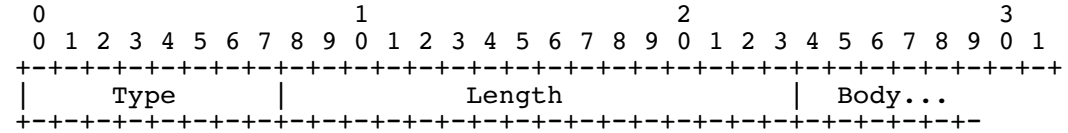
Advanced Networks — Laboratory 12

Juliusz Chroboczek

20 May 2025

Exercice 1. Type the command `ping ff02::1%eth0`, where `eth0` is the name of the interface that is used for routing IP packets on your machine (you may use the command `ip route show` to find its name). Explain.

The goal of this practical is to implement a peer-to-peer thoughtful quotation distribution protocol based UDP, where discovery happens over multicast and data transfer happens over unicast. The protocol is based on UDP, and every datagram contains exactly one TLV of the following format:



There are four kinds of messages:

- 0, *discovery request*, may be sent to either a unicast or a multicast address; the receiver is supposed to answer with a *discovery reply* message;
- 1, *discovery reply* replies to a *discovery request*, and can only be sent to a unicast address; the body contains the nickname of the sender;
- 2, *quotation request*, requests a *quotation reply*; it is always sent to a unicast address, and its body is empty;
- 3, *quotation reply*, can only be sent to a unicast address; the body contains a thoughtful quotation encoded in UTF-8.

Exercice 2.

1. Write a program that sends a *discovery request* message to the UDP port and group address `ff12::00fc:7c23:02e2:a7a9`, then waits for replies indefinitely and displays the source addresses and the nicknames contained in the replies.
2. What is specific about the addresses collected? Are they suitable for use in a cross-link algorithm such as gossip?
3. Modify your program so that it waits for five seconds then terminates.
4. Modify your program so that it sends three packets in one-second intervals while collecting the answers. Why is that necessary?

5. Modify your program so that it sends a *quotation request* to every address that it collected, and displays the collected quotations.
6. If you have time, implement resending of requests with exponential backoff.

Exercice 3.

1. Write a program that binds a UDP socket to port 4242, subscribes to the multicast group above, then:
   - upon receiving a *discovery request*, it waits for a random time between 0 and 200 ms, then answers with a *discovery reply* to the sender;
   - upon receiving a *quotation request*, it replies with a *quotation reply* with your favourite thoughtful quotation.

   You may use either the function `net.ListenMulticastUDP` or the module `x/net/ipv6`.
2. Why is there a delay for the *discovery reply* but not for the *quotation reply*?
3. What happens if your program receives a *discovery request* sent over multicast? Solve the problem if you have time left.