Internet Control Message Protocol (ICMP)

The IP protocol has **no error-reporting** or **error-correcting** mechanism.

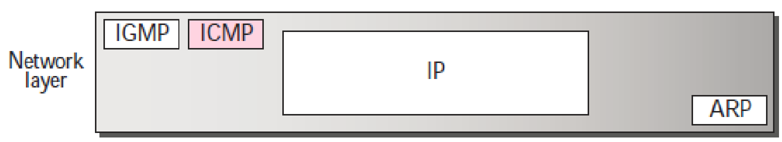
- **What happens if** something goes wrong?
- **What happens if** a router must discard a datagram because it cannot find a router to the final destination, or because the time-to-live field has a zero value?
- **What happens if** the final destination host must discard all fragments of a datagram because it has not received all fragments within a predetermined time limit?

These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.

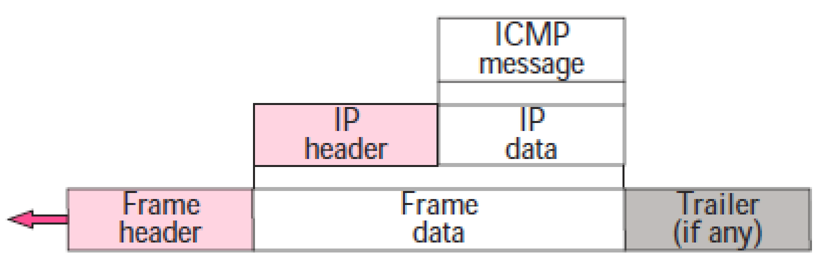
The IP protocol also **lacks a mechanism for host and management queries.**

- **A host sometimes needs** to determine if a router or another host is alive.
- **A network manager sometimes** needs information from another host or router.

The Internet Control Message Protocol (ICMP) **has been designed to compensate for the above two deficiencies.** **It is a companion to the IP protocol.**



ICMP itself is a network layer protocol. However, its messages are not passed directly to the data link layer as would be expected. **Instead,** the messages are first encapsulated inside IP datagrams before going to the lower layer.



ICMP Messages

ICMP messages are divided into two broad categories: **error-reporting messages** and **query messages.**

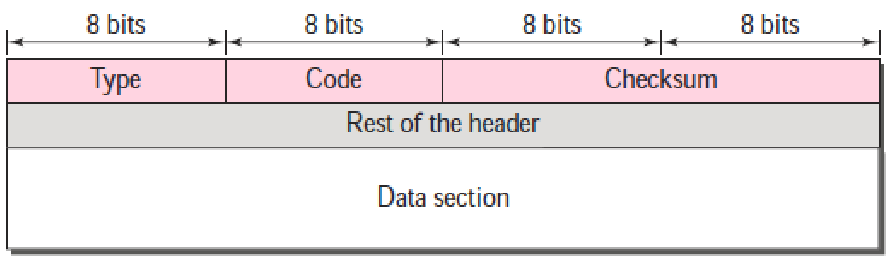
- **Error-reporting Messages:** The error-reporting messages **report problems** that a router or a host (destination) may encounter when it processes an IP packet.
- **Query Messages:** The query messages, which occur in pairs, **help a host or a network manager get specific information from a router or another host.** For example, nodes can discover their neighbors. Also, hosts can discover and learn about routers on their network and routers can help a node redirect its messages.

Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

Message Format

An ICMP message **has an 8-byte header** and a **variable-size data section**. Although the general format of the header is different for each message type, **the first 4 bytes are common to all:**

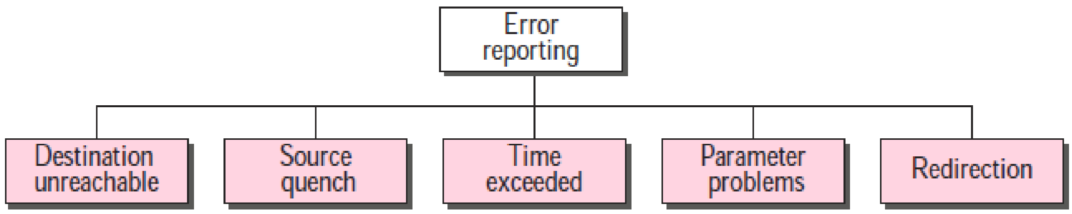
- **The ICMP type field** defines the type of the message.
- **The code field** specifies the reason for the particular message type.
- **The last common field is the checksum field** (the error detection method).
- **The rest of the header** is specific for each message type.



1. Error Reporting Messages

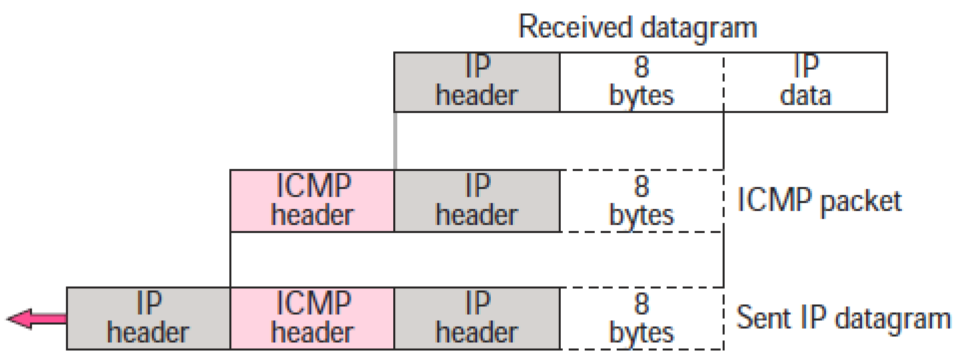
- **ICMP does not correct errors, it simply reports them.**
- **Error messages are always sent to the original source because the only information available in the datagram about the route is the source and destination IP addresses.**

Five types of errors are handled:



All error messages contain **a data section** that ***includes the IP header of the original datagram plus the first 8 bytes of data in that datagram.***

- **The original datagram header is added** to give the original source, which receives the error message, information about the datagram itself.
- **The 8 bytes of data are included** because the **first 8 bytes** provide information about the port numbers (UDP and TCP) and sequence number (TCP). This information is needed so the source can inform the protocols (TCP or UDP) about the error.
- ICMP forms an error packet, **which is then encapsulated in an IP datagram:**



A. Destination Unreachable

When a **router cannot route a datagram** or a **host cannot deliver a datagram**, the datagram is discarded and the router or the host **sends** a **destination-unreachable message back** to the source host that initiated the datagram.

Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The code field for this type specifies the reason for discarding the datagram:

Code	*Reason for discarding the datagram*
Code 0	**The network is unreachable, possibly due to hardware failure.**
Code 1	**The host is unreachable. This can also be due to hardware failure.**
Code 2	**The protocol is unreachable (such as UDP, TCP, and OSPF).**
Code 3	**The port is unreachable.**
Code 4	**Fragmentation is required, but the DF (do not fragment) field of the datagram has been set.**
Code 5	**Source routing cannot be accomplished.**
Code 6	**The destination network is unknown.**
Code 7	**The destination host is unknown.**
Code 8	**The source host is isolated.**
Code 9	**Communication with the destination network is administratively prohibited.**
Code 10	**Communication with the destination host is administratively prohibited.**
Code 11	**The network is unreachable for the specified type of service.**
Code 12	**The host is unreachable for the specified type of service.**
Code 13	**The host is unreachable because the administrator has put a filter on it.**
Code 14	**The host is unreachable because the host precedence is violated.**
Code 15	**The host is unreachable because its precedence was cut off.**

Note:

- Destination-unreachable messages with **codes 2 or 3** can be created only by the **destination host. Other destination-unreachable messages** can be **created only by routers**.
- **A router cannot detect all problems that prevent the delivery of a packet.**

B. Source Quench

Type: 4	Code: 0	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

- There is ***no flow-control*** or ***congestion-control mechanism*** in the **IP protocol**.
- **The source-quench message in ICMP was designed to add a kind of flow control and congestion control to the IP.** When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram.

- This message has two purposes.
  - **First**: **it informs the source** that the datagram **has been discarded**.
  - **Second: it warns the source** that there is **congestion somewhere** in the path and that **the source should slow down *(quench)* the sending process**.
- **The router or destination host** that has experienced the congestion sends **one source-quench message** for each discarded datagram to the source host.

C. Time Exceeded

Type: 11	Code: 0 or 1	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The time-exceeded message is generated in two cases:

- **First: If there are errors in one or more routing tables,** a packet **can travel in a loop or a cycle,** going from one router to the next or visiting a series of routers endlessly. Each datagram contains **a field called time to live that controls this situation.** When a datagram visits a router, the value of this field is **decremented by 1**. When the **time-to-live value** reaches **0,** after decrementing, the router discards the datagram. However, when the datagram is discarded, a time-exceeded message must be sent by the router to the original source.
- **Second: When the final destination does not receive all of the fragments in a set time,** it discards the received fragments and sends a time-exceeded message to the original source.

Note:

- **Code 0** is used when the datagram is discarded by the router **due to a time-to-live field value of zero.**
- **Code 1** is used when arrived fragments of a datagram are discarded **because some fragments have not arrived within the time limit**.

D. Parameter Problem

Type: 12	Code: 0 or 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Any ambiguity in the header part of a datagram can **create serious problems** as the datagram travels through the Internet. If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.

- **Code 0:** There is an error or ambiguity **in one of the header fields**.
- **Code 1:** The required part of **an option is missing**.

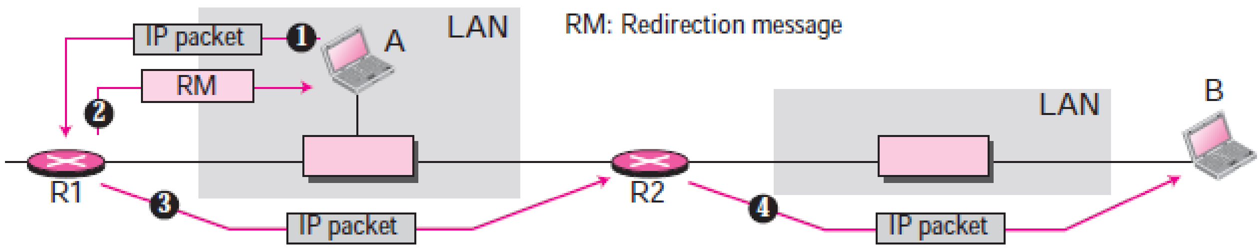
- ***A parameter-problem message can be created by a router or the destination host.***

E. Redirection

Type: 5	Code: 0 to 3	Checksum
IP address of the target router		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

When a host comes up, its routing table has a limited number of entries. **It usually knows only the IP address of one router**, ***the default router***. *For this reason,* the host may ***send a datagram***, which is **destined for another network**, ***to the wrong router***. In this case, the **router that receives the datagram will**

forward the datagram to the correct router. *However*, **to update the routing table of the host, it sends a redirection message to the host.**



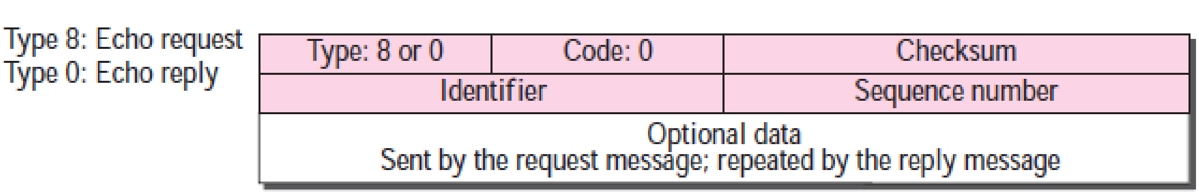
- ***Redirection message is different from other error messages.*** The router **does not discard** the datagram in this case; **it is sent to the appropriate router.**
- **A redirection message is sent from a router to a host on the same local network.**

2. Query Messages

- Addition to error reporting, ICMP ***can also diagnose some network problems.*** This is accomplished through the query messages.
  1. ***Echo request and replay.***
  2. ***Timestamp request and replay***.
- **In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node.**

A. Echo Request and Reply

- The **echo-request** and **echo-reply messages** ***are designed for diagnostic purposes***. Network managers and users utilize this pair of messages **to identify network problems.**
- A host or router can send an echo-request message. The host or router that receives an echo-request message creates an echo-reply message and returns it to the original sender.
- Echo-request and echo-reply messages can be used by network managers **to check the operation of the IP protocol.**
- Echo-request and echo-reply messages **can test the reachability of a host**. This is usually done by invoking the **packet Internet groper (ping) command**.



- **The optional data field** contains a message that **must be repeated exactly** by the responding node in its echo-reply message.
- **The identifier and sequence number fields are not formally defined by the protocol** and can be used arbitrarily by the sender. The identifier is often the same as the process ID.

B. Timestamp Request and Reply

- Two machines (hosts or routers) can use the **timestamp-request** and **timestamp-reply messages to determine the round-trip time needed for an IP datagram to travel between them.**
- **It can also be used to synchronize the clocks in two machines.**

Type 13: request
Type 14: reply

Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

The three timestamp fields are each **32 bits** long. Each field can hold a number representing time measured in **milliseconds from midnight in Universal Time** (formerly called **Greenwich Mean Time**).

(Note that **32 bits** can represent a number between **0** and **4,294,967,295**, but a timestamp in this case cannot exceed **86,400,000 = 24 ∗ 60 ∗ 60 ∗ 1000**).

The source creates a timestamp-request message:

- The source fills the ***original timestamp* field** with the Universal Time shown by its clock at departure time.
- The **other two timestamp fields are filled with zeros**.

The destination creates the timestamp-reply message:

- The destination **copies the original timestamp value** from the request message into the **same field** in its reply message.
- It then fills the ***receive timestamp* field** with the Universal Time shown by its clock at the time the request was received.
- Finally, it fills the ***transmit timestamp* field** with the Universal Time shown by its clock at the time the reply message departs.

The timestamp-request and timestamp-reply messages can be used to compute the one-way or round-trip time required for a datagram to go from a source to a destination and then back again. The formulas are:

Sending time = Receive timestamp − Original timestamp
Receiving time = Returned time − Transmit timestamp
Round-trip time = Sending time + Receiving time

Note that the sending and receiving time calculations are accurate only if the two clocks in the source and destination machines are synchronized. **However**, **the round-trip calculation is correct even if the two clocks are not synchronized because each clock contributes twice to the round-trip calculation,** thus canceling any difference in synchronization.

For example, given the following information:

Original timestamp: **46** Receive timestamp: **59**
Transmit timestamp: **60** Return time: **67**

We can calculate the round-trip time to be 20 milliseconds:

Sending time = **59** − **46** = **13** milliseconds

Receiving time = **67** − **60** = **7** milliseconds

Round-trip time = **13 + 7** = **20** milliseconds

Given the actual one-way time, the timestamp-request and timestamp-reply messages can also be used to synchronize the clocks in two machines using the following formula:

Time difference = Receive timestamp − (Original timestamp field + One-way time duration)

The one-way time duration can be obtained either by dividing the round-trip time duration by two (if we are sure that the sending time is the same as the receiving time) or by other means. **For example**, **we can tell that the two clocks in the previous example** are **3 milliseconds** out of synchronization because:

Time difference = 59 − (46 + 10) = 3