Practical Network-wide Packet Behavior Identification by AP Classifier

Huazhe Wang, Chen Qian, Yu Ye
University of Kentucky
Hongkun Yang and Simon S. Lam
University of Texas
NETWORK-WIDE PACKET BEHAVIOR IDENTIFICATION

- An control plane application identifying forwarding behaviors of packets in a flow:
  - Forwarding paths.
  - Where packets stop or are dropped.
  - Which boxes they traverse: routers, switches, and functional middle boxes.

![Diagram showing packet header processing through control and data plane](image-url)
WHY IT IS NECESSARY FOR SDN MANAGEMENT

- Verification of flow properties
  - Forwarding correctness, Policy enforcement, VLAN isolation
- Traffic Engineering
- Localization of network faults
- Security attack detection
- Label switching

- Can be used by many SDN management tools as a function
  - Reachability Analysis [NSDI12, ICNP13], Verification [NSDI13]
  - SPHINX [NDSS15]
Requirements…

- Practical network-wide packet behavior identification should have:
  • High query throughput
  • Small memory cost
  • Real-time updates

- None of the existing solutions can meet all of the requirements stated above.

- Our work: AP Classifier
  - AP stand for atomic predicates.

  Real-time Verification of Network Properties using Atomic Predicates. ICNP’ 13
AP CLASSIFIER NETWORK MODEL

- Forwarding and ACL (Access control list) rules are packet filters
  - represented by *predicates*.
  - guard input and output ports of boxes

- Predicate $P$ specifies the set of packets that can pass the filter
  - for which $P$ evaluates to *true*

- If a packet can travel through a sequence of packet filter, it is evaluated to true by the *conjunction* of predicates corresponding to the packet filters.
A SIMPLE NETWORK WITH THREE PREDICATES

- Given a set of predicates, we can compute a set of atomic predicates.

Each AP is the conjunction of $p$ or $\neg p$ for all predicates. Each predicate is the disconjunction of one or more APs.

Properties:

- $p_2 = a_3 \lor a_4$
- $a_4 = \neg p_1 \land p_2 \land p_3$
A SIMPLE NETWORK WITH THREE PREDICATES

- Each atomic predicate specifies a set of packets that have the same forwarding behaviors.

\[ a_4 = \neg p_1 \land p_2 \land p_3 \]

Forwarding path of a packet specified by \( a_4 \) at ingress box \( b_1 \)
AP Classifier Framework

- **AP Tree**
  - classifies the packet to the atomic predicate that evaluates to true.
  - Binary tree
  - Construction, search, and update.

- **Computing packets behaviors**
  - identifies all forwarding paths based on atomic predicate, network information, and ingress box of the packet.
AP TREE CONSTRUCTION

(Shaded cycles indicate leaf labels that are false)
AP TREE CONSTRUCTION

- Classify a packet to an atomic predicate
  - AP Classifier searches the AP Tree by evaluating the packet until a leaf is found.
  - At each node, the packet is evaluated by checking the predicate.
  - In the worst case: $2^k$ atomic predicates, $k$ predicates are checked.

- An observation
  - Number of atomic predicates are small for real networks.
  - Many leaves specify an empty set of packets.
AP TREE CONSTRUCTION

- Predicates in a different order, average depth of leaves may change.

\[ \frac{1+3+3+3+3}{5}=2.6 \]

\[ \frac{2+2+2+3+3}{5}=2.4 \]
CORRELATION OF AVERAGE DEPTH AND THRUROUGHPUT

Internet2: 100 random generated AP Tree

Smaller leaf depth, faster query speed
AP TREE OPTIMIZATION

- Optimal solution: NP-hard

- Two proposed heuristic algorithms
  - Quick-Ordering
  - Optimized AP Tree Construction (OAPT)
**AP Tree Update and Reconstruction**

- Real networks subject to dynamic changes
  - link failure/recovery, rule insertion/deletion
- Real-time update of an AP Tree
  - Add a predicate
  - Delete a predicate
- Parallel reconstruction of an AP Tree

---

**Diagram:**
- AP Tree querying
- Fast update
- AP Tree reconstruction

**Process Flow:**
- Query process: 1, 2, 3, 4, 5, 6, 7, 8
- Transmit new tree: 23
- Transmit new tree: 78

---

14
Evaluation

- Implementation on a general purpose desktop
- Real network data plane

Table 1: Statistics of the two real networks

<table>
<thead>
<tr>
<th></th>
<th>Stanford</th>
<th>Internet2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No. of rules</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarding</td>
<td>757170</td>
<td>126017</td>
</tr>
<tr>
<td>ACL</td>
<td>1584</td>
<td></td>
</tr>
<tr>
<td><strong>No. of predicates</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>507</td>
<td>161</td>
</tr>
<tr>
<td></td>
<td>71</td>
<td></td>
</tr>
<tr>
<td><strong>No. of atomic predicates</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>494</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>216</td>
</tr>
</tbody>
</table>
Overall construction time cost of AP Classifier

Construction time of AP Classifier is less than 0.4s.

Memory: 4.8MB and 2.2MB
The processing speed of AP Classifier is faster than existing tools by at least an order of magnitude.
200 updates/s. Reconstruction every 0.4s.
CONCLUSION

- AP Classifier: a tool to compute data plane behaviors of packets
- Using tree for classification to Atomic Predicates.
- Achieves >2M queries/s on a single core, using only a few MBs
Thank you!
Danke!

Questions?