ADVANCED DATA STRUCTURES IN TELECOMMUNICATION

János Tapolcai, Rétvári Gábor, Körösi Attila

Budapest University of Technology and Economics (BME)
Faculty of Electrical Engineering and Informatics (VIK)
Department of Telecommunications and Media Informatics (TMIT)
High-Speed Networks Laboratory (HSN*Lab*)
MTA-BME Future Internet Research Group

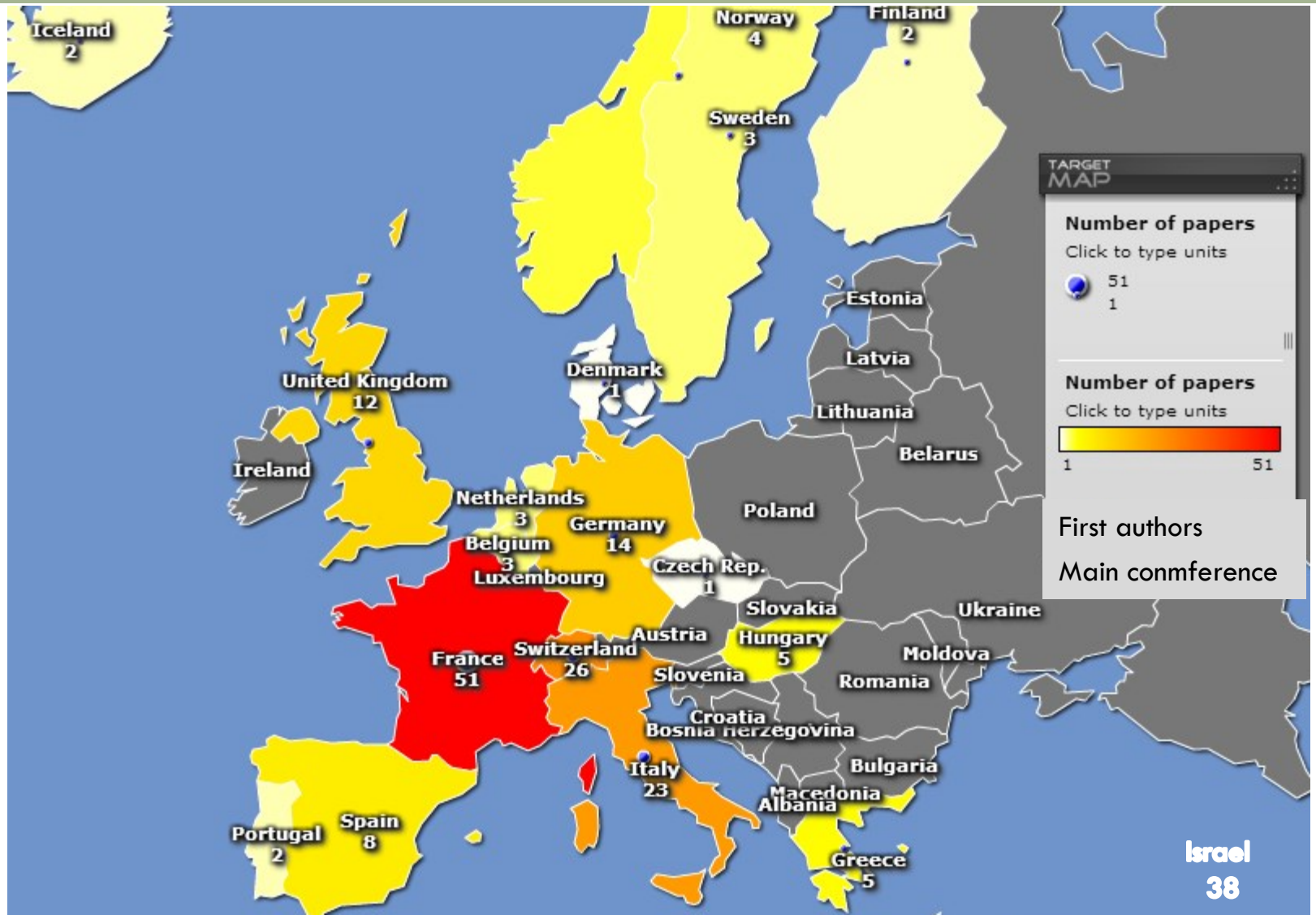
BME Modellalkotás szeminárium16/9/2014

MTA-BME Future Internet Research Group

Gábor Rétvári, András Gulyás, József Bíró, Zalán Heszberger, Péter Babarczi, Balázs Sonkoly, Felicián Németh, Attila Körösi, Biczók Gergely, Toka László, Csikor Levente

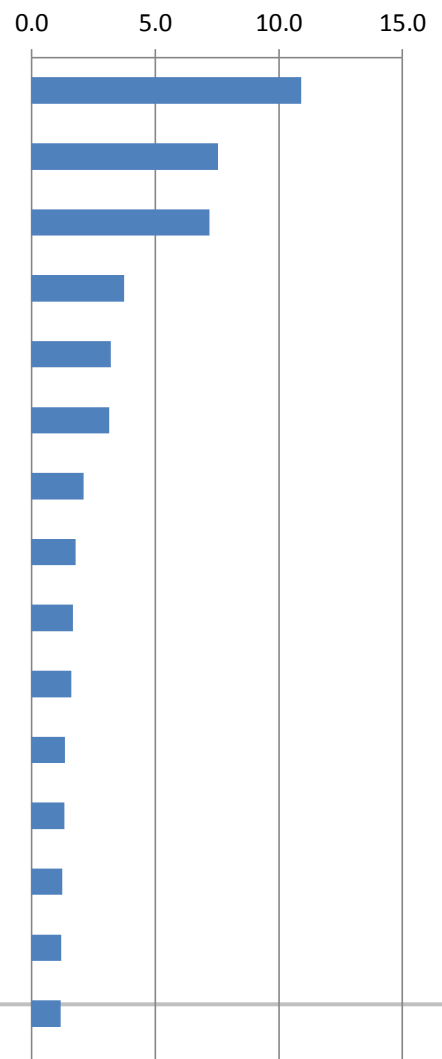
+ 8 PhD and >30 MSc students

IEEE Infocom papers 2009-2013

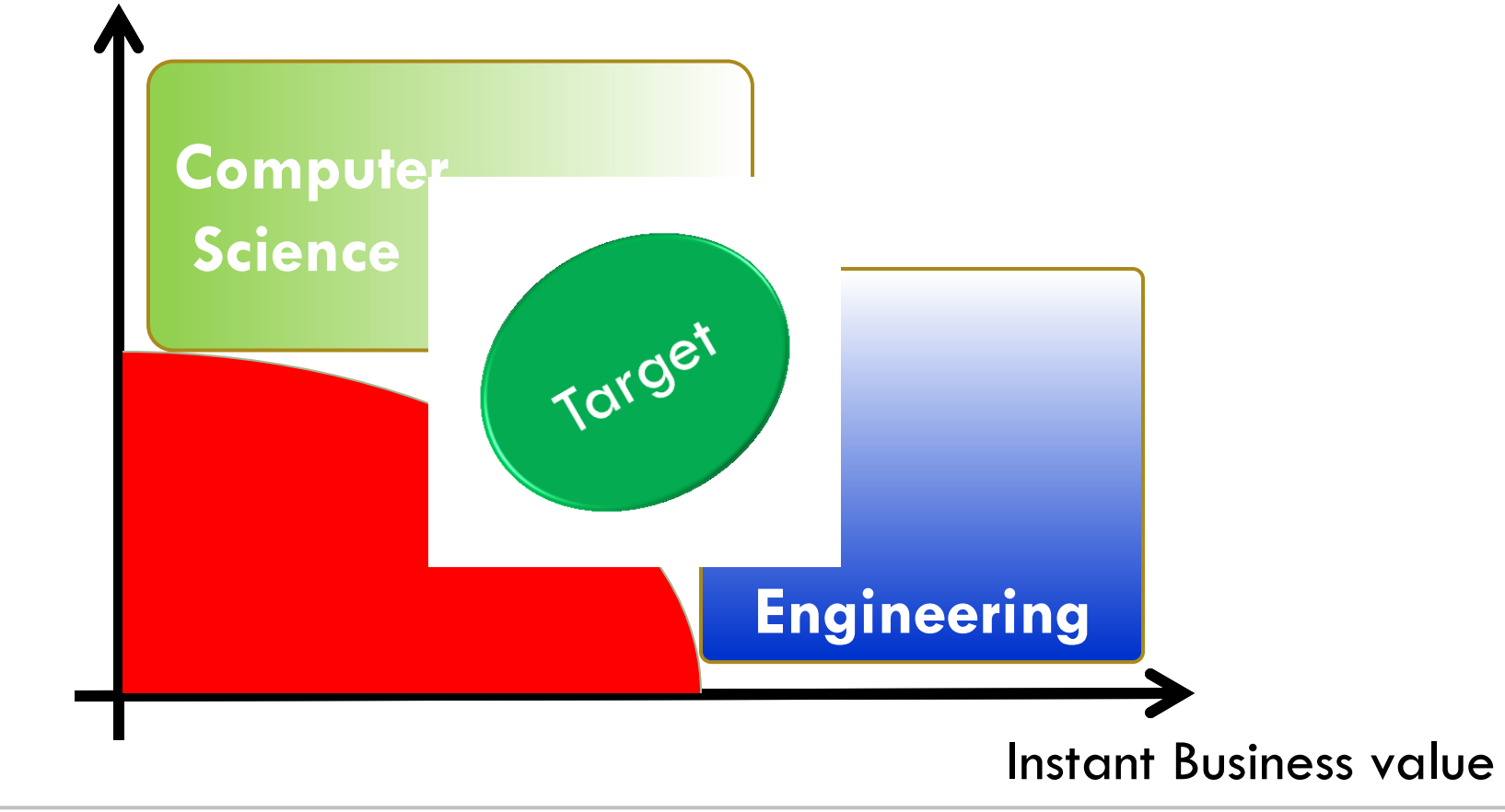


Weighted by the population (1999-2011)

Countries	#Citizents	#Papers	#Papers/million people
Israel	7521400	82	10,9
Hong Kong	7026400	53	7,5
Switzerland	7782900	56	7,2
Cyprus	801851	3	3,7
Singapore	4987600	16	3,2
Iceland	317630	1	3,1
Hungary	10013628	21	2,1
Taiwan	23119772	41	1,8
Finland	5356800	9	1,7
United States	308964000	496	1,6
France	65447374	88	1,3
Greece	11306183	15	1,3
United Kingdom	62041708	77	1,2
Italy	60276969	72	1,2
Sweden	9345135	11	1,2



Applied computer science



Main Research Topics

- Network design
  - Backbone – optical, Metro – IP, OpenFlow
- Reliability
  - „Your computer can go wrong, but the internet never.”
- Routing and addressing
  - Software Defined Networking
- Packet processing
  - Router design, Firewall rules

Packet processing

- G. Rétvári, J. Tapolcai, A. Kőrösi, A. Majdán Heszberger, "Compressing IP Forwarding Tables: Towards Entropy Bounds and Beyond", In ACM SigCOMM, 2013.
  - *Accepted to IEEE/ACM Transactions on Networking*

- A. Kőrösi, J. Tapolcai, B. Mihálka, G. Mészáros, G. Rétvári, "Compressing IP Forwarding Tables: Realizing Information-theoretical Space Bounds and Fast Lookups Simultaneously", In Proc. IEEE ICNP, 2014.

A Core Internet Router

- Holds info on the whereabouts of every single IP host
- That ought to be a huge amount of information

Cisco CRS-3 line card
up to 8 Gbyte memory
533 MHz DDR2
>300 Watt
http://www.cisco.com/en/US/docs/routers/

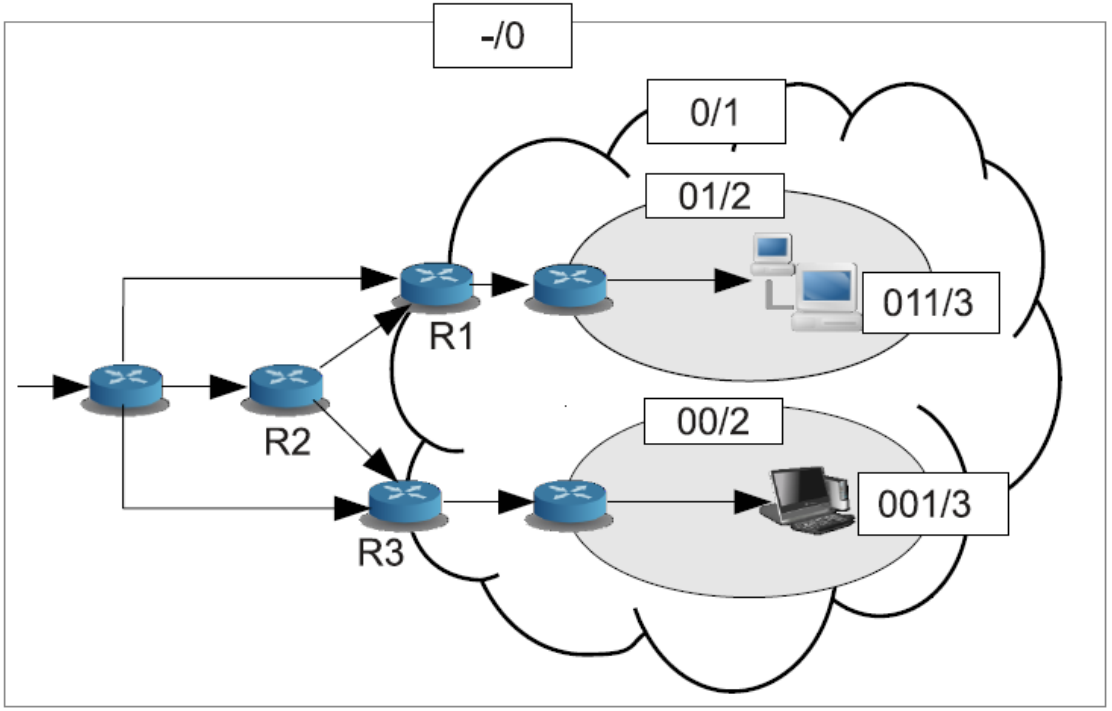
Internet Routing

- Packets are marked by 32 bit IP addresses

  weber.tmit.bme.hu=152.66.130.2

  =10011000010000101111101000110111 1

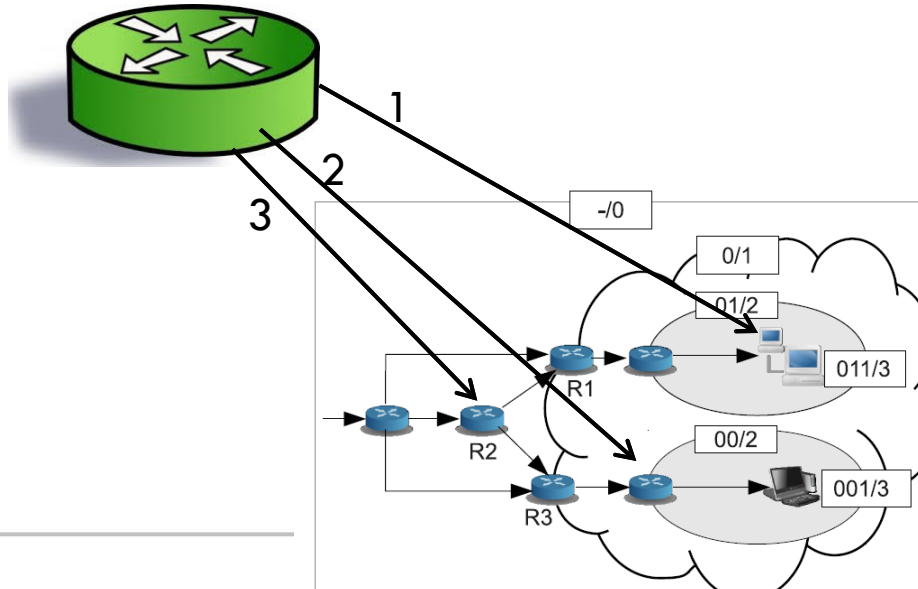
- Hierarchy of subnetworks by increasing prefix length



IP Forwarding Information Base

- The key data structure in Internet routing: tells a router where to forward a packet
- **FIB**: a database of prefix-to-next-hop associations
  - *lookup* a 32 bit long key: longest prefix match
  - *update* the association for some prefix

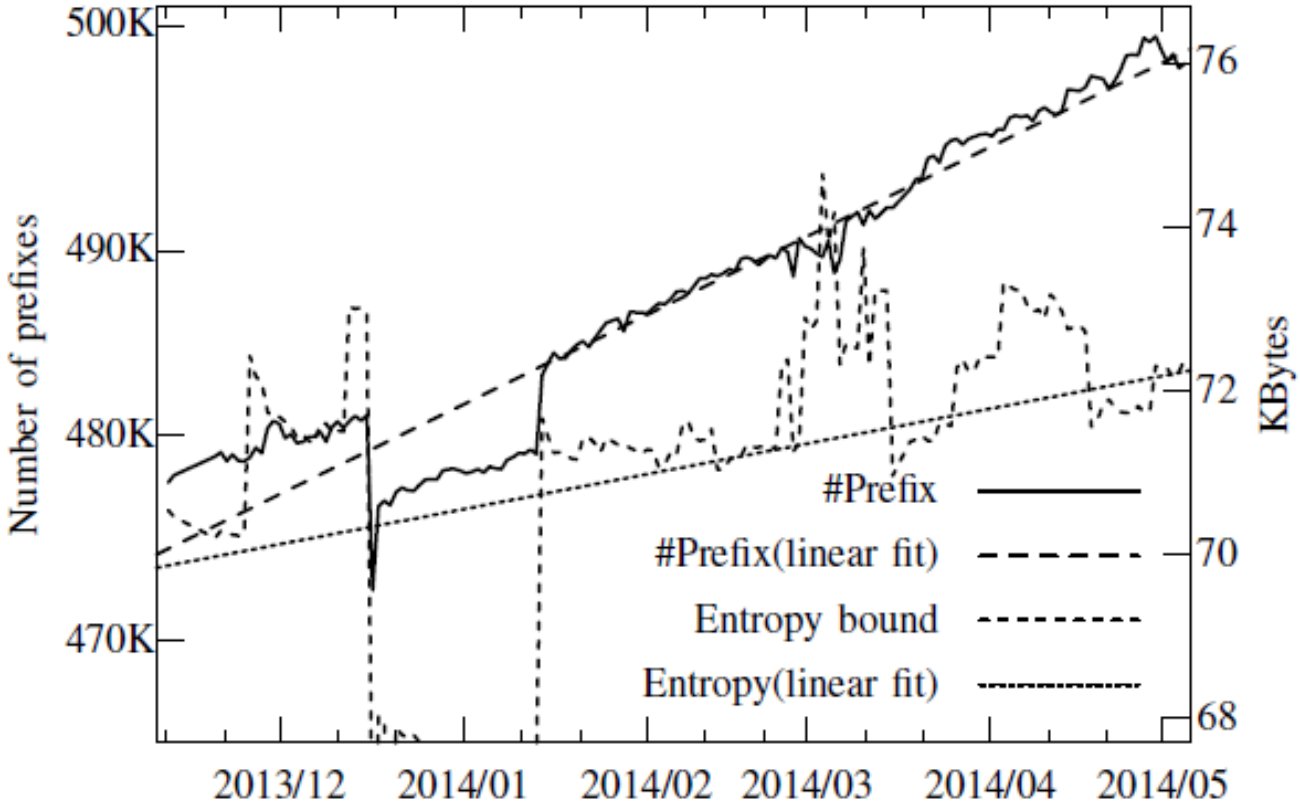
Address/prefix length	Label
-/0	2
0/1	3
00/2	3
001/3	2
01/2	2
011/3	1



IP Forwarding Information Base

- Stores more than 440K IP-prefix-to-nexthop mappings as of January, 2013
  - consulted on a packet-by-packet basis at line speed
  - queries are complex: longest prefix match
  - updated couple of hundred times per second
  - takes several MBytes of fast line card memory and counting
- May or may not become an Internet scalability barrier
  - http://lendulet.tmit.bme.hu/fib_comp/

Measurements

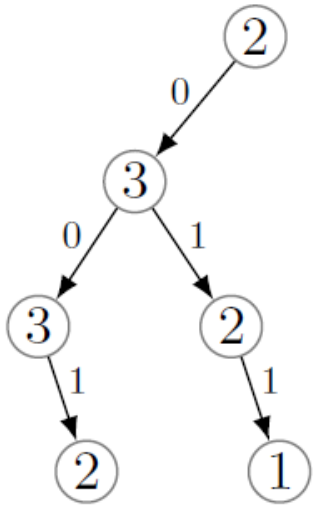


rtr.bme.hbone.hu

FIB Representations

- **Prefix tree:** search tree over the address space
- Essentially a labeled ordinal binary tree (binary trie)
- Lookup/ update are O(W) for W bit address size

Address/prefix length	Label
-/0	2
0/1	3
00/2	3
001/3	2
01/2	2
011/3	1

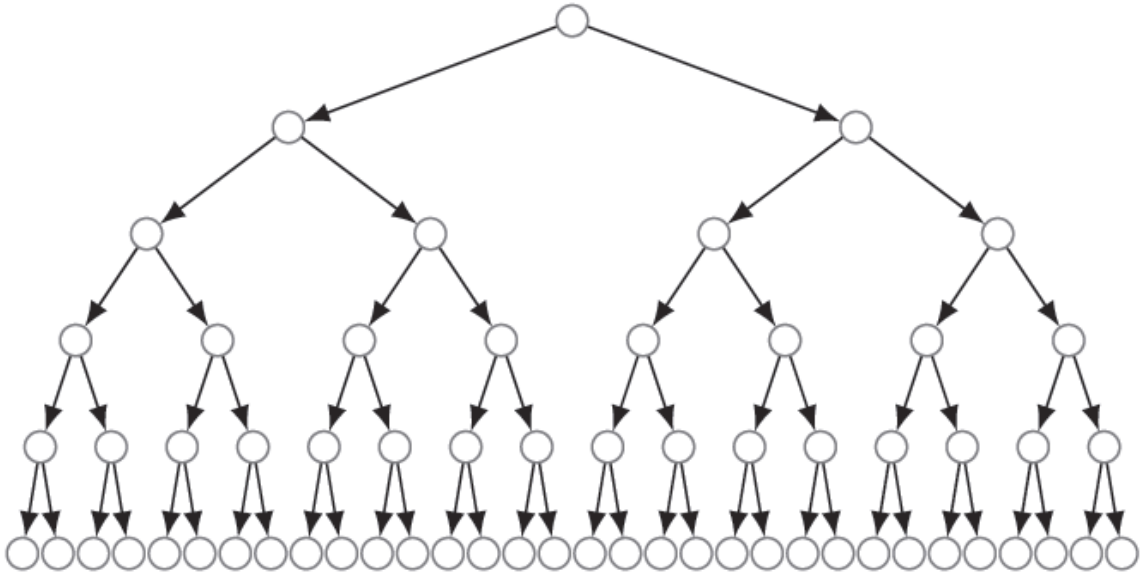


Compressed data structures

- Compression not necessarily sacrifices fast access!
- Store information in entropy-bounded space and provide fast in-place access to it
  - take advantage of regularity, if any, to compress data drifts closer to the CPU in the cache hierarchy operations are even faster than on the original uncompressed form
- No space-time trade-off!
- Goal: advocate compressed data structures to the networking community
  - IP forwarding table compression as a use case

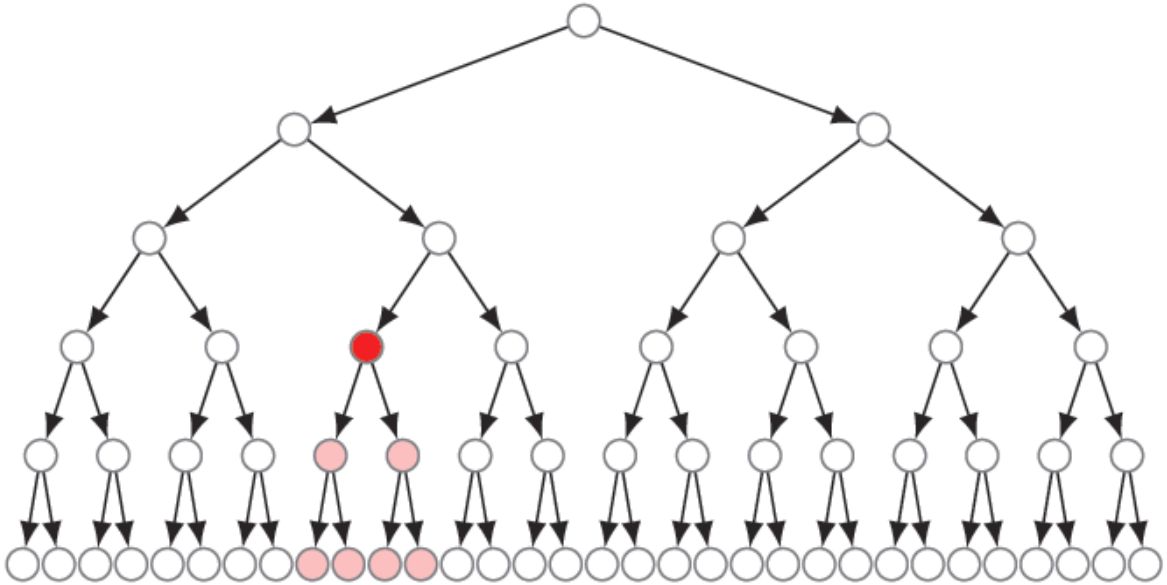
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



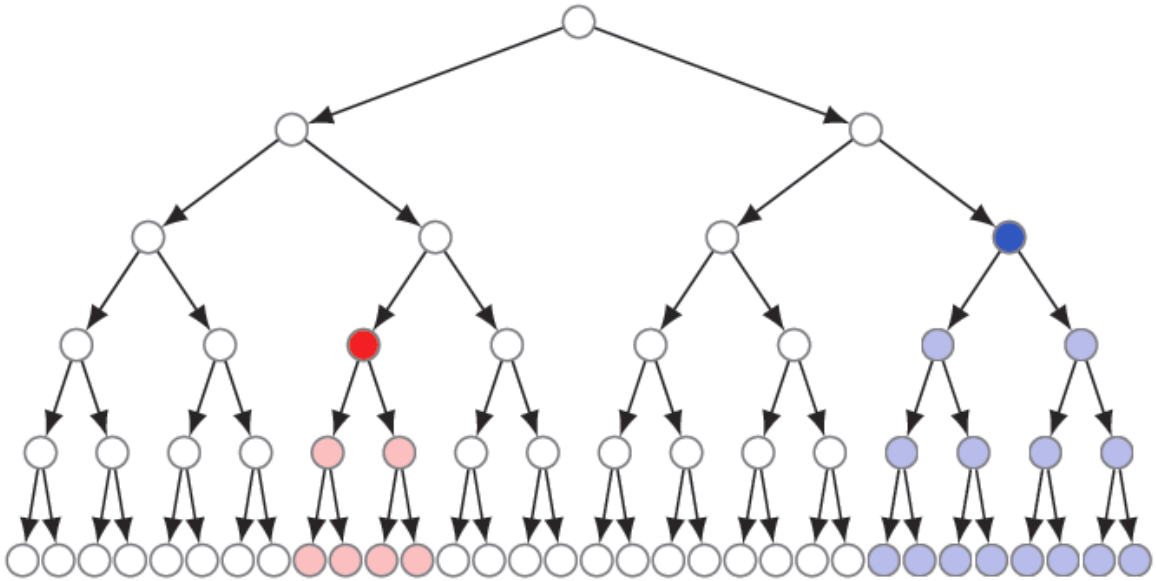
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



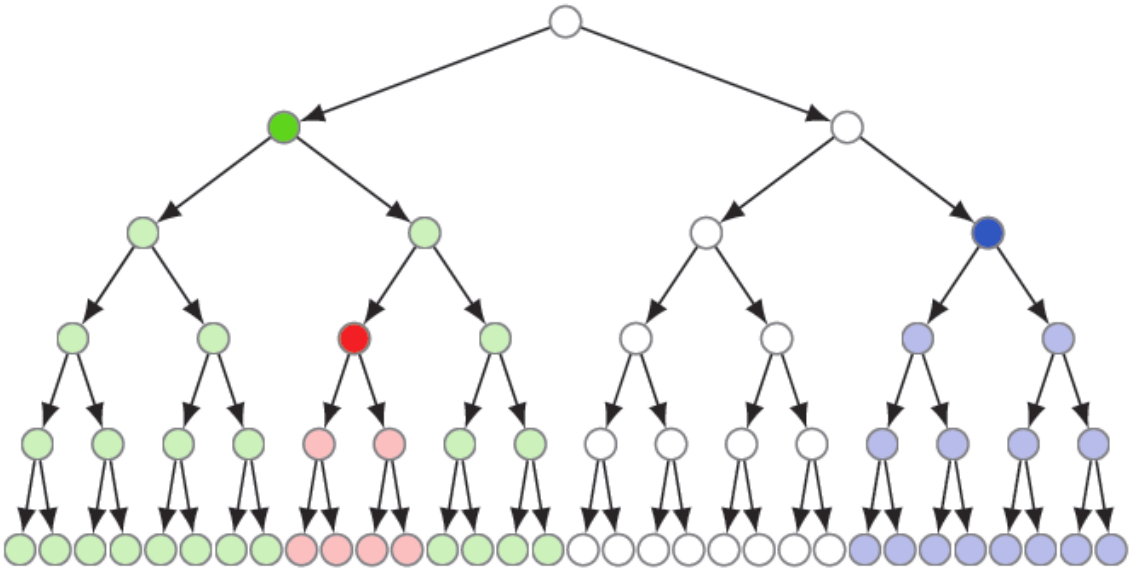
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



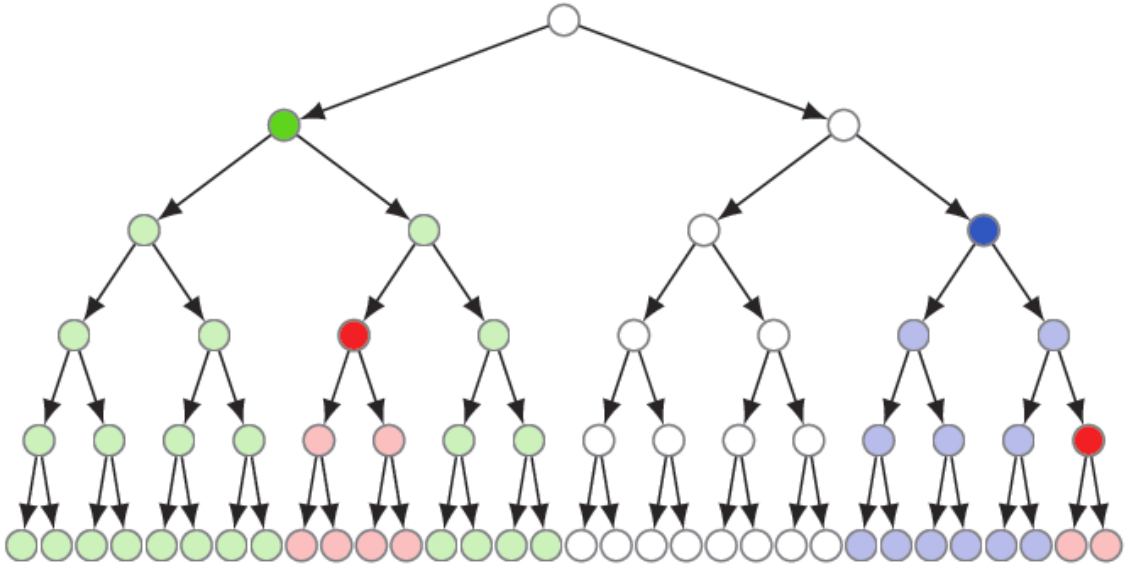
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



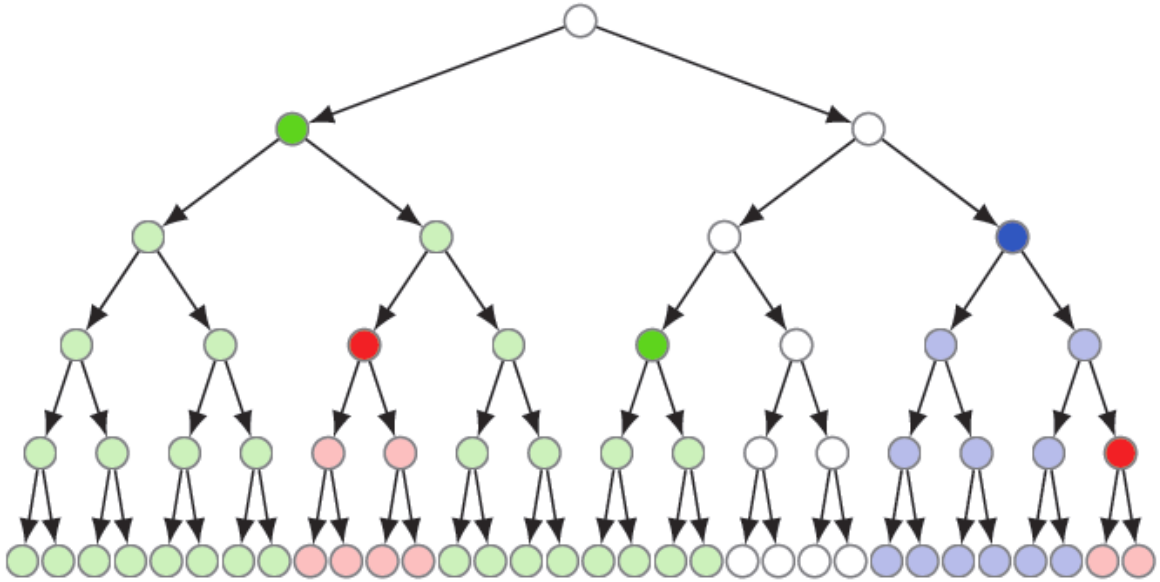
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	**1**
100/3	3
-/0	1



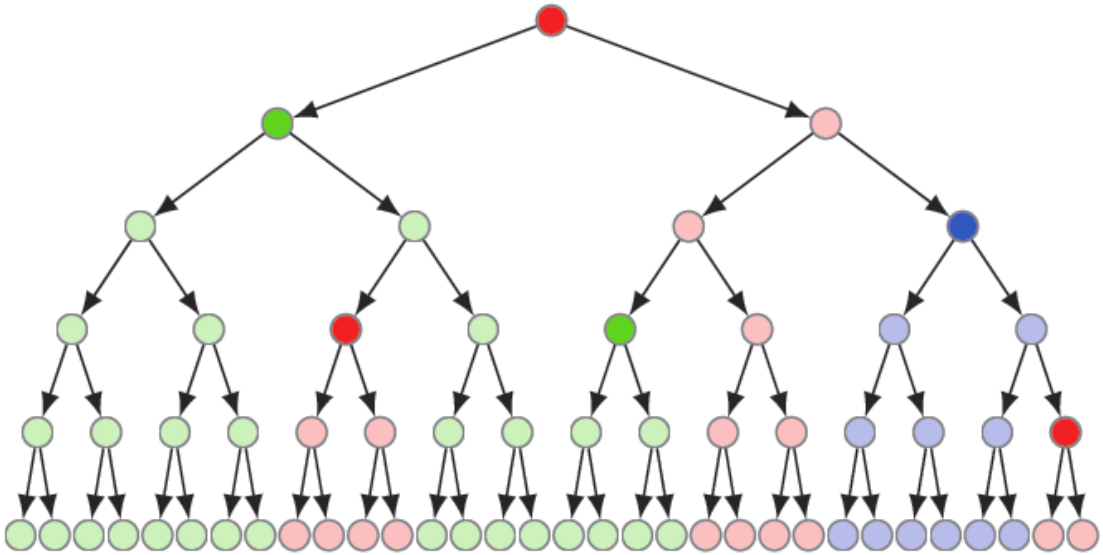
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



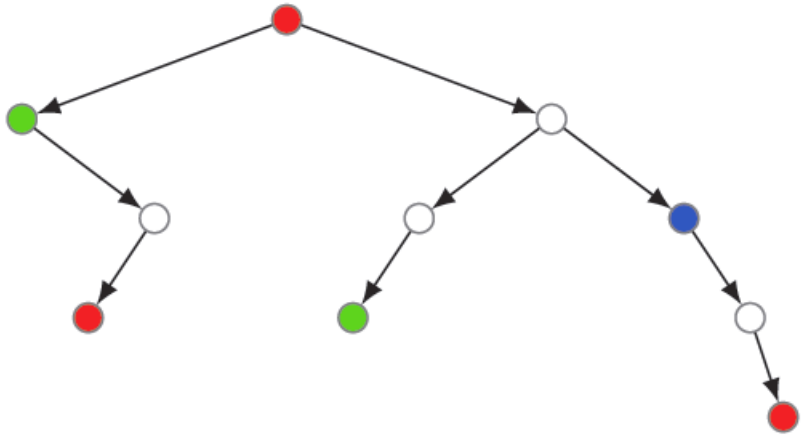
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



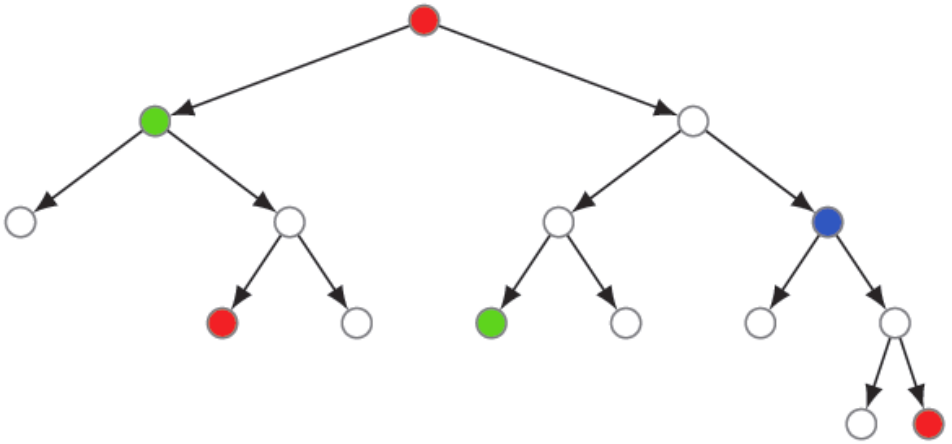
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



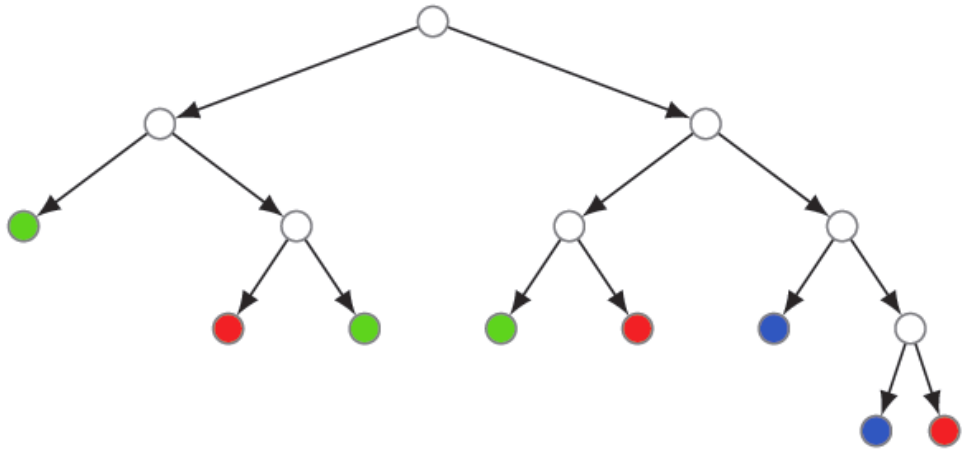
Example

prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



Example – Leaf Pushed

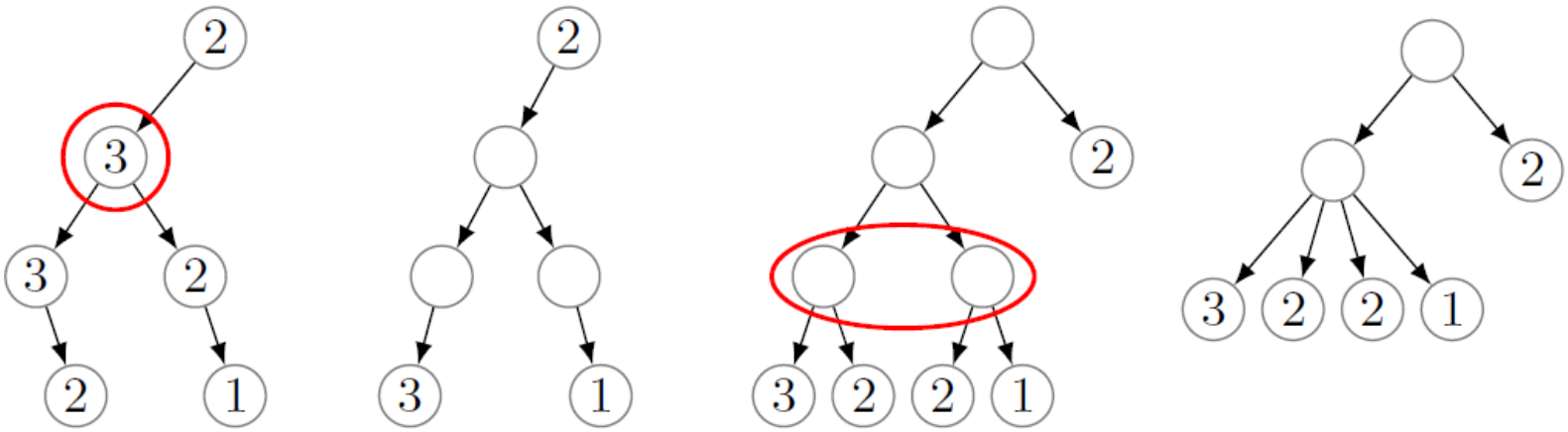
prefix	neighbour
010/3	1
11/2	2
0/1	3
1111/4	1
100/3	3
-/0	1



prefix	neighbour
010/3	1
11/2	2
1111/4	1
101/3	1
-/0	3

FIB Aggregation

- remove redundancy from the binary trie without changing forwarding behavior
  - Label-optimized trie
  - Leaf-pushed trie: proper leaf-labeled trie
    - unique (normalized) form
  - Level-compressed trie: remove excess levels

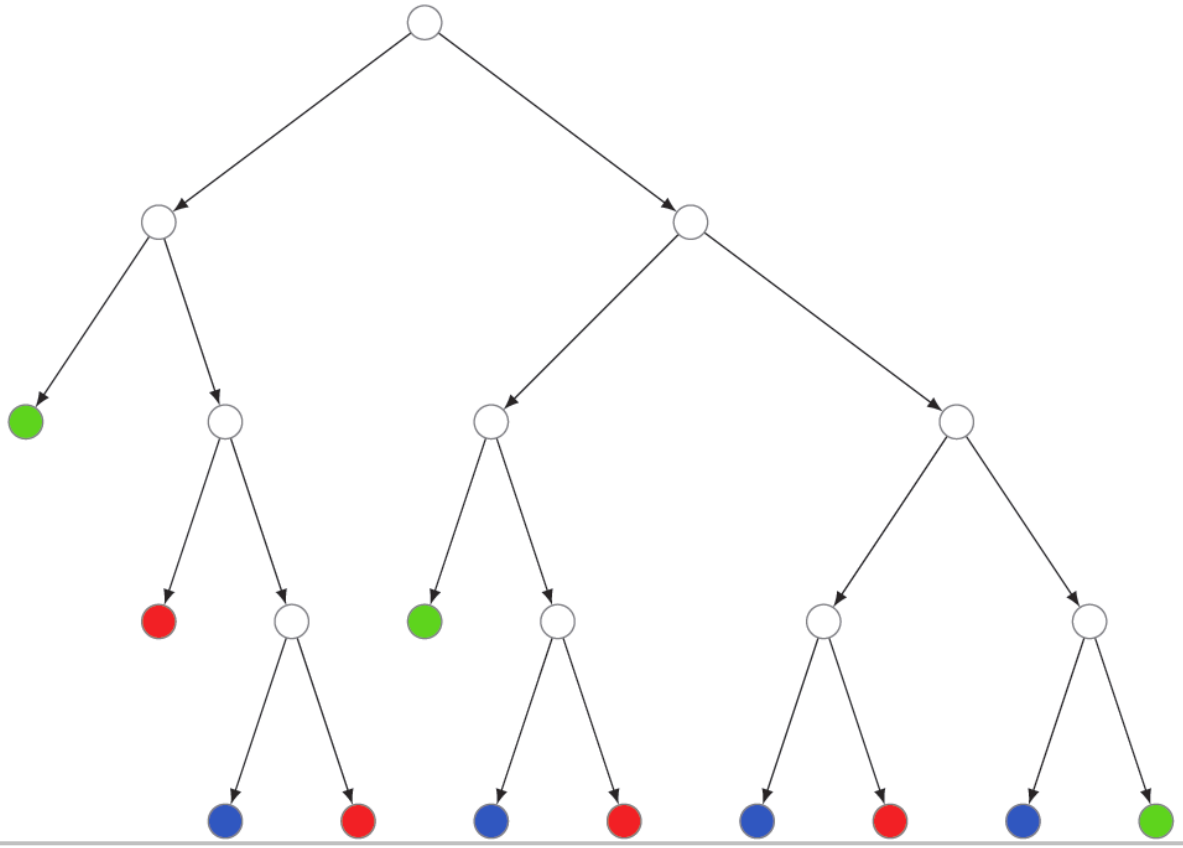


Min label

Min pointer

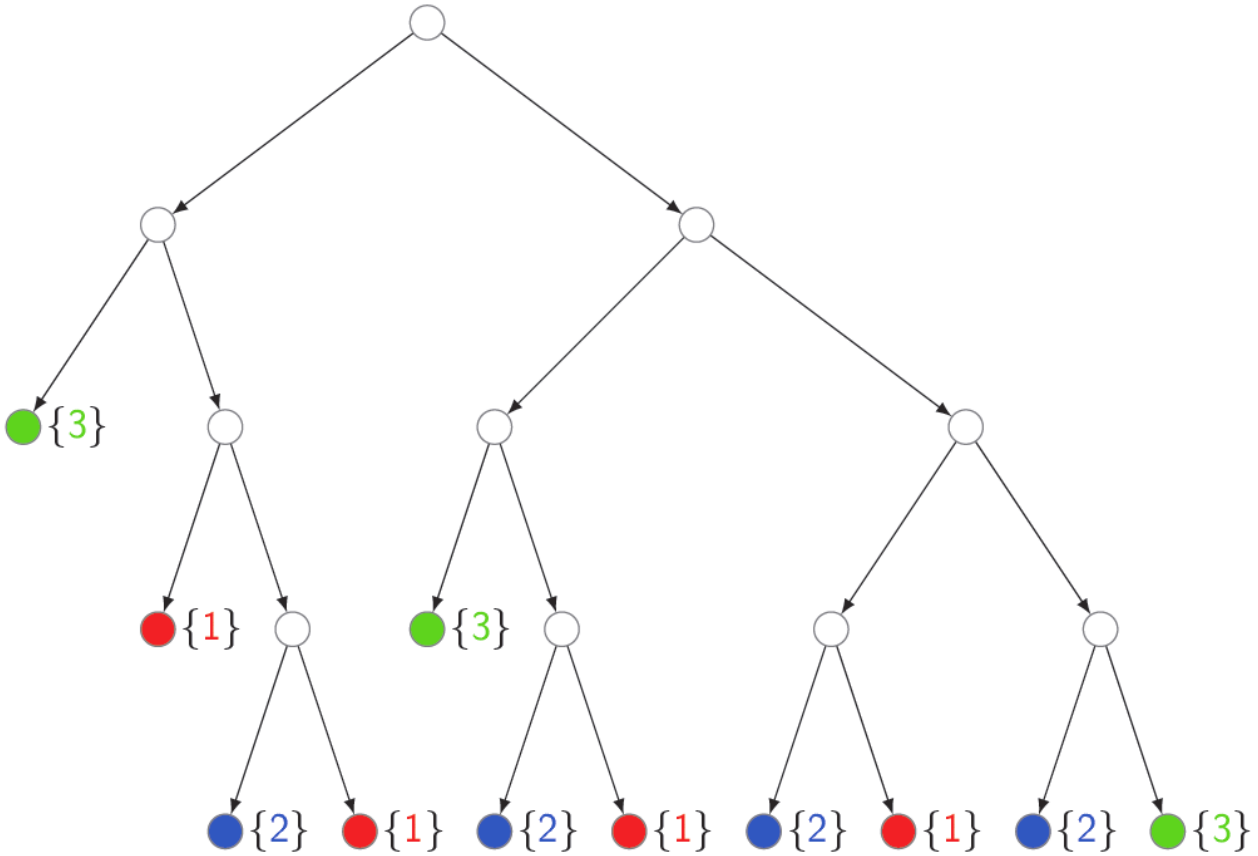
Optimal Routing Table Constructor (ORTC)

- Minimum label
  - Dynamic programming, INFOCOM'99



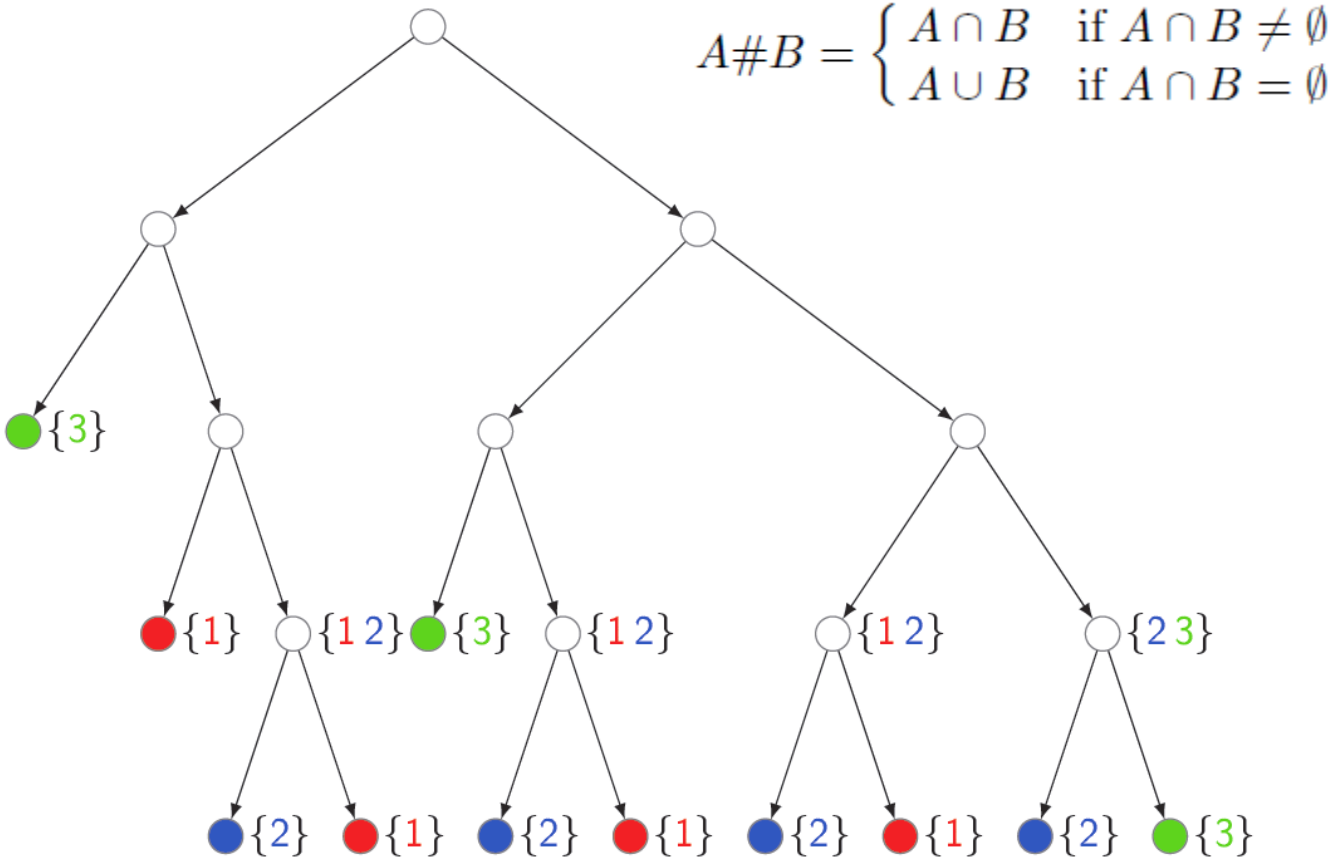
ORTC - Example

- Assign the set of labels for each branch



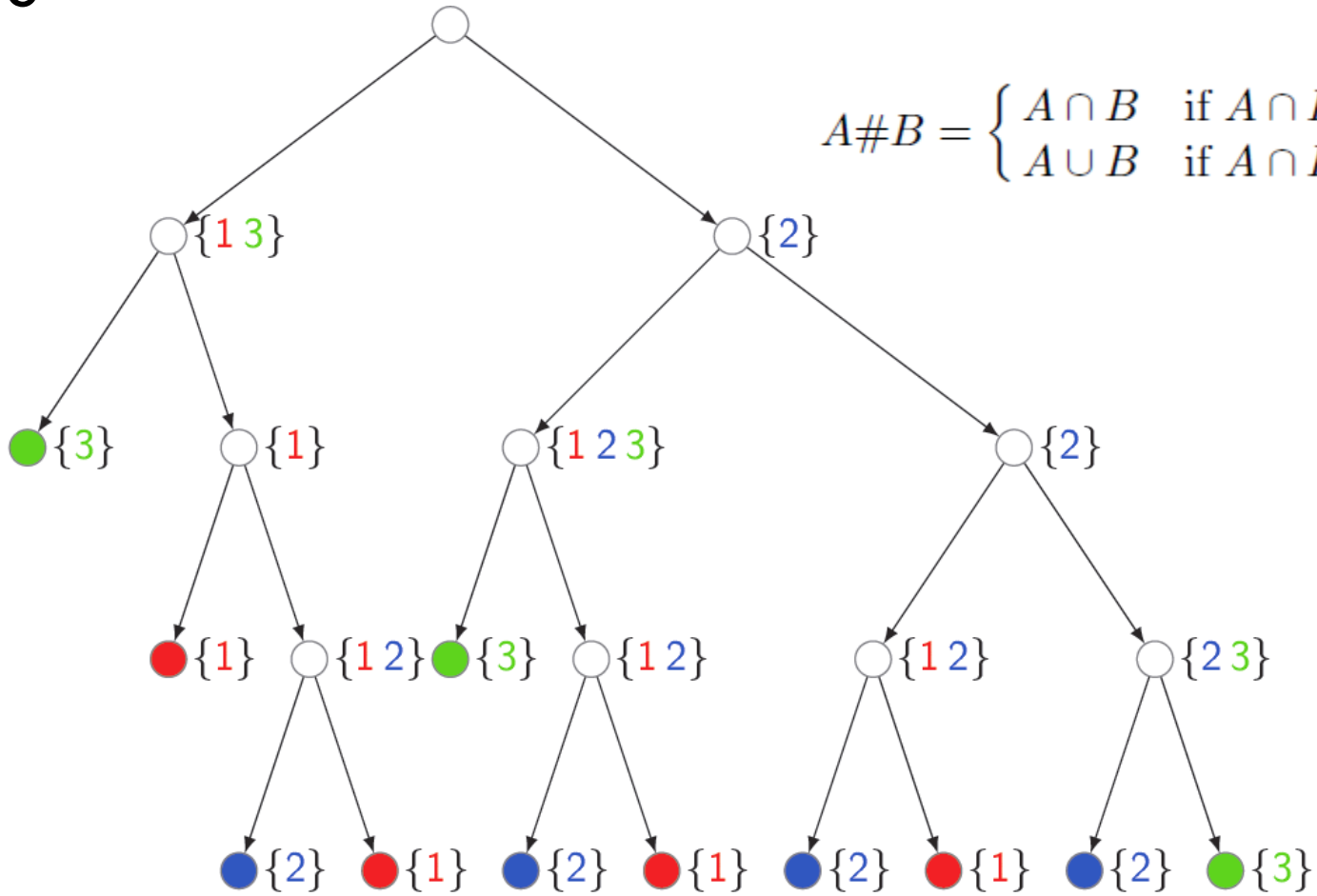
ORTC - Example

- Merge children



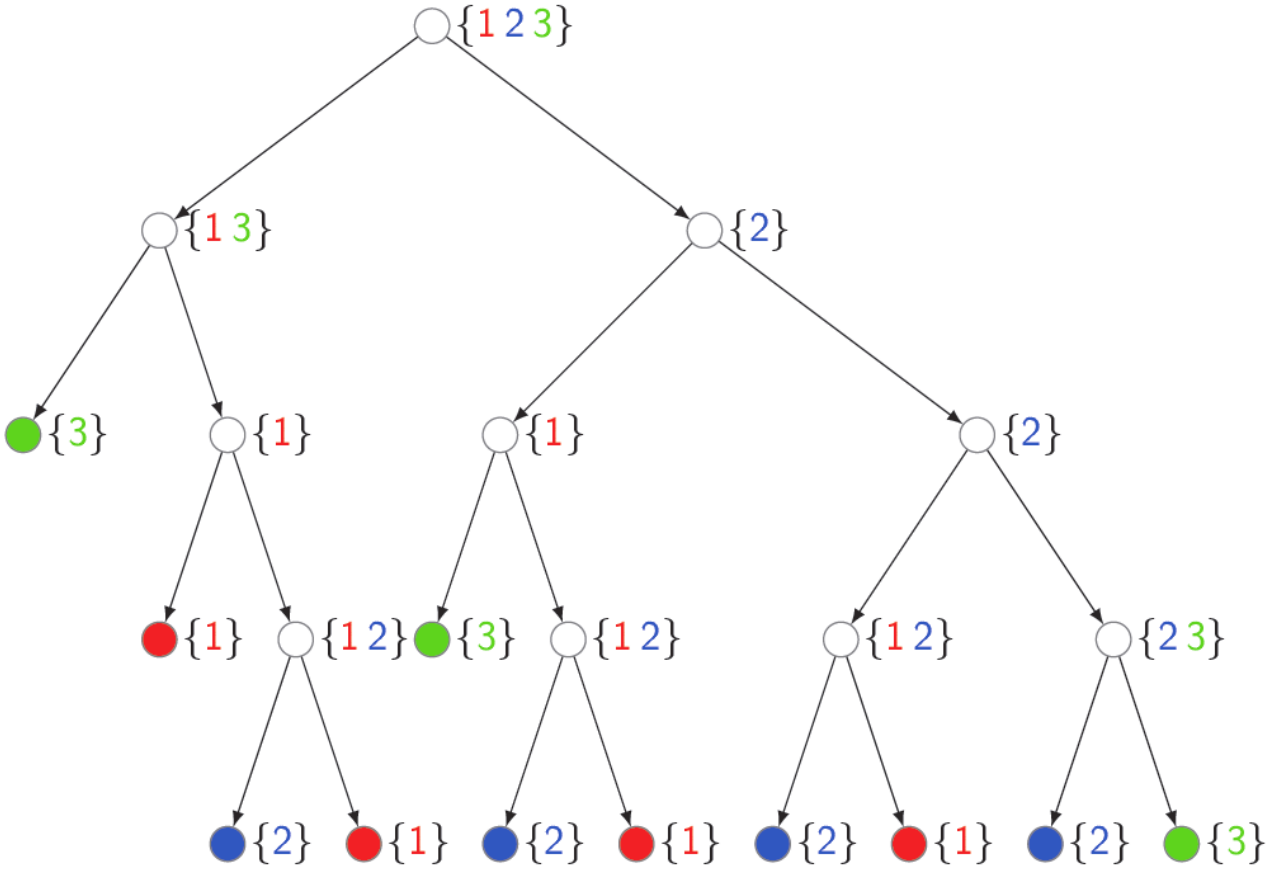
ORTC - Example

- Merge children



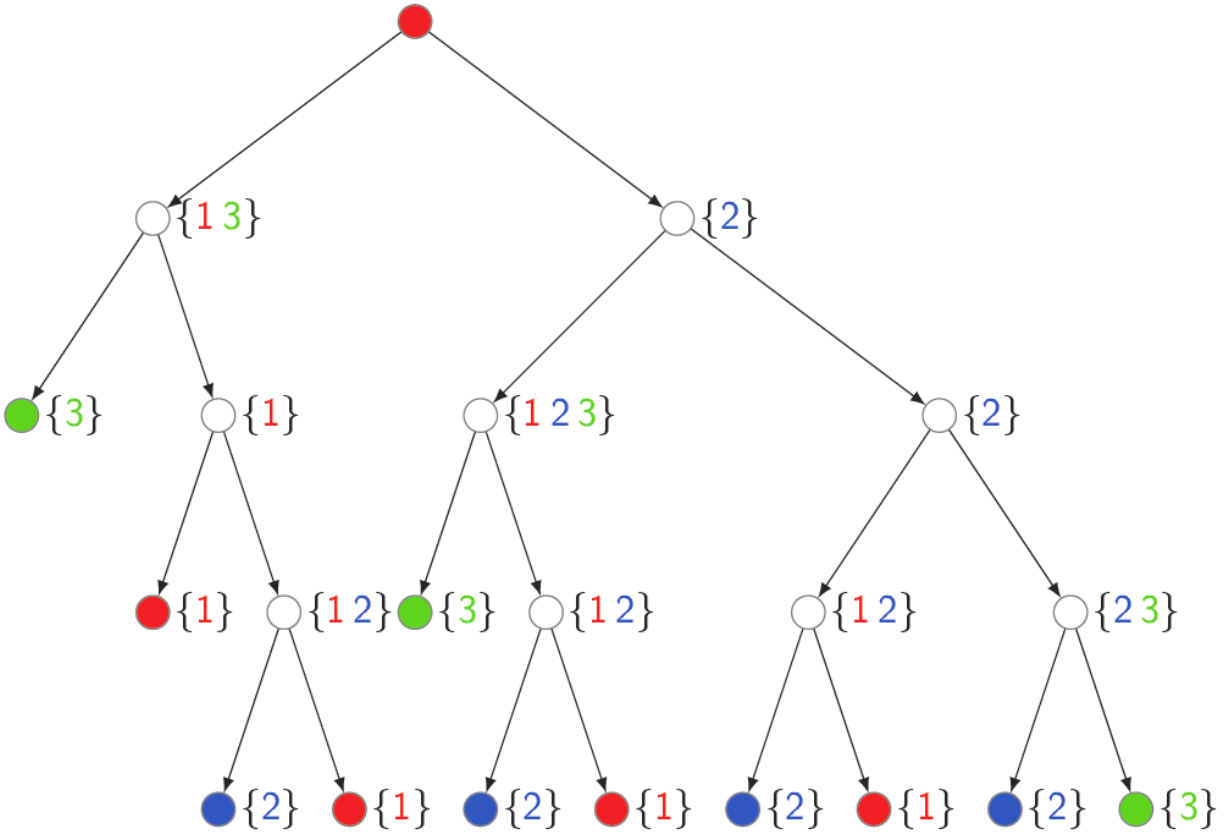
ORTC - Example

- Merging finished



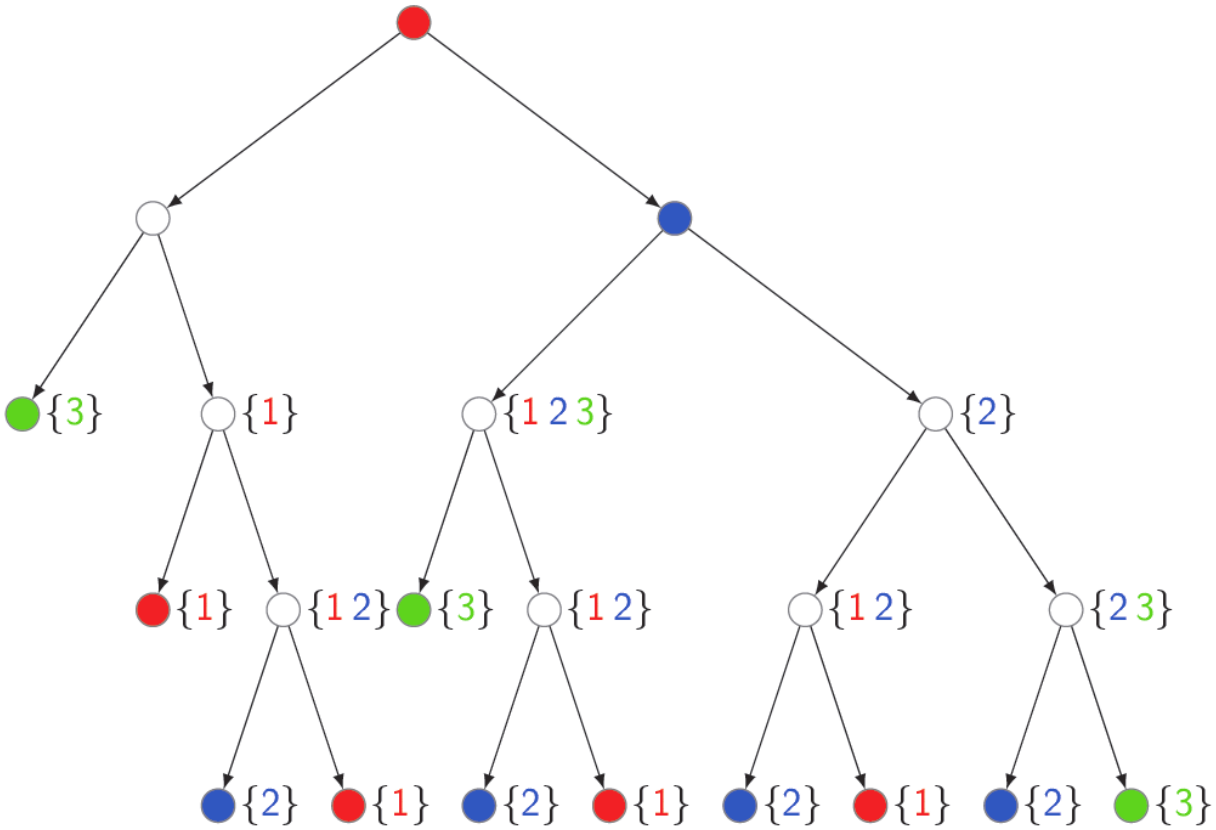
ORTC - Example

- From root of the tree
  - Skip if child next hop is the same as its parent



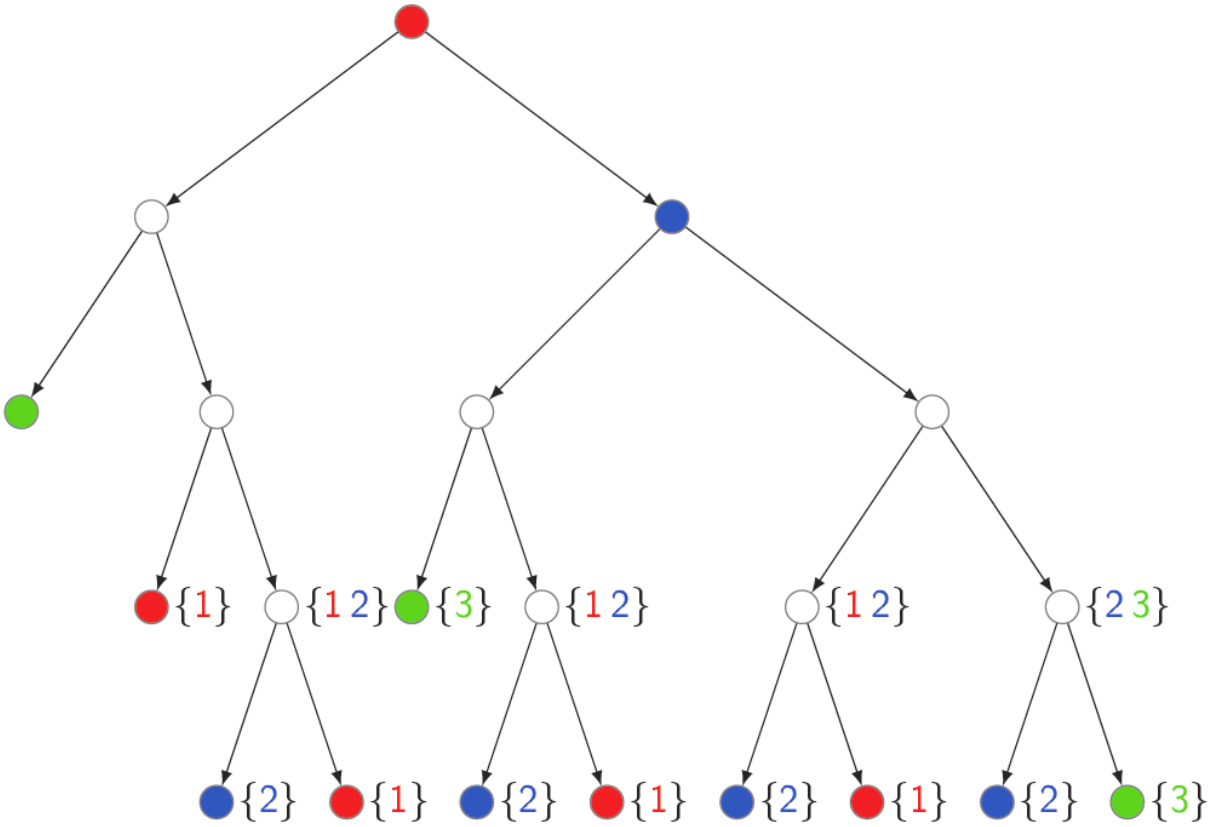
ORTC - Example

- From root of the tree
  - Skip if child next hop is the same as its parent



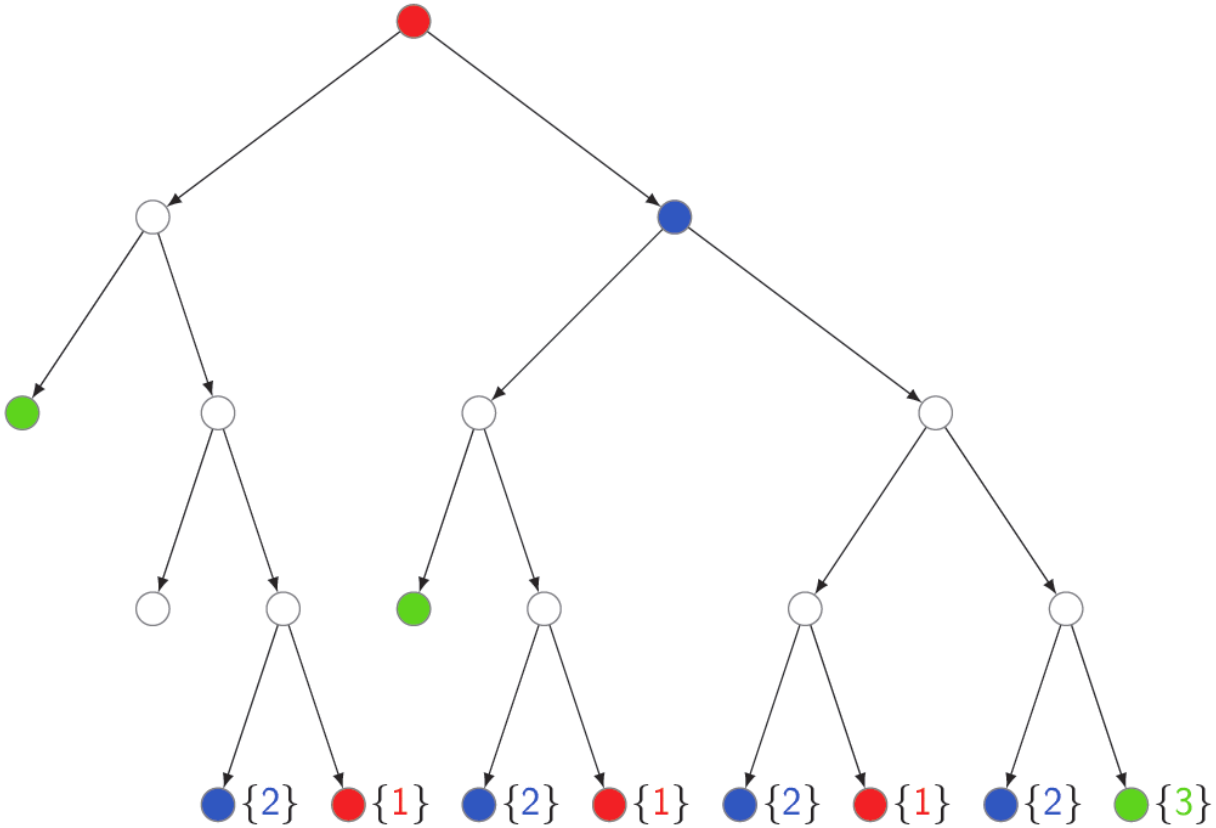
ORTC - Example

- From root of the tree
  - Skip if child next hop is the same as its parent



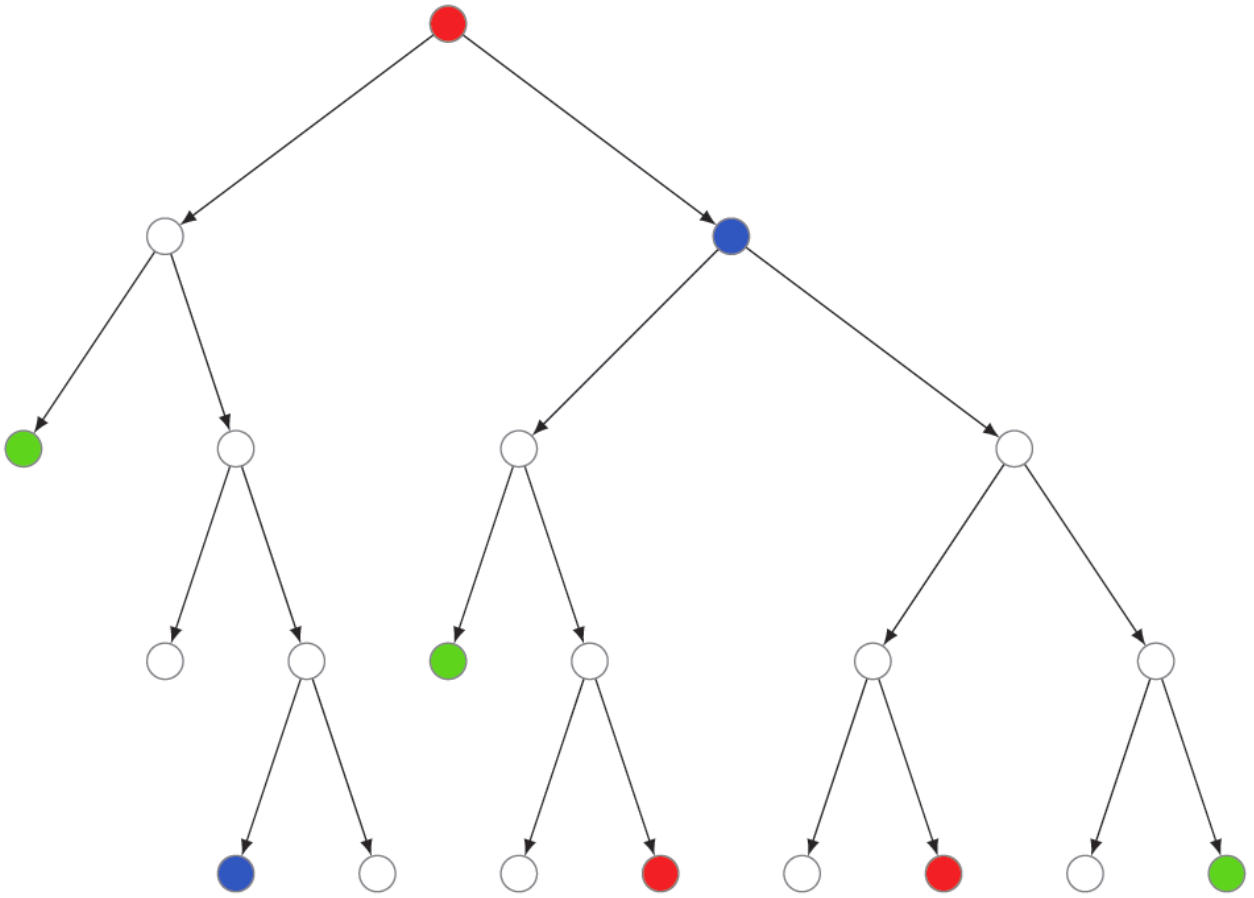
ORTC - Example

- From root of the tree
  - Skip if child next hop is the same as its parent



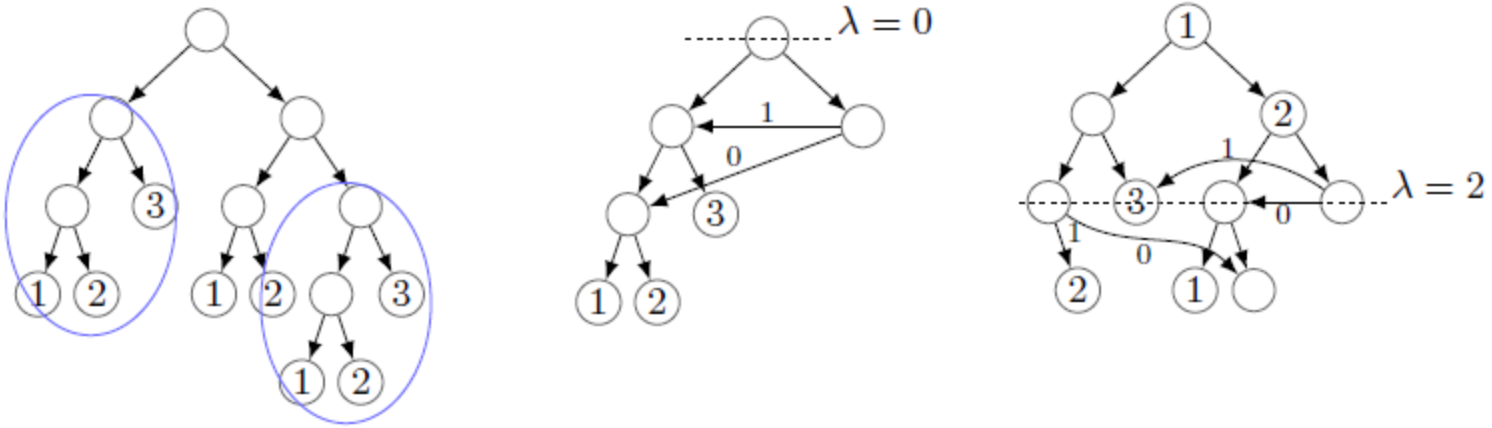
ORTC -Example

- Optimal solution



Dynamic FIBs: Trie-folding

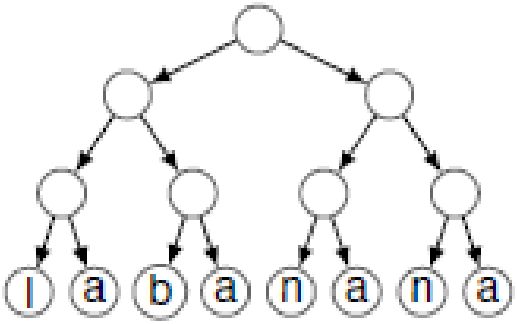
- Practical FIB compression, a good old pointer machine
- Fold the trie into a prefix DAG (DAFSA, DAWG, BDD)



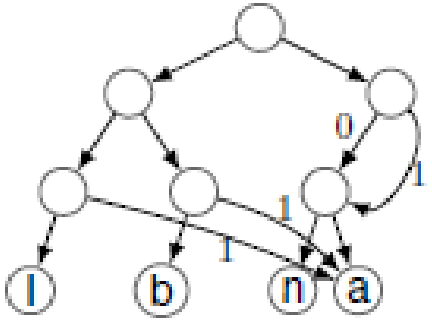
- For good compression, we need the tree to be in a prefix-free form
  - But prefix-free forms are expensive to update
  - Balance by a parameter λ, called the leaf-push barrier
- Deployable with minimal modification to router ASICs
  - ORTC (Min Label) for DAG is NP hard

Prefix DAG Size (Pointers)

- View the problem as string compression: encode a string *S* into a prefix DAG *D(S)*



S



$D(S)$

next hops

Shannon Entropy

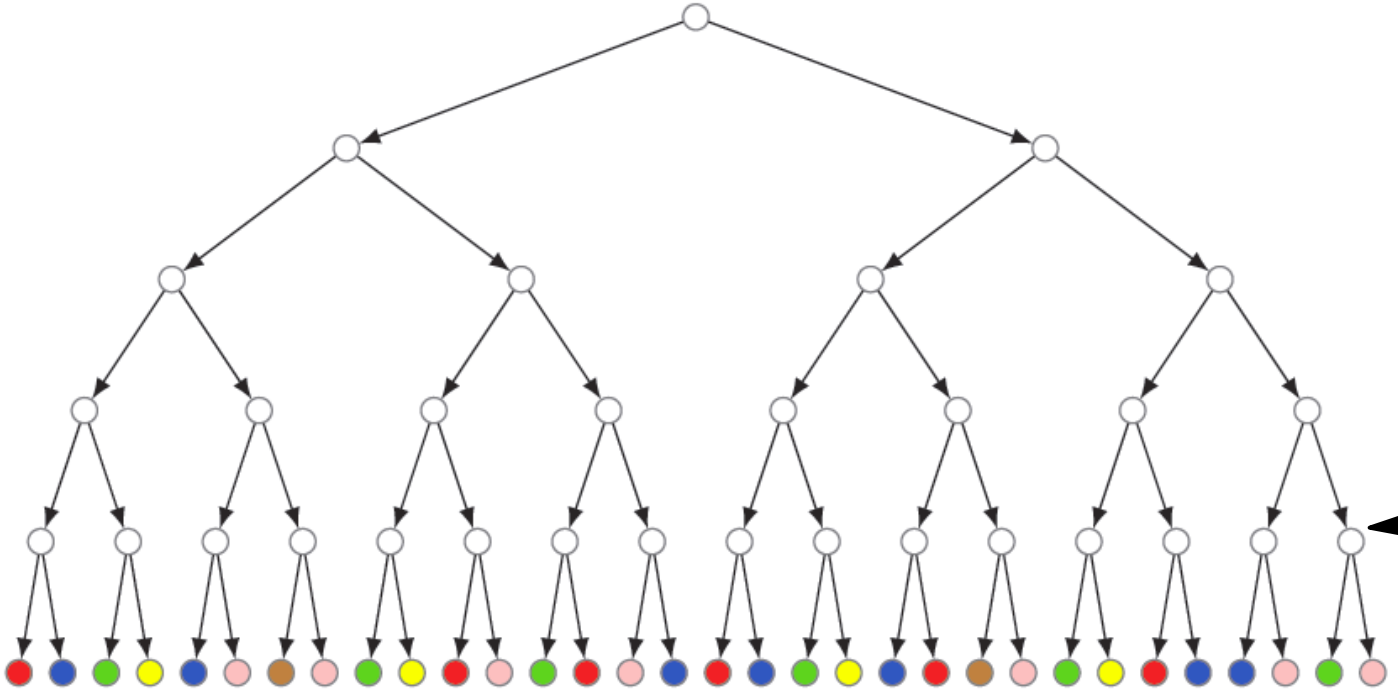
Theorem 1: *D(S) needs* $4n \log_2 \delta$ *bits at most*

Theorem 2: *D(S) can be squeezed into* $\sim 6nH_0$ *bits in expectation*

Theorem 3: *update goes in* $O((1 + 1/H_0) \log n)$ *steps*

Entropy bound

- For Shannon Entropy - We need a random model
  - String compression
  - Next hops are generated with a random process
  - p_s is the probability of a next hop



Coupon Collector's Problem O(nlog(n))

Coupon collector's problem with arbitrary coupon probabilities

- Set of coupon C, we draw *m* coupons
- At each draw coupon o has a probability p_o

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

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

Thus the expected cardinality of V is

$$E(|V|) = \sum_{o \in C} E(I(o \in V) = \sum_{o \in C} P(o \in V) =$$

$$= \sum_{o \in C} (1 - (1 - p_o)^m) \quad (2)$$

Let H_C denote the entropy of the coupon distribution

$$H_C = \sum_{o \in C} p_o \log_2 \frac{1}{p_o} \quad (3)$$

Coupon collector Lemma

Lemma 1. *n* number of coupons

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

for $m \geq 3$.

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

$$\sum_{o \in C}(1-(1-p_o)^m) \leq \frac{m}{\log_2(m)} \sum_{o \in C} p_o \log_2 \frac{1}{p_o} + 3 \qquad (4)$$

$3 > \frac{1}{e}$

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

$$1-(1-p_o)^m \leq \frac{m}{\log_2(m)} p_o \log_2 \frac{1}{p_o} \quad \text{if } p_o < \frac{1}{e} \qquad (5)$$

Case 1

$$1-(1-p_o)^m \leq \frac{m}{\log_2(m)} p_o \log_2 \frac{1}{p_o} \quad \text{if } p_o < \frac{1}{e} \tag{5}$$

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

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

Case 2

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

$$x := \log_{\frac{1}{p_o}} m \ . \tag{7}$$

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

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

$$= \frac{\frac{1}{p_o^x}}{x \log_2\left(\frac{1}{p_o}\right)} p_o \log_2\left(\frac{1}{p_o}\right) = \frac{\frac{1}{p_o^x}}{x} p_o = \frac{1}{x p_o^{x-1}} \ , \tag{8}$$

which can be reordered as

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

Case 2 Cont'

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

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

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

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

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

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

$$p_o \leq \frac{1}{p_o^{x-1}}. \tag{12}$$

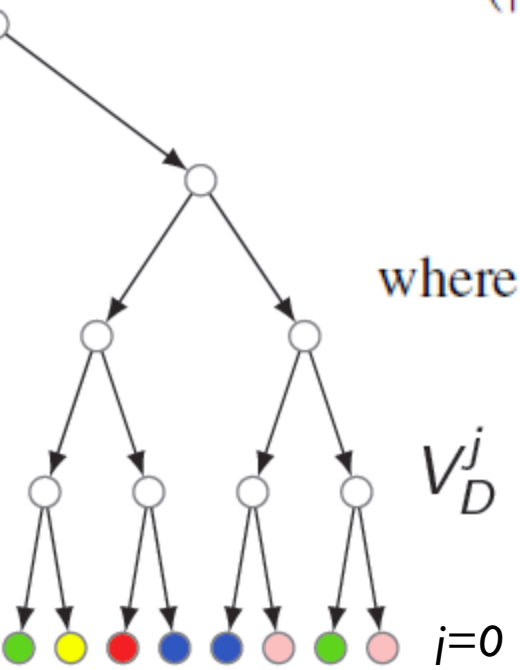
which holds because

$$p_o^x \leq 1. \tag{13}$$

Number of nodes in the DAG

- Coupon collector problem

h height



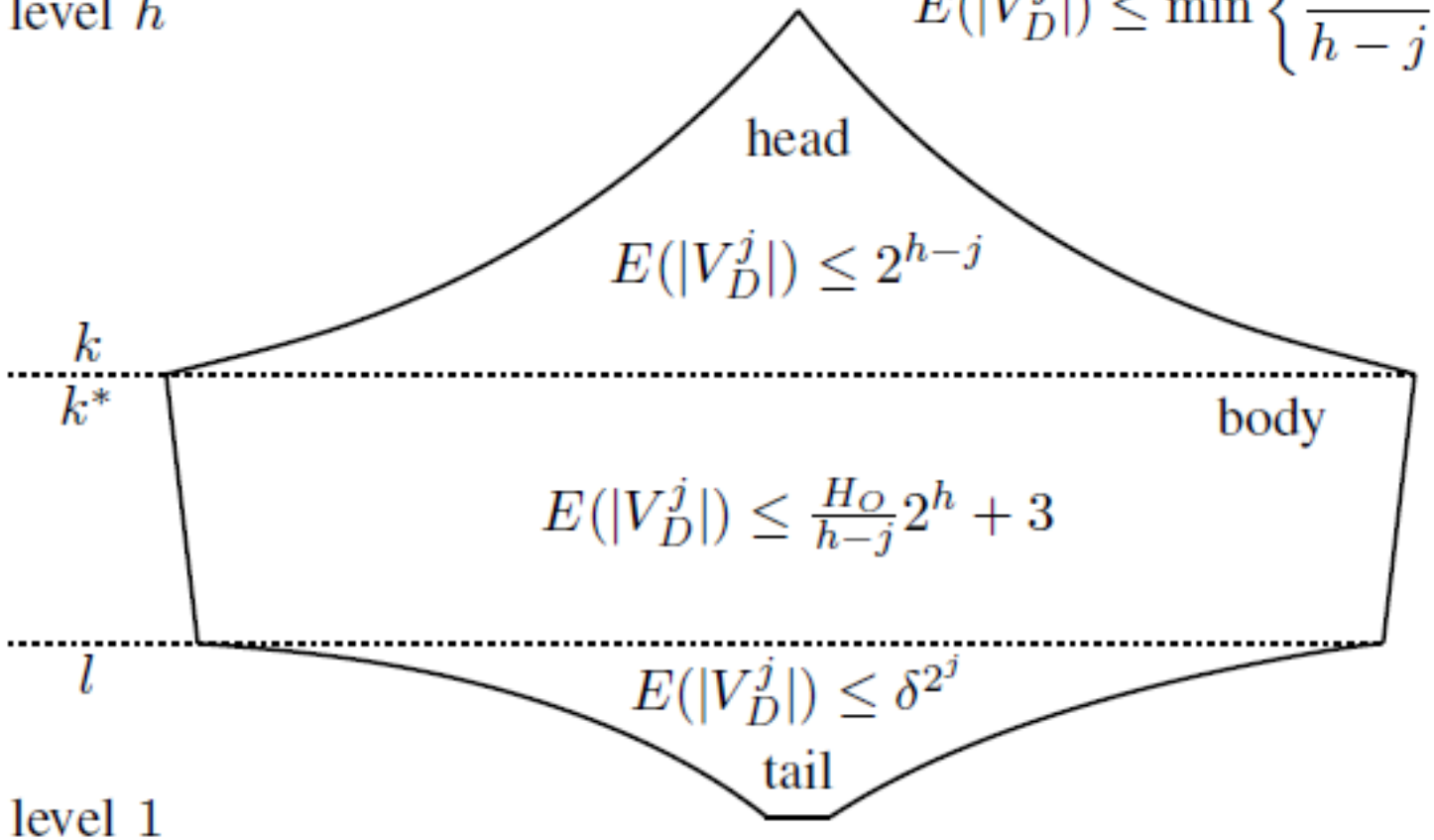
$$E(|V_D^j|) \leq \frac{2^{h-j}}{\log_2(2^{h-j})} H_C + 3 = \frac{2^{h-j}}{\log_2(2^{h-j})} H_O 2^j + 3$$
$$= \frac{2^h}{h-j} H_O + 3,$$

where $H_C = H_O 2^j$ is the entropy of a 2^j long string

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

δ number of next hops

Shape of DAG



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

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

Evaluation

Entropy bound (E) is way smaller than information theoretic limit (I)

	FIB	N	δ	H_0	I	E	XBW-b	pDAG
access	taz	410,513	4	1.00	94	56	63	178
	hbone	410,454	195	2.00	356	142	149	396
	access(d)	444,513	28	1.06	206	90	100	370
	access(v)	2,986	3	1.22	2.8	2.2	2.5	7.5
	mobile	21,783	16	1.08	0.8	0.4	1.1	3.6
core	as1221	440,060	3	1.54	130	115	111	331
	as4637	219,581	3	1.12	52	41	44	129
	as6447	445,016	36	3.91	375	277	277	748
	as6730	437,378	186	2.98	421	209	213	545
syn.	fib_600k	600,000	5	1.06	257	157	179	462
	fib_1m	1,000,000	5	1.06	427	261	297	782

- Several million lookups per sec both in HW and SW
  - faster than the uncompressed form
- pDAG tolerates more than 100, 000 updates per sec

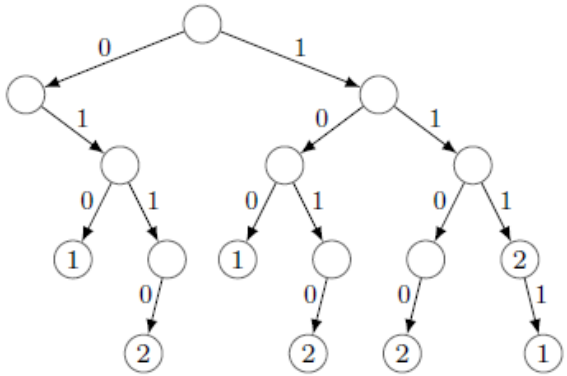
100–400 KBytes

Level Compressed prefix DAG

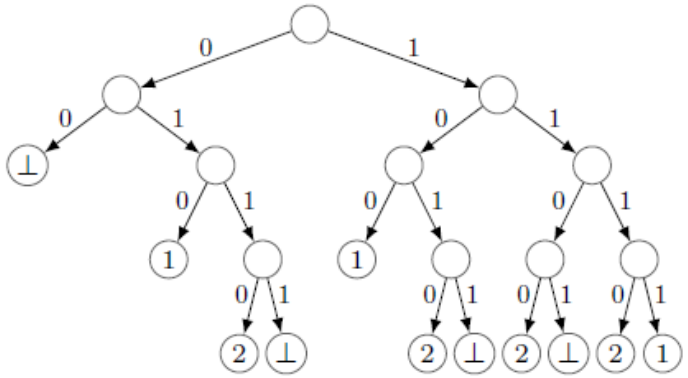
- Min Size DAG NP hard
- Lower bound by a Linear Program relaxation

prefix	label
010/3	1
0110/4	2
100/3	1
1010/4	2
1100/4	2
111/3	2
1111/4	1

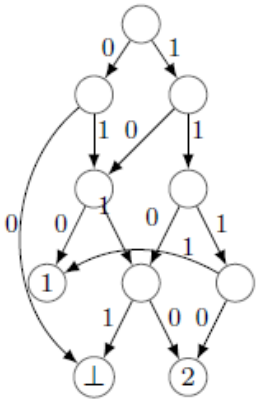
(a)



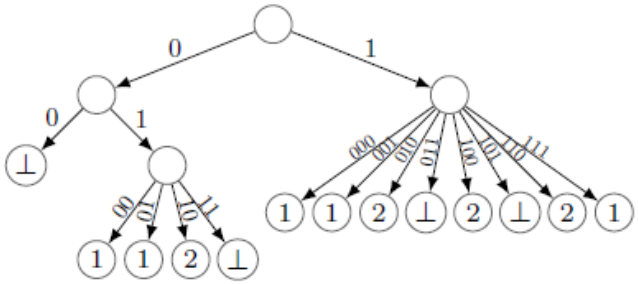
(b)



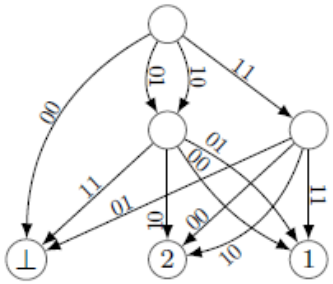
(c)



(d)



(e)



(f)

Evaluation

							fib_trie		lcTrie		bDAG		lcDAG	
		N	σ	H_0	I	E	M	ν	M	ν	M	ν	M	ν
HBONE	bme	499,211	89	1.20	196	71	32198	451.05	635.95	8.91	203.72	2.85	162.05	2.27
	szeged	499,236	87	1.20	196	71	32198	450.75	636.06	8.90	203.76	2.85	162.09	2.27
	vh1	499,143	207	2.34	407	185	32197	173.60	1157.48	6.24	503.22	2.71	393.34	2.12
	vh2	499,302	101	1.20	196	72	32202	450.31	636.49	8.90	204.13	2.85	162.36	2.27
Internet2	atlanta	14,312	93	1.91	38	14	1017	71.82	110.75	7.82	48.65	3.44	41.11	2.90
	houston	14,305	101	1.12	38	14	1016	71.93	109.47	7.75	49.00	3.47	41.49	2.94
	kansas	14,335	101	1.06	38	14	1019	73.31	110.43	7.95	47.38	3.41	39.99	2.88
access	taz	410,513	4	0.97	94	56	26698	474.62	519.23	9.23	172.95	3.07	138.75	2.47
	access(d)	444,513	28	1.61	206	95	28713	302.07	869.80	9.15	252.88	2.66	201.23	2.12
	access(v)	2,986	3	0.99	3	2	192	88.99	15.23	7.06	8.26	3.83	7.16	3.32
	mobile	21,783	10	1.62	1	0	290	655.67	2.47	5.59	1.27	2.88	1.19	2.69

		fib_trie					lcTrie					bDAG					lcDAG				
		$\overline{h}$	$h_{\max}$	mpps	#CPU	cm	$\overline{h}$	$h_{\max}$	mpps	#CPU	cm	$\overline{h}$	$h_{\max}$	mpps	#CPU	cm	$\overline{h}$	$h_{\max}$	mpps	#CPU	cm
HBONE	bme	2.43	8	4.07	758	23900	6.62	13	10.39	297	2145	25.85	31	10.69	289	1926	9.61	15	10.92	283	1903
	szeged	2.43	8	4.09	756	23583	6.62	13	10.40	297	2135	25.57	31	10.69	289	1960	9.49	15	10.92	283	1902
	vh1	2.43	8	4.09	754	23407	5.56	13	10.11	306	2533	25.52	31	9.92	311	2359	7.56	15	10.24	301	2247
	vh2	2.43	8	4.05	763	24463	6.62	13	10.40	297	2137	25.86	31	10.69	289	1965	9.55	15	10.88	284	1902
Internet2	atlanta	3.37	10	5.87	526	2442	6.31	14	9.94	311	1803	29.45	31	10.65	290	1842	11.04	17	11.15	277	1834
	houston	3.37	10	5.86	527	2476	6.15	14	9.99	309	1837	28.72	31	10.64	290	1839	11.12	16	11.11	278	1839
	kansas	3.37	10	5.87	526	2496	6.04	15	10.21	303	1796	29.44	31	10.65	290	1836	10.89	16	11.25	274	1823
access	taz	2.42	6	4.45	694	20735	6.53	13	10.57	292	2081	21.00	31	10.73	288	1882	7.75	16	10.96	282	1874
	access(d)	2.44	8	4.25	726	21941	6.06	14	9.25	334	2417	28.85	31	10.81	286	2105	10.96	16	11.00	281	1982
	access(v)	5.53	9	12.45	248	1798	10.31	15	12.94	239	1817	20.67	31	13.40	230	1828	10.67	18	14.83	208	1846
	mobile	6.68	9	12.33	250	1813	9.70	14	14.13	218	1776	26.44	31	17.67	174	1828	11.00	16	22.56	137	1832

Summary

- Applied mathematics
  - Combinatorial optimization, graph theory, algebra
- Compressed data structures
  - Packet Processing

tapolcai@tmit.bme.hu