TTM4200 Computer Networking Support Document - Lab 4

Christian Lewin and Sebastian Fuglesang

June 2020

1 Traffic Control

1.1 Concept

Traffic Control allows us to simulate different properties with our networks. We can use it to increase delay, packetloss, limit our bandwidth and more. Note that we can't use it to directly increase the performance. We can't use it to remove packetloss directly. It is a useful thing to know. It allows you to limit different applications and thus balance your system yourself. In this lab we will use it to simulate different properties and see how this impacts different programs.

1.2 Command

- We will be using linux traffic control. There are several ways to implements traffic control. Linux tc only impacts the specified interface outgoing traffic. So if you have two computers that are connected and you apply packetloss on one of them it will only count on that computer's outgoing packets.
- Syntax:

```
sudo tc qdisc add/del/replace dev ether0 root netem delay/loss/corrupt 100ms/10%/10%
```

  - `tc` : stands for traffic control and is the keyword for linux traffic control
  - `qdisc` : stands for queing discipline.
  - `add/del/replace` : you can either add, delete or replace a "rule". A single interface can only have one active "rule" at a time. If you already have a "rule" on that interface then use replace.
  - `dev ether0` : here you specify which interface that the TC will be applied. In this example ether0.
  - `netem` : is a classless qdisc and stands for network emulator. By adding it we say we will use netem to emulate the traffic.
  - Finally we have which property we want to emulate. For instance `loss` or `delay`. And then we say by how much.
- Examples
  - `sudo tc qdisc add dev ether0 root netem delay 200ms` : to add a TC rule for 200ms delay on ether0.
  - `sudo tc qdisc del dev ether0 root` : to remove the rule on ether0.
  - `sudo tc qdisc replace dev ether0 root netem corrupt 15%` : to replace the current acting TC with a new rule specifying 15% packet corruption.

- `sudo tc qdisc add dev eth0 root tbf rate 32kbit burst 32kbit latency 400ms`: to add a rule that limits the bandwidth usage to 32kbit/s and adds a 400 ms latency. `tbf` stands for "token bucket filter" which is a traffic shaper that limits the transmitted traffic within a sustained maximum `rate` and a maximum allowed `burst`. The packets with higher `latency` get dropped.

2 VLC

- VLC is a multimedia-player that can play many different file formats. It is used in this lab to stream videos over a network.
- Syntax:

```
cvlc -vvv /home/ttm4200/work_dir/videos/frog.mp4 --loop --ttl 12 --sout \
'#std{access=udp,mux=ts,dst=10.20.40.1:9082}'
```

  - `cvlc`: is a version of VLC without user interface that is ran from the terminal.
  - `v`: means verbosity. Verbosity is related to how error messages are written in console. The lower the level the less is written in console. There are three levels level 0, 1 and 2. By having -vvv we put it at level 2.
  - next is the filepath of the video we will be streaming.
  - `loop`: is used so that once the video has been played it will start a new round automatically.
  - `ttl 12`: sets the time to live of the packets to 12.
  - `sout`: stands for stream output.
  - `access`: sets what protocol will be used to send the packets of the stream.
  - `mux`: allows you to set the encapsulation method for the stream. ts stands for MPEG-TS and can be used for all access types.
  - `dst`: is the destination of the stream. Here both the IP address and the port number.

3 State machines

For this lab there will be provided two images that show the function of a sender and a receiver. In this section those will be explained.

3.1 Sender

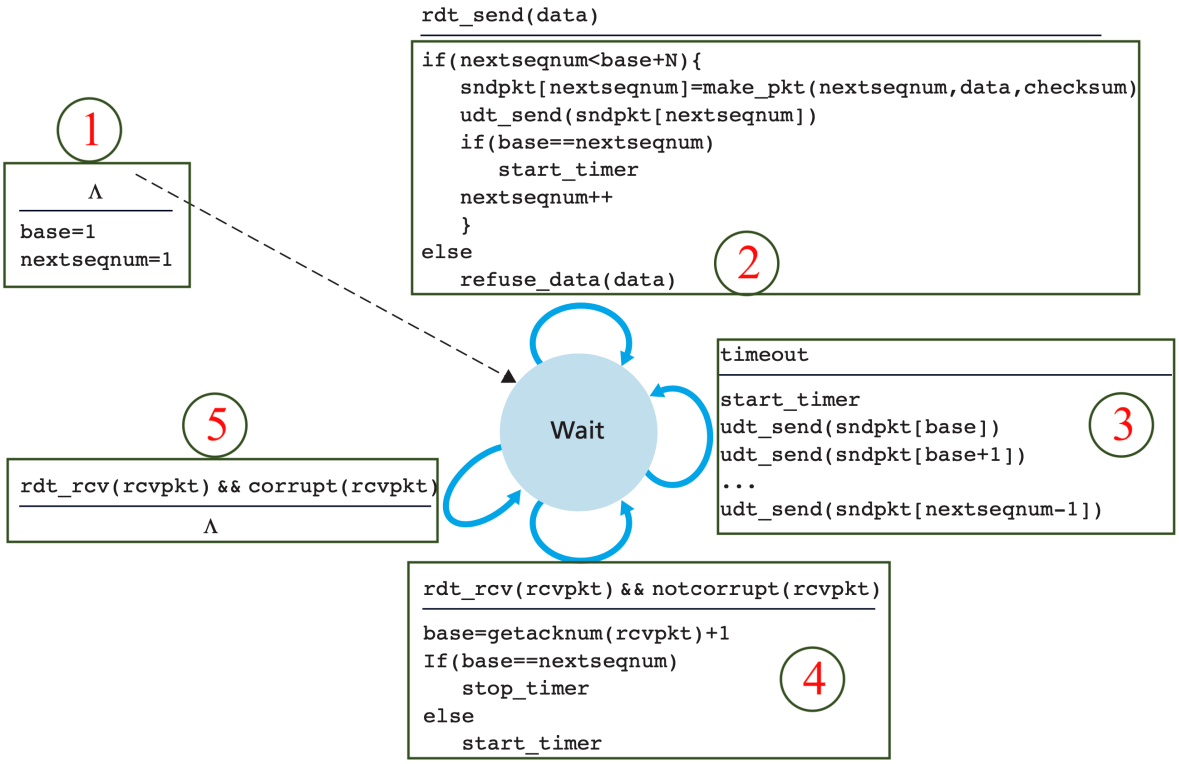

Figure 1: Sender (Kurose, J. and Ross, K)

- The sender prepares to start sending, sets base and nextseqnum to 1.
- If the sender has more “room” in the window it will attempt to send a packet.
  - If nextseqnum equals base, a timer is started to keep track of packet timeout.
  - nextseqnum is incremented by one.
  - If the sender doesn’t have more room in the window the sender will deny the data.
- If a timeout occurs the timer is restarted and the sender will attempt to resend all the packets in its window again.
- The sender here has logic to receive packets (acks).
  - If the sender is ready to receive a packet and the packet is not corrupted then base is incremented by one over acknum from the received ack.
  - If base equals nextseqnum the timer is stopped. Otherwise the timer is restarted
- If the received ack is corrupted the sender does nothing.

3.2 Receiver

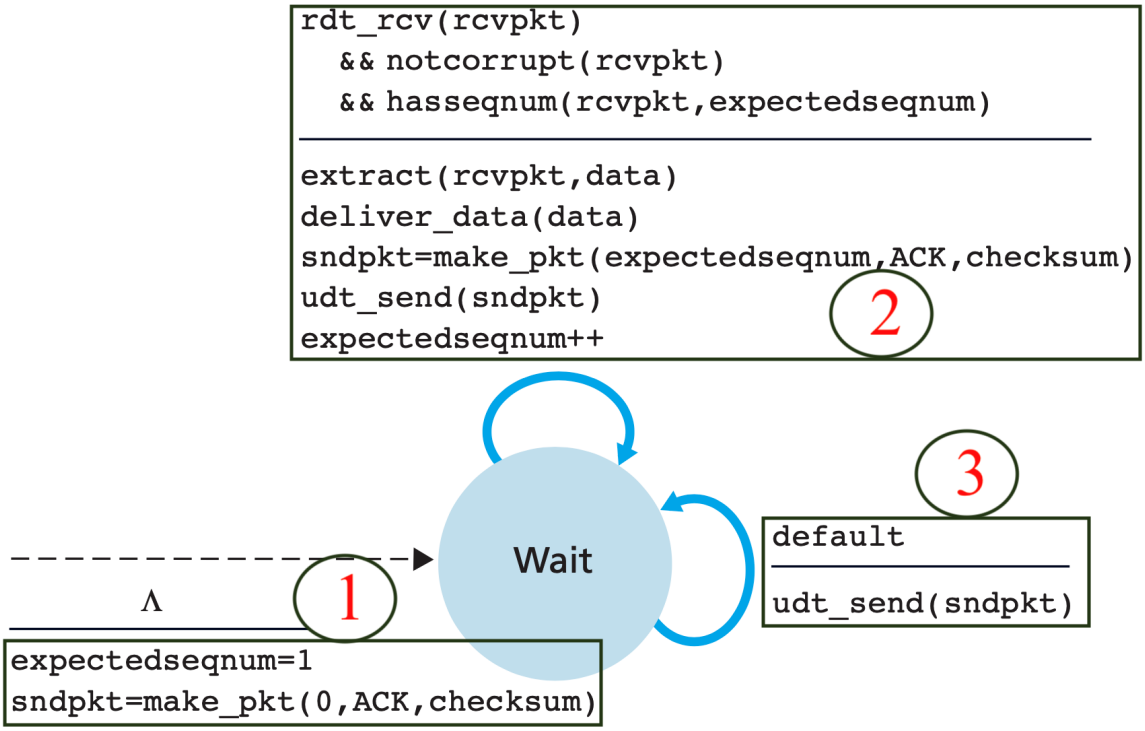

Figure 2: Receiver (Kurose, J. and Ross, K)

- The receiver has an expected sequence number called, expectedseqnum, set to 1. The receiver also makes a packet called, sndpkt, which functions as an ACK and will be sent if the receiver receives an unexpected packet.
- If the receiver receives a packet that is not corrupted and has the expected sequence number, then
  - It will extract the information from the packet and pass it along to the application
  - Make a new packet named, sndpkt, with the expected sequence number.
  - Send sndpkt over UDP and increment expectedseqnum with 1.
- If the received packet is not as expected. Either corrupted or not the expected sequence number, the receiver will send the latest sndpkt.