

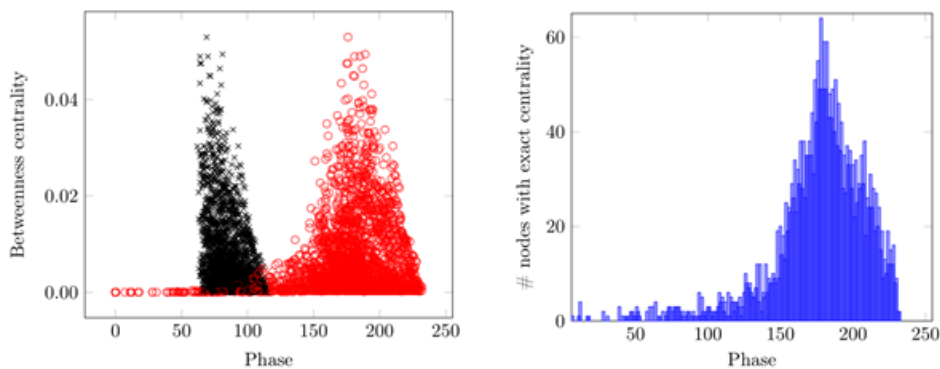
# Communication Technologies: Current Thesis/Praktikum-Topics

- The following list contains currently open thesis topics in the area of Communication Technologies.
- If you are interested in one of the topics below, **please contact**
  - **Prof Stefan Schmid** at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or the advisors mentioned directly at each topic
- Similarly, should you have an own idea for a potential thesis which you think might fit the research interests of our group, please contact us directly as well.

**Note for Bachelor students:** Even though the course Cooperative Systems (CS) is the classical "Gatekeeper" for the topics listed here, many topics also have a strong connection to other fields, e.g., Algorithms. If you are interested in a topic but have not taken CS, please let us know which Gatekeepers you have completed, there is a good chance you can still be admitted.

## Current CT-Thesis/Praktikum-Topics

- Centrality measures
- Spanning trees on real networks
- Coordination in changing networks
- What to do when your network fails? Fast local failover mechanisms
- Networks are in a constant state of change. How to update them consistently?
- Change the network topology on the fly: lasers, mirrors, antennas
- Change the network topology on the fly: algorithms, data structures, and self-adjustment
- Programmable matter
- Bitcoin from a networking perspective
- Distributed systems with cheap machines
- Distributed proofs
- Online exploration and evacuation
- Predictability of Virtual Network Control for Software-Defined Networks
- Update your Network: Loop Free!
- Your own project idea related to networks and communication technologies



### Centrality measures

In networks of various types, we are interested in understanding the relative importance of the nodes. Here, an “important” unit can be a router in a computer network that have many shortest paths going through it, or an opinion leader in a social network. There are several measures for quantifying the importance of different units in the system, called centrality measures. One important measure is called betweenness centrality: here, we give each unit “points” for every shortest path between a pair of other units it participates in, where the number of point decreases if there are more shortest paths for this pair.

In recent years, there were several works on distributed algorithm that allow each unit in a network to compute its own betweenness centrality, or other centrality measures. Our goals in this project are

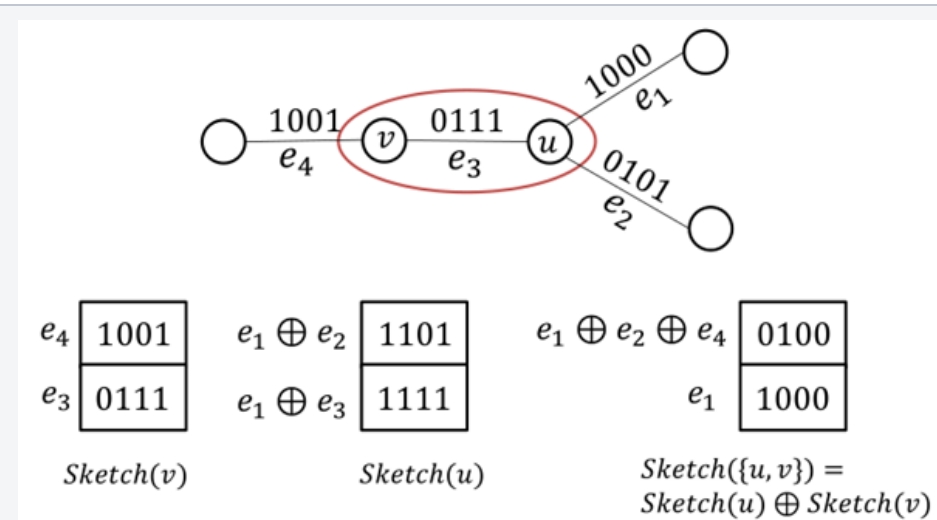
- Simulate a distributed betweenness centrality algorithm on a model of a real-world system;
- Compare the performance of different algorithms computing centrality measures, e.g., betweenness centrality vs. load centrality;
- Compare the outcomes of different algorithms computing centrality measures, i.e., the different centrality values under different measures.

For further reading, see

- A fast, centralized algorithm for computing betweenness centrality: <https://doi.org/10.1080/0022250X.2001.9990249>
- Computing betweenness centrality in real networks: <https://arxiv.org/pdf/2001.08108>
- Computing load centrality, a different centrality measure, in networks: <https://ieeexplore.ieee.org/document/8486345>

**This topic is available at all levels.**

For more information, please contact **Prof Stefan Schmid** at [Stefan\\_schmid@univie.ac.at](mailto:Stefan_schmid@univie.ac.at) or **Ami Paz** at [ami.paz@univie.ac.at](mailto:ami.paz@univie.ac.at)



## Spanning trees on real networks

Spanning trees are important mean of communication in real networks. A highly import type of spanning trees is a minimum spanning tree (MST), a tree that minimize the total cost, which could represent delays, infrastructure costs and more. In this project, we are interested in several theoretical algorithms for the problem of constructing an MST in a network that has a complete overly network. That is, in addition to the regular communication links, each node can also send small messages to all other nodes, in order to manage the network – in our case, in order to construct an MST.

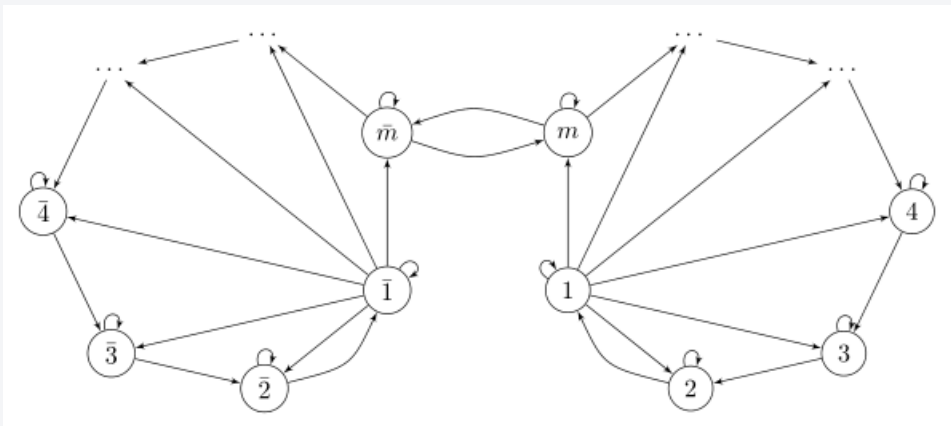
We will compare the real performance of several theoretical algorithms proposed in the last years:

- A constant-time, deterministic algorithm: <https://arxiv.org/abs/1912.04239>
- A constant-time, randomized algorithm: <https://arxiv.org/abs/1707.08484>
- A  $\log^*(n)$  time randomized algorithm: <https://dl.acm.org/doi/10.1145/2933057.2933103>

and maybe more...

**This topic is available at all levels.**

For more information, please contact **Prof Stefan Schmid** at [Stefan\\_schmid@univie.ac.at](mailto:Stefan_schmid@univie.ac.at) or **Ami Paz** at [ami.paz@univie.ac.at](mailto:ami.paz@univie.ac.at)



## Coordination in changing networks

We think of a highly dynamic network, say of sensors moving in space. Their goal is to reach agreement regarding some value, e.g., the average temperature measured, without having a global coordination, and while their interconnections change over time.

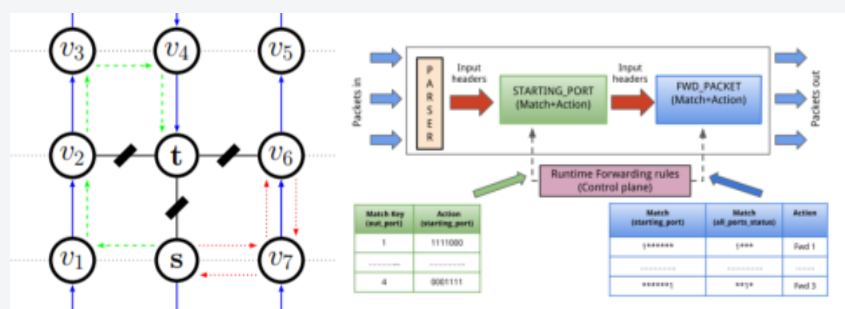
Many approaches and algorithms for this problem were proposed. We will focus on averaging algorithms, where the units use simple mechanisms, such as collecting values from their neighbors and adopting their average as the new value. The goal of this project is to give experimental comparison of several theoretical algorithms, on different networks.

For further reading, see:

- <https://dl.acm.org/doi/abs/10.1145/3212734.3212762>
- <https://drops.dagstuhl.de/opus/volltexte/2016/6281/>
- <https://arxiv.org/abs/1408.0620>

This topic is available at all levels.

For more information, please contact Prof Stefan Schmid at [Stefan\\_schmid@univie.ac.at](mailto:Stefan_schmid@univie.ac.at) or Ami Paz at [ami.paz@univie.ac.at](mailto:ami.paz@univie.ac.at)



## What to do when your network fails? Fast local failover mechanisms

Network failures happen all the time, especially the links that connect different parts of the networks can be faulty. Until these links are repaired, the network should still be usable. Ideally, the user should not even be aware that the links went down! As such, re-computation and re-convergence of routes is too slow for time-critical applications. Rather, operators are aiming for local fast failover methods, where in the best-case, no single packet is lost, but rather directly re-routed along different paths. To this end, alternate routing mechanisms are installed ahead of time, taking multiple failure scenarios into account. To give a simple example, when routing happens along a ring, standard routing could be clockwise, and if a link goes down, counter-clockwise.

We are interested in developing/improving mechanisms for such local fast failovers, but also real-world studies or simulation frameworks. Possible research directions include:

- theory/algorithms
- complexity and SAT-solver formulations
- verification and checking
- machine learning
- simulations and implementations

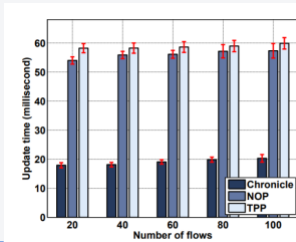
If you are interested, contact us and we can discuss possible directions for your thesis/project.

To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Paper: Bonsai: Efficient Fast Failover Routing Using Small Arborescences <https://www.univie.ac.at/ct/stefan/dsn19.pdf>
- Paper: Supporting Emerging Applications With Low-Latency Failover in P4 <https://www.net.t-labs.tu-berlin.de/~stefan/neat18.pdf>
- Paper: TI-MFA: Keep Calm and Reroute Segments Fast <https://www.net.t-labs.tu-berlin.de/~stefan/gi18.pdf>
- Paper: CASA: Congestion and Stretch Aware Static Fast Rerouting <https://www.univie.ac.at/ct/stefan/infocom2019e.pdf>
- Tutorial: Segment Routing: <https://www.segment-routing.net/tutorials/2016-09-27-topology-independent-lfa-ti-lfa/> and <https://segment-routing.org/index.php>

**This topic is available at all levels.**

For more information, please contact **Prof Stefan Schmid** at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or **Klaus-Tycho Förster** at [klaus-tycho.foerster@univie.ac.at](mailto:klaus-tycho.foerster@univie.ac.at) and/or **Mahmoud Parham** at [mahmoud.parham@univie.ac.at](mailto:mahmoud.parham@univie.ac.at)



blocked URL

## Networks are in a constant state of change. How to update them consistently?

Since the recent rise of Software-Defined Networking, computer networks have become *programmable*. Hence, network operators can change, e.g., routing behavior on the fly, all from the (comfortable) viewpoint of a central point of control. Such techniques allow to optimize the network's performance under different workloads and use-cases.

However, a computer network consists of many distributed components, sometimes even on the other side of the globe. Some parts of them can take longer to update their behavior, some might get conflicting commands – which can lead to inconsistent behavior, causing, e.g., congestion, packet drops, etc.

In this thesis, you will investigate causes and solutions to such inconsistencies. Possible research directions include:

- theory/algorithms
- machine learning
- simulations and implementations

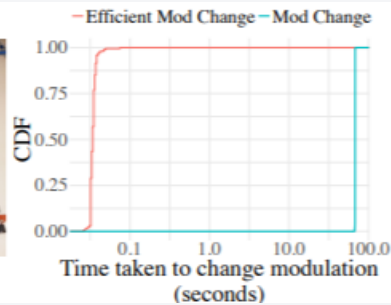
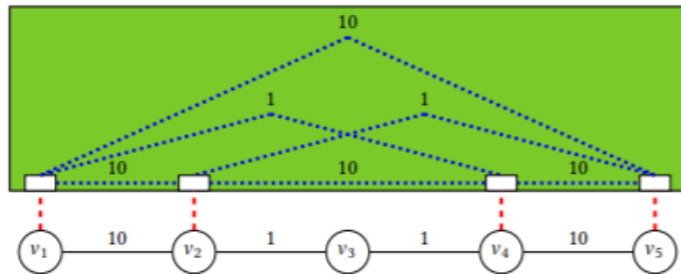
If you are interested, contact us and we can discuss possible directions for your thesis/project.

To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Poster on Algorithms for Consistent Flow Rerouting: <https://eprints.cs.univie.ac.at/5738/1/9-Algorithms.pdf>
- Video: The Art of Consistent SDN Updates: <https://www.youtube.com/watch?v=yjsfQyPLRV0>
- Video: Towards Lossless Data Center Reconfiguration <https://www.youtube.com/watch?v=PDQPkt5B9yw>
- Survey Paper: <https://eprints.cs.univie.ac.at/5495/1/2018-1609.02305v2.pdf>

**This topic is available at all levels. For a related topic, please also consider the topic: "Update your Network - Loop-Free!"**

For more information, please contact **Prof Stefan Schmid** at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or **Klaus-Tycho Förster** at [klaus-tycho.foerster@univie.ac.at](mailto:klaus-tycho.foerster@univie.ac.at)



## Change the network topology on the fly: lasers, mirrors, antennas

Usually, the cabling of networks is considered to be static. Unless new cables are laid out, routing algorithms have to work with what they have. Recent developments however allow the physical layer of networks to be *dynamic*! Data can be transferred over lasers, aimed at programmable mirrors, allowing near instantaneous establishment of connections. Similarly, wireless connections can employ beam-forming, such that WiFi is no longer a shared medium where multiple connections slow each other down. It is even possible to make the wide area network dynamic, by shifting wavelengths across fibers.

In your thesis, you will investigate and design routing mechanisms that take dynamic topologies into account. Possible research directions include:

- theory/algorithms
- machine learning
- simulations and implementations

If you are interested, contact us and we can discuss possible directions for your thesis/project.

To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Video: Agile Reconfigurable Data Center Interconnect <https://www.youtube.com/watch?v=PXIo3sltgLw>
- Paper: Automating Data Center Network Topologies Management with Machine Learning <http://cs.brown.edu/~tab/papers/NetAIM18.pdf>
- Paper: Characterizing the Algorithmic Complexity of Reconfigurable Data Center Architectures <https://eprints.cs.univie.ac.at/5658/11/2018-ancs.pdf>
- Paper: Demand-Aware Network Designs of Bounded Degree <https://www.net.t-labs.tu-berlin.de/~stefan/disc17.pdf>
- Paper: RADWAN: Rate Adaptive Wide Area Network <https://eprints.cs.univie.ac.at/5653/11/2018-sigcomm.pdf>

**This topic is available at all levels.**

For more information, please contact **Prof Stefan Schmid** at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or **Klaus-Tycho Förster** at [klaus-tycho.foerster@univie.ac.at](mailto:klaus-tycho.foerster@univie.ac.at)

blocked URLblocked URL

## Change the network topology on the fly: algorithms, data structures, and self-adjustment

Recent research in data center networks has made it possible to dynamically change parts of the physical network. That is, it is now possible to have a dynamic physical network topology for core parts of the data center network! Given that network traffic can be skewed, we can exploit this dynamicity to reduce the routing distance between frequently communicating nodes. There is a catch though: network adjustment comes at a cost! To that end, the research field of self-adjusting networks studies the trade-offs between adjustment and routing costs, combining knowledge from computer networks, online algorithms, data structures, machine learning, etc.

In your thesis, you will compare self-adjusting network algorithms in terms of performance. The exact topic in this area will be determined according to your background and interests. Our suggestion is that you will implement existing self-adjusting network algorithms, use real data center traffic data sets to test and demonstrate their performance, and propose improvements or (given the time) design novel algorithms. Thus, possible research directions include:

- theory/algorithms
- machine learning
- simulations and implementations

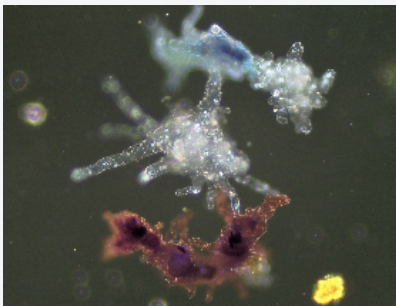
If you are interested, contact us and we can discuss possible directions for your thesis/project.

To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Video: Agile Reconfigurable Data Center Interconnect <https://www.youtube.com/watch?v=PXlo3sltgLw>
- Presentation: <https://www.univie.ac.at/ct/stefan/wroclaw18dan.pdf>
- Paper: Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks, by Chen Avin and Stefan Schmid: <https://net.t-labs.tu-berlin.de/~stefan/dan-san.pdf>
- Paper: Working Set Theorems for Routing in Self-Adjusting Skip List Networks, by Chen Avin, Iosif Salem, Stefan Schmid: [paper](#) / [slides](#)
- Paper: SplayNet: Towards Locally Self-Adjusting Networks, by Stefan Schmid, Chen Avin, Christian Scheideler, Michael Borokhovich, Bernhard Haeupler, Zvi Lotker: <https://www.univie.ac.at/ct/stefan/ton15splay.pdf>

**This topic is available at all levels.**

**For more information, please contact Prof Stefan Schmid at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or Iosif Salem at [iosif.salem@univie.ac.at](mailto:iosif.salem@univie.ac.at)**



## Programmable matter

Programmable matter refers to physical matter that can change, e.g., its shape, according to external stimuli respective sensing properties or input, and hence perform computation. To give an example, imagine a cluster of cells that is tasked to surround a specific object. While the underlying problems can be seen as distributed systems, unlike a normal computing system, these cells have very limited processing power and memory; nevertheless, they are still able to accomplish complicated tasks. The difficulty lies in creating simple algorithms, that can cope with the restricted resources, and especially take the distributed aspect into account. Besides biologically inspired models, recent work has also considered robots that use (un)folding of their geometric structure.

In this thesis, your focus will be on understanding the theory behind some chosen aspect of programmable matter, in turn developing new algorithms. There is also the option of simulating heuristics, to get a grasp of their performance.

If you are interested, contact us and we can discuss possible directions for your thesis/project.

To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Article: Printable robots <https://www.cbc.ca/news/canada/nova-scotia/origami-robots-halifax-child-prodigy-1.4569219>
- Paper: Amoebot: A New Model for Programmable Matter [https://cs.uni-paderborn.de/fileadmin/informatik/fg/ti/Publications/Publications2014/spaa0067-derakhshandeh\\_1\\_.pdf](https://cs.uni-paderborn.de/fileadmin/informatik/fg/ti/Publications/Publications2014/spaa0067-derakhshandeh_1_.pdf)
- Wikipedia: [https://en.wikipedia.org/wiki/Programmable\\_matter](https://en.wikipedia.org/wiki/Programmable_matter)

**This topic is available at all levels.**

**For more information, please contact Prof Stefan Schmid at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or Klaus-Tycho Förster at [klaus-tycho.foerster@univie.ac.at](mailto:klaus-tycho.foerster@univie.ac.at)**



## Bitcoin from a networking perspective

Chances are, you already heard a lot of things about Bitcoin and its variations: essentially, it is a decentralized currency, without any central bank /authority, allowing users to send payments (under pseudonymity) via a peer-to-peer (P2P) network. While the underlying security is well studied, privacy and performance aspects are not as well developed as they could be. For example, do you really want to wait 10 minutes until your payment is verified?

We are interested in studying the design of P2P networks for Bitcoin (and blockchains more generally) which support faster transactions and high liquidity, while providing a high degree of robustness and privacy. In particular, we aim to design a robust network topology which cannot easily be partitioned (risk of "double-spending"), together with a clever routing mechanism. Possible research directions include:

- theory/algorithms
- measurements
- simulations and implementations

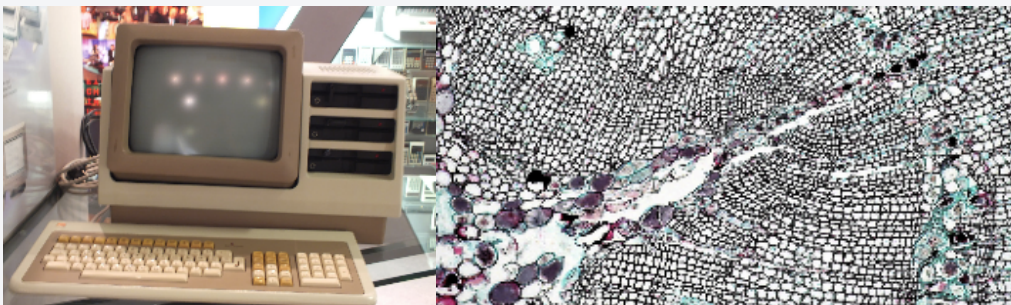
If you are interested, contact us and we can discuss possible directions for your thesis/project.

To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Wikipedia: [https://de.wikipedia.org/wiki/Lightning\\_Netzwerk](https://de.wikipedia.org/wiki/Lightning_Netzwerk) / [https://en.wikipedia.org/wiki/Lightning\\_Network](https://en.wikipedia.org/wiki/Lightning_Network)
- Website/Paper: Hijacking Bitcoin: Routing Attacks on Cryptocurrencies <https://btc-hijack.ethz.ch/>
- Paper: Scalable Funding of Bitcoin Micropayment Channel Networks [https://www.tik.ee.ethz.ch/file/a20a865ce40d40c8f942cf206a7cba96/Scalable\\_Funding\\_Of\\_Blockchain\\_Micropayment\\_Networks%20\(1\).pdf](https://www.tik.ee.ethz.ch/file/a20a865ce40d40c8f942cf206a7cba96/Scalable_Funding_Of_Blockchain_Micropayment_Networks%20(1).pdf)
- Website: Will Lightning Help or Hurt Bitcoin Privacy? <https://www.coindesk.com/will-lightning-help-hurt-bitcoin-privacy/>
- <https://www.whatbitcoindid.com/podcast/christian-decker-an-introduction-to-the-lightning-network>

**This topic is available at all levels.**

For more information, please contact **Prof Stefan Schmid** at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at)



## Distributed systems with cheap machines



When you think of distributed systems, you might think of powerful machines that even on their own are more advanced than your PC at home. However, in quite a few distributed systems (e.g., embedded systems for cars or the Internet-of-Things), the machines on their own are not so powerful: their computational power and memory might be limited, along with power constraints, only allowing for small size messages. Additionally, their knowledge of the global network topology could be outdated or restricted, with the further problem of network faults. Still, even with such cheap machines, many problems can still be solved well enough, although maybe different algorithms and paradigms are needed. Another important aspect in this area are biologically inspired systems: e.g., ants are rather primitive when compared to a modern computer, but still they can accomplish amazing tasks when working together.

In this thesis, you will dive into the fascinating world of distributed systems where each node only has very limited resources and abilities. Possible research directions include:

- theory/algorithms
- modelling/complexity
- simulations and implementations

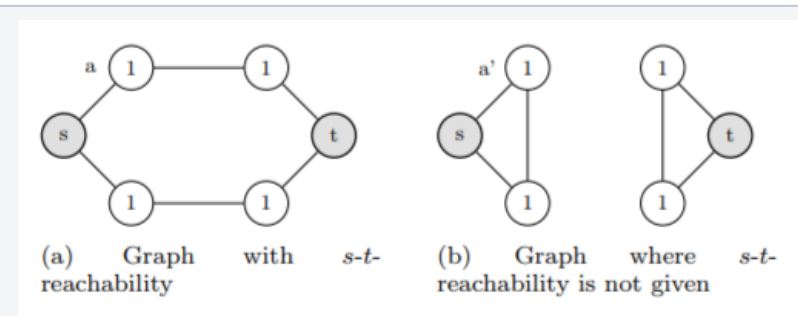
If you are interested, contact us and we can discuss possible directions for your thesis/project.

To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Paper: Stone Age Distributed Computing <https://www.tik.ee.ethz.ch/file/1239fec24129a5118fb41ef8a8338ba1/stone-age.pdf>
- Survey: Theoretical Distributed Computing meets Biology: A Review <https://www.irif.fr/~pandit/Review.pdf>
- Presentation: A biological solution to a fundamental distributed computing problem <http://www.dis.uniroma1.it/~becchett/Elective/Section1/MIS.pdf>

**This topic is available at all levels.**

**For more information, please contact Prof Stefan Schmid at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or Klaus-Tycho Förster at [klaus-tycho.foerster@univie.ac.at](mailto:klaus-tycho.foerster@univie.ac.at)**



## Distributed proofs

If you want to find out if there is a path between two nodes  $s$  and  $t$  in a network, then that is usually an easy problem: just do, e.g., a depth-first-search from  $s$  and check if you can reach  $t$ . Can we be faster if we want to verify such properties in a distributed fashion, with some pre-processing? Essentially, that is the idea of distributed proofs: one distributes some information ahead of time, and if a property is true, all nodes should respond yes, and if not, at least one node should raise an alarm. The speedup comes from the fact that the nodes only need to check with their neighbors to verify the global property.

To give an intuition for this example, consider the picture above: If you just label nodes along the paths from  $s$  to  $t$  with a "1", then locally, you cannot distinguish between an instance that is connected and one that is not: e.g., from the viewpoint of the nodes  $a, a'$ , it looks the same. In this case, one can resolve the issue by only labeling a single shortest path between  $s$  and  $t$  with "1"s, and label all other nodes with "0". There is a wide variety of topics where interesting research questions arise, e.g., for consistent network updates, for dynamic networks, or in the context of biology.

Possible research directions are mostly in the realm of algorithms and theory, but also some implementation could be possible.

If you are interested, contact us and we can discuss possible directions for your thesis/project.

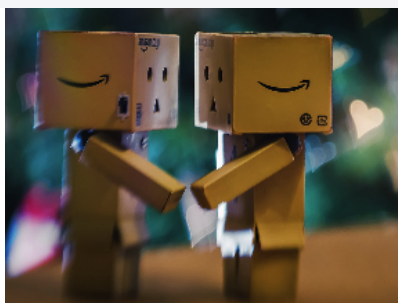
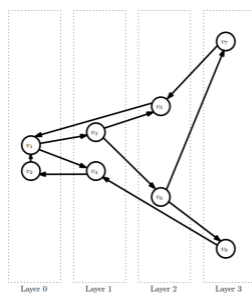
To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Paper: <https://ktfoerster.github.io/paper/2019-nca.pdf>
- Paper: <https://ktfoerster.github.io/paper/2017-dynamic-labelings.pdf>
- Paper: <https://ktfoerster.github.io/paper/2016-label-sdns.pdf>

**This topic is available at all levels.**

**For more information, please contact Prof Stefan Schmid at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or Klaus-Tycho Förster at [klaus-tycho.foerster@univie.ac.at](mailto:klaus-tycho.foerster@univie.ac.at)**





## Online exploration and evacuation

Imagine you are visiting Vienna for the first time. There are a lot of great places to see -- probably even places that you have not heard about before. Since you want to explore on your own, you do not buy a map and start walking. After some time, you notice that your tour is not quite optimal, you could have visited more places if you knew the layout of the town. Of course no minute exploring Vienna is wasted, but still, how much time would a map have saved you?

In more formal terms, this can be modeled as one of the standard problems in robotics. A robot starting at a base has to visit the whole graph and return to its home. If you know the whole graph, then this is much easier than if you start with zero information about the graph. A similar problem of interest is to evacuate, i.e., you have to find a specific spot where the exit is hidden. Collaboration can also be of interest, where multiple agents/robots explore the unknown environment.

Possible research directions are mostly in the realm of algorithms and theory, but also some implementation could be possible.

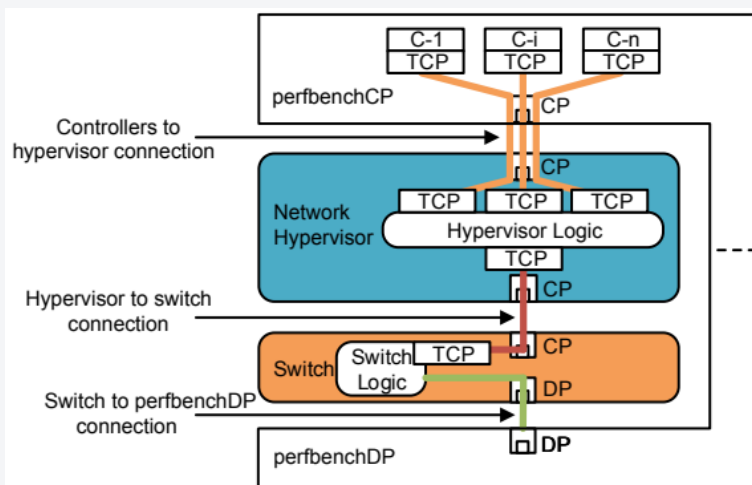
If you are interested, contact us and we can discuss possible directions for your thesis/project.

To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- Paper: <https://ktfoerster.github.io/paper/2017-evacuation.pdf>
- Paper: <https://ktfoerster.github.io/paper/2016-graph.pdf>
- Paper: <https://www.net.t-labs.tu-berlin.de/~stefan/icdcn18.pdf>
- Paper: <https://arxiv.org/pdf/1903.00581.pdf>
- Book: [Distributed Computing by Mobile Entities](#)
- Wikipedia: [https://en.wikipedia.org/wiki/Graph\\_traversal#Graph\\_exploration](https://en.wikipedia.org/wiki/Graph_traversal#Graph_exploration)

**This topic is available at all levels.**

For more information, please contact **Prof Stefan Schmid** at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at) and/or **Klaus-Tycho Förster** at [klaus-tycho.foerster@univie.ac.at](mailto:klaus-tycho.foerster@univie.ac.at)



## Predictability of Virtual Network Control for Software-Defined Networks

Network Virtualization (NV) and Software-Defined networking (SDN) are expected to play a major role in future communication networks. With network virtualization, services can run in an isolated manner within their own virtual networks. Software-defined networking is expected to introduce the programmability gain for each virtual network. In order to realize such setting, however, an additional entity is needed: the SDN network hypervisor realizing the functionalities of virtualization for software-defined networks. Interestingly, less is known about the potential sources of unpredictability of today's network hypervisor solutions. This stands in stark contrast to the overall goal to achieve programmable virtual networks with predictable control.

The aim of this thesis is to investigate the different mechanisms to provide virtual software defined networks and potential possibilities with particular focus on OnVisor. OnVisor has been recently published as a virtualization add on of the carrier grade software-defined networking controller ONOS. Furthermore, OnVisor performance should be compared with FlowVisor and OpenVirtex | hypervisors that are a little long in the tooth. This measurement study should identify scenarios in which interference among multi-tenant setups in virtualized SDN networks can happen. Different control plane messages and network topology settings should be investigated to reveal the cases when interference happens.

Moreover by looking at the measurement data, the power of Machine Learning should be used to answer the overall question: is the performance of multi-tenant SDN environments truly predictable?

### First steps

- Get into state of the art on Software-defined Networking with focus on SDN Network Hypervisors (OpenVirtex [1], HyperFlex [2], OnVisor [3], CoVisor [4], FlowVisor [5])
- Collect and study related work on predictability, e.g., for software switches.
- Setup a measurement testbed for studying the performance of OnVisor.
- Using data analysis technologies to study the predictability of OnVisor.

### What we offer

- The possibility to do cutting edge research
- Getting into research: how to write papers, how to publish
- The possibility to publish research results on workshops and conferences (We have already successfully published together with students)

### Expected knowledge

- Highly motivated students towards conducting research
- The will to spend extra time to learn new emerging concepts
- Programming skills needed: Python, Java, C++
- Machine learning and statistics knowledge

**Keywords:** Software Dened Networking, Network Virtualization, Machine Learning, OpenFlow, Predictability References

- [1] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, W. Snow, and G. Parulkar, OpenVirteX: a network hypervisor, in Proc. USENIX Open Netw. Summit (ONS), Santa Clara, CA, Mar. 2014.
- [2] A. Blenk, A. Basta, and W. Kellerer, HyperFlex: An SDN virtualization architecture with exible hypervisor function allocation, in Proc. IFIP/IEEE Int. Symp. IM Netw., IEEE, May 2015.
- [3] Y. Han, T. Vachuska, A. Al-Shabibi, J. Li, H. Huang, W. Snow, and J. W.-K. Hong, Onvisor: Towards a scalable and exible sdn-based network virtualization platform on onos, International Journal of Network Management, 28 (2018).
- [4] X. Jin, J. Gossels, J. Rexford, and D. Walker, CoVisor: A compositional hypervisor for software-dened networks, in Proc. USENIX Symp. NSDI, NSDI'15, USENIX Association, May 2015.
- [5] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. Mckeown, and G. Parulkar, FlowVisor: A network virtualization layer, tech. rep., OpenFlow Consortium, 2009.

**This topic is available at all levels.**

**For more information, please contact Prof Stefan Schmid at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at)**

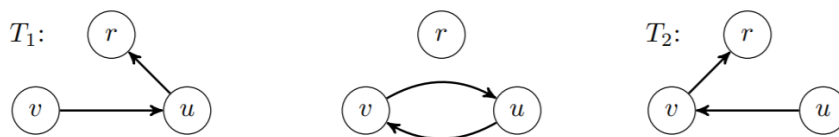


Figure 10: In the tree  $T_1$  to the left,  $v$  forwards all packets for  $r$  to  $u$ , which in turn forwards all packets to  $r$ . This situation is flipped in  $T_2$ :  $u$  forwards all packets for  $r$  to  $v$ , which in turn forwards all packets to  $r$ . Should  $u$  update before  $v$ , then the situation in the middle occurs: The graph is no longer a tree, and all packets loop between  $v$  and  $u$ .

**Update your Network: Loop Free!**

Correct routing is one of the fundamental properties a network should provide, after all, packets must be able to reach their destination. However, sometimes routes have to change due to maintenance, failures, congestion etc. Even though the old and the new route provide reachability, the inherent asynchrony in networks could mean that temporarily networks get lost in a loop. For an example, see the above figure, where the task is to update from routing Tree 1 to Tree 2. If node  $v$  updates too slowly, packets will enter a loop, and subsequently get dropped as the buffers overflow. Even though modern programmable networks offer great flexibility how to handle this problem, there are still many open questions. In this thesis, you will be able to investigate this problem from both a practical and a theoretical perspective, according to your background and strengths. For example, you could implement your algorithmic ideas and benchmark them - how and when can current techniques be improved?

#### First steps

- Understand the problem and its motivation
- Check related algorithms and their implementation

#### What we offer

- The possibility to do cutting edge research
- Getting into research: how to write papers, how to publish, how to provide implementations for the research community
- The possibility to publish research results on workshops and conferences (We have already successfully published together with students)

#### Expected knowledge

- Highly motivated students towards conducting research
- Good programming skills in C++
- Fundamental graph theory and related algorithms

**Keywords:** Algorithm engineering, graph algorithms, routing, consistency

If you are interested, contact us and we can discuss possible directions for your thesis/project. To get a feeling for some past research on these topics (non-exhaustive), please feel free to look at the following:

- How does Google handle such loops in their network (Section 4.4): <http://minlanyu.seas.harvard.edu/teach/ALL/b4andafter.pdf>
- How to handle the problem from a networking theory (e.g. Section 1,3,5) <https://ktfoerster.github.io/paper/2016-networking.pdf> respectively algorithm engineering perspective (e.g. Section 3) : <https://pdfs.semanticscholar.org/220c/e6b7b095b4608e671ca5ea9889ac0f5d09d8.pdf>

**This topic is available at all levels.**

For more information, please contact **Kathrin Hanauer** at [kathrin.hanauer@univie.ac.at](mailto:kathrin.hanauer@univie.ac.at) and/or **Klaus-Tycho Förster** at [klaus-tycho.foerster@univie.ac.at](mailto:klaus-tycho.foerster@univie.ac.at)



### Your own project idea related to networks and communication technologies

Should you have your own idea for a potential thesis which you think might fit the research interests of our group, do not hesitate to contact us directly.

It can be related to the topics above, but also something completely different, as long as it roughly fits into our area.

**This topic is available at all levels.**

For more information, please contact **Prof Stefan Schmid** at [stefan\\_schmid@univie.ac.at](mailto:stefan_schmid@univie.ac.at)