## National University of Computer and Emerging Sciences

# Assignment 04

## Chapter 04 to 06

CS-3001 Computer Networks – Spring 2026

**Section: BSE-6B1** **Max Marks: 140**

**Instructions:**

- **Answer all questions.** Partial credit may be awarded if your reasoning is clear, even if the final answer is incorrect.
- **Academic Integrity:** Students are expected to submit their own original work. Plagiarism, copying from others, or sharing solutions is strictly prohibited and may result in zero marks and disciplinary action.
- **Submission Guidelines:** Submit your assignment by the deadline in the format specified by the instructor. Late submissions may be penalized unless prior permission is granted.

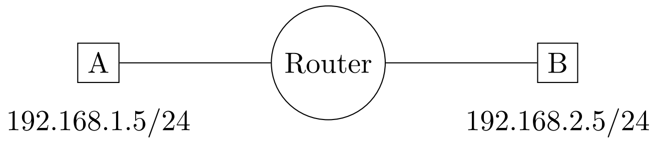
| Question | Max Marks | Obtained |
|---|---|---|
| 1 | 10 | |
| 2 | 10 | |
| 3 | 10 | |
| 4 | 10 | |
| 5 | 10 | |
| 6 | 10 | |
| 7 | 10 | |
| 8 | 10 | |
| 9 | 10 | |
| 10 | 10 | |
| 11 | 10 | |
| 12 | 10 | |
| 13 | 10 | |
| 14 | 10 | |
| **Total** | **140** | |

## Question 1: Subnetting and ARP (10 Marks)

**Scenario:** Host A (192.168.1.5/24) wants to send a packet to Host B (192.168.2.5/24). They are connected via a Router R. Interface R-left: 192.168.1.1, MAC R-L. Interface R-right: 192.168.2.1, MAC R-R. Host A's MAC: A-MAC. Host A performs a logical AND operation on its IP/Subnet and B's IP.

(a) Mathematically demonstrate why Host A decides **not** to ARP for Host B directly.

(b) Consequently, which IP address will Host A place in the ARP Request target field?




**Solution:**

(a) **Subnet Check:** Host A compares the Network ID of the destination with its own Network ID to determine if the destination is local or remote.

- Subnet Mask (/24): `255.255.255.0`
- **Host A Network ID:** 192.168.1.5 AND 255.255.255.0 = **192.168.1.0**
- **Destination B Network ID (from A's perspective):** 192.168.2.5 AND 255.255.255.0 = **192.168.2.0**

Since 192.168.1.0 $\neq$ 192.168.2.0, Host A concludes that Host B is on a **different subnet**. Therefore, it cannot ARP for B directly.

(b) Since B is on a different network, A must send the packet to its **Default Gateway**.

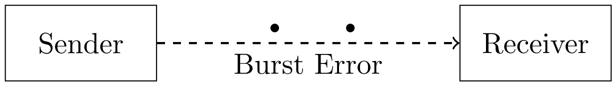
1. **Target IP in ARP Request:** `192.168.1.1` (The Router's Left Interface IP).

## Question 2: Cyclic Redundancy Check (CRC) Analysis (10 Marks)

**Scenario:** Consider a data transmission system using the CRC polynomial generator $G(x) = x^4 + x^3 + x^2 + x + 1$. The data to be transmitted is $D = 10111101$.

(a) Calculate the actual bit string transmitted by the sender.

(b) Suppose a burst error occurs during transmission such that the received bit string has the $3^{rd}$ and $4^{th}$ bits (counting from the left, 1-based index) inverted. Mathematically demonstrate whether the receiver accepts or rejects this frame.




**Solution:**

(a) **Calculate Transmitted Bit String:**

- Generator $G(x) = x^4 + x^3 + x^2 + x + 1 \Rightarrow$ **11111** (5 bits).
- Data $D = 10111101$. Append $r = 4$ zeros: 101111010000.
- Perform Binary Division ($101111010000 \div 11111$):

```
10111
-------
11111 ) 101111010000
11111
-----
100010
11111
------
111001
11111
------
001100
00000
------
011000
11111
------
011110
11111
------
0000100...  -> Remainder 01111
```

- Remainder $R = 01111$.
- Transmitted Frame $T = D + R =$ **101111010111**1.

(b) **Error Detection:**

- Original Data bits: `10111101...`
- Error in 3rd and 4th bits: `10001101...`
- Burst Error Length: The distance between the first and last incorrect bit. Here, bits 3 and 4 are wrong. Length $= 4 - 3 + 1 = 2$.
- **Property:** A CRC with a generator polynomial of degree $r$ (here $r = 4$) detects all burst errors of length $\leq r$.
- Since the burst length $2 \leq 4$, the error **will be detected**, and the receiver will **reject** the frame.

## Question 3: Parity Scheme Analysis (10 Marks)

**Scenario:** Suppose the information content of a packet is the bit pattern 1110 0110 1001 0101 and an even parity scheme is being used. What would the value of the field containing the parity bits be for the case of a two-dimensional parity scheme? Your answer should be such that a minimum-length checksum field is used.

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |

**Solution:**

We organize the bits into a 4x4 matrix and calculate Even Parity for rows and columns.

$$\begin{array}{cccc|c} 1 & 1 & 1 & 0 & \mathbf{1} \text{ (Row 1 sum=3)} \\ 0 & 1 & 1 & 0 & \mathbf{0} \text{ (Row 2 sum=2)} \\ 1 & 0 & 0 & 1 & \mathbf{0} \text{ (Row 3 sum=2)} \\ 0 & 1 & 0 & 1 & \mathbf{0} \text{ (Row 4 sum=2)} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{array}$$

- Column 1 sum=2 $\rightarrow$ 0
- Column 2 sum=3 $\rightarrow$ 1
- Column 3 sum=2 $\rightarrow$ 0
- Column 4 sum=2 $\rightarrow$ 0
- Corner bit checks parity of the parity row/column (sum=1) $\rightarrow$ 1.

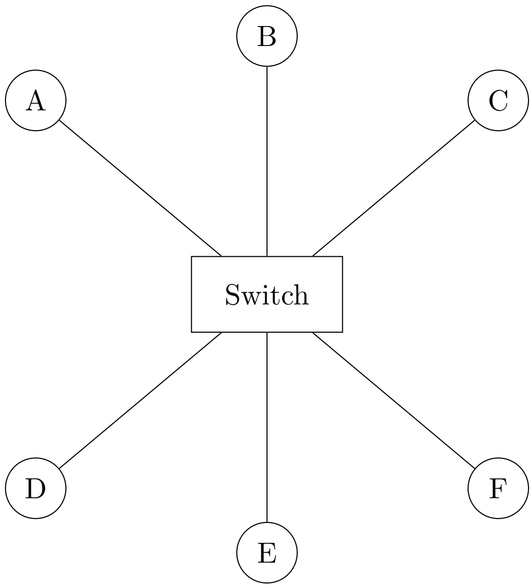
**The Parity Bits are:** Row Parity: `1000`, Column Parity: `0100`, Corner: `1`.

## Question 4: Switch Operations (10 Marks)

**Scenario:** Let's consider the operation of a learning switch in the context of a network in which 6 nodes labeled A through F are star connected into an Ethernet switch. Suppose that

1. B sends a frame to E,
2. E replies with a frame to B,
3. A sends a frame to B,
4. B replies with a frame to A.

The switch table is initially empty. Show the state of the switch table before and after each of these events. For each of these events, identify the link(s) on which the transmitted frame will be forwarded, and briefly justify your answers.

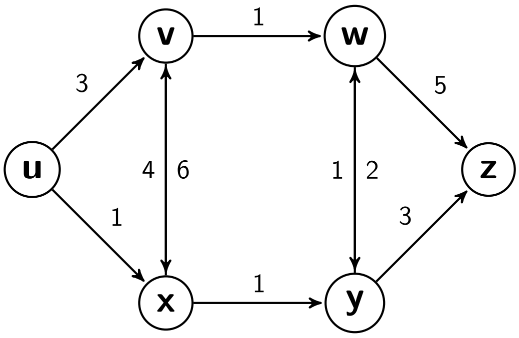



**Solution:**

| **Action** | **Switch Table State** | **Link(s) packet is forwarded to** | **Explanation** |
|---|---|---|---|
| B sends a frame to E | Switch learns interface corresponding to MAC address of B | A, C, D, E, and F | Since switch table is empty, so switch does not know the interface corresponding to MAC address of E |
| E replies with a frame to B | Switch learns interface corresponding to MAC address of E | B | Since switch already knows interface corresponding to MAC address of B |
| A sends a frame to B | Switch learns the interface corresponding to MAC address of A | B | Since switch already knows the interface corresponding to MAC address of B |
| B replies with a frame to A | Switch table state remains the same as before | A | Since switch already knows the interface corresponding to MAC address of A |

## Question 5: Directed Dijkstra's Algorithm (10 Marks)

**Scenario:** Consider the network below. Note that the links are **directed** and costs are asymmetric (e.g., $c(u, v) \neq c(v, u)$).




1. Run Dijkstra's algorithm for source node **u**. Show the table state (N', D(v), p(v), etc.) for every step until convergence. If there is a tie in costs, choose the node alphabetically (e.g., choose v over x).
2. Based on the final tree, construct the **Forwarding Table** for router **u**.

**Solution:**

### 1. Dijkstra's Execution (Source: u)

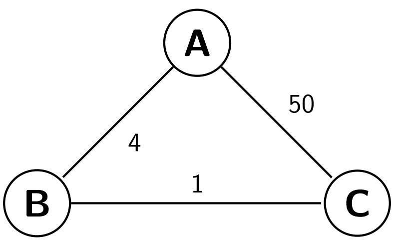
| Step | N' | D(v),p(v) | D(x),p(x) | D(w),p(w) | D(y),p(y) | D(z),p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 3,u | 1,u | $\infty$ | $\infty$ | $\infty$ |
| 1 | ux | 3,u | **1,u** | $\infty$ | 2,x | $\infty$ |
| | *Calc: D(v)=min(3, 1+4)=3* | | | | *D(y)=min($\infty$, 1+1)=2* | |
| 2 | uxy | 3,u | 1,u | 3,y | **2,x** | 5,y |
| | *Calc: D(w)=min($\infty$, 2+1)=3* | | | *D(z)=min($\infty$, 2+3)=5* | | |
| 3 | uxyv | **3,u** | 1,u | 3,y | 2,x | 5,y |
| | *(Tie w/v break: v)* | | | | | |
| 4 | uxyvw | 3,u | 1,u | **3,y** | 2,x | 5,y |
| | *Calc: D(z)=min(5, 3+5)=5* | | | | | |
| 5 | uxyvwz | 3,u | 1,u | 3,y | 2,x | **5,y** |

### 2. Forwarding Table for u

- Dest v: Direct (3,u) $\rightarrow$ Next Hop: **v**
- Dest x: Direct (1,u) $\rightarrow$ Next Hop: **x**
- Dest y: Path u-x-y $\rightarrow$ Next Hop: **x**
- Dest w: Path u-x-y-w $\rightarrow$ Next Hop: **x**
- Dest z: Path u-x-y-z $\rightarrow$ Next Hop: **x**

## Question 6: Asymmetric Link State Routing (10 Marks)

**Scenario:** Consider the triangular network below. The Bellman-Ford equation is used. Assume synchronous updates: all nodes compute their distance vectors at time $t$, then exchange messages, then compute for $t + 1$.



At time $t = 0$, the link costs are as shown above ($c(A, C) = 50$). The network has converged. Suddenly, at $t = 1$, the link cost $c(B, C)$ changes from **1 to 60**. Determine the distance $D_B(C)$ (cost from B to C) at $t = 1$, at $t = 2$ and at $t = 3$.

**Solution:**

This describes the "Count-to-Infinity" problem in Distance Vector routing. Initial State ($t = 0$):

- $D_B(C) = 1$ (direct)
- $D_A(C) = \min(50, c(A, B) + D_B(C)) = \min(50, 4 + 1) = 5$ (via B)

**Timeline:**

- **t=1:** B detects link change $1 \rightarrow 60$. B runs Bellman-Ford.

$$D_B(C) = \min(c(B, C)_{new}, c(B, A) + D_A(C))$$

Using A's old advertisement ($D_A(C) = 5$):

$$D_B(C) = \min(60, 4 + 5) = \mathbf{9}$$

- **t=2:** B advertises $D_B(C) = 9$ to A. A runs Bellman-Ford.

$$D_A(C) = \min(50, c(A, B) + D_B(C))$$

$$D_A(C) = \min(50, 4 + 9) = 13$$

(Note: Question asks for $D_B(C)$, which stays 9 at $t = 2$ while waiting for A's update, or conceptually updates at step boundaries).

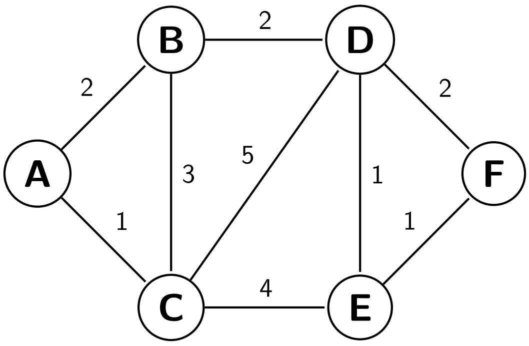
- **t=3:** A advertises $D_A(C) = 13$ to B. B runs Bellman-Ford.

$$D_B(C) = \min(60, 4 + 13) = \mathbf{17}$$

**Answer:** $D_B(C)$ is **9** at $t = 1$, and becomes **17** at $t = 3$.

## Question 7: Undirected Dijkstra's Algorithm (10 Marks)

**Scenario:** Consider the undirected network shown below. The numbers on the links represent the cost.




1. Run Dijkstra's algorithm for source node **A**. Show the table state ($N'$, $D(v)$, $p(v)$) for every step.
2. Tie-breaking rule: If two nodes have the same cost, choose the one that is alphabetically first (e.g., choose B over C).

3. Identify the shortest path from A to F and its total cost.

**Solution:**

### 1. Dijkstra Table (Source A)

| Step | N' | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 0 | A | 2,A | **1,A** | $\infty$ | $\infty$ | $\infty$ |
| 1 | AC | **2,A** | 1,A | 6,C | 5,C | $\infty$ |
| | *Calc: D(D)=min(∞, 1+5)=6, D(E)=min(∞, 1+4)=5* | | | | | |
| 2 | ACB | 2,A | 1,A | **4,B** | 5,C | $\infty$ |
| | *Calc: D(D)=min(6, 2+2)=4* | | | | | |
| 3 | ACBD | 2,A | 1,A | 4,B | **5,C** | 6,D |
| | *Calc: D(E)=min(5, 4+1)=5 (No change), D(F)=min(∞, 4+2)=6* | | | | | |
| 4 | ACBDE | 2,A | 1,A | 4,B | 5,C | **6,D** |
| | *Calc: D(F)=min(6, 5+1)=6 (No change)* | | | | | |
| 5 | ACBDEF | 2,A | 1,A | 4,B | 5,C | 6,D |

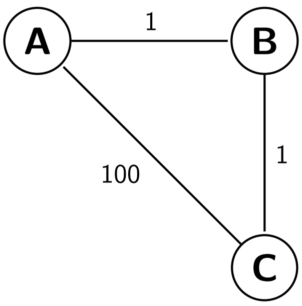
Note: At Step 3, $D(E) = 5$. Via C cost is 5 $(1 + 4)$. Via D cost is 5 $(4 + 1)$. Since value didn't improve, we keep old parent C or update depending on implementation. Assuming strict $<$, we keep C. Tie break at step 3 between E(5) and F(6)? No, D(4) was picked. At Step 4: E is 5, F is 6. Pick E.

### 2. Shortest Path A to F

- **Cost:** 6
- **Path:** $A \to B \to D \to F$ (Since parent of F is D, parent of D is B, parent of B is A).
- *Note:* $A \to C \to E \to F$ also has cost 6, but Dijkstra picked F via D because D(4) was finalized before E(5).

## Question 8: Poison Reverse Rule (10 Marks)

**Scenario:** We often say Poison Reverse fixes Count-to-Infinity for 2-node loops. Consider the 3-node loop below. $c(A, B) = 1, c(B, C) = 1, c(C, A) = 100$.




Suppose link $(C, A)$ fails (cost becomes $\infty$). At the moment of failure, the tables are: A: Dist to A=0. NextHop - B: Dist to A=1. NextHop A. C: Dist to A=2. NextHop B.

1. B and C exchange vectors. With Poison Reverse, what does C advertise to B regarding destination A?
2. If C detects the failure of (C,A), it updates its table. It calculates route to A via B. What is the new cost?

**Solution:**

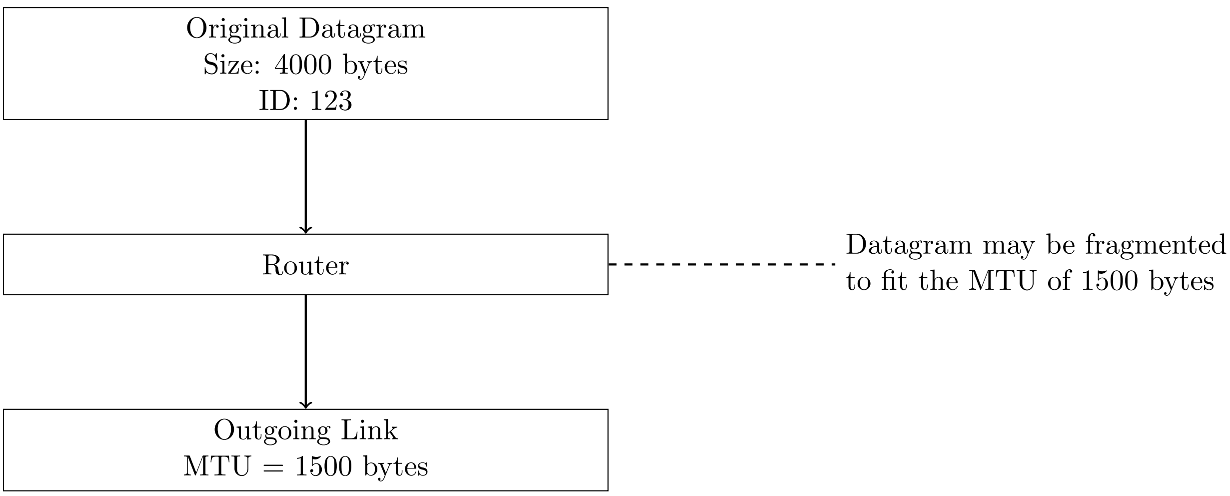
1. **Poison Reverse Advertisement:** Since C uses B to reach A ($NextHop = B$), C will advertise to B that its distance to A is $\infty$ (infinity).
2. **New Cost Calculation by C:** C detects (C,A) failure ($c(C, A) = \infty$). C checks its neighbor B.
   - Does B route via C? No (B routes directly to A).
   - So B advertises $D_B(A) = 1$.
   - C calculates: $D_C(A) = c(C, B) + D_B(A) = 1 + 1 = \mathbf{2}$.

   (Poison reverse prevents B from learning a false path through C, but C can still safely route through B).

## Question 9: Datagram Fragmentation (10 Marks)

**Scenario:** A datagram of 4,000 bytes arrives at a router and must be forwarded to a link with an MTU of 1,500 bytes. The datagram has an Identification field value of 123.

1. Determine into how many fragments the datagram will be divided.
2. For each fraagment give it's **Total length, Data size** and **Offset value** (assume an IPv4 header size of 20 bytes).




**Solution:**

**Parameters:** Total Size = 4000B, Header = 20B, Payload = 3980B. MTU = 1500B (Max Data = 1480B).

1. **Number of fragments:** $\lceil 3980/1480 \rceil = \lceil 2.68 \rceil = \mathbf{3}$ fragments.
2. **Fragment Details:**
   - **Frag 1:**

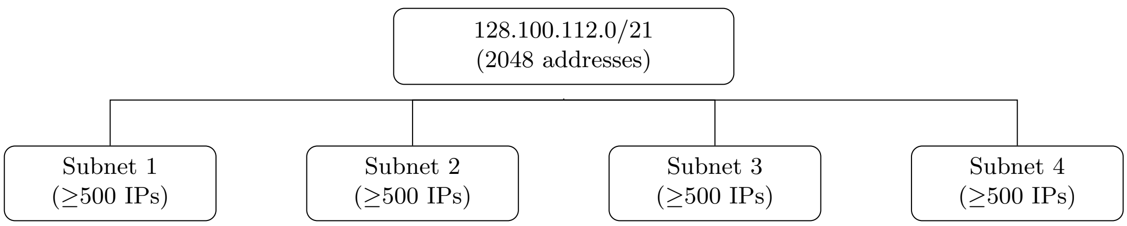
  - Data Size: 1480 bytes
  - Total Length: 1500 bytes
  - Offset: 0
  - MF Flag: 1

- **Frag 2:**
  - Data Size: 1480 bytes
  - Total Length: 1500 bytes
  - Offset: $1480/8 = \mathbf{185}$
  - MF Flag: 1
- **Frag 3:**
  - Remaining Data: $3980 - 1480 - 1480 = 1020$ bytes
  - Total Length: $1020 + 20 = \mathbf{1040}$ bytes
  - Offset: $(1480 + 1480)/8 = \mathbf{370}$
  - MF Flag: 0

## Question 10: IPv4 Subnetting Analysis (10 Marks)

**Scenario:** Consider the 128.100.112.0/21 block of IP addresses. This block of addresses must be divided into four subnetworks that have each at least 500 IP addresses.

1. Give the subnet mask of the four new subnets.
2. Specify the network address and the network prefix for each subnet.
3. Specify the broadcast IP address for each subnet.



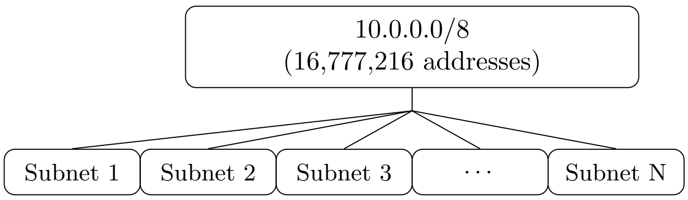

**Solution:**

1. **Subnet Mask:** Original: /21. Need 4 subnets → Borrow 2 bits ($2^2 = 4$). New Prefix: $/21+2 = /23$. Mask: **255.255.254.0**. (Check: /23 provides $2^9 = 512$ addresses. $512 > 500$. OK).
2. **Network Addresses:** Increment is $2^9 = 512$ (or 2 in the third octet).
   - Subnet 1: **128.100.112.0/23**
   - Subnet 2: **128.100.114.0/23**
   - Subnet 3: **128.100.116.0/23**
   - Subnet 4: **128.100.118.0/23**
3. **Broadcast Addresses:**

- Subnet 1: **128.100.113.255**
- Subnet 2: **128.100.115.255**
- Subnet 3: **128.100.117.255**
- Subnet 4: **128.100.119.255**

## Question 11: IPv4 Subnetting Estimate (10 Marks)

**Scenario:** Select a subnet mask for 10.0.0.0/8 so that there will be at least 16,000 subnets with at least 700 host addressed on each subnet.




Requirements:
- At least $\geq$ 16,000 subnets
- Each subnet $\geq$ 700 hosts

**Solution:**

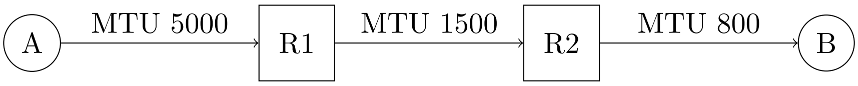
- **Subnet Requirement:** Need 16,000 subnets. $2^{13} = 8192$ (Too small), $2^{14} = 16384$. So we need at least **14 subnet bits**.
- **Host Requirement:** Need 700 hosts. $2^9 = 512$ (Too small), $2^{10} = 1024$. So we need at least **10 host bits**.
- **Total Bits Check:** Network Bits (8) + Subnet Bits (14) + Host Bits (10) = 32. This fits exactly.
- **Resulting Prefix:** $/8 + 14 = /\mathbf{22}$.
- **Subnet Mask: 255.255.252.0**

## Question 12: Nested Fragmentation (10 Marks)

**Scenario:** Host A sends a UDP datagram with **4000 bytes of payload** to Host B. The path MTUs are:

1. Link 1 (A to R1): MTU = 5000 bytes
2. Link 2 (R1 to R2): MTU = 1500 bytes
3. Link 3 (R2 to B): MTU = 800 bytes

Assume the IPv4 header is always **20 bytes**. Calculate the fragment fields (Total Length, Data Size and Offset) for the packets arriving at Host B after passing through Router R2. Note that R2 fragments the fragments received from R1.




**Solution:**

**R1 Fragmentation (MTU 1500):** Payload 4000B split into chunks of 1480B.

1. F1: 1480 data, Off 0
2. F2: 1480 data, Off 185
3. F3: 1040 data, Off 370

**R2 Fragmentation (MTU 800):** Max Data per frag $= 800 - 20 = 780$. Must be multiple of 8 $\rightarrow$ **776 bytes**.

- **Splitting F1 (1480, Off 0):**
  1. **Frag 1.1:** Data 776, **Total 796**, **Off 0**, MF 1.
  2. **Frag 1.2:** Data $1480 - 776 = 704$, **Total 724**, **Off 97** $(0 + 776/8)$, MF 1.
- **Splitting F2 (1480, Off 185):**
  1. **Frag 2.1:** Data 776, **Total 796**, **Off 185**, MF 1.
  2. **Frag 2.2:** Data 704, **Total 724**, **Off 282** $(185 + 97)$, MF 1.
- **Splitting F3 (1040, Off 370, MF 0):**
  1. **Frag 3.1:** Data 776, **Total 796**, **Off 370**, MF 1 (Changed from 0 because split).
  2. **Frag 3.2:** Data $1040 - 776 = 264$, **Total 284**, **Off 467** $(370 + 97)$, MF 0.

## Question 13: Routing Algorithms Analysis (10 Marks)

**Scenario:** Compare and contrast the properties of a centralized and a distributed routing algorithm. Give an example of a routing protocol that takes a centralized and a decentralized approach

**Solution:**

**Centralized Routing:** One controller has full network knowledge. Routes computed in one place. Fast convergence. Single point of failure. Less scalable. Example: SDN controller (OpenFlow).
**Distributed Routing:** Each router knows only its neighbors. Routes computed by all routers. Potentially slower convergence. No single point of failure. Highly scalable. Example: RIP, OSPF.

## Question 14: BGP Policy Analysis (10 Marks)

**Scenario:** True or false: When a BGP router receives an advertised path from its neighbor, it must add its own identity to the received path and then send that new path on to all of its neighbors. Also explain it?

**Solution:**

False. A BGP router adds its own AS number to a received path only when advertising it to other

ASes, and it does not send it to all neighbors it never sends a route back to the neighbor it came from and applies export policies.