Draft Proposal for VSI-E - Rev 2.0

MIT Haystack Observatory

July 2003

Table of Contents

List of Figures

List of Tables

1. Introduction

VSI-H[1] and VSI-S[2] define the electrical interfaces to and from a VLBI Data Transmission System (DTS), but explicitly refrain from specifying any details concerning the format of data traveling from the Data Input Module (DIM) to the Data Output Module (DOM) output. This philosophy is appropriate for traditional VLBI DTS's where media are shipped from station to correlator and where station and correlator are assumed to have compatible DIMs and DOMs. However, the recent upsurge in e-VLBI activity highlights the necessity to move a step further to define a standard e-VLBI (or media independent) data format that is transmitted from DIM to DOM and that is compatible between otherwise heterogeneous DTS's. The VSI-E specification is designed to meet this need.

2. Assumptions

The following assumptions are made in the development of the VSI-E specification:

- The DTS is compliant with the VSI-H specification
- All active bit streams and and associated relevant parameters must be derivable from the information arriving at the DOM, in particular:
    - Primary data stream (i.e. active bit-stream data)
    - Active bit-stream mask
    - DOT time-tagging
    - Bit-stream information rate (BSIR)
    - Valid-data indicator
    - TVG-data indicator
    - PDATA messages
- Underlying network structure is IP-based

3. Goals

The aim of this specification is to provide a framework that allows the transport of VLBI data across networks, with the following goals:

- Efficient transport mechanism
- Standard protocols
- Internet-friendly transport
- Scalable Implementation
- Ability to transport individual data-channel streams as individual packet streams
- Ability to make use of multicasting to transport data and/or control information in an efficient manner (could be used in the future for support of distributed correlation).

4. Data Formatting

4.1 Definitions

The following definitions are made:

1. A ***DIM sample*** is defined as a subset of 2^n active bit streams collected on a single DIM clock.
2. A ***packet sample*** is defined as a subset of 2^m bits ($m<n$) within a DIM sample that is selected to be transmitted in a single RTP packet stream..
3. A ***channel sample*** is defined as a subset of 2^p bits within a packet sample that corresponds to the sample bits of a single analog voltage waveform.
4. A ***DIM stream*** is defined as a contiguous set of DIM samples collected over some period of time.
5. A ***packet stream*** is defined as a contiguous set of packet samples collected over some period of time.
6. A ***channel stream*** is defined as a contiguous set of channel samples collected over some period of time.

4.2 Packetization by User Data Channel

The 32 VSI bit-streams into a DIM can normally be segregated to represent one or more user data (or signal) channels. We propose that VSI-E allow the segregation of these data channels into separate packet streams, where each packet stream may contain one or more channel streams. In normal usage, we would expect one channel stream to be mapped to one packet stream. This has several powerful advantages:

- ***Flexibility:*** Each packet stream may be treated independently, with each stream potentially targeted to different destinations or multiple destinations (multi-cast). The end-to-end network bandwidth requirements to any given destination is dependent only on the number of packet streams being sent to that destination.
- ***Scalability:*** Specified packet streams may be dropped, as necessary, to reduce the aggregate data rate to meet available network bandwidths. For example, if 16 data channels aggregate to 1024 Mbps, all 16 channels will not fit onto a single Gigabit Ethernet connection. However, if two channels are dropped, the rate drops to 896 Mbps, which will fit comfortably. The user is not required to drop back to 512 Mbps, as he/she would be if data from all channels were aggregated into every packet[1].
- ***Efficiency in a lossy data channel:*** In a packet stream per channel scheme, the loss of a single data packet causes only the loss of one data channel for a short period of time, has no effect on other channels, and does not force the invalidation of other received packets.

[1] The VSI-H interface specification allows only data rates of 2^n Mbps to be selected by the DIM and DOM. We propose that the VSI-H not be affected by the VSI-E specification, but rather work within the VSI-H framework.

The flexibility to map multiple data streams to a single channel stream also provides backwards compatibility with existing VSI-compliant equipment.

4.3 Channel Packetization

The channel samples from each channel are collected together into packet samples for transmission. The following rules specify the formation of data packets from these packet samples:

1. Packet streams are divided into a sequential set of data packets, each containing a fixed-length ***data payload*** which consists of sequential packet samples.
2. The data-payload array contains an integer number of 32-bit words, each of which contains one or more packet samples containing one or more channel samples.
3. All data packets have a common data-payload length.

4.4 Packet Data Array Format

Each packet data array is made up of an integral number of 32-bit words, each of which contains one or more sequential packet samples containing one ore more channel samples. The format of each 32-bit word is defined for 1, 2, 4, 8, 16 and 32 bits/channel sample, as shown in Figure 1. A 64-bit packet time stamp (which is effectively a sequence number) in each data packet serves to time-stamp that packet (refer to §10.3 for details of timestamp calculation and synchronization).

4.5 Endian

Because the vast majority of the platforms on which this data will be manipulated are Intel-derived platforms, all data is transmitted in little endian format (least significant bytes are transmitted first).

5. Why RTP for e-VLBI?

The Real-time Transport Protocol (RTP) is a protocol designed to transport real-time data across the Internet. It is an IETF[2] standard that provides an Internet standard way for transporting sampled analog data across the Internet and fits well with the requirements of e-VLBI. RTP provides the following facilities:

- Transmission of sampled analog data
- Dissemination of session information
- Monitoring of network and end system performance (by participants and third parties)

[2] The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

- Adaptation to varying network capability / performance
- Appropriate reliability / repair model
- Message Sequencing / un-reordering
- Multi-cast distribution of statistics, control and data

RTP has been designed for the transport of real-time data such as e-VLBI and has been used for many years for transporting real-time data across the Internet. There is a wealth of RTP experience and tools that can be re-used [3, 4]. Importantly, the networking community views RTP as "Internet Friendly". It has been designed with attention to efficiency, resource constraints and scaling issues.

Figure 2 shows a high level view of an RTP-based e-VLBI system. Data from the telescope Data Acquisition System is fed into a DIM via a VSI-H interface. The data is then formatted and transported over a network using RTP. RTP operates in tandem with its companion signaling protocol the Real-time Transport Control Protocol (RTCP). RTP is used to transport the sampled data streams, associated timing information and some information about the structure and validity of the data, while RTCP is used as a signaling channel to transport static or slowly time-varying information (such as PDATA, active bit stream masks and so on). The RTP channel uses the bulk of the network resources (95% or more of the network bandwidth dedicated to the e-VLBI connections), while the RTCP channel uses up to 5% for purposes such as:

- Distribution of transmission and reception statistics
- Distribution of information about sources
- Simple session management

6. Network Topologies

Figure 3 shows several possible network topologies that could be used in the transport of VLBI data from telescope sites to a correlation site(s). The network in Figure 3a is a scenario that is typical today, in which each telescope site transports its data channels to a single DOM at a single correlation site.

Figure 3b shows a configuration that may be used in the near future. In this configuration, a single data channel from each telescope is transported to the DOMs at a correlation site to allow real-time monitoring/correlation of the data stream. This would be useful for detecting fringes in real-time and would help to diagnose errors in experimental setup ***during*** an experiment.

The network shown in Figure 3c shows a more general case that may be applicable in the future when distributed correlation becomes practical. In this example, the channels from the different telescope sites are sent to multiple correlation sites where they are correlated in a distributed fashion.

7. RTP

The data from each channel stream are incorporated into a single RTP data stream. The Real-time Transport Protocol (RTP) is used to:

- encapsulate user data into a format that preserves timing and synchronization information

- identify the payload and source of the payload
- provide sequence number protection for in-order delivery
- provide a number of more advanced features such as merging, bridging and translation support[3].

RTP was designed with extensibility in mind. It provides a framework that can be customized to suit individual application's needs. An application specific ***RTP profile*** provides all of the information necessary to encode/transport/decode etc. an application's data using RTP across a wide area network. For an example of an RTP profile, refer to [5]. Wroclawski et. al[6] have developed a draft RTP profile for e-VLBI which will be discussed in §10.

8. RTCP

RTCP (RTP Control Protocol) operates as a low-bandwidth control and monitor channel in conjunction with the primary RTP data stream to:

- provide mechanisms for reliably monitoring a network's real-time data delivery performance
- collect statistics from participants and distribute these statistics to other participants and network management elements.
- adapt transmission strategies to prevailing network conditions.

For e-VLBI, each host will use a single RTCP stream to communicate with every other host within the RTCP session. However, the RTP standard also allows RTCP packets to be associated with multiple hosts.

9. Underlying Network Transport

In the RTP specification, the choice of transport level protocol is left up to the implementer. Some options include:

- User Datagram Protocol (UDP) [7]
- Transmission Control Protocol (TCP) [8]
- Datagram Congestion Control Protocol (DCCP) [9]
- User Datagram Protocol with Application Level flow control[4]

Whichever transport protocol is used, it is recommended that hosts implement an appropriate "network friendly" congestion control mechanism, for example:

- '***Less-than-Best-Effort' service:*** This is a service offered by some networks where packets are prioritized according to a special code in the packet header. Packets with the 'less-than-best-effort' code are dropped in favor of

[3] While some of these features may not be immediately applicable for e-VLBI, we do not preclude their use as they may prove useful in the future (for example, in the support of distributed correlation).

[4] UDP with Application Flow Control allows a great deal of implementation flexibility, including enabling the flow control mechanism to access the quality of service statistics distributed by RTP.

packets with the 'best effort' code during periods of network congestion. By marking packets with the 'less-than-best-effort' code, end systems minimize the impact their data has on regular users' 'best effort' data. For further information, see [10].

- ***Rate limiting of UDP transmission rate:*** UDP is a lightweight datagram protocol that does not implement congestion control. Uncontrolled high-bandwidth UDP flows can have a large negative impact on other network users' traffic. By rate-limiting UDP flows, it is possible to minimize the impact of UDP traffic on other users. For more information on this technique, refer to [11].

The use of TCP for the reliable end-to-end transport of RTCP control information is also recommended.

10. An RTP Profile for e-VLBI

The RTP profile defines the structure and semantics of the RTP packets that will be used to transport VLBI data. There are six packet types:

- ***RTP Data Packet:*** used to encapsulate and transport e-VLBI data. Contains payload type (number of bits per channel sample), sequence numbers, RTP timestamp[5], Experiment timestamp[6], source identifier and data samples.
- ***RTCP Sender Report Packet:*** used to allow sources (telescopes) to distribute transmission statistics and determine the relationship between UT and Experiment timestamp. Statistics include the sender's packet and octet count. Also included is an Experiment timestamp and a corresponding 32-bit UT Second and 64 bit Sample Offset.
- ***RTCP Receiver Report Packet:*** used by receivers (correlation sites) to distribute quality of reception statistics. Statistics include fractional packet losses, cumulative number of packets lost, inter-arrival jitter etc. which sources can use to adjust their transmission rates appropriately. Also provides a mechanism for sources to calculate round trip times.
- ***RTCP Source DEScription Packet:*** used by sources to distribute information about themselves to other session participants (in this case the

[5] The RTP timestamp is a 32-bit counter that keeps track of how many data samples have been transmitted since an arbitrary origin. The timestamp is initialized to a random number for security purposes. In this specification, the RTP timestamp is used for maintaining coarse grain timing synchronization between the DIM and the DOM for the purposes of collecting network statistics etc. For maintaining timing synchronization between the DAS and the DPS, we introduce a new timestamp: the Experiment Timestamp.

[6] The Experiment timestamp is a 64-bit counter located in the payload of a VLBI RTP data packet, immediately in front of the data samples. The Experiment timestamp is incremented by one for each data packet created. It is used to maintain timing synchronization between the DAS and the DPS in both real-time and non-real time modes and is independent of the RTP timestamp.

correlation site DOMs). The Source Description (SDES) packet can distribute information such as the "Canonical Name" of a source, the user name/email address of the person controlling the source and other such generic information. This specification extends the information elements to include[7]:
  - an ***Active Bit Stream Mask (ABM)*** which identifies the bit streams constituting the stream of packet samples corresponding to a particular SDES packet.
  - a ***Sampling Frequency (SFR)*** field which specifies the sampling frequency in kilo-samples per second
  - a ***Samples Per Packet(SPP)*** field which specifies the number of packet samples within a packet
- ***RTCP BYE Packet:*** used to indicate that a source is leaving a session and is no longer active. It is distributed to all session participants to allow them to update their internal state appropriately.
- ***Application Defined RTCP Packet:*** used by source applications to transmit their own control information. Currently, only one subtype is defined:
  - ***PDATA Packet:*** used for transmitting PDATA. This packet includes: a 32-bit UT Second and 64-bit Sample Offset, a source identifier and PDATA(which is transported as an ASCII string).

Figures 4-10 show the formats for these 5 packet types. See [5] for further information. Table 2 shows the mapping between the VLBI data/control signals transferred and the mechanisms used to transfer them.

10.1 RTP Data Encapsulation

Figure 12 shows a simplified VSI-H model with the elements relevant to VSI-E. The data input to the DIM is in the form of 32 individual ***bit streams***, from which a subset of 2^n are chosen to be 'active bit streams'. Any given sample of these active bit streams at the DIM is called a 'DIM sample'. The 'active bit streams' are further subdivided into 2^m packet sample streams, each sample of which is a ***packet sample***. A sequential set of packet samples is encapsulated into each RTP data packet, which is then time-stamped and transmitted.

Figure 4 shows the format of an e-VLBI RTP data packet. Table 3 provides a detailed explanation of each of the fields. An e-VLBI RTP source keeps track of three timescales: UT, experiment time and traditional RTP time. The experiment timescale is transmitted in a 64-bit "Experiment Timestamp" field at the start of each VLBI payload. This field keeps track of how many packets have been transmitted on each packet stream. It is incremented by one for every packet transmitted (note that it is initialized to a random number). The use of a 64-bit field alleviates the problem of wrap-around at high transmission rates. The 32-bit field in each RTP data packet (the "RTP timestamp"field) is used to maintain coarse grain timing synchronization between the source and destination for the purposes of monitoring real-time statistics. The use of a separate

[7] This information allows the receiver(s) to calculate the BSIR and packet sizes.

experiment timescale allows timing synchronization between the DAS and the DPS to be maintained when e-VLBI data is transported in non-real time.

10.2 RTCP

RTCP is a companion protocol to RTP that provides mechanisms for reliably monitoring a network's real-time data delivery performance, collecting statistics from receivers and delivering this information to senders and network management elements.

For more detailed information on the function of the different types of RTCP packets refer to Appendix A. For details of the order of transmission of RTCP packets etc., refer to Appendix B.

RTCP may transmit packets singly or as a "compound packet" to minimize lower-layer overhead. For VSI-E, we recommend the use of compound packets to minimize overhead. The RTP standard states that each RTCP compound packet must contain either a Sender Report or a Receiver Report, along with a Source Description packet with a "CNAME" item that identifies the transmitting host.

RTCP compound packets are transmitted using the algorithm described in Appendix B and Appendix C. Note that RTCP requires that the exact time of transmission be randomized to prevent synchronization of RTCP packets across large numbers of sources.

At the beginning of each session VLBI sources will transmit a compound RTCP packet containing a Source Report, an SDES packet and an Application Defined Message. The Sender Report contains two sets of synchronization points: a 64-bit Experiment timestamp with its corresponding 32-bit UT Second and 32-bit Sample Offset fields for providing DAS to DPS timing synchronization and a 32-bit RTP timestamp and its corresponding 64-bit Extended UT timestamp for DIM to DOM synchronization and. The SDES packet contains the following items:

- CNAME: mapping from the source's SSRC to its canonical identifier
- PRIV message with an EVLBI-ABM field: Active Bit Stream Mask or list of bit-streams that are active for this stream of packet samples.
- PRIV message with an EVLBI-SFR field: Sampling Frequency.
- PRIV message with an EVLBI-SPP field: number of packet Samples Per Packet.

This information allows a receiver to calculate the value of BSIR at the Data Acquisition System. It also allows the receiver to decode the RTP timestamp on each incoming RTP packet and convert it to UT.

Once a session has been established, senders and receivers generate reports at intervals determined by the algorithms described in Appendix B and Appendix C. The Sender Reports allow the receivers of the report to resynchronize their clocks at high resolution. Senders continue to transmit data for the duration of the session.

Before a sender terminates transmission, it must send an RTCP compound packet that contains a BYE packet to inform other participants that it is ceasing transmission.

10.3 Experiment Timestamp Synchronization

This specification provides a mechanism that allows the implicit labeling of every packet sample within an RTP packet with a UT timestamp. This mechanism is in two parts:

- ***periodic RTCP Sender Report Packet:*** contains an Experiment timestamp and its corresponding UT second and Sample Offset, which provides a synchronization point that receiver(s) can use to convert between Experiment timestamps and UT. Sender Reports are transmitted in a quasi-periodic fashion.
- ***per-packet Experiment timestamp:*** a counter that is incremented once per RTP packet (each stream of packet samples maintains its own set of Experiment timestamps).

By making use of the periodic[8] RTCP Sender Report packets (which act as synchronization packets), the per-packet Experiment timestamp and the known sampling rate, the destination(s) is able to calculate an explicit time for each packet sample that is received. Figure 13 illustrates the timestamp synchronization mechanism. On the left hand side of the figure is a DIM transmitting data through a network to a DOM on the output side. A sequence of packets is shown above the network diagram. These packets are typical of the exchange that would take place during timestamp synchronization. The first packet sent from the DIM is an RTCP sender report that contains a 64-bit Experiment timestamp and the corresponding UT second and Sample Offset. The DOM receives the Sender Report, decodes it and stores the information in a table. The next packet sent is an RTP Packet that contains some data. The Experiment timestamp can be converted to UT by using the synchronization reference point provided by the Sender Report (Experiment timestamp, UT second and Sample Offset). By using the difference between the Experiment timestamp of the current data packet and the Experiment timestamp in the Sender Report, and by using the known sampling frequency and packet size, it is possible to determine the exact UT time of any given sample (accuracy is limited only by the mathematical precision of the receiver). At the DIM side, succeeding RTP packets have their Experiment timestamp each incremented by the number of samples.

Figure 14 shows the initial timing synchronization exchange. Initially, an RTCP Sender Report is sent with a UT Second, Sample Offset and the corresponding Experiment Timestamp. The sender then starts transmitting "empty" RTP data packets that have their Invalid bit set for some period of time (for example, 10 s). This gives the receiver time to synchronize its sampling clock and take care of any other setup issues. After the initial "grace" period, the sender starts transmitting valid RTP data packets. Sender Reports are transmitted by the sender in an approximately periodic manner. Using the information from these Sender Reports, receivers are able to resynchronize their recovered sample time clocks.

[8] Note that the specification does not rely on periodicity of the RTCP Sender Report packets for synchronization. The information inside the RTCP Sender Reports are sufficient.

10.4 Example RTCP Message Flow

Figure 15 shows an example of a DIM and a DOM exchanging RTCP/RTP messages across an IP network. The DIM has an IP address of 192.168.1.100, while the DOM has an IP address of 192.168.2.200.

At the start of the session, the DIM sends a compound RTCP packet to the DOM. This serves three purposes: it allows the DIM to identify itself to the DOM, forward session information and provide an initial synchronization point. The packet consists of a Sender Report packet and a Source Description packet. The Sender Report contains timestamp synchronization information (Experiment timestamp and corresponding UT second and Sample Offset). The SDES packet contains the DIM's CNAME, which in this case is its IP address. The SDES packet also contains a number of other items, in particular the Active Bit-stream Mask ("evlbi-abm"), Sampling FRequency("evlbi-sfr'), and Samples Per Packet ("evlbi-spp"); all transmitted in little endian format.

The DOM sends an RTCP packet containing a Receiver Report and an SDES packet. The Receiver Report contains reception statistics for any sources that have transmitted packets to the DOM in the last RTCP interval. The SDES packet contains only the CNAME of the DOM, which is its IP address.

On its next RTCP transmission time, the DIM sends an RTCP packet that contains an Application defined packet (in addition to the mandatory Sender Report packet and SDES packets). The Application defined packet contains PDATA to be transferred to the DOM.

The DIM then starts transmitting RTP data frames. These packets contain the sampled VLBI data that is to be processed at the correlator site. Periodically, the source resynchronizes its sampling clock with the receiver using the procedure outlined in §10.3(not shown here).

Once the DIM has finished transmitting data, it sends an RTCP packet that contains a BYE packet to indicate to the DOM that it is going inactive. At this point, the session between the DIM and the DOM has been closed.

10.5 RTP Flow Control

In order to ensure that the RTP traffic generated by the e-VLBI application does not impact other network users in a shared-network environment, it is highly recommended that end systems implement some form of TCP friendly flow control and/or use a 'less than best effort service' (refer to §9).

One candidate reference model is the Experiment-Guided Adaptive Endpoint (EGAE) [12, 13]. The EGAE combines flow control, packet marking for less-than-best-effort service and the capability to switch between real-time and non-real-time modes of operation into a single unit. The network characteristics of this unit are controlled by an experimental profile supplied by a user. The EGAE implements a flow control algorithm that has been specially tailored to the requirements of e-VLBI data transfer, but that is also friendly to other network users.

10.6 Endian and Bit-Numbering Conventions

The VSI-E specification assumes standard network byte (big endian) ordering for RTP and RTCP packet fields except for the following fields:

- Sample data bits in the RTP data packet
- UT Second in the Sender Report
- Sample Offset in the Sender Report
- SDES PRIV Active Bit-stream Mask
- SDES PRIV Sampling FRequency item
- SDES PRIV Samples Per Packet item
- Application Defined Packet UT Second field
- Application Defined Packet UT Sample Offset field
- Application Defined Packet PDATA field

Network byte order is by convention “big endian” with the most significant bit (MSB) of a 32-bit word labeled 0 and the least significant bit (LSB) labeled 31. Standard form is to display the most significant bit on the left hand side of the page [14]. “Big endian” computer architectures store the ***most*** significant byte of a multi-byte word towards memory address “0”. Conversely, “little endian” architectures store the ***least*** significant byte of a multi-byte word towards memory address “0”. When transmitting a byte stream to the network, the most significant bit is always transmitted first.

In the commodity personal computer market, by far the majority of computers are Intel-based and so have a “little endian” architecture. The VSI-E specification assumes that the systems that participate in the transport of VLBI data across a network are commodity, intel-based, “little endian” computers. Therefore, any data field that is to be processed by the end systems is transmitted in “intel” or little-endian format. Our convention for displaying this is to number the most significant bit of a 32 bit word as 31 and the least significant bit as 0, displaying the most significant bit on the left hand side of the page. The rationale for transmitting the data in this manner is to avoid unnecessary byte swapping at the end systems in order for them to interpret the data being transmitted.

11. Tables

VPT Value	Description
0000	1 bit per channel sample
0001	2 bits per channel sample
0010	4 bits per channel sample
0011	8 bits per channel sample
0100	16 bits per channel sample
0101	32 bits per channel sample
0110 - 1111	undefined

Table 1: VLBI RTP Payload Types

Information	Transport Mechanism
Primary data stream	Transported using RTP Data Packet
Active bit-stream mask	Transported using RTCP SDES Packet with Active Bit Stream Mask item.
DOT time-tagging	Carried in each RTP data packet as a 64-bit Experiment timestamp. Synchronization with UT provided by RTCP Sender Report packets that contain a UT second, Sample Offset and Experiment timestamp.
Bit-stream information rate (BSIR)	Information to calculate the BSIR is distributed using the RTCP SDES packet with the following items: Sampling Frequency (SFR), Samples Per Packet (SPP) and the RTP VPT field which specifies how many bits per packet sample.
Valid-data indicator	Carried in the RTP Data Packet header as a single bit (I-bit).
TVG-data indicator	Carried in the RTP Data Packet header as a single bit (T-bit).
PDATA messages	Carried using the Application Defined RTCP Packet. This carries the PDATA as well as a 32-bit UT second and a 32-bit Sample Offset.

Table 2: e-VLBI Parameter Transport on RTP/RTCP

Field	Value
V	version field, which is always set to 2
P, X	always set to 0
CSRC	always set to 0
M	RTP marker bit. Use not specified by this specification.
I	Invalid bit. 1: invalid data samples in this packet; 0: valid samples in this packet
T	test vector bit. 1: the data in the packet payload is a test vector. 0: the data consists of packet samples (the validity of which is subject to the 'I' bit defined above).
VPT	VLBI Payload Type – number of bits per channel in the payload data. See Table 1
RTP timestamp	32 bit integer. Initialized to a random number and incremented by one for every packet transmitted.
SSRC	32 bit identifier used to indicate the Synchronization Source. In this specification, the SSRC is used to indicate a particular channel stream. The SSRC is unique amongst all channel streams that are transported as part of an experiment.
Experiment timestamp	64-bit integer. Incremented by one for each RTP data packet created. Provides timing synchronization between the DAS and the DPS.
Data Payload	channel-stream encapsulated into an integer number of 32-bit words in format given in Figure 1. This field is in little endian order.

Table 3: RTP Data Format

Field	Value
V	version field, which is always set to 2
P	always set to 0
RC	count of the number of Reception Reports that are included in this packet; due to the nature of e-VLBI, it is not necessary for Senders to include any reception reports in their Sender Report packets. Set to 0.
PT	always set to 200
length	length of this packet in 32-bit words (including the length of the header) minus 1
SSRC of sender	Synchronization Source (SSRC) identifier of the sender of this packet; this will be the SSRC identifier associated with the channel stream from a given DIM.
UT timestamp, msb	32 most significant bits of UT timestamp corresponding to the time at which this packet was transmitted. The 64-bit UT timestamp is an unsigned integer that represents the number of seconds relative to 0h UTC on 1 January 1900 [15]. The 32 most significant bits represent the integer number of seconds..
UT timestamp, lsb	32 least significant bits of the UT timestamp. This represents the fractional part of the UT time.
RTP timestamp	this is the RTP timestamp that corresponds to the UT timestamp above.
sender's packet count	number of packets that have been transmitted by the sender from the time it started transmission up until the time this Sender Report was generated
sender's octet count	number of octets that have been transmitted by the sender from the time it started transmission up until the time this Sender Report was generated
The next section of the packet is divided up into blocks, with each block summarizing the statistics for a particular source that is identified by a SSRC identifier at the start of the block	
SSRC_1	the source with SSRC_1 as its SSRC identifier
fraction lost	fraction of packets from SSRC_1 that have been lost since the last Receiver or Sender Report
cumulative # of packets lost	cumulative number of packets that have been lost from this source since the beginning of reception
extended highest sequence # rec'd	highest sequence number received, a 16-bit quantity, combined with an additional 16-bit extension that indicates how many sequence number cycles have occurred
interarrival jitter	estimate of the jitter for RTP packets received, as measured in timestamp units. The inter-arrival jitter is a measure of the difference in spacing between packets at the sender and the corresponding space between the same packets at the receiver
last SR (LSR)	middle 32 bits of the NTP timestamp received in the last Sender Report received. Set to 0 if not Sender Report had been received at the time of transmission.
delay since last SR (DLSR)	delay in units of 1/65536 s between the time the last Sender Report was received and this RTCP Receiver Report was transmitted; along with the 'last SR' field, allows the source, SSRC_1, to calculate the round trip time to the receiver issuing this receiver report block
SSRC_2	beginning of report for source 'SSRC_2, etc

The final section of the packet includes fields required to provide high precision timing synchronization between the DAS and the DPS for both real-time and non-real time delivery.	
UT second	32-bit field. Represents the integral part of a UT time. This field is in little endian format.
Sample Offset	32-bit field. Represents the fractional part of a UT time in units of sample intervals (where the sampling frequency is known to both the sender and the receiver). This field is in little endian format.
Experiment timestamp, msb	32-bit field. Represents the most significant bits of the timestamp corresponding to the UT time specified by the UT second and Sample Offset.
Experiment timestamp, lsb	32-bit field. Represents the least significant bits of the timestamp corresponding to the UT time specified by the UT second and Sample Offset.

Table 4: RTCP Sender Report Format

Field	Value
V	version field, which is always set to 2
P	always set to 0
RC	count of the number of Reception Reports that are included in this packet.
PT	always set to 201
length	length of this packet in 32-bit words (including the length of the header) minus 1
The next section of the packet is divided up into blocks, with each block summarizing the statistics for a particular source that is identified by a SSRC identifier at the start of the block	
SSRC_1	the source with SSRC_1 as its SSRC identifier
fraction lost	fraction of packets from SSRC_1 that have been lost since the last Receiver or Sender Report
cumulative # of packets lost	cumulative number of packets that have been lost from this source since the beginning of reception
extended highest sequence # rec'd	highest sequence number received, a 16-bit quantity, combined with an additional 16-bit extension that indicates how many sequence number cycles have occurred
interarrival jitter	estimate of the jitter for RTP packets received, as measured in timestamp units. The inter-arrival jitter is a measure of the difference in spacing between packets at the sender and the corresponding space between the same packets at the receiver
last SR (LSR)	middle 32 bits of the NTP timestamp received in the last Sender Report received. Set to 0 if not Sender Report had been received at the time of transmission.
delay since last SR (DLSR)	delay in units of 1/65536 s between the time the last Sender Report was received and this RTCP Receiver Report was transmitted; along with the 'last SR' field, allows the source, SSRC_1, to calculate the round trip time to the receiver issuing this receiver report block
SSRC_2	beginning of report for source 'SSRC_2, etc

Table 5: RTCP Receiver Report Format

Field	Value
V	version field, which is always set to 2
P	always set to 0
SC	count of the number of source description "chunks" that are included in this packet.
PT	always set to 202
length	Length of this packet in 32-bit words (including the length of the header) minus 1
The remainder of the packet is divided up into "chunks", with each chunk consisting of 0 or more "items" that describe the source with the SSRC at the start of the chunk.	
SSRC of sender	Synchronization Source (SSRC) identifier of the source that is described by the following SDES items. The SSRC and the following items are collectively referred to as a "chunk". Chunks must begin on a 32-bit word boundary.
Type	this is the item type for PRIVately defined items (this allows applications to define their own item types).
item length	this is the length of the item data in octets (no including the type and length bytes).
prefix length	for items of type PRIV, this describes the length of the following prefix string (which describes the item). The remaining octets of the packet contain data.
evlbi-abm	Active Bitstream Mask: this indicates that the next 32 bits will contain a bit mask that indicates which bitstreams at the interface between the DAS and the DIM make up this stream of packet samples. This field is in little endian order.
evlbi-sfr	Sampling Frequency: this indicates that the next 32 bits will indicate the sampling frequency of this channel stream (in kilo-samples per second). This field is in little endian order.
evlbi-spp	Samples Per Packet: this indicates that the next 32 bits will indicate how many packet samples are contained in each data packet. This field is in little endian order.

Table 6: RTCP SDES Packet

Field	Value
V	version field, which is always set to 2
P	always set to 0
SC	count of the number of source description "chunks" that are included in this packet.
PT	always set to 203
length	Length of this packet in 32-bit words (including the length of the header) minus 1
The remainder of the packet consists of a list of SSRC's. These SSRC's identify which sources are leaving the session.	
length	an optional field that indicates the length of the reason field in octets.
reason	A text field that contains a value indicating why the sources are leaving.

Table 7: RTCP BYE Packet

Field	Value
V	version field, which is always set to 2
P	always set to 0
subtype	allows for the specification of subtypes within the RTCP application packets. VSI-E currently supports a single type of Application Defined Packet(identified by a subtype of 1)the ***PDATA packet.*** This is used to transport PDATA and its corresponding timestamp.
PT	always set to 204
length	length of this packet in 32-bit words (including the length of the header) minus 1
SSRC	identifies the source sending this packet.
name	identifier for that specifies what information is carried in the payload of this packet. It is made up of 4 ASCII characters. The name "VLBI" is used to identify payloads specific to this specification.
UT Second	32 bit integer. Represents the integral part of a UT time. This field is in little endian format.
Sample Offset	32 bit integer. Representts the fractional part of a UT time in units of sample interval. This field is in little endian format.
PDATA	VLBI PDATA. This field is only present in the PDATA packet. This field is in little endian format.

Table 8: Application-defined RTCP Packet

12. Figures

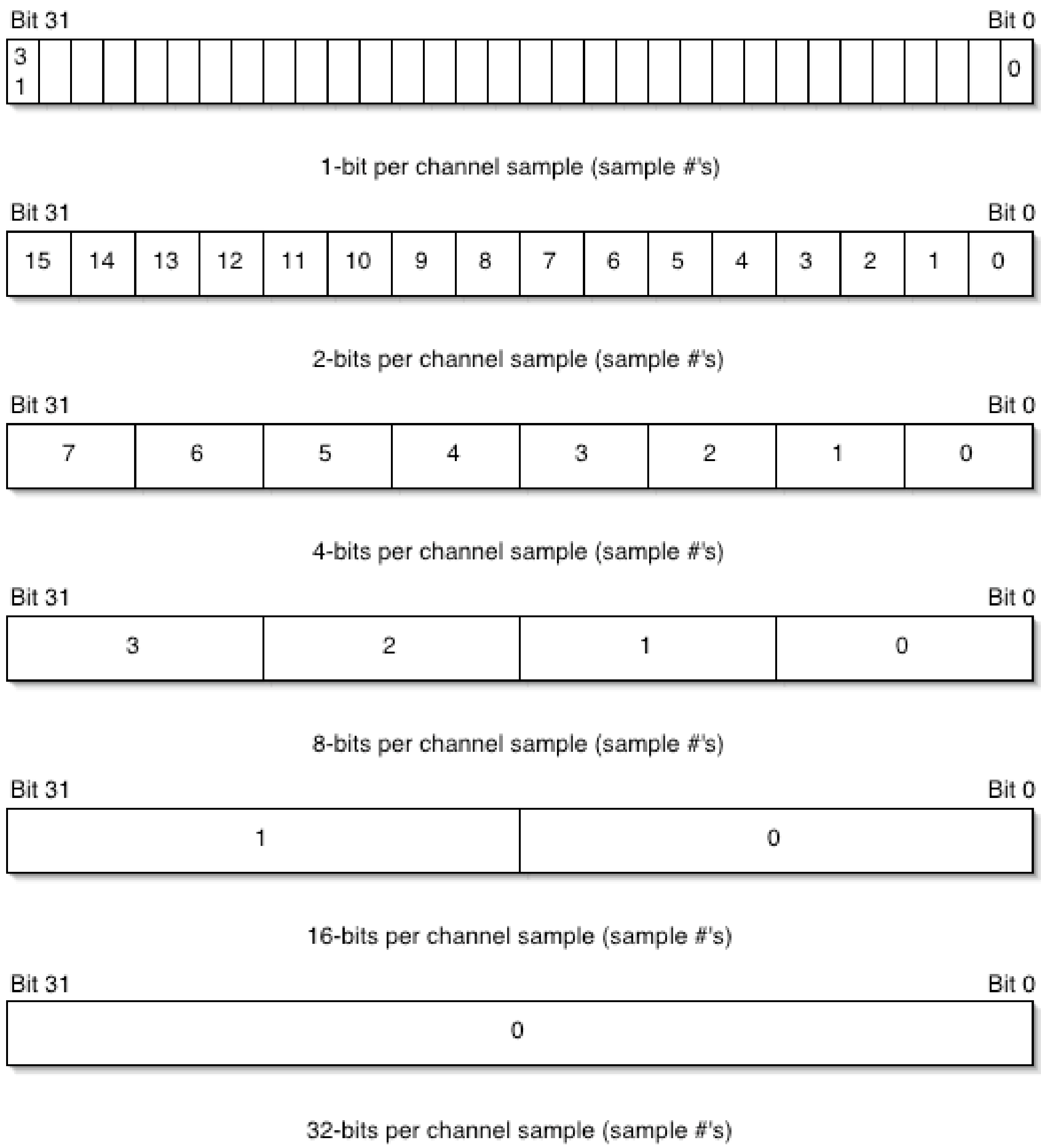


Figure 1: Packet Data Array Format

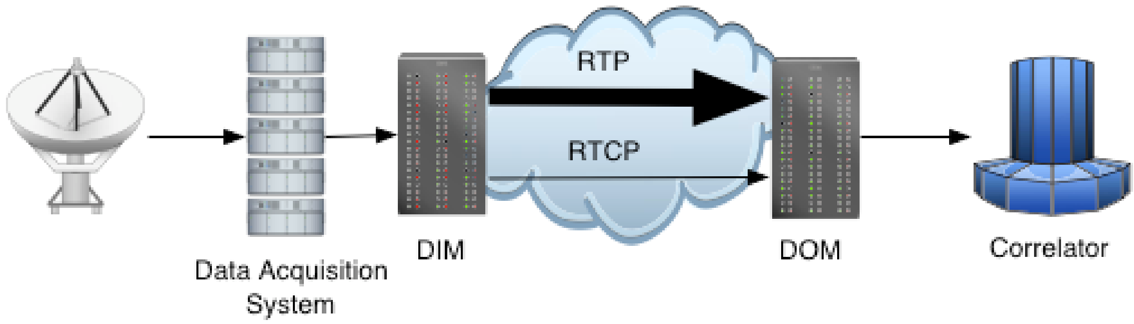


Figure 2: VSI-E RTP/RTCP Overview

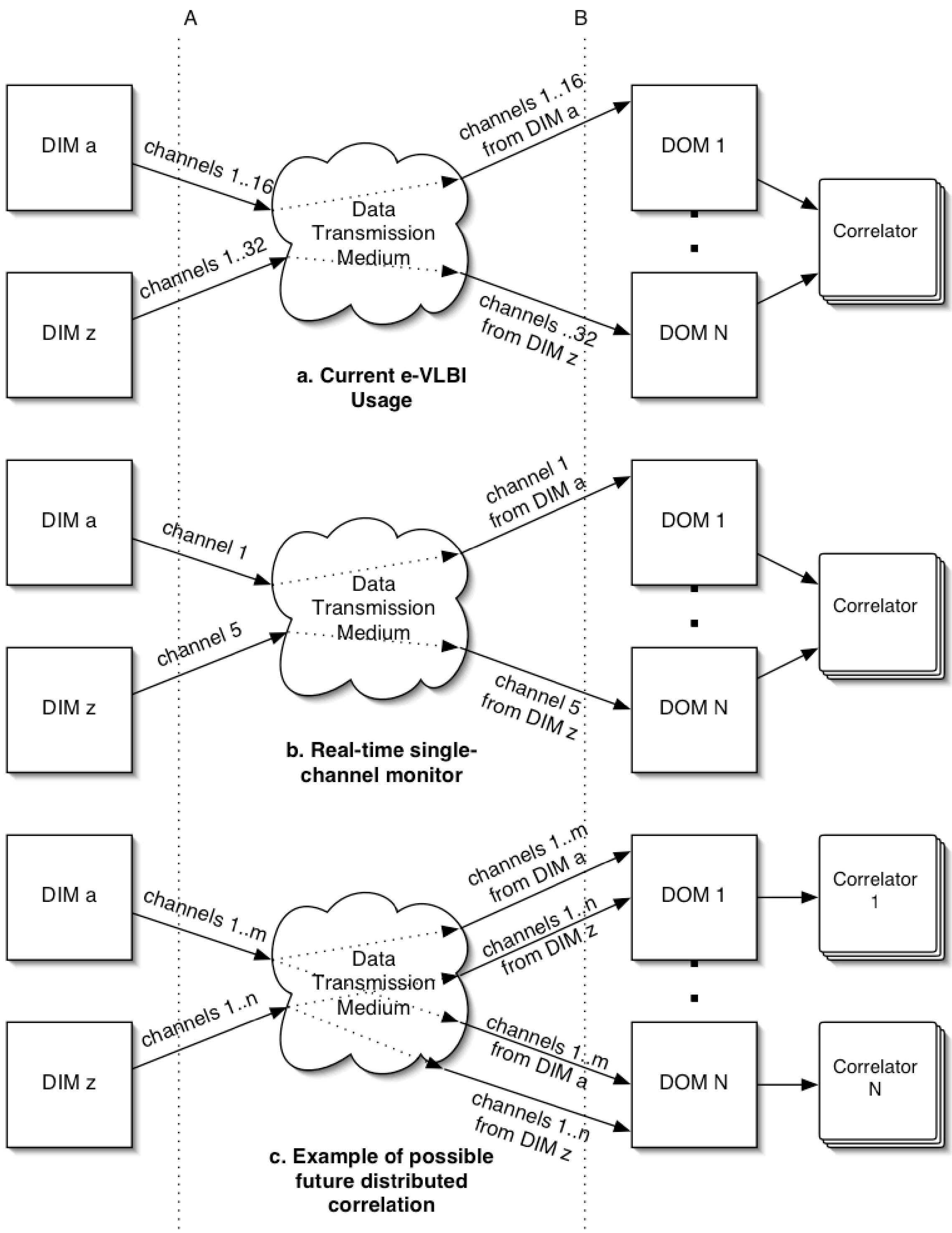
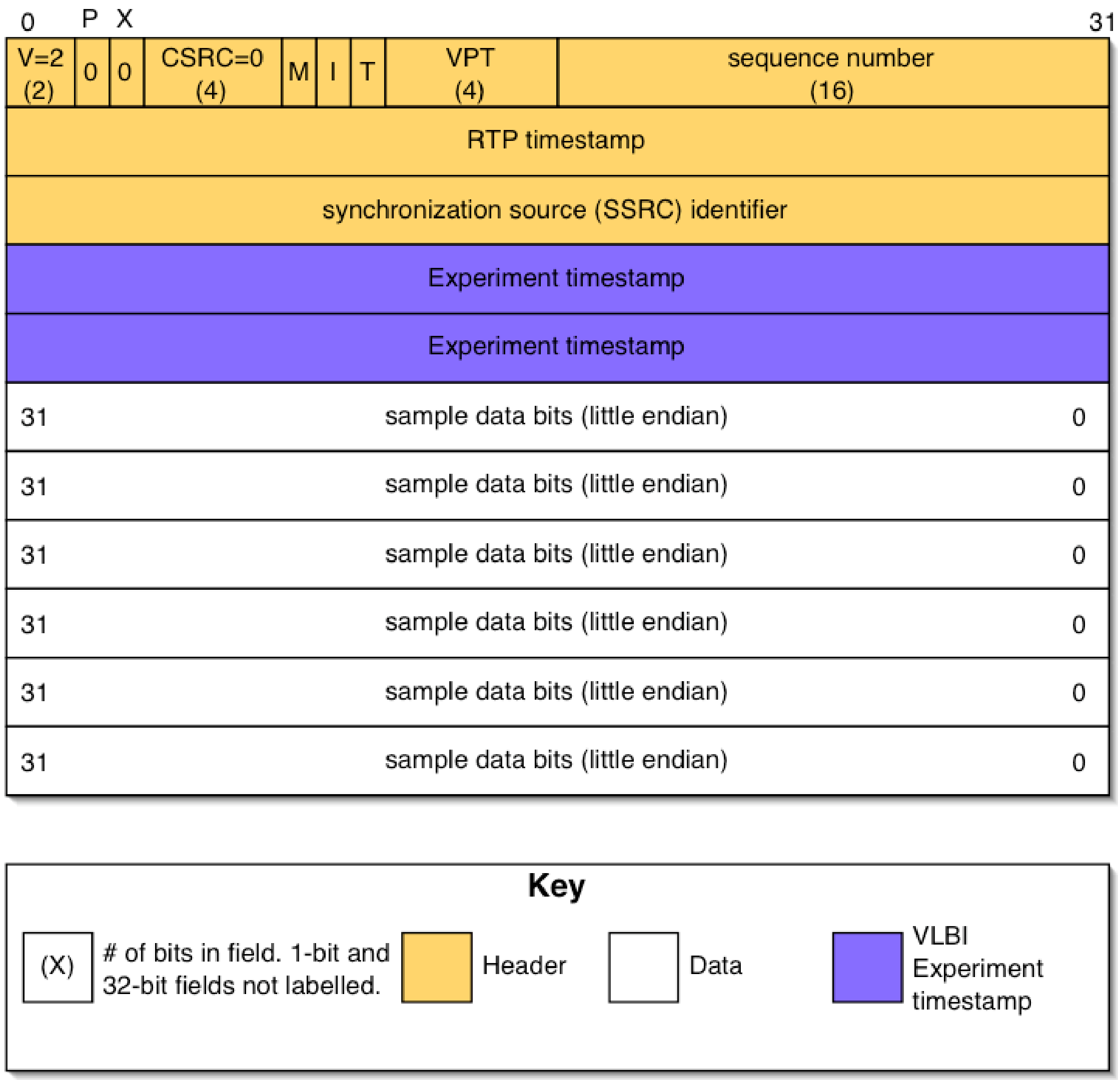


Figure 3 VSI-E Network Scenarios



Note

Data packet headers are in network byte order (big endian with most significant bit labeled as 0 and least significant bit labeled as 31). Within the packet, for practical reasons, the sample data bits are stored in 'intel' order (little endian with least significant bit labeled as 31 and most significant bit labeled as 0). Note that the order of bit transmission is transparent to the end systems. The order of bits as they leave the DOM are the same as they entered the DOM.

Figure 4: RTP Data Packet

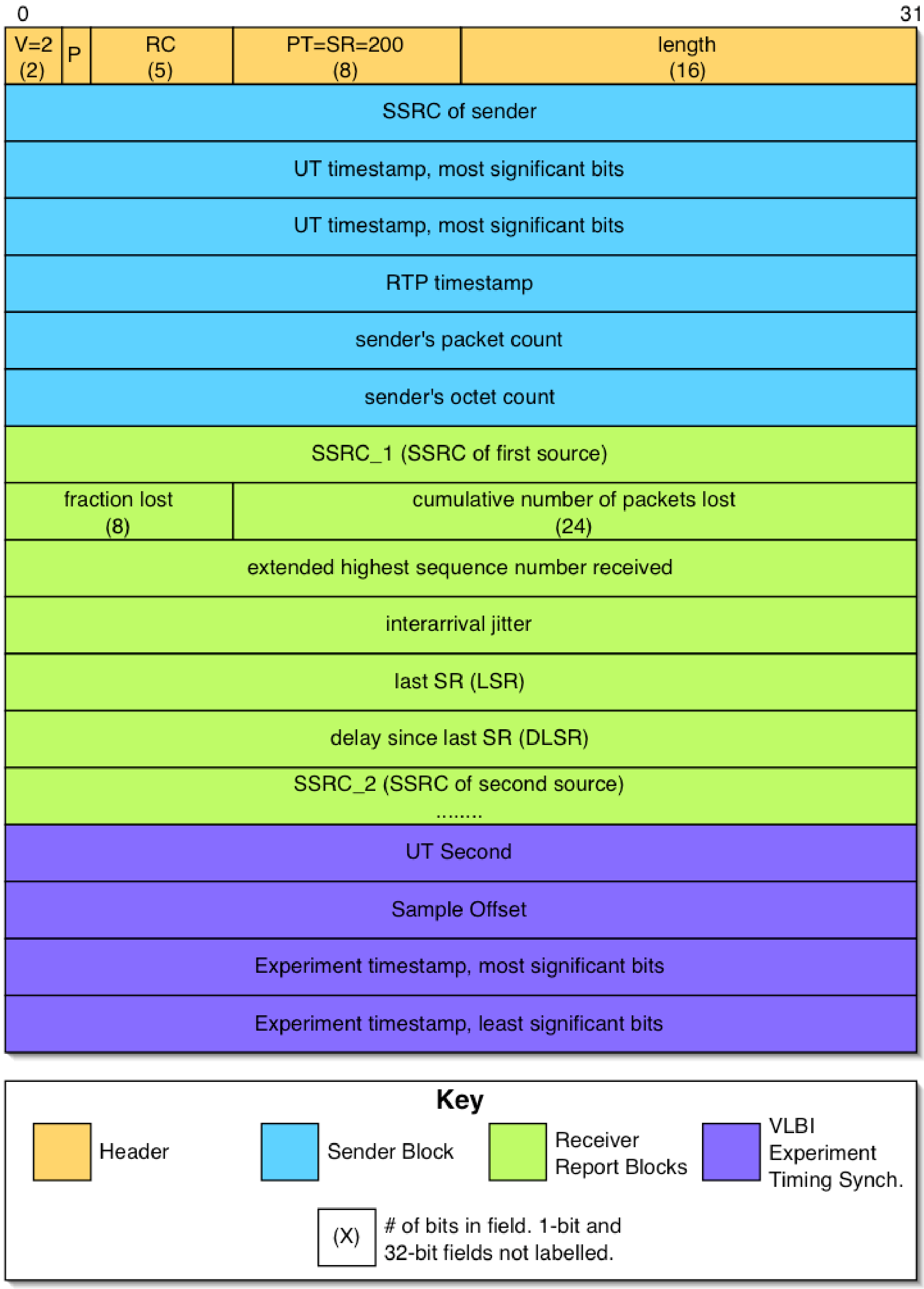


Figure 5: RTCP Sender Report Packet

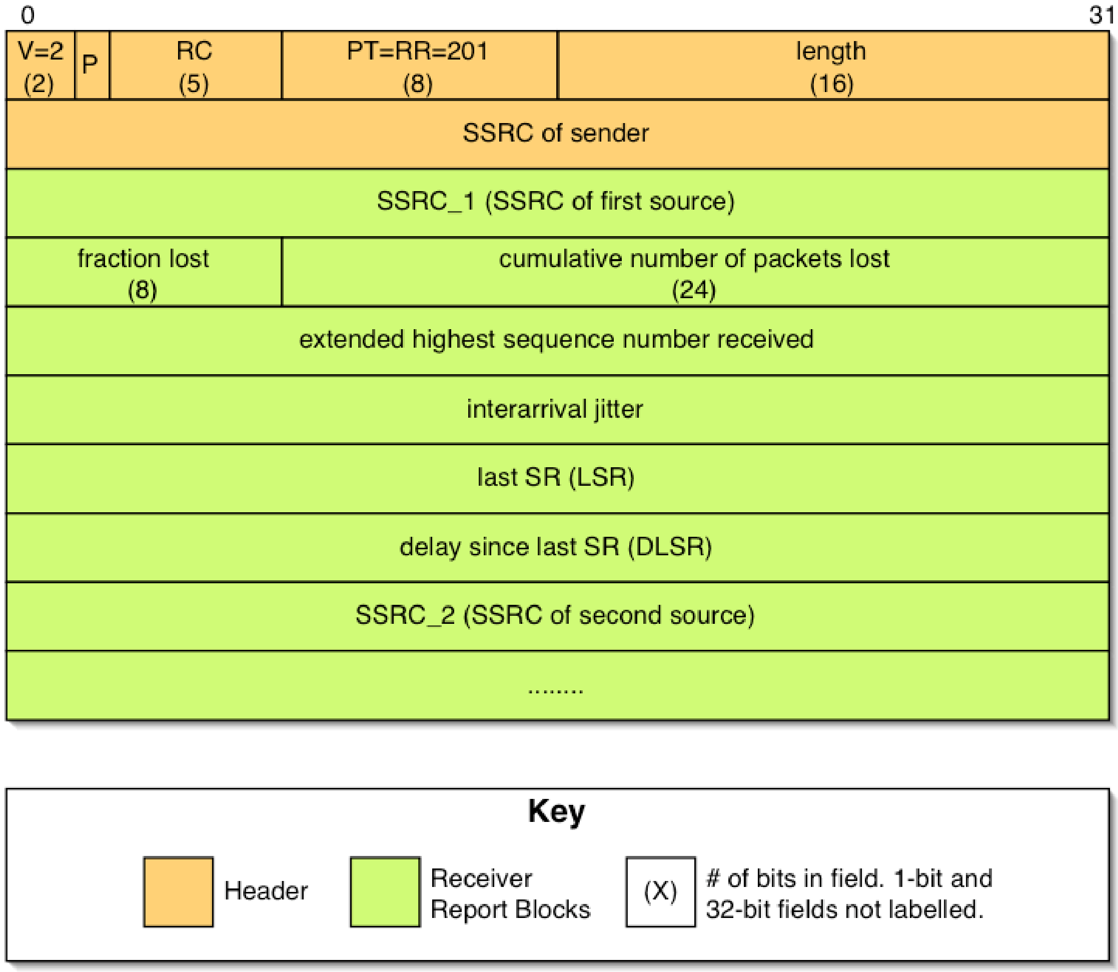


Figure 6: RTCP Receiver Report Packet

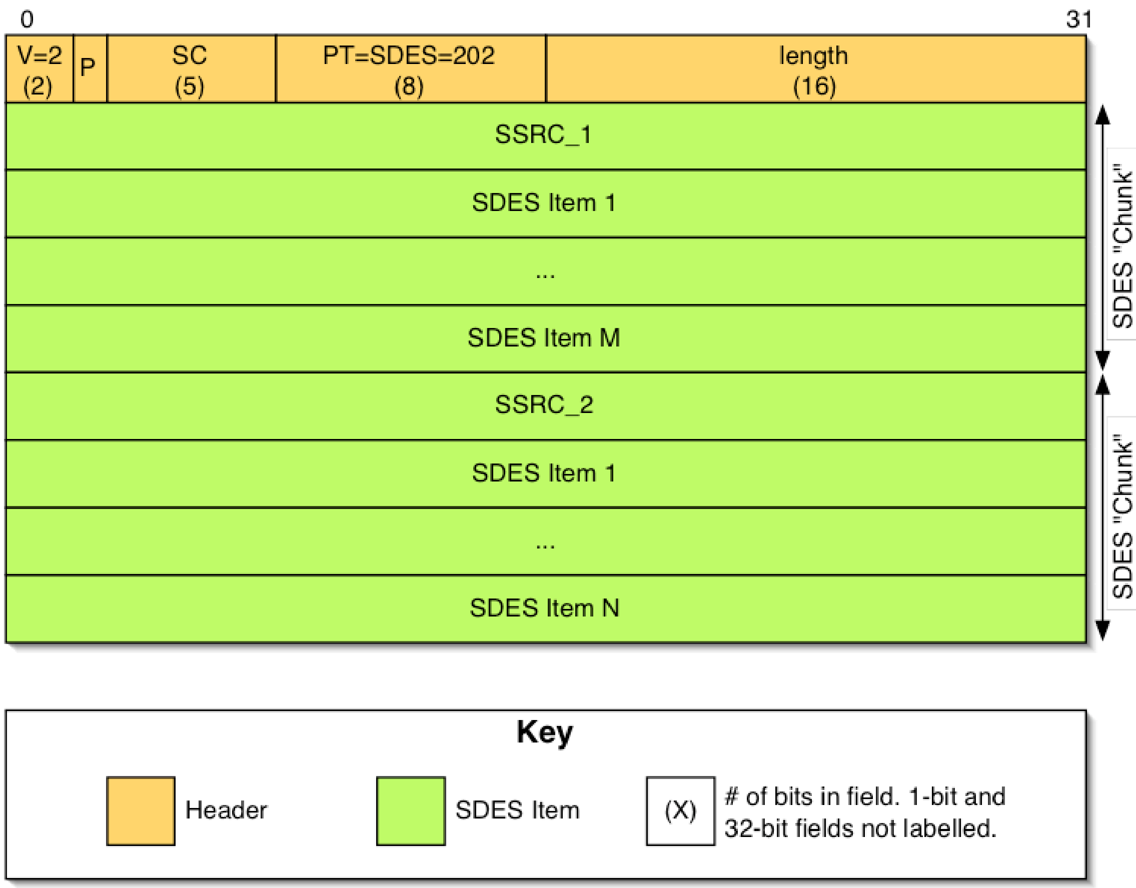
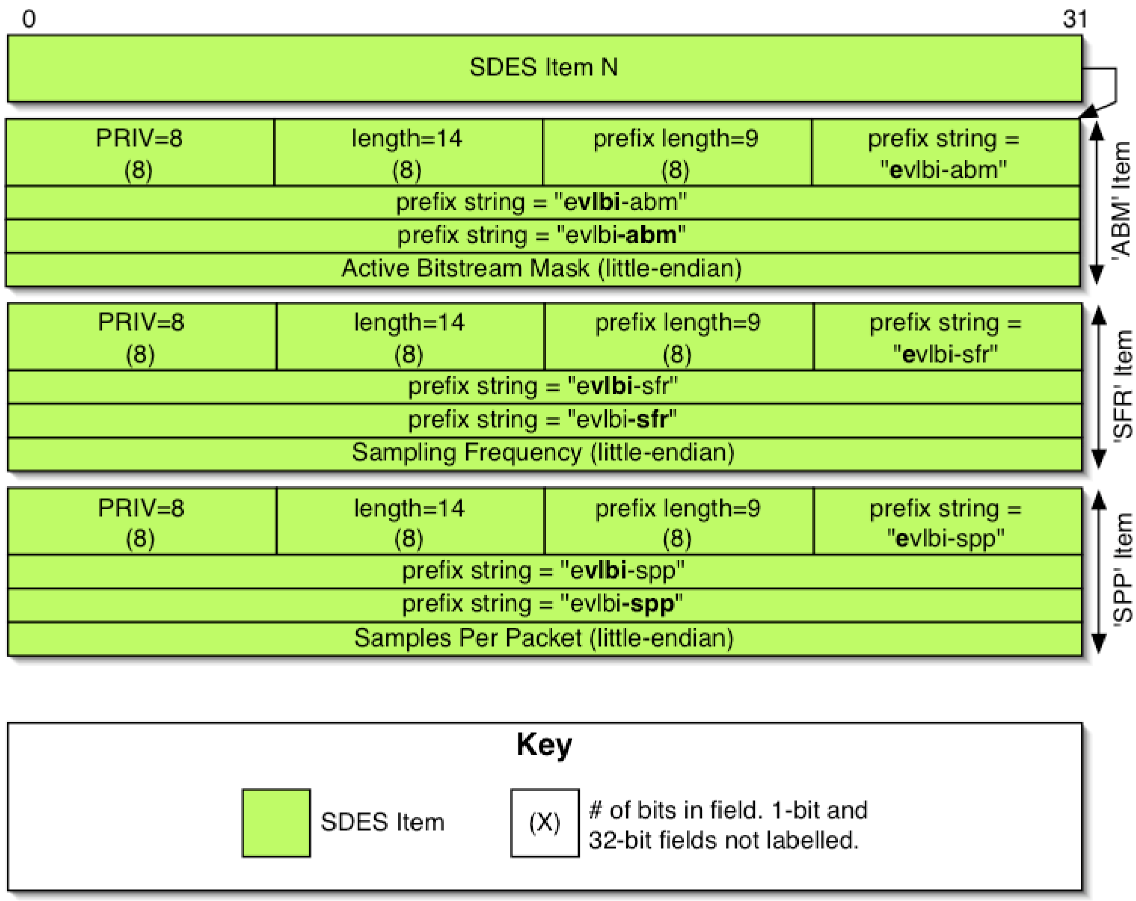


Figure 7: RTCP Source Description Packet Structure



Note
A 9 character prefix string occupies the last byte of the first word of the Item plus two additional words (i.e. 1 character per word). For example, in the first item above, the string "evlbi-abm" occupies the last byte of the first word of the item and the next two 32-bit words.

Figure 8: RTCP e-VLBI Source Description Items

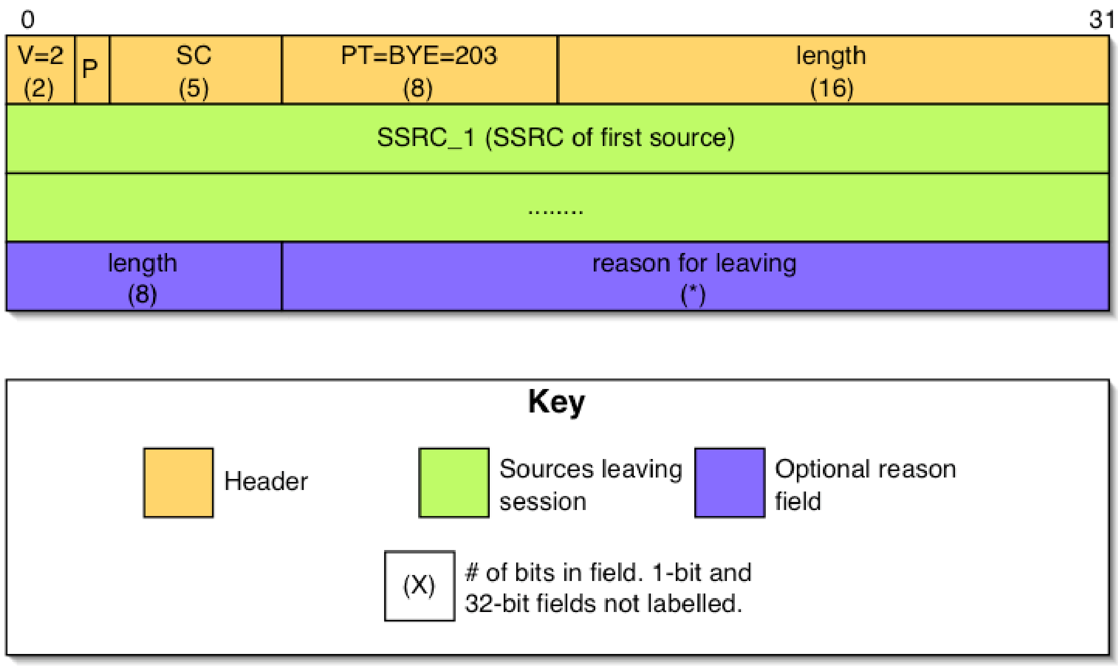


Figure 9: RTCP BYE Packet

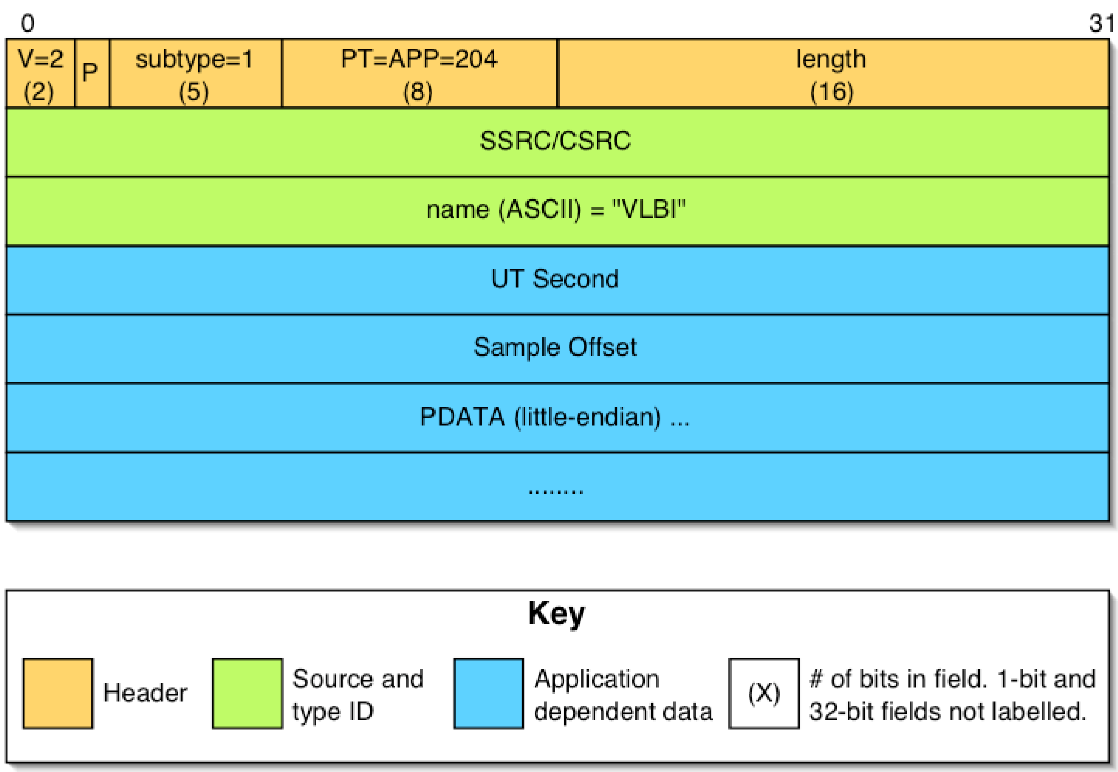
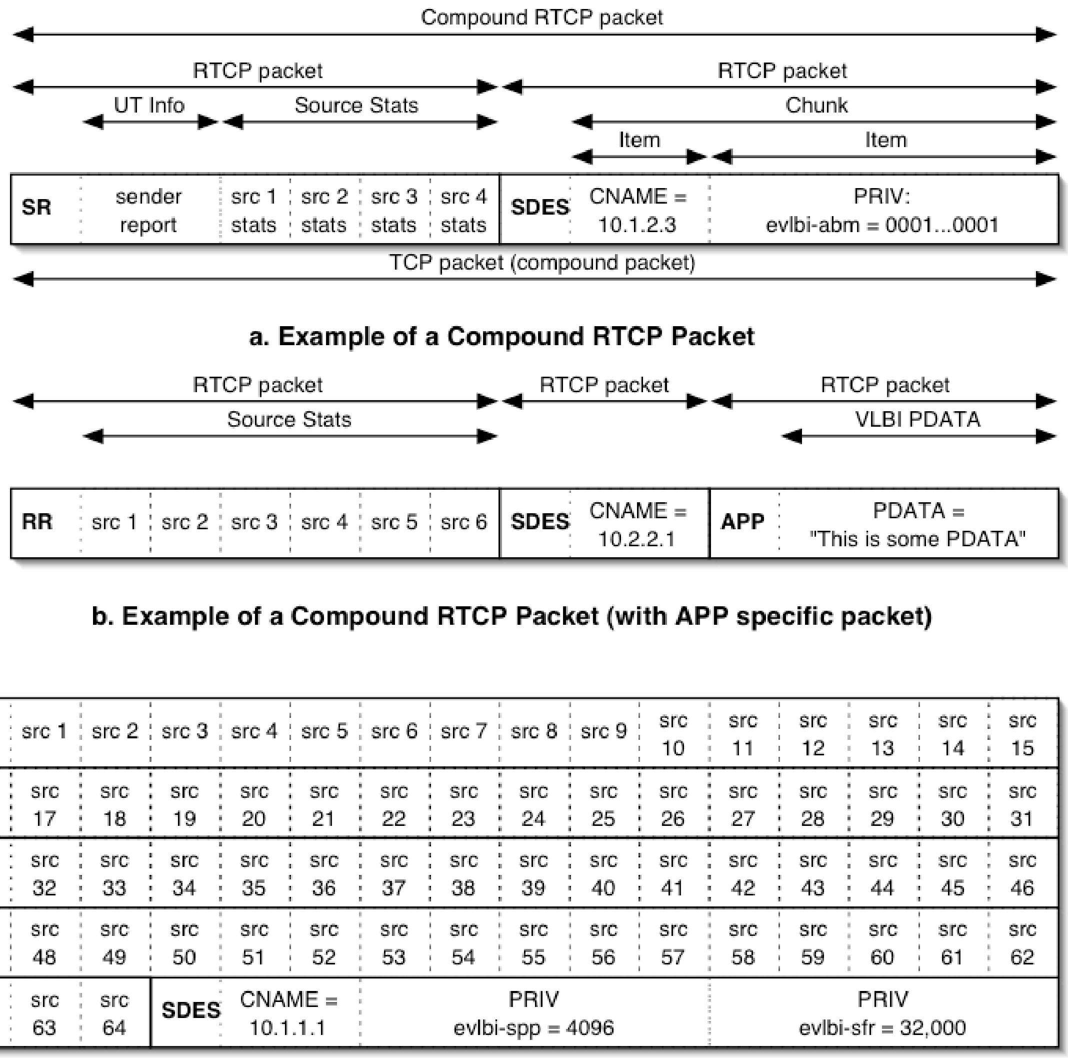


Figure 10: Application-defined RTCP Packet



a. Example of a Compound RTCP Packet

b. Example of a Compound RTCP Packet (with APP specific packet)

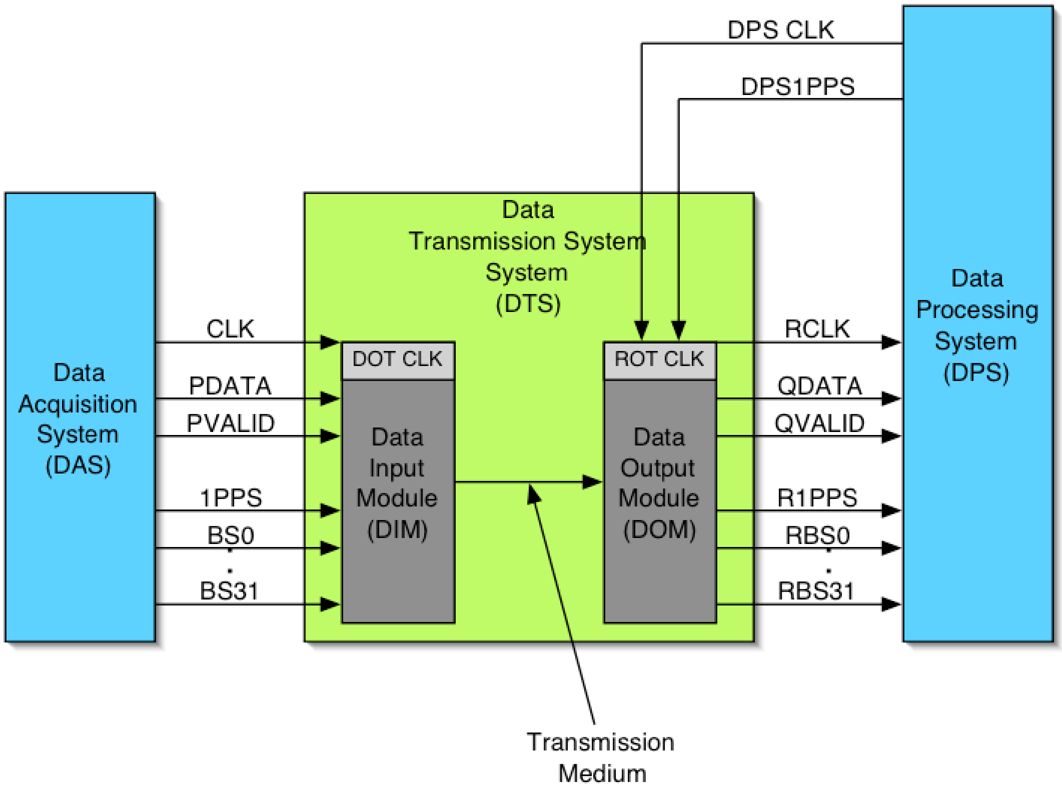
c. Example of a Compound RTCP Packet (with RR for 64 sources)

Note

R = random 32 bit integer for encryption
SR = Sender Report Packet
RR = Receiver Report Packet
SDES = Source Description Packet
APP = Application Specific Packet
PRIV = SDES Private (application defined) item. For e-VLBI there are three types of PRIV messages: evlbi-abm, evlbi-sfr, and evlbi-spp.

The packets above are not drawn to scale (i.e. relative packet/field sizes may not be accurate)

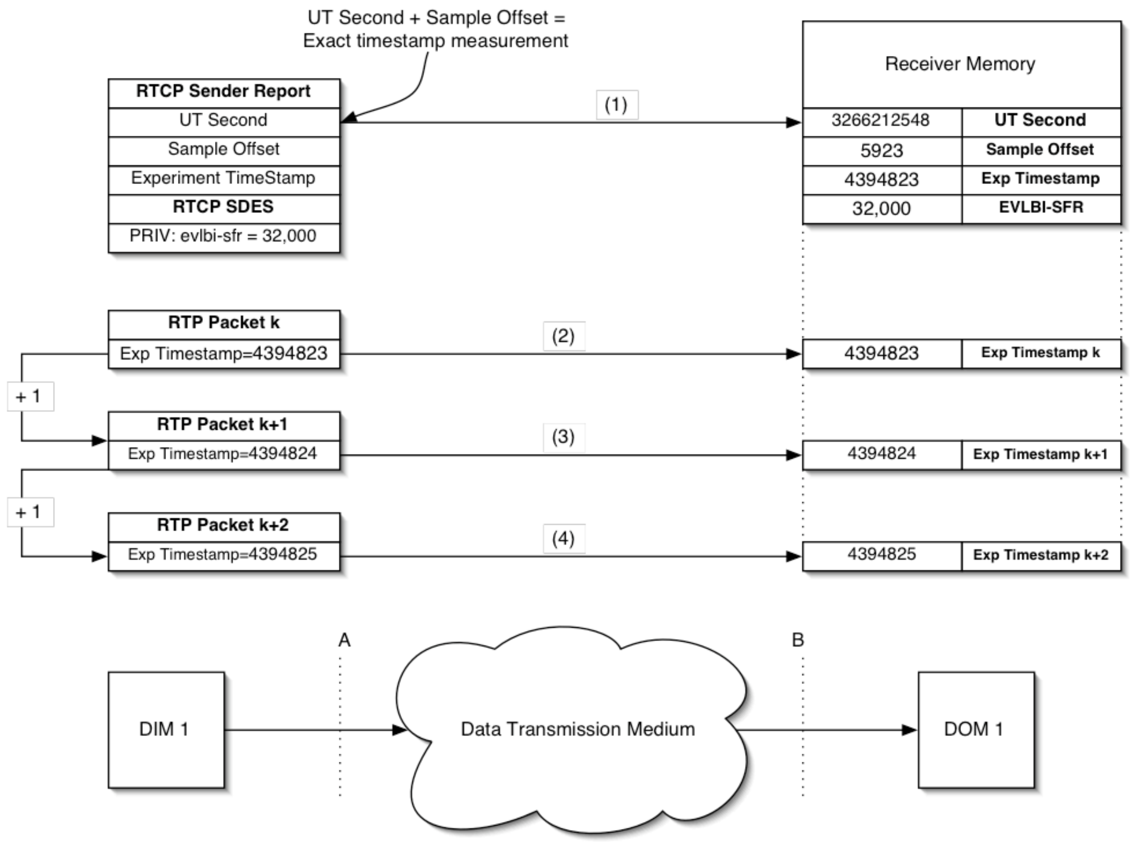
Figure 11: Compound RTCP Packets



Note

CLK = A clock accompanying the bit streams.Provides a reference frequencey for the DIM
PVALID = a signal that specifies the "validity" of the bit streams
PDATA = a standard 8-bit ASCII asynchronous serial data stream
1PPS = A 1 pulse per second tick which defines corresponding parallel data bits
BS0..BS31 = 32 parallel bit-streams, all sampled by the DIM at the same rate
DOT CLK = Data Observe Time Clock: master clock within the DIM used to time tag samples.
ROT CLK = Requested Observe Time Clock: maintains the reference time to which the re-constructed data are to be synchronized
RCLK = clock accompanying the reconstructed bit streams
QDATA = a standard 8-bit ASCII serial data stream
QVALID = 1-bit global signal indicating that the reconstructed data are judged by the DOM to be correct
R1PPS = reconstructed 1PPS accompanying the bit streams
RBS0..RBS31 = reconstructed bit streams. Accurate reproductions of the active sampled bit-streams transferred from the DIM
DPSCLK = a clock from the DPS which acts as a frequency reference for the DOM
DPS1PPS = a 1-pps tick used to set an internal DOM clock called the Requested Observe Time clock to an integer-second epoch

Figure 12: VSI-H Model



Note
Two timescales are in effect: UT or absolute time and the Experiment sampling clock that is used to explicitly timestamp the first data sample in each RTP data packet and implicitly timestamp every data sample.

1. RTCP Sender Report contains a synchronization point that relates UT and Experiment sampling clock. Using this information, the receiver can convert any Experiment Timestamp to a UT time. VSI-E utilizes a the 32-bit UT Second and a 32-bit Sampling Offset to specify the time of a sample exactly. The UT second field specifies the integral part of the UT time that a sample is taken. The Sample Offset specifies the fractional part of the UT time that a sample is taken in units of sampling interval (at the source sampling frequency). The accompanying SDES packet contains the sampling frequency (32,000 kS/s) that is used by the source to convert the Sampling Offset to a fractional part of a second.
2-4. RTP data packets are transmitted with the appropriate timestamp.

The receiver is able to reconstruct the absolute time for each data packet received by using the synchronization point from the RTCP Sender Report and the known sampling frequency.

For example, the UT time for the first sample of the first RTP data packet transmitted can be calculated as:
UT = (3266212548+5923/(evlbi-sfr x 1000))+(4394823-4394823) / (evlbi-sfr x 1,000)
= (3266212548+5923/(32,000 x 1,000))+(4394823-4394823) / (32,000 x 1,000)
= 3266212548.00018509375 s

Figure 13: RTCP Timestamp Synchronization

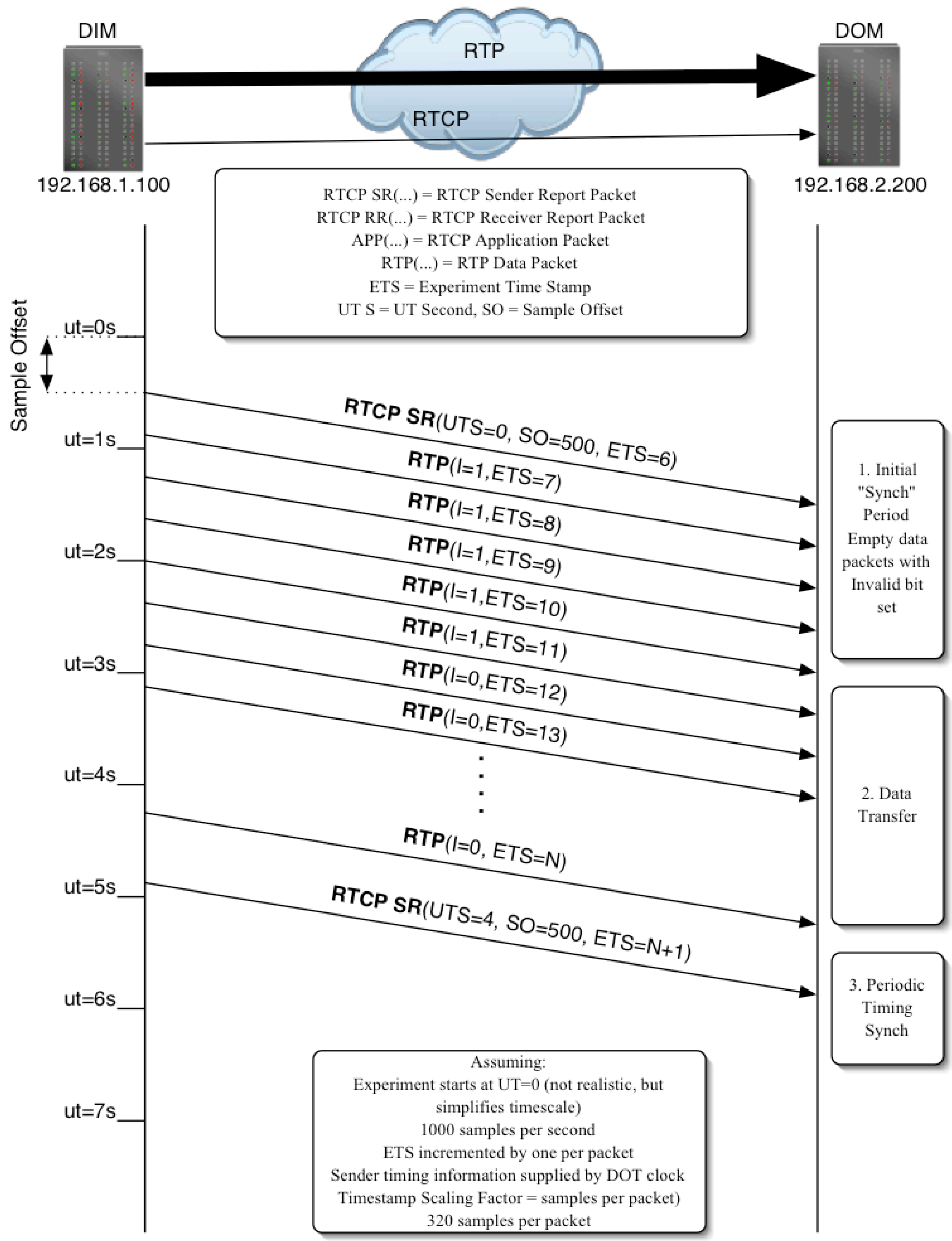


Figure 14: Initial Sample Sychronization

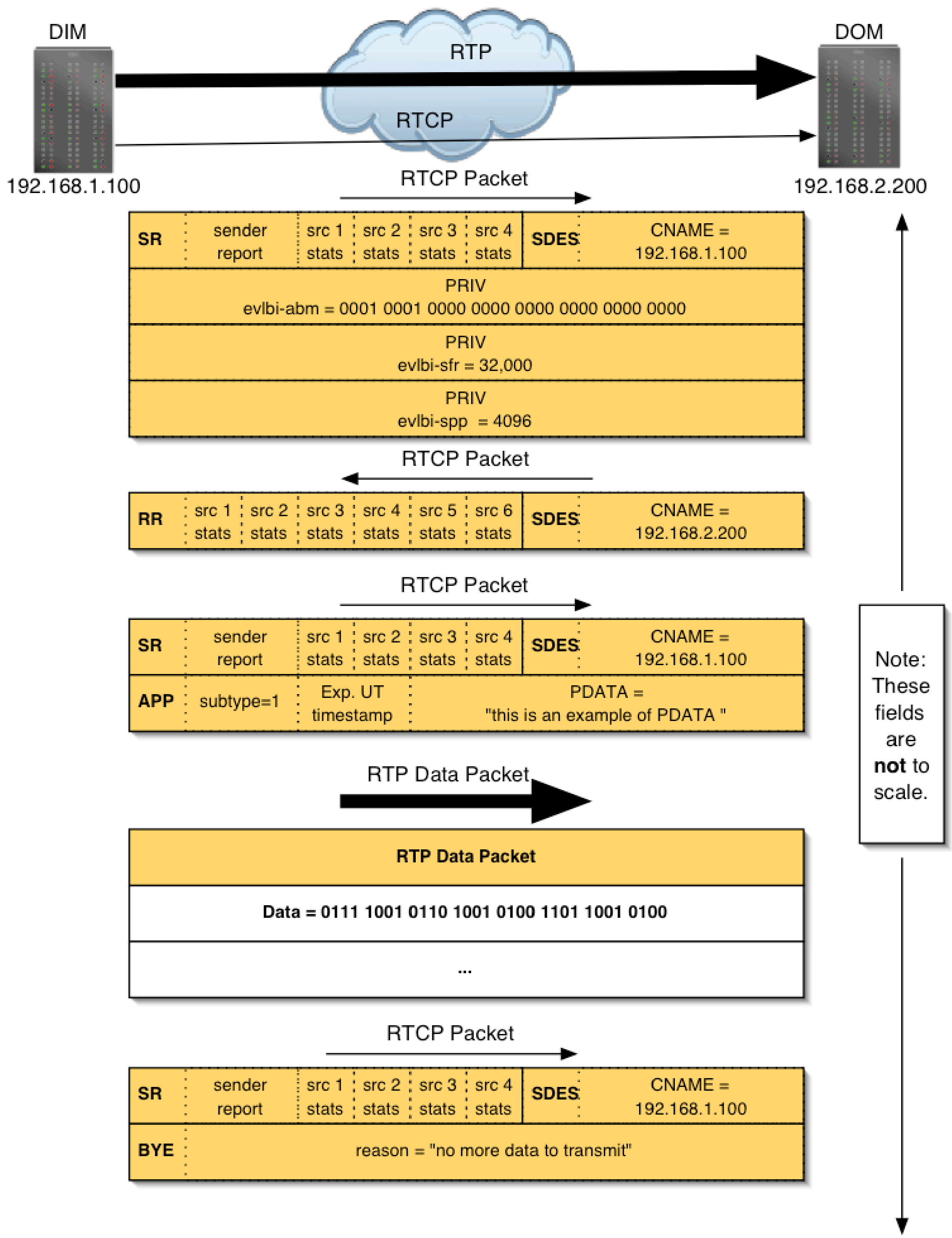


Figure 15: Example RTCP Message Sequence

Appendix A. RTCP

Section 1.01 RTCP Sender Reports

The sender report provides three functions:

- Reference points that allow receivers to synchronize their recovered sampling clocks with the experiment timescale.
- transmission statistics on the number of packets that have been transmitted by the sender.
- reception statistics for all of the sources that have sent packets to this source since the time of the last Sender Report.

Figure 5 shows the format of an RTCP Sender Report; Table 4 indicates the various parameters included in this packet.

The sender report contains a block for every source that has transmitted packets to this receiver since the last reception report. Note that there is a limit of 31 sources per receiver report. If statistics for more than 31 sources are required, multiple receiver reports should be sent.

The VSI-E specification extends the Sender Report to include an additional three fields for maintaining the experiment data timescale:

- UT Second: the integral portion of the UT time corresponding to the Experiment timestamp
- Sample Offset: the fractional portion of the UT time corresponding to the Experiment timestamp (specified in sample intervals).
- Experiment Timestamp: a timestamp in the experiment timescale that corresponds to the UT time specified by the UT Second and Sample Offset fields.

Section 1.02 RTCP Receiver Reports

Receivers generate Receiver Reports to inform other session members of the quality of their reception. Receiver reports include statistics on the fraction of packets lost, the cumulative number of packets lost and an approximation of the inter-arrival jitter for RTP data packets received at the receiver from a particular source.

Figure 6 shows the format of a Receiver Report Packet. It is essentially the same format as the Sender Report, but does not include the Sender Report block (UT timestamp, RTP timestamp) nor the experiment timescale synchronization fields (UT Second, Sample Offset, Experiment Timestamp). The first 32-bit word has the same semantics as the first 32-bit word in the Sender Report Packet. The Payload Type is set to 201 to indicate that this is an RTCP Receiver Report packet.

Section 1.03 RTCP Source Description Packets

Source Description (SDES) packets are used to describe the source of a particular stream of packet samples. The first level includes the header followed by a number of source description blocks. Each source description block (chunk) contains an SSRC that identifies the source channel stream described by the block, followed by a number of different items that describe the source.

Figure 7 shows the format of an SDES packet. Table 6 provides a detailed description of each of the fields within the packet.

Each SDES packet consists of source description "chunks". These consist of an SSRC identifier followed by a list of 0 or more items that describe the source. Each item consists of a type field, a length field which indicates the length of the text (not including the type and length bytes) followed by the text itself. Each chunk begins on a 32-bit word boundary. The following items are predefined:

- ***CNAME:*** Canonical endpoint Name Identifier. The CNAME is a constant identifier that is bound to each source for the duration of a session and is associated with the source's SSRC. It is unique across sources for the duration of a session.
- ***NAME:*** User Name SDES item. This is the real name associated with an endpoint. It is expected to be constant for the duration of a session, but is not necessarily unique amongst all participants.
- ***EMAIL:*** Electronic mail SDES item. Email address of source contact person.
- ***PHONE:*** Phone number SDES item. Phone number of source contact person.
- ***LOC:*** Geographical Location SDES item. Location of the source.
- ***TOOL:*** Application or tool name SDES item. Name of the `tool generating the stream.
- ***NOTE:*** Notice/Status SDES item. Transient packets describing the state of the source during a session.
- ***PRIV:*** Privation Extensions to SDES. Provides a mechanism to enable users to define application specific SDES packets.

This specification uses the PRIV packet type to add VLBI specific extensions to the SDES packet. Four additional message types are added, identified by their prefix string:

- ***Evlbi-abm:*** Active Bit-stream Mask - identifies the bit streams constituting the channel corresponding to this SDES packet.
- ***Evlbi-sfr***: Sampling FRequency - the sampling frequency of the channel samples.
- ***Evlbi-spp:*** Samples Per Packet - indicates how many packet samples are contained in a single RTP data packet.

This information can be used by receivers to determine the time between consecutive packet samples (that is, the rate at which the RTP timestamps are incrementing).

Section 1.04 RTCP BYE Packet

This packet is used to indicate that a source is leaving a session and is no longer active. It is distributed to all session participants to allow them to update their internal tables appropriately. In particular, it allows session participants to track the number of active sources, which is important for the calculation of RTCP bandwidth.

Figure 9 shows the format for an RTCP BYE packet. Table 7 describes each of the fields in detail.

Section 1.05 RTCP Application Defined Packet

The Application-defined RTCP Packet is used to communicate other VLBI control information. In the VSI-E profile, currenly only one subtype is defined:

- type 1: is used to transfer PDATA between DIMs and DOMs.

Figure 10 shows a detailed view of the Application-defined RTCP packet. Table 8 describes each of the fields in detail.

The Application defined packet may also be used to implement congestion and flow control functionality. This functionality is for further study.

Section 1.06 RTCP Packet Transmission

Section 6.1 of [16] describes the transmission of RTCP packets. We recommend the adoption of the guidelines in this section.

In this section, we give a brief overview of the guidelines outlined in section 6.1 of [16].

RTCP packets can be transmitted individually or concatenated together into ***compound packets***. Compound packets are recommended as they reduce the lower layer protocol overhead. There are certain recommendations on the contents of compound packets. In particular:

1. Each transmitted RTCP packet should include a Report packet (either Sender Report or Receiver Report).
2. Each compound packet should include an SDES CNAME.
3. All RTCP packets must be sent in a compound packet of at least two individual packets with the following recommended format:
    a. SR or RR: first RTCP packet must always be a report packet
    b. Additional RRs: if the number of sources is greater than 31 then additional RR packets should follow the initial report packet
    c. SDES: an SDES packet including a CNAME should be included in every RTCP compound packet
    d. BYE or APP: other RTCP packet types may follow in any order.

Figure 11 shows some examples of compound RTCP packets (based on Figure 1 of [16]). Part (a) of the figure shows a simple Compound RTCP packet. At the start of the packet is an SR with a sender report block followed by receiver statistics for four sources. An SDES packet that contains a CNAME (and may contain other attributes, e.g. PHONE and LOC items) and a "PRIV" chunk (containing a VLBI packet that contains the Active Bit-stream Mask for an active source) follows the SR or RR. All of these RTCP packets are included in a single lower layer protocol (in this case TCP is used).

The compound packet shown in part (b) has a similar structure to the one in part (a). The major differences are that the first RTCP packet within the compound packet has an RR packet and an APP packet has been added to carry some PDATA.

Part (c) of Figure 11 shows an important case: namely, the structure of a compound RTCP packet when the sender of the packet needs to report statistics for more than 31 senders (a maximum of 31 sender statistics blocks can be sent in a single RTCP Receiver Report packet). In this case, three RR's are sent to report the reception statistics for 64 unique senders. Note that a similar format could be used with SR's instead of RR's. An SDES packet is also included.

Section 1.07 Additional Capabilities of RTP and RTCP

This version of the VSI-E specification uses a small subset of the total functionality that is included in the RTP/RTCP specification[16]. RTP/RTCP originally was designed for use with audio/video conferencing type applications in which participating hosts would be both transmitting and receiving information. As such, much of the functionality built in to RTP/RTCP has been designed to meet the requirements of applications such as these. Fortunately for us, much of this functionality is also applicable to VSI-E.

RTP has the concept of mixers and translators. These are intermediate systems that take RTP streams and manipulate them in some way. A mixer takes multiple streams in and produces a single output stream that is the combination of all of these input streams. It also generates its own timing information. Thus, the mixer will become the SSRC for the new stream that is generated. In order to track the original sources that contributed to this new aggregate stream, the Contributing Source field or CSRC is used. In this version of the VSI-E specification, mixers are not considered (although future use is by no means precluded). Thus, the CSRC is not used in this version of the VSI-E specification.

A translator generates a single output RTP stream from a single RTP input stream. It does not modify the SSRC. However, translators may convert data encodings without mixing. In this version of the VSI-E specification translators are not considerd, however, their future use is not precluded.

Appendix B. RTCP Transmission Interval[9]

RTCP has been designed specifically to ensure that its bandwidth usage is scalable as the number of session participants increases. In particular, it is recommended in [16] that RTCP use no more than 5% of the session bandwidth. This is loosely defined in [16] as:

> “For each session, it is assumed that the data traffic is subject to an aggregate limit called the "session bandwidth" to be divided among the participants. This bandwidth might be reserved and the limit enforced by the network, or it might just be a reasonable share.”

A more precise definition is given in [17]:

> “The session bandwidth is the nominal data bandwidth plus the IP, UDP and RTP headers (40 bytes). For example, for 64 kb/s PCM audio packetized in 20 ms increments, the session bandwidth would be (160 + 40) / 0.02 bytes/second or 80 kb/s. If there are multiple senders, the sum of their individual bandwidths is used.”

The last line of this is applicable to e-VLBI in which there are N telescope sites sending data at rate R Mbps. In this case, the total session bandwidth is N x R. The session bandwidth is typically supplied to the RTP/RCTP application at startup.

The following bandwidth limitations apply to RTCP bandwidth usage (for more information refer to section 6.2 of [16])

1. Session bandwidth $B_{SESSION}$ = N x R [bps][10]
2. Total RTCP bandwidth $B_{RTCP} < 5\%$ of $B_{SESSION}$ [bps]
3. Total SDES bandwidth $B_{SDES} < 25\%$ of B_{RTCP} [bps]
4. Time between RTCP packets $T_{RTCP} > T_{RTCPMIN}$

Note that $T_{RTCPMIN}$ may be either 5 s or 360/ $B_{SESSION}$ where $B_{SESSION}$ is in kilobits per second. So, for a session bandwidth of 10 gigabits per second or $10\text{x}10^6$ kbps, $T_{RTCPMIN}$ is 36 us.

Appendix A includes sample C-code from [16] that implements the RTCP interval generation. Once the interval has been calculated , it is randomized to prevent the synchronization of RTCP packets from multiple sources. In particular,

$$T_{RTCP} = \max\left(\square\square T_{calc}, T_{RTCPMIN}\right) \text{ [s]} \qquad \textbf{E. 1}$$

where T_{RTCP} is the RTCP transmission interval, T_{CALC} is the calculated transmission interval, □ is a random number in the interval [0.5, 1.5], and $T_{RTCPMIN}$ is the minimum interval between RTCP packets.

The RTCP interval for the first packet of a source’s session is:

---

[10] Note that the session bandwidth may be specified as a separate parameter to the applications.

$$T_{RTCP} = \square\square T_{RTCPMIN} \text{ [s]} \qquad \textbf{E. 2}$$

where □is a random number in the interval [0.5, 1.5].

Appendix C. RTCP Transmission Interval Code

The following code is taken from Appendix A.7 of RFC3550 [16].

```
double rtcp_interval(int members,
                     int senders,
                     double rtcp_bw,
                     int we_sent,
                     double avg_rtcp_size,
                     int initial)
{
    /*
     * Minimum average time between RTCP packets from this site (in
     * seconds).  This time prevents the reports from `clumping' when
     * sessions are small and the law of large numbers isn't helping
     * to smooth out the traffic.  It also keeps the report interval
     * from becoming ridiculously small during transient outages like
     * a network partition.
     */
    double const RTCP_MIN_TIME = 5.;
    /*
     * Fraction of the RTCP bandwidth to be shared among active
     * senders.  (This fraction was chosen so that in a typical
     * session with one or two active senders, the computed report
     * time would be roughly equal to the minimum report time so that
     * we don't unnecessarily slow down receiver reports.)  The
     * receiver fraction must be 1 - the sender fraction.
     */
    double const RTCP_SENDER_BW_FRACTION = 0.25;
    double const RTCP_RCVR_BW_FRACTION = (1-RTCP_SENDER_BW_FRACTION);
    /*
    /* To compensate for "timer reconsideration" converging to a
     * value below the intended average.
     */
    double const COMPENSATION = 2.71828 - 1.5;

    double t;                   /* interval */
    double rtcp_min_time = RTCP_MIN_TIME;
    int n;                      /* no. of members for computation */

    /*
     * Very first call at application start-up uses half the min
     * delay for quicker notification while still allowing some time
     * before reporting for randomization and to learn about other
     * sources so the report interval will converge to the correct
     * interval more quickly.
     */
    if (initial) {
        rtcp_min_time /= 2;
    }
    /*
     * Dedicate a fraction of the RTCP bandwidth to senders unless
     * the number of senders is large enough that their share is
     * more than that fraction.
     */
    n = members;
    if (senders <= members * RTCP_SENDER_BW_FRACTION) {
        if (we_sent) {
            rtcp_bw *= RTCP_SENDER_BW_FRACTION;
            n = senders;
        } else {
            rtcp_bw *= RTCP_RCVR_BW_FRACTION;
            n -= senders;
        }
    }
```

```
    /*
     * The effective number of sites times the average packet size is
     * the total number of octets sent when each site sends a report.
     * Dividing this by the effective bandwidth gives the time
     * interval over which those packets must be sent in order to
     * meet the bandwidth target, with a minimum enforced.  In that
     * time interval we send one report so this time is also our
     * average time between reports.
     */
    t = avg_rtcp_size * n / rtcp_bw;
    if (t < rtcp_min_time) t = rtcp_min_time;

    /*
     * To avoid traffic bursts from unintended synchronization with
     * other sites, we then pick our actual next report interval as a
     * random number uniformly distributed between 0.5*t and 1.5*t.
     */
    t = t * (drand48() + 0.5);
    t = t / COMPENSATION;
    return t;
}
```

13. Acronyms

This specification uses the following acronyms:

BPS	Bits Per channel Sample
BSIR	Bit Stream Information Rate
CBM	Channel Bit Mask
CNAME	Canonical Name
DCCP	Datagram Congestion Control Protocol
DIM	Data Input Module
DOM	Data Output Module
DOT	Data Observe Time
DTS	Data Transmission System
EGAE	Experiment Guided Adaptive Endpoint
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	Initial Sample Offset
NTP	Network Time Protocol
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SDES	Source Description
SFR	Sampling Frequency
SO	Sample Offset
SPP	Samples Per Packet
TCP	Transmission Control Protocol
TVG	Test Vector Generator
UDP	User Datagram Protocol
UT	Universal Time
VSI-E	VLBI Standard Interface - Electronic
VSI-H	VLBI Standard Interface - Hardware
VSI-S	VLBI Standard Interface - Software

14. References

1. *VLBI Standard Hardware Interface Specification - VSI-H* available: http://web.haystack.edu/vsi/2002_12_12_vsi-h_draft_rev_1.1.pdf, 2002.
2. *VLBI Standard Software Interface Specification - VSI-S* available: http://web.haystack.edu/vsi/2003_02_13_vsi-s_final_rev_1.pdf, 2003.
3. Schulzrinne, H., *RTP Tools* available: http://www.cs.columbia.edu/~hgs/rtp/, 2003.
4. Schulzrinne, H., *RTP Libraries* available: http://www.cs.columbia.edu/~hgs/rtp/, 2003.
5. H. Schulzrinne, *RFC 1890: RTP Profile for Audio and Video Conferences with Minimal Control* available: http://www.ietf.org/rfc/rfc1890.txt?number=1890, IETF, 1996.
6. J. Wroclawski, a.o., *An RTP Profile for e-VLBI Data* available: 2003.
7. Postel, J., *RFC768: User Datagram Protocol* available: http://www.ietf.org/rfc/rfc0768.txt?number=768, IETF, 1980.
8. Postel, J., *RFC793: Transmission Control Protocol DARPA Internet Program Protocol Specification* available: http://www.ietf.org/rfc/rfc0793.txt?number=793, IETF, 1981.
9. Eddie Kohler, M.H., Sally Floyd, Jitendra Padhye, *INTERNET-DRAFT Datagram Congestion Control Protocol* available: http://www.ietf.org/internet-drafts/draft-ietf-dccp-spec-04.txt, IETF, 2003.
10. K. Nicols, S.B., F. Baker, D. Black, *RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers* available: http://www.ietf.org/rfc/rfc2474.txt?number=2474, IETF, 1998.
11. Jamshid Mahdavi, S.F., *TCP-Friendly Unicast Rate-based Flow Control* available: http://www.psc.edu/networking/papers/tcp_friendly.html, 1997.
12. Alan Whitney, J.W., *Experiment Guided Adaptive Endpoint: NSF Grant # ANI-0230759*. 2002, MIT: Boston. p. 17.
13. David Lapsley, J.W., Alan Whitney, *Development of new IP Protocols for e-VLBI* available: http://web.haystack.mit.edu/e-vlbi/lapsley_dwingeloo_03_files/frame.htm, MIT Haystack Observatory, 2003.
14. Postel, J., *RFC791: Internet Protocol* available: http://www.ietf.org/rfc/rfc0791.txt?number=791, IETF, 1981.
15. Mills, D.L., *RFC 1305: Network Time Protocol (Version 3) Specification, Implementation and Analysis* available: http://www.ietf.org/rfc/rfc1305.txt?number=1305, IETF, 1992.
16. H. Schulzrinne, S.C., R. Frederick, V. Jacobson, *RFC 1889: RTP: A Transport Protocol for Real-Time Applications* available: http://www.ietf.org/rfc/rfc1889.txt?number=1889, IETF, 1996.
17. Schulzrinne, H., *RTP Frequently Asked Questions* available: http://www.cs.columbia.edu/~hgs/rtp/, 2003.