

Master-Thesis Presentation by Özcan Karaca

# Testbed-Development for lectureStudio

---

TESTBED-ENTWICKLUNG FÜR LECTURESTUDIO

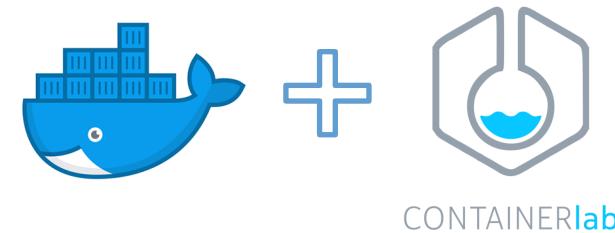
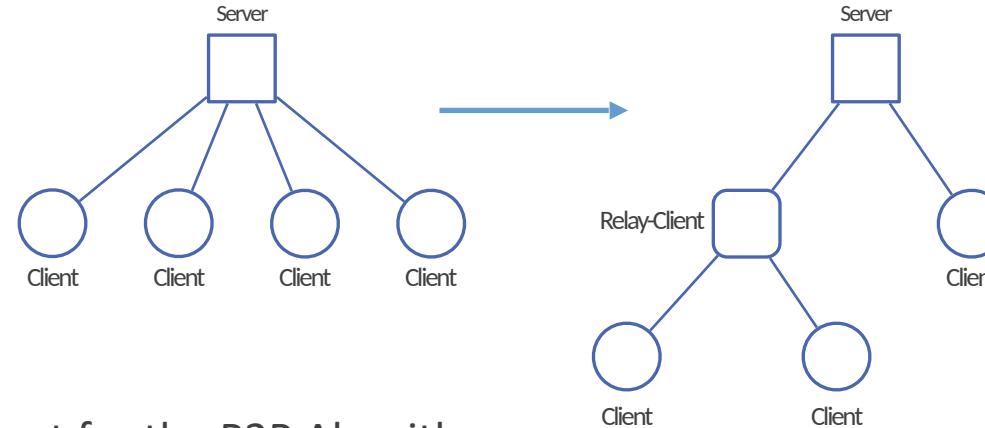
Fachgebiet Echtzeitsysteme  
Fachbereich Elektrotechnik und  
Informationstechnik  
Technische Universität Darmstadt



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

# Motivation

- 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
  - Simulation of Real Network Data
  - Configuration and Validation of the Network Characteristics
  - Communication and Data Transfer between Nodes
- Performance Evaluation of the P2P Algorithm
  - Analysis of Resource Consumption
  - Analysis of Total Duration



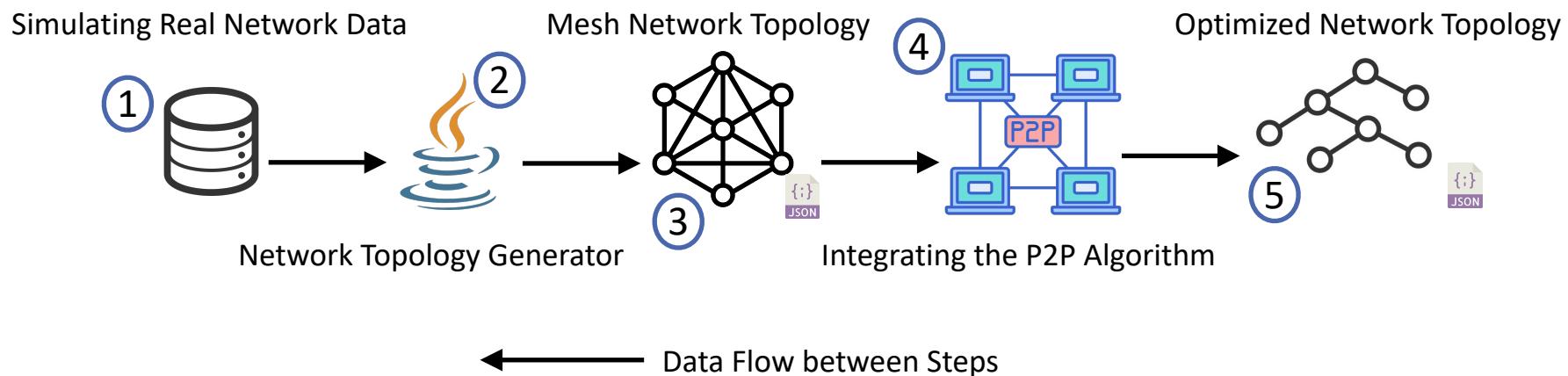
# Task Description

---

- Finding Real Network Data
  - Maximum Upload Speed, Maximum Download Speed, Latency, and Packet Loss
- Simulating Real Network Data Using Normal Distribution
- Generating Network Topology
- Integrating the P2P Algorithm
- Configuring the Components of the P2P Algorithm in the Testbed
  - Creating a Containerlab File and Configuring Network Management, Nodes and Links
- Implementing and Validating the Network Characteristics of the Connections
  - Bandwidth, Latency, and Packet Loss
- Managing Communication and Data Transfer
  - Tracking and Validating the Data Transfer Process
- Evaluating of the Testbed and the P2P Algorithm Performance

# Initial Steps (Not Repeated) of the Testbed

- Simulating Real Network Data Using Normal Distribution
- Generating Mesh Network Topology
  - Integrating the P2P Algorithm
  - Implementing the Traditional Server-Client Based Approach
- Calculating Optimized Network Topology with the P2P Algorithm



# Simulating 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 Germany Based Data

Name of Configuration	Mean ( $\mu$ )	Standard Deviation ( $\sigma$ )
First Configuration	From UK-Based Data	From UK-Based Data
Second Configuration	From Ger-Based Data	From UK-Based Data

[1], [2]

# 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 Studio 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  
    }  
  ]  
}
```

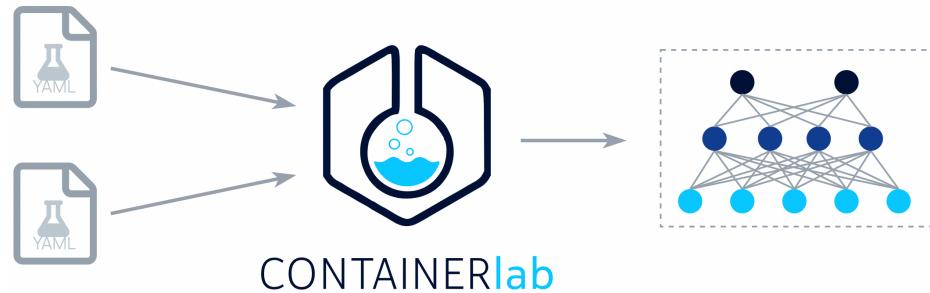
Network Topology Optimized by the P2P Algorithm

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



# 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)

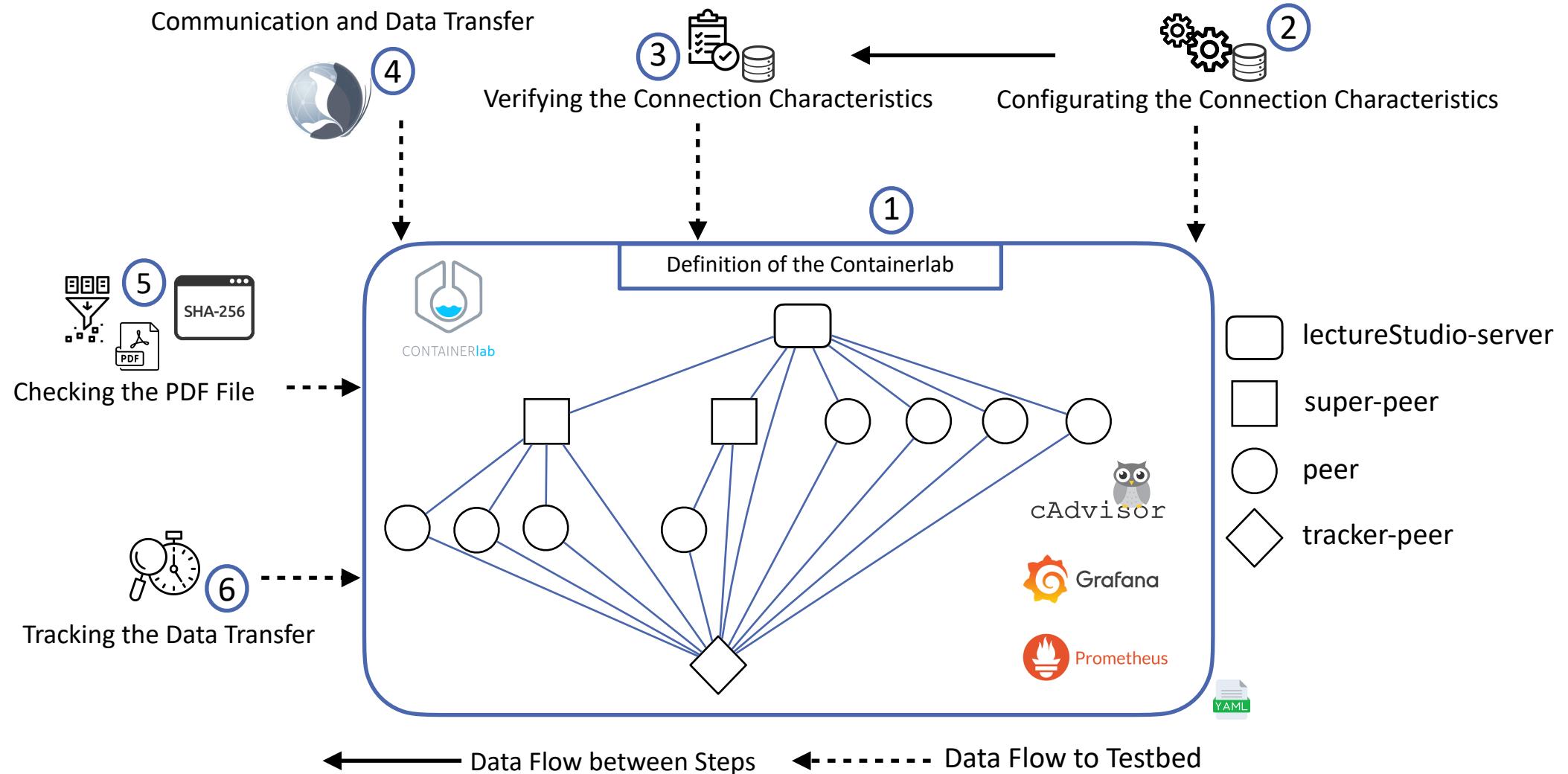
```
name: testbed

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

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



# Execution Steps (Repeated) of the Testbed



# Configuring the Components of the P2P Algorithm

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

name: testbed

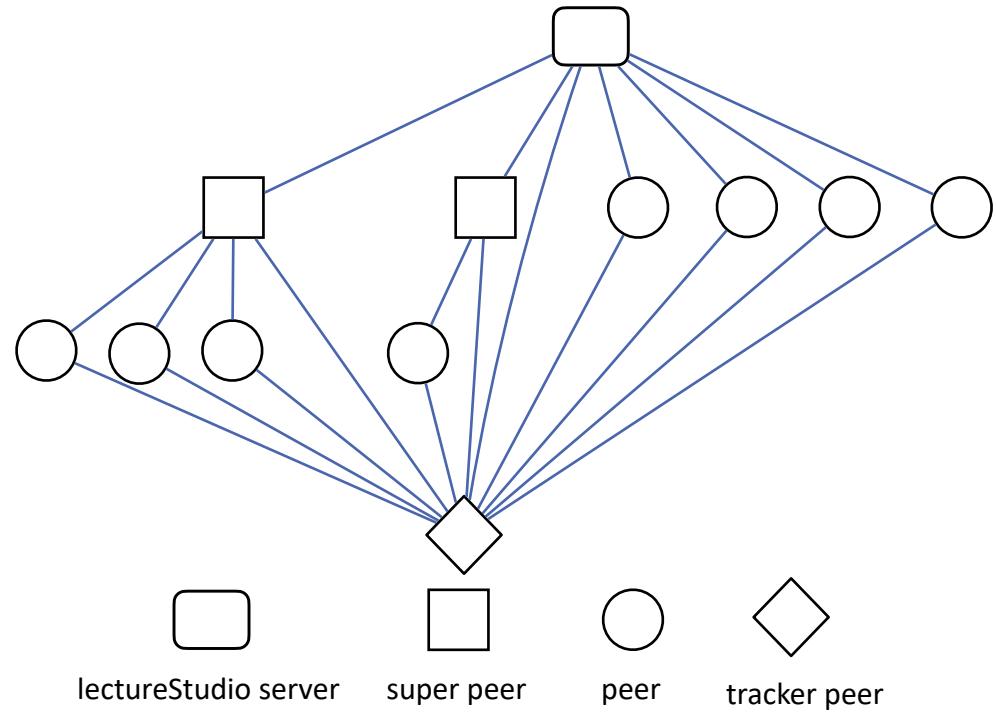
mgmt:  
network: fixedips  
ipv4-subnet: 172.100.100.0/24

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]

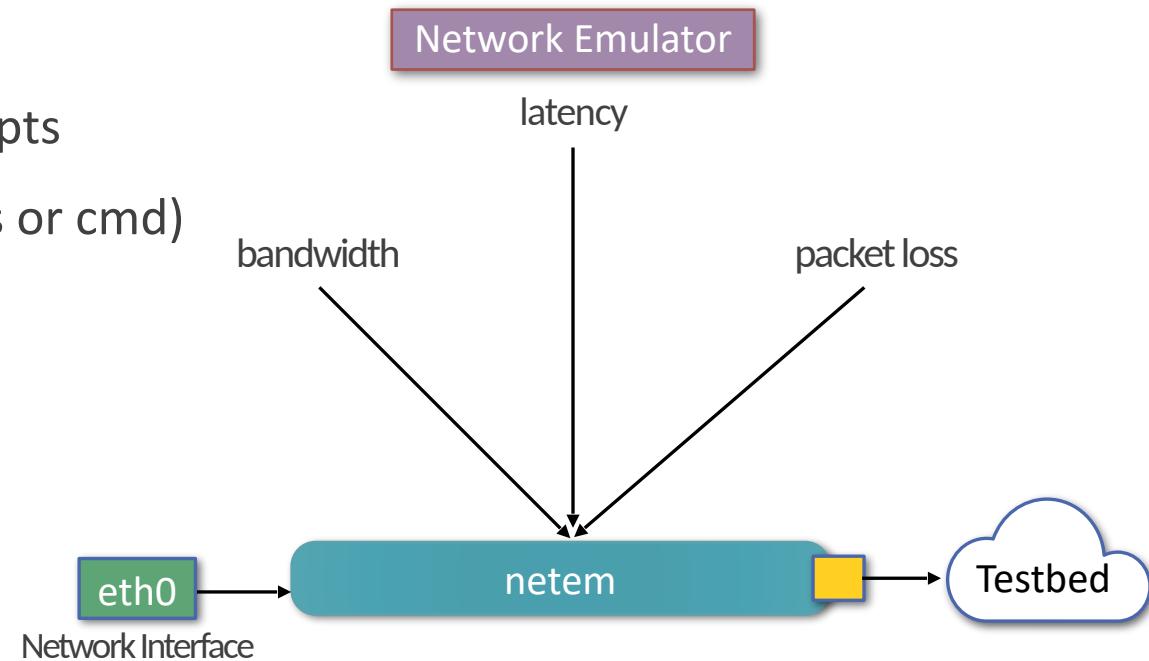


Configuration of Nodes



# Configuring and Verifying the Network Characteristics

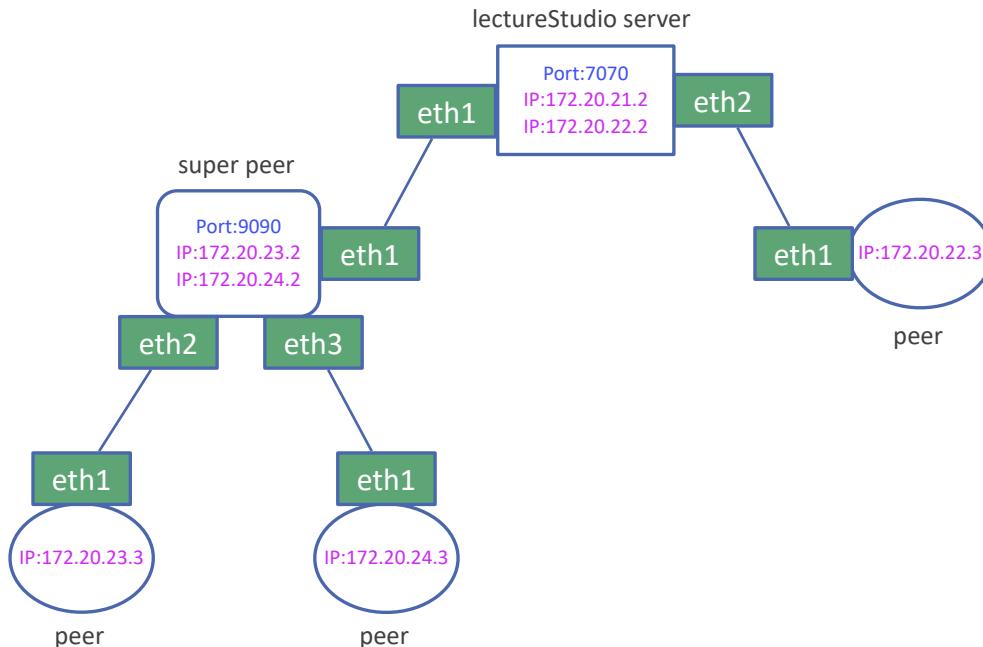
- Configuring the Connection Characteristics Using Traffic Control Commands
  - Bandwidth Limitation
  - Latency Addition
  - Packet Loss Simulation
- Configuring these Characteristics with Scripts
  - Properties of Containerlab (exec, binds or cmd)
- Verifying the Connection Characteristics
  - Using Tools like ping, iperf3



# Connection Strategy Among Peers

- Add Node Info in Testbed Setup
  - Port Number
  - IP Address
- Use Netty Framework for Server
  - Handshaking
  - Authenticating Peers
  - Establishing Connections
  - Preparing the Network for Data Transfer
- Monitor Process and Performance

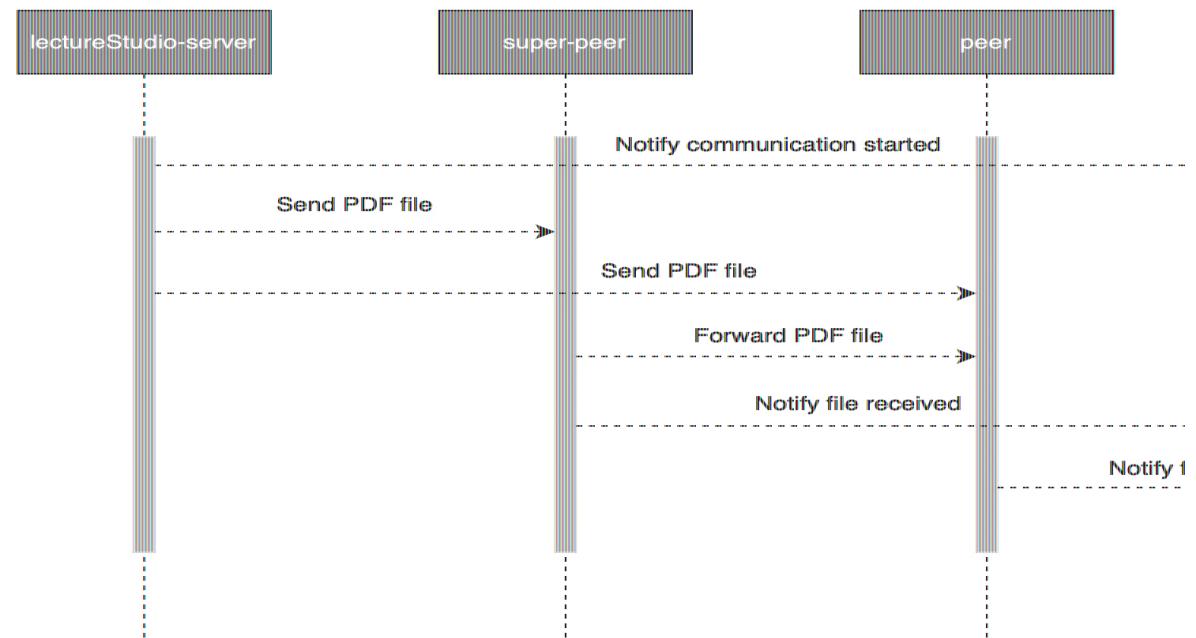
Communication via IP Address and Port



# Communication and Data Transfer Processes

- 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
  - Counting Received Confirmations
  - Total Duration Calculation
- Integrity Checks of PDFs with Hash Value Calculation

Connection and Communication Strategy



# Evaluation Research Questions

---

## Testbed

**RQ1.1** How Accurately Does the Testbed Measure the Configured Bandwidth, Latency and Packet Loss?

**RQ1.2** How Well Does the Testbed Scale with More Nodes and Complex Topologies Affects The Host in Terms of Resource Utilization?

## P2P Algorithm

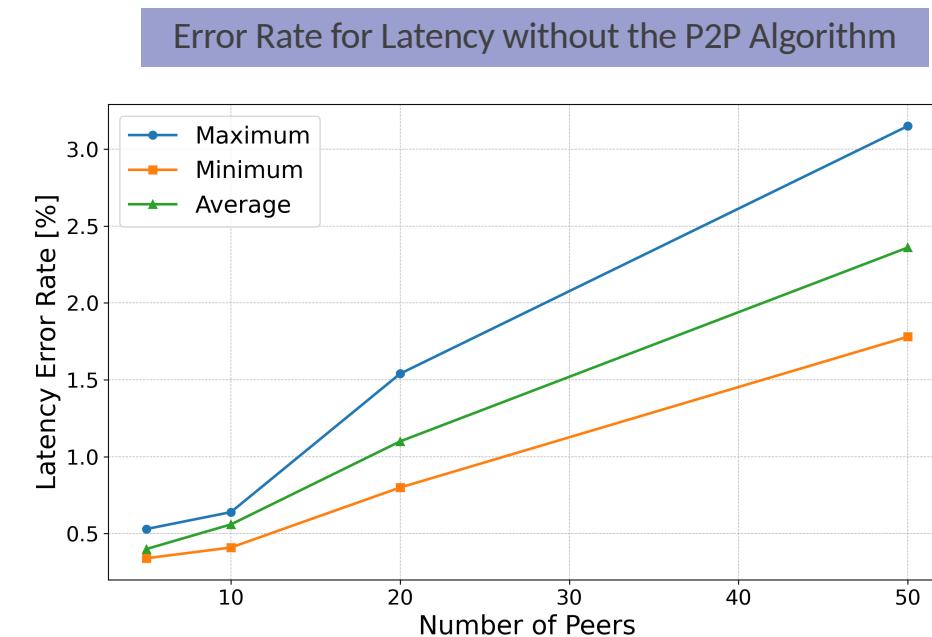
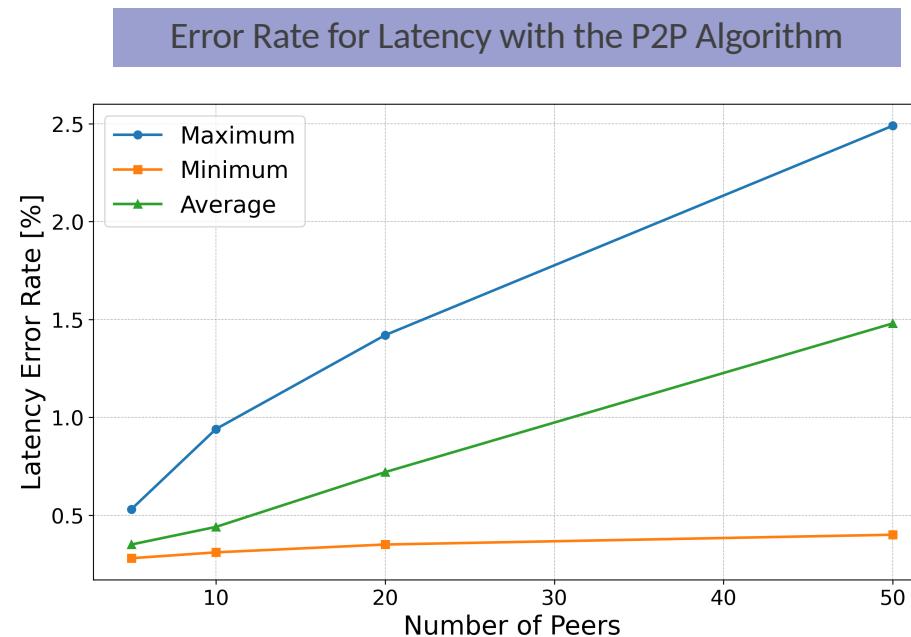
**RQ2.1** How Do CPU and Memory Usage Change of the Participants (the LectureStudio Server and Peers) in Tests with and without the P2P Algorithm?

**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 with the P2P Algorithm Respond to the Changing Number of Peers and Data Size?

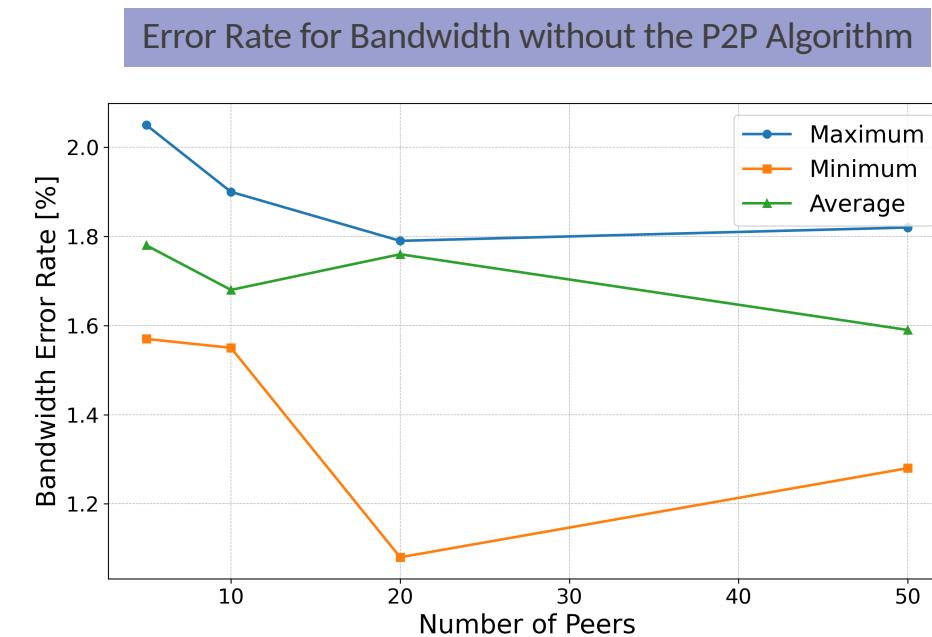
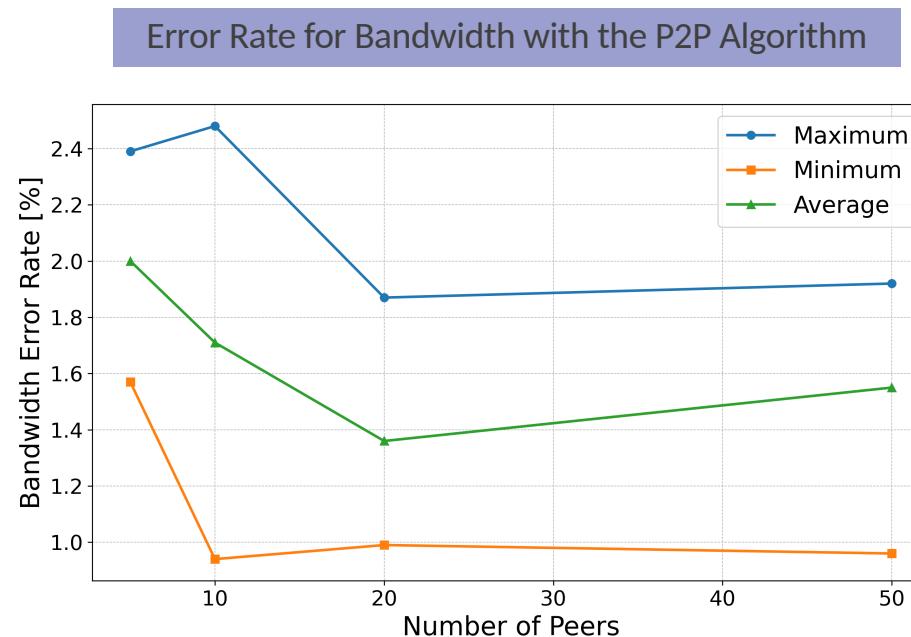
# Accuracy of the Testbed, RQ1.1 (1)

- Discrepancy between Desired and Measured Values
- More Nodes Decrease Bandwidth Allocation Per Connection
  - A Reduction in Bandwidth Corresponds to a Rise in Latency
- High CPU and Memory Usage Impacting Network Performance and Latency



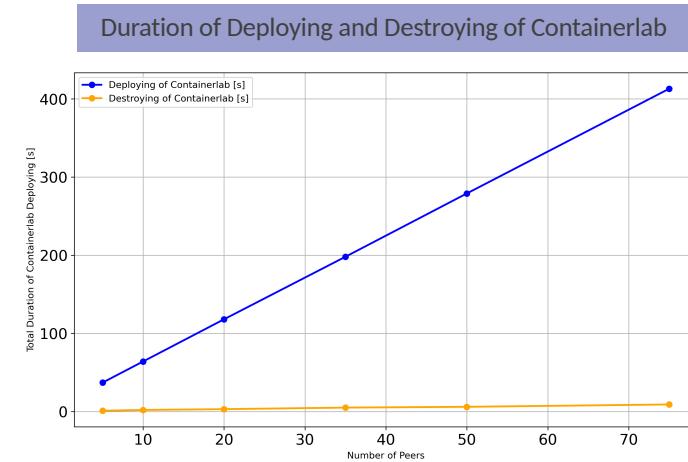
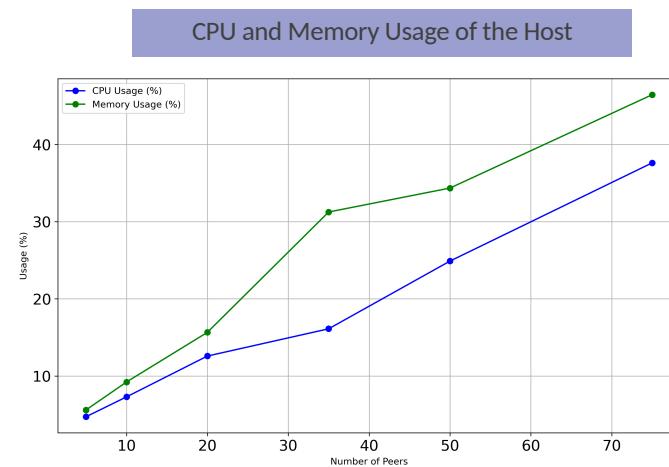
# Accuracy of the Testbed, RQ1.1 (2)

- Accuracy of Measuring Tools
  - Iperf Accuracy Results Variation from Actual Performance
  - Variables Like Network Conditions, Configuration, System Overhead



# Testbed Scaling and Resource Utilization, RQ1.2

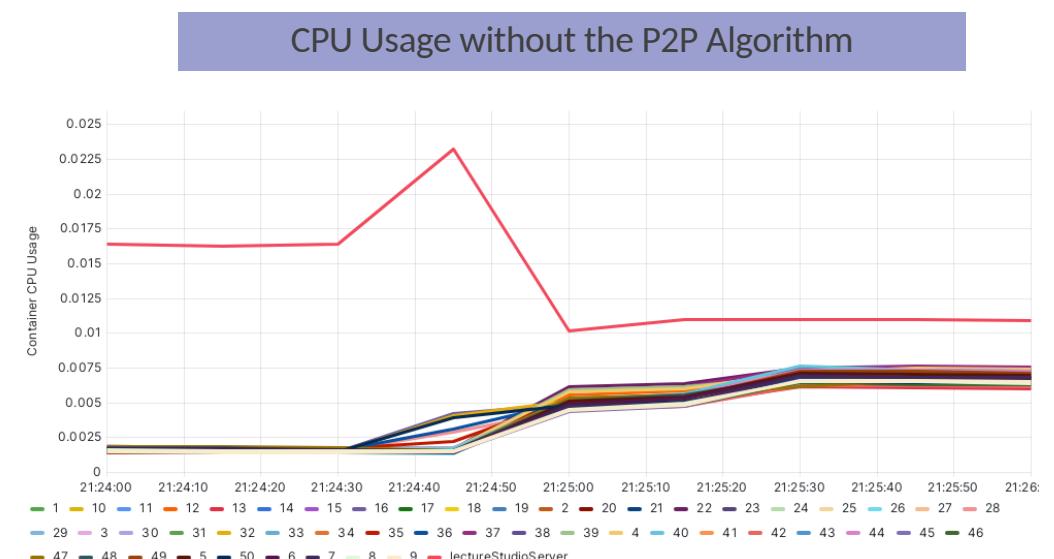
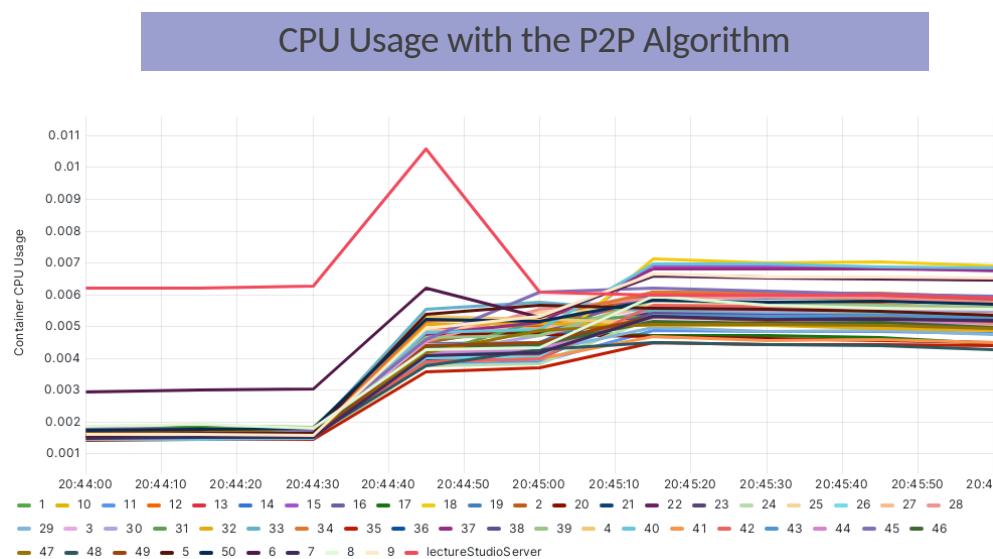
- Scaling Peers with Containerlab for Simplified Topology Deployment
- Deployment Time Increase of 5.37s per Additional Node
- Destroying Process Time Low, Approximately 9 Seconds for 75 Nodes



- CPU and Memory Usage Rise with More Peers Indicating Increased Host Load
- CPU and Memory Utilization Increase Linearly Showing Scalable Performance
- Memory Usage Growth Suggests Potential Bottleneck with More Peers

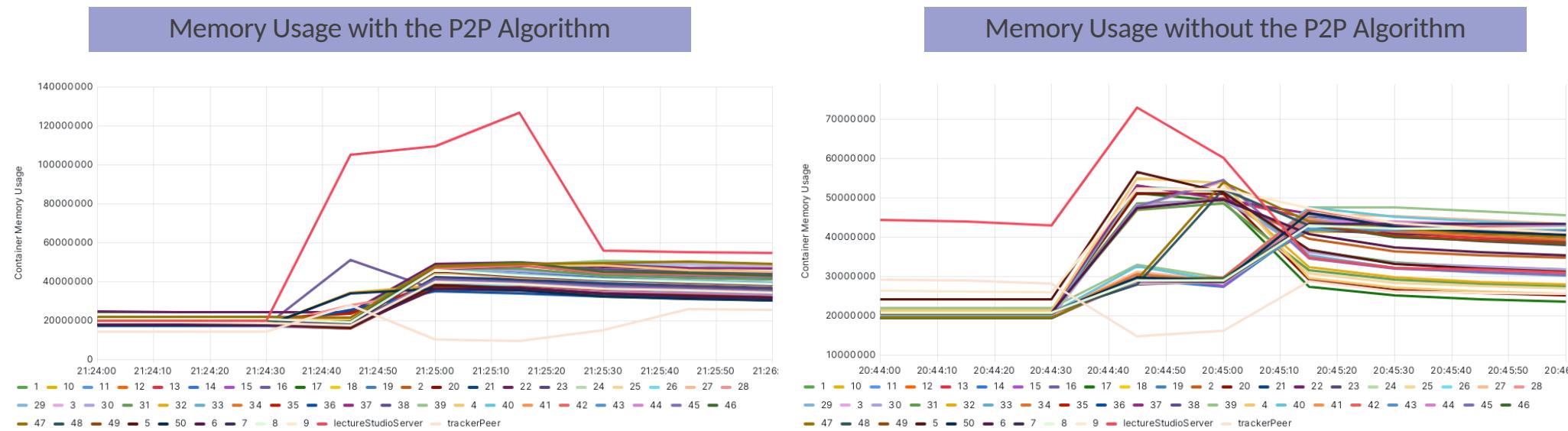
# CPU and Memory Usage Analysis of Nodes, RQ2.1 (1)

- Analysis of Resource Consumption (CPU and Memory Usage) of the P2P Algorithm Components (LectureStudio Server and Peers)
- High Initial CPU Usage on LectureStudio Server at Data Transfer Start
- Peak CPU Usage Reduction of 52% by the P2P Algorithm
- Load Distribution Leading to Decreased CPU Usage



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

- Memory Usage Reduction of 41% by the P2P Algorithm
- Resource Savings Due to the P2P Algorithm Utilization
- Task Distribution to Super Peers Lowers Server's Memory Usage Requirement



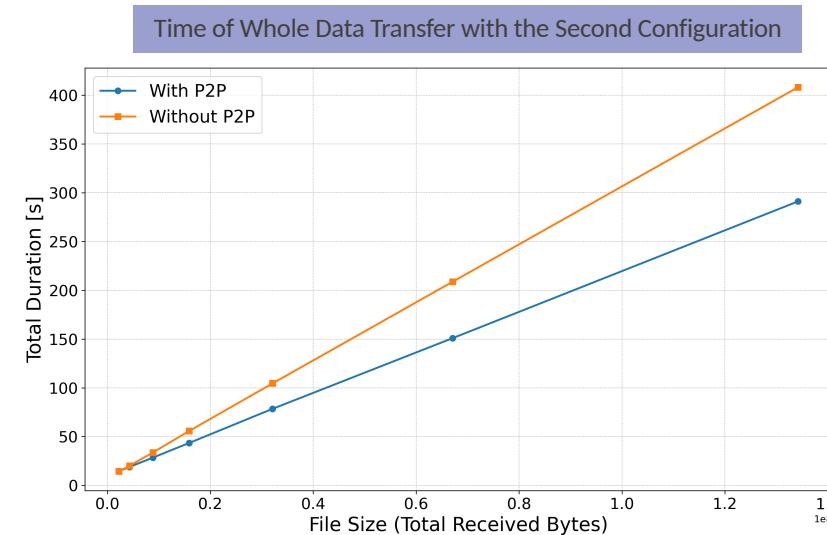
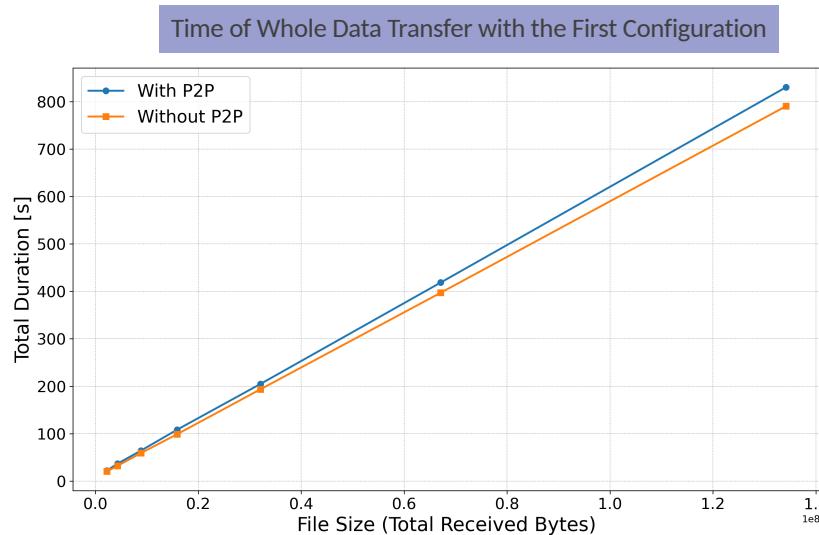
# Performance Evaluation of P2P Algorithm, RQ2.2 and RQ2.3 (1)

---

- Test Duration Measurement from First to Last Acknowledgment Message
  - Over 1000 Tests Conducted
  - Variation in Data Size or Number of Peers
- First Configuration for Real Network Data Simulation in Performance Analysis
  - Minimal Performance Difference between the P2P Algorithm and Server-Client Models
- Second Configuration with Varied Mean Values for Data Simulation
  - The P2P Algorithm Efficiency Increase with More Peers and Larger Data Size
- Small File Sizes Limited Benefit from P2P Optimization
- Larger File Transfers Indicative of the P2P Algorithm Efficiency
- Variation in Average Upload and Download Speeds Across Configurations Affects Performance
  - Higher Average Speeds in Second Configuration for Enhanced Network Efficiency, Bottleneck Reduction

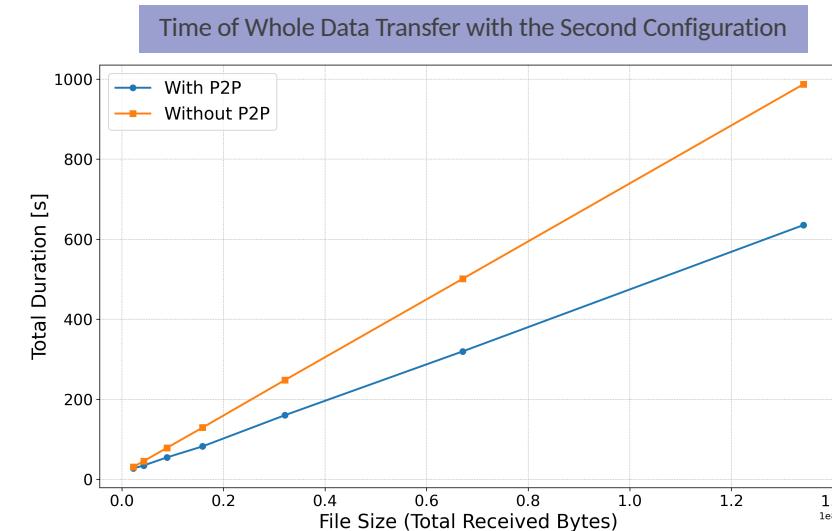
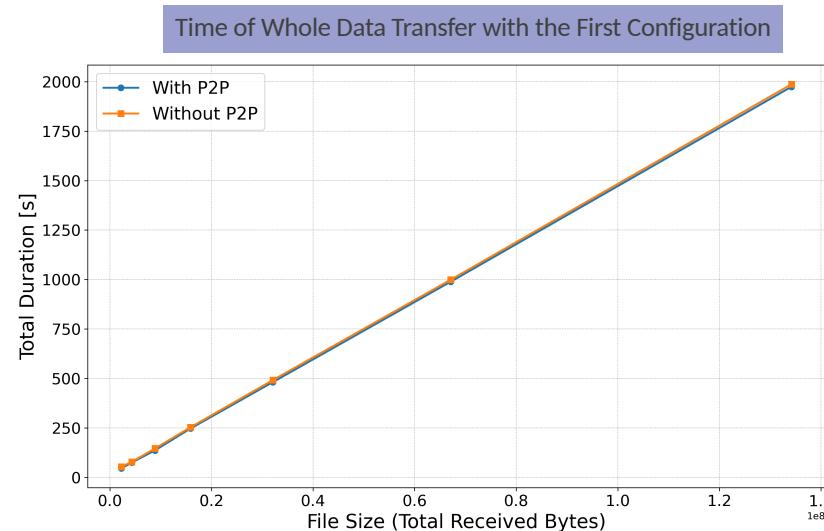
# Performance Evaluation of P2P Algorithm, RQ2.2 and RQ2.3 (2)

- First Configuration with the LectureStudio Server and 20 Peers
  - The P2P Algorithm Performance Little Worse to Traditional Approach
- Second Configuration with the LectureStudio Server and 20 Peers
  - Transferring 2MB Data Size, the P2P Algorithm Performance Nearly Identical to Traditional Approach
  - Transferring 128MB Data Size, the P2P Algorithm Performance 27% Faster than Traditional Approach



# Performance Evaluation of 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 Traditional Approach
  - Transferring 128MB Data Size, the P2P Algorithm Performance 5% Faster than Traditional Approach
- Second Configuration with the LectureStudio Server and 50 Peers
  - Transferring 2MB Data Size, the P2P Algorithm Performance 12% Faster than Traditional Approach
  - Transferring 128MB Data Size, the P2P Algorithm Performance 35% Faster than Traditional Approach



# Challenges

---

- Finding real network dataset
- Configuration of the Network Characteristics for Connections
  - Bandwidth Limitation
  - Latency Addition
  - Packet Loss Simulation
- Synchronisation Problem of Total Time
- Allocation of Bandwidth for the Connections between the LectureStudio Server and Peers
- Measurement of the Network Characteristics for Connections
- Data Transmission between the LectureStudio Server and Peers
- Monitoring by Grafana, Prometheus and cAdvisor

# Conclusion and Future Work

---

- Goal: Develop a Testbed for the P2P Data Distribution Algorithm
  - Utilization of Docker and Containerlab for An Efficient, Isolated Testing Environment
- Simulation of Real Network Environments and Complex Network Topologies
  - High Replication Accuracy of Bandwidth and Latency
  - Effective Scalability with Increasing Nodes
- High Resource Demand without the P2P Algorithm; Significant Reduction with the P2P Algorithm
  - Effective Data Distribution and Reduced Server Load through the P2P Algorithm
- Limited P2P Benefits for Small Files and Efficiency Improvements in Different Configurations
- Robustness of the P2P Algorithm with Increased Peers and Data Size
- Enhancement of Packet Loss Simulation for Accurate Network Behavior
- Automation between the Testbed and the P2P Algorithm 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/>