

# Protocol Compression

Compression is an optional feature of the Ice protocol; whether it is used for a particular message is determined by several factors:

1. Compression may not be supported on all platforms or in all language mappings.
2. Compression can be used in a request or batch request only if the endpoint [advertises](#) the ability to accept compressed messages.
3. For efficiency reasons, the Ice protocol engine does not compress messages smaller than 100 bytes.



A compliant implementation of the protocol is free to compress messages that are smaller than 100 bytes — the choice is up to the protocol implementation.

If compression is used, the entire message excluding the [header](#) is compressed using the bzip2 algorithm [1]. The `messageSize` member of the message header therefore reflects the size of the compressed message, including the uncompressed header, plus an additional four bytes.

The `compressionStatus` field of the message header indicates whether a message is compressed and whether the sender can accept a compressed reply, as shown in the table below.

Reply status	Encoding	Applies to	Description
Message is uncompressed, sender cannot accept a compressed reply.	0	Request, Batch Request, Reply, Validate Connection, Close Connection	A client that does not support compression always uses this value. A client that supports compression sets the value to 0 if the endpoint via which the request is dispatched indicates that it does not support compression. A server uses this value for uncompressed replies.
Message is uncompressed, sender can accept a compressed reply.	1	Request, Batch Request	A client uses this value if the endpoint via which the request is dispatched indicates that it supports compression, but the client has decided not to use compression for this particular request (presumably because the request is too small, so compression does not provide any saving).
Message is compressed and sender can accept a compressed reply.	2	Request, Batch Request, Reply	A client that supports compression sets this value only if the endpoint via which the request is dispatched indicates that it supports compression. A server uses this value for compressed replies.

The message body of a compressed request, batch request, or reply message is encoded by first writing the size of the uncompressed message (including its header) as a four-byte integer, followed by the compressed message body (excluding the header). It follows that the size of a compressed message is 14 bytes for the header, plus four bytes to record the size of the uncompressed message, plus the number of bytes occupied by the compressed message body. Writing the uncompressed message size prior to the body enables the receiver to allocate a buffer that is large enough to accommodate the uncompressed message body.

Note that compression is likely to improve performance only over lower-speed links, for which bandwidth is the overall limiting factor. Over high-speed LAN links, the CPU time spent on compressing and uncompressing messages is longer than the time it takes to just send the uncompressed data.

## See Also

- [Data Encoding for Proxies](#)
- [Protocol Messages](#)

## References

1. Red Hat, Inc. 2003. *The bzip2 and libbzip2 Home Page*. Raleigh, NC: Red Hat, Inc.