# Using the network diagram below, complete the virtual circuit tables (next page) for all switches after the following connections have been established.

* 1. Add rows to each virtual circuit table as needed
  2. Assume that the sequence of connections is cumulative; that is, the first connection is still up when the second connection is established and so on.
  3. Assume that the VCI assignment always picks the lowest unused VCI on each link, starting with zero.

Connections to be made:

1. Host A connects to Host E
2. Host B connects to Host E
3. Host C connects to Host F
4. Host F connects to Host A
5. Host E connects to Host C
6. Host D connects to Host B

## A picture containing application Description automatically generated



## B1 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 1 | B | 0 | A | 0 |
| 2 | N/A | N/A | A | 1 |
| 4 | A | 2 | B | 1 |
| 6 | A | 3 | N/A | N/A |

B2 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 6 | N/A | N/A | A | 0 |

B3 Virtual Circuit Table



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 3 | N/A | N/A | A | 0 |
| 5 | A | 1 | N/A | N/A |

B4 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 3 | A | 0 | N/A | N/A |
| 4 | N/A | N/A | A | 1 |

B5 Virtual Circuit Table



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 1 | N/A | N/A | A | 0 |
| 4 | A | 1 | N/A | N/A |

B6 Virtual Circuit Table



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 1 | A | 0 | N/A | N/A |
| 2 | A | 1 | N/A | N/A |
| 3 | A | 2 | B | 0 |
| 4 | B | 1 | A | 3 |
| 5 | N/A | N/A | A | 4 |

B7 Virtual Circuit Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Connection # | Incoming Interface | Incoming VCI | Outgoing Interface | Outgoing VCI |
| 1 | D | 0 | A | 0 |
| 2 | D | 1 | A | 1 |
| 3 | B | 0 | A | 2 |
| 4 | A | 3 | D | 2 |
| 5 | A | 4 | B | 1 |
| 6 | C | 0 | D | 3 |

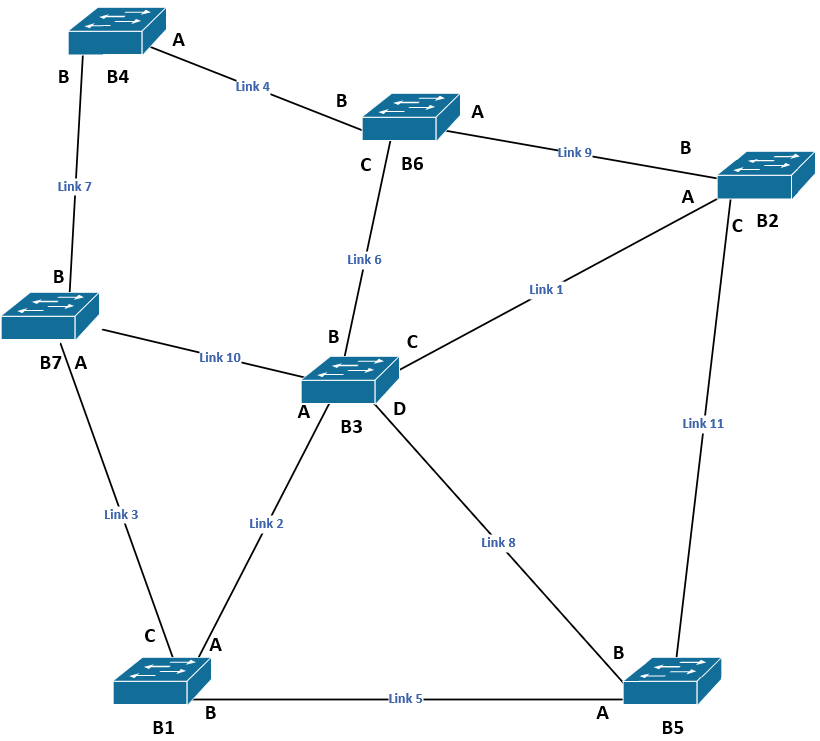


# Given the following information, list the links that form the spanning tree for the network diagram below.

|  |  |
| --- | --- |
| Link Bandwidths | |
| Link | Bandwidth |
| 1 | 10Gbit |
| 2 | 100Mbit |
| 3 | 100Gbit |
| 4 | 1Gbit |
| 5 | 1Gbit |
| 6 | 100Gbit |
| 7 | 1Gbit |
| 8 | 10Gbit |
| 9 | 1Gbit |
| 10 | 10Gbit |
| 11 | 10Gbit |

|  |  |
| --- | --- |
| Costs |  |
| Bandwidth | Cost |
| 100Mbit | 50 |
| 1Gbit | 26 |
| 10Gbit | 14 |
| 100Gbit | 8 |

The links in the spanning tree are:





Used [Kurskal Algorithm](https://www.simplilearn.com/tutorials/data-structure-tutorial/kruskal-algorithm#:~:text=with%20redundant%20paths.-,What%20is%20a%20Minimum%20Spanning%20Tree%3F,connect%20a%20set%20of%20vertices.) for creating Minimum Spanning Tree. Work pictured below:

A picture containing text, handwriting, diagram, line

Description automatically generated

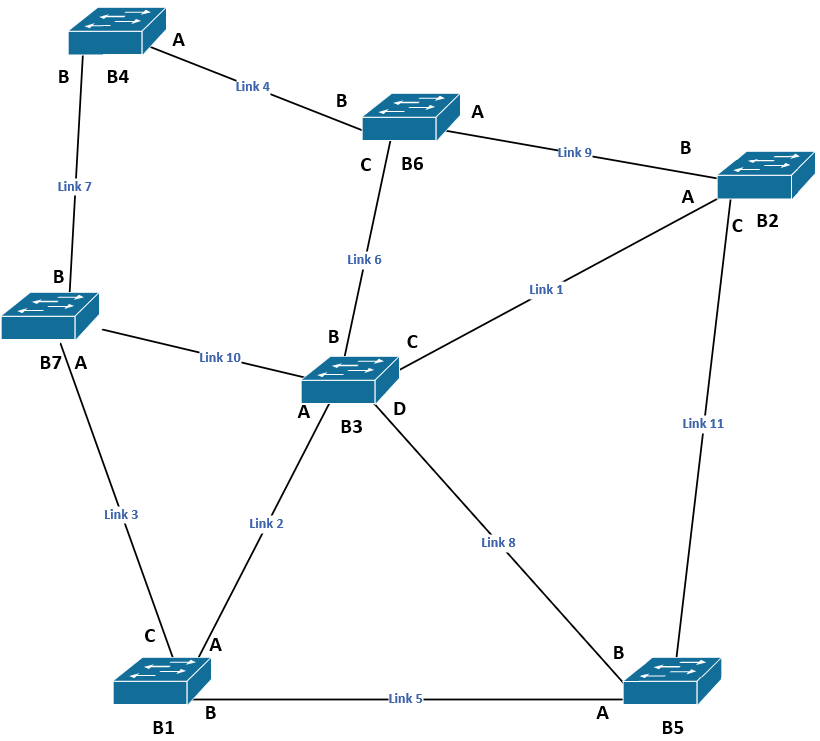
# Using the network from the previous problem, what would the spanning tree be if Link 2 were upgraded to 100Gbps?

The diagram is repeated below.

The links in the spanning tree are:

|  |  |
| --- | --- |
| Costs |  |
| Bandwidth | Cost |
| 100Mbit | 50 |
| 1Gbit | 26 |
| 10Gbit | 14 |
| 100Gbit | 8 |

|  |  |
| --- | --- |
| Link Bandwidths | |
| Link | Bandwidth |
| 1 | 10Gbit |
| **2** | **100Gbit** |
| 3 | 100Gbit |
| 4 | 1Gbit |
| 5 | 1Gbit |
| 6 | 100Gbit |
| 7 | 1Gbit |
| 8 | 10Gbit |
| 9 | 1Gbit |
| 10 | 10Gbit |
| 11 | 10Gbit |





A graph on a piece of paper

Description automatically generated with low confidence



# For the classful IP address and mask 177.82.145.43/19, list the subnet ID, the range of usable IP addresses, and the broadcast address. You must show your work for determining the subnet ID and the broadcast address:

19 – 16 = 3 1 bits in the 3rd octet, so our mask is:

11111111 11111111 111|00000 00000000

And since our 3rd octet is 145, we know the address:

xxxxxxxx xxxxxxxx 100|10001 00000000

We can now bitwise and those together, and we get:

xxxxxxxx xxxxxxxx 100|00000 00000000

Meaning our network subnet ID is:

177.82.128.0

We now need to set every bit right of the mask to 1 to get our broadcast address:

xxxxxxxx xxxxxxxx 100|11111 11111111

Which gives us the broadcast address:

177.82.159.255

Subnet ID: 177.82.128.0

1st Usable IP Address: 177.82.128.1

Last Usable IP Address: 177.82.159.254

Broadcast Address: 177.82.159.255

**Use the following scenario and network diagram for the remaining questions.**

Scenario: Host A wishes to send a message that contains 3,000 bytes of data to Host B on a TCP/IP network. At the link layer:

* The 802.11 and Ethernet links have a MTU of 1500 bytes
* The PPP link has a MTU of 532 bytes



# How many fragments will arrive at Host B?

# How many fragments will have the M bit set?

# How many data bytes will be in the last fragment?