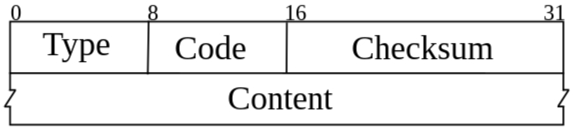
Internet Control Message Protocol

The **Internet Control Message Protocol** (**ICMP**) is a supporting protocol in the Internet protocol suite. It is used by network devices, including routers, to send error messages and operational information indicating success or failure when communicating with another IP address, for example, an error is indicated when a requested service is not available or that a host or router could not be reached.[2] ICMP differs from transport protocols such as TCP and UDP in that it is not typically used to exchange data between systems, nor is it regularly employed by end-user network applications (with the exception of some diagnostic tools like ping and traceroute).

Internet Control Message Protocol

Communication protocol



A general header for ICMPv4

Purpose	Auxiliary protocol for IPv4[1]
Developer(s)	DARPA
Introduction	1981
OSI layer	Network layer
RFC(s)	RFC 792

ICMP for IPv4 is defined in RFC 792. A separate ICMPv6, defined by RFC 4443, is used with IPv6.

Technical details

ICMP is part of the Internet protocol suite as defined in RFC 792. ICMP messages are typically used for diagnostic or control purposes or generated in response to errors in IP operations (as specified in RFC 1122). ICMP errors are directed to the source IP address of the originating packet.[2]

For example, every device (such as an intermediate router) forwarding an IP datagram first decrements the time to live (TTL) field in the IP header by one. If the resulting TTL is 0, the packet is discarded and an ICMP time exceeded in transit message is sent to the datagram's source address.

Many commonly used network utilities are based on ICMP messages. The traceroute command can be implemented by transmitting IP datagrams with specially set IP TTL header fields, and looking for ICMP time exceeded in transit and Destination unreachable messages generated in response. The related ping utility is implemented using the ICMP *echo request* and *echo reply* messages.

ICMP uses the basic support of IP as if it were a higher-level protocol, however, ICMP is actually an integral part of IP. Although ICMP messages are contained within standard IP packets, ICMP messages are usually processed as a special case, distinguished from normal IP processing. In many cases, it is necessary to inspect the contents of the ICMP message and deliver the appropriate error message to the application responsible for transmitting the IP packet that prompted the ICMP message to be sent.

ICMP is a network-layer protocol. There is no TCP or UDP port number associated with ICMP packets as these numbers are associated with the transport layer above.[3]

Datagram structure

The ICMP packet is encapsulated in an IPv4 packet.[2] The packet consists of header and data sections.

Header

The ICMP header starts after the IPv4 header and is identified by IP protocol number '1'.[4] All ICMP packets have an 8-byte header and variable-sized data section. The first 4 bytes of the header have fixed format, while the last 4 bytes depend on the type/code of that ICMP packet.[2]

ICMP header format

Offsets	Octet	0								1								2									
Octet	**Bit**	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	**0**	Type								Code								Checksum									
4	**32**	Rest of header																									

Type

ICMP type, see § Control messages.

Code

ICMP subtype, see § Control messages.

Checksum

Internet checksum (RFC 1071) for error checking, calculated from the ICMP header and data with value 0 substituted for this field.

Rest of header

Four-byte field, contents vary based on the ICMP type and code.

Data

ICMP error messages contain a data section that includes a copy of the entire IPv4 header, plus at least the first eight bytes of data from the IPv4 packet that caused the error message. The length of ICMP error messages should not exceed 576 bytes.[5] This data is used by the host to match the message to the appropriate process. If a higher level protocol uses port numbers, they are assumed to be in the first eight bytes of the original datagram's data.[6]

The variable size of the ICMP packet data section has been exploited. In the "Ping of death", large or fragmented ICMP packets are used for denial-of-service attacks. ICMP data can also be used to create covert channels for communication. These channels are known as ICMP tunnels.

Control messages

Control messages are identified by the value in the *type* field. The *code* field gives additional context information for the message. Some control messages have been deprecated since the protocol was first introduced.

Notable control messages[7][8]

Type	Code	Status	Description
0 – Echo Reply[6]:14	0		Echo reply (used to ping)
1 and 2		unassigned	*Reserved*
3 – Destination Unreachable[6]:4	0		Destination network unreachable
	1		Destination host unreachable
	2		Destination protocol unreachable
	3		Destination port unreachable
	4		Fragmentation required, and DF flag set
	5		Source route failed
	6		Destination network unknown
	7		Destination host unknown
	8		Source host isolated
	9		Network administratively prohibited
	10		Host administratively prohibited
	11		Network unreachable for ToS
	12		Host unreachable for ToS
	13		Communication administratively prohibited
	14		Host Precedence Violation
	15		Precedence cutoff in effect
4 – Source Quench	0	deprecated	Source quench (congestion control)
5 – Redirect Message	0		Redirect Datagram for the Network
	1		Redirect Datagram for the Host
	2		Redirect Datagram for the ToS & network
	3		Redirect Datagram for the ToS & host
6		deprecated	Alternate Host Address
7		unassigned	*Reserved*
8 – Echo Request	0		Echo request (used to ping)

9 – Router Advertisement	0		Router Advertisement
10 – Router Solicitation	0		Router discovery/selection/solicitation
11 – Time Exceeded[6]:6	0		TTL expired in transit
	1		Fragment reassembly time exceeded
12 – Parameter Problem: Bad IP header	0		Pointer indicates the error
	1		Missing a required option
	2		Bad length
13 – Timestamp	0		Timestamp
14 – Timestamp Reply	0		Timestamp reply
15 – Information Request	0	deprecated	Information Request
16 – Information Reply	0	deprecated	Information Reply
17 – Address Mask Request	0	deprecated	Address Mask Request
18 – Address Mask Reply	0	deprecated	Address Mask Reply
19		reserved	*Reserved* for security
20 through 29		reserved	*Reserved* for robustness experiment
30 – Traceroute	0	deprecated	Information Request
31		deprecated	Datagram Conversion Error
32		deprecated	Mobile Host Redirect
33		deprecated	Where-Are-You (originally meant for IPv6)
34		deprecated	Here-I-Am (originally meant for IPv6)
35		deprecated	Mobile Registration Request
36		deprecated	Mobile Registration Reply

37		deprecated	Domain Name Request
38		deprecated	Domain Name Reply
39		deprecated	SKIP Algorithm Discovery Protocol, Simple Key-Management for Internet Protocol
40			Photuris, Security failures
41		Experimental	ICMP for experimental mobility protocols such as Seamoby [RFC4065]
42 – Extended Echo Request[9]	0		Request Extended Echo (XPing - see **Extended Ping (Xping)** (https://tools.ietf.org/html/draft-bonica-intarea-eping-04))
43 – Extended Echo Reply[9]	0		No Error
	1		Malformed Query
	2		No Such Interface
	3		No Such Table Entry
	4		Multiple Interfaces Satisfy Query
44 through 252		unassigned	*Reserved*
253		Experimental	RFC3692-style Experiment 1 (RFC 4727)
254		Experimental	RFC3692-style Experiment 2 (RFC 4727)
255		reserved	Reserved

Source quench

Source Quench requests that the sender decrease the rate of messages sent to a router or host. This message may be generated if a router or host does not have sufficient buffer space to process the request, or may occur if the router or host buffer is approaching its limit.

Data is sent at a very high speed from a host or from several hosts at the same time to a particular router on a network. Although a router has buffering capabilities, the buffering is limited to within a specified range. The router cannot queue any more data than the capacity of the limited buffering space. Thus if the queue gets filled up, incoming data is discarded until the queue is no longer full. But as no acknowledgement mechanism is present in the network layer, the client does not know whether the data has reached the destination successfully. Hence some remedial measures should be taken by the network layer to avoid these kind of situations. These measures are referred to as source quench. In a source quench mechanism, the router sees that the incoming data rate is much faster than the outgoing data rate, and sends an ICMP message to the clients, informing them that they should slow down their data transfer speeds or wait for a certain amount of time before attempting to send more data. When a client receives this message, it will automatically slow down the outgoing data rate or wait for a sufficient amount of time, which enables the router to empty the queue. Thus the source quench ICMP message acts as flow control in the network layer.

Since research suggested that "ICMP Source Quench [was] an ineffective (and unfair) antidote for congestion",[10] routers' creation of source quench messages was deprecated in 1995 by RFC 1812. Furthermore, forwarding of and any kind of reaction to (flow control actions) source quench messages was deprecated from 2012 by RFC 6633.

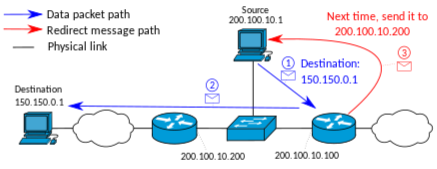
Source quench message[6]:9

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	2
Type = 4								Code = 0								Checksum										
unused																										
IP header and first 8 bytes of original datagram's data																										

Where:

Type must be set to 4
Code must be set to 0
IP header and additional data is used by the sender to match the reply with the associated request

Redirect



An example of how an ICMPv4 redirect message works

Redirect requests data packets be sent on an alternative route. ICMP Redirect is a mechanism for routers to convey routing information to hosts. The message informs a host to update its routing information (to send packets on an alternative route). If a host tries to send data through a router (R1) and R1 sends the data on another router (R2) and a direct path from the host to R2 is available (that is, the host and R2 are on the same subnetwork), then R1 will send a redirect message to inform the host that the best route for the destination is via R2. The host should then change its route information and send packets for that destination directly to R2. The router will still send the original datagram to the intended destination.[11] However, if the datagram contains routing information, this message will not be sent even if a better route is available. RFC 1122 states that redirects should only be sent by gateways and should not be sent by Internet hosts.

Redirect message[6]:11

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	2
Type = 5								Code								Checksum										
IP address																										
IP header and first 8 bytes of original datagram's data																										

Where:

Type must be set to 5.
Code specifies the reason for the redirection, and may be one of the following:

Code	Description
0	Redirect for Network
1	Redirect for Host
2	Redirect for Type of Service and Network
3	Redirect for Type of Service and Host

IP address is the 32-bit address of the gateway to which the redirection should be sent.
IP header and additional data is included to allow the host to match the reply with the request that caused the redirection reply.

Time exceeded

Time Exceeded is generated by a gateway to inform the source of a discarded datagram due to the time to live field reaching zero. A time exceeded message may also be sent by a host if it fails to reassemble a fragmented datagram within its time limit.

Time exceeded messages are used by the traceroute utility to identify gateways on the path between two hosts.

Time exceeded message[6]: 5

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	2
Type = 11								Code								Checksum										
unused																										
IP header and first 8 bytes of original datagram's data																										

Where:

Type must be set to 11
Code specifies the reason for the *time exceeded* message, include the following:

Code	Description
0	Time-to-live exceeded in transit.
1	Fragment reassembly time exceeded.

IP header and first 64 bits of the original payload are used by the source host to match the time exceeded message to the discarded datagram. For higher-level protocols such as UDP and TCP the 64-bit payload will include the source and destination ports of the discarded packet.

Timestamp

Timestamp is used for time synchronization. The originating timestamp is set to the time (in milliseconds since midnight) the sender last touched the packet. The receive and transmit timestamps are not used.

Timestamp message[6]:15

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	2
Type = 13								Code = 0								Checksum										
Identifier																Sequence number										
Originate timestamp																										
Receive timestamp																										
Transmit timestamp																										

Where:

Type must be set to 13

Code must be set to 0

Identifier and **Sequence Number** can be used by the client to match the timestamp reply with the timestamp request.

Originate timestamp is the number of milliseconds since midnight Universal Time (UT). If a UT reference is not available the most-significant bit can be set to indicate a non-standard time value.

Timestamp reply

Timestamp Reply replies to a *Timestamp* message. It consists of the originating timestamp sent by the sender of the *Timestamp* as well as a receive timestamp indicating when the *Timestamp* was received and a transmit timestamp indicating when the *Timestamp reply* was sent.

Timestamp reply message[6]:15

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	2
Type = 14								Code = 0								Checksum										
Identifier																Sequence number										
Originate timestamp																										
Receive timestamp																										
Transmit timestamp																										

Where:

Type must be set to 14

Code must be set to 0
Identifier and **Sequence number** can be used by the client to match the reply with the request that caused the reply.
Originate timestamp is the time the sender last touched the message before sending it.
Receive timestamp is the time the echoer first touched it on receipt.
Transmit timestamp is the time the echoer last touched the message on sending it.
All timestamps are in units of milliseconds since midnight UT. If the time is not available in milliseconds or cannot be provided with respect to midnight UT then any time can be inserted in a timestamp provided the high order bit of the timestamp is also set to indicate this non-standard value.

The use of Timestamp and Timestamp Reply messages to synchronize the clocks of Internet nodes has largely been replaced by the UDP-based Network Time Protocol and the Precision Time Protocol.[12]

Address mask request

Address mask request is normally sent by a host to a router in order to obtain an appropriate subnet mask.

Recipients should reply to this message with an *Address mask reply* message.

Address mask request

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	2
Type = 17								Code = 0								Checksum										
Identifier																Sequence number										
Address mask																										

Where:

Type must be set to 17
Code must be set to 0
Address mask can be set to 0

ICMP Address Mask Request may be used as a part of reconnaissance attack to gather information on the target network, therefore ICMP Address Mask Reply is disabled by default on Cisco IOS.[13]

Address mask reply

Address mask reply is used to reply to an address mask request message with an appropriate subnet mask.

Address mask reply

<table>
<tr><th>00</th><th>01</th><th>02</th><th>03</th><th>04</th><th>05</th><th>06</th><th>07</th><th>08</th><th>09</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>2</th></tr>
<tr><td colspan="8">Type = 18</td><td colspan="8">Code = 0</td><td colspan="11">Checksum</td></tr>
<tr><td colspan="16">Identifier</td><td colspan="11">Sequence number</td></tr>
<tr><td colspan="27">Address mask</td></tr>
</table>

Where:

- **Type** must be set to 18
- **Code** must be set to 0
- **Address mask** should be set to the subnet mask

Destination unreachable

Destination unreachable is generated by the host or its inbound gateway[6] to inform the client that the destination is unreachable for some reason. Reasons for this message may include: the physical connection to the host does not exist (distance is infinite); the indicated protocol or port is not active; the data must be fragmented but the 'don't fragment' flag is on. Unreachable TCP ports notably respond with **TCP RST** rather than a *destination unreachable* type 3 as might be expected. *Destination unreachable* is never reported for IP multicast transmissions.

Destination unreachable message[6]:3

<table>
<tr><th>00</th><th>01</th><th>02</th><th>03</th><th>04</th><th>05</th><th>06</th><th>07</th><th>08</th><th>09</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>2</th></tr>
<tr><td colspan="8">Type = 3</td><td colspan="8">Code</td><td colspan="11">Checksum</td></tr>
<tr><td colspan="16">unused</td><td colspan="11">Next-hop MTU</td></tr>
<tr><td colspan="27">IP header and first 8 bytes of original datagram's data</td></tr>
</table>

Where:

- **Type** field (bits 0–7) must be set to 3
- **Code** field (bits 8–15) is used to specify the type of error, and can be any of the following:

Code	Description
0	Network unreachable error.
1	Host unreachable error.
2	Protocol unreachable error (the designated transport protocol is not supported).
3	Port unreachable error (the designated protocol is unable to inform the host of the incoming message).
4	The datagram is too big. Packet fragmentation is required but the 'don't fragment' (DF) flag is on.
5	Source route failed error.
6	Destination network unknown error.
7	Destination host unknown error.
8	Source host isolated error.
9	The destination network is administratively prohibited.
10	The destination host is administratively prohibited.
11	The network is unreachable for Type Of Service.
12	The host is unreachable for Type Of Service.
13	Communication administratively prohibited (administrative filtering prevents packet from being forwarded).
14	Host precedence violation (indicates the requested precedence is not permitted for the combination of host or network and port).
15	Precedence cutoff in effect (precedence of datagram is below the level set by the network administrators).

Next-hop MTU field (bits 48–63) contains the MTU of the next-hop network if a code 4 error occurs.[14]

IP header and additional data is included to allow the client to match the reply with the request that caused the destination unreachable reply.

See also

- ICMP tunnel
- ICMP hole punching

- ICMP Router Discovery Protocol
- Pathping
- Path MTU Discovery
- Smurf attack

References

1. *F. Baker (June 1995). Baker, F (ed.). Requirements for IP Version 4 Routers. p. 52. RFC 1812 (https://tools.ietf.org/html/rfc1812) .*
2. *Forouzan, Behrouz A. (2007). Data Communications And Networking (https://archive.org/details/datacommunicatio00foro_184) (Fourth ed.). Boston: McGraw-Hill. pp. 621 (https://archive.org/details/datacommunicatio00foro_184/page/n657) –630. ISBN 978-0-07-296775-3.*
3. *"The OSI Model's Seven Layers Defined and Functions Explained" (https://support.microsoft.com/kb/103884) . Microsoft Support. Retrieved 2014-12-28.*
4. *"Protocol Numbers" (https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml) . Internet Assigned Numbers Authority. Retrieved 2011-06-23.*
5. *Requirements for IP Version 4 Routers (https://datatracker.ietf.org/doc/html/rfc1812) . doi:10.17487/RFC1812 (https://doi.org/10.17487%2FRFC1812) . RFC 1812 (https://datatracker.ietf.org/doc/html/rfc1812) .*
6. *Postel, J. (September 1981). Internet Control Message Protocol (https://datatracker.ietf.org/doc/html/rfc792) . IETF. doi:10.17487/RFC0792 (https://doi.org/10.17487%2FRFC0792) . RFC 792 (https://datatracker.ietf.org/doc/html/rfc792) .*
7. *"IANA ICMP Parameters" (https://www.iana.org/assignments/icmp-parameters) . Iana.org. 2012-09-21. Retrieved 2013-01-07.*
8. *Kurose, J.F; Ross, K.W. (2006). Computer Networking: A Top-Down Approach (https://books.google.com/books?id=QXIwPwAACAAJ) . World student series. Addison-Wesley. ISBN 9780321418494.*
9. *PROBE: A Utility for Probing Interfaces (https://datatracker.ietf.org/doc/html/rfc8335) . doi:10.17487/RFC8335 (https://doi.org/10.17487%2FRFC8335) . RFC 8335 (https://datatracker.ietf.org/doc/html/rfc8335) .*
10. *RFC 6633 (https://datatracker.ietf.org/doc/html/rfc6633)*
11. *"When Are ICMP Redirects Sent?" (http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094702.shtml) . Cisco Systems. 2008-06-28. Retrieved 2013-08-15.*
12. *D.L. Mills (September 1985). Network Time Protocol (NTP) (https://datatracker.ietf.org/doc/html/rfc958) . doi:10.17487/RFC0958 (https://doi.org/10.17487%2FRFC0958) . RFC 958 (https://datatracker.ietf.org/doc/html/rfc958) . "It is evolved from the Time Protocol and the ICMP Timestamp message and is a suitable replacement for both."*

13. "Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services, Release 12.3 - IP Addressing and Services Commands: ip mask-reply through ip web-cache" (https://web.archive.org/web/20130102124241/http://www.cisco.com/en/US/docs/ios/12_3/ipaddr/command/reference/ip1_i2g.html#wp1078496) . Cisco Systems. Archived from the original (http://www.cisco.com/en/US/docs/ios/12_3/ipaddr/command/reference/ip1_i2g.html#wp1078496) on 2013-01-02. Retrieved 2013-01-07.

14. Extended ICMP to Support Multi-Part Messages (https://datatracker.ietf.org/doc/html/rfc4884) . doi:10.17487/RFC4884 (https://doi.org/10.17487%2FRFC4884) . RFC 4884 (https://datatracker.ietf.org/doc/html/rfc4884) .

Sources

RFCs

- RFC 792 (https://datatracker.ietf.org/doc/html/rfc792) , *Internet Control Message Protocol*
- RFC 950 (https://datatracker.ietf.org/doc/html/rfc950) , *Internet Standard Subnetting Procedure*
- RFC 1016 (https://datatracker.ietf.org/doc/html/rfc1016) , *Something a Host Could Do with Source Quench: The Source Quench Introduced Delay (SQuID)*
- RFC 1122 (https://datatracker.ietf.org/doc/html/rfc1122) , *Requirements for Internet Hosts – Communication Layers*
- RFC 1716 (https://datatracker.ietf.org/doc/html/rfc1716) , *Towards Requirements for IP Routers*
- RFC 1812 (https://datatracker.ietf.org/doc/html/rfc1812) , *Requirements for IP Version 4 Routers*
- RFC 4884 (https://datatracker.ietf.org/doc/html/rfc4884) , *Extended ICMP to Support Multi-Part Messages*

External links

Wikiversity has learning resources about ***Internet Control Message Protocol***

- IANA ICMP parameters (https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml)
- IANA protocol numbers (https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml)

- Explanation of ICMP Redirect Behavior (https://web.archive.org/web/20150110205151/http://support.microsoft.com/kb/195686) at the Wayback Machine (archived 2015-01-10)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Internet_Control_Message_Protocol&oldid=1131465421"