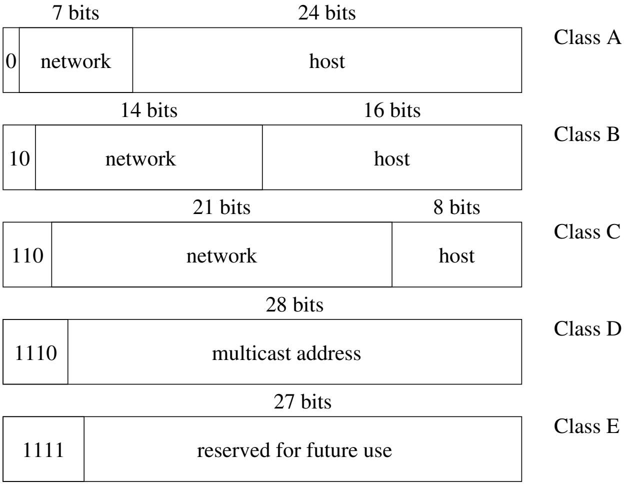
IP Addresses



IP address ranges

IP Addresses

An example: the University of Bath has been allocated addresses in the network `138.38.0.0`

IP Addresses

An example: the University of Bath has been allocated addresses in the network `138.38.0.0`

This is in the class B address range and so there are 65534 possible hosts

IP Addresses

An example: the University of Bath has been allocated addresses in the network `138.38.0.0`

This is in the class B address range and so there are 65534 possible hosts

Network 17.0.0.0, a class A address, is allocated to Apple

IP Addresses

An example: the University of Bath has been allocated addresses in the network `138.38.0.0`

This is in the class B address range and so there are 65534 possible hosts

Network 17.0.0.0, a class A address, is allocated to Apple

Network 193.0.0.0, a class C address, is allocated to *Réseaux IP Européens* (RIPE), the Internet Registry responsible for the allocation of IP addresses within Europe

IP Addresses

Two of the host addresses on each network are treated specially

IP Addresses

Two of the host addresses on each network are treated specially

Host parts of “all zeros” and “all ones” are not used as general host addresses, but are reserved for a special purpose

IP Addresses

Two of the host addresses on each network are treated specially

Host parts of “all zeros” and “all ones” are not used as general host addresses, but are reserved for a special purpose

E.g., `138.38.0.0` and `138.38.255.255` in a class B

IP Addresses

Two of the host addresses on each network are treated specially

Host parts of “all zeros” and “all ones” are not used as general host addresses, but are reserved for a special purpose

E.g., `138.38.0.0` and `138.38.255.255` in a class B

Thus the number of usable host addresses in a network is 2 fewer than you might think

IP Addresses

- Host part all 0s: “this host”. Originally specified to refer back to the originating host. But some implementations mistakenly used this as a broadcast address, so for safety it is not commonly supported as a valid host address. For, say, a class B network `17.16`, a packet sent to `172.16.0.0` *should* boomerang right back to the sender. But rarely does

IP Addresses

- Host part all 0s: “this host”. Originally specified to refer back to the originating host. But some implementations mistakenly used this as a broadcast address, so for safety it is not commonly supported as a valid host address. For, say, a class B network `17.16`, a packet sent to `172.16.0.0` *should* boomerang right back to the sender. But rarely does
- Host part all 1s: broadcast address to network. E.g., `172.16.255.255` sends to all hosts on the `172.16` network; very commonly used

IP Addresses

- Host part all 0s: “this host”. Originally specified to refer back to the originating host. But some implementations mistakenly used this as a broadcast address, so for safety it is not commonly supported as a valid host address. For, say, a class B network `17.16`, a packet sent to `172.16.0.0` *should* boomerang right back to the sender. But rarely does
- Host part all 1s: broadcast address to network. E.g., `172.16.255.255` sends to all hosts on the `172.16` network; very commonly used
- (Network part all 0s: “this network”. E.g., `0.0.12.34` would send to a host on the current B network. Again, not often implemented)

IP Addresses

So this is why you have two fewer addresses available than you might think

IP Addresses

So this is why you have two fewer addresses available than you might think

- ...255 is a broadcast address

IP Addresses

So this is why you have two fewer addresses available than you might think

- . . . 255 is a broadcast address
- . . . 0 may or may not be supported, so best to avoid it

IP Addresses

And there are several special addresses, for example loopback addresses:

IP Addresses

And there are several special addresses, for example loopback addresses:

- Network `127.0.0.0`: the *loopback network*. Always implemented. The address `127.0.0.1` is commonly used as a way for a host to send a packet to itself over the internal loopback network on interface `lo`.

IP Addresses

And there are several special addresses, for example loopback addresses:

- Network `127.0.0.0`: the *loopback network*. Always implemented. The address `127.0.0.1` is commonly used as a way for a host to send a packet to itself over the internal loopback network on interface `lo`.
- Notice this is different from the same host sending to itself via an external network (e.g., using the interface's own address) as the former packet possibly won't go through the normal Ethernet/whatever software and hardware.

IP Addresses

And there are several special addresses, for example loopback addresses:

- Network `127.0.0.0`: the *loopback network*. Always implemented. The address `127.0.0.1` is commonly used as a way for a host to send a packet to itself over the internal loopback network on interface `lo`.
- Notice this is different from the same host sending to itself via an external network (e.g., using the interface's own address) as the former packet possibly won't go through the normal Ethernet/whatever software and hardware.
- The loopback network is there even if there is no real network hardware attached

IP Addresses

So the class scheme allowed IANA to allocate large chunks of addresses to people who need them, and small chunks to those that only need a few

IP Addresses

So the class scheme allowed IANA to allocate large chunks of addresses to people who need them, and small chunks to those that only need a few

This scheme has been historically very successful, but with the growth of the Internet has revealed several weaknesses. These days, a *classless* allocation is used (CIDR, later)

IP Addresses

So the class scheme allowed IANA to allocate large chunks of addresses to people who need them, and small chunks to those that only need a few

This scheme has been historically very successful, but with the growth of the Internet has revealed several weaknesses. These days, a *classless* allocation is used (CIDR, later)

Thus this allocation is sometime called *classful*

IP Addresses

So the class scheme allowed IANA to allocate large chunks of addresses to people who need them, and small chunks to those that only need a few

This scheme has been historically very successful, but with the growth of the Internet has revealed several weaknesses. These days, a *classless* allocation is used (CIDR, later)

Thus this allocation is sometime called *classful*

To understand classless allocation, we first need to look at *subnetting*

IP Address Subnetting

Suppose you have been allocated class B network: 64 thousand host addresses are very hard to manage

IP Address Subnetting

Suppose you have been allocated class B network: 64 thousand host addresses are very hard to manage

Think of the broadcast traffic (e.g., ARP)

IP Address Subnetting

Suppose you have been allocated class B network: 64 thousand host addresses are very hard to manage

Think of the broadcast traffic (e.g., ARP)

Physical/Technical issues (e.g, limits on Ethernet)

IP Address Subnetting

Suppose you have been allocated class B network: 64 thousand host addresses are very hard to manage

Think of the broadcast traffic (e.g., ARP)

Physical/Technical issues (e.g, limits on Ethernet)

Political issues (e.g., traffic from one department must be kept separate from another department)

IP Address Subnetting

Suppose you have been allocated class B network: 64 thousand host addresses are very hard to manage

Think of the broadcast traffic (e.g., ARP)

Physical/Technical issues (e.g, limits on Ethernet)

Political issues (e.g., traffic from one department must be kept separate from another department)

A single big network is not a very good idea

IP Address Subnetting

We can use *subnetting* to split our network into smaller pieces

IP Address Subnetting

We can use *subnetting* to split our network into smaller pieces

Subnets can be administered by separate departments and are joined by routers

IP Address Subnetting

We can use *subnetting* to split our network into smaller pieces

Subnets can be administered by separate departments and are joined by routers

Just like the Internet!

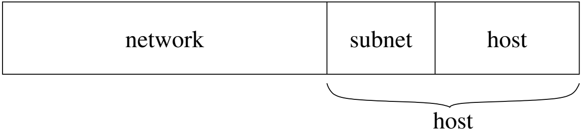
IP Address Subnetting

We can use *subnetting* to split our network into smaller pieces

Subnets can be administered by separate departments and are joined by routers

Just like the Internet!

And to do this, also just like the Internet, we further split the host part into some bits for the subnetwork and the rest for the actual hosts



Subnet addressing

IP Address Subnetting

Hosts will need to know which bits are the subnet part to be able to decide how to route packets: there is no class system here

IP Address Subnetting

Hosts will need to know which bits are the subnet part to be able to decide how to route packets: there is no class system here

We use a *subnet mask*

IP Address Subnetting

Hosts will need to know which bits are the subnet part to be able to decide how to route packets: there is no class system here

We use a *subnet mask*

For example, the University of Bath has a class B, address `138.38`. The top 16 bits are the network address

IP Address Subnetting

Hosts will need to know which bits are the subnet part to be able to decide how to route packets: there is no class system here

We use a *subnet mask*

For example, the University of Bath has a class B, address `138.38`. The top 16 bits are the network address

The netmask 111111111111000000000000 indicates which bits are in the network part

IP Address Subnetting

The Department of Mathematical Sciences has a subnet consisting of addresses `138.38.96.0` to `138.38.103.255` (2048 host addresses)

IP Address Subnetting

The Department of Mathematical Sciences has a subnet consisting of addresses `138.38.96.0` to `138.38.103.255` (2048 host addresses)

This corresponds to the netmask
11111111111111111111100000000000

IP Address Subnetting

network address	138.38.96.0	10001010 00100110 01100000 00000000
broadcast address	138.38.103.255	10001010 00100110 01100111 11111111
netmask	255.255.248.0	11111111 11111111 11111000 00000000

IP Address Subnetting

network address	138.38.96.0	10001010 00100110 01100000 00000000
broadcast address	138.38.103.255	10001010 00100110 01100111 11111111
netmask	255.255.248.0	11111111 11111111 11111000 00000000

A machine can tell if an address is on a network if the address ANDed with the netmask gives the network address

IP Address Subnetting

network address	138.38.96.0	10001010 00100110 01100000 00000000
broadcast address	138.38.103.255	10001010 00100110 01100111 11111111
netmask	255.255.248.0	11111111 11111111 11111000 00000000

A machine can tell if an address is on a network if the address ANDed with the netmask gives the network address

This is not on a nice byte boundary, so visually is harder for humans to work with using decimal `x.y.z.w` style notations

IP Address Subnetting

So `138.38.100.20` *is* on the subnet

host address	138.38.100.20	10001010 00100110 01100100 00010100
netmask	255.255.248.0	11111111 11111111 11111000 00000000
AND	138.38.96.0	10001010 00100110 01100000 00000000
network address	138.38.96.0	10001010 00100110 01100000 00000000

IP Address Subnetting

But `138.38.104.20` is *not* on the subnet

host address	138.38.104.20	10001010 00100110 01101000 00010100
netmask	255.255.248.0	11111111 11111111 11111000 00000000
AND	138.38.104.0	10001010 00100110 01101000 00000000
network address	138.38.96.0	10001010 00100110 01100000 00000000

IP Address Subnetting

`138.38` is split into many subnets of appropriate sizes for each Department, Centre or other sub-part of the University

IP Address Subnetting

`138.38` is split into many subnets of appropriate sizes for each Department, Centre or other sub-part of the University

Outside of `138.38` the subnetting is invisible so no changes to global routing tables are necessary if we rearrange our network

IP Address Subnetting

`138.38` is split into many subnets of appropriate sizes for each Department, Centre or other sub-part of the University

Outside of `138.38` the subnetting is invisible so no changes to global routing tables are necessary if we rearrange our network

Subnets can be further subnetted for exactly the same reason

IP Address Subnetting

The subnet is described as "`138.38.96.0`, netmask `255.255.248.0`"

IP Address Subnetting

The subnet is described as “`138.38.96.0`, netmask `255.255.248.0`”

More commonly as “`138.38.96.0/21`”, where 21 is the number of 1 bits in the netmask

IP Address Subnetting

The subnet is described as "`138.38.96.0`, netmask `255.255.248.0`"

More commonly as "`138.38.96.0/21`", where 21 is the number of 1 bits in the netmask

You don't have to use the top *n* bits for a netmask, but it is overwhelmingly common to do so

IP Address Subnetting

The subnet is described as "`138.38.96.0`, netmask `255.255.248.0`"

More commonly as "`138.38.96.0/21`", where 21 is the number of 1 bits in the netmask

You don't have to use the top *n* bits for a netmask, but it is overwhelmingly common to do so

The /*n* notation is only for a top-*n*-bit netmask

IP Address Subnetting

The subnet is described as "`138.38.96.0`, netmask `255.255.248.0`"

More commonly as "`138.38.96.0/21`", where 21 is the number of 1 bits in the netmask

You don't have to use the top *n* bits for a netmask, but it is overwhelmingly common to do so

The /*n* notation is only for a top-*n*-bit netmask

The "all 0s" and "all 1s" addresses now apply within the *subnet*: all 1's broadcasts to the subnet; and don't use all 0s

IP Address Exhaustion

So how does this relate to classful addressing?

IP Address Exhaustion

So how does this relate to classful addressing?

Everybody wants a class B as C is too small and A is too large

IP Address Exhaustion

So how does this relate to classful addressing?

Everybody wants a class B as C is too small and A is too large

Called the *Three Bears Problem*

IP Address Exhaustion

So how does this relate to classful addressing?

Everybody wants a class B as C is too small and A is too large

Called the *Three Bears Problem*

There are no class Bs left: they have all been allocated

IP Address Exhaustion

So how does this relate to classful addressing?

Everybody wants a class B as C is too small and A is too large

Called the *Three Bears Problem*

There are no class Bs left: they have all been allocated

But, as the Internet grows, people want more addresses

IP Address Exhaustion

Can we split some class As?

IP Address Exhaustion

Can we split some class As?

Doable, but needs everyone to take care their software understands that those addresses are no longer class A

IP Address Exhaustion

Can we split some class As?

Doable, but needs everyone to take care their software understands that those addresses are no longer class A

Most class A's have now been split and the subnets allocated to various institutions

IP Address Exhaustion

Can an institution simply use several class Cs?

IP Address Exhaustion

Can an institution simply use several class Cs?

Yes, but awkward as this leads to multiple networks, each needing separate routing

IP Address Exhaustion

Can an institution simply use several class Cs?

Yes, but awkward as this leads to multiple networks, each needing separate routing

For example, having eight class C networks 194.24.0.0 to 194.24.7.0 would require all global routers' tables to have eight entries that all point to the same destination

IP Address Exhaustion

Can an institution simply use several class Cs?

Yes, but awkward as this leads to multiple networks, each needing separate routing

For example, having eight class C networks 194.24.0.0 to 194.24.7.0 would require all global routers' tables to have eight entries that all point to the same destination

And internally to the institution there are eight separate networks, too

IP Address Exhaustion

Class E has 286 million reserved addresses; can we use them?

IP Address Exhaustion

Class E has 286 million reserved addresses; can we use them?

Wouldn’t last long; perhaps under a couple of years if allocated

IP Address Exhaustion

Class E has 286 million reserved addresses; can we use them?

Wouldn't last long; perhaps under a couple of years if allocated

More problematically, class E addresses are treated as illegal by much software, particularly on routers, so they are difficult to bring into play

IP Address Exhaustion

Class E has 286 million reserved addresses; can we use them?

Wouldn't last long; perhaps under a couple of years if allocated

More problematically, class E addresses are treated as illegal by much software, particularly on routers, so they are difficult to bring into play

(A recurrent problem with improving Internet protocols: a lot of software out there assumes the old way of doing things is the only way, and rejects any patterns or protocols it doesn't recognise)

IP Address Exhaustion

Some while ago it was recognised that the growth of the Internet meant that a new way of allocating addresses was needed

IP Address Exhaustion

Some while ago it was recognised that the growth of the Internet meant that a new way of allocating addresses was needed

Three solutions are used:

IP Address Exhaustion

Some while ago it was recognised that the growth of the Internet meant that a new way of allocating addresses was needed

Three solutions are used:

- Change the way classes are defined and used

IP Address Exhaustion

Some while ago it was recognised that the growth of the Internet meant that a new way of allocating addresses was needed

Three solutions are used:

- Change the way classes are defined and used
- Use private addresses with network address translation

IP Address Exhaustion

Some while ago it was recognised that the growth of the Internet meant that a new way of allocating addresses was needed

Three solutions are used:

- Change the way classes are defined and used
- Use private addresses with network address translation
- Increase the number of addresses available by changing the IP

IP Address Exhaustion

Some while ago it was recognised that the growth of the Internet meant that a new way of allocating addresses was needed

Three solutions are used:

- Change the way classes are defined and used
- Use private addresses with network address translation
- Increase the number of addresses available by changing the IP

We shall be looking at each of these

CIDR

Classless Interdomain Routing (CIDR) takes class C networks and joins them together in such a way that simplifies routing

CIDR

Classless Interdomain Routing (CIDR) takes class C networks and joins them together in such a way that simplifies routing

Blocks of C addresses are allocated to regions, e.g.,

194.0.0.0-195.255.255.255	Europe
198.0.0.0-199.255.255.255	North America
200.0.0.0-201.255.255.255	Central and S America
202.0.0.0-203.255.255.255	Asia and the Pacific

CIDR

Starting with about 32 million addresses per region

CIDR

Starting with about 32 million addresses per region

This allows easy routing: anything 194 or 195 goes to Europe

CIDR

Starting with about 32 million addresses per region

This allows easy routing: anything 194 or 195 goes to Europe

Repeat the idea within each region: contiguous block of C networks are allocated to ISPs or organisations

CIDR

Starting with about 32 million addresses per region

This allows easy routing: anything 194 or 195 goes to Europe

Repeat the idea within each region: contiguous block of C networks are allocated to ISPs or organisations

Keeps simple routing within the region

CIDR

Starting with about 32 million addresses per region

This allows easy routing: anything 194 or 195 goes to Europe

Repeat the idea within each region: contiguous block of C networks are allocated to ISPs or organisations

Keeps simple routing within the region

Note that the software within routers does need to be updated to support this: but this has now been done everywhere

CIDR

E.g., `194.24.0.0` to `194.24.7.255`, normally written `194.24.0.0/21` or even `194.24/21`: exactly like subnetting

CIDR

E.g., `194.24.0.0` to `194.24.7.255`, normally written `194.24.0.0/21` or even `194.24/21`: exactly like subnetting

194.24.0.0	11000010 00011000 00000000 00000000
194.24.7.255	11000010 00011000 00000111 11111111
255.255.248.0	11111111 11111111 11111000 00000000

CIDR

E.g., `194.24.0.0` to `194.24.7.255`, normally written `194.24.0.0/21` or even `194.24/21`: exactly like subnetting

194.24.0.0	11000010 00011000 00000000 00000000
194.24.7.255	11000010 00011000 00000111 11111111
255.255.248.0	11111111 11111111 11111000 00000000

Any packet with address that has addr AND `255.255.248.0` = `194.24.0.0` should be routed to that ISP or organisation

CIDR

E.g., `194.24.0.0` to `194.24.7.255`, normally written `194.24.0.0/21` or even `194.24/21`: exactly like subnetting

194.24.0.0	11000010 00011000 00000000 00000000
194.24.7.255	11000010 00011000 00000111 11111111
255.255.248.0	11111111 11111111 11111000 00000000

Any packet with address that has addr AND `255.255.248.0` = `194.24.0.0` should be routed to that ISP or organisation

A network of $2^{32-21} = 2^{11} = 2048$ addresses, i.e., 2046 hosts

CIDR

This is a very flexible and backwards-compatible scheme

CIDR

This is a very flexible and backwards-compatible scheme

End hosts do not need to know about CIDR

CIDR

This is a very flexible and backwards-compatible scheme

End hosts do not need to know about CIDR

Classless networks can be subnetted

CIDR

This is a very flexible and backwards-compatible scheme

End hosts do not need to know about CIDR

Classless networks can be subnetted

CIDR has allowed the continued growth of the Internet well beyond the original possible size by using addresses that would otherwise be wasted: allocated but not used

CIDR

This is a very flexible and backwards-compatible scheme

End hosts do not need to know about CIDR

Classless networks can be subnetted

CIDR has allowed the continued growth of the Internet well beyond the original possible size by using addresses that would otherwise be wasted: allocated but not used

And we have repurposed class A and B networks similarly

CIDR

In fact, **classful networks are no longer used**: CIDR is the only way addresses are currently allocated

CIDR

In fact, **classful networks are no longer used**: CIDR is the only way addresses are currently allocated

CIDR merges small networks into a larger one

CIDR

In fact, **classful networks are no longer used**: CIDR is the only way addresses are currently allocated

CIDR merges small networks into a larger one

Subnetting divides a large network into smaller ones

CIDR

In fact, **classful networks are no longer used**: CIDR is the only way addresses are currently allocated

CIDR merges small networks into a larger one

Subnetting divides a large network into smaller ones

CIDR is sometimes called *supernetting*

CIDR

In fact, **classful networks are no longer used**: CIDR is the only way addresses are currently allocated

CIDR merges small networks into a larger one

Subnetting divides a large network into smaller ones

CIDR is sometimes called *supernetting*

Thus we have:

- Classful: implicit, fixed split of network/host
- Classless: explicit (netmask), variable split of network/host

CIDR

CIDR has been very successful, and has extended the life of the Internet significantly by providing a source of addresses from the previously underutilised classful ranges

CIDR

CIDR has been very successful, and has extended the life of the Internet significantly by providing a source of addresses from the previously underutilised classful ranges

Not enough...

Addresses

There are currently about 51 billion devices connected to the Internet (`www.statistica.com`; 2023)

Addresses

There are currently about 51 billion devices connected to the Internet (`www.statistica.com`; 2023)

But there are only about 4.3 billion usable IPv4 addresses

Addresses

There are currently about 51 billion devices connected to the Internet (`www.statistica.com`; 2023)

But there are only about 4.3 billion usable IPv4 addresses

How is this possible?

NAT

This brings us to the second approach to address exhaustion

NAT

This brings us to the second approach to address exhaustion

Some IP addresses are reserved for *private networks*, originally reserved to allow local experimentation:

- 10.0.0.0-10.255.255.255 (Class A)
- 172.16.0.0-172.31.255.255 (Class B)
- 192.168.0.0-192.168.255.255 (Class C)

NAT

This brings us to the second approach to address exhaustion

Some IP addresses are reserved for *private networks*, originally reserved to allow local experimentation:

- 10.0.0.0-10.255.255.255 (Class A)
- 172.16.0.0-172.31.255.255 (Class B)
- 192.168.0.0-192.168.255.255 (Class C)

One class A-size network, 16 class B and 256 class C-size networks are guaranteed never to be allocated for public use in the Internet

NAT

Routers on the public Internet will never forward packets with such addresses, and will simply drop them immediately

NAT

Routers on the public Internet will never forward packets with such addresses, and will simply drop them immediately

They are called *unroutable* addresses

NAT

But such addresses can be used by anyone locally for any purpose: a common use is NAT

NAT

But such addresses can be used by anyone locally for any purpose: a common use is NAT

Network Address Translation (NAT) uses the malleability of packets to map many hosts onto a single address

NAT

But such addresses can be used by anyone locally for any purpose: a common use is NAT

Network Address Translation (NAT) uses the malleability of packets to map many hosts onto a single address

A private network can be set up, using one of the above address ranges, e.g., `10/8`

NAT

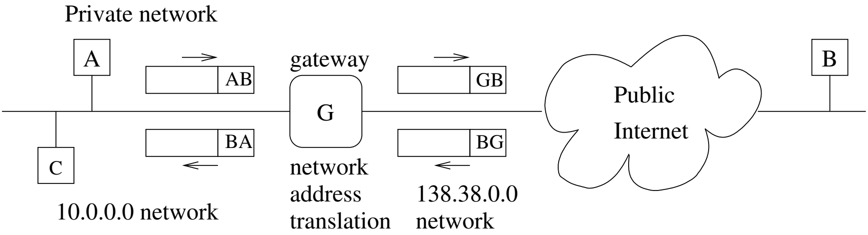
But such addresses can be used by anyone locally for any purpose: a common use is NAT

Network Address Translation (NAT) uses the malleability of packets to map many hosts onto a single address

A private network can be set up, using one of the above address ranges, e.g., `10/8`

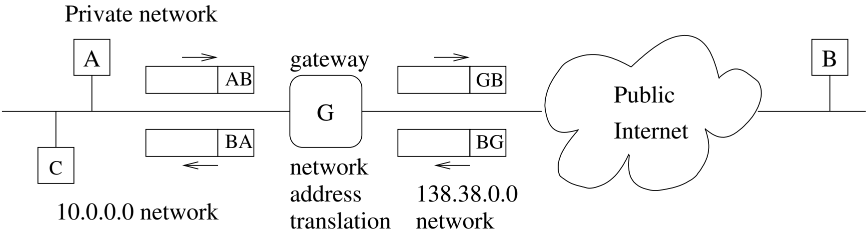
A gateway host joins the private network to the public Internet, rewriting the addresses on packets as they go past

NAT



Network Address Translation

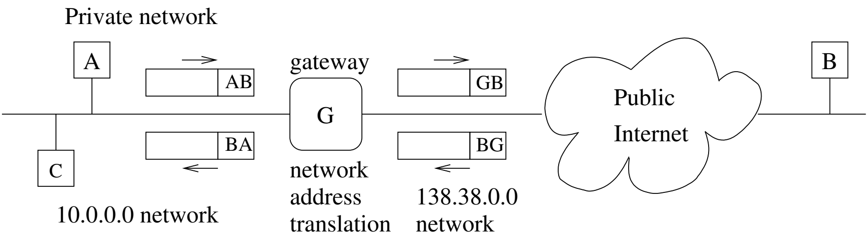
NAT



Network Address Translation

A packet from `10.0.1.1` (A) is sent to `212.58.226.33` (B); B is not on the local network so the packet is sent to the gateway;

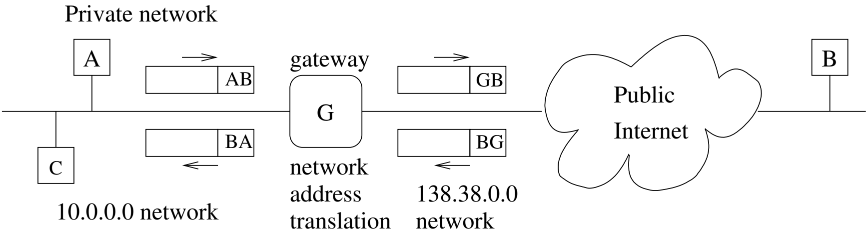
NAT



Network Address Translation

The gateway overwrites the source address with its own public address (G) and forwards the packet;

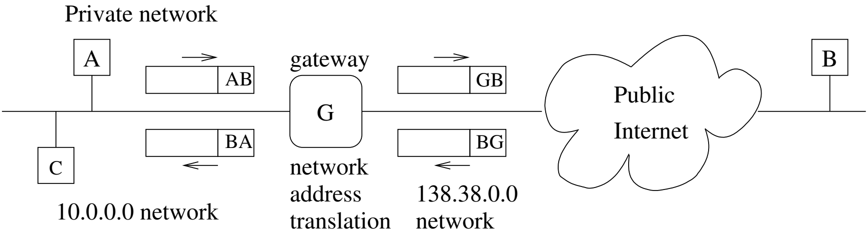
NAT



Network Address Translation

The packet reaches B in the normal way;

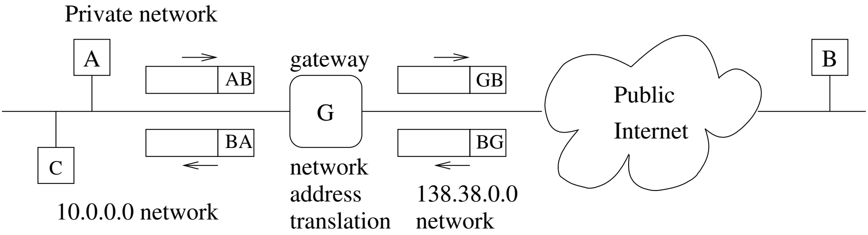
NAT



Network Address Translation

B replies with a packet with destination address G;

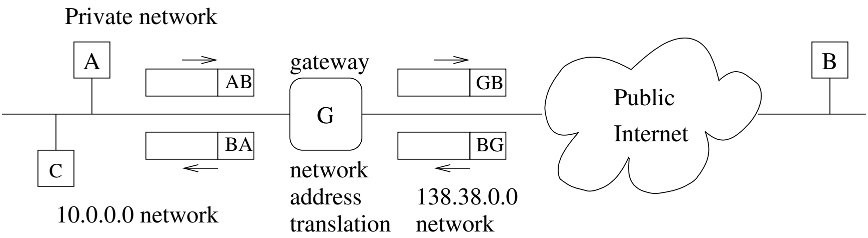
NAT



Network Address Translation

The gateway recognises this packet as a reply to A and rewrites the destination address to A before passing it on to the private network;

NAT



Network Address Translation

A thinks it is connected to the public Internet, and B thinks data is coming from G

NAT

G needs to keep a record of connections from A to the world and recognise replies to outward travelling packets

NAT

G needs to keep a record of connections from A to the world and recognise replies to outward travelling packets

C will want to do the same as A; so G must be able to distinguish replies to A from replies to C; even if both were communicating with B

NAT

G needs to keep a record of connections from A to the world and recognise replies to outward travelling packets

C will want to do the same as A; so G must be able to distinguish replies to A from replies to C; even if both were communicating with B

And rewrite the replies to C with C's address

NAT

G needs to keep a record of connections from A to the world and recognise replies to outward travelling packets

C will want to do the same as A; so G must be able to distinguish replies to A from replies to C; even if both were communicating with B

And rewrite the replies to C with C's address

This is all doable in practice!

NAT

G needs to keep a record of connections from A to the world and recognise replies to outward travelling packets

C will want to do the same as A; so G must be able to distinguish replies to A from replies to C; even if both were communicating with B

And rewrite the replies to C with C's address

This is all doable in practice!

Explanation later, in the next layer

NAT

Exercise If both A and C are communicating with B, what are the addresses on their packets as they reach B? And on the replies as they reach G?

NAT

Exercise If both A and C are communicating with B, what are the addresses on their packets as they reach B? And on the replies as they reach G?

Exercise Compare with *bridging*, a similar idea but for very different reasons

NAT

As a fortunate side-effect, NAT provides some measure of protection to hosts on the private network from external attack

NAT

As a fortunate side-effect, NAT provides some measure of protection to hosts on the private network from external attack

Machines on the public Internet (e.g., B) cannot initiate traffic to A as `10.0.1.1` is a private, unroutable address

NAT

As a fortunate side-effect, NAT provides some measure of protection to hosts on the private network from external attack

Machines on the public Internet (e.g., B) cannot initiate traffic to A as `10.0.1.1` is a private, unroutable address

No public router will forward a packet with such an address: it will simply drop it

NAT

As a fortunate side-effect, NAT provides some measure of protection to hosts on the private network from external attack

Machines on the public Internet (e.g., B) cannot initiate traffic to A as `10.0.1.1` is a private, unroutable address

No public router will forward a packet with such an address: it will simply drop it

External hosts will generally not even know what A's address is as they never get to see it

NAT

As a fortunate side-effect, NAT provides some measure of protection to hosts on the private network from external attack

Machines on the public Internet (e.g., B) cannot initiate traffic to A as `10.0.1.1` is a private, unroutable address

No public router will forward a packet with such an address: it will simply drop it

External hosts will generally not even know what A's address is as they never get to see it

Even if a packet somehow gets to the gateway, the gateway will not know how to rewrite its address as this was not a reply to an outgoing packet; so it get dropped here, too