2.b

1. **OPEN**: Each router picked its highest IP as Router-ID (192.4.5.4/5), advertised a 180 s hold time, its AS number, and common capabilities (Multiprotocol, Route-Refresh, 4-byte ASN).
2. **UPDATE**: Each side advertised its own prefixes, each tagged with ORIGIN=IGP, AS\_PATH=[self], NEXT\_HOP=(peer-facing IP), plus an optional MED.
3. **KEEPALIVE**: Exchanged every 60 s (= hold ÷ 3) to maintain the session.

This matches exactly what BGP’s RFCs require—and is why your session comes up cleanly and stays up without interruption.

2.c

**BGP-learned prefixes show up with code B and AD 20:**  
In R4’s table you see entries like

B 200.30.0.0/24 [20/0] via 192.4.5.5, …

The leading “B” marks a BGP route, and the administrative distance in the square brackets is **20**, the default for eBGP-learned routes.

**Metric equals the received MED:**  
The second number in [20/0] is the BGP *metric* field, which for eBGP is the **Multi-Exit Discriminator** advertised by the neighbor. Here, R5 sent MED = 0 for 200.30.0.0/24, so R4 installs it with cost 0.

**Parallel on R5:**  
R5’s routing table will likewise list R4’s prefixes with code B, AD 20, and the metric set to the MED R4 sent.

2.e

**OPEN & KEEPALIVE**

* **OPEN** : each side advertises its highest‐IP Router-ID (R3=192.3.5.3, R5=192.4.5.5), AS number, a 180 s Hold Timer, and the usual Capabilities (4-byte AS, Multiprotocol, Route-Refresh).
* **KEEPALIVE**: sent at Hold∕3 (180 s∕3) to prove liveliness.

**R5 → R3 UPDATE**

* **Advertised NLRI**: all of R5’s local LANs *plus* those it learned from R4 (200.40.0.0/24 and 200.40.1.0/24).
* **AS\_PATH** for those two: [400] (i.e. R5 is merely passing them along), so R5 behaves as a transit AS by default.
* **ORIGIN** = IGP, **NEXT\_HOP** = 192.4.5.5, **MED** = (absent or 0) for all eBGP-received routes.

**R3 → R5 UPDATE**

* **Advertised NLRI**: R3’s OSPF-learned routes (the 200.10.x/24, 200.20.x/24, etc.).
* **ORIGIN** = INCOMPLETE (because they were redistributed from OSPF).
* **MED** =
  + **0** for the two prefixes directly attached to R3, and
  + **IGP cost** (i.e. OSPF shortest-path metric) for the others—exactly as Cisco’s default “redistribute bgp” policy dictates.

2.f

**R3 Routing Table**

* **Learned via BGP from R5:**
  + **200.30.0.0/24, 200.40.0.0/24, 200.40.1.0/24 with [20/0].**
  + **Cost 0 implies either:**
    - **MED = 0 (from directly connected prefixes to R5), or**
    - **MED not set (Cisco defaults to 0).**
* **These were redistributed from R4 to R5 and then advertised to R3.**

**R5 Routing Table**

* **Learned from R3:**
  + **200.10.0.0/24 → [20/2]: MED = 2, derived from OSPF cost via R3.**
  + **200.10.1.0/24 → [20/0]: directly connected to R3 → MED = 0.**
* **Learned from R4:**
  + **200.40.0.0/24, 200.40.1.0/24 → [20/0]: directly connected to R4 → MED = 0.**

**R4 Routing Table**

* **Learned from R5:**
  + **200.10.0.0/24, 200.10.1.0/24, 200.30.0.0/24 → [20/0].**
  + **R5 advertises these with MED = 0 either because they're directly connected or no MED is set.**

**2.h**

**Wire trace** (Wireshark):

* **ICMP Echo Request** → src 192.3.5.3 (dst 200.40.0.100)
* **No Echo‐Reply**

**Why it fails:**

1. R3 chooses its **outgoing interface** address on f0/1 (192.3.5.3) as the source.
2. PC2 (200.40.0.100) replies back to its default gateway (R4), so the reply’s dest = 192.3.5.3.
3. **R4’s routing table does not know how to reach 192.3.5.0/24**, so it drops the packet.

**Ping sourcing f1/0**

**Wire trace:**

* **ICMP Echo Request** → src 200.10.1.3 (dst 200.40.0.100)
* **ICMP Echo Reply** ← src 200.40.0.100 (dst 200.10.1.3)

**Why it succeeds:**

1. By forcing the **f1/0** address (200.10.1.3), you pick a source subnet that **R4 does know**—it learned 200.10.1.0/24 via BGP from R5.
2. PC2’s reply to 200.10.1.3 is forwarded by R4 (via its BGP entry) back across R5 to R3.