

Lecture 7:

The Network Layer

Katerina Argyraki, EPFL

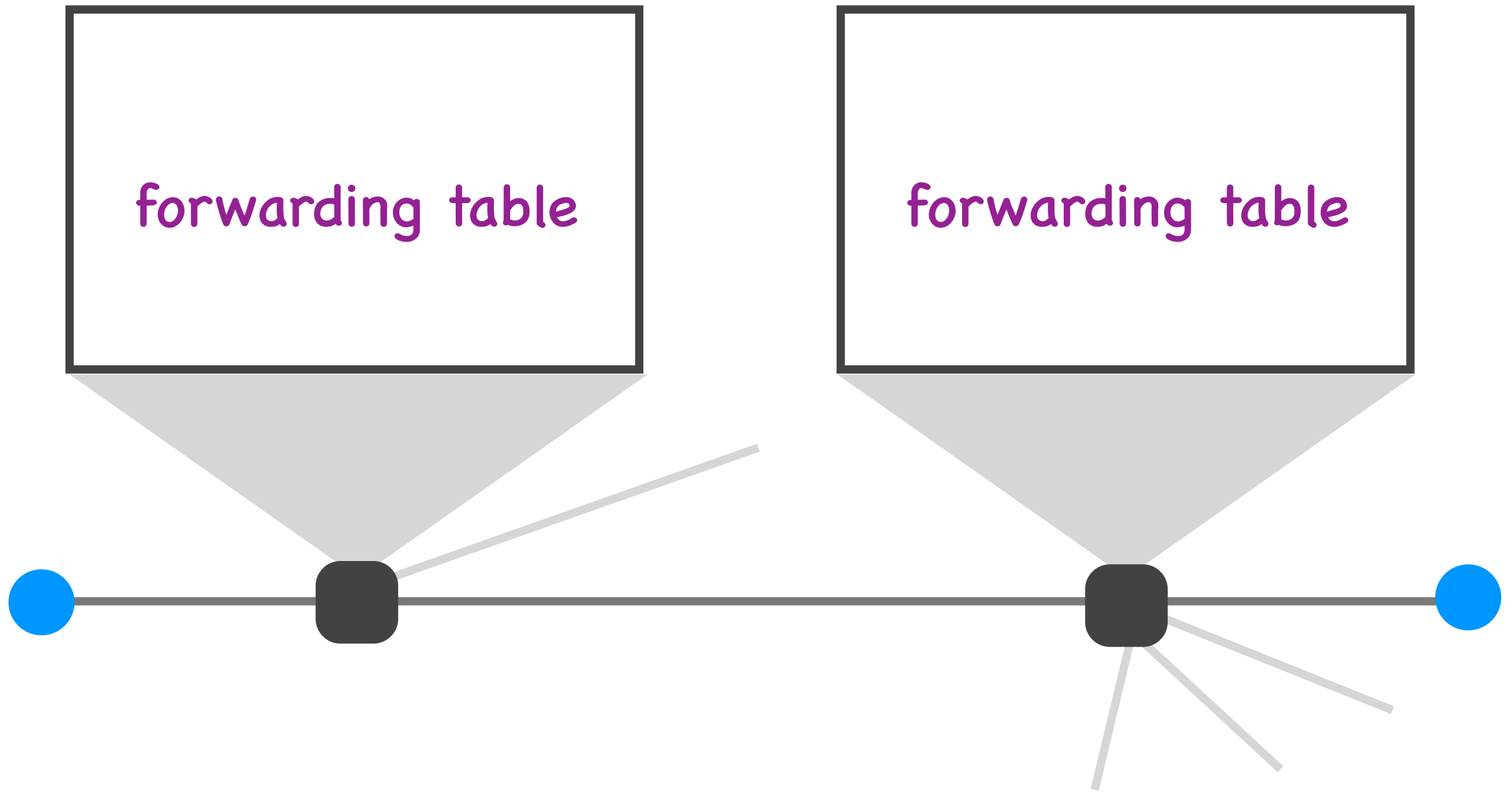
Outline

- Network-layer **functions**
 - ▶ forwarding
 - ▶ routing
- Network-layer **types**
 - ▶ virtual-circuit networks
 - ▶ datagram networks
- **IP forwarding**
- **IP routing**

Outline

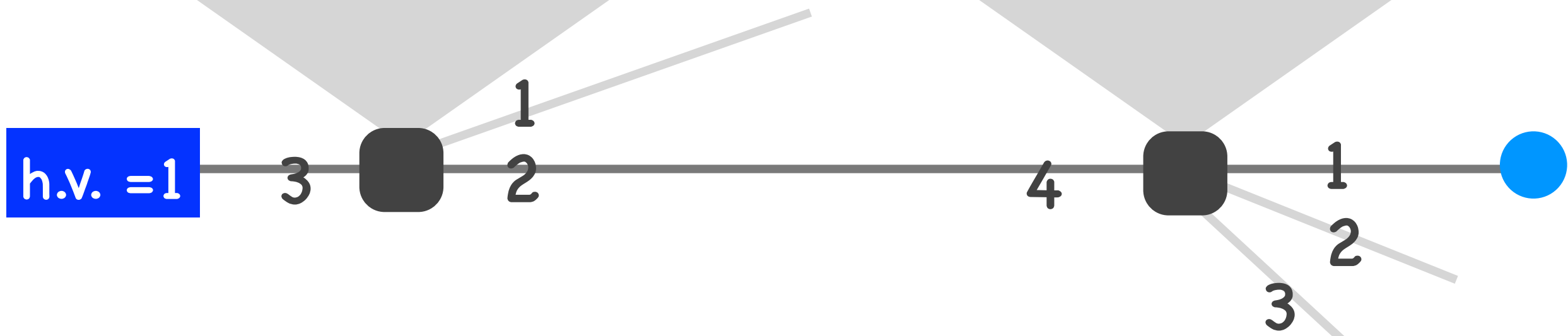
- **Network-layer functions**
 - ▶ forwarding
 - ▶ routing
- Network-layer types
 - ▶ virtual-circuit networks
 - ▶ datagram networks
- IP forwarding
- IP routing





header value	output link
0	1
1	2
2	2
3	1

header value	output link
0	3
1	1
2	2
3	4



Forwarding

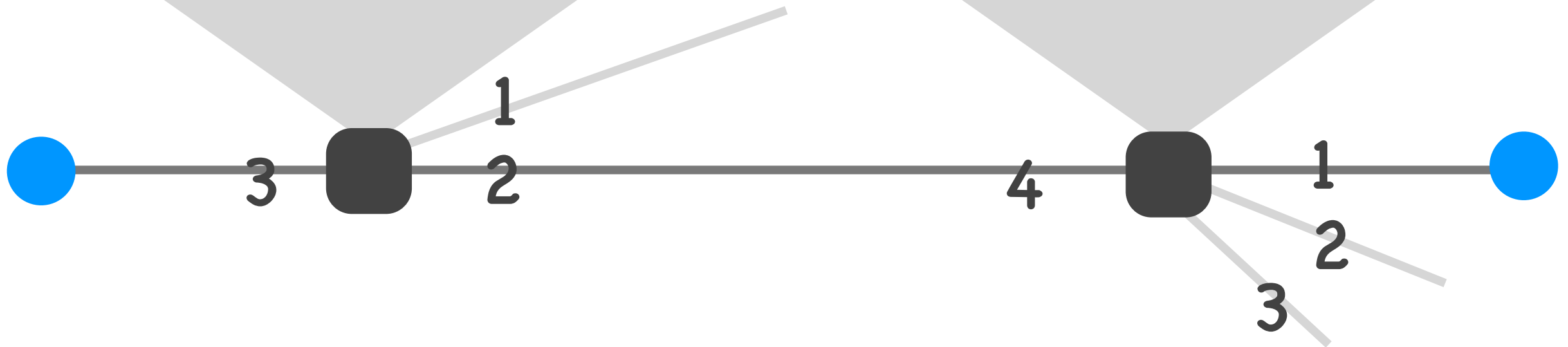
- **Local** process that takes place at a router and determines output link for each packet
- How: **read** value from packet's network-layer header, **search** forwarding table for output link

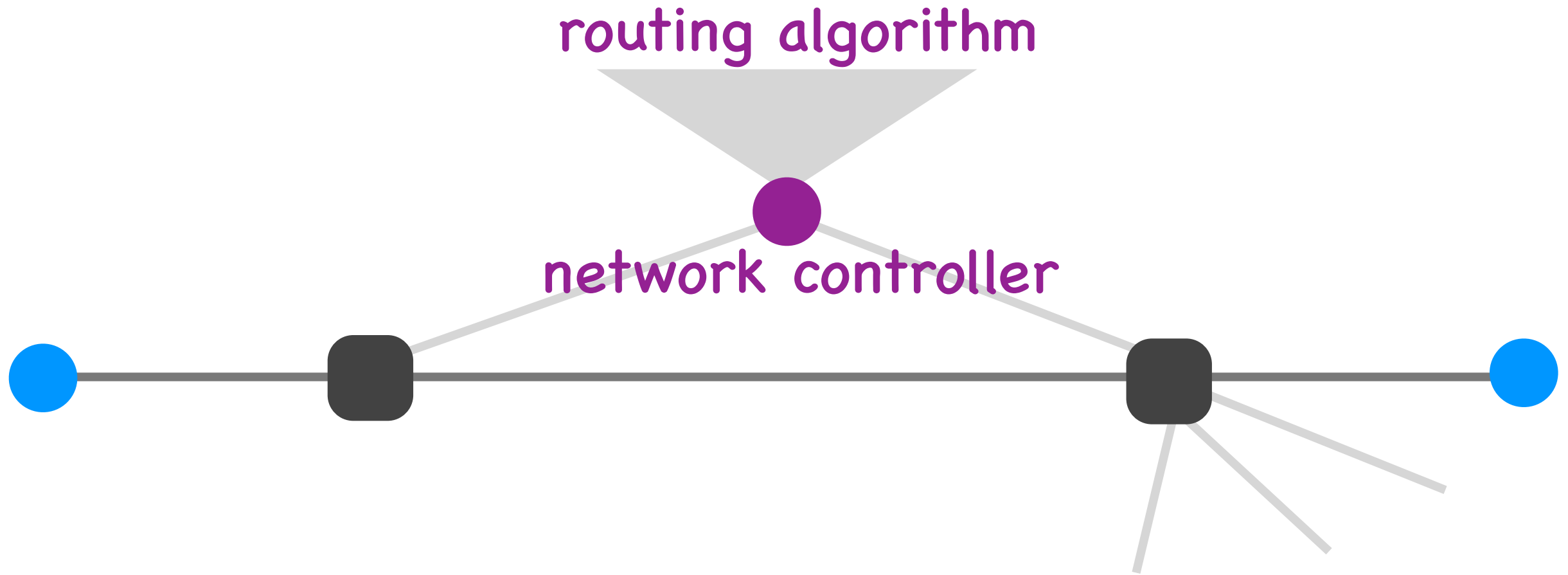
manually

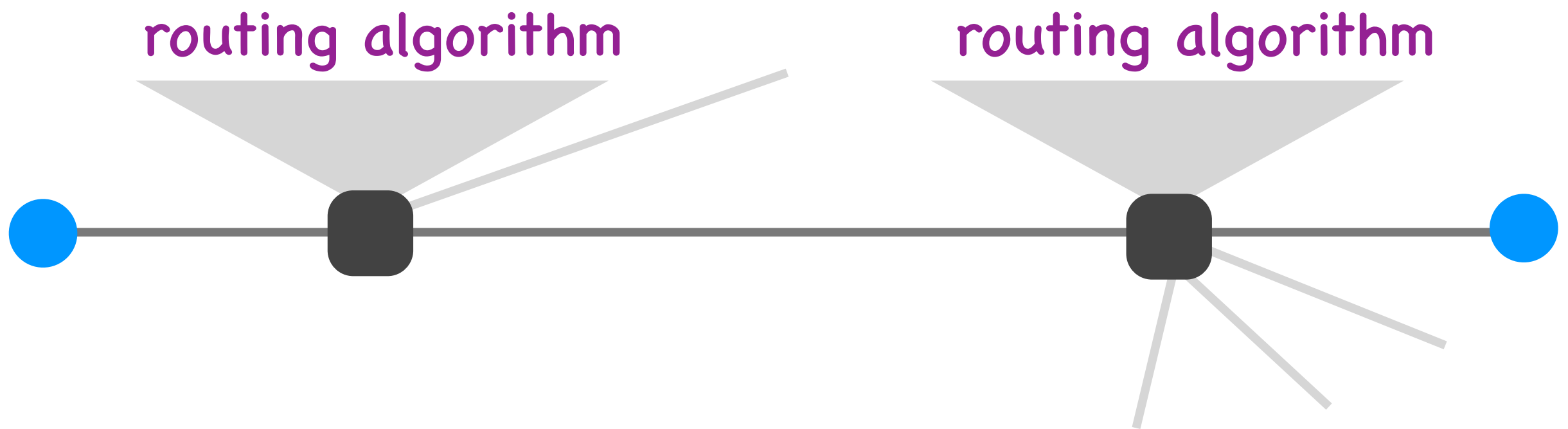
header value	output link
0	1
1	2
2	2
3	3

typically: routing

header value	output link
0	3
1	1
2	2
3	4







Routing

- **Network-wide** process that populates forwarding tables
- How: routing algorithm run on a logically centralised **network controller** or the **routers themselves**

Forwarding vs. routing

- Forwarding **determines output link** for each packet
- Routing **populates forwarding tables**

Outline

- Network-layer functions
 - forwarding
 - routing
- Network-layer **types**
 - virtual-circuit networks
 - datagram networks
- IP forwarding
- IP routing

Outline

- Network-layer functions
 - forwarding
 - routing
- Network-layer **types**
 - virtual-circuit networks
 - datagram networks
- IP forwarding
- IP routing

Possible guarantees

- Guaranteed (in-order) delivery
- Guaranteed maximum delay
- Guaranteed minimum throughput
- Security (confidentiality, authenticity)

Possible guarantees

- Guaranteed (in-order) delivery
- Guaranteed maximum delay
- **Guaranteed minimum throughput**
- Security (confidentiality, authenticity)

in link	VC	out link	VC
3	1	2	

in link	VC	out link	VC
4	5	1	

setup

Alice - Bob
100 Mbps
VC #1

Alice - Bob
100 Mbps
VC #5

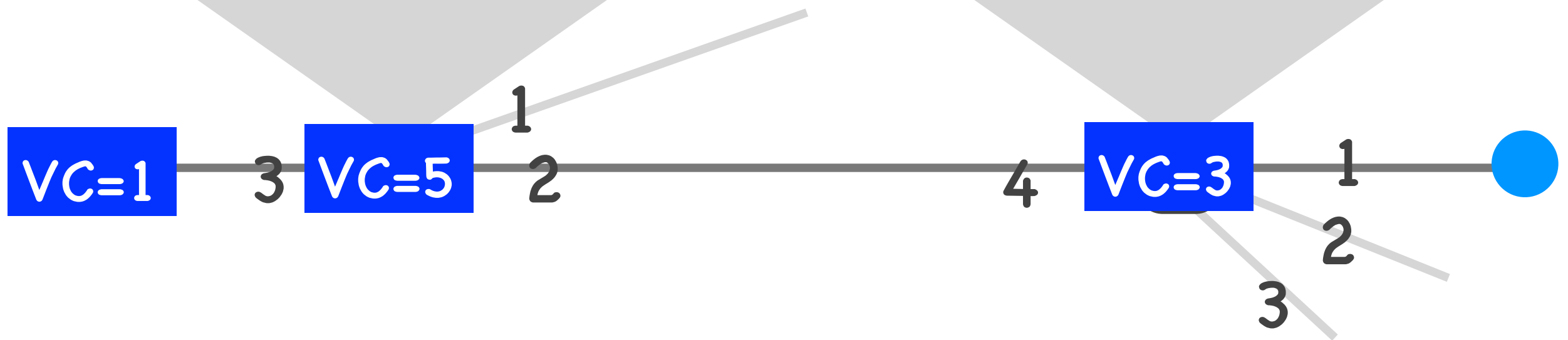
in link	VC	out link	VC
3	1	2	5

in link	VC	out link	VC
4	5	1	3



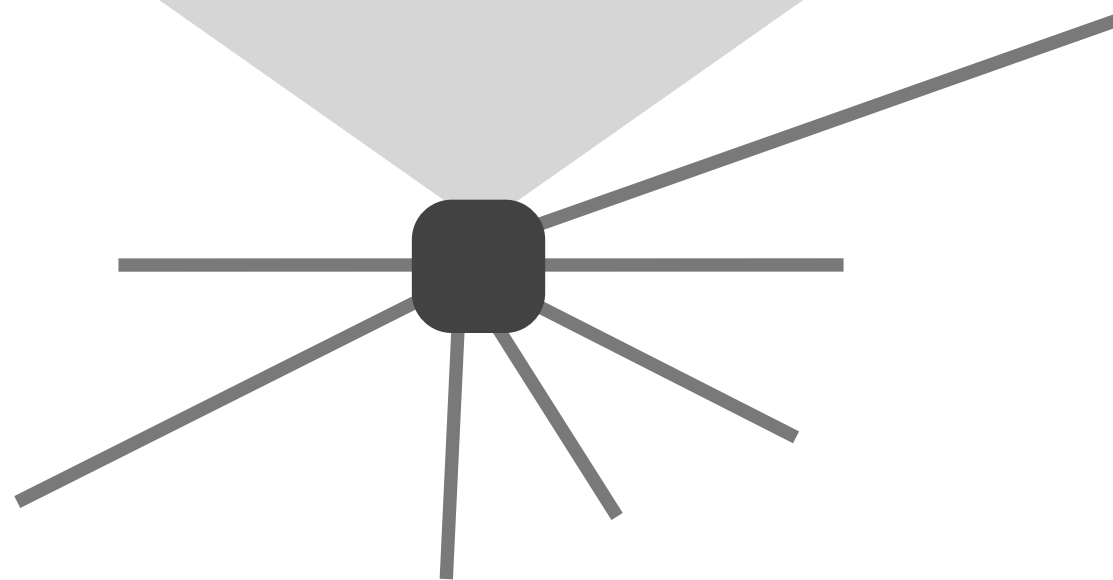
in link	VC	out link	VC
3	1	2	5

in link	VC	out link	VC
4	5	1	3



millions of
concurrent
connections

in link	VC	out link	VC
3	1	2	5
1	3	2	2
2	10	5	6
3	3	2	7
5	4	7	3
6	1	7	4



Virtual-circuit network

- Uses connection switching = network-layer **connections**
- Appropriate for **performance guarantees**
- Forwarding **state: per connection**
 - ▶ input link, VC #, output link, VC #
 - ▶ populated by connection setup (with help from routing)

Connections & state

- Many different kinds of connections
 - ▶ transport-layer (TCP) connections
 - ▶ network-layer connections
- Connection => per-connection state
 - ▶ at transport layer = at end-systems (OK)
 - ▶ at network layer = at all the routers (not OK)
- State => complexity

Outline

- Network-layer functions
 - forwarding
 - routing
- Network-layer **types**
 - virtual-circuit networks
 - datagram networks
- IP forwarding
- IP routing

Outline

- Network-layer functions
 - forwarding
 - routing
- Network-layer **types**
 - virtual-circuit networks
 - datagram networks
- IP forwarding
- IP routing

Outline

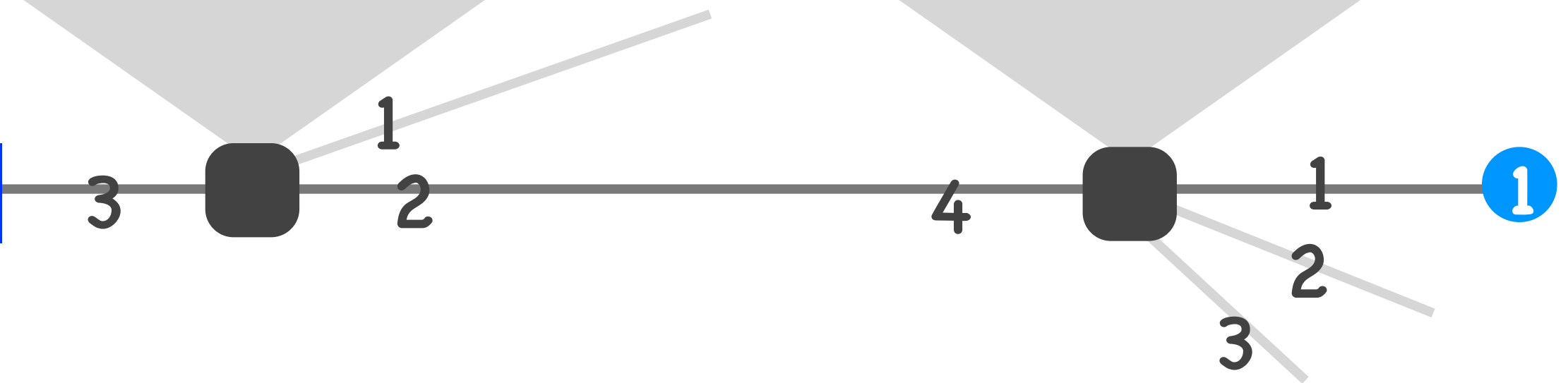
- Network-layer functions
 - forwarding
 - routing
- Network-layer types
 - virtual-circuit networks
 - datagram networks
- **IP forwarding**
- IP routing

Best-effort delivery

dest. address	output link
0	1
1	2
2	2
3	3

dest. address	output link
0	3
1	1
2	2
3	4

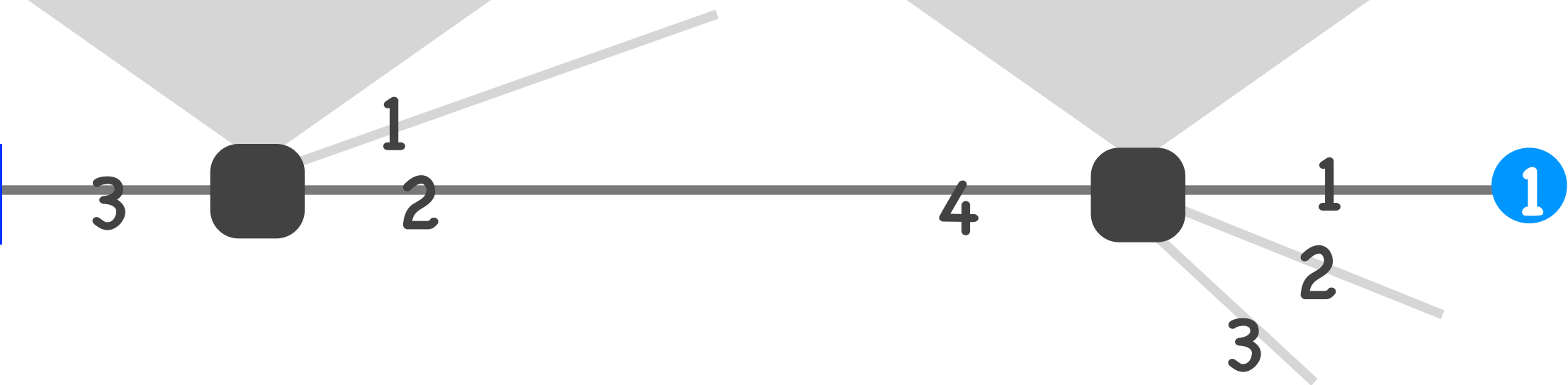
dest=1



dest. address	output link
0 - 1000	2
1001 - 1500	1
1501 - 1502	3
otherwise	1

dest. address	output link
0 - 255	1
256 - 46780	2
46781 - ...	3
otherwise	4

dest=1



dest. address range			output link
0 - 3	0000 - 0011	00**	1
4 - 7	0100 - 0111	01**	2
8 - 11	1000 - 1011	10**	3
12 - 15	1100 - 1111	11**	4

0100

dest. address range			output link
0 - 2	0000 - 0010	00**	1
3	0011	0011	2
4, 6, 7	0100, 0110, 0111	01**	3
5	0101	0101	2
8 - 15	1000 - 1111	1***	4

0000

dest. address range			output link
0 - 1	0000 - 0001	000*	1
2 - 3	0010 - 0011	001*	2
4 - 7	0100 - 0111	01**	3
8	1000	1000	2
9 - 15	1001 - 1111	1***	4

prefixes

longest prefix matching

IP / datagram network

- Uses packet switching =
no network-layer connections
- Appropriate for **best-effort service**
- Forwarding **state: per destination prefix**
 - ▶ destination prefix, output link
 - ▶ populated by routing

Why the datagram approach?

- It makes forwarding tables **smaller**
 - ▶ no per-connection state in routers
- It makes routers **simpler**
 - ▶ no support for connection setup and teardown in routers

dest. address range			output link
0 - 3	0000 - 0011	00**	1
4 - 7	0100 - 0111	01**	2
8 - 11	1000 - 1011	10**	3
12 - 15	1100 - 1111	11**	4

dest. address range			output link
0 - 2	0000 - 0010	00**	1
3	0011	0011	2
4, 6, 7	0100, 0110, 0111	01**	3
5	0101	0101	2
8 - 15	1000 - 1111	1***	4

dest. address range			output link
0 - 1	0000 - 0001	000*	1
2	0010	0010	2
3	0011	0011	3
4, 6, 7	0100, 0110, 0111	01**	2
5	0101	0101	4
8 - 15	1000 - 1111	1***	1
10	1010	1010	3

Location-dependent addresses

- Address embeds **location** information
 - ▶ address proximity implies location proximity
- Significantly **reduces forwarding state**
 - ▶ per destination prefix
 - ▶ (otherwise, it would be per destination)

IP address format

IP address = number from 0 to $2^{32}-1$

11011111 00000001 00000001 00000001

223 . 1 . 1 . 1

IP prefix format

IP prefix = range of IP addresses

223.1.1.0 / 24 ← mask

11011111 00000001 00000001 00000000

11011111 00000001 00000001 *****

223.1.1.*

IP prefix format

IP prefix = range of IP addresses

223.1.1.74 / 24 ← mask

11011111 00000001 00000001 01001001

11011111 00000001 00000001 *****

223.1.1.*

IP prefix format

IP prefix = range of IP addresses

223.1.1.74 / 24 ← mask

223.1.1.0 / 24

223.1.1.113 / 24

223.1.1.*

IP prefix format

IP prefix = range of IP addresses

223.1.1.0 / 8 ← **mask**

11011111 00000001 00000001 00000000

11011111 ***** ***** *****

223.***

IP prefix format

IP prefix = range of IP addresses

223.1.1.0 / 12 ← **mask**

11011111 00000001 00000001 00000000

11011111 0000**** *****

from: 11011111 00000000 00000000 00000000

to: 11011111 00001111 11111111 11111111

IP prefix format

IP prefix = range of IP addresses

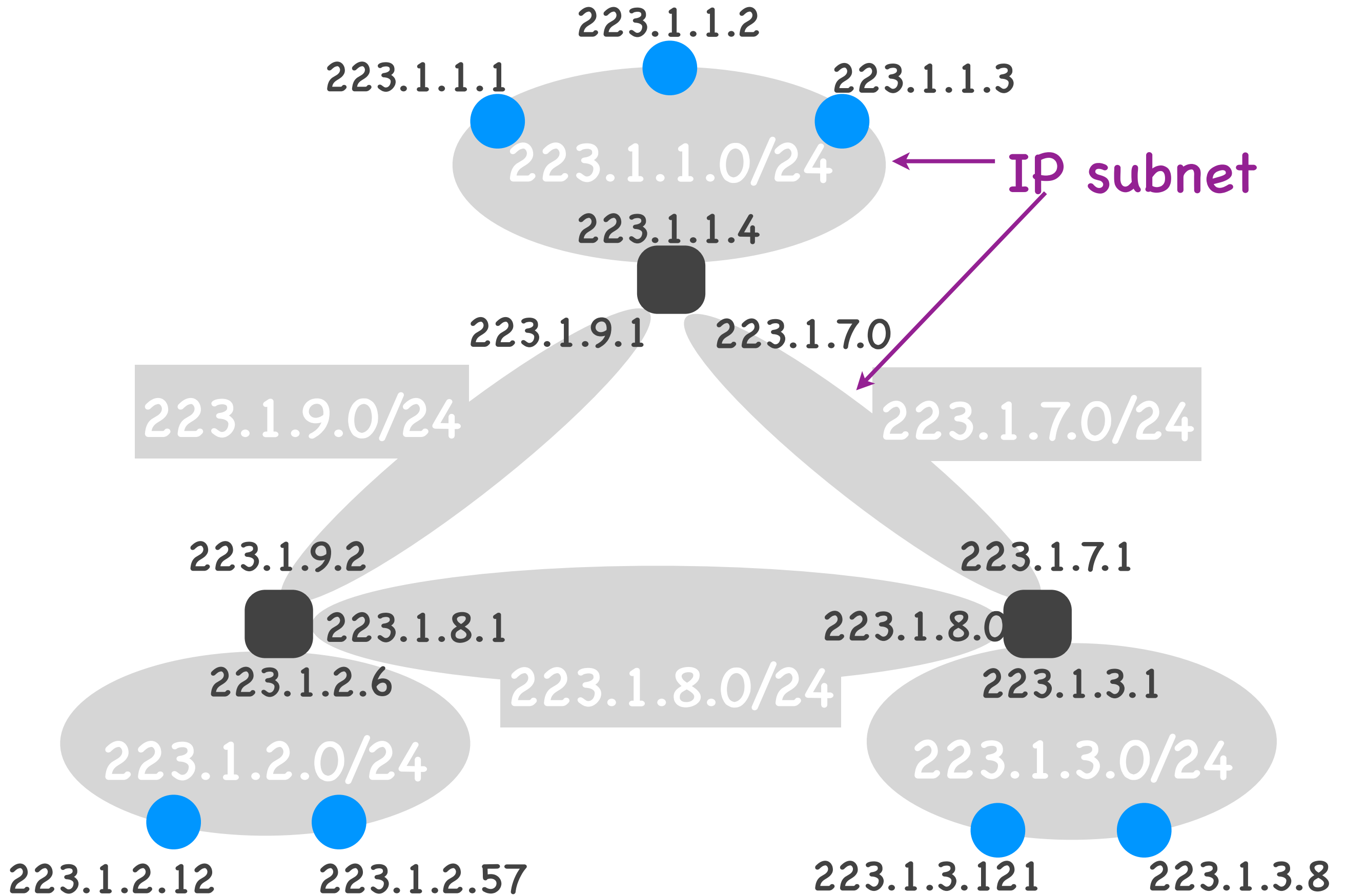
223.1.1.0 / 12 ← **mask**

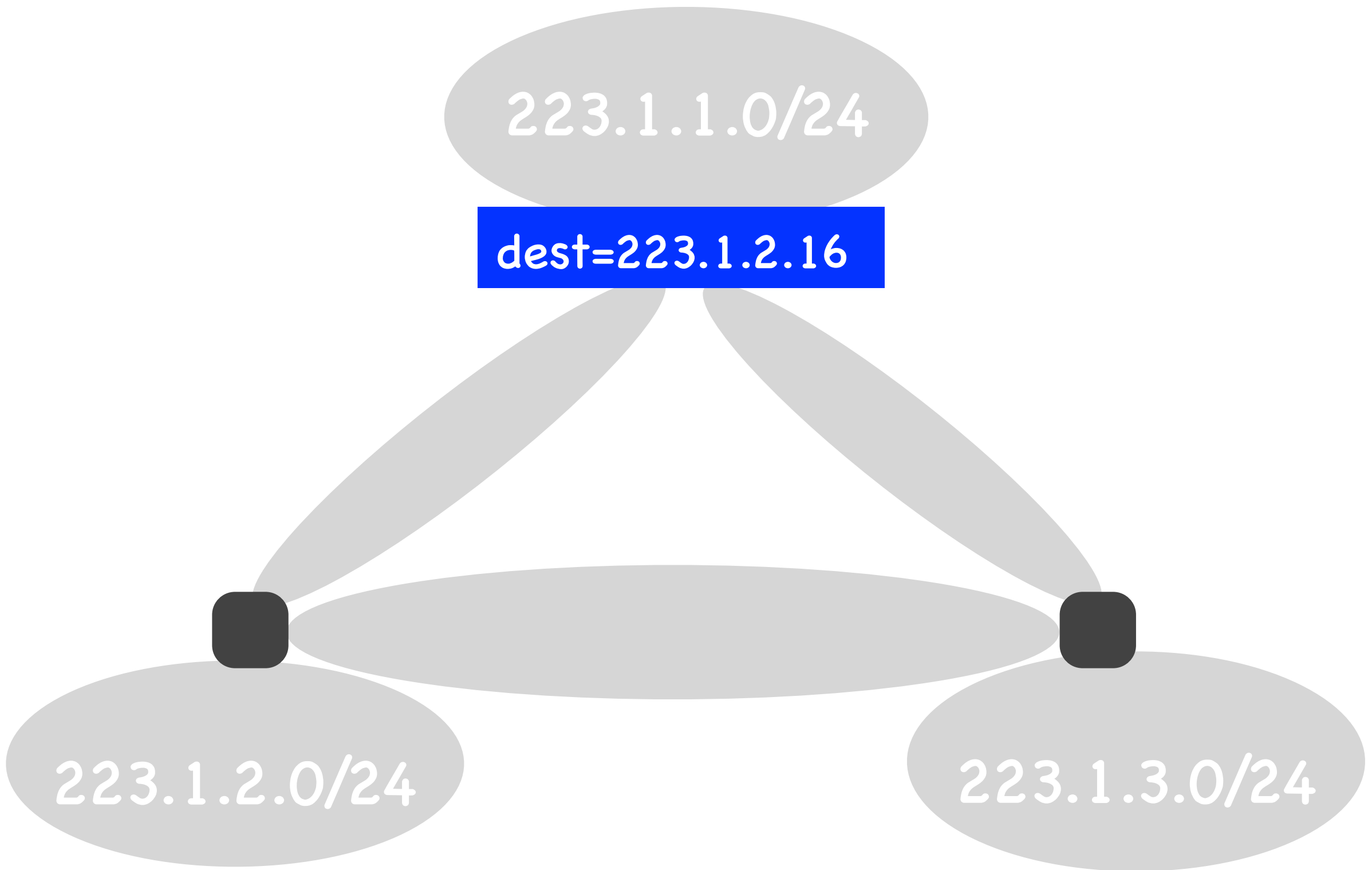
11011111 00000001 00000001 00000000

11011111 0000**** *****

from: 223.0.0.0

to: 223.15.255.255



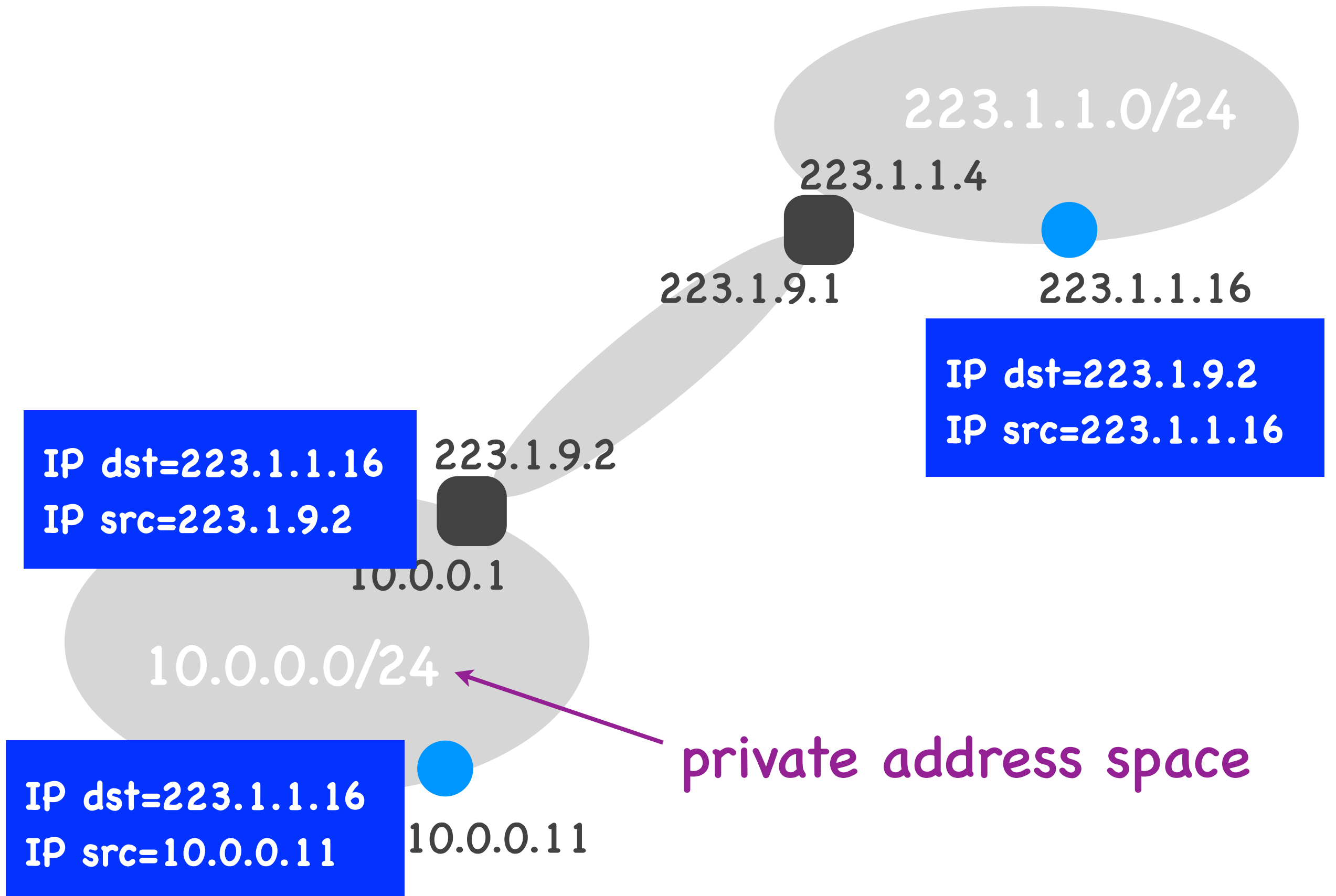


IP subnet

- (Informal) Contiguous network area that does not “contain” any routers
- All its end-systems and incident routers have IP addresses from the same IP prefix

IP address assignment

- Each organization **obtains** IP prefixes
 - ▶ from its ISP or from a regulatory body
- Network operator **assigns** IP addresses
 - ▶ to router interfaces: manually
 - ▶ to end-systems: manually or through DHCP

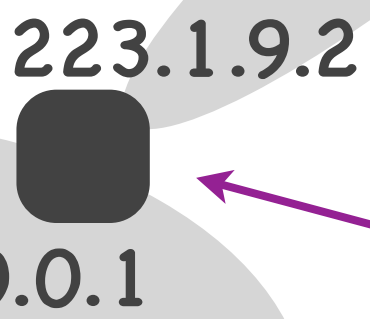


IP address	origin TCP port	new TCP port
10.0.0.11	756	954



IP dst=223.1.9.2
 IP src=223.1.1.16
 TCP dst port=954

IP dst=223.1.1.16
 IP src=223.1.9.2
 TCP src port=954



NAT gateway



IP dst=223.1.1.16
 IP src=10.0.0.11
 TCP src port=756



Network Address Translation (NAT)

Network Address Translation

- Resolves IP address depletion problem
- Requires **per-connection state** at the border routers of the private IP subnets
- End-systems inside private IP subnets are **unreachable** from the outside world

To read, if networks interest you:

- **Broadcasting & DHCP**
 - ▶ Ed. 6, p. 371 | Ed. 7, p. 370
- **ICMP & Traceroute**
 - ▶ Ed. 6, p. 379 | Ed. 7, p. 447
- **IPv6**
 - ▶ Ed. 6, p. 382 | Ed. 7, p. 376