***This document is expected to be responded point by point, and please do not change its format.***

|  |
| --- |
| You defined ***6 flows*** in the simulation as shown:    Q1-1: There is only four end-to-end simulation results as shown:    So this is wrong, there should be 6 end-to-end simulation results for 6 flows.  You should fix it.  Q1-2: The value of the end-to-end results are **wrong**. These end-to-end latency values are around 3.5 microseconds, which is impossible. This appears to be a one-hop delay, not an end-to-end delay. Note that end-to-end latency is the delay it takes to get from the source to the destination.  Q1-3: As shown below, the end-to-end delay name you define can easily be misunderstood by customers. For example, the first one is “End-to-end delay from gwTSNBus.app0”, however, the “gwTSNBus.app0” is not the source node, the source node must be a Bus node. Similarly, the second one is "End-to-end delay from gwTSNad1.app0", "gwTSNad1" is not the source, the source must be a certain wireless node. This is also the reason why your measured end-to-end latency is very low. Because you're not measuring the latency from source to destination.    **You may be able to measure end-to-end latency in segments.** For example, there are two flows:  Path of Flow1: Busnode1, gwTSNBus, TSNbackbone, TSNSW1, gwTSNad1, AP1, WirelessNode1, WirelessNode2  Path of Flow2: WirelessNode3, WirelessNode4, AP2, gwTSNad2, TSNSW2, TSNbackbone, gwTSNBus, Busnode2  The end-to-end delay you measured seems to be the path delay of the part highlighted in yellow above. You just need to add the existing delay to the delay of the remaining parts. I will explain in more detail.  Taking Flow2 as an example, your current method has obtained the delay from "gwTSNad2" to "gwTSNBus," which we denote as T1. Then, you need to measure:   * the delay from “WirelessNode3” to “AP2”, denoted as T2. * The delay from “AP2” to “gwTSNad2”, denoted as T3. * The delay from “gwTSNBus” to “Busnode2”, denoted as T4. * Finally, the actual end-to-end delay is:   Q1-3-1: You should fix the bug of measuring the end-to-end delay.  Q1-3-2:You should change the name of end-to-end delay as “E2EforFlowname”, so that we can clearly know the end-to-end delay of a specific flow. This is a mapping between end-to-end delay and traffic. For example, there are 6 flows in the networks as shown:    Then, the names of simulation end-to-end delay should be: E2EforFlow0, E2EforFlow1, and so on.  Meanwhile, other simulation results, i.e., jitter, bit error rate, packet loss rate, should also adopt the same naming convention. |
| R1-1: |
| R1-2: |
| R1-3-1: |
| R1-3-2: |

|  |
| --- |
| Q2-1: The name of “bit error rate” should be changed as same as Q1-3-2.    Q2-2:  There are 6 flows, why there are 9 results of bit-error-rate? There should be 6 results of BER for 6 flows. |
| R2-1: |
| R2-2: |

|  |
| --- |
| Q3-1: The name of “packet loss rate” should be changed as same as Q1-3-2.  Q3-2:  The packet loss rate for all flows should be zero, (i.e., NaN). There are 6 flows, why there are 9 results of packet loss rate. |
| R3-1: |
| R3-2: |

|  |
| --- |
| Q4-1: The name of “packet loss rate” should be changed as same as Q1-3-2.  Q3-2:  The packet loss rate for all flows should be zero, (i.e., NaN). There are 6 flows, why there are 9 results of packet loss rate. |
| R3-1: |
| R3-2: |

|  |
| --- |
| Q5: Even there are nodes go offline, the contents of shared memory “hyperNet-online” are not changed. This is a bug, when the nodes go offline, the topology should be updated, and also, the routes and TSN schedules should be updated. So the contents of “hyperNet-online” must be changed.  I utilized the C++ code “sharedMemoryCodes.txt” to test your program, please copy the contents to a C++ source file. If you think my codes have bugs, then you should provide your shared memory test codes. |
| R5 |