Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 12, 2012
S. Perreault
Viagenie
T. Tsou
Huawei Technologies (USA)
April 10, 2012

Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Translation ("IGMP/MLD Translation")

draft-perreault-mboned-igmp-mld-translation-01

Abstract

This document describes translation between IGMP and MLD. This allows single-stack multicast nodes to participate in multicast groups from a different address family. Both sending and receiving is supported by this translation mechanism. Both any-source and source-specific multicast (ASM and SSM) are supported as well.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction 3
2. Terminology 3
3. Overview 3
4. Overview of Implementation Types 4
4.1. Proxy 4
4.1.1. Mixed IPv4/IPv6 Networks 5
4.2. Multicast Router 6
5. Signalling Translation 6
5.1. IP Headers 6
5.1.1. Addresses 6
5.1.2. Router-Alert Option 7
5.2. IGMP/MLD Translation 7
5.2.1. Message Type Equivalences 7
5.2.2. The "Translated" Bit 8
5.2.3. Maximum Response {Delay,Time} 12
5.2.4. Multicast Group Address 13
5.2.5. Source Addresses 14
5.2.6. Additional and Auxiliary Data 14
5.3. MTU Considerations 14
6. Data Transfer 14
7. IANA Considerations 15
8. Security Considerations 15
9. Acknowledgements 15
10. Normative References 15
Authors' Addresses 16

1. Introduction

This document specifies IGMP/MLD translation, a mechanism for IPv4-IPv6 transition and coexistence. It enables single-stack (i.e., IPv4-only or IPv6-only) hosts to participate, either as listeners, sources, or both, in multicast groups belonging to a different address family.

This translation mechanism is intended to be considered as a "virtual function" that can be implemented in a listener, in a multicast router, in an IGMP/MLD proxy [RFC4605], in existing layer-2 equipment (i.e. as a "bump in the wire"), or as a standalone translating device. Therefore, this function could be located at the customer network edge (e.g., in customer-premises equipment (CPE)) or at the provider network edge (e.g., in a DSLAM for DSL networks, in a CMTS for cable networks, etc.), or any other node reachable by IGMP/MLD packets. Note that intputs and outputs of this translation function can also be virtual. For example, translated packets need not actually be sent on the wire if the function's output is fed directly into a multicast router process (e.g., a PIM daemon) running on the same host.

2. 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].

The unqualified term "proxy" refers to an IGMP/MLD proxy as defined in [RFC4605].

3. Overview

The translation function produces IGMPv1,2,3 packets from MLDv1,2 packets and vice-versa.

```
                  +-------------+
         IGMPv1 --|             |-- MLDv1
         IGMPv2 --| Translation |-- MLDv2
         IGMPv3 --|   <----->   |
                  +-------------+
```

Virtual translation function

IPv4 addresses are mapped to IPv6 addresses and vice versa as defined in [I-D.boucadair-64-multicast-address-format]. This mapping is

stateless. This implies that arbitrary IPv6 addresses are not handled. IPv6 addresses MUST be part of a predetermined prefix in order to be translateable.

The statelessness of the translation function is considered a desirable property, and is an objective of this specification. In general, stateless network elements makes simpler designs, allows better scalability, and requires less operational overhead.

This translation function is stateless when considering full IGMP or MLD packets. Fragmented packets MUST be reassembled before they can be fed to this translation function.

4. Overview of Implementation Types

The virtual translation function defined in this document can be implemented in various ways. This section presents an overview of some possibilities.

4.1. Proxy

As described in [RFC4605], an IGMP/MLD proxy is located between an upstream network and one or more downstream networks. Listeners are in the downstream networks while the rest of the multicast infrastructure is upstream. It is possible to arrange multiple proxies in a tree topology, where one proxy's upstream network is another's downstream network.

The translation function MUST be logically situated between the proxy function and the upstream network, as shown in Figure 1.

```
                             Upstream
                                |
                           +----+-----+
                           |Translator|
                           +----+-----+
                                |
                            +---+----+
                            | Proxy  |
                            +-+-+-+-++
                              | | | |
                           Downstream(s)
```

Figure 1: IGMP/MLD translator proxy topology

There are two reasons for locating the translator upstream of the proxy rather than downstream:

1. Translating addresses on downstream interfaces could interfere with Querier elections. As described in [RFC3376] section 6.6.2 and [RFC3810] section 7.6.2, IGMP and MLD routers elect a Querier amongst themselves. The criteria for winning the election is based on the source address of IGMP/MLD Queries. Having a translator on a downstream interface would impose a new constraint on the address mapping scheme: it would need to ensure the same election result before and after applying the mapping. This constraint is lifted when the translation is applied to the upstream interface instead. Since a proxy acts as a client on its upstream interface, it does not participate in Querier elections.

2. There is only one upstream interface whereas there may be multiple downstream interfaces. Applying the translation on the upstream interface has the advantage of having a single translation point, which can facilitate debugging and troubleshooting.

Conceptually, translation is applied to messages sent upstream by the client state machine and to messages received from upstream. This document specifies how translation is carried out in terms of packet translation. However, a particular implementation is at liberty to adopt any internal structure as long as externally observable behavior is identical.

4.1.1. Mixed IPv4/IPv6 Networks

In mixed networks, where there may be both IPv4 and IPv6 receivers, two logical proxies are used. Each maintains membership state and runs state machines in the address families of the receivers it is proxying. Translation is applied to only one of them. This is illustrated in Figure 2.

```
                                  Upstream
                             +-----IPvX-----+
                             |              |
                        +----+-----+        |
                        |Translator|        |
                        +----+-----+        |
                             |              |
                       +---+--------+ +-----+------+
                       | Proxy IPvY | | Proxy IPvX |
                       +-+-+-+-+-+--+ +-+-+-+-+-+--+
                         | | | | |      | | | | |
                       Downstream(s)   Downstream(s)
                           IPvX            IPvX
```

Figure 2: Mixed IPv4/IPv6 topology

4.2. Multicast Router

When implemented as part of a multicast router, the translation function is logically situated on the downstream interfaces, as shown in Figure 3. In this example, the router speaks PIM on an upstream interface and IGMP/MLD on a downstream interface.

```
                   PIM
                    |
               +--------+
               | Router |
               +--------+
                    |
             +-------------+
             | Translation |
             +-------------+
                    |
                 IGMP/MLD
```

Figure 3: IGMP/MLD translator router topology

Note that the translation function can be completely virtual in this case, as long as the externally observable behavior conforms to this specification.

5. Signalling Translation

This section describes how to translate between IGMP and MLD.

5.1. IP Headers

5.1.1. Addresses

Destination addresses are translated as follows:

MLDv2 and IGMPv3: The destination address is a well-known address and is translated as listed in Table 1.

MLDv1, IGMPv1, and IGMPv2: The destination address corresponds to the address of the multicast group. The multicast group address is mapped between IPv4 and IPv6 as described in [I-D.boucadair-64-multicast-address-format].

Description	IPv4	IPv6
All nodes on link	224.0.0.1	ff02::1
All routers on link	224.0.0.2	ff02::2
All IGMP/MLD-capable routers on link	224.0.0.22	ff02::16

Table 1: IPv4/IPv6 Well-Known Multicast Address Equivalences

Source addresses are translated as follows:

IGMP to MLD: The source address is set to a link-local IPv6 address assigned to the proxy's upstream interface.

MLD to IGMP: The source address is set to the IPv4 address assigned to the proxy's upstream interface.

IGMP and MLD Reports having an unspecified source address (0.0.0.0 or ::) are handled differently. In MLD, they are dropped ([RFC3810] section 5.2.13). In IGMP, they are accepted ([RFC3376] section 4.2.13). This means that translating 0.0.0.0 to :: and vice-versa would change the router's behaviour: it would accept a message that should have been dropped, and vice-versa. To eliminate this ambiguity, the translator MUST drop IGMP and MLD reports having an unspecified source address.

5.1.2. Router-Alert Option

IGMP messages are sent with a Router Alert IPv4 option [RFC2113] while MLD message are sent with a Router Alert option in a Hop-By-Hop IPv6 extension header [RFC2711]. The translator needs to convert between these two. In particular, a Router Alert option MUST be sent on output if and only if it was received on input. The value MUST be set to zero unconditionally because the IPv4 and IPv6 value spaces are not identical.

5.2. IGMP/MLD Translation

5.2.1. Message Type Equivalences

The IGMP/MLD messages to be handled by the translator belong to the proxy's upstream interface, on which the proxy acts as a listener. This means that translation will be applied to IGMP/MLD Reports and Leave/Done messages sent from the proxy as well as to to IGMP/MLD Queries received from routers.

Upon reception of an IGMP message with a type field containing one of

the values listed in Table 2, the translator will intercept the message and produce an equivalent MLDv2 message corresponding to an ICMPv6 message with the listed type number in the same row. The translator will perform the reverse operation upon reception of an MLDv2 message.

IGMP type number	ICMPv6 type number
17 IGMPv1/v2/v3 Query	130 MLDv1/v2 Query
18 IGMPv1 Report	131 MLDv1 Report
22 IGMPv2 Report	131 MLDv1 Report
23 IGMPv2 Leave	132 MLDv1 Done
34 IGMPv3 Report	143 MLDv2 Report

Table 2: IGMP/MLD Message Type Equivalences

5.2.2. The "Translated" Bit

Experience IPv6 transition in general and translation in particular has shown that it is often useful, for various reasons, most often of an operational nature, that can not necessarily be anticipated at the time a specification gets written, to include a mechanism allowing the identification of translated traffic.

This specification tries to anticipate that need by assigning a reserved bit in both IGMPv3 and MLDv2 for such a purpose. When set to 1, it indicates a message that has been translated according to this specification at least once. When set to 0, it indicates that no translation has been performed.

A translator conforming to this specification MUST set the Translated bit to 1 on output. The bit is ignored on input by default, meaning that a message could be translated twice or more. This will often be undesirable, but is allowed by this specification.

In an IGMPv3 Query, bit number 64 is the Translated bit, as shown in Figure 4.

In an IGMPv3 Report, bit number 32 is the Translated bit, as shown in Figure 5.

In an MLDv2 Query, bit number 192 is the Translated bit, as shown in Figure 6.

In an MLDv2 Report, bit number 32 is the Translated bit, as shown in Figure 7.

```
     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 = 0x11  | Max Resp Code |           Checksum            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Group Address                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |T|Resv |S| QRV |     QQIC      |     Number of Sources (N)     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                       Source Address [1]                      |
    +-                                                             -+
    |                       Source Address [2]                      |
    +-                              .                              -+
    .                               .                               .
    .                               .                               .
    +-                                                             -+
    |                       Source Address [N]                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: "Translated" bit in an IGMPv3 Query (identified by the letter T)

```
     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 = 0x22  |    Reserved   |           Checksum            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |T|         Reserved            |  Number of Group Records (M)  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    .                                                               .
    .                        Group Record [1]                       .
    .                                                               .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    .                                                               .
    .                        Group Record [2]                       .
    .                                                               .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               .                               |
    .                               .                               .
    |                               .                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    .                                                               .
    .                        Group Record [M]                       .
    .                                                               .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: "Translated" bit in an IGMPv3 Report (identified by the letter T)

```
     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 = 130   |      Code     |           Checksum            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Maximum Response Code      |           Reserved            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    *                                                               *
    |                                                               |
    *                       Multicast Address                       *
    |                                                               |
    *                                                               *
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |T|Resv |S| QRV |     QQIC      |     Number of Sources (N)     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    *                                                               *
    |                                                               |
    *                       Source Address [1]                      *
    |                                                               |
    *                                                               *
    |                                                               |
    +-                                                             -+
    |                                                               |
    *                                                               *
    |                                                               |
    *                       Source Address [2]                      *
    |                                                               |
    *                                                               *
    |                                                               |
    +-                              .                              -+
    .                               .                               .
    .                               .                               .
    +-                                                             -+
    |                                                               |
    *                                                               *
    |                                                               |
    *                       Source Address [N]                      *
    |                                                               |
    *                                                               *
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: "Translated" bit in an MLDv2 Query (identified by the letter T)

```
 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 = 143   |    Reserved   |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|T|         Reserved            |Nr of Mcast Address Records (M)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                  Multicast Address Record [1]                 .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                  Multicast Address Record [2]                 .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               .                               |
.                               .                               .
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                  Multicast Address Record [M]                 .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: "Translated" bit in an MLDv2 Report (identified by the letter T)

5.2.3. Maximum Response {Delay,Time}

IGMPv2 and IGMPv3 queries contain a field specifying the Maximum Response Time (MRT), which is the maximum time allowed before sending a Report, expressed in units of 1/10 seconds. Similarly, MLDv1 and MLDv2 queries contain a field specifying the Maximum Response Delay (MRD), expressed in units of milliseconds.

IGMPv2 (resp. MLDv1) encode the MRT (resp. MRD) value directly as a binary integer. IGMPv3 (resp. MLDv2) allows a floating-point encoding as well.

IGMPv2 and IGMPv3 uses an 8-bit field while MLDv1 and MLDv2 use a 16-bit field.

The conversion algorithm is as follows:

IGMPv2 to MLDv1: MRD = 100 * MRT

All values that can be represented in IGMPv2 can be equivalently represented in MLDv1 without loss of precision.

IGMPv3 to MLDv2: MRD = 100 * MRT

All values that can be represented in IGMPv3 can be equivalently represented in MLDv2 without loss of precision.

If MRT < 128, both MRT and MRD will be encoded as binary integers.

If 128 <= MRT < 336, MRT will be encoded as a floating-point value while MRD will be encoded as a binary integer.

If 336 <= MRT, both MRT and MRD will be encoded as floating-point values.

MLDv1 to IGMPv2: MRT = min(255, round(MRD / 100))

The MRD is divided by 100, rounded to the nearest integer, then capped at 255 (the largest MRT value representable in IGMPv2). There is both precision and range loss.

MLDv2 to IGMPv3: MRT = min(31744, round(MRD / 100))

The MRD is divided by 100, rounded to the nearest integer, then capped at 31744 (the largest MRT value representable in IGMPv3). There is both precision and range loss.

If MRD < 12800, both MRD and MRT will be encoded as binary integers.

If 12800 <= MRD < 32768, MRD will be encoded as a binary integer while MRT will be encoded as a floating-point value.

If 32768 <= MRD, both MRD and MRT will be encoded as binary integers.

5.2.4. Multicast Group Address

The multicast group address is mapped between IPv4 and IPv6 as described in [I-D.boucadair-64-multicast-address-format].

5.2.5. Source Addresses

Source addresses, if any, are mapped as follows:

IGMP to MLD: IPv4 source addresses are mapped as described in [RFC6052] section 2.2.

MLD to IGMP: An IPv4 address is extracted from an IPv6 source address as described in [RFC6052] section 2.2. MLD messages containing a source address outside the IPv4-Embedded IPv6 Prefix are dropped unless there exists an applicable statically configured mapping.

5.2.6. Additional and Auxiliary Data

All IGMPv3 and MLDv2 messages can contain Additional Data, as defined in [RFC3376] sections 4.1.10 and 4.2.11 and [RFC3810] sections 5.1.12 and 5.2.11.

In addition, IGMPv3 and MLDv2 Reports can contain Auxiliary Data, as defined in [RFC3376] section 4.2.10 and [RFC3810] section 5.2.10.

A translator MUST preserve Additional and Auxiliary Data. This is accomplished by treating such data an an opaque blob and setting the appropriate IPv4 or ICMPv6 length fields appropriately.

5.3. MTU Considerations

MTU issues should be handled at the application layer, not at the IP layer. That is, the translator MUST split large Report messages into smaller pieces that fit into an interface's MTU after translation, as described in [RFC3810] section 5.2.15 and [RFC3376] section 4.2.16..

6. Data Transfer

Note: Should this section be removed? We could say nothing about the data plane and instead focus exclusively on the signalling. Thoughts?

The IGMP/MLD translator is configured either to translate the headers of multicast packets or to encapsulate/decapsulate them. This applies to regular multicast traffic, not to IGMP/MLD signalling.

Translation is performed according to [RFC6145], with the address mapping specified in [I-D.boucadair-64-multicast-address-format]. IPv6 packets with a source or destination address outside MPREFIX64 are dropped unless there exists an applicable statically configured

mapping.

Encapsulation/decapsulation might be preferable when joining two IPvX islands across an IPvY network. Interfaces on which encapsulation/decapsulation takes place are configured into the translator. When encapsulating, the original packet is not modified. If it is an IPv6 packet, it gets encapsulated in an IPv4 packet, and vice-versa. The addresses of the encapsulating packet are obtained by mapping those of the original packet according to [I-D.boucadair-64-multicast-address-format]. When decapsulating, the original packet is obtained from the encapsulating packet's payload and is forwarded as-is. Refer to [RFC2473] for details.

7. IANA Considerations

TODO

8. Security Considerations

TODO

9. Acknowledgements

The authors wish to thank the following people for their feedback: Behcet Sarikaya, Dino Farinacci, Stig Venaas, and Tom Taylor.

10. Normative References

[I-D.boucadair-64-multicast-address-format]
Boucadair, M., Qin, J., Lee, Y., Venaas, S., Li, X., and M. Xu, "IPv4-Embedded IPv6 Multicast Address Format", draft-boucadair-64-multicast-address-format-00 (work in progress), January 2012.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

[RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option",

RFC 2711, October 1999.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

Authors' Addresses

Simon Perreault
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada

Phone: +1 418 656 9254
Email: simon.perreault@viagenie.ca
URI: http://viagenie.ca

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1 408 330 4424
Email: tina.tsou.zouting@huawei.com