

3GPP2 X.S0004-520-E

v 2.0

Date: July 2007



3RD GENERATION
PARTNERSHIP
PROJECT 2
"3GPP2"

Mobile Application Part (MAP) - TCAP APPLICATION SIGNALING PROTOCOLS

COPYRIGHT

3GPP2 and its Organizational Partners claim copyright in this document and individual OPs may copyright and issue documents or standards publications in individual Organizational Partner's name based on this document. Requests for reproduction of this document should be directed to the 3GPP2 Secretariat at secretariat@3gpp2.org. Requests to reproduce individual Organizational Partner's documents should be directed to that Organizational Partner. See www.3gpp2.org for more information.

REVISION HISTORY

| Revision | Date | Remarks |
|--------------------|-----------|---|
| X.S0004-520-E v2.0 | July 2007 | Incorporated changes from Miscellaneous Rev. E standards. |

PART 520

1 APPLICATION SERVICES

1.1 APPLICATION SERVICES ARCHITECTURE

The Application Services architecture introduced in MAP-500 is reproduced in the following figure.

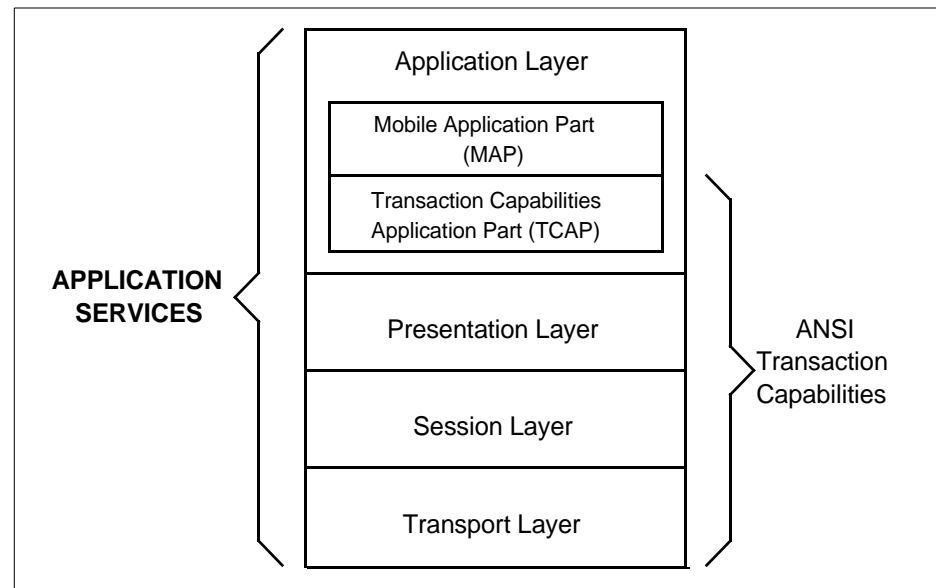


Figure 1 Application Services Architecture

Application Services are comprised of the *ANSI* Transaction Capabilities (TC) specified in *ANSI T1.114*, along with the Mobile Application Part (MAP).

The TC Transport, Session, and Presentation layers are null layers in this Standard, just as they are described in *ANSI T1.114*. They are included here for completeness; future revisions of this Standard may make use of them.

The Application Layer contains the working part of the *MAP* Application Services. It is described in the remainder of this part and in parts *MAP-540*, *-550*, *-551*.

1.2 APPLICATION LAYER STRUCTURE

The following figure illustrates the *TIA-41* Application Layer Structure.

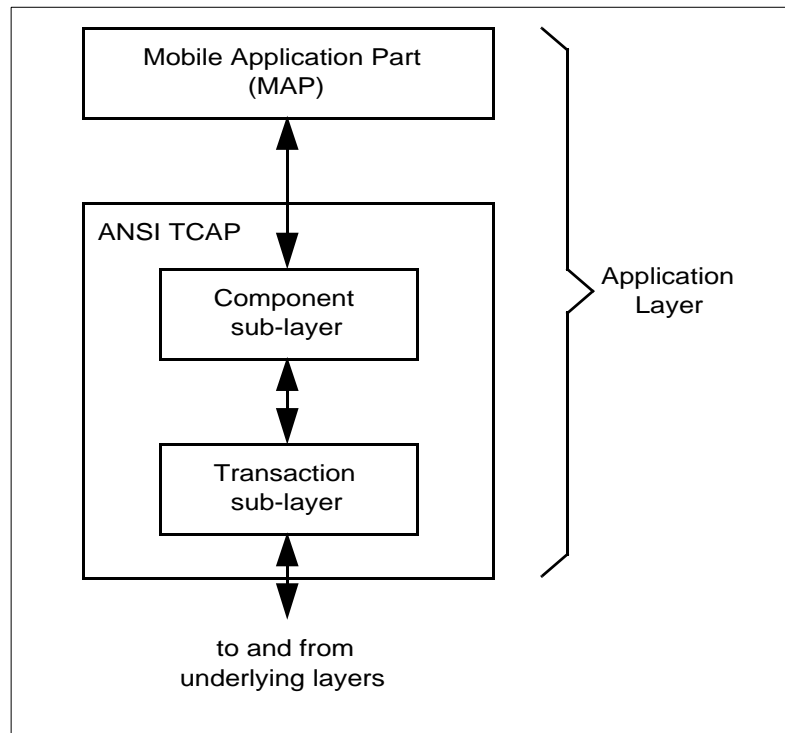


Figure 2 *TIA-41* Application Layer Structure

As illustrated, the Mobile Application Part (MAP) defined in this Standard is supported by the *ANSI* Transaction Capabilities Application Part (TCAP). TCAP, in turn, is structured in two sub-layers:

- the component sub-layer, which deals with individual actions or data, called components.
- the transaction sub-layer, which deals with the exchange of messages containing components between two MAP entities.

1.3 TCAP FORMATS AND PROCEDURES

1.3.1 Transaction Portion

This Standard employs the TCAP package formats and procedures defined in *ANSI T1.114* with the following exceptions and limitations:

- Special procedures for “Handover,” defined in the *ANSI TCAP* standard, are not supported.

1.3.2 Component Portion

This Standard employs the TCAP component formats and procedures defined in *ANSI T1.114* with the following exceptions and limitations:

- Special procedures for “Handover,” defined in the *ANSI TCAP* standard, are not supported.
- The Operation Code Identifier is coded as Private TCAP.
- The Operation Code is partitioned into an Operation Family followed by a Specifier associated with each Operation Family member. For *MAP* the Operation Family is coded as decimal 9. Bit H of the Operation Family is always coded as 0.
- A TCAP INVOKE component shall contain a Component ID Length greater than zero.
- A TCAP RETURN RESULT component shall only be transmitted in response to an INVOKE Component.
- A TCAP RETURN ERROR component shall only be sent in response to an INVOKE component, not a RETURN RESULT component.
- If a problem is detected by TCAP (i.e., the received message does not conform to *ANSI T1.114.3*), a TCAP REJECT component with one of the following Problem Specifiers shall be sent:
 - All families, General, Transaction Portion: All specifiers.
 - INVOKE: Unrecognized Correlation ID.
 - RETURN RESULT: Unrecognized Correlation ID, Unexpected Return Result.
 - RETURN ERROR: Unrecognized Correlation ID, Unexpected Return Error.
- If a problem is detected by MAP (i.e., the received message does not conform to *MAP*), a TCAP REJECT component with one of the following TCAP Problem Specifiers shall be sent:
 - INVOKE: Duplicate Invoke ID, Unrecognized Operation Code or Incorrect Parameter.
 - RETURN RESULT: Incorrect Parameter.
 - RETURN ERROR: Incorrect Parameter, Unrecognized Error, Unexpected Error.
- If an error is detected by a MAP user except SMSDeliveryBackward, SMSDeliveryForward, and SMSDeliveryPointToPoint, a TCAP RETURN ERROR component shall be sent. For SMSDeliveryBackward, SMSDeliveryForward, and SMSDeliveryPointToPoint, a TCAP RETURN RESULT with an SMS_CauseCode parameter shall be sent.

- As exceptions to the TCAP procedures related to the use of Component Types marked as “Last” if a ConnectResource INVOKE, a DisconnectResource INVOKE, a InstructionRequest INVOKE, a RemoteUserInteractionDirective INVOKE, a ResetTimer INVOKE, or an SRFDirective INVOKE is received encoded as a response to the INVOKE component of some other *MAP* operation, it should not be treated as the final response to that other INVOKE component.
- As exceptions to the TCAP procedures related to the use of Component Types marked as “Last” a SeizeResource RETURN RESULT should not be treated as the final response to a SeizeResource INVOKE component.

1.3.2.1 TCAP INVOKE Component

The structure of an *ANSI* TCAP package containing an INVOKE component is shown in the following table.

Table 1 Structure of TCAP Package with INVOKE Component

| |
|-------------------------------|
| Package Type Identifier |
| Total TCAP Message Length |
| Transaction ID Identifier |
| Transaction ID Length |
| Transaction IDs |
| Component Sequence Identifier |
| Component Sequence Length |
| Component Type Identifier |
| Component Length |
| Component ID Identifier |
| Component ID Length |
| Component IDs |
| Operation Code Identifier |
| Operation Code Length |
| Operation Code |
| Parameter Set Identifier |
| Parameter Set Length |
| Parameter Set |

1.3.2.1.1 INVOKE Response Philosophy

1. If a network entity receives an INVOKE with an incorrect TCAP message (e.g., ill-formatted), it shall respond with a REJECT to report this error.
2. If a network entity receives an INVOKE and is unable to initiate or complete the operation, it shall respond with a RETURN ERROR.

- 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
3. If a network entity receives an INVOKE and is able to complete the operation, it shall respond with a RETURN RESULT to report the completion of the operation, except for the messages HandoffMeasurementRequest and MobileOnChannel.

1.3.2.2 TCAP RETURN RESULT Component

The structure of a TCAP package containing a RETURN RESULT component is shown in the following table.

Table 2 Structure of TCAP Package with RETURN RESULT Component

| |
|-------------------------------|
| Package Type Identifier |
| Total TCAP Message Length |
| Transaction ID Identifier |
| Transaction ID Length |
| Transaction IDs |
| Component Sequence Identifier |
| Component Sequence Length |
| Component Type Identifier |
| Component Length |
| Component ID Identifier |
| Component ID Length |
| Component IDs |
| Parameter Set Identifier |
| Parameter Set Length |
| Parameter Set |

1.3.2.3 TCAP RETURN ERROR Component

The structure of a TCAP package containing a RETURN ERROR component is shown in the following table.

Table 3 Structure of TCAP Package with RETURN ERROR Component

| |
|-------------------------------|
| Package Type Identifier |
| Total TCAP Message Length |
| Transaction ID Identifier |
| Transaction ID Length |
| Transaction IDs |
| Component Sequence Identifier |
| Component Sequence Length |
| Component Type Identifier |
| Component Length |
| Component ID Identifier |
| Component ID Length |
| Component IDs |
| Error Code Identifier |
| Error Code Length |
| Error Code |
| Parameter Set Identifier |
| Parameter Set Length |
| Parameter Set |

The Parameter Set is encoded as follows:

| RETURN ERROR Parameters | | | | |
|-------------------------|-------------------|------|---------------------------|-------|
| Field | Value | Type | Reference | Notes |
| Identifier | Set [NATIONAL 18] | M | | |
| Length | variable octets | M | | |
| Contents | | | | |
| FaultyParameter | | O | 550-2.120 | a |

Notes:

- a. Include if the error involved a specific parameter.

1.3.2.3.1 Error Definitions

The detailed handling of operation errors is specified in the operations procedures.

1 The following definitions for errors are not exhaustive, but are included so that different
2 manufacturers and users have a common understanding of the mapping between a reported error and
3 the occurrence which generates it. The detection of the errors is not mandatory, however, if an error
4 is detected for one of the examples given, then the indicated error code should be used.
5 Implementations should not give cause for these reasons to be generated. For
6 SMSDeliveryBackward, SMSDeliveryForward, and SMSDeliveryPointToPoint, detected errors
7 should be mapped into the SMS_CauseCode parameter returned in the RETURN RESULT.
8

9
10 *UnrecognizedMIN*

- 11 a. Supplied MIN is not currently served by the VLR.
12
13 b. Supplied MIN is not currently served by the HLR.
14
15 c. Supplied MIN is not currently served by the Serving MSC.
16
17 d. Supplied MIN does not currently have an active call on an originating MSC.

18
19 *UnrecognizedESN*

- 20 a. Supplied ESN does not match HLR's stored value for subscriber's ESN.
21
22 b. Supplied ESN does not match VLR's stored value for subscriber's ESN.
23
24 c. Supplied ESN is negative listed.

25
26 *ID/HLRMismatch*

- 27 a. Supplied MSID is not resident on the HLR.
28
29 b. Supplied MDN is not resident on the HLR.
30
31 c. Supplied MSID is not in the AC's, MC's, or OTAF's range of MSIDs.

32
33 *OperationSequenceProblem*

- 34 a. Unexpected INVOKE in response to an INVOKE.
35
36 b. Operation is not allowed in the current state.

37
38
39 *ResourceShortage*

- 40 a. Network entity congestion.
41
42 b. Application (or function) congestion.
43
44 c. Network congestion.
45
46 d. No transaction IDs available.
47
48 e. Internal processing resource shortage (memory, I/O, disk, processor, etc.)
49
50 f. No TLDNs available.

51
52 *OperationNotSupported*

- 53 a. Operation is not supported on the addressed network entity.
54

55
56 *TrunkUnavailable*

- 57 a. Specific requested trunk is not available.
58
59 b. No trunks are available, but are required to perform the operation.
60

ParameterError

- a. Parameter errors and parameter encoding errors (550-2). Parameter errors include exactly one FaultyParameter parameter in the parameter set.

UnrecognizedParameterValue

- a. UnrecognizedParameterValue is a parameter value which is encoded properly (550-2), but its value is not recognized. UnrecognizedParameterValue errors should include exactly one FaultyParameter parameter in the parameter set.

SystemFailure

- a. System component failure.
- b. Network component failure.
- c. Chained operation failure.
- d. Required subsystem failure.

FeatureInactive

- a. Cannot reroute call because indicated feature is inactive.

MissingParameter

- a. Expected optional parameter is missing.
- b. All profile parameters are expected, but some are missing.
- c. All qualification parameters are expected, but some are missing.
- d. *MissingParameter* errors should include exactly one FaultyParameter parameter in the parameter set.

Note that this Error Code is not used to indicate a missing mandatory parameter; a REJECT component with a Problem Specifier of *Incorrect Parameter* is used in this case.

UnrecognizedIMSI/TMSI

- a. Supplied IMSI is not currently served by the VLR.
- b. Supplied IMSI is not currently served by the HLR.
- c. Supplied IMSI is not currently served by the Serving MSC.
- d. Supplied IMSI does not currently have an active call on an originating MSC.
- e. Supplied TMSI is not currently served by the Old Serving VLR. TMSI_ZONE may be matched, but TMSI_CODE is not matched.
- f. Supplied TMSI is not currently served by the Serving VLR. TMSI_ZONE may be matched, but TMSI_CODE is not matched.

TMSI/VLRMismatch

- a. Supplied TMSI is not resident on the Old Serving VLR.
- b. Supplied TMSI is not resident on the Serving VLR.

For *MAP* the Error Code Identifier is coded as Private TCAP. Error Codes are coded as follows:

Table 4 Error Codes

| Error Code Name | Error Code | | | | | | | |
|---|------------|---|---|---|---------|---|---|---|
| | H | G | F | E | D | C | B | A |
| UnrecognizedMIN | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| UnrecognizedESN | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| ID/HLRMismatch | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| OperationSequenceProblem | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| ResourceShortage | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| OperationNotSupported | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| TrunkUnavailable | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| ParameterError | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| SystemFailure | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| UnrecognizedParameterValue | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| FeatureInactive | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| MissingParameter | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| UnrecognizedIMSI/TMSI | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| TMSI/VLRMismatch | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Reserved (Note a) | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | | | | | through | | | |
| | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| Reserved for Protocol Extension (Note b) | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | | | | through | | | |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Notes:

- a. Treat a reserved value the same as value 133 (decimal), *ResourceShortage*.
- b. Error codes 224 to 255 (decimal) shall be reserved for protocol extension. If unknown, treat the same as value 133 (decimal), *Resource Shortage*.

1.3.3 TCAP REJECT Component

The structure of a TCAP package containing a REJECT component is shown in the following table.

Table 5 Structure of TCAP Package with REJECT Component

| |
|-------------------------------|
| Package Type Identifier |
| Total TCAP Message Length |
| Transaction ID Identifier |
| Transaction ID Length |
| Transaction IDs |
| Component Sequence Identifier |
| Component Sequence Length |
| Component Type Identifier |
| Component Length |
| Component ID Identifier |
| Component ID Length |
| Component IDs |
| Problem Code Identifier |
| Problem Code Length |
| Problem Code |
| Parameter Set Identifier |
| Parameter Set Length |
| Parameter Set |

A TCAP REJECT component with a TCAP Problem Code containing a Problem Specifier with one of the following values may include one occurrence of a FaultyParameter parameter identifying the parameter in error:

- a. INVOKE: *Incorrect Parameter*
- b. RETURN RESULT: *Incorrect Parameter*
- c. RETURN ERROR: *Incorrect Parameter*

When a network entity receives a message with protocol errors, it shall return with a REJECT component.

When a network entity receives a REJECT component it should:

- a. Stop timer for current task.
- b. Exit the current task.
- c. Begin recovery procedures according to the network entity's internal algorithm.
- d. If it needs to respond to another network entity due to an ongoing transaction, it should send a RETURN ERROR component indicating *System Failure*.