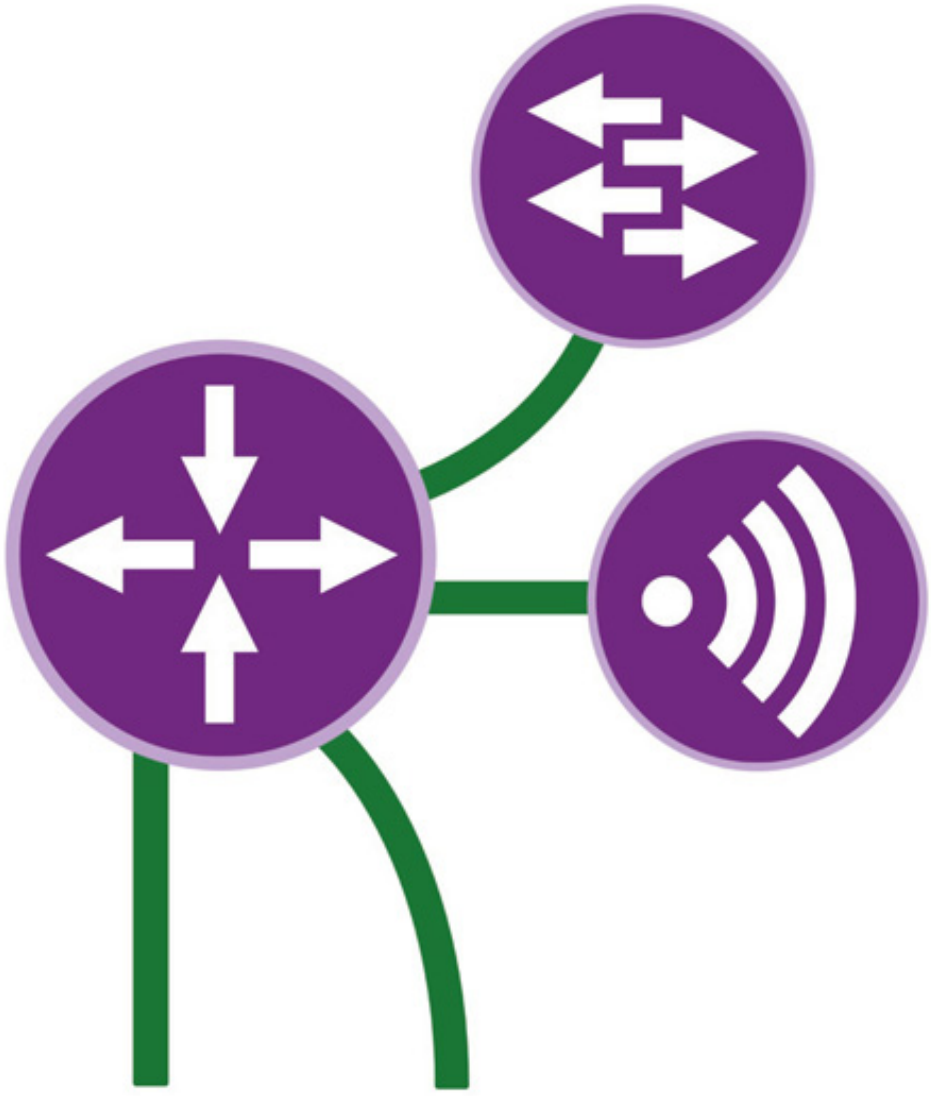

Der OpenWrt-Praktiker

Band 3: Anwendungsfälle

Markus Stubbig

Inhaltsverzeichnis

Einleitung

OpenWrt ist eine Linux-Distribution für Netzwerkgeräte, wie Router, Accesspoints und Switches. Für die Konfiguration bietet OpenWrt eine Weboberfläche und eine Kommandozeile. Der Fokus liegt auf WiFi, Firewall und Routing. OpenWrt läuft auf einer Vielzahl an Hardware-Architekturen oder als virtuelle Maschine.

Jeder Hersteller stattet seine Netzkomponenten mit einem eigenen Betriebssystem aus. OpenWrt verkauft selber keine Hardware, sondern portiert seine Linux-Distribution auf möglichst viele Geräte. OpenWrt ersetzt auf *anderen* Routern das Betriebssystem und macht die Hardware funktionsreicher und teilweise leistungsstärker.

Der erste Band der Buchreihe *Der OpenWrt-Praktiker* vermittelt einen Einstieg in OpenWrt und deckt die Grundlagen ab. Die Kapitel sind eine Bedienungsanleitung, die OpenWrt installieren, Netzschnittstellen einrichten und IP-Adressen vergeben. Nach der ersten Einrichtung behandelt Band 1 auch die Kommandozeile UCI, die Paketverwaltung und die Systemadministration mit Überwachung eines OpenWrt-Geräts.

Der erste Band richtet sich an Leser, die mit OpenWrt keine Erfahrung haben und ins Thema einsteigen wollen. Wer bereits einen OpenWrt-Router im Einsatz hat, kann mit dem zweiten oder dritten Band starten.

Der zweite Band zeigt, welche Möglichkeiten OpenWrt bietet, wie die Software intern arbeitet und welche Dienste aus der Cloud eine mögliche Ergänzung sind. Die Themen sind für Anwender mit Vorkenntnissen konzipiert. Sie vermitteln dem Leser fortgeschrittene Inhalte, Tipps für die Fehlersuche und ein großes Kapitel zur Firewall mit Adressumsetzung.

Übersicht

Band 3 zeigt sechs Anwendungsfälle aus der Praxis. In Kapitel 1 bauen mehrere WiFi-Geräte ein drahtloses Mesh-Netz auf und erweitern damit die Ausleuchtung bestehender Funknetze. Wenn viele OpenWrt-Router im Netz zusammenarbeiten sollen, zeigt Kapitel 2, wie sich die Geräte kennenlernen und dynamisch ihre Routen austauschen. Kapitel 3 widmet sich der Verfügbarkeit und demonstriert, wie zwei Router gemeinsam ein Cluster formen und im Fehlerfall den Betrieb aufrechterhalten können.

In großen Netzen mit vielen Routern sind Konfigurationsänderungen mühsam. Kapitel 4 präsentiert eine kostenfreie Software zur Automatisierung und Verwaltung von OpenWrt-Geräten. In Kapitel 5 wird OpenWrt zum Werbeblocker, der für alle Endgeräte im Netz die Werbung aus Webseiten ausfiltert. Zuletzt bedient OpenWrt in Kapitel 6 mehrere Internet-Zugänge und balanciert darüber die Datenströme seiner Clients.

Labornetz

Die Anwendungsfälle in den folgenden Kapiteln benutzen ein beispielhaftes Netzwerk, welches aus vier OpenWrt-Routern besteht. Das Labornetz ist als Netzdiagramm in Abbildung 1 dargestellt und ist identisch mit dem Diagramm in den ersten beiden Bänden. Es stellt ein kleines Netzwerk mit mehreren Standorten dar. In den Kapiteln werden meist nur Teile dieses Netzwerks zur Untersuchung benutzt. Tabelle 1 auf Seite → enthält die IP-Adressen der Router und mit welchen Netzsegmenten sie verbunden sind.

Version

Die Entwicklung von OpenWrt bleibt nicht stehen. Nicht immer kann die Dokumentation mithalten; aus diesem Grund verwenden die Bände der Buchreihe *Der OpenWrt-Praktiker* nicht die gleiche Version, sondern arbeiten stets mit den aktuellen Versionen von OpenWrt. Als Folge können die Screenshots und Kommandoausgaben zwischen den Bänden und den eigenen Experimenten unterschiedlich ausfallen.

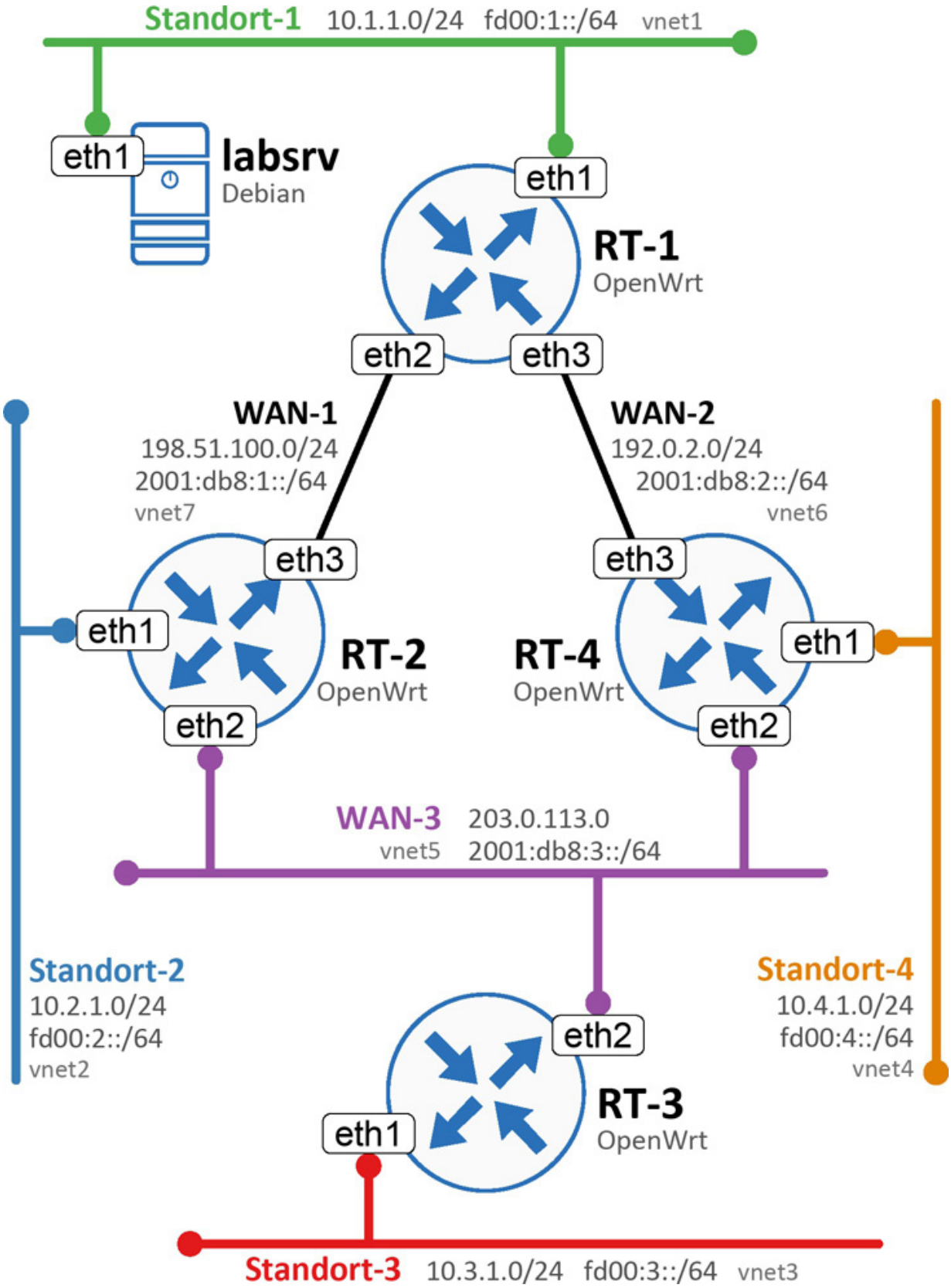

Abbildung 1: Das Labornetzwerk als Vorlage für die folgenden Kapitel

Gerät	Interface	Funktion/Netz	IPv4	IPv6
RT-1	eth0 eth1 eth2 eth3	Management Standort-1 WAN-1 WAN-2	10.5.1.1 10.1.1.1 198.51.100.1 192.0.2.1	fd00:5::1 fd00:1::1 2001:db8:1::1 2001:db8:2::1
RT-2	eth0 eth1 eth2 eth3	Management Standort-2 WAN-3 WAN-1	10.5.1.2 10.2.1.2 203.0.113.2 198.51.100.2	fd00:5::2 fd00:2::2 2001:db8:3::2 2001:db8:1::2
RT-3	eth0 eth1 eth2	Management Standort-3 WAN-3	10.5.1.3 10.3.1.3 203.0.113.3	fd00:5::3 fd00:3::3 2001:db8:3::3
RT-4	eth0 eth1 eth2 eth3	Management Standort-4 WAN-3 WAN-2	10.5.1.4 10.4.1.4 203.0.113.4 192.0.2.4	fd00:5::4 fd00:4::4 2001:db8:3::4 2001:db8:2::4
labsrv	eth0 eth1	Management Standort-1	10.5.1.7 10.1.1.7	fd00:5::7 fd00:1::7

Tabelle 1: Alle Geräte mit Netzadaptern, Funktion und IP-Adressen

Kapitel 1

Mesh-WLAN

In einem WiFi-Mesh-Netzwerk besteht das Kernnetz aus Funkverbindungen. Drahtlose WiFi-Router verbinden sich dynamisch miteinander und leiten die Pakete ihrer Teilnehmer zielgerichtet vom Sender zum Empfänger.

Der klare Vorteil zur „klassischen“ Vernetzung liegt darin, dass das Mesh-Netz ohne LAN-Kabel auskommt. Damit kann das Mesh-Netz auch Lokationen erreichen, die per Kabel nur schwer zugänglich sind oder ein verlegtes Kabel mit hohen Kosten verbunden ist.

Seit 2012 regelt der Standard IEEE 802.11s das Miteinander in Mesh-Netzen und bringt damit die Geräte verschiedener Hersteller unter einen Hut. Zusätzlich dazu definiert der Standard ein Routingprotokoll, welches die Pfadentscheidungen der Mesh-Teilnehmer festlegt.

Dieses Kapitel baut ein kleines Mesh-Netz mit vier Teilnehmern auf. Nach der ersten Einrichtung vernetzen sich die WiFi-Router und erreichen über verschiedene Routingprotokolle ihre Endgeräte.

Grundlagen

Die drahtlosen Kernkomponenten eines Mesh-Netzes sind *Mesh-Points* (MP). Sie leiten Datenpakete von anderen Mesh-Points weiter und bilden so die Infrastruktur des Netzes. Den Zugang zum Mesh-Netz bildet ein Mesh-Accesspoint (MAP), der zusätzlich die Funktion des Accesspoints hat und damit für reguläre WiFi-Clients sichtbar ist. Die Clients verbinden sich mit dem Accesspoint und betreten damit (unbewusst) das Mesh-Netz. Wenn ein Mesh-Point Zugang zu einem *anderen* Netz hat, wird er

zum *Mesh Portal Point* (MPP) und agiert als Gateway zwischen den beiden Netzen. Abbildung 1.1 zeigt die verschiedenen Rollen und das geplante Mesh-Netz.

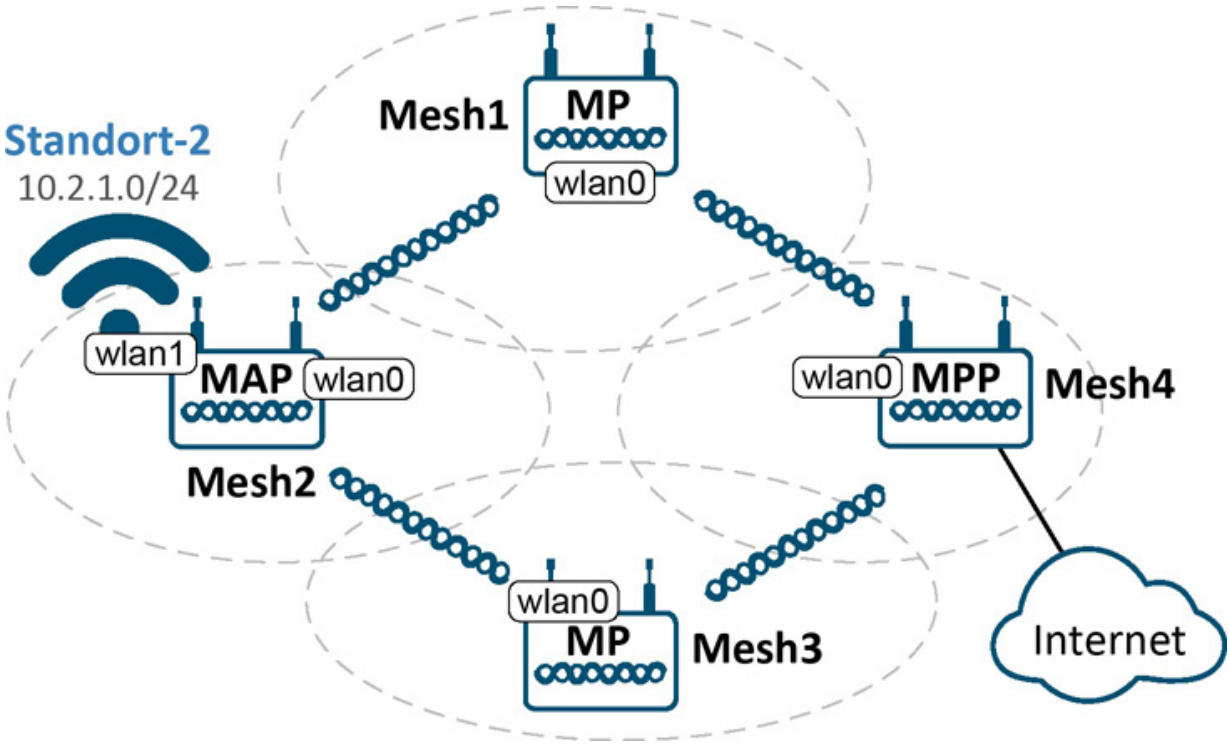

Abbildung 1.1: Die OpenWrt-Geräte bilden ein Mesh-Netz

Ein Mesh-Point kann ein OpenWrt-Router mit WiFi-Schnittstelle sein, aber auch ein Laptop mit mesh-fähigem WiFi-Adapter und der passenden Software.

Labor

Das verwendete Labor-Mesh-Netz aus Abbildung 1.1 verwendet vier Mesh-Points, von denen einer gleichzeitig als Mesh-Portal-Point den Zugang zum Internet darstellt und ein weiterer als Accesspoint arbeitet. Der Mesh-Accesspoint in Abbildung 1.1 hat zwei WiFi-Adapter und kann damit unterbrechungsfrei das Mesh und die WiFi-Clients aus Standort-2 bedienen. Der Mesh-Portal-Point hat einen Uplink ins Internet und stellt den Übergabepunkt dar.

Die Mesh-Router sind so platziert, dass sich alle Geräte per Funk direkt erreichen können, mit Ausnahme von Mesh2 und Mesh4. Die grauen Ovale um die einzelnen Geräte zeigen die gewollte Überlappung. Damit entstehen zwei mögliche Pfade von einem drahtlosen Client in Standort-2 zum Internet: via Mesh2, Mesh1 zu Mesh4 oder via Mesh2, Mesh3 zu Mesh4. Es wird Aufgabe des Routingprotokolls sein, den besten Pfad zu finden und zu verwenden.

Voraussetzung

Das 802.11s-fähige Mesh-Netz benötigt kompatible Mesh-Points. Das gilt für die verwendete WiFi-Hardware und für die Einstellungen von Mesh-ID und Kanal.

- *WiFi-Adapter*. Die verbaute WiFi-Schnittstelle und der verwendete Treiber müssen die Fähigkeit zum „meshen“ haben. Unter OpenWrt zeigt das iw-Kommando, ob der Netzadapter bereit für das Mesh ist:

```
root@mesh2:~# iw list
Wiphy phy0
[...]
       Supported interface modes:
                * IBSS
                * managed
                * AP
                * AP/VLAN
                * monitor
                * mesh point
                * P2P-client
                * P2P-GO
                * outside context of a BSS

[...]
```

- *Mesh-ID*. Die Mesh-ID ist die Kennung des Mesh-Netzwerks. Alle beteiligten Mesh-Points müssen dieselbe Mesh-ID verwenden. Damit hat die Mesh-ID die gleiche Funktion wie die SSID eines WiFi-Accesspoints.

- *Kanal*. Alle Mesh-Points müssen denselben Funkkanal belegen.

- *Routingprotokoll*. Obwohl der 802.11s-Standard ein Routingprotokoll mitbringt, gibt es mehrere gute Alternativen. Der Betreiber des Mesh-Netzes legt fest, welches Protokoll die Mesh-Points verwenden.

- *Verschlüsselung*. Die Verschlüsselung in Mesh-Netzen ist optional, muss aber einheitlich erfolgen. Es gelten dieselben Regeln wie bei der Mesh-ID und der Kanalwahl: Alle MPs verwenden dieselben Einstellungen.

Routingprotokoll

Das Routingprotokoll im Mesh-Netz hat die gleichen Aufgaben wie im kabelgebundenen Netz: Die Mesh-Points lernen sich gegenseitig kennen und erfahren von ihren verbundenen Clients. Damit kann jeder Mesh-Point die transportierten Pakete zielgerichtet weiterleiten und muss sie nicht ins Netz fluten.

In drahtlosen Netzen steht das Routingprotokoll vor weiteren Herausforderungen, dass sich das Netzwerk dynamisch verändert, die Verbindungen häufig asymmetrisch sind und ein unzuverlässiges Transportmedium benutzen. Für die Pfadentscheidung in Funknetzen kann das Routingprotokoll zusätzlich die Verbindungsqualität bewerten und damit Nachbarn mit guter Signalstärke bevorzugen.

Routingprotokolle arbeiten proaktiv, reaktiv oder beherrschen beide Formen. Bei der proaktiven Arbeitsweise baut der Router eine vollständige Routingtabelle und hat damit alle Informationen, bevor er sie tatsächlich braucht. Im reaktiven Modus informiert sich der Router nur bei Bedarf über den benötigten Pfad.

Beide Ansätze haben ihre Vorteile: Im proaktiven Modus müssen die IP-Pakete nicht „warten“, bis die Mesh-Points den Weg erfragt haben. Im reaktiven Modus spart der Mesh-Point Bandbreite und Leistung.

Tabelle 1.1 auf der nächsten Seite vergleicht Routingprotokolle, die im Verlauf dieses Kapitels näher untersucht werden. Zusätzlich dazu unterstützt OpenWrt die Protokolle *Babel*, *cjdns* und *OSPF* (vgl. Kap. 2).

Eingenschaft	**HWMP**	**OLSR**	**B.A.T.M.A.N.**	**Batman-adv**
OSI-Ebene	2	3	3	2
proaktiv	☑	☑	☑	☑
reaktiv	☑	☐	☐	☐
UCI	☐	☑	☑	☑
LuCI	☐	☑	☐	☐
Metrik	Linkqualität	Hop count	Paketverlust	IV: Linkqualität V: Datenrate
Standard	802.11s	RFC 3626	Draft-RFC	—

Tabelle 1.1: Vergleich der bekannten Routingprotokolle für Mesh-Netze

HWMP

Das *Hybrid Wireless Mesh Protocol* (HWMP) ist das „eingebaute“ Routingprotokoll. Ohne weitere Konfiguration verwenden die Mesh-Points HWMP und finden damit den richtigen Pfad durch das Netz. Der Vorteil vom HWMP liegt darin, dass jeder 802.11s-kompatible Mesh-Router das Protokoll versteht, da es fester Bestandteil des IEEE-Standards ist.

Unter OpenWrt läuft HWMP direkt im Linux-Kernel und benötigt kein zusätzliches Softwarepaket. Feintuning am Protokoll erlaubt OpenWrt über das iw-Kommando.

HWMP eignet sich für kleine Mesh-Netze, oder wenn die eingesetzten Geräte keinen Platz für ein zusätzliches Routingprotokoll haben.

OLSR

Wenn HWMP im eigenen Mesh nicht ausreicht oder ungünstige Pfadentscheidungen trifft, kann das *Optimized Link State Routing*-Protokoll (OLSR) aushelfen. OLSR ist spezialisiert auf Funknetze und arbeitet proaktiv. In diesem Modus kennt jeder Mesh-Point das gesamte Netz und benötigt daher entsprechend viel Arbeitsspeicher für die Routingtabelle und CPU-Leistung für deren Berechnung.

OLSR ist eins der ersten Routingprotokolle für Mesh-Netze. Seit Version 1 aus dem Jahr 2003 hat das Protokoll an Stabilität gewonnen und verwendet in der neueren Version die Verbindungsqualität für Pfadentscheidungen. OLSR ist nicht auf Mesh-Netze beschränkt und funktioniert auch in kabelgebundenen Ethernetsegmenten.

OpenWrt bietet Softwarepakete für beide OLSR-Versionen. Die Implementierung olsrd lässt sich per LuCI und UCI konfigurieren, ist vielseitig via Plug-ins erweiterbar und orientiert sich an OLSR-Version 1.

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

Das Projekt *Better Approach To Mobile Adhoc Networking* (BATMAN) entstand aus den Nachteilen von OLSR und dem Wunsch nach einem effizienteren Routingprotokoll für Mesh-Netze.

Genau wie OLSR kommunizieren bei BATMAN die Mesh-Points miteinander und lernen sich kennen. Die erste Version von BATMAN arbeitet proaktiv auf Ebene 3 des OSI-Modells und lässt die Mesh-Points per IP-Paketen kommunizieren. Mit diesem Ansatz ergibt sich noch nicht die gewünschte Skalierbarkeit gegenüber OLSR.

Daraus entstand *BATMAN Advanced*, welches auf OSI-Ebene 2 arbeitet und die Datenpakete aller Teilnehmer durch das Mesh-Netz tunnelt. Durch die Verkapselung müssen die BATMAN-Router nicht mehr alle Endgeräte kennen, sondern nur noch die anderen BATMAN-Router. Als Folge schont BATMAN-Advanced

den Prozessor und den Arbeitsspeicher, da der einzelne Mesh-Point nicht das gesamte Netz kennen muss. Auf OSI-Ebene 2 kann Batman-adv das Mesh-Netz den höheren Schichten als großen Ethernetswitch präsentieren. Damit ermöglicht Batman-adv ein Roaming der WiFi-Clients zwischen den Mesh-Accesspoints.

Abbildung 1.2 auf der nächsten Seite zeigt, wie die BATMAN-Router ein Datenpaket durch das Mesh-Netz tunneln. Sobald ein Paket das Mesh-Netz verlässt, entfernt der BATMAN-Router die zusätzlichen Kopfzeilen und sendet das Paket in seiner ursprünglichen Form weiter.

Bei der Pfadentscheidung scheinen die Entwickler mehrere Varianten auszuprobieren. BATMAN III (Spalte *B.A.T.M.A.N.* in Tabelle 1.1) entscheidet sich für einen benachbarten Mesh-Point, der die geringsten Paketverluste aufweist. Der daraus entstehende Pfad durch das Mesh kann asymmetrisch sein, da die Verlustrate zwischen zwei Mesh-Points in Sende- und Empfangsrich tung unterschiedlich ist. In BATMAN IV (Spalte *Batman-adv* in Tabelle 1.1) informieren sich die Mesh-Points gegenseitig über ihre Sendequalität und treffen die Routingentscheidung auf Basis der Sende- und Empfangsqualität. Die Implementierung in OpenWrt verwendet BATMAN IV.

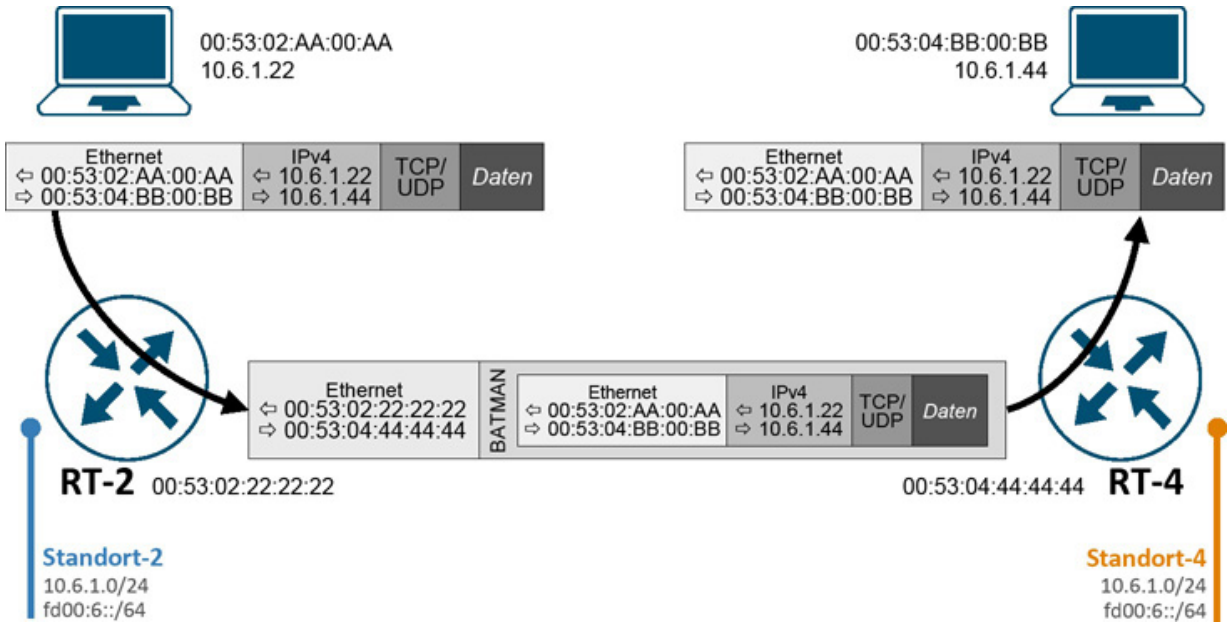

Abbildung 1.2: Im Mesh-Netz erhalten die Pakete den zusätzlichen BATMAN-Header

In BATMAN V wählen die Mesh-Points ihren Pfad ausschließlich auf Basis des Datendurchsatzes. Damit versprechen sich die Entwickler eine größere Skalierbarkeit, da die Mesh-Points weniger Messpakete durch die Luft senden müssen.

Bei der Implementierung haben sich die Entwickler für ein Linux-Kernel-Modul entschieden. Die Routingentscheidung und Paketweiterleitung erfolgt somit direkt im Linux-Kernel, was die Verarbeitungsgeschwindigkeit positiv beeinflusst.

Einrichtung

Zuerst benötigen die OpenWrt-Router die passende Software für die Verschlüsselung zwischen den Mesh-Points. Wenn das Mesh unverschlüsselt kommunizieren soll, kann dieser Schritt übersprungen werden. Da die Einrichtung einer soliden Verschlüsselung in OpenWrt denkbar einfach ist, ist dieser Schritt empfehlenswert:

```
opkg remove wpad-basic
opkg install wpad-mesh-openssl
```

Die Einstellungen für das drahtlose Netz befinden sich in der Weboberfläche von OpenWrt unter *Netzwerk → WLAN*. Der Button *Hinzufügen* rechts neben dem WLAN-Adapter öffnet das Dialogfenster für ein neues WiFi-Netz. Die Mesh-Points erhalten die beispielhaften Einstellungen aus Tabelle 1.2. Die Werte im Bereich *Erweiterte Einstellungen* bleiben in der Grundeinstellung, welche das Routingprotokoll HWMP von Seite → verwenden.

Einstellung	**Wert**
Kanal	6
Maximale Sendeleistung	Treiber-Standardwert
Ländercode	DE - (Germany)
Modus	802.11s
Mesh-ID	wrt.mesh
Netzwerk	wlan0 *(erzeugen)*
Verschlüsselung	WPA3-SAE (hohe Sicherheit)
Schlüssel	OpenWrt-Praktiker

Tabelle 1.2: Grundeinstellungen für die Mesh-Points

Anschließend kommt das neue Interface *wlan0* unter *Netzwerk → Schnittstellen* an die Reihe. Für die Kommunikation der Mesh-Points