**3GPP TSG-SA5 Meeting #158 *S5-247076***

Orlando, USA, 18 – 22 November 2024

**Source: Ericsson, Vodafone, Deutsche Telekom, Telecom Italia, Rakuten, Nokia**

**Title: Signalling traffic monitoring Report Format of the drafted TS28.abc**

**Document for: Approval**

**Agenda Item: 6.19.22**

# 1 Decision/action requested

***For approval.***

# 2 References

[1] S5-245981 new WID signalling monitoring

[2] S5-245336, initial skeleton of draft TS28.abc signalling monitoring

# 3 Rationale

Defines Signalling traffic monitoring management report format

# 4 Detailed proposal

\*\*\* START OF NEXT CHANGE \*\*\*

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[x7] IETF RFC8086: GRE-in-UDP Encapsulation

[x8] IETF draft-ietf-opsawg-pcapng-04: PCAP Next Generation (pcapng) Capture File Format

\*\*\* START OF NEXT CHANGE \*\*\*

## 7.1 Reporting format

### 7.1.1 Protocol

The STM report shall be sent as payload on either UDP or TCP transport protocols based on operator’s policy.

### 7.1.2 Format of the STM reports

The STM report may be formatted based on operator policy by UDP – GRE – PCAPNG encapsulation or by a generic Type – Length – Value encoding.

#### 7.1.2.1 UDP-GRE-PCAPNG Encapsulation

Encoding of STM Payloads shall be performed using GRE in UDP Encapsulation as specified in IETF RFC8086 [x7]. The packet is encapsulated as IP header, UDP header, GRE header, and STM Payload.

The encapsulated STM Payload is shown in Figure 7.1.2-1.

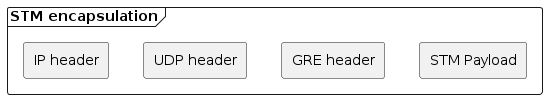


Figure 7.1.2.1-1 STM encapsulation



The STM payload contains the PCAPNG payload (IETF draft-ietf-opsawg-pcapng-04 [x8]). The PCAPNG payload contains PCAPNG header and the collected signalling messages. The STM payload has following attributes, as specified in the Table 7.1.2.1-2.

Table 7.1.2.1-2: STM Payload

|  |  |
| --- | --- |
| STM Payload attribute name | Description |
|  |  |
| PCAPNG Payload (M) | PCAPNG header and the collected signalling messages.  The collected signalling message is sent before security encapsulation, or received after security decapsulation. |

Editor’s note: the identification of the peer NFs of the message is to be determined.

#### 7.1.2.2 Generic Type – Length- Value encoding

Depending on operator policy the STM payload shall be transferred over UDP or TCP.

The STM payload is formatted as generic Type - Length - Value (TLV) encoding.

Each message starts with four bytes protocol ID, followed by two bytes to indicate the number of bytes of the whole message (including protocol ID and length). The rest of the message consists of a non-empty list of data chunks.

Each data chunk is composed by a chunk type, the total lengths of the chunk, and the payload of the chunk.

Each chunk type reflects an information element that is used by the various signalling protocols on the different protocol layers.

Editor’s note: The set of type definitions to be used for type - length – encoding of captured signalling messages used in telecommunication systems is to be determined.

\*\*\* START OF NEXT CHANGE \*\*\*

# Annex B (informative): Plant UML source code

## B.4 STM encapsulation

The following PlantUML source code is used to describe STM encapsulation. As depicted by Figure 7.1.2.-1:

@startuml

frame "STM encapsulation" {

rectangle "GRE header" as GRE

rectangle "STM Payload" as Payload

rectangle "UDP header" as UDP

rectangle "IP header" as IP

IP-[hidden]>UDP

UDP-[hidden]>GRE

GRE-[hidden]>Payload

}

@enduml

\*\*\* END OF CHANGE \*\*\*