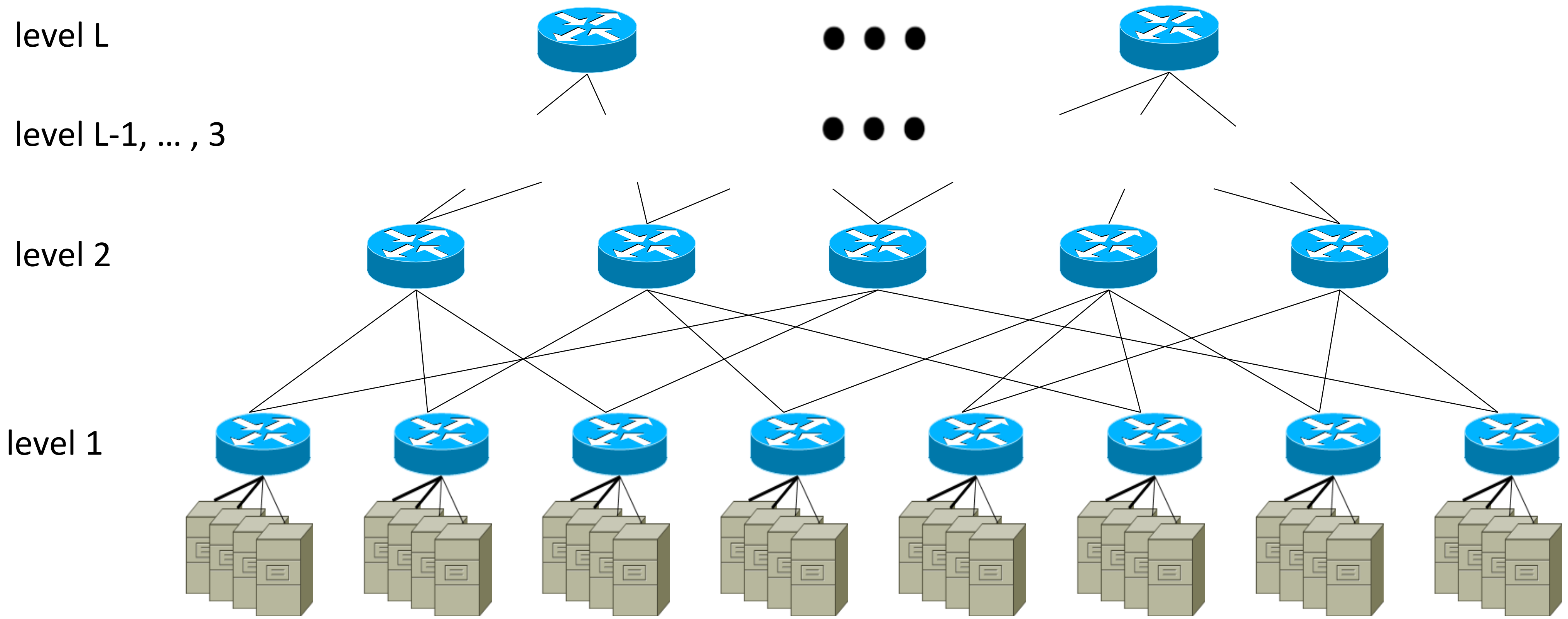
Path Routing Entry Compression Algorithm for Multi-level Multi-rooted Tree

Gaoxiong ZENG

SINGLab, CSE, HKUST

Topology

- Multi-level multi-rooted tree: Fat-tree, VL2 …



level L

level L-1, … , 3

level 2

level 1

Input

- Given a L-level multi-rooted tree

Output

- All desired paths between any two Edge Switch(i.e., l =1) and its ID;
- LPM routing entries of all switches;

Algorithm

- General step:

1) Search desired path;
2) Assign path ID according to Path ID Assignment Rules;
3) Add LPM routing entries to the switches along the path;

Notation

S^l - The switch set of level l; $l \in [1, L]$ where L is the size of the level

$R_{s_1-s_2}$ - The direct route (distance decrease by one level each hop) set from s_1 to s_2

DP_s - The downward port set of switch s

n_{s^l} - The sequence number (start from 0) of s^l in S^l

$n_{r_{s_1-s_2}}$ - The sequence number (start from 0) of $r_{s_1-s_2}$ in $R_{s_1-s_2}$

n_{dp_s} - The sequence number (start from 0) of dp_s in DP_s

Path ID Assignment Rule

- Desired Path : $s_{up}^{1} \rightarrow s_{up}^{2} \rightarrow \cdots\ \rightarrow s^{t+2} \rightarrow \cdots s_{down}^{2} \rightarrow s_{down}^{1}$;
- Therefore, there are L-1 types of path, i.e., $t \in [0, L-2]$;
- Path ID :

Type . Top . Route . DP_t+2 . DP_t+1 . … . DP_2

or

$$\text{t.}\, n_{s^{t+2}} \,.\, n_{r_{s_{up}^{1}-s^{t+2}}} \,.\, n_{dp_{s^{t+2}}} \,.\, n_{dp_{s_{down}^{t+1}}} \,.\, \ldots \,.\, n_{dp_{s_{down}^{2}}}$$

For each type, block size is fixed and should be calculated beforehand;

- Type: $\lceil \log_2(L-1) \rceil$ bits
- Top: $\lceil \log_2 |S^{t+2}| \rceil$ bits
- Route: $\left\lceil \log_2 \max_{s^1 \in S^1, s^{t+2} \in S^{t+2}} \left| R_{s^1 - s^{t+2}} \right| \right\rceil$ bits
- DP_l: $\left\lceil \log_2 \max_{s \in S^l} |DP_s| \right\rceil$ bits

Note:

$\lceil a \rceil$ denotes the smallest integer that is larger than a;

$|S|$ denotes the set size of S;

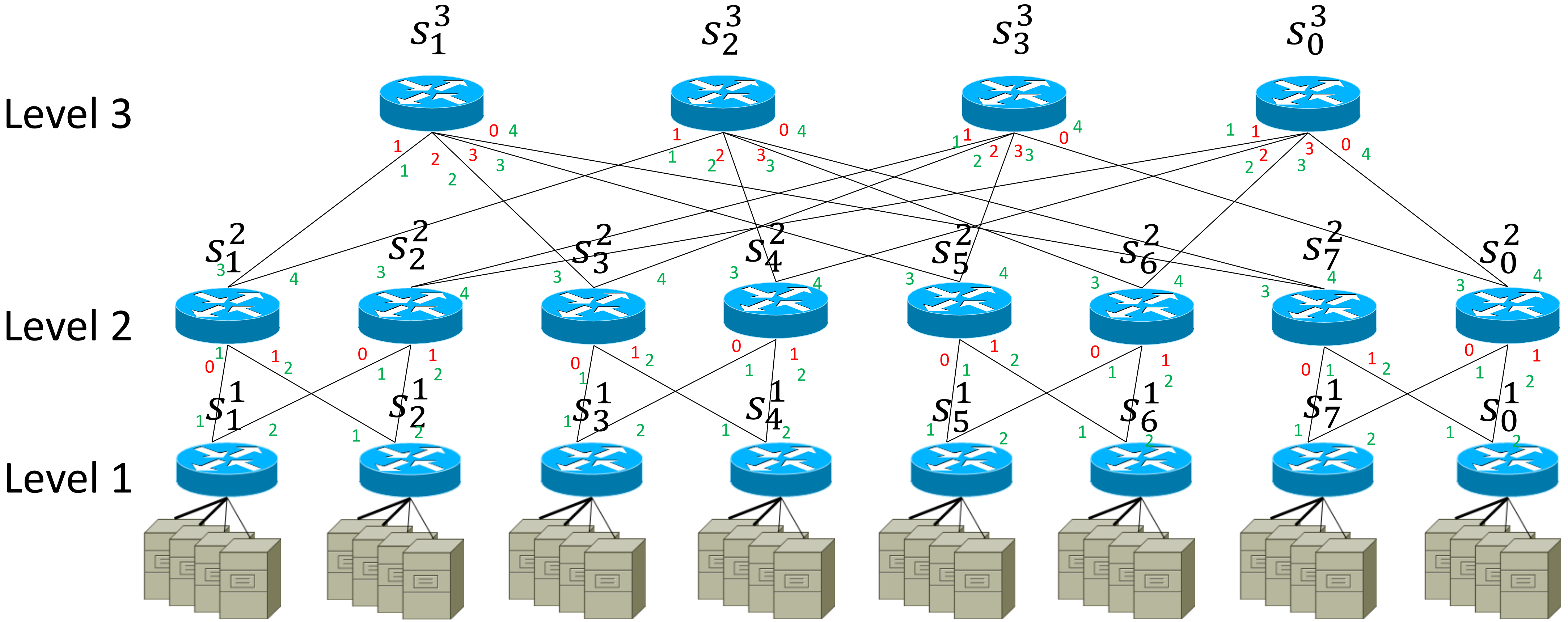
LPM routing entry

Typical for 3-level topology!

Switch level Path Type		level 1	level 2	level 3	…	level L
Type 0	Upward	0.Top.Route/				
	Downward		0.Top.Route.DP_2/			
Type 1	Upward	1.Top.Route/	1.Top.Route/			
	Downward		1.Top.Route.DP_3.DP_2/	1.Top.Route.DP_3/		
…						
Type L-2	Upward	t.Top.Route/	t.Top.Route/	t.Top.Route/		
	Downward		t.Top.Route.DP_L. … .DP _2/	t.Top.Route.DP_L. … .DP_3/		t.Top.Route.DP_L/

Path ID : Type . Top . Route . DP_t+2 . DP_t+1 . … . DP_2

Illustration – Fattree(4)



s_n^l denotes the n-th switch of level l ;

The green number denotes the physical port number;

The red number denotes the corresponding downward port sequence number.

Type: $\lceil \log_2(L-1) \rceil$ = $\lceil \log_2(3-1) \rceil$ = 1 bit

DP_3: $\left\lceil \log_2 \max_{s \in S^l} |DP_s| \right\rceil$ = $\left\lceil \log_2 \max_{s \in S^3} |DP_s| \right\rceil$ = 2 bits

DP_2: $\left\lceil \log_2 \max_{s \in S^l} |DP_s| \right\rceil$ = $\left\lceil \log_2 \max_{s \in S^2} |DP_s| \right\rceil$ = 1 bit

- For type 0, t = 0,

Top: $\lceil \log_2 |S^{t+2}| \rceil$ = 3 bits

Route: $\left\lceil \log_2 \max_{s^1 \in S^1, s^{t+2} \in S^{t+2}} \left| R_{s^1 - s^{t+2}} \right| \right\rceil$ = 0 bit

- For type 1, t = 1,

Top: $\lceil \log_2 |S^{t+2}| \rceil$ = 2 bits

Route: $\left\lceil \log_2 \max_{s^1 \in S^1, s^{t+2} \in S^{t+2}} \left| R_{s^1 - s^{t+2}} \right| \right\rceil$ = 0 bit

Note: 0 bit means we don’t need that block, or the block is null!

- LPM routing entries of some switches:

Entry Form: Path_ID_Prefix/ Physical_Port

Switch Path Type		s_1^1	s_1^2	s_1^3
Type 0	Upward	0.001.null/ 1 0.010.null/ 2		
	Downward		0.001.null.0/ 1 0.001.null.1/ 2	
Type 1	Upward	1.01.null/ 1 1.10.null/ 1 1.11.null/ 2 1.00.null/ 2	1.01.null/ 3 1.10.null/ 4	
	Downward		1.01.null.01.0/ 1 1.01.null.01.1/ 2 1.10.null.01.0/ 1 1.10.null.01.1/ 2	1.01.null.01/ 1 1.01.null.10/ 2 1.01.null.11/ 3 1.01.null.00/ 4

Bottleneck

Scalability

	Fat-tree(64)	Fat-tree(256)	VL2(100,96,100)
ID Space	22 bits	30 bits	22 bits
Max. Entries	1088	16640	5100