Kapitel 8

Ausfallschutz

Switches und ihre Verbindungen untereinander fallen manchmal aus. Und dann erfüllen sie ihre fundamentalste Aufgabe nicht mehr, die darin besteht, Pakete zu transportieren.
Und Switches fallen genauso gerne aus wie andere elektronische Bauteile. Das ist eine akzeptierte Tatsache und aus diesem Grund haben High-End–Geräte mehrere Netzteile, Lüfter, CPUs oder Uplinks. Zusätzlich hilft man sich meist damit, dass mehrere Switches als Gruppe (Cluster) auftreten. Dann entsteht ein Cluster für Hochverfügbarkeit und Ausfallschutz. Sehr beliebt ist auch die mehrfache Verkabelung zwischen zwei Geräten, um den Defekt einer einzelnen Verbindung abzufangen.
Cumulus Linux bevorzugt bei der Hochverfügbarkeit eine Aktiv/Aktiv-Konstellation. Dabei beteiligen sich alle Kabelverbindungen am Datentransport und verbessern damit die Verfügbarkeit und erhöhen gleichzeitig die Gesamtbandbreite.

Link Aggregation

Wenn zwei Switches über mehrere Kabel miteinander verbunden sind, wird das *Spanning-Tree Protokoll* (STP, vgl. Kap. 13) aufmerksam und sperrt alle bis auf eine Verbindung. Das ist kein böswilliges Verhalten von STP, sondern die Strategie zum Vermeiden von Schleifen im Netz. Und sobald die einzige genutzte Verbindung ausfällt, wird STP eine der anderen Netzadapter entsperren und die Daten können wieder fließen.

Das Prinzip ist ganz brauchbar, aber *alle* redundanten Leitungen sind inaktiv. Das geht besser, wenn auch bei STP nur mit Tricks. Die vorteilhaftere Methode ist die Bündelung von mehreren physikalischen Leitungen zu einer logischen Portgruppe, wobei jede Leitung aktiv ist. Neben dem Ausfallschutz steht auch noch zusätzliche Bandbreite zur Verfügung. Für STP gibt es nur noch die eine logische Verbindung und keinen Grund diese zu blockieren.
Die verschiedenen Hersteller waren bei der Namensgebung kreativ und die Bezeichnung der Kanalbündelung reicht vom standardisierten *Link Aggregation* über *Bonding* im Linux-Umfeld, *EtherChannel* bei Cisco, *Port Trunk* bei HPE und *Teaming* bei Microsoft Windows.

Grundlagen

Sobald mehrere Leitungen zwischen zwei Geräten als gemeinsamer Kanal arbeiten, hat der Sender die Aufgabe, die ausgehenden Pakete auf die verschiedenen Leitungen zu verteilen. Die parallele Nutzung kann eine höhere Bandbreite erreichen; im Maximum die Summe aller einzelnen Leitungen.
Die beiden Endpunkte eines Kanals müssen nicht unbedingt Switches sein. Üblich ist auch die mehrfache Anbindung eines Servers oder Routers an einen Switch.
Für die Bündelung gibt es den allgemein anerkannten Standard *Link Aggregation Control Protocol* (LACP nach IEEE 802.3ad) und häufig noch herstellerspezifische Erweiterungen. Cumulus Linux setzt auf LACP ohne weitere Zusätze.
Beide LACP-Partner verhandeln über ihre physikalischen Ports und bilden daraus den logischen Kanal. Im laufenden Betrieb tauschen die Partner kontinuierlich LACP-Pakete aus, um defekte Leitungen zu erkennen oder Änderungen zu propagieren.

Die Voraussetzungen für eine Kanalbündelung sind:

- Alle Leitungen müssen dieselbe Bandbreite haben
- Alle Leitungen müssen im Vollduplexmodus arbeiten
- Die Leitungen verbinden exakt zwei Geräte

Interessanterweise gehört die Kenntnis von LACP nicht zu der Liste, denn ein Cumulus-Switch bündelt auch Verbindungen zu unwissenden Partnern. Der Trick liegt darin, dass beide Geräte die konfigurierten Netzadapter bedingungslos zum Bündel hinzufügen und darauf vertrauen, dass die Gegenstelle dasselbe macht.

Die Anzahl der Ports im Bündel folgt keiner festen Regel. Der Algorithmus zum Verteilen der Last streut die ausgehenden Pakete brav über alle konfigurierten Interfaces, auch wenn die Anzahl ungerade ist.
Was ist mit der alten Daumenregel, dass die Anzahl der Ports immer auf der Basis von Zwei sein muss, um die Last optimal verteilen zu können? Diese Weisheit galt für Switches mit älteren Netzwerkprozessoren, die bei Bündeln aus 3, 5 oder 7 Netzadaptern relativ schief verteilt haben. Cumulus Linux hält sich beim Austeilen strikt an den Algorithmus, ohne einen bestimmten Ausgang zu bevorzugen. Auf moderner Hardware sind alle Ports im Bündel gleichmäßig beteiligt.

Laboraufbau

Die Kanalbündelung kann zwischen beliebigen Teilnehmern stattfinden, solange mindestens zwei Kabel von derselben Quelle zum selben Ziel verlaufen. Für den praktischen Anfang bilden die Switches sw01 und sw02 über ihre jeweiligen Netzadapter *swp5* und *swp6* eine gebündelte Leitung (Abbildung 8.1).

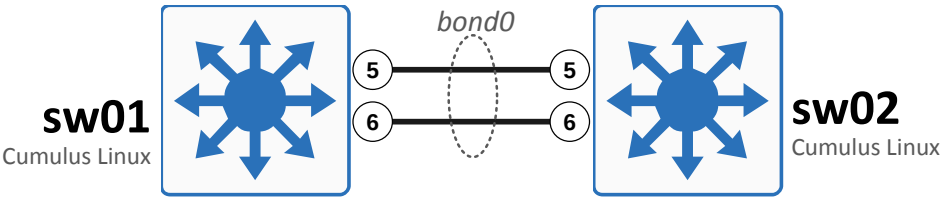

Abbildung 8.1: Die beiden Switches formen einen gemeinsamen Kanal per LACP

Das fertige Bündel erhält einen eigenen Netzadapter mit einem passenden Namen. Die Vorgabe von Linux ist *bond0*, wobei der Name auch den Zweck beschreiben kann, z. B. *bond_sw01_sw02*.

Die Planung ist das Aufwendigste, denn die Konfiguration ist ein Einzeiler. Sie erwartet auf beiden Enden den Namen des Multi-Link–Adapters und seine Teilnehmer.

```
net add bond bond0 bond slaves swp5,6
net add interface swp5,6
```

Die zweite Zeile stellt lediglich sicher, dass die Interfaces auch angeschaltet sind. Das wars – und die NCLU bestätigt nach einem `net commit` stolz den Verbund:

```
cumulus@sw01:~$ net show interface bonds
    Name    Speed    MTU  Mode  Summary
--  ------  -------  ----  ----  --------------------------------
UP  bond0   2G      1500  LACP  Bond Members: swp5(UP), swp6(UP)
```

Ausfallschutz

Die Switches sw01 und sw02 sind nun mehrpfadig verbunden und gewappnet, falls eine einzelne Leitung versagt. Dabei muss es sich nicht um einen physikalischen Defekt handeln. Geplante Umverkabelung im Serverschrank ohne Wartungsfenster ist ebenfalls eine mögliche Ursache für einen unvollständigen Leitungsverbund.
In beiden Fällen bemerkt LACP den Wegfall eines Netzadapters und schickt die Datenpakete über eine andere Leitung. Das LACP-Bündel bleibt im Status *UP*, aber nicht alle Teilnehmer sind bereit für die Arbeit. Die nutzbare Bandbreite reduziert sich um die Bandbreite des havarierten Netzadapters.

```
cumulus@sw01:~$ net show interface bonds
    Name    Speed    MTU  Mode  Summary
--  ------  -------  ----  ----  --------------------------------
UP  bond0   1G      1500  LACP  Bond Members: swp5(DN), swp6(UP)
```

Cumulus Linux protokolliert den Ausfall mit einer knappen Meldung im eigenen Logbuch.

```
Jun 01 20:29:32 sw01 kernel: bond0: link status definitely down \
  for interface swp5, disabling it
```

Im Fehlerfall wünscht sich das Monitoring-Team bestimmt eine Alarmierung und so kann Cumulus Linux seinen Besitzer per Syslog und/oder SNMP-Trap benachrichtigen (vgl. Kap. 5).

Lastverteilung

In der Voreinstellung hält sich Cumulus Linux brav an den vorgegebenen Algorithmus von LACP. Dieser berücksichtigt Quell- und Ziel-IP-Adresse und – falls vorhanden – die TCP/UDP-Portnummer. Diese Informationen wandern in eine XOR-Operation und das Ergebnis ist die Nummer des ausgehenden Netzadapters. Folglich wird eine einzelne TCP/IP-Verbindung immer über denselben Adapter versendet, denn während einer Verbindung ändert sich weder die Portnummer noch die IP-Adresse.

Interoperabilität

Sobald die ersten Switches mit Cumulus Linux im eigenen Datacenter an die Tür klopfen, beginnen die Prüfungen zur Verträglichkeit mit der Hausmarke. In der Theorie ist das kein Problem, denn LACP stellt eine gemeinsame Sprache für alle Hersteller dar. Die Praxis bringt kleinere Hürden, die einer der beiden Partner angleichen muss.
Die LACP-Implementierung von Cumulus Linux nutzt den Linux-Kernel und hat damit seine Flexibilität. Cumulus Networks hat sein Betriebssystem gegen die Switches seiner Kollegen positiv getestet. Die häufigsten Probleme während der Einrichtung eines LACP-Bündels waren:

- Die Rate der LACP-Statuspakete ist unterschiedlich. Es gibt zwei Raten: Schnell (jede Sekunde) oder langsam (alle 30 Sekunden) und beide Partner müssen dieselbe Rate nutzen. Empfohlen ist die schnelle Rate.

- Die Konfiguration und Liste der VLANs sind unterschiedlich. Die transportierten VLANs auf beiden Enden des Bündels müssen identisch sein. Das gilt auch für das *Native VLAN*.

Bei LACP gibt es noch den passiven und aktiven Modus, der bestimmt, ob die Aushandlung selbstständig begonnen werden darf, oder nur auf Rückfrage der Gegenstelle. Falls beide Partner passiv bleiben, beginnt keine Verhandlung und die Kanalbündelung bleibt aus. Cumulus Linux verzichtet auf den passiven Modus, sodass beide Teilnehmer immer aktiv werden und keine Fehlerquelle darstellen.

Wenn sich beide Partner gar nicht einigen wollen, bietet Cumulus Linux eine Alternative: Mit der Option *balance-xor* handelt der Switch nicht mehr nach Standard, sondern deaktiviert LACP und aktiviert alle Netzadapter im Bündel. Die Entscheidung für eine bedingungslose Lastverteilung trifft das Kommando:

```
net add bond bond0 bond mode balance-xor
```

Technischer Hintergrund

Die IEEE-Norm 802.3ad *Link aggregation* definiert, wie sich zwei Switches verhalten, um Datenpakete über mehrere aktive Leitungen auszutauschen. Der Standard ist seit dem Jahr 2000 verfügbar und alle namhaften Hersteller und Betriebssysteme haben gute Unterstützung dafür. 2008 strukturiert die IEEE um und führt den Standard unter der Bezeichnung 802.1AX fort. Für die Implementierung von LACP setzt Cumulus Networks auf den vorhandenen `bonding`-Treiber des Linux-Kernels. Damit übernimmt Cumulus Linux alle Fähigkeiten eines Linux-Bonds ohne in zusätzliche Programmierarbeit zu investieren. Der Ansatz ist legitim, da der Treiber unter der Lizenz GPL steht und die Weiterverwendung gestattet.
Die Einrichtung eines Bonds übernimmt die NCLU, aber das ist dem Linux-Treiber herzlich egal. Falls die Aussage der NCLU über Status und Statistik mal zu knapp ist, liefert `/proc/net/bonding/bond0` die volle Palette an Informationen über den Bond (hier `bond0`).

Multi-Chassis Link Aggregation

Der Verbund von mehreren Netzadaptern zu einem starken Multi-Gigabit–Bündel ist klasse, hilft aber nicht, wenn der gesamte Switch die Arbeit einstellt. Alternativ könnten ein paar Leitungen des Bündels zu einem weiteren Switch führen, um den Ausfall eines Chassis abzufangen, aber das macht LACP nicht mit.
Der Weg führt zur *Multi-Chassis Link Aggregation* (MLAG), die genau diesen Ansatz erlaubt. Zwei Switches stellen ein *Multi-Chassis*–System dar, welches eine gemeinsame Kanalbündelung zum Partner ermöglicht. Für den Partner sieht das Multi-Chassis-Gerät wie ein einzelner Switch aus, der sogar LACP spricht.

Der Trick bei MLAG ist, dass sich beide Teile des Chassis nach außen als *ein* Switch verkleiden und sich an den LACP-Standard halten. Nach innen gibt es zwischen den Geräten intensive Kommunikation, um den Schein nach außen zu wahren.
Abbildung 8.3 auf Seite 103 zeigt das Pärchen aus den Switches sw01 und sw02, welche eine Multi-Chassis-Gruppe bilden. Für den Partner sw11 ist es eine normale Kanalbündelung aus zwei Leitungen zu *einer* Gegenstelle.

Grundlagen

Leider hat sich MLAG nie als Standard etabliert, sodass jeder Hersteller seine eigene Implementierung zusammenstrickt. Auch die Namensgebung variiert: *Multi-Chassis Trunking* (Brocade), *Virtual PortChannel* (Cisco Nexus) oder *Distributed trunking* (HPE). Cumulus Linux benennt seine Implementierung als *Multi-Chassis Link Aggregation* (MLAG), wobei intern auch oft die Abkürzung CLAG auftaucht. Auf der Kommandozeile haben die Befehle mehrheitlich `clag` im Namen.
Die Verbindung *zwischen* den Teilnehmern des MLAG-Switches ist extrem wichtig, damit die Gegenstelle der Kanalbündelung nichts von der Täuschung bemerkt. Die Verbindung nennt Cumulus Linux *Peer-Link* und ihre Kommunikation besteht aus:

- Gemeinsame Verwaltung. Beide Switches müssen wissen, welche ihrer Netzadapter zum Bündel gehören und wie ihr Status ist.
- Synchronisierung von Protokollen. Nach außen müssen ebenfalls Infrastrukturprotokolle, wie Spanning-Tree und IGMP, überzeugt werden.
- Datenverkehr. Falls Pakete am „falschen“ Switch der MLAG-Gruppe ankommen, durchqueren diese den Peer-Link und tauchen am anderen Switch beim „richtigen“ Port wieder auf. Diese Umlenkung nutzen hauptsächlich Gegenstellen, die einpfadig angeschlossen sind.

Selbst in kleinen Setups haben Switches mehrere MLAG-Gruppen, und somit verwendet Cumulus Linux als Unterscheidungskriterium die `clag-id`, welche eine Zahl zwischen 1 und 65.535 annimmt. Dazu gehört eine gemeinsame MAC-Adresse `clagd-sys-mac`, die der Switch verwendet, wenn

er sich als virtueller MLAG-Switch ausgibt. Die Kommunikation zwischen den MLAG-Chassis verläuft über IP, also benötigen beide Enden des Peer-Links eine IP-Adresse in einem gemeinsamen Subnetz. Die Pakete betreten nicht das reguläre Datennetz, also genügen private Adressen oder sogar der Bereich der Link-Local–Adressen 169.254.0.0/16.

Die Voraussetzungen für MLAG sind:

- Ein MLAG-Pärchen besteht aus exakt zwei Switches, auf denen Cumulus Linux Version 2.5 oder moderner schnurrt.
- Es muss eine direkte Verbindung zwischen den MLAG-Chassis bestehen. Und am besten mehrere Kabel für eine robuste Kanalbündelung.
- Jeder chassis-übergreifende Bond benötigt eine eigene CLAG-Nummer, die auf beiden Switches gleich ist.

Laboraufbau

Die beiden Switches sw01 und sw02 verschmelzen zu einem „virtuellen“ Switch – zumindest aus den Augen des Gegenübers sw11. Der Aufbau in Abbildung 8.3 erweitert das Netzdiagramm aus Abschnitt *Link Aggregation* auf Seite 97, denn der vorhandene LACP-Channel wird als Peer-Link benötigt.
Zum unwissenden Partner sw11 führt jeweils nur ein Kabel, wodurch das Minimum eines Multi-Chassis-Channels entsteht. Mehrfache Anbindung ist denkbar und in der Praxis sogar sinnvoll, um Geräte- und Leitungsredundanz zu erreichen.
Die Sicht von sw11 auf das MLAG-Pärchen zeigt scheinbar einen einzelnen Switch und ist in Abbildung 8.2 dargestellt.

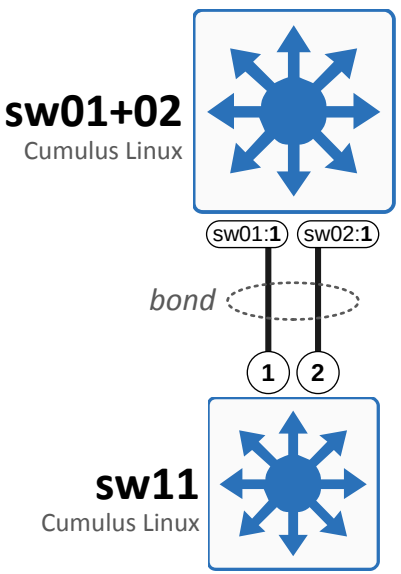

Abbildung 8.2: Das MLAG-Chassis aus der Sicht von Switch sw11

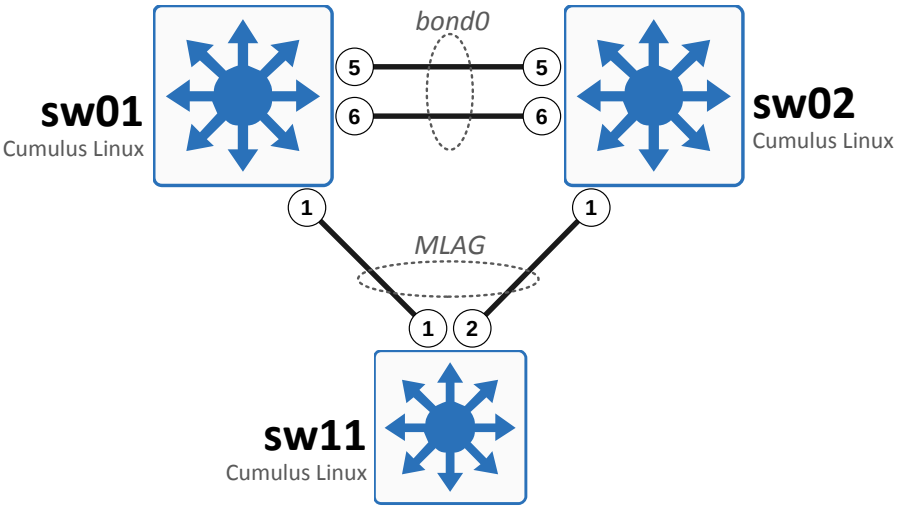

Abbildung 8.3: Zwei Switches stellen unsichtbare Redundanz auf OSI-Ebene 2 dar

Einrichtung

Sobald die drei Switches verkabelt sind, versucht die Konfiguration daraus scheinbar zwei Switches zu machen. Es beginnt mit dem Peer-Link zwischen sw01 und sw02. Dieser ist das Herz von MLAG und sollte stets redundant als Bündel ausgeführt sein.
Der Peer-Link erhält eine freie MAC-Adresse aus dem reservierten Bereich 44:38:39:ff:00:00 bis 44:38:39:ff:ff:ff.

```
1 net add bond isl bond slaves swp5,swp6
2 net add interface isl.4000 clag enable yes
3 net add interface isl.4000 ip address 192.0.2.1/24
4 net add interface isl.4000 clag peer-ip 192.0.2.2
5 net add interface isl.4000 clag sys-mac 44:38:39:00:00:11
6 net add bridge bridge ports isl
```

Der Peer-Link bekommt den beispielhaften Namen *isl* (Zeile 1) und VLAN 4000 transportiert die Inter-Chassis-Befehle. Die `sys-mac` (Zeile 5) ist auf beiden Chassis identisch. Die IP-Adressen zur Kommunikation (Zeilen 3 und 4) sind auf den Switches invers zueinander. Da VLAN-Tagging im Spiel ist, benötigt der Peer-Link in Zeile 6 den entsprechenden Hinweis auf eine Netzbrücke.

Damit ist der Peer-Link einsatzbereit, was die Ausgabe mit dem passenden `show`-Kommando bestätigt:

```
cumulus@sw01:~$ net show clag
The peer is alive
   Our Priority, ID, and Role: 32768 08:00:27:61:03:56 primary
  Peer Priority, ID, and Role: 32768 08:00:27:ec:cb:f0 secondary
        Peer Interface and IP: isl.4000 192.0.2.2
                    Backup IP:  (inactive)
                  System MAC: 44:38:39:00:00:11
```

Ab jetzt verhalten sich beide Switches sw01 und sw02 wie ein gemeinsamer Switch, der aus zwei Modulen besteht. Weiter geht es mit der Kanalbündelung, die ab sofort aus Netzadaptern beider Switches bestehen kann. Dem Laboraufbau folgend wird Switchport *swp1* auf beiden MLAG-Switches Teil der Kanalbündelung. Zur normalen Einrichtung eines LACP-Channels kommt hier der MLAG-Identifier hinzu, der als `clag-id` in der Konfiguration auftaucht. Die ID muss innerhalb des Verbundes einmalig sein und auf beiden Switches des LACP-Bündels die gleiche Zahl verwenden. Durch die Wahl des Netzadapters sind die Kommandos auf beiden Switches identisch. Der Name des Bündels *sw11* ist nur ein *Hinweis* darauf, welches Gerät die Gegenstelle ist.

```
net add bond sw11 bond slaves swp1
net add bond sw11 clag id 1
```

Am anderen Ende des Kanals sieht die Konfigurationslage deutlich einfacher aus, denn für sw11 ist es eine handelsübliche Verbindung aus mehreren Kabeln.

```
net add bond sw0102 bond slaves swp1,swp2
```

Technischer Hintergrund

Die *Multi-Chassis Link Aggregation* ist eine Softwareimplementierung, die Cumulus Networks für seine Switches entwickelt hat. Alle Bestandteile schnürt der Hersteller in das Paket *clag*, welches als Version 1.3.0 dem Betriebssystem beiliegt.
Im Hintergrund läuft der Dienst `clagd`, welcher den Peer-Link hält und die konfigurierten Netzadapter bedient. Seine Aufträge erhält `clagd` vom

Frontend `clagctl` mittels eines Sockets. `clagctl` wiederum bekommt seine Anweisungen von der NCLU und damit vom Administrator.
Alle Programmteile sind in Python verfasst und damit im Quellcode einsehbar. Leider wird damit *clag* nicht automatisch zu Open Source, denn die Software steht unter einer proprietären Lizenz. An der Dokumentation hat Cumulus Networks nicht gespart, denn die Programmteile sind sowohl im Quellcode als auch in der Man-Page ausgezeichnet beschrieben.
Für die Inter-Chassis-Kommunikation erfindet Cumulus Networks kein neues Protokoll, sondern sendet die Befehle im XML-Format per TCP-Port 5342 durch den Peer-Link.
Im Fehlerfall hilfreich: `clagd` protokolliert Änderungen an der Konfiguration oder am Zustand der Netzadapter in der Logdatei `/var/log/clagd.log`. Und bei ausgewachsenen Problemen hat `clagd` noch die Optionen `--debug` und `--verbose` im Gepäck, um den Fehlersuchenden mit ausreichend Material zu versorgen.

Doppel-MLAG

Der Gedanke an Redundanz führt zu noch verrückteren Netzdiagrammen, denn beide Endpunkte des LACP-Kanals können ein MLAG-Pärchen sein. Also vier Switches, die sich als zwei tarnen. In dieser Königsklasse der Redundanz entsteht ein Szenario, welches bei korrekter Verkabelung den Ausfall einer beliebigen Komponente verkraftet.

Laboraufbau

In diesem letzten Labordiagramm spielen die Switches sw01 und sw02 die erste MLAG-Gruppe, welche „über Kreuz“ mit dem MLAG-Pärchen sw11 und sw12 verbunden ist. Alle Verbindungen in Abbildung 8.4 auf der nächsten Seite sind redundant ausgelegt, um den optimalen Ausfallschutz zu erreichen.
Die Endgeräte außerhalb des redundanten Netzaufbaus werden dargestellt durch sw13 und server2. Sie nehmen keine besonderen Aufgaben wahr, außer dass sie per Kanalbündelung an den MLAG-Switch angeschlossen sind und über eine IP-Adresse für die Erfolgskontrolle verfügen.

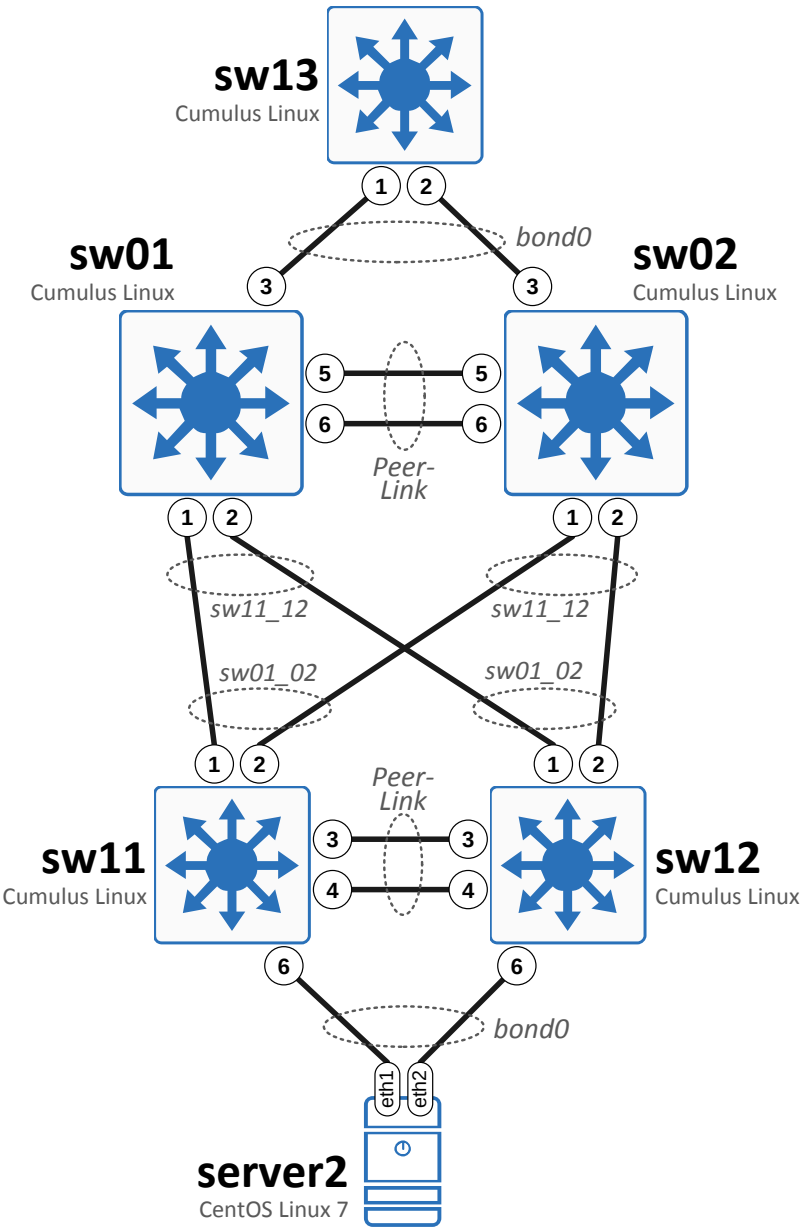

Abbildung 8.4: Din Enden des gebündelten Kanals bestehen aus mehreren Chassis

Um die Situation nicht unnötig zu verkomplizieren, verbleiben alle Netzadapter und Kanäle im VLAN 1.

Hinweis

Im virtuellen Umfeld akzeptiert *Cumulus VX* praktisch jeden Adaptertyp, aber im Bereich von MLAG entstehen mit *virtio-net* ungewollte Effekte. Für den Ablauf dieses Szenarios ist der Adaptertyp *e1000* stabiler.

Einrichtung

Die Konfiguration des „doppelten“ MLAG-Teams benutzt die bekannten Befehle. Zum besseren Verständnis unterteilt sich die folgende Einrichtung in nachvollziehbare Schritte, die den gesamten Aufbau teilen und beherrschen.

MLAG zwischen sw13 und sw01–sw02

Die Konfigurationsreise beginnt bei sw13, der als Server fungiert und mehrpfadig an das MLAG-Team aus sw01 und sw02 angeschlossen ist. Dazu erhält sw13 ein simples LACP-Bündel aus zwei Leitungen und eine IP-Adresse obendrauf, die im letzten Schritt die Ende-zu-Ende–Verbindung bestätigt.

```
net add bond bond0 bond slaves swp1,swp2
net add bond bond0 ip address 10.1.1.13/24
```

Zuerst benötigt die MLAG-Gruppe einen Peer-Link zwischen den Teilnehmern. Die Konfiguration entspricht fast unverändert dem Abschnitt *Einrichtung* auf Seite 103, da sie das vorherige Setup um die zweite MLAG-Gruppe erweitern soll.

```
1 net add bond isl bond slaves swp5,swp6
2 net add interface isl.4000 clag enable yes
3 net add interface isl.4000 ip address 192.0.2.1/24
4 net add interface isl.4000 clag peer-ip 192.0.2.2
5 net add interface isl.4000 clag sys-mac 44:38:39:00:00:13
6 net add bridge bridge ports isl
```

Die Kommandos auf dem Partner sw02 sind identisch, bis auf die gespiegelten IPv4-Adressen in den Zeilen 3 und 4.

Peer-Link fertig? Wenn sw01 in seiner Ausgabe von `net show clag` „The peer is alive“ vermeldet, beginnt die Kanalbündelung zum Server sw13.

```
1 net add bond sw13 bond slaves swp3
2 net add bond sw13 clag id 13
3 net add bond sw13 bridge access 1
4 net add bridge bridge ports sw13
```

Der Bond erhält den passenden Namen `sw13` und zusätzlich die 13 als unglückliche `clag-id`. Lediglich Netzadapter *swp3* wird Mitglied des LACP-Clubs. Der zweite Adapter ist auf Switch sw02, welcher exakt dieselben Befehle erhält.
Das Szenario nutzt das neue Interface auf OSI-Ebene 2, und daher benötigt Cumulus Linux den Hinweis in Zeile 4, um daraus eine Netzbrücke zu machen.
Anschließend vermeldet Server sw13 stolz, dass zwei Leitungen dem Bündel beigetreten sind und eine Bandbreite von zwei Gbit/s zur Verfügung steht.

```
cumulus@sw13:~$ net show interface bond0
    Name   MAC                Speed  MTU   Mode
--  -----  -----------------  -----  ----  ----
UP  bond0  0c:12:24:13:ff:03  2G     1500  LACP
```

MLAG zwischen server2 and sw11–sw12

Weiter geht es synonym im unteren Bereich des Labornetzwerks. Dort wollen Teilnehmer server2 und das Pärchen aus sw11 und sw12 einen LACP-Kanal per MLAG bilden. Die Einrichtung entspricht inhaltlich dem oberen Aufbau. Switch sw11 und sw12 erstellen einen Peer-Link, wobei die folgenden Kommandos für sw11 passen.

```
1 net add bond isl bond slaves swp3,swp4
2 net add interface isl.4000 clag enable yes
3 net add interface isl.4000 ip address 192.0.2.11/24
4 net add interface isl.4000 clag peer-ip 192.0.2.12
5 net add interface isl.4000 clag sys-mac 44:38:39:00:00:11
6 net add bridge bridge ports isl
```

Die Konfiguration von server2 ist abhängig von seinem Betriebssystem, da es sich bei diesem Server nicht um Cumulus Linux handelt. Das Labordiagramm nimmt ein Red Hat-basiertes Linux, aber es könnte ebenso ein Windows Server, VMware ESXi oder eine Variante von BSD sein. Praktische Beispiele für verschiedene Betriebssysteme und deren Anbindung an einen Cumulus-Switch liefert Kapitel 19. Wichtig sind hier nur die gebündelte Verbindung und eine IPv4-Adresse, die im selben IP-Netz wie Server sw13 ist.

MLAG im Kern zwischen sw01–sw02 und sw11–sw12

Die Außenstellen sind verkabelt und konfiguriert, aber der Kern ist noch Niemandsland. Die Teilnehmer der MLAG haben aus den vorherigen Abschnitten bereits den notwendigen Peer-Link und können direkt anfangen zu bündeln.

Das obere Pärchen bestehend aus sw01 und sw02 schaltet seine Netzadapter *swp1* und *swp2* zu einem 4-Port-Bündel zusammen. Die frei gewählte ID dafür ist 1112. Auch hier erhält der Bond den Namen der benachbarten Switches, um bei der späteren Fehlersuche die Bündel leichter unterscheiden zu können. Die Einrichtung ist für sw01 und sw02 identisch:

```
net add interface swp1-2
net add bond sw11_12 bond slaves swp1,swp2
net add bond sw11_12 clag id 1112
net add bond sw11_12 bridge access 1
```

Die Konfiguration der Gegenstellen sw11 und sw12 sieht ähnlich aus:

```
net add interface swp1-2
net add bond sw01_02 bond slaves swp1,swp2
net add bond sw01_02 clag id 102
net add bond sw01_02 bridge access 1
```

Von dem erstellten 4 Gbit/s-Bündel sieht jeder einzelne Switch lediglich zwei Leitungen. Und die sollten im Status UP sein, wie das folgende Kommando bestätigt.

```
cumulus@sw11:~$ net show interface bonds
    Name      Speed   MTU  Mode  Summary
--  -------   -----  ----  ----  --------------------------------
UP  isl       2G     1500  LACP  Bond Members: swp3(UP), swp4(UP)
UP  server2   1G     1500  LACP  Bond Members: swp6(UP)
UP   sw01_02   2G      1500   LACP   Bond Members: swp1(UP), swp2(UP)
```

Zum Vergleich zeigt sw01 den Status seiner Netzadapter, Peer-Links und Bündel:

```
cumulus@sw01:~$ net show interface
State Name      Spd  MTU    Mode         LLDP          Summary
----- --------- ---  -----  ----------   -----------   ----------------------
UP    lo        N/A  65536  Loopback                   IP: 127.0.0.1/8
      lo                                               IP: ::1/128
UP    eth0      1G   1500   Mgmt         sw12 (eth0)   IP: 10.5.1.1/24
UP    swp1      1G   1500   BondMember   sw11 (swp1)   Master: sw11_12(UP)
UP    swp2      1G   1500   BondMember   sw12 (swp1)   Master: sw11_12(UP)
UP    swp3      1G   1500   BondMember   sw13 (swp1)   Master: sw13(UP)
UP    swp5      1G   1500   BondMember   sw02 (swp5)   Master: isl(UP)
UP    swp6      1G   1500   BondMember   sw02 (swp6)   Master: isl(UP)
UP    bridge    N/A  1500   Bridge/L2
UP    isl       2G   1500   LACP                       Master: bridge(UP)
      isl                                              Bond Members: swp5(UP)
      isl                                              Bond Members: swp6(UP)
UP    isl.4000  2G   1500   SubInt/L3                  IP: 192.0.2.1/24
UP    sw11_12   1G   1500   LACP                       Master: bridge(UP)
      sw11_12                                          Bond Members: swp1(UP)
      sw11_12                                          Bond Members: swp2(UP)
UP    sw13      1G   1500   LACP                       Master: bridge(UP)
      sw13                                             Bond Members: swp3(UP)
```

Damit ist das Szenario komplett. Der robuste Aufbau erlaubt den Ausfall einer beliebigen einzelnen Komponente, ohne dass die Endgeräte dies merken.

Virtual Router Redundancy

Wenn die beiden MLAG-Switches für die angeschlossenen Server das Standardgateway darstellen, entsteht *Virtual Router Redundancy* (VRR). Die Server adressieren die virtuelle IP-Adresse des VRR-Pärchens und einer von beiden Switches wird sich der Aufgabe annehmen.

Das Prinzip ähnelt HSRP, VRRP, CARP oder GLBP mit dem wesentlichen Unterschied, dass *beide* Switches aktiv sind und Anfragen der Server beantworten. Da es keine Abstimmung zwischen MASTER und BACKUP gibt, gibt es auch kein Redundanzprotokoll, keine Heartbeat-Pakete und keinen Linux-Prozess.
Das vereinfacht die Einrichtung, wenn MLAG bereits konfiguriert ist. Beide Switches erhalten eine zusätzliche IP-Adresse, wahlweise für IPv4 oder IPv6:

```
net add vlan 2 ip address-virtual 00:00:5e:00:01:02 10.2.1.5/24
net add vlan 2 ipv6 address-virtual 00:00:5e:00:01:02 fd00:2:1::5/64
```

Um den Ausfallschutz und die Lastverteilung kümmert sich bereits MLAG, sodass die virtuelle IP-Adresse nur noch oben aufgesetzt wird. Dementsprechend gibt es auch keine zusätzlichen `show`-Kommandos, um den Status zu prüfen oder zu ändern.

Cumulus Linux verwendet MLAG mit VRR *oder* VRRP. Wenn Cumulus Switches unter sich sind, ist VRR die bessere Wahl, da die Geräte Aktiv/Aktiv-Verfügbarkeit bieten. VRR benutzt kein zusätzliches Protokoll zwischen den Teilnehmern und folglich wird ein echtes VRRP-Gateway den Cumulus-Partner nicht erkennen. Wenn Cumulus Linux ein VRRP-Pärchen darstellen oder ergänzen soll, gibt es seit Version 3.7.4 eine passende Implementierung dazu, die allerdings nicht gleichzeitig mit MLAG verwendbar ist.
Mit den beispielhaften Kommandos wird sw01 ein VRRP-Router und offeriert seine Dienste per IPv4 und IPv6:

```
net add interface swp1 vrrp 1 10.2.1.5/24
net add interface swp1 vrrp 1 fd00:2:1::5/64
```

Zusammenfassung

Cumulus Linux kann mehrere Netzadapter zusammenschalten und alle involvierten Leitungen aktiv benutzen. Damit erreicht der Switch höhere Bandbreiten und gleichzeitig noch Ausfallschutz. Denn wenn eine einzelne Leitung in den Status DOWN wechselt, bleibt das Bündel aktiv und benutzt die verbliebenen Verbindungen für den Datentransport. Bei der Auswahl der Gegenstelle zeigt sich Cumulus Linux offen, denn es unterstützt den

marktüblichen Standard LACP, der Kanalbündelung herstellerunabhängig macht.
Die Vielzahl der parallelen Leitungen verhindert leider eins nicht: Der Single-Point-of-Failure ist der Switch, auf dem die Kabel stecken. Dazu hat Cumulus die Doppel-Chassis–Bündelung im Portfolio, die genau diese Einschränkung adressiert. Mit *Multi-Chassis Link Aggregation* dürfen die Enden des Bündels auf zwei verschiedene Switches verzweigen. Und dabei sind alle Verbindungen aktiv und tragen zur Gesamtbandbreite bei. Fällt jetzt ein Switch aus, bleibt ein Teil des Bündels online. Leider ist dieses Feature nicht standardisiert und folglich darf nicht mit anderen Anbietern gemischt werden.

Kanalbündelung mit LACP oder MLAG ist eine feine Methode für Ausfallschutz und Lastverteilung, wobei gleichzeitig die Bandbreite aller teilnehmenden Kabelverbindungen aktiv zur Paketweiterleitung beiträgt.