DRCP_State.IPP_Activity: Current Behavior (my interpretation)

**Set IPP_Activity bit of Neighbor DRCP State to TRUE if config info received from Bob is compatible with Alice: “Home has heard from Neighbor”**

**Neighbor DRCP State. IPP_Activity:**
Means Bob has heard from Alice, but not that Alice has also heard from Bob or that Bob knows Alice has determined the config to be compatible (i.e. Bob is only sure that there is one-way communication)

IPP_Activity bit of Home DRCP State is never modified
Portal System 1
“Alice”

Portal System 2
“Bob”

IPL

After first DRCPDU exchange both Bob and Alice have Neighbor.Ipp_Activity TRUE: “I know I can hear you and that your config is compatible with mine”

Next Step: Bob assumes Alice can hear him (and vice versa) and declares the portal is active.
E.g. begins using the Aggregator Key value for the Portal as opposed to the Key value for this individual Portal System.
DRCP_State.IPP_Activity: What comment requests:

Set IPP_Activity bit of Home DRCP State to TRUE if config info received from Bob is compatible with Alice: “Home has heard from Neighbor”

Set IPP_Activity bit of Home DRCP State to TRUE if config info received from Alice is compatible with Bob: “Home has heard from Neighbor”

Neighbor DRCP State. IPP_Activity:
Means Bob has heard from Alice, AND that Alice has also heard from Bob AND that Bob knows Alice has determined the config to be compatible (i.e. Bob is sure that there is two-way communication)
Why is this change significant?

Portal System 1 “Alice” → Portal System 2 “Bob” via IPL

After first DRCPDU exchange both Bob and Alice have Home.Ipp_Activity TRUE: “I know I can hear you and that your config is compatible with mine”

After second DRCPDU exchange both Bob and Alice have Neighbor.Ipp_Activity TRUE: “I know you can hear me and you agree that my config is compatible with yours”

Next Step: Bob knows Alice can hear him (and vice versa) and declares the portal is active.
Proposed Response to the comment

• The variable DRF_Neighbor_Oper_DRCP_State.IPP_Activity is important for the DRCPs state machine operations and should not be redefined. The IPP_Activity variable in the transmitted DRCPDUs just references the DRF_Neighbor_Oper_DRCP_State.IPP_Activity value. The DRF_Neighbor_Oper_DRCP_State.IPP_Activity is set or cleared to indicate DRCPDUs have been received from the Neighbor. All the other bits of the DRF_Neighbor_Oper_DRCP_State are copied from the received DRCPDU. This needs to be clarified in the recordNeighborState function.

• The current text: “This function sets DRF_Neighbor_Oper_DRCP_State.IPP_Activity to TRUE and records the parameter values for the Drni_Portal_System_State[] and DRF_Home_Oper_DRCP_State carried in a received DRCPDU [item $s$] in 9.4.3.2] on the IPP, as the current parameter values for Drni_Neighbor_State[] and DRF_Neighbor_Oper_DRCP_State associated with this IPP respectively.” should be replaced with:

  “This function sets DRF_Neighbor_Oper_DRCP_State.IPP_Activity to TRUE and records the parameter values for the Drni_Portal_System_State[] and DRF_Home_Oper_DRCP_State carried in a received DRCPDU [item $s$] in 9.4.3.2] on the IPP, as the current parameter values for Drni_Neighbor_State[] and DRF_Neighbor_Oper_DRCP_State associated with this IPP respectively, with the exception of the IPP_Activity bit. In addition, if the received DRF_Home_Oper_DRCP_State.IPP_Activity == FALSE then DRCPDU DRF_Home_Oper_DRCP_State.Gateway_Sync = FALSE”
Proposed Response to the comment

- The variable DRF_Neighbor_Oper_DRCP_State.IPP_Activity is important for the DRCPs state machine operations and should not be redefined. **The IPP_Activity variable in the transmitted DRCPDUs just references the DRF_Neighbor_Oper_DRCP_State.IPP_Activity value.** The DRF_Neighbor_Oper_DRCP_State.IPP_Activity is set or cleared to indicate DRCPDUs have been received from the Neighbor. All the other bits of the DRF_Neighbor_Oper_DRCP_State are copied from the received DRCPDU. This needs to be clarified in the recordNeighborState function.

- The current text: “This function sets DRF_Neighbor_Oper_DRCP_State.IPP_Activity to TRUE and records the parameter values for the Drni_Portal_System_State[] and DRF_Home_Oper_DRCP_State carried in a received DRCPDU [item s) in 9.4.3.2] on the IPP, as the current parameter values for Drni_Neighbor_State[] and DRF_Neighbor_Oper_DRCP_State associated with this IPP respectively, with the exception of the IPP_Activity bit. In addition, if the received DRF_Home_Oper_DRCP_State.IPP_Activity == FALSE then DRCPDU DRF_Home_Oper_DRCP_State.Gateway_Sync = FALSE”

If the IPP_Activity variable in the transmitted DRCPDUs comes from the Neighbor DRCP state, how can we reference it as “the received DRF_Home_Oper_DRCP_State.IPP_Activity”?
DRCP_State.IPP_Activity: What the proposed response suggests:

Set IPP_Activity bit of Neighbor DRCP State to TRUE if config info received from Bob is compatible with Alice: “Home has heard from Neighbor”

Neighbor DRCP State. IPP_Activity: Means Bob has heard from Alice, but not that Alice has also heard from Bob or that Bob knows Alice has determined the config to be compatible (i.e. Bob is only sure that there is one-way communication)
What does the proposed response accomplish?

• On the plus side:
  – The proposed response does fix the bug that the Neighbor DRCP State.IPP_Activity is set TRUE and then immediately overwritten with the received DRCP State.IPP_Activity which is always FALSE.
    • The comment resolved this by making it the Home DRCP State.IPP_Activity is set TRUE and then the value received in the DRCPDU is stored in the Neighbor DRCP State.IPP_Activity

• On the minus side:
  – The way the IPP_Activity bit gets into the DRCPDU gets really screwy, but nothing fundamentally changes. Still only verify one-way communication.

• This resolution is not acceptable to the commenter