Traffic-aware Statistical Optimization of Firewall Packet Filtering

Ehab Al-Shaer and Will Marrero

NSF Grant No. CNS- 0716723

Work Objectives:

$$\text{Filtering Cost} = \sum_{i \in matched}^{n} Rate_i * Depth_{R_i} + \sum_{j \notin matched}^{n} Rate_j * Depth_{default\_deny}$$

- Optimizing the average filtering cost while maintaining guaranteed worst case performance using network traffic properties (traffic-aware)
- Creating rules dynamically to reject packets that match default-deny rule as early as possible without sacrificing the accept path performance.

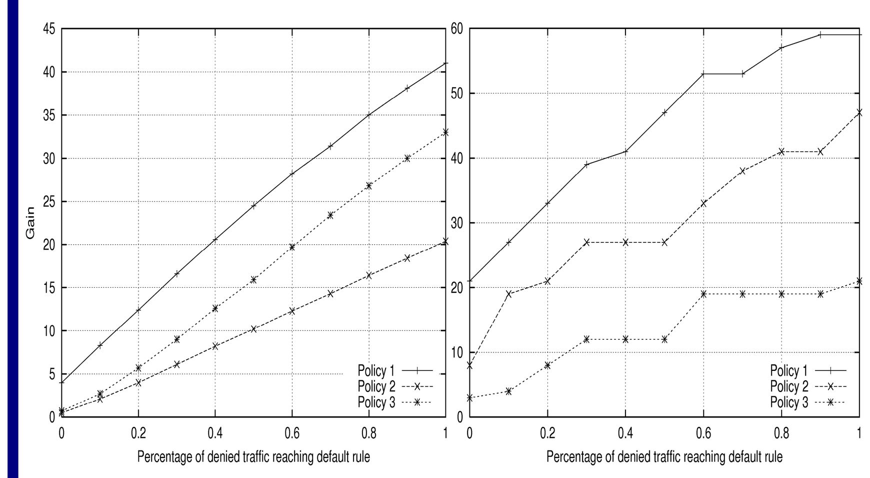
Early Rejection *(Reject Path)*:

- **Objectives:** (1) to create the *minimum* number of Early Rejection Rules dynamically that has a *maximum* discarding effect (covering Discard space), (2) to make RR adaptive to the recent discarded traffic (Dynamic rule selection)
- **The basic idea:** to add font-end rejection rules such that the overall average matching is decreased (min affect on accept packets) using set-cover approximation

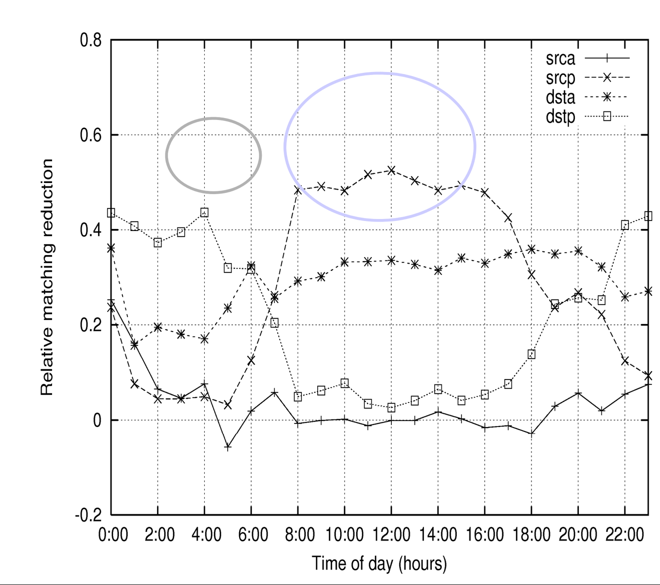
Example: (dst_port != 80) and (proto != 6) and (proto != 1) → reject
(saddr != 140.192.) and (dst_port != 22.) → reject

Evaluation Results:

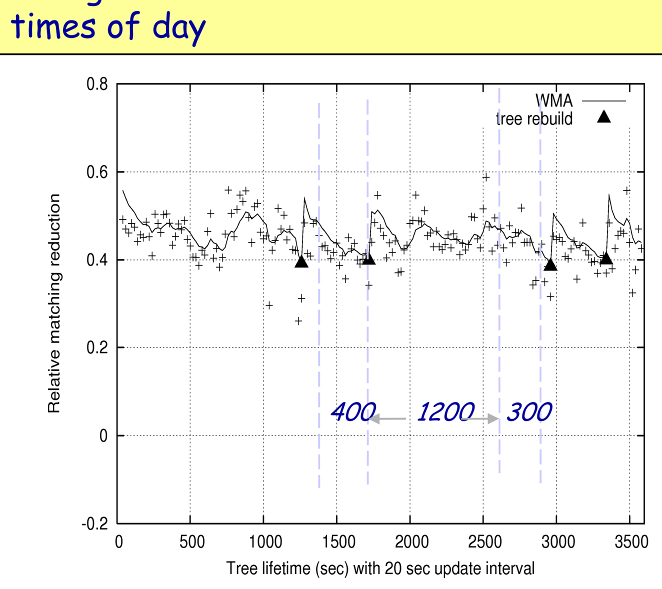
- **Early rejections:** Matching gain: 19%/25%; 50%/75%, and added RR rules is 4%-10%
- **Statistical Filtering:** Matching gain: up to 45% in busy hours, with 200s-400s update period The implementation of proposed techniques is simple and lightweight



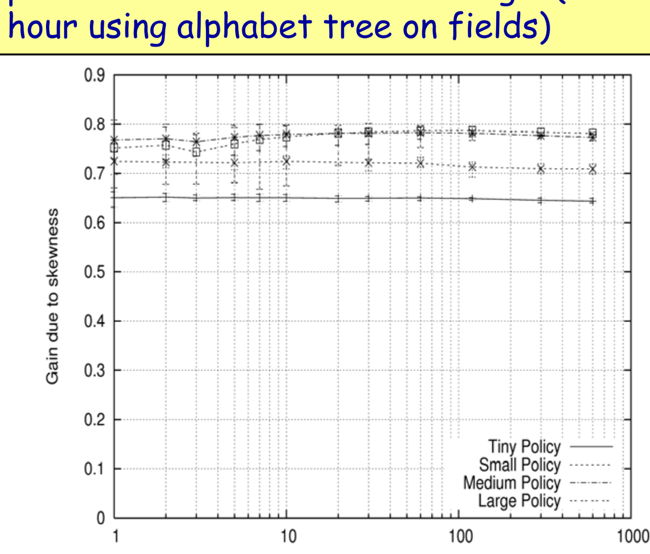
Early Rejection Reduction: 41% (Optimal is 50%)



Matching reduction for each field for different times of day

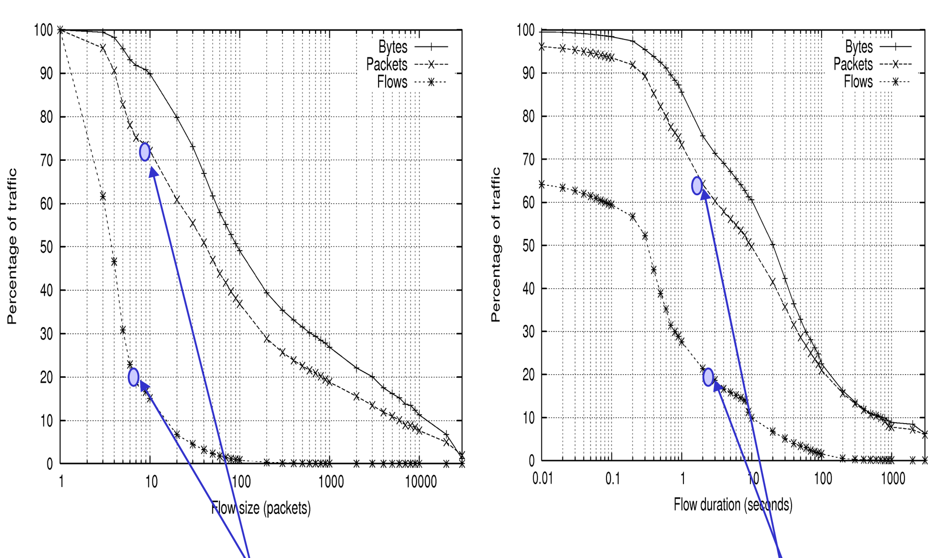


Triggered update of the search tree keeps the performance in the desired range. (over 1 hour using alphabet tree on fields)



Matching cost reduction by using segments-based Huffman Tree

"Locality of Matching" Traffic Property:



About 20% of the flows (of 10 packets or more) carry about 70% of the total traffic

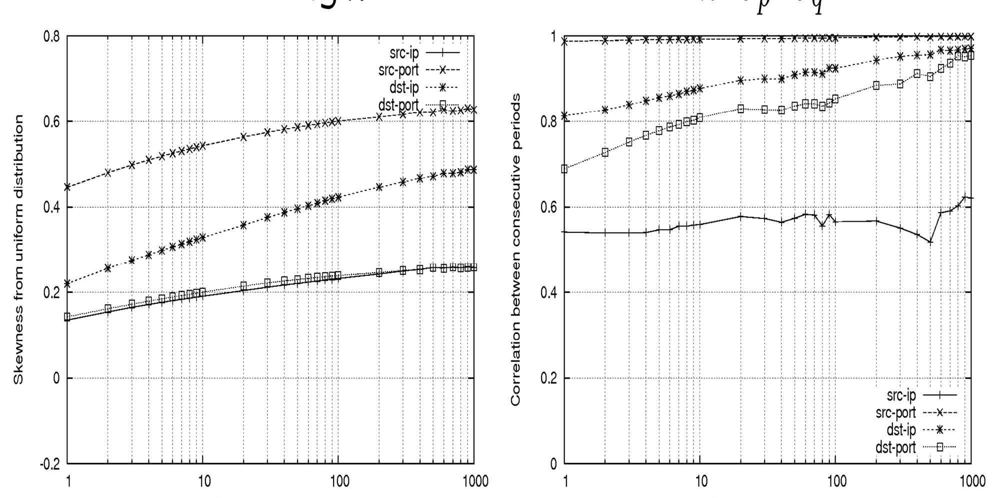
About 20% of the flows (that carry about 60% of the total traffic) last 5 seconds or more

Skewness:

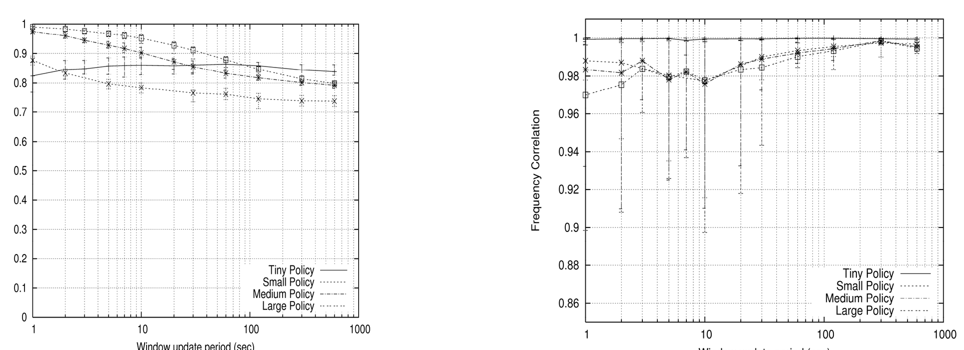
$$S_f = 1 - \frac{\sum_{i=1}^{n} p_i \lg p_i}{\lg n}$$

Autocorrelation

$$C_f = \frac{\sum_{i=1}^{n}(p_i - \mu_p)(q_i - \mu_q)}{n \cdot \sigma_p \cdot \sigma_q}$$



Skewness is an indication of the high frequency of few values for a particular field in the traffic, and high correlation indicates stability of this measure.



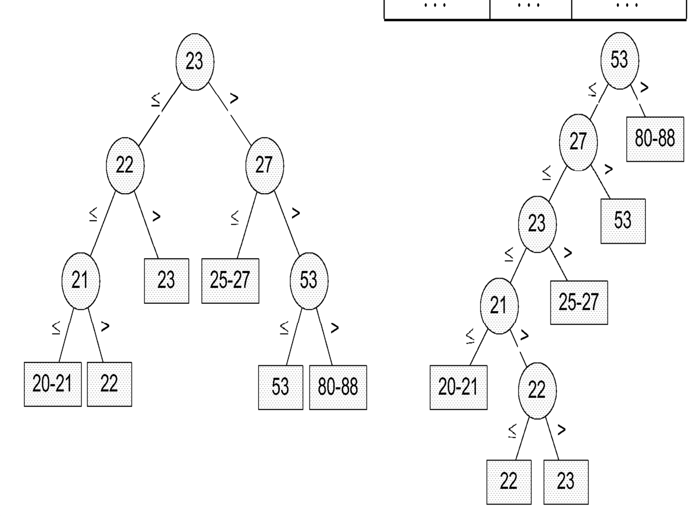
Same properties were observed in segments' distribution

Statistical Filtering *(Accept Path)*:

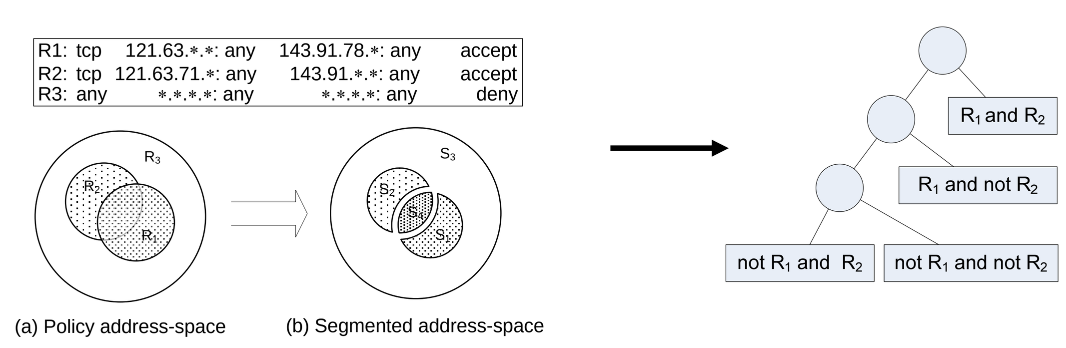
I. Field-based Alphabet Trees

Building filtering trees based on lower granularity (field values instead of rules) result in better search structure and overall performance.

Field	Value	Statistics	
dst_port	25-27	0.11	→ 3
dst_port	23	0.01	
dst_port	53	0.19	→ 2
dst_port	80-88	0.60	→ 1
dst_port	20-21	0.08	
dst_port	22	0.01	
...	...	...	



II. Segment-based Huffman Trees



Segments is a finer level of granularity, allowing us to build highly tuned filtering structures. Skewness is more evident over segments than over rules and fields.

III. Segment-based MRU Lists

Searching through segments on MRU-basis proved to be very simple to implement with no periodic maintenance cost, but with a more varying processing time, and less strict worst case bounds.