Computer Networks

Network Layer - Routing

Prof. Dr. Oliver Hahm

Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering
`oliver.hahm@fb2.fra-uas.de`
`https://teaching.dahahm.de`

December 14, 2021

Organizational (1/3)

- This week is a trial run for a hybrid lecture
- Next week we will have a guest lecturer:
  David Wagner, Researcher at DE-CIX
  This lecture will be online only
- What will happen afterwards (hybrid or online), remains to be seen ...
- The current exercise sheet is due to January 25, 2022
- Next week on February 18, 2022 at 14:15 CET I will give a talk on RIOT as part of the lecture on Operating Sytems via Zoom

  $\longrightarrow$ `http://www.christianbaun.de/BTS2122/index_en.html`

Organizational (2/3)

- The exam will take place at Messe Frankfurt, Hall 11 on February 21, 2022
- The time will be announced later.
- You will be allowed to bring a cheat sheet and a calculator
- All necessary formulas and concrete numbers will be given in the exam
- The exam will consist of similar tasks as in the exercise sheets
- Also take a look at the previous exams by Prof. Baun:
  `http://www.christianbaun.de/Netzwerke2021/index_en.html#Klausuren`
- A dedicated practise exam will follow
- The last week of the lecture is reserved for exam preparation

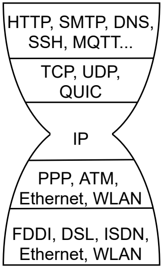
Organizational (3/3)

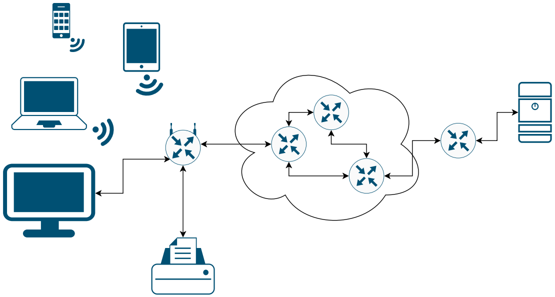
- The registration deadline is next week: January 17, 2022
  See examination schedules:
  `https://his-www.dv.fh-frankfurt.de/qisserver/rds?state=user&type=0`
- Please register - the registration deadlines must be strictly observed!
- The announced reporting, admission, and withdrawal deadlines are binding. All late registrations will be rejected and admission to the examination is excluded. Exceptions will only be granted for reasons beyond the student's control.
- A few examinations may be conducted online in exceptional cases.
- Only register for the exam if you seriously intend to participate in it.
- Changes can occur at short notice. We kindly ask you to pay attention to the updates of the exam schedules and the announcements on a regular basis.

The Narrow Waist of the Internet

Tasks of the Network Layer:

- Inter-Networking
- Providing logical addresses
- Forwarding packets
- Finding the best path → Routing
- Devices: Router
- Protocols: IPv4 (RFC 791) and IPv6 (RFC 2460)


HTTP, SMTP, DNS, SSH, MQTT...
TCP, UDP, QUIC
IP
PPP, ATM, Ethernet, WLAN
FDDI, DSL, ISDN, Ethernet, WLAN



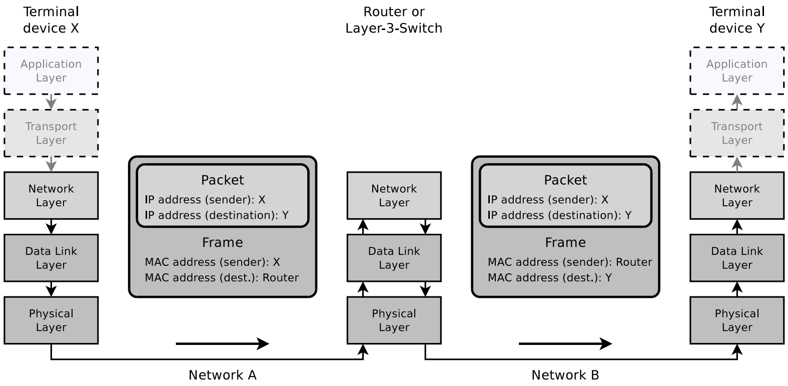
Agenda

Agenda

- Inter-Networking
- Routing Schemes
- Distance Vector Routing
- Link State Routing
- More Routing Protocols

Inter-Networking

- In the Internet we have multiple networks using different layer 1 and 2 technologies
- The network layer is now tasked to interconnect these networks
- Possible scenario for Inter-Networking:

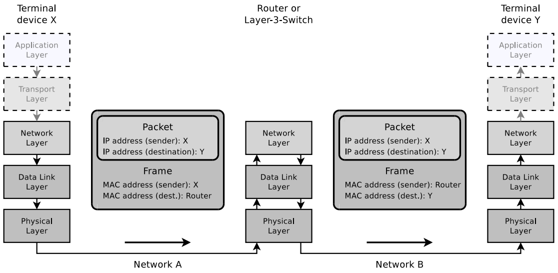


Example Scenario: (1/5)

Assumption

In this scenario, all communication partners have public IP addresses

- 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...

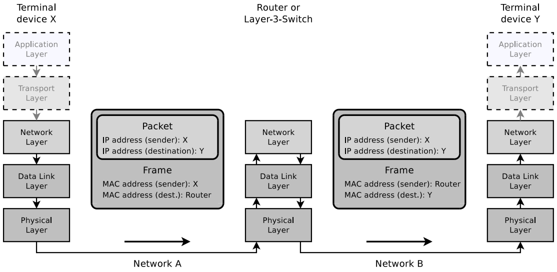
For the forwarding on the Data Link Layer, the **physical address** (MAC address) is required, too

Example Scenario (2/5)

- X calculates the subnet ID of its own network and the subnet ID of Y
- Different subnet IDs $\Longrightarrow$ X and Y are in different logical subnets

You already know . . .

X performs an `AND` operation of its own subnet mask with its own IP address and with the IP address of Y ($\Longrightarrow$ slide set 7)



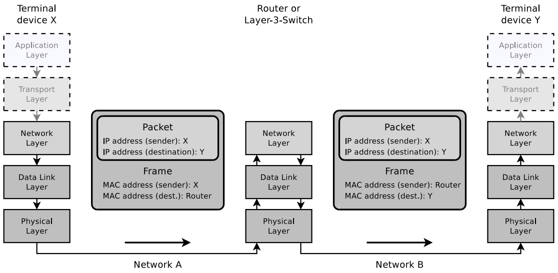
- $\longrightarrow$ In this scenario we have communication across logical and physical network boundaries

Example Scenario (3/5)

You already know ...

- 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

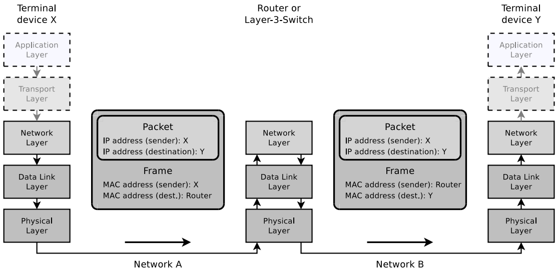
- The IP packet for Y is carried as payload inside the frame towards the router
- It has the IP address of X as source address and the IP address of Y as destination address



Example Scenario (4/5)

- The router receives the IP packet
  - It checks its local forwarding table to find the correct interface to forward the packet
  - The forwarding table contains information about all logical networks the router is connected to

- The router is connected via one of its NICs with the physical network, to which Y is connected
- The router finds out the MAC address of Y via address resolution with ARP

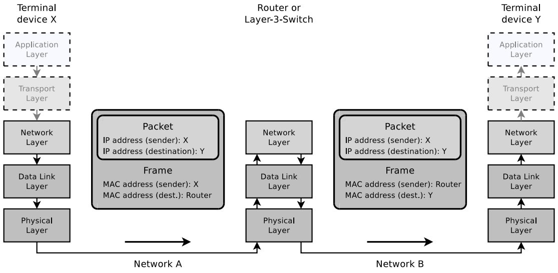


- The Router encapsulates the IP packet into a frame
  - The sender address field contains the MAC address of the Router
  - The destination address field contains the MAC address of Y

Example Scenario (5/5)

- Maybe the Maximum Transmission Unit of network B is smaller than of network A
- $\Longrightarrow$ depending on the size of the forwarded IP packet, that the router fragments the packet

- The IP addresses of the sender (X) and the receiver (Y) in the IP packet are not modified during the transmission



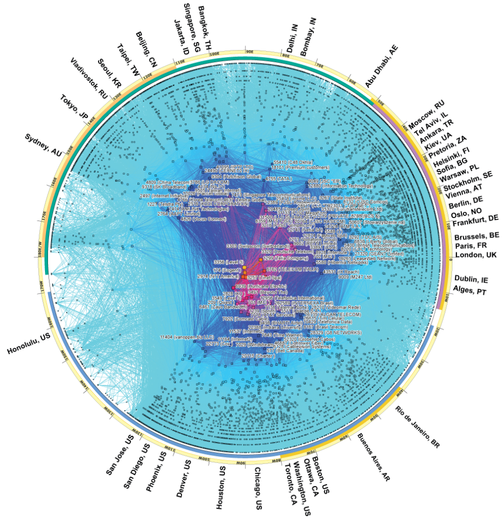
Autonomous Systems

- The Internet consists of a large number of Autonomous Systems (AS)
- The AS are interconnected via routers that are called gateways
- Each AS consists of a group of logical networks, which. . .
  - are operated and managed by the same organization (e.g. an Internet Service Provider, a corporation or university)
  - use the same routing protocol, which is called the Interior Gateway Protocol (IGP) or Intra-Domain Routing Protocol
- The routing between the various AS uses an Exterior Gateway Protocol (EGP) or Inter-Domain Routing Protocol

Autonomous Systems Number (ASN)

- Each AS has a unique Autonomous System Number (ASN)
- The ASNs are assigned by the IANA in blocks to the Regional Internet Registries (RIRs)
- The RIRs assign ASNs to entities inside their areas
  - For Europe: RIPE NCC: `http://www.ripe.net`

Map of the Internet in 2017



Source: https://www.caida.org/projects/cartography/as-core/2017/

- Shows the IPv4 AS of the Internet
- Various variations of Internet maps are available

Explanation of the author

"The CAIDA AS Core visualization depicts the Internet's Autonomous Systems' (ASes) geographic locations, number of customers, and interconnections. Each AS approximately corresponds to an Internet Service Provider (ISP). The geographic location of the individual AS is inferred from the weighted centroid of its address space according to NetAcuity, a commercial geolocation service. The number of direct or indirect customers of an ASA is inferred using its customer cone."

Routing vs. Forwarding

- Routing describes the path determination in a network
  - It is a distributed process among routers in the network
  - Based on the topology detection and path finding
  - As auxiliary tables routers usually use Routing Information Bases (RIBs)
- Forwarding describes the forwarding of packets
  - A local process on one host
  - Incoming packets will be passed to the outgoing port
  - The forwarding happens based on forwarding tables
  - These tables, the Forwarding Information Bases (RIBs) are results from a local routing decision

You can check your local *FIB* using the command `netstat -r` (Windows and Linux) or `ip route show` (Linux only).

Agenda

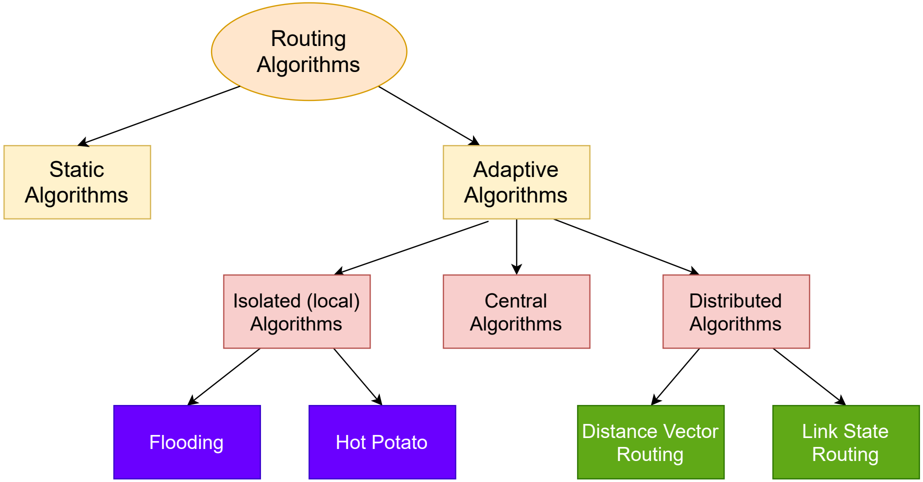
Routing Schemes

- Static routing tables are not flexible enough for the rapidly changing topologies of the Internet
- A routing scheme monitors and exchanges network topology information between routers to allow correct forwarding decisions on each router
- These protocols base on adequate routing algorithms that update the local forwarding tables

Requirements for a routing scheme

- Scalability
- Low overhead (local resources like memory or CPU time and throughput)
- Fault-tolerance
- Loop freedom

Routing Algorithms

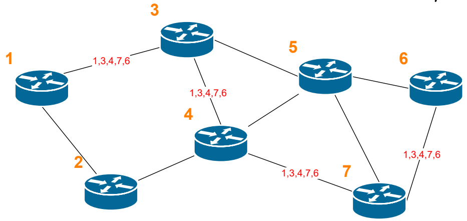


Local Routing Algorithms

- Flooding
  - epidemic principle: each router forwards each packet on all its interfaces (except the one where it was received on)
  - Advantage: very simple and safe against router failures
  - Drawbacks: tremendous overhead, requires loop detection to avoid *broadcast storms*
- Hot Potato
  - Each router immediately forwards any packet on the next available port
  - Advantage: simple, very fast, reduces traffic load on the local network
  - Drawback: inefficient

Source Routing vs. Hop-by-Hop Routing

- Source routing
  - The path is specified by the source node
  - Whole path is included in each packet
  - Advantage: intermediate routers do not need state
  - Drawback: no fault tolerance if a downstream router/link breaks



- Hop-by-hop routing
  - The path is determined by each intermediate router
  - Advantage: fault tolerance if a downstream router/link breaks
  - Drawback: Additional effort per router

Reactive vs. Proactive Routing

- Reactive routing
  - A path is discovered and maintained only on demand
  - Advantage: less control overhead
  - Drawback: path acquisition delay, user data buffering needed
- Proactive routing
  - All possible paths are discovered and maintained in advance
  - Advantage: no path acquisition delay, no user data buffering needed
  - Drawback: overhead due to control traffic

Routing Metrics

- **Questions:** How do determine the best path? How to compare the alternatives?
- The routing protocol defines the *costs* per link: the routing metric
- Typical routing metrics are:
  - Hop count
  - Maximum throughput
  - Latency
  - Link quality (e.g., *ETX (Estimated Transmission Count)*)
  - Energy resources
  - ...

Agenda

Distance Vector Routing

- Goal: Each router maintains a list of cheapest paths for each possible destination
- Each router periodically disseminates its forwarding table to its direct neighbors
- If received information contains a path to a new destination or a cheaper path to a known destination, the router updates its own forwarding table
- Implement the Bellman-Ford algorithm

Routing Information Protocol (RIP)

- The first IGP used in the Internet
- Specified in RFC 1058
- Proactive algorithm
- Functioning of RIP:
  - RIP messages are exchanged every 30 seconds as UDP datagrams between neighbors
  - Used metric is hop count
  - The maximum number of hops is limited to 15 ($\infty := 16$)
  - In a message (only) up to 25 entries of the routing table can be sent
- RIPv2
  - Specified in RFC 2453
  - Support for subnets, CIDR, authentication, multicast, etc.
- RIPng adds support for IPv6
  - Specified in RFC 2080

Bellman-Ford Algorithm

Algorithm

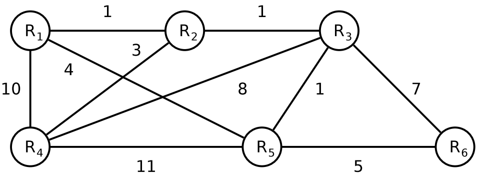
- Initialization of the tables via $\text{Hop}_{ij} \longleftarrow\ ?$ and $\text{Metric}_{ij} \longleftarrow \infty$ for $i \neq j$ and $\text{Hop}_{ij} \longleftarrow \text{R}_i$ and $\text{Metric}_{ij} \longleftarrow 0$ for $i = j$
- For each direct neighbor R_j of R_i this information is stored: $\text{Hop}_{ij} \longleftarrow \text{R}_j$ 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:
  $\text{Hop}_{ik} \longleftarrow \text{R}_j$ and
  $\text{Metric}_{ik} \longleftarrow \text{Metric}_{ij} + \text{Metric}_{jk}$

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

- Initialize tables

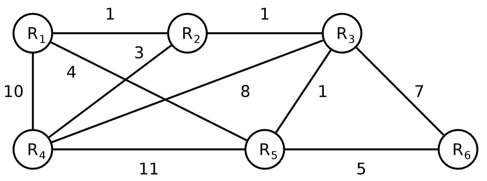
Source: Jörg Roth. Prüfungstrainer Rechnernetze: Aufgaben und Lösungen. Vieweg (2010)

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

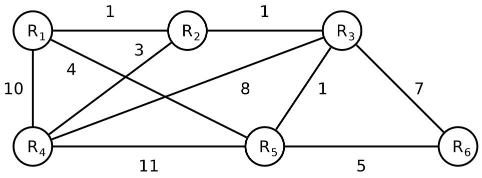
Source: Jörg Roth. Prüfungstrainer Rechnernetze: Aufgaben und Lösungen. Vieweg (2010)

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

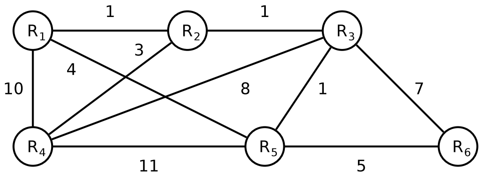
Source: Jörg Roth. Prüfungstrainer Rechnernetze: Aufgaben und Lösungen. Vieweg (2010)

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_2	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

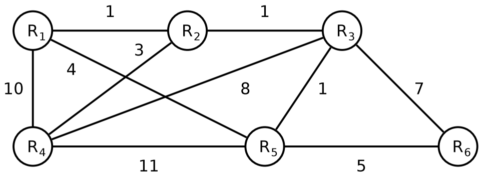
Source: Jörg Roth. Prüfungstrainer Rechnernetze: Aufgaben und Lösungen. Vieweg (2010)

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

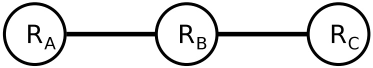
Dest.	Hop	Metric
R_1	R_5	8
R_2	R_5	7
R_3	R_5	6
R_4	R_5	10
R_5	R_5	5
R_6	R_6	0

- State of convergence is reached!
- Compare each entry in the local routing table with the tables of the direct neighbors and adjust table entries when necessary

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

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

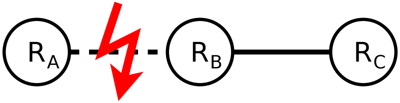
A	B	C	
0	∞	∞	Initial record
0	1	∞	After advertisement round 1
0	1	2	After advertisement round 2
$\vdots$	$\vdots$	$\vdots$	$\vdots$

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 it's 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

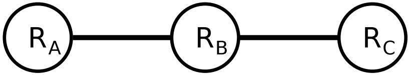
- Caused by the count-to-infinity problem, much time is wasted until the inaccessibility of a route is detected

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

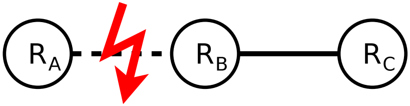
- Solution in some use cases: Split Horizon
- A routing information must not be published via the port through which it was received
  - This prevents a router from transmitting back a routing information to the router, from which it learned the route
- In order to implement *Split Horizon*, not only 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

Split Horizon – Example

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



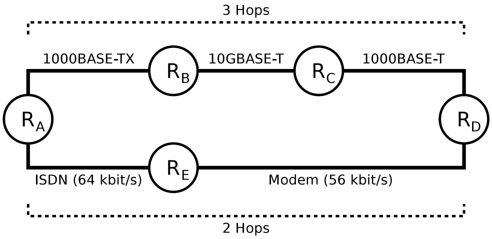
- Scenario: R_A cannot be reached any more



- Effect of split horizon:
  - R_B informs with it's next advertisement to R_C that R_A is not reachable any more
  - R_C modifies its routing table and neither now or in the future sends routing information for R_A to R_B
- Problem: Split horizon fails in many cases $\rightarrow$ Poison Reversed can be used to mark this path as non reachable in the reverse direction

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

Agenda

Link State Routing Protocols

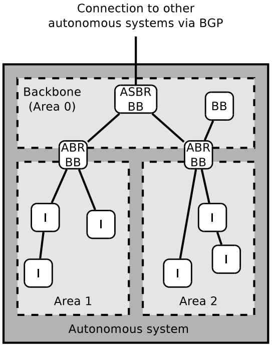
- Implement the *Dijkstra algorithm* (Shortest Path First)
  - Allows the calculation of the shortest path (route) between a starting node and all other nodes in a weighted graph
- Each router . . .
  - can determine the state of the connection to its neighbors and the cost to reach them
  - sends its connection information to all other routers
  - creates his own complete overview with topology information of the network
  - floods the network with regular link-state advertisements
- As a result, the protocol reacts more quickly to changes of the topology and failed nodes
- Drawback: All routers store topology information about the complete network and the network is flooded which creates some overhead

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 split 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
  - Improved security



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)

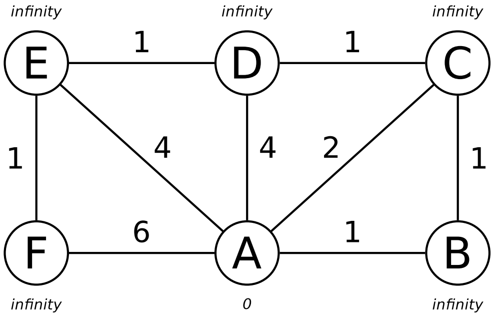
OSPF is far more complex compared with RIP and it will not be discussed in this course in detail

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

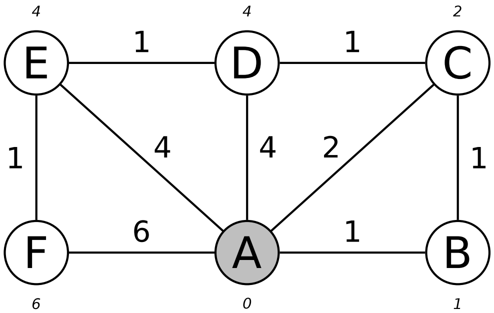
Dijkstra Algorithm – Example (1/7)



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 = {}

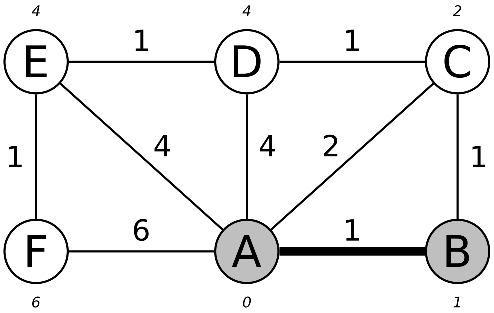
Dijkstra Algorithm – Example (2/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	$\longleftarrow$ 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}

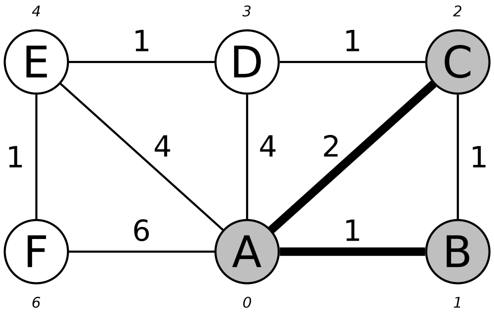
Dijkstra Algorithm – Example (3/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	visited
$d_C = 2$	$\longleftarrow$ 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$\longrightarrow$B}

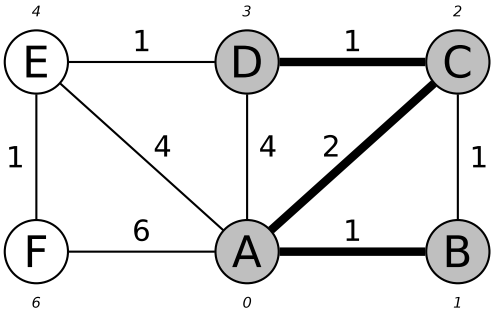
Dijkstra Algorithm – Example (4/7)



Distance values	
$d_A = 0$	visited
$d_B = 1$	visited
$d_C = 2$	visited
$d_D = 3$	$\longleftarrow$ 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}

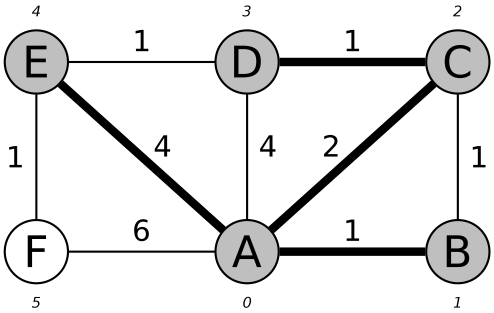
Dijkstra Algorithm – Example (5/7)



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}

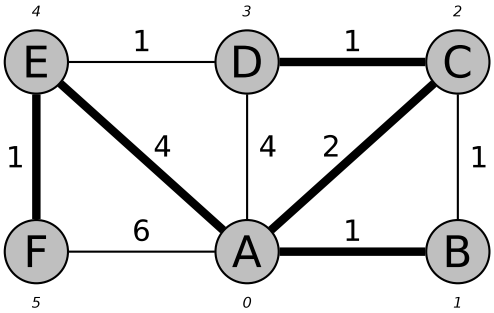
Dijkstra Algorithm – Example (6/7)



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

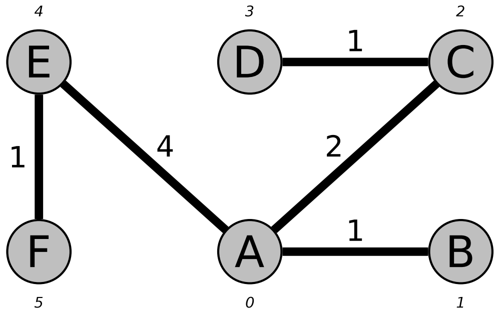
Dijkstra Algorithm – Example (7/7)



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
- 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

Agenda

- Inter-Networking
- Routing Schemes
- Distance Vector Routing
- Link State Routing
- More Routing Protocols

Intermediate System to Intermediate System (IS-IS)

- Standardized in the context of OSI for the connectionless layer 3 protocol
- Republished by IETF as RFC 1142
- Very similar to OSPF, proactive link-state protocol
- Neutral to layer 3 protocol $\rightarrow$ faster support for IPv6 in IS-IS in contrast to OSPF
- Routers also build a map of the network, calculate shortest path
- Lower overhead compared to OSPF
- Often used by network operators with large networks in terms number of routers

Border Gateway Protocol (BGP)

- Specified by IETF in RFCs 1771 – 1773, BGP4 specified in RFC 4271
- Currently the most common inter-domain routing protocol
- A path vector protocol (different from distance vector or link-state)
- BGP routers exchange path vectors with BGP neighbors (*peers*)
- Typically neighbors are connected directly or via a switch
- BGP peers accept or discard paths based on policies (e.g., shortest path, preferred neighbors, hot potato or cold potato)
- BGP router decides on outgoing advertisements based on policies

Routing Protocol for Low power, Lossy Networks (RPL)

- Specified by IETF RFC 6550
- Developed for resource-constrained node networks with lossy links
- Based on distance vector, proactive, tree-based
- Support for hop-by-hop **and** source routing to accommodate lack of memory
- Assumption on data traffic $\rightarrow$ mainly convergecast towards a sink
- Use of flexible Objective Function (OF) as metric
- P2P-RPL: extension to allow paths across the tree (see RFC 6997)

Optimized Link State Routing Protocol (OLSR)

- Specified by IETF in RFC 3626
- Developed for Mobile Ad-hoc NETworks (MANETs)
- Similar to OSPF with overhead reduction
- Proactive link state routing
- **Key principle**: Multipoint Relays (MPRs) selection process (reduces both price of flooding and packet size)

You should now be able to answer the following questions:

- What is the difference between routing and forwarding?
- What are routing metrics?
- Which different types of routing protocols do exist?
- How is the Internet structured and which purposes are fulfilled with IGP and EGP?
- Which algorithm is implemented in a Distance Vector Routing and how does it work?
- Which algorithm is implemented in a Link State Routing and how does it work?