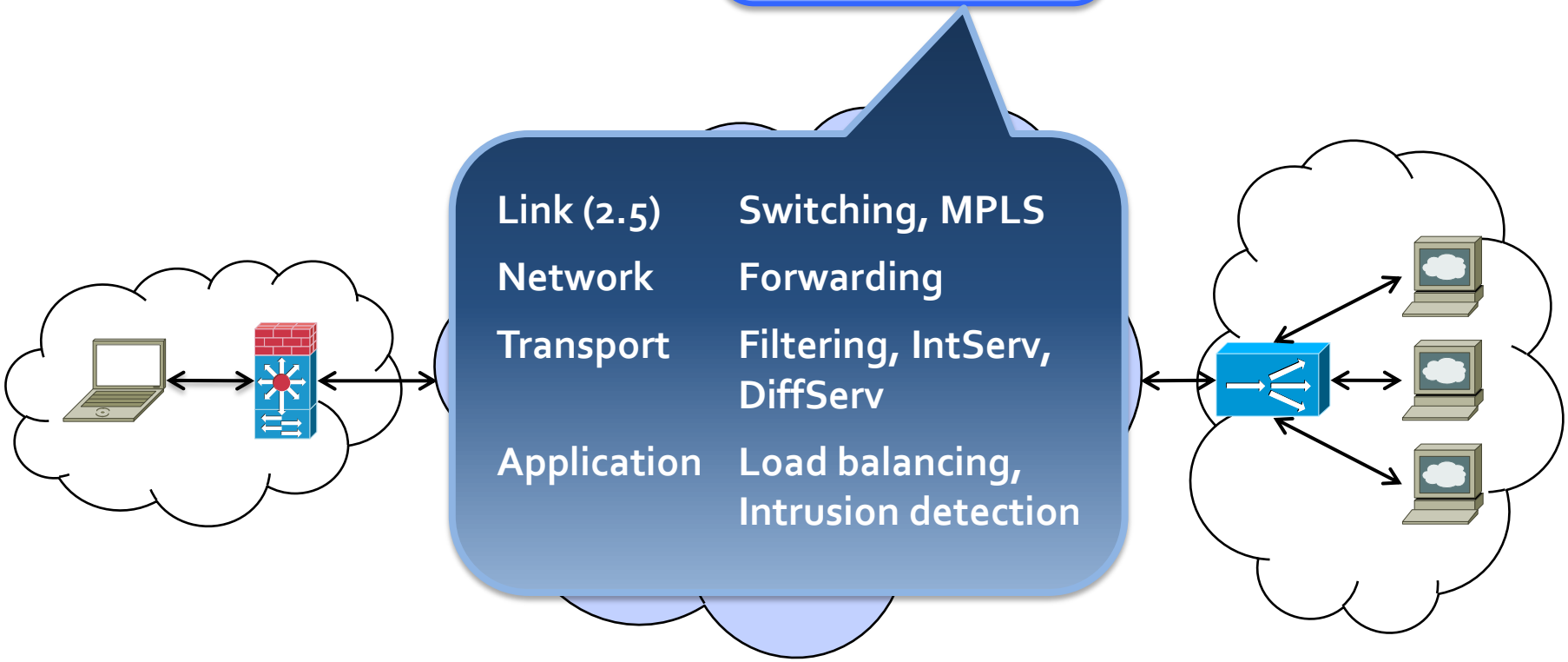
CLayer

Packet classification with explicit coordination

Mosharaf Chowdhury *with*
Sameer Agarwal, Dilip Joseph, and Ion Stoica

Motivation

Packet classification is everywhere



Problems

1

Existing approaches are point solutions for specific layer/service

2 3

Packet classification is expensive

- » Computation and memory requirements
- » Power hungry

4

Configuration complexity

- » Lack of coordination between entities involved

5

Semantic gap

Solution

CLayer is a cross-layer classification primitive

- » Generic mechanism to configure and implement capability-driven classification offloading
- » Explicit coordination between *classifiers* and *helpers*

"Classify once, verify thereafter"

Label-based per-flow classification

- » Labels are verifiable, confidential, and non-transferable

Outline

CLayer classification model

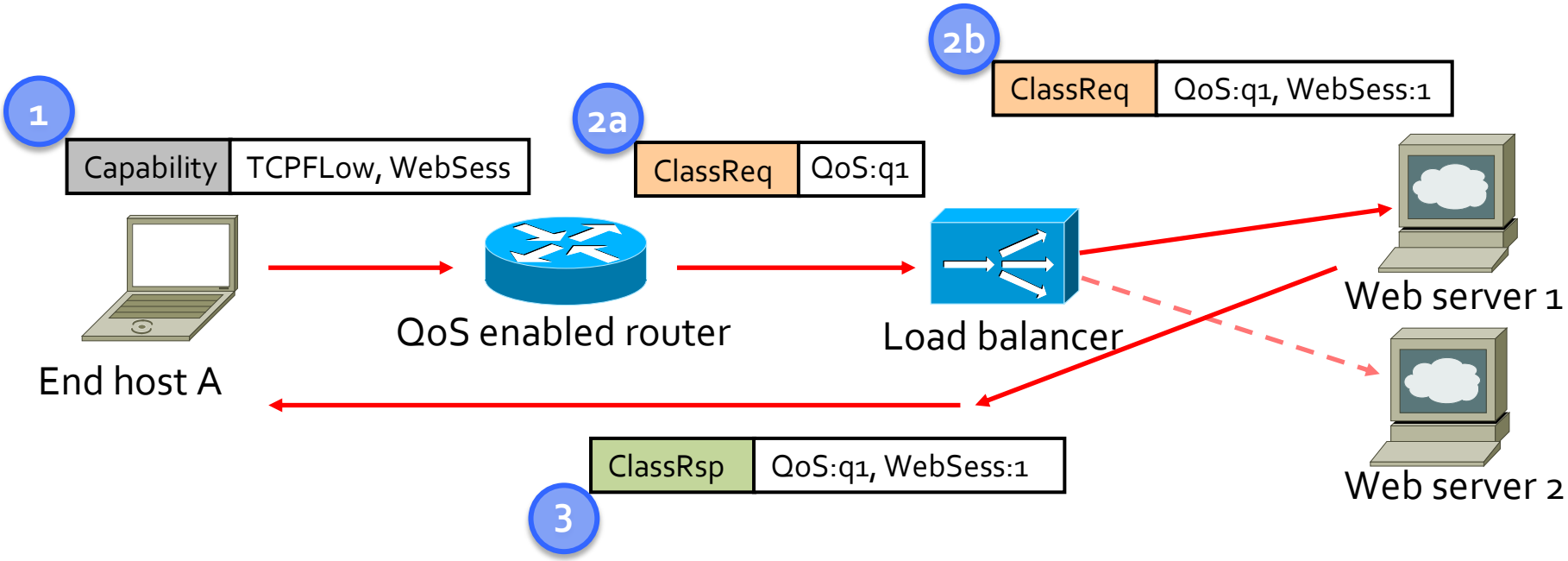
Fate-carrying labels (FCLs)

Implementation

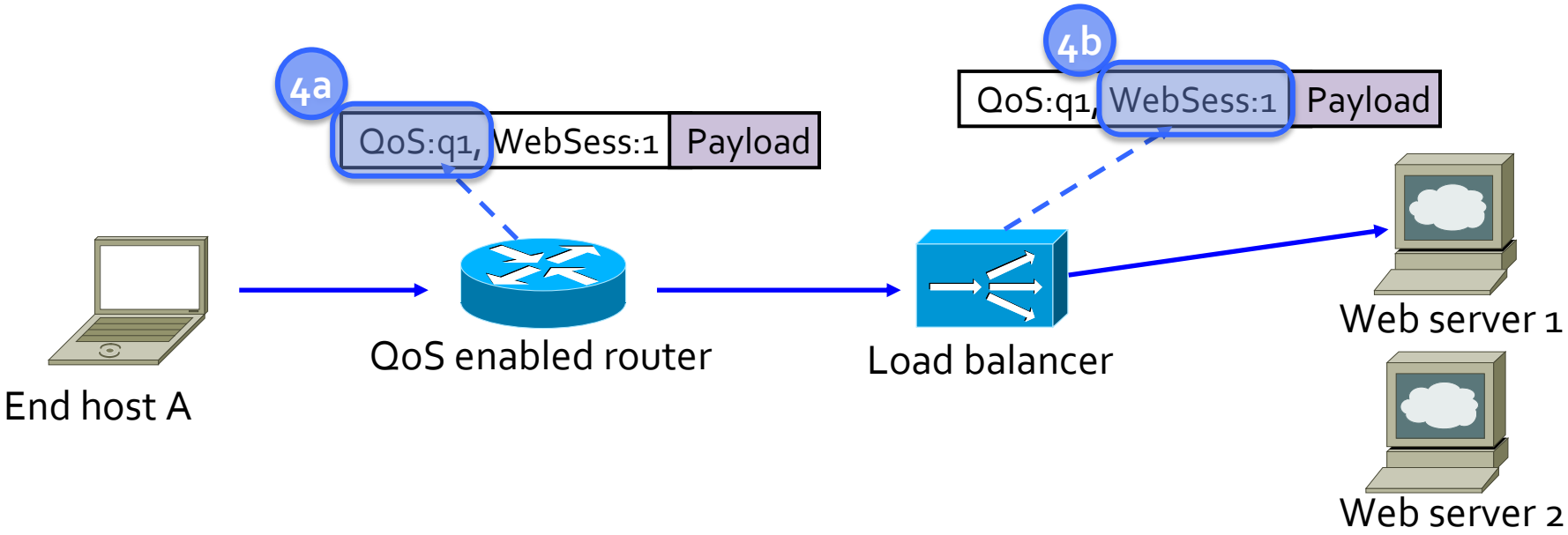
Results

Classification model

Control plane



Data plane


4a
QoS:q1,
WebSess:1
Payload
4b
QoS:q1,
WebSess:1
Payload
End host A
QoS enabled router
Load balancer
Web server 1
Web server 2

Fate-carrying labels

FCL basics

A *label* in CLayer is an opaque bag of bits

- Issued by a classifier for a particular flow
- Meaningful only to the issuer
- ‹*label* → *action*› lookup

A fate-carrying label carries the action itself

- No ‹*label* → *action*› lookup
- No states in classifiers

Requirements

Authenticity and Integrity
- » Verifiable and non-transferable
- » Unforgeable and single-use only

HMAC
Checksum

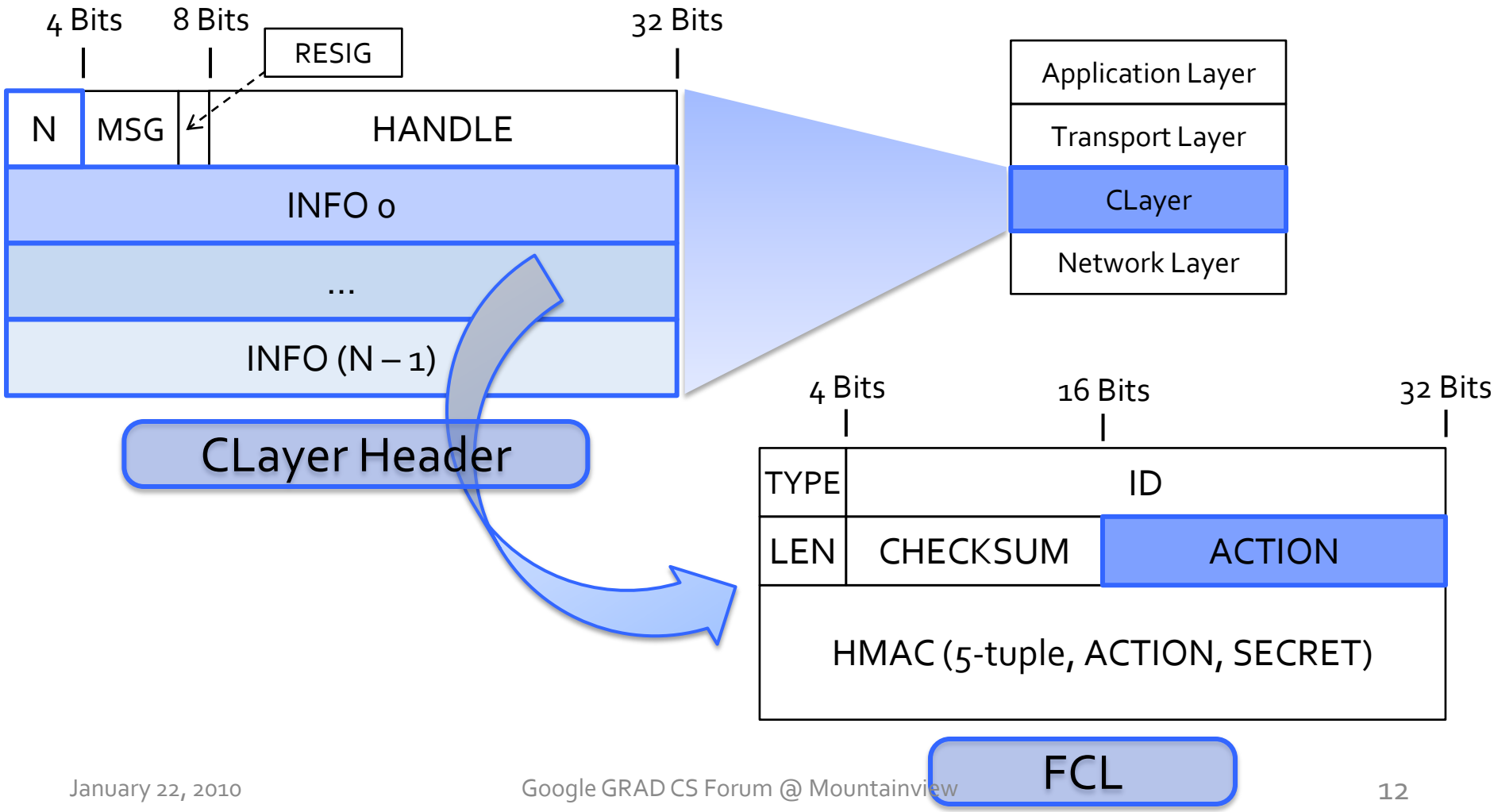
Confidentiality
- » Impossible to infer

Obfuscation
Periodic Invalidation

Performance
- » Not better off without CLayer

Line-speed hashing
Low overhead

Placement


4 Bits
8 Bits
RESIG
32 Bits
N
MSG
HANDLE
INFO 0
...
INFO (N – 1)
CLayer Header
Application Layer
Transport Layer
CLayer
Network Layer
4 Bits
16 Bits
32 Bits
TYPE
ID
LEN
CHECKSUM
ACTION
HMAC (5-tuple, ACTION, SECRET)
FCL

Implementation

Implementation stats

C++ Implementation using user level Click software router

Core components:

- » CLayer socket library and daemon (4025 lines)
- » Layer 4 firewall (308 lines)
- » Layer 4 load balancer (190 lines)

Ported applications:

- » *lighttpd*, *httperf*, *wget*, *nuttcp*, *elinks* (< 50 lines)

Results

Overheads

CLayer overheads at *helpers*:

- State: ~10 bytes per connection
- Processing: less than 1 µs

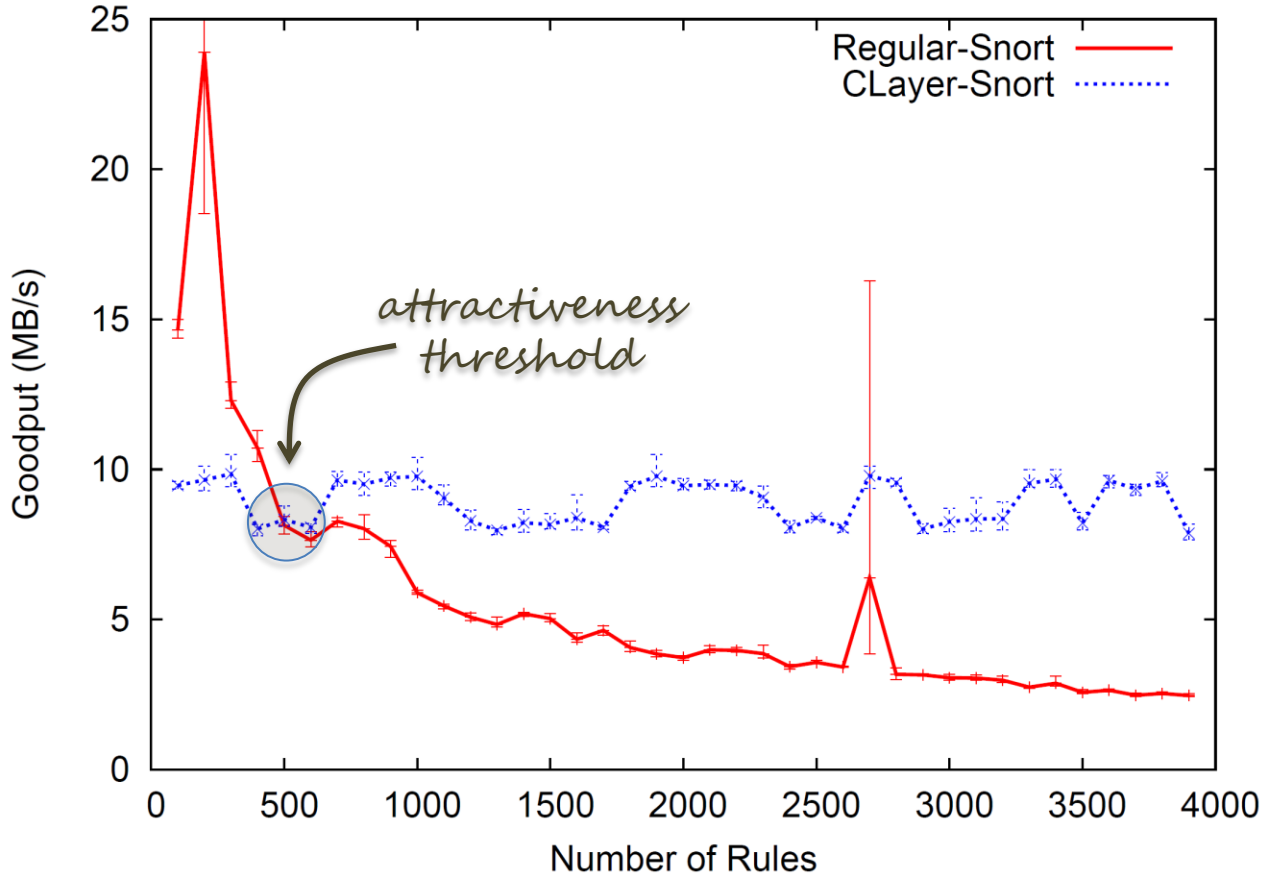
At *classifiers*:

- No state overheads
- Processing: varies in s/w and h/w implementations

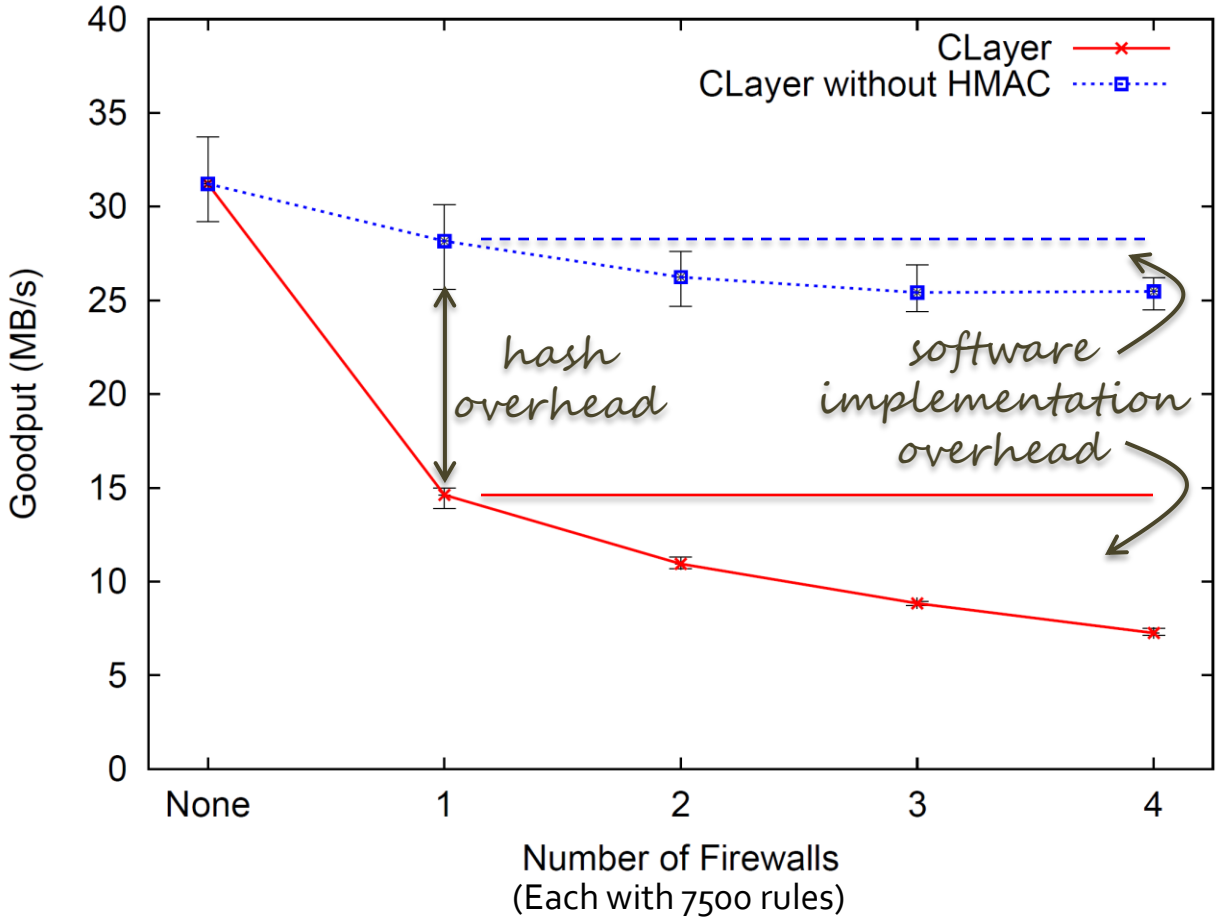
Per-packet overheads:

- Proportional to the number of labels
- Potential bottleneck

Performance



Multiple classifiers



Summary

Packet classification requires a dedicated layer

CLayer provides significant performance gain

- » 2-4 times increase in classifier throughput
- » Additional ~100% increase in throughput in trusted domains or with line-speed h/w hashing

CLayer adoption requires minimal change

- » Most suitable for controlled environments like data center and enterprise networks

Questions

Backup

CLayer handshaking