|  |
| --- |
| **Network Fundamentals Tutorial 3 – Network Models** |

**Q1.** Consider the following table for the OSI model and the TCP/IP model:

|  |  |  |
| --- | --- | --- |
| ***OSI Model*** | ***TCP/IP Model*** | ***Protocols*** |
|  |  |  |
|  |
|  |
| Transport | Transport |  |
|  |  |  |
|  |  |  |
|  |

(a) Fill in the different layers in each model and identify the following protocols to the last column:

* TCP, UDP
* DHCP, DNS, FTP, HTTP
* ICMP, IP (i.e. IPv4, IPv6)
* BOOTP, IMAP, POP, SMTP
* ATM, Ethernet, Frame Relay, PPP, WLAN

(b) Identify the OSI layer that performs each of the following functions:

* Maintains data frames \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Performs path determination and logical addressing \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Performs encoding/decoding for binary transmission \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Provides data representation and encryption \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Provides end-to-end connections and reliability \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Organizes dialogue and manages data exchange \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Maintains process-to-process communications \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(c) Identify the TCP/IP layer that performs each of the following functions:

* Controls the hardware devices and media \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Determines the best path through the network \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Represents data to the user plus encoding and dialog control \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Supports communication between diverse devices \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Q2.** In addition to identifying the source and destination, network protocols need to define the details of how a message is transmitted across a network.

1. Match each **protocol requirement** to the appropriate **descriptions** on the right.

|  |  |  |  |
| --- | --- | --- | --- |
| Message sizing | **\*** | **\*** | Breaks up a long message into smaller pieces |
| Message encoding | **\*** | **\*** | Places one message format inside another message format |
| Message timing | **\*** | **\*** | Manages access method, flow control, and response timeout |
| Message encapsulation | **\*** | **\*** | Converts information into another acceptable form for transmission |
| Message delivery options | **\*** | **\*** | Sends out message to just an individual or to a group of people or all people at the same time |

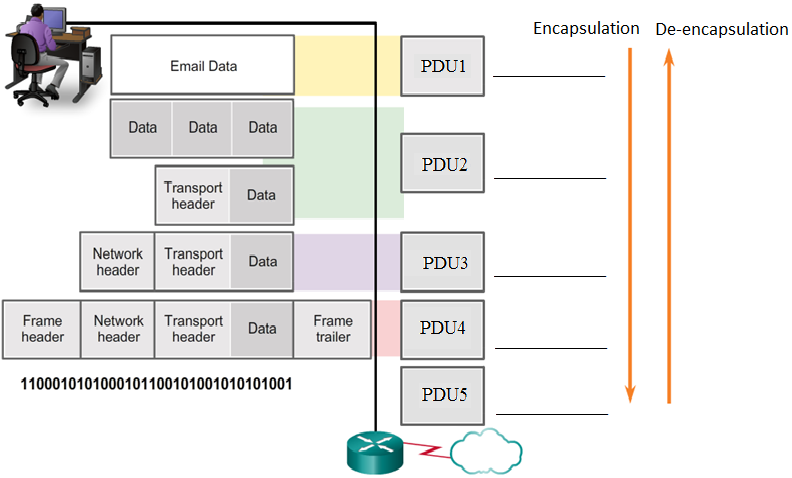
(b) Match each **message timing** to its appropriate **purpose**.

|  |  |  |  |
| --- | --- | --- | --- |
| Flow control | **\*** | **\*** | Determines when to begin sending messages |
| Access method | **\*** | **\*** | Ensure that packets are not dropped because too much data is being sent too quickly |
| Response timeout | **\*** | **\*** | Specifies how long to wait for responses and action to take if a response timeout occur |

(c)Match the **communication type** on the right to the appropriate **message** **delivery option**.

|  |  |  |  |
| --- | --- | --- | --- |
| Broadcast | **\*** | **\*** | One-to-one |
| Multicast | **\*** | **\*** | One-to-many |
| Unicast | **\*** | **\*** | One-to-all |

**Q3.** The form that a piece of data takes at a network layer is called a Protocol Data Unit (PDU).



1. Fill in the PDU names PDU1 to PDU5 at the various network layers.

(b) The sequence of PDU processed during the encapsulation of the email:

Data 🡪 \_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_

(c) The sequence is reversed in the de-encapsulation process as:

\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_ 🡪 Data

**Q4.** Consider the address settings for the devices in the LANs linked serially below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | **Logical**  **Address** | **MAC**  **Address** | | **HostA** | 192.168.1.110 | AAAA.AAAA.AAAA | | **HostB** | 172.16.2.99 | BBBB.BBBB.BBBB | | **ServerB** | 172.16.1.99 | BBBB.1234.5678 | | **RouterA-G0** | 192.168.1.1 | AAAA.CCCC.0000 | | **RouterA-S0** | 200.1.2.17 | AAAA.EEEE.0000 | | **RouterB-G1** | 172.16.1.1 | BBBB.CCCC.1111 | | **RouterB-S1** | 200.1.2.18 | BBBB.EEEE.1111 | |  |

1. **ServerB sending a packet to HostB**

For local communication, the following Ethernet frame containing the packet will be sent:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination  MAC Address | Source  MAC Address | Source  IP Address | Destination  IP Address | Packet  Data | Frame  Trailer |
| **A1** | **A2** | **A3** | **A4** | Data | Trailer |

Write down the addresses below:

A1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ A3: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ A4: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **ServerB sending a packet to HostA**

For remote communication, the following frame containing the packet will be sent **from ServerB to RouterB-G1**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination  MAC Address | Source  MAC Address | Source  IP Address | Destination  IP Address | Packet  Data | Frame  Trailer |
| **B1** | **B2** | **B3** | **B4** | Data | Trailer |

Write down the addresses below:

B1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ B3: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

B2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ B4: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

When RouterB receives the frame, RouterB will de-capsulate the packet and find that it should be sent out via exit S1. Finally RouterA will receive the packet and RouterA will encapsulate the packet into the frame below for HostA:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Destination  MAC Address | Source  MAC Address | Source  IP Address | Destination  IP Address | Packet  Data | Frame  Trailer |
| **B5** | **B6** | **B7** | **B8** | Data | Trailer |

Write down the addresses below:

B5: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ B7: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

B6: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ B8: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_