APF: Fast Network All-Pair Reachability Calculation

Jinghan Zhou† Danyang Li† Xiaohe Hu† Yan Sun* Shui Cao* Wei Xu* Jun Li† ◇
† Tsinghua University * Huawei Technologies ◇ Beijing National Research Center for Information Science and Technology

ABSTRACT

Network verification, which is mainly about checking network states against network invariants or operator beliefs, has been a popular research thesis recently. Fast calculation of all-pair reachability of the network beforehand for further query or overall analysis can be a huge assistance to network verification. In this paper, we propose a new fast all-pair reachability calculation algorithm Atomic Predicates Flooding(APF). Experiments on real-life datasets show that the new algorithm based on network atomization is about three to four orders of magnitude faster than existing algorithms without network atomization. On most kinds of datasets, APF is even 2 to 5 times faster than the Warshall based all pair reachability algorithm with atomization. We believe that our method is essential for developing more practical and more efficient network and verification tools.

1 INTRODUCTION

Computer network is getting more and more complex these days. Often, the whole picture of the network exceeds what an operator can grasp. Therefore, some automatic network verification tools which take the network configuration files and network topology as input, calculate and report the information required by the operator to troubleshoot the network are eagerly needed.

Essentially, the basis of network verification is reachability calculation. Since the intents or invariants operators are interested in mainly include reachability query, blackhole or loop detection, slice confirmation etc., all of them can be taken down to calculate the reachable packet header set between certain pairs of network nodes. On the bootstrap stage of network verification, fast calculation of the reachable packet header set between all pairs of nodes (referred to as all-pair reachability in further discussion) can give related verification work a huge boost.

Simply put, to calculate the all-pair reachability of a network is to find out: for each pair of nodes in the network, what is the full set of packet header that can travel from one node to another? Basically, it can be done by traversing the network from each node, injecting all-wildcard packets (packet with all possible packet headers) at that node, looking up the routing table along the path to reduce the remaining packet header set, and combining all packet headers out of all possible paths.

Most of the rules in a network (except some non-deterministic rules and packet modification rules) can be represented in this form: $r = (match, action)$. Taken into consideration the priority of each rule on a network device, the ruleset of a network node can be represented as R, $R = \{(r_j, rp_j) \mid j = 1, 2, ..., |R|\}$, rp_j is the priority of rule r_j, and since the $match$ of different rules may overlap, the $match$ space of rules with higher priority will shadow the $match$ space of rules with lower priority.

To provide fast reachability calculation, network verification tools use different kinds of data structures to represent rules. Works like [5] and [6] use Binary Decision Diagram (BDD), [2] uses disjoint difference Normal Form (ddNF), [7] goes further with BDD: they use BDD to compute atomic packet header set of a rule set. Atomic means that this set of packet header will be treated identically in this network, and AP-Verifier calls this equivalent set of packet header **AP** of the network rule set. Here in APF algorithm, we adopt this atomization method to reduce the complexity of rules and eliminate the overlap between rules.

In this paper, we propose a fast algorithm, Atomic Predicates Flooding(APF), derived from the network atomization method addressed by AP-Verifier[7], for calculating the all-pair reachability of networks. Experiments carried out on real-life datasets with different sizes show that the proposed algorithm is around 10 times faster than the normal flooding-based algorithms with network atomization, 2-5 times faster than the Warshall[3] algorithm with network atomization on most kinds of datasets, and more than 1000 times faster than non-atomization methods.

2 ALGORITHM

Some of the existing work such as HSA[4], AP-Verifier[7], and ATPG[8] can calculate all-pair reachability with a $O(n!)$

ANCS '18, July 23–24, 2018, Ithaca, NY, USA

ACM ISBN 978-1-4503-5902-3/18/07.
https://doi.org/10.1145/3230718.3232112

Table 1: Time used to calculate all-pair reachability of different networks

	GridNet	Pacific-Wave	Karen	Myren	Palmetto	Garr	ASNET-AM	Viatel	US Carrier	Cogent
#Router	9	18	25	37	45	55	65	88	158	197
#Link	20	27	30	40	70	72	79	92	189	245
Avg Path Len	5.17	3.89	4.46	3.34	7.65	4.49	4.61	40.22	11.09	10.82
#AP	168	242	438	411	875	563	683	1054	2418	2970
Flooding(s)	0.0044742	0.0055724	0.017149	0.020283	0.39961	0.11353	0.16418	9.2644	9.0112	15.058
Warshall(s)	0.005538	0.001015	0.003154	0.004064	0.02905	0.012734	0.022401	0.65428	0.61620	1.1065
APF(s)	0.000999	0.0009912	0.002591	0.002126	0.01835	0.005744	0.008095	0.5208	0.1542	0.2057

worst-case complexity by flooding all-wildcard packets in network. However, by adopting the atomization method and focus on each AP's reachability, our algorithm gets a much better complexity. In APF algorithm, we calculate the all-pair reachability by calculating the all-pair reachability of the network only with a single AP and summing the results up. Specifically, our algorithm floods the network with each AP represented packet header set from every source nodes to every destination nodes to get a submap of the reachability for each AP. Then we merge those submaps together to obtain the reachability of the whole network.

After rule atomization, for each AP, each node has only one port that may contain it (the atomization process combine all rules with the same *action* together), then calculation of single source node reachability of an AP can be fulfilled by traversing the whole network only once, and that is $O(|V|)$ complexity. To calculate all pair reachability, calculate the single source node reachability from each node, and cost $O(|V|^2)$ complexity. We can treat each propagation from one node to another(essentially checking the existence of the AP in nodes' integer set) as $O(1)$ time complexity. Based on the inference above, the total time complexity to calculate all-pair reachability using this algorithm is $O(N \cdot |V|^2)$ (N is the number of AP).

3 EVALUATION

We compared our algorithm with ATPG, AP-Verifier and the baseline flooding algorithm with network atomization by using the classical dataset in network verification, Stanford Backbone Network. ATPG takes 56056.8ms to calculate it and AP-Verifier takes 218.4ms, while the baseline flooding algorithm takes 29.4ms and our algorithm takes only 8.29ms.

Then, we generate some datasets from the network topologies from the Internet Topology Zoo[1], and use these data sets to evaluate the performance of APF and compare it with the basic flooding algorithm and Warshall algorithm(the classic algorithm to calculate transitive closure[3]) with network atomization. Details about the datasets and experiment result can be found in Table1. To quantitatively demonstrate the complexity of the datasets, we introduce two key factor: Average Path Length and the number of AP. The average path length factor indicates the round for ruleset looking up per packet propagation, and the AP number indicates the number of equivalent packet header classes in this network.

Experiment results demonstrated that our algorithm gives a huge boost on calculating reachability.APF is 5 to at most 20 times faster than the baseline flooding method, and faster than Warshall algorithm on most of the datasets, especially datasets with higher AP number or average path length.

4 CONCLUSION

This paper develops a fast all-pair reachability calculate algorithm based on network atomization. Experiments on real-life datasets show that this algorithm is more than 1000 times faster than existing algorithms without network atomization, 5 to 10 times faster than existing algorithms with atomization. With this proposed algorithm, network operators can get the all-pair reachability relationship of a middle-sized network in less than 1 second. We believe that this tool can provide huge assistance to network operators and become a key part of network analysis.

REFERENCES

[1] The Internet Topology Zoo. http://www.topology-zoo.org/dataset.html.

[2] Nikolaj Bjørner, Garvit Juniwal, Ratul Mahajan, Sanjit A. Seshia, and George Varghese. 2016. ddNF: An Efficient Data Structure for Header Spaces. In *Haifa Verification Conference*.

[3] Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest. 1990. The floyd-warshall algorithm. *Introduction to Algorithms* (1990), 558–565.

[4] Peyman Kazemian, George Varghese, and Nick McKeown. 2012. Header Space Analysis: Static Checking for Networks. In *NSDI*.

[5] Ahmed Khurshid, Wenxuan Zhou, Matthew Caesar, and Brighten Godfrey. 2012. VeriFlow: verifying network-wide invariants in real time. *Computer Communication Review* 42 (2012), 467–472.

[6] Nuno P. Lopes, Nikolaj Bjørner, Patrice Godefroid, Karthick Jayaraman, and George Varghese. 2015. Checking Beliefs in Dynamic Networks. In *NSDI*.

[7] Hongkun Yang and Simon S. Lam. 2013. Real-Time Verification of Network Properties Using Atomic Predicates. *2013 21st IEEE International Conference on Network Protocols (ICNP)* (2013), 1–11.

[8] Hongyi Zeng, Peyman Kazemian, George Varghese, and Nick McKeown. 2012. Automatic Test Packet Generation. *IEEE/ACM Transactions on Networking* 22 (2012), 554–566.