

# Projekt Netzwerk-Monitoring: BATMAN und collected u23 2014

mraerino, esssing

Chaos Computer Club Cologne e.V.  
<https://koeln.ccc.de>

2014-11-27



- 1 Mesh-Netzwerke und BATMAN
- 2 collectd und das BATMAN-Plugin



# 1 Mesh-Netzwerke und BATMAN

## 2 collected und das BATMAN-Plugin



# Was ist ein Mesh-Netzwerk im Freifunk-Sinn?



# Was ist ein Mesh-Netzwerk im Freifunk-Sinn?

## Begriffe:

- Nodes
- Clients
- Point-to-Point
- Uplink
- Neighbour



# Was ist ein Mesh-Netzwerk im Freifunk-Sinn?

## Begriffe:

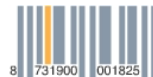
- Nodes
- Clients
- Point-to-Point
- Uplink
- Neighbour



# Was ist ein Mesh-Netzwerk im Freifunk-Sinn?

## Begriffe:

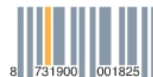
- Nodes
- Clients
- Point-to-Point
- Uplink
- Neighbour



# Was ist ein Mesh-Netzwerk im Freifunk-Sinn?

## Begriffe:

- Nodes
- Clients
- Point-to-Point
- Uplink
- Neighbour

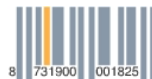




# Was ist ein Mesh-Netzwerk im Freifunk-Sinn?

## Begriffe:

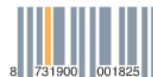
- Nodes
- Clients
- Point-to-Point
- Uplink
- Neighbour



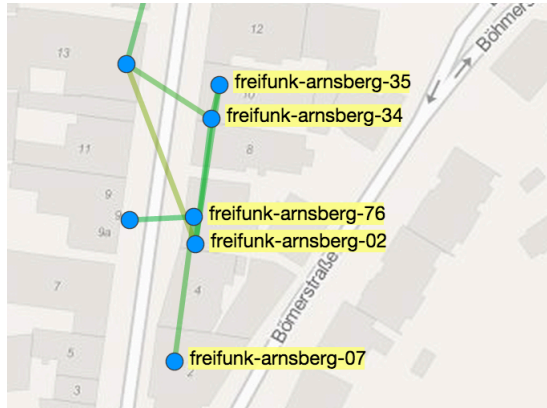
# Was ist ein Mesh-Netzwerk im Freifunk-Sinn?

## Begriffe:

- Nodes
- Clients
- Point-to-Point
- Uplink
- Neighbour



# Was ist ein Mesh-Netzwerk im Freifunk-Sinn?



# Was ist BATMAN?

- Layer 2
- dynamisch
- keine Graphen
- Begriffe: OGM (originator message)



# Was ist BATMAN?

- Layer 2
- dynamisch
- keine Graphen
- Begriffe: OGM (originator message)



# Was ist BATMAN?

- Layer 2
- dynamisch
- keine Graphen
- Begriffe: OGM (originator message)



# Was ist BATMAN?

- Layer 2
- dynamisch
- keine Graphen
- Begriffe: OGM (originator message)



# Was ist BATMAN?

- Layer 2
- dynamisch
- keine Graphen
- Begriffe: OGM (originator message)

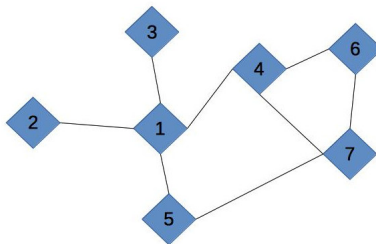




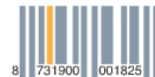
## Wie funktioniert das dynamische Routing?



# Was ist ein Mesh-Netzwerk im Freifunk-Sinn? - Beispiel



Originator	last-seen	(#/255)	Nexthop [outgoingIF]:
fe:fe:00:00:02:02	0.150s	(254)	fe:fe:00:00:02:02 [ wlan0]:
fe:fe:00:00:02:03	0.210s	(254)	fe:fe:00:00:02:03 [ wlan0]:
fe:fe:00:00:02:04	0.560s	(231)	fe:fe:00:00:02:04 [ wlan0]:
fe:fe:00:00:02:05	0.760s	(250)	fe:fe:00:00:02:05 [ wlan0]:
fe:fe:00:00:02:06	0.100s	(201)	fe:fe:00:00:02:04 [ wlan0]:
fe:fe:00:00:02:07	0.860s	(220)	fe:fe:00:00:02:05 [ wlan0]:



- 1 Mesh-Netzwerke und BATMAN
- 2 collectd und das BATMAN-Plugin



# Was ist collectd?

- ein Daemon
- sammelt Daten über das System
- und speichert diese



# Was ist collectd?

- ein Deamon
- sammelt Daten über das System
- und speichert diese



# Was ist collectd?

- ein Deamon
- sammelt Daten über das System
- und speichert diese



# Was ist collectd?

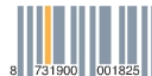
- ein Deamon
- sammelt Daten über das System
- und speichert diese



# Modularität

fast alles sind Plugins!

Name	Type	Manpage	Available since
AMQP plugin	Read, Write	<a href="#">collectd.conf(5)</a>	5.0
Apache plugin	Read	<a href="#">collectd.conf(5)</a>	3.9
APC UPS plugin	Read	<a href="#">collectd.conf(5)</a>	3.10
Apple Sensors plugin	Read		3.9
Aquaero plugin	Read	<a href="#">collectd.conf(5)</a>	5.4
Ascent plugin	Read	<a href="#">collectd.conf(5)</a>	4.4
Battery plugin	Read		3.7
BIND plugin	Read	<a href="#">collectd.conf(5)</a>	4.6
Carbon plugin	Write		4.9
cgroups plugin	Read	<a href="#">collectd.conf(5)</a>	5.4
ConnTrack plugin	Read		4.7
ContextSwitch plugin	Read		4.9
CPU plugin	Read		1.3
CPUFreq plugin	Read	<a href="#">collectd.conf(5)</a>	3.4
CSV plugin	Write	<a href="#">collectd.conf(5)</a>	4.0
cURL plugin	Read	<a href="#">collectd.conf(5)</a>	4.6
cURL-JSON plugin	Read	<a href="#">collectd.conf(5)</a>	4.8
cURL-XML plugin	Read	<a href="#">collectd.conf(5)</a>	4.10
DBI plugin	Read	<a href="#">collectd.conf(5)</a>	4.6
DE plugin	Read	<a href="#">collectd.conf(5)</a>	3.6

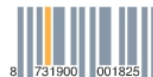




# Plugin-Architektur

Plugins müssen (bzw. können) Callbacks implementieren:

- config
- init
- log
- **read**
- write
- shutdown
- ...



# Plugin-Architektur

Plugins müssen (bzw. können) Callbacks implementieren:

- **config**
- init
- log
- **read**
- write
- shutdown
- ...



# Plugin-Architektur

Plugins müssen (bzw. können) Callbacks implementieren:

- config
- init
- log
- **read**
- write
- shutdown
- ...



# Plugin-Architektur

Plugins müssen (bzw. können) Callbacks implementieren:

- config
- init
- log
- **read**
- write
- shutdown
- ...



# Plugin-Architektur

Plugins müssen (bzw. können) Callbacks implementieren:

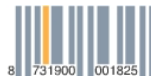
- config
- init
- log
- **read**
- write
- shutdown
- ...



# Plugin-Architektur

Plugins müssen (bzw. können) Callbacks implementieren:

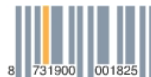
- config
- init
- log
- **read**
- write
- shutdown
- ...



# Plugin-Architektur

Plugins müssen (bzw. können) Callbacks implementieren:

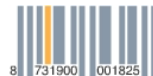
- config
- init
- log
- **read**
- write
- shutdown
- ...



# Plugin-Architektur

Plugins müssen (bzw. können) Callbacks implementieren:

- config
- init
- log
- **read**
- write
- shutdown
- ...





# round-robin database

- Datenpunkte mit konstanten Zeitabständen
- feste Datenmenge / Dateigröße
- „wrap around“, wenn die Datei „voll“ ist



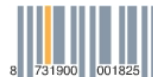
# round-robin database

- Datenpunkte mit konstanten Zeitabständen
- feste Datenmenge / Dateigröße
- „wrap around“, wenn die Datei „voll“ ist



# round-robin database

- Datenpunkte mit konstanten Zeitabständen
- feste Datenmenge / Dateigröße
- „wrap around“, wenn die Datei „voll“ ist



# round-robin database

- Datenpunkte mit konstanten Zeitabständen
- feste Datenmenge / Dateigröße
- „wrap around“, wenn die Datei „voll“ ist



# round-robin database

- Datenpunkte mit konstanten Zeitabständen
- feste Datenmenge / Dateigröße
- „wrap around“, wenn die Datei „voll“ ist



# Verwendung in FF-KBU / cserv

`https://kbu.freifunk.net/cserv`



# das Plugin

