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.

b)* How many addresses does each prefix contain?

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

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

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

Problem 2 ARP and IP fragmenting

Figure 2.1 shows an arrangement of network components with their IP and MAC addresses. The two computers PC1 and PC2 use the respective local router as default gateway. PC1 sends an IP packet with 1000 B payload data to PC2. The MTU on the WAN link between R1 and R2 is 580 B. Within the local networks the usual Ethernet MTU of 1500 B applies.

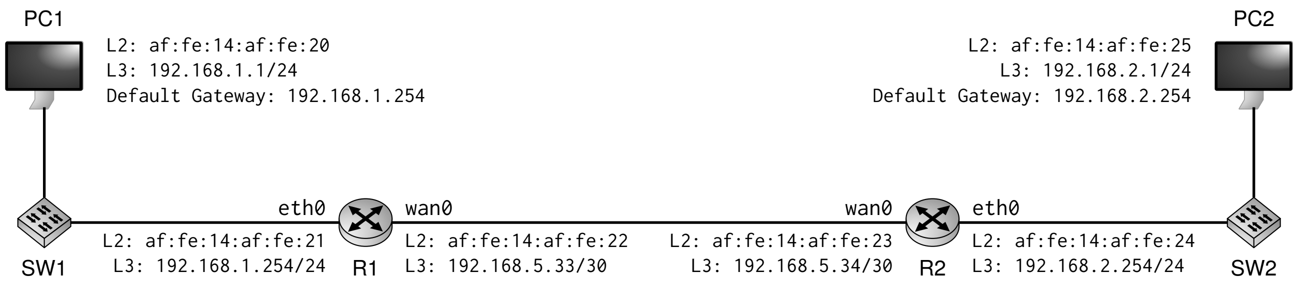


Figure 2.1: Network topology

In the following, the transfer of the packet with all necessary intermediate steps shall be traced. Assume initially that the ARP caches of all network components involved have been flushed.

a)* To what extent do the two switches SW1 and SW2 have an effect in this example?

b)* Into how many fragments must R1 break down the packet from PC1?

c)* At what point in the network are the fragments reassembled?

d) Sketch a simple path-time diagram that considers **all frames** that need to be transferred on each link. **Name the type of frames replaced and number the frames (1,2,3,...).** (The diagram does not need to be to scale. Serialization times and propagation delays are to be ignored.)

At the end of this exercise sheet you will find preprinted forms for Ethernet header, ARP packets (header and payload) and IP header (more than needed). It is not necessary to fill in the header in binary. Just be sure to clearly mark the number base, e.g., `0x10` for hexadecimal or $63_{(10)}$ for decimal.

e) For each of the first three frames from subtask d), fill in an Ethernet header and the appropriate payload (ARP packet or IP header with indicated payload). Label the dashed box next to each header/packet with the frame number assigned in subtask d).

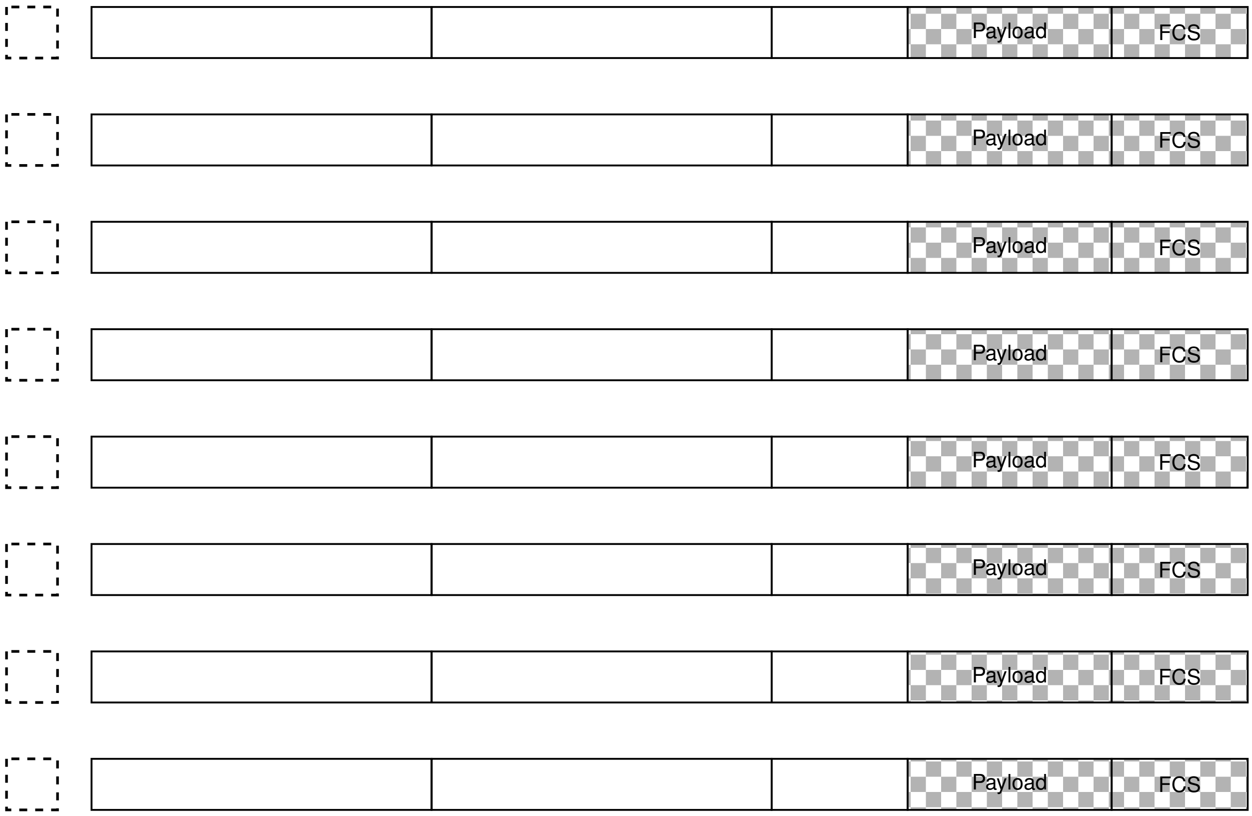
f) Fill in an Ethernet and IP header for each of the remaining frames that transport an IP payload. Label the dashed box next to each header with the frame number assigned in subtask d).

g)* Assume that PC1 and PC2 communicate via IPv6:

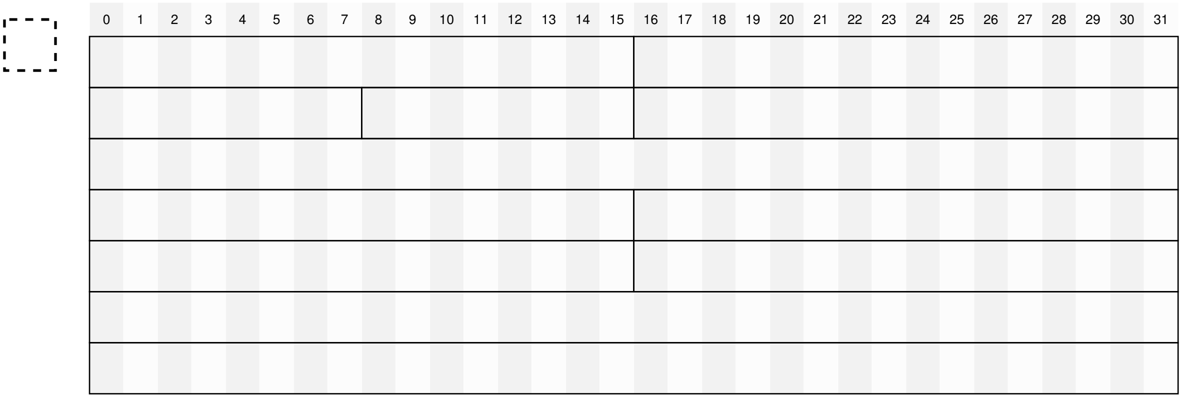
1. What impact would this have on switches SW1 and SW2?
2. In this case, would routers R1 and R2 also have to be IPv6-capable?
3. Where would the fragmentation of packets take place?

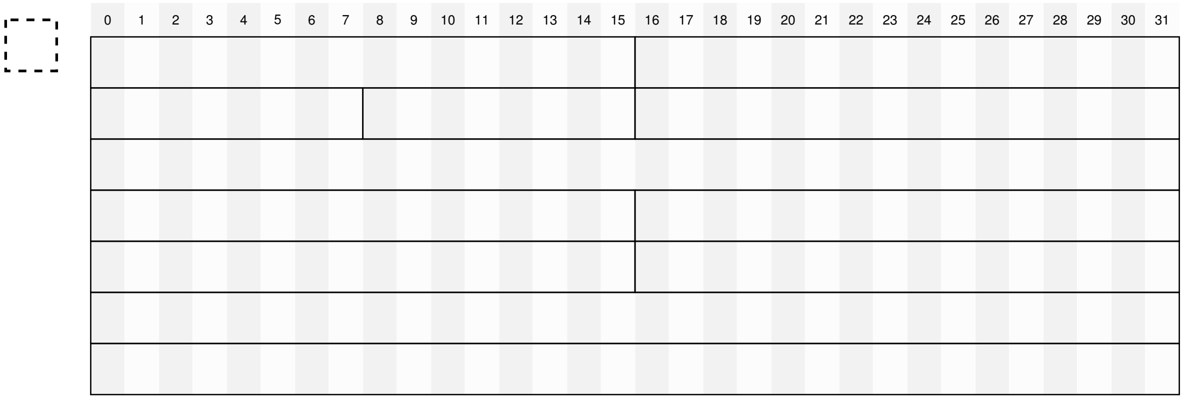
Preprints for protocol headers:

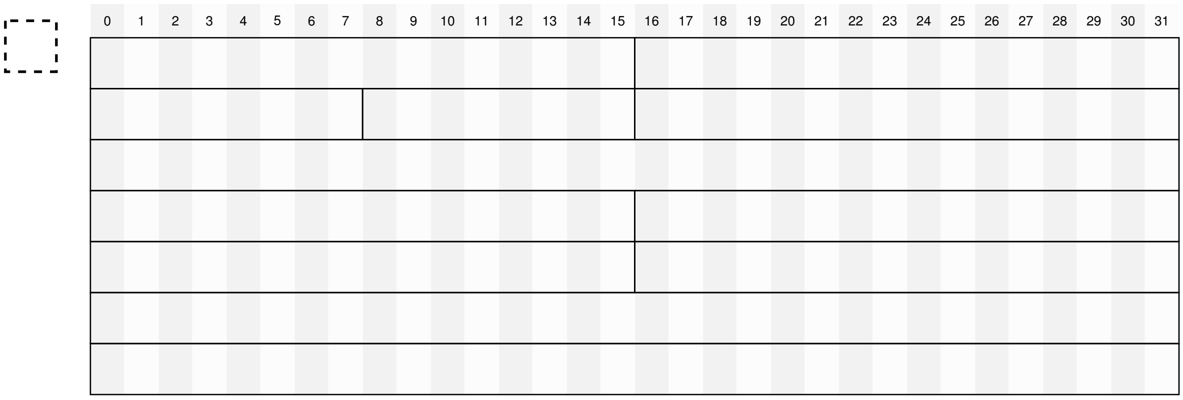
Ethernet frames



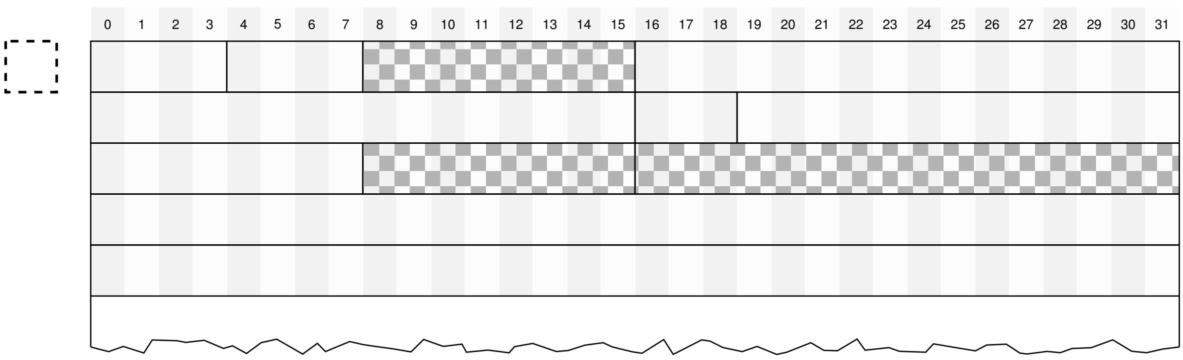
ARP packets

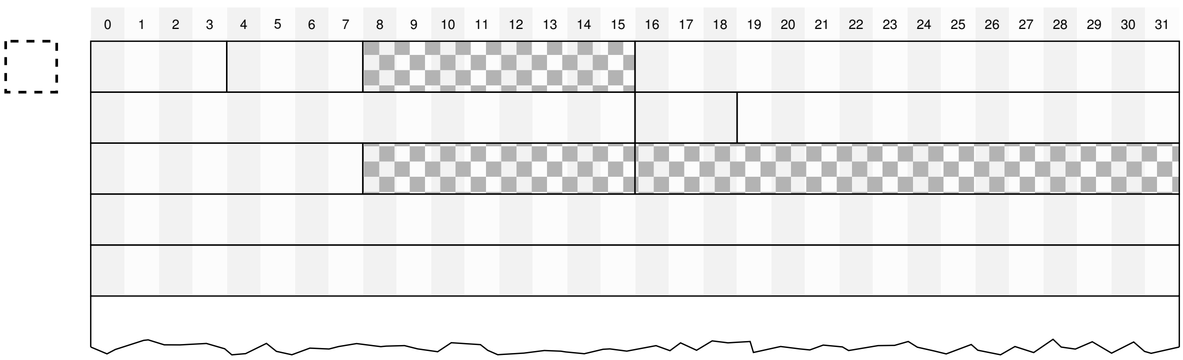


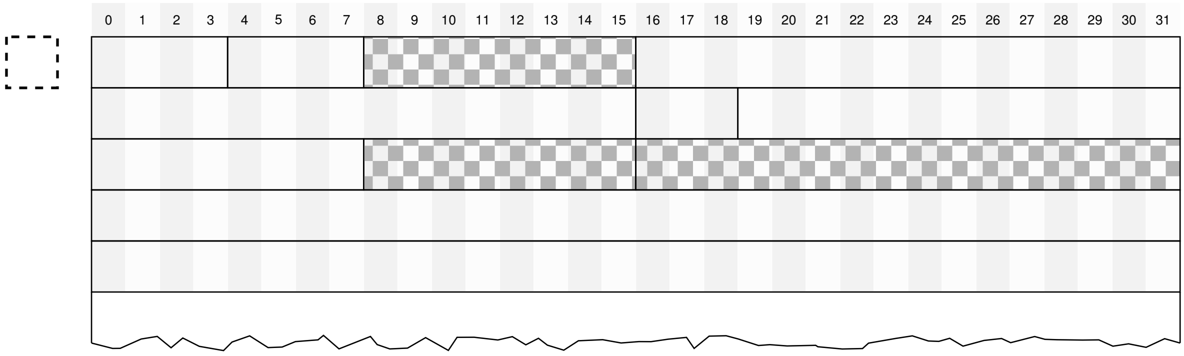

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

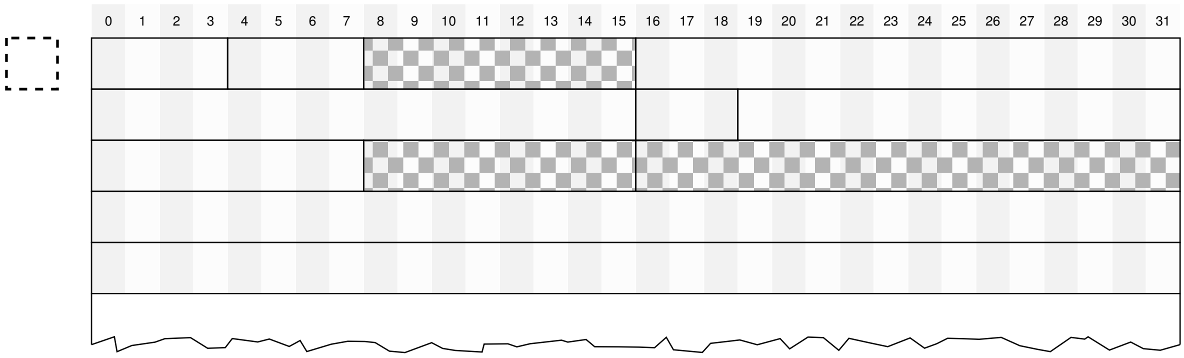

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

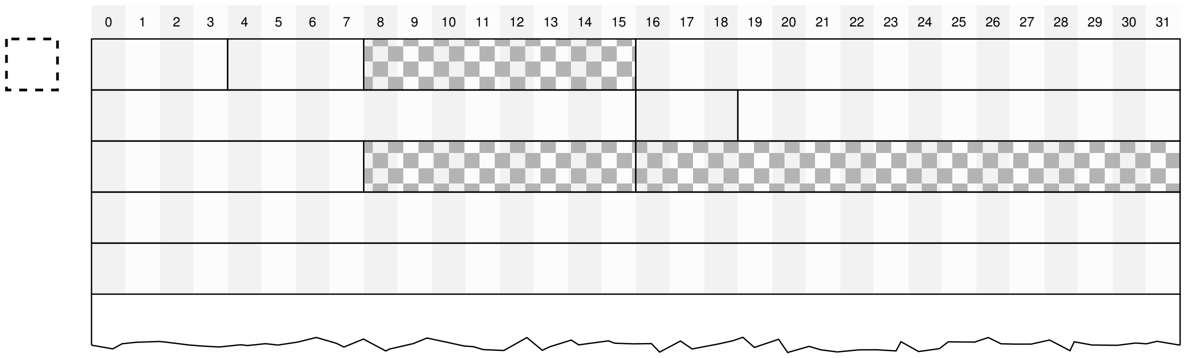
IP packets

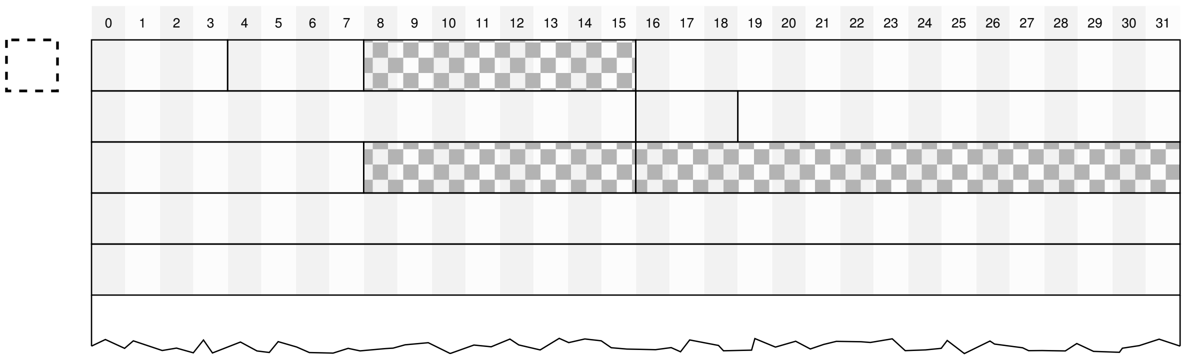

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

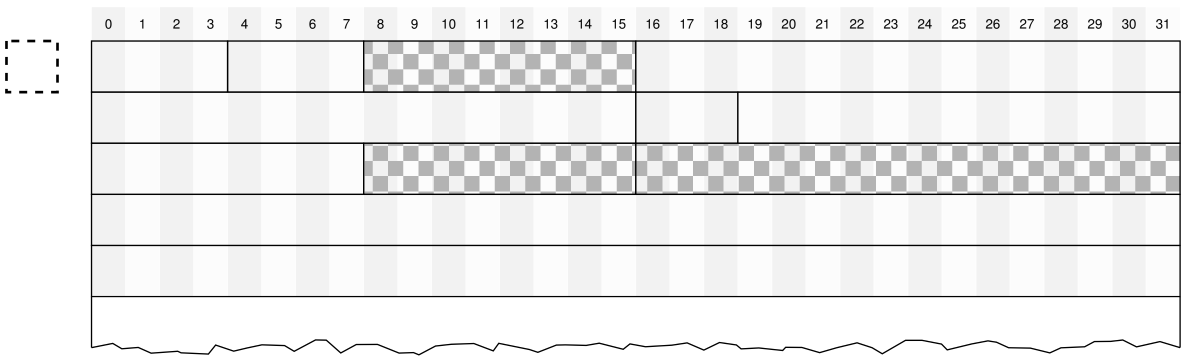

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

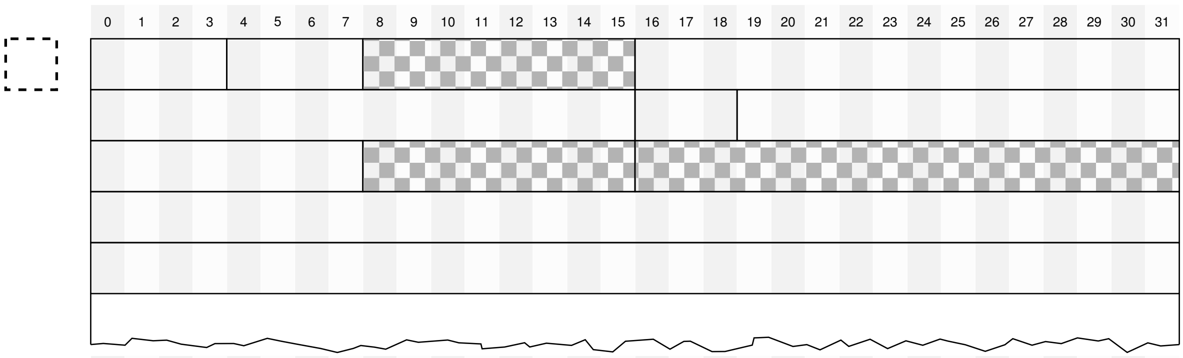
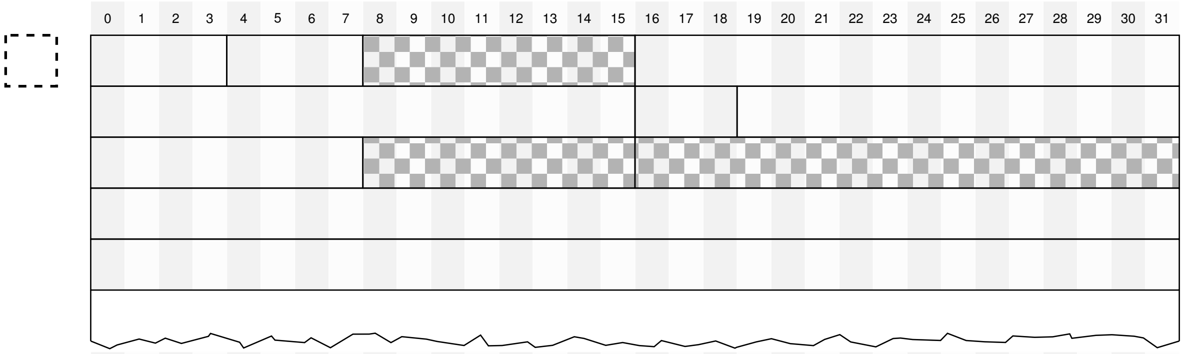
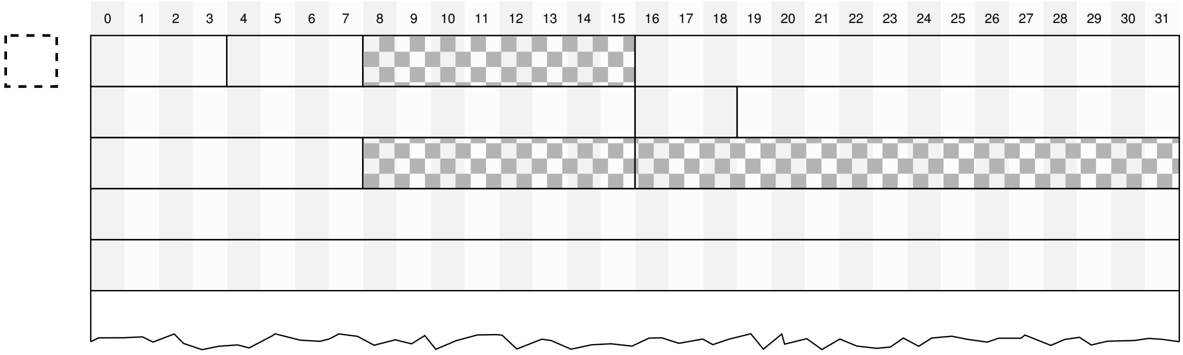

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31