Fast Configuration Change Impact Analysis for Network Overlay DCNs

Lizhao You, Hao Tang, Jiahua Zhang, Xiao Li
Huawei Technologies
youlizhao@huawei.com
Network Overlay DCNs

- ToR switches: VxLAN tunnel endpoints and distributed gateways
  - Layer 2: bridge-domain interface (BDIF)
  - Layer 3: virtual BDIF (VBDIF)
- Distributed protocol: BGP EVPN
- Policies: ACLs, micro-segmentation (MCS)

- To deploy/update services, users design the VPC network, and specify policies through the controller
- Delta configurations are generated, and pushed to the fabric
Problem: All-Pair Reachability Changes

- Re-computing full matrix is not scalable
- Incremental: find possibly changed pairs, and re-compute their reachability only
- Existing incremental approaches do not address the new challenges introduced by network overlay DCNs
Challenges #1: Model Expressiveness

- Feature 1: tunneling with overlay/underlay traffic
- Feature 2: packet rewrites for multiple tunnels
- Feature 3: MCS, group definitions are distributed
  - A: 1.1.1.0/24, 1.1.2.0/24 → Group ID: 10
  - B: 1.1.3.0/24 → Group ID: 10
  - C: 1.1.4.0/24 → Group ID: 20
- Policy on groups:
  - SRC ID 10, DST ID 20, SRC PORT 80, DENY

Existing BDD-based EC approaches (APV[ICNP’13, APKeep[NSDI’20]) and TBV-based EC approaches (VeriFlow[NSDI’13, DeltaNet[NSDI’17])
- Current EC computation does not support tunneling
- Extensive packet rewrites → Performance degradation
- The TBV model relies on IP rules instead of Group ID rules
  - Conversion → Excessive IP rules
  - Invalid: some SRCs may not reach DSTs
Challenges #2: Completeness of Indexing Methods

Indexing method (e.g., TenantGuard [NDSS'17])
- Compute a reachable path
- Associate the visited device with the reachable pair
  - B1 $\rightarrow$ {(A,B), (A,C)}
- Any changes on device B1
  - Re-compute (A,B) and (A,C)

Device-level association is coarse-grained and inefficient
- E.g., E only changes (A,C), no need to re-compute (A,B)
- Interface-level association may be preferable

Can not find new reachable pairs
- E.g., B2 adds a static route for A $\rightarrow$ D, and B has no association with (A,D) initially
Our Solutions

We follow the control-plane verification approach:
protocol simulation $\rightarrow$ reachability analysis
Port-Predicate Model

Symbolic Packet & Boolean Formula (Predicate)

<table>
<thead>
<tr>
<th>Outer IP</th>
<th>GroupID</th>
<th>VNI</th>
<th>Inner IP HDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>64bits</td>
<td>16bits</td>
<td>24bits</td>
<td>64bits</td>
</tr>
</tbody>
</table>

- i-th bit: 1 \( x_i \)
- i-th bit: 0 \( \bar{x}_i \)
- i-th bit: 0 or 1 \( x_i \cup \bar{x}_i = true \)

Each IP/Prefix \( f(x) = f(x_1, ..., x_{168}) \)

Header Space \( g(x) = g(f_1(x), ..., f_k(x)) \)

(union, intersection, difference)

All Boolean formulas are represented by BDD

Aggregation of allowed space of ACLs and FIB

\[ g_{ACL}^{BDIF1}(x) \cap g_{FIB}(x) \cap g_{NVE}^{ACL}(x) \]

Rewrite: \( erase \) and \( -set \)

Universal quantifier

if \( \forall x, x_{vni} == false \) then Overlay Packet

else Underlay Packet

Existence quantifier \( == erase \)

\( \exists x, x_{vni} \cap x_{vni}^{new} \)
All-Pair Reachability Matrix

Forwarding Graph

DFS for each starting endpoint

Endpoints

Indexing Table

<table>
<thead>
<tr>
<th>Type</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound</td>
<td>(L1,A)</td>
<td>(A,B)</td>
</tr>
<tr>
<td>Interface</td>
<td>(L2,NVE)</td>
<td>(A,B)</td>
</tr>
<tr>
<td>Outbound</td>
<td>(L1,B)</td>
<td>(A,B)</td>
</tr>
<tr>
<td>Interface</td>
<td>(L1,NVE)</td>
<td>(A,B)</td>
</tr>
</tbody>
</table>
Change-Impact Analysis

**Step 1: Changed Interface Detection**
- Old forwarding graph
- New forwarding graph
- Comparison

**Step 2: Waypoint Computation**
- Forward
- Backward
- Reachable Pairs

**Step 3: Comparison**
- Changed Interfaces
- Indexing
- Reachable Pairs
- DEL
- MOD
- ADD
Preliminary Evaluation

Implementation

- Based on Batfish*
  - Support Huawei CLI/YANG
  - BGP EVPN protocol
- JavaBDD

Synthesized Datasets (similar to real networks)

- Base dataset (20Leafs ~ 100Leafs)
  - Inter-VPC traffic
  - Intra-VPC (inter-subnet) traffic
  - 1 Leaf has 20 endpoints (subnets)
- Different service update cases
  - Case A/B: ADD SUBNET/ADD VPC
  - Case C: ADD cross-VRF Static Routes
  - Case D: MOD MCS

Metric: new graph modeling, detection, waypoint computation and comparison time

* https://github.com/batfish/batfish

Less than 25s for 2k EPs (4 mill. Pairs)

Case C: 6X

Case D: increase is mostly due to waypoint. To be optimized.

Cases A-C: increase is due to modeling, detection, comparison.

Case D: increase is mostly due to waypoint. To be optimized.

2X
Conclusion

- We are the first to design and demonstrate an incremental configuration verifier for network overlay DCNs.

- We design a fast incremental verification algorithm that leverages fine-grained indexing and waypoint computation methods to find all-pair reachability changes.

- In the future, we will further explore new forwarding features: policy-based routing, firewall zone policies, NAT policies, etc.
Thank you

www.huawei.com