

COMPUTER SCIENCE

UNLEASHED

HARNESS THE POWER OF
COMPUTATIONAL SYSTEMS

WLADSTON FERREIRA FILHO
RAIMONDO PICTET



**COMPUTER
SCIENCE**
UNLEASHED

COMPUTER SCIENCE UNLEASHED

HARNESS THE POWER OF
COMPUTATIONAL SYSTEMS

WLADSTON FERREIRA FILHO
RAIMONDO PICTET



code energy

Las Vegas

©2021 Wladston Ferreira Filho and Raimondo Pictet

All rights reserved.

Published by CODE ENERGY, INC.

✉ hi@code.energy

🌐 <http://code.energy>

📷 <http://instagram.com/code.energy>

🐦 http://twitter.com/code_energy

👍 <http://facebook.com/code.energy>

🏠 304 S Jones Blvd # 401 Las Vegas NV 89107 🇺🇸

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without permission from the publisher, except for brief quotations embodied in articles or reviews.

While every precaution has been taken in the preparation of this book, the publisher and the authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Publisher's Cataloging-in-Publication Data

Ferreira Filho, Wladston.

Computer Science Unleashed: harness the power of computational systems / Wladston Ferreira Filho; with Raimondo Pictet. — 1st ed. x, 260 p. : il.

ISBN 978-0-9973160-3-2 (Hardback)

ISBN 978-0-9973160-4-9 (ebook)

1. Computer networks. 2. Internet. 3. Computer network protocols. 4. Regular expressions (computer science). 5. Statistics. 6. Data mining. 7. Machine learning. I. Title.

004 – dc22

2020925732

First Edition, December 2022.

*To our friends Christophe and Mateus, one of whom
bet we would finish this book by the end of the year.*

Computer science has a lot in common with physics. Both are about how the world works at a rather fundamental level. The difference is that while in physics you're supposed to figure out how the world is made up, in computer science you create the world. In mathematics, as in programming, anything goes as long as it's self-consistent. You can have a set of equations in which three plus three equals two. You can do anything you want.

—LINUS TORVALDS

Explaining where his love for computers stems from.

CONTENTS

PREFACE	ix
I CONNECTIONS	1
1.1 Links	2
1.2 Internet	9
1.3 IP Addressing	16
1.4 IP Routing	23
1.5 Transport	32
2 COMMUNICATION	45
2.1 Names	46
2.2 Time	57
2.3 Access	63
2.4 Mail	66
2.5 Web	74
3 SECURITY	87
3.1 Legacy	88
3.2 Symmetry	96
3.3 Asymmetry	103
3.4 Hashing	108
3.5 Protocols	115
3.6 Hacking	119
4 ANALYSIS	133
4.1 Collection	135
4.2 Processing	139
4.3 Summarizing	146
4.4 Visualization	155
4.5 Testing	170
5 LEARNING	181
5.1 Features	185
5.2 Evaluation	196
5.3 Validation	200
5.4 Fine-Tuning	205

CONCLUSION **221**

BONUS: PATTERNS **223**

 Matching 224

 Quantifiers 230

 Anchors 233

 Groups 236

APPENDIX **241**

 I Numerical Bases 241

 II Cracking the Shift Cipher 242

 III Cracking the Substitution Cipher 244

 IV Evaluating Classifiers 246

INDEX **255**

PREFACE

I never liked the term ‘computer science’. The main reason I don’t like it is that there’s no such thing. Computer science is a grab bag of tenuously related areas thrown together by an accident of history, like Yugoslavia.

—PAUL GRAHAM

Most technological breakthroughs of our era are taking place in a new digital world created by programmers. Computer scientists combine different fields of study in order to empower this new world. This book explores the foundations of some of these fields, including networking, cryptography, and data science.

We’ll start with the story of how two computers can be linked to share information, and take you all the way to the rise of email and the Web. We’ll explore cryptography and understand how the Internet and other systems that deal with private data are made secure. Then, we’ll learn how knowledge can be obtained from raw data and how machines can be taught to forecast the future.

We hope these stories will familiarize you with important concepts that can benefit coders and tech enthusiasts alike. Our goal is to cover what beginners need in order to get up to speed in networking, security and data science, without the heavy academic rigor that sometimes makes these topics unbearable.

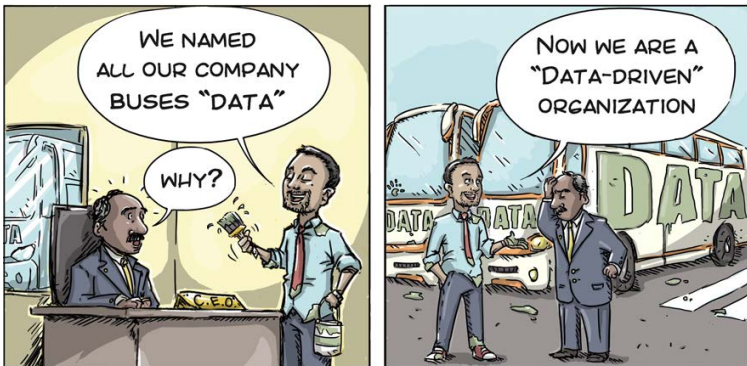


Figure 1 “Data is the new oil”, by Amit Dangle & Ivano Nardacchione.

This book was made possible by the supporters of our previous title, *Computer Science Distilled*. We had written our first book to explain the fundamental principles of computer science. Our enthusiastic readers asked for more, so we got back to work! This time, we don't explore the core of our discipline, but rather the new worlds it has enabled us to create.

Is this book for me?

If you're a novice programmer, this book was written for you. It doesn't require any programming experience, as it essentially presents ideas and mechanisms: we want you to learn how cool stuff works. If you're curious and want to understand how the Internet is built, how hackers attack computer systems, or why data is the gold of the 21st century, you'll find this book worthwhile. And for those who already studied computer science, this book is a great recap to consolidate your knowledge and expertise.

Acknowledgments

We are deeply grateful for everyone who supported our multi-year effort to create this book. We would especially like to thank Abner Marciano, André Lambert, Caio Magno, Carlotta Fabris, Damian Hirsch, Daniel Stori, Eduardo Barbosa, Gabriel Pictet, Guilherme Mattar, Jacqueline Wilson, Leonardo Conegundes, Lloyd Clark, Michael Ullman, Rafael Almeida, Rafael Viotti, and Ruhan Bidart. Finally, we're grateful to Claire Martin, our proofreader, and Pedro Netto, our illustrator, for making this book so much better.

May you create many worlds,
Wlad & Moto

CHAPTER 1






Connections

This is an entirely distributed system, there isn't any central control. The only reason it works is because everybody decided to use the same set of protocols.

—VINT CERF

HUMANS CRAVE CONNECTIONS, and the advent of the digital revolution has empowered us to be more connected than ever before. The Internet has unleashed upon billions of people unprecedented economic and political freedom, as well as powerful means of control and domination. Yet, the vast majority of us are oblivious to its inner workings.

Skilled people who can program computers to use the Internet are at the vanguard of the digital revolution. This chapter will teach you how the Internet works, so you can join this select group. You'll learn to:

-  **Link** computers to one another to make a network,
-  Combine networks using the **Internet Protocol**,
-  Locate a recipient from its Internet **address**,
-  Find a **route** through the Internet to that location,
-  **Transport** data between distant applications.

Before the Internet came along, telecommunication between two parties required a direct physical link. In the 1950s, each telephone had a wire leading directly to a central station. For a call to go through, an operator had to physically connect the wires of two telephones. For long distance calls, wires were laid out between distant stations, and several operators in different places had to physically connect the chain of wires linking the two phones.

The Internet did away with this. Wires aren't physically reconfigured to create direct, exclusive links. Instead, the information

is retransmitted step by step via a chain of linked devices until it reaches its destination. This eliminates the need for wire operators and central coordination. Also, wires are no longer constrained to serve a single connection—many concurrent connections can share the same wire. This allows global communications to be instant, cheap and accessible.

However, modern networking technology is more intricate than early telephony. It has many successive layers, each building on top of the previous. Let's explore how connections are made at these different levels, starting with the most basic layer.

1.1 Links

A direct connection between two computers is achieved through a **transmission medium**: a physical channel where signals can flow. This may be a copper wire carrying electric currents, a fiber-optic cable directing light, or air hosting radio waves. Each connected computer has a **network interface** to send and receive signals in the transmission medium. For instance, cellphones have a radio chip and antenna to handle radio signals traveling through the air.

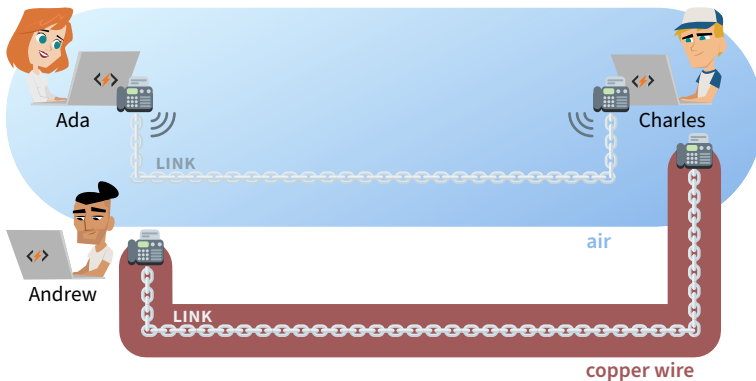


Figure 1.1 A link is established between two network interfaces if they share a transmission medium and agree on the rules of communication.

In order to communicate, network interfaces must agree on the rules to follow when sending and receiving signals. This set of rules is called the **link layer**.

When a medium *exclusively* connects two computers, we say they maintain a point-to-point connection, and their link layer relies on the most basic set of rules: the **Point-to-Point-Protocol (PPP)**. It merely ensures the two computers can identify each other and exchange data accurately.

However, connected computers don't always get to enjoy such an exclusive link. Often, they must share the transmission medium with several other computers.

Shared Links

One way to link computers in an office is to plug each of them into a hub with a wire. The hub physically connects all the wires that reach it, so a signal sent by one computer will be detected by *all* the others! This will also happen on your home WiFi, since the same radio frequency is used by all connected devices. Communications can become messy if all of them use the medium at the same time.

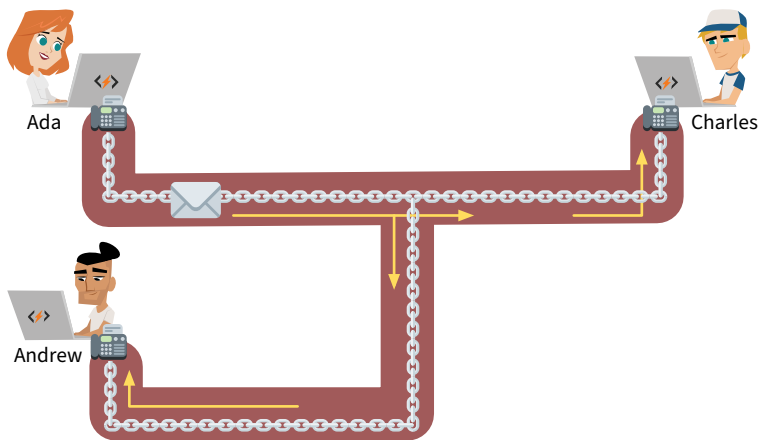


Figure 1.2 A message sent on a shared link will be detected by all.

The link layer contains a set of rules to define how computers should share their communication medium, fittingly called **Medium Access Control (MAC)**. The rules resolve two main challenges:

COLLISIONS If two computers send a signal through the same medium at the same time, the resulting interference garbles both transmissions. Such events are called **collisions**. A similar problem occurs when your group of friends or family talk over each other and no single voice can be clearly heard.

There are methods to avoid collisions. First, only start transmitting signals when no other computer is transmitting. Second, monitor your communications—if a collision occurs, wait for a brief but random amount of time before trying to transmit again.

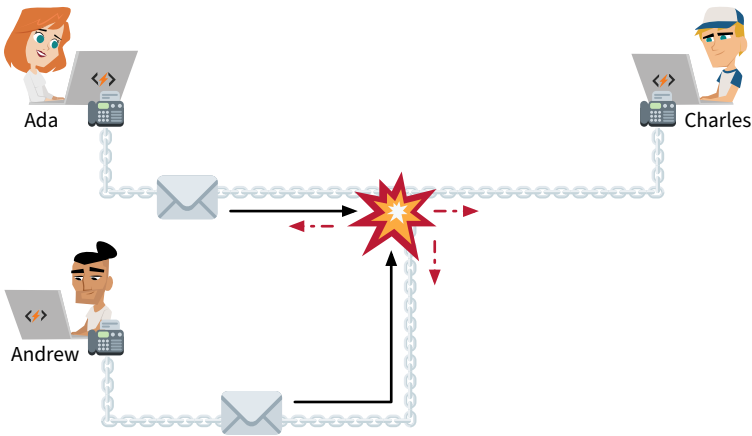


Figure 1.3 Collision between Ada and Andrew.

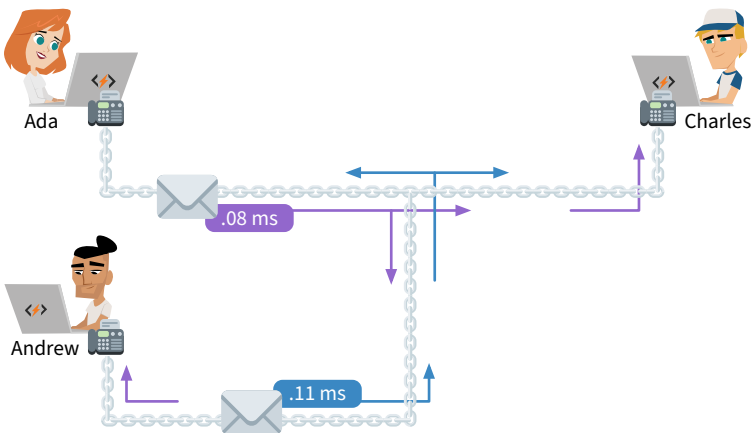


Figure 1.4 Ada and Andrew both resend after a random duration.

These methods have some limitations. When there are too many transmission attempts through a medium, collisions occur relentlessly. We say the link is **saturated** when excessive collisions break down communications. Have you ever been frustrated at a large venue because your phone wouldn't send text messages or make calls? This may happen if too many phones are attempting to communicate concurrently and the cellular link becomes saturated.

PHYSICAL ADDRESSING Ada and Charles have a direct link between their computers. Ada wants to talk with Charles, so she transmits a signal with her message through the medium. However, the medium is shared, so everyone linked to the medium gets the message. How can the other computers know that the signal they picked up was not destined for them?

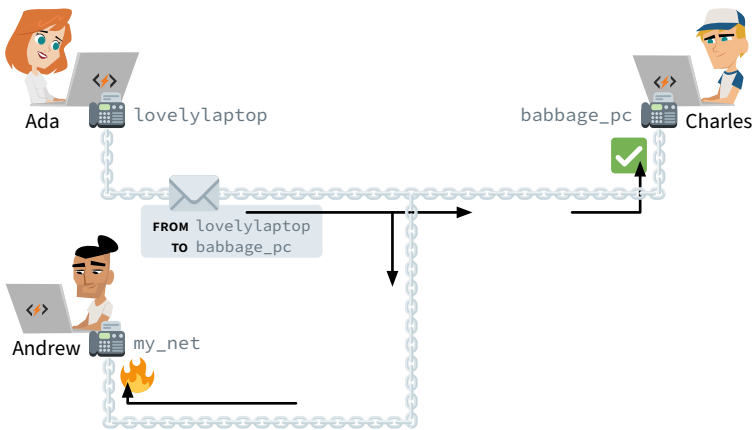


Figure 1.5 Andrew's network interface discards the message.

Each computer's network interface has an identifier, known as its **physical address** or **hardware address**. A transmission in a shared medium must begin with two such addresses: that of the recipient and that of the sender. Upon receiving a transmission, a computer will know if it should be ignored or picked up and to which address it should reply.

This can only work if physical addresses are unique: if two computers use “my_netinterface”, we’re back to square one. For this reason, virtually all network interfaces follow a naming scheme defined in the rules of Medium Access Control. These standard physical addresses are called **MAC addresses**.

MAC Addressing

Computers, smartphones, smart watches, and smart televisions can each have WiFi, Bluetooth, and Ethernet network interfaces. Each network interface has its own, unique MAC address marked into the hardware during production. You should not worry about assigning a MAC address to your computer: you can always use the one that came with its network interface.

Since MAC addresses are simply large random-looking numbers, network interface manufacturers around the world must coordinate to avoid accidentally assigning the same number to two different devices. To this end, they rely on the Institute of Electrical and Electronics Engineers, or **IEEE**, which assigns each of them a different range of MAC addresses.

A MAC address is expressed as six pairs of hexadecimals¹ separated by colons. The first half of the address is an identifier assigned by the IEEE to a unique manufacturer. This manufacturer then chooses a unique second half for each network interface.

```
60:8B:0E:C0:62:DE
```

Here, **608B0E** is the manufacturer number. This specific number was assigned by IEEE to Apple, so this MAC address should belong to an Apple device.² A device’s MAC address is often written on a label stuck to the packaging or on the device itself, next to the serial number.

¹In day-to-day life, we almost always express numbers in decimal form, where each digit is one of ten characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Computer scientists, on the other hand, like expressing numbers in hexadecimal form, where each digit can be one of sixteen characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f. For more about number bases, see Appendix I.

²You can look up who manufactured a device by entering the first six digits of its MAC address at <http://code.energy/mac-lookup>.

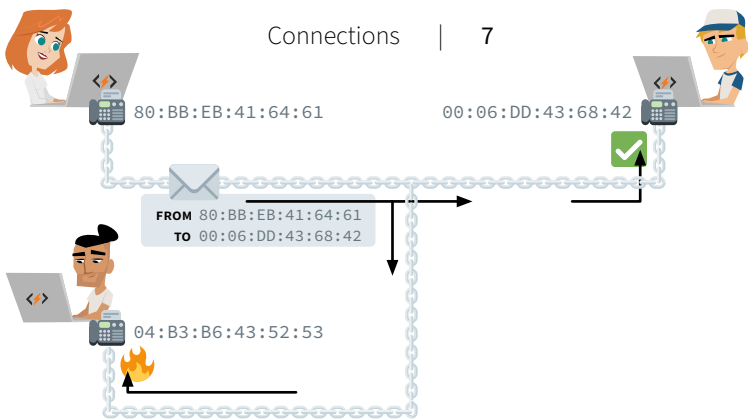


Figure 1.6 Each MAC address is unique.

There's a special address reserved for transmissions to all computers in a medium. It's called the **broadcast address**, and it reads `FF:FF:FF:FF:FF`. You use it when you try to connect to an unknown device. For instance, when your smartphone's WiFi card isn't deactivated, it persistently broadcasts to `FF:FF:FF:FF:FF` that it's looking for an access point. Discoverable access points will respond with their own MAC address so you can establish a link.

Such discovery broadcasts, like all other transmissions, contain the sender's MAC address. Walking around with a smartphone can therefore be like walking around with a loudspeaker shouting your name non-stop, only using radio waves instead of sound and the MAC address instead of your moniker. In 2013, Edward Snowden revealed that the NSA³ monitored the movements of people by sniffing WiFi transmissions in big cities, storing records of where each MAC address was seen.

You can also set your own network interface to **promiscuous mode**, and it will pick up all transmissions regardless of their intended recipient. Doing so allows you to discover hidden WiFi networks, to list which MAC addresses are in your area, and sometimes even to read the contents of other people's transmissions. Browsing the Internet through an unsecured WiFi network can therefore be unsafe: anyone in range can hear what you broadcast. This is why encryption⁴ is important for WiFi's link layer.

Be careful: a network interface can be configured for its transmissions to start with any MAC address both for the recipient *and*

³National Security Agency, a US government spying organization.

⁴Encryption allows messages to look garbled to eavesdroppers.

for the sender. Nothing stops a malicious agent from impersonating you by using your physical address in their transmissions. This type of attack is known as **MAC spoofing**. When the link layer was originally developed, security wasn't a concern. Protocols are evolving to become more secure and neutralize such attacks, but it's an ongoing process.

Frames

Sometimes, a transmission must contain a lot of data, and sending out a single, big fat message is impractical. Network interfaces and computers are not all capable of the same transmission speeds. Moreover, what would happen if a collision occurred in the middle of the transmission? The entire transmission would have to be discarded, as it would be difficult for the sender and receiver to determine exactly which parts of the message were received and which were not.

To solve these issues, long messages are always split into small parts, each sent as an independent transmission. The time between transmissions can vary according to the capabilities of both computers: slower devices need longer breaks. If an error occurs, it is only necessary to discard and resend the small transmission that failed.

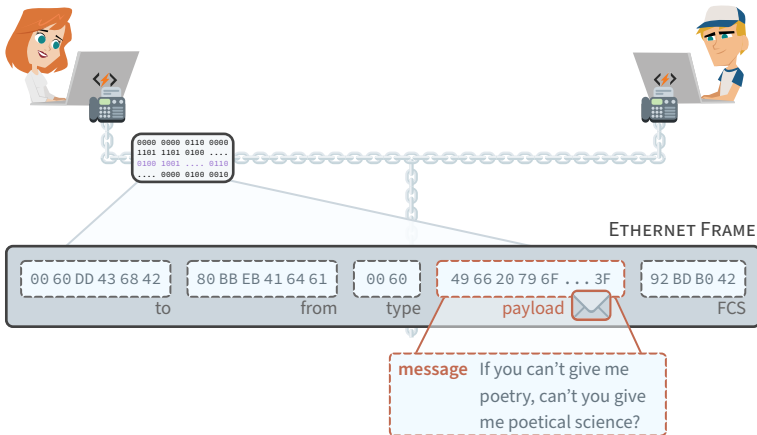


Figure 1.7 An Ethernet frame. Once it is transmitted in a copper wire, it becomes a series of electric signals that encode a number. The Ethernet protocol instructs how to interpret this number. For instance, the first 12 hex digits of the number encode the destination MAC address.

Each independent transmission is called a **frame**. Standard WiFi protocols cap the size of frames to 2,346 bytes. Thirty-four bytes are needed for MAC addresses and error-detecting codes. Therefore, a WiFi frame can ultimately carry up to 2,312 bytes of data, called the payload.⁵ In wired networks, the maximum frame size is usually 1,526 bytes, with room for a 1,500 byte payload.

On rare occasions, disturbances in the medium interfere with a transmission, and the receiver picks up signals that don't encode exactly the same information that the sender intended to transmit. Let's see the special field that was added to address this problem.

FCS The last part of the frame is the **FCS (Frame Check Sequence)**, and it ensures that information was transmitted accurately. The FCS doesn't add new information to the transmission: it is merely the result of a calculation using the contents of all other fields. Changing any content before the FCS should cause the FCS number to change as well.

Upon receiving a frame, a computer calculates the *expected* FCS number from the information it received and compares it to the *received* FCS. If they don't match, the frame is discarded. If they match, we know that the message wasn't garbled and trust that the received payload is error-free.

TYPE The frame shown in Figure 1.7 has one last field we haven't talked about: the payload **type**. It tells the receiver which rules should be followed to interpret the data in the frame's payload. In the next section, we'll explore the most common set of such rules.

1.2 Internet

We've seen that the link layer enables directly connected computers to exchange messages inside frames. The **internet layer**, also known as the **network layer**, specifies how to transmit these messages between computers that are *not* directly connected.

The trick is to equip some computers, called **routers**, with multiple network interfaces. All computers in a network are then linked

⁵If we encode one byte per character, a WiFi frame has room for about 500 words, enough to fill a page of text.

to at least one router, and all routers are linked to at least one other router. When a router receives a message at one of its network interfaces, it can forward it to another router through a different network interface.

LOCAL AREA NETWORKS We can ask a router we're linked with to forward a message to a computer we're not linked with. Suppose you have a wired network in your home connecting a router and a desktop computer. Suppose the router is also directly connected to a smartphone in a different, wireless network.

Even though the desktop computer and the smartphone are not directly connected to the same network, they can send messages to each other using the router as a relay. Computers from different networks in close vicinity that can talk to each other through routers form a larger network, called a **Local Area Network (LAN)**.

In a home or small office, one router will be enough to link all the computer networks in the area. When assembling a LAN that covers a large organization such as a university or hospital, many routers may be required to link all the different computers networks into a fully connected system.



Figure 1.8 In this small LAN, Ada and Andrew can send messages to each other through their router Charles.

WIDE AREA NETWORKS But why stop there? If your router is linked with a router outside your home, which in turn is linked with a router at the university, you can ask for your message to be forwarded to computers on the university's LAN. When distant LANs are connected to each other, they form a **Wide Area Network (WAN)**.

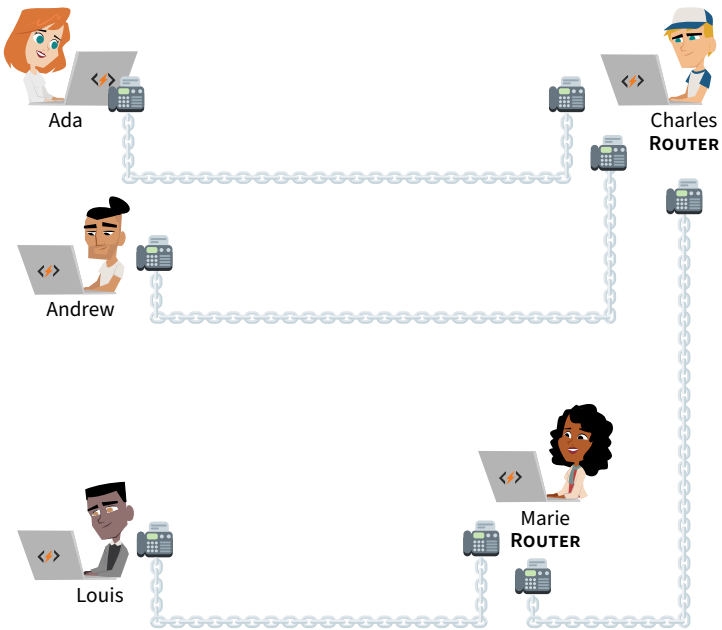


Figure 1.9 Charles is connected to a distant router, Marie, and they both forward messages around this WAN.

A WAN can grow larger as more LANs are connected to it. Different WANs can also be connected to form an even larger WAN. The largest WAN in the world is a collection of thousands of **interconnected networks** that we call **the Internet**. It's the network we use every day to send emails and browse the web; and as of 2020, it contained over a billion computers. Let's see how they all got connected.

Interconnection

The most straightforward way to connect your router to the Internet is to pay for it. Some organizations on the Internet will link one of their routers to yours, and allow messages to and from your network to pass through their network via this link. This paid service is called **transit**, as all of your messages will *transit* through their network before going to the specific router you're aiming for.

However, transiting through a third party network is not always necessary in order to connect to another router of the Internet. Say, for example, that two nearby universities communicate a lot; they can link their routers in order for messages to flow directly between their networks. This can save money, as these messages would otherwise have to transit through a paid connection. The free exchange of messages between the networks of different organizations is called **peering**.

Routing

Any computer linked to a router of the Internet can ask for its messages to be forwarded by other routers. Messages can be routed over large distances. For instance, there is a system of submarine cables linking routers in many coastal cities:



Figure 1.10 The SAM-1 system links routers in 16 cities from 11 different countries, using over 15 thousand miles of underwater cables.

There is no direct link between the routers in Miami and Buenos Aires. However, Miami is linked with Puerto Rico, which is itself linked with Fortaleza, which is linked with Rio de Janeiro, which is finally linked with Buenos Aires. Miami and Buenos Aires can exchange messages through these cables if routers along the way forward them back and forth. Today, there are submarine cables linking hundreds of coastal city routers around the globe:

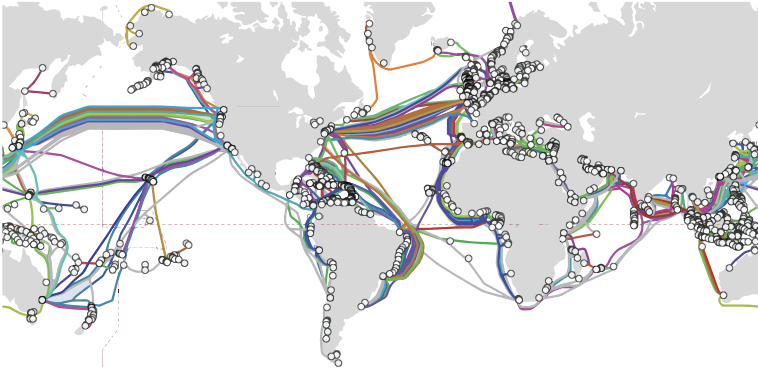


Figure 1.11 Fiber-optic submarine cables currently in service.

Virtually every other city on Earth is directly or indirectly linked to these coastal cities, often through cables in the ground. Communication satellites also have routers to establish wireless links to remote locations. All routers can forward messages, so a message you send on the Internet can be routed to any other computer on the Internet. That is, *if* a path to it can be found.

Location Addressing

In the link layer, computers are identified by a physical address. Physical addresses uniquely identify computers, but they don't give any hints on *where* a computer is connected and how it can be reached. If the computer moves to the other side of the world, it will retain its physical address!

Suppose you mailed a package to Louis through the post along with a picture of him instead of his address. This package has a defined destination; however, an international postal service would

have no way of knowing which direction the package should be sent in order to deliver it to Louis.

Post offices must first know to which country the package should go. The first post office in that country should then know to which province or state it should go. The next post office should know the city, and the final post office, the street address. We call an address containing all this information a **hierarchical address**. Similarly to post offices, routers require packages to carry a hierarchical address of their recipient's location:

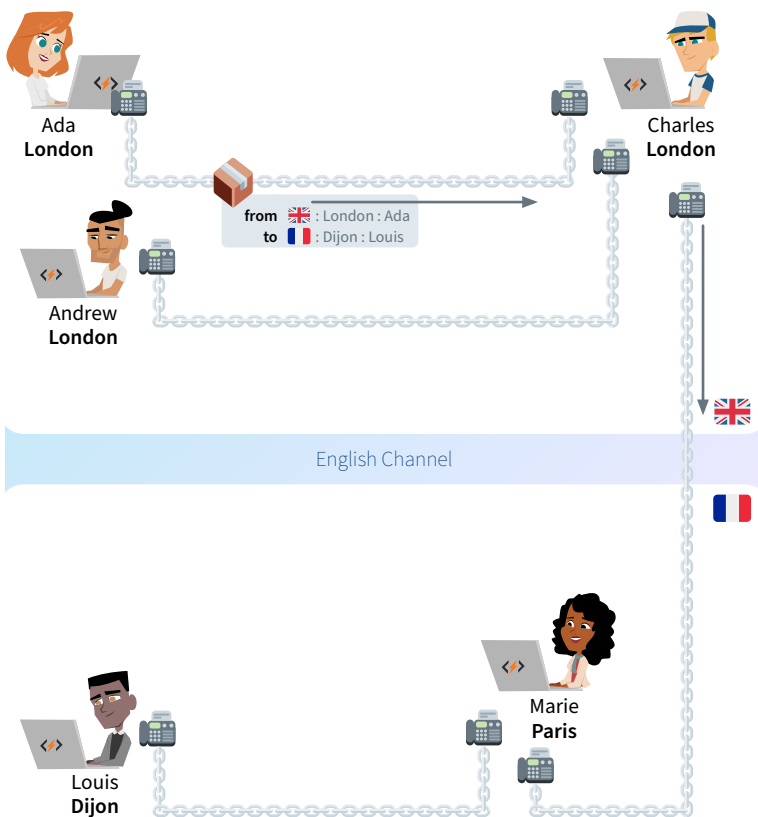


Figure 1.12 Ada wishes to send a package to Louis, so she requests her router Charles to forward it. She writes on the package a hierarchical address of Louis. Charles then knows he must send the package to France, so he sends it to the French router he is linked with: Marie.

For this mechanism to work on a global scale, *all* computers involved must follow the same set of rules to create and handle package forwarding requests. A computer in China must understand a request from a computer in Nigeria, even though the two may use different languages, operating systems and hardware.

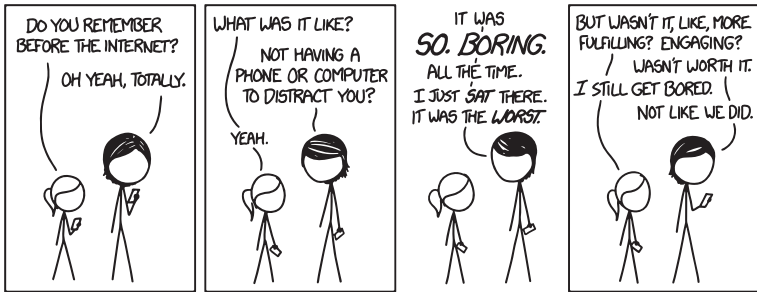


Figure 1.13 “Before the Internet”, courtesy of <http://xkcd.com>.

Internet Protocol

We’ve seen a computer must follow the rules of Medium Access Control to establish a link with another computer. Similarly, it must follow the **Internet Protocol**, or **IP**,⁶ to ask routers to forward messages to other computers on your LAN or on the Internet.

A message forwarding request that follows the IP rules is called an **IP packet**. The IP packet is essentially a big number, where digits in specific positions encode key information. Almost every computer produced in the past few decades understands IP packets and is able to forward them. This makes an IP packet easily movable from one computer to the next, until it reaches its destination.

An IP packet contains the *location* addresses of its sender and recipient, followed by whatever data they want. To send an IP packet, we transmit a frame where the payload is the IP packet, and the frame type is 86DD. When a router receives a frame of this type, the IP packet is re-transmitted in another frame to the next computer in the path of the packet’s destination.

⁶By “IP”, we mean its latest version, **IPv6**. A legacy version of the protocol, **IPv4**, is still used, despite being released in 1981. IPv4 can only support about 3 billion computers. IPv6, launched in 2012, can support a virtually unlimited number of computers. As of 2020, a third of the Internet’s computers use IPv6.

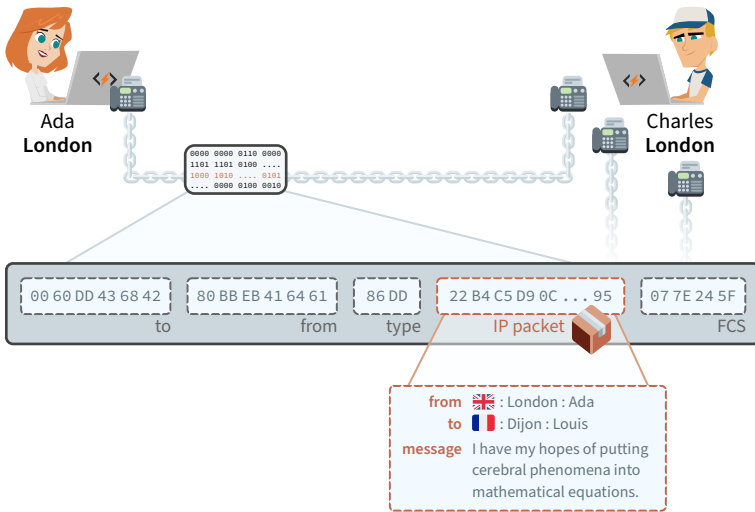


Figure 1.14 Ada sends an Ethernet frame to her router Charles containing an IP packet for Louis. The Ethernet frame therefore contains the physical address of Charles and the packet contains the location address of Louis. Charles will then forward the packet inside a new frame of his own containing the physical address of someone in France.

In order for IP packets to be forwarded around universally, everybody must agree on a standard for location addressing. We’ve seen how physical addresses are allocated by manufacturers according to the rules of Medium Access Control. Let’s now learn how the Internet Protocol does this for location addresses. We will then see how the Internet Protocol defines routing rules based on these addresses.

1.3 IP Addressing

The Internet Protocol sets the rules on how location addresses work—that’s why they’re called **IP addresses**. Computers can only send or receive IP packets after they get an IP address. Permission to use a group of IP addresses is first granted to an organization. These addresses are then assigned to computers which are directly or indirectly associated with the organization.

In order to explain how this process works, let's define what IP addresses are and how they're written.⁷ An IP address is a number 128 bits long.⁸ They're typically written in hex, with colons separating eight groups of four digits. This is Facebook server's IP address:

```
2a03:2880:f003:0c07:face:b00c:0000:0002
```

IP addresses can be shortened by omitting the leading zeros of any four-digit block:

```
2a03:2880:f003:c07:face:b00c::2
```

As with a postal address with country, city and street, IP addresses are hierarchical for routing to be possible. While the broadest part of a postal address is the country, the broadest part of an IP address is the **routing prefix**:

```
2a03:2880:f003:c07:face:b00c::2
```

routing prefix

The prefix shows up as the first digits of an IP address. Once an organization is granted such a prefix, it has the right to assign any IP address that begins with that prefix to its computers. The prefix has a variable length: organizations that have more computers to manage are granted shorter prefixes. Some organizations are even granted multiple prefixes.

For example, we know that all addresses that begin with the prefix **2a03:2880** are assigned to computers inside Facebook's network. Those that begin with **2c0f:fb50:4002** are in Google's network in Kenya. For its data center in Singapore, Google was granted the prefix **2404:6800**.

For routing purposes, the LANs and WANs that share the same prefix are organized in small networks called **subnets**. The digits

⁷We'll present IP addresses as defined in the latest version of IP. Legacy IPv4 addresses are still used. They are written as four groups of up to three digit decimal numbers, separated by dots, for example, 192.168.0.1.

⁸It takes 128 zeros and ones to write the number. This means it's a number between 0 and 340,282,366,920,938,463,374,607,431,768,211,456.

after the routing prefix and up to the middle of an IP address indicate in which subnet a computer can be found.

```
2a03:2880:f003:c07:face:b00c::2
```

└──────────┘
subnet

This means there's a network at Facebook where all computers have IP addresses that begin with **2a03:2880:f003:c07**. Together, the routing prefix and the subnet form the **network ID** of an IP address. The network ID is always 16 digits long (including omitted zeros). This means an organization with a longer routing prefix can have less subnets within it.

Finally, the next 16 digits of an IP address are called the **interface ID**, as they identify a specific network interface within a subnet. Many network administrators simply fill in this part of the IP address with the device's MAC address. These digits can be any number, as long as it's only used once per subnet.

```
2a03:2880:f003:c07:face:b00c::2
```

└──────────┘ └──────────┘
network ID interface ID

For this addressing system to work universally, there must be a mechanism to ensure no two organizations use the same routing prefix. As was the case for MAC addresses, engineers solved this through some international coordination.

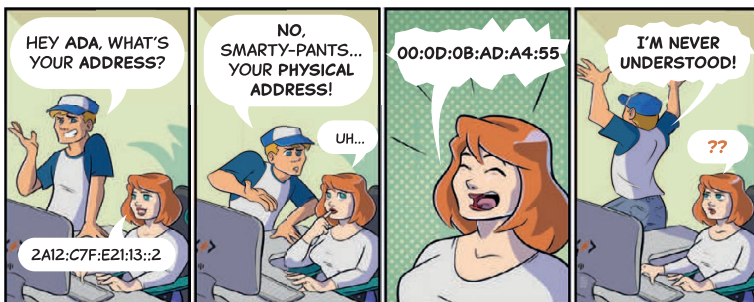


Figure 1.15 Don't ask your boss where she lives!

IANA

Engineers worldwide agreed that an American non-profit organization, the Internet Assigned Numbers Authority (**IANA**), decides who gets control over which IP routing prefixes. In practice, IANA delegates most of its power to five non-profit organizations called **Regional Internet Registries**, or **RIRs**. To do so, it allocates each RIR short hex combinations that they can use as the first digits of the routing prefixes they assign.

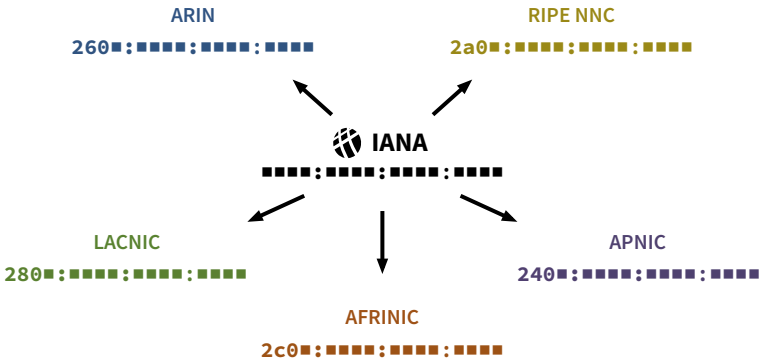


Figure 1.16 Examples of allocations to each RIR.

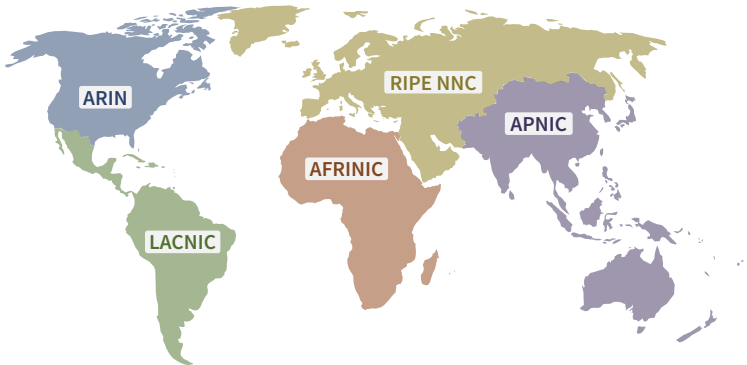


Figure 1.17 IANA delegates its IP addressing power geographically: each RIR is responsible for a different region.

To obtain a routing prefix for your organization, you must make a

request to the RIR of the region where your routers will be. That RIR will then assign you a prefix starting with one of their combinations of hex digits that IANA allocated them.

For example, Facebook, which has headquarters in Ireland, was granted its routing prefix by RIPE NCC. Likewise, the Swiss bank Credit Suisse has a Latin American branch that was granted a routing prefix by LACNIC:

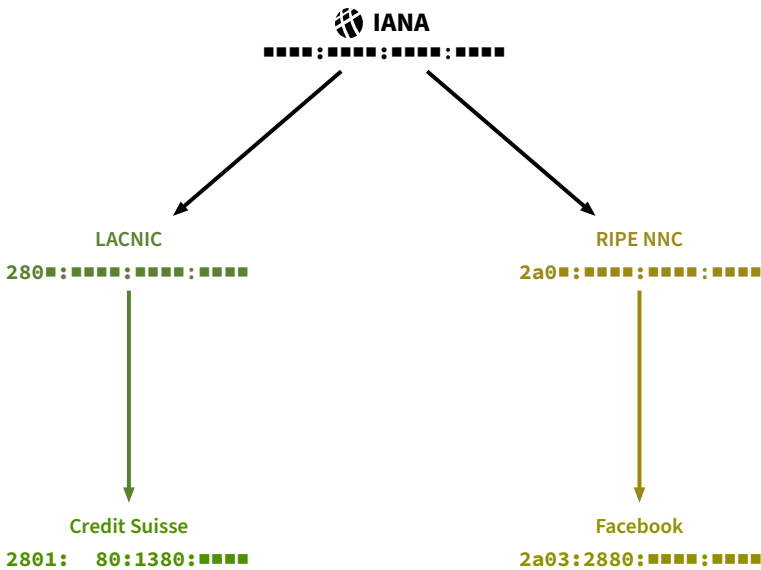


Figure 1.18 IP address allocation chain for two companies.

This means computers in the Latin American Credit Suisse branches may be assigned IP addresses as follows:

2801:80:1380:■■■■:____:____:____:____

Network administrators in the bank will assign a unique combination of hex digits to each of their subnets such that they fit in the remaining space of the network part ■■■■. Since each hex digit can have 16 different values, the bank has enough space for $16^4 = 65,536$ different subnets. Facebook, being a larger organization, was granted a prefix with room for over 4 billion subnets!

We've seen that network administrators can choose how the six-

teen blanks of the interface ID are to be filled for individual devices. Such devices may then send and receive IP packets to and from the Internet as long as their router has connectivity.

Internet Service Providers

Most individuals and small organizations don't deal directly with RIRs, nor do they maintain peering links to other computer networks. Instead, they buy Internet connectivity from specialized companies, which are called **Internet Service Providers (ISP)**. ISPs install routers close to their customers. That way, they can easily link one of their routers to a router in any customer's premises. They also allocate a routing prefix for each of their customers.

Let's see how it works in practice. In the United Kingdom, an ISP called Sky was granted the routing prefix **2a02:0c7f**. Sky operates in many British cities, so the prefix is divided between their regional bases. For instance, they assign **2a02:c7f:48** to their Milton Keynes network and **2a02:c7f:7e** to the one in Romford.⁹

Let's suppose Ada lives in Romford and wants to set up a network in her home. She has a desktop computer and a printer which she wants to connect using an Ethernet wire. She also wants her own WiFi network to connect her smartphone, tablet and laptop.

Ada hires Sky, and they link their Romford router to a router in her home. Sky assigns Ada's router a 14-digit routing prefix based on the one of their Romford base. Each network in Ada's home (wired and wireless) gets assigned a subnet, based on the routing prefix Sky allocated to Ada. Figure 1.19 on the next page shows the full IP address allocation path from IANA to each of Ada's devices.

Ada's router receives IP packets from several different computers, yet it's easy for her router to decide on which link to forward each packet it receives. Packets addressed to a computer in one of Ada's subnets can be directly delivered. All other IP packets it receives are forwarded through the link to the ISP.

⁹This information is public, you can look up the network location of any routing prefix. The practice is called **IP geolocation**, and it's how websites guess the country and city you browse from.

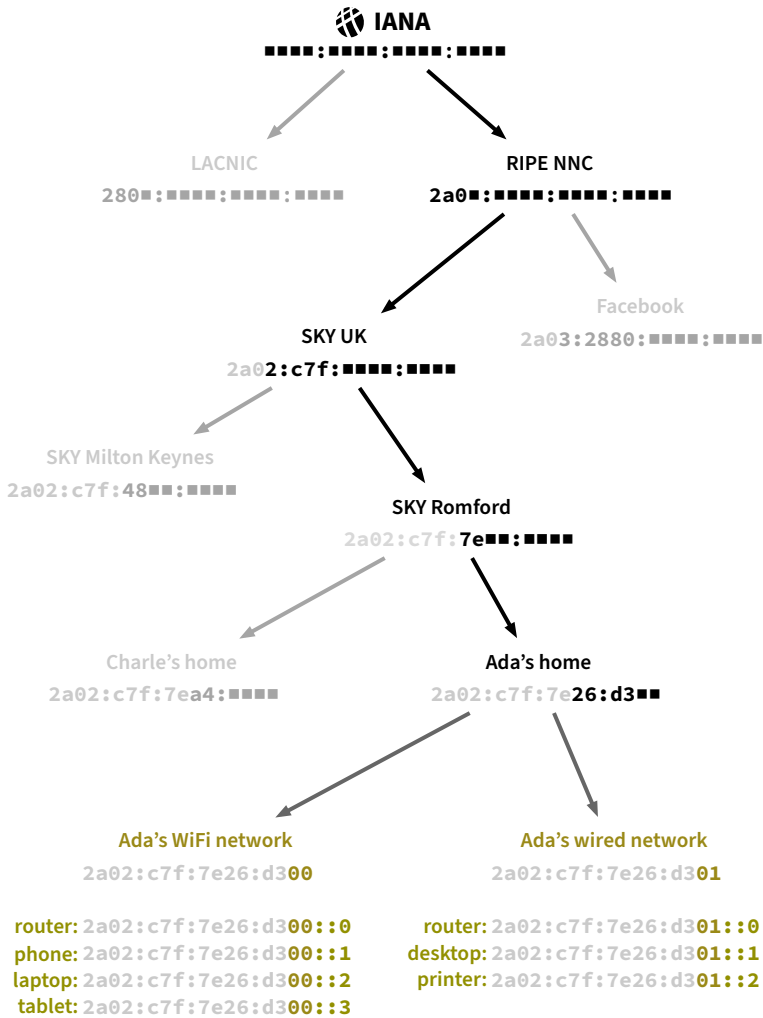


Figure 1.19 IP address allocations from IANA to Ada’s devices. Her router uses different subnets for her wireless and wired networks, and therefore has a different IP address for each.

For routers that don’t rely on an ISP, it’s not so easy: they obtain connectivity from links with several routers from multiple computer networks. But how do they decide on which link they should forward an IP packet? And even then, how can they be sure that they are forwarding it to a router closer to their final destination?

1.4 IP Routing

Suppose Ada wants to send a message to Facebook from her laptop. She will use the Internet Protocol, so she starts by crafting an IP packet that includes her own IP address, Facebook’s IP address, and her message as the payload. She then transmits the packet in a WiFi frame from her laptop to her home router:

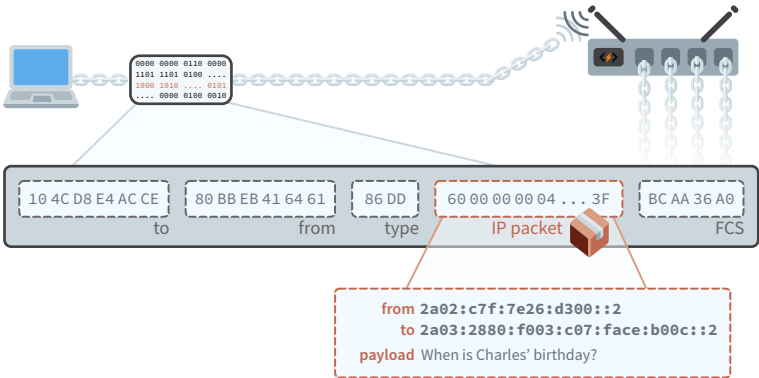


Figure 1.20 An IP packet transmitted over WiFi.¹⁰

Several routers, starting with the one at Ada’s home, retransmit the packet until it reaches Facebook. Along the way, each of those routers must choose in which direction the packet should “hop” to reach the next router. The last router will then make the packet “hop” towards its final destination computer.

Tables of Addresses

Routers choose the next hop of a packet based on its destination IP address. In order to do so, they are equipped with a table filled with IP addresses. Rows list possible addresses the router is configured to recognize. For each address, the table indicates which computer should be the next hop of a packet destined to that address. Every router has a unique table that reflects how the router is linked. For example, here is how Ada’s router is linked:

¹⁰We’ve included the fields of the WiFi frame which also exist in Ethernet frames. A WiFi frame has more fields, which were hidden for simplicity.

To purchase your copy of the full book, please visit

<http://code.energy/computer-science-unleashed>