

NooK 2024

Linus Lüssing



Die Sendung mit der Fledermaus



**B.A.T.M.A.N. Advanced: Geschichte & Technik
des Mesh-Routing-Protokolls**

1 Was ist B.A.T.M.A.N. Advanced?

2 Geschichte

3 Technik

- Routing Basics

- Layer 2 Features

WAS IST B.A.T.M.A.N. ADVANCED?



- Better Approach To Mobile Ad-hoc Networking

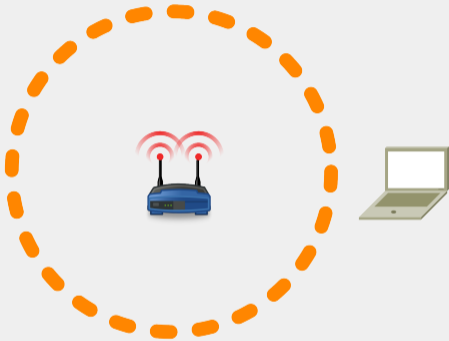


- Better Approach To Mobile Ad-hoc Networking
- Mesh Netzwerk Protokoll \Rightarrow dezentral



- Better Approach To Mobile Ad-hoc Networking
- Mesh Netzwerk Protokoll \Rightarrow dezentral
- Layer 2

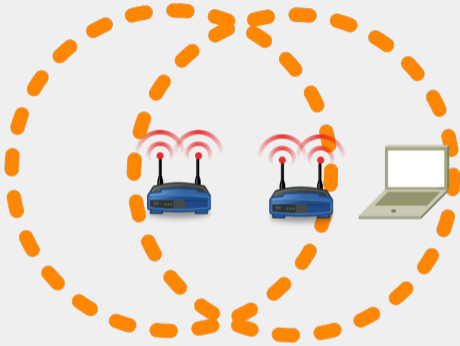
Mesh



Problem:

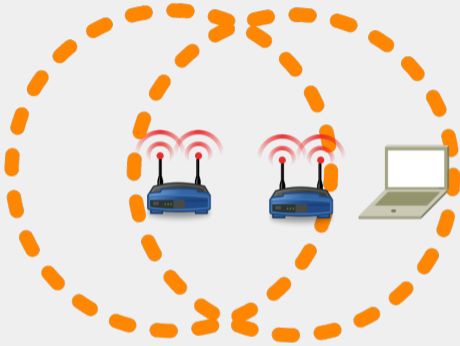
- zu kurze Funkreichweite

Mesh



Repeater:

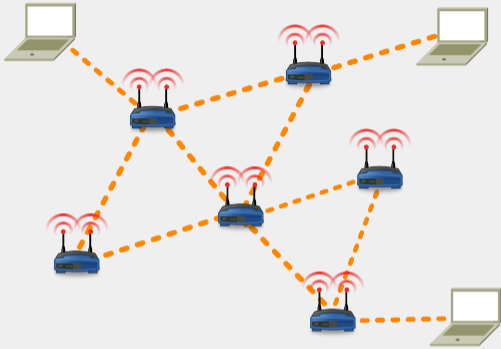
- (W)LAN Router leitet für einen anderen weiter



Repeater:

- (W)LAN Router leitet für einen anderen weiter
- statisch konfiguriert

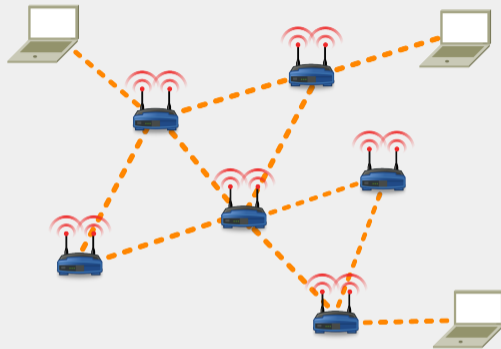
Mesh



Mesh:

- (W)LAN Router leiten für andere weiter

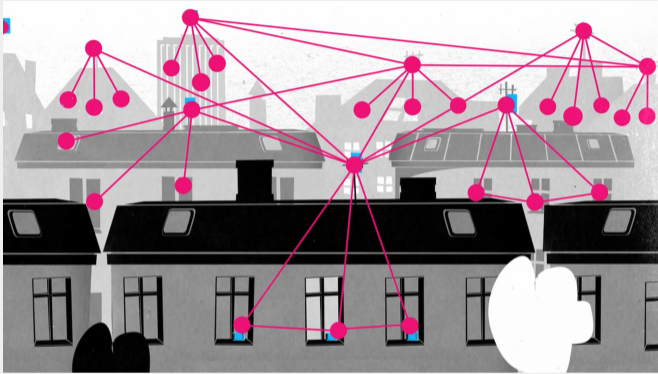
Mesh



Mesh:

- (W)LAN Router leiten für andere weiter
- dynamische Wegfindung
⇒ autonomes Navigationssystem für Datenpakete

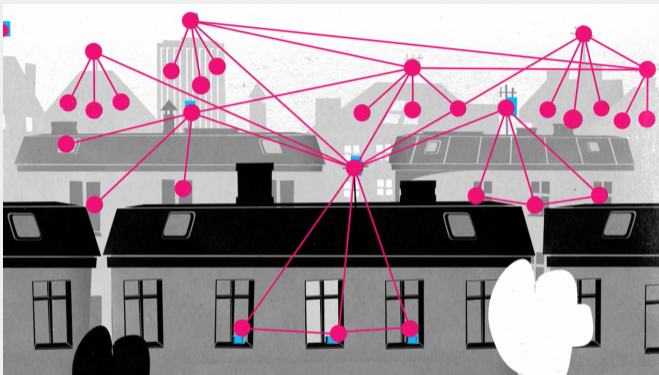
Mesh



© CC BY SA, Philipp Seefeldt

■ Freifunk: Mesh + MANET (Metropolitan Area Network)

Mesh



© CC BY SA, Philipp Seefeldt

- Freifunk: Mesh + MANET (Metropolitan Area Network)
- > 35'000 mit batman-adv
(pro Segment: bis zu 700)

- klassisch: routing auf OSI Layer 3

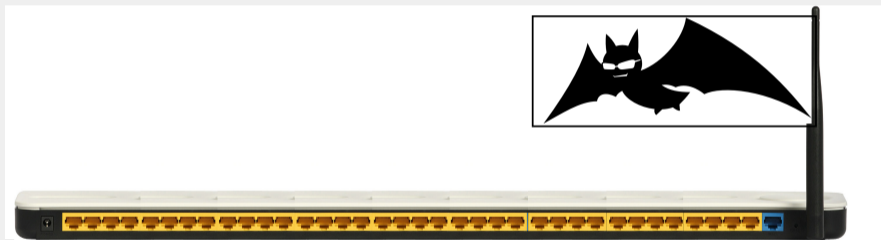
Layer 2

- klassisch: routing auf OSI Layer 3
- Layer 2: Forwarding/switching

Layer 2



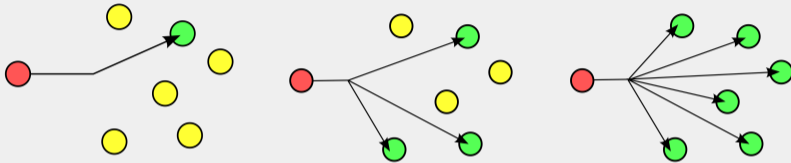
Layer 2



Vorteile:

- Netzwerkprotokoll agnostisch: IPv4, IPv6, IPX, NetBEUI, ...

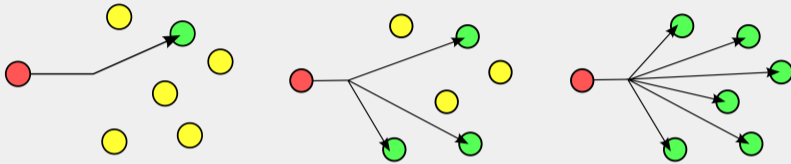
Layer 2



Vorteile:

- Netzwerkprotokoll agnostisch: IPv4, IPv6, IPX, NetBEUI, ...
- neben Unicast:
Broadcast & Multicast (zwangsläufig)

Layer 2



Vorteile:

- Netzwerkprotokoll agnostisch: IPv4, IPv6, IPX, NetBEUI, ...
- neben Unicast:
Broadcast & Multicast (zwangsläufig)
- Roaming / Handover von Client Geräten

GESCHICHTE

Entstehung B.A.T.M.A.N.

- Erfahrungen bei Wizards of OS III in 2004 und Freifunk Berlin
⇒ Routing Loops mit OLSR

Entstehung B.A.T.M.A.N.

- Erfahrungen bei Wizards of OS III in 2004 und Freifunk Berlin
⇒ Routing Loops mit OLSR
- "The OLSR Story" von Corinna 'Elektra' Aichele

Entstehung B.A.T.M.A.N.

- Erfahrungen bei Wizards of OS III in 2004 und Freifunk Berlin
⇒ Routing Loops mit OLSR
- "The OLSR Story" von Corinna 'Elektra' Aichele
- Konzept: Elektra + Thomas Lopatic

Entstehung B.A.T.M.A.N.

- Erfahrungen bei Wizards of OS III in 2004 und Freifunk Berlin
⇒ Routing Loops mit OLSR
- "The OLSR Story" von Corinna 'Elektra' Aichele
- Konzept: Elektra + Thomas Lopatic
- Erster Prototyp: Thomas Lopatic

Entstehung B.A.T.M.A.N.

- Erfahrungen bei Wizards of OS III in 2004 und Freifunk Berlin
⇒ Routing Loops mit OLSR
- "The OLSR Story" von Corinna 'Elektra' Aichele
- Konzept: Elektra + Thomas Lopatic
- Erster Prototyp: Thomas Lopatic
- Dann: batmand (layer 3)

Entstehung B.A.T.M.A.N.

batmand vo.1, Dez. 2006, commits:

- Marek Lindner (88)
- Stefan Sperling (20)
- Mirko Vogt (9)
- Andreas Langer (7)
- Corinna 'Elektra' Aichele (5)
- Axel Neumann (3)

Entstehung B.A.T.M.A.N. Advanced

- Chemnitzer Linuxtage, März 2007:
"AWDS - drahtlose Multihop-Netze leicht gemacht" (André Herms)
- ⇒ batman-adv-userspace
- Entwickler: Simon Wunderlich + Marek Lindner

batman-adv: kernelspace

- batman-adv-userspace: langsam

- batman-adv-userspace: langsam
- Linux kernel Implementierung

- batman-adv-userspace: langsam
- Linux kernel Implementierung
- Dez. 2007, 24c3 Vortrag:
"Wireless Kernel Tweaking - or how B.A.T.M.A.N. learned to fly "

batman-adv v0.1, Dez. 2007, commits:

- Simon Wunderlich (79)
- Marek Lindner (66)
- Sven Eckelmann (3)
- Andreas Langer (1)
- Scott Raynel (1)

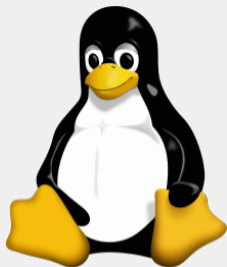
- 2009: Firmware Entwicklung in der MetaMeute

- 2009: Firmware Entwicklung in der MetaMeute
- auf OpenWrt Basis

- 2009: Firmware Entwicklung in der MetaMeute
- auf OpenWrt Basis
- Start von Freifunk Lübeck

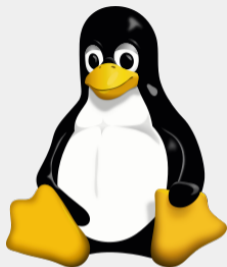
- 2009: Firmware Entwicklung in der MetaMeute
- auf OpenWrt Basis
- Start von Freifunk Lübeck
- 2013: Freifunk Firmware Framework "Gluon"

batman-adv: upstreaming



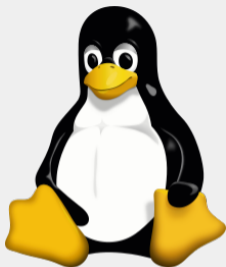
- Aufnahme Linux staging: 11. Dez. 2009

batman-adv: upstreaming



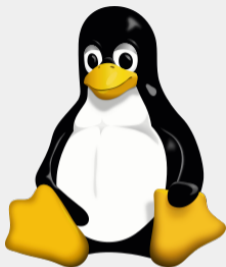
- Aufnahme Linux staging: 11. Dez. 2009
- (mein erster batman-adv Patch, yaiy! :-))

batman-adv: upstreaming



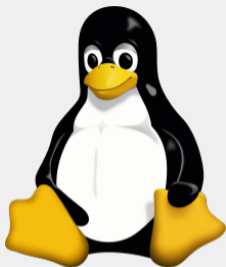
- Aufnahme Linux staging: 11. Dez. 2009
- (mein erster batman-adv Patch, yay! :-))
- Erster Linux release mit batman-adv (staging):
Linux v2.6.33 / 24. Feb. 2010

batman-adv: upstreaming



- Aufnahme Linux staging: 11. Dez. 2009
- (mein erster batman-adv Patch, yay! :-))
- Erster Linux release mit batman-adv (staging):
Linux v2.6.33 / 24. Feb. 2010
- Aufnahme Linux net: 16. Dez. 2010

batman-adv: upstreaming



- Aufnahme Linux staging: 11. Dez. 2009
- (mein erster batman-adv Patch, yay! :-))
- Erster Linux release mit batman-adv (staging):
Linux v2.6.33 / 24. Feb. 2010
- Aufnahme Linux net: 16. Dez. 2010
- Erster Linux release mit batman-adv (offiziell):
Linux v2.6.38 / 14. März 2011

batman-adv v2024.3

batman-adv v2024.3 / Linux v6.12, commits:

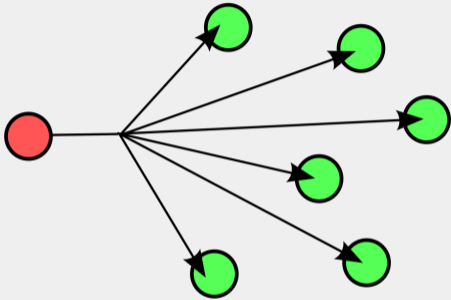
- Sven Eckelmann (1101)
- Marek Lindner (741)
- Simon Wunderlich (481)
- Antonio Quartulli (404)
- Linus Lüßing (192)
- Andrew Lunn (46)
- Martin Hundebøll (37)
- Matthias Schiffer (24)
- Markus Pargmann (21)
- Andreas Langer (15)
- Joe Perches (11)
- Johannes Berg (9)
- Markus Elfring (9)
- Eric Dumazet (8)
- Jakub Kicinski (8)
- Al Viro (7)
- Jiri Pirko (7)
- Andreas Pape (6)
- Dan Carpenter (6)
- Jesper Juhl (5)
- + 100 weitere

TECHNIK

TECHNIK

ROUTING BASICS

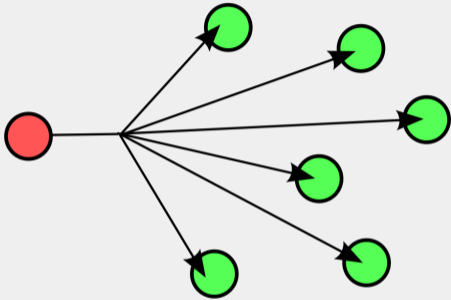
Broadcast



Broadcast Flooding:

■ rebroadcasts

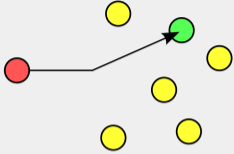
Broadcast



Broadcast Flooding:

- rebroadcasts
- Sequenznummern

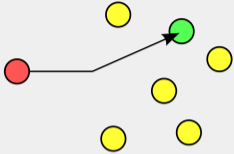
Unicast



B.A.T.M.A.N. IV / V

■ proaktiv

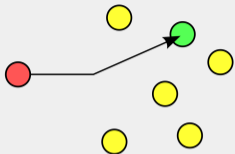
Unicast



B.A.T.M.A.N. IV / V

- proaktiv
- Distance Vector

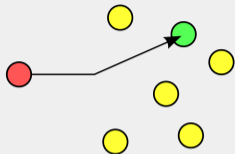
Unicast



B.A.T.M.A.N. IV / V

- proaktiv
- Distance Vector
 - ▶ kennt nur besten Nachbarn zum Ziel

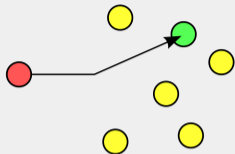
Unicast



B.A.T.M.A.N. IV / V

- proaktiv
- Distance Vector
 - ▶ kennt nur besten Nachbarn zum Ziel
 - ▶ vs. link-state (OLSR): kennt komplette Topologie

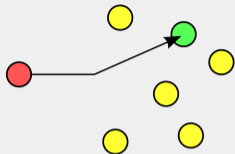
Unicast



B.A.T.M.A.N. IV / V

- proaktiv
- Distance Vector
 - ▶ kennt nur besten Nachbarn zum Ziel
 - ▶ vs. link-state (OLSR): kennt komplette Topologie
- loop-frei

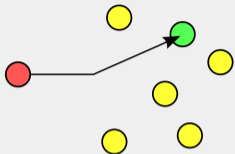
Unicast



B.A.T.M.A.N. IV / V

- proaktiv
- Distance Vector
 - ▶ kennt nur besten Nachbarn zum Ziel
 - ▶ vs. link-state (OLSR): kennt komplette Topologie
- loop-frei
- Metrik:

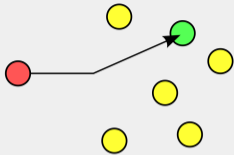
Unicast



B.A.T.M.A.N. IV / V

- proaktiv
- Distance Vector
 - ▶ kennt nur besten Nachbarn zum Ziel
 - ▶ vs. link-state (OLSR): kennt komplette Topologie
- loop-frei
- Metrik:
 - ▶ B.A.T.M.A.N. IV: Paketverlust (hauptsächlich)

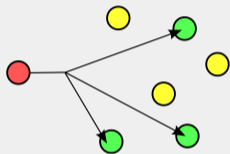
Unicast



B.A.T.M.A.N. IV / V

- proaktiv
- Distance Vector
 - ▶ kennt nur besten Nachbarn zum Ziel
 - ▶ vs. link-state (OLSR): kennt komplette Topologie
- loop-frei
- Metrik:
 - ▶ B.A.T.M.A.N. IV: Packetverlust (hauptsächlich)
 - ▶ B.A.T.M.A.N. V: Durchsatz (hauptsächlich)

Multicast

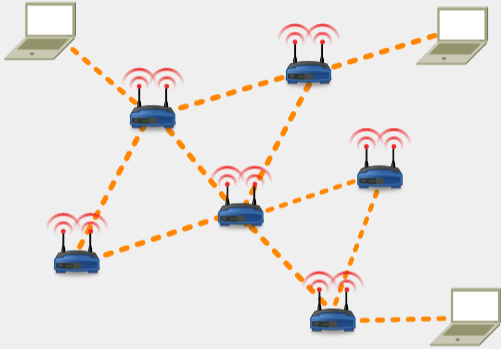


- einfache Lösung: wie Broadcast
 - ▶ Problem: skalierbarkeit / Overhead
- aber: batman-adv Optimierungen

TECHNIK

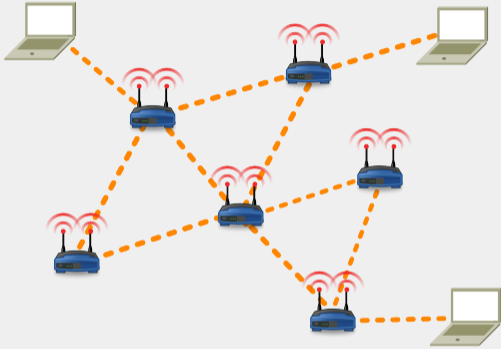
LAYER 2 FEATURES

Translation Table



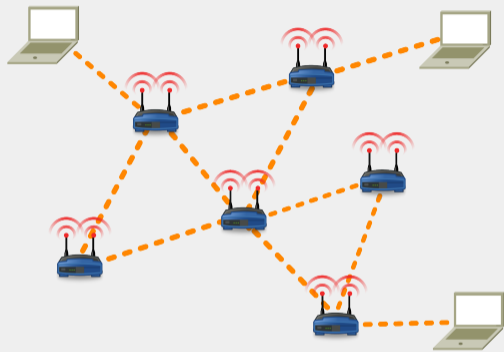
- Einbindung von nicht B.A.T.M.A.N. Geräten

Translation Table



- Einbindung von nicht B.A.T.M.A.N. Geräten
- keine Metrik (Bandbreite, Packetloss, Hopcount etc.)

Translation Table



- Einbindung von nicht B.A.T.M.A.N. Geräten
- keine Metrik (Bandbreite, Packetloss, Hopcount etc.)
- reaktiv statt proaktiv

Bridge Loop Avoidance

- verhindert Paketkreisen über (statt auf) batman-adv
- wie STP (Spanning Tree Protocol) bei Switches
- aber: berücksichtigt die Metrik

- zerteilt Datenpakete, die viel zu lang sind

- zerteilt Datenpakete, die viel zu lang sind
- in zwei handlichere Teile

Fragmentierung

- zerteilt Datenpakete, die viel zu lang sind
- in zwei handlichere Teile
- IP Fragmentierung: n/a

- DHCP Requests nur zum "besten" Gateway Knoten

- DHCP Requests nur zum "besten" Gateway Knoten
 - ▶ Optimierung der Route/Ziel

- DHCP Requests nur zum "besten" Gateway Knoten
 - ▶ Optimierung der Route/Ziel
 - ▶ Reduzierung von DHCP overhead \Rightarrow Unicast

Distributed ARP Table

- Reduzierung von ARP Overhead

Distributed ARP Table

- Reduzierung von ARP Overhead
- benutzt eine DHT (Distributed Hash Table)

Distributed ARP Table

- Reduzierung von ARP Overhead
- benutzt eine DHT (Distributed Hash Table)
 - ▶ ARP per Unicast, statt Broadcast


- IGMP/MLD snooping

- IGMP/MLD snooping
- batman-adv Multicast Paket Typ


- IGMP/MLD snooping
- batman-adv Multicast Paket Typ
- Näheres im Vortrag vom Wireless Battlemesh v16:
"Messages for the Masses: State of Multicast in batman-adv and Gluon"

Danke! Fragen?

- Website: <https://www.open-mesh.org/>
- Mailingliste: b.a.t.m.a.n@lists.open-mesh.org
- IRC (+Matrix) Chat: #batadv auf hackint

WLAN-Router Icon (Heimrouter kochbuch Szenarium w Wifi extender.svg):  CC BY SA, Wikipedia / Ademant

Paket Icon (Emojione 1F4E6.svg):  CC BY SA, Wikipedia / emojione

License:  – CC-BY-SA-4.0, unless noted otherwise