

# Dante für Einsteiger\*innen

*In einer Seminarreihe, die sich mit der digitalen Seite der Branche auseinandersetzt, werden Jörn und Sascha Euch einen Einblick in Dante-Digital-Audio-Netzwerke geben.*

*Es wird verständlich erklärt, Mythen aufgeklärt, Fallstricke genannt und die Ausflüge ins Netzwerk Nerdhausen werden sich in Grenzen halten. Versprochen!*

*Selbstverständlich werden auch tiefergehende Fragen gerne beantwortet.*

**Lernziel:**

*Nach ca. 90 Minuten sitzen die nötigen Begrifflichkeiten und Ihr könnt locker Euer neues, stressfreies Leben als Dante-User genießen.*

**Padlet Karte:**



**Padlet Fragen/Feedback**



**Download slides**



**11.13 10:00 UHR HIER**

**vdt**

Verband Deutscher  
Tonmeister e.V.

**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
[s.helmstaedt@gmx.de](mailto:s.helmstaedt@gmx.de)

Jörn Nettingsmeier  
[nettings@luchtbeweging.nl](mailto:nettings@luchtbeweging.nl)  
<https://luchtbeweging.nl>

**IGVW**  
4 EDUCATION



# Dante für Einsteiger\*innen

*In einer Seminarreihe, die sich mit der digitalen Seite der Branche auseinandersetzt, werden Jörn und Sascha Euch einen Einblick in Dante-Digital-Audio-Netzwerke geben.*

*Es wird verständlich erklärt, Mythen aufgeklärt, Fallstricke genannt und die Ausflüge ins Netzwerk Nerdhausen werden sich in Grenzen halten. – Versprochen!*

*Selbstverständlich werden auch tiefer gehende Fragen gerne beantwortet.*

**Lernziel:**

*Nach ca. 90 Minuten sitzen die nötigen Begrifflichkeiten und Ihr könnt locker Euer neues, stressfreies Leben als Dante-User genießen.*

**Padlet Karte:**



**Padlet Fragen/Feedback**



**Download slides**



**vdt**

Verband Deutscher  
Tonmeister e.V.

**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

Jörn Nettingsmeier  
nettings@luchtbeweging.nl  
<https://luchtbeweging.nl>

**IGVW**  
4 EDUCATION



2/48

# Themenübersicht

- Was ist Dante? Was ist Dante nicht?
- Auf welchem Netzwerk-*Layer* spielt Dante?
- Was man über IP-Adressen wissen muss
- Latenz im Dante-Netz
- Clocking und Synchronisation im Dante-Netz
- Einfache Netze: die *daisy chain*
- Einfache Netzwerke: der Stern
- Ausfallsicherheit von Netzen
- Redundante Netze (primary/secondary)
- Mehrere Switches: Netztopologien
- Multicasting in Dante-Netzen
- Switch-Tricks: VLANs + Trunking

# Dante ist komplex und kann viel.

Aber das meiste davon brauchen wir in einfachen Setups nicht. Deshalb geht das Seminar über viele Details hinweg. Das erkennst du an diesem Symbol:

D.h. in der Praxis und für Standard-Anwendungsfälle stimmt das Gesagte durchaus, aber wenn man genauer hinschauen möchte, müsste man es komplizierter ausdrücken. Machen wir gern für Dich, wenn Du fragst.



vdt

Verband Deutscher  
Tonmeister e.V.

**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
[s.helmstaedt@gmx.de](mailto:s.helmstaedt@gmx.de)

Jörn Nettingsmeier  
[nettings@luchtbeweging.nl](mailto:nettings@luchtbeweging.nl)  
<https://luchtbeweging.nl>

**IGWW**  
4 EDUCATION



# Dante ist komplex und kann viel.

Aber das meiste davon brauchen wir in einfachen Setups nicht. Deshalb geht das Seminar über viele Details hinweg. Das erkennst du an diesem Symbol:

D.h. in der Praxis und für Standard-Anwendungsfälle stimmt das Gesagte durchaus, aber wenn man genauer hinschauen möchte, müsste man es komplizierter ausdrücken. Machen wir gern für Dich, wenn Du fragst.



# Netzwerk-Technologie ist schwierig.

Wir wissen auch nicht alles. Das ist auch ok so.

Wichtig für erfolgreiche Einsätze ist:

Du musst wissen, was Du verstanden hast (das kannst Du dann benutzen) und was nicht (das benutzt Du dann lieber nicht).

So machen wir das auch.

vdt

Verband Deutscher  
Tonmeister e.V.

**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

Jörn Nettingsmeier  
nettings@luchtbeweging.nl  
<https://luchtbeweging.nl>

**IGVW**  
4 EDUCATION



# Was ist Dante?

-  **Dante**<sup>®</sup> ist ein proprietäres Produkt der Firma Audinate. Wer es in seinen Produkten nutzen möchte, muss die Hardware dieser Firma kaufen und in sein Gerät einbauen.
- Dante überträgt digitale Audio-Daten in einem Kupfer- oder Glasfasernetzwerk.
- Dante arbeitet als **Kreuzschiene**. Das ist eine Art Signaltabelle, die man sich z.B. im **Dante Controller** ansehen kann. Die Spalten sind Signalquellen, die Zeilen sind die Ziele oder Signal“senken“. Quelle und Senke werden verbunden, indem man im entsprechenden Feld ein Häkchen setzt.
- Anders als bei den ersten **digitalen Multicoresystemen**, die nur *zwei* Endpunkte miteinander verbunden haben, kann Dante flexibel zwischen *vielen* Geräten Signale hin- und her transportieren.
- Dante arbeitet mit minimaler **Latenz** (in der Größenordnung von einer viertel bis wenigen Millisekunden).

# Was ist Dante nicht?

- Wo Audio-Daten herkommen, wo sie hingehen, und wie sie bearbeitet werden, ist Dante egal. Es ist also

**kein Wandler, Signalprozessor oder Plugin-Host**

- Dante verbindet Signalquellen und Signal“senken“ (=Empfänger).

Eine Quelle kann auf beliebig viele Senken verteilt werden.

Eine Senke kann aber immer nur eine Quelle empfangen. Dante ist also auch

**kein Mischer**

- Dante hat eine globale Clock und eine globale Sampling Rate. Es ist

**kein Resampler**

- Dante taugt nicht für unzuverlässige Links (WiFi oder Internet). Es ist

**kein Streaming-Service**



**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

Jörn Nettingsmeier  
nettings@luchtbeweging.nl  
<https://luchtbeweging.nl>

# Auf welchem Netzwerk-Layer spielt Dante?

- Dante braucht ein **physikalisches Netzwerk** aus Kabeln oder Glasfasern und Sende/Empfängerelektronik, mit dem man Daten senden und empfangen kann (**Layer 1, die Post**).
- Die Art, wie die Daten verpackt und verschickt werden, ist auch festgelegt und heißt **Ethernet (Layer 2, das Paket)**. Ethernet ist für das lokale Netzwerk aus Endgeräten und Switches. Jedes Gerät hat eine Hardware-Adresse (MAC), und wie in einer Nachbarschaft weiß jedes Gerät, wo in der eigenen Straße die anderen wohnen.

Wenn man auf Layer 2 bleibt, hat man sehr wenig **Overhead** (= Bandbreiten-Verschwendung für Adressen), aber man kann die Daten **nicht routen** (= nicht das lokale Netzwerk verlassen). AVB/MILAN arbeitet so.

- Das Ethernet-Paket muss jetzt noch eine *Absender- und Empfängeradresse* bekommen und der Inhalt kommt in eine standardisierte Verpackung. Damit haben wir unseren **Layer 3, das Internet Protocol (IP)**.

# Dante ist ein Layer-3-Protokoll

- Dante arbeitet mit **IP-Paketen**. Dank deren Empfänger- und Absenderadresse können Dante-Audio-Signale über Netzwerkgrenzen hinweg zugestellt (= geroutet) werden.
- Dante-Pakete sind „fire and forget“ (**UDP, user datagram protocol**). Man erfährt nicht, ob ein Paket angekommen ist. Wenn ein Paket verloren geht, gibt es einen hörbaren Aussetzer im Ton.



Das ist ein Grund, warum Dante in WLANs und über Weitbereichsnetze nicht funktioniert.

- Zusätzlich zu **UDP** gibt es auch **TCP**, das **transport control protocol**. TCP-Pakete sind „per Einschreiben mit Rückschein“ – wenn eines nicht beim Empfänger ankommt, kommt keine Empfangsbestätigung und der Sender muss es noch einmal schicken. Das dauert aber für Audio-Daten mit minimaler Latenz viel zu lange und wird deshalb in Dante nicht verwendet.

# Dante ist ein Layer-3-Protokoll

- Dante-Geräte brauchen korrekte IP-Adressen, um miteinander zu funktionieren.
- Eine IP-Adresse besteht aus 4 **Bytes**. Jedes Byte kann irgendwo zwischen 0 und 255 liegen. Man schreibt sie meistens in Dezimalnotation mit Punkt dazwischen. Das sieht dann so aus: **192 . 168 . x . y** oder **10 . x . y . z**
- Jede IP-Adresse darf natürlich nur einmal vorkommen, sonst gibts Probleme bei der Paketzustellung. Deswegen ist die Adressvergabe durch internationale und nationale Stellen reglementiert.

# Dante ist ein Layer-3-Protokoll

Es gibt drei Möglichkeiten, korrekte IP-Adressen zu bekommen:

- Man vergibt händisch **statische** (= vom Benutzer festgelegte, unveränderliche) Adressen.
- Man überlässt die IP-Konfiguration einem Server-Dienst, dem Dynamic Host Configuration Protocol (**DHCP**). Auf Routern (und manchen *managed switches*) läuft oft ein DHCP-Server.
- Man überlässt es den Geräten, sich selbst Adressen zu suchen. Diese heißen **Link-Local-Adressen**, weil sie nur im lokalen Netz gültig sind, und haben die Form **169 . 254 . x . y**

# Was man über IP-Adressen wissen muss

- Geräte im selben Netzwerk können direkt miteinander reden.

Ich packe deine IP-Adresse auf das Paket und es kommt einfach bei Dir an. Deine Antwort an mich ebenso.

- Geräte in verschiedenen Netzwerken (= Internet, „zwischen den Netzen“) können nur dann miteinander reden, wenn sich **Router** um den Datentransport kümmern und jedes Endgerät seinen Router „kennt“.

Ich packe Deine IP-Adresse auf das Paket und schicke es an meinen **default router**. Der weiß, an wen er es weiterschicken muss, damit es bei Dir ankommt – dazwischen können viele weitere Router (= hops, „Sprünge“) sein. Ich muss das nicht wissen.

- Ob zwei IP-Adressen im selben Netzwerk sind, bestimmt die **Netzmaske**.

# Was man über IP-Adressen wissen muss.



**vdt**

Verband Deutscher  
Tonmeister e.V.

**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
[s.helmstaedt@gmx.de](mailto:s.helmstaedt@gmx.de)

Jörn Nettingsmeier  
[nettings@luchtbewegung.nl](mailto:nettings@luchtbewegung.nl)  
<https://luchtbewegung.nl>

**IGWW**  
4 EDUCATION



# Was man über IP-Adressen wissen muss.

- Eine Netzwerkkonfiguration eines Gerätes besteht aus
  - der eigenen IP-Adresse **w . x . y . z** (von 0 bis 255, d.h. je acht Bit = ein **Byte**)
  - der Netzwerk**maske**. Sie „maskiert“ den Netzwerk-Teil der IP-Adresse. Alles was übrig bleibt, sind Geräte-Nummern im lokalen Netz.

Beispiel: meine Adresse ist **192 . 168 . 42 . 17**, Netzwerkmaske **255 . 255 . 255 . 0**

255 in binär bedeutet „alle Bits auf 1“. Die Netzwerkmaske verdeckt also jetzt die kompletten ersten drei Bytes, übrig bleibt das letzte. Ich bin also im lokalen Netz die Nummer 17 und maximal passen 254 Geräte in dieses Netz, inklusive Router.

Manchmal sieht man auch ein **Netzwerk-“Prefix“** statt der Maske, das sieht dann so aus: **192 . 168 . 42 . 17 / 24**.

Das bedeutet: die ersten 24 Bits sind der Netzwerk-Teil der Adresse.

# Was man über IP-Adressen wissen muss.

- Außerdem gibt es noch die **Broadcast-Adresse** für Durchsagen an alle Geräte im lokalen Netz. Sie ist immer die höchste Adresse des Netzwerks, ergibt sich also aus Geräte-Adresse und Netzwerkmaske. Deswegen muss man sie meistens nicht extra einstellen. In unserem Beispiel wäre das die **192 . 168 . 42 . 255**.
- Zuletzt braucht man noch die Adresse des Standard- oder **default-Routers**. Dieser weiß, wie man Geräte erreicht, die außerhalb des lokalen Netzes liegen. In den meisten Fällen ist das die niedrigste Adresse im Netz [also im Beispiel die **192 . 168 . 42 . 1**] oder die höchste unter der Broadcast-Adresse [also die **192 . 168 . 42 . 254**]. Da man sie aber im Prinzip frei wählen kann, muss man sie extra einstellen.

Ein Gerät muss alle Pakete, deren Empfänger nicht zum lokalen Netz gehört, an den Default-Router schicken.

# Statische Netzwerkkonfiguration



Wenn Du ein eigenes Netz aufbaust, darfst Du nur bestimmte Adressbereiche benutzen, die sogenannten **private networks**.

Wenn Du nur maximal 254 Geräte brauchst, nimmst Du ein Netzwerk der Form **192 . 168 . x . y** mit der Netzmaske **255 . 255 . 255 . 0** (oder /24).

Es gibt also **x** verschiedene private Netzwerke mit bis zu **y** Einzeladressen, beides ist je ein Byte, also  $2^8$ , also 256.

Wenn Du mehr Geräte brauchst, nimmst Du ein Netzwerk der Form **10 . x . y . z** mit der Netzwerkmaske **255 . 255 . 0 . 0** (oder /16).

Es gibt also **x** (= 256) verschiedene Netzwerke mit jeweils **y \* z =  $2^{16}$  = 65536** Adressen.

Jetzt weißt du genug, um jedem Gerät eine korrekte **statische** IP-Adresse und Netzmaske zu geben, so dass sie im lokalen Netz miteinander reden können, ohne dass Du einen DHCP-Server brauchst.

Aber...

# Statische Netzwerkkonfiguration



...statische Netzwerkadressen sind nur eine gute Idee, wenn

- die beteiligten Geräte immer dieselben sind und auch immer nur in dieser Kombination eingesetzt werden (z.B. in einer Festinstallation)
- Du genau weißt, wie Du die Netzwerkadresse am Gerät einfach nachschauen und ändern kannst, z.B. über ein Display und Menü, oder eine entsprechende Konfigurations-Software

Wie jetzt, Konfigurations-Software? Wie findet die denn das Gerät, dem ich damit eine IP-Adresse zuweisen will, wenn das Gerät noch gar keine Adresse hat?

Ganz einfach: mit automatischer Netzwerkkonfiguration ohne DHCP-Server.

# Automatische Konfiguration ohne DHCP

Du erinnerst Dich: im lokalen Netz ging der Paketversand auch ohne IP, weil die Geräte sich ihre Hardware- („MAC“-)Adressen gegenseitig zurufen können. Damit kann man eine Adressvergabe ohne zentrales Kontrollorgan (also ohne DHCP-Server) hinkriegen.

Die Geräte handeln untereinander Adressen aus dem **Link-Local-Bereich** aus. Diese Adressen sehen so aus: **164.254. x . y /16**.

Dieser Trick hat viele Namen: Zeroconf, Bonjour, Rendezvous, oder ganz trocken „link-local-Adressierung“.



Für ältere Netzwerk-Admins so wie Jörn ist die Vorstellung ein bisschen gruselig, denn niemand garantiert dir, dass die Geräte beim nächsten Einschalten die gleichen Adressen bekommen (kriegen sie nicht!). Mit althergebrachten Fehlersuche-Tools wie **ping** findest Du sie also nicht wieder.

Link-Local-Adressen machen nur Spaß, wenn die Anwendung die Geräte auf andere Weise identifizieren kann. Dante kann das anhand des **device name** und einem Netzwerk-**Broadcast**.

# Automatische Konfiguration ohne DHCP

Bei Dante-Geräten im Verleih- und Touring-Einsatz mit häufig wechselnden Gerätekombinationen ist automatische Adressierung im 164.254.0.0/16-Bereich die einzig vernünftige Lösung.

Man muss sich einfach daran gewöhnen, dass IP-Adressen den Geräten nicht mehr fest zugeordnet sind, und der Dante Controller regelt den Rest.

# Kompromiss: DHCP

Wer Geräte in wechselnden Kombinationen betreibt *und* sich vor automagischer Adressvergabe gruselt, kann auch einen DHCP-Server in Form eines Routers oder Layer-3-Switches betreiben. Dieser kann die Geräte automatisch mit Adressen versorgen *und* man kann an zentraler Stelle die Geräte-Adressen nachlesen.

Der Nachteil ist ein zusätzliches Gerät auf der Baustelle, das dann auch korrekt konfiguriert werden muss.

Viele Dante-Geräte (aber leider nicht alle) lassen sich so einstellen, dass sie zunächst nach einem DHCP-Server suchen und ihn auch benutzen, und nur wenn keiner da ist auf link-local-Konfiguration wechseln.

# Latenz im Dante-Netz

Dante-Pakete brauchen eine bestimmte Zeit, um den Empfänger zu erreichen. Diese Zeit ist die minimale Netzwerk-Latenz. Sie besteht aus:

- dem Sende-Puffer, in dem Daten für ein Paket gesammelt werden (üblicherweise 16 Frames Audio, d.h. bei 48 kHz sind das  $16 / 48000 = 0,333$  ms)
- der Zeit, die ein Paket vom Sender zum Empfänger braucht (die halbe Ping-Zeit, ungefähr 0,1 bis 0,3 ms), darin steckt die Switching-Latenz (50 bis 150  $\mu$ s)
- der konfigurierten Dante-Latenz, d.h. der Sicherheitsreserve, bis das empfangene Paket ausgespielt werden muss.

Dante-Datenpakete haben eine „**presentation time**“: einen Zeitstempel, zu dem sie ausgespielt werden müssen.

# Latenz im Dante-Netz

Man kann mit dem Dante-Controller eine **Standard-Latenz** für das Dante-Netz und/oder eine Latenz für jedes Gerät einstellen.

Für den Wert ist Netzqualität und die Anzahl der Switches maßgeblich, durch die das Signal hindurch muss:

- Gigabit-Netz, drei Switches: **0,25 ms** (8,6 cm Flugzeit)
- Gigabit-Netz, fünf Switches: **0,5 ms** (17,2 cm Flugzeit)
- Gigabit-Netz, zehn Switches  
oder 100Mbit-Netz: **1,0 ms** (34,4 cm Flugzeit)



Viele Geräte haben intern schon einen Switch eingebaut, d.h.

Mischpult → Switch → Stagebox = **3 Switches!**

Diese Standard-Latenz wird nur verwendet, wenn beide Geräte das können. Braucht eines länger, dann wird zwischen diesem und seinen Partnern (aber nur hier!) eine höhere Latenz ausgehandelt. Beispiel: Dante Virtual Soundcard mit **4 ms**.

**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

Jörn Nettingsmeier  
nettings@luchtbeweging.nl  
<https://luchtbeweging.nl>

# Clocking im Dante-Netz

Dante-Geräte synchronisieren sich über das Netzwerk. Dabei wird ein Gerät zur Master-Clock gewählt.

Der Clock-Master schickt mir ein Paket, darin steht „es ist jetzt  $t_1 = 12\text{h}00:00.000$ “. Ich notiere mir meine Empfangszeit, sagen wir  $t_1^* = 12\text{h}00:00.003$ .

Jetzt will ich rauskriegen, wie viel meine Uhr zu spät ist: nennen wir das  $\Delta t$  („delta-t“, für „Differenz der Uhrzeit“). Aber dazu muss ich ja wissen, wie lange das erste Paket unterwegs war: das nennen wir mal  $\Delta p$  („delta-P“, für die Paketlaufzeit).

Aktuell weiß ich nur, dass der Unterschied der beiden Zeiten  $t_1$  und  $t_1^*$  die Summe aus meinem Zu-spät-gehen und der Paketlaufzeit ist:

$$\Delta t + \Delta p = t_1^* - t_1 = 3 \text{ ms}$$

# Clocking im Dante-Netz

Ich schicke ein Paket an den Clock-Master zurück.  
Darin steht „bei mir ist jetzt  $t_2 = 12\text{h}00:00.005$ “.

Der Clock-Master notiert sich die Empfangszeit bei sich und antwortet:  
„Dein Paket ist um  $t_2^* = 12\text{h}00:00.006$  angekommen“.

Mein Zu-spät-gehen wirkt ja jetzt genau andersrum: die Zeit zwischen Absenden  
und Empfangen wird kleiner. Wir wissen nun:

$$\Delta p - \Delta t = t_2^* - t_2 = 1 \text{ ms}$$

Zwei Gleichungen, zwei Unbekannte, das geht!

# Clocking im Dante-Netz

$$\Delta p - \Delta t = (t_2^* - t_2)$$

Wir addieren auf beiden Seiten das Zu-spät-gehen und kriegen die Paket-Latenz:

$$\Delta p = (t_2^* - t_2) + \Delta t$$

Das können wir nun in die erste Gleichung  $\Delta t + \Delta p = t_1^* - t_1$  einsetzen:

$$\Delta t + (t_2^* - t_2) + \Delta t = (t_1^* - t_1)$$

$$\Leftrightarrow 2 \Delta t = (t_1^* - t_1) - (t_2^* - t_2)$$

$$\Leftrightarrow \Delta t = (t_1^* - t_1) - (t_2^* - t_2) / 2$$

D.h. meine Uhr geht  $\Delta t = (3 \text{ ms} - 1 \text{ ms}) / 2 = 1 \text{ ms}$  nach und ich stelle sie jetzt richtig.

# Clocking im Dante-Netz

Wir erinnern uns: die Netzwerklatenz war

$$\Delta p = (t_2^* - t_2) + \Delta t$$

Das kann ich jetzt auch ausrechnen:

$$\Delta p = (1 \text{ ms}) + 1 \text{ ms} = 2 \text{ ms}$$

Jetzt weiß ich also, dass das nächste Clock-Paket mit der neuen Uhrzeit vom Master 2 ms brauchen wird, bis es bei mir ankommt, und ich kann meine Uhr dann direkt passend stellen.

Mit diesem Trick können jetzt alle Geräte ihre Clock zum Master synchronisieren, trotz Übertragungslatenz. Dieses Verfahren heisst **Precision Time Protocol**, oder **PTPv1**.

# Clocking im Dante-Netz

Wichtig: das PTP-Signal ist **kein Clock-Takt!**

Vielleicht kennst Du aus Deiner Praxis **Geräte mit Wordclock-, ADAT- oder AES/EBU-Schnittstellen**: hier wird tatsächlich ein Clock-Takt übertragen und die Empfangsgeräte **brauchen ihre eigene Uhr nicht**.

Bei Dante ist das anders:

**Geräte bestimmen Sende- und Empfangszeiten nach ihrer eigenen Uhr.**  
Aber PTP sorgt dafür, dass alle Uhren im Netz genau gleich gehen.

vdt

Verband Deutscher  
Tonmeister e.V.

**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

Jörn Nettingsmeier  
nettings@luchtbeweging.nl  
<https://luchtbeweging.nl>

**IGVW**  
**4 EDUCATION**



27/48

# Clocking im Dante-Netz

Anders als in diesem einfachen Beispiel erreicht das **Precision Time Protocol v1** in lokalen Netzen eine Präzision von  $< 0,1 \mu\text{s}$ , das ist weniger als ein hundertstel Sample bei 48 kHz.

Höchste Genauigkeit und minimale Latenz schafft man aber nur, wenn die Ethernet-Hardware die Zeitstempel selbst erstellen kann. Das können normale Computer-Netzwerkschnittstellen nicht (da muss das die CPU machen), und darum hat z.B. die Dante Virtual Soundcard auch eine Mindestlatenz von 4 ms und kann nicht als Clock im Netz dienen.

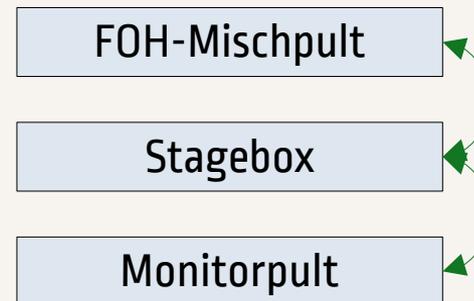
Fun fact am Rande: andere Technologien wie AES67/Ravenna benutzen das inkompatible **PTPv2**. Wenn man Dante- und AES67-Datenströme verbinden will, braucht man eine Art Uhren-“Adapter“, eine **boundary clock**.

# Einfache Netze: die *daisy chain*

Viele Dante-Geräte haben zwei Netzwerkanschlüsse: den **primary** und den **secondary port**.

Wenn das Gerät auf „**redundant**“-Betriebsmodus eingestellt ist, dient das der Ausfallsicherheit (jedes Gerät ist auf zwei unterschiedliche Weisen mit den andern verbunden, d.h. ein Netz kann ausfallen, ohne dass man eine Störung hört).

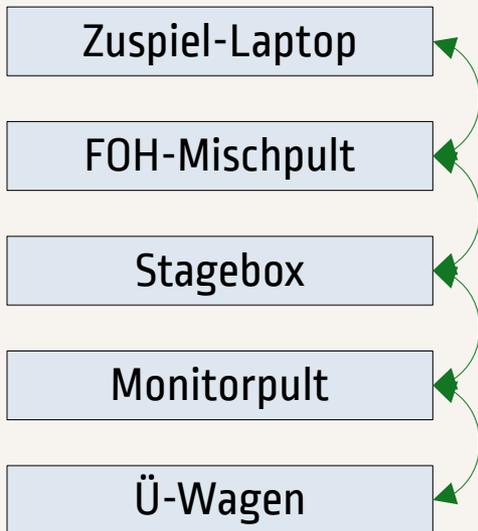
Für einfache Setups gibt es auch den „**daisy chaining**“-Modus: jedes Gerät verbindet über einen internen Switch seine beiden Ports, und ein Gerät hängt am nächsten, wie bei einer Gänseblümchenkette:



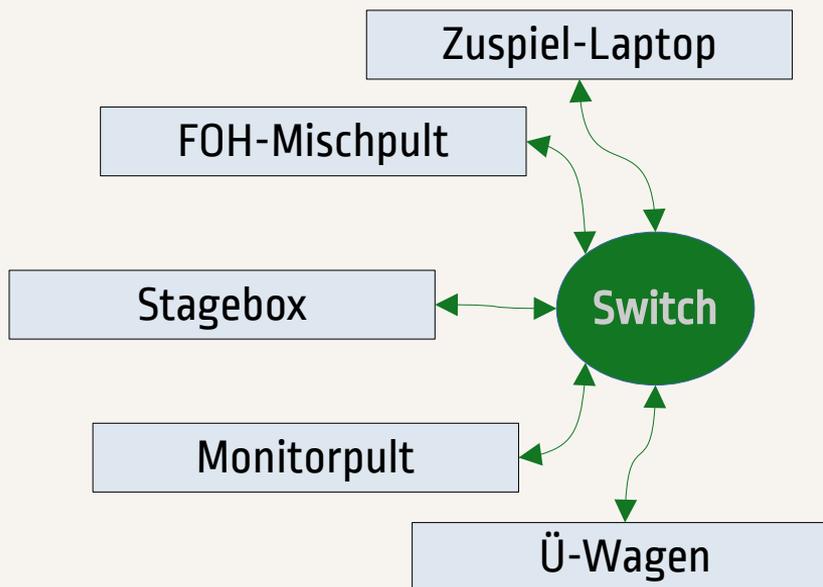
# Einfache Netze: der Stern

Alternativ kann man einen Switch einsetzen.  
Dieser bildet den zentralen Sternpunkt, und jedes Gerät ist direkt mit ihm verbunden.

Daisy chain:



Stern:



# Planen für den Fall der Fälle

Wenn was schiefeht, nennt man das **Havarie**. Es ist wichtig, sich vor der Produktion zu überlegen, was passieren kann und welche Konsequenzen das hat.

Dazu betrachtet man den schwierigsten Fall („**worst case**“-Analyse), d.h. bei einem Audio-Datennetzes den längsten Signalweg. Jeder Fehler, der *für sich allein* zum Versagen des System führt, heißt **single point of failure** (später mit „SPoF“ abgekürzt).

Weiterhin kann man sich noch überlegen, wie schlimm ein Ausfall ist (Pausenmusik-Zuspieler fällt aus: nicht so schlimm, Hauptbeschallung fällt aus: sehr schlimm).

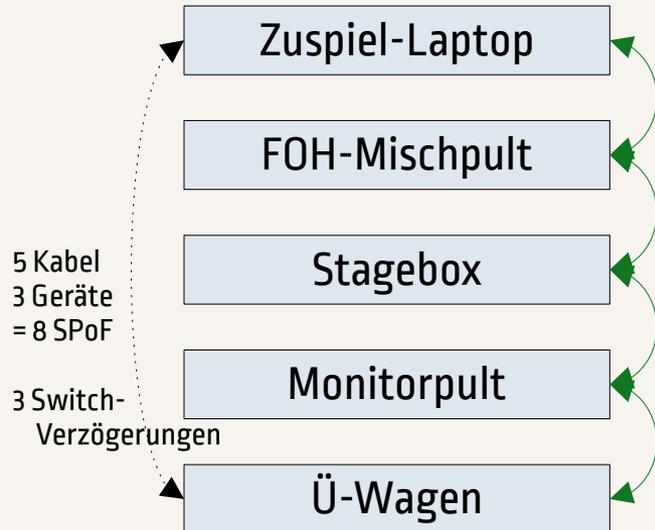
Im Fall einer Havarie muss man **mit dem Ausfall leben**, den **Fehler beheben**, oder ein **Havariesystem** an den Start bringen. Dabei ist auch die Dauer des Ausfalls entscheidend.

Beispiel: Club-PA ist tot, Ersatzsystem holen und aufbauen dauert eine Stunde. Zu lang. Wir drehen als Havarielösung die Monitorboxen ins Publikum und spielen (eingeschränkt aber sofort) weiter.

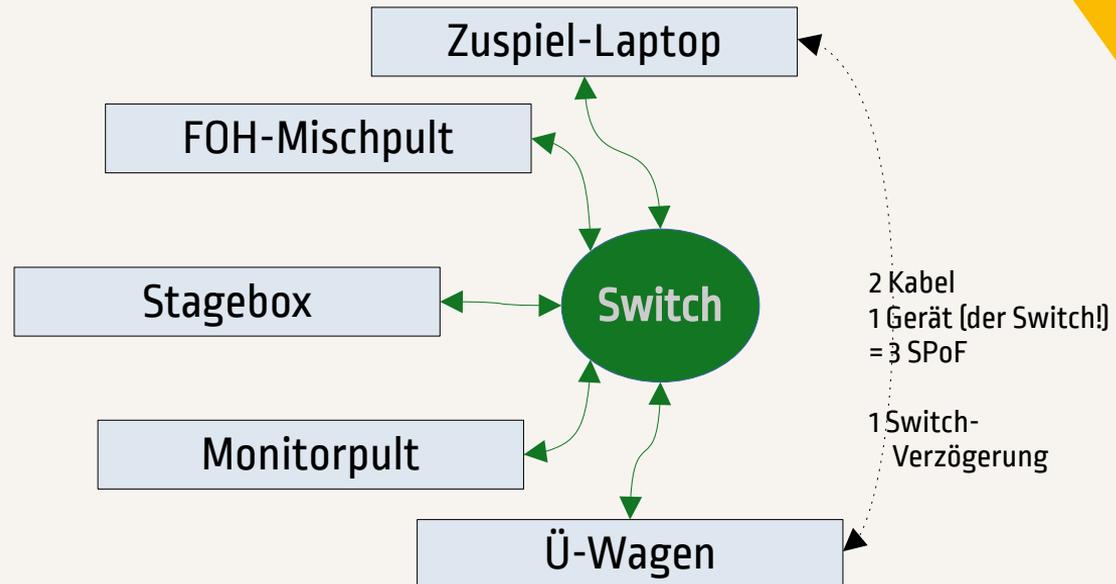
# Wie robust ist mein Netz?

Wir betrachten den längsten Weg unserer daisy chain, vom Zuspiel-Laptop bis zum Ü-Wagen (beim Stern ist das egal, jeder Weg ist gleich lang). Den Ausfall von Sendee- und Empfangsgerät ignorieren wir - wir wollen ja nur wissen, wie robust das Netz ist. Und wir zählen, wie viele Switch-Verzögerungen es unterwegs gibt.

## Daisy chain:



## Stern:



**Dante** für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

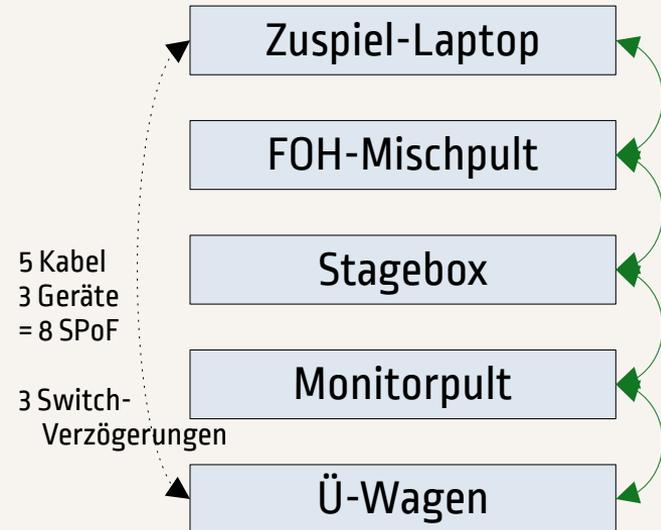
Jörn Nettingsmeier  
nettings@luichtbewegung.nl  
<https://luichtbewegung.nl>

# Wie robust ist mein Netz?

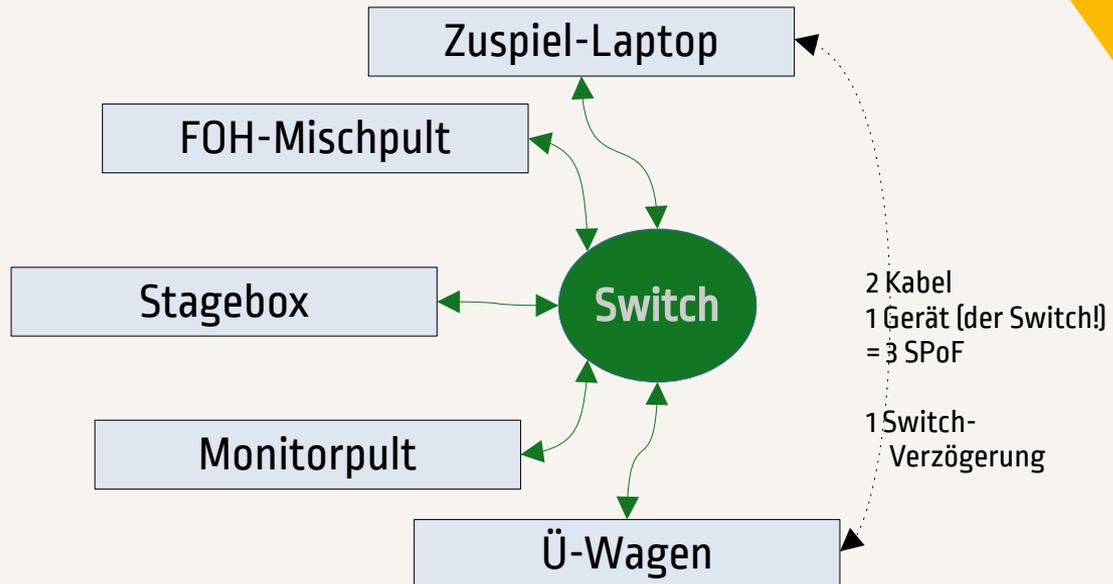
Man sieht: das daisy-chain-System hat mehr als die doppelte Ausfallwahrscheinlichkeit, und eine viel höhere Paketlaufzeit.

Der Stern ist in der Praxis immer vorzuziehen, mit einer Ausnahme...

## Daisy chain:



## Stern:



**Dante** für  
Einsteiger\*innen

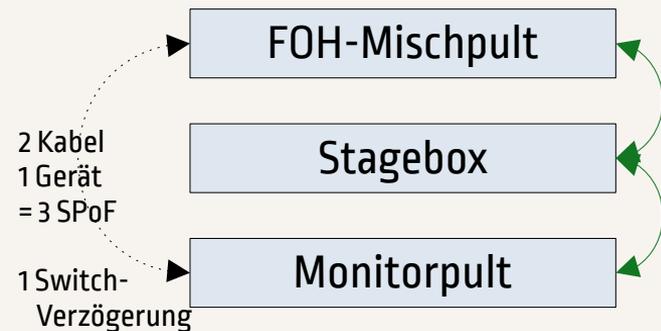
Sascha „Mon“ Helmstädt  
[s.helmstaedt@gmx.de](mailto:s.helmstaedt@gmx.de)

Jörn Nettingsmeier  
[nettings@luchtbeweging.nl](mailto:nettings@luchtbeweging.nl)  
<https://luchtbeweging.nl>

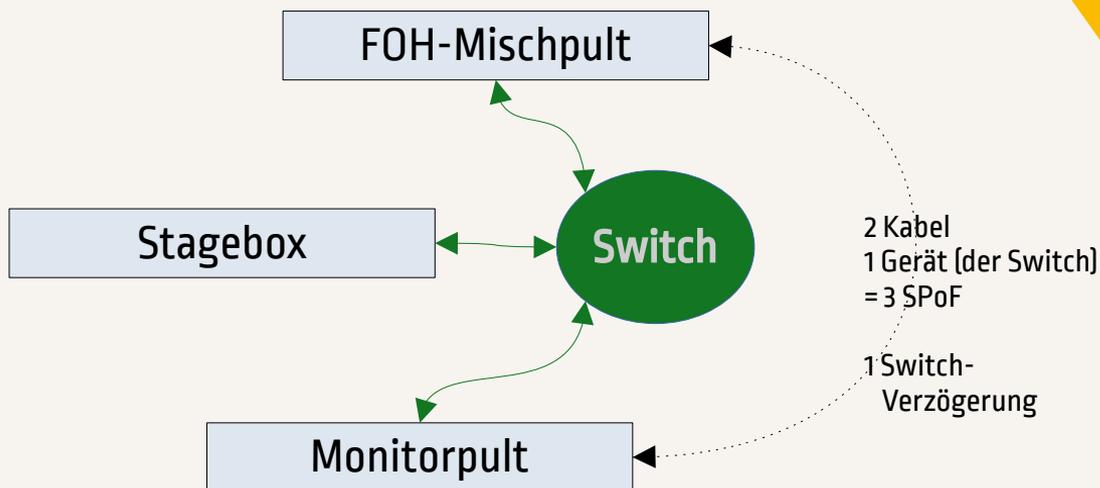
# Wie robust ist mein Netz?

... nämlich sehr kleine Netze: bei drei Geräten ist die Ausfallwahrscheinlichkeit für Stern und Daisy chain für den worst-case-Signalweg gleich hoch, und das Gesamtsystem hat ein Gerät weniger. Kann man also im Notfall mal machen:

Daisy chain:



Stern:



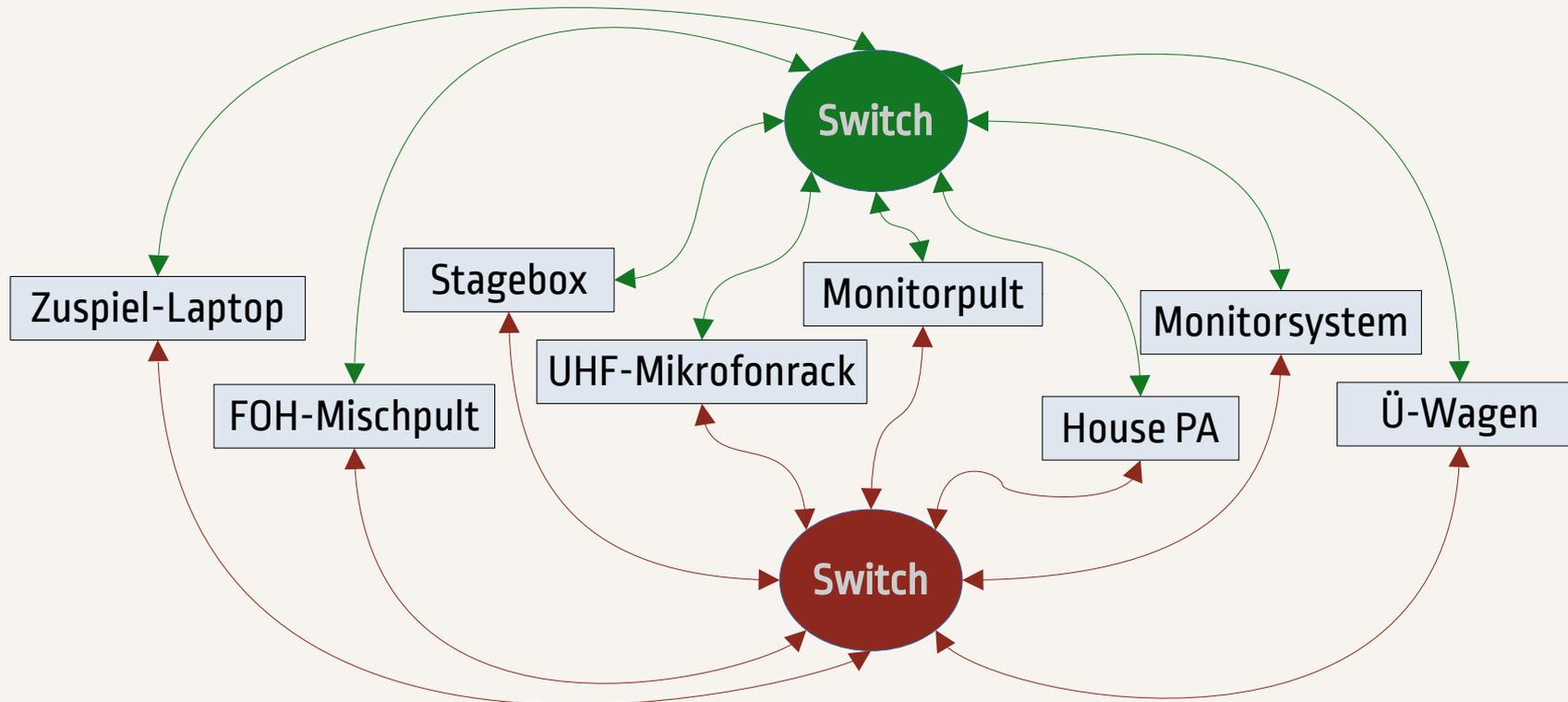
Dante für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

Jörn Nettingsmeier  
nettings@luchtbeweging.nl  
<https://luchtbeweging.nl>

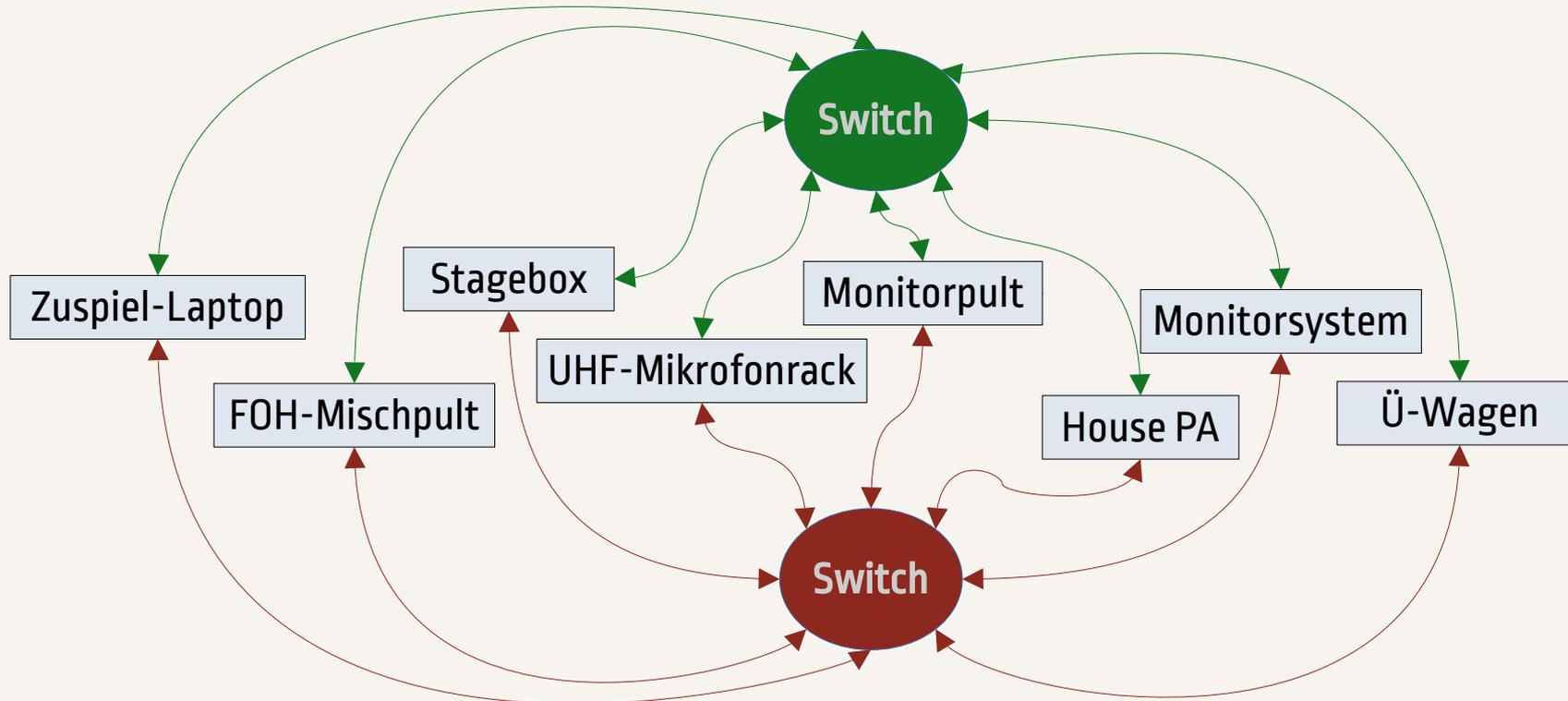
# Redundanz im Dante-Netz

Man kann zwei unabhängige Netze zwischen allen Geräten aufspannen, das **primary** und das **secondary network**. Das ist „mehr, als man eigentlich braucht“, also **redundant**.



# Redundanz im Dante-Netz

Alle Geräte senden ihre Daten in beide Netze. Verwendet werden aber nur die Daten aus dem **primären** Netz. Wenn es ausfällt, schaltet Dante innerhalb der normalen Paketlatenz auf das **sekundäre** Netz um, also **ohne Aussetzer** und **unhörbar**.

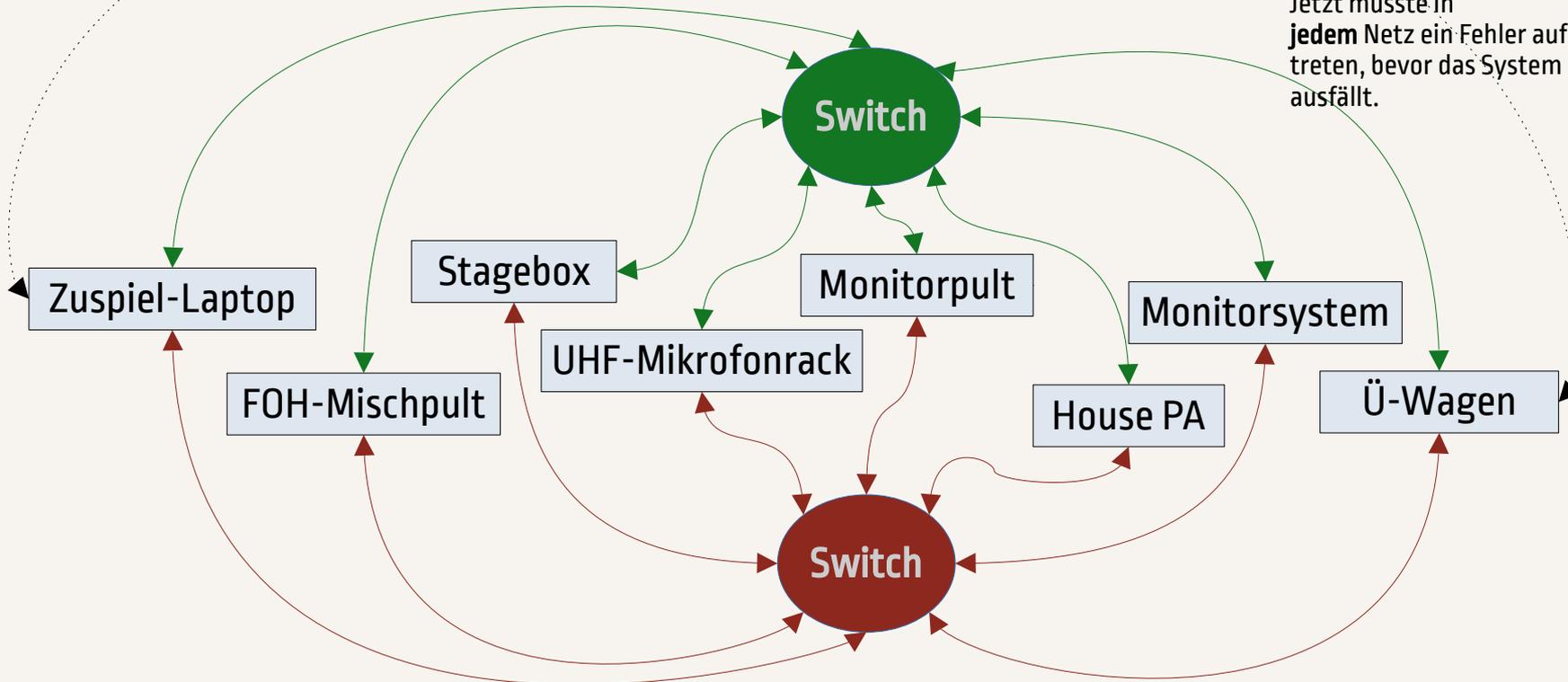


# Redundanz im Dante-Netz

Wie sieht jetzt die Havariewahrscheinlichkeit aus?

4 Kabel  
2 Geräte (die Switches)

Aber: es gibt keine  
**single points of failure** mehr.  
Jetzt müsste in  
**jedem** Netz ein Fehler auf-  
treten, bevor das System  
ausfällt.



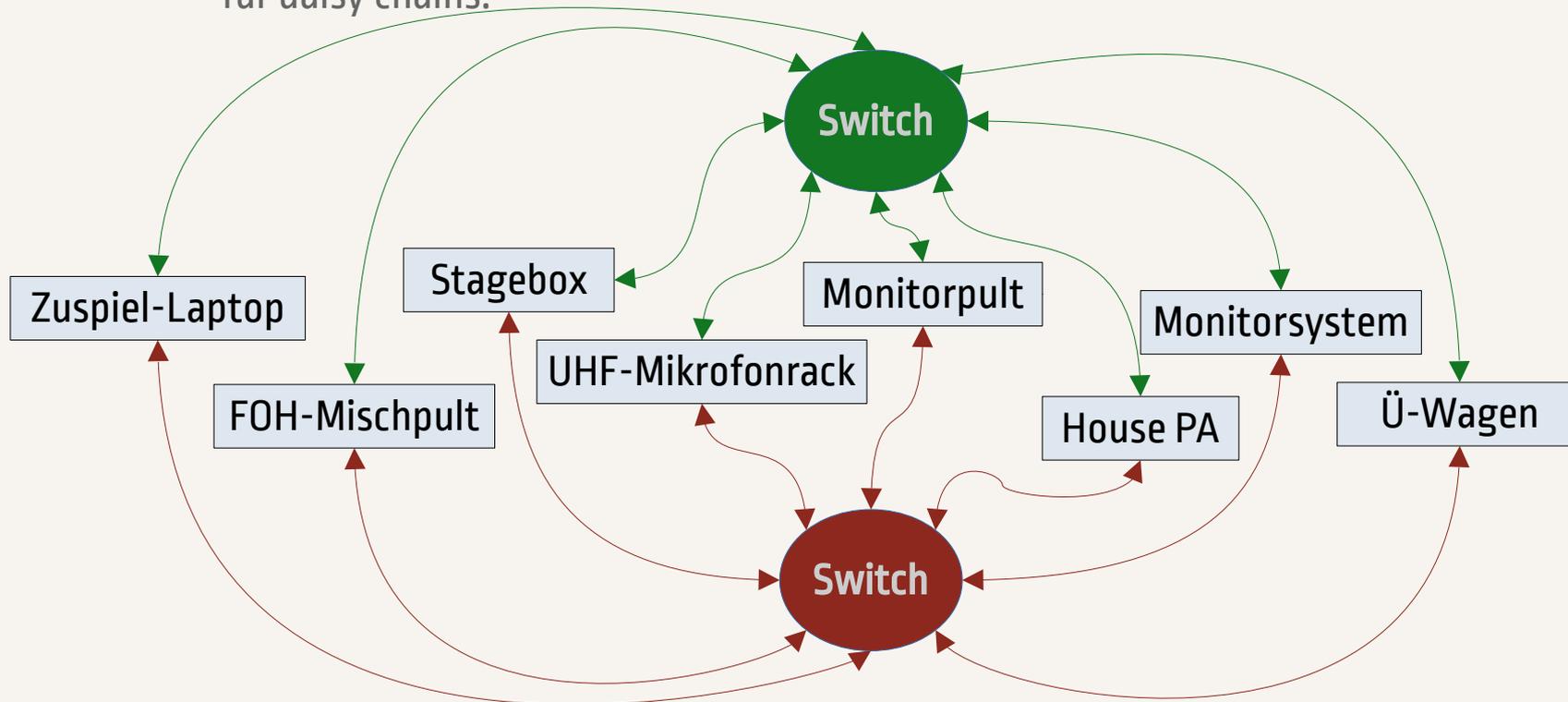
Dante für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

Jörn Nettingsmeier  
nettings@luchtbeweging.nl  
<https://luchtbeweging.nl>

# Redundanz im Dante-Netz

Man sieht: Redundanz ist spitze und kostet nur ein paar Kabel und einen Switch extra, und damit gibts für professionelle Anwendungen eigentlich keine Ausrede für daisy chains.



# Mehrere Switches: Netztopologien

Das Wort „**Topologie**“ kommt aus der Mathematik und meint: die geometrische Beschaffenheit einer Oberfläche.

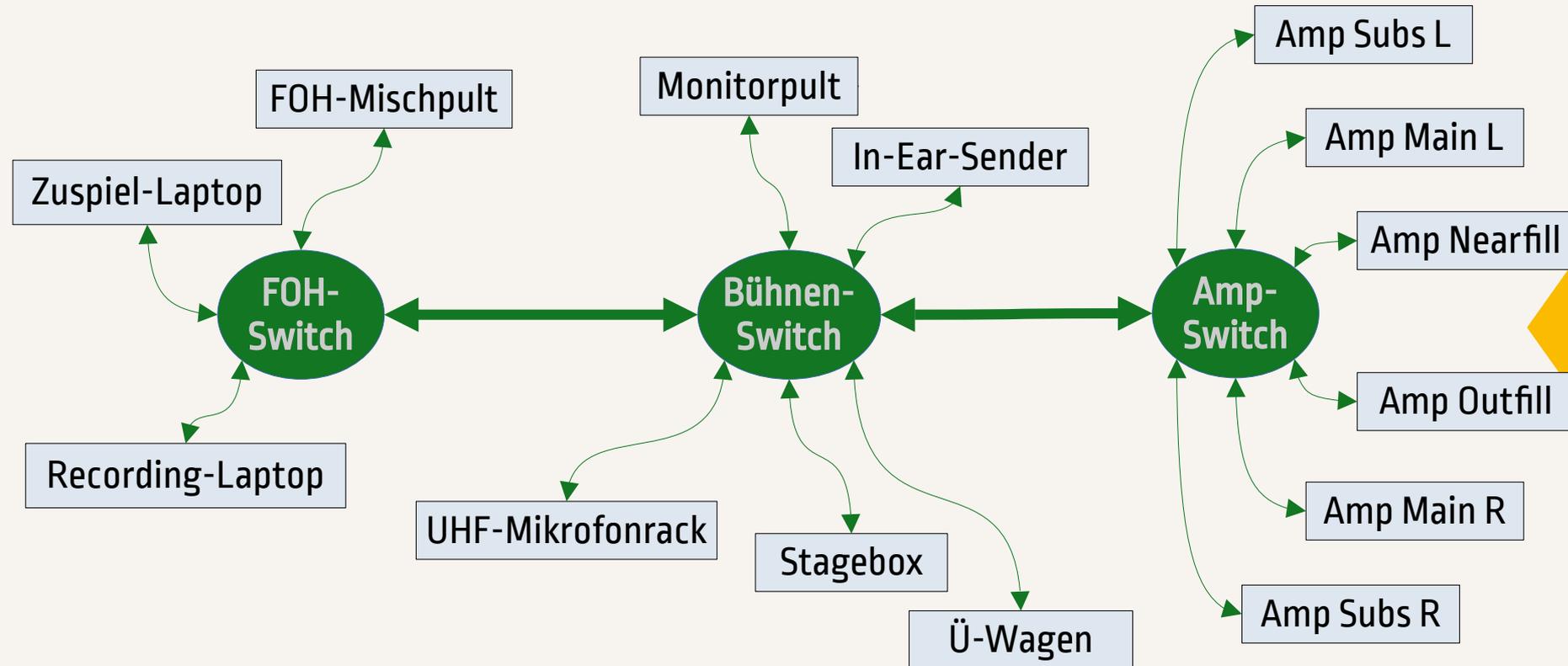
Ein Topologe kann eine Tasse nicht von einem Autoreifen oder einem Donut unterscheiden, weil alle drei genau ein Loch haben (klingt komisch, ist aber so).

Datennetze können auch unterschiedlich aussehen, aber im Prinzip die gleiche Topologie haben.

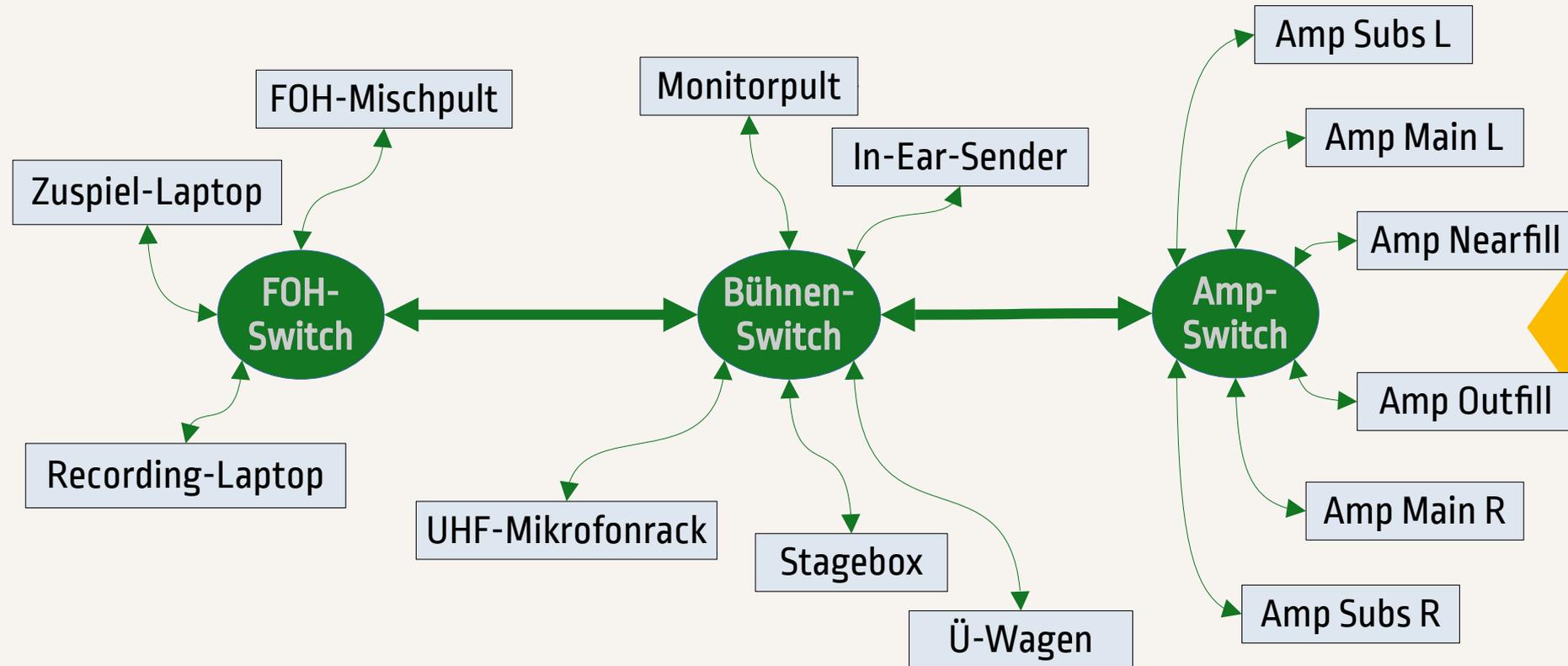
Mehrere Switches können als Kette verbunden sein, oder wiederum als Stern, mit einem Switch-Switch in der Mitte. Beides nennt man einen **Baum**.

Entscheidend für den Baum ist: es gibt **zwischen zwei Geräten** immer nur **genau eine Verbindung**.

# Mehrere Switches: der Baum

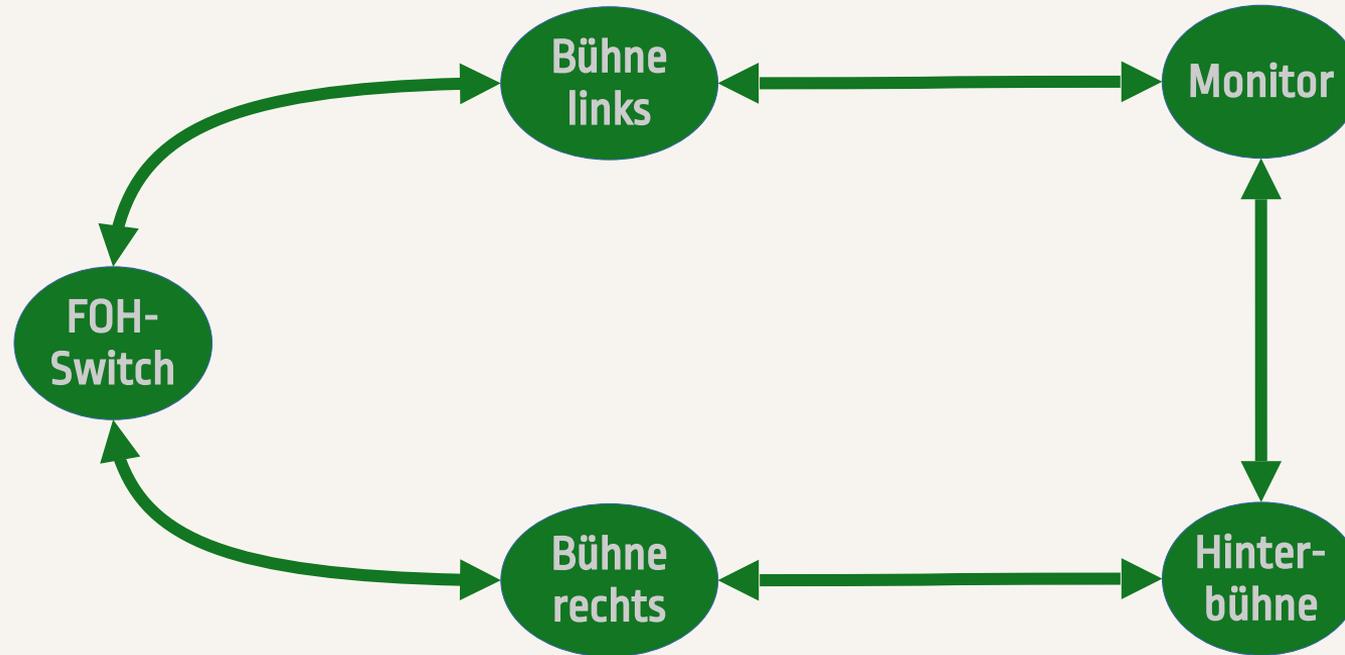


# Mehrere Switches: der Baum



Von jedem Gerät zu jedem anderen gibt es genau einen Weg.  
Natürlich haben wir noch ein redundantes **secondary-Netz**, das genau so aussieht.

# Mehrere Switches: Ringe



Hier geht beim Mathematiker die „Tasse/Autoreifen/Donut“-Lampe an, denn dieses Netz hat genau ein „Loch“. Es gibt jetzt zwischen manchen Geräten zwei Wege.

# Mehrere Switches: Ringe

Ringe sind im Ethernet-Netz verboten.

Der Ethernet-Postbote hat ein Paket vom FOH für das Monitorpult und ruft in die Runde: „wo ist denn hier das Monitorpult?“

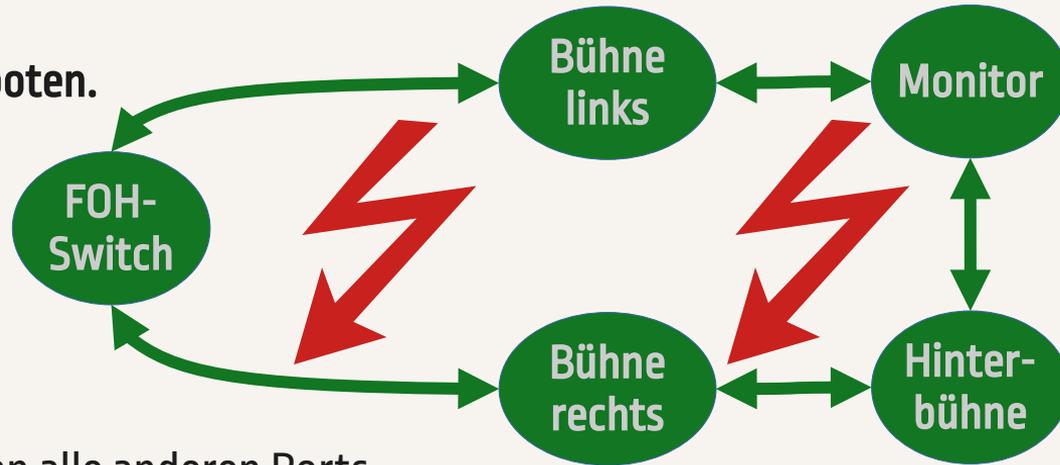
Switch Bühne links ruft „FOH will wissen, wo das Monitorpult ist“ an alle anderen Ports.

Switch Monitor ruft „Bühne links will wissen, wo das Monitorpult ist“ an alle anderen Ports.

Switch Hinterbühne ruft „Monitor will wissen, wo das Monitorpult ist“. Switch Bühne rechts ruft „Hinterbühne will wissen, wo das Monitorpult ist“. Switch FOH ruft „Bühne rechts will wissen, wo das Monitorpult ist. Und so weiter...

Das gleiche auch andersrum, gleichzeitig! Zwischendurch hat das Monitorpult mal ganz leise „hier“ gerufen, aber das geht im Geschrei unter: das Netz bricht zusammen.

Diese Feedback-Schleife nennt man **broadcast storm**.



# Mehrere Switches: Ringe

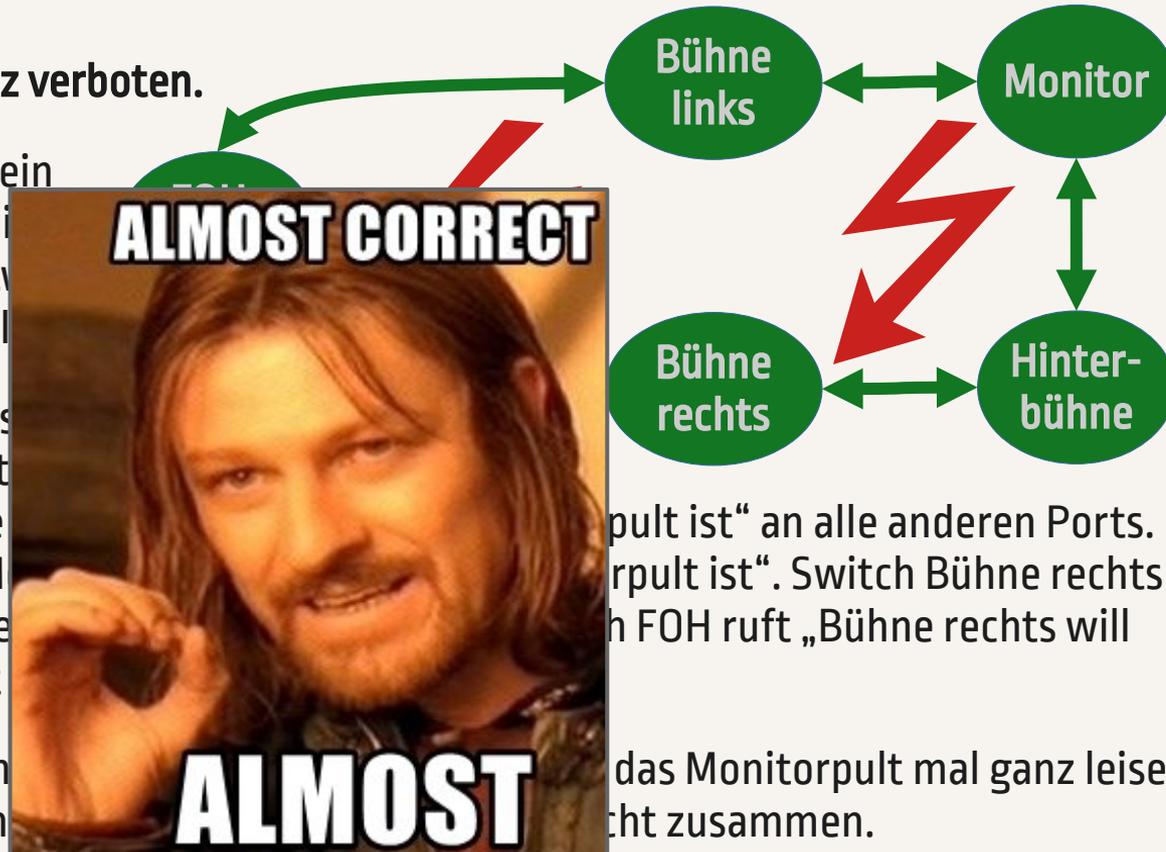
Ringe sind im Ethernet-Netz verboten.

Der Ethernet-Postbote hat ein Paket vom FOH für das Monitorpult und ruft in die Runde: „Wo ist denn hier das Monitorpult“

Switch Bühne links ruft „Postbote, wo das Monitorpult ist“  
Switch Monitor ruft „Bühne links will wissen, wo das Monitorpult ist“  
Switch Hinterbühne ruft „Monitorpult ist“  
Switch Bühne rechts ruft „Hinterbühne will wissen, wo das Monitorpult ist“

Das gleiche auch andersrum  
„hier“ gerufen, aber das geht nicht zusammen.

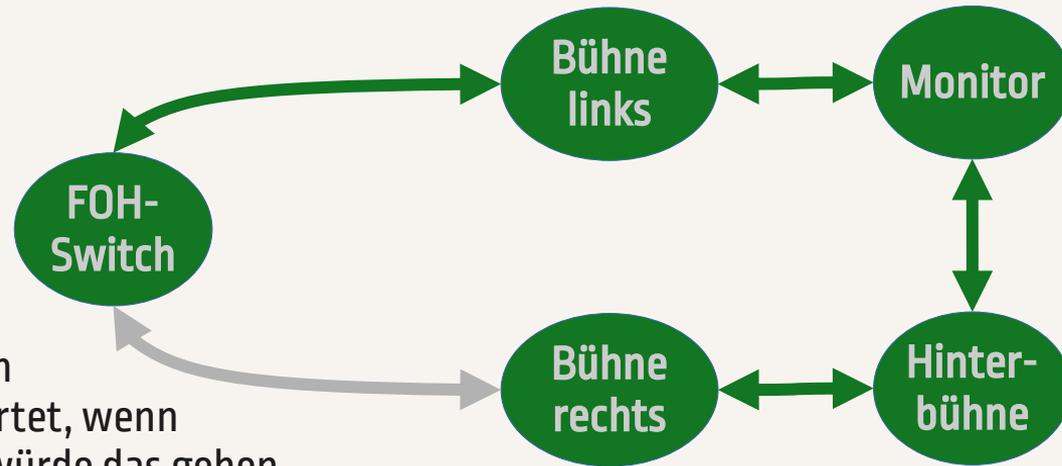
Diese Feedback-Schleife nennt man **broadcast storm**.



# Mehrere Switches: Ringe

Wir würden schon gern Ringe bauen, weil jetzt ein Kabel ausfallen kann, und trotzdem sind noch alle Switches miteinander verbunden.

Wenn die Switches sich absprechen könnten und nur noch einer antwortet, wenn der Ethernet-Postbote ruft, dann würde das gehen...



Gibts natürlich und heißt **(rapid) spanning tree protocol (STP/RTSP)**:

Die Switches checken unter sich die Netztopologie aus (das geht auf Layer 3, also einen höher als Ethernet, daher ohne Broadcast-Storm) und einigen sich, dass die Verbindung zwischen FOH und Bühne rechts inaktiv gemacht wird.

Bühne rechts hält also die Klappe, wenn vom FOH eine Anfrage kommt.

Im Hintergrund testen die Switches permanent die Verbindungen, und wenn eine aktive ausfällt, wird die inaktive reaktiviert. Das dauert zwischen einigen Millisekunden und mehreren 10 Sekunden.

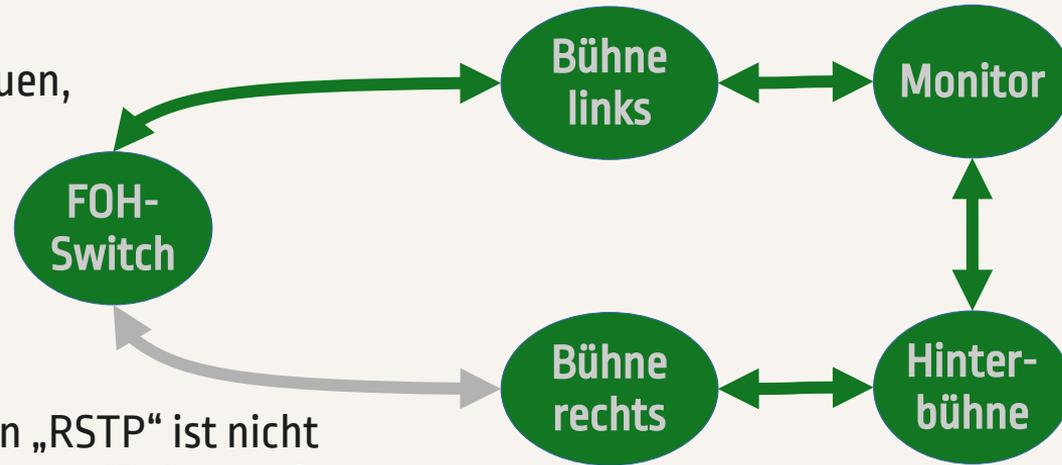
# Mehrere Switches: Ringe



Bevor wir einen Ring bauen, müssen wir sicher sein, dass alle Switches in dem Ring STP verstehen und benutzen, sonst bricht das Netz zusammen!



Auch die schnelle Version „RSTP“ ist nicht schnell genug für Audio-over-IP-Anwendungen. Bis ein neuer spanning tree ausgehandelt ist, haben wir schon Audio-Daten verloren und es gibt einen hörbaren Dropout.



RSTP ist also kein Ersatz für die Dante-Redundanz!

Der Hersteller Luminex verspricht mit seinem RlinkX-Protokoll eine Rekonfiguration innerhalb von 10 ms. Das ist sehr gut, aber auch zu lang für Dante. Es gibt einen kurzen Knackser (von einer Hunderstel Sekunde), also für Rock'n'Roll durchaus ok.

Dante für  
Einsteiger\*innen

Sascha „Mon“ Helmstädt  
s.helmstaedt@gmx.de

Jörn Nettingsmeier  
nettings@luchtbeweging.nl  
<https://luchtbeweging.nl>

# Multicasting in Dante-Netzen

- Einfache Dante-Netze benutzen **Unicast** (jede Quelle sendet Daten an alle Senken, die sie haben wollen). Das bringt die Sender irgendwann an ihre Grenzen, weil Daten mehrfach verschickt werden müssen.
- Alternativ können Quellen nur einen Datenstrom an eine sogenannte **Multicast-Gruppe** senden, und die Switches sorgen dafür, dass der bei allen Geräten ankommt, die ihn „abonniert“ haben.
  - Die Endgeräte senden **IGMP-Messages** (= „Internet Group Management Protocol“) zum Abonnieren eines Datenstroms
  - Die Switches müssen diese Messages abhören (daher **IGMP snooping**, schnüffeln), daraus Abo-Anfragen an die Quellen oder ihre Router generieren (**IGMP querying**) und die Datenströme entsprechend an alle Abonnenten kopieren



Switches, die kein IGMP-Snooping beherrschen, kopieren Multicast-Traffic stumpf auf alle Ports (das kann ein Netzwerk stark belasten)

# Switch-Tricks: VLANs, trunking, QoS

**VLAN:** *virtual LAN*, mehrere logisch voneinander getrennte Netze auf einem physikalischen Netz

**trunking:** Das Verbinden mehrerer Switches mit einem oder mehreren Datenleitungen

**QoS:** *quality of service*, das Priorisieren von Datenpaketen in dringende und weniger dringende. Dante benutzt die Implementierung DiffServ (= „differential services“), das sind Markierungen an den Datenpaketen.