be encoded in at most $g(W^1) + g(W^2)$ TCAM entries.
2. If only R^2 is an extremal range, their union can be encoded in at most $f(W^1) + g(W^2) + 1$ TCAM entries.
3. If both R^1 and R^2 are not extremal ranges, their union can be encoded in at most $f(W^1) + f(W^2) + 2$ TCAM entries.
All results can rely on prefix encoding.

For space reasons, the tedious proof of this lemma is not brought here and is presented in [34].

We now provide a definition that defines three possible types of a union of k disjoint ranges. It is illustrated in Fig. 4.

Definition 15 (Range-union types): Let a general *k-union* of disjoint ranges denote a union of the form $\bigcup_{i=1}^{k} R^i = \bigcup_{i=1}^{k} [y_1^i, y_2^i]$, where $(\forall i \in [1,k])(y_1^i \le y_2^i) \wedge (\forall i \in [1, k-1])(y_2^i < y_1^{i+1})$ and R^i is assigned with an arbitrary action a^i.
a. Let a *k-union with two extremal ranges* denote a k-union where $y_1^1 = 0$ and $y_2^k = (2^W - 1)$.
b. Let a *k-union with an extremal range* denote a k-union where $y_1^1 \neq 0$ and $y_2^k = (2^W - 1)$.
c. Let a *k-union with no extremal ranges* denote a k-union where $y_1^1 \neq 0$ and $y_2^k \neq (2^W - 1)$.

We now present the main theorem of this section. In its proof, presented in [34] for space reasons, we consider the expansion of each the different unions of k disjoint ranges from Definition 15.

Theorem 11: For all $W \in \mathbb{N}^*$ and $k \in \mathbb{N}$, any k ranges $\{R^i\}_{1 \le i \le k}$ of W-bit elements can be encoded in at most $\sum_{i=1}^{k} (W^i + 1)$ TCAM entries.

Last, we prove the asymptotic optimality of this theorem.

(a) (b)

Fig. 5. Two-dimensional range $R = (R_1, R_2)$. Fig. 5(a) presents the encoding of the two-dimensional range R using an internal encoding, yielding an expansion which is exponential in the number of fields. Fig. 5(b) demonstrates the encoding of R using the suggested encoding which yields a linear expansion.

Theorem 12 (Asymptotic Optimality): In the general case, as $k \to \infty$,
(i) any k ranges of W-bit strings can be encoded in at most $k \cdot (W - \log k + o(\log k))$ TCAM entries.
(ii) there are k ranges of W-bit strings that cannot be encoded using prefix encoding in less than $k \cdot (W - \log k + o(\log k))$ TCAM entries.

Proof: First, let's prove (i). Given three distinct ranges, at most one can have W significant bits, and at most two can have $W-1$ significant bits, because at most one range can cross the cut in the middle of the element space. Likewise, given seven ranges, at most one can have W significant bits, two can have $W-1$ significant bits, and four can have $W-2$ significant bits. Summing up for the case of k elements, we have $\sum_{i=1}^{k} (W^i + 1) \le k + kW - \sum_{i=1}^{l} i \cdot 2^i$, for l such that $k = \sum_{i=0}^{l} 2^i = 2^{l+1} - 1$, i.e. $l = \log(k+1) - 1$. Using the formula (for $r \neq 1$), $\sum_{i=1}^{n} i \cdot r^i = \frac{r - r^{n+2}}{(1-r)^2} - \frac{(n+1)r^{n+1}}{1-r}$, this upper bound equals $k + kW - ((2 - 2^{l+2}) + ((l+1) \cdot 2^{l+1})) = k + kW - (2 + (l-1) \cdot 2^{(l+1)}) = k \cdot (W + 1 - \frac{(l-1) \cdot 2^{(l+1)} + 2}{k})$. Using the value of l, we have $k \cdot (W + 1 - \frac{(\log(k+1) - 2)(k+1) + 2}{k}) = k \cdot (W + 3 - \frac{k \cdot \log(k+1) + \log(k+1)}{k}) = k \cdot (W + 3 - \log(k+1) - \frac{\log(k+1)}{k}) = k \cdot (W - \log(k) - o(\log(k)))$. We can see that k elements can get a maximum average number of significant bits that decreases as $\log k \cdot (1 + o(1))$.

To prove (ii), when $k = 2^l$, we cut the space of 2^W into k sub-spaces of 2^{W-l} elements each, and apply Theorem 7 on each subspace to get a lower-bound on the encoding of each range in $\log(2^{W-l}) = W - l = W - \log k$ entries, with a total number of $k \cdot (W - \log k)$ entries. Using prefix encoding, we can see that the ranges cannot be encoded together. ■
Due to this relatively complicated encoding of a union of ranges, the rule update might be complicated. We leave this issue for future work.

VIII. Multidimensional ranges

A. Exponential Number of TCAM Entries

Our objective is to find an encoding scheme of a classification rule $R = ((R_1, \ldots, R_d) \to a)$ defined over d fields, given that we already have some encoding schemes for each range rule R_i in field (dimension) F_i, where $i \in [1, d]$. While the result is well-known when using internal Binary-Prefix encoding, the following theorem deals with any encoding, including external encoding that starts with encoding the complimentary range.

Theorem 13: Given a classification rule $R = ((R_1, \ldots, R_d) \to a)$ and d encoding schemes $\{\phi_i\}_{i=1}^d$ of the ranges $\{R_i\}_{i=1}^d$ with expansions $\{n_{\phi_1}(R_1), \ldots, n_{\phi_d}(R_d)\}$,
(i) R can be encoded in at most $n = \prod_{i=1}^d n_{\phi_i}(R_i)$ TCAM entries;
(ii) in particular, given $d' \leq d$ fields with range rules, R can always be encoded in $W^{d'}$ TCAM entries.

For space reasons, the proof of the first part of the theorem is not brought here and is presented in [34]. The second part directly follows Theorem 3 and the first part.

Example 8: Consider a general range of two W-bit fields $R = (R_1, R_2)$, presented in Fig. 5(a). For $i = 1, 2$ let r'_i be the expansion of R_i using our improved encoding scheme. R can be encoded in $r'_1 \cdot r'_2 \leq f(W)^2 \leq W^2$ TCAM entries.

B. Linear Number of TCAM Entries

The main drawback of encoding a hyper-rectangle with d dimensions is the curse of dimensionality, i.e. the typical exponential dependency in the number of fields d. We show here how to encode a hyper rectangle with a linear dependency in d.

Example 9: Consider again the range R from Example 8. As illustrated in Fig. 5(b), we can first negatively encode the four striped regions, using an internal encoding of the corresponding four one-dimensional extremal intervals (using at most $4W$ entries [19]), and then add a default positive entry (using one entry), thus yielding a linear expansion upper-bound of $4W+1$. More generally, we get the following tighter upper-bound:

Theorem 14: Any classification rule R of d fields can be encoded in at most $d \cdot (2W - 2) + 1$ TCAM entries without any additional logic.

Proof: We remind that an extremal W-bit range R can be internally encoded in at most W TCAM entries [19]. Further, if the extremal W-bit range R is not a shifted version of one of the ranges $[0, 2^W - 2]$, $[1, 2^W - 1]$ then R can be internally encoded in at most $W - 1$ TCAM entries.

For a general d-dimensional rule R, we assume that its i-th dimension range is $R^i = [a, b]$. We also assume that $a \neq b$. Otherwise, R^i is an exact match and its encoding does not require any additional TCAM entries besides the encodings of the other dimensions. We define $R^{LE} = [0, a-1]$ and $R^{RE} = [b+1, 2^W - 1]$ such that $R^{LE} \bigcup R^i \bigcup R^{RE} = [0, 2^W - 1]$. We want to show that we can internally encode the extremal ranges R^{LE}, R^{RE} in a total number of $2W - 2$ TCAM entries. We consider 3 possible cases:

(i) If $a \leq 2^{W-1} - 1$ and $b \geq 2^{W-1}$, i.e. $R^{LE} \subseteq [0, 2^{W-1} - 1], R^{RE} \subseteq [2^{W-1}, 2^W - 1]$, then R^{LE}, R^{RE} are $(W-1)$-bit ranges and therefore each of them can be encoded using internal encoding in at most $W - 1$ TCAM entries.

(ii) Else if $b < 2^{W-1}$, i.e. $R^i \subseteq [0, 2^{W-1} - 1]$, then the $(W-1)$-bit range R^{LE} holds $R^{LE} \neq [0, 2^{W-1} - 2]$ and can be encoded in at most $W - 2$ TCAM entries. By internally encoding R^{RE} in at most W TCAM entries, we have a total number of at most $2W - 2$ TCAM entries.

(iii) Else $a > 2^{W-1} - 1$ and $R^i \subseteq [2^{W-1}, 2^W - 1]$. We internally encode $R^{RE} \neq [2^{W-1}+1, 2^W - 1]$ in at most $W-2$ TCAM entries and R^{LE} in at most W TCAM entries, having a total number of at most $2W - 2$ TCAM entries.

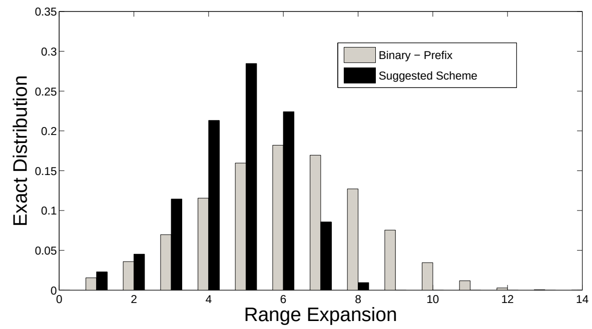


Fig. 6. Range expansion distribution over all possible ranges for $W = 8$.

Therefore, after encoding negatively the range complimentary for each field, we add the last default entry with a positive action and have at most $d \cdot (2W - 2) + 1$ TCAM entries. ■

C. Linear Number of TCAM Entries with Additional Logic

The above results for multidimensional ranges assume that the classifier has only one classification rule. In the Appendix we suggest hardware changes that enable us to efficiently encode $k > 1$ classification rules, as stated in the following theorem.

Theorem 15: Let $C = (R^1, ..., R^k)$ be a classifier with k classification rules defined over d fields. Using additional logic, C can be encoded in at most $k \cdot d \cdot W$ TCAM entries.

IX. Experimental Results

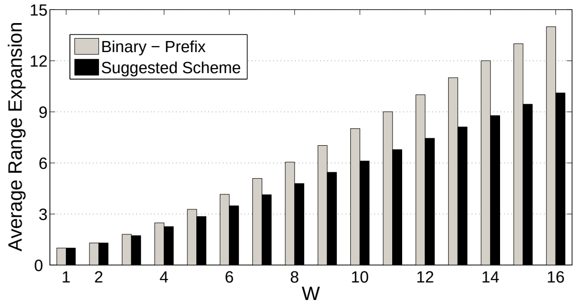
In this section, we evaluate the effectiveness of our approach on both real-life and synthetic packet classifiers. We also compare the suggested scheme with known encoding algorithms and in particular SRGE [17].

A. Worst-Case Range Expansion (Theorem 3)

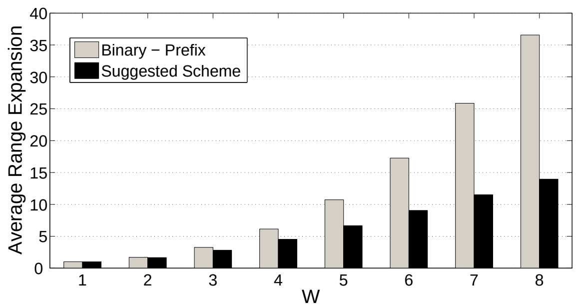
Figure 6 presents the range expansion distribution over all the ranges in $[0, 2^W - 1]$ with $W = 8$ bits. The worst-case expansion of the well-known internal Binary-Prefix approach is $2W - 2 = 14$ (with negligible probability), while it is $W = 8$ in our suggested scheme, thus confirming Theorem 3.

B. Average Range Expansion

Figure 7 shows the average range expansion in the Binary-Prefix encoding and in our suggested scheme. First, Fig. 7(a) shows for $W \in [1, 16]$ the average expansion of a uniformly-distributed range in $[0, 2^W - 1]$. For instance, for $W = 2$ the average expansion is calculated among the 10 different ranges in $[0, 3]$. In both schemes the average expansion is 1.3 since in both of them only the ranges $[0, 2], [1, 2], [1, 3]$ require two entries while the other seven ranges can be encoded in one entry. For $W = 16$, the average Binary-Prefix expansion is approximately 13.00 and the improved expansion is only 9.44. The average expansion for $W = 8$ can be deduced from the range expansion distribution in Figure 6. Likewise, Fig. 7(b) shows for $W \in [1, 8]$ the average range expansion



(a)



(b)

Fig. 7. Average expansion of ranges. Fig. 7(a) presents the average range expansion for a one-dimensional range in $[0, 2^W - 1]$ for $W \in [1, 16]$. Fig. 7(b) presents the average range expansion for a two-dimensional range in $[0, 2^W - 1] \times [0, 2^W - 1]$ for $W \in [1, 8]$.

of a two-dimensional range in $[0, 2^W - 1] \times [0, 2^W - 1]$. We encode any given range in the most efficient scheme among the two suggested schemes presented in Section VIII-A and in Section VIII-B. For example, for $W = 8$, the average Binary-Prefix expansion is 36.56 and using our scheme it is reduced to only 13.96.

X. Conclusion

This paper is unique in that it deals with the fundamental capacity region of TCAMs. In the paper, we presented new upper-bounds on the TCAM worst-case rule expansions. In particular, we proved that a W-bit range can be encoded in W TCAM entries using prefix encoding, improving upon the previously-known bound of $2W - 5$. We also introduced fundamental analytical tools based on independent sets and alternating paths, and used these tools to prove the tightness of the upper bounds. We then showed that the developed expansion lower bounds hold in any hierarchical code and provided asymptotically optimal upper bounds for the encoding of a union of ranges.

In addition, we suggested several modified TCAM architectures that provide clear tradeoffs between better range expansion guarantees with less TCAM active entries and more complex logic within the TCAM. Last, we showed that it is possible to encode ranges using a number of TCAM entries that increases only *linearly* instead of *exponentially* with the number of fields.

Acknowledgment

We would like to acknowledge Anat Bremler-Barr for kindly accepting to run several simulations. We would also like to thank David Hay, Danny Hendler, Ran Ginosar, Zigi Walter and Tuvi Etzion for their helpful suggestions.

This work was partly supported by the European Research Council Starting Grant No. 210389, by the Jacobs-Qualcomm fellowship, by an Intel graduate fellowship, by a Gutwirth Memorial fellowship, by an Intel research grant on Heterogeneous Computing, and by the Hasso Plattner Center for Scalable Computing.

References

[1] O. Rottenstreich and I. Keslassy, "Worst-Case TCAM Rule Expansion," in *IEEE INFOCOM Mini-Conference*, 2010.

[2] R. Cohen and D. Raz, "Simple efficient TCAM based range classification," in *IEEE INFOCOM Mini-Conference*, 2010.

[3] O. Rottenstreich and I. Keslassy, "On the Code Length of TCAM Coding Schemes," in *IEEE ISIT*, 2010.

[4] D. E. Taylor, "Survey and taxonomy of packet classification techniques," *ACM Comput. Surv.*, vol. 37, no. 3, pp. 238-275, 2005.

[5] G. Varghese, *Network Algorithmics,* Morgan Kaufmann, 2005.

[6] J. Chao and B. Liu, *High Performance Switches and Routers,* Wiley, 2007.

[7] J. Naous *et al.*, "Implementing an OpenFlow Switch on the NetFPGA platform," in *ACM ANCS*, 2008.

[8] NetLogic Microsystems [Online]. Available: www.netlogicmicro.com/

[9] Renesas [Online]. Available: www.renesas.com/

[10] P. Gupta and N. McKeown, "Packet classification on multiple fields," in *ACM SIGCOMM*, 1999.

[11] S. Singh *et al.*, "Packet classification using multidimensional cutting," in *ACM SIGCOMM*, 2003.

[12] H. Song, J. Turner and S. Dharmapurikar, "Fast packet classification using Bloom filters," in *ACM ANCS*, 2006.

[13] H. Song, J. Turner and S. Dharmapurikar, "Packet classification using coarse-grained tuple spaces," in *ACM ANCS*, 2006.

[14] K. Pagiamtzis and A. Sheikholeslami, "Content-Addressable Memory (CAM) circuits and architectures: A tutorial and survey, *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 712-727, 2006.

[15] C. R. Meiners, A. X. Liu and E. Torng, "Bit weaving: A non-prefix approach to compressing packet classifiers in TCAMs," in *IEEE ICNP*, 2009.

[16] J. Baliga *et al.*, Photonic switching and the energy bottleneck. in *Photonics in Switching*, 2007.

[17] A. Bremler-Barr and D. Hendler, "Space-efficient TCAM-based classification Using Gray coding," *IEEE Trans. on Computers*, vol. 61, no. 1, pp. 18-30, 2012.

[18] K. Lakshminarayanan, A. Rangarajan and S. Venkatachary, "Algorithms for advanced packet classification with ternary CAMs," in *ACM SIGCOMM*, 2005.

[19] V. Srinivasan *et al.*, "Fast and scalable layer four switching," in *ACM SIGCOMM*, 1998.

[20] B. Schieber, D. Geist and A. Zaks, "Computing the minimum DNF representation of Boolean functions defined by intervals," *Discrete Applied Mathematics*, vol. 149, no. 1-3, pp. 154-173, 2005.

[21] R. Roth. Personal communication.

[22] C. R. Meiners, A. X. Liu and E. Torng, "TCAM Razor: A systematic approach towards minimizing packet classifiers in TCAMs," in *IEEE ICNP*, 2007.

[23] E. Spitznagel, D. E. Taylor and J. S. Turner, "Packet classification using extended TCAMs" in *IEEE ICNP*, 2003.

[24] H. Hwang *et al.*, "Minimization of ACL storage by adding minimal hardware of range matching and logical gates to TCAM," in *IEEE HPSR*, 2008.

[25] H. Che *et al.*, "DRES: Dynamic range encoding scheme for TCAM Coprocessors," *IEEE Trans. on Computers*, vol. 57, no. 7, pp. 902-915, 2008.

[26] C. R. Meiners *et al.*, "Topological transformation approaches to optimizing TCAM-based packet processing systems," in *ACM SIGMETRICS*, 2009.

[27] A. X. Liu *et al.*, "All-match based complete redundancy removal for packet classifiers in TCAMs," in *IEEE INFOCOM*, 2008.

[28] A. X. Liu and M. G. Gouda, ”Complete redundancy removal for packet classifiers in TCAMs,” *IEEE Trans. on Parallel Distrib. Syst.*, vol. 21, no. 4, pp. 424-437, 2010.
[29] A. Bremler-Barr et al., ”Layered interval codes for TCAM based classification,” in *IEEE INFOCOM*, 2009.
[30] S. Suri, T. Sandholm and P. R. Warkhede, ”Compressing two-dimensional routing tables,” Springer Algorithmica, vol. 35, no. 4, pp. 287-300, 2003.
[31] T. Sasao, ”On the complexity of classification functions,” in *IEEE ISMVL*, 2008.
[32] D. E. Taylor and J. S. Turner, ”ClassBench: A packet classification benchmark,” in *IEEE INFOCOM*, 2005.
[33] A. Bremler-Barr et al., ”PEDS: A parallel error detection scheme for TCAM devices,” *IEEE/ACM Trans. on Networking*, vol. 18, no. 5, pp. 1665-1675, 2010.
[34] O. Rottenstreich and I. Keslassy, ”Worst-case TCAM rule expansion,” Technical Report TR09-01, Comnet, Technion, Israel. http://webee.technion.ac.il/ isaac/papers.html.

Ori Rottenstreich received the B.S. degree in computer engineering from the Electrical Engineering department of the Technion, Haifa, Israel in 2008. He is now pursuing a Ph.D. degree at the same department. His current research interests include exploring novel coding techniques for networking applications. He is the recipient of the Jacobs-Qualcomm fellowship, the Intel graduate fellowship and the Gutwirth Memorial fellowship.

Rami Cohen received the Ph.D. and M.Sc. from the Technion, Israel Institute of Technology in 2007 and 2005 respectively. He has joined IBM Research Lab in Haifa, Israel in 2010. His research interests include network architectures, cloud computing, and optimization algorithms of network related problems.

Danny Raz received his doctoral degree from the Weizmann Institute of Science, Israel, in 1995. From 1995 to 1997 he was a post-doctoral fellow at the International Computer Science Institute, (ICSI) Berkeley, CA, and a visiting lecturer at the University of California, Berkeley. Between 1997 and 2001 he was a Member of Technical Staff at the Networking Research Laboratory at Bell Labs, Lucent Technologies. In October 2000, Danny Raz joined the faculty of the computer science department at the Technion, Israel. He served as the general chair of OpenArch 2000, TPC co-chair of MMNS 2007, TPC co-chair of IM2009, and was an Editor of the IEEE/ACM Transactions on Networking (ToN) and of JCN. His primary research interest is the theory and application of management related problems in IP networks and cloud computing.

Isaac Keslassy received the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University, Stanford, CA, in 2000 and 2004, respectively. He is currently an associate professor in the Electrical Engineering department of the Technion, Haifa, Israel. His recent research interests include the design and analysis of high-performance routers and on-chip networks. The recipient of the ERC Starting Grant, the Yigal Alon Fellowship and the ATS-WD Career Development Chair, he is a senior member of the IEEE and an editor for the IEEE/ACM Transactions on Networking.