Finding the best AS exit point with Round Trip Time probing

2G1703 Home assignment, Author: Magnus Larsson, [magnus at strilnetworks dot com], 2004-12-12

This document proposes a solution to find the best exit point for selected traffic in a multihomed AS. The idea is to let the eBGP border routers measure the Round Trip Time (RTT) for a selected set of routes received from its eBGP peer, and communicate the result to other BGP routers inside the AS with the LOCAL_PREFERENCE attribute.

Background and problem definition

Consider the scenario shown in figure 1. AS3 is a non transit AS multihomed to two different ISP:s. Traffic exiting AS3 should be routed through the gateway closest to the destination. The Cisco BGP4 implementation (and possibly other vendors too) looks at the length of the AS path to find the best exit point. From a user perspective, distance metrics such as RTT and bandwidth would be more relevant. A study by CAIDA [1] shows that selecting routes based on AS path length is useless if the path with lowest RTT is preferred. In fact they show that comparing AS path length is not better than a random choice when finding the lowest RTT path. How can a router choose the exit point out from an AS having the best RTT to the destination?

General Solution

Each boarder router marks a subset of the prefixes learned from its eBGP peer to be targets for RTT probing. The probe can be an ICMP Echo Request (Ping) packet sent to the prefix. An ICMP "Net Unreachable" message will be returned from the router hosting the more specific routes of the prefix address and the RTT can be measured. Next step is to translate the measured RTT to a LOCAL_PREFERENCE (LP) attribute value. An important property of the translation is that a low RTT should map to a high LP. The BGP4 specification (RFC1771) has defined a very generous LP space of 2^32 (4 bytes) and we can afford using a range if these exclusively for our translations. Translation is done by this formula:

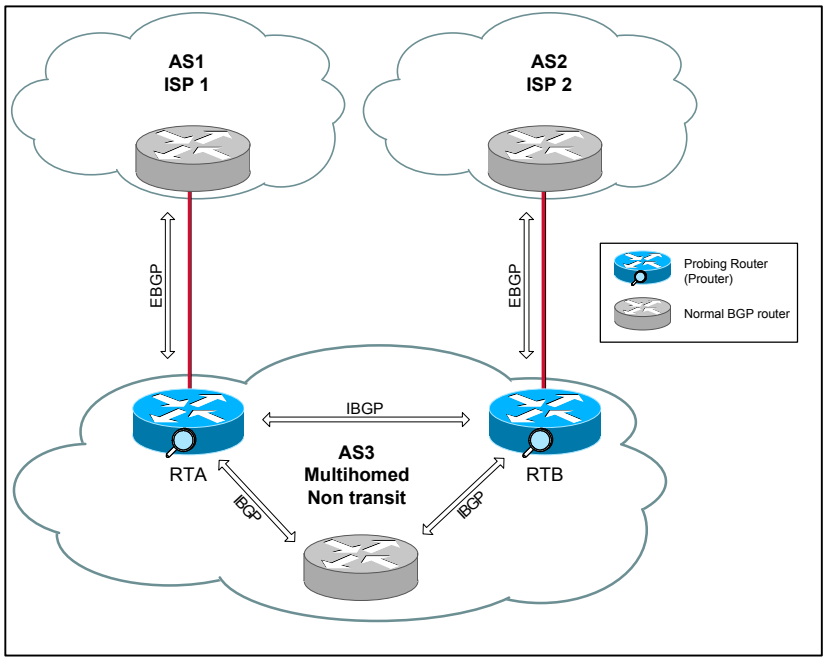
LP:= BASE + RANGESIZE / (RTT+1)

BASE is the start of the range. E.g. 1000

RANGESIZE is the size of the LP space reserved. E.g. 10.000

In this reminder of this document I will call this range starting with BASE and ending with (BASE+RANGESIZE) simply the RTT_LP range. The combined mechanism of probing selected prefixes and mapping measured RTT times to LP attribute values are called "Probing Routing" or shortly *Prouting* in the continuation of this document. Further, an eBGP router implementing Prouting is simply called a *Prouter*. That is, a Prouter is a router running both an eBGP and a Probing Routing process.

Figure 1



Example: RTA (being a Prouter) measures the RTT for a prefix to 125 milliseconds and will readvertise the route to its iBGP neighbours with a LP value of
1000 + 10000/(125+1) = 1079

For practical reasons the Prouting process should time-out if not answered within a certain amount if time and set RTT to infinity (or any other value high enough).

AS administrators should keep the LP translation range separate from the statically assigned LP values to clearly indicate, both to humans and other Prouters, which prefixes have been probed. Both the BASE and RANGE constants can be adapted to local requirements within the AS, as long as all Prouters share the same values. iBGP routers peering with Prouters choose the route with highest LP just as usual but with the difference that they now automatically choose the lowest RTT path (assuming RTT differences to reach each boarder router inside the AS can be neglected).

Choosing routes to be probed

To avoid having to probe all BGP entries, each Prouter will maintain a local list, the "ProbeList", of prefixes to be targets for probing. There could be several methods to add entries to the ProbeList, both manually and dynamically. Here are some examples:

- (Manual) **Access lists**. All routes matching an IP address, netmask range or AS path regexp.

- (Dynamic) **Top 100**. Most frequently utilized routes.

- (Dynamic) **Protocol based**. Routes used for RTT sensitive protocols (VoIP etc).

- (Dynamic) **IBGP_PEER_TRIGGERED**.

The last entry method is crucial to Prouting and used to let a Prouter distribute its ProbeList entries to other Prouters in the AS. If one Prouter finds a prefix interesting to probe, its neighbour Prouters should also probe this prefix so the LP(RTT) values can compared. A Prouter should insert a prefix into its local ProbeList if these conditions are both met:

(1) The prefix is received from another Prouter over iBGP with a LP inside the RTT_LP range. I.e. Another Prouter is probing this prefix.

(2) The prefix is received from this Prouter's own eBGP peer.

When any of these two conditions become false, the prefix should be removed from the ProbeList.

To ensure that the probe packet is sent out on the Prouter's external link and not routed internally to another exit point, it must be sent directly to the eBGP peer without consulting the local routing table. The eBGP peer won't send it back as we are not a transit AS.

Stability

Two factors have significant impact on the stability of Prouting: How often probes are sent and the impact of short lived RTT fluctuations. Both of these properties has to be carefully adjusted to provide good load balancing without causing oscillation of traffic between the exit points as well as unnecessary probe traffic on the Internet. One probe strategy might be to initially probe a prefix once it inserted into the ProbeList, resample every 100 hours, and take the mean of the last 8 probes to be the RTT value to use.

One aspect to consider is the impact of the link utilization of the exit points. Suppose a host in our AS is sending enough traffic to a single destination via RTA to saturate its link to ISP 1. If RTA at the same time probes the prefix of the destination it will measure long RTT times and traffic for that prefix will now be directed to RTB instead, causing saturation and increased RTT on the external link of RTB, and we have an oscillating behaviour. To prevent this type of traffic oscillation, we can store the link utilization with each probe result and take the weighted mean so that RTT sample values sampled at low link utilization will have greater weight in the RTT mean than samples taken at high utilization. To model a decay of sample importance over time it would also be suitable to apply an exponential reduction of the weights for the older values.

Compatibility

Prouting has to be implemented at the external border routers. Internal BGP routers do not have to be modified. It is not even required to run Prouting at all exit points. Consider an AS multihomed to 3 ASes but only two of the eBGP routers does Prouting. The "normal" eBGP router sets the default LP below the RTT_LP range. If a prefix is probed, the preferred exit point will be the one of the Prouters having the best RTT to that prefix. It might happen that the "normal" EGBP router had better RTT, but it did not measure it, so statistically it is favourable to choose the best exit between the Prouters since we know it is better than at least one of the other exit paths.

To override the Prouting behaviour for certain routes eBGP routers (including Prouters) can set LP above the RTT_LP range.

Additional considerations

For a multihomed AS with 2 exit points, 50% the inbound ICMP replies to the probes sent out will not return to the AS at the same Prouter which originated it. With reference to picture 1 consider this scenario: RTA sends a probe for destination D. The ICMP reply might come back to RTB which will forward it to RTA inside the AS. RTA's measured RTT now includes the intra AS transit time from RTB to RTA. If this intra AS transit time adds a significant amount to the measured RTT, the router being the entry point for traffic from destination D (in this case RTB) would always be favoured. But if inter AS transit times are large compared to RTTs measured on the internet it would be better to prefer the exit point with the lowest IGP metric instead of using Prouting.

References

[1] Bradley Huffaker, Marina Fomenkov, Daniel J. Plummer, David Moore and k claffy, *"Distance Metrics in the Internet"*, 2002, http://www.caida.org/outreach/papers/2002/Distance/distance.pdf