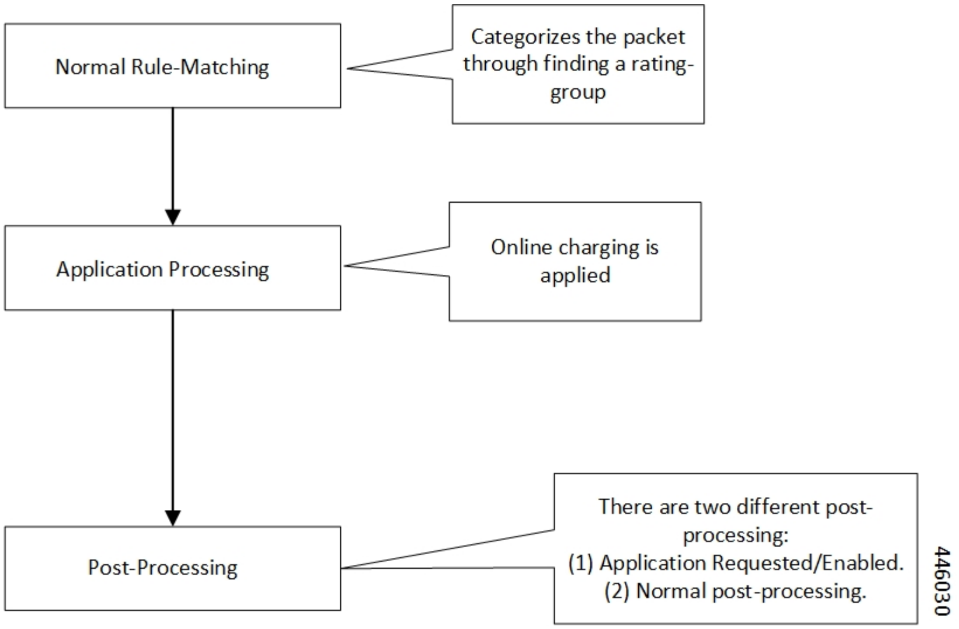
Post Processing Interaction for DCCA

Feature Description

The following diagram explains about the packet processing.

Figure 1: Post Processing Interaction for DCCA



Normal Rule-Matching
Categorizes the packet through finding a rating-group
Application Processing
Online charging is applied
Post-Processing
There are two different post-processing:
(1) Application Requested/Enabled.
(2) Normal post-processing.
446030

Normal Rule Matching

In this phase, A comparison happens between the incoming packet against the configured rules in the box. This rule matching process is nothing but categorizing the packet. Use the following CLIs for the Rule Matching configuration in the box.

```
action priority <priority-number> ruledef <ruledef-name>
charging-action <charging-action>
```

Based on priority order, the Rule Matching happens against the packet. The first rule that matches categorizes the packet.

The corresponding charging-action applies to the packet. If the charging-action configuration contains “cca charging credit”, then it triggers the online charging, for which the packet moves to the DCCA application.

Application Processing

Once the packet reaches the DCCA Application, it checks the quota for the packet (rating-group/content-id) and makes the necessary processing. When there are no more credits for that rating-group, the Final-Unit-Actions takes place on the packet. If no-credit is present in that rating-group, DCCA can also blacklist the rating-group. When the application is blacklisting, the packet gets marked for Discard/Drop. The packet is in the disposition-action to inform the ACS mgr. If the quota is present, the packet goes for forwarding. The DCCA application can alternatively populate the post processing rules/filter list and mark the packet for postprocessing. The postprocessing happens when the OCS has requested for applying the filter-ids or filter-rules along with the Final-Unit-Indication AVPs. Once the DCCA application processing completes on the packet, it goes back to the ACS mgr.

Post Processing

When the packet returns from the application, the ACS MGR, sees the disposition action value set by the DCCA Application. If it’s marked for discard, it gets discarded.

- Application Requested Post-Processing: If the disposition-action applies for PP_RESTRICTION_RULE or PP_FILTER_ID, it tries to get the corresponding restrict-rules-list or restrict-filter-id-list for the content-id/rating-group. It applies the postprocessing. The packet doesn’t attempt for the below-post-processing (General Post-Processing).

    - ACS_CONTROL_PP_RESTRICTION_RULE: This disposition action applies, when the DCCA activates Restriction-Filter-Rules sent by OCS, inside the Final-Unit-Indication Grouped-AVP, as per RFC 4006. The Restriction-Filter-Rules are applicable in “restriction_list”, inside the “fui_restrict_access”.

    - ACS_CONTROL_PP_FILTER_ID: This disposition action applies, when the DCCA activates the Filter-Ids, the OCS inside the Final-Unit-Indication grouped-AVP, as per RFC4006. The Filter-Ids are nothing but the rule def names, and are applicable in “filter_id_list”, inside the “fui_restrict_access”

      DCCA Application can set both the disposition actions. Disposition-action is nothing but a bitmask.

      These postprocessing restrict rules or postprocessing filter ids, that came from OCS and enabled/activated by DCCA Application. This rule is rating-group specific rules. The rule-matches happen in the order in which the OCS sends.

      For each acs_sub_sess, there’s a list of “dcca_mscc_fui_restrict_access_t”, indexed on “service_id & rating_group”. For each of this combination, the preceding type structure exists. This “dcca_mscc_fui_restrict_access_t” structure contains the “filter_id_list” & “fui_restrict_access” lists. This structure gets empty by default. The DCCA application can fill it when it activates the corresponding post processing filtering for that service-id + rating-group.

- General Post Processing: If it’s forward, the post processing starts. During the post processing, the packet gets matched against the configured post processing rules in the boxer.

  Configure the post processing rules in boxer using the following CLIs:

Post processing priority *<priority-number> ruledef* **<ruledef-name> charging-action** *<charging-action-name>*

These post processing rules get matched against the packet in the order of the priority-number.

Limit Reached Post Processing

In addition to the preceding two disposition action values, there's one more value for limit-reached scenarios, it's ACS_CONTROL_PP_LIMIT_REACHED. Here the limit-reached indicates that the user quota-limit is over. When the user quota is over, the packets get dropped by default, by application, and no post processing applies. The feature is to add control on this limit-reached scenario, where post processing configuration happens, even for this quota exhausted scenario.

A configurable option is available for enabling the post processing for limit-reached/quota-exhausted packets. Use the following CLI for this configuration:

```
configure
   active-charging service service_name
      rulebase rulebase_name
         post-processing policy { always | not-for-dynamic-discard }
         end
```

The option "not-for-dynamic-discard" is the default option. This option indicates that the post processing doesn't apply for the limit-reached/quota-exhausted scenarios.

In case of the "post processing policy always" CLI, the post processing rules applies for the limit-reached/quota-exhausted scenarios. The "ACS_CONTROL_PP_LIMIT_REACHED" value in the disposition action is to communicate about this behavior. If there are post processing priority-based rules, it checks for any redirection rules, else discards the packets by default. No other post processing actions like forward, next-hop, X-header-insertion applies on these limit-reached packets.

Configuring Post Processing

The post processing rule def with the limit-reached case have "cca qutoa-state = limit-reached" configured, along with the "rule-application post processing" option. This configuration is to indicate that this rule def is for the limit-reached scenario.

```
    ruledef http_low
      http any-match = TRUE
      cca quota-state = limit-reached
      rule-application postprocessing
    #exit
```

The corresponding charging-action has the "flow action redirect "configuration. Any other flow action values are invalid for the limit-reached scenario.

```
charging-action redirect
      flow action redirect-url http://webpages/index.html
    #exit
```

Configure the post processing priority rules in the rule base in such a way that the limit reached post processing rules is of the high priority. So that the packets get matched first against the limit-reached rule def.

```
rulebase base1
      ....................................
      post processing priority 1 ruledef http_low charging-action redirect
    #exit
```