Chair of Distributed Systems and Security
School of Computation, Information and Technology
Technical University of Munich

Computer Networking and IT Security (INHN0012)

Tutorial 8

Problem 1 IPv6 and Supernetting

TUMexam AG has now been assigned the IPv6 address ranges `2001:0db8:0001:000d:0000:0000:0000:0000/64` (*NET1*) and `2001:0db8:0001:000e:0000:0000:0000:0000/64` (*NET2*).

a)* Specify the IPv6 address contained in *NET1* `2001:0db8:0001:000d:0000:00f0:0000:0000` in its compact notation.

- leading zeros are omitted: `2001:db8:1:d:0:f0:0:0`
- the largest consecutive block of at least 2 „zero“ blocks can be abbreviated by `::` resulting in `2001:db8:1:d:0:f0::`

b)* How many addresses does each prefix contain?

$2^{128-64} = 18\,446\,744\,073\,709\,551\,616$

c) How many times can the entire IPv4 address range (`0.0.0.0/0`) be mapped into *NET1*?

$2^{(128-64)-32} = 2^{32} = 4\,294\,967\,296$

d)* What conditions must be met for 2 subnets to be aggregated?

- same size, which implies same prefix length n
- adjacent (the last address in the first network must be followed directly by the next network)
- There must exist a valid prefix mask with length $n-1$, i.e. the two nets must differ only exactly in the last bit of their prefix.

e)* Can the two subnets *NET1* and *NET2* be aggregated into one `/63` subnet?

Although the nets are of the same size and are adjacent, they cannot be aggregated because they are not in the same `/63` prefix. For bits 61 to 64: $d_{16} = 11\underline{0}1_2$, $e_{16} = 11\underline{1}0_2$.
`2001:db8:1:c::/62` would include the two networks, but in addition would also include

- `2001:db8:1:c::/64` and
- `2001:db8:1:f::/64`.

Problem 2 Static Routing

We consider the network topology of *TUMexam AG*, which is shown in Figure 2.1. The goal is to ensure the accessibility of subnets NET1-3 to each other and to the Internet.
Routers R1 and R2 each get the hightest highest usable IP address in the respective subnets assigned.
Transport networks with only two usable addresses each are available for the connection between the routers. In these cases, the router with the lexicographically smaller name (e.g. R1 $\prec$ R2) receives the lower IP address.
The gateway of the *TUMexam AG* is connected to the Internet via its public interface `ppp0`. Its default gateway is 93.221.23.1.

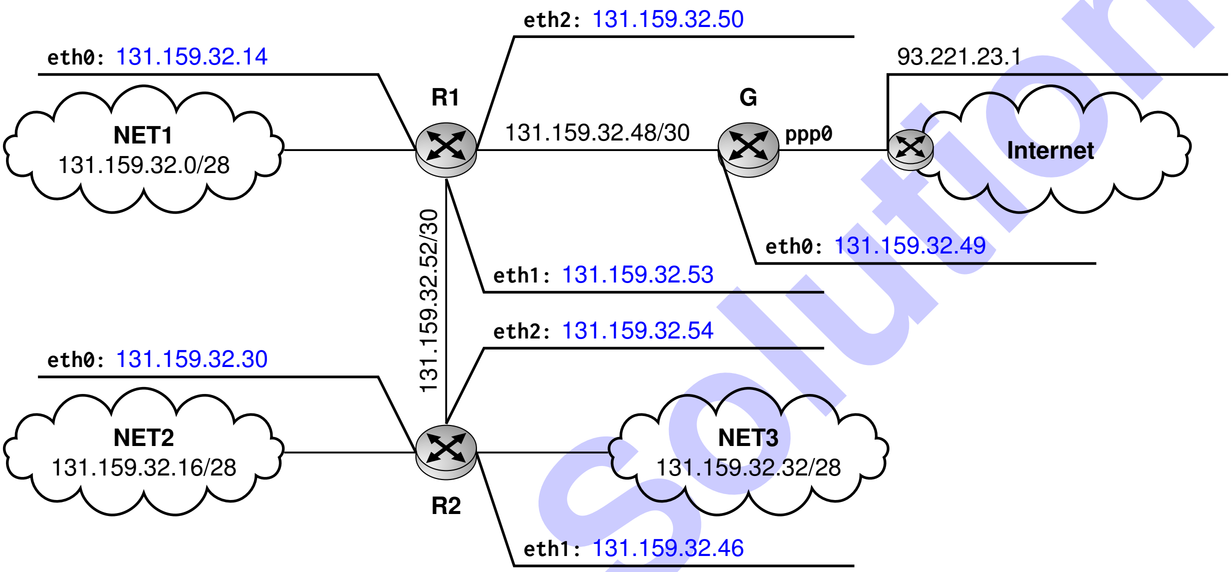

Figure 2.1: Network topology and IPv4 addressing

a)* Assign an IPv4 address to each interface of routers R1, R2, and G (router G only interface `eth0`). Enter the addresses directly in Figure 2.1.

Let the routing table of R2 be given as follows:

Destination	Next Hop	Iface
131.159.32.52/30	0.0.0.0	eth2
131.159.32.16/28	0.0.0.0	eth0
131.159.32.32/28	0.0.0.0	eth1
0.0.0.0/0	131.159.32.53	eth2

Table 2.1: Routing-Tabelle of R2

The entry `0.0.0.0` in the column „Next Hop“ means that no gateway is needed (network is directly connected). The last line is the entry for the so-called *Default-Gateway*. Packets are forwarded there to all those networks for which no better route is known.

b) Specify the routing tables of routers R1 and G. Combine individual routes as far as possible and sort the entries in descending order of the prefix length.

Destination	Next Hop	Iface
131.159.32.48/30	0.0.0.0	eth2
131.159.32.52/30	0.0.0.0	eth1
131.159.32.0/28	0.0.0.0	eth0
131.159.32.16/28	131.159.32.54	eth1
131.159.32.32/28	131.159.32.54	eth1
0.0.0.0/0	131.159.32.49	eth2

Routing table of R1

Destination	Next Hop	Iface
131.159.32.48/30	0.0.0.0	eth0
131.159.32.52/30	131.159.32.50	eth0
131.159.32.32/28	131.159.32.50	eth0
131.159.32.0/27	131.159.32.50	eth0
0.0.0.0/0	93.221.23.1	ppp0

Routing table of G

c)* Why does Router G not necessarily need a route to the transport network 131.159.32.52/30?

Router G needs this route only if it has to reach destinations within this transport network. However, this is not required (according to the task). This has no influence on the accessibility of the subnets NET2 and NET3!

In 2015, the management of the *TUMexam AG* decided to finally start the migration to IPv6. The additional IPv6 addressing is shown in Figure 2.2.

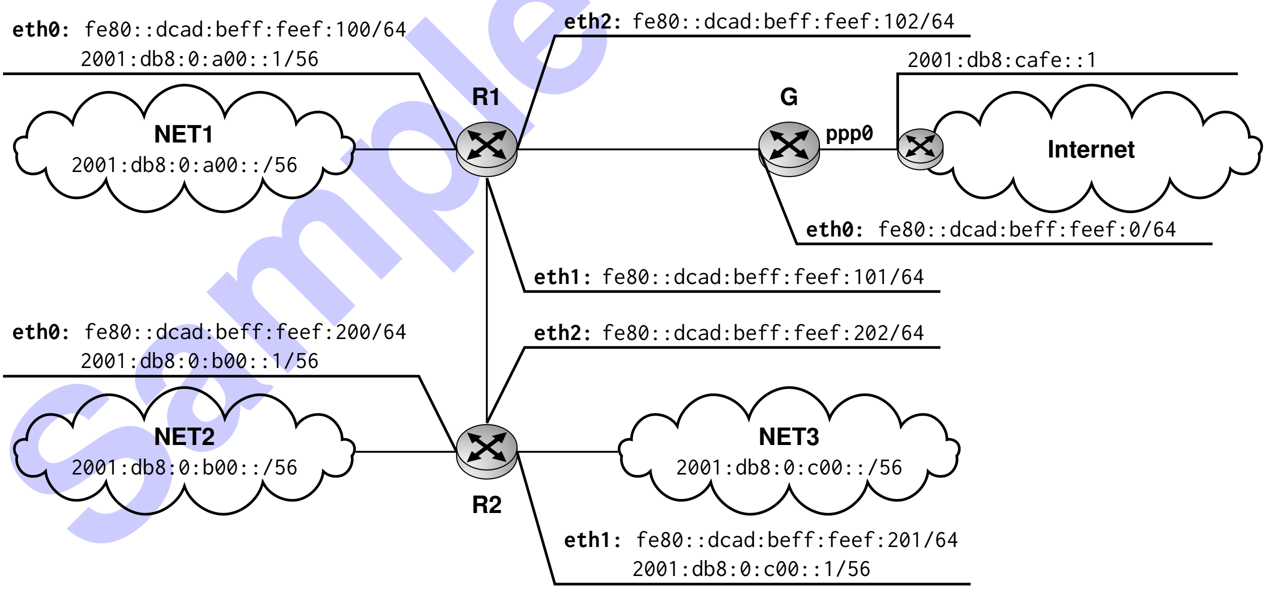

Figure 2.2: Network topology and IPv6 addressing

d)* What is the difference between the two IPv6 addresses `fe80::dcad:beff:feef:201/64` and `2001:db8:0:c00::1/56` of interface `eth1` of R2?

`fe80::dcad:beff:feef:201/64` is a link-local address assigned using SLAAC. `2001:db8:0:c00::1/56`, on the other hand, is a global-unique address.

e)* Specify the first and last address of the subnet to which the address `fe80::dcad:beff:feef:201/64` belongs.

`fe80::` – `fe80::ffff:ffff:ffff:ffff`

f) In which subnet are the link-local addresses of the other devices from the figure 2.2 located?

All in the same: `fe80::/64`.

g) Is it a problem that the `fe80::/64` subnet seems to be assigned multiple times?

No, because link-local addresses are only valid in the local subnet (scope link) and are never routed.

h) The default gateway of G is `2001:db8:cafe::1` and is reachable via its external interface `ppp0`. Set up the IPv6 routing table for Router G. To do this, group entries together again as far as possible and sort the entries in descending order of the length of the prefix.

Destination	Next Hop	Iface
`fe80::/64`	`::`	eth0
`2001:db8:0:c00::/56`	`fe80::dcad:beff:feef:102`	eth0
`2001:db8:0:a00::/55`	`fe80::dcad:beff:feef:102`	eth0
`::/0`	`2001:db8:cafe::1`	ppp0

IPv6 Routing-Tabelle of G

Problem 3 Distance Vector Routing (Homework)

Assume the topology shown in Figure 3.1 with the four routers A to D. The link costs are indicated at each edge. We note the routing tables in short form as a vector $[(x_A, y_A), \ldots, (x_D, y_D)]$. The tuples (x, y) indicate the cost and the next hop to the destination.
For example, the shortest path from A to B is via B with a cost of 2, from A to C via C with a cost of 1, and from A to D via C with a cost of 2. Routers reach themselves by definition with cost 0. This then results in the routing table `[(0,A) (2,B) (1,C) (2,C)]` for router A (the position within the vector indicates the respective destination).
At the beginning, the routing tables are still empty, i.e., the routers do not even know their direct neighbors. This is indicated by the notation `(/,/)`. The routers of course reach themselves with a cost of 0.

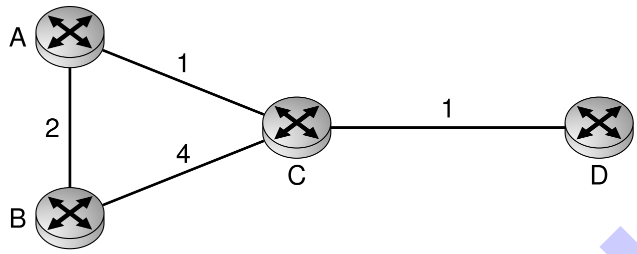

Figure 3.1: Network topology

The routers now start exchanging their distance vectors with their direct neighbors at periodic time intervals. For example, router B sends an update to router C, which only contains the distance to the respective destination (but not the next hop). Now if router A receives such an update from B and would find a route to D in it, A would know that he reaches D via B. The cost to D is then equal to the cost to B plus the cost to B to achieve the goal.

In the following, we will examine this behavior. However, since the result depends on the order in which updates are exchanged, we make the idealized assumption that all routers send their updates at exactly the same time.

a)* According to the above definitions, specify the routing tables of all four routers in the following steps. Abort as soon as a convergent state is reached.

Step	Router A	Router B
0	[(0,A) (/,/) (/,/) (/,/)]	[(/,/) (0,B) (/,/) (/,/)]
1	[(0,A) (2,B) (1,C) (/,/)]	[(2,A) (0,B) (4,C) (/,/)]
2	[(0,A) (2,B) (1,C) (2,C)]	[(2,A) (0,B) (3,A) (5,C)]
3	[(0,A) (2,B) (1,C) (2,C)]	[(2,A) (0,B) (3,A) (4,A)]

Step	Router C	Router D
0	[(/,/) (/,/) (0,C) (/,/)]	[(/,/) (/,/) (/,/) (0,D)]
1	[(1,A) (4,B) (0,C) (1,D)]	[(/,/) (/,/) (1,C) (0,D)]
2	[(1,A) (3,A) (0,C) (1,D)]	[(2,C) (5,C) (1,C) (0,D)]
3	[(1,A) (3,A) (0,C) (1,D)]	[(2,C) (4,C) (1,C) (0,D)]

b) Which (graph) algorithm is used here?

A distributed (decentralized) implementation of the Bellman-Ford algorithm is used.

Now the connection between nodes C and D fails. Nodes C and D notice this and set the corresponding path cost to infinity.

c) What happens in the following steps where the active nodes continue to exchange their distance vectors? After each step, specify the distance tables until the further result is clear.

Step	Router A	Router B
4	[(0,A) (2,B) (1,C) (2,C)]	[(2,A) (0,B) (3,A) (4,A)]
5	[(0,A) (2,B) (1,C) (6,B)]	[(2,A) (0,B) (3,A) (4,A)]
6	[(0,A) (2,B) (1,C) (4,C)]	[(2,A) (0,B) (3,A) (7,C)]
7	[(0,A) (2,B) (1,C) (8,C)]	[(2,A) (0,B) (3,A) (6,A)]
8	[(0,A) (2,B) (1,C) (6,C)]	[(2,A) (0,B) (3,A) (9,C)]
9	[(0,A) (2,B) (1,C) (10,C)]	[(2,A) (0,B) (3,A) (8,A)]
10	[(0,A) (2,B) (1,C) (8,C)]	[(2,A) (0,B) (3,A) (11,C)]
⋮	⋮	⋮

Step	Router C	Router D
4	[(1,A) (3,A) (0,C) (/,/)]	[(/,/) (/,/) (/,/) (0,D)]
5	[(1,A) (3,A) (0,C) (3,A)]	[(/,/) (/,/) (/,/) (0,D)]
6	[(1,A) (3,A) (0,C) (7,A)]	[(/,/) (/,/) (/,/) (0,D)]
7	[(1,A) (3,A) (0,C) (5,A)]	[(/,/) (/,/) (/,/) (0,D)]
8	[(1,A) (3,A) (0,C) (9,A)]	[(/,/) (/,/) (/,/) (0,D)]
9	[(1,A) (3,A) (0,C) (7,A)]	[(/,/) (/,/) (/,/) (0,D)]
10	[(1,A) (3,A) (0,C) (11,A)]	[(/,/) (/,/) (/,/) (0,D)]
⋮	⋮	⋮

d)* In the lecture, **Split Horizon**, **Triggered Updates**, and **Path Vector** were mentioned as possible countermeasures for the count-to-infinity problem. As a group, explain how these procedures work.

- **Split Horizon**
  „Do not advertise a route over the interface over which the route was originally learned.“ Thus, in the concrete case, routers A and B would not send a route to D at C. However, this still does not fix the problem due to the circular topology.
- **Triggered Update**
  Instead of sending routing updates only at periodic intervals (by default 30 s in RIP), updates are sent immediately when topology changes are detected. While this does not solve the count-to-infinity problem, it does speed up the process. The problem is that a lot of traffic is caused by routing updates for a short time. Triggered updates are supported by most routing protocols (RIP only from version 2).
- **Path Vector**
  It would be possible for each router to remember where an update came from and include this information in routing updates. In this way, each router could track the full path that a packet will take to its destination. If a router discovers itself in this path, it knows thata loop is present. It can discard the update in this case. This procedure is used by BGP.