3GPP TSG CN Plenary Meeting #18 **NP-020670**
4th - 6th December 2002. New Orleans, USA. (revision of NP-020551 and NP-020647)

Source: **TSG CN WG 1**

Title: **CR to Rel-5 on Work Item IMS-CCR towards 24.008**

Agenda item: **8.1**

Document for: **APPROVAL**

Introduction:

This document contains **1** CR, **Rel-5** Work Item **"IMS-CCR"**, that have been agreed by **TSG CN WG1**, and are forwarded to TSG CN Plenary meeting #18 for approval.

Spec	CR #	Rev	CAT	Rel	Tdoc Title	Meeting	TDoc #	C_Version
24.008	701	3	F	Rel-5	Flow Identifier Encoding	N1-26	N1-022159	5.5.0

3GPP TSG-CN1 Meeting #26
Miami Beach, Florida, USA, 23 – 27 September 2002

Tdoc N1-022159
(revised from N1-022117)

CR-Form-v7

CHANGE REQUEST

⌘ **24.008** **CR** **701** ⌘**rev** **3** ⌘ Current version: **5.5.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Flow Identifier Encoding
Source:	⌘	Nortel Networks
Work item code:	⌘	IMS-CCR
Date:	⌘	13/09/2002
Category:	⌘	**F**
Release:	⌘	Rel-5

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	The specifications do not specify how the UE encodes the Flow Identifier in the TFT IE. 24.008 indicates the parameter contents field of the Flow Identifier contains the binary representation of a flow identifier as specified in 24.229. 24.229 refers to 29.207 for a detailed description of how the flow identifiers are constructed. 29.207 states that the flow identifier is a 2-tuple (<Media component number, IP flow number>) where both are numbered starting from 1. Since the flow identifier is 2-tuple, it is proposed that each tuple be encoded as two octets.
Summary of change:	⌘	The Flow Identifier in the TFT IE is encoded as an octect string with two octets for each entry in the 2-tuple: Media Component number and IP flow number.
Consequences if not approved:	⌘	Lack of specification on the Flow Identifier encoding in the UE.

Clauses affected:	⌘	10.5.6.12			
Other specs affected:	⌘	**Y**	**N**		
		X		Other core specifications	⌘ 29.207
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Table 10.5.162/3GPP TS 24.008: *Traffic flow template* information element

TFT operation code (octet 3)
Bits
8 7 6

0 0 0 Spare
0 0 1 Create new TFT

0 1 0 Delete existing TFT
0 1 1 Add packet filters to existing TFT

1 0 0 Replace packet filters in existing TFT

1 0 1 Delete packet filters from existing TFT
1 1 0 No TFT operation

1 1 1 Reserved

E bit (bit 5 of octet 3)

The *E bit* indicates if a *parameters list* is included in the TFT IE and it is encoded as follows:
0 *parameters list* is not included
1 *parameters list* is included

Number of packet filters (octet 3)

The *number of packet filters* contains the binary coding for the number of packet filters in the *packet filter list*. The *number of packet filters* field is encoded in bits 4 through 1 of octet 3 where bit 4 is the most significant and bit 1 is the least significant bit. For the "delete existing TFT" operation and for the "no TFT operation", the *number of packet filters* shall be coded as 0. For all other operations, the number of packet filters shall be greater than 0 and less than or equal to 8.

Packet filter list (octets 4 to z)

The *packet filter list* contains a variable number of packet filters. For the "delete existing TFT" operation, the *packet filter list* shall be empty.

For the "delete packet filters from existing TFT" operation, the *packet filter list* shall contain a variable number of packet filter identifiers. This number shall be derived from the coding of the *number of packet filters* field in octet 3.

For the "create new TFT", "add packet filters to existing TFT" and "replace packet filters in existing TFT" operations, the *packet filter list* shall contain a variable number of packet filters. This number shall be derived from the coding of the *number of packet filters* field in octet 3.

Each packet filter is of variable length and consists of

- a packet filter identifier (1 octet);
- a packet filter evaluation precedence (1 octet);
- the length of the packet filter contents (1 octet); and
- the packet filter contents itself (v octets).

The *packet filter identifier* field is used to identify each packet filter in a TFT. Since the maximum number of packet filters in a TFT is 8, only the least significant 3 bits are used. Bits 8 through 4 are spare bits.

The *packet filter evaluation precedence* field is used to specify the precedence for the packet filter among all packet filters in all TFTs associated with this PDP address. Higher the value of the *packet filter evaluation precedence* field, lower the precedence of that packet filter is. The first bit in transmission order is the most

significant bit.

The *length of the packet filter contents* field contains the binary coded representation of the length of the *packet filter contents* field of a packet filter. The first bit in transmission order is the most significant bit.

Parameters list (octets z+1 to v)

The *parameters list* contains a variable number of parameters that might need to be transferred in addition to the packet filters. If the *parameters list* is included, the *E bit* is set to 1; otherwise, the *E bit* is set to 0.

Each parameter included in the *parameters list* is of variable length and consists of:

- a parameter identifier (1 octet);
- the length of the parameter contents (1 octet); and
- the parameter contents itself (v octets).

The *parameter identifier* field is used to identify each parameter included in the *parameters list* and it contains the hexadecimal coding of the parameter identifier. Bit 8 of the *parameter identifier* field contains the most significant bit and bit 1 contains the least significant bit. In this version of the protocol, the following parameter identifiers are specified:

- 01H (Authorisation Token);
- 02H (Flow Identifier).

If the *parameters list* contains a parameter identifier that is not supported by the receiving entity the corresponding parameter shall be discarded.

The *length of parameter contents* field contains the binary coded representation of the length of the *parameter contents* field. The first bit in transmission order is the most significant bit.

When the *parameter identifier* indicates Authorization Token, the *parameter contents* field contains an authorization token, as specified in 3GPP TS 29.207. The first octet is the most significant octet of the authorization token and the last octet is the least significant octet of the authorization token.

The *parameters list* shall be coded in a way that an Authorization Token (i.e. a parameter with identifier 01H) is always followed by one or more Flow Identifiers (i.e. one or more parameters with identifier 02H).

If the *parameters list* contains two or more consecutive Authorization Tokens without any Flow Identifiers in between, the receiver shall treat this as a semantical TFT error.

When the *parameter identifier* indicates Flow Identifier, the *parameter contents* field contains the binary representation of a flow identifier~~, as specified in 3GPP TS 24.229~~. The Flow Identifier consists of four octets. Octets 1 and 2 contain the Media Component number as specified in 3GPP TS 29.207 [100]. Bit 1 of ~~the first~~ octet 2 is least significant bit, and bit 8 of ~~the last~~ octet 1 is the most significant bit. Octets 3 and 4 contain the IP flow number as specified in 3GPP TS 29.207 [100]. Bit 1 of octet 4 is least significant bit, and bit 8 of octet 3 is the most significant bit.

Table 10.5.162/3GPP TS 24.008 (continued): *Traffic flow template* information element

The *packet filter contents* field is of variable size and contains a variable number (at least one) of *packet filter components*. Each *packet filter component* shall be encoded as a sequence of a one octet *packet filter component type identifier* and a fixed length *packet filter component value* field. The *packet filter component type identifier* shall be transmitted first.

In each packet filter, there shall not be more than one occurrence of each packet filter component type. Among the "IPv4 source address type" and "IPv6 source address type" packet filter components, only one shall be present in one packet filter. Among the "single destination port type" and "destination port range type" packet filter components, only one shall be present in one packet filter. Among the "single source port type" and "source port range type" packet filter components, only one shall be present in one packet filter.

Packet filter component type identifier

Bits	
8 7 6 5 4 3 2 1	
0 0 0 1 0 0 0 0	IPv4 source address type
0 0 1 0 0 0 0 0	IPv6 source address type
0 0 1 1 0 0 0 0	Protocol identifier/Next header type
0 1 0 0 0 0 0 0	Single destination port type
0 1 0 0 0 0 0 1	Destination port range type
0 1 0 1 0 0 0 0	Single source port type
0 1 0 1 0 0 0 1	Source port range type
0 1 1 0 0 0 0 0	Security parameter index type
0 1 1 1 0 0 0 0	Type of service/Traffic class type
1 0 0 0 0 0 0 0	Flow label type

All other values are reserved.

For "IPv4 source address type", the *packet filter component value* field shall be encoded as a sequence of a four octet *IPv4 address* field and a four octet *IPv4 address mask* field. The *IPv4 address* field shall be transmitted first.

For "IPv6 source address type", the *packet filter component value* field shall be encoded as a sequence of a sixteen octet *IPv6 address* field and a sixteen octet *IPv6 address mask* field. The *IPv6 address* field shall be transmitted first.

For "Protocol identifier/Next header type", the *packet filter component value* field shall be encoded as one octet which specifies the IPv4 protocol identifier or IPv6 next header.

For "Single destination port type" and "Single source port type", the *packet filter component value* field shall be encoded as two octet which specifies a port number.

For "Destination port range type" and "Source port range type", the *packet filter component value* field shall be encoded as a sequence of a two octet *port range low limit* field and a two octet *port range high limit* field. The *port range low limit* field shall be transmitted first.

For "Security parameter index", the *packet filter component value* field shall be encoded as four octet which specifies the IPSec security parameter index.

For "Type of service/Traffic class type", the *packet filter component value* field shall be encoded as a sequence of a one octet *Type-of-Service/Traffic Class* field and a one octet *Type-of-Service/Traffic Class mask* field. The *Type-of-Service/Traffic Class* field shall be transmitted first.

For "Flow label type", the *packet filter component value* field shall be encoded as three octet which specifies the IPv6 flow label. The bits 8 through 5 of the first octet shall be spare whereas the remaining 20 bits shall contain the IPv6 flow label.