Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address
- **Routing**
    - Finding “how to” move towards the destination
    - E.g., which airplane should the stuff go on
- **Forwarding**
    - Actually “moving” towards the destination
    - E.g., Using airplane/truck/rail

Network layer — Forwarding

Attempt 1: Broadcast

- **Send to everybody**
- **Goods**
  - Oh, well, simplicity
- **Not-so-goods**
  - Oh, well, everything else
  - Bandwidth overheads

Network layer — Forwarding

Attempt 2: Time division Multiplexing

- **Each source-destination pair assigned a time slot**
  - Can send data only during that slot
- **Goods**
  - No collisions
- **Not-so-goods**
  - Underutilization of resources

Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

- **Each source-destination pair assigned a subset of resources**
    - Can use only “assigned” resources (e.g., bandwidth)
- **Goods**
    - Predictable performance
- **Not-so-goods**
    - Underutilization of resources

Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

- **Source establishes connection**
  - Resources along the path are reserved
- **Source sends data**
  - Transmit data using the reserved resources
- **Source tears down connection**
  - Free resources for others to use

Network layer — Forwarding

Circuit Switching

- **Goods:**
  - Predictable performance
  - Reliable delivery
  - Simple forwarding mechanism
- **Not-so-goods**
  - Resource underutilization
  - Blocked connections
  - Connection set up overheads
  - Per-connection state in switches (scalability problem)

Network layer — Forwarding

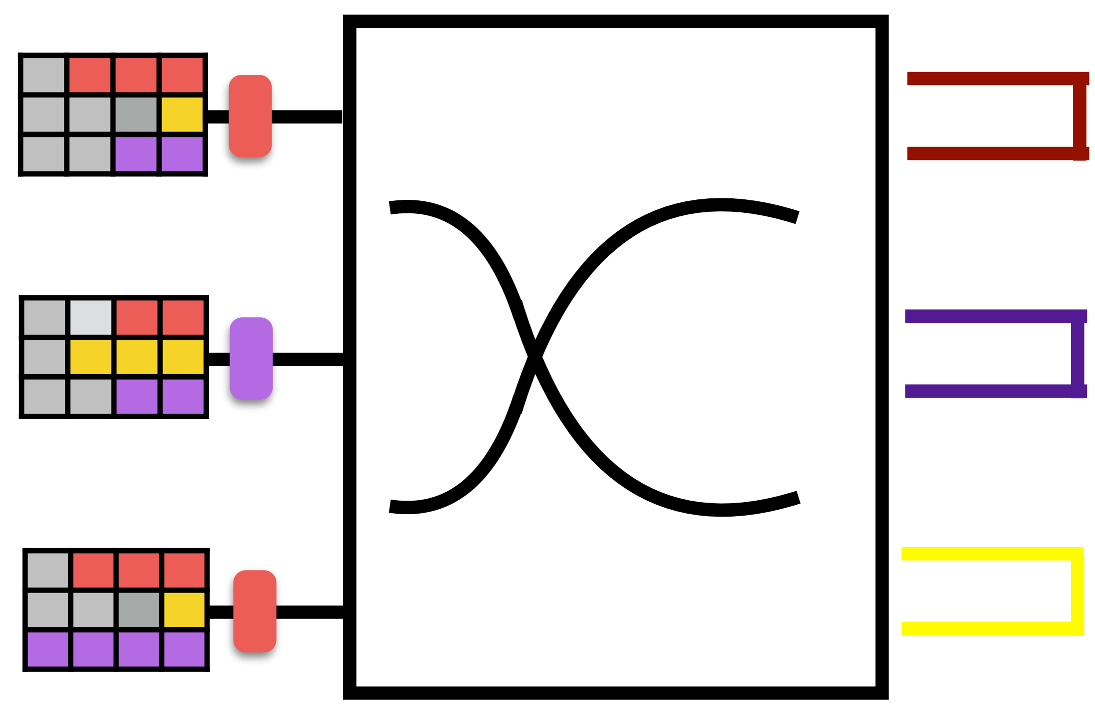
Attempt 4: Packet Switching

- **Divide the message into packets**
- **Put destination address in the header of each packet**
    - Just like shipping stuff
- **Each device stores a “look-up table”**
    - Whats the next hop towards the destination?
- **Destination receives the packet(s)**
    - And reconstructs the message

Network layer — Forwarding

Packet Switched forwarding

- **Hop-by-hop forwarding**
- **Each router has a "look-up table" (forwarding information base)**
    - What should be stored in this table?
    - Prefix-based forwarding **(longest-prefix matching)**
        - Maps **prefixes** to the next-hop



Network layer — Forwarding

Packet Switching

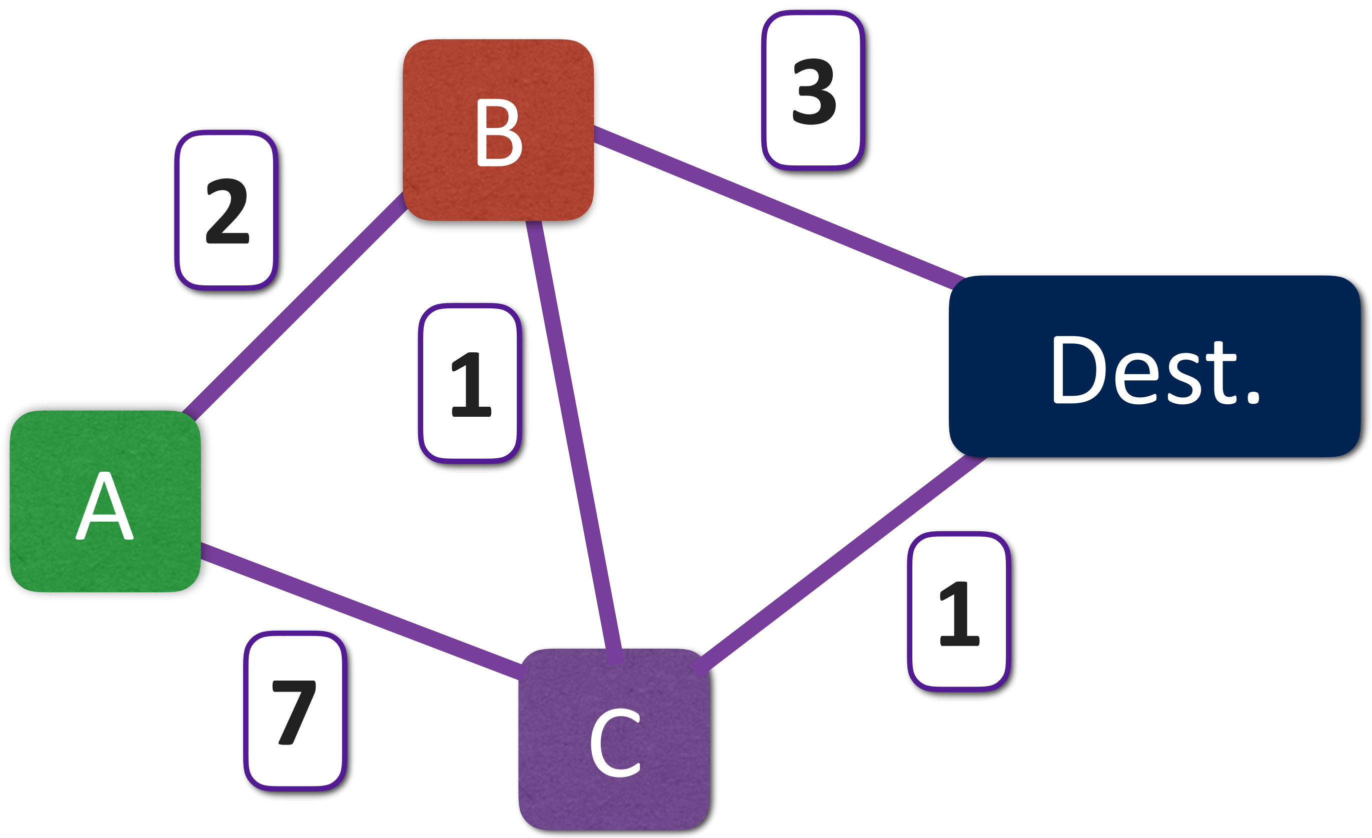
- **Goods:**
  - No resource underutilization
    - A source can send more if others don’t use resources
  - No blocked connection problem
  - No per-connection state
  - No set-up cost
- **Not-so-goods:**
  - Packet header overhead
  - Network failures become a problem

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address
- **Routing**
    - Finding “how to” move towards the destination
    - E.g., which airplane should the stuff go on
- **Forwarding**
    - Actually “moving” towards the destination
    - E.g., Using airplane/truck/rail

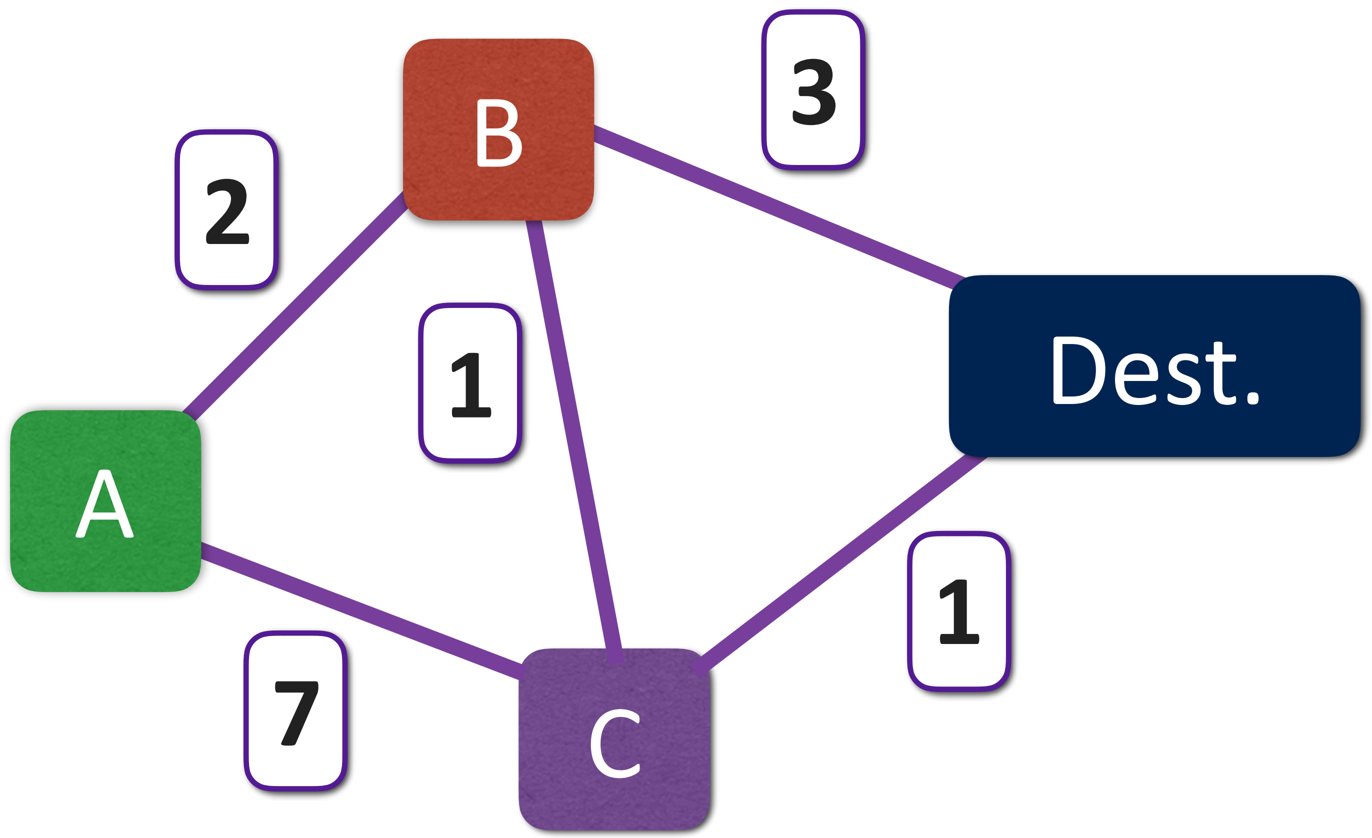
Network layer — Example



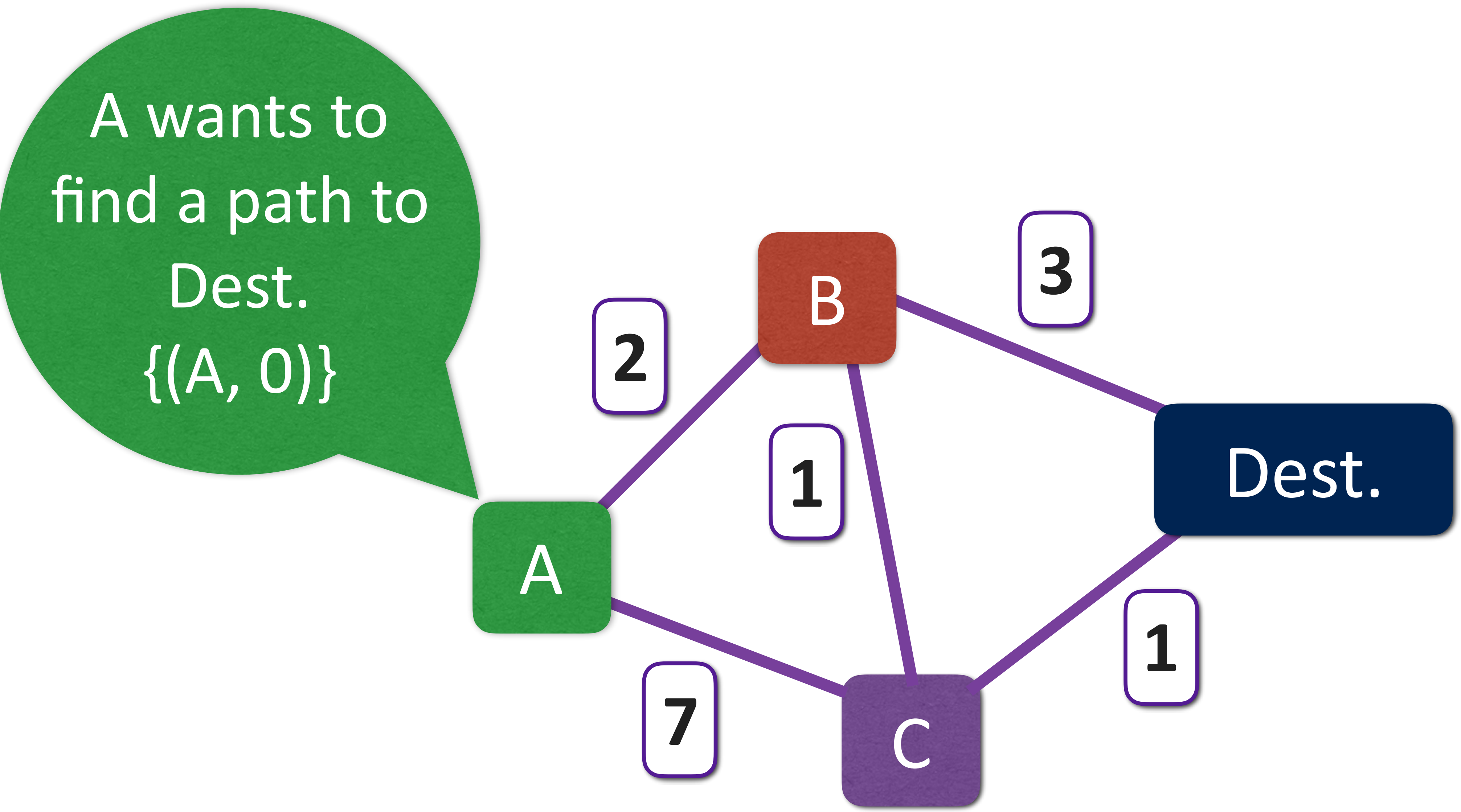
Network layer — Routing

Lets come up with a routing scheme

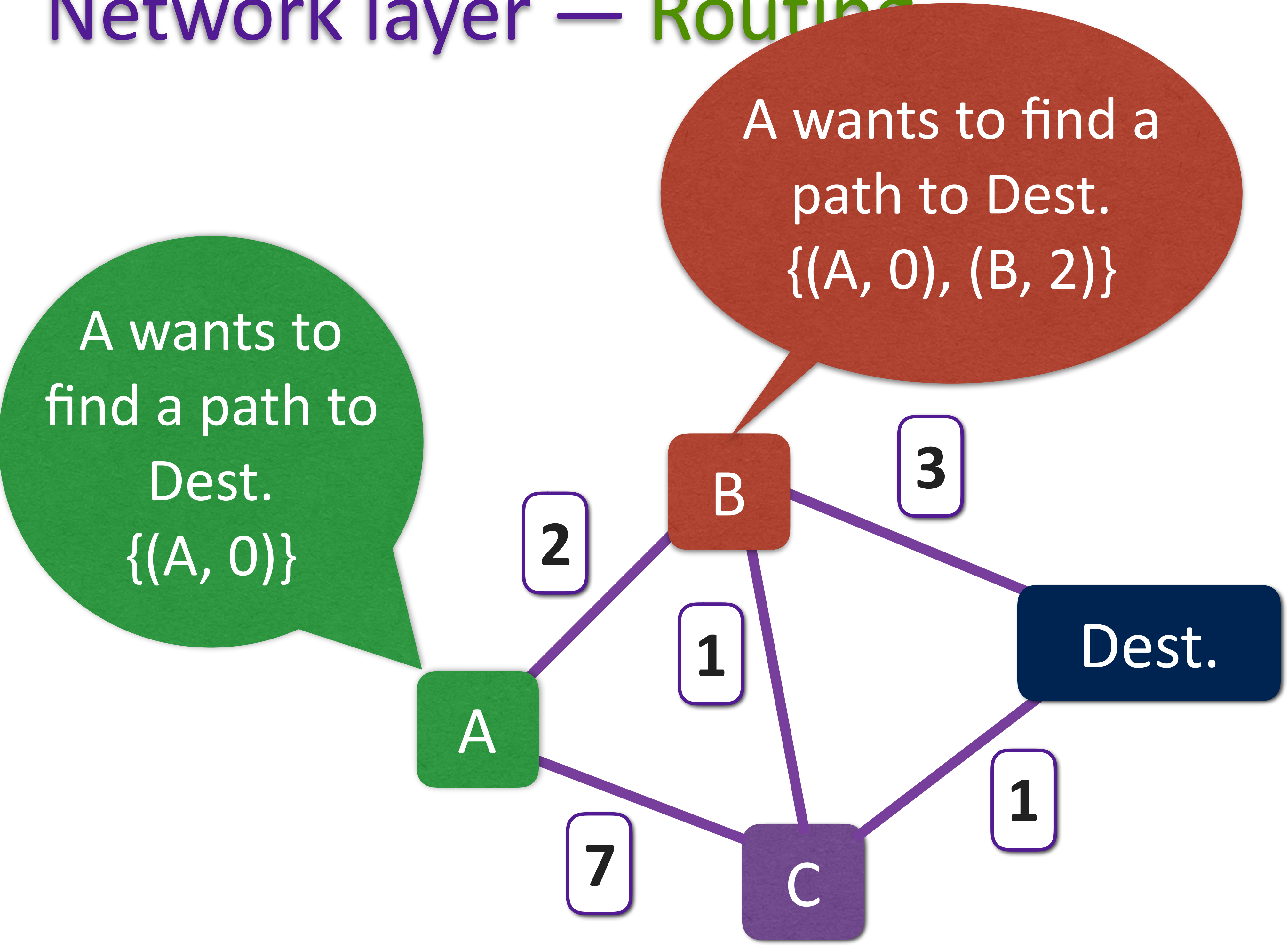
Network layer — Routing



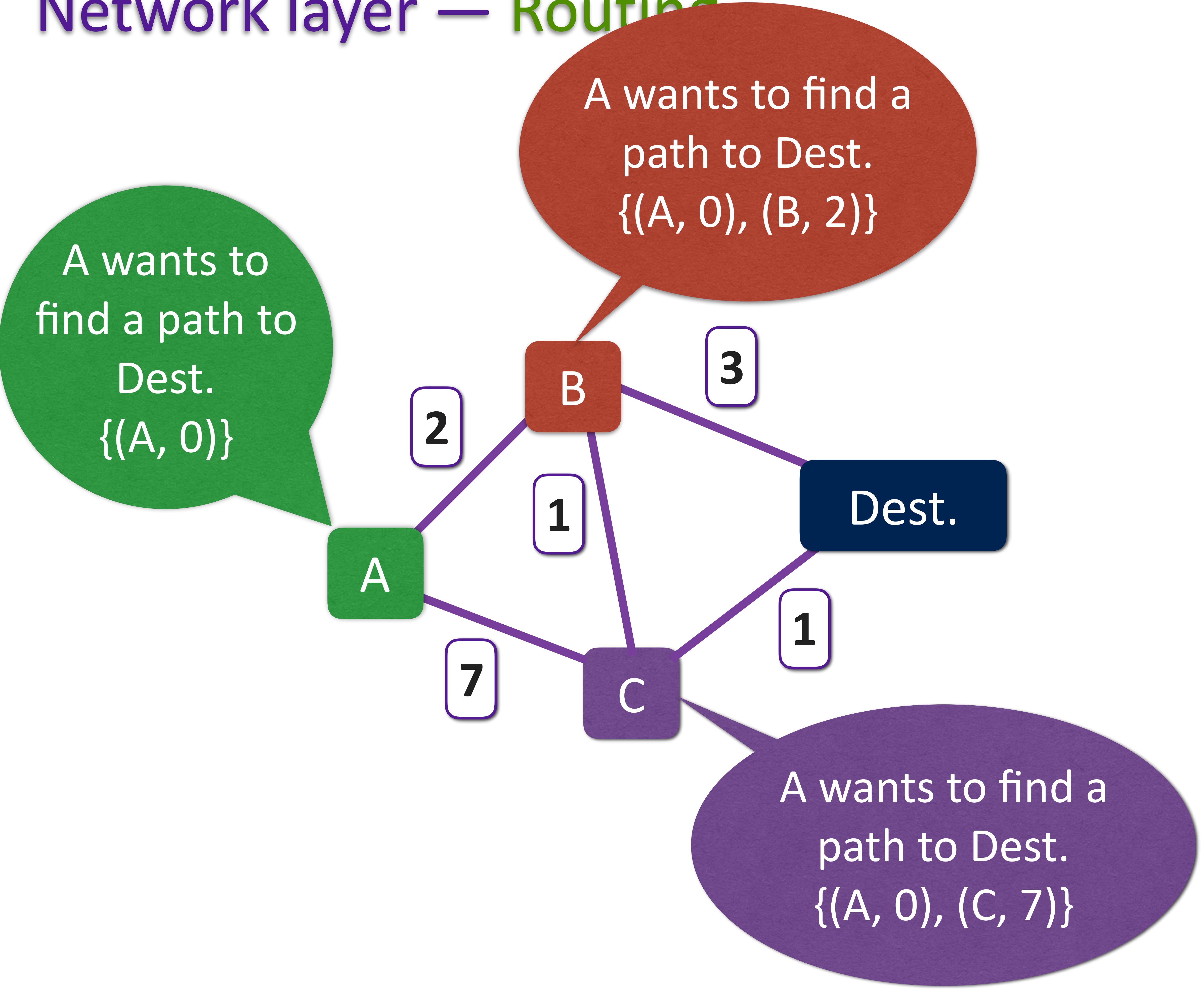
Network layer — Routing



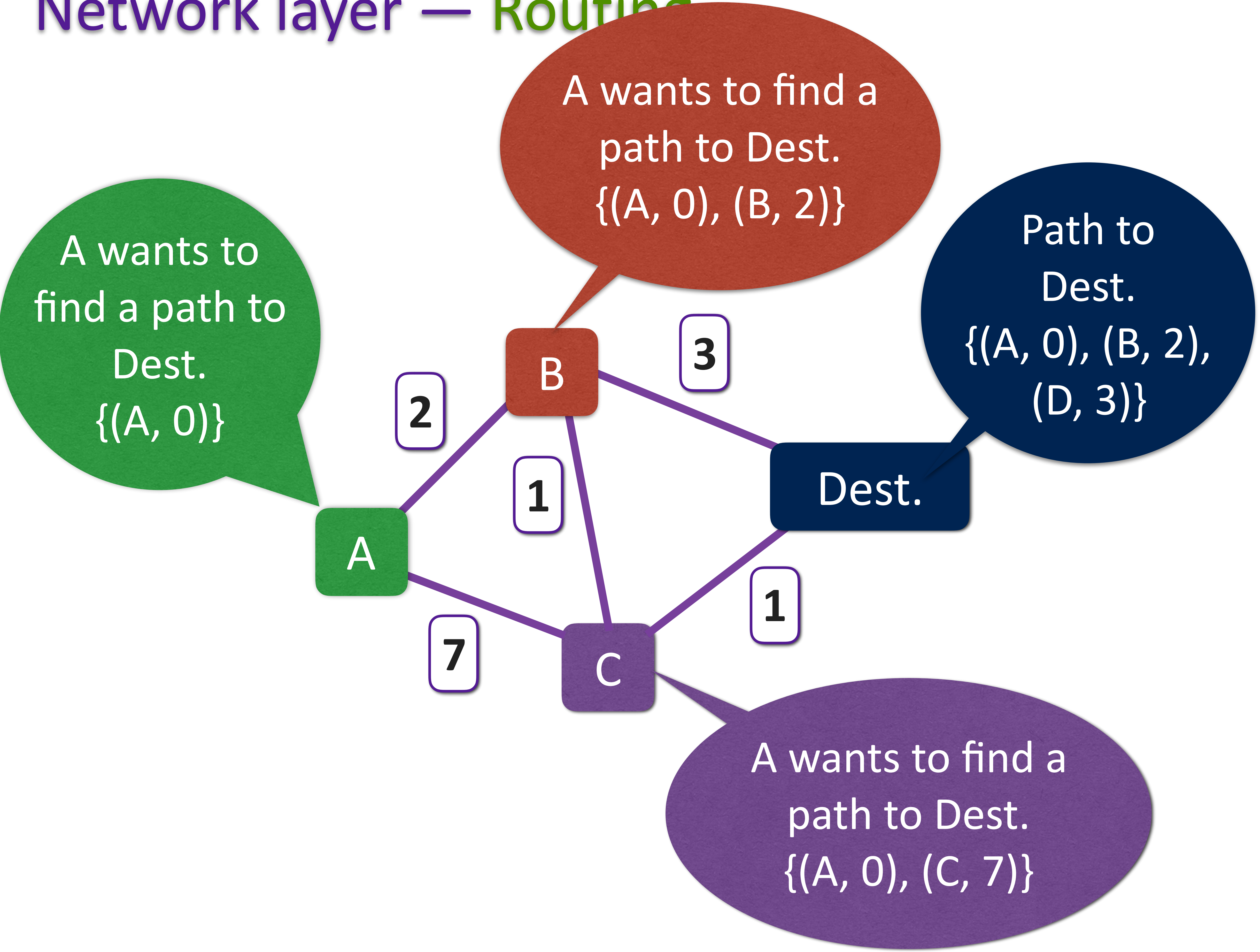
Network layer — Routing



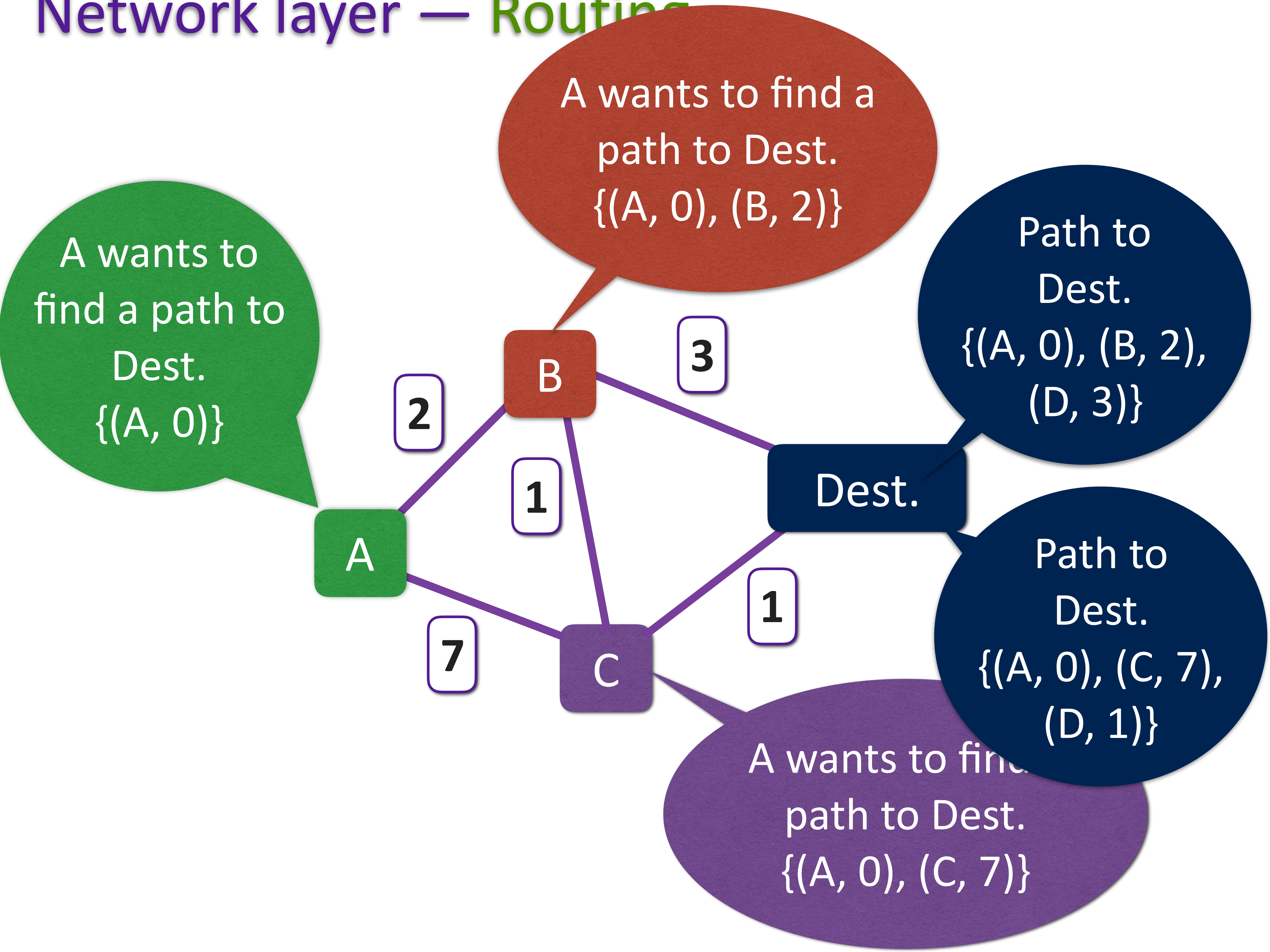
Network layer — Routing



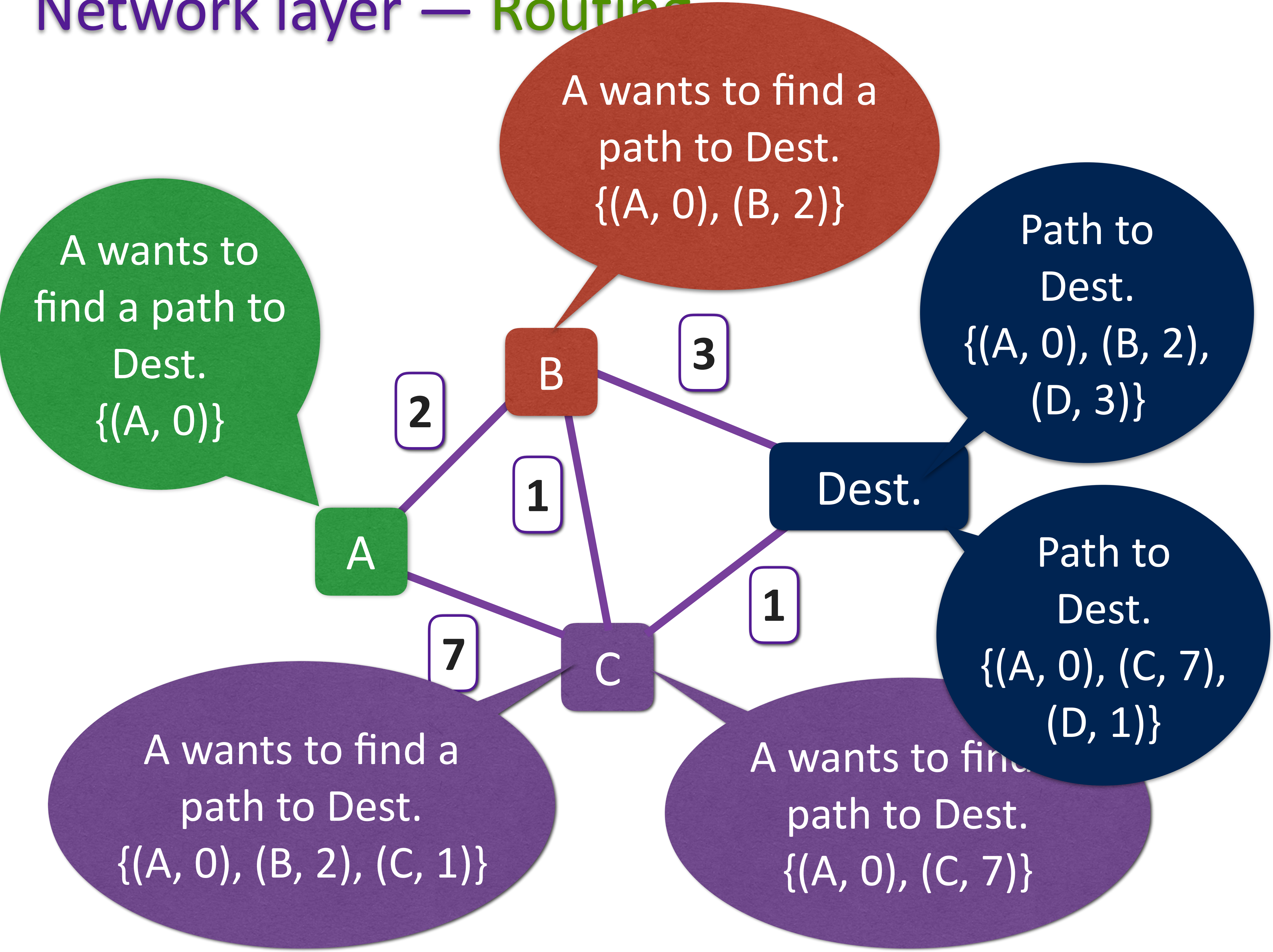
Network layer — Routing



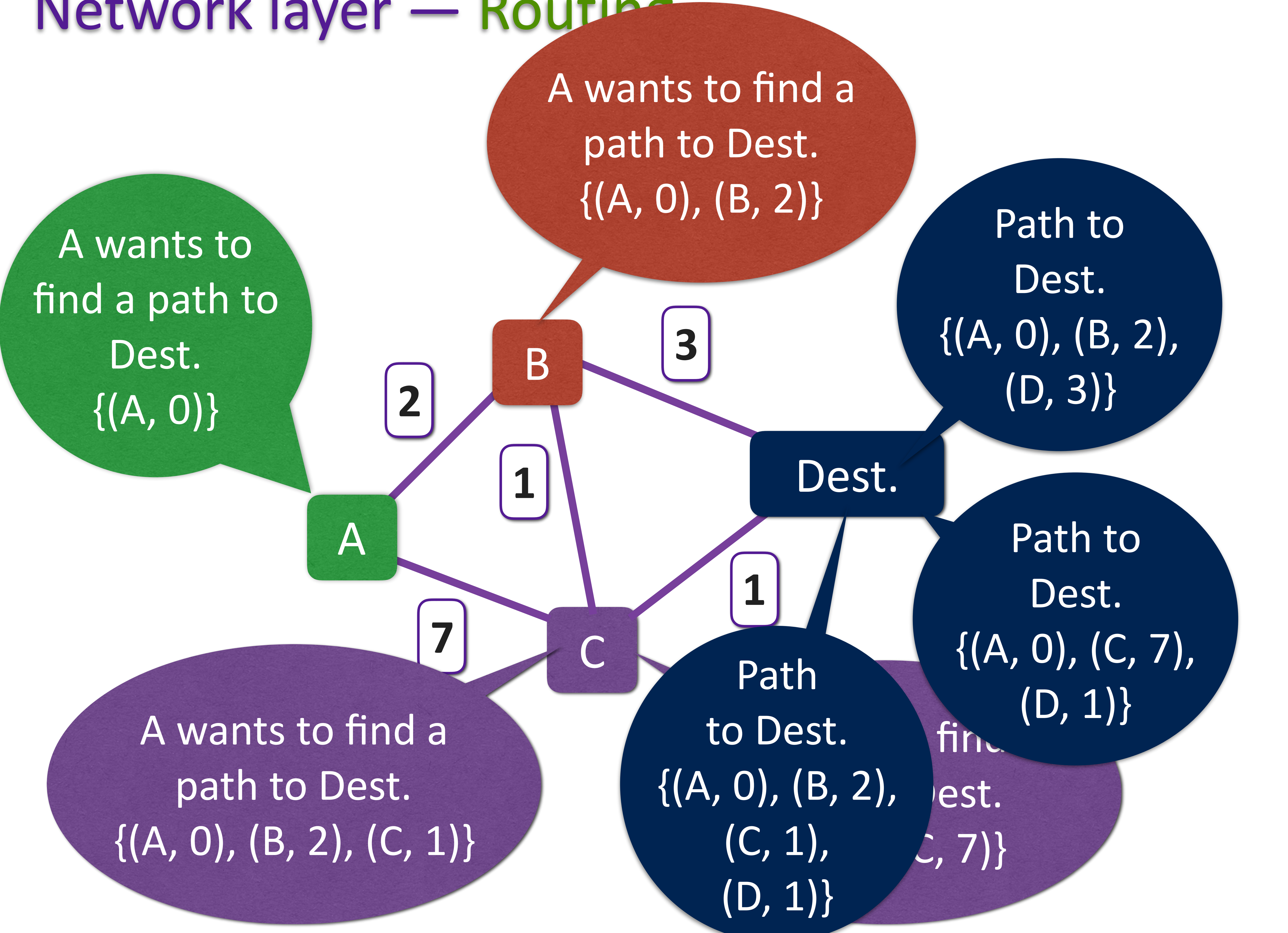
Network layer — Routing



Network layer — Routin



Network layer — Routin

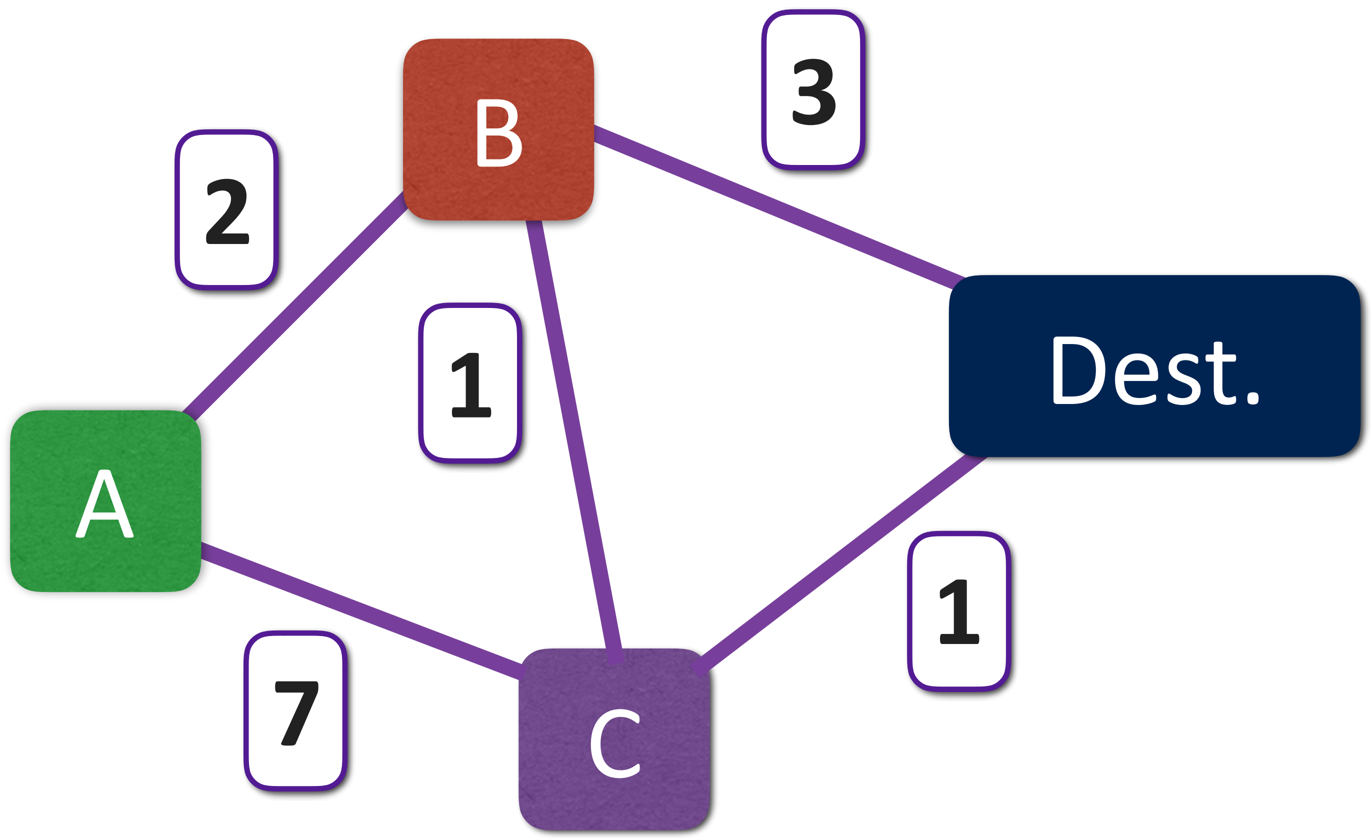


Network layer — Routing

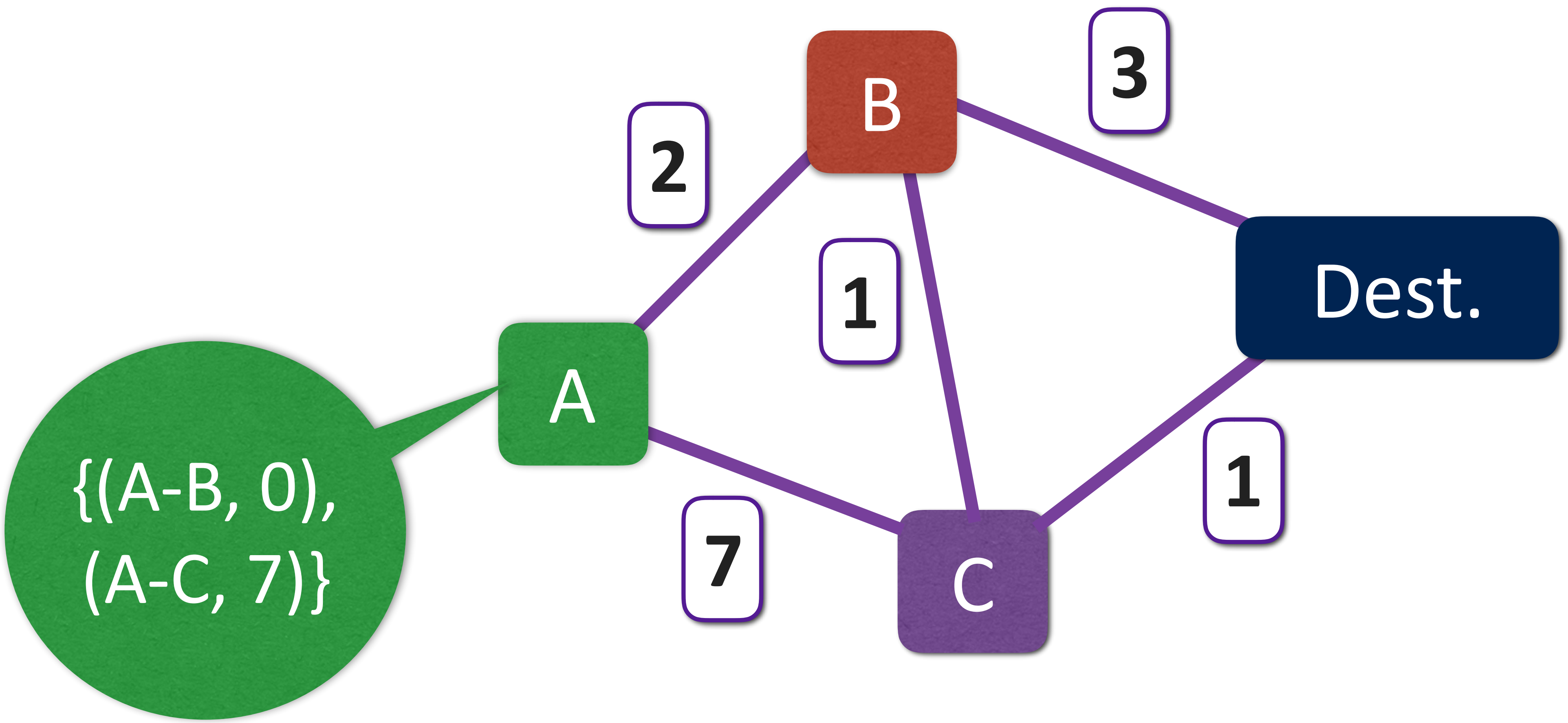
Attempt 1: Dynamic Source Routing

- **Broadcast a Route Request Packet for destination d**
  - Put source ID in the packet header
- **At each router**
  - If a path not known to the destination
    - Put its {ID, cost} in the packet header
    - Broadcast the Route Request Packet
  - Else
    - Respond with a Route Reply packet
    - Put known path in the packet header
- **Challenge?**

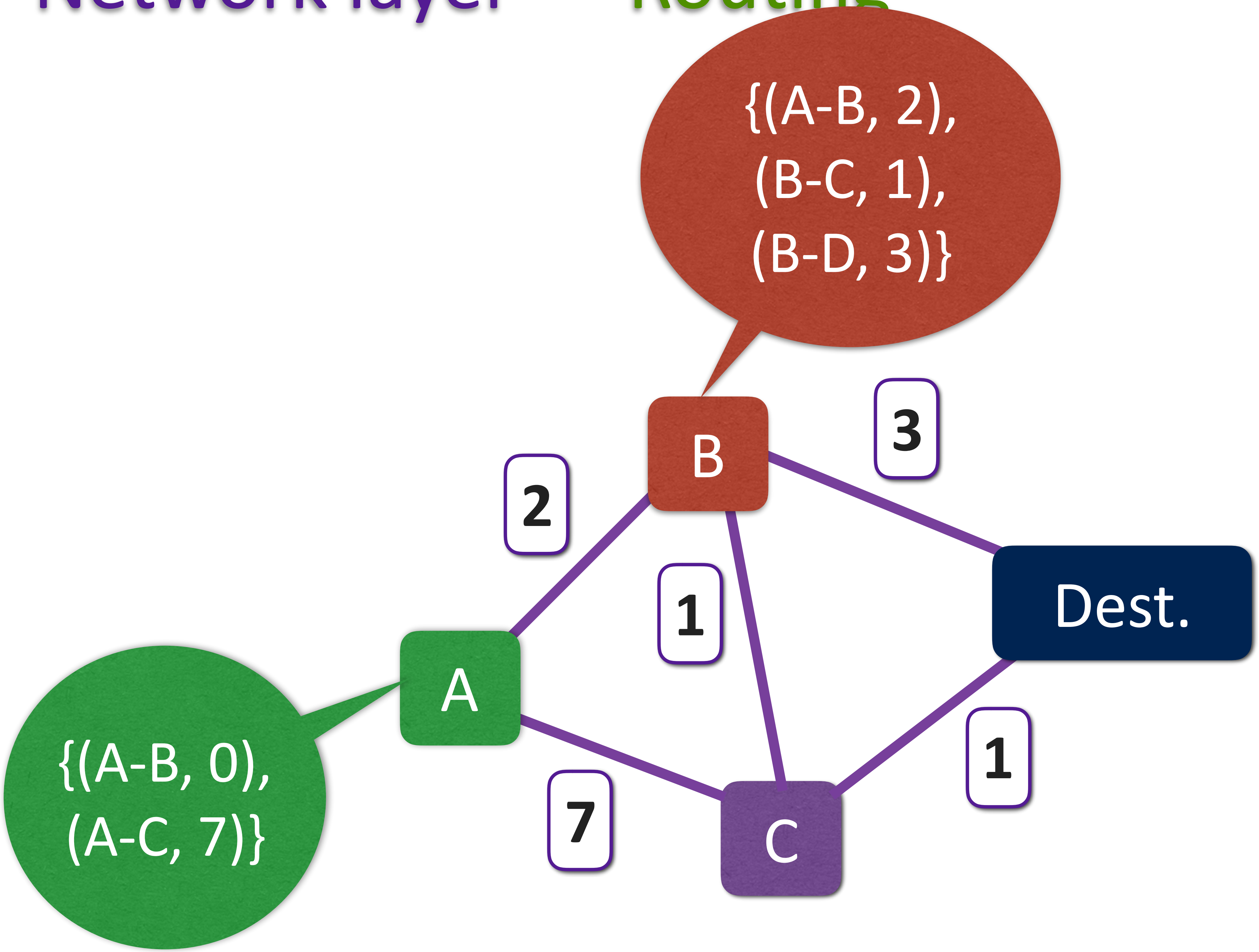
Network layer — Routing



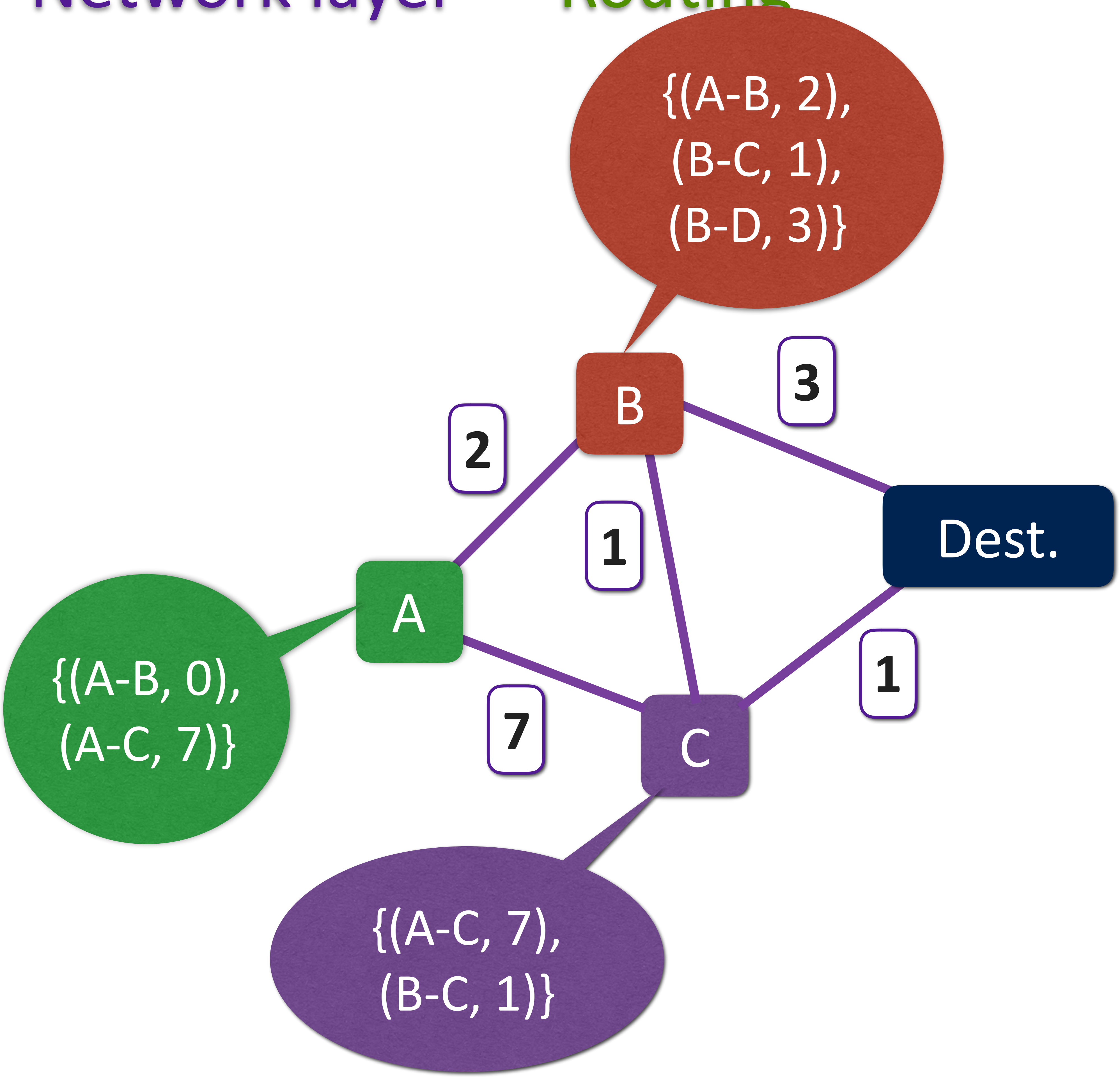
Network layer — Routing



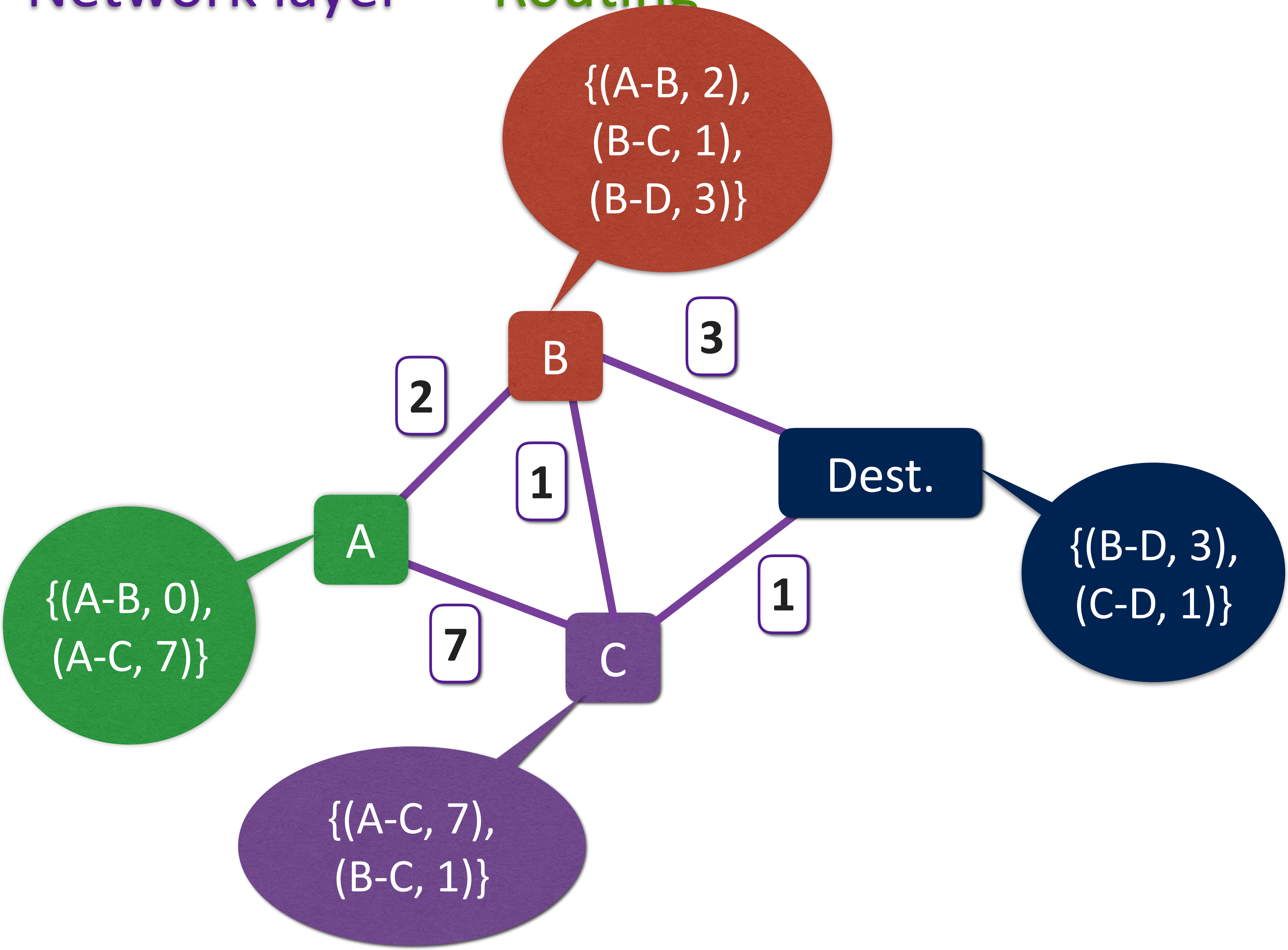
Network layer — Routing



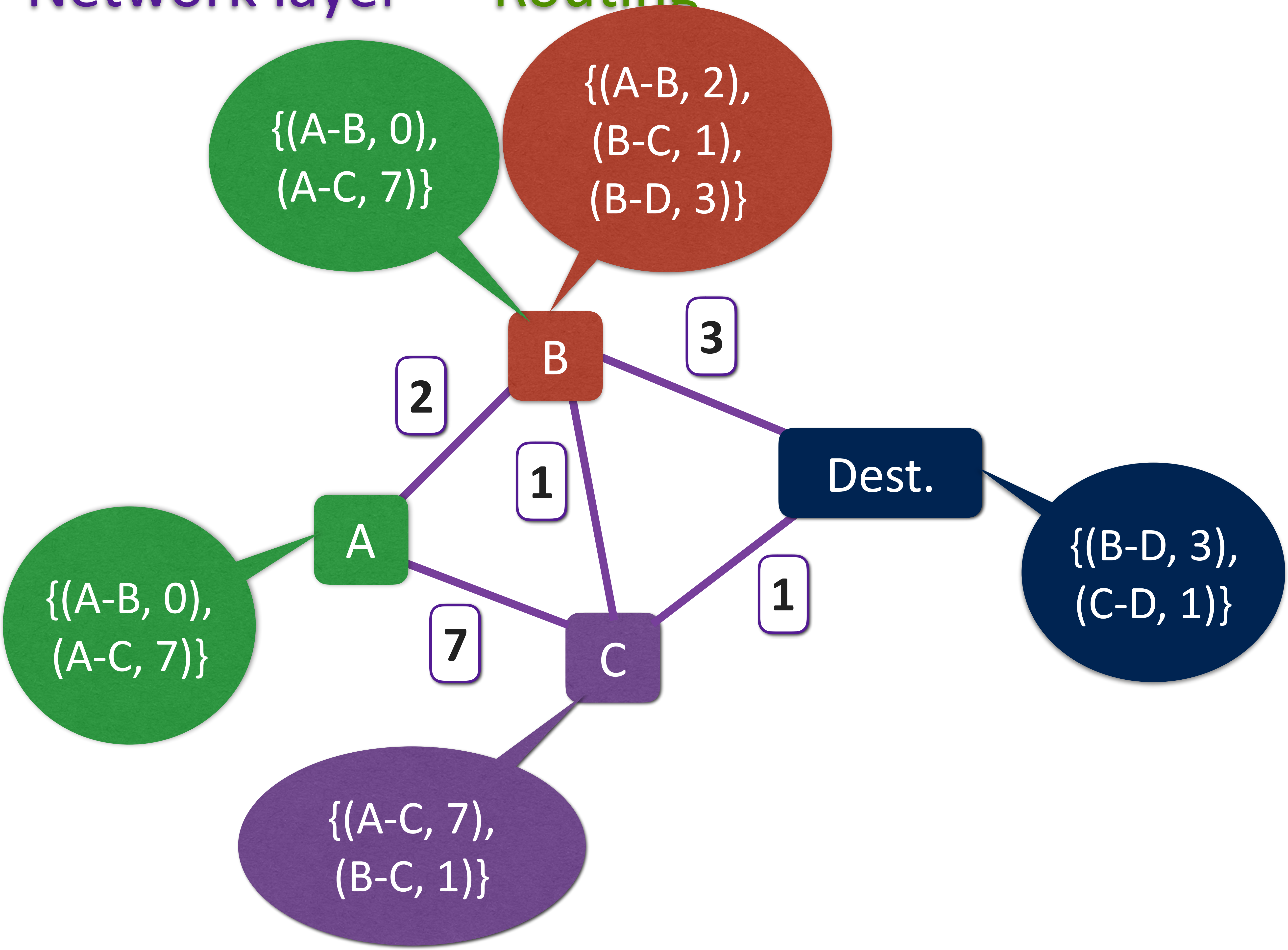
Network layer — Routing



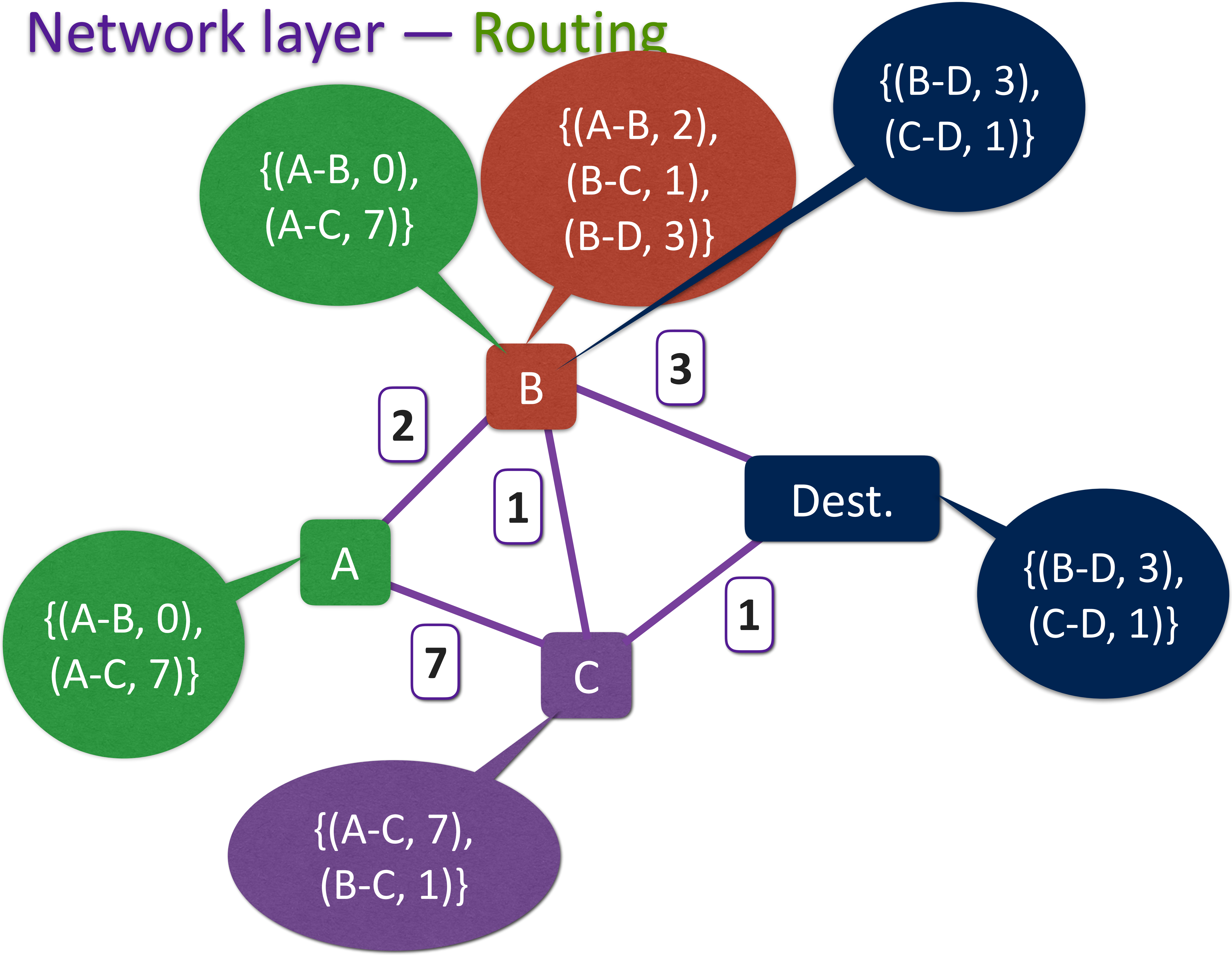
Network layer — Routing



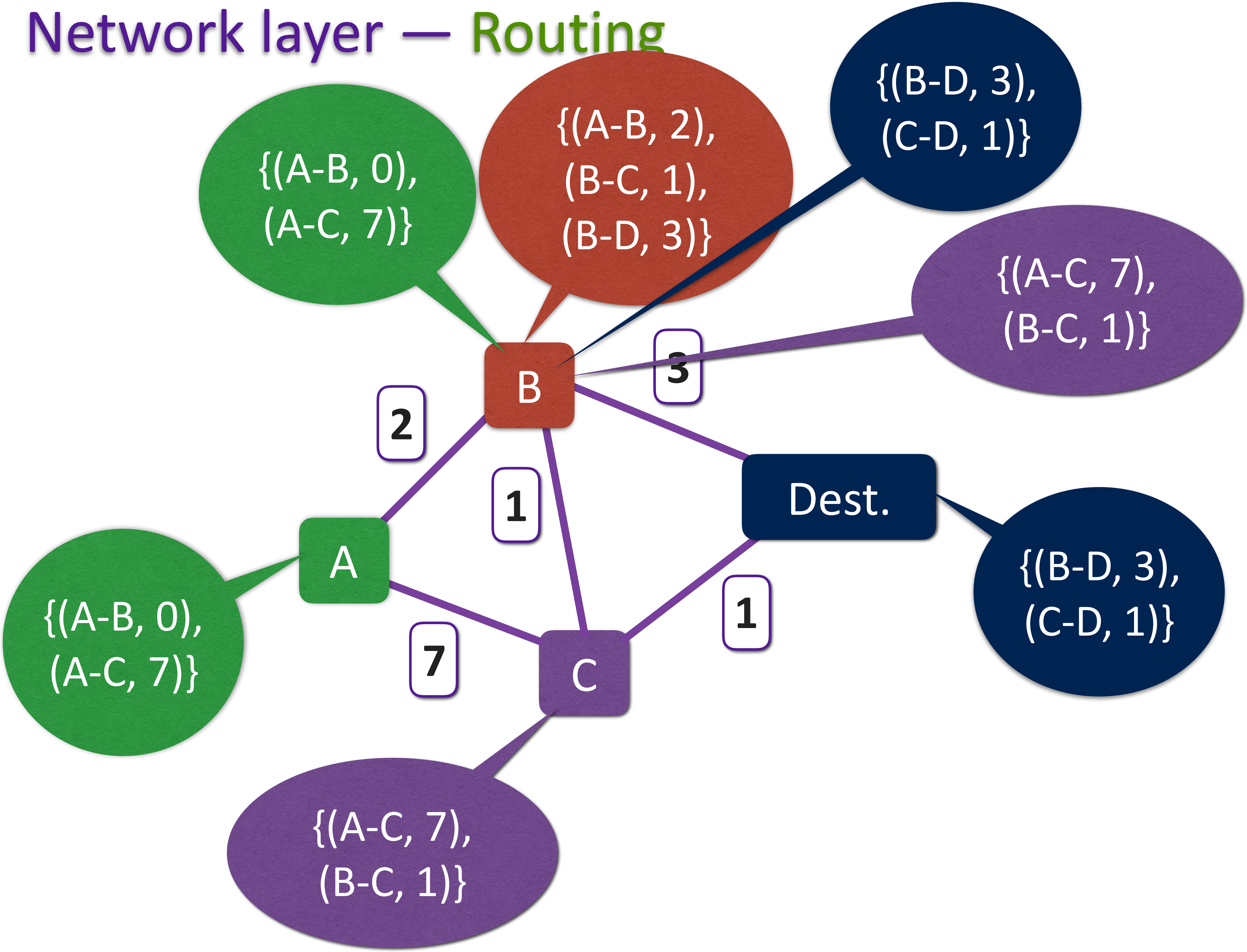
Network layer — Routing



Network layer — Routing



Network layer — Routing



Network layer — Routing

Attempt 2: Link State Routing

- **Each router maintains its local "link state" (LS)**
- **Each router periodically "floods" its LS**
  - And forwards all the LS received from other routers
- **At one point**
  - Every router knows the entire topology
- **Run a shortest path algorithm (e.g., Dijkstra) locally**
  - Find path to the destination
  - More importantly, find next-hop to the destination
- **Challenge?**

Network layer — Routing

Attempt 3: Distance Vector Routing

- **Each router**
  - **maintains its "current distance to destination"**
  - **Periodically announces it to all its neighbors**
  - **Update its local table**
    - d(A, dest) = min{d(A, neighbor) + d(neighbor, dest)}
    - {dest — distance, neighbor-that-minimizes-distance}
  - **Broadcast to all its neighbors**