Filtering Sources of Unwanted Traffic

(or: dealing with good, bad and ugly IP addresses)

F.Soldo, K. El Defrawy, A. Markopoulou
UC Irvine
B. Krishnamurthy, K. van der Merwe
AT&T Labs-Research

Outline

- Background/Motivation
- Filtering Algorithms
- Conclusion

Motivation

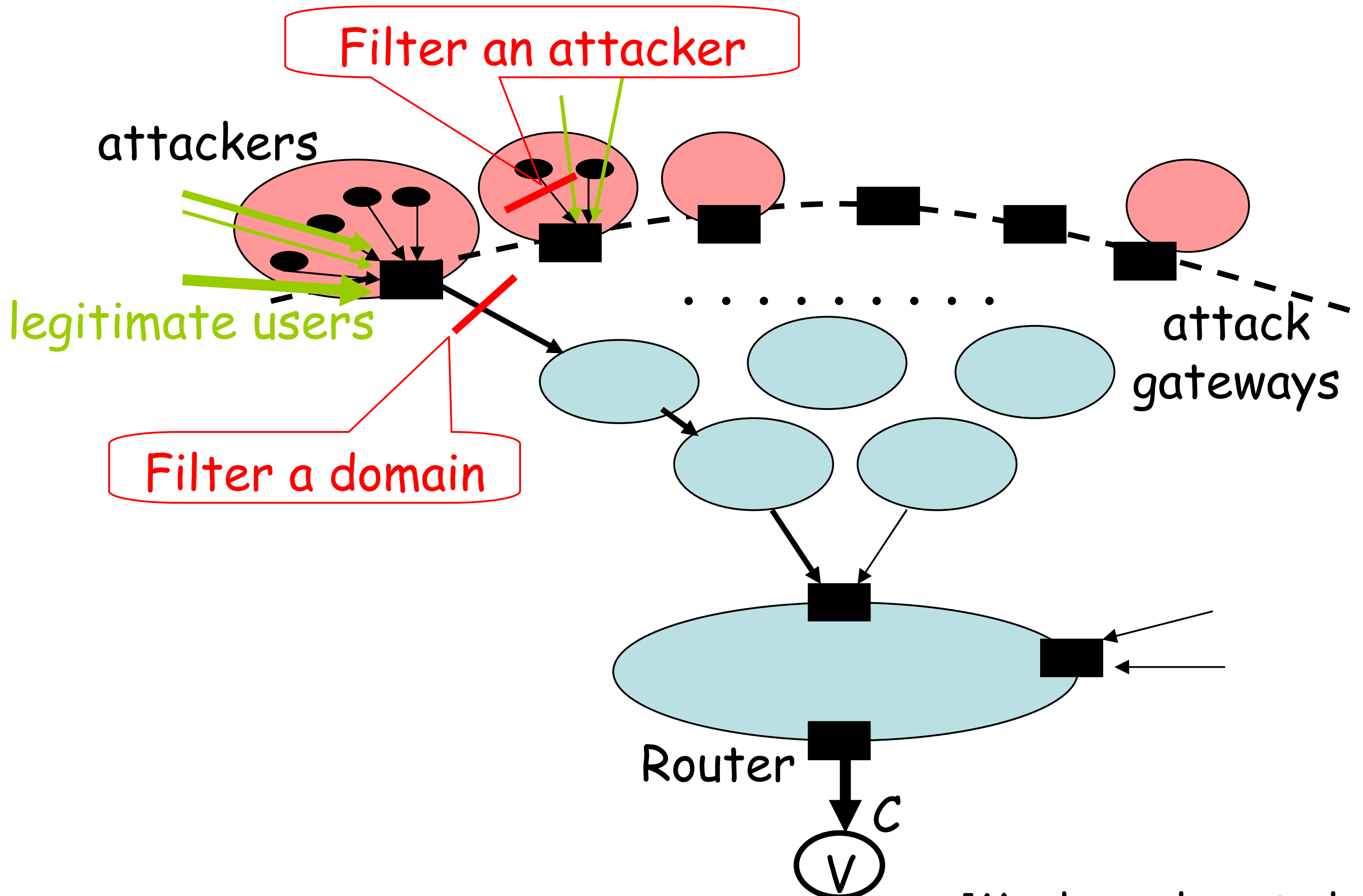
- Unwanted traffic on the Internet
  - denial-of-service attacks
  - spam
  - port scanning
  - etc..
- "Internet background radiation"
  - [Barford et al. PAM 06]

Part of the Solution
filtering at the routers

- Access Control Lists (ACLs)
  - match a packet header against rules, e.g. source and destination IP addresses.
- Filters are an expensive resource
  - at most 256K filters per TCAM chip
  - each victim gets only a few 1000s of filters
- There are more attackers than filters
  - An attack can consist of millions of flows

A Filtering Example

tradeoff: filters vs. collateral damage

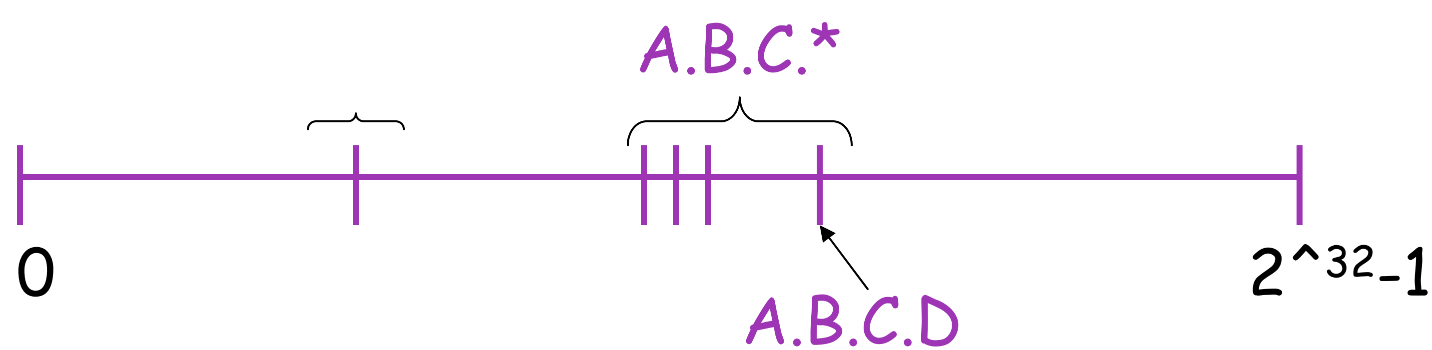


[Markopoulou et al, ITA 07]

Key observation 1

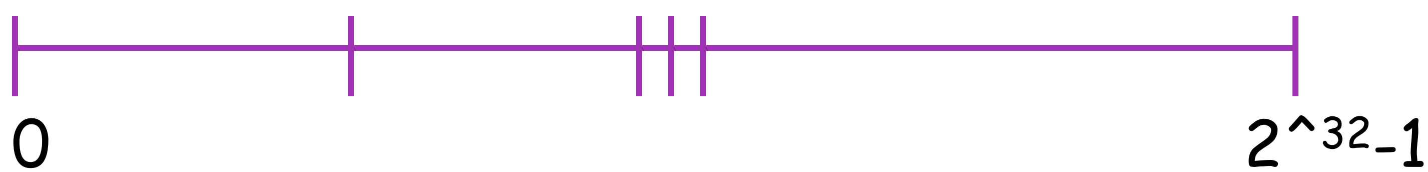
Source based filtering: 1-dim problem

- Any 32-bit source IP address A.B.C.D can be mapped to an integer in [0, 2^32-1]
- Blacklists report "bad" source IPs
- Aggregate ranges of nearby IP sources into a single filtering rule (e.g. prefix).



Key observation 2
"Bad" Source IPs are clustered

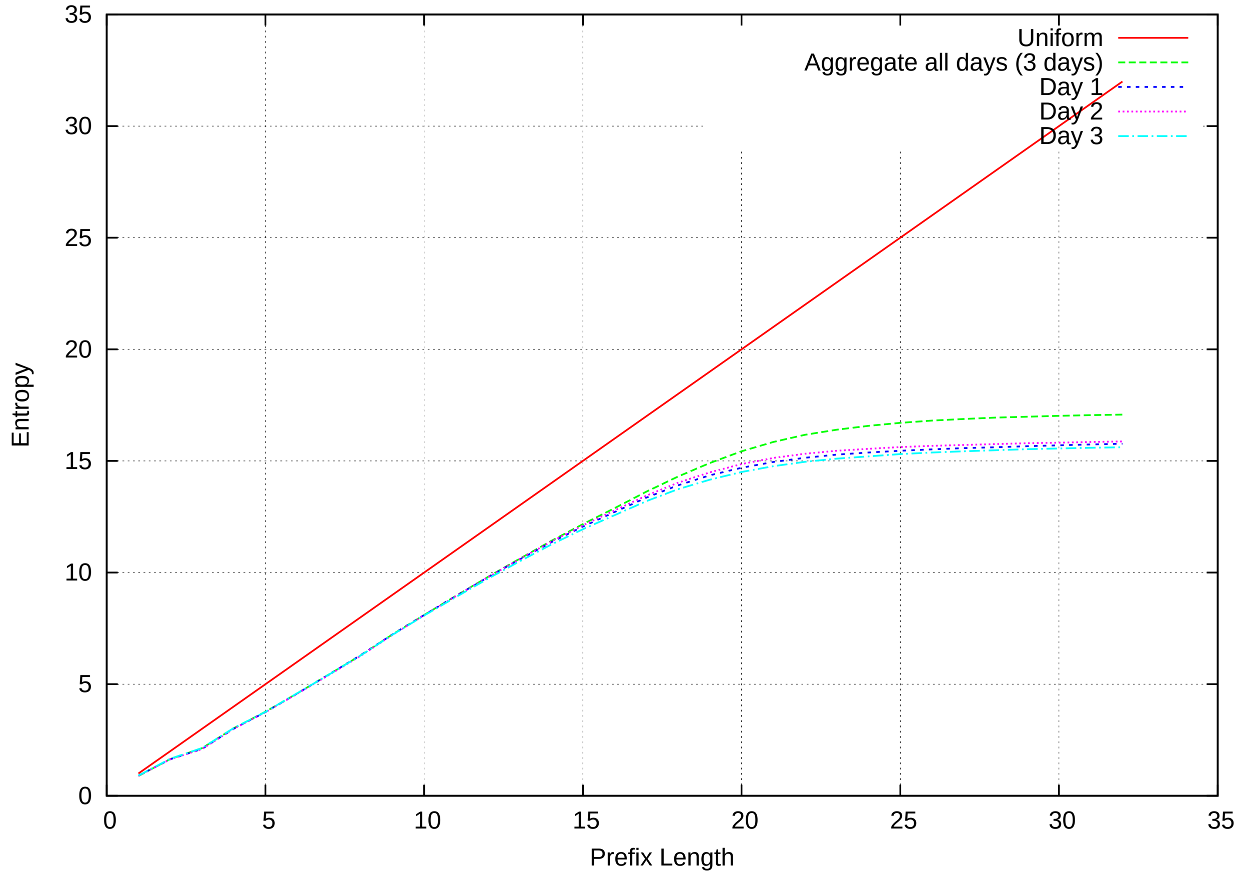
- Spatial and Temporal Clustering
  - Barford et al.,"A model for source addresses of Internet background radiation", [PAM'06]
  - Collins et al., "Using uncleanliness to predict future botnet addersses", [IMC 07]
  - Chen and Ji, "Measuring network-aware worm spreading capabilities', [INFOCOM 07]
- And there is a reason for that..



Clustering Evidence

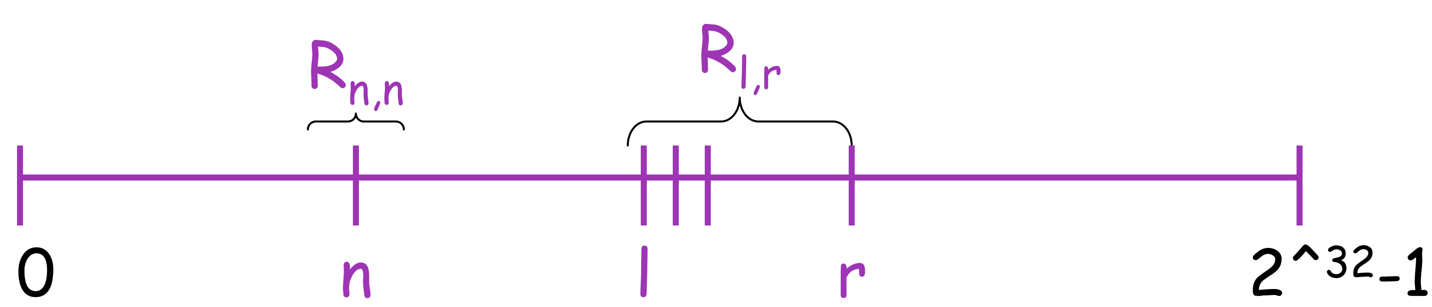
from DShield.org data

- Look at distribution of (N) bad addresses to intervals
- Prefix length l, i=1,…2^l, /l subnets, each with prob. $p_i=N_i/N$



Goal

- Design a family of filtering algorithms that
  - take as input a blacklist of “bad” addresses
  - produce compact filtering rules
  - to maximize the number of bad addresses filtered and minimize collateral damage



Outline

- Background/Motivation
- Filtering Algorithms
- Conclusion

Filtering Algorithms
Overview

<table>
<tr><td colspan="2"></td><td colspan="2">Input blacklist</td></tr>
<tr><td colspan="2"></td><td>A single (static) blacklist</td><td>Time-varying</td></tr>
<tr><td rowspan="2">filter all bad IPs?</td><td>yes</td><td>P1: FILTER-ALL-STATIC</td><td>P3: FILTER-ALL-DYNAMIC</td></tr>
<tr><td>no</td><td>P2: FILTER-SOME-STATIC</td><td>P4: FILTER-SOME-DYNAMIC</td></tr>
</table>

P1: FILTER-ALL-STATIC
Problem Statement

- Given: a blacklist and F_{max} filters
- choose: filters $R_{l,r}$
- so as to: filter *all* bad addresses
  and minimize collateral damage $C_{l,r}$

$$\min \sum_{l \leq r} \tilde{C}_{l,r} R_{l,r}$$

$$\sum_{l \leq r} R_{l,r} \leq F_{max}$$

$$\sum_{l \leq i \leq r} R_{l,r} \geq 1 \ \forall i \in \{b_1, b_2, \ldots, b_N\}$$

$$R_{l,r} \in \{0, 1\} \ \forall l, r \in \{1, 2, \ldots, m\}$$

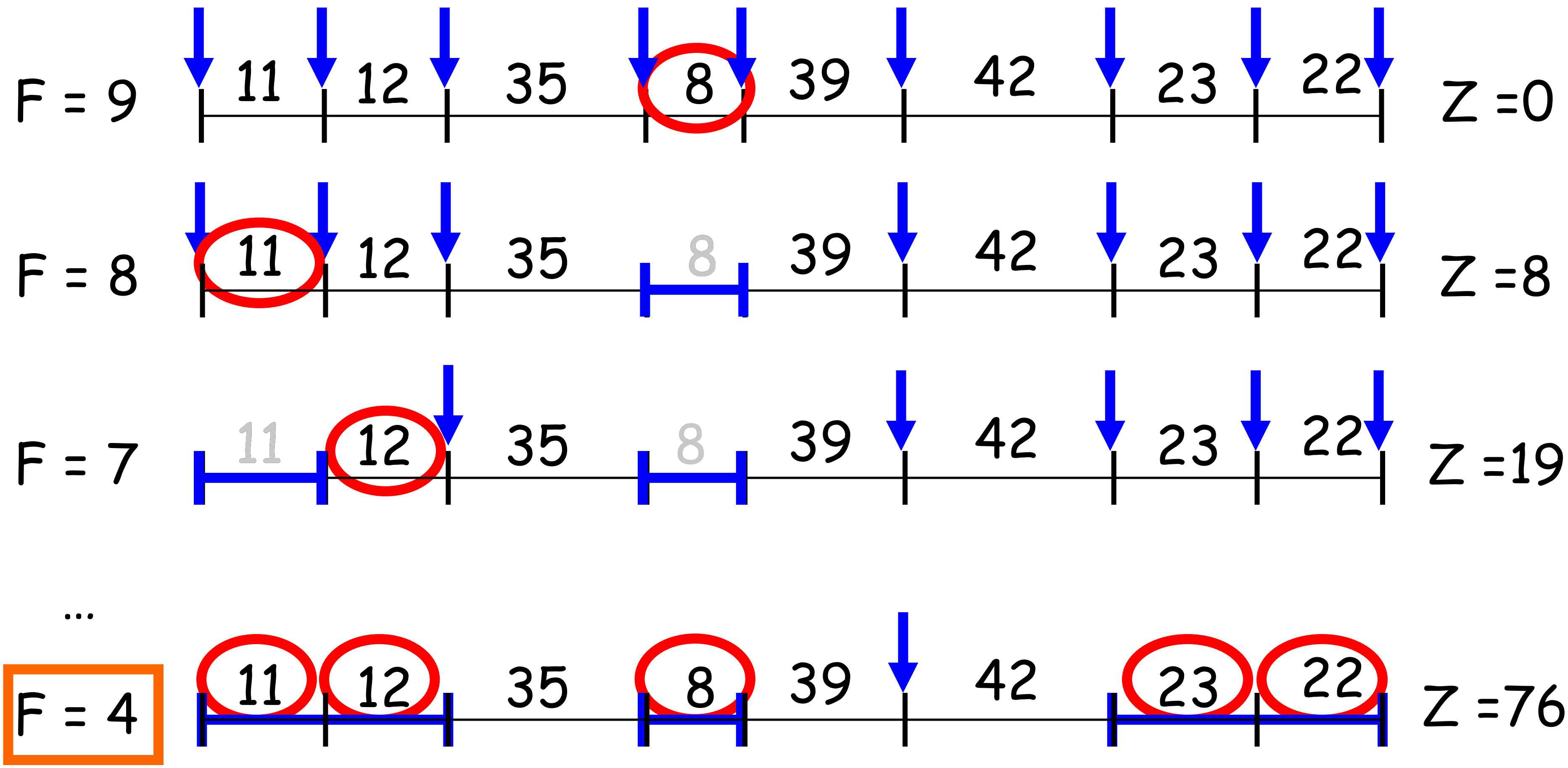
P1: FILTER-ALL-STATIC
Greedy Algorithm

- Let F=N
  - assign one filter to each bad address
- While $F>F_{max}$
  - make the following greedy decision:
    - pick the two “closest” bad IPs/intervals
    - remove a filter and extend an existing one to cover this interval
  - decrease F=F-1

P1: FILTER-ALL-STATIC

Example of running Greedy

$F_{max} = 4$, $N = 9$



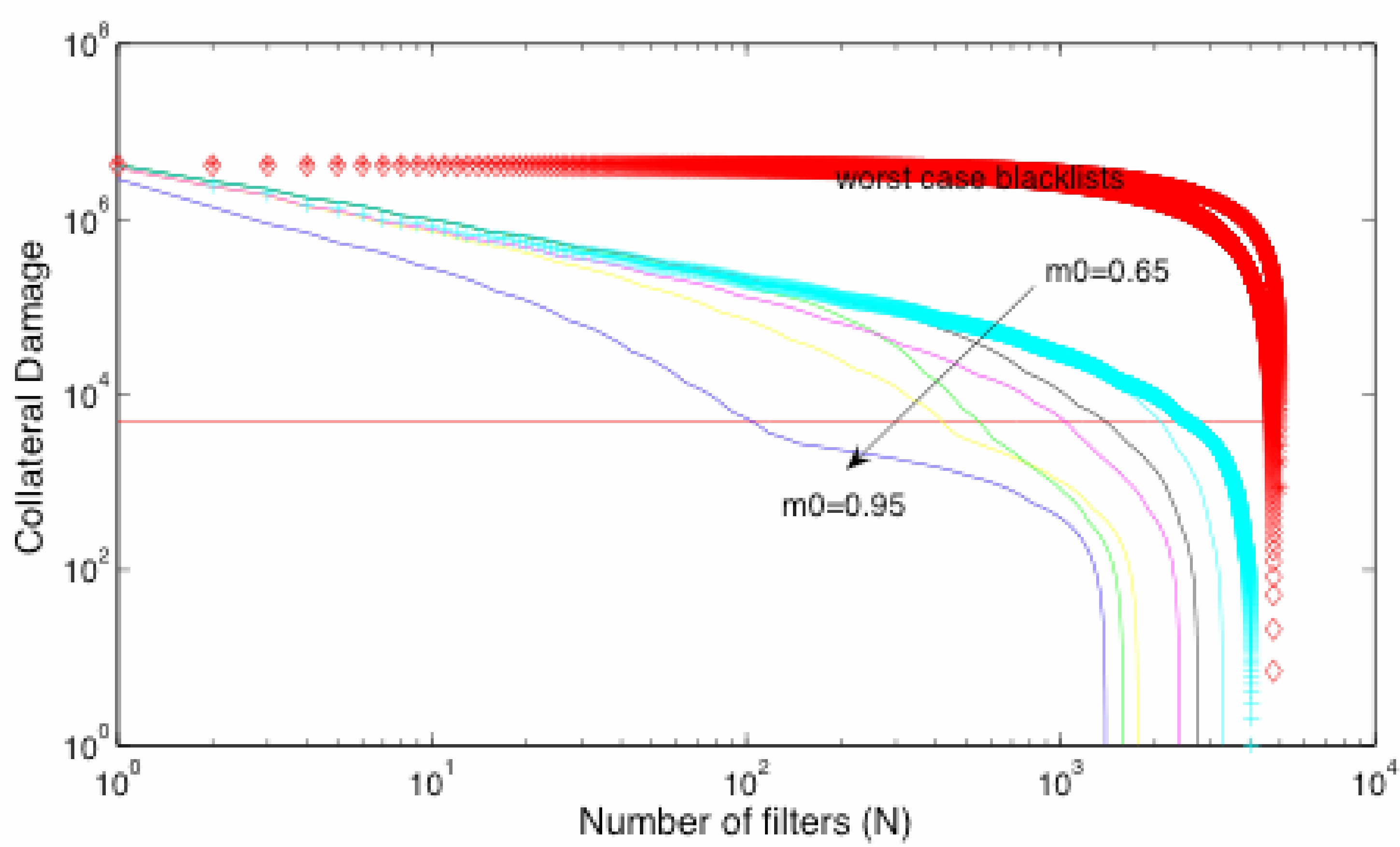
P1: FILTER-ALL-STATIC

Greedy Algorithm: Properties

- Optimality
  - the greedy algorithm computes the optimal solution to P1
- Complexity
  - sorting $O(N\,log(N))$ and $N-F_{max}$ steps

P1: FILTER-ALL-STATIC
Simulations

- Address structure generated using a multifractal cantor measure
  - [Kohler *et al. TON'06*, Barford *et al. PAM'06*]



P2: FILTER-SOME-STATIC

Problem Statement

- Given: a blacklist, weight w_i of address i, and F_{max} filters
- choose: filters $R_{l,r}$
- so as to: filter *some* bad addresses and the total weight (which is the sum of collateral damage + the cost of unfiltered bad addresses)

$$\min \sum_{l \le r} \sum_{l \le i \le r} w_i R_{l,r}$$

$$\sum_{l \le r} R_{l,r} \le F_{max}$$

$$\sum_{i \le l \text{ or } j \le r} R_{i,j} \le 1 \quad \forall l, r \in \{1, 2, \ldots, N\}$$

$$R_{l,r} \in \{0, 1\} \quad \forall l, r \in \{1, 2, \ldots, N\}$$

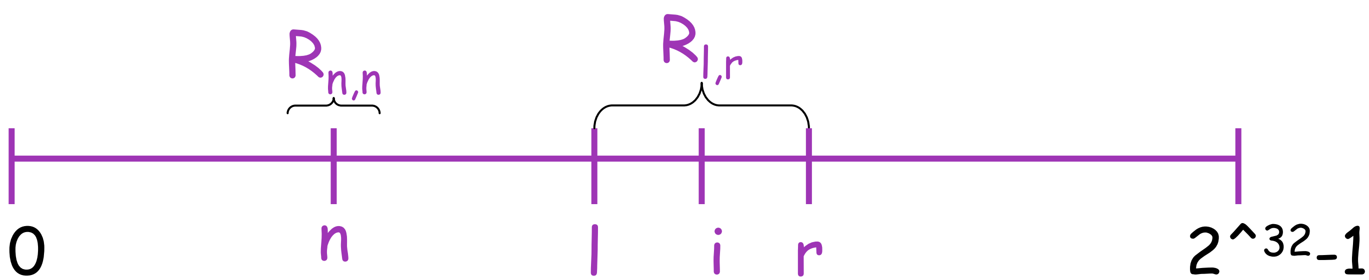
P2: FILTER-SOME-STATIC

Problem Statement

$$\min \sum_{l\le r} \sum_{l\le i\le r} w_i R_{l,r}$$

$$\Downarrow$$

$$\min \sum_{l\le r}\left(\sum_{l\le i\le r,} w_i \mathbb{I}_{\mathcal{G}}(i) + \sum_{l\le i\le r} w_i \mathbb{I}_{\mathcal{B}}(i)\right) R_{l,r}$$



P2: FILTER-SOME-STATIC
Problem Statement

$$\min \sum_{l \le r} \sum_{l \le i \le r} w_i R_{l,r}$$

$$\Downarrow$$

$$\min \sum_{l \le r} \left(\sum_{l \le i \le r,} w_i \mathbb{I}_{\mathcal{G}}(i) + \sum_{l \le i \le r} w_i \mathbb{I}_{\mathcal{B}}(i) \right) R_{l,r}$$

- Assignment of weights W_i is the operator's knob:
  - $W_i > 0$ (good source i), $W_i < 0$ (bad source i), $W_i = 0$ (indifferent)
  - $W_g = 1$ for all good addresses g, $W_b = -W$ for all bad addresses b
  - $W_g = 1$ for all good, $W_b \rightarrow -\infty$ for all bad: filter all bad (Problem P1)

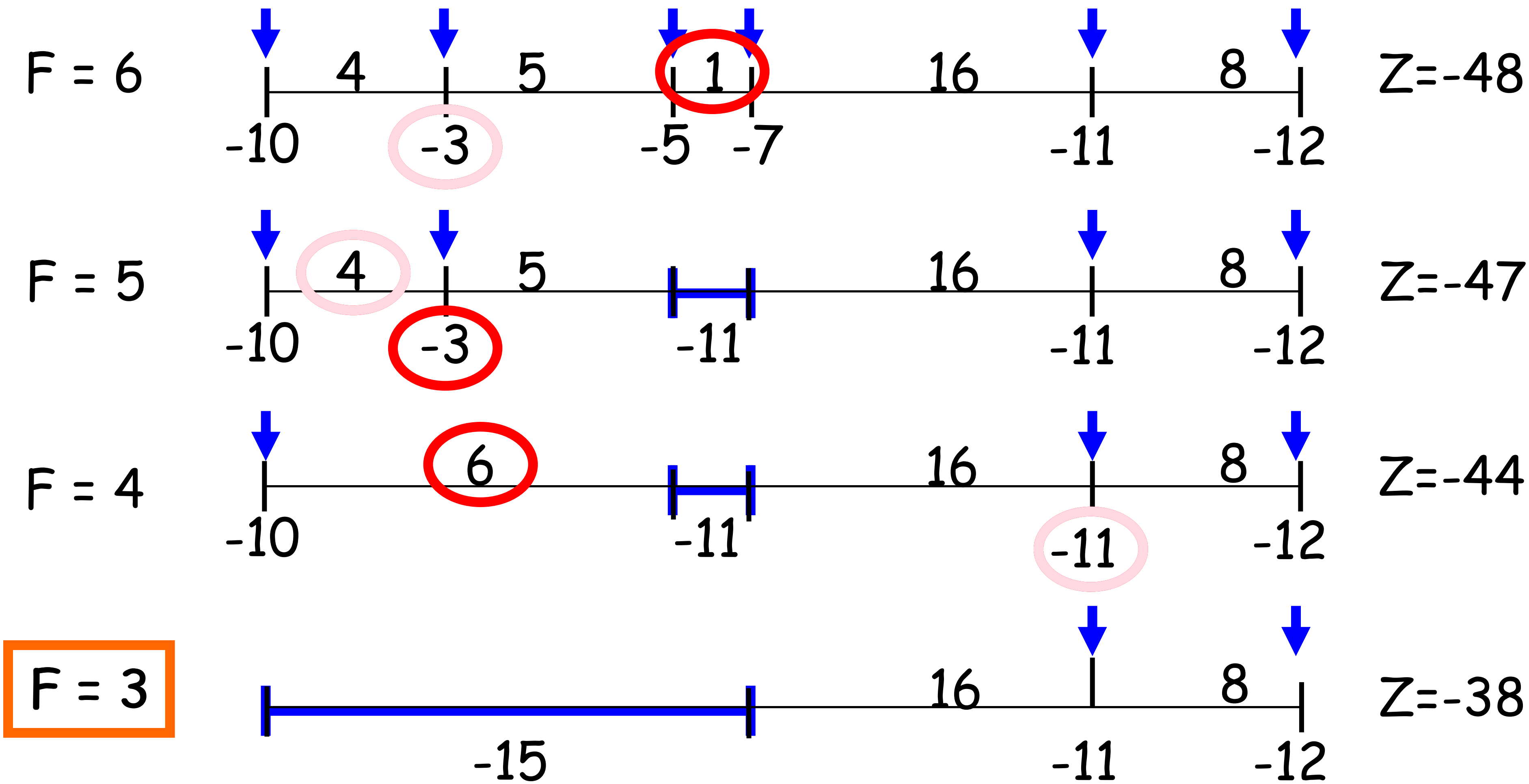
P2: FILTER-SOME-STATIC
Greedy Algorithm

- Let F=N
  - assign one filter to each bad address
- While $F>F_{max}$
  - make the following greedy decision:
    - merge the two "closest" filters,
    - or release a filter,
    - whichever causes the smallest increase in objective Z
  - decrease F=F-1

P2: FILTER-SOME-STATIC

Example of running Greedy

$F_{max} = 3$, $N = 6$



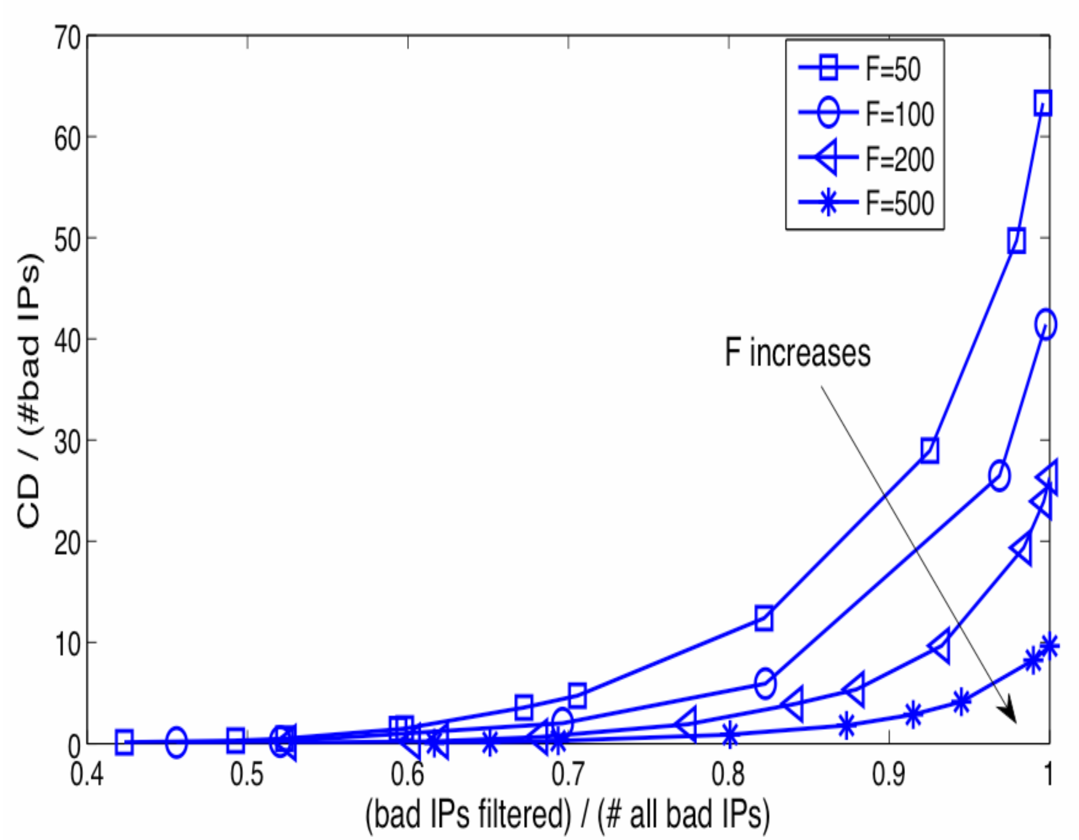
P2: FILTER-SOME-STATIC

Greedy Algorithm: Properties

- Optimality
  - the greedy algorithm computes the optimal solution to P2
- Complexity
  - sorting O(N*log*(N)) and $N\text{-}F_{max}$ steps

P2: FILTER-ALL-STATIC
Simulations

- Addresses from the same multifractal distribution



The Time-Varying Case

- Source IPs appear/disappear/reappear in a blacklist over time
- New input: A set of blacklists collected at different times $\{BL_{T0}, BL_{T1},\ldots BL_{Ti}, \ldots\}$

Problem Statement

- P3 (P4)
  - Given: a set of blacklists $\{BL_{T0}, BL_{T1},\ldots\}$ collected at different times, and F_{max} filters
  - Goal: find set of filter rules $\{S_{T0}, S_{T1},\ldots\}$ s.t. S_{Ti} solves P1 (P2) for blacklist BL_{Ti} at all times

- Solution
  - run P1(P2) from scratch at every time T_i
  - …or exploit temporal correlation and just update filtering as needed

P3: FILTER-ALL-DYNAMIC
Greedy Algorithm

- At time T_0
  - Run greedy for BL_{T0}
  - Store a sorted list of distances

- At time T_i
  - Upon arrival or departure of addresses, update sorted list of distances
    - [e.g. one new arrival, 2 removals]
  - place filters to the pairs of addresses with the *N-F* shortest distances.
    - [e.g.: no change, remove 1 – add 1, shrink 1 – extend 1]

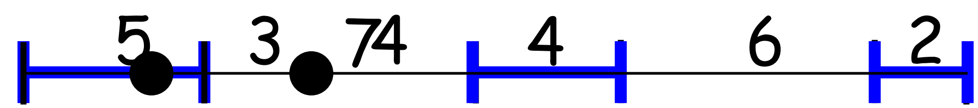
P3: FILTER-ALL-DYNAMIC

Example of new address appearing

$F_{max} = 3$

$N = 6$

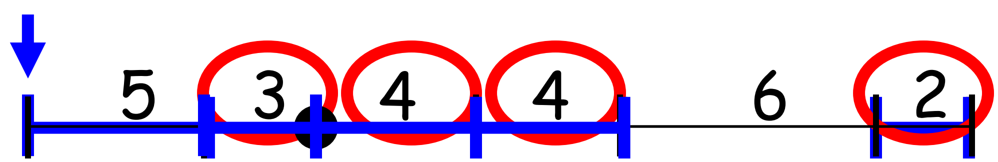
$N - F_{max} = 3$



$F_{max} = 3$

$N = 7$

$N - F_{max} = 4$



Outline

- Background/Motivation
- Filtering Algorithms
- Conclusion

Conclusion

- Summary
  - Formulated a family of filtering problems
  - Designed greedy optimal algorithms
- Ongoing work
  - Prefix-based filtering rules
  - Characterization of real blacklists

Thank you!

athina@uci.edu
http://aegean.eng.uci.edu/