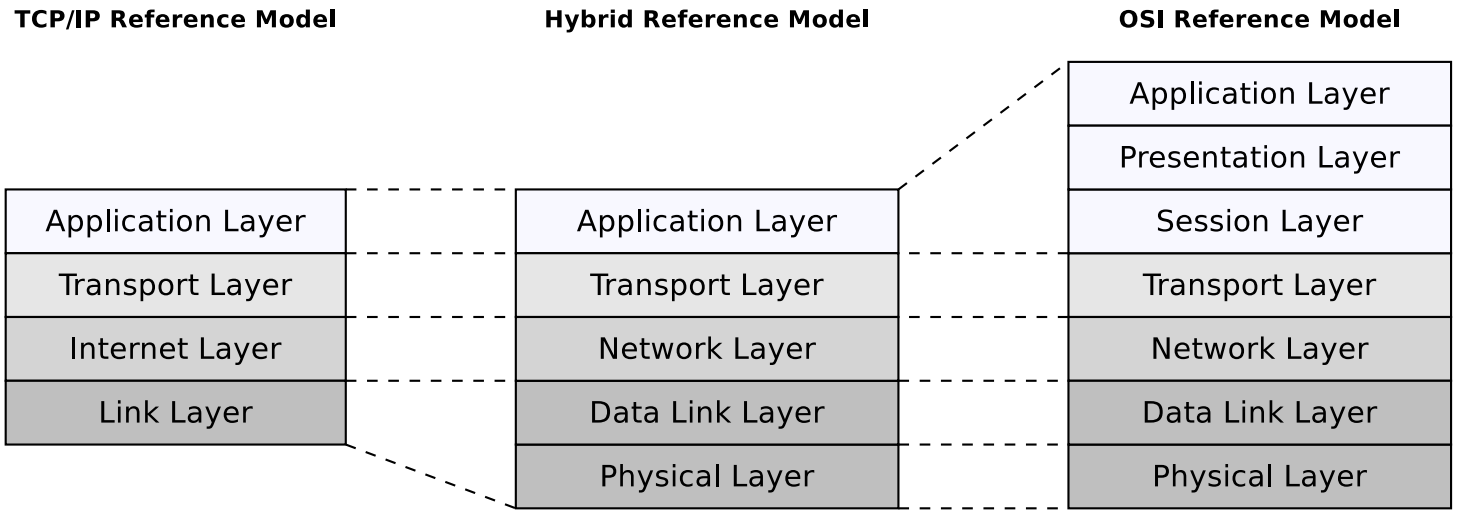
8th Slide Set
Computer Networks

Prof. Dr. Christian Baun

Frankfurt University of Applied Sciences
(1971–2014: Fachhochschule Frankfurt am Main)
Faculty of Computer Science and Engineering
christianbaun@fb2.fra-uas.de

Network Layer

- Functions of the Network Layer
  - Sender: Pack segments of the Transport Layer into pakets
  - Receiver: Identify the packets inside the frames of Data Link Layer
  - Provide logical addresses (IP addresses)
  - Determine the best path to the destination = Routing
  - Forward packets between logical networks (across different physical networks)



Exercise sheet 4 repeats the contents of this slide set which are relevant for these learning objectives

- Devices: Router, Layer-3-Switch (Router without WAN port)
- Protocols: IPv4, IPv6, ICMP, IPX/SPX, DECnet

Learning Objectives of this Slide Set

- Network Layer (part 2)
  - Forwarding and path determination
    - Distance-vector routing protocols
    - Link-state routing protocols
  - Diagnosis information and error messages via ICMP
  - Internetworking (summary)
  - Network Address Translation (NAT)

Forwarding

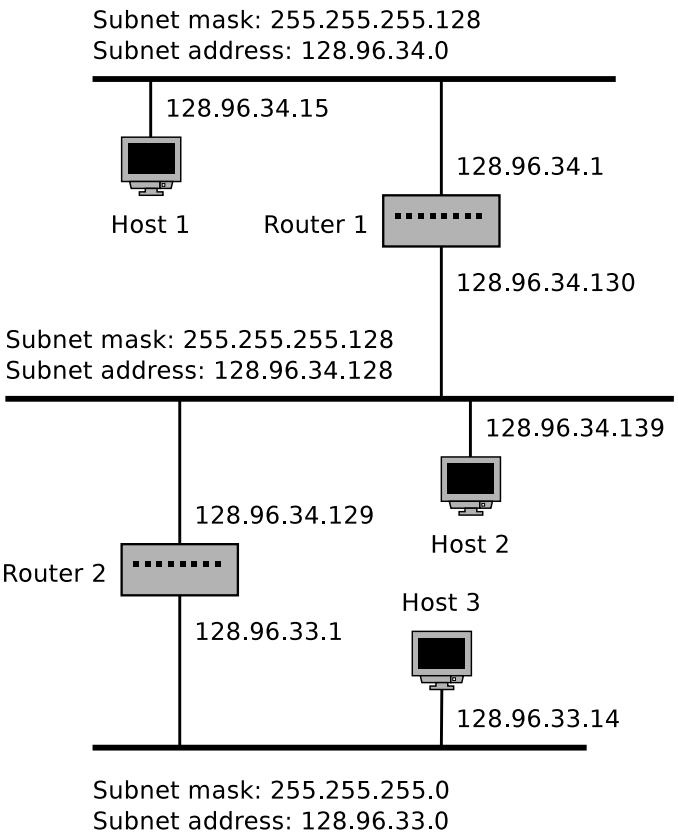


Image source: Computernetzwerke. Peterson and Davie. dpunkt (2000)

- Primary task of Routers: **Forwarding** IP packets
  - To carry out this task, Routers must determine for each incoming packet, the correct port
- Each Router maintains a local **routing table**
  - The routing table contains...
    - the **logical networks**, a Router knows about
    - the information, which logical network can be accessed via which **port**

See the addressing example for Network Layer in slide set 7

- A Router transmits the IP packets in the direction, which is specified in the routing table

Path Determination (Routing)

- The **path determination** (**routing**) is the process of creating **routing tables** by using **routing protocols**
  - The routing tables are necessary for calculating the best path, which allows to reach the destination at the lowest cost possible
  - The routing protocols are carried out between the Routers
  - Routing protocols base on **distributed algorithms**
    - Reason: scalability
- 2 major classes of routing protocols exist:
  - **Distance-vector routing protocols** (implement the Bellman-Ford algorithm)
    - Example: **Routing Information Protocol** (RIP)
  - **Link-state routing protocols** (implement the Dijkstra algorithm)
    - Example: **Open Shortest Path First** (OSPF)

The Border Gateway Protocol (BGP) implements path-vector routing which has some similarities with distance-vector-routing

"*BGP is a form of distance-vector protocol*"
Source: Computer Networks. *Andrew S. Tanenbaum, David J. Wetherall*. 5th edition. Pearson (2011). P.481

Areas of application – Autonomous Systems

- Routers are organized in **Autonomous Systems** (AS)
  - Each AS consists of a group of logical networks, which...
    - use the Internet Protocol (IP)
    - are operated and managed by the same organization (e.g. an Internet Service Provider, a corporation or university)
    - use the same routing protocol
  - The interconnected AS in their totality form the internet
- Each AS has a unique **Autonomous System Number** (ASN)
  - The ASNs are assigned by the Internet Assigned Numbers Authority (IANA) in blocks to the Regional Internet Registries
  - The Regional Internet Registries assign ASNs to entities inside their areas
    - For Europe: RIPE NCC: `http://www.ripe.net`

An ASN can be a 16-bit integer value (old standard) or a 32-bit integer value (new standard)

Map of the Internet in 2011

Source: Rajan Sodhi. PEER 1 Hosting Blog. March, 1st 2011

`http://www.peer1.com/blog/2011/03/map-of-the-internet-2011`
`https://www.vice.com/en/article/ypp5yg/a-map-of-the-internet`
`https://www.ebmag.com/something-so-geeky-its-cool-the-map-of-the-internet-9153/`

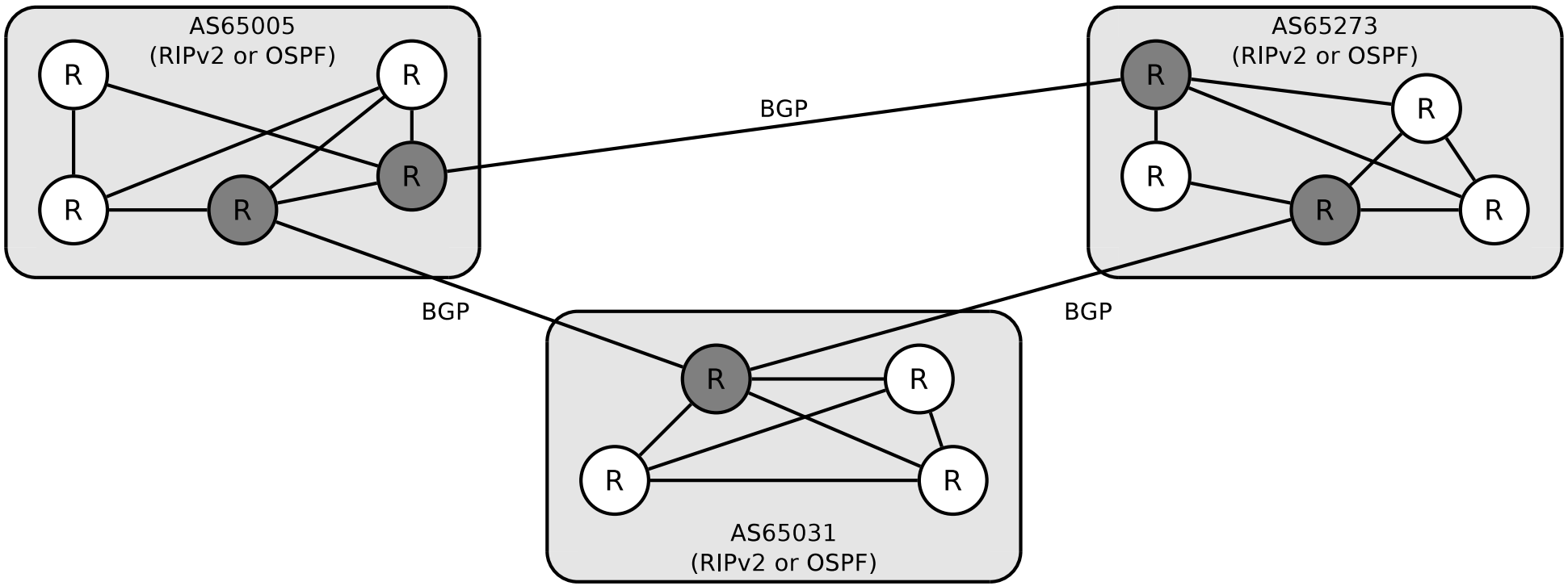
- Shows the connections between the autonomous systems
- An interactive version is available

Explanation of the author

"Each autonomous system is a network operated by a single organization, and has routing connections to some number of neighboring autonomous systems. The image depicts a graph of 19,869 autonomous system nodes, joined by 44,344 connections. The sizing and layout of the autonomous systems are based on their eigenvector centrality, which is a measure of how central to the network each autonomous system is: an autonomous system is central if it is connected to other autonomous systems that are central. This is the same graph-theoretical concept that forms the basis of Google's PageRank algorithm."

Intra-AS-Routing and Inter-AS-Routing

- For the routing inside AS ($\Longrightarrow$ **Intra-AS routing**), the operators of the AS themselves are responsible
  - Protocols for intra-AS routing are e.g. RIP and OSPF
- For the routing between AS ($\Longrightarrow$ **inter-AS routing**), the BGP is used



Convergence and Convergence Time

- **Convergence**
  - This state is reached, when after a network topology change, all Routers again have a unified view of the network
  - From this point of time, the entries in the local routing tables of the Routers are adjusted in a way that they consider the topology change
- **Convergence time**
  - The time it takes for a routing protocol to adjust the entries in the routing tables after a topology change occurred

Distance-Vector Routing Protocols (1/3)

- Implement the *Bellman-Ford algorithm*
- Example: **Routing Information Protocol** (RIP)
  - RIP allows routing inside autonomous systems (**Intra-AS-Routing**)
- Functioning of RIP:
  - Every Router sends during initialization via all its ports a **RIP request** via broadcast
    - This way, the new Router requests all neighboring (accessible) Routers to transmit their routing tables
  - With the routing information from the incoming **RIP responses**, the Router fills its local routing table with entries

Source: Ethernet. Jörg Rech. Heise (2008)

- RIPng (*RIP next generation*) supports IPv6

RFC 2080 (1997)

Distance-Vector Routing Protocols (2/3)

- Functioning of RIP (continuation):
  - Every 30 seconds, each Router sends its routing table via the connectionless Transport Layer protocol UDP to its direct neighbors
    - This periodic message is called **advertisement**
  - If a Router receives an advertisement, it checks whether the message contains entries, which are better than the entries in the routing table
    - If the received advertisement contains better routes, the Router updates the appropriate entries in its local routing table

Some words about the RIP protocol overhead

- The protocol overhead caused by RIP is low compared with other routing protocols like OSPF
- RIP does not flood the network with routing information
- One drawback of this is that the convergence time is longer

Distance-Vector Routing Protocols (3/3)

- Functioning of RIP (continuation):
  - In addition to the periodic advertisements, a Router sends an advertisement to its direct neighbors, whenever it made a change to a metric in its local routing table ($\Longrightarrow$ **triggered updates**)
  - If the Internet Protocol (IP) is used, the cost to reach a network depends only on the number of Routers, which must be passed on the way
    - The number of Routers is counted in **hops**
  - Each Router increments the hop count by 1
  - If RIP is used, each Router only knows the content of its own routing table
    - No Router has an overview of the complete network

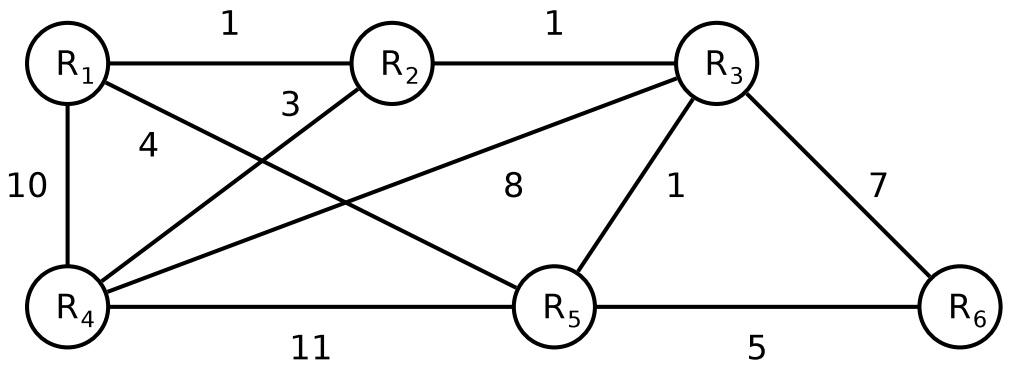
- Because no Router has an overview of the entire network, the protocol implements a **distributed algorithm**
  - This is the only way to achieve good scalability

Distance-Vector Routing Protocol – Example (1/5)

Dest.	Hop	Metric
R_1	R_1	0
R_2	?	∞
R_3	?	∞
R_4	?	∞
R_5	?	∞
R_6	?	∞

Dest.	Hop	Metric
R_1	?	∞
R_2	R_2	0
R_3	?	∞
R_4	?	∞
R_5	?	∞
R_6	?	∞

Dest.	Hop	Metric
R_1	?	∞
R_2	?	∞
R_3	R_3	0
R_4	?	∞
R_5	?	∞
R_6	?	∞



Dest.	Hop	Metric
R_1	?	∞
R_2	?	∞
R_3	?	∞
R_4	R_4	0
R_5	?	∞
R_6	?	∞

Dest.	Hop	Metric
R_1	?	∞
R_2	?	∞
R_3	?	∞
R_4	?	∞
R_5	R_5	0
R_6	?	∞

Dest.	Hop	Metric
R_1	?	∞
R_2	?	∞
R_3	?	∞
R_4	?	∞
R_5	?	∞
R_6	R_6	0

- Initialization of the tables via $\text{Hop}_{ij} \longleftarrow$? and $\text{Metric}_{ij} \longleftarrow \infty$ for $i \neq j$ and $\text{Hop}_{ij} \longleftarrow R_i$ and $\text{Metric}_{ij} \longleftarrow 0$ for $i = j$
- For each direct neighbor R_j of R_i this information is stored: $\text{Hop}_{ij} \longleftarrow R_j$ and $\text{Metric}_{ij} \longleftarrow \text{Distance}(R_i, R_j)$
- The distance is set to value 1 when the hop metric is used
- Each direct neighbor R_j of R_i sends his routing table to R_i
- For a table entry to R_k it is verified if $\text{Metric}_{ij} + \text{Metric}_{jk} < \text{Metric}_{ik}$
- If this is true, these assignments are made: $\text{Hop}_{ik} \longleftarrow R_j$ and $\text{Metric}_{ik} \longleftarrow \text{Metric}_{ij} + \text{Metric}_{jk}$

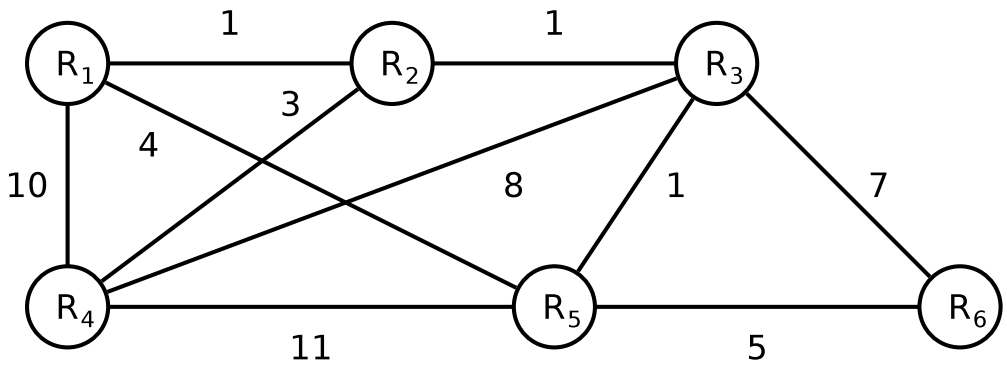
Source: Jörg Roth. **Prüfungstrainer Rechnernetze: Aufgaben und Lösungen**. Vieweg (2010)

Distance-Vector Routing Protocol – Example (2/5)

Dest.	Hop	Metric
R_1	R_1	0
R_2	R_2	1
R_3	?	∞
R_4	R_4	10
R_5	R_5	4
R_6	?	∞

Dest.	Hop	Metric
R_1	R_1	1
R_2	R_2	0
R_3	R_3	1
R_4	R_4	3
R_5	?	∞
R_6	?	∞

Dest.	Hop	Metric
R_1	?	∞
R_2	R_2	1
R_3	R_3	0
R_4	R_4	8
R_5	R_5	1
R_6	R_6	7



Dest.	Hop	Metric
R_1	R_1	10
R_2	R_2	3
R_3	R_3	8
R_4	R_4	0
R_5	R_5	11
R_6	?	∞

Dest.	Hop	Metric
R_1	R_1	4
R_2	?	∞
R_3	R_3	1
R_4	R_4	11
R_5	R_5	0
R_6	R_6	5

Dest.	Hop	Metric
R_1	?	∞
R_2	?	∞
R_3	R_3	7
R_4	?	∞
R_5	R_5	5
R_6	R_6	0

- Store distances to the direct neighbors

This example has a major drawback

- The example demonstrates very well the functioning of the Bellman-Ford algorithm
- But the example is more complex than the reality with IP
- When IP is used, the path cost to reach a network depends only on the number of Routers, which must be passed on the way
- Path cost for certain connections, as shown in the example, does not exist in IP

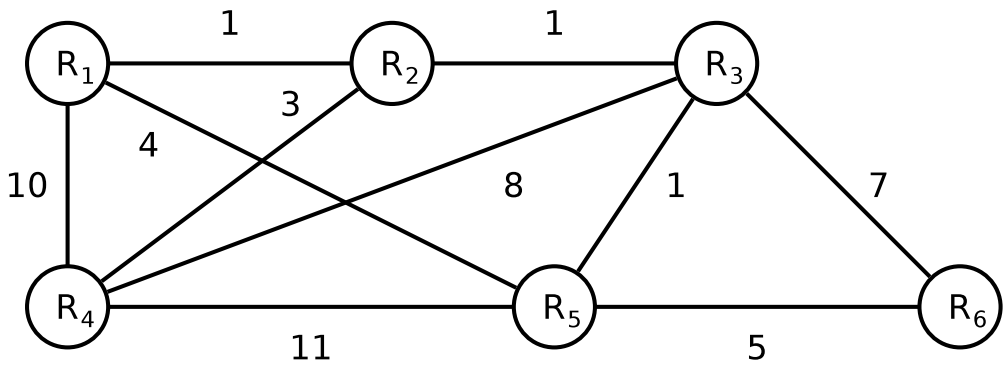
Source: Jörg Roth. **Prüfungstrainer Rechnernetze: Aufgaben und Lösungen**. Vieweg (2010)

Distance-Vector Routing Protocol – Example (3/5)

Dest.	Hop	Metric
R_1	R_1	0
R_2	R_2	1
R_3	R_2	2
R_4	R_2	4
R_5	R_5	4
R_6	R_5	9

Dest.	Hop	Metric
R_1	R_1	1
R_2	R_2	0
R_3	R_3	1
R_4	R_4	3
R_5	R_3	2
R_6	R_3	8

Dest.	Hop	Metric
R_1	R_2	2
R_2	R_2	1
R_3	R_3	0
R_4	R_2	4
R_5	R_5	1
R_6	R_5	6



Dest.	Hop	Metric
R_1	R_2	4
R_2	R_2	3
R_3	R_2	4
R_4	R_4	0
R_5	R_3	9
R_6	R_3	15

Dest.	Hop	Metric
R_1	R_1	4
R_2	R_3	2
R_3	R_3	1
R_4	R_3	9
R_5	R_5	0
R_6	R_6	5

Dest.	Hop	Metric
R_1	R_5	9
R_2	R_3	8
R_3	R_5	6
R_4	R_3	15
R_5	R_5	5
R_6	R_6	0

- Compare each entry in the local routing table with the tables of the direct neighbors and adjust table entries when necessary

This example has a major drawback

- The example demonstrates very well the functioning of the Bellman-Ford algorithm
- But the example is more complex than the reality with IP
- When IP is used, the path cost to reach a network depends only on the number of Routers, which must be passed on the way
- Path cost for certain connections, as shown in the example, does not exist in IP

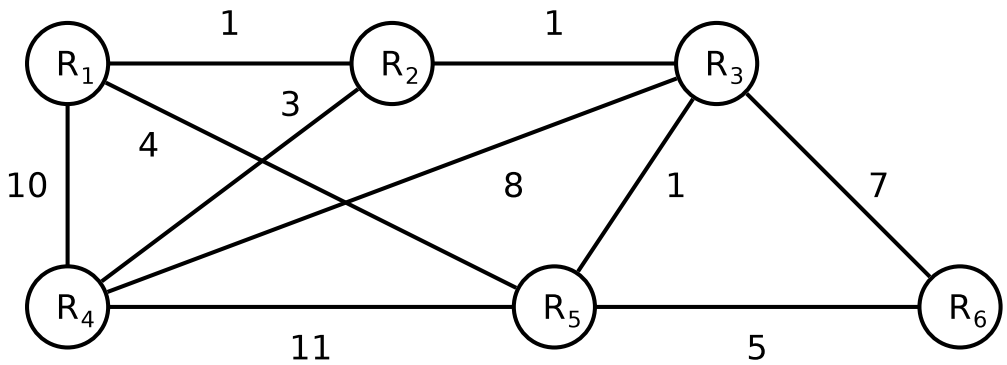
Source: Jörg Roth. **Prüfungstrainer Rechnernetze: Aufgaben und Lösungen**. Vieweg (2010)

Distance-Vector Routing Protocol – Example (4/5)

Dest.	Hop	Metric
R_1	R_1	0
R_2	R_2	1
R_3	R_2	2
R_4	R_2	4
R_5	R_2	3
R_6	R_5	9

Dest.	Hop	Metric
R_1	R_1	1
R_2	R_2	0
R_3	R_3	1
R_4	R_4	3
R_5	R_3	2
R_6	R_3	7

Dest.	Hop	Metric
R_1	R_1	2
R_2	R_2	1
R_3	R_3	0
R_4	R_2	4
R_5	R_5	1
R_6	R_5	6



Dest.	Hop	Metric
R_1	R_2	4
R_2	R_2	3
R_3	R_2	4
R_4	R_4	0
R_5	R_2	5
R_6	R_2	11

Dest.	Hop	Metric
R_1	R_3	3
R_2	R_3	2
R_3	R_3	1
R_4	R_3	5
R_5	R_5	0
R_6	R_6	5

Dest.	Hop	Metric
R_1	R_5	9
R_2	R_5	7
R_3	R_5	6
R_4	R_3	11
R_5	R_5	5
R_6	R_6	0

- Compare each entry in the local routing table with the tables of the direct neighbors and adjust table entries when necessary

This example has a major drawback

- The example demonstrates very well the functioning of the Bellman-Ford algorithm
- But the example is more complex than the reality with IP
- When IP is used, the path cost to reach a network depends only on the number of Routers, which must be passed on the way
- Path cost for certain connections, as shown in the example, does not exist in IP

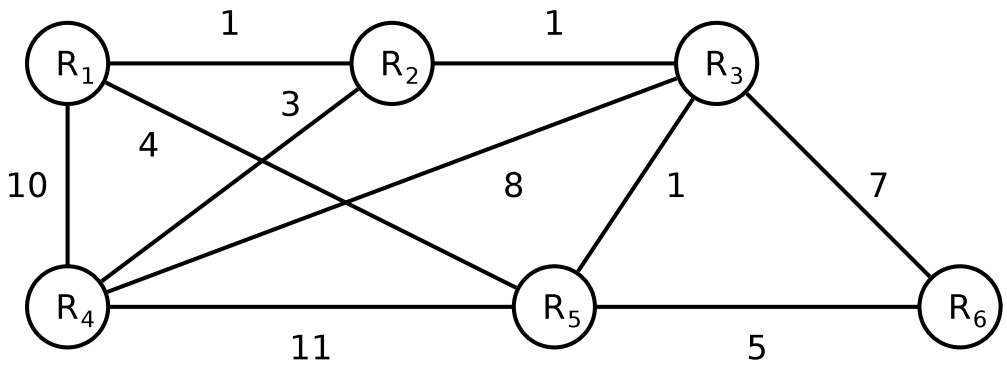
Source: Jörg Roth. **Prüfungstrainer Rechnernetze: Aufgaben und Lösungen**. Vieweg (2010)

Distance-Vector Routing Protocol – Example (5/5)

Dest.	Hop	Metric
R_1	R_1	0
R_2	R_2	1
R_3	R_2	2
R_4	R_2	4
R_5	R_2	3
R_6	R_2	8

Dest.	Hop	Metric
R_1	R_1	1
R_2	R_2	0
R_3	R_3	1
R_4	R_4	3
R_5	R_3	2
R_6	R_3	7

Dest.	Hop	Metric
R_1	R_1	2
R_2	R_2	1
R_3	R_3	0
R_4	R_2	4
R_5	R_5	1
R_6	R_5	6



Dest.	Hop	Metric
R_1	R_2	4
R_2	R_2	3
R_3	R_2	4
R_4	R_4	0
R_5	R_2	5
R_6	R_2	10

Dest.	Hop	Metric
R_1	R_3	3
R_2	R_3	2
R_3	R_3	1
R_4	R_3	5
R_5	R_5	0
R_6	R_6	5

Dest.	Hop	Metric
R_1	R_5	8
R_2	R_5	7
R_3	R_5	6
R_4	R_5	10
R_5	R_5	5
R_6	R_6	0

- Compare each entry in the local routing table with the tables of the direct neighbors and adjust table entries when necessary

This example has a major drawback

- The example demonstrates very well the functioning of the Bellman-Ford algorithm
- But the example is more complex than the reality with IP
- When IP is used, the path cost to reach a network depends only on the number of Routers, which must be passed on the way
- Path cost for certain connections, as shown in the example, does not exist in IP

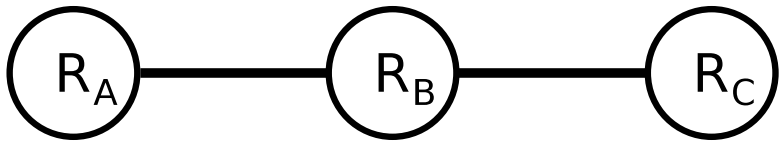
Source: Jörg Roth. **Prüfungstrainer Rechnernetze: Aufgaben und Lösungen**. Vieweg (2010)

Limited Metric

- RIP has a limited metric
- The **metric** (= **cost**) is the effort, which is required to reach a network
- If the Internet Protocol (IP) is used, the **only supported metric is the hop count**
  - It describes the number of Routers, which need to be passed on the path to the destination network
- If a network cannot be reached, its hop count with RIP is value 16 ($= \infty$ cost)
  - Therefore, RIP only allows computer networks with a maximum length of 15 Routers

Count-to-Infinity Problem (1/2)

- Drawback of the algorithm, which is implemented by RIP:
  - **Slow propagation of bad news**
- Example:



- With each advertisement round the distance values (route and cost) to Router A are propagated more and more
  - The table contains the stored distance to Router R_A inside the routing tables of R_A, R_B and R_C

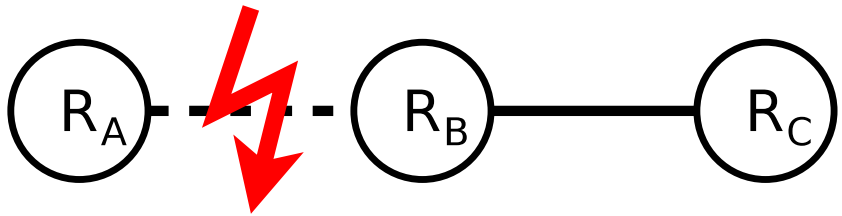
A	**B**	**C**	
0	∞	∞	Initial record
0	1	∞	After advertisement round 1
0	1	2	After advertisement round 2
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Count-to-Infinity Problem (2/2)

A	B	C	
0	1	2	Initial record
0	3	2	After advertisement round 1
0	3	4	After advertisement round 2
0	5	4	After advertisement round 3
0	5	6	After advertisement round 4
0	7	6	After advertisement round 5
0	7	8	After advertisement round 6
0	9	8	After advertisement round 7
0	9	10	After advertisement round 8
0	11	10	After advertisement round 9
0	11	12	After advertisement round 10
0	13	12	After advertisement round 11
0	13	14	After advertisement round 12
0	15	14	After advertisement round 13
0	15	∞	After advertisement round 14
0	∞	∞	After advertisement round 15

$\Longrightarrow$ **Count-to-Infinity**

- Scenario: The link to Router A fails



- During advertisement round 1, R_B gets no more information from R_A and supposes that the best path to reach R_A is via R_C
- During advertisement round 2, R_C receives the information that its neighbor R_B can reach R_A with 3 hops and therefore it stores hop count value 4 in its local routing table
- ...

Split Horizon

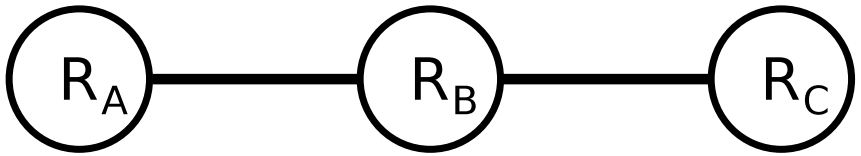
- Caused by the count-to-infinity problem, much time is wasted until the inaccessibility of a Route is detected

Advertisement messages are exchanged every 30 s. Without triggered updates, it may take up to $15 * 30\,\text{s} = 7{:}30$ minutes until a network failure between 2 Routers is detected and the affected routes get marked as not available in the routing tables

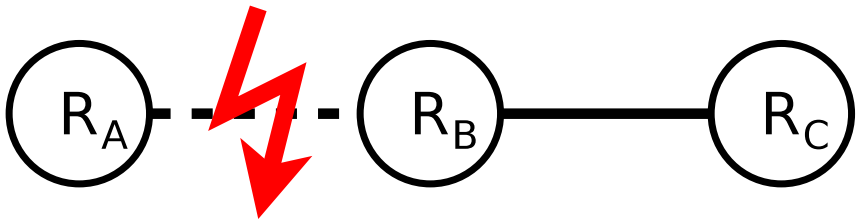
- Solution **in some use cases**: Split Horizon
  - It prevents routing loops between **2 Routers**
- A routing information must not be published via the port through which it was received
  - This prevents a Router from transmitting back a routing information to the Router, from which it learned the route
- In order to implement Split Horizon, not only the hop count and the address of the next Router (next hop) needs to be recorded in the routing table for every destination network, but also the information from which Router (port) the information was received (learned)

Split Horizon – Example

- R_c learned from R_B that R_A can be reached via R_B



- Scenario: R_A cannot be reached any more

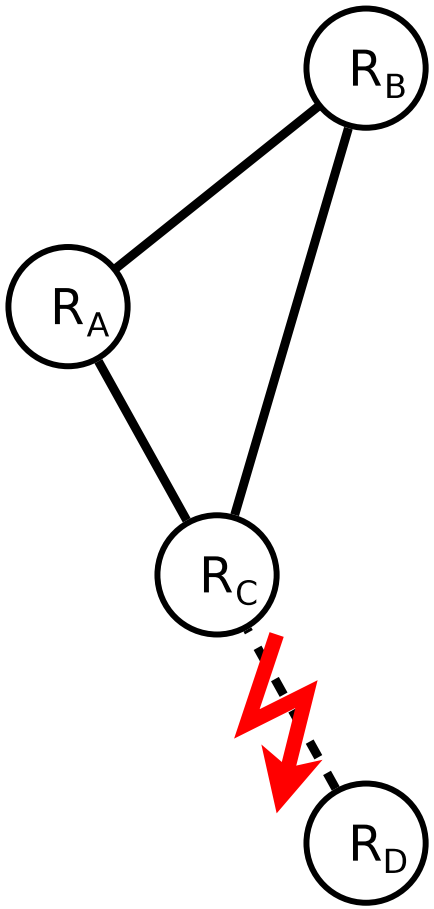


- Effect of split horizon:
  - R_B informs with its next advertisement to R_C that R_A is not reachable any more
  - R_C modifies its routing table and neither now nor in the future sends routing information for R_A to R_B

Problem: Split horizon fails in many cases

An Example where Split Horizon fails

- If the connection between R_C and R_D fails, R_C labels R_D in his local routing table as not accessible

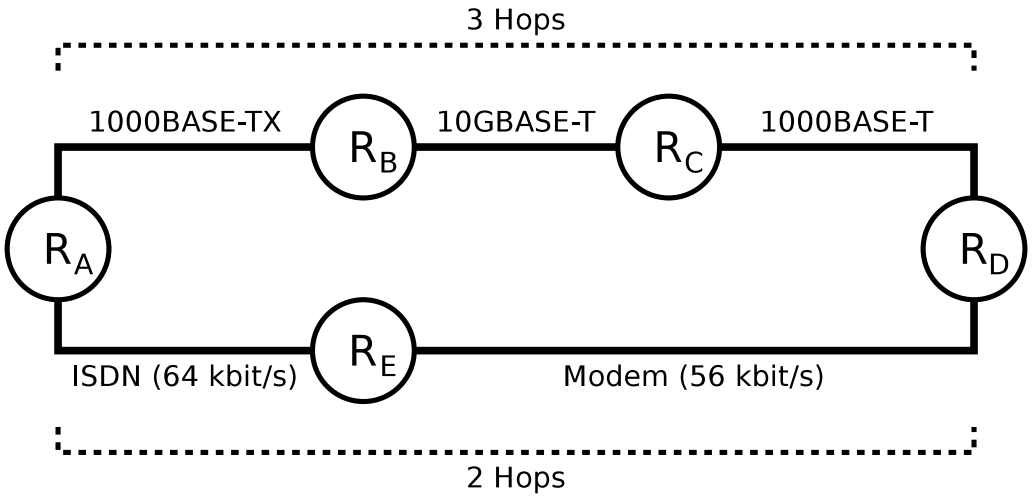


- R_C informs R_A and R_B, that R_D cannot be reached
- If the advertisement message arrives first at R_A, it assumes the best route to R_D is via R_B
- R_A informs R_B that R_D cannot be reached and informs R_C that it reaches R_D by 3 hops
- R_C believes that it can reach R_D via R_A by 4 hops and it informs R_B that it has a route to R_D
- R_B informs R_A that it reaches R_D by 5 hops
- R_A informs R_C that it reaches R_D by 6 hops
- ...

$\Longrightarrow$ **Count-to-Infinity**

RIP – Conclusion

- RIPv1 (RFC 1058) was developed and became established at a time, when computer networks were relatively small
  - RIPv1 only supports network classes and no subnets
- When RIPv1 was developed, computer networks contained seldom different transmission media with significant differences regarding connection quality and transmission rate



- Today, the hop count metric often results in **routes, which are not optimal**, because all network segments have an **equal weight**
- RIPv2 (RFC 2453) supports subnets and can distinguish between internal and external routes (see slide 26)

Structure of RIPv1 Messages (*Advertisements*)

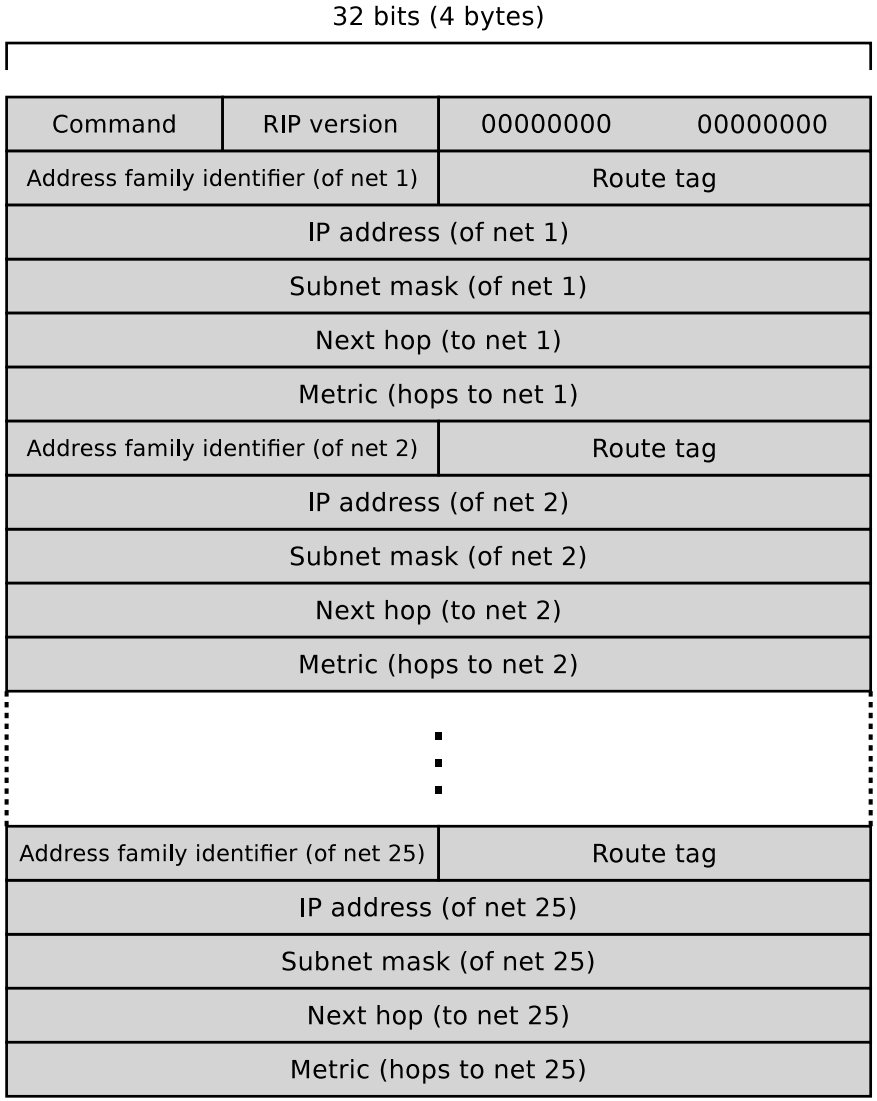
32 bits (4 bytes)

Command	RIP version	00000000	00000000
Address family identifier (of net 1)		00000000	00000000
IP address (of net 1)			
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
Metric (hops to net 1)			
Address family identifier (of net 2)		00000000	00000000
IP address (of net 2)			
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
Metric (hops to net 2)			
⋮			
Address family identifier (of net 25)		00000000	00000000
IP address (of net 25)			
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
Metric (hops to net 25)			

- If the **command** field contains value 1, the message is a RIP request
  - This way, a Router is requested to transmit its routing table
- If the **command** field contains value 2, the message is a RIP response
  - This way, a Router sends its routing table
- For IP networks, the field **address family identifier** contains value 2

RFC 1058 (1988)

Structure of RIPv2 Messages (*Advertisements*)



- The field **route tag** specifies, whether a route is an internal or external route

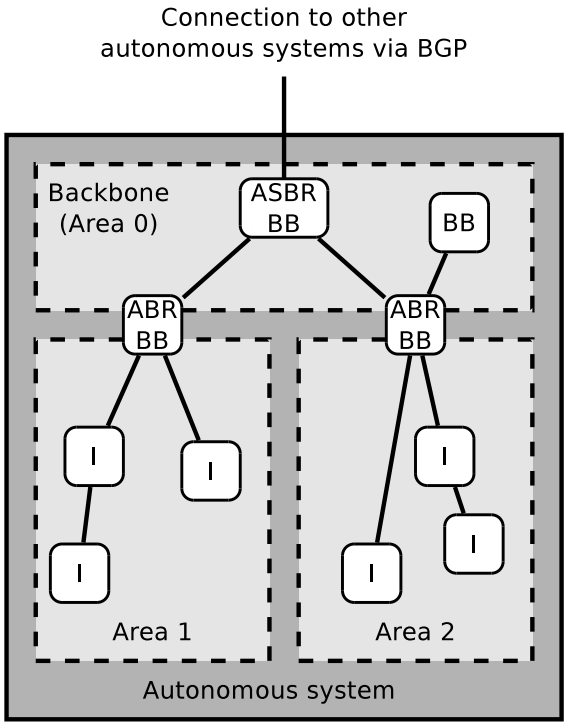
RFC 2453 (1998)

Link State Routing Protocols

- Implement the *Dijkstra algorithm* (**Shortest Path First**)
  - Allows the calculation of the shortest path (route) between a starting node and all other nodes in a weighted graph
- Example: **Open Shortest Path First** (OSPF)
  - Allows routing inside autonomous systems (**Intra-AS routing**)
  - OSPF messages are transmitted directly, without a Transport Layer protocol, in the payload section of IPv4 packets
    - In the header of the IPv4 packet, the field **protocol ID** contains value 89
  - The functioning of OSPF is complicated compared with RIP
    - RFC 2328 contains a detailed description of the protocol

Constructing Routing Hierarchies with OSPF

- Big difference compared to RIP:
  - With OSPF, routing hierarchies can be created
- For this, autonomous systems are spit into **areas**
  - Each area consists of a group of Routers
  - Each area is invisible for other areas of the autonomous system
  - Each Router can be assigned to multiple areas
- An advantage, which results from routing hierarchies:
  - Better scalability



ASBR = Autonomous System Boundary Router
ABR = Area Border Router
BB = Backbone Router
I = Internal Router

Helpful OSPF resources

Ethernet, *Jörg Rech*, Heise (2008)
Computernetzwerke, *James F. Kurose, Keith W. Ross*, Pearson (2008)
TCP/IP, *Gerhard Lienemann, Dirk Larisch*, Heise (2011)

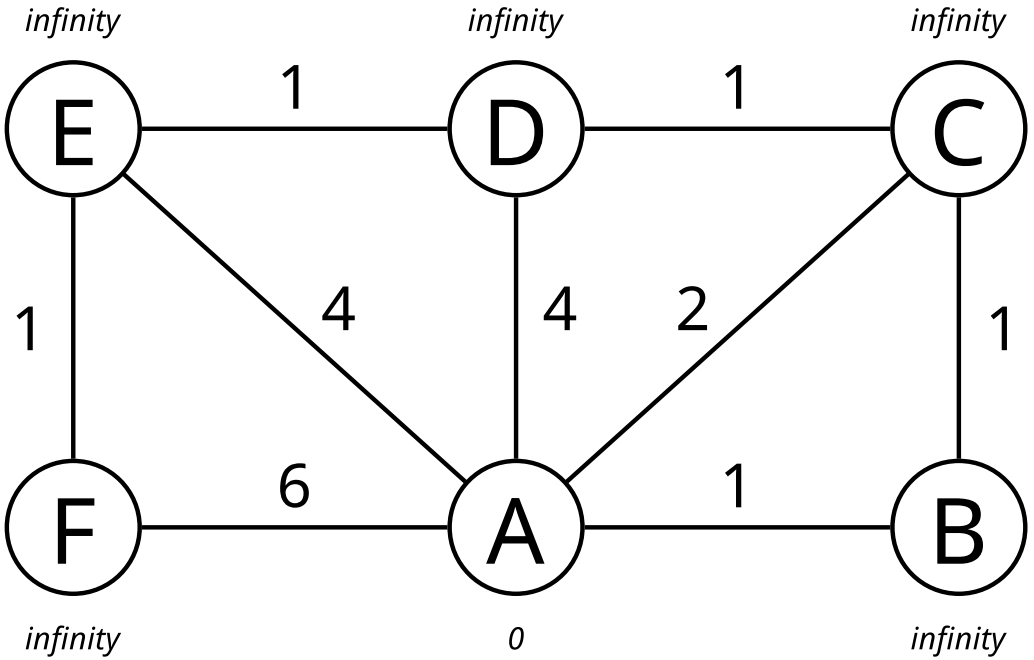
OSPF is far more complex compared with RIP and it will not be discussed in this course in detail

Dijkstra Algorithm

- **Calculates the shortest path between a start node (initial node) and all other nodes in an edge-weighted graph**
  - The algorithm can not be used on graphs with negative edge weights
- Steps:
  1. Assign to every node the properties **distance** and **predecessor**
     - Set the distance to 0 for the initial node and to ∞ for all other nodes
  2. As long as there are unvisited nodes, select the node with the minimal distance
     - Mark the node as visited
     - Compute for all unvisited neighbors, the sum of the edge weights via the current node
     - If this value is smaller than the stored distance for a node, update the distance and set the current node as predecessor

If only the path to a specific node needs to be determined, the algorithm can stop during step 2, if the requested node is the active one

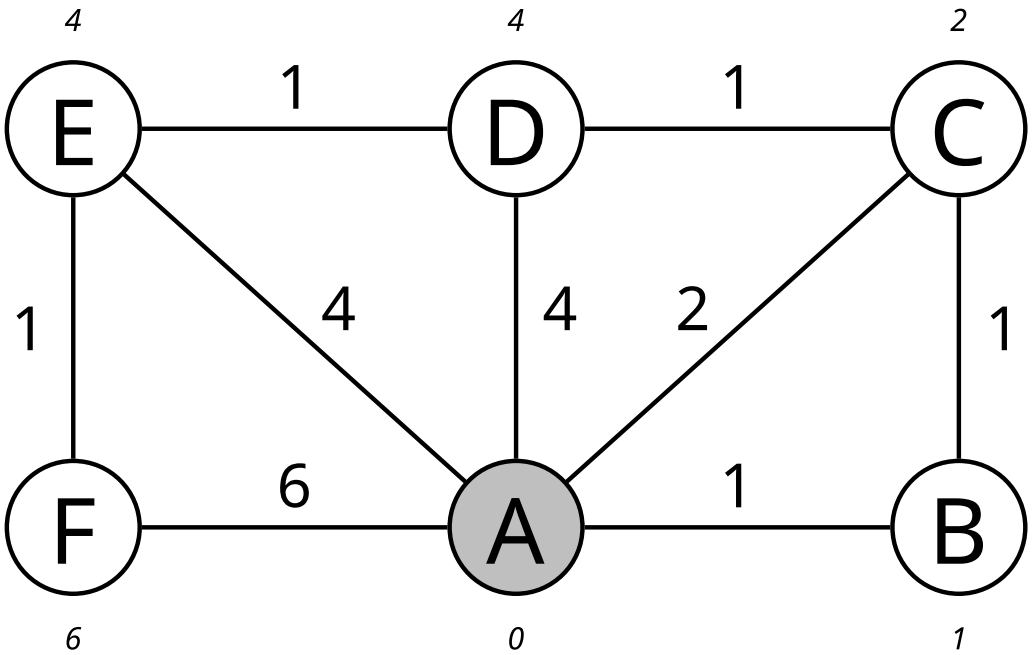
Dijkstra Algorithm – Example (1/7)



Distance values	
$d_A = 0$	
$d_B = \infty$	
$d_C = \infty$	
$d_D = \infty$	
$d_E = \infty$	
$d_F = \infty$	

- Step 1: Initialize with 0 and ∞
  - A is the starting node
  - A has the minimum distance
- Nodes visited $= \{\}$
- Shortest paths $= \{\}$

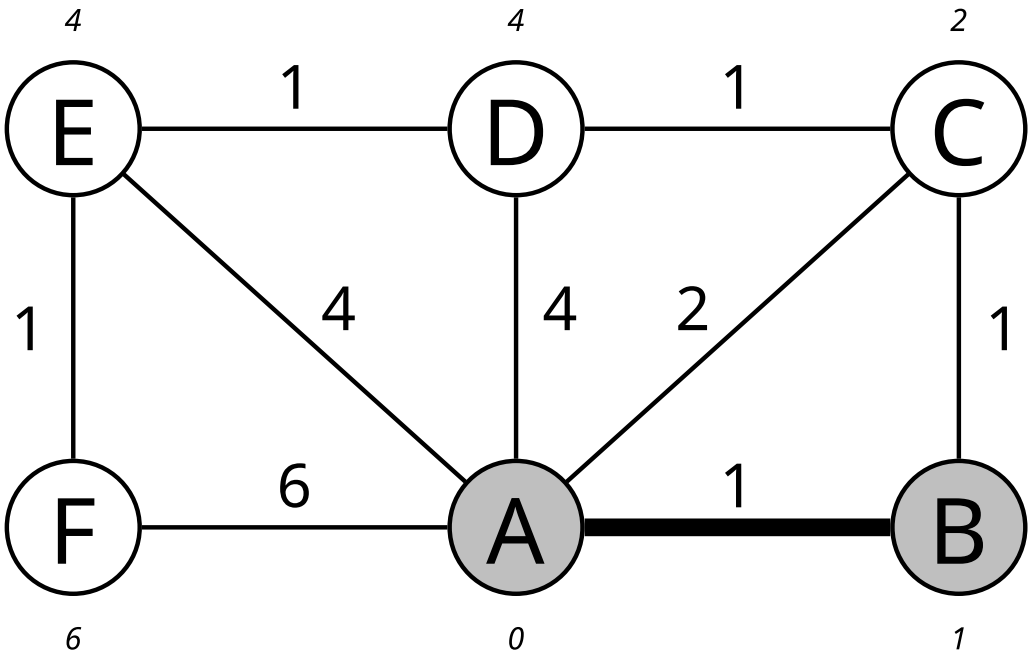
Dijkstra Algorithm – Example (2/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	⟵ minimum distance
$d_C = 2$	
$d_D = 4$	
$d_E = 4$	
$d_F = 6$	

- Step 2: Calculate the sum of the edge weights
  - B has the minimum distance
- Nodes visited $= \{A\}$
- Shortest paths $= \{A\}$

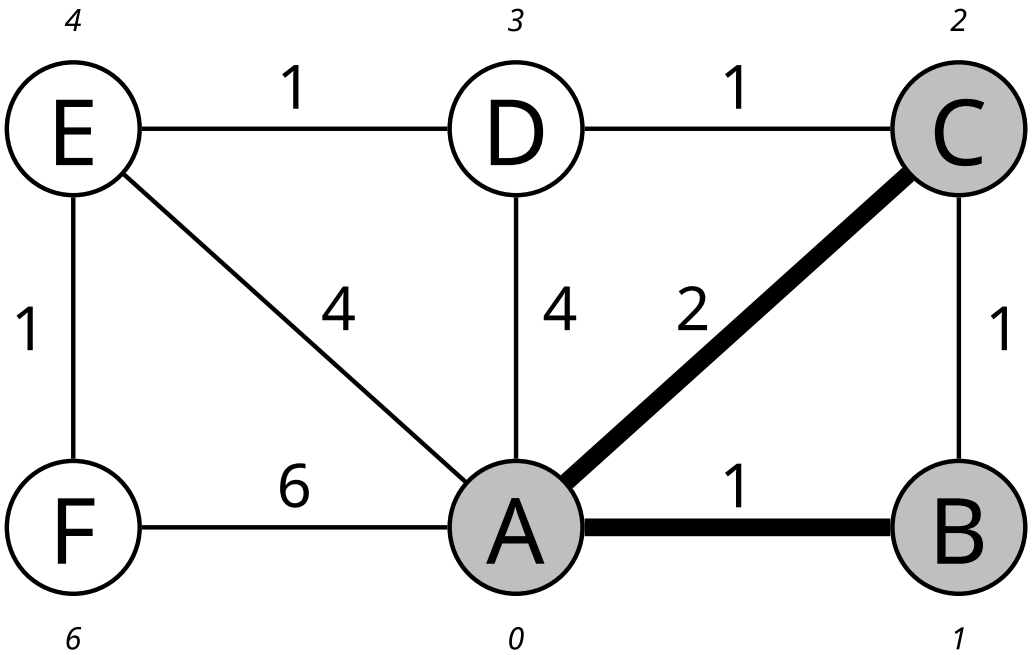
Dijkstra Algorithm – Example (3/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	visited
$d_C = 2$	⟵ minimum distance
$d_D = 4$	
$d_E = 4$	
$d_F = 6$	

- Step 3: Visit node B
  - No change to C
  - C has the minimum distance
- Nodes visited = {A, B}
- Shortest paths = {A, A⟶B}

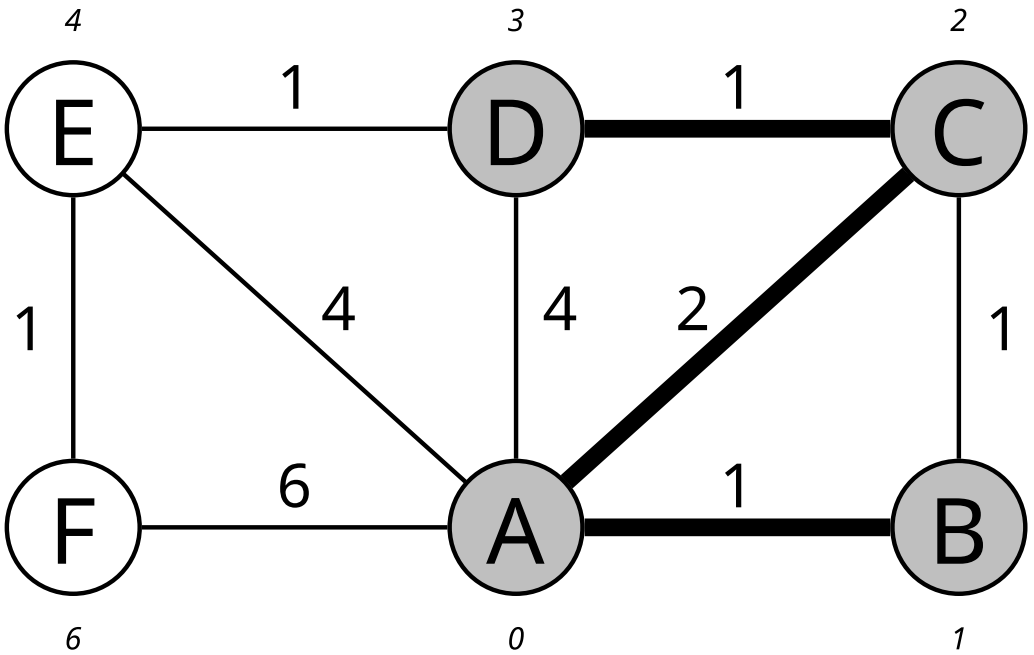
Dijkstra Algorithm – Example (4/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	visited
$d_C = 2$	visited
$d_D = \mathbf{3}$	$\longleftarrow$ minimum distance
$d_E = 4$	
$d_F = 6$	

- Step 4: Visit node C
  - No change to B
  - Change to D (path via C is shorter than the direct path)
  - D has the minimum distance
- Nodes visited = {A, B, C}
- Shortest paths = {A, A$\longrightarrow$B, A$\longrightarrow$C}

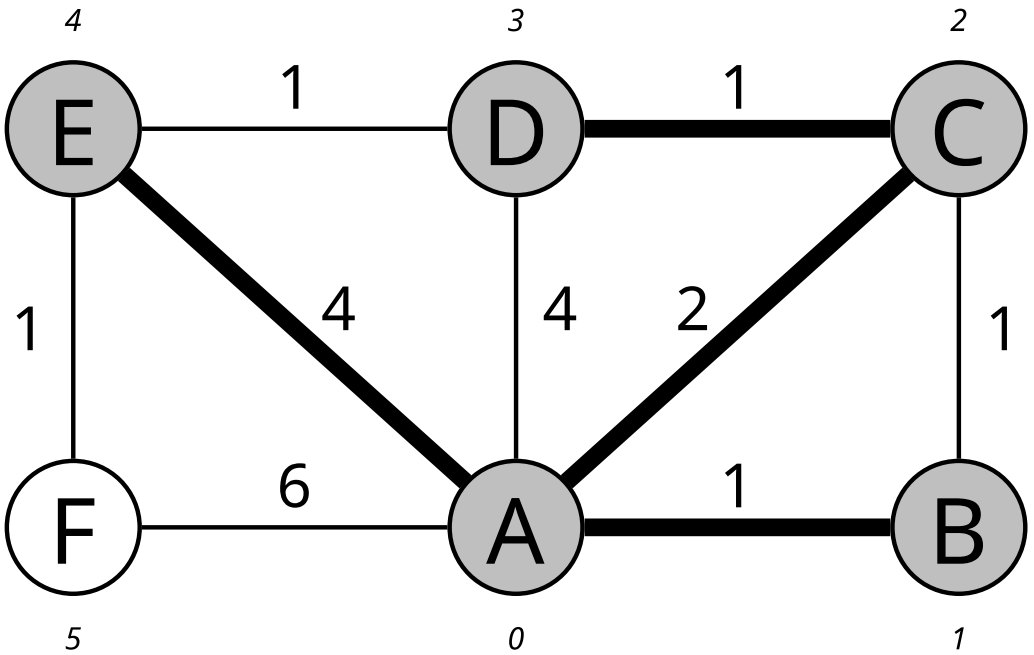
Dijkstra Algorithm – Example (5/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	visited
$d_C = 2$	visited
$d_D = 3$	visited
$d_E = 4$	$\longleftarrow$ minimum distance
$d_F = 6$	

- Step 5: Visit node D
  - No change to C
  - No change to E
  - E has the minimum distance
- Nodes visited = {A, B, C, D}
- Shortest paths = {A, A$\longrightarrow$B, A$\longrightarrow$C, C$\longrightarrow$D}

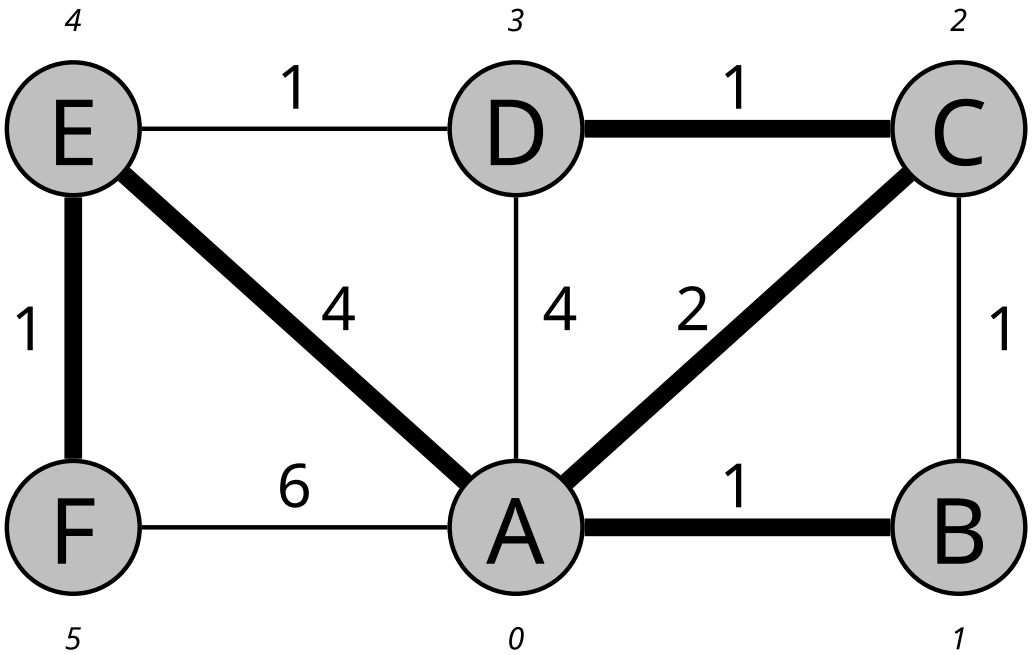
Dijkstra Algorithm – Example (6/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	visited
$d_C = 2$	visited
$d_D = 3$	visited
$d_E = 4$	visited
$d_F = \mathbf{5}$	$\longleftarrow$ minimum distance

- Step 6: Visit node E
  - No change to D
  - Change to F (path via E is shorter than the direct path)
  - F has the minimum distance
- Nodes visited = {A, B, C, D, E}
- Shortest paths = {A, A$\longrightarrow$B, A$\longrightarrow$C, C$\longrightarrow$D, A$\longrightarrow$E}

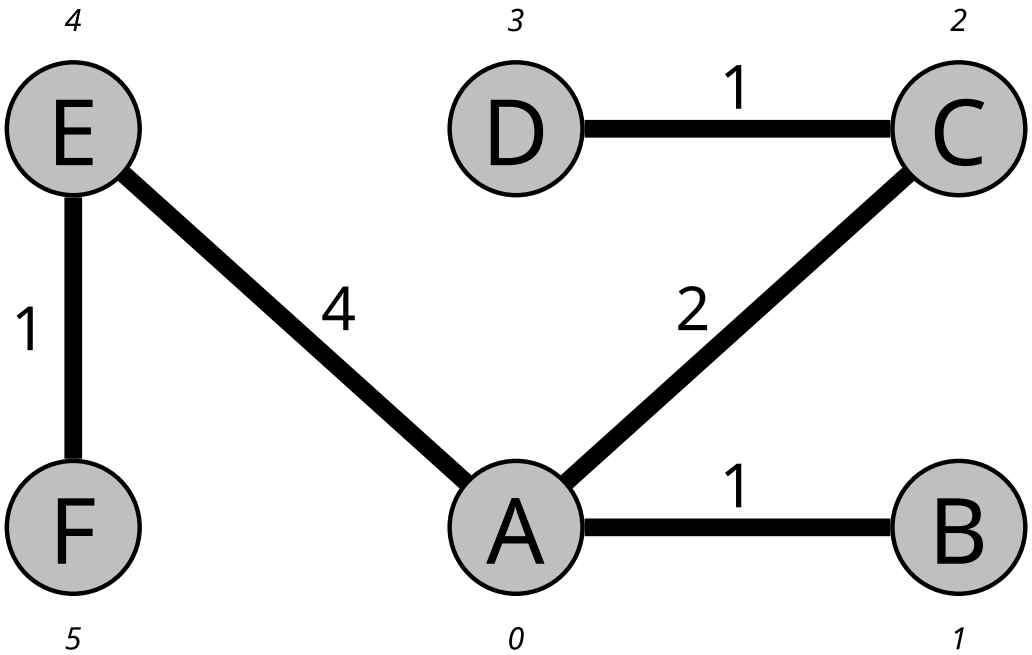
Dijkstra Algorithm – Example (7/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	visited
$d_C = 2$	visited
$d_D = 3$	visited
$d_E = 4$	visited
$d_F = 5$	visited

- Step 7: Visit node F
  - No change to E
- Nodes visited = {A, B, C, D, E, F}
- Shortest paths = {A, A$\longrightarrow$B, A$\longrightarrow$C, C$\longrightarrow$D, A$\longrightarrow$E, E$\longrightarrow$F}

Dijkstra Algorithm – Example (Result)



- Result: Shortest path spanning tree

Distance-Vector Routing Protocol vs. Link-State Routing Protocol

- Distance-vector routing protocol (*Bellman-Ford*)
  - Each Router communicates only with its **direct** neighbors
    - Advantage: The network is not flooded
      $\Longrightarrow$ protocol causes little overhead
    - Drawback: Long convergence time because updates *propagate* slowly
  - **No Router has knowledge about the complete network's topology**
  - The path cost (**metric**) depend only on the number of Routers (**hops**), which need to be passed on the way to the destination network
- Link-state routing protocol (*Dijkstra*)
  - **All** Routers communicate with each other
    - Advantage: Short convergence time
    - Drawback: The network is flooded
      $\Longrightarrow$ protocol causes strong overhead
  - Each Router maintains a **complex database of topology information**
  - With Areas, **routing hierarchies** are realized
    - This improves scalability
  - Metric: Network segments have different **path cost**

Diagnosis and Error Messages via ICMP

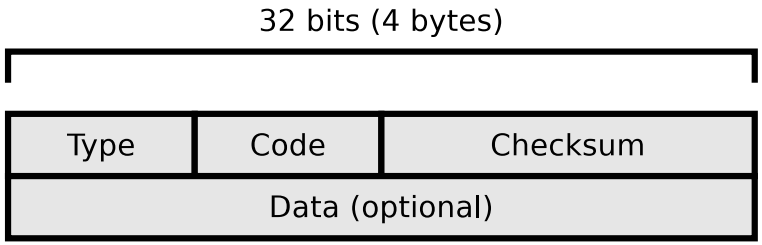
- The **Internet Control Message Protocol** (ICMP) is used for the exchange of...
  - diagnostic messages
  - control messages
  - error messages
- ICMP is a component (*sub-protocol*) of IPv4
  - but it is treated as a separate protocol

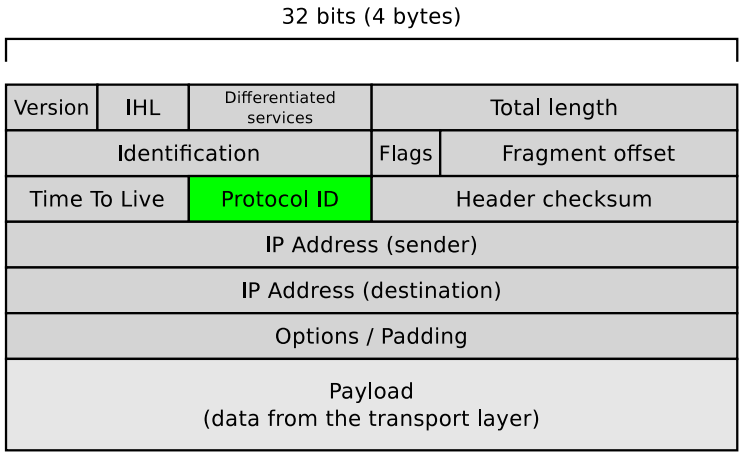
For IPv6, a similar protocol called ICMPv6 exists

- All Routers and terminal devices can handle ICMP
- Typical situations where ICMP is used:
  - A Router discards an IP packet, because it does not know how to forward it
  - Only a single fragment of an IP packet arrives at the destination
  - The destination of an IP packet cannot be reached, because the Time To Live (TTL) has expired

ICMP

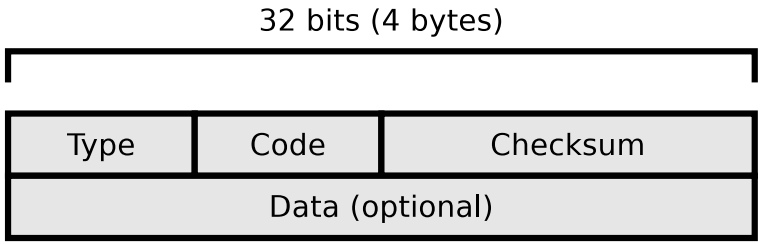
- Example of an application, which sends ICMP packets: `ping`
- ICMP specifies different sorts of messages, which can be sent by a Router as response to provide diagnostic information





- ICMP messages are transmitted as payload inside IPv4 packets
  - In the header of the IPv4 packet the field **protocol ID** contains value 1
  - For ICMPv6 the protocol ID is 58
- If an ICMP packet cannot be delivered, no further action is done

ICMP Messages

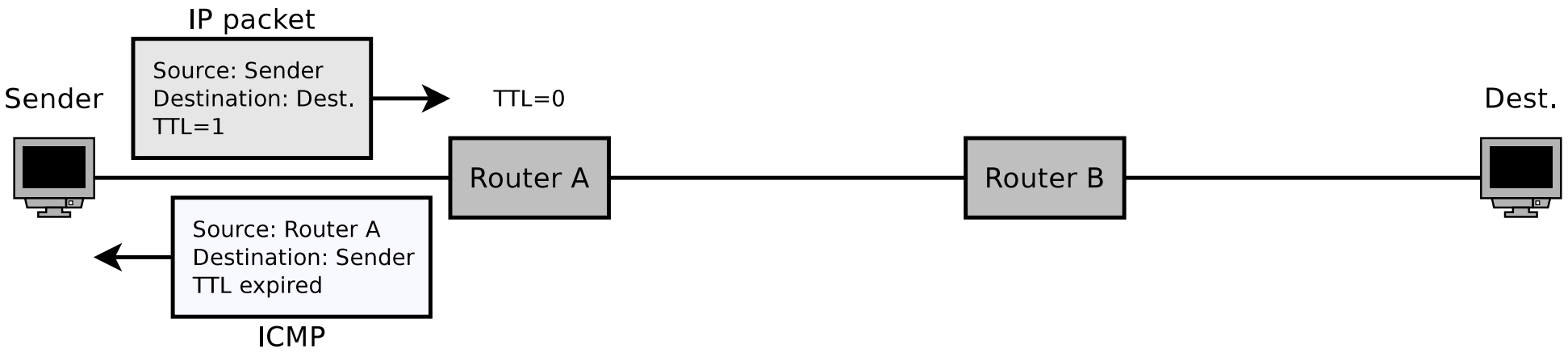


- The field **Type** in the ICMP header specifies the ICMP message type
- The field **Code** specifies the subtype of the message type
- The table contains some type-code combinations of ICMP messages

Type	Name of type	Code	Description
0	Echo reply	0	Echo reply (reply for `ping`)
3	Destination unreachable	0	Destination network unreachable
		1	Destination host unreachable
		2	Destination protocol unreachable
		3	Destination port unreachable
		4	Fragmentation required, but forbidden by the IP packet's flags
		13	Firewall at destination site rejects the IP packet
5	Redirect	0	Router informs sender about a better route (IP of first hop) to destin. network
		1	Router informs sender about a better route (IP of first hop) to destination host
8	Echo Request	0	Echo request (`ping`)
11	Time Exceeded	0	TTL (Time To Live) expired (exceeded in transit)
		1	Fragment reassembly time exceeded

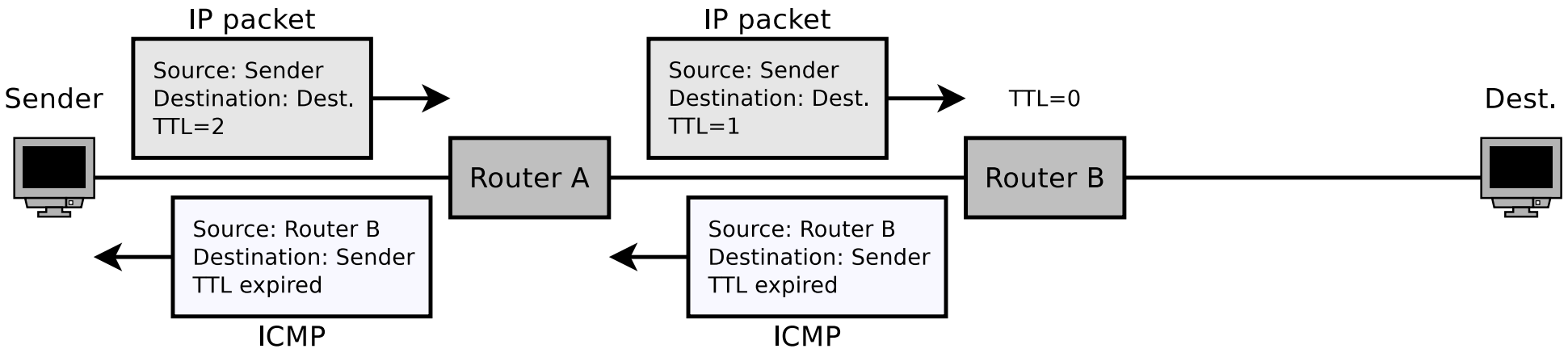
The ICMP protocol includes many more type-code combinations (see RFC 792), but most were seldom or never used in practice and are considered deprecated (see RFC 6633 and RFC 6918)

Example of using ICMP: `traceroute` (1/3)



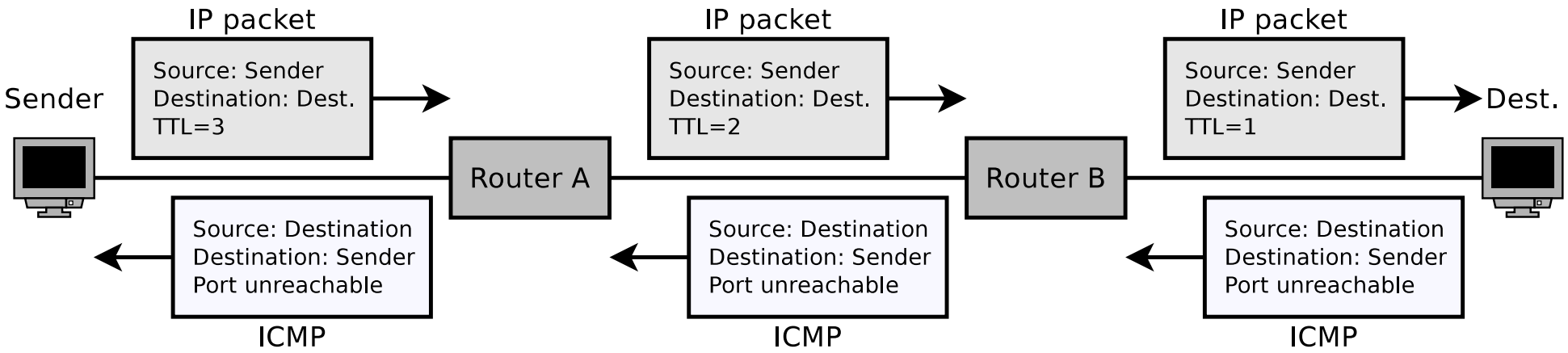
- Another application example of ICMP is the tool `traceroute`
- `traceroute` determines, which Routers are used to forward packets to the destination site
- The sender transmits an IP packet to the destination with `TTL=1`
- Router A receives the IP packet, sets `TTL=0`, discards the IP packet and transmits an ICMP message of message type 11 and code 0 to the sender

Example of using ICMP: `traceroute` (2/3)



- Next, the sender transmits an IP packet to the destination with `TTL=2`
- The IP packet is forwarded by Router A and thereby the value of `TTL` is decremented
- Router B receives the IP packet, sets `TTL=0`, discards the IP packet and transmits an ICMP message of message type 11 and code 0 to the sender

Example of using ICMP: traceroute (3/3)

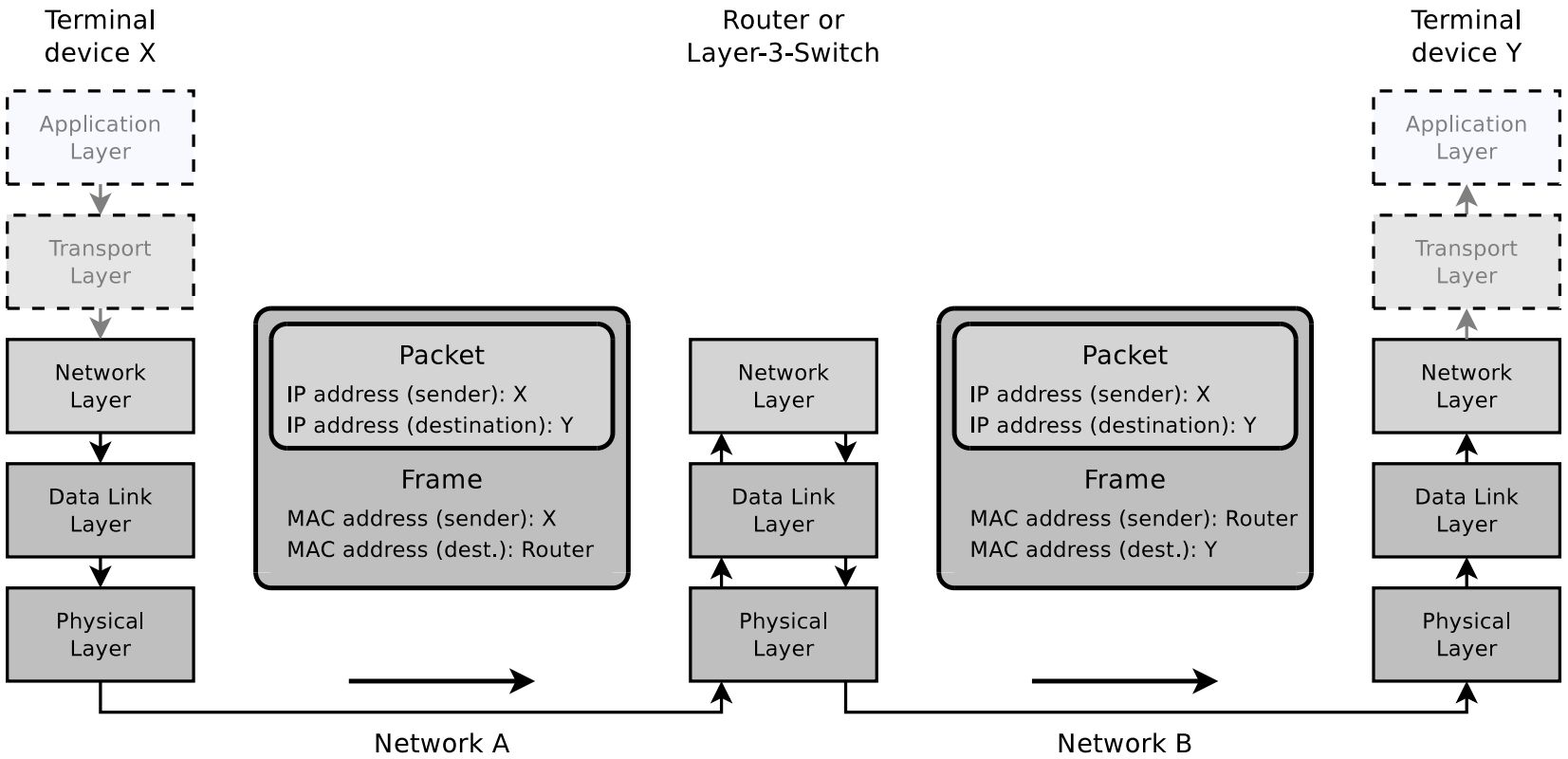


- Once the value of TTL is big enough that the destination site can be reached, the receiver transmits an ICMP message of message type 3 and code 3 to the sender
- This way, the path from sender to receiver can be traced via ICMP

```
$ traceroute -q 1 wikipedia.de
traceroute to wikipedia.de (134.119.24.29), 30 hops max, 60 byte packets
 1  fritz.box (10.0.0.1) 1.834 ms
 2  p3e9bf6a1.dip0.t-ipconnect.de (62.155.246.161) 8.975 ms
 3  217.5.109.50 (217.5.109.50) 9.804 ms
 4  ae0.cr-polaris.fra1.bb.godaddy.com (80.157.204.146) 9.095 ms
 5  ae0.fra10-cr-antares.bb.gdinf.net (87.230.115.1) 11.711 ms
 6  ae2.cgn1-cr-nashira.bb.gdinf.net (87.230.114.4) 13.878 ms
 7  ae0.100.sr-jake.cgn1.dcnet-emea.godaddy.com (87.230.114.222) 13.551 ms
 8  wikipedia.de (134.119.24.29) 15.150 ms
```

Internetworking (1/6)

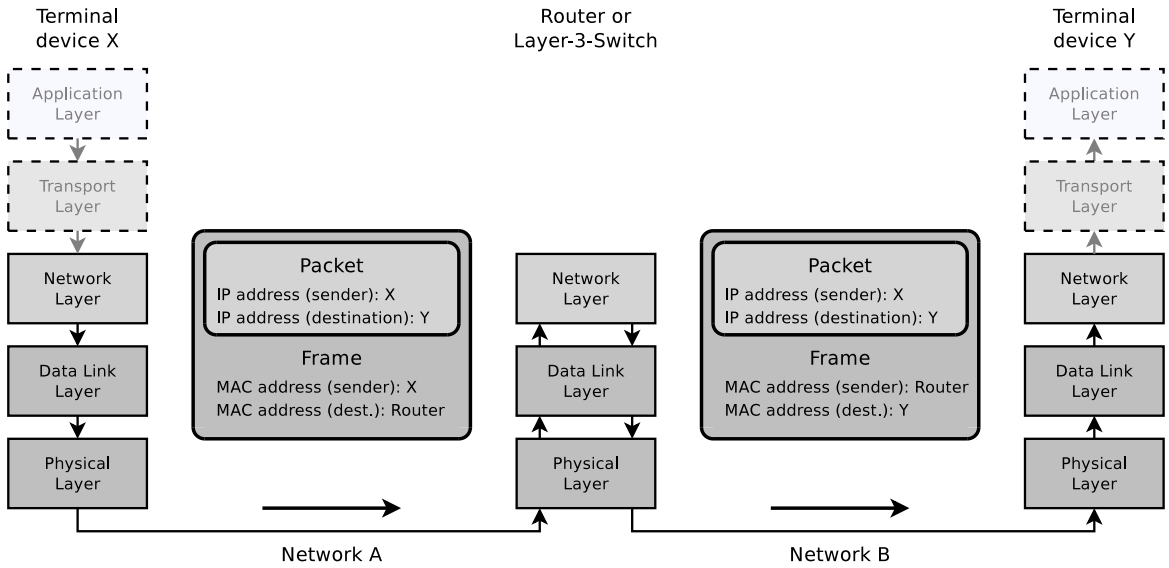
- **Internetworking** = communication between network devices with the protocols of the Data Link Layer and Network Layer through networks, which may base on different networking technologies
- Possible scenario for internetworking



Internetworking (2/6)

In this scenario, all communication partners have public IP address!

- X wants to transmit an IP packet to Y
  - To do this, X needs to know the **logical address** (IP address) of Y



You already know (from slide set 4)...

For the forwarding on the Data Link Layer, the **physical address** (MAC address) is required too

- X calculates the subnet IDs ($\Longrightarrow$ slide set 7)
  - subnet mask$_X$ `AND` IP address$_X$ = subnet ID of the own network
  - subnet mask$_X$ `AND` IP address$_Y$ = subnet ID of the network where Y is

Internetworking (3/6)

- Identical subnet IDs $\Longrightarrow$ X and Y are in the same logical subnet

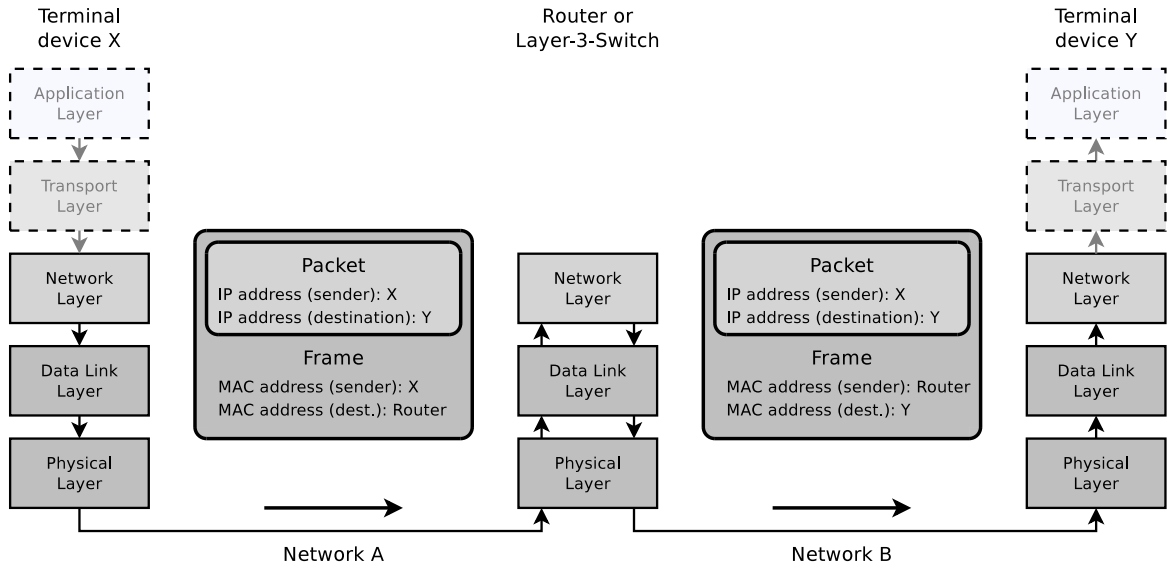
You already know (from slide set 7)...

A logical subnet covers at least one physical network and can only be connected with one interface of a Router

- Different subnet IDs $\Longrightarrow$ X and Y are in different logical subnets

You already know (from slide set 6)...

If 2 communication partners are in the same logical and physical network, the sender can discover the MAC address of the receiver via address resolution with ARP



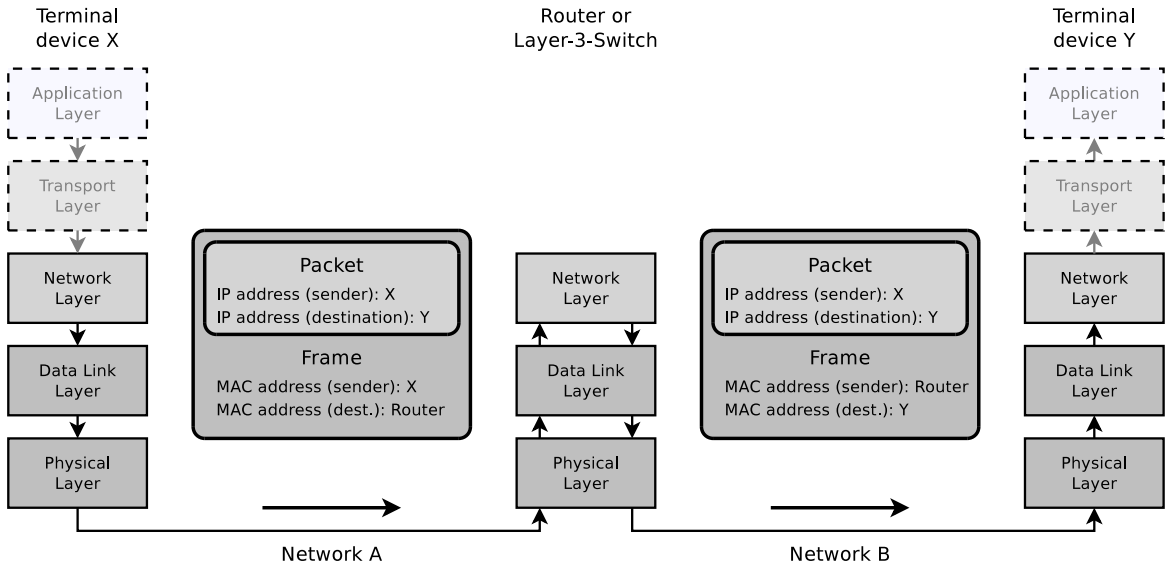
- **In this scenario we have communication across logical and physical network boundaries**

Internetworking (4/6)

You already know (from slide set 6)...

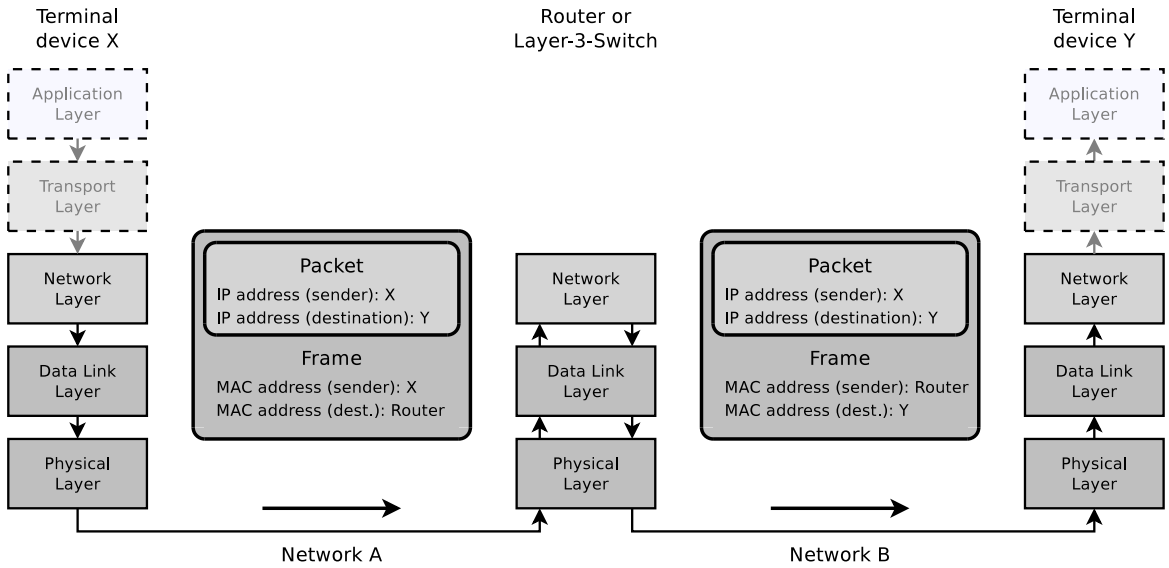
- ARP is only suited for the resolution of MAC addresses in the local physical network
- Reason: ARP requests are sent in frames of the Data Link Layer
- The destination address field contains the broadcast address
- Bridges and Switches do not forward such frames
  $\Longrightarrow$ Therefore, with ARP, cross-network address resolution is impossible

- The frame contains as payload the IP packet for Y with the IP address of X as sender address and the IP address of Y as destination address



Internetworking (5/6)

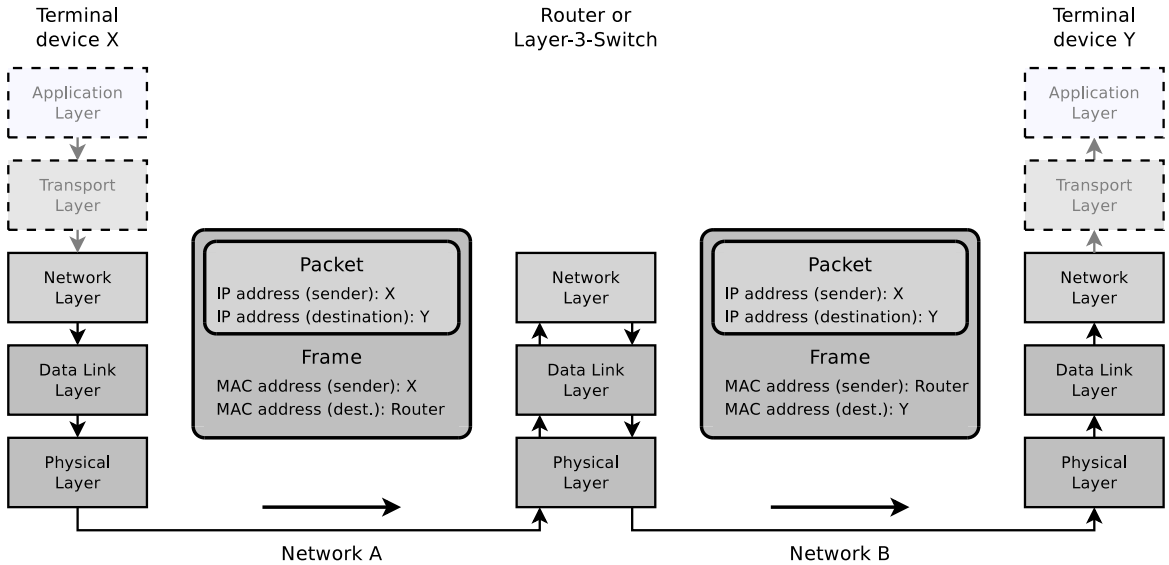
- The Router receives the IP packet
  - It finds out with its local routing table the correct interface for forwarding the packet
    - The local routing table contains all logical networks, the Router knows
- The Router is connected via one of its ports with the physical network, to which Y is connected
- The Router finds out the MAC address of Y via address resolution with ARP



- The Router packs the IP packet into a frame
  - The sender address field contains the MAC address of the Router
  - The destination address field contains the MAC address of Y

Internetworking (6/6)

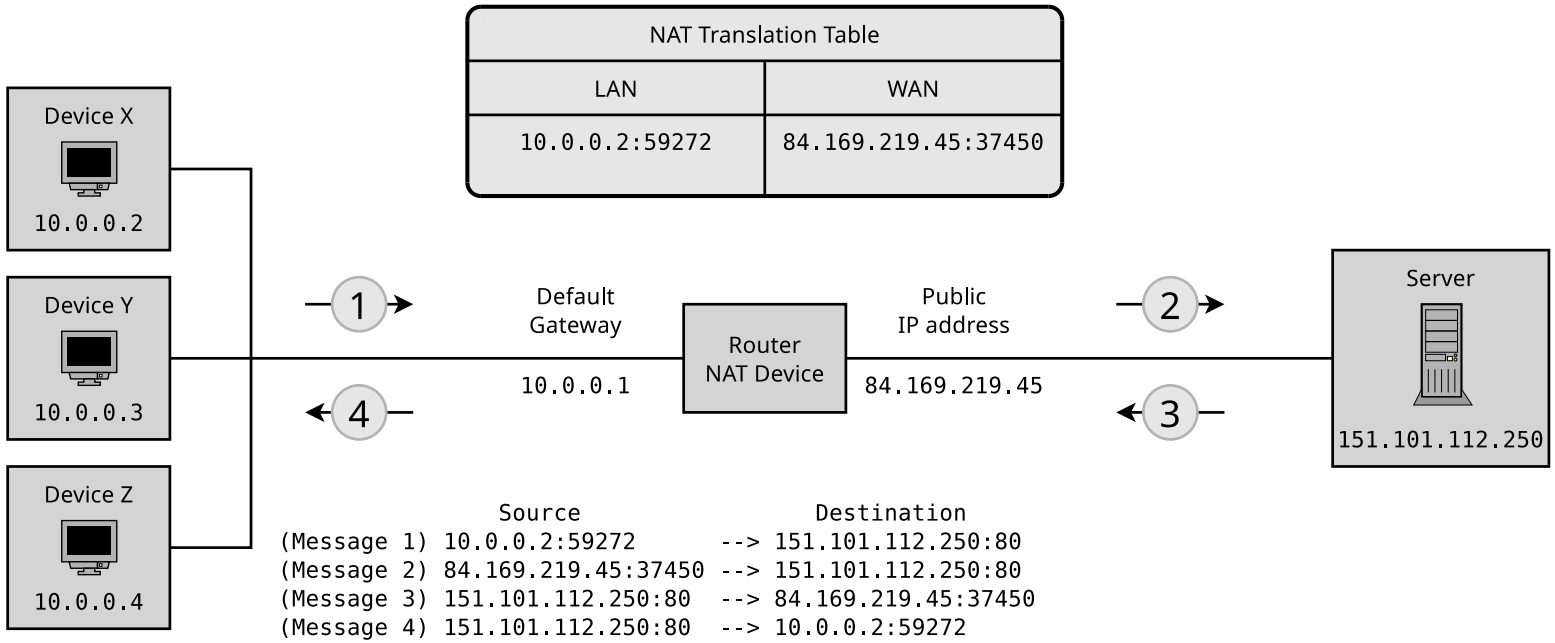
- Maybe the maximum packet length (*Maximum Transmission Unit*) of network B is smaller than the one of network A
  - In this case it may be required, depending on the size of the forwarded IP packet, that the Router fragments ($\Longrightarrow$ see slide set 7) the received packet into multiple smaller packets
- The IP addresses of the sender (X) and the receiver (Y) in the IP packet are not modified during the transmission



Network Address Translation (1/5)

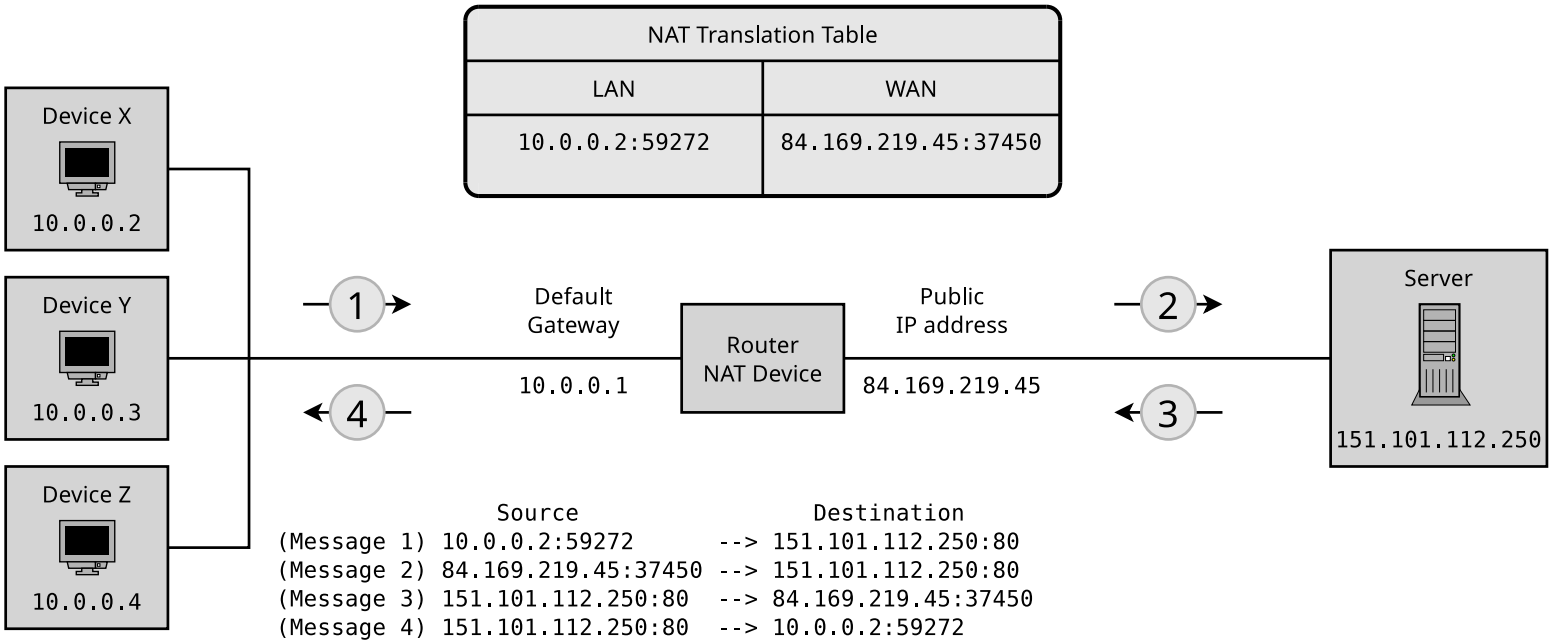
- Problem: Few households, businesses and educational/research institutions have enough public IPv4 addresses to equip all their network devices with own IPs
  - Therefore, LANs usually use a private IPv4 address space (see slide set 7)
  - How can network devices in private networks communicate with network devices that have globally accessible addresses?
  - Solution: **Network Address Translation** (NAT)
    - The local Router presents itself as the source of those IP packets that it forwards from the directly connected private network to the Internet
    - In addition, it forwards incoming replies to the participants in the directly connected private networks

Network Address Translation (2/5)



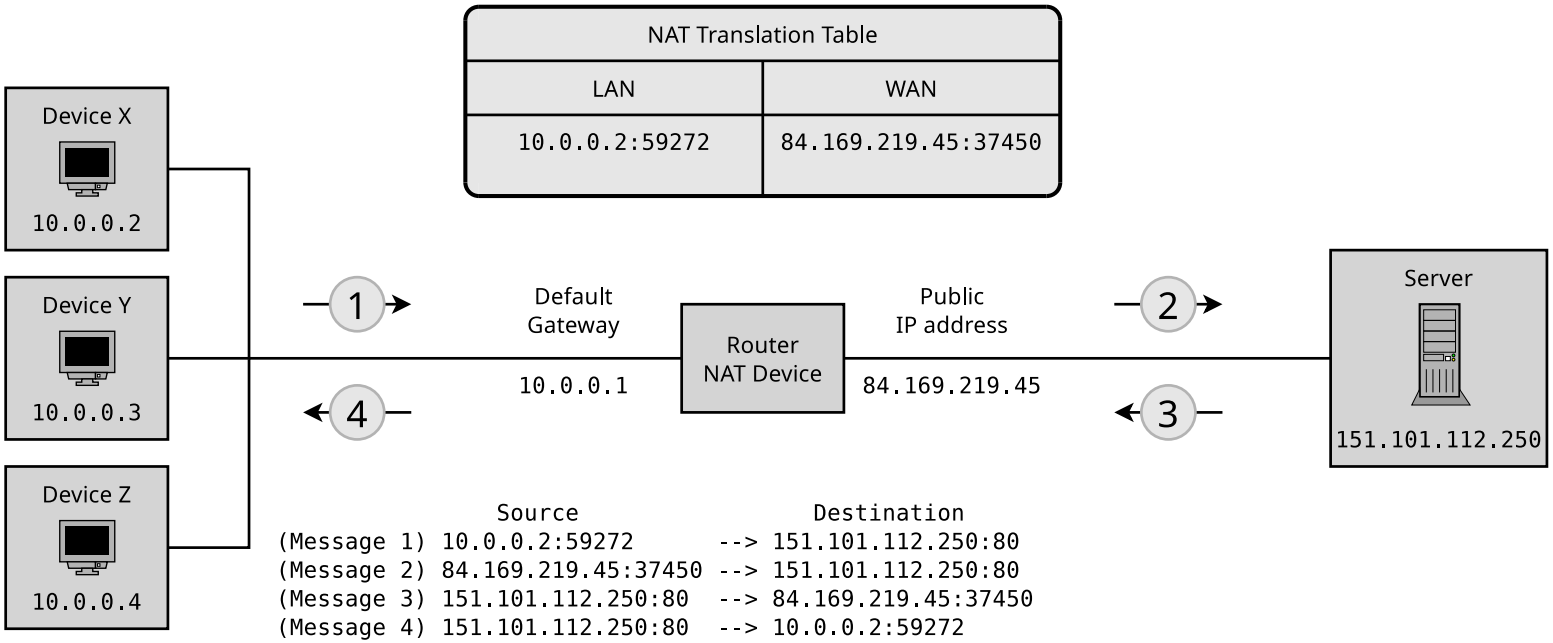
- Clients X, Y, and Z are inside a network with a private IP address range
- Only the Router has a globally accessible IP address
  - It does appear to the outside world as just a network device with a single public IP address and not as a Router

Network Address Translation (3/5)



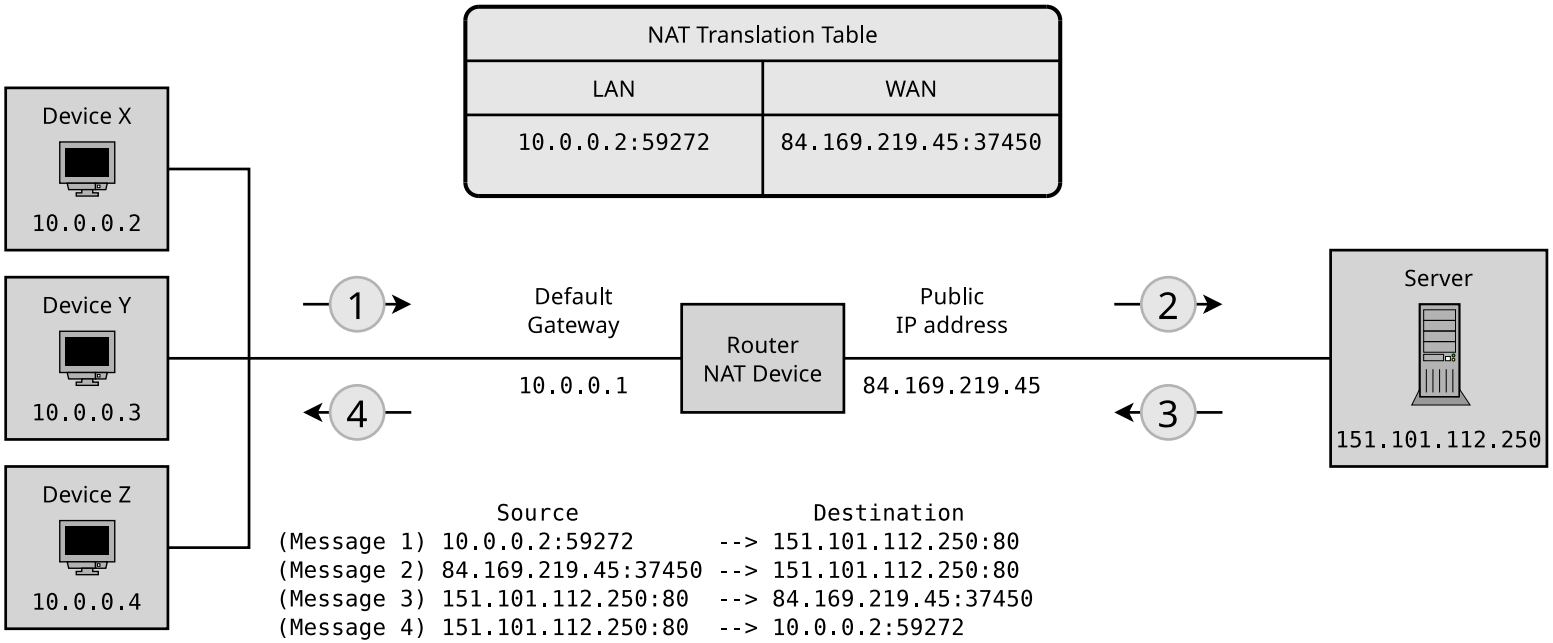
- Client X sends a request for a web page
  - The request (message 1) contains the IP address and port number of X as source addresses and the IP address and port number of the server as destination addresses
- The Router replaces the IP and port number of the client with its own addresses inside the forwarded request (message 2)

Network Address Translation (4/5)



- The Router stores the mappings between the Router ports and the corresponding network devices inside its local **NAT translation table**
- The destination address inside the reply of the server (message 3) is the IP of the Router
  - The Router replaces the address information according to the table and forwards the reply to X (message 4)

Network Address Translation (5/5)



- With IPv6, NAT is unnecessary because the address space is large enough to allocate globally accessible addresses to all network devices
  - Whether this is advisable for reasons of security, is controversial
    - NAT improves network security because it hides the topology of the local network from the outside world
- NAT with IPv6: **IPv6-to-IPv6 Netw. Address Translation** (NAT66)