

# Release14 SMS Filtering

Configuration and maintenance



---

## Contents

1	Legal notices . . . . .	3
2	Introduction . . . . .	3
2.1	Intended audience . . . . .	3
2.2	Document conventions . . . . .	3
3	Using the CLI . . . . .	3
3.1	Command modes . . . . .	3
3.2	Accessing the CLI . . . . .	3
3.3	Predefined user account . . . . .	3
3.4	Command prompts . . . . .	4
3.5	Command completion . . . . .	4
3.6	Command history . . . . .	4
3.7	Command editing . . . . .	4
3.8	Command timeout . . . . .	4
4	Configuration basics . . . . .	4
4.1	Terminology . . . . .	4
4.2	Configuration hierarchy . . . . .	5
4.3	Entering and exiting configuration mode . . . . .	5
4.4	Navigating in configuration mode . . . . .	5
4.5	Commands for navigating in configuration mode . . . . .	5
4.6	Viewing the configuration . . . . .	5
5	Changing configuration information . . . . .	7
5.1	Adding or modifying configuration . . . . .	7
5.2	Deleting configuration . . . . .	7
5.3	Committing configuration changes . . . . .	8
5.4	Discarding configuration changes . . . . .	8
6	Managing system configuration . . . . .	8
6.1	Saving the running configuration . . . . .	8
6.2	Loading the saved configuration . . . . .	9
6.3	Archiving configuration versions on commit . . . . .	9
6.4	Rolling back to previous version . . . . .	9
6.5	Show configuration commands . . . . .	9
7	Commands for working with configuration . . . . .	12
8	SMS filtering . . . . .	12
8.1	Filtering gateway node . . . . .	13
8.1.1	Manual start-up procedure . . . . .	13
8.1.2	Init script start-up procedure . . . . .	14
8.1.3	Runtime node configuration . . . . .	16
8.1.3.1	Performance and memory . . . . .	16
8.1.3.2	Rating . . . . .	17
8.1.3.3	Static filtering lists . . . . .	17
8.2	Filtering rules processing system . . . . .	18
8.2.1	Filtering rules . . . . .	19
8.2.2	Rule match part . . . . .	22
8.2.2.1	Release14 Application Layer Protocol (R14P) matching . . . . .	22
8.2.2.2	MTP Level 3 (MTP3) User Adaptation Layer (M3UA) matching . . . . .	23

8.2.2.3	Signalling Connection Control Part (SCCP) matching . . . . .	24
8.2.2.4	Transaction Capabilities Application Part (TCAP) matching . . . . .	26
8.2.2.5	Mobile Application Part (MAP) matching . . . . .	26
8.2.2.6	Short message TPDU 3GPP TS 23.040 (SMSTPDU) matching . . . . .	28
8.2.2.7	Short Message Peer-to-Peer (SMPP) matching . . . . .	31
8.2.3	Basic matching . . . . .	35
8.2.3.1	Basic matching syntax . . . . .	35
8.2.3.2	Operand types . . . . .	37
8.2.3.2.1	NUMBER type operand . . . . .	37
8.2.3.2.2	STRING type operand . . . . .	37
8.2.3.2.3	REGEX type operand . . . . .	37
8.2.3.2.4	VARIABLE type operand . . . . .	37
8.2.3.2.5	LIST type operand . . . . .	38
8.2.3.2.6	LUA type operand . . . . .	38
8.2.3.3	Operator types . . . . .	38
8.2.3.3.1	Equality '==' operator . . . . .	38
8.2.3.3.2	Inequality '!=' operator . . . . .	39
8.2.3.3.3	Greater than '>' operator . . . . .	39
8.2.3.3.4	Greater than equal '>=' operator . . . . .	39
8.2.3.3.5	Less than '<' operator . . . . .	39
8.2.3.3.6	Less than equal '<=' operator . . . . .	39
8.2.3.3.7	Add to list '>>' operator . . . . .	40
8.2.3.3.8	Remove from list '<<' operator . . . . .	40
8.2.3.3.9	Remove list '—' operator . . . . .	40
8.2.3.4	Predefined constants and variables . . . . .	40
8.2.4	Advanced matching . . . . .	48
8.2.4.1	Inline scripting . . . . .	48
8.2.4.2	External scripting . . . . .	49
8.2.4.3	Lua pMINK module . . . . .	49
8.2.4.3.1	Variant parameter management . . . . .	49
8.2.4.3.2	Standard dynamic filtering lists . . . . .	52
8.2.4.3.3	Flood management . . . . .	56
8.2.4.3.4	Predefined constants . . . . .	58
8.2.5	Rule translate part . . . . .	62
8.2.5.1	REGEX_SR type operand . . . . .	62
8.2.5.2	R14P translations . . . . .	63
8.2.5.3	MTP Level 3 (MTP3) User Adaptation Layer (M3UA) translations . . . . .	63
8.2.5.4	Signalling Connection Control Part (SCCP) translations . . . . .	64
Acronyms	. . . . .	66
Glossary	. . . . .	66

## 1 Legal notices

This document, or any part thereof, may not, without the written consent of Release 14 LTD, be copied, reprinted or reproduced in any material form including but not limited to photocopying, transcribing, transmitting or storing it in any medium or translating it into any language, in any form or by any means, be it electronic, mechanical, optical, magnetic or otherwise.

The information contained herein is believed to be accurate and reliable. Release14 LTD accepts no responsibility for its use by any means or in any way whatsoever. Release14 LTD shall not be liable for any expenses, costs or damage that may result from the use of the information contained within this document. The information contained herein is subject to change without notice.

## 2 Introduction

This document describes various configuration options for the R14 SMS Filtering software.

### 2.1 Intended audience

This guide is intended for experienced network and system administrators who are familiar with using command line interface languages.

### 2.2 Document conventions

This document uses the following typographic conventions.

Monospace	- examples, command line output
<b>Bold Monospace</b>	- what you type at the command line
<b>Bold</b>	- commands, keywords and file names
<i>Italics</i>	- variables and values
<key>	- A key on your keyboard, such as <Space>, - Combinations of keys are joined with a "+" sign

## 3 Using the CLI

This chapter provides an overview of the SMS Filtering command line interface which is the primary interface for SMS Filtering software.

### 3.1 Command modes

There are two operational modes in SMS Filtering, operational and configuration mode. In the current version operational mode is used solely for accessing the configuration mode. When you log into the system you are in operational mode. Configuration mode provides access to commands for creating, modifying, deleting, committing and showing configuration information, as well as commands for navigating through the configuration hierarchy.

- To enter configuration mode from operational mode issue **configure** command
- To return to the operational mode press <Ctrl>+d. If there are uncommitted configuration changes they will be discarded after a timeout? Pressing <Ctrl>+d again in operational mode will log you out of the system.

### 3.2 Accessing the CLI

To access the command line interface of R14 SMS Filtering software you connect with SSH on port 22 with a predefined username and password.

### 3.3 Predefined user account

By default, currently the system has one predefined user account, the user **config**. The default password for user config is **config**. The **config** user has administrator level privileges and can execute all SMS Filtering commands

### 3.4 Command prompts

The command prompt shows the user account you are logged in, the hostname of the system and whether you are in operational mode or configuration mode.

The format of command prompt in operational mode is as follows

```
config@hostname:/ >
```

The format of command prompt in configuration mode is as follows:

```
config@hostname:/configure >
```

### 3.5 Command completion

SMS Filtering software can auto complete command syntax by entering the following:

<Tab>	Auto-completes a command
	<ul style="list-style-type: none"> <li>• If the command is unambiguous the system generates the next token in the syntax</li> <li>• If more than one completion is possible the system displays a set of next possible tokens</li> </ul>
? followed by	Pressing the <Tab>key after question mark generates context aware command completion assistance
<Tab>	command completion assistance
<Tab>	Displays all available SMS Filtering commands

### 3.6 Command history

The SMS Filtering command line interface supports a command history, where commands you executed are stored in a internal buffer and can be re-executed or edited.

<Up-Arrow>	Move to the previous command
<Down-Arrow>	Move to the next command

### 3.7 Command editing

The SMS Filtering command line interface supports command editing

<Right-Arrow>	Move forward in the command line
<Left-Arrow>	Move backward in the command line

### 3.8 Command timeout

SMS Filtering command line interface has a timeout of 60 seconds. If there is no activity on the system in that time frame, configuration changes that havent been committed will be lost.

## 4 Configuration basics

### 4.1 Terminology

Several versions of system configuration exist:

- **Active or running configuration.** This is the configuration that is actually loaded and used by the system.

- **Working configuration.** When you enter configuration mode and make changes, changes remain in working configuration until you commit the changes, at which time the configuration becomes active or running.
- **Saved configuration.** If you save configuration with the **save** command to a file.

## 4.2 Configuration hierarchy

SMS Filtering configuration is organized as a hierarchy of configuration statements with a hierarchical tree of nodes similar to the directory structure on a UNIX file system. There are three kinds of statements:

- Configuration nodes. This can be either:
  1. Single nodes
  2. Multi-nodes
  3. Attribute statements. These set the values or characteristics for parameters within a node.

From the systems point of view, a configuration node is different from a simple configuration attribute statement. A configuration *attribute statement* takes the form *attribute value*, as in the following example.

```
priority    100
```

A configuration *node* always has an enclosing pair of braces as in the following example:

```
rule_1 {
    priority "100"
}
```

## 4.3 Entering and exiting configuration mode

- To enter configuration mode from operational mode issue **configure** command.

```
config@hostname:/ > configure
Trying '127.0.0.1:10000'...OK
config@hostname:/configure >
```

- To return to the operational mode press <Ctrl>+d
- To disconnect and exit press <Ctrl>+d

## 4.4 Navigating in configuration mode

You can tell where you are in the configuration tree by the path in tree:

```
config@hostname:/configure/mno/fgn/fgn1 >
```

## 4.5 Commands for navigating in configuration mode

Command	Result
<b>edit config-node</b>	Navigates to a subnode in the configuration tree for editing
<b>top</b>	Exits to the top level of configuration mode
<b>up</b>	Navigates up one level in the configuration tree

## 4.6 Viewing the configuration

Use the show command in configuration mode to display node configuration:

```
config@hostname:/configure > show mno fgn fgn1
    pool [ 10000 ] - Correlation and rule processor pool size
    timeout [ 5 ] - Default correlation timeout
    timer_res - Timer resolution in seconds
    fworkers [ 4 ] - Number of rule processor threads
    sl_list_max [ 1000000 ] - Maximum number of elements in standard list (default = 1000)
```

fl_list_max	- Maximum number of elements in flood list (default = 1000)
fl_list_ttl	- Maximum flood list TTL in seconds (default = 3600)
routing	- Routing nodeset configuration
rating	- Rating nodeset configuration
lists	- Filtering lists
rules	- Filter configuration

Use the **configuration** command to display the whole SMS Filtering configuration

```
config@hostname:/configure > configuration
mno {
  fgn {
    fgn1 {
      pool      "10000"
      timeout   "5"
      fworkers  "4"
      sl_list_max "1000000"
      rating {
        rating-01 {
          remote {
            address "192.168.0.20"
            port    "33000"
          }
          local {
            address "192.168.0.10"
            port    "34001"
          }
          weight "1"
        }
      }
    }
  }
  rules {
    test-rule-00000 {
      description  "--- Block Alphasenders ---"
      priority     "10000"
      filter_result "2"
      definition {
        match {
          r14p {
            trunk_label "[+]TEST_T01_IN"
          }
          map {
            context {
              sm {
                imsi ":24412.*"
              }
            }
          }
          smstpdu {
            tp-oa {
              ton    "80"
            }
          }
        }
        translate {
          filter_result "2"
          filter_jump   "_lvl3-sms-originating-alphasender-blocked_10100000"
        }
      }
    }
  }
  _lvl3-sms-originating-alphasender-blocked_10100000 {
    description  "--- Block Alphasender --- Level 3 ---"
    priority     "10100000"
    filter_result "0"
    definition {
      translate {
        filter_result "0"
      }
    }
  }
}
```

```

    }
  }
}
}
}
}
}
}

```

## 5 Changing configuration information

### 5.1 Adding or modifying configuration

Add new configuration by creating a configuration node, using the **set** command in configuration mode. Modify existing configuration using the **set** command in configuration mode, as in the following example:

```

config@hostname:/configure > set mno fgn fgn1 rules test-rule-00000 description "test text"
config@hostname:/configure

```

Then us the **show** command to see the change:

```

config@hostname:/configure > show mno fgn fgn1 rules test-rule-00000
+description [ test text ] - Rule description
  priority [      10000 ] - Priority level
filter_result [          2 ] - Default filtering result
  definition - Rule configuration

```

Note the "+" in front of the new statement. This shows that this statement has been added to the configuration but the change is not yet committed. The change does not take effect until configuration is committed using the **commit** command.

The "+" is also visible while using the **configuration** command:

```

test-rule-00000 {
+description "test text"
priority      "10000"
filter_result "2"
definition {
  match {
    r14p {
      trunk_label "[+]TEST_T01_IN"
    }
  }
  map {
    context {
      sm {
        imsi ":24412.*"
      }
    }
  }
  smstpdu {
    tp-oa {
      ton      "80"
    }
  }
}
translate {
  filter_result "2"
  filter_jump   "_lvl3-sms-originating-alphasender-blocked_10100000"
}
}
}

```

You can modify configuration from the root of the configuration tree or use the **edit** command to navigate to the part of the tree where you want to change or add.

### 5.2 Deleting configuration

Use the **delete** command to delete a value, as in the following example:

```
config@hostname:/configure > delete mno fgn fgn1 rules test-rule-00000 description
```

You can use the **show** command to see the change:

```
config@hostname:/configure > show mno fgn fgn1 rules test-rule-00000
-description [ test text ] - Rule description
  priority [      10000 ] - Priority level
filter_result [          2 ] - Default filtering result
  definition                - Rule configuration
```

Note the "-" in front of the deleted statement. This shows that this statement has been deleted from the configuration but the change is not yet committed. The change does not take effect until configuration is committed using the **commit** command.

Use the **delete** command to delete a complete configuration node, as in the following example:

```
config@hostname:/configure > delete mno fgn fgn1 rules test-rule-00000
```

You can use the **show** command to see the change:

```
config@hostname:/configure > show mno fgn fgn1 rules
-test-rule-00000
```

Note the "-" in front of the deleted statement. This shows that this statement has been deleted from the configuration but the change is not yet committed. The change does not take effect until configuration is committed using the **commit** command.

## 5.3 Committing configuration changes

In the SMS Filtering software, configuration changes do not take effect until you commit them with the **commit** command:

```
config@hostname:/configure > commit
```

Lines with uncommitted changes are flagged as follows:

- "+" to indicate the line has been added
- "-" to indicate the line has been removed

Once you commit the changes, the flag disappears.

## 5.4 Discarding configuration changes

If you don't want to commit the changes, you can discard them with the **discard** command:

```
config@hostname:/configure > discard
```

# 6 Managing system configuration

## 6.1 Saving the running configuration

Save the running configuration by running the **save** command in configuration mode. You must specify the file name and path to the save file location:

```
config@hostname:/configure > save file-name /tmp/testconfig.cfg
Saving configuration to "/tmp/testconfig.cfg"...
Done
```

You can also save a configuration file to a location path other than the standard configuration directory by specifying a different path. You can save to a hard drive, compact Flash, or USB device. Note that the **save** command writes only committed changes.



## 6.2 Loading the saved configuration

To load a previously saved configuration use the **load** command in configuration mode. You must specify the file name and path to the save file location:

```
config@hostname:/configure > load file-name /tmp/testconfig.cfg
Loading new configuration file "/tmp/testconfig.cfg"...
Done
Merging new configuration file...
Done
```

## 6.3 Archiving configuration versions on commit

The system automatically archives the configuration whenever you commit a configuration change. By default system maintains unlimited versions of configuration in the archive.

## 6.4 Rolling back to previous version

You can roll back system configuration to any archived version using the **rollback** command.

```
config@hostname:/configure > rollback revision 1
Loading rollback configuration...
Done
Merging rollback configuration file...
Done
Committing rollback configuration...
Done
```

The latest committed configuration is revision 0. You can see the list of revisions by using the context aware completion assistance:

```
config@hostname:/configure > rollback revision ?<Tab>
0 - 2013-05-18 10:47:59
1 - 2013-05-18 10:45:11
2 - 2013-05-18 10:44:52
3 - 2013-05-18 10:44:24
4 - 2013-05-18 10:43:24
5 - 2013-05-17 17:50:48
6 - 2013-05-17 17:50:19
7 - 2013-05-17 17:37:43
8 - 2013-05-17 17:37:43
9 - 2013-05-17 17:37:43
```

## 6.5 Show configuration commands

You can show all configuration commands for the current configuration by using the **commands** command:

```
set !pmink
set pmink !routing
set !openli
set openli !leac
set openli leac !db
set openli !dr
set openli !drnfl9
set openli !mediations
set openli !probes
set !mno
set mno !dr
set mno !psgn
set mno !sgn
set mno !fgn
set mno fgn !fgln
set mno fgn fgln pool "10000"
set mno fgn fgln timeout "5"
set mno fgn fgln fworkers "4"
set mno fgn fgln sl_list_max "1000000"
```

```

set mno fgn fgin !routing
set mno fgn fgin !rating
set mno fgn fgin rating !rating-01
set mno fgn fgin rating rating-01 !remote
set mno fgn fgin rating rating-01 remote address "192.168.0.20"
set mno fgn fgin rating rating-01 remote port "33000"
set mno fgn fgin rating rating-01 !local
set mno fgn fgin rating rating-01 local address "192.168.0.10"
set mno fgn fgin rating rating-01 local port "34001"
set mno fgn fgin rating rating-01 weight "1"
set mno fgn fgin !lists
set mno fgn fgin !rules
set mno fgn fgin rules !test-rule-00000
set mno fgn fgin rules test-rule-00000 description "--- Block Alphasenders ---"
set mno fgn fgin rules test-rule-00000 priority "10000"
set mno fgn fgin rules test-rule-00000 filter_result "2"
set mno fgn fgin rules test-rule-00000 !definition
set mno fgn fgin rules test-rule-00000 definition !match
set mno fgn fgin rules test-rule-00000 definition match !commands
set mno fgn fgin rules test-rule-00000 definition match commands !sri-for-sm
set mno fgn fgin rules test-rule-00000 definition match commands sri-for-sm !cgpa
set mno fgn fgin rules test-rule-00000 definition match commands sri-for-sm cgpa !gt
set mno fgn fgin rules test-rule-00000 definition match commands sri-for-sm !cdpa
set mno fgn fgin rules test-rule-00000 definition match commands sri-for-sm cdpa !gt
set mno fgn fgin rules test-rule-00000 definition match commands sri-for-sm !nnn
set mno fgn fgin rules test-rule-00000 definition match commands sri-for-sm !an
set mno fgn fgin rules test-rule-00000 definition match commands sri-for-sm !msisdn
set mno fgn fgin rules test-rule-00000 definition match commands sri-for-sm !sca
set mno fgn fgin rules test-rule-00000 definition match !r14p
set mno fgn fgin rules test-rule-00000 definition match r14p trunk_label "[+]TEST_T01_IN"
set mno fgn fgin rules test-rule-00000 definition match !m3ua
set mno fgn fgin rules test-rule-00000 definition match !sccp
set mno fgn fgin rules test-rule-00000 definition match sccp !cgpa
set mno fgn fgin rules test-rule-00000 definition match sccp cgpa !gt
set mno fgn fgin rules test-rule-00000 definition match sccp !cdpa
set mno fgn fgin rules test-rule-00000 definition match sccp cdpa !gt
set mno fgn fgin rules test-rule-00000 definition match !tcp
set mno fgn fgin rules test-rule-00000 definition match !map
set mno fgn fgin rules test-rule-00000 definition match map !context
set mno fgn fgin rules test-rule-00000 definition match map context !sri-for-sm
set mno fgn fgin rules test-rule-00000 definition match map context sri-for-sm !msisdn
set mno fgn fgin rules test-rule-00000 definition match map context sri-for-sm !sca
set mno fgn fgin rules test-rule-00000 definition match map context sri-for-sm !nnn
set mno fgn fgin rules test-rule-00000 definition match map context sri-for-sm !an
set mno fgn fgin rules test-rule-00000 definition match map context !sm
set mno fgn fgin rules test-rule-00000 definition match map context sm imsi ":24412.*"
set mno fgn fgin rules test-rule-00000 definition match map context sm !scda
set mno fgn fgin rules test-rule-00000 definition match map context sm !scoa
set mno fgn fgin rules test-rule-00000 definition match map context sm !msisdn
set mno fgn fgin rules test-rule-00000 definition match !smstpdu
set mno fgn fgin rules test-rule-00000 definition match smstpdu !tp-da
set mno fgn fgin rules test-rule-00000 definition match smstpdu !tp-oa
set mno fgn fgin rules test-rule-00000 definition match smstpdu tp-oa ton "80"
set mno fgn fgin rules test-rule-00000 definition match !smpp
set mno fgn fgin rules test-rule-00000 definition !translate
set mno fgn fgin rules test-rule-00000 definition translate filter_result "2"
set mno fgn fgin rules test-rule-00000 definition translate filter_jump "_lvl3-sms-originating-
    ↪ alphasender-blocked_10100000"
set mno fgn fgin rules test-rule-00000 definition translate !commands
set mno fgn fgin rules test-rule-00000 definition translate commands !sri-for-sm
set mno fgn fgin rules test-rule-00000 definition translate commands sri-for-sm !map
set mno fgn fgin rules test-rule-00000 definition translate commands sri-for-sm !cgpa
set mno fgn fgin rules test-rule-00000 definition translate commands sri-for-sm cgpa !gt
set mno fgn fgin rules test-rule-00000 definition translate commands sri-for-sm !cdpa
set mno fgn fgin rules test-rule-00000 definition translate commands sri-for-sm cdpa !gt
set mno fgn fgin rules test-rule-00000 definition translate commands sri-for-sm !msisdn

```

```

set mno fgn fgln rules test-rule-00000 definition translate commands sri-for-sm !sca
set mno fgn fgln rules test-rule-00000 definition translate !r14p
set mno fgn fgln rules test-rule-00000 definition translate !m3ua
set mno fgn fgln rules test-rule-00000 definition translate !sccp
set mno fgn fgln rules test-rule-00000 definition translate sccp !cgpa
set mno fgn fgln rules test-rule-00000 definition translate sccp cgpa !gt
set mno fgn fgln rules test-rule-00000 definition translate sccp !cdpa
set mno fgn fgln rules test-rule-00000 definition translate sccp cdpa !gt
set mno fgn fgln rules !_lvl3-sms-originating-alphasender-blocked_10100000
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 description "--- Block
    ↪ Alphasender --- Level 3 ---"
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 priority "10100000"
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 filter_result "0"
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 !definition
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition !match
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match !commands
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands !
    ↪ sri-for-sm
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands
    ↪ sri-for-sm !cgpa
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands
    ↪ sri-for-sm cgpa !gt
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands
    ↪ sri-for-sm !cdpa
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands
    ↪ sri-for-sm cdpa !gt
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands
    ↪ sri-for-sm !nnn
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands
    ↪ sri-for-sm !an
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands
    ↪ sri-for-sm !msisdn
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match commands
    ↪ sri-for-sm !sca
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match !r14p
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match !m3ua
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match !sccp
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match sccp !cgpa
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match sccp cgpa !
    ↪ gt
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match sccp !cdpa
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match sccp cdpa !
    ↪ gt
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match !tcap
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match !map
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map !
    ↪ context
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ !sri-for-sm
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ sri-for-sm !msisdn
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ sri-for-sm !sca
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ sri-for-sm !nnn
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ sri-for-sm !an
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ !sm
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ sm !scda
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ sm !scoa
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match map context
    ↪ sm !msisdn
set mno fgn fgln rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match !smstpdu

```

```

set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match smstpdu !tp
  ↪ -da
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match smstpdu !tp
  ↪ -oa
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition match !smpp
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition !translate
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ filter_result "0"
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate !
  ↪ commands
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ commands !sri-for-sm
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ commands sri-for-sm !map
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ commands sri-for-sm !cgpa
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ commands sri-for-sm cgpa !gt
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ commands sri-for-sm !cdpa
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ commands sri-for-sm cdpa !gt
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ commands sri-for-sm !msisdn
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate
  ↪ commands sri-for-sm !sca
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate !r14p
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate !m3ua
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate !sccp
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate sccp !
  ↪ cgpa
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate sccp
  ↪ cgpa !gt
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate sccp !
  ↪ cdpa
set mno fgn fgin rules _lvl3-sms-originating-alphasender-blocked_10100000 definition translate sccp
  ↪ cdpa !gt

```

## 7 Commands for working with configuration

List of all commands for working with the configuration which you can access using the <Tab>key:

```

config@hostname:/configure >
    set - Creates a new node or modifies a value in an existing node
    show - Displays configuration node
    edit - Navigates to a subnode in the configuration tree for editing
    delete - Deletes a configuration node
    top - Exits to the top level of configuration mode
    up - Navigates up one level in the configuration tree
configuration - Displays configuration file contents
    commands - Displays configuration file commands
    commit - Applies any uncommitted configuration changes
    discard - Discard all configuration changes
    rollback - Load previous configuration revision
    load - Loads a saved configuration
    save - Saves the running configuration to a file

```

## 8 SMS filtering

R14 SMS Filtering is a high performance system for mobile operators used for network security enforcement. Using a plethora of SMS filtering features, it accomplishes ultimate flexibility and system security. The most powerful feature at the very core of the system is a highly flexible configuration and rule management.

## 8.1 Filtering gateway node

Filtering Gateway Node (FGN) is a logical node used for SMS filtering and/or data analysis. It was built on top of R14 Project MINK Framework (pMINK) and is mostly used in active network environments, connected directly or indirectly to both Signalling Gateway Node (SGN) and Signalling Transfer Point (STP).

With high availability in mind, logical nodes are always clustered together to ensure continuous operation and proper failover.

Project MINK Framework (pMINK) is an abstract platform offering high availability clustering with various failover and routing methods, depending on context. There are two types of signalling categories available in pMINK platform; **internal** and **external**. **External** signalling is context dependant and can vary greatly; therefore pMINK platform uses Signalling Gateway Node (SGN) as the main entry point and signalling converter. After successful conversion, **external** signalling is classified as **internal** signalling and transferred freely between various pMINK nodes using context free Release14 Application Layer Protocol (R14P).

R14P load balancing and label based routing exists as a core feature of R14P protocol library; any pMINK logical node can act as an R14P router and load balancer. To simplify system configuration and management, pMINK platform uses one specialized logical node for routing and load balancing of internal signalling; an R14P Routing Daemon (ROUTINGD).

The majority of pMINK nodes require two levels of configuration; **start-up** and **runtime**. Start-up configuration is based on command line arguments and varies between different types of nodes. Its intended use is to provide a simple configuration interface for **one-time-setup** parameters, R14P connections being one of them. Runtime configuration is handled by pMINK Configuration Daemon (CONFIGD), a specialized node featuring transaction based configuration handling and distribution. Another key component closely tied to pMINK Configuration Daemon (CONFIGD) is a pMINK Command Line Interface (CLI); a client interface to pMINK CONFIGD.

All pMINK nodes follow a simple start-up procedure; manual or via init script. pMINK init script support is currently limited to **Debian** since it is our preferred distribution for production. In case of other Linux environments, manual start-up procedure should be used.

### 8.1.1 Manual start-up procedure

Starting Filtering Gateway Node (FGN) without any command line arguments will display a list of start-up parameters.

```
fgnd - pMINK Filtering Gateway Node daemon
Copyright (c) 2012 Release14.org

Options:
-?      help
-i      unique daemon id
-t      daemon type override
-r      routing daemon address (ipv4:port)
-c      config daemon address (ipv4:port)
-D      start in debug mode

R14P Options:
=====
--r14p-streams      R14P Session stream pool          (default = 1000)
--r14p-stimeout     R14P Stream timeout in seconds      (default = 5)
--r14p-smsg-pool    R14P Service message pool          (default = 1000)
--r14p-sparam-pool  R14P Service message parameter pool (default = 5000)

RRP Options:
=====
--rrp-stimeout      RRP Sequence timeout in seconds    (default = 5)
--rrp-sequences     RRP Session sequence pool          (default = 1000)
```

Filtering Gateway Node (FGN) general start-up arguments explained:

Argument	Description
<b>-?</b>	Display a list of start-up parameters
<b>-i</b>	Unique node daemon id
<b>-t</b>	Override default <b>fgnd</b> daemon type, used for failover and clustering
<b>-r</b>	Address of R14P Routing Daemon (ROUTINGD)
<b>-c</b>	Address of pMINK Configuration Daemon (CONFIGD); used if runtime configuration cannot be obtained via R14P Routing Daemon (ROUTINGD)
<b>-D</b>	Start in debug mode; print extra logging information on start-up

Release14 Application Layer Protocol (R14P) start-up arguments explained:

Argument	Description
<b>-r14p-streams</b>	Maximum number of concurrent R14P streams
<b>-r14p-stimeout</b>	Maximum duration for each R14P stream
<b>-r14p-smsg-pool</b>	Maximum number of concurrent R14P Service Messages
<b>-r14p-sparam-pool</b>	R14P Service Message Parameter pool size

Release14 Routing and Rating Protocol (RRP) start-up arguments explained:

Argument	Description
<b>-rrp-stimeout</b>	Maximum duration for each RRP sequence
<b>-rrp-sequences</b>	Maximum number of concurrent RRP sequences

Example of Filtering Gateway Node (FGN) manual start-up command:

```
$ ./fgnd -i fgn1 -r 192.168.0.10:15000 \
--r14p-smsg-pool=1000 \
--r14p-streams=1000 \
--r14p-sparam-pool=100000 \
--rrp-sequences=1000
```

The previous command initializes Filtering Gateway Node (FGN) with daemon id set to **'fgn1'** and connects to R14P Routing Daemon (ROUTINGD) listening on **192.168.0.10:15000**. After successful initialization, FGN tries to establish a CONFIGD connection, requesting it via already connected ROUTINGD. In the final start-up stage, active configuration is pushed from CONFIGD to FGN and processed accordingly.

R14P and RRP start-up parameters can also be set, as seen in this example. The general rule of thumb is; **-r14p-streams**, **-r14p-smsg-pool** and **-rrp-sequences** should be set to the same value and recommended value for **-r14p-sparam-pool** can be calculated using the following formula:

```
r14p-sparam-pool = 100 * r14p-smsg-pool
```

### 8.1.2 Init script start-up procedure

Starting Filtering Gateway Node (FGN) using an init script makes it easier to configure start-up parameters and use third party tools for process monitoring and statistics. This start-up method consists of two files; **/etc/default/pmink\_fgnd** and **/etc/init.d/pmink\_fgnd**. The first one is a configuration file used for managing start-up parameters while the latter is the actual init script used for node start-up.

Example of **/etc/default/pmink\_fgnd** init script configuration file:

```
# *****
# Basic daemon parameters
# *****
# daemon id
DAEMON_ID="fgn1"
# daemon type, defaults to fgnd if not defined
DAEMON_TYPE=""
# router daemon connections (multiple separated by space)
R14P_ROUTERS="127.0.0.1:15000"
# explicit config daemon connections (priority over r14p router connections)
CFGD_NODES=""
```



```
# start in debug mode
DEBUG_MODE="1"
# core dump
CORE_DUMP="0"

# *****
# R14P parameters
# *****
# r14p stream pool
R14P_STREAMS="1000"
# r14p stream timeout
R14P_STIMEOUT="5"
# r14p service message pool
R14P_SMSG_POOL="1000"
# r14p service message param pool
R14P_SPARAM_POOL="5000"

# *****
# RRP parameters
# *****
# rrp sequence timeout
RRP_STIMEOUT="5"
# rrp sequence pool
RRP_SEQUENCES="1000"
```

Filtering Gateway Node (FGN) general start-up parameters explained:

Parameter	Description
<b>DAEMON_ID</b>	Unique node daemon id
<b>DAEMON_TYPE</b>	Override default <b>fgnd</b> daemon type, used for failover and clustering
<b>R14P_ROUTERS</b>	Address of R14P Routing Daemon (ROUTINGD)
<b>CFGD_NODES</b>	Address of pMINK Configuration Daemon (CONFIGD); used if runtime configuration cannot be obtained via R14P Routing Daemon (ROUTINGD)
<b>DEBUG_MODE</b>	Start in debug mode; print extra logging information on start-up
<b>CORE_DUMP</b>	Generate core dump if process crashes; important for debugging and issue reporting
<b>R14P_STREAMS</b>	Maximum number of concurrent R14P streams
<b>R14P_STIMEOUT</b>	Maximum duration for each R14P stream
<b>R14P_SMSG_POOL</b>	Maximum number of concurrent R14P Service Messages
<b>R14P_SPARAM_POOL</b>	R14P Service Message Parameter pool size
<b>RRP_STIMEOUT</b>	Maximum duration for each RRP sequence
<b>RRP_SEQUENCES</b>	Maximum number of concurrent RRP sequences

Release14 Application Layer Protocol (R14P) start-up arguments explained:

Parameter	Description
<b>R14P_STREAMS</b>	Maximum number of concurrent R14P streams
<b>R14P_STIMEOUT</b>	Maximum duration for each R14P stream
<b>R14P_SMSG_POOL</b>	Maximum number of concurrent R14P Service Messages
<b>R14P_SPARAM_POOL</b>	R14P Service Message Parameter pool size

Release14 Routing and Rating Protocol (RRP) start-up arguments explained:

Parameter	Description
<b>RRP_STIMEOUT</b>	Maximum duration for each RRP sequence
<b>RRP_SEQUENCES</b>	Maximum number of concurrent RRP sequences

Example of Filtering Gateway Node (FGN) start-up command:

```
$ /etc/init.d/pmink_fgnd start
[ ok ] Starting pMINK Filtering Gateway Node Daemon....
```

Example of Filtering Gateway Node (FGN) termination command:

```
$ /etc/init.d/pmink_fgnd stop
[ ok ] Stopping pMINK Filtering Gateway Node Daemon...
```

Example of Filtering Gateway Node (FGN) re-start command:

```
$ /etc/init.d/pmink_fgnd restart
[ ok ] Stopping pMINK Filtering Gateway Node Daemon....
[ ok ] Starting pMINK Filtering Gateway Node Daemon....
```

Example of Filtering Gateway Node (FGN) status command:

```
$ /etc/init.d/pmink_fgnd status
[ ok ] is running
```

### 8.1.3 Runtime node configuration

Filtering Gateway Node (FGN) filtering rule management is maintained by pMINK Configuration Daemon (CONFIGD) and accessed via pMINK Command Line Interface (CLI). Runtime configuration is split into four major sections; **performance and memory**, **rating**, **lists** and **rules**. Although *performance and memory* section is a part of runtime configuration, it is processed only once, during FGN start-up. Any changes made to this section of runtime configuration will not take effect until FGN node is restarted.

#### 8.1.3.1 Performance and memory

Example of Filtering Gateway Node (FGN) runtime configuration, *performance and memory* section:

```
config@hostname:/configure > show mno fgn fgn1
    pool [ 10000 ] - Correlation and rule processor pool size
    timeout [ 5 ] - Default correlation timeout
    timer_res [ 10 ] - Timer resolution in seconds
    fworkers [ 4 ] - Number of rule processor threads
    sl_list_max [ 1000000 ] - Maximum number of elements in standard list (default = 1000)
    fl_list_max [ 1000000 ] - Maximum number of elements in flood list (default = 1000)
    fl_list_ttl [ 3600 ] - Maximum flood list TTL in seconds (default = 3600)
    routing - Routing nodeset configuration
    rating - Rating nodeset configuration
    lists - Filtering lists
    rules - Filter configuration
```

Configuration items explained:

Item	Description
<b>pool</b>	Maximum number of rule processors and buffered Short Message Service (SMS) messages. Filtering process can become an asynchronous operation, during which the SMS data needs to be buffered. Maximum number of buffered SMS messages is finite and specified by this configuration item.
<b>timeout</b>	Maximum duration for each buffered SMS
<b>timer_res</b>	Time interval at which buffered SMS messages are checked for timeout
<b>fworkers</b>	Number of threads used for processing filtering rules
<b>sl_list_max</b>	Maximum number of items in each user defined static list
<b>fl_list_max</b>	Maximum number of items in each user defined flood list
<b>fl_list_ttl</b>	Maximum duration of each item in user defined flood list



### 8.1.3.2 Rating

This FGN runtime configuration section consists of dynamic block nodes; configuration containers used for defining Release14 Routing and Rating Protocol (RRP) connections to dedicated Call Data Record (CDR) collectors. Each Filtering Gateway Node (FGN) can have multiple RRP rating connections defined; they are pooled together and selected using Weighted Round Robin (WRR) network scheduling algorithm.

Example of Filtering Gateway Node (FGN) runtime configuration, *rating* section:

```
config@hostname:/configure/mno/fgn/fgn1/rating > configuration
rating-01 {
    remote {
        address "192.168.0.20"
        port    "33000"
    }
    local {
        address "192.168.0.10"
        port    "34001"
    }
    weight "1"
}
```

Detailed view of *rating/rating-01* section:

```
config@hostname:/configure/mno/fgn/fgn1/rating > show rating-01
remote      - Remote address
local       - Local address
weight [ 1 ] - Priority
```

Detailed view of *rating/rating-01/remote* section:

```
config@hostname:/configure/mno/fgn/fgn1/rating > show rating-01 remote
address [ 192.168.0.20 ] - IP address
port [      33000 ] - Port
```

Detailed view of *rating/rating-01/local* section:

```
config@hostname:/configure/mno/fgn/fgn1/rating > show rating-01 local
address [ 192.168.0.10 ] - IP address
port [      34001 ] - Port
```

Configuration items explained:

Item	Description
<b>rating-01</b>	User defined dynamic block node, represents a Call Data Record (CDR) collector named ' <b>rating-01</b> '
<b>remote address</b>	IP address of Call Data Record (CDR) collector
<b>remote port</b>	RRP Listening port of Call Data Record (CDR) collector
<b>local address</b>	Local IP address to bind to when connecting to Call Data Record (CDR) collector
<b>local port</b>	Local RRP port to bind to when connecting to Call Data Record (CDR) collector
<b>weight</b>	Weighted Round Robin (WRR) weight parameter used for current rating connection

### 8.1.3.3 Static filtering lists

This section of FGN runtime configuration features user defined static lists. The term *static*, when used in this context, describes a list which is predefined with known values and its content should only be changed via pMINK Command Line Interface (CLI). Although *static* lists overlap with *dynamic* lists, to simplify rule management and maintenance, **content-changing**<sup>1</sup> list management commands should not reference any of the *statically* defined lists.

<sup>1</sup> Described in more detail starting from section 8.2.3.3.7

Example of Filtering Gateway Node (FGN) runtime configuration, *lists* section:

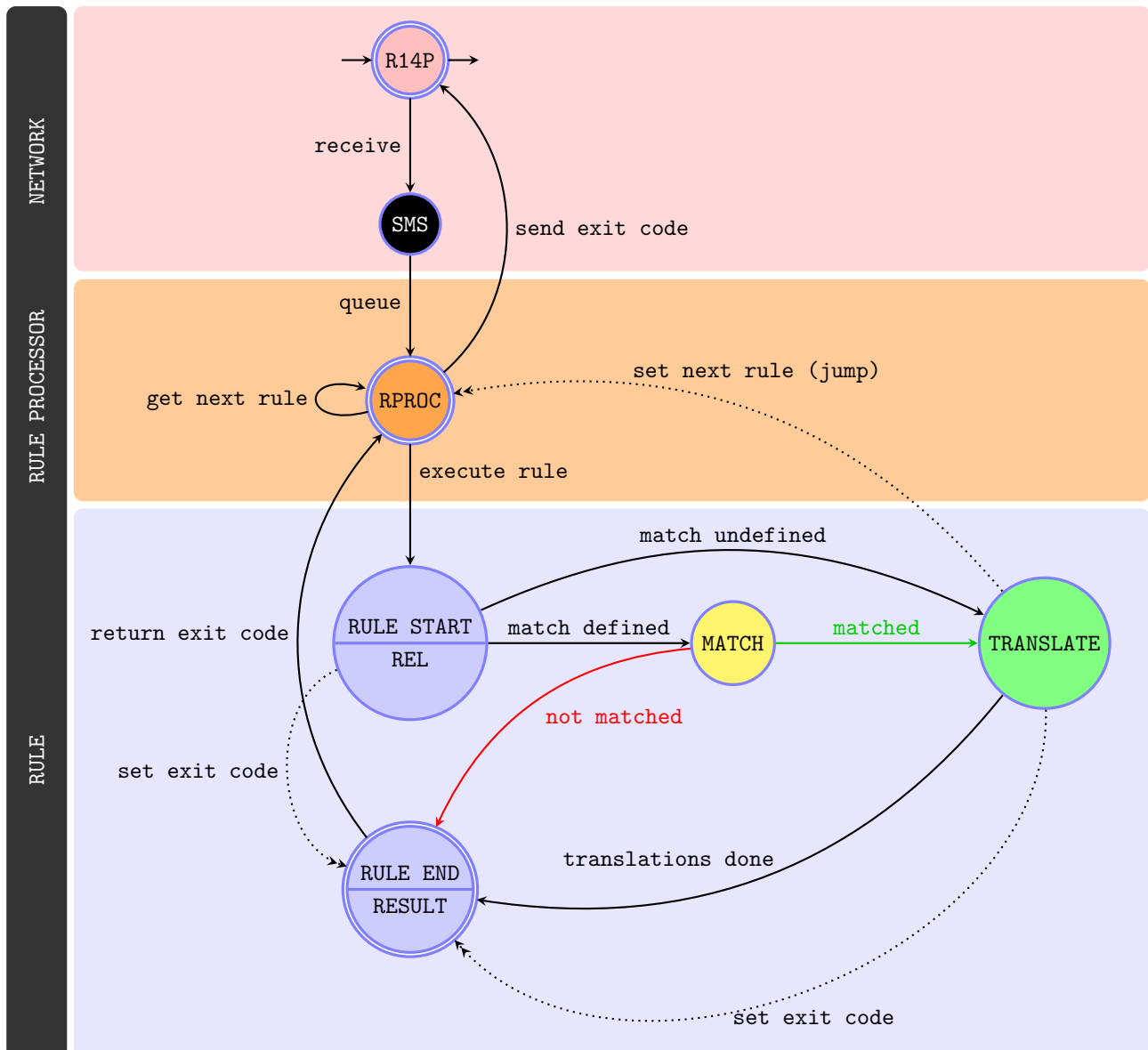
```
config@hostname:/configure/mno/fgn/fgn1/lists > configuration
list-01 {
    values {
        value1
        value2
    }
}
```

Implementation of *static* lists is based on dynamic block nodes; each user defined list is represented by one block container whose values part is a list of unique alphanumeric values.

Configuration items explained:

Item	Description
<b>list-01</b>	User defined dynamic block node, represents a static user list named ' <b>list-01</b> '
<b>values</b>	Grouping node, contains all user defined values for the current list
<b>value1 and value2</b>	User defined alphanumeric values for the current list

## 8.2 Filtering rules processing system



Filtering Gateway Node (FGN) rule processing system in a Finite-state machine (FSM) used primarily for, but not limited to, filtering of Short Message Service (SMS) messages.

Project MINK Framework (pMINK) implements an abstract Rule Processing Engine (RPE) used in two types of daemons; Signalling Transfer Point (STP) and Filtering Gateway Node (FGN). Although both daemon types use the same engine, they are quite different. While STP daemon's main focus is *routing*, FGN daemon offers an extended set of features, providing extra aid in *threat detection* and *filtering*.

pMINK Rule Processing Engine (RPE) comprises three main components; a Rule Processor (RPROC), Rule Execution Logic (REL) and a set of rules. The *first two components* exist as a core part of FGN daemon; they are referenced in this chapter for clarification purposes only, to provide a better overall understanding of system internals.

Rule Processor is a starting point of Rule Processing Engine; it serves a purpose of being a rule flow control coordinator and initiator of Rule Execution Logic. The separation between Rule Processor and Rule Execution Logic was introduced due to necessity to make Rule Processing Engine more versatile and context free solution. Different scenarios may require a different interpretation of rules but still retain the same method of rule execution; Signalling Transfer Point (STP) and Filtering Gateway Node (FGN) both share the same Rule Processor (RPROC) but use different Rule Execution Logic (REL) implementations.

Rule Execution Logic represents a link between a rule and a Rule Processor. Filtering rules, as seen in the pMINK Command Line Interface (CLI), present themselves in a human readable form; one of two available rule representations. The second one, used exclusively by Rule Execution Logic, is a compressed version optimized for machine interpretation. The most accurate definition of Rule Execution Logic (REL) would be that it is a rule *bytecode interpreter* governed by a Rule Processor (RPROC).

The final component of Rule Processing Engine is a set of rules. As already mentioned, rules are interpreted by Rule Execution Logic (REL) which is executed by a Rule Processor (RPROC). Each rule consists of four major states; "START", "MATCH", "TRANSLATE" and "END". The "START" state becomes active upon successful rule initialization; a default **exit code**<sup>2</sup> is set before further action takes place. The flow of rule execution from this point on depends on rule definition; if "match" section is empty or evaluates to true, execution will be shifted to "translate" section and current rule will change its state to "TRANSLATE". In other case, "match" section will evaluate to false and execution of "translate" section will be skipped. Rules are always executed in a sequential way and sorted by their priority values. This behaviour can be changed by using the "translate" section to enable rule jumping and temporarily change the next rule in Rule Processor's execution queue. The final state of rule execution is the "END" state; a default **exit code** is returned to a Rule Processor and evaluated accordingly.

There are three available **exit code** values currently in use; "ACCEPT(0)", "DROP(1)" and "CONTINUE(2)". They can be set during rule initialization ("START" state) and overridden in "translate" section ("TRANSLATE" state); if **exit code** value remains unset, it will default to "ACCEPT". The first two, "ACCEPT" and "DROP", will cause a Rule Processor to stop executing more rules and forward the **exit code** to the network layer. The "CONTINUE" **exit code** is used when rule execution should continue to the next rule in Rule Processor's queue; if rule jumping is used, **exit code** is automatically overridden and set to "CONTINUE".

**Note:** Network layer translates "DROP" **exit code** to **unknownSubscriber** MAP error in case of SS7, **ESME Receiver Permanent App Error** will be used in in case of SMPP. These error code translations are set by default; they can be overridden at any point in "match" section or rule definition.

### 8.2.1 Filtering rules

FGN runtime configuration of filtering rules is based on dynamic block nodes; each user defined rule is represented by one block container whose content contains initialization, "match" and "translate" sections of rule definition.

Naming of rules has no meaning to Rule Processing Engine (RPE); literal strings used for rule naming are required only in case of rule jumping, as a *filter\_jump* field value.

It is recommended to use consistent naming conventions for both rule names and priorities; maintaining a large set of rules without proper planing can soon become a quite daunting task.

<sup>2</sup> Runtime configuration uses different terminology; **filter\_result** corresponds to **exit code**

Example of Filtering Gateway Node (FGN) runtime configuration, *rules* section:

```
config@hostname:/configuration/mno/fgn/fgn1/rules > configuration
rule-00000 {
    description    "--- Alphasender check ---"
    priority       "1"
    filter_result  "2"
    definition {
        match {
            smstpdu {
                tp-oa {
                    ton      "@smstpdu.const.noa_alphanumeric"
                }
            }
        }
        translate {
            filter_jump    "blocked"
        }
    }
}
allowed {
    description    "--- Allowed ---"
    priority       "100000"
    filter_result  "0"
}
blocked {
    description    "--- Blocked ---"
    priority       "200000"
    filter_result  "1"
}
```

Detailed view of *rules/rule-00000* section:

```
config@hostname:/configuration/mno/fgn/fgn1/rules > show rule-00000
description [ --- Alphasender check --- ] - Rule description
priority [ 1 ] - Priority level
filter_result [ 2 ] - Default filtering result
definition - Rule configuration
```

Detailed view of *rules/rule-00000/definition/match/smstpdu/tp-oa* section:

```
config@hostname:/configuration/mno/fgn/fgn1/rules > show rule-00000 definition match smstpdu tp-oa
ton [ @smstpdu.const.noa_alphanumeric ] - Type of number
np - Numbering plan
address - Address
```

Detailed view of *rules/rule-00000/definition/translate* section:

```
config@hostname:/configuration/mno/fgn/fgn1/rules > show rule-00000 definition translate
filter_result - Filtering result
filter_jump [ blocked ] - Jump to rule label
commands - Special commands
r14p - R14P pmink framework data
m3ua - MTP Level 3 (MTP3) User Adaptation Layer
sccp - Signalling Connection Control Part
```

Detailed view of *rules/allowed* section:

```
config@hostname:/configuration/mno/fgn/fgn1/rules > show allowed
description [ --- Allowed --- ] - Rule description
priority [ 100000 ] - Priority level
filter_result [ 0 ] - Default filtering result
definition - Rule configuration
```

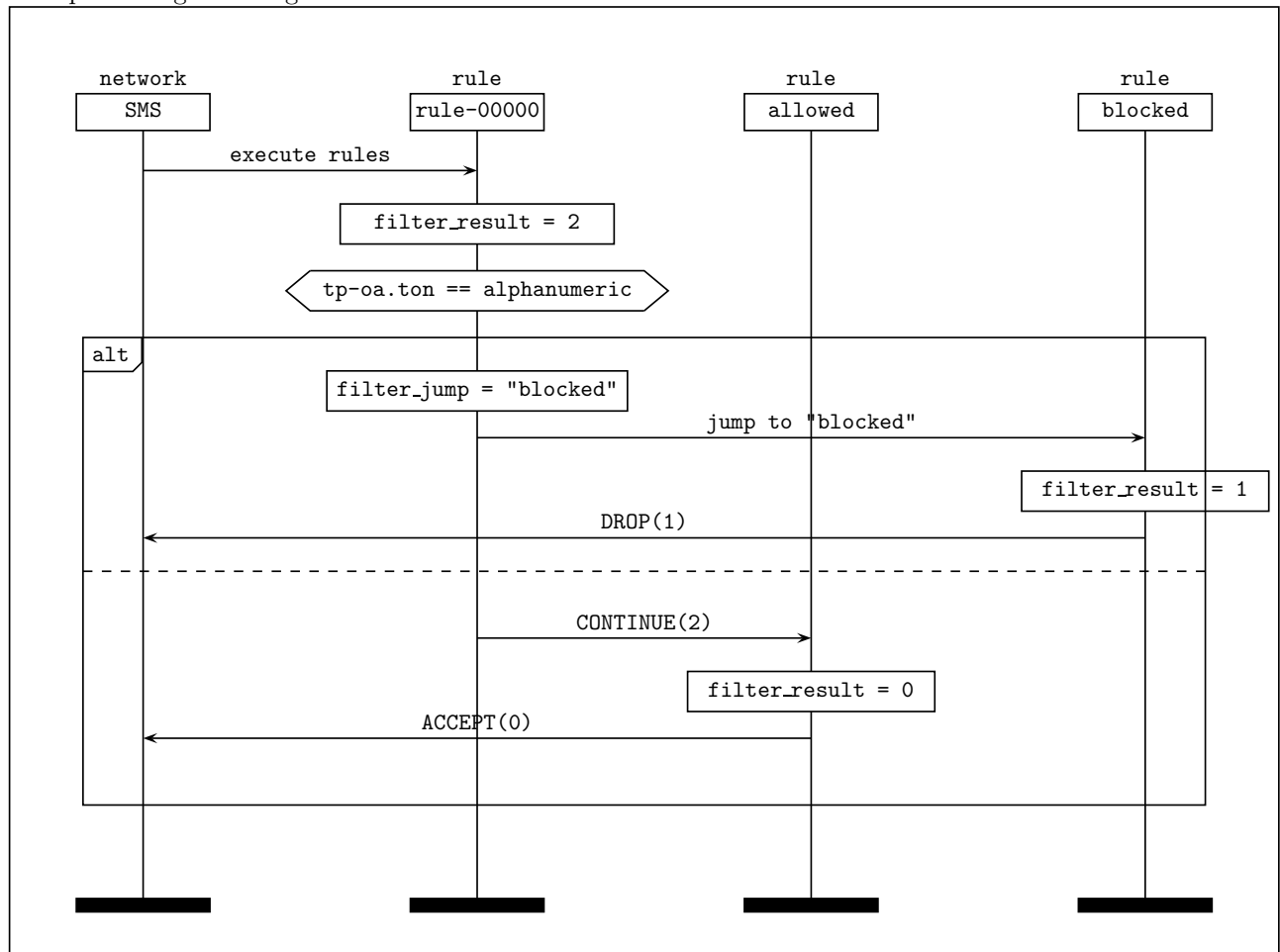
Detailed view of *rules/blocked* section:

```
config@hostname:/configuration/mno/fgn/fgn1/rules > show blocked
  description [ --- Blocked --- ] - Rule description
    priority [      100001 ] - Priority level
  filter_result [          1 ] - Default filtering result
    definition          - Rule configuration
```

Configuration items explained:

Item	Description
<b>rule-00000, allowed, blocked</b>	User defined dynamic block nodes, each represents a uniquely named rule
<b>definition, match, translate</b>	Grouping nodes, contain "match" and "translate" sections for each rule
<b>description</b>	Informative rule description
<b>priority</b>	Rule priority value dictates the order by which the rules are executed. Both Rule Processor and pMINK Command Line Interface keep the list of rules sorted by this value
<b>filter_result</b>	Rule exit code <sup>3</sup> value; it can be set for the first time or overridden if previously set
<b>filter_jump</b>	Rule name to jump to if normal rule execution flow should be overridden
<b>smstpdu tp-oa ton</b>	SMSTPDU TP-Originating-Address (TP-OA) Type Of Number (TON)

Rule processing flow diagram:



<sup>3</sup> Described in section 8.2

### 8.2.2 Rule match part

Available categories in rule "match" section:

```
dummy - Dummy field for generic scripting
r14p - R14P pMINK framework data
m3ua - MTP Level 3 (MTP3) User Adaptation Layer
sccp - Signalling Connection Control Part
tcap - Transaction Capabilities Application Part
map - Mobile Application Part
smstpd - Short message TPDU 3GPP TS 23.040
smpp - Short Message Peer-to-Peer
```

Configuration items explained:

Item	Description
<b>dummy</b>	Context free field without default value, used only for advanced inline or external scripting
<b>r14p</b>	Release14 Application Layer Protocol (R14P)
<b>m3ua</b>	MTP Level 3 (MTP3) User Adaptation Layer (M3UA)
<b>sccp</b>	Signalling Connection Control Part (SCCP)
<b>tcap</b>	Transaction Capabilities Application Part (TCAP)
<b>map</b>	Mobile Application Part (MAP)
<b>smstpd</b>	Short message TPDU 3GPP TS 23.040 (SMSTPDU)
<b>smpp</b>	Short Message Peer-to-Peer (SMPP)

#### 8.2.2.1 Release14 Application Layer Protocol (R14P) matching

R14P is a Release14 protocol, used for internal communication between various Project MINK Framework (pMINK) daemons. It is transferred via Stream Control Transmission Protocol (SCTP) as an X.690 encoded ASN.1 data.

Configuration items listed:

```
trunk_label - Trunk label
service_id - Service id
src_type - Source daemon type
src_id - Source daemon id
cmd_id - Command id
conn_type - Source connection type
loop_count - Loop count
```

Configuration items explained:

Item	Description
<b>trunk_label</b>	Alphanumeric value assigned by Signalling Transfer Point (STP), used for mobile network traffic identification
<b>service_id</b>	Type of R14P ServiceMessage, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 42 - sid-openli</li> <li>• 43 - sid-sms-data-retention</li> <li>• 44 - sid-stp-routing</li> <li>• 45 - sid-sgn-forward</li> <li>• 46 - sid-smshub-forward</li> <li>• 47 - sid-fgn-filtering</li> <li>• 48 - sid-security</li> <li>• 49 - sid-pdn-filtering</li> </ul>
<b>src_type</b>	Source daemon type; an alphanumeric field containing an R14P type of sender
<b>src_id</b>	Source daemon id; an alphanumeric field containing an R14P id of sender

Item	Description
<b>cmd_id</b>	Special Project MINK Framework (pMINK) command id; a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 1 - shci-sri-sm-req</li> <li>• 2 - shci-sri-sm-ack</li> <li>• 3 - shci-corr-ntf</li> <li>• 4 - shci-sms-ack</li> <li>• 5 - shci-sms-dlvr-rcpt</li> <li>• 6 - shci-smpp-generate-udh</li> <li>• 7 - shci-tcap-continue</li> </ul>
<b>conn_type</b>	External <sup>4</sup> connection type the current packet originated from; a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 0 - UNKNOWN</li> <li>• 1 - Stream Control Transmission Protocol (SCTP)</li> <li>• 2 - MTP Level 3 (MTP3) User Adaptation Layer (M3UA)</li> <li>• 3 - Project MINK Framework (pMINK)</li> <li>• 4 - Transmission Control Protocol (TCP)</li> <li>• 5 - Short Message Peer-to-Peer (SMPP)</li> </ul>
<b>loop_count</b>	Loop protection feature of Signalling Gateway Node (SGN); a numeric field indicating how many times the current packet has repeated itself

### 8.2.2.2 MTP Level 3 (MTP3) User Adaptation Layer (M3UA) matching

M3UA stands for MTP Level 3 (MTP3) User Adaptation Layer as defined by the IETF SIGTRAN working group in RFC 4666 (which replaces and supersedes RFC 3332). M3UA enables the SS7 protocol's User Parts (e.g. ISUP, SCCP and TUP) to run over IP instead of telephony equipment like ISDN and PSTN. It is recommended to use the services of SCTP to transmit M3UA.

Configuration items listed:

```
opc - Originating Point Code
dpc - Destination Point Code
si - Service indicator
ni - Network indicator
mp - Message priority
sls - Signalling link selection code
as - Application Server label
asp - Application Server Process label
```

Configuration items explained:

Item	Description
<b>opc, dpc</b>	The Originating and Destination Point Code fields contain the OPC and DPC from the routing label of the original SS7 message in Network Byte Order, justified to the least significant bit. Unused bits are coded '0'
<b>si</b>	The Service Indicator field contains the SI field from the original SS7 message justified to the least significant bit. Unused bits are coded '0'
<b>ni</b>	The Network Indicator contains the NI field from the original SS7 message justified to the least significant bit. Unused bits are coded '0'
<b>mp</b>	The Message Priority field contains the MP bits (if any) from the original SS7 message, both for ANSI-style and TTC-style [29] message priority bits. The MP bits are aligned to the least significant bit. Unused bits are coded '0'
<b>sls</b>	The Signalling Link Selection field contains the SLS bits from the routing label of the original SS7 message justified to the least significant bit and in Network Byte Order. Unused bits are coded '0'
<b>as</b>	Alphanumeric field containing an Application Server (AS) label the current packet originated from. The label was set in Signalling Gateway Node (SGN) runtime configuration
<b>asp</b>	Alphanumeric field containing an Application Server Process (ASP) label the current packet originated from. The label was set in Signalling Gateway Node (SGN) runtime configuration

<sup>4</sup> Described in section 8.1

### 8.2.2.3 Signalling Connection Control Part (SCCP) matching

The Signalling Connection Control Part (SCCP) is a network layer protocol that provides extended routing, flow control, segmentation, connection-orientation, and error correction facilities in Signaling System 7 telecommunications networks. SCCP relies on the services of MTP for basic routing and error detection.

The base SCCP specification is defined by the ITU-T, in recommendations Q.711 to Q.714, with additional information to implementors provided by Q.715 and Q.716. There are, however, regional variations defined by local standards bodies. In the United States, ANSI publishes its modifications to Q.713 as ANSI T1.112. The TTC publishes as JT-Q.711 to JT-Q.714, and Europe ETSI publishes ETSI EN 300-009-1: both of which document their modifications to the ITU-T specifications.

Although MTP provides routing capabilities based upon the Point Code, SCCP allows routing using a Point Code and Subsystem number or a Global Title. A Point Code is used to address a particular node on the network, whereas a Subsystem number addresses a specific application available on that node. SCCP employs a process called Global Title Translation to determine Point Codes from Global Titles so as to instruct MTP on where to route messages.

In the SIGTRAN suite of protocols, there are two primary methods of transporting SCCP applications across Internet Protocol networks: SCCP can be transported indirectly using the MTP level 3 User Adaptation protocol (M3UA), a protocol which provides support for users of MTP-3 including SCCP. Alternatively, SCCP applications can operate directly over the SCCP User Adaptation protocol (SUA) which is a form of modified SCCP designed specifically for use in IP networking. Signalling Gateway Node (SGN) uses the first method of SCCP transport; MTP Level 3 (MTP3) User Adaptation Layer (M3UA).

Configuration items listed:

```
cgpa - Calling Party
cdpa - Called Party
```

Configuration sub-items listed ("*cgpa/cdpa*" section):

```
routing-indicator - Routing indicator
    gti - Global Title Indicator
    ssn - SubSystem Number
    point-code - Point code
    gt - Global Title
```

Configuration sub-items listed ("*cgpa/cdpa gt*" section):

```
tt - Translation type
np - Numbering plan
nai - Nature Of Address
address - GT Address
```

Configuration items explained:

Item	Description
<b>cgpa, cdpa</b>	Grouping nodes, contain fields for " <i>Calling Party Address</i> " and " <i>Called Party Address</i> "
<b>routing-indicator</b>	Routing type, identifies which address element shall be used for routing: <ul style="list-style-type: none"> <li>• 0 - route on GT</li> <li>• 1 - route on SSN</li> </ul>
<b>gti</b>	Global Title Indicator field contains the type of Global Title included. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 0 - no global title included</li> <li>• 4 - global title includes nature of address indicator only</li> <li>• 8 - global title includes translation type only</li> <li>• 12 - global title includes translation type, numbering plan and encoding scheme</li> <li>• 16 - global title includes translation type, numbering plan, encoding scheme and nature of address indicator</li> </ul>



Item	Description
<b>ssn</b>	<p>The Subsystem Number (SSN) is a numerical fields identifying an SCCP user function. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - SSN not known/not used</li> <li>• 1 - SCCP management</li> <li>• 2 - reserved for ITU-T allocation</li> <li>• 3 - ISDN user part</li> <li>• 4 - OMAP (Operation, Maintenance and Administration Part)</li> <li>• 5 - MAP (Mobile Application Part)</li> <li>• 6 - HLR (Home Location Register)</li> <li>• 7 - VLR (Visitor Location Register)</li> <li>• 8 - MSC (Mobile Switching Centre)</li> <li>• 9 - EIC (Equipment Identifier Centre)</li> <li>• 10 - AUC (Authentication Centre)</li> <li>• 11 - ISDN supplementary services</li> <li>• 12 - reserved for international use</li> <li>• 13 - broadband ISDN edge-to-edge applications</li> <li>• 14 - TC test responder</li> </ul>
<b>point-code</b>	Signalling point code (numerical)
<b>gt</b>	Grouping node, contains fields for <i>"Global Title"</i>
<b>tt</b>	Translation type, a numerical value
<b>np</b>	<p>Numbering Plan, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - unknown</li> <li>• 16 - ISDN/telephony numbering plan (Recommendations E.163 and E.164)</li> <li>• 32 - generic numbering plan</li> <li>• 48 - data numbering plan (Recommendation X.121)</li> <li>• 64 - telex numbering plan (Recommendation F.69)</li> <li>• 80 - maritime mobile numbering plan (Recommendations E.210, E.211)</li> <li>• 96 - land mobile numbering plan (Recommendation E.212)</li> <li>• 112 - ISDN/mobile numbering plan (Recommendation E.214)</li> <li>• 224 - private network or network-specific numbering plan</li> </ul>
<b>nai</b>	<p>Nature Of Address, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - unknown</li> <li>• 1 - subscriber number</li> <li>• 2 - reserved for national use</li> <li>• 3 - national significant number</li> <li>• 4 - international number</li> </ul>
<b>address</b>	Address signals (usually digits)

#### 8.2.2.4 Transaction Capabilities Application Part (TCAP) matching

Transaction Capabilities Application Part, from ITU-T recommendations Q.771-Q.775 or ANSI T1.114 is a protocol for Signalling System 7 networks. Its primary purpose is to facilitate multiple concurrent dialogs between the same sub-systems on the same machines, using Transaction IDs to differentiate these, similar to the way TCP ports facilitate multiplexing connections between the same IP addresses on the Internet. TCAP is used to transport INAP in Intelligent Networks and MAP in mobile phone networks.

Configuration items listed:

<b>tcmt</b>	- TC message type
<b>sid</b>	- Source transaction id
<b>did</b>	- Destination transaction id
<b>cc</b>	- Component count
<b>ct</b>	- Component type
<b>iid</b>	- Component invoke id
<b>opcode</b>	- Component operation code
<b>dlg_ctx</b>	- Dialogue application context

Configuration items explained:

Item	Description
<b>tcmt</b>	TCMessage type, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 1 - Unidirectional</li> <li>• 2 - Begin</li> <li>• 4 - End</li> <li>• 5 - Continue</li> <li>• 7 - Abort</li> </ul>
<b>sid</b>	Originating Transaction ID, a numerical value present in <i>"Begin"</i> and <i>"Continue"</i> message types
<b>did</b>	Destination Transaction ID, a numerical value present in <i>"Continue"</i> , <i>"End"</i> and <i>"Abort"</i> message types
<b>cc</b>	ComponentPortion component count
<b>ct</b>	Component type, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 1 - Invoke</li> <li>• 2 - ReturnResultLast</li> <li>• 3 - ReturnError</li> <li>• 4 - Reject</li> <li>• 7 - ReturnResultNotLast</li> </ul>
<b>iid</b>	Invoke ID, a numerical TCAP reference for a specific TCAP operation
<b>opcode</b>	TCAP operation code, a numerical value
<b>dlg_ctx</b>	TCAP Dialogue Application Context OID

#### 8.2.2.5 Mobile Application Part (MAP) matching

The Mobile Application Part (MAP) is an SS7 protocol that provides an application layer for the various nodes in GSM and UMTS mobile core networks and GPRS core networks to communicate with each other in order to provide services to mobile phone users. The Mobile Application Part is the application-layer protocol used to access the Home Location Register, Visitor Location Register, Mobile Switching Center, Equipment Identity Register, Authentication Centre, Short message service center and Serving GPRS Support Node (SGSN).

The Mobile Application Part specifications were originally defined by the GSM Association, but are now controlled by ETSI/3GPP. MAP is defined by two different standards, depending upon the mobile network type:

- MAP for GSM (prior to Release 4) is specified by 3GPP TS 09.02 (MAP v1, MAP v2)
- MAP for UMTS ("3G") and GSM (Release 99 and later) is specified by 3GPP TS 29.002 (MAP v3)

In cellular networks based on ANSI standards (currently CDMA2000, in the past AMPS, IS-136 and cdmaOne) plays the role of the MAP a similar protocol usually called IS-41 or ANSI-41 (ANSI MAP). Since 2000 it is maintained by 3GPP2 as N.S0005 and since 2004 it is named 3GPP2 X.S0004. Project MINK Framework (pMINK) supports GSM MAP v1, v2 and v3.

Configuration items listed:

**context** - Mobile application component context

Configuration sub-items listed ("*context*" section):

**sri-for-sm** - Send routing info for short message  
**sm** - Short message

Configuration sub-items listed ("*context sri-for-sm/sm*" section):

**msisdn** - Mobile Station International Subscriber Directory Number  
**sca** - Service centre address  
**imsi** - International mobile Subscriber Identity  
**nnn** - Network node number  
**an** - Additional number

Configuration sub-items listed ("*context sri-for-sm/sm msisdn/sca/nnn/an/scda/scoa*" section):

**nai** - Nature of address indicator  
**np** - Numbering plan  
**address** - Address

Configuration items explained:

Item	Description
<b>context</b>	Grouping node, contains fields for " <i>sri-for-sm</i> " and " <i>sm</i> " contexts; " <i>sri-for-sm</i> " context is identified by <b>sendRoutingInfoForSM(45)</b> TCAP opcode value and " <i>sm</i> " context by both <b>mt-forwardSM(44)</b> and <b>mo-forwardSM(46)</b> TCAP opcode values
<b>sri-for-sm, sm</b>	Grouping nodes, contain fields for " <i>sri-for-sm</i> " and " <i>sm</i> " contexts
<b>msisdn, sca, nnn, an,</b>	Grouping nodes, contain fields for " <i>Mobile Station International Subscriber Directory Number</i> ", " <i>Service centre address</i> ", " <i>Network node number</i> " and " <i>Additional number</i> "
<b>scda, scoa</b>	Grouping nodes, contain fields for " <i>Service centre address DA</i> " and " <i>Service centre address OA</i> "
<b>imsi</b>	International mobile Subscriber Identity
<b>nai</b>	Nature Of Address Indicator (NAI), a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 0 - unknown</li> <li>• 16 - international number</li> <li>• 32 - national significant number</li> <li>• 48 - network specific number</li> <li>• 64 - subscriber number</li> <li>• 96 - abbreviated number</li> </ul>
<b>np</b>	Numbering Plan (NP) Indicator, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 0 - unknown</li> <li>• 1 - ISDN/Telephony Numbering Plan (Rec ITU-T E.164)</li> <li>• 3 - data numbering plan (ITU-T Rec X.121)</li> <li>• 4 - telex numbering plan (ITU-T Rec F.69)</li> <li>• 6 - land mobile numbering plan (ITU-T Rec E.212)</li> <li>• 8 - national numbering plan</li> <li>• 9 - private numbering plan</li> </ul>
<b>address</b>	Digits of an address, an alphanumeric field

### 8.2.2.6 Short message TPDU 3GPP TS 23.040 (SMSTPDU) matching

GSM 03.40 or 3GPP TS 23.040 is a mobile telephony standard describing the format of the Transfer Protocol Data Units (TPDU) of the Short Message Transfer Protocol (SM-TP) used in the GSM networks to carry Short Messages. This format is used throughout the whole transfer of the message in the GSM mobile network. In contrast, application servers use different protocols, like Short Message Peer-to-Peer or Universal Computer Protocol, to exchange messages between them and the Short message service centre.

GSM 03.40 is the original name of the standard. Since 1999 it is being developed by the 3GPP under the name 3GPP TS 23.040. However, the original name is often used to refer even to the 3GPP document.

The GSM 03.40 TPDU's are used to carry messages between the Mobile Station (MS) and Mobile Switching Centre (MSC) using the Short Message Relay Protocol (SM-RP), while between MSC and a Short Message Service Centre (SMSC), the TPDU's are carried as a parameter of a Mobile Application Part (MAP) package. Project MINK Framework (pMINK) support SMSTPDU carried as a MAP parameter.

Configuration items listed:

tp-rp	- Reply path indicator
tp-udhi	- TP-UD header indicator
tp-srr	- MS status report request
tp-vpf	- TP-VP field format
tp-rd	- Reject duplicates
tp-mti	- Message type indicator
tp-mr	- Message reference
tp-sri	- SME status report indicator
tp-mms	- More message to send indicator
tp-da	- Destination address
tp-oa	- Originating address
tp-pid	- Protocol identifier
tp-dcs	- Data coding scheme
tp-vp	- Validity period
tp-udl	- Length of user data TP-UD
tp-scts	- Service centre time stamp
ie-msg-id	- Concatenated short message reference number
ie-msg-parts	- Concatenated short message total parts
ie-msg-part	- Concatenated short message part number

Configuration sub-items listed ("*tp-da/tp-oa*" section):

ton	- Type of number
np	- Numbering plan
address	- Address

Configuration items explained:

Item	Description
<b>tp-rp</b>	<p>The TP-Reply-Path is a 1-bit field, located within bit no 7 of the first octet of both SMS-DELIVER and SMS-SUBMIT, and to be given the following values:</p> <ul style="list-style-type: none"> <li>• 0 - TP-Reply-Path parameter is not set in this SMS-SUBMIT/DELIVER</li> <li>• 1 - TP-Reply-Path parameter is set in this SMS-SUBMIT/DELIVER</li> </ul>
<b>tp-udhi</b>	<p>TP-UDHI has the following value:</p> <ul style="list-style-type: none"> <li>• 0 - The TP-UD field contains only the short message</li> <li>• 1 - The beginning of the TP-UD field contains a Header in addition to the short message</li> </ul>

Item	Description
<b>tp-srr</b>	<p>The TP-Status-Report-Request is a 1-bit field, located within bit no. 5 of the first octet of SMS-SUBMIT and SMS-COMMAND, and to be given the following values:</p> <ul style="list-style-type: none"> <li>• 0 - A status report is not requested</li> <li>• 1 - A status report is requested</li> </ul>
<b>tp-vpf</b>	<p>The TP-Validity-Period-Format is a 2-bit field, located within bit no 3 and 4 of the first octet of SMS-SUBMIT, and to be given the following values:</p> <ul style="list-style-type: none"> <li>• 0 - TP-VP field not present</li> <li>• 16 - TP-VP field present - relative format</li> <li>• 8 - TP-VP field present - enhanced format</li> <li>• 24 - TP-VP field present - absolute format</li> </ul>
<b>tp-rd</b>	<p>The TP-Reject-Duplicates is a 1 bit field located within bit 2 of the first octet of SMS-SUBMIT and has the following values:</p> <ul style="list-style-type: none"> <li>• 0 - Instruct the SC to accept an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as a previously submitted SM from the same OA.</li> <li>• 1 - Instruct the SC to reject an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as the previously submitted SM from the same OA. In this case the response returned by the SC is as specified in Short message TPDU 3GPP TS 23.040 (SMSTPDU), section 9.2.3.6</li> </ul>
<b>tp-mti</b>	<p>The TP-Message-Type-Indicator is a 2-bit field, located within bits no 0 and 1 of the first octet of all PDUs which can be given the following values:</p> <ul style="list-style-type: none"> <li>• 0 - SMS-DELIVER</li> <li>• 1 - SMS-DELIVER-REPORT</li> <li>• 2 - SMS-SUBMIT</li> <li>• 3 - SMS-SUBMIT-REPORT</li> <li>• 4 - SMS-STATUS-REPORT</li> <li>• 5 - SMS-COMMAND</li> </ul>
<b>tp-mr</b>	<p>The TP-Message-Reference field gives an integer representation of a reference number of the SMS-SUBMIT or SMS-COMMAND submitted to the SC by the MS. The MS increments TP-Message-Reference by 1 for each SMS-SUBMIT or SMS-COMMAND being submitted</p>
<b>tp-sri</b>	<p>The TP-Status-Report-Indication is a 1-bit field, located within bit no. 5 of the first octet of SMS-DELIVER, and to be given the following values:</p> <ul style="list-style-type: none"> <li>• 0 - A status report shall not be returned to the SME</li> <li>• 1 - A status report shall be returned to the SME</li> </ul>
<b>tp-mms</b>	<p>The TP-More-Messages-to-Send is a 1-bit field, located within bit no 2 of the first octet of SMS-DELIVER and SMS-STATUS-REPORT, and to be given the following values:</p> <ul style="list-style-type: none"> <li>• 0 - More messages are waiting for the MS in this SC</li> <li>• 1 - No more messages are waiting for the MS in this SC</li> </ul>
<b>tp-da, tp-oa tp-pid</b>	<p>Grouping nodes, contain fields for "<i>TP-Destination-Address</i>" and "<i>TP-Originating-Address</i>"</p> <p>The TP-Protocol-Identifier is the information element by which the SM-TL either refers to the higher layer protocol being used, or indicates interworking with a certain type of telematic device</p>
<b>tp-dcs</b>	<p>TP-Data-Coding-Scheme represents a character set being used for SMS text. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - GSM 7 bit default alphabet</li> <li>• 4 - 8 bit data</li> <li>• 8 - UCS2 (16bit)</li> </ul>

Item	Description
<b>tp-vp</b>	The TP-Validity-Period comprises 1 octet in integer representation, giving the length of the validity period, counted from when the SMS-SUBMIT is received by the SC. Project MINK Framework (pMINK) automatically converts all 3 types of validity period formats to UNIX timestamp. This field contains a maximum timestamp value for which the SMS is still considered valid
<b>tp-udl</b>	If the TP-User-Data is coded using the GSM 7 bit default alphabet, the TP-User-Data-Length field gives an integer representation of the number of septets within the TP-User-Data field to follow. If the 7bit default-alphabet extension mechanism is used within the TP-User-Data (see 3GPP TS 23.038 [9]), the actual number of characters in the message shall be less than the number of septets. If a TP-User-Data-Header field is present, then the TP-User-Data-Length value is the sum of the number of septets in the TP-User-Data-Header field (including any padding) and the number of septets in the TP-User-Data field which follows. If the TP-User-Data is coded using 8-bit data, the TP-User-Data-Length field gives an integer representation of the number of octets within the TP-User-Data field to follow. If a TP-User-Data-Header field is present, then the TP-User-Data-Length value is the sum of the number of octets in the TP-User-Data-Header field and the number of octets in the TP-User-Data field which follows. If the TP-User-Data is coded using UCS2 [24] data, the TP-User-Data-Length field gives an integer representation of the number of octets within the TP-User-Data field to follow. If a TP-User-Data-Header field is present, then the TP-User-Data-Length value is the sum of the number of octets in the TP-User-Data-Header field and the number of octets in the TP-User-Data field which follows.
<b>tp-scts</b>	TP-Service-Centre-Time-Stamp converted to UNIX timestamp
<b>ie-msg-id</b>	Concatenated short message reference number
<b>ie-msg-parts</b>	Maximum number of short messages in the concatenated short message
<b>ie-msg-part</b>	Sequence number of the current short message within the concatenated short message
<b>ton</b>	Type Of Number (TON), a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 0 - Unknown</li> <li>• 16 - International number</li> <li>• 32 - National number</li> <li>• 48 - Network specific number</li> <li>• 64 - Subscriber number</li> <li>• 80 - Alphanumeric, (coded according to 3GPP TS 23.038 [9] GSM 7-bit default alphabet)</li> <li>• 96 - Abbreviated number</li> <li>• 112 - Reserved for extension</li> </ul>
<b>np</b>	Numbering Plan (NP), a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 0 - Unknown</li> <li>• 1 - ISDN/telephone numbering plan (E.164 [17]/E.163[18])</li> <li>• 3 - Data numbering plan (X.121)</li> <li>• 4 - Telex numbering plan</li> <li>• 6 - Land mobile numbering plan</li> <li>• 8 - National numbering plan</li> <li>• 9 - Private numbering plan</li> <li>• 10 - ERMES numbering plan (ETSI DE/PS 3 01-3)</li> <li>• 15 - Reserved for extension</li> </ul>
<b>address</b>	Address digits

### 8.2.2.7 Short Message Peer-to-Peer (SMPP) matching

The Short Message Peer-to-Peer (SMPP) in the telecommunications industry is an open, industry standard protocol designed to provide a flexible data communication interface for the transfer of short message data between External Short Messaging Entities (ESME), Routing Entities (RE) and Message Centres.

SMPP is often used to allow third parties (e.g. value-added service providers like news organizations) to submit messages, often in bulk, but it may be used for SMS peering as well. SMPP is able to carry short messages including EMS, Voice Mail notifications, Cell Broadcasts, WAP messages including WAP Push messages (used to deliver MMS notifications), USSD messages and others. Because of its versatility and support for non-GSM SMS protocols, like UMTS, IS-95 (CDMA), CDMA2000, ANSI-136 (TDMA) and iDEN, the SMPP is the most commonly used protocol for short message exchange outside SS7 networks.

Project MINK Framework (pMINK) uses Signalling Gateway Node (SGN) as the main entry point and external<sup>5</sup> signalling converter. Short Message Peer-to-Peer (SMPP) connections follow the same organizational principles as MTP Level 3 (MTP3) User Adaptation Layer (M3UA) connections by using both Application Server (AS) logical entities and Application Server Process (ASP) instances.

Configuration items listed:

```

as - Application Server label
asp - Application Server Process label
command_id - SMPP PDU message type
source_addr_ton - Address type of number
dest_addr_ton - Address type of number
source_addr_npi - Address numbering plan indicator
dest_addr_npi - Address numbering plan indicator
source_addr - SME originating address
destination_addr - SME destination address
esm_class_mm - ESM class message mode
esm_class_mt - ESM class message type
esm_class_gsm - ESM class GSM network specific features
protocol_id - Protocol identifier according to GSM 03.40
priority_flag - Short message priority level
delivery_time - Scheduled time at which the message delivery should be first attempted
validity_period - Validity period
rd_smsc_dlvr_rcpt - SMSC Delivery Receipt
rd_sme_orig_ack - SME originated Acknowledgement
rd_intrmd_ntf - Intermediate Notification
replace_if_present_flag - Request SMSC to replace a previously submitted message
data_coding - Short message data coding
sm_default_msg_id - SMSC index of a pre-defined (canned) message
sm_length - Length of short_message parameter in octets
sar_msg_ref_num - Reference number for a particular concatenated short message
sar_total_segments - Total number of short messages within the concatenated short message
sar_segment_seqnum - Sequence number of a particular short message within the concatenated short
↪ message

```

Configuration items explained:

Item	Description
<b>as</b>	Alphanumeric field containing an Application Server (AS) label the current packet originated from. The label was set in Signalling Gateway Node (SGN) runtime configuration
<b>asp</b>	Alphanumeric field containing an Application Server Process (ASP) label the current packet originated from.

<sup>5</sup> Described in section 8.1

Item	Description
<b>command_id</b>	<p>The command_id field identifies the type of message the SMPP PDU represents, for example, submit_sm, query_sm etc. A command identifier is allocated to each SMPP request primitive. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <p>2147483648 GENERIC_NACK</p> <p>1 BIND Operation - BIND_RECEIVER</p> <p>2147483649 BIND Operation - BIND_RECEIVER_RESP</p> <p>2 BIND Operation - BIND_TRANSMITTER</p> <p>2147483650 BIND Operation - BIND_TRANSMITTER_RESP</p> <p>3 QUERY_SM Operation - QUERY_SM</p> <p>2147483651 QUERY_SM Operation - QUERY_SM_RESP</p> <p>4 SUBMIT_SM Operation - SUBMIT_SM</p> <p>2147483652 SUBMIT_SM Operation - SUBMIT_SM_RESP</p> <p>5 DELIVER_SM Operation - DELIVER_SM</p> <p>2147483653 DELIVER_SM Operation - DELIVER_SM_RESP</p> <p>6 UNBIND Operation - UNBIND</p> <p>2147483654 UNBIND Operation - UNBIND_RESP</p> <p>7 REPLACE_SM Operation - REPLACE_SM</p> <p>2147483655 REPLACE_SM Operation - REPLACE_SM_RESP</p> <p>8 CANCEL_SM Operation - CANCEL_SM</p> <p>2147483656 CANCEL_SM Operation - CANCEL_SM_RESP</p> <p>9 BIND Operation - BIND_TRANSCEIVER</p> <p>2147483657 BIND Operation - BIND_TRANSCEIVER_RESP</p> <p>11 OUTBIND Operation - OUTBIND</p> <p>21 ENQUIRE_LINK Operation - ENQUIRE_LINK</p> <p>2147483669 ENQUIRE_LINK Operation - ENQUIRE_LINK_RESP</p> <p>33 SUBMIT_MULTI Operation - SUBMIT_MULTI</p> <p>2147483681 SUBMIT_MULTI Operation - SUBMIT_MULTI_RESP</p> <p>258 ALERT_NOTIFICATION Operation - ALERT_NOTIFICATION</p> <p>259 DATA_SM Operation - DATA_SM</p> <p>2147483907 DATA_SM Operation - DATA_SM_RESP</p>
<b>source_addr_ton</b> <b>dest_addr_ton</b>	<p>Type Of Number (TON), a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - Unknown</li> <li>• 1 - International</li> <li>• 2 - National</li> <li>• 3 - Network Specific</li> <li>• 4 - Subscriber Number</li> <li>• 5 - Alphanumeric</li> <li>• 6 - Abbreviated</li> </ul>
<b>source_addr_npi</b> <b>dest_addr_npi</b>	<p>Numbering Plan (NP) Indicator, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - Unknown</li> <li>• 1 - ISDN (E163/E164)</li> <li>• 3 - Data (X.121)</li> <li>• 4 - Telex (F.69)</li> <li>• 6 - Land Mobile (E.212)</li> <li>• 8 - National</li> <li>• 9 - Private</li> <li>• 10 - ERMES</li> <li>• 14 - Internet (IP)</li> <li>• 18 - WAP Client Id (to be defined by WAP Forum)</li> </ul>



Item	Description
<b>source_addr</b>	Specifies the address of SME which originated this message. An ESME which is implemented as a single SME address, may set this field to NULL to allow the SMSC to default the source address of the submitted message.
<b>destination_addr</b>	Specifies the destination SME address. For mobile terminated messages, this is the directory number of the recipient MS.
<b>esm_class_mm</b>	<p>The esm_class_mm parameter is used to indicate special message mode attribute associated with the short message. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - Default SMSC Mode (e.g. Store and Forward)</li> <li>• 1 - Datagram mode</li> <li>• 2 - Forward (i.e. Transaction) mode</li> <li>• 3 - Store and Forward mode (use to select Store and Forward mode if Default SMSC Mode is non Store and Forward)</li> </ul>
<b>esm_class_mt</b>	<p>The esm_class_mt parameter is used to indicate special message type attribute associated with the short message. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - Default message Type (i.e. normal message)</li> <li>• 4 - Short Message contains SMSC Delivery Receipt</li> <li>• 8 - Short Message contains ESME Delivery Acknowledgement</li> <li>• 16 - Short Message contains ESME Manual/User Acknowledgement</li> <li>• 24 - Short Message contains Conversation Abort (Korean CDMA)</li> <li>• 32 - Short Message contains Intermediate Delivery Notification</li> </ul>
<b>esm_class_gsm</b>	<p>The esm_class_gsm parameter is used to indicate special GSM attribute associated with the short message. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - No specific features selected</li> <li>• 64 - UDHI Indicator (only relevant for MT short messages)</li> <li>• 128 - Set Reply Path (only relevant for GSM network)</li> <li>• 192 - Set UDHI and Reply Path (only relevant for GSM network)</li> </ul>
<b>protocol_id</b>	A numerical value set according to GSM 03.40
<b>priority_flag</b>	<p>The priority_flag parameter allows the originating SME to assign a priority level to the short message. Four priority levels are supported; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - Level 0 (lowest) priority</li> <li>• 1 - Level 1 priority</li> <li>• 2 - Level 2 priority</li> <li>• 3 - Level 3 (highest) priority</li> </ul>
<b>delivery_time</b>	<p>This parameter specifies the scheduled time at which the message delivery should be first attempted. It defines either the absolute date and time or relative time from the current SMSC time at which delivery of this message will be attempted by the SMSC. This is a numerical field; Project MINK Framework (pMINK) converts both Absolute and Relative time formats of scheduled_delivery_time to UNIX timestamp.</p>
<b>validity_period</b>	<p>The validity_period parameter indicates the SMSC expiration time, after which the message should be discarded if not delivered to the destination. This is a numerical field; Project MINK Framework (pMINK) converts both Absolute and Relative time formats of validity_period to UNIX timestamp</p>

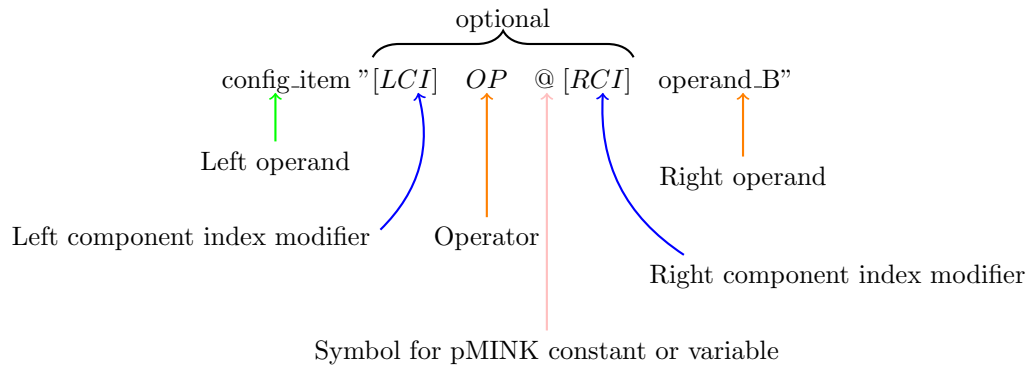
Item	Description
<b>rd_smsc_dlvr_rcpt</b>	<p>The rd_smsc_dlvr_rcpt parameter is used to request an SMSC delivery receipt. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - No SMSC Delivery Receipt requested (default)</li> <li>• 1 - SMSC Delivery Receipt requested where final delivery outcome is delivery success or failure</li> <li>• 2 - SMSC Delivery Receipt requested where the final delivery outcome is delivery failure</li> </ul>
<b>rd_sme_orig_ack</b>	<p>The rd_sme_orig_ack parameter is used to request an SME originated acknowledgements. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - No recipient SME acknowledgment requested (default)</li> <li>• 4 - SME Delivery Acknowledgement requested</li> <li>• 8 - SME Manual/User Acknowledgment requested</li> <li>• 12 - Both Delivery and Manual/User Acknowledgment requested</li> </ul>
<b>rd_intrmd_ntf</b>	<p>The rd_intrmd_ntf parameter is used to request an SMSC intermediate notification. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - No Intermediate notification requested (default)</li> <li>• 16 - Intermediate notification requested</li> </ul>
<b>replace_if_present_flag</b>	<p>The replace_if_present_flag parameter is used to request the SMSC to replace a previously submitted message, that is still pending delivery. The SMSC will replace an existing message provided that the source address, destination address and service_type match the same fields in the new message. The following values are supported;</p> <ul style="list-style-type: none"> <li>• 0 - Don't replace (default)</li> <li>• 1 - Replace</li> </ul>
<b>data_coding</b>	<p>SMS text data coding scheme, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - SMSC Default Alphabet</li> <li>• 1 - IA5 (CCITT T.50)/ASCII (ANSI X3.4)</li> <li>• 2 - Octet unspecified (8-bit binary)</li> <li>• 3 - Latin 1 (ISO-8859-1)</li> <li>• 4 - Octet unspecified (8-bit binary)</li> <li>• 5 - JIS (X 0208-1990)</li> <li>• 6 - Cyrillic (ISO-8859-5)</li> <li>• 7 - Latin/Hebrew (ISO-8859-8)</li> <li>• 8 - UCS2 (ISO/IEC-10646)</li> <li>• 9 - Pictogram Encoding</li> <li>• 10 - ISO-2022-JP (Music Codes)</li> <li>• 13 - Extended Kanji JIS(X 0212-1990)</li> <li>• 14 - KS C 5601</li> </ul>
<b>sm_default_msg_id</b>	<p>The sm_default_msg_id parameter specifies the SMSC index of a pre-defined (canned) message</p>
<b>sm_length</b>	<p>The sm_length parameter specifies the length of the short_message parameter in octets. The sm_length should be set to 0 in the submit_sm, submit_multi, and deliver_sm PDUs if the message_payload parameter is being used to send user data larger than 254 octets.</p>

Item	Description
<b>sar_msg_ref_num</b>	The sar_msg_ref_num parameter is used to indicate the reference number for a particular concatenated short message.
<b>sar_total_segments</b>	The sar_total_segments parameter is used to indicate the total number of short messages within the concatenated short message.
<b>sar_segment_seqnum</b>	The sar_segment_seqnum parameter is used to indicate the sequence number of a particular short message within the concatenated short message.

### 8.2.3 Basic matching

Filtering Gateway Node (FGN) Rule Processing Engine (RPE) supports two levels of parameter matching; *basic* and *advanced*. Basic matching, implemented as an extension of a standard set of boolean operators, is further extended with Perl compatible regular expressions and pMINK specific field modifiers.

#### 8.2.3.1 Basic matching syntax



The syntax explained in this chapter focuses on the lowest configuration level; a configuration item ("*config\_item*") represents an external<sup>6</sup> or internal<sup>6</sup> signalling data available for matching. Data matching is a process of comparing a relation between two operands (left and right); in case of basic matching, the first operand (left) is not under user control and always points to a runtime configuration item ("*config\_item*").

The second operand (right), displayed as "*operand\_B*" in this diagram, can be accessed and maintained through pMINK Command Line Interface (CLI), and is entirely under user control. Several different operand types are supported; their purpose and format will be covered in the following chapters.

The relationship between two operands is examined by creating a relational expression whose result evaluates to "**true**" or "**false**", depending on operator "*OP*". This evaluation result is used by Rule Processor (RPROC)<sup>7</sup> to determine the next step in rule execution flow. Basic syntax, used in "*match*" section of runtime configuration, features two extra components; left and right component index modifiers ("*LCI*" and "*RCI*").

Protocol Data Units sometimes contain multiple instances of various parameters; Signalling Gateway Node (SGN) generates numerical zero-based indexes to differentiate between those parameters. Left Component Index (LCI), used for selecting a specific instance of runtime configuration item ("*config\_item*"), uses zero-based indexes received from SGN. Both left and right component index modifiers operate on the same set of indexes; the only difference between the two is that Right Component Index (RCI) is used exclusively with predefined pMINK variables.

Most components of basic syntax are optional; if they are not used, default values are automatically selected.

Basic syntax component	Default value
Left Component Index (LCI)	0(zero)
Operator " <i>OP</i> "	==(equality)
Right Component Index (RCI)	0(zero)

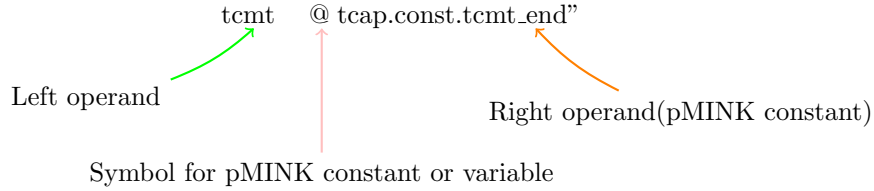
<sup>6</sup> Described in section 8.1

<sup>7</sup> Described in section 8.2

Example rule ("match tcap" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/tcap > configuration
tcmt      "@tcap.const.tcmt_end"
ct        "[0]!@tcap.const.ct_error"
```

Explanation of "match tcap tcmt" field:

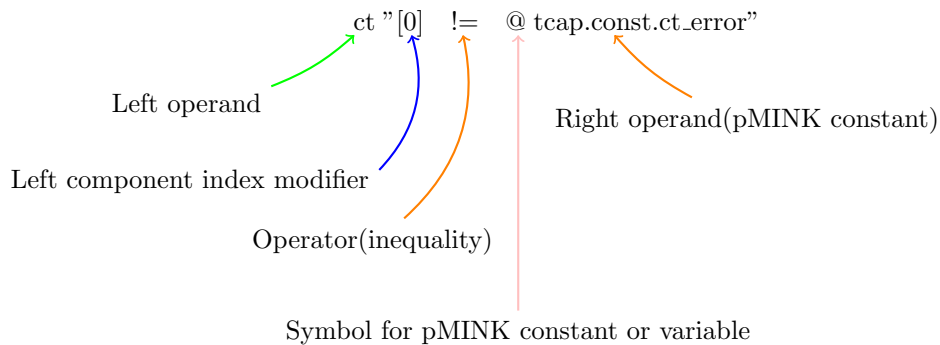


Basic syntax component	Value
Left operand	tcmt
Left Component Index (LCI)	0(zero)
Operator "OP"	==(equality)
pMINK variable/constant	yes
Right Component Index (RCI)	0(zero)
Right operand	tcap.const.tcmt_end

Description:

Check if Transaction Capabilities Application Part (TCAP) TCMMessage is of type "End"

Explanation of "match tcap ct" field:



Basic syntax component	Value
Left operand	ct
Left Component Index (LCI)	0(zero)
Operator "OP"	!=(inequality)
pMINK variable/constant	yes
Right Component Index (RCI)	0(zero)
Right operand	tcap.const.ct_error

Description:

Check if Transaction Capabilities Application Part (TCAP) component at index 0 is anything other than ReturnError

### 8.2.3.2 Operand types

Project MINK Framework (pMINK) basic syntax supports 6 types of operands; all of them are supported in "match"<sup>8</sup> section and only some in "translate"<sup>8</sup> section.

**8.2.3.2.1 NUMBER type operand** is a numerical operand consisting of one or more digits.

Example rule ("match tcap" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/tcap > configuration
tcmt      "4"
```

Description:

Check if Transaction Capabilities Application Part (TCAP) TCMessage is of type associated with number 4

**8.2.3.2.2 STRING type operand** is a sequence of characters enclosed in matching single quotes(')<sup>9</sup>.

Example rule ("match smstpdu section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/smstpdu > configuration
tp-oa {
    address "'38670007007'"
}
```

Description:

Check if Short message TPDU 3GPP TS 23.040 (SMSTPDU) TP-Originating-Address (TP-OA) address digits match "38670007007"

**8.2.3.2.3 REGEX type operand** is a Perl compatible Regular expression indicated by a sequence of characters starting with a colon(:) sign; more detailed information can be found on the following address: <http://perldoc.perl.org/perlre.html#Regular-Expressions>

Example rule ("match m3ua" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/m3ua > configuration
opc      ":(?=^..25).*"
```

Description:

Check if MTP Level 3 (MTP3) User Adaptation Layer (M3UA) Originating Point Code starts with any two characters followed by "25"

**Note:** When using REGEX type operand, Right Component Index (RCI) and operator "OP" are not supported; regular expression always operates on left operand ("config\_item") and evaluates to "true" if match is found.

**8.2.3.2.4 VARIABLE type operand** if a special pMINK predefined variable or constant, indicated by a sequence of characters starting with at(@) sign. A list of supported values can be found in section 8.2.3.4.

Example rule ("match m3ua" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/smstpdu > configuration
tp-oa {
    ton      "@smstpdu.const.noa_alphanumeric"
}
```

Description:

Check if Short message TPDU 3GPP TS 23.040 (SMSTPDU) TP-Originating-Address (TP-OA) Type Of Number (TON) is alphanumeric

<sup>8</sup> Described in section 8.2

<sup>9</sup> String operands be used, although not recommended, without single quotes

**8.2.3.2.5 LIST type operand** is a special pMINK operand used for referencing two types of lists; *"static"*<sup>10</sup> and *"dynamic"*<sup>11</sup>. It is implemented as an extension of variable type operand; a list is referenced by using an `at(@)` sign followed by sequence of characters enclosed in curly brackets.

Example rule (*"match sccp"* section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/sccp > configuration
cgpa {
    gt {
        address "@{gt_black_list}"
    }
}
```

Description:

Check if Signalling Connection Control Part (SCCP) Calling Party Global Title address exists in a list named `"gt_black_list"`

**8.2.3.2.6 LUA type operand** is an external scripting feature used for extending basic syntax and features of Filtering Gateway Node (FGN); Project MINK Framework (pMINK) uses Lua programming language as a powerful and simple solution for FGN customization. Lua type operand is indicated by two consecutive grave accent (`) signs followed by inline scripting block or absolute path to external script. This topic will be discussed in more detail starting from section 8.2.4.

Example rule (*"match sccp"* section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/sccp > configuration
cgpa {
    gt {
        address "``return F.sl_get('gt_black_list', F.vpval())"
    }
}
```

Description:

Check if Signalling Connection Control Part (SCCP) Calling Party Global Title address exists in a list named `"gt_black_list"`

### 8.2.3.3 Operator types

Project MINK Framework (pMINK) supports 6 relational and 3 content-changing operators; the latter provide a basic syntax interface to advanced list manipulation techniques.

**8.2.3.3.1 Equality '==' operator** evaluates to *"true"* if both left and right operands are equal.

Example rule (*"match map"* section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/map > configuration
context {
    sm {
        scda {
            nai      "==@map.const.noa_international"
        }
    }
}
```

Description:

Check if Short Message Service (SMS) ServiceCentreAddressDA Nature Of Address Indicator (NAI) is set to *"international"*

<sup>10</sup> Described in section 8.1.3.3

<sup>11</sup> Described in section 8.2.4.3.2

**8.2.3.3.2 Inequality '!=' operator** evaluates to **"true"** if both left and right operands are different.

Example rule ("match map" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/map > configuration
context {
    sm {
        scda {
            np      "!=@map.const.np_national"
        }
    }
}
```

Description:

Check if Short Message Service (SMS) ServiceCentreAddressDA Numbering Plan (NP) is anything other than national"

**8.2.3.3.3 Greater than '>' operator** evaluates to **"true"** if left operand is greater than the right operand.

Example rule ("match m3ua" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/m3ua > configuration
dpc ">5000"
```

Description:

Check if MTP Level 3 (MTP3) User Adaptation Layer (M3UA) Destination Point Code is greater than 5000"

**8.2.3.3.4 Greater than equal '>=' operator** evaluates to **"true"** if left operand is greater than or equal to the right operand.

Example rule ("match m3ua" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/m3ua > configuration
opc ">=4000"
```

Description:

Check if MTP Level 3 (MTP3) User Adaptation Layer (M3UA) Originating Point Code is greater than or equal to 4000"

**8.2.3.3.5 Less than '<' operator** evaluates to **"true"** if left operand is less than the right operand.

Example rule ("match m3ua" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/m3ua > configuration
dpc "<5000"
```

Description:

Check if MTP Level 3 (MTP3) User Adaptation Layer (M3UA) Destination Point Code is less than 5000"

**8.2.3.3.6 Less than equal '<=' operator** evaluates to **"true"** if left operand is less than or equal to the right operand.

Example rule ("match m3ua" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/m3ua > configuration
opc "<=4000"
```

Description:

Check if MTP Level 3 (MTP3) User Adaptation Layer (M3UA) Originating Point Code is less than or equal to 4000"

**8.2.3.3.7 Add to list '>>' operator** is a special type of operator used exclusively for adding data to both static and dynamic lists; data contained in left operand is added to the listed specified by right operand. The resulting expression will always evaluate to **"true"**.

Example rule ("match sccp" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/sccp > configuration
cdpa {
    gt {
        address ">>@{gt_black_list}"
    }
}
```

Description:

Add Signalling Connection Control Part (SCCP) Called Party Global Title address to a list named "gt\_black\_list"

**8.2.3.3.8 Remove from list '<<' operator** is another special type of operator used exclusively for removing data from both static and dynamic lists; data contained in left operand is removed from the listed specified by right operand. The resulting expression will always evaluate to **"true"**.

Example rule ("match sccp" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/sccp > configuration
cdpa {
    gt {
        address "<<@{gt_black_list}"
    }
}
```

Description:

Remove Signalling Connection Control Part (SCCP) Called Party Global Title address from a list named "gt\_black\_list"

**8.2.3.3.9 Remove list '—' operator** is the last special type of operator used exclusively for removing both static and dynamic lists; a list specified by right operand is cleared and removed. The resulting expression will always evaluate to **"true"**.

Example rule ("match" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match > configuration
dummy "--@{gt_black_list}"
```

Description:

Remove a list named "gt\_black\_list"

**Note: "dummy"** is a special runtime configuration field used in cases like this one, when left operand is not needed or ignored

#### 8.2.3.4 Predefined constants and variables

Project MINK Framework (pMINK) features a set of predefined constants and variables to provide a more human friendly way of referencing certain protocol fields and/or parameters.



pMINK variable	Value
Project MINK Framework (pMINK)	
pmink.timestamp	Current UNIX timestamp
Short Message Peer-to-Peer (SMPP)	
smpp.command_id	SMPP PDU message type
smpp.source_addr_ton	Source address type of number
smpp.dest_addr_ton	Destination address type of number
smpp.source_addr_npi	Source address numbering plan indicator
smpp.dest_addr_npi	Destination address numbering plan indicator
smpp.source_addr	SME originating address
smpp.destination_addr	SME destination address
smpp.esm_class_mm	ESM class message mode
smpp.esm_class_mt	ESM class message type
smpp.esm_class_gsm	ESM class GSM network specific features
smpp.protocol_id	Protocol identifier according to GSM 03.40
smpp.priority_flag	Short message priority level
smpp.delivery_time	Scheduled time at which the delivery should be first attempted
smpp.validity_period	Validity period
smpp.rd_smsc_dlv_rcpt	SMSC Delivery Receipt
smpp.rd_sme_orig_ack	SME originated Acknowledgement
smpp.rd_intrmd_ntf	Intermediate Notification
smpp.replace_if_present_flag	Request SMSC to replace a previously submitted message
smpp.data_coding	Short message data coding
smpp.sm_default_msg_id	SMSC index of a pre-defined (canned) message
smpp.sm_length	Length of short_message parameter in octets
smpp.sar_msg_ref_num	Reference number for a particular concatenated short message
smpp.sar_total_segments	Number of messages within the concatenated message
smpp.sar_segment_seqnum	Message sequence number within the concatenated message
Transaction Capabilities Application Part (TCAP)	
tcap.tcmt	TCAP message type
tcap.sid	Source transaction id
tcap.did	Destination transaction id
tcap.cc	Component count
tcap.ct	Component type
tcap.iid	Component invoke id
tcap.opcode	Component operation code
tcap.dlg_ctx	Dialogue application context
Mobile Application Part (MAP)	
map.imsi	International mobile Subscriber Identity
map.msisdn.nai	MSISDN Nature of address indicator
map.msisdn.np	MSISDN Numbering plan
map.msisdn.address	MSISDN address digits
map.sca.nai	Service centre address Nature of address indicator
map.sca.np	Service centre address Numbering plan
map.sca.address	Service centre address digits
map.scoa.nai	Service centre address OA Nature of address indicator
map.scoa.np	Service centre address OA Numbering plan
map.scoa.address	Service centre address OA digits
map.scda.nai	Service centre address DA Nature of address indicator
map.scda.np	Service centre address DA Numbering plan
map.scda.address	Service centre address DA digits
map.nnn.nai	Network node number Nature of address indicator
map.nnn.np	Network node number Numbering plan
map.nnn.address	Network node number digits
map.an.nai	Additional number Nature of address indicator
map.an.np	Additional number Numbering plan
map.an.address	Additional number digits

Short message TPDU 3GPP TS 23.040 (SMSTPDU)	
smstpdu.tp-rp	Reply path indicator
smstpdu.tp-udhi	TP-UD header indicator
smstpdu.tp-srr	MS status report request
smstpdu.tp-vpf	TP-VP field format
smstpdu.tp-rd	Reject duplicates
smstpdu.tp-mti	Message type indicator
smstpdu.tp-mr	Message reference
smstpdu.tp-sri	SME status report indicator
smstpdu.tp-mms	More message to send indicator
smstpdu.tp-da.ton	Destination address Type Of Number
smstpdu.tp-da.np	Destination address Numbering Plan
smstpdu.tp-da.address	Destination address digits
smstpdu.tp-oa.ton	Originating address Type Of Number
smstpdu.tp-oa.np	Originating address Numbering Plan
smstpdu.tp-oa.address	Originating address digits
smstpdu.tp-pid	Protocol identifier
smstpdu.tp-dcs	Data coding scheme
smstpdu.tp-vp	Validity period
smstpdu.tp-udl	Length of user data TP-UD
smstpdu.tp-scts	Service centre time stamp
smstpdu.ie.msg_id	Concatenated short message reference number
smstpdu.ie.msg_parts	Concatenated short message total parts
smstpdu.ie.msg_part	Concatenated short message part number
MTP Level 3 (MTP3) User Adaptation Layer (M3UA)	
m3ua.opc	Originating Point Code
m3ua.dpc	Destination Point Code
m3ua.si	Service indicator
m3ua.ni	Network indicator
m3ua.mp	Message priority
m3ua.sls	Signalling link selection code
m3ua.as	Application Server label
m3ua.asp	Application Server Process label
Signalling Connection Control Part (SCCP)	
sccp.cgpa.routing-indicator	Calling Party Address Routing indicator
sccp.cgpa.gti	Calling Party Address Global Title Indicator
sccp.cgpa.ssn	Calling Party Address SubSystem Number
sccp.cgpa.point-code	Calling Party Address Point code
sccp.cgpa.gt.tt	Calling Party Address Global Title Translation type
sccp.cgpa.gt.np	Calling Party Address Global Title Numbering plan
sccp.cgpa.gt.nai	Calling Party Address Global Title Nature Of Address
sccp.cgpa.gt.address	Calling Party Address Global Title digits
sccp.cdpa.routing-indicator	Called Party Address Routing indicator
sccp.cdpa.gti	Called Party Address Global Title Indicator
sccp.cdpa.ssn	Called Party Address SubSystem Number
sccp.cdpa.point-code	Called Party Address Point code
sccp.cdpa.gt.tt	Called Party Address Global Title Translation type
sccp.cdpa.gt.np	Called Party Address Global Title Numbering plan
sccp.cdpa.gt.nai	Called Party Address Global Title Nature Of Address
sccp.cdpa.gt.address	Called Party Address Global Title digits

pMINK constant	Value
Release14 Application Layer Protocol (R14P)	
r14p.const.srvcid_openli	Service id 42 - sid-openli
r14p.const.srvcid_sms_dr	Service id 43 - sid-sms-data-retention
r14p.const.srvcid_stp_routing	Service id 44 - sid-stp-routing
r14p.const.srvcid_sgn_fwd	Service id 45 - sid-sgn-forward
r14p.const.srvcid_smshub_fwd	Service id 46 - sid-smshub-forward
r14p.const.srvcid_fgn_filtering	Service id 47 - sid-fgn-filtering
r14p.const.srvcid_security	Service id 48 - sid-security
r14p.const.srvcid_pdn_filtering	Service id 49 - sid-pdn-filtering
r14p.const.cmdid_srism_req	Command id 1 - shci-sri-sm-req
r14p.const.cmdid_srism_ack	Command id 2 - shci-sri-sm-ack
r14p.const.cmdid_corr_ntf	Command id 3 - shci-corr-ntf
r14p.const.cmdid_sms_ack	Command id 4 - shci-sms-ack
r14p.const.cmdid_sms_dlvr_rcpt	Command id 5 - shci-sms-dlvr-rcpt
r14p.const.cmdid_smpp_generate_udh	Command id 6 - shci-smpp-generate-udh
r14p.const.cmdid_tcap_continue	Command id 7 - shci-tcap-continue
r14p.const.connt_sctp	Source connection type 1 - SCTP
r14p.const.connt_m3ua	Source connection type 2 - M3UA
r14p.const.connt_tcp	Source connection type 4 - TCP
r14p.const.connt_smpp	Source connection type 5 - SMPP
Short Message Peer-to-Peer (SMPP)	
smpp.const.command_generic_nack	PDU type 0x80000000 - GENERIC_NACK
smpp.const.command_bind_receiver	PDU type 0x00000001 - BIND_RECEIVER
smpp.const.command_bind_receiver_resp	PDU type 0x80000001 - BIND_RECEIVER_RESP
smpp.const.command_bind_transmitter	PDU type 0x00000002 - BIND_TRANSMITTER
smpp.const.command_bind_transmitter_resp	PDU type 0x80000002 - BIND_TRANSMITTER_RESP
smpp.const.command_query_sm	PDU type 0x00000003 - QUERY_SM
smpp.const.command_query_sm_resp	PDU type 0x80000003 - QUERY_SM_RESP
smpp.const.command_submit_sm	PDU type 0x00000004 - SUBMIT_SM
smpp.const.command_submit_sm_resp	PDU type 0x80000004 - SUBMIT_SM_RESP
smpp.const.command_deliver_sm	PDU type 0x00000005 - DELIVER_SM
smpp.const.command_deliver_sm_resp	PDU type 0x80000005 - DELIVER_SM_RESP
smpp.const.command_unbind	PDU type 0x00000006 - UNBIND
smpp.const.command_unbind_resp	PDU type 0x80000006 - UNBIND_RESP
smpp.const.command_replace_sm	PDU type 0x00000007 - REPLACE_SM
smpp.const.command_replace_sm_resp	PDU type 0x80000007 - REPLACE_SM_RESP
smpp.const.command_cancel_sm	PDU type 0x00000008 - CANCEL_SM
smpp.const.command_cancel_sm_resp	PDU type 0x80000008 - CANCEL_SM_RESP
smpp.const.command_bind_transceiver	PDU type 0x00000009 - BIND_TRANSCEIVER
smpp.const.command_bind_transceiver_resp	PDU type 0x80000009 - BIND_TRANSCEIVER_RESP
smpp.const.command_outbind	PDU type 0x0000000B - OUTBIND
smpp.const.command_enquire_link	PDU type 0x00000015 - ENQUIRE_LINK
smpp.const.command_enquire_link_resp	PDU type 0x80000015 - ENQUIRE_LINK_RESP
smpp.const.command_submit_multi	PDU type 0x00000021 - SUBMIT_MULTI
smpp.const.command_submit_multi_resp	PDU type 0x80000021 - SUBMIT_MULTI_RESP
smpp.const.command_alert	PDU type 0x00000102 - ALERT_NOTIFICATION
smpp.const.command_data_sm	PDU type 0x00000103 - DATA_SM
smpp.const.command_data_sm_resp	PDU type 0x80000103 - DATA_SM_RESP
smpp.const.ton_unknown	Type Of Number - 0 - Unknown
smpp.const.ton_international	Type Of Number - 1 - International
smpp.const.ton_national	Type Of Number - 2 - National
smpp.const.ton_network_specific	Type Of Number - 3 - Network Specific
smpp.const.ton_subscriber	Type Of Number - 4 - Subscriber Number
smpp.const.ton_alphanumeric	Type Of Number - 5 - Alphanumeric
smpp.const.ton_abbreviated	Type Of Number - 6 - Abbreviated
smpp.const.npi_unknown	Numbering Plan - 0 - Unknown
smpp.const.npi_isdn_telephone	Numbering Plan - 1 - ISDN (E163/E164)
smpp.const.npi_data_x121	Numbering Plan - 3 - Data (X.121)
smpp.const.npi_telex	Numbering Plan - 4 - Telex (F.69)

smpp.const.npi_land_mobile	Numbering Plan - 6 - Land Mobile (E.212)
smpp.const.npi_national	Numbering Plan - 8 - National
smpp.const.npi_private	Numbering Plan - 9 - Private
smpp.const.npi_ermes	Numbering Plan - 10 - ERMES
smpp.const.npi_internet_ip	Numbering Plan - 14 - Internet (IP)
smpp.const.npi_wap_client_id	Numbering Plan - 18 - WAP Client Id
smpp.const.dc_default	Data coding - 0 - SMSC Default Alphabet
smpp.const.dc_ia5_ascii	Data coding - 1 - IA5 (CCITT T.50)/ASCII
smpp.const.dc_8bit_binary_1	Data coding - 2 - 8-bit binary
smpp.const.dc_iso_8859_1	Data coding - 3 - Latin 1 (ISO-8859-1)
smpp.const.dc_8bit_binary_2	Data coding - 4 - 8-bit binary
smpp.const.dc_jis	Data coding - 5 - JIS (X 0208-1990)
smpp.const.dc_8859_5	Data coding - 6 - Cyrillic (ISO-8859-5)
smpp.const.dc_8859_8	Data coding - 7 - Latin/Hebrew (ISO-8859-8)
smpp.const.dc_ucs2	Data coding - 8 - UCS2 (ISO/IEC-10646)
smpp.const.dc_pictogram	Data coding - 9 - Pictogram Encoding
smpp.const.dc_iso_2011_jp	Data coding - 10 - ISO-2022-JP (Music Codes)
smpp.const.dc_extended_kanji	Data coding - 13 - Extended Kanji JIS
smpp.const.dc_ks_c_5601	Data coding - 14 - KS C 5601
smpp.const.gsm_no_features	GSM - No specific features selected
smpp.const.gsm_udhi	GSM - UDHI Indicator
smpp.const.gsm_reply_path	GSM - Set Reply Path
smpp.const.gsm_udhi_reply_path	GSM - Set UDHI and Reply Path
smpp.const.int_no	No Intermediate notification requested
smpp.const.int_yes	Intermediate notification requested
smpp.const.mm_default_smsc	ESM class MM - Default SMSC
smpp.const.mm_datagram	ESM class MM - Datagram mode
smpp.const.mm_forward	ESM class MM - Forward (Transaction) mode
smpp.const.mm_store_forward	ESM class MM - Store and Forward mode
smpp.const.mst_enroute	Message state - Enroute
smpp.const.mst_delivered	Message state - Delivered
smpp.const.mst_expired	Message state - Expired
smpp.const.mst_deleted	Message state - Deleted
smpp.const.mst_undeliverable	Message state - Undeliverable
smpp.const.mst_accepted	Message state - Accepted
smpp.const.mst_unknown	Message state - Unknown
smpp.const.mst_rejected	Message state - Rejected
smpp.const.mt_default	ESM class MT - Default message Type
smpp.const.mt_smsc_delivery_rcpt	ESM class MT - SMSC Delivery Receipt
smpp.const.mt_delivery_ack	ESM class MT - ESME Delivery Ack
smpp.const.mt_manual_user_ack	ESM class MT - ESME Manual/User Ack
smpp.const.mt_cnvrns_abort	ESM class MT - Conversation Abort
smpp.const.mt_intrm_dlv_rntf	ESM class MT - Intermediate Delivery Notification
smpp.const.soa_no_sme_ack	SME Ack - No SME Ack
smpp.const.soa_sme_ack	SME Ack - SME Ack Requested
smpp.const.soa_sme_manual_user_ack	SME Ack - SME Manual/User Ack
smpp.const.soa_sme_both	SME Ack - Delivery and Manual/User Ack
smpp.const.sdr_no_smsc_delivery	No SMSC Delivery Receipt
smpp.const.sdr_success_failure	SMSC Delivery Receipt success or failure
smpp.const.sdr_failure	SMSC Delivery Receipt failure
smpp.const.si_not_screened	Screening Indicator - not screened
smpp.const.si_verified_passed	Screening Indicator - verified and passed
smpp.const.si_verified_failed	Screening Indicator - verified and failed
smpp.const.si_network_provided	Screening Indicator - network provided
smpp.const.pi_allowed	Presentation Indicator - allowed
smpp.const.pi_restricted	Presentation Indicator - restricted
smpp.const.pi_not_available	Presentation Indicator - n/a
smpp.const.dfr_destination_unavailable	Delivery Failure Reason - destination n/a
smpp.const.dfr_destination_address_invalid	Delivery Failure Reason - address invalid
smpp.const.dfr_perm_net_err	Delivery Failure Reason - permanent error

smpp.const.dfr_temp_net_err	Delivery Failure Reason - temporary error
smpp.const.das_unknown	Dest Addr Subunit Type - Unknown
smpp.const.das_ms_display	Dest Addr Subunit Type - MS Display
smpp.const.das_mobile_equipment	Dest Addr Subunit Type - Mobile Equipment
smpp.const.das_smart_card_1	Dest Addr Subunit Type - Smart Card 1
smpp.const.das_external_unit_1	Dest Addr Subunit Type - External Unit 1
smpp.const.db_unknown	Dest Bearer Type - Unknown
smpp.const.db_sms	Dest Bearer Type - SMS
smpp.const.db_csd	Dest Bearer Type - Circuit Switched Data
smpp.const.db_packet_data	Dest Bearer Type - Packet Data
smpp.const.db_ussd	Dest Bearer Type - USSD
smpp.const.db_cdpd	Dest Bearer Type - CDPD
smpp.const.db_data_tac	Dest Bearer Type - DataTAC
smpp.const.db.flex_reflex	Dest Bearer Type - FLEX/ReFLEX
smpp.const.db_cell_broadcast	Dest Bearer Type - Cell Broadcast
smpp.const.dn_unknown	Dest Network Type - Unknown
smpp.const.dn_gsm	Dest Network Type - GSM
smpp.const.dn_ansi_136	Dest Network Type - ANSI-136/TDM
smpp.const.dn_is_95	Dest Network Type - IS-95/CDMA
smpp.const.dn_pdc	Dest Network Type - PDC
smpp.const.dn_phs	Dest Network Type - PHS
smpp.const.dn_iden	Dest Network Type - iDEN
smpp.const.dn_amps	Dest Network Type - AMPS
smpp.const.dn_paging_network	Dest Network Type - Paging Network
smpp.const.ds_nsap_even	Dest Subaddress Type - NSAP (Even) [ITUT X.213]
smpp.const.ds_nsap_odd	Dest Subaddress Type - NSAP (Odd) [ITUT X.213]
smpp.const.ds_user_specified	Dest Subaddress Type - User specified
smpp.const.dt_temporary	Display Time - Temporary
smpp.const.dt_default	Display Time - Default
smpp.const.dt_invoke	Display Time - Invoke
smpp.const.irc_digit	Its Reply Type - Digit
smpp.const.irc_number	Its Reply Type - Number
smpp.const.irc_telephone_no	Its Reply Type - Telephone No.
smpp.const.irc_password	Its Reply Type - Password
smpp.const.irc_char_line	Its Reply Type - Character Line
smpp.const.irc_menu	Its Reply Type - Menu
smpp.const.irc_date	Its Reply Type - Date
smpp.const.irc_time	Its Reply Type - Time
smpp.const.irc_continue	Its Reply Type - Continue
smpp.const.li_unspecified	Language Indicator - Unspecified
smpp.const.li_english	Language Indicator - English
smpp.const.li_french	Language Indicator - French
smpp.const.li_spanish	Language Indicator - Spanish
smpp.const.li_german	Language Indicator - German
smpp.const.li_portuguese	Language Indicator - Portuguese
smpp.const.mas_available	MS Availability Status - Available
smpp.const.mas_denied	MS Availability Status - Denied
smpp.const.mas_unavailable	MS Availability Status - Unavailable
smpp.const.mmwf_voicemail	MS msg Wait Facilities - Voicemail Message Waiting
smpp.const.mmwf_fax	MS msg Wait Facilities - Fax Message Waiting
smpp.const.mmwf_email	MS msg Wait Facilities - Electronic Mail Message Waiting
smpp.const.mmwf_other	MS msg Wait Facilities - Other Message Waiting
smpp.const.mv_store_indef	MS Validity - Store Indefinitely
smpp.const.mv_power_down	MS Validity - Power Down
smpp.const.mv_sid_based_reg_area	MS Validity - SID based registration area
smpp.const.mv_display_only	MS Validity - Display Only
smpp.const.nec_ansi_136	Network Error Code - ANSI-136
smpp.const.nec_is_95	Network Error Code - IS-95
smpp.const.nec_gsm	Network Error Code - GSM

smpp.const.pc_default	Payload Code - Default
smpp.const.pc_wcmp	Payload Code - WCMP message
smpp.const.pi_plevel_0	Privacy Indicator - Level 0 (Not Restricted)
smpp.const.pi_plevel_1	Privacy Indicator - Level 1 (Restricted)
smpp.const.pi_plevel_2	Privacy Indicator - Level 2 (Confidential)
smpp.const.pi_plevel_3	Privacy Indicator - Level 3 (Secret)
smpp.const.sas_unknown	Source Addr Subunit - Unknown
smpp.const.sas_ms_display	Source Addr Subunit - MS Display
smpp.const.sas_mobile_equipment	Source Addr Subunit - Mobile Equipment
smpp.const.sas_smart_card_1	Source Addr Subunit - Smart Card 1
smpp.const.sas_external_unit_1	Source Addr Subunit - External Unit 1
smpp.const.sbc_unknown	Source Bearer Type - Unknown
smpp.const.sbc_sms	Source Bearer Type - SMS
smpp.const.sbc_csd	Source Bearer Type - Circuit Switched Data
smpp.const.sbc_packet_data	Source Bearer Type - Packet Data
smpp.const.sbc_ussd	Source Bearer Type - USSD
smpp.const.sbc_cdpd	Source Bearer Type - CDPD
smpp.const.sbc_data_tac	Source Bearer Type - DataTAC
smpp.const.sbc_flex_reflex	Source Bearer Type - FLEX/ReFLEX
smpp.const.sbc_cell_broadcast	Source Bearer Type - Cell Broadcast
smpp.const.sn_unknown	Source Network Type - Unknown
smpp.const.sn_gsm	Source Network Type - GSM
smpp.const.sn_ansi_136	Source Network Type - ANSI-136/TDMA
smpp.const.sn_is_95	Source Network Type - IS-95/CDMA
smpp.const.sn_pdc	Source Network Type - PDC
smpp.const.sn_phs	Source Network Type - PHS
smpp.const.sn_iden	Source Network Type - iDEN
smpp.const.sn_amps	Source Network Type - AMPS
smpp.const.sn_paging_network	Source Network Type - Paging Network
smpp.const.ss_nsap_even	Source Subaddress - NSAP (Even) [ITUT X.213]
smpp.const.ss_nsap_odd	Source Subaddress - NSAP (Odd) [ITUT X.213]
smpp.const.ss_user_specified	Source Subaddress - User Specified
smpp.const.uso_pssd_indication	Ussd Service Op - PSSD indication
smpp.const.uso_pssr_indication	Ussd Service Op - PSSR indication
smpp.const.uso_ussr_request	Ussd Service Op - USSR request
smpp.const.uso_ussn_request	Ussd Service Op - USSN request
smpp.const.uso_pssd_response	Ussd Service Op - PSSD response
smpp.const.uso_pssr_response	Ussd Service Op - PSSR response
smpp.const.uso_ussr_confirm	Ussd Service Op - USSR confirm
smpp.const.uso_ussn_confirm	Ussd Service Op - USSN confirm
smpp.const.da_sme_address	Dest Flag - SME Address
smpp.const.da_dl_address	Dest Flag - Distribution List Name
Short message TPDU 3GPP TS 23.040 (SMSTPDU)	
smstpdu.const.noa_unknown	Type Of Number - Unknown
smstpdu.const.noa_international	Type Of Number - International
smstpdu.const.noa_national	Type Of Number - National
smstpdu.const.noa_network_specific	Type Of Number - Network specific
smstpdu.const.noa_subscriber	Type Of Number - Subscriber
smstpdu.const.noa_alphanumeric	Type Of Number - Alphanumeric
smstpdu.const.noa_abbreviated	Type Of Number - Abbreviated
smstpdu.const.np_unknown	Numbering Plan - Unknown
smstpdu.const.np_isdn_telephone	Numbering Plan - ISDN/telephone numbering plan
smstpdu.const.np_data_x121	Numbering Plan - Data numbering plan
smstpdu.const.np_telex	Numbering Plan - Telex numbering plan
smstpdu.const.np_land_mobile	Numbering Plan - Land mobile numbering plan
smstpdu.const.np_national	Numbering Plan - National numbering plan
smstpdu.const.np_private	Numbering Plan - Private numbering plan
smstpdu.const.np_ermes	Numbering Plan - ERMES numbering plan
smstpdu.const.mti_deliver	TP-MTI Type - SMS-DELIVER
smstpdu.const.mti_submit	TP-MTI Type - SMS-SUBMIT
smstpdu.const.vpf_np	TP-Validity-Period-Format Type - field not present

smstpdu.const.vpf_relative	relative format
smstpdu.const.vpf_enhanced	enhanced format
smstpdu.const.vpf_absolute	absolute format
smstpdu.const.dcs_gsm7	Alphabet type - GSM 7 bit default alphabet
smstpdu.const.dcs_8bit	Alphabet type - 8 bit data
smstpdu.const.dcs_ucs2	Alphabet type - UCS2 (16bit)
Transaction Capabilities Application Part (TCAP)	
tcap.const.tcmt_begin	TCMessage type - Begin
tcap.const.tcmt_continue	TCMessage type - Continue
tcap.const.tcmt_end	TCMessage type - End
tcap.const.tcmt_abort	TCMessage type - Abort
tcap.const.ct_invoke	Component type - Invoke
tcap.const.ct_result_last	Component type - ReturnResultLast
tcap.const.ct_error	Component type - ReturnError
tcap.const.ct_reject	Component type - Reject
tcap.const.ct_result_not_last	Component type - ReturnResultNotLast
Mobile Application Part (MAP)	
map.const.noa_unknown	Nature Of Address - Unknown
map.const.noa_international	Nature Of Address - International
map.const.noa_national	Nature Of Address - National
map.const.noa_network_specific	Nature Of Address - Network specific
map.const.noa_subscriber	Nature Of Address - Subscriber
map.const.noa_abbreviated	Nature Of Address - Abbreviated
map.const.np_unknown	Numbering plan - Unknown
map.const.np_isdn_telephone	Numbering plan - ISDN
map.const.np_data_x121	Numbering plan - Data (X.121)
map.const.np_telex	Numbering plan - Telex
map.const.np_land_mobile	Numbering plan - Land mobile
map.const.np_national	Numbering plan - National
map.const.np_private	Numbering plan - Private
MTP Level 3 (MTP3) User Adaptation Layer (M3UA)	
m3ua.const.si_snmp	Service Indicator - SNMP
m3ua.const.si_sntmm	Service Indicator - SNTMM
m3ua.const.si_sntmsm	Service Indicator - SNTMSM
m3ua.const.si_sccp	Service Indicator - SCCP
m3ua.const.si_tup	Service Indicator - TUP
m3ua.const.si_isup	Service Indicator - ISUP
m3ua.const.si_dup_call_circuit	Service Indicator - DUP_CALL_AND_CIRCUIT
m3ua.const.si_dup_reg_canc	Service Indicator - DUP_REG_AND_CANC
m3ua.const.si_mtp_test	Service Indicator - MTP_TESTING
m3ua.const.si_bisup	Service Indicator - BISUP
m3ua.const.si_sisup	Service Indicator - SISUP
m3ua.const.si_gcp	Service Indicator - GCP
Signalling Connection Control Part (SCCP)	
sccp.const.route_on_ssn	Routing Indicator - Route on SSN
sccp.const.route_on_gt	Routing Indicator - Route on GT
sccp.const.gti_noa	Global Title Indicator - nature of address indicator only
sccp.const.gti_tt	Global Title Indicator - translation type only
sccp.const.gti_ttnpe	Global Title Indicator - translation type, numbering plan and encoding scheme
sccp.const.gti_ttnpenoa	Global Title Indicator - translation type, numbering plan, encoding scheme and nature of address indicator
sccp.const.np_unknown	Numbering Plan - unknown
sccp.const.np_isdn_telephone	Numbering Plan - ISDN/telephony
sccp.const.np_generic	Numbering Plan - generic
sccp.const.np_data_x121	Numbering Plan - data X.121
sccp.const.np_telex	Numbering Plan - telex
sccp.const.np_maritime	Numbering Plan - maritime mobile
sccp.const.np_land_mobile	Numbering Plan - land mobile
sccp.const.np_isdn_mobile	Numbering Plan - ISDN/mobile



sccp.const.np_private	Numbering Plan - private
sccp.const.noa_unknown	Nature Of Address - unknown
sccp.const.noa_subscriber_number	Nature Of Address - subscriber number
sccp.const.noa_national	Nature Of Address - national number
sccp.const.noa_national_significant	Nature Of Address - national significant number
sccp.const.noa_international	Nature Of Address - international
sccp.const.ssn_unknown	Subsystem Number - SSN not known/not used
sccp.const.ssn_sccp_mngmt	Subsystem Number - SCCP management
sccp.const.ssn_itut	Subsystem Number - reserved for ITU-T allocation
sccp.const.ssn_isup	Subsystem Number - ISDN user part
sccp.const.ssn_omap	Subsystem Number - OMAP (Operation, Maintenance and Administration Part)
sccp.const.ssn_map	Subsystem Number - MAP (Mobile Application Part)
sccp.const.ssn_hlr	Subsystem Number - HLR (Home Location Register)
sccp.const.ssn_vlr	Subsystem Number - VLR (Visitor Location Register)
sccp.const.ssn_msc	Subsystem Number - MSC (Mobile Switching Centre)
sccp.const.ssn_eic	Subsystem Number - EIC (Equipment Identifier Centre)
sccp.const.ssn_auc	Subsystem Number - AUC (Authentication Centre)
sccp.const.ssn_isdn_sup	Subsystem Number - ISDN supplementary services
sccp.const.ssn_international	Subsystem Number - reserved for international use
sccp.const.ssn_broadband_isdn	Subsystem Number - broadband ISDN edge-to-edge applications
sccp.const.ssn_tc_test	Subsystem Number - TC test responder

#### 8.2.4 Advanced matching

Filtering Gateway Node (FGN) offers a wide range of filtering techniques; basic syntax<sup>12</sup>, although quite powerful and adequate for most filtering requirements, is inherently limited. In some circumstances, threat detection may incur additional complexity and surpass the limited rule definition capabilities of basic syntax.

Project MINK Framework (pMINK) features an Application Programming Interface (API) for FGN Rule Processing Engine (RPE); as already mentioned in section 8.2.3.2.6, Lua programming language is used for basic syntax extension development.

The procedure for activating *advanced* matching mode is quite simple; it only requires a LUA type operand to be used as part of *basic* matching. After successful activation, indicated by two consecutive grave accent(`) signs, Lua scripting will be enabled.

There are two ways of importing Lua source code into FGN Rule Processing Engine; the first one, inline mode, is to input the actual code right after LUA operand activation sequence. Although more convenient, inline mode should not be used for large and/or multi-line Lua scripts, only for single line function calls. External mode, indicated by an absolute path appearing right after the activation sequence, is a universal solution for both multi-line and single line Lua scripts, primarily targeting the former.

Project MINK Framework (pMINK) contains a specialized Lua module<sup>13</sup>, included implicitly in inline mode, required for successful communication between FGN Rule Processing Engine and user defined scripts.

##### 8.2.4.1 Inline scripting

Example rule ("match sccp" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/sccp > configuration
cgpa {
    gt {
        address "`return F.sl_get('gt_black_list', F.vpval())"
    }
}
```

Inline Lua script ("match sccp cgpa gt address" field):

```
return F.sl_get('gt_black_list', F.vpval())
```

<sup>12</sup> Described in section 8.2.3.1

<sup>13</sup> Described in section 8.2.4.3



### 8.2.4.2 External scripting

Example rule ("match sccp" section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/match/sccp > configuration
cgpa {
    gt {
        address "`/tmp/test.lua"
    }
}
```

External("/tmp/test.lua") Lua script ("match sccp cgpa gt address" field):

```
-- pmink module
local F = require('pmink')(...)
-- return
return F.sl_get('gt_black_list', F.vpval())
```

### 8.2.4.3 Lua pMINK module

When using inline Lua scripting mode, pMINK module is included implicitly; the following line is automatically inserted at the beginning of the script.

```
local F = require('pmink')(...)
```

External scripting mode does not enforce this behaviour; pMINK module, if needed, should be included explicitly by user.

#### 8.2.4.3.1 Variant parameter management

In advanced matching mode, both external and internal signalling data exists as a variant parameter map. Unlike basic syntax, where left operand("config-item") is already set and cannot be changed, advanced matching mode imposes no such restriction. Variant parameters, when used in context of data matching, should be considered as an analogy for basic syntax operands(left and right).

vpget([var, index])		
get VariantParam referenced by <u>var</u> / <u>index</u> parameters		
Parameters		
var	number	Predefined pMINK variable id; if omitted, value assigned to the current field will be used instead
index	number	Predefined pMINK variable instance index; if omitted, value assigned to the current field will be used instead
Returns		
VariantParam pointer (cdata) or nil if not found		

Example:

```
-- =====
-- vpget example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- get VariantParam (vpget)
vp = F.vpget(F.CONST._pt_sccp_called_pa_gt_address)

-- printing out vp would produce the following output
-- cdata<void *>: 0x7fffe8158f60

-- return
return true
```

vpset(vp, val, [len])		
set VariantParam value		
Parameters		
vp	cdata	VariantParam pointer (returned by <code>vpval</code> or <code>vpnew</code> )
val	number	New value for VariantParam ( <code>vp</code> )
len	number	New value length in bytes (needed only for DPT_OCTETS VariantParam type)
Returns		
true if value for VariantParam ( <code>vp</code> ) was set, false or nil on error		

Example:

```
-- =====
-- vpset example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- create VariantParam (vpnew)
vp = F.vpnew(F.CONST._pt_tcap_error_code)

-- set VariantParam value (vpset)
F.vpset(vp, 30)

-- return
return true
```

vpval([vp])		
get VariantParam ( <code>vp</code> ) value		
Parameters		
vp	cdata	VariantParam pointer (returned by <code>vpval</code> or <code>vpnew</code> ); if omitted, <code>vp</code> will be set to VariantParam pointer assigned to the current field
Returns		
VariantParam value (number, boolean or cdata) or nil on error		

Example:

```
-- =====
-- vpval example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- get VariantParam (vpget)
vp = F.vpget(F.CONST._pt_sccp_called_pa_gt_address)

-- get vp value (cdata)
vp_val = F.vpval(vp)
-- printing vp_val would produce the following output
-- cdata<char *>: 0x7ffe81178d0

-- for native lua string, conversion is done using F.string
vp_str = F.string(vp_val)

-- return
return true
```

vpsize(vp)		
get VariantParam value size in bytes		
Parameters		
vp	cdata	VariantParam pointer (returned by <u>vpval</u> or <u>vpnew</u> )
Returns		
size of VariantParam value in bytes or 0 on error		

Example:

```
-- =====
-- vpsize example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- get VariantParam (vpget)
vp = F.vpget(F.CONST._pt_sccp_called_pa_gt_address)

-- get vp value size
vp_val_size = F.vpsize(vp)
-- cdata<char *> VariantParam includes C string NULL character in value size calculation

-- return
return true
```

vpnew(var, [index])		
create new VariantParam object		
Parameters		
var	number	Predefined pMINK variable id
index	number	Predefined pMINK variable instance index
Returns		
VariantParam pointer (cdata) or nil on error		

Example:

```
-- =====
-- vpnew example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- create VariantParam (vpnew)
vp = F.vpnew(F.CONST._pt_tcap_error_code)

-- set VariantParam value (vpset)
F.vpset(vp, 30)

-- return
return true
```

getval([var, index])		
get VariantParam value for VariantParam referenced by <u>var</u> / <u>index</u> parameters; faster alternative for <u>vpval(vpget(var, [index]))</u>		
Parameters		
var	number	Predefined pMINK variable id; if omitted, value from the current field will be used instead
index	number	Predefined pMINK variable instance index; if omitted, value from the current field will be used instead
Returns		
VariantParam value (number, boolean or cdata) or nil on error		

Example:

```
-- =====
-- getval example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- get VariantParam value (getval)
vp_val = F.getval(F.CONST._pt_sccp_called_pa_gt_address)

-- printing vp_val would produce the following output
-- cdata<char *>: 0x7ffe81178d0

-- for native lua string, conversion is done using F.string
vp_str = F.string(vp_val)

-- return
return true
```

#### 8.2.4.3.2 Standard dynamic filtering lists

Advanced matching mode differentiates between two types of lists; *standard* and *flood*<sup>14</sup>. The term "*standard*" refers to an extension of basic syntax terminology; both *static*<sup>15</sup> and *dynamic* lists are grouped together and referred to as "*standard lists*".

The only difference between *static* and *dynamic* lists is that the former can be predefined via pMINK Command Line Interface (CLI).

Basic syntax equivalents:

Advanced mode function	Basic syntax operator/operand
<b>sl_add</b>	>>@{list_name}
<b>sl_get</b>	==@{list_name}
<b>sl_del</b>	<<@{list_name} for item or —@{list_name} for the whole list
<b>sl_new</b>	>>@{list_name} (adding item to unknown list will create it automatically)
<b>sl_size</b>	n/a
<b>sl_getbi</b>	n/a

<sup>14</sup> Described in section 8.2.4.3.3

<sup>15</sup> Described in section 8.1.3.3

sl_add(list, item)		
add new item to standard <sup>16</sup> list		
Parameters		
list	string	Standard list name
item	string	String item to be inserted
Returns		
true if item was successfully inserted or false on error		

Example:

```
-- =====
-- sl_add example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- add '123456' to a standard list named 'gt_black_list' (sl_add)
res = F.sl_add('gt_black_list', '123456')

-- return
return res
```

sl_get(list, item)		
Check if item exists in standard list		
Parameters		
list	string	Standard list name
item	string	String item to search for
Returns		
true if item was found or false if list or item do not exist		

Example:

```
-- =====
-- sl_get example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- check if '123456' exists in a standard list named 'gt_black_list' (sl_get)
res = F.sl_get('gt_black_list', '123456')

-- return
return res
```

<sup>16</sup> The term "standard list" is used for both dynamic and static (section 8.1.3.3) lists

sl_del(list, [item])		
Delete item from list or the whole list		
Parameters		
list	string	Standard list name
item	string	String item to delete
Returns		
true, false if list parameter is nil		

Example:

```
-- =====
-- sl_del example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- remove '123456' from list 'gt_black_list'
F.sl_del('gt_black_list', '123456')

-- remove the whole 'gt_black_list' list
F.sl_del('gt_black_list')

-- return
return true
```

sl_new(label)		
Create new standard list		
Parameters		
label	string	Standard list name
Returns		
true, false if list parameter is nil		

Example:

```
-- =====
-- sl_new example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- create new standard list named 'new_test_list'
F.sl_new('new_test_list')

-- return
return true
```

**sl\_size(list)**

Get number of items in a list

**Parameters**

list	string	Standard list name
------	--------	--------------------

**Returns**

number of items in a list, 0 on error

Example:

```
-- =====
-- sl_size example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- get number of items in a list named 'gt_black_list' (sl_size)
sl_size = F.sl_size('gt_black_list')

-- return
return true
```

**sl\_getbi(list, index)**

Get list item at a specific index

**Parameters**

list	string	Standard list name
index	number	Item index

**Returns**

Item (cdata&lt;char \*&gt;)

Example:

```
-- =====
-- sl_getbi example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- get item at index 1 from list named 'gt_black_list' (sl_getbi)
sl_item = F.sl_getbi('gt_black_list', 1)

-- if item cannot be found, sl_getbi will return cdata
-- containing NULL pointer
if(sl_item == nil) then return false end

-- return
return true
```

### 8.2.4.3.3 Flood management

Flooding, in context of Short Message Service (SMS), is a term used to describe a sudden increase in particular type of network traffic; a repetitive pattern is detected and handled in accordance with user requirements.

Filtering Gateway Node (FGN) uses a manual approach for dealing with various types of flooding threats; all flooding related actions and events should be handled by filtering rules in runtime configuration. This approach may seem complex at first due to lack of predefined logic and automation, but it clearly offers a much wider range of threat detection possibilities.

Advanced mode offers a set of functions for dealing with flooding; the majority of situations can be handled successfully by using `"fl_add"` and `"fl_get"` respectively, as part of rule definition.

Detecting a pattern is a semi-automatic process; assessing which protocol parameters should be monitored for increased level of repetition is a responsibility which lies with the user. As stated earlier, every incoming SMS should be processed, and parameters of interest added to their assigned flood lists using `"fl_add"`. For better accuracy and more realistic results, different parameter types should be assigned to their own separate flood lists (e.g. `"gt_flood_list"` for GTs, `"imsi_flood_list"` for IMSIs).

Flood lists have a tendency to expand quite a bit; the growth factor will greatly depend on the amount of traffic flowing through Filtering Gateway Node (FGN), and total number of simultaneous flood lists in use. To minimize memory consumption, an approximate value of network's **MSU/sec** should be taken into account when setting up flood list limits<sup>17</sup> in FGN runtime configuration.

fl_add(list, item)		
add new item to flood list		
Parameters		
list	string	Flood list name
item	string	String item to be inserted
Returns		
true if item was successfully inserted or false on error		

Example:

```
-- =====
-- fl_add example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- add '123456' to a flood list named 'gt_flood_list' (fl_add)
res = F.fl_add('gt_flood_list', '123456')

-- return
return res
```

<sup>17</sup> Described in section 8.1.3.1



fl_get(list, interval, item)		
Get number of item repetitions during an <code>[now - interval, now]</code> interval		
Parameters		
list	string	Flood list name
interval	number	How many seconds to rewind (start of interval)
item	string	String item to search for
Returns		
number of item repetitions		

Example:

```
-- =====
-- fl_get example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- check how many times '123456' has repeated itself in the last hour (60 seconds)
-- in 'gt_flood_list' flood list (fl_get)
count = F.fl_get('gt_flood_list', 60, '123456')

-- return true if '123456' has more than 10 repetitions
-- in the last hour
return count > 10
```

fl_del(list, [item])		
Delete item from list or the whole list		
Parameters		
list	string	Flood list name
item	string	String item to delete
Returns		
true, false if list parameter is nil		

Example:

```
-- =====
-- fl_del example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- remove '123456' from list 'gt_flood_list'
F.fl_del('gt_flood_list', '123456')

-- remove the whole 'gt_flood_list' list
F.fl_del('gt_flood_list')

-- return
return true
```

fl_new(label)		
Create new standard list		
Parameters		
label	string	Flood list name
Returns		
true, false if list parameter is nil		

Example:

```
-- =====
-- fl_new example
-- =====
-- pMINK module
local F = require('pmink')(...)

-- create new flood list named 'new_flood_list'
F.fl_new('new_flood_list')

-- return
return true
```

#### 8.2.4.3.4 Predefined constants

Advanced matching mode(lua) features a set of predefined constants to provide a more human friendly way of referencing certain protocol fields and/or parameters.

pMINK lua constant	Value
Project MINK Framework (pMINK)	
_pt_pmink_daemon_type	Originating daemon type
_pt_pmink_daemon_id	Originating daemon id
_pt_pmink_timestamp	UNIX timestamp (sec part)
_pt_pmink_timestamp_nsec	UNIX timestamp (nsec part)
_pt_pmink_loop_count	Packet loop count (SGN feature)
_pt_pmink_routing_destination	R14P next routing destination (STP feature)
_pt_pmink_routing_index	Index of last used R14P routing destination
_pt_pmink_trunk_label	Trunk label string
_pt_pmink_connection_type	External connection type
_pt_pmink_service_id	R14P service id
_pt_pmink_command_id	pMINK command id
_pt_pmink_routing_sub_destination	Next routing sub-destination
_pt_pmink_routing_sub_destination_type	Next routing sub-destination type
_pt_pmink_correlation_notification	Request notification from SGN
_pt_pmink_guid	pMINK correlation GUID
_pt_pmink_filter_result	FGN filtering result
_pt_pmink_filter_exit	FGN filtering exit label (last executed rule)
Short Message Peer-to-Peer (SMPP)	
_pt_smpp_originator_ton	Source address type of number
_pt_smpp_originator_np	Source address numbering plan indicator
_pt_smpp_originator_address	SME originating address
_pt_smpp_recipient_ton	Destination address type of number
_pt_smpp_recipient_np	Destination address numbering plan indicator
_pt_smpp_recipient_address	SME destination address

<code>_pt.smpp.esm.message.mode</code>	ESM class message mode
<code>_pt.smpp.esm.message.type</code>	ESM class message type
<code>_pt.smpp.esm.gsm.features</code>	ESM class GSM network specific features
<code>_pt.smpp.protocol.id</code>	Protocol identifier according to GSM 03.40
<code>_pt.smpp.priority.flag</code>	Short message priority level
<code>_pt.smpp.delivery.time</code>	Scheduled time at which the delivery should be first attempted
<code>_pt.smpp.validity.period</code>	Validity period
<code>_pt.smpp.rd.smsc.receipt</code>	SMSC Delivery Receipt
<code>_pt.smpp.rd.sme.ack</code>	SME originated Acknowledgement
<code>_pt.smpp.rd.intermediate.notification</code>	Intermediate Notification
<code>_pt.smpp.replace.if.present.flag</code>	Request SMSC to replace a previously submitted message
<code>_pt.smpp.data.coding</code>	Short message data coding
<code>_pt.smpp.sm.default.msg.id</code>	SMSC index of a pre-defined (canned) message
<code>_pt.smpp.sm.length</code>	Length of short message parameter in octets
<code>_pt.smpp.sm</code>	Short message octets
<code>_pt.smpp.command.id</code>	SMPP PDU message type
<code>_pt.smpp.sar.msg.ref.num</code>	Reference number for a particular concatenated short message
<code>_pt.smpp.sar.total.segments</code>	Number of messages within the concatenated message
<code>_pt.smpp.sar.segment.seqnum</code>	Message sequence number within the concatenated message
<code>_pt.smpp.header.data</code>	SMPP raw header data
<code>_pt.smpp.peer.ip</code>	Peer connection IP address
<code>_pt.smpp.peer.port</code>	Peer connection port
<code>_pt.smpp.sequence</code>	Sequence number
<code>_pt.smpp.as.label</code>	Application Server label
<code>_pt.smpp.asp.label</code>	Application Server Process label
<code>_pt.smpp.dlvr.rcpt.id</code>	Delivery receipt id
<code>_pt.smpp.dlvr.rcpt.sub</code>	Delivery receipt sub
<code>_pt.smpp.dlvr.rcpt.dlvr</code>	Delivery receipt dlvr
<code>_pt.smpp.dlvr.rcpt.submit.date</code>	Delivery receipt submit date
<code>_pt.smpp.dlvr.rcpt.done.date</code>	Delivery receipt done date
<code>_pt.smpp.dlvr.rcpt.stat</code>	Delivery receipt status
<code>_pt.smpp.dlvr.rcpt.err</code>	Delivery receipt error
<code>_pt.smpp.dlvr.rcpt.text</code>	Delivery receipt text
<code>_pt.smpp.message.id</code>	ACK message id (Submit_sm.resp)
<code>_pt.smpp.command.status</code>	Command status
Transaction Capabilities Application Part (TCAP)	
<code>_pt.tcap.source.transaction.id</code>	Source transaction id
<code>_pt.tcap.destination.transaction.id</code>	Destination transaction id
<code>_pt.tcap.opcode</code>	Component operation code
<code>_pt.tcap.component.type</code>	Component type
<code>_pt.tcap.component.invoke.id</code>	Component invoke id
<code>_pt.tcap.error.type</code>	Error type
<code>_pt.tcap.error.code</code>	Error code
<code>_pt.tcap.dialogue.context.oid</code>	Dialogue application context
<code>_pt.tcap.message.type</code>	TCAP message type

Mobile Application Part (MAP)	
<code>_pt_gsmmap_scoa.digits</code>	Service centre address OA digits
<code>_pt_gsmmap_scoa.type_of_number</code>	Service centre address OA Nature of address indicator
<code>_pt_gsmmap_scoa.numbering_plan</code>	Service centre address OA Numbering plan
<code>_pt_gsmmap_scda.digits</code>	Service centre address DA digits
<code>_pt_gsmmap_scda.type_of_number</code>	Service centre address DA Nature of address indicator
<code>_pt_gsmmap_scda.numbering_plan</code>	Service centre address DA Numbering plan
<code>_pt_gsmmap_imsi</code>	International mobile Subscriber Identity
<code>_pt_gsmmap_msisdn.digits</code>	MSISDN address digits
<code>_pt_gsmmap_msisdn.type_of_number</code>	MSISDN Nature of address indicator
<code>_pt_gsmmap_msisdn.numbering_plan</code>	MSISDN Numbering plan
<code>_pt_gsmmap_nnn.digits</code>	Network node number digits
<code>_pt_gsmmap_nnn.type_of_number</code>	Network node number Nature of address indicator
<code>_pt_gsmmap_nnn.numbering_plan</code>	Network node number Numbering plan
<code>_pt_gsmmap_an.digits</code>	Additional number digits
<code>_pt_gsmmap_an.type_of_number</code>	Additional number Nature of address indicator
<code>_pt_gsmmap_an.numbering_plan</code>	Additional number Numbering plan
<code>_pt_gsmmap_sca.digits</code>	Service centre address digits
<code>_pt_gsmmap_sca.type_of_number</code>	Service centre address Nature of address indicator
<code>_pt_gsmmap_sca.numbering_plan</code>	Service centre address Numbering plan
<code>_pt_gsmmap_version</code>	MAP version (1, 2 or 3)
Short message TPDU 3GPP TS 23.040 (SMSTPDU)	
<code>_pt_smstpdu_tp_udhi</code>	TP-UD header indicator
<code>_pt_smstpdu_tp_sri</code>	SME status report indicator
<code>_pt_smstpdu_tp_mms</code>	More message to send indicator
<code>_pt_smstpdu_tp_mti</code>	Message type indicator
<code>_pt_smstpdu_tp_oa.type_of_number</code>	Originating address Type Of Number
<code>_pt_smstpdu_tp_oa.numbering_plan</code>	Originating address Numbering Plan
<code>_pt_smstpdu_tp_oa.digits</code>	Originating address digits
<code>_pt_smstpdu_tp_pid</code>	Protocol identifier
<code>_pt_smstpdu_tp_dcs</code>	Data coding scheme
<code>_pt_smstpdu_tp_scts</code>	Service centre time stamp
<code>_pt_smstpdu_tp_udl</code>	Length of user data TP-UD
<code>_pt_smstpdu_tp_ud</code>	TP-UD decoded data (if applicable)
<code>_pt_smstpdu_tp_rp</code>	Reply path indicator
<code>_pt_smstpdu_tp_srr</code>	MS status report request
<code>_pt_smstpdu_tp_vpf</code>	TP-VP field format
<code>_pt_smstpdu_tp_rd</code>	Reject duplicates
<code>_pt_smstpdu_tp_da.type_of_number</code>	Destination address Type Of Number
<code>_pt_smstpdu_tp_da.numbering_plan</code>	Destination address Numbering Plan
<code>_pt_smstpdu_tp_da.digits</code>	Destination address digits
<code>_pt_smstpdu_tp_vp</code>	Validity period
<code>_pt_smstpdu_msg_id</code>	Concatenated short message reference number
<code>_pt_smstpdu_msg_parts</code>	Concatenated short message total parts
<code>_pt_smstpdu_msg_part</code>	Concatenated short message part number
<code>_pt_smstpdu_tp_mr</code>	Message reference
<code>_pt_smstpdu_message_class</code>	Message class

MTP Level 3 (MTP3) User Adaptation Layer (M3UA)	
_pt_m3ua_protocol_data_service.indicator	Service indicator
_pt_m3ua_protocol_data_network.indicator	Network indicator
_pt_m3ua_protocol_data_message.priority	Message priority
_pt_m3ua_protocol_data_destination.point.code	Destination Point Code
_pt_m3ua_protocol_data_originating.point.code	Originating Point Code
_pt_m3ua_protocol_data_signalling.link.selection.code	Signalling link selection code
_pt_m3ua_header_data	M3UA raw header data
_pt_m3ua_as_label	Application Server label
_pt_m3ua_asp_label	Application Server Process label
Signalling Connection Control Part (SCCP)	
_pt_sccp_called_pa_routing.indicator	Called Party Address Routing indicator
_pt_sccp_called_pa_global.title.indicator	Called Party Address Global Title Indicator
_pt_sccp_called_pa_ssn.indicator	Called Party Address SubSystem Number Indicator
_pt_sccp_called_pa_point.code.indicator	Called Party Address Point code Indicator
_pt_sccp_called_pa_point.code.number	Called Party Address Point code
_pt_sccp_called_pa_subsystem.number	Called Party Address SubSystem Number
_pt_sccp_called_pa_gt.numbering.plan	Called Party Address Global Title Numbering plan
_pt_sccp_called_pa_gt.nature.of.address	Called Party Address Global Title Nature Of Address
_pt_sccp_called_pa_gt.address	Called Party Address Global Title digits
_pt_sccp_called_pa_gt.translation.type	Called Party Address Global Title Translation type
_pt_sccp_calling_pa_routing.indicator	Calling Party Address Routing indicator
_pt_sccp_calling_pa_global.title.indicator	Calling Party Address Global Title Indicator
_pt_sccp_calling_pa_ssn.indicator	Calling Party Address SubSystem Number Indicator
_pt_sccp_calling_pa_point.code.indicator	Calling Party Address Point code Indicator
_pt_sccp_calling_pa_point.code.number	Calling Party Address Point code
_pt_sccp_calling_pa_subsystem.number	Calling Party Address SubSystem Number
_pt_sccp_calling_pa_gt.numbering.plan	Calling Party Address Global Title Numbering plan
_pt_sccp_calling_pa_gt.nature.of.address	Calling Party Address Global Title Nature Of Address
_pt_sccp_calling_pa_gt.address	Calling Party Address Global Title digits
_pt_sccp_calling_pa_gt.translation.type	Calling Party Address Global Title Translation type
_pt_sccp_message.type	SCCP message type

### 8.2.5 Rule translate part

Translate section of rule definition, although quite similar to *"match"* section, uses slightly different variation of basic syntax. While *"match"* section was constructed to facilitate data matching, the purpose of *"translate"* section is entirely different.

As already explained in chapter 8.2, this section of rule definition focuses primarily on rule execution; default exit code and next rule in execution queue can both be overridden. A secondary role of *"translate"* section, shared with Signalling Transfer Point (STP), is to provide MTP Level 3 (MTP3) User Adaptation Layer (M3UA) and Signalling Connection Control Part (SCCP) field translations.

The syntax used for field translations is compatible with basic syntax with few minor differences; Left Component Index (LCI), operator (OP) and Right Component Index (RCI) are not supported. Right side operand(*"operand.B"*) is limited to the following operand types: NUMBER<sup>18</sup>, STRING<sup>19</sup>, VARIABLE<sup>20</sup> and REGEX\_SR<sup>21</sup>.

Regular expression type operand uses an extended syntax in *"translate"* section; unlike *"match"* section where REGEX is used only for **searching(S)**, *"translate"* section extends this behaviour to **search-and-replace(SR)**.

Configuration items listed:

```
filter_result - Filtering result
filter_jump - Jump to rule label
r14p - R14P pmink framework data
m3ua - MTP Level 3 (MTP3) User Adaptation Layer
sccp - Signalling Connection Control Part
```

Configuration items explained:

Item	Description
<b>filter_result</b>	Rule <b>exit code</b> <sup>22</sup>
<b>filter_jump</b>	Rule label to jump <sup>22</sup> to; set next rule in Rule Processor's execution queue
Group	Description
<b>r14p</b>	Release14 Application Layer Protocol (R14P)
<b>m3ua</b>	MTP Level 3 (MTP3) User Adaptation Layer (M3UA)
<b>sccp</b>	Signalling Connection Control Part (SCCP)

**8.2.5.1 REGEX\_SR type operand** is a Perl compatible Regular expression indicated by a sequence of characters starting with a colon(:) sign; more detailed information can be found on the following address: <http://perldoc.perl.org/perlre.html#Regular-Expressions>

Unlike REGEX type operand, REGEX\_SR enforces the following syntax: **"/search\_pattern/replace\_pattern"**

Example rule (*"translate m3ua"* section):

```
config@hostname:/configure/mno/fgn/fgn1/rules/rule-00000/definition/translate/m3ua > configuration
opc ":(...)(.*)/99$2"
```

Description:

Replace first two digits of Originating Point Code with "99"

<sup>18</sup> Described in section 8.2.3.2.1

<sup>19</sup> Described in section 8.2.3.2.2

<sup>20</sup> Described in section 8.2.3.2.4

<sup>21</sup> REGEX\_SR operand used in *"translate"* section uses different syntax and is not compatible with standard REGEX described in section 8.2.3.2.3

<sup>22</sup> Described in section 8.2

### 8.2.5.2 R14P translations

R14P is a Release14 protocol, used for internal communication between various Project MINK Framework (pMINK) daemons. It is transferred via Stream Control Transmission Protocol (SCTP) as an X.690 encoded ASN.1 data.

Configuration items listed:

```
trunk_label - Trunk label
service_id - Service id
```

Configuration items explained:

Item	Description
<b>trunk_label</b>	Alphanumeric value assigned by Signalling Transfer Point (STP), used for mobile network traffic identification
<b>service_id</b>	Type of R14P ServiceMessage, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 42 - sid-openli</li> <li>• 43 - sid-sms-data-retention</li> <li>• 44 - sid-stp-routing</li> <li>• 45 - sid-sgn-forward</li> <li>• 46 - sid-smshub-forward</li> <li>• 47 - sid-fgn-filtering</li> <li>• 48 - sid-security</li> <li>• 49 - sid-pdn-filtering</li> </ul>

### 8.2.5.3 MTP Level 3 (MTP3) User Adaptation Layer (M3UA) translations

M3UA stands for MTP Level 3 (MTP3) User Adaptation Layer as defined by the IETF SIGTRAN working group in RFC 4666 (which replaces and supersedes RFC 3332). M3UA enables the SS7 protocol's User Parts (e.g. ISUP, SCCP and TUP) to run over IP instead of telephony equipment like ISDN and PSTN. It is recommended to use the services of SCTP to transmit M3UA.

Configuration items listed:

```
opc - Originating Point Code
dpc - Destination Point Code
si - Service indicator
ni - Network indicator
mp - Message priority
sls - Signalling link selection code
```

Configuration items explained:

Item	Description
<b>opc, dpc</b>	The Originating and Destination Point Code fields contain the OPC and DPC from the routing label of the original SS7 message in Network Byte Order, justified to the least significant bit. Unused bits are coded '0'
<b>si</b>	The Service Indicator field contains the SI field from the original SS7 message justified to the least significant bit. Unused bits are coded '0'
<b>ni</b>	The Network Indicator contains the NI field from the original SS7 message justified to the least significant bit. Unused bits are coded '0'
<b>mp</b>	The Message Priority field contains the MP bits (if any) from the original SS7 message, both for ANSI-style and TTC-style [29] message priority bits. The MP bits are aligned to the least significant bit. Unused bits are coded '0'
<b>sls</b>	The Signalling Link Selection field contains the SLS bits from the routing label of the original SS7 message justified to the least significant bit and in Network Byte Order. Unused bits are coded '0'

#### 8.2.5.4 Signalling Connection Control Part (SCCP) translations

The Signalling Connection Control Part (SCCP) is a network layer protocol that provides extended routing, flow control, segmentation, connection-orientation, and error correction facilities in Signaling System 7 telecommunications networks. SCCP relies on the services of MTP for basic routing and error detection.

The base SCCP specification is defined by the ITU-T, in recommendations Q.711 to Q.714, with additional information to implementors provided by Q.715 and Q.716. There are, however, regional variations defined by local standards bodies. In the United States, ANSI publishes its modifications to Q.713 as ANSI T1.112. The TTC publishes as JT-Q.711 to JT-Q.714, and Europe ETSI publishes ETSI EN 300-009-1: both of which document their modifications to the ITU-T specifications.

Although MTP provides routing capabilities based upon the Point Code, SCCP allows routing using a Point Code and Subsystem number or a Global Title. A Point Code is used to address a particular node on the network, whereas a Subsystem number addresses a specific application available on that node. SCCP employs a process called Global Title Translation to determine Point Codes from Global Titles so as to instruct MTP on where to route messages.

In the SIGTRAN suite of protocols, there are two primary methods of transporting SCCP applications across Internet Protocol networks: SCCP can be transported indirectly using the MTP level 3 User Adaptation protocol (M3UA), a protocol which provides support for users of MTP-3 including SCCP. Alternatively, SCCP applications can operate directly over the SCCP User Adaptation protocol (SUA) which is a form of modified SCCP designed specifically for use in IP networking. Signalling Gateway Node (SGN) uses the first method of SCCP transport; MTP Level 3 (MTP3) User Adaptation Layer (M3UA).

Configuration items listed:

```
cgpa - Calling Party
cdpa - Called Party
```

Configuration sub-items listed ("cgpa/cdpa" section):

```
routing-indicator - Routing indicator
    gti - Global Title Indicator
    ssn - SubSystem Number
    point-code - Point code
    gt - Global Title
```

Configuration items explained:

Item	Description
<b>cgpa, cdpa</b>	Grouping nodes, contain fields for " <i>Calling Party Address</i> " and " <i>Called Party Address</i> "
<b>routing-indicator</b>	Routing type, identifies which address element shall be used for routing: <ul style="list-style-type: none"> <li>• 0 - route on GT</li> <li>• 1 - route on SSN</li> </ul>
<b>gti</b>	Global Title Indicator field contains the type of Global Title included. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available: <ul style="list-style-type: none"> <li>• 0 - no global title included</li> <li>• 4 - global title includes nature of address indicator only</li> <li>• 8 - global title includes translation type only</li> <li>• 12 - global title includes translation type, numbering plan and encoding scheme</li> <li>• 16 - global title includes translation type, numbering plan, encoding scheme and nature of address indicator</li> </ul>



Item	Description
<b>ssn</b>	<p>The Subsystem Number (SSN) is a numerical fields identifying an SCCP user function. A list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - SSN not known/not used</li> <li>• 1 - SCCP management</li> <li>• 2 - reserved for ITU-T allocation</li> <li>• 3 - ISDN user part</li> <li>• 4 - OMAP (Operation, Maintenance and Administration Part)</li> <li>• 5 - MAP (Mobile Application Part)</li> <li>• 6 - HLR (Home Location Register)</li> <li>• 7 - VLR (Visitor Location Register)</li> <li>• 8 - MSC (Mobile Switching Centre)</li> <li>• 9 - EIC (Equipment Identifier Centre)</li> <li>• 10 - AUC (Authentication Centre)</li> <li>• 11 - ISDN supplementary services</li> <li>• 12 - reserved for international use</li> <li>• 13 - broadband ISDN edge-to-edge applications</li> <li>• 14 - TC test responder</li> </ul>
<b>point-code</b>	Signalling point code (numerical)
<b>gt</b>	Grouping node, contains fields for <i>"Global Title"</i>
<b>tt</b>	Translation type, a numerical value
<b>np</b>	<p>Numbering Plan, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - unknown</li> <li>• 16 - ISDN/telephony numbering plan (Recommendations E.163 and E.164)</li> <li>• 32 - generic numbering plan</li> <li>• 48 - data numbering plan (Recommendation X.121)</li> <li>• 64 - telex numbering plan (Recommendation F.69)</li> <li>• 80 - maritime mobile numbering plan (Recommendations E.210, E.211)</li> <li>• 96 - land mobile numbering plan (Recommendation E.212)</li> <li>• 112 - ISDN/mobile numbering plan (Recommendation E.214)</li> <li>• 224 - private network or network-specific numbering plan</li> </ul>
<b>nai</b>	<p>Nature Of Address, a list of supported constants can be found in section 8.2.3.4; manual numerical equivalents are also available:</p> <ul style="list-style-type: none"> <li>• 0 - unknown</li> <li>• 1 - subscriber number</li> <li>• 2 - reserved for national use</li> <li>• 3 - national significant number</li> <li>• 4 - international number</li> </ul>
<b>address</b>	Address signals (usually digits)

## Acronyms

API Application Programming Interface. 1, 48

AS Application Server. 1, 23, 31

ASP Application Server Process. 1, 23, 31

CDR Call Data Record. 1, 17

CLI pMINK Command Line Interface. 1, 13, 16, 17, 19, 21, 35, 52

CONFIGD pMINK Configuration Daemon. 1, 13–16

FGN Filtering Gateway Node. 1, 13–20, 35, 38, 48, 56, 58

FSM Finite-state machine. 1, 19

LCI Left Component Index. 1, 35, 36, 62

M3UA MTP Level 3 (MTP3) User Adaptation Layer. 1, 22–24, 31, 37, 39, 42, 43, 47, 61, 62, 64

MAP Mobile Application Part. 1, 19, 22, 28, 41, 47, 60

NAI Nature Of Address Indicator. 1, 27, 38

NP Numbering Plan. 1, 27, 30, 32, 39

PDU Protocol Data Unit. 1, 35

pMINK Project MINK Framework. 1, 13, 19, 22, 23, 26, 28, 30, 31, 33, 35–38, 40, 41, 43, 48, 49, 51, 52, 58, 63

R14P Release14 Application Layer Protocol. 1, 13–15, 22, 43, 58, 62, 63

RCI Right Component Index. 1, 35–37, 62

REL Rule Execution Logic. 1, 19

ROUTINGD R14P Routing Daemon. 1, 13–15

RPE Rule Processing Engine. 1, 19, 35, 48, 67

RPROC Rule Processor. 1, 19, 21, 35

RRP Release14 Routing and Rating Protocol. 1, 14, 15, 17

SCCP Signalling Connection Control Part. 1, 22, 38, 40, 42, 47, 61, 62

SCTP Stream Control Transmission Protocol. 1, 22, 23, 43, 63

SGN Signalling Gateway Node. 1, 13, 23, 24, 31, 35, 58, 64

SMPP Short Message Peer-to-Peer. 1, 19, 22, 23, 31, 41, 43, 58

SMS Short Message Service. 1, 16, 19, 29, 30, 34, 38, 39, 56

SMSTPDU Short message TPDU 3GPP TS 23.040. 1, 21, 22, 28, 29, 37, 42, 46, 60

STP Signalling Transfer Point. 1, 13, 19, 22, 58, 62, 63

TCAP Transaction Capabilities Application Part. 1, 22, 26, 27, 36, 37, 41, 47, 59

TCP Transmission Control Protocol. 1, 23, 43

TON Type Of Number. 1, 21, 30, 32, 37

TP-DA TP-Destination-Address. 1

TP-OA TP-Originating-Address. 1, 21, 37

WRR Weighted Round Robin. 1, 17

## Glossary

- ASN.1** Abstract Syntax Notation One (ASN.1) is a standard and notation that describes rules and structures for representing, encoding, transmitting, and decoding data in telecommunications and computer networking. 1, 22, 63
- boolean** In computer science, a Boolean expression is an expression in a programming language that produces a Boolean value when evaluated, i.e. one of true or false. A Boolean expression may be composed of a combination of the Boolean constants true or false, Boolean-typed variables, Boolean-valued operators, and Boolean-valued functions. 1, 35
- bytecode** Bytecode is computer object code that is processed by a program, usually referred to as a virtual machine, rather than by the "real" computer machine, the hardware processor. 1
- bytecode interpreter** A virtual machine, or bytecode interpreter, is a sort of simulation of a computer, usually used to implement a script language. You define an instruction set for this machine according to the needs of your script language. The instructions for the virtual computer are called "bytecodes". 1, 19
- cluster** A cluster consists of one or more nodes which share the same pMINK daemon type. 1
- daemon** Daemon is a computer program that runs as a background process, rather than being under the direct control of an interactive user. 1, 14, 15, 19, 22, 63
- lua** Lua is a lightweight multi-paradigm programming language designed primarily for embedded systems and clients. Lua is cross-platform since it is written in ANSI C, and has a relatively simple C API. 1, 38, 48, 49, 58
- node** A node is a running instance of pMINK daemon, usually belonging to a cluster. 1, 13–15, 17
- perl** Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages. The languages in this family include Perl 5 and Perl 6. 1, 35, 37, 62
- regular expression** In theoretical computer science and formal language theory, a regular expression (sometimes called a rational expression) is a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. "find and replace"-like operations. 1, 37, 62
- routing** In internetworking, the process of moving a packet of data from source to destination. Routing is usually performed by a dedicated device called a router. 1, 19
- rule** A piece of system logic that is evaluated by Rule Processing Engine (RPE). 1, 19, 21, 22, 35, 48, 56, 62
- threat** A circumstance or event that has or indicates the potential to exploit vulnerabilities and to adversely impact (create adverse consequences for) organizational operations, organizational assets (including information and information systems), individuals, other organizations, or society. 1, 19, 48
- variant** A Variant is a special data type that can contain any kind of data. 1, 49
- X.690** X.690 is an ITU-T standard specifying several ASN.1 encoding formats: Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). 1, 22, 63
- zero-based** Zero-based numbering or index origin = 0 is a way of numbering in which the initial element of a sequence is assigned the index 0, rather than the index 1 as is typical in everyday non-programming circumstances. 1, 35