IP

The optional header fields allow for items that are either

- not common, so you don't want to pay the overhead of always having them, or
- protocol extensions that are useful but were not included in the original IP specification

IP

The optional header fields allow for items that are either

- not common, so you don't want to pay the overhead of always having them, or
- protocol extensions that are useful but were not included in the original IP specification

When IP was first devised, networks ran over telephone modems at a few thousand bits/sec. Now we expect giga and terabits. The physical layer has changed immensely while basic TCP/IP is pretty much unchanged!

IP

The optional header fields allow for items that are either

- not common, so you don't want to pay the overhead of always having them, or
- protocol extensions that are useful but were not included in the original IP specification

When IP was first devised, networks ran over telephone modems at a few thousand bits/sec. Now we expect giga and terabits. The physical layer has changed immensely while basic TCP/IP is pretty much unchanged!

A flexible approach across the entire TCP/IP suite such as allowing for options is part of this

IP

IP layer options are not much used. Options include

IP

IP layer options are not much used. Options include

- Security: privacy and authentication

IP

IP layer options are not much used. Options include

- Security: privacy and authentication
- Record Route. Each router records its address in the option header as the datagram passes by

IP

IP layer options are not much used. Options include

- Security: privacy and authentication
- Record Route. Each router records its address in the option header as the datagram passes by
- Timestamp. Each router records its address and the current time in the header as the datagram passes by

IP

IP layer options are not much used. Options include

- Security: privacy and authentication
- Record Route. Each router records its address in the option header as the datagram passes by
- Timestamp. Each router records its address and the current time in the header as the datagram passes by
- Strict Source Routing. A list of addresses that give the entire path from source to destination

IP

IP layer options are not much used. Options include

- Security: privacy and authentication
- Record Route. Each router records its address in the option header as the datagram passes by
- Timestamp. Each router records its address and the current time in the header as the datagram passes by
- Strict Source Routing. A list of addresses that give the entire path from source to destination
- Loose Source Routing. A list of addresses that must be included in the path from source to destination

IP

Most IP options are for debugging or profiling behaviour, and not much used

IP

Most IP options are for debugging or profiling behaviour, and not much used

These days, things like privacy and authentication are mostly done in other layers

IP

Most IP options are for debugging or profiling behaviour, and not much used

These days, things like privacy and authentication are mostly done in other layers

Moreover, the IP header length field has a maximum value of 60 bytes; the fixed part of the header is 20 bytes; thus options can use up to 40 bytes

IP

Most IP options are for debugging or profiling behaviour, and not much used

These days, things like privacy and authentication are mostly done in other layers

Moreover, the IP header length field has a maximum value of 60 bytes; the fixed part of the header is 20 bytes; thus options can use up to 40 bytes

This severely restricts what we can do in the options

IP Addresses

We now go back and look at the IP addresses in more detail

IP Addresses

We now go back and look at the IP addresses in more detail

Roughly (and incorrectly) speaking, every machine on the Internet has a unique address

IP Addresses

We now go back and look at the IP addresses in more detail

Roughly (and incorrectly) speaking, every machine on the Internet has a unique address

These are not random, but allocated in such a way to make routing between hosts much easier

IP Addresses

We now go back and look at the IP addresses in more detail

Roughly (and incorrectly) speaking, every machine on the Internet has a unique address

These are not random, but allocated in such a way to make routing between hosts much easier

If there were no structure on the addresses every router everywhere would have to know where every host in the world was

IP Addresses

We now go back and look at the IP addresses in more detail

Roughly (and incorrectly) speaking, every machine on the Internet has a unique address

These are not random, but allocated in such a way to make routing between hosts much easier

If there were no structure on the addresses every router everywhere would have to know where every host in the world was

Impossible, for technical, political and security reasons

IP Addresses

We now go back and look at the IP addresses in more detail

Roughly (and incorrectly) speaking, every machine on the Internet has a unique address

These are not random, but allocated in such a way to make routing between hosts much easier

If there were no structure on the addresses every router everywhere would have to know where every host in the world was

Impossible, for technical, political and security reasons

(Note that MAC addresses do not encode any network information, another reason for separate software addresses!)

IP Addresses

Recall the Internet is a collection of networks

IP Addresses

Recall the Internet is a collection of networks

An IP address is split into two parts:

- A network number
- A host number on that network

IP Addresses

Recall the Internet is a collection of networks

An IP address is split into two parts:

- A network number
- A host number on that network

The host number defines the host “uniquely” on a network

IP Addresses

Recall the Internet is a collection of networks

An IP address is split into two parts:

- A network number
- A host number on that network

The host number defines the host “uniquely” on a network

The network number defines a network “uniquely” on the Internet

IP Addresses

As we have already seen, to an end host routing is trivial

IP Addresses

As we have already seen, to an end host routing is trivial

- If the destination is on the same network, simply put the packet out on the network

IP Addresses

As we have already seen, to an end host routing is trivial

- If the destination is on the same network, simply put the packet out on the network
- If not, send the packet to a gateway, and let it deal with the problem

IP Addresses

As we have already seen, to an end host routing is trivial

- If the destination is on the same network, simply put the packet out on the network
- If not, send the packet to a gateway, and let it deal with the problem

It can tell if the destination is on the same network as itself by comparing the network part of their addresses

IP Addresses

As we have already seen, to an end host routing is trivial

- If the destination is on the same network, simply put the packet out on the network
- If not, send the packet to a gateway, and let it deal with the problem

It can tell if the destination is on the same network as itself by comparing the network part of their addresses

If they are the same, they are on the same network!

IP Addresses

To a gateway or router the problem is to send the packet on towards the destination network

IP Addresses

To a gateway or router the problem is to send the packet on towards the destination network

Which one? This is the difficult bit

IP Addresses

To a gateway or router the problem is to send the packet on towards the destination network

Which one? This is the difficult bit

But there are very many fewer networks than hosts, so this is already a great simplification

IP Addresses

To a gateway or router the problem is to send the packet on towards the destination network

Which one? This is the difficult bit

But there are very many fewer networks than hosts, so this is already a great simplification

Currently (2023): around 115000 top-level networks

IP Addresses

To a gateway or router the problem is to send the packet on towards the destination network

Which one? This is the difficult bit

But there are very many fewer networks than hosts, so this is already a great simplification

Currently (2023): around 115000 top-level networks

A router contains a table of IP addresses, with a next-hop neighbour router associated with each address

IP Addresses

To a gateway or router the problem is to send the packet on towards the destination network

Which one? This is the difficult bit

But there are very many fewer networks than hosts, so this is already a great simplification

Currently (2023): around 115000 top-level networks

A router contains a table of IP addresses, with a next-hop neighbour router associated with each address

(Actually a more complicated datastructure than a simple table, but we can think of it as a table)

IP Addresses

To a gateway or router the problem is to send the packet on towards the destination network

Which one? This is the difficult bit

But there are very many fewer networks than hosts, so this is already a great simplification

Currently (2023): around 115000 top-level networks

A router contains a table of IP addresses, with a next-hop neighbour router associated with each address

(Actually a more complicated datastructure than a simple table, but we can think of it as a table)

Containing network addresses, but individual host addresses are possible, too

IP Addresses

Each row in the table contains

IP Addresses

Each row in the table contains

- A destination address. This can be the address of a single host, but is usually a network address

IP Addresses

Each row in the table contains

- A destination address. This can be the address of a single host, but is usually a network address
- The address of the next hop router, i.e., the address of where to send the packet next. This is the address of a router that is directly connected to the current one

IP Addresses

Each row in the table contains

- A destination address. This can be the address of a single host, but is usually a network address
- The address of the next hop router, i.e., the address of where to send the packet next. This is the address of a router that is directly connected to the current one
- Which interface to send the packet out on to get to that router. A router has many interfaces and this describes which one to use

IP Addresses

When a packet arrives at a router it checks the table

IP Addresses

When a packet arrives at a router it checks the table

- If the packet destination matches a host address in the table, send the packet to the indicated host on the indicated interface

IP Addresses

When a packet arrives at a router it checks the table

- If the packet destination matches a host address in the table, send the packet to the indicated host on the indicated interface
- Else if the packet destination's network part matches a network address in the table, send the packet to the indicated host on the indicated interface

IP Addresses

When a packet arrives at a router it checks the table

- If the packet destination matches a host address in the table, send the packet to the indicated host on the indicated interface
- Else if the packet destination's network part matches a network address in the table, send the packet to the indicated host on the indicated interface
- Else if there is an entry in the table marked "default", send the packet to the indicated host on the indicated interface

IP Addresses

When a packet arrives at a router it checks the table

- If the packet destination matches a host address in the table, send the packet to the indicated host on the indicated interface
- Else if the packet destination's network part matches a network address in the table, send the packet to the indicated host on the indicated interface
- Else if there is an entry in the table marked "default", send the packet to the indicated host on the indicated interface
- Else error: drop the packet and return an error message "network unreachable" to the source

IP Addresses

For now, we can regard routers as machines with tables like this that tell them where to send packets. We will see how the tables are created later

IP Addresses

For now, we can regard routers as machines with tables like this that tell them where to send packets. We will see how the tables are created later

End hosts have routing tables, too: they are very simple, just encoding the local/non-local decision

IP Addresses

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
138.38.96.0	*	255.255.248.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	138.38.96.254	0.0.0.0	UG	0	0	0	eth0

A simple routing table as might be found in an end host on network `138.38.96`

IP Addresses

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
138.38.96.0	*	255.255.248.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	138.38.96.254	0.0.0.0	UG	0	0	0	eth0

A simple routing table as might be found in an end host on network `138.38.96`

- Send local traffic directly to the destination out on interface `eth0`

IP Addresses

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
138.38.96.0	*	255.255.248.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	138.38.96.254	0.0.0.0	UG	0	0	0	eth0

A simple routing table as might be found in an end host on network `138.38.96`

- Send local traffic directly to the destination out on interface `eth0`
- Otherwise send to the default gateway `138.38.96.254`, also on interface `eth0`

IP Addresses

The *netmask* (`Genmask` in the example) tells us how to divide an IP address into network and host parts. More details later, but a 1 bit set in the mask indicates this bit is part of the network address

IP Addresses

The *netmask* (`Genmask` in the example) tells us how to divide an IP address into network and host parts. More details later, but a 1 bit set in the mask indicates this bit is part of the network address

Work down the table ANDing the destination address on a packet with each netmask in turn. If the result equals the `Destination` value, we use this row to route the packet

IP Addresses

The *netmask* (`Genmask` in the example) tells us how to divide an IP address into network and host parts. More details later, but a 1 bit set in the mask indicates this bit is part of the network address

Work down the table ANDing the destination address on a packet with each netmask in turn. If the result equals the `Destination` value, we use this row to route the packet

A mask of 255.255.248.0 is 1111111111111111111110000000, so for this example the network part is the top 21 bits of the IP address

IP Addresses

The *netmask* (`Genmask` in the example) tells us how to divide an IP address into network and host parts. More details later, but a 1 bit set in the mask indicates this bit is part of the network address

Work down the table ANDing the destination address on a packet with each netmask in turn. If the result equals the `Destination` value, we use this row to route the packet

A mask of 255.255.248.0 is 1111111111111111111110000000, so for this example the network part is the top 21 bits of the IP address

"`default`" is actually destination 0.0.0.0, and so always matches any address after an AND with mask 0.0.0.0

IP Addresses

There is also a *loopback* address `127.0.0.1` for a virtual internal network connecting the machine to itself on (virtual) interface `lo0`

IP Addresses

There is also a *loopback* address `127.0.0.1` for a virtual internal network connecting the machine to itself on (virtual) interface `lo0`

This is useful for many things, such as testing

IP Addresses

A table from a machine with more than one real interface:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
213.121.147.69	*	255.255.255.255	UH	0	0	0	ppp0
172.18.0.0	*	255.255.0.0	U	0	0	0	eth0
172.17.0.0	*	255.255.0.0	U	0	0	0	eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	213.121.147.69	0.0.0.0	UG	0	0	0	ppp0

There are three interfaces: `eth0`, `eth1` and `ppp0` (as well as `lo`)

IP Addresses

A table from a machine with more than one real interface:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
213.121.147.69	*	255.255.255.255	UH	0	0	0	ppp0
172.18.0.0	*	255.255.0.0	U	0	0	0	eth0
172.17.0.0	*	255.255.0.0	U	0	0	0	eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	213.121.147.69	0.0.0.0	UG	0	0	0	ppp0

There are three interfaces: `eth0`, `eth1` and `ppp0` (as well as `lo`)

A packet with address `213.121.147.69` goes directly out on interface ppp0;

IP Addresses

A table from a machine with more than one real interface:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
213.121.147.69	*	255.255.255.255	UH	0	0	0	ppp0
172.18.0.0	*	255.255.0.0	U	0	0	0	eth0
172.17.0.0	*	255.255.0.0	U	0	0	0	eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	213.121.147.69	0.0.0.0	UG	0	0	0	ppp0

There are three interfaces: `eth0`, `eth1` and `ppp0` (as well as `lo`)

Packets with addresses in the network `172.18` go directly on interface eth0;

IP Addresses

A table from a machine with more than one real interface:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
213.121.147.69	*	255.255.255.255	UH	0	0	0	ppp0
172.18.0.0	*	255.255.0.0	U	0	0	0	eth0
172.17.0.0	*	255.255.0.0	U	0	0	0	eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	213.121.147.69	0.0.0.0	UG	0	0	0	ppp0

There are three interfaces: `eth0`, `eth1` and `ppp0` (as well as `lo`)

Packets with addresses in the network `172.17` go directly on interface eth1;

IP Addresses

A table from a machine with more than one real interface:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
213.121.147.69	*	255.255.255.255	UH	0	0	0	ppp0
172.18.0.0	*	255.255.0.0	U	0	0	0	eth0
172.17.0.0	*	255.255.0.0	U	0	0	0	eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	213.121.147.69	0.0.0.0	UG	0	0	0	ppp0

There are three interfaces: `eth0`, `eth1` and `ppp0` (as well as `lo`)

Otherwise packets are routed to the gateway `213.121.147.69` on the interface ppp0;

IP Addresses

A table from a machine with more than one real interface:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
213.121.147.69	*	255.255.255.255	UH	0	0	0	ppp0
172.18.0.0	*	255.255.0.0	U	0	0	0	eth0
172.17.0.0	*	255.255.0.0	U	0	0	0	eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	213.121.147.69	0.0.0.0	UG	0	0	0	ppp0

There are three interfaces: `eth0`, `eth1` and `ppp0` (as well as `lo`)

The first row of the table is actually redundant here, but is kept as it can speed up the checking in a common case

IP Addresses

A table from a machine with more than one real interface:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
213.121.147.69	*	255.255.255.255	UH	0	0	0	ppp0
172.18.0.0	*	255.255.0.0	U	0	0	0	eth0
172.17.0.0	*	255.255.0.0	U	0	0	0	eth1
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	213.121.147.69	0.0.0.0	UG	0	0	0	ppp0

There are three interfaces: `eth0`, `eth1` and `ppp0` (as well as `lo`)

Other information, in particular the flags, will be explained later

IP Addresses

Exercise Look at the routing table on your machine. E.g.,

```
ip -r route show
```

under Linux.

IP Addresses

How should we divide the 32 bit IP address into network and host parts?

IP Addresses

How should we divide the 32 bit IP address into network and host parts?

8 bits for network? Then $2^8 = 256$ networks each with $2^{24} = 16777216$ possible hosts

IP Addresses

How should we divide the 32 bit IP address into network and host parts?

8 bits for network? Then $2^8 = 256$ networks each with $2^{24} = 16777216$ possible hosts

- Not enough networks for the entire Internet!

IP Addresses

How should we divide the 32 bit IP address into network and host parts?

8 bits for network? Then $2^8 = 256$ networks each with $2^{24} = 16777216$ possible hosts

- Not enough networks for the entire Internet!
- Too many hosts for most institutions

IP Addresses

24 bits for network? Then 16777216 networks each with 256 possible hosts

IP Addresses

24 bits for network? Then 16777216 networks each with 256 possible hosts

- Plenty of networks

IP Addresses

24 bits for network? Then 16777216 networks each with 256 possible hosts

- Plenty of networks
- Not enough hosts per network for large installations

IP Addresses

16 bits for network? Then 65536 networks each with 65536 possible hosts

IP Addresses

16 bits for network? Then 65536 networks each with 65536 possible hosts

- Not really enough networks

IP Addresses

16 bits for network? Then 65536 networks each with 65536 possible hosts

- Not really enough networks
- Plenty of hosts per network

IP Addresses

One solution: do *all* of the above

IP Addresses

One solution: do *all* of the above

Divide the space of IP addresses into parts, where each part has a different network/host split

IP Addresses

One solution: do *all* of the above

Divide the space of IP addresses into parts, where each part has a different network/host split

We shall now describe a *class* scheme that is historically important, but no longer used — for reasons that will become clear

IP Addresses

Class A networks. From `0.0.0.0` to `127.255.255.255`

IP Addresses

Class A networks. From `0.0.0.0` to `127.255.255.255`

The leading bit of the address is 0

IP Addresses

Class A networks. From `0.0.0.0` to `127.255.255.255`

The leading bit of the address is 0

Have 7 bits for network and 24 bits for host

IP Addresses

Class A networks. From `0.0.0.0` to `127.255.255.255`

The leading bit of the address is 0

Have 7 bits for network and 24 bits for host

This is 126 networks (networks 0 and 127 are reserved) each with 16777216 host addresses

IP Addresses

Class A networks. From `0.0.0.0` to `127.255.255.255`

The leading bit of the address is 0

Have 7 bits for network and 24 bits for host

This is 126 networks (networks 0 and 127 are reserved) each with 16777216 host addresses

The dotted-decimal representation is helpful here: address `x.y.z.w` has `x` as network, `y.z.w` as host

IP Addresses

Class B networks. From `128.0.0.0` to `191.255.255.255`

IP Addresses

Class B networks. From `128.0.0.0` to `191.255.255.255`

The leading bits of the address are 10

IP Addresses

Class B networks. From `128.0.0.0` to `191.255.255.255`

The leading bits of the address are 10

Have 14 bits for network and 16 bits for host

IP Addresses

Class B networks. From `128.0.0.0` to `191.255.255.255`

The leading bits of the address are 10

Have 14 bits for network and 16 bits for host

This is 16384 networks each with 65536 host addresses

IP Addresses

Class B networks. From `128.0.0.0` to `191.255.255.255`

The leading bits of the address are 10

Have 14 bits for network and 16 bits for host

This is 16384 networks each with 65536 host addresses

The address `x.y.z.w` has `x.y` as network, `z.w` as host

IP Addresses

Class C networks. From `192.0.0.0` to `223.255.255.255`

IP Addresses

Class C networks. From `192.0.0.0` to `223.255.255.255`

The leading bits of the address are 110

IP Addresses

Class C networks. From `192.0.0.0` to `223.255.255.255`

The leading bits of the address are 110

Have 21 bits for network and 8 bits for host

IP Addresses

Class C networks. From `192.0.0.0` to `223.255.255.255`

The leading bits of the address are 110

Have 21 bits for network and 8 bits for host

This is 2097152 networks each with 256 host addresses

IP Addresses

Class C networks. From `192.0.0.0` to `223.255.255.255`

The leading bits of the address are 110

Have 21 bits for network and 8 bits for host

This is 2097152 networks each with 256 host addresses

The address `x.y.z.w` has `x.y.z` as network, `w` as host

IP Addresses

The remaining addresses are kept for separate purposes

IP Addresses

The remaining addresses are kept for separate purposes

Class D. `224.0.0.0` to `239.255.255.255`; leading bits 1110. Used for *multicasting* (details later)

IP Addresses

The remaining addresses are kept for separate purposes

Class D. `224.0.0.0` to `239.255.255.255`; leading bits 1110. Used for *multicasting* (details later)

Class E. `240.0.0.0` to `255.255.255.255`; leading bits 1111. Reserved for future use