USRP3 Concepts

1 Introduction

The USRP3 is the 3rd generation of H/W products to support Software Defined Radio from Ettus Research. It extends existing USRP2 functionality and concepts further with a new transport protocol, CHDR, that provides features designed to support use of packetized data between signal processing blocks.

2 CHDR – Compressed Header VITA Packets.

The VITA49.x protocol group contains a lot of header fields that have largely constant or unused fields. USRP3 implements a packet format named CHDR that compresses and eliminates certain VITA49 fields for more efficient bandwidth utilization. CHDR packet to VITA49 packet bi-directional mapping, and hence Ethernet/IP/UDP encapsulation can be easily done with some simple programmable lookup tables. CHDR can be applied to either VITA49 IF Data packets or to Context Extension packets. CHDR overloads the 32bit SID field defined as part of VITA49 to hold a pair of 16bit CHDR src/dst addresses which allow CHDR packets to be efficiently labeled, switched and routed within USRP3 architecture devices.

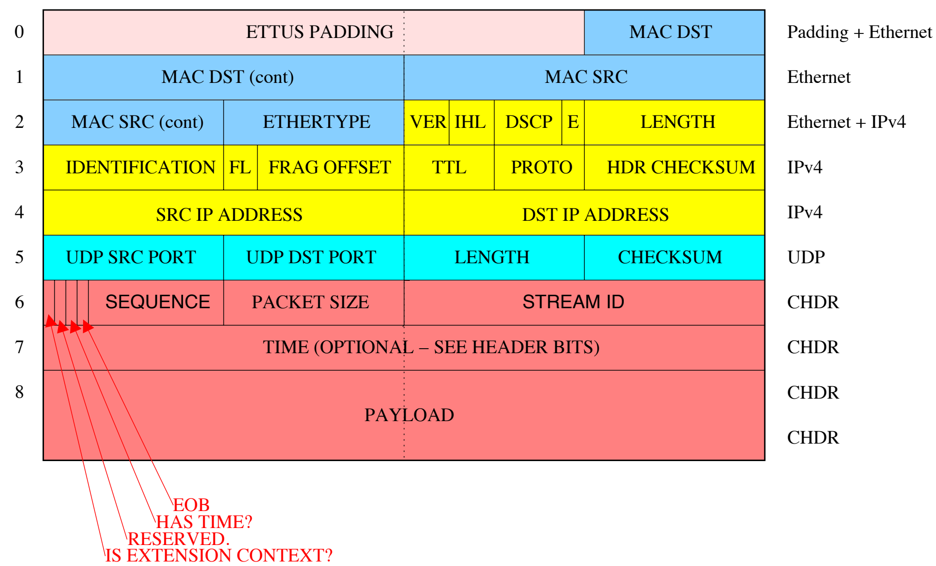


Figure 1 - CHDR Packet framed with Ethernet, IP and UDP headers

2.1 CHDR Packet Types

CHDR defines a small number of packet types used for specific tasks. These types are:

2.1.1 IF Data

This packet type streams radio sample data to and from the radio. The sample format is flexible and can be fixed or float formats of various bit sizes, however the packet itself does not identify the same format used.

2.1.2 Async Control/Response

Async Control/Response packets allow internal USRP control bus transactions to be initiated over a network transport, and results returned to the controlling device.

2.1.3 Source Flow Control

Source flow control packets provide point-to-point flow control for paired IF Data flows by reporting data consumption at a sink to the source.

2.1.4 Status

Status packets provide reporting of error status or acknowledgement to a controlling device.

3 Dataflow, Packet Addressing, and Switching

Previous USRP architectures have not provided flexibility in the internal routing of dataflows of sample data, requiring FPGA changes to change source-sink pairings. USRP3 introduces the concept of a source and destination address by using the SID field in CHDR packets to hold this information. This provides the ability to alter the topology inside a USRP3 device under software control, and to allow complex topologies of multiple USRP3 devices each with multiple network interfaces that can directly communicate.

3.1 Address Format

The SID (Stream Identifier) field defined by VITA49 provides 32bits to uniquely identify all the packets belonging to a particular stream. By subdividing the SID field into two 16bit fields it can be used to specify directly the source and sink of a dataflow, whilst still satisfying it's original purpose. USRP3 then further subdivides these 16bit addresses into two 8 bit fields, which by convention indicate device (USRP or Host) and endpoint within the device.

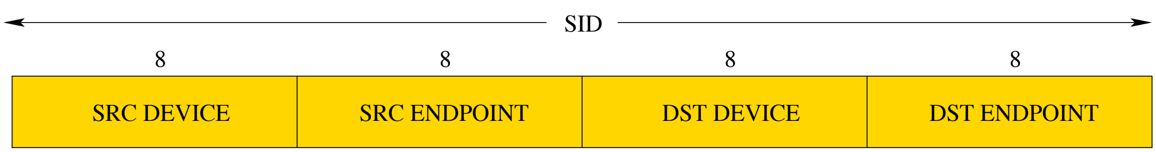


Figure 2 - SID to CHDR address mapping

3.2 Switching

There are 3 primary hardware mechanisms in USRP3 to provide multiple paths between endpoints, the Mux, the Demux, and the Crossbar Switch

3.2.1 Mux

The Mux is the simplest block, since it is a many to one device, having no alternative egress options, the address field is not used and packets from all inputs are switched onto the output. Muxes can be configured at FPGA design time as having a fair round robin arbitration, or a fixed weighted priority where higher arbitrating ports always preempt lower arbitrating ports.

3.2.2 Demux

The demux switches a single ingress port between a number of egress ports. Since there is only one input no arbitration is required. The decode of Dst address that select the egress port is determined at FPGA design time and could be either fixed or programmable.

3.2.3 Crossbar Switch

The Crossbar Switch is the most flexible packet routing function in the USRP3 architecture and there tends to be at least one instance used in complex USRP3 implementations. The Crossbar switch implements a fully non-blocking NxM matrix switch using all bits of the packet Dst addresses. The crossbar switch contains a lookup table that functions as a Tertiary CAM to perform a hierarchical look up of egress port based on the Dst address. The Dst.Device field is first matched and if it refers to a device that is not the local device then it directly returns an egress port for the packet (on the presumption that all endpoints of a remote device will route via the same path). However if the Dst.Device field matches the local device address then the Dst.Endpoint field is used to perform a further lookup to return the egress port that routes to the desired endpoint. In case of contention for a particular egress port, the Crossbar Switch implements a round-robin arbitration scheme. The lookup table is populated by software and can be programmed on the fly in a live system.

3.3 IF Data Bursts.

Ettus Research IF Data transport protocols extend VITA49 with a concept of a Burst, which can be defined as a time contiguous stream of IF Data samples, framed in one or more packets.

CHDR format packets encode the end of bursts with a dedicated header bit, where as VITA49 packets use one of the reserved header bits (bit24).

Bursts are only defined for IF Data packets, not the other defined CHDR packet formats.

3.4 Timed Transmission

Transmissions can be asynchronous, i.e they start transmission whenever the initial sample has propagated through the network transport and then to the Radio DSP, or synchronous, where they are tagged with a 64bit VITA time that is compared with the local VITA time in the Radio DSP. The IF Data is held at the input to the Radio DSP until the local VITA time matches the VITA time encoded in the start of burst packet, at which point the sample data is passed into the DSP.

The DSP has known and constant group delay for a given programmed configuration, and so the time that the samples are presented to the data converter is offset (later) from the VITA clock based start time by a constant and known amount. Synchronous bursts arriving late at the Radio DSP generate an error as detailed elsewhere and the error recovery behavior is determined by a programmed policy. Synchronous bursts are essential to correct operation of MIMO systems, as the start and stop of streaming can be precisely synchronized across multiple channels.

Packets that are not the start packet of a burst, and that contain a VITA timestamp are considered to have valid syntax, however the USRP3 Radio DSP will ignore the subsequent VITA timestamp(s) in an ongoing burst. A burst can be of any length, including a single packet.

3.5 Underflow on Transmission

The scheduled on-air time of sample data is utter inflexible, when IF Data packets containing sample data have failed to arrive within the Radio by the nominal on-air time for there contained sample data then underflow is the result. The typical root causes of underflow are numerous; Host CPU scheduling, Host computational overload, network congestion or propagation delays. USRP's contain large sample buffers to minimize the possibility of underflow, but it is always a possibility in a typical USRP & SDR Host based system. USRP3 supports various programmable policies to control error recovery in this situation.

Upon detection of underfow, the USRP3 H/W transitions into an error state, ceases transmission, and signals underflow in a status message to the host. Recovery from an error state is discussed in 3.6 Error Recovery During Transmission.

3.6 Error Recovery During Transmission

From this point onwards, recovery from the error state depends on which programmable policy has been selected using **Error! Reference source not found.**. Three different policies exist: WAIT, NEXT_PACKET, and NEXT_BURST.

WAIT causes the transmit section of the radio to remain in the error state indefinitely, until the host initiates recovery by resetting the USRP.

NEXT_PACKET causes the transmit section of the radio to discard sample data as it becomes available to the radio until the start of a new packet is found, where upon the transmit logic attempts to re-start transmission.

NEXT_BURST causes the transmit section of the radio to discard sample data as it becomes available to the radio until the start of a new packet that follows a packet that had the EOB bit set is found, where upon the transmit logic attempts to re-start transmission.

In both the NEXT_BURST and NEXT_PACKET cases the radio logic can discard sample data faster than the maximum legal sample rate and so it has the ability to catch from error conditions

3.7 Flow Control

The extensive use of point-to-point links in the USRP3 architecture, implemented as AXI4-Stream interfaces in hardware and carrying CHDR formatted data, allows both data and control to stream between blocks with very deterministic performance. Whilst flow control over a single hop of a point-to-point link is handled perfectly by the low-level hardware ready-valid handshaking, end-to-end flow control for a particular dataflow requires a different solution. The need for end-to-end flow control is driven by two main factors:

1. Head of line blocking. If multiple dataflows are multiplexed over a single point-to-point link then the packets that form these dataflows are serialized at the packet level. If one of the dataflows begins to be back pressured from it's ultimate sink then this causes all dataflows that share segments of point-to-point links involved in the back pressured dataflow to also stall. This in turn causes other dataflows that share some of these point-to-point links to also be unnecessarily stalled because the transmission of a packet over a link is an atomic operation and the stalled partially transmitted packet can not be preempted or moved aside to allow other unblocked dataflows to pass.
2. Buffer Management. The USRP3 architecture distributes relatively large amounts of elastic buffering throughout the pipeline. This occupancy of this buffer space needs to be efficiently managed for maximum performance. The buffering allows transport links, such as Ethernet that have high throughput and potentially high access latency, as well as multitasking application hosts which again have high throughput but also high latency, to interface losslessly with the inflexible sample clock driven radio frontend. Thus managing the occupancy of this elastic buffer space is important for it to function correctly...if it fills or empties completely during active streaming then overflow/underflow will result.

Because the capacity of the elastic buffering, in any given dataflow, for any given USRP3 based implementation, is well known, then occupancy can be managed by timely reporting of packet consumption at the dataflow sink back to the dataflow source, rather than monitoring the various buffers directly. In USRP3 the reporting of consumption is implemented using a protocol that uses CHDR formatted packets, as shown in Figure 3 - Source Flow Control CHDR Packet Format, as a transport.

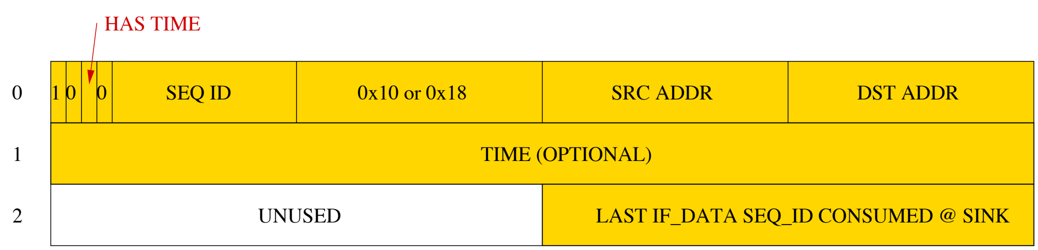


Figure 3 - Source Flow Control CHDR Packet Format

The Source Flow Control Packet is addressed back to the source of the dataflow, and the TIME field may be populated in some applications though it is not used. Each Source Flow Control Packet has it's own unique SEQ_ID, unrelated to the SEQ_ID's of the dataflow being controlled. At dataflow initialization the SEQID of the stream is initialized to zero, and whilst only the 12LSB's are embedded in the CHDR packet header it is maintained to a modulo-2^32 precision at both sink and source, with the full 32bit value being reported back to the source in the Source Flow Control Packet. Note that every packet consumed does not have to be directly reported, Source Flow Control Packets can for example be sent once for every N packets consumed as the embedded SEQ_ID shows the latest packet consumed and packets are assumed in order and reliably delivered. Reporting of SEQ_ID discontinuity and other error conditions resulting from lost or corrupted packets is not part of the flow control protocol.

3.8 Status Reporting

The USRP3 architecture provides for status and error handling and reporting for a number of common situations that can occur with flows of IF Data packets. Errors and acknowledgments are reported back to the upstream source using CHDR Status packets. Status and error report packets include: a 64bit VITA Time that timestamps when the packet was actually generated, a status or error code, and the sequence ID of the IF Data packet that triggered the report packet. The layout of the report packet is shown in Figure 4 - Status and Error Report, CHDR Status Packet.

0	1	0	1	0	SEQ ID	0x6	SRC ADDR	DST ADDR	
1	VITA TIME								
2	ERROR STATUS CODE						0x0		SEQ_ID

ACK	0x00
EOB (End of Burst)	0x01
UNDERRUN	0x02
SEQUENCE ERROR	0x04
TIME ERROR	0x08
RESERVED	0x10
MID–BURST SEQUENCE ERROR	0x20

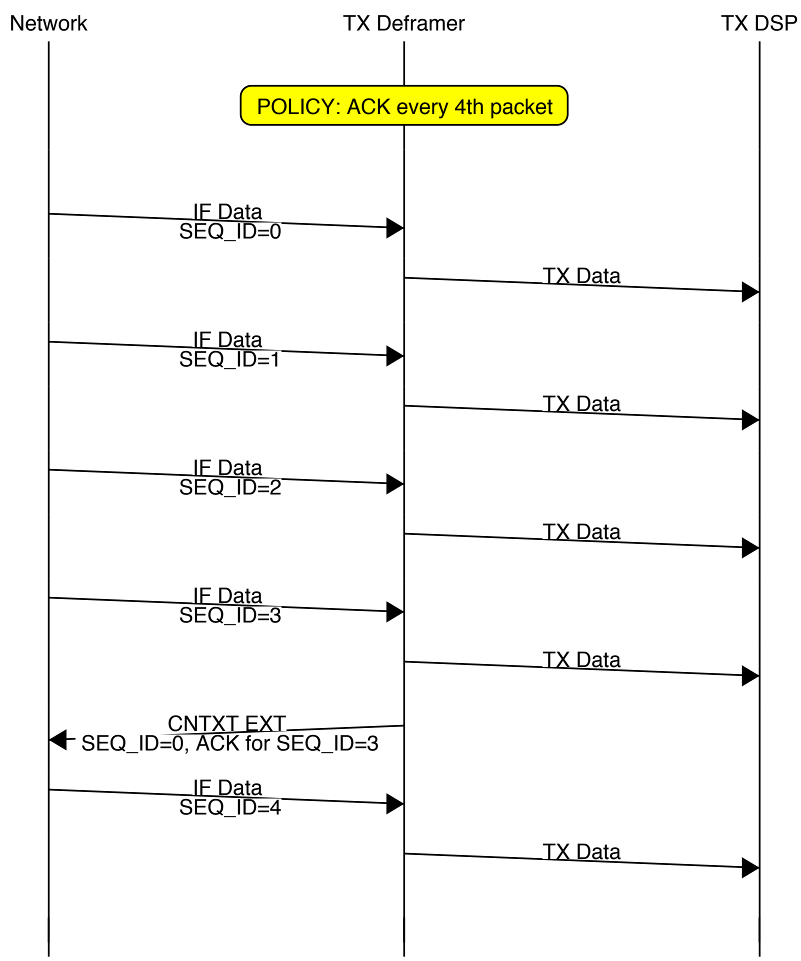
Figure 4 - Status and Error Report, CHDR Status Packet

Status and error reporting is generally only implemented in a transmit direction, since it's generated at the sink, which is already at the host in the receive direction.

3.8.1 Status and Error Codes.

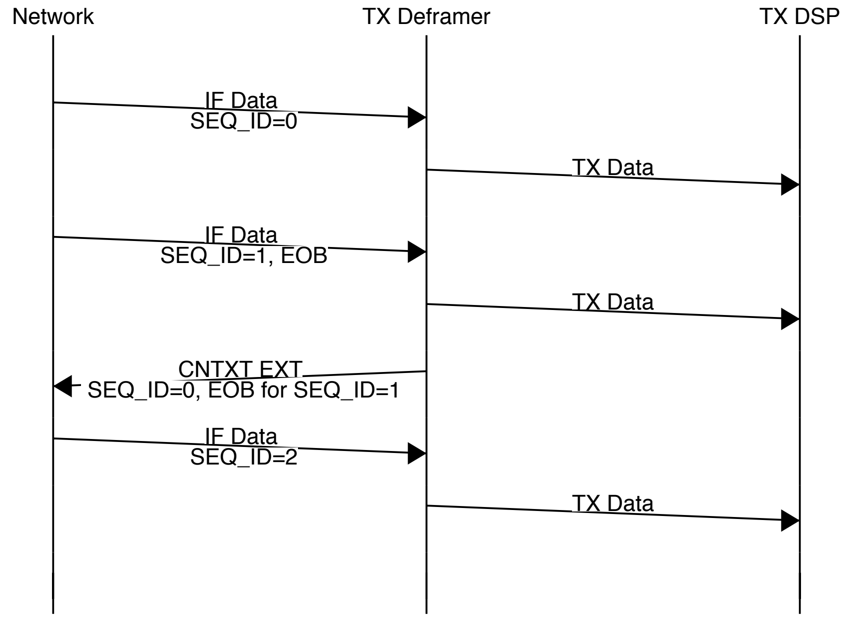
3.8.1.1 ACK

ACK can be issued for any non-error packet and serves to timestamp the moment that a packet is fully consumed by the Radio DSP. Whilst all packets could generate an ACK response, in general implementations allow ACK to be generated every Nth IF Data Packet and/or after a timed interval expires.



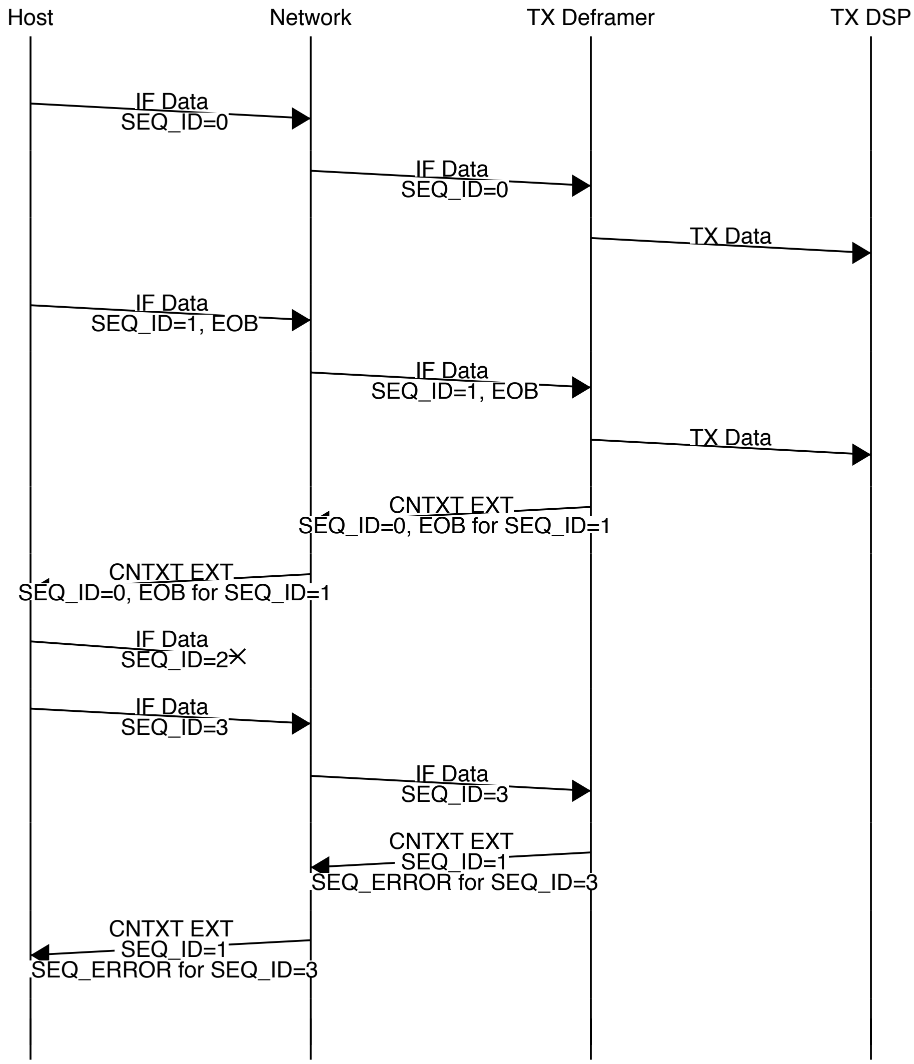
3.8.1.2 EOB (End of Burst)

EOB is issued when a CHDR packet with the EOB bit set is fully consumed by the Radio DSP.



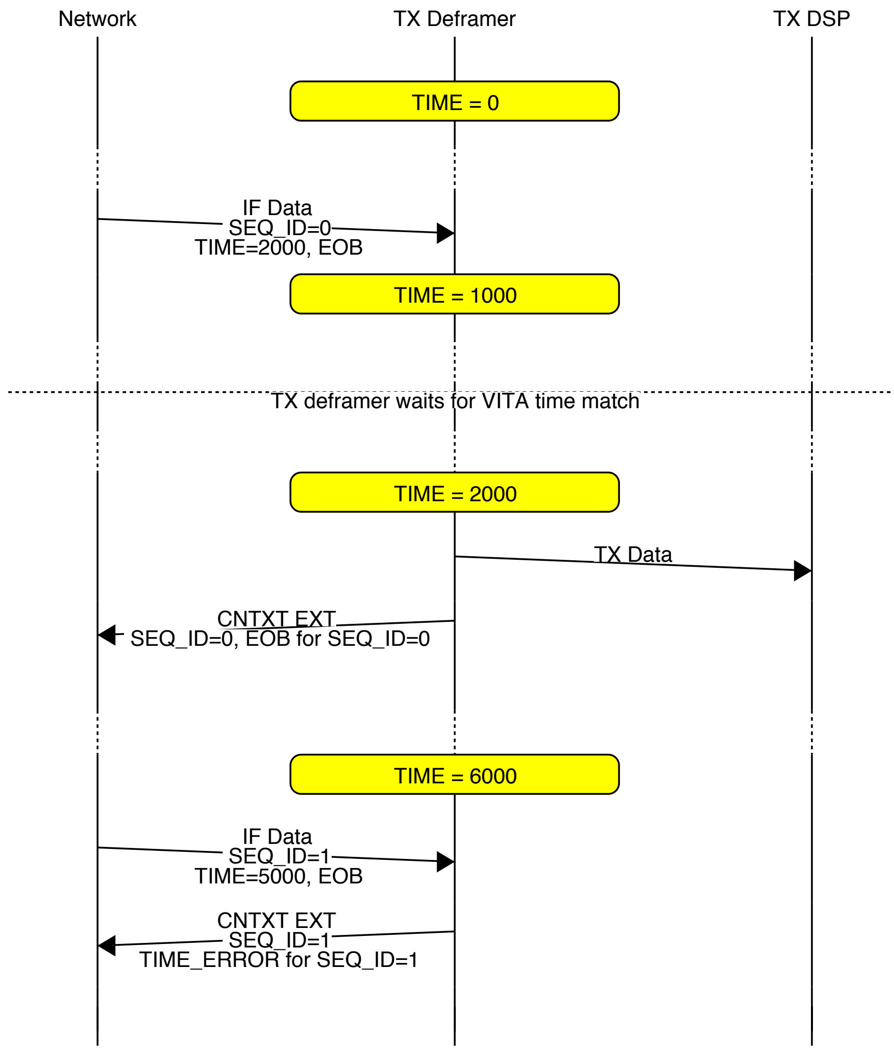
3.8.1.3 Sequence Error

Sequence Error is issued when there is a sequence ID discontinuity at the start of a new burst (i.e previous packet had EOB flag set).



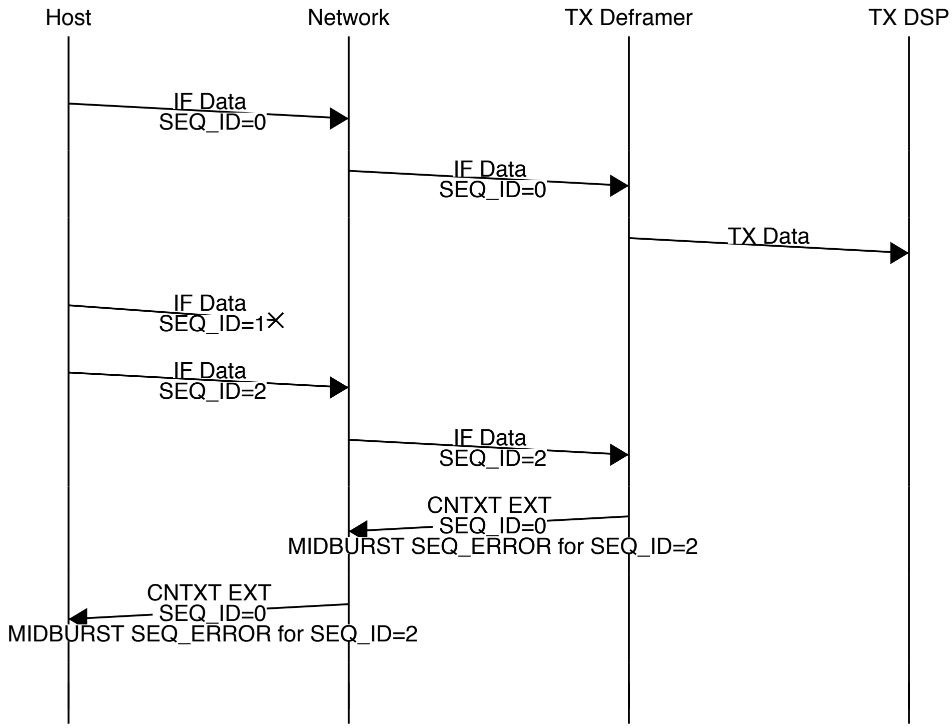
3.8.1.4 Time Error

Time Error is issued when a packet that starts a new burst for transmission arrives at the Radio DSP with a VITA timestamp for a time already in the past.



3.8.1.5 Mid-Burst Sequence Error

Similar to a Sequence Error, a Sequence ID discontinuity is detected, however in this case within a burst (i.e on any packet except the first packet in a burst).


Host
Network
TX Deframer
TX DSP
IF Data
SEQ_ID=0
IF Data
SEQ_ID=0
TX Data
IF Data
SEQ_ID=1
IF Data
SEQ_ID=2
IF Data
SEQ_ID=2
CNTXT EXT
SEQ_ID=0
MIDBURST SEQ_ERROR for SEQ_ID=2
CNTXT EXT
SEQ_ID=0
MIDBURST SEQ_ERROR for SEQ_ID=2