INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: October 12, 2017

J. Lan
D. Cheng
Y. Hu
G. Cheng
Z. Wang
L. Sun
National Digital Switching System Engineering and Technological Research Center, P.R.China

April 12, 2017

QoS-level aware Transmission Protocol (QTP) for virtual networks
draft-lan-nvo3-qtp-05

Abstract

This document provides a QoS-level aware Transmission Protocol (QTP) for virtual networks.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Copyright and License Notice

Table of Contents

1 Introduction

1.1 Motivation and background

Virtual network has been designed to implement all types of network, to achieve scalability in today's Internet, for example, multi-tenants sharing network in data center network. This document provides a QoS-level aware Transmission Protocol (QTP) for virtual networks. Since virtual network transmits data in the form of "best effort" under today's connectionless network, its performance cannot be guaranteed. In addition, due to the elasticity of network traffic, virtual networks sharing the same physical network cannot achieve minimum resource guarantee and fairness under network congestion. QTP constructs various data transmission tunnels, namely, QTP-path, for different virtual networks that have special requests. We embed a field named QTP-path identifier (PID) in QTP header, and explicitly contains QoS level to indicate the resource level sharing when data transmission.

1.2 QTP-path

The applications on the virtual network always originate requirements that have identical path and performance. Therefore, QTP aggregates these requirements, and constructs one data transmission tunnel to maximize network throughput. The tunnel, we say, QTP-path is labeled by a QoS level according with the requests in the protocol header.

1.3 QTP Overview

We assume that all the network nodes can handle global knowledge about network loads and application requests. If there are several pairs in network originate requests for identical QoS level and have common in path, then the management plane decides to establish a QTP-path in the common part of the path. Firstly, the starting node assign a PID to the path, and send QTP-path establish request. Then the relative nodes on the path update their path information base (PIB) which storage the PIDs of QTP-paths across it. Finally, when the QTP-path has been established, the data transmitted over it can be transfer based on PIB matching PID in PDUs, and realize QoS level guarantee.

1.4 Terminology

QTP-path the tunnel established for the same requests by QTP can be used to transmission data.
QTP-path identifier represented as a 32-bit number in a 4 octet field
Node Identifier a four octet quantity used to identify an QTP Node.

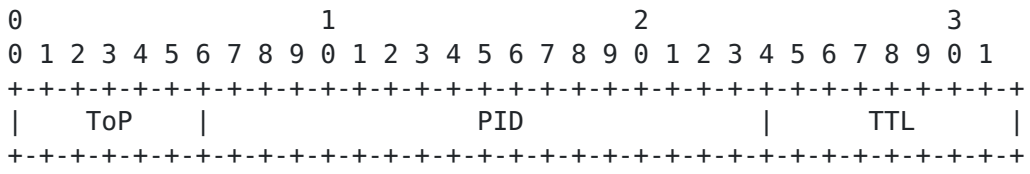
1.5 Acronyms and Abbreviations

PDU	**protocol data unit**
TTL	Time to live
ToP	Type of QTP-path
PID	QTP-path Identifier
TTL	Time to live
TLV	Type-Length-Value
PIB	QTP-path information base

2 QTP Data Encapsulation Specification

2.1 QTP Header Specification

Each packet over QTP-path MUST have a QTP header. The QTP header is represented by 4 octets. This is shown in Figure 1.



ToP: Type of QTP-path, 6 bits
PID: QTP-path Identifier, 18 bits
TTL: Time to live, 8 bits

Figure 1

The QTP header appears AFTER the data link layer header, but BEFORE any network layer header. Each QTP header is broken down into the following fields:

- Type of QTP-path (ToP)

  This six-bit field is used to identify the type of QTP-path. Each type of QTP-path SHALL carry only one class of traffic. RFC 4594 has defined twelve different classes of traffic, namely, network control, telephony, signaling, multimedia conferencing, real-time interactive, multimedia streaming, broadcast video, low-latency data, OAM, high-throughput data, standard, and low-priority data. End-to-end quantitative performance requirements may be obtained from ITU-T Recommendations Y.1541 and Y.1540. It is RECOMMENDED that the value of ToP be the same with DSCP value.

- QTP-path Identifier (PID)

  This 18-bit field carries the actual value of the QTP-Path

identifier. When a QTP-path packet is received, the PID value is looked up to obtain the next hop to which the packet is to be forwarded. Before forwarding, the PID may be replaced with another, or be popped off with the whole QTP-path header.

o Time to live (TTL)

This eight-bit field is used to encode a time-to-live value. The processing of this field is the same with that of MPLS TTL field in RFC 3032.

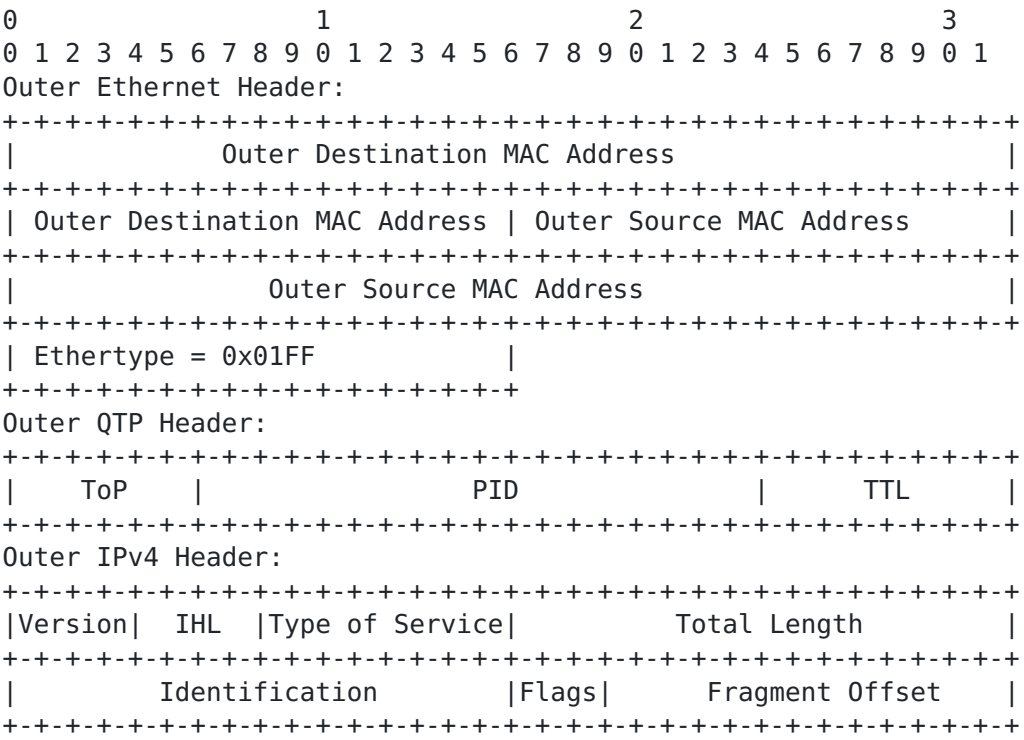
Different types of QTP-paths MUST have different behaviors on QoS guarantee and resource allocation to achieve their required traffic characteristics.

2.2 QTP Data Encapsulation

The packet format for QTP-path is shown in Figure 2.

QTP header encapsulation is used to realize a QTP-path, and the header by definition in section 2.1 contains a ToP, a PID, and a TTL.

NVGRE encapsulation is used to realize an overlay virtual network. The virtual network identifier is carried as the 24-bit VSID in the NVGRE header.



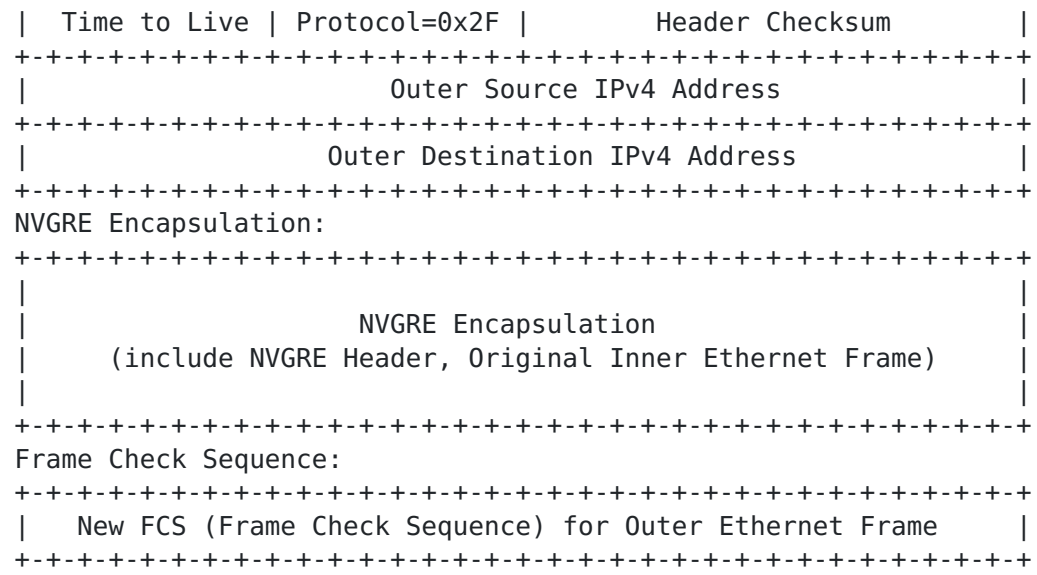


Figure 2

The QTP data encapsulation includes the outer Ethernet header, the QTP-path header, the outer IPv4 header, and the NVGRE encapsulation:

- Outer Ethernet header: The source MAC address in the outer frame is set to the MAC address associated with the NVGRE endpoint. The destination MAC address is set to the MAC address of the nexthop IP address for the destination NVE. The EtherType field is set to 0x01FF which is reserved for experimental use.

- QTP header: The ToP is used to identify the type of QTP-path, the PID is used to identify the QTP-path, and the TTL is used to record the time to live for the QTP-path.

- Outer IPv4 header: IPv4 is used as the delivery protocol for NVGRE. The protocol field in the IPv4 header is set to 0x2F. The IP address in the outer frame is referred to as the Provider Address (PA).

- NVGRE encapsulation: It includes NVGRE header, and original inner Ethernet frame. The VSID in NVGRE header can be used to identify the different virtual networks. The original inner Ethernet frame comprises of an inner Ethernet header followed by the inner IP header, followed by the IP payload.

3 QTP Message Specification

QTP message exchanges are accomplished by sending QTP protocol data units (PDUs) over TCP connections. Each QTP PDU can carry one or more

QTP messages. Note that the messages in a QTP PDU need not be related to one another. For example, a single PDU could carry a PID Request message, another message for PID Responding message, and a third Notification message that signals some event.

3.1 Destination Prefix and Node Identifiers

3.1.1 Destination Prefix (DestPrefix)

A DestPrefix specification for each QTP-Path is provided to precisely specify which packets may be mapped to each QTP-Path. The DestPrefix identifies the set of IP packets that may be mapped to that QTP-Path.

This specification defines DestPrefix as an address prefix of any length from 0 to a full address, inclusive. We give the rules used for mapping packets to QTP-Path that have been set up using a DestPrefix in the remainder of this section.

We say that a particular address matches a particular address prefix if and only if that address begins with that prefix. We also say that a particular packet matches a particular QTP-Path if and only if the packet's destination address matches the DestPrefix of that QTP-Path.

The following rules describe the procedure for mapping a particular packet to a particular QTP-Path.

- If a packet matches exactly one QTP-Path, the packet is mapped to that QTP-Path.

- If a packet matches multiple QTP-Paths, it is mapped to the QTP-Path whose matching DestPrefix is the longest.

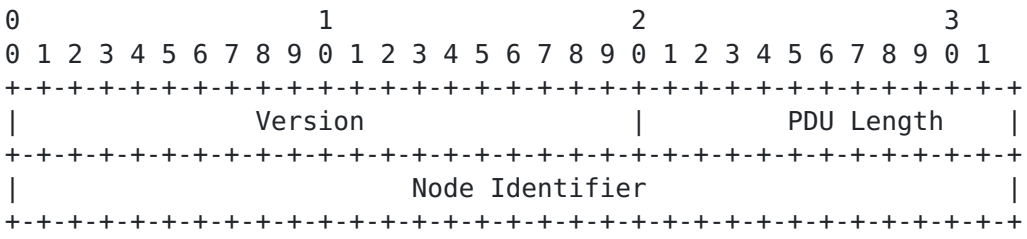
- If it is known that a packet must traverse a particular egress router, and there is a QTP-Path with a DestPrefix that is a /32 address of that router, then the packet is mapped to that QTP-Path.

3.1.2 Node Identifiers

A Node Identifier is a four octet quantity used to identify a QTP Node. The Node Identifier must be a globally unique value, such as a 32-bit router Id assigned to the QTP Node.

3.2 QTP PDU

Each QTP PDU consists of a QTP header and one or more QTP messages. The QTP header is:



Version
Two octet unsigned integer containing the version number of the protocol. This version of the specification specifies QTP protocol version 1.

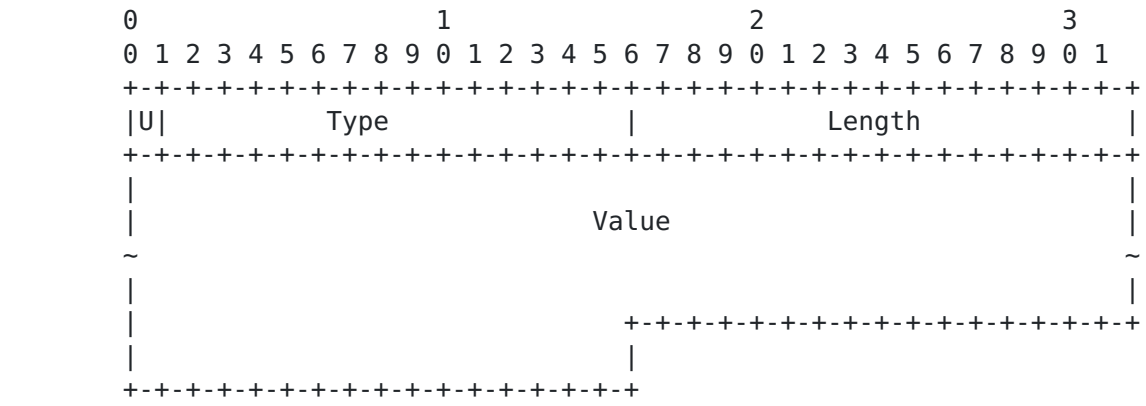
PDU Length
Two octet integer specifying the total length of this PDU in octets, excluding the Version and PDU Length fields.

Node Identifier
Four octet field that uniquely identifies the node and MUST be a globally unique value. It SHOULD be a 32-bit router Id assigned to the Node .

3.3 TLV Encoding

QTP uses a Type-Length-Value (TLV) encoding scheme to encode the information carried in QTP messages.

A QTP TLV is encoded as a 2 octet field that uses 15 bits to specify a Type and 1 bits to specify behavior when a Node doesn't recognize the Type, followed by a 2 octet Length field, followed by a variable length Value field.



U-bit
   Unknown TLV bit. When an unknown TLV is recieved, if U is clear(=0), a notification MUST be returned to the message originator and the entire message MUST be ignored; if U is set (=1), the unknown TLV MUST be silently ignored and the rest of the message processed as if the unknown TLV did not exist.

Type
   Encodes how the Value field is to be interpreted.

Length
   Specifies the length of the Value field in octets.

Value
   Octet string of Length octets that encodes information to be interpreted as specified by the Type field.

Note that there is no alignment requirement for the first octet of a TLV. TLVs may be nested, that is the Value field itself may contain TLV encodings.

The TLV encoding scheme is very general. In principle, everything in a QTP PDU could be encoded as a TLV. Note that the TLV scheme is not used to its full generality in this specification.

The specification assigns type values for related TLVs, such as the PID TLVs, from a contiguous block in the 16-bit TLV type number space. Section "TLV Summary" lists the TLVs defined in this version of the protocol.

In this section, the TLV encodings for some commonly used parameters are specified.

3.3.1 DestPrefix TLV

PIDs are bound to DestPrefixes. A DestPrefix TLV encodes a list of one or more DestPrefix Items.

DestPrefix Item 1 to DestPrefix Item n

The DestPrefix Item encoding depends on the type of DestPrefix.

A DestPrefix value is encoded as a 1 octet field that specifies the element type, and a variable length field that is the type-dependent DestPrefix value.

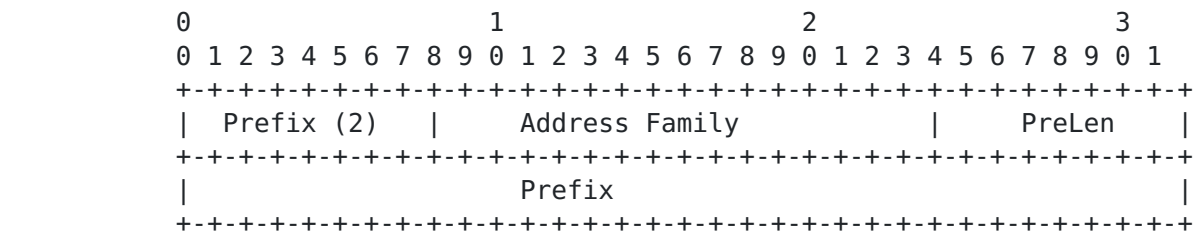
The DestPrefix value encoding is:

DestPrefix	Type	Value type name
Wildcard	0x01	No value; i.e., 0 value octets; See below.
Prefix	0x02	See below.

Wildcard DestPrefix

To be used only in the PID Release messages indicating that the release is to be applied to all DestPrefixes associated with the PID within the following PID TLV. Must be the only DestPrefix Item in the DestPrefix TLV.

DestPrefix value encoding:



Address Family
   Two octet quantity containing a value from ADDRESS FAMILY NUMBERS in [ASSIGNED_AF] that encodes the address family for the address prefix in the Prefix field.

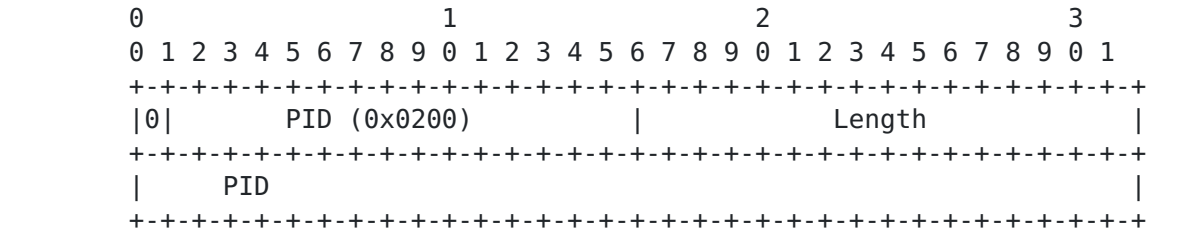
PreLen
   One octet unsigned integer containing the length in bits of the address prefix that follows. A length of zero indicates a prefix that matches all addresses (the default destination), in this case, the Prefix itself is zero octets.

Prefix
   An address prefix encoded according to the Address Family field, whose length was specified in bits in the PreLen field, padded to a byte boundary.

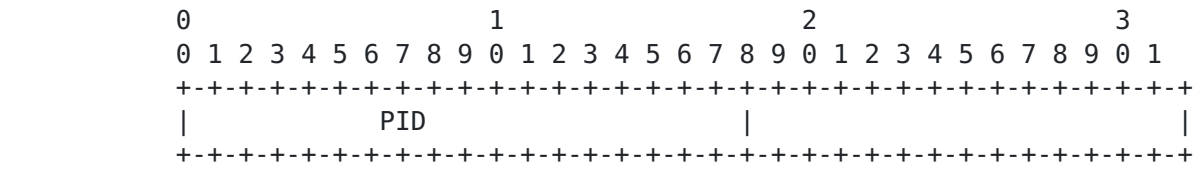
When decoding a DestPrefix TLV, if a Node encounters a DestPrefix with an Address Family it does not support, it SHOULD stop decoding the DestPrefix TLV, abort processing the message containing the TLV, and send an "Unsupported Address Family" Notification message to its QTP peer. If a Node encounters a DestPrefix type it cannot decode, it SHOULD stop decoding the DestPrefix TLV, abort processing the message containing the TLV, and send an "Unknown DestPrefix " Notification message to its QTP peer.

3.3.2 PID TLV

A PID TLV are used to encode a PID. The encoding for PID TLV is:

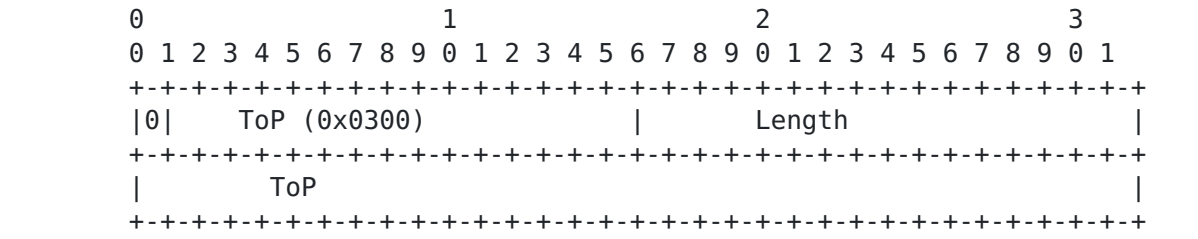


PID
   This is a 18-bit PID value represented as a 18-bit number in a 4 octet field as follows:



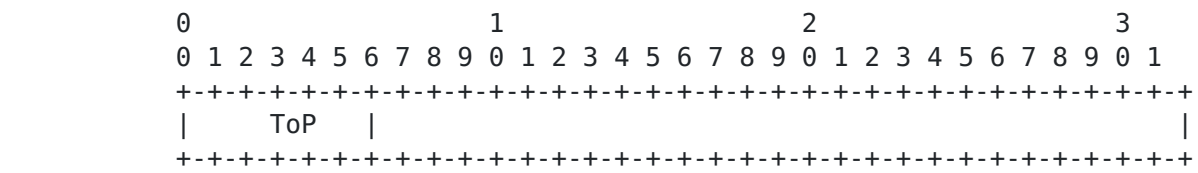
3.3.3 ToP TLV

When establishing a QTP-Path, a ToP TLV is used to specify type of QTP-Path and the QoS guarantee feature of the QTP.



ToP

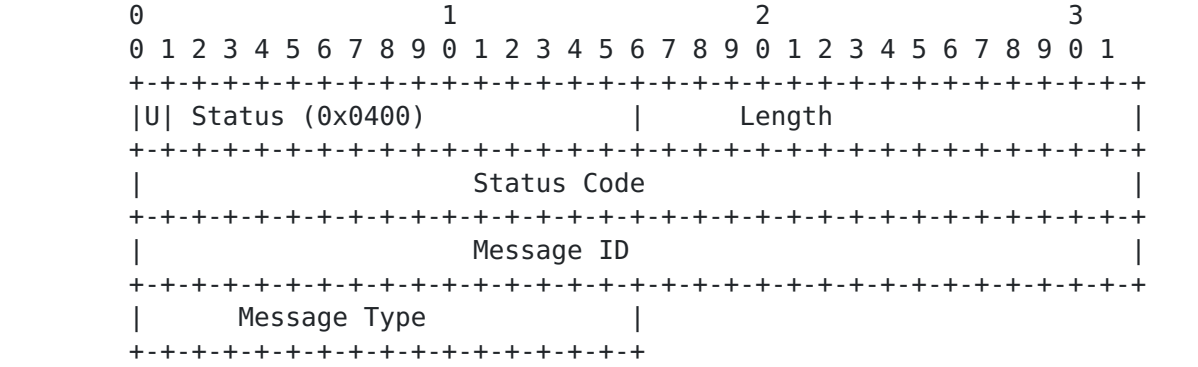
This is a 6-bit ToP value represented as a 6-bit number in a 4 octet field as follows:



A ToP TLV with Length 0 is a WildCard ToP.

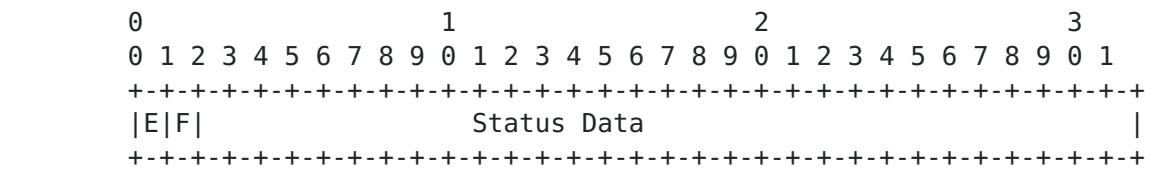
3.3.4 Status TLV

Status TLVs are carried in Notification messages to specify events being signaled. The encoding for the Status TLV is:



U-bit
SHOULD be 0 when the Status TLV is sent in a Notification message.
SHOULD be 1 when the Status TLV is sent in some other message.

Status Code
32-bit unsigned integer encoding the event being signaled. The structure of a Status Code is:



E-bit
Error bit. If set (=1), this is a fatal Error Notification. If clear (=0), this is an Advisory Notification.

F-bit
Forward bit. If set (=1), the notification SHOULD be forwarded to the Node for the next-hop or previous-hop of the QTP-Path. If clear (=0), the notification SHOULD NOT be forwarded.

Status Data
30-bit unsigned integer that specifies the status information.

Status Codes are 32-bit unsigned integers with the above encoding. The Status Codes are defined in this specification.

A Status Code of 0 signals success.

Message ID
If non-zero, 32-bit value that identifies the peer message to which the Status TLV refers. If zero, no specific peer message is being identified.

Message Type
If non-zero, the type of the peer message to which the Status TLV refers. If zero, the Status TLV does not refer to any specific message type.

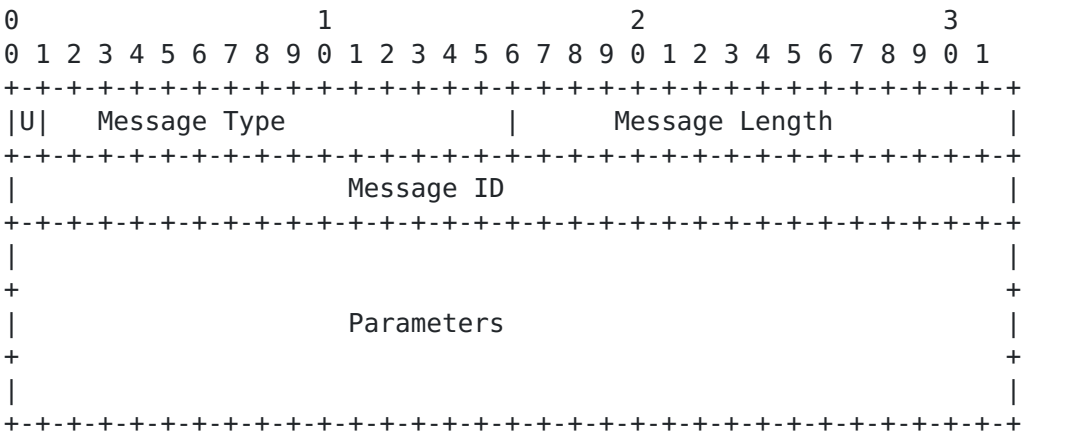
3.4 QTP messages

The messages for the QTP procedures are described in following sections.

The QTP procedures are complex and are difficult to describe fully, coherently, and unambiguously as a collection of separate message and

TLV specifications. Section "QTP Procedures" describes the QTP procedures in terms of QTP-Path events that may occur at a Node and how the Node must respond.

All QTP messages have the following format:



U-bit
Unknown message bit. When an unknown message is received, if U is clear (=0), a notification is returned to the message originator; if U is set (=1), the unknown message is silently ignored. The sections following that define messages specify a value for the U-bit.

Message Type
Identifies the type of message.

Message Length
Specifies the cumulative length in octets of the Message ID and Parameters.

Message ID
32-bit value used to identify this message. Used by the sending Node to facilitate identifying Notification messages that may apply to this message. A Node sending a Notification message in response to this message SHOULD include this Message ID in the Status TLV carried by the Notification message; see Section "Notification Message".

Parameters
Variable length set of required message parameters. Some messages have no required parameters.

For messages that have required parameters, the required parameters MUST appear in the order specified by the individual

message specifications in the sections that follow. Note that there is no alignment requirement for the first octet of a QTP message and that there is no padding at the end of a message; that is, parameters can end at odd-byte boundaries.

The following message types are defined in this specification:

Message Name	Section Title
Notification	"Notification Message"
KeepAlive	"KeepAlive Message"
PID Request	"PID Request Message"
PID Response	"PID Response Message"
PID Release	"PID Release Message"

The sections that follow specify the encodings and procedures for these messages.

3.4.1 Notification Message

A Node sends a Notification message to inform its QTP peer of a significant event. A Notification message signals a fatal error or provides advisory information such as the outcome of processing a QTP message.

The encoding for the Notification message is:

Message ID
32-bit value used to identify this message.

Status TLV
Indicates the event being signaled. The encoding for the Status TLV is specified in Section "Status TLV".

ToP TLV
Indicates the ToP associate with the event being signaled. The

encoding for the ToP TLV is specified in Section "ToP TLV".

3.4.1.1 Notification Message Procedures

If a Node encounters a condition requiring it to notify its peer with advisory or error information, it sends the peer a Notification message containing a Status TLV that encodes the information about the condition and a ToP TLV indicating the ToP associate the condition.

If a fatal error occurs, the Status Code carried in the Notification will indicate that. In this case, after sending the Notification message the Node SHOULD discard all PID-DestPrefix bindings learned from the peer.

3.4.1.2 Events Signaled by Notification Messages

For descriptive purpose, it is useful to classify events signaled by Notification messages into the following categories. Note that the ToP field is fill with a WildCard ToP TLV if not mentioned in the following circumstances.

3.4.1.2.1 Malformed PDU or Message

A QTP PDU received on a TCP connection is malformed if:

- The Node Identifier in the PDU header is unknown to the receiver, or it is known but is not the QTP Identifier associated by the receiver with the QTP peer. This is a fatal error signaled by the Bad QTP Identifier Status Code.

- The QTP protocol version is not supported by the receiver. This is a fatal error signaled by the Bad Protocol Version Status Code.

- The PDU Length field is too small (< 12) or too large (>maximum PDU length). This is a fatal error signaled by the Bad PDU Length Status Code.

A QTP message is malformed if:

- The Message Type is unknown.

  If the Message Type is < 0x8000 (high order bit = 0), it is an error signaled by the Unknown Message Type Status Code.

  If the Message Type is >= 0x8000 (high order bit = 1), it is silently discarded.

- The Message Length is too large indicating that the message extends beyond the end of the containing QTP PDU. This is a fatal error signaled by the Bad Message Length Status Code.

- The Message Length is too small, that is, smaller than the smallest possible value component. This is a fatal error signaled by the Bad Message Length Status Code.

3.4.1.2.2 Unknown or Malformed TLV

A TLV contained in a QTP message received on a TCP connection is malformed if:

- The TLV Length is too large indicating that the TLV extends beyond the end of the containing message. This is a fatal error signaled by the Bad TLV Length Status Code.

- The TLV type is unknown.

  If the TLV type is < 0x8000 (high order bit = 0), it is an error signaled by the Unknown TLV Status Code.

  If the TLV type is >= 0x8000 (high order bit = 1), the TLV is silently dropped.

- The TLV Value is malformed. This occurs when the receiver handles the TLV but cannot decode the TLV Value. It is a fatal error signaled by the Malformed TLV Value Status Code.

3.4.1.2.3 Session KeepAlive Timer Expiration

This is a fatal error signaled by the KeepAlive Timer Expired Status Code.

3.4.1.2.4 Unilateral Connection Shutdown

This is a fatal event signaled by the Shutdown Status Code.

3.4.1.2.5 Events Resulting from Other Messages

Messages may result in events that must be signaled to QTP peers via Notification messages. These events and the Status Codes used in the Notification messages to signal them are described in the sections that describe these messages.

3.4.1.2.6 Internal Errors

A QTP implementation may detect problem conditions specific to its

implementation. When such a condition prevents an implementation from interacting correctly with a peer, the implementation should, when capable of doing so, use the Internal Error Status Code to signal the peer. This is a fatal error.

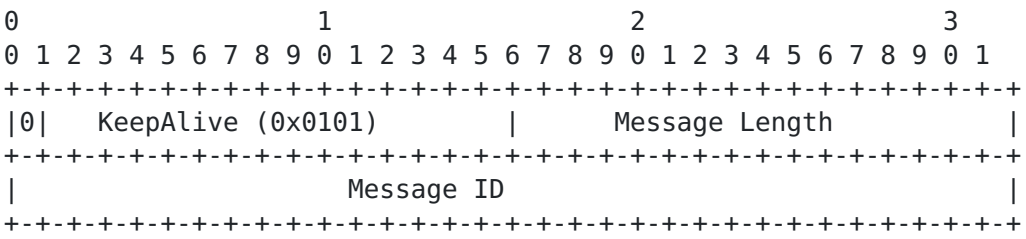
3.4.1.2.7 Miscellaneous Events

These are events that fall into none of the categories above. There are no miscellaneous events defined in this version of the protocol.

3.4.2 KeepAlive Message

A Node sends KeepAlive messages as part of a mechanism that monitors the integrity of the TCP connection with another peer.

The encoding for the KeepAlive message is:



Message ID
   32-bit value used to identify this message.

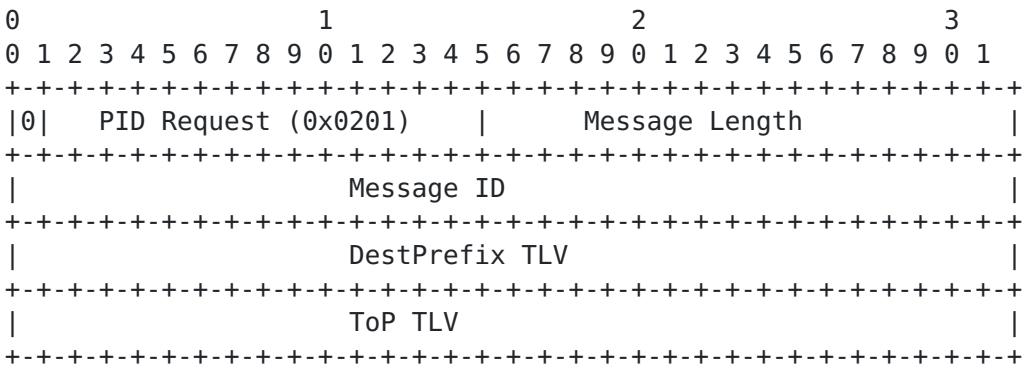
3.4.2.1 KeepAlive Message Procedures

The KeepAlive Timer resets a KeepAlive Timer every time a QTP PDU is received. The KeepAlive message is provided to allow reset of the KeepAlive Timer in circumstances where a Node has no other information to communicate to a peer.

A Node MUST arrange that its peer receive a QTP message of any type from it at least every KeepAlive Time period. But a KeepAlive message MUST be sent in circumstances where no other QTP protocol messages have been sent within the period.

3.4.3 PID Request Message

A Node sends the PID Request message to a peer to request a binding (mapping) for a DestPrefix. The encoding for the PID Request message is:



Message ID
32-bit value used to identify this message.

DestPrefix TLV
The DestPrefix for which a PID is being requested. See Section "DestPrefix TLV" for encoding.

ToP TLV
The ToP of the QTP being established. See Section "ToP TLV" for encoding.

3.4.3.1 PID Request Message Procedures

The Request message is used by an upstream Node to request that the downstream Node assign and advertise a PID for a DestPrefix of a specific ToP.

A Node may transmit a Request message under any of the following conditions:

1. The Node is informed by the management modular (manually or automatically) to establish a QTP for the given DestPrefix and the given ToP while it has recognized the DestPrefix via the forwarding table with the next hop being a QTP peer and the Node not already having a mapping from the next hop for the given DestPrefix.

2. The next hop to the DestPrefix changes, and the Node doesn't already have a mapping from that next hop for the given DestPrefix.

   Note that if the Node already has a pending PID Request message for the new next hop, it SHOULD NOT issue an additional PID Request in response to the next hop change.

3. The Node receives a PID Request for a DestPrefix from an

upstream peer, the DestPrefix next hop is a QTP peer, and the Node doesn't already have a mapping from the next hop.

The receiving Node SHOULD respond to a PID Request message with a PID Response message with a requested PID or with an error status indicating why it cannot satisfy the request.

When the DestPrefix for which a PID is requested is a DestPrefix, the receiving Node uses its routing table to determine its response. Unless its routing table includes an entry that exactly matches the requested Prefix, the Node MUST respond with a No Route Notification message.

This version of the protocol defines the following Status Codes for the Notification message that signals a request cannot be satisfied:

No Route
: The DestPrefix for which a PID was requested includes a DestPrefix for which the Node does not have a route.

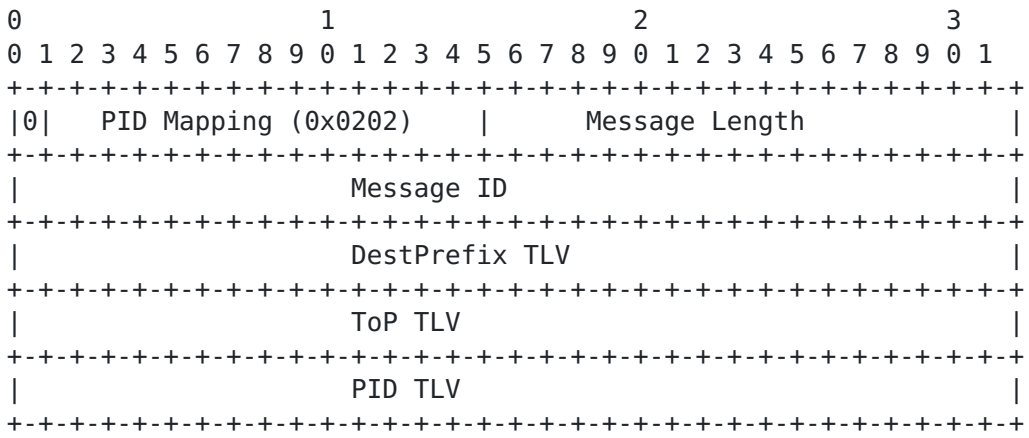
No PID Resources
: The Node cannot provide a PID because of resource limitations. When resources become available, the Node MUST notify the requesting Node by sending a Notification message with the PID Resources Available Status Code.

See Section "QTP Procedures" for more details.

3.4.4 PID Response Message

A Node sends a PID Response message to a peer to advertise DestPrefix-PID bindings to the peer.

The encoding for the PID Mapping message is:



Message ID
   32-bit value used to identify this message.

DestPrefix TLV
   Specifies the DestPrefix of the DestPrefix-PID mapping being advertised. See Section " DestPrefix TLVs" for encoding.

ToP TLV
   Specifies the ToP of the DestPrefix-PID mapping being advertised. See Section " ToP TLVs" for encoding.

PID TLV
   Specifies the PID component of the DestPrefix-PID mapping. See Section "PID TLV" for encoding.

3.4.4.1 PID Response Message Procedures

The PID Response message is used by a Node to distribute a PID for a DestPrefix to a QTP peer.

A Node is responsible for the consistency of the PID mappings it has distributed and that its peers have these mappings.

A Node receiving a PID Response message from a downstream Node for a DestPrefix SHOULD NOT use the PID for forwarding unless its routing table contains an entry that exactly matches the DestPrefix.

See Section "QTP Procedures" for more details.

3.4.5 PID Release Message

An Node sends a PID Release message to an LDP peer to signal the peer that the Node no longer needs specific DestPrefix-PID mappings previously requested of and/or advertised by the peer.

The encoding for the PID Release Message is:

Message ID
32-bit value used to identify this message.

DestPrefix TLV
Identifies the DestPrefix for which the DestPrefix-PID mapping is being released.

ToP TLV
Identifies the ToP for which the DestPrefix-PID mapping is being released.

PID TLV
The PID being released (see procedures below).

3.4.5.1 PID Release Message Procedures

A Node transmits a PID Release message to a peer when it no longer needs a PID previously received from or requested of that peer.

A Node MUST transmit a PID Release message under any of the following conditions:

1. The Node that sent the PID Response message is no longer the next hop for the mapped DestPrefix.

2. The Node receives a PID Response message from a Node that is not the next hop for the DestPrefix.

The DestPrefix TLV specifies the DestPrefix for which PIDs are to be released. If a WildCard PID TLV is received, all PIDs associated with the DestPrefix and the ToP are to be released; otherwise, only the

PID specified in the PID TLV is to be released.

The DestPrefix TLV may contain the Wildcard DestPrefix; In this case, if the PID Release message contains a none-WildCard PID TLV, then the PID is to be released for all DestPrefixes to which it is bound. If there is a WildCard PID TLV in the PID Release message, then the sending Node is releasing all PID mappings previously learned from the receiving peer of the ToP in the message.

See Section "QTP Procedures" for more details.

3.5 Summary

3.5.1 Message Summary

The following are the QTP messages defined in this version of the protocol.

Message Name	Type	Section Title
Notification	0x0001	"Notification Message"
KeepAlive	0x0101	"KeepAlive Message"
PID Request	0x0201	"PID Request Message"
PID Response	0x0202	"PID Response Message"
PID Release	0x0203	"PID Release Message"

3.5.2 TLV Summary

The following are the TLVs defined in this version of the protocol.

TLV	Type	Section Title
DestPrefix	0x0100	"DestPrefix TLV"
PID	0x0200	"PID TLV"
ToP	0x0300	"ToP TLV"
Status	0x0400	"Status TLV"

3.5.3 Status Code Summary

The following are the Status Codes defined in this version of the protocol.

The "E" column is the required setting of the Status Code E-bit; the "Status Data" column is the value of the 32-bit Status Data field in the Status Code TLV.

Status Code	E	Status Data	Section Title

Success	0	0x00000000	"Status TLV"
Bad LDP Identifier	1	0x00000001	"Events Signaled by ..."
Bad Protocol Version	1	0x00000002	"Events Signaled by ..."
Bad PDU Length	1	0x00000003	"Events Signaled by ..."
Unknown Message Type	0	0x00000004	"Events Signaled by ..."
Bad Message Length	1	0x00000005	"Events Signaled by ..."
Unknown TLV	0	0x00000006	"Events Signaled by ..."
Bad TLV Length	1	0x00000007	"Events Signaled by ..."
Malformed TLV Value	1	0x00000008	"Events Signaled by ..."
Shutdown	1	0x00000009	"Events Signaled by ..."
Unknown DestPrefix	0	0x0000000A	"DestPrefix TLV"
No Route	0	0x0000000B	"PID Request Message"
No PID Resources	0	0x0000000C	"PID Request Message"
PID Resources/ Available	0	0x0000000D	"PID Request Message"
KeepAlive Timer/ Expired	1	0x0000000E	"Events Signaled by ..."
Unsupported Address/ Family	0	0x0000000F	"DestPrefix TLV"
Internal Error	1	0x00000010	"Events Signaled by ..."

4 QTP Procedure

4.1 Establishment of QTP-Path

4.1.1 The trigger condition of QTP-Path establishing

When the business is running in the virtual network, the data can be transmitted through QTP-Path. After the management plane determines establish QTP-Path between two nodes, it carries out the process of QTP-Path establishing through the control plane. The determination of QTP-Path establishing can be obtained through the management plane's gathering business requests and the perception of network status, or the network administrator manually configures. Once the establishment determination is made, the control plane needs the parameters of QTP-Path, i.e. source node, destination node, flow types and so on, after obtaining those parameters, the control plane will establish and maintain QTP-Path according to the protocol.

4.1.2 Path establishment process

Path establishment start from the source node, and sends request message hop by hop. The node receives the message and judges if it meets the requests, if no, the node sends warning message reversely, else it continues to send request message to the next node until to the destination node. The destination node sends reply message reversely along the path to the source node, the establishment of QTP-Path is successful. This section describes the establishment

process in detail.

A Node processes a received PID Request message as follows:

1. The Node looks up in the local routing table for the next hop of DestPrefix. If the node can not find the next hop, it sends Notification message of "No Route", and the process goes to 8.

2. Check whether the node has established TCP connection with the next hop, if not, this node initiates a request for establishing the TCP connection, and after the connection is done, start KeepAlive timer. If the connection can not be established, then the node sends Notification message of "ShutDown", and the process goes to 8.

3. Check whether PIB consists tables of DestPrefix and ToP, and judges whether the next hop in PIB is the one in routing tables:

   a. If so, insert the PID value of the table to PID Response Message, and forward the message to the upstream node, then the process goes to 8.

   b. If not,insert the PID value of the table to PID Release,and forward the message to the next hop, then delete the table.

4. Check whether this node supports ToP traffic forwarding, if not, forward Notification message of " Internal Error" to the upstream node, then the process goes to 8.

5. Check whether this node has residual PIDs, if not, forward Notification message of " No PID Resources" to the upstream node, then the process goes to 8.

6. Forward the message of PID Request message to the next hop, and wait to receive the downstream message.

7. The reply message received from the downstream nodes:

   a. If the message is PID Response, insert new tables which consist of DestPrefix, ToP, PID, NextPID, Nexthop etc.to PIB according to the PID value in reply message. Insert the PID value locally generated to PID Response Message, and forward the message to upstream nodes.

   b. If the message is Notification, then forward the message to upstream nodes

8. Over

4.2. Removal of the QTP-Path

4.2.1 The trigger condition of removal process

When the management plane determines to remove QTP-Path between two nodes, it carries out the removal process of QTP-Path through the control plane. The determination of QTP-Path removal can be obtained through the perception of network status and learning, or the network administrator manually configures. After the removal determination is made, the management plane gives the source node, destination node and business types, in this way, the control plane can carry out the removal process successfully.

4.2.2 QTP-Path removal process

The removal of QTP-Path starts from the source node and sends removal message along the path.

The procedure in which node processes PID Release message is as follows:

1. If the message comes from upstream nodes, forward it to downstream nodes; if it comes from downstream nodes; forward it to upstream nodes.

2. If the value of PID equals to the value of WildCard, delete the tables in PIB which meets DestPrefix and ToP, else delete the tables in PIB which meets DestPrefix, ToP and PID

4.3. QTP-Path adjustment process

4.3.1 The trigger condition of QTP-Path adjustment

When the management plane determines to adjust QTP-Path between two nodes, it carries out the adjustment process of QTP-Path through the control plane. The determination of QTP-Path adjustment can be obtained through the management plane's gathering business requests and the perception of network status, or the network administrator manually configures. Under the two conditions below, the management plane will adjust QTP-Path:

1. A link of QTP-Path fails.

2. A link or a section of QTP-Path congests, resulting in that data transmission can not meet business needs. Once the adjustment determination is made, the control plane needs the

parameters of QTP-Path, e.g. source node, destination node, business types etc., after obtaining those parameters, the control plane will adjust QTP-Path according to the protocol.

4.3.2 QTP-Path Adjustment Process

When the management plane determines to adjust QTP-Path between two nodes, it sends QTP-Path establishment message to source nodes, then the source node sends PID Request message to downstream nodes. The adjustment process is similar to the establishment process. Because the adjustment of QTP-Path is based on the link availability or QoS performance, the adjustment process depends on the routing protocol on the node.

5 Security Considerations

TBA.

6 IANA Considerations

This memo includes no request to IANA.

7 References

7.1 Normative References

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

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, August 2006.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[I-D.sridharan-virtualization-nvgre] Sridharan, M., Greenberg, A., Venkataramaiah, N., Wang, Y., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-02 (work in progress), February 2013.

7.2 Informative References

[EVILBIT] Bellovin, S., "The Security Flag in the IPv4 Header", RFC 3514, April 1 2003.

[RFC5513] Farrel, A., "IANA Considerations for Three Letter Acronyms", RFC 5513, April 1 2009.

[RFC5514] Vyncke, E., "IPv6 over Social Networks", RFC 5514, April 1 2009.

Authors' Addresses

Julong Lan
National Digital Switching System Engineering and Technological Research Center
NDSC, No.7 , Jianxue Street,Jinshui District
Zhengzhou, 450002
P.R.China

Phone: +86-371-8163-2988
Email: ndscljl@163.com
URI: http://www.ndsc.com.cn/

Dongnian Cheng
National Digital Switching System Engineering and Technological Research Center
NDSC, No.7 , Jianxue Street,Jinshui District
Zhengzhou, 450002
P.R.China

Phone: +86-371-8163-2743
Email: cdn@mail.ndsc.com.cn
URI: http://www.ndsc.com.cn/

Yuxiang Hu
National Digital Switching System Engineering and Technological Research Center
NDSC, No.7 , Jianxue Street,Jinshui District
Zhengzhou, 450002
P.R.China

Phone: +86-371-8163-2737
Email: chxachxa@126.com

Guozhen Cheng
National Digital Switching System Engineering and Technological

Research Center
NDSC, No.7 , Jianxue Street,Jinshui District
Zhengzhou, 450002
P.R.China

Phone: +86-371-8163-2725
Email: chengguozhen1986@163.com

Zhiming Wang
National Digital Switching System Engineering and Technological
Research Center
NDSC, No.7 , Jianxue Street,Jinshui District
Zhengzhou, 450002
P.R.China

Phone: +86-371-8163-2651
Email: wangzm05@gmail.com

Lu Sun
National Digital Switching System Engineering and Technological
Research Center
NDSC, No.7 , Jianxue Street,Jinshui District
Zhengzhou, 450002
P.R.China

Phone: +86-371-8163-2846
Email: sunlu@gmail.com