Implement your own **Reliable** Transport Protocol

Recover from packet loss and reordering.

Part 1

Part 2

Part 3

Part 4
Implement your own Reliable Transport Protocol

Recover from packet loss and reordering.

Part 1
Complete a simple Go-Back-N implementation
Retransmit packets after a timeout.

Part 2

Part 3

Part 4
Implement your own **Reliable** Transport Protocol

- Recover from packet loss and reordering.

**Part 1**
Complete a simple **Go-Back-N** implementation
Retransmit packets after a timeout.

**Part 2**
Add support for **Selective Repeat**
Fast retransmission after duplicate ACKs

**Part 3**

**Part 4**
Implement your own Reliable Transport Protocol

Recover from packet loss and reordering.

Part 1
Complete a simple Go-Back-N implementation
Retransmit packets after a timeout.

Part 2
Add support for Selective Repeat
Fast retransmission after duplicate ACKs

Part 3
Add support for Selective Acknowledgments
Receiver indicates correctly received segments

Part 4
Implement your own **Reliable** Transport Protocol

Recover from packet loss and reordering.

**Part 1**
Complete a simple **Go-Back-N** implementation
Retransmit packets after a timeout.

**Part 2**
Add support for **Selective Repeat**
Fast retransmission after duplicate ACKs

**Part 3**
Add support for **Selective Acknowledgments**
Receiver indicates correctly received segments

**Part 4**
Implement your own **Congestion Control** algorithm
Adjust the window size according to the current network conditions.
The assignment text on the GitLab Wiki contains a lot more details.

https://gitlab.ethz.ch/nsg/lectures/lec_commnet/projects/2024/transport_project/assignment/-/wikis/home

You will implement a GBN sender and receiver using Python as a programming language.

We already provide you with a skeleton.
Have a look at the comments in the skeleton files.

Make sure to follow the instructions carefully.
They contain critical information to get all points.
A new VM and GitLab Repository wait for you.

Please let us know if you have no access!

Your **group numbers changed!**
Your VM Port number is **2000 + group number**.

Use the password from your GitLab repository.
Use our Test Framework

- **run test**  
  Test your implementation  
  There are > 60 test cases!

- **run self**  
  Transmit bytes from your sender to your receiver.

- **run other**  
  Transmit bytes to a different group.

- **run cc**  
  Plot the throughput, pkt drops, and window size over time.
You will mainly use **run test**

- **run test**: Test your implementation.
  This is how we determine your grade!

- **run self**: Transmit bytes from your sender to your receiver.

- **run other**: Transmit bytes to a different group.

- **run cc**: Plot the throughput, pkt drops, and window size over time.
KEEP CALM
IT'S
DEMO TIME!!!
Use `run cc` to debug your congestion control
Your project grade counts **10%** to your final grade

**project grade**

\[=\]

**public grade**

\[=\]

Computed based on your number of points

\[1 + \text{POINTS} / 100\]

Maximum: **5.5**

**run test -a** will show your current public grade!

**secret grade**

We will run additional test cases

These are **not** available to you during the project.

We also check your code and **comments**!

Maximum: **0.5**
We will check your submitted code for plagiarism using automated tools!

Do not copy code from other groups or any other project-related sources!

We had multiple such cases in the past! You will immediately receive a 1.0 as final grade.
If you have questions:

Ask us on Matrix or email us.
Please use the #transport_project channel.

We will announce additional Q&A sessions if needed
Final remarks

Deadline: **May 31, 2024, 23:59**
Submit your code and the declaration of originality via GitLab

Read the assignment text carefully,
make sure you follow all specifications.

You do **not** have to write a report,
but don’t forget to comment your code.