

Fed4FIRE+ Experiment Report

Author
Fred Stefan

Date
20.11.2020

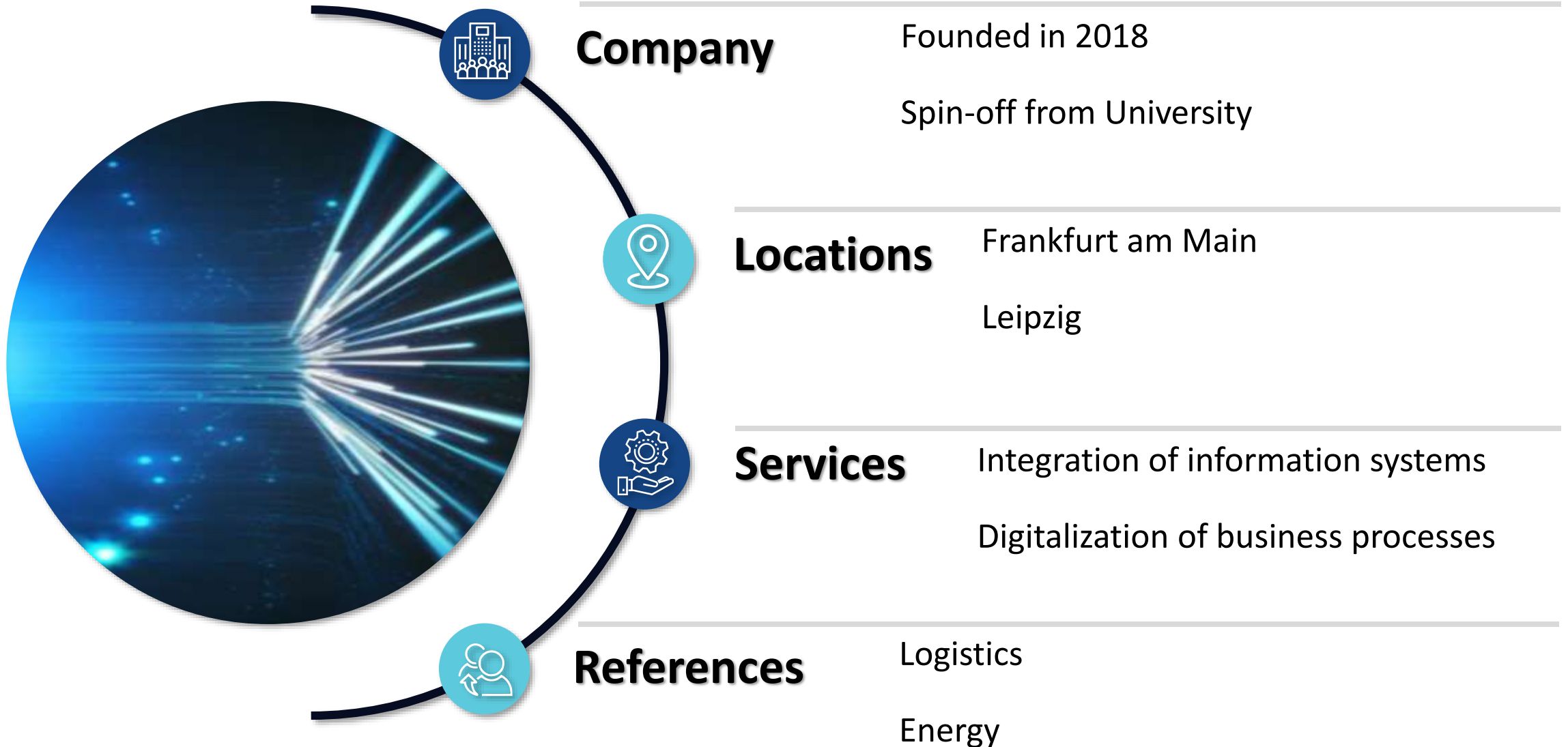


MANET4E

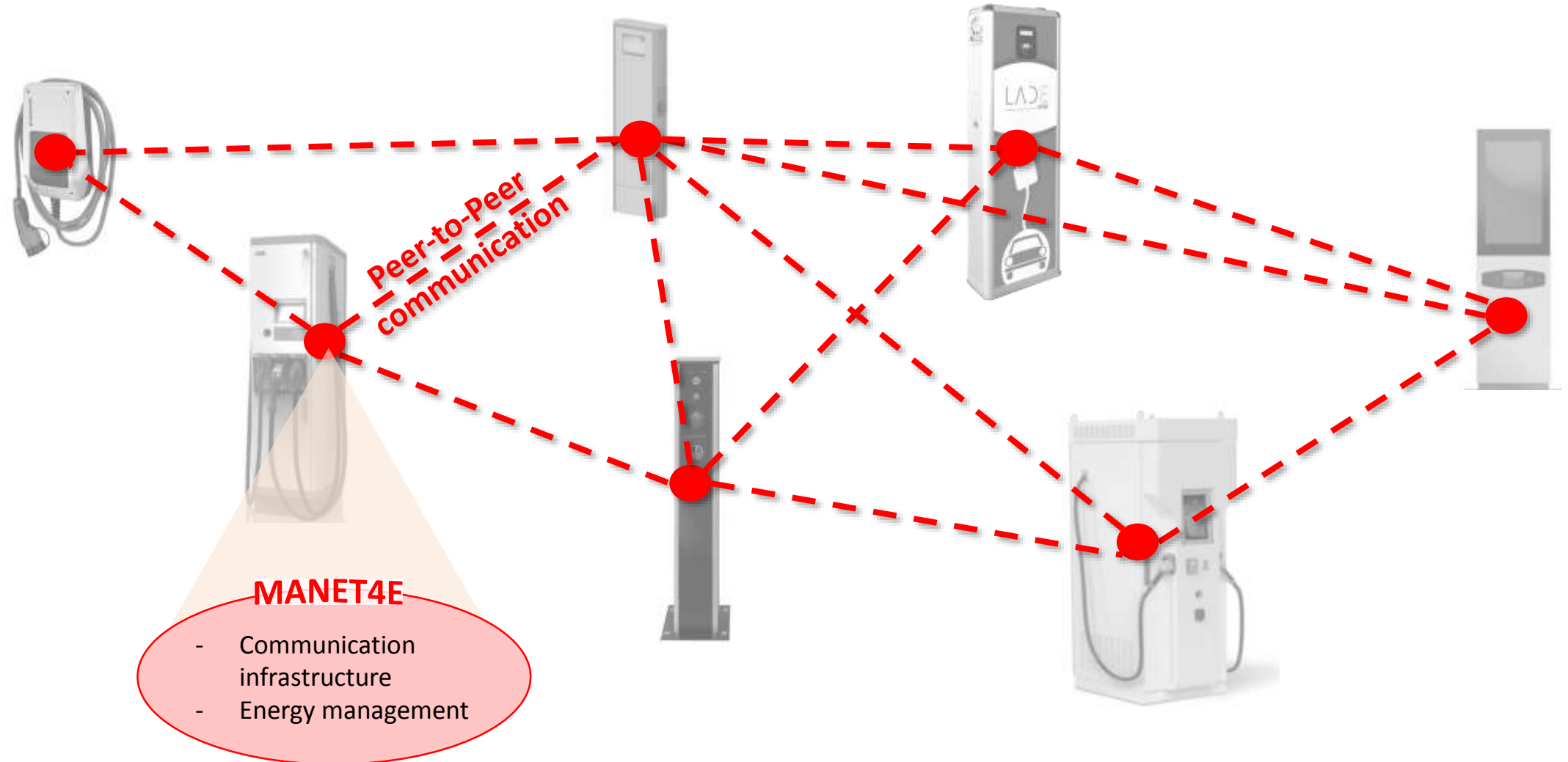
Mobile ad-hoc peer-to-peer network for self-organizing energy management systems

Stage 1 experiment

Runtime: 01.07.2020 - 30.09.2020



Energy management system for charging stations



Transfer of the MANET4E approach to other consumer and prosumer units

Explosion of communication and control complexity

Conventional power plants



New prosumer units



> 100.000.000 units

- 3,6 million companies
 - 40 million houses
 - 44 million cars
- => 100 million consumer and prosumer units to be controlled

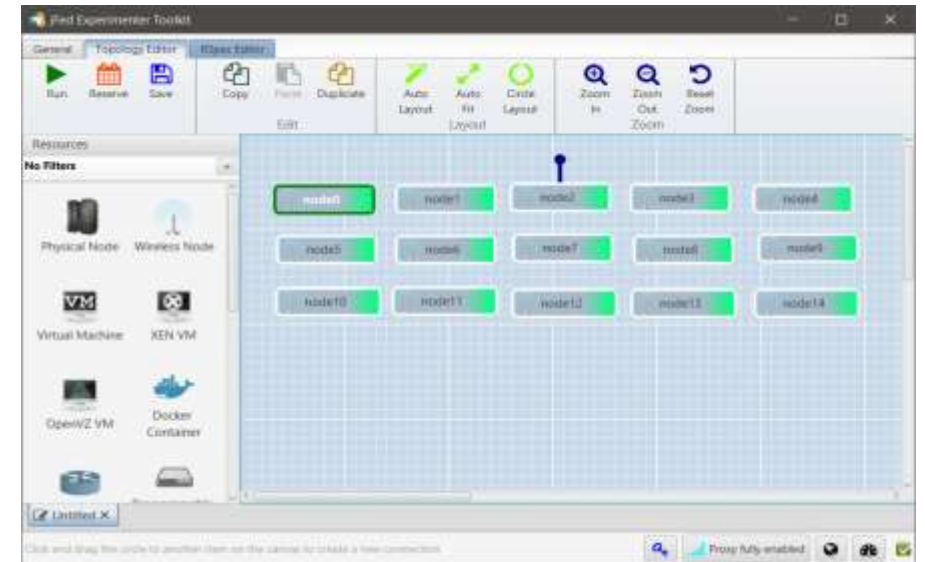
New consumer units





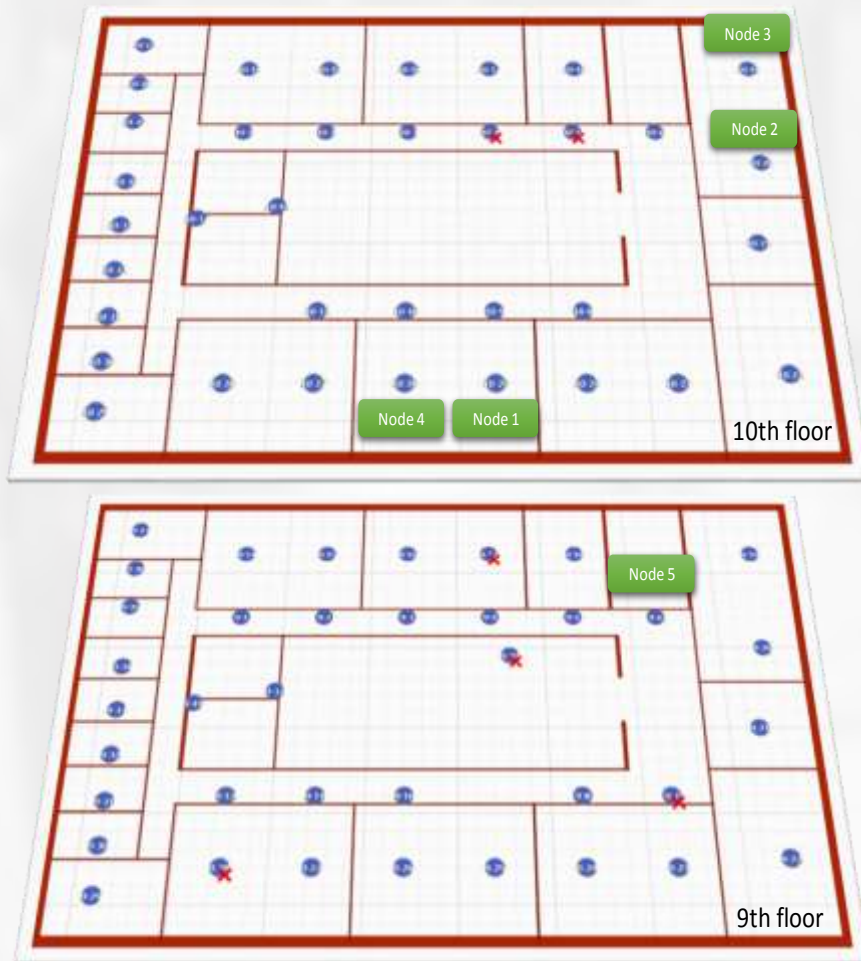
- **Enlargement of the test field**
- **Quantitative evaluation**
 - Performance
 - Reachability
 - Reliability
- **Behavior of the ad-hoc network in structural dynamics**
 - Node density
 - Error rates

- **Testbed:** w-iLab.1
- Experiment resource specification with JFed
- **Node setup**
 - Intel NUC devices
 - Ubuntu 18.04.2 LTS
 - WIFI interface 802.11abgn
 - BATMAN batctl debian-2018.0-1 [batman-adv: 2017.4]

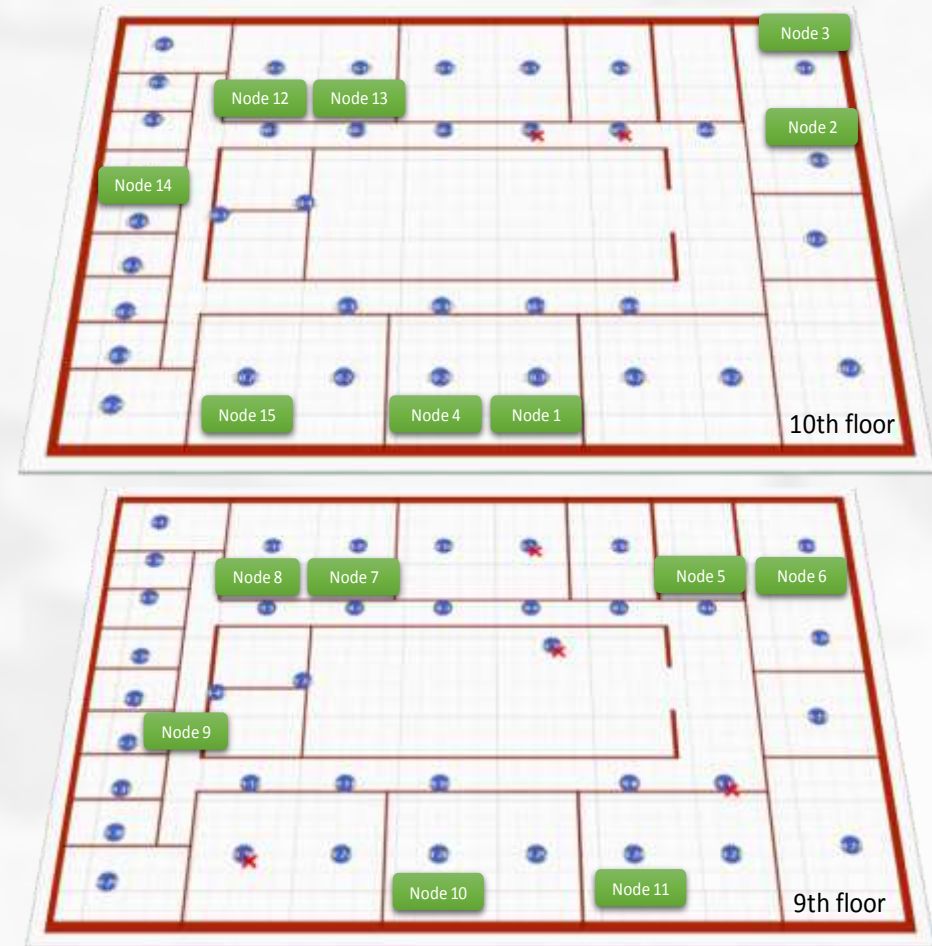


Net topologies

5-MeshNetwork



15-MeshNetwork



Measurement procedure

Ping

```
dsi@node1:~$ ping 192.168.0.15
PING 192.168.0.15 (192.168.0.15) 56(84) bytes of data.
64 bytes from 192.168.0.15: icmp_seq=1 ttl=64 time=2.98 ms
64 bytes from 192.168.0.15: icmp_seq=2 ttl=64 time=3.96 ms
64 bytes from 192.168.0.15: icmp_seq=3 ttl=64 time=4.09 ms
64 bytes from 192.168.0.15: icmp_seq=4 ttl=64 time=3.37 ms
64 bytes from 192.168.0.15: icmp_seq=5 ttl=64 time=9.08 ms
...
--- 192.168.0.15 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19028ms
rtt min/avg/max/mdev = 2.986/7.219/27.756/6.519 ms
```

Traceroute

```
dsi@node1:~$ sudo batctl traceroute 192.168.0.15

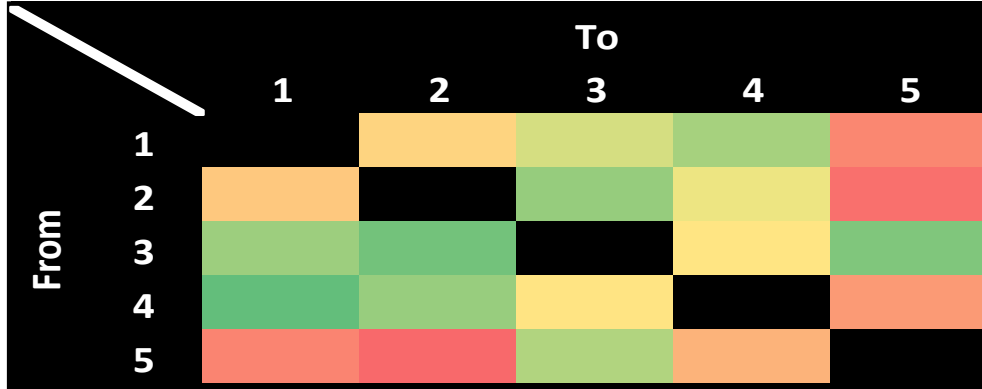
traceroute to 192.168.0.15 (00:0e:8e:6b:02:fd), 50 hops max, 20 byte packets
 1: 00:0e:8e:6b:2b:89  4.692 ms  1.848 ms  1.688 ms
 2: 00:0e:8e:6b:02:fd  3.203 ms  3.041 ms  11.112 ms
```

iPerf

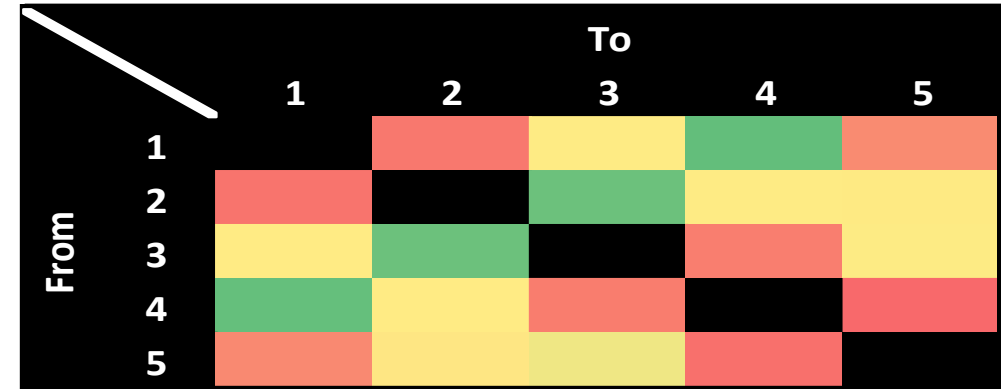
```
dsi@node1:~$ iperf -c 192.168.0.15
-----
Client connecting to 192.168.0.15, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[  3] local 192.168.0.11 port 34382 connected with 192.168.0.15 port 5001
[ ID] Interval          Transfer          Bandwidth
[  3]  0.0-10.4 sec  1.25 MBytes  1.01 Mbits/sec
```

Measurement in 5-MeshNet

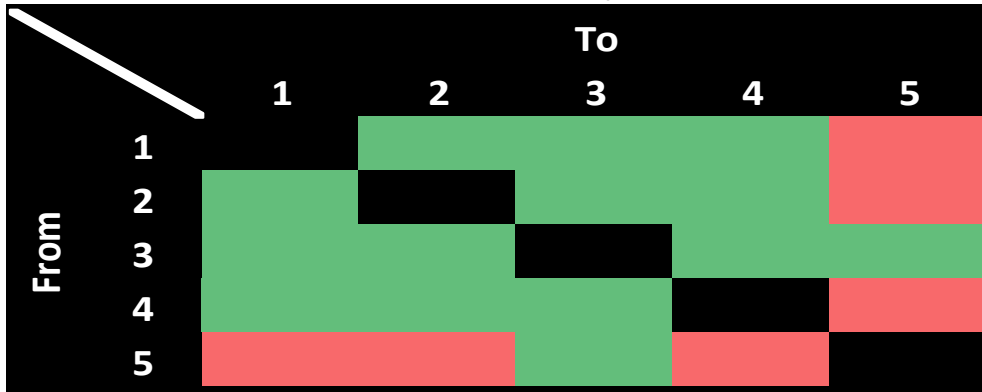
Average ping time in ms



Bandwith in Mbits/sec



Traceroute hops



Ping time

- MeshNet was between 1 and 8 ms
- Good ping over WiFi is between 1 and 3 ms

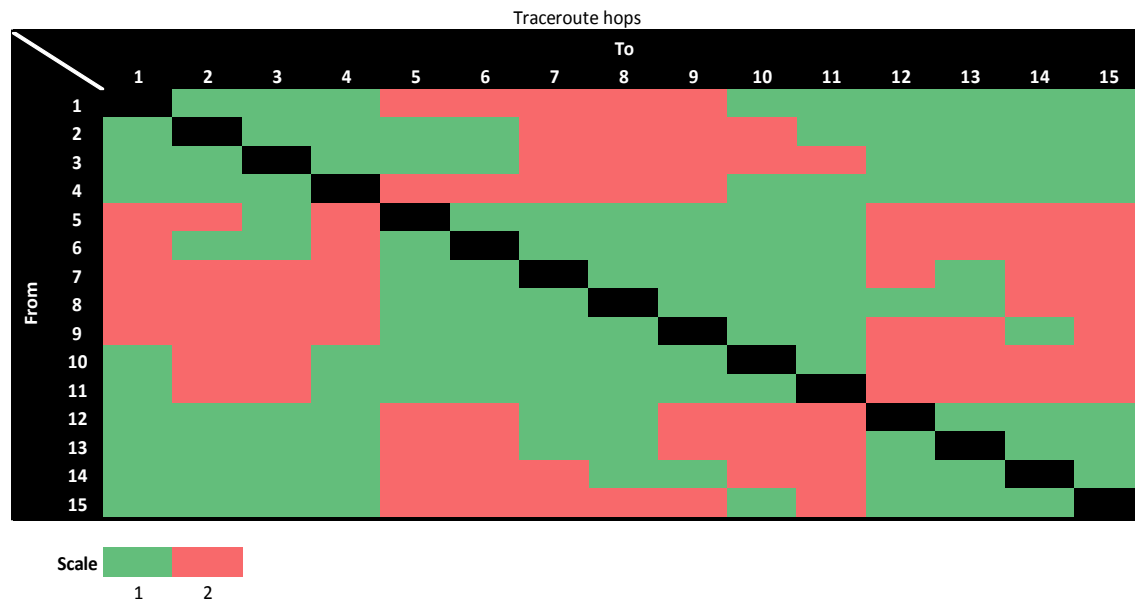
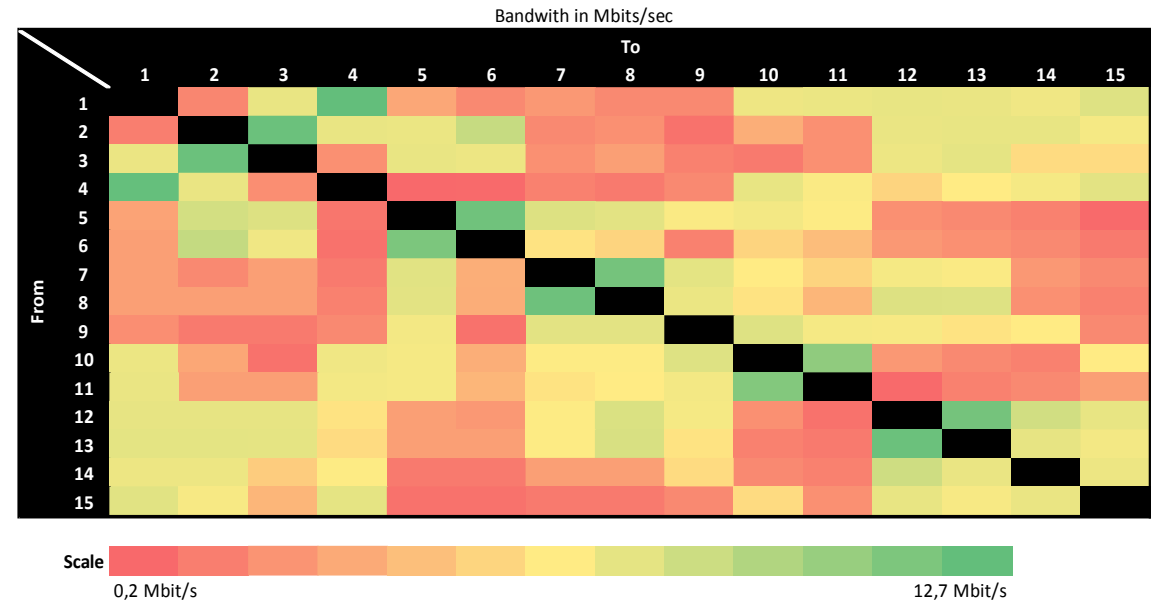
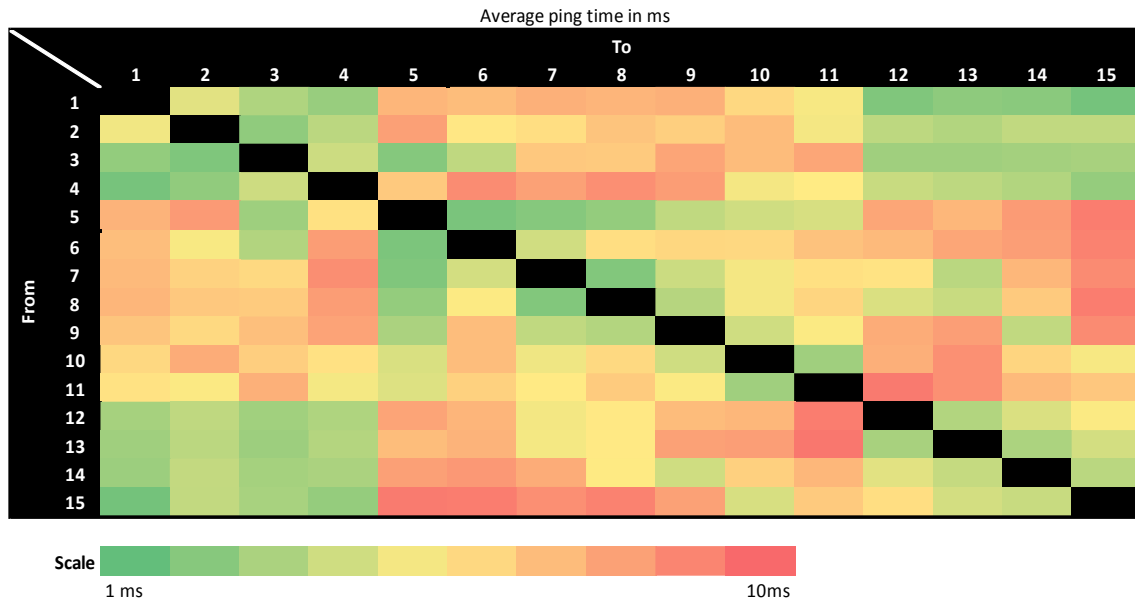
Traceroute

- Max 2 hops because close distance to each other
- Hops were between floor 9 and 10

Transfer rate

- 0.2 – 12.7 Mbits/s
- Depends on WIFI signal -> distance and walls between nodes

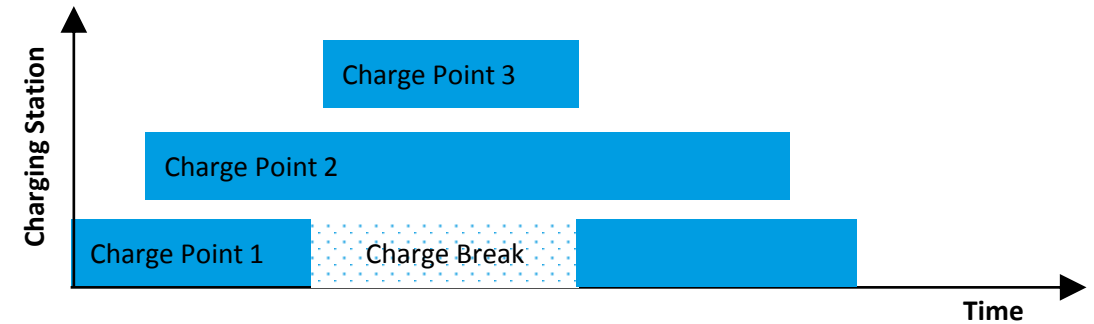
Measurement in 15-MeshNet



- Results similiar to 5-MeshNet
- Depends on WIFI signal (walls, distance, floors)

Charging schedule

- A charging schedule contains the planned energy consumption of all charging station in next time
- Each charging station needs this schedule
- Data size of charging schedule depends on the amount of stations
- Data size of blocks in the blockchain increases with the amount of exchanges



```

Ladeskule: 1
ScheduleEntry(time=0, currentOutput=20, remainingParkingDuration=10, currentLoadQuantity=20, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=1, currentOutput=10, remainingParkingDuration=9, currentLoadQuantity=30, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=2, currentOutput=3, remainingParkingDuration=8, currentLoadQuantity=50, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=3, currentOutput=3, remainingParkingDuration=7, currentLoadQuantity=53, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=4, currentOutput=3, remainingParkingDuration=6, currentLoadQuantity=56, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=5, currentOutput=3, remainingParkingDuration=5, currentLoadQuantity=59, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=6, currentOutput=30, remainingParkingDuration=4, currentLoadQuantity=72, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=7, currentOutput=30, remainingParkingDuration=3, currentLoadQuantity=73, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=8, currentOutput=30, remainingParkingDuration=2, currentLoadQuantity=87, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=9, currentOutput=30, remainingParkingDuration=1, currentLoadQuantity=92, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=10, currentOutput=0, remainingParkingDuration=0, currentLoadQuantity=100, totalLoadQuantity=100, scheduleState=SCHEDULED_FINISHED)
Ladeskule: 2
ScheduleEntry(time=1, currentOutput=30, remainingParkingDuration=3, currentLoadQuantity=70, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=2, currentOutput=30, remainingParkingDuration=2, currentLoadQuantity=80, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=3, currentOutput=30, remainingParkingDuration=1, currentLoadQuantity=90, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=4, currentOutput=0, remainingParkingDuration=0, currentLoadQuantity=100, totalLoadQuantity=100, scheduleState=SCHEDULED_FINISHED)
Ladeskule: 3
ScheduleEntry(time=2, currentOutput=7, remainingParkingDuration=3, currentLoadQuantity=80, totalLoadQuantity=100, scheduleState=CHARGED)
ScheduleEntry(time=3, currentOutput=7, remainingParkingDuration=2, currentLoadQuantity=87, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=4, currentOutput=7, remainingParkingDuration=1, currentLoadQuantity=94, totalLoadQuantity=100, scheduleState=SCHEDULED)
ScheduleEntry(time=5, currentOutput=0, remainingParkingDuration=0, currentLoadQuantity=100, totalLoadQuantity=100, scheduleState=SCHEDULED_FINISHED)

```

Example

- Network with 30 nodes
 - After 12 exchanges
 - Worst case
- > size of a schedule is ~2.5 Kbytes
- > size of schedule is 2.5Kbytes x 12 exchanges = 30 Kbytes
- > transfer rate is 0.2 Mbit/s = 25 Kbytes/s
- First exchange: $2.5 \text{ Kbytes} / 25 \text{ Kbytes/s} = 0.1\text{s}$
 - After 12 exchanges: $30 \text{ Kbytes} / 25 \text{ Kbytes/s} = 1.2\text{s}$

Summary

- **Fed4FIRE+** is a very useful environment for experiments
- **Hardware as a service** is very convenient (saves time and money)
- We could investigate the scalability of our approach
 - From 3 nodes in our office up to 15 nodes with Fed4FIRE
 - The w-iLab.1 office building was ideal to emulate a parking garage
- We could learn in Stage-1-experiment how to use Fed4FIRE environment
- We derived some further requirements for our solution

Outlook for Stage 2

- **Usage of Blockchain-technologies to enable future energy trading**
- More qualitative investigation
- Investigation of the optimal block size
- Synchronization concepts
- Development of fault mechanisms
- Further upscaling

Fed4FIRE+ Experiment Report

Thank you.

Author
Fred Stefan

Date
20.11.2020

