Filtering Malicious IP Sources

Models and Algorithms

Fabio Soldo, Athina Markopoulou

UC Irvine

Parts of this talk are joint work with K.Argyraki @EPFL
and B.Krishnamurhty, J. van der Merwe @ AT&T

IPAM Workshop II: Applications of Internet MRA to Cyber-Security
UCLA, Oct.13-17 2008

Overview

- Problem Overview and Motivation
- Filtering Algorithms
- Conclusion

Context

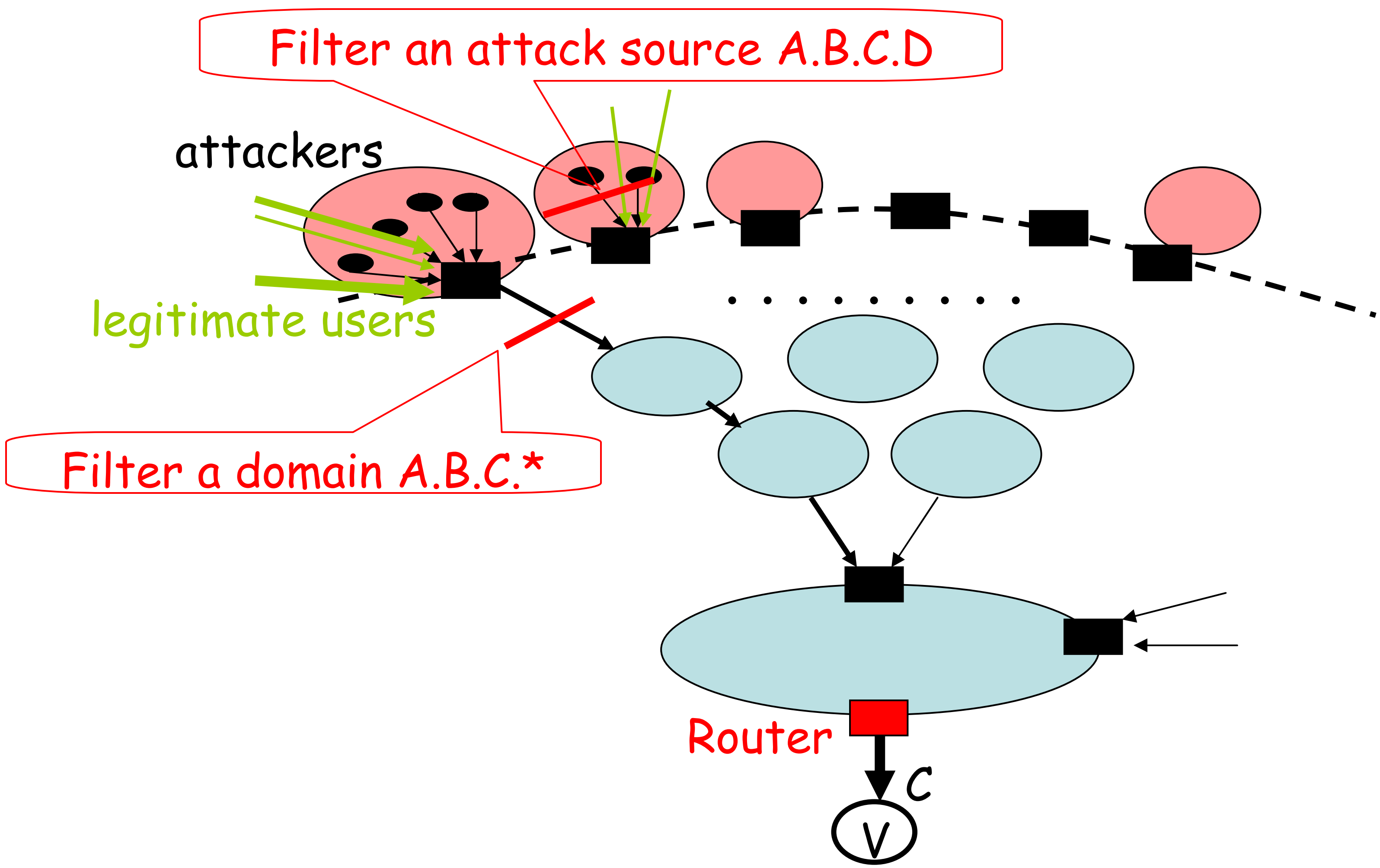
- Problem: Malicious IP Traffic
  - denial-of-service attacks
  - port scanning
  - spam
  - ...
- Solution requires many components
  - Detection of malicious traffic
  - Action: filtering, capabilities…
  - Anti-spoofing, accountability
  - …

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
  - filter: ACL that denies access to a source IP/prefix
- Filters implemented in TCAM
  - can keep up with high speeds
  - are a limited resource
    - ~tens of thousands per router, less per victim
- There are less filters than attack sources

Filter Selection at a Single Router

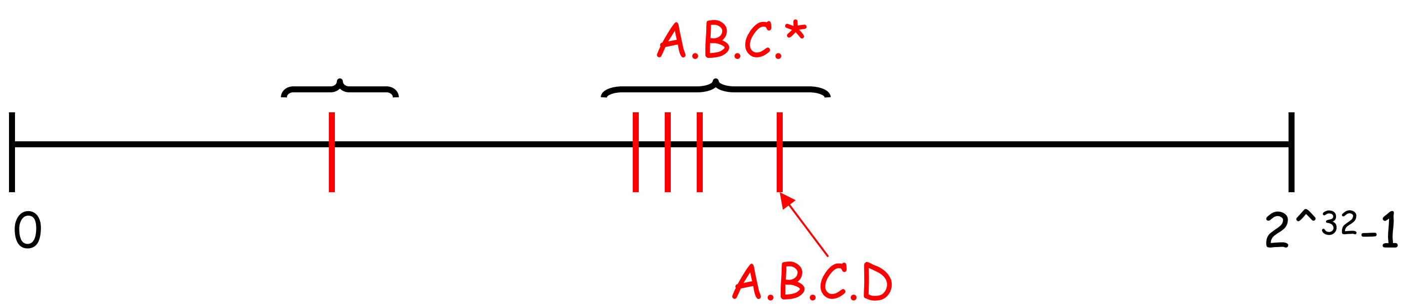
tradeoff: number of filters vs. collateral damage



Our Goal: Filter Selection
as a resource allocation problem

Design a family of filtering algorithms that:

- take as input:
  - a blacklist of malicious sources
    - and possibly a whitelist of legitimate sources
  - a constraint on the number of filters Fmax
    - and possibly other constraints, e.g., link capacities
  - the operator's policy
- select a compact set of filtering rules
  - so as to optimize the operator's objective
    - (filter as many malicious and as few legitimate sources)
  - subject to the constraints



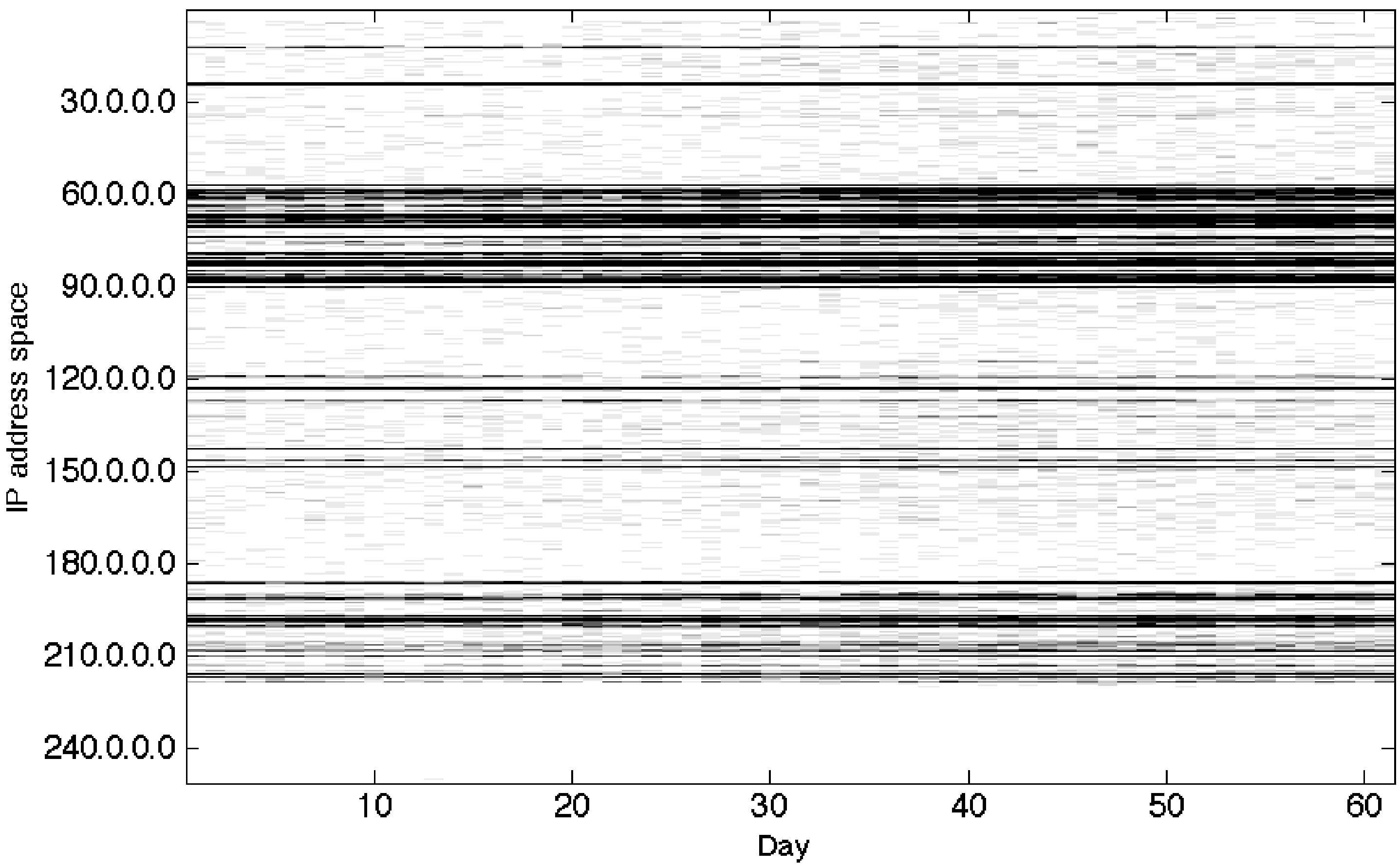
Observation #1

Malicious Source IPs are clustered

- Spatial & Temporal Clustering of Malicious IP Sources
  - 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, Ji, "Measuring network-aware worm spreading capabilities', [INFOCOM 07]
  - Chen, Ji, Barford, "Spatial-Temporal Characteristics of Internet Malicious Sources", [Infocom Miniconf. 2008]
  - Ramachandran, Feamster, "Understanding the Network-Level Behavior of Spammers", [SIGCOMM 2006].
  - ...

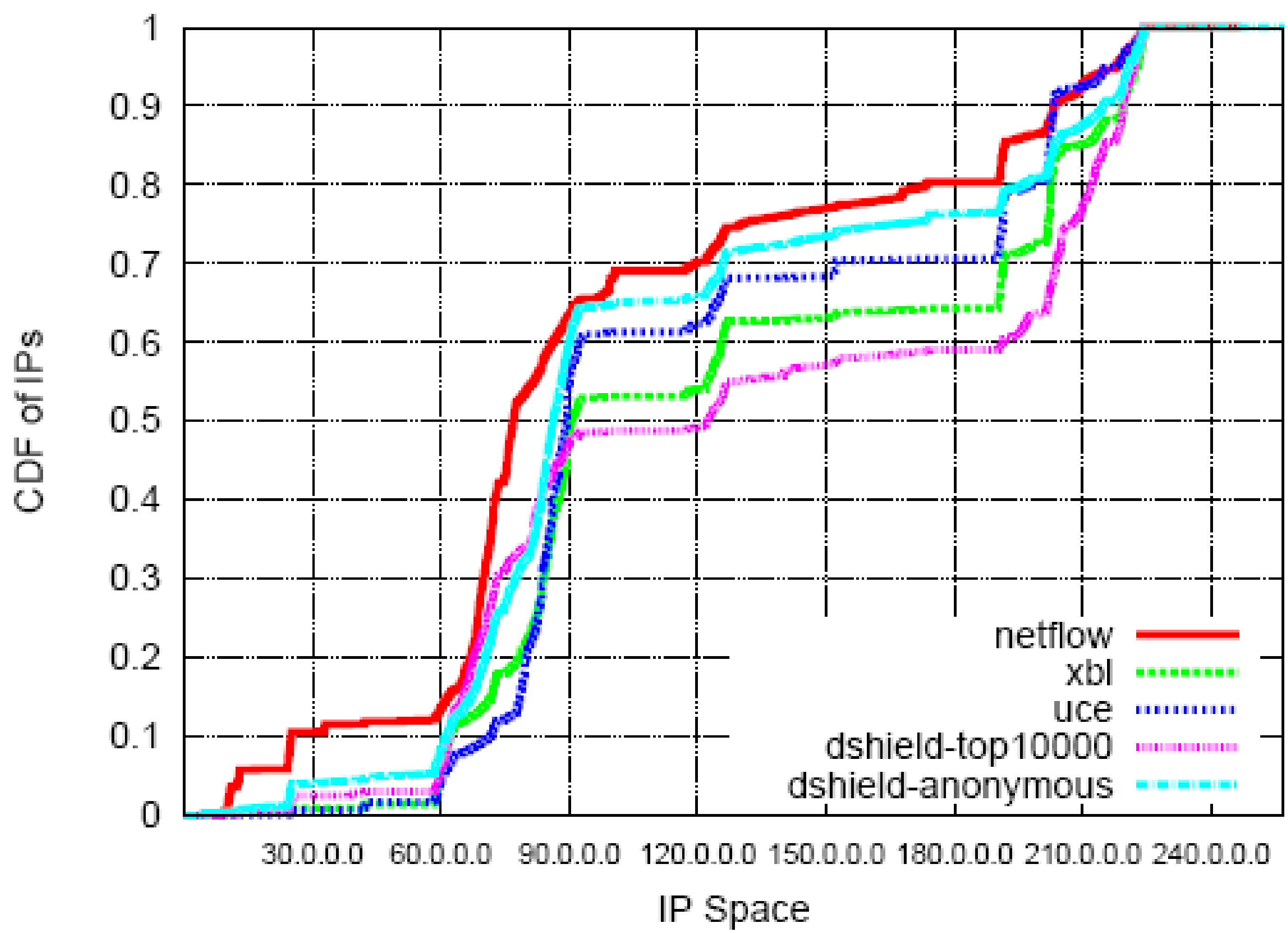
Observation #1

Evidence from DShield.org data



Observation #1

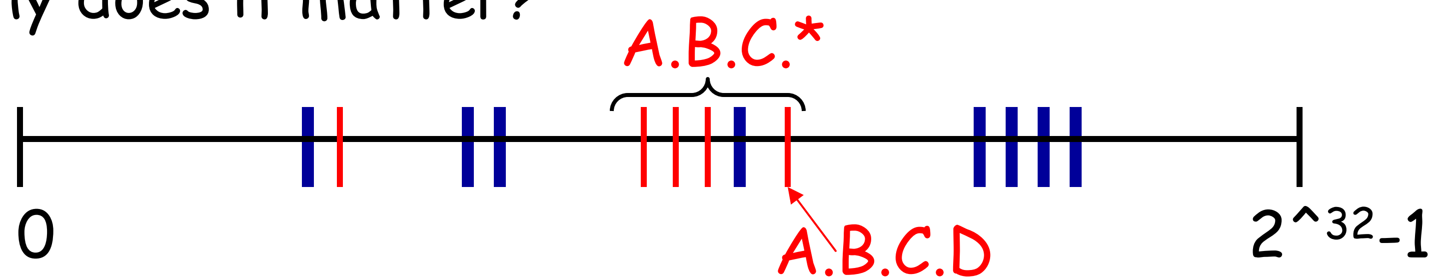
Evidence from DShield.org, Spamhaus XBL, UCE-protect



Observation #2

The distribution of good and bad sources is different

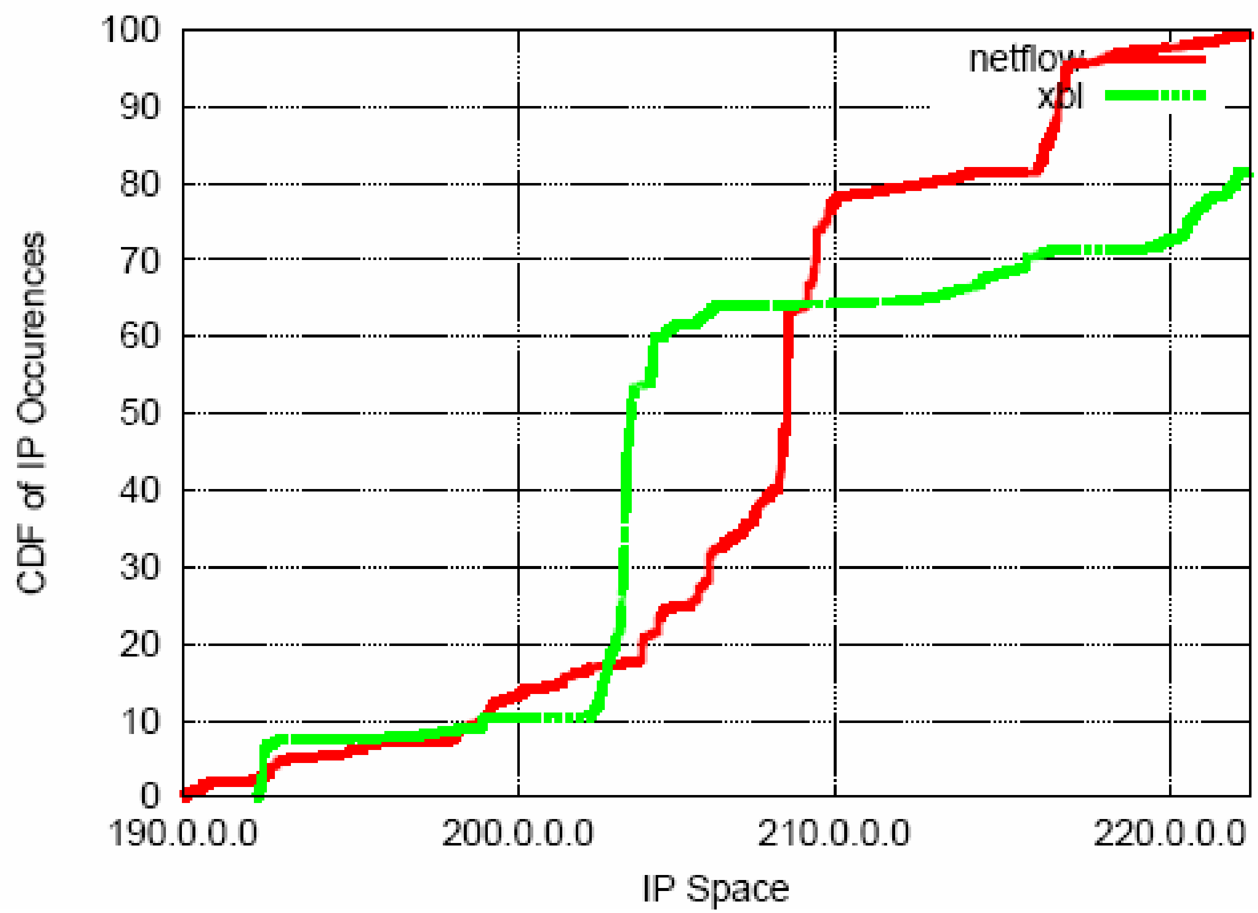
- Why does it matter?



- Often used hypothesis:
  - to do pre-filtering before traffic scrubbing
    - [Pack, Yoon, Collins, Estan, "On Filtering of DDoS Attacks Based on Source Address Prefixes", *SecureComm 06*]
  - to distinguish good vs. bad communities-of -interest (CoI)
    - [Vervaik, Spatscheck, van der Merwe, Snoeren, "PRIMED: A Community-of-Interest-Based DDoS Mitigation System", SIGCOMM LSAD 2006]
  - to distinguish spam from legitimate email
    - [A. Ramachandran, N. Feamster, S. Vempala, "Filtering Spam with Behavioral Blacklisting", CCS 2007]

Observation #2

Evidence



The Filter Selection Problem

as a resource allocation problem

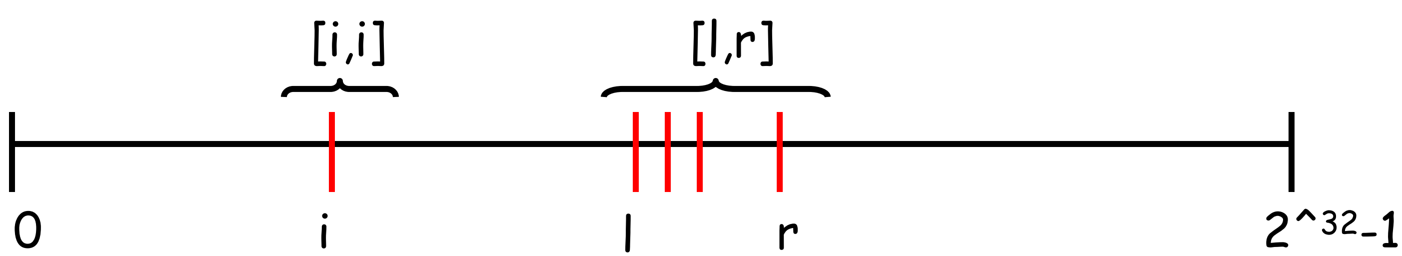
Design a family of filtering algorithms that:

- take as input:
  - a blacklist of malicious sources
    - [and possibly a whitelist of legitimate sources]
  - a constraint on the number of filters Fmax
    - [and possibly other constraints, e.g. link capacities]
  - the operator's policy
    - [weights indicating the importance of good/bad addresses]
- select a compact set of filtering rules
  - so as to optimize the operator's objective
    - (filter as many malicious and as few legitimate sources)
  - subject to the constraints

Filter Selection

Notation

- i: source IP address in [0, 2^{32}-1]
- w_i weight assigned to IP address *i* :
  - amount of flow sent
  - "importance" assigned by the operator
    - e.g. monetary loss (gain) in filtering out that address
- $x_{[l,r]} \in \{0,1\}$: decision variable
  - indicates whether we filter out IP sources in the range [l,r]



Filter Selection

as a Knapsack Problem

Different "profits"

$$\min \sum_{[l,r]\in\mathcal{D}} \sum_{i\in[l,r]} w_i \cdot x_{[l,r]}$$

Same "weights"

$$\text{s.t.} \sum_{[l,r]} 1 \, x_{[l,r]} \leq F_{max}$$

Correlation of knapsack "items"

$$\sum_{[l,r]:i\in[l,r]} x_{[l,r]} \leq 1 \quad \forall i \in \mathcal{BL}$$

$$x_{[l,r]} \in \{0,1\} \quad \forall [l,r] \in \mathcal{D}$$

Filtering Problems
Overview

		Input blacklist	
		A static blacklist	A time varying blacklist
filter all bad IPs?	yes	**FILTER-ALL**	**FILTER-ALL-DYNAMIC**
	no	**FILTER-SOME**	**FILTER-SOME-DYNAMIC**

- Aggregation of source addresses using: Ranges or **Prefixes**
- Constraint on the (single) link capacity: **FLOODING**
- Filter at multiple routers: **DISTRIBUTED FILTERING**

Filter Selection Problem
Scope and Assumptions

- Focus on resource allocation/optimization
  - not on architecture/protocols – complementary
- Blacklist and Whitelist are given as input
  - e.g., from an IDS/blacklists and NetFlow modules respectively
- Source IPs are accurate (not spoofed)
  - Small % spoofable or spoofed [Spoofer05, Park01, …]
  - Today's botnets no longer need/use spoofing
  - Proposals to increase accountability: ingress filtering, packet passports [Yang07], self-certifying addresses [Andersen08]
  - Can address some "error" within our framework
- Who needs to participate
  - Single router, or routers of same ISP

Overview

- Problem Overview and Motivation
- Filtering Algorithms
  - RANGE-based (filter IP or range [l,r])
    - FILTER-ALL-RANGE
    - FILTER-SOME-RANGE
    - FILTER-ALL-DYNAMIC-RANGE
  - PREFIX-based (filter IP source or prefix)
    - FILTER-ALL-PREFIX
    - FILTER-SOME-PREFIX
    - FILTER-ALL-DYNAMIC-PREFIX
    - **FILTER-FLOODING**
    - **DISTRIBUTED-FILTERING**
- Conclusion

FILTER-ALL-RANGE
Problem Statement

- Given: a blacklist BL, weight w_i (associated with each good IP) and F_{max} filters
- choose: filters $X_{[l,r]}$
- so as to: filter *all* bad addresses

  and minimize collateral damage

$$\min \sum_{[l,r]} g_{[l,r]} x_{[l,r]}$$

$$\text{s.t.} \quad \sum_{[l,r]} x_{[l,r]} \leq F_{max}$$

$$\sum_{[l,r]:i \in [l,r]} x_{[l,r]} = 1 \quad \forall i \in \mathcal{BL}$$

$$x_{[l,r]} \in \{0,1\} \quad \forall l < r \in [0, ..., 2^{32}]$$

$$g_{[l,r]} = \sum_{i \in [l,r] \cap \mathcal{G}} w_i$$

is the weighted sum of good addresses in range [l,r]

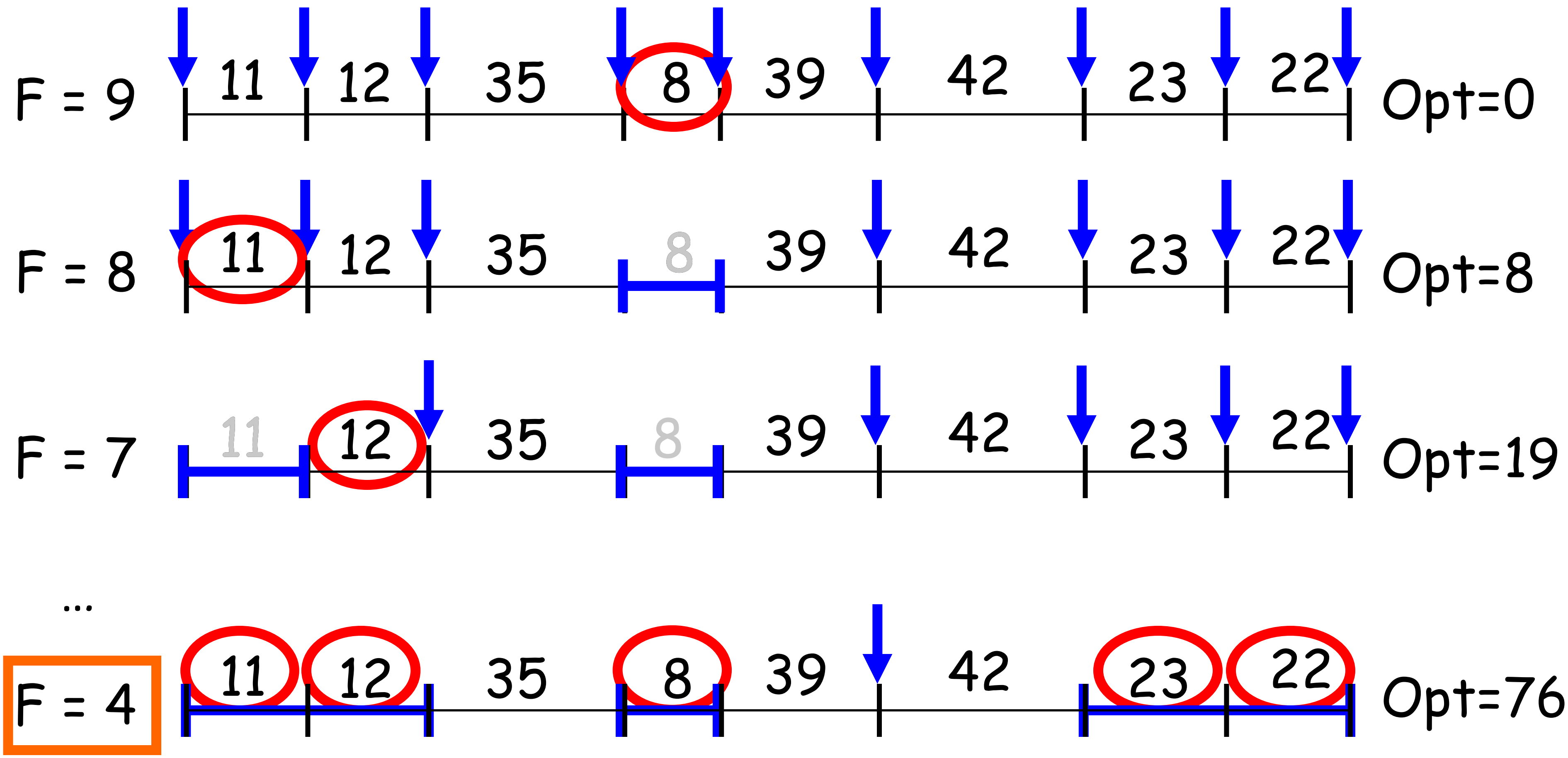
FILTER-ALL-RANGE
Greedy Algorithm

- Let F=N
  - assign one filter to each bad address
- While $F>F_{max}$
  - make the following <u>greedy decision:</u>
    - pick the two "closest" bad IPs/intervals
    - Merge them in a single interval
  - decrease F=F-1

FILTER-ALL-RANGE

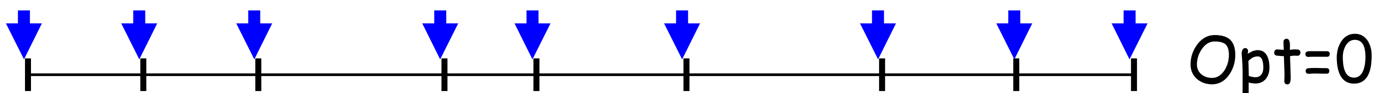
Example of running Greedy

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

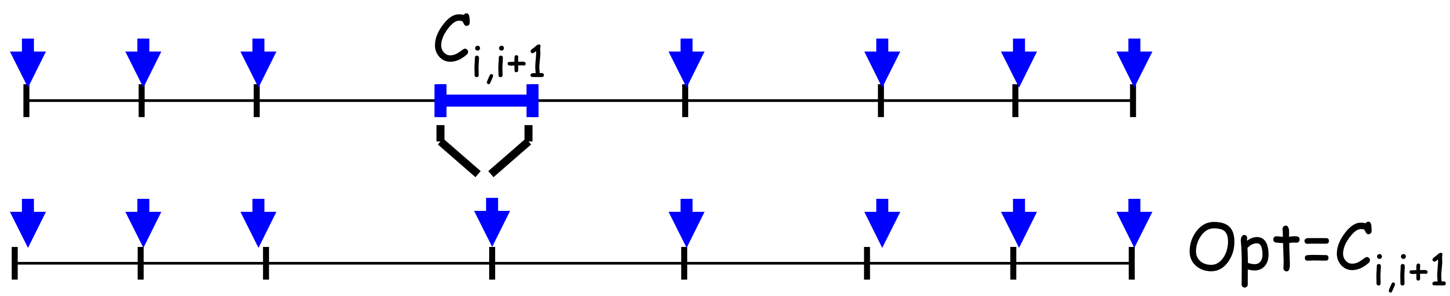


FILTER-ALL-RANGE
Greedy Algorithm: Properties

- Optimality, outline of proof:
  - If F=N



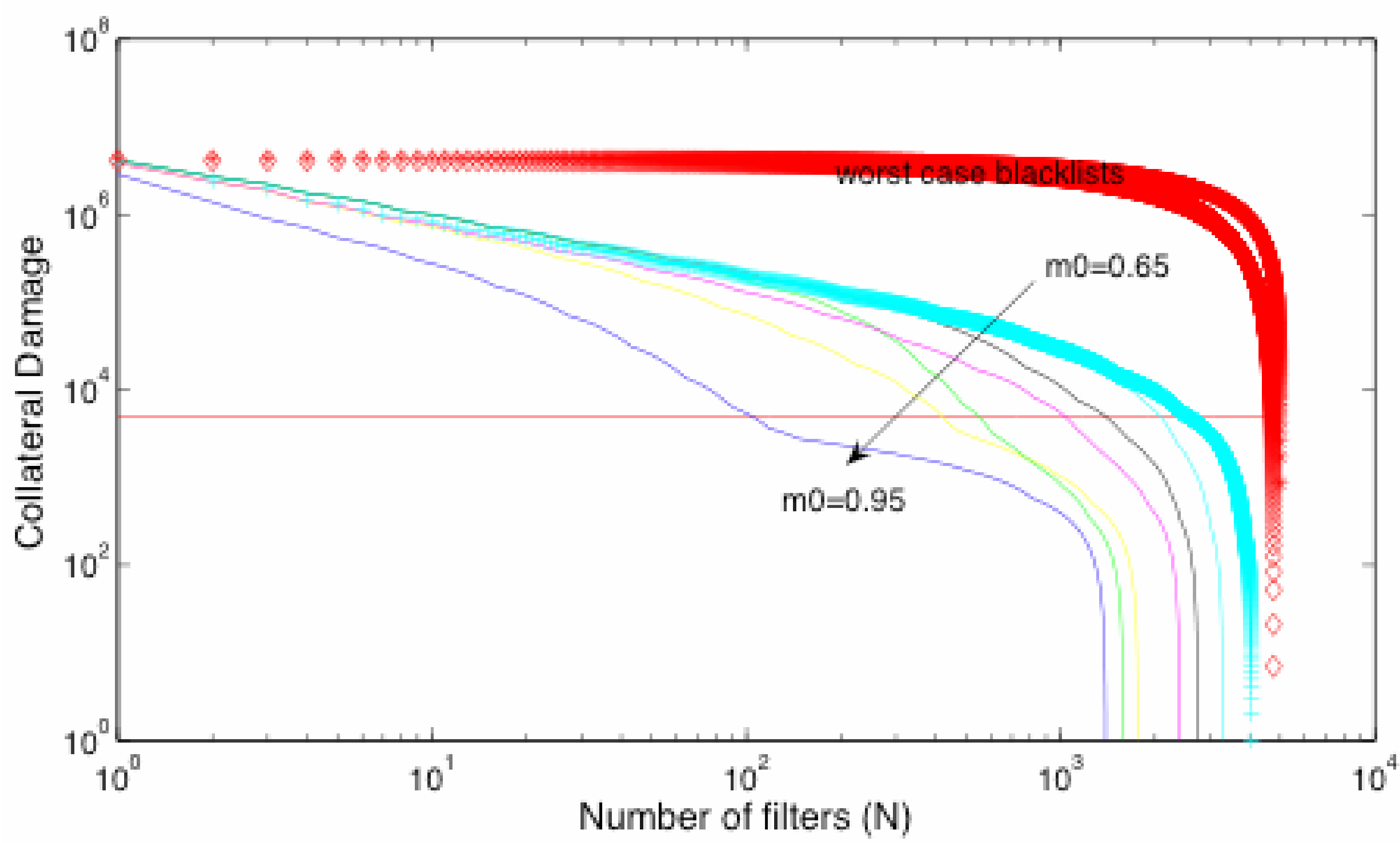
  - If F=N-1



- Complexity:
  - We need to find the (N-F_{max}) smallest components in a vector of size N. This can be done in O(N)
  - This is smallest achievable complexity for this problem.

FILTER-ALL-RANGE
Simulations

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



FILTER-SOME-RANGE
Problem Statement

- Given: a blacklist BL
  weight w_i for every address i (>0 for good, <0 for bad)
  and F_{max} filters
- choose: filters $X_{[l,r]}$
- so as to: (filter ***some*** bad addresses) & minimize the total weight

$$\min \sum_{[l,r]} \left(g_{[l,r]} - b_{[l,r]}\right) x_{[l,r]}$$

$$\text{s.t.} \sum_{[l,r]} x_{[l,r]} \leq F_{max}$$

$$\sum_{[l,r]:i\in[l,r]} x_{[l,r]} \leq 1 \quad \forall i \in \mathcal{BL}$$

$$x_{[l,r]} \in \{0,1\} \quad \forall l < r \in [0, ..., 2^{32}]$$

$$g_{[l,r]} = \sum_{i\in[l,r]\cap\mathcal{G}} w_i$$

is the weighted sum of good addresses in range [l,r]

$$b_{[l,r]} = \sum_{i\in[l,r]\cap\mathcal{BL}} |w_i|$$

is the weighted sum of bad addresses in range [l,r]

FILTER-SOME-RANGE

Address Weights

- Objective Function:

$$\min \sum_{[l,r]} \sum_{i \in [l,r]} w_i x_{[l,r]} =$$

$$\min \sum_{[l,r]} \Big(\sum_{i \in [l,r] \cap \mathcal{G}} w_i + \sum_{i \in [l,r] \cap \mathcal{BL}} w_i \Big) x_{[l,r]} =$$

$$\min \sum_{[l,r]} \left(g_{[l,r]} - b_{[l,r]} \right) x_{[l,r]}$$

- 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

P2: FILTER-SOME-RANGE
Greedy Algorithm

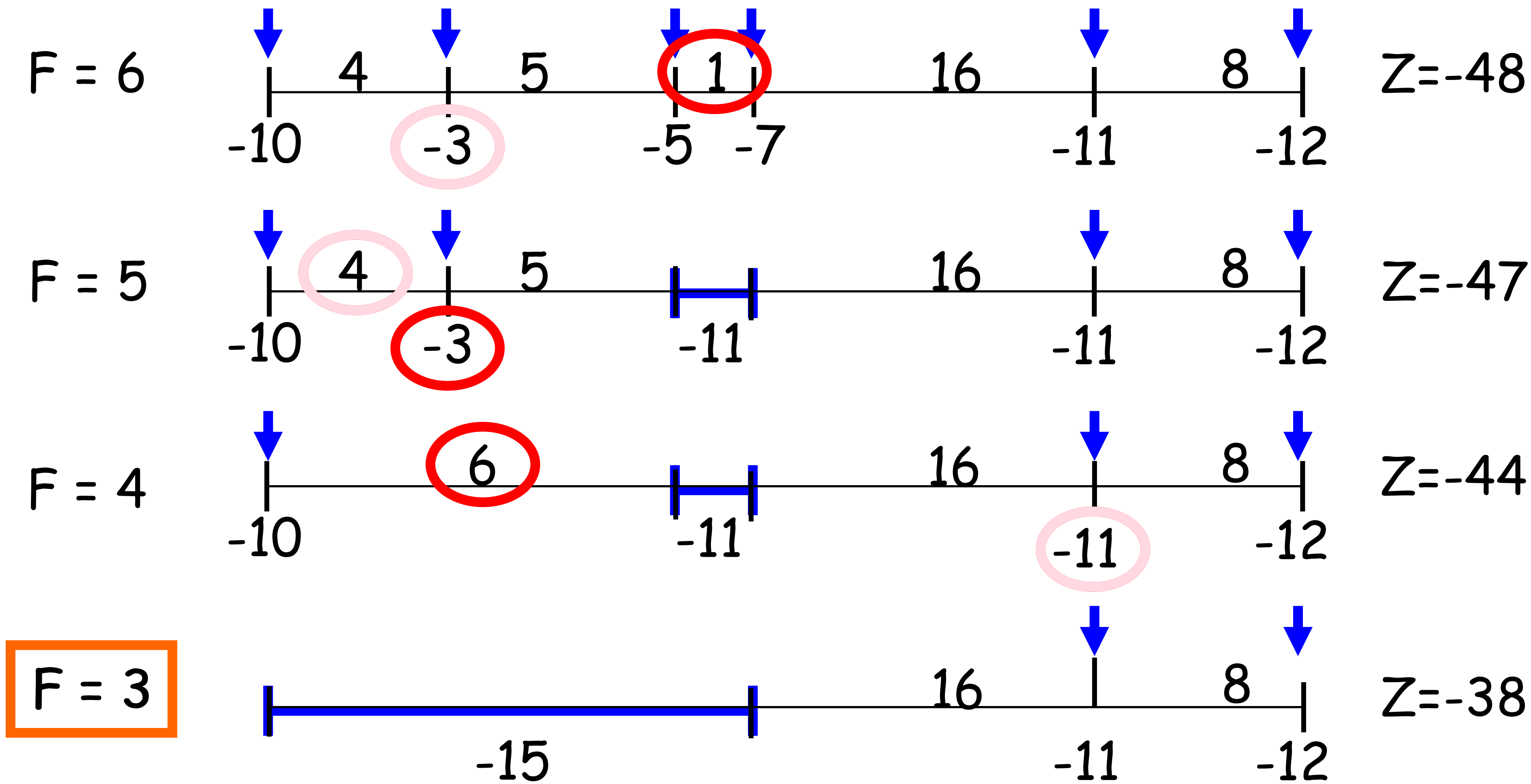
- Let F=N
  - assign one filter to each bad address
- While $F>F_{max}$
  - make the following greedy decision:
    - merge the two "closest" filters, or release a filter,

    whichever causes the smallest increase in objective function
  - decrease F=F-1

FILTER-SOME-RANGE

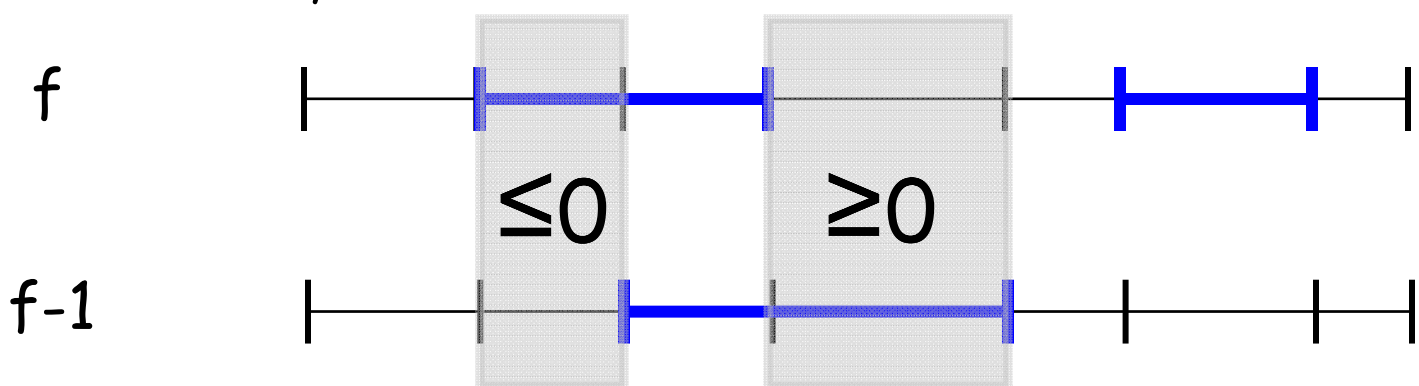
Example of running Greedy

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



FILTER-SOME-RANGE
Greedy Algorithm: Properties

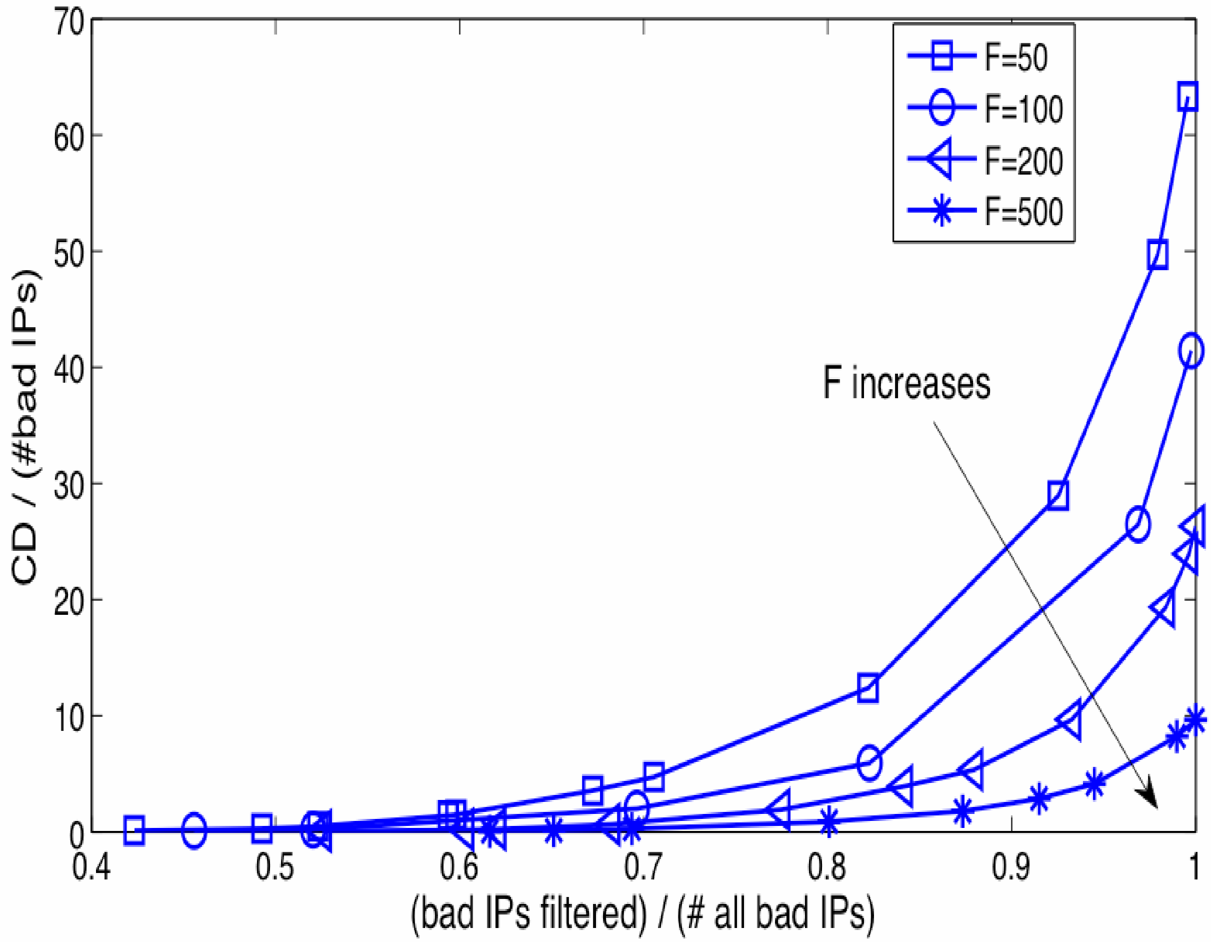
- Optimality, proof outline:
  - Construct an initial optimal (but infeasible) solution.
  - At every step:
    - *f* is reduced and optimality of the solution is preserved.
    - Merge 2 filters or release 1: the only possibilities
      - by contradiction:



- Complexity
  - partial sorting O(N), as in FILTER-ALL

FILTER-SOME-RANGE

Simulations



FILTER-ALL vs. FILTER-SOME

Comparison

- FILTER-ALL
  - explicitly controls the addresses to block
  - easier to use by operators
  - too strict, may cause large collateral damage
- FILTER-SOME
  - explicitly controls the weights of the addresses – may be tricky
  - always blocks less ($N1 \le N$) addresses and achieves less collateral damage than FILTER-ALL
- As $|W_b| \gg |W_g|$, FILTER-SOME $\rightarrow$ FILTER-ALL
- FILTER-SOME finds the best set of N1 addresses to use as input to FILTER-ALL

The Time-Varying Case

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

Time-Varying Blacklists
Problem Statement

- FILTER-ALL(SOME)-DYNAMIC
  - 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 FILTER-ALL(SOME) for blacklist BL_{Ti} at all times

- Solution
  - run the algorithm for the static problem from scratch at every time T_i
  - …or exploit temporal correlation and just update filtering as needed

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 interval added, 2 intervals deleted]
  - 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]

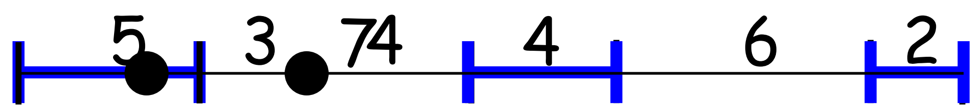
FILTER-ALL-DYNAMIC

Example of a new address appearing

$F_{max} = 3$

$N = 6$

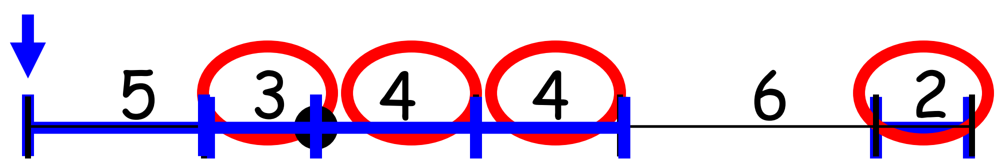
$N - F_{max} = 3$



$F_{max} = 3$

$N = 7$

$N - F_{max} = 4$



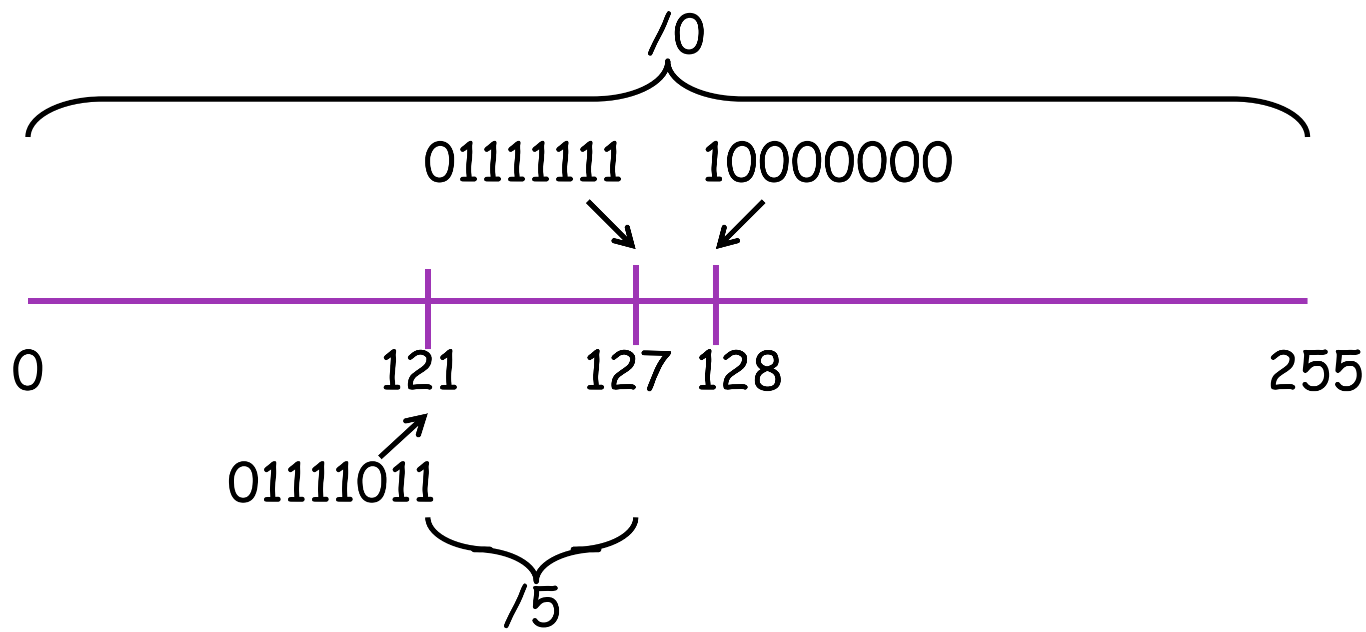
Overview

- Problem Overview and Motivation
- **Filtering Algorithms**
  - RANGE-based (filter IP or range [l,r])
    - FILTER-ALL-RANGE
    - FILTER-SOME-RANGE
    - FILTER-ALL-DYNAMIC-RANGE
  - PREFIX-based (filter IP source or prefix)
    - FILTER-ALL-PREFIX
    - FILTER-SOME-PREFIX
    - FILTER-ALL-DYNAMIC-PREFIX
    - **FILTER-FLOODING**
    - **DISTRIBUTED-FILTERING**
- Conclusion

Source Prefix-based Filtering

- In router ACLs, we cannot use arbitrary ranges to aggregate source addresses
- IP prefix = range, [l,r], such that:
  - $(r-l) = 2^L-1$, for some $L=0,1,\ldots,32$
  - $l=0 \bmod 2^L$
- We use the traditional notation:
  - Address/mask: p/l

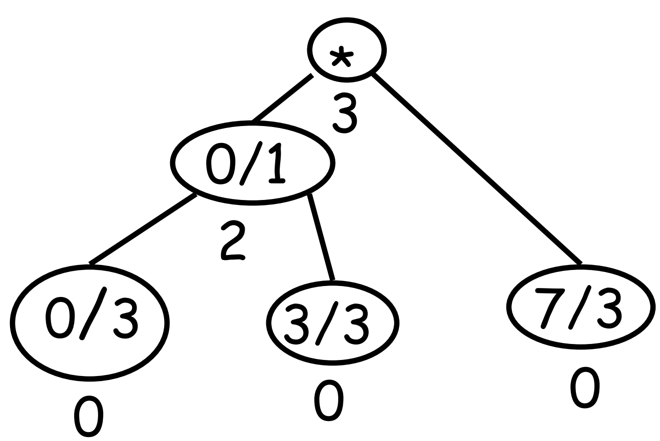
IP prefixes harden the problem



- Merging the “closest” IPs may cause arbitrarily high collateral damage
  - We cannot apply the same greedy approach as in ranges
  - We cannot use the range-based optimal solution to approximate the prefix-based optimal solution (no interesting bound)

Longest Common Prefix (LCP) Tree

- Definition
  - given a set of addresses, S, we define LCP-Tree(S):
  - the binary tree whose leaves are addresses in S, and intermediate nodes represent all and only the longest common prefixes between addresses in S
  - cost associated with each node - problem specific
- Example
  - For 3bit addresses, S={0,3,7}, the LCP-Tree(S) is:



LCP-Tree

- Complexity:
  - Given a blacklist of N addresses, building the LCP-tree requires N insertions in a Patricia tree: $O(mN)$, where m is the bit-length (32 if IPv4)
- Property:
  - Provides a concise representation of the problem: It encodes all and only the prefixes necessary to compute an optimal solution
    - There exists an optimal solution which can be represented as a subtree of the LCP-tree
    - If we remove any of those prefixes, we can construct an instance of the filtering problem s.t. the OPT requires exactly that prefix.

FILTER-ALL-PREFIX
Problem Statement

- Given: a blacklist, weight w_i (associated with each good IP) and F_{max} filters
- choose: source IP prefixes, $X_{p/l}$
- so as to: filter *all* bad addresses
  and minimize collateral damage

$$\min \sum_{p/l} g_{p/l} x_{p/l}$$

$$\text{s.t.} \quad \sum_{p/l} x_{p/l} \leq F_{max}$$

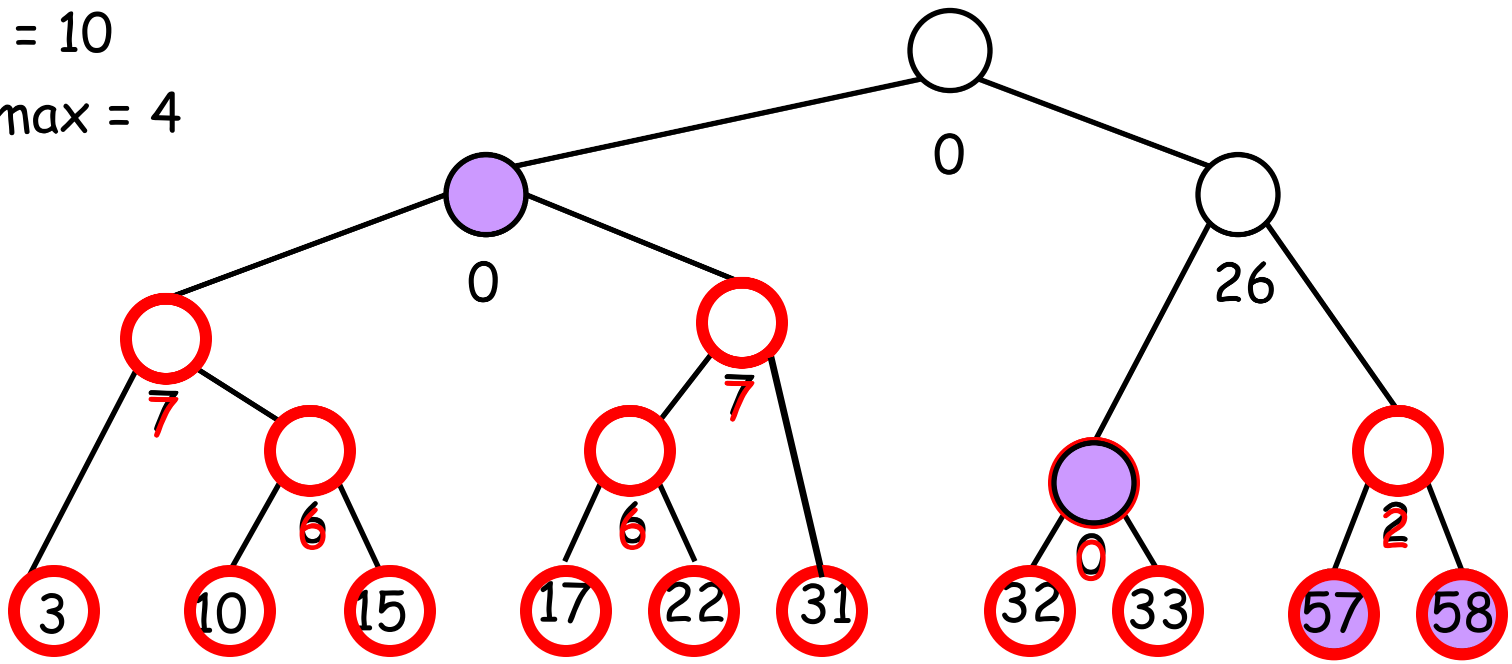
$$\sum_{p/l: i \in p/l} x_{p/l} = 1 \quad \forall i \in \mathcal{BL}$$

$$x_{p/l} \in \{0,1\} \quad \forall l = 0, ..., 32, p = 0, ..., 2^l$$

Simple greedy strategies do not work

N = 10

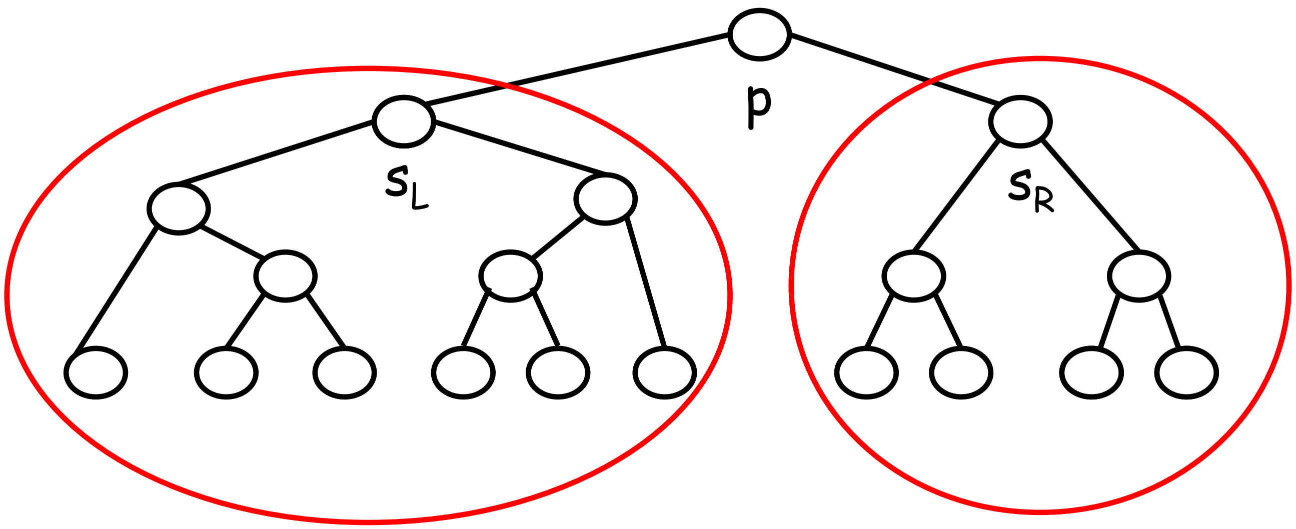
Fmax = 4



- Merging (N-Fmax) closest leaves: 28
- Optimal solution: 26

FILTER-ALL-PREFIX
DP Algorithm (1)

- F: filters available at node (prefix) *p*



$F-n\geq 1$, filters within left subtree

$n\geq 1$, filters within right subtree

$$z_p(1) = g_p \ \forall p$$

$$z_p(F) = \min_{n=1,\ldots,F-1} \left\{ z_{s_l}(F-n) + z_{s_r}(n) \right\}, \ F > 1$$

FILTER-ALL-PREFIX
DP Algorithm (2)

- Build LCP-Tree(BL)
- For all leaves: $z_{leaf}(F)=0,\ \ F=1,\ldots,F_{max}$
- level=level(leaf)-1
- While: level≥level(root)
  - For all node, *p*, s.t. *level(p)=level*

$$z_p(1) = g_p \ \forall p$$
$$z_p(F) = \min_{n=1,\ldots,F-1} \left\{ z_{s_l}(F-n) + z_{s_r}(n) \right\},\ \ F > 1$$

  - level=level-1
- Return: $z_{root}(F_{max})$

FILTER-ALL-PREFIX
DP Algorithm: Properties

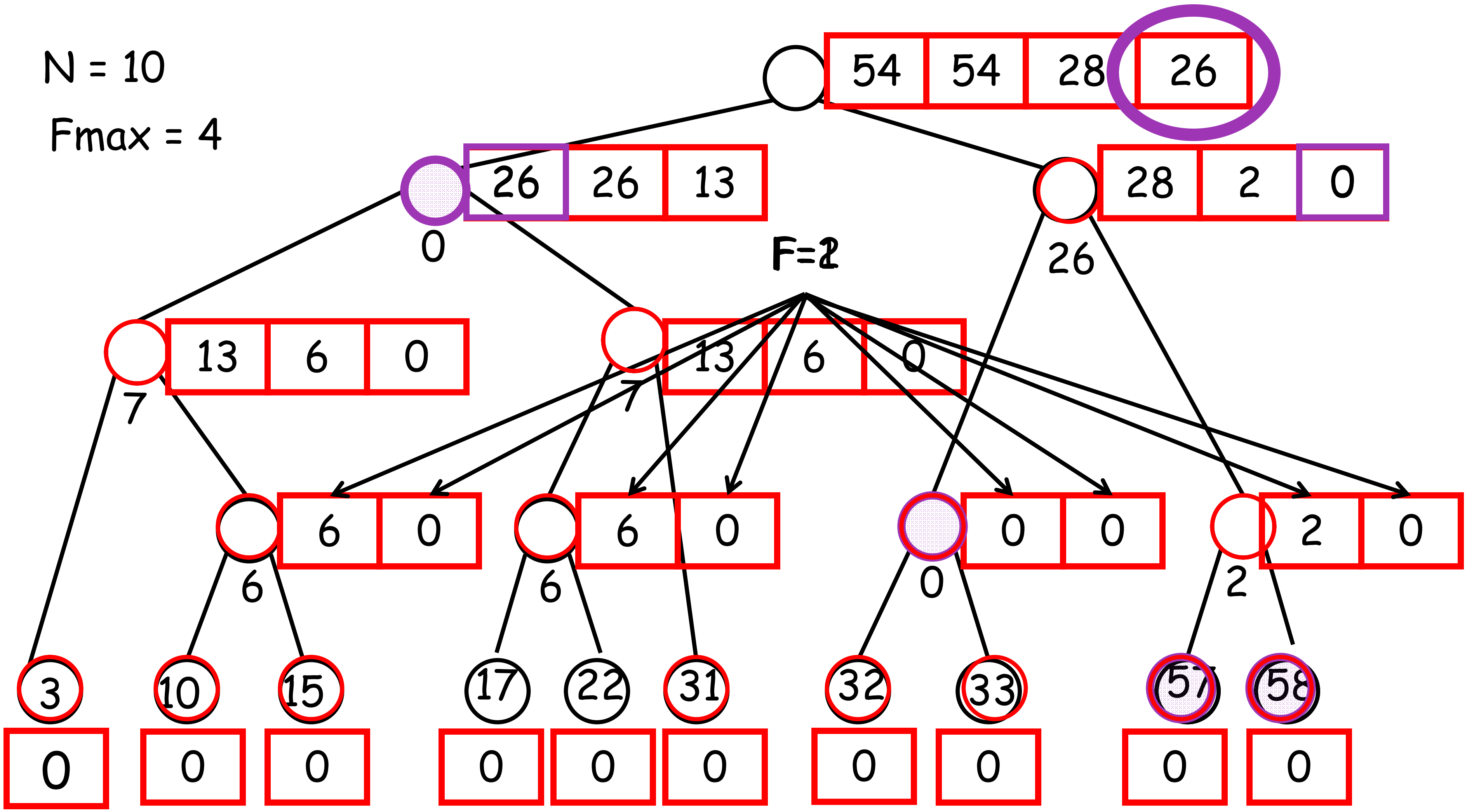
- Optimality
- Complexity
  - Linearly increasing with N:
    - O(mN) + O(FmaxN), where m=32 and Fmax<<N

FILTER-ALL-PREFIX

DP Algorithm: Example

N = 10

Fmax = 4

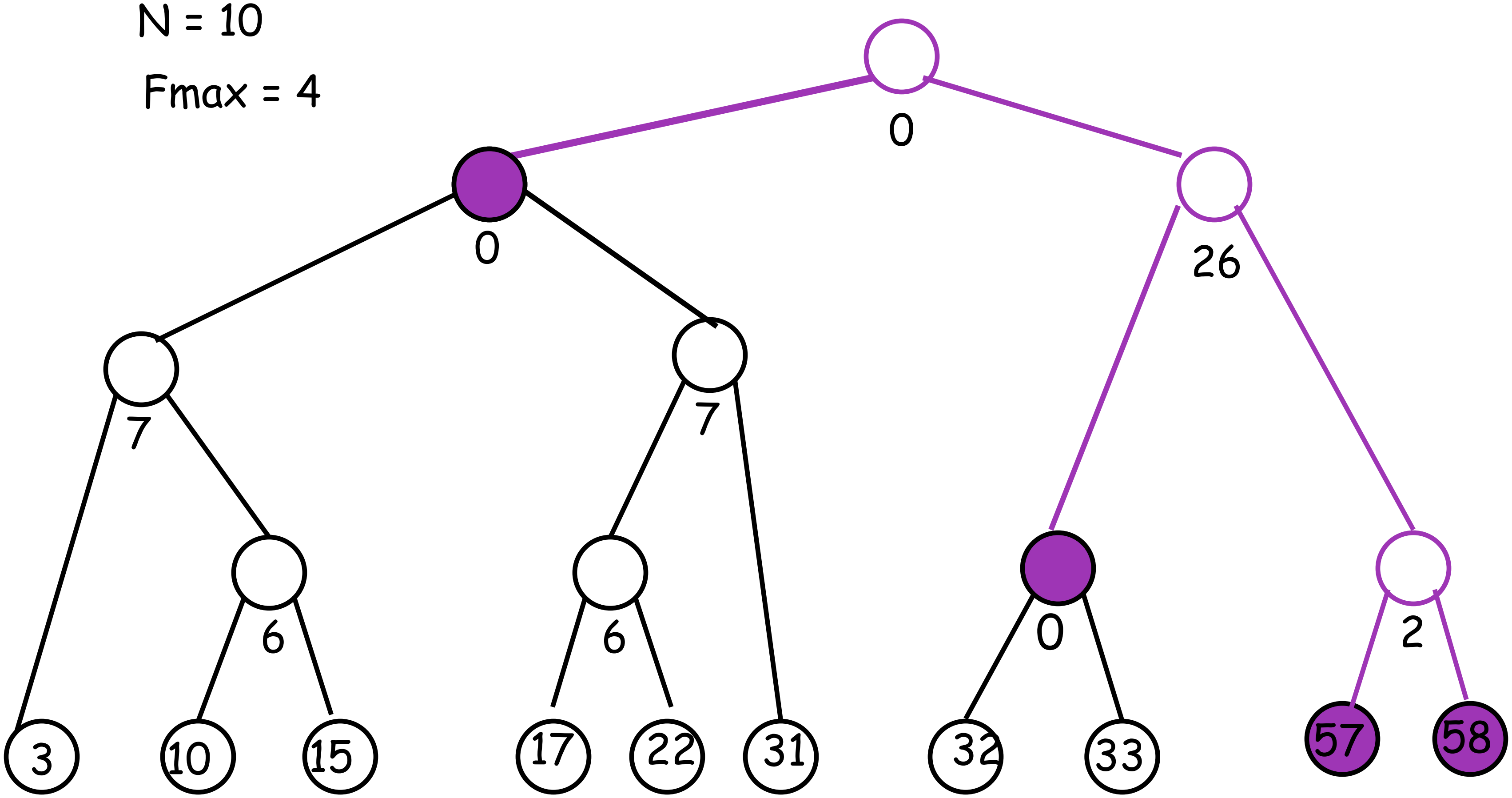


FILTER-ALL-PREFIX

Example - Solution

N = 10

Fmax = 4



FILTER-SOME-PREFIX
Problem Statement

- Given: a blacklist BL
  weight w_i of every address i (>0 for good and <0 for bad)
  and F_{max} filters
- choose: source IP prefixes, $X_{p/l}$
- so as to: filter *some* bad addresses
  minimize total weight

$$\min \sum_{p/l} \left(g_{p/l} - b_{p/l}\right) x_{p/l}$$

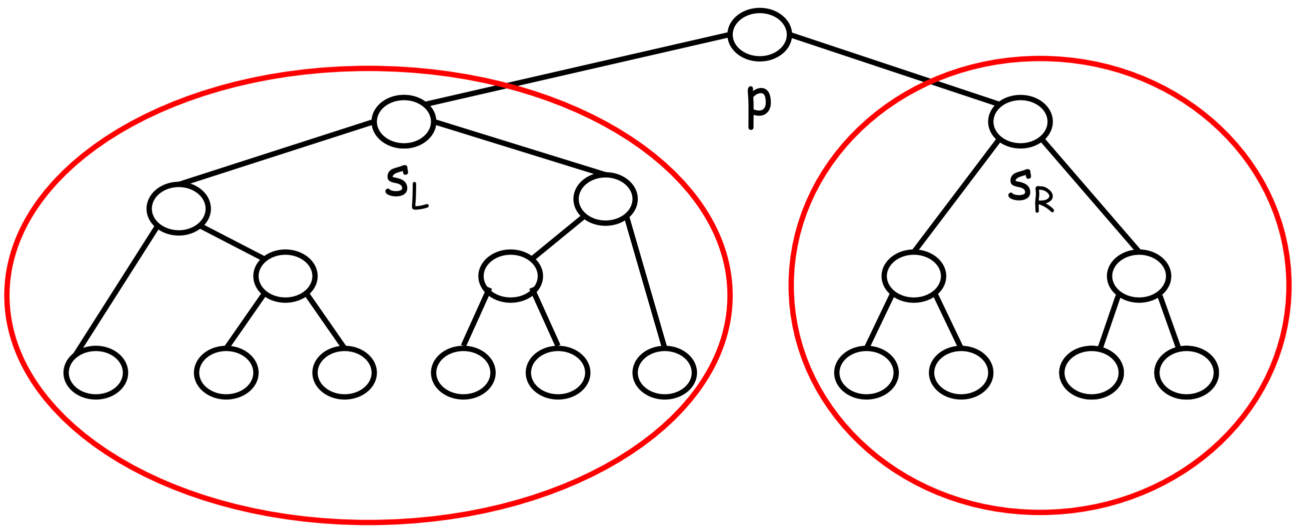
$$\text{s.t.} \sum_{p/l} x_{p/l} \leq F_{max}$$

$$\sum_{p/l : i \in p/l} x_{p/l} \leq 1 \quad \forall i \in \mathcal{B}L$$

$$x_{p/l} \in \{0, 1\} \quad \forall l = 0, ..., 32, p = 0, ..., 2^l$$

FILTER-SOME-PREFIX
DP Algorithm

- F_{max}: filters available at node (prefix) *p*



$F_{max}-n\geq 0$, filters within left subtree

$n\geq 0$, filters within right subtree

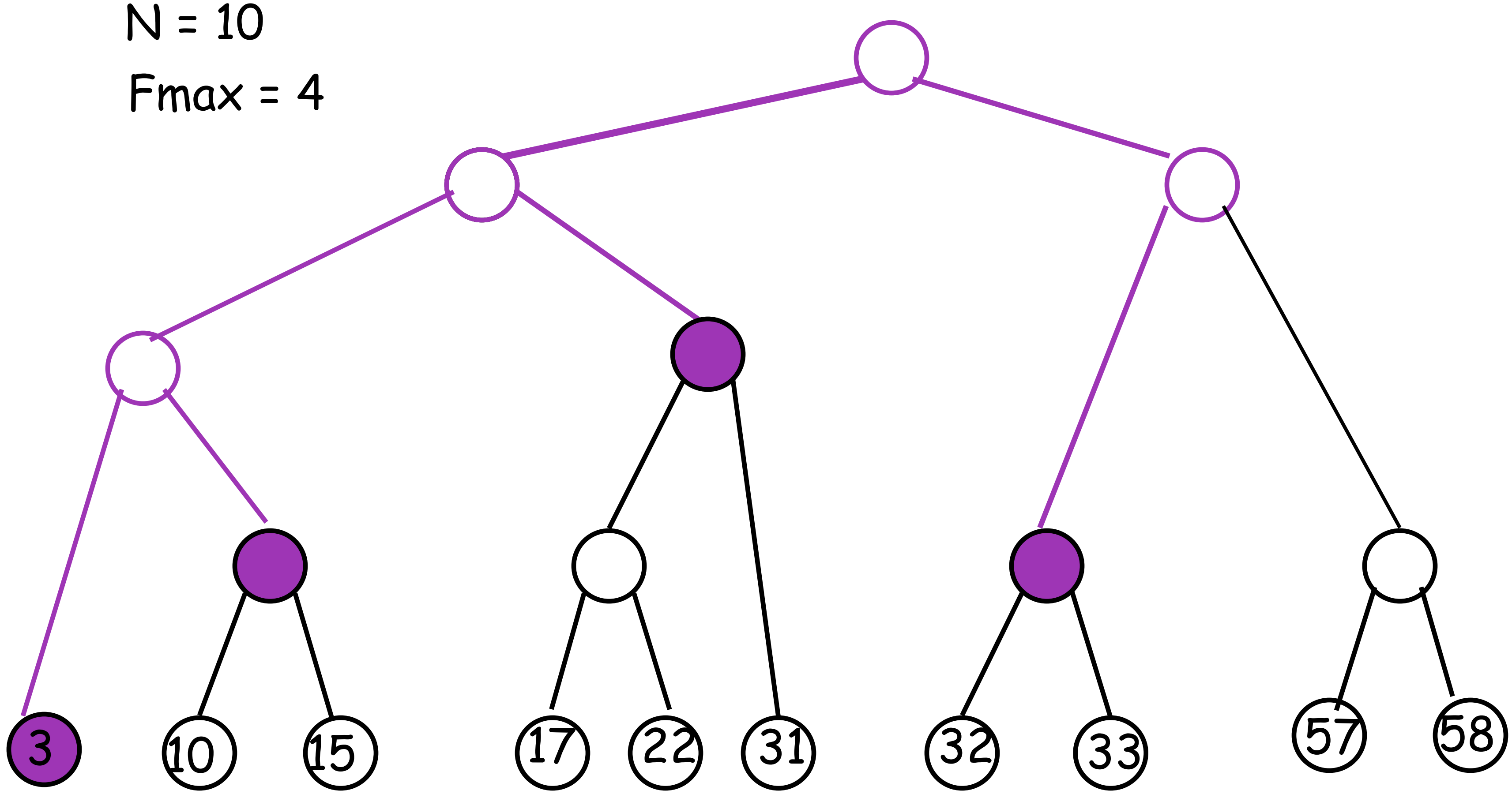
- $n=0,2\ldots F_{max}$: means we may not block all malicious IPs
- choose n to minimize collateral damage

FILTER-SOME-PREFIX

Example – Feasible Solution

N = 10

Fmax = 4



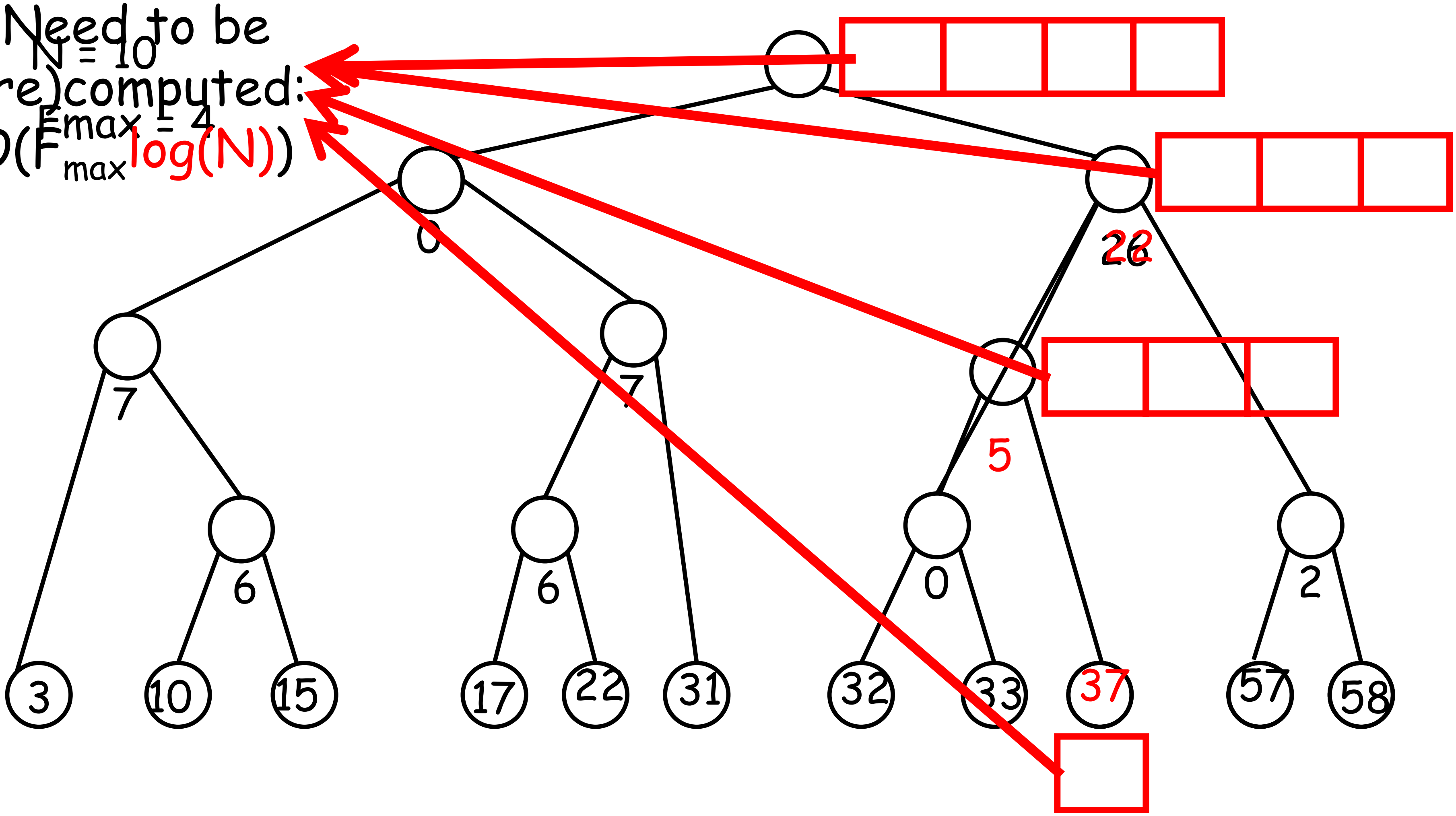
The Time-Varying case

FILTER-ALL-PREFIX-DYNAMIC

N = 10

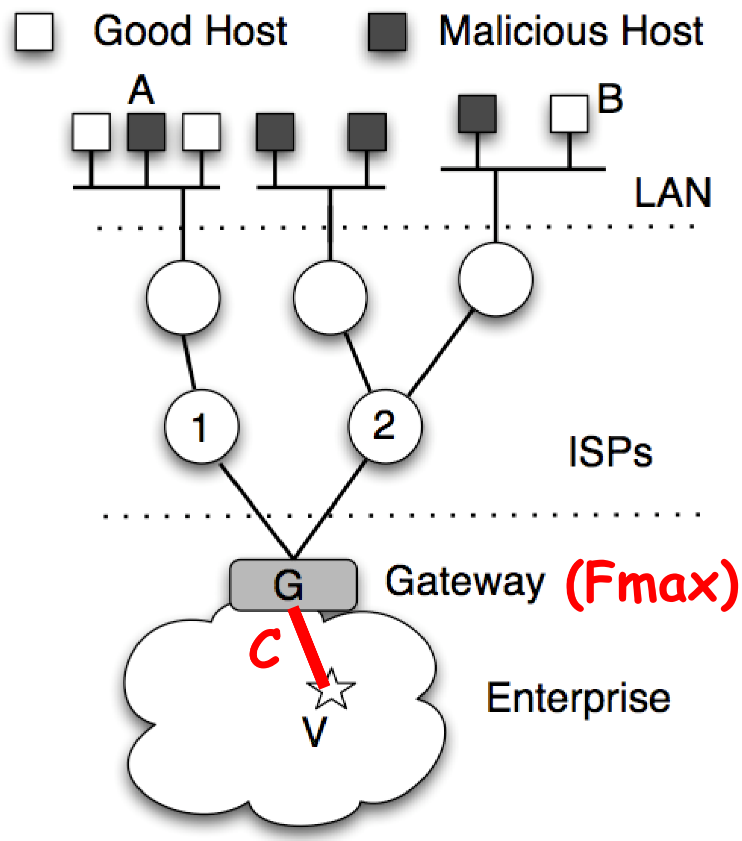
Fmax = 4

Need to be (re)computed: $O(F_{max} \log(N))$



FLOODING
Motivation

- DDoS: Malicious hosts coordinate to flood the access link to a victim
- Weights of every address represent the traffic volume



FLOODING
Problem Statement

- Given: a blacklist BL, a set of legitimate sources G, weight of address = traffic volume generated, a constraint on the link capacity C, and F_{max} filters
- choose: source IP prefixes, $X_{p/l}$
- so as to: minimize the collateral damage and the total traffic fits within the link capacity

$$\min \sum_{p/l} g_{p/l} x_{p/l}$$

$$\text{s.t.} \sum_{p/l} x_{p/l} \leq F_{max}$$

$$\sum_{p/l} \left(g_{p/l} + b_{p/l}\right)(1 - x_{p/l}) \leq C$$

$$\sum_{p/l:i \in p/l} x_{p/l} \leq 1 \quad \forall i \in \mathcal{BL}$$

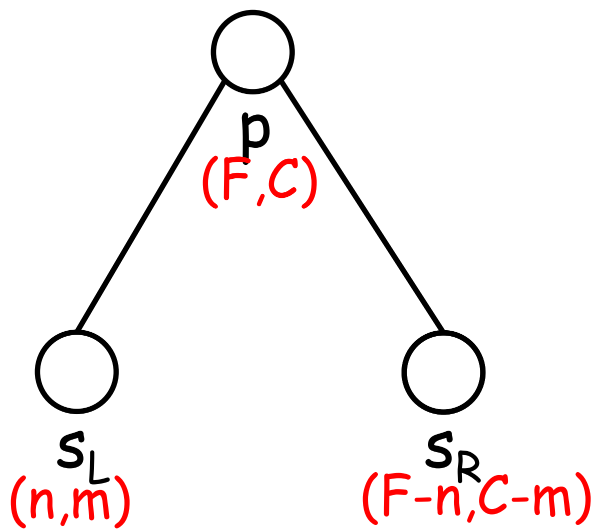
"Weights" change per every item

FLOODING
Solution

- FLOODING is NP-hard
  - reduces to multidimensional knapsack
- A pseudo-polynomial algorithm, solves the problem in $O(CF_{max}N)$
  - similar to the DP for FILTER-All/SOME-PREFIX
  - extended to take into account the capacity constraint
  - the LCP-Tree includes both good and bad addresses

FLOODING
DP Algorithm (1)

- At node (prefix) p, we have available:
  - F filters and capacity C
- Allocate
  - to the left subtree: n filters and m capacity
  - To the right subtree: F-f filters and C-m capacity
- Minimize collateral damage
  - By choosing appropriate n=0…F, m=0,…C to

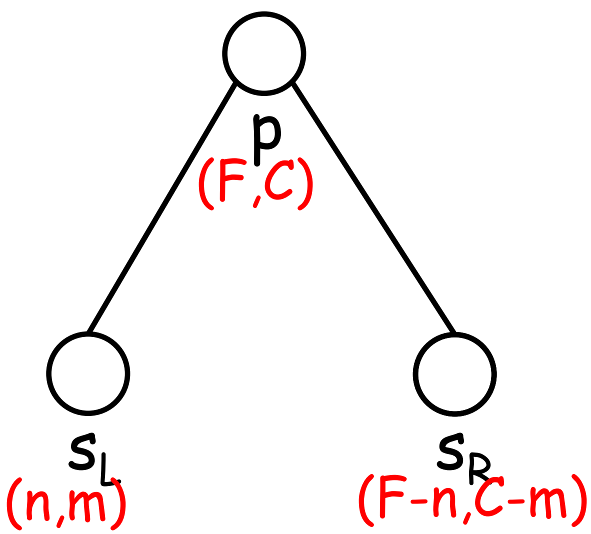


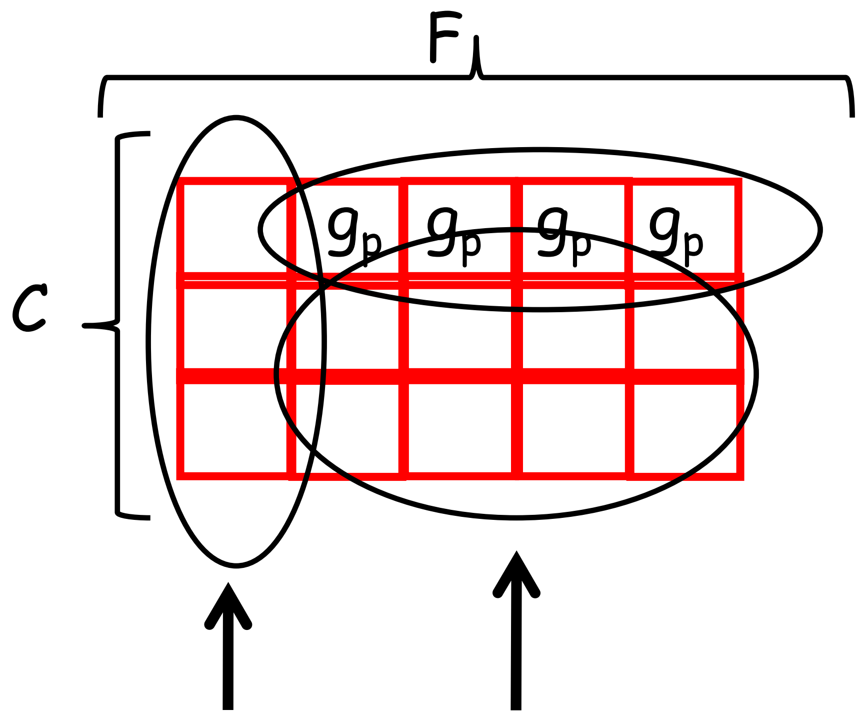
FLOODING

DP Algorithm (2)

Recursive equation:

Boundary conditions:





=0 if C > g +b

$$z_p(F,c) = \min_{\substack{n=0,\ldots,F \\ m=0,\ldots,c}} \{z_{s_l}(F-n, c-m) + z_{s_r}(n,m)\}$$

FLOODING vs. FILTER-SOME
Relation

- Consider the dual of FLOODING:

$$\max_{\lambda \geq 0} \Big\{ \min \sum_{p/l} \Big[(1-\lambda) g_{p/l} - \lambda b_{p/l} \Big] x_{p/l} + \\ + \sum_{p/l} \lambda (g_{p/l} + \lambda b_{p/l}) - \lambda C \Big\}$$

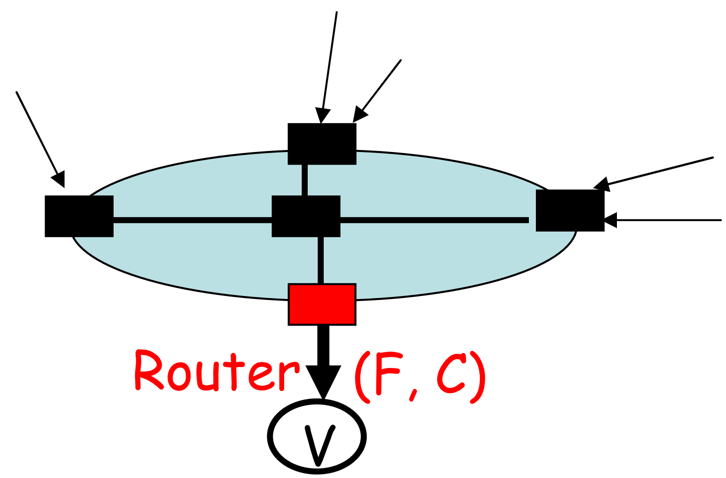
$$\text{s.t.} \sum_{p/l} x_{p/l} \leq F_{max}$$

$$\sum_{p/l: i \in p/l} x_{p/l} \leq 1 \quad \forall i \in \mathcal{BL}$$

- Per every fixed λ, we have a different instance of FILTER-SOME, for specific assignment of weights

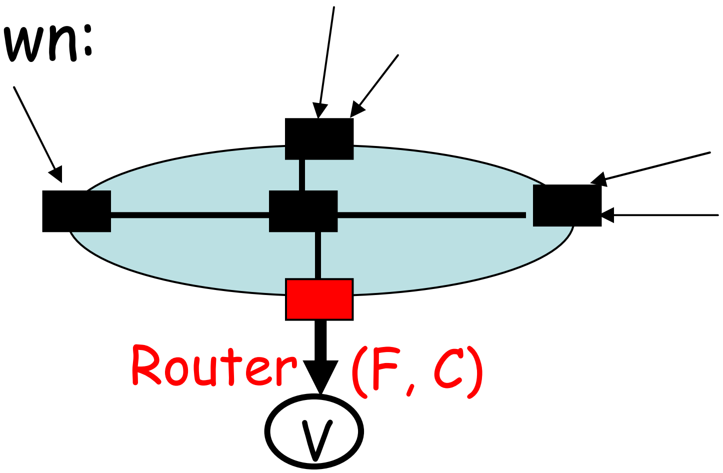
DISTRIBUTED FILTERING against FLOODING
Motivation

- A single network (ISP or enterprise) may collaboratively deploy filters on several routers
  - increase filter budget
- Also applicable to across ISPs, if they collaborate



DISTRIBUTED FILTERING
Problem Statement

- Similar problem as FLOODING
- But there are several routers
- And each router (u) has its own:
  - view of good/bad traffic
  - capacity in downstream link
  - filter budget
- The question is to choose
  - not only which prefix
  - but also on which router



DISTRIBUTED FILTERING

Problem Formulation

FLOODING, single router	DIST. FLOODING several routers (u)
$\min \sum_{p/l} g_{p/l} x_{p/l}$	$\min \sum_{u \in \mathcal{R}} \sum_{p/l} g^{(u)}_{p/l} x^{(u)}_{p/l}$
s.t. $\sum_{p/l} x_{p/l} \leq F_{max}$	s.t. $\sum_{p/l} x^{(u)}_{p/l} \leq F^{(u)}_{max} \quad \forall u \in \mathcal{R}$
$\sum_{p/l} \left(g_{p/l} + b_{p/l}\right)(1 - x_{p/l}) \leq C$	$\sum_{p/l} \left(g^{(u)}_{p/l} + b^{(u)}_{p/l}\right)(1 - x^{(u)}_{p/l}) \leq C^{(u)} \quad \forall u \in \mathcal{R}$
$\sum_{p/l: i \in p/l} x_{p/l} \leq 1 \quad \forall i \in \mathcal{BL}$	$\sum_{u \in \mathcal{R}} \sum_{p/l \ni i} x^{(u)}_{p/l} \leq 1 \quad \forall i \in \mathcal{BL}$
	This constraint couples the routers, preventing a direct decomposition

DISTRIBUTED FILTERING
Solution

- Consider the partial lagrangian:

$$L(x,\lambda) = \sum_{u\in\mathcal{R}} \sum_{p/l} g_{p/l}^{(u)} x_{p/l}^{(u)} + \sum_{A\in\mathcal{BL}} \lambda_i \Big(\sum_{u\in\mathcal{R}} \sum_{p/l\ni i} x_{p/l}^{(u)} - 1 \Big)$$

$$= \sum_{u\in\mathcal{R}} \Big(\sum_{p/l} \Big(g_{p/l}^{(u)} + \lambda_{p/l} \Big) x_{p/l}^{(u)} \Big) - \sum_{A\in\mathcal{BL}} \lambda_i$$

- Each Subproblem
  - Is an instance of FLOODING, can be solved independently at each router

$$\min \sum_{p/l} \Big(g_{p/l}^{(u)} + \lambda_{p/l} \Big) x_{p/l}^{(u)}$$

$$\text{s.t.} \sum_{p/l} x_{p/l}^{(u)} \le F_{max}^{(u)}$$

$$\sum_{p/l} \Big(g_{p/l}^{(u)} + b_{p/l}^{(u)} \Big) (1 - x_{p/l}^{(u)}) \le C^{(u)}$$

- Master Problem
  - Can be solved using a subgradient method

$$\max_{\lambda_i \ge 0} \sum_{u\in\mathcal{R}} h_u(\lambda) - \sum_{i\in\mathcal{BL}} \lambda_i$$

Overview

- Problem Overview and Motivation

- Filtering Algorithms
  - RANGE-based (filter IP or range [l,r])
    - FILTER-ALL-RANGE
    - FILTER-SOME-RANGE
    - FILTER-ALL-DYNAMIC-RANGE
  - PREFIX-based (filter IP source or prefix)
    - FILTER-ALL-PREFIX
    - FILTER-SOME-PREFIX
    - FILTER-FLOODING
    - **DISTRIBUTED-FILTERING**
    - **FILTER-ALL-DYNAMIC-PREFIX**

- Conclusion

Conclusion

- A framework for filter selection
- Part of a larger system for collecting and data from multiple sensors and taking appropriate action
  - Look at the interaction of detection and filtering

Thank you!

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