Filter Policies

In This Chapter

The SROS supports filter policies for services and network interfaces (described in this chapter), subscriber management (integrated with service filter policies with the subscriber management specifics defined in the SROS Triple Play Guide), and CPM security and Management Interface (described in SROS Router Configuration Guide).

Topics in this chapter include:

- ACL Filter Policy Overview on page 410
  - → Filter Policy Packet Match Criteria on page 413
  - → IPv4/IPv6 Filter Policy Entry Match Criteria on page 413
  - → MAC Filter Policy Entry Match Criteria on page 417
  - → Filter Policy Actions on page 419
  - → Filter Policy Statistics on page 422
  - → Filter Policy Logging on page 423
  - → Filter Policy cflowd Sampling on page 423
  - → Filter Policy Management on page 424
  - → Match-list for Filter Policies on page 425
  - → Embedded Filters on page 429
  - → System-level IPv4/IPv6 Line Card Filter Policy on page 431
  - → Network-port VPRN Filter Policy on page 432
  - → ISID MAC Filters on page 432
  - → VID MAC filters on page 433
  - → Redirect Policies on page 437
  - → HTTP-redirect (Captive Portal) on page 440
  - → Filter Policies and Dynamic, Policy-Driven Interfaces on page 442

ACL Filter Policy Overview

ACL Filter policies, also referred to as Access Control Lists (ACLs) or filters for short, are sets of ordered rule entries specifying packet match criteria and actions to be performed to a packet upon a match. Filter policies are created with a unique filter ID, but each filter can also have a unique filter name configured once the filter policy has been created. Either filter ID or filter name can be used throughout the system to manage filter policies and assign them to interfaces.

There are three main types of filter policies: IPv4, IPv6, and MAC filter policies. Additionally MAC filter policies support three sub-types: (**configure filter mac-filter type** {**normal** | **isid** | **vid**}). These sub-types allow operators to configure different L2 match criteria for a L2 MAC filter.

There are different kinds of filter policies as defined by the filter policy **scope**:

- An **exclusive** filter allows defining policy rules explicitly for a single interface. An exclusive filter allows highest-level of customization but uses most resources, since each exclusive filter consumes H/W resources on line cards the interface exists.
- A **template** filter allows usage of identical set of policy rules across multiple interfaces. Template filters use a single set of resources per line card, regardless of how many interfaces use a given template filter policy on that line card. Template filter policies used on access interfaces, consume resources on line cards only if at least one access interface for a given template filter policy is configured on a given line card.
- An **embedded** filter allows defining common set of policy rules that can then be used (embedded) by other exclusive or template filters in the system. This allows optimized management of filter policies.
- A **system** filter policy allows defining common set of policy rules that can then be activated within other exclusive/template filters. A system filter policy is intended mainly for system-level blacklisting rules but can be used for other applications as well. This allows optimized management of common rules (similarly to embedded filters); however, active system filter policy entries are not duplicated inside each policy that actives the system policy (as is the case when embedding is used). The active system policy is downloaded once to line cards, and activating filter policies are chained to it.

Once created, filter policies must then be associated with interfaces/services/subscribers or with other filter policies (if the created policy cannot be directly deployed on interface/services/subscriber), so the incoming/outgoing traffic can be subjected to filter rules. Filter policies are associated with interfaces/services/subscribers separately in ingress and in egress direction. A policy deployed on ingress and egress direction can be same or different. In general, it is recommended to use different filter policies per-ingress and per-egress directions and to use different filter policies per service type, since filter policies support different match criteria and different actions for different direction/service contexts. A filter policy is applied to a packet in the ascending rule entry order. When a packet matches all the parameters specified in a filter entry's match criteria, the system takes the action defined for that entry. If a packet does not match the

entry parameters, the packet is compared to the next higher numerical filter entry rule and so on. If the packet does not match any of the entries, the system executes the **default-action** specified in the filter policy: **drop** or **forward**.

For Layer 2, either an IPv4/IPv6, and MAC filter policy can be applied. For Layer 3 and network interfaces, an IPv4/IPv6 policy can be applied. For r-VPLS service, a L2 filter policy can be applied to L2 forwarded traffic and L3 filter policy can be applied to L3 routed traffic. For dual stack interfaces, if both IPv4 and IPv6 filter policies are configured, the policy applied will be based on the outer IP header of the packet. Note that non-IP packets are not hitting an IP filter policy, so the default action in the IP filter policy will not apply to these packets.

ACL Filter Policy Commands

The command **action** (with all types and related parameters defining) used to define an action to be performed on a packet matching IPv4/IPv6/MAC ACL policy entry has been deprecated and replaced by a new **action** command that allow operator to enter a new CLI context under which individual actions can be selected using **drop**, **forward**, **gtp-local-breakout**, **http-redirect**, **nat**, reassemble action type commands and their parameters as applicable to a given action type and a given filter type.

Operational impact of the above-described restructuring:

- Since all command and parameter names were preserved, any ACL configuration prior to Release 13.0 R4 remains valid and results in same configuration result for all but the below highlighted case
- Prior to release 13R4 executing an action command without any parameters

  → config>filter>ip-filter>entry>action

  → config>filter>ipv6-filter>entry>action

  → config>filter>mac-filter>entry>action

  would result in "**action drop**" configuration (implicit **action drop** configuration). After an upgrade to release 13.0 R4, this functionality is no longer supported and **action drop** must be explicitly specified. An operator must add drop keyword to any existing manually edited CLI configuration files that do not explicitly specify **action drop**. Note that the system would always save a configuration with implicit action drop defined as explicit **action drop**.
- Starting with release 13.0 R4, **admin save** and **info** commands will save/display filter entry action configuration in a multi-line format (as illustrated in Table 9).
- Note that CLI configuration continues to accept a single line format to specify an action with its type and related parameters.

Table 9: Display Filter Entry Action

Command prior to Release 13.0R4	Command in Release 13.0R4
configure filter {ip-filter | ipv6-filter | mac-filter} entry action drop [optional parameters]	configure filter {ip-filter | ipv6-filter | mac-filter} entry action drop [optional parameters]
configure filter {ip-filter | ipv6-filter | mac-filter} entry action forward [optional parameters]	configure filter {ip-filter | ipv6-filter | mac-filter} entry action forward [optional parameters]
configure filter {ip-filter | ipv6-filter | mac-filter} entry action http-redirect [optional parameters]	configure filter {ip-filter | ipv6-filter | mac-filter} entry action http-redirect [optional parameters]
configure filter {ip-filter | ipv6-filter} entry action nat [optional parameters]	configure filter {ip-filter | ipv6-filter} entry action nat [optional parameters]
configure filter ip-filter entry action gtp-local-breakout	configure filter ip-filter entry action gtp-local-breakout
configure filter ip-filter entry action reassemble	configure filter ip-filter entry action reassemble

Filter Policy Basics

The following subsections define main functionality supported by filter policies.

Filter Policy Packet Match Criteria

This section defines packet match criteria supported on SROS-based routers/switches for IPv4, IPv6 and MAC filters. Types of criteria supported depends on the hardware platform and filter direction, please see your Alcatel-Lucent representative for further details.

General notes:

- If multiple unique match criteria are specified in a single filter policy entry, all criteria must be met in order for the packet to be considered a match against that filter policy entry (logical AND).
- Any match criteria not explicitly defined is ignored during match.
- An ACL filter policy entry with match criteria defined but no action configured, is considered incomplete and inactive (an entry is not downloaded to the line card). A filter policy must have at least single entry active for the policy to be considered active.
- An ACL filter entry with no match conditions defined matches all packets.
- Because an ACL filter policy is an order list, entries should be configured (numbered) from the most explicit to the least explicit.

IPv4/IPv6 Filter Policy Entry Match Criteria

The below lists IPv4 and IPv6 match criteria supported by SROS routers/switches. The criteria are evaluated against outer IPv4/IPv6 header and a L4 header that follows (if applicable). Support for a given match criteria may depend on H/W and/or filter direction as per below description. It is recommended not to configure a filter in a direction or on a H/W where a given match condition is not supported as this may lead to undesired behavior. Some match criteria may be grouped in match lists and may be auto-generated based on router configuration – see Advanced Filter Policy topics for more details.

Basic L3 match criteria:

- **dscp** — Match for the specified DSCP value against the Differentiated Services Code Point/Traffic Class field in the IPv4/v6 packet header.

- **src-ip/dst-ip** — Match for the specified source/destination IPv4/IPv6 address-prefix against the source/destination IPv4/IPv6 address field in the IPv4/IPv6 packet header. Operator can optionally configure a mask to be used in a match.
- **flow-label** — Match for the specified flow label against the Flow label field in IPv6 packet . Operator can optionally configure a mask to be used in a match. Supported for ingress filters only. Requires minimum chassis mode C.

Conditional action match criteria:

- **packet-length** — Match for the specified packet-length value/range against the Total Length field in IPv4 packet header or Payload Length field in IPv6 packet header. This match condition is supported for drop action only and is part of action evaluation – i.e. after packet is determined to match the entry based on other match criteria configured. Packets that match all match criteria for a given filter policy entry are dropped if the packet-length match criterion is met and forwarded if the packet match criterion is not met. When a filter entry with a packet-length condition is used as a mirror source, only forwarded packets are mirrored. Supported for ingress filters only. Requires minimum FP-2 based line cards. The packet-length match condition is always true if a filter is configured on egress or on an older H/W.
- **TTL** — Match for the specified TTL value/range against the Total Length field in IPv4 packet header . This match condition is supported for drop action only and is part of action evaluation – i.e. after packet is determined to match the entry based on other match criteria configured. Packets that match all match criteria for a given filter policy entry are dropped if the TTL match criterion is met and forwarded if the TTLmatch criterion is not met. When a filter entry with a TTL condition is used as a mirror source, only forwarded packets are mirrored. When a filter entry with a TTL condition is used in cflowd processing, the TTL condition is ignored for cflowd processing. Supported for ingress filters only and requires minimum FP-2 based line cards. The TTL match condition is always true if a filter is configured on egress or on an older H/W.

Fragmentation match criteria:

- **fragment** — Enable fragmentation support in filter policy match. For IPv4, match against MF bit or Fragment Offset field to determine whether the packet is a fragment or not. For IPv6, match against Next Header Field for Fragment Extension Header value to determine whether the packet is a fragment or not. Up to 6 extension headers are matched against to find Fragmentation Extension Header.

  Additional, match against whether the fragment is an initial fragment or non-initial fragment is also supported for IPv6 filters.

  IPv4 match fragment criteria are supported on both ingress and egress. IPv6 match fragment criteria are supported on ingress only and require minimum FP-2 based line cards.

IPv4 options match criteria:

- **ip-option** — Match for the specified option value in the first option of the IPv4 packet. Operator can optionally configure a mask to be used in a match.
- **option-present** — Match for the presence or absence of the IP options in the IPv4 packet. Padding and EOOL are also considered as IP options. Up to 6 IP options are matched against.
- **multiple-options** — Match for the presence of multiple IP options in the IPv4 packet.
- **src-route-option** — Match for the presence of IP Option 3 or 9 (Loose or Strict Source Route) in the first 3 IP Options of the IPv4 packet. A packet will also match this rule if the packet has more than 3 IP Options.

IPv6 next-header match criteria (see also Upper-layer protocol match next-header description below):

- **ah-ext-header** — Match for presence/absence of the Authentication Header extension header in the IPv6 packet. This match criterion is supported on ingress only and requires minimum FP-2 based line cards. Up to 6 extension headers are matched against.
- **esp-ext-header** — Match for presence/absence of the Encapsulating Security Payload extension header in the IPv6 packet. This match criterion is supported on ingress only and requires minimum FP-2 based line cards. Up to 6 extension headers are matched against.
- **hop-by-hop-opt** — Match for the presence/absence of Hop-by-hop options extension header in the IPv6 packet. This match criterion is supported on ingress only and requires minimum FP-2 based line cards. Up to 6 extension headers are matched against.
- **routing-type0** — Match for the presence/absence of Routing extension header type 0 in the IPv6 packet. This match criterion is supported on ingress only and requires minimum FP-2 based line cards. Up to 6 extension headers are matched against.

Upper-layer protocol match:

- **next-header** — Match for the specified upper layer protocol (for example, TCP, UDP, IGMPv6) against the Next Header field of the IPv6 packet header. "*" can be used to specify TCP or UDP upper-layer protocol match (Logical OR). Note: next-header matching allows also matching on presence of a subset of IPv6 extension headers. See CLI section for details on which extension header match is supported.
- **protocol** — Match for the specified protocol against the Protocol field in the IPv4 packet header (for example, TCP, UDP, IGMP) of the outer IPv4. "*" can be used to specify TCP or UDP upper-layer protocol match (Logical OR).
- **icmp-code** — Match for the specified value against the Code field of the ICMP/ICMPv6 header of the packet. This match is supported only for entries that also define protocol/ next-header match for "ICMP"/"ICMPv6" protocol.
- **icmp-type** — Match for the specified value against the Type field of the ICMP/ICMPv6 header of the packet. This match is supported only for entries that also define protocol/ next-header match for "ICMP"/"ICMPv6" protocol.

- **src-port/dst-port/port** – Match for the specified port value or port range against the Source Port Number/Destination Port Number of the UDP/TCP/SCTP packet header. An option to match either source or destination (Logical OR) using a single filter policy entry is supported by using a directionless “port” command. Source/destination match is supported only for entries that also define protocol/next-header match for “TCP”, “UDP”, “SCTP”, or “TCP or UDP” protocols. Note that a non-initial fragment will never match an entry with non-zero port criteria specified.
- **tcp-ack/tcp-syn** — Match for the TCP ACK/TCP SYNC flag presence/absence in the TCP header of the packet. This match is supported only for entries that also define protocol/next-header match for “TCP” protocol.

MAC Filter Policy Entry Match Criteria

The below lists MAC match criteria supported by SROS routers/switches for all types of MAC filters (normal, isid, and vid). The criteria are evaluated against the Ethernet header of the Ethernet frame. Support for a given match criteria may depend on H/W and/or filter direction as per below description. Match criterion is blocked if it is not supported by a specified frame-type or MAC filter sub-type. It is recommended not to configure a filter in a direction or on a H/W where a given match condition is not supported as this may lead to undesired behavior.

- **frame-type** — Entering the frame type allows the filter to match for a specific type of frame format. For example, configuring frame-type ethernet_II will match only Ethernet-II frames.
- **src-mac**— Entering the source MAC address allows the filter to search for matching a source MAC address frames. Operator can optionally configure a mask to be used in a match.
- **dst-mac**— Entering the destination MAC address allows the filter to search for matching destination MAC address frames. Operator can optionally configure a mask to be used in a match.
- **dot1p** — Entering an IEEE 802.1p value allows the filter to search for matching 802.1p frames. Operator can optionally configure a mask to be used in a match.
- **etype**— Entering an Ethertype value allows the filter to search for matching Ethernet II frames. The Ethernet type field is a two-byte field used to identify the protocol carried by the Ethernet frame.
- **ssap**— Entering an Ethernet 802.2 LLC SSAP value allows the filter to search for matching frames with a source access point on the network node designated in the source field of the packet. Operator can optionally configure a mask to be used in a match.
- **dsap**— Entering an Ethernet 802.2 LLC DSAP value allows the filter to search for matching frames with a destination access point on the network node designated in the destination field of the packet.. Operator can optionally configure a mask to be used in a match.
- **snap-oui**— Entering an Ethernet IEEE 802.3 LLC SNAP OUI allows the filter to search for matching frames with the specified the three-byte OUI field.
- **snap-pid**— Entering an Ethernet IEEE 802.3 LLC SNAP PID allows the filter to search for the matching frames with the specified two-byte protocol ID that follows the three-byte OUI field.
- **isid** — Entering an Ethernet IEEE 802.1ag ISID from the I-TAG value allows the filter to search for the matching Ethernet frames with the 24 bits ISID value from the PBB I-TAG. This match criterion is mutually exclusive with all the other match criteria under a particular mac-filter policy and is applicable to MAC filters of type isid only. The resulting mac-filter can only be applied on a BVPLS SAP or PW in the egress direction.

- **inner-tag/outer-tag** — Entering inner-tag/outer-tag VLAN ID values allows the filter to search for the matching Ethernet frames with the non-service delimiting tags as described In “VID MAC filters” subsection later-on this. This match criterion is mutually exclusive with all other match criteria under a particular mac-filter policy and is applicable to MAC filters of type vid only.

Filter Policy Actions

The following lists actions supported by ACL filter policies

- **drop** — This action allows operator to deny traffic to ingress/egress the system
- **forward** — This action allows operator to permit traffic to ingress/egress the system and be subject to regular processing
- **forward** "Policy-based Routing/Forwarding (PBR/PBF) action"— PBR/PBF actions allows operator to permit ingress traffic but change the regular routing/forwarding packet would be a subject to. The PBR/PBF is applicable to unicast traffic only. The following PBR/PBF actions are supported (See CLI section for command details):
  - → **egress-pbr** — enabling **egress-pbr** activates a PBR action on egress, while disabling **egress-pbr** activates a PBR action on ingress (default).

    The following subset of the below-defined PBR actions can be activated on egress: **redirect-policy**, **next-hop-router**, and **esi**.

    Egress PBR is supported in IPv4 and IPv6 filter policies for ESM only. Unicast traffic that is subject to slow-path processing on ingress (for example IPv4 packets with options or IPv6 packets with hop-by-hop extension header) will not match egress pbr entries. Filter logging, cflowd, and mirror source are mutually exclusive to configuring a filter entry with an egress PBR action. Configuring **pbr-down-action-override**, if supported with a given PBR ingress action type, is also supported when the action is an egress PBR action. Processing defined by **pbr-down-action-override** does not apply if the action is deployed in the wrong direction. If a packet matches a filter PBR entry and the entry is not activated for the direction in which the filter is deployed, **action forward** is executed. Egress PBR cannot be enabled in system filters.

    Egress PBR functionality requires chassis mode D.
  - → **esi** — forwards the incoming traffic using VXLAN tunnel resolved using EVPN MP BGP control plane to the first service chain function identified by ESI (L2) or ESI/SF-IP (L3). Supported with VPLS (L2) and IES/VPRN (L3) services. If the service function forwarding cannot be resolved, traffic matches an entry and action forward is executed.

    For VPLS, no cross service PBF is supported – i.e. the filter specifying ESI PBF entry must be deployed in the VPLS service where BGP EVPN control plane resolution takes place as configured for a given ESI PBF action. The functionality is supported in filter policies deployed on ingress VPLS interfaces. BUM traffic that matches a filter entry with ESI PBF will be unicast forwarded to the VTEP:VNI resolved through PBF forwarding.

For IES/VPRN, the outgoing R-VPLS interface can be in any VPRN service. The outgoing interface and VPRN service for BGP EVPN control plane resolution must again be configured as part of ESI PBR entry configuration. The functionality is supported in filter policies deployed on ingress IES/VPRN interfaces and in filter policies deployed on ingress and egress for ESM subscribers. Only uncast traffic is subject to ESI PBR, any other traffic matching a filter entry with L3 ESI action will be subject to **action forward**.

The functionality requires chassis mode D. When deployed in unsupported direction, traffic matching a filter policy ESI PBR/PBF entry will be subject to action forward.

→ **interface** — forwards the incoming traffic onto the specified IPv4 interface. Supported for ingress IPv4 filter policies in global routing table instance. If the configured interface is down or not of the supported type, traffic is dropped.

→ **lsp** — forwards the incoming traffic onto the specified LSP. Supports RSVP-TE LSPs (type static or dynamic only) or MPLS-TP LSPs. Supported for ingress IPv4/IPv6 filter policies only deployed on IES SAPs or network interfaces. If the configured LSP is down, traffic matches the entry and action forward is executed.

→ **next-hop** — changes the IP destination address used in routing from the address in the packet to the address configure in this PBR action. The operator can configure whether the next-hop IP address must be direct (local subnet only) or indirect (any IP). Supported for ingress IPv4/IPv6 filter policies only, deployed on L3 interfaces. If configured next-hop is not reachable, traffic is dropped and "ICMP destination unreachable" message is sent. For IPv6, requires minimum Chassis mode C.

→ **redirect-policy** — implements PBR **next-hop** or PBR **next-hop router** action with ability to select and prioritize multiple redirect targets and monitor the specified redirect targets so PBR action can be changed if the selected destination goes down. Supported for ingress IPv4 and IPv6 filter policies deployed on L3 interfaces only. See Redirect Policies in this chapter for more details.

→ **router** — changes the routing instance a packet is routed in from the upcoming interface's instance to the routing instance specified in the PBR action (supports both GRT and VPRN redirect). This action requires incoming interfaces to be on FP2 line cards or newer. It is supported for ingress IPv4/IPv6 filter policies deployed on L3

interfaces. The action can be combined with the **next-hop** action specifying direct/indirect IP/IPv6 next hop. Packets are dropped if they cannot be routed in the configured routing instance.

→ **sap** — forwards the incoming traffic onto the specified VPLS SAP. Supported for ingress IPv4/IPv6 and MAC filter policies deployed in VPLS service. The SAP traffic is to egress on must be in the same VPLS service as the incoming interface. If the configured SAP is down, traffic is dropped.

→ **sdp** — forwards the incoming traffic onto the specified VPLS SDP. Supported for ingress IPv4/IPv6 and MAC filter policies deployed in VPLS service. The SDP traffic is to egress on must be in the same VPLS service as the incoming interface. If the configured SDP is down, traffic is dropped.

- **forward** "isa action" — ISA processing actions allow operator to permit ingress traffic and send it for ISA processing as per specified isa action. The following isa actions are supported (see CLI section for command details):

  → **gtp-local-breakout** — forwards matching traffic to NAT instead of being GTP tunneled to the mobile operator's PGW or GGSN. The action applies to GTP-subscriber-hosts. If filter is deployed on other entities, action forward is applied. Supported for IPv4 ingress filter policies only. If ISAs performing NAT are down, traffic is dropped.

  → **nat** — forwards matching traffic for NAT. Supported for IPv4/IPv6 filter policies for L3 services in GRT or VPRN. If ISAs performing NAT are down, traffic is dropped. (see CLI for options)

  → **reassemble** — forwards matching packets to the reassembly function. Supported for IPv4 ingress filter policies only. If ISAs performing reassemble are down, traffic is dropped.

- **http-redirect** — implements HTTP redirect captive portal. HTTP GET is forwarded to CPM card for captive portal processing by router. See HTTP-redirect (Captive Portal) section for further details.

In addition to the above actions:

- An operator can select a **default-action** for a filter policy. The default action is executed on packets subjected to an active filter when none of the filter's active entries matches the packet. By default, filter policies have default action set to drop but operator can select a default action to be forward instead.
- An operator can override default action applied to packets matching a PBR/PBF entry when the PBR/PBF target is down using **pbr-down-action-override**. Supported options are to drop the packet, forward the packet, or apply the same action as configured for filter's **default-action**. The override is supported for the following PBR/PBF actions:

  → **forward esi**

  The following table defines default behavior for packets matching a PBR/PBF filter entry when the target is down:

Table 10: Display Filter Entry Action

PBR/PBF action	Default behavior when down
forward esi (any type)	Forward
forward lsp	Forward
forward next-hop (any type)	Drop
forward redirect-policy	Forward when redirect policy is shutdown
forward redirect-policy	Forward - when destination tests are enabled and the best destination is not reachable
forward redirect-policy	Drop - when destination tests are not enabled and the best destination is not reachable
forward sap	Drop
forward sdp	Drop
forward router	Drop

Filter Policy Statistics

Filter policies support per-entry, packet/Byte match statistics. The cumulative matched packet/Byte counters are available per ingress and per egress direction. Every packet arriving on an interface/service/subscriber using a filter policy increments ingress or egress (as applicable) matched packet/Byte count for a filter entry the packet matches (if any) on the line card the packet ingresses/egresses. For each policy, the counters for all entries are collected from all line cards, summed up and made available to an operator.

Starting with SROS Release 11.0 R4, filter policies applied on access interfaces are downloaded only when active and only to line cards that have interfaces associated with those filter policies. If a filter policy is not downloaded to any line card, the statistics show 0 (zero). If a filter policy is being removed from any of the line cards the policy is currently downloaded to (as result of association change or when a filter becomes inactive), the statistics for the filter are reset to 0 (zero). Downloading a filter policy to a new line card keeps incrementing existing statistics.

Starting with SR-OS Release 13.0R4, filter policies support bulk requests CPM cache for policy interfaces created entries. The cache is periodically refreshed through a background collection of counters from hardware. The counters are also refreshed when the ACL entry corresponding to the cache entry has statistics read from hardware through any direct-read from hardware mechanism. If a cache entry represents an entry for an ACL filter policy not downloaded to any line cards, the cache returns values of 0 (zero). If a cache entry represents an ACL filter entry that was removed

from a line card since the previous refresh, the current refresh will reload the cache with the most recent values from hardware. The cache has to be rebuilt on a High Availability (HA) switchover, thus initial statistics requests after an HA switchover may require reads from hardware.

Operational Notes:

- Two consecutive bulk requests for one entry will return the same values if the cache has not been refreshed between the two requests. The refresh interval is platform/release dependent. Please contact your Alcatel-Lucent representative for further details.
- The cache is currently used only for Open Flow statistics retrieval. Please see " Hybrid OpenFlow Switch" section for more details.

Filter Policy Logging

SROS supports logging of the information from the packets that match given filter policy. Logging is configurable per filter policy entry by specifying pre-configured filter log (**config filter log**). A filter log can be applied to ACL filters and CPM hardware filters. Operator can configure multiple filter logs and specify: memory allocated to a filter log destination, syslog id for filter log destination, filter logging summarization, and wrap-around behavior.

Notes related to filter log summarization:

- The implementation of the feature applies to filter logs with destination syslog.
- Summarization logging is the collection and summarization of log messages for 1 specific log-id within a period of time.
- The summarization interval is 100 seconds.
- Upon activation of a summary, a mini-table with src/dst-address and count is created for each type (IP/IPv6/MAC).
- Every received log packet (due to filter hit) is examined for source or destination address.
- If the log packet (source/destination address) matches a source/destination address entry in the mini-table a packet received previously), the summary counter of the matching address is incremented.
- If source or destination address of the log messages does not match an entry already present in the table, the source/destination address is stored in a free entry in the mini-table.
- In case the mini-table has no more free entries, only total counter is incremented.
- At expiry of the summarization interval, the mini-table for each type is flushed to the syslog destination.

Filter Policy cflowd Sampling

Filter policies can be used to control how cflowd sampling is performed on an IP interface. If an IP interface has cflowd sampling enabled, an operator can exclude some flows for interface sampling by configuring filter policy rules that match the flows and by disabling interface sampling as part of the filter policy entry configurations (**interface-disable-sample**). If an IP interface has cflowd sampling disabled, an operator can enable cflowd sampling on a subset of flows by configuring filter policy rules that match the flows and by enabling cflowd sampling as part of the filter policy entry configurations (**filter-sample**).

Note that the above cflowd filter sampling behavior is exclusively driven by match criteria: The sampling logic applies regardless of whether an action was executed or not (including evaluation of conditional action match criteriam, for example, **packet-length** or **ttl**).

Filter Policy Management

Modifying Existing Filter Policy

There are several ways to modify an existing filter policy. A filter policy can be modified through configuration change or can have entries populated through dynamic, policy-controlled dynamic interfaces like Radius or OpenFlow or Flowspec or Gx for example. Although in general, the SROS ensures filter resources exist before a filter can be modified, because of a dynamic nature of the policy-controlled interfaces, a configuration that was accepted may not be applied in H/W due to lack of resources. When that happens, an error is raised.

A filter policy can be modified directly – by changing/adding/deleting the existing entry in that filter policy or indirectly. Examples of indirect change to filter policy include, among others, changing embedded filter entry this policy embeds (see Embedded filters section), changing redirect policy this filter policy uses.

Finally, a filter policy deployed on a given interface can be changed by changing the policy the interface is associated with.

All of the above changes can be done in service. Note that a filter policy that is associated with service/interface cannot be deleted unless all associations are removed first.

For a large (complex) filter policy change, it may take a few seconds to load and initiate the filter policy configuration. It should also be noted, that filter policy changes are downloaded to line cards immediately, therefore operators should use filter policy copy or transactional CLI to ensure partial policy change is not activated.

Filter Policy Copy and Renumbering

To assist operators in filter policy management, SROS supports entry copy and entry renumbering operations.

Filter **copy** allows operators to perform bulk operations on filter policies by copying one filter's entries to another filter. Either all entries or a specified entry of the source filter can be selected for copy. When entries are copied, entry order is preserved unless destination filter's entry ID is selected (applicable to single entry copy). The filter copy allows overwrite of the existing entries in the destination filter by specifying "overwrite" option during the copy command. Filter copy can be used, for example, when creating new policies from existing policies or when modifying an existing filter policy (an existing source policy is copied to a new destination policy, the new destination policy is modified, then the new destinations policy is copied back the source policy with overwrite specified).

Entry renumbering allows operator to change relative order of a filter policy entry by changing the entry Id. Entry renumbering can also be used to move 2 entries closer together or further apart, thus creating additional entry space for new entries.

Filter Policy Advanced Topics

Match-list for Filter Policies

Figure 15 depicts an approach to implement logical OR on a list of matching criterion (IPv4 address prefixes in this example) in one or more filter policies prior to introduction of match list.

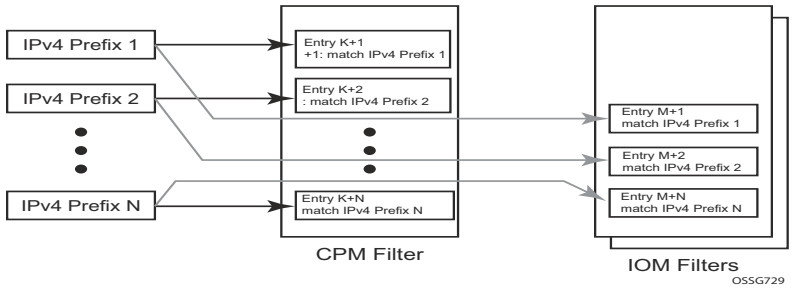

Figure 15: IOM/CPM Filter Policy using Individual Address Prefixes

An operator has to create one entry for each address prefix to execute a common action. Each entry defines a match on a unique address prefix from the list plus any other additional match criteria and the common action. If the same set of address prefixes needs to be used in another IOM or CPM filter policy, an operator again needs to create one entry for each address prefix of the list in those filter policies. Same procedure applies (not shown above) if another action needs to be performed on the list of the addresses within the same filter policy (when for example specifying different additional match criteria). This process can introduce large operational overhead, especially when a list contains many elements or/and needs to be reused multiple times across one or more filter policies.

Match list for CPM and IOM filter policies are introduced to eliminate above operational complexity by simplifying the IOM and CPM filter policy management on a list of a match criterion. Instead of defining multiple filter entries in any given filter, an operator can now group same type of the matching criteria into a single filter match list, and then use that list as a match criterion value, thus requiring only single filter policy entry per each unique action. The same match list can be used in one or more IOM filter policies as well as CPM filter policies.

The match lists further simplify management and deployment of the policy changes. A change in a match-list content is automatically propagated across all policies employing that list in their match criteria, thus only a single configuration change is required to trigger policy changes when a list is used by multiple entries in one or more filter policies.

Figure 16 depicts how the IOM/CPM filter policy illustrated at the top of this section changes with a filter match list usage (using IPv4 address prefix list in this example).

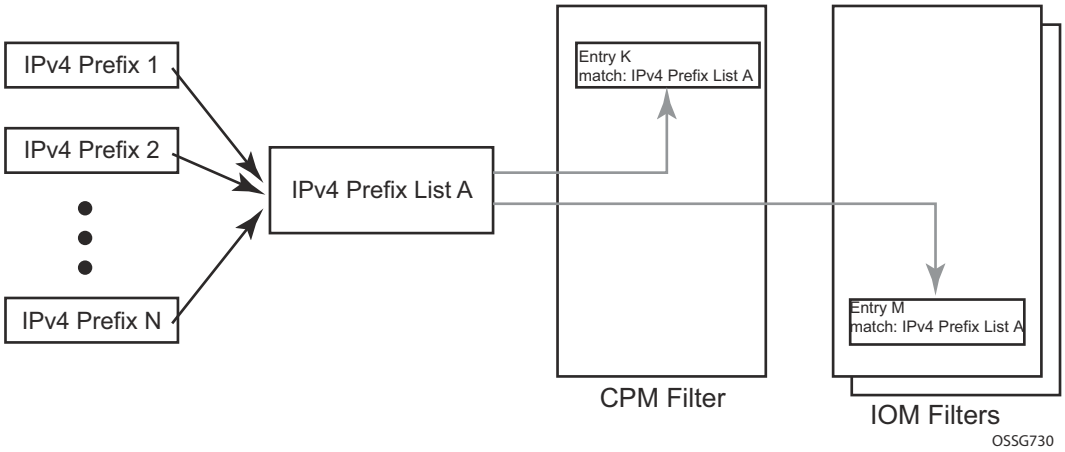

Figure 16: IOM/CPM Filter Policy Using an Address Prefix Match List

Note: The hardware resource usage does not change whether filter match lists are used or whether operator creates multiple entries (each per one element of the list): however, a careful consideration must be given to how the lists are used to ensure only desired match permutations are created in a filter policy entry (especially when other matching criteria that are also lists or ranges are specified in the same entry). The system verifies that a new list element, for example, an IP address prefix, cannot be added to a given list or a list cannot be used by a new filter policy unless resources exist in hardware to implement the required filter policy (ies) that reference that list. If that is not the case, addition of a new element to the list or use of the list by another policy will fail.

Some use cases like those driven by dynamic policy changes, may result in acceptance of filter policy configuration changes that cannot be programmed in hardware because of the resource exhaustion. If that is the case, when attempting to program a filter entry that uses a match list(s), the operation will fail, the entry will be not programmed, and a notification of that failure will be provided to an operator.

Please refer to SROS Release Notes for what objects can be grouped into a filter match list for IOM and CPM filter policies.

Auto-generation of Filter-policy Address Prefix Match Lists

It is often desired to automatically update a filter policy when the configuration on a router changes. To allow such a touch-less filter policy management, SROS allows auto-generation of address prefixes for IPv4 or IPv6 address prefix match lists based on operator-configured criteria. When the configuration on a router changes, the match lists address prefixes are automatically updated and, in-turn, all filter policies (CPM or IOM) that use these match lists are automatically updated.

When using auto-generation of address prefixes inside an address prefix match list operators can:

- Specify one or more *regex* expression matches against SROS router configuration per list.
- Specify wildcard matches by specifying *regex* wildcard match expression (".*").
- Mix auto-generated entries with statically configured entries within a match list.

The following additional rules apply to auto-generated entries:

- Operational and administrative states of a given router configuration are ignored when auto-generating address prefixes.
- Duplicates are not removed when populated by different auto-generation matches and static configuration.
- A configuration will fail if auto-generation of address prefix would result in filer policy resource exhaustion on a filter entry, system, or line-card level.

NOTE: See Release notes and CLI section for details on what configuration supports address prefix list auto-generation.

The following may apply to this feature:

If filter policy resources are not available for newly auto-generated address prefixes when a BGP configuration changes, new address-prefixes will not be added to impacted match lists or filter policies as applicable. An operator must free resources and change filter policy configuration or must change BGP configuration to recover from this failure.

Embedded Filters

When a large number of standard filter policies are configured in a system, a set of policies will often contain one or more common blocks of entries that define, for example, system-wide and/or service-wide security rules. Prior to introduction of the embedded filters, such common rules would have to be configured separately in each exclusive/template policy.

To simplify management of such common rules across multiple filter policies, operator can now use embedded filter policies. An embedded filter policy is a special type of a filter policy that cannot be deployed directly but instead is used to define a common filter policy rules that are then included in (embedded by) other filter policies in the system. Thanks to embedding, a common set of rules can now be defined and changed in a single place but deployed across multiple filter policies. The following main rules apply when embedding an embedded filter policy:

1. An operator can explicitly define an offset at which to embed a given embedded filter into a given embedding filter—the embedded filter entry number X becomes an entry (X + offset) in the embedding filter.
2. An exclusive/template filter policy may embed multiple embedded filter policies as long as the embedded entries do not overlap.
3. A single embedded filter policy may be embedded in many exclusive/template filter policies.
4. When embedding an embedded filter, an operator may wish to change or deactivate an embedded filter policy entry in one of the embedding filter, thus allowing for customizing of the common embedded filter policy rules by the embedding filter. This can be achieved by either defining an entry in the embedding filter that will match ahead of the embedded filter entry or by overwriting the embedded filter entry in the embedding filter.

   For example: If embedded filter 99 has entry 20 that drops packets that match IP source address **src_address**, and filter 200 embeds filter 99 at offset 100, then to *deactivate* the embedded entry 20, an operator could define an entry 120 (embedded entry number 20 + offset 100) in filter policy 200, that has the same match criteria and has either no action defined (this will deactivate the embedded entry and allow continued evaluation of filter policy 200), or has action forward defined (packets will match the new entry and will be forwarded instead of dropped, evaluation of filter policy 200 will stop).
5. Any embedded policy rule edits are automatically applied to all filter policies that embed that embedded filter policy.
6. The system verifies whether system and h/w resources exist when a new embedded filter policy is created, changed or embedded. If resources are not available, the configuration is rejected. In rare cases, filter policy resource check may pass but filter policy can still fail to load due to a resource exhaustion on a line card (for example when other filter policy entries are dynamically configured by applications like RADIUS in parallel). If that is the case, the embedded filter policy configured will be de-activated (configuration will be changed from **activate** to **inactivate**).

7. An embedded filter is never embedded partially into an exclusive/template filter; that is, resources must exist to embed all embedded filter entries in a given exclusive/template filter. Although a partial embedding into a single filter will not take place, an embedded filter may be embedded only in a subset of embedding filters (only those where there are sufficient resources available).

Figure 17 shows implementation of embedded filter policy using IPv4 ACL filter policy example with an embedded filter 10 being used to define common filter rules that are then embedded into filter 1 and 20 (with filter 20 overwriting rule at offset 50):

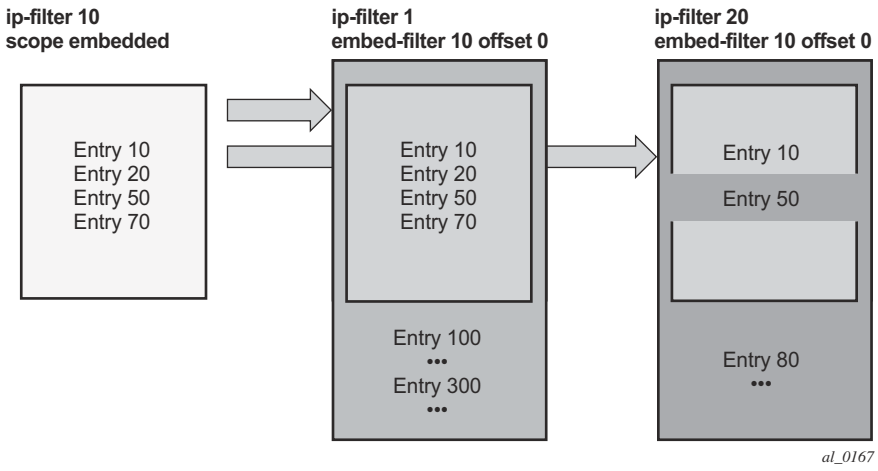

Figure 17: Embedded Filter Policy

NOTE: Embedded filter policies are supported for line card IP(v4) and IPv6 filter policies only.

System-level IPv4/IPv6 Line Card Filter Policy

A system filter policy allows the definition of a common set of policy rules that can then be activated within other exclusive/template filters. IPv4/IPv6 system filter policies supports all IPv4/IPv6 filter policy match rules and actions respectively but system policy entries cannot be the sources of mirroring.

System filter policy cannot be used directly; the active system policy is deployed by activating it within any IPv4 or IPv6 exclusive/template filter policy (chaining the system policy and a given interface policy). When an IPv4/IPv6 filter policy is chained to the active IPv4/IPv6 system filter, system filter rules are evaluated first before any rules of the chaining filter are evaluated (i.e. chaining filter's rules are only matched against if no system filter match took place).

A system filter policy is intended mainly for system-level blacklisting rules, thus it is recommended to use system policies with drop/forward actions. Other actions like, for example, PBR actions, or redirect to ISAs should not be used unless the system filter policy is activated only in filters used by services that support such action. The “nat” action is not supported and should not be configured. Failure to observe these restrictions can lead to undesired behavior as system filter actions are not verified against the services the chaining filters are deployed for.

System filter policies can be populated using CLI/SNMP/Netconf management interfaces and Openflow policy interface. System filter policy entries cannot be populated using flowspec, Radius, or Gx.

System filter policy scale is identical to a corresponding IPv4 or IPv6 filter policy scale. System filter policy consumes single set of H/W resources on each line card as soon as it is activated, regardless of how many IPv4/IPv6 filters chain to that system policy. This optimizes resource allocation when multiple filter policies activate a given system policy.

System filter policy requires chassis mode D.

An example (IPv4) configuration is shown below:

```
*A:vm1>config>filter#
# Configure system-policy
        ip-filter 1 create
            scope system
            entry 5 create
                match protocol *
                    fragment true
                exit
                action drop
            exit
        exit
# Activate it
        system-filter
            ip 1
        exit
# Use it in another filter:
```

```
ip-filter 10 create
    chain-to-system-filter
    filter-name "test-name"
    embed-filter open-flow "test" offset 100
    exit
exit
```

Network-port VPRN Filter Policy

Network-port L3 service-aware filter feature allows operators to deploy VPRN service aware ingress filtering on network ports. A single ingress filter of scope template can each be defined for IPv4 and for IPv6 against a VPRN service. The filter applies to all unicast traffic arriving on auto-bind and explicit-spoke network interfaces for that service. The network interface can be either Inter-AS, or Intra-AS. The filter does not apply to traffic arriving on access interfaces (SAP, spoke-sdp, network-ingress (CsC), rVPLS, eVPN).

The same filter can be used on access interfaces of the given VPRN, can embed other filters (including OpenFlow), can be chained to a system filter, and can be used by other L2 or L3 services.

The filter is deployed on all line cards (chassis network mode D is required). There are no limitations related to filter match/action criteria or embedding. The filter is programmed on line cards against ILM entries for this service. All label-types are supported. If an ILM entry has a filter index programmed, that filter is used when the ILM is used in packet forwarding; otherwise, no filter is used on the service traffic.

Caveats:

- Network port L3 service-aware filters do not support flowspec and LI (cannot use filter inside LI infrastructure nor have LI sources within the VPRN filter).

ISID MAC Filters

ISID filters are a type of MAC filters that allows filtering based on the ISID values rather than L2 criteria used by MAC filters of type "**normal**" or "**vid**". ISID filters can be deployed on iVPLS PBB SAPs and ePipe PBB SAPs in the following scenarios:

The MMRP usage of the mrp-policy ensures automatically that traffic using Group BMAC is not flooded between domains. However; there could be a small transitory periods when traffic originated from PBB BEB with unicast BMAC destination may be flooded in the BVPLS context as unknown unicast in the BVPLS context for both IVPLS and PBB Epipe. To restrict distribution of this traffic for local PBB services ISID filters can be deployed. The mac-filter configured with

ISID match criterion can be applied to the same interconnect endpoint(s), BVPLS SAP or PW, as the mrp-policy to restrict the egress transmission any type of frames that contain a local ISID. The ISID filters will be applied as required on a per B-SAP or B-PW basis just in the egress direction.

The ISID match criteria are exclusive with any other criteria under mac-filter. A new mac-filter type attribute is defined to control the use of ISID match criteria and must be set to ISID to allow the use of ISID match criteria.

VID MAC filters

VID Filters are a type of MAC filters that extend the capability of current Ethernet Ports with null or default SAP tag configuration to match and take action on VID tags. Service delimiting tags (for example QinQ 1/1/1:10.20 or dot1q 1/1/1:10, where outer tag 10 and inner tags 20 are service delimiting) allow fine grain control of frame operations based on the VID tag. Service delimiting tags are exact match and are stripped from the frame as illustrated in Figure 18. Exact match or service delimiting Tags do not require VID filters. VID filters can only be used to match on frame tags that are after the service delimiting tags.

With VID Filters operators can choose to match VID tags for up to two tags on ingress or egress or both.

- The outer-tag is the first tag in the packet that is carried transparently through the service.
- The inner-tag is the second tag in the packet that is carried transparently through the service.

VID Filters add the capability to perform VID value filter policies on default tags (1/1/1:* or 1/1/1:x.*, or 1/1/1/:*.0), or null tags (1/1/1, 1/1/1:0 or 1/1/1:x.0). The matching is based on the port configuration and the SAP configuration.

In the industry the QinQ tags are often referred to as the C-VID (Customer VID) and S-VID (service VID). The terms outer tag and inner tag allow flexibility without having to refer to C-TAG and an S-TAG explicitly. The position of inner and outer tags is relative to the port configuration and SAP configuration. Matching of tags is allowed for up to the first two tags on a frame. Since service delimiting tags may be 0, 1 or 2 tags.

The meaning of inner and outer has been designed to be consistent for egress and ingress when the number of non service delimiting tags is consistent. Service 1 in Figure 18 shows a conversion from qinq to a single dot1q example where there is one non-service delimiting tag on ingress and egress. Service 2 shows a symmetric example with two non-service delimiting tags (plus and additional tag for illustration) to two non-service delimiting tags on egress. Service 3 illustrates single non-service delimiting tags on ingress and to two tags with one non-service delimiting tag on ingress and egress.

SAP-ingress QoS setting allows for MAC-criteria type VID which uses the VID filter matching capabilities QoS and VID Filters (moved to QoS guide) on page 313.

A VID filter entry can also be used as a debug or lawful intercept mirror source entry.

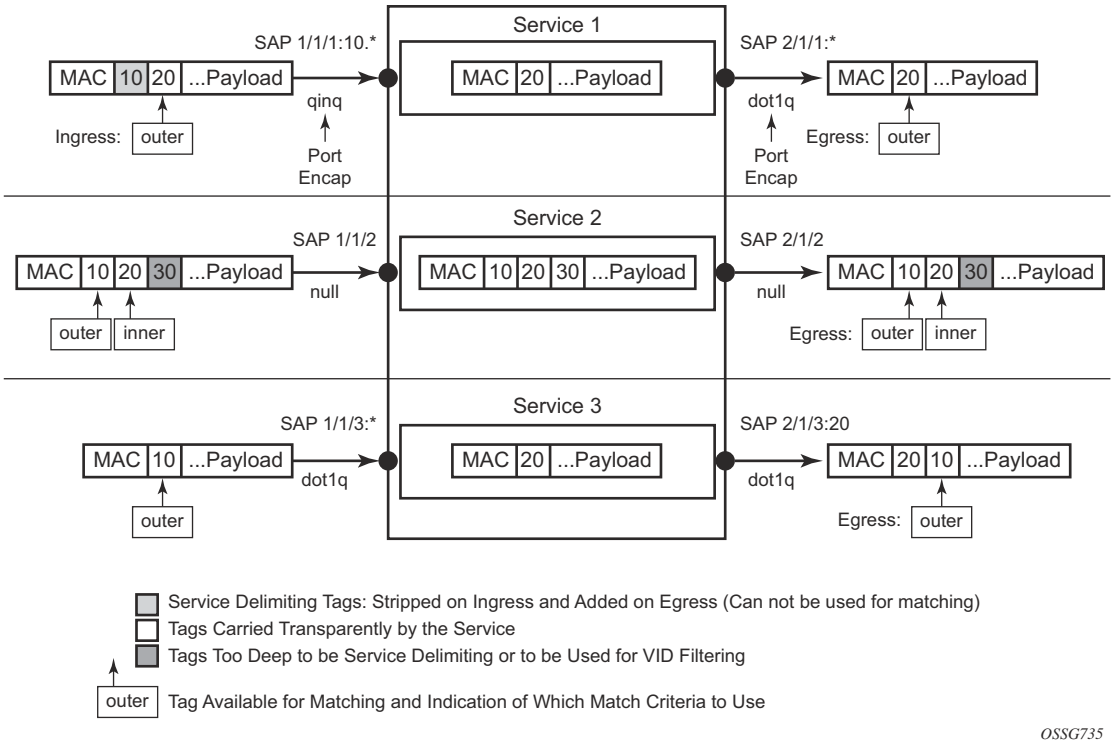

Figure 18: VID Filtering Examples

VID filters are available on Ethernet SAPs for Epipe, VPLS or I-VPLS including eth-tunnel and eth-ring services.

Arbitrary Bit Matching of VID Filters

In addition to matching an exact value, a VID filter mask allows masking any set of bits. The masking operation is ((value & vid-mask) = = (tag and vid-mask)). For example: A value of 6 and a mask of 7 would match all VIDs with the lower 3 bits set to 6. VID filters allow explicit matching of VIDs and matching of any bit pattern within the VID tag.

When using VID filters on SAPs only VID filters are allowed on this SAP. Filters of type normal and ISID are not allowed.

An additional check for the “0” VID tag may be required when using certain wild card operations. For example frames with no tags on null encapsulated ports will match a value of 0 in outer tag and inner tag because there are no tags in the frame for matching. If a zero tag is possible but not desired it can be explicitly filtered using exact match on “0” prior to testing other bits for “0”.

Note that **configure>system>ethernet>new-qinq-untagged-sap** is a special QinQ function for single tagged QinQ frames with a null second tag. Using this in combination with VID filters is not recommended. Note that the outer-tag is the only tag available for filtering on egress for frames arriving from MPLS SDPs or from PBB services even though additional tags may be carried transparently.

Port Group Configuration Example

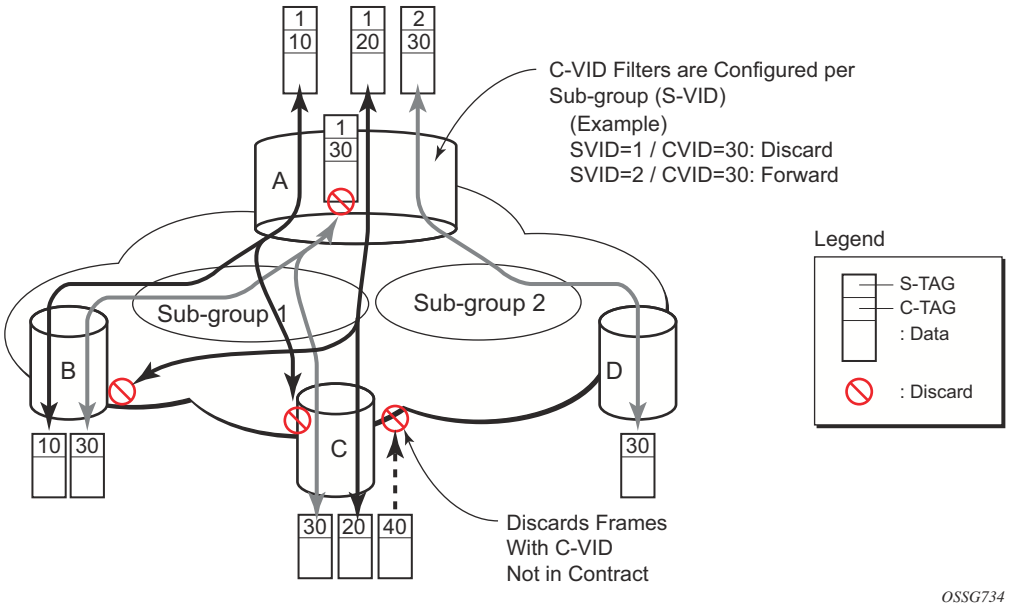

Figure 19: Port Groups

Figure 19 shows a customer use example where some VLANs are prevented from ingressing or egressing certain ports. In the example, port A sap 1/1/1:1.* would have a filter as shown below while port A sap 1/1/1:2.* would not.:

```
mac-filter 4 create
     default-action forward
            type vid
            entry 1 create
                match frame-type ethernet_II
                    outer-tag 30 4095
                exit
                action drop
            exit
        exit
```

Redirect Policies

SROS-based routers support configuring of IPv4 and IPv6 redirect policies. Redirect policies allow specifying multiple redirect target destinations and defining health check test methods used to validate the ability for a given destination to receive redirected traffic. This destination monitoring allows router to react to target destination failures. To specify IPv4 redirect policy, define all destinations to be IPv4. To specify IPv6 redirect policy define all destinations to be IPv6. IPv4 redirect policy can only be deployed in IP filter policies. IPv6 redirect policy can only be deployed in IPv6 filter policies.

Redirect policy supports the following destination tests:

- **ping test** – with configurable interval, drop-count, and time-out for the test
- **url-test** – with configurable URL to test, interval, drop-count, timeout, and configurable action (disable destination, lower or raise priority) based upon return error code
- **snmp-test** – with configurable OID and Community strings, interval, drop-count, timeout for the test, and configurable action (disable destination, lower or raise priority) based upon SNMP return value.
- **unicast-rt-test** – unicast routing reachability, supported only when router instance is configured for a given redirect policy. The test yields true if the route to the specified destination exists in RTM for the configured router instance.

Each destination is assigned an initial or base priority describing this destination's relative importance within the policy. The destination with the highest priority value is selected as most-preferred destination and programmed on line cards in filter policies using this redirect policy as an action. Note that only destinations that are not disabled by the programmed test (if configured) are considered when selecting the most-preferred destination.

In some deployments, it may not be desirable to switch from a currently active, most-preferred redirect-policy destination when a new more-preferred destination becomes available. To support such deployments, operators can enable the sticky destination functionality (**config>filter>redirect-policy>sticky-dest**). When enabled, the currently active destination remains active unless it goes down or an operator forces the switch using the **tools>perform>filter>redirect-policy>activate-best-dest** command. An optional sticky destination **hold-time-up** is available to delay programming the sticky destination in redirect-policy (transition from "action forward" to PBR action to the most-preferred destination). When the timer is enabled, the first destination that comes up will not be programmed and instead the timer is started. Once the timer expires, the most-preferred destination at that time will be programmed (which may be a different destination from the one that started the timer). Note the following:

- When manual switchover to most-preferred destination is executed as described above, the hold-time-up is stopped

- When the timer value is changed, the new value takes immediate effect and the timer is restarted with the new value (or expired if **no-hold-time-up** is configured)

Operational note: unicast-rt-test will fail when performed in the context of a VPRN routing instance when the destination is routable only through **grt-leak** functionality. **ping-test** is recommended in such cases.

Feature caveats:

- Redirect policy is supported for ingress IPv4 and IPv6 filter policies only.
- SNMP and URL tests are not supported for IPv6.
- Different platforms support different scale for redirect policies. Please contact your local Alcatel-Lucent representative to ensure the planned deployment does not exceed recommended scale.

Router Instance Support for Redirect Policies

There are two modes of deploying redirect policies on VPRN interfaces. The functionality supported depends on the configuration of redirect-policy router option with (**config**>**filter**>**redirect-policy-router**):

- Redirect policy with router option enabled (recommended):
  - → When a PBR destination is up, the PBR lookup is performed in the redirect policy's configured routing instance. When that instance differs from the incoming interface where the filter policy using the given redirect policy is deployed, the PBR action is equivalent to forward next-hop router filter policy action.
  - → When all PBR destinations are down (or a given hardware does not support action router), action forward is programmed and the PBR lookup is performed in the routing instance of the incoming interface where the filter policy using the given redirect policy is deployed.
  - → Any destination tests configured are executed in the routing context specified by the redirect-policy.
  - → Note that changing router configuration for a redirect policy, brings all destinations with a test configured down. The destinations are brought up once the test confirm reachability based on the new redirect policy router configuration.

- Redirect policy without router option disabled (**no router**) or with router options not supported (legacy):
  - → When a PBR destination is up, the PBR lookup is performed in the routing instance of the incoming interface where the filter policy using the given redirect policy is deployed.
  - → When all PBR destinations are down, action forward is programmed and the PBR lookup is performed in the routing instance of the incoming interface where the filter policy using the given redirect policy is deployed.
  - → Any destination tests configured are always executed in the "Base" router instance regardless of the router instance of the incoming interface where the filter policy using the given redirect policy is deployed.

Feature caveats:

- Only unicast-rt-test and ping-test are supported when router option is enabled.

HTTP-redirect (Captive Portal)

Web redirection policies can be configured on SR OS routers/switches. The http redirection action can block a customer's request from an intended recipient and force the customer to connect to the service's portal server. 255 unique entries with **http-redirect** are allowed.

Traffic Flow

The following example provides a brief scenario of a customer connection with web redirection.

1. The customer gets an IP address using DHCP (if the customer is trying to set a static IP he will be blocked by the anti-spoofing filter).
2. The customer tries to connect to a website.
3. The router intercepts the HTTP GET request and blocks it from the network
4. The router then sends the customer an HTTP 302 (service temporarily unavailable/moved). The target URL should then include the customer's IP and MAC addresses as part of the portal's URL.
5. The customer's web browser will then close the original connection and open a new connection to the web portal.
6. The web portal updates the ACL (directly or through SSC) to remove the redirection policy.
7. The customer connects to the original site.

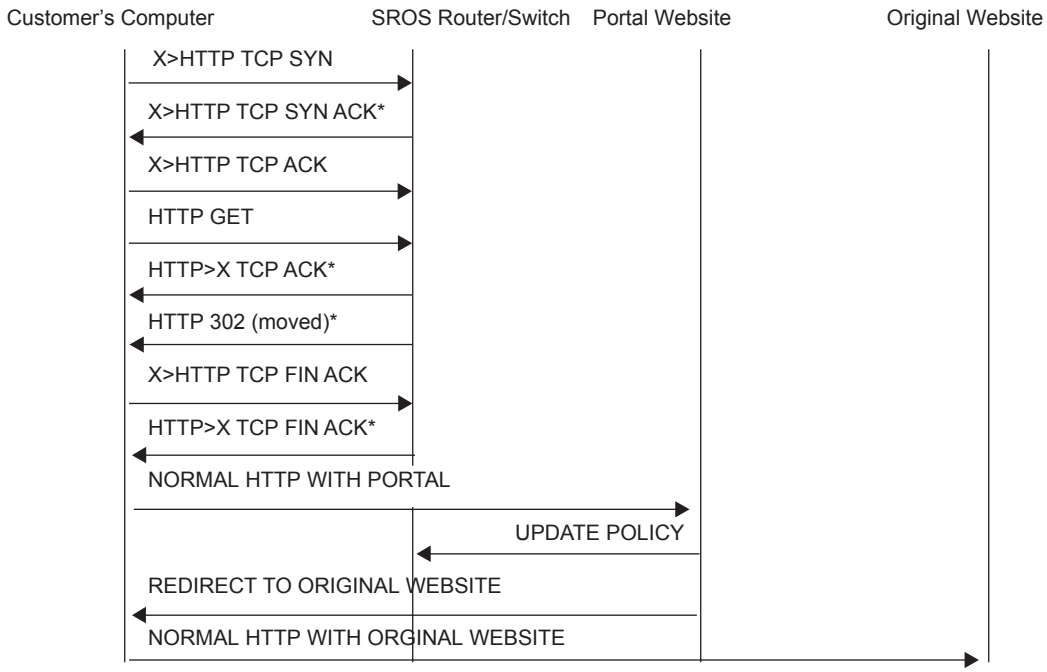

Figure 20: Web Redirect Traffic Flow

Starred entries (*) are items the router performs masquerading as the destination, regardless of the destination IP address or type of service.

The following displays information that can optionally be added as variables in the portal URL (http-redirect url):

- $IP – The customer's IP address.
- $MAC – The customer's MAC address.
- $URL – The original requested URL.
- $SAP – The customer's SAP.
- $SUB – The customer's subscriber identification string".
- $CID — A string that represents the circuit-id or interface-id of the subscriber host (hexadecimal format).
- $RID — A string that represents the remote-id of the subscriber host (hexadecimal format).
- $SAPDESC – A configurable string that represents the configured SAP description.

Note that the subscriber identification string is available only when used with subscriber management. Refer to the subscriber management section of the SROS Triple Play Guide and the SR OS Router Configuration Guide.

Since most web sites are accessed using the domain name the router allows either DNS queries or responds to DNS with the portal's IP address.

Filter Policies and Dynamic, Policy-Driven Interfaces

In addition to configuration interfaces like CLI/SNMP for example; filter policies can be modified and/or assigned to by dynamic, policy-driven interfaces. Example of such interfaces include: BGP flowspec, OpenFlow, Radius.

For BGP flowspec, system may auto-create internal filter policies (if an interface on which BGP flowspec is enabled does not have a filter policy assigned). Then upon receiving of a flowspec rule, system will attach flowspec filter rules at the end of the filter policy used on the interface up to the supported flowspec limit. Please see BGP flowspec for more information.

For Radius, operator can assign filter policies to a subscriber, and populate filter policies used by subscriber within a pre-configured block reserved for Radius filter entries. See TPSDA guide and filter RADIUS-related commands for more details.

For OpenFlow, embedded filter infrastructure is used to inject OpenFlow rules into an existing filter policy. Please see "Hybrid OpenFlow Switch" section for more details.

Policy-controlled auto-created filters are recreated on system reboot. Policy-controlled filter-entries are lost on system reboot and need to be reprogrammed.

Filter Policy-based ESM Service Chaining

In some deployments, operators may select to redirect ESM subscribers to Value Added Services (VAS). Various deployment models can be used but often subscribers are assigned to a particular residential tier-of-service, which also defines the VAS available to subscribers of the given tier. The subscribers are redirected to VAS based on tier-of-service rules but such an approach can be hard to manage when many VAS services/tiers of service are desired. Often the only way to identify a subscriber's traffic with a particular tier-of-service is to pre-allocate IP/IPv6 address pools to a given service tier and use those addresses in VAS PBR match criteria. This creates an application-services to network infrastructure dependency that can be hard to overcome, especially if fast and flexible application service delivery is desired.

Filter policy-based ESM service chaining removes ESM VAS steering to network infrastructure inter-dependency. An operator can configure per tier of service or per individual VAS service upstream and downstream service chaining rules without a need to define subscriber or

tier-of-service match conditions. Figure 21 shows a possible ACL model (embedded filters are used for VAS service chaining rules).

On the left in Figure 21, the per-tier-of-service ACL model is depicted. Each tier of service (Gold or Silver) has a dedicated embedded VAS filter ("Gold VAS", "Silver VAS") that contains all steering rules for all service chains applicable to the given tier. Each VAS filter is then embedded by the ACL filter used by a given tier. A subscriber is subject to VAS service chain rules based on the per-tier ACL assigned to that subscriber (for example, via Radius). If a new VAS rule needs to be added, an operator must program that rule in all applicable tiers. Upstream and downstream rules can be configured in a single filter (as shown) or can use dedicated ingress and egress filters.

On the right in Figure 21, the per-VAS-service ACL model is depicted. Each VAS has a dedicated embedded filter ("VAS 1", "VAS 2", "VAS 3") that contains all steering rules for all service chains applicable to that VAS service. A tier of service is then created by embedding multiple VAS-specific filters: Gold: VAS 1, VAS 2, VAS 3; Silver: VAS 1 and VAS 3. A subscriber is subject to VAS service chain rules based on the per-tier ACL assigned to that subscriber. If a new VAS rule needs to be added, an operator needs to program that rule in a single VAS-specific filter only. Again, upstream and downstream rules can be configured in a single filter (as shown) or can use dedicated ingress and egress filters.

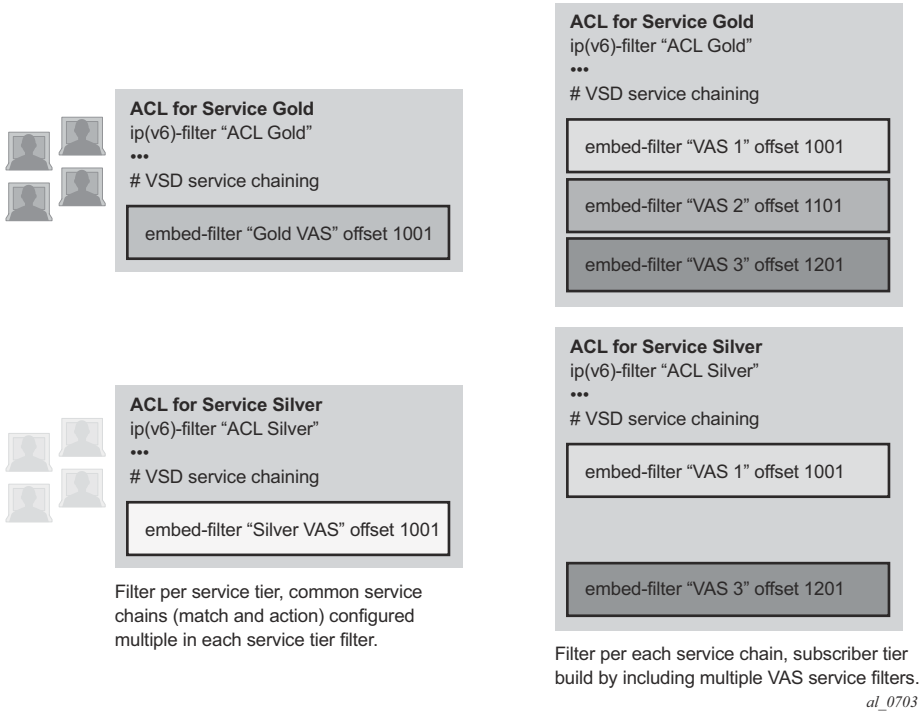

Figure 21: ACL filter modeling for ESM Service Chaining

Figure shows upstream VAS service chaining steering using filter policies. Upstream subscriber traffic entering Res-GW is subject to the subscriber's ingress ACL filter assigned to that subscriber by a policy server. If the ACL contains VAS steering rules, the VAS-rule-matching subscriber traffic is steered for VAS processing over a dedicated to-from-access VAS interface in the same or a different routing instance. After the VAS processing, the upstream traffic can be returned to Res-GW by a to-from-network interface (shown in Figure) or can be injected to WAN to be routed towards the final destination (not shown).

Upstream ESM ACL-policy based service chaining Figure 22 shows downstream VAS service chaining steering using filter policies. Downstream subscriber traffic entering Res-GW is forwarded to a subscriber-facing line card. On that card, the traffic is subject to the subscriber's egress ACL filter policy processing assigned to that subscriber by a policy server. If the ACL contains VAS steering rules, the VAS rule-matching subscriber's traffic is steered for VAS processing over a dedicated to-from-network VAS interface (in the same or a different routing instance). After the VAS processing, the downstream traffic must be returned to Res-GW via a "to-from-network" interface (shown in Figure 22) to ensure the traffic is not redirected to VAS again when the subscriber-facing line card processes that traffic.

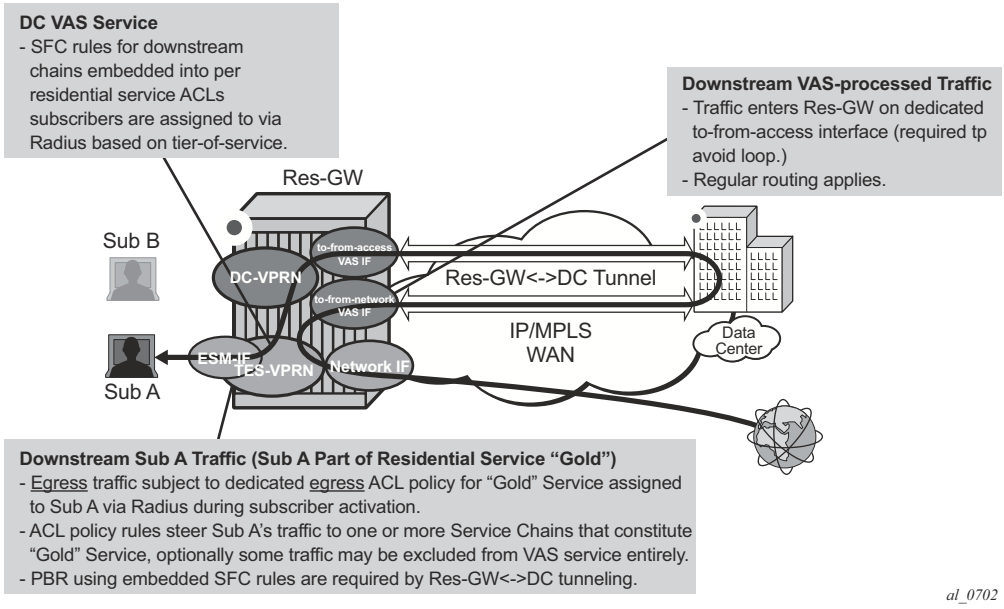

Figure 22: Downstream ESM ACL-policy based service chaining

The ESM filter policy-based service chaining allows operators to do the following:

- Steer upstream and downstream traffic per-subscriber with full ACL-flow-defined granularity without the need to specify match conditions that identify subscriber or tier-of-service
- Steer both upstream and downstream traffic on a single Res-GW

- Flexibly assign subscribers to tier-of-service by changing the ACL filter policy a given subscriber uses
- Flexibly add new services to a subscriber or tier-of-service by adding the subscriber-independent filter rules required to achieve steering
- Achieve isolation of VAS steering from other ACL functions like security through the use of embedded filters
- Deploy integrated Application Assurance (AA) as part of a VAS service chain - both upstream and downstream traffic is processed by AA before a VAS redirect
- Select whether to use IP-Src/IP-Dst address hash or IP-Src/IP-Dst address plus TCP/UDP port hash when LAG/ECMP connectivity to DC is used. L4 inputs are not used in hash with IPv6 packets with extension headers present.

ESM filter policy-based traffic steering supports the following:

- IPv4 and IPv6 steering of unicast traffic using IPv4 and IPv6 ACLs
- **action forward redirect-policy** or **action forward next-hop router** for IP-steering with TCAM-based load-balancing, fail-to-wire, and sticky destination
- **action forward esi sf-ip vas-interface router** for integrated service chaining solution

Operational notes:

- Downstream traffic steered towards a VAS on the subscriber-facing IOM is reclassified (FC and profile) based on the subscriber egress QoS policy, and is queued towards the VAS based on the network egress QoS configuration. Packets sent toward VAS will not have DSCP remarked (since they are not yet forwarded to a subscriber). DSCP remarking based on subscriber's egress QoS profile will only apply to traffic ultimately forwarded to the subscriber (after VAS or not subject to VAS).
- If mirroring of subscriber traffic is configured using ACL entry/subscriber/SAP/port mirror, the mirroring will apply to traffic ultimately forwarded to subscriber (after VAS or not subject to VAS). Note that traffic that is being redirected to VAS cannot be mirrored using an ACL filter implementing PBR action (the same egress ACL filter entry being a mirror source and specifying egress PBR action is not supported).
- Use dedicated ingress and egress filter policies to prevent accidental match of an ingress PBR entry on egress and vice-versa that will result in forwarding/dropping of traffic matching the entry (based on the filter's default action configuration).

Feature caveats:

- Requires chassis mode D
- Is not supported with HSMDAs on subscriber ingress
- Is not supported when the traffic is subject to non-AA ISA on Res-GW

- Traffic that matches an egress filter entry with egress PBR action cannot be mirrored, cannot be sampled using cflowd, and cannot be logged using filter logging while being redirected to VAS on a sub-facing line card.