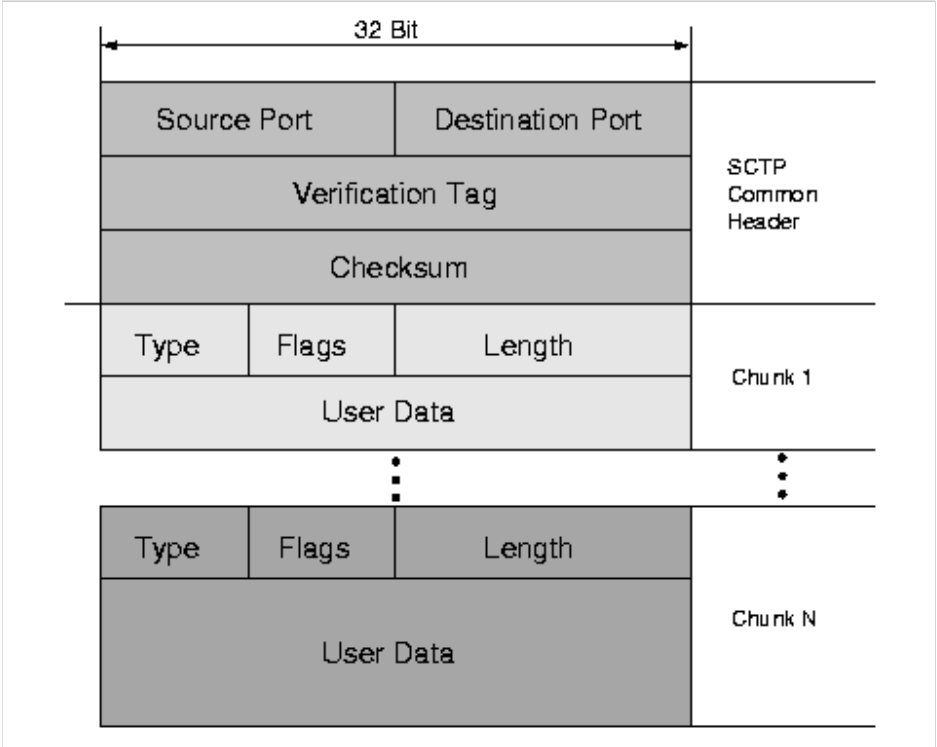
SCTP Packets

The protocol data units (PDU) of SCTP are called SCTP packets. If SCTP runs over IP (as described in RFC2960), an SCTP packet forms the payload of an IP packet. An SCTP packet is composed of a common header and chunks. Multiple chunks may be multiplexed into one packet up to the Path-MTU size. A chunk may contain either control information or user data.



An SCTP-Protocol Data Unit with several chunks

The Common Header

The common header consists of 12 bytes. For the identification of an association, SCTP uses the same port concept as TCP and UDP. For the detection of transmission errors, each SCTP packet is protected by a 32 bit checksum (Adler-32 algorithm), which is more robust than the 16 bit checksum of TCP and UDP. SCTP packets with an invalid checksum are silently discarded. The common header also contains a 32 bit value called **verification tag**. The verification tag is association specific, and exchanged between the endpoints at association startup. So there are two tag values used in one association. See section SCTP states for detailed information on tags.

Chunks

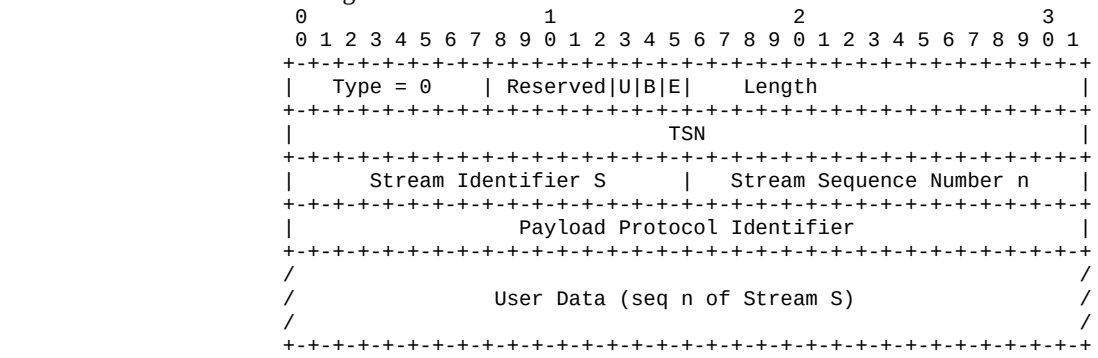
Each chunk begins with a chunk type field, which is used to distinguish data chunks and different types of control chunks, followed by chunk specific flags and a chunk length field needed because chunks have a variable length. The value field contains the actual payload of the chunk.

So far there are 13 chunk types defined for standard use. They are listed here. For the sake of simplicity, their definitions have been copied here from RFC2960 :

```
ID Value    Chunk Type
-----       ----------
0           - Payload Data (DATA)
1           - Initiation (INIT)
2           - Initiation Acknowledgement (INIT ACK)
3           - Selective Acknowledgement (SACK)
4           - Heartbeat Request (HEARTBEAT)
5           - Heartbeat Acknowledgement (HEARTBEAT ACK)
6           - Abort (ABORT)
7           - Shutdown (SHUTDOWN)
8           - Shutdown Acknowledgement (SHUTDOWN ACK)
9           - Operation Error (ERROR)
10          - State Cookie (COOKIE ECHO)
11          - Cookie Acknowledgement (COOKIE ACK)
12          - Reserved for Explicit Congestion Notification Echo (ECNE)
13          - Reserved for Congestion Window Reduced (CWR)
14          - Shutdown Complete (SHUTDOWN COMPLETE)
15 to 62    - reserved by IETF
63          - IETF-defined Chunk Extensions
64 to 126   - reserved by IETF
127         - IETF-defined Chunk Extensions
128 to 190  - reserved by IETF
191         - IETF-defined Chunk Extensions
192 to 254  - reserved by IETF
255         - IETF-defined Chunk Extensions
```

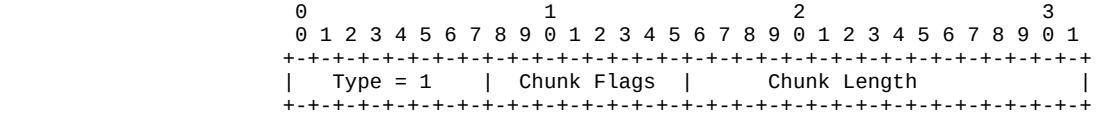
Payload Data

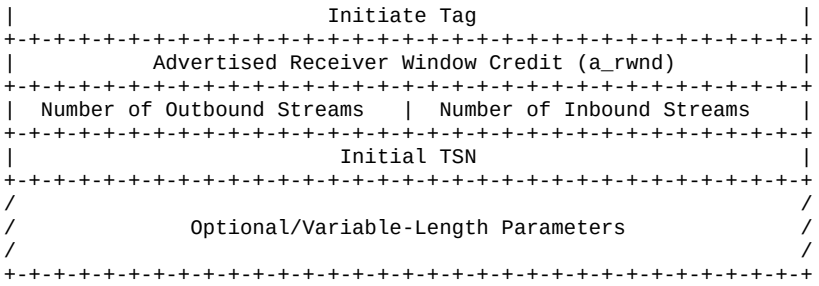
The following format MUST be used for the DATA chunk:



Initiation (INIT)

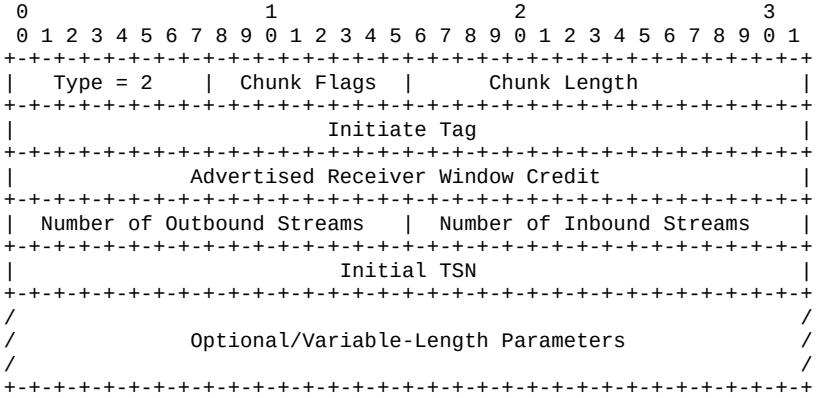
This chunk is used to initiate a SCTP association between two endpoints. The format of the INIT chunk is shown below:





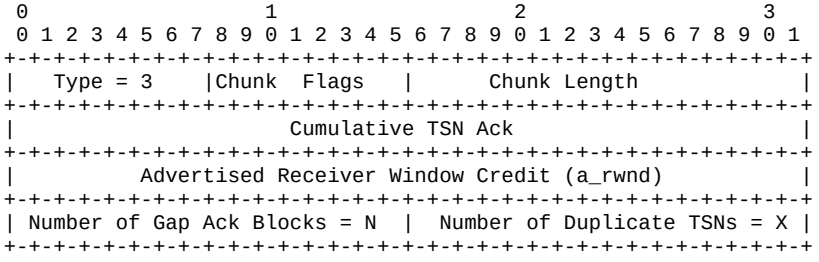
Initiation Acknowledgement (INIT ACK)

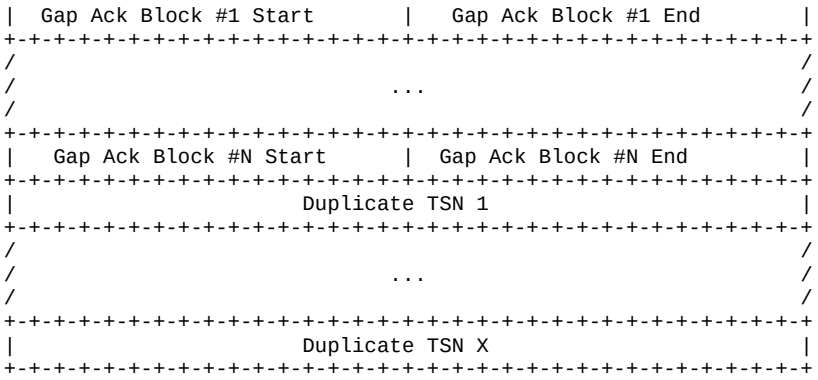
The INIT ACK chunk is used to acknowledge the initiation of an SCTP association. The parameter part of INIT ACK is formatted similarly to the INIT chunk. It uses two extra variable parameters: The State Cookie and the Unrecognized Parameter:



Selective Acknowledgement (SACK)

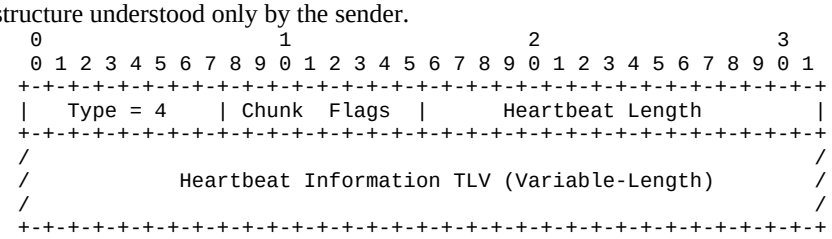
This chunk is sent to the peer endpoint to acknowledge received DATA chunks and to inform the peer endpoint of gaps in the received subsequences of DATA chunks as represented by their TSNs. The SACK MUST contain the Cumulative TSN Ack and Advertised Receiver Window Credit (a_rwnd) parameters.
By definition, the value of the Cumulative TSN Ack parameter is the last TSN received before a break in the sequence of received TSNs occurs; the next TSN value following this one has not yet been received at the endpoint sending the SACK. This parameter therefore acknowledges receipt of all TSNs less than or equal to its value.
The SACK also contains zero or more Gap Ack Blocks. Each Gap Ack Block acknowledges a subsequence of TSNs received following a break in the sequence of received TSNs. By definition, all TSNs acknowledged by Gap Ack Blocks are greater than the value of the Cumulative TSN Ack.





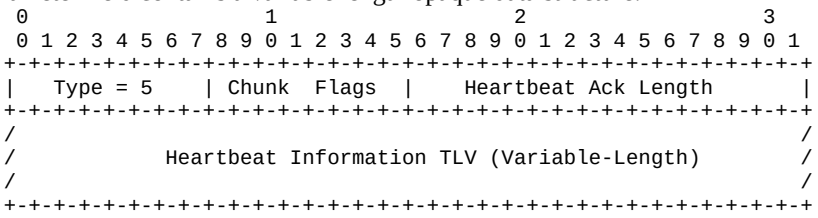
Heartbeat Request (HEARTBEAT)

An endpoint should send this chunk to its peer endpoint to probe the reachability of a particular destination transport address defined in the present association.
The parameter field contains the Heartbeat Information which is a variable length opaque data structure understood only by the sender.



Heartbeat Acknowledgement (HEARTBEAT ACK)

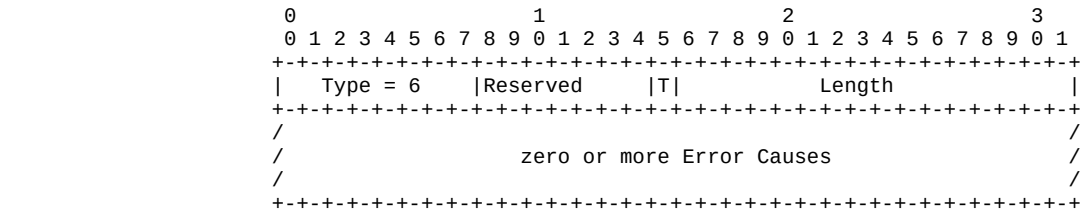
An endpoint should send this chunk to its peer endpoint as a response to a HEARTBEAT chunk (see Section 8.3). A HEARTBEAT ACK is always sent to the source IP address of the IP datagram containing the HEARTBEAT chunk to which this ack is responding.
The parameter field contains a variable length opaque data structure.



Abort Association (ABORT)

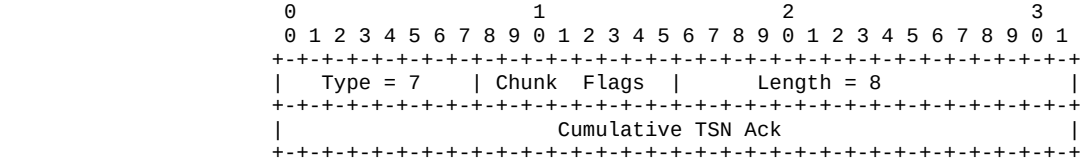
The ABORT chunk is sent to the peer of an association to close the association. The ABORT chunk may contain Cause Parameters to inform the receiver the reason of the abort. DATA chunks MUST NOT be bundled with ABORT. Control chunks (except for INIT, INIT ACK and SHUTDOWN COMPLETE) MAY be bundled with an ABORT but they MUST be placed before the ABORT in the SCTP packet, or they will be ignored by the receiver.
If an endpoint receives an ABORT with a format error or for an association that doesn't

exist, it MUST silently discard it. Moreover, under any circumstances, an endpoint that receives an ABORT MUST NOT respond to that ABORT by sending an ABORT of its own.



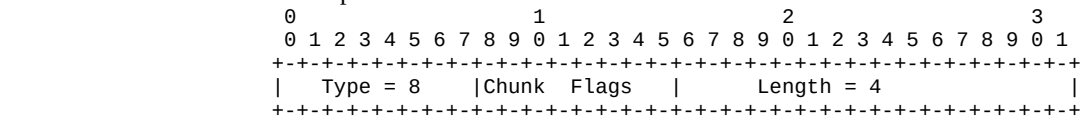
Shutdown Association (SHUTDOWN)

An endpoint in an association MUST use this chunk to initiate a graceful close of the association with its peer. This chunk has the following format.



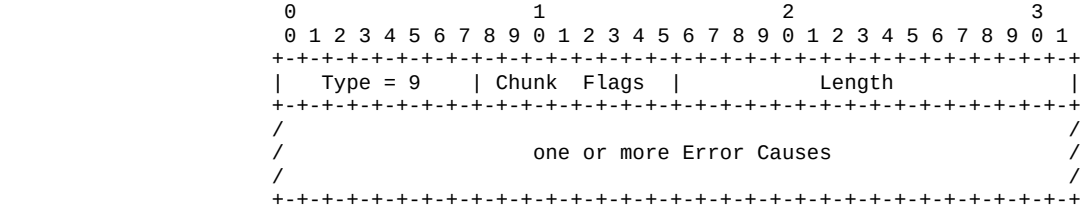
Shutdown Acknowledgement (SHUTDOWN ACK)

This chunk MUST be used to acknowledge the receipt of the SHUTDOWN chunk at the completion of the shutdown process, see Section 9.2 for details. The SHUTDOWN ACK chunk has no parameters.

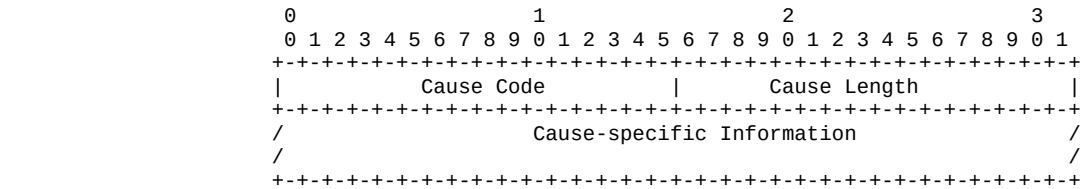


Operation Error (ERROR)

An endpoint sends this chunk to its peer endpoint to notify it of certain error conditions. It contains one or more error causes. An Operation Error is not considered fatal in and of itself, but may be used with an ABORT chunk to report a fatal condition.



It contains as parameters a variable length field containing the type of error that has occurred:

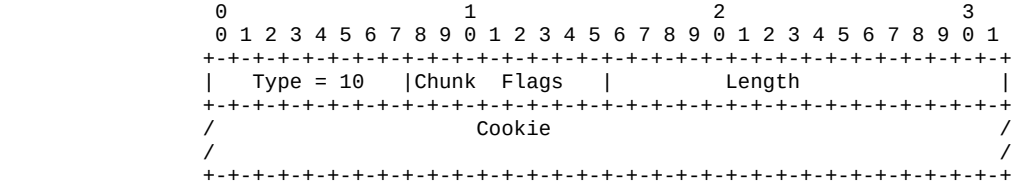


Each error cause may carry its own set of parameters. The error causes that have been defined are:

Cause Code Value	Cause Code
1	Invalid Stream Identifier
2	Missing Mandatory Parameter
3	Stale Cookie Error
4	Out of Resource
5	Unresolvable Address
6	Unrecognized Chunk Type
7	Invalid Mandatory Parameter
8	Unrecognized Parameters
9	No User Data
10	Cookie Received While Shutting Down

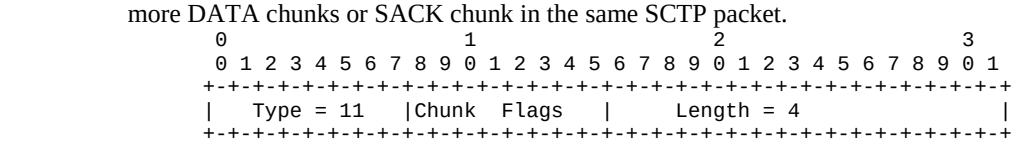
Cookie Echo (COOKIE ECHO)

This chunk is used only during the initialization of an association. It is sent by the initiator of an association to its peer to complete the initialization process. This chunk MUST precede any DATA chunk sent within the association, but MAY be bundled with one or more DATA chunks in the same packet.



Cookie Acknowledgement (COOKIE ACK)

This chunk is used only during the initialization of an association. It is used to acknowledge the receipt of a COOKIE ECHO chunk. This chunk MUST precede any DATA or SACK chunk sent within the association, but MAY be bundled with one or more DATA chunks or SACK chunk in the same SCTP packet.



Shutdown Complete (SHUTDOWN COMPLETE)

This chunk MUST be used to acknowledge the receipt of the SHUTDOWN ACK chunk at the completion of the shutdown process. The SHUTDOWN COMPLETE chunk has no parameters.