NR Software N3IWF

Version: 2024-12-23

Table of Contents

1 Introduction 1

2 Installation 2

2.1 Linux setup 2

2.1.1 Packages 2

2.1.2 OpenSSL 2

2.1.3 License key installation 2

2.2 LTEN3IWF installation 3

2.2.1 Basic LTEN3IWF configuration 3

2.3 Initial testing 3

3 Command line monitor reference 4

4 Configuration reference 5

4.1 Configuration file syntax 5

4.1.1 JSON merge rules 6

4.2 Properties 7

5 Remote API 15

5.1 Messages 15

5.2 Startup 16

5.3 Errors 17

5.4 Sample nodejs program 17

5.5 Common messages 17

5.6 N3IWF messages 22

6 Log file format 23

6.1 NAS layer 23

6.2 IP layer 23

6.3 NGAP and GTP-U layers 23

7 Change history 25

7.1 Version 2024-12-13 25

7.2 Version 2024-09-13 25

7.3 Version 2024-06-14 25

7.4 Version 2024-03-15 25

7.5 Version 2023-12-15 25

7.6 Version 2023-09-08 25

7.7 Version 2023-06-10 25

7.8 Version 2023-03-17 25

7.9 Version 2022-12-16 26

7.10 Version 2022-09-16 26

7.11 Version 2022-06-17 26

7.12 Version 2022-03-18 26

7.13 Version 2021-12-17 26

7.14 Version 2021-09-17 26

8 License **27**

1 Introduction

LTEN3IWF is a N3IWF (Non-3GPP Interworking element) implementation.

LTEN3IWF interfaces with a 5GS Core Network thru the standard NG interface whilst supporting IPsec connectivity towards the UE.

2 Installation

2.1 Linux setup

2.1.1 Packages

LTEN3IWF uses the SCTP protocol for which the necessary packages are not usually installed. In order to install them, do as root user:

- Fedora

  ```
  dnf install lksctp-tools kernel-modules-extra
  ```

- Ubuntu

  ```
  sudo apt-get install lksctp-tools linux-image-extra-3.13.0-24-generic
  ```

  Note that linux-image-extra package name may differ depending on your kernel version.

To verify that SCTP kernel module is running, do as root user:

```
checksctp
```

If it reports that the protocol is not supported,

- check if you have a /etc/modprobe.d/sctp-blacklist.conf file
- edit it to comment the 'blacklist sctp' line

Then reboot the PC in case the Linux kernel was upgraded too.

2.1.2 OpenSSL

LTEN3IWF has been compiled against openssl version 1.1.1w.
If your system does not have compatible version installed you may have this error message at startup:

```
error while loading shared libraries: libssl.so.1.1: cannot open shared object file: No such file or directory
```

To overcome this problem, you may:

- Copy libssl.so.1.1 and libcrypto.so.1.1 from `libs` subdirectory of your release tarball.
  If you have installed software with automatic install script, this should have been done automatically.
- Compile and install proper openssl version yourself

In case of persisting issue, raise a ticket from our support site at `https://support.amarisoft.com/` with the information provided by below commands executed in LTEN3IWF directory:

```
uname -a
ls -l
ldd ./lten3iwf
openssl version
```

2.1.3 License key installation

LTEN3IWF needs a license key file to run. *It is associated to your PC, so if you replace it or change its hardware configuration you must contact Amarisoft to get a new license key.*

The following steps are needed to get this license file:

- Run LTEN3IWF:

  ```
  ./lten3iwf config/n3iwf.cfg
  ```

  It says that the license key is not present and prints a 16 digit hexadecimal code.

- Send by mail to `delivery@amarisoft.com` this hexadecimal code to your contact at Amarisoft. You will get back the `lten3iwf.key` license key file.
- Copy the `lten3iwf.key` file to the `${HOME}/.amarisoft/` directory (`${HOME}` is the home directory of the `root` user). You can use the shell variable `AMARISOFT_PATH` to change this path.

Once the license key is installed, lten3iwf should start normally.

2.2 LTEN3IWF installation

Decompress the LTEN3IWF archive to a convenient place. The executable `lten3iwf` can be launched from this directory.

2.2.1 Basic LTEN3IWF configuration

The main configuration file is `config/n3iwf.cfg`. It uses a superset of the JSON syntax.

2.3 Initial testing

Customize and start the `lte_init.sh` script as `root` user to configure the network and CPU governors.

Start the LTEMME software as `root` user. `root` priviledges are needed to set up the virtual network interface.

```
./ltemme config/mme.cfg
```

In another terminal, start the LTEN3IWF software as `root` user. `root` priviledges are needed to use real time scheduling priority.

```
./lten3iwf config/n3iwf.cfg
```

The N3IWF is now running. Type `ng` in the command line monitor of LTEN3IWF to verify that it is connected to LTEMME.

3 Command line monitor reference

The following commands are available:

`help` Display the help. Use `help` *`command`* to have a more detailed help about a command.

`ue` List connected UEs.

`ng` Dump the NG connection state. It is useful to see if the N3IWF is connected to the AMF.

`ngconnect [`*`amf_addr`*`]`
Force a NG (re)connection to the AMF. The AMF IP address and optional port can be given as an optional parameter.

`ngdisconnect`
Force a NG disconnect from the AMF.

`ngadd` Adds a new AMF to the list of NGAP connections.
Message definition
The message must contain the same parameters as one of the object defined in amf_list array. See [amf_list], page 9.

`ngdelete` Removes a AMF address from the list of NGAP connections.
Message definition

`addr` String. AMF address to be removed from the list.

4 Configuration reference

4.1 Configuration file syntax

The main configuration file uses a syntax very similar to the Javascript Object Notation (JSON) with few extensions.

1. Supported types:
   - Numbers (64 bit floating point). Notation: `13.4`
   - Complex numbers. Notation: `1.2+3*I`
   - Strings. Notation: `"string"`
   - Booleans. Notation: `true` or `false`.
   - Objects. Notation: `{ field1: value1, field2: value2, .... }`
   - Arrays. Notation: `[ value1, value2, .... ]`
2. The basic operations `+`, `-`, `*` and `/` are supported with numbers and complex numbers. `+` also concatenates strings. The operators `!`, `||`, `&&`, `==`, `!=`, `<`, `<=`, `>=`, `>` are supported too.
3. The numbers `0` and `1` are accepted as synonyms for the boolean values `false` and `true`.
4. {} at top level are optional.
5. " for property names are optional, unless the name starts with a number.
6. Properties can be duplicated.
   If properties are duplicated, they will be merged following [JSON merge rules], page 6, with overriding occuring in reading direction (last overrides previous).
   Ex:

```
{
    value: "foo",
    value: "bar",
    sub: {
        value: "foo"
    },
    sub: {
        value: "bar"
    }
}
```

   Will be equivalent to:

```
{
    value: "bar",
    sub: {
        value: "bar"
    }
}
```

7. Files can be included using *include* keyword (must not be quoted) followed by a string (without :) representing the file to include (path is relative to current file) and terminating by a comma.
   Arrays can't be included.
   Merge will be done as for duplicate properties.
   If *file1.cfg* is:

```
    value: "foo",
    include "file2.cfg",
    foo: "foo"
```

And *file2.cfg* is:

```
value: "bar",
foo: "bar"
```

Final config will be:

```
{
    value: "bar",
    foo: "foo"
}
```

8. A C like preprocessor is supported. The following preprocessor commands are available:

   `#define` *var expr*
   : Define a new variable with value *expr*. *expr* must be a valid JSON expression. Note that unlike the standard C preprocessor, *expr* is evaluated by the preprocessor.

   `#undef` *var*
   : Undefine the variable *var*.

   `#include` *expr*
   : Include the file whose filename is the evaluation of the string expression *expr*.

   `#if` *expr* Consider the following text if *expr* is true.

   `#else` Alternative of `#if` block.

   `#elif` Composition of `#else` and `#if`.

   `#endif` End of `#if` block.

   `#ifdef` *var*
   : Shortcut for `#if defined(var)`

   `#ifndef` *var*
   : Shortcut for `#if !defined(var)`

   In the JSON source, every occurrence of a defined preprocessor variable is replaced by its value.

9. Backquote strings: JSON expression can be inserted in backquote delimited strings with the `${expr}` syntax. Example: `` `abc${1+2}d` `` is evaluated as the string `"abc3d"`. Preprocessor variables can be used inside the expression. Backquote strings may span several lines.

4.1.1 JSON merge rules

Merge overriding direction depends on context, i.e source may override destination or the opposite.
JSON merge is recursive for Objects and Arrays.

Example, merging

```
{
    foo: { value: "bar" },
    same: "one",
    one: 1
}
```

with

```
{
    foo: { value: "none", second: true },
```

```
    same: "two",
    two: 1
}
```

Will become:

```
{
    foo: { value: "bar", second: true },
    same: "one",
    one: 1
    two: 1
}
```

assuming first object overrides second one.

In case of Array merging, the final array length will be the maximum length of all merged arrays.
For each element of the final array, merge will be done considering defined elements only.
Ex:

```
{
    array: [0, 1, 2, { foo: "bar" } ],
    array: [3, 4],
    array: [5, 6, 7, { bar: "foo" }, 8 ]
}
```

Will be merged to:

```
{
    array: [5, 6, 7, { foo: "bar", bar: "foo" }, 8 ],
}
```

4.2 Properties

`log_filename`
: String. Set the log filename. If no leading `/`, it is relative to the configuration file path. See [Log file format], page 22.

`log_options`
: String. Set the logging options as a comma separated list of assignments.

- *layer*.level=*verbosity*. For each layer, the log verbosity can be set to `none`, `error`, `info` or `debug`. In debug level, the content of the transmitted data is logged.
- *layer*.max_size=*n*. When dumping data content, at most `n` bytes are shown in hexa. For ASN.1, NAS or Diameter content, show the full content of the message if `n > 0`.
- *layer*.payload=[0|1]. Dump ASN.1, NAS, SGsAP or Diameter payload in hexadecimal.
- *layer*.key=[0|1]. Dump security keys (NAS and RRC layers).
- *layer*.crypto=[0|1]. Dump plain and ciphered data (NAS and PCDP layers).
- *layer*.verbose=[0|1]. If `layer` is `ipsec`, dump all packets filtering informations.
- time=[sec|short|full]. Display the time as seconds, time only or full date and time (default = time only).
- time.us=[0|1]. Dump time with microseconds precision.
- file=cut. Close current file log and open a new one.

- file.rotate=now. Rename current log with timestamp and open new one.
- file.rotate=*size*. Rename current log every time it reaches *size* bytes open new one. Size is an integer and can be followed by K, M or G.
- file.path=*path*. When log rotation is enabled, move current log to this path instead of initial log path.
- append=[0|1]. (default=0). If 0, truncate the log file when opening it. Otherwise, append to it.

Available layers are: `nas`, `ip`, `gtpu`, `ngap`, `n3iwf`, `ikev2`, `ipsec`

`log_sync` Optional boolean (default = false). If true, logs will be synchronously dumped to file.
Warning, this may lead to performances decrease.

`com_addr` Optional string. Address of the WebSocket server remote API. See [Remote API], page 14.
If set, the WebSocket server for remote API will be enabled and bound to this address.
Default port is 9011.
Setting IP address to [::] will make remote API reachable through all network interfaces.

`com_name` Optional string. Sets server name. N3IWF by default

`com_ssl_certificate`
Optional string. If set, forces SSL for WebSockets. Defines CA certificate filename.

`com_ssl_key`
Optional string. Mandatory if *com_ssl_certificate* is set. Defines CA private key filename.

`com_ssl_peer_verify`
Optional boolean (default is false). If *true*, server will check client certificate.

`com_ssl_ca`
Optional string. Set CA certificate. In case of peer verification with self signed certificate, you should use the client certificate.

`com_log_lock`
Optional boolean (default is false). If *true*, logs configuration can't be changed via `config_set` remote API.

`com_log_us`
Optional boolean (default is false). If *true*, logs sent by `log_get` remote API response will have a `timestamp_us` parameters instead of `timestamp`

`com_auth` Optional object. If set, remote API access will require authentication.
Authentication mechanism is describe in [Remote API Startup], page 16, section.

`passfile` Optional string. Defines filename where password is stored (plaintext). If not set, `password` must be set

`password` Optional string. Defines password.
If not set, `passfile` must be set.

`unsecure` Optional boolean (default false). If set, allow password to be sent plaintext.
NB: you should set it to true if you access it from a Web Browser (Ex:

Amarisoft GUI) without SSL (https) as your Web Browser may prevent secure access to work.

`com_log_count`
: Optional number (Default = 8192). Defines number of logs to keep in memory before dropping them.
Must be between 4096 and 2097152).

`sim_events`
: Array of object. Each element gives an event configuration to execute for this UE. Event configuration is exactly the same as for [Remote API], page 14, messages except that message field must be event.

`sim_events_loop_count`
: If set, will define `loop_count` for each event of `sim_events`, See [loop_count], page 15.

`sim_events_loop_delay`
: If set, will define `loop_delay` for each event of `sim_events`, See [loop_delay], page 15.

`gtp_addr`
: String. Set the IP address (and optional port) on which the GTP-U packets are received. The default port is 2152. It is normally the IP address of the network interface connected to the core network.

`gtp_payload_mtu`
: Optional integer (range 68 to 16384, default = 1500). MTU in bytes for the GTP-U payload. Do not forget to update the network interface MTU accordingly for optimal performance. For example with a GTP MTU of 1500 bytes, interface should have a MTU of at least 1564 bytes.

`gtp_use_packet_bundling`
: Optional boolean (default = false). Concatenate multiple GTP-U PDUs within a single UDP datagram. Be careful, this is a non-standard option that must not be activated if the peer is not an Amarisoft AMF with this option activated.

`amf_list` Array of objects. List of AMF to which the N3IWF is connected. Each object contains the following properties:

`amf_addr`
: String. Set the IP address (and optional port) of NGAP SCTP connection to the AMF. The default port is 38412.

`gtp_ext_addr`
: Optional string. Set the IP address on which the Core Network should transmit the GTP-U packets. It is the same as gtp_addr by default. It can be different if the N3IWF is behind a NAT.

`ngap_bind_addr`
: Optional string. IP address and optional port on which the NGAP SCTP connection is bound.

`5qi_dscp_mapping`
: Optional array of objects. Allows to define a specific IP differentiated services code point for a given 5QI. 5QI not explicitly configured use the default DSCP value 0.
Each object must contain the following properties:

`5qi` Integer (range 1 to 254). 5QI value.

`dscp` Integer (range 0 to 63). DSCP value.

`backup_amf_addr`
Optional string. Defines the IP address (and optional port) of the backup AMF to be used if the NG connection is not established with the current AMF. If the NG connection is established, the backup AMF will not be used. There must be a corresponding object for the backup AMF in the `amf_list` array.

`priority` Optional integer (range 0 to 1, default 0). Defines the priority of a given AMF. When performing AMF selection, if no candidate is found with priority n, the candidates with priority n+1 are tested.

`n3iwf_id` Integer in range 0-0xFFFF. The N3IWF global identifier.

`n3iwf_name`
Optional string. Set N3IWF name used in NG connection setup request.

`plmn_list`
List of objects. List of PLMNs and NPNs supported. The total number of PLMNs (identified by a PLMN identity in plmn) and SNPNs (identified by a PLMN identity and a NID in snpn) shall not exceed 12. Each object contains the following properties:

`plmn` String or array of strings. PLMN (5 or 6 digits). The array can contain up to 12 PLMNs.

`snpn` Optional array of 1 to 12 objects. List of Stand-Alone Non-Public Network.
Each element contains the following parameters:

`plmn` PLMN string (5 or 6 digits).

`nid_list` Array of NID as defined in 23.003 12.7 Stand-Alone Non-Public Network Identifier an contains the following parameters. Each element contains the following parameters:

`nid_value`
String (10 hexadecimal digits). NID value.

`assignment_mode`
Optional enumeration ("self", "coordinated_1", "coordinated_2"). Default value is "self". Each combination of a PLMN and NID identifies a Stand-Alone Non-Public Network.

`tac` Integer (range 0 to 16777215). Tracking Area Code of the cell.

`nssai` Optional array. List of supported S-NSSAIs.
Default content is sst: 1 (eMBB).
Each entry will set a S-NSSAI value as defined below:

`sst` Integer (range 0-255). Slice Service Type.

`sd` Optional integer (range 0-0xFFFFFE). Slice Differentiator.

`remote_ip_config`
Object describing the IP allocation of the UE inner address as defined in 3GPP TS 23.502. Contains the following properties:

`first_ip_addr`
String. First available IPv4 address.

`last_ip_addr`
: String. Last available IPv4 address.

`ipv4_auto_increment`
: Optional boolean (default = false). If set to false, the same IPv4 address is allocated for successive UE connection/disconnection. If set to true, the IPv4 address is incremented for UE connection/disconnection.

`ip_addr_shift`
: Optional integer (default = 0). The allocated IPv4 addresses are allocated starting from `first_ip_addr` with a difference of `2^ip_addr_shift`. Hence `last_ip_addr - first_ip_addr` must be a multiple of `2^ip_addr_shift`. This option can be useful in case of inter-UE communication to ensure that the IPv4 address of a given UE is the only one in its netmask.

`local_ip_config`
: Object describing the IP allocation of the UP_IP_ADDRESS associated with a child Sa as defined in 3GPP TS 23.502. Contains the following properties:

`first_ip_addr`
: String. First available IPv4 address.

`last_ip_addr`
: String. Last available IPv4 address.

`ipv4_auto_increment`
: Optional boolean (default = false). If set to false, the same IPv4 address is allocated for successive UE connection/disconnection. If set to true, the IPv4 address is incremented for UE connection/disconnection.

`ip_addr_shift`
: Optional integer (default = 0). The allocated IPv4 addresses are allocated starting from `first_ip_addr` with a difference of `2^ip_addr_shift`. Hence `last_ip_addr - first_ip_addr` must be a multiple of `2^ip_addr_shift`. This option can be useful in case of inter-UE communication to ensure that the IPv4 address of a given UE is the only one in its netmask.

`nas_ip_addr`
: Optional string. Address of the local TCP server for NAS signalling. If not present, the first IP of the subnet (See [local_ip_config], page 11) will be used.

`nwu`
: Configuration of the NWu connection. This object contains the following properties:

  `bind_addr`
  : IP address on which the NWu connection is bound.

  `private_key`
  : String. Defines the N3IWF private key filename.

  `certificate`
  : String. Defines the N3IWF certificate filename. The default files n3iwf_private_key.pem and n3iwf_cert.pem are built for N3IWF FQDN "n3iwf.5gc.mnc001.mcc001.pub.3gppnetwork.org" following the procedure decribed below. For another N3IWF FQDN, these files shall be re-built by setting the FQDN in subjectAltName field. Procedure to generate and check the private key file n3iwf_private_key.pem and the certificate file n3iwf_cert.pem:

    ```
    openssl genrsa -out ca.key 2048
    ```

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt
openssl req -newkey rsa:2048 -nodes -keyout n3iwf_private_key.pem
openssl x509 -req -extfile <(printf "subjectAltName=DNS:n3iwf.5gc
openssl x509 -in n3iwf_cert.pem -text
openssl rsa -in n3iwf_private_key.pem -text
```

`esp_duration`
: Optional integer in range 10 to 5*3600 (default = 300). Gives the duration in seconds of the ESP-Sa.

`ike_duration`
: Optional integer in range 20 to 48*3600 (default = 24*3600). Gives the duration in seconds of the IKE-Sa.

`omit_auth_in_first_auth_rsp`
: Optional boolean (default = false). If set, configures the N3IWF to not send the AUTH payload in the first IKE_AUTH exchange.

`ike_encryption_algo_list`
: Optional list of IKE-Sa supported encryption algorithms "aes-cbc-128" (AES CBC 128 bits key length), "aes-cbc-192" (AES CBC 192 bits key length), "aes-cbc-256" (AES CBC 256 bits key length), "aes-gcm-128-16" (AES GCM 128 bits key length and 16 bytes ICV), "aes-gcm-256-16" (AES GCM 256 bits key length and 16 bytes ICV), "des", "3des", "blowfish", "aes-ctr-128" (AES CTR 128 bits key length), "aes-ctr-192" (AES CTR 192 bits key length), and aes-ctr-256 (AES CTR 256 bits key length) ordered from most preferred to least preferred.
Default value is ["aes-cbc-128", "aes-cbc-192", "aes-cbc-256", "aes-gcm-128-16", "aes-gcm-256-16", "des", "3des", "blowfish", "aes-ctr-128", "aes-ctr-192", "aes-ctr-256"].

`ike_integrity_algo_list`
: Optional list of IKE-Sa supported integrity algorithms "hmac-sha-1-96", "hmac-sha-1-160", "hmac-sha-256-128", "hmac-sha-384-192", "hmac-sha-512-256", "hmac-md5-96", "hmac-md5-128" and "aes-cmac-96" ordered from most preferred to least preferred.
Default value is ["hmac-sha-1-96", "hmac-sha-1-160", "hmac-sha-256-128", "hmac-sha-384-192", "hmac-sha-512-256", "hmac-md5-96", "hmac-md5-128", "aes-cmac-96"];

`ike_prf_list`
: Optional list of IKE-Sa supported pseudo-random functions "prf-hmac-sha1", "prf-hmac-sha2-256", "prf-hmac-sha2-384", "prf-hmac-sha2-512" and "prf-hmac-md5" ordered from most preferred to least preferred.
Default value is ["prf-hmac-sha1", "prf-hmac-sha2-256", "prf-hmac-sha2-384, "prf-hmac-sha2-512", "prf-hmac-md5"].

`ike_dh_group_list`
: Optional list of IKE-Sa supported Diffie-Hellman groups "group_1", "group_2", "group_5", "group_14", "group_15", "group_16", "group_17", "group_18", "group_19", "group_22", "group_23" and "group_24" ordered from most preferred to least preferred.
Default value is ["group_5", "group_14", "group_15", "group_16", "group_17", "group_18", "group_19", "group_22", "group_23", "group_24"].

`esp_encryption_algo_list`
: Optional list of ESP-Sa supported encryption algorithms "null", "aes-cbc-128" (AES CBC 128 bits key length), "aes-cbc-192" (AES CBC 192 bits key length), "aes-cbc-256" (AES CBC 256 bits key length), "des", "3des", "blowfish", "aes-ctr-128" (AES CTR 128 bits key length), "aes-ctr-192" (AES CTR 192 bits key length), and aes-ctr-256 (AES CTR 256 bits key length) ordered from most preferred to least preferred.
Default value is ["null", "aes-cbc-128", "aes-cbc-192", "aes-cbc-256", "des", "3des", "blowfish", "aes-ctr-128", "aes-ctr-192", "aes-ctr-256].

`esp_integrity_algo_list`
: Optional list of ESP-Sa supported integrity algorithms "null", "hmac-sha-1-96", "hmac-sha-1-160", "hmac-sha-256-128", "hmac-sha-384-192", "hmac-sha-512-256", "hmac-md5-96", "hmac-md5-128" and "aes-cmac-96" ordered from most preferred to least preferred.
Default value is ["null", "hmac-sha-1-96", "hmac-sha-1-160", "hmac-sha-256-128", "hmac-sha-384-192", "hmac-sha-512-256", "hmac-md5-96", "hmac-md5-128", "aes-cmac-96"].

`esp_dh_group_list`
: Optional list of ESP-Sa supported Diffie-Hellman groups "none", "group_1", "group_2", "group_5", "group_14", "group_15", "group_16", "group_17", "group_18", "group_19", "group_22", "group_23" and "group_24" ordered from most preferred to least preferred.
This list is used for rekeying ESP-Sa. Default value is ["none", "group_5", "group_14", "group_15", "group_16", "group_17", "group_18", "group_19", "group_22", "group_23", "group_24"].

`dpd_timer_value`
: Optional integer in range 5 to 300 (default = 300). Gives the "dead peer detection" timer value in seconds.

`mobike`
: Optional boolean (default = true). Indicates MOBIKE support.

`dont_fragment`
: Optional boolean (default = true) used to enable/disable the fragmentation of the ESP packets.

`ike_generate_error`
: Optional object. Allows to ignore a message or generate an error during the initial exchanges.
It contains the following objects:

  `exchange`
  : String. Gives the exchange to ignore or on which the error must be sent. Possible values are "none", "ike_sa_init", "ike_auth_step1", "ike_auth_step2", "ike_auth_step3", "dpd".

  `error`
  : Optional integer. Gives the value of 'Notify Message Type' to send in the Notify payload rejecting the exchange.
  It present, the message received during the exchange will be rejected.
  If absent, the message received during the exchange will be ignored.

  Example:

```
ike_generate_error: {
```

```
error: 9002,
exchange: "ike_auth_step1"
}
```

5 Remote API

You can access LTEN3IWF via a remote API.
Protocol used is WebSocket as defined in RFC 6455 (`https://tools.ietf.org/html/rfc6455`).

Note that Origin header is mandatory for the server to accept connections.
This behavior is determined by the use of `nopoll` library.
Any value will be accepted.

5.1 Messages

Messages exchanged between client and LTEN3IWF server are in strict JSON format.

Each message is represented by an object. Multiple message can be sent to server using an array of message objects.

Time and delay values are floating number in seconds.

There are 3 types of messages:

- Request

  Message sent by client.
  Common definition:

  `message` String. Represent type of message. This parameter is mandatory and depending on its value, other parameters will apply.

  `message_id`
  Optional any type. If set, response sent by the server to this message will have same message_id. This is used to identify response as WebSocket does not provide such a concept.

  `start_time`
  Optional float. Represent the delay before executing the message.
  If not set, the message is executed when received.

  `absolute_time`
  Optional boolean (default = false). If set, `start_time` is interpreted as absolute.
  You can get current clock of system using `time` member of any response.

  `standalone`
  Optional boolean (default = false). If set, message will survive WebSocket disconnection, else, if socket is disconnected before end of processing, the message will be cancelled.

  `loop_count`
  Optional integer (default = 0, max = 1000000). If set, message will be repeated `loop_count` time(s) after `loop_delay` (From message beginning of event).
  Response will have a `loop_index` to indicate iteration number.

  `loop_delay`
  Optional number (min = 0.1, max = 86400). Delay in seconds to repeat message from its `start_time`. Mandatory when `loop_count` is set > 0.

- Response

  Message sent by server after any request message as been processed.
  Common definition:

  `message` String. Same as request.

  `message_id`
  Optional any type. Same as in request.

  `time` Number representing time in seconds since start of the process.
  Usefull to send command with absolute time.

  `utc` Number representing UTC seconds.

- Events

  Message sent by server on its own initiative.
  Common definition:

  `message` String. Event name.

  `time` Number representing time in seconds.
  Usefull to send command with absolute time.

5.2 Startup

When WebSocket connections is setup, LTEN3IWF will send a first message with name set to `com_name` and type set to N3IWF.

If authentication is not set, message will be `ready`:

```
{
    "message": "ready",
    "type": "N3IWF",
    "name": <com_name>,
    "version": <software version>,
    "product": <Amarisoft product name (optional)>
}
```

If authentication is set, message will be `authenticate` :

```
{
    "message": "authenticate",
    "type": "N3IWF",
    "name": <com_name>,
    "challenge": <random challenge>
}
```

To authenticate, the client must answer with a `authenticate` message and a `res` parameter where:

```
res = HMAC-SHA256( "<type>:<password>:<name>", "<challenge>" )
```

`res` is a string and HMAC-SHA256 refers to the standard algorithm (`https://en.wikipedia.org/wiki/HMAC`)

If the authentication succeeds, the response will have a `ready` field set to `true`.

```
{
    "message": "authenticate",
    "message_id": <message id>,
    "ready": true
}
```

If authentication fails, the response will have an `error` field and will provide a new challenge.

```
{
    "message": "authenticate",
    "message_id": <message id>,
    "error": <error message>,
    "type": "N3IWF",
    "name: <name>,
    "challenge": <new random challenge>
}
```

If any other message is sent before authentication succeeds, the error `"Authentication not done"` will be sent as a response.

5.3 Errors

If a message produces an error, response will have an error string field representing the error.

5.4 Sample nodejs program

You will find in this documentation a sample program: `ws.js`.
It is located in `doc` subdirectory.
This is a nodejs program that allow to send message to LTEN3IWF.
It requires nodejs to be installed:

```
dnf install nodejs npm
npm install nodejs-websocket
```

Use relevant package manager instead of NPM depending on your Linux distribution.

Then simply start it with server name and message you want to send:

```
./ws.js 127.0.0.1:9011 '{"message": "config_get"}'
```

5.5 Common messages

`config_get`
: Retrieve current config.

  Response definition:

  `type` Always "N3IWF"

  `name` String representing server name.

  `logs` Object representing log configuration.
  With following elements:

  `layers` Object. Each member of the object represent a log layer configuration:

  `layer name`
  : Object. The member name represent log layer name and parameters are:

    `level` See [log_options], page 7,

    `max_size` See [log_options], page 7,

    `key` See [log_options], page 7,

`crypto` See [log_options], page 7,

`payload` See [log_options], page 7,

`verbose` Optional boolean. See [log_options], page 7,

`count` Number. Number of bufferizer logs.

`rotate` Optional number. Max log file size before rotation.

`path` Optional string. Log rotation path.

`bcch` Boolean. True if BCCH dump is enabled (eNB only).

`mib` Boolean. True if MIB dump is enabled (eNB only).

`locked` Optional boolean. If `true`, logs configuration can't be changed with `config_set` API.

`config_set`

Change current config.
Each member is optional.
Message definition:

`logs` Optional object. Represent logs configuration. Same structure as config_get (See [config_get logs member], page 17).
All elements are optional.
Layer name can be set to `all` to set same configuration for all layers.
If set and logs are locked, response will have `logs` property set to `locked`.

`nwu` Optional object allowing to configure N3IWF options. It may contain the following object:

`esp_duration`
Optional integer in range 10 to 5*3600 (default = 300). Gives the duration in seconds of the ESP-Sa.

`ike_duration`
Optional integer in range 20 to 48*3600 (default = 24*3600). Gives the duration in seconds of the IKE-Sa.

`mobike` Optional boolean. Indicates MOBIKE support.

`dont_fragment`
Optional boolean used to enable/disable the fragmentation of the ESP packets.

`ike_generate_error`
Optional object. Allows to ignore a message or generate an error during the initial exchanges.
It contains the following objects:

`exchange` String. Gives the exchange to ignore or on which the error must be sent. Possible values are "none", "ike_sa_init", "ike_auth_step1", "ike_auth_step2", "ike_auth_step3".

`error` Optional integer. Gives the value of 'Notify Message Type' to send in the Notify payload rejecting the exchange.

It present, the message received during the exchange will be rejected.
If absent, the message received during the exchange will be ignored.

`log_get` Get logs.
This API has a per connection behavior. This means that the response will depend on previous calls to this API within the same WebSocket connection.
In practice, logs that have been provided in a response won't be part of subsequent request unless connection is reestablished. To keep on receiving logs, client should send a new `log_get` request as soon as the previous response has been received.
If a request is sent before previous request has been replied, previous request will be replied right now without considering specific min/max/timeout conditions.

Message definition:

`min` Optional number (default = 1). Minimum amount of logs to retrieve. Response won't be sent until this limit is reached (Unless timeout occurs).

`max` Optional number (default = 4096). Maximum logs sent in a response.

`timeout` Optional number (default = 1). If at least 1 log is available and no more logs have been generated for this time, response will be sent.

`allow_empty`
Optional boolean (default = false). If set, response will be sent after timeout, event if no logs are available.

`rnti` Optional number. If set, send only logs matching rnti.

`ue_id` Optional number. If set, send only logs with matching ue_id.

`layers` Optional Object. Each member name represents a log layer and values must be string representing maximum level. See [log_options], page 7.
If *layers* is not set, all layers level will be set to *debug*, else it will be set to *none*.
Note also the logs is also limited by general log level. See [log_options], page 7.

`short` Optional boolean (default = false). If set, only first line of logs will be dumped.

`headers` Optional boolean. If set, send log file headers.

`start_timestamp`
Optional number. Is set, filter logs older than this value in milliseconds.

`end_timestamp`
Optional number. Is set, filter logs more recent than this value in milliseconds.

`max_size` Optional number (default = 1048576, i.e. 1MB). Maximum size in bytes of the generated JSON message. If the response exceeds this size, the sending of logs will be forced independently from other parameters.

Response definition:

`logs` Array. List of logs. Each item is a an object with following members:

`data` Array. Each item is a string representing a line of log.

`timestamp`
: Number. Milliseconds since January 1st 1970. Not present if `com_log_us` is set in configuration.

`timestamp_us`
: Number. Microseconds since January 1st 1970. Only present if `com_log_us` is set in configuration.

`layer` String. Log layer.

`level` String. Log level: *error*, *warn*, *info* or *debug*.

`dir` Optional string. Log direction: *UL*, *DL*, *FROM* or *TO*.

`ue_id` Optional number. UE_ID.

`cell` Optional number (only for PHY layer logs). Cell ID.

`rnti` Optional number (only for PHY layer logs). RNTI.

`frame` Optional number (only for PHY layer logs). Frame number (Subframe is decimal part).

`channel` Optional string (only for PHY layer logs). Channel name.

`src` String. Server name.

`idx` Integer. Log index.

`headers` Optional array. Array of strings.

`discontinuity`
: Optional number. If set, this means some logs have been discarded due to log buffer overflow.

`microseconds`
: Optional boolean. Present and set to true if `com_log_us` is set in configuration file.

`log_set` Add log.
Message definition:

`log` Optional string. Log message to add. If set, *layer* and *level* are mandatory.

`layer` String. Layer name. Only mandatory if *log* is set.

`level` String. Log level: *error*, *warn*, *info* or *debug*. Only mandatory if *log* is set.

`dir` Optional string. Log direction: *UL*, *DL*, *FROM* or *TO*.

`ue_id` Optional number. UE_ID.

`flush` Optional boolean (default = false). If set, flushes fog file.

`rotate` Optional boolean (default = false). If set, forces log file rotation.

`cut` Optional boolean (default = false). If set, forces log file reset.

`log_reset`
: Resets logs buffer.

`license` Retrieves license file information.

`quit` Terminates lten3iwf.

`help` Provides list of available messages in *messages* array of strings and events to register in *events* array of strings.

`stats` Report statistics for LTEN3IWF.
Every time this message is received by server, statistics are reset.
Warning, calling this message from multiple connections simultaneously will modify the statistics sampling time.

Response definition:

`cpu` Object. Each member name defines a type and its value cpu load in % of one core.

`instance_id`
Number. Constant over process lifetime. Changes on process restart.

`ipsec` Report ipsec SAs.

Response definition:

`SAs` Array. List of object representing a security association witth following definition:

`type` String. IP version, can be `IPv4` or `IPv6`.

`dir` String. Direction, can be `in` or `out`.

`spi` Number. SPI.

`ue_id` Number. Associated `ue_id`.

`mode` String. ESP type, can be `tunnel` or `transport`

`src` String. Source IP address.

`dst` String. Destination IP address.

`tun_src` Optional string. Tunnel source IP address.

`tun_dst` Optional string. Tunnel destination IP address.

`src_prefix`
Number. Source network prefix.

`dst_prefix`
Number. Destination network prefix.

`authent_key`
String. Authentication key in hexadecimal form (Empty string authentication is disabled).

`cipher_key`
String. Ciphering key in hexadecimal form (Empty string ciphering is disabled).

5.6 N3IWF messages

`ng` Get AMF link state.
Response definition:

`ng_list` Array of object. One for each AMF connection defined as follow:

`state` Link state: *disconnected*, *connecting*, *connected*, *inactive* or *setup_done*.

`address` AMF address.

`name` AMF name.

`PLMN` If connection complete, PLMN.

`ngconnect`
Forces connection to an AMF.
Message definition

`address` Optional string. If not set, will try to connect to all registered AMF, else will try with the specified address.

`ngdisconnect`
Forces disconnection from an AMF.
Message definition

`address` Optional string. If not set, will to disconnect from all registered AMF, else will try with the specified address.

`ngadd` Adds a new AMF to the list of NGAP connections.
Message definition
The message must contain the same parameters as one of the object defined in amf_list array. See [amf_list], page 9.

`ngdelete` Removes a AMF address from the list of NGAP connections.
Message definition

`addr` String. AMF address to be removed from the list.

`ue_ctx_rel`
Forces a UE context release.
Message definition:

`ran_ue_id`
Integer. RAN UE id.

6 Log file format

6.1 NAS layer

When a NAS message is dumped, the format is:

```
time layer - message
```

When a NAS data PDU is dumped (debug level), the format is:

```
time layer dir MME_UE_ID message_type
        long_content
```

`time` Time using the selected format

`layer` Indicate the layer (`[NAS]` here).

`dir` UL (uplink) or DL (downlink).

`MME_UE_ID`
MME S1AP UE identifier (hexadecimal).

`message_type`
NAS message type.

`long_content`
Full content of the NAS message if `nas.max_size > 0`.

6.2 IP layer

When a IP data PDU is dumped (debug level), the format is:

```
time layer dir short_content
        long_content
```

`time` Time using the selected format

`layer` Indicate the layer (`[IP]` here).

`dir` UL (uplink) or DL (downlink).

`short_content`
Single line content (at least the IP protocol and the source and destination address).

`long_content`
Optional hexadecimal dump of the PDU if `ip.max_size > 0`.

6.3 NGAP and GTP-U layers

When a message is dumped, the format is:

```
time layer - message
```

When a data PDU is dumped (debug level), the format is:

```
time layer dir ip_address short_content
        long_content
```

`time` Time using the selected format.

`layer` Indicate the layer (`[NGAP]` or `[GTPU]` here).

`dir` Direction: `TO` or `FROM`.

`ip_address`
source or destination IP address, depending on the `dir` field.

`short_content`
: Single line content.

`long_content`
: - NGAP: full ASN.1 content of the message if `layer.max_size > 0`.
  - GTPU: hexadecimal dump of the message if `layer.max_size > 0`.

7 Change history

7.1 Version 2024-12-13

- NGAP ASN.1 is updated to v18.3.0

7.2 Version 2024-09-13

- added `license` remote API
- `mobike` parameter is added in `nwu` object and `config_set` remote API
- `dont_fragment` parameter is added to `nwu` configuration object and `config_set` remote API
- `encr-null-auth-aes-gmac-128`, `encr-null-auth-aes-gmac-192` and `encr-null-auth-aes-gmac-256` values are added to `esp_encryption_algo_list`
- `com_logs_lock` parameter is renamed to `com_log_lock`. `com_logs_lock` is still supported for backward compatibility
- added `com_log_us` parameter

7.3 Version 2024-06-14

- OpenSSL library is upgraded to 1.1.1w
- added `backup_amf_addr` and `priority` parameters to `amf_list` object

7.4 Version 2024-03-15

- added MOBIKE support
- added more remote APIs documentation
- added AMF name to `ng` monitor command

7.5 Version 2023-12-15

- added `loop_count` and `loop_delay` to remote API messages
- added `sim_events`, `sim_events_loop_count` and `sim_events_loop_delay`
- added `com_ssl_ca` parameter for SSL verification

7.6 Version 2023-09-08

- NGAP ASN.1 is updated to v17.5.0
- `gtp_use_packet_bundling` parameter is added for GTP-U PDUs bundling support
- `ipsec` remote API added

7.7 Version 2023-06-10

- NGAP ASN.1 is updated to v17.4.0
- `com_logs_lock` parameter added to disable logs configuration change via remote API

7.8 Version 2023-03-17

- `com_addr` parameter now uses [::] address instead of 0.0.0.0 in the delivered configuration file to allow IPv6 connection

7.9 Version 2022-12-16

- NGAP ASN.1 is updated to v17.2.0
- added new IKE-Sa and ESP-Sa algorithms
- added `snpn` parameter to `plmn_list` object for NPN support
- added `dpd` value to `exchange` parameter
- added `utc` parameter to remote API response messages

7.10 Version 2022-09-16

- "ipsec debug" monitor is now deprecated. Set ipsec.verbose to 1 in log configuration
- added `dpd_timer_value` parameter

7.11 Version 2022-06-17

- OpenSSL library is upgraded to 1.1.1n
- added new IKE-Sa and ESP-Sa algorithms and groups
- added `start_timestamp` and `end_timestamp` to `log_get` API
- added `ike_duration` parameter
- `esp_duration` and `ike_duration` parameters can be changed with `config_set` API

7.12 Version 2022-03-18

- `ike_generate_error` configuration object is added
- added NAT traversal support

7.13 Version 2021-12-17

- `ike_encryption_algo_list`, `ike_integrity_algo_list`, `ike_prf_list`, `ike_dh_group_list`, `esp_encryption_algo_list`, `esp_integrity_algo_list` and `esp_dh_group_list` parameters are added to make the list of N3IWF supported algorithms configurable
- `license` monitor command is added

7.14 Version 2021-09-17

- Initial release

8 License