DHCP

The best way of approaching this fairly simple but time consuming task is to use a computer

DHCP

The best way of approaching this fairly simple but time consuming task is to use a computer

The *Dynamic Host Configuration Protocol* (DHCP) does just this

DHCP

The best way of approaching this fairly simple but time consuming task is to use a computer

The *Dynamic Host Configuration Protocol* (DHCP) does just this

When a machine needs an IP address it can use DHCP to get one

DHCP

When a host boots and finds it needs an IP address, it makes a DHCP *broadcast*

DHCP

When a host boots and finds it needs an IP address, it makes a DHCP *broadcast*

This is like the host saying “can anyone give me an IP address?” to the network

DHCP

When a host boots and finds it needs an IP address, it makes a DHCP *broadcast*

This is like the host saying “can anyone give me an IP address?” to the network

In contrast to ARP, with DHCP there is usually just one (occasionally more as backup) host that is configured to respond to DHCP requests, as allocation of addresses must be centrally managed to avoid duplication

DHCP

When a host boots and finds it needs an IP address, it makes a DHCP *broadcast*

This is like the host saying “can anyone give me an IP address?” to the network

In contrast to ARP, with DHCP there is usually just one (occasionally more as backup) host that is configured to respond to DHCP requests, as allocation of addresses must be centrally managed to avoid duplication

Again in contrast with ARP, this request is a network layer local broadcast, actually using an IP packet with address `255.255.255.255`

DHCP

A DHCP server (i.e., the DHCP program running on some host) listens for such requests; it will choose a currently unused IP address and send it back to the requesting client

DHCP

A DHCP server (i.e., the DHCP program running on some host) listens for such requests; it will choose a currently unused IP address and send it back to the requesting client

The value might be chosen by the server according to some defined policy, or (more usually) the next free address taken from a list of currently unused addresses

DHCP

A DHCP server (i.e., the DHCP program running on some host) listens for such requests; it will choose a currently unused IP address and send it back to the requesting client

The value might be chosen by the server according to some defined policy, or (more usually) the next free address taken from a list of currently unused addresses

The client gets this reply and reads its IP address which it can then use to configure itself

DHCP

In outline:

DHCP

In outline:

1. the client broadcasts “Who out there is willing to do DHCP with me?” (a `DHCPDISCOVER` message)

DHCP

In outline:

1. the client broadcasts “Who out there is willing to do DHCP with me?” (a `DHCPDISCOVER` message)
2. one or more servers broadcast a reply. “I will. Here’s an address” (`DHCPOFFER`)

DHCP

In outline:

1. the client broadcasts “Who out there is willing to do DHCP with me?” (a `DHCPDISCOVER` message)
2. one or more servers broadcast a reply. “I will. Here’s an address” (`DHCPOFFER`)
3. the client picks a server and broadcasts “Can I have that address, please?” (`DHCPREQUEST`)

DHCP

In outline:

1. the client broadcasts “Who out there is willing to do DHCP with me?” (a `DHCPDISCOVER` message)
2. one or more servers broadcast a reply. “I will. Here’s an address” (`DHCPOFFER`)
3. the client picks a server and broadcasts “Can I have that address, please?” (`DHCPREQUEST`)
4. the chosen server broadcasts “OK, it’s yours” (`DHCPACK`)

DHCP

In outline:

1. the client broadcasts “Who out there is willing to do DHCP with me?” (a `DHCPDISCOVER` message)
2. one or more servers broadcast a reply. “I will. Here’s an address” (`DHCPOFFER`)
3. the client picks a server and broadcasts “Can I have that address, please?” (`DHCPREQUEST`)
4. the chosen server broadcasts “OK, it’s yours” (`DHCPACK`)
5. the client sets its IP address

DHCP

Exercise Find out the details, e.g., what happens if a packet gets lost? For example, the `DHCPACK`

DHCP

DHCP runs over (UDP over) IP, so DHCP packets must have IP source and destination addresses

DHCP

DHCP runs over (UDP over) IP, so DHCP packets must have IP source and destination addresses

But the client doesn't yet know its own IP address, or any server's address, but it must fill in the IP address fields with something

DHCP

DHCP runs over (UDP over) IP, so DHCP packets must have IP source and destination addresses

But the client doesn't yet know its own IP address, or any server's address, but it must fill in the IP address fields with something

Source: `0.0.0.0`. This is what we are trying to find

DHCP

DHCP runs over (UDP over) IP, so DHCP packets must have IP source and destination addresses

But the client doesn't yet know its own IP address, or any server's address, but it must fill in the IP address fields with something

Source: `0.0.0.0`. This is what we are trying to find

Destination: `255.255.255.255`. A local network broadcast

DHCP

DHCP runs over (UDP over) IP, so DHCP packets must have IP source and destination addresses

But the client doesn't yet know its own IP address, or any server's address, but it must fill in the IP address fields with something

Source: `0.0.0.0`. This is what we are trying to find

Destination: `255.255.255.255`. A local network broadcast

Exercise So what would the link layer address be?

DHCP

Packets returning from the DHCP server will have the server's IP address as source, and the broadcast `255.255.255.255` as destination

DHCP

Packets returning from the DHCP server will have the server's IP address as source, and the broadcast `255.255.255.255` as destination

Again, the client doesn't yet have an IP address, so we have to resort to a broadcast to everybody

DHCP

Packets returning from the DHCP server will have the server's IP address as source, and the broadcast `255.255.255.255` as destination

Again, the client doesn't yet have an IP address, so we have to resort to a broadcast to everybody

This is extra work for all hosts on the network (reading then ignoring the DHCP reply packets), but DHCP exchanges are relatively rare so it's not so bothersome

DHCP

Packets returning from the DHCP server will have the server's IP address as source, and the broadcast `255.255.255.255` as destination

Again, the client doesn't yet have an IP address, so we have to resort to a broadcast to everybody

This is extra work for all hosts on the network (reading then ignoring the DHCP reply packets), but DHCP exchanges are relatively rare so it's not so bothersome

There are security implications though...

DHCP

Packets returning from the DHCP server will have the server's IP address as source, and the broadcast `255.255.255.255` as destination

Again, the client doesn't yet have an IP address, so we have to resort to a broadcast to everybody

This is extra work for all hosts on the network (reading then ignoring the DHCP reply packets), but DHCP exchanges are relatively rare so it's not so bothersome

There are security implications though...

There is an identification field in the `DHCPOFFER` that allows a host to recognise a reply is for itself and not mistakenly take an offer for some other host that is doing DHCP at the same time

DHCP

DHCP

A DHCP server has a pool of available addresses that it can assign to hosts as they need them

DHCP

DHCP

A DHCP server has a pool of available addresses that it can assign to hosts as they need them

When a host leaves the network, it should send a `DHCPRELEASE` to the DHCP server

DHCP

DHCP

A DHCP server has a pool of available addresses that it can assign to hosts as they need them

When a host leaves the network, it should send a `DHCPRELEASE` to the DHCP server

Thus releasing its IP address to be reused for another host

DHCP

DHCP

A DHCP server has a pool of available addresses that it can assign to hosts as they need them

When a host leaves the network, it should send a `DHCPRELEASE` to the DHCP server

Thus releasing its IP address to be reused for another host

But not all clients are well behaved, or might have crashed before sending a release

DHCP

DHCP

To fix this, DHCP gives a *lease time* on an address

DHCP

DHCP

To fix this, DHCP gives a *lease time* on an address

The address is usable by the requesting host for this period of time

DHCP

DHCP

To fix this, DHCP gives a *lease time* on an address

The address is usable by the requesting host for this period of time

If the lease expires the host can request a *renewal* of the lease from the server

DHCP

DHCP

To fix this, DHCP gives a *lease time* on an address

The address is usable by the requesting host for this period of time

If the lease expires the host can request a *renewal* of the lease from the server

Which will then grant a further lease on the address

DHCP

DHCP

To fix this, DHCP gives a *lease time* on an address

The address is usable by the requesting host for this period of time

If the lease expires the host can request a *renewal* of the lease from the server

Which will then grant a further lease on the address

The renewal request and reply can be a normal unicast (non-broadcast) interchange, as the client already has an IP address

DHCP

If a host leaves the network or crashes, a renewal request will not be forthcoming

DHCP

If a host leaves the network or crashes, a renewal request will not be forthcoming

Thus the server can know, when the lease has expired, that the allocated IP address is no longer needed, and can be put back into the pool

DHCP

If a host leaves the network or crashes, a renewal request will not be forthcoming

Thus the server can know, when the lease has expired, that the allocated IP address is no longer needed, and can be put back into the pool

(There are many protocols like this, that need a timeout to catch something bad happening)

DHCP

How long is the lease time?

DHCP

How long is the lease time?

This is configurable by the DHCP server's administrator

DHCP

How long is the lease time?

This is configurable by the DHCP server's administrator

A short period is used when there is a fast turnover of machines (e.g., laptops in the library)

DHCP

How long is the lease time?

This is configurable by the DHCP server's administrator

A short period is used when there is a fast turnover of machines (e.g., laptops in the library)

A long period, up to infinity, is used for more permanent machines, e.g., desktops

DHCP

How long is the lease time?

This is configurable by the DHCP server's administrator

A short period is used when there is a fast turnover of machines (e.g., laptops in the library)

A long period, up to infinity, is used for more permanent machines, e.g., desktops

The administrator of the DHCP server needs to pick suitable values

DHCP

How long is the lease time?

This is configurable by the DHCP server's administrator

A short period is used when there is a fast turnover of machines (e.g., laptops in the library)

A long period, up to infinity, is used for more permanent machines, e.g., desktops

The administrator of the DHCP server needs to pick suitable values

Exercise What is the lease time from your access point on your home network?

DHCP

Besides addresses, DHCP can supply

- IP address
- netmask
- gateway
- name servers
- lease times
- print servers
- boot servers
- mail servers
- host name
- web servers
- and so on

DHCP

But usually just

- IP address
- netmask
- gateway
- name servers (for DNS, see later)

which is the minimum needed to get a host up and running and talking to the wider Internet

DHCP

But usually just

- IP address
- netmask
- gateway
- name servers (for DNS, see later)

which is the minimum needed to get a host up and running and talking to the wider Internet

And the lease time

DHCP

After getting a new address, a client might broadcast an ARP reply containing its new address

DHCP

After getting a new address, a client might broadcast an ARP reply containing its new address

This unrequested *gratuitous ARP* informs other hosts on the network of the new address association so they can update their ARP caches, e.g., invalidating an old association with this IP address

DHCP

Thus, DHCP solves the address, gateway and netmask (and other) configuration problem

DHCP

Thus, DHCP solves the address, gateway and netmask (and other) configuration problem

But there is a wider issue we've alluded to several times that we must now discuss

Internet/Network Layer

Many times, in many circumstances, when things go wrong, we have said things like “blah blah and send an error message back”

Internet/Network Layer

Many times, in many circumstances, when things go wrong, we have said things like “blah blah and send an error message back”

For example, in MTU discovery we had “drop the packet and send an error message back”

Internet/Network Layer

Many times, in many circumstances, when things go wrong, we have said things like “blah blah and send an error message back”

For example, in MTU discovery we had “drop the packet and send an error message back”

Or when a TTL drops to zero, we had “drop the packet and send an error message back”

Internet/Network Layer

So how is such a message sent?

Internet/Network Layer

So how is such a message sent?

We only have packets, so the message must be in a packet

Internet/Network Layer

So how is such a message sent?

We only have packets, so the message must be in a packet

Just another IP datagram, with particular contents

Internet/Network Layer

So how is such a message sent?

We only have packets, so the message must be in a packet

Just another IP datagram, with particular contents

An *Internet Control Message Protocol* (ICMP) packet

ICMP

ICMP is used for general control of the Internet, in particular errors

ICMP

ICMP is used for general control of the Internet, in particular errors

ICMP packets are contained within IP packets, but are considered to be part of the network layer

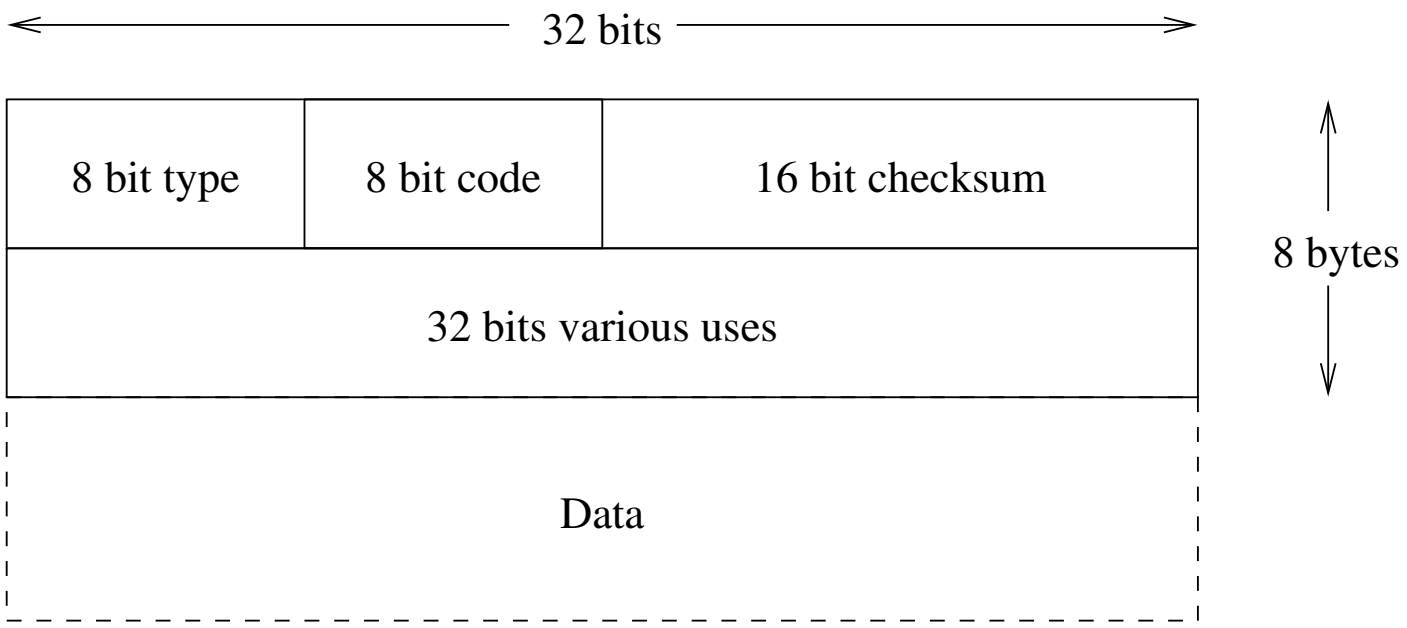
ICMP

ICMP is used for general control of the Internet, in particular errors

ICMP packets are contained within IP packets, but are considered to be part of the network layer

Thus the data field in an IP datagram might contain transport layer stuff, or it might contain network layer stuff

ICMP



ICMP Packet

ICMP

- Type: kind of message, e.g., “TTL expired”, “destination unreachable”, “fragmentation needed but DF set”

ICMP

- Type: kind of message, e.g., “TTL expired”, “destination unreachable”, “fragmentation needed but DF set”
- Code: additional information, e.g., “destination unreachable” has “network unreachable” and “host unreachable” codes

ICMP

- Type: kind of message, e.g., “TTL expired”, “destination unreachable”, “fragmentation needed but DF set”
- Code: additional information, e.g., “destination unreachable” has “network unreachable” and “host unreachable” codes
- Checksum

ICMP

- Type: kind of message, e.g., “TTL expired”, “destination unreachable”, “fragmentation needed but DF set”
- Code: additional information, e.g., “destination unreachable” has “network unreachable” and “host unreachable” codes
- Checksum
- A fixed size field that has varying purposes for different types

ICMP

- Type: kind of message, e.g., “TTL expired”, “destination unreachable”, “fragmentation needed but DF set”
- Code: additional information, e.g., “destination unreachable” has “network unreachable” and “host unreachable” codes
- Checksum
- A fixed size field that has varying purposes for different types
- A general data field, if needed

ICMP

Thus ICMP packets of various types are used to indicate the different kinds of error message

ICMP

Thus ICMP packets of various types are used to indicate the different kinds of error message

For example, when a TTL on a packet decrements to zero, the router drops the packet, creates an ICMP “TTL expired” packet and sends it back to the source address, as given in the dropped packet

ICMP

Thus ICMP packets of various types are used to indicate the different kinds of error message

For example, when a TTL on a packet decrements to zero, the router drops the packet, creates an ICMP “TTL expired” packet and sends it back to the source address, as given in the dropped packet

This message (in an IP packet) will have IP source address of the router; and destination address the source of the problem packet

ICMP

But, remember, ICMP packets are IP packets and so can be lost, delayed, duplicated or otherwise corrupted

ICMP

But, remember, ICMP packets are IP packets and so can be lost, delayed, duplicated or otherwise corrupted

And so ICMP errors can be generated for ICMP packets, with certain reservations

ICMP

But, remember, ICMP packets are IP packets and so can be lost, delayed, duplicated or otherwise corrupted

And so ICMP errors can be generated for ICMP packets, with certain reservations

ICMP messages are classed as either a *query* or an *error*

ICMP

But, remember, ICMP packets are IP packets and so can be lost, delayed, duplicated or otherwise corrupted

And so ICMP errors can be generated for ICMP packets, with certain reservations

ICMP messages are classed as either a *query* or an *error*

E.g., ICMP “echo request” (ping) is a query, but “TTL expired” is an error

ICMP

ICMP errors are not generated for

- ICMP errors (e.g., TTL expires on a ICMP packet)
- a packet whose destination is a broadcast or multicast
- a packet whose source is a broadcast or multicast
- a packet whose link-layer address is a broadcast
- any fragment other than the first

ICMP

ICMP errors are not generated for

- ICMP errors (e.g., TTL expires on a ICMP packet)
- a packet whose destination is a broadcast or multicast
- a packet whose source is a broadcast or multicast
- a packet whose link-layer address is a broadcast
- any fragment other than the first

This is to prevent broadcast storms, where a single error is multiplied up into many ICMP packets

ICMP

ICMP errors are not generated for

- ICMP errors (e.g., TTL expires on a ICMP packet)
- a packet whose destination is a broadcast or multicast
- a packet whose source is a broadcast or multicast
- a packet whose link-layer address is a broadcast
- any fragment other than the first

This is to prevent broadcast storms, where a single error is multiplied up into many ICMP packets

Non-initial IP fragments don't contain enough identifying information for the OS to do anything useful with them, so don't bother with them (**Exercise** How do you know if you have an initial fragment?)

ICMP

Type	Err	Code
ECHOREPLY		reply from a ping
DEST_UNREACH	e	network unreachable
	e	host unreachable
	e	port unreachable
	e	fragmentation wanted but DF set
REDIRECT	e	routing redirect for network
	e	routing redirect for host
ECHO		ping
TIME_EXCEEDED	e	TTL reached 0
	e	fragment reassembly time exceeded

Messages marked “e” are errors. There are many other types and codes, but the above are the most common in practice.

ICMP

Ping

ICMP has many other uses

ICMP

Ping

ICMP has many other uses

For example, we can discover if a machine is up and running using ICMP *ping*

ICMP

Ping

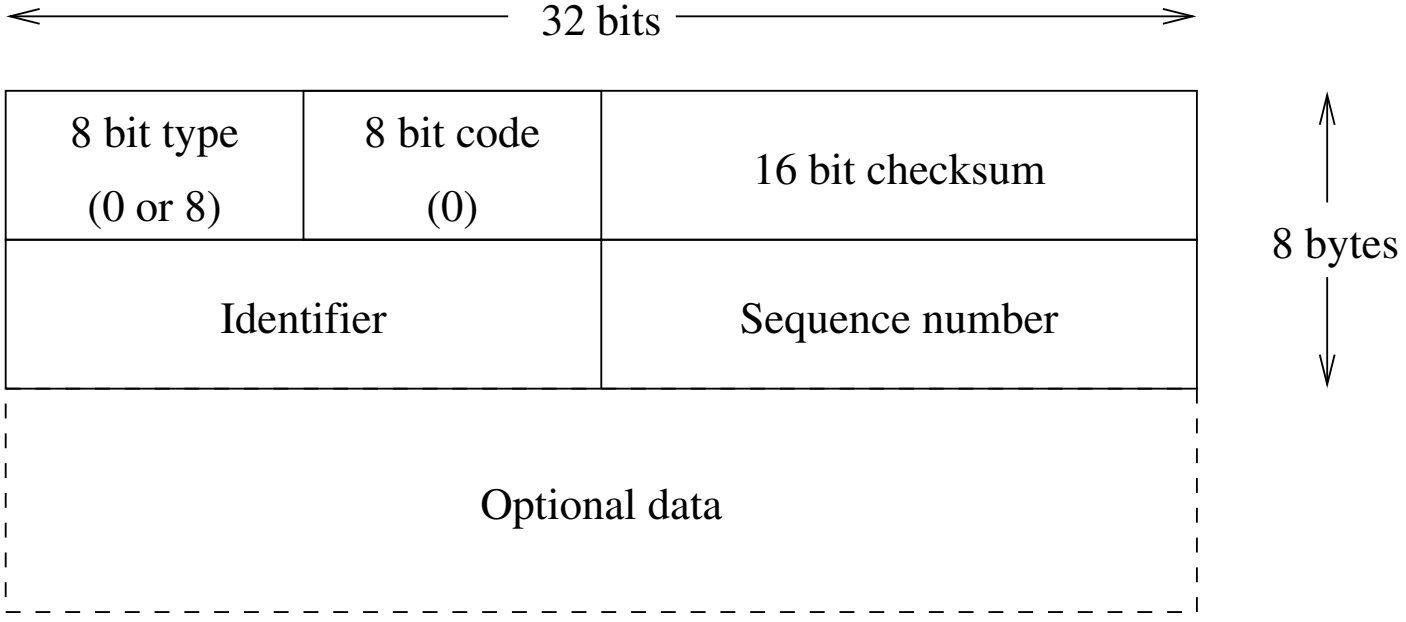
ICMP has many other uses

For example, we can discover if a machine is up and running using ICMP *ping*

A program, usually called `ping`, sends an ICMP “echo request” (also usually called a “ping”) packet, waits a second, then repeats

ICMP

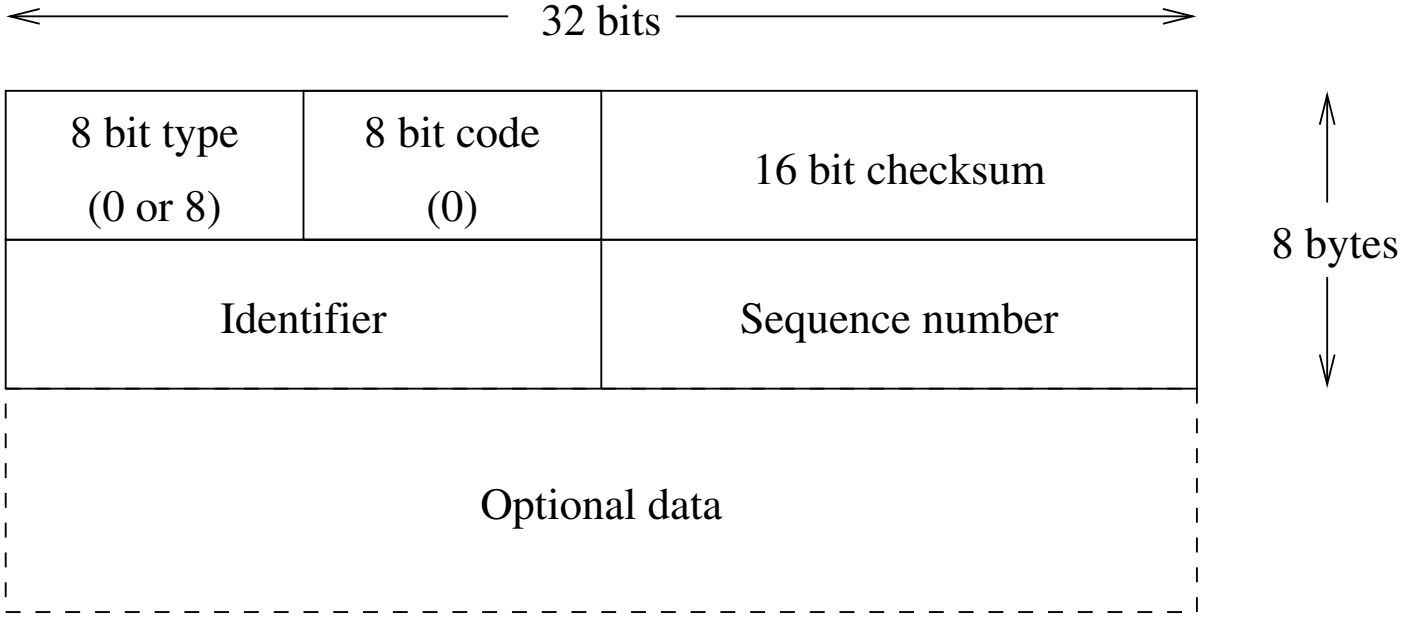
Ping



ICMP ping headers

ICMP

Ping

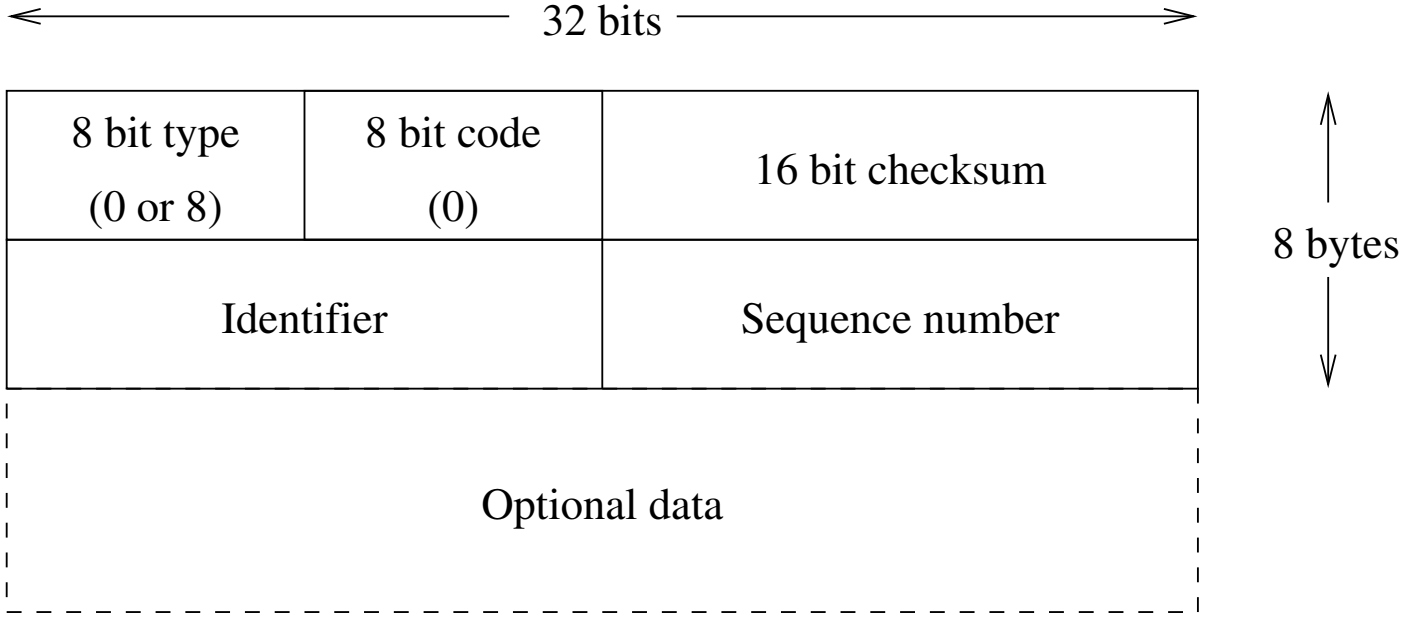


ICMP ping headers

- ICMP type 0, code 0, with some random identifier

ICMP

Ping

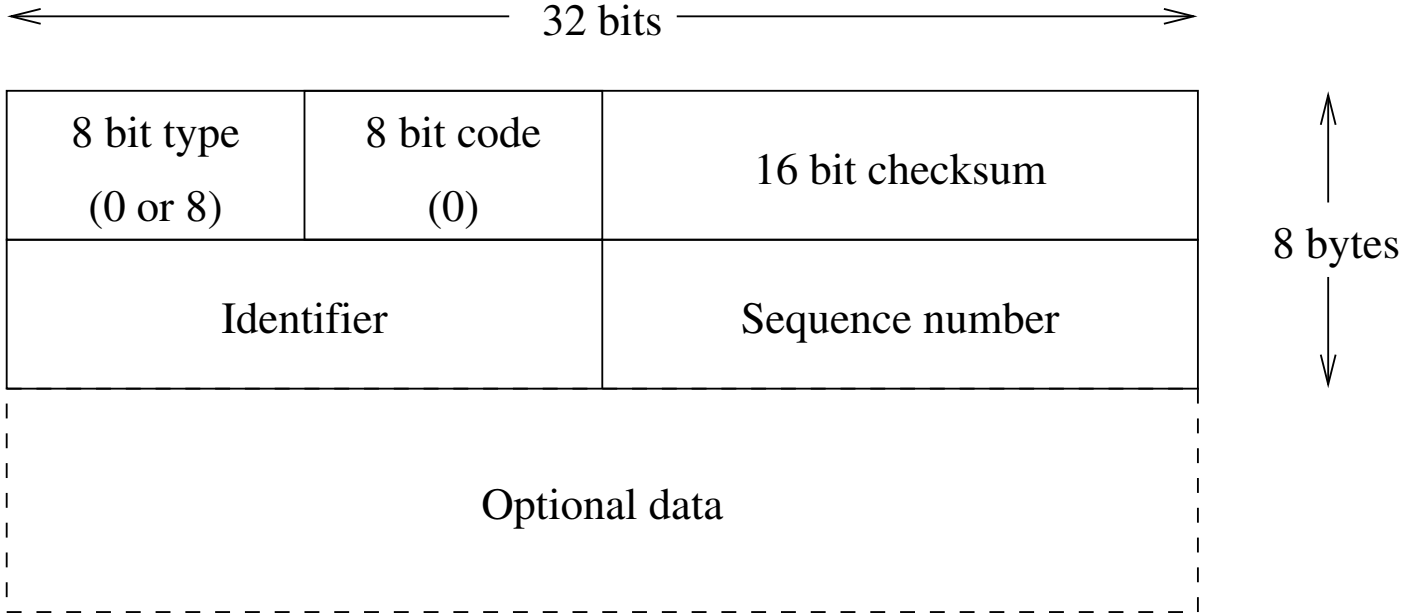


ICMP ping headers

- ICMP type 0, code 0, with some random identifier
- A functioning host OS that gets a ping should return a "echo reply"

ICMP

Ping



ICMP ping headers

- ICMP type 0, code 0, with some random identifier
- A functioning host OS that gets a ping should return a “echo reply”
- This has ICMP type 8, code 0, and a copy of the identifier, sequence and data

ICMP

Ping

- The identifier field allows the originator OS to match up replies with requests

ICMP

Ping

- The identifier field allows the originator OS to match up replies with requests
- The sequence starts at 0 and increases by 1 for each ping sent

ICMP

Ping

- The identifier field allows the originator OS to match up replies with requests
- The sequence starts at 0 and increases by 1 for each ping sent

This allows us to spot lost, duplicated or reordered packets

ICMP

Ping

```
% ping www.yahoo.co.uk
PING homerc.europe.yahoo.com: 56 data bytes
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=0. time=160. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=1. time=154. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=2. time=176. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=3. time=159. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=4. time=161. ms
^C
----homerc.europe.yahoo.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 154/162/176
```

ICMP

Ping

```
% ping www.yahoo.co.uk
PING homerc.europe.yahoo.com: 56 data bytes
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=0. time=160. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=1. time=154. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=2. time=176. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=3. time=159. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=4. time=161. ms
^C
----homerc.europe.yahoo.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 154/162/176
```

The `ping` command also keeps track of *round trip time* (RTT), the time between sending a request and getting the corresponding reply

ICMP

Ping

```
% ping www.yahoo.co.uk
PING homerc.europe.yahoo.com: 56 data bytes
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=0. time=160. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=1. time=154. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=2. time=176. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=3. time=159. ms
64 bytes from rc3.europe.yahoo.com (194.237.109.72): icmp_seq=4. time=161. ms
^C
----homerc.europe.yahoo.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 154/162/176
```

The `ping` command also keeps track of *round trip time* (RTT), the time between sending a request and getting the corresponding reply

Note lots of variance in the RTT: this is typical

ICMP

Ping

Some versions of ping can enable the IP header option record route: this makes IP save the address of each intermediate router in the header

ICMP

Ping

Some versions of ping can enable the IP header option record route: this makes IP save the address of each intermediate router in the header

But, as noted earlier, there can only be 60 bytes of options in IPv4, giving space for up to 9 addresses (with the overheads of the option header and other bits and pieces), so only 9 addresses are recorded

ICMP

Ping

```
% ping -R www.bbc.co.uk
PING www.bbc.net.uk (212.58.244.70) 56(124) bytes of data.
64 bytes from bbc-vip115.telhc.bbc.co.uk (212.58.244.70): icmp_seq=1 ttl=52
 time=89.0 ms
RR:     rjb.cs.bath.ac.uk (172.16.2.1)
        fire.cs.bath.ac.uk (138.38.108.253)
        swan-fwsm.bath.ac.uk (138.38.1.46)
        university-of-bath.ja.net (146.97.144.38)
        xe-0-0-0.bathbc-rbr1.ja.net (146.97.67.46)
        xe-1-0-0.brisub-rbr1.ja.net (146.97.67.33)
        swr.londpg-sbr1.ja.net (146.97.37.202)
        ae29.londpg-sbr1.ja.net (146.97.33.2)
        ae0.londhx-sbr1.ja.net (146.97.35.105)

64 bytes from bbc-vip115.telhc.bbc.co.uk (212.58.244.70): icmp_seq=2 ttl=52
 time=25.7 ms        (same route)
^C
--- www.bbc.net.uk ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 25.734/57.370/89.006/31.636 ms
```