1

Introduction

Recently, Internet access has been revolutionized by mobile broadband. However, mobile Internet access is not a new technology – it has been available since the beginning of the 2000s, but only during the past last few years has the growth of mobile usage of the Internet exploded. This explosion is due to the increased data speeds that have brought mobile Internet access speeds close to those of fixed broadband access, and the prices dropping to affordable and competitive ranges. In addition, the exploding usage is due in very large part to the introduction of the smartphone.

At the same time, and partly as a result, the Internet is facing its biggest change and its biggest challenge since its introduction. This is the transition to the new version of the Internet Protocol (IP) – IP version 6. The old version – IP version 4 (IPv4) – has been in use since 1983 when the ARPANET transitioned from Network Control Program (NCP) to the Internet Protocol. Now, the exhaustion of readily available Internet Protocol version 4 (IPv4) addresses at the beginning of 2011 puts the growth of the whole Internet at risk.

The ongoing transition of the Internet to the new version of IP will, obviously, have implications for mobile networks as well. We have written this book to look at these important two topics together – mobile broadband access to the Internet, and the transition to Internet Protocol version 6 (IPv6). In Chapter 1, we start with an overview of the Internet technologies, and the background and implications of the transition to IPv6 to the Internet. Chapter 2 explains the basics of the Third Generation Partnership Project (3GPP) specified mobile broadband technologies, and Chapter 3 examines the IPv6 technology, giving a good understanding of how IPv6 works. Chapter 4 goes through how IPv6 is intended to work in the 3GPP mobile broadband networks. Chapter 5 concentrates on giving an understanding of different transition strategies that can be used in 3GPP networks. Chapter 6 gives a forward-looking view by the authors of some areas relevant to the future of IPv6 in 3GPP networks.

We wish the reader interesting reading moments, and we hope that this book provides help to the reader, whether a student, operator, network vendor, application developer, or handset manufacturer, to learn about and navigate through the IPv6 transition in the 3GPP network ecosystem.

Deploying IPv6 in 3GPP Networks: Evolving Mobile Broadband from 2G to LTE and Beyond, First Edition.
Jouni Korhonen, Teemu Savolainen and Jonne Soininen.

1.1 Introduction to Internet and the Internet Protocol

The Internet and the Internet Protocol creation were originally funded by the Defense Advanced Research Agency (DARPA) in the United States. Yet, today the Internet has become the global network of the whole world connecting all continents, virtually all of the countries, and already has significantly over two billion users. This path from a relatively small research project to the global information superhighway has been both fascinating and relatively quick. The DARPA project was started at the very end of the 1960s, the current version of the Internet Protocol was introduced in the early 1980s, and the first commercial Internet access providers came online in the end of 1980s or early 1990s depending on country and region. As late as 2006, one of the main topics of the Internet Governance Forum (IGF) – a United Nations (UN) organization discussing matters that concern the governance of the Internet, both technical and non-technical – was to connect the unconnected, that is how to get Internet access to the developing countries. Since that day, most of the developing countries have at least Internet access in the bigger cities, usually through mobile networks. The Internet has very quickly encompassed our lives, regardless where we live.

This chapter concentrates on explaining what are the guiding principles that led and enabled this evolution, and to describe what is the Internet's most important building block – the Internet Protocol. For the interested reader, at the end of the chapter there are additional reading materials for more information about the fascinating history of the Internet.

1.2 Internet Principles

Today the Internet is used for file transfer, email, voice, video, gaming, and many, many other applications. We have become dependent on the Internet starting from, the world economy, via businesses big and small, to the normal people who trust the Internet either to keep them connected to the artery of the economic world, or to keep up relationships with their loved ones. It is surprising how big and important the Internet has become in such a short time. However, the versatility of the Internet is not accidental. The reasons lie in the design of the Internet.

In the heart of the design of the Internet and the technology that powers it has certain principles. These principles have ensured that the Internet and the Internet technologies enable the current usage of the Internet, which is way beyond the usage and expectations envisioned by anybody at the dawn of the Internet. Let's look at the main principles:

- packet switched networking;
- the end-to-end principle;
- layered architecture;
- Postel's robustness principle; and
- creative anarchy.

Packet Switched Networking

This first principle is very widely used in modern communication networks. However, traditionally, the voice centric networks, such as the Public Switched Telephony Network

(PSTN), were based on a different technological principle – circuit switched networking. In circuit switched networking connections or calls are switched through the network by reserving a circuit from the caller to the callee. Each connection has its own circuit, and the same resource is reserved for the call regardless how much traffic is transferred through that circuit. For instance, a voice call uses exactly the same resources within the network whether the participants speak or are silent.

In contrast, the modern data networks are built based on Packet Switched (PS) networking. In packet switched networking, the sent data is divided into smaller packets that travel independently to their destinations, each transmitted through a route that seems best for a given packet at a given moment. In the Internet, the packets can travel different routes even to the same destination, can get out of order at transit, and even get completely lost, never arriving to their destination.

End-to-End Principle

The end-to-end principle is one of the most important principles of the Internet and of the Internet technology. It states that the network should not interfere with or alter traffic on layers that are above the network layer. Sometimes this principle is also known as *'Intelligent endpoints – dump network'* principle, but that name does not do justice to the concept. Basically, what the principle states is that the network should not make any assumptions about any particular service or characteristic of the data in transit. The network must concentrate on moving the data from its source to its destination. It is up to the end points to understand the traffic and its use. This principle allows the Internet to be used for many different services and applications. Even applications and services can be supported that were at the time of the design of the Internet either technically unfeasible, completely impossible, or even unimaginable. Thus, the principle enables us to create new services without changing the network. Only the end points have to be changed to support the new service or application.

Layered Architecture

The layered architecture principle is closely linked to the end-to-end principle. The layered architecture principle states that there are different protocol layers that talk to each other on the same level. Figure 1.1 shows the principle in a drawing. The main idea behind the principle is that each layer does its work – no more and no less. This strict separation allows the layers to be independent of each other, and make sure that layers can be changed without changing other layers. The Open System Interconnect (OSI) [1] model defines seven layers. The Internet model, however, only defines five. The Internet technologies range from the layer three (network layer) upwards.

Robustness Principle

The Postel's robustness principle, also known as Postel's law, has gotten its name from its inventor – Jon Postel. Jon Postel was one of the Internet pioneers who has had an enormous effect on the Internet and its design through his contribution in engineering and governance. The principle is quoted as *'Be liberal in what you accept, and conservative*

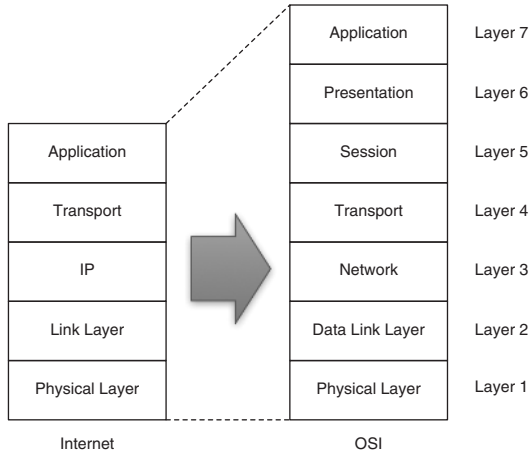

Figure 1.1 Networking layers.

in what you send.' [2] This principle means that the sender should be very conservative and strictly cohere with the standard, whereas the receiver should be able to receive even input that does not strictly conform to the standard. This is a very important principle for interpretability, and when making future extensions and improvements to the technologies.

Creative Anarchy

The final principle, creative anarchy, is something that seems to be at odds with proper telecommunications networking. However, this means that anyone (from a single individual to large enterprises or operators) can create and distribute new services and applications. This principle can be perhaps seen more as a symptom of the other principles, but comprehending it is still important to understand the Internet and its technologies. Without the creative anarchy principle, the Googles, Facebooks, and the like of this world would never have seen the light of day!

1.3 The Internet Protocol

The most important building block in the Internet is the Internet Protocol. IP is the networking protocol that ensures that packets travel from the source through the network and reach the right destination. Currently two versions of the Internet Protocol exist: IPv4 [3] and IPv6 [4]. At the time of writing, IPv4 is much more widely spread and used. IPv6 is a newer, upcoming protocol version which the world is transitioning towards. As you may have realised from this book's title and the number of pages we have been able to write, this transition is not quite straightforward. However, despite a few differences, the functionality of the two IP versions is very similar and in parts, the same. Therefore, the overview that this chapter gives to IP is applicable to the both versions.

The Internet Protocol has two parts: the header and the payload. The header consists of fixed length binary fields that show, among other information, where the packet has

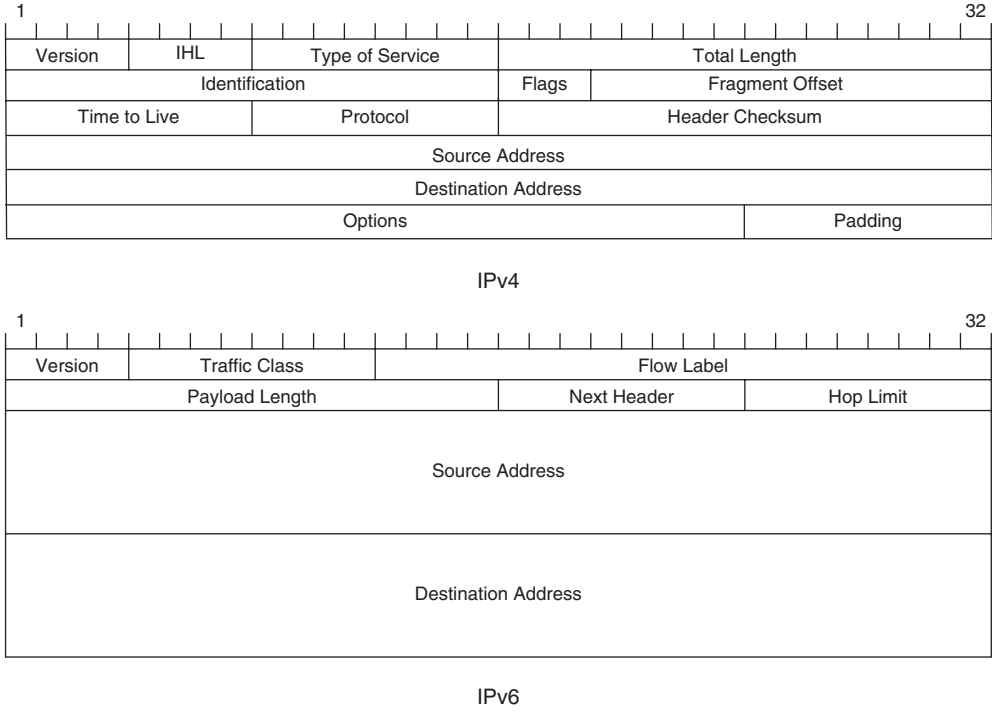

Figure 1.2 IPv4 header [3] and IPv6 header [4].

come from (source address), where it is going to (destination address), and what is in the payload – or better put, what is the upper layer protocol. The payload can be the transport layer protocol, and the protocols and data above it, or even an IP packet. The IPv4 and IPv6 headers are depicted in Figure 1.2.

IP provides an unreliable packet delivery service. Basically the main part of the service is addressing – the capability to tell where a packet should be delivered. In addition to the packet delivery, IP also performs other tasks. For instance it includes a Time To Live (TTL) field, which provides the mechanism to make sure that packets are removed from the network if they do not reach their destination due to, for instance, a loop in the network. Practically, this is achieved by decrementing the TTL at every hop in the network. When the TTL reaches zero the packet is discarded. There is also a simple per packet prioritization mechanism (Type of Service or Quality of Service). Everything needed for the communication, such as reliable transport, packet ordering, and identification of applications, is provided by upper layer protocols such as Transport Control Protocol (TCP) [5].

The nodes of an IP network are connected to the network via a network connection or an interface. A node can have one or more interfaces connected at a given time to the same, or different networks. Each interface has to have a unique IP address from the network to which they are connected. Network interfaces can be either physical or virtual. Physical interfaces are usually created by hardware based network technology support in the node. Physical network interfaces can be wired, wireless, or even inter- and intra-microchip

connections within one physical device or even inside one microchip. Virtual interfaces are created usually by tunneling interfaces. Tunneling means transporting IP packets inside other IP packets. Practically the outer IP flow creates a 'tunnel' – a virtual link over a network. The inner IP packets then are transported via this tunnel to another network. The virtual interface is connected to this remote network, and the IP address comes from that network. This approach is used for instance in Virtual Private Networks (VPN) (see Section 3.8) or Mobile IP (see Section 3.7.2) technologies. In addition, one special kind of virtual interface is the loopback interface. This is a node internal interface that loops back, that is, sends back, all packets sent to it. The loopback interface is used for node internal communication.

Generally, the IP network nodes are divided into two categories: hosts and routers. Hosts are nodes that send and receive IP packets, but do not pass on packets for other nodes. A router is a device that connects two different networks, and forwards packets between them. Though this distinction is relatively straightforward, nodes can sometimes have both roles. A good example is the home routers that connect a home network to a wide area network. These usually function as routers, but also often implement, for instance, a web server for management purposes. As a web server the device is a host, but as a home gateway it is a router. It is important to note that IP does not have any special distinction between clients and servers. From IP's point of view, they are all hosts.

In addition to routers, hosts can also have multiple network interfaces connected to multiple IP networks at the same time. Hosts with multiple interfaces, and multiple IP addresses, are called multi-interfaced hosts. Logically, a host with just one interface is called a single-interfaced host or just a host. It is actually quite usual for hosts, such as personal computers, to be multi-interfaced. For instance, today it is normal for a laptop to be connected to a wireless and a wireline networks at the same time. In addition, when a host has a virtual interface, such as a VPN tunnel, it is always multi-interfaced: it has at least the one physical interface and the virtual interface.

The IP nodes are connected to each other by a network of lower layer network connections called links. These links can be either point-to-point links connecting two nodes, or complete layer 2 networks shared between multiple nodes. Different link-layers are used in modern networks. By far the most commonly used link-layer technology is Ethernet [6]. As Ethernet is a very common technology in IP, some parts of the IP technology expect Ethernet-like functionality as well as from other link-layer technologies, and even some non-Ethernet technologies simulate Ethernet for easier integration to their operating systems' IP implementations (which is not without issues, as we will see in Section 4.9.1).

1.3.1 Networks of Networks

When reading about IP, and the biggest IP network of all – the Internet – sometimes the usage of the word 'network' is very confusing. The IP networks come big and small. Some of them are connected to the Internet, and some of them are not. An example of the smallest possible IP network is two nodes directly connected to each other exchanging data. An example of such a small network may be just a single peripheral connected to a computer. As IP is a well standardized and supported networking protocol, sometimes

even these simple connections are more favorable to do with IP rather than using an application specific protocol.

The Internet is said to be a network of networks. This description actually fits smaller IP networks, as well. Usually a section of a network, which is under one administrative control, is called a network. This can range from a peripheral connected to a single computer under the 'administrative control' of the computer's user, through a small home network, to large corporate or operator networks with thousands of nodes. These administrative domains can either be isolated or interconnected to other networks under a different administrative control. A good example is a home network; this has different nodes such as network attached storage, game consoles, different computers, etc. These create a network and communicate to each other sharing music, pictures, and other information. This home network may then be interconnected to an operator network such as an Internet Service Provider (ISP). The ISP is then connected to other Internet Service Providers. All these different ISPs (and the small home networks, and bigger enterprise networks) create the Internet.

IP itself does not have separate User-to-Network Interface (UNI) and Network-to-Network Interface (NNI) protocols, but IP handles the networking between the end-hosts, the network nodes, and the different networks in the same way. Hence, IP creates networks of networks.

1.3.2 Routing and Forwarding

The hosts in an IP network communicate between each other by sending IP packets. The source host has to know the IP address of the destination where it wants to send the traffic. The two hosts can be either directly connected to each other, or there can be one or more routers in the path between them. Directly connected hosts, such as hosts in the same Local Area Network (LAN), may be able to send packets directly between each other. Alternatively, a host can send packets to a router for further delivery.

Routers pass packets between different networks and different routers, as described in the previous section. Theoretically, this packet passing can be divided into two different processes: routing and forwarding. Routing involves selecting the route where the packet should be going next. The idea of routing is to find the shortest or fastest route to a destination, and then selecting the outgoing interface based on that information. Forwarding, on the other hand, is the process that actually moves the packet through the router to the intended direction. When a packet is forwarded by a router, the router decreases the TTL by one. If the TTL reaches zero, the packet is discarded.

Routing decisions are based on routes. Routes define which networks or destinations are reachable through which interface. In addition to routes, the routing decisions can be affected by routing policy. Routing policy is basically a rule that overrides the technical routing calculation. There are various reasons to set a routing policy. For instance one reason is cost – sending a packet through one operator may cheaper than a possibly shorter path through another operator. Routes can be either statically configured, or dynamically calculated. Statically configured routes are usually configured by a network administrator. These routes do not change until the network administrator changes the configuration, regardless of the network status. For dynamic route calculation, routers use dynamic

routing protocols to exchange network topology information between each other. The most used routing protocols are Open Shortest Path First (OSPF) [7, 8], Intermediate System to Intermediate System (IS-IS) [9], and Border Gateway Protocol (BGP) [10]. Sometimes routers do not have the complete picture of the whole network, such as the whole Internet. A router might not know which of its interfaces would bring the packet to the right direction. Therefore, the router may have a default route. The default route basically states that if there is no more specific information – a specific route that fits the packet's destination address – by default the packet should be sent through a specific interface to a specific router.

Directly connected routers exchange information about the routes they have gotten from other routers to which they are connected. A router uses this information to create a view of the network topology. This network topology map is stored in an internal data structure called a routing table. When some new information about new routes or old routes disappearing comes to the router either by the routing protocol updates, or by the router noticing a neighboring router disappearing, the router updates the internal routing table. It also informs the other neighboring routers about the changes it has noticed.

Route information changes when new networks are connected or old networks are disconnected from the IP network. New networks appear when a network administrator somewhere installs a new router on the network, perhaps exposing a complete network behind that router. Networks can be detached either administratively when a network administrator takes a router away from the network, or because of a link failure – perhaps detaching a whole network behind it. To reduce the risk of a complete network being detached, networks are sometimes connected via multiple links and even through multiple routers. A good example is when an enterprise buys Internet services from two ISPs to protect against a case where one fails for some reason or other.

When an IP packet arrives at a router, generally the router looks at the destination address of the packet, and forwards the packet to one of its network interfaces. The forwarding decision is usually done by looking up the destination address from another data structure called a forwarding table. The forwarding table is created by the router using the routing table and possible routing policies existing in the router. The router then creates a table where it lists which networks are accessible through which of the router's network interfaces. An IP router looks at every IP packet individually and does the forwarding decision on a packet-by-packet basis. If something changes in the router's routing table, a packet can take a different route from a preceding packet even if it has the same destination.

In addition to routers, hosts make forwarding decisions. For instance, a host has to decide if it thinks the destination host is directly connected to it, or if it has to forward the packet to a router instead. This first router is often called the first-hop router. There may be multiple routers available in a network of which the host can choose. One of these routers is usually assigned as the default router for the host. If the host has no more specific information about the location of the destination host, the host sends the packet to the default router. Hence, the host's default route points to that default router. In the same way as routers may be configured with routing policy, the host might have a set routing or forwarding policy. The reasons for the policies include, security, some assumption about the cost or characteristics of a network connection, or just user

preference. A corporate IT-department might configure the user's traffic to always go through an active VPN tunnel regardless of where the packet is going, in order to make sure that company secrets do not traverse through unknown networks. Some operating systems prefer a wireline connection over wireless because it is assumed to have more bandwidth and be more reliable, and some operating systems allow the user to define the preference order of the available interfaces.

1.4 Internet Protocol Addresses

Regardless of whether a network is big or small, the IP addresses have to be unique within the network. As described above, the IP packet's destination address is used to route the packet through the network to its final destination. If the addresses were not unique, there would be no way of knowing where a packet should be delivered to. In a simple network, which is not connected to the Internet, it is enough that the addresses are unique locally within that domain. On the Internet, the IP addresses have to be globally unique.

The IP address itself is a fixed length binary identifier. An IPv4 address is 32 bits, and an IPv6 address is 128 bits long. The IP address has two functions in an IP network: it uniquely identifies an interface or a node in the network, and it represents a location in the network topology.

The identity is related to the uniqueness of the IP addresses – no other node in the network should have the same address as the node that is identified by a certain address. That address uniquely identifies a single network interface of that node and no other.

The location means that the address uniquely identifies a network where the node is within the IP network. The IP address can be divided into two parts; the network prefix and the host identifier, which together make the complete IP address. In the following pages, we will describe in more detail what this means in practice for IPv4 and IPv6.

In this section we will have a more in-depth look at the IP addresses. First we will explain the IPv4 addresses, and then IPv6. The focus is on the addresses themselves: what are addresses, how do we represent them, and what kind of different address types exist?

1.4.1 IPv4 Addresses

As describe above, an IP address is a fixed length binary field. In IPv4, the IP address field is 32 bits long, thus giving a theoretical address space of $2^{32} = 4$ billion addresses. Because a binary bit sequence would be a bit difficult to remember, represent, and tell to a friend, a text representation is used to denote the address information. In IPv4, the 32 bits are broken into four 8-bit (one-byte) fields each separated by a dot. These eight bit fields are presented in decimal numbers, for instance, `198.51.123.234`.

The network prefix length is variable in IPv4. The size of the prefix is denoted by putting a slash, '/', and the length of the prefix in bits to the end of an address. For example, `192.51.100.0/24` shows that the network prefix size is 24 bits. Blocks of addresses are presented in the same way, according to the prefix length. For example, a block of addresses from `198.51.100.0` to `198.51.100.255` is usually written as `198.51.100.0/24`, or `198.51.100/24` (leaving the zero out at the end). This means that the prefix length is 24 bits, and the host part is the remaining 8 bits out of

Table 1.1 IPv4 address classes

Address Class	Prefix Length	Number of Addresses
Class A	/8	16,777,216
Class B	/16	65,536
Class C	/24	256

the full 32 bits. When a prefix has more bits, it is considered longer, and a prefix with fewer bits is considered to be shorter. The longer the prefix, the shorter the host part, and hence, fewer addresses are available for the hosts in that network.

Originally, the IPv4 address space was divided by the length of the network prefix into different classes. The class A address blocks were blocks of /8 – 16,777,216 addresses, class B address blocks were /16 – 65,536, and class C blocks of /24 – 256 addresses. Table 1.1 shows the address classes and their lengths. In the 1990s, the Internet moved to classless addressing – Classless Inter-Domain Routing (CIDR) [36, 37] – and the network prefix length is no longer dependent on the address class; the length can be different from the original address classes.

The addresses used for normal unicast communication between hosts are called unicast addresses. Quite obviously, the main part of the allocated address space is dedicated to unicast addresses. These addresses are sometimes also called public addresses or globally routable addresses. In addition, other address types also exist. The specification describing special use IPv4 addresses [11] lists 15 different special address blocks. We will here go through the most important ones.

As will be seen below, the IPv4 addresses are mainly used for the unicast communication between network nodes. In addition, however, there are special addresses with special meanings. Most of the IPv4 address space's 4 billion addresses are used for normal unicast addresses for the communication over the Internet, but there is quite a bit of address space that cannot be used for that purpose. This restricts the number of addresses in the Internet even further. We will discuss how IPv4 address scarcity is an issue later on.

Private Use Networks

`10/8`, `172.16/12` and `192.168/16` are reserved for private use networks [12]. Commonly this part of the address space is called private address space, and the addresses private addresses. These addresses can be used only within a private network. They cannot be used directly in the Internet, because the addresses given from a private address block are not unique over the whole Internet.

Shared Address Space

`100.64/10` is reserved for shared address space [13]. Shared address space is similar to private addresses introduced above, but instead of general use, the shared address space is intended to be used within an operator network.

Loopback

`127.0.0.0/8` is reserved to loopback addresses. These addresses are exclusively used for host internal communication. Packets with loopback addresses should not be seen on any network.

Link-local Addresses

`169.254/16` address block is used for communication constrained to one link. These addresses are especially used before a host has gotten a real address, private or public. In some cases, such as with directly connected hosts, these may be the only addresses ever available on a link RFC 3927 [44].

Multicast

`224.0.0.0/4` block is used for multicast services. How the multicast address space is used is specified in RFC 5771 [14].

Reserved for Future Use

The block `240.0.0.0/4` is marked as reserved for future use. There has been much controversy about the usage of this address block. It seems that some routers think it is an error if they see a packet from this address space. Therefore, the use of the address space is relatively difficult – at least at the level of the Internet. It is clear that there has been a lot of pressure and interest in doing something with this block as it is about 6 per cent of the overall address space. However, at the time of writing, there has been no use found for it.

Limited Broadcast Address

The address `255.255.255.255/32` is a special address for link-local broadcast. Packets with this destination address are not forwarded on the IP layer, but all the nodes within the link-scope get the packet. Some applications use this address as their destination for initial boot strapping.

1.4.2 IPv6 Addresses

The IPv6 address field is 128 bits long, and hence gives much bigger address space than IPv4. The theoretical maximum of the IPv6 address space is $2^{128} = 3.4 \times 10^{38}$. A number this big is rather difficult to understand, and people have tried to explain the number in various ways. One of the examples is that there are 6.5×10^{23} addresses per every square meter of the earth. It is not clear why anybody would want to allocate addresses per square meter, but the message is clear: the IPv6 address space is very, very big.

To distinguish IPv6 addresses from IPv4 addresses, and to have an address format that is at least vaguely readable, IPv6 has its own textual format. The textual format has been defined in RFC 4291 [15] and RFC 5952 [16]. The textual

representation of an IPv6 address consists of eight 16-bit hexadecimal fields separated by colons – `x:x:x:x:x:x:x:x`. As the IPv6 address format is quite long, and rather complex, so are the rules on how to represent the addresses.

Below are examples of legitimate address representations:

```
2001:41d0:1:7827::1
2001:db8::a:0:0
2001:db8::a
::1
```

The 16 bit value of each `x` is presented without leading zeros. Furthermore, when the address has 32 or more consecutive zero bits, the zero bit sequence can be abbreviated with syntax `::`. The `::` can be used only once in an address. If an address has multiple series of zeros, the `::` must be used where it makes the most difference. Here is an example of compressing zeros away from an IPv6 address: `2001:0db8:0000:0000:0000:0000:0000:000a` can be presented in much more readable form as `2001:db8::a`

A special textual addressing format has also been defined for IPv6 addresses that are carrying IPv4 address in the lowest 32 bits. The address format for such addresses is `x:x:x:x:x:x:d.d.d.d`, where `d`s represent an IPv4 address [15]. An example of such embedded address is: `64:ff9b::192.0.2.1` [17].

When an IPv6 address needs to be shown with a port number, square brackets, `[` and `]`, should be used to make the address unambiguous. Without brackets it might be difficult to determine whether the last digit is a port number or part of an IPv6 address, as illustrated here: `2001:db8::a:80`. A correct example, using brackets, makes the distinction clear: `[2001:db8::a]:80`. Many applications, such as web browsers, expect IPv6 addresses to be put in square brackets.

As in IPv4, so in IPv6, the address is divided into the network prefix and the host part. In IPv6, the host part is called the Interface IDentifier (IID). The IID for the unicast addresses is defined to be 64 bits long [15]. The IID must at least be unique in the scope of the link where the interface (and hence, the node) is located. However, the IID can also be unique in a larger scope as well. The network prefix length is noted in the same way in IPv6 as previously described for IPv4. For instance, `2001:41d0:1:7827::/64` means that the network prefix is 64 bits long, and in `2001:41d0:1::/48` the prefix length is 48 bits.

Consequently, the length of the IID to 64 bits also constrains the network prefix length at a maximum of 64 bits. Hence, allocating at least a network prefix of /64 for a single link is the norm, regardless of how many nodes are in that subnet. In IPv6, it is important not to count single addresses anymore, but count prefixes of /64 per link. This is quite a different philosophy from IPv4, and may be unintuitive for many. While in IPv4 the address policy was driven by conservation, in IPv6 the driver is network and numbering simplicity, with additional security considerations, as we will discuss in Chapter 3.

1.5 Transport Protocols

As described previously, the IP provides unreliable packet delivery over the IP network, such as the Internet. The only multiplexing between two end points that it provides is the

distinction between the different protocols it is carrying. Thus, anything else has to be provided by the upper-layer protocols. These protocols are called transport protocols, as their job is to transport the actual application payload end-to-end. The two most important transport protocols used today are User Datagram Protocol (UDP) [18] and TCP [5]. Generally, the transport protocols are IP-address family agnostic. Thus, the same protocols can be used with IPv4 and with IPv6. Of course, in practice transport protocols themselves need to be able to work with different network layers, as we shall see in Chapter 3.

We will try to give a short overview of these transport protocols in this section.

1.5.1 User Datagram Protocol

UDP is a very simple protocol. It only provides two services: service and connection multiplexing, and a checksum for the receiving end point to check if bit-errors have been introduced during transport. The multiplexing is done by port number fields in the UDP header – the source and the destination ports. The quintuple – source address, destination address, and protocol number in the IP header – and the source and destination port numbers in the UDP header uniquely identify the connection for the end-host.

UDP does not guarantee that the packets are transported in order, or actually even that they are received by the other end at all. Thus, there is no reliability mechanism. Therefore, UDP is usually used for transporting application data, where occasional loss of packets is not fatal, and the transporting of the packets as fast as possible is more important than reliability. These are applications where it is better to forget the packet rather than transport it late. These applications include for instance Voice over IP (VoIP). If an application needs reliable packet delivery, it should either implement the reliability mechanism itself, or use TCP.

1.5.2 Transmission Control Protocol

TCP is one of the most important protocols in the IP protocol family. As a matter of fact, the IP protocol family is very often called TCP/IP. TCP provides reliable, ordered transport, with service and connection multiplexing, and with congestion control. Most Internet applications require reliable transport. Therefore, for example, file transfers, video steaming, and especially web traffic are transported over TCP. Hence, most applications use TCP for communicating over the Internet.

As in UDP, the application and service multiplexing is achieved by source and destination port numbers. The reliability is achieved by the receiving host acknowledging the packets it receives, and through this the sending host notices which packets have gone missing. These packets are then retransmitted. The TCP congestion control algorithms monitor this packet loss. TCP actually assumes packet loss to be always caused by congestion, and TCP will drop its transmission rate to adjust to this perceived congestion. Due to this behavior, TCP has received a bad reputation as an unfriendly protocol for wireless networks, because in wireless networks packet loss can be caused for many reasons other than just congestion. However, in recent years the new wireless access technologies are more TCP friendly, and new TCP extensions have taken wireless networks much more into account. Hence, TCP's bad reputation is mostly outdated.

TCP is relatively complex. Therefore, we will not try to describe it more than necessary in the book. However, an interested reader will find good suggestions for further reading in the additional reading section at the end of this chapter.

1.5.3 Port Numbers and Services

The port numbers perform an important function in the IP networks, and hence merit their own section. The port number field in UDP and in TCP is 16 bits long and therefore can carry numbers 0–65535. This port range is divided into different usages. The range of 0–1023 is called well known ports, or system ports, ports 1024–49151 are called registered ports or user ports, and the remaining 49152–65535 are called private, dynamic, or ephemeral ports [19].

The well known or registered ports generally identify a service, an application, or a usage. This means that a specific application or service is listening to that port, and is understood to be that service. For example, port 21 is the control part of the File Transfer Protocol (FTP) [20], port 23 is Telnet [21], and port 80 is HyperText Transfer Protocol (HTTP) [22, 23]. Therefore, a service listening and answering at port 80, for instance, is expected to be a web server.

1.6 Domain Name Service

IP addresses are used by computers – the hosts and routers that communicate to each other. However, the IP addresses are not the most intuitive identifier to be remembered by people. People remember names better than numbers. To address this, the Domain Name System (DNS) was developed RFC 1034 and RFC 1035 [43, 24]. DNS allows human-readable names to be used for protocols and user interfaces, though allowing the Internet still to function with numbers with better computational characteristics. Basically, DNS is a distributed database that provides name-to-address, and address-to-name mapping. So DNS is a bit like a big telephone book of the Internet.

1.6.1 DNS Structure

The DNS is a global database covering the whole Internet. Therefore, scalability is very important. Hence, DNS was designed to be a distributed, hierarchical database. Figure 1.3 shows the DNS structure. This structure is seen also in the DNS names – if we examine for instance the name '`www.example.com`.'. A complete DNS name like this is called a Fully Qualified Domain Name (FQDN). It consists of the following parts, which can be found in Figure 1.3.

root is noted by the final dot ('.') at the end of an (FQDN). Root is the top of the hierarchy, and is the central point of the (DNS) database. For the user, this final dot is usually hidden, and the user usually sees '`www.example.com`' rather than '`www.example.com`.'.

Top Level Domain (TLD) is, as the name suggests, the highest point of the domain names. They are, for example, the `.com`, `.org`, and `.de` at the end of a domain name. There

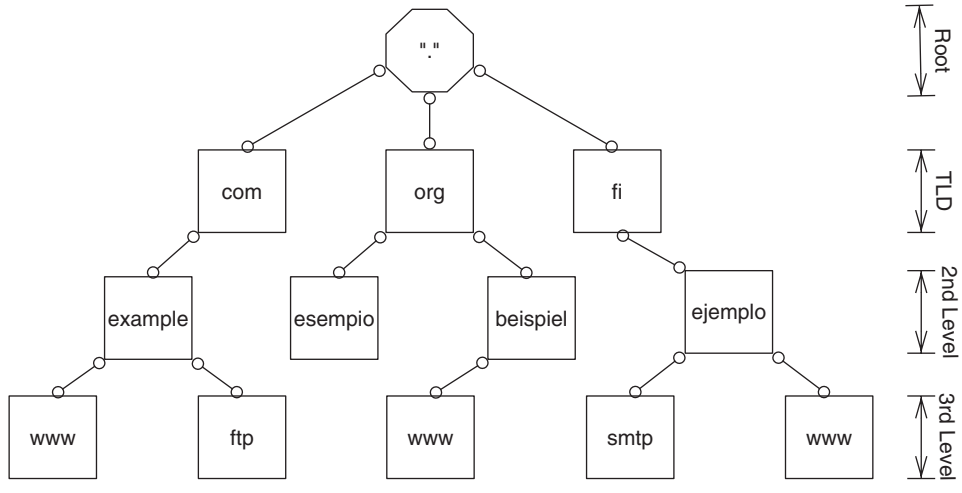

Figure 1.3 DNS structure.

are three categories of TLDs – generic Top Level Domains (gTLDs), country code Top Level Domains (ccTLDs), and an infrastructure TLD. TLDs are more closely examined in Section 1.6.3. An organization, which operates and owns a TLD is called a registrar. Registrars sell or distribute 2nd level domains to the organizations that need domain names.

2nd level domains are owned and controlled by registrars. Registrars include companies, organizations, and even private persons. A 2nd level domain owner can create 3rd level domain names, and distribute those in the way the 2nd level domain owner wishes.

3rd or lower level domain are domains created under the 2nd level. They can indicate a service (such as `www`, `ftp`, etc.) or they can be otherwise descriptive – for instance `cs.helsinki.fi` is the domain name of the computer science department of the University of Helsinki.

1.6.2 DNS Operation

The previous section explained the structure of the DNS name. This section will now examine how the actual DNS resolution – mapping a DNS name to an IP address – works actually. Figure 1.4 shows a simplified overview of the process, and in the following we examine the resolution, step by step:

1. The process starts when a host has an FQDN that it wishes to convert to an IP address. The host's DNS resolver sends a DNS query to a Recursive DNS Server (RDNSS) that has been configured in the host's DNS configuration (e.g., by means described in Section 3.10.2).
2. The DNS query for `www.example.com` is received by a RDNSS, which has to first find what server is responsible for the `.com` TLD. The RDNSS knows the root server

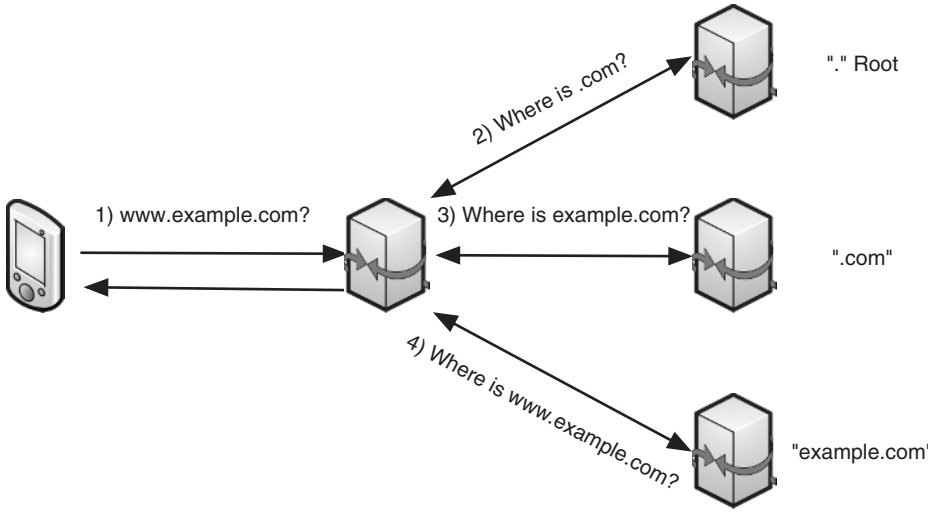

Figure 1.4 DNS name resolution.

addresses, and it sends the query to one of the root servers. The root server refers to the server responsible for `.com`.
3. The RDNSS sends the query to the authoritative server responsible for `.com`. This server responds by referring to the server responsible for `example.com`.
4. The RDNSS sends the query to DNS server authoritative for `example.com`. The DNS server answers with the DNS record that contains the IP address corresponding to `www.example.com`.
5. The RDNSS sends the record containing the IP address of `www.example.com` to the host.

The IP addresses are stored in DNS records. The IPv4 records are stored in A [24] records, and the IPv6 addresses in `AAAA` records [25] (also called quad-A records). Please see more detailed implications of IPv6 to DNS in Sections 3.9.4 and 3.10.2.

1.6.3 Top Level Domain

There are three categories of TLDs – gTLD, ccTLD, and infrastructure TLD. The Internet Assigned Number (IANA) Root Zone Database [26] lists the current TLDs. In the following, the different TLD categories are explained.

gTLD is a generic name TLD, which indicates something generic about the organizations or content on the next level. For instance, `.com` indicates commercial, `.org` organization (usually non-for-profit), `.net` network, and `.mobi` mobile communication. These indications are, however, not very strict in reality. A gTLD is always three or more latin characters long. The gTLDs are usually owned and operated by private organizations, which sell or distribute the second level names to other organizations somehow

connected to that category. The gTLDs are administered by Internet Corporation for Assigned Names and Numbers (ICANN). At the time of writing, 21 unique gTLDs exist. However, ICANN has an ongoing new gTLD process in which new gTLDs are evaluated for approval. Almost two thousand unique applications were received by ICANN during the application window [27].

ccTLDs indicate a country, or a territory as defined in the ISO-3166 [28], and the ccTLDs consist of two-character strings defined in the same standard. The ccTLDs are usually operated and owned by the government of the country or territory, a government agency, or an organization appointed by the local government. (There are two-character TLDs that are used like gTLDs and might even be owned by companies. For instance, these include `.tv`. However, these are originally ccTLDs that have either been sold, or are just used in gTLD manner.)

Infrastructure (TLD) is the `.arpa` TLD. IANA administers arpa for the Internet Engineering Task Force (IETF) for technical purposes as a part of the Internet's infrastructure. The `.arpa` TLD is used, for example, for DNS reverse queries RFC 3172 [45] – mapping IP addresses to DNS names.

1.6.4 Internationalized Domain Names

Originally, the DNS supported only latin characters. This, obviously, had its origins in the birth of Internet in the USA, and in the challenges with computer systems of taking as input, and reproducing other character sets. However, modern computing can support many more scripts. In addition, the Internet has become a global phenomenon where most of its users are not native English speakers, or even use the latin script. Therefore, the IETF has specified the support for Internationalized Domain Names (IDNs) [29–32].

The IDNs started off in second level domains under different TLDs. In addition, the ICANN ccTLD Fast Track process enabled certain ccTLD IDN variants to be accepted to the root. It is expected that multiple IDN gTLDs will be introduced by the new gTLD process.

1.7 IPv4 Address Exhaustion

As explained earlier, the IPv4's 32-bit address field sets a theoretical maximum number of addresses to $2^{32} = 4{,}294{,}967{,}296$ addresses. To understand the address space usage more clearly, we can consider the IPv4 address space to consist of 256 address blocks of the size of /8. As described earlier, we have address space that cannot be used for normal communication purposes in the internet. Including all the different special use address spaces in RFC 5735 [11], we have 35.078 /8s (14%) that cannot be used on the Internet. This leaves 220.922 /8s, which can be used for Internet nodes – a range of 3,706,456,113 addresses – just under four billion.

As all nodes that are connected to the Internet need their own unique address, just under four billion individual nodes can be at the same time on the Internet. These nodes include the routers, servers, end-hosts – basically all nodes connected to the Internet, which have to be reachable from the Internet. At the time writing, the world population is around 7 billion, and there are already over two billion Internet users, and the number is growing

fast. Introduction of the Internet of Things (see Section 6.5) creates significant additional pressure on the address space consumption. It is clear that the IPv4 address space is a seriously constrained resource. To clearly understand the IP address allocation problem, and the IPv4 address space exhaustion, we will now look at how IP addresses are allocated to the end users, what is the history of IPv4 address space exhaustion, and what has been done to mitigate the depletion of the IPv4 address space. Finally, we will look at the situation at the time of writing this book.

1.7.1 IP Address Allocation

The IP address allocation is a hierarchical system. Figure 1.5 shows how the IP address allocation hierarchy is set up. This is one representation of the hierarchy. You can certainly find other descriptions where the boxes are in a different order. The IP address allocation should be more of a technical, mechanical function. However, in the past few years the different interests around IP address allocation have politicized the process, and even the setup has been questioned. However, in our view, this picture makes sense when looking at the Internet from a technical perspective.

The highest level of the hierarchy is the IETF. The IETF is the standardization organization, which is responsible for specifying the IP technology – including the Internet Protocol itself. The IETF specifications specify how long is the address, what is the addressing architecture, and what are the special address ranges that are not allocated as unicast addresses.

The global allocation of IP unicast addresses is the responsibility of the IANA [33], which is operated by ICANN [34]. In IPv4, IANA allocated /8 address blocks for

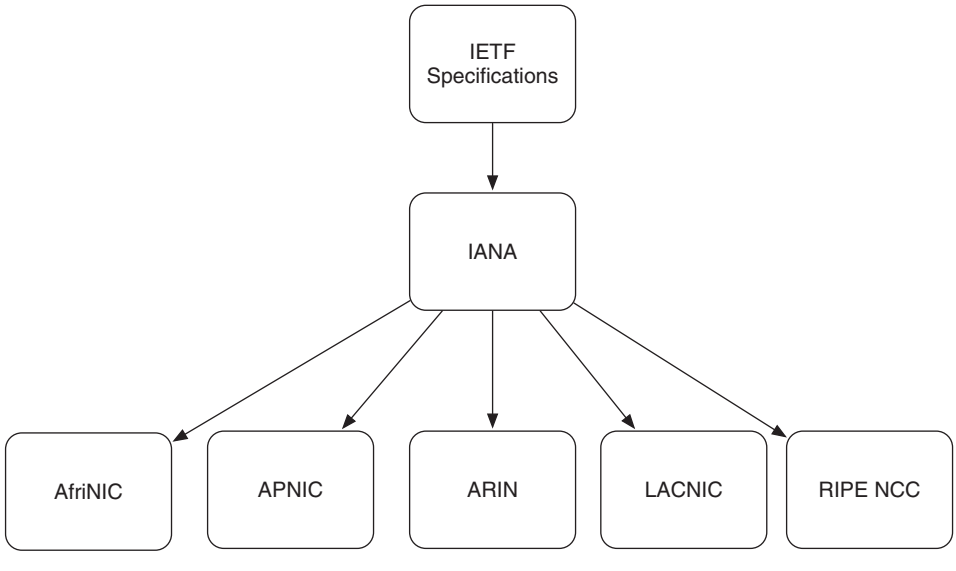

Figure 1.5 IP address allocation hierarchy.

the Regional Internet Registries (RIRs) [35]. The RIRs are responsible for allocating addresses regionally. Today five RIRs exist: African Network Information Center (AfriNIC) in Africa, Asia-Pacific Network Information Center (APNIC) in Asia Pacific, American Registry for Internet Numbers (ARIN), Latin America and Caribbean Network Information Center (LACNIC) in Latin America, and Réseaux IP Européens Network Coordination Centre (RIPE-NCC) in Europe.

The RIRs are responsible of allocating IP addresses to the Local Internet Registries (LIRs) in their region. The LIRs are generally operators, or larger organizations that allocate the IP addresses to the final users of the addresses – the hosts and routers that use the addresses. The allocation of IP addresses to the RIRs from IANA, and to the LIRs from the RIRs, are based on the global or RIR policies. The LIRs usually allocate addresses to the end users based on commercial or technical grounds. Basically, how many IP addresses are allocated to different computers behind a consumer broadband line is based on the contract that the end user has made with the operator.

Different urban legends exist about address allocation. Stories about how some North American universities have more addresses than some very large countries are used to show that the address allocation process is not fair, and favors certain geographic regions over others. Although there is some merit to these stories, the universities, organizations, corporations, and also countries, that participated in the Internet when it was still mostly a research network have gotten relatively large (IP) address allocations – such as class `A`s in the beginning. However, as the Internet has grown, the address allocation policies and process have evolved to the one we have today, which is based on need. Hence, these stories are now mostly outdated.

1.7.2 History of IPv4 Address Exhaustion

On 3 February 2011, IANA stated that it had allocated the final /8 IPv4 address blocks to the RIRs. In April 2011, APNIC said it had reached its final block of /8, and RIPE NCC indicated the same situation in September 2012. Other RIRs are expected to follow suit very soon. Perhaps by the time you have picked up this book, others will have run out of IPv4 addresses as well. Hence, today easily available IPv4 addresses have been exhausted.

The exhaustion of the IPv4 address space, however, was not a surprise, and the technical community has been preparing for it for decades. The first time that the IPv4 addresses were about to be exhausted was at the beginning of 1990s. At that time, the issue was the address classes. As explained before, the addresses were divided in different classes, and the size of the IPv4 address allocation was dependent on the class. This system was very rigid, and it was hard to find a good fit per organization from the classes. Therefore, most bigger organizations needed at least a class `B`. Hence, the class `B` addresses were running out fast as the Internet started to grow at the beginning of the 1990s.

The Internet technical community at the IETF designed the CIDR [36, 37] to enable the allocation of address space in needed block sizes throughout the whole address space. This change prolonged IPv4's lifetime adequately for the community to start redesigning the Internet Protocol, and specifying a new IP protocol version that would have adequate address space for the growth of the Internet. CIDR addressed the efficiency of IPv4 address allocation well enough to accommodate the growth of the Internet throughout the 1990s and the beginning of 2000s.

The growth of broadband networking, and the proliferation of networked computers and other network devices, fueled the Internet's growth. Networks in enterprises, and even in homes grew. This introduced the need to have multiple addresses per subscriber both for enterprise connections, and for private subscribers. However, operators were conserving IP addresses allocated to the subscriber lines. Most operators gave just one address per subscriber, hence allowing just one computer to be connected to the network at a time. There was a need to multiplex multiple computers to a single public IP address. The introduction of private address space [12] and Network Address Translation (NAT) [38] made this possible. NAT is a technique that allows multiplexing of multiple private IP addresses to a single public IP address. The multiplexing is done by using upper layer protocol identifiers – for example the TCP and UDP port numbers. Figure 1.6 describes the NAT principle. Practically, the NAT follows the traffic coming from the inside the network (from the left in the picture) going to the Internet. The NAT rewrites the outgoing packet by replacing the source IP address with its external IP address, and the source port in the transport protocol with one of its available ports. When a downlink packet comes destined to the NAT's address, and the port, it knows to rewrite the packet to go to the host inside the private network. Hence, a NAT extends the IP address range using port numbers.

As the IPv4 address space approaches exhaustion, many operators deploy NATs in their networks, and give private addresses to their customers. Especially in mobile networks NATs have been widespread between the end user and the Internet. That fact combined with the NATs already existing at end user networks, means that usage of NATs can only increase in the future. In addition to CIDR, NAT is the main reason why IPv4 address space has lasted as long as it has.

Although these techniques increase the efficiency of IPv4 address space usage significantly, they do not change the fundamentals of the Internet Protocol and the need for globally routable IPv4 addresses. The IPv4 address space will be exhausted. It has just taken a longer time with these technologies.

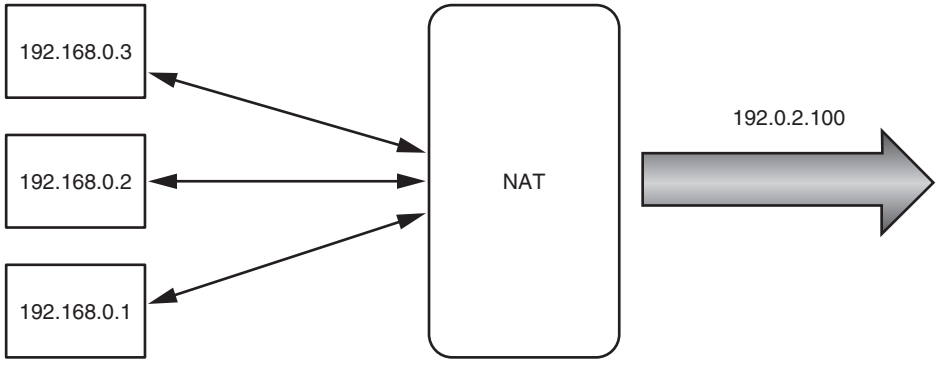

Figure 1.6 Network Address Translation.

1.8 IPv6 History Thus Far

Here we'll try to give a very short summary of the history of IPv6, and give a snapshot in time of the current IPv6 deployment. Writing down the current state of the IPv6 deployment is a calculated risk by the authors; by the time this book actually reaches its readers, the IPv6 deployment will have progressed significantly from what is written here. On the other hand, the authors have stated that the major IPv6 deployment will happen next year and we have said this now for a number of years. However, looking at the current IPv6 deployments seriously, it is clear that the technology maturity, and the overall deployment has taken major steps forward during the last few years. There is no reason to expect that this progress will stop. On the contrary, the deployment can only be expected to accelerate as the exhaustion of the IPv4 addresses progresses.

1.8.1 IPv6 Technology Maturity

At the beginning of 1990s, the IETF started to investigate new technology options for the technical evolution of the Internet Protocol. The consensus of the IETF decided on the new version of the Internet Protocol – the IPv6. By 1998 the IETF had finished the IPv6 base specification RFC 2460 [4]. At the same time of the standardization, multiple implementation projects also existed both in academia, and in the private sector. The best known IPv6 implementation has been done by the WIDE project in Japan [39]. The WIDE Project's IPv6 implementation KAME [40] is the basis of the IPv6 support of many widely used operating systems – Linux and FreeBSD, for instance.

Today all modern mainstream operating systems and many of their variants, including Windows, Mac OS X, Linux, FreeBSD, Symbian, and Android, support IPv6. Most of them also have IPv6 enabled by default. This means that if the local network to which the device is connected to supports IPv6, the operating system can use the capability without the user explicitly enabling IPv6. In addition, many of the mainstream applications have been enabled to support IPv6, including all modern web browsers. Yet, there are many applications that may not support IPv6, with no clear path to migrate to IPv6. Especially, older applications with limited to no support may never be updated.

The major switch, router, and other network equipment vendors providing equipment to operators and corporate networks have offered IPv6 capable devices for some time already. Initially, IPv6 support was included in the test or experimental branches of the software, and sometimes IPv6 was implemented only in software when the IPv4 support might have been hardware accelerated. In addition, initially the IPv6 support might have lacked features present for IPv4 in the same product – this is referred as feature imparity. Hence, the first IPv6 support in products might have been inferior to IPv4 in terms of features and performance. This has changed dramatically in the last few years, and most of the major vendors claim feature parity support with IPv4 and IPv6.

However, a segment that has been slower to adopt IPv6 is the home router vendor community. Only recently has IPv6 capable home network equipment become generally available. Sadly, the major part of the current, and only slowly being upgraded, installed base does not support IPv6 – at least not to the same level as IPv4.

At the beginning of 2000s, the introduction of 3rd Generation (3G) network services and mobile phones was seen as a major driver for IPv6 adoption. However, for a long time 3G phones were used mostly for voice calls, and only secondarily for Internet access. The introduction of mobile broadband services, and of smartphones, changed the course of events, and exactly what was expected happened – just about ten years later. Until recently basically only Nokia Symbian based smartphones supported IPv6. However, now IPv6 is slowly becoming a standard feature in mobile phones due to the requirements of the major mobile network operators.

1.8.2 IPv6 Network Deployments

The fist global IPv6 network was the IPv6 technology testbed 6bone (6bone) [41]. The 6bone was created by the IETF IPv6 community to provide a place where the IPv6 technology could be tested, connecting different IPv6 test projects, and for gathering IPv6 operational experience. The network itself was mostly a virtual network deployed over the global Internet infrastructure. It had its own network prefix `3FFE::/16` [42] that was temporarily assigned to the testbed. The testbed was operational from 1996 to 2006 when it was decommissioned on 6 June 2006 (note the date: 6.6.6) marking both the end of the 6bone and the first IPv6 day. Sometimes erroneously, 6bone has been said to be the Internet IPv6 network backbone. However, the intention was always for 6bone to be a temporary testbed in the test phase of the technology. As IPv6 technology matured, and the Internet itself became increasingly IPv6 capable, 6bone was no longer needed. The 6bone prefixes are not routed over the Internet anymore.

The definition of the Internet core is a difficult one. Due to the distributed nature of the Internet, it is difficult to state which operators provide the true core of the Internet. Often, so-called Internet Tier 1 operators are considered to be the core of the Internet. But this is not quite true. These operators are often the larger operators with global presence. The major operators providing Internet transit to other operators, including the Tier 1 operators, have provided IPv6 service both to their operator and business customers. In that sense, one can claim that the Internet core network has been IPv6 capable already for some time. This development has happened incrementally over many years, and has been mostly invisible to the end user. The operators have also had good transition tools, which have made it relatively easy to support IPv6 in the Internet core.

Though, some progressive access network operators have deployed IPv6 and provided it to the end users, most of the world's access operators do not provide IPv6, yet. There are many reasons for the slowness of IPv6 deployment in the Internet access networks. First of all, IPv6 is a not a feature that operators can sell to the end users. This absence of an evident business case has most probably been the number one reason for the slow adoption of IPv6 in access networks. This is not a feature that the users would currently demand, or for which they would be prepared to pay. Other important contributors are the relatively slow investing cycle – especially in the fixed access networks – and the result of the presence of very old equipment in the networks. Even though new equipment does support IPv6 well, access operators may have equipment that is as much as ten years old in their network, which would have to be replaced. Investment without a prospect of more revenue is never popular in the management of a company. In addition, as mentioned earlier, the lack of IPv6 capable home routers may have influenced the

deployment plans. On the other hand, it is hard to say what is the cause and what is the symptom. Looking on the bright side, the number of IPv6 capable access networks is steadily growing, including big fixed access operators like Comcast providing commercial service already.

One reason stated for the slow uptake of the IPv6 access has been the lack of IPv6 services. For a long time, the major Internet services (like Google, Facebook, Yahoo and others) did not support IPv6, or supported it only partially. This was also recognized by the Internet community. The Internet Society started to organize World IPv6 Day events. The 2011 World IPv6 Day provided a day during which major Internet service providers turned IPv6 on their main websites for a day to operationally test IPv6. This was considered to be a great success. In 2012, the Internet Society coordinated the World IPv6 Launch, where many Internet service providers turned IPv6 on permanently. This has at least partly released the chicken-and-egg situation between the access network operators and the Internet content providers. It gives good proof of how positive coordination, or perhaps rather peer pressure, can help to bring the industry together.

1.9 Ongoing Cellular Deployments

The first 3GPP operator to deploy IPv6 in a commercial network was Sonera (nowadays TeliaSonera) in Finland. They started a trial of IPv6 in their commercial network back in 2004. The IPv6 support was not commercialized, and therefore not accessible to normal end users. Perhaps, the first commercial service deployments were in Slovenia where the national cellular network providers commercialized IPv6 at the same time in 2010, coordinated by the Slovenian GO6 institute – a not-for-profit industry association in Slovenia.

Recently major operators in North America have started open IPv6 trials, and some of them have commercialized IPv6 with the new Long Term Evolution (LTE) cellular networks. In addition, some Asian operators have launched their IPv6 service, together with the LTE network launch.

The major network equipment vendors have had at least basic support for IPv6 service for the end user for quite some time now. However, advanced services that the operators use in their current IPv4 Internet offering might not have been supported for IPv6. The same applies to mobile network equipment vendors as it does for fixed network equipment vendors – the support for IPv6 has been improved dramatically over the last few years. The network equipment, subscriber database solution, and network management solutions, are available with IPv6 support.

The deployment in operators' networks has been painfully slow. Though, in the early days of IPv6, 3G deployment was thought to be the driver of IPv6 deployment, the mobile broadband operators have been slow in adopting the new Internet technology. However, it seems that even the cellular deployment is now gaining momentum. All major cellular operators are either busy deploying IPv6 in their networks, or are planning the deployment. The mobile handset operating systems, the cellular chipsets, and the mobile applications increasingly support IPv6, eliminating one major problem from the IPv6 transition in the mobile broadband market. It should be expected that the cellular community will be very busy with IPv6 in the coming few years. A major reason that this book was written was to help people joining, or already participating in, the IPv6 deployment activities to get up to speed quicker.

1.10 Chapter Summary

In this chapter, we concentrated on giving an introduction to the Internet, Internet technology, address exhaustion, and the deployment status at the time this book was written. The Internet has been built on a set of principles, which are embedded in the technology and in the Internet itself. These principles are repeated in the following:

- packet switched networking;
- the end-to-end principle;
- layered architecture;
- Postel's robustness principle; and
- creative anarchy.

The IP is the technology that builds the Internet. It provides an unreliable packet delivery service where the packet delivery is based on the destination IP address. Currently, two versions of IP exist: IPv4 and IPv6. The IPv4 is the protocol that is predominantly used in the Internet. However, IPv4 addresses have been mostly depleted, posing a risk to the growth of the whole Internet. Therefore the Internet, including 3GPP cellular networks, is in the biggest technical transition of its existence – the transition to IPv6.

1.11 Suggested Reading

- Where Wizards Stay Up Late, The origins of the Internet, K. Hafner and M. Lyon
- Routing in the Internet, C. Huitema
- TCP/IP Illustrated, Volume 1: The Protocols (2nd Edition), K. Fall, W. R. Stevens
- Brief History of the Internet – Internet timeline http://www.internetsociety.org/internet/internet-51/history-internet/brief-history-internet
- DNS and BIND, 5th Edition, C. Liu, P. Albitz

References

1. ITU-T. Data Networks and Open System Communication Open System Interconnection – Model and Notation Information Technology – Open System Interconnection – Basic Reference Model: The Basic Model. Recommendation Q.700, International Telecom Union – Telecommunication Standardization Sector (ITU-T), July 1994.
2. Braden, R. *Requirements for Internet Hosts – Communication Layers*. RFC 1122, Internet Engineering Task Force, October 1989.
3. Postel, J. *Internet Protocol*. RFC 0791, Internet Engineering Task Force, September 1981.
4. Deering, S. and Hinden, R. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460, Internet Engineering Task Force, December 1998.
5. Postel, J. *Transmission Control Protocol*. RFC 0793, Internet Engineering Task Force, September 1981.
6. IEEE Society Computer. *Part 3: IEEE Standard for Information technology – Specific requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*. IEEE Standard for Information Technology 802.3, Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA), December 2008.
7. Moy, J. *OSPF Version 2*. RFC 2328, Internet Engineering Task Force, April 1998.
8. Coltun, R., Ferguson, D., Moy, J., and Lindem, A. *OSPF for IPv6*. RFC 5340, Internet Engineering Task Force, July 2008.

9. ISO. *Information technology – Telecommunications and information exchange between systems – Intermediate System to Intermediate System intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)*. International Standard 10589, International Organization for Standardization (ISO), March 2008.
10. Rekhter, Y., Li, T., and Hares, S. *A Border Gateway Protocol 4 (BGP-4)*. RFC 4271, Internet Engineering Task Force, January 2006.
11. Cotton, M. and Vegoda, L. *Special Use IPv4 Addresses*. RFC 5735, Internet Engineering Task Force, January 2010.
12. Rekhter, Y., Moskowitz, B., Karrenberg, D., deGroot, G. J., and Lear, E. *Address Allocation for Private Internets*. RFC 1918, Internet Engineering Task Force, February 1996.
13. Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and Azinger, M. *IANA-Reserved IPv4 Prefix for Shared Address Space*. RFC 6598, Internet Engineering Task Force, April 2012.
14. Cotton, M., Vegoda, L., and Meyer, D. *IANA Guidelines for IPv4 Multicast Address Assignments*. RFC 5771, Internet Engineering Task Force, March 2010.
15. Hinden, R. and Deering, S. *IP Version 6 Addressing Architecture*. RFC 4291, Internet Engineering Task Force, February 2006.
16. Kawamura, S. and Kawashima, M. *A Recommendation for IPv6 Address Text Representation*. RFC 5952, Internet Engineering Task Force, August 2010.
17. Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and Li, X. *IPv6 Addressing of IPv4/IPv6 Translators*. RFC 6052, Internet Engineering Task Force, October 2010.
18. Postel, J. *User Datagram Protocol*. RFC 0768, Internet Engineering Task Force, August 1980.
19. Cotton, M., Eggert, L., Touch, J., Westerlund, M., and ire, S. Cheshire. *Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry*. RFC 6335, Internet Engineering Task Force, August 2011.
20. Postel, J. and Reynolds, J. *File Transfer Protocol*. RFC 0959, Internet Engineering Task Force, October 1985.
21. Postel, J. and Reynolds, J. K. *Telnet Protocol Specification*. RFC 0854, Internet Engineering Task Force, May 1983.
22. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616, Internet Engineering Task Force, June 1999.
23. IANA. Service Name and Transport Protocol Port Number Registry, http://www.iana.org//assignments/service-names-port-numbers/service-names-port-numbers.xml.
24. Mockapetris, P. V. *Domain names – implementation and specification*. RFC 1035, Internet Engineering Task Force, November 1987.
25. Thomson, S., Huitema, C., Ksinant, V., and Souissi, M. *DNS Extensions to Support IP Version 6*. RFC 3596, Internet Engineering Task Force, October 2003.
26. IANA. Root Zone Database, http://www.iana.org/domains/root/db/.
27. ICANN, New Generic Top-Level Domains, http://www.icann.org.
28. ISO. *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes (ISO 3166-1:2006)*. International Standard 3166-1:2006, International Organization for Standardization (ISO), November 2006.
29. Klensin, J. *Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework*. RFC 5890, Internet Engineering Task Force, August 2010.
30. Klensin, J. *Internationalized Domain Names in Applications (IDNA): Protocol*. RFC 5891, Internet Engineering Task Force, August 2010.
31. Faltstrom, P. *The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)*. RFC 5892, Internet Engineering Task Force, August 2010.
32. Alvestrand, H. and Karp, C. *Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)*. RFC 5893, Internet Engineering Task Force, August 2010.
33. IANA. Internet Assigned Numbers Authority(IANA), http://www.iana.org.
34. ICANN. Internet Corporation for Assigned Names and Numbers (ICANN), http://newgtlds.icann.org/en/.
35. Organization, Number Resource. Regional Internet Registries, http://www.nro.net/about-the-nro/regional-internet-registries.

36. Rekhter, Y. and Li, T. *An Architecture for IP Address Allocation with CIDR*. RFC 1518, Internet Engineering Task Force, September 1993.
37. Fuller, V., Li, T., Yu, J., and Varadhan, K. *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*. RFC 1519, Internet Engineering Task Force, September 1993.
38. Egevang, K. and Francis, P. *The IP Network Address Translator (NAT)*. RFC 1631, Internet Engineering Task Force, May 1994.
39. Project, WIDE. The WIDE Project, http://www.wide.ad.jp/.
40. Project, WIDE. The KAME Project, http://www.kame.net/.
41. Fink, R. and Hinden, R. *6bone (IPv6 Testing Address Allocation) Phaseout*. RFC 3701, Internet Engineering Task Force, March 2004.
42. Hinden, R., Fink, R., and Postel, J. *IPv6 Testing Address Allocation*. RFC 2471, Internet Engineering Task Force, December 1998.
43. Mockapetris, P.V. *Domain names – concepts and facilities*. RFC 1034, Internet Engineering Task Force, November 1987.
44. Cheshire, S., Aboba, B., and Guttman, E. *Dynamic Configuration of IPv4 Link-Local Addresses*. RFC 3927, Internet Engineering Task Force, May 2005.
45. Huston, G. *Operational Requirements for the Address and Routing Parameter Area Domain (arpa)*. RFC 3172, Internet Engineering Task Force, September 2001.