Workgroup: MPLS Working Group
Internet-Draft: draft-ietf-mpls-1stnibble-04
Updates: 4928 (if approved)
Published: 29 February 2024
Intended Status: Standards Track
Expires: 1 September 2024
Authors: K. Kompella, Juniper Networks
S. Bryant, University of Surrey 5GIC
M. Bocci, Nokia
G. Mirsky, Ed., Ericsson
L. Andersson, Huawei Technologies
J. Dong, Huawei Technologies

IANA Registry for the First Nibble Following a Label Stack

Abstract

The goal of this memo is to create a new IANA registry (called the Post-stack First Nibble registry) for the first nibble (4-bit field) immediately following an MPLS label stack. The memo offers a rationale for such a registry, describes how the registry should be managed, and provides some initial entries. Furthermore, this memo sets out some documentation requirements for registering new values. Finally, it provides some recommendations that make processing MPLS packets easier and more robust.

The relationship between the IANA IP Version Numbers (RFC 2780) and the Post-stack First Nibble registry is described in this document.

This memo, if published, would update RFC 4928.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

An MPLS packet consists of a label stack, an optional "post-stack header" (PSH) and an optional embedded packet (in that order). By PSH, we mean existing artifacts such as Control Words, BIER headers and the like, as well as new types of PSH being discussed by the MPLS Working Group. However, in the data plane, there are scant clues regarding the PSH, and no clue as to the type of embedded packet; this information is communicated via other means, such as the routing protocols that signal the labels in the stack. Nonetheless, in order to better handle an MPLS packet in the data plane, it is common practice for network equipment to "guess" the type of embedded packet. Such equipment may also need to process the

PSH. Both of these require parsing the data after the label stack. To do this, the "first nibble" (the top four bits of the first octet following the label stack) is often used. Although some existing network devices may use such a method, it needs to be stressed that the correct interpretation of the Post-stack First Nibble (PFN) in a PSH can be made only in the context of the Label Stack Element (LSE) or group of LSEs in the preceding label stack that characterize the type of the PSH, and that any attempt to rely on the value in any other context is unreliable.

The semantics and usage of the first nibble are not well documented, nor are the assignments of values. This memo serves four purposes:

* To document the assignments already made.

* To provide straightforward documentation of future assignments through the creation of a "Post-stack First Nibble registry".

* Provide a method for tracking usage by requiring more consistent documentation.

* To stress the importance that any MPLS packet not carrying plain IPv4 or IPv6 packets MUST contain a PSH.

This memo introduces a requirement and a recommendation, the first building on the Section 2.1.1; the second deprecating the use of the heuristic in Section 2.1.1.1. The intent of both of these is that legacy routers continue to operate as they have, with no new problems introduced as a result of this memo. However, new implementations SHOULD follow these recommendations for a more robust operation.

1.1. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

MPLS packet: one whose Layer 2 header declares the type to be MPLS. For Ethernet, that means the Ethertype is 0x8847 or 0x8848.

Label Stack: (of an MPLS packet) all labels (four-octet fields) after the Layer 2 header, up to and including the label with the BoS bit set ([RFC3032]).

Post-stack First Nibble (PFN): the most significant four bits of the first octet following the label stack.

MPLS Payload:
all data after the label stack, including the PFN, an optional post-stack header, and the embedded packet.

Post-stack Header (PSH): optional field of interest to the egress LSR (and possibly to transit LSRs). Examples include a control word or an associated channel. The PSH MUST indicate its length, so that a parser knows where the embedded packet starts.

Embedded Packet: all octets beyond the PSH (if any). That could be an IPv4 or IPv6 packet , an Ethernet packet (for VPLS ([RFC4761], [RFC4762]) or EVPN [RFC7432]), or some other type of Layer 2 frame [RFC4446].

Deprecation: regardless of how the deprecation is understood in other IETF documents, the interpretation in this document is that if a practice has been deprecated, that practice should not be included in the new implementations or deployed in deployments.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
 X |                         Layer 2 Header                        | |
    |                                                               | |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/
                                                TC   S       TTL
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
 Y |              Label-1                  | TC  |0|      TTL      | |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Label-2                  | TC  |0|      TTL      | |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                ...                    | TC  |0|      TTL      | |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Label-n                  | TC  |1|      TTL      | |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/
```

Figure 1: Example of an MPLS Packet With Label Stack

```
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
A | (PFN) |                   IP packet                            | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                             data                               | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 ...                                 |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        end of IP packet                        | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/


  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
B | (PFN) |                 non-IP packet                          | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                             data                               | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 ...                                 |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                      end of non-IP packet                      | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/


  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+\
C | (PFN) |                      PSH                               | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                              PSH                               | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                 ...                                 |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                           end of PSH                           | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                        embedded packet                         | |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+/
```

Figure 2

Figure 1 shows an MPLS packet with Layer 2 header X and a label stack Y ending with Label-n. Then, there are three examples of an MPLS payload. The complete MPLS packet thus would consist of [X Y A], or [X Y B], or [X Y C].

A. The first payload is a bare IP packet, i.e., no PSH. The PFN in this case overlaps with the IP version number.

B. The next payload is a bare non-IP packet; again, no PSH. The PFN here is the first nibble of the payload, whatever it happens to be.

C. The last example is an MPLS Payload that starts with a PSH followed by the embedded packet. Here, the embedded packet could be IP or non-IP.

1.2. Abbreviations

LSR: Label Switching Router

LSE: Label Stack Element

PSH: Post-Stack Header

PFN: Post-stack First Nibble

2. Rationale

2.1. Why Look at the First Nibble

An MPLS packet can contain many types of embedded packets. The most common types are:

1. An IPv4 packet (whose IP header has version number 4).

2. An IPv6 packet (whose IP header has version number 6).

3. A Layer 2 Ethernet frame (i.e., not including the Preamble or the Start frame delimiter), starting with the destination MAC address.

Many other packet types are possible, and in principle, any Layer 2 embedded packet is permissible; indeed, in the past, PPP, Frame Relay and ATM packets were reasonably common.

In addition, there may be a PSH ahead of the embedded packet, and it needs to be parsed. The PFN is currently used for both of these purposes.

2.1.1. Load Balancing

There are four common ways to load balance an MPLS packet:

1. One can use the top label alone.

2. One can do better by using all of the (non-SPL) labels in the stack.

3. One can do even better by "divining" the type of embedded packet, and using fields from the guessed header. The ramifications of using this load-balancing technique are discussed in detail in Section 2.2.

4. One can do best by using either an Entropy Label [RFC6790] or a FAT Pseudowire Label [RFC6391]; see Section 2.2.)

Load balancing based on just the top label means that all packets with that top label will go the same way -- this is far from ideal. Load balancing based on the entire label stack (not including SPLs) is better, but it may still be uneven. If, however, the embedded packet is an IP packet, then the combination of (<source IP address>, <dest IP address>, <transport protocol>, <source port>, and <dest port>) from the IP header of the embedded packet forms an excellent basis for load-balancing. This is what is typically used for load balancing IP packets.

An MPLS packet doesn't, however, carry a payload type identifier. There is a simple (but dangerous) heuristic that is commonly used to guess the type of the embedded packet. The first nibble, i.e., the four most significant bits of the first octet, of an IP header contains the IP version number. That, in turn, indicates where to find the relevant fields for load-balancing. The heuristic goes roughly as follows:

2.1.1.1. Heuristic for Load Balancing

1. If the PFN is 0x4 (0100b), treat the payload as an IPv4 packet, and find the relevant fields for load-balancing on that basis.

2. If the PFN is 0x6 (0110b), treat the payload as an IPv6 packet, and find the relevant fields for load-balancing on that basis.

3. If the PFN is anything else, the MPLS payload is not an IP packet; fall back to load-balancing using the label stack.

This heuristic has been implemented in many (legacy) routers, and performs well in the case of Figure 2, A. However, this heuristic can work very badly for Figure 2, B. For example, if payload B is an Ethernet frame, then the PFN is the first nibble of the OUI of the destination MAC address, which can be 0x4 or 0x6, and if so would lead to very bad load-balancing. This behavior can happen to other types of non-IP payloads as well.

That, in turn, led to the idea of inserting a PSH (e.g., a pseudowire control word [RFC4385], a DetNet control word [RFC8964] or a BIER header [RFC8296]) where the PFN is not 0x4 or 0x6, to explicitly prevent forwarding engines from confusing the MPLS payload with an IP packet. [RFC8469] recommends the use of a control word when the embedded packet is an Ethernet frame. RFC 8469 was published at the request of the operator community and the IEEE RAC as a result of operational difficulties with pseudowires that did not contain the control word.

It is RECOMMENDED that where load-balancing of MPLS packets is desired, the load-balancing mechanism uses the value of a dedicated label, for example, either an Entropy Label [RFC6790] or a FAT

Pseudowire Label [RFC6391]. Furthermore, the heuristic of guessing the type of the embedded packet, as discussed above, SHOULD NOT be used.

A consequence of the latter approach is that, while legacy routers may look for an PFN of 0x4 [RFC0791] or 0x6 [RFC8200], no router will look for a PFN of 0x7 (or whatever the next IP version number will be) for load-balancing purposes. This means that the values 0x4 and 0x6 are used to (sometimes incorrectly) identify IPv4 and IPv6 packets, but no other First Nibble values will be used to identify IP packets.

This also expands the PFN Registry to all 16 possible values, not just 0x0 and 0x1.

2.2. Updates of RFC 4928

Paragraph 3 in Section 3 of RFC 4928 [RFC4928] states that:

It is REQUIRED, however, that applications depend upon in-order packet delivery restrict the first nibble values to 0x0 and 0x1. This will ensure that their traffic flows will not be affected if some future routing equipment does similar snooping on some future version(s) of IP.

The text in RFC 4928 [RFC4928] concerning the first nibble after the MPLS Label Stack has been updated by [I-D.ietf-mpls-1stnibble] and the heuristic for snooping this nibble has been deprecated. RFC 4928 is now updated as follows:

Network equipment that complies with [I-D.ietf-mpls-1stnibble] MUST use a PSH (Post-Stack Header) with a PFN (Post-stack First Nibble) value that is neither 0x4 nor 0x6 in all cases when the MPLS payload is not an IP packet.

The recommendation (see Section 2.1.1.1) replaces the paragraph 4 in Section 3 of RFC 4928 [RFC4928] as follows:

OLD TEXT:

This behavior implies that if in the future an IP version is defined with a version number of 0x0 or 0x1, then equipment complying with this BCP would be unable to look past one or more MPLS headers, and load-split traffic from a single LSP across multiple paths based on a hash of specific fields in the IPv0 or IPv1 headers. That is, IP traffic employing these version numbers would be safe from disturbances caused by inappropriate load-splitting, but would also not be able to get the performance benefits.

NEW TEXT:

[I-D.ietf-mpls-1stnibble] deprecated the practice of deducing the payload type to avoid inaccurate load balancing based on the PFN value. This means that older implementations and deployments can continue to use that heuristic, while it must not be part of new implementations or deployments. The deprecation also means that concerns about load balancing for future IP versions with a version number of 0x0 or 0x1 are now moot.

A new document is to be published to obsolete MPLS encapsulations without PSH of non-IP payload when sufficient evidence exists that there are no marketed or deployed implementations using the heuristic practice.

END

Furthermore, the following text is appended to Section 1.1 of RFC 4928 [RFC4928]:

PSH: Post-Stack Header

PFN: Post-stack First Nibble

2.3. Why Create a Registry

The MPLS WG is currently engaged in updating the MPLS architecture; part of this work may involve the use of PSHs. That might be more challenging if PSH values are allocated on an ad hoc basis, and their parsing and semantics are ill-specified. Consider that the PFN value of 0x0 has two different formats, depending on whether the PSH is a pseudowire control word or a DetNet control word; disambiguation requires the context of the service label. This was a considered decision; documenting this would be helpful to future implementors.

With a registry, PSHs become easier to parse; not needing means outside the data plane to interpret them correctly; and their semantics and usage are documented. (Thank you, IANA!)

2.4. The Relationship between IANA IP Version Numbers [RFC2780] and Post-stack First Nibble Registries

The use of the PFN stemmed from the desire to heuristically identify IP packets for load-balancing purposes. It was then discovered that non-IP packets, misidentified as IP when the heuristic failed, were being badly load balanced, leading to [RFC4928]. This situation may confuse some as to the relationship between the Post-stack First

Nibble Registry and the IP Version Numbers registry. These registries are quite different:

1. The IP Version Numbers registry's explicit purpose is to track IP version numbers in an IP header.

2. The Post-stack First Nibble registry's purpose is to track PSH types.

The only intersection points between the two registries is for values 0x4 and 0x6 (for backward compatibility). There is no need to track future IP version number allocations in the Post-stack First Nibble registry.

3. IANA Considerations

3.1. The Post-stack First Nibble Registry

This memo recommends the creation of an IANA registry called "The Post-stack First Nibble Registry" with the following values:

Code points allocated for BIER		
Value	PSH Type	Reference
0x5	BIER header - Normal traffic	RFC 8296

Code points allocated for DETNET		
Value	PSH Type	Reference
0x0	DetNet Control Word	RFC 8964
0x1	DetNet Associated Channel	draft-ietf-detnet-mpls-oam

Code points allocated for Network Service Header (NSH)		
Value	PSH Type	Reference
0x0	NSH Base Header, Payload	RFC 8300
0x2	NSH Base Header, OAM	RFC 8300

Code points allocated for Pseudowires (PW)		
Value	PSH Type	Reference
0x0	PW Control Word	RFC 4385
0x1	PW Associated Channel	RFC 4385

Code points allocated for the MPLS Generic Associated Channel		
Value	PSH Type	Reference
0x1	MPLS G-ACh	RFC 5586

Reserved Code Points, not to be allocated		
Value	Usage	Reference
0x4	IPv4 Protocol Number	RFC 791
0x6	IPv6 Protocol Number	RFC 8200

Unassigned Code Points		
Value	PSH Type	Reference

0x3	-	-
0x7-0xF	-	-

Figure 3: The Post-stack First Nibble Values

3.1.1. Allocation Policy

All new values registered here MUST use the Standards Action policy [RFC8126].

4. Security Considerations

This document proposes a new IANA registry and does not raise any security concerns or issues in addition to ones common to networking and those specific to MPLS networks.

5. References

5.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <https://www.rfc-editor.org/info/rfc4385>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC4928] Swallow, G., Bryant, S., and L. Andersson, "Avoiding Equal Cost Multipath Treatment in MPLS Networks", BCP 128, RFC 4928, DOI 10.17487/RFC4928, June 2007, <https://www.rfc-editor.org/info/rfc4928>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

[RFC8469] Bryant, S., Malis, A., and I. Bagdonas, "Recommendation to Use the Ethernet Control Word", RFC 8469, DOI 10.17487/RFC8469, November 2018, <https://www.rfc-editor.org/info/rfc8469>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <https://www.rfc-editor.org/info/rfc8296>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <https://www.rfc-editor.org/info/rfc8964>.

[RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <https://www.rfc-editor.org/info/rfc6391>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, DOI 10.17487/RFC2780, March 2000, <https://www.rfc-editor.org/info/rfc2780>.

[I-D.ietf-mpls-1stnibble]
Kompella, K., Bryant, S., Bocci, M., Mirsky, G., Andersson, L., and J. Dong, "IANA Registry for the First Nibble Following a Label Stack", Work in Progress, Internet-Draft, draft-ietf-mpls-1stnibble-03, 9 February

2024, <https://datatracker.ietf.org/doc/html/draft-ietf-mpls-1stnibble-03>.

5.2. Informative References

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, DOI 10.17487/RFC4446, April 2006, <https://www.rfc-editor.org/info/rfc4446>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <https://www.rfc-editor.org/info/rfc4762>.

Authors' Addresses

Kireeti Kompella
Juniper Networks
1133 Innovation Way
Sunnyvale, 94089
United States of America

Email: kireeti.ietf@gmail.com

Stewart Bryant
University of Surrey 5GIC

Email: sb@stewartbryant.com

Matthew Bocci
Nokia

Email: matthew.bocci@nokia.com

Greg Mirsky (editor)
Ericsson

Email: gregimirsky@gmail.com

Loa Andersson
Huawei Technologies

Email: loa@pi.nu

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing, 100095
China

Email: jie.dong@huawei.com