Draft Proposal for VSI-E - Rev 2.7

29 January 2004

Table of Contents

1. INTRODUCTION ... 4

2. ASSUMPTIONS ... 4

3. GOALS ... 4

4. DATA FORMATTING ... 5

4.1 'CHANNEL' AND 'CHANNEL SAMPLE' DEFINITIONS ... 5
4.2 PACKETIZATION BY USER CHANNEL ... 5
4.3 CHANNEL PACKETIZATION ... 6
4.4 PACKET DATA ARRAY FORMAT ... 6
4.5 ENDIAN ... 6

5. WHY RTP FOR E-VLBI? ... 6

6. NETWORK TOPOLOGIES ... 7

7. RTP ... 8

8. RTCP ... 8

9. UNDERLYING NETWORK TRANSPORT ... 8

10. AN RTP PROFILE FOR E-VLBI ... 9

10.1 RTP DATA ENCAPSULATION ... 10
10.2 RTCP ... 11
10.3 RTP TIMESTAMP SYNCHRONIZATION ... 12
10.4 EXAMPLE RTCP MESSAGE FLOW ... 13
10.5 RTP FLOW CONTROL ... 14
10.6 ENDIAN AND BIT-NUMBERING CONVENTIONS ... 14

11. IMPLEMENTATION ISSUES ... 15

11.1 TIME SYNCHRONIZATION ... 15
11.2 SENDER REPORT GENERATION ... 16
11.3 RTP TIMESTAMP WRAP-AROUND ... 16

12. TABLES ... 17

13. FIGURES ... 23

APPENDIX A. RTP TIMESTAMP SCALING ... 38

SECTION 1.01 INTRODUCTION ... 38
SECTION 1.02 DEFINITIONS ... 38
SECTION 1.03 SCALING BIT ... 39
SECTION 1.04 TIMESTAMP SCALING FACTOR ... 40
SECTION 1.05 SYNCHRONIZATION OF THE TIMESTAMP SCALING FACTOR ... 40
SECTION 1.06 SCALING OPERATION ... 40

APPENDIX B. RTCP ... 41

SECTION 2.01 RTCP SENDER REPORTS ... 41
SECTION 2.02 RTCP RECEIVER REPORTS ... 41
SECTION 2.03 RTCP SOURCE DESCRIPTION PACKETS ... 41
SECTION 2.04 RTCP BYE PACKET ... 42
SECTION 2.05 RTCP APPLICATION DEFINED PACKET ... 43
SECTION 2.06 RTCP PACKET TRANSMISSION ... 43

SECTION 2.07 ADDITIONAL CAPABILITIES OF RTP AND RTCP ... 44
APPENDIX C. RTCP TRANSMISSION INTERVAL ... 45
APPENDIX D. RTCP TRANSMISSION INTERVAL CODE ... 47
14. ACRONYMS ... 49
15. REFERENCES ... 50

List of Figures

FIGURE 1: PACKET DATA ARRAY FORMAT ... 23
FIGURE 2: VSI-E RTP/RTCP OVERVIEW ... 24
FIGURE 3 VSI-E NETWORK SCENARIOS ... 25
FIGURE 4: RTP DATA PACKET ... 26
FIGURE 5: RTCP SENDER REPORT PACKET ... 27
FIGURE 6: RTCP RECEIVER REPORT PACKET ... 28
FIGURE 7: RTCP SOURCE DESCRIPTION PACKET STRUCTURE ... 29
FIGURE 8: RTCP E-VLBI SOURCE DESCRIPTION ITEMS ... 30
FIGURE 9: RTCP BYE PACKET ... 31
FIGURE 10: APPLICATION-DEFINED RTCP PACKET ... 31
FIGURE 11: COMPOUND RTCP PACKETS ... 32
FIGURE 12: VSI-H MODEL ... 33
FIGURE 13: RTCP TIMESTAMP SYNCHRONIZATION ... 34
FIGURE 14: INITIAL SAMPLE SYCHRONIZATION ... 35
FIGURE 15: EXAMPLE RTCP MESSAGE SEQUENCE ... 36
FIGURE 16 RTP HEADER WITH SCALING BIT ... 39

List of Tables

TABLE 1: VLBI RTP PAYLOAD TYPES ... 17
TABLE 2: E-VLBI PARAMETER TRANSPORT ON RTP/RTCP ... 17
TABLE 3: RTP DATA FORMAT ... 18
TABLE 4: RTCP SENDER REPORT FORMAT ... 20
TABLE 5: RTCP RECEIVER REPORT FORMAT ... 20
TABLE 6: RTCP SDES PACKET ... 21
TABLE 7: RTCP BYE PACKET ... 21
TABLE 8: APPLICATION-DEFINED RTCP PACKET ... 22
TABLE 9 RTP TIMESTAMP WRAP-AROUND TIMES ... 38

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 DTSs 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 "on the wire" from DIM to DOM and that is compatible between otherwise heterogeneous DTSs. 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 'Channel' and 'Channel Sample' Definitions

The following definitions are made:

1. A ***channel*** is defined as an exclusive subset of 2^n of the active bit streams. The intent of the channel abstraction is that it carry the digitized data from a single analog data source.
2. A ***channel sample*** is defined as 2^n bits collected from a single 'channel' on a single DIM CLOCK cycle[1]. The DIM collects channel samples at the Bit-Stream Information Rate (BSIR).
3. A ***channel stream*** is defined as a contiguous set of channel samples collected over some period of time.

4.2 Packetization by User Channel

RTP allows each channel to be segregated into a separate packet stream, which has several powerful advantages:

- ***Flexibility:*** The packet stream from each channel 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 channel streams being sent to that destination.
- ***Scalability:*** Specified channel 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[2].
- ***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.

The following rule applies:

[1] The 'channel sample' may or may not correspond to an actual signal sample; for purposes of VSI-E we define the selected subset of 2^n selected bit streams as a "channel". For backwards compatibility with the limited capabilities of some older systems not able to separate the data into actual physical channels for e-VLBI transmission, the set of all active bit streams may be chosen to be a "channel" in the VSI-E sense.

[2] 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.

- Each channel is made up of 2^n exclusive bit-stream inputs taken from the set of active DIM input bit streams.

For example, 16 active bit-streams entering a DIM might represent 8 channels, each represented by 2 bit-streams. Different channels may contain different numbers of bit-streams, though this would not normally be the case.

4.3 Channel Packetization

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

1. Each channel-stream is divided into a sequential set of packets, each containing a fixed-length ***data payload*** which consists of sequential channel samples from the channel.
2. The data-payload array contains an integer number of 32-bit words, each of which contains one or more channel samples from one channel.

Each channel-stream may have a different data payload length, if desired.

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 more sequential channel samples from one channel. 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 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 are transmitted in little endian format (least significant bytes are transmitted first). Should big endian formats become important in the future, they could easily be implemented by defining additional RTP payload types with the data in big-endian format. All control information is transmitted in network byte or "big endian" order for consistency with the RTP protocol [3].

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[3] 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

[3] 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.

- Dissemination of session information
- Monitoring of network and end system performance (by participants and third parties)
- 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 [4, 5]. 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 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 3b 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 different 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[4].

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 [6]. Wroclawski et. al[7] 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, a single RTCP stream will be used to communicate control information between a sender and its receiver(s). In this case, a "sender" refers to a computational process or thread that generates RTP packets corresponding to a particular "channel" (see §4.1). It is possible that a single host computer may have multiple "sender" processes active at one time and hence have multiple RTCP streams associated with it.

RTCP is also used to transmit PDATA information, which will be used to transmit various important auxiliary information, such as station, experiment and source IDs, and other information necessary or useful for later processing.

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) [8]
- Transmission Control Protocol (TCP) [9]
- Datagram Congestion Control Protocol (DCCP) [10]

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

- User Datagram Protocol with Application Level flow control[5]

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 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 on regular users' 'best effort' data. For further information, see [11].
- ***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 [12].

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

One of the strengths of the VSI-E proposal is that the same RTP-based structure can be used with different transport protocols to meet the needs of different networking environments. For example, UDP transport may be more suitable for dedicated networks, while TCP transport may be more suitable for shared networks. Guidelines and profiles for various transport alternatives will be provided in the near future.

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[6], source identifier and data samples.
- ***RTCP Sender Report Packet:*** used to allow sources (antennas) to distribute transmission statistics and define the relationship between UT and RTP packet sequence number. Statistics include the sender's packet and octet count. Also included is an RTP packet sequence number and a corresponding 64-bit UT timestamp.

---

[5] 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.

[6] The RTP timestamp is a 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. The VSI-E specification allows for the use of RTP timestamp scaling (as discussed in Appendix A) to minimize the impact of timestamp "wrap-around" at high data transmission rates.

- ***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. 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 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:
    - an ***Active Bit Stream Mask (ABM)*** which identifies the bit streams constituting the channel corresponding to a particular SDES packet.
    - a ***Channel IDentifier: (CID)*** which identifies which VLBI channel was the origin for the samples in this packet. This provides the mapping between a channel stream and its corresponding analog source.
    - 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 channel samples within a packet
    - a ***Timestamp Scaling Factor(TSF)*** field which is used to support the RTP timestamp scaling mechanism defined in Appendix A.
- ***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. One subtype is currently defined:
    - ***PDATA Packet:*** used for transmitting PDATA. This packet includes: a 64-bit UT time stamp, a source identifier and PDATA(which is transported as an ASCII string).

Figures 4-10 show the formats for these 5 packet types. See [6] 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 is chosen to be 'active bit streams'. The 'active bit streams' are further subdivided into some number of mutually exclusive channels, each sample of which is a ***channel sample***. A sequential set of channel samples from a single channel 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.

10.2 RTCP

The RTP Control Protocol (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 C.

There are five different types of RTCP packets:

- ***Sender Reports:*** used to transmit quality of services statistics for transmitters; also used to provide timing synchronization.
- ***Receiver Reports:*** used to transmit quality of service statistics from receiver(s) back to the sources (allows the sources to adjust their transmission strategies based on this feedback).
- ***Source Description Packets:*** used to transmit Active Bitstream Mask, Channel Identifier, Sampling Frequency, Samples Per Packet and Timestamp Scaling Factor. This information allows the receiver(s) to calculate: the BSIR[7] and packet sizes.
- ***BYE Packets:*** used to signal that a sender or multiple senders are going inactive and leaving a session.
- ***Application Defined Packets:*** asynchronous PDATA, and other control signals.

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 C and Appendix D. 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, a Source Description packet and an Application Defined Message. The Sender Report contains an RTP timestamp and its corresponding 64-bit UT timestamp for synchronization purposes. The Source Description packet contains the following items:

- CNAME: mapping from the source's Synchronization Source identifier to its canonical identifier
- A PRIV[8] message with an EVLBI-ABM field: Active Bit Stream Mask or list of channels that are active for this channel stream

[7] Bit Stream Information Rate, as defined in: [1].

[8] A PRIV message is a special type of SDES message that allows applications to define their own PRIVate message types for the exchange of application specific messages. An

- PRIV message with an EVLBI-CID field: Channel IDentifier used to indicate which channel was the source of this data stream.
- PRIV message with an EVLBI-SFR field: Sampling Frequency.
- PRIV message with an EVLBI-SPP field: number of channel Samples Per Packet.
- PRIV message with an EVLBI-TSF field: RTP Timestamp Scaling Factor.

This information allows a receiver to calculate the value of Bit Stream Information Rate (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 C and Appendix D. 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 RTP Timestamp Synchronization

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

- ***periodic RTCP Sender Report Packet:*** contains an RTP timestamp and its corresponding 64-bit UT timestamp which provides a synchronization point that receiver(s) can use to convert between RTP packet sequence numbers and UT. Sender Reports are transmitted in a quasi-periodic fashion.
- ***per-packet RTP timestamp:*** a counter that is incremented once per RTP sample (each channel packet stream maintains its own set of RTP sequence numbers). The VSI-E specification RECOMMENDS the use of the Timestamp Scaling Factor (TSF) parameter to ensure that the RTP timestamp is only incremented once per RTP packet[9].

By making use of the periodic[10] RTCP Sender Report packets (which act as synchronization packets), the per-packet sequence number and the known sampling rate, the destination(s) is able to calculate an explicit timestamp for each data channel sample that is received.

Figure 13 illustrates the RTP 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

---

application can define many different message sub-types within a PRIV message by specifiying different "prefix-strings" within the PRIV message (refer to Figure 8).

[9] Refer to Appendix A for a full discussion of this parameter and the mechanism used to scale RTP timestamps.

[10] 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.

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 SDES report. This contains the TSF that is used to scale the RTP timestamp (as described in Appendix A). In this case, the TSF factor is equal to the packet size of 72,000 samples. This means that the value in the 32-bit RTP timestamp field in the RTP data packet header and RTCP Sender Report is incremented once per packet. This helps to alleviate the problem of “wrap-around” of this field at high speeds. The SDES packet also contains the number of Samples Per Packet and the Sampling FRequency which are required to work out an explicit timestamp for each sample.

The DIM then transmits an RTCP sender report that contains a 64-bit UT timestamp and an RTP timestamp. 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 RTP timestamp in the data packet can be converted to UT by using the synchronization reference point provided by the Sender Report (UT Timestamp and RTP timestamp). By using the difference between the RTP timestamp of the current data packet and the RTP timestamp in the Sender Report, and by using the known sampling frequency and packet size, the UT time of any given sample may be determined (see also §11.1). At the DIM side, succeeding RTP packets have their RTP timestamp each incremented by 1. Note that a receiver can “join” in to an RTP “broadcast” at any time after the start of data transmission. The initial timing synchronization sequence is a special case that allows the sender and receiver to synchronize their time scales without the loss of data.

Figure 14 shows the initial lossless timing synchronization exchange. Initially, an RTCP Sender Report is sent with a UT timestamp (UTS) and the corresponding RTP Timestamp(RTS). 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.

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 intial synchronization point. The packet consists of a Sender Report packet and a Source Description packet. The Sender Report contains timestamp synchronization information (RTP timestamp and UT timestamp). 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 Bitstream Mask (“evlbi-abm”), Channel Identifier(“evlbi-cid”), Sampling FRequency(“evlbi-sfr’), Samples Per Packet ("evlbi-spp"), and Timestamp Scaling Factor("evlbi-tsf"); 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) [13, 14]. 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

Because the vast majority of the platforms on which this data will be manipulated are Intel-derived platforms, all data must be transmitted in little endian format (least significant bytes are transmitted first). Should big endian formats become important in the future, they could easily be implemented by defining additional RTP payload types with the data in big-endian format.

All RTP and RTCP control fields, including e-VLBI extensions, must be transmitted in network byte or "big endian" order. This is to maintain consistency with the RTP specification.

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 [15]. "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 of high bandwidth data streams at the end systems.

11. Implementation Issues

11.1 Time Synchronization

The 64-bit UT timestamp transported in Sender Reports and used to provide the destination with the relationship between UT and the RTP timestamp timescale is represented as a 64-bit unsigned fixed-point number, with the first 32 bits representing the integer number of seconds and the least significant bits representing the fractional portion of a second to an accuracy of 2^{-32} s (~ 2.328×10^{-10} s). While this is quite accurate, it is not able to exactly represent all possible sample times, which are sampled at base-10 frequencies (e.g. 32 Mhz). For example, assume that the sampling frequency is 32 MHz and is generating a bit stream of 32 Mbps. The time between each sample is 3.125×10^{-8} s. The RTP timestamp clock increments at a multiple of this basic period. Conceptually, a packet could be transmitted at any timestamp interval and we would need to represent this transmission time. However, the closest this number can be represented by the UT timestamp is: 5.9606×10^{-8} on the upper side or 3.1199×10^{-8} s on the lower side. These two values represent a proportional error of 90.7% and 0.163% respectively, which are not insignificant errors. At higher frequencies, these errors become more significant.

In order to eliminate these errors completely, the VSI-E specification recommends that the transmission time of sender reports be selected so that the UT timestamp exactly represents the sample time corresponding to the RTP timestamp. This is quite an easy thing to do. A naive scheme for example, might transmit Sender Reports only at times that are integral multiples of half second boundaries (e.g. 0.0, 5.5, 10.5, 15.5, …). While this would eliminate representational errors in the UT timestamp, it would reduce the degree of randomness in the timing of RTP Sender Report transmission. The RTCP protocol requires Sender Reports to be transmitted in a pseudo-random manner with a constant mean time between transmissions. This helps to reduce the occurrences of synchronization, whereby a large number of participants in an RTP session could potentially send their Sender Reports at the same time and cause quite large bursts of traffic.

The VSI-E specification requires that finer grained intervals be chosen to allow a reasonable degree of randomness in the transmission of a Sender Report. As an example, at 32MHz sampling, there exist 2048 discrete values within a second at which the UT timestamp corresponding to a particular RTP timestamp (sample) can be represented exactly. So one possible sequence could be: 0.0, 5.0019043, 10.999023, etc.

11.2 Sender Report Generation

One feature of e-VLBI is the many modes of operation: e-VLBI data can be transmitted in real-time, near real-time, delayed transmission or a combination of modes. Data may be buffered at the sender, receiver or both. In order to ensure that experimental timing information is preserved while maintaining real-time network statistics, care must be taken with the generation and transmission of RTP Sender Reports. The VSI-E specification recommends the following:

1. Sender Reports be recorded with UT timestamps and RTP timestamps at the same time data is captured.
2. Immediately prior to the transmission of a Sender Report, transmission statistics from the outgoing interface be recorded into the Transmission block of a sender report.
3. Upon the transmission of a Sender Report, the transmitting process should store the wall-clock time of transmission, along with the RTP timestamp in the Sender Report in a table, which can then be used to calculate round trip times, etc. upon the receipt of a Receiver Report with the RTP timestamp of the Sender Report in the Last Sender Report (LSR) field.

Provided these guidelines are followed, it should be possible to preserve experimental timing information over the wide variety of e-VLBI operational modes, while at the same time maintaining real-time network statistics.

11.3 RTP Timestamp Wrap-around

The 32-bit RTP timestamp in the header of data packets can represent 2^{32} distinct values (4,294,967,296). In the original RTP standard, the RTP timestamp must be incremented by 1 for every transmitted sample. However, at high speeds this can cause rapid wrap-around of the RTP timestamp. For example, assuming a line transmission rate of 10 Gbps, 28 bytes of overhead (RTP+UDP+Ethernet) per packet, 1500 byte MTUs and 11,776 data samples per packet the wraparound time for the 32-bit RTP timestamp is ~ 0.438 seconds. Thus, a burst error that causes more than 0.438 seconds worth of data to be lost will result in the loss of timing synchronization between the source and the destination. In order to avoid this, the VSI-E specification recommends the use of the RTP Timestamp Scaling Factor (TSF) extension, as defined in Appendix A, to scale the RTP timestamp down at the sender and up at the receiver. This effectively reduces the rate of wrap-around by a factor equal to the Timestamp Scaling Factor. So, for example, a TSF of 11,776 will increase the wrap around time in the example above to ~ 5154 s. This is more than sufficient to avoid loss of synchronization due to burst errors as it is reasonable to expect that the duration of burst errors in any functioning, reasonably congested network will be orders of magnitude smaller in duration. The VSI-E specification requires that a Timestamp Scaling Factor equal to the number of sample per packet be used in order to increase the RTP timestamp wraparound time.

12. Tables

VPT Value	Description
00000	1 bit per channel sample
00001	2 bits per channel sample
00010	4 bits per channel sample
00011	8 bits per channel sample
00100	16 bits per channel sample
00101	32 bits per channel sample
00110 - 111111	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 Channel Bit Mask item.
DOT time-tagging	Carried in the RTP header as a 32-bit scaled RTP timestamp. Synchronization with UT provided by RTCP Sender Report packets that contain a 64-bit UT 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 channel sample.
Valid-data indicator	Carried in the RTP Data Packet header as a single bit (I-bit).
TVG-data indicator	Carried in the header of RTP data packets (T-bit).
PDATA messages	Carried using the Application Defined RTCP Packet. This carries the PDATA as well as a 64-bit NTP (UT) timestamp.

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	1: invalid data samples in this packet; 0: valid samples in this packet
T	test vector bit. A value of 1 indicates that the data in the packet payload is a test vector. A value of 0 indicates that the data consists of channel samples (the validity of which is subject to the 'I' bit defined above).
S	RTP timestamp scaling bit. A value of 1 indicates that the RTP timestamp has been scaled by a factor supplied in the SDES message A value of 0 indicates that the RTP Timestamp has not been scaled. Refer to Appendix A for a full description.
VPT	VLBI Payload Type – number of bits per channel in the payload data. See Table 1
RTP timestamp	32 bit integer that is initialized to a random number and incremented by one for every packet transmitted. If the S-bit is set to 1, then the RTP timestamp is scaled by the TSF factor transported in SDES messages.
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.
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.
NTP timestamp, msb	32 most significant bits of the 64-bit UT timestamp corresponding to the time at which this packet was transmitted. The 64-bit NTP timestamp is an unsigned integer that represents the number of seconds relative to 0h UTC on 1 January 1900 [16]. The 32 most significant bits represent the integer number of seconds..
NTP timestamp, lsb	32 least significant bits of the 64-bit UT timestamp. These represent the fractional part of the 64-bit extended UT timestamp.
RTP timestamp	this is the scaled RTP timestamp that corresponds to the NTP 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

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 channel stream. This field is in little endian order.
evlbi-cid	Channel Identifier: this indicates which signal channel the sample bits in this packet belong to. 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 samples are contained in each packet of this channel stream. This field is in little endian order.
evlbi-tsf	Timestamp Scaling Factor: this is the amount by which the sender and receiver scale RTP timestamps. The VSI-E specification requires that the TSF be equal to the number of samples per packet. This ensures that the RTP timestamp is incremented by 1 per packet. This alleviates the problem of rapid wrap around of the RTP timestamp at high data transmission speeds. Refer to Appendix A for a full description of the timestamp scaling mechanism.

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 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.
NTP timestamp, msb	32 most significant bits of the 64-bit UT timestamp corresponding to the PDATA timestamp. See Table 4. This field is in little endian format.
NTP timestamp, lsb	32 least significant bits of the 64-bit UT timestamp that correspond to the fractionaly portion of the timestamp. See Table 4. 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

13. Figures

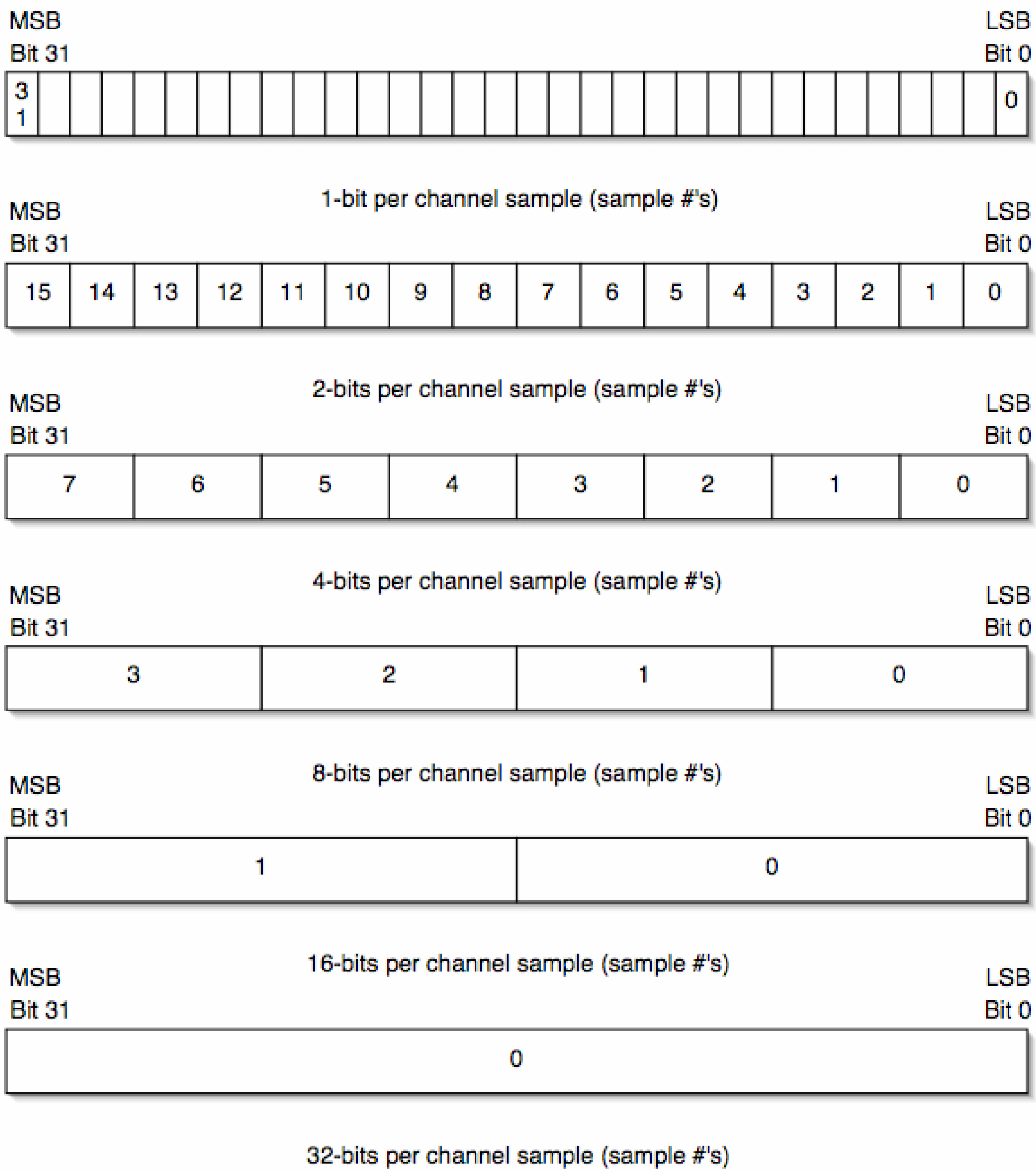


Figure 1: Packet Data Array Format

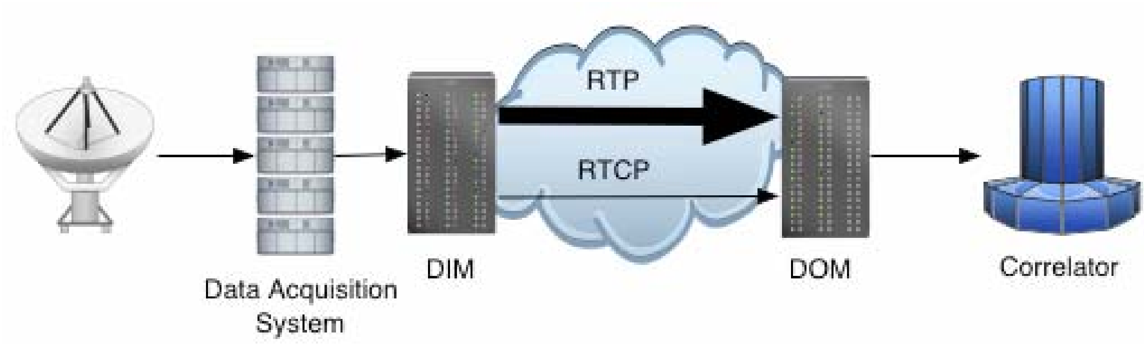


Figure 2: VSI-E RTP/RTCP Overview

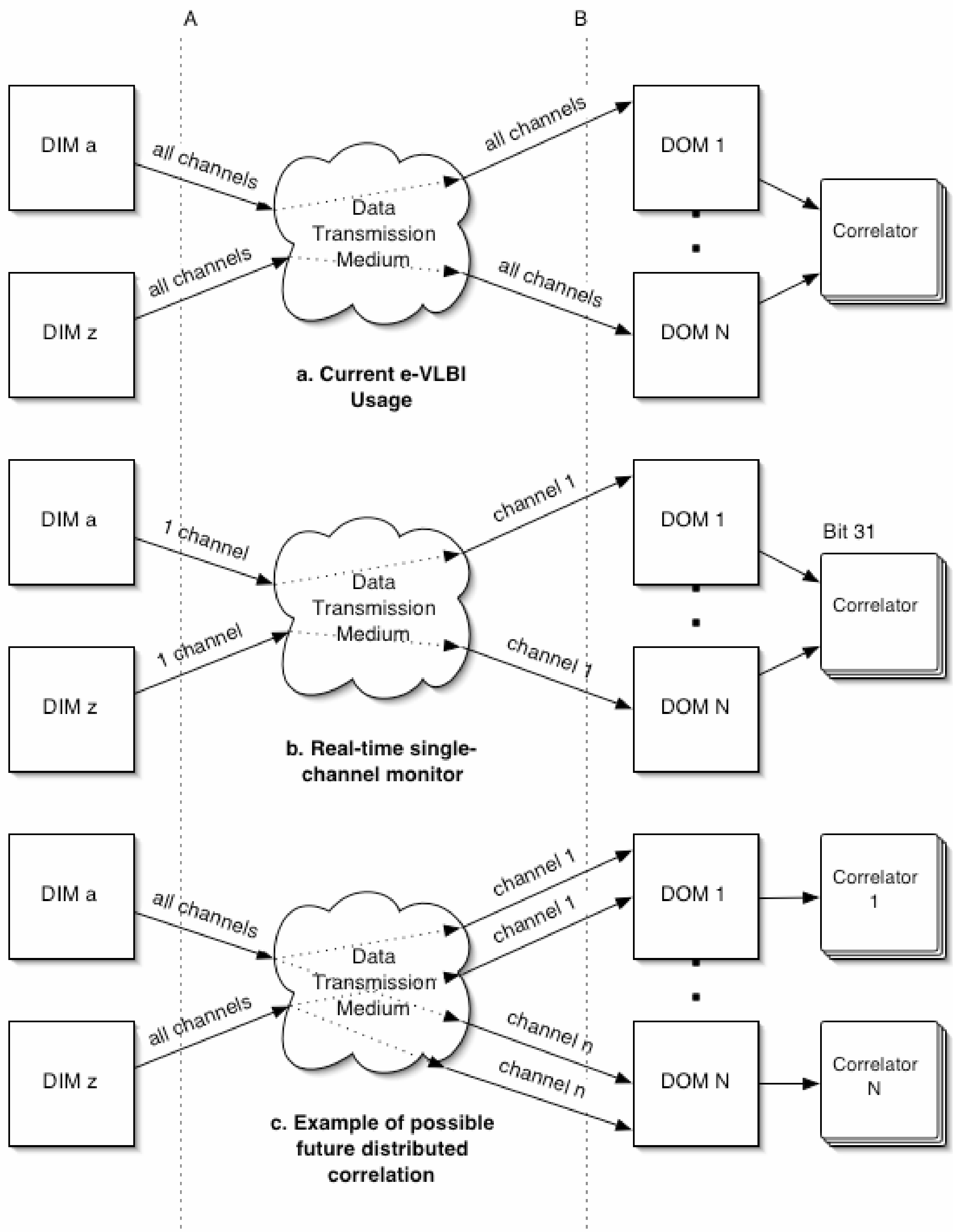


Figure 3 VSI-E Network Scenarios

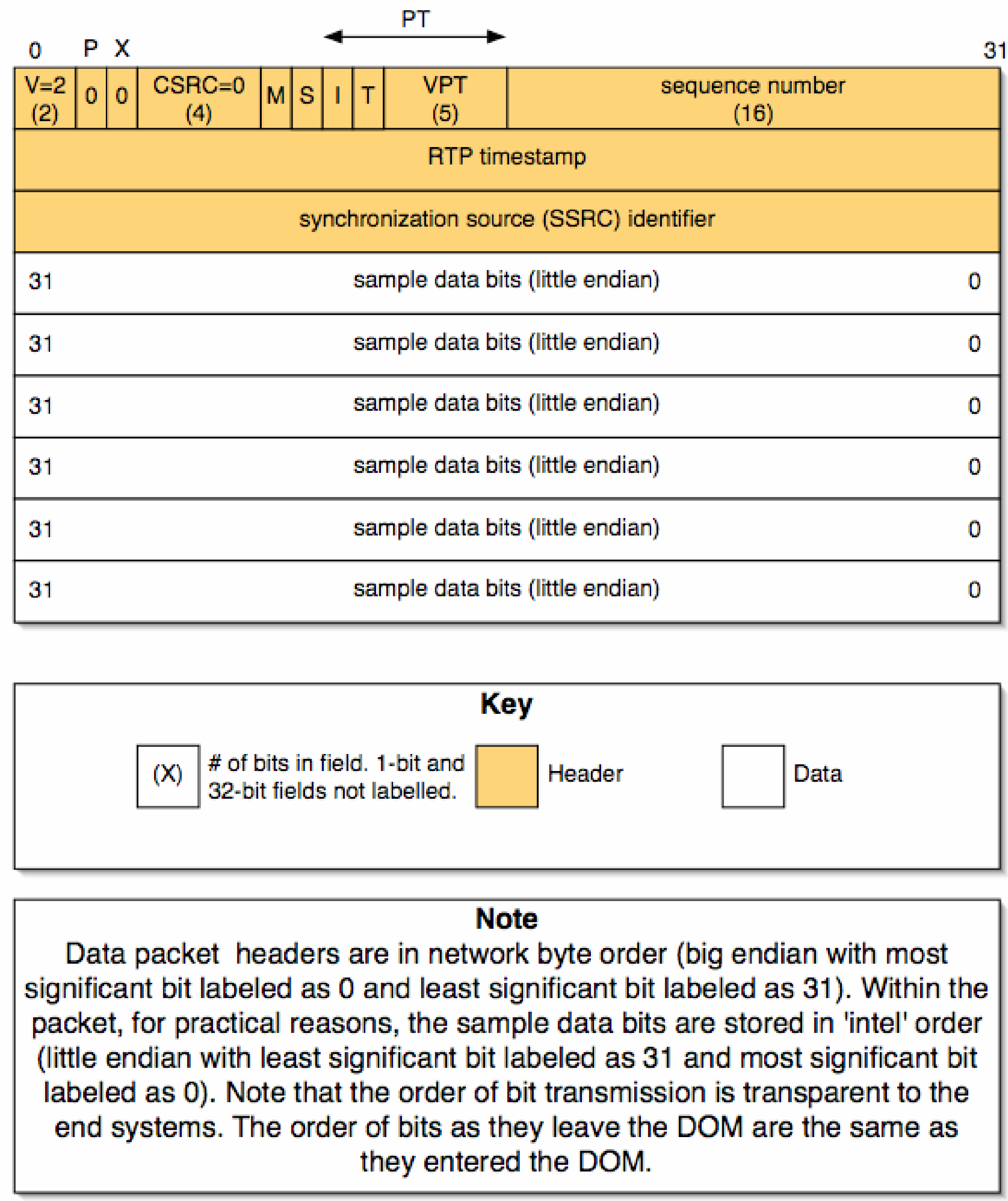


Figure 4: RTP Data Packet

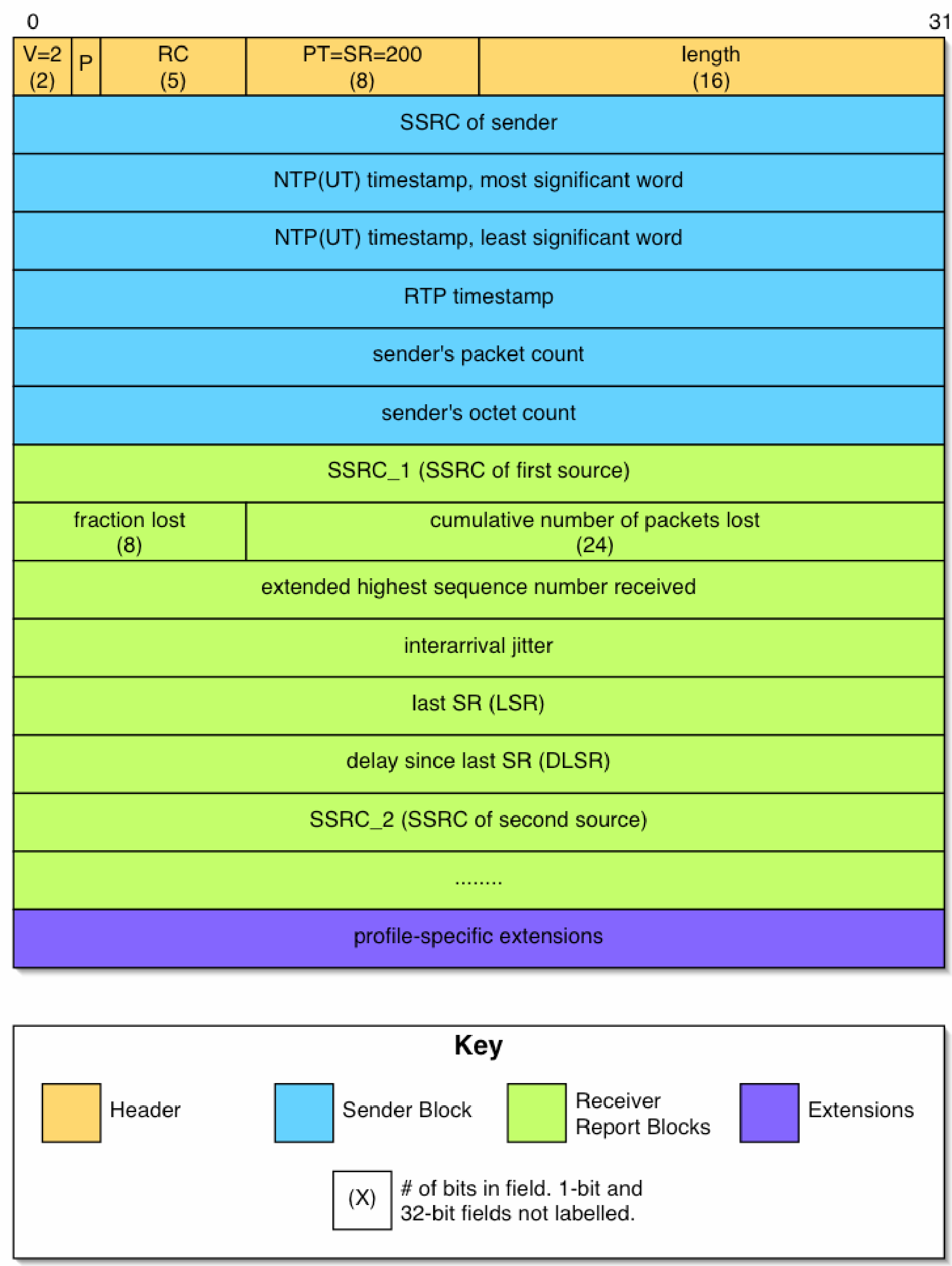


Figure 5: RTCP Sender Report Packet

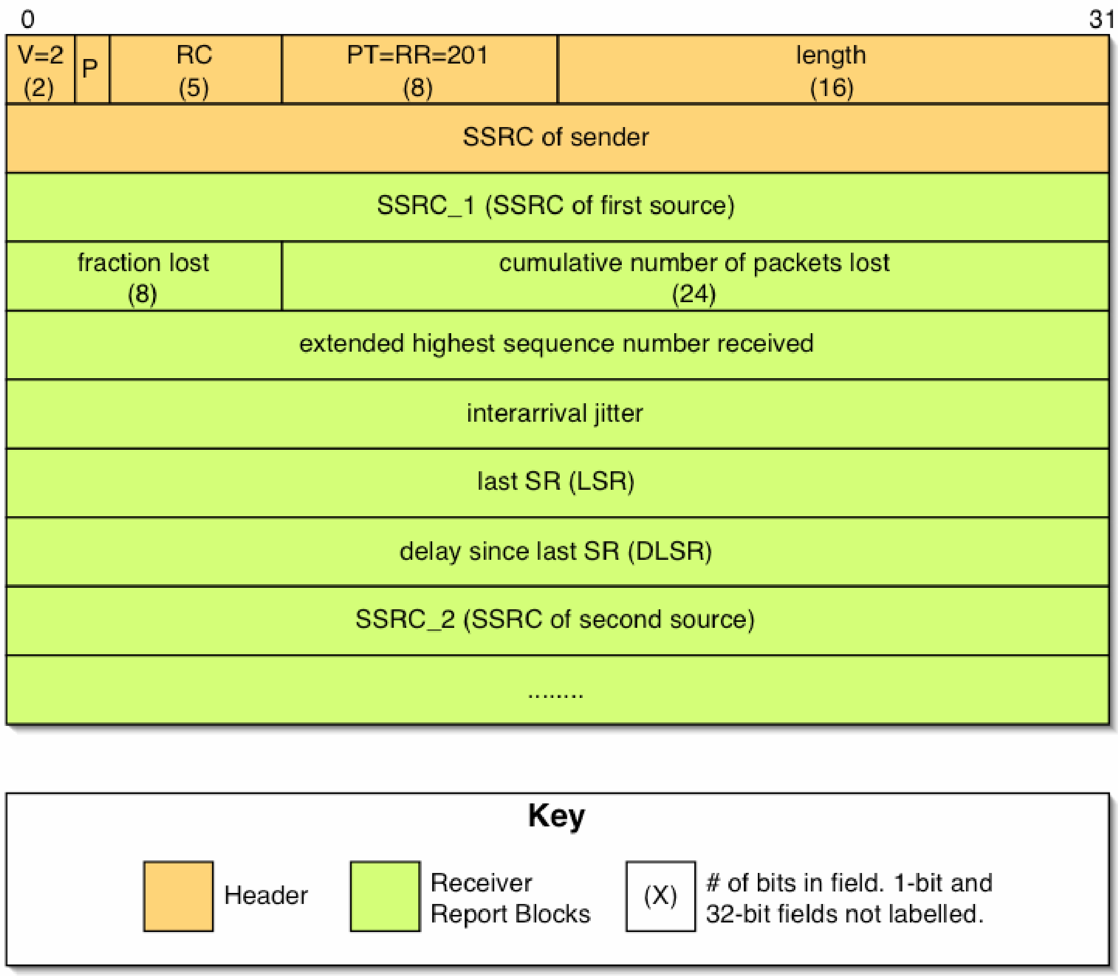


Figure 6: RTCP Receiver Report Packet

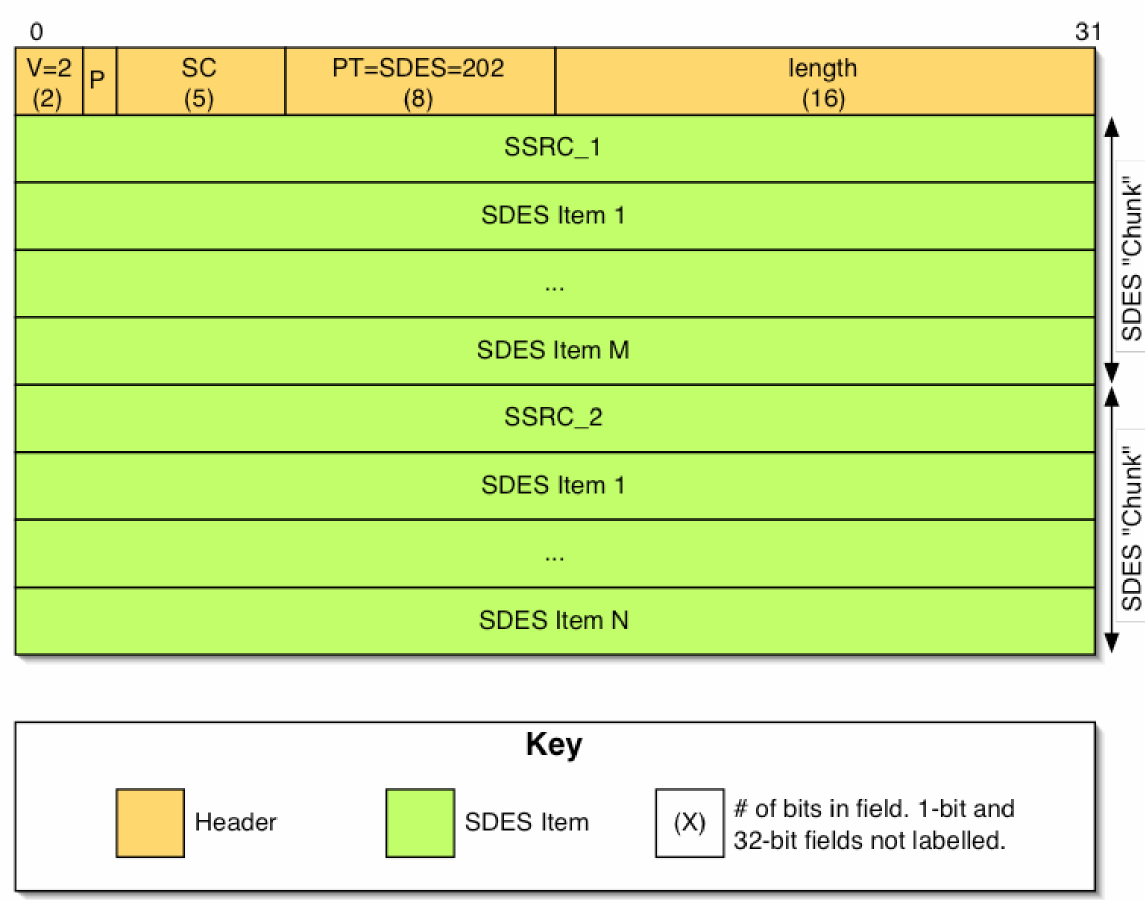


Figure 7: RTCP Source Description Packet Structure

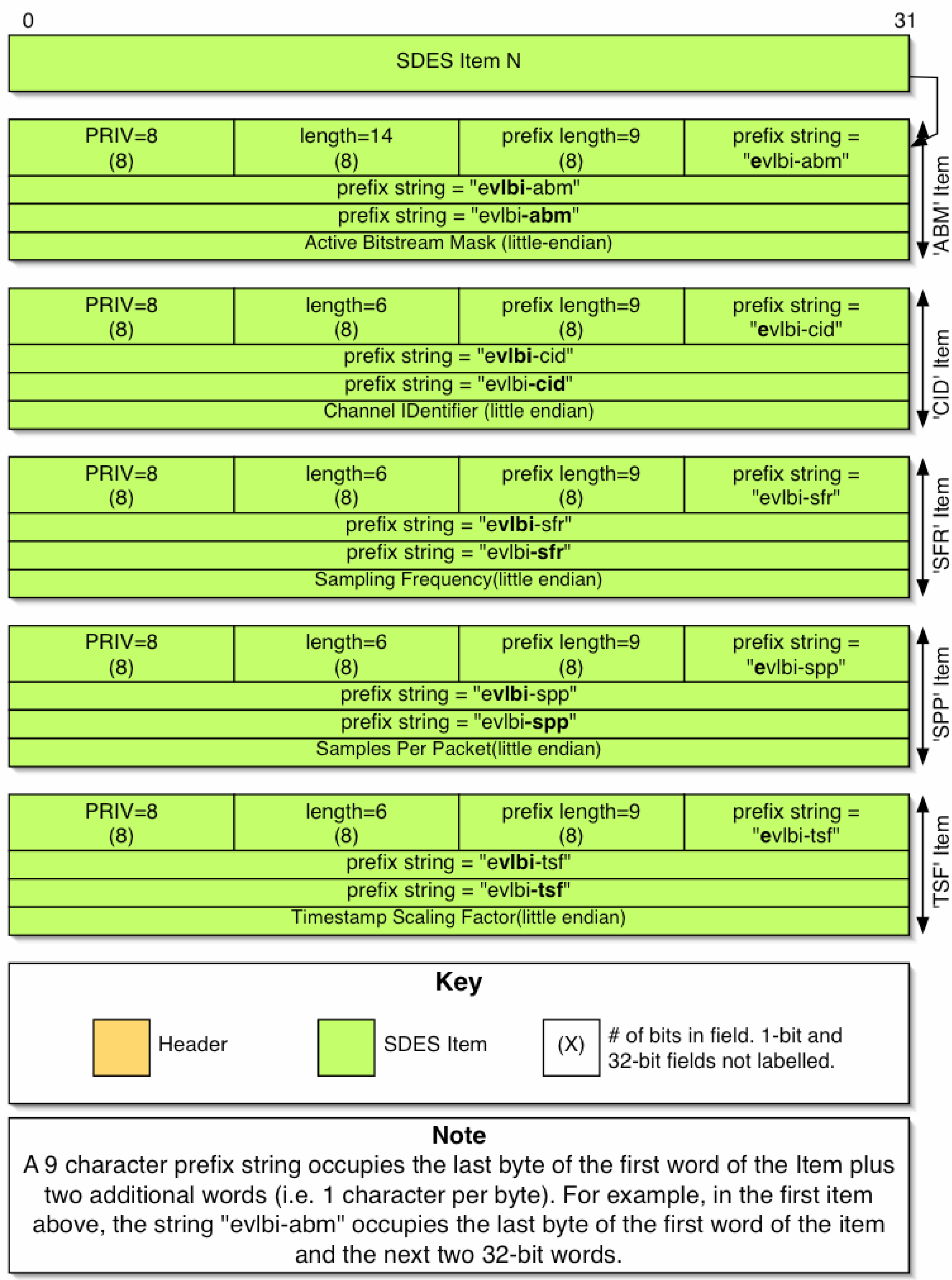


Figure 8: RTCP e-VLBI Source Description Items

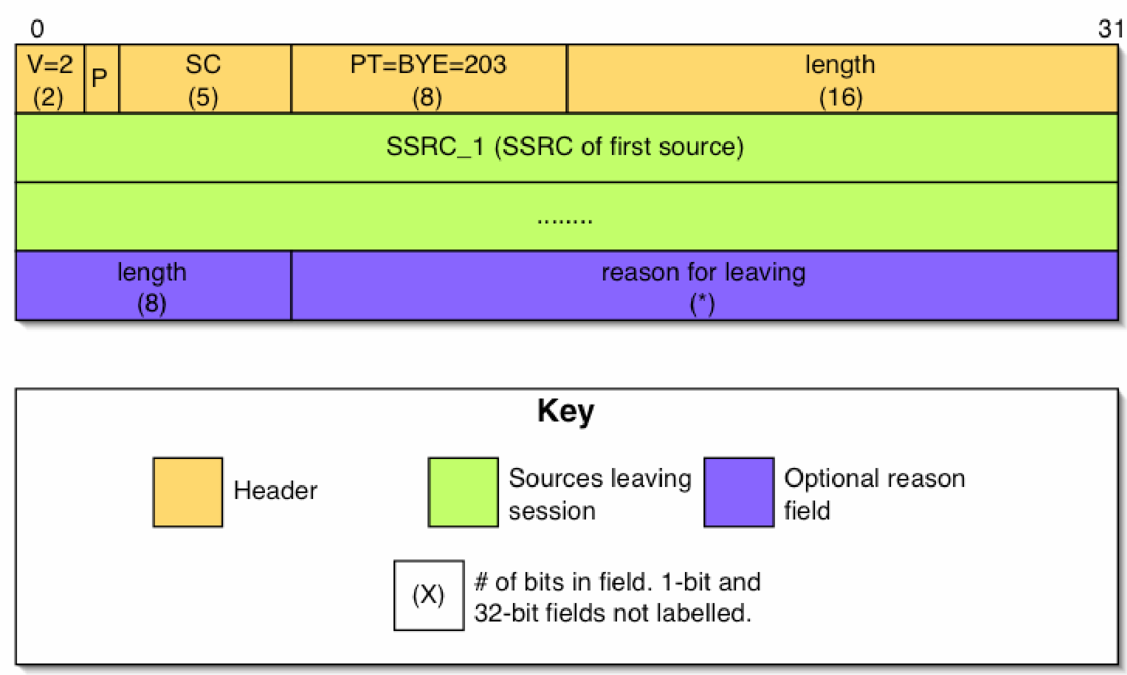


Figure 9: RTCP BYE Packet

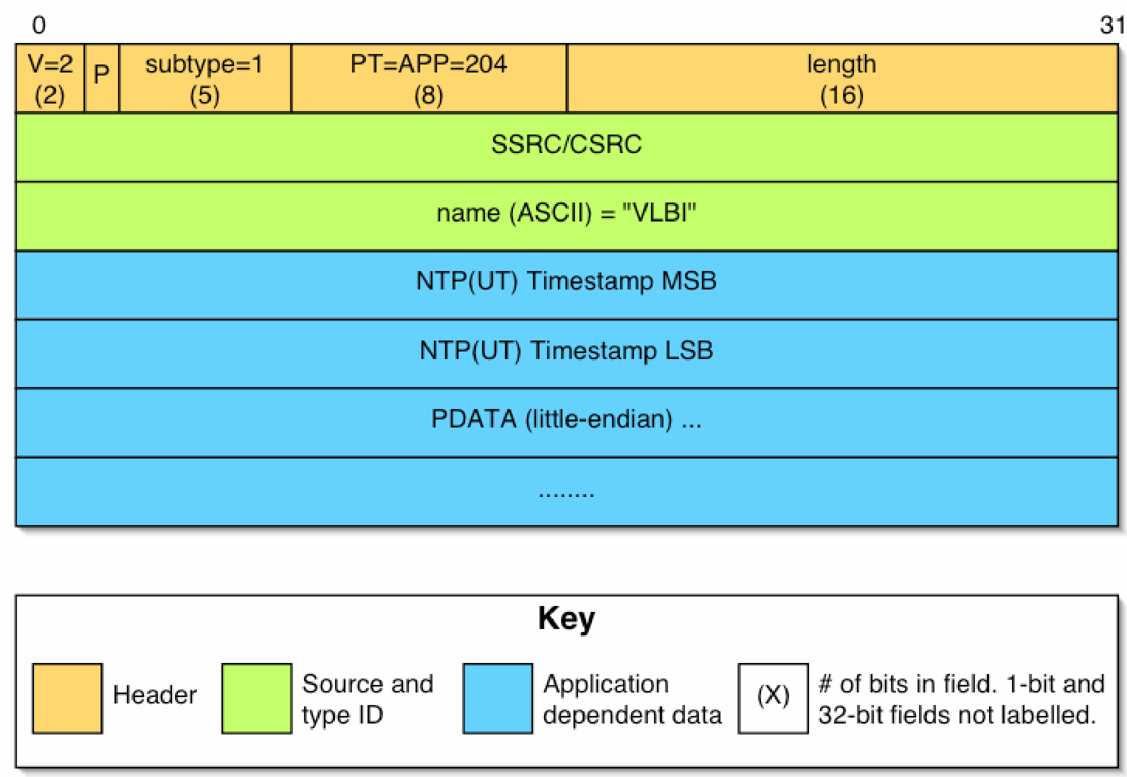
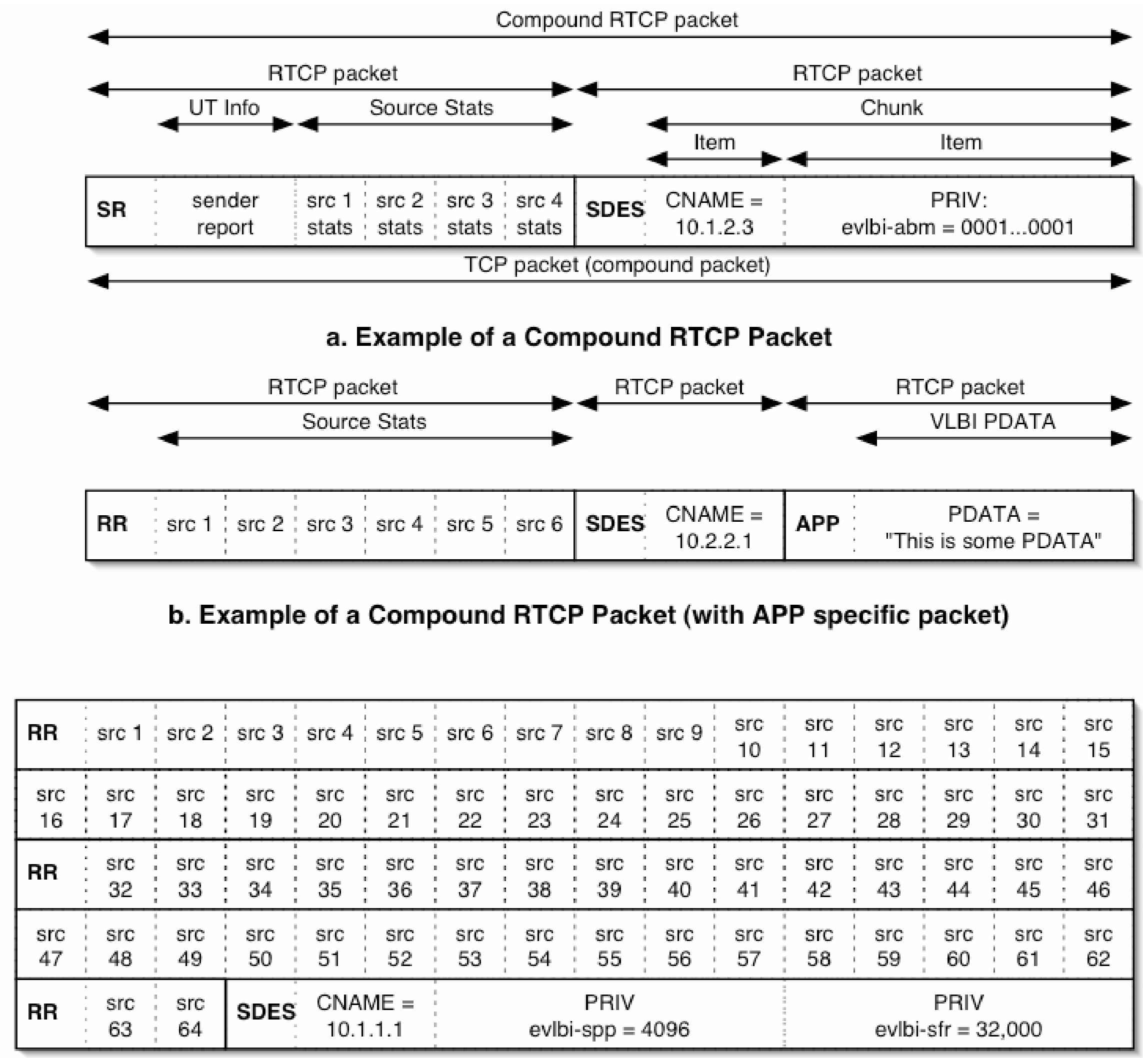


Figure 10: Application-defined RTCP Packet



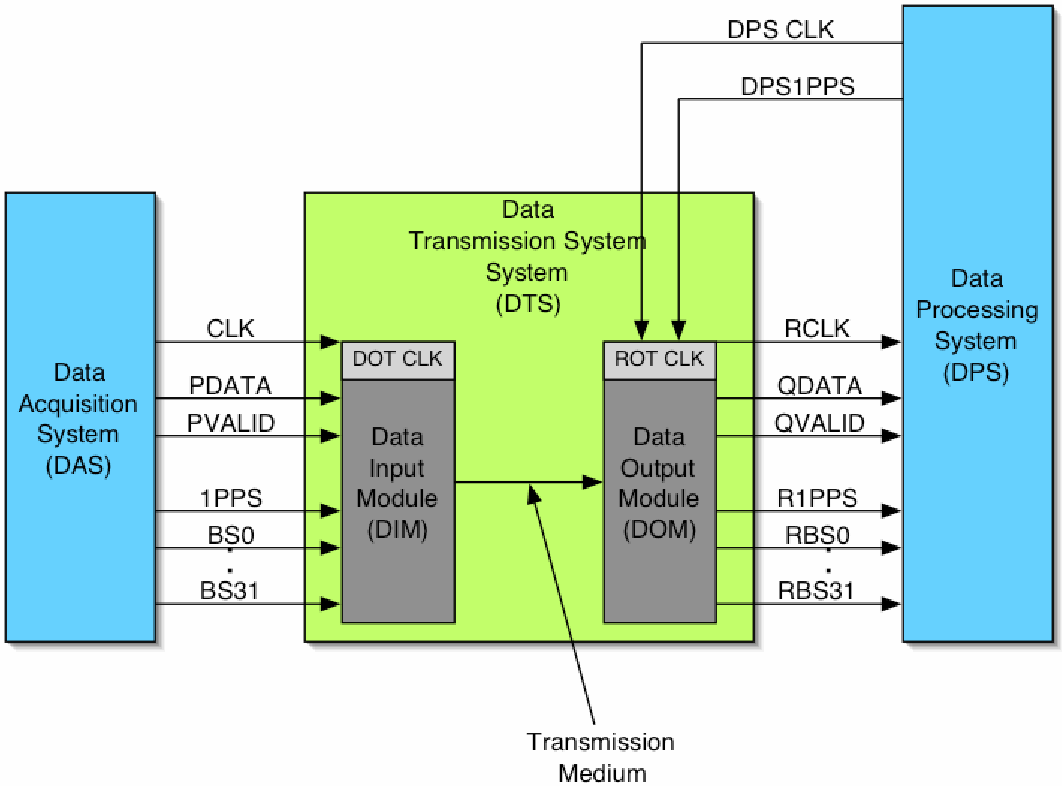
c. Example of a Compound RTCP Packet (with RR for 64 sources)

Note

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 five types of PRIV messages: evlbi-abm, evlbi-cid, evlbi-sfr, evlbi-spp and evlbi-tsf.

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

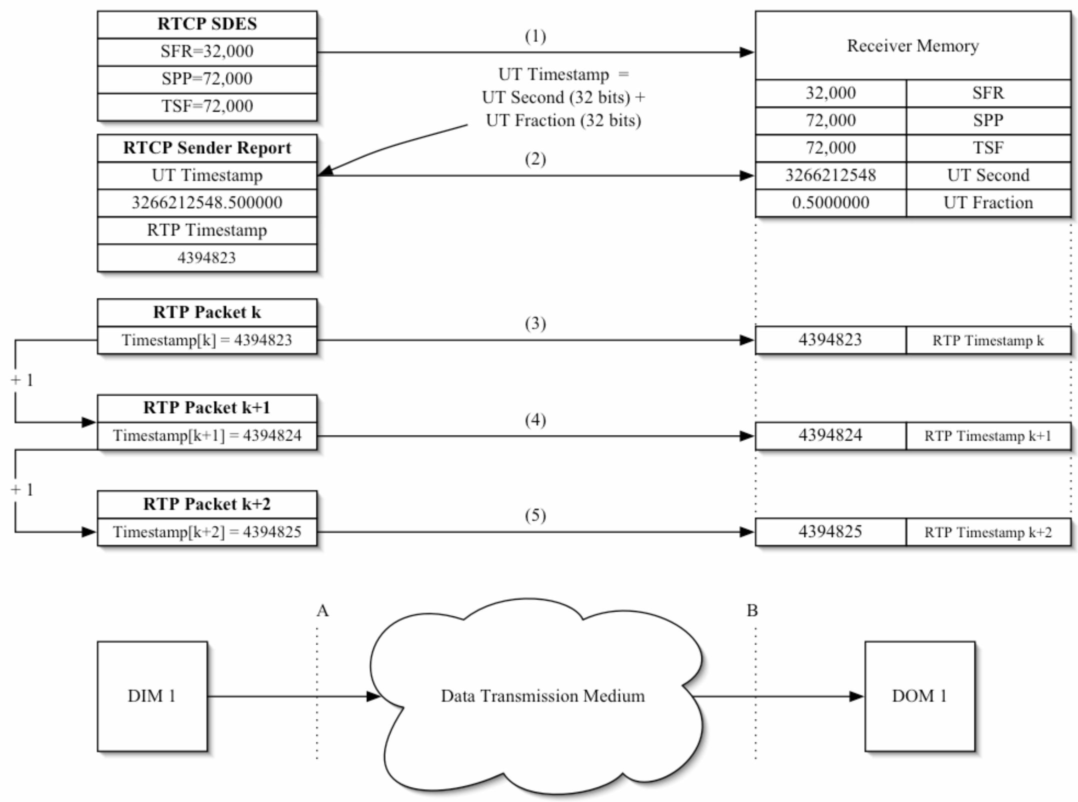
Figure 11: Compound RTCP Packets



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 RTP 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 SDES report is used to communicate the RTP Timestamp Scaling Factor(TSF) to the receiver so that the receiver knows how much the RTP Timestamp is scaled by. TSF is used to scale down the RTP Timestamp at the source(alleviates the problem of rapid wraparound of the RTP timestamp at high speeds). The source divides its internal RTP Timestamp by TSF prior to writing it into the header of RTP data packets. The receiver multiplies the received RTP Timestamps by the TSF to obtain the non-scaled RTP timestamp. The SDES message also contains the number of samples per packet (72,000) and the sampling frequency (32,000 kS/s).

2. RTCP Sender Report contains a synchronization point that relates UT and RTP sampling clock. Using this information, the receiver can convert any RTP Timestamp to a UT timestamp. VSI-E uses the 64-bit UT Timestamp in RTP: 32-bits for the number of integer seconds since 0 hour on January 1, 1900 and 32-bits for the fractional part of the second ($\sim 2.328 \times 10^{-10}$ s). 3-5. RTP data packets are transmitted with the appropriate RTP Timestamp. In this case, the source's RTP sample clock is incremented by 1024 for each packet. The actual RTP timestamp written into the data packet is the RTP timestamp divided by the TSF. Thus, the RTP timestamp seen in the RTP data header is incremented by one for each packet transmitted.

The receiver is able to reconstruct the absolute time for each data packet received by using the synchronization point from the RTCP Sender Report, the known TSF, known sampling frequency(SFR) and samples per packet (SPP).

For example, the UT time for the first sample of the first RTP data packet transmitted can be calculated as:
UT = 3266212548.5+(4394823-4394823) x TSF / (SFR x 1,000)
= 3266212548.5+(4394823-4394823) x 72,000 / (SFR x 1,000)
= 3266212548.5+(4394823-4394823) x 72,000 / (32,000 x 1,000)
= 3266212548.5

Figure 13: RTCP Timestamp Synchronization

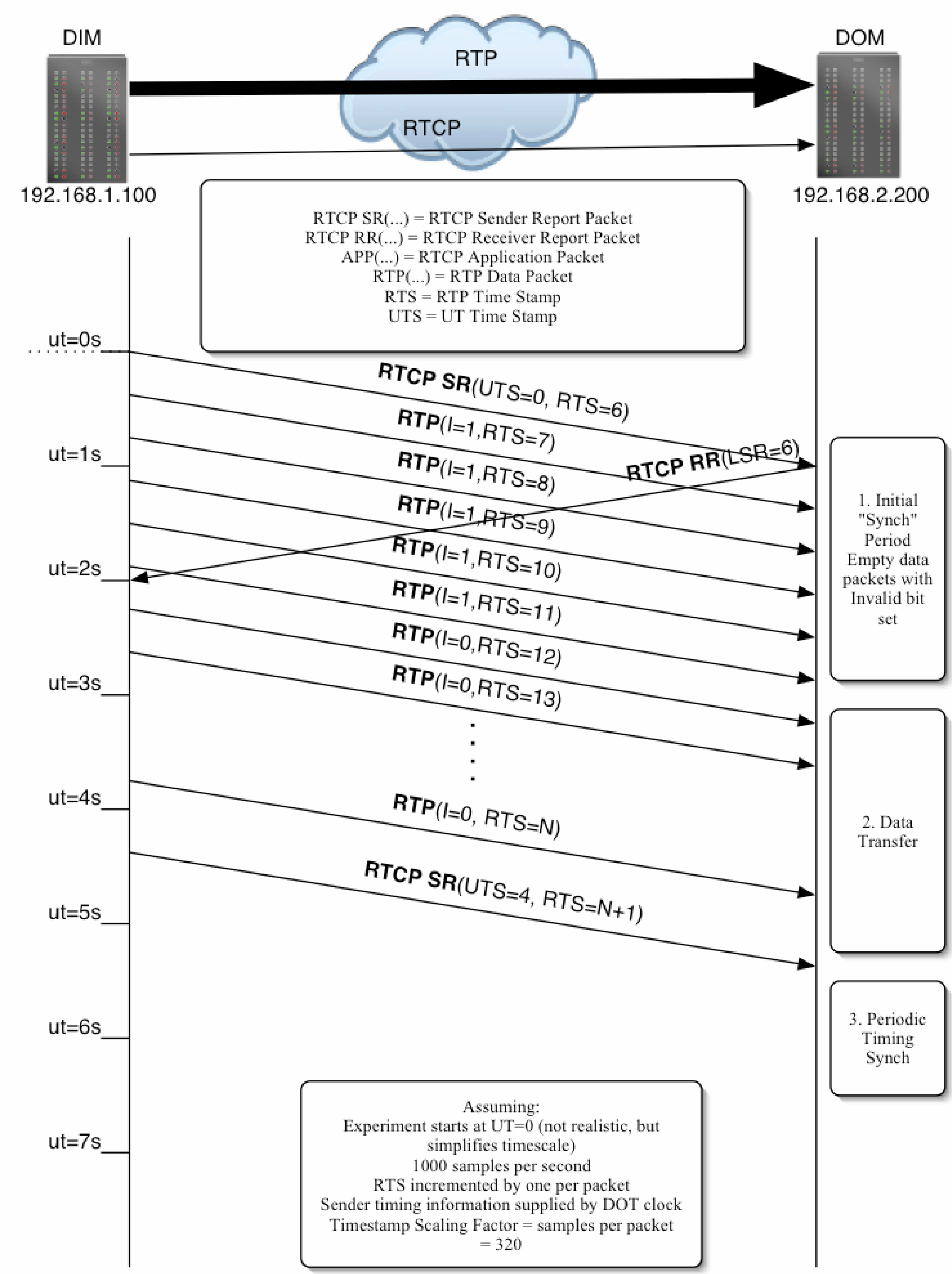


Figure 14: Initial Sample Sychronization

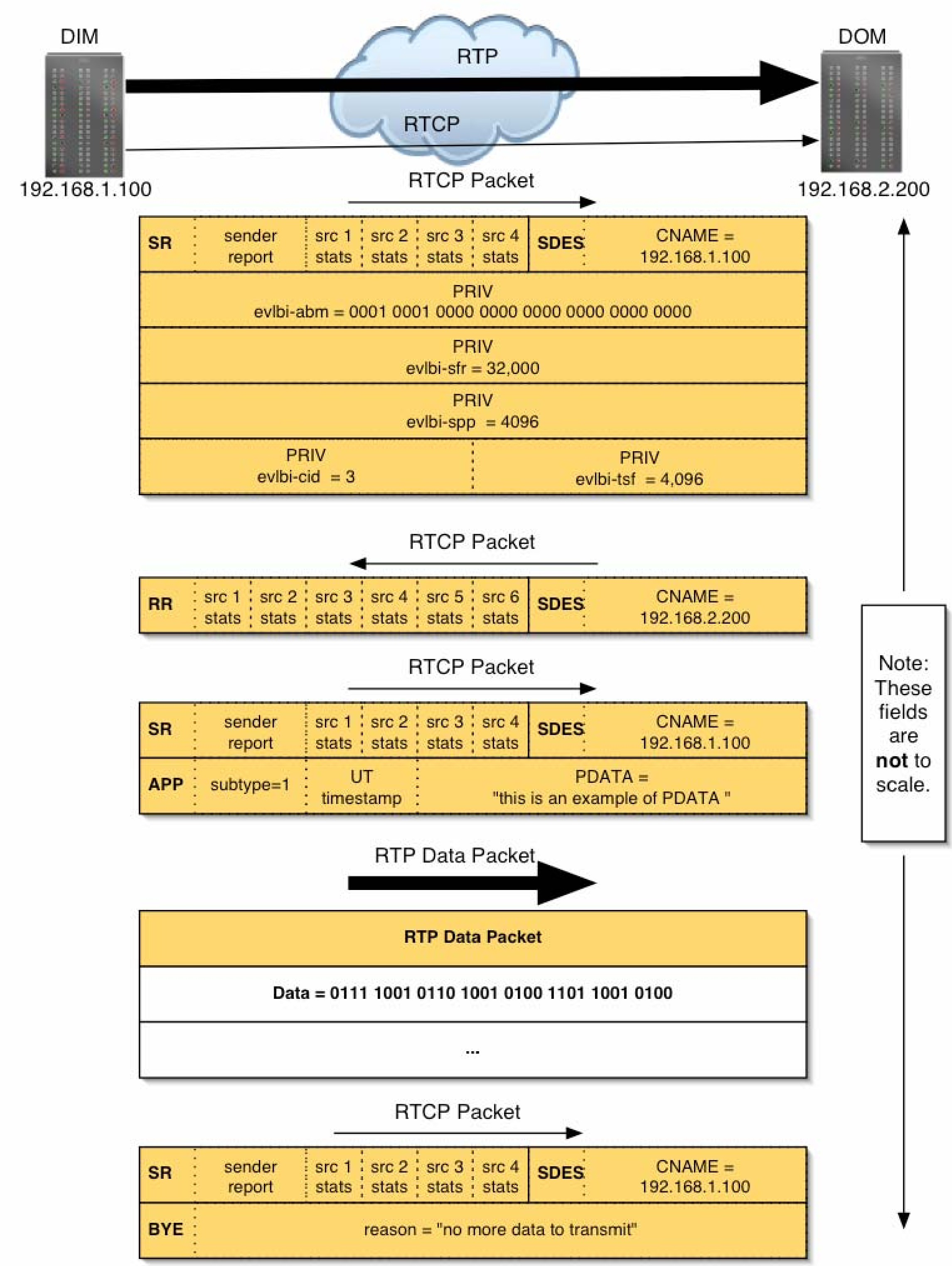


Figure 15: Example RTCP Message Sequence

Appendix A. RTP Timestamp Scaling

Section 1.01 Introduction

The 32-bit RTP timestamp in the header of data packets can represent 2^{32} distinct values (4,294,967,296). In the RTP standard[XXXX], the RTP timestamp must be incremented by 1 for every transmitted sample. However, at high speeds this can cause rapid wrap-around[11] of the RTP timestamp. For example, assuming 28 bytes of overhead (RTP+UDP+Ethernet) per packet, 1500 byte MTUs and 11,776, 1-bit data samples per packet the wraparound time for the 32-bit RTP timestamp at various transmission rates is given in Table 9.

Rate (bps)	Wrap around time (s)
1000000	4377
10000000	437.7
100000000	43.77
1000000000	4.377
10000000000	0.4377

Table 9 RTP Timestamp Wrap-around Times

Thus, at 10 Gbps a burst error that causes more than 0.4377 seconds worth of data to be lost will result in the loss of timing synchronization between the source and the destination. This is not acceptable for modern applications, such as e-VLBI that are approaching these types of data transmission rates.

In order to avoid this, we propose a timestamp scaling mechanism that will make use of a single "scaling" bit in the header of an RTP packet and a Time Stamp Scaling Factor (TSF) known to both the sender and the receiver. The scaling bit indicates whether or not the accompanying RTP timestamp has been scaled by the TSF. In packets that have the scaling bit set, the value of the RTP timestamp is equal to the value as calculated at the source divided by the TSF. Thus, the destination knows to multiply the received RTP timestamp by the TSF to retrieve the unscaled value of the RTP timestamp. This effectively reduces the rate of RTP timestamp wrap-around by a factor equal to the TSF.

Section 1.02 Definitions

In this section, we define the fields used to implement the RTP timestamp scaling mechanism. The mechanism requires a single bit in the RTP header:

Scaling bit (s-bit)

[11] "Wrap-around" in this context occurs when the 32-bit counter used to keep track of RTP samples reaches its maximum value of 4,294,967,295 and is incremented again. This resets the counter to 0 and the sequence starts again.

This is carried in the RTP data header and indicates to the destination whether or not the accompanying RTP timestamp has been scaled.

Timestamp Scaling Factor (TSF)

This is the factor used to scale the RTP timestamp carried in the RTP timestamp field of each packet. When scaling is enabled at a source, RTP timestamps are divided by the TSF before being written into the RTP timestamp field of a data packet. The Scaling bit of this packet is then set to indicate to the destination(s) that the RTP timestamp has been scaled. If a destination receives a data packet that has the Scaling bit set, it will multiply the RTP timestamp carried in that packet by the TSF to recover the original, unscaled RTP timestamp.

Section 1.03 Scaling bit

The scaling bit is included in the header of each RTP data packet. Figure 16 shows the location of the S-bit in the RTP header (bit 9).

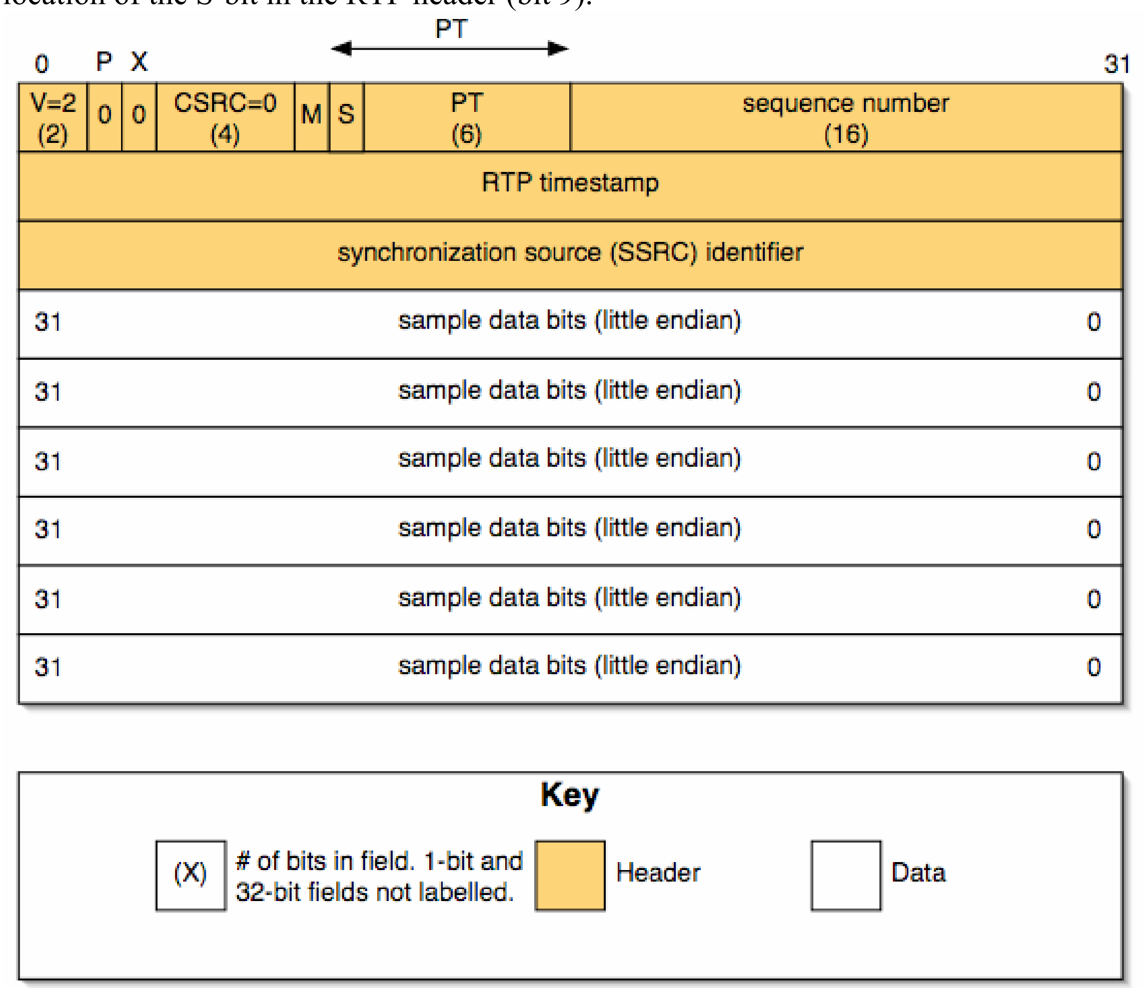


Figure 16 RTP Header with Scaling Bit

Section 1.04 Timestamp Scaling Factor

The Timestamp Scaling Factor is transported as an unsigned 32-bit integer that represents the factor by which RTP timestamps should be scaled (down at the source, and up at the destination). Note that the use of the TSF results in a loss of precision in the RTP timestamp. However, provided the RTP timestamp clock is incrementing at a sufficiently fast rate and the TSF and packet size are chosen correctly, this will not affect the accuracy of the RTP timestamp sequence.

For example, assuming a transmission rate, R, of 10 Gbps, MTU size of 1500 bytes, 28 overhead bytes, 1472 data bytes and 11,776 1-bit samples per packet, the wrap around time of the unscaled RTP timestamp is 0.4377 seconds. If scaling is used, with a TSF of 11,776, then the wrap-around time would increase to 5,153 seconds. Note that the use of a TSF equal to the number of samples per packet means that the RTP timestamp is only incremented once per packet.

Section 1.05 Synchronization of the Timestamp Scaling Factor

In order for the timestamp scaling mechanism to work correctly, both the sender and receiver must agree on when to apply timestamp scaling and the value of the Timestamp Scaling Factor. The scaling bit defined in the previous section is used to communicate when an RTP timestamp has been scaled. There are many ways in which the value of the Timestamp Scaling Factor can be agreed upon by the sender and the receiver. We do not define how this is done here, but leave that to be specified in the profile. The profile must also specify what to do in the case of an error (e.g. a mismatch in Timestamp Scaling Factor between a source and destination).

Section 1.06 Scaling Operation

As mentioned in previous sections, when timestamp scaling is in operation, the RTP timestamps at the source are divided by the TSF prior to transmission. The reverse operation takes place at the destination upon receipt of a packet with the S-bit set. Care must be taken to ensure that the implementation has sufficient precision and that round off problems do not occur.

Let R[k] be the sequence of RTP timestamps, S[k], be the sequence of scaled RTP timestamps (that are actually carried in the packet header), TSF be the Timestamp Scaling Factor and n be the total number of samples per packet. Then,

$$S[k] = R[k]/TSF$$

Let us assume that the TSF is a power of 2 and that dividing by TSF is equivalent to a binary right shift of X bits, while multiplying by the TSF is equivalent to a binary left shift of X bits. The use of scaling is effectively increasing the range of the RTP timestamp by X bits at the source and destination, while at the same time increasing the minimum increment of the RTP timestamp to 2^X. It is important that the packet size, RTP timestamp combination is chosen such that this minimum increment matches the packet size. One easy way to do this is to ensure that the TSF is chosen to be equal to n.

The choice of TSF is left up to the implementer.

Appendix B. RTCP

Section 2.01 RTCP Sender Reports

The sender report provides three functions:

- Reference points that allow receivers to synchronize the RTP timestamps received from this source with UT time.
- 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.

Section 2.02 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:64-bit NTP timestamp, RTP timestamp and sender packet and octet count). 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 2.03 RTCP Source Description Packets

Source Description (SDES) packets are used to describe the source of a particular packet stream. 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 Bitstream Mask - indicates which bits in a channel stream are active.
- ***Evlbi-cid***: Channel Identifier - indicates which channel was the source of this stream of samples.
- ***Evlbi-sfr***: Sampling FRequency - the sampling frequency of the channel samples.
- ***Evlbi-spp:*** Samples Per Packet - indicates how many channel samples are contained in a single RTP data packet.
- ***Evlbi-tsf:*** Timestamp Scaling Factor - used to communicate the Timestamp Scaling Factor described in Appendix A between source and destination.

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

Section 2.04 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 2.05 RTCP Application Defined Packet

The Application-defined RTCP Packet is used to communicate other VLBI control information (for example, 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 2.06 RTCP Packet Transmission

Section 6.1 of [17] 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 [17].

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 [17]). 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 two sources. The Sender Report also has a 64-bit UT timestamp field for providing high resolution timing synchronization. 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 Channel Bit 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 2.07 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[17]. 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 C. RTCP Transmission Interval[12]

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 [17] that RTCP use no more than 5% of the session bandwidth. This is loosely defined in [17] 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 [18]:

> "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 [17])

1. Session bandwidth $B_{SESSION} = N \times R$ [bps][13]
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 [17] 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,

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

where T_{RTCP} is the RTCP transmission interval, T_{CALC} is the calculated transmission interval, ϕ 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:

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

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

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

Appendix D. RTCP Transmission Interval Code

The following code is taken from Appendix A.7 of RFC3550 [17].

```
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;
}
```

14. 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

15. 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. H. Schulzrinne, S.C., R. Frederick, V. Jacobson, *RFC 3550: RTP: A Transport Protocol for Real-Time Applications* available: http://www.ietf.org/rfc/rfc3550.txt?number=3550, IETF, 2003.
4. Schulzrinne, H., *RTP Tools* available: http://www.cs.columbia.edu/~hgs/rtp/, 2003.
5. Schulzrinne, H., *RTP Libraries* available: http://www.cs.columbia.edu/~hgs/rtp/, 2003.
6. 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.
7. J. Wroclawski, a.o., *An RTP Profile for e-VLBI Data* available: 2003.
8. Postel, J., *RFC768: User Datagram Protocol* available: http://www.ietf.org/rfc/rfc0768.txt?number=768, IETF, 1980.
9. Postel, J., *RFC793: Transmission Control Protocol DARPA Internet Program Protocol Specification* available: http://www.ietf.org/rfc/rfc0793.txt?number=793, IETF, 1981.
10. 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.
11. 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.
12. Jamshid Mahdavi, S.F., *TCP-Friendly Unicast Rate-based Flow Control* available: http://www.psc.edu/networking/papers/tcp_friendly.html, 1997.
13. Alan Whitney, J.W., *Experiment Guided Adaptive Endpoint: NSF Grant # ANI-0230759.* 2002, MIT: Boston. p. 17.
14. 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.
15. Postel, J., *RFC791: Internet Protocol* available: http://www.ietf.org/rfc/rfc0791.txt?number=791, IETF, 1981.
16. 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.
17. 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.

18. Schulzrinne, H., *RTP Frequently Asked Questions* available: http://www.cs.columbia.edu/~hgs/rtp/, 2003.