The Great Slowdown

As many countries and ICP providers tend to optimize their "internal" data access by applying tools to control or monitor users, the overall network condition deteriorates and performance can be reduced to significant levels. This situation leads to considerable increases in the cost for companies who want to provide applications to end users. Our research has shown that the Internet speed - not only in China - varies more than expected on a global scale and quite often, the Internet does not use optimized routes leading to temporary or permanently slow access to some applications or for a group of users. GrandeNet can optimize up to 80% of the connectivity of certain servers by usingRe6st and maintaining optimized and stable routes between all connected servers.

Can you Re6st?

Re6st is a multiprotocol random mesh generator that uses theBabel routing protocol to discover optimizal routes between each point in the mesh. It supports IPv6 and IPv4 with RINA support coming soon. It is commercially used byVIFIB our distributed cloud provider helping to solve the current lack of reliability of Internet connectivity for distributed enterprise applications due to bugs in routers, packet inspection breaking TCP protocol, government filters filtering too much, etc. Without re6st, it would have been impossible to deploy critical business applications used by large companies (Mitsubishi, SANEF, Aide et Action, etc.) on a decentralized cloud. It would also be impossible to manage the deployment of distributed cloud in Brazil, China or Ivory Coast where the Internet is even less reliable.

IPython Notebook

IPython Notebook is a web-based interactive computational environment for creating Executable Notebooks with Embeeded Python Code. IPython Notebook is largely used by researchers to produce and share their scientific work. We chose to use IPython Notebook for this article to provide a transparent walkthrough of what we are doing.

This article is fully reproducible by importing this notebook on your IPython Notebook Instance.

In order to execute the this notebook we need some well-known python libraries, specifically, panda, numpy, scipy and matplotlib. Below are the imports required to initialize the necessary libraries.

▶ Details

Next, we'll define the core code - written as methods which will perform the Data Collection and the Calculation of the results for this article. If you are not interested in Code, you can move directly to the next section of this article.

▶ Details

Measuring Performance with SlapOS Distributed Monitoring

The core of GrandeNet Infrastructure is based on servers distributed on multiples cloud providers (Amazon, Qincloud, OVH, Rackspace, UCloud...) as well as standalone machines distributed on companies offices and/or people's home. Customers may add their servers located on their premises or even at their homes to be used as their main production servers.

This hybrid and heterogenious infrastrucuture ofGrandeNet uses SlapOS to manage and monitor all distributed servers around the globe.

In this article we used a small set of servers (12) with public IPv4 runningSlapOS Distributed Monitoring. Each server tries to contact (using ICMP Protocol) all other 12 servers using IPv4 and IPv6 addresses. Tests are performed 10 times (10 pings) every 10 minutes and we get the average and packet loss for testing and comparison.

The image bellow ilustrates the tests with using just 3 servers:

Below we initialize the location for each servers' logs along with labels to improve the readability of the charts and results.

▶ Details

We also limit the scope of this article to the tests performed on a certain date range, shown below by the variable "DAY".

▶ Details

Collecting Data from Distributed SlapOS Monitoring

In order to produce results for this article, we use the methods defined above, crawl the logs and turn them into dataframes. These dataframes contain the test results for the indicated period (DAY above).

▶ Details

Internet IPv4 vs Grandenet IPv6

Using the dataframes we can visualize a comparison of the response time (in milliseconds) between using Internet IPv4 (red) vs Grandenet IPv6 (blue). As we are using the ICMP Protocol to measure the response time, the charts below use the name "ping" for IPv4 and ping6 for IPv6 and highlight the differences between the Internet IPv4 and the IPv6. The smaller the response time, the lower the plotted line, the better.

▶ Details

Comparison of Average Ping Response for IPv4 and IPv6 Address

It is well-know that latency has a direct influence on performance of Web Applications. By comparing the averages of ping responses on IPv4 and IPv6, you may notice significant improvements between the servers.

By using Grandenet, we can consider 3 states for a connection between 2 servers:

- Connection is OPTIMIZED when IPv6 is faster than IPv4 by more than 15% or 20ms
- Connection is BAD when IPv4 is faster than IPv6 by more than 15% or 20 ms.
- Connection is OK when IPv4 is faster than IPv6 by less than 15% or 20 ms

These states acknowledge the fact that re6st may increase a ping response by 20ms in the worst case, which is 2 servers being really far apart having a direct connection.

An optimal scenario is to get all servers on OPTIMIZED or OK states while testing each other. The table below contains a list of all connection between servers previously mentioned. You can notice that not only connections involving a Chinese Server got optimized, but connection between Japan to Singapore, Japan to Hongkong, US to Hongkong and others were optimized as well.

▶ Details

How fast can we get? Is there a Speed Limit?

Our monitoring experiment managed to surpass even our high expectations. We observed that GrandeNet globally reduced latency between 15% to 30% and packet loss to lower than 1% (a 5-10x improvement). for a group of computers if compared with standard IPv4.

Considering all individual links between monitored servers and ping response (latency), between 50% and 70% of connections were optimized through use of GrandeNet with packet loss being lowered on 45% to 65% of all connections.

GrandeNet still doesn't use a fully connected mesh configuration, so we also observed that a small portion of connections decreases in quality when using GrandeNet IPv6. However, this issue can be resolved through monitoring by creating additional direct tunnels between the servers that show standard IPv4 to be faster. By creating these direct tunnels, we thus stabilize the mesh because it will only include optimized or same speed links (our additional direct tunnels) on GrandeNet.

▶ Details

Conclusion

As mentioned above our sample showed GrandeNet IPv6 reducing latency between 15-30% and packet loss lowered to below 1% (5-10x improvement) versus using standard IPv4 connections. 50-70% of overall connections could be optimized lowering packet loss on 45-65% of them. The GrandeNet IPv6 final average response time of 127.46ms finished far ahead of the default IPv4 average response of 161.75ms.

Overall 21.20% of all pings were optimized using GrandeNet with packet loss being reduced from 3.12% (IPv4) to 0.52% (IPv6) showing that monitoring and optimizing for network connectivity can result in significant benefits. Creating tools automating our experiments and providing easy to use applications for users to display the routing situation in real time along with connectivity problems will be the next logical step.