NAT

NAT has helped immensely to mitigate the address exhaustion problem

NAT

NAT has helped immensely to mitigate the address exhaustion problem

Previously, every host on a network would need a unique public IP address

NAT

NAT has helped immensely to mitigate the address exhaustion problem

Previously, every host on a network would need a unique public IP address

The growth of the Internet at home, for example, would have sucked up addresses at a huge rate

NAT

NAT has helped immensely to mitigate the address exhaustion problem

Previously, every host on a network would need a unique public IP address

The growth of the Internet at home, for example, would have sucked up addresses at a huge rate

But now all your home appliances can share just one public address

NAT

NAT has helped immensely to mitigate the address exhaustion problem

Previously, every host on a network would need a unique public IP address

The growth of the Internet at home, for example, would have sucked up addresses at a huge rate

But now all your home appliances can share just one public address

Exercise Count the number of network attached devices you have at home

NAT

Problems arise when the *data* in the packet contain IP addresses that, say, will be used to set up new connections. E.g., the *File Transfer Protocol* (FTP)

NAT

Problems arise when the *data* in the packet contain IP addresses that, say, will be used to set up new connections. E.g., the *File Transfer Protocol* (FTP)

(Original) FTP would send an IP address to the server to indicate where to set up a new connection

NAT

Problems arise when the *data* in the packet contain IP addresses that, say, will be used to set up new connections. E.g., the *File Transfer Protocol* (FTP)

(Original) FTP would send an IP address to the server to indicate where to set up a new connection

In our example, this would be its private, unroutable address that the external server couldn’t contact

NAT

Problems arise when the *data* in the packet contain IP addresses that, say, will be used to set up new connections. E.g., the *File Transfer Protocol* (FTP)

(Original) FTP would send an IP address to the server to indicate where to set up a new connection

In our example, this would be its private, unroutable address that the external server couldn't contact

Unless the gateway is intelligent enough to realise this is an FTP exchange, look inside the data and know where the IP addresses are to be found (in the application layer data) and rewrite them (in the application layer data) the addresses will remain untranslated and the protocol will fail

NAT

Not many protocols do this kind of thing these days, but each one of those that do must be treated specially by the NAT gateway

NAT

Not many protocols do this kind of thing these days, but each one of those that do must be treated specially by the NAT gateway

Note this is a problem due to a violation of layering in the protocol: IP layer information in the application layer

NAT

Not many protocols do this kind of thing these days, but each one of those that do must be treated specially by the NAT gateway

Note this is a problem due to a violation of layering in the protocol: IP layer information in the application layer

Exercise Read about FTP, *Universal Plug and Play* (UPnP) and the *Simple Service Discovery Protocol* (SSDP)

NAT

NAT is used widely as it is very effective

NAT

NAT is used widely as it is very effective

It allows you to have many machines but only use one public address

NAT

NAT is used widely as it is very effective

It allows you to have many machines but only use one public address

Many mobile phone companies are now using *carrier grade NAT* to supply IP connectivity to the millions of phones they manage

NAT

NAT is used widely as it is very effective

It allows you to have many machines but only use one public address

Many mobile phone companies are now using *carrier grade NAT* to supply IP connectivity to the millions of phones they manage

Carrier grade NAT: NAT done in the ISP rather than by the end-user

NAT

NAT is used widely as it is very effective

It allows you to have many machines but only use one public address

Many mobile phone companies are now using *carrier grade NAT* to supply IP connectivity to the millions of phones they manage

Carrier grade NAT: NAT done in the ISP rather than by the end-user

Exercise What IP address does your phone have for its mobile data connection (not its Wi-Fi connection)?

NAT

NAT is used widely as it is very effective

It allows you to have many machines but only use one public address

Many mobile phone companies are now using *carrier grade NAT* to supply IP connectivity to the millions of phones they manage

Carrier grade NAT: NAT done in the ISP rather than by the end-user

Exercise What IP address does your phone have for its mobile data connection (not its Wi-Fi connection)?

Exercise Read RFC6598 and about `100.64.0.0/10`

NAT

Without NAT, public IP addresses would have run out years ago

NAT

Without NAT, public IP addresses would have run out years ago

But there are costs to NAT

NAT

Without NAT, public IP addresses would have run out years ago

But there are costs to NAT

- Complexity in the gateway software

NAT

Without NAT, public IP addresses would have run out years ago

But there are costs to NAT

- Complexity in the gateway software
- Scalability problems in the gateway tracking large numbers of connections

NAT

Without NAT, public IP addresses would have run out years ago

But there are costs to NAT

- Complexity in the gateway software
- Scalability problems in the gateway tracking large numbers of connections
- Bad interactions with some protocols

NAT

Without NAT, public IP addresses would have run out years ago

But there are costs to NAT

- Complexity in the gateway software
- Scalability problems in the gateway tracking large numbers of connections
- Bad interactions with some protocols
- Difficulty of making end-to-end connections when both ends are behind a NAT gateway (e.g., Skype, SIP)

NAT

Without NAT, public IP addresses would have run out years ago

But there are costs to NAT

- Complexity in the gateway software
- Scalability problems in the gateway tracking large numbers of connections
- Bad interactions with some protocols
- Difficulty of making end-to-end connections when both ends are behind a NAT gateway (e.g., Skype, SIP)
- Loss of “an IP address identifies a host uniquely”: a problem for law enforcement

NAT

There is also the inability for external hosts to initiate connections to hosts behind NAT

NAT

There is also the inability for external hosts to initiate connections to hosts behind NAT

So you can’t run servers on hosts behind the NAT

NAT

There is also the inability for external hosts to initiate connections to hosts behind NAT

So you can’t run servers on hosts behind the NAT

But this invisibility is generally a good security feature

NAT

There is also the inability for external hosts to initiate connections to hosts behind NAT

So you can’t run servers on hosts behind the NAT

But this invisibility is generally a good security feature

This can be worked around, though not neatly

NAT

There is also the inability for external hosts to initiate connections to hosts behind NAT

So you can’t run servers on hosts behind the NAT

But this invisibility is generally a good security feature

This can be worked around, though not neatly

Exercise Read about port forwarding (later)

NAT

There is also the inability for external hosts to initiate connections to hosts behind NAT

So you can’t run servers on hosts behind the NAT

But this invisibility is generally a good security feature

This can be worked around, though not neatly

Exercise Read about port forwarding (later)

Exercise Read about STUN

NAT

NAT is the reason the Internet did not grind to a halt many years ago through the lack of available addresses

NAT

NAT is the reason the Internet did not grind to a halt many years ago through the lack of available addresses

Thus putting off the need for a proper solution to the problem

NAT

NAT is the reason the Internet did not grind to a halt many years ago through the lack of available addresses

Thus putting off the need for a proper solution to the problem

Some people still argue that there is no reason to do anything else than use more NAT

NAT

NAT is the reason the Internet did not grind to a halt many years ago through the lack of available addresses

Thus putting off the need for a proper solution to the problem

Some people still argue that there is no reason to do anything else than use more NAT

Even to the extent of using multi-level NAT (NAT within NAT)!

NAT

But even with CIDR and NAT, the entire range of usable IPv4 addresses has now been allocated

NAT

But even with CIDR and NAT, the entire range of usable IPv4 addresses has now been allocated

2011: IANA has distributed all its reserves of addresses to the Regional Internet Registries (RIRs)

NAT

But even with CIDR and NAT, the entire range of usable IPv4 addresses has now been allocated

2011: IANA has distributed all its reserves of addresses to the Regional Internet Registries (RIRs)

2019: RIPE’s allocation (covering Europe, Middle East and Central Asia) have run out

NAT

But even with CIDR and NAT, the entire range of usable IPv4 addresses has now been allocated

2011: IANA has distributed all its reserves of addresses to the Regional Internet Registries (RIRs)

2019: RIPE's allocation (covering Europe, Middle East and Central Asia) have run out

Old addresses that are no longer needed get recycled; there is even a black market in IP addresses!

NAT

But even with CIDR and NAT, the entire range of usable IPv4 addresses has now been allocated

2011: IANA has distributed all its reserves of addresses to the Regional Internet Registries (RIRs)

2019: RIPE's allocation (covering Europe, Middle East and Central Asia) have run out

Old addresses that are no longer needed get recycled; there is even a black market in IP addresses!

We need a more radical solution

IPv6

The next approach to the IP address exhaustion problem is to change IP itself

IPv6

The next approach to the IP address exhaustion problem is to change IP itself

The next version of the IP is IPv6 (occasionally called IPng for “IP next generation”)

IPv6

The next approach to the IP address exhaustion problem is to change IP itself

The next version of the IP is IPv6 (occasionally called IPng for “IP next generation”)

Slowly growing in use, it will take a while to replace all of IPv4

IPv6

The next approach to the IP address exhaustion problem is to change IP itself

The next version of the IP is IPv6 (occasionally called IPng for “IP next generation”)

Slowly growing in use, it will take a while to replace all of IPv4

128 bit addresses; CIDR-style allocation only

IPv6

The next approach to the IP address exhaustion problem is to change IP itself

The next version of the IP is IPv6 (occasionally called IPng for “IP next generation”)

Slowly growing in use, it will take a while to replace all of IPv4

128 bit addresses; CIDR-style allocation only

Exercise Find out about IPv5. And IPv0-IPv3

IPv6

IPv6 was designed to

IPv6

IPv6 was designed to

- have a larger address space

IPv6

IPv6 was designed to

- have a larger address space
- reduce the size of router tables

IPv6

IPv6 was designed to

- have a larger address space
- reduce the size of router tables
- simplify the protocol so routers can process packets faster

IPv6

IPv6 was designed to

- have a larger address space
- reduce the size of router tables
- simplify the protocol so routers can process packets faster
- provide security and authentication

IPv6

IPv6 was designed to

- have a larger address space
- reduce the size of router tables
- simplify the protocol so routers can process packets faster
- provide security and authentication
- pay proper attention to type of service (DS)

IPv6

- have better multicasting support

IPv6

- have better multicasting support
- have mobile hosts with fixed IP addresses

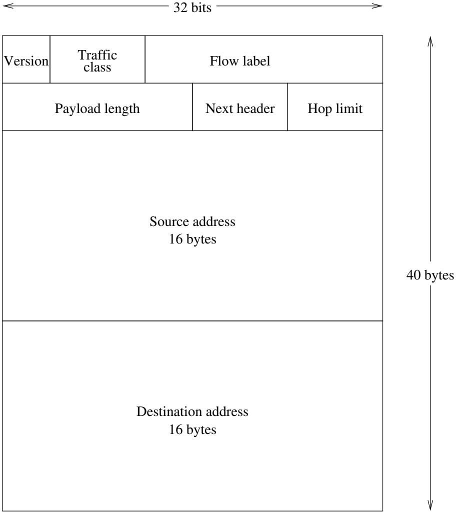
IPv6

- have better multicasting support
- have mobile hosts with fixed IP addresses
- allow room for evolution of the protocol

IPv6

- have better multicasting support
- have mobile hosts with fixed IP addresses
- allow room for evolution of the protocol
- permit IPv4 and IPv6 to coexist during the transition

IPv6



32 bits

Version

Traffic class

Flow label

Payload length

Next header

Hop limit

Source address
16 bytes

Destination address
16 bytes

40 bytes

IPv6 Header

IPv6

- Version, 4 bits. The number 6. This is identical in position to IPv4 and can be used to distinguish packets in mixed-version environments. Additionally, in an Ethernet frame, IPv4 has protocol number 0800, while IPv6 is 86DD, but remember you might be using a different physical layer that does not give the type of its data

IPv6

- Version, 4 bits. The number 6. This is identical in position to IPv4 and can be used to distinguish packets in mixed-version environments. Additionally, in an Ethernet frame, IPv4 has protocol number 0800, while IPv6 is 86DD, but remember you might be using a different physical layer that does not give the type of its data
- Traffic class, 8 bits. Like TOS (DS) in v4

IPv6

- Version, 4 bits. The number 6. This is identical in position to IPv4 and can be used to distinguish packets in mixed-version environments. Additionally, in an Ethernet frame, IPv4 has protocol number 0800, while IPv6 is 86DD, but remember you might be using a different physical layer that does not give the type of its data
- Traffic class, 8 bits. Like TOS (DS) in v4
- Flow label, 20 bits. Allows routers to recognise related packets in a single flow and treat them identically (and so faster)

IPv6

- Payload length, 16 bits. The number of bytes following the fixed 40 byte header. Unlike v4, this not the packet length as it does not include the header in the count

IPv6

- Payload length, 16 bits. The number of bytes following the fixed 40 byte header. Unlike v4, this not the packet length as it does not include the header in the count
- Next header, 8 bits. Like the protocol field in v4, but also allows for v6 optional header fields, if any

IPv6

- Payload length, 16 bits. The number of bytes following the fixed 40 byte header. Unlike v4, this not the packet length as it does not include the header in the count
- Next header, 8 bits. Like the protocol field in v4, but also allows for v6 optional header fields, if any
- Hop limit, 8 bits. The TTL field, renamed to make it clear how it is actually used

IPv6

- Source and destination addresses, 128 bits each.

IPv6

- Source and destination addresses, 128 bits each.

Four times as long as v4 addresses

IPv6

- Source and destination addresses, 128 bits each.

Four times as long as v4 addresses

$2^{128} = 3 \times 10^{38}$ addresses, enough for an address for every molecule on the surface of the Earth

IPv6

- Source and destination addresses, 128 bits each.

Four times as long as v4 addresses

$2^{128} = 3 \times 10^{38}$ addresses, enough for an address for every molecule on the surface of the Earth

There are unicast, multicast and anycast addresses: details later

IPv6

Addresses are typically written in hex, with colon separators, e.g., `fe80:0000:0000:0000:21c:c0ff:fea3:99f4`

IPv6

Addresses are typically written in hex, with colon separators, e.g., `fe80:0000:0000:0000:21c:c0ff:fea3:99f4`

A `::` may appear once as a shorthand for a string of `0`s. As many as you need to make the address up to 128 bits

IPv6

Addresses are typically written in hex, with colon separators, e.g., `fe80:0000:0000:0000:21c:c0ff:fea3:99f4`

A `::` may appear once as a shorthand for a string of `0`s. As many as you need to make the address up to 128 bits

Thus the above address can be written `fe80::21c:c0ff:fea3:99f4`

IPv6

Addresses are typically written in hex, with colon separators, e.g., `fe80:0000:0000:0000:21c:c0ff:fea3:99f4`

A `::` may appear once as a shorthand for a string of `0`s. As many as you need to make the address up to 128 bits

Thus the above address can be written
`fe80::21c:c0ff:fea3:99f4`

Remember this is notation for:

```
1111111010000000 0000000000000000 0000000000000000
0000000000000000 0000001000011100 1100000011111111
1111111010100011 1001100111110100
```

IPv6

The University of Bath has been allocated
`2001:0630:00e1::/48`

IPv6

The University of Bath has been allocated `2001:0630:00e1::/48`

Meaning $128 - 48 = 80$ bits of address for hosts on the University network

IPv6

The University of Bath has been allocated
`2001:0630:00e1::/48`

Meaning $128 - 48 = 80$ bits of address for hosts on the University network

$2^{80} = 1.2 \times 10^{24}$ addresses, which is about 280 trillion times the size of the whole current IPv4 Internet!

IPv6

The University of Bath has been allocated `2001:0630:00e1::/48`

Meaning $128 - 48 = 80$ bits of address for hosts on the University network

$2^{80} = 1.2 \times 10^{24}$ addresses, which is about 280 trillion times the size of the whole current IPv4 Internet!

Exercise Check my arithmetic

Exercise Look up the IPv6 address of `facebook.com`

IPv6

Back to the other IPv6 header fields

IPv6

Back to the other IPv6 header fields

There are no fragmentation fields

IPv6

Back to the other IPv6 header fields

There are no fragmentation fields

A router never fragments IPv6, but simply drops the packet and sends back a “packet too big” message to the source. The source can then send smaller packets

IPv6

Back to the other IPv6 header fields

There are no fragmentation fields

A router never fragments IPv6, but simply drops the packet and sends back a “packet too big” message to the source. The source can then send smaller packets

Processing within a router is therefore much simpler and packets can be sent onwards much faster

IPv6

Back to the other IPv6 header fields

There are no fragmentation fields

A router never fragments IPv6, but simply drops the packet and sends back a “packet too big” message to the source. The source can then send smaller packets

Processing within a router is therefore much simpler and packets can be sent onwards much faster

Every IPv6 host is required to do path MTU discovery

IPv6

The flow label helps identify packets within a single “flow”, i.e., a connection or session

IPv6

The flow label helps identify packets within a single “flow”, i.e., a connection or session

Packets with the same flow label can be treated identically and so sent onwards faster by a router

IPv6

The flow label helps identify packets within a single “flow”, i.e., a connection or session

Packets with the same flow label can be treated identically and so sent onwards faster by a router

In essence a session identifier

IPv6

The flow label helps identify packets within a single “flow”, i.e., a connection or session

Packets with the same flow label can be treated identically and so sent onwards faster by a router

In essence a session identifier

Exercise Reflect on this: aren’t sessions supposed to be done in a different layer?

IPv6

No header length field: the header is always 40 bytes

IPv6

No header length field: the header is always 40 bytes

No fragmentation fields: no fragmentation in routers

IPv6

No header length field: the header is always 40 bytes

No fragmentation fields: no fragmentation in routers

No checksum field: there are checksums in other layers. The protocol designers thought that yet another checksum would not be helpful here. IP is not required to be reliable, anyway

IPv6

No header length field: the header is always 40 bytes

No fragmentation fields: no fragmentation in routers

No checksum field: there are checksums in other layers. The protocol designers thought that yet another checksum would not be helpful here. IP is not required to be reliable, anyway

Also we don’t have to recompute a checksum in every router as the TTL decreases. Again, faster in routers

IPv6

v4 has 13 fixed fields; v6 has 8; much simpler for a router to process

IPv6

v4 has 13 fixed fields; v6 has 8; much simpler for a router to process

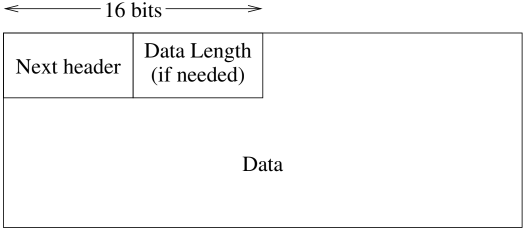
v6 addresses are 4 times the length, but the header is only twice as long

IPv6

The *next header* field daisy-chains options, called *extension headers*, or gives the protocol (TCP, UDP, etc.) of the next layer

IPv6

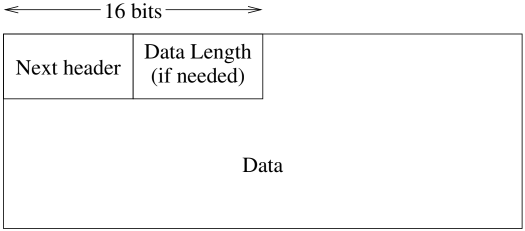
The *next header* field daisy-chains options, called *extension headers*, or gives the protocol (TCP, UDP, etc.) of the next layer



Option Header

IPv6

The *next header* field daisy-chains options, called *extension headers*, or gives the protocol (TCP, UDP, etc.) of the next layer

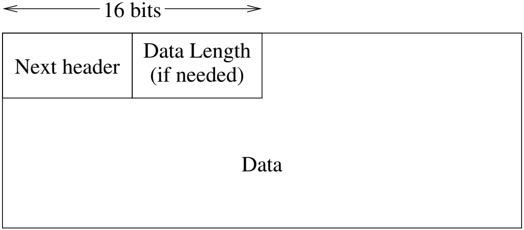


Option Header

Thus the only limit on the options is the total datagram limit

IPv6

The *next header* field daisy-chains options, called *extension headers*, or gives the protocol (TCP, UDP, etc.) of the next layer



Option Header

Thus the only limit on the options is the total datagram limit

Furthermore, most options are not even looked at by routers: again to get faster processing in the routers

IPv6

Optional headers include:

IPv6

Optional headers include:

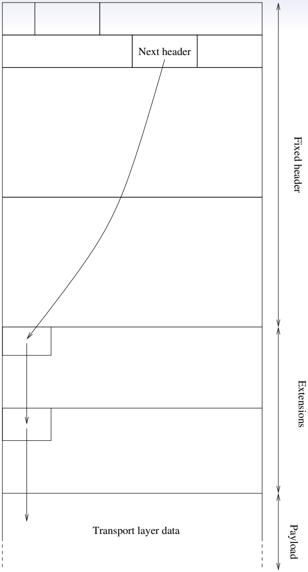
- Routing options: c.f., loose source routing in IPv4
- Authentication
- Security
- Jumbograms: packets up to 4GB in length!
- And others

IPv6

Optional headers include:

- Routing options: c.f., loose source routing in IPv4
- Authentication
- Security
- Jumbograms: packets up to 4GB in length!
- And others

Note the type of the header option is given in the *previous* header option, or the main IPv6 header for the first option



IPv6 options daisychain

IPv6 Jumbograms

A note on IPv6 jumbograms

IPv6 Jumbograms

A note on IPv6 jumbograms

It needs, of course, the hardware, link and transport layers to support large packets

IPv6 Jumbograms

A note on IPv6 jumbograms

It needs, of course, the hardware, link and transport layers to support large packets

For example, Infiniband supports up to 4k frames, while a lot of modern Ethernet hardware seems to support 9216 byte frames

IPv6 Jumbograms

A note on IPv6 jumbograms

It needs, of course, the hardware, link and transport layers to support large packets

For example, Infiniband supports up to 4k frames, while a lot of modern Ethernet hardware seems to support 9216 byte frames

We'll see later that UDP and the handshake TCP MSS have only 16 bit length fields (64k bytes), so tweaks are needed there, too (RFC2675)

IPv6 Jumbograms

Jumbograms can only be used locally, e.g., within datacentres, as the outside world almost certainly won’t support them!

IPv6 Jumbograms

Jumbograms can only be used locally, e.g., within datacentres, as the outside world almost certainly won’t support them!

Exercise Frame CRC algorithms were designed when frames were small. Read about the problems they have with jumbograms

IPv6

Transition to v6

IPv4 address allocations have run out, so we need to move to IPv6

IPv6

Transition to v6

IPv4 address allocations have run out, so we need to move to IPv6

But it is expensive to do so, as it needs application rewrites, as a lot of application software assumes IP addresses are 4 bytes long and thus fits in an integer on a typical computer

IPv6

Transition to v6

IPv4 address allocations have run out, so we need to move to IPv6

But it is expensive to do so, as it needs application rewrites, as a lot of application software assumes IP addresses are 4 bytes long and thus fits in an integer on a typical computer

So many people (ISPs, websites etc.) are pretending the exhaustion problem does not exist

IPv6

Transition to v6

IPv4 address allocations have run out, so we need to move to IPv6

But it is expensive to do so, as it needs application rewrites, as a lot of application software assumes IP addresses are 4 bytes long and thus fits in an integer on a typical computer

So many people (ISPs, websites etc.) are pretending the exhaustion problem does not exist

Even though the majority of modern routers and end hosts contain the necessary IP (and transport) level software support

IPv6

Transition to v6

We can't turn off the Internet and replace v4 by v6 overnight

IPv6

Transition to v6

We can’t turn off the Internet and replace v4 by v6 overnight

Though, by design, the two protocols can run side-by-side on the same networks

IPv6

Transition to v6

IPv6 was devised in 1996, but has yet to achieve mainstream use

IPv6

Transition to v6

IPv6 was devised in 1996, but has yet to achieve mainstream use

As of October 2023 figures from `ipv6-test.com` say they see about 64% of world traffic is IPv6

IPv6

Transition to v6

IPv6 was devised in 1996, but has yet to achieve mainstream use

As of October 2023 figures from `ipv6-test.com` say they see about 64% of world traffic is IPv6

About 56% of UK traffic is IPv6

IPv6

Transition to v6

IPv6 was devised in 1996, but has yet to achieve mainstream use

As of October 2023 figures from `ipv6-test.com` say they see about 64% of world traffic is IPv6

About 56% of UK traffic is IPv6

Uruguay (top): 94%

IPv6

Transition to v6

IPv6 was devised in 1996, but has yet to achieve mainstream use

As of October 2023 figures from `ipv6-test.com` say they see about 64% of world traffic is IPv6

About 56% of UK traffic is IPv6

Uruguay (top): 94%

Many countries are under 1%

IPv6

Transition to v6

Some large companies, e.g., Google, support IPv6 connections (as well as IPv4): they want to encourage the transition

IPv6

Transition to v6

Some large companies, e.g., Google, support IPv6 connections (as well as IPv4): they want to encourage the transition

But many ISPs don’t as it requires extra work and support: so many home users can’t use it

IPv6

Transition to v6

Some large companies, e.g., Google, support IPv6 connections (as well as IPv4): they want to encourage the transition

But many ISPs don’t as it requires extra work and support: so many home users can’t use it

There have been a variety of transition mechanisms suggested, often based on NAT-like packet mangling

IPv6

Transition to v6

Some large companies, e.g., Google, support IPv6 connections (as well as IPv4): they want to encourage the transition

But many ISPs don’t as it requires extra work and support: so many home users can’t use it

There have been a variety of transition mechanisms suggested, often based on NAT-like packet mangling

But they are all complicated and unsatisfactory, for the same reasons NAT is unsatisfactory

IPv6

Transition to v6

Exercise Read about NAT64 (RFC6146) and DNS64 (RFC6147) for connecting IPv6-only clients to IPv4 servers

Exercise Read about *IPv4 mapped addresses*, that allows server code that is purely IPv6, but accepts IPv4 client packets

Exercise Read about 464XLAT (RFC6877) for IPv4-only clients that translates IPv4 addresses to IPv6 addresses for transport and then back to IPv4 addresses for the destination server

IPv6

Transition to v6

In the near future IPv6 will need to be supported properly by everybody

IPv6

Transition to v6

In the near future IPv6 will need to be supported properly by everybody

Exercise Find out if your home ISP supports IPv6

IPv6

Transition to v6

In the near future IPv6 will need to be supported properly by everybody

Exercise Find out if your home ISP supports IPv6

Exercise RFC6177 suggests giving home users a /56 network. How many host addresses does this correspond to?