# ResourSim
# –
# Evaluating the end-user Device usage

Jens Dede, Anna Förster

University of Bremen
Sustainable Communication Networks

7th September 2018

# Motivation

Talking about *Smart Devices*...

- Restrictions of background services ($\Rightarrow$ Doze etc.)
- Not optimal in all cases

# Motivation

Talking about *Smart Devices*...

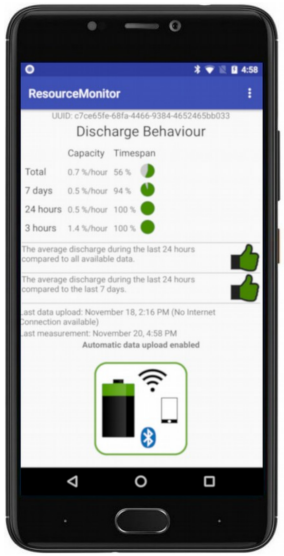
- Restrictions of background services ($\Rightarrow$ Doze etc.)
- Not optimal in all cases

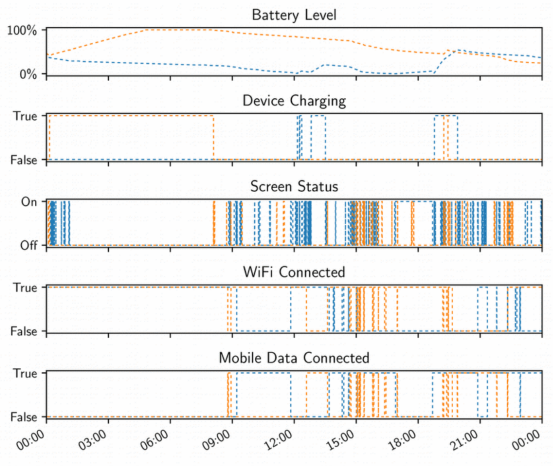
$\Rightarrow$ Can we do better: Optimize on the **real** (human) usage?

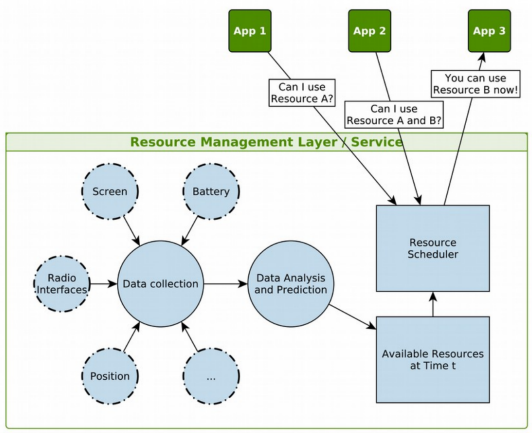
Beneficial for

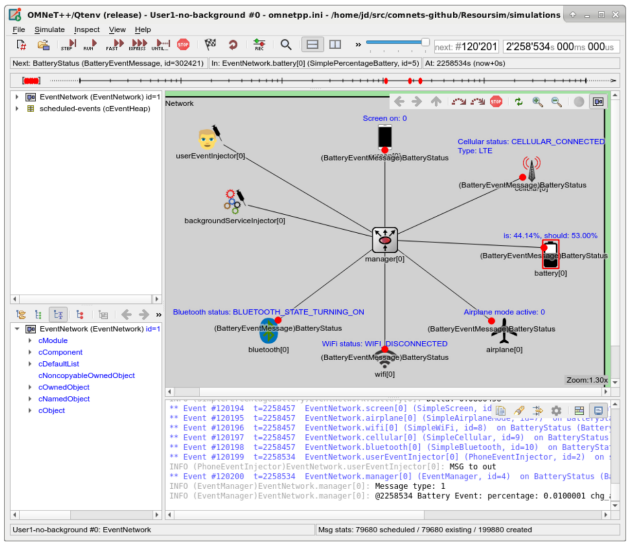
- OppNets
- Background Computing
- ...

# Motivation

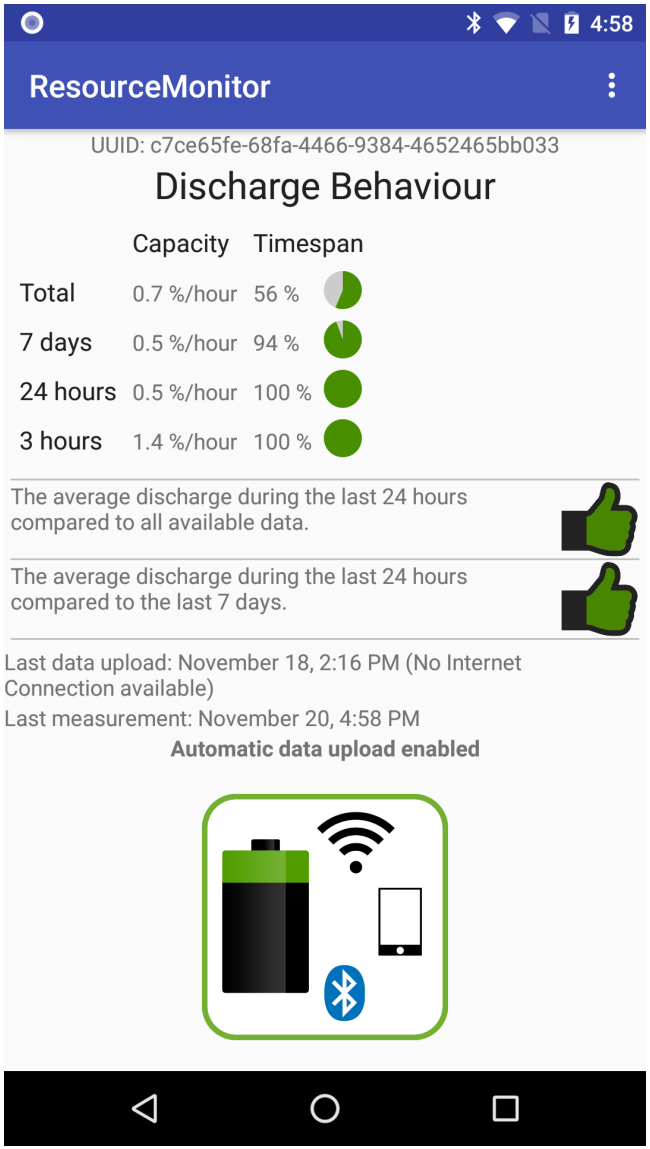

ResourceMonitor
Discharge Behaviour
Capacity Timespan
Total
7 days
24 hours
3 hours


Battery Level
100%
0%
Device Charging
True
False
Screen Status
On
Off
WiFi Connected
True
False
Mobile Data Connected
True
False
00:00
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00


App 1
App 2
App 3
Can I use Resource A?
Can I use Resource A and B?
You can use Resource B now!
Resource Management Layer / Service
Screen
Battery
Radio Interfaces
Data collection
Data Analysis and Prediction
Resource Scheduler
Position
...
Available Resources at Time t


OMNeT++/Qtenv (release) - User1-no-background #0 - omnetpp.ini - /home/jd/src/comnets-github/Resoursim/simulations
File Simulate Inspect View Help
Network
Screen on: 0
Cellular status: CELLULAR_CONNECTED
Type: LTE
Bluetooth status: BLUETOOTH_STATE_TURNING_ON
Airplane mode active: 0
WiFi status: WIFI_DISCONNECTED

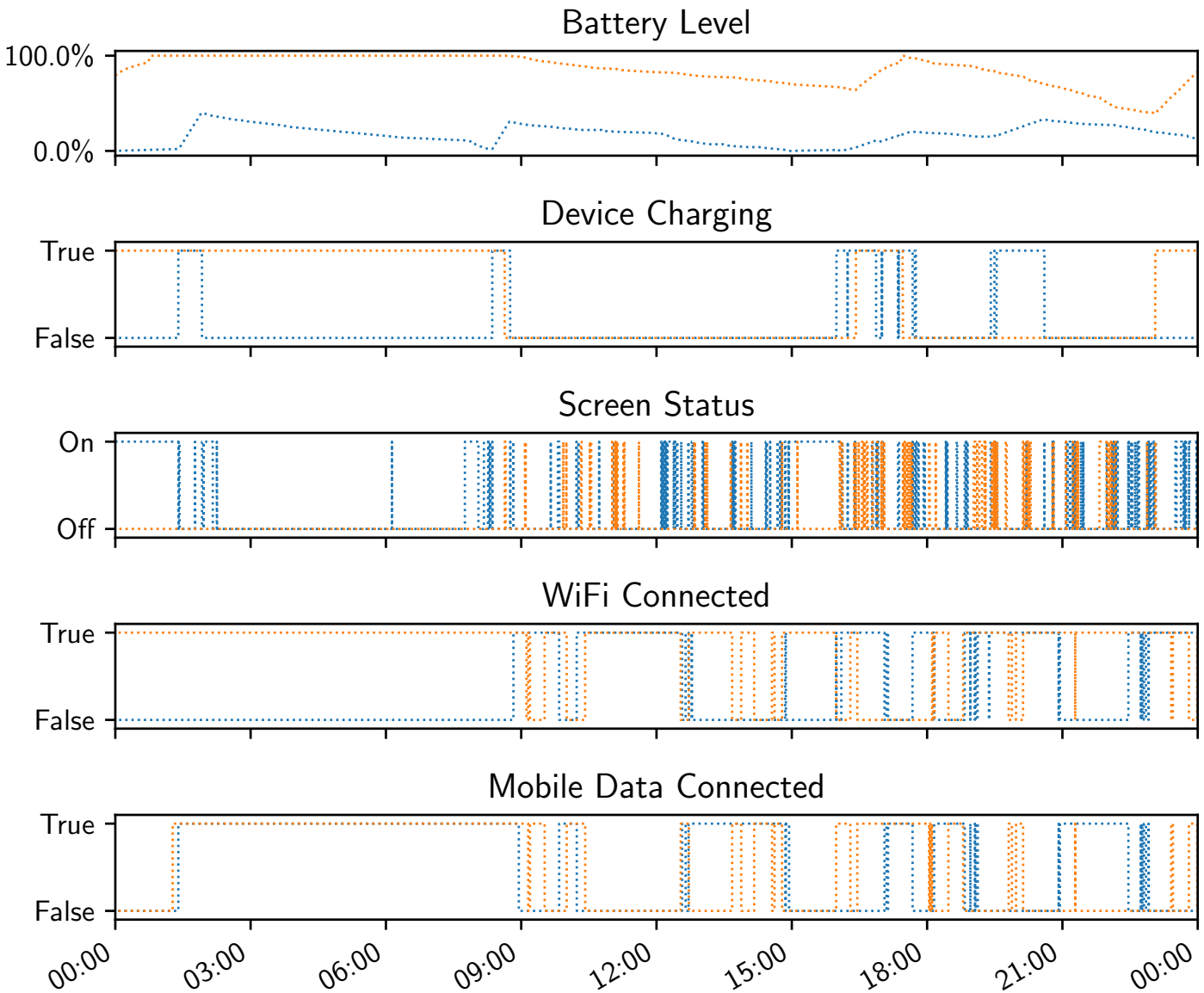
# Android App: ResourceMonitor




## Collected Information:

- Battery status
- Bluetooth interface status
- WiFi interface status
- Cellular interface status
- Amount of transmitted data
- Status of the screen

# Resource Graph for two users

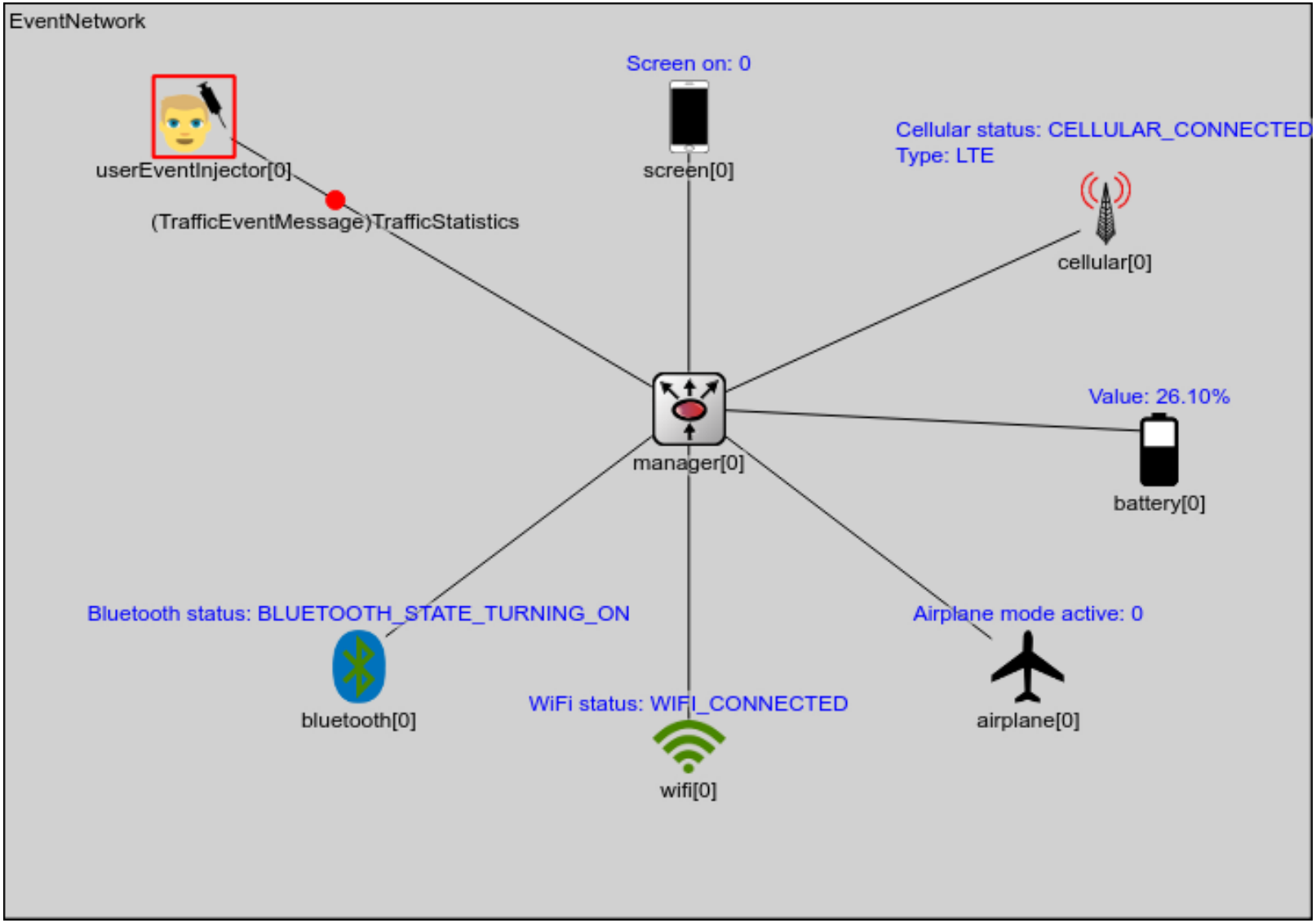


# The Event list

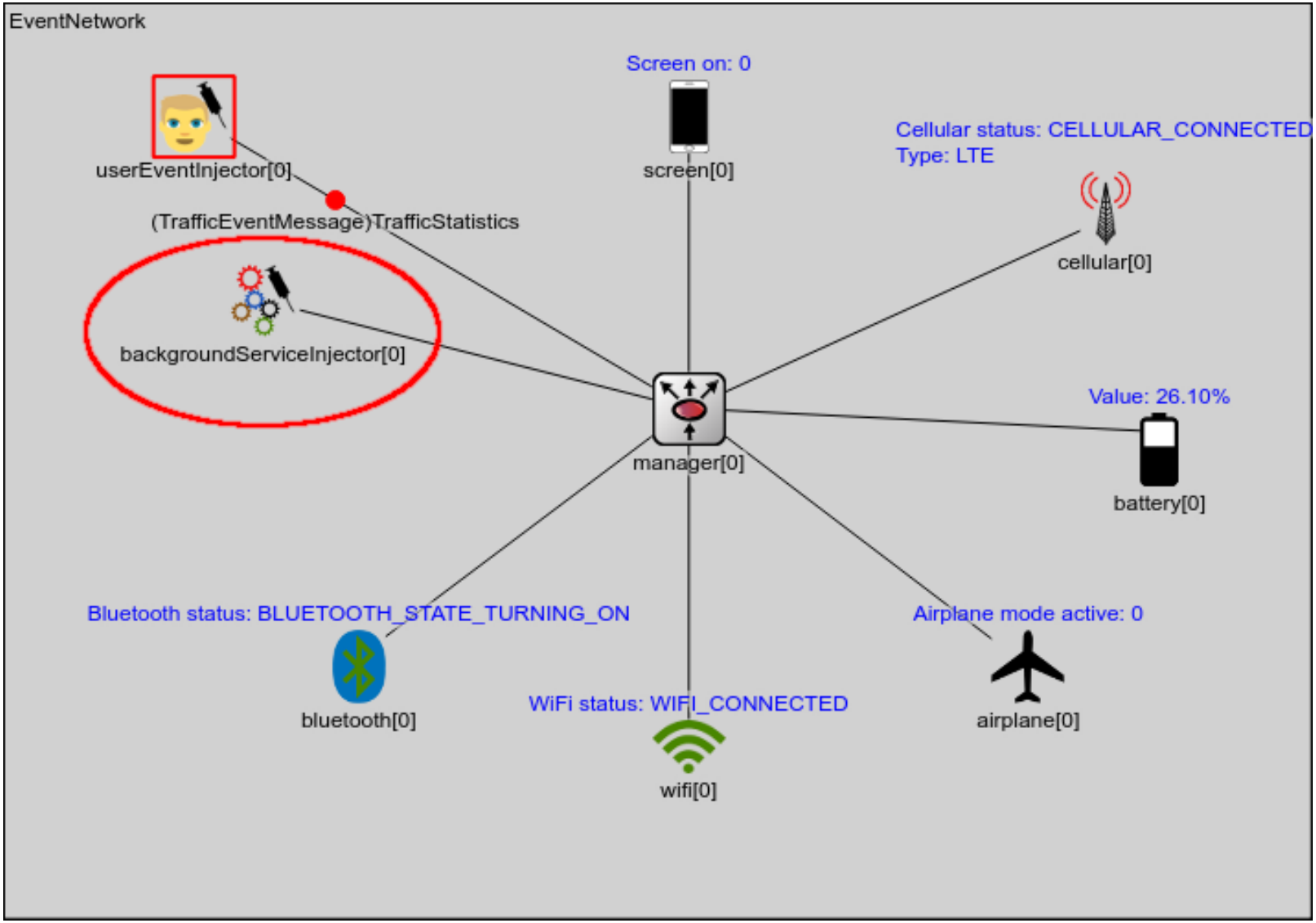
```
1 ...
2 <event chg_ac="0" chg_usb="0" chg_wireless="0" data_type
    ↪ ="BatteryStatus" delta_percentage
    ↪ ="-0.020000040531158447" is_charging="0"
    ↪ percentage="0.949999988079071" sequential_number
    ↪ ="3"timestamp_s="34201.0">34201.0</event>
3 <event data_type="WiFiStatus" sequential_number="135"
    ↪ timestamp_s="37030.0" wifi_status="12">37030.0</
    ↪ event>
4 <event data_type="ScreenStatus" screen_status="0"
    ↪ sequential_number="136" timestamp_s
    ↪ ="37032.0">37032.0</event>
5 ...
```

⇒ Use OMNeT++ for the event simulation

# ResourceSim: Replay Scenario



# ResourceSim: Inject additional Events



# Evaluation Scenarios

**Scenario I**
Replay scenario, used to evaluate the battery behaviour

**Scenario II**
Regular WiFi usage, assuming neighborhood scanning
frequency: `uniform(500s, 700s)`
duration: `truncnormal(2s, 1s)`

**Scenario III**
Scenario I + CPU usage. Data is processed regularly
frequency: `normal(3600s, 10s)`
duration: `truncnormal(10s, 5s)`

**Scenario IV**
Heavy WiFi usage every 120 s + CPU usage from Scenario III
frequency: constant: 120s
duration: `truncnormal(10s, 5s)`

# Evaluation

**In OMNeT++:**

- Run those scenarios with traces from two different users
- Count the collisions

**Collision types:**

Background Negative effect on the background service

User Negative effect on the human user

**Resources:**

- WiFi background vs. WiFi human user
- CPU usage vs. Screen status

# Evaluation with user Traces

| | Collision Type | User 1 $[\frac{1}{\text{day}}]$ | User 2 $[\frac{1}{\text{day}}]$ |
|---|---|---|---|
| **Scenario II** | WiFi: user | 0.125 | 0.069 |
| | WiFi: background | 110.366 | 88.090 |
| | usage: user | 0.0 | 0.0 |
| | usage: background | 0.0 | 0.0 |
| **Scenario III** | WiFi: user | 0.062 | 0.094 |
| | WiFi: background | 110.420 | 88.199 |
| | usage: user | 0.187 | 0.484 |
| | usage: background | 1.572 | 5.163 |
| **Scenario IV** | WiFi: user | 4.179 | 2.682 |
| | WiFi: background | 536.560 | 437.113 |
| | usage: user | 0.241 | 0.369 |
| | usage: background | 1.712 | 5.104 |

# Results

- Effect on the human user is lower compared to the effect on the background service
- How many conflicts are acceptable?
- How to handle conflicts?

Main challenges:

⇒ **Effect on the battery?**
⇒ **Model which matches the real battery behaviour?**
(⇒ with adequate complexity)

# Battery: Methodology

- Extract the battery charging / discharging behaviour from the trace
- Use those to parameterize a battery model

**Approaches:**

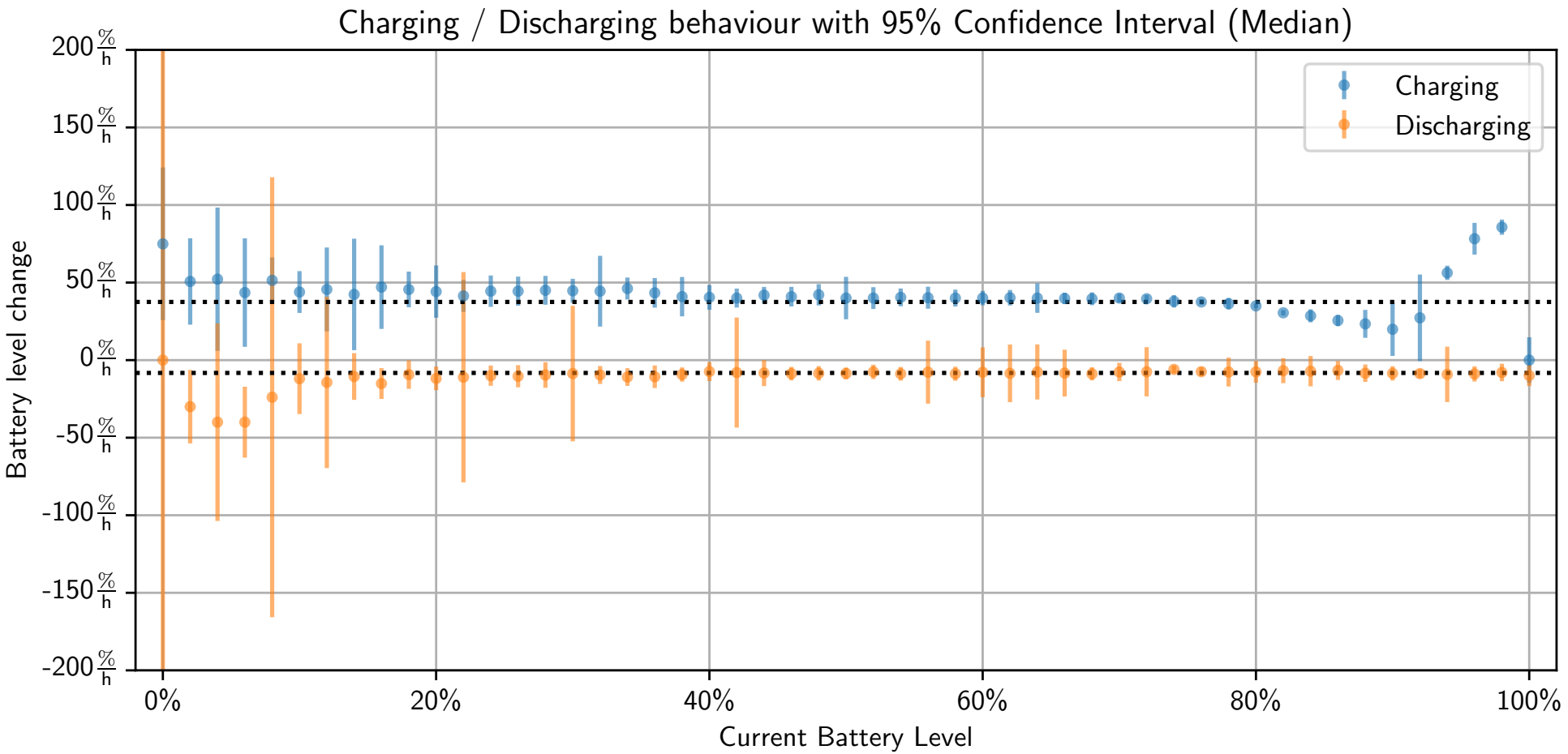
- Mean values
- Median values
- Array of mean values
- Array of median values

⇒ Use Scenario I for the evaluation
⇒ Use mean difference between recorded and calculated value as metric

# Mean and Median Values

| | | **User 1** $[\frac{\%}{hour}]$ | **User 2** $[\frac{\%}{hour}]$ |
|---|---|---|---|
| Mean Values | Charging | 47.85 | 58.59 |
| | Discharging | -23.95 | -17.78 |
| Median Values | Charging | 37.50 | 56.25 |
| | Discharging | -8.2 | -7.96 |

# Mean and Median Values



# Difference: Model vs. real Values

| **Setup** | **Delta to real value** | |
|---|---|---|
| | Mean ($\mu$) | Standard deviation ($\sigma$) |
| User 1, mean values | 56.97 % | 22.77 % |
| User 1, median values | 44.55 % | 20.17 % |
| User 1, array of mean values | 56.06 % | 20.97 % |
| User 1, array of median values | 43.92 % | 20.00 % |
| User 2, mean values | 42.06 % | 26.54 % |
| User 2, median values | 45.76 % | 27.50 % |
| User 2, array of mean values | 42.02 % | 26.80 % |
| User 2, array of median values | 45.31 % | 27.64 % |

# Difference: Model vs. real Values

| **Setup** | **Delta to real value** | |
|---|---|---|
| | Mean ($\mu$) | Standard deviation ($\sigma$) |
| User 1, mean values | 56.97 % | 22.77 % |
| User 1, median values | 44.55 % | 20.17 % |
| User 1, array of mean values | 56.06 % | 20.97 % |
| User 1, array of median values | 43.92 % | 20.00 % |
| User 2, mean values | 42.06 % | 26.54 % |
| User 2, median values | 45.76 % | 27.50 % |
| User 2, array of mean values | 42.02 % | 26.80 % |
| User 2, array of median values | 45.31 % | 27.64 % |

$\Rightarrow$ **How to do better?**

# Conclusion

- Introduced ResourceSim to evaluate the effect of background service on the real user
- Real usage traces
- Shown effect in example use cases
- Battery modeling based on real traces requires more investigation

# Future Work

- Further analyze battery charging and discharging behaviour
- Implement a more realistic battery model based on real traces
- Evaluate more complex scenarios
- **Test and evaluate optimization algorithms**

# Source Code

- ResourSim:
  `github.com/ComNets-Bremen/Resoursim`
- ResourceMonitor:
  `github.com/ComNets-Bremen/ResourceMonitor`
- ResourceMonitor on Google Play:
  `play.google.com/store/apps/details?`
  `↪id=de.uni_bremen.comnets.resourcemonitor`

# Thank You!