National University of Computer and Emerging Sciences (Lahore Campus)

Quiz 5: Network Layer - Control Plane (Chapter 5)

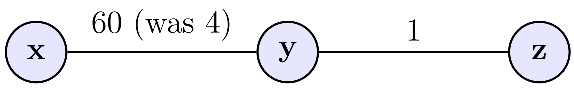
Name: ______________ **Roll No:** ______________ **Section:** BSE-6B1 (Spring 2026)

---

1. (5 points) **Distance Vector Analysis**

   Consider the 3-node linear topology below. Bellman-Ford Distance Vector routing is used. Link costs are: $c(x, y) = 4, c(y, z) = 1$. At t_0, the network is stabilized. At t_1, the link cost $c(x, y)$ dramatically increases to 60. Router y detects this change immediately. Router z does not see the link $x - y$ directly. Assume Poison Reverse is **NOT** enabled.

   1. Calculate the Distance Vector entries $D_y(x)$ and $D_z(x)$ for steps t_1, t_2, t_3.
   2. Why does the network not stabilize (What is this problem called)? Give one way it can be prevented?



   Solution:

   1. **Calculations:**
      - **Initial (t_0):** $D_y(x) = 4$, $D_z(x) = 5$ (via y).
      - t_1 **(y detects change):** y updates its distance to x.

        $$D_y(x) = \min(c(y, x) + D_x(x), c(y, z) + D_z(x)) = \min(60 + 0, 1 + 5) = \mathbf{6}$$

        (Note: y chooses path via z because $1 + 5 < 60$. z still thinks dist to x is 5).
      - t_2 **(z receives update from y):** z sees $D_y(x) = 6$.

        $$D_z(x) = \min(c(z, y) + D_y(x)) = 1 + 6 = \mathbf{7}$$

      - t_3 **(y receives update from z):** y sees $D_z(x) = 7$.

        $$D_y(x) = \min(60, 1 + 7) = \mathbf{8}$$

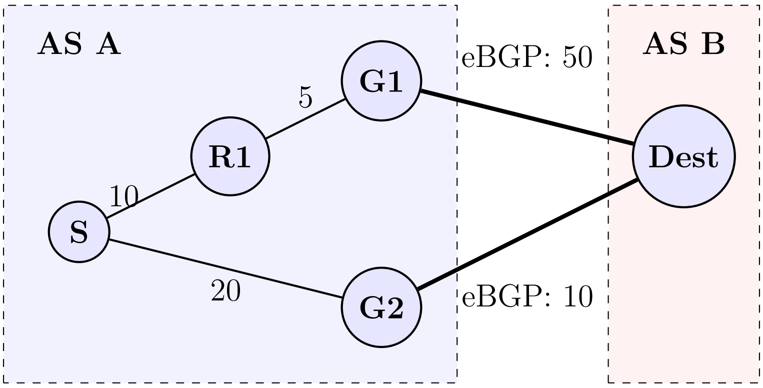
   2. **Problem & Prevention:**
      - **Problem: Count-to-Infinity**. The nodes bounce updates back and forth, slowly incrementing the cost until it reaches the direct link cost (60) or infinity.
      - **Prevention: Poisoned Reverse**. If z routes to x via y, z should advertise $D_z(x) = \infty$ to y.

2. (5 points) **Hot vs. Cold Potato Routing Calculation**

   AS A needs to send a packet to destination IP in AS B. AS A has two gateways ($G1, G2$) connected to AS B. The IGP (Intra-AS) costs within AS A are shown in the diagram. The eBGP costs (latency) on the links between AS A and AS B are also shown.

- **Hot Potato Routing:** Minimize intra-AS cost.
- **Cold Potato Routing:** Minimize total cost (Intra + Inter).

Calculate the path and total cost for the source node S for both routing strategies. What is the efficiency penalty (Total Cost Difference) of using Hot Potato routing in this specific topology?

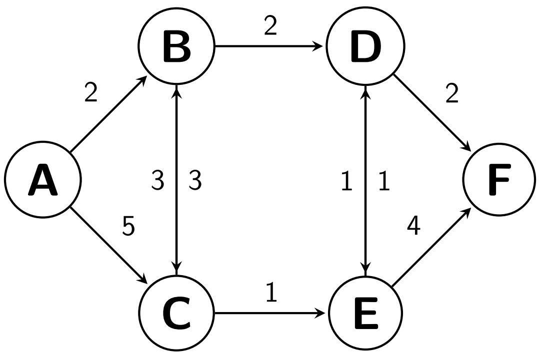


Solution:

- **Internal Costs:** Path to G1 ($S \to R1 \to G1$) $= 10 + 5 = 15$. Path to G2 ($S \to G2$) $=$ 20.
- **Hot Potato (Minimize Internal Cost):**
  - Compares internal cost 15 (via G1) vs 20 (via G2). Chooses G1.
  - **Path:** $S \to R1 \to G1 \to$ Dest.
  - **Total Cost:** $15(\text{Intra}) + 50(\text{Inter}) = \mathbf{65}$.
- **Cold Potato (Minimize Total Cost):**
  - Via G1: $15 + 50 = 65$.
  - Via G2: $20 + 10 = 30$.
  - Chooses G2 (since $30 < 65$).
  - **Path:** $S \to G2 \to$ Dest.
  - **Total Cost: 30**.
- **Efficiency Penalty:** Cost(Hot) - Cost(Cold) $= 65 - 30 = \mathbf{35}$.

3. (5 points) **Dijkstra's Algorithm and Forwarding tables**

Compute the Forwarding Table for Node **E** (not the source A) using Dijkstra's algorithm. Note the directed edges and asymmetric costs. Run Dijkstra's algorithm originating at Node **E**. Provide the resulting Forwarding Table for Node E. (Format: Destination — Next Hop).



Solution: Running Dijkstra from Source E:

1. **Initialization:** $D(E) = 0$. Neighbors: $D(D) = 1, D(F) = 4$. All others ∞.
2. **Step 1:** Select node with min dist: **D** (dist=1).
3. **Step 2:** Relax neighbors of D (D goes to E, F).
   - Path to F via D: $D(D) + c(D, F) = 1 + 2 = 3$.
   - Since 3 < 4 (current D(F)), update D(F) to 3. Predecessor of F becomes D.
4. **Step 3:** Select node with min dist: **F** (dist=3). No outgoing edges.
5. Nodes A, B, C are unreachable from E in this directed graph.

Forwarding Table for E:

Destination	**Next Hop**	**Cost**
D	D	1
F	D	3
A	- (Unreachable)	∞
B	- (Unreachable)	∞
C	- (Unreachable)	∞