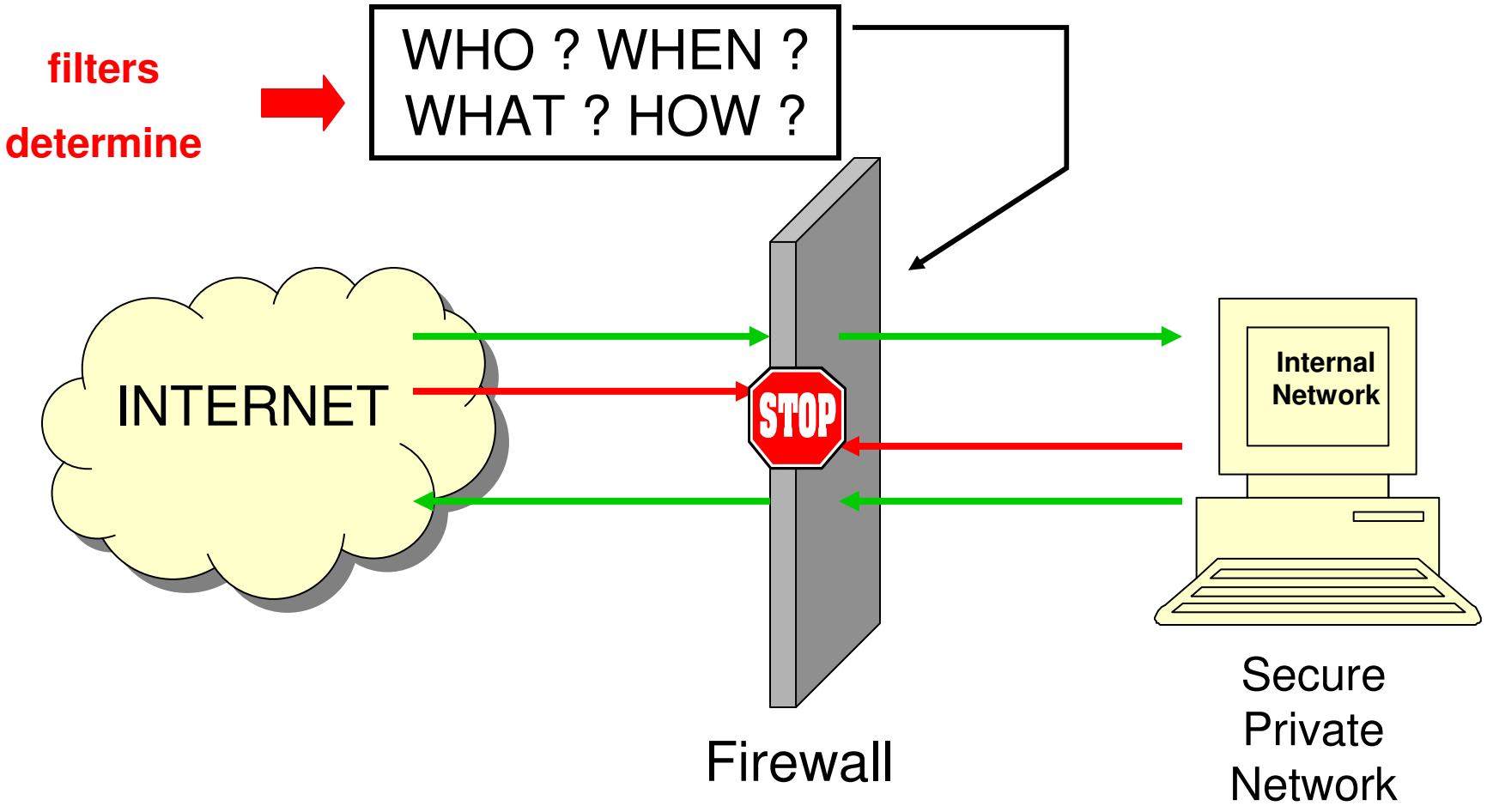
OPTimizing fireWALL
A Hierarchical Traffic Aware Firewall

Subrata Acharya, Mehmud Abliz,
Bryan Mills, Taieb Znati
Department of Computer Science
University of Pittsburgh, PA

Jia Wang, Zihui Ge
AT&T Labs Research, Florham Park, NJ
Albert Greenberg
Microsoft Research, Redmond, WA

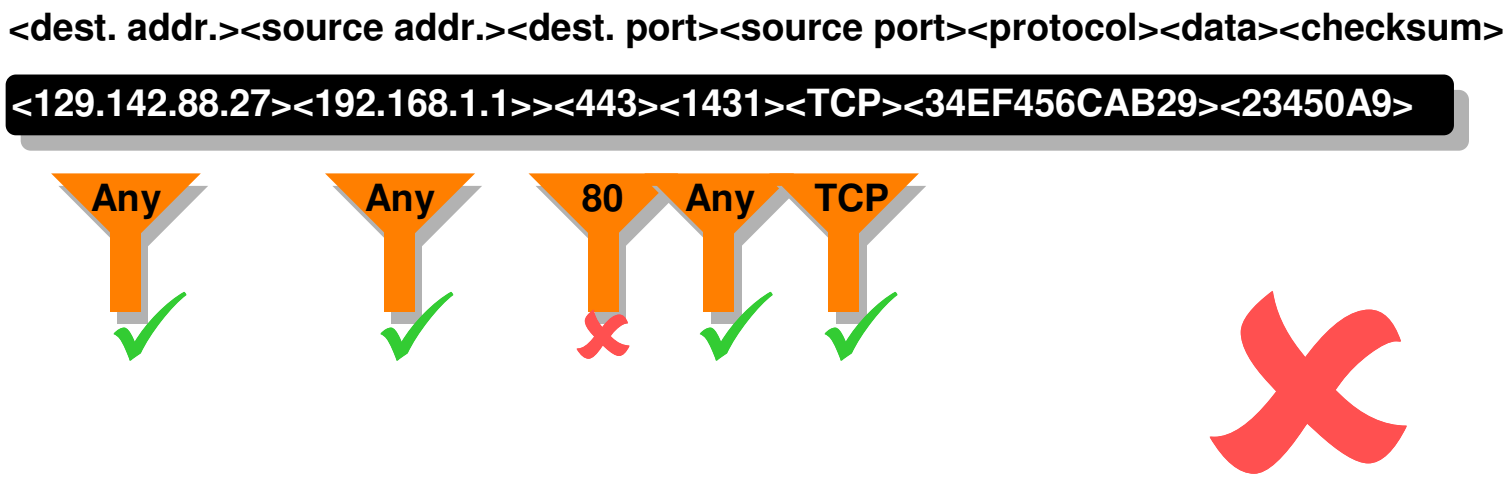
What is a Firewall?



A security system that acts as a protective boundary between a **network** and the **outside world**

Packet Filtering …

- Below is a rule defining that only IP packets normally used for web browser traffic are allowed to reach any computer on the Internet.



Background and Motivation

- High demand on reducing packet matching time
- Deterministic design
  - Traffic characteristics not considered
- Rules non-adaptive to traffic (**bottlenecked under attack**)
- Rules added independently on per project basis (**redundancy**)
- Rules hardly reordered in fear of violating integrity (**sub-optimal**)

Background and Motivation

- Optimizing Firewall is hard
  - Enterprise Firewalls have **5K-15K** multi-dimensional rules
  - Inherent complexity - **multi-dimensional** rules
  - Zero-tolerance on semantic integrity violation

- Most security devices still match rules **sequentially** or are **'list-based'** firewalls
- Design of non-linear **K**-partition (**K** > 2) policy subsets is NP hard
- Finally, exploiting **'dynamism'** in Internet traffic is a very challenging problem

Problem Statement

- Build a management tool to analyze and optimize firewall policies based on traffic dynamics
- Develop non-linear optimization models
- Traffic-aware optimizations to aid
  - Policy optimization - long term traffic profile
  - Dynamic policy adaptation - short term traffic anomaly

Outline

- Background and Motivation
- Problem Statement
- Drawbacks of present solutions
- Proposed Solution **- OPTWALL**
- Multi dimensional (N,K) splitting approaches
- Evaluation
- Summary and Future Work

Present Solutions

- Policy modeling and optimization without traffic characteristics taken into consideration
- Lack of multi-dimensional policy optimization
- Can handle very few policy size (< 200)
- Linear or list-based optimization

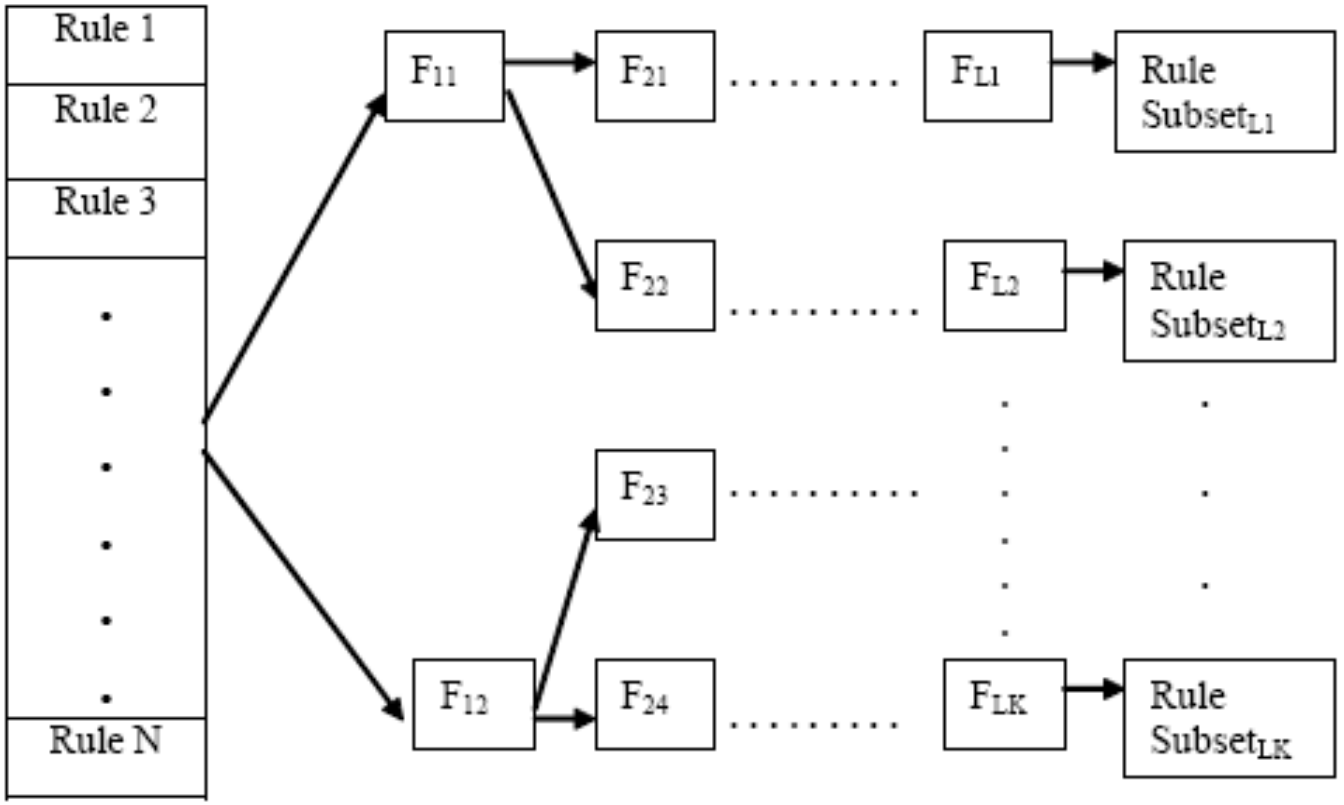
Contributions

- Design of OPTWALL
- Multi dimensional splitting solutions
  - Optimal and Heuristic
- Multi-dimensional filters considered, nearly one million
- Adaptive traffic-aware protocol to defend against attacks
- Evaluation study to access the potential of OPTWALL

OPTWALL – Design Goals

- Reduce operational cost of firewalls
- Preserve semantic integrity
- Dynamically maintain the optimal rule sub - structure with traffic changes

Basic operation of OPTWALL

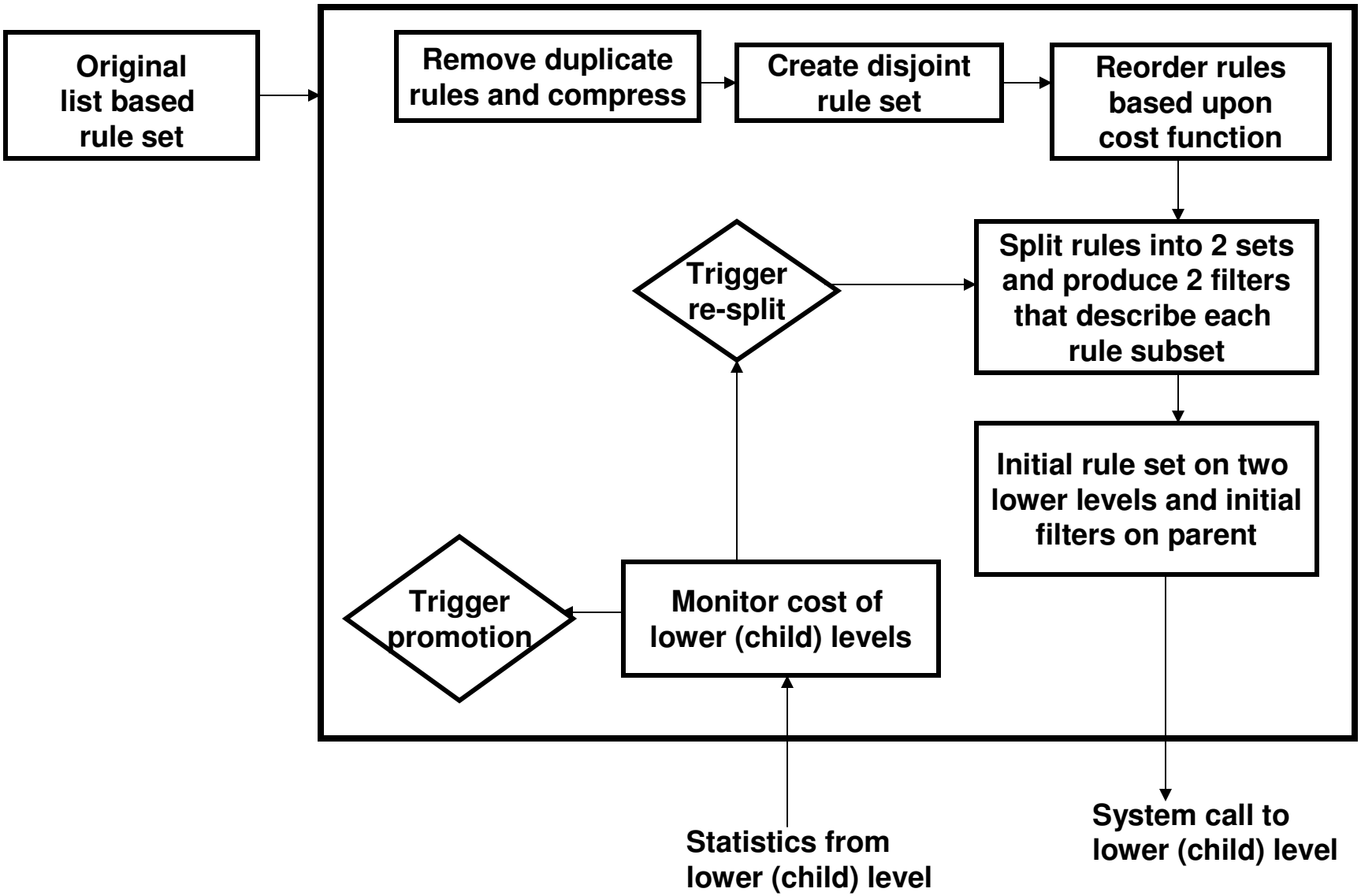


$F_i \cap F_j = \varphi$, $N >>> K$ (F = Filter)

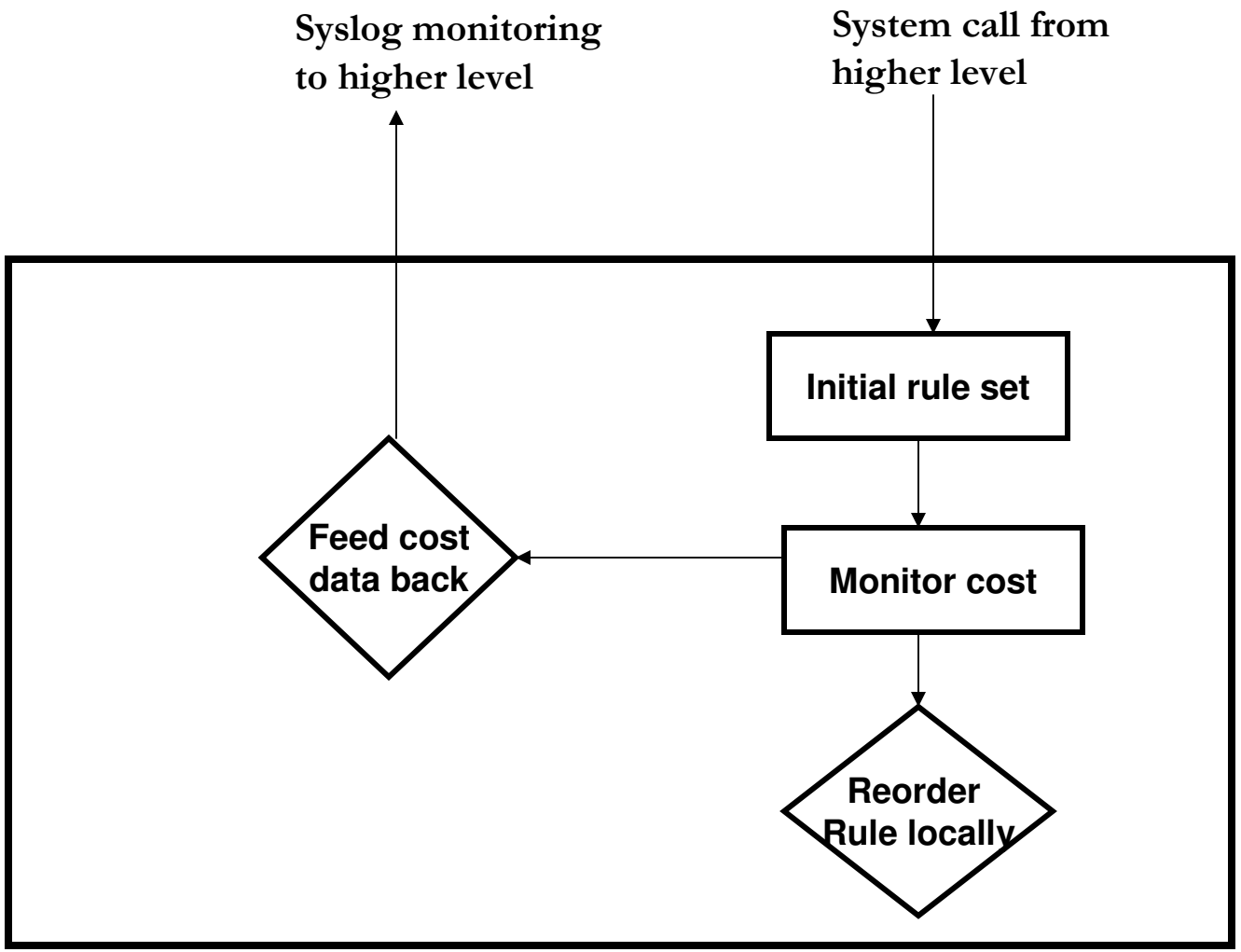
List Based Firewall Rule Set (N Rules)

OPTWALL Hierarchical K-Partition Rule Subsets

OPTWALL: Higher (parent) level



OPTWALL: Lower (child) level



OPTWALL Architecture

- Hierarchical structure building
  - Preprocessing
  - Ordering
  - Splitting
- Hierarchical structure maintenance
  - Re-ordering
  - Re-splitting
  - Promoting

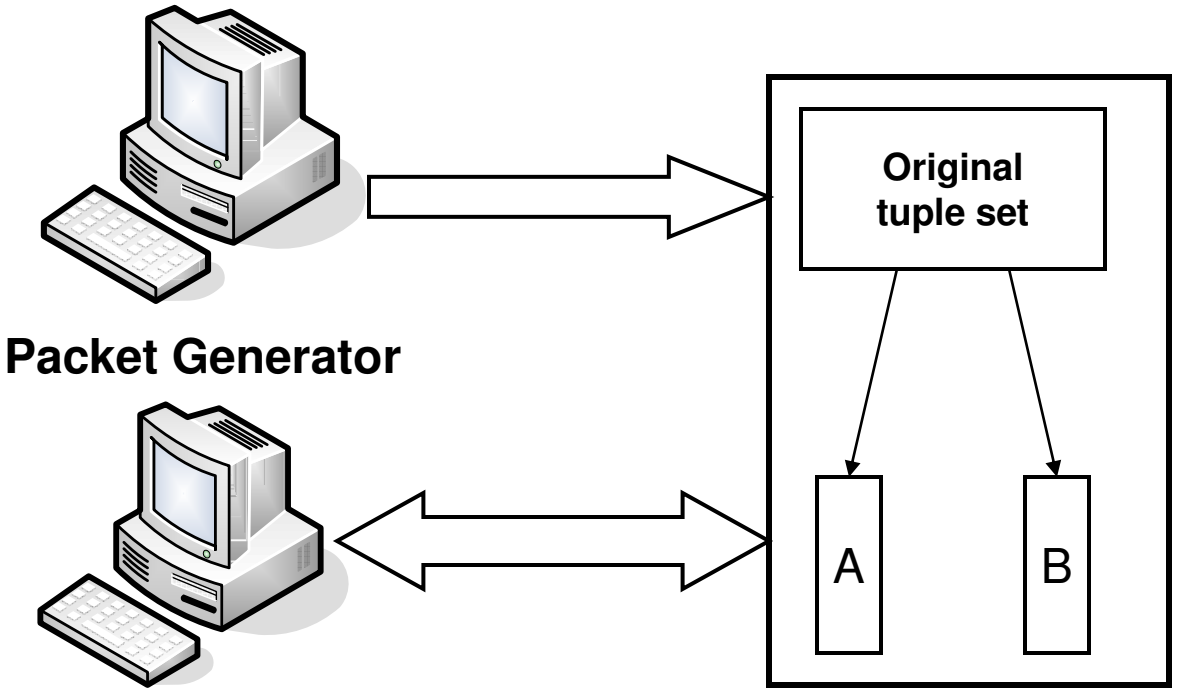
Splitting approach - Optimal

- Based on A* search technique
  - Tunable threshold to terminate search
  - Memory inefficient for filter size > 1000
    - Solution: Parallel A*, pruning
  - Complexity α 2^N, N = number of filters

Splitting approach - Heuristic

- Local greedy approach
  - Choice of initial filter
    - Hit count – hit count
    - Hit count – Max distance
    - Random – Random
    - Max distance – Max distance
  - Sub-optimal solutions
  - Complexity α N, the number of filters

Evaluation framework



Running AMD Athlon^{tm} 64 bit Processor 3000+ with Ubuntu Linux OS

Data

- Daily snapshots from ~**50 AT&T** partner networks
  - ~**1900** rules => Nearly **one million** tuples
  - Multi-dimensional rule set
    - <*src, dst, svc, action, comments, …*>
    - Each rule has multiple src/dst prefixes and services
  - First hit principle

- Traffic log
  - One entry per session (i.e., flow)
  - <*id;date;time;orig;type;action;alert;i/f_name;i/f_dir;product;src;dst;s_port;service;proto;…*>

Example

```
:rule (
        :src (
                : h_192.168.113.2
                : h_224.0.0.6
        )
        :dst (
                : h_192.168.113.1
                : h_224.0.0.6
        )
        :services (
                : ospf
                : traceroute
        )
        :action (
                : accept
        )
        ….
}
```

1;27Jul2005;23:59:04;192.168.113.2;log;accept;;qfe1;inbound;VPN1&FireWall1;135.89.152.116;192.168.10.35;53480;161;udp;;;;;40;;;

2;27Jul2005;23:59:04;192.168.113.2;log;drop;;qfe0;inbound;VPN1&FireWall1;192.168.112.1;192.168.57.2;;;icmp;;

3;27Jul2005;23:59:04;192.168.113.2;log;accept;;qfe1;inbound;VPN1&FireWall1;135.89.152.116;192.168.98.105;;;icmp;;;;;3;;;

Metric

- Cost (rule$_i$) =

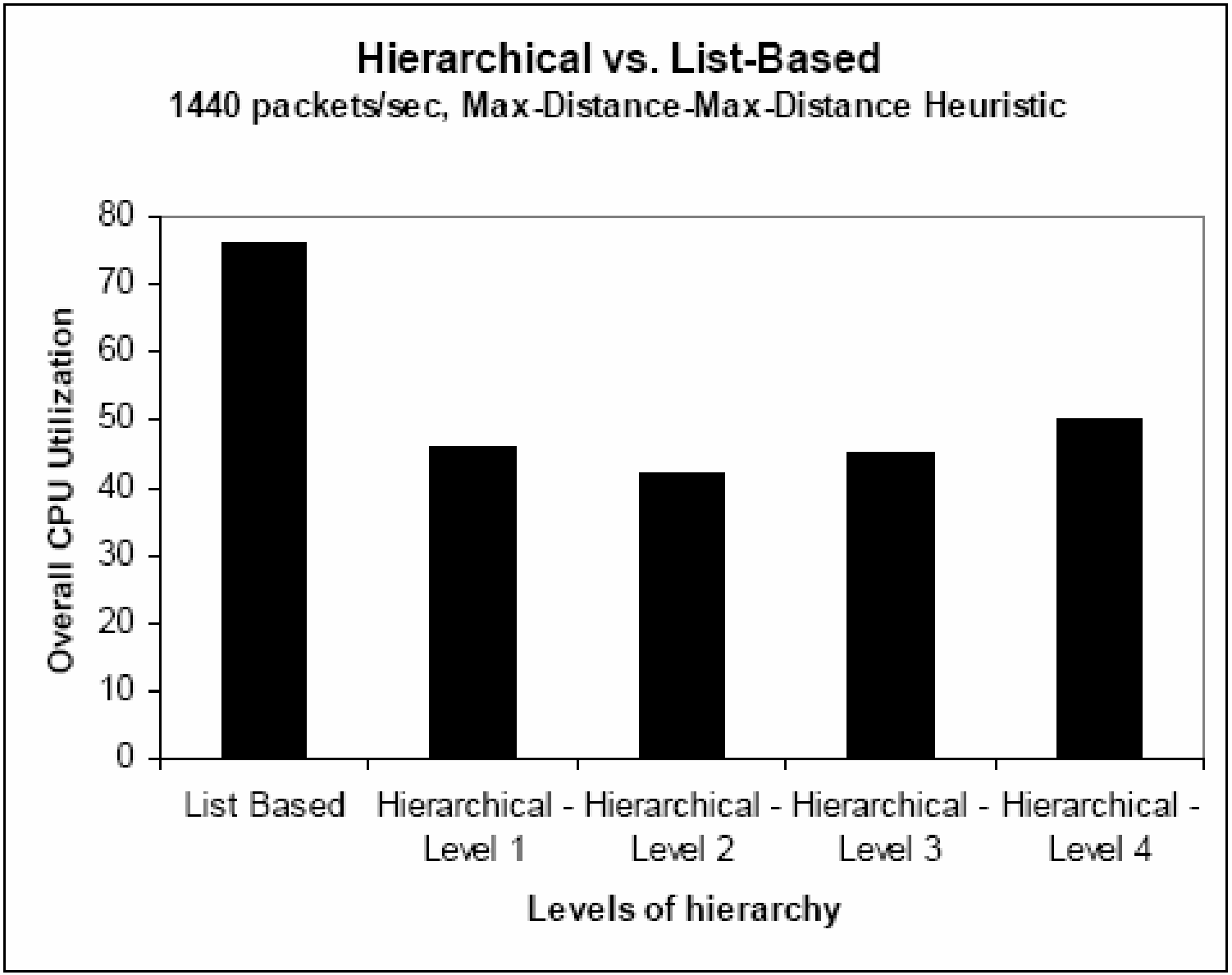
  hit-count (rule$_i$) * $\sum_{\{k=1,\ldots,i-1\}}$ (size(rule$_k$))

  – Cost of rule$_i$ depends on rule rank, rule size, and hit count
  – Complexity of matching is proportional to size of a rule

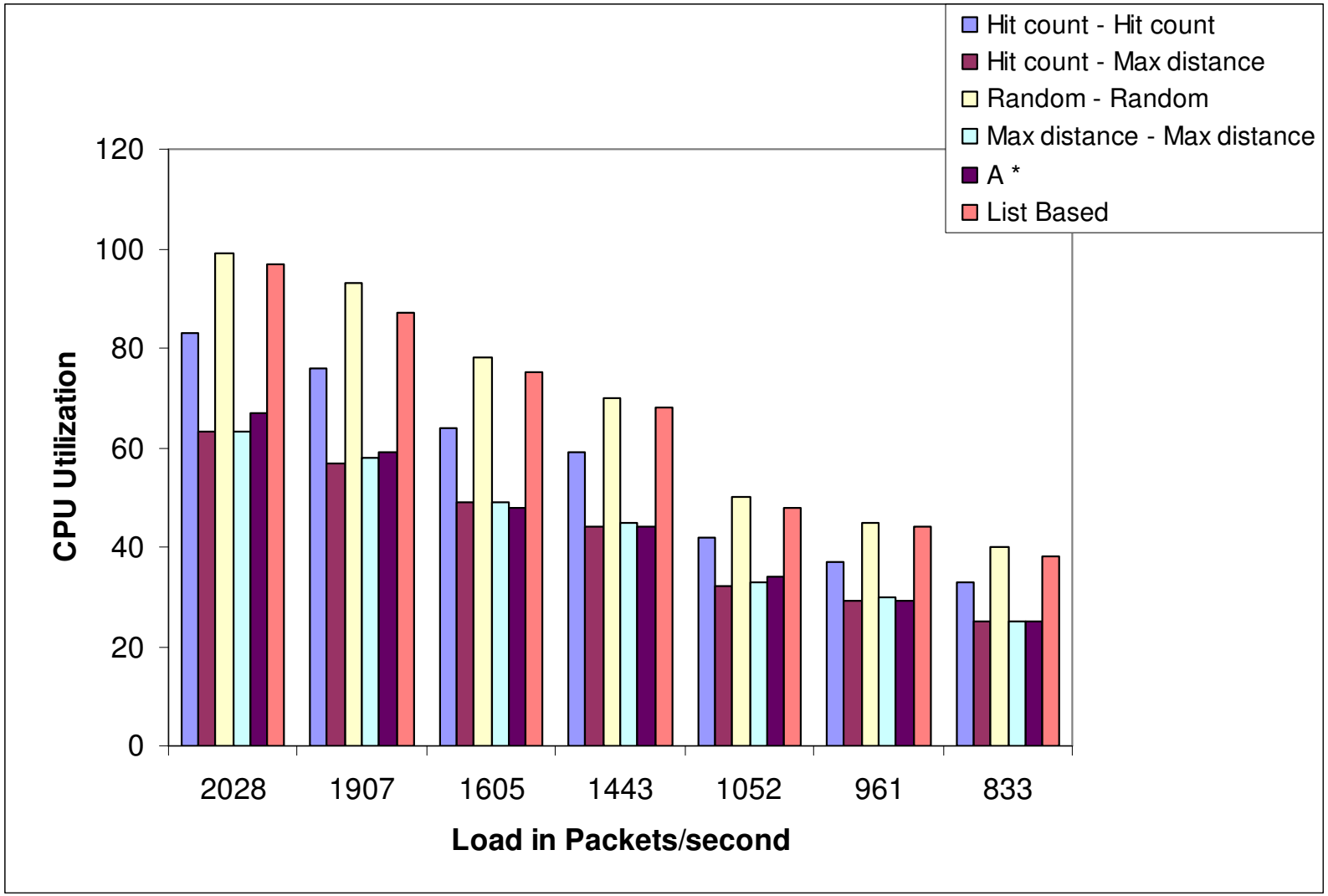
- Size (rule) =

  a_1 * $\sum$ {number of bits to represent source prefix } +
  a_2 * $\sum$ {number of bits to represent destination prefix} +
  β * {number of services * (number of bits to represent protocol + number of bits to represent port number)},
  $a_1 = a_2 = \beta = 1$

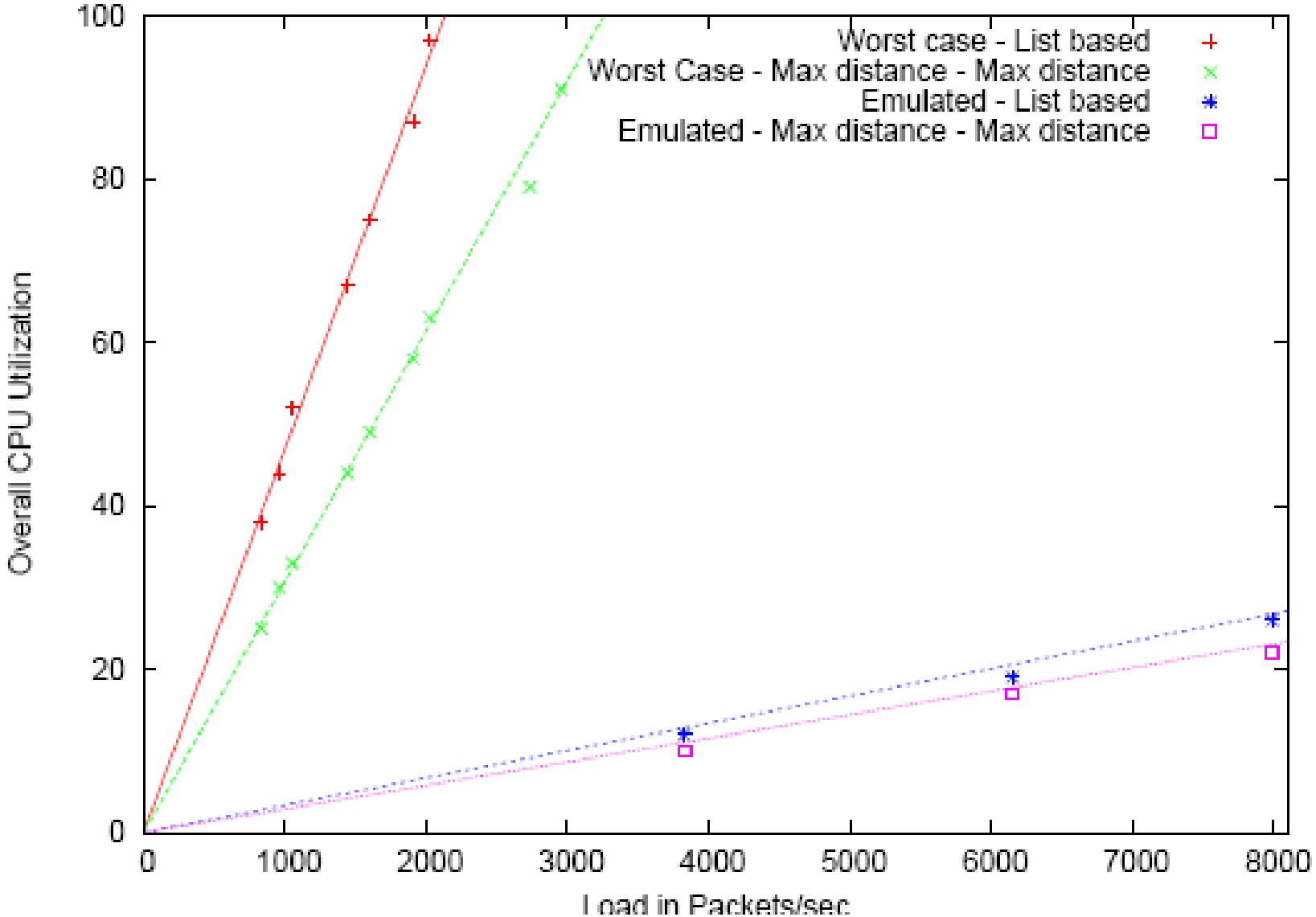
Results


Hierarchical vs. List-Based
1440 packets/sec, Max-Distance-Max-Distance Heuristic
Overall CPU Utilization
80
70
60
50
40
30
20
10
0
List Based
Hierarchical - Level 1
Hierarchical - Level 2
Hierarchical - Level 3
Hierarchical - Level 4
Levels of hierarchy

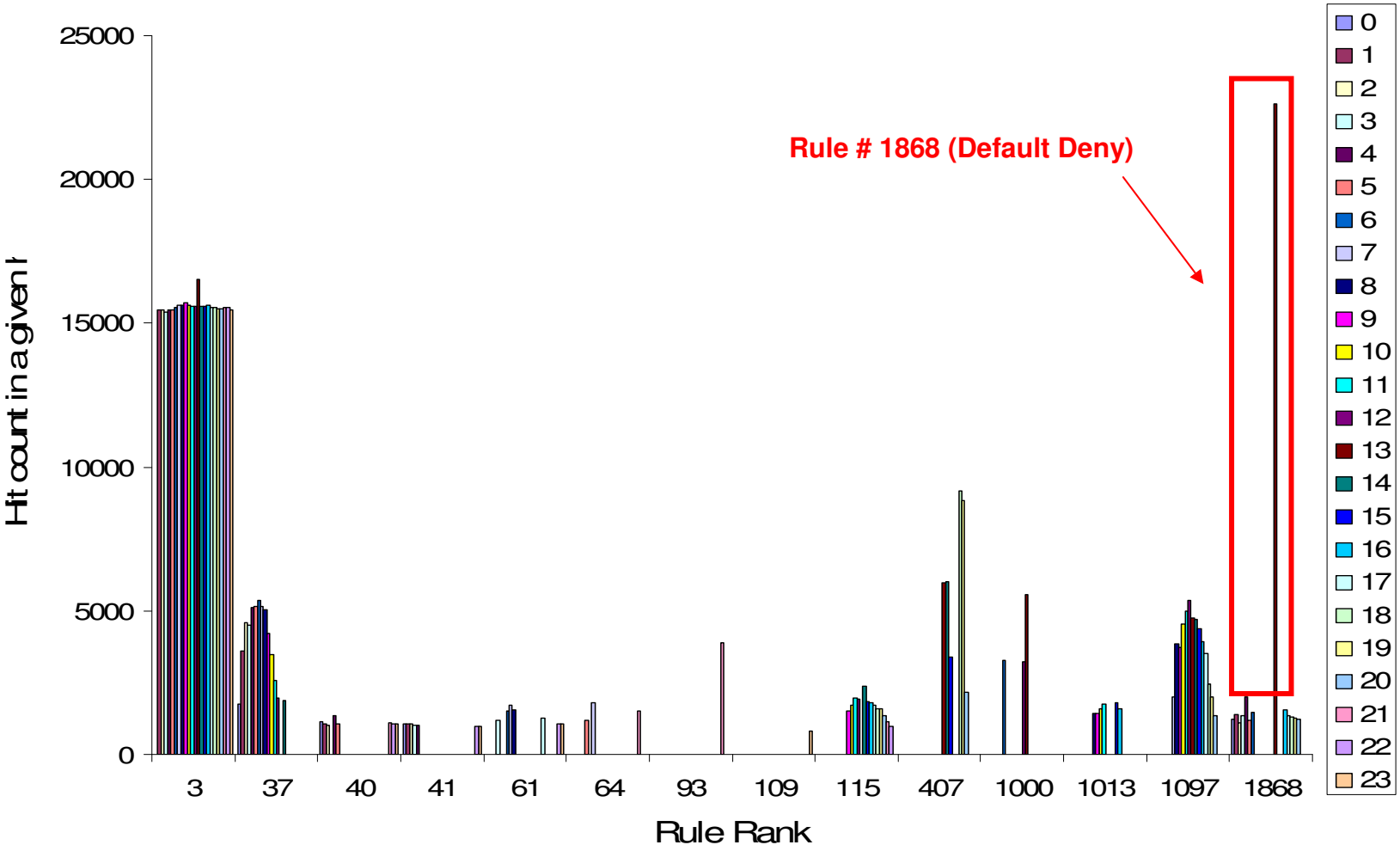
Worst case performance



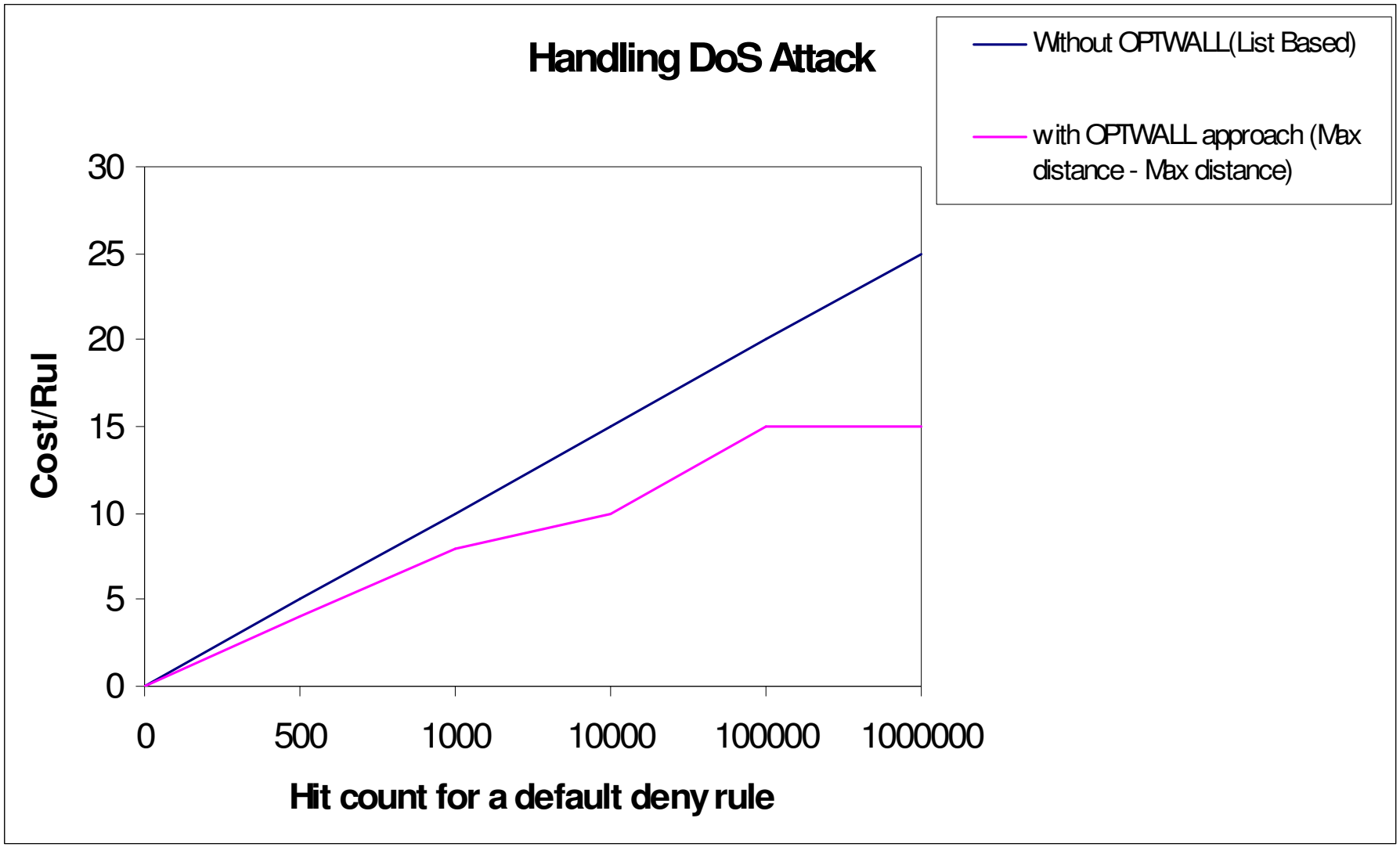
Emulated Traffic performance



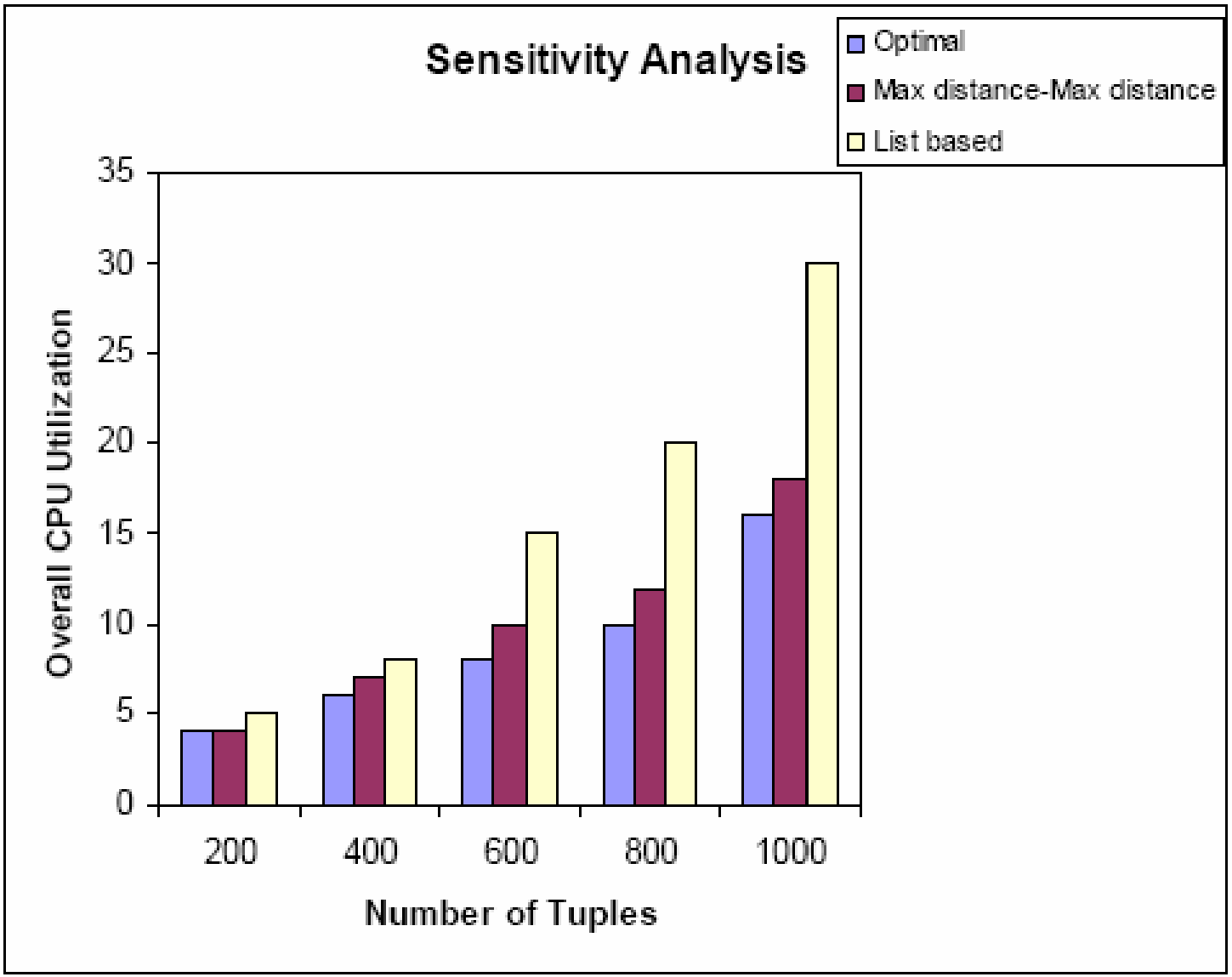
Handling DoS attacks



Handling DoS attacks (cont ..)



Sensitivity Analysis



Summary of contributions

- Hierarchical traffic aware optimization framework
- Multidimensional rule splitting solutions (optimal and heuristic)
- Nearly one million multi-dimensional filters considered
- Adaptive traffic-aware protocol to defend against attacks
- Evaluation study demonstrates potential of OPTWALL

Future Work

- Extend the OPTWALL onto physically distributed firewalls
  - Collaborate to prevent distributed attacks
- Evaluate the strength of OPTWALL with other commercial firewalls and firewall data sets
- Analysis & design of an improved cost metric

Thank You!!!