How is MOS calculated in PingPlotter Pro?

Saved From: https://www.pingman.com/kb/article/how-is-mos-calculated-in-pingplotter-pro-50.html

Question

PingPlotter Pro now has an 'MOS' column and alert. How is MOS calculated in PingPlotter?

Solution

MOS (mean opinion score) is a voice call quality metric. It is used by the VoIP industry to put a number on voice quality. In reality, this number is a subjective 'opinion' rating of call quality given by someone who was just on a voice call.

Obviously, with PingPlotter we don't have someone doing a voice call on your line all the time, and then giving an opinion number. In fact, there's probably not even a voice call going on over your network much of the time for us to measure.

There are a number of factors that play into an MOS rating, and some hardware and tool vendors have put formulas in place that can 'estimate' an MOS, based on a number of variables. This is usually done by having both sides of the connection measure certain characteristics of the connection, and collaborate on the quality of the line and connection. For best results, you might take samples from the line at a similar rate to what a voice call might send data (numerous times a second).

PingPlotter's collection methods are a bit different from all of this - we're sitting on one side of a connection, and the other end has no idea that we're measuring anything. It might be a web server, or a game server, or any variety of things, none of which care to collaborate on estimating how a person would measure voice quality over that connection. We're also measuring only once every second - or possibly less often, depending on the trace interval.

MOS is an interesting number, though, and if we take a few variables that we know negatively impact call quality, we can put together a very rough approximation of what the voice call quality might be over that line.

There are 3 factors that significantly impact call quality:

- Latency
- Packet loss
- Jitter

All of these are well known by PingPlotter, covering the samples we take. Is there any way to put these three values together and come up with a MOS 'guess' that has some similarity to real life? We think there is, and PingPlotter Pro attempts to do so.

One advantage of estimating MOS with PingPlotter is that we also know Latency, Packet Loss and Jitter at any intermediate hops, so we should be able to estimate MOS for not just the final destination, but also for each intermediate hop! This would give us an easy way to see at which hop MOS is deterioriating, and can help us identify where the problem is.

So, the question is: How do we convert packet loss, latency and jitter numbers to an MOS score? That's a

This knowledgebase article and content copyright by Pingman Tools, LLC

great question! We've built one model for this, but we'd certainly love participation from anyone who wants to improve this formula!

First, let's define our 3 metrics: Packet Loss, Latency and Jitter. Remember that PingPlotter lets you pick a 'window' of samples to examine - we call this 'Samples to Include'.

1) Packet Loss percentage.

For the sample set, this is the percentage of packets that never made it from us, to the target server (or intermediate hop) and then back again. If we sent out 100 packets and only recieved 97 back (3 didn't make it), then we have 3% packet loss.

2) Average Latency.

The average latency (in PingPlotter) is the average (mean) time it takes a packet to get from your computer, to the target server, and then back again.

The average latency is the total of all latencies, divided by the number of samples that we're measuring. If we sent out 100 samples and received back 97, then total all latencies and divide by 97 to get the average.

3) Jitter

Jitter is a measurement of how much latency changes, from sample to sample. A low jitter number indicates a solid, good connection. A high jitter number is a sign that there is conjestion on the network. Too high of a jitter number has a negative impact on voice quality, causing delays, requiring resends, and wreaking havoc on transmission of voice.

To measure Jitter, we take the difference between samples, then divide by the number of samples (minus 1).

Here's an example. We have collected 5 samples with the following latencies: 136, 184, 115, 148, 125 (in that order). The average latency is 142 - (add them, divide by 5). The 'Jitter' is calculated by taking the difference between samples.

136 to 184, diff = 48
184 to 115, diff = 69
115 to 148, diff = 33
148 to 125, diff = 23
(Notice how we have only 4 differences for 5 samples). The total difference is 173 - so the jitter is 173 / 4, or 43.25.

We use this same mechanism no how many samples we have - it works on 5, 50 or 5000.

OK, so we have some numbers - what formula do we plug this in to?

MOS is a scale of 1 to 5, 5 being an 'excellent' call, 1 being completely unacceptable. In reality, even a perfect connection is impacted by the compression algorythms of the codec, so the highest score most codecs can get is in the 4.2 to 4.4 range. Most tool-based solutions calculate what is called an 'R' value and then apply a formula to convert that to an MOS score. We do the same. This R to MOS calculation is relatively standard. The R value score is from 0 to 100, where a higher number is better.

So, this gets us to a the bottom line - how do we get an R Value. This documentation is taken from a pre-release of PingPlotter Pro - this formula may have changed since this was written. In addition, all MOS calculations are done with scripts in PingPlotter Pro - which can be modified by you.

This knowledgebase article and content copyright by Pingman Tools, LLC

We start out with an R-value of 93.2, and deduct from there, based on network conditions.

Here is the formula we use (note that this is 'psuedo-code', not real code):

```
' Take the average latency, add jitter, but double the impact to latency
' then add 10 for protocol latencies
EffectiveLatency = ( AverageLatency + Jitter * 2 + 10 )
' Implement a basic curve - deduct 4 for the R value at 160ms of latency
' (round trip).  Anything over that gets a much more agressive deduction
if EffectiveLatency < 160 then
   R = 93.2 - (EffectiveLatency / 40)
else
   R = 93.2 - (EffectiveLatency - 120) / 10
' Now, let's deduct 2.5 R values per percentage of packet loss
R = R - (PacketLoss * 2.5)
' Convert the R into an MOS value.(this is a known formula)
MOS = 1 + (0.035) * R + (.000007) * R * (R-60) * (100-R)
```

That's it! This formula is captured in the 'MOS Column.ppx' script in your PingPlotter Pro 'scripts' directory. If you want to change the equation, please do so!

Note: In PingPlotter Pro v3.0, the MOS script is turned off by default. To turn it on, go to Edit -> Options..., Plugins & Scripting. Turn on the checkmark beside the 'MOS Column' script. Close the dialog (OK) and then restart PingPlotter Pro.

This knowledgebase article and content copyright by Pingman Tools, LLC