Journal of Computer Science and Software Development
Received: Nov 28, 2015, Accepted: Jan 08, 2016, Published: Jan 11, 2016
J ComputSciSoftwDev, Volume 1, Issue 1
http://crescopublications.org/pdf/jcssd/JCSSD-1-002.pdf
Article Number: JCSSD-1-002

Research Article Open Access

Tool to Detect Multipath Routing When Tracking Path of Packet

SirishYennam, GouthamReddyGayapu, ArunMerugu, CharanKaranam andTarikEltaieb*

Department of Computer science, University Of Bridgeport, USA

***Corresponding Author:** TarikEltaieb, Department of Computer science, University Of Bridgeport, USA,
Email: teltaeib@my.bridgeport.edu

Citation: SirishYennam, GouthamReddyGayapu, ArunMerugu, CharanKaranam and TarikEltaieb (2016) Tool to Detect Multipath Routing When Tracking Path of Packet. J ComputSciSoftwDev 1: 002.

Abstract

Trace route is widely used, from the diagnosis of network problems to the assemblage of internet maps. However, there are a few serious problems with this tool, in particular due to the presence of load balancing routers in the network[1]. We provide a new publicly-available traceroute, called Paris traceroute, which controls packet header contents to obtain a more precise picture of the actual routes that packets follow. This new tool allows us to find conclusive explanations for some of the anomalies, and to suggest possible causes for others.

Objective

Traceroute is a tool to find the route packets take from a source to destination in network. It is used in diagnosis of network problems. Network administrators employ load balancing to increase the utilization of available bandwidth [2]. Traditional traceroute does not function as intended when there are load balancers in the path. The tool fails to discover the true links and nodes and may report false links between nodes because a load-balancer can direct the probes used by traceroute along different paths.

We introduce a tool that can detect the multipath routing of IP packets from a source to destination. It is a modification of classical traceroute algorithm, taking into account load balancer routing. It detects multipath routing and lists all the routers in the pathway. The implementation was done on IP level datagrams by creating specific ICMP echo packets with varying TTL and varying the data along with it to vary the checksum, such that multiple data arriving at any router will have different load flow parameters. With this mechanism, upon having sufficient number of data packets we map the network with some small probability of error.

Introduction

Traceroute developed by Jacobson is used in most of the places to find a path between two nodes. It lists all the network devices in the path between source and destination. It is used by almost all network administrators in troubleshooting. But there are issues with traceroute when there are load balancers in the path.

Brice Augustine et al. introduced a new tool called Paris traceroute [2] which tackles the issues in traceroute. They identify multipath routing in the path between two points. They show anomalies that they categorized as loops, diamonds and cycles.

This paper is inspiration from Paris traceroute. It introduces and explains why traceroute fails when there is load balancers in the path[3]. It proposed a method to identify multiple paths between source and destination. The implementation was done on IP level datagrams by creating specific ICMP echo packets with varying TTL and varying the data along with it to vary the checksum, such that multiple data arriving at any router will have different load flow parameters.

Background

Trace Route

Classical traceroute algorithm was developed to display the route of packets across an IP network. It displays the IP address of all the network devices between source and destination that the packets encounter with. It uses range from the diagnosis of ne network problems to the assemblage of internet maps, to learn the routers that lie between the originator and target[4].

The following explains how the traceroute algorithm works. Every IP packet contains a field called Time to Live (TTL) value. TTL value indicates the remaining lifespan of the packet[5]. TTL value is measured in the number of router hops. The main function of this field is to prevent routing loops from consuming an infinite amount of network resources by setting a finite limit on the number of hops that a packet can be routed through. In the IP routing process whenever a router receives a new packet, the router will decrement the TTL field by 1. If TTL value reaches 0, then the packet is dropped and an ICMP TTL value exceeded message is send to the original sender. Thus the original sender will know that the packet is dropped. Traceroute exploits this inherent behavior of the IP routing process to map each router in the path. Traceroute first sends a packet with TTL field initialized to value 1. When the first router receives the packet, it decrements the TTL value which now becomes 0. So the router discards the packet and sends an ICMP TTL value exceeded message which contains the router's IP address [6]. Traceroute records this IP address. So Traceroute sends packets with incrementing TTL values until the packet reaches the final destination. So when the final destination is reached, Traceroute will have the record of all the routers that the packets have been forwarded through which means it has the entire path between sender and final destination (Figure 1).

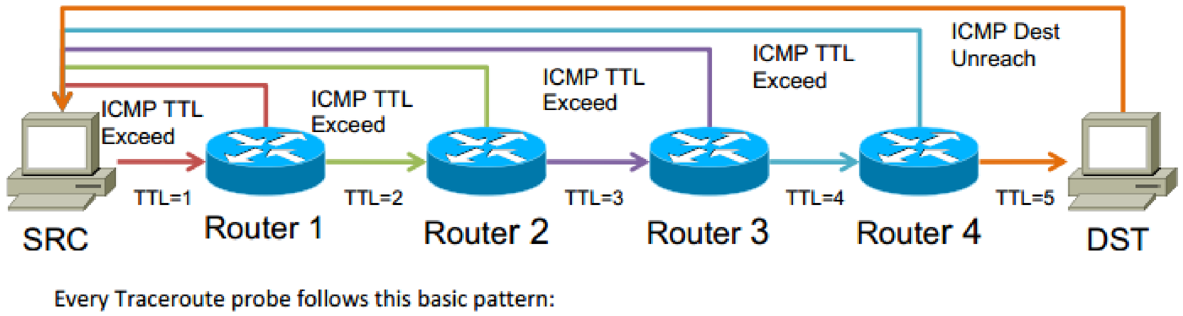


Figure 1: Basic Trace Route

Load Balancers and Traceroute

In a complex IP network, there is often a need to load balance the traffic across multiple paths to destination. There are two primary ways to accomplish this, a layer-2 based Link Aggregation protocol (LAG) and layer-3 based Equal Cost Multi-Path (ECMP) routing. Layer 2 based LAGs are invisible to traceroute, but layer-3 based ECMP is often detectable. When multiple paths are included in Traceroute results, it can significantly increase the difficulty of correctly interpreting the results and diagnosing any potential problems[7].This is where Traceroute measurements can be inaccurate and incomplete when the measured route traverses a load balancing router, or load balancer.

Load Balancers improves network utilization and reliability, so network managers prefer to have load balancing. They used routing protocols like OSPF and IS-IS to achieve that. Router divided the traffic based on three different policies – per packet, per flow or per destination.

Per-packet load balancers send the packets for same source-destination pair through different paths. They do so to reduce the congestion in the network. Per-destination load balancers divide the packets based on destination IP address. Per-flow load balancers send all the packets that belong to same flow through same path. The packets are grouped into flows based on the IP header information.

As cited in [8], the five fields from the IP header: Source address, Destination address, Source port, Destination port and Protocol forms a natural identifier. Routers can also use the following fields along with five mentioned above to assign a flow identifier: IP Type of Service, Code and Checksum fields[9]. A unique value in the probe header ensures a uniquely tagged response. Varying any field in the first four octets of the transport-layer header amounts to changing the flow identifier for each probe.

Classical traceroute cannot distinguish one route from another. It can discover the nodes in the path but cannot discover true nodes and true links between those nodes. So when load balancers direct the packets along different routes, traceroute forms incorrect links between routers that are in different paths. We will present a solution for this problem in the next section.

Implementation

In this section, a new traceroute like algorithm is introduced. In our implementation we need load balancers that follow per-flow load balancing. The flow identifier of a packet can be modified by modifying the transport layer header. A common flow identifier can be assigned to multiple packets by modifying the values in the transport layer header. When header field values are modified for different packets to generate same flow identifier, the checksum needs to be made constant so that the packets are not discarded [9]. We can never definitely confirm that for a route to a specific destination, we have multiple paths or not, but we can always state it with a definite probability if multiple path does not exist. Thus implementation of this traceroute algorithm increases the probability that the same path will be followed by all the packets seen in the network. As previously explained there can always be routers that route on a per packet basis and do not generally route on a per flow basis [6-8].

In this implementation, we are focusing on the sending ICMP packets with a constant checksum for a particular load flow, while gathering the packet TTL information from the sequence number. We can always choose a checksum and add 16 bit data to the ICMP message to satisfy the initial checksum chosen. This prevents the packets from giving a checksum error and reduces its probability of being dropped from the network[4]. It also gives us the freedom of varying various fields in the packet without changing its flow identifier.

The first step is to map the network by flooding the network with ICMP packets with randomly chosen checksums. This just ensures that the network has enough packets of varying flows at each TTL level such that all the routers are discovered at their respective levels. Here we differ from traceroute in the case that traceroute sends the same packet three times in a network therefore increasing the probability of finding the same router in a specific value of TTL, as the header remains the same, with the sequence number remaining the same. Therefore after the execution of the first loop we hope to find all the nodes present in the network for that specific path.

The next stage of the algorithm relies on mapping the interconnections between these nodes at each TTL level. Therefore we now proceed sequentially resending packets that are sent earlier to each of the routers, with the TTL value increased by 1. The packet then arrives at a router that is present at the next level, and we can map the flow of packets or the interconnection from one level to another.

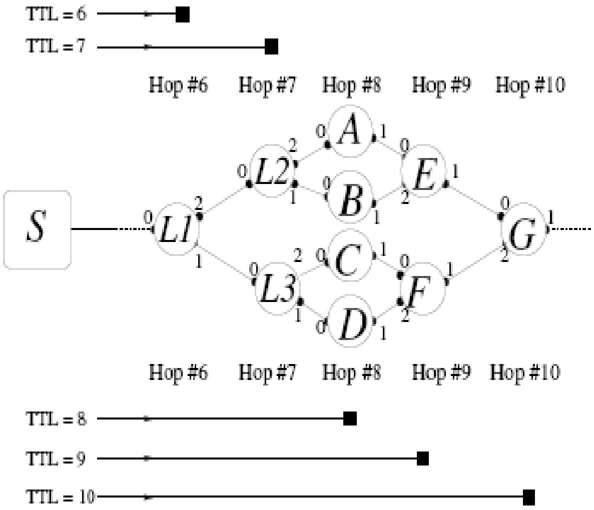
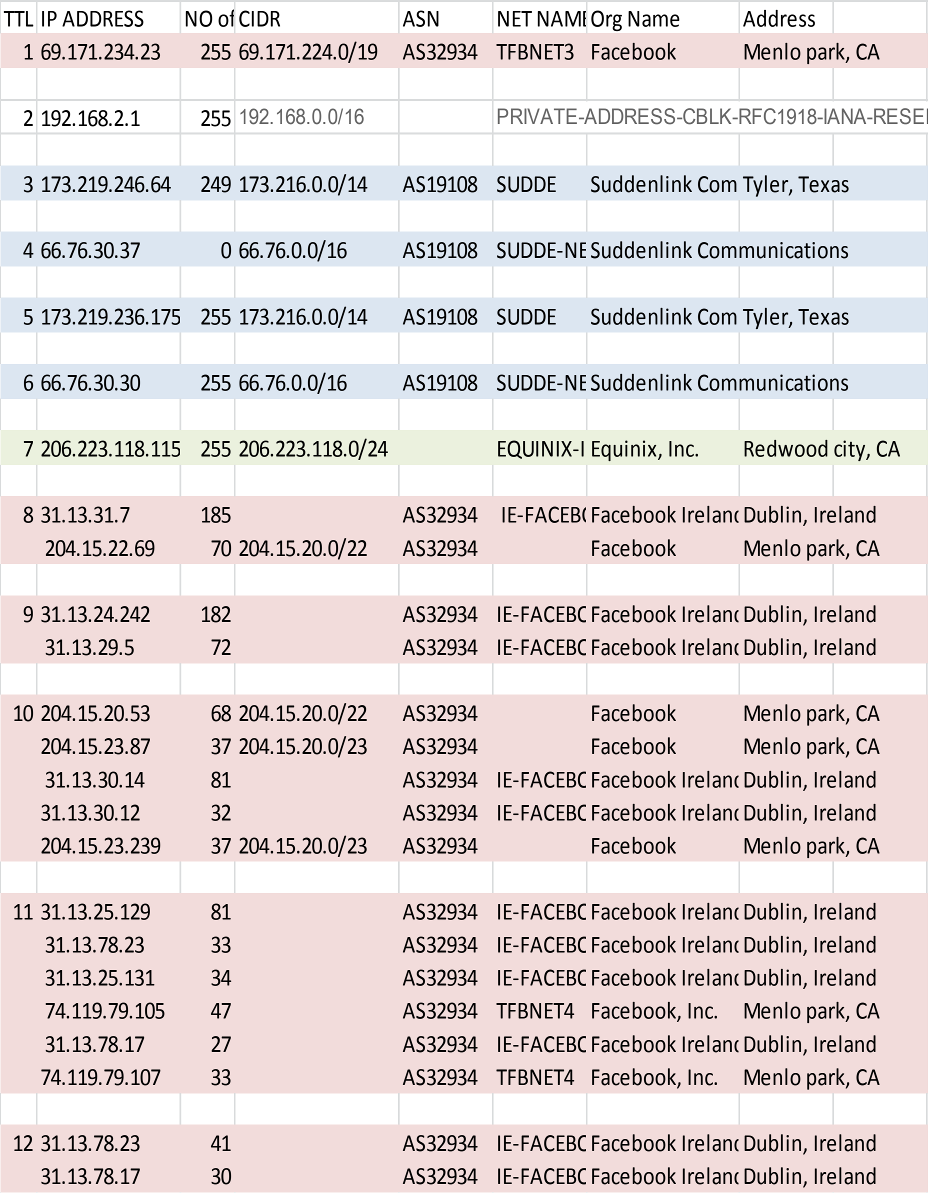


Figure 2: Multi path network with load balancers

Using Figure 2 as example network, the first step of our algorithm maps all the different nodes present in the network at various hop levels. So after first step, we know that L1 is hop 6;L2 & L3 are at hop 7 and so on. However, we do not know the interconnection between these nodes. In the second stage we send the same packets but with the TTL values increased by 1. Therefore all the packets present at hop 7 in the first step, would then send an ICMP message from hop 8. As we know the packet's parent node in hop7 to be either L2 or L3 we can then map the interconnections at each level between these nodes[7]. This will give us the full mapping of multiple routes in the network.

Results

Find the results as below…



TTL	IP ADDRESS	NO of	CIDR	ASN	NET NAMI	Org Name	Address	
1	69.171.234.23	255	69.171.224.0/19	AS32934	TFBNET3	Facebook	Menlo park, CA	
2	192.168.2.1	255	192.168.0.0/16		PRIVATE-ADDRESS-CBLK-RFC1918-IANA-RESEI			
3	173.219.246.64	249	173.216.0.0/14	AS19108	SUDDE	Suddenlink Com	Tyler, Texas	
4	66.76.30.37	0	66.76.0.0/16	AS19108	SUDDE-NE	Suddenlink Communications		
5	173.219.236.175	255	173.216.0.0/14	AS19108	SUDDE	Suddenlink Com	Tyler, Texas	
6	66.76.30.30	255	66.76.0.0/16	AS19108	SUDDE-NE	Suddenlink Communications		
7	206.223.118.115	255	206.223.118.0/24		EQUINIX-I	Equinix, Inc.	Redwood city, CA	
8	31.13.31.7	185		AS32934	IE-FACEB(	Facebook Irelan(	Dublin, Ireland	
	204.15.22.69	70	204.15.20.0/22	AS32934		Facebook	Menlo park, CA	
9	31.13.24.242	182		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
	31.13.29.5	72		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
10	204.15.20.53	68	204.15.20.0/22	AS32934		Facebook	Menlo park, CA	
	204.15.23.87	37	204.15.20.0/23	AS32934		Facebook	Menlo park, CA	
	31.13.30.14	81		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
	31.13.30.12	32		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
	204.15.23.239	37	204.15.20.0/23	AS32934		Facebook	Menlo park, CA	
11	31.13.25.129	81		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
	31.13.78.23	33		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
	31.13.25.131	34		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
	74.119.79.105	47		AS32934	TFBNET4	Facebook, Inc.	Menlo park, CA	
	31.13.78.17	27		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
	74.119.79.107	33		AS32934	TFBNET4	Facebook, Inc.	Menlo park, CA	
12	31.13.78.23	41		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	
	31.13.78.17	30		AS32934	IE-FACEBC	Facebook Irelan(	Dublin, Ireland	

These observations are from the website www.facebook.com. We see that it has a very good load balancing network as soon as it enters the Face book autonomous system.

Conclusion

This paper explains the traceroute behavior when there is load balancing in the network. It identifies the importance of packet header fields in load balancing. Then a new traceroute like algorithm is introduced which is an improvement over classical traceroute. The new traceroute can successfully identify the paths between source-destination when there are per-flow load balancers in the network. This algorithm fails when load balancers in the network using per-packet mode while forwarding packets. Future research can focus on improving this algorithm to handle the per-packet case.

References

[1] C. P. J. Adolphs and P. Berenbrink (2012) "Improved Bounds for Discrete Diffusive Load Balancing," in Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International, pp. 820-826.

[2] J. S. Marean, M. Losavio, and I. Imam (2008) "A Research Configuration for a Digital Network Forensic Lab," in Systematic Approaches to Digital Forensic Engineering, 2008. SADFE '08. Third International Workshop on, 2008, pp. 141-142.

[3] K. Hyoung Jun, S. Won Jay, and K. Sang Ha (2008) "Light-weighted Internet protocol version 6 for low-power wireless personal area networks," in Consumer Electronics, 2008. ISCE 2008. IEEE International Symposium on, 2008, pp. 1-4.

[4] L. Jie, W. Xingwei, L. Feng, and J. Jie (2011) "Efficient Traffic Aware Multipath Routing Algorithm in Cognitive Networks," in Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on, 2011, pp. 303-306.

[5] Y. Jiang, J. Ren, Y. Zhao, and B. Fang (2009) "Using Mixed and Hybrid TCP Probe Methods in Forward IP Paths Inference," in Parallel and Distributed Computing, Applications and Technologies, 2009 International Conference on, 2009, pp. 175-179.

[6] K. A. Magade and A. Patankar (2014) "Techniques for load balancing in Wireless LAN's," in Communications and Signal Processing (ICCSP), 2014 International Conference on, 2014, pp. 1831-1836.

[7] V. A. Kumar and D. Das (2012) "Feedback manipulation flooding attack: Feasibility evaluation and impact quantification on Stream Control Transmission Protocol," in Internet Technology And Secured Transactions, 2012 International Conference for, 2012, pp. 420-425.

[8] B. Huffaker, D. Plummer, D. Moore, and K. Claff (2002) "Topology discovery by active probing," in Applications and the Internet (SAINT) Workshops, 2002. Proceedings. 2002 Symposium on, 2002, pp. 90-96.

[9] V. Paxson (1999) "End-to-end Internet packet dynamics," Networking, IEEE/ACM Transactions on, vol. 7, pp. 277-292.