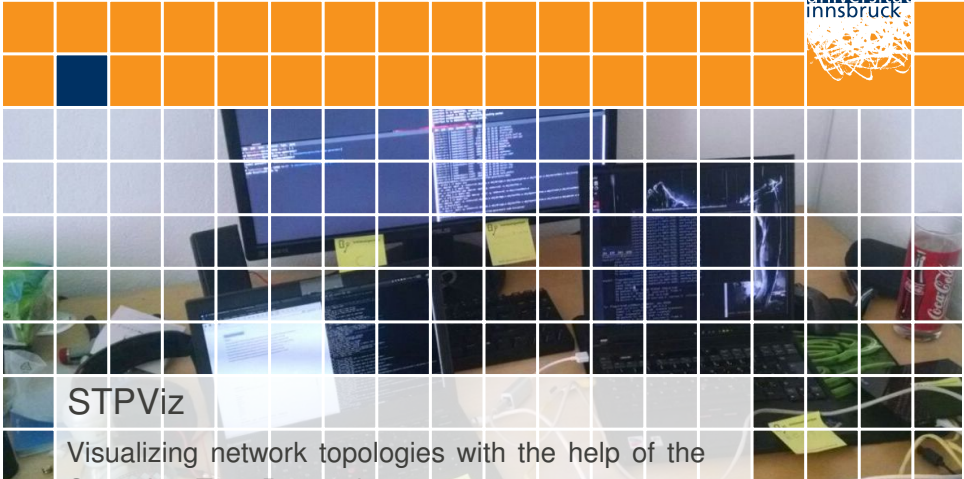


# universität innsbruck



STPViz

Visualizing network topologies with the help of the  
Spanning Tree Protocol

- ▶ **Einleitung & Motivation**
- ▶ **Spanning Tree Protocol (STP)**
- ▶ **STPViz**
- ▶ **Software-Switch**
- ▶ **Tests**
- ▶ **Zusammenfassung & Ausblick**



# Überblick

- ▶ **Einleitung & Motivation**
- ▶ **Spanning Tree Protocol (STP)**
- ▶ **STPViz**
- ▶ **Software-Switch**
- ▶ **Tests**
- ▶ **Zusammenfassung & Ausblick**



# Warum STP?

- ▶ Redundanz in großen Netzwerken erwünscht



# Warum STP?

- ▶ Redundanz in großen Netzwerken erwünscht
- ▶ Schleifen im Netzwerk entstehen

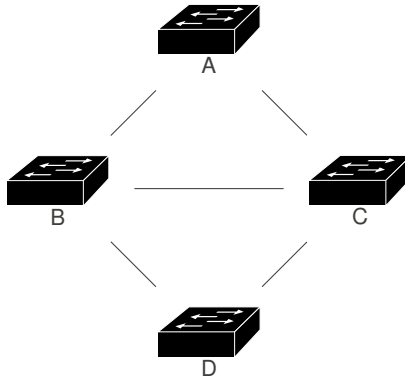


# Warum STP?

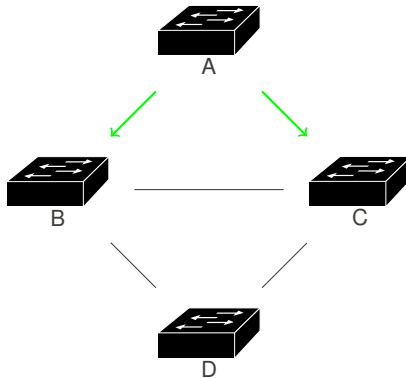
- ▶ Redundanz in großen Netzwerken erwünscht
- ▶ Schleifen im Netzwerk entstehen
- ▶ Das führt zu Broadcast Storms



# Broadcast Storms

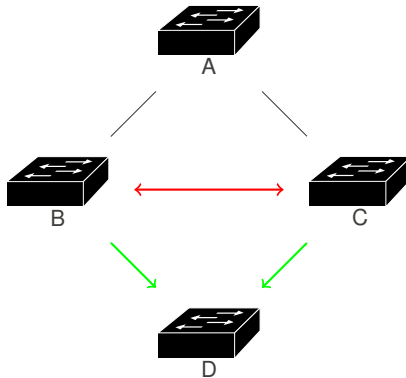


# Broadcast Storms

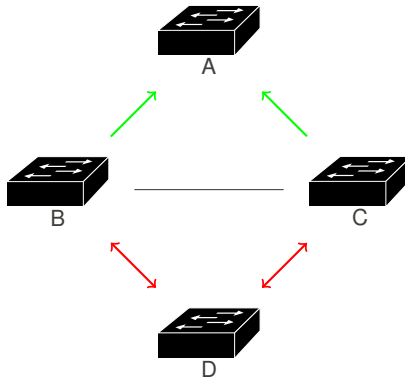




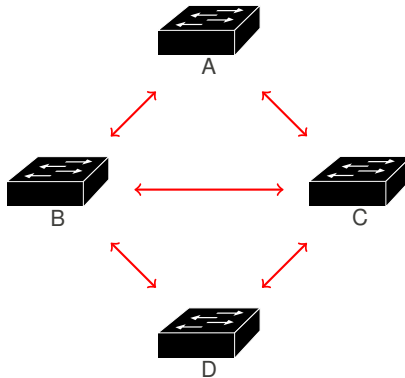
# Broadcast Storms



# Broadcast Storms



# Broadcast Storms



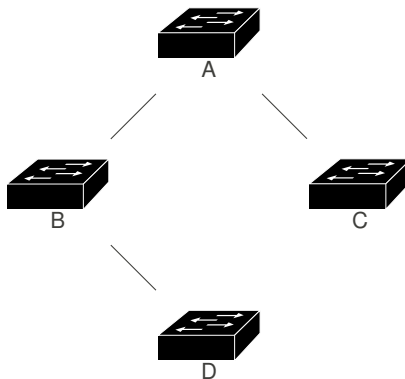
# Broadcasts mit STP

STP deaktiviert doppelte Verbindungen.  
Baumtopologie entsteht



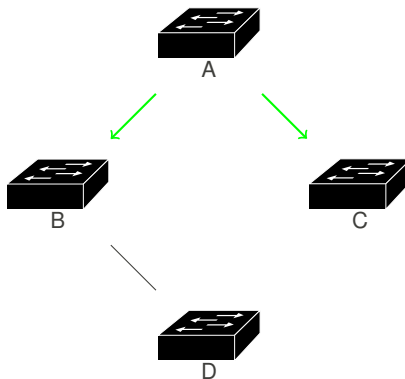
# Broadcasts mit STP

STP deaktiviert doppelte Verbindungen.  
Baumtopologie entsteht



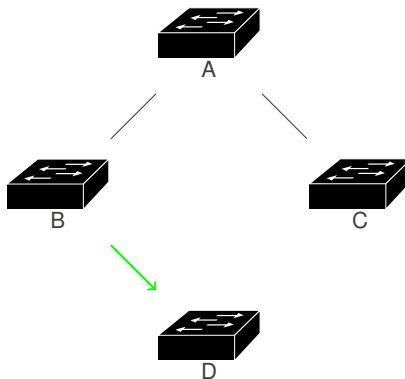
# Broadcasts mit STP

STP deaktiviert doppelte Verbindungen.  
Baumtopologie entsteht



# Broadcasts mit STP

STP deaktiviert doppelte Verbindungen.  
Baumtopologie entsteht



- ▶ Große Netzwerke sind schwer zu administrieren





# Grundidee

- ▶ Große Netzwerke sind schwer zu administrieren
- ▶ STP verbirgt Fehler und Änderungen zusätzlich



# Grundidee

- ▶ Große Netzwerke sind schwer zu administrieren
- ▶ STP verbirgt Fehler und Änderungen zusätzlich
- ▶ STP Konfiguration ist komplex



# Grundidee

- ▶ Große Netzwerke sind schwer zu administrieren
- ▶ STP verbirgt Fehler und Änderungen zusätzlich
- ▶ STP Konfiguration ist komplex



# Grundidee

- ▶ Große Netzwerke sind schwer zu administrieren
- ▶ STP verbirgt Fehler und Änderungen zusätzlich
- ▶ STP Konfiguration ist komplex

Kommerzielle Tools existieren.

Aber: Diese nutzen SNMP



# Was waren unsere Ziele?

- ▶ Nur STP



# Was waren unsere Ziele?

- ▶ Nur STP
- ▶ Geringe verursachte Netzwerkauslastung



# Was waren unsere Ziele?

- ▶ Nur STP
- ▶ Geringe verursachte Netzwerkauslastung
- ▶ Passiv



# Was waren unsere Ziele?

- ▶ Nur STP
- ▶ Geringe verursachte Netzwerkauslastung
- ▶ Passiv
- ▶ Verteilt





# Was waren unsere Ziele?

- ▶ Nur STP
- ▶ Geringe verursachte Netzwerkauslastung
- ▶ Passiv
- ▶ Verteilt
- ▶ Geringe Hardwareauslastung



# Was waren unsere Ziele?

- ▶ Nur STP
- ▶ Geringe verursachte Netzwerkauslastung
- ▶ Passiv
- ▶ Verteilt
- ▶ Geringe Hardwareauslastung
- ▶ Keine Instandhaltung notwendig

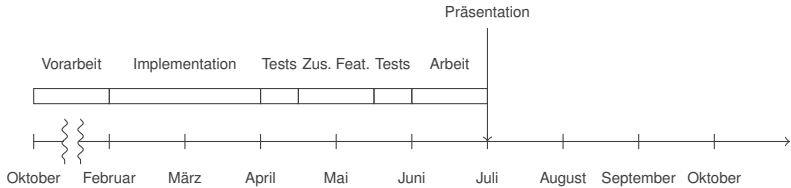


# Was waren unsere Ziele?

- ▶ Nur STP
- ▶ Geringe verursachte Netzwerkauslastung
- ▶ Passiv
- ▶ Verteilt
- ▶ Geringe Hardwareauslastung
- ▶ Keine Instandhaltung notwendig
- ▶ Mehrere Bäume möglich



# Zeitplanung



# Überblick

- ▶ Einleitung & Motivation
- ▶ **Spanning Tree Protocol (STP)**
- ▶ STPViz
- ▶ Software-Switch
- ▶ Tests
- ▶ Zusammenfassung & Ausblick



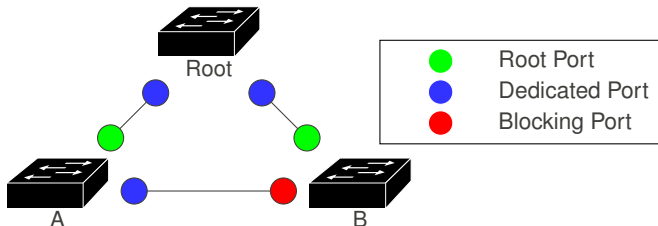
# STP Funktionsweise

Globale Root wird durch Prioritäten bestimmt.  
Nur kürzester Pfad zur Root bleibt erhalten



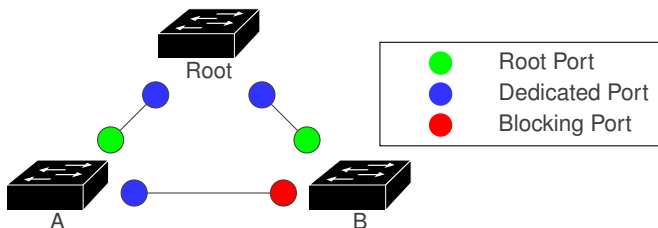
# STP Funktionsweise

Globale Root wird durch Prioritäten bestimmt.  
Nur kürzester Pfad zur Root bleibt erhalten



# STP Funktionsweise

Globale Root wird durch Prioritäten bestimmt.  
Nur kürzester Pfad zur Root bleibt erhalten



Bemerkung: Wir werden Switches Bridges nennen



# STP Pakete

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Protocol Identifier																Version Id								BPDU Type							
Flags								Root Identifier																							
Root Identifier																															
Root Identifier								Root Path Cost																							
Root Path Cost								Bridge Identifier																							
Bridge Identifier																															
Bridge Identifier								Port Identifier																Message Age							
Message Age								Max Age																Hello Time							
Hello Time								Forward Delay																							



# STP Pakete

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Protocol Identifier																Version Id								BPDU Type							
Flags								Root Identifier																							
Root Identifier																															
Root Identifier								Root Path Cost																							
Root Path Cost								Bridge Identifier																							
Bridge Identifier																															
Bridge Identifier								Port Identifier																Message Age							
Message Age								Max Age																Hello Time							
Hello Time								Forward Delay																							



# STP Pakete

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Protocol Identifier																Version Id								BPDU Type							
Flags								Root Identifier																							
Root Identifier																															
Root Identifier								Root Path Cost																							
Root Path Cost								Bridge Identifier																							
Bridge Identifier																															
Bridge Identifier								Port Identifier																Message Age							
Message Age								Max Age																Hello Time							
Hello Time								Forward Delay																							



# STP Pakete

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Protocol Identifier																Version Id								BPDU Type							
Flags								Root Identifier																							
Root Identifier																															
Root Identifier								Root Path Cost																							
Root Path Cost								Bridge Identifier																							
Bridge Identifier																															
Bridge Identifier								Port Identifier																Message Age							
Message Age								Max Age																Hello Time							
Hello Time								Forward Delay																							



# STP Pakete

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Protocol Identifier																Version Id								BPDU Type							
Flags								Root Identifier																							
Root Identifier																															
Root Identifier								Root Path Cost																							
Root Path Cost								Bridge Identifier																							
Bridge Identifier																															
Bridge Identifier								Port Identifier																Message Age							
Message Age								Max Age																Hello Time							
Hello Time								Forward Delay																							



- ▶ Einleitung & Motivation
- ▶ Spanning Tree Protocol (STP)
- ▶ **STPViz**
- ▶ Software-Switch
- ▶ Tests
- ▶ Zusammenfassung & Ausblick



# Struktur & Funktion

- Mehrere Clients senden ihre Daten an einen Server



# Struktur & Funktion

- ▶ Mehrere Clients senden ihre Daten an einen Server
- ▶ Server kombiniert Daten





# Struktur & Funktion

- ▶ Mehrere Clients senden ihre Daten an einen Server
- ▶ Server kombiniert Daten
- ▶ Parser erstellt Visualisierung



# Struktur & Funktion

- ▶ Mehrere Clients senden ihre Daten an einen Server
- ▶ Server kombiniert Daten
- ▶ Parser erstellt Visualisierung



# Struktur & Funktion

- ▶ Mehrere Clients senden ihre Daten an einen Server
- ▶ Server kombiniert Daten
- ▶ Parser erstellt Visualisierung

Client sammelt Daten aus STP Paketen und kombiniert sie zu einem Pfad zur Root

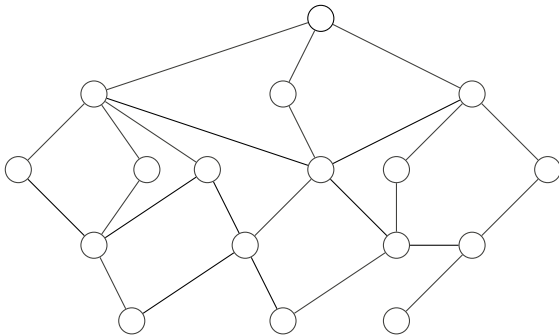


# STP Pakete

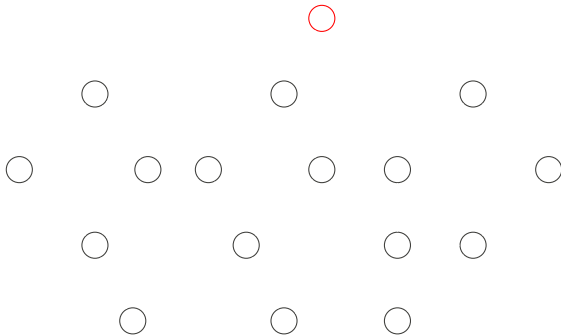
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Protocol Identifier																Version Id								BPDU Type							
Flags								Root Identifier																							
Root Identifier																															
Root Identifier								Root Path Cost																							
Root Path Cost								Bridge Identifier																							
Bridge Identifier																															
Bridge Identifier								Port Identifier																Message Age							
Message Age								Max Age																Hello Time							
Hello Time								Forward Delay																							



# Pfadkonstruktion



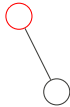
# Pfadkonstruktion



# Pfadkonstruktion

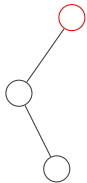


# Pfadkonstruktion

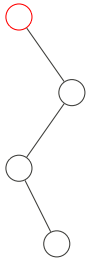




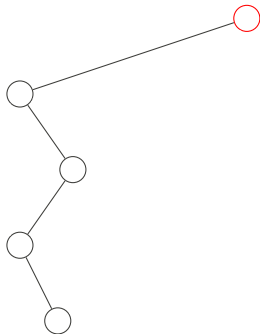
# Pfadkonstruktion



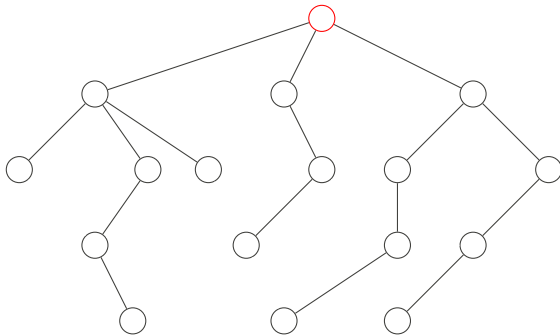
# Pfadkonstruktion



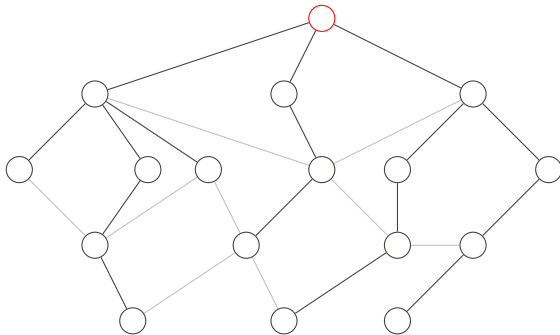
# Pfadkonstruktion



# Pfadkonstruktion



# Pfadkonstruktion



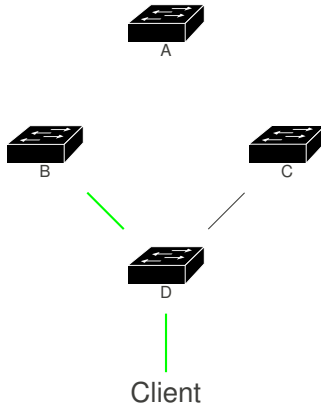
# Problem

Durch diese Konstruktion können Fehler entstehen.  
Diese müssen korrigiert werden



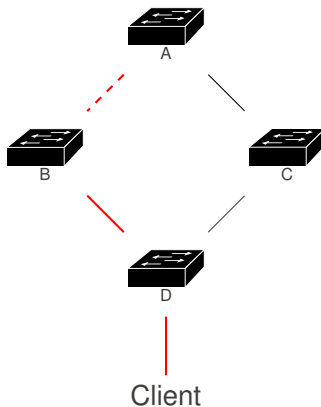
# Problem

Durch diese Konstruktion können Fehler entstehen.  
Diese müssen korrigiert werden



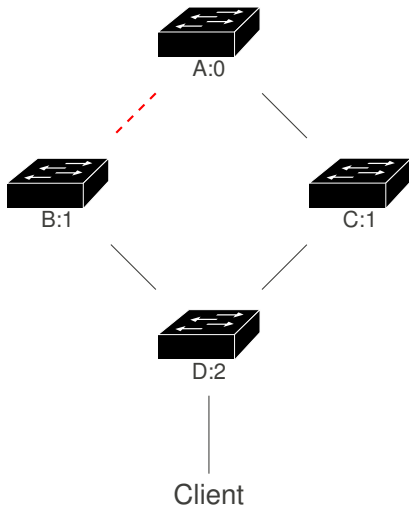
# Problem

Durch diese Konstruktion können Fehler entstehen.  
Diese müssen korrigiert werden

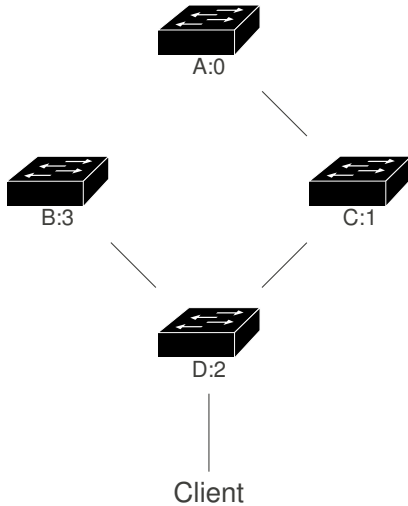




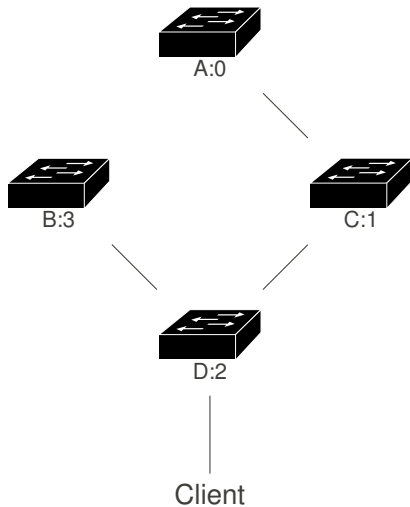
# Fehlerkorrektur



# Fehlerkorrektur



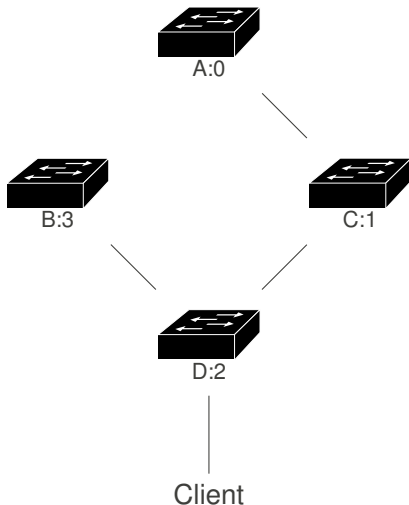
# Fehlerkorrektur



1. Vergleiche alle Bridges mit allen anderen



# Fehlerkorrektur



1. Vergleiche alle Bridges mit allen anderen
2. Entferne Duplikate mit höherer Message Age

Dadurch bleiben alle korrekten Pfade erhalten

# Visualisierung



Root



Root

A, 1

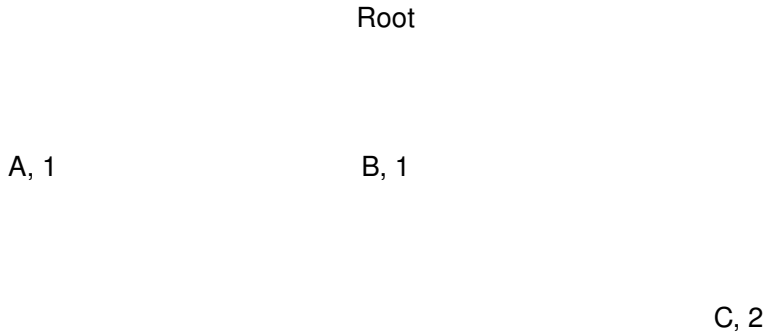


# Visualisierung

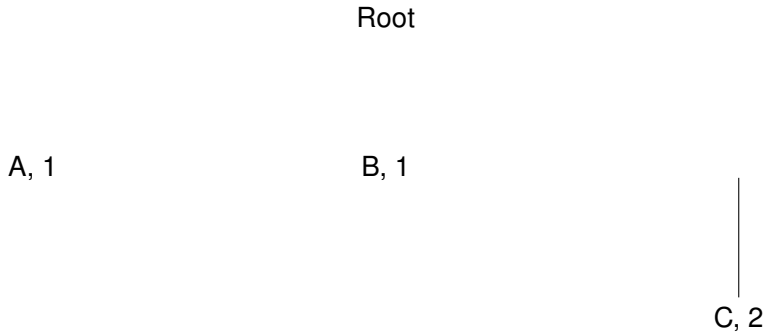




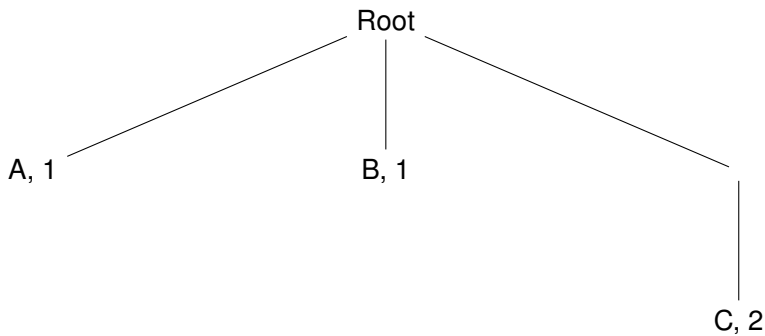
# Visualisierung



# Visualisierung



# Visualisierung



# Testen?



# Überblick

- ▶ Einleitung & Motivation
- ▶ Spanning Tree Protocol (STP)
- ▶ STPViz
- ▶ **Software-Switch**
- ▶ Tests
- ▶ Zusammenfassung & Ausblick



# Software-Switch

- ▶ STP Implementation in OpenWrt und dd-wrt fehlerhaft



# Software-Switch

- ▶ STP Implementation in OpenWrt und dd-wrt fehlerhaft
- ▶ Emuliert STP fähige Bridge



# Software-Switch

- ▶ STP Implementation in OpenWrt und dd-wrt fehlerhaft
- ▶ Emuliert STP fähige Bridge
- ▶ Multithreaded (POSIX Threads)





# Software-Switch

- ▶ STP Implementation in OpenWrt und dd-wrt fehlerhaft
- ▶ Emuliert STP fähige Bridge
- ▶ Multithreaded (POSIX Threads)
- ▶ Probleme mit NetworkManager



# Software-Switch

- ▶ STP Implementation in OpenWrt und dd-wrt fehlerhaft
- ▶ Emuliert STP fähige Bridge
- ▶ Multithreaded (POSIX Threads)
- ▶ Probleme mit NetworkManager
- ▶ Nicht sehr ausgiebig getestet



# Software-Switch

- ▶ STP Implementation in OpenWrt und dd-wrt fehlerhaft
- ▶ Emuliert STP fähige Bridge
- ▶ Multithreaded (POSIX Threads)
- ▶ Probleme mit NetworkManager
- ▶ Nicht sehr ausgiebig getestet



# Software-Switch

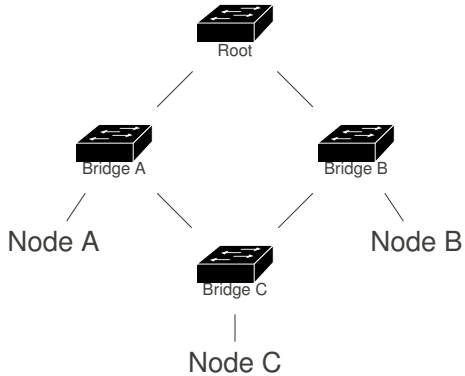
- ▶ STP Implementation in OpenWrt und dd-wrt fehlerhaft
- ▶ Emuliert STP fähige Bridge
- ▶ Multithreaded (POSIX Threads)
- ▶ Probleme mit NetworkManager
- ▶ Nicht sehr ausgiebig getestet



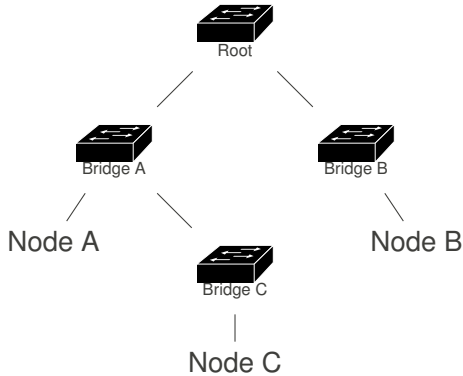
- ▶ **Einleitung & Motivation**
- ▶ **Spanning Tree Protocol (STP)**
- ▶ **STPViz**
- ▶ **Software-Switch**
- ▶ **Tests**
- ▶ **Zusammenfassung & Ausblick**



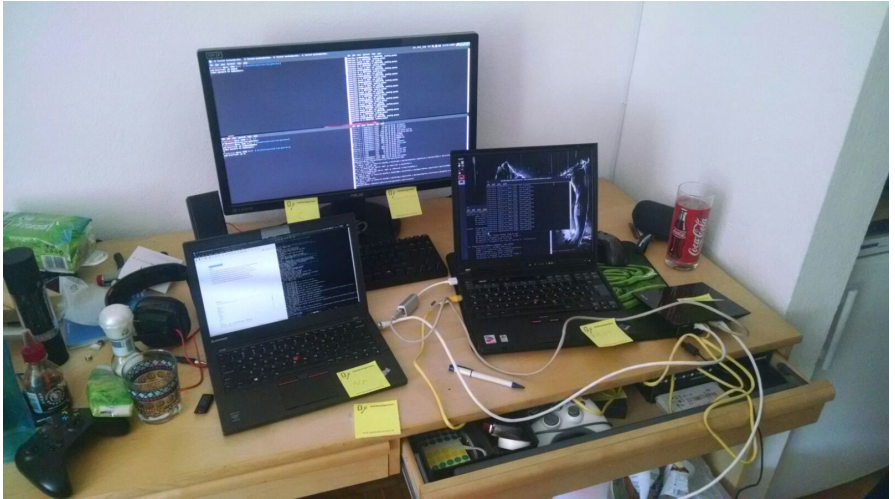
# Setup



# Setup



# Physisches Setup





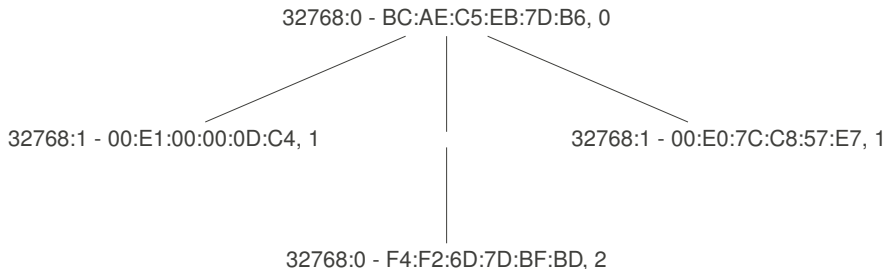
# Plug and Play Test

Verbindung wird hergestellt, bevor STPViz gestartet wird



# Plug and Play Test

Verbindung wird hergestellt, bevor STPViz gestartet wird



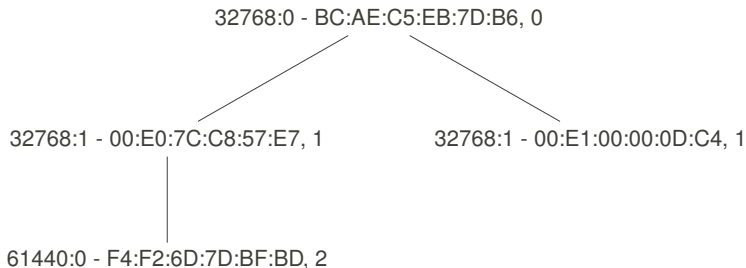
# Tree Establishment Test

STPViz wird gestartet, bevor Verbindung hergestellt wird



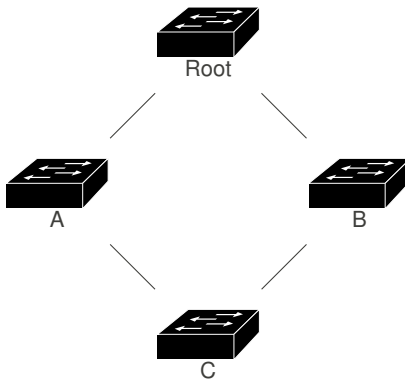
# Tree Establishment Test

STPViz wird gestartet, bevor Verbindung hergestellt wird



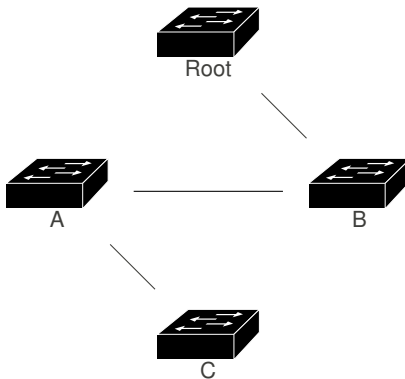
# Dynamische Tests

(Physische) Topologie wird nach der Erkennung verändert



# Dynamische Tests

(Physische) Topologie wird nach der Erkennung verändert



# Dynamische Tests

(Physische) Topologie wird nach der Erkennung verändert

32768:0 - BC:AE:C5:EB:7D:B6, 0



32768:1 - 00:E0:7C:C8:57:E7, 2



61440:0 - F4:F2:6D:7D:BF:BD, 3



# Überblick

- ▶ **Einleitung & Motivation**
- ▶ **Spanning Tree Protocol (STP)**
- ▶ **STPViz**
- ▶ **Software-Switch**
- ▶ **Tests**
- ▶ **Zusammenfassung & Ausblick**





## Ergebnisse & Limitationen

## STPViz Fähigkeiten:



# Ergebnisse & Limitationen

STPViz Fähigkeiten:

1. Bridges und ihre Entfernung zur Root werden erkannt



# Ergebnisse & Limitationen

STPViz Fähigkeiten:

1. Bridges und ihre Entfernung zur Root werden erkannt
2. Pfade zur Root werden angenommen



# Ergebnisse & Limitationen

## STPViz Fähigkeiten:

1. Bridges und ihre Entfernung zur Root werden erkannt
2. Pfade zur Root werden angenommen
3. Fehler werden entfernt



# Ergebnisse & Limitationen

## STPViz Fähigkeiten:

1. Bridges und ihre Entfernung zur Root werden erkannt
2. Pfade zur Root werden angenommen
3. Fehler werden entfernt
4. Ergebnisse werden sinnvoll visualisiert



# Ergebnisse & Limitationen

## STPViz Fähigkeiten:

1. Bridges und ihre Entfernung zur Root werden erkannt
2. Pfade zur Root werden angenommen
3. Fehler werden entfernt
4. Ergebnisse werden sinnvoll visualisiert

## STPViz Limitationen:



# Ergebnisse & Limitationen

## STPViz Fähigkeiten:

1. Bridges und ihre Entfernung zur Root werden erkannt
2. Pfade zur Root werden angenommen
3. Fehler werden entfernt
4. Ergebnisse werden sinnvoll visualisiert

## STPViz Limitationen:

1. STP Erweiterungen werden nicht genutzt



# Ergebnisse & Limitationen

## STPViz Fähigkeiten:

1. Bridges und ihre Entfernung zur Root werden erkannt
2. Pfade zur Root werden angenommen
3. Fehler werden entfernt
4. Ergebnisse werden sinnvoll visualisiert

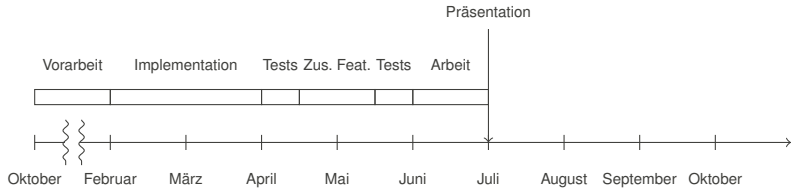
## STPViz Limitationen:

1. STP Erweiterungen werden nicht genutzt
2. Mehr Information als notwendig wird verworfen

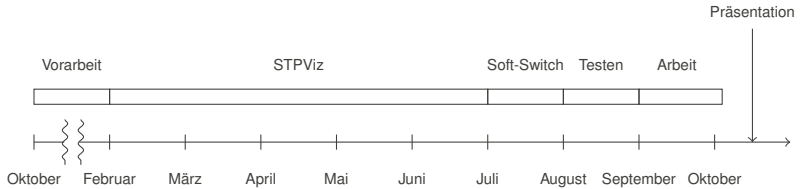
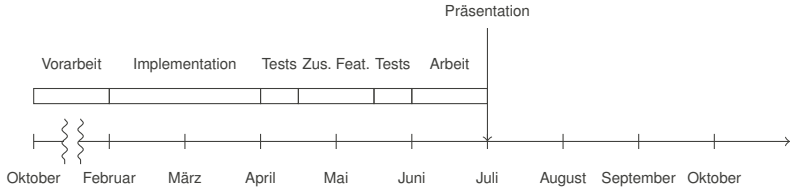




# Zeitplanung



# Zeitplanung



Sinnvolle nächste Schritte:



Sinnvolle nächste Schritte:

1. Features von STP Erweiterungen nutzen



Sinnvolle nächste Schritte:

1. Features von STP Erweiterungen nutzen
2. Minimale Information verwerfen



# Ausblick

Sinnvolle nächste Schritte:

1. Features von STP Erweiterungen nutzen
2. Minimale Information verwerfen
3. Kabeltypen bestimmen



# Ausblick

Sinnvolle nächste Schritte:

1. Features von STP Erweiterungen nutzen
2. Minimale Information verwerfen
3. Kabeltypen bestimmen
4. Topologien direkt aus *.pcapng* Dateien erstellen



Sinnvolle nächste Schritte:

1. Features von STP Erweiterungen nutzen
2. Minimale Information verwerfen
3. Kabeltypen bestimmen
4. Topologien direkt aus *.pcapng* Dateien erstellen
5. STP Implementation in OpenWrt und dd-wrt verbessern





Sinnvolle nächste Schritte:

1. Features von STP Erweiterungen nutzen
2. Minimale Information verwerfen
3. Kabeltypen bestimmen
4. Topologien direkt aus *.pcapng* Dateien erstellen
5. STP Implementation in OpenWrt und dd-wrt verbessern
6. Software-Switch voll funktionstüchtig machen



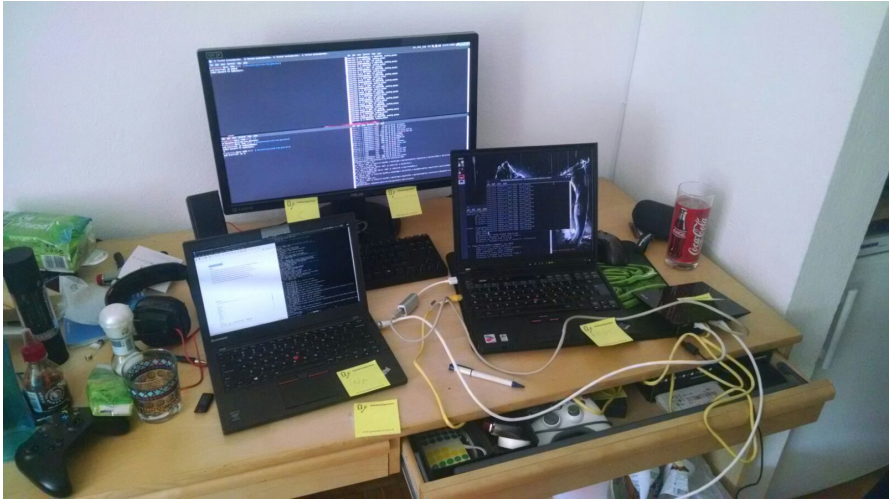
# Konklusion

Was wurde erreicht?



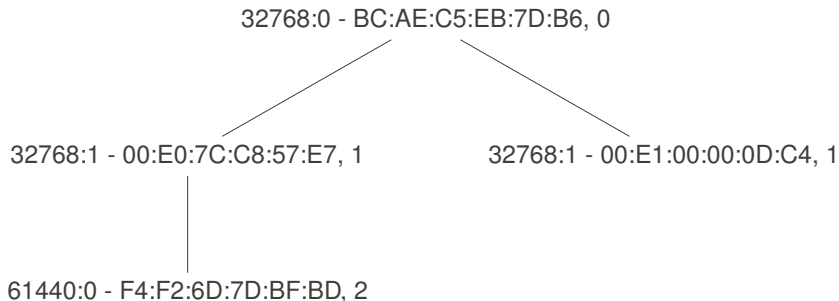
# Konklusion

Was wurde erreicht?



# Konklusion

Was wurde erreicht?



# Konklusion

In Worten:



# Konklusion

In Worten:

- ▶ Graphenkonstruktion nur mit lokaler Information ist schwer



# Konklusion

In Worten:

- ▶ Graphenkonstruktion nur mit lokaler Information ist schwer
- ▶ Besonders, da STP keine Information über Nachbarn überträgt



# Konklusion

In Worten:

- ▶ Graphenkonstruktion nur mit lokaler Information ist schwer
- ▶ Besonders, da STP keine Information über Nachbarn überträgt
- ▶ STPViz schafft dies mit Annahmen





# Konklusion

In Worten:

- ▶ Graphenkonstruktion nur mit lokaler Information ist schwer
- ▶ Besonders, da STP keine Information über Nachbarn überträgt
- ▶ STPViz schafft dies mit Annahmen
- ▶ Billige Fehlerkorrektur mit  $\mathcal{O}(n^2)$  Komplexität



# Konklusion

In Worten:

- ▶ Graphenkonstruktion nur mit lokaler Information ist schwer
- ▶ Besonders, da STP keine Information über Nachbarn überträgt
- ▶ STPViz schafft dies mit Annahmen
- ▶ Billige Fehlerkorrektur mit  $\mathcal{O}(n^2)$  Komplexität
- ▶ Weniger mächtig als kommerzielle Tools, setzt aber weniger voraus



# Konklusion

In Worten:

- ▶ Graphenkonstruktion nur mit lokaler Information ist schwer
- ▶ Besonders, da STP keine Information über Nachbarn überträgt
- ▶ STPViz schafft dies mit Annahmen
- ▶ Billige Fehlerkorrektur mit  $\mathcal{O}(n^2)$  Komplexität
- ▶ Weniger mächtig als kommerzielle Tools, setzt aber weniger voraus

STPViz bietet eine Grundlage für zukünftige mächtigere Tools



# Code

Code und Arbeit sind verfügbar unter:

<https://github.com/alxshine/stp-tree-generator>

<https://github.com/alxshine/software-switch>

