TCP/IP unplugged

# Introduction

This unplugged activity has been designed to give learners a hands-on experience of the TCP/IP model. Learners will take on various roles of the internet to request a web page and an image file. They will use envelopes to simulate encapsulation as the data works its way through the layers and the network.

Detailed instructions are provided on the right-hand side of each simulation. These contain questions that you can ask the class to check their understanding and keep them involved with the unplugged activity. There are also shorter “steps” on the left-hand side as a quick prompt for when you are more familiar with the activity.

# You will need:

* 12 volunteers
* String, cut to lengths as required
* 8 envelopes for simulation one (or A4 paper folded in half to symbolise encapsulation)
* 12 envelopes for simulation two
* 3 envelopes for simulation three
* 3 pens
* 2 pairs of scissors
* A printed copy of the **image** on page 16 of this document
* A printed copy of the **web page** on page 17 of this document
* Several copies of the **network map** on page 14 of this document (as required)
* **Routing tables** for each router on page 15 of this document (you will need to cut this page into four, one piece for each routing table)
* A printed copy of the IP tables (cut into two)
* Optional: sticky labels
* Optional: sticky tape (sellotape)
* Optional: spare envelopes for if learners make a mistake

# Getting started

1. After selecting your **12 volunteers**, give each of them a defined role from this list (you could use sticky labels for each role):
   * 2× learners to act as the server (running the transport and internet layers). There are a lot of practical steps here that need to be shared.
   * Web service at port 80
   * File transfer service at port 21
   * 2× learners to act as the client PC (running the transport and internet layers). There are lots of practical steps here that need to be shared.
   * Web browser at port 40
   * FTP software at port 60
   * Client router
   * Server router
   * Blackpool router
   * Cambridge router
2. Ask learners to position themselves using the **network map** to create the network.
3. Use the **string** to simulate the links between devices. Each connection should have its own piece of string instead of having one long piece for the entire network. Note that dotted lines do not require any string as they are part of the same device.
4. Give each learner with a **router** role, their corresponding **routing table**.
5. Give the **web browser** a copy of the **IP address DNS table for the web browser**, one small piece of scrap paper, an envelope, and a pen.
6. Give the **FTP software** a copy of the **IP address DNS table for the FTP software**, one small piece of scrap paper, an envelope, and a pen.
7. Give the **client PC** and the **server** a stack of the remaining envelopes.
8. Give the **server** a pen and a pair of scissors.
9. Give the **web service at port 80** a copy of the website page.
10. Give the **FTP service at port 21** a copy of the image.
11. PRIOR TO THE LESSON: You will need to create a spare version of the ‘lost packet’ for simulation two.
    * Print a second copy of the image file
    * Take the second section of the image and place it in an envelope
    * On the envelope write ‘Segment: 2 of 3, Protocol: TCP Source port: 21, Destination port: 60’
    * Then place it in another envelope to form an IP packet. Write on the envelope ‘Source IP: 189.0.0.2 Destination IP: 192.168.0.2‘

## Simulation one Requesting a web page

For this activity the learners will use the HTTP protocol to request a web page from a server. The protocol used at the transport layer will be TCP, which checks for the number of packets received and requests new copies if any packets are lost in transit. This simulation will run smoothly and all packets will arrive safely at their destination. A walkthrough is provided for this simulation on the slides (from slide 20).

| **Steps** | **Instructions** |
| --- | --- |
| 1. **Explain** the map to your learners | * Use the map on slide 19 to help with your explanation or distribute copies of the map around your classroom as a guide. * Point out to learners that they will be familiar with this type of set up from the networks that they have used in Packet tracer. * Point to the learners that make up the home network. * Show them that data will travel through several routers that would be distributed around the world. * Point to the learners that make up the server’s network. This is where the website and the image file will be transmitted from upon request. |
| 1. **Enter the URL** in the **web browser** | Note: The activities in this step are carried out by the web browser, not the client PC.   * Tell the class that the user at the client PC would like to access the **RaspberryPi.org** homepage so they type this into the web browser. * The web browser has a record of this site being visited before and knows the IP address. * Ask the learner playing the ‘web browser’ to read out the IP address for RaspberryPi.org from their IP DNS table. * Ask the class**:**   + **Question:** What would happen if the browser didn’t know the IP address?   + **Answer:** It would request the IP address from the DNS server. |
| 3. Use the **application layer** to formalise the request | * Explain to the class that the web browser is operating in the **application layer**. * The web browser puts in a ‘Get’ request for the web page. * Ask the learner who is playing the web browser to write “Get RaspberryPi.org from 189.0.0.2” on their slip of scrap paper. * Ask the class:   + **Question:** What protocol is used to request a webpage from a web server?   + **Answer:** HTTP * Instruct the ‘web browser’ to add “Protocol: HTTP” to the slip of paper and then pass the request to the ‘client PC’. This is received by the **transport layer**. * Explain that the protocols that now handle the communications are part of the computer’s operating system. The collection of protocols is called the TCP/IP stack. |
| 4. Use the **transport layer** to add port numbers | * Explain to the class that the data (request) now needs to get ready for transport so it uses the **transport layer**. * The transport layer is not concerned with the data that it is transmitting, it just wants to know if it needs to be separated into smaller chunks plus one more thing. Ask the class:   + **Question:** What else does the transport layer do?   + **Answer:** It states the port number of the source and the receiver. * In our case, the request is quite small so it does not need to be divided up into chunks. * Instruct the ‘client PC’ to place the received data into an envelope, making a note of the destination IP address. The envelope is used to symbolise **encapsulation**. * They should then write the source and destination port addresses on the envelope, along with the protocol **TCP**. * Additionally they should write ‘1 of 1’ on the envelope to illustrate this is the first of a total of one envelopes.   “Protocol: TCP Source port: 40*(randomly chosen)* Destination port: 80 *(http server)*  1 of 1”  **Note:** The TCP protocol will be explained in more detail later on through the next simulation. You might wish to point out at this stage that the TCP protocol allows for the data to be sent reliably across the internet. |
| 5. Use the **internet layer** to add the IP addresses | * Explain that the **internet layer** is used to turn the TCP segments into **IP packets**. This involves adding the source and destination IP addresses. * Ask the ‘client PC’ to place the segment from the transport layer into another envelope (further **encapsulation**). * On this envelope, the ‘client PC’ writes the source and destination IP addresses:   “Source IP address: 192.168.0.2  Destination IP address: 189.0.0.2” |
| 6. The **fourth layer** | * Explain to the class that at this point, the IP packet would move to the fourth layer. Ask the class:   + **Question**: What is the name of the fourth layer?   + **Answer:** The link layer * Ask the class:   + **Question**: What is the role of the link layer?   + **Answer:** The link layer determines the **physical** location of the next node in the network. It also uses the protocol related to the technology being used for data transmission. E.g. the ethernet protocol when an ethernet cable is used. * Explain that to help simplify this demonstration, this layer will not be used right now. |
| 7. **Routing** the packet | * **Note:** As you pass the IP packets around the routers (learners), ask them to read out the address for the next hop from their routing tables. * Explain to the class that the IP packet will now make several ‘hops’ around the internet before it reaches its final destination. Each router in the network knows how to get to the final destination by knowing the ‘next hop’. This information is listed in its routing table. * The ‘client PC’ passes the IP packet to the local router. The local (client) router reads the destination IP address and finds its network in the routing table. The network for 189.0.0.2 is 189.0.0.0. * It looks at the ‘next hop’ column to know which network to send the IP packet to. * The ‘client router’ passes the IP packet to the ‘Blackpool router’. * The same thing happens again. The ‘Blackpool router’ reads the destination IP address and finds it on the routing table. It then sends it onto the next hop, which is the ‘Cambridge router’. * Again, the ‘Cambridge router’ checks the routing table and sends it to the ‘Server’s router’. * Explain that in the simplified network there was only one path to the destination. In a real network there will be hundreds of routers and many different paths. The routers work out the best path for your IP packets to take. |
| 8. I am the **network** you are looking for! | * The ‘Server router’ recognises the IP address as being on its own network and forwards the IP packet to the server with that IP address. * Instruct the ‘server’s router’ to pass the envelope to the ‘server’. |
| 9. Moving back up the layers to the **transport layer** | * Explain that the IP packet has reached its correct destination and is passed into the TCP/IP stack and now needs to move up the layers to be decoded and actioned. * The IP packet is received by the **internet layer** in the server. * Ask the ‘server’ to open the first envelope. This symbolises **decapsulation**. The next envelope inside tells the ‘server’ which port the data should be sent to and which protocol is being used – TCP port 80. * Ask the class:   + **Question:** Which layer is above the internet layer?   + **Answer:** The transport layer * The TCP envelope is now passed to the **transport layer** within the server. * Instruct the ‘server’ to open the TCP envelope. * The envelope contains a HTTP request, which needs to be passed up to the appropriate layer. |
| 10. Moving up again to the **application layer** | * Explain to the class that the HTTP message now needs to move up to the final layer to be decoded and actioned. * Ask the class:   + **Question:** What is the layer above the transport layer?   + **Answer:** The application layer * The application layer is part of the application process. * Instruct the ‘server’ to pass the HTTP request to the ‘web service’. * Ask the ‘web service’ to read the request. * It should say “Get RaspberryPi.org from 189.0.0.2”. |
| 11. Using the **application layer** to send the web page to the **transport layer** | * Explain to the class that the application is now used to retrieve the data for the requested web page. This data is then passed to the application layer to start the journey back.. * Ask the class:   + **Question:** Which protocol is responsible for retrieving web pages?   + **Answer:** HTTP * If you wish, you can point out here that in reality the web service might need to find various elements of the web page from other servers and databases to get the whole web page. The data would include the HTML code, the CSS code, and potentially images, animations, and videos. * **Note:** The web page doesn’t get cut up in the application layer as this is the responsibility of the transport layer. * Instruct the ‘web service’ at port 80 to hand the print out of the web page to the ‘server’. This is received by the **transport layer** in the server. |
| 12. Segmenting the file in the **transport layer** | * Remind the class that we are now operating in the **transport layer**. * Ask the class:   + **Question:** What is the responsibility of the transport layer?   + **Answer:** To segment the data if it is too large to send in one go and to add the port address of the source and receiver. * Explain to the class that this web page is too large to be sent in one go and needs to be separated into segments. * Instruct the ‘server’ to cut the webpage into two segments using the cutting guides on the print out. * Explain to the class that each segment now needs to be identifiable at the receiving end. We are using the **TCP** protocol for sending this data. This means that it needs to keep track of how many packets are sent and the order in which they should be put back together. * Instruct the ‘server’ to place the top segment into an envelope. * On the envelope the ‘server’ should write:   “Segment: 1 of 2, Protocol: TCP Source’s port: 80, Destination port: 40” *(note the port source and destination have been swapped)*   * Continue this pattern with the other segment. |
| 13. Moving back down to the **internet layer** | * Explain to the class that the data is then passed to the **internet layer**. This is still carried out by the server. * Ask the class:   + **Question:** What does the internet layer do?   + **Answer:** It takes the segments and wraps them up as IP packets. Each IP packet shows the source and receiver's IP address. * Remind the learners that the rest of the internet is not concerned with the data that is being sent across the network. It just wants to know where the IP packets need to go. * Instruct the ‘server’ to take each segment and place it into another envelope. There should be two new envelopes for the two segments. * On each envelope, instruct the ‘server’ to write the source and destination IP address:   “Source’s IP address: 189.0.0.2  Destination IP address: 192.168.0.2” |
| 14. Routing the **packets** | * Follow the same routine with the packets as you did in step 7. A packet reaches a router, it checks the routing table, and moves it along. Keep this going until all packets have reached the ‘client PC’. * Remember to ask the learners to read out the ‘next hop’ address to allow all learners to visualise what is happening. * Remind the learners that the client router recognises the IP address as being on its own network and passes it along to the client PC. * **Note:** The routers are only concerned with the destination address of the IP packets. They do not look into any of the envelopes so could be carrying any type of data. |
| 15. Moving back up the layers again to the **transport layer** | * The **internet layer** has now done its job and the packages are at the correct destination. * The IP packets are unwrapped and sent to the **transport layer**. * The **transport layer** now has two segments; each with the segment number, number of segments, and port information. * Explain to the learners that the protocol being used here is the **TCP** protocol. This protocol is a reliable way to send data because it checks if all of the segments have arrived. If one is missing then it waits for a certain period of time before requesting another copy of that segment. * Both segments have arrived safely so they are reconstructed into the complete web page. * Instruct the ‘client PC’ to unwrap each of the envelopes and stick the contents back together. * The transport layer in the client PC has now rebuilt the full HTTP reply from the destination port and can match this reply to the request that was initially sent out. * If you like, you can add in here that there may be many ports used by the web browser. For example, there will be different ports for each tab that the client has open. The destination port is used to match the outgoing request to the correct reply, ensuring that the data goes to the correct tab i.e. the correct web browser process. * Ask the class:   + **Question:** Which layer is used next?   + **Answer:** The application layer * Instruct the ‘client PC’ to send the full HTTP reply to the ‘web browser at port 40’. |
| 16. Building the web page at the **application layer** | * Point out to the learners that the data would not be sent as an image like the example being used here. It would be HTML code that would be decoded by the application layer. * The full HTTP reply is received by the ‘web browser’. Instruct the ‘web browser’ to display the page to all. * Explain that the web page will contain static HTTP information as well as further links to images and potentially other data sources. Separate requests will be made for each of these resources. |

## Simulation two Requesting an image file

This simulation works in a similar way to simulation one. However, the user now requests an image file using the FTP protocol rather than a website that uses the HTTP protocol. The transport layer will still use TCP, but this time, a packet will be lost. This simulation demonstrates how the TCP protocol is used to detect a lost packet and request a new copy.

| 1. **Request** the image file using FTP | * Explain to the class that a user can request files through FTP via a web browser or via FTP software. In this scenario, the user is using FTP software. * Ask the class:   + **Question:** What does FTP stand for?   + **Answer:** File Transfer Protocol * Ask the class:   + **Question:** What is the role and function of FTP?   + **Answer:** It is a protocol that deals with uploading and downloading files from a server. Users can often access a directory of files and choose what to download and where to upload. * The user at the client PC requests the file ‘**flamingo.jpg**’ from **ftp.raspberrypi.org**. * Ask the learner playing the ‘FTP software’ to read out the IP address for ftp.raspberrypi.org from their IP table. * Again, you can reiterate that if this was not known then the DNS would be used to find the address. |
| --- | --- |
| 1. Use the **application layer** to formalise the request | **Note:** The activities in this step are carried out by the FTP software, not the client PC.   * Explain to the class that the FTP software is operating in the application layer. * Instruct the ‘FTP software’ to write a get request on a small piece of paper. The request should read:   “Get flamingo.jpg from ftp.raspberrypi.org at 189.0.0.2 Protocol: FTP”   * Now instruct the ‘FTP software’ to place this into an envelope and write FTP and the IP address on the front of it. * The ‘FTP software’ should now pass this envelope to the ‘client PC’. |
| 3. Moving to the **transport layer** | * The ‘client PC’ receives the envelope at the **transport layer**. * Ask the learners to describe what will happen next at the **transport layer**.   + It checks if the package is too large (it isn’t).   + It places it in an envelope and writes on the protocol, the source port, and the receiver’s port.   “Protocol: **FTP**, Source port: **60**, Receiver port: **21**” |
| 4. Moving to the **internet layer** | * The segment is received by the **internet layer**. * Ask the learners to describe what will happen next at the **internet layer**.   + It is placed in an envelope, this creates an IP packet.   + The source IP and receiver IP is written on the envelope.   “Source IP: 192.168.0.2  Receiver IP: 189.0.0.2” |
| 5. **Routing** the packet | * Just as with simulation one, the IP packet is routed through the network using the routing tables. * Ask the learners to talk you through this part to check their recall. |
| 6. Your package has reached its **destination**! | * Just like simulation one, the IP packet arrives at the server’s router and is then passed to the server. * The IP packet is then unwrapped by the ‘server’ to reveal the port number that needs to deal with the request. * The port number takes the segment to the FTP service, which opens the final envelope and reveals the request. * Again, you can ask learners to talk you through this part to check their recall. Remember to ask them about the layers that are being used and the roles that each layer plays. |
| 7. Back down the **layers** | * Again, just as with simulation one, the **application layer** reads the request. It then actions the request and sends the data to the transport layer. * The **transport layer** then segments the data ready for transport. * The **internet layer** then creates the IP packets and sends them back through the internet to the destination address. |
| 8. **Destroying** a packet! | * As the learners start to pass the packets through the network using the routing tables, your job is to get rid of a packet. * Make sure that you get rid of packet 2 of 3. * You can use your acting skills here to make this quite a dramatic moment and take the packet from one of the routers, rip it up, and throw it in the bin. * Instruct the learners to carry on sending their IP packets. * Explain that the router would not notice that the packet has been lost because that is not the router’s job. |
| 9. Packet **lost** | * When the other two packets arrive at the client PC, the transport layer should notice that packet 2 of 3 is missing. * Ask the class:   + **Question:** What should the transport layer do now? It doesn’t have the correct packets to send to the application layer!   + **Answer:** Make a request for the 2nd packet. |
| 10. **Replacing** the lost packet | * You should have already prepared segment 2 of 3 as a backup for this scenario (listed in the preparation section). * Explain to the learners that at this point, the transport layer would send a request for packet 2 of 3. The request would be sent in the same way as all of the other packets. When it reaches the transport layer of the server, the server will resend the missing packet. * Give the ‘missing packet’ to the ‘server’ and ask them to send it back through the internet.   Note: This part **isn’t simulated on the slides** because it will be pretty clear what is happening at this point. |
| 11. Image **complete** | * Just like with simulation one, the segments are packaged together and sent to the application layer. The FTP software builds the image before presenting it back to the client PC. |
| 12. TCP vs UDP | * Use this moment to question learners about TCP and introduce them to UDP. * Ask the class:   + **Question:** How long did it take to notice the packet was missing, request a new packet, and receive the packet?   + **Answer:** Quite some time, maybe a few minutes. * Explain to the learners that in reality this would probably take a few seconds or perhaps less than that. Explain that if you are waiting for a web page to load then a few seconds is an acceptable time to wait. * Ask the class:   + **Question:** What if you were on a video call? Would those few seconds matter? If the sound and video cut out completely for 2 seconds every few seconds would that become annoying?   + **Answer:** Yes, it probably would! (You also might get responses like, it does cut out though sometimes?) * Explain to the class that the TCP protocol is concerned with reliability. It ensures that all packets arrive and that the data is intact. This is perfect for web pages, emails, and documents because they must arrive as one whole. * However, video calls work differently. It doesn’t matter if the sound cuts out for a short time or the video becomes blurry. With video calls and voice calls, we are more concerned with speed so that we can have real-time conversations. * For this type of network communication, the UDP protocol is used instead of TCP. The UDP protocol will just send all of the packets in turn, but doesn’t request any lost packets. The application ignores them or tries to fill in the gaps where it can. |

## Simulation three The link layer

For this activity the learners will be given a brief demonstration of how the link layer provides another layer of encapsulation. The protocols used in the link layer depend on the technology being used to get the packet from one node to another.

| 1. **Reposition** the learners | * For this demonstration you only need the client PC, the client router, and the Blackpool router. Ask the other learners to sit down. |
| --- | --- |
| 1. Prepare an **IP packet** | * Take a spare envelope and write the source IP address and the receiver IP address onto it. |
| 1. Using the **link layer** | * Instruct the ‘client PC’ to place the IP packet into another envelope. * Explain to the class that the **link layer** deals with the physical aspect of transmitting the data. The protocol that is used will depend on the type of technology being used to transmit the data. The IP packets are now ‘wrapped’ in a **frame**. * Explain that the technology used to connect the client PC to the client’s router is WiFi. * Instruct the ‘client PC’ to write “Protocol: IP” onto their envelope, because these WiFi frames are carrying IP packets * Ask the class:   + **Question:** Can you remember what other job the link layer does?   + **Answer:** It states the MAC address for the next node in the network. * Explain that in this case, the next node is the client router. * The client PC has a method to obtain the MAC address of the client router, but for now, use the MAC address on the slides. * Instruct the ‘client PC’ to add the MAC destination address, which is 2. * The client PC also adds its own MAC address (1) as the source MAC address. * Next, instruct the ‘client PC’ to pass the envelope to the ‘client router’. |
| 1. The **next node** | * The ‘client router’ receives the envelope (frame). It knows the envelope was meant for it because the destination MAC address matches its MAC address. * Instruct the ‘client router’ to remove the envelope and examine the IP packet within. * This is the **link layer** passing the contents of the frame up to the **internet layer**. * By comparing the destination IP address with its routing table, the client router can decide which way to send the outgoing IP packet. The next hop is to the Blackpool router. * The technology used to go from the ‘client router’ to the ‘Blackpool router’ is an ethernet cable. * Add a new envelope (frame). * Instruct the ‘client router’ to write “Protocol: IP” on the envelope. * This is the process of the IP packet being passed from the **internet layer** down to the **link layer**. * Again the client router has a method of obtaining the MAC address of the Blackpool router, but for now use the number 3. * Ask them to add the MAC destination address of the Blackpool router. e.g. ‘3’. They should also include the current MAC address (2) as the source MAC address. * They then pass this envelope to the Blackpool router. * Explain that between the client and Blackpool router there may be many other nodes connected. If so, a switch is used to interconnect all the devices in the ethernet network. The switch will use the destination MAC addresses to switch the ethernet frame to the correct destination. * Explain to the class that the **link layer** does this at every node that it comes across on the network. The outer **frame** is discarded and a new **frame** is added depending on the technology used for the next connection. |

## 

# Network map



### Client router – routing table Blackpool router – routing table

| Network | Next hop |  | Network | Next hop |
| --- | --- | --- | --- | --- |
| 189.0.0.0 | 178.43.0.0 |  | 192.168.0.0 | 178.43.0.0 |
| 192.168.0.0 | -- |  | 189.0.0.0 | 58.23.0.0 |
| 58.23.0.0 | 173.43.0.0 |  | 162.15.0.0 | 58.23.0.0 |
| 162.15.0.0 | 178.43.0.0 |  |  |  |

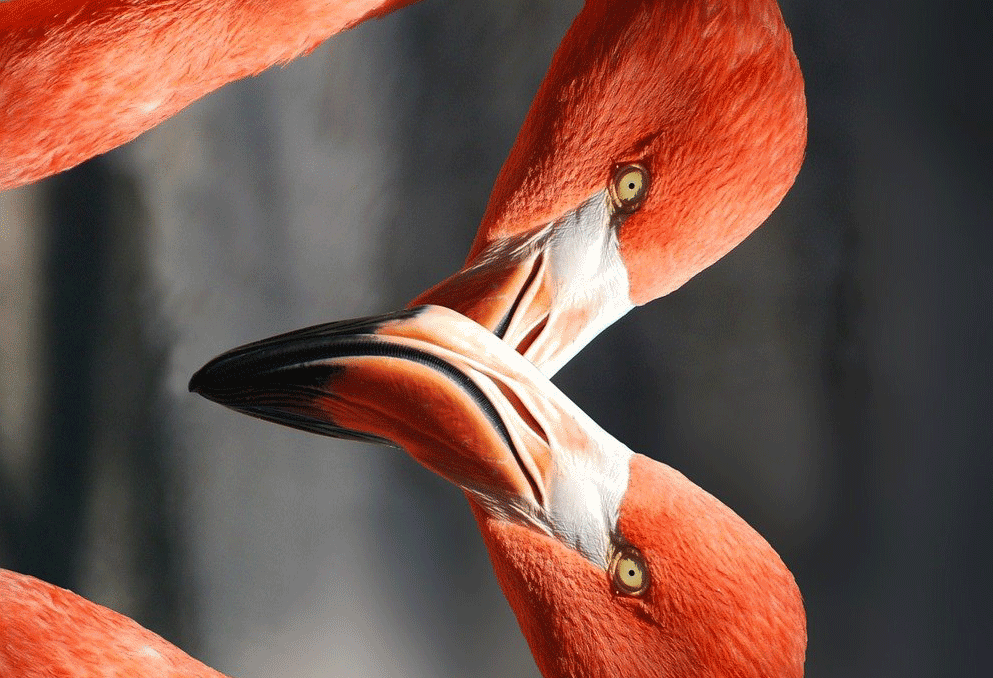
### Cambridge router – routing table Server router – routing table

| Network | Next hop |  | Network | Next hop |
| --- | --- | --- | --- | --- |
| 192.168.0.0 | 58.23.0.0 |  | 192.168.0.0 | 162.15.0.0 |
| 178.43.0.0 | 58.23.0.0 |  | 58.23.0.0 | 162.15.0.0 |
| 189.0.0.0 | 162.15.0.0 |  | 178.43.0.0 | 162.15.0.0 |
|  |  |  | 189.0.0.0 | -- |

✀ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -



✀ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -



✀ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

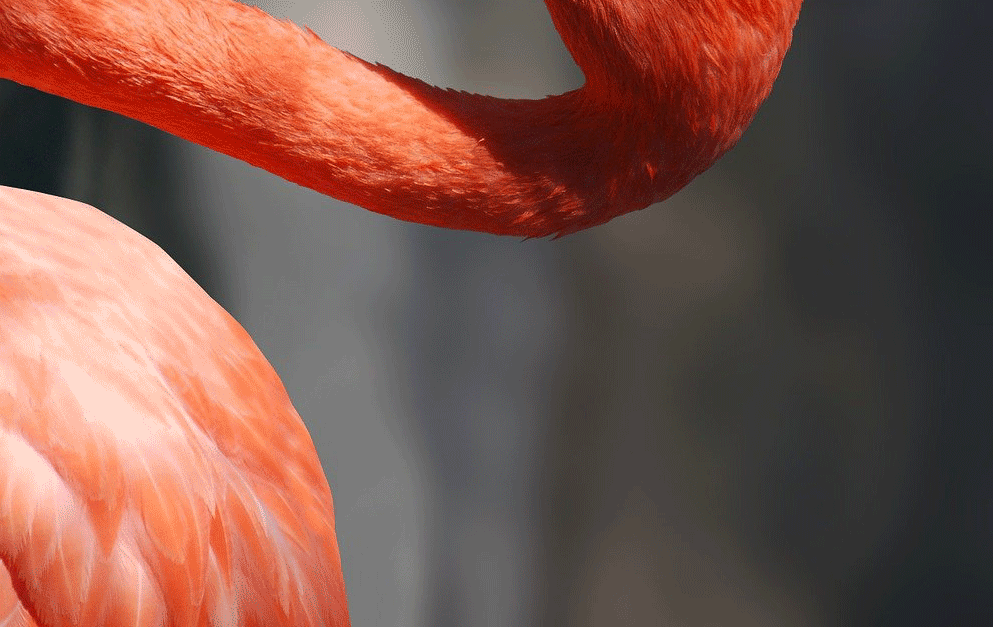
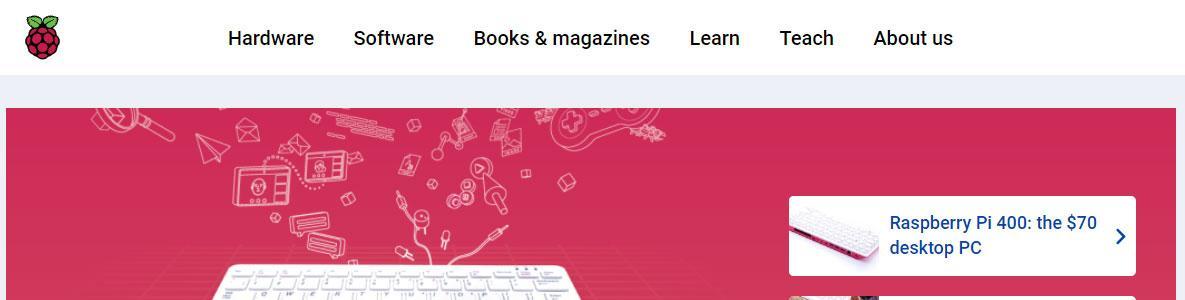


Image source: <https://pixabay.com/photos/flamingo-valentine-heart-600205/>

✀ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -



✀ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -



✀ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Image/website source: Raspberrypi.org

## IP address DNS table for the web browser

| **Website address** | **IP address** |
| --- | --- |
| www.google.com | 8.8.8.8 |
| www.bbc.co.uk | 212.58.226.75 |
| www.raspberypi.org | 189.0.0.2 |
| www.yahoo.com | 203.32.0.1 |

✀ - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

## IP address DNS table for the FTP software

| **Website address** | **IP address** |
| --- | --- |
| ftp.raspberrypi.org | 189.0.0.2 |
| ftp.images.com | 172.0.0.5 |
| ftp.freevideos.com | 89.21.0.38 |
| ftp.vectorstore.com | 181.1.0.123 |