

# Deterministic Multipath Backhaul

Nicolas Zunker

n.zunker@campus.tu-berlin.de

May 30, 2023

MASTER'S THESIS EXPOSEE

Telecommunication Systems Institute

Technische Universität Berlin

Examiner 1: Prof. Dr. Thomas Magedanz

Advisor: Hauke Buhr

Examiner 2: Prof. Dr. Marius Corici



# Chapter 1

## Deterministic Multipath Backhaul Exposee - Masters Thesis

### 1.1 Motivation and Problem Description

One of the aims for the fifth generation of mobile networks (5G) and its successors will be a greater diversification of the classes of service. As the use cases for these networks evolve, there is a greater need for quality of service (QoS) tailored to each use case. For example, in Industrial Internet of Things (IIoT) applications the bandwidth requirements are often quite low, however the requirements on latency, jitter, and reliability may be extremely stringent. Supporting these kinds of classes of service can be a challenge for mobile network operators (MNOs) and will require novel approaches to familiar problems, such as backhaul.

With more backhaul solutions becoming available (i.e. LEO satellite links, and mmWave backhaul) and adding to the existing set of options (DSL, optical fibre, DOCSIS, dedicated lines, etc.) network operators may choose to utilize more than one backhaul connection at once in order to increase the available bandwidth or to utilize the different qualities of the backhaul links. Such a situation, could then be used to achieve higher quality of service

by intelligently selecting on which link to forward packets. This approach bears similarity to multihoming as well as to multi-path routing, and can take inspiration from the existing body of research in these fields which has demonstrated that QoS can be improved by using multiple paths simultaneously [1, 2, 8, 10, 11, 16].

## 1.2 Goal of the Thesis

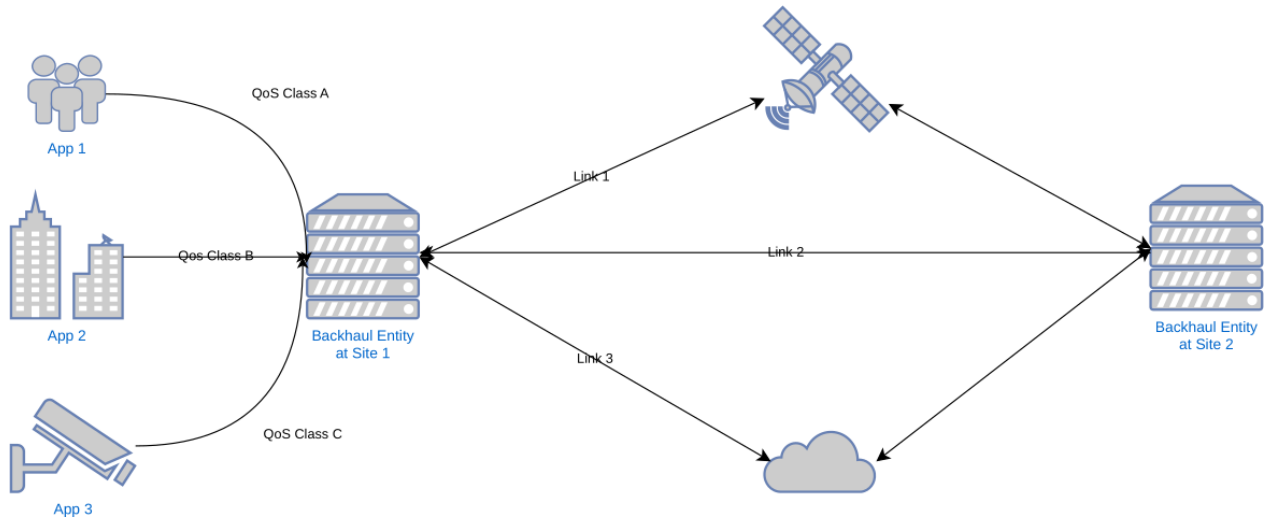


Figure 1.1: Use Case for the Backhaul Entity

The goal of this thesis is to design a backhaul entity, that can be placed at the ingress and egress point of connected networks, and which can intelligently forward packets on one or more links in order to meet specific QoS requirements. The performance of this approach will then be quantitatively analysed in experiments. Looking at figure 1.1 we can see how this is envisioned to work: A backhaul entity is deployed in 2 or more sites which have more than one egress link. Then, using the multiple links, the traffic is backhauled to the second site, while respecting it's QoS requirements. An example of how this entity could be deployed in multiple sites, interconnected in a mesh network, can be seen in figure 1.2. This could allow network operators to provide more reliable quality of service for it's users, especially for important applications (e.g. between industrial sites).

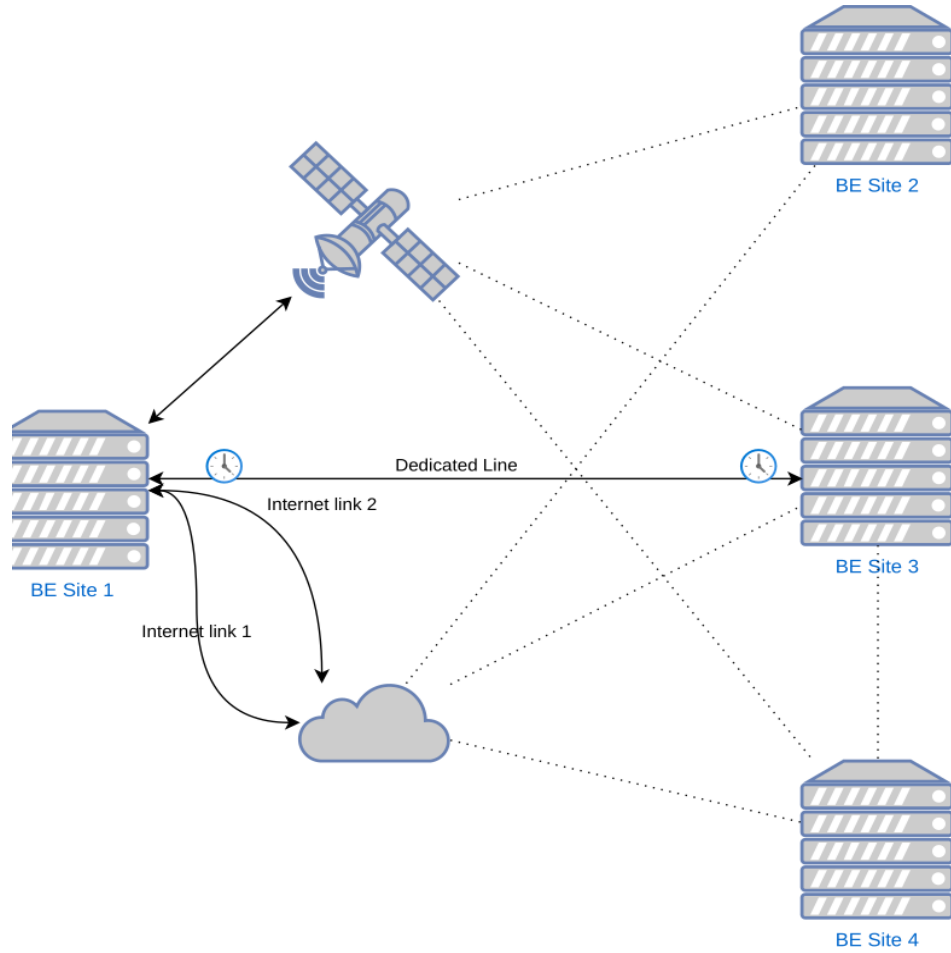


Figure 1.2: Network of Backhaul Entities

## 1.3 Structure of the Thesis

The planned work for the thesis will be structured as follows:

1. *Design*: Research existing approaches and solutions, and design an approach for reliable backhaul over multiple paths/links
2. *Implementation*: Implement said approach in a basic testbed consisting of a traffic generator, various link emulators, measurement devices, and the backhaul entities

3. *Evaluation*: Analyse the performance of the backhaul entity according to its ability to reduce latency, improve jitter, and provide any other QoS requirements. And compare it's performance with the performance of each individual link, as well as the performance of a round robin packet forwarding approach which utilises each link equally.

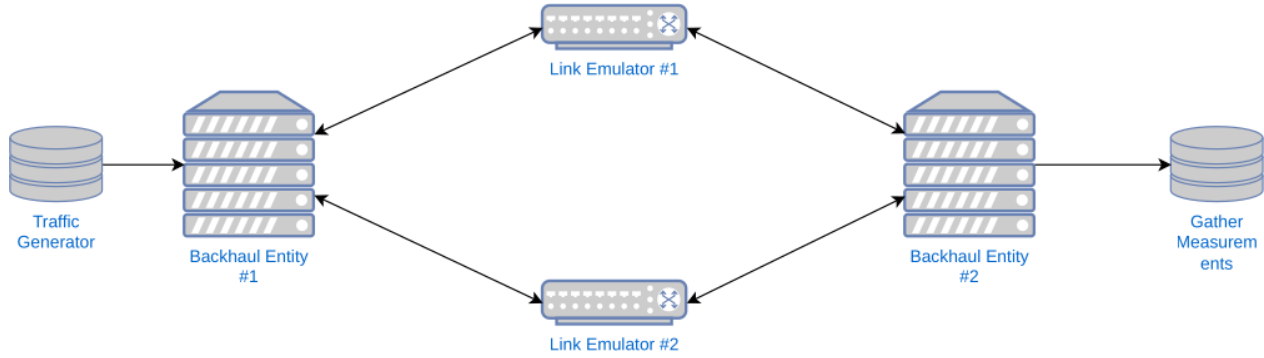


Figure 1.3: Testbed Setup

The structure of the report will be a simple 5 chapter format: 1) Introduction, 2) Background and Related Work, 3) Approach, 4) Evaluation, 5) Conclusion. The approach chapter will discuss the proposed solution, and the evaluation chapter will encompass the design of the testbed and an analysis of the results.

### 1.3.1 Timescale

A thesis should take 6 months, or roughly 26 weeks. At present it is proposed to divide those 26 weeks into 4 equal sized chunks. The first 3 chunks will correspond to the 3 items mentioned previously in this section, the final "chunk" will be to finish writing up the report, however it is planned to do some writing during the other 3 chunks as well. In the first period - the *Design* period, chapters 1 and 2 will also be worked on. After the *Implementation* period is finished, chapter 3 can be written. Chapters 4 and 5 can then be written after the *Evaluation* period finishes. That leaves the final chunk of time, which can be used to revise the document, as well as a buffer in case other parts of the thesis take longer to do.

The first phase of the experimental work will consist of setting up the testbed. This means installing the measurement modules as well as the link emulators. The next phase will require the implementation of the backhaul entity, as designed. The backhaul entity will require a module for collecting per-metric links, as well as the implementation of an algorithm to select the best link or links for a given flow. In the course of designing and preparing the backhaul entity there may be room for adjusting the initial plan based on what we experience during development. Next, the entity must be tested. During this phase traffic replays will be played over the testbed and evaluated. Thereafter the round robin approach and the "god routing" approach will also be evaluated. These two approaches will serve for comparison with the performance of the backhaul entity. Finally, this data must be analyzed, and the thesis can reach its conclusions, with 1 or 2 weeks buffer room for last minute revisions.

## **1.4 Envisioned Approach**

### **1.4.1 Collecting Per-Link Metrics**

In order to realize such a backhaul entity, the entity needs to be able to maintain a set of metrics about the available links to aid it in choosing which links to utilize. This may not always be straightforward, and could require prediction of link quality based on previous measurements.

#### **Measurement-based Metrics**

In [2] the authors collected both passive metrics (looking at response times for outgoing packets), and active measurements (sending ICMP ping, or TCP SYN messages and measuring the response time). It is important to note that using the passive measurements enabled their multihomed approach to perform well, but when using the active measurements the

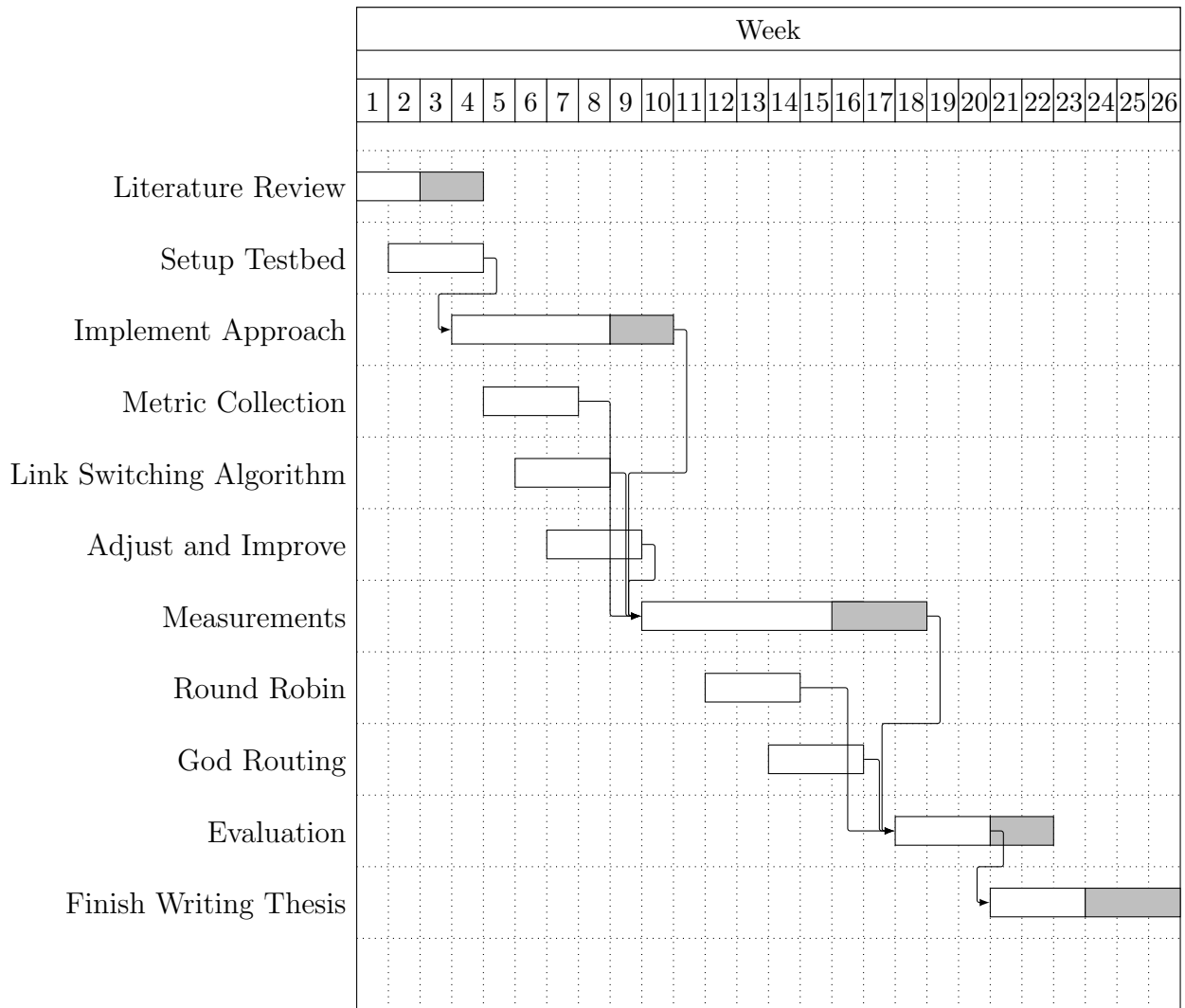


Figure 1.4: Gantt Chart: 01.06.23 - 01.12.23



performance was better. Crucially, the passive measurements worked better over larger sampling periods because it took longer to get a full overview of all the possible routes. Whereas the active sampling approach acquired its measurements faster and was thus more effective over smaller sampling intervals.

Considering these results, it is proposed to utilize both active and passive measurements. All three metrics- packet loss, latency, and jitter- will be periodically measured in an active manner, on a site to site basis. The period over which to perform these measurements is an important design decision.

Beyond this, these metrics will also be monitored on a passive basis wherever possible. For all three metrics a response is required for each sent message, in order to measure either the time needed (for latency and jitter), or to ascertain a packet has been lost. This may only be possible for TCP SYN and SYN ACK messages, and other protocols which are guaranteed to contain certain request-response handshakes.

### **1.4.2 Pre-configured Metrics**

Since certain links may be come with hard guarantees on packet loss, latency and jitter, this would drastically simplify the process, and thus allow us to save ourselves the measurement process for said links. Therefore the backhaul entity which will be designed, shall allow for a link to have the jitter, reliability and latency metrics pre-configured, as well as allowing them to be updated while the entity is running.

### **1.4.3 How to Guarantee QoS**

The design of this approach is also an interesting challenge. One idea to improve jitter when backhauling across multiple links is to duplicate packets and forward them on multiple links, and have the backhaul entity on the other end buffer incoming packets and release them at

a constant rate. This way in the event of a packet being lost on one link, the other link is still able to receive it and the delay caused by retransmission is avoided. The downside with this approach is that it guarantees the latency will always be that of the slowest link.

For reducing latency it would appear likely that the simplest approach may be a greedy method (as in [8] in the online case) which always selects the lowest latency connection. However there is room for nuance here since the connection must not be overloaded and also because certain traffic may have very relaxed latency requirements but use up more bandwidth. This means monitoring the load on any one link will be important. Finally there are also more intelligent approaches, i.e. integer linear programming (used in [11] and for the offline case in [8]) which could be used to optimally satisfy certain requirements.

The timescale over which to use a chosen link is also of interest. In [10] the time for which a link should be used is varied based on the predicted qualities of the link. These predictions are made based on past performance.

Reliability presents yet another challenge, however in a multihomed scenario it becomes easier to guarantee this via duplication, and/or forward error correction (FEC). For example if a packet flow requires 99% reliability this can be guaranteed by duplicating packets across two links which are both only 90% reliable. Alternatively, in such a situation, FEC could be used to pre-code the packets in order to provide the additional guarantees against packet loss and thus increase the reliability to the required level.

## **Proposed Solution**

The solution proposed for solving this multi-constrained QoS problem is to use integer linear programming (ILP). Although ILP is NP-Complete, we can parameterize the problem by the number of outgoing links, which never needs to be more than 4, and thus brute force the solutions. The ILP constrained problem will be to select those links on which to forward packets while minimizing the overall number of links used, and making sure to satisfy the

latency, jitter and reliability requirements of the given flow.

$$\text{Minimize } \sum_{i=1}^P x_i \quad (1.1)$$

$$\text{Where, } d(i) * x_i \leq D \quad (1.2)$$

$$j(i) * x_i \leq J \quad (1.3)$$

$$\text{and } 1 - \prod_{i=1}^P (1 - r(i) * x_i) \geq R \quad (1.4)$$

$$\text{for } x_i \in \{0, 1\} \quad (1.5)$$

Here the variables  $D$ ,  $J$ , and  $R$  are the flow's delay, jitter, and reliability requirements, while the functions  $d(i)$ ,  $j(i)$ ,  $r(i)$  are the predicted delay, jitter, and reliability of the link  $i$  out of the  $P$  total links. The predicted values will usually just be the latest measurement, as recommended in [2], however there is room here to use more advanced metrics to predict the future link quality and thus perform preemptive path switching. The  $x_i$  variable indicates whether or not link  $i$  shall be used. If a solution is found, then the flow's packets will be forwarded on each link  $i$  where  $x_i = 1$ , and if no solution can be found which satisfies these conditions then the flow is rejected because its QoS cannot be guaranteed.

#### 1.4.4 Potential Issues and Design Questions which are Still Open

At present the idea would be to use the GTP protocol to tunnel data between any two backhaul entities, and then use the tunnel endpoint IDs (TEIDs) to differentiate between different traffic flows. In the control plane of the backhaul entity, one link would be chosen to be used for control and co-ordination messages between the two backhaul entities.

This approach could run into trouble if there is fragmentation. Some applications attempt

to base their packet sizes on the most common MTU values for ethernet links (1500 bytes minus 20 bytes for the IPv4 header and 2 bytes for UDP), and this causes a problem when the packet is tunneled because the overhead of another IP header on top of the tunnel header pushes the packet beyond the MTU and thus it has to be fragmented, which can degrade performance. In the proposed design of the testbed, there is a traffic generator at use, so this generator can be configured not to generate packets exceeding 1450 bytes, but this issue is of practical concern for any realistic deployments.

### 1.4.5 Evaluating Performance

In order to evaluate the success of the proposed approach, three scenarios will be set up and investigated. Each setup will consist of two backhaul entities, and some number of emulated links going between them. The first scenario will feature 2 WAN links, a dedicated line, and a satellite connection. The second scenario will not feature the dedicated line, since it is expected that a leased line may be too obvious a choice for any of the traffic with strict QoS requirements. Finally, in order to further investigate these situations with an obviously superior candidate, the third scenario will just feature two links, a dedicated line and a WAN link.

In all of these scenarios the same traffic flows will be replayed. This traffic will contain various types of flows, with various QoS requirements. Before a new flow is started, the flow's requirements are sent to the backhaul entity and it is either accepted or rejected. During the traffic replay, the delay, jitter, and reliability will be measured.

These measurements will be performed once with the ILP approach proposed earlier and once with a simple round-robin approach for selecting links, and finally compared against an offline approach, where the optimal decision is computed with complete knowledge of the future. This offline approach will serve as the baseline for optimal performance, against which the other two approaches can be compared.

# Bibliography

- [1] Aditya Akella et al. “A measurement-based analysis of multihoming”. In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. 2003, pp. 353–364.
- [2] Aditya Akella et al. “On the performance benefits of multihoming route control”. In: *IEEE/ACM Transactions on Networking* 16.1 (2008), pp. 91–104.
- [3] Hind Alwan and Anjali Agarwal. “Multi-objective reliable multipath routing for wireless sensor networks”. In: *2010 IEEE Globecom Workshops*. IEEE. 2010, pp. 1227–1231.
- [4] Jennifer Andreoli-Fang and John T Chapman. “Mobile-aware scheduling for low latency backhaul over DOCSIS”. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE. 2017, pp. 1–6.
- [5] Fuqiao Chen and Chenyang Yin. “A Collaborative Traffic Scheduling Mechanism for Multi-homed Networks”. In: *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE. 2020, pp. 177–181.
- [6] Doğanalp Ergenç and Mathias Fischer. “On the reliability of ieee 802.1 cb frer”. In: *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE. 2021, pp. 1–10.
- [7] Igor Ganichev et al. “YAMR: Yet another multipath routing protocol”. In: *ACM SIGCOMM Computer Communication Review* 40.5 (2010), pp. 13–19.

- [8] David K Goldenberg et al. “Optimizing cost and performance for multihoming”. In: *ACM SIGCOMM Computer Communication Review* 34.4 (2004), pp. 79–92.
- [9] Fanglu Guo et al. “Experiences in building a multihoming load balancing system”. In: *IEEE INFOCOM 2004*. Vol. 2. IEEE. 2004, pp. 1241–1251.
- [10] Ahsan Habib and John Chuang. “Improving application QoS with residential multihoming”. In: *Computer Networks* 51.12 (2007), pp. 3323–3337.
- [11] Xiaoxia Huang and Yuguang Fang. “Multiconstrained QoS multipath routing in wireless sensor networks”. In: *Wireless Networks* 14.4 (2008), pp. 465–478.
- [12] Ralf Kundel et al. “User Plane Hardware Acceleration in Access Networks: Experiences in Offloading Network Functions in Real 5G Deployments.” In: *HICSS*. 2022, pp. 1–10.
- [13] Stanislav Lange et al. “Performance benchmarking of a software-based LTE SGW”. In: *2015 11th International Conference on Network and Service Management (CNSM)*. IEEE. 2015, pp. 378–383.
- [14] Jonathan Prados-Garzon and Tarik Taleb. “Asynchronous time-sensitive networking for 5G backhauling”. In: *IEEE Network* 35.2 (2021), pp. 144–151.
- [15] Shu Tao and Roch Guérin. “Application-specific path switching: A case study for streaming video”. In: *Proceedings of the 12th annual ACM international conference on Multimedia*. 2004, pp. 136–143.
- [16] Shu Tao et al. “Improving VoIP quality through path switching”. In: *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 4. IEEE. 2005, pp. 2268–2278.
- [17] Mohammed Tarique et al. “Survey of multipath routing protocols for mobile ad hoc networks”. In: *Journal of network and computer applications* 32.6 (2009), pp. 1125–1143.

- [18] Jack Tsai and Tim Moors. “A review of multipath routing protocols: From wireless ad hoc to mesh networks”. In: *ACoRN early career researcher workshop on wireless multihop networking*. Vol. 30. Citeseer. 2006.
- [19] Simon Tschöke et al. “Time-sensitive networking over metropolitan area networks for remote industrial control”. In: *2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE. 2021, pp. 1–4.
- [20] Gongzheng Zhang et al. “Fundamentals of heterogeneous backhaul design—Analysis and optimization”. In: *IEEE Transactions on Communications* 64.2 (2016), pp. 876–889.