NFOpt: Eliminating Redundant Logic in NF Programs using Operation-Time Configurations

Bangwen Deng, Wenfei Wu
IIIS, Tsinghua University

Introduction

Background

Network functions (NFs) are critical components in the network data plane. Their efficiency is important to the whole network's end-to-end performance. And a lot of works recognize this critical efficiency issue and propose the corresponding optimization, such as accelerating the NF execution, parallelizing NF (chains), and consolidating NFs. A recent trend of DevOps inspires us to propose an orthogonal approach — using the operation-time configurations to optimize NF programs.

Contributions:

- Show the existence of redundant logic in NFs in the scenario of DevOps.
- Design compiler-based solutions called NFOpt to eliminate the redundant logic.
- Evaluation NFOpt on commodity NFs and platform NFs completely.

NFOpt Overview

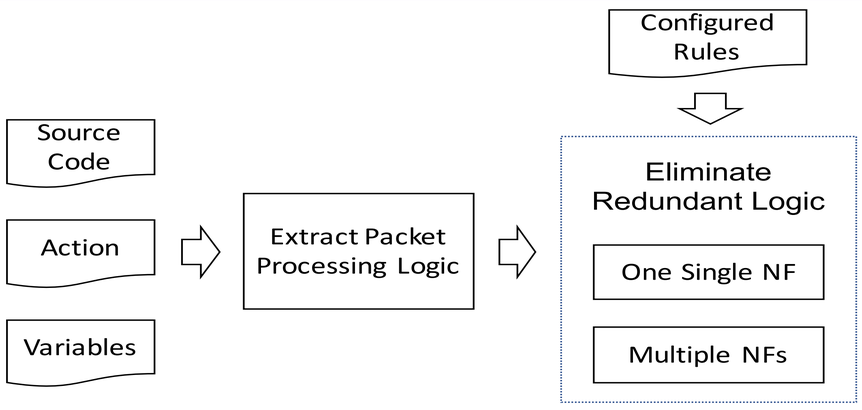


Figure 1: NFOpt Overview

Taking as input:

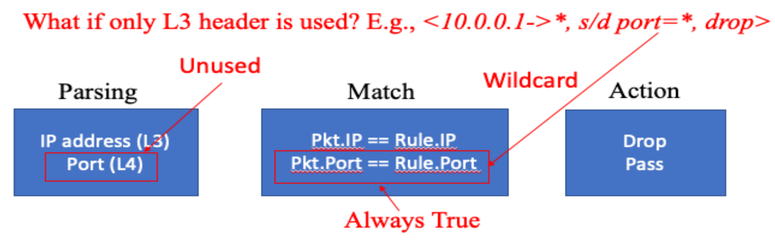
- NF Source Code,
- NF Configuration
- Identified Variables and Actions

Program Analyzing Techniques:

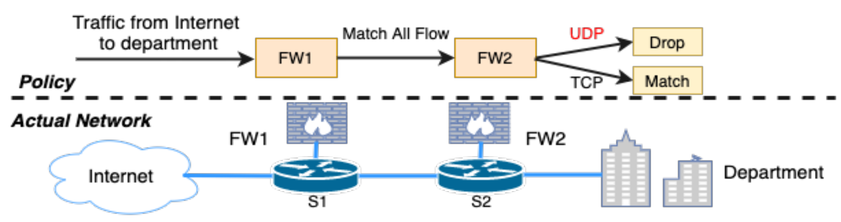
- Constant Propagation & Expression Fold
- Dead Code Elimination
- Common Sub-expression Elimination
- Copy Propagation
- Symbolic Execution
- Program Slicing

Eliminating three kinds of NF redundant logic:

- Unused Logic.



- Duplicate Logic.
- Overwritten Logic.



Evaluation

Unused Logic Redundancy

- Unused Layer Redundancy

The throughput of the two IDSes and the firewall increase significantly. (e.g., 15% for Snort, 21% for OpenNetVM-Firewall, 15%-10× and 40% to 3× for Suricata in single-thread mode and multi-thread mode, respectively)

- Unused Protocol Redundancy

When the proportion of UDP packets increases to 50%, removing the redundancy can achieve 40% and 2.5× performance gain for Snort and Suricata, respectively. In OpenNetVM-Firewall, configurations are embedded in code, the compiler would apply some optimizations before we apply NFOpt, the performance gain is moderated.

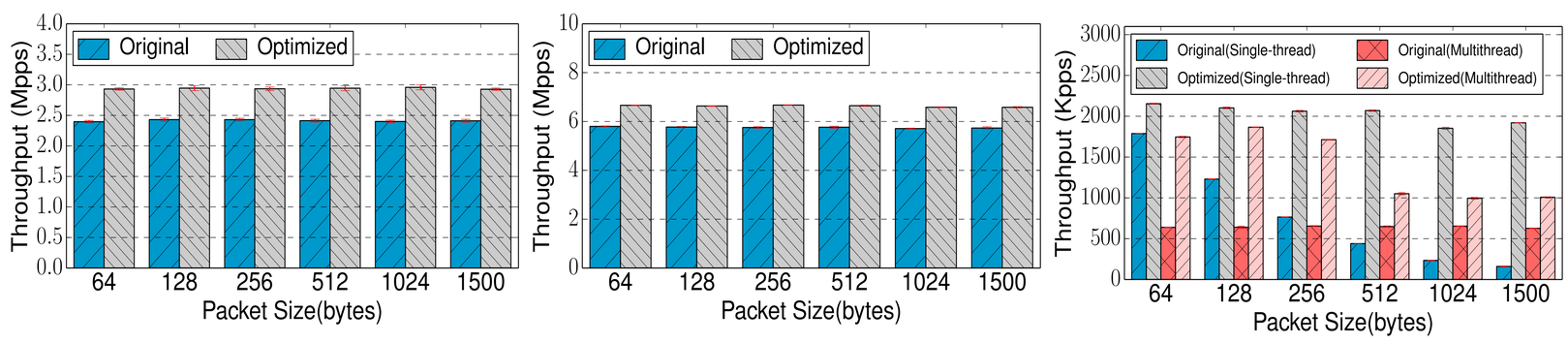


Figure 2: Eliminate Unused Layer Logic Performance Gain of OpenNetVM-FW(left), Snort (middle), Suricata (right)

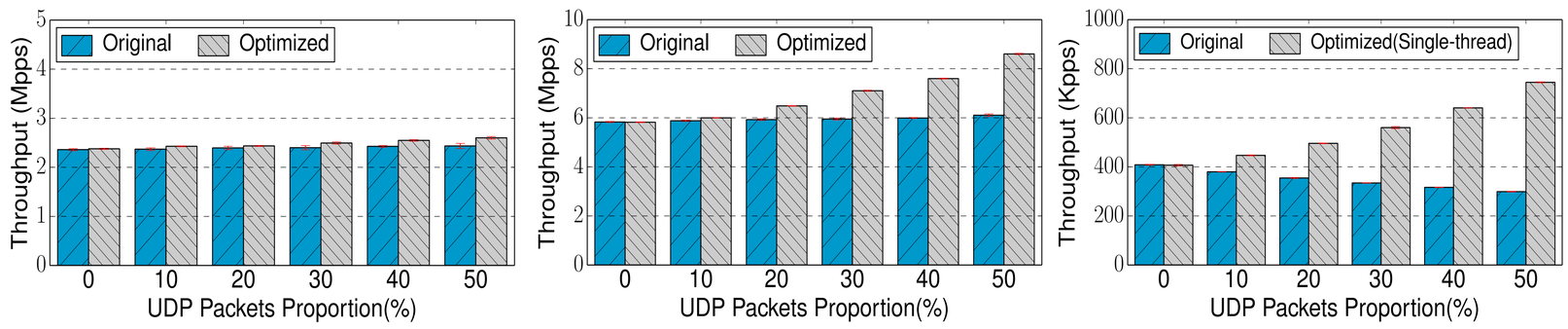


Figure 3: Eliminate Unused Protocol Logic Performance Gain of OpenNetVM-FW(left), Snort (middle), Suricata (right)

Duplicated Logic Redundancy

The Duplicated Logic elimination can help improve throughout by more than 25% for monitor and Snort IDS chain, and improve throughout by nearly 55% for the two OpenNetVM-Firewall instances.

Overwritten Logic Redundancy

After the elimination of the Overwritten Logic elimination Optimized program can achieve about 7% performance gain in both case when UDP proportion reaches to 50%.

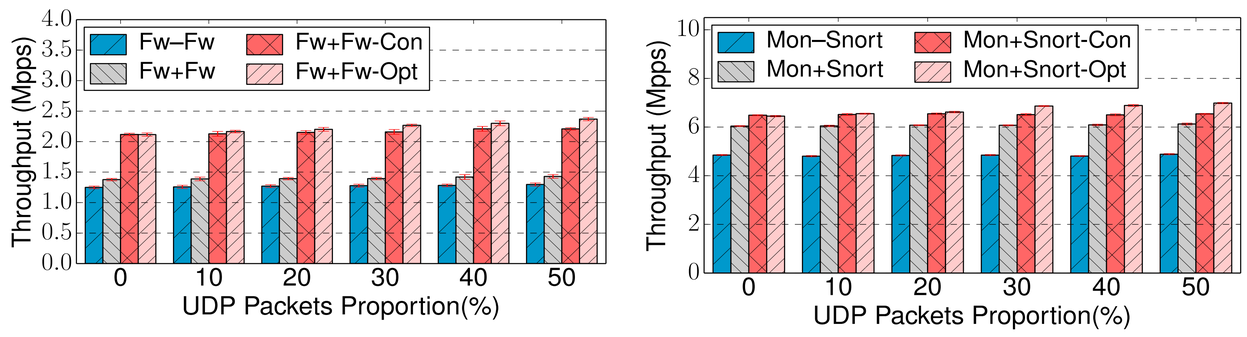


Figure 4: Performance Gain of Eliminating Duplicate Logic and Overwritten Logic

References

[1] https://llvm.org/.
[2] https://www.snort.org/.
[3] J. Khalid, A. Gember-Jacobson, R. Michael, A. Abhashkumar, and A. Akella. Paving the way for NFV: Simplifying middlebox modifi- cations using statealyzr. In 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), pages 239–253, Santa Clara, CA, 2016. USENIX Association.
[4] A. Saadaoui, H. Benmoussa, A. Bouhoula, and A. Kalam. Automatic classification and detection of snort configuration anomalies - a formal approach. pages 27–39, 01 2015.
[5] W. Wu, Y. Zhang, and S. Banerjee. Automatic synthesis of nf models by program analysis. In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, HotNets '16, pages 29–35, New York, NY, USA, 2016. ACM.