**IOT communication protocols for extreme conditions at remote locations.**



Marijn Verschuren 05-05-2023

# Table of contents

[1. Table of contents 2](#_Toc1)

[3. Introduction 3](#_Toc2)

[4. HTTP 4](#_Toc3)

[4.1. TCP 4](#_Toc4)

[5. CoAP 8](#_Toc5)

[5.1. UDP 8](#_Toc6)

[6. Conclusion 9](#_Toc7)

[7. Bibliography 10](#_Toc8)

2. Table of figures

[Figure 1. OSI model 4](#_Toc1)

[Figure 2. IP packet 4](#_Toc2)

[Figure 3. TCP packet 5](#_Toc3)

[Figure 4. TCP handshake 6](#_Toc4)

[Figure 5. TCP data transfer 6](#_Toc5)

[Figure 6. TCP packet loss 7](#_Toc6)

[Figure 7. TCP packet timeout 7](#_Toc7)

# 3. Introduction

This paper aims to determine the best IOT protocol for remote glacier size monitoring to track the effects of climate change on water levels globally. The protocols that were chosen for this research are HTTP and CoAP because of their similarities and relative simplicity. Since the project revolves around a remote application low power operation is held in high regard, other important metrics include expandability and reliability, these points are crucial to a maintenance free deployment. This paper however will not speak on the physical construction and deployment of the sensors and will merely compare the chosen IOT protocols for this application.

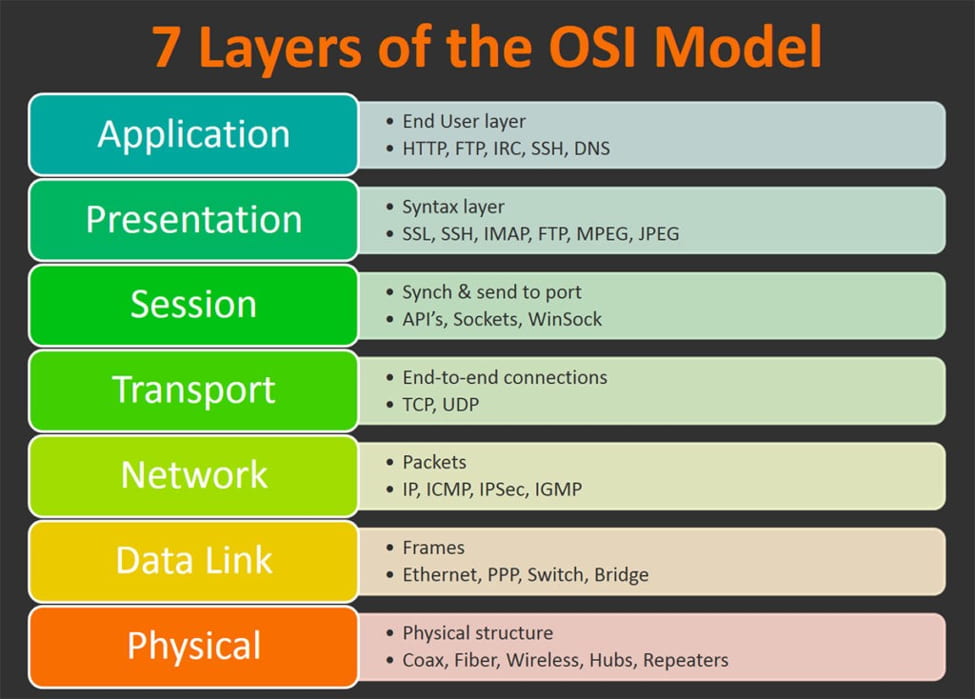
# 4. HTTP

HTTP (HyperText Transfer Protocol) is a widely used client-server internet protocol meant for hypertext, a form of text that can be used for things such as a web page or structuring of information. It features multiple request types that a client can issue the most used types are: GET, POST, DELETE and PUT. GET is obviously used to request resources from the server as DELETE is obvious to delete resources from it. POST and PUT are used to upload resources to the server but PUT is just a little different than POST because it is used to replace/update data. In return of a request the server will send a response with a response code ranging from 100 to 599, the first digit of this code represents the code type. The different code types are: Info (1xx), success (2xx), redirection (3xx), client error (4xx) and server error (5xx) *(MDN, 2023)*.

HTTP uses TCP (Transmission Control Protocol) as its transport layer which will ensure reliability and connection. In the image below the connection between HTTP and TCP is explained by showing what is built on top of what.

Figure 1. OSI model

*The-Physical-Layer-in-OSI-Model-Explained-thumbnail.jpg (JPEG Image, 975 × 699 pixels). (n.d.). Retrieved May 5, 2023, from https://shardeum.org/blog/wp-content/uploads/2022/09/The-Physical-Layer-in-OSI-Model-Explained-thumbnail.jpg*

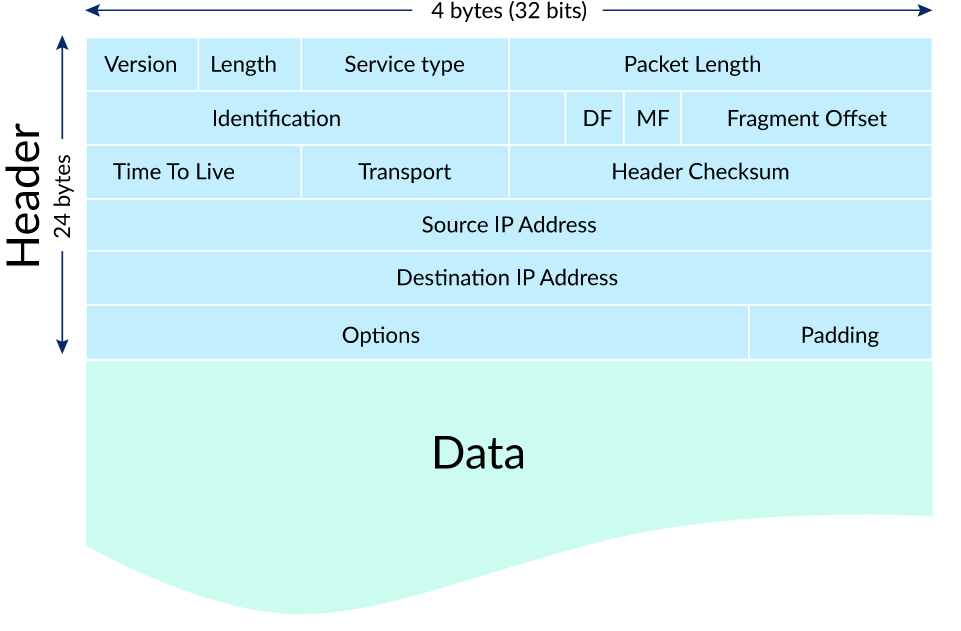


## 4.1. TCP

TCP is a transport protocol that is built on top of the IP network protocol (see Figure 1). The IP protocol works in packets that make sure that the packet is sent to the correct computer and that both parties know where the data came from, if it (the header) was received correctly and how large the data segment is among other things (see Figure 2 for an image of an IP packet).

Figure 2. IP packet

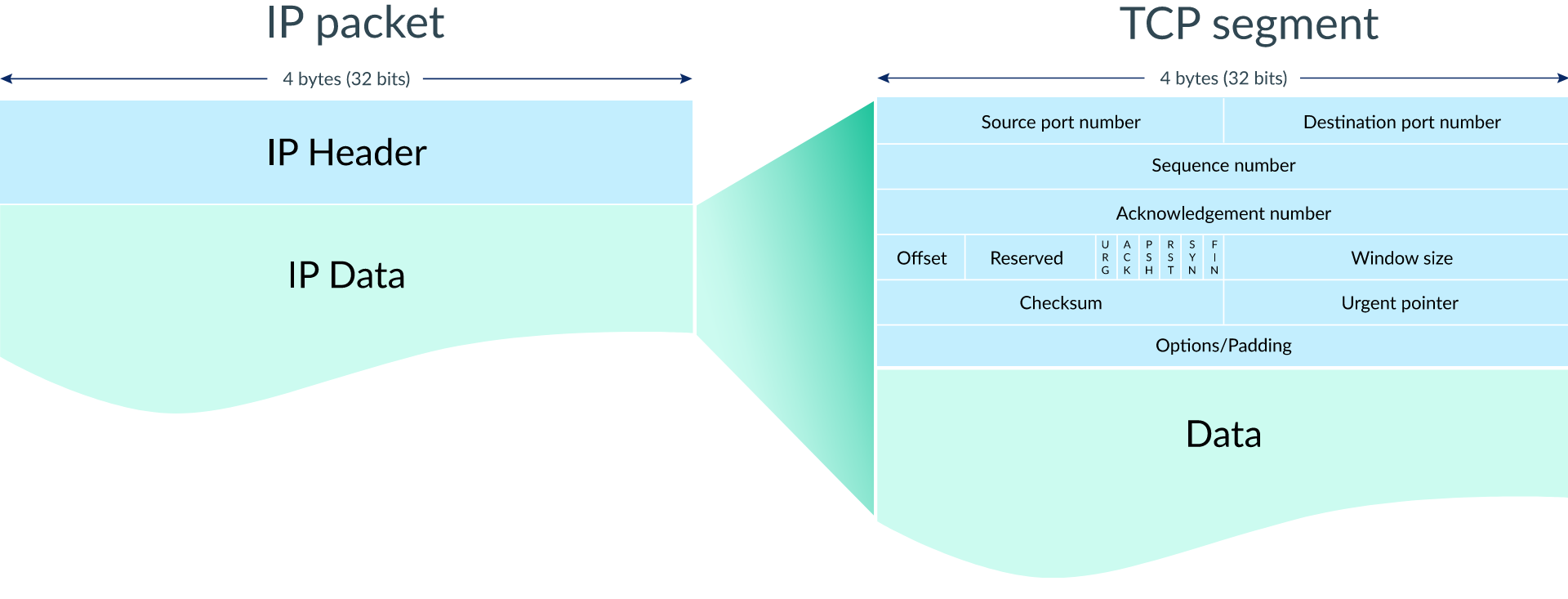
*IP packet*. (n.d.). Retrieved May 6, 2023, from <https://cdn.kastatic.org/ka-perseus-images/337190cba133e19ee9d8b5878453f915971a59cd.svg>



The TCP segment (header) is then placed inside the data section of the IP packet (along with the data) this can be seen in Figure 3. In the TCP segment you can find a checksum which is used to make sure the data is received correctly, there is also some additional data in the form of flags and sequence variables that are used to establish and hold a connection.

Figure 3. TCP packet

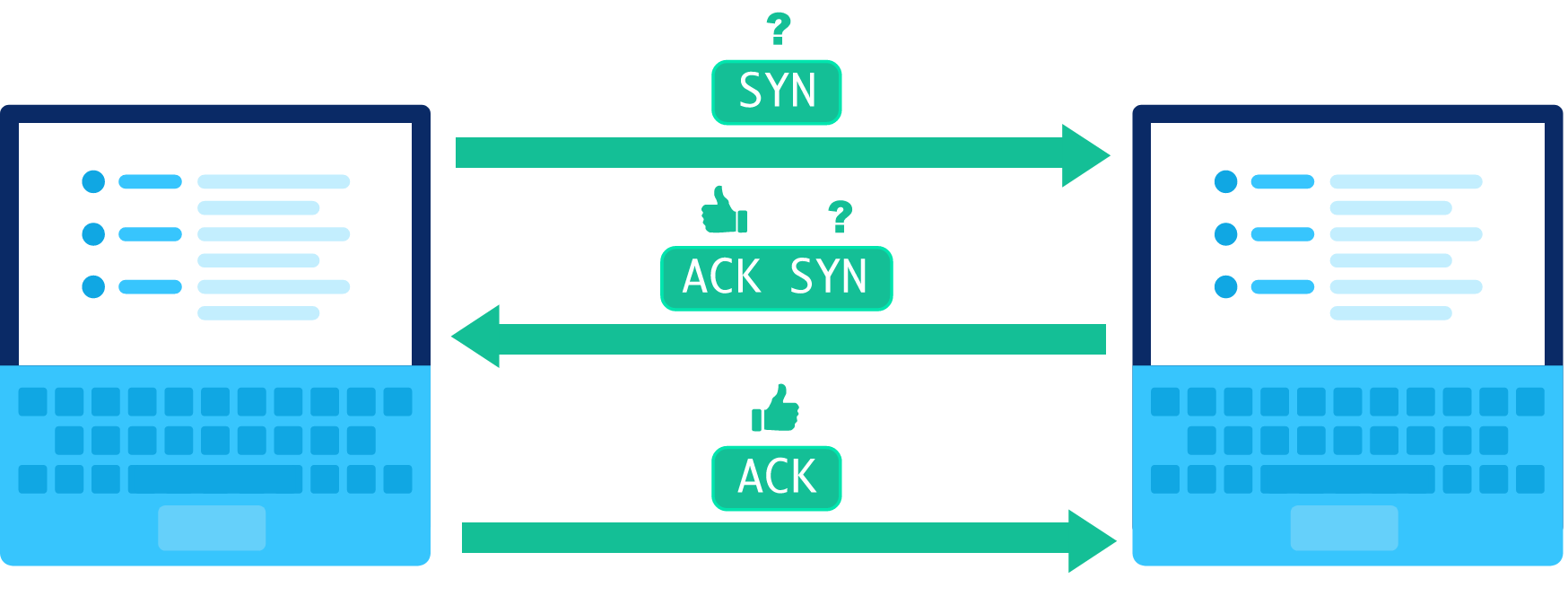
*TCP segment*. (n.d.). Retrieved May 6, 2023, from https://cdn.kastatic.org/ka-perseus-images/e5fdf560fdb40a1c0b3c3ce96f570e5f00fff161.svg



When initiating a connection the process shown in Figure 4 is used this is commonly known as the handshake. In this process the client will a sync request to the server (SYN bit set in header) to which the server will respond with a sync acknowledgment (SYN and ACK bit set in header), When this is received by the client an acknowledgment is sent and the connection is officially established. These handshake packets typically do not include data *(Khan Academy, n.d.).*

Figure 4. TCP handshake

*TCP handshake*. (n.d.). Retrieved May 6, 2023, from https://cdn.kastatic.org/ka-perseus-images/d09f9d37ff2a2deb21a8822f8c99ba6b86319f0b.svg



Now that the handshake is completed the data transfer can begin. As displayed in Figure 5 a sequence number is added to the header, this is set to the amount of bytes that were transferred before it so that the receiver can detect packet loss. Packet loss is detected by keeping track of the received bytes this is done by incrementing the acknowledgment number, now the incoming sequence number can be compared to the last acknowledgment number (these need to be equal). When packet loss is detected the packet is requested again by sending another acknowledgment with the expected sequence number as acknowledgment number (see Figure 6) *(Khan Academy, n.d.)*. Additionally a timeout is added so that when the sender does not receive an acknowledgment in time the packet is sent again (see Figure 7).

Figure 5. TCP data transfer

*TCP send packet*. (n.d.). Retrieved May 6, 2023, from <https://cdn.kastatic.org/ka-perseus-images/2cfc6b88b3b5c3a27386503d347524c2065a57d9.svg>

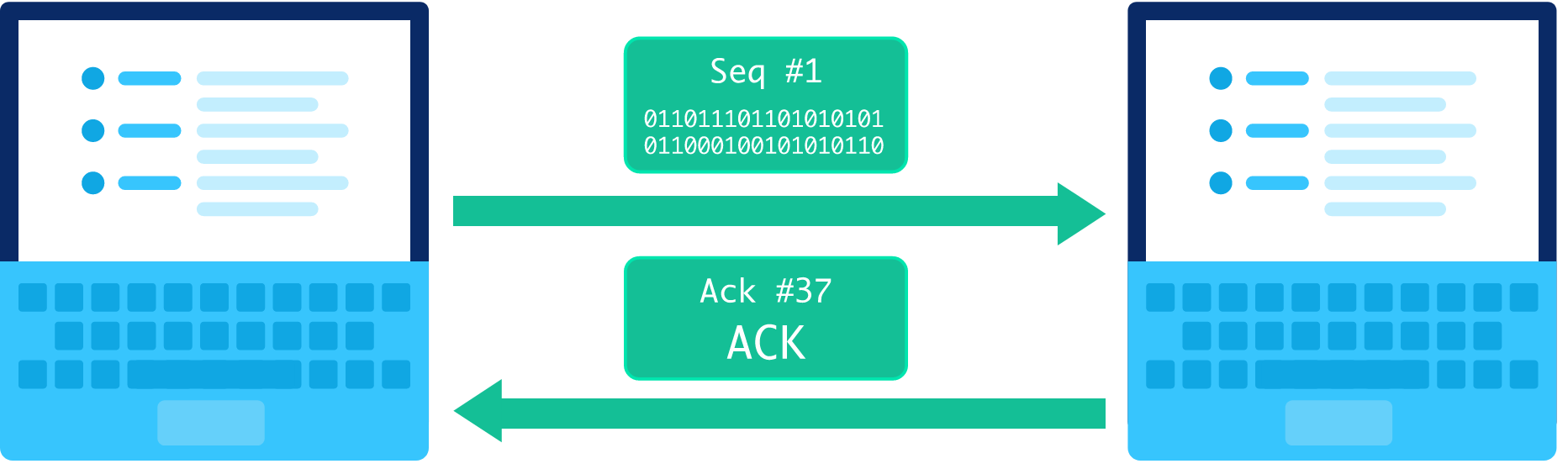


Figure 6. TCP packet loss

*TCP out of order packet*. (n.d.). Retrieved May 9, 2023, from <https://cdn.kastatic.org/ka-perseus-images/27f4fa1915c98689623e0ee224416c5290afc65a.svg>

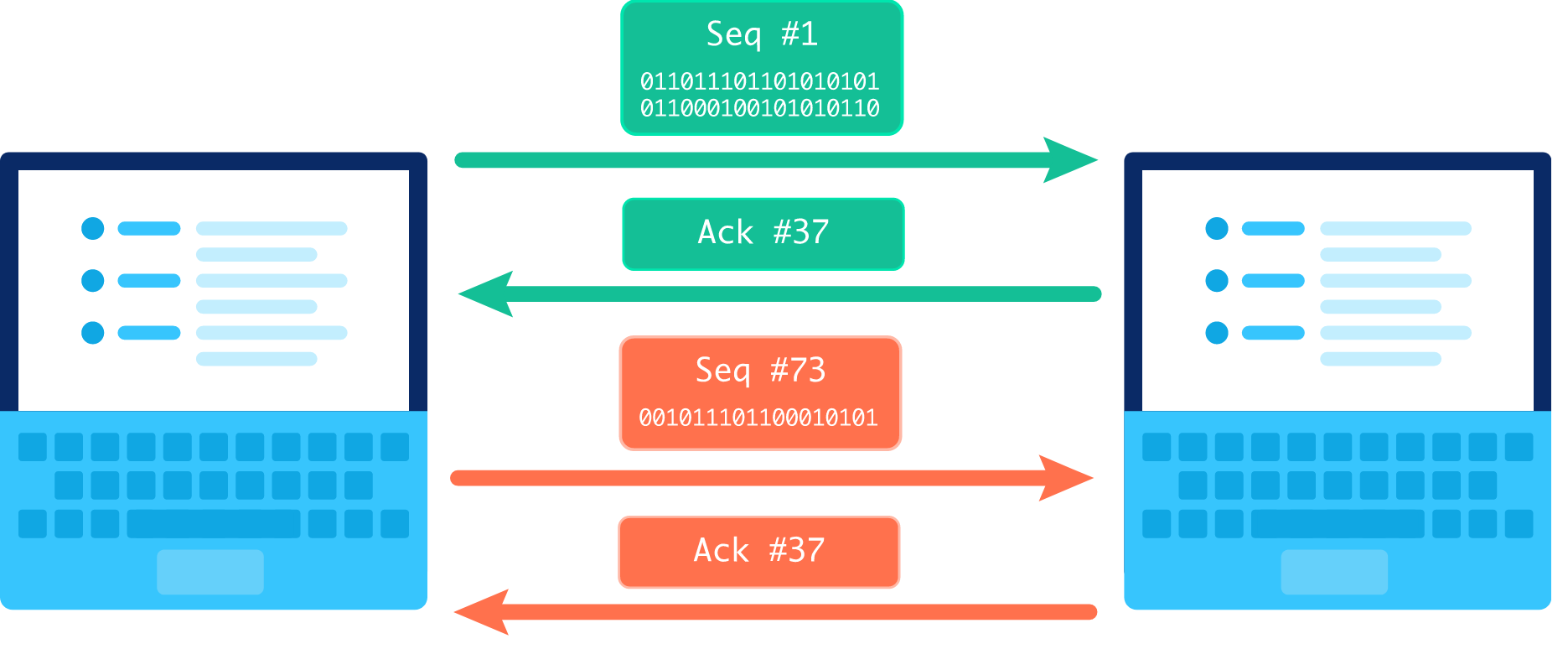
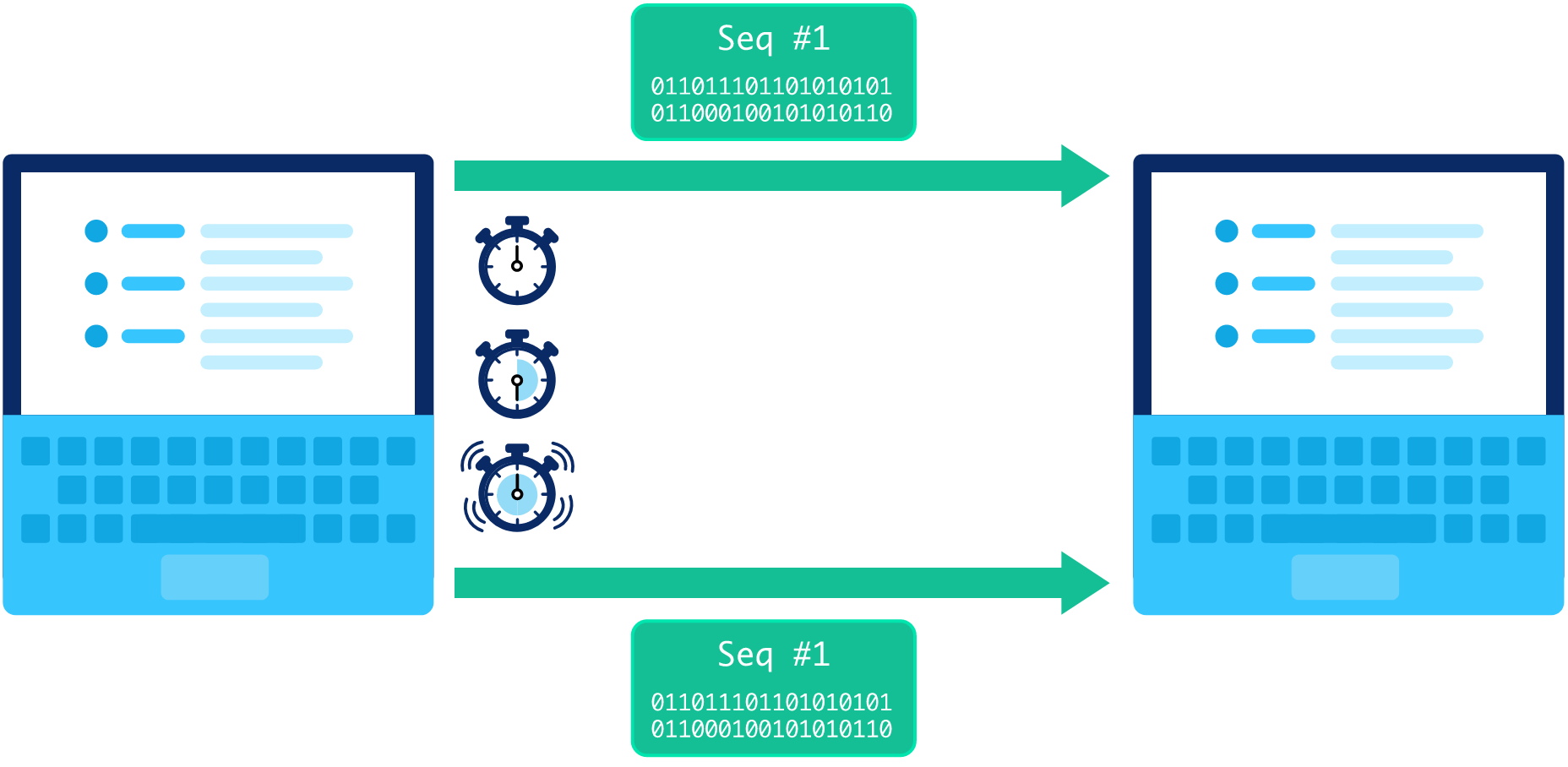


Figure 7. TCP packet timeout



*TCP timeout*. (n.d.). Retrieved May 6, 2023, from <https://cdn.kastatic.org/ka-perseus-images/b1017461d232cd46fa5b445f80e75568bf31c57c.svg>

Finally the connection has to be closed this is the exact same process as the handshake but with the FIN flag set instead of SYN.

# 5. CoAP

CoAP (Constrained Application Protocol) is a specialized internet protocol for lossy or low power networks that is designed to resemble HTTP.

## 5.1. UDP

# 6. Conclusion

# 7. Bibliography

*The-Physical-Layer-in-OSI-Model-Explained-thumbnail.jpg (JPEG Image, 975 × 699 pixels)*. (n.d.). Retrieved May 5, 2023, from <https://shardeum.org/blog/wp-content/uploads/2022/09/The-Physical-Layer-in-OSI-Model-Explained-thumbnail.jpg>

*TCP handshake*. (n.d.). Retrieved May 6, 2023, from <https://cdn.kastatic.org/ka-perseus-images/d09f9d37ff2a2deb21a8822f8c99ba6b86319f0b.svg>

*TCP segment*. (n.d.). Retrieved May 6, 2023, from <https://cdn.kastatic.org/ka-perseus-images/e5fdf560fdb40a1c0b3c3ce96f570e5f00fff161.svg>

*Transmission Control Protocol (TCP) (article) | Khan Academy*. (n.d.). Retrieved May 6, 2023, from https://www.khanacademy.org/\_render

*TCP header only copy*. (n.d.). Retrieved May 6, 2023, from https://cdn.kastatic.org/ka-perseus-images/ec71832edb1f2ff1d2ad12da494033ce2b25aafa.svg

*TCP send packet*. (n.d.). Retrieved May 6, 2023, from https://cdn.kastatic.org/ka-perseus-images/2cfc6b88b3b5c3a27386503d347524c2065a57d9.svg

*TCP timeout*. (n.d.). Retrieved May 6, 2023, from <https://cdn.kastatic.org/ka-perseus-images/b1017461d232cd46fa5b445f80e75568bf31c57c.svg>

*HTTP response status codes - HTTP | MDN*. (2023, April 10). https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

*TCP out of order packet*. (n.d.). Retrieved May 9, 2023, from https://cdn.kastatic.org/ka-perseus-images/27f4fa1915c98689623e0ee224416c5290afc65a.svg