SymPerf:
Predicting Network Function Performance

European Research Council
Established by the European Commission

The Need for Predictability

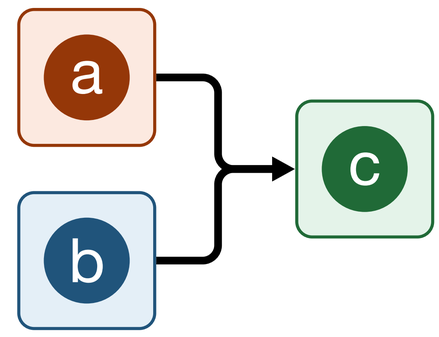
- Software-based on-path Network Functions
  - Would be very nice to have!
  - But operators fear uncertainty of code execution
    - Performance degradation
    - Buggy behavior
    - Interference with other flows/services
    - Generation of unwanted traffic
- To gain trust in software-based NFs, we need to:
  - Predict, assess, analyze, …, know!
  - The behavior and performance impact of a network function before deploying it
  - Ideally: a rigorous, precise and automated tool

High Level Overview

a Symbolic Analysis of NF Code
- Rigorous analysis of all feasible execution paths
- Set of all possible instruction chains
- Detect bugs, e.g., buffer-overflows

b Per-Platform Calibration
- Instruction cost and execution model

c Performance Prediction
- Predicts required computational effort of the NF
- For a given traffic pattern, best/worst case, equal distribution…
- Predicts impact on network ressources



SymPerf

Per Function

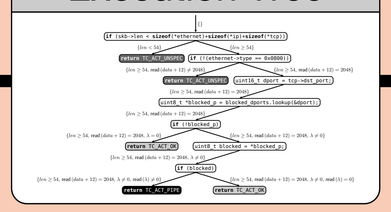
Network Function

```c
#include <bcc/proto.h>
#include <linux/pkt_cls.h>
#include <net/sock.h>
BPF_TABLE("hash", uint16_t, uint8_t, blocked_dports, 4096);

int act_main(struct __sk_buff *skb) {
    u8 *cursor = 0;
    struct ethernet_t *ethernet;
    struct ip_t *ip;
    struct tcp_t *tcp;
    if (skb->len < sizeof(*ethernet)+sizeof(*ip)+sizeof(*tcp)) return TC_ACT_UNSPEC;

    ethernet = cursor_advance(cursor, sizeof(*ethernet));
    if (!(ethernet->type == 0x0800)) return TC_ACT_UNSPEC;
    ip = cursor_advance(cursor, sizeof(*ip));
    if (ip->nextp != 0x06) return TC_ACT_UNSPEC;

    tcp = cursor_advance(cursor, sizeof(*tcp));
    uint16_t dport = tcp->dst_port;
    uint8_t *blocked_p = blocked_dports.lookup(&dport);
    if (!blocked_p) return TC_ACT_OK;
    uint8_t blocked = *blocked_p;
    if (blocked) return TC_ACT_PIPE;
    else return TC_ACT_OK;
}
```

1 Symbolically analyze the NF
- Generate Execution Tree
- All feasible paths

2 If bugs are found, fix NF

Execution Tree



3 Walk the Execution Tree
- Each Path corresponds to exactly one Instruction Chain
- Extract finegrained information

Instruction Chains

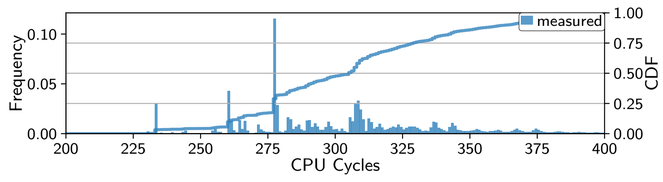
```
1 alloca
2 getelementptr
3 load
4 icmp
5 br
6 bpf_load_half
7 icmp
8 br
```

…

```
1 alloca
2 getelementptr
3 load
4 icmp
5 br
6 bpf_load_half
7 icmp
8 br
9 bpf_load_byte
10 icmp
11 br
12 bpf_load_half
13 trunc
14 store
15 bpf_pseudo
16 bpf_map_lookup
17 hash(2, 1)
18 icmp
19 br
20 load
21 icmp
22 br
```

Per Platform

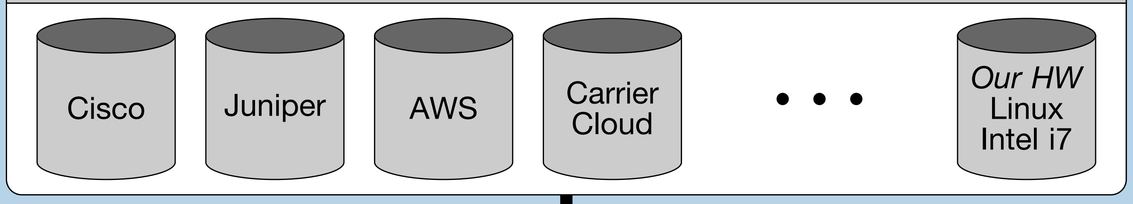
4 Calibrate the Platform
- Each platform has different performance characteristics
- Perform a series of microbenchmarks
- *Once* per platform
- Is done *offline* and can be *shared*
- **No testbed required!**



Offline Calibration

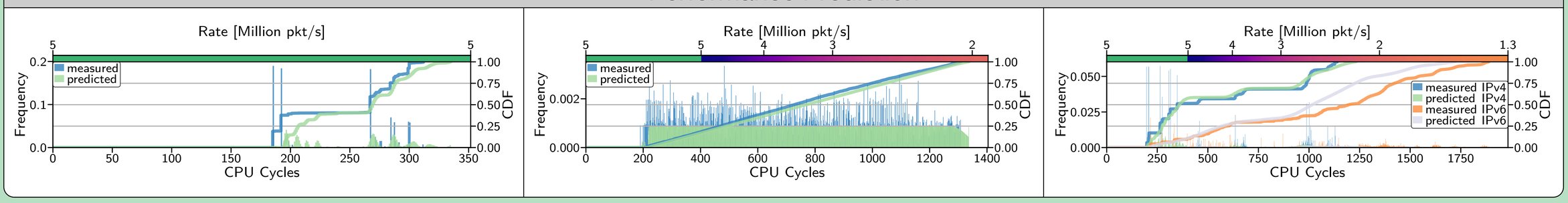
Measure instruction costs on target platform, e.g., CPU cycles needed for an `add` instruction on our hardware (i7-870)

Per Platform Instruction Cost Databases



Performance Prediction

Performance Prediction



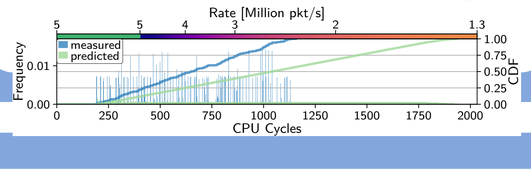
5 Predict the performance of a Network Function
- Convolute the costs for each instruction in a chain
- Combine path predictions by traffic pattern (e.g., from `.pcap` traces)
- Predict average case or worst case

6 Improve the performance of a Network Function
- Create test cases to assist the NF engineer with…
- … increasing packet rate
- … reducing latency
- … hardening against attacks

Opportunities

CPU Effects

- Thoroughly analyze CPU effects
  - Caching probably has little impact
  - Branch prediction on the other hand…



System Interactions

- Which packet rate is actually achievable for a certain NF?
- How do multiple NFs interact with one another?
  - When used serially on the same packets, or …
  - … independently on different flows
- How does the persistent state influence behavior?
  - Can invariants be proven?
  - … in the presence of race conditions?

Symbolic Analysis

- Symbolic Execution is prone to path explosion, even when bounded
  - How far can we push it in this specific area?
- SMT solvers, while impressive, still have to solve NP-hard problems
  - How well can cryptographic functions be emulated?
- How can interactions between multiple NFs be modeled?
  - On the same system, or…
  - … over the network

Funding

European Research Council
Established by the European Commission

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 647295 (SYMBIOSYS).

SYMBIOSYS

Felix Rath, Johannes Krude, Jan Rüth, Daniel Schemmel, Oliver Hohlfeld, Jó Á. Bitsch, Klaus Wehrle
https://www.comsys.rwth-aachen.de/research/projects/symbiosys/

COMSYS | RWTH AACHEN UNIVERSITY