Master-Thesis Presentation by Özcan Karaca

Testbed-Development for lectureStudio

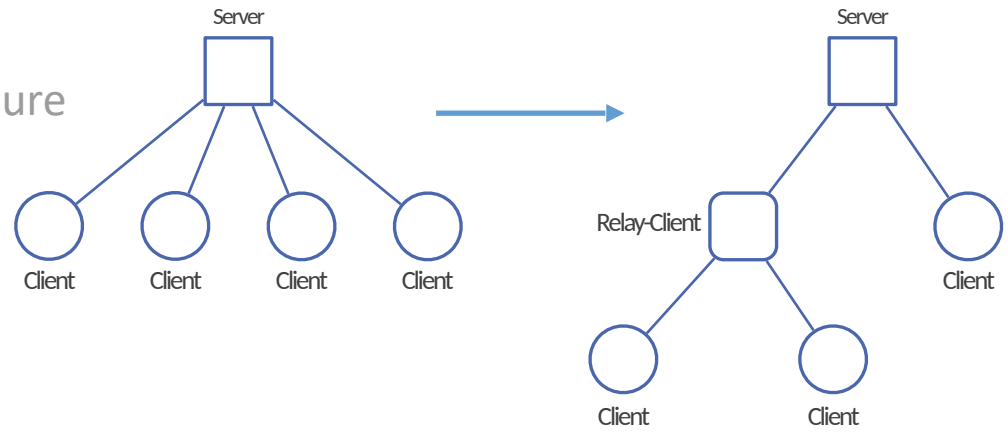
TESTBED-ENTWICKLUNG FÜR LECTURESTUDIO

Fachgebiet Echtzeitsysteme
Fachbereich Elektrotechnik und Informationstechnik
Technische Universität Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Motivation

- The lectureStudio tool
  - Reduced bandwidth requirements
  - Transmitting essential information only during lecture
- A P2P algorithm for lectureStudio
  - Direct distribution among clients
  - Reducing the central server's load
  - Optimizing bandwidth



- Developing a container-based testbed environment for the P2P algorithm

CONTAINERlab

- Performance evaluation of the P2P algorithm

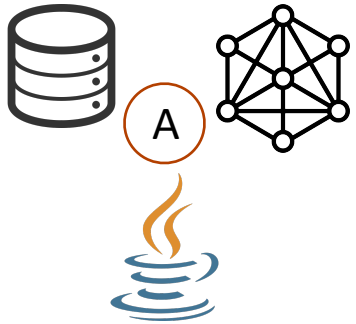
& cAdvisor Prometheus Grafana

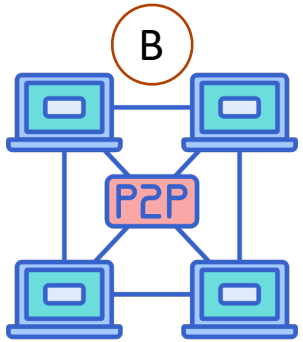
Overall Idea of the Testbed

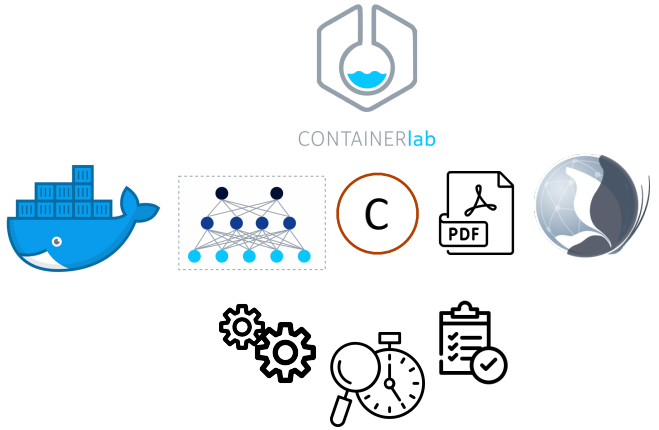
A. Preparing Network Data and Topology for the Participants (lectureStudio server and peers)

B. Calculating the Result Topology by the P2P Algorithm

C. Deploying and Simulating the Network Topology for Data Transfer in the Testbed

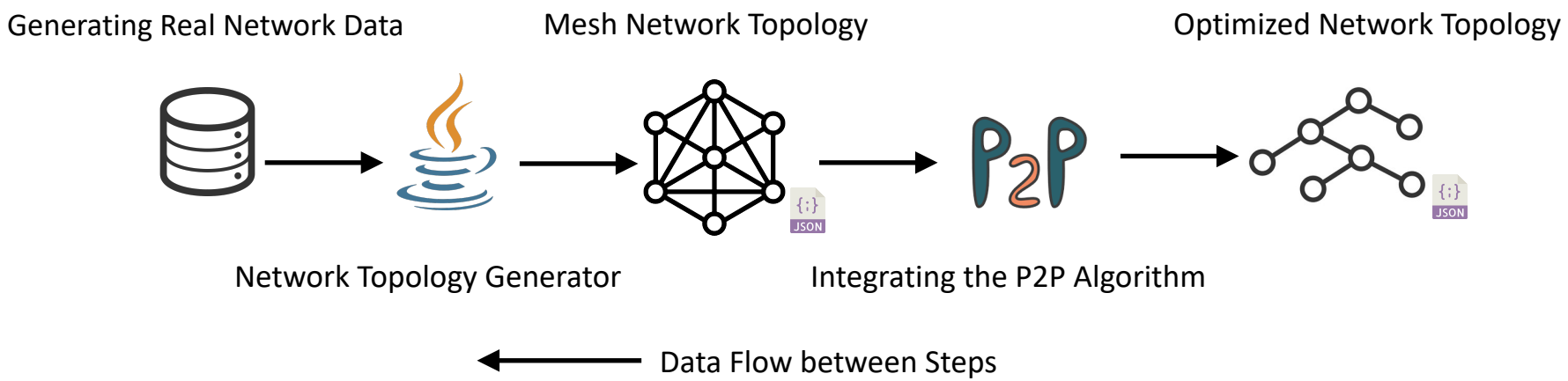






Initial Steps (Not Repeated) of the Testbed

- Generating real network data using normal distribution
- Creating mesh network topology by the testbed
- Integrating the P2P algorithm
  - Calculating optimized network topology by the P2P algorithm
- Implementing the traditional server-client based approach



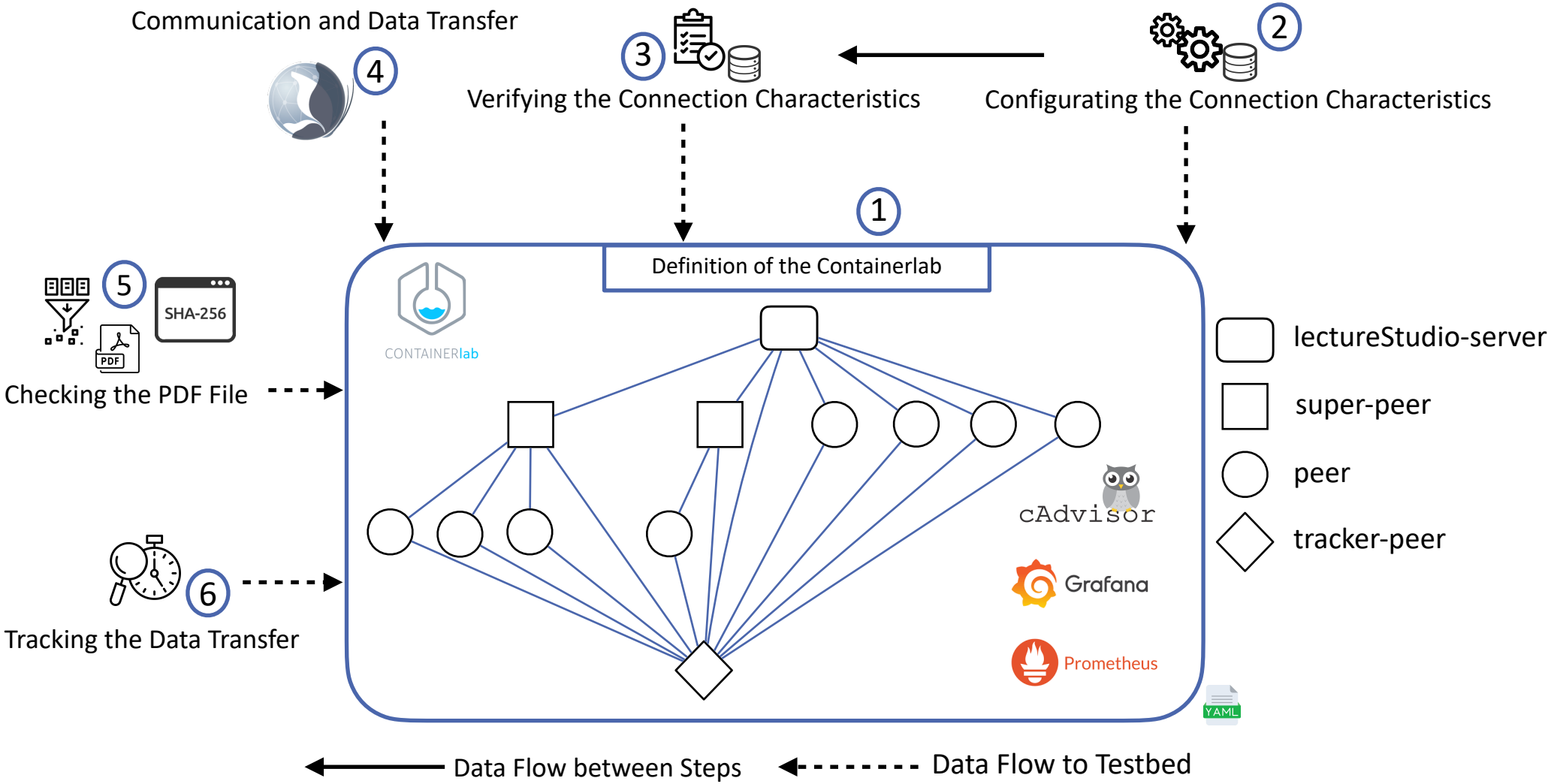
Generating Real Network Data

- Analyzing real network dataset
  - Max download speed, max upload speed
  - Latency
  - Packet loss
- Reading real network data from CSV file
- Generating network data using normal distribution
  - Having mean and standard deviation from UK and DE-based data

Name of Configuration	Mean (μ)	Standard Deviation (σ)
First Configuration	From UK-Based Data	From UK-Based Data
Second Configuration	From DE-Based Data	From UK-Based Data

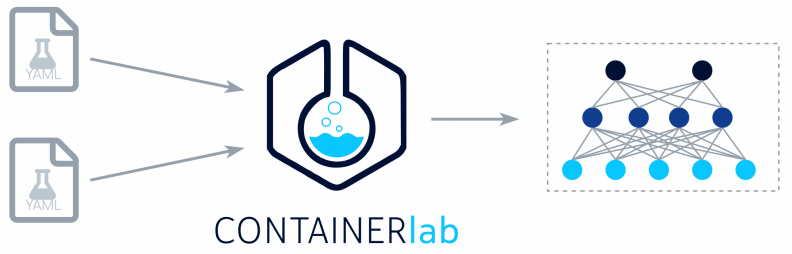
[1], [2]

Execution Steps (Repeated) of the Testbed



Container-Based Testbed Environment

- Docker and Containerlab: efficient, isolated simulation environment
- Creation and management of user-defined network topologies
- Advantages of using Containerlab for the testbed
  - Properties of Containerlab
    - name, image, kind, env, binds, etc.
  - Speed, ease of use, repeatability
  - Creation of complex network topologies



[3]

Configuration of Containerlab File (YAML)

```yaml
name: testbed

topology:
  nodes:
    peer1:
      kind: linux
      image: image-testbed
    peer2:
      kind: linux
      image: image-testbed

  links:
    - endpoints: [peer1:eth1, peer2:eth1]
```

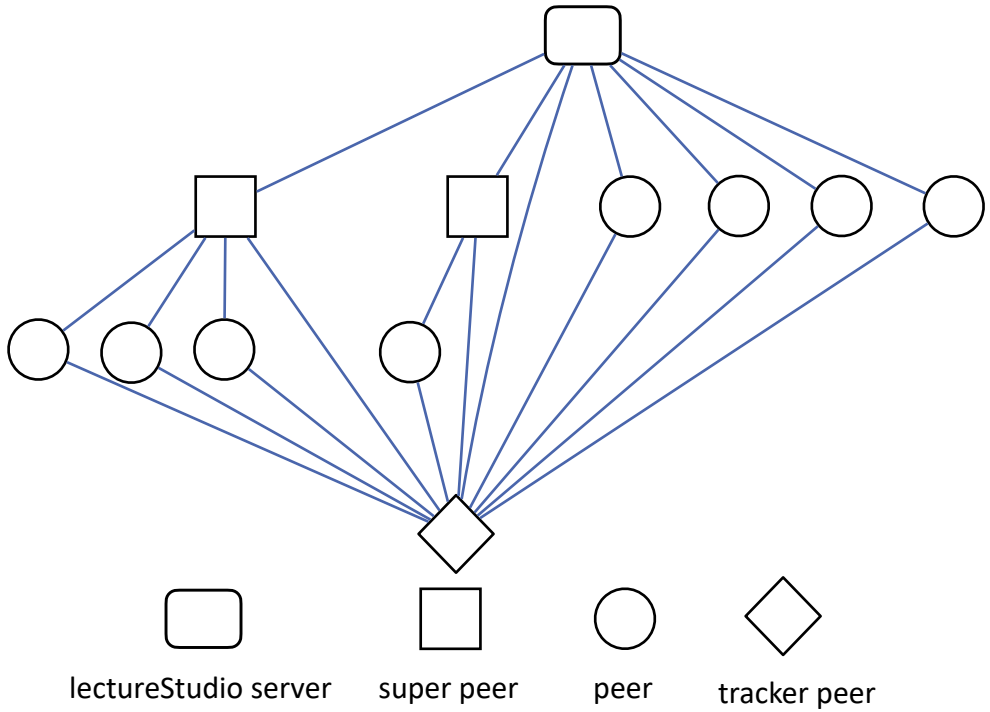
Configuring the Components of the P2P Algorithm ①

- In each test, Containerlab file and all other processes automatically in the testbed

Configuration of Containerlab File
(Network Management, Nodes and Links)

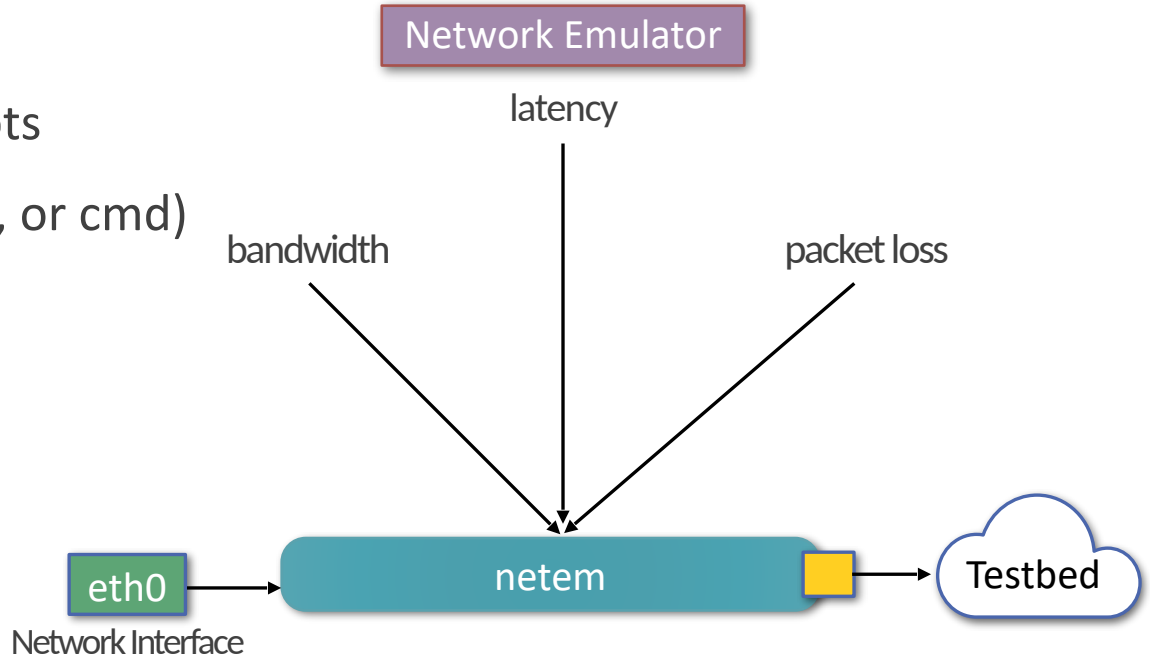
```
name: testbed

mgmt:
  network: fixedips
  ipv4-subnet: 172.100.100.0/24

topology:
   nodes: ...

   links:
     - endpoints: [lectureStudioserver:eth1, 1:eth1]
     - endpoints: [lectureStudioserver:eth2, 2:eth1]
     - endpoints: [lectureStudioserver:eth3, 3:eth1]
     - endpoints: [lectureStudioserver:eth4, 4:eth1]
     - endpoints: [lectureStudioserver:eth5, 5:eth1]
     - endpoints: [lectureStudioserver:eth6, 6:eth1]
     - endpoints: [5:eth2, 7:eth1]
     - endpoints: [6:eth2, 8:eth1]
     - endpoints: [6:eth3, 9:eth1]
     - endpoints: [6:eth4, 10:eth1]
```

Configuration of Nodes



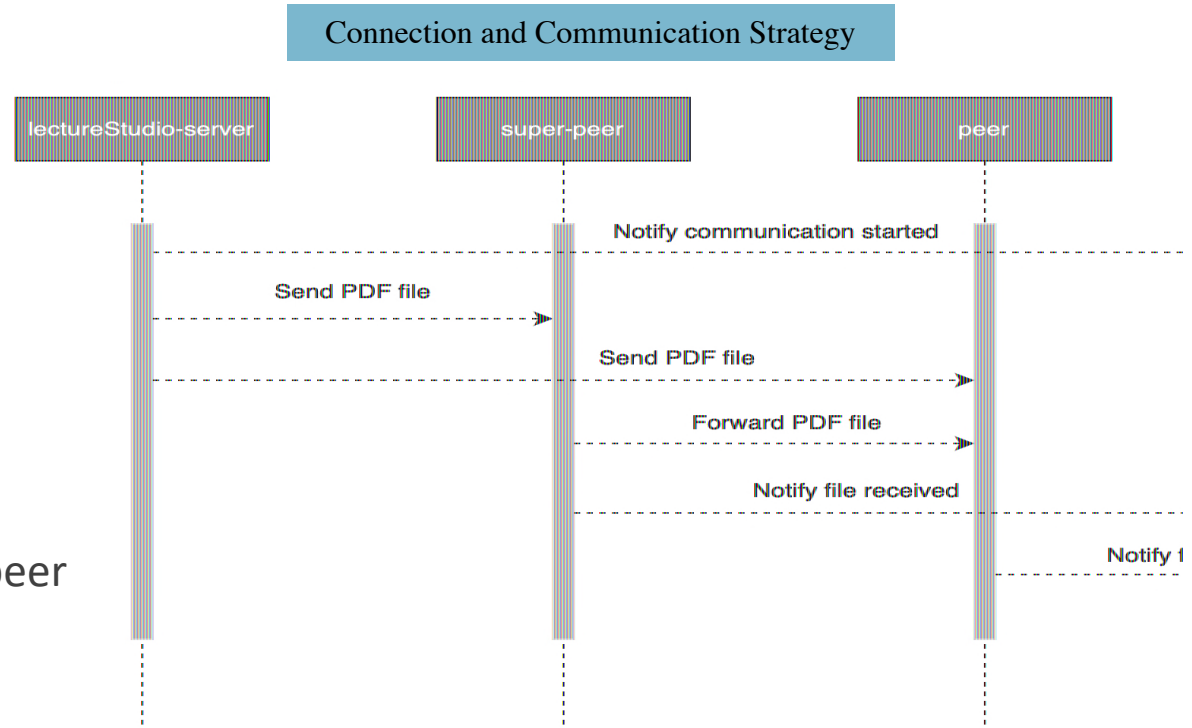
Configuring and Verifying the Network Characteristics ② ③

- Configuring the connection characteristics using traffic control commands
  - Bandwidth limitation
  - Latency addition
  - Packet loss simulation
- Configuring these characteristics with scripts
  - Properties of Containerlab (exec, binds, or cmd)
- Verifying the connection characteristics
  - Using tools like ping, iperf3



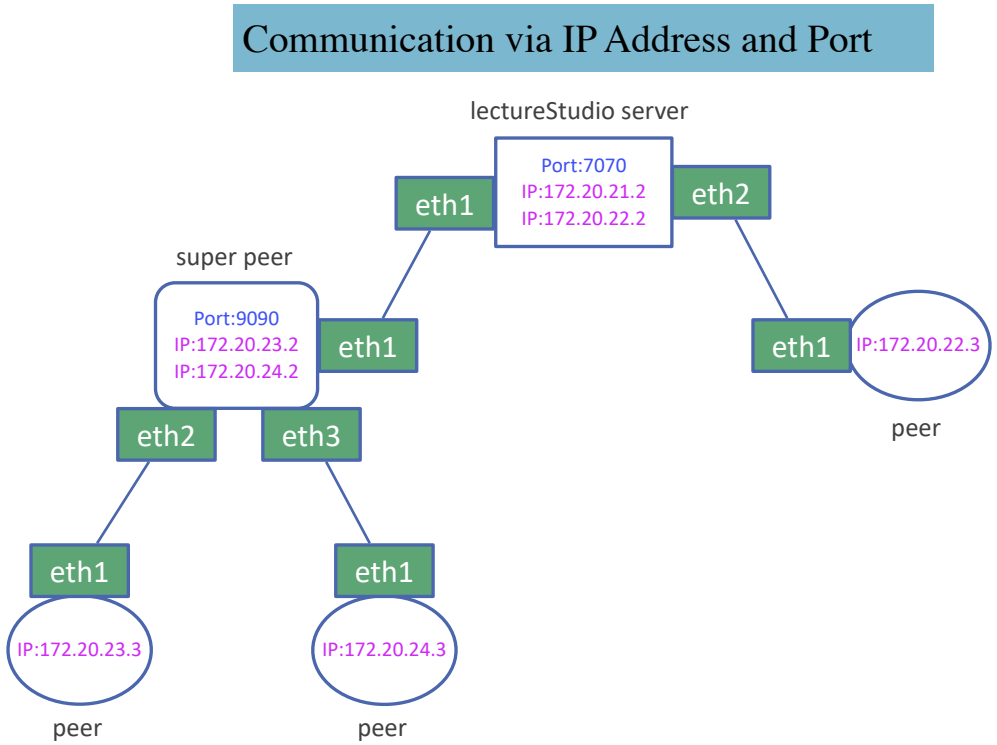
Communication and Data Transfer Processes ④ ⑤ ⑥

- Initial notification by the lectureStudio server
  - Notification to tracker peer
  - Start of data transfer process
- Role of super peers
  - Reception and forwarding of PDF file
  - Transition from receiver to sender
- Confirmation messages
  - From peers and super peers
- Calculation of data transfer duration by tracker peer
  - Counting received confirmations
  - Total duration calculation
- Integrity checks of PDFs with hash value calculation



Establishing P2P Connections

- Addition node infos in the testbed setup
  - Port number
  - IP address
- Using Netty framework for server
  - Handshaking
  - Authenticating peers
  - Establishing connections
  - Preparing the network for data transfer
- Monitoring process and performance

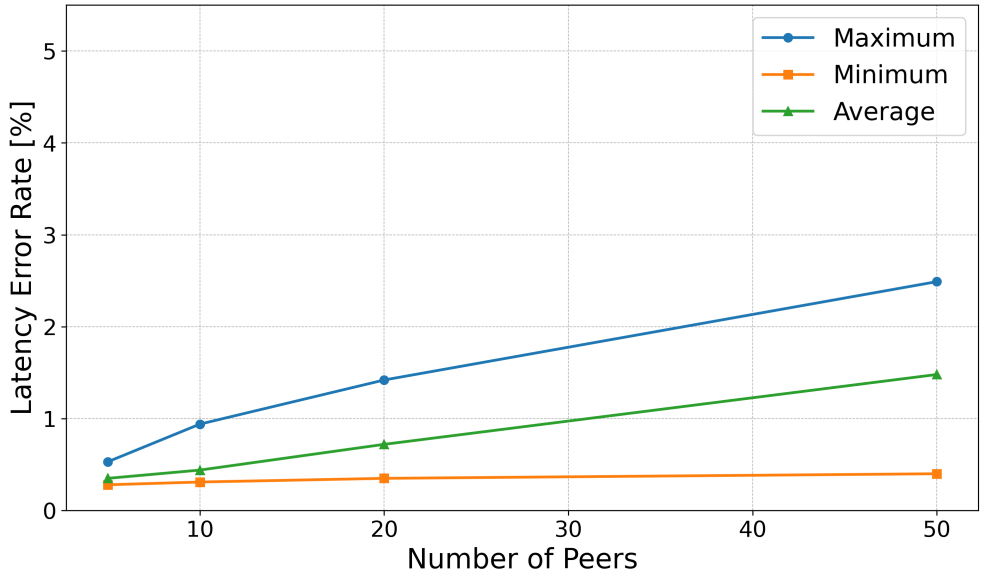


Accuracy of the Testbed, RQ1.1 (1)

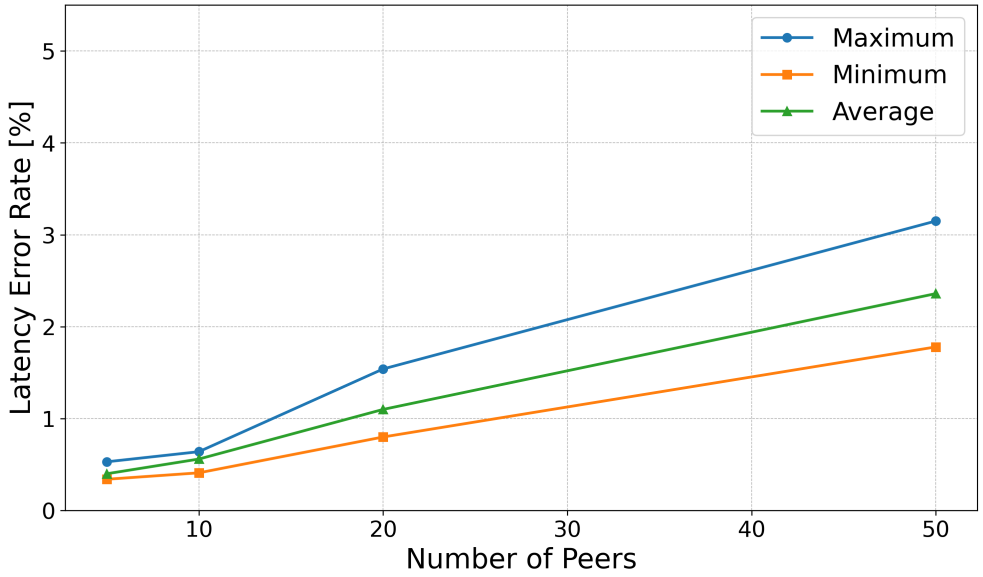
RQ1.1 How well does the testbed simulate the configured bandwidth, latency, and packet loss?

- Discrepancy between configured and measured values
- More nodes result in reduced bandwidth allocation, leading to increased latency
- High CPU and memory usage, affecting network performance and latency

Error Rate for Latency with the P2P Algorithm



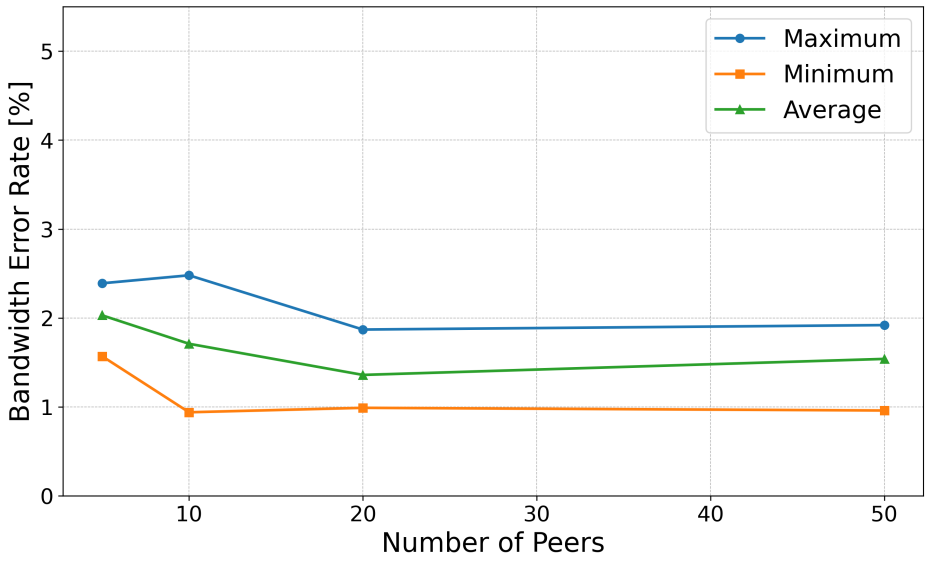
Error Rate for Latency without the P2P Algorithm



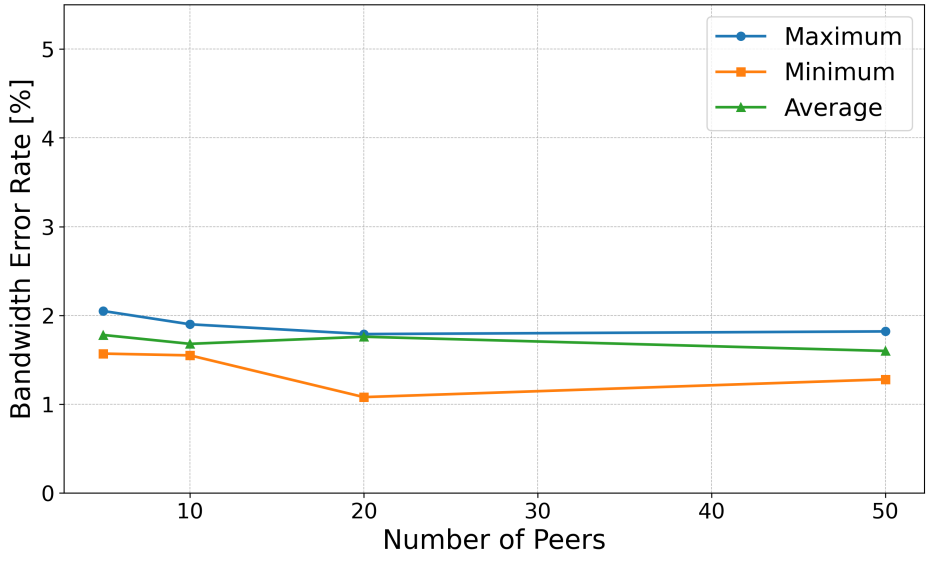
Accuracy of the Testbed, RQ1.1 (2)

- Accuracy of measuring tools
  - Iperf accuracy results variation from actual performance
  - Variables like network conditions, configuration, system overhead

Error Rate for Bandwidth with the P2P Algorithm



Error Rate for Bandwidth without the P2P Algorithm



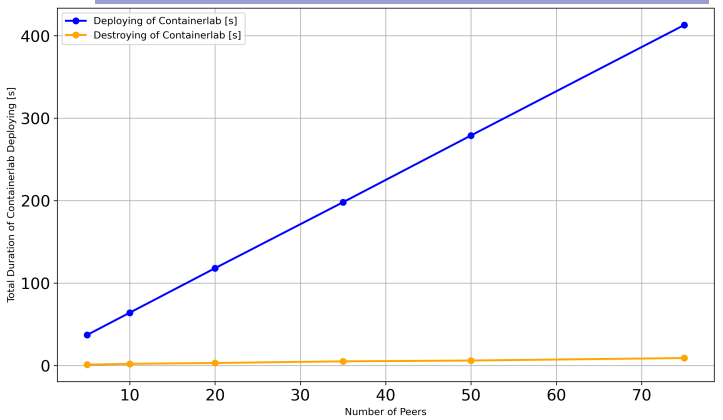
- Testbed configured with packet loss values from real network data (%0.001)
  - Noted limitation in accurately measuring packet loss
  - Increase of latency with observed packet loss

Testbed Scaling and Resource Utilization, RQ1.2

RQ1.2 How well does the testbed scale with more nodes and complex topologies affect the host in terms of resource utilization?

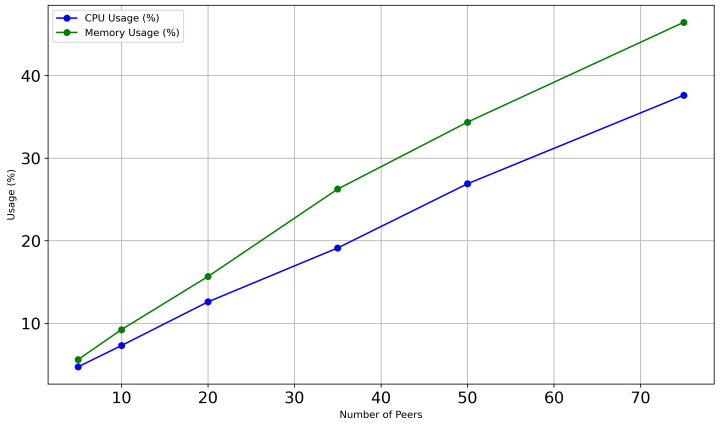
- Scaling peers with Containerlab
  - Deployment time increase of 5.37s per additional node
  - Destroying time low, approximately 9s for 75 nodes

Time of Deploying and Destroying of Containerlab



- Utilization of CPU and memory usage with more peers
  - CPU and memory usage increase linearly, showing scalable performance
  - Memory usage growth suggests potential bottleneck with more peers

CPU and Memory Usage of the Host

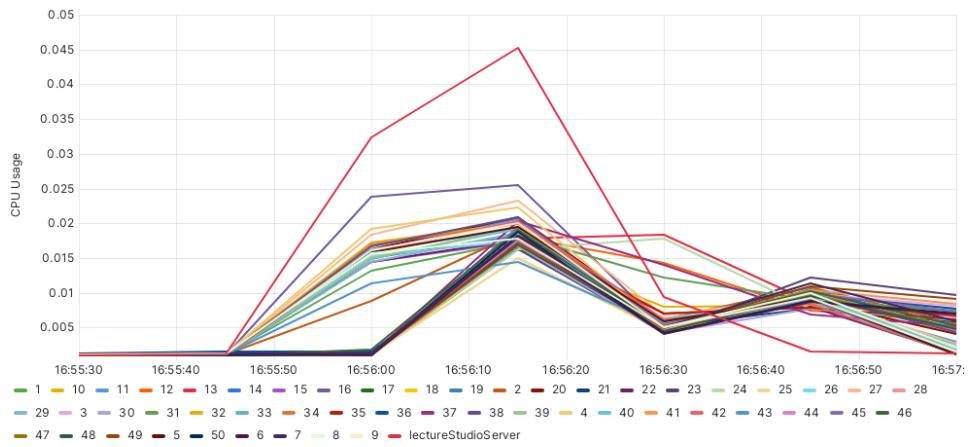


CPU and Memory Usage Analysis of Nodes, RQ2.1 (1)

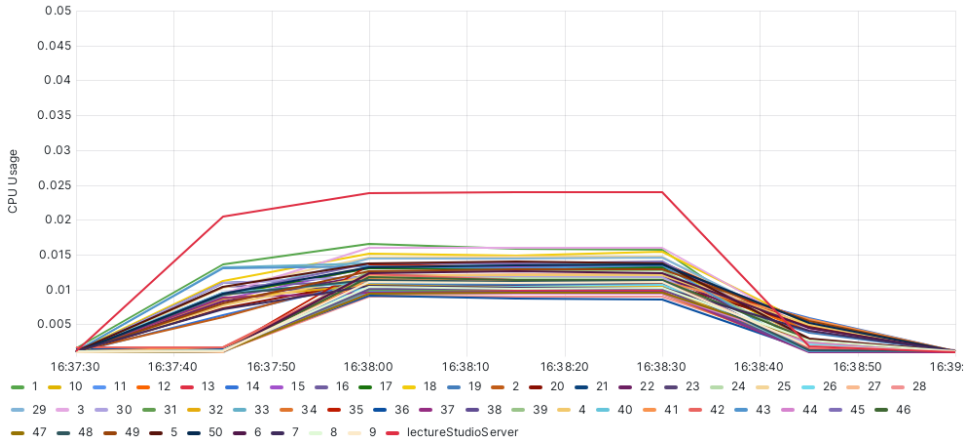
RQ2.1 How do CPU and memory usage change of the participants (the lectureStudio server and peers) in tests with and without the P2P algorithm?

- CPU Usage evaluation with and without the P2P algorithm in the testbed
  - Direct data transfer, increased CPU usage on the lectureStudio server
  - P2P algorithm, data distribution leading to decreased CPU usage on the lectureStudio server
  - Peak CPU usage reduction of 47% by the P2P algorithm

CPU Usage without the P2P Algorithm



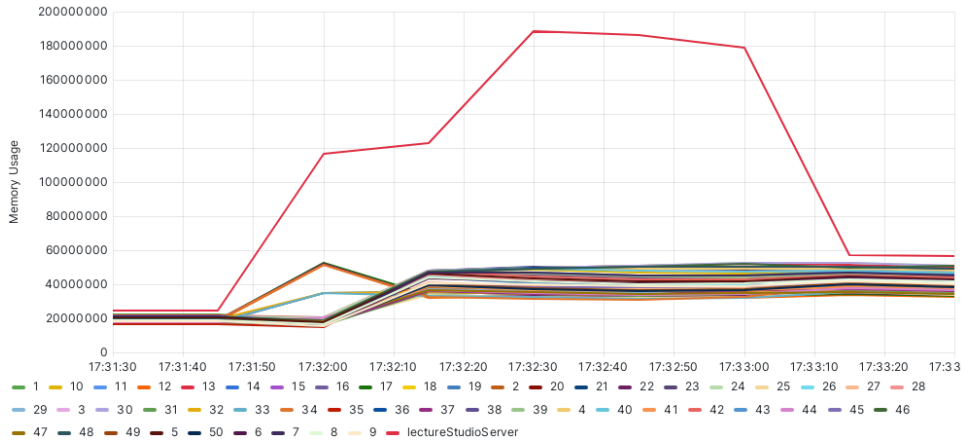
CPU Usage with the P2P Algorithm



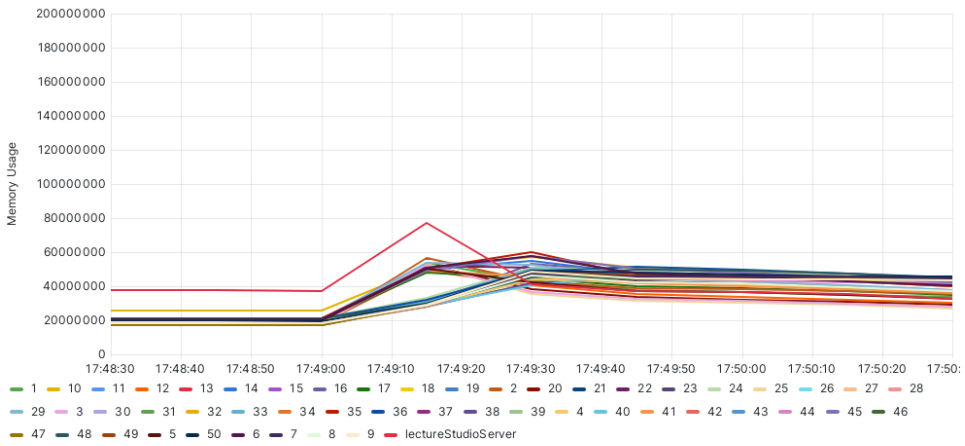
CPU and Memory Usage Analysis of Nodes, RQ2.1 (2)

- Memory usage evaluation with and without the P2P algorithm in the testbed
  - Direct data transfer, increased memory usage on the lectureStudio server
  - P2P algorithm, data distribution leading to decreased memory usage on the lectureStudio server
  - Peak memory usage reduction of 58% by the P2P algorithm

Memory Usage without the P2P Algorithm



Memory Usage with the P2P Algorithm



- Manageable increase in CPU and memory usage of super peers handling data flow

Performance Evaluation of the P2P Algorithm, RQ2.2 and RQ2.3 (1)

RQ2.2 How does the P2P algorithm react to changing network characteristics (bandwidth, latency, packet loss)?

RQ2.3 Overall, is the P2P algorithm efficient for data transfer? How does the total duration obtained by the P2P algorithm respond to the changing number of peers and data size?

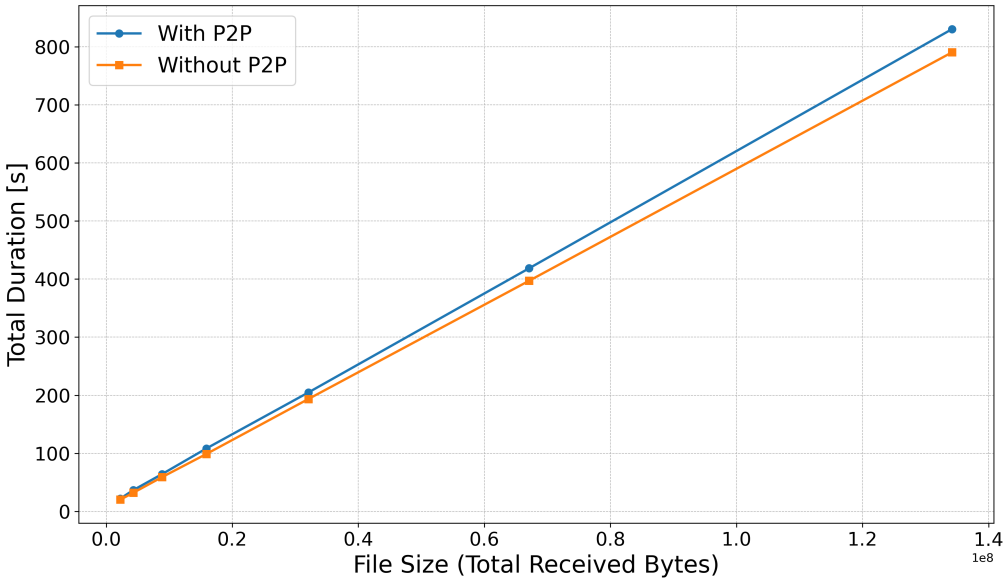
- Test duration measurement from first to last acknowledgment message
  - Over 1000 tests performed with two configurations
  - Variation in data size or number of peers
- 1.Configuration:
  - Normal distribution with UK-based mean and standard deviation
  - Minimal difference between the P2P algorithm and server-client approach
- 2.Configuration:
  - Normal distribution with DE-based mean and UK-based standard deviation
  - P2P algorithm effective with increasing peers numbers/data size

From	Download Speed	Upload Speed
UK-Based	53 Mbps	5 Mbps
DE-Based	90 Mbps	30 Mbps

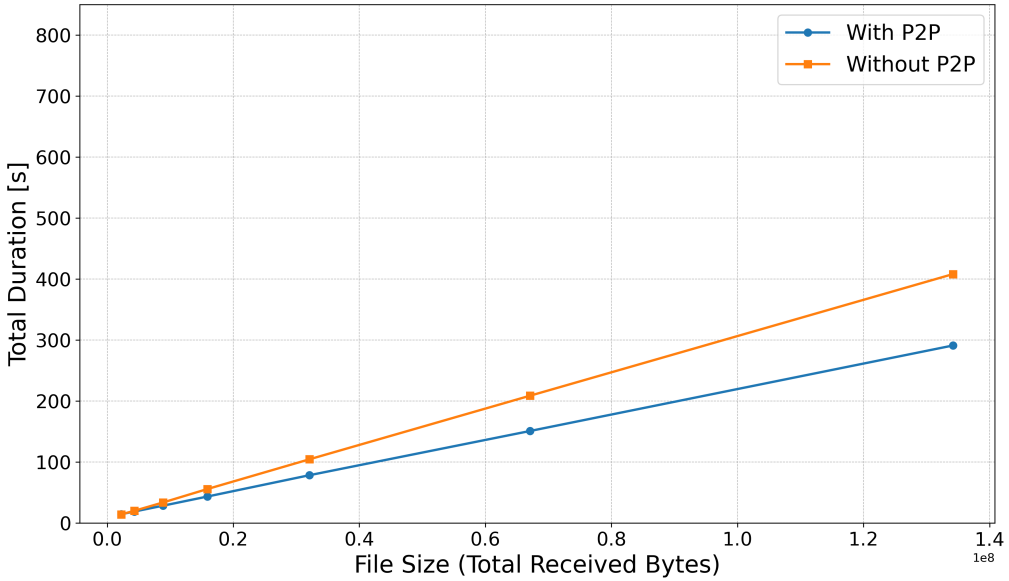
Performance Evaluation of the P2P Algorithm, RQ2.2 and RQ2.3 (2)

- First configuration with **the lectureStudio server and 20 peers**
  - The P2P algorithm performance little worse to the traditional approach
- Second configuration with **the lectureStudio server and 20 peers**
  - Transferring 2MB data size, the P2P algorithm performance nearly identical to the traditional approach
  - Transferring 128MB data size, the P2P algorithm performance 27% faster than the traditional approach

Time of Whole Data Transfer with the First Configuration



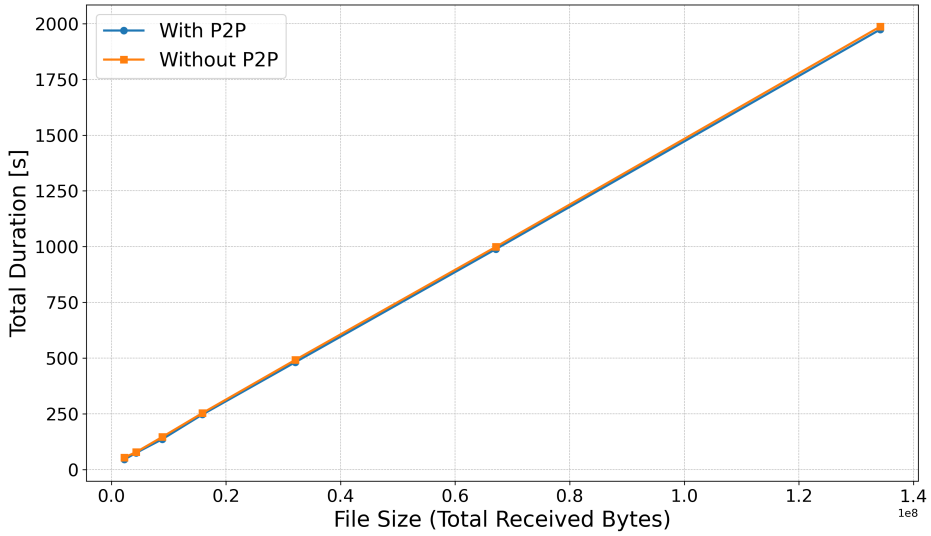
Time of Whole Data Transfer with the Second Configuration



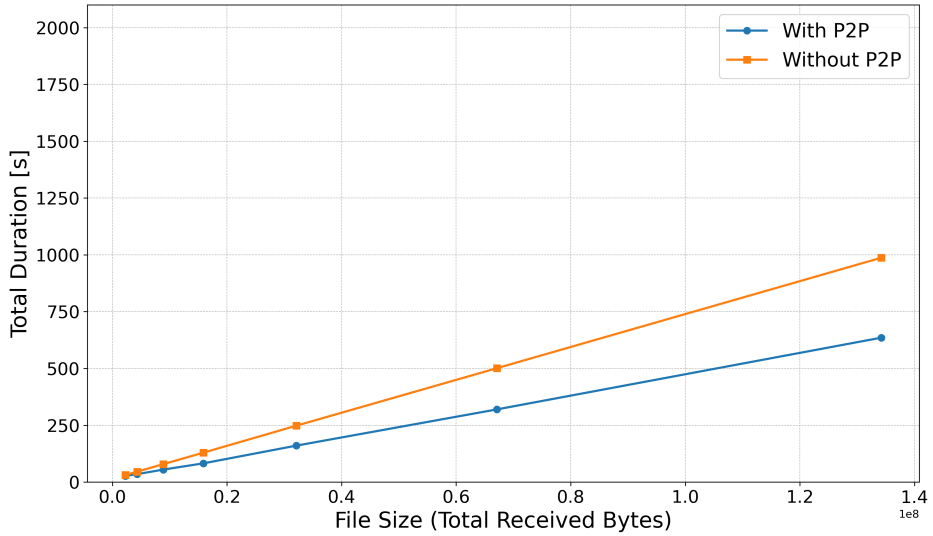
Performance Evaluation of the P2P Algorithm, RQ2.2 and RQ2.3 (3)

- First configuration with **the lectureStudio server** and **50 peers**
  - Transferring 2MB data size, the P2P algorithm performance nearly identical to the traditional approach
  - Transferring 128MB data size, the P2P algorithm performance 5% faster than the traditional approach
- Second configuration with **the lectureStudio server** and **50 peers**
  - Transferring 2MB data size, the P2P algorithm performance 12% faster than the traditional approach
  - Transferring 128MB data size, the P2P algorithm performance 35% faster than the traditional approach

Time of Whole Data Transfer with the First Configuration



Time of Whole Data Transfer with the Second Configuration



Conclusion and Future Work

- Goal: Developing a testbed for the P2P data distribution algorithm
- Results
  - Generating network topology: Normal distribution with data from the UK and DE
  - Creating the testbed: Docker and Containerlab
  - Deploying network topology: Data transfer between lectureStudio server and peers
  - Evaluating the testbed: Accuracy and scalability performance
  - Evaluating the P2P algorithm: Resource utilization and total time performance
- Future Works
  - Enhancement of packet loss simulation for accurate network behavior
  - Automatic integration between the testbed and the P2P algorithm(s) optimization
  - Development of a graphical testbed interface for easier configuration and real-time analysis

Thank you for your attention!

ANY QUESTIONS?

References

[1] https://www.data.gov.uk/dataset/dfe843da-06ca-4680-9ba0-fbb27319e402/uk-fixed-line-broadband-performance

[2] https://www.speedtest.net/

[3] https://containerlab.dev/manual/topo-def-file/

Additional Slides (1) - Network Topology

- Generating network topology by the testbed
  - Listing nodes with network characteristics
  - Detailing connections between the lectureStudio server and all peers
- Optimizing network topology by the P2P algorithm
  - Identifying super peers
  - Optimizing connections between the lectureStudio server and peers

Network Topology Generated by the Testbed

```
{
  "filename": "test.pdf",
  "filesize": 5000,
  "peers": [
    {
      "name": "lectureStudioServer",
      "maxDownload": 29150,
      "maxUpload": 9209
    },
    {
      "name": "1",
      "maxDownload": 1080,
      "maxUpload": 373
    }
  ]
}
```

```
"connections": [
    {
      "sourceName": "lectureStudioServer",
      "targetName": "1",
      "bandwidth": 1080,
      "latency": 57,
      "loss": 0.0035
    },
    {
      "sourceName": "1",
      "targetName": "lectureStudioServer",
      "bandwidth": 373,
      "latency": 57.15,
      "loss": 0.0035
    }
  ]
}
```

{:} JSON

Network Topology Optimized by the P2P Algorithm

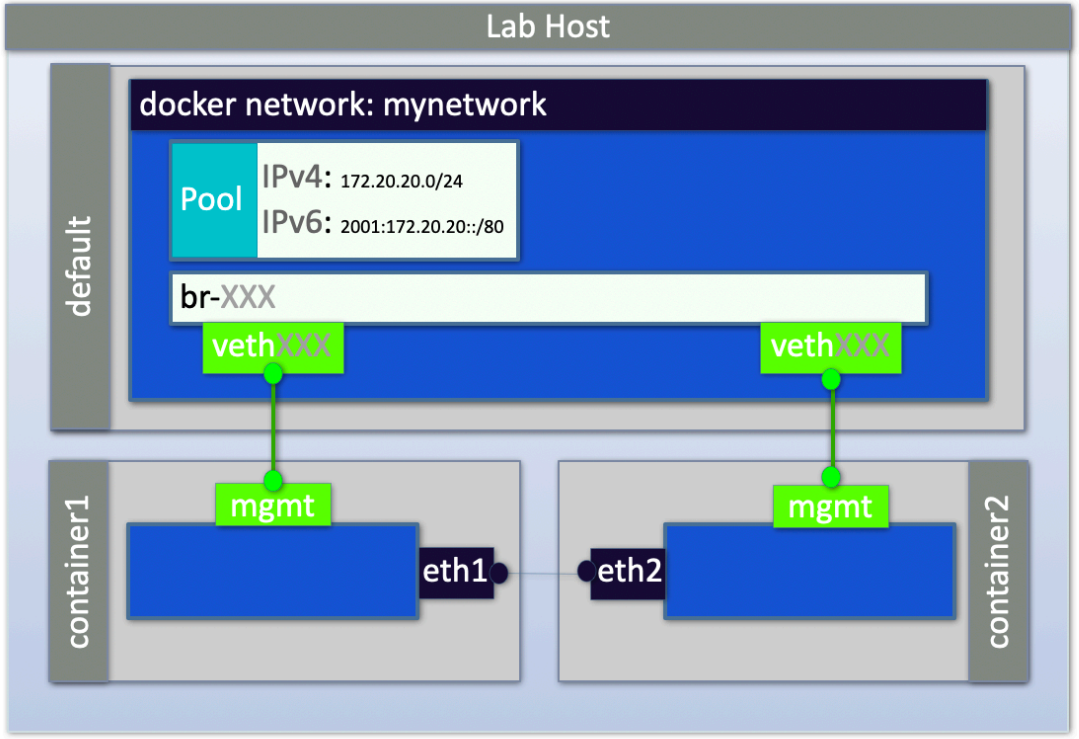
```
{
  "superpeers": [
    {
      "name": "1"
    }
  ],
  "peer2peer": [
    {
      "sourceName": "lectureStudioServer",
      "targetName": "1"
    },
    {
      "sourceName": "1",
      "targetName": "2"
    }
  ]
}
```

{:} JSON

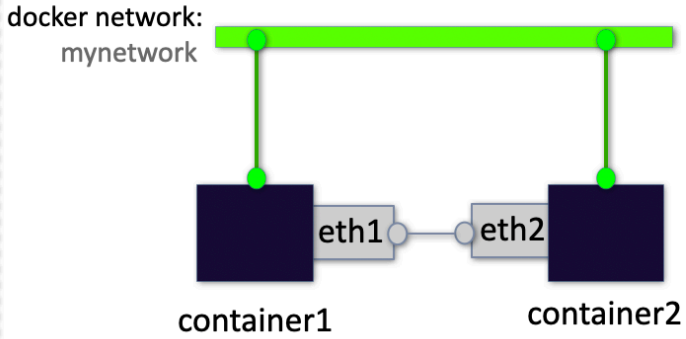
Additional Slides (2) - Challenges

- Finding real network dataset
- Configuration and validation of the network characteristics for connections
  - Bandwidth limitation
  - Latency addition
  - Packet loss simulation
- Data transmission between the lectureStudio server and peers (or super peers)
- Synchronisation problem of total time
- Monitoring by Grafana, Prometheus, and cAdvisor

Additional Slides (3) - Network Wiring Concepts in Containerlab



Lab Wiring Diagram



Lab Logical Diagram