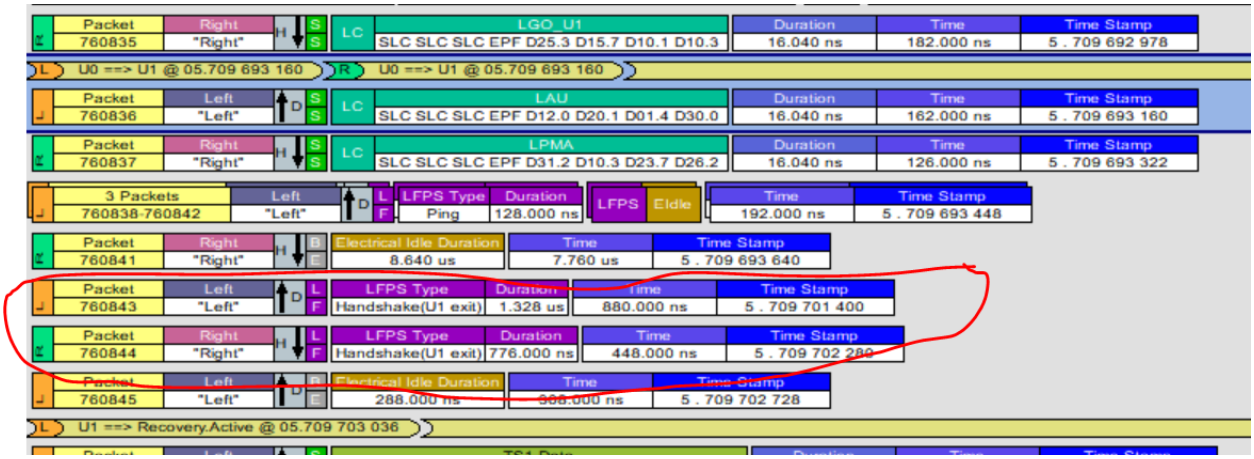
We have analyzed a typical working case and your case as follows:

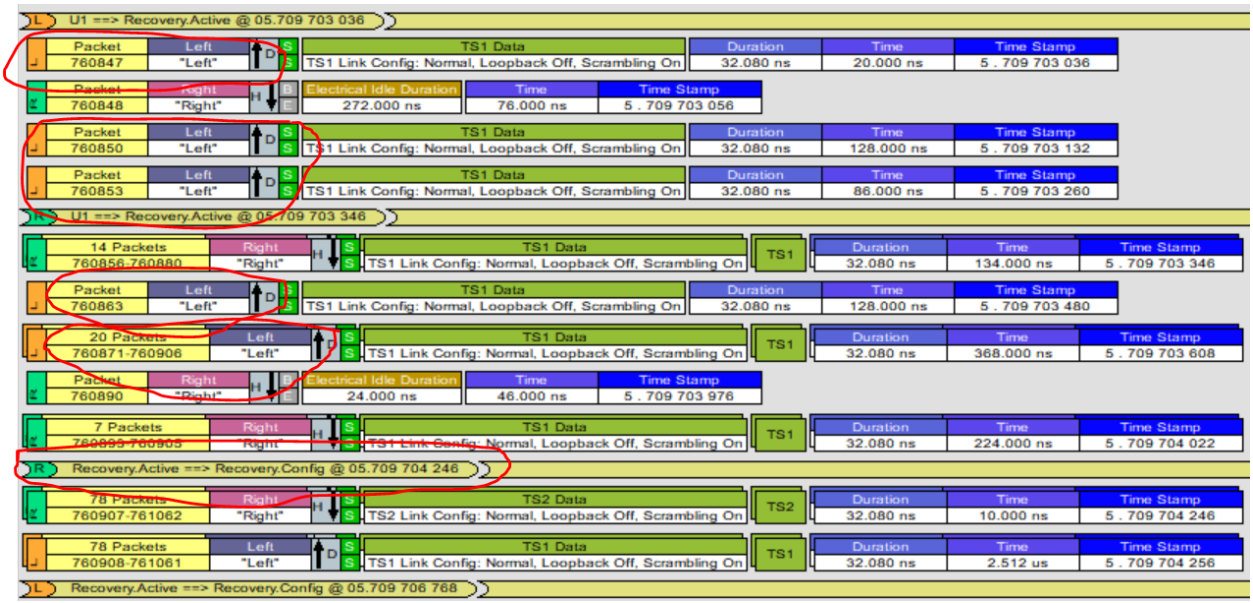
A typical U1 to U0 Sequence is mentioned below:

1. Both the ports (UpStream and DownStream) exit from the U1 state after exchanging the LFPS handshake (U1_Exit) and enter to Recovery.Active. The following snapshots are taken with FX3 Explorer Kit and Intel Host Controller.

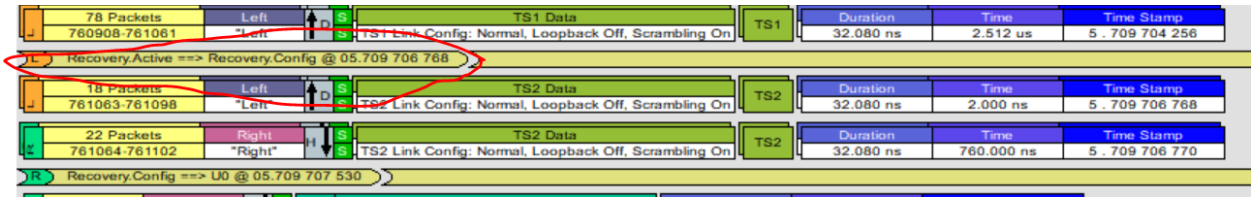
   Note that either device or host can initiate the U1 Exit. Hence, either device or host can start sending TS1 and enter into Recovery.Active state. In this case, device initiated the U1 exit hence device started sending TS1 first. (Here, R is the host and L is the device)



2. The port in SuperSpeed operation shall transition to Recovery.Configuration after eight consecutive and identical TS1 or TS2 ordered sets are received.
   In this, the host entered to Recovery. Configuration first as shown below.

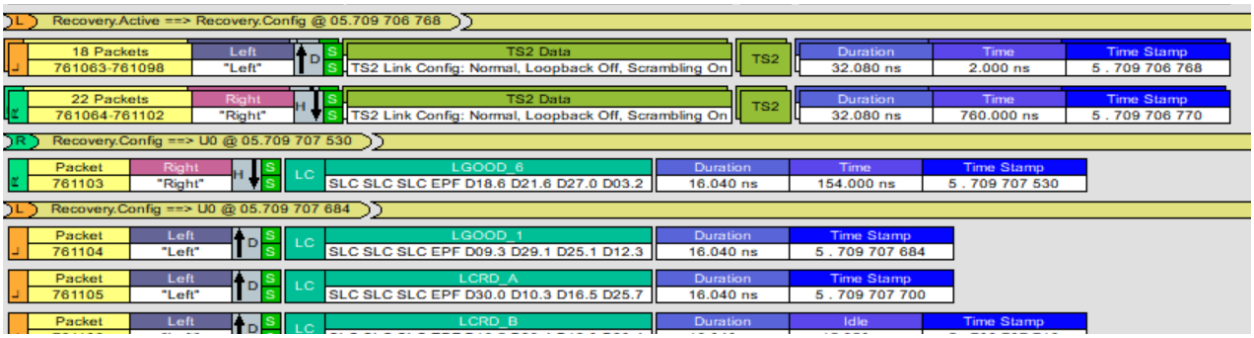


3. Later, device enters to Recovery. Configuration as shown below



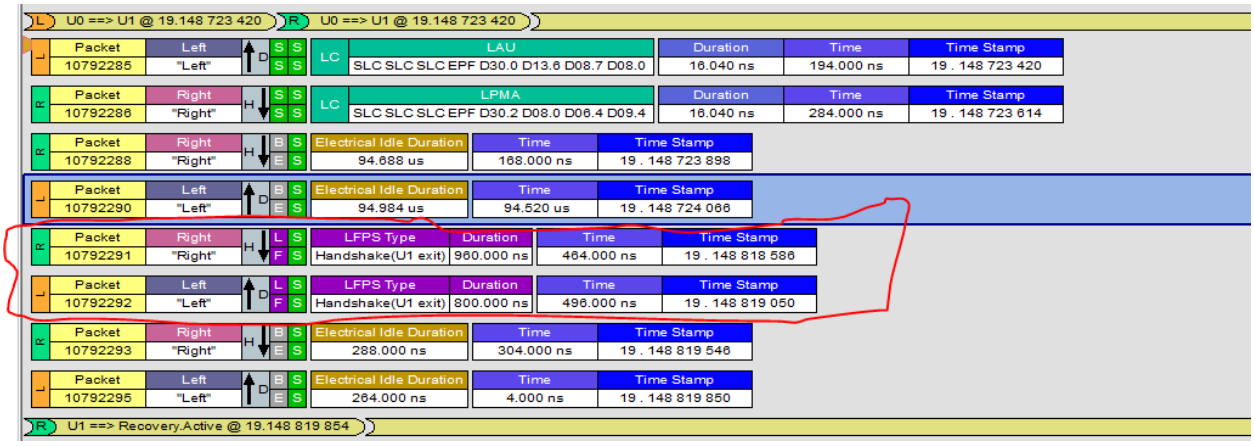
4. The port in SuperSpeed operation shall transition to Recovery.Idle after the following two conditions are met:
   a. Eight consecutive and identical TS2 ordered sets are received.
   b. Sixteen TS2 ordered sets are sent after receiving the first of the eight consecutive and identical TS2 ordered sets.

5. Note that we cannot see Recovery.Idle in the Trace. We can see that the device and host enter into U0 instead. Below is snapshot showing the same.

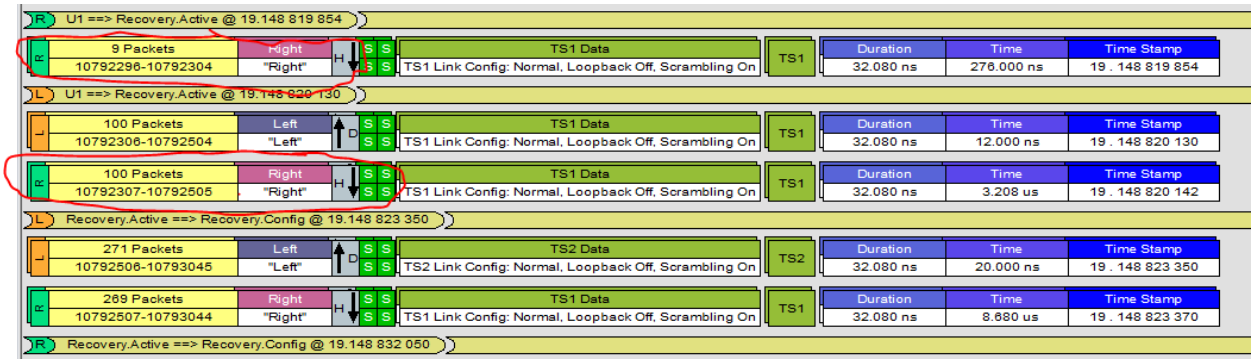


Let us analyze your case.

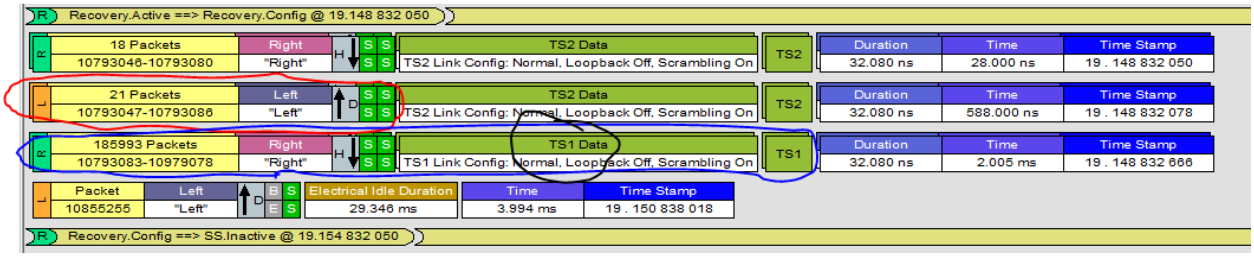
1. In your case, host initiated U1 Exit then start sending TS1 and enter into Recovery.Active state. The snapshot is shown below:



2. Device has seen 8 consecutive and identical TS1 from the host and entered into Recovery.Config. Then Device started sending TS2s.



3. Later, host seen 8 consecutive and identical TS1 from device then entered into Recovery.Config. Then Host started sending TS2s



4. In above snapshot, we can see that the host has sent TS2 after entering into Recovery. Config. Then it sent TS1, which is abnormal behavior of the host controller.
   Once the port starts sending TS2s after entering Recovery.Config, it should not send TS1 again.
   You can also see that the device is continuously sending TS2s as expected.

The work around for this host problem is – Disabling the Low Power Entry using LpmDisable in the right place in the firmware:

If you call LPMDisable in SET_CONFIG as done USB Bulkloop example FW, the device will fail the compliance test.

You need to call LPMDisable just before the data transfer starts. You can send a vendor command for the doing the same OR you can handle it in DMA Producer Event based on Timer (Start the timer in application start and reset in the DMA_Producer Event. If the timer expires, it means there is no data transfer, then enable the LPM by calling LPMEnable API) in the case of USB Bulk Loop example. You need configure the DMA as Manual DMA Channel to get DMA_Producer Event.

However, since your application is UVC, please refer to AN75779 example firmware. Here, we have called LPMDisable in **`CyFxUvcApplnStart`**, which is called when there is STREAM_START Event form the host.