<u>Three Codes of TCP Vegas Modification</u>

The idea if to let Vegas find out for itself the optimum value for α and β rather than fixing it at 1 and 3 respectively.

Code 3:

- No change in the slow start mechanism.

In the increase-decrease mode:

- <u>If $Th_{diff} < \alpha$:</u>
  - If $\alpha > 1$
    - If $Th_{act(n)} > Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → Increase cwnd only.
    - If $Th_{diff} < \alpha$ and $\alpha \neq 1$ and $Th_{act(n)} < Th_{act(n-1)}$ and cwnd has not been decremented after the packet has been sent → Decrease cwnd. Set $\alpha = \alpha - 1$ and $\beta = \alpha + 2$.

      This is to compensate for the increase that may occur in next step, when needed. When α is too large and congestion occurs, a small throughput can still make $Th_{diff} < \alpha$. Instead of decreasing the cwnd (congestion), we would have been increasing cwnd, had we followed the present algorithm.
  - If $\alpha = 1$
    - Increase cwnd.

- <u>If $\beta > Th_{diff(n)} > \alpha$:</u>
  - $Th_{act(n)} > Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → Increase cwnd. Set $\alpha = \alpha + 1$ and $\beta = \alpha + 2$

    The reasoning is that even though $Th_{diff} > \alpha$, the Th_{diff} is decreasing. This means that actual throughput is increasing. So the value of α that we have assumed is preventing us from making use of more available bandwidth. So we have got to increase α.
  - $Th_{act(n)} <= Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → No change in cwnd, α and β.

- If $Th_{diff} > \beta$, decrease cwnd, α, β.

- If none of the above conditions are satisfied, do not update cwnd, α and β.

Code 2:

In the increase-decrease mode:

- If $Th_{diff} < \alpha$:
  - If $\alpha > 1$
    - If $Th_{act(n)} > Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → Increase cwnd, $\boldsymbol{\alpha = \alpha+1, \beta = \beta+1}$.

    - If $Th_{diff} < \alpha$ and $\alpha \neq 1$ and $Th_{act(n)} < Th_{act(n-1)}$ and cwnd has not been decremented after the packet has been sent → Decrease cwnd. Set $\alpha = \alpha - 1$ and $\beta = \beta - 1$.

      This is to compensate for the increase that may occur in next step, when needed. When α is too large and congestion occurs, a small throughput can still make $Th_{diff} < \alpha$. Instead of decreasing the cwnd (congestion), we would have been increasing cwnd, had we followed the present algorithm.
  - If $\alpha = 1$
    - Increase cwnd, $\boldsymbol{\alpha = \alpha+1, \beta = \beta+1}$.

- If $\beta > Th_{diff(n)} > \alpha$:

  - $Th_{act(n)} > Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → Increase cwnd. Set $\alpha = \alpha+1$ and $\beta = \alpha+2$

    The reasoning is that even though $Th_{diff} > \alpha$, the Th_{diff} is decreasing. This means that actual throughput is increasing. So the value of α that we have assumed is preventing us from making use of more available bandwidth. So we have got to increase α.

  - $Th_{act(n)} \le Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → No change in cwnd, α and β.

- If $Th_{diff} > \beta$, decrease cwnd, α, β.

- If none of the above conditions are satisfied, do not update cwnd, α and β.

Code 1:

In the increase-decrease mode:

- If $Th_{diff} < \alpha$:
    - If $\alpha > 1$
        - If $Th_{act(n)} > Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → Increase cwnd, **$\alpha = \alpha+1$, $\beta = \beta+1$**.

        - If $Th_{diff} < \alpha$ and $\alpha \neq 1$ and $Th_{act(n)} < Th_{act(n-1)}$ and cwnd has not been decremented after the packet has been sent → Decrease cwnd. Set $\alpha = \alpha - 1$ and $\beta = \beta - 1$.

          This is to compensate for the increase that may occur in next step, when needed. When α is too large and congestion occurs, a small throughput can still make $Th_{diff} < \alpha$. Instead of decreasing the cwnd (congestion), we would have been increasing cwnd, had we followed the present algorithm.
    - If $\alpha = 1$
        - Increase cwnd, **$\alpha = \alpha+1$, $\beta = \beta+1$**.

- If $\beta > Th_{diff(n)} > \alpha$:

    - $Th_{act(n)} > Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → Increase cwnd. **No change in α and β values**.

      The reasoning is that even though $Th_{diff} > \alpha$, the Th_{diff} is decreasing. This means that actual throughput is increasing. So the value of α that we have assumed is preventing us from making use of more available bandwidth. So we have got to increase α.

    - $Th_{act(n)} <= Th_{act(n-1)}$, and the cwnd has not been incremented after the packet was sent → No change in cwnd, α and β.

- If $Th_{diff} > \beta$, decrease cwnd, α, β.

- If none of the above conditions are satisfied, do not update cwnd, α and β