BACKORDERS: Using Random Forests to Detect DDoS Attacks in Programmable Data Planes

Bruno Coelho, Alberto Schaeffer-Filho

Federal University of Rio Grande do Sul (UFRGS), Brazil

Context

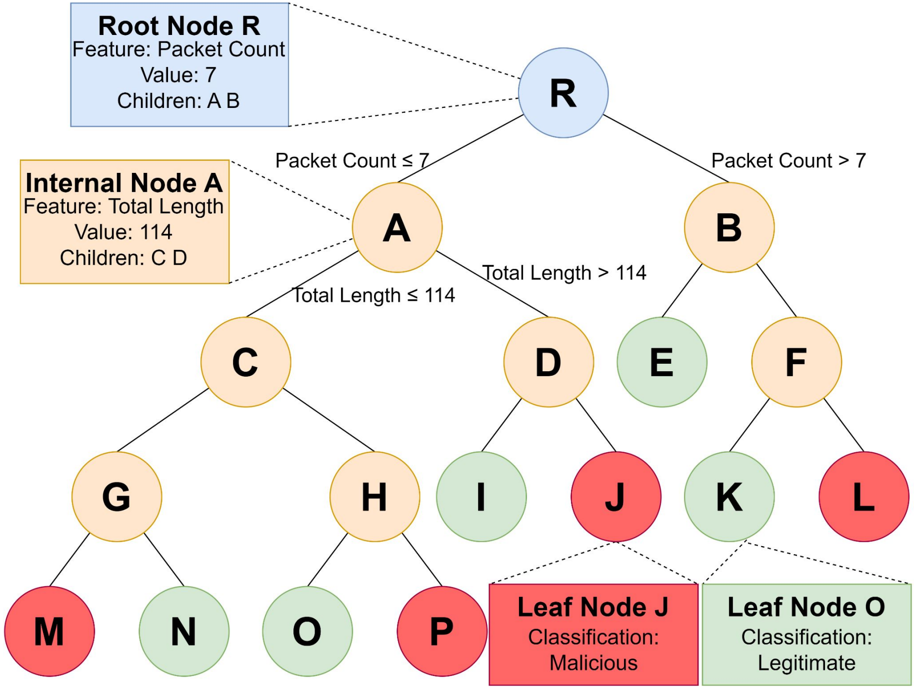
- Distributed Denial of Service (DDoS) attacks remain an issue
- Even short downtime can result in losses
  - Amazon's 1 hour of downtime cost over $72 million on Prime Day 2018
- Detection is difficult
  - IP and Port Spoofing
  - Application-layer exploits
  - Accuracy vs Scalability

Motivation

- Programmable Data Planes (PDP)
  - Custom logic defined by software artifacts
  - Designed to process packets at line-rate
- Random Forests (RF)
  - Able to identify patterns to classify network traffic
  - Requires simple logic and arithmetic operations
  - Processing classification trees can be parallelized
  - Relatively compact data structures

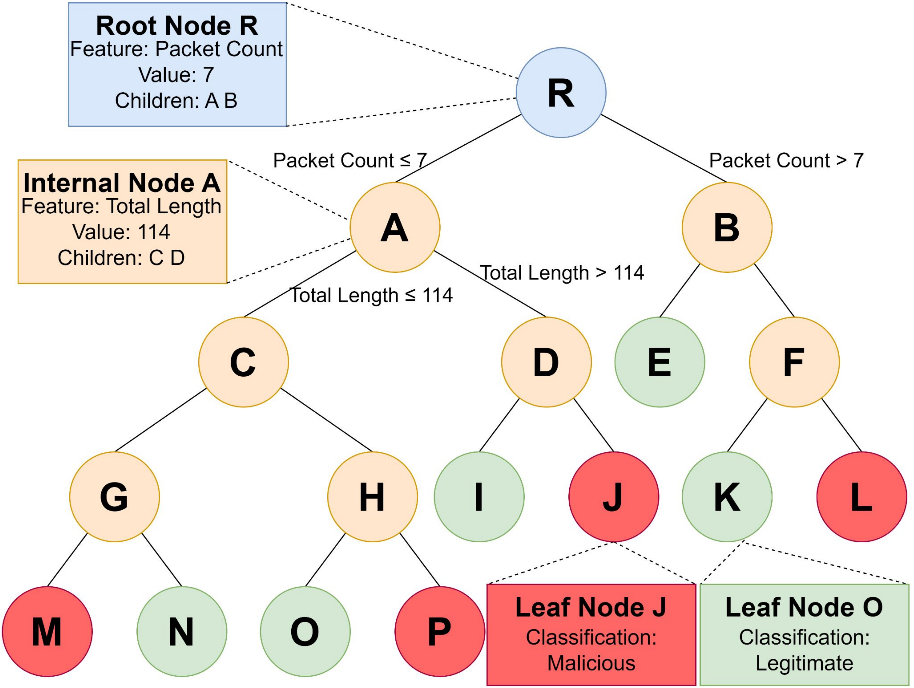
Classification Tree Nodes

- Internal Nodes
  - Feature
  - Threshold value
  - Children
- Node structures are naturally recursive
  - A node contains another node (children)
- P4 does not support recursion
  - Cannot predict number of calls
- Leaf Nodes
  - Classification



Bruno Coelho, Alberto Schaeffer-Filho

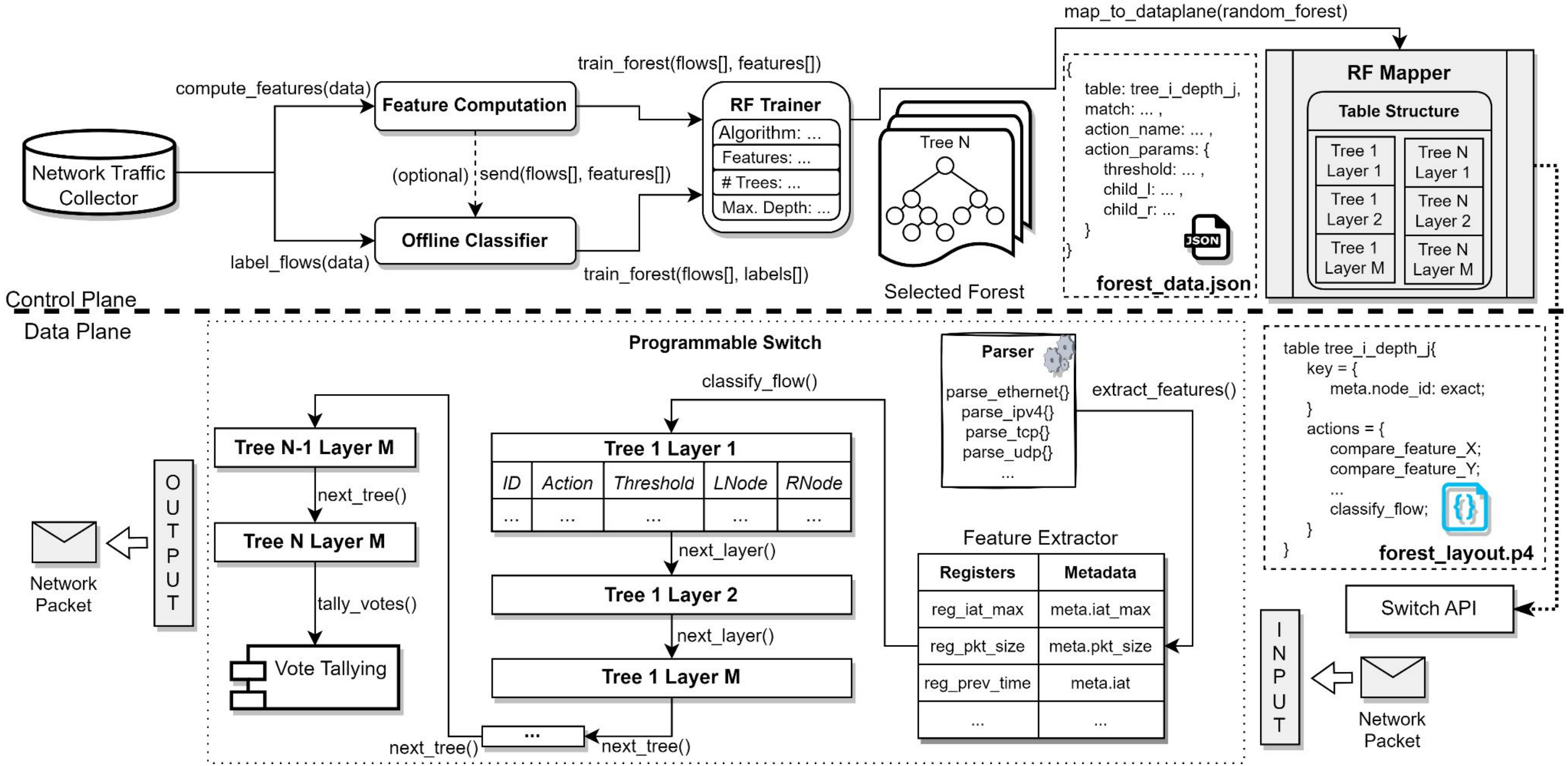
Mapping nodes to the Data Plane



Match Value	Action	Parameters		
Node ID		Threshold	Child 1	Child 2
0	compare_pkt_count	7	1	2
1	compare_total_length	114	3	4
2	compare_feature_B	y	5	6
8	compare_feature_H	z	15	16

Match Value Node Identifier	Action	Parameters Classification
5	classify_flow	LEGITIMATE
9	classify_flow	LEGITIMATE
10	classify_flow	MALICIOUS
11	classify_flow	LEGITIMATE
12	classify_flow	MALICIOUS
13	classify_flow	MALICIOUS

BACKORDERS Architecture



Feature extraction in the Data Plane

- RFs require flow features as input
- Most statistical features are simple
  - Sum, max, min, duration
- Some statistical features require complex operations
  - Quantiles, means, variance
- We focused on approximating moving means (averages)
  - P4 does not support division

Approximating Means

i	V_i	$S_e(i)$	$S_a(i)$	$M_a(i)$	**Mean**	**Formula:** $S_a(i)$	**Formula:** $M_a(i)$
8	15	160	160	20	20	$S_e(8)$	$S_e(8)/8$

$$S_e(7) = 145 \qquad V_8 = 15 \quad M_a(8) = \frac{160}{8} = 20$$

$$S_e(8) = S_a(8) = 145 + 15$$

Approximating Means

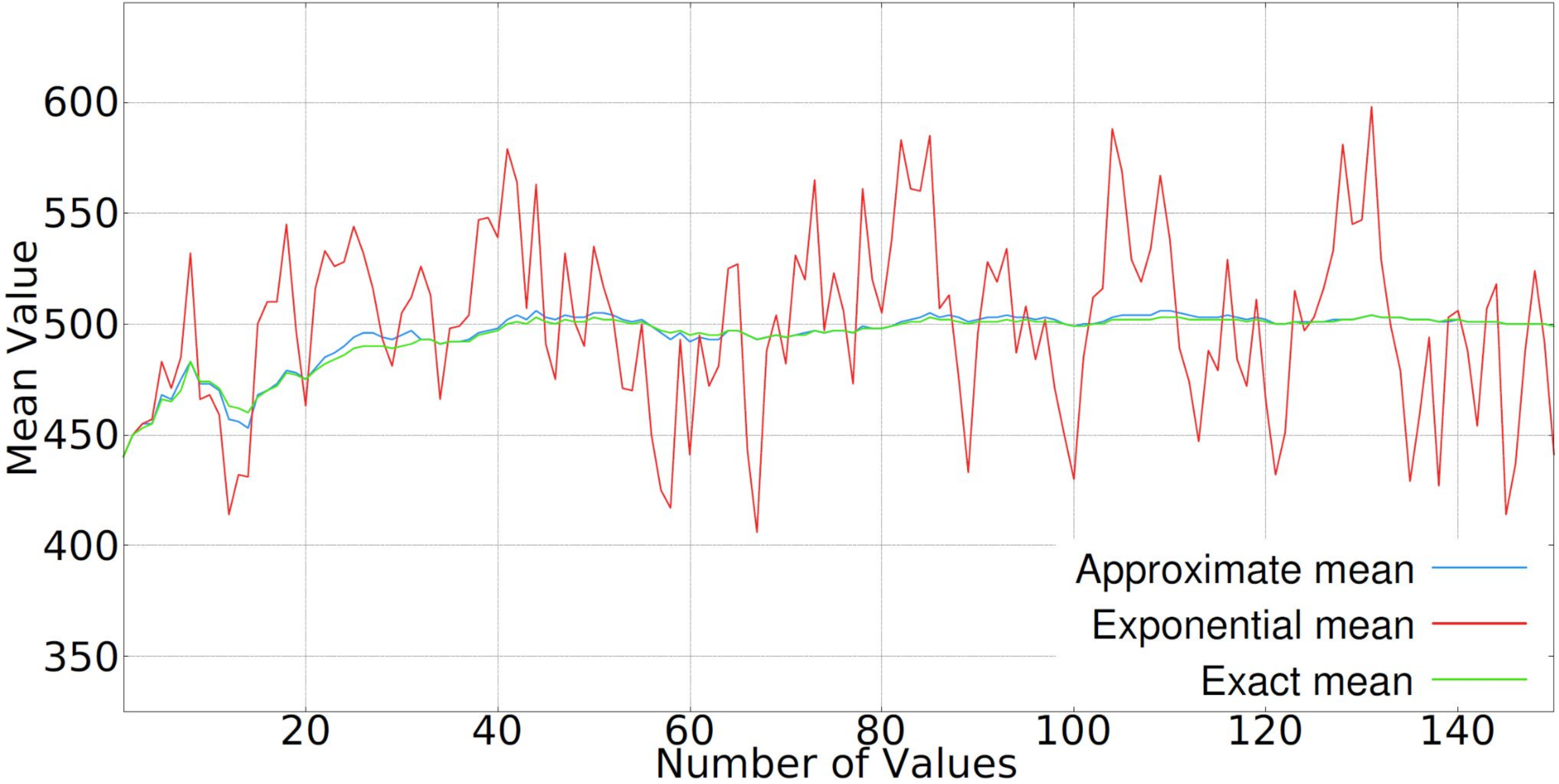
i	V_i	$S_e(i)$	$S_a(i)$	$M_a(i)$	**Mean**	**Formula:** $S_a(i)$	**Formula:** $M_a(i)$
8	15	160	160	20	20	$S_e(8)$	$S_e(8)/8$
9	25	185	165	20.625	20.5	$S_a(8) - M_a(8) + V_9$	$S_a(9)/prev\_pow2(9)$

$$V_9 = 25 \qquad S_a(9) = S_a(8) - M_a(8) + V_9$$

$$S_a(9) = 160 - 20 + 25 = 165$$

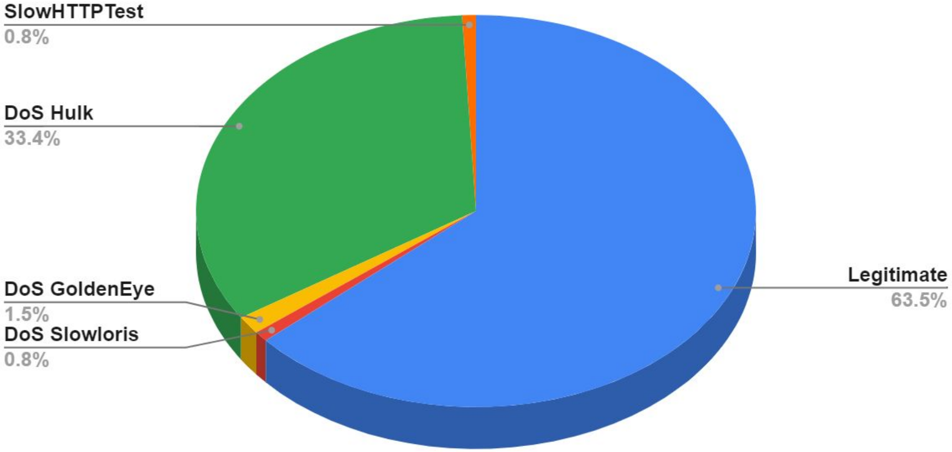
$$M_a(9) = \frac{S_a(9)}{prev\_pow2(9)} \quad M_a(9) = \frac{165}{8} = 20.625$$

Approximating Means



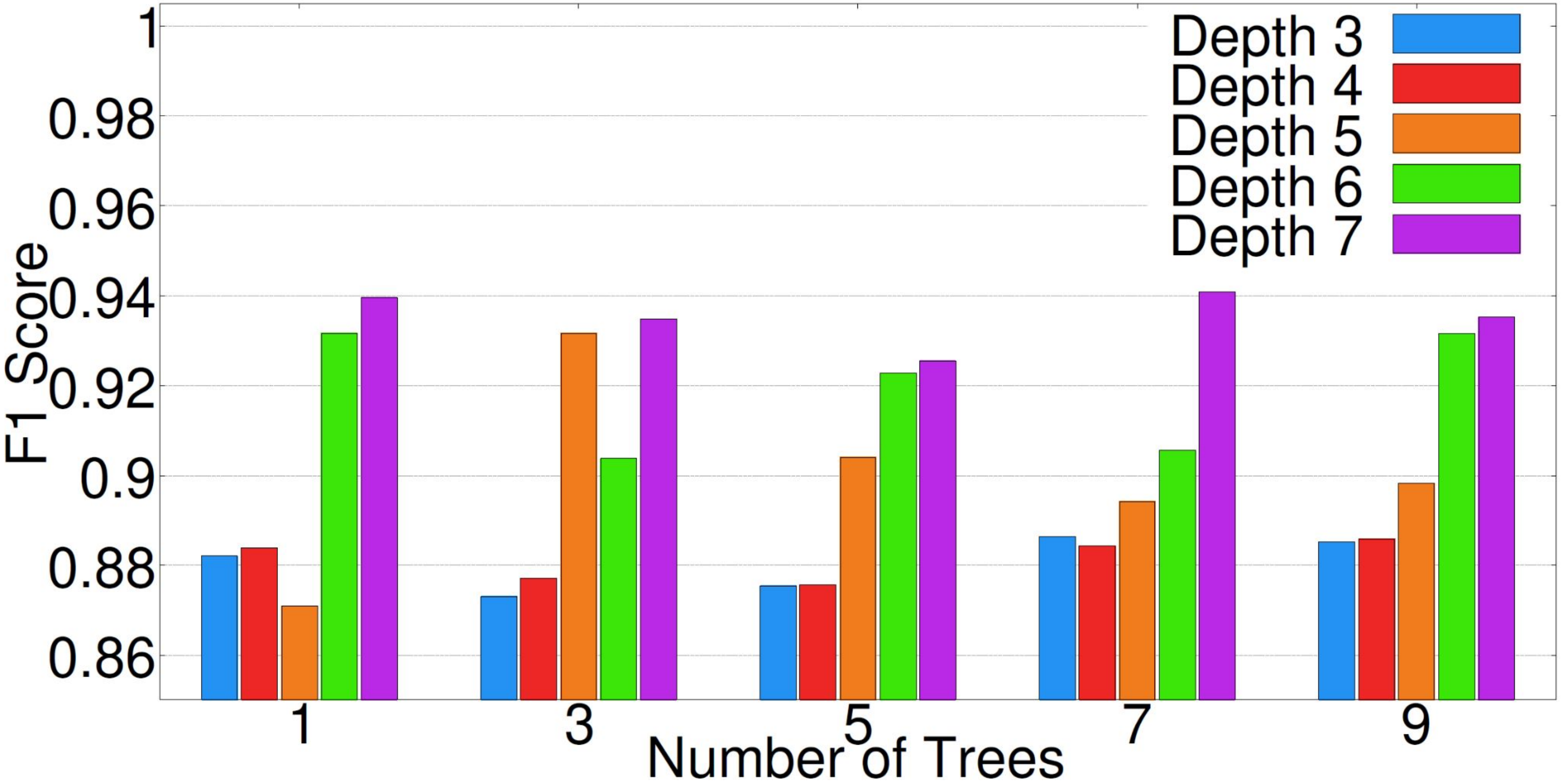
Evaluation - Dataset

- CICIDS 2017 Dataset
  - 692,703 flows
    - 440,031 legitimate (63.52%)
    - 5,796 DoS Slowloris
    - 5,499 DoS SlowHTTPTest
    - 231,073 DoS Hulk
    - 10,293 DoS GoldenEye
    - 11 Heartbleed
  - Binary division of classes
    - Legitimate
    - DoS (including all classes)



Bruno Coelho, Alberto Schaeffer-Filho

F1-Score for RF configurations



Conclusion

- BACKORDERS
- Classification of network flow in programmable data planes
  - Assisted by Machine Learning technique
- Maps nodes into match+action table entries
  - Sequential evaluation as opposed to recursive
- Extraction of features in the data plane
  - Approximation of means
- Proof-of-concept for utilizing ML in the data plane
  - Small forests with over 90% accuracy

Thank you for your time!

Bruno Coelho, Alberto Schaeffer-Filho
blcoelho@inf.ufrgs.br, alberto@inf.ufrgs.br

Federal University of Rio Grande do Sul (UFRGS), Brazil

INSTITUTO DE INFORMÁTICA UFRGS

INSTITUTO DE INFORMÁTICA UFRGS

Approximating Means

i	V_i	$S_e(i)$	$S_a(i)$	$M_a(i)$	**Mean**	**Formula:** $S_a(i)$	**Formula:** $M_a(i)$
8	15	160	160	20	20	$S_e(8)$	$S_e(8)/8$
9	25	185	165	20.625	20.5	$S_a(8) - M_a(8) + V_9$	$S_a(9)/prev\_pow2(9)$
10	10	195	154.375	19.29875	19.5	$S_a(9) - M_a(9) + V_{10}$	$S_a(10)/prev\_pow2(10)$

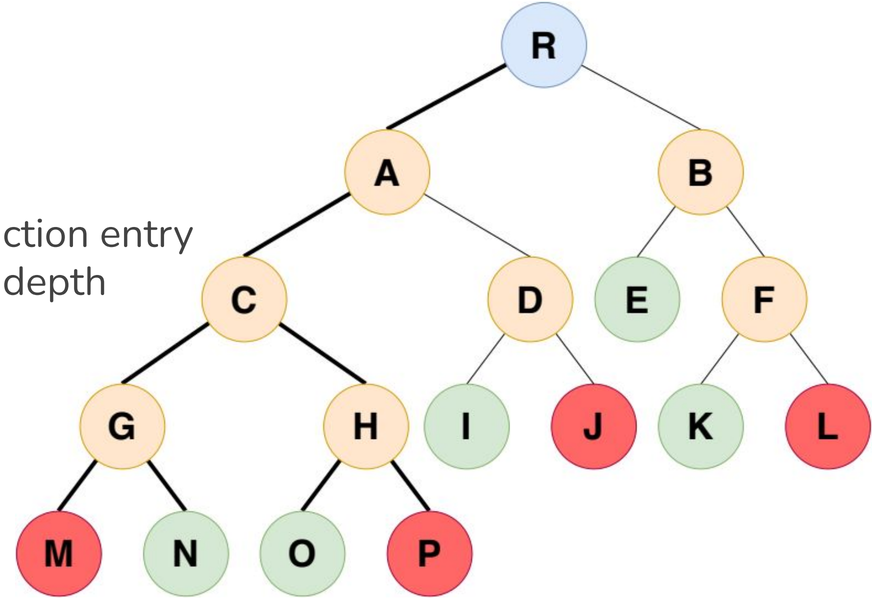
$$V_{10} = 10$$

$$S_a(10) = 165 - 20.625 + 10 = 154.375$$

$$M_a(10) = \frac{154.375}{8} = 19.296875$$

Scalability Analysis

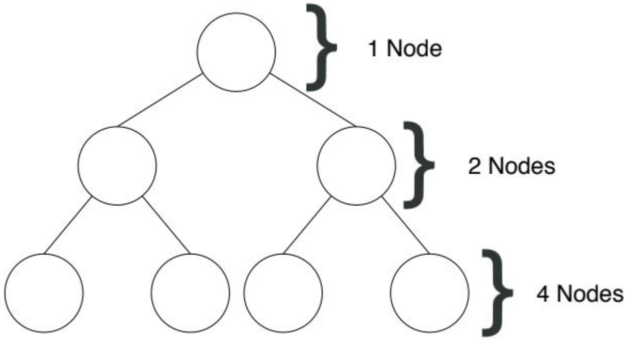
- Processing time is limited by maximum depth
  - *O(M)* per tree
  - *O(NM)* per forest
- Memory
  - Each node is mapped into a single match+action entry
  - Table entry number is limited by maximum depth
    - 1 layer = 1 node
    - 2 (full) layers = 3 nodes
    - 3 (full) layers = 7 nodes
  - $O(2^M)$ per tree
  - $O(N(2^M))$ per forest



Bruno Coelho, Alberto Schaeffer-Filho

Scalability Analysis

- Processing time is limited by maximum depth
  - *O(M)* per tree
  - *O(NM)* per forest
- Memory
  - Each node is mapped into a single match+action entry
  - Table entry number is limited by maximum depth
    - 1 layer = 1 node
    - 2 (full) layers = 3 nodes
    - 3 (full) layers = 7 nodes
  - $O(2^M)$ per tree
  - $O(N(2^M))$ per forest



Scalability Analysis

# Trees	Max. Depth	Comparisons/tree	Total comparisons	Memory/tree	Total memory
1	6	6	6	63	63
	7	7	7	127	127
3	5	5	15	31	93
	6	6	18	63	189
	7	7	21	127	381
5	5	5	25	31	155
	6	6	30	63	315
	7	7	35	127	635
9	6	6	54	63	567
	7	7	63	127	1143

Future Work

- Optimize memory utilized per feature
  - Current implementation may not scale for a high number of flows
- Include only the features that were selected by trees
  - Less memory utilization per flow
- Feature selection
  - Less registers
  - Lower depth