

Internet Control Message Protocol (ICMP)

ICMP Concepts and General Operation

The Internet Control Message Protocol (ICMP) is one of the under-appreciated “worker bees” of the networking world. Everyone knows how important key protocols such as the Internet Protocol are to TCP/IP, but few realize that the suite as a whole relies on many functions that ICMP provides. Originally created to allow the reporting of a small set of error conditions, ICMP messages are now used to implement a wide range of error-reporting, feedback and testing capabilities. While each message type is unique, they are all implemented using a common message format, sent and received based on relatively simple protocol rules. This makes ICMP one of the easiest TCP/IP protocols to understand. (Yes, I actually said something in this Guide was **easy**!)

In this section I provide a general description of ICMP. I begin with an overview of ICMP, discussing its purpose, history, and the versions and standards that define it. I describe the general method by which ICMP operates, and also discuss the rules that govern how and when ICMP messages are created and processed. I then outline the common format used for ICMP messages in ICMPv4 and ICMPv6, and how data is encapsulated in them in general terms. I conclude with a discussion of ICMP message classifications, and a summary of different message types and codes for both version 4 and version 6.

*Quick navigation to **subsections** and regular topics in this section*

- [ICMP Overview, History, Versions and Standards](#) (Parts: [1](#) [2](#) [3](#))
- [ICMP General Operation](#) (Parts: [1](#) [2](#))
- [ICMP Message Classes, Types and Codes](#) (Parts: [1](#) [2](#) [3](#))
- [ICMP Message Creation and Processing Conventions and Rules](#) (Parts: [1](#) [2](#) [3](#))
- [ICMP Common Message Format and Data Encapsulation](#) (Parts: [1](#) [2](#))

ICMP Overview, History, Versions and Standards

The Internet Protocol is the foundation of the TCP/IP protocol suite, since it is the mechanism responsible for delivering datagrams. Three of the main [characteristics that describe IP's datagram delivery method](#) are *connectionless*, *unreliable* and *unacknowledged*. This means that datagrams are “just sent” over the internetwork with no prior connection established, no assurance they will show up, and no acknowledgement sent back to the sender that they arrived. On the surface, this seems like it would result in a protocol that is difficult to use and

impossible to rely on, and therefore a poor choice for designing a protocol suite. However, even though IP “makes no guarantees”, it works very well because most of the time, IP internetworks are sufficiently robust that messages get where they need to go.

Even the best-designed system still encounters problems, of course. Incorrect packets are occasionally sent, hardware devices have problems, routes are found to be invalid, and so forth. IP devices also often need to share specific information to guide them in their operation, and to perform tests and diagnostics. However, IP itself includes no provision to allow devices to exchange low-level control messages. Instead, these features are provided in the form of a “companion” protocol to IP called the *Internet Control Message Protocol (ICMP)*.

The Relationship Between IP and ICMP

I think a good analogy for the relationship between IP and ICMP is to consider the one between a high-powered executive, and her experienced administrative assistant. The executive is busy and her time is very expensive. She is paid to do a specific job and to do it well, and not to spend time on administrative tasks. However, without **someone** doing those tasks, the executive could not do her job properly. The administrative assistant does the also-important support jobs that make it possible for the executive to focus on her work. The working relationship between them is very important; a good pair will work together like a cohesive team, even anticipating each others' needs.

In TCP/IP, the [Internet Protocol](#) is the executive, and ICMP is its “administrative assistant”. IP focuses on its core activities, such as addressing, datagram packaging and routing. ICMP provides critical support to IP in the form of *ICMP messages* that allow different types of communication to occur between IP devices. These messages use a common general format, and are encapsulated in IP datagrams for transmission. They are divided into different categories, and each type has a specific use and internal field format.

Just as an administrative assistant often has a special location in an organization chart, usually connecting with a “dotted line” directly to the executive he or she assists, ICMP occupies a unique place in the [TCP/IP protocol architecture](#). Technically, one might consider ICMP to belong to layer four, since it creates messages that are encapsulated in IP datagrams and sent using IP at layer three. However, in the standard that first defined it, ICMP is specifically declared to be not only part of the network layer, but:

“actually an integral part of IP, [that] must be implemented by every IP module”.

ICMP General Operation

ICMP is one of the simplest protocols in the TCP/IP protocol suite. Most protocols implement a particular type of functionality to either facilitate basic operation of a part of the network stack, or an application. To this end they include many specific algorithms and tasks that define the protocol, which is where most of the complexity lies. ICMP, in contrast, is exactly what its name suggests: a protocol that defines control messages. As such, pretty much all of what ICMP is about is providing a mechanism for any IP device to send control messages to another device.

The ICMP Message-Passing Service

Various message types are defined in ICMP that allow different types of information to be exchanged. These are usually either generated for the purpose of reporting errors, or for exchanging important information of different sorts that is needed to keep IP operating smoothly. ICMP itself doesn't define how all the different ICMP messages are used; this is done by the protocols that use the messages. In this manner, ICMP describes a simple message-passing service to other protocols.



Key Concept: ICMP is not like most other TCP/IP protocols in that it does not perform a specific task. It defines a mechanism by which various control messages can be transmitted and received to implement a variety of functions.

As mentioned in [the preceding overview topic](#), ICMP is considered an integral part of IP, even though it uses IP to send its messages. Typically, the operation of ICMP involves some portion of the TCP/IP protocol software on a machine detecting a condition that causes it to generate an ICMP message. This is often the IP layer itself, though it may be some other part of the software. The message is then encapsulated and transmitted like any other TCP/IP message, and is given no special treatment compared to other IP datagrams. The message is sent over the internetwork to the IP layer at the receiving device, as shown in Figure 137.

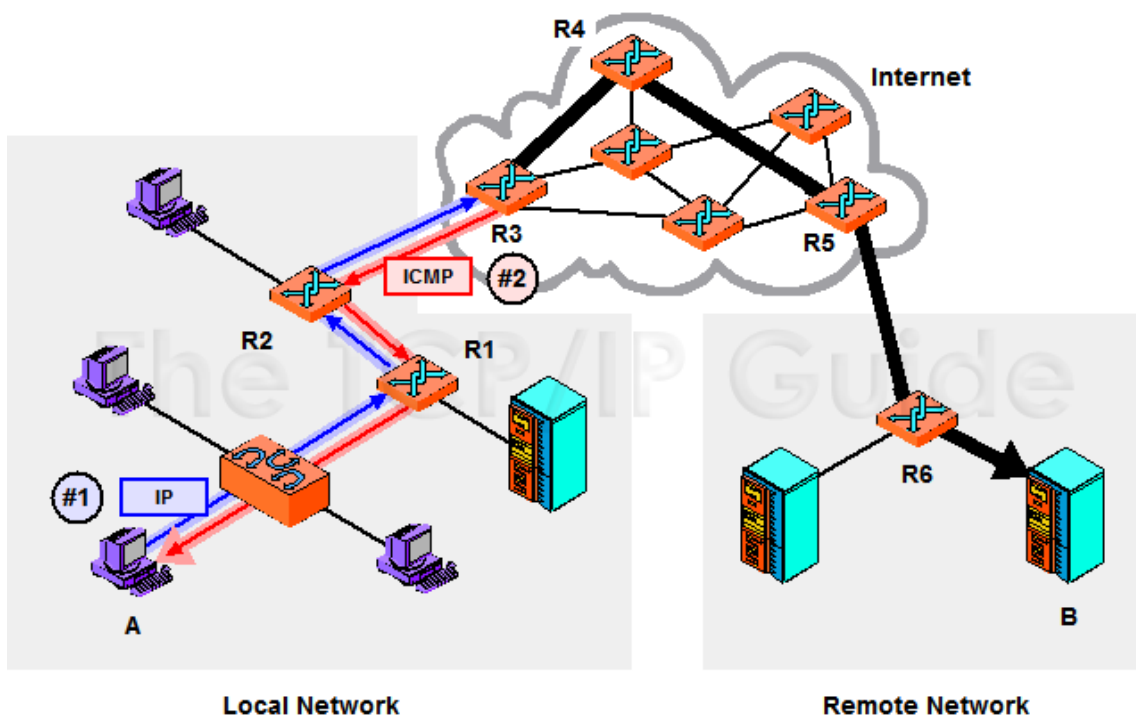


Figure 137: ICMP General Operation

A typical use of ICMP is to provide a feedback mechanism when an IP message is sent. In this example, device A is trying to send an IP datagram to device B.

However, when it gets to router R3 a problem of some sort is detected that causes the datagram to be dropped. R3 sends an ICMP message back to A to tell it that something happened, hopefully with enough information to let A correct the problem, if possible. R3 can only send the ICMP message back to A, not to R2 or R1.

Again, since many of the ICMP messages are actually intended to convey information to a device's IP software, the IP layer itself may be the “ultimate destination” of an ICMP message once a recipient gets it. In other cases, the ultimate destination may be some other part of the TCP/IP protocol software, which is determined by the type of message received. ICMP does not use ports like UDP or TCP to direct its messages to different applications on a host; the software recognizes the message type and directs it accordingly within the software.

ICMP was originally designed with the idea that most messages would be sent by routers, but they can be sent by both routers and by regular hosts as well, depending on the message type. Some are obviously only sent by routers, such as *Redirect* messages, while others may be sent by either routers or hosts. Many


of the ICMP messages are used in matched pairs, especially various kinds of *Request* and *Reply* messages, and *Advertisement* and *Solicitation* messages.

ICMP Error-Reporting Limitations

One interesting general characteristic of ICMP's operation is that when errors are detected, they can be reported using ICMP, but only back to the original source of a datagram. This is actually a big drawback in how ICMP works. Refer back to Figure 137 and consider again client host *A* sending a message to server host *B*, with a problem detected in the datagram by router *R3*. Even if *R3* suspects that the problem was caused by one of the preceding routers that handled the message, such as *R2*, it **cannot** send a problem report to *R2*. It can only send an ICMP message back to host *A*.

This limitation is an artifact of how the Internet Protocol works. You may recall from looking at [the IP datagram format](#) that the only address fields are for the original source and ultimate destination of the datagram. (The only exception is if the IP *Record Route* [option](#) is used, but devices cannot count on this.) When *R3* receives a datagram from *R2* that *R2* in turn received from *R1* (and prior to that, from *A*), it is only *A*'s address in the datagram. Thus, *R3* **must** send a problem report back to *A*, and *A* must decide what to do with it. Device *A* may decide to change the route it uses, or to generate an error report that an administrator can use to troubleshoot the *R2* router.

In addition to this basic limitation, several [special rules and conventions](#) have been put in place to govern the circumstances under which ICMP messages are generated, sent and processed.

 **Key Concept:** ICMP error-reporting messages sent in response to a problem seen in an IP datagram can only be sent back to the originating device. Intermediate devices cannot be the recipient of an ICMP message because their addresses are normally not carried in the IP datagram's header.

ICMP Message Classes, Types and Codes

ICMP messages are used to allow the communication of different types of information between IP devices on an internetwork. The messages themselves are used for a wide variety of purposes, and are organized into general categories, as well as numerous specific types and subtypes.

ICMP Message Classes

At the highest level, ICMP messages are divided into two classes:

- **Error Messages:** These messages are used to provide feedback to a source device about an error that has occurred. They are usually generated specifically in response to some sort of action, usually the transmission of a datagram, as shown in the example of Figure 137. Errors are usually related to the structure or content of a datagram, or to problem situations on the internetwork encountered during datagram routing.
- **Informational (or Query) Messages:** These are messages that are used to let devices exchange information, implement certain IP-related features, and perform testing. They do not indicate errors and are typically not sent in response to a regular datagram transmission. They are generated either when directed by an application, or on a regular basis to provide information to other devices. An informational ICMP message may also be sent in reply to another informational ICMP message, since they often occur in request/reply or solicitation/advertisement functional pairs.



Key Concept: ICMP messages are divided into two general categories: *error messages* that are used to report problem conditions, and *informational messages* that are used for diagnostics, testing and other purposes.

ICMP Message Types

Each individual kind of message in ICMP is given its own unique *Type* value, which is put into the field of that name in the ICMP common message format. This field is 8 bits wide, so a theoretical maximum of 256 message types can be defined. A separate set of *Type* values is maintained for each of ICMPv4 and ICMPv6.

In ICMPv4, *Type* values were assigned sequentially, to both error and informational messages, on a “first come, first served” basis (sort of) so one cannot tell just by the *Type* value what type of message each is. One minor improvement made in ICMPv6 was that the message types were separated. In IPv6, error messages have *Type* values from 0 to 127, and informational messages have values from 128 to 255. Of course, only some of the *Type* values are currently defined.



Key Concept: A total of 256 different possible message types can be defined for each of ICMPv4 and ICMPv6. The *Type* field that appears in the header of each message specifies the kind of ICMP message. In ICMPv4 there is no relationship between *Type* value and message type; in ICMPv6 error messages have a *Type* value of 0 to 127, informational messages 128 to 255.

ICMP Message Codes

The message type indicates the general purpose of each kind of ICMP message. ICMP also provides an additional level of detail within each message type in the form of a *Code* field, which is also 8 bits. You can consider this field as a message “subtype”. Thus, each message type can have up to 256 subtypes that are more detailed subdivisions of the message's overall functionality. A good example is the *Destination Unreachable* message, which is generated [when a datagram cannot be delivered](#). In this message type, the *Code* value provides more information on exactly why the delivery was not possible.

ICMP Message Class and Type Summary

A complete section describing all of the major ICMP message types for both ICMPv4 and ICMPv6 has been included in this Guide. For convenience, I have summarized all these message types in Table 86, which shows each of the *Type* values for the messages covered in this Guide, the name of each message, a very brief summary of its purpose, and the RFC that defines it. (To keep the table from being egregiously large I have not shown each of the *Code* values for each *Type* value; these can be found in the individual message type descriptions.) The table is organized into sections in the same way as the [ICMP Message Types and Formats section](#), except this table is sorted by ascending *Type* value within each category, for easier reference.

Table 86: ICMP Message Classes, Types and Codes				
Message Class	Type Value	Message Name	Summary Description of Message Type	Defining RFC Number
ICMPv4 Error Messages	3	<i>Destination Unreachable</i>	Indicates that a datagram could not be delivered to its destination. The <i>Code</i> value provides more information on the nature of the error.	792
	4	<i>Source Quench</i>	Lets a congested IP device tell a device that is sending it datagrams to slow down the rate at which it is sending them.	792
	5	<i>Redirect</i>	Allows a router to inform a	792

			host of a better route to use for sending datagrams.	
	11	<i>Time Exceeded</i>	Sent when a datagram has been discarded prior to delivery due to expiration of its <i>Time To Live</i> field.	792
	12	<i>Parameter Problem</i>	Indicates a miscellaneous problem (specified by the <i>Code</i> value) in delivering a datagram.	792
ICMPv4 Informational Messages (part 1 of 2)	0	<i>Echo Reply</i>	Sent in reply to an <i>Echo (Request)</i> message; used for testing connectivity.	792
	8	<i>Echo (Request)</i>	Sent by a device to test connectivity to another device on the internetwork. The word "Request" sometimes appears in the message name.	792
	9	<i>Router Advertisement</i>	Used by routers to tell hosts of their existence and capabilities.	1256
	10	<i>Router Solicitation</i>	Used by hosts to prompt any listening routers to send a <i>Router Advertisement</i> .	1256
	13	<i>Timestamp (Request)</i>	Sent by a device to request that another send it a timestamp value for propagation time calculation and clock synchronization. The word "Request" sometimes appear in the message name.	792
	14	<i>Timestamp</i>	Sent in response to a	792

		Reply	<i>Timestamp (Request)</i> to provide time calculation and clock synchronization information.	
	15	Information Request	Originally used to request configuration information from another device. Now obsolete.	792
ICMPv4 Informational Messages (part 2 of 2)	16	Information Reply	Originally used to provide configuration information in response to an <i>Information Request</i> message. Now obsolete.	792
	17	Address Mask Request	Used to request that a device send a subnet mask.	950
	18	Address Mask Reply	Contains a subnet mask sent in reply to an <i>Address Mask Request</i> .	950
	30	Traceroute	Used to implement the experimental “enhanced” <i>traceroute</i> utility.	1393
ICMPv6 Error Messages	1	Destination Unreachable	Indicates that a datagram could not be delivered to its destination. <i>Code</i> value provides more information on the nature of the error.	2463
	2	Packet Too Big	Sent when a datagram cannot be forwarded because it is too big for the maximum transmission unit (MTU) of the next hop in the route. This message is needed in IPv6 and not IPv4 because in IPv4, routers can fragment oversized messages, while in IPv6 they cannot.	2463
	3	Time	Sent when a datagram	2463

		<i>Exceeded</i>	has been discarded prior to delivery due to the <i>Hop Limit</i> field being reduced to zero.	
	4	<i>Parameter Problem</i>	Indicates a miscellaneous problem (specified by the <i>Code</i> value) in delivering a datagram.	2463
ICMPv6 Informational Messages	128	<i>Echo Request</i>	Sent by a device to test connectivity to another device on the internetwork.	2463
	129	<i>Echo Reply</i>	Sent in reply to an <i>Echo (Request)</i> message; used for testing connectivity.	2463
	133	<i>Router Solicitation</i>	Prompts a router to send a <i>Router Advertisement</i> .	2461
	134	<i>Router Advertisement</i>	Sent by routers to tell hosts on the local network the router exists and describe its capabilities.	2461
	135	<i>Neighbor Solicitation</i>	Sent by a device to request the layer two address of another device while providing its own as well.	2461
	136	<i>Neighbor Advertisement</i>	Provides information about a host to other devices on the network.	2461
	137	<i>Redirect</i>	Redirects transmissions from a host to either an immediate neighbor on the network or a router.	2461
	138	<i>Router Renumbering</i>	Conveys renumbering information for router renumbering.	2894

You can see that several of the message types are quite similar in ICMPv4 and ICMPv6, but there are some slight differences. An obvious one is that *Redirect* is considered an error message in ICMPv4, but an informational message in ICMPv6. The way that the messages is used also often different. In IPv6, the use of many of the ICMP informational messages is described in the [Neighbor Discovery \(ND\) protocol](#), which is new to IPv6.

Note that the *Information Request* and *Information Reply* messages were originally created to allow devices to determine an IP address and possibly other configuration information. This function was later implemented using protocols such as [RARP](#), [BOOTP](#) and [DHCP](#), and these message types obsoleted.

ICMP Common Message Format and Data Encapsulation

As we have seen in the preceding topics, ICMP is not so much a protocol that performs a specific function as a [framework for the exchange of error reports and information](#). Since each of the message types is used for a different purpose, they differ in the types of information each contains. This means each ICMP message has a slightly different format. At the same time, however, ICMP message types also have a degree of commonality—a portion of each message is common between message types.

ICMP Common Message Format

The structure of an ICMP message can be generally thought of as having a *common part* and a *unique part*. The common part consists of three fields that have the same size and same meaning in all ICMP messages (though the values in the fields aren't the same for each ICMP message type, of course). The unique part contains fields that are specific to each type of message.

Interestingly, the common message format is basically the same for ICMPv4 and ICMPv6. It is described in Table 87 and Figure 138.

Table 87: ICMP Common Message Format		
Field Name	Size (bytes)	Description
Type	1	Type: Identifies the ICMP message type. For ICMPv6, values from 0 to 127 are error messages and values 128 to 255 are informational messages. Common values for this field are given in the table in the topic on ICMP message classes and types .

Code	1	Code: Identifies the “subtype” of message within each ICMP message Type value. Thus, up to 256 “subtypes” can be defined for each message type. Values for this field are shown in the individual ICMP message type topics .
Checksum	2	Checksum: 16-bit checksum field that is calculated in a manner similar to the IP header checksum in IPv4 . It provides error detection coverage for the entire ICMP message. Note that in ICMPv6, a pseudo-header of IPv6 header fields is prepended for checksum calculation; this is similar to the way this is done in TCP .
Message Body / Data	Variable	Message Body: Contains the specific fields used to implement each message type. This is the unique part of the message as I mentioned above.

The message body typically contains one or several fields that carry information of relevance to each specific type of ICMP message. For error messages, we see here one more area of commonality in ICMP messages: all ICMP error messages include a portion of the original IP datagram that led to the ICMP error message. This aids in diagnosing the problem that caused the ICMP message to be generated, by allowing the error to be communicated to higher layers.

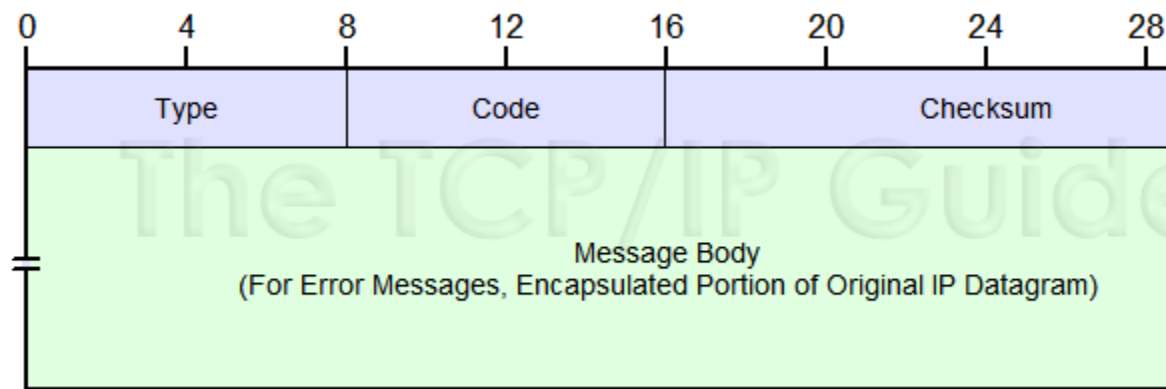


Figure 138: ICMP Common Message Format

This overall, generic message format is used for both ICMPv4 and ICMPv6 message types.

Original Datagram Inclusion In ICMP Error Messages

The inclusion of original IP datagram information is done differently for the two ICMP versions.

ICMPv4 Error Messages

Each error message includes the full IP header and the first 8 bytes of the payload. Since the beginning of the payload will contain the encapsulated higher-layer header, this means the ICMP message also carries either the full UDP header, or the first 8 bytes of the TCP header. In both cases, the [source and destination port numbers](#) are part of what is included.

If the original header was a standard IP header with no options, the *Message Body* will therefore have a length of 28 bytes; if options are present, it will be larger.

ICMPv6 Error Messages

Each error message includes as much of the IPv6 datagram as will fit without causing the size of the ICMPv6 error message (including its IP header encapsulation) to exceed the [minimum IPv6 maximum transmission unit size](#), which is 1280 bytes. This provides additional information for diagnostic purposes compared to ICMPv4, while ensuring that no ICMPv6 error messages will be too large for any physical network segment. The larger size of the included data allows the IPv6 extension headers to be included in the error message, since the error could be in one of those extension headers.

Remember that [in IPv6, routers cannot fragment IP datagrams](#); any datagram that is “oversized” for an underlying physical network is dropped. ICMPv6 is thus designed to ensure that this does not happen by not creating ICMPv6 datagrams over the universal IPv6 MTU size of 1280.



Key Concept: Each kind of ICMP message contains data unique to that message type, but all messages are structured according to a common ICMP message format. ICMP error messages always include in their message body field some portion of the original IP datagram that resulted in the error being generated.