rtgwg
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

S. Hu
China Mobile
Z. Wang
Huawei
F. Qin
Z. Li
China Mobile
J. Song
Huawei
T. Chua
Singapore Telecommunications Limited
July 2, 2018

Control-Plane and User-Plane separation BNG control channel Protocol
draft-cuspdt-rtgwg-cu-separation-bng-protocol-01

Abstract

This document specifies the CU Separation BNG control channel Protocol (CUSP) for communications between a Control Plane (CP) and a set of User Planes (UPs). CUSP is designed to be flexible and extensible so as to easily allow for the addition of further messages and objects, should further requirements be expressed in the future.

Table of Contents

1. Introduction

BNG is an Ethernet-centric IP edge router, and the aggregation point for the user traffic. To provide centralized session management, flexible address allocation, high scalability for subscriber management capacity, and cost-efficient redundancy, the CU separated BNG is introduced [TR-384]. The CU separated Service Control Plane

could be virtualized and centralized, which is responsible for user access authentication and setting forwarding entries to user planes. The routing control and forwarding plane, i.e. BNG user plane (local), could be distributed across the infrastructure.

This document specifies the CU Separation BNG control channel Protocol (CUSP) for communications between a Control Plane (CP) and a set of User Planes (UPs). CUSP is designed to be flexible and extensible so as to easily allow for the addition of further messages and objects, should further requirements be expressed in the future.

2. Concept and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.1. Terminology

BNG: Broadband Network Gateway. A broadband remote access server (BRAS, B-RAS or BBRAS) routes traffic to and from broadband remote access devices such as digital subscriber line access multiplexers (DSLAM) on an Internet service provider's (ISP) network. BRAS can also be referred to as a Broadband Network Gateway (BNG).

CP: Control Plane. CP is a user control management component which supports the management of UP's resources such as the user entry and forwarding policy

UP: User Plane. UP is a network edge and user policy implementation component. The traditional router's Control Plane and Forwarding Plane are both preserved on BNG devices in the form of a user plane.

3. Protocol Overview

3.1. Initialization Phase

```
            UP                              CP
             |                               |
             |                               |
             |                               |
             |         HELLO (version)       |
             |------------------------------>|
             |                               |
             |                               |
             |                               |
             |        HELLO (version)        |
             |<----------------------------- |
             |                               |
             |                               |
             |                               |
```

The initialization phase consists of two successive steps:

1) Establishment of a TCP connection (3-way handshake) between the CP and the UP.

2) Establishment of a CUSP session over the TCP connection.

Once the TCP connection is established, the CP and the UP initiate CUSP session establishment during which the version negotiation are performed. The version's information are carried within Hello messages. If the CUSP session establishment phase fails because the CP or UP disagree on the version parameters or one of the CP or UP does not answer after the expiration of the establishment timer, the TCP connection is immediately closed.

Details about the Hello message can be found in Sections 6.2 respectively.

3.2. Network Resource Report

The CP configures the BNG's access interface via NETCONF, and UPs report attributes of according interfaces and slots.

```
          UP                              CP
           |                               |
           |        slot attributes report |
           |------------------------------>|
           |                               |
           |        port attributes report |
           |------------------------------>|
           |         Configure BNG access  |
           |<-------interface via netconf->|
           |                               |
           |                               |
           |                               |
```

Details about the Resource Report Message can be found in Sections 8 respectively.

3.3. IPoE Session Establishment

```
      UP                              CP
        |    UP report the resources    |
        |----via CUSP------------------>|
        |                               |
        |         Configur BNG access   |
        |<--------interface via netconf-|
        |                               |
        |    CP sends ACCESS_IF_INFO    |
        |<---to UPs via CUSP------------|
        |                               |
        |    User dialup via VXLAN      |
        |<----------------------------->|
        |                               |
        |    CP sends USER_BASEC_INFO   |
        |<---to UPs via CUSP------------|
        |                               |
        |    CP sends USER_IPV4_INFO    |
        |<---to UPs via CUSP------------|
        |                               |
        |    CP sends ROUTEV4 INFO      |
        |<---to UPs via CUSP------------|
        |                               |
        |    UP report the USER_DETECT_RESULT_INFO
        |----to CP via CUSP------------>|
        |                               |
        |                               |
        |    UP report the USER_TRAFFIC_INFO
        |----to CP via CUSP------------>|
        |                               |
```

Once a CUSP session has been established, if an IPoE session be required that the UPs report attributes of corresponding interfaces and slots via CUSP, and the CP initiate a NETCONF session to configure requested access interface of BNG.

Once above process has been accomplished, the CP sends the ACCESS_IF_INFO (Access Interface Information) message to UPs that contains a variety of objects that specify the set of constrains and attributes for the BNG access interface. For example, ifname = 0001, BNG service enable, IPv4 connection trigger enable, neighbor detection enable, etc.

And then the user dialup via VXLAN, the CP sends the USER_BASIC_INFOR message USER_IPV4_INFOR, and USER_ROUTEV4_INFO to UPs that contains a

variety of objects that specify the attributes for the user's basic information, user's ipv4 information, and routing information.

Upon receiving above messages from a CP, the UPs reports the user detection results and user's traffic status via USER_DETECT_RESULT_INFO message and USER_TRAFFIC_INFO, etc.

3.4. PPPoE Session Establishment

```
      UP                              CP
       |                               |
       |    UP report the resources    |
       |----via CUSP------------------>|
       |         Configur BNG access   |
       |<-------interface via netconf->|
       |                               |
       |    CP sends ACCESS_IF_INFO    |
       |<---to UPs via CUSP------------|
       |                               |
       |    User dialup via VXLAN      |
       |<----------------------------->|
       |                               |
       |    CP sends USER_BASEC_INFO   |
       |<---to UPs via CUSP------------|
       |                               |
       |    CP sends USER_IPV4_INFO    |
       |<---to UPs via CUSP------------|
       |                               |
       |    CP sends ROUTEV4 INFO      |
       |<---to UPs via CUSP------------|
       |                               |
       |    CP sends USER_PPP_INFO     |
       |<---to UPs via CUSP------------|
       |                               |
       |    UP report the USER_DETECT_RESULT_INFO
       |----to CP via CUSP------------>|
       |                               |
       |    UP report the USER_TRAFFIC_INFO
       |----to CP via CUSP------------>|
       |                               |
```

Once a CUSP session has been established, if an PPPoE session be required that the UPs report attributes of corresponding interfaces and slots via CUSP, and the CP initiate a NETCONF session to configure requested access interface of BNG.

Once above process has been accomplished, the CP sends the ACCESS_IF_INFO (Access Interface Information) message to UPs that contains a variety of objects that specify the set of constrains and attributes for the BNG access interface. For example, ifname = 0001, BNG service enable, IPv4 connection trigger enable, neighbor detection enable, etc.

And then the user dialup via VXLAN, the CP sends the USER_BASIC_INFOR message, USER_PPP_INFO message, USER_IPV4_INFOR message, and USER_ROUTEV4_INFO message to UPs that contains a variety of objects that specify the attributes for the user's basic information, user's PPP information, user's ipv4 information, and routing information.

Upon receiving above messages from a CP, the UPs reports the user detection results and user's traffic status via USER_DETECT_RESULT_INFO message and USER_TRAFFIC_INFO, etc.

3.5. Set User's QoS Information

```
       UP                              CP
        |                               |
        |    UP report the resources    |
        |----via CUSP------------------>|
        |                               |
        |   Configure BNG Access interface
        |<-----via netconf--------------|
        |                               |
        |   Configure QOS template      |
        |<-----via netconf--------------|
        |                               |
        |    User dialup via VXLAN/     |
        |<---CP sends objecitve tlv/event
        |    report,etc.                |
        |                               |
        |    CP sends USER_QOS_INFO     |
        |<---to UPs via CUSP------------|
        |                               |
```

Once a CUSP session has been established, if a user's QoS needs to be set dynamically that the UPs report attributes of according interfaces and slots via CUSP, and the CP initiate a NETCONF session to configure requested access interface of BNG and User's configuration template. And then the user dialup via VXLAN, the CP sends the USER_BASIC_INFOR message, USER_IPV4_INFOR message, and

USER_ROUTEV4_INFO message to UP, the UPs reports the user detection results and user's traffic status.

Once above process has been accomplished, the CP sends the USER_QOS_AUTH_INFO message to UPs that contains a variety of objects that specify the set of constrains and attributes for the user's required QoS.(Note that the format of these QoS attributes should synchronize with QoS configuration templates.)

3.6. CUSP session statistic

```
          UP                                  CP
           |                                    |
           |                                    |
           |<-----statistic REQUEST ------------|
           |                                    |
           |------statistic_REQUEST (ACK)------>|
           |                                    |
           |------statistic_BEGIN-------------->|
           |                                    |
           |<-----statistic_BEGIN (ACK)---------|
           |                                    |
           |------statistic_DATA--------------->|
           |                                    |
           |------statistic_END---------------->|
           |                                    |
           |<-----statistic_END (ACK)-----------|
           |                                    |
           |                                    |
```

If the CUSP session down, the CU separation BNG required that the users' information should be reserved. And if the CUSP session restart, the CP may request the UP to report the previous session's statistics to synchronize user information. Above figure describe this process, and the details about the session statistic message can be found in Sections 6.3 respectively.

4. CUSP common header

A CUSP message consists of a common header followed by a variable-length body made of a set of objects. A CUSP message with a missing mandatory object MUST trigger an Error message (see Section 5.6). Conversely, if an object is optional, the object may or may not be present.

Common header:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message-Type |F|     Resv    |       Message-Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Transaction id                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

CUSP Message Common Header

Message-Type (8 bits): The following message types are currently defined:

Value	Meaning
1	Update objective
2	Hello
3	Smooth Request
4	Smooth Begin
5	Smooth Data
6	Smooth End
7	Source Report
8	Event Report
9	Error

Flags (1 bits): The control message ACK mode be enabled by setting it to one.

Resv (7 bits): Unassigned bits are considered as reserved. They MUST be set to zero on transmission and MUST be ignored on receipt.

Message-Length (16 bits): total length of the CUSP message including the common header, expressed in bytes.

5. Objective Message Formats

CUSP objects have a common format. They begin with a CUSP common header (see Section 4). This is followed by object-specific fields defined for each different object. The object may also include one or more type-length-value (TLV) encoded data sets. Each TLV has the same structure as described in Section 5.1.

5.1. Objective TLV Format

A CUSP object may include a set of one or more optional TLVs. All CUSP objective TLVs have the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             type              |       Message-Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Value                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

    Type:   2 bytes
Length: 2 bytes
Value:  variable
```

A CUSP object TLV is comprised of 2 bytes for the type, 2 bytes specifying the TLV length, and a value field.

The first 4 bits of Type field indicate the operation of this TLV, currently, there are two types: 0 - update the objectives; 1 - delete the objectives.

The other bits of Type field indicate the TLV's type (4-15 bits), the following message types are currently defined:

Value	Meaning
0	USER_BASIC_INFO
1	USER_PPP_INFO
2	ACCESS_IFSRV_INFO
3	USER_IPV4_INFO
4	USER_IPV6_INFO
5	USER_QOS_AUTH_INFO
6	ROUTEV4_INFO
7	ROUTEV6_INFO
8	STATIC_USER_INFO

The Length field defines the length of the value portion in bytes. The TLV is padded to 4-bytes alignment; padding is not included in the Length field (so a 3-byte value would have a length of 3, but the total size of the TLV would be 8 bytes).

Unrecognized TLVs MUST be ignored.

IANA management of the CUSP Object TLV type identifier codespace is described in Section 11.

The details about the attributes of Objective TLV are specified in [Section 4.1 of draft-cuspdt-rtgwg-cu-separation-infor-model-00]

6. Control Message Format

CUSP Control TLV have a common format. They begin with a CUSP common header (see Section 3). It is followed by control TLV fields defined for each different control operations. It may also include one or more type-length-value (TLV) encoded control data sets. Each TLV has the same structure as described in Section 6.1.

For each CUSP message type, rules are defined that specify the set of objects that the message can carry. We use the Backus-Naur Form (BNF) (see [RBNF]) to specify such rules. Square brackets refer to optional sub-sequences. An implementation MUST form the CUSP messages using the object ordering specified in this document.

6.1. Control TLV Format

A CUSP control may include a set of one or more optional TLVs. All CUSP control TLVs have the following format:

```
Type:   2 bytes
Length: 2 bytes
Value:  variable
```

A CUSP control TLV is comprised of 2 bytes for the type, 2 bytes specifying the TLV length, and a value field.

Control Type (8 bits): The following message types are currently defined:

Value	Meaning
0	Hello
1	Smooth

The Length field defines the length of the value portion in bytes. The TLV is padded to 4-bytes alignment; padding is not included in

the Length field (so a 3-byte value would have a length of 3, but the total size of the TLV would be 8 bytes).

Unrecognized TLVs MUST be ignored.

IANA management of the CUSP Object TLV type identifier codespace is described in Section 11.

6.2. Hello Message

The Hello message is a CUSP message sent by a UP to a CP and by a CP to a UP in order to establish a CUSP session. The Type field of the CUSP common header for the Hello message is set to 2.

Once the TCP connection has been successfully established, the first message sent by the UP to the CP or by the CP to the UP MUST be a Hello message.

Any message received prior to a Hello message MUST trigger a protocol error condition causing an ERROR message to be sent with Error-Type Version_ Negotiation_Failed and the CUSP session establishment attempt MUST be terminated by closing the TCP connection.

The Hello message is used to establish a CUSP session between the CUSP peers. During the establishment phase, the CUSP peers exchange version information. If both parties agree on such version negotiation, the CUSP session is successfully established.

The format of a Hello message is as follows:

```
<Hello Message>::= <Common Header>
                   <HELLO_TLV>
<Hello_TLV>:: = <version>
```

Version (4 bytes) : specifies the CP/UP supported CUSP's version, currently, the version is 1.

6.3. Smooth Message

If the CUSP session down, the CU separation BNG required that the users' information should be reserved. And if the CUSP session restart, the CP may request the UP to report the previous session's statistics to synchronize user information.

The Type field of the CUSP common header for the Smooth message is set to 3/4/5/6.

The format of a Smooth message is as follows:

```
<Hello Message>::= <Common Header>
                   <Smooth_TLV>
<Smooth_TLV>::= <ClassID><Event><Resv>
```

ClassID (2 bytes): specified the statistics type of CUS session, the following statistics types are currently defined:

Value	Meaning
0	objective message statistic
1	Source report message statistic
2	Event report message statistic

Event (2 bytes): specified the Smooth message's subtypes, the following subtypes are currently defined:

Value	Meaning
0	request smooth message
1	begin smooth message
2	Smooth data message
3	End smooth message

Note that, the event value MUST be synchronized with the type of comment header.

7. Event TLV Format

CUSP Event TLV have a common format. They begin with a CUSP common header (see Section 3). It is followed by Event TLV fields defined for each different Events. It may also include one or more type-length-value (TLV) encoded Event data sets. Each TLV has the same structure as described in Section 7.1.

For each CUSP message type, rules are defined that specify the set of objects that the message can carry. We use the Backus-Naur Form (BNF) (see [RBNF]) to specify such rules. Square brackets refer to optional sub-sequences. An implementation MUST form the CUSP messages using the object ordering specified in this document.

7.1. Event TLV Format

A CUSP Event may include a set of one or more optional TLVs. All CUSP Event TLVs have the following format:

Type: 2 bytes
Length: 2 bytes
Value: variable

A CUSP Event TLV is comprised of 2 bytes for the type, 2 bytes specifying the TLV length, and a value field.

Event Type (8 bits): The following message types are currently defined:

Value	Meaning
0	USER_TRAFFIC_INFORMATION
1	USER_DETECT_RESULT_ INFORMATION

The Length field defines the length of the value portion in bytes. The TLV is padded to 4-bytes alignment; padding is not included in the Length field (so a 3-byte value would have a length of 3, but the total size of the TLV would be 8 bytes).

Unrecognized TLVs MUST be ignored.

IANA management of the CUSP Object TLV type identifier codespace is described in Section 11.

7.2. USER_TRAFFIC_INFORMATION Message

The USER_TRAFFIC_INFORMATION Message be used to reported the user's traffic statistics by UP.

The format of a USER_TRAFFIC_INFORMATION message is as follows:

```
 <USER_TRAFFIC_INFORMATION Message>::= <Common Header>
                                       <USER_TRAFFIC_INFORMATION_TLV>
 <USER_TRAFFIC_INFORMATION _TLV>::= <UserId><StatisticsType>
                                    <IngressPackets><IngressBytes>
                                    <EngressPackets>< EgressBytes >
```

USER_ID (4 bytes): is the identifier of user. This parameter is unique and mandatory. This attribute is used to distinguish different users.

StatisticsType (4 bytes): be used to indicate the Statistics type, the following types are currently defined:

Value	Meaning
0	IPv4 traffic statistic
1	IPv6 traffic statistic

IngressPackets (8 bytes): be used to present the ingress packets.

IngressBytes (8 bytes): be used to present the ingress bytes.

EgressPackets (8 bytes): be used to present the egress packets.

EgressBytes (8 bytes): be used to present the egress bytes.

7.3. USER_DETECT_RESULT_ INFORMATION Message

The USER_TRAFFIC_INFORMATION Message be used to reported the user detect fail by UP.

The format of a USER_DETECT_RESULT_ INFORMATION message is as follows:

```
< USER_DETECT_RESULT_ INFORMATION Message>::= <Common Header>
                                              < USER_DETECT_RESULT_ INFORMATION
_TLV>
< USER_DETECT_RESULT_ INFORMATION _TLV>::= <UserId><DetectFail>
```

USER_ID (4 bytes): is the identifier of user. This parameter is unique and mandatory. This attribute is used to distinguish different users.

DetectFail (2 bytes): be used to indicate that the user detect fail.

8. Resource Report TLV Format

CUSP Resource Report TLV have a common format. They begin with a CUSP common header (see Section 3). It is followed by Event TLV fields defined for each different Resources. It may also include one or more type-length-value (TLV) encoded Resource Report data sets. Each TLV has the same structure as described in Section 7.1.

8.1. Resource Report TLV Format

A CUSP Resource Report may include a set of one or more optional TLVs. All CUSP Resource Report TLVs have the following format:

Type: 2 bytes
Length: 2 bytes
Value: variable

A CUSP Resource Report TLV is comprised of 2 bytes for the type, 2 bytes specifying the TLV length, and a value field.

Resource Type (8 bits): The following message types are currently defined:

Value	Meaning
0	RESOURCE_IF_INFO
1	RESOURCE_SLOT_INFO

The Length field defines the length of the value portion in bytes. The TLV is padded to 4-bytes alignment; padding is not included in the Length field (so a 3-byte value would have a length of 3, but the total size of the TLV would be 8 bytes).

Unrecognized TLVs MUST be ignored.

IANA management of the CUSP Object TLV type identifier codespace is described in Section 11.

The details about the attributes of Resource Report TLV are specified in [Section 4.2 of draft-cuspdt-rtgwg-cu-separation-infor-model-00]

9. Error Message Format

Error messages are used by the CP or UPs to notify the other side of the connection of problems. They are mostly used by the UPs to indicate a failure of a request initiated by the CP.

The format of an Error message is as follows:

```
<Err Message> ::= <Common Header>
                  <ERRID>
```

ERRID (4 bytes): be used to indicate the error type, the following types are currently defined:

Value	Meaning
00~1000	Reserved
1001	version negotiation failed
1002	TLV type cannot be recognized
1003	TLV length Anomaly
1004	TLV objective Anomaly
1005	Smooth failed
1006	Smooth request not support

10. Security Considerations

None.

11. IANA Considerations

None.

12. Normative References

[I-D.cuspdt-rtgwg-cu-separation-bng-deployment]
Gu, R., Hu, S., and Z. Wang, "Deployment Model of Control Plane and User Plane Separation BNG", draft-cuspdt-rtgwg-cu-separation-bng-deployment-00 (work in progress), October 2017.

[I-D.cuspdt-rtgwg-cu-separation-infor-model]
Wang, Z., Gu, R., Lopezalvarez, V., and S. Hu, "Information Model of Control-Plane and User-Plane separation BNG", draft-cuspdt-rtgwg-cu-separation-infor-model-00 (work in progress), February 2018.

[I-D.cuspdt-rtgwg-cusp-requirements]
Hu, S., Gu, R., Lopezalvarez, V., Song, J., and Z. Wang, "Requirements for Control Plane and User Plane Separated BNG Protocol", draft-cuspdt-rtgwg-cusp-requirements-01 (work in progress), February 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <https://www.rfc-editor.org/info/rfc2863>.

[RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, DOI 10.17487/RFC5511, April 2009, <https://www.rfc-editor.org/info/rfc5511>.

[RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <https://www.rfc-editor.org/info/rfc5837>.

Authors' Addresses

Shujun Hu
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: hushujun@chinamobile.com

Zitao Wang
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: wangzitao@huawei.com

Fengwei Qin
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: qinfengwei@chinamobile.com

Zhenqiang Li
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing, Beijing 100053
China

Email: lizhenqiang@chinamobile.com

Jun Song
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Tee Mong Chua
Singapore Telecommunications Limited
31 Exeter Road, #05-04 Comcentre Podium Block
Singapore City 239732
Singapore

Email: teemong@singtel.com