



Nhat-Nam Le

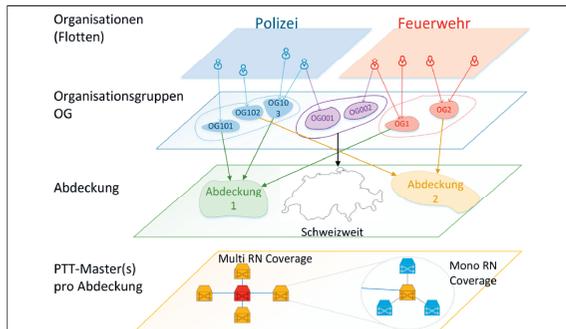


Daniela Peter

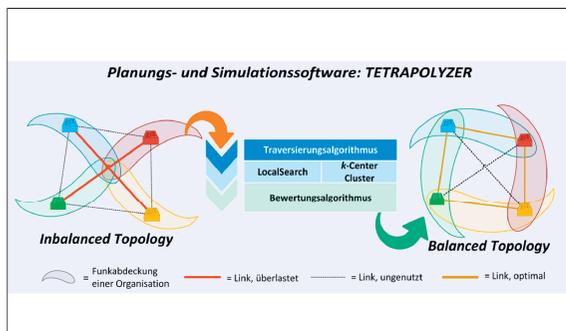
Diplomanden	Nhat-Nam Le, Daniela Peter
Examinator	Prof. Dr. Andreas Rinkel
Experte	Dr. Andreas Jarosch, Swisscom Innovation AG, Bern, BE
Themengebiet	Software
Projektpartner	Bundesamt für Bevölkerungsschutz BABS, Bern, BE

Dynamische Optimierung des Polycom-Backbone-Netzes

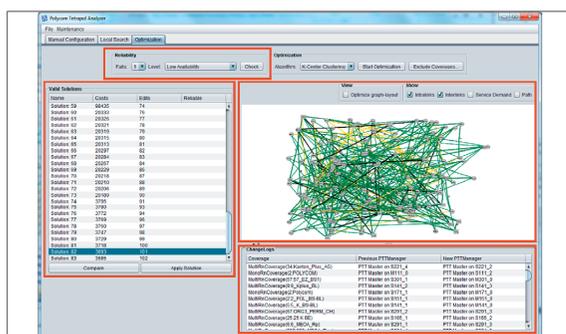
Durch Simulation zu mehr Stabilität



Überblick der POLYCOM-Ebenen



Ablauf einer automatischen Optimierung



Programmsicht im Tetrapolyzer

Ausgangslage: Die Kommunikation zwischen den Blaulichtorganisationen verläuft schweizweit über das Funknetz POLYCOM. Diese wird von Leitungsausfällen im POLYCOM-Backbone-Netz gestört. Eine manuelle Optimierung an den Problemstellen stellt für den verantwortlichen Netzwerkadministrator eine grosse Belastung dar. Diese Belastung zeigt sich in Form eines extrem zeitintensiven Konfigurationsprozesses, verbunden mit fehleranfälligen Lastberechnungen. Die Planungs- und Simulationssoftware «Tetrapolyzer» entlastet den Netzwerkadministrator erheblich, etwa durch eine automatische und effiziente Lastberechnung. Problembereiche werden optisch dargestellt und können mithilfe von automatischen Optimierungsalgorithmen per Knopfdruck ausgeglichen werden. In der vorliegenden Arbeit «Dynamische Optimierung des POLYCOM-Backbone-Netzes» werden diese automatischen Optimierungsalgorithmen erweitert. Der Netzwerkadministrator soll dynamisch auf die Optimierung Einfluss nehmen können. Die dynamische Optimierung gestaltet sich somit realitätsgetreu und vereinfacht eventuelle Backtracking-Arbeiten.

Vorgehen/Technologien: Vor der Umsetzung der dynamischen Optimierung muss die momentane Softwarearchitektur analysiert werden. Die Herausforderung liegt darin, die vorhandenen Strukturen und Funktionen beizubehalten und gleichzeitig die Features der dynamischen Optimierung einzugliedern. Grössere Refactorings und Anpassungen in der Softwarearchitektur stützen sich auf bewährte Design Patterns wie etwa das MVC- oder das Layer-Pattern. Die Implementierungen sind in Java/Java Swing geschrieben. Getestet wird automatisch mit JUnit-Tests über Jenkins.

Ergebnis: Entstanden ist eine funktionsfähige und realitätsgetreue Erweiterung des Tetrapolyzers. Der Netzwerkadministrator erhält nun einen transparenten Einblick in den Optimierungsverlauf des Tetrapolyzers. Die Anzeige aus der dynamischen Optimierung ist so weit angepasst, dass nun die Möglichkeit besteht, mehrere Lösungsvarianten gleichzeitig anzuzeigen und miteinander zu vergleichen. Als Ergebniskontrolle dient ein Ablaufszenario. In diesem Szenario werden alle neu implementierten Funktionen für einen möglichen Use Case des BABS angewendet.