Filtering Malicious IP Sources

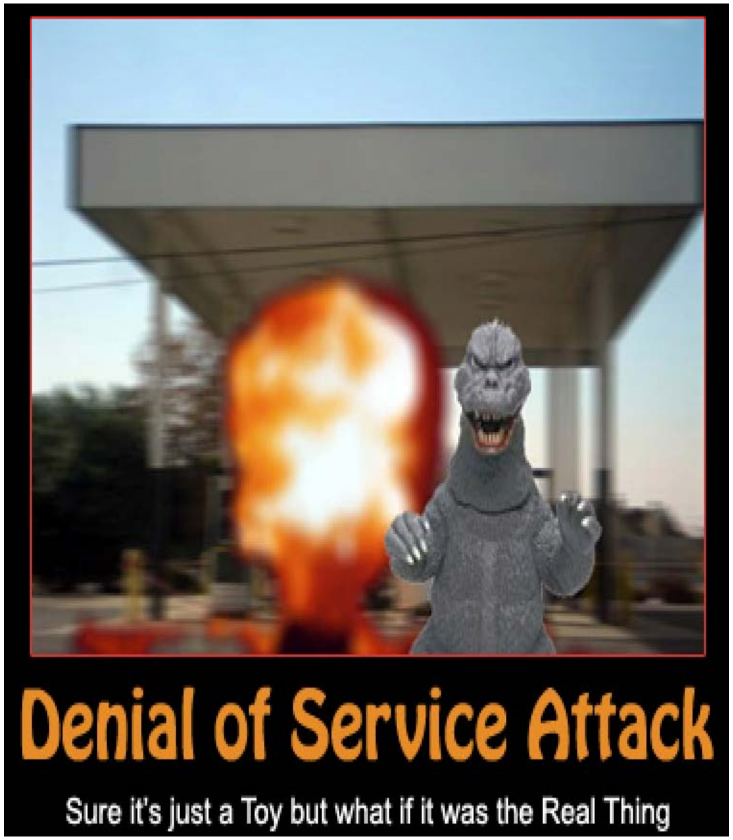
Models and Algorithms

Fabio Soldo, Athina Markopoulou,
Katerina Argyraki

UC Irvine, EPFL

Context

- Problem: **Malicious IP Traffic**
  - Denial-of-service attacks
  - Spam
  - …
- Solution requires many components
  - Detection of malicious traffic
  - Action: filtering
  - Anti-spoofing, accountability
  - …



http://www.networkliquidators.com/gallery/tech_notions/a1-godzilla-denial-of-service-attack.jpg

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

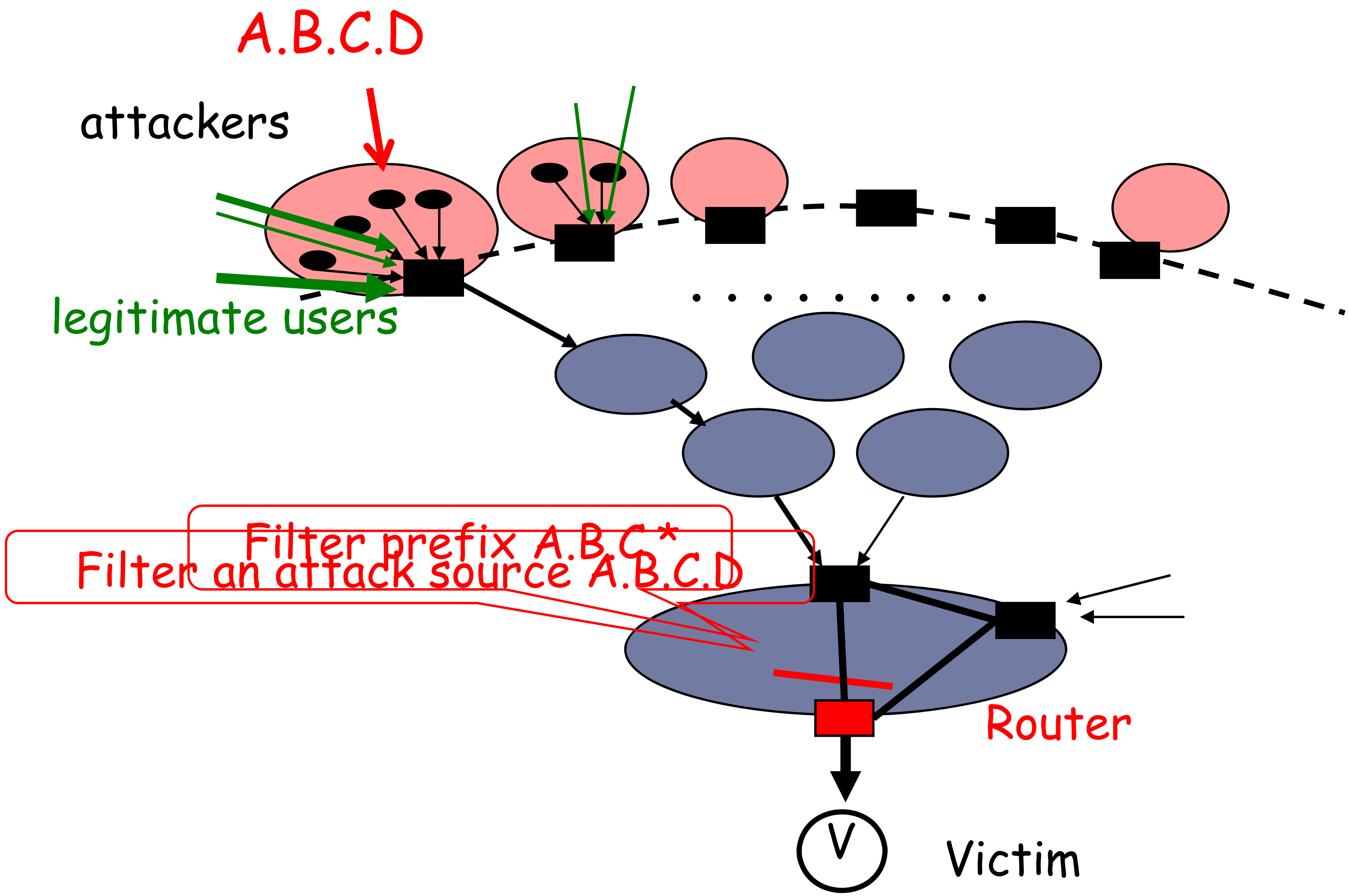
- Filters implemented in TCAM
  - are a limited resource
    (< tens of thousands per router)

There are fewer filters than attack sources

http://www.microimages.com/getstart/imgs/filter.jpg

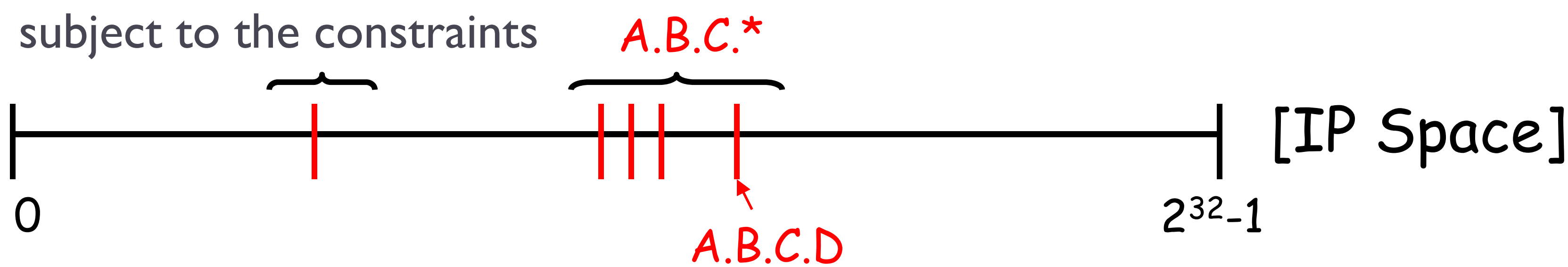
Filter Selection at a Single Router

tradeoff: number of filters vs. collateral damage



Our Goal: Filter Selection

- 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,
    - and possibly other constraints, e.g., link capacities
  - the operator's policy
- produce a compact set of filtering rules:
  - so as to optimize the operator's objective
    - (e.g. filter as many malicious and as few legitimate sources)
  - subject to the constraints



Filter Selection Problem
Notations

- *p/l* : IP prefix
- w_i : weight assigned to IP address *i*
  - <0 "bad" (blacklisted) addresses; >0 for "good" addresses
  - amount of flow sent
  - importance assigned by the operator
    - e.g. monetary loss (gain) in filtering out that address
- $x_{p/l} \in \{0, 1\}$: decision variable
  - indicates whether or not we filter out IP sources in prefix p/l
- F_{max} : maximum number of available filters

Filter Selection as a Knapsack Problem

A General Framework

"price/cost"

Knapsack "capacity"

$$\min \sum_{p/l} \sum_{i \in p/l} w_i \, x_{p/l}$$

Same "weights"

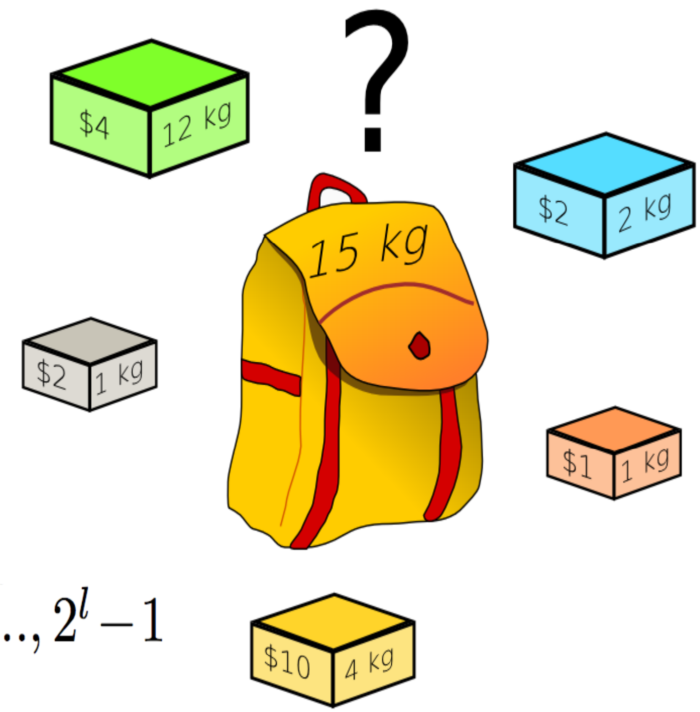
$$\text{s.t.} \sum_{p/l} 1 \, x_{p/l} \leq F_{max}$$

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

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

Correlation of knapsack "items"

Knapsack "items"



Filtering Problems

Overview

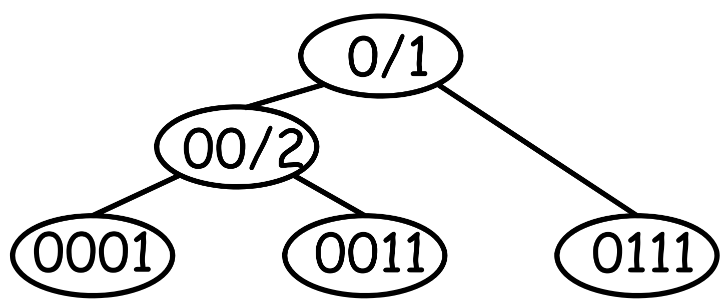
FILTER-ALL	**FILTER-SOME**	
	FLOODING	Add constraint on (single) link capacity
	DISTRIBUTED FILTERING	Multiple routers

Longest Common Prefix (LCP) Tree

- Definition
  - the binary tree whose **leaves** are **addresses in BL**, and **intermediate nodes** represent all and only the **longest common prefixes** between addresses in BL

- Example
  - For 4bit addresses, BL={1,3,7}, the LCP-Tree(BL) is:

1 = 0001
3 = 0011
7 = 0111



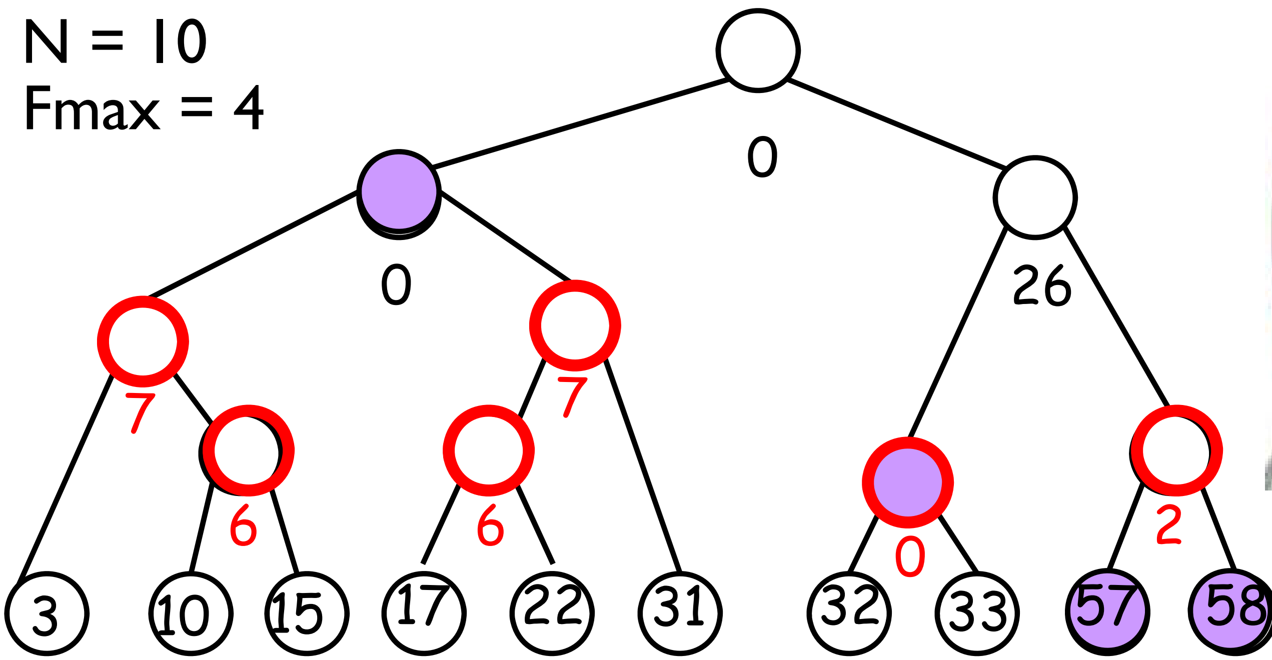
FILTER-ALL
Problem Statement

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

FILTER-ALL	GENERAL FRAMEWORK
$\min \sum_{p/l} g_{p/l} x_{p/l}$ $g_{p/l} = \sum_{i \in p/l \cap \text{whitelist}} w_i$	$\min \sum_{p/l} \sum_{i \in p/l} w_i \cdot x_{p/l}$
s.t. $\sum_{p/l} x_{p/l} \le F_{max}$	s.t. $\sum_{p/l} x_{p/l} \le F_{max}$
$\sum_{p/l: i \in p/l} x_{p/l} = 1 \quad \forall i \in \mathcal{BL}$	$\sum_{p/l: i \in p/l} x_{p/l} \le 1 \quad \forall i \in \mathcal{BL}$
$x_{p/l} \in \{0,1\} \quad \forall l = 0, ..., 32, p = 0, ..., 2^l$	$x_{p/l} \in \{0,1\} \quad \forall l = 0, ..32, p = 0, ..2^l$

FILTER-ALL
Simple greedy strategies do not work

N = 10
Fmax = 4



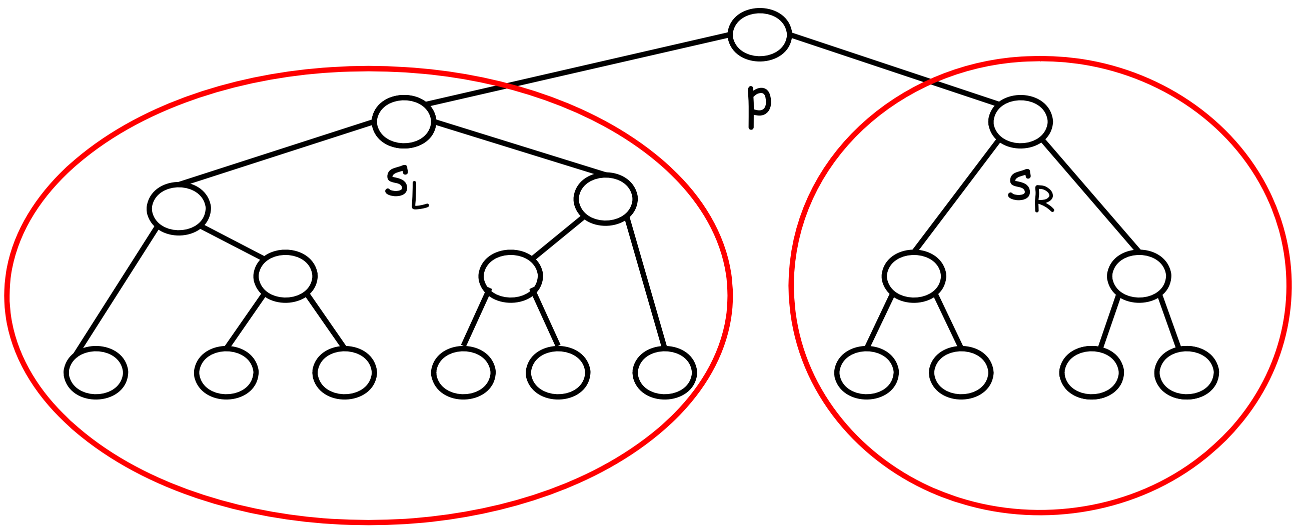
Merging (N-Fmax) closest leaves: 28

Optimal solution: 26

http://cache.consumerist.com/assets/resources/2007/08/con_greedymoneyman.jpg

FILTER-ALL
DP Algorithm (1)

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



F-n≥1, filters within left subtree

n≥1, filters within right subtree

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

FILTER-ALL
DP Algorithm (2)

Algorithm:

- Build LCP-Tree(BL)
- Initialize leaves: $z_{leaf}(F)=0$, F=1,…,Fmax
- For all other nodes:

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

- Return: z_{root}(Fmax)

Analysis:

- Optimality
- Complexity: **linearly increasing with size of blacklist**, N:
  - O(mN) + O(FmaxN), where m=32 (length of bit string) and Fmax<<N

FILTER-SOME
Problem Statement

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

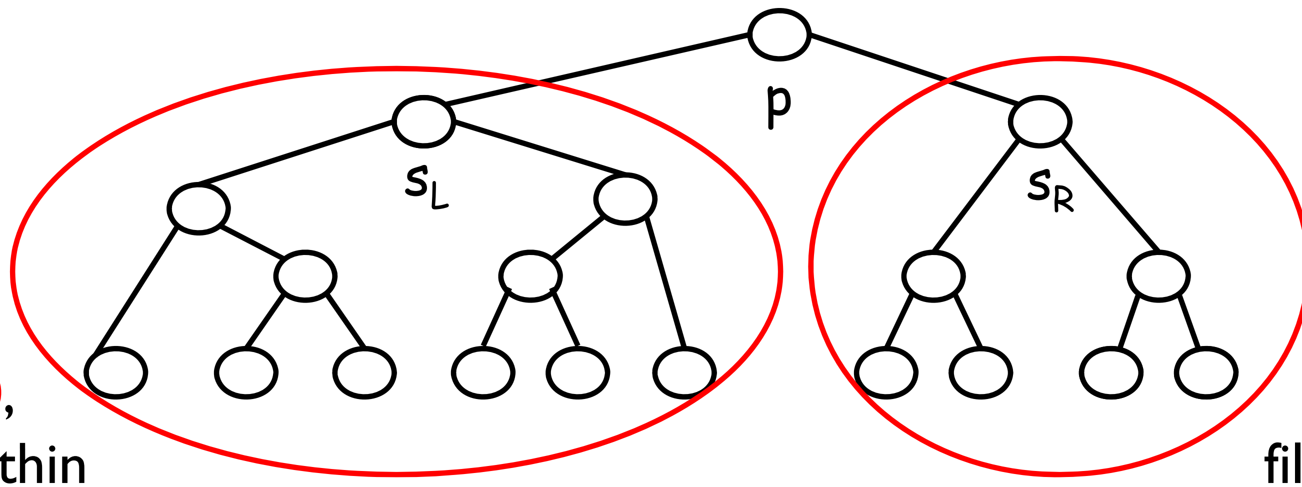
FILTER-SOME	FILTER-ALL
$\min \sum_{p/l} (g_{p/l} - b_{p/l}) x_{p/l}$	$\min \sum_{p/l} g_{p/l} x_{p/l}$
s.t. $\sum_{p/l} x_{p/l} \le F_{max}$	s.t. $\sum_{p/l} x_{p/l} \le F_{max}$
$\sum_{p/l: i \in p/l} x_{p/l} \le 1 \quad \forall i \in \mathcal{BL}$	$\sum_{p/l: i \in p/l} x_{p/l} = 1 \quad \forall i \in \mathcal{BL}$
$x_{p/l} \in \{0,1\} \quad \forall l = 0, ..., 32, p = 0, ..., 2^l$	$x_{p/l} \in \{0,1\} \quad \forall l = 0, ..., 32, p = 0, ..., 2^l$

$$g_{p/l} = \sum_{i \in p/l \cap \text{whitelist}} w_i$$

$$b_{p/l} = \sum_{i \in p/l \cap \text{blacklist}} |w_i|$$

FILTER-SOME
DP Algorithm

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



F-n≥0,
filters within
left subtree

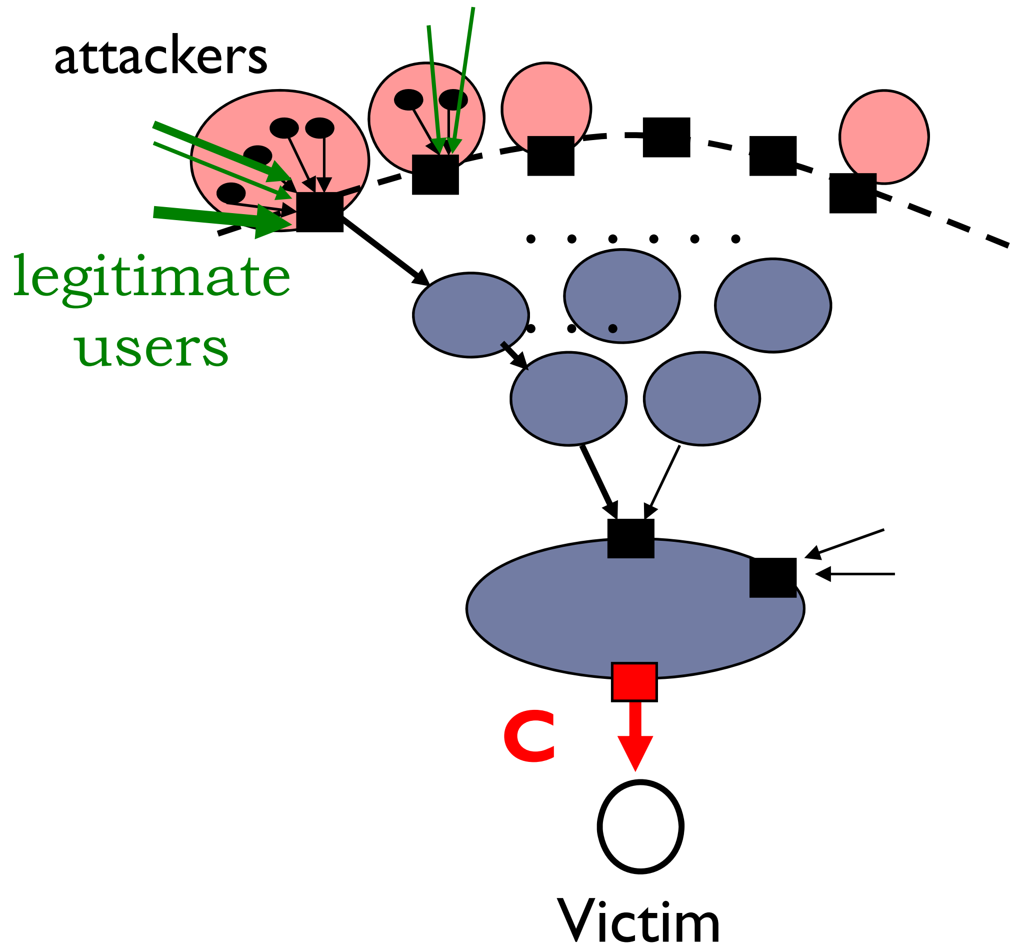
n≥0,
filters within
right subtree

$$z_p(F) = \min_{n=0,\dots,F} \left\{ z_{s_l}(F-n) + z_{s_r}(n) \right\}$$

n=0,1,…,F: means we may not block all malicious Ips (leaves)

FLOODING
Motivation

- DDoS: Malicious hosts coordinate to flood the access link to a victim
- **Weights** of every address represent the traffic **volume**
- Bound on link **capacity, C**



FLOODING
Problem Statement

- Given: a blacklist BL, a set of legitimate sources, 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 fit the total traffic within the link capacity

FLOODING	FILTER-SOME
$\min \sum_{p/l} g_{p/l} x_{p/l}$	$\min \sum_{p/l} \left(g_{p/l} - b_{p/l}\right) x_{p/l}$
s.t. $\sum_{p/l} x_{p/l} \leq F_{max}$	s.t. $\sum_{p/l} x_{p/l} \leq F_{max}$
$\sum_{p/l} \left(g_{p/l} + b_{p/l}\right)(1 - x_{p/l}) \leq C$	$\sum_{p/l: i \in p/l} x_{p/l} \leq 1 \quad \forall i \in \mathcal{BL}$
$\sum_{p/l: i \in p/l} x_{p/l} \leq 1 \quad \forall i \in \mathcal{BL}$	

coefficient changes per every variable

FLOODING
Solution

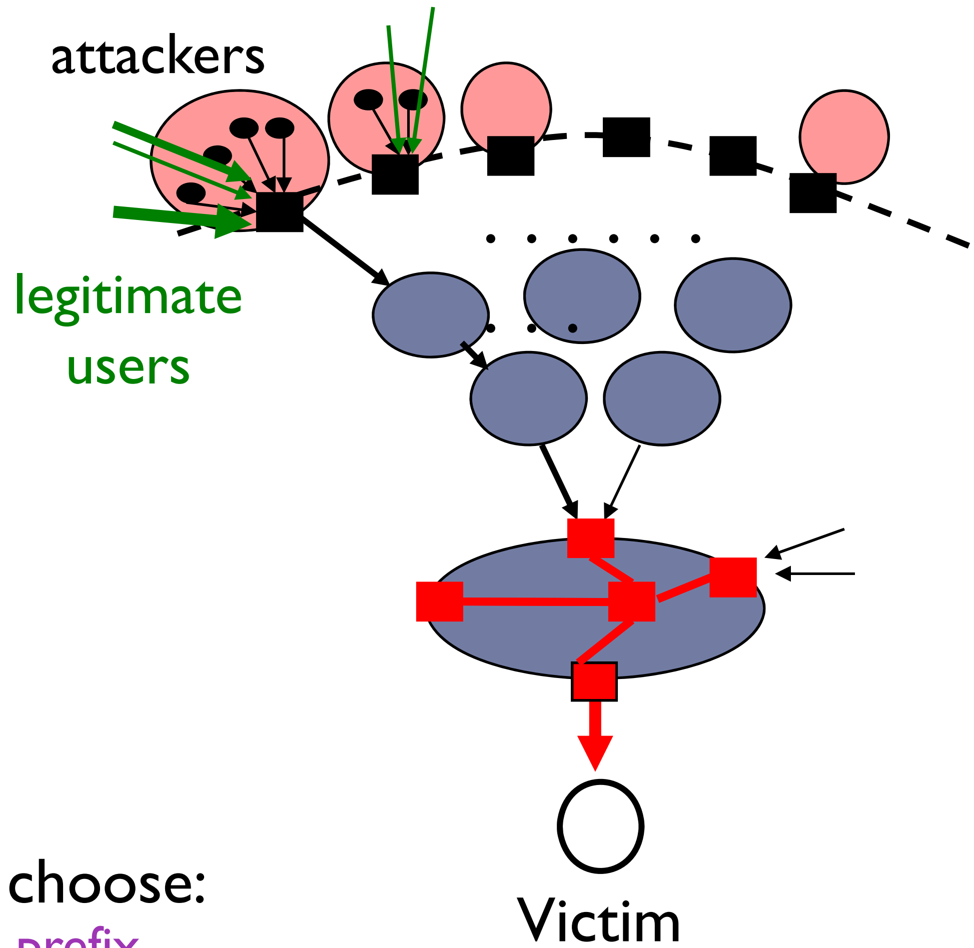
- FLOODING is NP-hard
  - reduces to knapsack with cardinality constraint

- An optimal pseudo-polynomial algorithm, solves the problem in: O(CN)
  - similar to the DP for FILTER-ALL/SOME
  - extended to take into account the capacity constraint

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

DISTRIBUTED FILTERING for FLOODING
Motivation

- A single network (ISP or enterprise) may deploy filters on several routers
  - increase filter budget
- 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 Statement

DISTRIBUTED FILTERING	FLOODING
$\min \sum_{u\in\mathcal{R}} \sum_{p/l} g_{p/l}^{(u)} x_{p/l}^{(u)}$	$\min \sum_{p/l} g_{p/l} x_{p/l}$
s.t. $\sum_{p/l} x_{p/l}^{(u)} \leq F_{max}^{(u)} \quad \forall u \in \mathcal{R}$	s.t. $\sum_{p/l} x_{p/l} \leq F_{max}$
$\sum_{p/l} \left(g_{p/l}^{(u)} + b_{p/l}^{(u)}\right)(1 - x_{p/l}^{(u)}) \leq C^{(u)} \quad \forall u \in \mathcal{R}$	$\sum_{p/l} \left(g_{p/l} + b_{p/l}\right)(1 - x_{p/l}) \leq C$
$\sum_{u\in\mathcal{R}} \sum_{p/l\ni i} x_{p/l}^{(u)} \leq 1 \quad \forall i \in \mathcal{BL}$	$\sum_{p/l:i\in p/l} x_{p/l} \leq 1 \quad \forall i \in \mathcal{BL}$

Couples the routers!

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

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

Conclusion

- Introduce a framework to model filter selection as a resource allocation problem
- Designed and analyzed efficient algorithms to solve filter selection problems