Efficient and Scalable Internet Mapping : Record Route Revisited

Vivek Ramachandran, Cisco Systems, Inc.
Sukumar Nandi, IIT-Guwahati
Amrit Kumar, UMAS
Indrajit Chakrobarty IIT-Kharagpur

Abstract

The IPv4 Record Route option was designed to accurately map the topology between any two nodes on the Internet. The IP protocol design allows only a maximum of nine IP addresses to be accommodated in the record route header field. As nine hops are insufficient to map the current extent of the Internet, the record route technique was replaced by Traceroute and Border Gateway Protocol (BGP) based techniques. These current techniques consume more bandwidth and host resources compared to record route. This paper revives the record route option by proposing various packet-marking schemes to be deployed on routers. The proposed technique also ensures a minimum of computational overhead for both routers and end hosts. It is faster, scalable and consumes lesser bandwidth compared to the Traceroute and BGP techniques.

1. Introduction

The Internet has grown from a relatively small size during the early days of its inception into a globally huge interconnected infrastructure. Visualization of the Internet and obtaining a router level map has been a major concern to Internet Service Providers (ISP). The Ipv4 Record Route option was designed to meet the above demands. The record route option contains nine spaces for subsequent routers to fill their IP addresses. The nine spaces soon proved to be insufficient as the Internet expanded. In today's context the average number of hops between two hosts on the Internet is around 18. This insufficiency to map more than nine hops led to the replacement of the record route technique by the Traceroute[7] mechanism or BGP[2] data collection to make such a router map .

The BGP based data collection technique uses the Border Gateway Protocol. BGP is an inter Autonomous System protocol, used by border routers of ISP's to exchange network reachability information with each other. The technique works by querying BGP enabled border routers for their tables and then reconstructing the network map based on the connectivity information. Note that this method has not gained much popularity due to the huge amount of data collection involved by querying the routers. An attempt has been made recently to map the Philippine Internet [4] using this scheme.

The Traceroute technique works by sending UDP packets to arbitrary ports on the destination host. The Time To Live (TTL) in the IP header of these UDP packets is set in an increasing order for each successive packet, starting from unity. Whenever a router receives an IP packet with the TTL set to one or zero, then it does not forward the packet to the next hop router. Instead, it sends an Internet Control Message Protocol (ICMP) error message (ICMP "time exceeded in transit" Code 0) to the source of the packets. This error message indicates that the packet exceeded its maximum transit time before reaching the desired destination. If the UDP service corresponding to the destination port number is not available, then the destination host will generate an ICMP "Port Unreachable" error message to the source of the UDP packet. It is by differentiating between these error messages i.e. Time Exceeded and Port Unreachable that the source of the UDP packets differentiates between routers and the final destination host. The source now reconstructs the exact path between itself and the destination based on the information in these error messages.

Among the above two methods Traceroute is more widely used. Traceroute's main drawback is the large number of packets required which on an average is double the number of routers in the path. The bandwidth requirements and errors due to packet loss are proportional to the number of hosts to be mapped.

We propose to solve the problem of the limited space constraint in the record route option by deploying various packet-marking schemes on routers. The marking schemes are simple to implement and are easily scalable. We use the same Ipv4 record route option along with the Reserved Bit field and the End Of List field in the Record route field in the IP header for marking the packets.

Section 2 proposes the various marking schemes. Comparative results are reported in Section 3. Concluding remarks are reported in Section 4.

2. Proposed Record Route Techniques

The proposed algorithm works by sending multiple record route packets to map the path. Each of these packets maps nine routers. We accomplish this by deploying various marking schemes to let the routers know when they should mark/ not mark the packets with their IP addresses. Using this technique we can effectively map more than nine routers between two given nodes. This paper proposes three "packet marking" schemes in increasing order of effectiveness in mapping the path. They are as listed below:

1. Odd – Even Approach
2. Timeout Method
3. Time To Mark (TTM) Method

2.1. Odd – Even Approach

In this approach the source maps the odd and even routers in separate record route packets. This is accomplished by marking the packets differently for odd and even routers. We use the Reserved Bit in the record route header to mark the packet. When a router receives a record route packet, it will act as per the *algorithm odd_even.*.

Algorithm odd - even

*/*Fields used: Reserve bit and record route fields of Ipv4 header. */*

/ When the router gets a record route packet it checks the record route and the Reserved bit fields. */*

Step 1: If record route option address field is full, forward packet without marking, otherwise follow Step 2.

Step 2: If (Reserve bit = 1) then
a) Set Reserve bit to 0.
b) Mark the address (of router) in the record route option address field.
c) Forward the packet.

Else, go to Step 3.

Step 3: a) Set Reserve bit to 1.
b) Forward packet.

To mark even (odd) routers from sender side, the sender of record route packet should set Reserve bit to 0 (1).

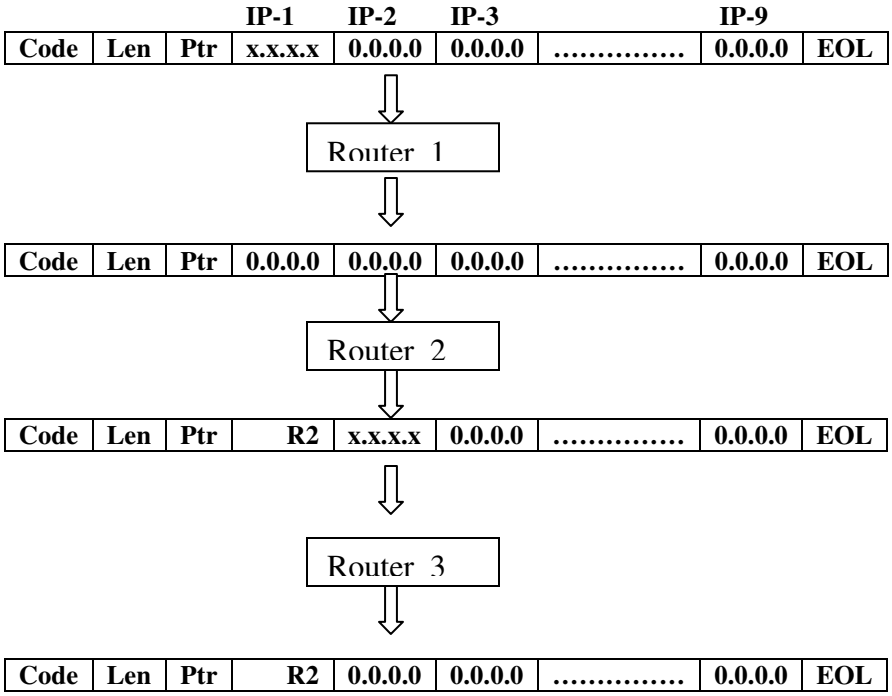

Figure 1. Even Router Marking

Simply put the routers flip the reserved bit set by the sender to allow successive routers to know if they should/should not mark their addresses on the packet. The sender of the Record route packets decides which hops it wants to map on the packet viz odd or even. The *odd - even* method can record up to 18 consecutive routers. We can extend the scheme to map another 18 consecutive routers from the other end point. This we term as *Reflection technique.*

In the *reflection technique*, whenever a host receives a packet with the record route option set, it generates an additional packet with the record route option enabled and sends it to the source of the received packet. As an example let a host receive two ICMP record route enabled, echo request packets (odd – even pair) with the Reserved bit set to 1(0). It replies back with echo replies containing the recorded route as in the previously received packets and additionally it will send two more record route packets(Odd – Even pair). These additional packets will be able to map 18 consecutive routers from the receiver side. Finally, the initiator of the record route packet will get four packets, two echo replies and two additional record route packets from the responder side. The sender of course would have to maintain state so that it does not send two more record route packets in reply to the receivers. These additional By this way we can map a maximum of 36 routers between two end points.

To avoid additional bit requirement such as Reserved bit, the record route address field can be used to indicate the routers when to mark the record route packet and when not to. We term this scheme as *alternate odd – even* scheme. This is done by modifying the *algorithm odd - even* as follows. Instead of setting the Reserve bit 0 (1), we can mark the next available address space in the record route packet with the address x.x.x.x (0.0.0.0). Here x.x.x.x is an invalid address which no router can have. This serves as our marker. The routers will mark the packet only if the next available address space using the Ptr field in the header contains the address 0.0.0.0. If it sees the address space to be some invalid address it will just change it to 0.0.0.0 and forward the packet to the next router without marking it's address. The invalid address can be taken to be 255.255.255.255, as no router is going to have this IP address. Rest of the steps will be same as *Algorithm odd – even*. Few steps of the *alternate Odd – Even* scheme is shown in figure 1 with the invalid address marked as x.x.x.x.

The *alternate odd – even* scheme requires more computation than *odd – even* scheme, but it does not requires any additional field bit for implementation. However, these two schemes can map a maximum of 36 routers (implemented along with *reflection method*).

2.2 Timeout Method

To map more than 36 routers, we propose another technique, termed as *Timeout Method*. In this method every router maintains a "Timeout Table". The parameters in this table are the source and destination IP address, a timeout period and an Identifier. The Identifier is actually a unique number generated for each record route packet sent by a host. The Identifier helps multiple processes send record route packets, without interfering with each other, as will be seen later. This identifier can be stored in the End Of List field, in the record route region. The Timeout period is the interval (say 255 seconds) after which the entry will be deleted. The algorithm is as follows :

Algorithm Timeout

/ When a packet is received, check the database for corresponding addresses and identifier. The (source, destination, identifier) combination will henceforth be referred to as an "entry". */*

Step 1: If entry exists or if no address space available for marking.

a) Forward the packet without marking.

Step 2: If no corresponding entry exists,

a) Add the source, destination IP and identifier to the database.
b) Set the timeout to 255 seconds.
c) Mark the address in the packet.
d) Forward the packet

The Router decrements the timeout values of all the entries in the table periodically. If the timeout value of an entry reaches 0, the entry is deleted.

We have chosen a value of 255 seconds which is sufficiently long to map a route by sending multiple record route packets. The *Timeout* scheme requires making sure that an identifier reuse is done only after a sufficiently long time. Also if another process also simultaneously sends another set of record route packets, they will be marked independently with their own timeouts as their Identifier would be different from the previous set of packets.

The sender of the Record Route packets would have to set a common identifier for all the packets used in mapping the route to the same destination. Packets should be sent at a small interval apart to make sure that the entries have been added on the routers for the previous packets. The first packet will cause the first nine routers in the path to add entries into their tables and also to mark their addresses on the packet. When the second packet arrives on the first nine routers it will not be marked by them as entries for the same source and destination Ip and identifier exists, instead it will be marked by the next nine routers in line. This will continue till all the hops have been successfully marked.

The *Timeout* scheme requires maintaining these timeout tables by the in-between routers. This additional book keeping might require extra resources and computing power. The acceptability of this scheme depends upon the storage and computational resources available to the routers. The advantage is of course that we can map any number of routers in between two nodes. Also if one decides to wait for the timeout to expire on all routers in the path before trying to map the same source-destination route then no changes on the host network stack would be required as the identifier value can always be set to 0 (default value of EOL field set by the host network stack).

2.3 Time To Mark (TTM) Method

The *Timeout* method has an overhead of additional computational and storage requirement. We propose another scheme termed as Time to Mark (TTM) Method to overcome this overhead. This scheme has no storage requirements and involves very less computation. It works by having the sender of record route packet store a value in the End Of List sub field. This value will be henceforth referred to as TTM. The TTM is an integer value and take a maximum value of 255. The intermediate router will check the TTM value and will act according to the algorithm presented below.

Algorithm TTM

/ Check the the Record route address space and the TTM value stored in the EOL field */*

Step 1: If TTM > 0,

- *a) Decrement the TTM by unity.*
- b) *Forward the packet to the next router without marking our address.*

Step 2: If TTM = 0,

If (record route marking space is available)

- *a) Mark the address in the record route packet.*
- b) *Forward the packet to the next router, letting the TTM to be zero.*

Else (no space available)

- *a) Forward the packet without modification to the next router.*

Depending upon which consecutive nine routers one wants to map the sender of the record route packet has to set the value of the TTM accordingly. The value of the TTM will be zero after the TTMth router processes the packet. The (TTM+1)th to (TTM+9)th routers will mark their addresses in the packet. Note that after the (TTM+9)th router has marked the packet, even though the value of TTM is still zero, rest of the routers in the path will not be able to add addresses as there is no space left. The End Of List (EOL) field is a part of the record route header itself so using it to store the TTM is advantageous. The computational overhead is small, as we only require to decrement the TTM value (i.e. EOL), like the TTL field in the Ip header. This method allows us to map the whole route comfortably with minimum computational overhead for the routers as well as end hosts. Also using this method we have the additional flexibility of starting the marking on the Record route packets from any intermediate router by setting the value of TTM accordingly.

3. Comparative study

All the three methods presented in this paper are distinctively different from each other. They are novel in approach and have their own advantages and disadvantages. To map a path having up to

36 hops the Odd – Even method is most optimal and requires a maximum of six packets to be sent across the network. For routes consisting of more than 36 hops, the Timeout and TTM methods are to be used. The choice between the Timeout and TTM methods depends on the availability of storage space and computational capacity of the routers. The Timeout method clearly requires more storage space and computational capacity as compared to TTM. The advantage of Timeout method is that the routers do not have to modify any extra fields in the packet. The TTM requires no storage space in the routers but routers have to change the TTM value in the packet, which is a minimal extra overhead.

3.1 Comparison with Traceroute

Once deployed on routers this "marking scheme" clearly wins over Traceroute in speed as well as bandwidth requirements.

In case of Traceroute, to map a path with *n* hops we need to send *n* UDP packets and then wait for *n* ICMP error messages. So *2n* packets are to be transmitted across the network. On the other hand if we use the record route packets, we require a maximum of 6 packets to map a route less than 36 hops with the Odd – Even method. For routes greater than 36 hops we use either the Timeout or TTM method. In these two methods we need to send only $\lceil n/9 \rceil$ ICMP echo request packets with record route enabled and then wait for their $\lceil n/9 \rceil$ respective replies, making a total of just $2*\lceil n/9 \rceil$ packets. We are able to decrement the number of packets to be sent by about $\lceil 8/9 \rceil * 100\% \approx 90\%$. Hence there is a drastic reduction of packets needed for mapping the route. This is achieved with a minor computational overhead added to the routers.

4. Conclusion

In this paper we have revived the Ipv4 Record Route option and are able to map networks of any size. Our schemes require minimal changes to the network functionalities of both routers as well as end hosts. Our schemes are scalable and easily deployable. Also all these schemes are independent of each other and can be deployed simultaneously. Our proposal also drastically reduces the bandwidth requirements compared to Traceroute and requires lesser packets to be sent across the network, making it less prone to packet losses and retransmissions. We would like to mention that we could easily extend such "marking schemes" for the IP Timestamp options as well, which has also become obsolete due to the space constraint problem.

5. References

1. TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994, ISBN 0-201-63346-9.

2. RFC 1771 - BGP – Border gateway protocol.

3. M.A. Paraz and W.S. Yu 2002. "Philippine Internet Content Performance Metrics". Sixth International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN2002), May 2002.

4. "Mapping the Philippine Internet using the BGP", Gino LV Ledesma . http://cng.ateneo.net/cng/wyu/classes/cs197/students/ph_internet_mapping.pdf

5. R. Govindan and H. Tangmunarunkit, "Heuristics for Internet map discovery," ftp://ftp.usc.edu/pub/csinfo/tech-reports/papers/99-717.ps.Z.

6. Cheswick, B. and Burch, H. Internet Mapping Project. Accessed 5 December 2003; available from http://research.lumeta.com/ches/map/index.html.

7. J. Rickard, "Mapping the Internet with traceroute," http://boardwatch.internet.com/mag/96/dec/bwm38.html

8. R. Albert, H. Jeong, and A. Barabasi, "Diameter of the World-Wide Web," in *Nature*, No. 401, 9 Sept. 1999, pp 130-131. Macmillan Publishers Ltd.

9. Macroscopic Internet Visualization and Measurement. Accessed 8 December 2003; available from http://www.caida.org/tools/visualization/mapnet/

10. Cooperative Association for Internet Data Analysis. Skitter. Accessed 8 December 2003; available from http://www.caida.org/tools/measurement/skitter/

11. Cooperative Association for Internet Data Analysis. Walrus -Graph Visualization Tool. Accessed 8 December 2003; available from http://www.caida.org/tools/visualization/walrus/

12. French Network Operators' Group. Graphical Autonomous System Path. Accessed 12 December 2003; available from http://mogwai.frnog.org/sysctl/gasp/

13. Matrix Maps Quarterly. Accessed 5 December 2003; available from http://www.mids.org/mmq/; Internet.

14. Alex C. Snoeren , "Hash Based IP Traceback" ACM Sigcomm 2001. http://www.acm.org/sigs/sigcomm/sigcomm2001/p1-snoeren.pdf.

15. RFC 1393 "Traceroute using an IP Option"

16. RFC 1812 "Requirements for IPv4 Routers"

17. RFC 791 "Internet Protocol IPv4 "