BGP-Routing mit Mikrotik-Routern

Thomas Osterried, DL9SAU
Egbert Zimmermann, DD9QP

RouterOS 6.x oder RouterOS 7.x ?

Im HAMNET werden zumeist Router von Mikrotik eingesetzt. Sehr verbreitet und nach wie vor empfohlen ist die RouterOS Version 6.x. Diese Reihe wird von Mikrotik aktuell weiter gepflegt und mit Updates versehen (Stand Juni 2023). Seit einiger Zeit ist auch die RouterOS-Version 7.x verfügbar. Mit ROS v7.x hat sich die BGP Konfiguration gegenüber ROS v6.x grundlegend geändert, leider ist sie deutlich komplexer geworden. Wer RouterOS-Version 7.x einsetzen muss, weil

- die Hardware des neuen Gerätes die alte, stabile Version 6 nicht mehr unterstützt
- das in ROS 7.x neu hinzu gekommene VPN Protokoll Wireguard genutzt werden soll
- es bgp-sessions effizienter verwalten kann (Multicore Unterstützung) und dadurch laut Mikrotik schneller arbeitet

muss sich zwingend mit der neuen Syntax der Version 7 auseinandersetzen. In allen anderen Fällen (insbesondere bei Link-Devices) bringt im HAMNET ein Wechsel von ROS v6 auf ROS v7 aktuell keinerlei Vorteile oder Mehrwert.

Wer im HAMNET einfach ein automatisches Upgrade von ROS6 auf ROS7 macht, wird ohne vorherige Maßnahmen das Announcement vom Site-Network verlieren. Der Router ist dann nur noch über seine Transfernetz-IP-Adressen oder per RoMon oder direkt vom Linknachbarn aus erreichbar.

Mit diesem Dokument möchten wir zeigen, was mit Blick auf unsere HAMNET Policies zu beachten ist.

RouterOS 6.x

Angenommen wir haben lokal zwei Linkpartner (in den Transfernetzen 44.148.78.x/29):

```
44.148.78.0/29 auf dem interface eth1-link-db0aaa
```

und

```
44.148.78.8/29 auf dem interface eth2-link-db0bbb
```

Das uns zugewiesene Site-Netz beinhaltet alle User/Services Netze und sei

```
44.149.36.128/27
```

Es teilt sich auf in das Service-Netz für die lokalen Server und Devices

```
44.149.36.128/28 auf der bridge br-lan
```

und das User-Netz für die HAMNET-Nutzereinstiege

```
  44.149.36.144/28 auf der bridge br-users
```

HAMNET-POLICY: Wir announcen die uns zugewiesenen User/Services-Netze als einen Block (Site-Netz), also als 44.149.36.128/27

(und nicht separat die Subnetze 44.149.36.128/28 und 44.149.36.144/28, und auch nicht alle drei!).

Hintergrund: *Jeder Router im HAMNET lernt diese Netze. Bricht irgendwo auf dem Weg ein Link, muss für alle Netze, die über diesen Linkpartner announced wurden, vom Router neu errechnet werden, was jetzt der bessere Weg ist, und dieser wird in die Routing-Table eingetragen. Diese Berechnungen kosten Zeit und die Routen belegen RAM.* ***Die Routingtabellen sollten also so klein wie möglich gehalten werden.*** *Auch beim Routing ist eine kleinere Routingtable schneller durchsucht als eine große .*

Wir wollen nicht den mutmaßlichen Performance-Gewinn von ROS v7 wieder verlieren. Auch wollen wir nicht ältere ROS v6 Router (oftmals mit 1-Core-Prozessoren und wenig RAM) an ihre Grenzen bringen. Deshalb achten wir darauf, dass wir die zugewiesenen Blöcke so groß und so zusammenhängend wie möglich announcen.

Wir senden also unsere User/Services Netze als einen großen Block (/27, egal ob wir ihn selbst in kleinere Subnetze unterteilt haben). Lediglich die Transfernetze (in denen unser Router, die Linkeinheiten, remote Router und remote Linkeinheit liegen), aggregieren wir NICHT. Wir legen sie normalerweise in ein /29 Netz.

Diese Vorgehensweise ist nicht neu! Wir haben das schon von Anfang an im HAMNET so gemacht, also auch zu Zeiten von ROS v5 oder ROS v6.

Unsere Konfiguration im alten ROS v6 unterscheidet sich beim User/Services-Netz von den Transfernetzen:

```
/routing bgp network
add network=44.148.78.0/29 synchronize=yes
add network=44.148.78.8/29 synchronize=yes
add network=44.149.36.128/27 synchronize=no
```

Site-Netze mit synchronize=no

Das „synchronize=no“ ist bei ROS v6 wichtig, weil ROS nur Netze announced, die in der Routing-Table aktiv eingetragen sind und exakt matchen. Im obigen Beispiel haben wir aber das Netz geteilt und als 44.149.36.128/28 und 44.149.36.144/28 auf zwei getrennte services- und user-Bridge-Interfaces gelegt.

Deshalb haben wir in der BGP-Konfiguration für das /27 Netz synchronize=no angegeben.

Randbemerkung: Best Practice der ISPs im Internet ist, das Netz auf das loopback interface lo zu legen. Das ist immer „up“. Unter ROS gibt es aber kein loopback interface. Man kann sich zwar eines bauen, z. B. als bridge mit

```
/interface bridge add name=br-lo
```

und konfiguriert kein teilnehmendes Interface. Man kann aber auch eine sogenannte „blackhole“-Route setzen:

```
/ip route add dst-address=44.149.36.128/27 type=blackhole
```

Diese Konfiguration hat den Vorteil, dass man auch für alle Subnetze zuständig ist, auch wenn man beispielsweise das 44.149.36.144/28 noch gar nicht konfiguriert hat, weil der User-Zugang noch nicht aufgebaut worden ist. In Routingentscheidungen gewinnt übrigens immer das „kleinere“ Netz, also das /28 („more specific“). Dass wir das /27-Netz über loopback oder blackhole „geerdet“ haben, hat keinen negativen Einfluss auf unsere darin enthaltenen /28-Subnetze.

Transfernetze mit synchronize=yes

Gehen wir noch kurz auf den Vorteil von synchronize=yes ein. Unsere Transfernetze announcen wir so. Beispielsweise sei unsere Linkeinheit direkt am ethernet-Port eth1-link-db0aaa angeschlossen. Jetzt gehe das Netzteil kaputt. Weil das Interface jetzt nicht mehr „up“ ist, wird automatisch das betreffende /29 Transfernetz nicht mehr per bgp announced.

Vorteil: Unser Linkpartner announced dieses Netz ja auch. Wenn wir wissen möchten, was passiert ist, wollen wir zum Abschätzen der Ursache des Ausfalls auf dessen Funkeinheit nachsehen (log events, oder zu schlechte SNR, …). Announced aber unser defekter Standort weiterhin dieses Netz, und hat dies einen kürzeren Pfad als das Announcement vom Nachbarn db0aaa, dann erreichen wir dessen Funkeinheit nicht. Also ist es praktisch, dass unser Announcement automatisch unterbleibt, wenn das Link-Interface down ist. Bis zur Reparatur könnte man das /29-Announcement in ein /30-Announcement ändern (natürlich ist dann synchronize=no in der BGP Einstellung erforderlich). Das ist der bessere „Match“, und so kommt man während der Reperatur, egal über welchen Weg, wieder an die eigene Funkeinheit. Achtung! Nach der Reparatur nicht vergessen das /30-Announcement wieder zu entfernen!

Tipp: *Füllen wir das /29 bei Seite A von unten und Seite B von oben (also bb-A=.1, trx-A=.2, trx-B=.5, bb-B=.6), dann funktioniert oben beschriebener /30-Announcement-Trick. Broadcast Adresse in A ist 3, Netzwerk-Adresse in B ist 4. Läge trx-B auf der .3, dann läge er noch im /30-Announcement von A!*

Nach diesen Vorüberlegungen betrachten wir nun ROS v7:

RouterOS 7.x

ROS v7 kennt kein "synchronize=no" mehr!

Wir können also nur announcen, für was wir zuständig sind. Zuständig sind wir, wenn wir eine Route in unserer Routingtable haben. Das können sein:

- Interfaces. Ein Interface hat automatisch eine Route wenn es „up“ ist, also das Interface auf enabled steht und der Ethernet-Port belegt ist
- Statische Routen (dazu gehört auch eine blackhole Route)
- über OSPF- oder anderweitig gelernte Routen

Da wir weiterhin unser /27 als einen Block announcen möchten, dürfen wir bei RouterOS 7.x das Anlegen einer blackhole Route auf keinen Fall vergessen!

Weitere Unterschiede

„/routing bgp peer“ heißt in ROS v7 „/routing/bgp/connection“. Und „/routing bgp instance“ ist in „/routing/bgp/template“ umbenannt. Das „template“ konfiguriert nun die eigene ASN, und andere „defaults“. Die konkreten /routing/bgp/connection Einträge der Peers beziehen sich auf das template (templates=xxx), so dass die Konfiguration ererbt wird, aber noch (wenn nötig) für diese spezielle Verbindung mit anderen Werten überschrieben werden kann.

bgp network in Firewall verschoben

Mit ROS v7 wurde das Menu „/routing bgp network“ in die Firewall-Konfiguration als Adress-Liste verschoben! Auf diese wird dann in der bgp/connection-Konfiguration (ehemals „routing bgp peer“) auf die „/ip/firewall/adress-list“-Einträge verwiesen.

Gut zu wissen: Vergisst man in der bgp connections Konfiguration für einen Peer die Referenz auf die zu announcenden Netze in der address-list anzugeben, lernt der BGP-Partner zwar die Netze die wir selbst über bgp gelernt haben, nicht aber unsere eigenen!

Dies unterscheidet sich grundlegend von der Konfiguration aus ROS v6!

In ROS v6 wirkten sich „/routing bgp networks“ grundsätzlich erst einmal auf alle Peers aus und ließen sich, wenn nötig, über „/routing filter“ ausfiltern.

In ROS v7 wurde „/routing/filter“ deutlich verschlankt. Sie sind nun als input.xxx, output.xxx Filter nach /routing/template/<templatename> bzw. nach /routing/connections/<peername> gewandert.

ROS v6 und ROS v7 Konfiguration

Stellen wir nun die alte Konfiguration der neuen gegenüber:

ROS v6

```
/routing bgp network
add network=44.148.78.0/29 synchronize=yes
add network=44.148.78.8/29 synchronize=yes
add network=44.149.36.128/27 synchronize=no

/routing bgp instance
```

```
set default as=4226267900 router-id=44.149.36.129

/routing bgp peer
add address-families=ip,ipv6 name=DB0AAA nexthop-choice=force-self \
    remote-address=44.148.78.1 remote-as=4226267901 ttl=1
add address-families=ip,ipv6 name=DB0BBB nexthop-choice=force-self \
    remote-address=44.148.78.9 remote-as=4226267902 ttl=1

Optional:
/ip route
add dst-address=44.149.36.128/27 type=blackhole
```

Eine solche, in ROS v6 optionale, blackhole-Route für das gesamte Site-Netz sollte man vor einem automatischen Upgrade von ROS v6 auf ROS v7 unbedingt eintragen. Dadurch verringert sich die Chance, dass man nach erfolgtem Upgrade und Neustart die Remote-Erreichbarkeit des Devices verliert.

ROS v7

```
/ip/firewall/address-list
add address=44.148.78.0/29 comment=DB0AAA list=bgp-advertise
add address=44.148.78.8/29 comment=DB0BBB list=bgp-advertise
add address=44.149.36.128/27 comment=userservices list=bgp-advertise

/routing/bgp/template
set default as=4226267900 router-id=44.149.36.129 \
    output.network=bgp-advertise nexthop-choice=force-self

/routing/bgp/connection
add address-families=ip,ipv6 as=4226267900 name=DB0AAA \
    nexthop-choice=force-self output.network=bgp-advertise \
    remote.address=44.148.78.1 .as=4226267901 .ttl=1 templates=default
add address-families=ip,ipv6 as=4226267900 name=DB0BBB \
    nexthop-choice=force-self output.network=bgp-advertise \
    remote.address=44.148.78.9 .as=4226267902 .ttl=1 templates=default

/ip/route
add dst-address=44.149.36.128/27 blackhole
```

Die Blackhole-Route wird bei ROS v7 zwingend nötig damit das eigene /27 Site-Netz announced wird.

Hinweis: *Beachte dass „type=“ nicht mehr vor „blackhole“ steht.*

Die Address-List führt den Listennamen „bgp-advertise“ ein, auf den in bgp/template und bgp/connection Bezug genommen werden kann (output.network=).

Zu /routing/bgp/connection:
as, output.network, nexthop-choice sollten eigentlich vom template geerbt werden.. ..aber sicher ist sicher.

Zur seltsam anmutenden Syntax „.as“, .ttl„ und andere mit .irgendwas:
Diese bezieht sich auf eine vorherige Zuweisung. remote.address=44.148.78.1 .as=4226267901 ist also gleichwertig zu remote.address=44.148.78.1 remote-address.as=4226267901

Wer das allerdings sortieren will zu as=4226267900 .as=422626790 name=DB0AAA fällt auf die Nase weil .as nun als as.as interpretiert würde. Das gibt es genauso wenig wie beispielsweise input.network.

URL dieser Seite: https://de.ampr.org/hamnet/bgp-mikrotik