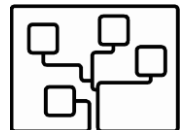


Path Determination for Network Slicing in Wireless Mesh Disaster Networks

Alexander Seng

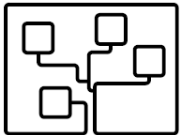
seng@e-technik.org

Frankfurt University of Applied Sciences, Germany
Research Group for Telecommunication Networks



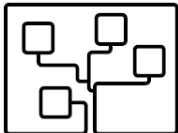
Inhalt

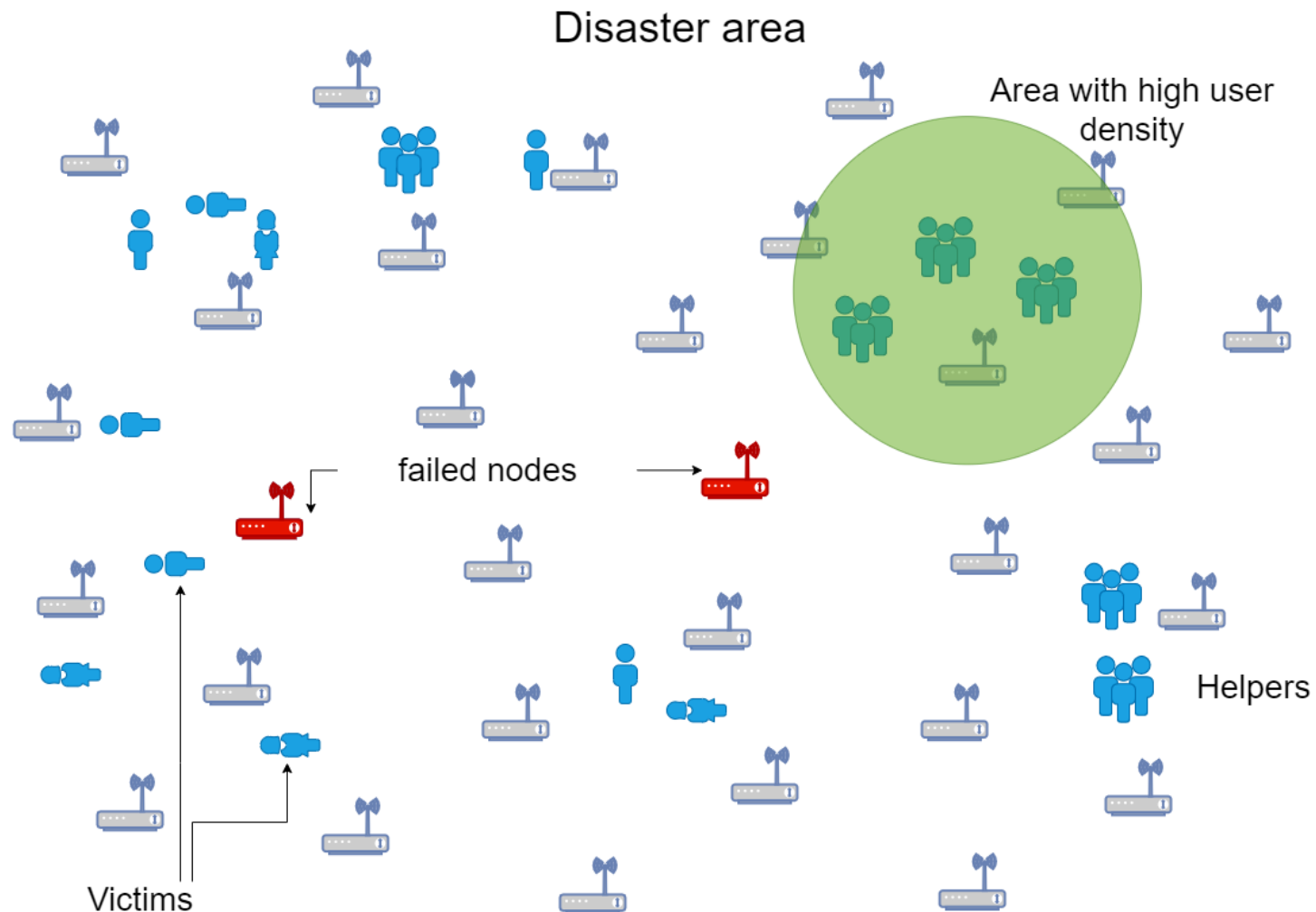
1. Einführung
2. Modellierung der Umgebung
3. Pfadermittlung und Testaufbau
4. Ergebnisse und zukünftige Arbeit



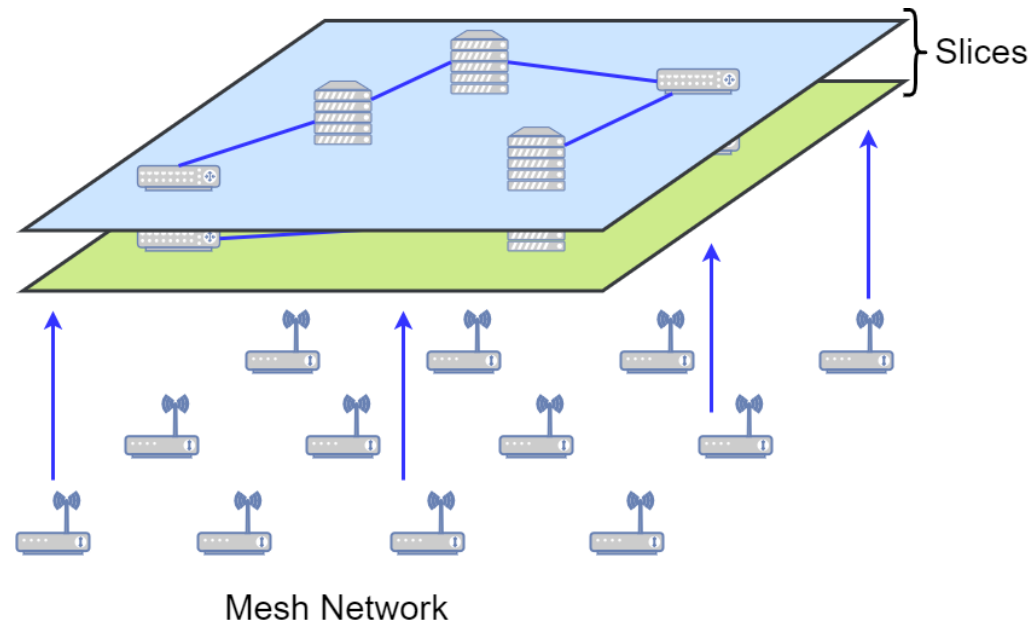
1 Einführung

- **Network Slicing: wichtige Neuerung in 5G**
- **Eigenständige virtuelle Netzwerke auf der gleichen Infrastruktur [1]**
- **Speziell für Services angepasst**
- **Network Functions Virtualisation (NFV), Software-Defined Networking (SDN) und Mobile Edge Computing (MEC)**
- **Wireless Mesh Networks (WMNs): selbst organisierende, selbst konfigurierende IEEE 802.11 Funknetzwerke [2]**
- **Schneller Aufbau einer Kommunikationsinfrastruktur**
- **Potenzielle Anwendung: in Katastrophengebieten [3]**

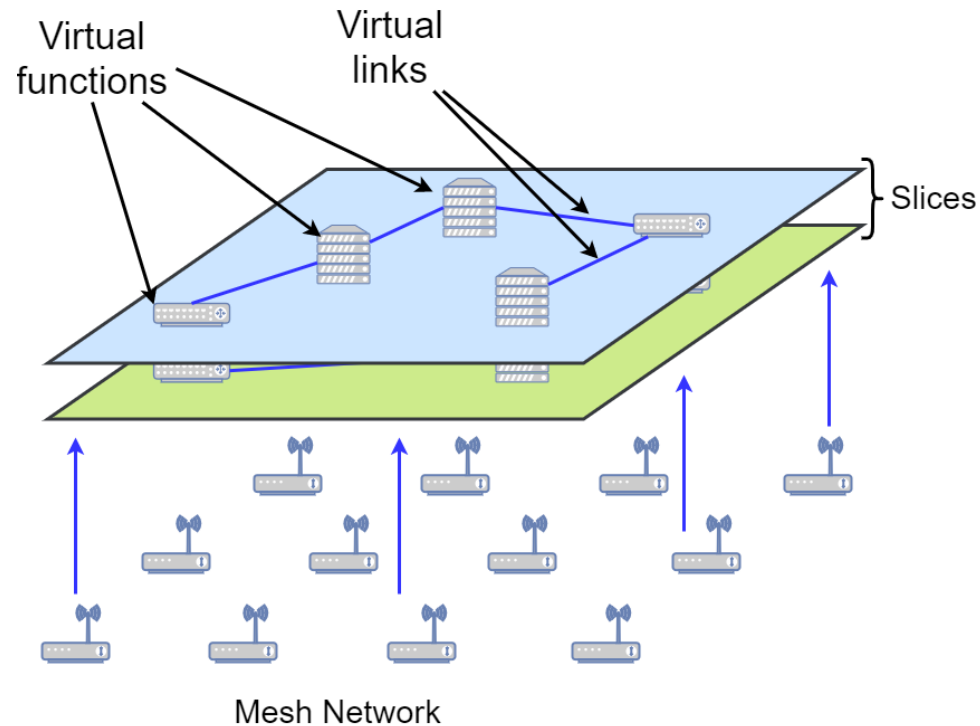




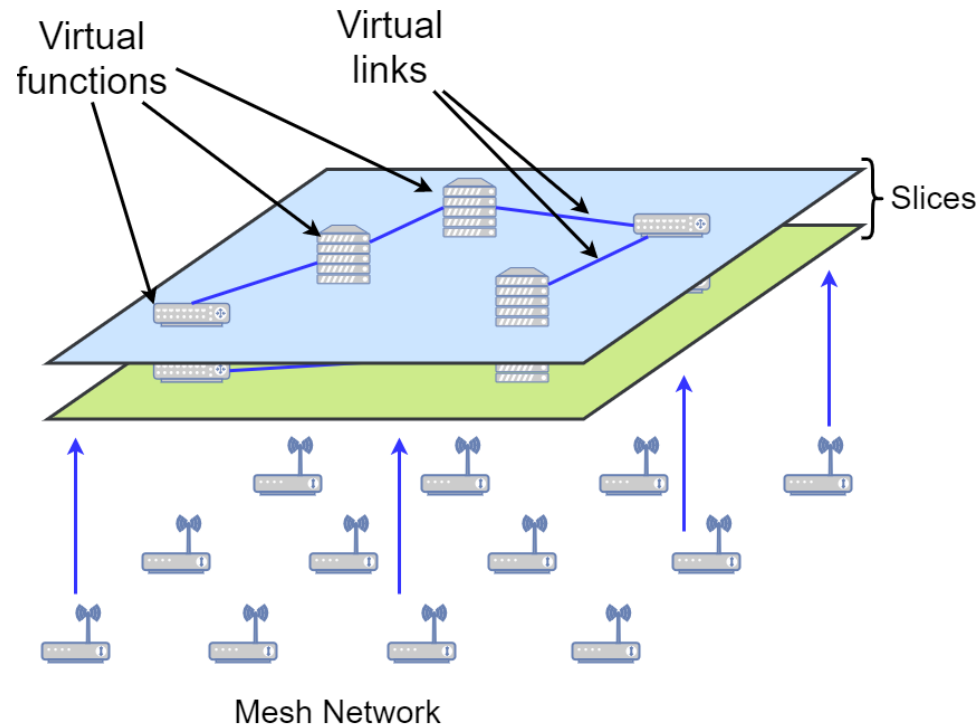
- Bereitstellung eines angepassten Netzes



- Bereitstellung eines angepassten Netzes
- **Slice = virtuelle Netzwerkfunktionen + virtuelle Links**

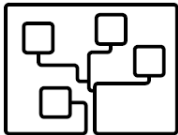


- Bereitstellung eines angepassten Netzes
- Slice = virtuelle Netzwerkfunktionen + virtuelle Links
- Übersetzung auf physische Infrastruktur

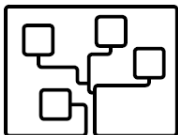
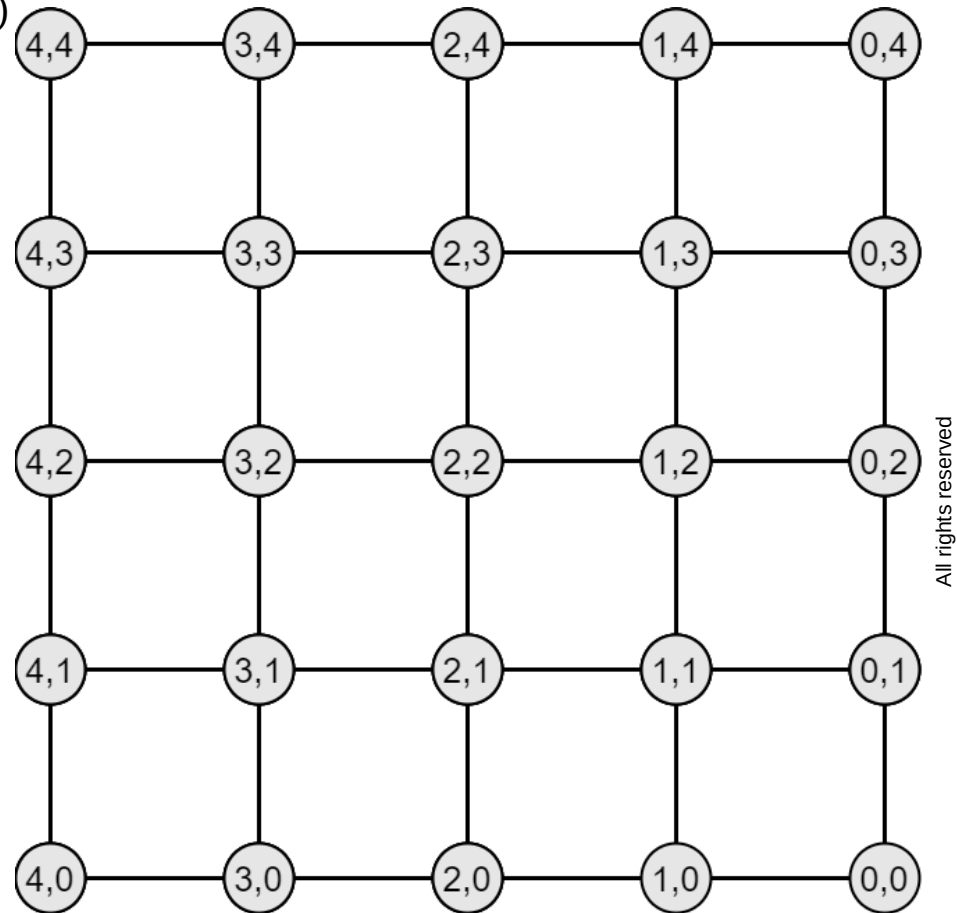


2 Modellierung der WMN Umgebung

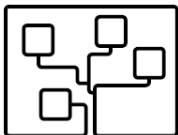
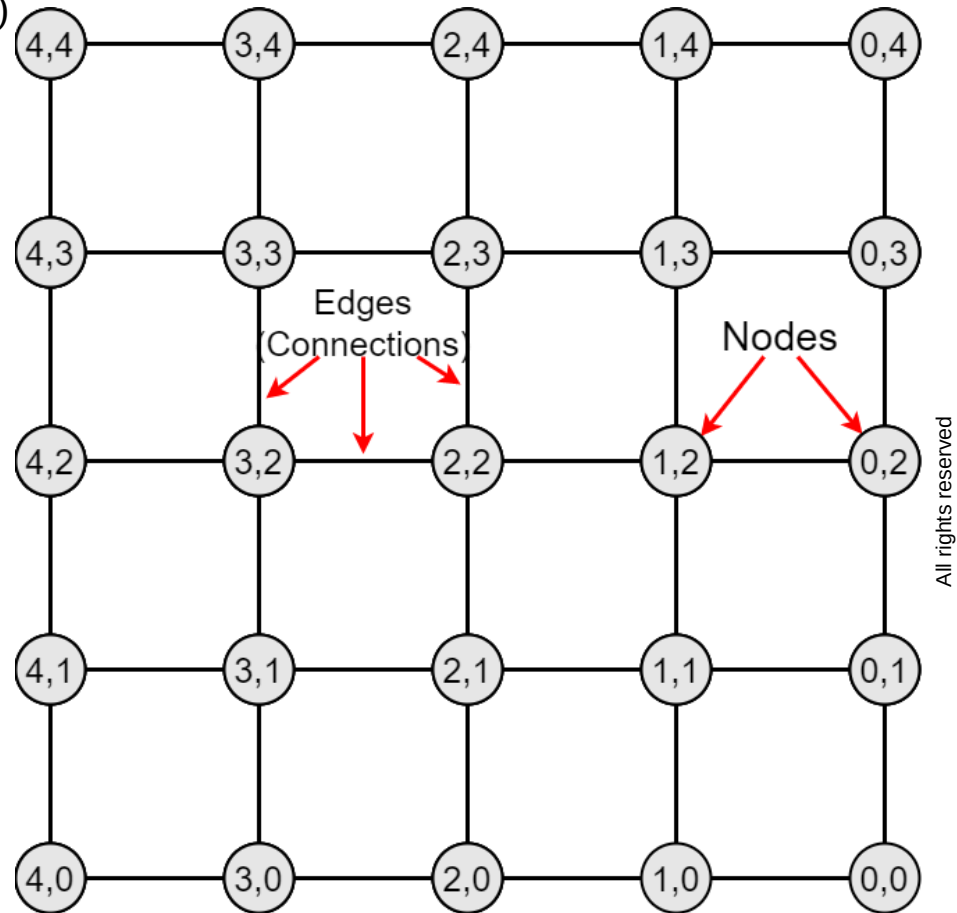
- **Abstrahierung der Netzwerkumgebung**
- **Netzwerke können als Graph betrachtet werden**
- **Zum Testen von Pfadalgorithmen wird das WMN als Graph betrachtet**
- **Laborbedingungen für Tests**



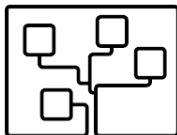
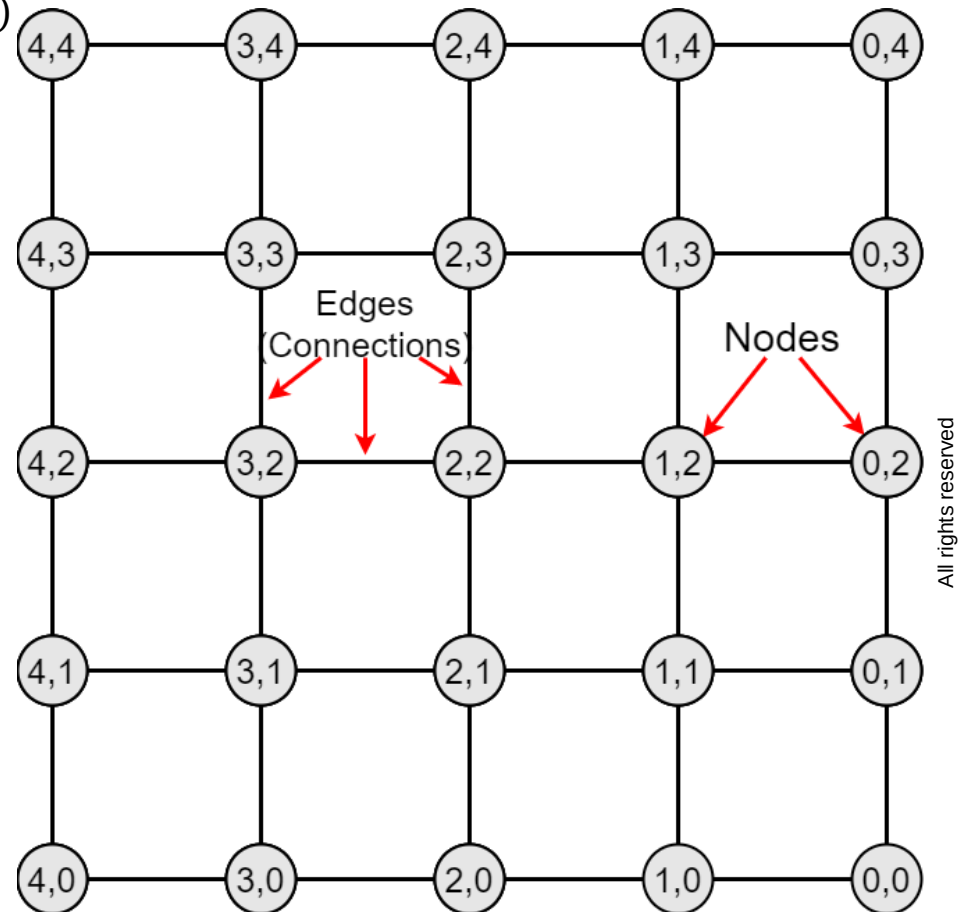
- WMN als ungerichteter Graph $G(N, E)$
[4]



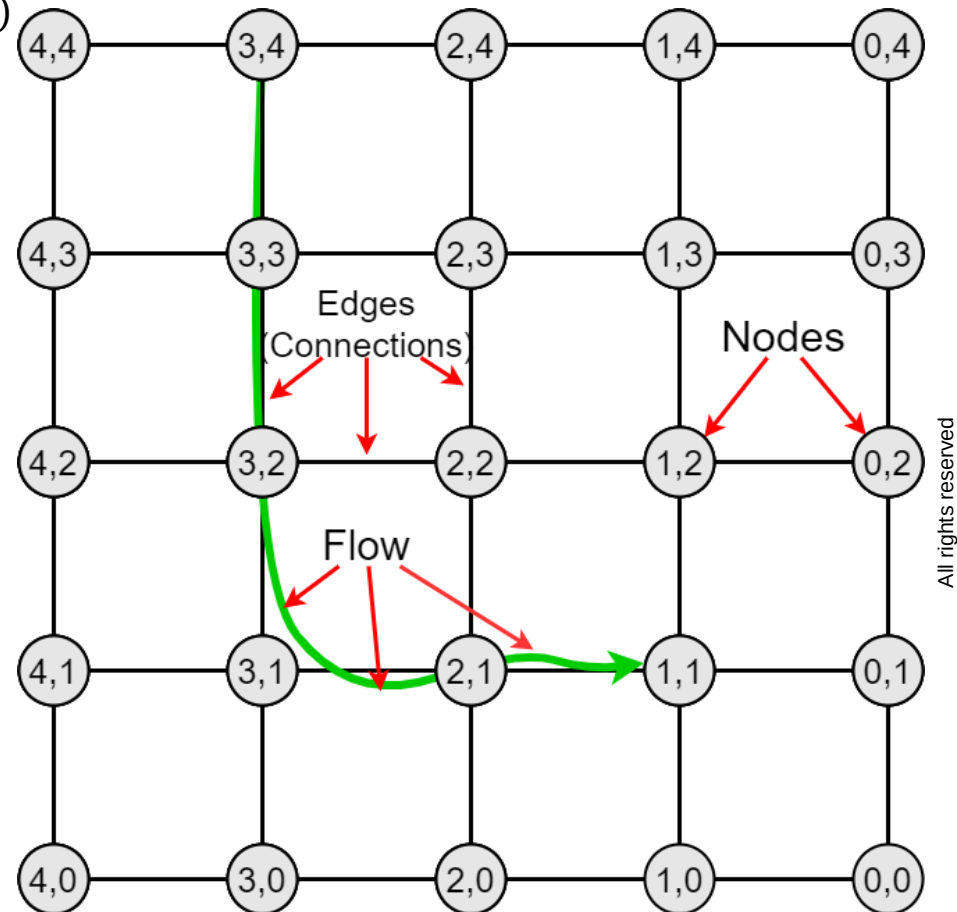
- WMN als ungerichteter Graph $G(N, E)$ [4]
- N = WMN-Knoten, E = Verbindungen



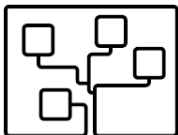
- WMN als ungerichteter Graph $G(N, E)$ [4]
- N = WMN-Knoten, E = Verbindungen
- Slice als Teilgraph $G_s(N_s, E_s) \subseteq G(N, E)$



- WMN als ungerichteter Graph $G(N, E)$ [4]
- N = WMN-Knoten, E = Verbindungen
- Slice als Teilgraph $G_s(N_s, E_s) \subseteq G(N, E)$
- Kommunikationsbeziehung zwischen Knoten: Flow f
- Maximal konsumierbare Bitrate $B(e)$

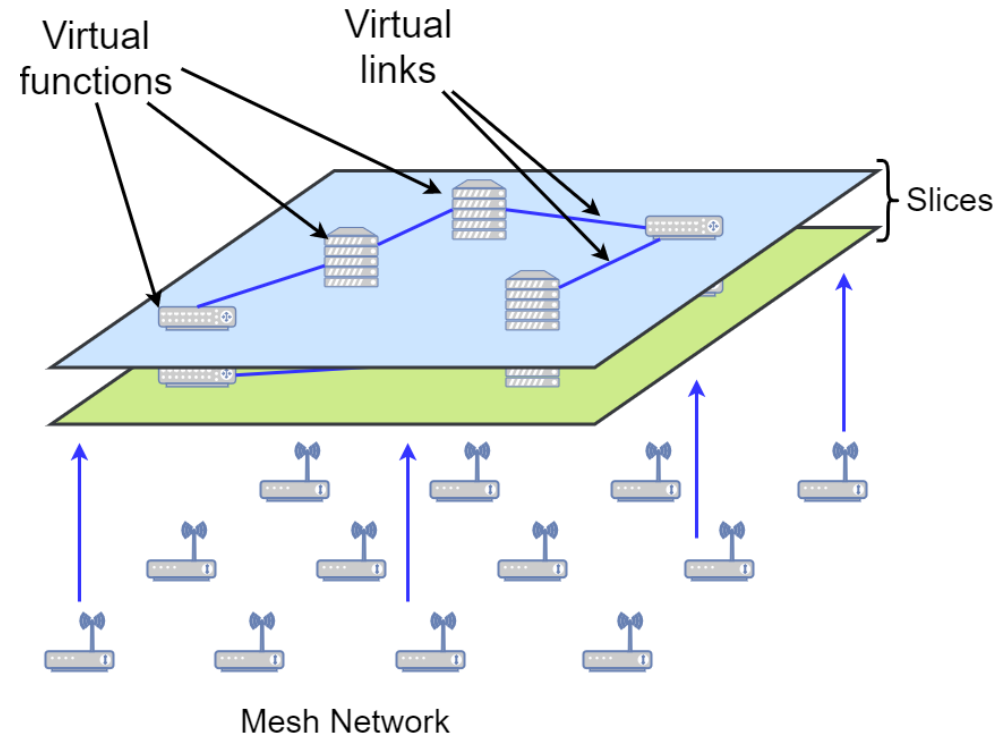


All rights reserved

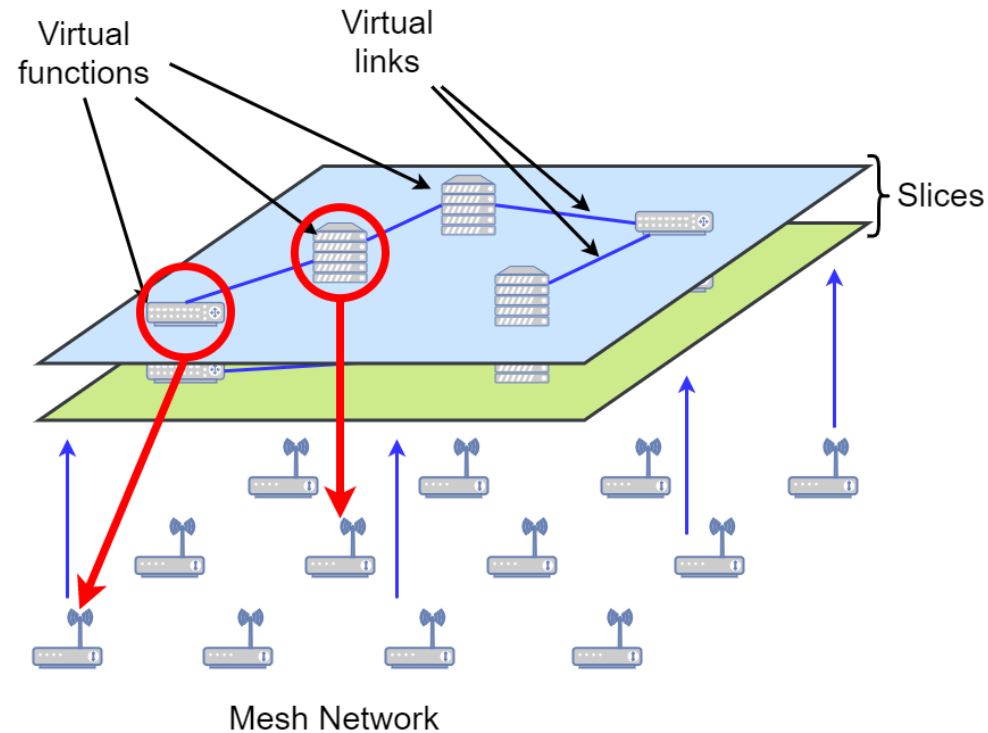


3 Pfadermittlung und Testaufbau

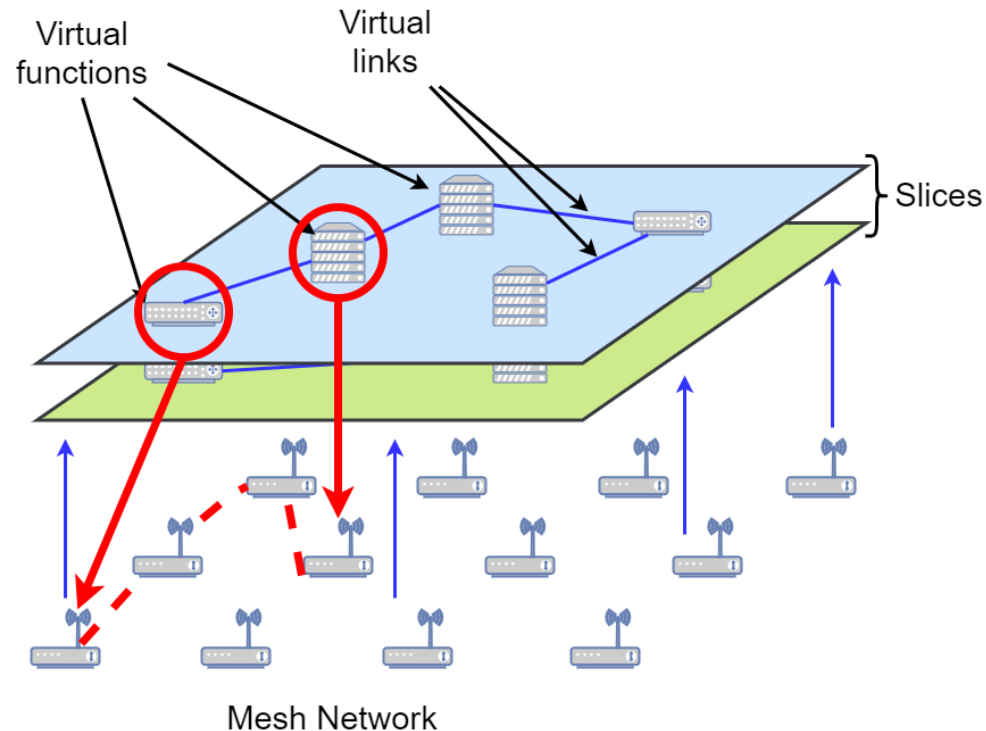
- Zweck: Übertragen der virtuellen Verbindungen auf die Infrastruktur



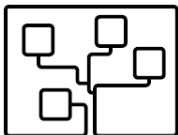
- Zweck: Übertragen der virtuellen Verbindungen auf die Infrastruktur
- Bestmögliche Verbindungen



- **Zweck: Übertragen der virtuellen Verbindungen auf die Infrastruktur**
- **Bestmögliche Verbindungen**
- **Drei verschiedene Algorithmenarten**
 - **Uninformierte Suche: Breadth-First Search (BFS)**
 - **Greedy-Algorithmus: Dijkstra**
 - **Greedy-Algorithmus mit Heuristik: A***



- **Graph G : 25 Knoten, 40 Verbindungen**
- **$B(e) = 100 \text{ Mbit/s}$**
- **Zwei unterschiedliche Tests:**
 - **Erster Test:**
 - **Flows mit einer festgelegten Bitrate hinzugefügt**
 - **Werte aufsummiert und den Verbindungen hinzugefügt**
 - **Testpfad zwischen den Knoten 1,0 und 3,4 berechnet**
 - **Berechnung der maximalen Bitrate auf dem Pfad**
 - **Zweiter Test:**
 - **Hinzufügen zufälliger Flows zum Netz**
 - **Prüfen, wann der erste überlastete Link auftaucht**



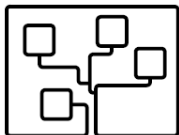
4 Ergebnisse und zukünftige Arbeit

- **Drei unterschiedliche Setups**
- **Setup 1 und 2: Anzahl der Pfade gleich, Bitrate verdoppelt**
- **Setup 3: fünf zusätzliche Flows**

Flows	From	To	Rate 1	Rate 2
f1	4,0	1,4	15	30
f2	4,3	0,3	10	20
f3	3,0	3,4	15	30
f4	1,1	2,3	20	40
f5	0,4	3,1	10	20
f6	2,4	4,1	10	20
f7	1,0	3,4	15	30
f8	4,3	2,0	20	40
f9	2,2	4,1	10	20
f10	2,2	0,0	10	20

Flows	From	To	Rate
f1	4,0	1,4	30
f2	4,3	0,3	20
f3	3,0	3,4	30
f4	1,1	2,3	40
f5	0,4	3,1	20
f6	2,4	4,1	20
f7	1,0	3,4	30
f8	4,3	2,0	40
f9	2,2	4,1	20
f10	2,2	0,0	20
f11	2,4	2,1	15
f12	0,3	4,0	20
f13	4,1	2,3	15
f14	3,1	1,4	30
f15	3,4	0,1	15

All rights reserved

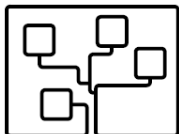


- Ergebnis: A* und Dijkstra nah beieinander
- Unterschied zu BFS deutlich
- Gleiches im zweiten Test
- A* und Dijkstra gleichauf

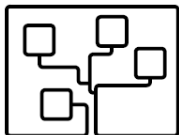
Test Setup	Algorithmus	Max. bitrate	Overloaded links
Setup 1	A*	20	0
	Dijkstra	20	0
	BFS	40	0
Setup 2	A*	30	0
	Dijkstra	40	0
	BFS	80	1
Setup 3	A*	70	0
	Dijkstra	60	0
	BFS	125	4

All rights reserved

Algorithmus	Max. paths
A*	31
Dijkstra	31
BFS	16



- **Verwendung des Dijkstra-Algorithmus für weitere Arbeiten**
- **In Zukunft: Optimale Platzierung der virtuellen Netzwerkfunktionen**
- **Kombinierung mit der Pfadermittlung**
- **Weiterer Aspekt: Ressourcenaufteilung zwischen Zugangspunkt und Endgeräten**

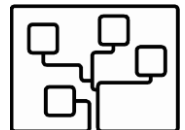


Path Determination for Network Slicing in Wireless Mesh Disaster Networks

Alexander Seng

seng@e-technik.org

Frankfurt University of Applied Sciences, Germany
Research Group for Telecommunication Networks



Referenzen

- [1] 5GPPP. View on 5g architecture: 5gppp architecture working group: Version 2.0, december 2017
- [2] IEEE, editor. IEEE standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements: Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications ; Amendment 10: Mesh networking. Institute of Electrical and Electronics Engineers, New York, 2011
- [3] G. Frick, A. Paguem Tchinda, B. Shala, U. Trick, A. Lehmann, and B. Ghita. Requirements for a distributed nfv orchestration in a wmn-based disaster network. International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), 2019.
- [4] Namwon An, Yonggang Kim, Juman Park, Dae-Hoon Kwon, and Hyuk Lim. Slice management for quality of service differentiation in wireless network slicing. Sensors, 19(12), 2019

