Multipath detection

NETMET Lab Exercises 6

Introduction

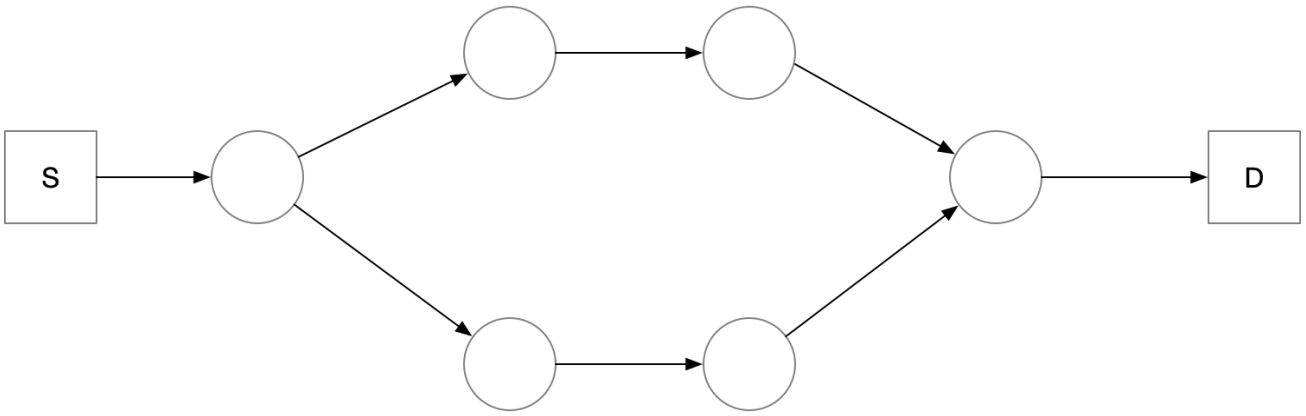
In lab 3 we described how Traceroute is working in order to discover the router's ingress IP address along a path between a source and a destination. But we quickly mentioned that Traceroute can make wrong discoveries if there is multipath along the way.

Today, we will discuss a modification of the algorithm of Traceroute in order to resolve the false links issue.
We will first try to understand the problem and find a way to avoid these errors, and then we will study an actual implementation of the corrected algorithm : ParisTraceroute.

Analyse the issue

For the next sections of the lab, unless it's explicitly mentioned, we refer to "classic Traceroute" as the Traceroute implemented in UNIX operating systems used with UDP probes.

Let's first remember the problem by analysing the behavior of classic Traceroute on this diamond topology :



In order to be able to match the probe sent with the corresponding response (in order to calculate the RTT for instance), classic Traceroute modifies the UDP *destination port* of each probe sent. That way, by keeping a state between the probe sent and the destination port used, classic Traceroute just has to look at the source port of the responses to be able to match the request and the response.

When a router have a choice of sending the packet to two different path, it can follow mainly two types of load balancing algorithms :

- *Per-packet load-balancing* : The router simply load-balances the packets with a round-robin in order to have evenly used interfaces. This algorithm is not used a lot.
- *Per-flow load-balancing* : The router attempts to keep the packets belonging to the same flow in the same path. A flow is often defined by the header fields :
  IP source address, *IP destination address*, *protocol*, *source port*, *destination port*

Questions :

- In a network perspective, what could be the issue of using *Per-packet load-balancing* ?
- Can you formulate precisely the problem of false links with *Per-flow load-balancing* that lead to errors in the classic Traceroute output ?

Formulate a fix

Now we understand the problem correctly, let's try to fix the issue.

Questions :

- What modification can we make to the algorithm of the classic Traceroute in order to avoid the false link issue ?
- What other problem occurs if we implement that solution ? How can we overcome this issue ?
- Traceroute is sending multiple probes per TTL. How can we know which request is matching the responses between these 3 probes with the same (flow, ttl) in our case ?

Note : ICMP error messages contain a data section that includes a copy of the entire IPv4 header, plus at least the first eight bytes of data from the IPv4 packet that caused the error message.

Analyze a fixed implementation

Take a look at this paper : https://paris-traceroute.net/images/imc2006.pdf

Questions :

- Describe the algorithm that is employed with UDP probes. Was it the same as you formulated in the previous exercise ?
- Describe the implementation for ICMP and TCP probes. How did they manage to keep the state between the requests and the responses ?

To go further

Think about the problem of formalization: If I do not know the underlying topology, how should I probe efficiently to be sure I can reveal all the paths ?

Litterature :

- https://paris-traceroute.net/images/infocom2009.pdf
- https://dl.acm.org/citation.cfm?id=3278536