6.1800 Spring 2025

Lecture #9: Routing

distance-vector, link-state, and how they scale

Katrina LaCurts | lacurts@mit.edu | 6.1800 2025

6.1800 in the news

when cables are damaged, the Internet has to **reroute** traffic

Red Sea cables have been damaged, disrupting internet traffic

By Hanna Ziady, CNN

3 minute read · Updated 9:02 AM EST, Mon March 4, 2024

London (CNN) — Damage to submarine cables in the Red Sea is disrupting telecommunications networks and forcing providers to reroute as much as a quarter of traffic between Asia, Europe and the Middle East, including internet traffic.

HGC estimates that 25% of traffic between Asia and Europe as well the Middle East has been impacted, it said in a statement Monday.

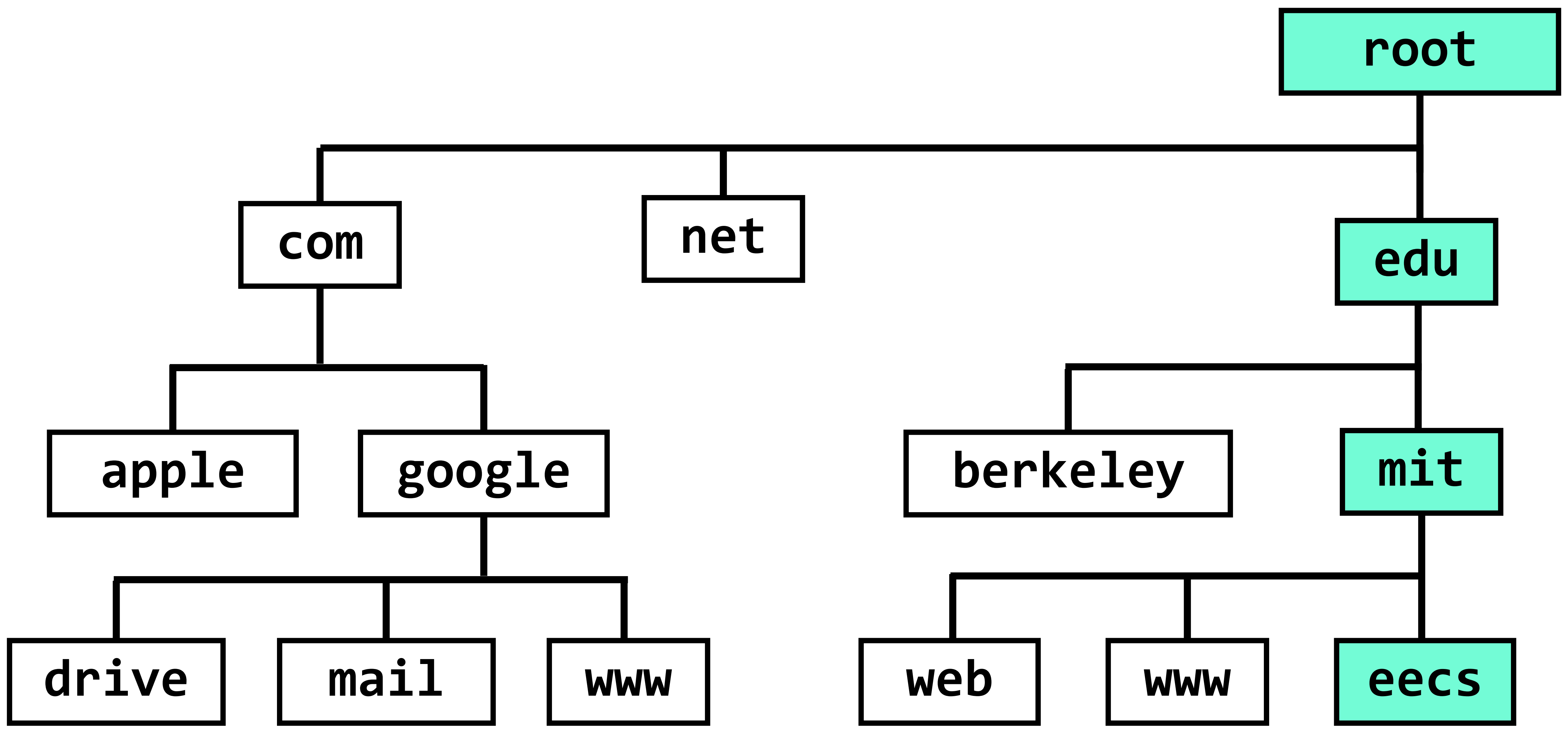
Most large telecoms companies rely on multiple undersea cable systems, allowing them to reroute traffic in the event of an outage to ensure uninterrupted service.

(this is from last year, but still relevant)

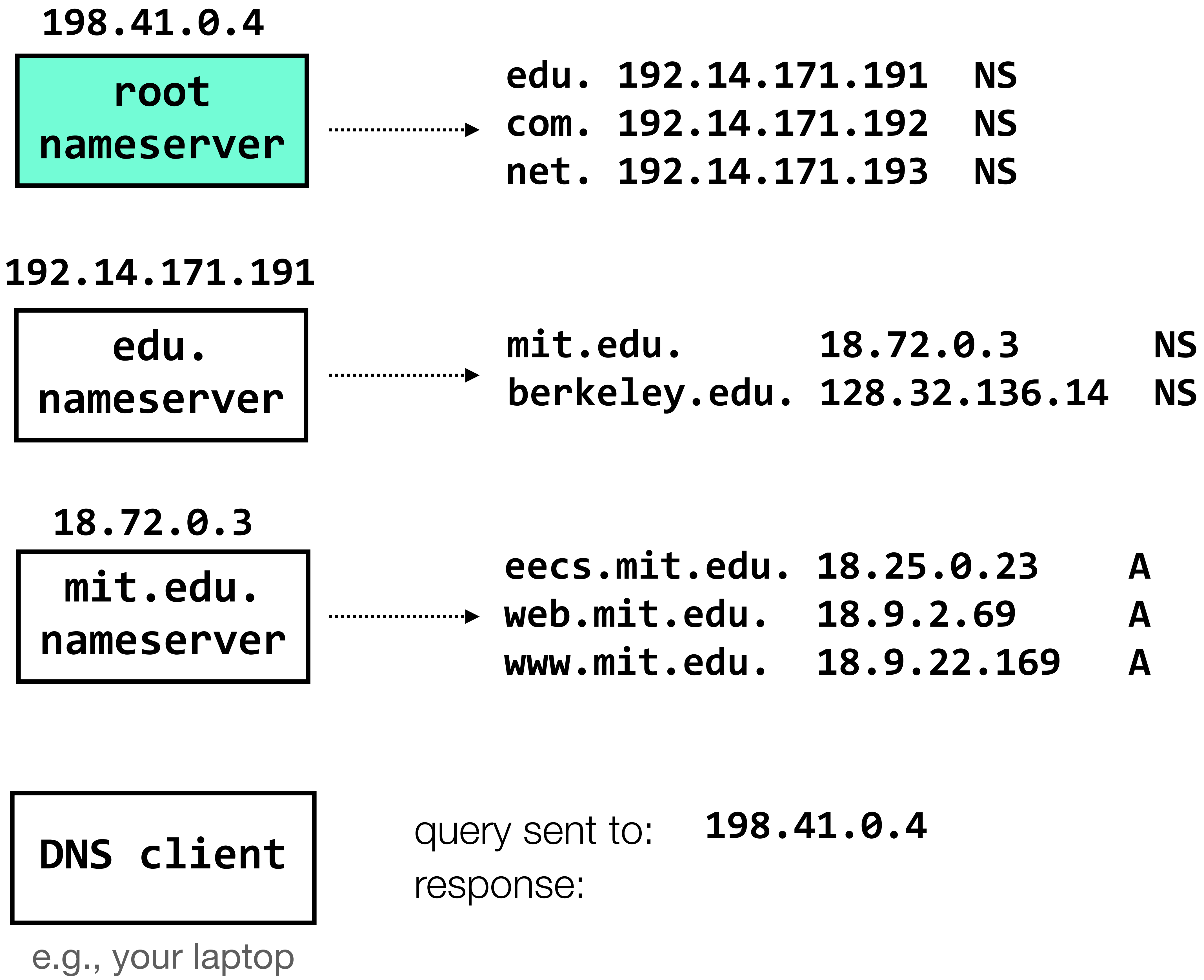
https://www.cnn.com/2024/03/04/business/red-sea-cables-cut-internet/index.html

6.1800 in the past

the **domain name system (DNS)**, which maps **hostnames** (eecs.mit.edu) to **IP addresses** (18.25.0.23)



a partial view of the DNS hierarchy. each box represents a **zone**. name servers within a zone keep track of that zone's mappings



how does the DNS client's query *get* to 198.41.0.4?

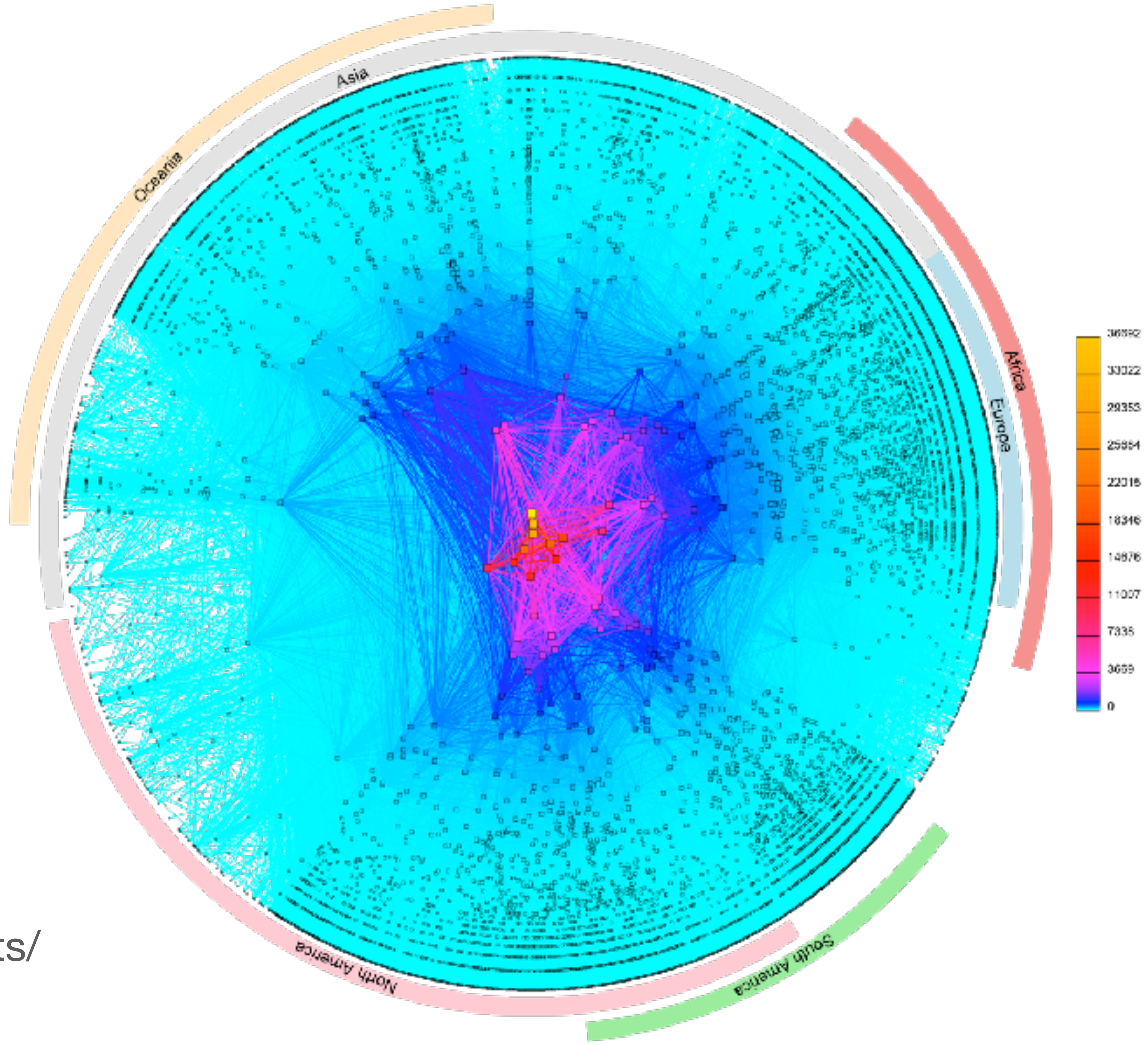
1970s: ARPAnet — hosts.txt

1978: flexibility and layering — distance-vector routing

early 80s: growth → change — TCP, UDP; OSPF, EGP, DNS

late 80s: growth → problems — congestion collapse; policy routing; CIDR

1993: commercialization



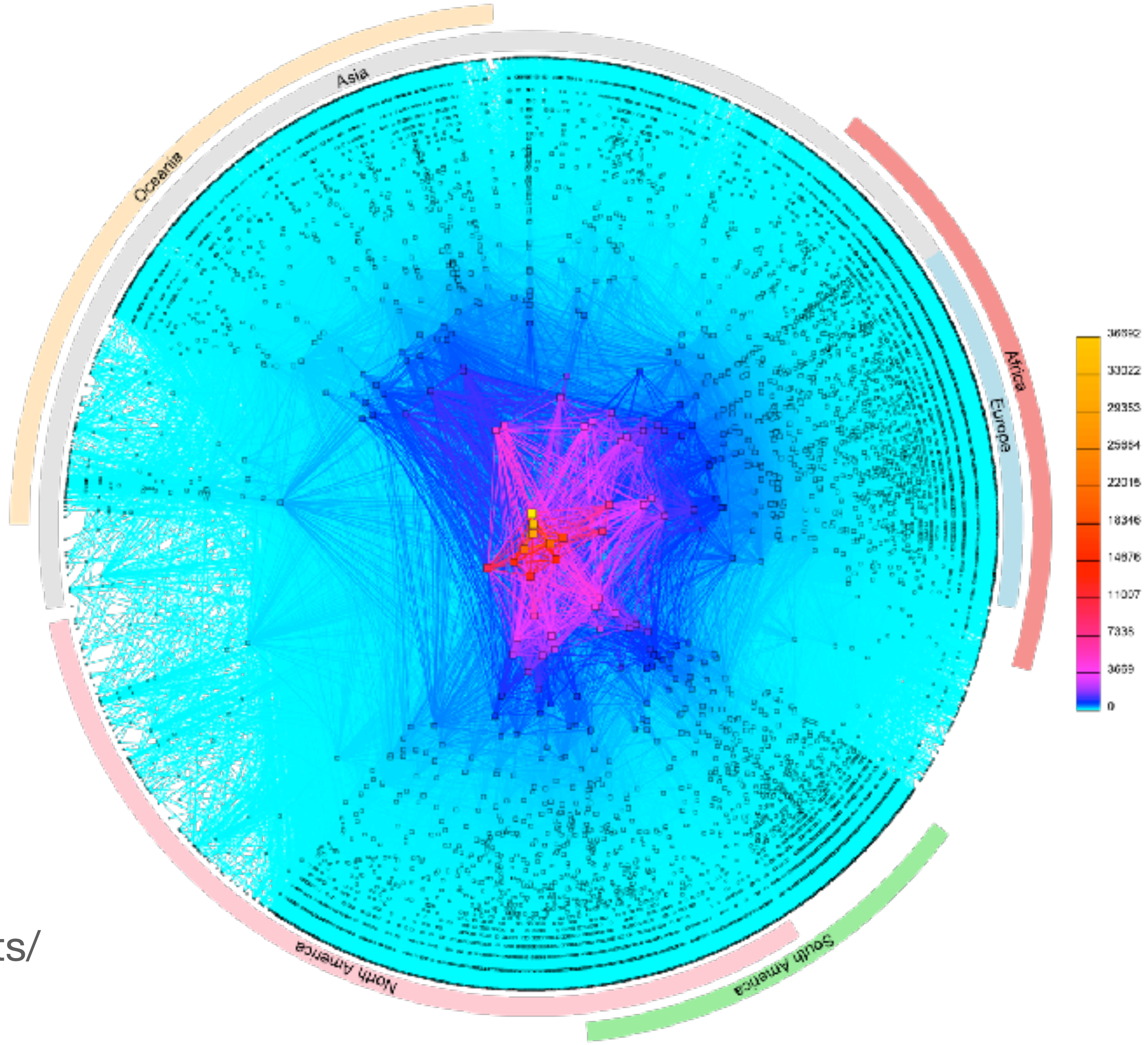
CAIDA's IPv4 AS Core, January 2020 (https://www.caida.org/projects/cartography/as-core/2020/)

on the Internet, we have to solve all of the "normal" networking problems (addressing, routing, transport) **at massive scale, while supporting a diverse group of applications and competing economic interests**

`application`	`the things that actually generate traffic`
`transport`	`sharing the network, reliability (or not)` *examples: TCP, UDP*
`network`	`naming, addressing, routing` *examples: IP*
`link`	`communication between two directly-connected nodes` *examples: ethernet, bluetooth, 802.11 (wifi)*

1970s: ARPAnet — 1978: flexibility and layering — early 80s: growth → change — late 80s: growth → problems — 1993: commercialization

hosts.txt — **distance-vector routing** — TCP, UDP — **OSPF**, EGP, DNS (a link-state routing protocol) — congestion collapse — policy routing — CIDR



CAIDA's IPv4 AS Core, January 2020 (https://www.caida.org/projects/cartography/as-core/2020/)

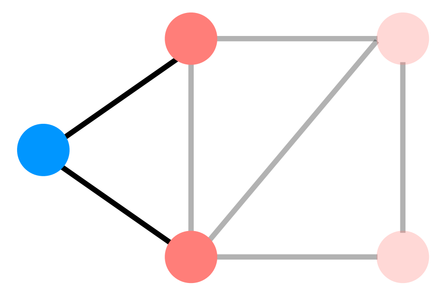
today: routing in general
(not specifically on the Internet)

`application`	`the things that actually generate traffic`
`transport`	`sharing the network, reliability (or not)` *examples: TCP, UDP*
`network`	`naming, addressing, routing` *examples: IP*
`link`	`communication between two directly-connected nodes` *examples: ethernet, bluetooth, 802.11 (wifi)*

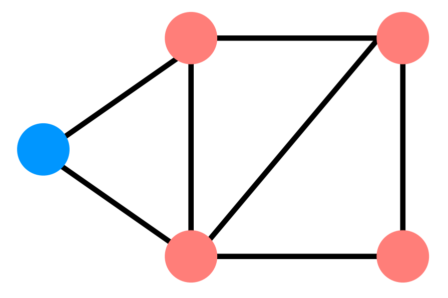
goal of a routing protocol: allow each switch to know, for every node `dst` in the network, a **minimum-cost** route to `dst`

distributed routing: nodes build up their own routing tables, rather than having tables given to them by a centralized authority

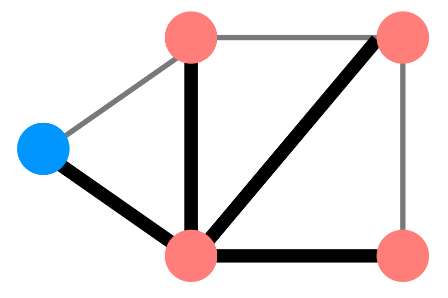
1. nodes learn about their neighbors via the `HELLO` protocol



2. nodes learn about other reachable nodes via advertisements



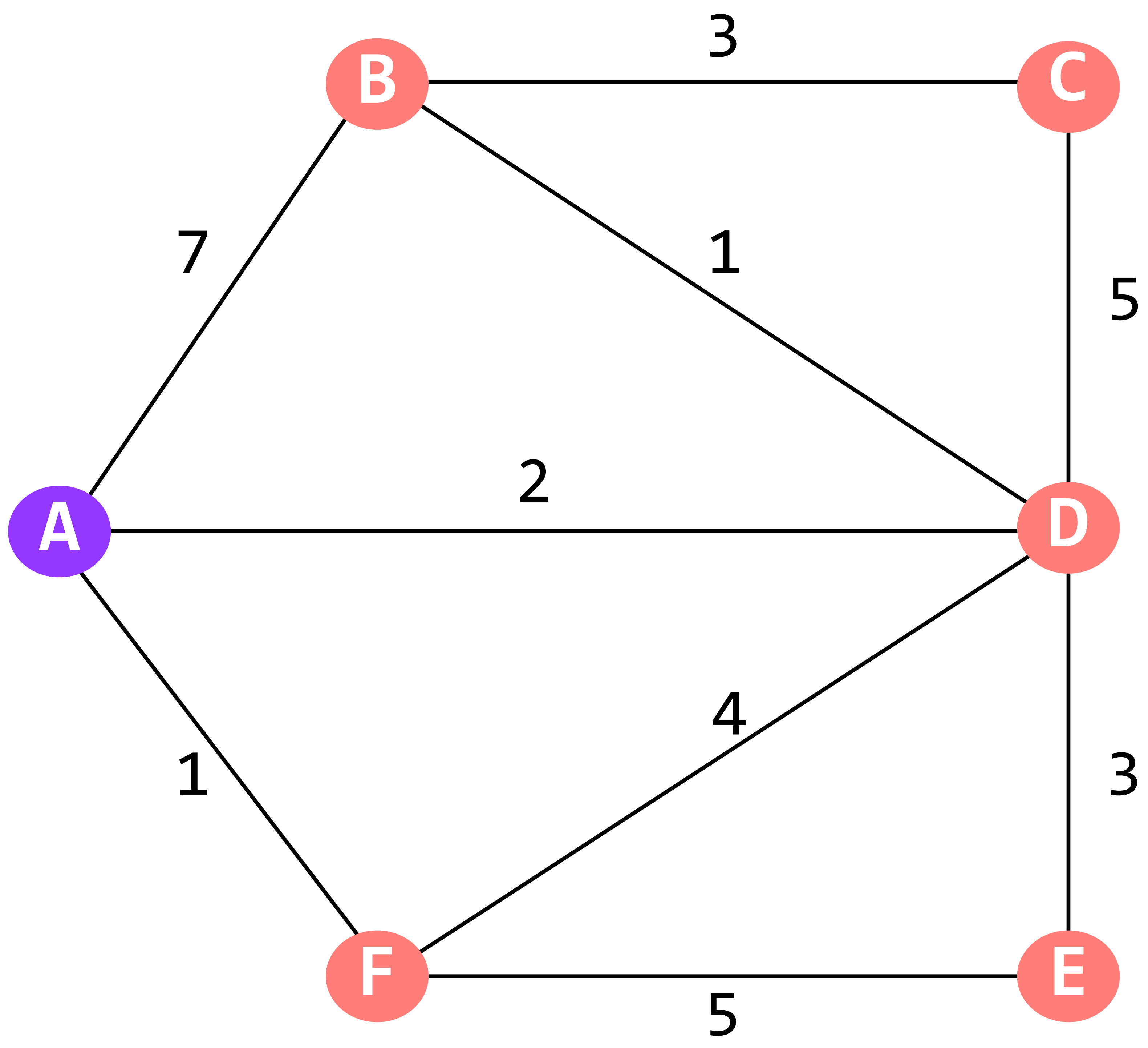
3. nodes determine the minimum-cost routes (of the routes they know about)



what the advertisements contain, and how the nodes use those advertisements to determine the min-cost routes, will change depending on the specific protocol

all of these steps happen periodically, which allows the routing protocol to detect and respond to failures, and adapt to other changes in the network

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm




```
A’s advertisement: [(B,7),(D,2),(F,1)]
```

link state

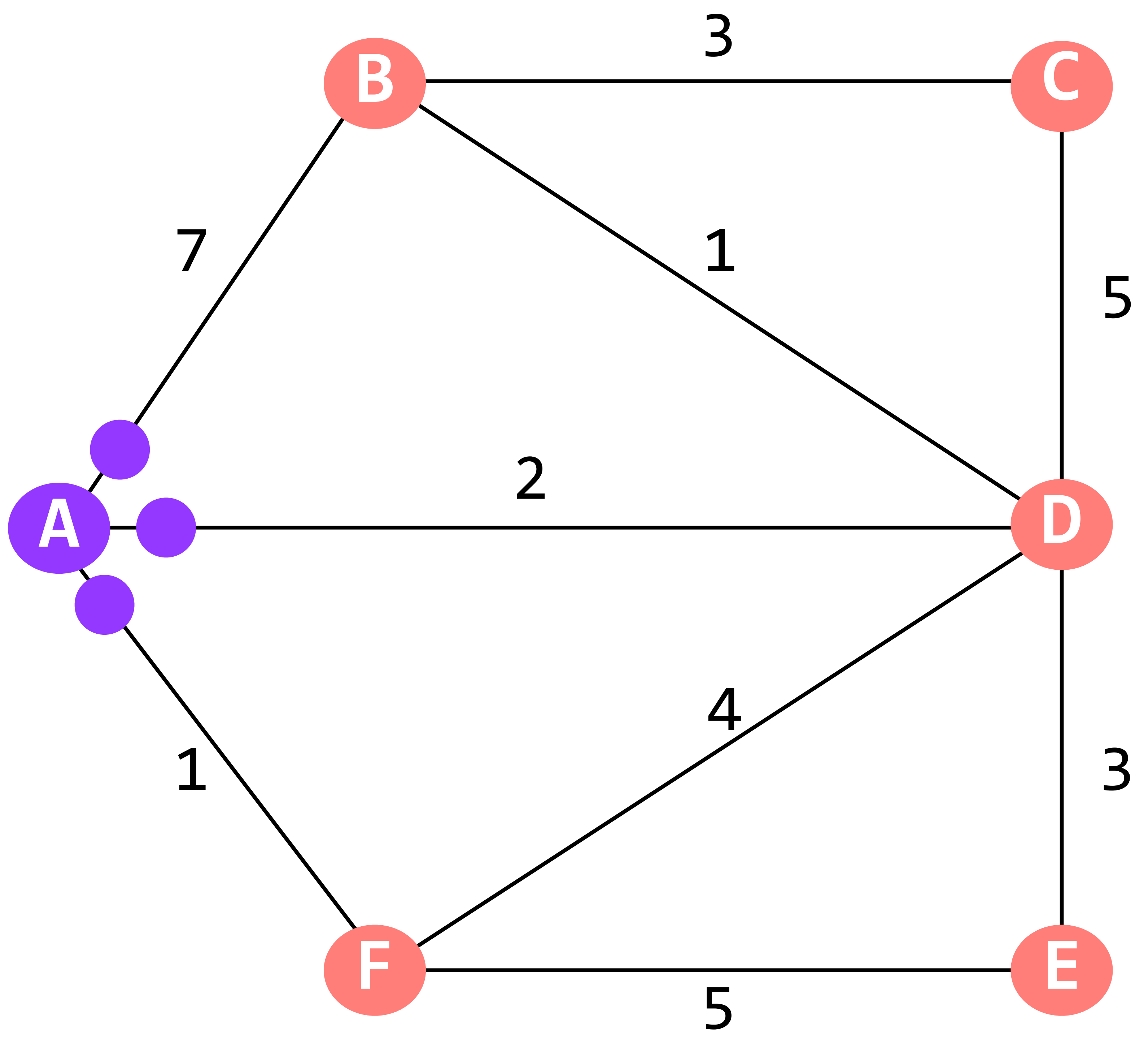
what’s in a node’s advertisement

its **link costs** to each of its **neighbors**

who gets a node’s advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm




```
A's advertisement: [(B,7),(D,2),(F,1)]
```

link state

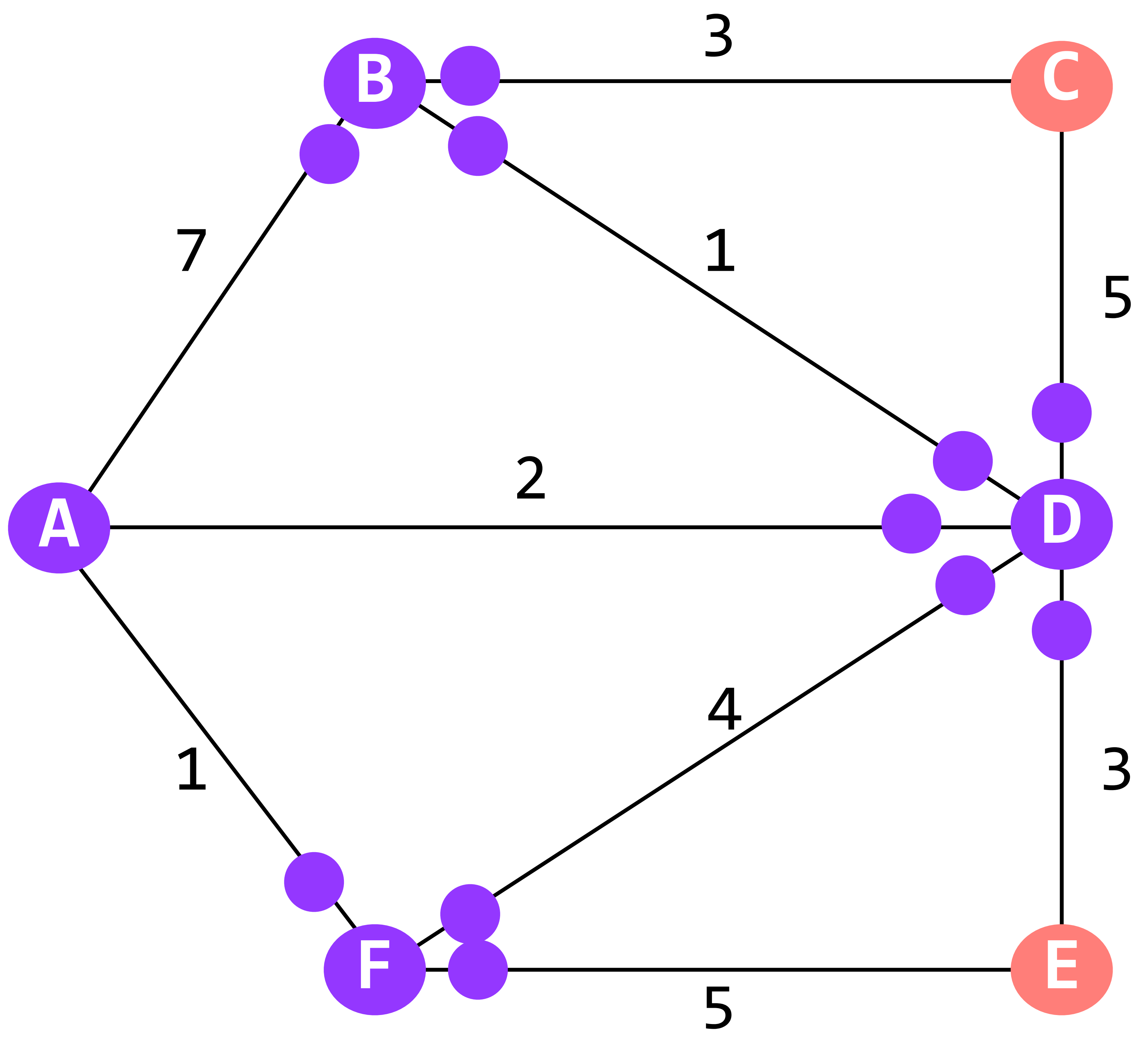
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm




```
A's advertisement: [(B,7),(D,2),(F,1)]
```

link state

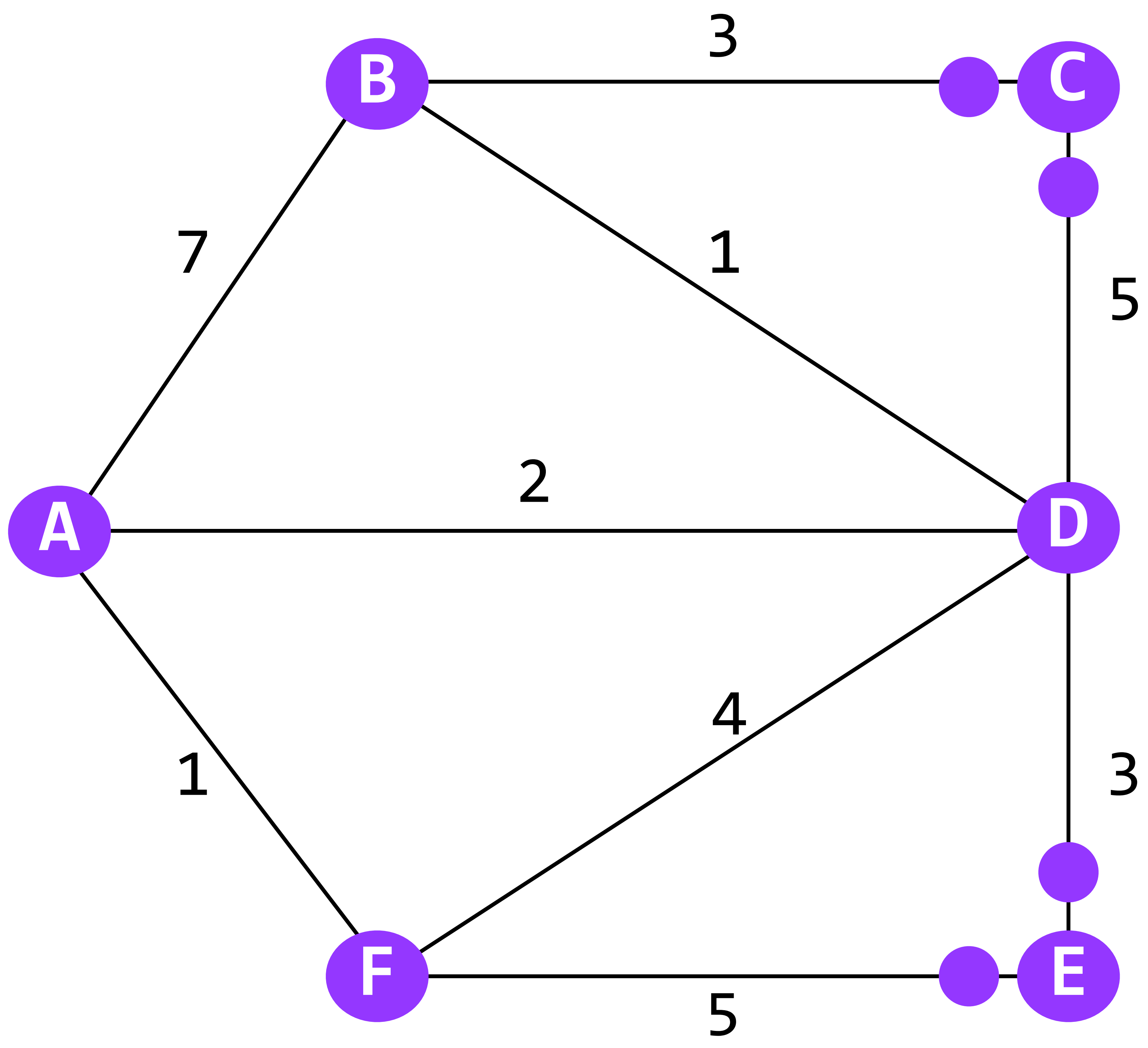
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A’s advertisement: [(B,7),(D,2),(F,1)]

link state

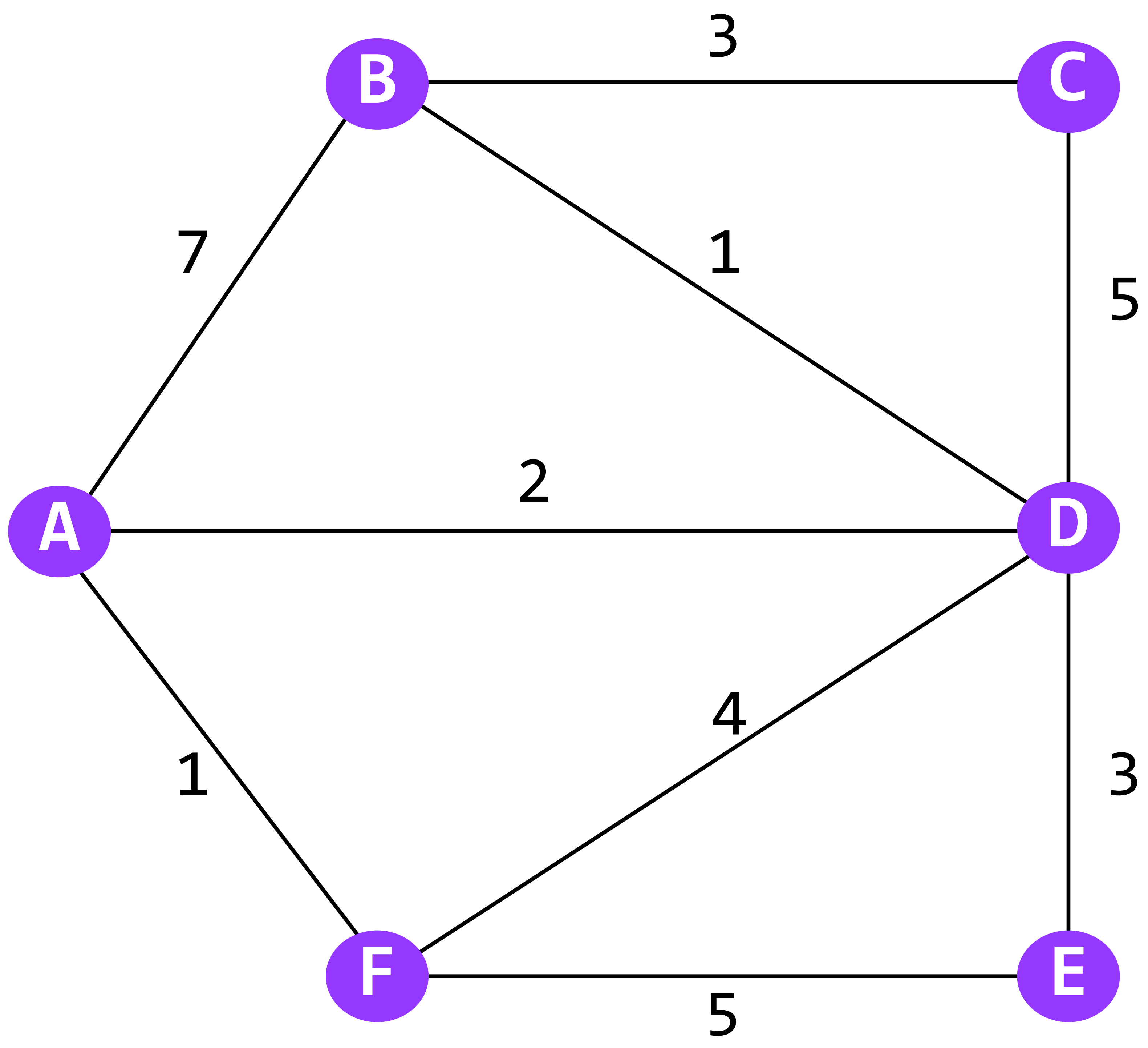
what’s in a node’s advertisement

its **link costs** to each of its **neighbors**

who gets a node’s advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



`A’s advertisement: [(B,7),(D,2),(F,1)]`

nodes keep track of which advertisements they’ve forwarded so that they don’t re-forward them

they can also be a bit smarter about flooding, and not forward an advertisement back to the node that sent it

link state

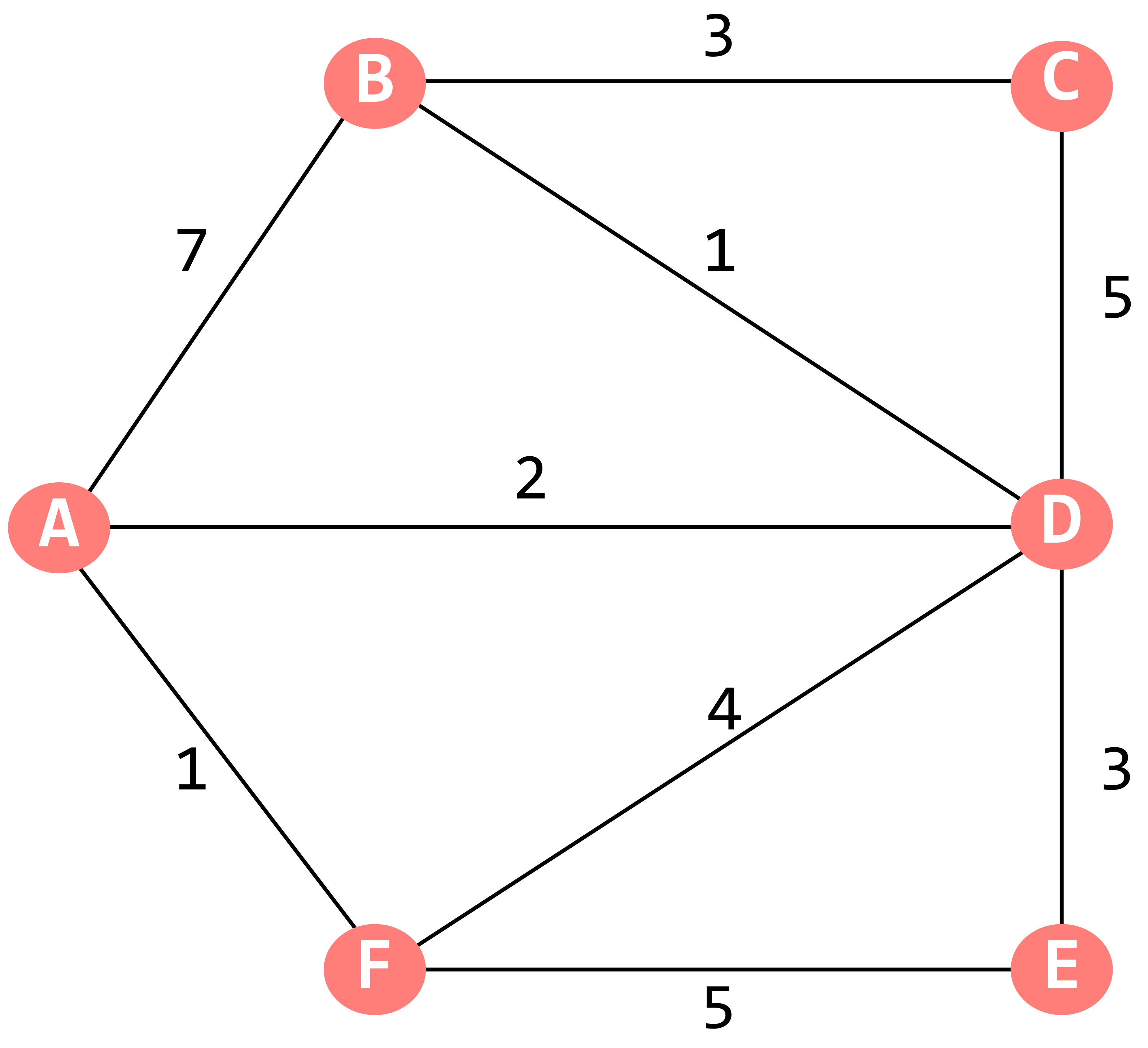
what’s in a node’s advertisement

its **link costs** to each of its **neighbors**

who gets a node’s advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



nodes *integrate* advertisements by running Dijkstra's Algorithm

link state

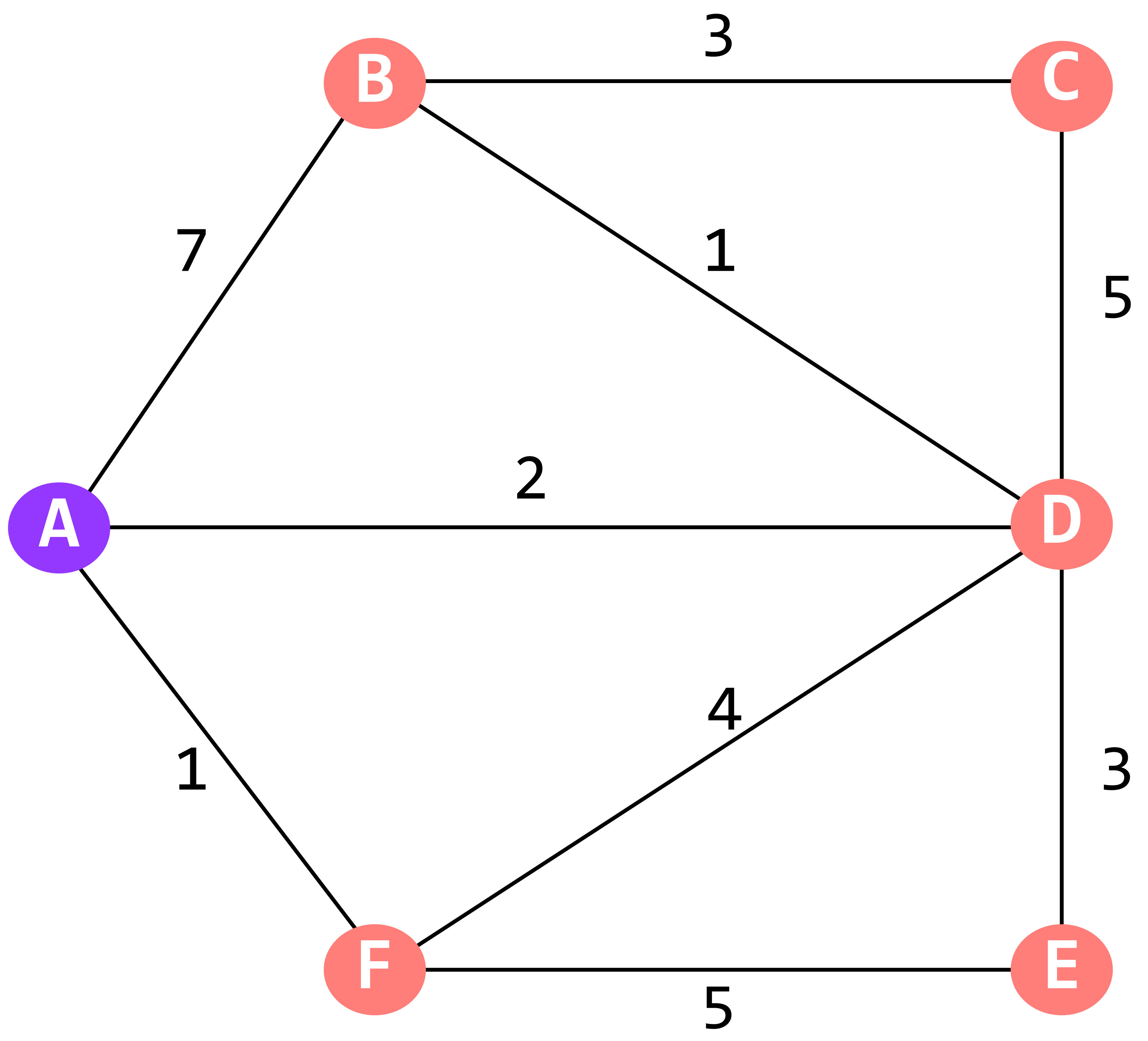
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-B	7
C	?	∞
D	A-D	2
E	?	∞
F	A-F	1

link state

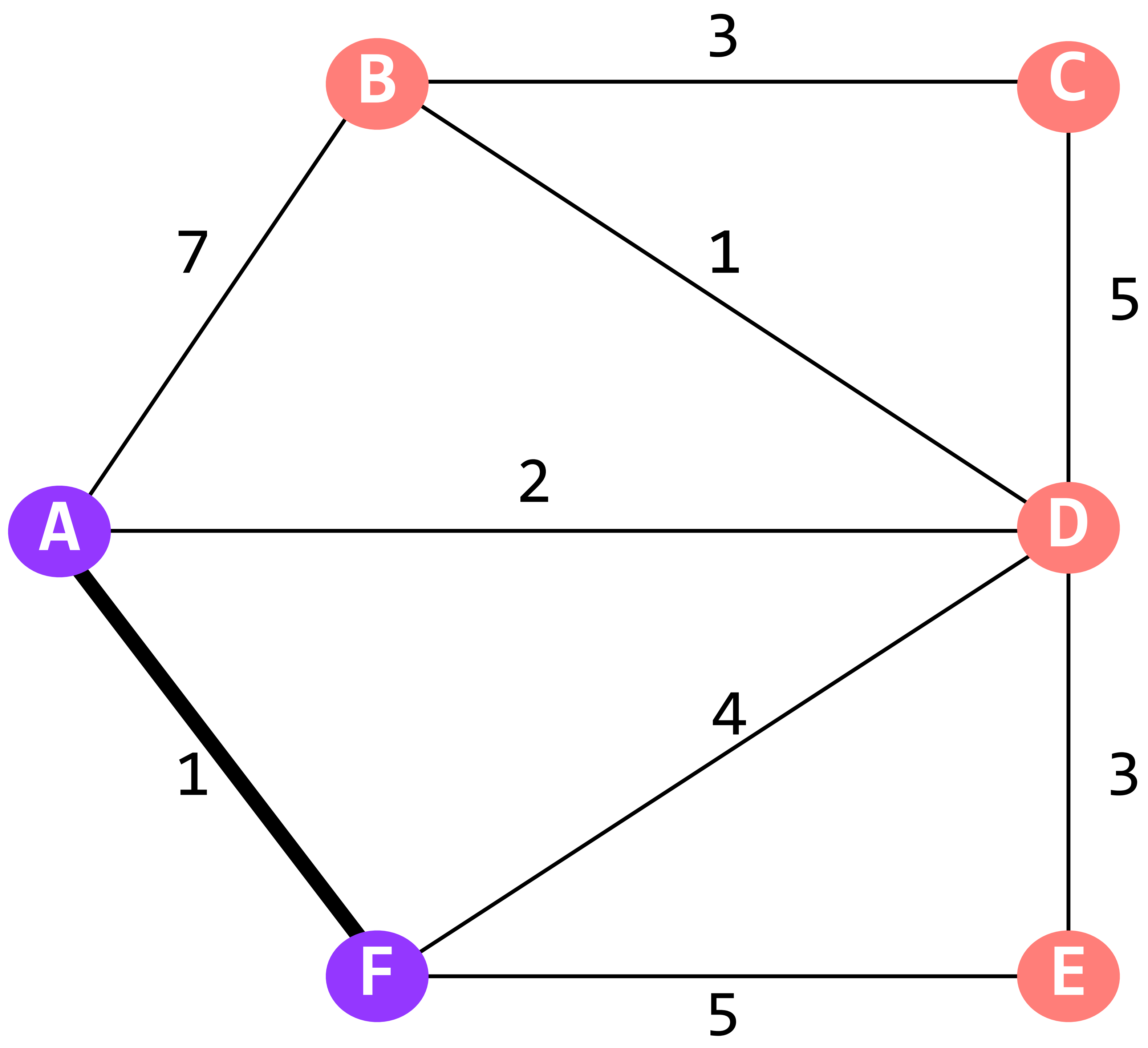
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-B	7
C	?	∞
D	A-D	2
E	?	∞
F	A-F	1

link state

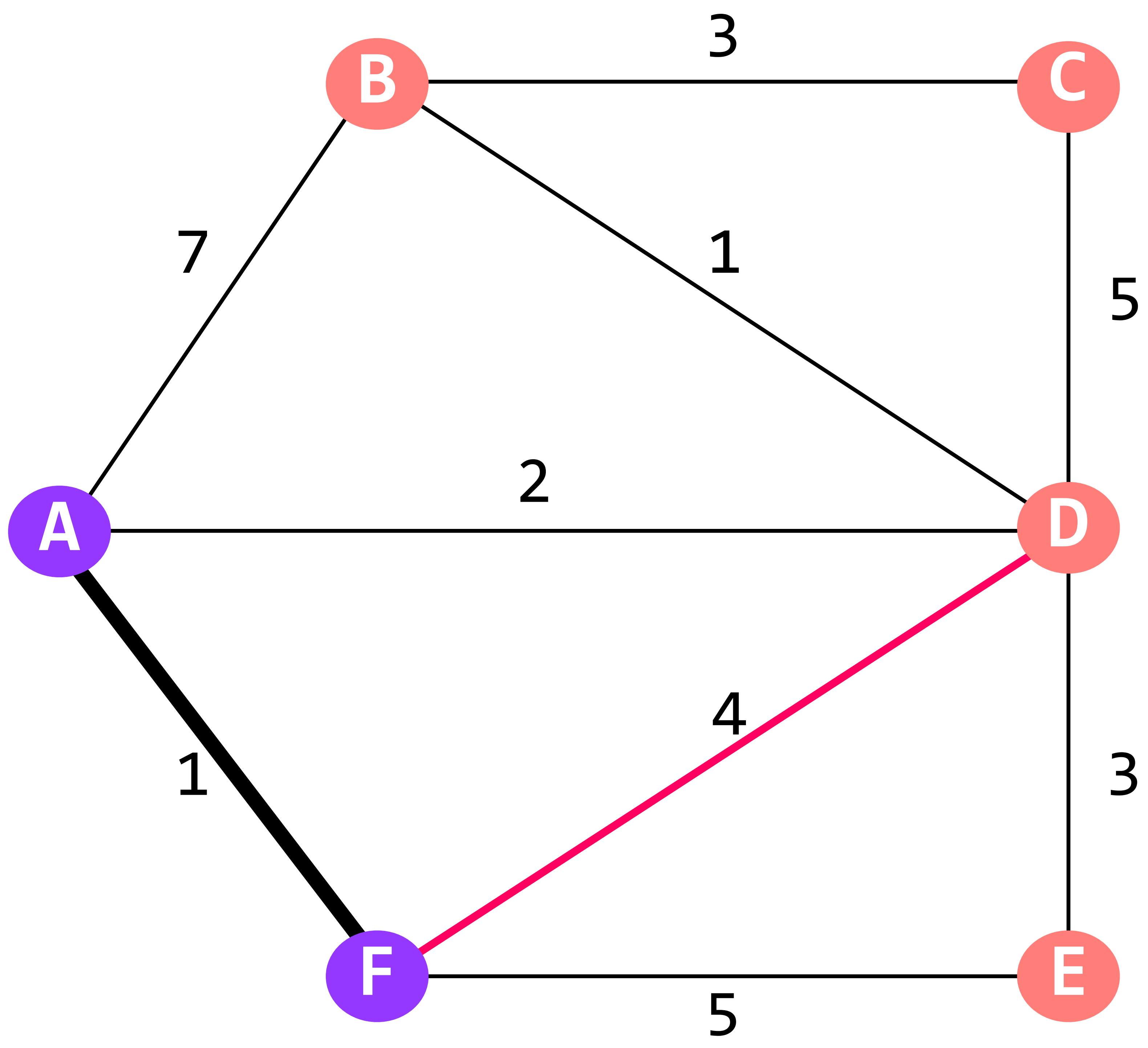
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-B	7
C	?	∞
D	A-D	2
E	?	∞
F	A-F	1

F does not provide A with a better route to D

link state

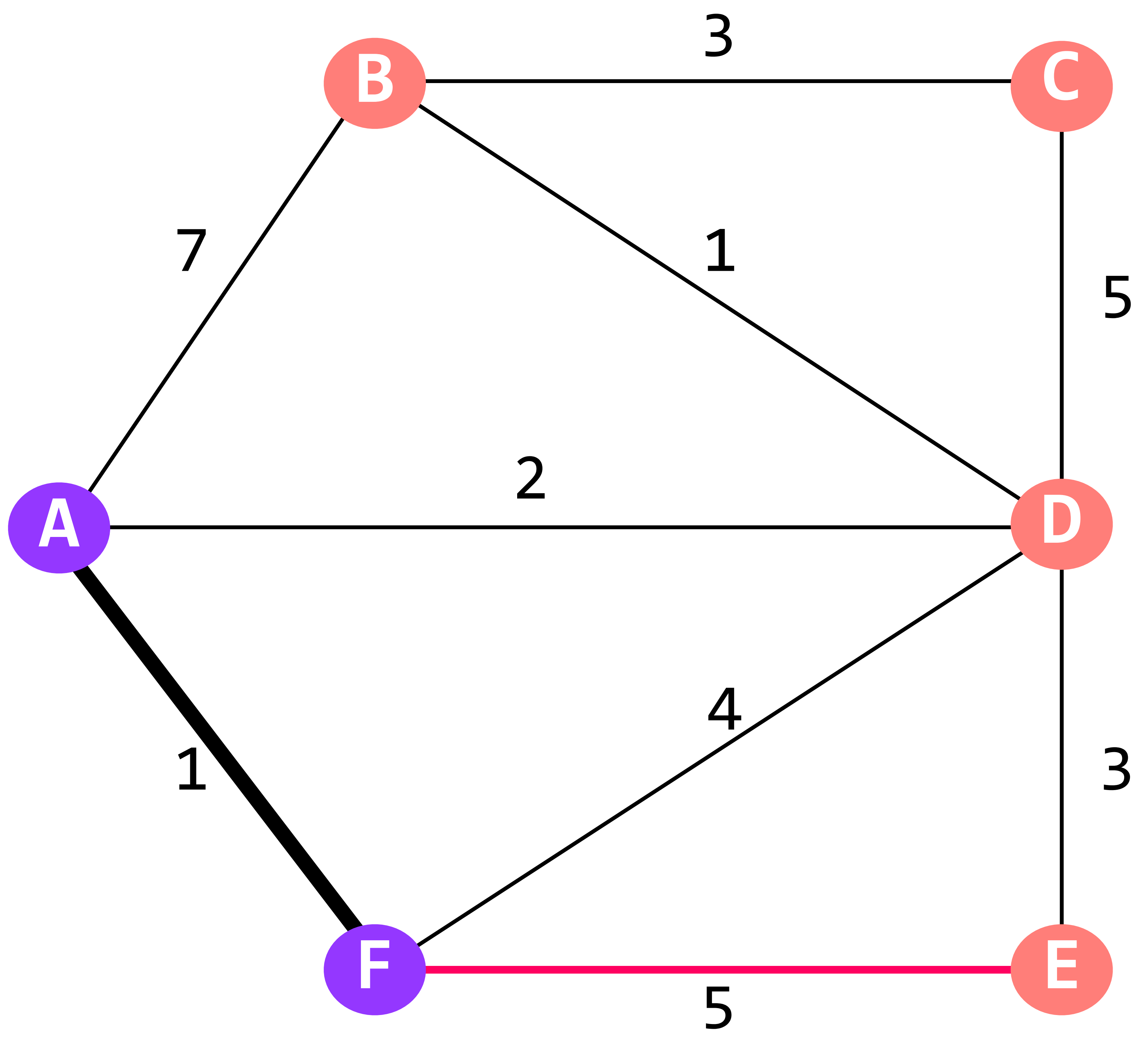
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-B	7
C	?	∞
D	A-D	2
E	A-F	6
F	A-F	1

E: 6 = the cost from A to F + the cost from F to E

link state

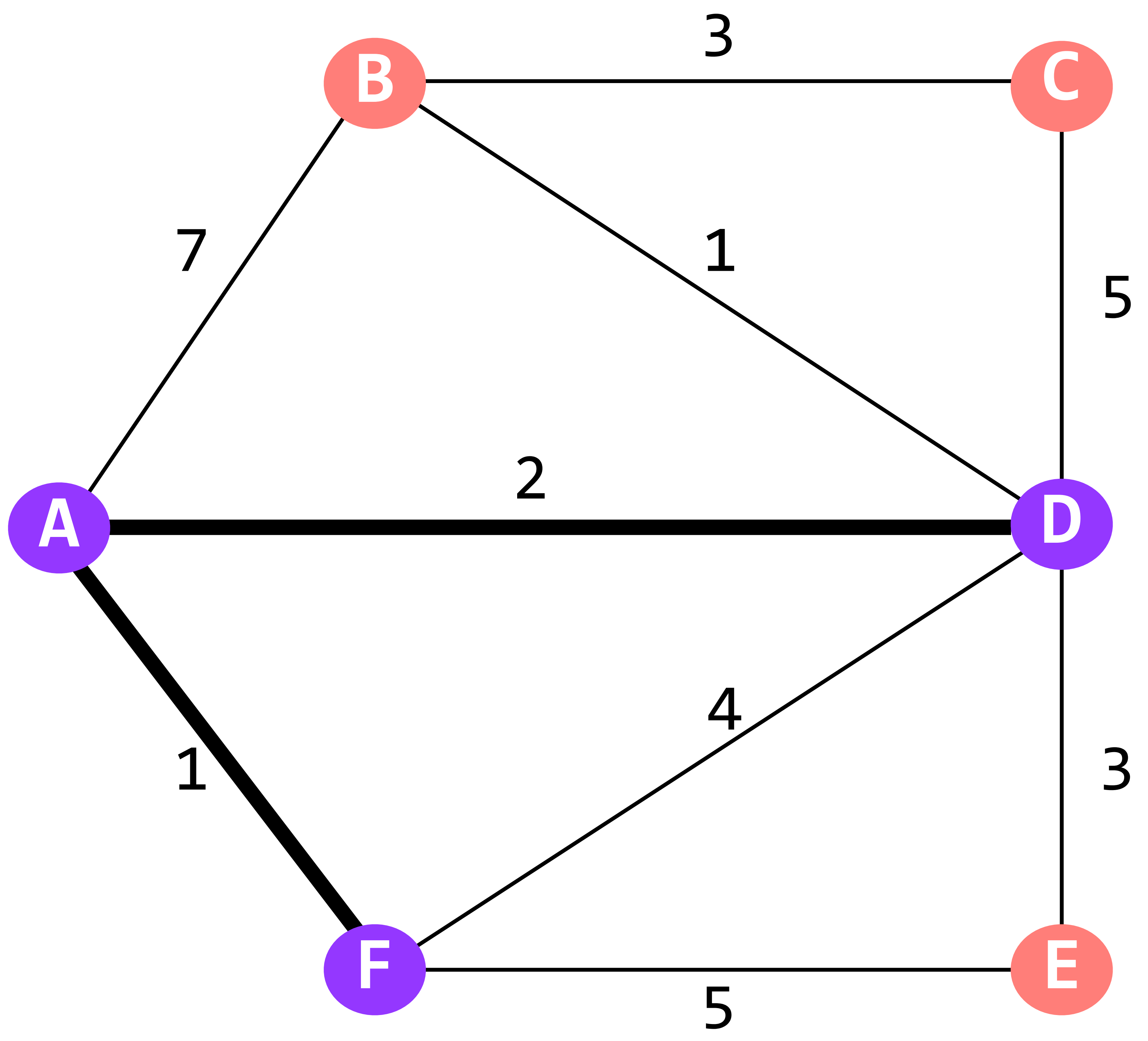
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-B	7
C	?	∞
D	A-D	2
E	A-F	6
F	A-F	1

question: what will A's routing table look like after we're done visiting all of D's neighbors?

link state

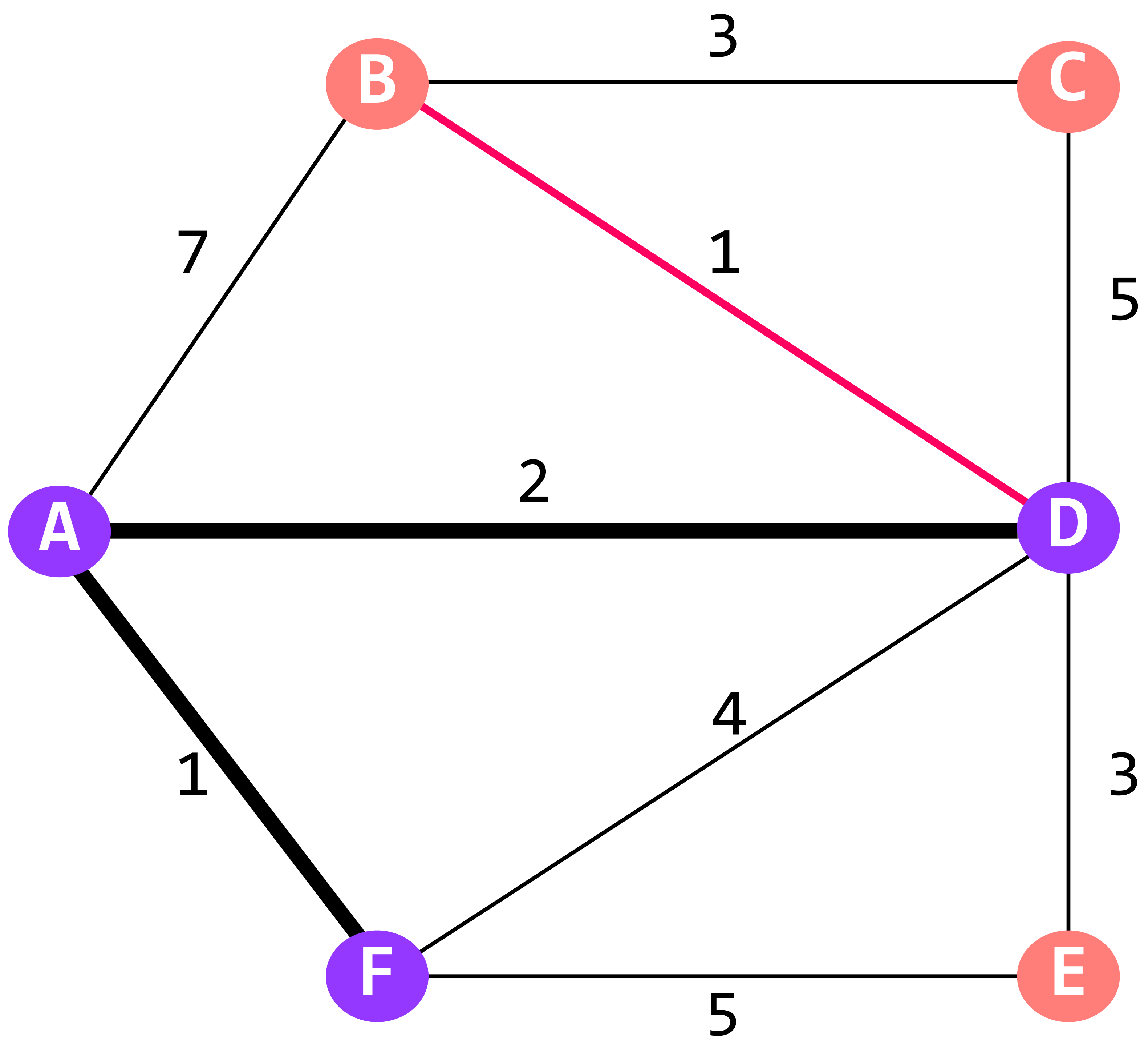
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-D	3
C	?	∞
D	A-D	2
E	A-F	6
F	A-F	1

link state

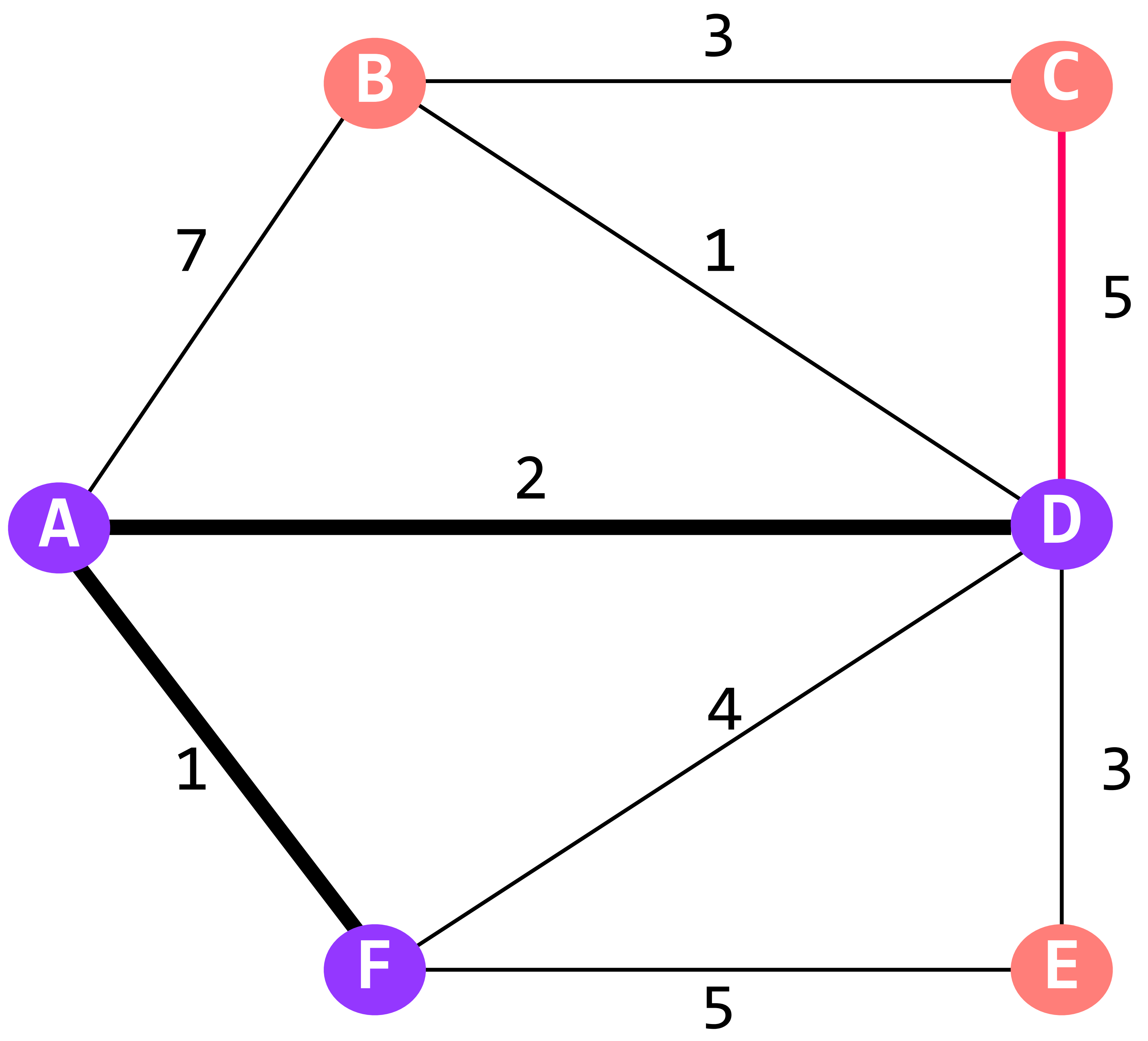
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
E	A-F	6
F	A-F	1

link state

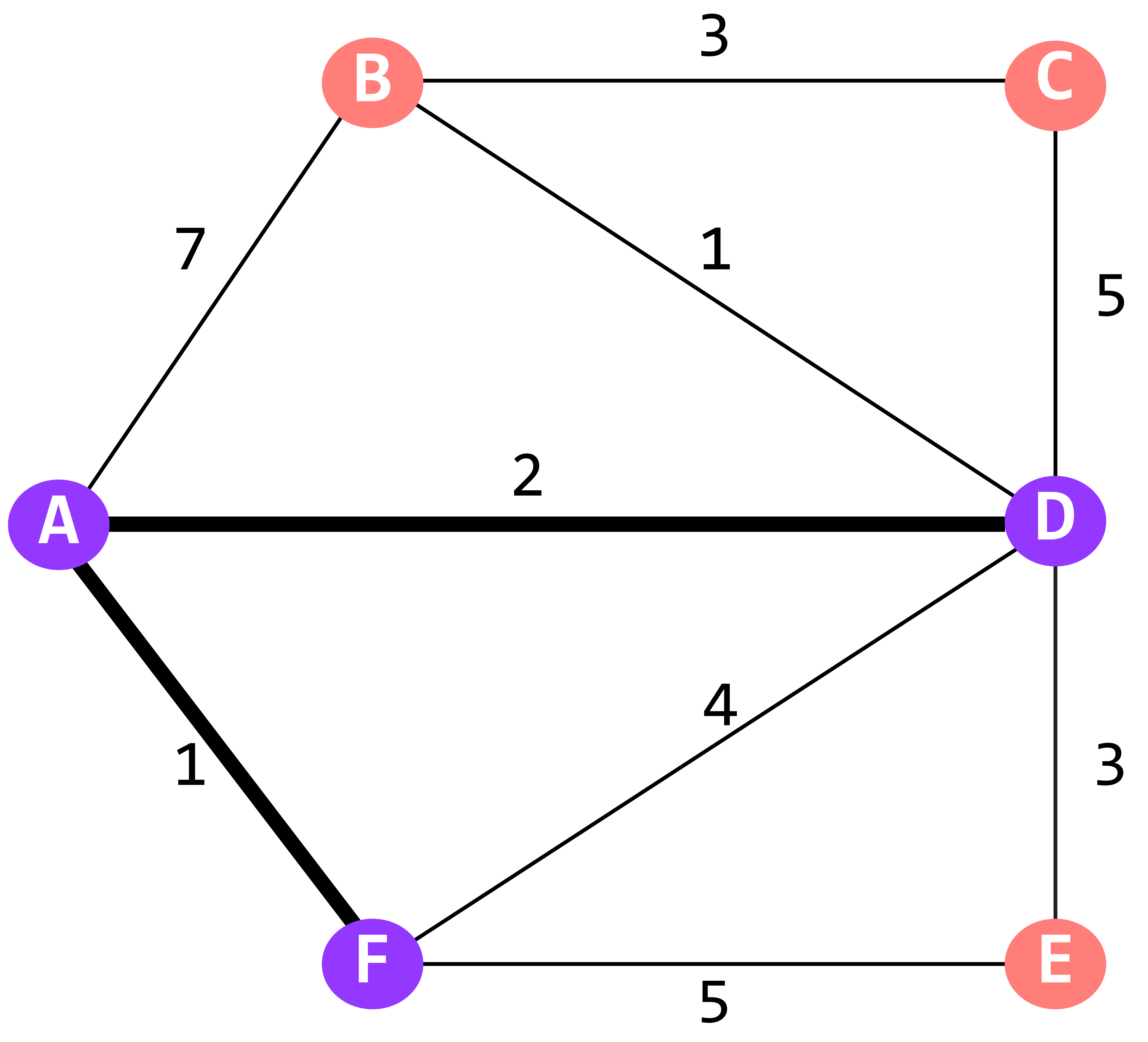
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
E	A-D	5
F	A-F	1

we don't need to "visit" F; we already know the shortest path to it

link state

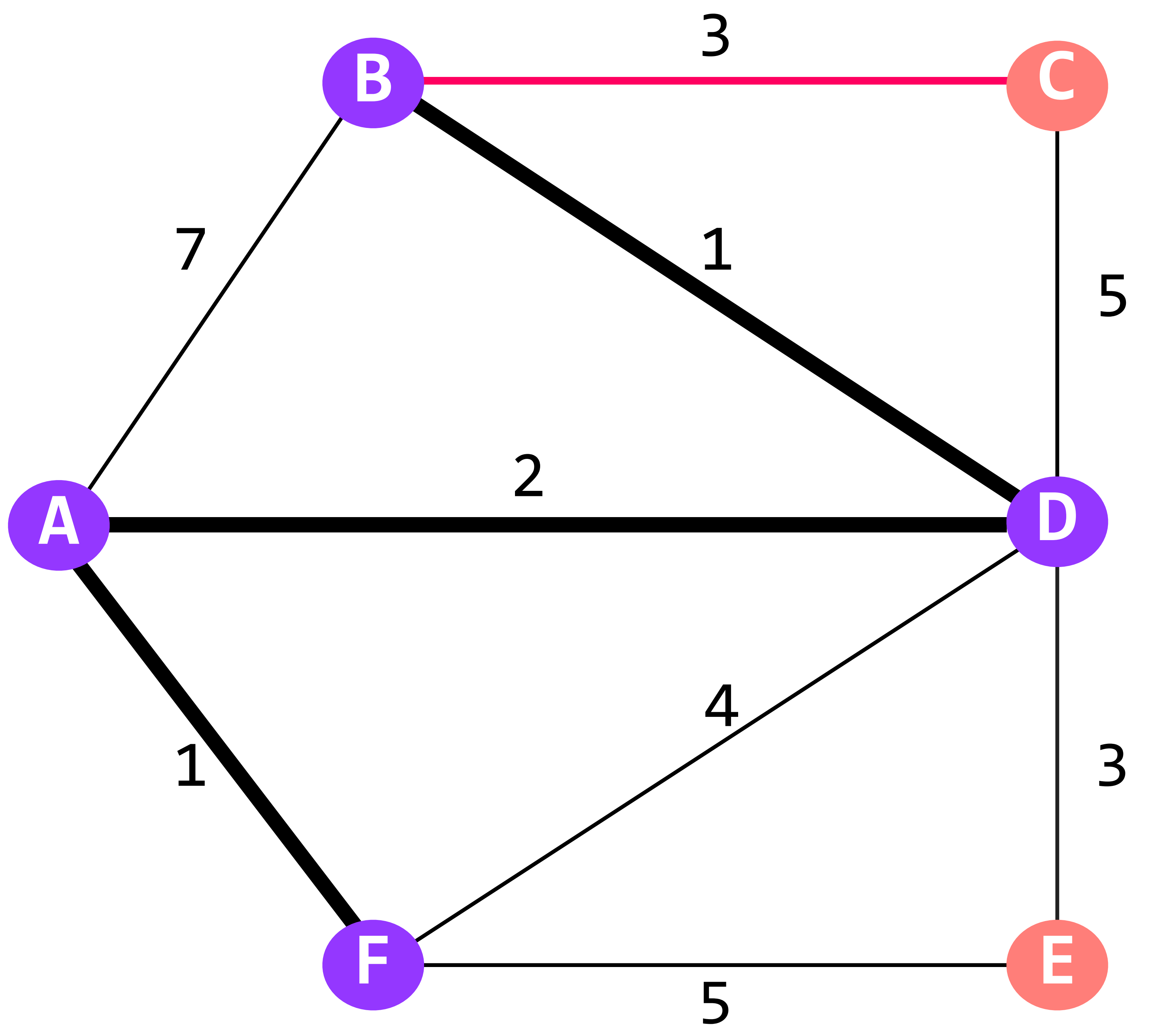
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-D	3
C	A-D	6
D	A-D	2
E	A-D	5
F	A-F	1

notice that A's *route* doesn't change, but the cost needs to update
(and the actual path of the packets from A to C has changed)

link state

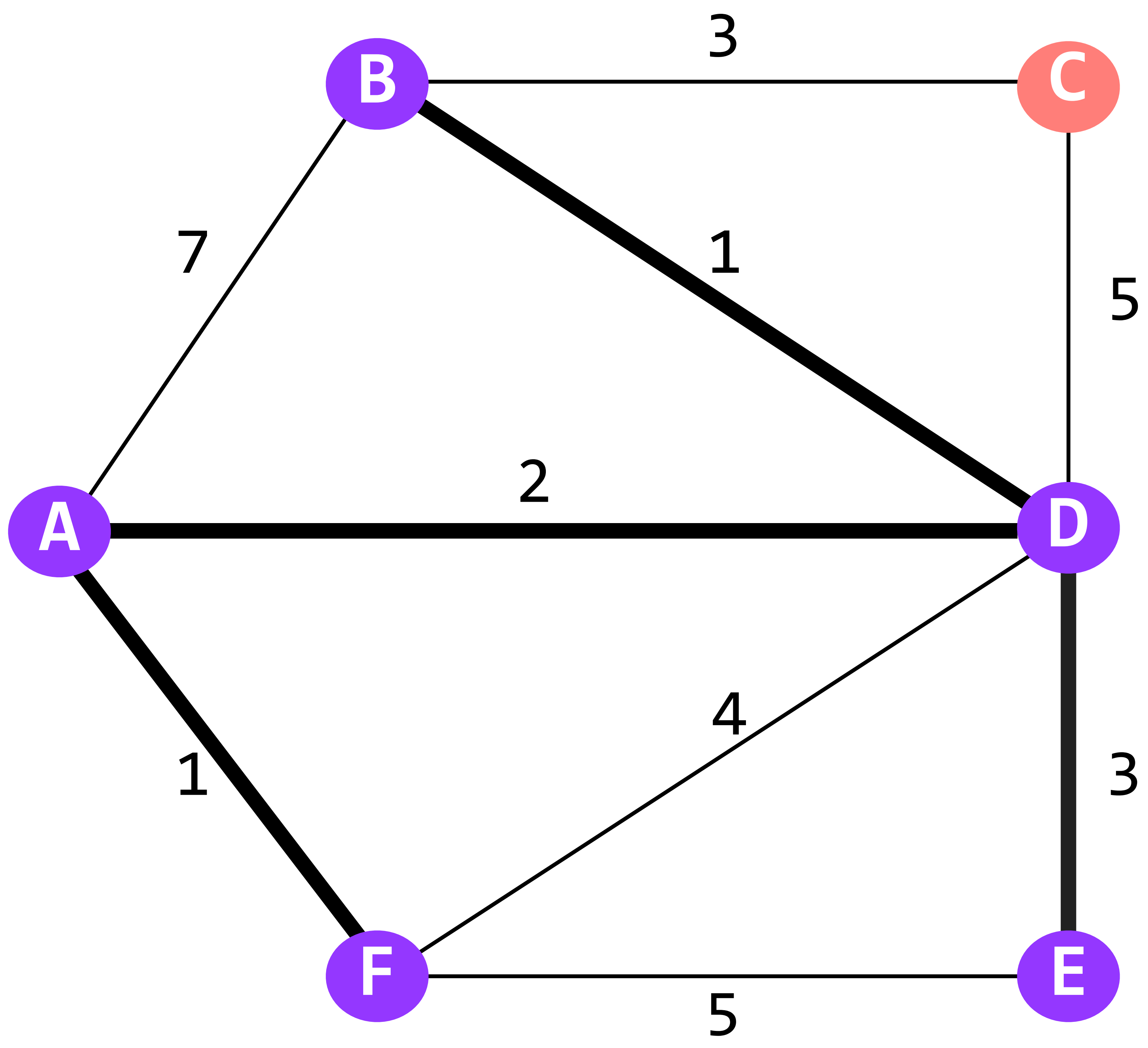
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-D	3
C	A-D	6
D	A-D	2
E	A-D	5
F	A-F	1

link state

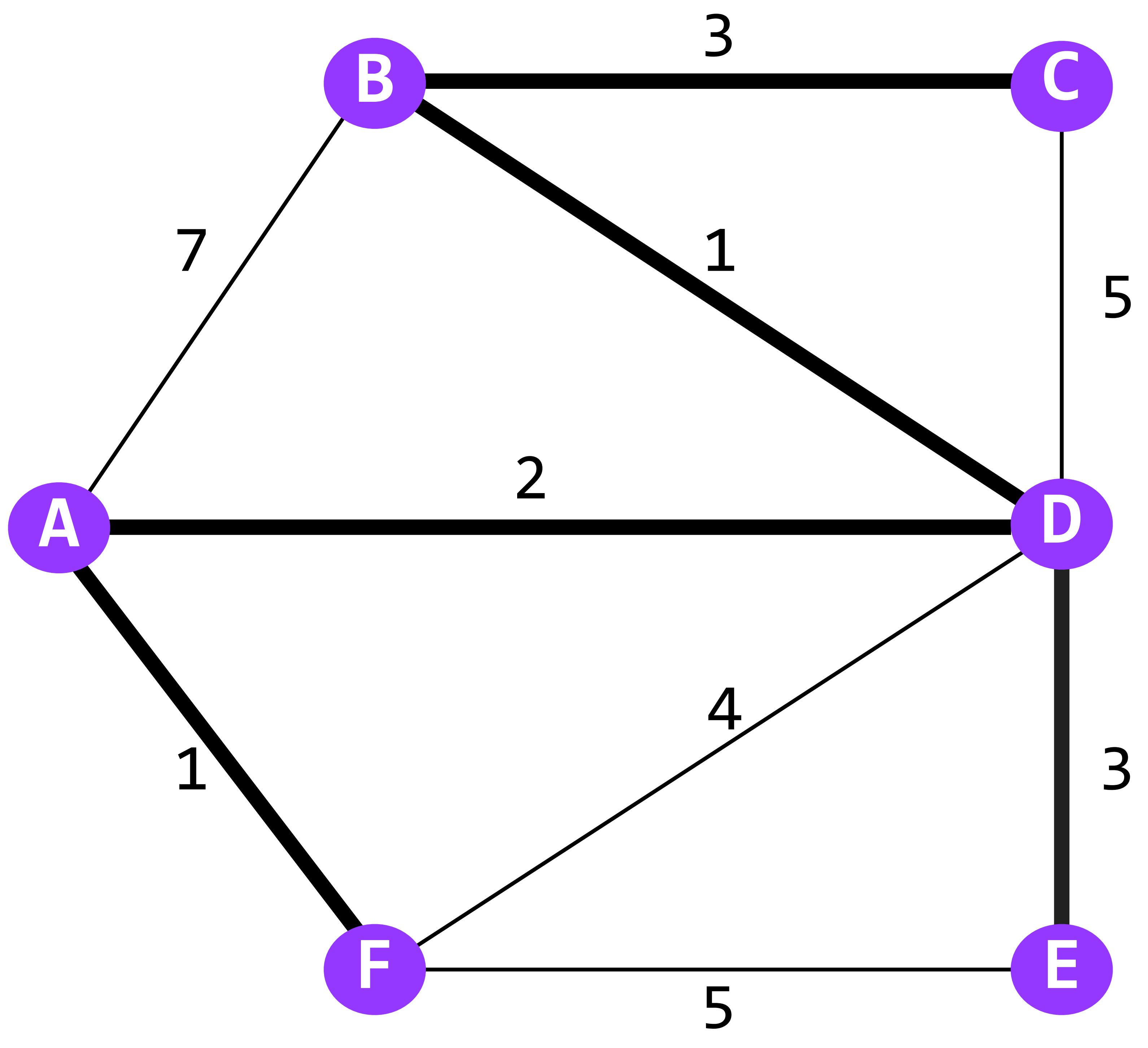
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



A's routing table

dst	route	cost
B	A-D	3
C	A-D	6
D	A-D	2
E	A-D	5
F	A-F	1

link state

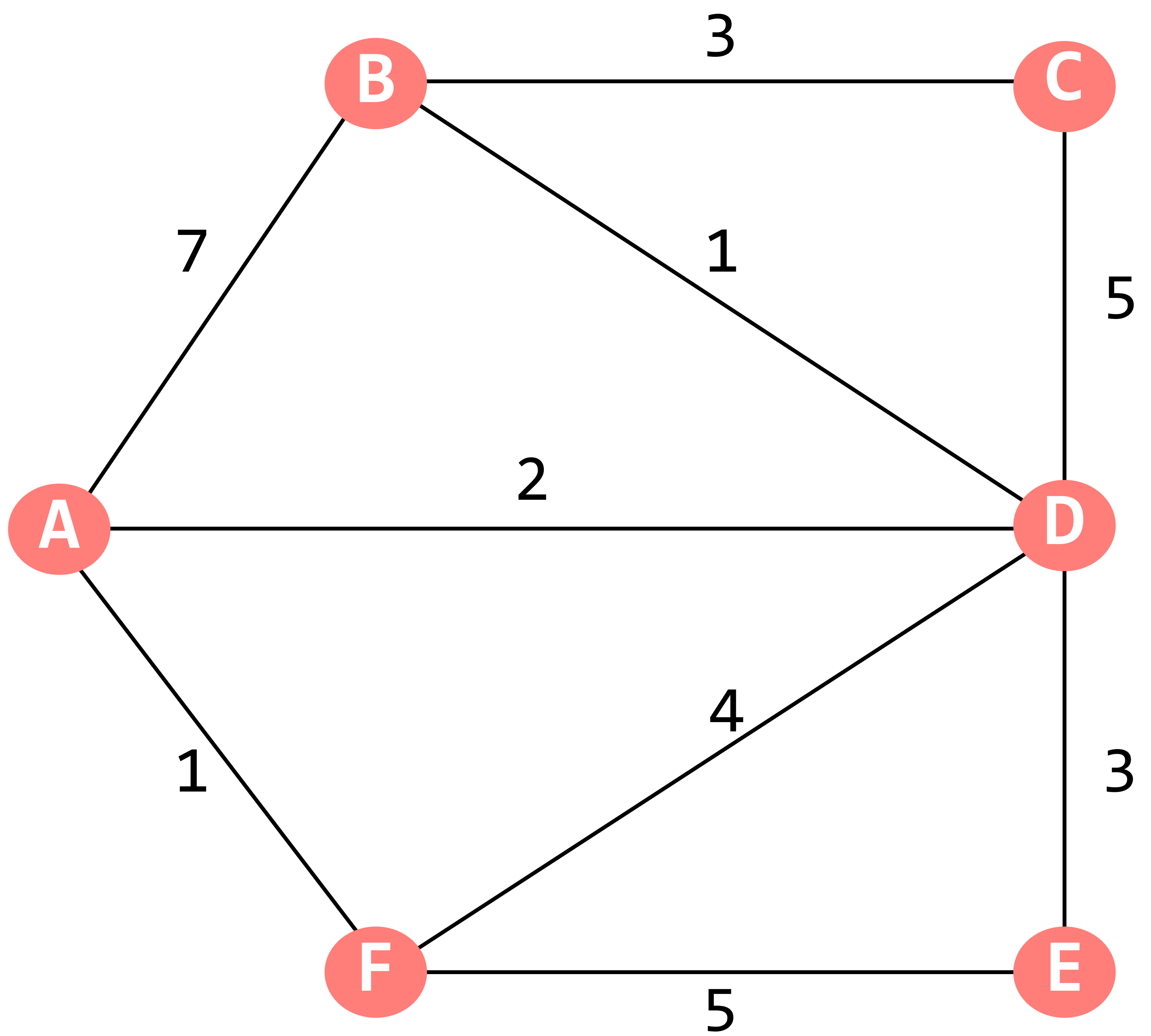
what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

link-state routing: disseminate full topology information so that nodes can run a shortest-path algorithm



link state

what's in a node's advertisement

its **link costs** to each of its **neighbors**

who gets a node's advertisement

effectively, **every other node** (via flooding)

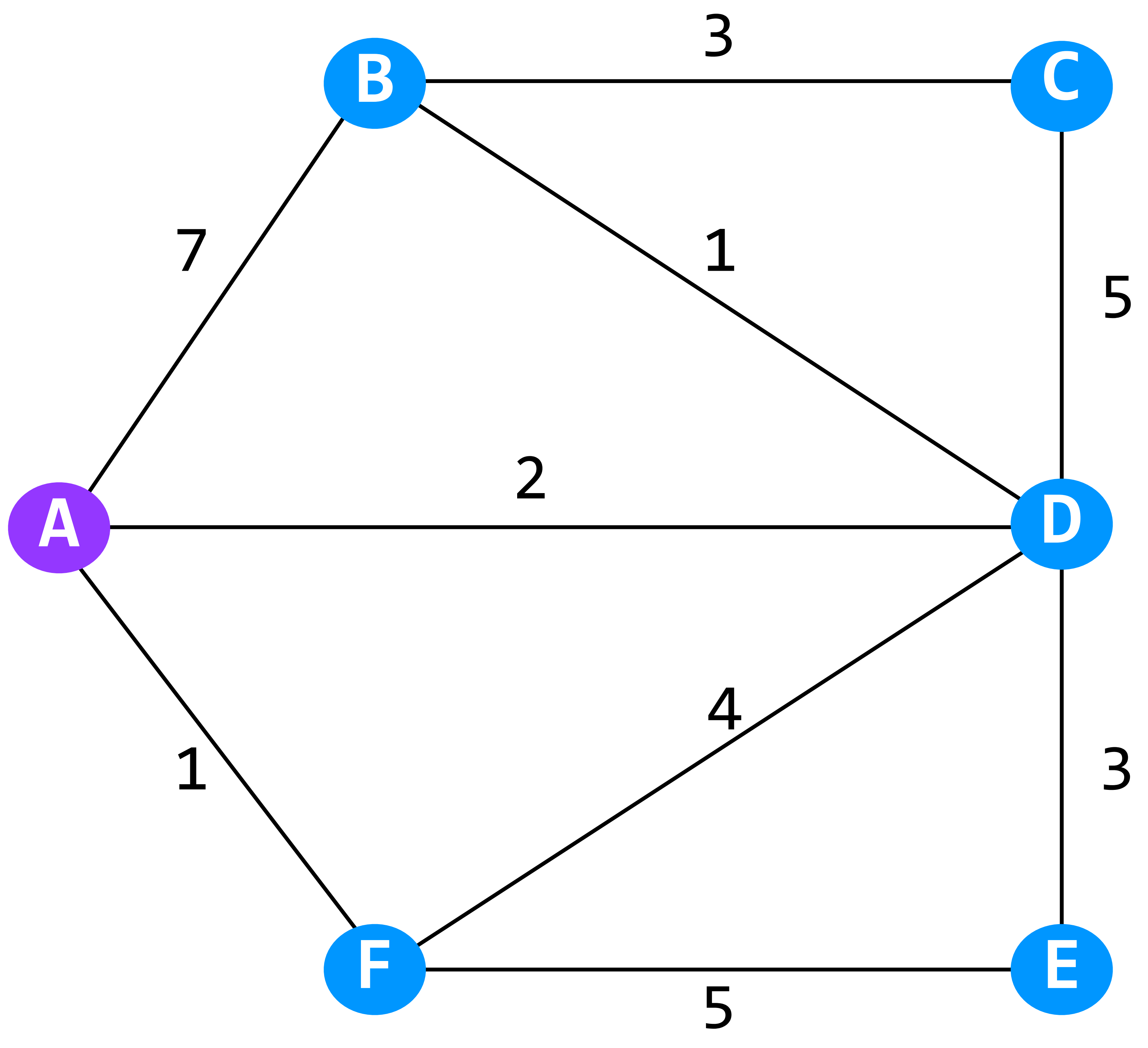
what happens when things fail?

flooding makes link-state routing very resilient to failure

what limits scale?

the **overhead** of flooding

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



`A's first advertisement: [(B,7),(D,2),(F,1)]`

A could also include `(A,0)` here

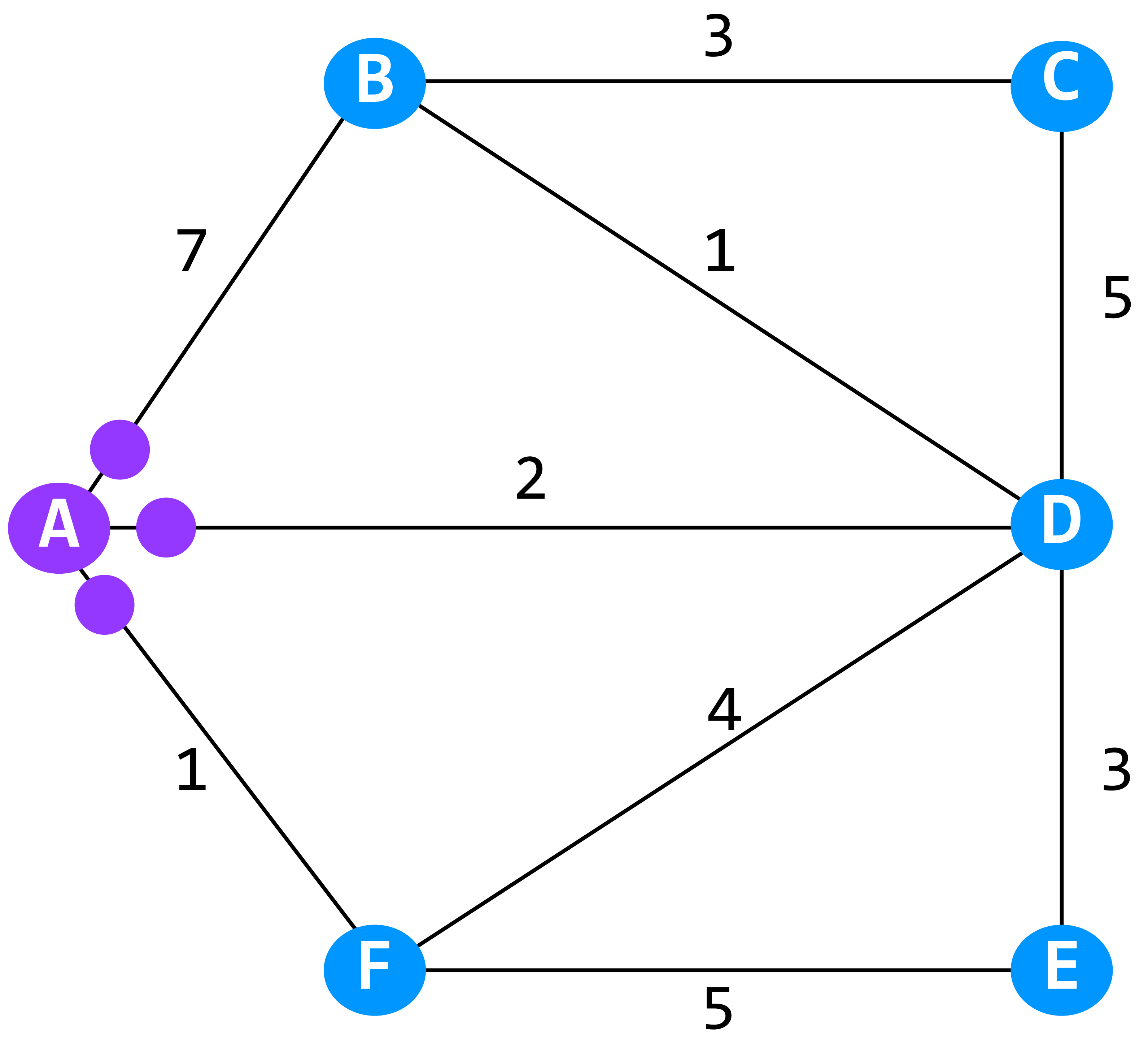
A's routing table

dst	route	cost
B	A-B	7
D	A-D	2
F	A-F	1

A's advertisement reflects its routing table, and right now, A only knows about its neighbors

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



`A's first advertisement: [(B,7),(D,2),(F,1)]`

A could also include `(A,0)` here

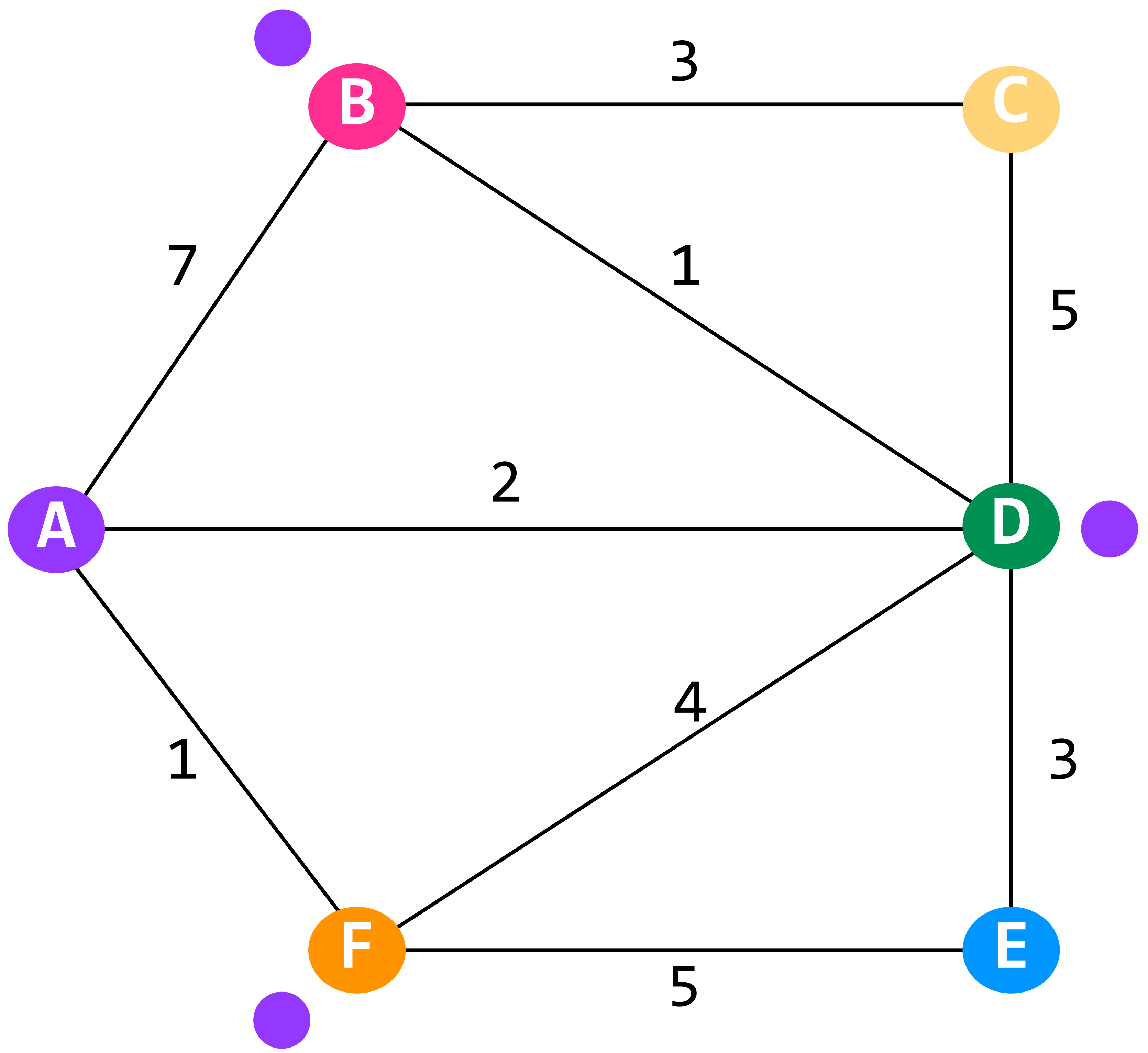
A's routing table

dst	route	cost
B	A-B	7
D	A-D	2
F	A-F	1

A's advertisement reflects its routing table, and right now, A only knows about its neighbors

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology




```
A's first advertisement: [(B,7),(D,2),(F,1)]
```

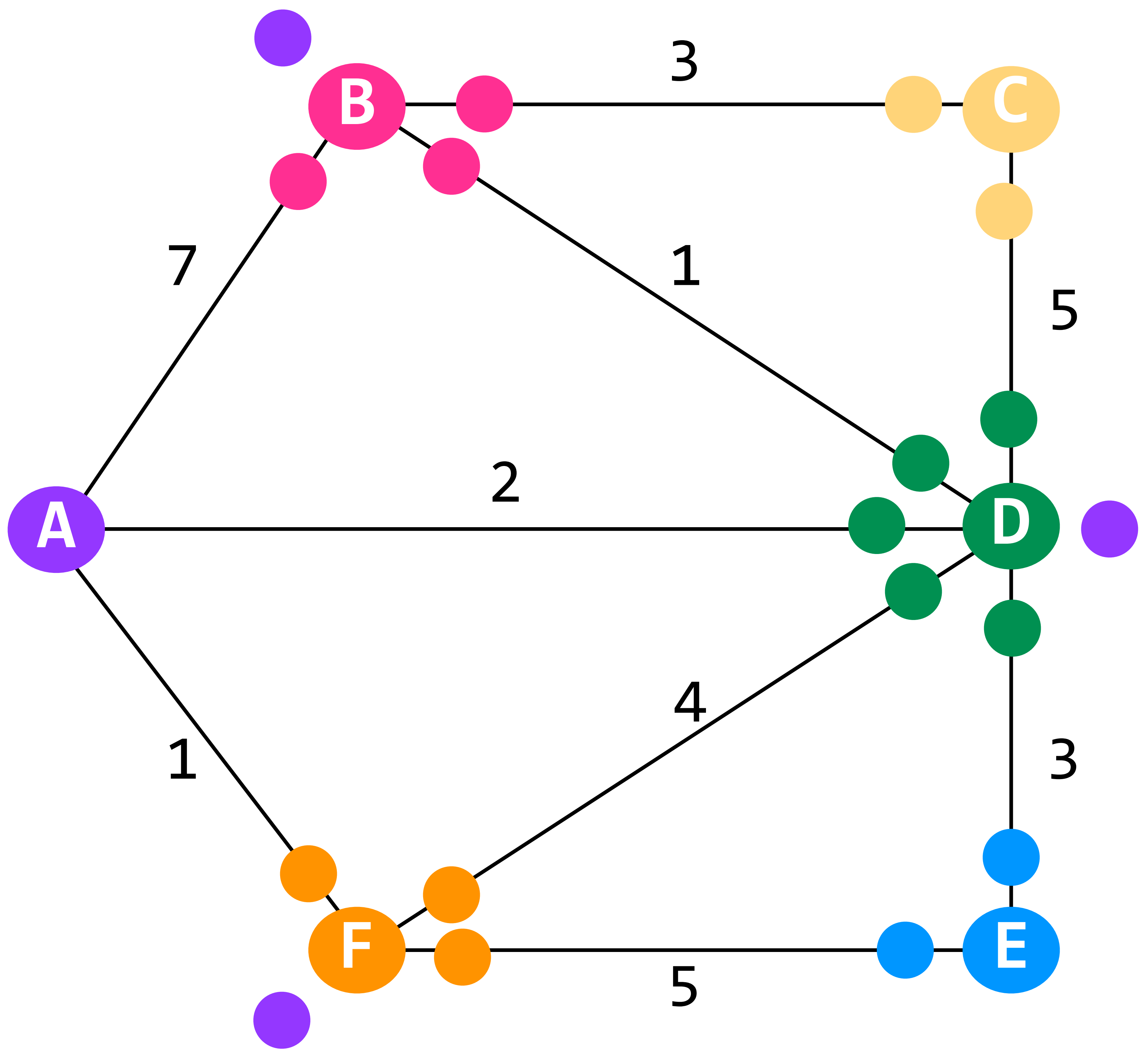
A's routing table

dst	route	cost
B	A-B	7
D	A-D	2
F	A-F	1

A's neighbors **do not** forward A's advertisements; they *do* send advertisements of their own to A

link state	distance vector
what's in a node's advertisement	
its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	
effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	
flooding makes link-state routing very resilient to failure	
what limits scale?	
the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology




```
A's first advertisement: [(B,7),(D,2),(F,1)]
```

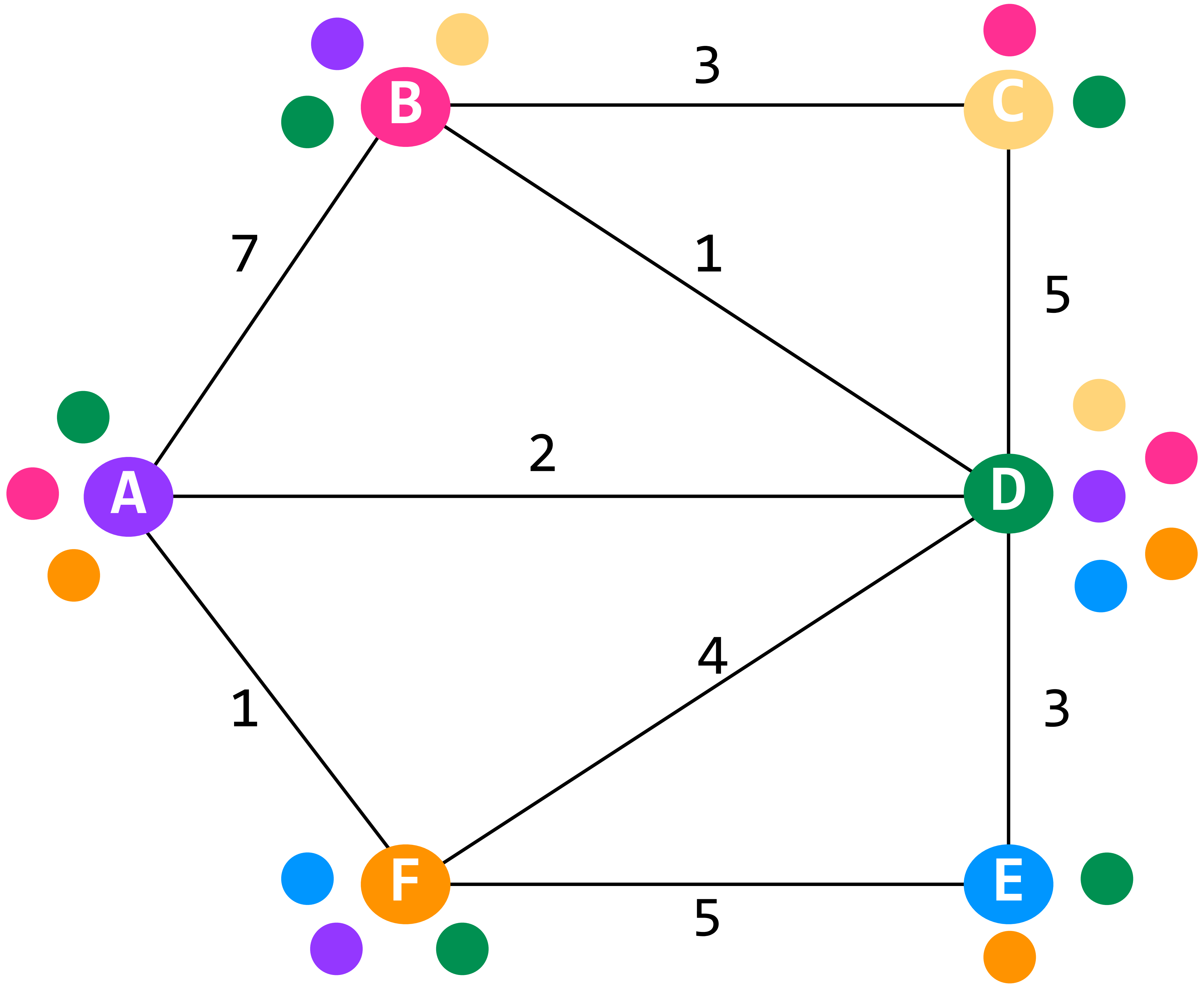
A's routing table

dst	route	cost
B	A-B	7
D	A-D	2
F	A-F	1

A's neighbors **do not** forward A's advertisements; they *do* send advertisements of their own to A

	link state	**distance vector**
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



`A's first advertisement: [(B,7),(D,2),(F,1)]`

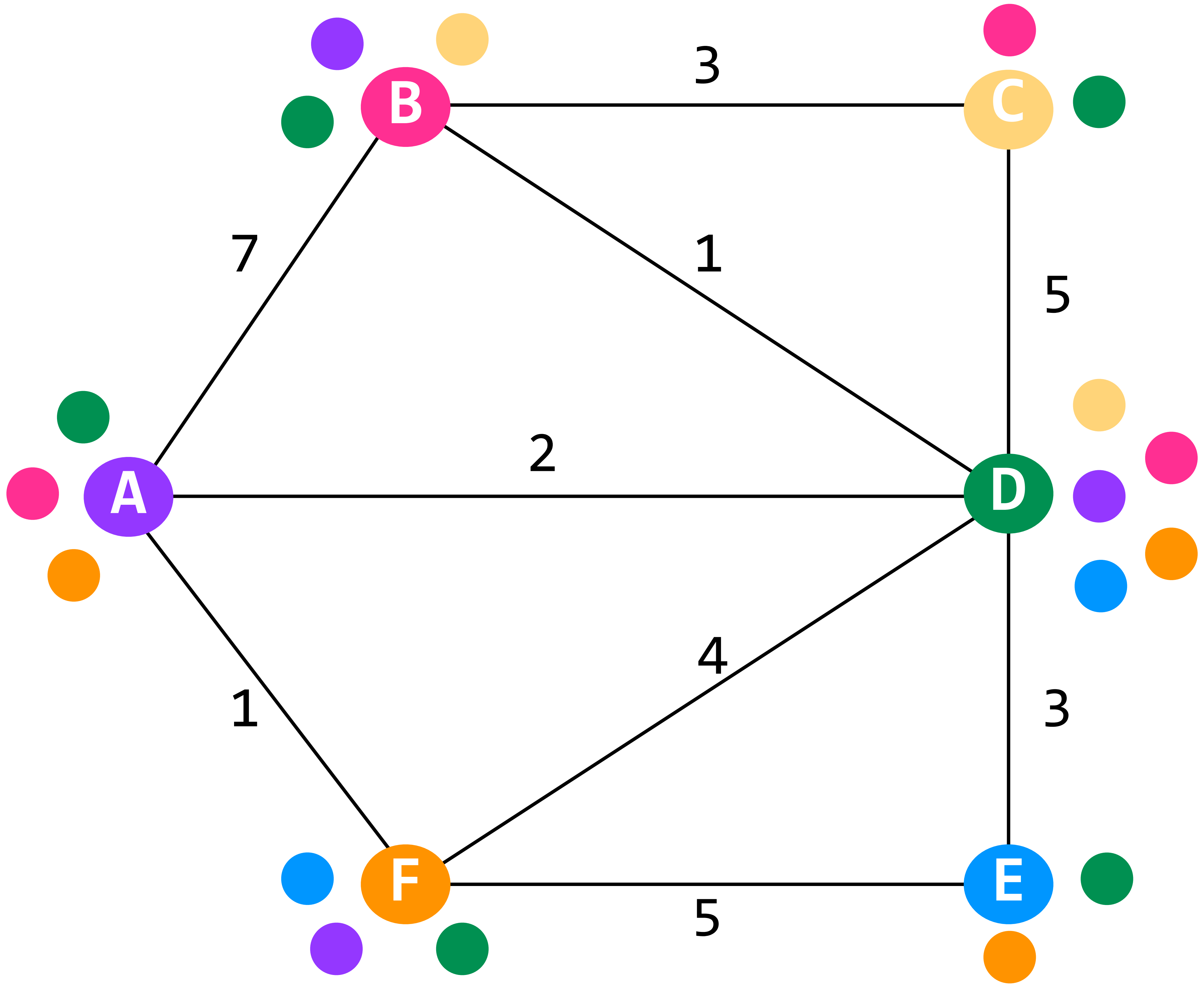
A's routing table

dst	route	cost
B	A-B	7
D	A-D	2
F	A-F	1

question: what are the contents of B's first advertisement?

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



A's routing table

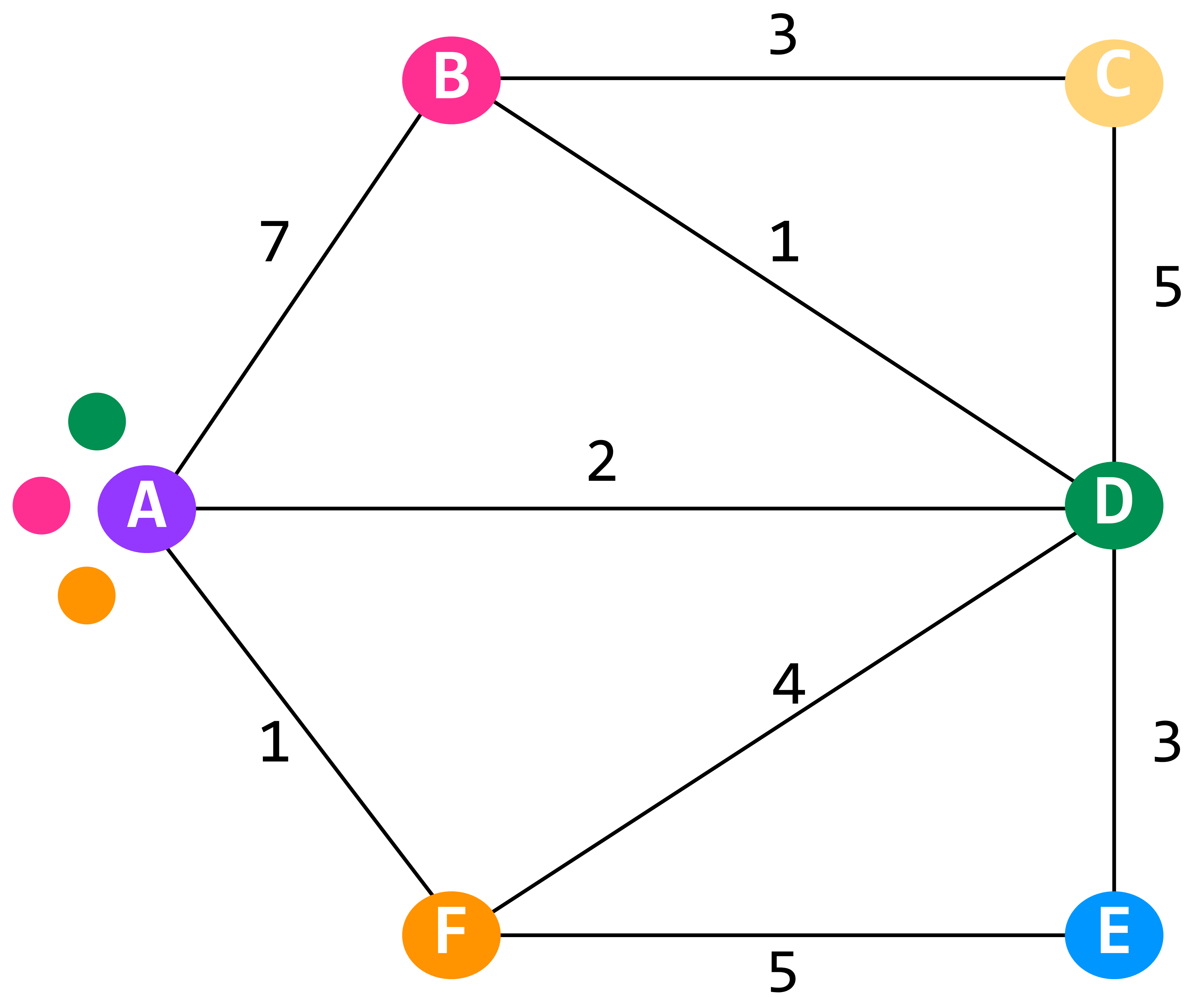
dst	route	cost
B	A-B	7
D	A-D	2
F	A-F	1

```
B's first adv: [(A,7), (C,3), (D,1)]
D's first adv: [(A,2), (B,1), (C,5), (E,3), (F,4)]
F's first adv: [(A,1), (D,4), (E,5)]
```

A receives advertisements from B, D, and F

	link state	**distance vector**
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



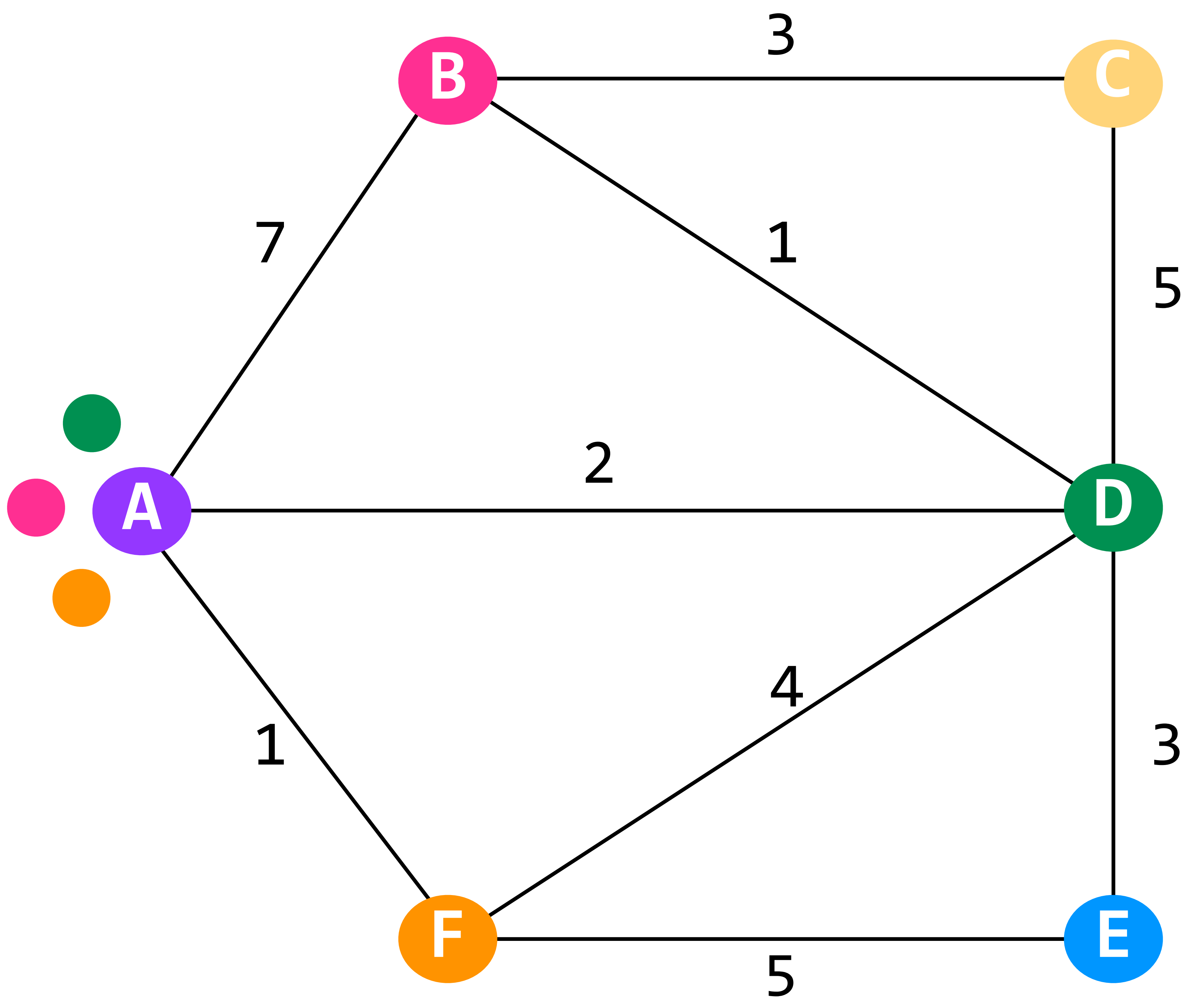

```
B's first adv: [(A,7), (C,3), (D,1)]
```

A's routing table

dst	route	cost
B	A-B	7
D	A-D	2
F	A-F	1

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



A's routing table

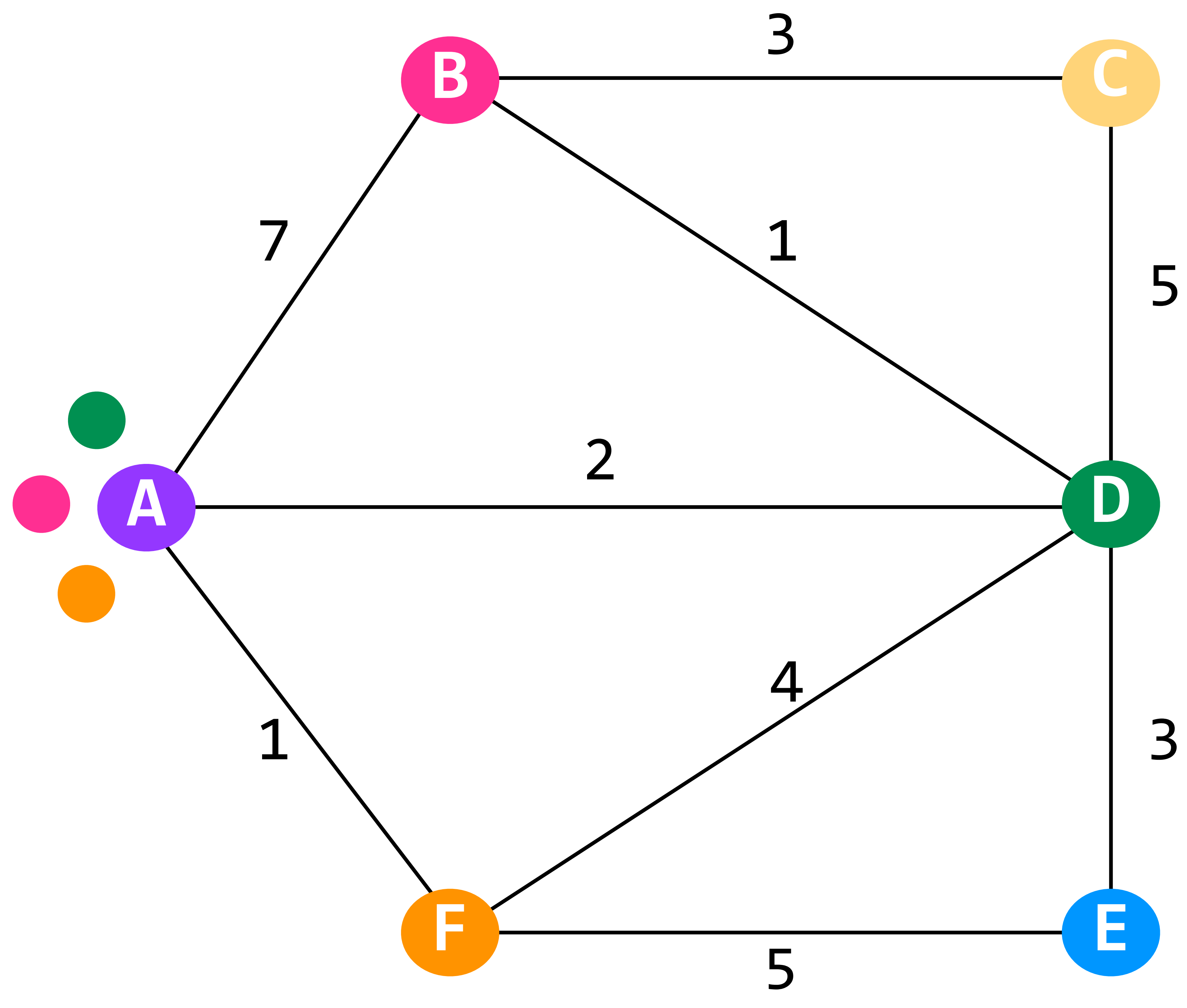
B's *first* adv: [(A,7), (C,3), (D,1)]

dst	route	cost
B	A-B	7
C	A-B	10
D	A-D	2
F	A-F	1

A's cost to B + B's cost to C

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



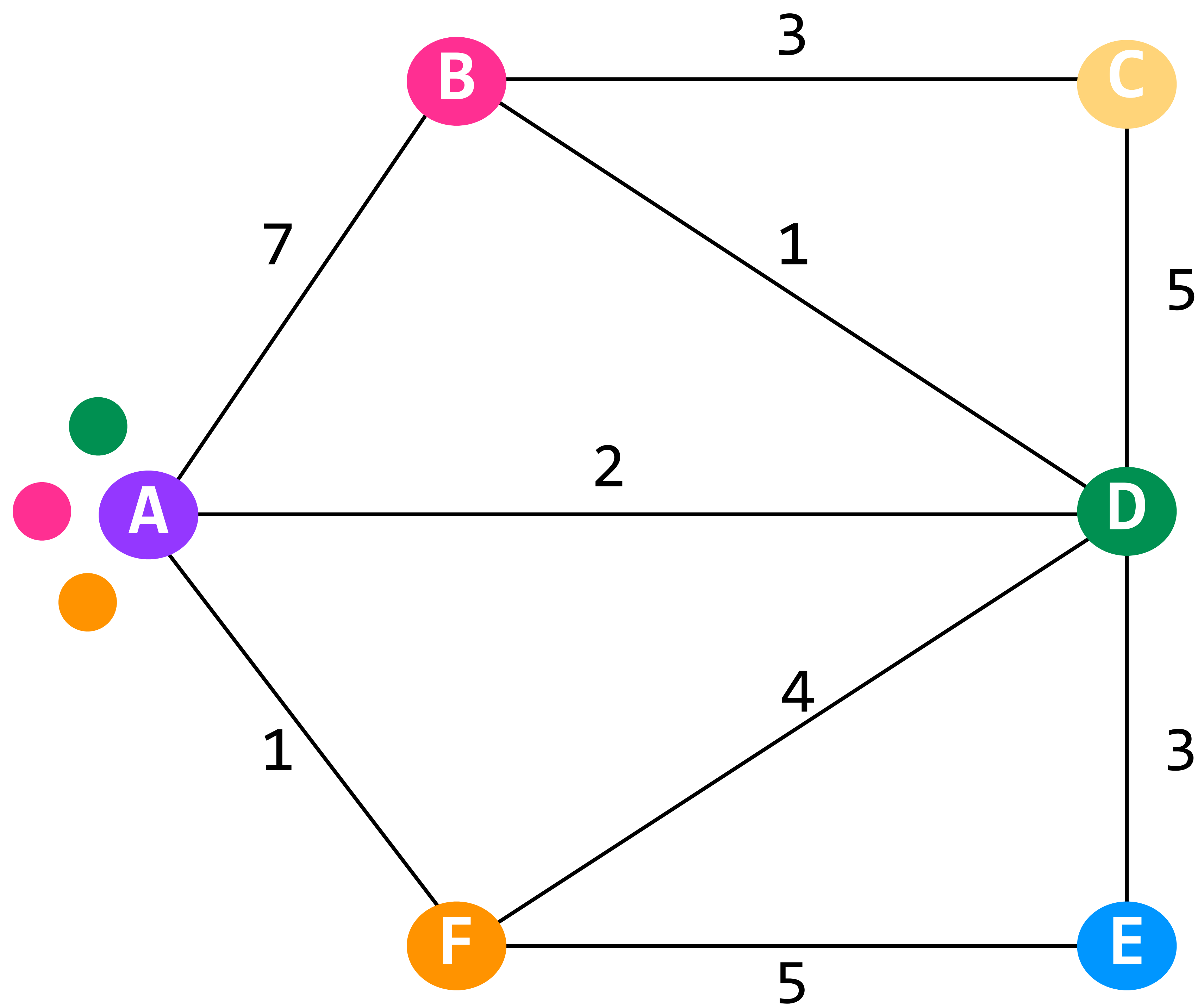
A's routing table

dst	route	cost
B	A-B	7
C	A-B	10
D	A-D	2
F	A-F	1

`B's first adv: [(A,7), (C,3), (D,1)]`

	link state	**distance vector**
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



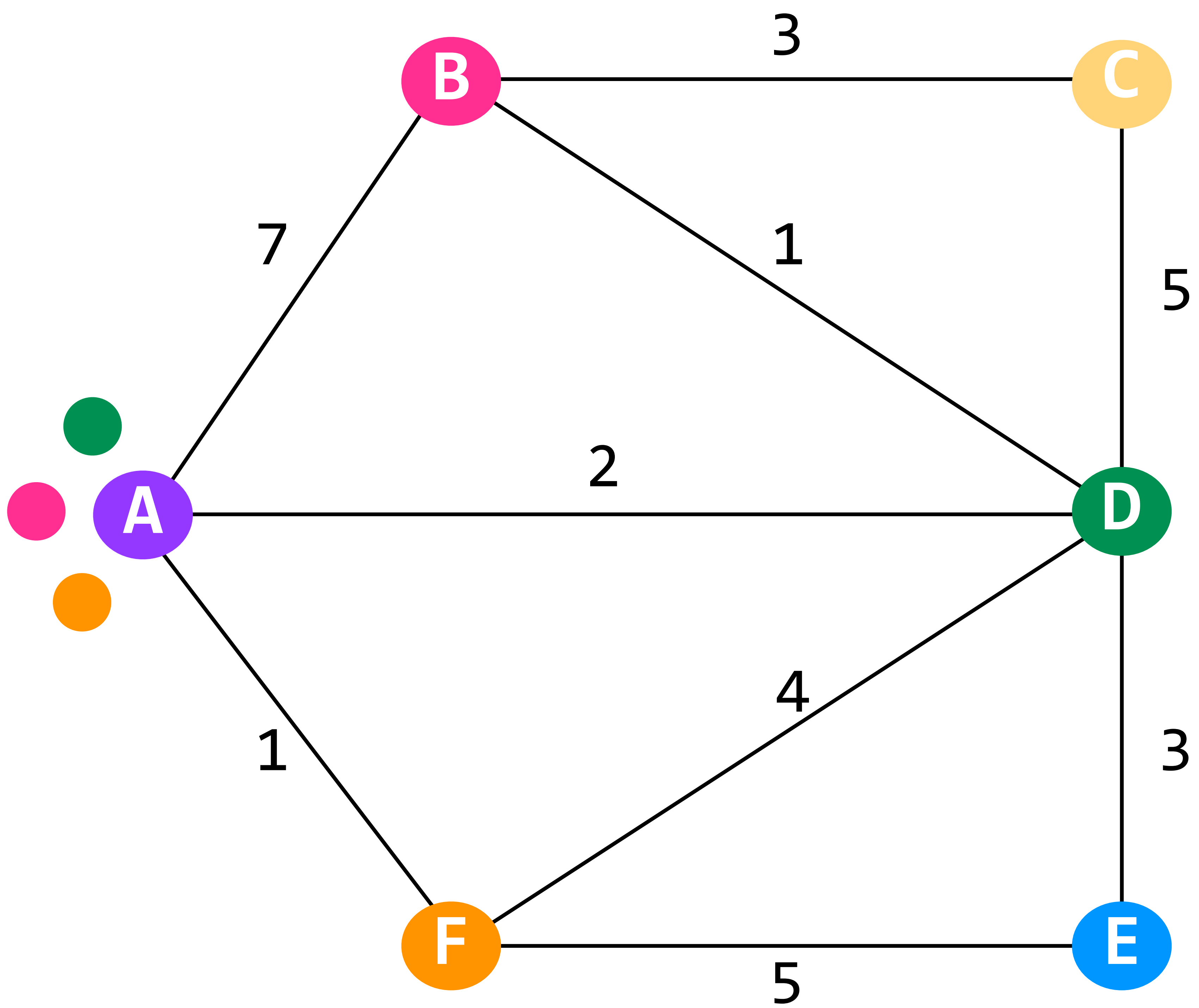
A's routing table

dst	route	cost
B	A-D	3
C	A-B	10
D	A-D	2
F	A-F	1

D's *first* adv: [(A,2), **(B,1)**, (C,5), (E,3), (F,4)]

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



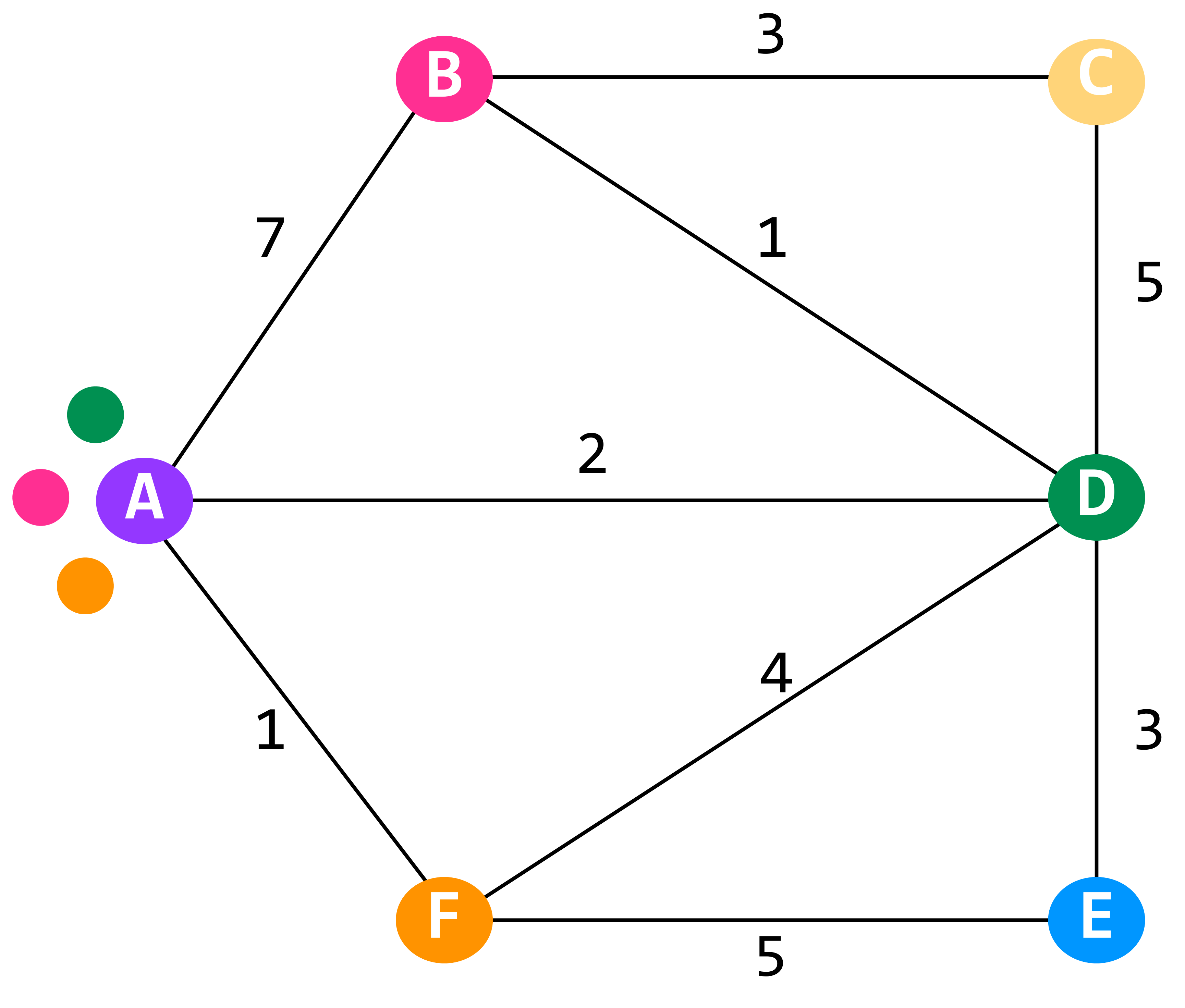
A's routing table

dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
F	A-F	1

D's *first* adv: [(A,2), (B,1), (C,5), (E,3), (F,4)]

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



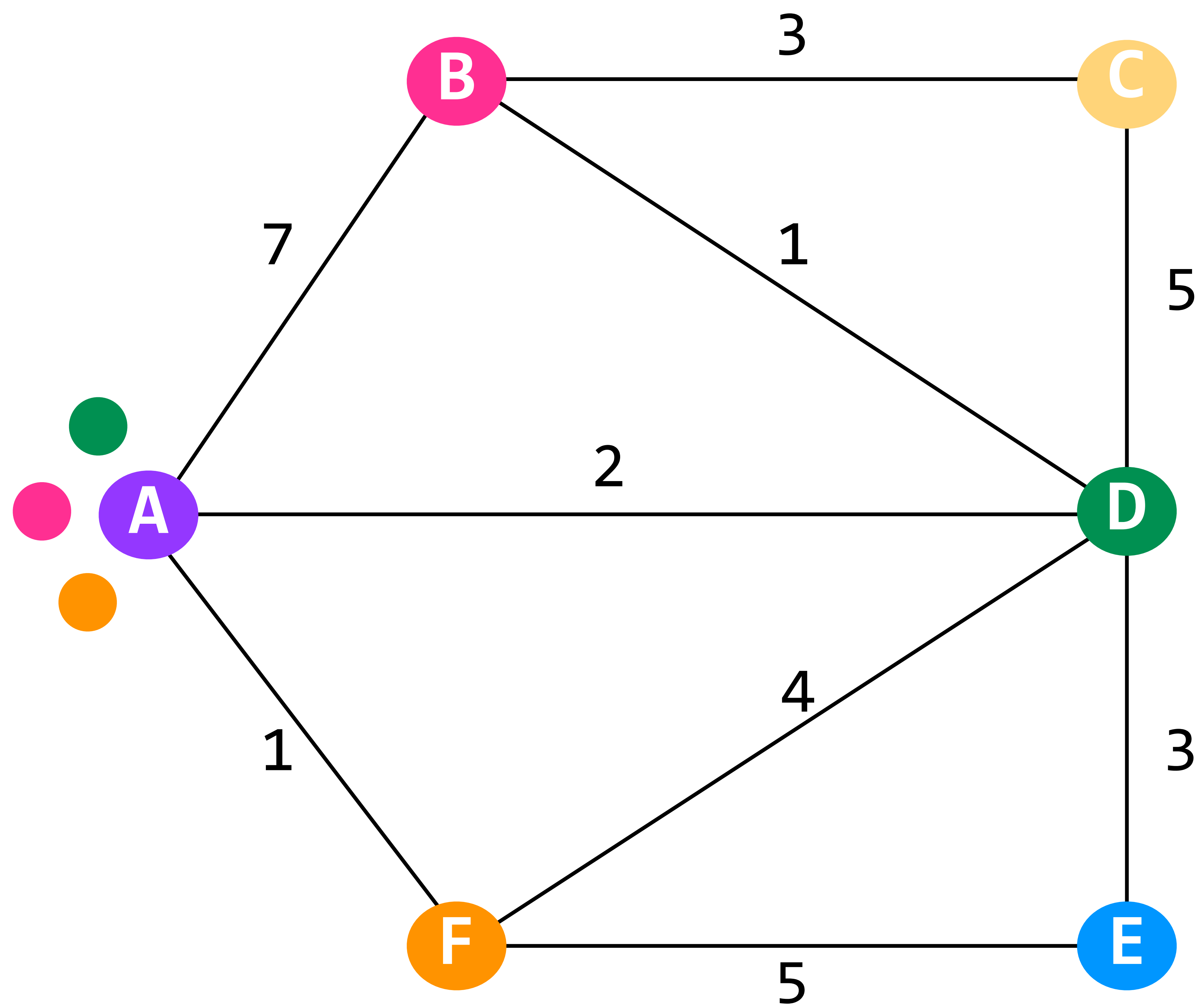
A's routing table

dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
E	A-D	5
F	A-F	1

`D's` *first* `adv: [(A,2), (B,1), (C,5), (E,3), (F,4)]`

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



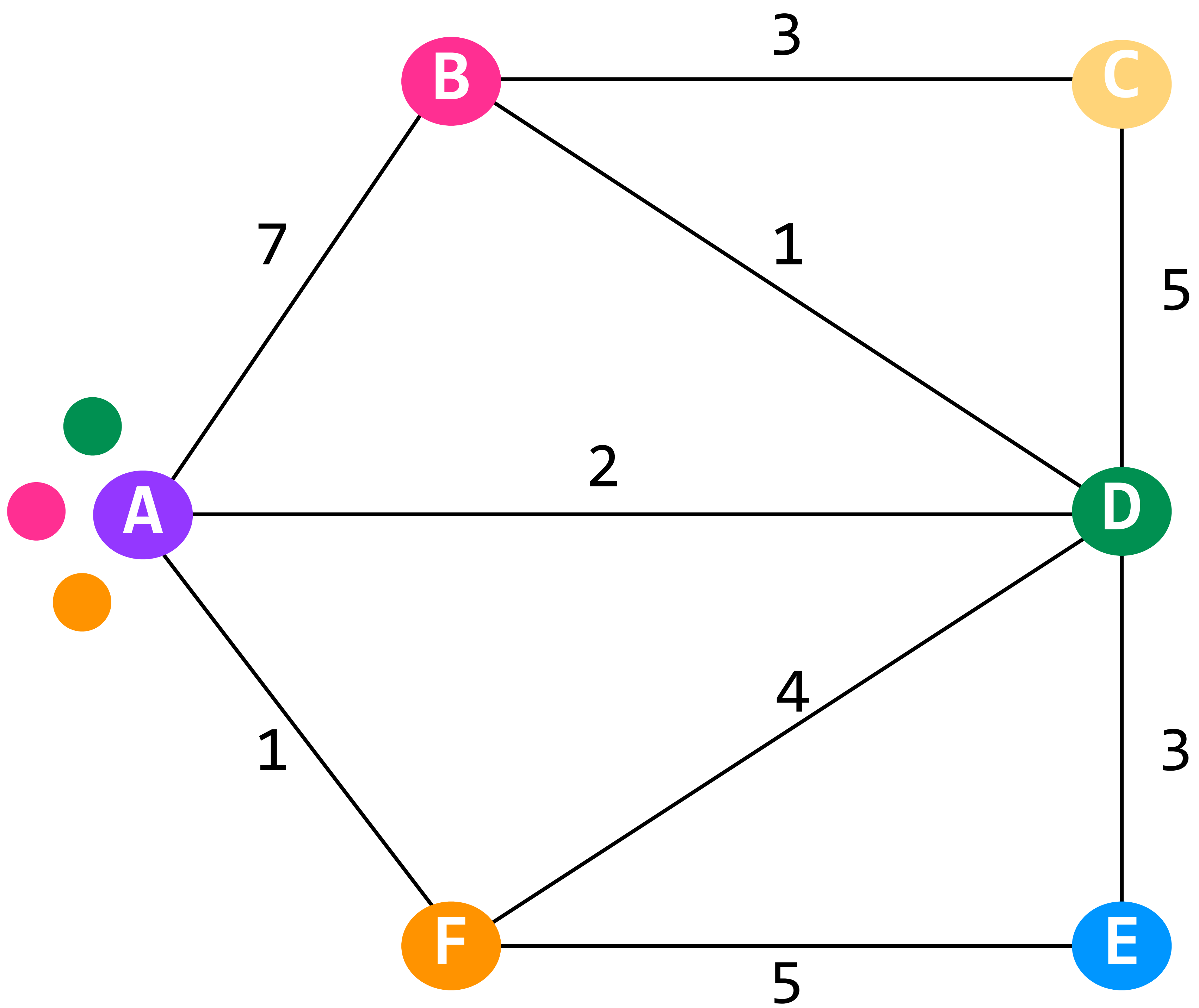
A's routing table

dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
E	A-D	5
F	A-F	1

`D's first adv: [(A,2), (B,1), (C,5), (E,3), (F,4)]`

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



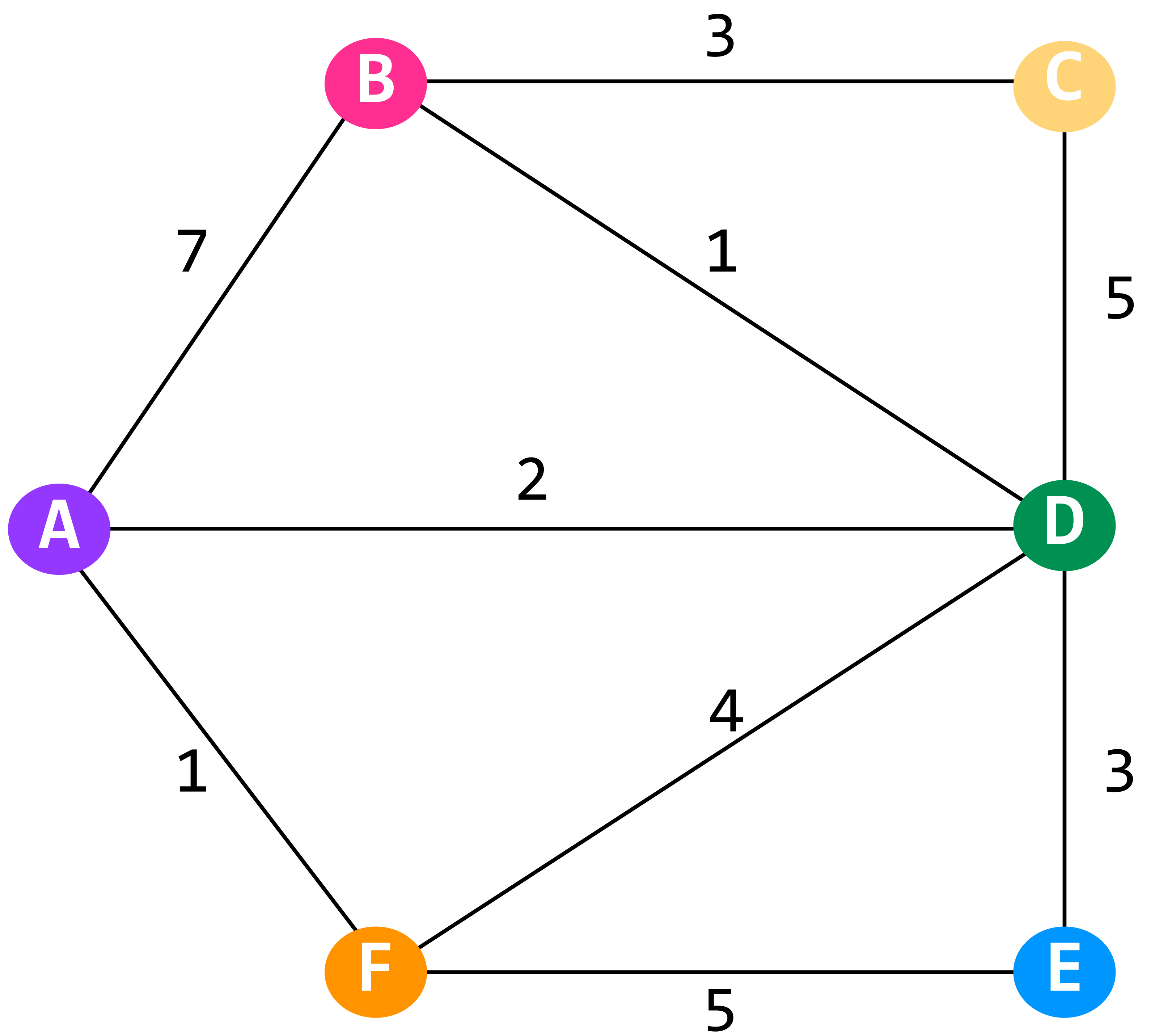
A's routing table

dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
E	A-D	5
F	A-F	1

`F's first adv: [(A,1), (D,4), (E,5)]`

	link state	**distance vector**
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



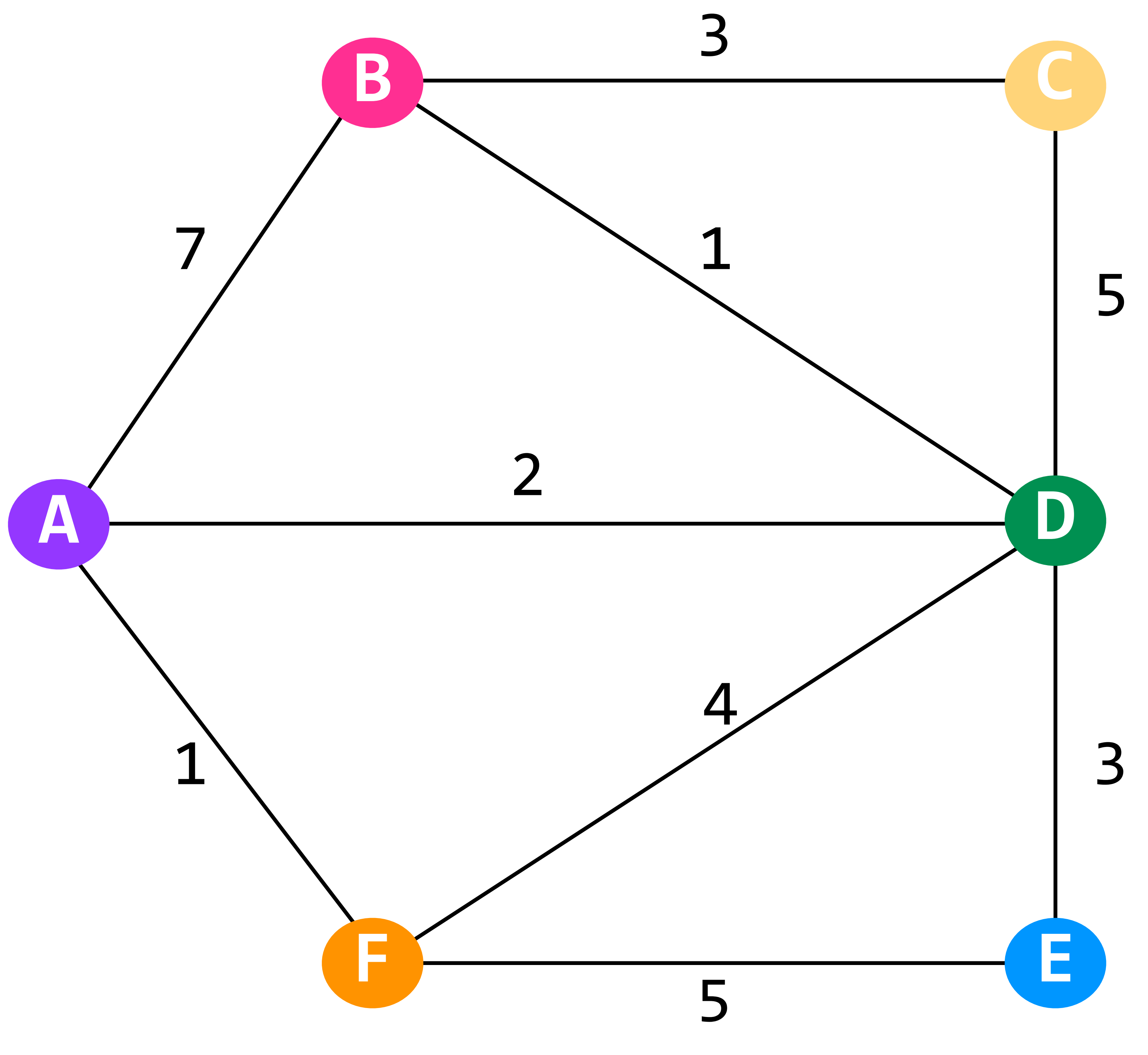
A's routing table

dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
E	A-D	5
F	A-F	1

this is A's routing table after one round of advertisements; note that it does not have knowledge of the min-cost path to C yet

	link state	**distance vector**
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



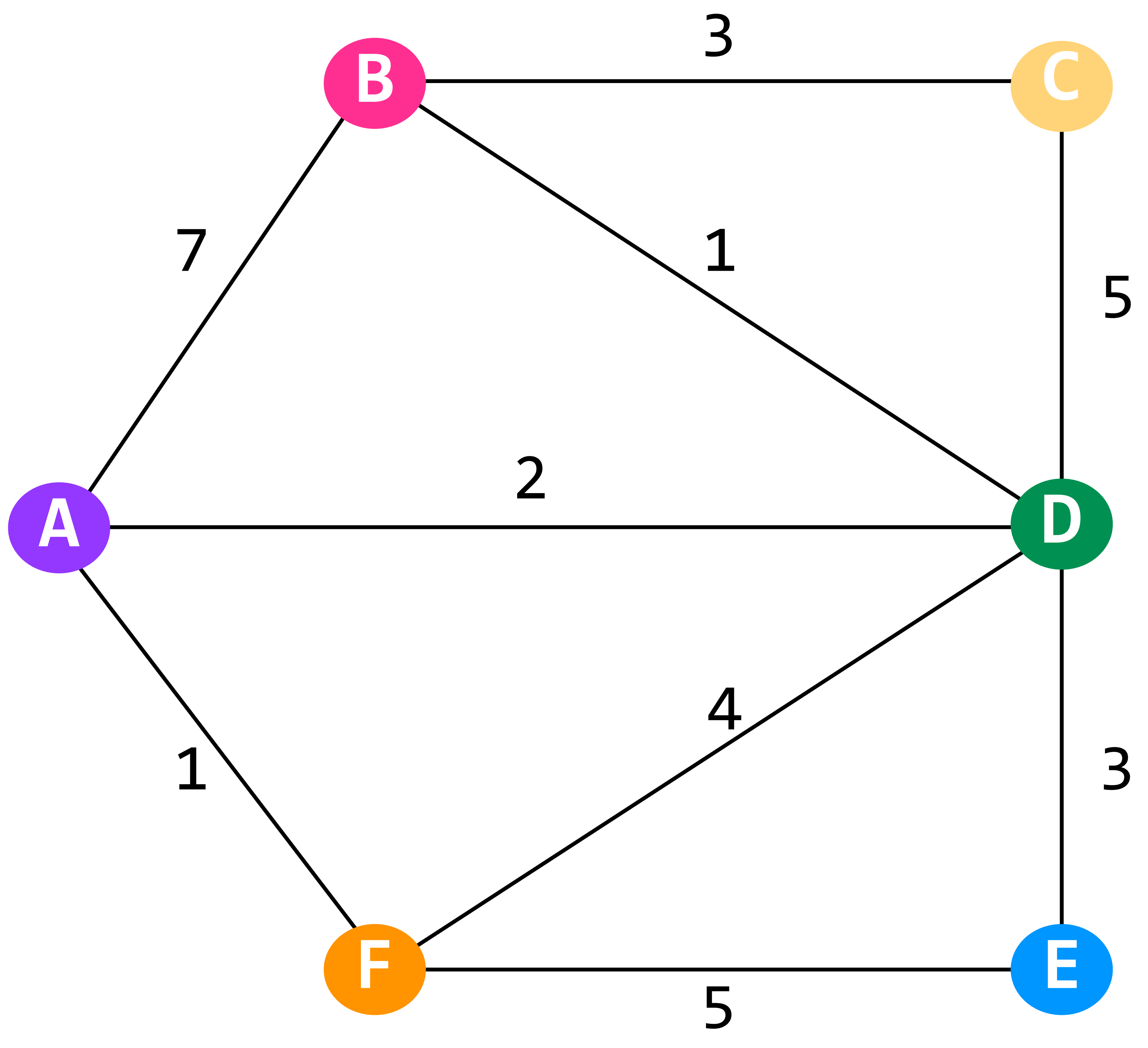
A's routing table

dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
E	A-D	5
F	A-F	1

question: what does A's *next* advertisement look like?

link state	distance vector
what's in a node's advertisement	
its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	
effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	
flooding makes link-state routing very resilient to failure	
what limits scale?	
the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



A's routing table

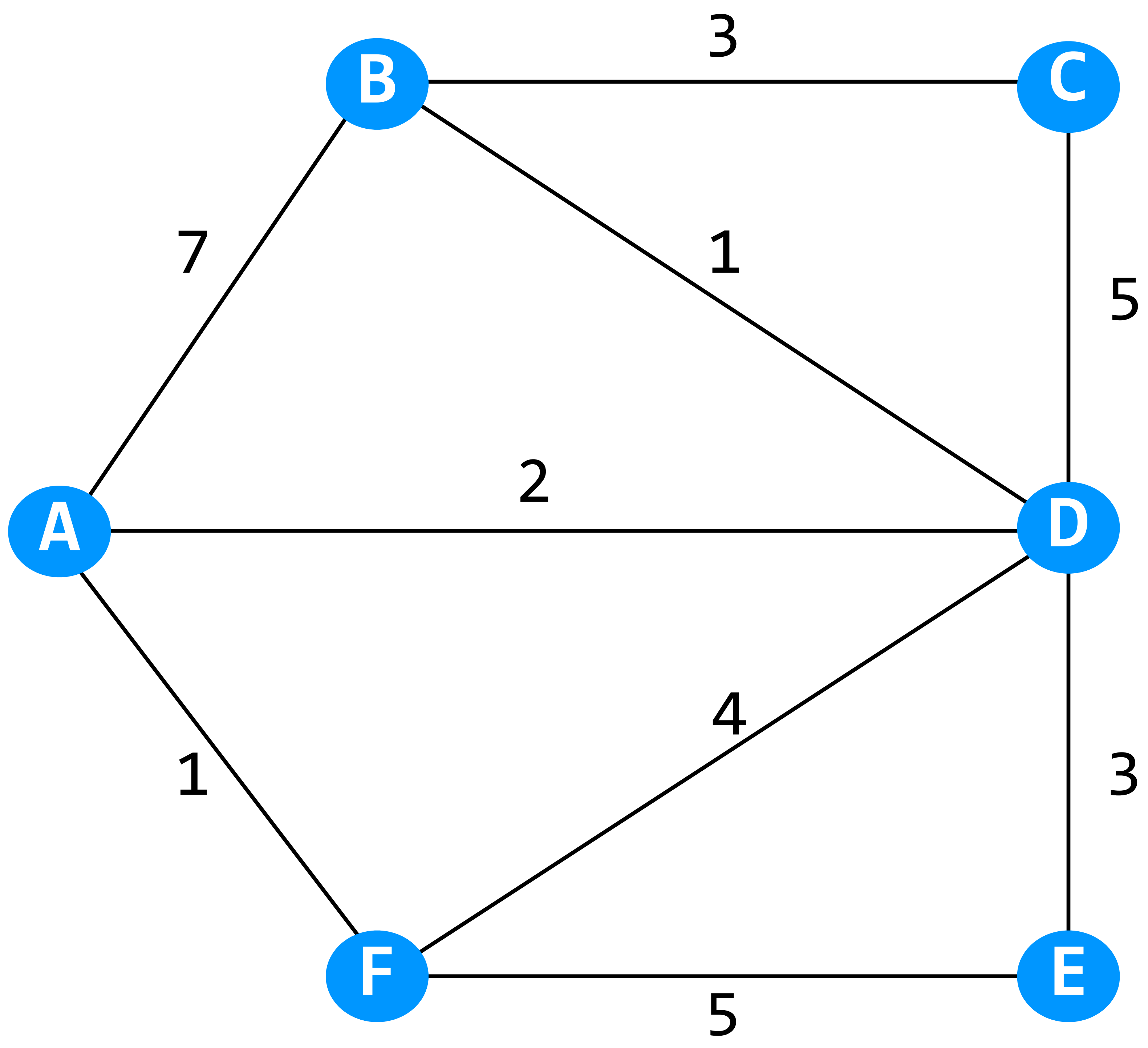
dst	route	cost
B	A-D	3
C	A-D	7
D	A-D	2
E	A-D	5
F	A-F	1

A's *second* adv:
[(B,3), (C,7), (D,2), (E,5), (F,1)]

A will learn about the correct min-cost path to C in the next round of advertisements; try that out for yourself!

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	
what limits scale?	the **overhead** of flooding	

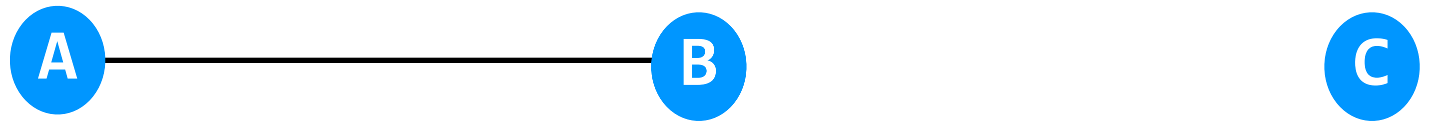
distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	failures can be complicated because of timing
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology

A sends advertisements at t=0, 10, 20,..; B sends advertisements at t=5, 15, 25,..
every link has cost 1




```
A: Self, 0    A: B->A, 1
B: A->B, 1    B: Self, 0
C: A->B, 2    C: None, inf
```

`t=9: B<->C` fails

B discovers this failure thanks to the HELLO protocol and updates its routing table quickly

```
A: Self, 0    A: B->A, 1
B: A->B, 1    B: Self, 0
C: A->B, 2    C: B->A, 3   (2+1)
```

`t=10:` B receives the following advertisement from A:
`[(A,0),(B,1),(C,2)]`

```
A: Self, 0    A: B->A, 1
B: A->B, 1    B: Self, 0
C: A->B, 4    C: B->A, 3
```

`t=15:` A receives the following advertisement from B:
`[(A,1),(B,0),(C,3)]`

A updates its routing table because it's using B to get to C, and B's cost to C has changed

```
A: Self, 0    A: B->A, 1
B: A->B, 1    B: Self, 0
C: A->B, 4    C: B->A, 5
```

`t=20:` B receives the following advertisement from A:
`[(A,0),(B,1),(C,4)]`

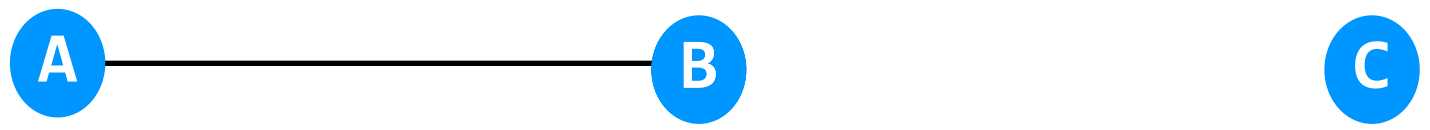
continues until both costs to C are INFINITY

in this example, nodes will explicitly include their route/cost to themselves in their advertisements; you can make distance-vector work either way

	link state	distance vector
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	failures can be complicated because of timing
what limits scale?	the **overhead** of flooding	

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology

A sends advertisements at t=0, 10, 20,..; B sends advertisements at t=5, 15, 25,..
every link has cost 1




```
A: Self, 0      A: B->A, 1
B: A->B, 1      B: Self, 0
C: A->B, 2      C: None, inf
```

t=9: B<->C fails

```
A: Self, 0      A: B->A, 1
B: A->B, 1      B: Self, 0
C: A->B, 2      C: None, inf
```

t=10: B receives the following advertisement from A:
`[(A,0)]`

```
A: Self, 0      A: B->A, 1
B: A->B, 1      B: Self, 0
C: None, inf    C: None, inf
```

t=15: A receives the following advertisement from B:
`[(B,0),(C,inf)]`

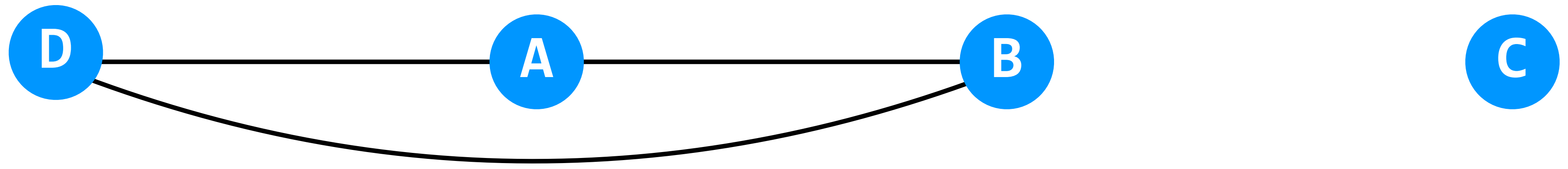
new strategy ("split horizon"): don't send advertisements about a route to the node providing the route

split horizon takes care of this particular case

	link state	**distance vector**
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	failures can be complicated because of timing
what limits scale?	the **overhead** of flooding	

in this example, nodes will explicitly include their route/cost to themselves in their advertisements; you can make distance-vector work either way

distance-vector routing: disseminate information about the current *min costs* to each node, rather than the complete topology



D	A	B	C
`C: D->B, 2`	`C: A->B, 2`	**`C: None, inf`**	B<->C fails
`C: None, inf`	`C: A->B, 2`	`C: None, inf`	B's advertisement to A gets lost (so A makes no changes)
`C: D->A, 3`	`C: A->B, 2`	`C: None, inf`	A advertises about C to D (not to B because of split horizon)
`C: D->A, 3`	`C: A->B, 2`	**`C: B->D, 4`**	D advertises about C to B
`C: D->A, 3`	**`C: A->B, 5`**	`C: B->D, 4`	B advertises about C to A

continues until all costs to C are INFINITY

new strategy ("split horizon"): don't send advertisements about a route to the node providing the route

	link state	**distance vector**
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	failures can be complicated because of timing
what limits scale?	the **overhead** of flooding	failure handling

in this example, nodes will explicitly include their route/cost to themselves in their advertisements; you can make distance-vector work either way

Katrina LaCurts | lacurts@mit.edu | 6.1800 2025

neither one of these algorithms alone will scale to the size of the internet, nor do either of them allow for *policy routing*

	link state	**distance vector**
what's in a node's advertisement	its **link costs** to each of its **neighbors**	its **current costs** to **every node it's aware of**
who gets a node's advertisement	effectively, **every other node** (via flooding)	only its **neighbors**
what happens when things fail?	flooding makes link-state routing very resilient to failure	failures can be complicated because of timing
what limits scale?	the **overhead** of flooding	failure handling

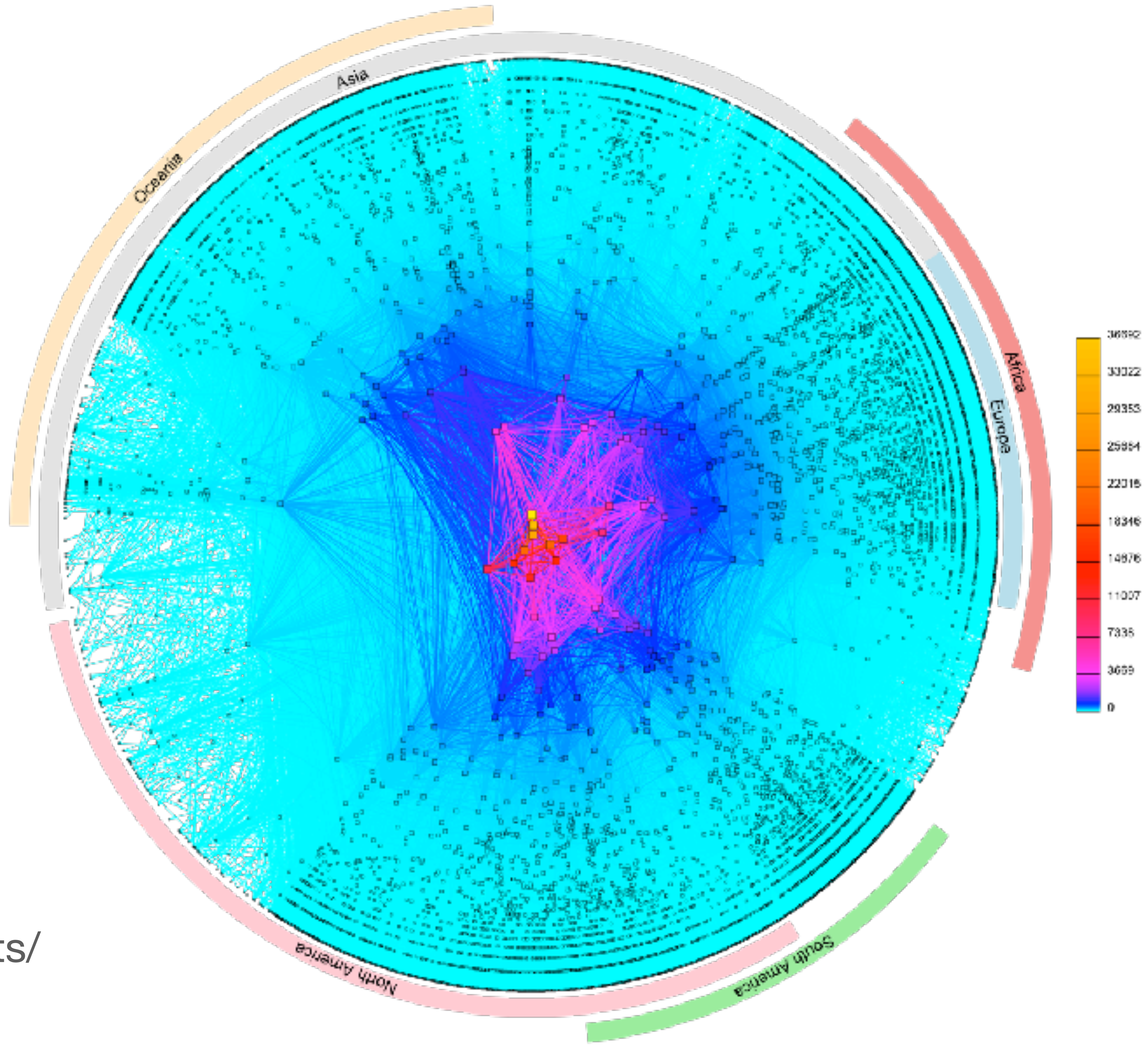
1970s: ARPAnet — hosts.txt

1978: flexibility and layering — **distance-vector routing**

early 80s: growth → change — TCP, UDP; **OSPF**, EGP, DNS (a link-state routing protocol)

late 80s: growth → problems — congestion collapse; policy routing; CIDR

1993: commercialization



CAIDA's IPv4 AS Core, January 2020 (https://www.caida.org/projects/cartography/as-core/2020/)

`application`	`the things that actually generate traffic`
`transport`	`sharing the network, reliability (or not)` *examples: TCP, UDP*
`network`	**`naming, addressing, routing`** *examples: IP*
`link`	`communication between two directly-connected nodes` *examples: ethernet, bluetooth, 802.11 (wifi)*

IP networks can route using either distance-vector routing (RIP) or link-state routing (OSPF)

Katrina LaCurts | lacurts@mit.edu | 6.1800 2025