IPv6 Working Group
Internet Draft
draft_qos.txt

Rahul Banerjee
N. Preethi
M. Sethuraman
BITS, Pilani (India)
January 2002

Design and Implementation of the Quality-of-Service in IPv6 using the modified Hop-by-Hop Extension header - A Transitional Mechanism

Status of This Memo

Abstract

This paper proposes a temporary solution to the QoS implementation in IPv6, the design of which uses the Hop-by-Hop Extension header and not the 20-bit flow label field in the IPv6 main header. This paper deals extensively with Integrated Services type of QoS model (like the one supported by RSVP) and gives the definition of the important TLV options that will be needed to specify the Type of QoS and the corresponding resource requirements in the Hop-by-Hop Extension Header. This design can also support the Differentiated Services type of QoS model, which is dealt in a brief manner. The paper also elaborates on the data structures that will be required at the routers and finally gives the algorithm that the source and the router should follow while trying to implement this design.

Table of Contents

1. Introduction

This paper talks about the design and gives an overview of the implementation details of Quality of Service (QoS) in IPv6. Though IPv6 main header has a 20-bit flow label field for QoS implementation purposes, it has not yet been exploited. This paper doesn’t talk of using those 20 bits but explores the possibility of using the hop-by-hop extension header to implement QoS. This design is based on the Integrated Services model and can act as an effective temporary solution till the specifications to use the 20-bit flow control field in the IPv6 main header is developed.

2. Motivation for using the hop-by-hop extension header implementing QoS

To implement any model of QoS, all the routers en-route have to be requested for the particular resources required and it is important that they give their consent on the same. The hop-by-hop extension header is one that will be processed by all the routers en-route to the destination. So all the routers in the path will see any information that is embedded in this header.

The TLV options in the hop-by-hop extension header have not yet been fully exploited. By exploiting those options to our convenience, it is possible to specify the requisite information for each flow (i.e. the type and the resources required) to all the intermediate routers. The individual routers can send appropriate messages to the source if it cannot meet the resource requirements.

3. The Hop-by-Hop Extension header

Before using the Hop-by-Hop Extension header, it is necessary to know about it in detail. According to RFC 2460 – the formal Specification for IPv6, the Hop-by-Hop Extension Header is used to carry optional information that must be examined by every node along a packet's delivery path. It is identified by a Next Header value of 0 (Zero) in the IPv6 header, and has the following format:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                                                               |
.                                                               .
.                            Options                            .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next Header: It’s an 8-bit field that identifies the type of header immediately following the Hop-by-Hop Options header.

Hdr Ext Len: It’s an 8-bit unsigned integer field, which tells the length of the Hop-by-Hop Options header in 8-octet units, not including the first 8 octets.

Options: It’s a variable-length field, of length such that the complete Hop-by-Hop Options header is an integer multiple of 8 octets long.

4. Type – length – value (TLV) options

4.1 Introduction

The hop-by-hop options header can carry a variable number of TLV encoded "options", of the following format [RFC 2460]:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
|  Option Type  | Opt Data Len | Option Data
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
```

Option Type: 8-bit identifier of the type of option.
Opt Data Len: 8-bit unsigned integer. Length of the Option Data field of this option, in octets.
Option Data: Variable-length field. Option-Type-specific data.

The sequence of options within a header must be processed strictly in the order they appear in the header; a receiver must not, for example, scan through the header looking for a particular kind of option and process that option prior to processing all preceding ones.

The Option Type identifiers are internally encoded such that their highest-order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type.

00 - skip over this option and continue processing the header.
01 - discard the packet.
10 - discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.
11 - discard the packet and, only if the packet's Destination Address was not a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.

The third-highest-order bit of the Option Type specifies whether or not the Option Data of that option can change en-route to the packet's final destination.

0 - Option Data does not change en-route
1 - Option Data may change en-route

The three high-order bits described above are treated as part of the Option Type, not independent of the Option Type. That is, a full 8-bit Option Type, not just the low-order 5 bits of an Option Type, identifies a particular option.

4.2 The Already defined TLV options

The only hop-by-hop options defined in RFC 2460 (IPv6 Specification) are the Pad1 and PadN options specified as follows:

4.2.1 Pad1 option

```
+-+-+-+-+-+-+-+-+
|       0       |
+-+-+-+-+-+-+-+-+
```

The format of the Pad1 option is special in the sense that it does not have length and value fields. The Pad1 option is used to insert one octet of padding into the Options area of a header. [RFC 2460].

4.2.2 PadN option

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
|       1       | Opt Data Len | Option Data
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- - - - - - - - -
```

The PadN option is used to insert two or more octets of padding into the Options area of a header. For N octets of padding, the Opt Data Len field contains the value N-2, and the Option Data consists of N-2 zero-valued octets. [RFC 2460]

4.2.3 The router alert option

This option has been defined in the RFC 2711 - titled Router Alert Option and has the following format:

```
  Type         Length=2          Value
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0|0 0 1 0 1|0 0 0 0 0 0 1 0|     Value (2 octets)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The first three bits of the first byte are zero and the value 5 in the remaining five bits is the Hop-by-Hop Option Type number. By zeroing all three, this specification requires that, nodes not recognizing this option type should skip over this option and continue processing the header and that the option must not change en route.

The above 3 are the options that have been defined in RFCs. The rest of the values for the option type of the hop-by-hop options header haven't been defined yet. [RFC 2711]

5. Using the TLV options to implement QoS

This design hopes to exploit the remaining non-defined and possible values of the option type in the Hop-by-Hop options header, (after leaving some values for future use) to indicate some important QoS types.

5.1 QoS Models and their representation in the options field

Since this work focuses to provide a transitional mechanism for providing QoS-support (by complementing the 20-bit flow control field in the IPv6 base header), it deals with a Integrated Services (IntServ) model like that supported by RSVP [Paul et al.], wherein each and every flow needs to specify its TYPE and the RESOURCES that it needs en-route. (This design can also support the Differentiated Services (DiffServ) model of QoS, in which case each flow is aggregated to a particular class of traffic. This design can then act as a substitute for the concept behind the Traffic Class bits (8-bit field) in the IPv6 base header.)

The source tells the routers that it is using the Integrated Services model by setting the nineteenth bit of the first 32 bits.

```
 0               8               16    Type       24  Length
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header   |  Hdr Ext Len  | 0 0 0|1 0 0 0 0|              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                    Options data                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Differentiated Services (DiffServ) feature, if required, can be specified by setting the twentieth bit of the first 32 bits.

```
   0              8               16    Type       24  Length
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Next Header  |  Hdr Ext Len  | 0 0 0|0 1 0 0 0|              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 +                   options data                                +
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This report talks extensively of the IntServ model only and hopes that the DiffServ model gets developed in the future.

5.2 The IntServ Model

The two main Types of flows in the IntServ model are [Paul et al]
Guaranteed flow service
Controlled Load Service

The last three bits of the Type field i.e. the bits numbered 21, 22, 23 are used to represent one of these types.

```
   0              8               16    Type       24  Length
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Next Header  |  Hdr Ext Len  | 0 0 0|0 0 0 0 0|              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 +                   options data                                +
 |                                                               |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

There are a total of 8 possible combinations out of which the IntServ model uses two. The rest can be can be exploited by the DiffServ model and future use.

5.2.1 The QoS Identifier

This is an 8-bit identifier and occupies the first byte in the options data field as shown in the figure below. There might be many applications from the same source wherein each one has its own flow specifications. So there arises a need to uniquely identify each such flow. The QoS identifier does this job. A particular source can establish a maximum of 256 connections that need QoS guarantee.

```
 0               8               16    Type       24  Length
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  | 0 0 0|0 0 0 0 0|              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| QoS Identifier|                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.2.2 Resource Identifier

This is a 4-bit identifier that specifies the type of the resource needed by a particular flow. The different types of resources needed are indicated using these identifiers in a list. This list follows the QoS Identifier in the option data field, which in turn is followed by a list of 32 bit values that specify the amount of resource required for each of the resource types. Some of the identified resource types are:

0000 – End of List Identifier
This is a special identifier that specifies the end of the resource-required list (brief explanation in section 5.2.3).

0001 – Constant Data Transfer Rate
This identifies the Constant Bandwidth required and the value is given in a 32-bit field specified in Kbps (Kilo bits per second). (Max value = 512 GBps)

0010 – Average Data Transfer Rate
This identifies the Average Bandwidth required and the value is given in a 32-bit field in Kbps (Kilo bits per second).

0011 – Maximum Data Transfer Rate
This identifies the Maximum Bandwidth required and the value is given in a 32-bit field specified in Kilobits per second (Kbps).

0100 – Minimum Delay Requirement
This identifies the Minimum Delay that the application demands and the required value is given in a 32-bit field specified in nanoseconds. (Max value = 4.3 sec)

0101 – Average Delay Requirement
This identifies the Average end-to-end delay that the application can tolerate and the value is given in a 32-bit field specified in nanoseconds.

0110 – Buffer Requirement
This identifies the Buffer Requirement by the flow at each router and the amount required is expressed as a 32-bit quantity specified in bytes. (Max value = 4 GB)

5.2.3 Resources Required List

The Type of flow (Guaranteed/Controlled Load, briefly explained in section 5.3) is specified in the option type bits of the Hop-by-Hop Extension header. The resources needed by this flow at each router are specified in the bits following the 8-bit QoS identifier in the options data field. The resource identifiers (4 bits each) are specified one after the other and the list ends with the 0000 End of List Identifier (as mentioned above). The corresponding amount of resource required (a 32 bit quantity only) for all the resource types listed is specified in the same order as that of the resource types, starting from the next aligned 32 bits.

5.3 The Different TYPES OF FLOW in the IntServ Model

5.3.1 Guaranteed Flow Service

This service is meant for RTI (Real Time Intolerant) or hard Real Time applications, which demand minimal latency and jitter. For example, consider a two-person videoconference. Delay is unacceptable and ends should be brought as close as possible. [Paul et al] The whole application should simulate two persons talking face to face. For this videoconference case, the required resource reservations are
a. Constant bandwidth for the application traffic
b. Deterministic Minimum delay that can be tolerated.
These types of applications can decrease delay by increasing demands for bandwidth.

5.3.1.1 Option Definition

The above example of a two-person videoconferencing, which is a Guaranteed Type of Service, can be defined in the following way.

```
 0               8               16    Type       24  Length
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  | 1 0 0|1 0 0 1 0|              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| QoS Identifier| 0 0 0 1 0 1 0 0 0 0 0 0                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             32 bit value – constant bandwidth in bps          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             32 bit value – min delay in microseconds          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.3.1.2 Explanation

The first 3 bits being 1,0,0 say that if the router is not able to recognize the option type, it should discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type and the value of the option data field should not be changed en route by any routers.

The value of 18 in the 5 bits numbered 19,20,21,22,23 defines this QoS type of IntServ and Guaranteed Service. The numeric decimal value specifying this type is 146.

5.3.1.3 The Resource Required List and its Specification

a. Constant Bandwidth Requirement: The bit value of 0001 after the QoS identifier is the identifier for this and the first 32-bit value gives the amount of bytes to be reserved.

b. Minimum delay Requirement: The deterministic minimal delay nanoseconds. The identifier is 0010 and the second 32-bit value corresponds to this.
The 0000 identifier ends this list.

5.3.1.4 Examples

Interactive applications like Videoconferencing/Audio Conferencing or real time applications.

5.3.2) Controlled Load Service

This service is meant for RTT (Real Time tolerant) or soft Real Time applications, which have an average bandwidth requirement and an indeterminate end-to-end delay for an arbitrary packet. [Paul et al]. These RTI applications demand weak bounds on the maximum delay over the network. Occasional packet loss is acceptable. For example, consider video applications which use buffering.

The required resource reservations can be

a. Average bandwidth for the application traffic
b. Buffer requirement at each relevant intermediate router

5.3.2.1 Option Definition:

The above example can be defined as follows.

```
 0               8               16    Type       24  Length
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  | 1 0 0|1 0 0 1 1|              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| QoS Identifier| 0 0 1 0 0 1 1 0 0 0 0 0                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             32 bit value –average bandwidth in bps            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             32 bit value –buffer req. in bytes                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.3.2.2) Explanation

The first 3 bits being 1,0,0 say that if the router is not able to recognize the option type, it should discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type and the value of the option data field should not be changed en route by any routers.

The value of 19 in the 5 bits numbered 19,20,21,22,23 defines this QoS type of IntServ and Controlled Load Service. The numeric decimal value specifying this type is 147.

5.3.2.3 The Resource Required List and its Specification.

a. Average Bandwidth Requirement: The bit value of 0010 after the QoS identifier is the identifier for this and the first 32-bit value gives the required value in Kbps.

b. Buffer Requirement: The bit value of 0110 following the Average Bandwidth Resource type is the identifier for this and the second 32 bit value gives the number of bytes to be reserved.
This list is ended by the 0000 identifier.

5.3.2.4) Examples

Video/Audio applications that require buffering like video mail, voice mail.

5.4 Overview of some important facts.

There are two Types of Flows (Guaranteed/Controlled Load). Under each type, the Resource Requirement List can vary for each and every application that needs QoS. The application has to specify the following.

-Whether it requires Guaranteed/Controlled Load treatment
-The List of Resources it requires
-The required amount of these resources.

For applications that do not need QoS, it can specify the No Flow Control option defined as defined in the next section

5.5 No Flow Management

This type indicates that the source requires no QoS and will be content with a 'best effort' treatment.

5.5.1 Option Definition

The option type is defined as

```
  0               8               16    Type        24  Length
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Next Header   |  Hdr Ext Len  | 0 0 0|0 0 0 0 0|0 0 0 0 0 0 0 1|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The value of 0 in the least significant 5 bits numbered 19,20,21,22,23 signifies the type for No QoS required at the intermediate routers. The numeric decimal value specifying this type is 0. But it is different from the Pad1 option in the following way. The Pad1 option doesn’t have a length and data field. But the No flow control option has a value of 1 in the length field and no data field.

6. At the Router

Any router that tries to implement QoS maintains a QoS routing table and keeps track of the QoS available to each destination through the required number of hops. [RFC 2676]. Apart from this table, the router needs to keep track of the allotted QoS to each and every flow. This table is the AllottedQoS table.

6.1 The AllottedQoS table

It has the following entries:

1. Source address

2. QoS identifier for that particular flow from the source.

3. Information regarding whether it is the IntServ Model or the Diffserv Model.

```
Enum MODEL_ID{
INTSERV=0, // the IntServ Model
DIFFSERV=1 // the DiffServ Model
};
```

4. List of resources allotted to that entry (i.e.) an array of values like the following.

```
Struct RESOURCE_ALLOCATED{
Short int Res_identifier; //the 4 bit identifier of the resource
Int Res_allocated; //the 32 bit value of the allocated resource
};
```

6.2 Resource Required List

The list of resources will be an array of pointers to the structure RESOURCE_ALLOCATED as declared below.

```
Struct RESOURCE_ALLOCATED *res_allocated[MAX];
```

This array will be maintained for each source address. The QoS Identifier will be the array subscript for each source. The pointer value stored acts as the head of the list of the resources allotted for that particular QoS identifier.

6.3 Defining the different Resource Identifiers

```
enum RES_ID{
ENDOFLIST=0, // End of List Identifier
CONSTBW  =1, // Constant Data Transfer Rate
AVBW     =2, // Average Data Transfer Rate
MAXBW    =3, // Maximum Data Transfer Rate
MINDELAY =4, // Minimum Delay Requirement
AVDELAY  =5, // Average Delay Requirement
BUFFREQ  =6  // Buffer Requirement
};
```

6.4 Template for the AllottedQos table entry

```
#define MAX 256 //maximum of 256 QoS Ids for every source
typedef struct {
struct sockaddr_in6 *srcaddr; //the source IPv6 address
struct  RESOURCE_ALLOCATED  *res_allocated[MAX];  //a  pointer  which
//acts  as  the  head  for  each  of  the  lists  i.e.  for  each  of  the
//0..MAX QoS Identifiers for the particular source address.
MODEL_ID model; // IntServ or DiffServ
}ALLOTTEDQOS_TABLE;
```

7. Overview of the whole design.

This section describes the whole process by taking an example. Consider any application (like Videoconferencing or Video/Audio on Demand) that needs some specified QoS.

7.1 Function of the Source

It gets a unique QoS Identifier for that particular flow and fills it in the Hop-by-Hop header.
It specifies the IntServ model by setting the appropriate bit.
The source application then fills in the resource-required list and the corresponding 32 bit values (the amount of each resource needed) in the options data part of the Hop-by-Hop header.
This packet is put on the network and it reaches the intermediate routers.

7.2 Function of each relevant intermediate router

7.2.1 Initial Processing

It gets the option type value from the header.
Checks if its the default (no QoS required which is indicated by a value of all bits being 0 in the 5 bits numbered 19,20,21,22,23
If its not the default QoS, it gets the QoS identifier from the first byte of the options data field.

7.2.2 Searching for the entry

1. The ALLOTTED_QOS table is searched based on the source address.
2. If an entry is found, then for that particular source, a search is made based on the QoS Identifier got during the Initial Processing stage. (the array index for the res_allocated structure is the corresponding QoS Identifier and this pointer is NULL if its a new entry).
3. If the entry already exists, the IPv6 packet is processed so that the reserved QoS is met.
4. If the entry is not found, a new entry is made in the ALLOTTED_QOS table for the source and the QoS Identifier and further processing of this new entry is done as follows.

7.2.3 New Entry

1. The router now checks if its the IntServ Model or the Diffserv Model by checking the appropriate bits in the options type field and stores this information in the model variable of type MODEL_ID in the ALLOTTED_QOS table.

2. The router then gets the Resources required list and their corresponding values from the options data field and updates the res_allocated array structure.
3. It then checks with the QoS Routing table, to find out if this reservation is possible. If yes, it updates the new entry in the ALLOTTED_QOS table in the memory or else this entry is removed.
4. If any relevant router en-route is not able to guarantee the requested QoS, an ICMPv6 message is sent to the source and the other routers (that had guaranteed the QoS) are also notified of the same so that they delete the corresponding entry from their QoS tables.

This process happens at all the intermediate routers between the source and the destination.

8. Conclusion

This work has dealt extensively with the design of the Integrated Services model of Quality of Service in IPv6 using the Hop-by-Hop Extensions Header. This is being suggested primarily as a transitional mechanism / solution although it has a definite potential to qualify as an effective QoS support measure.

Acknowledgements

Authors acknowledge technical inputs and support from the members of the “Project IPv6@BITS” at the Birla Institute of Technology and Science, Pilani, India, Dr. Latif Ladid of Ericsson Telebit, (Luxembourg); Dr. Torstern Braun of University of Bern (Switzerland); Dr. Pascal Lorenz of I.U.T. at the University of Haute Alsace, Colmar (France); Dr. S. Rao of Telscom A.G. (Switzerland); Dr. Bernardo Martinez of Versaware Inc. (Spain); Dr. Juan Quemada of UPM, Madrid (Spain) and Dr. Merce and Dr. Paulo at the EC.

References

[RFC 2460] RFC 2460, Internet Protocol version 6 Specification.

[RFC 2117] RFC 2117, Router Alert Option in IPv6.

[Paul et al] QoS in Data Networks, Protocols and Standards by Arindam Paul.

[RFC 2676] RFC 2676, QoS Routing Mechanisms and OSPF Extensions.

[NGNI-MMI-QoS: D1] Rahul Banerjee (BITS), Juan Quemda (UPM), P. Lorenz (UHA), Torsten Braun (UoB), Bernardo Martinez (Versaware): "Use of Various Parameters for Attaining QoS in IPv6-based Multimedia Internetworks", Nov. 15, 2001 available at http://IPv6.bits-pilani.ac.in/ngni/.

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems arising from correct or incorrect implementation or use of this specification.

Author Information

Rahul Banerjee
3256, Centre for Software Development
BITS, Pilani – 333031
Rajasthan, India.

Phone: +91-159-7645073 Ext. 335
Email: rahul@bits-pilani.ac.in