R2-D2: Filter Rule set Decomposition and Distribution in Software Defined Networks
Ahmad Abboud, Rémi Garcia, Abdelkader Lahmadi, Michaël Rusinowitch, Adel Bouhoula

To cite this version:
Ahmad Abboud, Rémi Garcia, Abdelkader Lahmadi, Michaël Rusinowitch, Adel Bouhoula. R2-D2: Filter Rule set Decomposition and Distribution in Software Defined Networks. CNSM 2020 - 16th International Conference on Network and Service Management, Nov 2020, Izmir/Virtual, Turkey. hal-03036292

HAL Id: hal-03036292
https://inria.hal.science/hal-03036292
Submitted on 2 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
R2-D2: Filter Rule set Decomposition and Distribution in Software Defined Networks

Ahmad Abboud 1,2, Rémi Garcia 3,1, Abdelkader Lahmadi 2, Michaël Rusinowitch 3, Adel Bouhouda 4
1Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
2NUMERYX TECHNOLOGIES, France
3IPB Enseirb-Matmeca, France
4College of Graduate Studies, Arabian Gulf University, Kingdom of Bahrain

Background and Motivation

Context
- Large number of filtering rules due to the increasing number of hosts and applications.
- Increase in number of attacks that affect entries in access-control lists (ACL).
- Expensive and power-hungry ternary content-addressable memory (TCAM).

Research question

How to decompose and distribute filtering rules on a set of limited size switch tables?

Overview of Longest prefix matching (LPM) representation

<table>
<thead>
<tr>
<th>Rule</th>
<th>Address field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 * *</td>
<td>A1</td>
</tr>
<tr>
<td>2</td>
<td>0 0 0 *</td>
<td>A2</td>
</tr>
</tbody>
</table>

Table: Example of a rule set in a switch table.

If a switch receives a packet with 0001 as address

Prioritized list strategy : Rule 1 is first, $A_1$ is applied.

LPM strategy : Rule 2 is most specific, $A_2$ is applied.

Rule Representation

- Single field filtering.
- Sufficient for blacklists.
- Rules represented in a binary tree.
- One rule at most on each tree node.

- Input a set of rules and switches.
- Search the binary tree in order to find the best candidates.
- All rules present in the chosen node and all nodes below it will be added to the switch.
- Minimize the number of generated rules by merging old forward rules with the ones from the best candidates.
- No rule duplication.

Decomposition Algorithm

Forward Rule Generation

- Forward rules avoid processing packets filtered by previous switches multiple times.
- Rules with deny action does not require a forward rule to be generated.

Decomposition over a graph

- Series-parallels graphs.
- Build the binary tree from a parallel and series composition.
- Simplify the binary tree using $S$-components.
- Packets with different sources will be processed by the same rule table of an intersection switch.

Evaluation

- 12 sets of data generated using ClassBench.
- Percentage of forward action field between 0 and 100%.
- Rules with same action type have zero overhead.
- Around 15% overhead on 8 switches path length.

Acknowledgement

This work is supported by a CIFRE convention between the ANRT (National Association of Research and Technology) and the company NUMERYX Technologies.