Internet Engineering Task Force Expires: 16 April 2003
Audio/Video Transport Working Group

Timur Friedman, Paris 6
Ramon Caceres, ShieldIP
Kevin Almeroth, UCSB
Kamil Sarac, UCSB
Alan Clark, Telchemy
Robert Cole, AT&T
Kaynam Hedayat, Brix Networks

RTCP Reporting Extensions

draft-ietf-avt-rtcp-report-extns-00.txt

Status of this Memo

Copyright Notice

Abstract

This document defines the XR (extended report) RTCP packet type and eight XR block types. The purpose of the extended reporting format is to convey information that supplements the six statistics that are contained in the report blocks used by SR (sender report) and RR

(receiver report) packets. Some applications, such as MINC (multicast inference of network characteristics) or VoIP (voice over IP) monitoring, require other and more detailed statistics. In addition to the block types defined here, additional block types may be defined in the future by adhereing to the simple framework that this document provides.

1. Introduction

This document defines the XR (extended report) RTCP packet type for RTCP, the control portion of RTP [8]. The definition consists of three parts. First, Section 2 of this document defines a general packet framework capable of including a number of different "extended report blocks." Second, Section 3 defines the general format for such blocks. Third, Section 4 defines a number of such blocks.

The extended report blocks convey information beyond that which is already contained in the reception report blocks of RTCP's SR or RR packets. For example, while a reception report block contains an average loss rate field, an application might opt to use an extended report block that details exactly which packets were received and which were lost. Or, for example, a voice over IP application might require information concerning packets that were discarded from the jitter buffer, in addition to those that were lost.

The framework for these blocks is minimal: only a type field and a length field are required. The purpose is to maintain flexibility and to keep overhead low. While some specific block formats are provided here, others may be defined as the need arises.

1.1 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 RFC 2119 [2] and indicate requirement levels for compliant RTP implementations.

2. XR Packet Format

The XR packet consists of a header of two 32-bit words, followed by a number, possibly zero, of extended report blocks.

This packet format has been implemented as an RTCP APP (application-specific) packet and deployed in the Internet, as described in [3] and [1]. The differences between the APP packet header and the header defined here are that the name field is removed and the subtype field is replaced by a reserved field.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|reserved |   PT=XP=205   |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           SSRC/CSRC                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                         report blocks                         :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

version (V): 2 bits
Identifies the version of RTP. This specification applies to RTP ver¡ sion two (2).

padding (P): 1 bit
If the padding bit is set, this individual RTCP packet contains some additional padding octets at the end that are not part of the control information but are included in the length field. The last octet of the padding is a count of how many padding octets should be ignored, including itself (it will be a multiple of four). A full description of padding in RTCP packets may be found in the RTP specification.

reserved: 5 bits
This field is reserved for future definition. The bits in this field MUST be set to zero unless otherwise defined.

packet type (PT): 8 bits
Contains the constant 205 to identify this as an RTCP XR packet. This is a proposed value, pending assignment of a number by the Internet Assigned Numbers Authority (IANA) [7].

length: 16 bits
The length of this RTCP packet in 32-bit words minus one, including the header and any padding. (The offset of one makes zero a valid length and avoids a possible infinite loop in scanning a compound RTCP packet, while counting 32-bit words avoids a validity check for a multiple of 4.)

SSRC: 32 bits
The synchronization source identifier for the originator of this XR packet.

report blocks: variable length.
Zero or more extended report blocks. The blocks MUST be a multiple

of 32 bits long. They MAY be zero bits long.

3. Extended Report Block Framework

Extended report blocks MUST be stacked, one after the other, at the end of an XR packet. An individual block's length MUST be a multiple of 4 octets. The XR header's length field MUST describe the total length of the packet, including these extended report blocks.

Each block has block type and length fields that facilitate parsing. A receiving application can demultiplex the blocks based upon their type, and can use the length information to locate each successive block, even in the presence of block types it does not recognize.

An extended report block has the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      BT       | type-specific |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                      type-specific data                       :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

block type (BT): 8 bits
   Identifies the specific block format.

type-specific: 8 bits
   The use of these bits is defined by the particular block type.

length: 16 bits
   The length of this report block in 32-bit words minus one, including the header.

type-specific data: variable length
   This MUST be a multiple of 32 bits long. It MAY be zero bits long.

4. Specific Extended Report Blocks

This section defines eight extended report blocks: an experimental block type, and block types for losses, duplicates, packet reception timestamps, detailed reception statistics, receiver timestamps, receiver inter-report delays, and VoIP metrics. An implementation MAY ignore incoming blocks with types either not relevant or unknown to it. Additional block types MAY be registered with the Internet Assigned Numbers Authority (IANA) [7].

4.1 Experimental Block

This type MUST be used for extended report block types that have not been standardized. In addition to the standard type and length fields, it includes a 32 bit name field that serves to distinguish one experimental block type from another.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      BT=0     |  app-specific |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          name (ASCII)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                   application-specific data                   :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

block type (BT): 8 bits
   Block type 0 identifies this as an experimental block.

app-specific: 8 bits
   The use of these bits is defined by the application that uses this block.

length: 16 bits
   The length of this report block in 32-bit words minus one, including the header.

name: 4 octets
   A name chosen by the person definining the experimental block to be unique with respect to other experimental blocks the application might receive.

application-specific data: variable length.
   This MUST be a multiple of 32 bits long. It MAY be zero bits long.

4.2 Loss RLE Block

With this block type, a Boolean trace of lost and received packets can be conveyed in compressed form using run length encoding. This block type has been deployed on the Internet, as part of an RTCP APP (application-specific) packet, as described in [3] and [1].

Caution SHOULD be used in sending such blocks because, even with com¡
pression, they can easily consume bandwidth out of proportion with
normal RTCP packets.

Each block reports on a single source, identified by its SSRC. The
receiver that is supplying the report is identified in the header of
the RTCP packet.

The beginning and ending sequence numbers for the trace are specified
in the block, the ending sequence number being the last sequence num¡
ber in the trace plus one. The last sequence number in the trace MAY
or may not be the sequence number reported on accompanying SR or RR
packets, depending on the needs of the application.

The encoding itself consists of a series of 16 bit chunks. Each
chunk either specifies a run length or a bit vector, or, if the trace
otherwise encodes into an odd number of chunks, MUST be a terminating
null chunk used to round out the block to a 32 bit word boundary.

The mapping from a sequence of lost and received packets into a
sequence of chunks is not unique and is left to the application. A
run length chunk can describe runs of between 1 and 16,383 packet
losses or receipts whereas a bit vector chunk can describe a sequence
of 15 packet losses and receipts. It is RECOMMENDED that the
description of run lengths of 14 or shorter be subsumed into bit vec¡
tor chunks, for purposes of brevity.

A bit vector chunk MAY purport to contain information on packets at
or beyond the ending sequence number. Any such purported information
MUST be ignored.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     BT=17     | rsvd. |   T   |         block length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        SSRC of source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          begin_seq            |             end_seq           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          chunk 1              |             chunk 2           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          chunk n-1            |             chunk n           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

block type (BT): 8 bits
A Loss RLE block is identified by the constant 17 = 0x11.

rsvd.: 4 bits
This field is reserved for future definition. The bits in this field MUST be set to zero unless otherwise defined.

thinning (T): 4 bits
The amount of thinning performed on the sequence space. Only those packets with sequence numbers 0 mod 2^T are reported on by this block. A value of 0 indicates that there is no thinning, and all packets are reported on. The maximum thinning is one packet in every 32,768 (amounting to two packets within each 16-bit sequence space).

length: 16 bits
The length of this report block in 32-bit words minus one, including the header.

begin_seq: 16 bits
The first sequence number that this block reports on.

end_seq: 16 bits
The last sequence number that this block reports on plus one.

chunk i: 16 bits
There are three chunk types: run length, bit vector, and terminating null. If the chunk is all zeroes then it is a terminating null chunk. Otherwise, the leftmost bit of the chunk determines its type: 0 for run length and 1 for bit vector.

4.2.1 Run-Length Chunk

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|R|        run length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

chunk type (C): 1 bit
   A zero identifies this as a runlength chunk.

run type (R): 1 bit
   Zero indicates a run of losses. One indicates a run of received packets.

run length: 14 bits
   A value between 1 and 16,383. The value MUST not be zero (zeroes in both the run type and run length fields would make the chunk a termi¡ nating null chunk). Run lengths of 15 or less MAY be described with a run length chunk despite the fact that they could also be described as part of a bit vector chunk.

4.2.2 Bit Vector Chunk

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C|        bit vector           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

chunk type (C): 1 bit
   A one identifies this as a bit vector chunk.

bit vector: 15 bits
   In the bit vector, as in the run length chunk, a zero indicates a loss and a one indicates a received packet.

4.2.3 Terminating Null Chunk

This chunk is all zeroes.

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.3 Duplicate RLE Block

This block is identical in format to the Loss RLE Block type but car¡ ries information about individual or runs of duplicate packets. A zero indicates the presence of duplicate packets for a given sequence number, whereas a one indicates that no duplicates were received. Note that a packet loss is encoded as a one in this case.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    BT=33      |   reserved    |            length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        SSRC of source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          begin_seq                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          end_seq                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          chunk 1              |             chunk 2           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          chunk n-1            |             chunk n           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

block type (BT): 8 bits
A Duplicate RLE block is identified by the constant 33 = 0x21.

reserved: 8 bits
This field is reserved for future definition All of the bits in this field MUST be set to zero unless otherwise defined.

length: 16 bits
The length of this report block in 32-bit words minus one, including the header.

begin_seq: 32 bits

The first sequence number that this block reports on.

end_seq: 32 bits
The last sequence number that this block reports on plus one.

chunk i: 16 bits
There are three chunk types: run length, bit vector, and terminating null. All zeroes indicates a terminating null. Otherwise, the left¡ most bit of the chunk determines its type: 0 for run length and 1 for bit vector. See the descriptions of these block types in the section on the Loss RLE Block, above, for details.

4.4 Timestamp Report Block

This block carries RTCP-style timestamps for each packet in the range of packet sequence numbers. A similar caution, but more emphatic, is made for timestamp report blocks as was made for Loss RLE Block pack¡ ets. For each packet in the sequence number range, a 32 bit value MUST be recorded and sent. This could easily consume significant bandwidth. Care SHOULD be taken in the size of the sequence space over which to monitor timestamps.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     BT=48     |   reserved    |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        SSRC of source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          begin_seq                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          end_seq                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     RTP timestamp (pkt n)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

block type (BT): 8 bits
A Timestamp block is identified by the constant 48 = 0x30.

reserved: 8 bits
This field is reserved for future definition. All bits in this field MUST be set to zero unless otherwise defined.

length: 16 bits
The length of this report block in 32-bit words minus one, including the header.

begin_seq: 32 bits
The first sequence number that this block reports on.

end_seq: 32 bits
The last sequence number that this block reports on plus one.

RTP timestamp: 32 bits
Corresponds to the same units as the RTP timestamp in RTP data pack¡ ets. The timestamp is established upon packet arrival. It can be used to measure partial path characteristics and to model distribu¡ tions for packet jitter.

4.5 Statistics Summary Block

This block reports detailed statistics above and beyond the informa¡ tion carried in the standard RTCP packet format. Information is recorded about lost packets, duplicate packets, jitter measurements, and TTL values. The packet contents are dependent upon a bit vector carried in the first part of the header. Not all values need to be carried in each packet. Header fields for values not carried are not included in the packet.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     BT=1      |L|D|J|T|resvd. |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        SSRC of source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          begin_seq                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          end_seq                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        lost_packets                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        dup_packets                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         min_jitter                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         max_jitter                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         avg_jitter                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         dev_jitter                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   min_ttl     |   max_ttl     |   avg_ttl     |     dev_ttl   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

block type (BT): 8 bits
   A Statistics Summary block is identified by the constant 1 = 0x01.

content bits (L,D,J,T): 4 bits
   Bit set to 1 if packet contains (L)oss, (D)uplicate, (J)itter, and/or (T)TL report.

resvd.: 4 bits
   This field is reserved for future definition. All bits in this field MUST be set to zero unless otherwise defined.

length: 16 bits
   The length of this report block in 32-bit words minus one, including the header.

begin_seq: 32 bits
   The first sequence number that this block reports on.

end_seq: 32 bits
   The last sequence number that this block reports on plus one.

lost_packets: 32 bits
Number of lost packets in the above sequence number interval.

dup_packets: 32 bits
Number of duplicate packets in the above sequence number interval.

min_jitter: 32 bits
The minimum relative transit time between two packets in the above sequence number interval. All jitter values are measured as the difference between a packet’s RTP timestamp and the reporter’s clock at the time of arrival, measured in the same units.

max_jitter: 32 bits
The maximum relative transit time between two packets in the above sequence number interval.

avg_jitter: 32 bits
The average relative transit time between each two packet series in the above sequence number interval.

dev_jitter: 32 bits
The standard deviation of the relative transit time between each two packet series in the above sequence number interval.

min_ttl: 8 bits
The minimum TTL value of data packets in sequence number range.

max_ttl: 8 bits
The maximum TTL value of data packets in sequence number range.

avg_ttl: 8 bits
The average TTL value of data packets in sequence number range.

dev_ttl: 8 bits
The standard deviation of TTL values of data packets in sequence number range.

4.6 Receiver Timestamp Report Block

This block extends RTCP’s timestamp reporting so that non-senders may also send timestamps. It recapitulates the NTP timestamp fields from the RTCP Sender Report [7, Sec. 6.3.1]. A non-sender may estimate its RTT to other participants, as proposed in [9], by sending this report block and receiving DLRR report blocks (see next section) in reply.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     BT=2      |                   reserved                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              NTP timestamp, most significant word             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             NTP timestamp, least significant word             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

block type (BT): 8 bits
A Receiver Timestamp block is identified by the constant 2 = 0x02.

reserved: 24 bits
This field is reserved for future definition. The bits in this field MUST be set to zero unless otherwise defined.

NTP timestamp: 64 bits
Indicates the wallclock time when this block was sent so that it may be used in combination with timestamps returned in DLRR report blocks from other receivers to measure round-trip propagation to those receivers. Receivers should expect that the measurement accuracy of the timestamp may be limited to far less than the resolution of the NTP timestamp. The measurement uncertainty of the timestamp is not indicated as it may not be known. A report block sender that can keep track of elapsed time but has no notion of wallclock time may use the elapsed time since joining the session instead. This is assumed to be less than 68 years, so the high bit will be zero. It is permissible to use the sampling clock to estimate elapsed wallclock time. A report sender that has no notion of wallclock or elapsed time may set the NTP timestamp to zero.

4.7 DLRR Report Block

This block extends RTCP’s DLSR mechanism [7, Sec. 6.3.1] so that non-senders may also calculate round trip times, as proposed in [9]. It is termed DLRR for Delay since Last Receiver Report, and may be sent in response to a Receiver Timestamp report block (see previous sec¡ tion) from a receiver to allow that receiver to calculate its round trip time to the respondant. The report consists of one or more 3 word sub-blocks: one sub-block per receiver report.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     BT=3      |   reserved    |             length            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_1 (SSRC of first receiver)               | sub-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
|                         last RR (LRR)                         |   1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   delay since last RR (DLRR)                  |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                 SSRC_2 (SSRC of second receiver)              | sub-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ block
:                               ...                             :   2
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

block type (BT): 8 bits
A DLRR block is identified by the constant 3 = 0x03.

reserved: 8 bits
This field is reserved for future definition. All bits in this field MUST be set to zero unless otherwise defined.

length: 16 bits
The length of this report block in 32-bit words minus one, including the header. The number of sub-blocks is length divided by three (3).

last RR timestamp (LRR): 32 bits
The middle 32 bits out of 64 in the NTP timestamp (as explained in the previous section) received as part of a Receiver Timestamp report block from participant SSRC_n. If no such block has been received, the field is set to zero.

delay since last RR (DLRR): 32 bits
The delay, expressed in units of 1/65536 seconds, between receiving the last Receiver Timestamp report block from participant SSRC_n and sending this DLRR report block. If no Receiver Timestamp report block has been received yet from SSRC_n, the DLRR field is set to zero (or the DLRR is omitted entirely). Let SSRC_r denote the receiver issuing this DLRR report block. Participant SSRC_n can com¡ pute the round-trip propagation delay to SSRC_r by recording the time A when this Receiver Timestamp report block is received. It calcu¡ lates the total round-trip time A-LSR using the last SR timestamp (LSR) field, and then subtracting this field to leave the round-trip propagation delay as (A- LSR - DLSR). This is illustrated in [7, Fig. 2].

4.8 VoIP Metrics Report Block

4.8.1 Summary

The VoIP Metrics report block provides metrics for monitoring voice over IP (VoIP) calls. These metrics include packet loss and discard metrics, delay metrics, analog metrics, and voice quality metrics. The block reports separately on packets lost on the IP channel, and those that have been received but then discarded by the receiving jitter buffer. It also reports on the combined effect of losses and discards, as both have equal effect on call quality.

In order to properly assess the quality of a Voice over IP call it is desirable to consider the degree of burstiness of packet loss [4]. Following a Gilbert-Elliott model [5], an interval, bounded by lost and/or discarded packets, with a high rate of losses and/or discards is a "burst," and an interval between two bursts is a "gap." Bursts correspond to intervals of time during which the packet loss rate is high enough to produce noticeable degradation in audio quality. Gaps correspond to periods of time during which only isolated lost packets may occur, and in general these can be masked by packet loss con¡ cealment. Delay reports include the transit delay between RTCP end points and the VoIP end system processing delays, both of which con¡ tribute to the user perceived delay. Additional metrics include sig¡ nal, echo, noise, and distortion levels. Call quality metrics include R factors (E Model) [5] and MOS scores (Mean Opinion Scores).

An implementation that sends these blocks SHOULD send at least one every ten seconds for the duration of a call, and SHOULD send one upon call termination. An implementation MUST supply values for all fields defined here.

4.8.2 VoIP Metrics block structure

The block is encoded as seven 32-bit words:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     BT=64     |   reserved    |          length=7             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   loss rate   | discard rate  |       burst duration          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| burst density |         gap duration          | gap density   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     round trip delay          |       end system delay        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| signal power  | doubletalk    | noise level   |       Gmin    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   R factor    | ext. R factor |    MOS-LQ     |    MOS-CQ     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   RX Config   | JB Nominal    |  JB Maximum   | JB Abs Max    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

block type (BT): 8 bits
A VoIP Metrics block is identified by the constant 64 = 0x40.

reserved: 8 bits
This field is reserved for future definition. All bits in this field MUST be set to zero unless otherwise defined.

length: 16 bits
The length of this report block in 32-bit words minus one, including the header. This is the constant 6 = 0x06.

4.8.3 Packet loss and discard metrics

It is very useful to distinguish between packets lost by the network and those discarded due to jitter. Both have equal effect on the quality of the voice stream however having separate counts is very useful when trying to identify the source of quality degradation. These fields MUST be populated.

loss rate: 8 bits
The fraction of RTP data packets from the source lost since the beginning of reception, expressed as a fixed point number with the binary point at the left edge of the field. This value is calculated by dividing the total number of packets lost (after the effects of applying any error protection such as FEC) by the total number of packets expected, multiplying the result of the division by 256, and taking the integer part. The numbers of duplicated packets and dis¡ carded packets do not enter into this calculation. Since receivers cannot be required to maintain unlimited buffers, a receiver MAY

categorize late-arriving packets as lost. The degree of lateness that triggers a loss SHOULD be significantly greater than that which triggers a discard.

discard rate: 8 bits
The fraction of RTP data packets from the source that have been dis carded since the beginning of reception, due to late or early arrival, under-run or overflow at the receiving jitter buffer. This value is expressed as a fixed point number with the binary point at the left edge of the field. It is calculated by dividing the total number of packets discarded (excluding duplicate packet discards) by the total number of packets expected, multiplying the result of the division by 256, and taking the integer part.

burst metrics:
A burst is defined as a longest sequence of packets bounded by lost or discarded packets with the constraint that within a burst the num¡ ber of successive packets that were received, and not discarded due to delay variation, is less than some value Gmin. A gap is defined as the interval between bursts, and has the property that any lost or discarded packets must be preceded and followed by at least Gmin packets that were received and not discarded. This gives a maximum loss/discard density within a gap of 1 / (Gmin + 1).

burst duration: 16 bits
The mean duration, expressed in milliseconds, of the burst intervals that have occurred since the beginning of reception. The duration of each interval is calculated based upon the packets that mark the beginning and end of that interval. It is equal to the timestamp of the end packet, plus the duration of the end packet, minus the times tamp of the beginning packet. If the actual values are not avail able, estimated values MUST be used. If there have been no burst intervals, the burst duration value MUST be zero.

burst density: 8 bits
The fraction of RTP data packets within burst intervals since the beginning of reception that were either lost or discarded. This value is expressed as a fixed point number with the binary point at the left edge of the field. It is calculated by dividing the total number of packets lost or discarded (excluding duplicate packet dis¡ cards) within burst intervals by the total number of packets expected within the burst intervals, multiplying the result of the division by 256, and taking the integer part.

gap duration: 16 bits
The mean duration, expressed in milliseconds, of the gap intervals that have occurred since the beginning of reception. The duration of each interval is calculated based upon the packet that marks the end

of the prior burst and the packet that marks the beginning of the subsequent burst. It is equal to the timestamp of the subsequent burst packet, minus the timestamp of the prior burst packet, plus the duration of the prior burst packet. If the actual values are not available, estimated values MUST be used. In the case of a gap that occurs at the beginning of reception, the sum of the timestamp of the prior burst packet and the duration of the prior burst packet are replaced by the reception start time. In the case of a gap that occurs at the end of reception, the timestamp of the subsequent burst packet is replaced by the reception end time. If there have been no gap intervals, the gap duration value MUST be zero.

gap density: 8 bits
The fraction of RTP data packets within inter-burst gaps since the beginning of reception that were either lost or discarded. The value is expressed as a fixed point number with the binary point at the left edge of the field. It is calculated by dividing the total num¡ ber of packets lost or discarded (excluding duplicate packet dis cards) within gap intervals by the total number of packets expected within the gap intervals, multiplying the result of the division by 256, and taking the integer part.

For example, if the packet spacing is 10mS and a 1 denotes a received packet and 0, a lost, and X, a discarded, packet then the following pattern:

```
  11110111111111111111111X111X1011110111111111111111111X111111111
                         |--burst---|
```

would have a burst duration of 120mS, a burst density of 0.33, a gap duration of 510mS and a gap density of 0.04, for a GMIN value of 4 or larger.

4.8.4 Delay metrics

For the purpose of the following definitions, the RTP interface is the interface between the RTP instance and the voice application (i.e. FEC/de-interleaving/ de-multiplexing, jitter buffer). For example, the time delay due to RTP payload multiplexing would be con¡ sidered to be part of the voice application or end-system delay whereas delay due to multiplexing RTP frames within a UDP frame would be considered part of the RTP reported delay. This distinction is consistent with the use of RTCP for delay measurements.

round trip delay: 16 bits
The most recently calculated round trip time between RTP interfaces,

expressed in milliseconds. This value is the time of receipt of the most recent RTCP packet from source SSRC, minus the LSR (last SR) time reported in its SR (sender report), minus the DLSR (delay since last SR) reported in its SR. A non-zero LSR value is REQUIRED in order to calculate round trip delay. A value of 0 is permissible dur¡ ing the first 2-3 RTCP exchanges as insufficient data may be avail¡ able to determine delay however MUST be populated as soon as a delay estimate is available.

end system delay: 16 bits
The most recently estimated end system delay, expressed in millisec¡ onds. End system delay is defined as the total encoding, decoding and jitter buffer delay determined at the reporting endpoint. This is the time required for an RTP frame to be buffered, decoded, con¡ verted to analog form, looped back at the local analog interface, encoded, and made available for transmission as an RTP frame. The manner in which this value is estimated is implementation dependent. This parameter MUST be provided in all VoIP metrics reports.

Note that the one way symmetric VoIP segment delay may be calculated from the round trip and end system delays as follows. If the round trip delay is denoted RTD and the end system delays associated with the two endpoints are ESD(A) and ESD(B) then:

one way symmetric voice path delay = (RTD + ESD(A) + ESD(B)) / 2

4.8.5 Signal related metrics

The following metrics are intended to provide real time information related to the non-packet elements of the voice over IP system to assist with the identification of problems affecting call quality. The values identified below must be determined for the received audio signal. The information required to populate these fields may not be available in all systems, although it is strongly recommended that this data SHOULD be provided to support problem diagnosis.

signal level: 8 bits
The voice signal relative level is defined as the ratio of the signal level to overflow signal level, expressed in decibels as a signed integer in two’s complement form. This is measured only for packets containing speech energy. The intent of this metric is not to pro¡ vide a precise measurement of the signal level but to provide a real time indication that the signal level may be excessively high or low. If the full range (overflow level) of the Vocoder’s Digital to Analog conversion function is +/- L and the value of a decoded sample during a talkspurt is V then the signal level is given by

Signal level = 10 log10 (mean(abs(V) / L))

A value of 127 indicates that this parameter is unavailable.

doubletalk level: 8 bits
The doubletalk level is defined as the proportion of voice frame intervals during which speech energy was present in both sending and receiving directions. High levels of doubletalk can provide an indi¡ cation of delay or echo related problems. The value is expressed as a fixed point number with the binary point at the left edge of the field. It is calculated by dividing the total number of voice frame intervals by the number of voice frame intervals during which energy was present in both sending and receiving directions, multiplying the result of the division by 256, and taking the integer part.

A value of 255 indicates that this value is unavailable

noise level: 8 bits
The noise level is defined as the ratio of the silent period back ground noise level to overflow signal power, expressed in decibels as a signed integer in two’s complement form. If the full range (over¡ flow level) of the Vocoder’s Digital to Analog conversion function is +/- L and the value of a decoded sample during a silence period is V then the noise level is given by

Noise level = 10 log10 (mean(abs(V) / L))

A value of 127 indicates that this parameter is unavailable.

4.8.6 Call quality/ transmission quality metrics

The following metrics are direct measures of the transmission quality or call quality, and incorporate the effects of CODEC type, packet loss, discard, burstiness, delay etc. These metrics may not be available in all systems however SHOULD be provided in order to sup¡ port problem diagnosis.

R factor: 8 bits
The R factor is a voice quality metric describing the segment of the call that is carried over this RTP session. It is expressed as an integer in the range 0 to 100, with a value of 94 corresponding to "toll quality" and values of 50 or less regarded as unusable. This metric is defined as including the effects of delay, consistent with ITU-T G.107 [6] and ETSI TS 101 329-5 [5].

A value of 127 indicates that this parameter is unavailable.

ext. R factor: 8 bits
The external R factor is a voice quality metric describing the seg ment of the call that is carried over a network segment external to the RTP segment, for example a cellular network. Its values are interpreted in the same manner as for the RTP R factor. This metric is defined as including the effects of delay, consistent with ITU-T G.107 [6] and ETSI TS 101 329-5 [5], and relates to the outward voice path from the Voice over IP termination for which this metrics block applies.

Note that an overall R factor may be estimated from the RTP segment R factor and the external R factor, as follows:

R total = RTP R factor + ext. R factor - 94

A value of 127 indicates that this parameter is unavailable.

MOS-LQ: 8 bits
The estimated mean opinion score for listening quality (MOS-LQ) is a voice quality metric on a scale from 1 to 5, in which 5 represents excellent and 1 represents unacceptable. This metric is defined as not including the effects of delay and can be compared to MOS scores obtained from listening quality (ACR) tests. It is expressed as an integer in the range 10 to 50, corresponding to MOS x 10. For exam¡ ple, a value of 35 would correspond to an estimated MOS score of 3.5.

A value of 127 indicates that this parameter is unavailable.

MOS-CQ: 8 bits
The estimated mean opinion score for conversational quality (MOS-CQ) is defined as including the effects of delay and other effects that would affect conversational quality. The metric may be calculated by converting an R factor determined according to ITU-T G.107 [6] or ETSI TS 101 329-5 [5] into an estimated MOS using the equation speci¡ fied in G.107

A value of 127 indicates that this parameter is unavailable.

4.8.7 Configuration parameters:

Gmin: 8 bits
The gap threshold. This field contains the value used for this report block to determine if a gap exists. The recommended value of 16 = 0x10 corresponds to a burst interval having a minimum density of

6.25% of lost or discarded packets, which may cause noticeable degra¡ dation in call quality; during gap intervals, if packet loss or dis card occurs, each lost or discarded packet would be preceded by and followed by a sequence of at least 16 received non-discarded packets. Note that lost or discarded packets that occur within Gmin packets of a report being generated may be reclassified as being part of a burst or gap in later reports. ETSI TS 101 329-5 [5] defines a computa¡ tionally efficient algorithm for measuring burst and gap density using a packet loss/discard event driven approach. Gmin MUST not be zero and MUST be provided.

Receiver Configuration byte:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|PLC|JBA|JB rate|
+-+-+-+-+-+-+-+-+
```

PLC - packet loss concealment

Standard (11) / enhanced (10) / disabled (01) / unspecified (00). When PLC=11 then a simple replay or interpolation algorithm is being used to fill-in the missing packet - this is typically able to con¡ ceal isolated lost packets with loss rates under 3%. When PLC=10 then an enhanced interpolation algorithm is being used - this would typically be able to conceal lost packets for loss rates of 10% or more. When PLC=01 then silence is inserted in place of lost packets. When PLC = 00 then no information is available concerning the use of PLC however for some CODECs this may be inferred.

JBA - Jitter Buffer Adaptive

Adaptive (11) / non-adaptive (10) / reserved (01)/ unknown (00). When Jitter Buffer is adaptive then its size is being dynamically adjusted to deal with varying levels of jitter. When non-adaptive then the Jitter Buffer size is maintained at a fixed level. When either adap¡ tive or non-adaptive modes are specified then the Jitter Buffer Size parameters below MUST be specified.

JB Rate - Jitter Buffer Rate

J = adjustment rate (0-15). This represents the implementation spe¡ cific adjustment rate of a Jitter Buffer in adaptive mode. This parameter is defined in terms of the approximate time taken to fully adjust to a step change in peak to peak jitter from 30mS to 100mS such that:

adjustment time = 2* J * frame size (mS)

This parameter is intended only to provide a guide to the degree of "aggressiveness" of a an adaptive jitter buffer and may be estimated. A value of 0 indicates that the adjustment time is unknown for this implementation.

4.8.7 Jitter Buffer Parameters

Jitter Buffer - nominal size in frames (8 bit)
This is the current nominal fill point within the jitter buffer, which corresponds to the nominal jitter buffer delay for packets that arrive exactly on time. This parameter MUST be provided for both fixed and adaptive jitter buffer implementations.

Jitter Buffer Maximum - size in frames (8 bit)
This is the current maximum jitter buffer level corresponding to the earliest arriving packet that would not be discarded. In simple queue implementations this may correspond to the nominal size. In adaptive jitter buffer implementations this value may dynamically vary up to Jitter Buffer Absolute Maximum. This parameter MUST be provided for both fixed and adaptive jitter buffer implementations.

Jitter Buffer Absolute Maximum - size in frames (8 bit)
This is the absolute maximum size that the adaptive jitter buffer can reach under worst case jitter conditions. This parameter MUST be provided for adaptive jitter buffer implementations and its value MUST be set to JB Maximum for fixed jitter buffer implementations.

Example of burst packet loss calculation.

This is an event driven algorithm for measuring burst characteristics and is hence extremely computationally efficient.

Given the following definition of states:

State 1 = received a packet during a gap
State 2 = received a packet during a burst
State 3 = lost a packet during a burst
State 4 = lost an isolated packet during a gap

The "c" variables below correspond to state transition counts, i.e. c14 is the transition from state 1 to state 4. It is possible to infer one of a pair of state transition counts to an accuracy of 1 which is generally sufficient for this application. "pkt" is the count of packets received since the last packet was declared lost or discarded and "lost" is the number of packets lost within the current burst.

```
if ( packet_lost ) loss_count++;
if ( packet_discarded ) discard_count++;
if (pkt >= gmin)
{
    if (lost == 1)
       c14++;
    else
        c13++;
    lost = 1;
    c11 += pkt;
}
else
{
    lost++;
    if (pkt == 0)
        c33++;
    else
    {
        c23++;
        c22 += (pkt - 1);
    }
}
```

At each reporting interval the burst and gap metrics can be calcu¡ lated as follows.

```
/* calculate additional transition counts */
c31 = c13;
c32 = c23;
ctotal = c11 + c14 + c13 + c22 + c23 + c31 + c32 + c33;

/* calculate burst and densities */
p32 = c32 / (c31 + c32 + c33);
if ((c22 + c23) < 1)
    p23 = 1;
else
    p23 = 1 - c22/(c22 + c23);
burst_density = 256 * p23 / (p23 + p32);
gap_density = 256 * c14 / (c11 + c14);

/* calculate burst and gap durations in mS */
m = frameDuration_in_mS * framesPerRTPPkt;
gap_length = (c11 + c14 + c13) * m / c13;
burst_length = ctotal * m / c13 - lgap;

/* calculate loss and discard densities */
loss_density = 256 * loss_count / ctotal;
discard_density = 256 * discard_count / ctotal;
```

5. Acknowledgements

We thank the following people: Colin Perkins, Steve Casner, and Hen¡ ning Schulzrinne for their considered guidance; Nick Duffield for extensive ongoing contributions; Sue Moon for helping foster collabo¡ ration between the authors of this document; and Mounir Benzaid for drawing our attention to the reporting needs of MLDA.

6. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to per¡ tain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP 11 [7]. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such propri¡ etary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

7. References

[1] A. Adams, T. Bu, R. Caceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, and D. Towsley, "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior," IEEE Communications Magazine, May 2000.

[2] S. Bradner, "Key words for use in RFCs to indicate requirement levels," BCP 14, RFC 2119, IETF, March 1997.

[3] R. Caceres, N.G. Duffield, and T. Friedman, "Impromptu measure¡ ment infrastructures using RTP," Proc. IEEE Infocom 2002.

[4] A. D. Clark, "Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality," Proc. IP Telephony Workshop 2001.

[5] ETSI, "Quality of Service (QoS) measurement methodologies," ETSI TS 101 329-5 V1.1.1 (2000-11), November 2000.

[6] ITU-T, "The E-Model, a computational model for use in transmis¡ sion planning," Recommendation G.107 (05/00), May 2000.

[7] J. Reynolds and J. Postel, "Assigned Numbers," STD 2, RFC 1700, IETF, October 1994.

[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 1889, IETF, February 1996.

[9] D. Sisalem and A. Wolisz, "MLDA: A TCP-friendly Congestion Con¡ trol Framework for Heterogeneous Multicast Environments", Proc. IWQoS 2000.

8. Full Copyright Statement

9. Authors’ Addresses

Timur Friedman <timur.friedman@lip6.fr>
University of Paris 6
Laboratoire LiP6-CNRS
8, rue du Capitaine Scott
75015 PARIS, FRANCE

Ramon Caceres <ramon@shieldip.com>
ShieldIP, Inc.
11 West 42nd Street, 31st Floor
New York, NY 10036, USA

Kevin Almeroth <almeroth@cs.ucsb.edu>
Department of Computer Science
University of California
Santa Barbara, CA 93106, USA

Kamil Sarac <ksarac@cs.uscb.edu>
Department of Computer Science
University of California
Santa Barbara, CA 93106, USA

Alan Clark <alan@telchemy.com>
Telchemy Incorporated
3360 Martins Farm Road, Suite 200
Suwanee, GA 30024
Tel: +1 770 614-6944
Fax: +1 770 614-3951

Robert Cole <rgcole@att.com>
AT&T Labs
330 Saint Johns Street,
2nd Floor
Havre de Grace, MD, USA 21078
Tel: +1 410 939-8732
Fax: +1 410 939-8732

Kaynam Hedayat <khedayat@brixnet.com>
Brix Networks
285 Mill Road
Chelmsford, MA 01824
Tel: +1 978 367-5600
Fax: +1 978 367-5700

10. Expiry

This draft expires 18 January 2003.