We started with a simple task set consisting of two tasks and a single resource, for which both tasks contend. The task set we considered is $(\phi, p, e, d)$:

A (0, 8, 2, 8) and B (1, 5, 1, 5).

Resource: X
Resource Usage: A requires X for 1 time unit, from 0. B requires X for 1 time unit starting from 0.

Under normal EDF-PIP, the blocking term per resource per task, is calculated as follows:

$$b_{i,R} = \max_{1 \leq k \leq n}(\theta_k) \rightarrow (1)$$

[where $\theta_k$ is the length of the critical section of every task using the resource $R_j$]

Now, the blocking term for task i is

$$b_i = \max_{1 \leq l \leq j}(b_{i,R_l}) \rightarrow (2)$$

Here,

$b_A = 1$ and $b_B = 1$.

This task set is schedulable (from the following schedulability test).

$$\sum_{k=1}^{n} \frac{e_k}{\min(d_k, p_k)} + \frac{b_i}{\min(d_i, p_i)} \leq 1 \rightarrow (3)$$

Now, we consider EDF-PIP with DVS. We consider the static DVS mechanism.

We need to find the scaling factor $\alpha$, such that

$$\frac{e_1}{p_1} + \frac{e_2}{p_2} \leq \alpha \rightarrow (4)$$

The lowest possible value of $\alpha$ is 0.45 in this case. Now, we scale the execution time of each task by this factor. The scaled values for the execution times are:

$e_1 = 2 / \alpha = 2 / 0.45 = 4.44$
$e_2 = 1 / \alpha = 1 / 0.45 = 2.22$

Similarly, the blocking times also need to be scaled.

$b_1 = 1 / \alpha = 1 / 0.45 = 2.22$
\[ b_2 = 1 / \alpha = 1 / 0.45 = 2.22 \]

Once the scaling factor has been considered for both the execution and blocking times, the scheduling is exactly like normal EDF-PIP and is schedulable.

The blocking term for \( n \) tasks scheduled using EDF-PIP with Static-DVS is the same as the results obtained in equations (1) and (2) above. Care must be taken to ensure that the blocking times thus obtained are also scaled by \( 1 / \alpha \).

Note: In this particular example, we have used the lowest possible value for \( \alpha \) obtained from equation (4). However, this may prove to be an over-optimistic value as blocking has not been considered in the calculation for \( \alpha \). In general, all task sets need not be schedulable if \( \alpha \) is calculated in this way.

The correct way to estimate \( \alpha \) is by using the following equation in place of equation (4):

\[
\sum_{k=1}^{n} e_k / \min(d_k, p_k) + b_i / \min(d_i, p_i) \leq \alpha \implies (5)
\]

Since this is a static algorithm, it might be worthwhile to do the following:
(a) Find the lowest possible value of \( \alpha \) using equation (4)
(b) Scale the value of \( b_i \) using this value of \( \alpha \)
(c) Perform the schedulability test according to equation (3)
(d) If this test fails, find \( \alpha \) using equation (5) and proceed.

The value of \( \alpha \) calculated in both cases would be significantly different.

Future course of work:
(a) Perform a formal analysis to verify the above results.
(b) Analyse for Cycle-Conserving DVS for EDF-PIP.
(c) Analyse for Look-Ahead DVS for EDF-PIP.

The work presented thus far had equal contributions by all three team members. We intend to continue working together for phase (a) mentioned above and then split the remaining work.