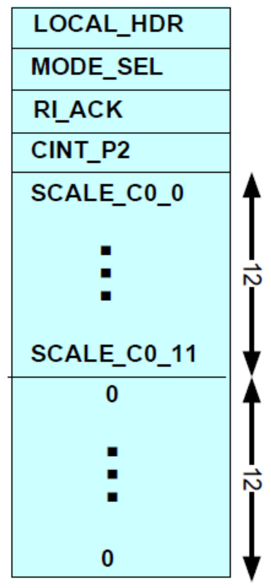
BCP SSL Noise scaling factors

1. Explanation: One of the inputs to the BCL SSL is a set of noise scaling factors. The noise scaling factor is defined as $c = \frac{1}{2\sigma_N^2}$ where σ_N^2 is 1-dimensional effective noise variance. This is the effective noise at the input to the soft slicer. This scaling factor may change from symbol (PAM or QAM) to symbol (WiMAX) or it may be the same for a group of symbols (LTE, WCDMA case). If the variance of the effective complex noise in the complex QAM symbol at the input to the soft slicer, σ_C^2, is estimated then $\sigma_C^2 = 2\sigma_N^2$ and $c = \frac{1}{\sigma_C^2}$.

The noise scaling factors are supplied to the BCP (in IEEE 754 format) in SSL header parameters as shown below for LTE single layer case (from BCP user guide) and defined in sections **4.7.3.16 SCALE_C0_0_CFG** to **4.7.3.39 SCALE_C1_11_CFG**.

Figure 4-82 Single Layer LTE Packet Header (28 words)



2. Sample code: The following example code taken from the BCP PDK LTE test provides a way to calculate the noise scaling parameters for LTE single layer case, given the one dimensional noise variance.

```
float snrEven = +1.9542e+00    # Odd symbols (test value read from text file in the PDK test)
float snrOdd =  +6.4167e-01    # Even symbols (test value read from text file in the PDK test)
Uint16 rms = 1024; //convert the data into fix point
float noiseVar;

 /*snrEven is 1/noiseVar (one-dimension) */
```

```
    noiseVar = (float) snrEven/rms/rms/2.f;
    scaleEven = * ((Int32 *) &noiseVar);

   /*snrOdd is 1/noiseVar (one-dimension) */
    noiseVariance = (float) snrOdd/rms/rms/2.f;
    scaleOdd = * ((Int32 *) &noiseVar);
```

The above code explained:
As indicated in item 1, the noise scaling factor should be computed as scale = 1/(2*noiseSigma). Now, SNR is defined as SNR = signalSigma/noiseSigma, where, in turn, signalSigma = rms*rms.

```
So, scale = 1/(2*noiseSigma)
          = (1/2)*(1/ noiseSigma)
          = (1/2)*(SNR/ signalSigma)
          = (1/2)*(SNR/( rms*rms)), i.e. the computation implemented in the highlighted code.
```

The next statement saves the integer representing the floating point number. The above code computes two scaling factors namely scaleEven and scaleOdd which are mapped to the SSL header parameters as per the following code (file test_lte_ul.c in BCP PDK test):

```
if (radioStd == Bcp_RadioStd_LTE)
{
   for(idx = 0; idx < (numOFDMSymPerSubfrm + 1)/2; idx++)
   {
      pSslHdrCfg->scale_c0 [idx] = scaleEven;
   }

   for(idx = (numOFDMSymPerSubfrm + 1)/2; idx < numOFDMSymPerSubfrm; idx++)
   {
      pSslHdrCfg->scale_c0 [idx] = scaleOdd;
   }

   if (numLayers == 2)
   {
      for(idx = 0; idx<(numOFDMSymPerSubfrm + 1)/2; idx++)
      {
         pSslHdrCfg->scale_c1 [idx] = scaleEven;
      }

      for(idx = (numOFDMSymPerSubfrm + 1)/2; idx<numOFDMSymPerSubfrm; idx++)
      {
         pSslHdrCfg->scale_c1 [idx] = scaleOdd;
      }
   }
}
```