Two important pillars of today’s Internet

Internet-wide routing
Covered in the first project

Reliable transport
Main focus of the second project
Implement your own **Reliable** Transport Protocol

recover from packet loss
and reordering
Implement your own **Reliable** Transport Protocol

- recover from packet loss
- and reordering

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

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

**Part 3**  
Add support for Selective Acknowledgements (SACK)  
SACK contains blocks of correctly received segments
Proposed timetable

<table>
<thead>
<tr>
<th>Part 1</th>
<th>Complete a simple Go-Back-N implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>20.05.2022</strong></td>
<td>Retransmit all packets after a timeout</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part 2</th>
<th>Add support for Selective Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>27.05.2022</strong></td>
<td>Fast retransmission after duplicated ACKs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part 3</th>
<th>Add support for Selective Acknowledgements (SACK)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>03.06.2022</strong></td>
<td>SACK contains blocks of correctly received segments</td>
</tr>
</tbody>
</table>

We use a custom header for the GBN protocol

<table>
<thead>
<tr>
<th>T</th>
<th>Padding</th>
<th>F</th>
<th>S</th>
<th>Segment Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Header Length</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sequence Number</td>
<td></td>
<td></td>
<td>Window</td>
</tr>
<tr>
<td></td>
<td>Block Length</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Left edge 1st block</td>
<td></td>
<td></td>
<td>Length 1st block</td>
</tr>
<tr>
<td></td>
<td>Padding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Left edge 2nd block</td>
<td></td>
<td></td>
<td>Length 2nd block</td>
</tr>
<tr>
<td></td>
<td>Padding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Left edge 3rd block</td>
<td></td>
<td></td>
<td>Length 3rd block</td>
</tr>
</tbody>
</table>

Payload
The assignment text on the GitLab wiki contains a lot more details

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

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

Make sure that you follow the instructions carefully
Interoperability will be difficult otherwise
A new VM and **GitLab repository** wait for you

let us know if you have no access

The skeleton files are already on your VM

Use git (preferred) or scp to transfer files

Your group numbers changed

**Important:** VM port number is 2000 + group number

Use the password from your GitLab repository
Let’s see how the you can execute your GBN implementation and test it
There are multiple options to test your implementation

Run your sender against your receiver
This should be your main focus

Use the various tests in the test framework
Will allow you to test specific features

Run against code of another group
The test framework handles that as well
We also generate an interoperability matrix on our website
Your project grade counts as 10% to your final grade

Project grade = Test performance (public) + Additional tests (secret)

Computed based on the amount of points you get
TEST_POINTS / 400 * 4.5 + 1

This way you can get up to a 5.5
Running all the tests will show you this grade

We will run your submitted code through additional tests
They are not available to you during the project

In addition we check your code for basic comments
This way you can get the remaining 0.5 grade points
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 clear cases in the past
You will immediately receive a one as final grade
If you have questions

Ask on Slack or send us an email
Please use the #transport_project channel

We will announce additional Q&A sessions if needed
During online sessions we will also be available for calls

Follow the exercise session this Thursday
More demos and information about the test framework
No exercise session during the second week

Due to Ascension day
ETH is closed

Soon we will announce an additional in-person session
More information follows via email and Slack
Final comments

Deadline: **June 03 2022, at midnight**
Submit your code and decl. of originality via GitLab

Read the assignment text carefully
Make sure you follow all the specifications

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