SFC WG
Internet-Draft
Intended status: Standards Track
Expires: July 9, 2016

C. Wang
W. Meng
ZTE Corporation

January 6, 2016

IPv6 Service function Chain
draft-wang-6man-ipv6-service-function-chain-01

Abstract

Service function chain is the definition of an ordered set of service functions. After instantiated, the service function path is created and the classified traffic is steered through the corresponding service function path and then forwarded to the final destination. This document tries to describe how to use IPv6 data plane and IPv6 extension headers to realize service function chain in IPv6 network.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 9, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction

Service function chain is the definition of an ordered set of service functions. After instantiated, the service function path is created and the classified traffic is steered through the corresponding service function path and then forwarded to the final destination.

This document tries to describe how to use IPv6 data plane and IPv6 extension headers to realize service function chain in IPv6 network. Specifically, this document tries to provide:

In section 3, an introduction on service function chain.

In section 4, an analysis on service function chain over IPv6 network

In section 5, service function path information header over IPv6 data plane

In section 6, context headers over IPv6 data plane

In section 7, the procedures of IPv6 service function chain

In section 8, simple offload over IPv6 network

In section 9, security for IPv6 service function chain

2. Convention 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].

The terms about SFC are all defined in [RFC7665].

The terms about IPv6 are all defined in [RFC2460].

3. An introduction on service function chain

3.1. service function chain components

Service function chain, defined in [RFC7665], defines several requisite components to implement SFC, including classifier, which performs classification for incoming packets, and Service Function Forwarder/SFF, which is responsible for forwarding traffic to one or more connected service functions according to the information carried in the SFC encapsulation, as well as handling traffic coming back from the SF and transporting traffic to another SFF and terminating the SFP. And what's more, another significant component is Service Function/SF, which is responsible for specific treatment of received packets.

Based on these SFC components, the service function paths are created. Architecturally, within the same SFC-enabled domain, some SFPs may be fully specified, defining the exactly SFFs and SFs visited by classified packets, which are also named as Render Service Paths or RSPs. While other SFPs may be relatively vague, some SFFs or SFs are not designated at the classifier, instead, SFFs can decide which attached SFs to visit when packets arrive.

3.2. data plane in service function chain

In SFC data plane, there also defines another important concept named SFC encapsulation. The SFC encapsulation includes service function path information which determines how the packets are forwarded in the SFC domain, and metadata/context data information when such metadata is required. In [I-D.ietf-sfc-nsh], the SFC encapsulation is defined as Network Service Header(NSH).

In Figure 1, there is a Service Function Chain described as SFC1: Firewall(SF2) -- > DPI(SF4) -- > IPS(SF6). After instantiated, SFC1 is specified as a SFP:
SFF1-->Firewall(SF2)-->SFF3-->DPI(SF4)-->SFF5-->IPS(SF6),which is identified by a SFPID.

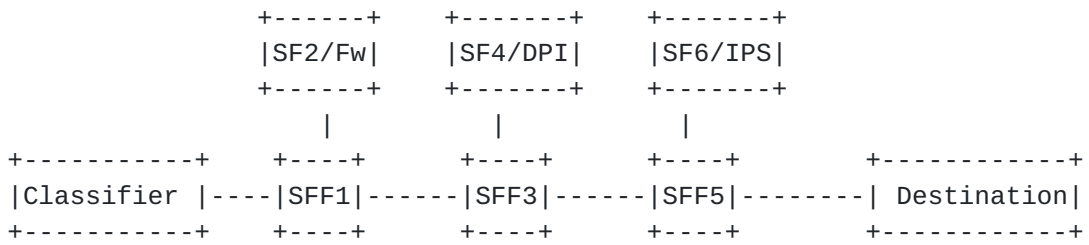

Figure 1: SFC Example

In Figure 2, it illustrates the NSH format which is defined in [I-D.ietf-sfc-nsh].

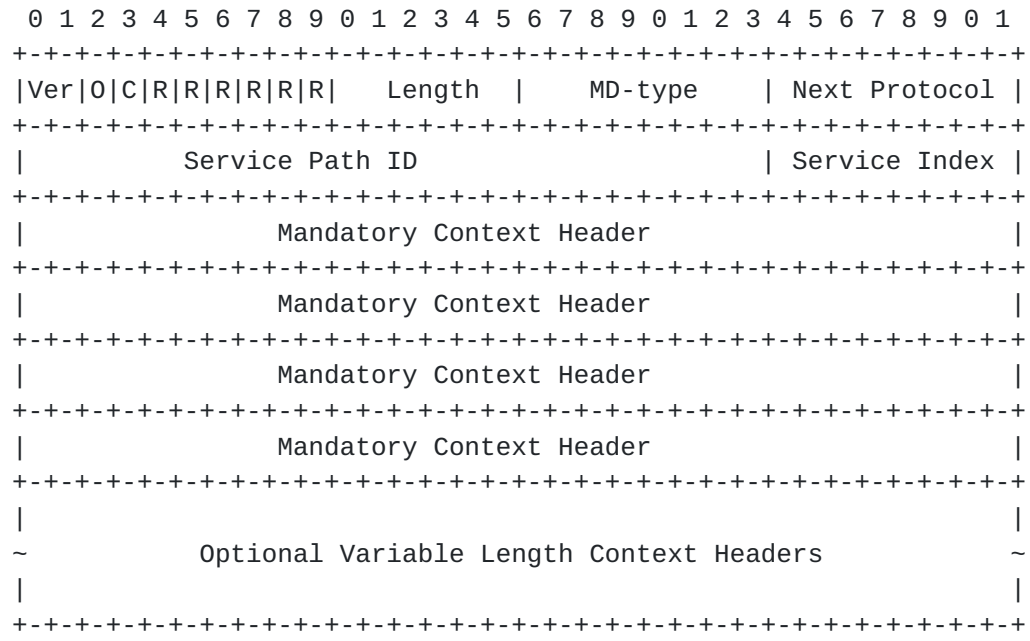

Figure 2: Network Service Header format

When packets arrived at classifier, classification is triggered. Packets that satisfy classification rules are forwarded according to a specific SFP. And also, there may be some metadata/context data resulted from the classification. Then, the classifier encapsulates the SFP information and metadata information in the NSH, then encapsulates the NSH in the original packets, after that, chooses the appropriate overlay technology as transport layer to encapsulate the NSH packets, and forwards these encapsulated packets through overlay network to the next SFF.

Packets arrive at an SFF from the overlay network. The SFF determines the appropriate SF the packets should be forwarded to via SFP information carried in NSH. After SF, the packets are returned to the SFF. According to the SFP information, if the next hop is another SF associated with that SFF, packets then are forwarded to another SF. If the next hop is not local SFs, packets then are encapsulated with appropriate overlay technology and forwarded to the next SFF along the path. If there is no next hop and the last SF has been serviced, the SFF then removes the NSH and delivers the original packets to the network.

Sometimes, re-classification may occur on the SFF. After re-classification, the service function path and/or the metadata information may change; new NSH then is encapsulated in the packets instead of original NSH.

In some cases, there may be some SFC-unaware SFs attached to the SFF, and then the SFF acts as a SFC Proxy to remove the NSH and forward the packets to the SFC-unaware SF. After receiving the served packets from the SFC-unaware SF, SFF then encapsulates the NSH again.

Packets arrive at an SFC-aware SF from the attached SFF with NSH information. The SF acquires the metadata if there is metadata information in the NSH, and then serves the packets. If there is any requisite sharing metadata resulted from this SF, then this SF updates the metadata information in the NSH and then forwards the packets to the attached SFF.

4. An analysis on service function chain over IPv6 network

To achieve Service Function Chain, all the visited SFFs and SFs need recognize NSH. Based on this new technology, there derives several new companion technologies to achieve integrated SFC, such as SFC Proxy, SFC OAM, SFC Loop Detection and avoidance, etc.

In fact, in IPv6 network, there has an alternative method to realize service function chain, which may be easier to deploy service function chain function rapidly than SFC technology.

In the following sections, how to rapidly realize service fucntion chain in IPv6 is proposed. Specifically, the mechanism tries to encapsulate service function chain information in IPv6 extension headers of IPv6 packets, and then send the IPv6 packet along the path according to the service function chain informaiton.

4.1. Routing header in IPv6 network

Section 4.4 in [RFC2460] defines routing headers in IPv6 network.

The routing header is used by an IPv6 source to list one or more intermediate nodes to be visited on the way to a packet's destination. And the routing header is not examined or processed until it reaches the node identified in the Destination Address field of the IPv6 header. The routing header is identified by a Next Header value of 43 in the immediately preceding header, and has the following format in Figure 3:

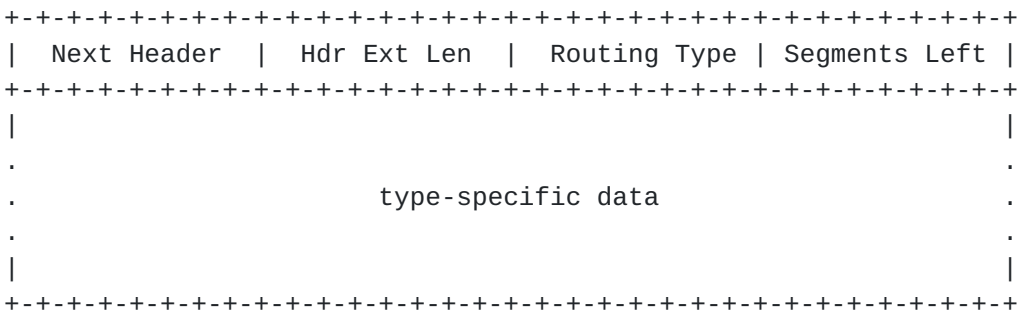

Figure 3: Routing header in IPv6 network

Analyzing the mechanism of the routing header, it is a little similar to some features of service function chain. For example, in IPv6 network, the source node encapsulates routing header according to the packets' destination, and the information in the routing header is the forwarding path information. Similarly, in SFC domain, the

classifier encapsulates NSH according to the packets' 3-tuple or 5-tuple or other information in the packets, and the information in the NSH includes the forwarding path information. Certainly, there are some differentiated feathers in SFC, such as re-classification, simple offload, bypass and so on. There still has corresponding measures to work them out in IPv6 network.

4.2. Destination options headers in IPv6 network

Section 4.6 in [RFC2460] defines destination option headers in IPv6 network.

The destination option header is used to carry optional information that need be examined only by a packet's destination node(s), and has the following format in Figure 4:

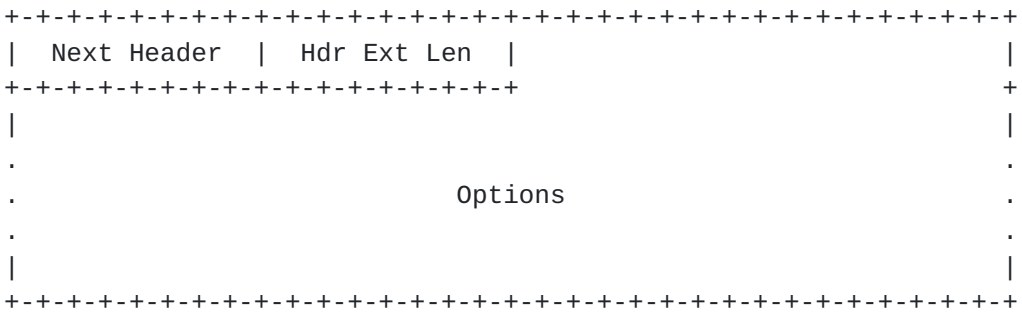

Figure 4: Destination options headers in IPv6 network

In SFC domain, metadata/context data is the context information shared between classifiers and SFs and among SFs, which means metadata only need be examined by classifiers and SFs. As for classifier, it is the source of SFC packets. As for SFs, they are intermediate destination nodes for SFC packets.

Note that there are two possible ways to encode optional destination information in an IPv6 packet: either as an option in the Destination Option header, or as a separate extension header.

5. Service function path information header over IPv6 data plane

To carry service function paths information in IPv6 network, this document tries to extend IPv6 routing header to carry them. A new type of routing header is defined: the Service Function Chain routing type. And it is illustrated as follow in Figure 5:

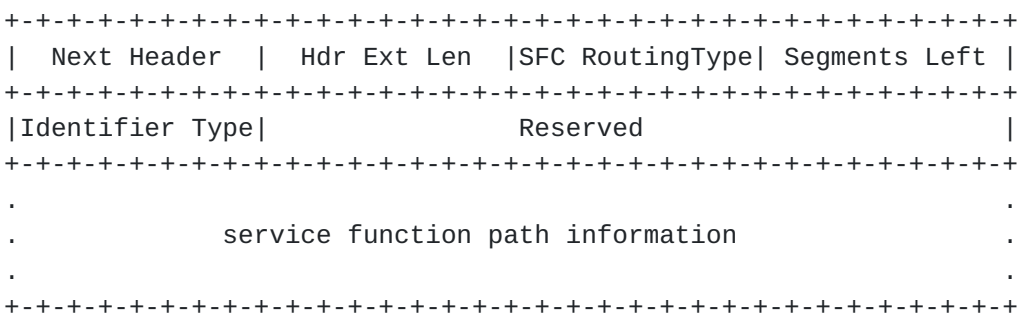

Figure 5: service funciton path information header over IPv6 data plane

Next Header: identifies the type of header immediately following the SFC routing header.

Hdr Ext Len: identifies the length of the SFC routing header in 8-octes units.

SFC Routing Type: identifies the service function chain information is carried in the following data field, and need to be assigned by IANA.

Segments Left: similar to service index in NSH. Segments Left is decremented at each destination node and it is used as a service index to locate the next destination node along the service function path.

Identifier Type: identifies how to describe the service function path information. It may be a service function path identifier which identifies the service function path uniquely, or it may be a list of identifiers which identify the SFFs and/or SFs in the service function path, such as a list of addresses.

Note that, except IPv6 routing header, carrying the service function paths information in a new IPv6 extension header can work out as well.

6. Metadata/Context headers over IPv6 data plane

To carry metadata/context headers information in IPv6 network, this document tries to extend IPv6 destination option header to carry them. A new option of destination option header is defined: Metadata Option. And it is illustrated as follow in Figure 6:

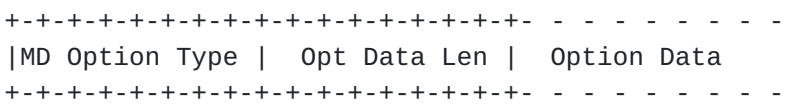

Figure 6: metadata/context headers over IPv6 network

Option data: identifies the metadata/context header information which needs to be shared between classifiers and SFs and among SFs.

In order to distinguish different metadata type, here tries to define several metadata sub-options. The format is as follow in Figure 7. For example, in Figure 8, there are network platform context information sub-option, network shared context information sub-option, service platform context information sub-option, service shared context information sub-option and some other optional variable length Context information sub-options.

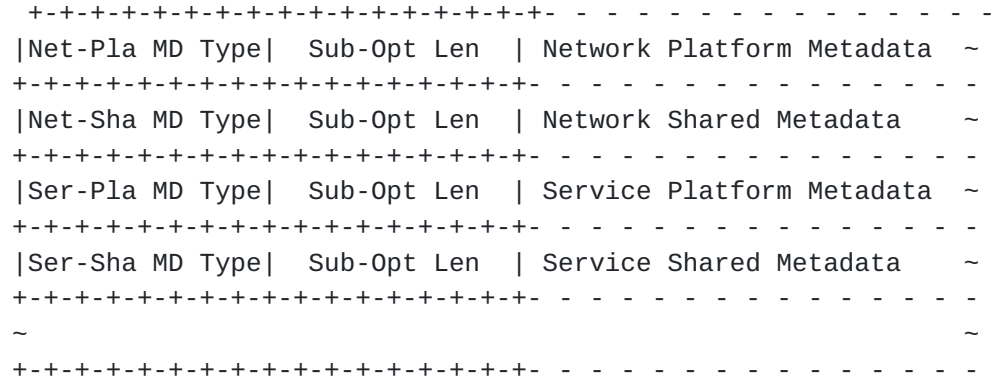

Figure 7: sub-options for differentiated metadata/context headers

Note that, except IPv6 destination option header, carrying the metadata/context headers in a new IPv6 extension header can work out as well.

7. The procedures of IPv6 service function chain

This section tries to explain how to work out SFC in IPv6 network through IPv6 extension headers.

7.1. Example 1 (Identifier type= a list of addresses which identify SFFs and/or SFs)

Here consider the case of a source node S sending a packet to destination node D, after classification on the source node S, it turns out that the packet need to be served by SFC1: SF2 --> SF4. After instantiated, the exactly service function path is: SFF1-->SF2-->SFF3-->SF4. Every SFFs and SFs are identified by an IPv6 address respectively. So the packet is routed via intermediate nodes SFF1, SF2, SFF3, SF4 when traveling from the source node S to the destination node D. Here, tries to put SFF1/SF2/SFF3/SF4/D's IPv6 addresses in a sequence list, and encapsulate this list of address in the extended IPv6 routing header defined in Section 5 in this document, which is illustrated in Figure 8. What's more, the source node S may acquire some metadata/context headers information, which need to be encapsulated in the extended IPv6 routing header defined in Section 6 in this document.

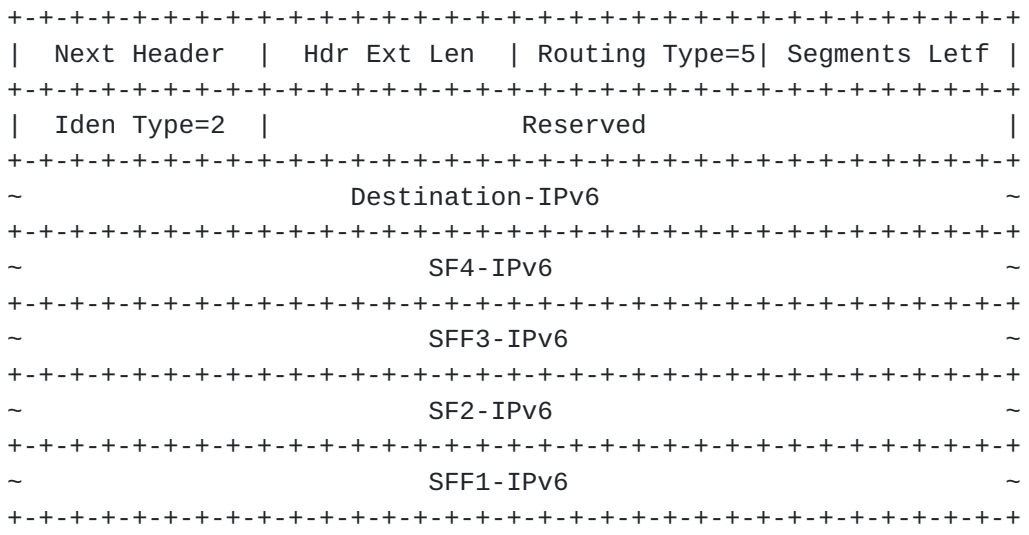

Figure 8: a list of addresses to identify Service funciton path information

After that, mostly, the forwarding procedures are similar to the procedures defined in section 4.4 and section 4.6 in RFC2460. Such as, SFFs and SFs as intermediate destination nodes need to process the metadata/context headers information in the destination option

headers, if needed, update the metadata/context headers information to be shared by the following nodes. And also, SFFs and SFs as intermediate destination nodes need to analyze the service function path information and extract the next destination to update the IPv6 destination address in IPv6 header.

In some cases, the SFs' IPv6 address may be local IPv6 address, and then the SFFs need to recognize them and forward them correctly.

In some other cases, the source node cannot get all the exactly SFFs/ SFs' IPv6 addresses, then the intermediate nodes may need to update the IPv6 addresses list in the service function path information.

Also, re-classification may be occurred in some intermediate nodes to re-steer the packets with updated list of addressed in service function path information and with updated metadata/context headers.

What's more, sometimes, according to the IPv6 destination address, packets are forwarded to the next destination nodes may be through different transport layer protocols, such as native IPv6 transport layer protocol or MPLS transport layer protocol, or GRE transport layer protocol or other transport layer protocols.

7.2. Example 2 (Identifier type = service function path identifier)

Here still consider the case of a source node S sending a packet to destination node D, after classification on the source node S, it turns out that the packet need to be served by SFC1: SF2 --> SF4. After instantiated, the exactly service function path is: SFF1-->SF2-->SFF3-->SF4, which is correspond to a service function path identifier: SFPID1. Here tries to encapsulate a SFPID to describe the whole service function path information in the IPv6 extended header defined in section 5 in this document, which is relatively short and simple to carry and is illustrated in Figure 9. After acquiring the SFPID information, then acquire the corresponding metadata/context header, and encapsulate them in the extended IPv6 routing header defined in Section 6 in this document.

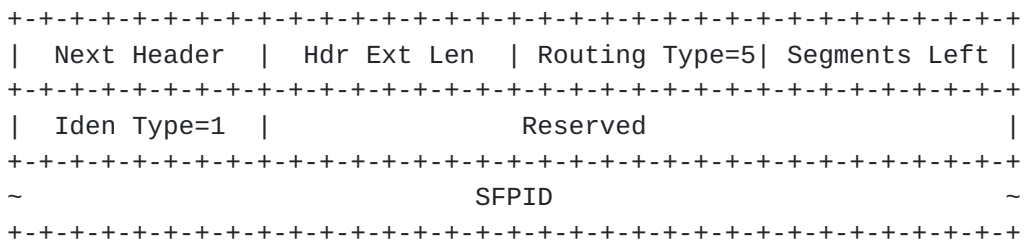

Figure 9: Network Service Header format

The source node S analyzes the SFPID and Segments Left in the routing header, and extracts the next SFF IPv6 address to update the IPv6 destination address in IPv6 header, then forwards the packet to the destination through transport layer protocols, including native IPv6 transport layer protocol or MPLS transport layer protocol, or GRE transport layer protocol or other transport layer protocols.

When SFFs and/or SFs receive the packets, analyze the service function path information and metadata/context headers information, and take advantage of these information to determine the next destination node. To help extract the next destination node, here may be need a service function path forwarding table corresponding to each SFPID to identifier where is the next station.

Sometimes, re-classification may be occurred in some intermediate nodes to re-steer the packets with updated SFPID in service function path information and with updated metadata/context headers.

8. Service function chain simple offload over IPv6 network

TBD

9. Security Considerations

TBD

10. IANA Considerations

TBD

11. References

11.1. Normative References

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <http://www.rfc-editor.org/info/rfc7665>.

11.2. Informative References

[I-D.ietf-sfc-nsh] Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-01 (work in progress), July 2015.

Authors' Addresses

Cui(Linda) Wang
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: wang.cui1@zte.com.cn

Wei Meng
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: meng.wei2@zte.com.cn,vally.meng@gmail.com