CS 725/825 & IT 725
Lecture 18
Network Layer

October 30, 2024

IPv6 - Motivation

▶ What's wrong with IPv4?

 − not enough addresses

 − to complex to process in routers

 − autoconfiguration

 − security

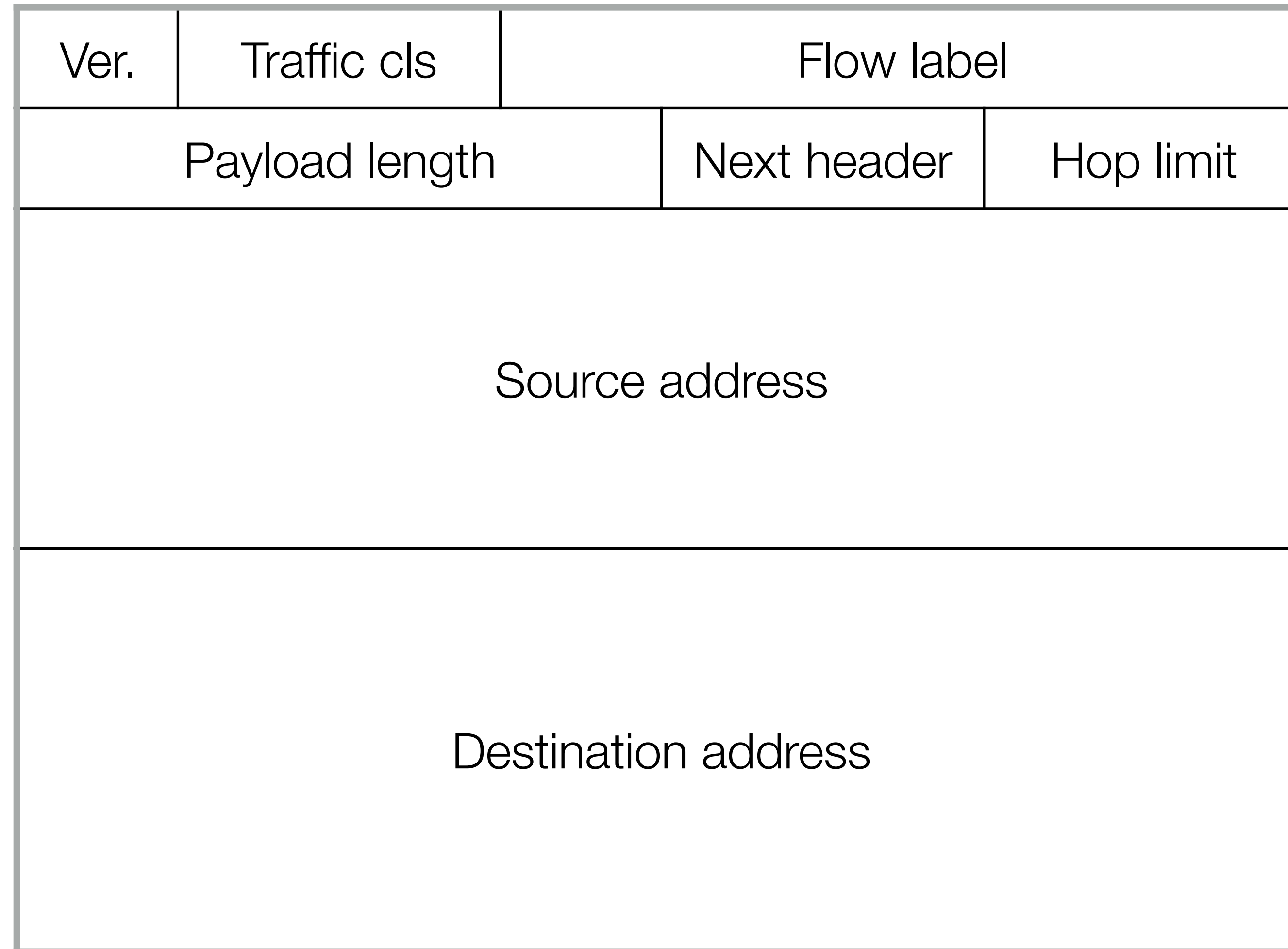▶ Can we avoid switching to IPv6?

 − Network Address Translation (NAT)

IPv6 - Protocol Design

▸ Keep the good stuff...

  – unreliable datagram service

  – TTL, TOS (for compatibility)

▸ Eliminate the unnecessary...

  – no fragmentation (only as an option)

  – no header checksums

▸ Address the issues...

  – longer addresses and more

IPv6 Header

Ver.	Traffic cls	Flow label	
Payload length		Next header	Hop limit
Source address			
Destination address			

IPv6 Address Representation

▸ An IPv6 address is represented by 8 groups of 16-bit hexadecimal values separated by colons (:)

▸ Can be abbreviated:

– omit leading zeroes in a 16-bit value

– replace one group of consecutive zeroes by a double colon

▸ Example:

– 2606:4100:38c0:9::5   vs 2606:4100:38c0:0009:0000:0000:0000:0005

Special Use IPv6 Addresses

▶ ::/128 - Unspecified address

▶ ::1/128 - Loopback address

▶ ::FFFF:0:0/96 - IPv4-mapped address

▶ FE80::/10 - Link-local unicast

▶ FF00::/8 - Multicast

Routing

▸ Approaches:

– First find a path from source to destination and then follow it… (Source Routing)

– Go to the first corner, ask for direction to the next corner that is on the way to the destination*. Repeat until you reach the destination… (Hop-by-hop Forwarding)

* Routing tables give you that information

Routing Table Content

▶ Automatically populated with entries based in local L2 configuration

▶ Static entries - added by the network administrator

▶ Dynamic entries - added by dynamic routing protocols

Static
configuration
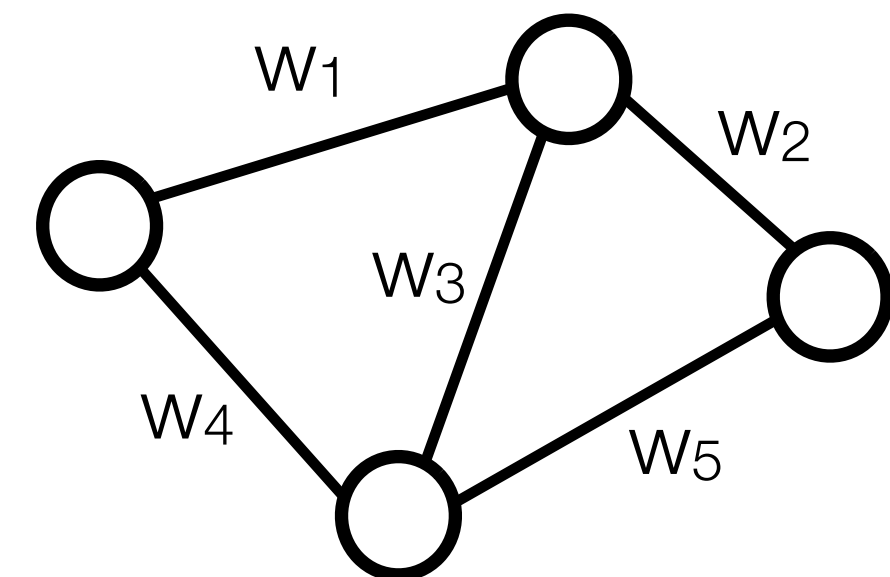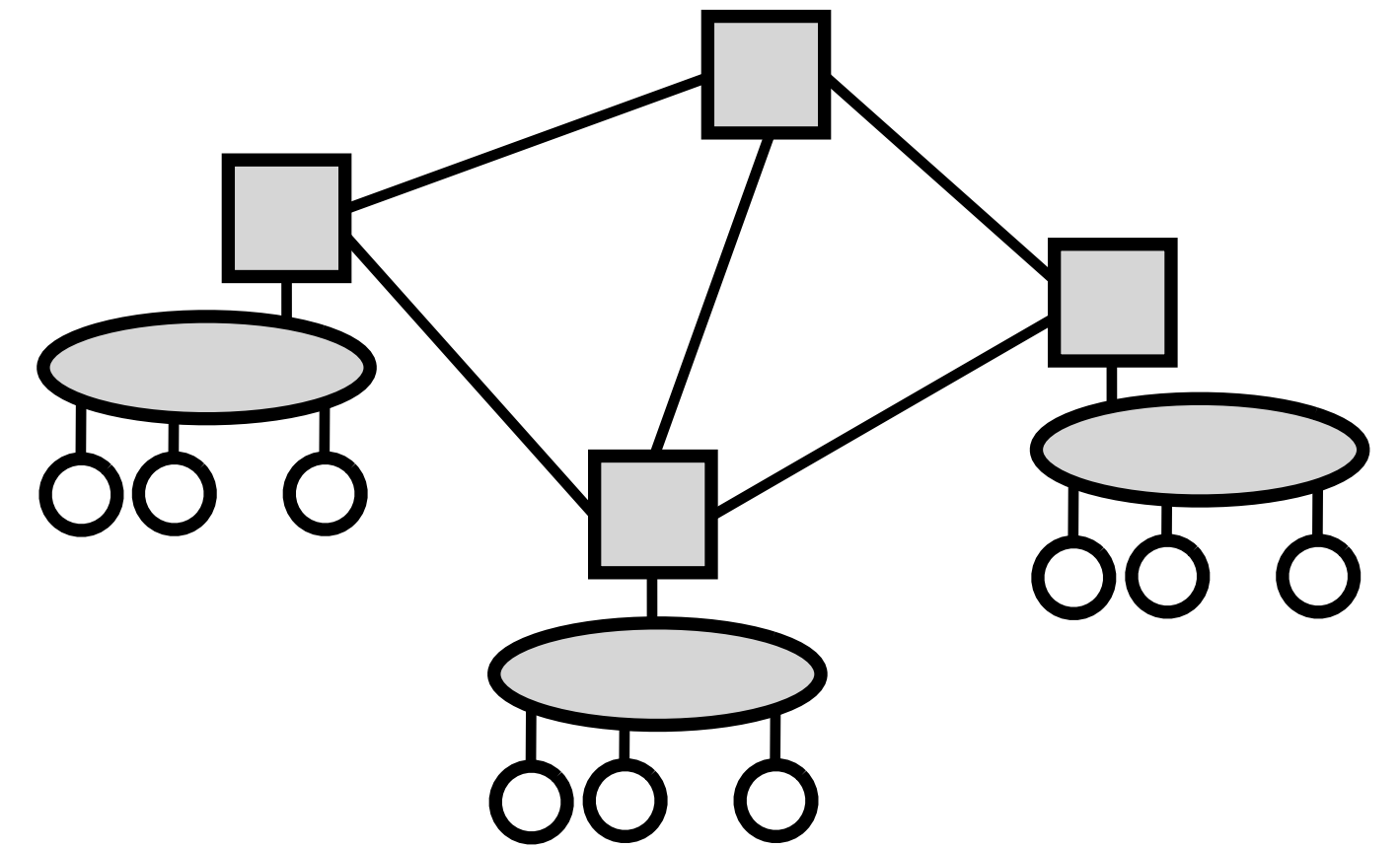
Routing protocol
domain

L2 configuration
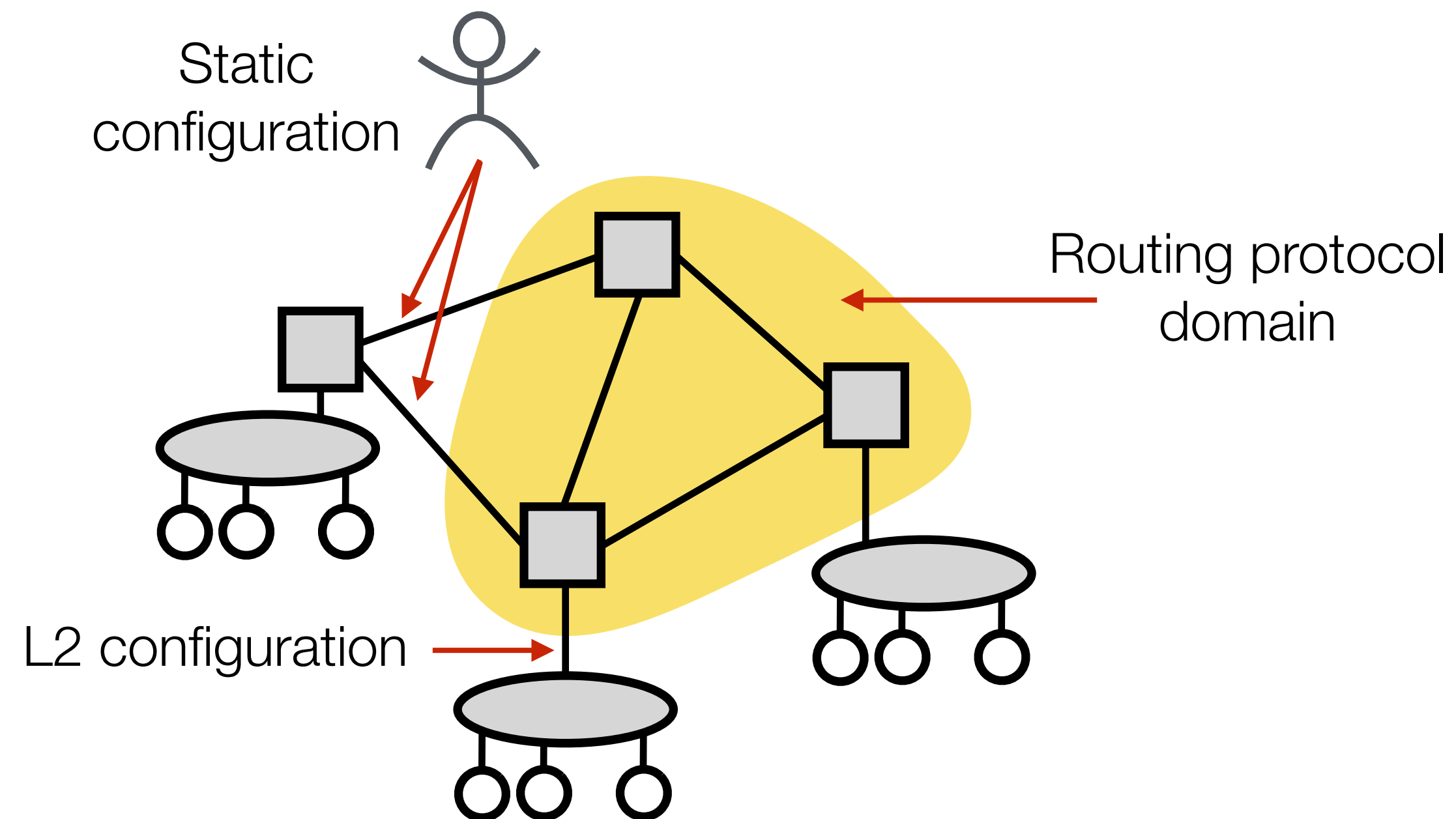
Routing

▸ Finding a good path from source to destination

  – topology discovery
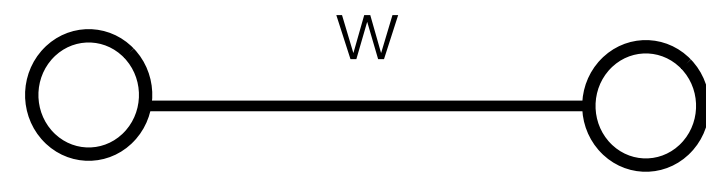
  – route selection

▸ Network as a graph...

  – links (point to point and L2 subnets) and routers

  – destinations are typically L2 subnets, not individual nodes
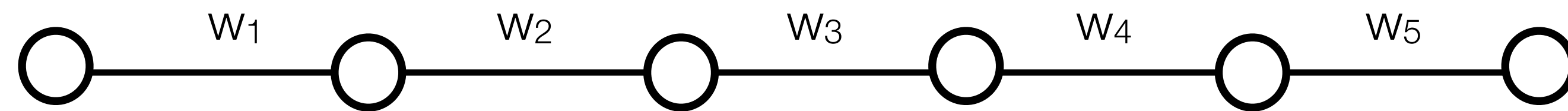
  – links may have "weights"

Link weights

▸ What is a good measure of "weight" of a link?

▸ Weight of a path?

Link & Path Measures

▶ **Link measures:**

- Throughput / bit rate

- Latency

- Loss probability

- Availability

- Current load

- Security

- Monetary cost

▶ **Path measures:**

- Sum

    Latency

    Monetary cost

- Min/Max

    Throughput / bit rate

- Product

    Loss probability

Trivial routing methods

▸ Hot potato routing (not practical)

— send to randomly chosen outgoing link…

▸ Flooding (not practical)

— send a copy to every outgoing link…

▸ Limited flooding

— every packet has a sequence number (together with the source address, this makes a copy of a packet uniquely identifiable)

— send a copy to every other outgoing link

— keep track of forwarded packets so that copies are sent only once

Routing Protocols - Categories

▶ Link State

- – exact neighbor information flooded to everyone

- – topology of the entire networks is discovered in each node

- – shortest paths calculated and used to populate the routing tables

▶ Distance Vector

- – estimates of distances to all nodes in the network sent to all neighbors

- – estimates are improved based on information from neighbors

- – the process is repeated and routing tables are populated based on the estimates

Distance Vector (recap)

▸ Estimates of distances to all nodes in the network (Distance Vector) is sent to all neighbors

▸ Estimates are improved based on information from neighbors

▸ The process is repeated and routing tables are populated based on the estimates
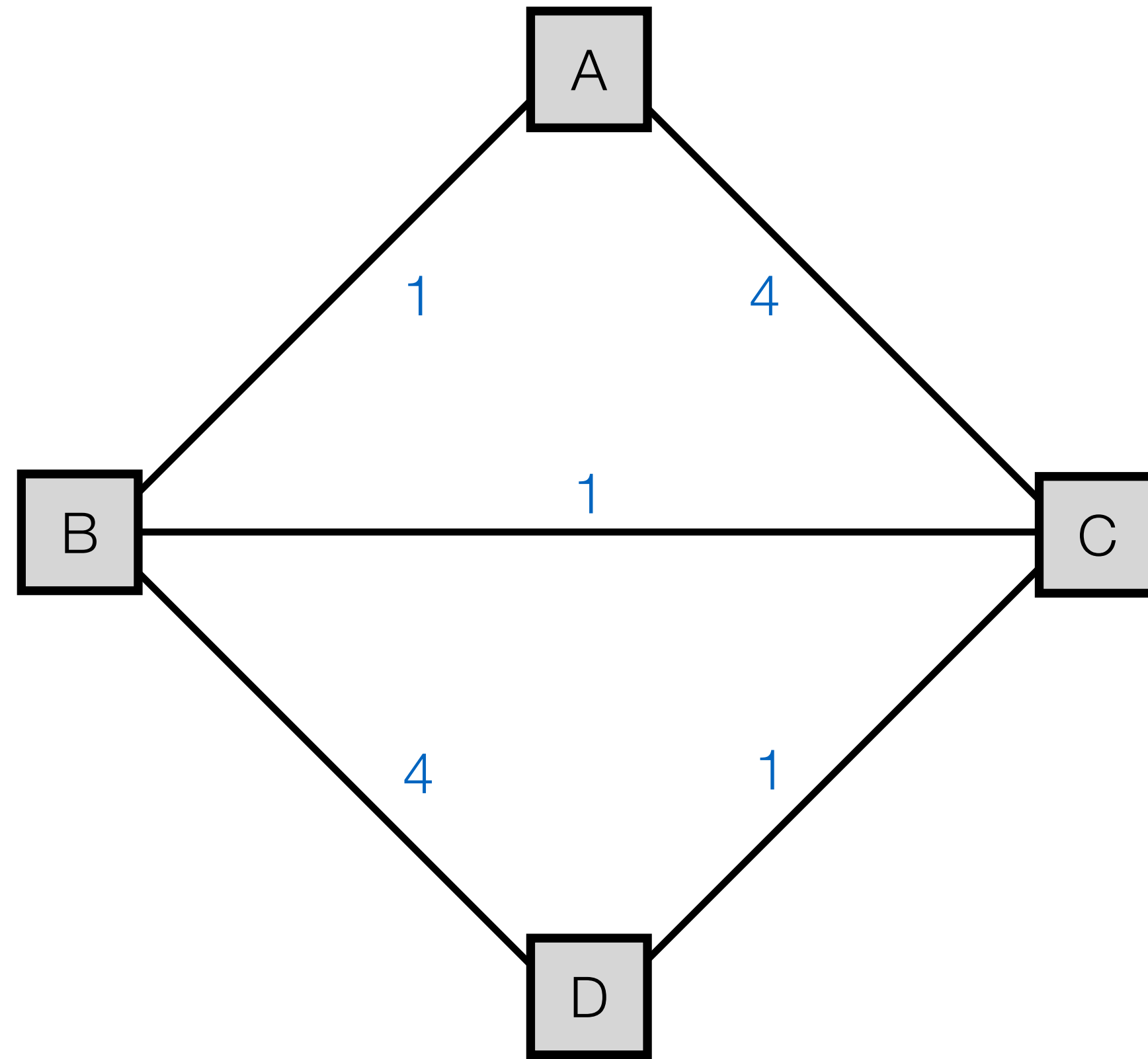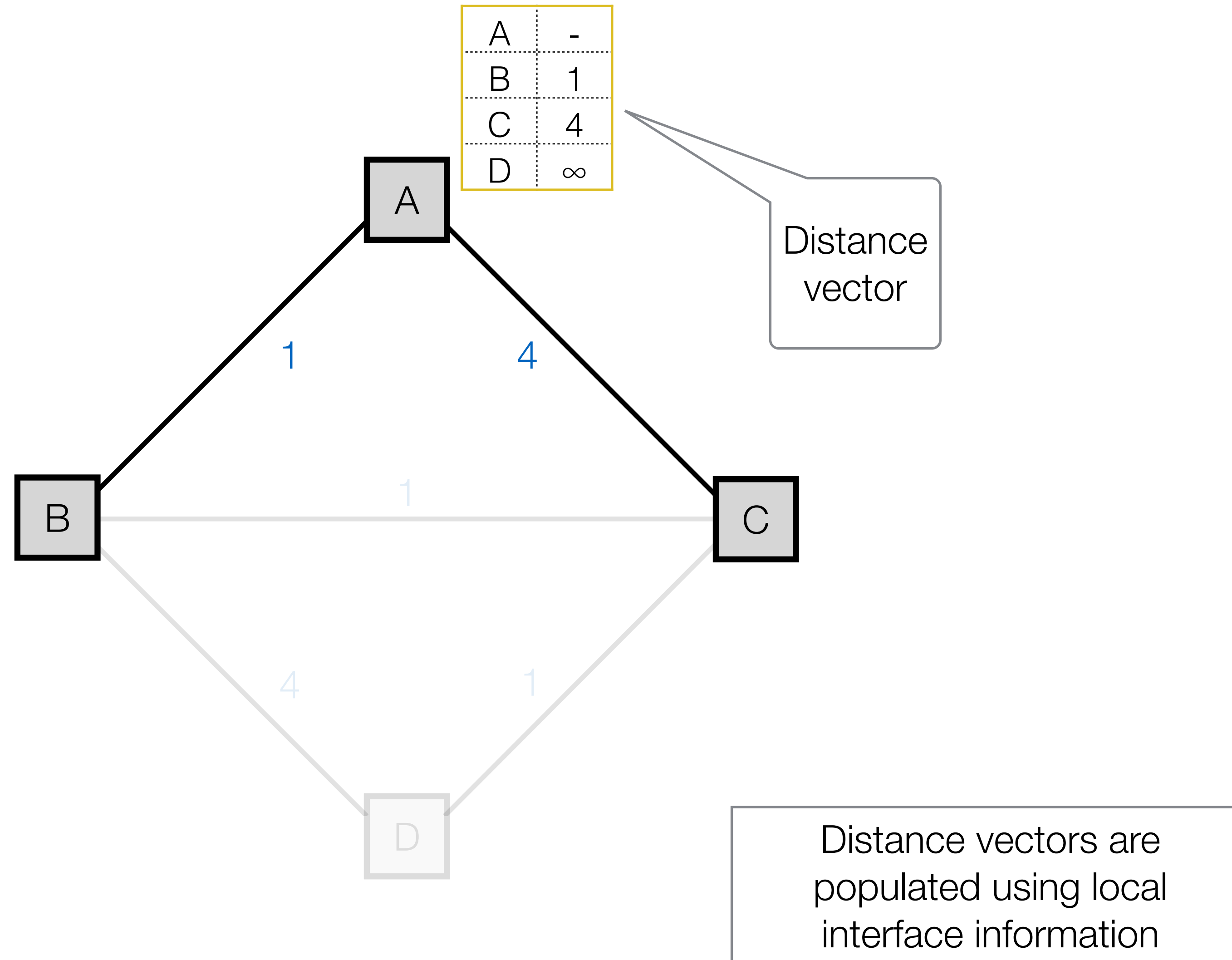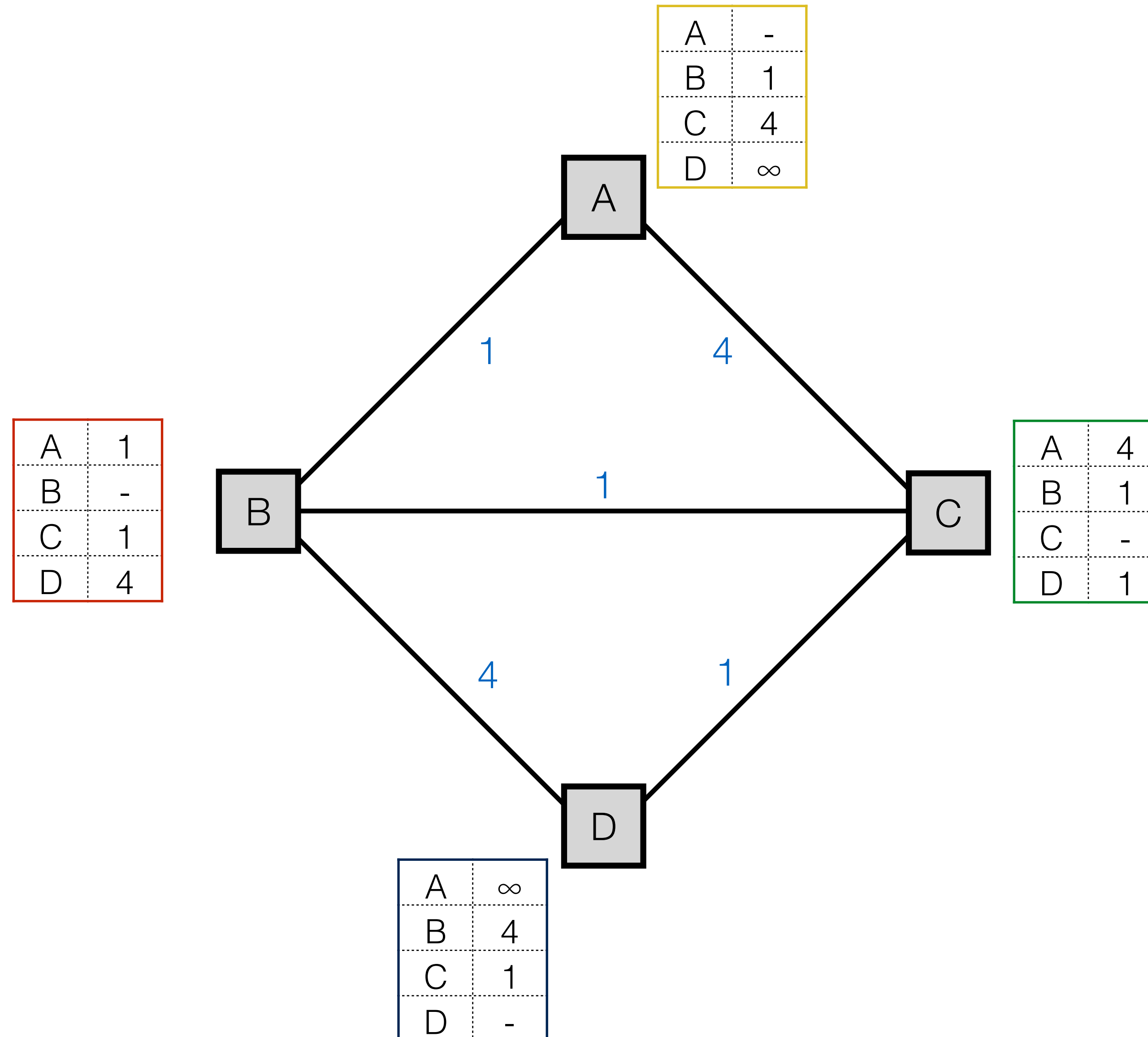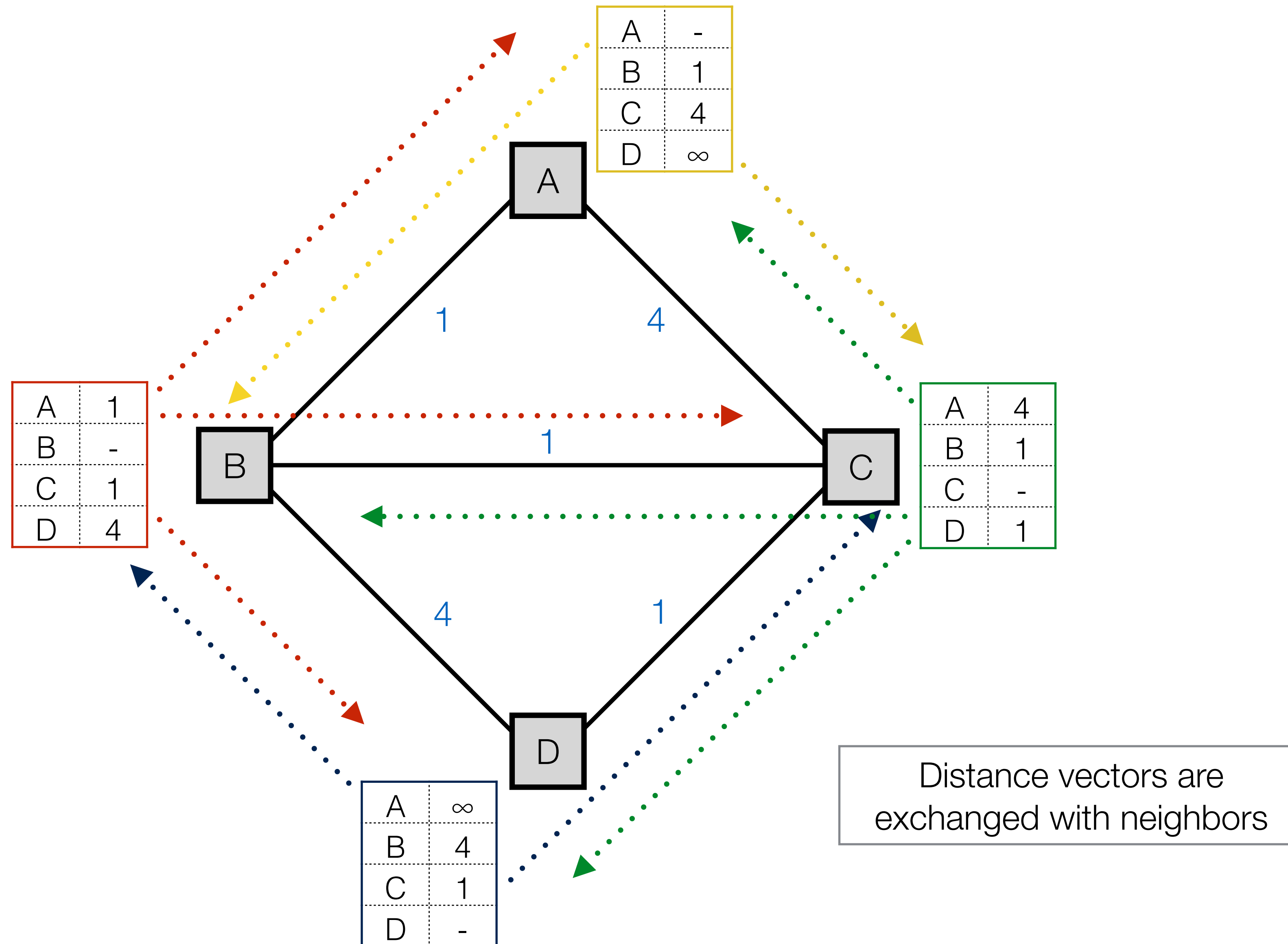
Distance Vector routing

Distance Vector routing

Distance Vector routing

Distance Vector routing

A	-
B	1
C	4
D	∞

A

A	1
B	-
C	1
D	4

B

A	4
B	1
C	-
D	1

C

A	∞
B	4
C	1
D	-

D

1   4

1

4   1

Distance vectors are
exchanged with neighbors
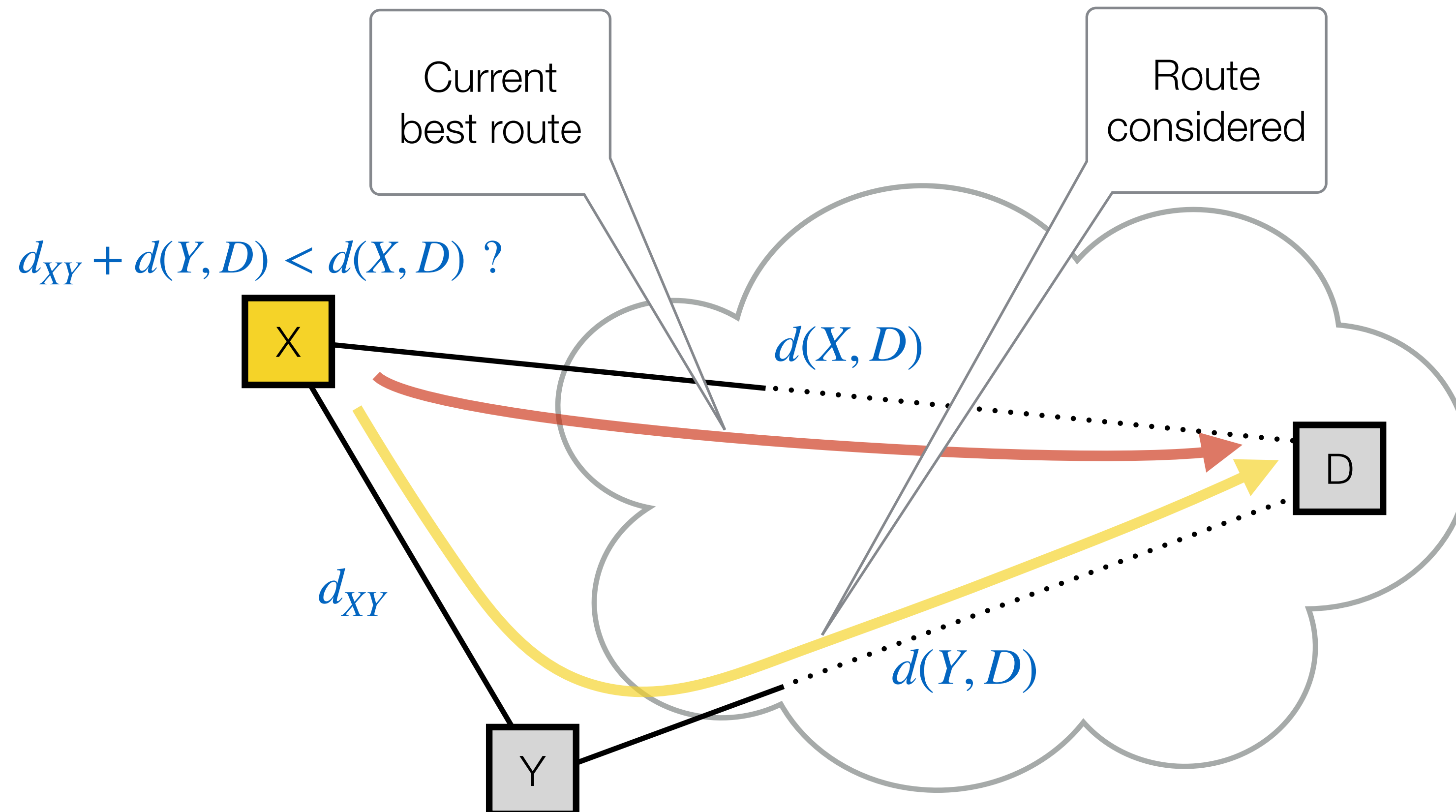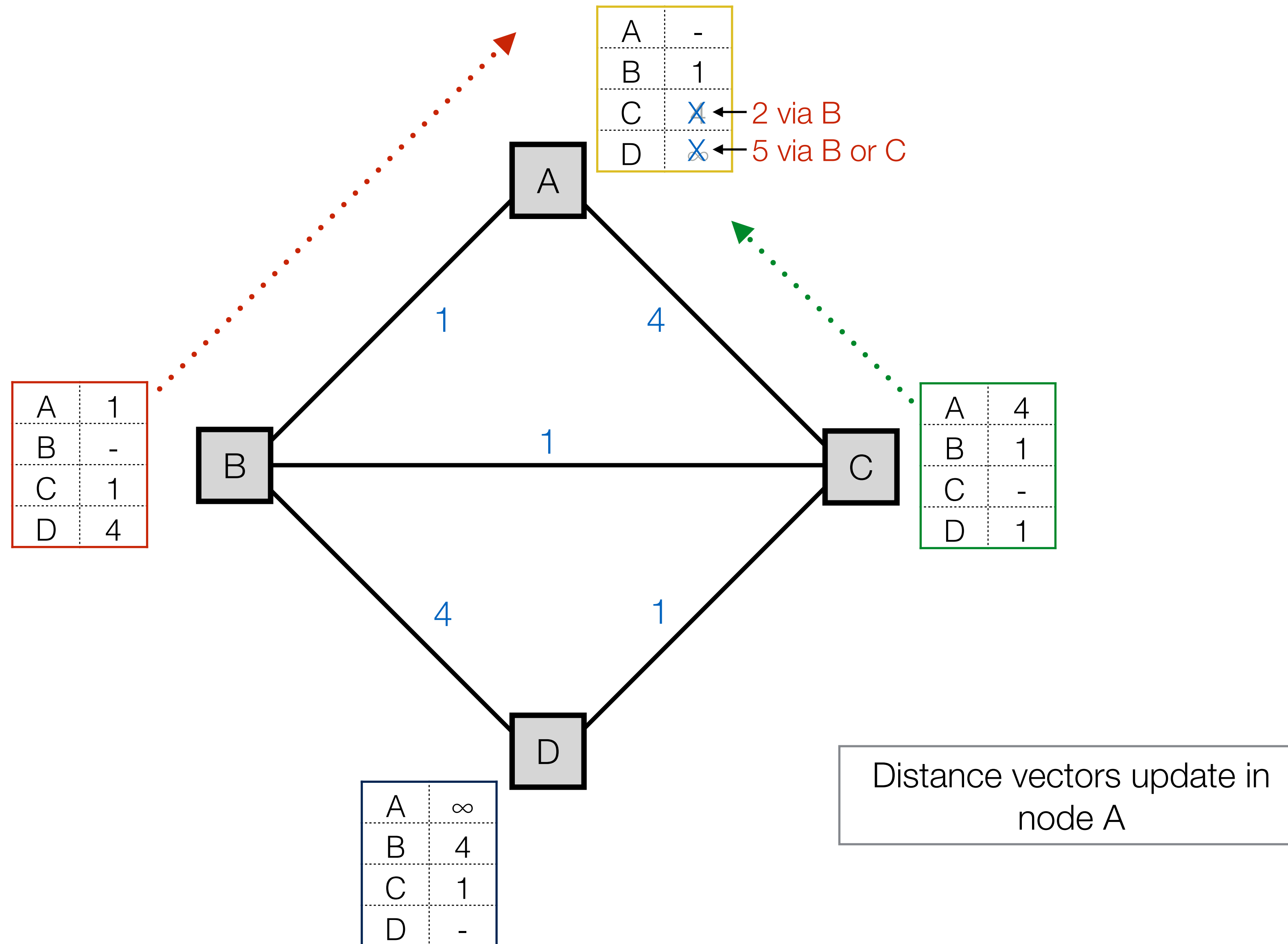
Distance Vector update

▸ Is routing to D through X's neighbor Y (with distance $d_{XY} + d(Y,D)$) better than the current best route from X (with distance $d(X,D)$)?

Distance Vector routing

A	-
B	1
C	X	← 2 via B
D	X	← 5 via B or C

A	1
B	-
C	1
D	4

A	4
B	1
C	-
D	1

A	∞
B	4
C	1
D	-

Distance vectors update in node A

Distance Vector routing

Distance vectors after the first round of updates. Shortest paths of length 2 hops or less discovered.

Distance Vector routing

Table at A:

A	-
B	1
C	X	← 2 via B
D	X	← X via B or C ← 3 via B

Table at B:

A	1
B	-
C	1
D	X 4	← 2 via C

Table at C:

A	X 4	← 2 via B
B	1
C	-
D	1

Table at D:

3 via C → X via B or C

A	∞ X
B	X 4	← 2 via C
C	1
D	-

Distance vectors after the second round of updates. Shortest paths of length 3 hops or less discovered.

Distance Vector routing