ERICSSON

SACC Optimization

USER GUIDE

1/1553-AXB 250 12-V3 Uen AU

Copyright

© Ericsson AB 2012–2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.

Contents

1 Introduction

This document provides a set of recommendations to optimize the capacity and performance of the EPG for GPRS and EPS networks.

1.1 Scope

The performance of the EPG can be increased by optimizing Service-Aware Charging and Control (SACC) with or without Packet Inspection and Service Classification (PISC).

This document covers the following optimization topics:

— Introductory concepts, such as traffic category, type of subscription and UE

— No SACC or SACC without PISC

— SACC with PISC

A complete list of the optimization recommendations is described in Section 5 on page 35.

For an overview of the SACC business solution, refer to SACC Overview.

For an overview of the PISC functionality, refer to Packet Inspection and Service Classification (PISC). For information on how to configure PISC, refer to PISC Configuration.

For assistance with PISC configuration optimization, contact Ericsson support.

1.2 Target Groups

This document is intended for personnel to optimize SACC. The document assumes that the readers have a basic knowledge of data communication and telecommunication.

2 Factors Affecting EPG Performance and Capacity

Processing payload packets results in high load on the Packet Processing Board (PPB), which is the main cause of degraded performance and capacity of the EPG. Before considering the optimization, it is important to understand the following factors that have significant impacts on payload processing:

— Categories of traffic

— Type of UE device and subscription

The improper combination of these factors can lead to poor performance, for example, 10 times the relative CPU cost as the recommended combination. To have an optimized performance, the combination of these factors must be considered.

2.1 Categories of Traffic

The traffic can be divided into three main processing categories:

No SACC The packets are terminated on the GPRS Tunneling Protocol for User Plane (GTP-U) tunnel and forwarded to the Gi interface. Only postpaid traffic belongs to this category.

SACC without PISC The packets are tagged with a default Service Data Flow Identifier (SDF-ID) without any PISC. Using this function it is possible to apply policy control on the Gx+ interface and charging on the Gy+ interface. Service differentiation is not possible.

SACC with PISC The packets are inspected on header level (L3, L4) and on protocol level (L7). Service differentiation is possible.

Figure 1 shows that the traffic is divided by having separate Access Point Names (APNs) or rule spaces for each traffic category. The CPU cost varies among different traffic categories.

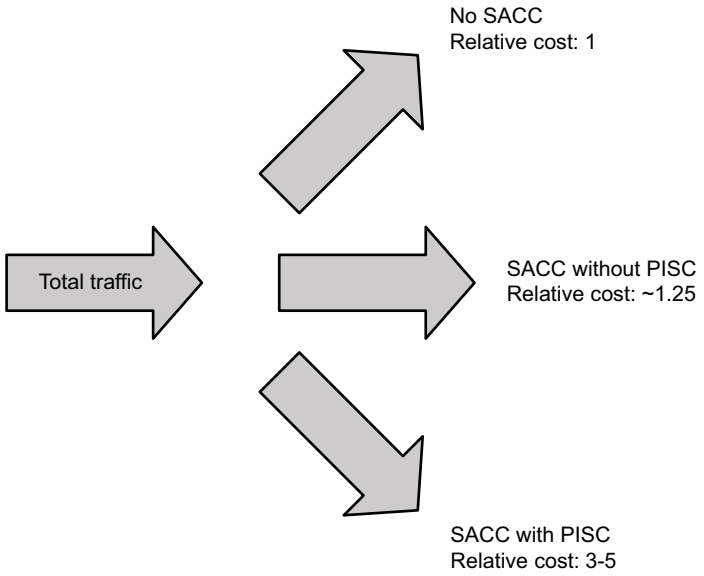

Figure 1 Example of CPU Cost per Packet with Optimized Configuration

— The CPU costs for the No SACC and SACC without PISC categories are both constant values.

— The CPU cost for the SACC with PISC category depends on the complexity of packet inspection. The value 3 is for header inspection and the value 5 is for Deep Packet Inspection (DPI) with a high number of rules.

2.2 Type of UE Device and Subscription

Table 1 shows an example of how UE devices can be divided and the relative volumes generated per UE.

Table 1 Example of UE Types and Relative Volume per UE Device

Type of UE	**Relative Volume per UE Device**
Feature phone	1
Smartphone	20
Tablet	50
PC, Laptop	150

The traffic can be handled in different ways depending on the subscription, for example:

— Mobile Subscription

  • Intended for mobile devices, such as feature phone, smartphone, tablet

- Monthly fee

— Mobile Broadband Subscription

- Intended for PCs and laptops
- Monthly fee
- Limited bandwidth if the use exceeds 5 GB during the month

From the performance perspective, the SACC with PISC category is recommended for mobile subscription, and the SACC without PISC category is recommended for mobile broadband subscription. This does not consider other factors such as user charging.

3 Direct Traffic to No SACC or SACC without PISC

The amount of processing depends on which route the packet takes through the EPG. With a given SACC solution, the capacity can differ significantly. Figure 1 shows that the No SACC category or the SACC without PISC category has relatively low CPU cost per packet compared to the SACC with PISC category. If all or most traffic can be directed to the former two categories, the impact of the traffic subject to the SACC with PISC category is small and the PPB capacity is improved.

Recommendation: Direct as much traffic as possible to the No SACC or SACC without PISC category. Consider the type of UE and its penetration when dividing into categories.

If it is not possible to follow this recommendation, optimizing PISC filter becomes important since all traffic is subject to the SACC with PISC category.

4 Optimize PISC Configuration

This section describes how to perform the PISC configuration optimization.

4.1 PISC Capacity Model

Figure 2 shows a simplified model of how header rules and protocol inspection rules are evaluated for one packet and can be used for capacity analysis.

— Minimize CPU intensive operations.

— The assignment of an SDF-ID to the packet suggests the end of the evaluation.

— Optimize the configuration to shorten the path to reach the assignment of the SDF-ID and avoid CPU intensive operations as much as possible.

In the model, the following two concepts play an important role in the capacity analysis:

Standard path — The initial IP packets in a service data flow (refer to SACC Overview for more information) are fully analyzed and classified in PISC. This first phase is referred to as standard path.

Fast path — Once a packet has been classified in an SDF-ID (other than the default SDF-ID of the service-set or the fallback SDF-ID of a header rule that directs traffic to a DPI rule set), the following packets belonging to the same service data flow can normally be classified without full analysis. This fast classification phase is referred to as fast path, which generally improves PISC performance.

Packets that are not classified in any PISC rule are classified in the default SDF-ID of the service-set. After some packets, a fast path is created to improve PISC performance.

Recommendation: Classify as much traffic as possible to PISC rules.

Sometimes traffic and enforcement require switching between fast path and standard path. For more information on how an IP flow is inspected, classified, and assigned an SDF-ID, refer to Packet Inspection and Service Classification (PISC).

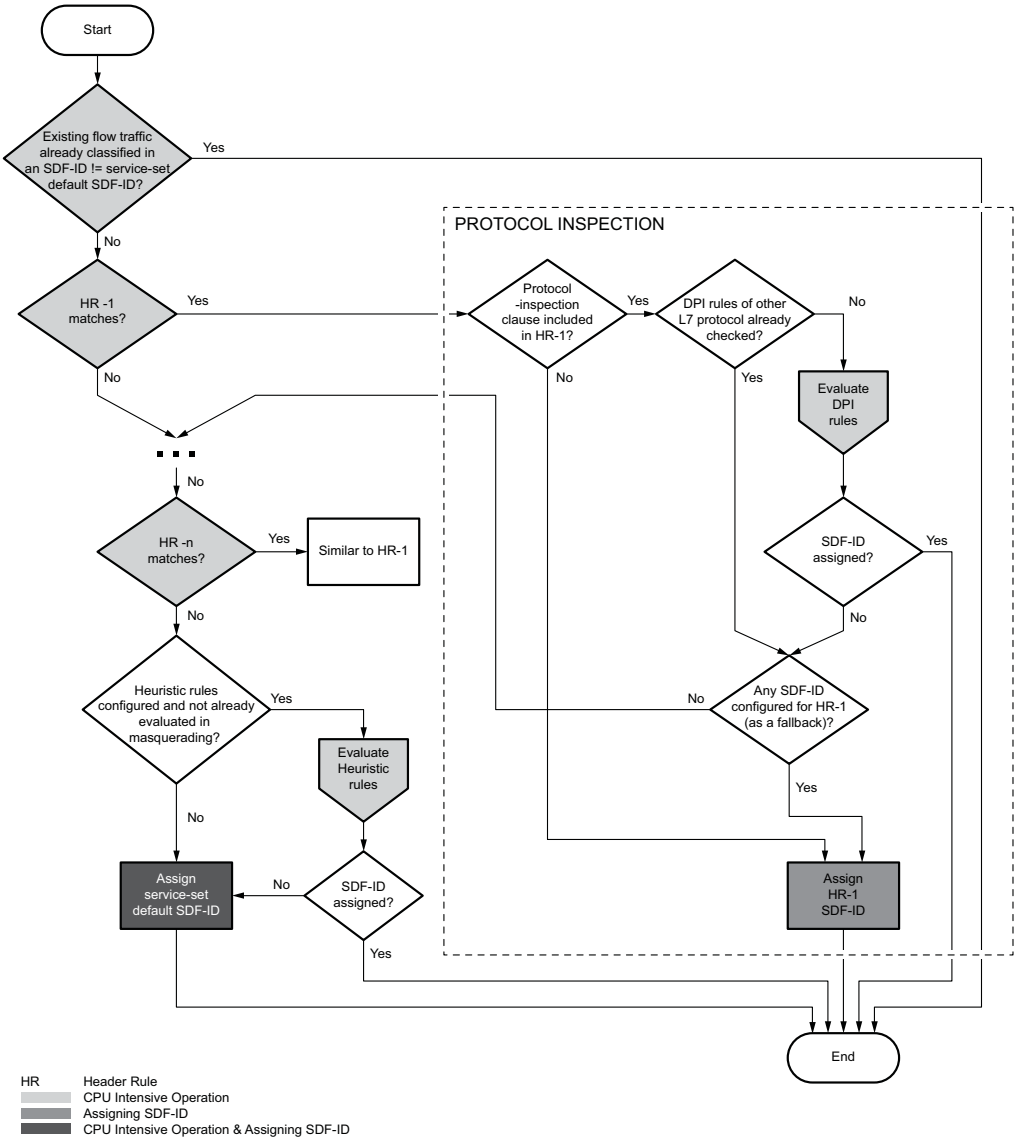

Figure 2 Header and Protocol Inspection Evaluation Overview

When a packet matches a header rule that contains a protocol inspection clause, the associated DPI rules are evaluated.

A header rule set can have multiple header rules with the same match conditions. Similarly a header rule can have multiple terms with the same match conditions. After the DPI evaluation for the first header rule or term according to the inspection order, the following header rules or terms are subject to DPI evaluation if the following conditions are met:

— No fallback SDF-ID is set in the preceding header rules or terms for which the DPI has been evaluated.

— The following header rules or terms are associated with different DPI rules which must be of the same L7 protocol.

The following header rules or terms are not subject to DPI evaluation if they are associated with the same DPI rules as the first header rule or term. That is, the algorithm does not check the DPI rules of L7 protocols again. Therefore, header rules in a header rule set or terms in a header rule must be as specific as possible. For additional information and examples, see Section 4.3.1 on page 13 and Section 4.3.2 on page 14.

Figure 3 shows the evaluation of DPI rules, including those rules with masquerading enabled.

If there are heuristic rules, they are evaluated after all header rules and the DPI rules are evaluated without match. However, if masquerading is enabled, the heuristics rules are evaluated after each masquerading-related DPI rule set that is evaluated without match. Figure 4 shows the evaluation of heuristic rules.

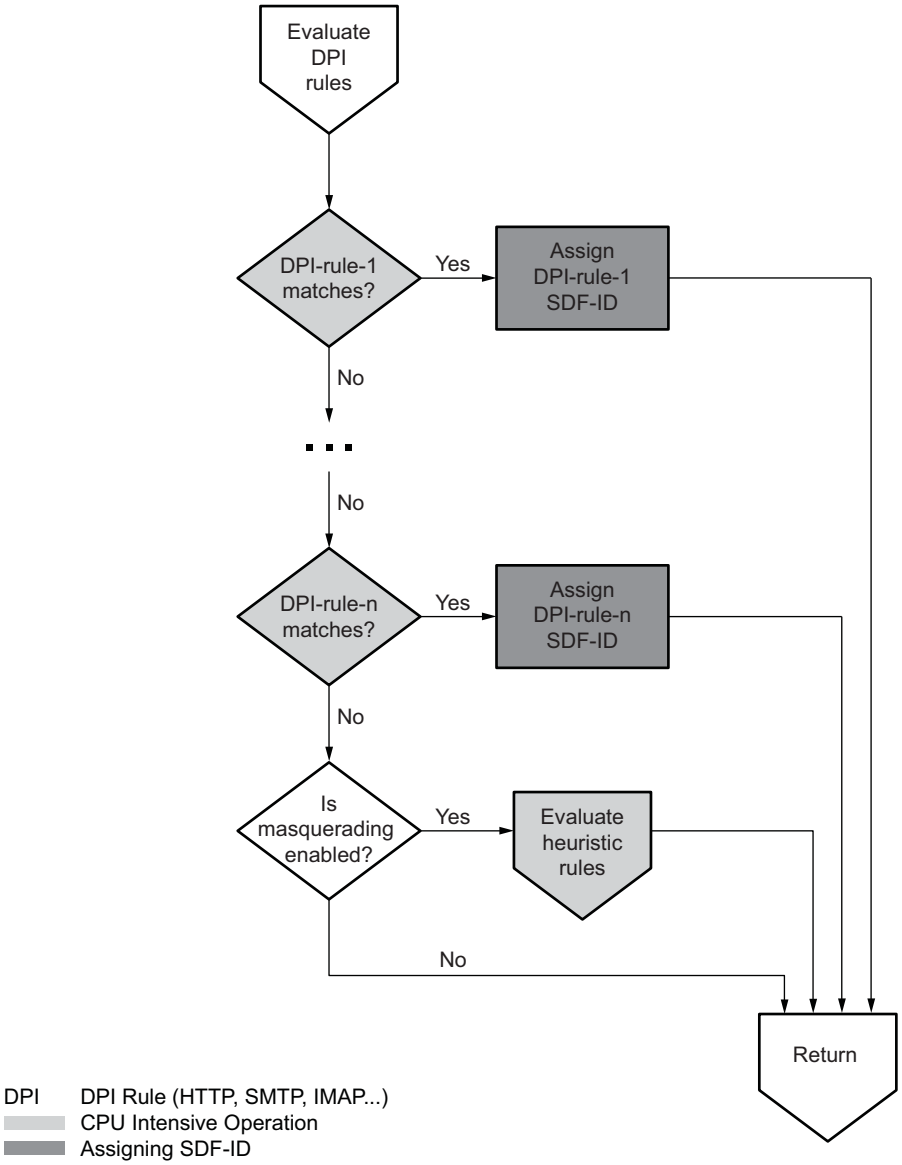

Figure 3 Evaluation of DPI Rules

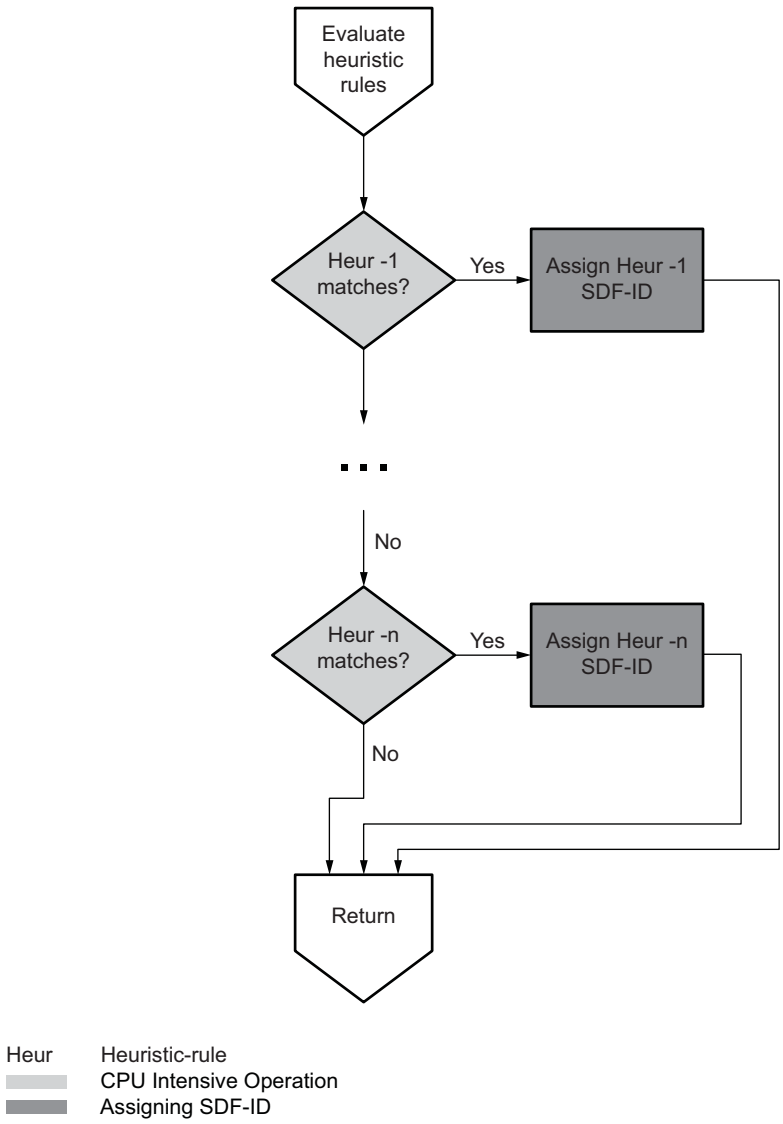

Figure 4 Evaluation of Heuristic Rules

Figure 5 illustrates the fast path and the different ways it can be created for a data flow.

Header rule fast path

It is created when a packet matches a header rule with no protocol inspection. The following packets of the same data flow do not require full analysis, and they are classified in the same SDF-ID.

DPI fast path

It is created when a packet matches a DPI rule. The following packets of the same data flow do not require full analysis, and they are classified in the same SDF-ID.

Heuristic fast path

It is created when a packet matches a heuristic rule. The following packets of the same data flow do not require full analysis, and they are classified in the same SDF-ID.

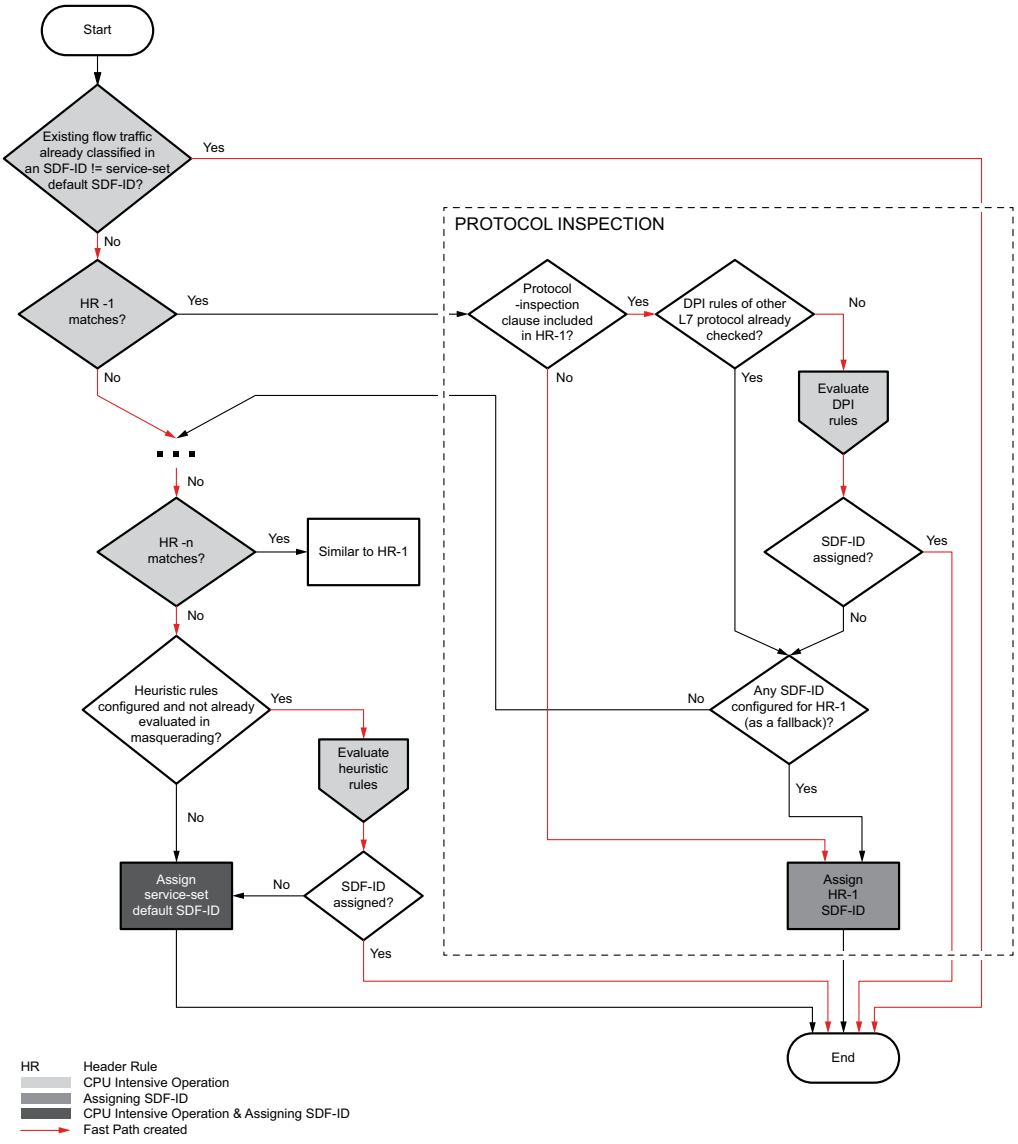

Figure 5 Fast Path

These figures are used in the following sections to illustrate the optimization recommendations.

4.2 Optimize Traffic Routing Observation

SDF-IDs are used to identify different paths for the traffic through PISC. Using identified paths, it is possible to see if the configuration can be optimized further. For example, if the analysis of the paths shows that many packets match the default SDF-ID of the service-set, the configuration can be optimized.

Identified paths are also required when contacting Ericsson support for troubleshooting, for example, for investigating CPU problems.

Recommendation: Use one unique SDF-ID per main path to make paths visible.

Use a unique default SDIF-ID for the service-set and do not share SDF-IDs between different path types. Figure 6 shows the main paths where unique SDF-IDs are needed:

Main Path	**SDF-ID**
Service-set (default)	0
Header rule match	11...1n
DPI rule match	21...2n
Heuristics rule match	31...3n

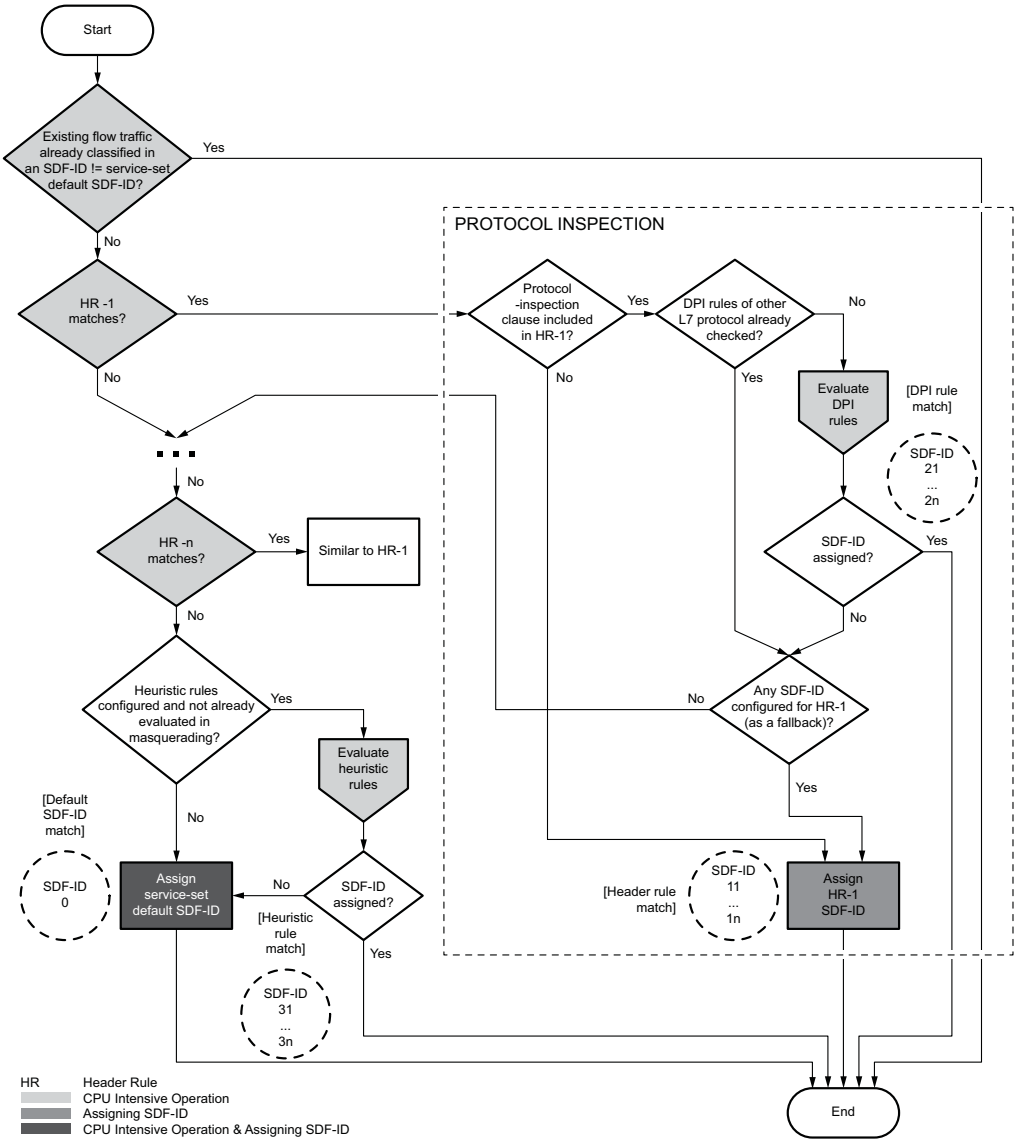

Figure 6 SDF-IDs for Main Paths

Example 1 shows an example of how to configure an SDF-ID to identify one path for the header rule match.

```
epg pgw service-identification header-rule hr1
 term 1
  name t1
  then service-data-flow-id payload 11
  from network-prefix [ 60.1.133.0/24 60.2.133.0/24 ]
 !
!
```

Example 1 Configuring an SDF-ID for the Header Rule Match

4.3 Optimize Header Inspection

This section describes the recommendations for optimizing header inspection in descending order of their impact on CPU load.

See Figure 2 and Figure 5 to facilitate the understanding of the recommendations in this section.

4.3.1 Configure Priority for Header Rules and Header Rule Sets

The evaluation order of header rule sets can be configured using the parameter `priority`. If no `priority` parameter is configured, header rule sets are evaluated in alphabetical order. The evaluation order of header rules in a header rule set is configured using the parameter `rule-id`. Both parameters, `priority` and `rule-id`, are evaluated in ascending order. Header rules takes the priority according to its position in the configuration file.

Recommendation: Reduce the number of rules traversed by configuring priority for header rules.

Example 2 shows an example of how to configure the priority of header rule sets. The hrs_1 header rule set has higher priority than hrs_2 header rule set and evaluated first.

```
epg pgw service-set A
 service-identification service-data-flow-id default payload 2
 service-identification header-rule-sets hrs2
  priority 2
 !
 service-identification header-rule-sets hrs_1
  priority 1
 !
!
```

Example 2 Configuring Header Rule Set Priority

Recommendation: Configure header rules in ascending order in the configuration file.

Example 3 shows an example of how to configure the priority of header rules in a header rule set. The hr_1 header rule has higher priority than hr_2 and hr3 header rules and evaluated first.

```
epg pgw service-identification header-rule-set hrs_test1
 rule 1
  name hr_1
 !
 rule 2
  name hr_2
 !
 rule 3
  name hr_3
 !
!
```

Example 3 Configuring Header Rule Priority

4.3.2 Reduce Traffic Not Matching Header Rules

It is possible that part of the traffic that matches the header rule does not match any of the rules in the associated DPI rule set. An SDF-ID can be configured as a fallback to match the traffic of this category, reducing the number of rules traversed significantly.

Recommendation: Reduce the number of rules traversed by configuring a fallback SDF-ID for header rules directing traffic to a DPI rule set.

Fallback SDF-IDs can be configured with either different IDs or the same. Configuring fallback SDF-IDs different from each other, and different from the default SDF-ID for the service set, separates traffic flows. This makes the paths distinguishable, and thus helps to troubleshoot performance issues, or be used for charging or other enforcement.

Example 4 shows an example of how to configure an SDF-ID as a fallback for the header rule `hr1`.

```
epg pgw service-identification header-rule hr1
 term 1
  name web
  then service-data-flow-id payload 11
  then protocol-inspection http-wsp-rule-set web-urs
  from network-address [ 60.170.0.1 ]
  from network-port [ 8080 ]
  from protocol tcp
 !
!
```

Example 4 Fallback SDF-ID for Header Rule

It is not recommended to exclude fallback SDF-IDs for header rules. However, if it is the case, Example 5 shows how DPI rules are processed in packet traverse.

```
epg pgw service-identification header-rule myhr-11
 term 10
  name t1
  then protocol-inspection http-wsp-rule-set mypi_11
  from domain is myurl1.com
  from network-port [ 80 8080 ]
  from protocol tcp
 !
!
epg pgw service-identification header-rule myhr-repeated
 term 10
  name t2
  then protocol-inspection http-wsp-rule-set mypi_11
  from domain is myurl1.com
  from network-port [ 80 8080 ]
  from protocol tcp
 !
!
epg pgw service-identification header-rule myhr-22
 term 10
  name t3
  then protocol-inspection http-wsp-rule-set mypi_22
  from domain is myurl1.com
  from network-port [ 80 8080 ]
  from protocol tcp
 !
!
epg pgw service-identification header-rule myhr-33
 term 10
  name t4
  then protocol-inspection ssl-tls-rule-set mypi_33
  from domain is myurl1.com
  from network-port [ 80 8080 ]
  from protocol tcp
 !
!
```

Example 5 DPI Evaluation in Packet Traverse

4.3.3 Configure Domain-Based Header Rules

If the IP addresses retrieved for the domain only serve HTTP traffic, a domain-based header rule can be configured instead of an HTTP rule. This helps to improve the performance.

Recommendation: Configure a domain-based header rule instead of an HTTP rule for the same domain.

Note: This recommendation cannot be applied if HTTP-based features, such as redirection, pipelining, or content filtering, are activated.

In Example 6 and Example 7, `ericsson.com` DNS resolution gives IP addresses `1.1.1.1` and `fe80::2.2.2.2`. Configuration in Example 6 is not recommended, whereas configuration with domain-based header rule in Example 7 is recommended.

```
epg pgw service-identification header-rule hr_dns_wrong
 term 1
  name term1
  then service-data-flow-id payload 1
  from network-address [ 1.1.1.1 ]
 !
 term 2
  name term2
  then service-data-flow-id payload 1
  from network-address [ fe80::2.2.2.2 ]
 !
!
```

Example 6 Configuring DNS Resolution Wrongly

```
epg pgw service-identification header-rule hr_dns_right
 term 1
  name term1
  then service-data-flow-id payload 1
  from domain contains ericsson.com
 !
!
```

Example 7 Configuring DNS Resolution Correctly in Domain-Based Header Rule

4.3.4 Group Header Rules

Header rules can be grouped and evaluated simultaneously if the following conditions are met:

— Five or more header rules are configured with any of the following match conditions:

- UE prefix, refer to PISC Configuration
- UE address, refer to PISC Configuration
- UE port, refer to PISC Configuration
- Network prefix, refer to PISC Configuration
- Network address, refer to PISC Configuration
- Network port, refer to PISC Configuration
- Protocol, refer to PISC Configuration
- Domain, refer to PISC Configuration

    - TTL match conditions except `notIs`, refer to PISC Configuration

— Either of the followings happens:

    - Any of the grouped header rules is not associated with application layer protocol inspection.

    - More than one of the grouped header rules is associated with the same application layer protocol inspection without a payload configured.

— Domain and TTL match conditions are not configured in the same header rule.

The following features are not affected by this optimization:

— Tethering detection

— Content filtering

— WSP setup signalling

— TCP teardown signalling

— TCP setup signalling (if the classification is based on network address)

When configuring DPI, header rules that are not associated with application layer protocol inspection can be configured at the beginning, to avoid standard path.

Recommendation: Group first the header rules that are not associated with application layer protocol inspection. Then group the header rules that are associated with the same application layer protocol inspection. Conditions listed in Section 4.3.4 on page 16 must be taken into account.

Example 8 shows an example of how to group the `hr1`, `hr2`, `hr3`, `hr4`, and `hr5` header rules. `hr1`, `hr2`, `hr3`, `hr4`, and `hr5` are not associated with any application layer protocol inspection, so they are grouped. Whereas `hr6` and `hr7` header rules are associated with the same `redirect-deep-inspection-set` application layer protocol inspection, but they belong to WSP setup signalling with payload configured, so`hr6` and `hr7` header rules are not grouped.

```
epg pgw service-identification header-rule-set hr
 rule 1
  name hr1
 !
 rule 2
  name hr2
 !
 rule 3
  name hr3
 !
 rule 4
  name hr4
 !
 rule 5
  name hr5
 !
 rule 6
  name hr6
 !
 rule 7
  name hr7
 !
!
epg pgw service-identification header-rule hr1
 term 1
  name h1t1
  then service-data-flow-id payload 1
  from network-port [ 25 ]
  from protocol tcp
 !
!
epg pgw service-identification header-rule hr2
 term 1
  name h2t1
  then service-data-flow-id payload 2
  from network-prefix [ 60.1.48.0/24 60.2.48.0/24 ]
  from protocol udp
 !
!
epg pgw service-identification header-rule hr3
 term 1
  name h3t1
  then service-data-flow-id payload 3
  from network-port [ 30 ]
 !
!
epg pgw service-identification header-rule hr4
 term 1
  name h4t1
  then service-data-flow-id payload 4
  from ttl-uplink 1
   is 64
  !
 !
!
epg pgw service-identification header-rule hr5
 term 1
  name h5t1
  then service-data-flow-id payload 5
  from ms-address [ 60.1.47.3 ]
```

Example 8 Grouping Header Rules

If both domain-based header rules and non-domain-based header rules are configured in one header rule set, grouping domain-based header rules together can improve the performance.

Recommendation: Group domain-based header rules.

Example 9 shows an example of how to group the header rules with the combination of both domain-based and non-domain-based header rules.

```
epg pgw service-identification header-rule-set hr
 rule 1
  name hr1
 !
 rule 2
  name hr2
 !
 rule 3
  name hr3
 !
 rule 4
  name hr4
 !
 rule 5
  name hr5
 !
!
epg pgw service-identification header-rule hr1
 term 1
  name h1t1
  then service-data-flow-id payload 1
  from network-port [ 25 ]
  from protocol tcp
 !
!
epg pgw service-identification header-rule hr2
 term 1
  name h2t1
  then service-data-flow-id payload 2
  from network-prefix [ 60.1.48.0/24 60.2.48.0/24 ]
  from protocol udp
 !
!
epg pgw service-identification header-rule hr3
 term 1
  name h3t1
  then service-data-flow-id payload 3
  then protocol-inspection http-wsp-rule-set redirect-deep-inspection-set
  from network-address [ 60.1.1.3 ]
 !
!
epg pgw service-identification header-rule hr4
 term 1
  name h4t1
  then service-data-flow-id payload 4
  from domain starts-with www.corporate1.com
  from network-port [ 80 8080 ]
  from protocol tcp
 !
!
epg pgw service-identification header-rule hr5
 term 1
  name h5t1
  then service-data-flow-id payload 5
  then protocol-inspection http-wsp-rule-set redirect-deep-inspection-set
  from domain ends-with .corporate2.com
  from network-port [ 80 8080 ]
  from protocol tcp
 !
!
```

4.3.5 Configure Header Rules for Unsolicited Traffic Classification

The header rules for unsolicited traffic classification must be grouped and given the highest priority. See the recommendation in Section 4.3.4 on page 16.

The header rule set which includes such header rules must also take the highest priority when being associated with the service set.

These actions can improve the performance.

Recommendation: Configure a header rule set that includes any header rule with traffic type unsolicited with the highest priority when associating the service set.

Example 10 shows an example of how to configure the header rules for unsolicited traffic classification and associate the header rule set with the service set. In the example, the header rule sets `hrs1` and `hrs2` do not include any header rule with traffic type unsolicited.

Recommendation: Configure the header rules with traffic type unsolicited with only the addresses and ports that the stateful firewall needs to control.

In Example 10, `ms-address=[60.1.1.3,60.2.2.4]` in the header rule `hr1` only specifies the addresses for which the hole punching is allowed. If `ms-prefix=60.1.0.0/14` is configured to replace `ms-address=[60.1.1.3,60.2.2.4]`, the addresses not requiring the hole punching are also included. This means more traffic analysis and classification, and thus requests more memory and CPU.

```
epg pgw service-identification header-rule-set hrs
 rule 1
  name hr1
 !
 rule 2
  name hr2!
 rule 3
  name hr3!
!
epg pgw service-identification header-rule hr1
 term 1
  name h1t1
  then service-data-flow-id payload 1
  from traffic-type unsolicited
  from ms-address [ 60.1.1.3 60.2.2.4 ]
  from network-address [ 60.1.0.2 60.2.0.2 ]
  from ms-port    [ 1414 2828 ]
  from protocol udp
 !
!
epg pgw service-identification header-rule hr2
 term 1
  name h2t1
  then service-data-flow-id payload 2
  from traffic-type unsolicited
  from ms-prefix [ 10.0.0.2/24 ]
  from network-prefix [ 10.0.0.1/24 ]
  from protocol udp
 !
!
epg pgw service-identification header-rule hr3
 term 1
  name h3t1
  then service-data-flow-id payload 3
  then protocol-inspection http-wsp-rule-set redirect-deep-inspection-set
  from network-prefix [ 60.1.48.0/24 60.2.48.0/24 ]
 !
!
epg pgw service-identification header-rule-set hrs1
!
epg pgw service-identification header-rule-set hrs2
!
epg pgw service-set ss1
 service-identification header-rule-sets hrs
  priority 1
 !
!
epg pgw service-set ss1
 service-identification header-rule-sets hrs1
  priority 2
 !
!
epg pgw service-set ss1
 service-identification header-rule-sets hrs2
  priority 3
 !
!
```

Example 10 Header Rules for Unsolicited Traffic Classification

4.3.6 Avoid Performance Degradation for HTTPS Traffic

The header rule for the HTTPS traffic has the port 443. However, if only 443 is configured for the port, all HTTPS traffic has to be analyzed to check the matching of the SSL/TLS rule, which has a large performance degradation. To avoid degrading the EPG node performance, the header rule associated with the SSL/TLS rule can include the IP address of the HTTP proxy, or the domain filter which is even better. The HTTP proxy or the domain filter leads to a CDN providing premium contents.

Recommendation: Avoiding performance degradation for HTTPS traffic by including the IP address of the HTTP proxy or domain filter in the header rule.

```
epg pgw service-identification header-rule https_proxy_ip
 term 1
  name web
  then service-data-flow-id payload 1
  then protocol-inspection ssl-tls-rule-set deep-inspection-set
  from network-address [ 172.18.100.223 ]
  from network-port [ 443 ]
 !
!
```

Example 11 HTTPS Header Rule with Proxy IP

```
epg pgw service-identification header-rule https_domain
 term 1
  name web
  then service-data-flow-id payload 1
  then protocol-inspection ssl-tls-rule-set deep-inspection-set
  from domain starts-with www.corporate.com
  from network-port [ 443 ]
 !
!
```

Example 12 HTTPS Header Rule with Domain

4.4 Optimize Deep Packet Inspection

This section describes the recommendations for optimizing DPI in descending order of their impact on CPU load.

See Figure 2 and Figure 3 to facilitate the understanding of the recommendations in this section.

4.4.1 Direct Well-Defined Traffic to DPI

To direct traffic to DPI classification, a header rule is needed. One common rule is “protocol TCP”. This rule directs all TCP traffic to DPI classification. Using this

common rule only can degrade the performance significantly and thus is not recommended. The L7 protocol can be any TCP-based protocol. Typically most of the traffic is HTTP. But in case of mobile broadband connections, non-HTTP traffic (for example P2P) is substantial. To ensure optimal performance, direct only well-defined traffic to DPI. If the well-defined traffic is HTTP, ports 80 and 8080 are used together with protocol TCP to filter out HTTP traffic.

Recommendation: Direct only well-defined traffic to DPI.

```
epg pgw service-identification header-rule all_http
 term 1
  name web
  then service-data-flow-id payload 21
  then protocol-inspection http-wsp-rule-set http_web
  from network-port [ 80 8080 ]
  from protocol tcp
 !
!
```

Example 13 Directing Only HTTP Traffic to DPI

This method can also be used for other types of traffic. If most of the traffic is UDP, configure the header rule to filter out the well-defined traffic for UDP.

```
epg pgw service-identification header-rule udp
 term 1
  name t1
  then service-data-flow-id payload 11
  then protocol-inspection http-wsp-rule-set http_rs_udp
  from network-prefix [ 60.1.48.0/24 60.2.48.0/24 60.3.48.0/24 ]
  from protocol udp
 !
!
```

Example 14 Directing UDP Traffic to DPI

4.4.2 Group Header Rules with DPI

If an HTTP-WSP rule is used for both HTTP and WSP, it is recommended to split the header rule to one TCP-based rule and one UDP-based rule. These rules are then grouped with the rest of the TCP/UDP rules.

Recommendation: Split the HTTP-WSP header rule into one TCP-based rule and one UDP-based rule if the rule is shared or reused for the two protocols.

The header rules from the HTTP-WSP header rule can be grouped under the TCP or the UDP header rule.

Recommendation: Group the header rules from the HTTP-WSP header rule under the TCP and the UDP header rules respectively.

Example 15 shows HTTP-WSP header rules used for both HTTP and WSP.

```
epg pgw service-identification header-rule http_wsp
 term 1
  name http1
  then service-data-flow-id payload 11
  then protocol-inspection http-wsp-rule-set HTTP
  from network-prefix [ 60.1.48.0/24 ]
  from protocol tcp
 !
 term 2
  name udp1
  then service-data-flow-id payload 12
  then protocol-inspection http-wsp-rule-set WSP
  from network-prefix [ 60.1.48.0/24 ]
  from protocol udp
 !
!
epg pgw service-identification header-rule ftp
 term 1
  name ftp
  then service-data-flow-id payload 22
  then protocol-inspection http-wsp-rule-set FTP
  from network-prefix [ 50.1.48.0/24 ]
 !
!
epg pgw service-identification header-rule http2
 term 1
  name http2
  then service-data-flow-id payload 11
  then protocol-inspection http-wsp-rule-set HTTP
  from network-prefix [ 120.1.48.0/24 ]
  from protocol tcp
 !
 term 2
  name udp2
  then service-data-flow-id payload 12
  then protocol-inspection http-wsp-rule-set WSP
  from network-prefix [ 120.1.48.0/24 ]
  from protocol udp
 !
!
epg pgw service-identification header-rule dns
 term 1
  name dns
  then service-data-flow-id payload 22
  then protocol-inspection http-wsp-rule-set DNS
  from network-prefix [ 40.1.48.0/24 ]
 !
!
```

Example 15 HTTP-WSP Header Rules for Both HTTP and WSP

Example 16 shows the recommended configuration of the rules described in Example 15.

```
epg pgw service-identification header-rule tcp
 term 1
  name http1
  then service-data-flow-id payload 11
  then protocol-inspection http-wsp-rule-set HTTP
  from network-prefix [ 60.1.48.0/24 ]
  from protocol tcp
 !
 term 2
  name ftp
  then service-data-flow-id payload 22
  then protocol-inspection http-wsp-rule-set FTP
  from network-prefix [ 50.1.48.0/24 ]
 !
 term 3
  name http2
  then service-data-flow-id payload 11
  then protocol-inspection http-wsp-rule-set HTTP
  from network-prefix [ 120.1.48.0/24 ]
  from protocol tcp
 !
!
epg pgw service-identification header-rule udp
 term 1
  name udp1
  then service-data-flow-id payload 12
  then protocol-inspection http-wsp-rule-set WSP
  from network-prefix [ 60.1.48.0/24 ]
  from protocol udp
 !
 term 2
  name udp2
  then service-data-flow-id payload 12
  then protocol-inspection http-wsp-rule-set WSP
  from network-prefix [ 120.1.48.0/24 ]
  from protocol udp
 !
 term 3
  name dns
  then service-data-flow-id payload 22
  then protocol-inspection http-wsp-rule-set DNS
  from network-prefix [ 40.1.48.0/24 ]
 !
!
```

Example 16 Splitting HTTP-WSP Header Rules to TCP and UDP Header Rules and Grouping Header Rules

4.4.3 Group DPI Rules with Associated Authentication Rules

The EPG authenticates extended HTTP headers based on authentication rules and authentication profiles. MD5 algorithm authentication and time stamp authentication can be enabled or disabled independently per authentication profile. If both types of authentication are enabled, it is recommended to configure all terms in the HTTP-WSP rule with an associated authentication rule without any other term configured in the same HTTP-WSP rule.

Recommendation: Group all terms with an associated authentication rule in the HTTP-WSP rule.

Example 17 shows how to group authentication rules in an HTTP-WSP rule.

```
epg pgw service-identification http-wsp-rule httpRule1
 term 1
  name term1
  then payload 5
  from http auth-rule-name authruleA
 !
 term 2
  name term2
  then payload 900
  from http auth-rule-name authruleB
 !
 term 3
  name term3
  then payload 50
  from http auth-rule-name authruleC
 !
 term 4
  name term4
  then payload 90
  from http auth-rule-name authruleD
 !
!
```

Example 17 Grouped Authentication Rules

Example 18 shows authentication rules that are not grouped in an HTTP-WSP rule.

```
epg pgw service-identification http-wsp-rule httpRule2
 term 1
  name term1
  then payload 5
  from http auth-rule-name authruleA
 !
 term 2
  name term2
  then payload 900
  from uri starts-with platinum
  from uri ends-with premium
 !
 term 3
  name term3
  then payload 50
  from http auth-rule-name authruleC
 !
 term 4
  name term4
  then payload 90
  from uri not-ends-with [ platinum premium ]
 !
!
```

Example 18 Not Grouped Authentication Rules

4.4.4 Reduce Impact of HTTP Traffic Classification Based on Parameters from HTTP Reply

HTTP traffic classifications based on parameters from HTTP reply such as response codes have a significant impact on CPU. It is thus recommended that the HTTP traffic towards the servers for which such classification is required goes through a protocol inspection that is started from a header rule with the following characteristics:

— A rule that is restricted as much as possible, for example, including only the IP addresses for the servers

— A rule that includes a default SDF-ID

This helps to reduce the percentage of traffic suffering from the costly analysis.

Recommendation: Use a restricted header rule with a default SDF-ID to classify HTTP traffic based on parameters from HTTP reply.

```
epg pgw service-identification header-rule http_wsp
 term 1
  name http
  then service-data-flow-id payload 11
  then protocol-inspection http-wsp-rule-set HTTP
  from network-address [ 60.1.48.10 ]
  from protocol tcp
 !
!
epg pgw service-identification http-wsp-rule-set HTTP
 rule 1
  name HTTP_response
 !
!
epg pgw service-identification http-wsp-rule HTTP_response
 term 1
  name term_11001
  then payload 11001
  then activate-event-tracking track [ response ]
  from uri starts-with http://video.corporate.com
  from http response-code is [ 200-299 ]
 !
!
```

Example 19 Reducing Impact of HTTP Traffic Classification Based on Parameters from HTTP Reply

4.4.5 Specify Protocol in HTTP-WSP Rule

If an HTTP-WSP rule is used only for HTTP or WSP, specifying the protocol (`http` or `wsp`) improves the throughput and requires less memory because the analysis can be focused on the specified protocol.

Recommendation: Specify the protocol (HTTP or WSP) in an HTTP-WSP rule if the rule is not shared or reused for the two protocols.

Note: This recommendation is recommended to be used only if the HTTP-WSP header rule cannot be split in two according to the recommendations in Section 4.4.2 on page 24.

If the protocol is specified, it is also recommended to configure different HTTP-WSP rule set in the dedicated header rule or term for the specified protocol. If `http` and `wsp` are specified in separate HTTP-WSP rules, place the corresponding header rule or term for `http` before the one for `wsp`.

```
epg pgw service-identification header-rule http_wsp
 term 1
  name http
  then service-data-flow-id payload 11
  then protocol-inspection http-wsp-rule-set HTTP
  from network-prefix [ 60.1.48.0/24 ]
  from protocol tcp
 !
 term 2
  name udp
  then service-data-flow-id payload 12
  then protocol-inspection http-wsp-rule-set WSP
  from network-prefix [ 60.1.48.0/24 ]
  from protocol udp
 !
epg pgw service-identification http-wsp-rule-set HTTP
 rule 1
  name HTTP_starts_with_http
 !
!
epg pgw service-identification http-wsp-rule-set WSP
 rule 1
  name HTTP_starts_with_wsp
 !
!
!
epg pgw service-identification http-wsp-rule HTTP_starts_with_http
 term 1
  name term_11001
  then payload 11001
  from uri starts-with http://video.baidu.com
  from http
 !
!
epg pgw service-identification http-wsp-rule HTTP_starts_with_wsp
 term 1
  name term_11002
  then payload 11002
  from uri starts-with http://video.baidu.com
  from wsp
 !
!
```

Example 20 Specifying Protocol in HTTP-WSP Rule

4.4.6 Use Separate Header Rule for Only MMS DPI Rules

Filtering HTTP, WSP, and MMS requires considerable processing resources. Differentiating the traffic in the associated header rules can decrease the impact on capacity. For example, TCP and UDP traffic can be divided into two or more

header rules. If DPI is configured for mixed traffic including MMS, HTTP and WSP, the associated header rule can be divided so that a dedicated header rule is used for only MMS DPI rules.

Recommendation: Use a separate header rule for only MMS DPI rules with the IP address filter set to MMS Center (MMSC) addresses.

```
epg pgw service-identification header-rule mms_HR
 term 1
  name mms-1
  then service-data-flow-id payload 21
  then protocol-inspection http-wsp-rule-set mms_rule_set
  from network-address [ 60.170.0.1 60.170.0.2 ]
  from protocol tcp
 !
!
```

Example 21 Header Rule for Only MMS DPI Rules

Where `60.170.0.1` and `60.170.0.2` are the IP addresses of the MMSC.

4.4.7 Limit Use of "contains" Condition in DPI Rules

The condition `contains` in DPI rules must be used carefully because it consumes more CPU resources than other conditions.

Recommendation: Avoid the use of condition "contains" in DPI engine when possible, replace with "startsWith" and "endsWith", and check if match conditions are desired.

Note: In general, two conditions over the same URI cost much more CPU resources. Therefore avoid the use of two conditions over the same URI when possible. If both `contains` and `startsWith` are present, only use `contains`.

```
epg pgw service-identification http-wsp-rule hwr-facebook
 term 1200
  name hwr_facebook_1200
  then payload          5223421
  then redirect-unauthorized
  from uri starts-with http://
  from uri contains [ .fbcdn.net ]
 !
 term 1300
  name hwr_facebook_1300
  then payload          5223421
  then redirect-unauthorized
  from uri starts-with http://
  from uri contains [ .facebook.com ]
 !
 term 1400
  name hwr_facebook_1400
  then payload          5223421
  then redirect-unauthorized
  from uri starts-with http://
  from uri contains [ .facebook.net ]
 !
!
epg pgw service-identification http-wsp-rule hwr_mlearning
 term 400
  name hwr_mlearning_400
  then payload 5223314
  then edit-content addmsisdn
  from uri starts-with http://webapps.andalabs.com:
  from uri contains [ / ]
 !
!
```

Example 22 Use of "contains" in DPI Rules

```
epg pgw service-identification http-wsp-rule hwr-facebook
 term 1200
  name hwr_facebook_1200
  then payload          5223421
  then redirect-unauthorized
  from uri contains [ .fbcdn.net ]
 !
 term 1300
  name hwr_facebook_1300
  then payload          5223421
  then redirect-unauthorized
  from domain ends-with .facebook.com
 !
 term 1400
  name hwr_facebook_1400
  then payload          5223421
  then redirect-unauthorized
  from domain ends-with .facebook.net
 !
!
epg pgw service-identification http-wsp-rule hwr_mlearning
 term 400
  name hwr_mlearning_400
  then payload 5223314
  then edit-content addmsisdn
  from uri starts-with http://webapps.andalabs.com:
 !
!
```

Example 23 Use of "startsWith" and "endsWith" in DPI Rules

4.5 Optimize Heuristic Analysis

This section describes the recommendations for optimizing heuristic analysis.

See Figure 2, Figure 3, and Figure 4 to facilitate the understanding of the recommendations in this section.

4.5.1 HTTP Masquerading

If HTTP masquerading detection is enabled, packets matching a header rule but not matching any of the associated HTTP-WSP rules are evaluated against the heuristic rules. See Figure 3.

For information on configuring HTTP masquerading detection for a service set, refer to PISC Configuration.

To minimize the performance impact of enabling HTTP masquerading detection, subject only the necessary traffic to heuristic analysis. If HTTP classification can

catch as much traffic as possible, the traffic that could be subject to heuristic classification is reduced.

Recommendation: Configure as many explicit HTTP rules as possible to minimize traffic subject to heuristic packet inspection.

4.6 Optimize Resource-Consuming Function Configuration

Several PISC-related functions, when enabled, can have an impact on the capacity and performance of the EPG. Therefore, configure these functions only when required. For example, header packet inspection is less resource-consuming than DPI. Therefore, use DPI only when necessary.

Recommendation: Configure resource-consuming functions only when required.

Note: HTTP, WSP, SMTP, and IMAP protocols are much less resource-consuming than the rest of the DPI protocols.

The following are resource-consuming functions:

— FTP path tracking

— Inspection of pipelined HTTP requests

— Reassembly of fragmented WTP packets

— Event-based tracking (except HTTP events)

— Independent classification of WSP teardown signaling

— Independent classification of TCP teardown signaling

— Independent classification of TCP teardown signaling without payload

— Volume counting on application level (except HTTP application volume counting)

— URI volume reporting, which has an important impact on the traffic it is configured for

— Heuristic packet inspection, which has an important impact on the traffic not captured through header rules or DPI rules

— IPv6 address normalization, which has an important impact on all IPv6 traffic

Contact Ericsson support for further information on how the capacity and performance are affected when these functions are enabled. All these functions are optional and disabled by default.

Dynamic charging rules are stateless. Using these rules with inspection rules affects stateful functions such as detection of handshakes (setup or teardown)

for any protocol with such features. Disruption of the stateful detection can thus result in incorrect classification. For example, the WTP reassembly feature or connection-oriented WSP is not used with dynamic charging rules.

4.6.1 Enable Content Filtering Cache

When using the content-filtering function, it is recommended to enable the `Cfiltering` cache. By using some extra memory, this cache avoids going to an external node for URLs whose verdict is already known, thus saving time and traffic bandwidth with the ICAP server.

Recommendation: Enable content filtering cache.

4.6.2 Increase HTTP/TCP Time-out When Header Enrichment Is Used

The default time-out value for HTTP and TCP flows is 30 seconds. However, if HTTP flows are content enriched, it is necessary to increase the time-out for the HTTP flows. The time-out for the TCP flows must be set to the same value as that for the HTTP flows that are content enriched. Otherwise, after a time-out, packets belonging to the same TCP flow can be classified into different SIs, and thus are charged differently.

In case of header enrichment, a packet that is retransmitted after flow time-out is not correctly enriched, and causes a storm of ACKs and retransmissions. This happens when a server is not responding.

Recommendation: Increase time-out values of HTTP and TCP for service sets with header enrichment to 180.

```
epg pgw service-set ss
 service-identification flow-timeout tcp timeout 180
 service-identification flow-timeout http timeout 180
!
```

Example 24 Increasing HTTP/TCP Time-out

5 Summary of SACC Optimization Recommendations

Table 2 lists all the recommendations described in the previous sections:

Table 2 SACC Optimization Recommendations by Area

Area	Recommendation
Traffic Direction	Direct as much traffic as possible to the No SACC or SACC without PISC category. Consider the type of UE and its penetration when dividing into categories.
PISC Capacity Model	Classify as much traffic as possible to PISC rules.
Traffic Routing Observation	Use one unique SDF-ID per main path to make paths visible.
Header Inspection(1)	• Reduce the number of rules traversed by configuring priority for header rules. • Reduce the number of rules traversed by configuring a fallback SDF-ID for header rules directing traffic to a DPI rule set. • Configure a domain-based header rule instead of an HTTP rule for the same domain. • Recommendation: Group first the header rules that are not associated with application layer protocol inspection. Then group the header rules that are associated with the same application layer protocol inspection. Conditions listed in Section 4.3.4 on page 16 must be taken into account. • Group domain-based header rules. • Configure a header rule set that includes any header rule with traffic type unsolicited with the highest priority when associating the service set. • Configure the header rules with traffic type unsolicited with only the addresses and ports that the stateful firewall needs to control. • Avoiding performance degradation for HTTPS traffic by including the IP address of the HTTP proxy or domain filter in the header rule.

Area	Recommendation
DPI(1)	• Direct only well-defined traffic to DPI. • Split the HTTP-WSP header rule into one TCP-based rule and one UDP-based rule if the rule is shared or reused for the two protocols. • Group the header rules from the HTTP-WSP header rule under the TCP and the UDP header rules respectively. • Group all terms with an associated authentication rule in the HTTP-WSP rule. • Use a restricted header rule with a default SDF-ID to classify HTTP traffic based on parameters from HTTP reply. • Specify the protocol (HTTP or WSP) in an HTTP-WSP rule if the rule is not shared or reused for the two protocols. • Use a separate header rule for only MMS DPI rules with the IP address filter set to MMS Center (MMSC) addresses. • Avoid the use of condition "contains" in DPI engine when possible, replace with “startsWith” and “endsWith”, and check if match conditions are desired.
Heuristic Analysis	Configure as many explicit HTTP rules as possible to minimize traffic subject to heuristic packet inspection.
Resource-Consuming Function Configuration	• Configure resource-consuming functions only when required. • Enable content filtering cache. • Increase time-out values of HTTP and TCP for service sets with header enrichment to 180.

(1) The listed recommendations are sorted by their impact on CPU load in descending order.