



MCastor**2.0**

Customer Requirements Specification

MultiCastor 2.0

V 1.0

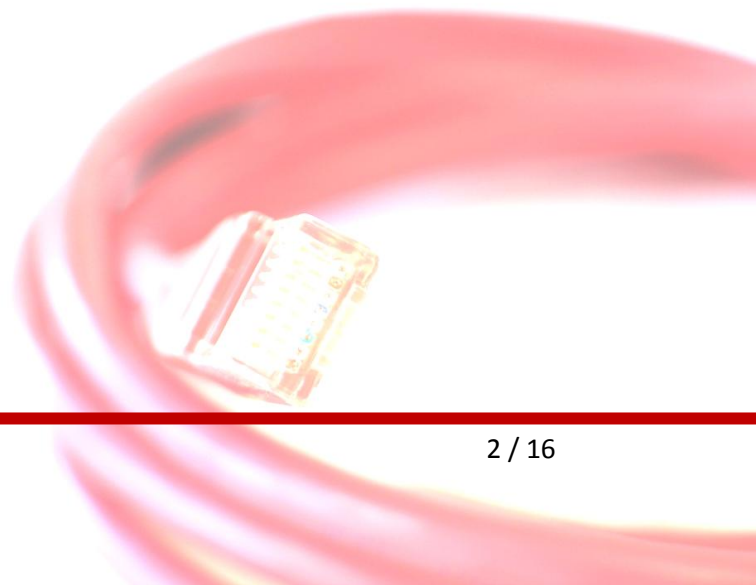
Projekt: MultiCastor 2.0

Auftraggeber: Rentschler & Stuckert
Rotebühlplatz 41/1
70178 Stuttgart

Auftragnehmer: TIT10AID - Team 4 - MCastor2.0

Fabian Fäßler
Filip Haase
Matthis Hauschild
Sebastian Koralewski
Jonas Traub
Christopher Westphal

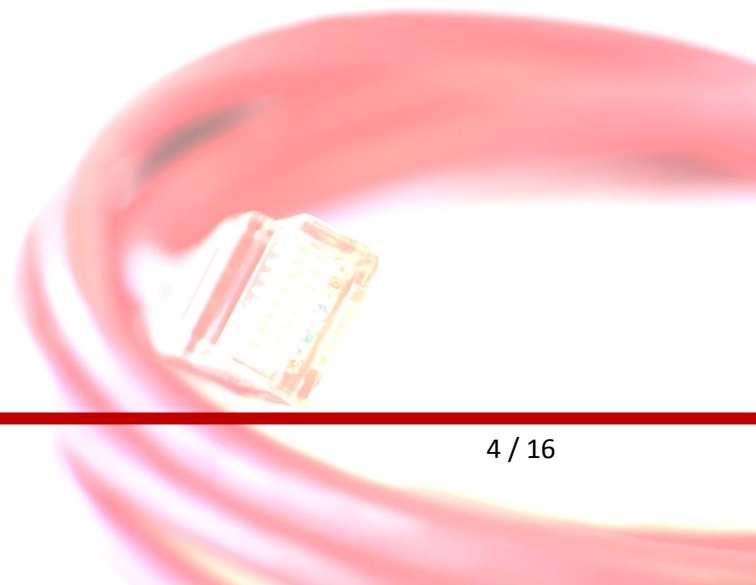
Rotebühlplatz 41 – Raum 0.10
70178 Stuttgart



Inhaltsverzeichnis

1. Zielbestimmung	5
2. Produkteinsatz	6
2.1. Einsatzgebiet	6
2.2. Senden und Empfangen von Multicast-Daten	6
2.3. Registrierung von Multicast-Pfaden	7
2.4. Reservierung von Multicast-Pfaden	8
2.5. Zusammenfassung	8
3. Produktfunktionen	9
3.1. Use Cases	9
3.1.1. /LUC10/ Send Multicast	9
3.1.2. /LUC20/ Receive Multicast	9
3.1.3. /LUC30/ Analyse Multicast	9
3.1.4. /LUC40/ Configure Settings	9
3.1.5. /LUC41/ Save Configuration File	10
3.1.6. /LUC42/ Load Configuration File	10
3.1.7. /LUC43/ Set/Delete Multicast Group Membership	10
3.1.8. /LUC44/ Activate/Deactivate Multicast Group	10
3.1.9. /LUC45/ Configure Server Settings	10
3.1.10. /LUC46/ Configure Language Settings	10
3.1.11. /LUC50/ Multi-Instance-Ability	10
3.1.12. /LUC60/ Functional Testing	10
3.2. Anforderungen	11
3.3. Bekannte Bugs	13
3.3.1. /BUG 10/ Aboutfenster schließen	13
3.3.2. /BUG 20/ Active/Inactive Button Fehlfunktion	13
3.3.3. /BUG 30/ GUI Option Menu Schrift	13
3.3.4. /BUG 40/ Logfile	13
3.3.5. /BUG 50/ Partielles Laden von Konfigurationen	13
4. Produktdaten	14
4.1. /LD10/ Das Logfile	14
4.2. /LD20/ Das Programm-Configurationsfile	14

4.3.	/LD30/ Das Predefined-Value-Configurationfile	14
4.3.1.	/LD31/ Multicastor.xml	14
4.3.2.	/LD32/ Inhalt des Configurationsfiles.....	14
5.	Produktcharakteristiken.....	15
5.1.	Systemumgebung	15
5.1.1.	Hardwareumgebung.....	15
5.1.2.	Softwareumgebung	15
	Dokumentversionen	16



1. Zielbestimmung

Die bestehende Open-Source-Software „MultiCastor 1.0“ ist ein Tool zum Testen der Multicast-Funktionalitäten von Netzwerkgeräten. Es soll mit neuen Features und Verbesserungen der Usability zu „**MultiCastor 2.0**“ weiterentwickelt werden.

Insbesondere ist die Erweiterung des Tools um das Protokoll nach IEEE 802.1ak (MMRP) vorgesehen. Die Multicastingregistrierung in den Netzwerk-Switchen erfolgt dann entweder über IGMP (IPv4), MLD (IPv6) oder MMRP (Layer 2). Während die Protokolle MLD und IGMP bereits im Windows und Linux Stack vorhanden sind, muss das MMRP Protokoll vollständig im MultiCastor 2.0 integriert werden.

Für die Anwendung des Tools zu Testautomatisierungszwecken soll außerdem die Integration in das Automatisierungsframework STAF/STAX ermöglicht und nachgewiesen werden. Zu diesem Zweck erfolgt die Kommunikation von STAF/STAX über das Command Line Interface, die Ethernet Schnittstelle und das XML-Configfile mit dem Multicastor 2.0.

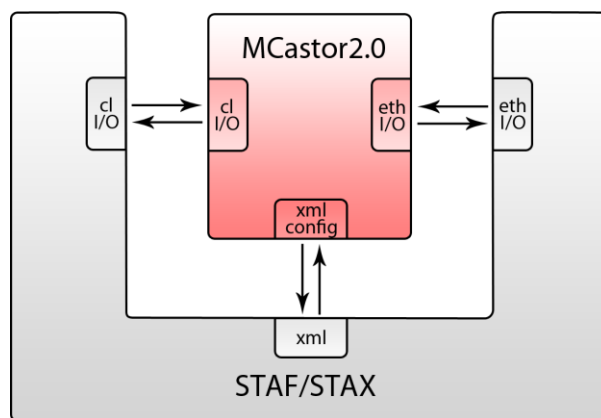


Abbildung 1: Kommunikationskanäle zwischen STAF/STAX und MultiCastor 2.0

Darüber hinaus soll ein (teil)automatisiertes Regressionstestkonzept auf Black-Box-Ebene für die Software selbst entworfen werden.

Die erzielten Ergebnisse sollen anschließend in das zugehörige Open-Source-Projekt einfließen.

2. Produkteinsatz

In diesem Abschnitt werden das Einsatzgebiet sowie die Grundfunktionen des MultiCastors 2.0 dargestellt.

2.1. Einsatzgebiet

Das Tool MultiCastor 1.0 wurde entwickelt um Schwachstellen, wie den Verlust von Paketen oder Netzwerküberlastungen, in multicastfähigen Netzwerken zu entdecken. Dies funktioniert in kleinen privaten Netzwerken bis hin zu großen Unternehmensnetzwerken. Haupterweiterung in Version 2.0 wird die Integration des Layer-2 Multiple MAC Registration Protokolls (MMRP) sein.

2.2. Senden und Empfangen von Multicast-Daten

Multicast erlaubt einem Sender Informationen an multiple Empfänger zu übertragen, ohne das sich dabei die genutzte Bandbreite des Senders mit der Anzahl der Empfänger erhöht. Der Sender registriert eine Multicast-Gruppe mit einer Multicast-Adresse beim Switch. Empfänger können sich nun bei einer Gruppe anmelden. Zum Übertragen einer Nachricht schickt der Sender diese nun an den Switch, welcher die Nachricht an alle angemeldeten Empfänger der Multicast-Gruppe verteilt. Zudem ist eine Kommunikation über die Loop Back Funktion auch ohne externen Switch möglich.

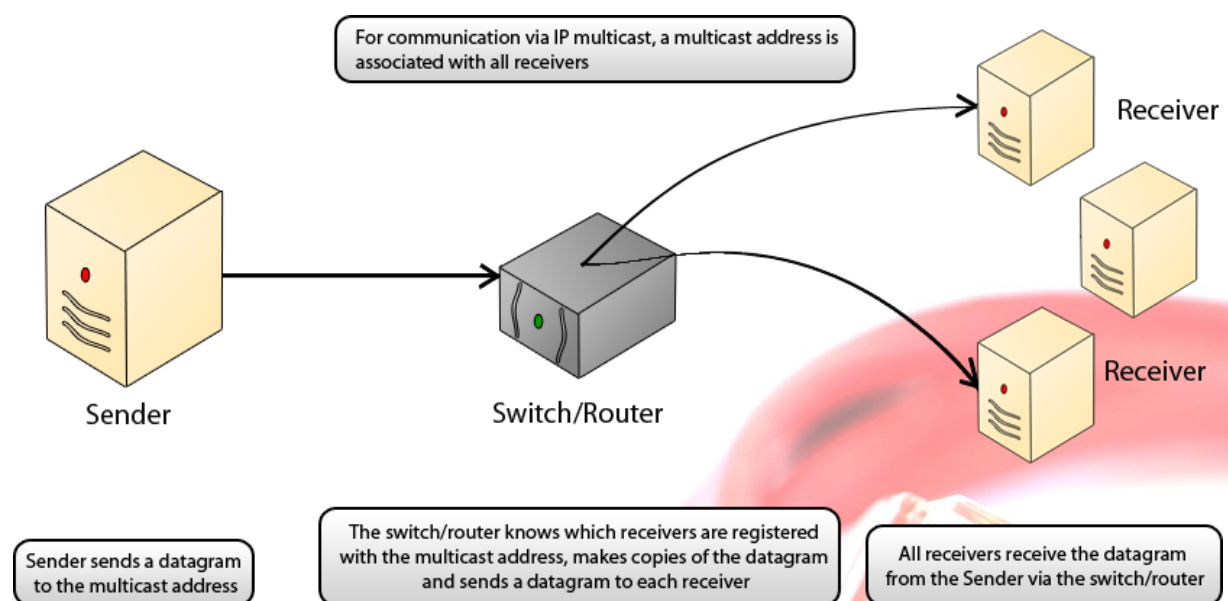


Abbildung 2: Senden und Empfangen von Multicast-Daten

2.3. Registrierung von Multicast-Pfaden

Damit ein Empfänger Nachrichten einer Multicast-Gruppe zugestellt bekommt, muss er sich zuvor am Switch für diese Gruppe registrieren. Der Switch speichert nun die IP- bzw. MAC-Adresse des Empfängers und vermerkt diese als zu der Multicast-Gruppe zugehörig.

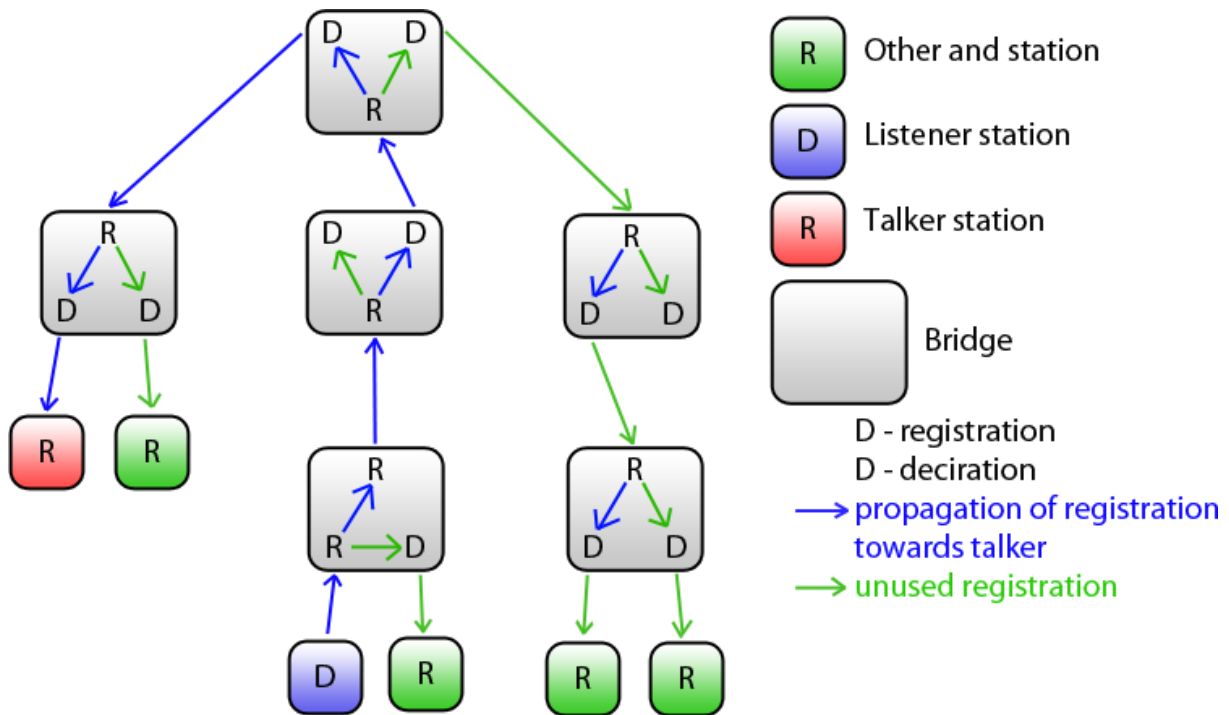


Abbildung 3: Registrierung von Multicast-Pfaden

2.4. Reservierung von Multicast-Pfaden

Der Sender ist über die Anzahl der Empfänger nicht informiert. Da der Switch die Verteilung der Nachrichten übernimmt, spielt dies jedoch für den Sender keine Rolle. Der Switch weiß, welcher Pfad für welchen Empfänger reserviert ist und schickt dementsprechend die Nachrichten an die interessierten Empfänger.

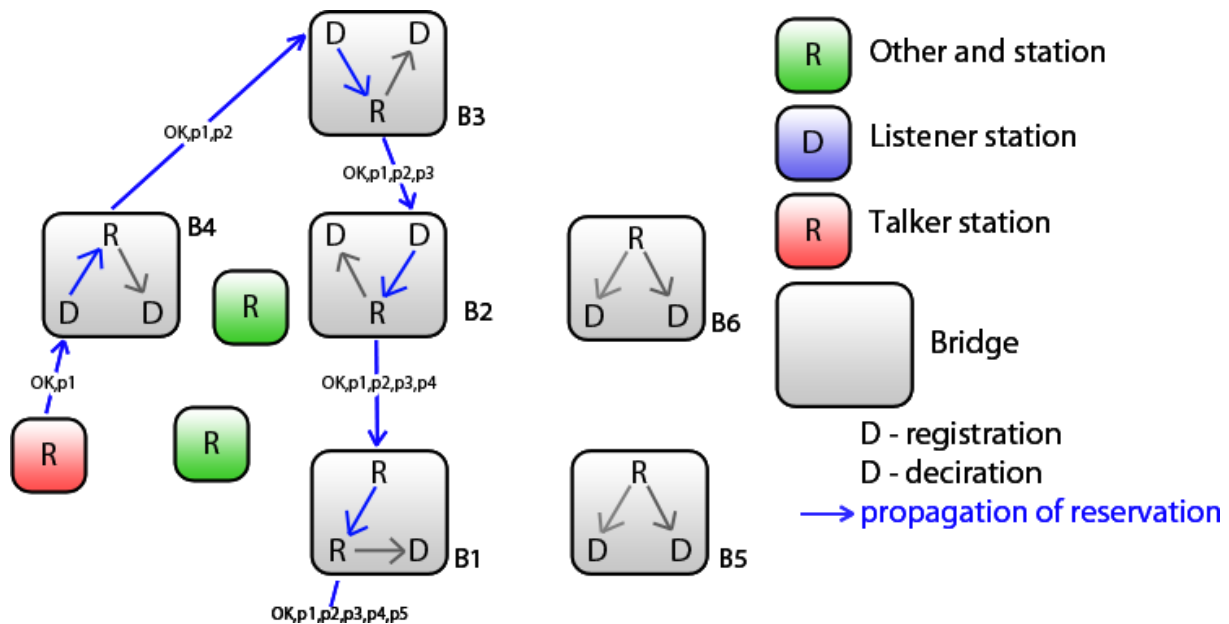


Abbildung 4: Reservierung von Multicast-Pfaden

2.5. Zusammenfassung

Der MultiCastor 1.0 ist im Stande Multicastverbindungen auf Layer-3 herzustellen. Für IPv4 verwendet er das IGM-Protokoll und für IPv6 das MLD-Protokoll. Der MultiCastor 2.0 soll zusätzlich Multicastverbindungen auf Layer-2 herstellen können. Dazu soll das MMR-Protokoll implementiert werden.

3. Produktfunktionen

3.1. Use Cases

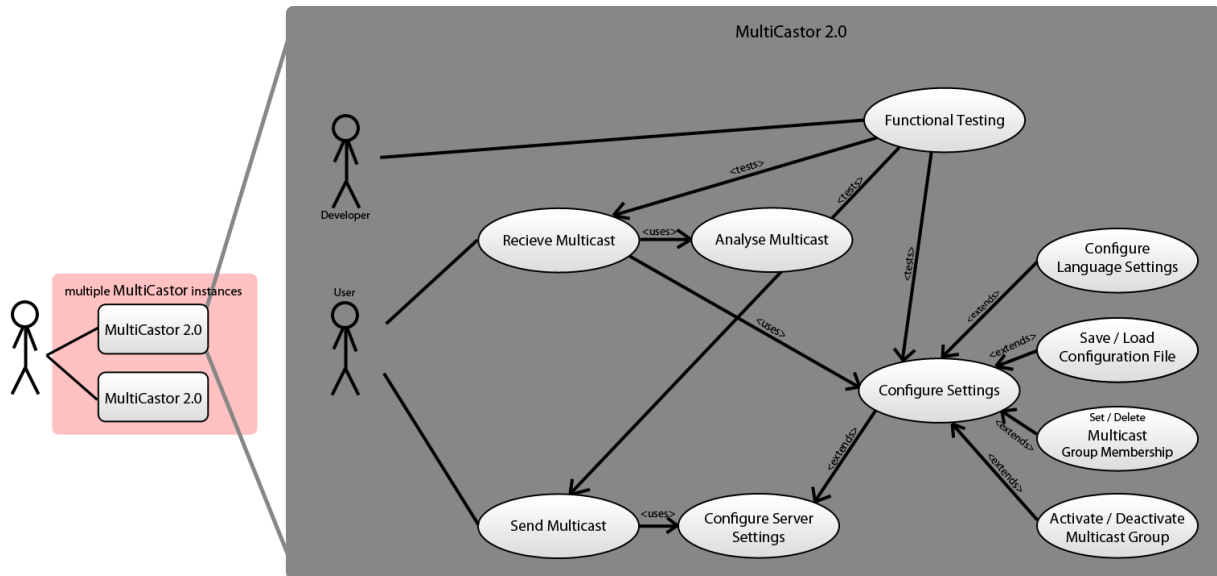


Abbildung 5: Use-Case-Diagramm

Beschreibung der Use-Cases aus Abbildung 4:

3.1.1. /LUC10/ Send Multicast

Das Programm soll, an verschiedene Multicast-Gruppen über wahlweise einen mit Layer-3-Protokoll (IGMP/MLD) oder einen Layer-2-Protokoll (MMRP) reservierten Netzwerkpfad Daten senden. Die gesendeten Pakete sollen auf der Empfängerseite identifizierbar sein.

3.1.2. /LUC20/ Receive Multicast

Das Programm soll einzelne Multicastströme aus zuvor definierten Multicast-Gruppen empfangen können. Wahlweise sollen diese Ströme über einen durch Layer-3 Protokoll (IGMP/MLD) oder ein durch Layer-2 Protokoll (MMRP) reservierten Netzwerkpfad übertragen werden. Anschließend müssen die Daten identifiziert und analysiert werden können.

3.1.3. /LUC30/ Analyse Multicast

Diese Funktion wird von Receive Multicast genutzt. Dadurch ist das Programm in der Lage, die empfangenen Daten zu analysieren und die einzelnen Datenströme zu identifizieren. Außerdem soll nach bestimmten Kriterien wie Zeitintervall zwischen den Daten und verlorenen Daten analysiert werden können. Genauere Informationen finden sich im Kapitel 3.2 Anforderungen (Seite 11).

3.1.4. /LUC40/ Configure Settings

Der Multicaster 2.0 soll ein System bereit stellen über das eine Reihe von Einstellungen, mit Hilfe von Configuration Files, vorgenommen werden können. Diese Files können sowohl über die GUI, einen Texteditor als auch die Kommandozeile konfiguriert werden. Sie beinhalten, getrennt, die Programmsprachen, Standartwerte für Formulare, gespeicherte Sender- und Receiverdaten sowie sonstige Einstellungsmöglichkeiten.

3.1.5. /LUC41/ Save Configuration File

Das Programm soll Einstellungen, die während der Benutzung des Programmes vorgenommen werden, in Konfigurationsfiles speichern. Dies geschieht modular, damit sowohl einzelne Sender/Receiver als auch mehrere Sender/Receiver gleichzeitig gespeichert werden können.

3.1.6. /LUC42/ Load Configuration File

Das Programm soll, bestehende Konfigurationsfiles laden können. Dies geschieht modular, damit sowohl einzelne Sender/Receiver als auch mehrere Sender/Receiver gleichzeitig geladen werden können (inkrementelles Laden).

3.1.7. /LUC43/ Set/Delete Multicast Group Membership

Diese Funktion soll es dem Benutzer gestatten einer Multicast-Gruppe beizutreten als auch die Gruppe wieder zu verlassen. Für den Server bedeutet das Setzen oder Entfernen der Mitgliedschaft zu einer Multicast-Gruppe, dass er an diese Gruppe sendet oder nicht.

3.1.8. /LUC44/ Activate/Deactivate Multicast Group

Bei dieser Einstellung sollen Multicast-Gruppen deaktiviert werden können, ohne sie zu löschen. So muss man eine Gruppe nicht neu hinzufügen sobald wieder Daten von ihr empfangen oder an sie gesendet werden sollen.

3.1.9. /LUC45/ Configure Server Settings

Der MultiCastor 2.0 soll es dem Server ermöglichen zwischen verschiedenen Sendarten zu unterscheiden. Dies wird in den Server Settings gespeichert. Dabei kann der Nutzer zwischen einer so genannten Low-Time-Resolution und einer High-Time-Resolution wählen. Bei der High-Time-Resolution werden die Daten in wesentlich kleineren Zeitabständen gesendet. Diese Funktion bringt jedoch den Server oft an die Grenzen seiner Rechenleistung.

3.1.10. /LUC46/ Configure Language Settings

Über Languagefiles soll die Sprache des MultiCastors 2.0 geändert werden können. Es wird ein System bereitgestellt mit dem diese Files erstellt und geladen werden können.

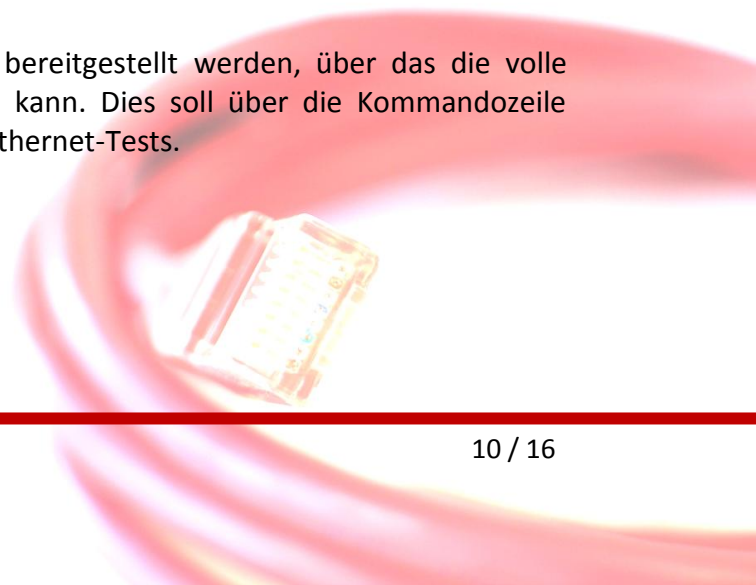
3.1.11. /LUC50/ Multi-Instance-Ability

Es sollen mehrere Programminstanzen gleichzeitig gestartet werden können. Die verschiedenen Instanzen können über die Loopback-Funktion auch lokal untereinander kommunizieren.

Des Weiteren ist das Fenstermanagement des MultiCastors 2.0 so aufgebaut, dass diese Instanzen eindeutig voneinander unterschieden werden können.

3.1.12. /LUC60/ Functional Testing

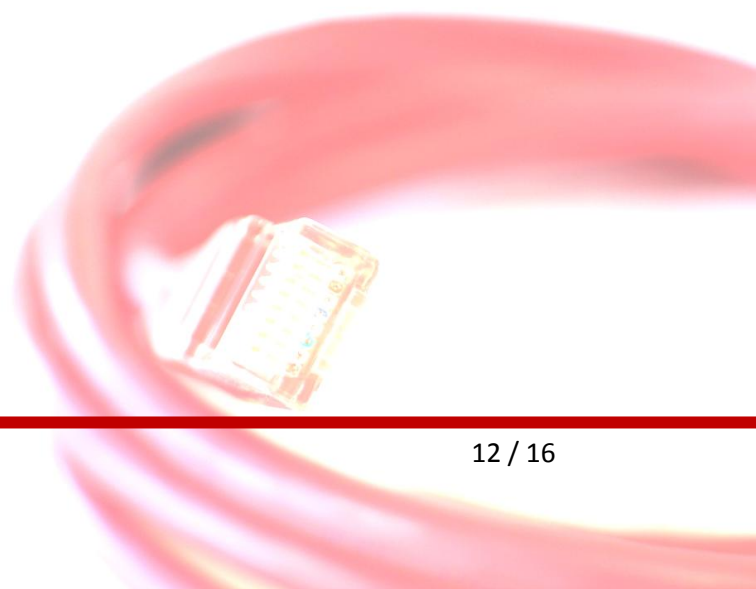
Es soll ein auf STAF/STAX beruhendes System bereitgestellt werden, über das die volle Funktionalität des Programmes geprüft werden kann. Dies soll über die Kommandozeile geschehen. Dazu gehören Config-, Logging- und Ethernet-Tests.



3.2. Anforderungen

Nummer	Anforderung
/LF10/	MultiCastor 2.0 muss das Multicast-Registrierungsprotokoll nach IEEE802.1ak (MMRP) unterstützen. Es muss sowohl ein Betrieb zwischen Sender und Empfänger mit einem zwischengeschalteten MMRP-fähigen Netzwerk als auch direkt Back-to-Back möglich sein.
/LF11/	Empfängerseitig muss für jede MAC-Adresse ein „ApplicantOnly“-ähnlicher Zustandsautomat implementiert werden.
/LF12/	Im Multicastor 2.0 muss das Senden nach IEEE802.1ak implementiert werden. Für die Kommunikation auf Layer 2 wird nach Möglichkeit die Jnet PCap API verwendet.
/LF20/	Der MultiCastor 2.0 soll kontrolliert auf unumgängliche Probleme reagieren und das Logging in diesem Zusammenhang erweitert werden
/LF21/	Es muss auf die Ressourcengrenze im Heap geachtet werden und entsprechend skaliert, geloggt und der User gewarnt werden
/LF22/	Ein Netzwerkabriss muss kontrolliert abgefangen werden
/LF22.1/	Ein Netzwerkabriss muss geloggt werden
/LF22.2/	Der User muss über den Netzwerkabriss informiert werden
/LF22.3/	Es muss versucht werden, die Verbindung wieder herzustellen
/LF30/	Die Usability der Benutzeroberfläche muss überarbeitet werden
/LF31/	Überarbeitung des „Tab-Managements“
/LF31.1/	Tabs sollen verschiebbar werden
/LF31.2/	Tabs sollen einzeln ein- und ausblendbar sein
/LF31.3/	Titel der Tabs sollen verändert werden, damit die Tabs eindeutig identifiziert werden können
/LF32/	Aus den Fenstertiteln sollen einzelne Instanzen eindeutig identifizierbar sein
/LF33/	Es sollen die Buttons überarbeitet werden, um ein schlankeres Design zu bekommen
/LF34/	IPv4- und IPv6-Fenster sollen sinnvoll zusammengelegt werden
/LF35/	Der About Dialog soll überarbeitet werden
/LF35.1/	Die Bilder sollen durch eine „Contributor“ Übersicht (wie „Wireshark“) ersetzt werden
/LF35.2/	Genauere Versionsangaben sollen eingefügt werden
/LF36/	Es soll einen MultiCastor 1.5 geben, der alle Änderungen an der GUI enthält, allerdings auf neue Funktionalitäten verzichtet (MMRP)
/LF40/	Die Konfigurationsmöglichkeiten müssen angepasst werden
/LF41/	Die Konfiguration der Sender und Receiver soll von der Konfiguration des MultiCastors selbst getrennt sein
/LF42/	Config-Files können modular gespeichert werden
/LF43/	Einstellungen können modular aus den Config-Files geladen werden
/LF44/	Sprachunterstützung von Deutsch und Englisch
/LF44.1/	Die Sprache soll über ein Sprachfile geladen werden können
/LF44.2/	Eine Dokumentation und ein Grundgerüst um weitere Sprachen einzubauen soll bereitgestellt werden

Nummer	Anforderung
/LF50/	Die About und Help Dialoge sollen überarbeitet werden
/LF51/	Deutsche Dokumentation wird entsprechend der neuen Funktionen überarbeitet und angepasst
/LF52/	Englischsprachiges Manual PDF mit ausführlichem Inhalt
/LF53/	Historienkapitel mit Bildern der Ersteller im Manual anhängen
/LF60/	Überarbeitung der Sender und Receiver
/LF61/	Sender sollen eine für MultiCastor-Instanzen eindeutige Signatur mitsenden
/LF62/	Receiver sollen Signaturen erkennen und damit verschiedene MultiCastor-Ströme voneinander unterscheiden können
/LF63/	Im Receiver muss ein Receive-Counter pro Multicast-Strom eingefügt werden.
/LF70/	Es müssen mehr Informationen in der Konfigurationsdatei abgespeichert werden
/LF71/	Es wird das Datum gespeichert
/LF72/	Es wird die PC-Kennung gespeichert
/LF80/	Für MultiCastor 2.0 müssen verschiedene automatisierte Testszenarien beschrieben und eine STAF/STAX-Unterstützung nachgewiesen werden
/LF81/	Test der einzelnen Komponenten von MultiCastor 2.0
/LF81.1/	Config-File einlesen und davon abhängige Tests
/LF81.2/	Log-File auswerten
/LF81.3/	gesendete/empfangene Pakete auswerten
/LF90/	Erstellung einer automatisierten Regressionstestsuite
/LF91/	Einzelne Tests sollen in eine Testsuite eingebaut werden können
/LF92/	Einzelne Tests sollen aus einer Testsuite entfernt werden können
/LF93/	Testsuites sollen gestartet werden können
/LF93.1/	Die Testsuite soll möglichst vollständig abgearbeitet werden, ohne dass es zu Abbrüchen kommt
/LF93.2/	Ergebnisse der Einzeltests sollen dem Benutzer angezeigt werden



3.3. Bekannte Bugs

3.3.1. /BUG 10/ Aboutfenster schließen

Das Aboutfenster schließt sich beim drücken auf das „X“ nicht.

(Linux Ubuntu 10.04)

3.3.2. /BUG 20/ Active/Inactive Button Fehlfunktion

Nach allgemeinem Verständnis sollte ein Multicastor beim drücken auf den Active/Inactive Button in der Multicastor Config ebenfalls umschalten. Leider erfolgt dies nicht. Ab- und anschaltbar ist der Multicastorthread nur über den "State" in der Config Übersicht

(Linux Ubuntu 10.04)

3.3.3. /BUG 30/ GUI Option Menu Schrift

Beim klick auf Menü ist außer "Userlevel" kein Auswahlpunkt ohne Mousehover lesbar.

(Linux Ubuntu 10.04)

3.3.4. /BUG 40/ Logfile

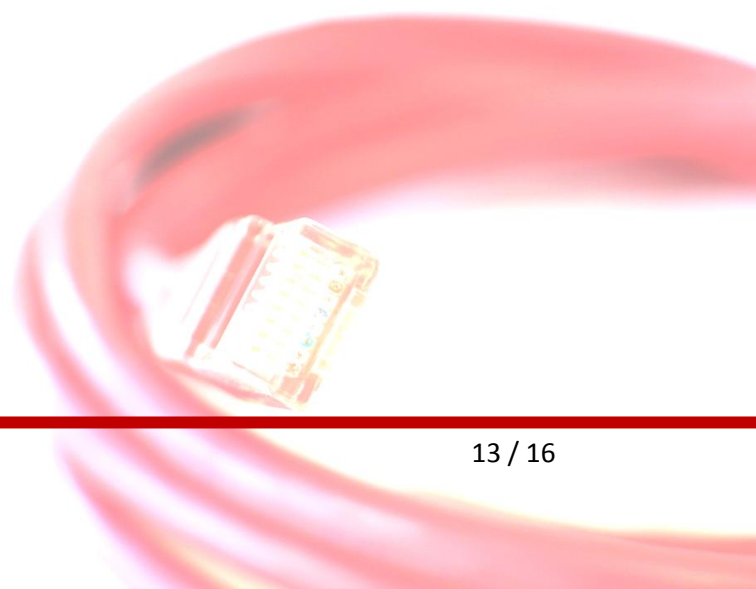
Ungewollte Zeilenumbrüche im Logfile (Ansicht mit Notepad) verhindern die Lesbarkeit der Datei. Mitten im Datum wird umgebrochen. In einigen Fällen gar nicht. Eventuell nur unter Linux korrekte Darstellung (nicht geprüft).

(Windows 7)

3.3.5. /BUG 50/ Partielles Laden von Konfigurationen

Die Beschreibungen Checkboxes für partielles Laden werden im Load Config Fenster nicht vollständig anzeigt. Dieser Fehler wird mit /LUC 42/ weiter geführt. Die Funktion wird neu gestaltet. Die fehlerhaften Kontrollkästchen werden entfernt.

(Windows XP, Windows Vista, Windows 7)



4. Produktdaten

4.1. /LD10/ Das Logfile

Das Logfile beinhaltet alle Informationen, Warnungen und Fehlermeldungen, die das Programm über die Laufzeit generiert.

4.2. /LD20/ Das Programm-Configurationsfile

Eine XML-Datei, in der alle Programmeinstellungen gespeichert werden (bspw. Sprache oder Autospeichern).

4.3. /LD30/ Das Predefined-Value-Configurationfile

Eine XML-Datei, in der Sender- und Receiverkonfigurationen gespeichert sind

4.3.1. /LD31/ Multicastor.xml

Automatisch erstellte XML-Datei mit der aktuellen Sender- und Receiverkonfiguration.

4.3.2. /LD32/ Inhalt des Configurationsfiles

Ein Sender bzw. Receiver besteht aus:

- Multicast-IP-Adresse
- Netzwerk-Interface
- UDP-Port
- Paketrate
- Paketlänge
- „time to live“

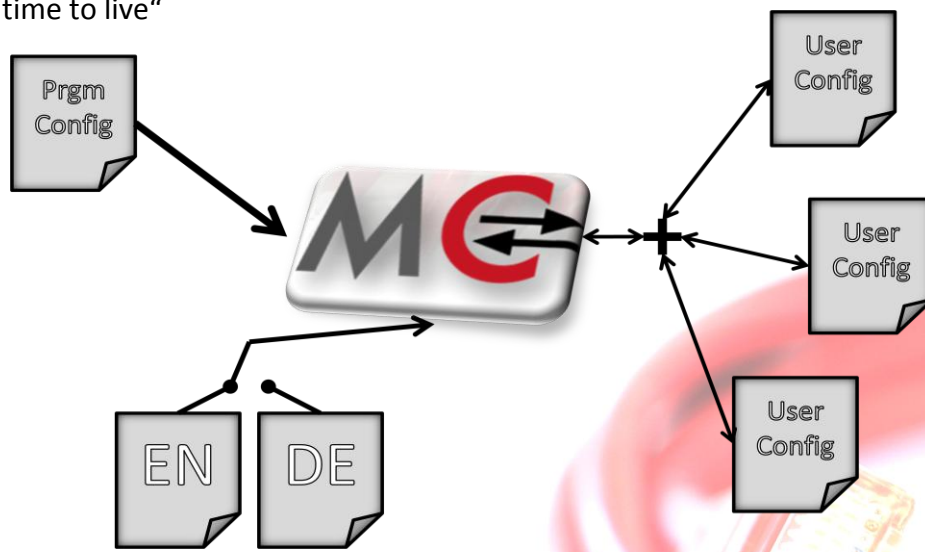


Abbildung 6 - Kommunikation des MultiCastor 2.0 mit Programmdateien

5. Produktcharakteristiken

5.1. Systemumgebung

Der MultiCastor stellt einige Anforderungen an die Hardware- und Softwareumgebung. Im Folgenden werden die genauen Abhängigkeiten und Voraussetzungen aufgelistet.

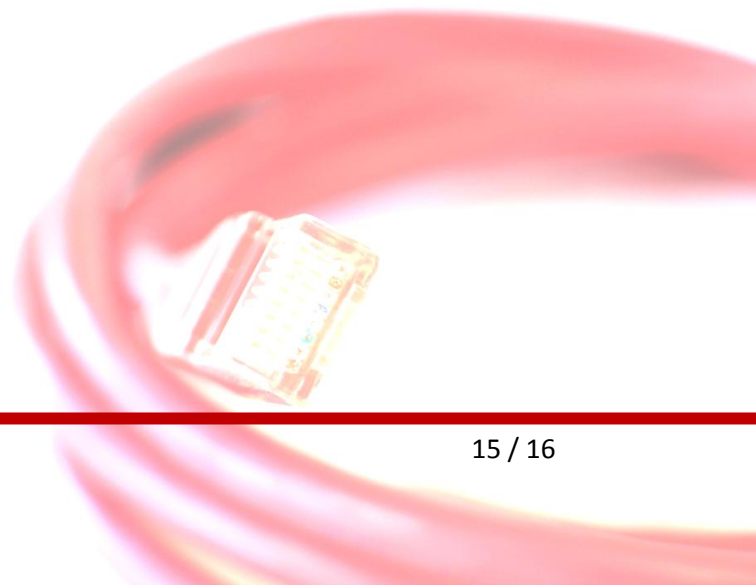
Das Multicastor-Tool läuft sowohl auf Windows als auch auf unixbasierten Betriebssystemen. Das System muss über eine intakte Netzwerkverbindung oder eine Loopback-Funktion verfügen, um Tests durchzuführen.

5.1.1. Hardwareumgebung

Die Hardware muss so gewählt werden, dass eine Java-Laufzeitumgebung auf Windows oder einem unixbasierten Betriebssystem installiert und dauerhaft betrieben werden kann. Die Hardware muss über eine Netzwerkschnittstelle verfügen auf die über eine Softwareschnittstelle zugegriffen werden kann. Außerdem wird ein MMRP-fähiger Switch benötigt, um MMRP-Anfragen verarbeiten zu können.

5.1.2. Softwareumgebung

Der MultiCastor wurde und wird komplett in der Programmiersprache Java V1.6 umgesetzt. Es werden auch bei einer Weiterentwicklung des Multicastor nur die Standard-Java-Bibliotheken verwendet. Demnach muss eine Java-Laufzeitumgebung der Version 1.6 auf dem Hostsystem installiert und verfügbar sein.



Dokumentversionen

Versionsnr	Datum	Autor(en)	Kommentar
V 0.1	12.08.2011	Stuckert/ Rentschler	Dokument angelegt
V 0.2	27.09.2011	Traub	Template/Formatvorlage erstellt
V 0.3	29.09.2011	Koralewski	Inhaltliche Ausarbeitung „Produkteinsatz“
V 0.4	30.09.2011	Haase	Inhaltliche Ausarbeitung „Produktfunktionen“
V 0.5	30.09.2011	Hauschild	Inhaltliche Ausarbeitung „Produktdaten“
V 0.6	01.10.2011	Westphal	Inhaltliche Ausarbeitung „Produktcharakteristik“
V 1.0 (beta)	02.10.2011	Hauschild	Abschließende Formulierung Zusammenführung der Inhalte
V 1.0 (beta 2)	05.10.2011	Traub	Abschlusskorrektur und Formatierung.
V 1.0	09.10.2011	Traub	Umsetzung angesprochener Punkte aus dem Review im Plenum

Lizenz/License:

© Fäßler, Haase, Hauschild, Koralewski, Traub, Westphal

Dokumentversion/Document Version:

Titel: Customer Requirements Specification

Version: V 1.0 (9. Oktober 2011)

Autoren/Authors Projektteam/Project team:

- Jonas Traub (Projektleiter)
- Filip Haase (Leading Engineer)
- Matthis Hauschild (Documentation)
- Fabian Fäßler (Engineer/Tester, Expert on STAF/STAX)
- Christopher Westphal (Engineer/Tester, Expert on usability)
- Sebastian Koralewski (Engineer/Tester, Expert on MMRP)